FINITE DIFFERENCE METHODS FOR ELLIPTIC PDEs

A. DONEV, 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

E.g.: Sturm–Liouville (SL) BVPs in 1D:

$$\begin{cases} (\mathcal{L}u)(x) = -(p(x)u'(x))' + q(x)u(x) \\ p(x) > 0, \quad q(x) > 0 \quad \text{on } [a,b] \end{cases}$$

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?

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

$$\begin{cases} \mathcal{L} u_k = \lambda_k u_k \\ B u_k = 0 \end{cases}$$

eigen function

eigenvalue

Countably infinitely many eigenpairs

(λ_k, u_k), $k = 0, 1, 2, \ldots$

for bounded domains

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, then

1) All eigenvalues are real
2) There is a complete set of orthonormal eigenfunctions in L_w that is enumerable.

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

$$\lambda_k > 0 \quad \forall k$$

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

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

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

If $f = \sum_k b_k u_k$, where

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

then

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

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

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

$$\begin{cases} \mathcal{L} G(\vec{x};\vec{y}) = \delta(\vec{y}\in\Omega) \\ \mathcal{B} G(\vec{x}\in\partial\Omega;\vec{y}) = 0 \end{cases}$$

↙ quadrature!

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

+ particular solution

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*)

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:

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

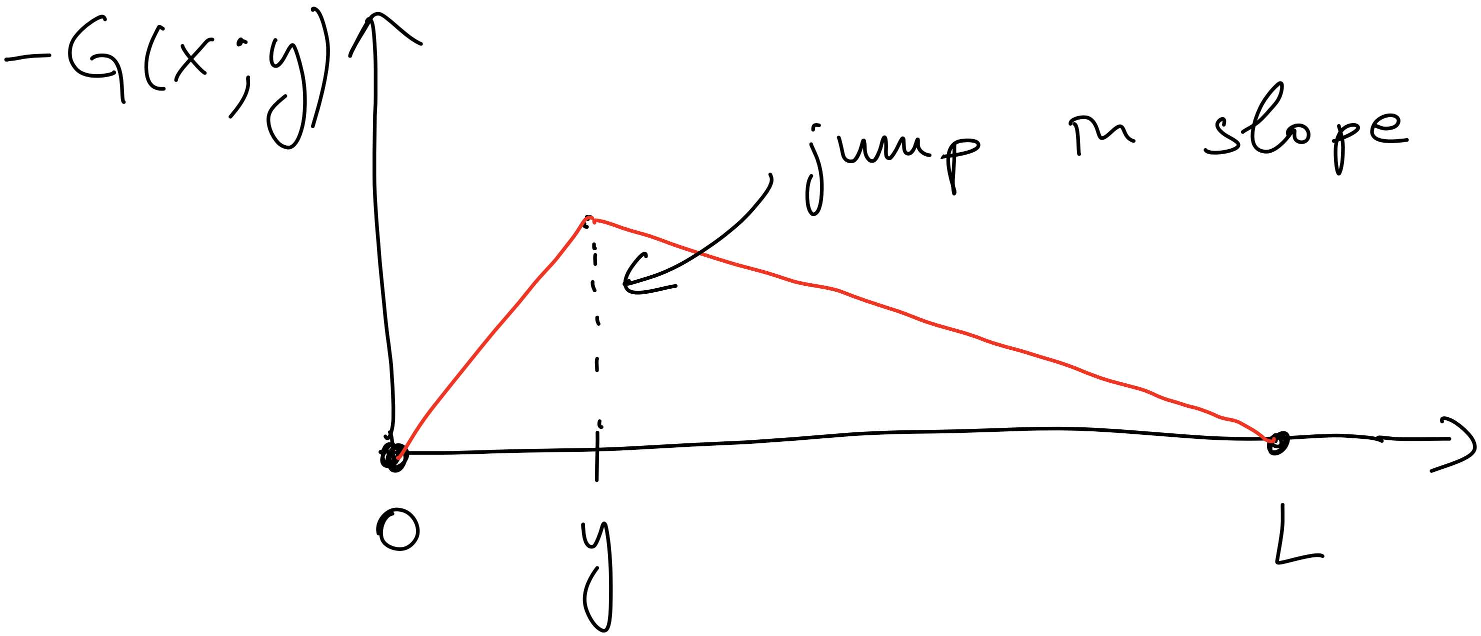
$$\lambda_k = \left(\frac{2\pi k}{L}\right)^2$$

Green's function :

$$\begin{cases} G'' = \delta(y) \\ G'(y^+) - G'(y^-) = 1 \end{cases} \Rightarrow$$

