# An Overview of the Multiscale Mixed Finite-Element Method

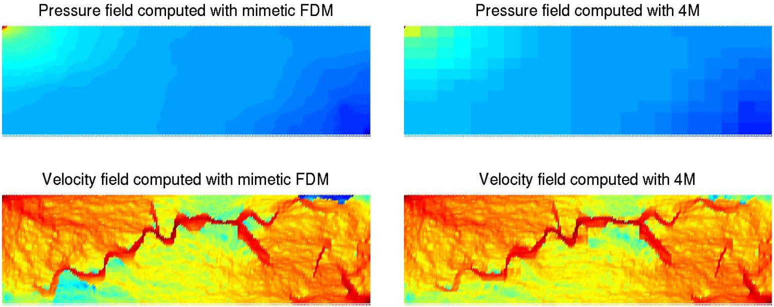
SINTEF ICT, Department of Applied Mathematics

Multiscale Workshop, Dr. Holms, Geilo, Dec 5, 2008

## Basic idea:

- Upscaling and downscaling in one step
- Pressure on coarse grid (subresolution near wells)
- Velocity with subgrid resolution everywhere

## Example: Layer 36 from SPE 10

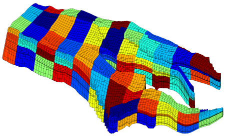


## Multiscale mixed finite elements

Developed by SINTEF

Main focus on complex grids

- Corner-point grids in 3D
- Triangular/nonuniform/PEBI
- Automated coarsening



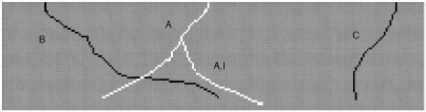
+ Stokes–Brinkman, wells, black-oil
Applications: history match, optimization

## Multiscale finite volumes

Developed by Jenny/Lee/Tchelepi/..

Focus on flow physics

- Gravity and capillarity
- Black-oil
- Compressibility
- Complex wells

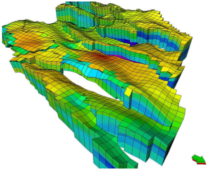


Only for Cartesian grids, so far.

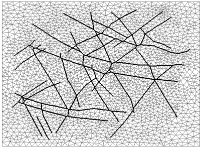
**Challenges:**

- Industry-standard grids are often nonconforming and contain skewed and degenerate cells
- There is a trend towards unstructured grids
- Standard discretization methods produce wrong results on skewed and rough cells
- The combination of high aspect and anisotropy ratios can give *very large* condition numbers

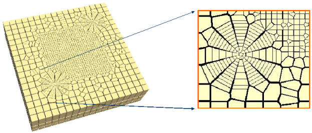
Corner point:



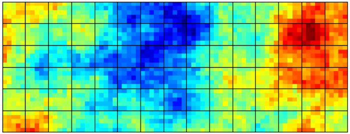
Tetrahedral:



PEBI:

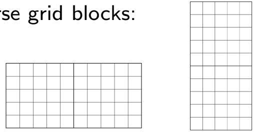


**Standard upscaling:**



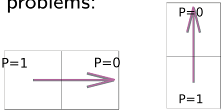
⇓

Coarse grid blocks:

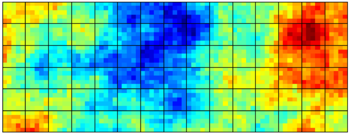


⇓

Flow problems:

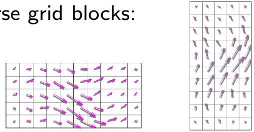


**Standard upscaling:**



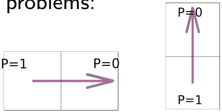
⇓

Coarse grid blocks:



⇓ ⇑

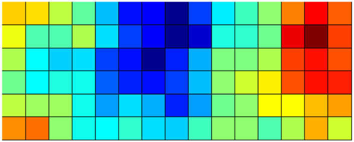
Flow problems:



# The MsMFE Method in a Nutshell

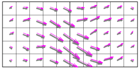
## From upscaling to multiscale methods

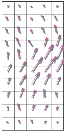
**Standard upscaling:**



$\Downarrow \quad \Uparrow$

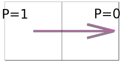
Coarse grid blocks:

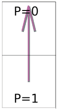




$\Downarrow \quad \Uparrow$

Flow problems:

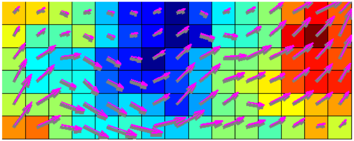


# The MsMFE Method in a Nutshell

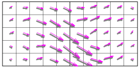
## From upscaling to multiscale methods

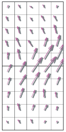
**Standard upscaling:**



⇓ ⇑

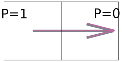
Coarse grid blocks:

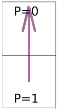




⇓ ⇑

Flow problems:

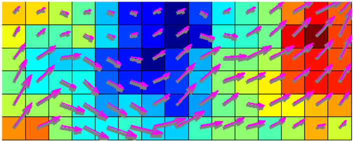


# The MsMFE Method in a Nutshell

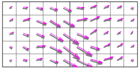
## From upscaling to multiscale methods

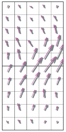
**Standard upscaling:**



⇓ ⇑

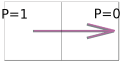
Coarse grid blocks:

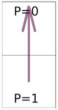




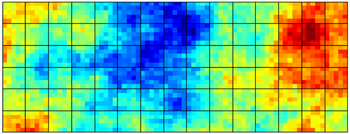
⇓ ⇑

Flow problems:



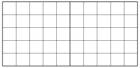



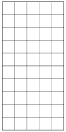

**Multiscale method:**



⇓

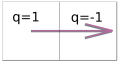
Coarse grid blocks:

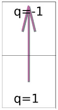




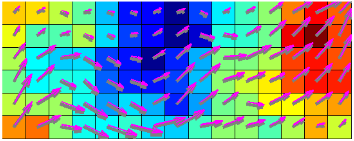
⇓

Flow problems:



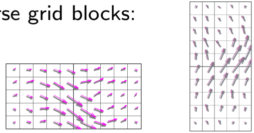



**Standard upscaling:**



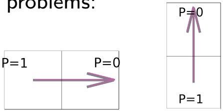
⇓ ⇑

Coarse grid blocks:

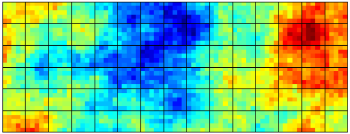


⇓ ⇑

Flow problems:

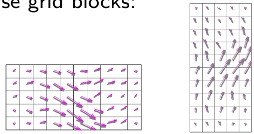



**Multiscale method:**



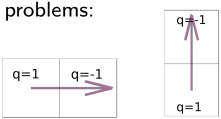
⇓

Coarse grid blocks:



⇓ ⇑

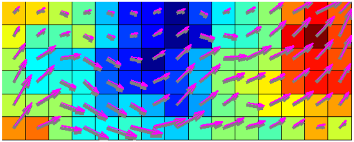
Flow problems:



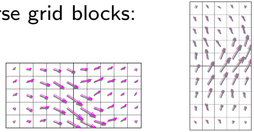
# The MsMFE Method in a Nutshell

## From upscaling to multiscale methods

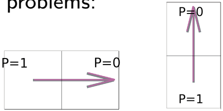
### Standard upscaling:



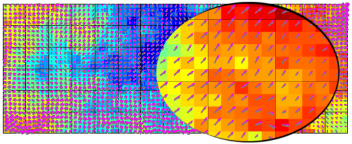
Coarse grid blocks:



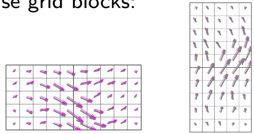
Flow problems:



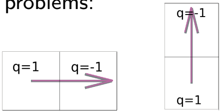

### Multiscale method:



Coarse grid blocks:



Flow problems:



## Mixed formulation:

Find $(v, p) \in H_0^{1,\text{div}} \times L^2$ such that

$$\int (\lambda K)^{-1} u \cdot v \, dx - \int p \nabla \cdot u \, dx = 0, \qquad \forall u \in H_0^{1,\text{div}},$$
$$\int \ell \nabla \cdot v \, dx = \int q \ell \, dx, \quad \forall \ell \in L^2.$$

## Multiscale discretization:

Seek solutions in low-dimensional subspaces in which local fine-scale properties are incorporated into the basis functions

# The MsMFE Method in a Nutshell

## Linear system and basis functions

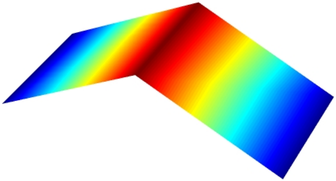
Discretisation matrices:

$$\begin{pmatrix} B & C \\ C^T & 0 \end{pmatrix} \begin{pmatrix} v \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix},$$

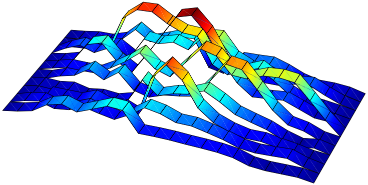
$$b_{ij} = \int_\Omega \psi_i (\lambda K)^{-1} \psi_j \, dx,$$

$$c_{ik} = \int_\Omega \phi_k \nabla \cdot \psi_i \, dx$$

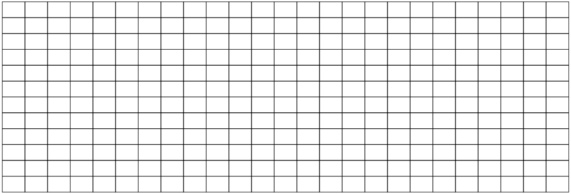
Raviart–Thomas:



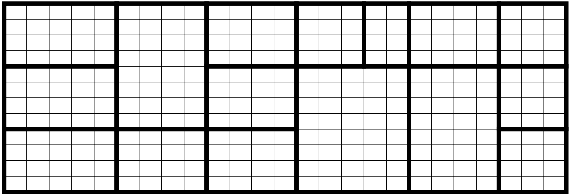
Multiscale basis function:



We assume we are given a *fine* grid with permeability and porosity attached to each fine-grid block.

