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.

Spectral Methods

- Method of choice for high spatial resolution in multidimensions.
- 3-d finite difference code:
  resolution $\times$ 2 $\implies$ # of grid points $\times$ 8, error $\times$ 1/4.
- Spectral code: error $\times$ 10^{-8}.

- Good for smooth solutions.
- Discontinuities like shocks are bad.
- Even mild non-smoothness (e.g. discontinuity in high-order derivative of solution) spoils convergence of spectral methods.

Spectral vs. Finite Difference

Finite difference methods: approximate the *equation*.

Spectral methods: approximate the *solution*.

Finite difference: replace continuum equation by equation on grid points.

Spectral method: solution = truncated expansion in basis functions:

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

(Basis functions) + (Methods of computing a_n) $\longrightarrow$

(Flavors of spectral methods)

Example

One-sided wave equation (advective equation) in 1-d:

$$\frac{\partial u}{\partial t} = \frac{\partial u}{\partial x}, \qquad \text{periodic on } [0, 2\pi], \qquad u(t=0,x) \; = \; f(x)$$

Analytic spectral solution: expand u in Fourier series

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

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

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

Get $a_n(0)$ from the initial condition $u(t = 0, x) = f(x)$: expand

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

$$a_n(0) = f_n$$

For example,

$$u(t=0,x) \;=\; \sin(\pi\cos x)$$

$$u(t,x) = \sin[\pi\cos(x+t)]$$

Spectral coefficients:

$$a_n(0) = \frac{1}{2\pi}\int_0^{2\pi} \sin(\pi\cos x)e^{-inx}dx$$

$$= (-1)^{(n-1)/2}J_n(\pi), \qquad n \text{ odd}$$

Properties of Basis Functions for Analytic Spectral Method

1. *Complete* set of basis functions.

2. Each basis function by itself obeys the boundary conditions.

3. Eigenfunctions of the operator in the problem, d/dx.

(Separation of variables)

Only property (1) essential for numerical spectral methods.

Convergence of Spectral Solution

r.m.s. error: $$L_2 = \left[\frac{1}{2\pi}\int_0^{2\pi} |u(t,x) - u_N(t,x)|^2\, dx\right]^{1/2}$$

$$= \left[\frac{1}{2\pi}\int_0^{2\pi} \left|\sum_{|n|>N} a_n(0) e^{inx} e^{int}\right|^2 dx\right]^{1/2}$$

$$= \left[\sum_{|n|>N} |a_n(0)|^2\right]^{1/2}$$

$J_n(\pi) \to 0$ exponentially as $n \to \infty$.
So error decreases *exponentially* with N for any $t \geq 0$.

Key feature of a good spectral method. (2nd order FD error $\sim 1/N^2$.)

Resolution of Spectral Methods

Exponential convergence sets in when solution resolved.
Need $\sim \pi$ basis functions per wavelength.
(In example, need $n \gtrsim \pi$ for $J_n(\pi)$ small.)

2nd order FD needs ~ 20 points per wavelength for 1% accuracy.
(And accuracy improves much more slowly than with spectral methods.)

Choice of Basis Functions

Can't always use Fourier series as basis functions — it depends on the boundary conditions. Recipe for 99% of cases:

- Solution periodic: use Fourier series.
- Solution not periodic, domain a square or a cube: use Chebyshev polynomials along each dimension.
- Domain spherical: use spherical harmonics.

Reason: *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 (sine, cosine, exponential): periodic boundary conditions.

- Non-periodic solutions: orthogonal polynomials (Legendre, Chebyshev, . . .) Eigenfunctions of singular Sturm-Liouville problems.

- Spherical domains: $Y_{lm}(\theta, \phi) \propto P_l^m(\cos\theta)e^{im\phi}$

Why are Chebyshev Polynomials Popular?

- Eigenfunctions of singular Sturm-Liouville equation.

- Mapped trig. functions:

$$T_n(x) = \cos(n\theta), \qquad x = \cos\theta$$

  So expansion in Chebyshev polynomials $\Longleftrightarrow$ FFT.

