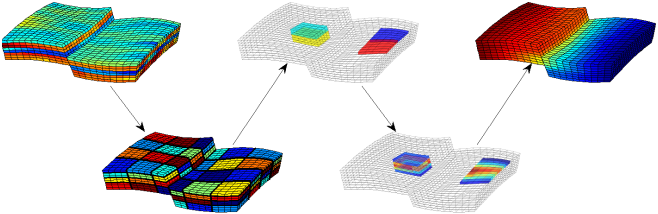
# Multiscale Methods for Flow in Porous Media: An Overview of Research at SINTEF

Knut–Andreas Lie, SINTEF, Norway



Houston, February 24, 2011

# What is multiscale simulation?
## Definition

**Generally:**

Methods that incorporate fine-scale information into a set of coarse scale equations in a way that is consistent with the local property of the differential operator

**Herein:**

  Multiscale pressure solver (fine and coarse grid)

$+$ Transport solver (on fine, intermediate, or coarse grid)

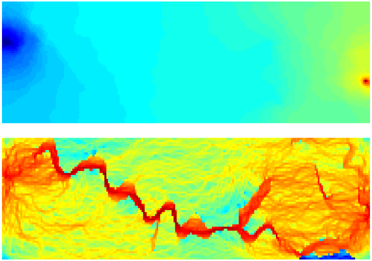
$=$ Multiscale simulation of models with higher detail

# What is multiscale simulation?
## Key ideas

### Use of sparsity (multiscale) structure

- effects resolved on different scales
- small changes from one step to next
- small changes from one simulation to next

### Multiscale idea



SPE10, Layer 36

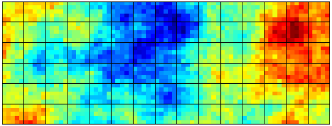
- Upscaling and downscaling in one step
- Pressure on coarse grid
- Velocity on fine grid

Incorporate impact of subgrid heterogeneity in approximation spaces

Advantages: utilize more geological data, more accurate solutions, geometrical flexibility

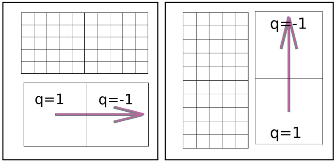
# What is multiscale simulation?
## Graphical illustration

Coarse partitioning:



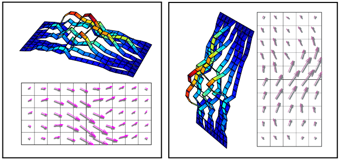
⇓

Local flow problems:



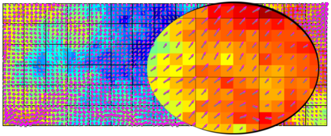

⇒

Flow solutions → basis functions:



⇑

Flow field with subresolution:



# Multiscale simulation

## Prerequisites for real-field applications

**More efficient than standard solvers:**

- easy to parallelise,
- less memory requirements than fine-grid solvers.

**Ability to handle industry-standard grids:**

- (highly) skewed and degenerate grid cells,
- non-matching cells,
- unstructured connectivities.

**Compatible with current solvers:**

- can be built on top of commercial/inhouse solvers,
- must be able to use existing linear solvers.

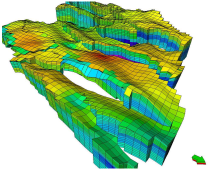
# Discretization of the fine-grid problem

Complex reservoir geometries

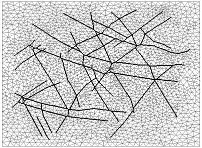
**Challenges:**

- 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

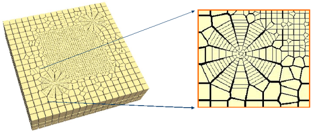
Corner point:



Tetrahedral:



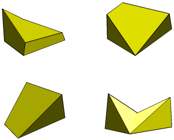
PEBI:



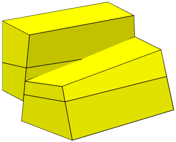
# Discretization of the fine-grid problem

