Neumann Boundary Conditions

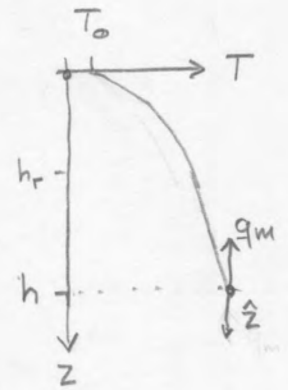
- Dirichlet BC's prescribe the value of the unknown, u, on the boundary. Therefore we can eliminate it from the solution
- Neumann BC's prescribe the derivative of the unknown, $\nabla u \cdot \hat{n}$, on the boundary, so the unknown, u, itself still has to be determined. Hence, Neumann BC's cannot be implemented with constraints as before

- In many problems, including heat flow, the gradient of the unknown, ∇u, corresponds to a flux. So that Neumann BC's prescribe a flux into and out of the domain.
- The physical BC is on the flux, not the derivative
  $\Rightarrow$ implementation that prescribes boundary fluxes

Example: Continental geotherm with mantle heat flow

PDE $\quad -\nabla\cdot(k\nabla T) = \rho H_0 e^{-z/h_r} \quad z \in [0,h]$

BC. $\quad T(0) = T_0 \qquad q\cdot\hat{z}|_h = -|q_m|$

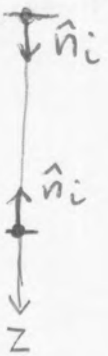


here $|q_m|$ is the magnitude of mantle heat flow entering the crust and the minus sign is necessary because q_m and $\hat{z}$ point opposite directions.

Note the awkward notation is due to the fact that we use a depth coordinate but want to refer to mantle heat flow as a positive number

Boundary flux convention in this class

We refer to the boundary fluxes as q_b and assume that inflows are positive ($q_b > 0$) and outflows are negative ($q_b < 0$) because it is physically intuitive!



This means that we are using the <u>inward</u> normal in statement of BC: $\underline{q} \cdot \hat{n}_i = q_b$

So in our geotherm example the mantle heat flow, q_m, is an inflow and prescribed as a positive boundary flux. ($q_b > 0$)

In contrast, the surface heat flow, q_0, is an outflow and would be prescribed as a negative boundary flux ($q_b < 0$)

Hence we prescribe the mantle heatflow in our problem as

```
BC.dof_neu = Grid.dof_xmax,     % specify bottom
BC.dof_f_neu = Grid.dof_f_xmax;
BC.qb = qm; % where qm > 0
```

We implement this flux across the boundary as an equivalent source/sink term, f_n, in the boundary cell.

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

Compute equivalent volumetric source term: $Q_b = V f_n$

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

Note that sign is automatically correct because an inflow corresponds to a positive source term.

This will be implemented in the function build_bnd.m.

The BC structure contains following relevant information:

BC.dof-neu = Nn by 1 vector of cells with Neumann BC's
BC.dof-f-neu = Nn by 1 vector of faces with Neumann BC's
BC.qb = Nn by 1 vector of prescribed bnd fluxes

here Nn is the number of applied Neumann BC's

Note that all vectors must be same length, i.e. each cell is associated with one face and flux!

Need to construct fn a Nx by 1 r.h.s. vector with Nn non-zero entries. as follows.

```
fn(BC.dof-neu) = qb * Grid.A(BC.dof-f_neu)/Grid.V(BC.dof-neu);
```

Hence the Neumann BC's can be added with a single line of code. This will work for any Nn.

Here Grid.A is a Nfx by 1 vector of face areas and Grid.V is a Nx by 1 vector of cell volumes
Assume the third dimension is unity.