Lecture 19: Discretization in 2D

Logistics: - HW 7 problem 1 posted more will be added

Last time: - Numerical solution for unconfined flow

general non-linear diffusion equ

$$s(u)\,\frac{\partial u}{\partial t} - \nabla\cdot\left[f(u)\,\nabla u\right] = f_s$$

$$s(u) = \phi_0\, u^m \qquad f(u) = \frac{k_0}{n+1}\, u^{n+1}$$

- Backward Euler

$$\underline{r} = \{\underline{s}(\underline{u}^{n+1})\}_c\,(\underline{u}^{n+1} - \underline{u}^{n}) - \Delta t\, \underline{\underline{D}}\left[\{\underline{\underline{M}}\,\underline{f}(\underline{u})\}_f\, \underline{\underline{G}}\, \underline{u}\right] - \Delta t\, f_s$$

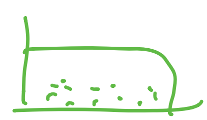
$$\underline{\underline{J}}(\underline{u},\underline{u}^n) = \underline{\underline{dS}}(\bar{\underline{u}})\,\underline{\underline{U}}(\underline{u},\underline{u}^n) + \underline{\underline{S}}(\underline{u}) + \Delta t\, \underline{\underline{D}}\left[\underline{\underline{G}}\,\underline{\underline{U}}(\bar{\underline{u}})\,\underline{\underline{M}}\,\underline{\underline{dF}}(\bar{\underline{u}}) + \underline{\underline{F}}(\underline{u})\,\underline{\underline{G}}\right]$$

Application to highland aquifer

- to match published crustal structures

$$m \in [2,3] \qquad n \in [5, 10]$$

$\Rightarrow$ highly non-linear



- Looks like drainage rates are extremely slow ~ Ga time scales !

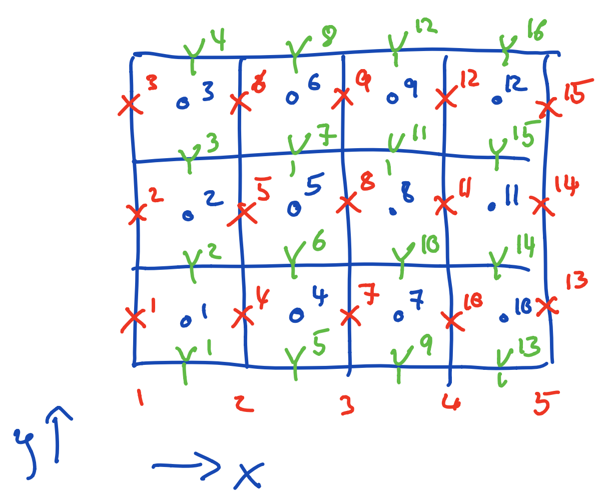
Today: Introduction to 2D numerics

- look at some Matlab functions
- Discrete operators

→ tensor products to go from 1D → 2D

Discrete operators ($\underline{D}$, $\underline{G}$) in 2D

Staggered grid in 2D



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

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

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

total faces: $N_f = N_{fx} + N_{fy} = 31$

Discrete gradient in 2D

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

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

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

$$\nabla h \sim \underline{dh} = \begin{bmatrix} dhx \\ dhy \end{bmatrix}$$ (choice)

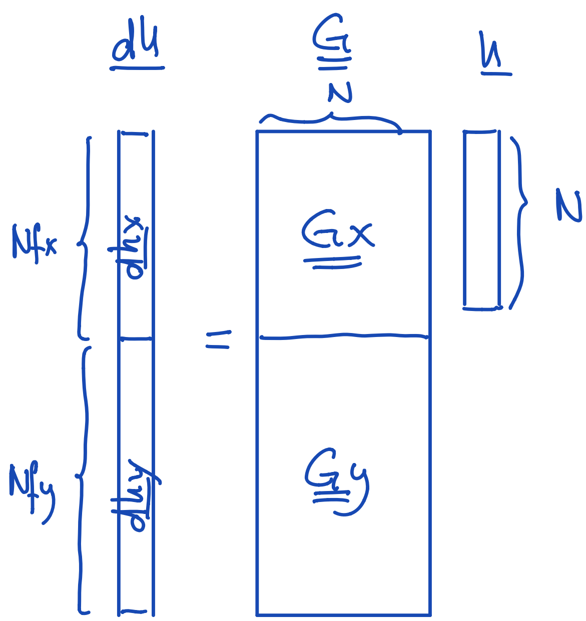
⇒ 2D gradient matrix can be decomposed

$$\underline{dh} = \underline{\underline{G}}\;\underline{h}$$

$$\begin{bmatrix}\underline{dhx}\\ \underline{dhy}\end{bmatrix} = \begin{bmatrix}\underline{\underline{Gx}}\\ \underline{\underline{Gy}}\end{bmatrix}\underline{h}$$

