

FINITE DIFFERENCE METHODS FOR ELLIPTIC PDEs

A. DOBREV, COURANT

PDE Theory

Consider a linear boundary value problem (BVP)

$$\begin{cases} \mathcal{L}u(\vec{x}) = f(\vec{x}) & \text{in } \Omega \\ \mathcal{B}u(\vec{x} \in \partial\Omega) = g(\vec{x} \in \partial\Omega) \end{cases}$$

where \mathcal{L} is an elliptic operator
and Ω is a bounded domain

①

E.g.: Sturm - Liouville (SL) OVPs

In 1D:

$$S(\mathcal{L}u)(x) = -(p(x)u'(x))' + q(x)u(x)$$

$p(x) > 0, q(x) > 0 \text{ on } [a, b]$

with either periodic BCs or
(inhomogeneous) Robin BCs

$$\alpha u'(a) + \beta u(a) = g(t)$$

and same for $x = b$.

Can we write

$$u = \mathcal{L}^{-1} f$$

if BCs are homogeneous?

②

Inverse has to be given meaning through the eigen functions/values of the elliptic operator \mathcal{L} :

$$\left\{ \begin{array}{l} \mathcal{L} u_k = \lambda_k u_k \\ B u_k = 0 \end{array} \right.$$

eigen function
eigen value

Countably infinitely many eigen pairs

(λ_k, u_k) , $k = 0, 1, 2, \dots$
for bounded domains

③

If \mathcal{L} is Hermitian (self-adjoint)

$$\mathcal{L}^* = \mathcal{L}$$

in some inner product

$$(\mathcal{L}f, g)_w = (f, \mathcal{L}^*g)_w$$

on a Hilbert function space, Hei

1) All eigenvalues are real

2) There is a complete set of
orthonormal eigenfunctions in L_2

that is enumerable.

(4)

If \mathcal{L} is symmetric positive definite (elliptic) then

$$\lambda_k \geq 0 \quad \forall k$$

This is true, for example, of SL problems in 1D.

If $\forall \lambda_k \geq 0$, then \mathcal{L}^{-1} BVP exists and homogeneous solution:

$$u = \mathcal{L}^{-1} f$$

(5)

If $f = \sum_k b_k u_k$, where

$$b_k = (f, u_k)_W$$

then

$$\chi^{-1} f = \sum_k \frac{b_k}{\lambda_k} u_k$$

For inhomogeneous BCs
we need to find one particular
solution (best done using
boundary integral methods which
we will cover briefly later).

⑥

Another approach is to use the Green's function for the PDE with the specific homogeneous BSs:

$$\left\{ \begin{array}{l} \mathcal{L}G(\vec{x}; \vec{y}) = \delta(\vec{y} \in \Omega) \\ \mathcal{B}G(\vec{x} \in \partial\Omega; \vec{y}) = 0 \end{array} \right.$$

quadrature!

$$\Rightarrow u(\vec{x} \in \Omega) = \int_{y \in \Omega} f(\vec{y}) G(\vec{x}; \vec{y}) d\vec{y}$$

+ particular solution

(7)

Sadly, it is harder to compute eigenfunctions or Green's functions than to solve the BVP, except in special simple cases (e.g.) Poisson in a circle). Furthermore, the Green's function is generally singular, so the quadrature is very tricky (singular, hyper singular, or weakly singular)



⑧

Since we will use these later, though, let's just compute the eigenpairs and Green's function for the Laplace operator in 1D with homogeneous Dirichlet BCs :
 on $[0, L]$

Eigenpairs:

$$\begin{cases} u_k'' = \lambda_k u_k \\ u_k(0) = u_k(L) = 0 \end{cases} \Rightarrow u_k \sim \sin\left(\frac{2\pi k}{L}x\right)$$

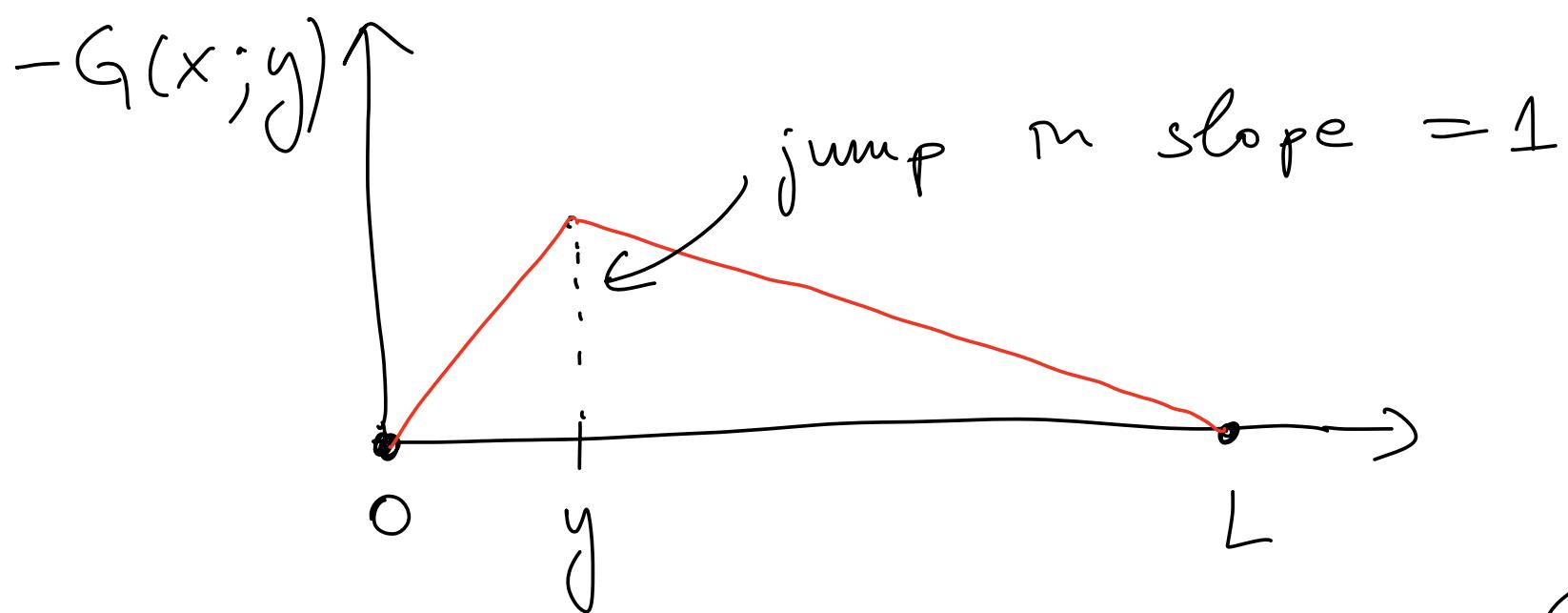
$$\lambda_k = \left(\frac{2\pi k}{L}\right)^2$$