- Derivatives of expansion $\Longleftrightarrow$ FFTs.

Computing the Expansion Coefficients a_n

1. Tau method.
2. Galerkin method. (cf. separation of variables, QM)
3. Collocation or Pseudospectral (PS) Method.

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

The Pseudospectral Method

- Reason: Easy to implement, especially for nonlinear problems.

- Instead of a_n, work with $y(x_j)$.
  $\{x_j\}$ = *Collocation points*.
  Gaussian quadrature points associated with basis functions.

- physical space $\Longleftrightarrow$ spectral space.

PS is an Interpolating Method

$$y_N(x) = \sum_{n=0}^{N} a_n \phi_n(x)$$

- Polynomial that interpolates the solution.
- Require $y_N(x) = y(x)$ at the $N + 1$ collocation points.
- As $N \to \infty$, errors in between x_j tend to zero exponentially fast (if we do things right).

Spectral Methods and Gaussian Quadrature

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

$w(x)$ = *weight function*

Factors out singular behavior, so $y(x)$ smooth.

w_i = *weights* (confusing!)

x_i = *abscissas*

Derivation: choose the $2N+2$ w_i and x_i, so that formula is exact for polynomials $1, x, x^2, \ldots, x^{2N+1}$.

Textbooks: Gaussian quadrature related to the orthogonal polynomials w.r.t. $w(x)$

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

x_i = $N + 1$ roots of $\phi_{N+1}(x)$.
w_i = formula in textbooks.

Discrete inner product of two functions:

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

Subscript G $\rightarrow$ Gaussian.

Discrete orthogonality relation

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

Proof: Evaluate $\int_a^b \phi_n(x)\phi_m(x)w(x)\,dx = \delta_{mn}$ by Gaussian quadrature.
Integrand = polynomial of degree $m + n \leq 2N + 1$.
But Gaussian quadrature integrates polynomials of degree $\leq 2N + 1$ exactly. QED.

Formula for PS Coefficients by Gaussian Quadrature

Approximate $y(x)$ by PS interpolating polynomial,
Collocation points = Gaussian quadrature points:

$$P_N(x) = \sum_{n=0}^{N} \bar{a}_n \phi_n(x), \qquad P_N(x_i) = y(x_i), \qquad i = 0, 1, \ldots, N$$

Theorem: $\bar{a}_n = \langle y|\phi_n\rangle_{\mathrm{G}}$ *exactly*.
Proof:

$$\begin{array}{ccc} \langle P_N|\phi_m\rangle_{\mathrm{G}} & = \displaystyle\sum_{n=0}^{N} \bar{a}_n & \langle\phi_n|\phi_m\rangle_{\mathrm{G}} \\ \Big\downarrow{\scriptstyle P_N(x_i)=y(x_i)} & & \Big\downarrow \\ y & & \delta_{mn} \end{array}$$

Relation Between Spectral and Pseudospectral Expansions

$$y(x) = \sum_{n=0}^{\infty} a_n \phi_n(x), \qquad a_n = \langle y | \phi_n \rangle \; = \int_a^b y(x)\phi_n(x) w(x)\, dx$$

$$P_N(x) = \sum_{n=0}^{N} \bar{a}_n \phi_n(x), \qquad \bar{a}_n = \langle y | \phi_n \rangle_{\mathrm{G}} = \sum_{i=0}^{N} w_i y(x_i) \phi_n(x_i)$$

(Lanczos 1938)

Relation:

$$
\begin{aligned}
\bar{a}_n &= \langle y|\phi_n\rangle_{\mathrm{G}} \\
&= \sum_{m=0}^{\infty} a_m \langle \phi_m|\phi_n\rangle_{\mathrm{G}} \qquad \text{since} \quad y = \sum_{m=0}^{\infty} a_m\phi_m \\
&= \sum_{m=0}^{N} a_m \langle \phi_m|\phi_n\rangle_{\mathrm{G}} + \sum_{m>N} a_m \langle \phi_m|\phi_n\rangle_{\mathrm{G}} \\
&= a_n + \sum_{m>N} a_m \langle \phi_m|\phi_n\rangle_{\mathrm{G}}
\end{aligned}
$$

Thus $\bar{a}_n$ exponentially close to a_n if N large enough.
Reason for name: PS coefficients are not the actual spectral coefficients, but very close to them. Don’t distinguish.

