Lecture 10: Fluxes and Flux Boundary Conditions

Logistics: - HW2 due today (1: 16 2: 14 3: 12)

- HW3 will be posted today

$\Rightarrow$ start early and get help in office hours

Last time: - Heterogeneous coefficients

Continuous: $-\nabla \cdot [K(\underline{x}) \nabla h] = f_s$

Discrete: $-\underline{\underline{D}} * [\underline{\underline{Kd}}\ \underline{\underline{G}}\ \underline{h}] = f_s$

$\underline{\underline{Kd}}$ Nx+1 by Nx+1 diagonal matrix

with Kmean on diagonal

$$\boxed{\text{Kmean} = (\underline{\underline{M}} * \underline{K}.\hat{\ } p).\hat{\ }(1/p)}$$ power-law

$\rightarrow p=1$ arithmetic

$\rightarrow p=-1$ harmonic

- Radial coordinates:

div: $\nabla \cdot = \frac{1}{\underline{x}^{(d-1)}} \frac{d}{dx} \underline{x}^{(d-1)}$

$\underline{\underline{D}} = \underset{\uparrow x_c}{\text{Rinv}} * \underline{\underline{D}} * \underset{\uparrow x_f}{\underline{\underline{R}}}$

Today: Fluxes and flux BC's

Neumann / Flux Boundary Conditions

Dirichlet BC prescribe h on bnd.

$\Rightarrow$ eliminate h on bnd as constraint

Neuman BC prescribe flux $q \sim \frac{dh}{dx}$

$\Rightarrow$ h on Neu BC is still unknow

$\Rightarrow$ Neuman BC's are <u>not</u> implemented as constraints

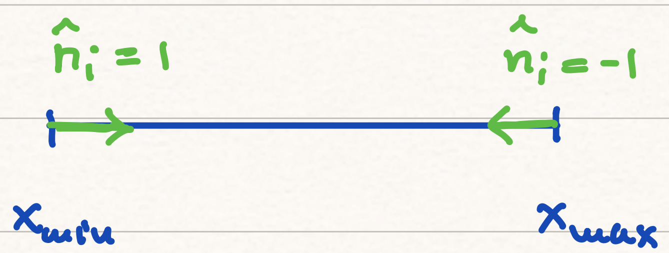
Sign Convention

In this class we consider <u>inflows</u> to be <u>positive</u> for reasons that will become clear when we discuss flux computation

$$\boxed{q_B = q \cdot \hat{n}_i}$$

$\hat{n}_i =$ inward normal

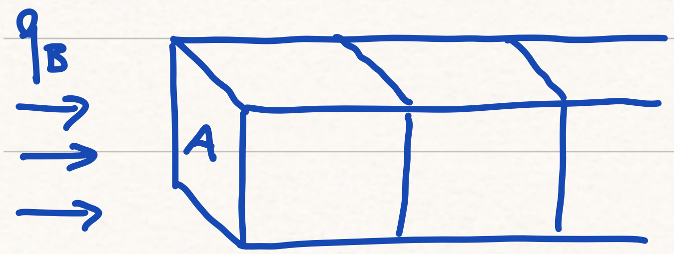
$q_B =$ bnd flux



$\Rightarrow$ $q_B > 0$ inflow on bnd

Implementation of Neuman BC

We implement flux BC as an equivalent source / sink term. to ensure mass conservation.



Total flow rate across bnd face: $Q_b = A\, q_b$

Equivalent source term: $Q_b = V f_n$

cell volume

equating and solve for f_n

$$\boxed{f_n = q_b \frac{A}{V}}$$

Note: sign of f_n is automatically correct
because $q_b > 0$ is an inflow

In general $\underline{f}_n$ is N_x by 1 r.h.s vector
with N_n non-zero entries, one for each
Neuman BC applied.

For a problem with Neuman BC's the linear system becomes:

$$\underline{\underline{L}}\,\underline{h} = \underline{f}_s + \underline{f}_n$$

To construct fn we define:

BC.dof_neu = Nn by 1 vectors of cells on Neuman Bnd

BC.dof_f_neu = Nn by 1 vector of faces on Neu. Bnd

BC.qb = Nn by 1 vector of bnd fluxes

also need to add cell volumes and face area to Grid

Grid.V = Nx by 1
Grid.A = Nfx by 1 } assume other dimensions are unity

In [B,N,fn] = build_bnd(BC, Grid)

add following line

$$\underbrace{\text{fn(BC.dof\_neu)}}_{Nn\cdot 1} = \underset{\uparrow\; Nn\cdot 1}{\text{BC.qb}} \ast \text{Grid.A(BC.dof\_f\_neu)} \;./\; \text{Grid.V(BC.dof\_neu)}$$

Compute fluxes of Gradient Fields

We regularly need to compute fluxes from the gradient of a scalar potential field.

