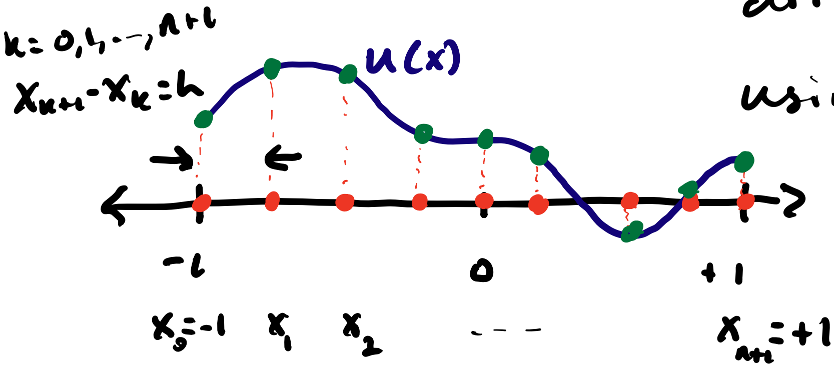
Numerics on 2D Separable Domains

Finite difference ⇒ local approximations of differential operators using difference quotients



$$Lu = -a(x)\frac{d^2u}{dx^2} + b(x)\frac{du}{dx} + c(x)u \qquad \text{s.t.} \quad u(\pm 1) = 0$$

$$\underbrace{\begin{bmatrix} a(x_1) & & \\ & \ddots & \\ & & a(x_n) \end{bmatrix}}_{M_a}\underbrace{\begin{bmatrix} -2 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{bmatrix}}_{D_2} + \underbrace{\begin{bmatrix} b(x_1) & & \\ & \ddots & \\ & & b(x_n) \end{bmatrix}}_{M_b}\underbrace{\begin{bmatrix} 0 & 1 & & \\ -1 & 0 & \ddots & \\ & \ddots & \ddots & 1 \\ & & -1 & 0 \end{bmatrix}}_{D_1} + \underbrace{\begin{bmatrix} c(x_1) & & \\ & \ddots & \\ & & c(x_n) \end{bmatrix}}_{M_c}$$

M_v = "multiplication by $v(x)$" = diagonal

D_k : k^{th} derivative = constant along diags + banded

⇒ Extend to 2D separable domains via Kronecker Product & Vectorization.

$$D_2 U + U D_2 = F \implies \left[(I \otimes D_2) + (D_2 \otimes I)\right]\mathrm{vec}(U) = \mathrm{vec}(F)$$

Q: "Discretize" $x^2+y^2 = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial xy} + \frac{\partial^2 u}{\partial y^2} + xy^2 u$, using D_2, M_x, M_{y^2} to obtain an $n \times n$ matrix equation for unknowns $(U)_{ij}$ on grid.

A: $$D_2 U + U D_2 + D_1 U D_1 + M_x U M_{y^2} = F$$

$(F)_{ij} = x_i^2 + y_j^2$

Q: Use the identity $\operatorname{vec}(AXB) = (B^T \otimes A)\operatorname{vec}(X)$ to obtain an $n^2 \times n^2$ linear system for the unknowns $(U)_{ij}$.

A: $$M \operatorname{vec}(U) = \operatorname{vec}(F)$$

where

$$M = \left[(I \otimes D_2) + (D_2 \otimes I) - (D_1 \otimes D_1) + (M_{y^2} \otimes M_x)\right]$$

Q: What if $xy^2 u$ is replaced by $v(x,y)u$?

A: $M_{y^2} \otimes M_x$ is replaced by $\operatorname{diag}(\operatorname{vec}(V))$

$(V)_{ij} = V(x_i, y_j)$

Spectral Methods

Idea: Choose basis $e_1, e_2, \ldots, e_n, \ldots$ for function space and write diff. op. L as a (usually infinite) matrix acting on coeffs of functions:

$$(A)_{ij} = \langle e_i, L e_j \rangle \qquad i,j = 1, 2, \ldots, n, \ldots$$

First entry of first "column"

Then truncate to first $n \times n$ section:

$$A_n = \begin{bmatrix} \langle e_1, L e_1 \rangle & \cdots & \langle e_1, L e_n \rangle \\ \vdots & & \\ \langle e_n, L e_1 \rangle & \cdots & \langle e_n, L e_n \rangle \end{bmatrix}$$

Our numerical (approximate) solution of $Lu = f$ is

$$\begin{bmatrix} \langle e_1, L e_1 \rangle & \cdots & \langle e_1, L e_n \rangle \\ \vdots & & \\ \langle e_n, L e_1 \rangle & \cdots & \langle e_n, L e_n \rangle \end{bmatrix} \begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \vdots \\ \hat{u}_n \end{bmatrix} = \begin{bmatrix} \hat{f}_1 \\ \hat{f}_2 \\ \vdots \\ \hat{f}_n \end{bmatrix}$$

where $\hat{f}_k = \langle e_k, f \rangle$, so that

$$u \approx \sum_{i=1}^{n} \hat{u}_i e_i$$

$\Rightarrow$ We have computed an approximation of u as a combo of basis functions $e_1, \ldots, e_n$.