Gauss-Lobatto Quadrature

Gaussian quadrature collocation points = roots of $\phi_{N+1}(x)$.
All lie *inside* (a, b).

Another version of Gaussian quadrature that includes the two endpoints of the interval: Gauss-Lobatto quadrature.

Gauss-Lobatto quadrature points are as effective as ordinary Gaussian points.

Advantage: can impose boundary conditions at the endpoints.

Digression: Gaussian Quadrature is Itself a Spectral Method

Integration with equally spaced points:

$$N + 1 \text{ weights} \implies \text{degree of exactness} = N.$$

Gaussian integration: degree of exactness = $2N + 1$.

But main advantage: converges exponentially with N for smooth functions:

$$\bar{a}_0 = \langle y|\phi_0\rangle_{\mathrm{G}} = \phi_0 \sum_{i=0}^{N} w_i y(x_i)$$

converges exponentially to

$$a_0 = \langle y|\phi_0\rangle = \phi_0 \int_a^b y(x)w(x)\,dx$$

Fourier Series and Gaussian Quadrature

Fourier collocation points equally spaced. E.g.,

$$x_j = 2\pi j/N, \qquad j = 0, 1, \ldots, N-1$$

Fourier series: interpolates $y(x)$ by a *trigonometric* polynomial.

Gaussian quadrature: midpoint rule.
Gauss-Lobatto quadrature: trapezoidal rule.

Textbooks: low-order methods.
True for arbitrary functions.
But for *periodic* functions, exponentially convergent.

Cardinal Functions

Polynomial interpolation for *any* function $f(x)$:

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

$C_i(x)$ = *cardinal functions*.

Polynomial of degree N, 1 at ith collocation point, zero at others:

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

One explicit representation (Lagrange interpolation formula):

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

Choice of basis functions $\iff$ choice of collocation points x_j

$\iff$ choice of cardinal functions

Alternative Expression for $C_i(x)$

$\phi_n(x)$ = set of orthogonal polynomials
Collocation points = 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. Near $x = x_i$:

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

so divide out the zero at $x = x_i$

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

PS Interpolation vs. the Runge Phenomenon

Runge phenomenon: If grid points *equally spaced*,
error in $P_N(x)$ can $\rightarrow \infty$ as $N \rightarrow \infty$.

But error shows up near endpoints.

Fix: make points more concentrated toward endpoints
(e.g., Gaussian points).

Practical Formulas

Textbooks: formulas for $C_i(x)$ for standard basis functions.

In practice, will see we need derivatives of $C_i(x)$,
the *differentiation matrices*.

Spectral vs. Grid Point Representation

$$\mathcal{L}y = f \qquad (\mathcal{L} \text{ linear for simplicity})$$

Spectral Space

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

$$\sum_{n=0}^{N} a_n \mathcal{L}\phi_n(x) = f(x)$$

Physical Space

$$y(x) = \sum_{j=0}^{N} y_j C_j(x)$$

$$\sum_{j=0}^{N} y_j \mathcal{L}C_j(x) = f(x)$$

Impose at collocation points only:

$$\sum_{n=0}^{N} a_n \mathcal{L}\phi_n(x_j) = f(x_j)$$

i.e., $La = f$, where $L_{jn} = \mathcal{L}\phi_n(x_j)$

$$\sum_{j=0}^{N} y_j \mathcal{L}C_j(x_i) = f(x_i)$$

i.e., $L^{(c)}y = f$, where $L^{(c)}_{ij} = \mathcal{L}C_j(x_i)$

Relation Between Representations

Grid point values $\rightarrow$ spectral coefficients:

$$a_i = \langle \phi_i \,|\, y \rangle = \sum_j w_j \phi_i(x_j) y_j \qquad \text{(spectral} \iff \text{PS)}$$

