Discretisation issues related to tensorial relative permeabilities

Eirik Keilegavlen, Annette Stephansen, Jan Nordbotten

University of Bergen & CIPR

Oktober 14 2010

Relative permeability

Traditionally, the relative permeability is modelled as a scalar

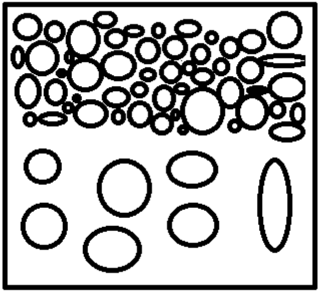
- Convenient for measurements
- Well established numerical treatment (upstream weighting)

Tensorial relative permeability

Claim: Rel perm is in general tensor; scalar only in special cases.

Tensor rel perm can arise in

- Multi-phase upscaling
- Vertically averaging
- Pore scale



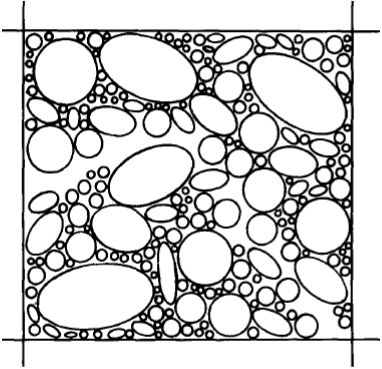
Focus here: Find a way to discretise flow with tensor rel perm.

Tensorial relative permeability

Claim: Rel perm is in general tensor; scalar only in special cases.

Tensor rel perm can arise in

- Multi-phase upscaling
- Vertically averaging
- Pore scale



Focus here: Find a way to discretise flow with tensor rel perm.

Tensorial relative permeability

Properties of a rel perm tensor:

- Anisotropic
- May be 0 in one direction
- May rotate the flow field
- Is a function of saturations

Discretisation

Rising level of difficulty:

- Scalar rel perm
- Tensor, aligned with $\mathbf{K}$
- Tensor, not aligned with $\mathbf{K}$ (rotation)

Issues:

- No upstream direction
- Rotation changes with saturation

Governing equations

- Scalar case:
  - Pressure: $\mathbf{u}_T = -\lambda_T \mathbf{K} \nabla p \,, \qquad \nabla \cdot \mathbf{u}_T = q$
  - Transport: $\frac{\partial S}{\partial t} - \nabla \cdot (\frac{\lambda_w}{\lambda_T} \mathbf{u}_T) = q$

Governing equations

- Scalar case:
  - Pressure: $\mathbf{u}_T = -\lambda_T \mathbf{K} \nabla p\,, \qquad \nabla \cdot \mathbf{u}_T = q$
  - Transport: $\frac{\partial S}{\partial t} - \nabla \cdot (\frac{\lambda_w}{\lambda_T} \mathbf{u}_T) = q$
- Tensor:
  - Pressure: $-\nabla \cdot (\boldsymbol{\Lambda}_T \mathbf{K} \nabla p) = q$
  - Transport: $\frac{\partial S}{\partial t} - \nabla \cdot (\boldsymbol{\Lambda}_w \mathbf{K} \nabla p) = q_w$

Standard pressure discretisation

Apply a control volume method:

- Discretise $-\nabla \cdot (\mathbf{K}\nabla p)$ (preprocessing)
- For each time step: Multiply each edge with an upstream mobility value

Standard pressure discretisation

Apply a control volume method:

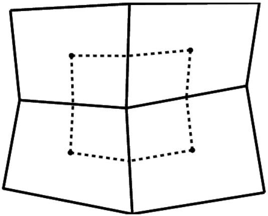
- Discretise $-\nabla \cdot (\mathbf{K}\nabla p)$ (preprocessing)
- For each time step: Multiply each edge with an upstream mobility value

Difficulties:

- The upstream direction is not known - consistency problems
- Brute force approaches (test all upstream directions) expensive
- The principle axes of the tensor $\mathbf{\Lambda K}$ varies with saturation

MPFA

Discretise $-\nabla \cdot (\mathbf{K}\nabla p)$



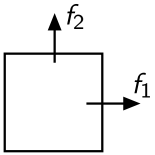
- Introduce a dual grid
- Pressure in cell centres and on edges
- Linear pressure on each subcell
- Edge pressures eliminated

Adjustments to the traditional approach

Flux discretisation:

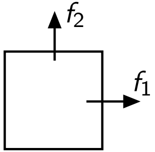
$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \mathbf{Q}\mathbf{\Lambda}\mathbf{K}\mathbf{R}^{-1} \begin{bmatrix} \Delta p_1 \\ \Delta p_2 \end{bmatrix}$$

$\mathbf{Q}$: Normal vectors
$\mathbf{R}^{-1}$: Gradients



Adjustments to the traditional approach

Flux discretisation:



$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \mathbf{Q\Lambda K R}^{-1} \begin{bmatrix} \Delta p_1 \\ \Delta p_2 \end{bmatrix}$$

$\mathbf{Q}$: Normal vectors
$\mathbf{R}^{-1}$: Gradients

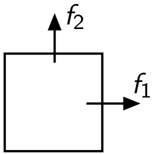
Single phase discretisation:

$$\begin{bmatrix} fs_1 \\ fs_2 \end{bmatrix} = \mathbf{Q K R}^{-1} \begin{bmatrix} \Delta p_1 \\ \Delta p_2 \end{bmatrix}$$

