Neumann Boundary Conditions

Dirichlet BC's prescribe the unknown on boundary, so that it can be eliminate it. Neumann BC's prescribe the flux/derivative, so that we still have to solve for the unknown on boundary.

$\Rightarrow$ Neumann BC's are not implemented as constraints

For Example 2 Aquifer with polar recharge we have the following Neumann BC

$$q' = -\left.\frac{dh'}{dx'}\right|_0 = \Pi$$

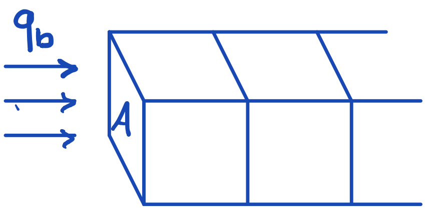
Note: In this class we consider inflows to be positive, i.e. we multiply with inward normal

$$q \cdot \hat{n}_i = q_B \qquad \hat{n}_i = \text{inward normal}$$

$q_B > 0$ · inflow

Implementation of Neumann BC

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



face area

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

Equivalent source term: $Q_b = V f_n$

cell volume

$$\Rightarrow \boxed{f_n = q_p \frac{A}{V}} \qquad \text{(for a single cell)}$$

Note: sign of f_n is automatically correct because $q_b > 0$ is an inflow and f_n has same sign.

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

For a problem with Neumann BC's the linear system is : $\underline{\underline{L}}\,\underline{h} = \underline{f_s} + \underline{f_n}$

To construct fn we define:

BC.dof_neu = Nn by 1 vector of cells with Neumann BC

BC.dof_f_neu = Nn by 1 vector of faces with Neum. BC

BC.qb = Nn by 1 vector of prescribed fluxes

and add cell volumes and face areas to Grid.

Grid.A = Nf by 1 } assume other dimensions

Grid.V = N by 1 } are unity!

Compute and place the Nn entries of fn

$$\boxed{\text{fn(BC.dof\_f\_neu) = } \underline{\text{qb}} \text{ * Grid.A(BC.dof\_f\_neu)/Grid.V(BC.dof\_neu)}}$$

⇒ Neumann BC can be implemented in one line in build_bnd.m!