Lecture 20: Advection-Diffusion Equation

Logistics:
- Today is last in person lecture
- Going forward I will record lectures
- HW7 due next Thursday
  - Problem with 2D BC's
- HW 5 last chance next Thursday
- There will be more homeworks

  HW8: Advection 1D + Transient 1D

  → HW9: Advection 2D + ADE Flow + Transport

  HW10: Newtons Method + Richards Eqn

Last time: Advection Equation $Pe \to \infty$

→ $\phi \frac{\partial c}{\partial t} + \nabla \cdot (q c) = f_s$ 2D variable ϕ

→ $\frac{\partial c}{\partial t} + v \frac{\partial c}{\partial x} = f_s$ $v = \frac{q}{\phi}$ 1D const. ϕ

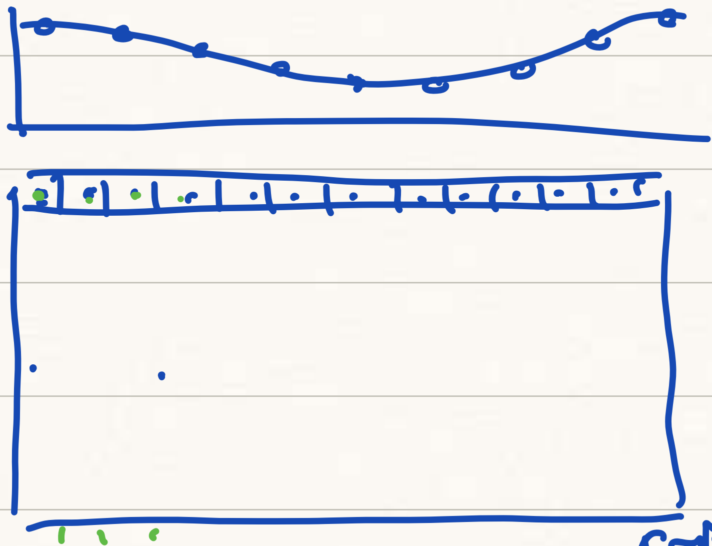
- Method of characteristics

  → $c(x,t) = c_0(x - vt)$

  initial cond. traveling wave coord

Today: Advection-Diffusion Equation

Problems with 2D BC:



In 2D BC.dof_dir = Grid.dof_ymax BC.g are vectors

=> errors inside solve_lbvp.m

need to have same size

[Xc, Yc] = meshgrid(Grid.xc, Grid.yc)

↑ matrices ↑ vector

need to evaluate analytic soln at cell centers

hana(Xc(BC.dof_dir), Yc(BC.dof_dir))

Xc(Grid.dof_ymax)

Grid.xc (Height − Grid.dy/2) ones(Nx, 1)

How to discretize advection?

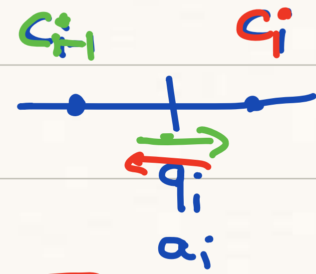
PDE: $\nabla \cdot (\overset{a}{\widetilde{\underline{q} c}} - D_m \nabla c) = f_s$

Discrete: $\underbrace{\underline{\underline{D}} * (\underline{\underline{A}}(\underline{q}) - \underline{\underline{K_d}}\, \underline{\underline{G}})}_{\underline{\underline{L}}}\ \underline{c} = f_s$

Steady
advection
diffusion

Compute advective flux: $\underline{a} = \underline{q}\, c$

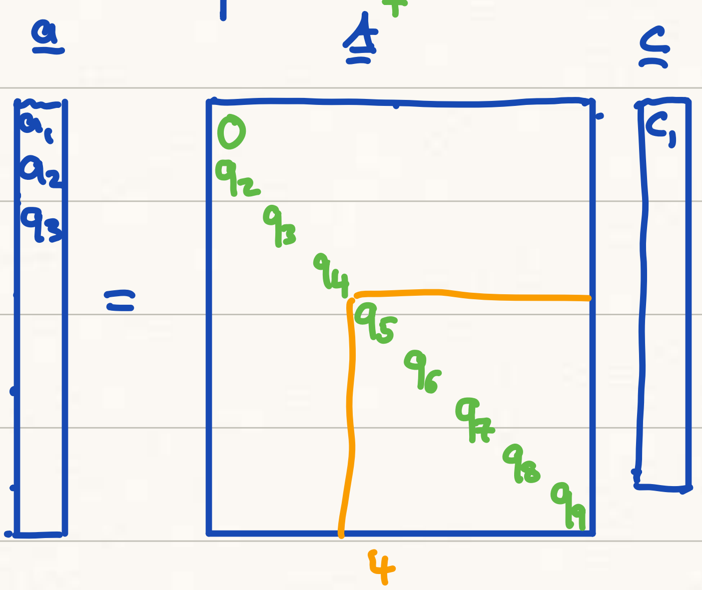
$a_i = q_i\, c_{i-\frac{1}{2}}$



Upwind method: $c_{i-\frac{1}{2}} = \begin{cases} c_{i-1} & q_i > 0 \\ c_i & q_i \le 0 \end{cases}$