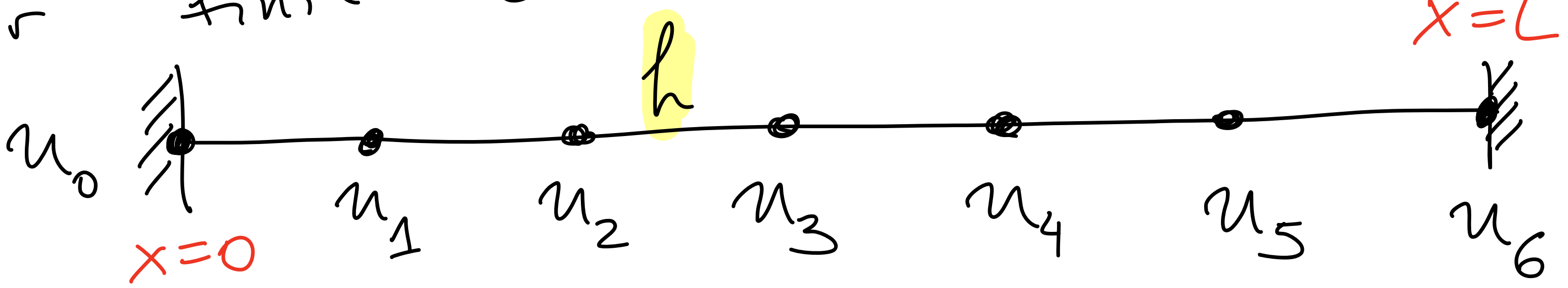
$$G(0) = G(L) = 0$$

$$G(x,y) = \frac{1}{L} \cdot \begin{cases} (y-L)x, & x \leq y \\ (x-L)y, & x \geq y \end{cases}$$



FINITE DIFFERENCE (FD) methods

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



For <u>equispaced</u> grids with grid spacing h :

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

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

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

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

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

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

one-sided differences

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

Observe that D_0 does **not** involve $u(x)$ 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...

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

↑

down-biased FD (good for hyperbolic)

The local truncation error (LTE) can be computed easily using Talyor 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:

$$D_3 = \frac{1}{h}\left[\begin{array}{ccccccc} \bullet & — & \bullet & — & \circ & — & \bullet \\ \frac{1}{6} & & -1 & & \frac{1}{2} & & \frac{1}{3} \end{array}\right]$$

Second-order derivative

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

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

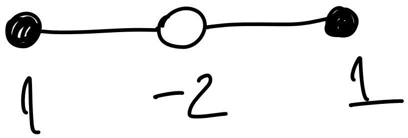
$$D^2 = D^+ D^- = D^- D^+$$

$$D_2 u(x) = \frac{u(x-h) - 2u(x) + u(x+h)}{h^2}$$

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

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

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

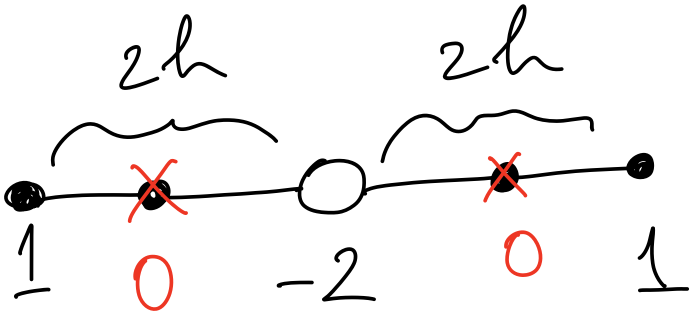


$$\equiv \begin{bmatrix} -2 & 1 & & & \textcircled{1} \\ 1 & -2 & 1 & & \\ & 1 & \ddots & \ddots & \\ & & \ddots & \ddots & \ddots \\ \textcircled{1} & & 1 & -2 & 1 \end{bmatrix}$$

If periodic

Option 2: Since the centered difference $D_0 \leftrightarrow \partial_x$ is second order, we are certain to get a 2nd order finite difference if we do:

$$\widetilde{D}^2 = D_0^2 = \frac{1}{4h^2}$$



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

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

i.e., $\tilde{D}^2$ has a nontrivial null space

unlike the continuum operator ∂_{xx} which has only constant functions in it's 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 $-\partial_{xx}$ with Dirichlet BCs) or positive semi definiteness with only constants in null space ($-\partial_{xx}$ + periodic)

Continuum picture

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

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

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

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

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

$$+ \underbrace{\int_{\partial\Omega} \left(\frac{\partial \vec{u}}{\partial n} \cdot \vec{n} \right) \vartheta \, dA}_{\text{zero for periodic or homogeneous Neumann or Dirichlet}} = -(\vec{u}, \nabla \vartheta)_{L_2}$$

$\Rightarrow \quad -\Delta = \nabla^* \nabla \succeq 0$ is a symmetric positive-semidefinite operator with the null space being the null space of ∇.

