Spectral Methods

Notes for Summer School on General Relativistic Hydrodynamics

July 2003 — Saul Teukolsky

1 Introduction

Spectral methods are a very powerful tool for solving PDEs. When they can be used, they are the method of choice if you need high spatial resolution in multidimensions. For a 3-d finite difference code, increasing the resolution by a factor of 2 in each dimension requires a factor of 8 more grid points, and improves the error typically by a factor of 4. In a spectral code, a similar increase in resolution can often give an improvement of a factor of 10^8.

Spectral methods work well for smooth solutions. Discontinuities like shocks are bad—don't even try spectral methods. Even mild non-smoothness (like a discontinuity in some high-order derivative of the solution) spoils the convergence of spectral methods.

The key difference between finite difference methods and spectral methods is that in finite difference methods you approximate the *equation* you are trying to solve, whereas in spectral methods you approximate the *solution* you are trying to find. While finite differencing replaces the continuum equation by an equation on grid points, a spectral method expresses the solution as a truncated expansion in a set of basis functions:[1]

$$f(x) \simeq f_N(x) = \sum_{n=0}^{N} a_n \phi_n(x) \tag{1.1}$$

Different choices of basis functions and methods of computing a_n give different flavors of spectral methods.

1.1 Resources

Fornberg (1996) is a nice short book to get started. But there's not enough to get into serious large-scale problems.

Boyd (2001) is the best single book—very complete, and not too formal. It's also available online.

Canuto et al. (1988) is the standard reference for fluid dynamics applications. They also cover a lot of the general stuff that's applicable to all problems.

Gottlieb & Orszag (1977) was the first book, and is still useful.

None of the books deals properly with hyperbolic systems of equations in general. Bjørhus (1995) is the best place to consult.

2 Example

We illustrate the idea of spectral methods with an example. Consider the one-sided wave equation (advective equation) in 1-d:

$$\frac{\partial u}{\partial t} = \frac{\partial u}{\partial x} \tag{2.1}$$

[1] Later we will see that the most important category of spectral methods, so-called pseudospectral methods, can also be thought of as a special kind of high-order finite difference method.

with periodic boundary conditions on $[0, 2\pi]$ and initial condition

$$u(t = 0, x) \;=\; f(x) \tag{2.2}$$

You get the analytic spectral solution by expanding u in a Fourier series,

$$u(t, x) = \sum_{n=-\infty}^{\infty} a_n(t) e^{inx} \tag{2.3}$$

Substituting this expansion into eqn. (2.1) gives

$$\frac{da_n}{dt} = ina_n \tag{2.4}$$

with solution

$$a_n(t) = a_n(0) e^{int} \tag{2.5}$$

You get $a_n(0)$ from the initial condition: expand

$$f(x) = \sum_{n=-\infty}^{\infty} f_n e^{inx} \tag{2.6}$$

from which you see that

$$a_n(0) = f_n \tag{2.7}$$

For example, suppose

$$f(x) = \sin(\pi \cos x) \tag{2.8}$$

which gives the analytic solution

$$u(t, x) = \sin[\pi \cos(x + t)] \tag{2.9}$$

The spectral coefficients in the solution (2.3) are

$$\begin{aligned} a_n(0) &= \frac{1}{2\pi} \int_0^{2\pi} \sin(\pi \cos x) e^{-inx} dx \\ &= (-1)^{(n-1)/2} J_n(\pi), \qquad n \text{ odd} \end{aligned} \tag{2.10}$$

There are three properties of the exponential functions that are crucial for this analytic spectral solution, which is just the separation of variables technique:

1. They are a complete set of basis functions.
2. Each basis function by itself obeys the boundary conditions.
3. They are eigenfunctions of the operator in the problem, d/dx.

As we'll see, only property (1) is essential for numerical spectral methods.

Suppose now we truncate this solution at $n = N$. How well does $u_N(t, x)$ approximate the exact solution? One measure is the root-mean-square error,

$$\begin{aligned} L_2 &= \left[\frac{1}{2\pi} \int_0^{2\pi} |u(t, x) - u_N(t, x)|^2 \, dx \right]^{1/2} \\ &= \left[\frac{1}{2\pi} \int_0^{2\pi} \left| \sum_{|n|>N} a_n(0) e^{inx} e^{int} \right|^2 dx \right]^{1/2} \\ &= \left[\sum_{|n|>N} |a_n(0)|^2 \right]^{1/2} \end{aligned} \tag{2.11}$$

Now $J_n(\pi)$ goes to zero exponentially as $n \to \infty$, so the error decreases *exponentially* with N for any $t \geq 0$. This is the key feature of a good spectral method, one you should always strive for. By contrast, a second order finite difference method has an error that scales as $1/N^2$.

This exponential convergence of spectral methods sets in when one has resolved the main features of the solution. In Appendix A we show that you need about π basis functions per wavelength to resolve the solution. (You can see that above, where we need $n \gtrsim \pi$ for the Bessel functions to start becoming small.) We also show that, by contrast, second order accurate finite differencing needs about 20 points per wavelength for 1% accuracy. Moreover, once the solution is resolved, the accuracy improves much more quickly with spectral methods.

3 Choice of Basis Functions

It turns out you can't simply use Fourier series as basis functions for all problems—it depends on the boundary conditions. Here is a recipe that will take care of 99% of cases you'll encounter:

- If the solution is periodic, use Fourier series.
- If the solution is not periodic and the domain is a square or a cube, use Chebyshev polynomials along each dimension.
- If the domain is spherical, use spherical harmonics.