How do we construct $\underline{\underline{A}}(\underline{q})$

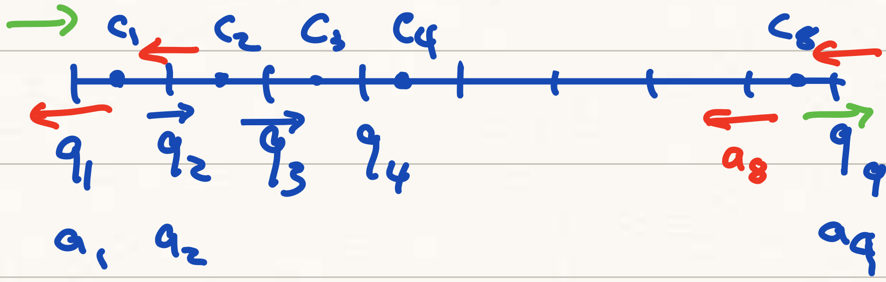
Assume $\underline{q} > 0$: $c_{i-\frac{1}{2}} = c_{i-1}$



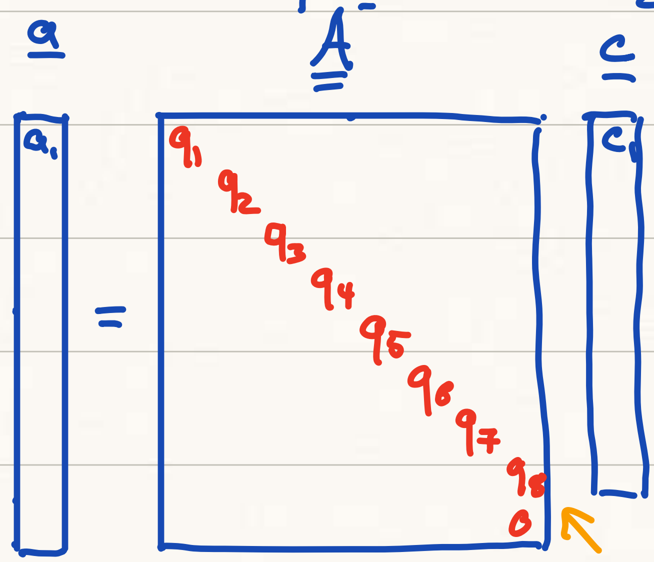
$a_2 = \underline{q_2\, c_1}$

$a_3 = q_3\, c_2$

$a_9 = q_9\, c_8$



Assume $q < 0$: $c_{i-\frac{1}{2}} = c_i \Rightarrow a_i = q_i c_i$



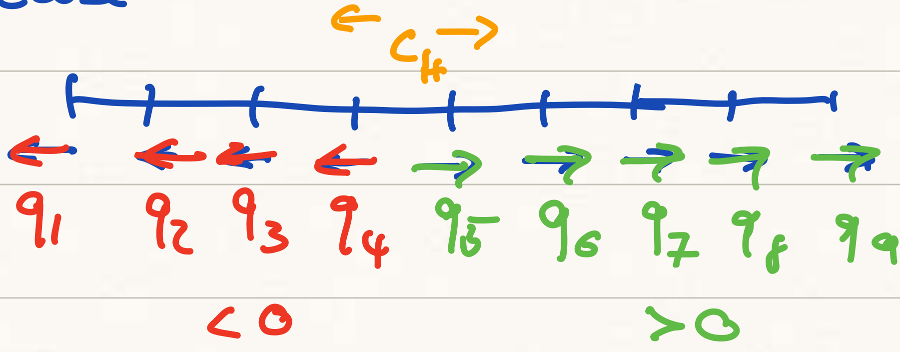
$a_1 = q_1 c_1$

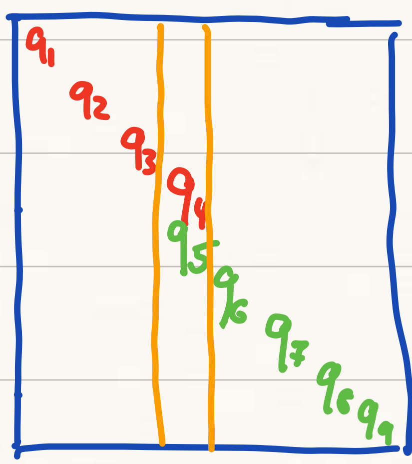
$a_2 = q_2 c_2$

$\vdots$

$a_8 = q_8 c_8$

Case



$\underline{\underline{A}} =$ 

$\underline{\underline{A}} =$ 

$\underline{\underline{A}}^{+} + \underline{\underline{A}}^{-}$

need to automatically turn of fluxes in qn and qp with wrong sign

qn = min(q(1:Nx), 0); replaces pos. entries in q with zero

qp = max(q(2:Nx+1), 0);

