Compute Fluxes of Gradient Fields

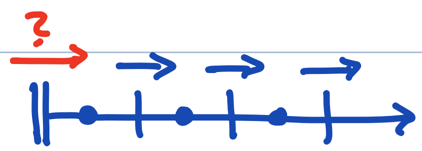
We regularly need to compute fluxes of the gradients of scalar potential fields.

$$q = -\kappa \nabla T \qquad T = \text{scalar potential}$$

$$\Rightarrow \quad \nabla \times q = 0$$

Discrete approximation: $\boxed{\underline{q} = -\kappa \underline{\underline{G}}\, \underline{u}}$

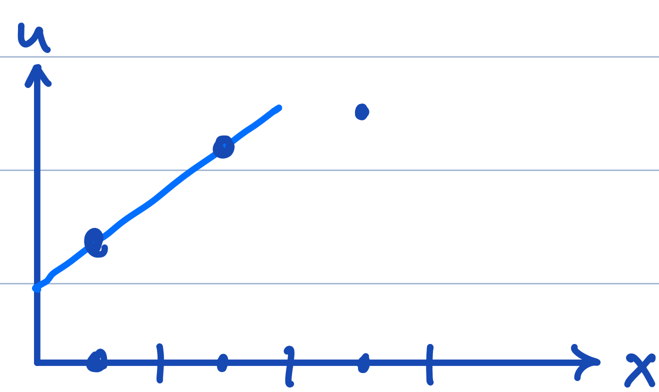
This works in the <u>interior</u> of the domain, but on boundary $\underline{\underline{G}}\,\underline{u}$ is zero by construction.



$\Rightarrow$ need to <u>reconstruct</u> boundary flux

Option 1: Extrapolate to bnd

Equivalent to using a one-sided derivative



Problem: lose discrete conservation because error in interpolation

Option 2: Reconstruct from discrete balance

Idea: Use the discrete balance in the bnd cell to compute the exact bnd flux required for conservation.

Consider a discrete linear system

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

Discrete residual of equation

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

If the discrete equations are satisfied $\underline{r} = \underline{0}$.
In the bnd cells $\underline{r} \neq 0$ because $\underline{\underline{G}}$ arbitrarily sets the gradient/flux to zero.

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

Consider a system if flux boundary

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

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

$$\Rightarrow \underline{r} \equiv \underline{f}_N \quad \text{on bnd!}$$

The residual on bnd is equal to the r.h.s. vector due to the boundary fluxes!

Entries of $\underline{f}_N$ on bnd are: $f_n = q_b \frac{A}{V}$

If we are given $\underline{r} = \underline{f}_n$ we can reverse this argument and solve for flux: $q_b = f_n \frac{V}{A} = r \frac{V}{A}$

This also works on Dirichlet bnd so that the boundary flux is generally given by:

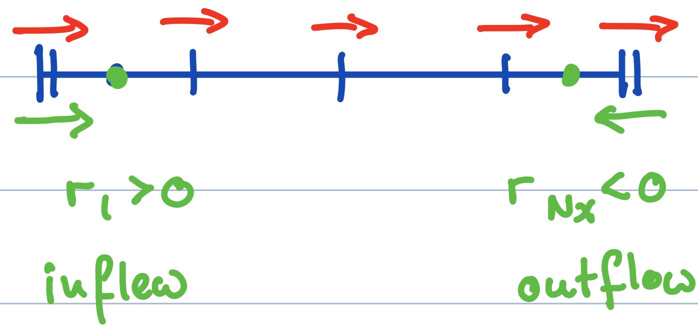
$$\boxed{|q_b| = |r| \frac{V}{A}}$$ note: up to a sign!

Note: We assume only 1 face of bnd cell has flux!

Sign change:

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

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



Need to change sign on x_{max} bnd.

Implementation

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

Define two vectors:

dof_cell : column vector containing all bnd cells

dof_face : column vector containing all associated bnd faces

These vectors are same length because we assume only one face is associated with each bnd cell.

Compute all bnd fluxes togeather in one line

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

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

You can use ismember.m to detect bnd.