If you want to understand why these choices work, in Appendix B we prove the key result: *eigenfunction expansions based on singular Sturm-Liouville problems converge at a rate governed by the smoothness of f, not by any special boundary conditions that f satisfies.*

Fourier series are used when there *are* special boundary conditions—periodicity. For non-periodic solutions, we turn to orthogonal polynomials like Legendre polynomials, Chebyshev polynomials, etc., which are eigenfunctions of singular Sturm-Liouville problems. For example, in the notation of Appendix B, Chebyshev polynomials have

$$p(x) = \sqrt{1-x^2}, \qquad w(x) = 1/\sqrt{1-x^2}, \qquad q(x) = 0, \qquad \lambda_n = n^2, \qquad -1 \leq x \leq 1 \tag{3.1}$$

The reason for the popularity of Chebyshev polynomials is that they are really just trigonometric functions whose argument θ has been mapped by $x = \cos\theta$:

$$T_n(x) = \cos(n\theta), \qquad x = \cos\theta \tag{3.2}$$

Thus an expansion in Chebyshev polynomials can be evaluated efficiently by the FFT. Moreover, one can show that the derivatives of such an expansion can also be evaluated by FFT techniques.

For spherical domains, spherical harmonics are products of Legendre functions in $\cos\theta$ and Fourier series in ϕ.

4 Computing the Expansion Coefficients

How do we compute the a_n? There are 3 ways:

1. Tau method.
2. Galerkin method. (This is what we used in the Mathieu equation problem in HW #8. It is also what you typically do when you separate variables in solving a pde, as in §2, or in quantum mechanics when you find expansion coefficients for $\psi(x) = \sum c_n \phi_n(x)$.)

3. Collocation or Pseudospectral (PS) Method.

See Fornberg (1996) Appendix B for an example done all 3 ways.

We will focus only on PS. PS has the big advantage that it is easy to implement for nonlinear problems. Instead of working with the spectral coefficients, you work with the values of the solution at the special grid points associated with the basis functions (typically, the Gaussian quadrature points). These are called the *collocation points.* Often we say we are working with the solution in *physical space* as opposed to in *spectral space*.

A PS method is an *interpolating* method: think of the representation

$$y(x) = \sum_{n=0}^{N} a_n \phi_n(x) \tag{4.1}$$

as a polynomial that interpolates the solution. Require this interpolating polynomial to be exactly equal to the solution at the $N+1$ collocation points. If we do things right, then as $N \to \infty$, the errors in between the points tend to zero exponentially fast.

4.1 Spectral Methods and Gaussian Quadrature

Recall the formula for Gaussian quadrature:

$$\int_a^b y(x)w(x)\,dx \approx \sum_{i=0}^{N} w_i y(x_i) \tag{4.2}$$

Here $w(x)$ is the so-called *weight function* that typically factors out some singular behavior of the integrand, leaving $y(x)$ as a smooth function. The formula is derived by choosing the $2N+2$ weights and abscissas, w_i and x_i, by requiring that the formula be exact for the polynomials $1, x, x^2, \ldots, x^{2N+1}$. (Don't be confused by the notation: there is no direct relationship between w_i and $w(x)$.) As shown in the textbooks, Gaussian quadrature is related to the orthogonal polynomials $\phi_n(x)$ with the given weight function:

$$\langle \phi_n | \phi_m \rangle \equiv \int_a^b \phi_n(x)\phi_m(x)w(x)\,dx = \delta_{mn} \tag{4.3}$$

The abscissas x_i turn out to be the $N+1$ roots of $\phi_{N+1}(x)$, and the weights w_i are given by a formula you can find in the textbooks.

We can use Gaussian quadrature to define the discrete inner product of two functions:

$$\langle f | g \rangle_{\mathrm{G}} \equiv \sum_{i=0}^{N} w_i f(x_i) g(x_i) \tag{4.4}$$

Here the subscript G stands for Gaussian.

An important property of Gaussian quadrature is the *discrete orthogonality relation*

$$\langle \phi_n | \phi_m \rangle_{\mathrm{G}} = \delta_{mn}, \qquad m+n \le 2N+1 \tag{4.5}$$

Proof: Eqn. (4.5) is the Gaussian quadrature version of eqn. (4.3). By assumption, the integrand $\phi_n(x)\phi_m(x)$ of eqn. (4.3) is a polynomial of degree $m+n \le 2N+1$. But Gaussian quadrature is arranged to integrate polynomials of degree $\le 2N+1$ exactly. QED.

Now suppose we approximate $y(x)$ by the PS interpolating polynomial

$$P_N(x) = \sum_{n=0}^{N} \bar{a}_n \phi_n(x) \tag{4.6}$$

where the collocation points are chosen to be the Gaussian quadrature points:

$$P_N(x_i) = y(x_i), \qquad i = 0, 1, \ldots, N \tag{4.7}$$

This is always possible, since the interpolating polynomial through $N + 1$ points is a polynomial of degree N, and the functions up to $\phi_N(x)$ are a basis for such polynomials. The perhaps unexpected result is that the coefficients $\{\bar{a}_n\}$ of the expansion (4.6) are given *exactly* by the Gaussian quadrature

$$\bar{a}_n = \langle y|\phi_n\rangle_{\mathrm{G}} \tag{4.8}$$

To see this, take the discrete inner product of both sides of eqn. (4.6) with ϕ_m:

$$\langle P_N|\phi_m\rangle_{\mathrm{G}} = \sum_{n=0}^{N} \bar{a}_n \langle\phi_n|\phi_m\rangle_{\mathrm{G}} \tag{4.9}$$

If we use the discrete orthogonality relation (4.5), the right-hand side evaluates to $\bar{a}_m$. On the left-hand side, we can replace $P_N(x_i)$ in the Gaussian quadrature by $y(x_i)$ since P_N is the interpolating polynomial. Hence the result follows.

