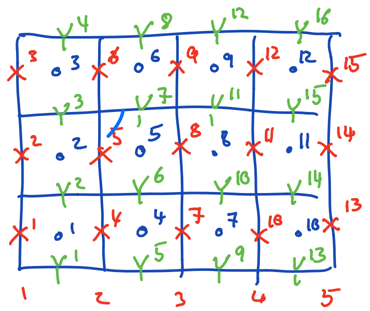
Lecture 20: Discrete operators in 2D

Logistics: - HW 7 is due Thursday

Last time: - 2D discretization

- Matlab basics: - meshgrid / - reshape $\Big\}$ y-first

- 2D staggered grid



$$q = \begin{bmatrix} q_x \\ q_y \end{bmatrix}$$

$$\begin{bmatrix} dh_x \\ dh_y \end{bmatrix} = \underline{dh} = \underline{\underline{G}}\,\underline{h}$$

Discrete gradient: $\underline{\underline{G}} = \begin{array}{c} \begin{bmatrix} \underline{\underline{G_x}} \\ \underline{\underline{G_y}} \end{bmatrix} \end{array}\begin{array}{c} 1 \\ 2 \end{array}$

Discrete divergence: $\underline{\underline{D}} = \begin{array}{cc} 3 & 4 \\ \left[\underline{\underline{D_x}} \right. & \left. \underline{\underline{D_y}} \right] \end{array}$

$$\underline{\underline{D_x}}\, q_x + \underline{\underline{D_y}}\, q_y$$

Assembly of $\underline{\underline{D_y}}$:

$$\underline{\underline{D_y}}^{(2)} = \begin{bmatrix} \underline{\underline{D_y}}^{(1)} & & & \\ & 1\cdot\underline{\underline{D_y}}^{1} & & \\ & & \underline{\underline{D_y}}^{1} & \\ 0\cdot\underline{\underline{D_y}}^{1} & & & \ddots\ \underline{\underline{D_y}}^{1} \end{bmatrix}$$

Nx by Nx block matrix

$\Rightarrow$ Kronecker products

$$D_y^2 = \left(\underline{\underline{I_x}} \otimes \underline{\underline{D_y^1}} \right)$$

coeff — replicate

Matlab : $\underline{\underline{D_y}} = \text{kron}(\underline{\underline{I_x}}, \underline{\underline{D_y}});$

Today: - Assembly of $\underline{\underline{D_x}}$

- Getting $\underline{\underline{G}}$ from $\underline{\underline{D}}$

- Transition from 1D to 2D

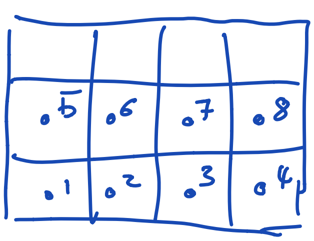
Aside: outer product | Kronecker product

$$\underline{a} \otimes \underline{b} = \begin{bmatrix} a_1 b_1 & a_1 b_2 \\ a_2 b_1 & a_2 b_2 \end{bmatrix}$$

$$\underline{\underline{A}} \otimes \underline{\underline{B}} \qquad A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

$$\begin{bmatrix} a_{11}\underline{\underline{B}} & a_{12}\underline{\underline{B}} \\ a_{21}\underline{\underline{B}} & a_{22}\underline{\underline{B}} \end{bmatrix}$$

How do we build $\underline{\underline{D_x}}$?



