Numerics using Chebyshev polynomials

Chebyshev differentiation is another way to solve differential equations.

We interpolate $u(x)$ on $[-1,1]$ using a polynomial of degree $N+1$ at a specific set of non-evenly spaced grid points which have high density close to the end points:

$$x_k = \cos(k\pi/N) \qquad k=0,\dots,N$$ Chebyshev Points

Why? For any interpolating polynomial $P(x)$

$$f(x) - P(x) = \frac{\overbrace{(x-x_1)\cdots(x-x_n)}^{q(x)}}{n!} f^{(n)}(c)$$

and these Chebyshev points minimize $|q(x)|$.

Also: Chebyshev interpolation is agnostic about boundary values.

Let $u_k \approx u(x_k)$ and consider the vector $(u_0, u_1, \ldots, u_N) \in \mathbb{R}^{N+1}$.

Then $D_N = (d_{k\ell}) \in \mathbb{R}^{(N+1)(N+1)}$

the Chebyshev differentiation matrix.

What does it do? It gives the derivative of the appropriate interpolant of the u vector.

Thus

$$D_N\, u \approx \begin{pmatrix} u'(x_0) \\ u'(x_1) \\ \vdots \\ u'(x_N) \end{pmatrix}$$

with extreme (spectral) accuracy as long as $u(x)$ is sufficiently smooth.

Example Solve

"DId"

$$\boxed{\begin{array}{ll} u''(x) = f(x) & \text{on } (-1,1) \\ u(-1) = u(1) = 0 & \text{boundary} \end{array}}$$

Steps

① $u = (u_0, \dots, u_N)$ unknown

$f = (f(x_0), \dots, f(x_N))$ calculated

Then $D_N^2 u = f$

is the correct equation for $u_1, \dots, u_{N-1}$

But what about the boundary?

We want $u_0 = u_N = 0$.

How to incorporate this into the system? Modify D_N^2 in first & last row!

$$A = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ \hline d_{1,0} & d_{11} & \cdots & d_{1N} \\ \vdots & & & \\ d_{N-1\,0} & d_{N-1\,1} & \cdots & d_{N-1\,N} \\ \hline 0 & \cdots & 0 & 1 \end{pmatrix} \qquad g = \begin{pmatrix} 0 \\ \hline f(x_1) \\ \vdots \\ f(x_{N-1}) \\ \hline 0 \end{pmatrix}$$

$u = A \backslash g$

Linear $\Rightarrow$ eventual linear solve

"NId" **Example** $u'' + \lambda u = f \qquad u'(-1) = u'(1) = 0$

again f & u are vectors as above.

Inside the interval

$$\left(D_N^2 + \lambda I_{N+1}\right) u = f$$

gives our discretized equation

What about boundary conditions?

We know that

$$D_N u \simeq u' \text{ (the derivative approx.)}$$

$$\begin{pmatrix} \boxed{\text{TOP ROW}} \\ D_N \\ \boxed{\text{BOTTOM ROW}} \end{pmatrix} \begin{pmatrix} \\ u \\ \\ \end{pmatrix} = \begin{pmatrix} u_1' \\ \\ u_N' \end{pmatrix} \begin{matrix} \leftarrow \text{want } 0 \\ \\ \leftarrow \text{want } 0 \end{matrix}$$

Thus let $A = D_N^2 + \lambda I_{N+1}$

and now

$$\tilde{A} = \begin{pmatrix} \boxed{\text{TOP ROW}} \\ \hline — a_1 — \\ \vdots \\ — a_{N-1} — \\ \hline \boxed{\text{BOTTOM ROW}} \end{pmatrix} \Rightarrow \text{solve} \quad \tilde{A} u = \begin{pmatrix} 0 \\ \hline f_1 \\ \vdots \\ f_{N-1} \\ \hline 0 \end{pmatrix}$$

Example

What about Robin conditions?