Now comes the key point. The actual spectral expansion of $y(x)$ is

$$y(x) = \sum_{n=0}^{\infty} a_n \phi_n(x) \tag{4.10}$$

where the *exact* spectral coefficients are

$$a_n = \langle y|\phi_n\rangle = \int_a^b y(x)\phi_n(x)w(x)\,dx \tag{4.11}$$

The PS expansion coefficients $\bar{a}_n$ are the exact expansion coefficients of $P_N(x)$, the interpolating polynomial (4.6). The relation between the exact spectral coefficients and the PS expansion coefficients follows from eqn. (4.8):

$$\begin{aligned}
\bar{a}_n &= \langle y|\phi_n\rangle_{\mathrm{G}} \\
&= \sum_{m=0}^{\infty} a_m \langle\phi_m|\phi_n\rangle_{\mathrm{G}} \qquad \text{(using eqn. 4.10)} \\
&= \sum_{m=0}^{N} a_m \langle\phi_m|\phi_n\rangle_{\mathrm{G}} + \sum_{m>N} a_m \langle\phi_m|\phi_n\rangle_{\mathrm{G}} \\
&= a_n + \sum_{m>N} a_m \langle\phi_m|\phi_n\rangle_{\mathrm{G}}
\end{aligned} \tag{4.12}$$

Thus, since for large N the exact spectral coefficients give an exponentially good approximation to $y(x)$, so do the PS coefficients. (This key result, which is what makes the PS method so powerful, is due to Lanczos in 1938. Lanczos made many important contributions to applied mathematics and also to GR theory.) By the way, this is the reason for the name pseudospectral method: we use coefficients that are not the actual spectral coefficients, but are very close to them. From now on we won't bother to distinguish between the two sets of coefficients; we just write a_n for either a_n or $\bar{a}_n$.

The Gaussian quadrature collocation points, the roots of $\phi_{N+1}(x)$, all lie inside the interval (a, b), away from the endpoints. There is another version of Gaussian quadrature that includes the two endpoints of the interval. This is called Gauss-Lobatto quadrature, and the collocation points are the Gauss-Lobatto points. These points are as effective as the ordinary Gaussian points, and are more convenient when you need to impose boundary conditions at the endpoints.

As a slight digression, you may be under the mistaken impression that the only advantage of Gaussian integration over integration with equally spaced points is that its degree of exactness is $2N + 1$ as opposed to N, the maximum you can get with only the $N + 1$ weights at your disposal. In fact, however, the main advantage of Gaussian integration is that it converges exponentially with N for smooth functions. You can see this explicitly from the above formulas by setting $m = 0$ in eqn. (4.8):

$$\bar{a}_0 = \phi_0 \sum_{i=0}^{N} w_i y(x_i) \tag{4.13}$$

where ϕ_0 is a constant. But this converges exponentially to the expression given by eqn. (4.11):

$$a_0 = \phi_0 \int_a^b y(x) w(x)\, dx \tag{4.14}$$

as claimed.

How do Fourier series fit into this discussion? After all, the collocation points are equally spaced (usually $x_j = 2\pi j/N$, $j = 0, \ldots, N-1$). But in fact these are the correct collocation points if we think of Fourier series as interpolating $y(x)$ by a *trigonometric* polynomial. The corresponding Gaussian quadrature (using the equally spaced points) is the midpoint rule, and the Gauss-Lobatto quadrature, which includes the endpoints, is the trapezoidal rule. The textbooks tell you that the midpoint and trapezoidal rules are low-order methods. This is true for arbitrary functions. But if you apply them to *periodic* functions they are in fact exponentially convergent, like any self-respecting Gaussian quadrature method should be.

4.2 Cardinal Functions

You can write *any* polynomial interpolation formula for a function $f(x)$ as

$$P_N(x) = \sum_{i=0}^{N} f(x_i) C_i(x) \tag{4.15}$$

where the $C_i(x)$ are called *cardinal functions*. They are polynomials of degree N that satisfy

$$C_i(x_j) = \delta_{ij} \tag{4.16}$$

i.e., $C_i(x)$ is 1 at the ith collocation point and zero at all the others.

One explicit representation of cardinal functions comes from the formula for Lagrange interpolation (see *Numerical Recipes*, eq. 3.1.1):

$$C_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{N} \frac{x - x_j}{x_i - x_j} \tag{4.17}$$

If you substitute this in eqn. (4.15), it is just the Lagrange interpolation formula. Each choice of basis functions implies a corresponding choice of collocation points x_j, and so a corresponding choice of cardinal functions by eqn. (4.17).