$$\underline{dhx} = \underline{\underline{Gx}}\;\underline{h}$$

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



Matrix dimensions:

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

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

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

Gradient takes values at cell centers and returns derivatives on cell faces

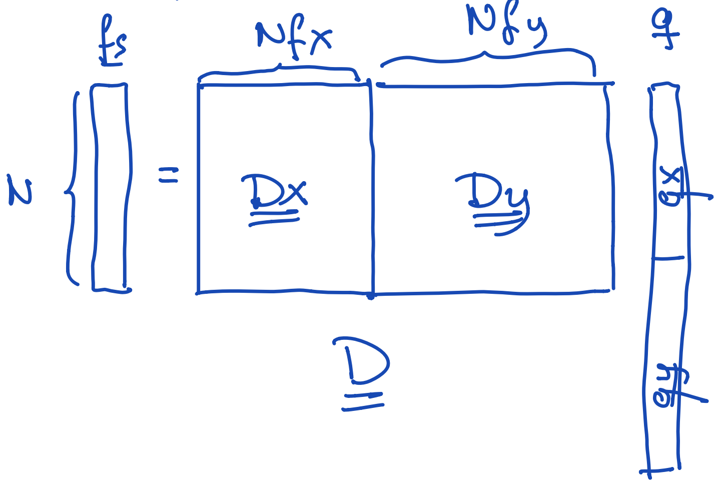
2D discrete divergence

$$\nabla\cdot q = \frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} \approx \underline{\underline{D}}\, q = \underline{\underline{D_x}}\, q_x + \underline{\underline{D_y}}\, q_y$$

$q = \begin{pmatrix} q_x \\ q_y \end{pmatrix}$ ← continuum

↑ discrete flux vector $q = \begin{bmatrix} q_x \\ q_y \end{bmatrix}$

$$\nabla\cdot q = f_s$$

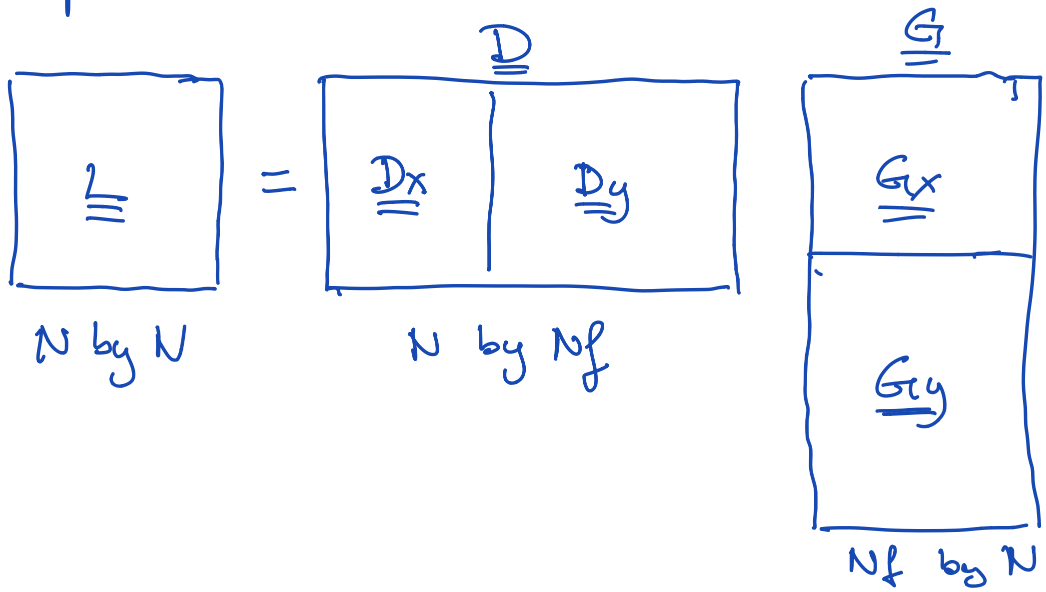


$\underline{\underline{D}}$ N by Nf

$\underline{\underline{D_x}}$ N by Nfx

$\underline{\underline{D_y}}$ N by Nfy

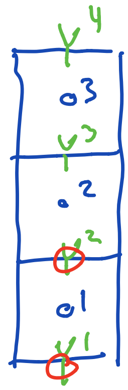
Laplacian: $\nabla^2 = \nabla\cdot\nabla$