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

Spectral space $La = f \rightarrow LMy = f$. So physical space $L^{(c)} y = f \rightarrow$

$$L^{(c)} = LM, \qquad L = L^{(c)} M^{-1}$$

$$\text{Also,} \qquad a = My \implies y = M^{-1} a$$

Since $y = \sum a_n \phi_n$, M^{-1} = matrix that sums spectral series $\rightarrow y_i$:

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

Check:

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

Using the FFT

Large N, Fourier or Chebyshev basis:
Use FFT for transformations $a = My$ and $y = M^{-1}a$

Simple programs: just do matrix multiplication.

Differentiation Matrices

Key ingredient in PS method:

$$L_{ij}^{(c)} = \mathcal{L} C_j(x_i)$$

So must take derivatives of $C_j(x)$ at the $\{x_i\}$:

$$D_{ij}^{(1)} = \partial_x C_j(x_i), \qquad D_{ij}^{(2)} = \partial_x^2 C_j(x_i), \quad \ldots$$

- Compute ahead of time and store.
- $$\frac{\partial y}{\partial x} \quad \longleftrightarrow \quad \sum_{j=0}^{N} D_{ij}^{(1)} y_j \qquad \text{(matrix multiplication)}$$

Using the FFT for Differentiation

Matrix multiplication of a vector is $O(N^2)$.

Fourier basis functions e^{ikx}, alternative:

$$\begin{aligned} y &\overset{\text{FFT}}{\longrightarrow} a \\ a &\longrightarrow ika \qquad \text{(A)} \\ ika &\overset{\text{inverse FFT}}{\longrightarrow} y' \end{aligned}$$

Chebyshev basis functions: $O(N)$ recurrence in step (A).

Procedure is $O(N \log N)$.
Typically faster than matrix multiplication only for $N \gtrsim 16$ – 128.
So just use matrix multiplication for simple programs.

Options for Computing Differentiation Matrices

1.
$$C_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{N} \frac{x - x_j}{x_i - x_j}$$

2.
$$C_i(x) = \frac{\phi_{N+1}(x)}{(x - x_i)\phi'_{N+1}(x_i)}$$

3. Look up the explicit formulas in books.

4. Use the program given by Fornberg (1998). Algorithm computes any order of differentiation matrix given only $\{x_i\}$.

Obviously, the last choice is the easiest.

Interpolation by Matrix Multiplication

To evaluate solution at points ≠ collocation points: interpolation.
Method 1:

$$a_i = \sum_j w_j \phi_i(x_j) y_j, \qquad y(x) = \sum_n a_n \phi_n(x) \quad \text{(Clenshaw)}$$

Method 2:

$$y(x_k) = \sum_j y_j C_j(x_k) \qquad \text{matrix multiplication)}$$

Fornberg's program → $C_j(x_k)$ for any set $\{x_k\}$.
(Differentiation matrix of order 0 = interpolation matrix.)

PS Derivatives vs. FD Derivatives

At center of equally spaced grid:

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

Centered differences: $\lim_{N\to\infty}$(weights) = finite.
One-sided approximations (or partially one-sided): weights diverge.
So high order FD approximations $\to$ large errors near boundaries.

Grid points closer together near end points (Gaussian points):
FD approximation convergent as $N \to \infty$.

PS method: exact derivative of $P_N(x)$ passing through data
at the $N+1$ grid points.
FD method using same points $\rightarrow$ same result ($P_N(x)$ unique).

PS method:

- Way to find high-order numerical approximations to derivatives at grid points.
- Satisfy the equation at the grid points (like FD).
- Variable coefficients or nonlinearities: multiply the functions at the grid points. (Big advantage over tau and Galerkin methods.)

Example (from Appendix B of Fornberg 1996)

$$y'' + y' - 2y + 2 = 0, \qquad -1 \le x \le 1,$$

$$y(-1) = y(1) = 0$$

Exact solution:

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

Use Chebyshev polynomials with $N = 4$:

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

Gauss-Lobatto collocation points (endpoints for b.c.'s):

$$x_i = \cos \frac{i\pi}{4}, \qquad i = 0, \ldots, 4$$

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

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

Require differential equation to hold at interior points x_k, $k = 1, 2, 3$.
Uses middle 3 rows of these matrices.
B.c.'s $y_0 = y_4 = 0 \implies$ omit first and last columns.