There are other equivalent ways of writing $C_i(x)$. For example, if $\phi_n(x)$ is a set of orthogonal polynomials, and the collocation points are the zeros of $\phi_{N+1}(x)$ (Gaussian quadrature points), then $C_i(x)$ is almost $\phi_{N+1}(x)$, except $\phi_{N+1}(x)$ vanishes at *all* the grid points. Since near $x = x_i$

$$\phi_{N+1}(x) = \phi_{N+1}(x_i) + (x - x_i)\phi'_{N+1}(x_i) + \cdots \tag{4.18}$$

we divide out the zero at $x = x_i$ by writing

$$C_i(x) = \frac{\phi_{N+1}(x)}{(x - x_i)\phi'_{N+1}(x_i)} \tag{4.19}$$

You might be nervous about using very high order polynomial interpolation to represent your solution. Remember the Runge phenomenon (HW #2): If the grid points are *equally spaced*, then the error in $P_N(x)$ can tend to infinity as $N \to \infty$. But recall that the error shows up near the endpoints of the interval—the middle is fine. The fix is to make the points more concentrated toward the endpoints, exactly what choosing the Gaussian points does.

In practice you don't need to know any of the formulas like (4.17) or (4.19). The books have formulas for the $C_i(x)$ for all the standard basis functions if you are curious. What you do need to use are the derivatives of the cardinal functions, the *differentiation matrices* (see § 4.4).

4.3 Spectral vs. Grid Point Representation

Let's contrast the representations of the solution of

$$\mathcal{L}y = f \tag{4.20}$$

in spectral space and in physical space. Assume that $\mathcal{L}$ is a linear differential operator for simplicity.

Spectral Space	**Physical Space**
$y(x) = \sum_{n=0}^{N} a_n \phi_n(x)$	$y(x) = \sum_{j=0}^{N} y_j C_j(x)$
$\sum_{n=0}^{N} a_n \mathcal{L}\phi_n(x) = f(x)$	$\sum_{j=0}^{N} y_j \mathcal{L}C_j(x) = f(x)$

Impose at collocation points only:

$\sum_{n=0}^{N} a_n \mathcal{L}\phi_n(x_j) = f(x_j)$	$\sum_{j=0}^{N} y_j \mathcal{L}C_j(x_i) = f(x_i)$
i.e., $La = f$, where $L_{jn} = \mathcal{L}\phi_n(x_j)$	i.e., $L^{(c)}y = f$, where $L^{(c)}_{ij} = \mathcal{L}C_j(x_i)$

The two representations are related as follows: To go from grid point values to spectral coefficients you project $y(x)$ along each basis function:

$$a_i = \langle \phi_i \,|\, y \rangle \tag{4.21}$$

$$= \sum_j w_j \phi_i(x_j) y_j \qquad \text{(doing the integral by Gaussian quadrature)} \tag{4.22}$$

That is,

$$a = My, \qquad \text{where} \quad M_{ij} = \phi_i(x_j)w_j \tag{4.23}$$

Thus the relation in spectral space $La = f$ becomes $LMy = f$. But in physical space $L^{(c)}y = f$, so

$$L^{(c)} = LM \tag{4.24}$$

with inverse

$$L = L^{(c)}M^{-1} \tag{4.25}$$

Note also that eqn. (4.23) implies

$$y = M^{-1}a \tag{4.26}$$

Since $y = \sum a_n\phi_n$, we see that M^{-1} is the matrix that sums the spectral series to get the grid point values, i.e.

$$M^{-1}_{ij} = \phi_j(x_i) \tag{4.27}$$

You can check that these relations are all consistent:

$$\begin{aligned} MM^{-1}|_{ij} &= \sum_k M_{ik}M^{-1}_{kj} \\ &= \sum_k [\phi_i(x_k)w_k][\phi_j(x_k)] \\ &= \langle\phi_i|\phi_j\rangle_{\mathrm{G}} \\ &= \delta_{ij} \qquad \text{(by discrete orthogonality)} \end{aligned} \tag{4.28}$$

In practice, the transformations (4.23) and (4.26) are often done with FFTs for Fourier or Chebyshev basis functions if N is large. For simple programs, just do matrix multiplication.

4.4 Differentiation Matrices

We've seen above that the key ingredient in the PS method is to form

$$L^{(c)}_{ij} = \mathcal{L}C_j(x_i) \tag{4.29}$$

which involves taking derivatives of the cardinal functions at the collocation points. Consider the first derivative ∂_x. You then need the matrix

$$D^{(1)}_{ij} = \partial_x C_j(x_i) \tag{4.30}$$

This quantity can be computed ahead of time and stored. Then to compute the vector of first derivatives at the grid points, just do a matrix multiplication:

$$\frac{\partial y}{\partial x} \quad \longleftrightarrow \quad \sum_{j=0}^{N} D^{(1)}_{ij} y_j \tag{4.31}$$

Similarly one can define the second derivative matrix $D^{(2)}_{ij}$, and so on.

Note:
The matrix multiplication in eqn. (4.31) requires $O(N^2)$ operations. For Fourier basis functions e^{ikx}, one can compute the derivative alternatively as follows:

$$
\begin{aligned}
y &\overset{\text{FFT}}{\rightarrow} a \\
a &\rightarrow ika \\
ika &\overset{\text{inverse FFT}}{\rightarrow} y'
\end{aligned}
$$

For Chebyshev basis functions there is a simple $O(N)$ recurrence relation in the middle step to get the coefficients for the derivative from the coefficients for the function (see, e.g., § 2.4.2 of Canuto et al. (1988)). This procedure is $O(N \log N)$. However, it is typically faster than the $O(N^2)$ matrix multiplication only for $N \gtrsim 16 – 128$, depending on the computer. So just use matrix multiplication for simple programs.