③

Green's function :

$$G(x,y) = \frac{1}{L}.$$

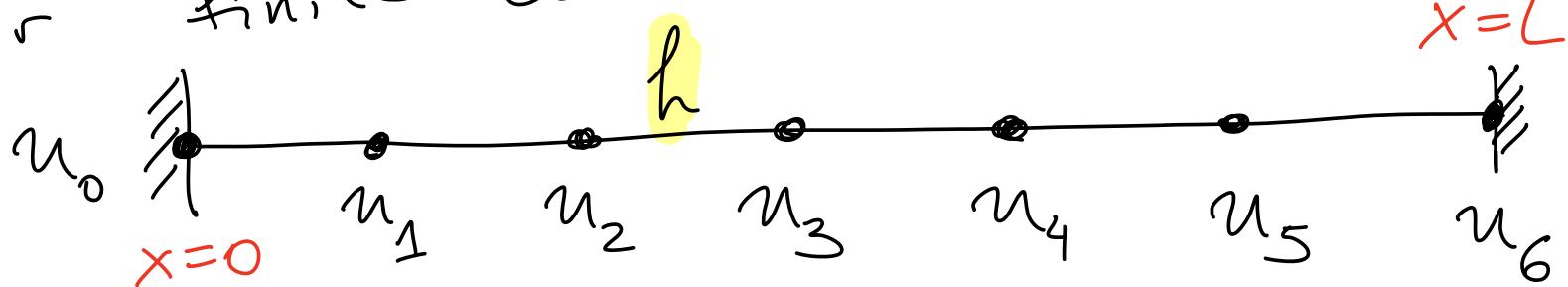
$$\begin{cases} G'' = \delta(y) & \Rightarrow \\ G'(y^+) - G'(y^-) = 1 & \begin{cases} (y-L)x, & x \leq y \\ (x-L)y, & x \geq y \end{cases} \\ G(0) = G(L) = 0 & \end{cases}$$



10

FINITE DIFFERENCE (FD)

In FD methods, we represent functions $f(x)$ with a vector of their pointwise values on a grid of different or finite element from either finite volume or finite difference methods!



11

For equispaced grids with grid spacing h :

$$u_k \approx u(x = kh)$$

We can approximate derivatives of $u(x)$ by using polynomial interpolation through several nearby grid points = stencil by a polynomial interpolant, and differentiating the interpolant.

(12)

This gives us finite difference approximations of derivatives of a certain degree of accuracy:

$$u'(x \in \text{grid}) \approx$$

$$\textcircled{1} \quad (D_+ u)(x) = \frac{u(x+h) - u(x)}{h} + O(h)$$

$$(D_- u)(x) = \frac{u(x) - u(x-h)}{h} + O(h)$$

one-sided differences

$$\textcircled{2} \quad \text{centered difference} \quad (D_0 u)(x) = \frac{u(x+h) - u(x-h)}{2h} + O(h^2)$$
(12)

Observe that D_0 does not store $u(x)$ itself, i.e., the matrix that represents D_0 has zeros on the diagonal. This will turn out to be a problem for hyperbolic PDEs later on...

$$(D_3 u)(x) = \frac{1}{6h} \left[2u(x+h) + 3u(x) - 6u(x-h) + u(\bar{x}-2h) \right] + O(h^3)$$

down-biased FD (good for hyperbolic)

(19)

The local truncation error (LTE) can be computed easily using Taylor series, e.g.)

$$(D_3 u)(x) = u'(x) + \frac{h^3}{12} u^{(4)}(x) + O(h^4)$$

We can represent these FDs by a stencil:

$$D_3 = \frac{1}{h} \left[\begin{array}{cccc} \bullet & \bullet & \circ & \bullet \\ \frac{1}{6} & -1 & \frac{1}{2} & \frac{1}{3} \end{array} \right] \quad (15)$$

Second-order derivative

Let's discretize now $\partial_{xx} u$ to second order.

Option 1 : Most commonly used is the 3 pt Laplacian :

$$D^2 = D^+ D^- = D^- D^+$$

$$\begin{aligned} D_2 u(x) &= \frac{u(x-h) - 2u(x) + u(x+h)}{h^2} \\ &= u''(x) + \frac{h^2}{12} u^{(3)}(x) + O(h^4) \end{aligned}$$

No $O(h^3)$ terms
due to symmetry

16

$$D^2 = \frac{1}{h^2} \begin{array}{c} \bullet - \circ - \bullet \\ | -2 \quad 1 \end{array} = \boxed{\begin{matrix} -2 & 1 \\ 1 & -2 & 1 \\ & \ddots & \ddots & \ddots \\ & & 1 & -2 & 1 \end{matrix}} \quad (1)$$