For finite dimensional discretizations as matrices, ideally we want:

$\nabla \to G$

$\nabla \cdot \to D$

$\nabla^2 \to L$

$$\begin{cases} D = -G^T & (L_2 \text{ adjoint in } \mathbb{R}^{n \times n}) \\ -L = -DG = G^* G \succeq 0 & \\ \text{null space of } G & \\ \text{only constants} & \end{cases}$$

Indeed:

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

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

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

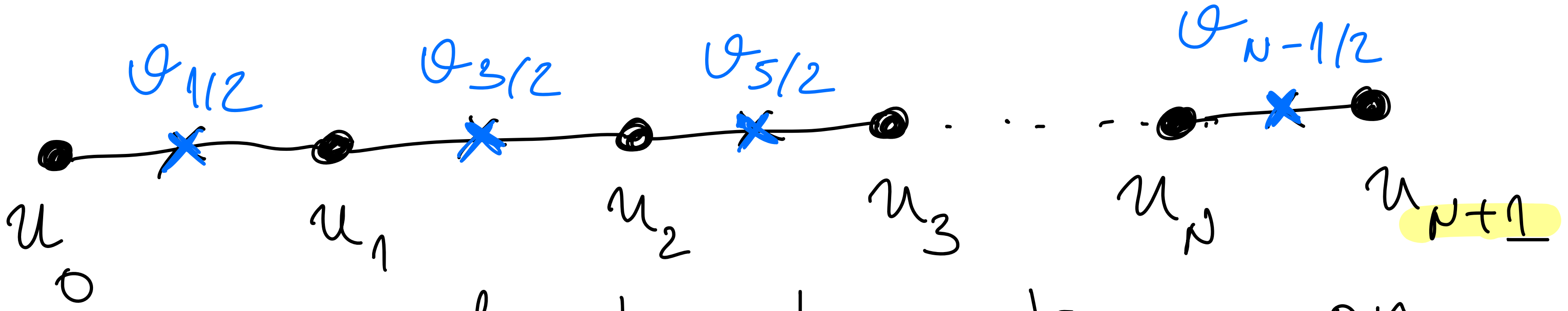
Null space of $D^{+/-}$ is constant vectors only, as desired.

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

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

$$c(x) > 0 \quad \text{in} \quad \Omega$$

Use a staggered grid!



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

$$\begin{cases} u \text{ grid} = \text{nodes} \\ v \text{ grid} = \text{centers} \end{cases}$$

Define linear mapping

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

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

$$\left(\hat{D}_0 u\right)_{j+1/2} = \frac{u_{j+1} - u_j}{h} = u'(x_{j+1/2}) + O(h^2)$$

centered

$$\left(\hat{D}_0^{*} v\right)_i = \frac{v_{i+1/2} - v_{i-1/2}}{h}$$

also second order

Confirm by yourself that

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

This can be generalized to higher dimensions (see comp PDE 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

$$L = \hat{D}_0^* C \hat{D}_0 \qquad (\text{HW5!})$$

diagonal $C = \text{Diag}\{c(x_{1/2}), c(x_{3/2}), \ldots\}$

Recall from spectral methods:

Do <u>NOT</u> use chain rule

$$(c(x)\,u'(x))' = c'u' + c\,u'' \quad ✗$$

since this destroys the adjoint structure of the elliptic operator.

Chain rule does not work for FDs.

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

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

Theorems :

① L is an **SPD** matrix since

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

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

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

$\Rightarrow$ u achieves extremum on boundary!

$$\min(u(x \in \partial\Omega)) \le u(x \in \Omega) \le \max(u(x \in \partial\Omega))$$

Is this true discretely?

In 1D:

$$Lu = 0 \implies$$

$$u_i = \underbrace{\frac{c_{i+1/2}}{c_{i-1/2}+c_{i+1/2}}}_{\text{in } [0,1]} u_{i+1} + \underbrace{\frac{c_{i-1/2}}{c_{i-1/2}+c_{i+1/2}}}_{\text{in } [0,1]} u_{i-1}$$

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

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

"High" order FDs

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

Matrix is symmetric with periodic BCs:

$$\tilde{D}_4^{2} = \frac{1}{12h^2}\begin{bmatrix} -30 & 16 & -1 & \cdots & 1 & 16 \\ 16 & -30 & 16 & -1 & \cdots & 1 \\ -1 & 16 & -30 & 16 & 1 & \cdots \\ & & \cdots & & & \end{bmatrix}$$

Is this a <u>definite</u> matrix?

? Will max principle be satisfied?

Compact Finite Differences

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

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

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

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

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

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

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

Solve this linear system <u>instead</u> 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 definite matrix?