4.4.1 Options for Computing Differentiation Matrices

1. Derive the formulas by differentiating the Lagrange polynomial representation eqn. (4.17).
2. Derive the formulas by differentiating the basis function representation eqn. (4.19).
3. Look up the explicit formulas that people have derived for the various basis functions in books, e.g. Chapter 2 of Canuto et al. (1988).
4. Use the program given by Fornberg (1998). This algorithm computes any order of differentiation matrix given only a set of grid points $\{x_i\}$. A translation of Fornberg's Fortran program into C++ is given in Appendix C.

Obviously, the last choice is the easiest.

4.4.2 A Note on Interpolation

Often you want to evaluate the solution at points that are not the collocation points. This requires an interpolation. To preserve the full spectral accuracy, you want to use all the information in the solution. However, it is not necessary to transform the solution to spectral space and then evaluate the representation (1.1) at the desired point, e.g. by Clenshaw's method. Simply use the interpolation formula (4.15). A simple way to do this is to use Fornberg's program in Appendix C, which will return the interpolation weights $C_i(x_k)$ for any set of target points x_k when m, the second dimension of c in the code, is zero. So interpolating to a set of points can again be done as a matrix multiplication.

4.5 PS Derivatives vs. FD Derivatives

Consider finite difference approximations for d/dx at the center of an equally spaced grid, for example

$$
\begin{aligned}
hf'(x) &= -\tfrac{1}{2}f(x-h) + \tfrac{1}{2}f(x+h) + O(h^2) && (4.32)\\
&= \tfrac{1}{12}f(x-2h) - \tfrac{2}{3}f(x-h) + \tfrac{2}{3}f(x+h) - \tfrac{1}{12}f(x+2h) + O(h^4) && (4.33)\\
&= \quad \ldots && (4.34)
\end{aligned}
$$

For centered differences like these, the limit as $N \to \infty$ of the weights (coefficients of f) is finite. But for one-sided approximations, or partially one-sided approximations, the weights diverge. Since one has to use such approximations near the end points of the grid, it's not surprising that high order FD approximations have large errors near the boundaries.

But suppose the grid points are not equally spaced. In particular, suppose they are closer together near the end points, like the Gaussian quadrature points. Then the FD approximation is convergent as $N \to \infty$.

The PS method gives the exact derivative of the interpolating polynomial that passes through the data at the $N + 1$ grid points. You would get the same result for a FD method that uses all $N + 1$ grid points, This follows from uniqueness of the interpolating polynomial, a polynomial of degree N through all $N + 1$ points.

With this point of view, think of a PS method as a way to find high-order numerical approximations to derivatives at grid points. Then just like FD methods, satisfy the equation you want to solve at the grid points. If the equation has variable coefficients or nonlinearities, just multiply the functions at the grid points. (This is the big advantage over tau and Galerkin methods—handling nonlinearities in spectral space is complicated.)

5 An Example

This example comes from Appendix B of Fornberg (1996). Consider the equation

$$y'' + y' - 2y + 2 = 0, \qquad -1 \le x \le 1, \tag{5.1}$$

$$y(-1) = y(1) = 0 \tag{5.2}$$

The exact solution is

$$y(x) = 1 - \frac{e^x \sinh 2 + e^{-2x} \sinh 1}{\sinh 3} \tag{5.3}$$

Let's make an expansion in Chebyshev polynomials with $N = 4$:

$$y = \sum_{n=0}^{4} a_n T_n(x) \tag{5.4}$$

Choose the collocation points to be

$$x_i = \cos \frac{i\pi}{4}, \qquad i = 0, \dots, 4 \tag{5.5}$$

These are the "Gauss-Lobatto" points associated with Chebyshev polynomials, i.e., they include the end points. We always include the end points when we want to impose Dirichlet boundary conditions, that is, function values on the boundaries. Using one of the methods for finding differentiation matrices, we get

$$[D^{(1)}y]_i = \begin{bmatrix} -\frac{11}{2} & 4 + 2\sqrt{2} & -2 & 4 - 2\sqrt{2} & -\frac{1}{2} \\ -1 - \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} & \sqrt{2} & -\frac{1}{2}\sqrt{2} & 1 - \frac{1}{2}\sqrt{2} \\ \frac{1}{2} & -\sqrt{2} & 0 & \sqrt{2} & -\frac{1}{2} \\ -1 + \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} & -\sqrt{2} & -\frac{1}{2}\sqrt{2} & 1 + \frac{1}{2}\sqrt{2} \\ \frac{1}{2} & -4 + 2\sqrt{2} & 2 & -4 - 2\sqrt{2} & \frac{11}{2} \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} \tag{5.6}$$

and

$$[D^{(2)}y]_i = \begin{bmatrix} 17 & -20-6\sqrt{2} & 18 & -20+6\sqrt{2} & 5 \\ 5+3\sqrt{2} & -14 & 6 & -2 & 5-3\sqrt{2} \\ -1 & 4 & -6 & 4 & -1 \\ 5-3\sqrt{2} & -2 & 6 & -14 & 5+3\sqrt{2} \\ 5 & -20+6\sqrt{2} & 18 & -20-6\sqrt{2} & 17 \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} \tag{5.7}$$