If periodic $\rightarrow (1)$

Option 2: Since the centered difference is second order, we are certain to get a 2nd order difference if we do:

$$\tilde{D}^2 = D_0^2 = \frac{1}{4h^2} \begin{array}{c} \bullet - \circ - \bullet \\ | \quad 2h \quad 2h \\ 1 \quad 0 \quad -2 \quad 0 \quad 1 \end{array}$$

(17)

We see that in \tilde{D}^2 , odd and even points are decoupled, so

if

$$u = \begin{bmatrix} \alpha \\ \beta \\ \alpha \\ \beta \\ \vdots \end{bmatrix} \text{ then } \tilde{D}^2 u = 0,$$

i.e., \tilde{D}^2 has a nontrivial null space unlike the continuum operator Δ_{xx} which has only constant functions in its null space (for periodic BCs)

Key lesson:

{ Accuracy is not the whole story
We also need to worry about
(physical) ROBUSTNESS

It is much better to find FDs
that preserve key properties of
elliptic operators, namely,
positive definiteness (for
 $-\Delta_{xx}$ with Dirichlet BCs) or
positive semi-definiteness with only
constants in null space ($-\Delta_{xx}$ + periodic)

Continuum picture

$$\nabla^2 u = \nabla \cdot (\nabla u)$$

$$\Delta u = \operatorname{div} \operatorname{grad} u$$

$$\Delta = \operatorname{div} \operatorname{grad}$$

Adjoint relation : $(\operatorname{div})^* = -\operatorname{grad}$

$$(\nabla \cdot \vec{u}, \vartheta)_{L_2} = \int_{\Omega} (\nabla \cdot \vec{u}) \vartheta \, dx = - \int_{\Omega} \vec{u} \cdot (\nabla \vartheta) \, dx$$

$$+ \int_{\partial\Omega} \left(\frac{\partial \vec{u}}{\partial n} \cdot \vec{n} \right) \vartheta \, dA = - (\vec{u}, \nabla \vartheta)_{L_2}$$

zero for periodic or
homogeneous Neumann or
Dirichlet

$\Rightarrow -\Delta = \nabla^* \nabla \geq 0$ is
 a symmetric positive - semidefinite
 operator with the null space
 being the null space of ∇ .
 For finite dimensional discretizations
 as matrices, ideally we want:

$$\left. \begin{array}{l} \nabla \rightarrow G \\ \nabla^* \rightarrow D \\ \nabla^2 \rightarrow L \end{array} \right\} \begin{array}{l} D = -G^T \quad (\text{L_2 adjoint}) \\ \text{in } \mathbb{R}^{n \times n} \\ -L = -DG = G^*G \geq 0 \end{array}$$

null space of G
only constants

Indeed:

$$D^+ = \begin{bmatrix} -1 & 1 \\ -1 & 1 \\ \vdots & \ddots \\ -1 & 1 \end{bmatrix} \quad D^- = \begin{bmatrix} 1 \\ -1 & 1 \\ \vdots & \ddots \\ -1 & 1 \end{bmatrix}$$

$$D^+ = -(D^-)^T \text{ as desired}$$

$$L = D^+ D^- = D^- D^+ \succcurlyeq 0$$

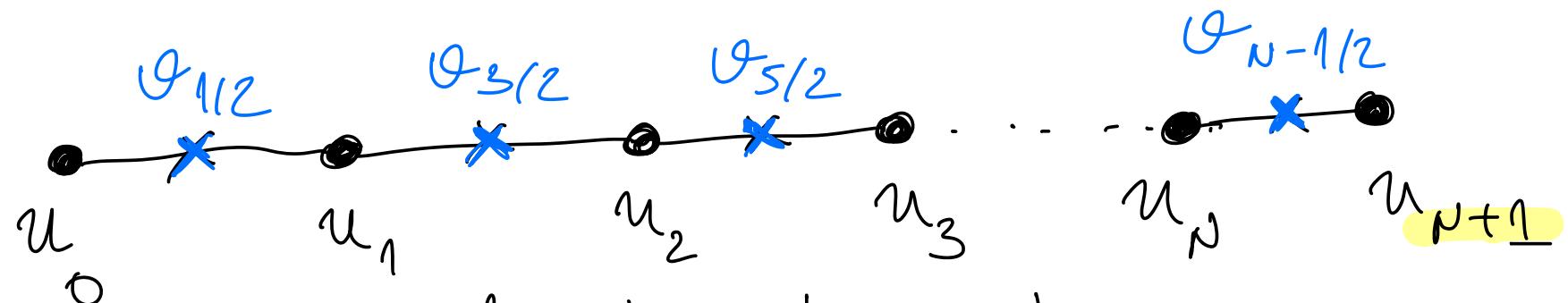
Null Space of D^{+-} is constant
vectors only, as desired.

How do we generalize this to higher dimensions (e.g., fluid flow) or the SPD operator

$$\mathcal{L}u = -\nabla \cdot (c(x)\nabla u)$$

$c(x) > 0 \quad n \sim$

Use a staggered grid!



Evaluate first derivatives on a grid staggered by $h/2$

(23)

$\{ u \text{ grid} = \text{nodes}$
 $\{ \vartheta \text{ grid} = \text{centers}$

Define linear mapping

$$\hat{D}_0 \approx \frac{\partial}{\partial x} : \text{nodes} \rightarrow \text{centers} \Rightarrow$$

$$\hat{D}_0^* \approx -\frac{\partial}{\partial x} : \text{centers} \rightarrow \text{nodes}$$

$$(\hat{D}_0 u)_{j+1/2} = \frac{u_{j+1} - u_j}{h} \underset{\text{centered}}{\xrightarrow{h \nwarrow}} u(x_{j+1/2}) + O(h^2)$$

$$(\hat{D}_0^* \vartheta)_i = \frac{\vartheta_{i+1/2} - \vartheta_{i-1/2}}{h} \quad \begin{matrix} \text{also} \\ \text{second} \\ \text{order} \end{matrix}$$

(24)

Confirm by yourself that

$$D^2 = - \hat{D}_0 \hat{D}_0^*$$

This can be generalized to PDE higher dimensions (see comp class in Fall), e.g., MAC or staggered grid discretization of the Navier-Stokes equations.