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

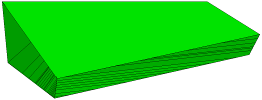
Skewed and deformed blocks:



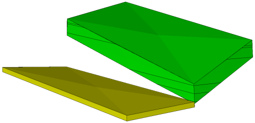
Non-matching cells:



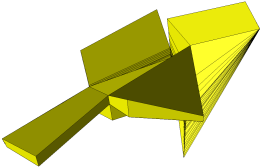
Many faces:



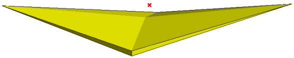
Small interfaces:



Difficult geometries:



(Very) high aspect ratios:



$800 \times 800 \times 0.25$ m

# Discretization of the fine-grid problem

## General family of conservative methods

### Basic formulation

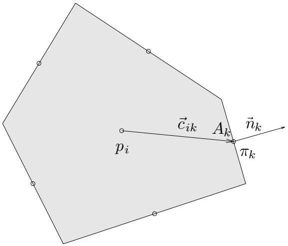
$$\boldsymbol{u}_i = \boldsymbol{T}_i(\boldsymbol{e}_i p_i - \boldsymbol{\pi}_i), \qquad \boldsymbol{e}_i = (1, \ldots, 1)^\mathsf{T}$$

$p_i$ – the pressure at the center of cell $i$

$\boldsymbol{u}_i$ – the vector of outward face fluxes

$\boldsymbol{\pi}_i$ – the vector of face pressures

$\boldsymbol{T}_i$ – the one-sided transmissibilities



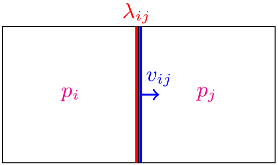

Special cases:

- The standard two-point method: $\boldsymbol{T}_{ii} = \vec{n}_i \cdot \boldsymbol{K}\vec{c}_i / |\vec{c}_i|^2$
- Multipoint flux-approximation methods (MPFA)
- Mixed finite-element methods
- Mimetic methods

# Discretization of the fine-grid problem

## Linear system: mixed hybrid form

$$\begin{bmatrix} B & C & D \\ C^T & 0 & 0 \\ D^T & 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ -p \\ \pi \end{bmatrix} = \begin{bmatrix} 0 \\ g \\ 0 \end{bmatrix},$$



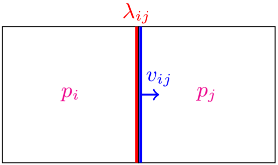

B defines an inner product. The matrix blocks read,

$$b_{ij} = \int_\Omega \psi_i \big(\lambda K\big)^{-1} \psi_j \, dx, \quad c_{ik} = \int_\Omega \phi_k \nabla \cdot \psi_i \, dx, \quad d_{ik} = \int_{\partial\Omega} |\psi_i \cdot n_k| \, dx$$

# Discretization of the fine-grid problem

## Linear system: mixed hybrid form

$$\begin{bmatrix} B & C & D \\ C^T & 0 & 0 \\ D^T & 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ -p \\ \pi \end{bmatrix} = \begin{bmatrix} 0 \\ g \\ 0 \end{bmatrix},$$




B defines an inner product. The matrix blocks read,

$$b_{ij} = \int_\Omega \psi_i \big(\lambda K\big)^{-1} \psi_j \, dx, \quad c_{ik} = \int_\Omega \phi_k \nabla \cdot \psi_i \, dx, \quad d_{ik} = \int_{\partial\Omega} |\psi_i \cdot n_k| \, dx$$

Positive-definite system obtained by a Schur-complement reduction

$$\big(D^T B^{-1} D - F^T L^{-1} F\big)\pi = F^T L^{-1} g,$$
$$F = C^T B^{-1} D, \quad L = C^T B^{-1} C.$$

Reconstruct cell pressures and fluxes by back-substition,

$$Lp = q + F^T \pi, \qquad Bv = Cp - D\pi.$$

# Discretization of the fine-grid problem

Herein: a mimetic method, Brezzi *et al.*, 2005

$$\boldsymbol{M}\boldsymbol{u} = (\boldsymbol{e}p - \boldsymbol{\pi}) \quad \longleftrightarrow \quad \boldsymbol{u} = \boldsymbol{T}(\boldsymbol{e}p - \boldsymbol{\pi})$$