Requiring that the differential equation hold at the interior collocation points x_k, $k = 1, 2, 3$, uses the middle 3 rows of these matrices. Enforcing the boundary conditions $y_0 = y_4 = 0$ means we don't need the first and last columns. So eqn. (5.1) gives

$$\begin{bmatrix} -16+\frac{1}{2}\sqrt{2} & 6+\sqrt{2} & -2-\frac{1}{2}\sqrt{2} \\ 4-\sqrt{2} & -8 & 4+\sqrt{2} \\ -2+\frac{1}{2}\sqrt{2} & 6-\sqrt{2} & -16-\frac{1}{2}\sqrt{2} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} -2 \\ -2 \\ -2 \end{bmatrix} \tag{5.8}$$

with solution

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} \frac{101}{350} + \frac{13}{350}\sqrt{2} \\ \frac{13}{25} \\ \frac{101}{350} - \frac{13}{350}\sqrt{2} \end{bmatrix} \tag{5.9}$$

The exact solution (5.3) gives for example $y(x = 0) = 0.52065$, compared with $y_2 = 0.52000$. Not bad for 5 grid points! The real point however, is that the error is about 10^{-16} for $N = 16$. With a second-order FD scheme, the error would go down by only a factor of 10 or so with this increase in N (see Fig. 1).

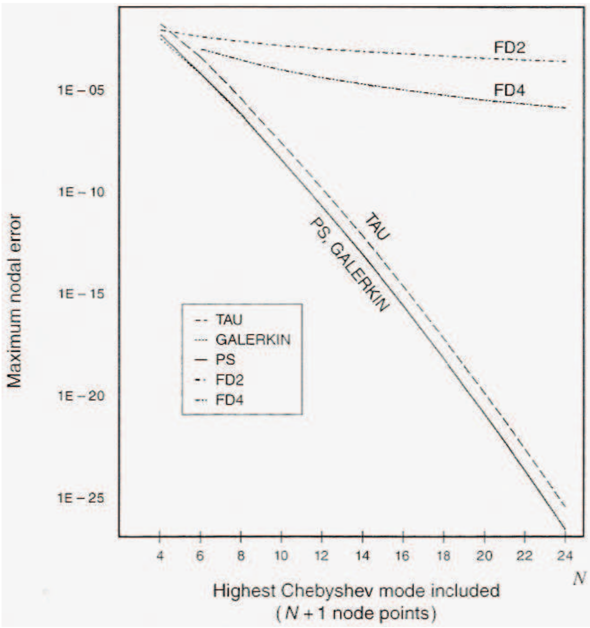

Figure 1: Maximum nodal errors for different methods applied to eqn. (5.1): comparison among three spectral implementations and equi-spaced FD methods of second and fourth order. (From Fornberg 1996.)

Exercise

Solve the same problem as above, but with the boundary condition

$$y'(1) = 0 \tag{5.10}$$

instead of $y(1) = 0$. One way of doing this is to set the first row of the $D^{(1)}$ matrix to zero. You will then have to include the point $x = 1$ with the interior collocation points. The exact solution for checking is

$$y = 1 - \frac{2e^{x+1} + e^{4-2x}}{2 + e^6} \tag{5.11}$$

Appendix

A Resolution of FD and Spectral Methods

Let's compare the resolution of finite difference and spectral methods by considering the eigenvalue problem for a single Fourier component,

$$\frac{d^2 f(x)}{dx^2} = \lambda f(x) \tag{A.1}$$

with eigenfunction and eigenvalue

$$f(x) = \sin m\pi x, \qquad \lambda = -m^2\pi^2 \tag{A.2}$$

For second-order central differencing, you get

$$\begin{aligned}
\left.\frac{d^2 f(x)}{dx^2}\right|_{x_i} &= \frac{\sin m\pi(x_i + \Delta x) - 2\sin m\pi x_i + \sin m\pi(x_i - \Delta x)}{(\Delta x)^2} \\
&= \frac{1}{(\Delta x)^2}[\sin m\pi x_i \cos m\pi\Delta x + \cos m\pi x_i \sin m\pi\Delta x - 2\sin m\pi x_i \\
&\qquad + \sin m\pi x_i \cos m\pi\Delta x - \cos m\pi x_i \sin m\pi\Delta x] \\
&= \frac{2(\cos m\pi\Delta x - 1)}{(\Delta x)^2}\sin m\pi x_i
\end{aligned} \tag{A.3}$$

The finite difference approximation gives the correct eigenfunction in this simple example, but the eigenvalue (the coefficient of $\sin m\pi x_i$ in the last line) is

$$\begin{aligned}
\lambda_{\mathrm{FD}} &= \frac{2(\cos m\pi\Delta x - 1)}{(\Delta x)^2} \\
&= \frac{2}{(\Delta x)^2}\left[-\frac{1}{2}(m\pi\Delta x)^2 + \frac{1}{24}(m\pi\Delta x)^4 - \cdots\right]
\end{aligned} \tag{A.4}$$

The fractional error in the eigenvalue is

$$\left|\frac{\lambda_{\mathrm{FD}} - \lambda_{\mathrm{exact}}}{\lambda_{\mathrm{exact}}}\right| = \frac{\frac{1}{12}m^4\pi^4(\Delta x)^2}{m^2\pi^2} = \frac{1}{12}m^2\pi^2(\Delta x)^2 \tag{A.5}$$

So for 1% fractional error we require $m^2\pi^2(\Delta x)^2 \lesssim 0.12$, or $m\Delta x \lesssim 0.1$. Now the eigenfunction $\sin m\pi x$ has $m/2$ wavelengths in $[0, 1]$, and there are $1/\Delta x$ grid points in $[0, 1]$. Therefore

$$\begin{aligned}\text{No. of grid points per wavelength} &= \frac{1/\Delta x}{m/2} \\ &\gtrsim \frac{2}{0.1} \\ &= 20\end{aligned} \tag{A.6}$$

Thus the rule of thumb: second order accurate finite differencing needs about 20 points per wavelength for 1% accuracy.

