Structure & Symmetry in BVPs

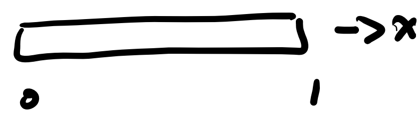
Boundary Value Problem (*)

$$-\partial_x^2 u = f$$

s.t.

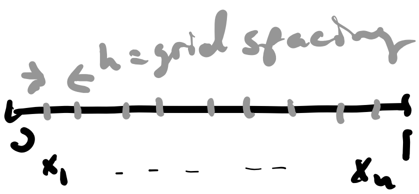
$$u(0) = (1) = 0$$

$u(x) = $ temp.



Step 1: Discretize $\quad \partial_x^2 u \simeq \frac{1}{h^2}\left(-u(x+h) + 2u(x) - u(x-h)\right)$

second-order difference



$$\underbrace{\frac{1}{h^2}\begin{bmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}}_{A_n} \underbrace{\begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix}}_{\underline{u}_n} = \underbrace{\begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix}}_{\underline{f}_n}$$

Step 2: Solve $A_n \underline{u}_n = \underline{f}_n$ in $O(n)$ FLOPs w/ banded solver

Step 3: Analyze error $\|\underline{u}_n - \underline{u}_n^*\| \sim h^2 = \left(\frac{1}{N+1}\right)^2$

Higher-order FDs

bandwidth (cost to factor A_n) $\Leftrightarrow$ accuracy (size n of disc.)

E.g. 4^{th} order stencil $\Rightarrow$ pentadiagonal & $O\left(\frac{1}{N+1}\right)^4$

Q: What structure/symmetry appears in (*) and how does it appear in A_n?

$\Rightarrow$ These structures/symmetries are what we exploit to design fast solvers.

E.g.	$-\partial_x^2 u = f$ (smooth)	A_n
(step 3)	smooth soln	order-m approx
(step 2)	locality	bandedness
?	*translation invariant	Toeplitz (const. diags)
(geometry)	symmetric about $x=\frac{1}{2}$	odd/even soln's decouple

*Careful about boundary conditions!

Fast Finite Differences (1D Warm Up)

Can we develop fast solvers for any order scheme?

$\Rightarrow$ Leverage other structure in (*)/A_n

Example 1: periodic boundary conditions

$$-\partial_x^2 u = f \qquad \text{s.t.} \quad u(0) = (1) \ \text{ and } \ u'(0) = u'(1)$$

$$\underbrace{\frac{1}{h^2}\begin{bmatrix} 2 & -1 & & & -1 \\ -1 & 2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ -1 & & & -1 & 2 \end{bmatrix}}_{A_n} \underbrace{\begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix}}_{\underline{u}_n} = \underbrace{\begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix}}_{\underline{f}_n}$$

A_n is called a "circulant" matrix

It's eigenvectors are <u>very</u> special!

<u>j^{th} eigenvector:</u>

$j = 0, \ldots, n-1$

$\omega = e^{2\pi i/n}$

$$v_j = \frac{1}{\sqrt{n}}\left(1, \omega^j, \omega^{2j}, \ldots, \omega^{(n-1)j}\right)$$

$$\begin{aligned} \lambda_j &= 2 - e^{-2\pi i j/n} - e^{-2\pi i j(n-1)/n} \\ &= 2 - e^{-2\pi i j/n} - e^{2\pi i j/n}\, \overbrace{\cancel{e^{-2\pi i j n/n}}}^{=1} \\ &= 2 - 2\cos\left(\frac{2\pi j}{n}\right) \end{aligned}$$

$$A_n = \underbrace{\begin{bmatrix} v_1 & \cdots & v_n \end{bmatrix}}_{V} \underbrace{\begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}}_{\Lambda} \underbrace{\begin{bmatrix} - & \bar{v}_1^T & - \\ & \vdots & \\ - & \bar{v}_n^T & - \end{bmatrix}}_{V^T}$$

$\Rightarrow$ The matrix V is called the

"Discrete Fourier Transform"

$\Rightarrow$ It has a powerful algorithm for mat-vecs in $\mathcal{O}(n\log n)$ flops

"Fast Fourier Transform"

New fast algorithm:

1) Compute $\underline{\hat{f}}_n = V^* \underline{f}_n$ — FFT $\mathcal{O}(n\log n)$

2) Solve $\Lambda \underline{\hat{u}}_n = \underline{\hat{f}}_n$ — diagonal $\mathcal{O}(n)$

3) Solve $V^* \underline{u}_n = \hat{u}_n$ — (Inverse) FFT $\mathcal{O}(n\log n)$

Fast transform to discrete Fourier basis leads to a fast $\mathcal{O}(n\log n)$ algorithm for any order FD scheme, b/c it exploits the Circulant (translation invariance) property intstead of the bandedness (locality) property.

More about FFT and FFT-based schemes in Lecture 3.