For $\mathcal{L} = - \partial_x c(x) \partial_x$ use

$$L = \begin{smallmatrix} & * \\ \hat{D}_0^* & C & \hat{D}_0 \end{smallmatrix} \quad (\text{HW5!})$$

Diagonal $C = \text{Diag} \{ c(x_{1/2}), c(x_{3/2}), \dots \}$ 25

Recall from Spectral methods:

Do not use chain rule

$$(C(x)u')' = C'u' + Cu'' \quad \times$$

since this destroys the adjoint structure of the elliptic operator.

Chain rule does not work for FDs.

$$(\mathcal{L}u)(x_i) = (Lu)_i = -\frac{1}{h} \left[c_{i+1/2} \left(\frac{u_{i+1} - u_i}{h} \right) + c_{i-1/2} \left(\frac{u_i - u_{i-1}}{h} \right) \right]$$

$c_{i+1/2} > 0 \rightarrow$

(26)

Theorems :

① L is an SPD matrix since

$$L = \hat{D}_0^* C \hat{D}_0$$

② Numerical solution satisfies just like the continuum solution
maximum principle does.

$$\nabla \cdot (c(x) \nabla u) = 0 \quad \text{in } \Omega \\ + \text{BCs}$$

$\Rightarrow u$ achieves extremum on boundary.

$$\min_{\Omega} (u(x \in \partial\Omega)) \leq u(x \in \Omega) \leq \max_{\Omega} (u(x \in \partial\Omega))$$

(27)

Is this true discretely?

In 1D:

$$\text{Lu} = 0 \Rightarrow$$

u_i^- = convex linear combination of
 u_{i-1}^- and $u_{i+1}^- \Rightarrow$

$$\min(u_{i-1}, u_{i+1}) \leq u_i \leq \max(u_{i-1}, u_{i+1})$$

max principle

"High" Order FDs

$$u''(jh) = \frac{1}{12h^2} \left(-u_{j-2} + 16u_{j-1} - 30u_j - u_{j+1} + 16u_{j+2} \right) + O(h^4)$$

Matrix is symmetric with
periodic BCS:

$$D_4 = \frac{1}{12h^2} \begin{bmatrix} -30 & 16 & -1 & \dots & 1 & 16 \\ 16 & -30 & 16 & -1 & \dots & 1 \\ -1 & 16 & -30 & 16 & 1 & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \end{bmatrix}$$

Is this a definite matrix?
Will max principle be satisfied? (29)

Compact Finite Differences

Let's now look at a trick to do a priori error correction in periodic domains in 1D:

$$u'' = f(x) \approx \frac{u(x-h) - 2u(x) + u(x+h)}{h^2}$$

$$= u''(x) + \frac{1}{12} u'''(x) h^2 + O(h^4)$$

$$= f(x) + \frac{1}{12} f''(x) h^2 + O(h^4)$$

But $f''(x) \approx (D^2 f)(x) \Rightarrow$

$$D^2 u = f + \frac{h^2}{12} D^2 f$$

$$D^2 u = \left(I + \frac{h^2}{12} D^2 \right) f$$

Solve this linear system instead of $D^2 u = f$ and you get 4th order compact FD:

$$D_4^2 = \left(I + \frac{h^2}{12} D^2 \right)^{-1} D^2$$

Is this a negative Leibniz matrix?

(31)

To analyze some of this, let us assume a periodic domain, allowing us to use a Discrete Fourier Series / Transform (DFD):

$$u_j = \sum_k \hat{u}_k e^{ikjh}$$

$$\begin{aligned} \Rightarrow (D^2 u)_j &= \sum_k e^{ik(j+1)h} - 2e^{ikh} + e^{ik(j-1)h} \hat{u}_k \\ &= \sum_k \left(\frac{e^{ikh} - e^{-ikh}}{h^2} \right) \hat{u}_k e^{ikh} \end{aligned}$$

(32)

$$(\hat{D}^2 u)_j = \sum_k (\hat{D}^2 u)_k e^{ikjh} =$$

$$- \sum_k \frac{\sin^2(kh/2)}{(h/2)^2} \hat{u}_k e^{ikh}$$

$$\Rightarrow (\hat{D}^2 \hat{u})_k = - \frac{\sin^2(kh/2)}{(h/2)^2} \hat{u}_k$$

$$\Rightarrow \hat{D}^2 = \text{Diag} \left\{ \frac{\sin^2(kh/2)}{(h/2)^2} \right\}$$

in Fourier Space

Compare this to continuum /
spectral :

$$\hat{\partial}_{xx} = -k^2$$

$$\frac{\sin(kh/2)}{(h/2)^2} = k^2 + O(k^4 h^2)$$
$$= k^2 \left(1 + O((kh)^2) \right)$$

"Symbol" of
3 pt Laplacian
This is called

von-Neumann analysis

second
order

Note

$$\frac{\sin^2(kh/2)}{(kh/2)^2} \geq 0$$

for $|kh| \leq \pi$ (actual range)

$\Rightarrow -D^2$ is symmetric positive semidefinite matrix in Fourier space
since diagonal (think why)
and D^2 is 2nd order accurate.

We can do the same now for
the other finite differences also!

