# Lecture 11: Discretization in 2D

Logistics: - HW 4 is due 10/12

$\Rightarrow$ come see me in office hrs

- HW5 will be posted $\to$ radial coord.

Last time: - Flux computation

interior: $\underline{q} = -\underline{\underline{Kd}}\,\underline{\underline{G}}\,\underline{h}$ direct

boundary: $q = \pm\, r\,\frac{V}{A}$ reconstruct from residual

$\uparrow$ sign change on xmax bnd

- radial coordinate systems (1D)
  - gradient remains the same
  - divergence : $\nabla \cdot = \boxed{\frac{1}{r^{d-1}}\frac{d}{dr} r^{(d-1)}}$ d = dim + 1

$\Rightarrow \boxed{\underline{\underline{D}} = \underline{\underline{R}}_{inv} * \underline{\underline{D}} * \underline{\underline{R}}}$

cyl./spheri. linear

$\underline{\underline{R}}_{inv}$ diagonal (cell centers)

$\underline{\underline{R}}$ diagonal (faces)

Today: Transition to 2D ?

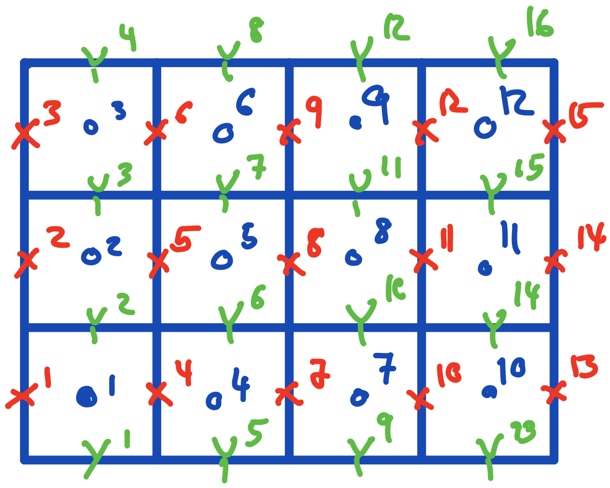
# Matlab basics

→ recognize inbuilt ordering in Matlab function

⇒ meshgrid is ordered y-first

## Staggered grid in 2D

$N_x = 4 \qquad N_y = 3 \qquad N = N_x N_y = 12$



x-faces: $N_{fx} = (N_x+1) N_y = 15$

y-faces: $N_{fy} = N_x (N_y+1) = 16$

Total faces: $N_f = N_{fx} + N_{fy}$

Number dof in y-dir first

## Discrete gradient

continuous gradient: $\nabla h = \begin{pmatrix} \frac{\partial h}{\partial x} \\ \frac{\partial h}{\partial y} \end{pmatrix}$

approximate $\frac{\partial h}{\partial x} \sim \underline{dhx}$ on x-faces

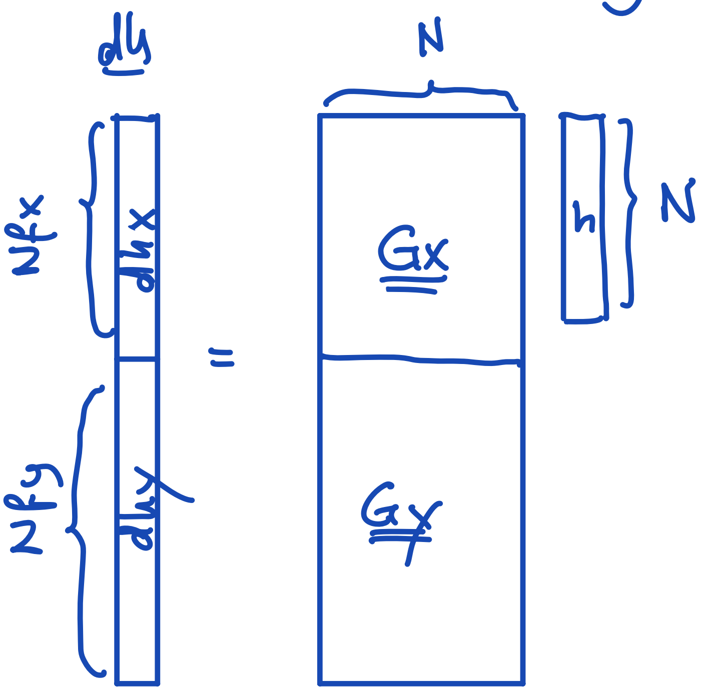
$\frac{\partial h}{\partial y} \sim \underline{dhy}$ on y-faces

