Lecture 9: Fluxes and flux BC's

Logistics: - no HW this week (sorry)

Last time: - Discretization of heterog. coefficients

$$-\nabla \cdot [K(x) \nabla h] = f_s$$

$$-\underline{\underline{D}} [\underline{\underline{K_d}}\ \underline{\underline{G}}]\ \underline{h} = \underline{f_s}$$

- K_d is N_f by N_f diagonal matrix

$K_{means} = (\underline{H} * \underline{K}.\hat{}\, p).\hat{}\, (1/p)$ 'power-mean'

$$\underline{\underline{K_d}} = \begin{pmatrix} K_1 & & & \\ & \frac{K_{i-1}+K_i}{2} & & \\ & & K_{means} & \\ & & & K_N \end{pmatrix}$$

- Use harmonic mean ($p=-1$) for $\underline{K}$

Today: - Flux boundaries (Neumann BC's)

- Computing fluxes

Reconstruction on boundary

Neumann BC's / Flux

Dir. BC prescribe unknown on bnd

⇒ eliminate it

Neu. BC prescribe flux / derivative, but unknown itself is still unknown!

⇒ cannot eliminate it

Neu BC are not implemented as constraints

Sign convention

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

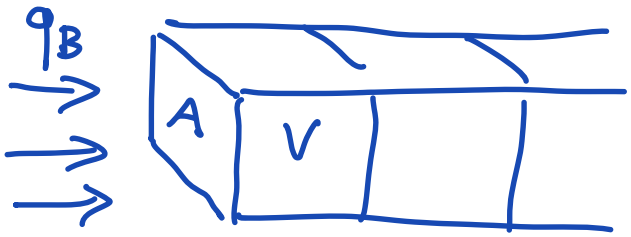
$\hat{n}_i$ = inward unit normal

q_B = flux across bnd

$q_B > 0$ inflow into domain

Implementation of Neumann BC

Implement flux BC as an equivalent source/sink term to ensure conservation.



Total flow rate across bnd face: $Q_B = q_B A$

Equivalent source term: $Q_B = V f_n$ ← vol. source term $\frac{L^3}{L^3 T}$

↑ cell volume

Setting them equal: $\boxed{f_n = q_b \frac{A}{V}}$ (single bnd cell)

A: face, V: cell

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

In general $\underline{f}_n$ is N by 1 r.h.s. vector

with Nu non-zero entries, where Nu is # of cells with Neu. BC's.

For a problem with Neumann BC's the linear system looks as follows:

$$\underline{\underline{h}} = \underline{f_s} + \underline{f_n}$$

$\underline{f_s}$: physical source term in PDE

$\underline{f_n}$: numerical from implementation of Neu BC's

To construct $\underline{f_n}$ we define:

BC.dof_neu = Nu by 1 col. vector of cells with (non-zero) Neu. BC's

BC.dof_f_neu = Nu by 1 col. vect of faces with Neu. BC

BC.qb = Nu by 1 col. vector of bnd fluxes

We need to add two vectors to Grid.

Grid.A = Nf by 1
Grid.V = N by 1
} assume other dimensions are unity

Generate fn as sparse vector

fn = spalloc(N, 1, Nn);

Compute and place Nn entries of fn

```
fn(BC.dof_neu) = BC.qb .* Grid.A(BC.dof_f_neu) ./
                                    Grid.V(BC.dof_neu)
```

Neu BC's can be implemented in 1 line in build_bnd !

Note: We assume each bnd cell has only one face with an ~~is~~ bnd flux.

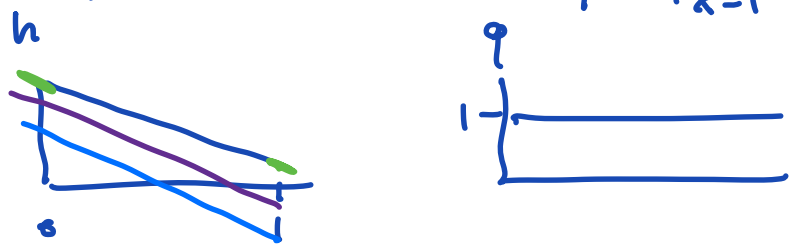
Pure Neumann BC

If a problem has only Neumann BC's there are two potential issues.

