Lecture 4: Discrete Operators

Logistics: Monday office hours in JGB 4.216 G

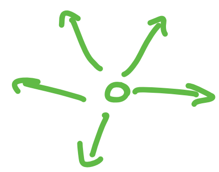
HW 1 has been posted

→ please make sure you can access it

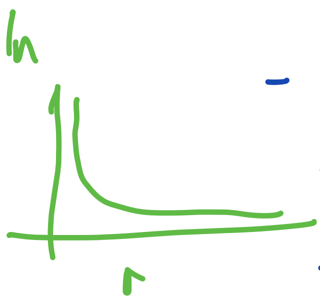
Last time: - Finite Differences: - one-sided diff.

- central diff.

- Differentiation matrix



- Example of flow near well



⇒ challenging due to "boundary layer"

- Tried and failed twice!

⇒ conservative finite differences (CFD)

Today: - Staggered grid

- Conservative differences

- Discrete operators

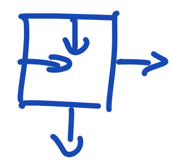
- coding basics

Discrete operators

Best to discretize the eqns in conservation form:

1) $$\nabla \cdot q = f_s$$

2) $$q = -K \nabla h \quad \left(\nabla \to \frac{dh}{dx}\right)$$



Highlights two basic operators in vector calculus:

1) Divergence of flux vector
2) Gradient of scalar potential
(3) Curl)

⇒ most PDE's in science and engineering are built from these

If we had discrete analogs of these operators:

- solve different problems
- clean & readable implementation
- dimension & coordinate system independent

Linear differential operators → matrices

We are looking for two matrices $\underline{\underline{D}}$ and $\underline{\underline{G}}$ so that

$$\nabla \cdot q = f_s \quad \rightarrow \quad \underline{\underline{D}}\, q = f_s$$

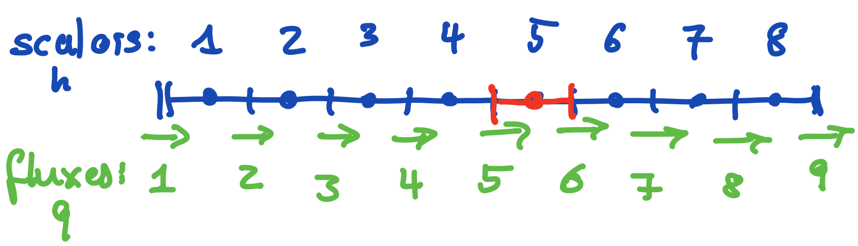
$$(K=1) \qquad q = -\nabla h \quad \rightarrow \quad q = -\underline{\underline{G}}\, \underline{h}$$

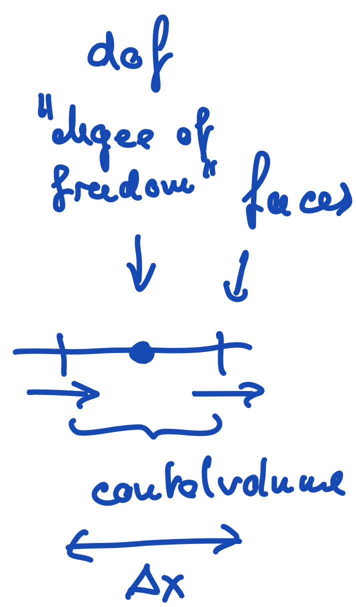
$$-\underbrace{\nabla \cdot \nabla}_{\nabla^2} h = f_s \qquad \boxed{-\underline{\underline{D}}\,\underline{\underline{G}}\,\underline{h} = f_s}$$

continuous   discrete

Staggered grid

⇒ needed "compact stencil"





$N = 8$ cells

$N+1 = 9$ faces

Discrete divergence operator

Divergence takes a flux vector, $\underline{q}$, and returns a scalar, f_s.