E.g. compact FD:

$$(I + \frac{h^2}{12} D^2) u'' = D^2 u + O(h^4)$$

$$\left(1 - \frac{h^2}{12} \frac{\sin^2(kh/2)}{(kh/2)^2}\right) \hat{u}_k'' = -\frac{\sin^2(kh/2)}{(kh/2)^2} \hat{u}_k$$

$$\Rightarrow \hat{D}_4^2 = -\frac{\hat{D}_k^2}{1 - h^2/12 \hat{D}_k} \leq 0 \quad \text{for } |kh| < \pi$$

$$= -k^2 \left(1 - \frac{1}{240} (kh)^4\right)$$

Do this for
 \hat{D}_4^2 @ home

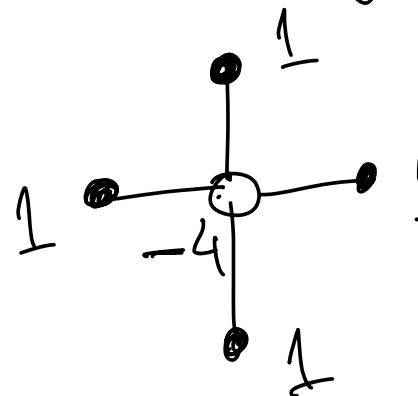
\uparrow
 fourth
 order

36

Two Dimensions

$$\nabla^2 u = u_{xx} + u_{yy} = f(x, y)$$

$$f_{ij} : \left(\frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h_x^2} \right) + \left(\frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{h_y^2} \right) = f_{ij}$$



$$1 \cdot 1 * \frac{1}{h^2} = 5^{pt}$$
Laplacian
(second order)

37

$$D_{2D,5}^2 = D_x^2 + D_y^2 = -L$$

$$-(Lu)_{i,j} = (\nabla^2 u)(x_i, y_j) + \frac{h^2}{12} (u_{xxxx} + u_{yyyy})$$

local truncation
 error (2nd order)

This inherits all of the nice properties of the 3^{pt} Laplacian in 1D, e.g., maximum principle is still satisfied (prove on your own).

(33)

But this is not the only good option in 2D. There is also a ^{gpt} Laplacian

$$(\nabla^2_{2D,9} u)_i = \frac{1}{h^2} \left(-\frac{20}{6} u_i + \frac{4}{6} u_{i+1} + \frac{1}{6} u_{i+2} \right)$$

$$(\nabla^2_{2D,9} u)_{i,j} = (\nabla^2 u)(x_i, y_j) + \frac{h^2}{12} (u_{xxxx} + u_{yyyy} + 2u_{xx,yy}) + O(h^4)$$

(39)

Observe :

$$(\partial_{xx} + \partial_{yy})^2 = \partial_{xxxx} + \partial_{yyyy} + 2\partial_{xxyy}$$

$$\left(\begin{smallmatrix} 2 \\ D \\ 2D, 9 \end{smallmatrix} u \right)_{i,j} = (\nabla^2 u)(x_i, x_j) + \frac{h^2}{12} \nabla^2 (\nabla^2 u)$$

4th order for Laplace eq + $O(h^4)$

Still 2nd order accurate (only)

but now error is isotropic so
the grid "artifacts" or "imprints"
in the numerical solution will be reduced.

Accuracy is not the whole story!

think why the compact FD

$$D_{2D,9}^2 u = \left(I + \frac{h^2}{12} D_{2D,5 \text{ or } 9}^2 \right) f$$

is 4th order and isotropic to
second order.

This is in fact a great FD
discretization of the Poisson equation
in two dimensions.

Can be generalized to 3D!
(not here)

Boundary Conditions

Consider first Dirichlet BCs

$$u(0) = u_0 \quad u(L) = u_L$$

This means that u_0 and u_N are known and not variables to solve for, so just set

$$u_0 = u_0, \quad u_{N+1} = u_L$$

e.g.

$$D^2 = \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & 1 \\ & & \ddots & \ddots & 1 \\ & & & 1 & -2 \end{bmatrix}$$

(42)

E.g. $\begin{cases} u''(x) = f(x), x \in (0, 1) \\ u(0) = \alpha, u(1) = \beta \end{cases}$

$$\frac{1}{h^2} (u_{j-1} - 2u_j + u_{j+1}) = f(x_j)$$

$j = 1, \dots, N$

$$u_0 = \alpha, \quad u_N = \beta$$

We need to solve a linear system $\vec{A}\vec{u} = \vec{f} = \vec{f} + \text{inhomogeneous}$

\Rightarrow Solving elliptic linear PDEs amounts to solving large linear systems (43)

$$A = D^2, \quad \vec{f} = \begin{bmatrix} f_1 - \alpha/h^2 \\ f_2 \\ \vdots \\ f_{N-1} \\ f_N - \beta/h^2 \end{bmatrix}$$

For periodic BCs

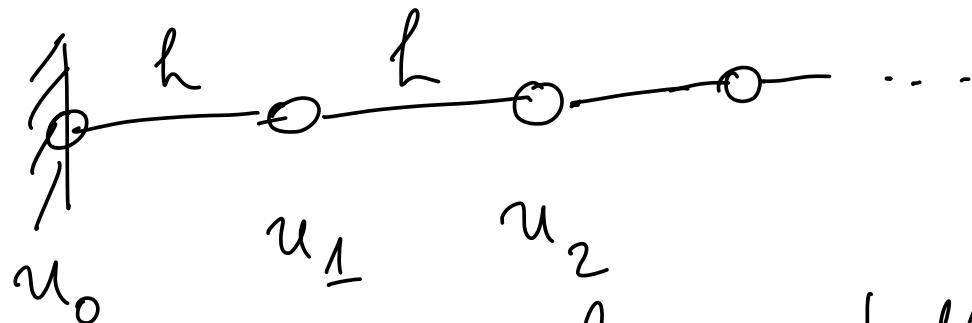
$$A = D^2 = \begin{bmatrix} -2 & 1 & & & & 1 \\ 1 & -2 & 1 & \ddots & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & 1 \\ & & & 1 & -2 & \\ & & & & & \ddots \end{bmatrix}$$

We will return to this circulant matrix shortly.