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

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

$$\Rightarrow (D^2 u)_j = \sum_k \frac{e^{ik(j+1)h} - 2e^{ikjh} + e^{ik(j-1)h}}{h^2} \hat{u}_k$$

$$= \sum_k \left(\frac{e^{ikh} - 2 + e^{-ikh}}{h^2} \right) \hat{u}_k e^{ikjh}$$

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

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

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

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

in Fourier Space

Compare this to continuum/ spectral:

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

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

$$= -k^2\left(1 + O\left((kh)^2\right)\right)$$

second order

"symbol" of 3pt Laplacian

This is called <u>von-Neumann analysis</u>

Note $\dfrac{\sin^2(kh/2)}{(h/2)^2} \geqslant 0$

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

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

and D^2 is 2^{nd} order accurate.

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

E.g. compact FD:

$$\left(I + \frac{h^2}{12} D^2\right) D_4^2 u = D^2 u + O(h^4)$$

$$\left[1 - \frac{h^2}{12} \frac{\sin^2(kh/2)}{(h/2)^2}\right] \left(\hat{D}_4^2 \hat{u}\right)_k = - \frac{\sin^2(kh/2)}{(h/2)^2} \hat{u}_k$$

$$\Rightarrow \left(\hat{D}_4^2\right)_{kk} = - \frac{\hat{D}_k^2}{1 - h^2/12\, \hat{D}_k^2} \le 0 \quad \text{for } |kh| < \pi$$

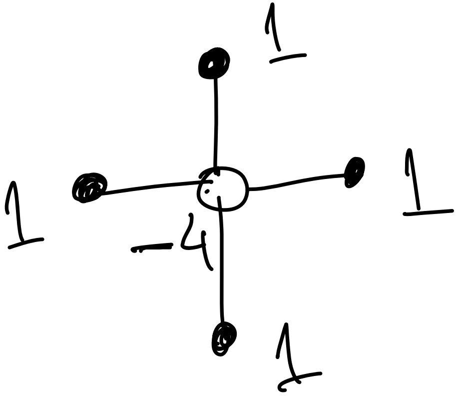
$$= -k^2 \left[1 - \frac{1}{240} \underset{\substack{\uparrow \\ \text{fourth} \\ \text{order}}}{(kh)^4} + O((kh)^6)\right]$$

$\boxed{\text{Do this for } \tilde{D}_4^2 \text{ @ home}}$

Two Dimensions

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

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



$$* \frac{1}{h^2} = 5^{pt} \text{ Laplacian}$$

(second order)

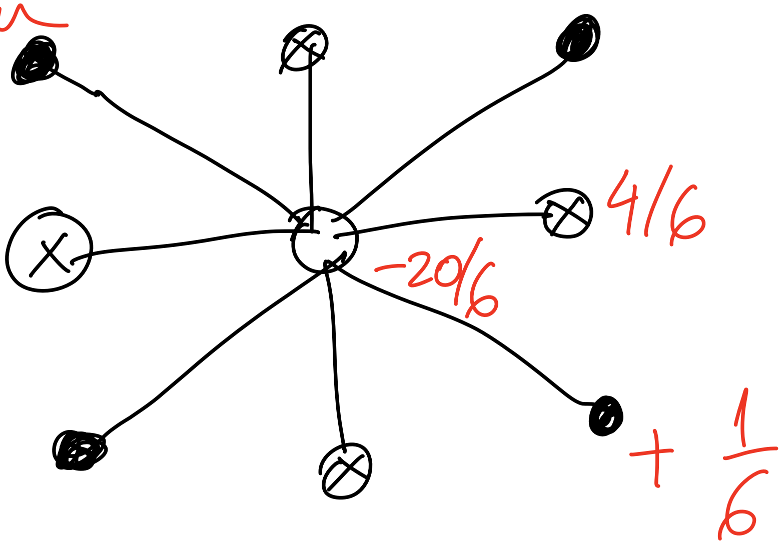
$$D^2_{2D,5} = D^2_x + D^2_y = -L$$

$$-(Lu)_{i,j} = (\nabla^2 u)(x_i, y_j) + \underbrace{\frac{h^2}{12}(u_{xxxx} + u_{yyyy})}_{\text{local truncation error (2nd order)}}$$

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

But this is not the only good option in 2D. There is also a 9pt Laplacian

$$D^2_{2D,9}\,u =$$



$$\ast\ \frac{1}{h^2}$$

$$\left(D^2_{2D,9}\,u\right)_{i,j} = (\nabla^2 u)(x_i, y_j) + \frac{h^2}{12}\left(u_{xxxx} + u_{yyyy} + 2u_{xxyy}\right) + O(h^4)$$ ㊴

Observe :

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

$$\left(D^2_{2D,9}\, u\right)_{i,j} = \left(\nabla^2 u\right)(x_i, x_j) + \frac{h^2}{12}\nabla^2\left(\nabla^2 u\right) + O\left(h^4\right)$$

4th order for <u>Laplace</u> eq

Still 2nd order accurate (only) but now error is <u>isotropic</u> so the grid "artifacts" or "imprints" in the numerical solution will be reduced.