⇒ Works well when "Truncation Error" is small, meaning that discarded coeffs of rhs, $\hat{f}_{n+1}, \hat{f}_{n+2}, \dots,$ and soln, $\hat{u}_{n+1}, \hat{u}_{n+2}, \dots,$ are small (small loss from truncating to $e_1, e_2, \dots, e_n$).

⇒ Coeffs $\hat{f}_1, \hat{f}_2, \dots,$ are typically approximated by a numerical quadrature rule

$$\hat{f}_j = \int \overline{e_j(x)} f(x)\,dx \approx \sum_{k=1}^{m} \underset{\uparrow \text{ Quad. weights}}{w_k}\, e_j(\underset{\downarrow \text{ Quad. nodes}}{x_k})$$

E.g. $u(x) = \frac{1}{\sqrt{2}} \sum_{j=-n}^{n} \hat{u}_k e^{i\pi k x}$ (Fourier Basis)

Solve $\frac{d^2 u}{dx^2} + \underbrace{(\cos x)}_{\frac{1}{2}(e^{i\pi x} + e^{-i\pi x}) \leftarrow \text{Euler's ID}} u = f(x)$ $\quad u(-1) = u(1)$, $u'(-1) = u'(1)$

$$\left(\begin{bmatrix} -(n\pi)^2 & & & & \\ & \ddots & & & \\ & & -\pi^2 & & \\ & & & 0 & \\ & & & & -\pi^2 \\ & & & & & \ddots \\ & & & & & & -(n\pi)^2 \end{bmatrix} + \begin{bmatrix} 0 & 1/2 & & & \\ 1/2 & 0 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 1/2 \\ & & & 1/2 & 0 \end{bmatrix} \right) \begin{bmatrix} \hat{u}_{-n} \\ \vdots \\ \hat{u}_0 \\ \vdots \\ \hat{u}_n \end{bmatrix} = \begin{bmatrix} \hat{f}_{-n} \\ \vdots \\ \hat{f}_0 \\ \vdots \\ \hat{f}_n \end{bmatrix}$$

$$(\underset{\uparrow \text{ diagonal}}{D_2} + \underset{\uparrow \text{ constant diags}}{M_{\cos \pi x}})\, \tilde{u} = \tilde{f}$$

Q: What needs to change if we use Boundary Conditions $u(\pm 1)=0$?

Extend to 2D on separable domains:

$$u(x,y) \approx \sum_{i=-n}^{n} \sum_{j=-n}^{n} \hat{u}_{ij}\, e_i(x) e_j(x)$$

$$f(x,y) \approx \sum_{i=-n}^{n} \sum_{j=-n}^{n} \hat{f}_{ij}\, e_i(x) e_j(x)$$

$$\Delta u = f \quad \text{w/ periodic B.C.'s}$$

$$D_2 \hat{U} + \hat{U} D_2 = \hat{F}$$

$(\hat{U})_{ij} = \hat{u}_{ij}$ unknowns (coeffs)

$(\hat{F})_{ij} = \hat{f}_{ij}$ (coeffs of f)

Solve $(2n+1)^2 \times (2n+1)^2$ linear system for coeffs $\Rightarrow$ $\left[(I \otimes D_2) + (D_2 \otimes I)\right] \mathrm{vec}(\hat{U}) = \mathrm{vec}(\hat{F})$

Fourier Coeffs

The computational efficiency depends on how quickly the Fourier coeffs of u and its derivatives decay. On periodic domains, Fourier Spectral Methods can be much more efficient/accurate than Finite Diff. Methods b/c coeffs often decay exponentially.

1) $u(x)$ real analytic on $[-1,1)$ periodic.

$\Rightarrow$ Fourier Coeffs decay exponentially
$(|\hat{u}_n| \leq C e^{-n})$

2) $u(x)$ k-times continuously diff. on " "

$\Rightarrow$ Fourier coeffs decay algebraically
$(|\hat{u}_n| \leq C n^{-k})$
decay rate derived in HW2

3) $u(x)$ not periodic

$\Rightarrow$ slow or no convergence!