We choose to build $\underline{\underline{G}}$ such that the resulting gradient vector $\underline{dh}$ is orderd as follows: $\underline{dh} = \begin{bmatrix} \underline{dhx} \\ \underline{dhy} \end{bmatrix}$

$\Rightarrow$ 2D Gradient can be decomposed as

$$\underline{\underline{G}} = \begin{bmatrix} \underline{\underline{Gx}} \\ \underline{\underline{Gy}} \end{bmatrix} \qquad \underline{dhx} = \underline{\underline{Gx}}\, \underline{h}$$

$$\underline{dhy} = \underline{\underline{Gy}}\, \underline{h}$$




$\underline{\underline{G}}$ is Nf by N

$\underline{\underline{Gx}}$ is Nfx by N

$\underline{\underline{Gy}}$ is Nfy by N

## Discrete divergence

$$\nabla \cdot \underline{q} = \frac{\partial qx}{\partial x} + \frac{\partial qy}{\partial y} \approx \underline{\underline{D}}\, \underline{q} = \underline{\underline{Dx}} * \underline{qx} + \underline{\underline{Dy}} * \underline{qy}$$

$$\underline{q} = \begin{bmatrix} qx \\ qy \end{bmatrix} \qquad \underline{\underline{D}} = [\underline{\underline{Dx}} \;\; \underline{\underline{Dy}}]$$

$$N\left\{\underline{f_s}\right. = \underbrace{\begin{bmatrix} \overbrace{\underline{\underline{D_x}}}^{N_{fx}} & \overbrace{\underline{\underline{D_y}}}^{N_{fy}} \end{bmatrix}}_{N_f} \begin{bmatrix} \underline{g_x} \\ \underline{g_y} \end{bmatrix} \begin{matrix} \}\,N_{fx} \\ \}\,N_{fy} \end{matrix}$$

matrix sizes:

$\underline{\underline{D}}$ is $N$ by $N_f$

$\underline{\underline{D_x}}$ is $N$ by $N_{fx}$

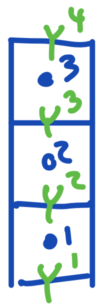
$\underline{\underline{D_y}}$ is $N$ by $N_{fy}$

Laplacian

$$\underset{N\cdot N}{\underline{\underline{L}}} = -\underset{N\cdot N_f}{\underline{\underline{D}}} * \underset{N_f\cdot N}{\underline{\underline{G}}} = -\underline{\underline{D_x}} * \underline{\underline{G_x}} + \underline{\underline{D_y}} * \underline{\underline{G_y}}$$

# Building the 2D discrete divergence matrix

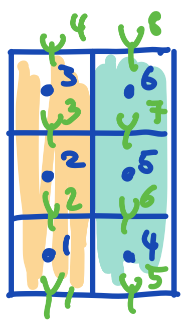
Start with $\underline{\underline{D_y}}$ in 1D :