Now let's consider Neumann BCS:

$$u'(0) = 6$$

Now, u_0 is not known so it is a real variable:



There are a few different ways to view / think about imposing BCs. Impose BC on boundary:

- a) { instead of PDE
- b) { in addition to PDE

Eg. of a)

$$\left(\frac{D_u}{h}\right)_{1/2} = \frac{u_1 - u_0}{h} = u'(0) + O(h) = 6$$

which is clearly only first order.

In the code, we can keep u_0 as a variable and add this as an additional equation:

$$u_0 = u_1 - 6h$$

This feels like treating u_0 as a ghost cell and extrapolating to first order from interior to u_0 .

Example of Q:

Use a 2nd centered difference instead:

$$\frac{1}{2h} (u_1 - \underset{\text{ghost cell}}{\tilde{u}_{-1}}) = 6$$

$$(*) \dots u_{-1} = u_1 - 2h 6 \leftarrow \begin{array}{l} \text{linear} \\ \text{extrapolation} \\ \text{to ghost cell} \end{array}$$

But since now both u_0 and u_{-1} are variables unknown, we need also to enforce the PDE as well at the boundary:

$$\frac{1}{h^2} (u_{-1} - 2u_0 + u_1) = f_0 = f(x_0)$$

↙ substitute (*)

$$\frac{u_1 - u_0}{h} = 6 + \frac{h}{2} f(x_0)$$

To see this is now second order,
use Taylor series:

$$\begin{aligned}\frac{u_1 - u_0}{h} &= u'(x_0) + \frac{h}{2} u''(x_0) + O(h^2) \\ &= 6 + \frac{h}{2} f(x_0) + O(h^2)\end{aligned}$$

In matrix notation:

$$A = \begin{bmatrix} -1 & 1 & \cdots & \cdots & \cdots \\ 1 & -2 & 1 & \cdots & \cdots \\ 0 & 1 & -2 & 1 & \cdots \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix}$$

is negative definite

for 6th a) and b), but

for b)

$$z^f = \begin{bmatrix} 6h + \frac{h^2}{2} f(x_0) \\ 0 \\ 0 \\ \vdots \end{bmatrix}$$

second order correction

Option a) revisited :

How about we use a second-order one-sided tree-point FD:

$$u'(x_0) = -\frac{1}{h} \left(\frac{3}{2} u_0 - 2u_1 + \frac{u_2}{2} \right) + O(h)$$
$$= u'(0) + O(h^2) = 6 + O(h^2)$$

$$A = \frac{1}{h^2}$$

$$\begin{matrix} -3h/2 & +2h & -h/2 & \dots \\ 1 & -2 & 1 & \\ 0 & 1 & -2 & 1 \\ \vdots & \ddots & \ddots & \end{matrix}$$

Matrix is no longer symmetric!

How to implement?

We need two pieces :

- a) Discretize forward operator \mathcal{L} so that you can compute $\mathcal{L}u \approx \mathcal{X}u$
- b) Solve linear system efficiently to discretize Inverse operator

Key: For finite difference methods matrix L is very sparse!

(A)

Forward operator

Best implemented using ghost or virtual points (cells) to store u_0 and u_n even though they are not actual variables:

In pseudo - Fortran:

function ApplyOp(Op, how) result(Lu)

real, dimension(:), intent(in) :: u

real, dimension(size(u)) :: Lu

logical, intent(in) :: how ! homogeneous
des?

!-----

* real, dimension(0 : size(u)+1) :: u-ext

integer :: N

(B)

$N = \text{size}(u)$
 $u_{\text{ext}}(1:N) = u$! copy input
call Fill Ghost (u_{ext})

implement homogeneous or inhomogeneous BCs
(physical BCs)

! Now do finite difference:

$$Lu(1:N) = (u_{\text{ext}}(0:N-1) + u_{\text{ext}}(2:N)) \\ - 2 * u_{\text{ext}}(1:N)$$

↑
! can be optimized / vectorized

end function Apply Op

c

subroutine FillGhost (u_{ext} , h_{om})
 real, dimension (0:), intent(inout) :: u_{ext}
 logical, intent(in) :: h_{om}
 integer :: N
 $N = \text{size} (u_{\text{ext}}) - 2$! two ghost values
 if (periodic) then
 $u_{\text{ext}}(0) = u_{\text{ext}}(N)$
 $u_{\text{ext}}(N+1) = u_{\text{ext}}(1)$
 else ! Dirichlet BCs
 if (h_{om}) then
 $u_{\text{ext}}(0) = 0 ; u_{\text{ext}}(N+1) = 0$
 else
 $u_{\text{ext}}(0) = \alpha ; u_{\text{ext}}(N+1) = \beta$
 end if ; end if

(D)

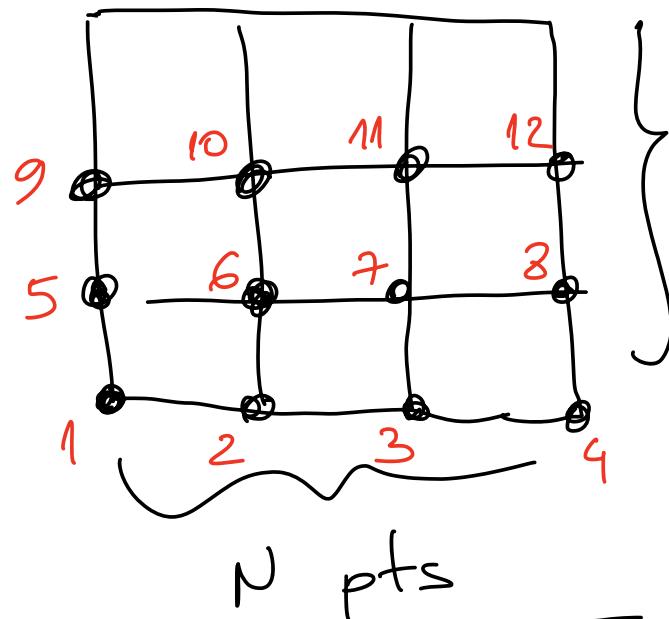
Inverse Operator

We need to solve $Au = f$
efficiently for very large A
(e.g. $256^3 \approx 17K$ DOFs, so
 $A = [17K \times 17K]$ matrix)