Accuracy is <u>not</u> the whole story!

Think why the compact FD

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

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

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

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

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

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

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

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

$$j = 1, \dots, N$$

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

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

$\Rightarrow$ Solving elliptic linear PDEs amounts to solving $\underline{\underline{\text{large}}}$ linear systems

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

For periodic BCs

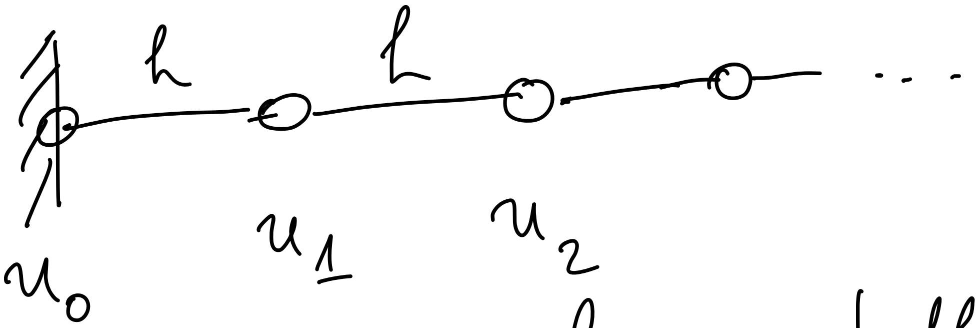
$$A = D^2 = \begin{bmatrix} -2 & 1 & \cdots & 1 \\ & \ddots & \ddots & 1 \\ & & \ddots & \\ 1 & \cdots & 1 & -2 \end{bmatrix}$$

We will return to this <u>circulant</u> matrix shortly.

Now let's consider **Neumann BCs**:

$$u'(0) = \sigma$$

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)

$$\left(\hat{D}_0 u\right)_{1/2} = \frac{u_1 - u_0}{h} = u'(0) + O(h) = \sigma$$

which is clearly only first order.

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

$$u_0 = u_1 - \sigma h$$

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

Example of b):

Use a 2^{nd} centered difference instead:

$$\frac{1}{2h}\left(u_1 - \underset{\overset{\uparrow}{\text{ghost cell}}}{u_{-1}} \right) = \sigma$$

$$(*) \ldots u_{-1} = u_1 - 2h\sigma \quad \leftarrow \text{linear extrapolation to ghost cell}$$

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

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

substitute (*)

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

To see this is now <u>second</u> order, use Taylor series:

$$\frac{u_1 - u_0}{h} = u'(x_0) + \frac{h}{2} u''(x_0) + O(h^2)$$

$$= \sigma + \frac{h}{2} f(x_0) + o(h^2)$$

In matrix notation:

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

is negative definite

for both a) and b), but for b)

$$\underset{\sim}{f} = \begin{bmatrix} 6h + \underbrace{\frac{h^2}{2} f(x_0)}_{\text{second order correction}} \\ f_1 \\ f_2 \\ \vdots \end{bmatrix}$$

Option a) revisited :