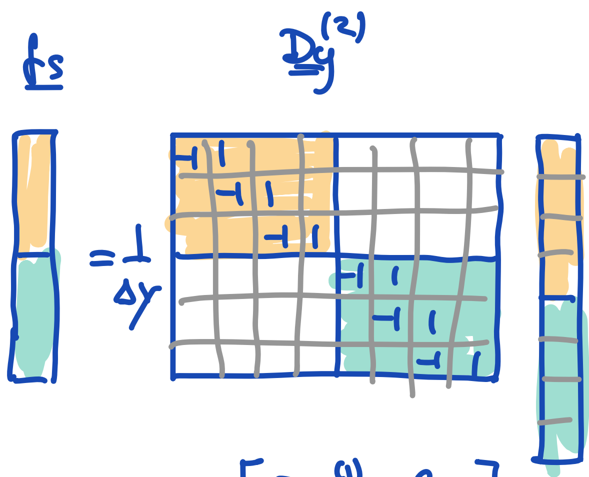



$$\underline{\underline{D_y}} = \frac{1}{\Delta y}\begin{bmatrix} -1 & 1 & & \\ & -1 & 1 & \\ & & -1 & 1 \end{bmatrix} \qquad \underline{q_y}$$

$N_y \qquad N_y+1$

Suppose we add a second column




$N_x = 2$

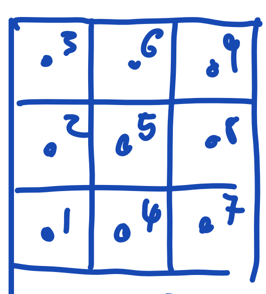
$N_y = 3$

$N_{fy} = 8$

$N = 6$

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

Suppose you add a third column




$N_x = 3$

$$\underline{\underline{D_y}}^{(3)} = \begin{bmatrix} \underline{\underline{D_y}}^{(2)} & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{D_y}}^{(1)} \end{bmatrix} = \begin{bmatrix} \underline{\underline{D_y}}^{(1)} & \underline{\underline{0}} & \underline{\underline{0}} \\ & \underline{\underline{D_y}}^{(1)} & \underline{\underline{0}} \\ & & \underline{\underline{D_y}}^{(1)} \end{bmatrix}$$

In general

$\underline{\underline{Dy}}^{2D}$ is a block matrix with $Nx$ by $Nx$ blocks of size $Ny$ b $(Ny+1)$. The diagonal blocks are $\underline{\underline{Dy}}^{1D}$ and all others are zero.

## Tensor product construction of $\underline{\underline{Dy}}^{2D}$

Note: not easily implemented with spdiags!

But $\underline{\underline{Dy}}^{2D}$ can be assembled with Kronecker or Tensor product.

## Definition:

If $\underline{A}$ is a $m \times n$ matrix and $\underline{B}$ is a $p \times q$ matrix, then the Kronecker product $\underline{A} \otimes \underline{B}$ is the following $mp \times nq$ matrix
block

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

Hence we can construct $\underline{\underline{D_y}}^{2D}$ as

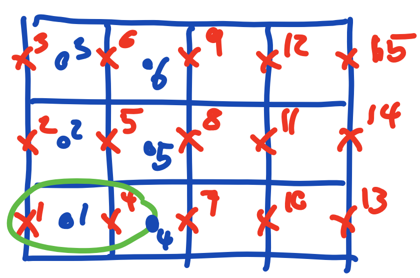
$$\boxed{\underline{\underline{D_y}}^{2D} = \underline{\underline{I_x}} \otimes \underline{\underline{D_y}}^{1D}} = \begin{bmatrix} \underline{\underline{D_y}}^{1D} & & & & \\ & \underline{\underline{D_y}}^{1D} & & & \\ & & \underline{\underline{D_y}}^{1D} & & \\ & & & \underline{\underline{D_y}}^{1D} & \\ & & & & \ddots \end{bmatrix}$$

where $\underline{\underline{I_x}}$ is $N_x$ by $N_x$ identity matrix

In Matlab the tensor product is obtained

$$\underset{\substack{\uparrow \\ \text{2D op}}}{\underline{\underline{D_y}}} = \text{kron}(\underline{\underline{I_x}}, \underset{\substack{\uparrow \\ \text{1D opr}}}{\underline{\underline{D_y}}})$$