Options:

- 1) Use a direct sparse solver
Does this work in 1D? [Discuss]
Does a direct solver "work" in 2D/3D?
- 2) Use an iterative solver:
How fast does it converge?
How can we precondition? E

Standard 5pt Laplacian in 2D:



\leftarrow standard numbering
(why is it arbitrary?)

$$A = \frac{1}{h^2}$$

$$\begin{bmatrix} -4 & 1 & & & & \\ 1 & -4 & 1 & & & \\ & 1 & -4 & 1 & & \\ & & 1 & -4 & 1 & \\ & & & 1 & -4 & \\ & & & & 1 & -4 \end{bmatrix}$$

Annotations in pink: "diagonals" with arrows pointing to the main diagonal and the super/sub-diagonals; "zeros" with arrows pointing to the off-diagonal elements.

(F)

Think about why solving
 $A u = f$
with a direct solver gives
computational complexity of $O(N^3)$
i.e. $O(N_{\text{pts}}^{3/2})$ where $N_{\text{pts}} = N^2$.

This is not linear. Can we
do (log) linear solve?

{ Theorem (George) : Any direct solver
for $D^2 u = f$ requires at least
 $O(N_{\text{pts}}^{3/2})$ FLOPs and memory

G

But, if we are in a periodic domain, we can solve in (log), linear time using the FFT.

$$\hat{u}_{k_x, k_y} = \frac{-\hat{f}_{k_x, k_y}}{\sin^2(k_x h_x/2) + \sin^2(k_y h_y/2)}$$

Same works for $D_{2D, 9}^2$ or D_4^2 or any FD! But the best is still spectral solver

$$\hat{u} = -\frac{1}{k^2} \hat{f}$$

(H)

How about an iterative solver,
 say conjugate gradients (why?)
 convergence of PCG:

$$\frac{\|e_k\|_A}{\|e_0\|_A} < 2 \left(\frac{\sqrt{K_A} - 1}{\sqrt{K_A} + 1} \right)^k \approx 2 e^{-2k/\sqrt{K_A}}$$

$$K_A = L_2 \text{ conditioning number of } A$$

$$\Rightarrow \ln \frac{\|e_k\|_A}{\|e_0\|_A} \sim -2k/\sqrt{K_A} \Rightarrow$$

$$\# \text{ of CG iterations} \sim \sqrt{K_A} \sim N$$

$$\Rightarrow \text{total cost} = O(N^2) \cdot N = O(N^3) !$$

(I)

So to get better than $O(N_{\text{pts}}^{3/2})$
we need a preconditioner.

A very effective preconditioner
for elliptic PDEs is the
multigrid method (algebraic for
unstructured FEM grids or geometric
for structured FD / FV grids).

Multigrid achieves log linear
complexity for elliptic PDEs
in both 2D and 3D, just
like the FFT! Even with BCS.

③

Stability of FD methods

(for elliptic PDES)

$$\begin{cases} \mathcal{L}u = f \text{ in } \Omega \\ \mathcal{B}u = g \text{ on } \partial\Omega \end{cases} \Rightarrow Au = F$$

True PDE solution :

$$\hat{u} = [u(x_1), \dots, u(x_N)]$$

↑
pointwise solution

Global error

$$E = u - \hat{u}$$

I

NA question: Does $\|E\| \rightarrow 0$ as $N \rightarrow \infty$

Local (truncation) error

$$\bar{z} = \hat{A}\hat{u} - F \quad (\text{LTE})$$

can be estimated by simple Taylor series for smooth solutions.

e.g. $\|\bar{z}\| = O(h^2)$

$$\left\{ \begin{array}{l} \hat{A}\hat{u} = F + \bar{z} \\ A u = F \end{array} \right. \Rightarrow AE = -\bar{z}$$

Error solution satisfies same equation as
solution but with LTE on the R.h.s.

II

From the Taylor series we used to derive the 4th order compact FD for ∂_{xx} , we saw

$$e(x) \approx -\frac{h^2}{12} u'''(x) + O(h^4)$$

↑

$$\approx -\frac{h^2}{12} f'(x)$$

$$E_i \approx e(x_i)$$

So we estimate the error to be $O(h^2)$, meaning the method converges with 2nd order accuracy.

Is this true?



$$E = A^{-1} \bar{z} \Rightarrow$$

$$\|E\| \leq \|A^{-1}\| \|\bar{z}\|$$

\Downarrow
 $\begin{matrix} \| \\ O(h^2) \end{matrix} \quad \begin{matrix} \| \\ O(1) \end{matrix} \quad \begin{matrix} \| \\ O(h^2) \end{matrix}$

A method to solve a linear
BVP is stable if

$$\|A^{-1}\| \in C \quad + h < h_0$$

"uniform" ellipticity
of PDE is preserved
by discretization

\uparrow
grid
spacing

IV

Stability + consistency \Rightarrow convergence

$$\|A^{-1}\| < c + \text{LTE} = O(h^p) \Rightarrow \|E\| = O(h^p)$$

Important: Choice of norm now matters, since infinite dimensional as $h \rightarrow 0$!

Let's start with L_2 .