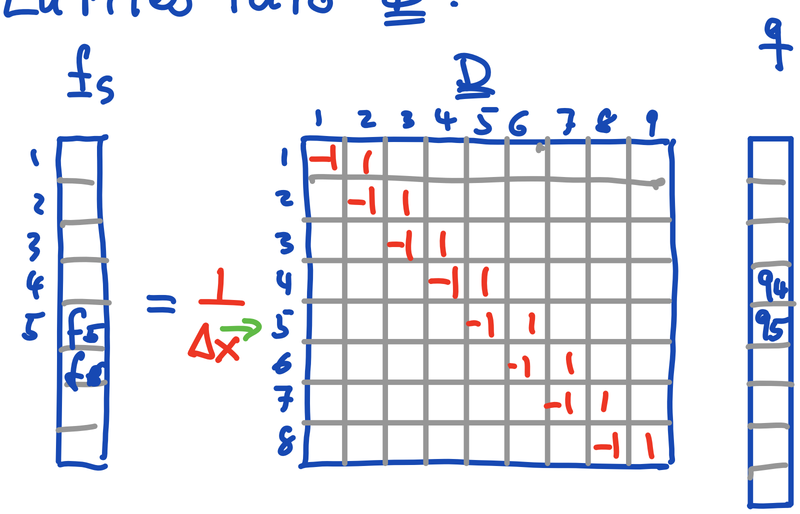
$$\nabla \cdot \underline{q} = f_s$$

discrete q: $N_x + 1$ by 1 (column vector)

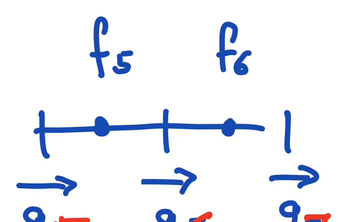
discrete $\underline{f_s}$: N_x by 1

$$\underset{N_x \cdot 1}{\underline{f_s}} = \underset{N_x \cdot (N_x+1)}{\underline{\underline{D}}} \; \underset{(N_x+1)\cdot 1}{\underline{q}}$$

Entries into $\underline{\underline{D}}$:



$$\nabla \cdot q = \frac{dq}{dx} \approx \frac{q_{i+1} - q_i}{\Delta x}$$



$$f_5 = \frac{q_6 - q_5}{\Delta x}$$

$$f_6 = \frac{q_7 - q_6}{\Delta x}$$

bidiagonal matrix

⇒ implement in Matlab with spdiags.m

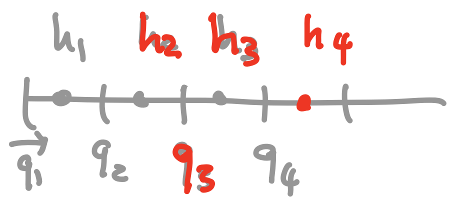
Discrete gradient operator

Takes a scalar and returns a vector

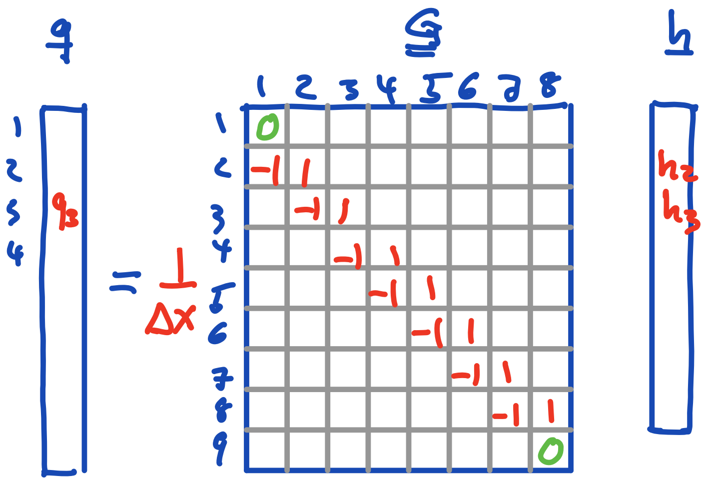
Cont: $q = -\nabla h \quad (k = 1)$

Discrete: $\underline{q} = -\underline{\underline{G}}\,\underline{h}$

$(N_x+1)\cdot 1 \quad (N_x+1)\cdot N_x \quad (N_x \cdot 1)$



Entries into $\underline{\underline{G}}$



$$q = -\nabla h \approx -\frac{h_i - h_{i-1}}{\Delta x}$$

$$q_3 = -\frac{h_3 - h_2}{\Delta x}$$

grad

$$q_4 = -\frac{h_4 - h_3}{\Delta x}$$

$$q = -\nabla h \qquad \underline{dh} = \underline{\underline{G}}\,\underline{h}$$

$$q = -\underline{dh}$$

On bnd's we set the flux to zero

$\Rightarrow$ natural/no flow bc

Compare $\underline{\underline{D}}$ and $\underline{\underline{G}}$:

$$\boxed{\underline{\underline{G}} = -\underline{\underline{D}}^T}$$ (in the interior)

⇒ because div & grad are adjoint operators

General coding comments

- No for-loops !

  ⇒ vectorized programming

  $\underline{a}$ $\underline{b}$ $\underline{c} = \underline{a} * \underline{b}$ component wise

```
for i=1:N
    c(i) = a(i)*b(i);
end
```

vectorized way `c = a.*b`

(. ↑ element wise)