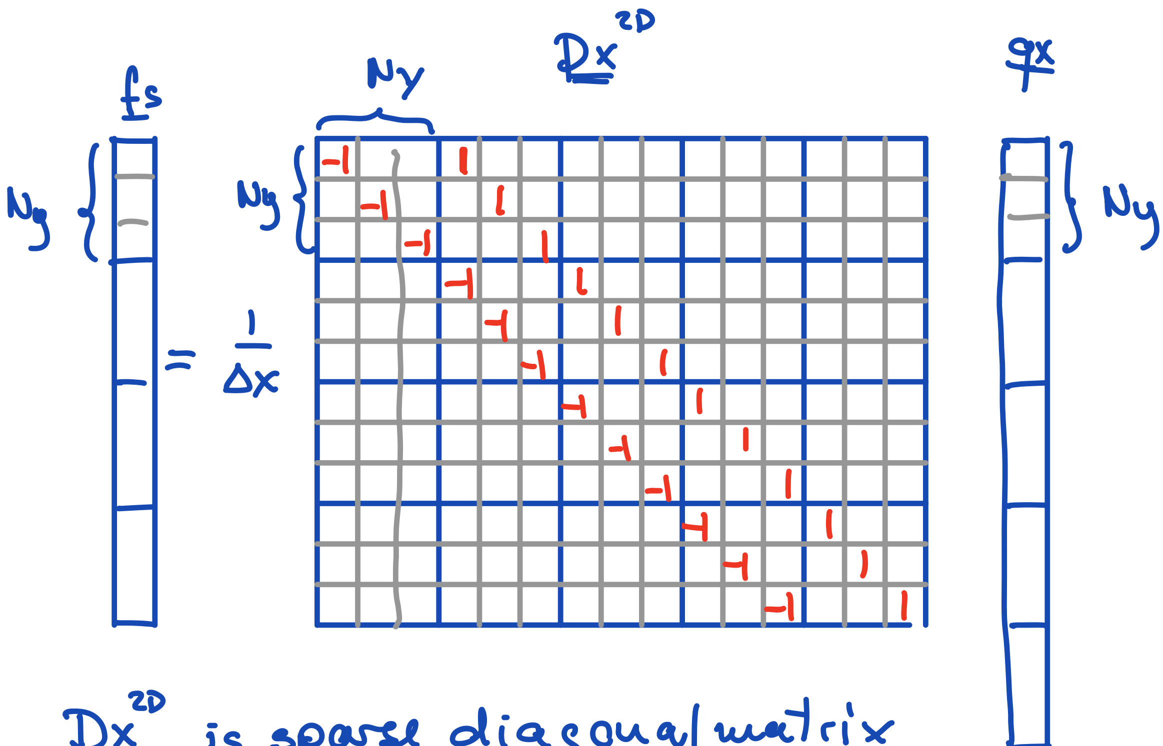
How do we build $\underline{\underline{D_x}}$




$N_x = 4 \quad N_y = 3$

$N = 12$

$Nfx = 15$




$\underline{\underline{Dx}}^{2D}$ is sparse diagonal matrix
(could be assembled with spdiags)

but $\underline{\underline{Dx}}^{2D}$ is also a block matrix
built from $N_y \cdot N_y$ blocks → identities

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

⇒ use Kronecker product

In Matlab: $\boxed{\underline{\underline{Dx}} = \mathrm{kron}(\underline{\underline{Dx}}, \underline{\underline{I_y}})}$

$$\boxed{\begin{aligned} \underline{\underline{D_y}} &= \mathrm{kron}\left(\underline{\underline{I_x}}, \underline{\underline{D_y}}\right) \\ \underline{\underline{D_x}} &= \mathrm{kron}\left(\underline{\underline{D_x}}, \underline{\underline{I_y}}\right) \end{aligned}}$$

$$\underline{\underline{D}} = \left[\underline{\underline{D_x}} \quad \underline{\underline{D_y}}\right]$$

Note : $\underline{\underline{G}} = -\underline{\underline{D}}^T$

zero out bouderies