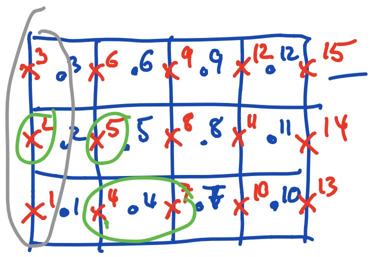
on x-first grid

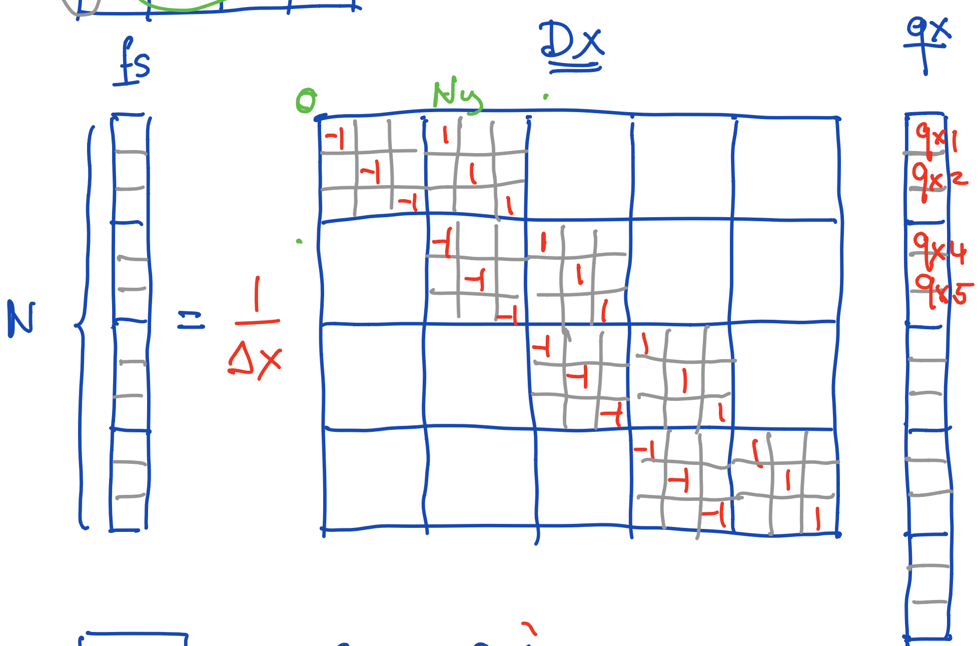
$$\underline{\underline{D_x^2}} = \text{kron}(\underline{\underline{I_y}}, \underline{\underline{D_x}})$$

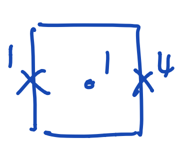
But we can't switch ordering

how do we build $\underline{\underline{D_x}}$ on y-first grid?

$\underline{D}\, q = \underline{fs}$







$$\frac{qx_4 - qx_1}{\Delta x}$$

$\underline{Dx^2}$ is sparse diagonal matrix

$\Rightarrow$ assemble with $\underline{\text{spdiags}}$

$\underline{Dx^2}$ is also a block diagonal matrix

built from Ny by Ny identities

$$\underline{\underline{Dx}}^2 = \begin{bmatrix} -\underline{\underline{I}}_y & \underline{\underline{I}}_y & & & \\ & -\underline{\underline{I}}_y & \underline{\underline{I}}_y & & \\ & & -\underline{\underline{I}}_y & \underline{\underline{I}}_y & \\ & & & -\underline{\underline{I}}_y & \underline{\underline{I}}_y \end{bmatrix} = \underline{\underline{Dx}}^1 \otimes \underline{\underline{I}}_y$$

$$\underline{\underline{Dx}}^1 = \frac{1}{\delta x} \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & -1 & 1 & \\ & & & -1 & 1 \end{bmatrix}$$

In Matlab: $\underbrace{\underline{\underline{Dx}}}_{2D} = \text{kron}(\underbrace{\underline{\underline{Dx}}}_{1D}, \underline{\underline{I}}_y);$

In summary:

$$\boxed{\begin{aligned} \underline{\underline{Dy}} &= \text{kron}(\underline{\underline{I}}_x, \underline{\underline{Dy}}) \\ \underline{\underline{Dx}} &= \text{kron}(\underline{\underline{Dx}}, \underline{\underline{I}}_y) \end{aligned}}$$

$\underline{\underline{I}}_x$ is Nx by Nx identity

$\underline{\underline{I}}_y$ is Ny by Ny identity

$$\boxed{\underline{\underline{D}} = [\underline{\underline{Dx}}; \underline{\underline{Dy}}]}$$

Discrete gradient

The $\underline{\underline{G}}_x$ and $\underline{\underline{G}}_y$ matrices could be built using 1D operators and Kronecker products.

Instead, we use the fact that $\underline{\underline{D}}$ and $\underline{\underline{G}}$ are adjoints

$$\boxed{\underline{\underline{G}} = -\underline{\underline{D}}^T} \qquad \text{true in interior}$$

Still need to impose the Natural BC's (hom. Neu).

$\Rightarrow$ Set $\underline{\underline{G}} = 0$ on all bnd faces.

Make vector containing all bnd faces:

dof_f_bnd = [dof_f_xmin; dof_f_xmax;
dof_f_ymin; dof_f_ymax];

Zero out corresponding rows

G(dof_f_bnd) = 0;

In non-Cartesian coordinates $\underline{\underline{G}} \neq -\underline{\underline{D}}^T$!