With the Fourier basis, you only need one basis function per wavelength to get the exact eigenvalue. This is not quite a fair comparison, since we used the exact eigenfunction as the basis function. A fairer comparison is to use say Chebyshev polynomials as the basis functions. The expansion of $\sin m\pi x$ in Chebyshev polynomials is

$$\sin m\pi x = 2 \sum_{n \text{ odd}} (-1)^{(n-1)/2} J_n(m\pi) T_n(x) \tag{A.7}$$

For $n > m\pi$, $J_n(m\pi) \to 0$ exponentially as $n \to \infty$. So you need $n \sim m\pi$ to represent the sine wave accurately. But on $[-1, 1]$ (the domain of $T_n(x)$), $\sin m\pi x$ has m wavelengths. So you need about π T_n's per wavelength.

There's nothing special about Chebyshev polynomials. Using Legendre polynomials gives

$$\sin m\pi x = \frac{1}{(2m)^{1/2}} \sum_{n \text{ odd}} (-1)^{(n-1)/2} (2n+1) J_{n+1/2}(m\pi) P_n(x) \tag{A.8}$$

so again you need about π basis functions per wavelength.

B Eigenfunctions of Sturm-Liouville Problems as Basis Functions

B.1 Convergence of Fourier Series

Consider the finite Fourier series

$$f_N(x) = \sum_{k=-N}^{N} a_k e^{ikx} \tag{B.1}$$

where the coefficients are given by

$$a_k = \frac{1}{2\pi} \int_0^{2\pi} e^{-ikx} f(x)\, dx \tag{B.2}$$

Integrating eqn. (B.2) by parts gives

$$a_k = -\left.\frac{e^{-ikx} f(x)}{2\pi i k}\right|_0^{2\pi} + \frac{1}{2\pi i k} \int_0^{2\pi} e^{-ikx} f'(x)\, dx \tag{B.3}$$

If $f(x)$ is continuous and periodic, the first term gives zero. You can then integrate the term with $f'(x)$ by parts, getting another power of k in the denominator. Continuing in this way, you see that $a_k \sim 1/k^m$ if the first $m-1$ derivatives of f are continuous and periodic.

Suppose first that $m = 1$, i.e., f itself is discontinuous at some point. The error in the representation (B.1) can be written

$$f(x) - f_N(x) = \sum_{|k|=N+1}^{\infty} a_k e^{ikx} \tag{B.4}$$

Far from the discontinuity, the terms in the series have random phases, so the sum is approximately equal to the leading term, a_{N+1}, which decreases like $1/N$ for large N. Within a distance $1/N$ of the discontinuity, the terms in the series are in phase, and the error in eqn. (B.4) is of order unity. This is the famous Gibbs phenomenon. One can show that the fractional error at the discontinuity is an overshoot of 9% as $N \to \infty$.

More generally, if the lowest order derivative of f that has a discontinuity or non-periodicity is $f^{(m-1)}$, then the error is $O(1/N^m)$ far from the discontinuity and $O(1/N^{m-1})$ near the discontinuity. If f is infinitely differentiable and periodic, then the approximation (B.1) converges to f faster than any power of $1/N$ as $N \to \infty$. We say that the convergence is *exponential*.

B.2 Why are Chebyshev Polynomials Magical?

The exponential convergence of Fourier series depends on two properties: *smoothness* of f and its derivatives, and the *values* of f and its derivatives at the boundaries. By using different basis functions, it is possible to get exponential convergence for smooth functions *independent* of the boundary values. The key is to use eigenfunctions of a singular Sturm-Liouville problem.

Recall that the standard Sturm-Liouville eigenfunction equation is

$$\frac{d}{dx}\left[p(x)\frac{d\phi_n(x)}{dx}\right] + [\lambda_n w(x) - q(x)]\phi_n(x) = 0, \qquad x \in [a, b], \qquad p, q, w \geq 0 \tag{B.5}$$

Here λ_n are the eigenvalues, determined by imposing boundary conditions at $x = a$ and $x = b$. If $p(a) = p(b) = 0$, we say the Sturm-Liouville operator is *singular*.