Adjustments to the traditional approach

Flux discretisation:

$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{K}\mathbf{R}^{-1} \begin{bmatrix} \Delta p_1 \\ \Delta p_2 \end{bmatrix}$$



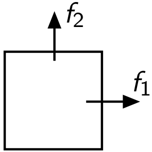
$\mathbf{Q}$: Normal vectors
$\mathbf{R}^{-1}$: Gradients

Introduce a diagonal tensor $\hat{\boldsymbol{\Lambda}} = \mathrm{diag}(\hat{\lambda}_1, \hat{\Lambda}_2)$

$$\begin{bmatrix} \hat{f}_1 \\ \hat{f}_2 \end{bmatrix} = \hat{\boldsymbol{\Lambda}}\mathbf{Q}\mathbf{K}\mathbf{R}^{-1} \begin{bmatrix} \Delta p_1 \\ \Delta p_2 \end{bmatrix}$$

Adjustments to the traditional approach

Flux discretisation:



$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \mathbf{Q\Lambda K R}^{-1} \begin{bmatrix} \Delta p_1 \\ \Delta p_2 \end{bmatrix}$$

Q: Normal vectors
$\mathbf{R}^{-1}$: Gradients

Introduce a diagonal tensor $\hat{\mathbf{\Lambda}} = \mathrm{diag}(\hat{\lambda}_1, \hat{\Lambda}_2)$

$$\begin{bmatrix} \hat{f}_1 \\ \hat{f}_2 \end{bmatrix} = \hat{\mathbf{\Lambda}}\mathbf{QKR}^{-1} \begin{bmatrix} \Delta p_1 \\ \Delta p_2 \end{bmatrix}$$

Write

$$\Lambda_1 = \omega_1 \Lambda_{1,1} + \omega_2 \Lambda_{1,2} + \omega_3 \Lambda_{2,1} + \omega_4 \Lambda_{2,2}$$

Define $\hat{\Lambda}_i$ so that $\|f_i - \hat{f}_i\|$ is minimised.

Adjustments to the traditional approach

ω_i:

- Depends on $\mathbf{Q}$, $\mathbf{R}$, and $\mathbf{K}$ (both grid and permability).
- Independent of saturation

Adjustments to the traditional approach

ω_i:

- Depends on $\mathbf{Q}$, $\mathbf{R}$, and $\mathbf{K}$ (both grid and permability).
- Independent of saturation

Discretisation:

- Computations can be preprocessed
- Very similar to the traditional approach
- Two rel. perm. for each edge
- Convenient for legacy code

Adjustments to the traditional approach

ω_i:

- Depends on **Q**, **R**, and **K** (both grid and permability).
- Independent of saturation

Discretisation:

- Computations can be preprocessed
- Very similar to the traditional approach
- Two rel. perm. for each edge
- Convenient for legacy code

Speculations:

- Equal to traditional approach for scalar rel perm
- Should work for anisotropic tensor aligned with Cartesian grid
- General case: Unknown

Full discretisation

Recall: Pressure discretisation - upstream direction not known

Full discretisation

Recall: Pressure discretisation - upstream direction not known

Suggestion: Discretise the pressure equation (MPFA) with tensor $\mathbf{\Lambda}_T\mathbf{K}$ in each time step.

- More accurate approach (?)
- Computationally expensive

Gives

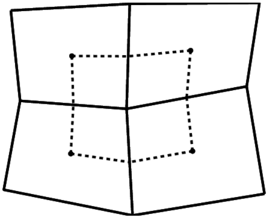
- Pressure field
- Fluxes (harmonically averaged)

pressure field, and fluxes (based on harmonically averaged mobilities).

Full discretisation

We want upstream weighting of mobilities.

Flux over an edge: $-\mathbf{n}^T \mathbf{\Lambda} \mathbf{K} \nabla p$



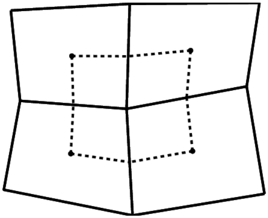
- Construct ∇p based on MPFA (cell and edge pressures)
- One flux for each side of each edge
- If directions are equal: Use upstream flux
- Competing fluxes: Use the largest (Godunov)

Full discretisation

We want upstream weighting of mobilities.

Flux over an edge: $-\mathbf{n}^T \mathbf{\Lambda} \mathbf{K} \nabla p$

- Construct ∇p based on MPFA (cell and edge pressures)
- One flux for each side of each edge
- If directions are equal: Use upstream flux
- Competing fluxes: Use the largest (Godunov)



Corresponds to solving a Riemann problem with discontinuous flux function

Full discretisation

Features:

- Natural extension of upstream scheme
- May be able to handle rotations

Full discretisation

Features:

- Natural extension of upstream scheme
- May be able to handle rotations
- One MPFA discretisation for each time step

Full discretisation

Features:

- Natural extension of upstream scheme
- May be able to handle rotations
- One MPFA discretisation for each time step
- May not work at all

Summary

- Tensor rel perm introduce additional challenges for discretisations
- Two approaches presented:
  - Best possible diagonal tensor
  - Godunov-like method