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

Solution:

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

Exact: $y(x = 0) = 0.52065$, compared with $y_2 = 0.52000$.
Error is about 10^{-16} for $N = 16$.
Second-order FD: error $\sim 1/10$ smaller for $N = 16$.

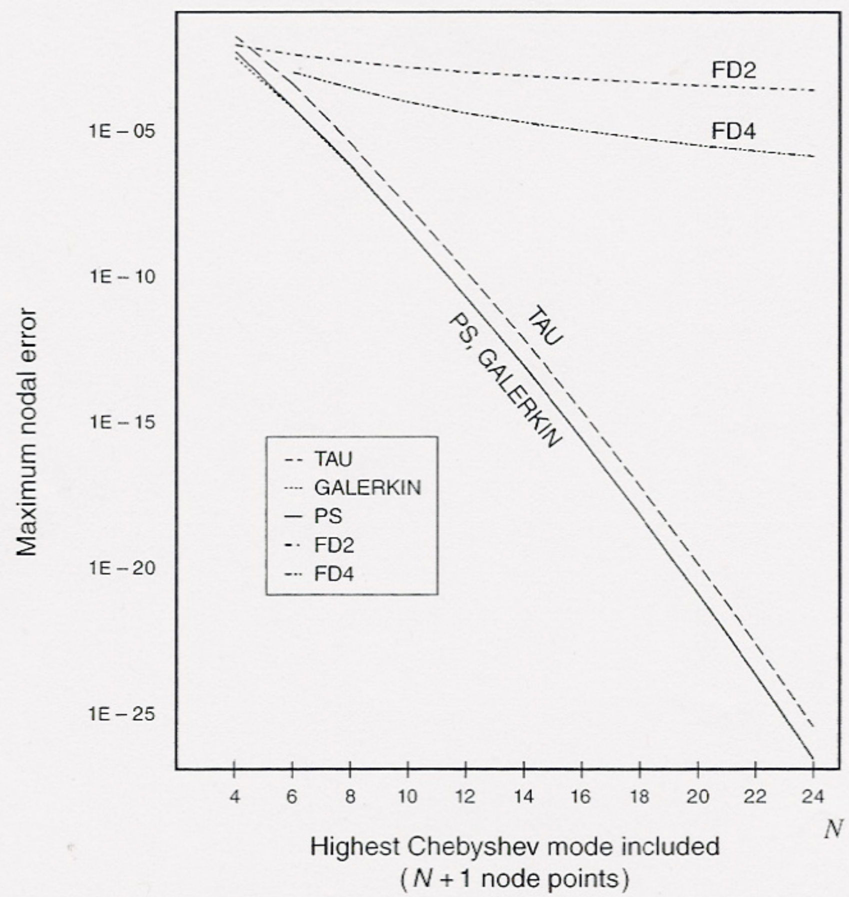

Figure B-1. Maximum nodal errors for different methods when applied to the model in Appendix B: comparison among three spectral implementations and equi-spaced FD methods of second and fourth order.

Exercise

Solve same problem as above, but with b.c.

$$y'(1) = 0$$

instead of $y(1) = 0$.

One way: set first row of $D^{(1)}$ matrix to zero.
Then have to include point $x = 1$ with interior collocation points.

Exact solution for checking:

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

The Method of Lines

Spectral in space, ODE in time, e.g.:

$$\frac{\partial y}{\partial t} = \frac{\partial y}{\partial x}$$

$$y(t, x) = \sum_j C_j(x) y_j(t)$$

$$\left.\frac{\partial y}{\partial t}\right|_i = \dot{y}_i, \qquad \left.\frac{\partial y}{\partial x}\right|_i = \sum_j D^{(1)}_{ij} y_j$$

Now use e.g. Runge-Kutta.