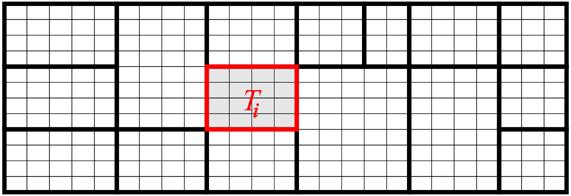


We assume we are given a *fine* grid with permeability and porosity attached to each fine-grid block.



We construct a *coarse* grid, and choose the discretisation spaces $V$ and $U^{ms}$ such that:

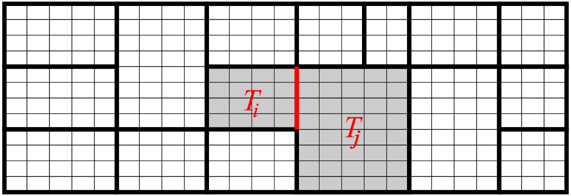
We assume we are given a *fine* grid with permeability and porosity attached to each fine-grid block.




We construct a *coarse* grid, and choose the discretisation spaces $V$ and $U^{ms}$ such that:

- For each coarse block $T_i$, there is a basis function $\phi_i \in V$.

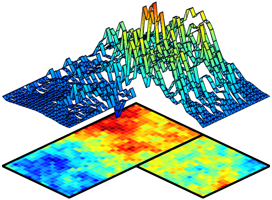
We assume we are given a *fine* grid with permeability and porosity attached to each fine-grid block.




We construct a *coarse* grid, and choose the discretisation spaces $V$ and $U^{ms}$ such that:

- For each coarse block $T_i$, there is a basis function $\phi_i \in V$.
- For each coarse edge $\Gamma_{ij}$, there is a basis function $\psi_{ij} \in U^{ms}$.

For each coarse edge $\Gamma_{ij}$, define a basis function with unit flux through $\Gamma_{ij}$ and no flow across $\partial(T_i \cup T_j)$.



## Local flow problem:

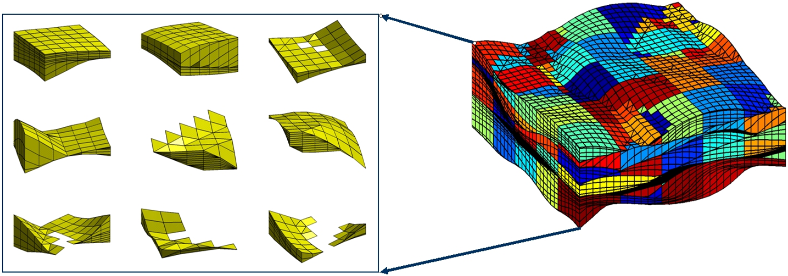
$$\psi_{ij} = -\lambda K \nabla \phi_{ij}, \qquad \nabla \cdot \psi_{ij} = \begin{cases} w_i(x), & \text{for } x \in T_i, \\ -w_j(x), & \text{for } x \in T_j, \end{cases}$$

with boundary conditions $\psi_{ij} \cdot n = 0$ on $\partial(T_i \cup T_j)$.

## Global velocity:

$v = \sum_{ij} v_{ij} \psi_{ij}$, where $v_{ij}$ are (coarse-scale) coefficients.

The MsMFE method allows fully automated coarse gridding strategies: grid blocks need to be connected, but can have arbitrary shapes

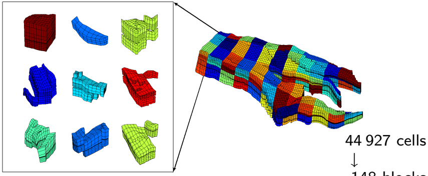


Corner-point grids: the coarse blocks are logically Cartesian in index space

# The MsMFE Method in a Nutshell

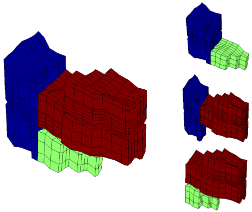
## Workflow with automated upgridding in 3D

1) Coarsen grid by uniform partitioning in index space for corner-point grids



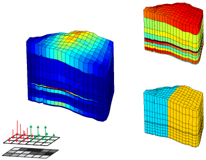

9 different coarse blocks

2) Detect all adjacent blocks



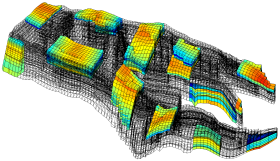
3) Compute basis functions

$$\nabla\cdot\psi_{ij} = \begin{cases} w_i(x), \\ -w_j(x), \end{cases}$$



for all pairs of blocks

4) Block in coarse grid: component for building global solution

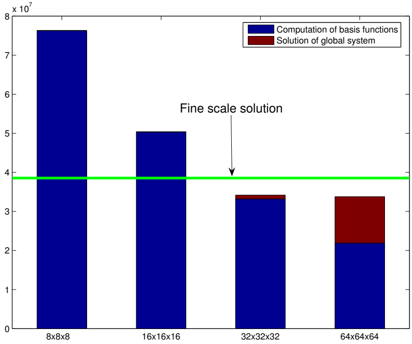


Multigrid more efficient when computing pressure once.
Why bother with multiscale pressure solvers?

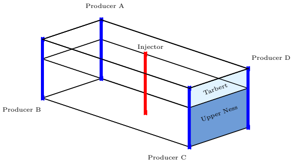
- Full simulation: $\mathcal{O}(10^2)$ time steps.
- Basis functions need not be recomputed

Also:

- Possible to solve very large problems
- Easy parallelization



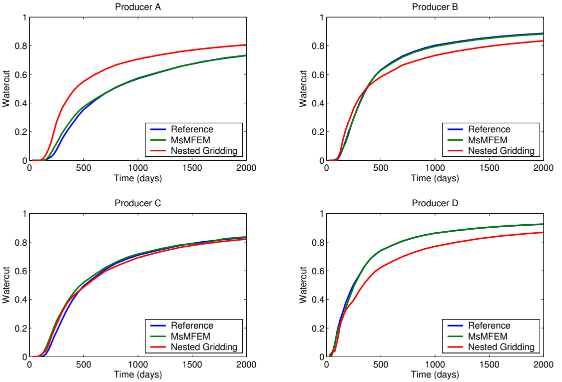
SPE 10, Model 2:




Fine grid: $60 \times 220 \times 85$
Coarse grid: $5 \times 11 \times 17$
2000 days production

4M + streamlines:
2 min 22 sec on 2.4 GHz desktop PC

Water-cut curves at the four producers




upscaling/downscaling, 4M/streamlines, fine grid

# Implementation Details for MsMFE

## There are certain choices....

- Choice of weighting function in definition of basis functions
- Boundary conditions (overlap and global information)
- Assembly of linear system
- Fine-grid discretization
- Generation of coarse grids

## Interpretation of the weight function:

$$(\nabla \cdot v)|_{T_i} = \sum_j w_i \nabla \cdot (v_{ij}\psi_{ij}) = w_i \sum_j v_{ij}$$
$$= w_i \int_{\partial_{T_i}} v \cdot n ds = w_i \int_{T_i} \nabla \cdot v$$

That is, $w_i$ distributes $\nabla \cdot v$ among the cells in the coarse grid

## Different roles:

Incompressible flow: $\nabla \cdot v = q$

Compressible flow: $\nabla \cdot v = q - c_t \partial_t p - \sum_j c_j v_j \cdot \nabla p$

For incompressible flow, we have that

$$(\nabla \cdot v)|_{T_i} = w_i \sum_j v_{ij}, \qquad \sum_j v_{ij} = \begin{cases} 0, & \text{if } \int_{T_i} q\,dx = 0, \\ \int_{T_i} q\,dx, & \text{otherwise} \end{cases}$$

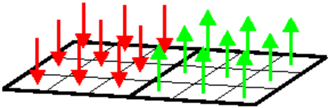
Thus

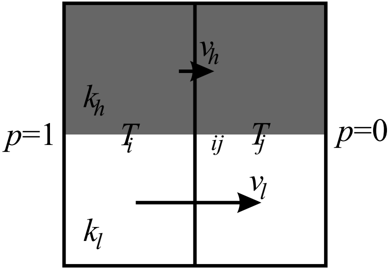
$$\int_{T_i} q\,dx = 0 \quad \Rightarrow \quad \nabla \cdot v = 0, \quad \forall w_i > 0$$

$$\int_{T_i} q\,dx \neq 0 \quad \Rightarrow \quad \nabla \cdot v = q, \quad \text{if } w_i = \frac{q}{\int_{T_i} q\,dx}$$

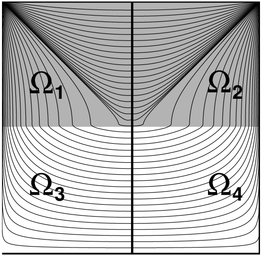
Uniform source:

$$w_i(x) = \frac{1}{|T_i|}$$





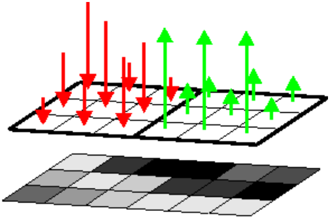

low ($k_l$) and high ($k_h$) permeability

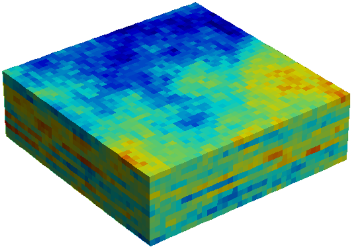


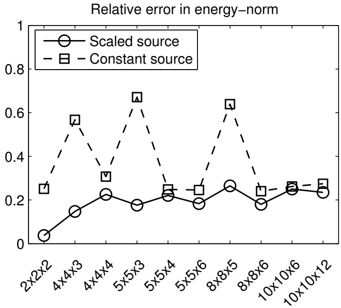

streamlines from basis function

Scaled source:

$$w_i(x) = \frac{\text{trace}(K(x))}{\int_{T_i} \text{trace}(K(\xi))\, d\xi}$$







Compressible flow:

$$(\nabla \cdot v)|_{T_i} = w_i \sum_j v_{ij},$$

$$\sum_j v_{ij} = \int_{T_i} \Big( q - c_t \frac{\partial p}{\partial t} + \sum c_\alpha v_\alpha \cdot \nabla p \Big) dx$$