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

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

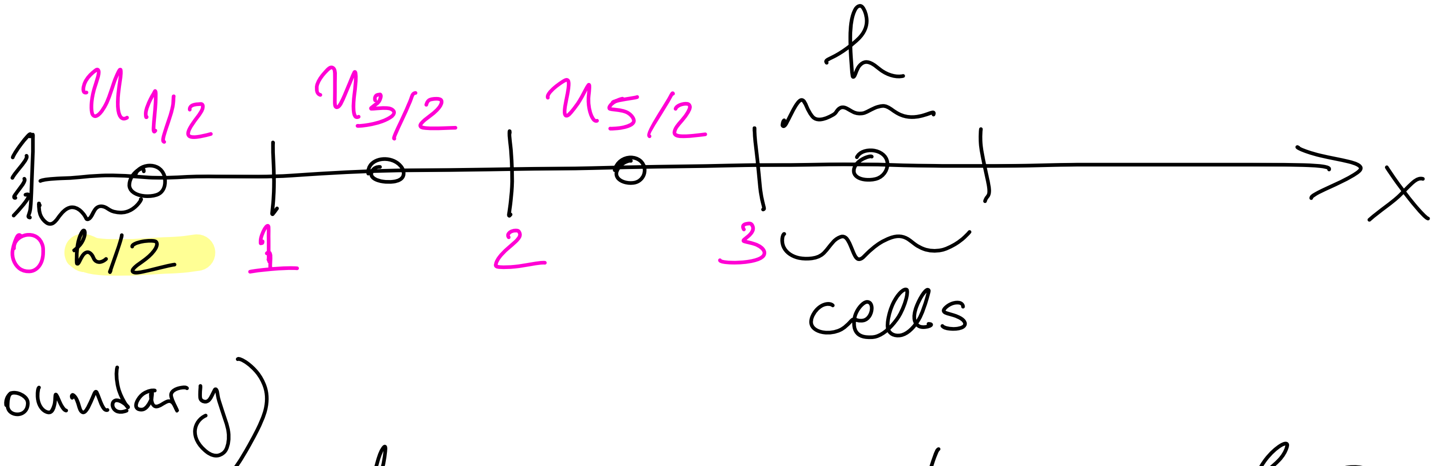
$$= u'(0) + O(h^2) = \sigma + O(h^2)$$

$$A = \frac{1}{h^2}\left[\begin{array}{cccc} -3h/2 & +2h & -h/2 & \cdots \\ 1 & -2 & 1 & \\ 0 & 1 & -2 & 1. \\ \vdots & \cdots & \cdots & \cdots \end{array}\right.$$

Matrix is $\underline{\underline{\text{no}}}$ longer symmetric!

Staggered Grid

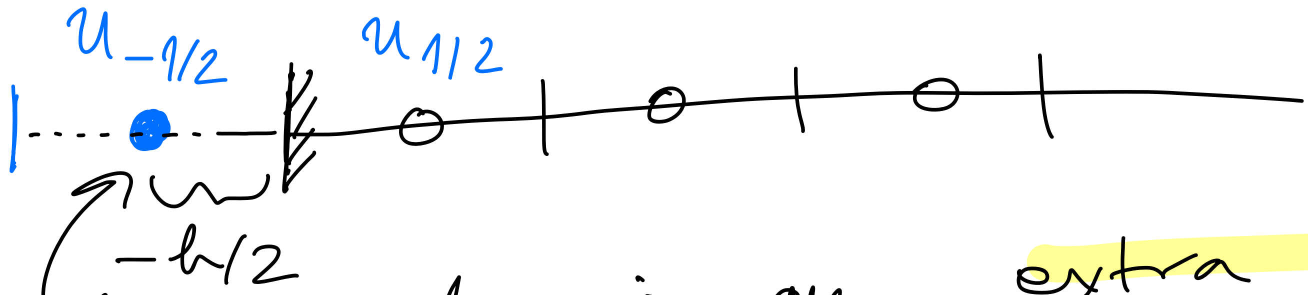
Sometimes we may use a staggered grid in our method, where the first grid point is $h/2$ away from the boundary. This is also called a cell-centered or a finite volume grid:



It makes sense to number the grid points with half-integers

One can use a single ghost cell to enforce BCs to 2nd order:

Neumman BC $\quad u'(x=0)=\sigma$



ghost value is an extra unknown

$$u'(0) \simeq \frac{u_{1/2}-u_{-1/2}}{h} = \sigma \leftarrow \text{extra equation}$$

Dirichlet $\quad u(0)=u_0$

$$u(0) \simeq \frac{u_{1/2}+u_{-1/2}}{2} = u_0 + O(h^2)$$

How to implement?

We need two pieces:

a) Discretize forward operator $\mathcal{L}$ so that you can compute

$$Lu \approx \mathcal{L} u$$

b) Solve linear system efficiently to discretize inverse operator

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

Ⓐ

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, hom) result (Lu)
    real, dimension (:), intent (in) :: u
    real, dimension ( size (u)) :: Lu
    logical, intent (in) :: hom ! homogeneous BCs ?
!--------
  real, dimension (0 : size (u)+1) :: u_ext
  integer :: N
```

Ⓑ

```
N = size(u)
u_ext(1:N) = u     ! copy input
call FillGhost(u_ext)
     implement homogeneous or inhomogeneous BCs
                                (physical BCs)

! Now do finite difference:

Lu(1:N) = (u_ext(0:N-1) + u_ext(2:N)
                 - 2 * u_ext(1:N))
! can be optimized / vectorized
end function ApplyOp
```

ⓒ