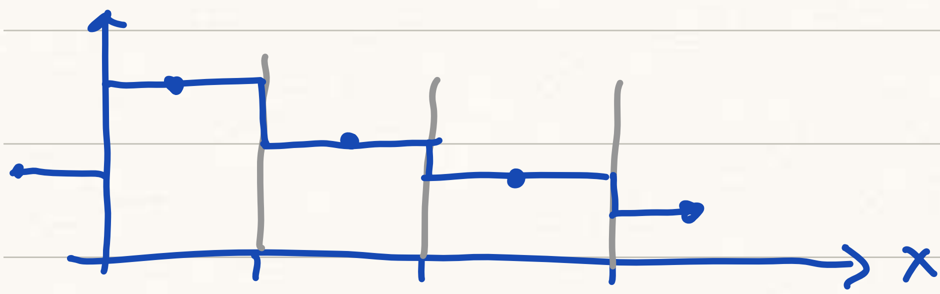
$\underline{\underline{A}}$ = spdiags([q_p, q_n], [-1, 0], Nfx, N)

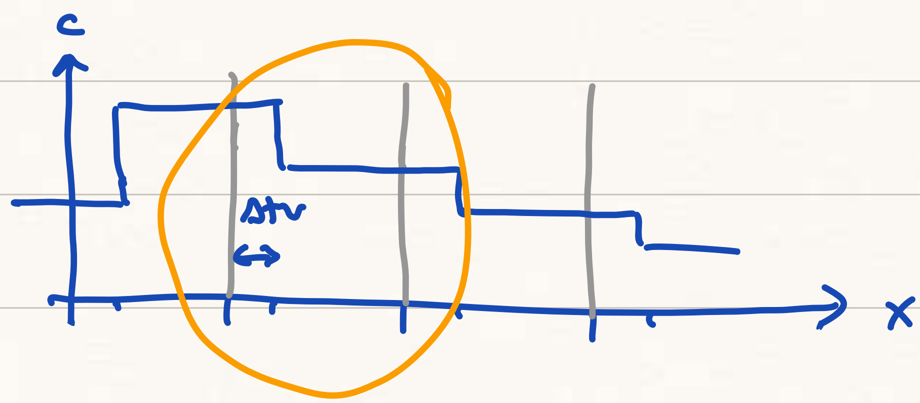
Unlike $\underline{\underline{D}}$, $\underline{\underline{G}}$ $\underline{\underline{A}}(q)$

Origin of time step restriction

for explicit time integration of advection

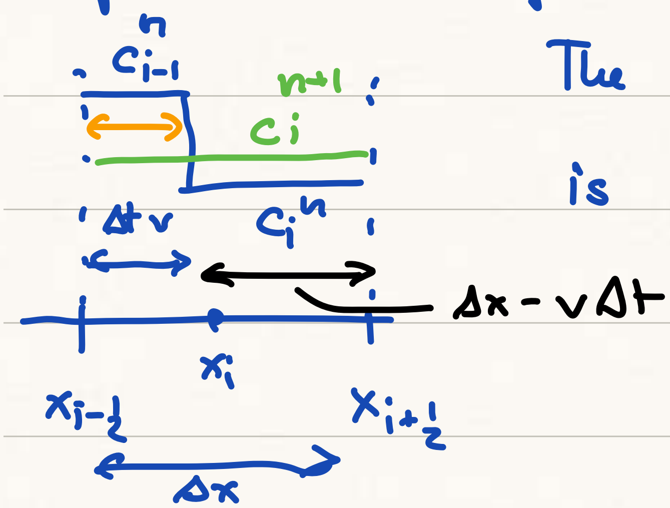
c^n





- value at cell centers represents average conc. over cell.
- discrete soln is a set of steps
- Soln. moves to right with velocity $v = \frac{q}{\phi}$

after Δt the fronts have moved $\Delta t v$



The new c_i^n at end of time step is the average over the cell

$$c_i^{n+1} = \frac{1}{\Delta x}\left(\int_{x_{i-\frac{1}{2}}}^{x_{i-\frac{1}{2}}+v\Delta t} c_{i-1}^{n}\, dx + \int_{x_{i+\frac{1}{2}}+v\Delta t}^{x_{i+\frac{1}{2}}} c_i^n\, dx\right)$$

$$= \frac{1}{\Delta x}\left(c_{i-1}^{n} v\Delta t + c_i^n (\Delta x - v\Delta t)\right)$$

rearrange:

$$\boxed{c_i^{n+1} = \frac{v\Delta t}{\Delta x} c_{i-1}^{n} + \left(1 - \frac{v\Delta t}{\Delta x}\right) c_i^n}$$

Note: $\alpha = \frac{v\Delta t}{\Delta x}$ CFL number

Interpretation: $c_i^{n+1} = \alpha\, c_{i-1}^{n} + (1-\alpha)\, c_i^{n}$

- weighted average of two previous values
- if $\alpha > 1 \Rightarrow$ extrapolation
  because front has moved through the cell

Stability condition for explicit time step:

$$\alpha = \frac{v \Delta t}{\Delta x} \le 1$$

$$\Rightarrow \boxed{\Delta t \le \frac{\Delta x}{v}}$$ <u>CFL</u> condition