Ideas from incompressible flow do not apply directly:

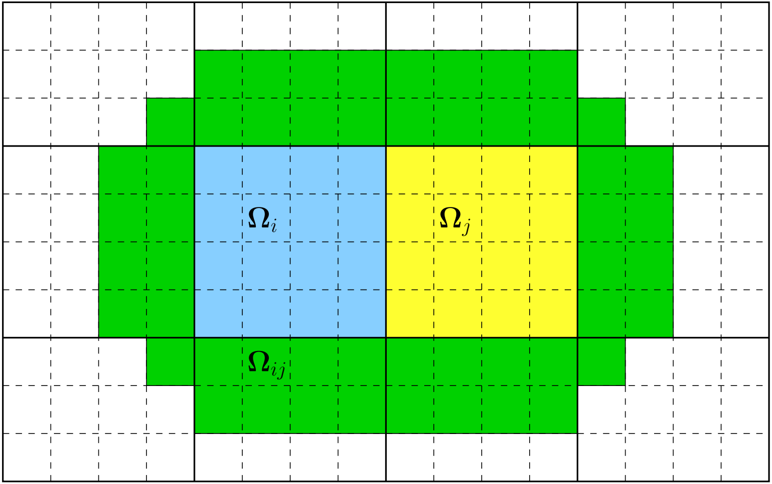
- $w_i \propto q$ concentrates compressibility effects where $q \neq 0$
- $w_i \propto K$ overestimates $\nabla \cdot v$ in high-permeable zones and underestimates in low-permeable zones

Better choice:

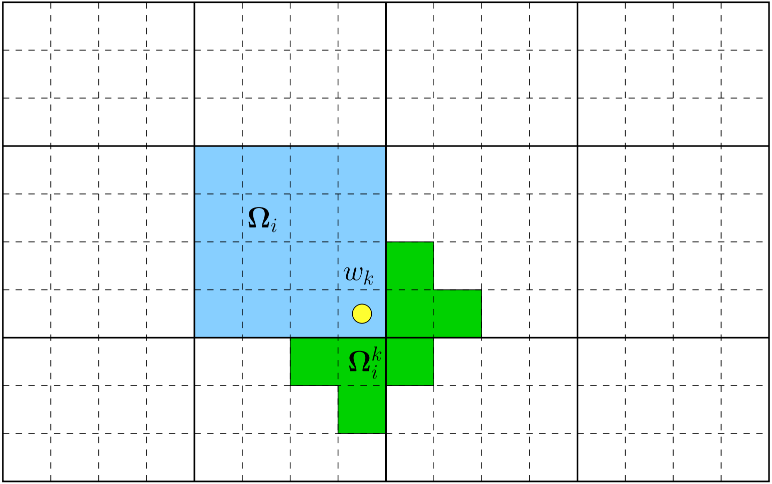
$w_i = \frac{\phi}{\int_{T_i} \phi dx}$ Motivation: $c_t \partial_t p \propto \phi$

# Domain of Support Basis Functions

## Here with overlap (green region)

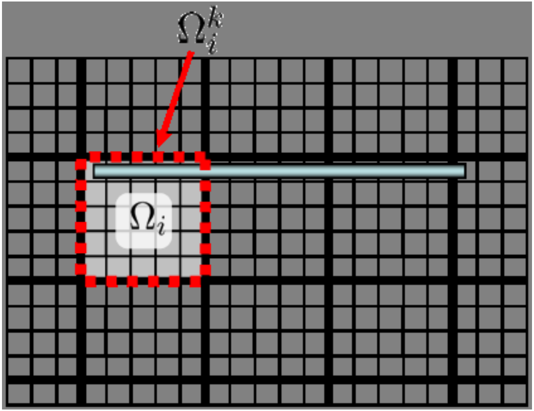


# Domain of Support Basis Functions
## Here with overlap (green region)



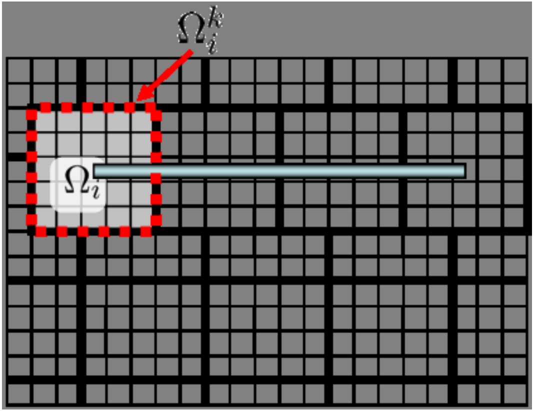
## Strategy

**Standard**: Use initial partitioning as is



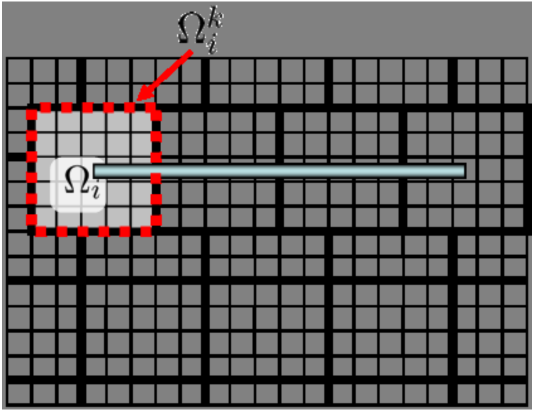
## Strategy

**Adapted**: Initial partition altered to put wells near block center



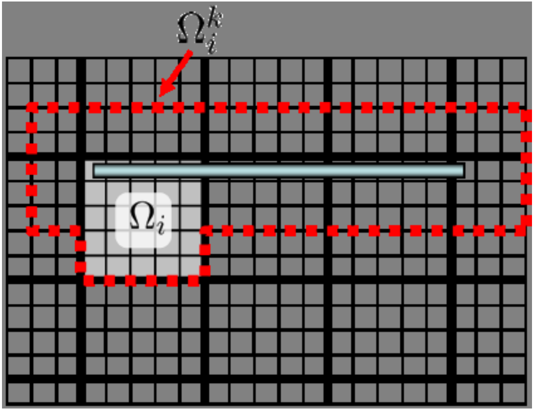
## Strategy

**Refined**: Altered partition further sub-divided near wells



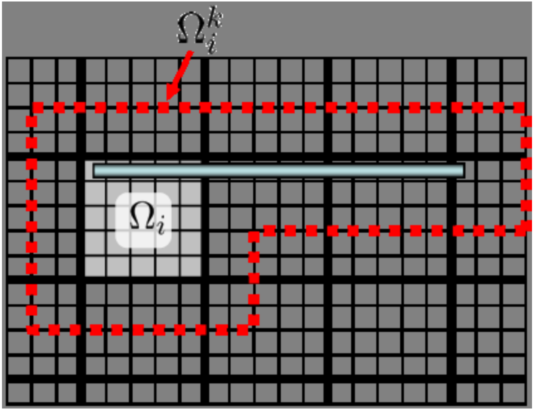
### Strategy

**Well oversampling**: Support domain for well/block enlarged to include additional cells about well trajectory



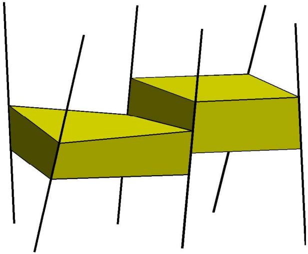
### Strategy

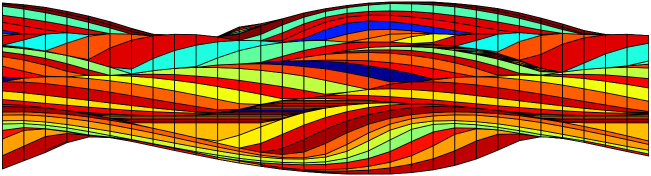
**Well & block oversampling**: Well oversampling + inclusion of additional cells about coarse blocks



Corner-point grids:

- areal 2D mesh of vertical or inclined pillars
- each volumetric cell is restriced by four pillars
- each cell is defined by eight corner points, two on each pillar

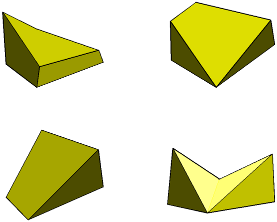




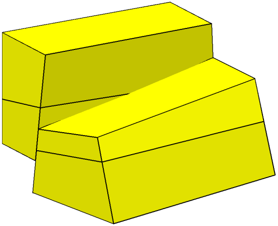
# Implementation Details for MsMFE

## Cell geometries are challenging from a discretization point-of-view

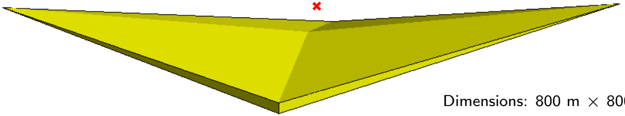
Skewed and deformed grid blocks:



Non-matching cells:



Very high aspect ratios (and centroid outside the cell):



Dimensions: 800 m × 800 m × 0.25 m

Mimetic finite-difference methods may be interpreted as a finite-volume counterpart of mixed finite-element methods.

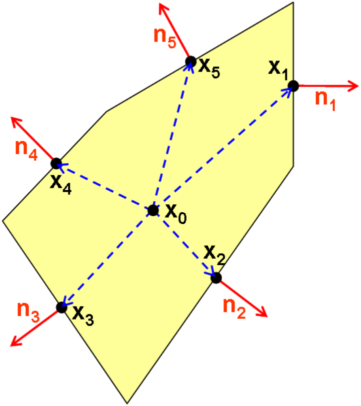
**Key features:**

- Applicable for models with general polyhedral grid-cells.
- Allow easy treatment of non-conforming grids with complex grid-cell geometries (including curved faces).
- Generic implementation: same code applies to all grids (e.g., corner-point/PEBI, matching/non-matching, ...).

Express fluxes $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)^\mathsf{T}$ as:

$$\boldsymbol{v} = -\boldsymbol{T}(\boldsymbol{p} - p_0),$$

where $\boldsymbol{p} = (p_1, p_2, \ldots, p_n)^\mathsf{T}$.