1) Undetermined coefficient

PDE: $-\nabla^2 h = 0 \qquad x \in (0,1]$

BC: $q \cdot \hat{n}_i|_{x=0} = 1 \qquad q \cdot \hat{n}|_{x=1} = -1$



⇒ solved by prescribing one constraint

2) Compatability condition

$$-\nabla^2 h = f_s \qquad x \in [0,1]$$

$$q \cdot \hat{n}_i|_{x=0} = q_L \qquad q \cdot \hat{n}_i|_{x=1} = q_R$$

$$\int_0^1 f_s \, dx = q_L + q_R \qquad f_s = f_1 + f_2 +$$

Compute fluxes

Flux: $q = -K \nabla h$

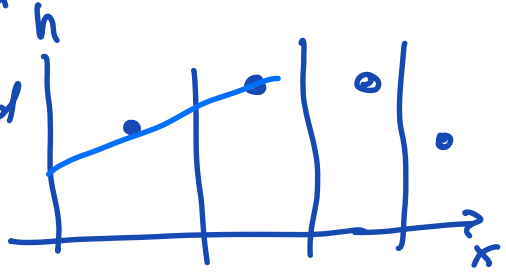
discrete $\boxed{\underline{q} = -\underline{\underline{K_d}}\,\underline{\underline{G}}\,\underline{h}}$

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

⇒ need to reconstruct bnd fluxes

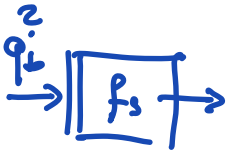
Option 1: Extrapolate h to bnd

Equivalent to using a one-sided approx. for derivative



⇒ has error and we loose discrete conservation.

Option 2: Reconstruct from discrete balance in bnd cell



use discre balance on bnd cell to compute the exact bnd flux required for conservation.

$$(q_b + q_1) A = V f_s \Rightarrow \text{solve for } q_b$$

Consider linear system:

$$\underline{\underline{L}}\,\underline{h} = \underline{f_s}$$

Discrete residual of eqn:

$$\underline{r}(\underline{h}) = \underline{\underline{L}}\,\underline{h} - \underline{f_s}$$

If discrete eqn. is satisfied $\underline{r}(\underline{h}) = \underline{0}$

but on the bnd cells $\underline{r} \neq 0$ because

$q_b = 0$ by natural bc in $\underline{\underline{G}}$

$\Rightarrow$ non-zero residual in bnd cell contains the information about the bnd flux

Consider system with Neu BC's

$$\underline{\underline{L}}\,\underline{h} = \underline{f_s} + \underline{f_n}$$

from def. of residual. $\underline{r} = \underline{\underline{L}}\,\underline{h} - \underline{f_s}$

~~on bnd~~ $\underline{r} = \underline{f_n}$

Residual on bnd is equal to flux source/sink term

Entries of f_w are: $f_w = q_b \frac{A}{V}$

Now if we are given $f_w = r$ we can reverse this argument and solve for flux

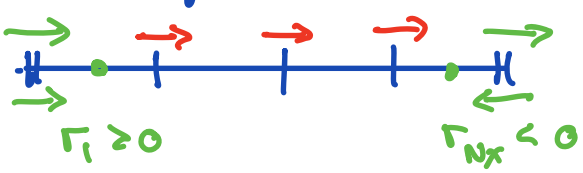
$$|q_b| = |f_w| \frac{V}{A} = |r| \frac{V}{A}$$

This also works on Dirichlet ~~BC~~ bnd's

Sign change

We want q_b to have sign that fits with the rest of fluxes computed by $q = -\underline{\underline{K}} \underline{\underline{G}} \underline{h}$.

These q's >0 if they point in pos. x-dir.



$\Rightarrow$ need to change sign on x_{max} bnd.

Implementation

In function comp_flux_res.m we will compute fluxes as follows.

dof_cells = column vector containing all bnd cells

dof_face = column vectos containing all bnd faces

We assume they are the same length.

Compute all bnd fluxes in single line

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

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

You can use ismember.m