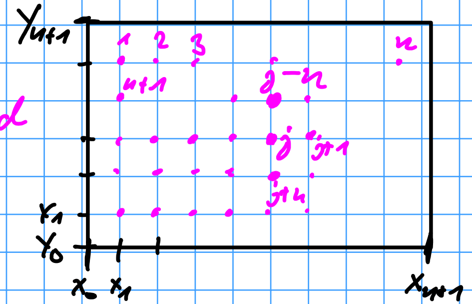
(10.14) Poisson equation in 2D on unit square

$$\Delta u = f \,, \qquad u|_{\partial\Omega} = 0$$

Ω_h discrete grid



start with Ω unit square

use geometry of Ω

discretize Δ:

use 1D stencil in every direction

$$\frac{1}{h^2}\begin{bmatrix} & -1 & \\ -1 & 2+2 & -1 \\ & -1 & \end{bmatrix} = \frac{1}{h^2}\begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}$$

5-point-stencil

\# unknowns $= n \cdot n = n^2$

vector of unknowns $u_h = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n^2} \end{pmatrix}$

apply stencil at every unknown grid point

$\Rightarrow$ n^2 equations

jth equation:

$$\frac{1}{h^2}\left(0 \cdots 0 \underset{(j-n)\text{th col.}}{-1} 0 \cdots 0 \; -1 \; \underset{j\text{th col.}}{4} \; \overset{j+1}{-1} \; 0 \cdots 0 \underset{(j+n)\text{th col}}{-1} \; 0 \; 0\right) \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n^2} \end{pmatrix} = f_j$$

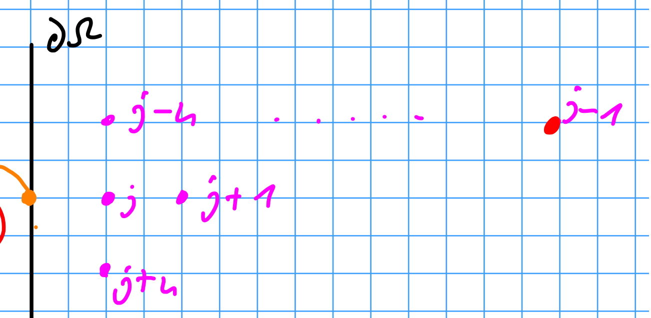
if this is a grid point that has 4 neighbors that are unknowns!
(inner grid point)

if neighbors are at the boundary then the value of this neighbor is known.
This values go to the right-hand side of the equation.

e.g. no left neighbor

$$\frac{1}{h^2}\left(-1 \;\cdots\; 0 \;\; 4 \;\; -1 \;\cdots\; -1\right) u_h = f_j + \frac{1}{h^2}\underbrace{u(\downarrow)}_{=0}$$

because of $u|_{\partial\Omega} = 0$



the whole system ($n^2 \times n^2$ system):

distance n

$$\frac{1}{h^2}\begin{pmatrix} \underbrace{\begin{bmatrix} 4 & -1 & & \\ -1 & 4 & -1 & \\ & \ddots & \ddots & -1 \\ & & -1 & 4 \end{bmatrix}}_{M,\ n\times n} & \begin{bmatrix} -1 & & 0 \\ & \ddots & \\ 0 & & -1 \end{bmatrix}_{-I} & & \\ \begin{bmatrix} -1 & & 0 \\ & \ddots & \\ 0 & & -1 \end{bmatrix}_{-I} & \begin{bmatrix} 4 & -1 & & \\ -1 & 4 & -1 & \\ & \ddots & \ddots & -1 \\ & & -1 & 4 \end{bmatrix}_{M} & \begin{bmatrix} -1 & & \\ & \ddots & \\ & & -1 \end{bmatrix}_{-I} & \\ & & \ddots & \\ & & \begin{bmatrix} -1 & & 0 \\ & \ddots & \\ & & -1 \end{bmatrix}_{-I} & \begin{bmatrix} 4 & -1 & & \\ -1 & 4 & -1 & \\ & \ddots & \ddots & -1 \\ & & -1 & 4 \end{bmatrix}_{M} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n^2} \end{pmatrix} = \begin{pmatrix} f_1 \\ \vdots \\ \vdots \\ f_{n^2} \end{pmatrix} \begin{array}{l} +0_l + 0_t \\ +0_t \\ \vdots \\ +0_t \\ +0_t + 0_r \end{array}$$

$$\underset{(n^2\times n^2)}{T_h}\ \underset{(n^2\times 1)}{u_h} = f_h^{\,(n^2\times 1)}$$

10.15 Compact writing: Kronecker products

1D case: $\begin{pmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & -1 & \ddots & \ddots \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix} = A_n^{(1)}$

$n = 3 \rightsquigarrow u_1, \dots, u_9$

1D discrete Laplace + $2I$

$$\underbrace{\frac{1}{h^2}\left(\begin{array}{ccc|ccc|ccc} 4 & -1 & & -1 & & & & & \\ -1 & 4 & -1 & & -1 & & & & \\ & -1 & 4 & & & -1 & & & \\ \hline -1 & & & 4 & -1 & & -1 & & \\ & -1 & & -1 & 4 & -1 & & -1 & \\ & & -1 & & -1 & 4 & & & -1 \\ \hline & & & -1 & & & 4 & -1 & \\ & & & & -1 & & -1 & 4 & -1 \\ & & & & & -1 & & -1 & 4 \end{array}\right)}_{A_n^{(2)}} = \frac{1}{h^2}\left(\begin{array}{c|c|c} A_n^{(1)} + 2I & -I & \\ \hline -I & A_n^{(1)} + 2I & -I \\ \hline & -I & A_n^{(1)} + 2I \end{array}\right)$$

$$= \frac{1}{h^2}\left\{\left(\begin{array}{c|c|c} A_n^{(1)} & & \\ \hline & A_n^{(1)} & \\ \hline & & A_n^{(1)} \end{array}\right) + \underbrace{\left(\begin{array}{c|c|c} 2I & -I & \\ \hline -I & 2I & -I \\ \hline & -I & 2I \end{array}\right)}_{\text{looks like } A_n^{(1)} \text{ with all numbers replaced by mult. of } I}\right\}$$

Kronecker Tensor product of two matrices A, B:

$$A \otimes B = \begin{pmatrix} a_{11} B & \cdots & a_{1n} B \\ \vdots & & \vdots \\ a_{m1} B & \cdots & a_{mn} B \end{pmatrix}$$

in Matlab

kron(A, B)

works efficiently for sparse matrices

$$A_n^{(2)} = \underbrace{I \otimes A_n^{(1)}}_{\begin{pmatrix} 1 \cdot A_n^{(1)} & & \\ & 1 \cdot A_n^{(1)} & \\ & & \ddots \\ & & & 1 \cdot A_n^{(1)} \end{pmatrix}} + \underbrace{A_n^{(1)} \otimes I}_{\begin{pmatrix} 2 \cdot I & -1 \cdot I & & \\ -1 \cdot I & 2 \cdot I & \ddots & \\ & \ddots & \ddots & -1 \cdot I \\ & & -1 \cdot I & 2 \cdot I \end{pmatrix}}$$

in Matlab:

A2 = kron(speye(n), A1)

+ kron(A1, speye(n))

speye(n) sparse identity matrix

Compare

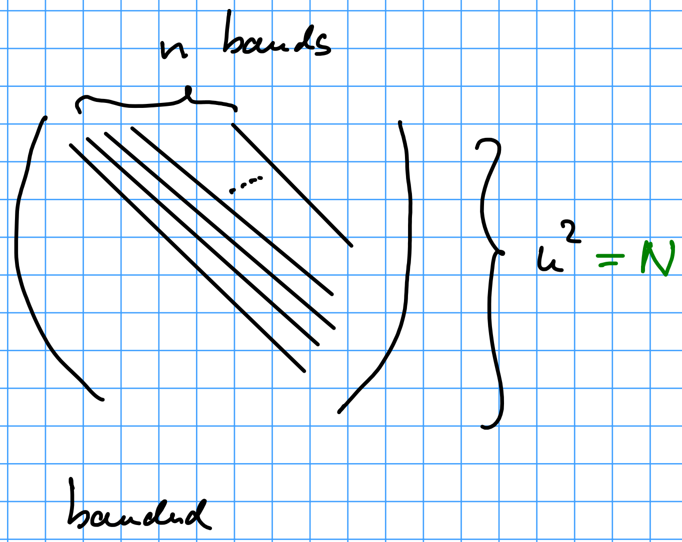
$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

in 2D

(10.16) Complexity of Cholesky factorization

observation: fill-in between inner and outer diagonals:

$R = \mathrm{chol}(A2)$ has the structure



full Cholesky would need $O(N^3) = O(n^6)$

banded Cholesky needs $O(N^2)$

$$N^2 = \underbrace{n^2}_{\#\text{rows}} \cdot \underbrace{n^2}_{\text{bandwidth}^2}$$

computer experiments: $n = 1000$ CPU-time $\approx 15\,s$

$n = 2000$ CPU-time $\approx 240\,s$

ultimate goal would be $O(N)$ computing time

$O(N)$ is possible with multigrid methods

but $O(N \log N)$ is possible for Poisson using FFT $\longrightarrow$ next week