```
subroutine FillGhost (u_ext, hom)
real, dimension(0:), intent(inout) :: u_ext
 logical, intent(in) :: hom
  integer :: N
  N = size(u_ext) - 2   ! two ghost
                            values
if (periodic) then
  u_ext(0) = u_ext(N)
  u_ext(N+1) = u_ext(1)
else ! Dirichlet BCs
   if (hom) then
      u_ext(0) = 0 ; u_ext(N+1) = 0
   else
      u_ext(0) = α ; u_ext(N+1) = β
   end if ; end if
```

Ⓓ

Inverse Operator

We need to solve $Ax = f$ efficiently for very large A

(e.g. $256^3 \simeq 17K$ DOFs, so

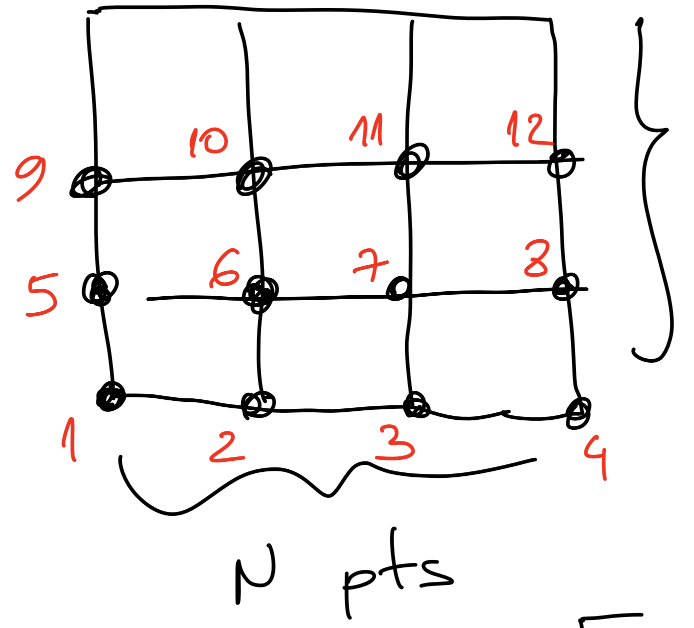
$$A = [17K \times 17K] \text{ matrix}$$

Options:

1) Use a direct sparse solver
   Does this work in 1D? [Discuss]
   Does a direct solver "work" in 2D/3D?

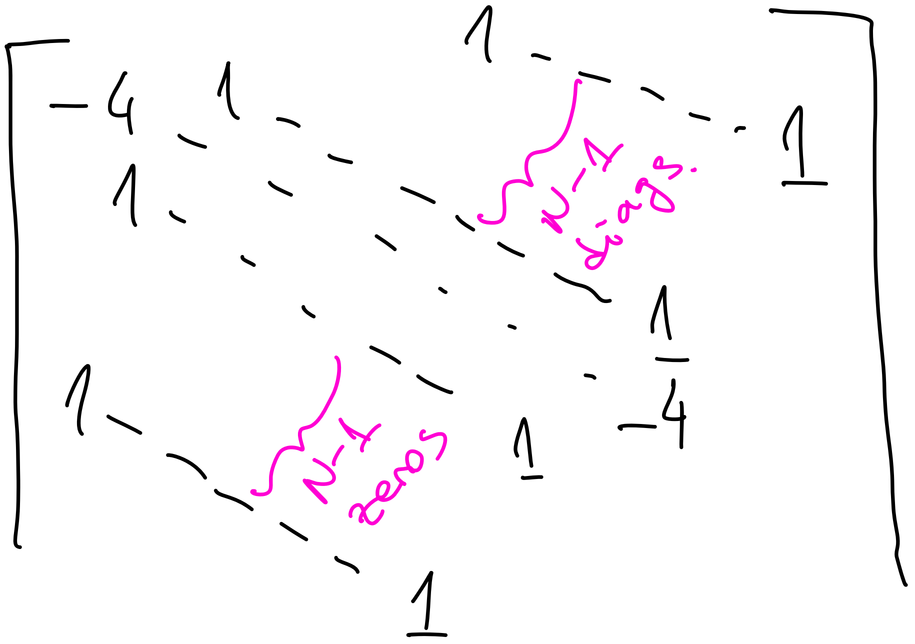
2) Use an iterative solver:
   How fast does it converge?
   How can we precondition? Ⓔ

Standard 5pt Laplacian in 2D:



← standard numbering (why is it arbitrary?)

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



Ⓕ

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

This is <u>not</u> linear. Can we do (log) linear solve?

<u>Theorem</u> (George): <u>Any</u> direct solver for $D^2_{2D,5}\, u = f$ requires at least $O(N_{pts}^{3/2})$ FLOPs and memory

Ⓖ

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

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

Same works for $D^2_{2D,9}$ or D^2_4 or any FD! But the best is still spectral solver $\hat{u} = -\frac{1}{k^2}\hat{f}$

Ⓗ

How about an iterative solver,

say conjugate gradients (why?)

Convergence of PCG:

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

K_A = L_2 conditioning number of A

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

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

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

Ⓘ

So to get better than $O(N_{pts}^{3/2})$ 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.

☺

Stability of FD methods

(for elliptic PDEs)

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

True PDE solution:

$$\hat{u} = [\underset{\uparrow}{u(x_1)}, \ldots, u(x_N)]$$

pointwise solution

Global error $\quad E = u - \hat{u}$

Ⓘ

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

Local (truncation) error

$$\tau = A\hat{u} - F \qquad \text{(LTE)}$$

can be estimated by simple Taylor series for smooth solutions.

e.g. $\|\tau\| = O(h^2)$

$$\begin{cases} A\hat{u} = F + \tau \\ A\,u = F \end{cases} \Rightarrow AE = -\tau$$

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

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

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

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

$$E_i \approx e(x_i)$$

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

Is this true?

III

$$E = A^{-1}\tau \Rightarrow$$

$$\underset{\substack{\| \\ O(h^2)}}{\|E\|} \le \underset{\substack{\downarrow \\ O(1)}}{\|A^{-1}\|}\ \underset{\substack{\| \\ O(h^2)}}{\|\tau\|}$$

A method to solve a linear BVP is stable if