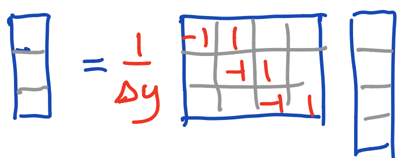


Building the 2D discrete divergence

Start with D_y in 1D $\qquad \nabla \cdot q = f_s$

$f_s \qquad D_y \qquad q = q_y \qquad \frac{\partial q}{\partial y} =$



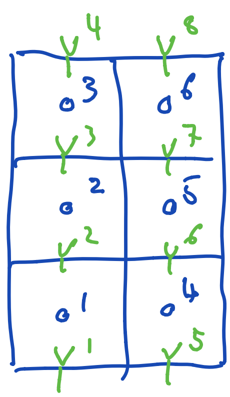


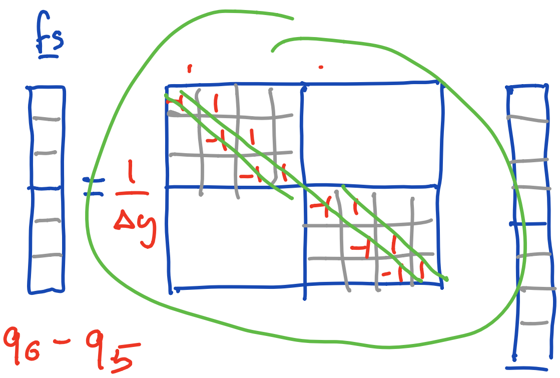
N_y by (N_y+1)

$$\frac{q_2 - q_1}{\Delta y}$$

Suppose we add second column

f_s



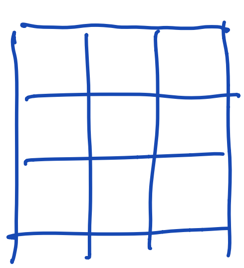


$$\frac{q_6 - q_5}{\Delta y}$$

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

2 by 2 block matrix with D_y^1 on diagonal

Suppose we add another row:



$$D_y^2 = \begin{bmatrix} D_y^1 & & \\ & D_y^1 & \\ & & D_y^1 \end{bmatrix}$$

3 by 3 block matrix

In general:

$\underline{\underline{D_y}}^2$ is a N_x by N_x block matrix with blocks of size N_y by (N_y+1). Diagonal entries are $\underline{\underline{D_y}}'$ and all others are zero.

Tensor product construction of $\underline{\underline{D_y}}^2$

The discrete 2D operators can easily and efficently be assembled using Kronecker/tensor products.

Definition:

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

$$\underline{\underline{A}} \otimes \underline{\underline{B}} = \begin{bmatrix} a_{11}\underline{\underline{B}} & a_{12}\underline{\underline{B}} & \cdots & a_{1n}\underline{\underline{B}} \\ a_{21}\underline{\underline{B}} & \ddots & & \vdots \\ \vdots & & & \\ a_{m1}\underline{\underline{B}} & \cdots & & a_{mn}\underline{\underline{B}} \end{bmatrix}$$

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

$$\boxed{\underline{\underline{D_y}}^2 = \underline{\underline{I_x}} \otimes \underline{\underline{D_y}}^1} = \begin{bmatrix} 1\,\underline{\underline{D_y}}^1 & & 0\cdot\underline{\underline{D_y}}^1 & \\ & \underline{\underline{D_y}}^1 & & \\ & & \underline{\underline{D_y}}^1 & \\ & & & \ddots & \\ & & & & \underline{\underline{D_y}}^1 \end{bmatrix}$$

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

In matlab : $\boxed{\underline{\underline{D_y}} = \mathrm{kron}(\underline{\underline{I_x}}, \underline{\underline{D_y}})}$

Next time → complete this