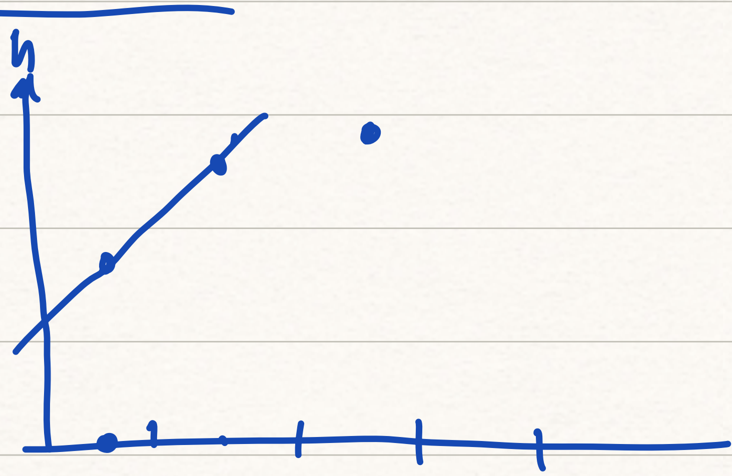
Darcy: $\underline{q} = -K \nabla h$ $\qquad$ h = scaler potential

Discrete approx: $\boxed{\underline{q} = - \underline{\underline{Kd}}\ \underline{\underline{G}}\ \underline{h}}$

This works in the interior of the domain, but on bnd $\underline{\underline{G}}\,\underline{h}$ is zero by construction.

$\Rightarrow$ need to reconstruct boundary flux

Option 1: Extrapolate to bnd

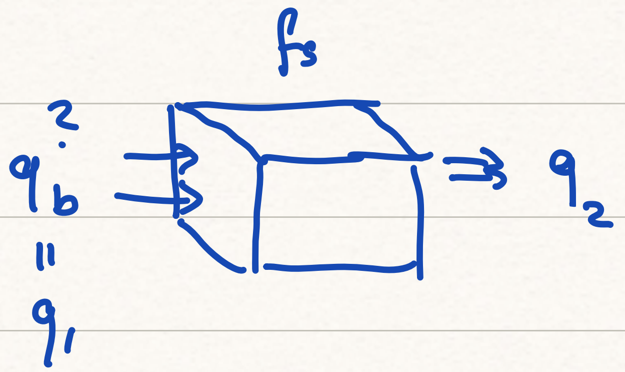


Equivalent to using one-sided derivative

Problem: loose discrete conservation because of interpolation error

Option 2: Reconstruct from discrete balance

Idea: Use discrete mass balance in bnd cell
to compute the exact bnd flux required
to conserve mass.



$$\nabla \cdot q = f_s$$

$$\frac{q_2 - q_1 \overset{= q_b}{}}{\Delta x} = f_s$$

Consider our discrete system:

$$\underline{\underline{L}}\,\underline{u} = \underline{f}_s \qquad \underline{u} = \text{unknown } (\underline{h})$$

Residual of equation

$$\underline{r}(\underline{u}) = \underline{\underline{L}}\,\underline{u} - \underline{f}_s$$

If discrete eqns are satisfied $\underline{r} = \underline{0}$

In bnd cells $\underline{r} \neq \underline{0}$ because $\underline{\underline{G}}$ has natural BC

$\Rightarrow$ non-zero residual in bnd cells
contains information about bnd flux!

$$r = \frac{q_2 - 0\; q_b}{\Delta x} - f_s \neq 0$$

Consider system with flux BC's

$$\underline{\underline{L}}\,\underline{u} = \underline{f}_S + \underline{f}_N$$

residual: $\underline{r} = \underline{\underline{L}}\,\underline{u} - \underline{f}_S = \underline{f}_N$

on bnd $\Rightarrow \underline{r} = \underline{f}_N$

The residual on bnd is equal to the rhs vector due to the flux BC's!

Entries of f_N on Neu bnd are: $f_n = q_b \frac{A}{V}$

If we are given $\underline{r} = f_N$ we can reverse this argument and solve for flux: $q_b = f$

$$q_b = f_n \frac{V}{A} = \underline{r} \frac{V}{A}$$

This is also true on Dirichlet BC so that the bnd flux in general is given by:

$$\boxed{|q_b| = |r| \frac{V}{A}}$$

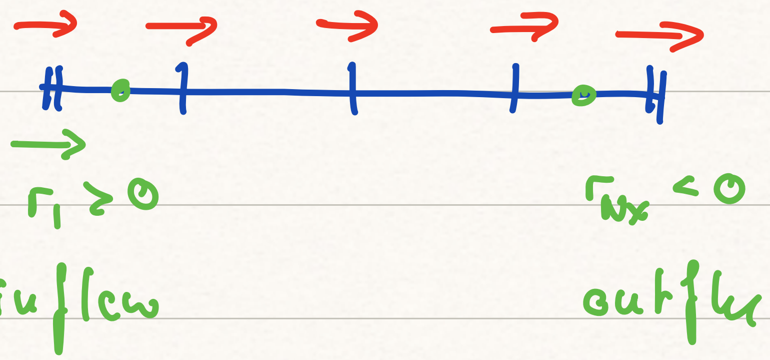
note: up to a sign

Sign change

We want q_b to have sign that matches with the rest $\underline{q} = -\underline{\underline{Kd}}\,\underline{\underline{G}}\,\underline{h}$.

Here q's are positive if they point in pos. x-dir.

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



Need to change sign on x_{max} bnd

Implementation:

In function comp_flux_res.m we compute fluxes as follows:

Define 2 vectors:

dof_cell : column vector that contains all bnd cells (dir, neu)

dof_face : column vector that contains all associated faces

To compute all bound. fluxes:

$$\underline{q} = -\underline{\underline{Kof}}\ \underline{G}\ h$$

$$* \quad res = \underline{\underline{L}}\ \underline{h} - f_s$$

res(dof_cell)

$$q(dof\_face) = \underline{sign} .* \underline{r}(dof\_cell, :) .* V(dof\_cell) ./ A(dof\_face);$$

where $\underline{sign} = \begin{cases} -1, & dof\_face \in \text{ max bnd} \\ 1, & dof\_face \in \text{ min bnd} \end{cases}$

You can use ismember.m to check which boundary you are on.