$$\underbrace{\|A^{-1}\| \le C}_{\text{“uniform” ellipticity of PDE is preserved by discretization}} \quad \forall\, \underset{\substack{\uparrow \\ \text{grid spacing}}}{h} < h_0$$

$$\boxed{\text{stability + consistency} \Rightarrow \text{convergence}}$$

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

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

Let's start with L_2.

Recall $$\|\underset{\text{symmetric}}{A}\|_2 = \underset{\text{spectral radius}}{\rho}(A) = \max_p |\lambda_p|$$

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

V

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

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

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

Based on continuum eigenfunctions, guess that eigenvectors are:

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

VI

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

$$\lambda_p = \frac{2}{h^2}\left(\cos(p\pi h) - 1\right)$$

$$= \underbrace{-\pi^2 p^2}_{\text{continuum eigenvalue}} + \underbrace{\frac{1}{12}\pi^4 p^4 h^2}_{\text{error} = O(h^2)} + O(h^4)$$

Smallest eigenvalue = smallest wave vector (largest wavelength)

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

VII

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

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

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

Convergence in L_∞

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

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

But turns out this is too pessimistic and, in fact,

$$\|E\|_\infty = O(h^2) \text{ as well.}$$

Remember $\|A\|_\infty$ is the largest absolute <u>column</u> 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

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

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

This looks like a discretization of

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

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

⊗

This means the columns of A^{-1} are discrete Green's functions of the elliptic PDE. Specifically

$\tilde{G}^{(j)}$ tells us how the LTE at node/point j spreads to the other points.

For the Poisson eq. with Dirichlet,

$$\tilde{G}_i^{(j)} = hG(x_i ; x_j) \qquad (\text{exactly!})$$

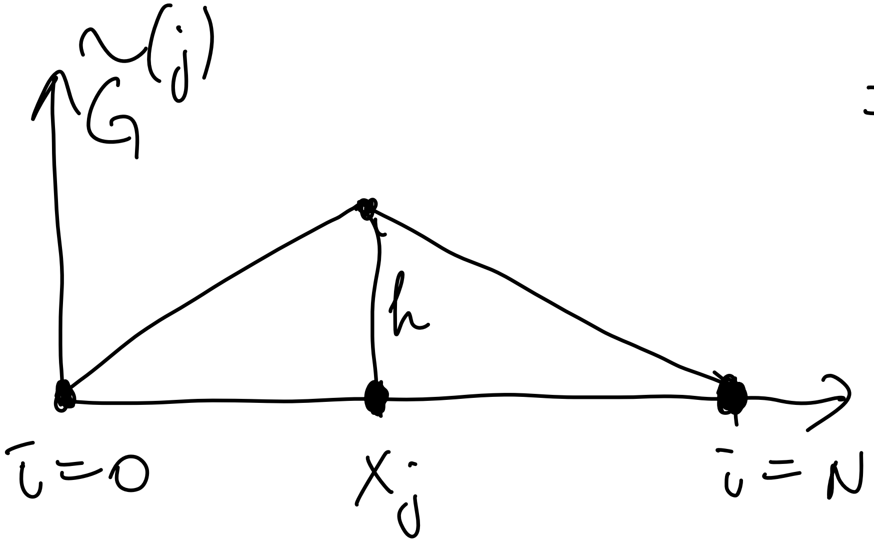
$$= h \begin{cases} (x_j - 1)\, x_i , & i = 1, \dots, j \\ (x_i - 1)\, x_j , & i = j, j+1, \dots, N \end{cases}$$

(XI)

$$\tilde{G}^{(j)}_i = h\, G(x_i, x_j) = A^{-1}_{ij}$$

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

$$\leq N \cdot h = L = 1$$



$\Rightarrow$ Method is also $O(h^2)$ in L_∞

(and in L_1)

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

$$|\tau_j| = O(h^q), \quad q < p$$

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

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

$$= O(h^{q+1})$$

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 poluting 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 the r.h.s. of $\tilde{G}^{(j)}$ by making the PDE be a "δ function", which is useful in 2D or for complicated PDEs ⊗