Recall $\|A\|_2 = \rho(A) = \max_p |\lambda_p|$

Symmetric \uparrow spectral radius

$$\Rightarrow \|A^{-1}\|_2 = \left(\min_p |\lambda_p| \right)^{-1}$$

IV

Let's take Poisson eq. in 1D
with Dirichlet BCs:

$$\begin{cases} u'' = f \text{ on } [0, 1] \\ u(0) = u(1) = 0 \end{cases}$$

$$A = \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & 1 \\ & & \ddots & \ddots & 1 \\ 1 & & & -2 & \end{bmatrix}$$

Based
guess

on continuum eigenfunctions,
that eigenvectors are:

$$u_j^{(P)} = \sin(P\pi j h)$$

(VI)

Indeed, plug into $A u^{(P)} = \lambda_p u^{(P)}$
to confirm

$$\begin{aligned}\lambda_p &= \frac{2}{h^2} (\cos(p\pi h) - 1) \\ &= -\underbrace{\pi^2 p^2}_{\text{continuum}} + \underbrace{\frac{1}{12} \pi^4 p^4 h^2}_{\text{error} = O(h^2)} + O(h^4)\end{aligned}$$

Smallest eigenvalue = smallest
wave vector (largest wavelength)

Note: $K_2(A) = |\lambda_{\max}| / |\lambda_{\min}| = O(N^2)$

VII

$$\lambda_1 = \frac{2}{h^2} (\cos(\pi h) - 1) \approx -\pi^2 + O(h^2)$$

$$\Rightarrow \|A^{-1}\| \lesssim \frac{1}{\pi^2} = \text{const}$$

and therefore the method is
stable \Rightarrow convergence to 2nd order.

Convergence in L_∞

Since N is finite (though large),
linear algebra says

$$\|E\|_\infty \leq \frac{1}{\sqrt{h}} \|E\|_2 = O(h^{3/2})$$

VIII

But turns out this is too pessimistic and, in fact,
 $\|E\|_\infty = O(h^2)$ as well.

Remember $\|A\|_\infty$ is the largest absolute column sum:

$$\|A\|_\infty = \max_i \sum_{j=1}^n |a_{ij}|$$

What are the columns of A^{-1} ?

(IX)

the j^{th} column of A^{-1} is

$$\tilde{G}^{(j)} = A^{-1} e_j = A^{-1} \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \leftarrow j^{\text{th}} \text{ entry}$$

$$\Rightarrow A \tilde{G}^{(j)} = e_j$$

This looks like a discretization
of

$$\begin{cases} \mathcal{L} G = \delta(x_j) \\ \mathcal{B} G = 0 \end{cases}$$

$$\Rightarrow \tilde{G}_i^{(j)} \sim G(x_i; x_j)$$

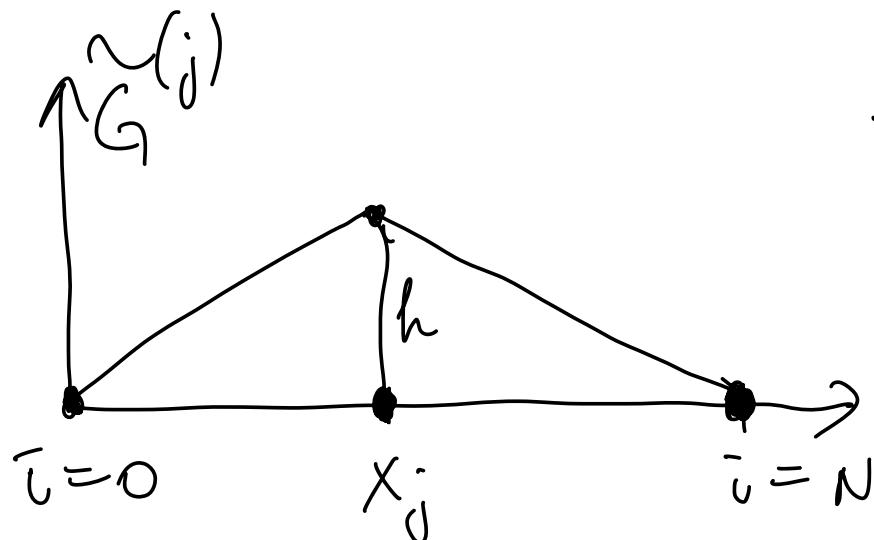
(x)

This means the columns of A^{-1}
 are discrete Green's functions
 of the elliptic PDE. Specifically
 $\{G^{(j)}\}$ tells us how the LFE at
 node / point j spreads to the
 other points.

For the Poisson eq. with Dirichlet,
 $G_i^{(j)} = G(x_i; x_j)$ (exactly!)
 $= h \begin{cases} (x_{j-1}) x_i, & i=1, \dots, j \\ (x_i - 1) x_j, & i=j, j+1, \dots, N \end{cases}$

$$G_i^{(j)} = h G(x_i, x_j) = A_{ij}^{-1}$$

$$\Rightarrow \|A^{-1}\|_\infty \leq \max_{1 \leq j \leq N} \|G^{(j)}\|_1 \leq N \cdot h = L = 1$$



\Rightarrow method is also
 $O(h^2)$ in L_∞
 (and in L_1)

(XII)

Now imagine we made a local error of $O(h^q)$ at only a few points, e.g., at the boundary

$$|\bar{z}_j| = O(h^q), \quad q < p$$

$$\Rightarrow E^{(j)} = \hat{z}^{(j)} - \hat{u} = A^{-1} \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} O(h^q)$$

$$\|E^{(j)}\|_{\infty} \leq \|G^{(j)}\|_{\infty} O(h^q) = O(h^{q+1})$$

XIII

This shows that even if we made a first order error at the boundary, the scheme would still be 2^{nd} order accurate globally.

Elliptic regularity often implies that we can make a large error locally without polluting the error globally.

Elliptic PDEs smear & smooth the error globally, often

But not always.

Consider on your own Neumann
BCs, $u'' = f(x)$, $u'(0) = u'(1) = 0$
Now a local error of $O(h^q)$
causes a global error of $O(h^q)$,
so we do not gain an extra order.

Note: Numerically, you can compute
 $G(j)$ by making the r.h.s. of
the PDE be a "delta function",
which is useful in 2D or for
complicated PDEs 