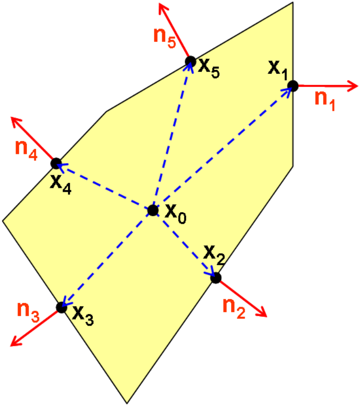
Express fluxes $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)^\mathsf{T}$ as:

$$\boldsymbol{v} = -\boldsymbol{T}(\boldsymbol{p} - p_0),$$

where $\boldsymbol{p} = (p_1, p_2, \ldots, p_n)^\mathsf{T}$.
Impose exactness for any *linear* pressure field $p = \boldsymbol{x}^\mathsf{T}\boldsymbol{a} + c$ (which gives velocity equal to $-\mathbf{K}\boldsymbol{a}$):

$$v_i = -A_i \boldsymbol{n}_i^\mathsf{T} \mathbf{K}\boldsymbol{a}$$

$$p_i - p_0 = (\boldsymbol{x}_i - \boldsymbol{x}_0)^\mathsf{T}\boldsymbol{a}.$$



# Implementation Details for MsMFE

## The mimetic finite difference method, Brezzi *et al.*, 2005

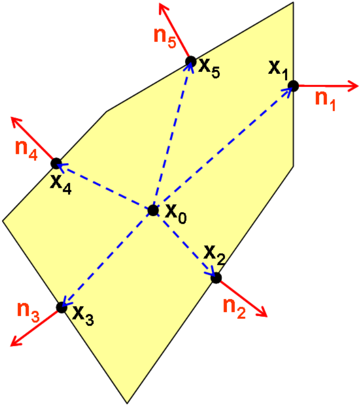
Express fluxes $\boldsymbol{v} = (v_1, v_2, \dots, v_n)^\mathsf{T}$ as:

$$\boldsymbol{v} = -\boldsymbol{T}(\boldsymbol{p} - p_0),$$

where $\boldsymbol{p} = (p_1, p_2, \dots, p_n)^\mathsf{T}$.
Impose exactness for any *linear* pressure field $p = \boldsymbol{x}^\mathsf{T}\boldsymbol{a} + c$ (which gives velocity equal to $-\mathbf{K}\boldsymbol{a}$):

$$v_i = -A_i \boldsymbol{n}_i^\mathsf{T} \mathbf{K} \boldsymbol{a}$$

$$p_i - p_0 = (\boldsymbol{x}_i - \boldsymbol{x}_0)^\mathsf{T} \boldsymbol{a}.$$




As a result, $\mathbf{T}$ must satisfy

$$\mathbf{T} \times \mathbf{C} = \mathbf{N} \times \mathbf{K}$$

where $\boldsymbol{C}(i,:) = (\boldsymbol{x}_i - \boldsymbol{x}_0)^\mathsf{T}$ and $\boldsymbol{N}(i,:) = A_i \boldsymbol{n}_i^\mathsf{T}$

# Implementation Details for MsMFE

## The mimetic finite difference method, Brezzi *et al.*, 2005

Express fluxes $\boldsymbol{v} = (v_1, v_2, \dots, v_n)^\mathsf{T}$ as:

$$\boldsymbol{v} = -\boldsymbol{T}(\boldsymbol{p} - p_0),$$

where $\boldsymbol{p} = (p_1, p_2, \dots, p_n)^\mathsf{T}$.
Impose exactness for any *linear* pressure field $p = \boldsymbol{x}^\mathsf{T}\boldsymbol{a} + c$ (which gives velocity equal to $-\mathbf{K}\boldsymbol{a}$):

$$v_i = -A_i \boldsymbol{n}_i^\mathsf{T} \mathbf{K} \boldsymbol{a}$$

$$p_i - p_0 = (\boldsymbol{x}_i - \boldsymbol{x}_0)^\mathsf{T} \boldsymbol{a}.$$

As a result, $\mathbf{T}$ must satisfy

$$\mathbf{T} \times \mathbf{C} = \mathbf{N} \times \mathbf{K}$$

where $\boldsymbol{C}(i,:) = (\boldsymbol{x}_i - \boldsymbol{x}_0)^\mathsf{T}$ and $\boldsymbol{N}(i,:) = A_i \boldsymbol{n}_i^\mathsf{T}$

Family of valid solutions:

$$\boldsymbol{T} = \frac{1}{|E|} \boldsymbol{N} \mathbf{K} \boldsymbol{N}^\mathsf{T} + \boldsymbol{T}_2,$$

where $\boldsymbol{T}_2$ is such that $\boldsymbol{T}$ is s.p.d. and $\boldsymbol{T}_2 \boldsymbol{C} = \boldsymbol{O}$.

# Implementation Details for MsMFE

## The mimetic finite difference method, Brezzi *et al.*, 2005

Express fluxes $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)^\mathsf{T}$ as:

$$\boldsymbol{v} = -\boldsymbol{T}(\boldsymbol{p} - p_0),$$

where $\boldsymbol{p} = (p_1, p_2, \ldots, p_n)^\mathsf{T}$.
Impose exactness for any *linear* pressure field $p = \boldsymbol{x}^\mathsf{T}\boldsymbol{a} + c$ (which gives velocity equal to $-\mathbf{K}\boldsymbol{a}$):

$$v_i = -A_i \boldsymbol{n}_i^\mathsf{T}\mathbf{K}\boldsymbol{a}$$

$$p_i - p_0 = (\boldsymbol{x}_i - \boldsymbol{x}_0)^\mathsf{T}\boldsymbol{a}.$$

As a result, $\mathbf{T}$ must satisfy

$$\mathbf{T} \times \mathbf{C} = \mathbf{N} \times \mathbf{K}$$

where $\boldsymbol{C}(i,:) = (\boldsymbol{x}_i - \boldsymbol{x}_0)^\mathsf{T}$ and $\boldsymbol{N}(i,:) = A_i\boldsymbol{n}_i^\mathsf{T}$

Family of valid solutions:

$$\boldsymbol{T} = \frac{1}{|E|}\boldsymbol{N}\mathbf{K}\boldsymbol{N}^\mathsf{T} + \boldsymbol{T}_2,$$

where $\boldsymbol{T}_2$ is such that $\boldsymbol{T}$ is s.p.d. and $\boldsymbol{T}_2\boldsymbol{C} = \boldsymbol{O}$.

Imposing continuity across edges/faces and conservation yields a *hybrid* system:

$$\begin{pmatrix} \boldsymbol{B} & \boldsymbol{C} & \boldsymbol{D} \\ \boldsymbol{C}^\mathsf{T} & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{D}^\mathsf{T} & \boldsymbol{O} & \boldsymbol{O} \end{pmatrix} \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{p} \\ \boldsymbol{\pi} \end{pmatrix} = \text{RHS}$$

$$\Downarrow$$

Reduces to s.p.d. system for face pressures $\boldsymbol{\pi}$.

# Treating Wells as Boundary Conditions

Discrete system still amenable to schur complement reduction

## Discrete Pressure System

$$\begin{pmatrix} \boldsymbol{B} & \mathbf{0} & \boldsymbol{C} & \boldsymbol{D} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{B}_w & \boldsymbol{C}_w & \mathbf{0} & \boldsymbol{D}_w \\ \boldsymbol{C}^\mathsf{T} & \boldsymbol{C}_w^\mathsf{T} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{D}^\mathsf{T} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{D}_w^\mathsf{T} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{v} \\ -\boldsymbol{q}_w \\ -\boldsymbol{p} \\ \boldsymbol{\pi} \\ \boldsymbol{p}_w \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ -\boldsymbol{q}_{w,\text{tot}} \end{pmatrix}$$

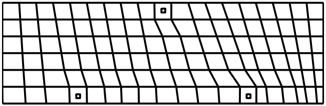
## Well Model, Peaceman

$$-q_i^k = -\lambda_t(s_{k_i})\, WI_i^k (p_{E_{k_i}} - p_{w_k}), \quad i = 1, \ldots, n_k$$

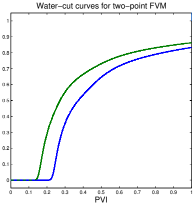
$$q_{\text{tot}}^k = \sum_{i=1}^{n_k} q_i^k.$$

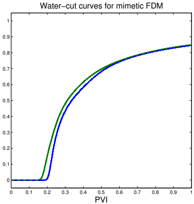
# Implementation Details for MsMFE
## Mimetic: method applicable to general polyhedral cells

Standard method $+$ skew grids $=$ grid-orientation effects

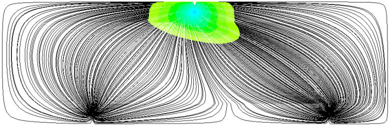


$\mathbf{K}$: homogeneous and isotropic, symmetric well pattern
$\longrightarrow$ symmteric flow

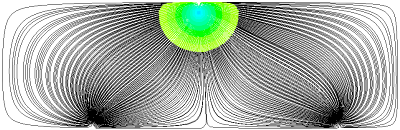



Streamlines with two-point method



Streamlines with mimetic method



There is freedom in choosing the inner product ($\mathbf{T}_2$), so that e.g.,

- MFDM coincides with TPFA on Cartesian grids
- MFDM coincides with MFEM on Cartesian grids

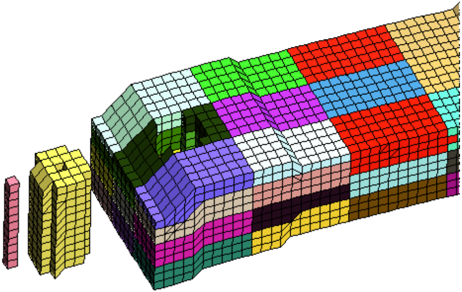
Positive definite system is guaranteed. Monotonicity properties are similar as for MPFA.

**Challenge:**

Local adjustment of the inner product to reduce the condition number (and appearance of cycles) on complex grids.

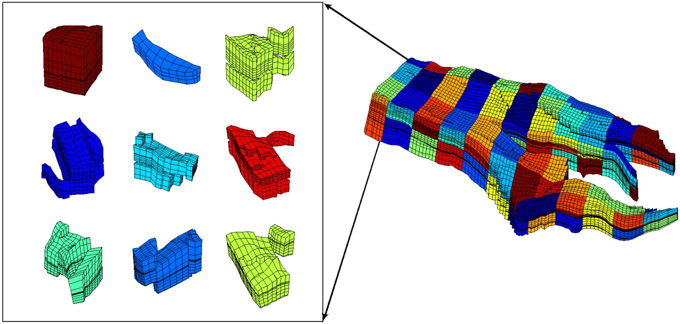
**(Unique) grid flexibility:**

Given a method that can solve local flow problems on the subgrid, the MsMFE method can be formulated on any coarse grid in which the coarse blocks consist of a connected collection of fine-grid cells



**(Unique) grid flexibility:**

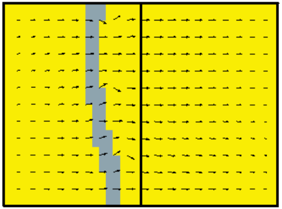
Given a method that can solve local flow problems on the subgrid, the MsMFE method can be formulated on any coarse grid in which the coarse blocks consist of a connected collection of fine-grid cells



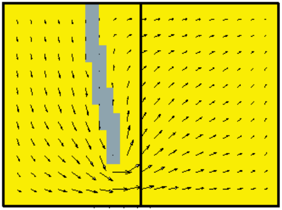
# Implementation Details for MsMFE

## Coarse grid generation

Problems occur when a basis function tries to force flow through a flow barrier



problem



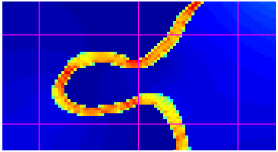
no problem

Can be detected automatically through the indicator

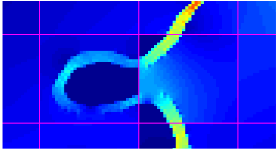
$$v_{ij} = \psi_{ij} \cdot (\lambda K)^{-1} \psi_{ij}$$

If $v_{ij}(x) > C$ for some $x \in T_i$, then split $T_i$ and generate basis functions for the new faces

Problems if there is a strong bi-directional flow over a coarse-grid interface



fine grid



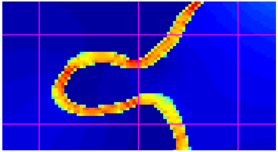
multiscale

Can be detected automatically through the indicator

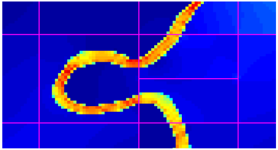
$$\left|\int_{\Gamma_{ij}} v\cdot n\,ds\right| \ll \int_{\Gamma_{ij}} |v\cdot n|\,ds, \qquad c \le \int_{\Gamma_{ij}} |v\cdot n|\,ds$$

If so, split $T_i$ and generate basis functions for the new faces.

Problems if there is a strong bi-directional flow over a coarse-grid interface



fine grid



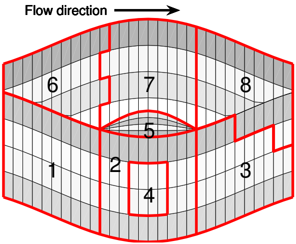
multiscale

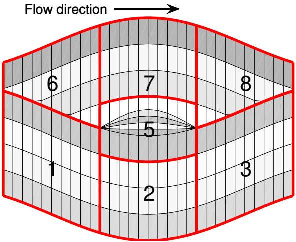
Can be detected automatically through the indicator

$$\left|\int_{\Gamma_{ij}} v\cdot n\, ds\right| \ll \int_{\Gamma_{ij}} |v\cdot n|\, ds, \qquad c \le \int_{\Gamma_{ij}} |v\cdot n|\, ds$$

If so, split $T_i$ and generate basis functions for the new faces.

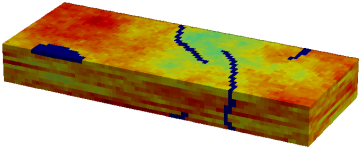
1. Minimize bidirectional flow over interfaces:
   - Avoid unnecessary irregularity ($\Gamma_{6,7}$ and $\Gamma_{3,8}$)
   - Avoid single neighbors ($T_4$)
   - Ensure that there are faces transverse to flow direction ($T_5$)
2. Blocks and faces should follow geological layers ($T_3$ and $T_8$)
3. Blocks should adapt to flow obstacles whenever possible
4. For efficiency: minimize the number of connections
5. Avoid having too many small blocks

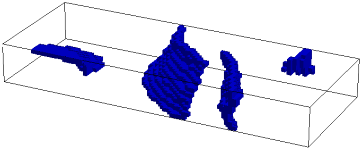


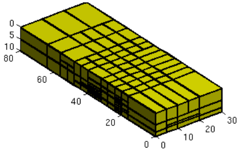
# Implementation Details for MsMFE

## Example: adaption to flow obstacles

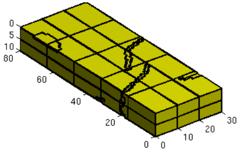




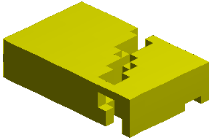
Non-uniform grid, hexahedral cells



Non-uniform grid, general cells



General grid-cell



Four new developments in the last year:

- Extension of the MsMFE method to compressible three-phase flow
- A prototype implementation in FrontSim, applied to fractured media
- Extension of the MsMFE method to the Stokes-Brinkman equations to model flow in vuggy and naturally-fractured porous media
- Combination of the MsMFE method and the flow-based nonuniform coarsening method to give a very efficient solver

## Semi-discrete pressure equation

$$c_t \frac{p_\nu^n - p^{n-1}}{\Delta t} + \nabla \cdot \vec{u}_\nu^n - \zeta_{\nu-1}^n \vec{u}_{\nu-1}^n \cdot \mathsf{K}^{-1} \vec{u}_\nu^n = q, \quad \vec{u}_\nu^n = -\mathsf{K}\lambda \nabla p_\nu^n$$

## Discretization using a mimetic method

$$\boldsymbol{u}_E = \lambda \boldsymbol{T}_E(p_E - \boldsymbol{\pi}_E), \quad \boldsymbol{T}_E = |E|^{-1} \boldsymbol{N}_E \mathsf{K}_E \boldsymbol{N}_E^\mathsf{T} + \tilde{\boldsymbol{T}}_E$$

$\boldsymbol{N}_E$: face normals, $\boldsymbol{X}_E$: vector from face to cell centroids,
$\tilde{\boldsymbol{T}}_E$ chosen arbitrarily provided $\tilde{\boldsymbol{T}}_E \boldsymbol{X}_E = 0$.

## Hybrid system:

$$\begin{bmatrix} \boldsymbol{B} & \boldsymbol{C} & \boldsymbol{D} \\ \boldsymbol{C}^\mathsf{T} - \boldsymbol{V}_{\nu-1}^\mathsf{T} & \boldsymbol{P} & \boldsymbol{0} \\ \boldsymbol{D}^\mathsf{T} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_\nu \\ -\boldsymbol{p}_\nu \\ \boldsymbol{\pi}_\nu \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{P}\boldsymbol{p}^{n-1} + \boldsymbol{q} \\ \boldsymbol{0} \end{bmatrix},$$

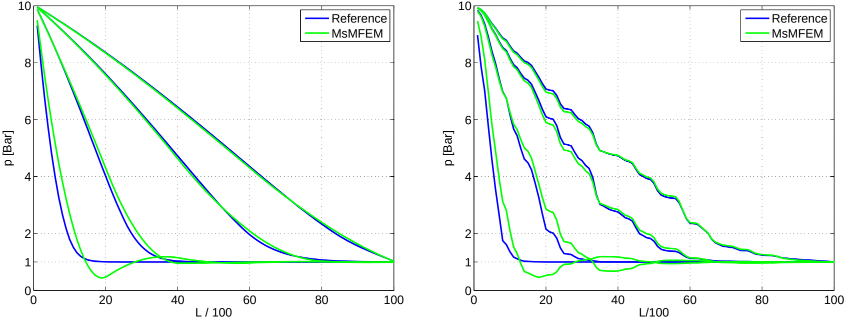
$$\begin{bmatrix} \boldsymbol{\Psi}^\mathsf{T} B_f \boldsymbol{\Psi} & \boldsymbol{\Psi}^\mathsf{T} C_f \boldsymbol{\mathcal{I}} & \boldsymbol{\Psi}^\mathsf{T} D_f \boldsymbol{\mathcal{J}} \\ \boldsymbol{\mathcal{I}}^\mathsf{T} (C_f - V_f)^\mathsf{T} \boldsymbol{\Psi} & \boldsymbol{\mathcal{I}}^\mathsf{T} P_f \boldsymbol{\mathcal{I}} & \mathbf{0} \\ \boldsymbol{\mathcal{J}}^\mathsf{T} D_f^\mathsf{T} \boldsymbol{\Psi} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ -\boldsymbol{p} \\ \boldsymbol{\pi} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\mathcal{I}}^\mathsf{T} P_f \boldsymbol{p}_f^n \\ \mathbf{0} \end{bmatrix}$$

- $\boldsymbol{\Psi}$ – velocity basis functions
- $\boldsymbol{\Phi}$ – pressure basis functions
- $\boldsymbol{\mathcal{I}}$ – prolongation from blocks to cells
- $\boldsymbol{\mathcal{J}}$ – prolongation from block faces to cell faces

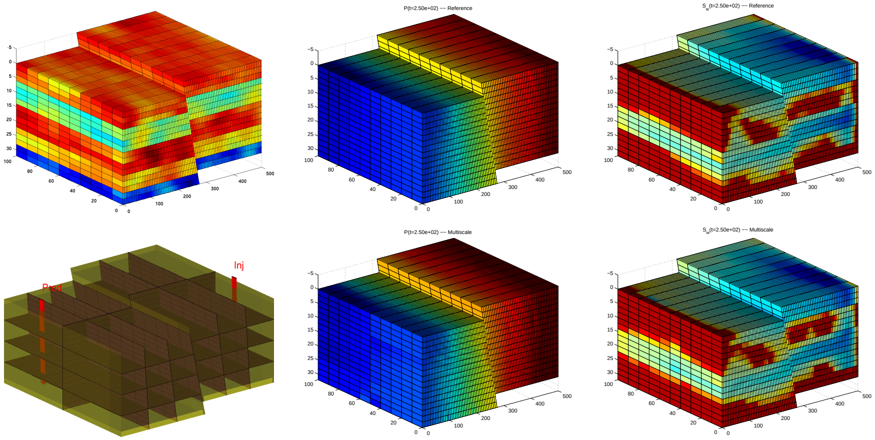
New feature: fine-scale pressure

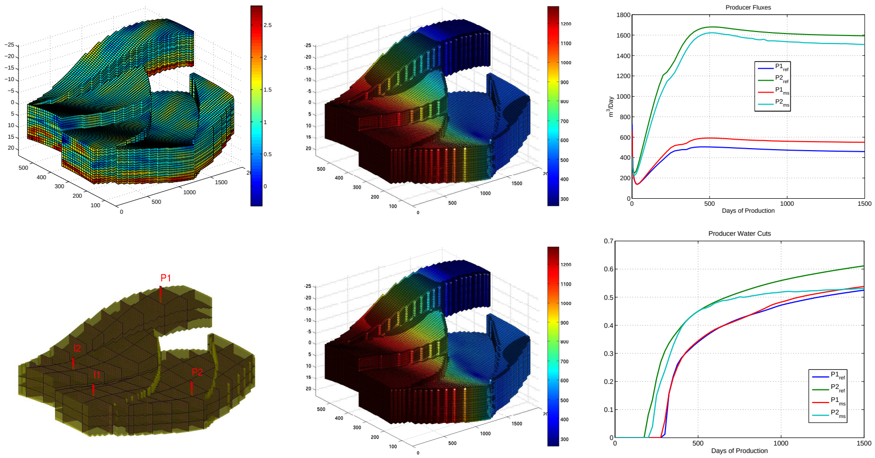
$$\boldsymbol{p}^f \approx \boldsymbol{\mathcal{I}} \boldsymbol{p} + \boldsymbol{\Phi} D_\lambda \boldsymbol{u}, \qquad D_\lambda = \mathrm{diag}(\lambda_i^0/\lambda_i)$$

# MsMFE for Compressible Black-Oil Models
## Example 1: tracer transport in gas (Lunati&Jenny 2006)



# MsMFE for Compressible Black-Oil Models
## Example 2: block with a single fault

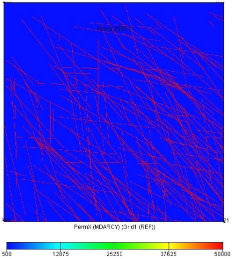


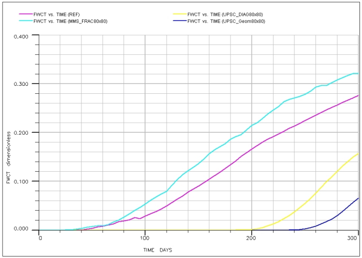


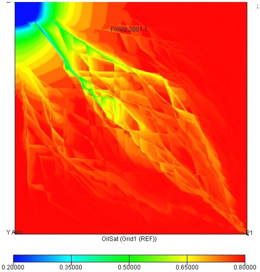
P1
I2
I1
P2
Producer Fluxes
m$^3$/Day
Days of Production
P1 ref
P2 ref
P1 ms
P2 ms
Producer Water Cuts
Days of Production

# MsMFE Prototype Solver in FrontSim

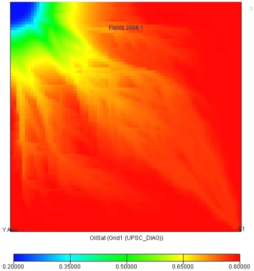
## Example: a dense system of fracture corridors



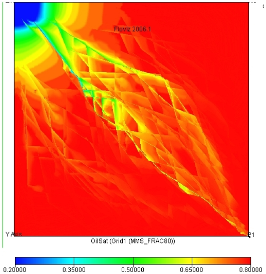




800 × 800



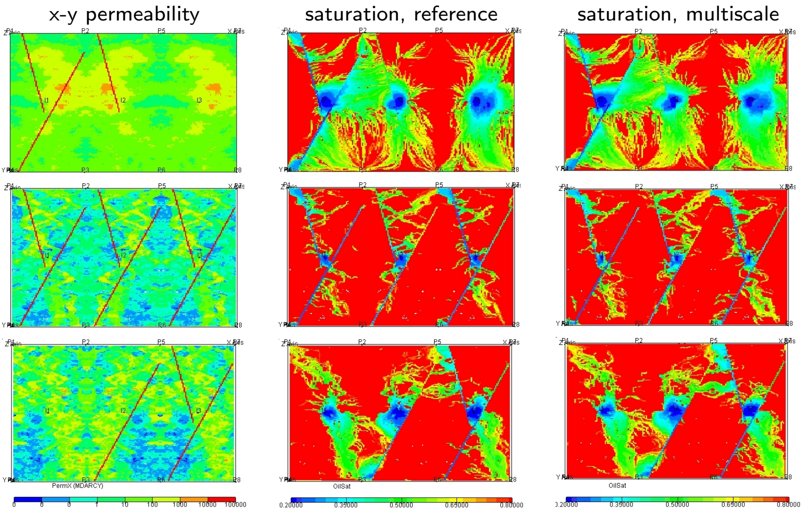

80 × 80 upscaled




80 × 80 multiscale

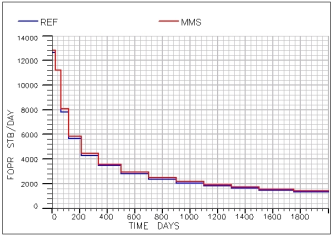
# MsMFE Prototype Solver in FrontSim

## Example: SPE 10 with fracture corridors

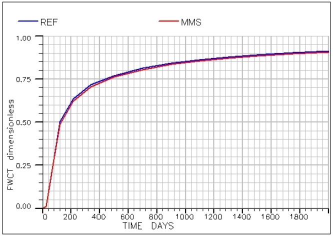


# MsMFE Prototype Solver in FrontSim

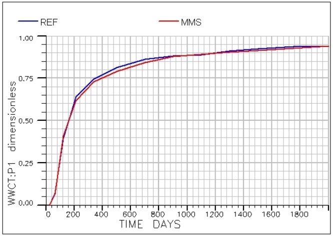
## Example: SPE 10 with fracture corridors



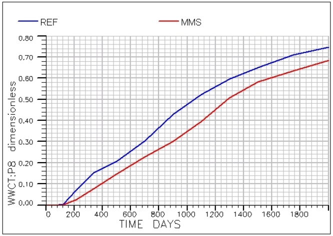

field oil-production rate




field water cut




water cut in P1




water cut in P8

## Standard approach:

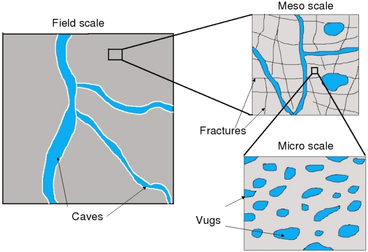
Porous region (Darcy):

$$\mu \mathbf{K}^{-1}\vec{u}_D + \nabla p_D = \vec{f}, \quad \nabla \cdot \vec{u}_D = q.$$

Free-flow region (Stokes):

$$-\mu\nabla\cdot(\nabla\vec{u}_S+\nabla\vec{u}_S^{\mathsf{T}})+\nabla p_S = \vec{f}, \quad \nabla\cdot\vec{u}_S = q$$

Problem: requires interface conditions and explicit geometry



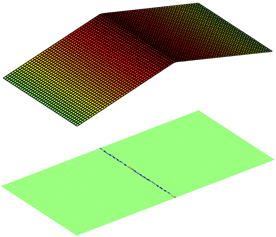

### Stokes–Brinkman (following Popov et al.)

$$\mu \mathbf{K}^{-1}\vec{u} + \nabla p - \tilde{\mu}\Delta\vec{u} = \vec{f}, \qquad \nabla\cdot\vec{u} = q$$

Here: seamless transition from Darcy to Stokes (with $\mu = \tilde{\mu}$)

## Local flow problems discretized using Taylor–Hood elements

$$\mu \mathbf{K}^{-1}\vec{\psi}_{ij} + \nabla\varphi_{ij} - \tilde{\mu}\Delta\vec{\psi}_{ij} = 0, \qquad \nabla\cdot\vec{\psi}_{ij} = \begin{cases} w_i(\vec{x}), & \text{if } \vec{x}\in\Omega_i, \\ -w_j(\vec{x}), & \text{if } \vec{x}\in\Omega_j, \\ 0, & \text{otherwise,} \end{cases}$$

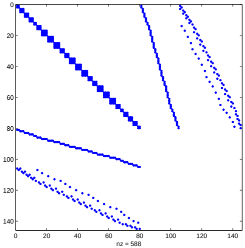


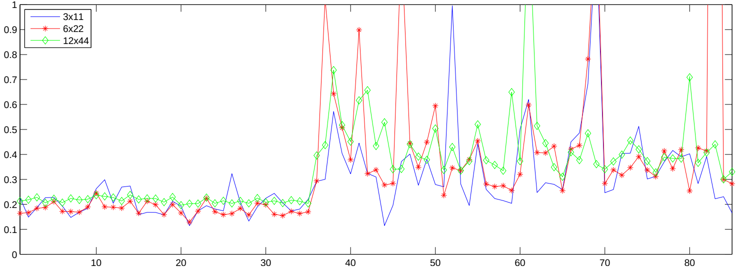
# MsMFE for the Stokes–Brinkman Equations
## Coarse-scale hybrid mixed system

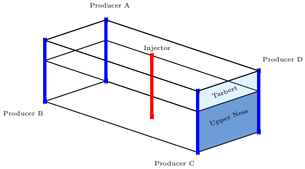
$$\begin{bmatrix} (\boldsymbol{A}^{-1})^{\mathsf{T}}\boldsymbol{\Psi}^{\mathsf{T}}\boldsymbol{B}_D^f\boldsymbol{\Psi}\boldsymbol{A}^{-1} & \boldsymbol{C} & \boldsymbol{D} \\ \boldsymbol{C}^{\mathsf{T}} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{D}^{\mathsf{T}} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^c \\ -\boldsymbol{p}^c \\ \boldsymbol{\lambda}^c \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{q}^c \\ \mathbf{0} \end{bmatrix}$$

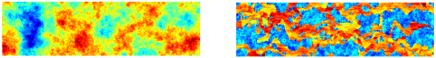
$\boldsymbol{A}$ – matrix with face areas
$\boldsymbol{\Psi}$ – matrix with basis functions
$\boldsymbol{B}_D^f$ – fine-scale *Darcy* TH-discretization

Fine-scale flux reconstructed as $\boldsymbol{u}^f = \boldsymbol{\Psi}\boldsymbol{u}^c$

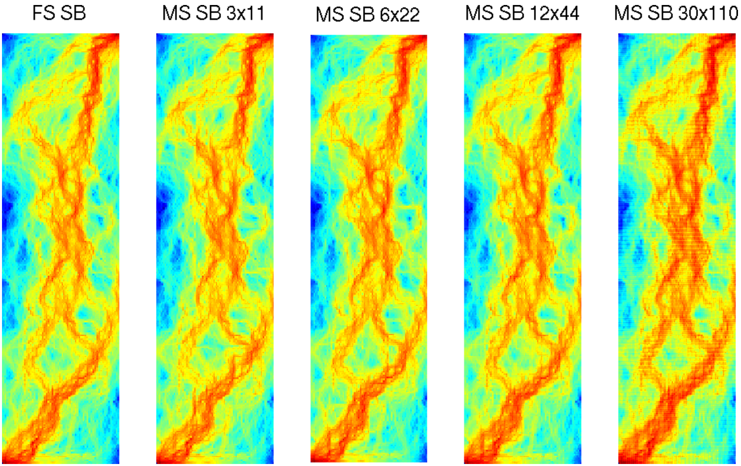




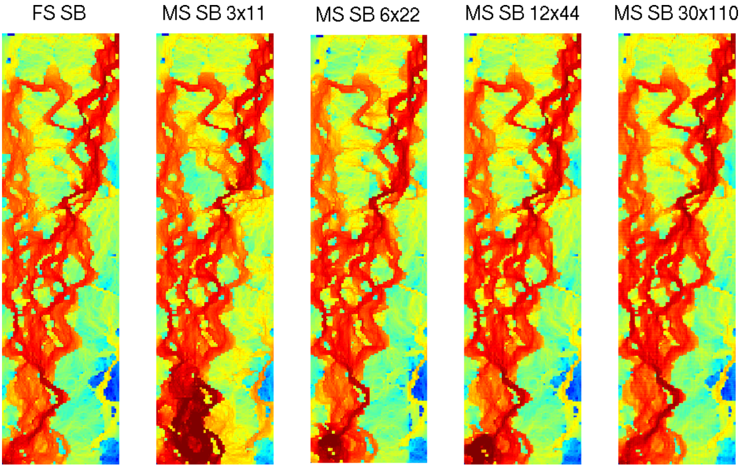


Tarbert (1–35)    Upper Ness (36–85)

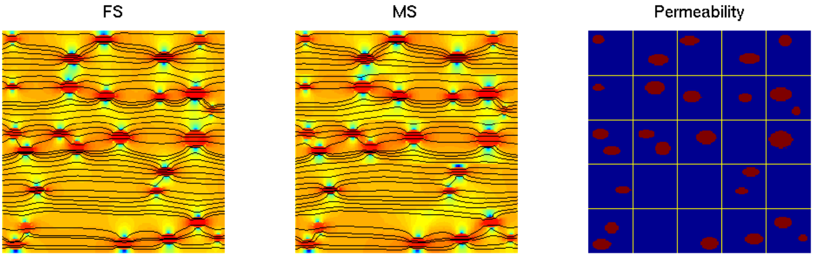
# MsMFE for the Stokes–Brinkman Equations
## Example 1: Layer 20 of SPE10



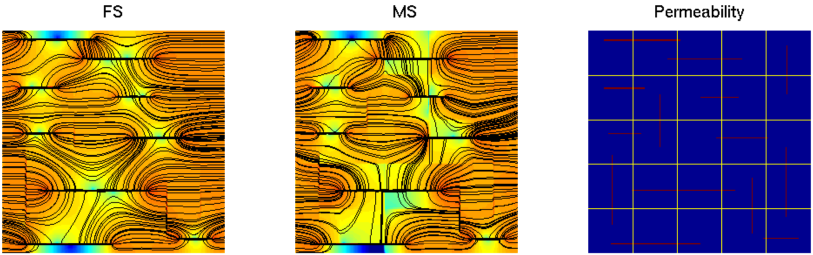
# MsMFE for the Stokes–Brinkman Equations

## Example 1: Layer 60 of SPE10 (worst case with injector in low-permeable block)





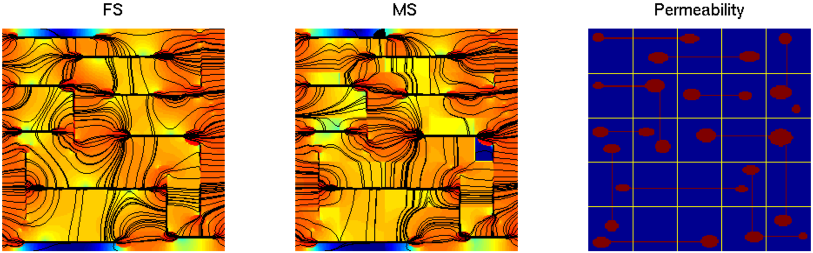

Fine-scale model consists of $200 \times 200$ cells
26 random vugs of sizes 1.8–10.4 $m^2$
Permeability in vugs is $10^7$ higher than in matrix




- Fine-scale model consists of 200 × 200 cells
- 14 random fractures of varying length
- Permeability in fractures is $10^7$ higher than in matrix

# MsMFE for the Stokes–Brinkman Equations

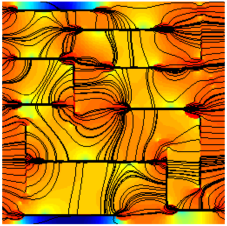
## Example 4: Vuggy and fractured reservoir (short and long correlation)



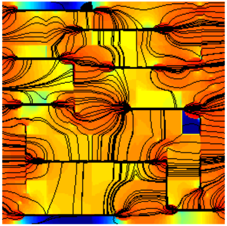
# MsMFE for the Stokes–Brinkman Equations

## Example 4: Vuggy and fractured reservoir (short and long correlation)

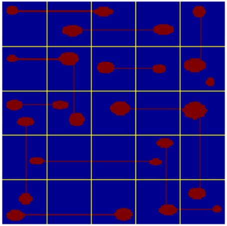
FS



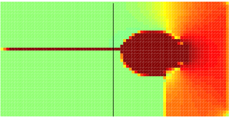
MS



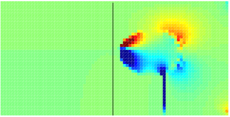
Permeability



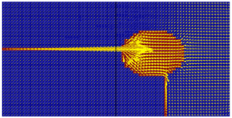
**Basis functions in x−direction**



**Basis functions in y−direction**



**Permeability and velocity vectors**



**Task:**

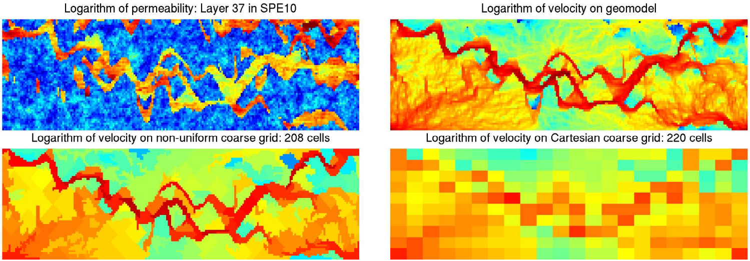
Given the ability to model velocity on geomodels and transport on coarse grids: Find a suitable coarse grid that best resolves fluid transport and minimizes loss of accuracy.

**Idea (Aarnes & Efendiev):**

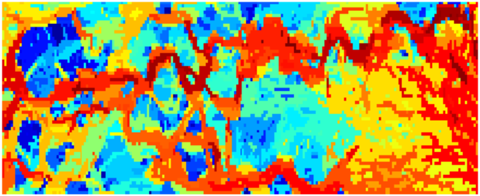
Use flow velocities to make a nonuniform grid in which each cell has approximately the same total flow

1. Segment the domain according to ln $|\vec{v}|$
2. Combine small blocks
3. Split blocks with too large flow
4. Combine small blocks

## SPE 10, Layer 37



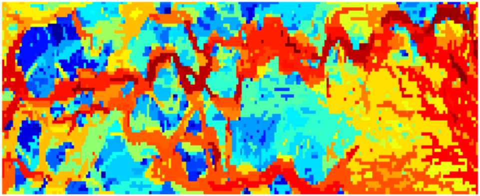
**Step 1:** Segment ln $|v|$ into $N$ level sets



Robust choice: $N = 10$

Step 1: 1411 cells

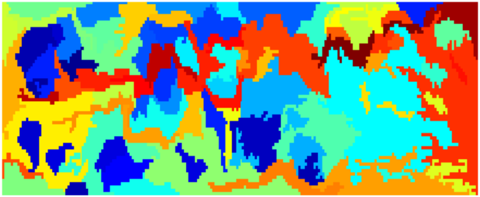
**Step 1:** Segment $\ln|v|$ into $N$ level sets



Robust choice: $N = 10$

Step 1: 1411 cells

**Step 2:** Combine small blocks ($|B| < c$) with a neighbour

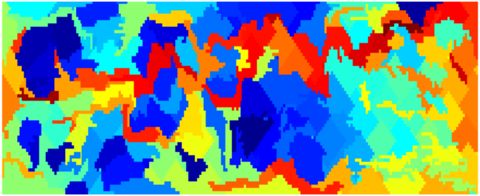


Merge $B$ and $B'$ if

$$\frac{1}{|B|}\int_B \ln|v| \approx \frac{1}{|B'|}\int_{B'} \ln|v|$$

Step 2: 94 cells

**Step 3:** Refine blocks with too much flow ($\int_B \ln|v|dx > C$)

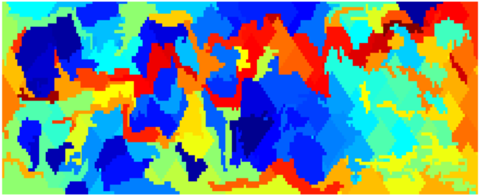


Build $B'$ inwards from $\partial B$
Restart with $B = B \setminus B'$

Step 3: 249 cells

# Flow-Based Nonuniform Coarsening
## Algorithm (for Layer 68 of SPE 10)

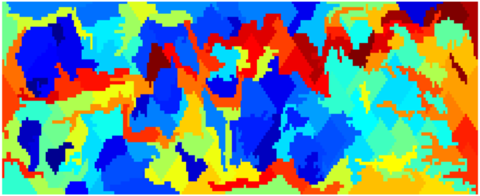
**Step 3:** Refine blocks with too much flow ($\int_B \ln|v|dx > C$)



Build $B'$ inwards from $\partial B$
Restart with $B = B \setminus B'$

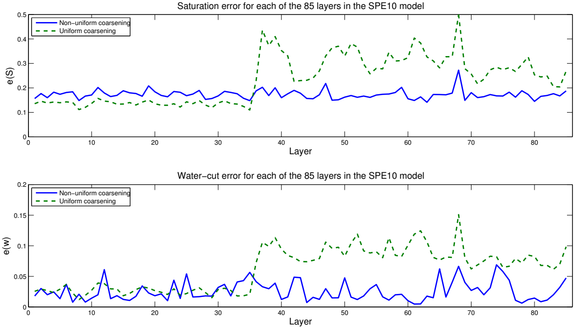
Step 3: 249 cells

**Step 4:** Combine small blocks with a neighbouring block



Step 2 repeated

Step 4: 160 cells

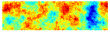
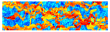



Geomodel:
$60 \times 220 = 13\,200$

Uniform:
$15 \times 44 = 660$

Non-uniform:
619–734 blocks

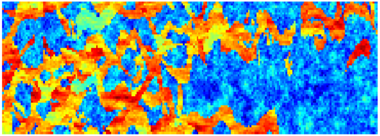
**Observations:**

- First 35 layers:  $\Rightarrow$ uniform grid adequate.
- Last 50 layers:  $\Rightarrow$ uniform grid inadequate.
- Non-uniform grid gives consistent results for all layers.

# Flow-Based Nonuniform Coarsening

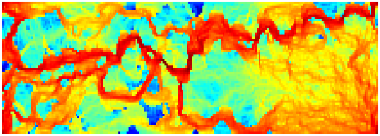
## Example 1: Layer 68, SPE10, 5-spot well pattern

Logarithm of permeability: Layer 68

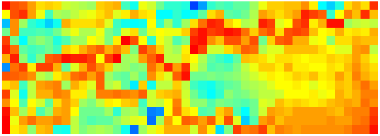


Geomodel: 13200 cells

Logarithm of velocity on geomodel

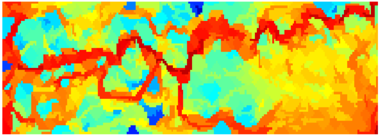


Logarithm of velocity on Cartesian coarse grid



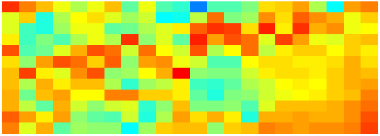
Coarse grid: 660 cells

Logarithm of velocity on non-uniform coarse grid



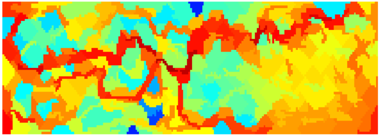
Coarse grid: 649 cells

Logarithm of velocity on Cartesian coarse grid



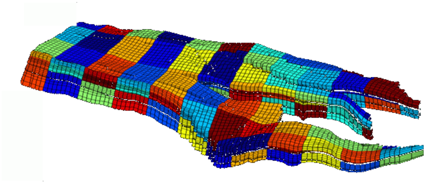
Coarse grid: 264 cells

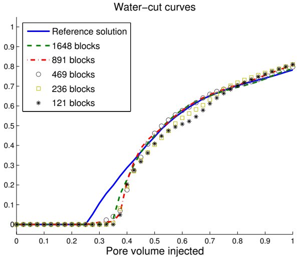
Logarithm of velocity on non-uniform coarse grid

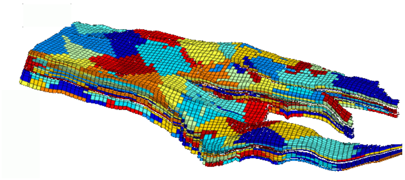


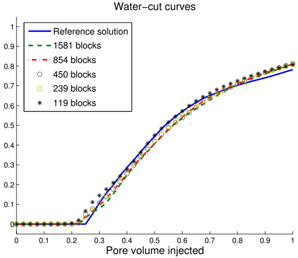
Coarse grid: 257 cells

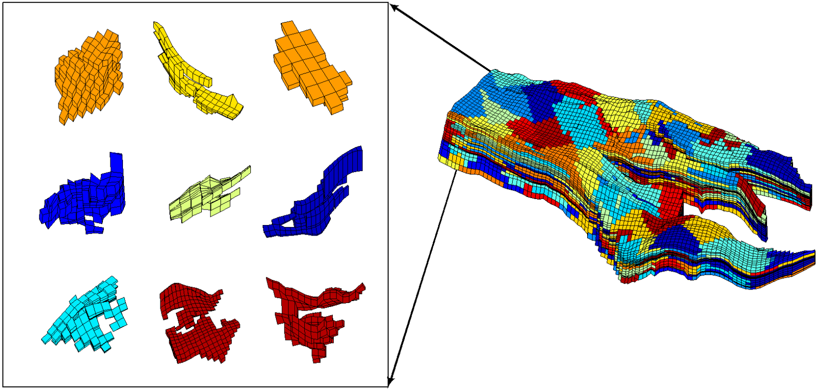
# Flow-Based Nonuniform Coarsening
## Example 2: real-field model





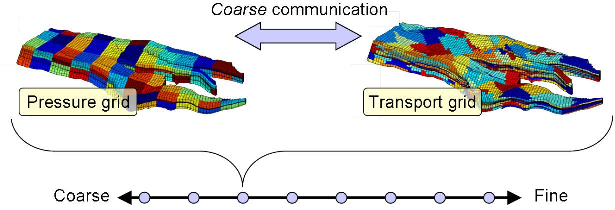




Both methods fast by themselves, but not optimal if they communicate via fine grid.

- Saturation piecewise constant on coarse *saturation* grid.
- Saturation-solver only requires fine-grid fluxes over coarse-grid interfaces.

→ Compute *coarse mappings* as a preprocessing step



Multiscale pressure system:

$$\begin{bmatrix} \boldsymbol{\Psi}^\mathsf{T} \boldsymbol{B}_f(\boldsymbol{\mathcal{I}} \boldsymbol{s}_{n-1}) \boldsymbol{\Psi} & \boldsymbol{C} & \boldsymbol{D} \\ \boldsymbol{C}^\mathsf{T} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{D}^\mathsf{T} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^n \\ -\boldsymbol{p}^n \\ \boldsymbol{\lambda}^n \end{bmatrix} = \begin{bmatrix} -\boldsymbol{D}_D \boldsymbol{\pi}_D^n \\ \mathbf{0} \\ \boldsymbol{v}_N^n \end{bmatrix}$$

Coarse-scale transport:

$$\boldsymbol{s}^n = \boldsymbol{s}^{n-1} + \Delta t \boldsymbol{\mathcal{I}}^\mathsf{T} \boldsymbol{\Lambda}_{\phi,f} \boldsymbol{\mathcal{I}} \Big( \boldsymbol{\mathcal{I}}^\mathsf{T} \boldsymbol{V}(\boldsymbol{v}_f^n) \boldsymbol{\mathcal{I}} \boldsymbol{f}(\boldsymbol{s}^n) + \boldsymbol{\mathcal{I}}^\mathsf{T} \boldsymbol{q}_+ \Big)$$

Reducing computational complexity

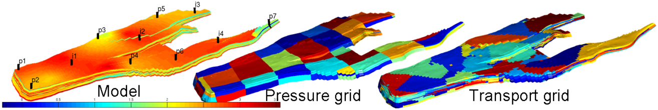
- rewrite time-dependent block of matrix

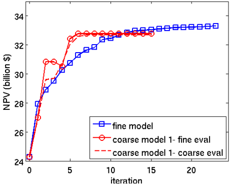
$$\boldsymbol{\Psi}^\mathsf{T} \boldsymbol{B}_f(\boldsymbol{\mathcal{I}} \boldsymbol{s}_{n-1}) \boldsymbol{\Psi} = \sum_{k=1}^{N_p} \boldsymbol{\Psi}^\mathsf{T} \boldsymbol{B}_f(\boldsymbol{\mathcal{I}} \boldsymbol{e}_k s_k^{n-1}) \boldsymbol{\Psi},$$

  where $\lambda(s_k^{n-1}) \boldsymbol{\Psi}^\mathsf{T} \boldsymbol{B}_f(\boldsymbol{\mathcal{I}} \boldsymbol{e}_k s_k^{n-1}) \boldsymbol{\Psi}$ is *time-independent*
- need only store $\boldsymbol{\mathcal{I}}^\mathsf{T} \boldsymbol{V}(\boldsymbol{v}_f^n) \boldsymbol{\mathcal{I}}$ on coarse-grid interfaces

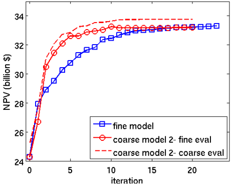
# MsMFEM and Nonuniform Coarsening

## Example: Water-flooding optimization (45 000 cells, real-field model)

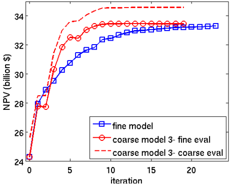



p: $4 \times 9 \times 2$, S: 136 blocks




p: $4 \times 9 \times 2$, S: 291 blocks




p: $4 \times 9 \times 2$, S: 800 blocks

Simulation time (20 time-steps) using simple MATLAB implementation on standard work-station:

- 80 sec if updating fine system for every step
- $<$ 5 sec if using precomputed coarse mappings

# The GeoScale Project Portfolio

## Research funded mainly by the Research Council of Norway