The eigenfunctions of eqn. (B.5) form a complete orthonormal set with weight function $w(x)$:

$$\int_a^b \phi_n(x)\phi_m(x)w(x)\,dx = \delta_{mn} \tag{B.6}$$

So we can use them to represent the solution of our problem,

$$f_N(x) = \sum_{n=0}^{N} a_n \phi_n(x) \tag{B.7}$$

where

$$a_n = \int_a^b f(x)\phi_n(x)w(x)\,dx \tag{B.8}$$

We are interested in the convergence of eqn. (B.7) as $N \to \infty$, i.e., in the convergence of the a_n. We can derive this by an argument analogous to the integration-by-parts argument for Fourier coefficients. Substitute for $\phi_n(x)w(x)$ in eqn. (B.8) from eqn. (B.5):

$$a_n = -\frac{1}{\lambda_n}\int_a^b f(x)\frac{d}{dx}\left[p(x)\frac{d\phi_n(x)}{dx}\right]dx + \frac{1}{\lambda_n}\int_a^b f(x)q(x)\phi_n(x)\,dx \tag{B.9}$$

Integrating the first term by parts gives

$$\frac{1}{\lambda_n} f(x)p(x)\left.\frac{d\phi_n(x)}{dx}\right|_a^b + \frac{1}{\lambda_n}\int_a^b \frac{df}{dx}p(x)\frac{d\phi_n}{dx}dx \tag{B.10}$$

Integrating the second term in eqn. (B.10) by parts again gives a contribution

$$\frac{1}{\lambda_n}\frac{df}{dx}p(x)\phi_n(x)\bigg|_a^b - \frac{1}{\lambda_n}\int_a^b \frac{d}{dx}\left[p(x)\frac{df}{dx}\right]\phi_n\,dx \tag{B.11}$$

Putting all the pieces together, we find

$$a_n = \frac{1}{\lambda_n}p(x)\left[\frac{df}{dx}\phi_n(x) - f\frac{d\phi_n(x)}{dx}\right]_a^b - \frac{1}{\lambda_n}\int_a^b h(x)\phi_n(x)w(x)\,dx \tag{B.12}$$

where we've denoted by $h(x)$ the terms involving f and its derivatives that make up the new integrand.

Now here's the key point: If the Sturm-Liouville problem is singular, then $p(a) = p(b) = 0$ and so the boundary term in eqn. (B.12) is zero, *independent* of the values of f and df/dx at the boundaries, so long as they are finite. Moreover, the integral in eqn. (B.12) is of the same form as the integral we started with in eqn. (B.8), with h replacing f. So we can continue to repeat the integration by parts, each time gaining a factor of $1/\lambda_n$. The procedure terminates only if we encounter an unbounded derivative of f. Otherwise the convergence will be exponential in $1/\lambda_n$. Now recall that $\lambda_n \sim n^2$ as $n \to \infty$. (You can prove this e.g. from the WKB approximation.) So the convergence is exponential [2] in n.

C Program to Calculate Differentiation Matrices

Fornberg (1998) gives a Fortran program to calculate any order differentiation matrix given only the collocation points. Here we give a C++ translation, using the *Numerical Recipes* vector and matrix classes. Note that the elements of the zeroth derivative matrix are returned in `c[0..n][0]`. These are just the values of the cardinal functions, i.e., the weights for interpolation.

```
#include "nr.h"

void weights(DP z, Vec_I_DP &x, Mat_O_DP & c)
// input:
// z               location where approximations are to be accurate
// x[0..n]         grid point locations
// n               total no. of grid points = n+1
// m               highest derivative for which weights are sought
// output:
// c[0..n][0..m]   weights at grid locations x[0..n] for derivatives
//                 of order 0..m, found in c[0..n][0..m]
//                 The element c[j][k] contains the weight to be applied
//                 at x[j] when the kth derivative is approximated by
//                 a stencil over x[0..n]
{
    int i,j,k,mn;
    DP c1,c2,c3,c4,c5;
    int n=c.nrows()-1;
```

[2]The correct fall-off of a_n with n depends on the boundary conditions for f, since the ϕ_n' term in eqn. (B.12) contributes a factor of $\sqrt{\lambda_n}$. See Gottlieb & Orszag (1977) pp. 29–33 for a careful treatment.

```
    int m=c.ncols()-1;
    c1=1.0;
    c4=x[0]-z;
    for (k=0;k<=m;k++)
        for (j=0;j<=n;j++)
            c[j][k]=0.0;
    c[0][0]=1.0;
    for (i=1;i<=n;i++) {
        mn=MIN(i,m);
        c2=1.0;
        c5=c4;
        c4=x[i]-z;
        for (j=0;j<i;j++) {
            c3=x[i]-x[j];
            c2=c2*c3;
            if (j == i-1) {
                for (k=mn;k>0;k--)
                    c[i][k]=c1*(k*c[i-1][k-1]-c5*c[i-1][k])/c2;
                c[i][0]=-c1*c5*c[i-1][0]/c2;
            }
            for (k=mn;k>0;k--)
                c[j][k]=(c4*c[j][k]-k*c[j][k-1])/c3;
            c[j][0]=c4*c[j][0]/c3;
        }
        c1=c2;
    }
}
```

References

M. Bjørhus (1995), "The ODE Formulation of Hyperbolic PDEs Discretized by the Spectral Collocation Method," *SIAM J. Sci. Comput.* **16**, 542.

J. P. Boyd (2001), *Chebyshev and Fourier Spectral Methods*, 2nd ed., Dover Publications, Mineola, New York. Available online at `http://www-personal.engin.umich.edu/~jpboyd`.

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

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

B. Fornberg (1998), "Calculation of Weights in Finite Difference Formulas," *Siam Rev.* **40**, 685.

D. Gottlieb and S. A. Orszag (1977), *Numerical Analysis of Spectral Methods: Theory and Applications*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia.