Matrix vector multiplication & diffusion

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

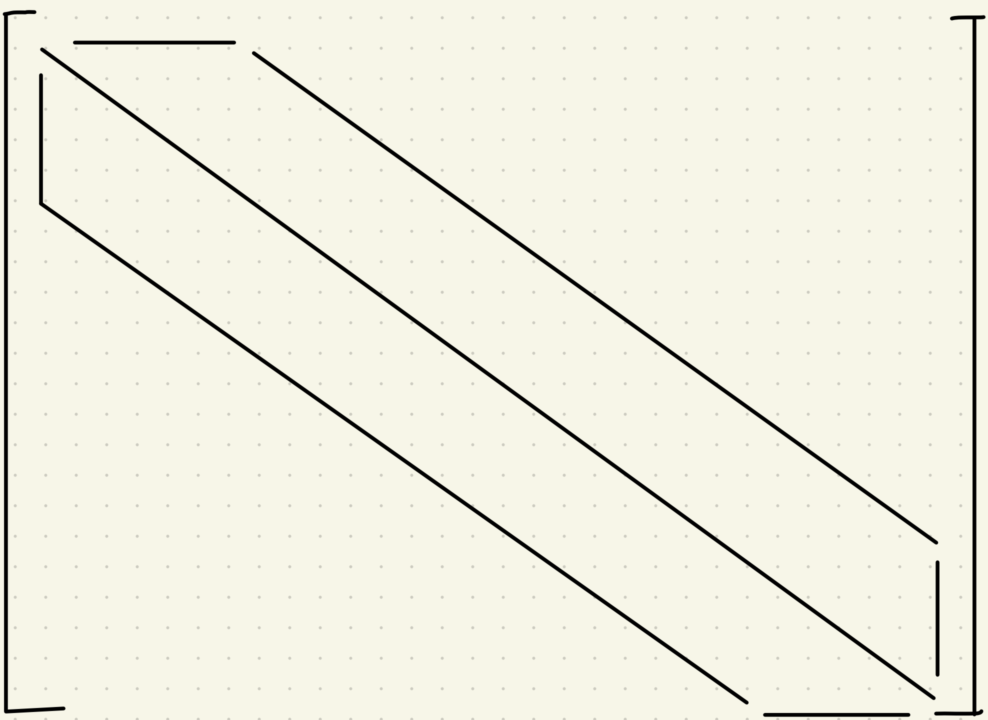
$\Big\downarrow$ finite difference

$$u_{ij}^{n+1} = u_{ij}^{n} + \frac{\Delta t}{h^2}\left(u_{i+1,j}^{n} + u_{i-1,j}^{n} + u_{i,j+1}^{n} + u_{i,j-1}^{n} - 4u_{ij}^{n}\right)$$

$\Big\downarrow$ note that we can write this as Matrix vector multiplication

$$\underbrace{\begin{bmatrix} u_{0,0}^{n+1} \\ \vdots \\ u_{0,N-1}^{n+1} \\ u_{1,0}^{n+1} \\ \vdots \\ u_{1,N-1}^{n+1} \\ \vdots \\ u_{N-1,0}^{n+1} \\ \vdots \\ u_{N-1,N-1}^{n+1} \end{bmatrix}}_{\vec{u}^{n+1}} = \underbrace{\begin{bmatrix} u_{0,0}^{n} \\ \vdots \\ u_{0,N-1}^{n} \\ u_{1,0}^{n} \\ \vdots \\ u_{1,N-1}^{n} \\ \vdots \\ u_{N-1,0}^{n} \\ \vdots \\ u_{N-1,N-1}^{n} \end{bmatrix}}_{\vec{u}^{n}} + \; \underbrace{\cdots}_{\frac{\Delta t}{h^2} A \vec{u}^{n}}$$

$A \in \mathbb{R}^{n^2 \times n^2}$



So A is a huge matrix if N is big but it only has 5 non-zero entries for each row $\longrightarrow$ So A is a sparse matrix!

there are efficient ways to store sparse matricies

- for example c-compressed-row storage (CSR-format)
  - stores all non zero entries
  - row indicies where non-zero starts

Question 1: CSR format

Write down the representation:

$$A[k] = (2, -1, -1, -1, 2, -1, -2, 4, -2, -1, -1, 2)^T$$

$$J[k] = (0, 1, 3, 0, 1, 2, 1, 2, 3, 0, 2, 3)^T$$

$$K[i] = (0, 3, 6, 9, 12)^T$$

Compute the matrix product Au

$$Au = \begin{array}{cc} & j \\ i & \begin{bmatrix} (\quad\quad) \\ \\ \\ \end{bmatrix} \end{array} \begin{bmatrix} \\ \\ \\ \end{bmatrix} = \begin{bmatrix} \boxed{} \\ \\ \\ \end{bmatrix} =: v$$

$$v_i = \sum_{j=0}^{N} A_{ij} u_j = \langle A_i, u \rangle$$

i-th row of matrix

With the CSR-format we can compute the product as

$$v_i = \langle A[K[i] : K[i+1]], u[J[K[i] : K[i+1]]] \rangle$$

$$\left.\begin{aligned}
v_0 &= \begin{pmatrix} 2 \\ -1 \\ -1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 3 \end{pmatrix} = -4 \\
v_1 &= \begin{pmatrix} -1 \\ 2 \\ -1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix} = 0 \\
v_2 &= \begin{pmatrix} -2 \\ 4 \\ -2 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = 0 \\
v_3 &= \begin{pmatrix} -1 \\ -1 \\ 2 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 2 \\ 3 \end{pmatrix} = 4
\end{aligned}\right\} Au = \begin{pmatrix} -4 \\ 0 \\ 0 \\ 4 \end{pmatrix}$$

Question 2: Matrix-vector product with MPI

a) derive A s.t. $u^{n+1} = u^n + A u^n$

$$u^{n+1}_{ij} = u^n_{ij} + \frac{\Delta t}{h^2}\left(u^n_{i+1,j} + u^n_{i-1,j} + u^n_{i,j+1} + u^n_{i,j-1} - 4u^n_{ij}\right)$$

note that we can write this as Matrix vector multiplication

$$\underbrace{\begin{bmatrix} u^{n+1}_{0,0} \\ \vdots \\ u^{n+1}_{0,N_j} \\ u^{n+1}_{1,0} \\ \vdots \\ u^{n+1}_{1,N_j} \\ \vdots \\ u^{n+1}_{N_i-1,0} \\ \vdots \\ u^{n+1}_{N_i-1,N_j} \end{bmatrix}}_{\vec{u}^{n+1}} = \underbrace{\begin{bmatrix} u^{n}_{0,0} \\ \vdots \\ u^{n}_{0,N_j} \\ u^{n}_{1,0} \\ \vdots \\ u^{n}_{1,N_j} \\ \vdots \\ u^{n}_{N_i-1,0} \\ \vdots \\ u^{n}_{N_i-1,N_j} \end{bmatrix}}_{\vec{u}^{n}} + \frac{\Delta t}{h^2} \underbrace{\begin{array}{c} \begin{array}{cccc} (0,0)\cdots(0,N_j) & (1,0)\cdots(1,N_j) & \cdots & (N_i-1,0)\cdots(N_i-1,N_j) \end{array} \\ \begin{bmatrix} \begin{matrix} -4 & 1 & 0 & \cdots & 0 & 1 \\ 1 & -4 & 1 & 0 & \cdots & 0 \\ & & \ddots & \ddots & & \\ & & & 1 & -4 & 1 \\ 1 & 0 & \cdots & & 1 & -4 \end{matrix} & \begin{matrix} 1 & 0 & \cdots \\ 0 & 1 & 0 \cdots 0 \\ & \ddots & \\ & 1 & 0 \\ 0 & \cdots & 0 \; 1 \end{matrix} & \begin{matrix} 0 \\ 0 \\ \\ \\ 0 \end{matrix} & \begin{matrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & 0 \cdots & 0 \\ & & \ddots & \\ & & 1 & 0 \\ 0 & \cdots & 0 & 1 \end{matrix} \\ \vdots & & & \\ \vdots & & & \\ \vdots & & & \end{bmatrix} \end{array}}_{A} \begin{bmatrix} u^{n}_{0,0} \\ \vdots \\ u^{n}_{0,N_j} \\ u^{n}_{1,0} \\ \vdots \\ u^{n}_{1,N_j} \\ \vdots \\ u^{n}_{N_i-1,0} \\ \vdots \\ u^{n}_{N_i-1,N_j} \end{bmatrix}$$

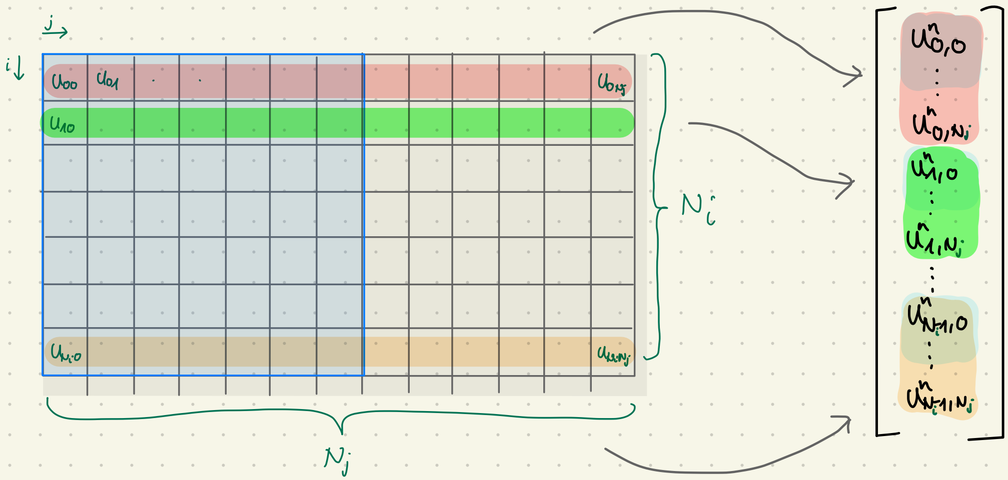
$$\vec{u}^{n+1} = \vec{u}^n + \frac{\Delta t}{h^2} A \vec{u}^n$$

$$X := \underbrace{\begin{bmatrix} -4 & 1 & 0 & \cdots & 0 & 1 \\ 1 & & & & & 0 \\ 0 & & & & & \vdots \\ \vdots & & & & & 0 \\ 0 & & & & & 1 \\ 1 & 0 & \cdots & 0 & 1 & -4 \end{bmatrix}}_{N_j} \Bigg\} N_j \qquad 1 := \underbrace{\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \vdots \\ \vdots & & & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix}}_{N_j} \Bigg\} N_j \qquad 0 := \underbrace{\begin{bmatrix} 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{bmatrix}}_{N_j} \Bigg\} N_j$$

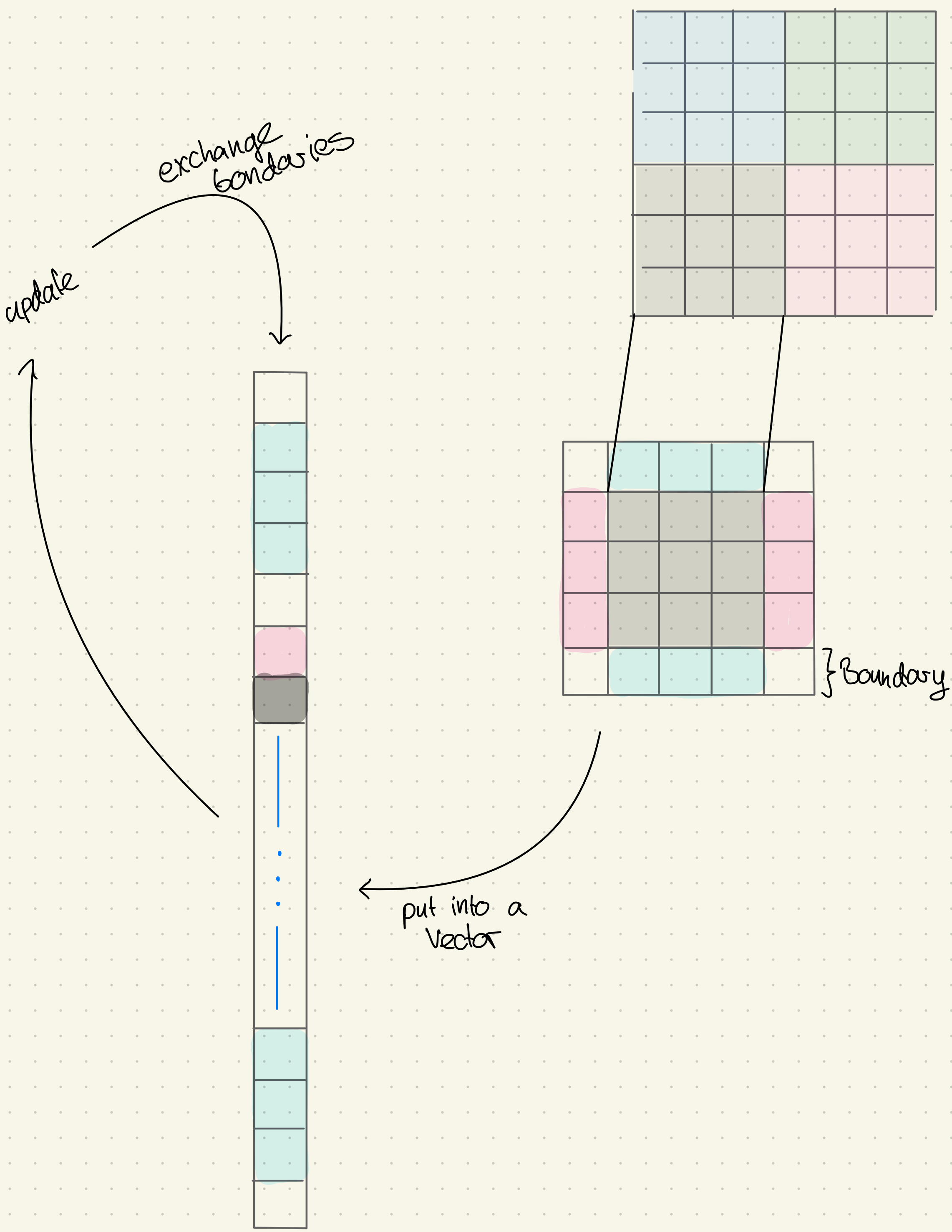
$$A = \underbrace{\begin{bmatrix} x & 1 & 0 & \cdots & \cdots & 0 & 1 \\ 1 & x & & & & & 0 \\ 0 & & \ddots & & & & \vdots \\ \vdots & & & \ddots & & & \vdots \\ \vdots & & & & \ddots & & 0 \\ 0 & & & & & & 1 \\ 1 & 0 & \cdots & \cdots & 0 & 1 & x \end{bmatrix}}_{N_i-1} \left.\vphantom{\begin{bmatrix} x \\ 1 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix}}\right\} N_i-1$$

Question 2

b)



We first split the grid and then exchage boundaries as before & then update each cell with matrix multiplication!



exchange boundaries
update
Boundary
put into a Vector