Nonlinear $\Rightarrow$ Eventually Newton's Method

"ac robin1d" $\Delta u + \lambda(u - u^3) = 0$ on $\Omega = (-1,1)$

$$\alpha_0 \frac{\partial u}{\partial n}(-1) + \beta_0 u(-1) = \gamma_0$$

$$\alpha_1 \frac{\partial u}{\partial n}(+1) + \beta_1 u(+1) = \gamma_1$$

Idea:

$$f(u) = u - u^3 = \begin{pmatrix} u(x_0) - u(x_0)^3 \\ u(x_1) - u(x_1)^3 \\ \vdots \\ u(x_N) - u(x_N)^3 \end{pmatrix}$$

$$\Delta u \cong D_N^2 u$$

Let $A = D_N^2$

$$D_N = \begin{pmatrix} d_{00} & d_{01} & \cdots & d_{0N} \\ \hline & & & \\ \hline d_{N0} & d_{N1} & \cdots & d_{NN} \end{pmatrix}$$

$$\tilde{A} = \begin{pmatrix} \beta_0 + \alpha_1 d_{00} & \alpha_1 d_{01} & \cdots & \alpha_1 d_{0N} \\ \hline & a_1 & & \\ & \vdots & & \\ & a_{N-1} & & \\ \hline -\alpha_0 d_{N0} & -\alpha_0 d_{N1} & \cdots & \beta_0 - \alpha_0 d_{NN} \end{pmatrix} \qquad \tilde{f} = \begin{pmatrix} 0 \\ \hline f_1 \\ \vdots \\ f_{N-1} \\ \hline 0 \end{pmatrix}$$

Solve $\tilde{A} u = \tilde{f}$

Example

What about 2D problems

Mixed Partial

Linear $\Rightarrow$ eventually linear solve

$$u_{xx} + u_{yy} + u_{xy} = f \quad \text{on} \quad (-1,1)^2$$

with $u = 0$ on $\partial\Omega$

Let D_N be the derivative matrix

x = vector of Chebyshev points

Then $[xx, yy]$ = meshgrid (x, x)

is the grid for 2D Chebshev.

Thus $\overset{u_x}{\text{kron}}(D_N, I)$ & $\overset{u_y}{\text{kron}}(I, D_N)$ are the corresponding matrices for finding u_x & u_y.

$$u_{xx} + u_{yy} + u_{xy}$$

$$\approx \left(\text{kron}(D_N^2, I) + \text{kron}(I, D_N^2) + \text{kron}(I, D_N)\,\text{kron}(D_N, I)\right) u$$

Boundary conditions : $u_{k,0} = u_{kN} = u_{0\ell} = u_{N\ell} = 0$

How can we insert these :

b = find((abs(xx) == 1) | (abs(yy) == 1))

entries corresponding to boundary elements

$A(b;:)$ = rows of A corresponding to boundary elements.

We want that if u_{kl} is a boundary element, then $u_{kl} = 0$.

Thus replace row $A(b;:)$ with all zeros except a 1 along the diagonal

ie $\quad A(b;:) = \text{zeros}(4\cdot N, (N+1)^2) \leftarrow$ zero out

Now $\quad A(b,b) = \text{eye}(4N, 4N) \leftarrow$ add 1 on diagonal

$\quad f(b) = \text{zeros}.$

Now $\quad v = A \backslash b$

Example

The "even the kitchen sink" example

2D, Robin boundary conditions

"robin2d" $\Delta u = f$ $\quad (-1,1)^2$

$$u(-1,y) = y^2-1 \quad , \quad u(1,y) = 1-y^2$$

$$\alpha_0 \frac{\partial u}{\partial n}(x,-1) + \beta_0 u(x,-1) = \gamma_0 \quad , \quad \alpha_1 \frac{\partial u}{\partial n}(x,1) + \beta_1 u(x,1) = \gamma_1$$

Eg bxl = Find (xx == -1) etc

Clear out the corresponding rows and replace them with the appropriate boundary conds. Also replace the appropriate rows of f with

ff(bxl) = -(1 - yy(bxl).^2)

Note Conformal mappings can be used as domain transformations so as to solve PDEs on non square domains.