Requiring exact solution of linear flow ($p = \boldsymbol{x}^\mathsf{T}\boldsymbol{a} + k$):

$$\boldsymbol{M}\boldsymbol{N}\boldsymbol{K} = \boldsymbol{C} \qquad \boldsymbol{N}\boldsymbol{K} = \boldsymbol{T}\boldsymbol{C}$$

$\boldsymbol{C}$ – vectors from cell to face centroids. $\boldsymbol{N}$: area-weighted normal vectors

Family of schemes (given by explicit formulas):

$$\boldsymbol{M} = \frac{1}{|\Omega_i|}\boldsymbol{C}\boldsymbol{K}^{-1}\boldsymbol{C}^\mathsf{T} + \boldsymbol{Q}_N^{\perp\,\mathsf{T}}\boldsymbol{S}_M\boldsymbol{Q}_N^\perp$$

$$\boldsymbol{T} = \frac{1}{|\Omega_i|}\boldsymbol{N}\boldsymbol{K}\boldsymbol{N}^\mathsf{T} + \boldsymbol{Q}_C^{\perp\,\mathsf{T}}\boldsymbol{S}\boldsymbol{Q}_C^\perp$$

$\boldsymbol{Q}_N^\perp$ is an orthonormal basis for the null space of $\boldsymbol{N}^\mathsf{T}$, and $\boldsymbol{S}_M$ is any positive definite matrix. Herein, we use null-space projection

$$\boldsymbol{P}_N^\perp = \boldsymbol{Q}_N^\perp\boldsymbol{S}_M\boldsymbol{Q}_N^\perp = \boldsymbol{I} - \boldsymbol{Q}_N\boldsymbol{Q}_N^\mathsf{T}$$

# Multiscale mixed finite elements

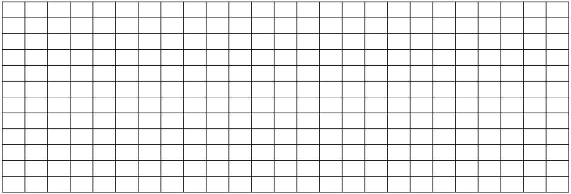
## Mixed formulation for incompressible flow

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

$$\begin{aligned} \int (\lambda K)^{-1} u \cdot v \, dx - \int p \nabla \cdot u \, dx &= 0, && \forall u \in H_0^{1,\mathsf{div}}, \\ \int \ell \nabla \cdot v \, dx &= \int q \ell \, dx, && \forall \ell \in L^2. \end{aligned}$$

### Standard MFE method

- Seek solution in $\mathbf{V}_h \times W_h \subset H_0^{1,\mathsf{div}} \times L^2$
- Approximation spaces: piecewise polynomials



# Multiscale mixed finite elements

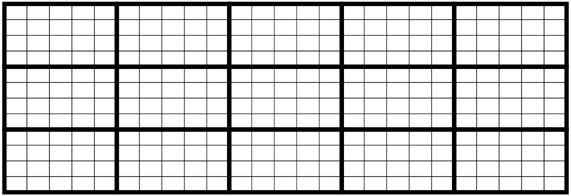
## Mixed formulation for incompressible flow

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

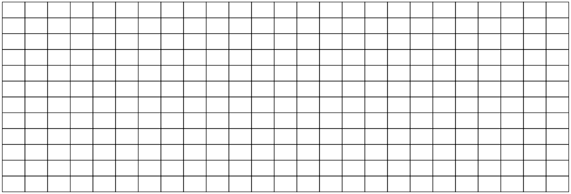
$$\begin{aligned} \int (\lambda K)^{-1} u \cdot v \, dx - \int p \nabla \cdot u \, dx &= 0, && \forall u \in H_0^{1,\text{div}}, \\ \int \ell \nabla \cdot v \, dx &= \int q \ell \, dx, && \forall \ell \in L^2. \end{aligned}$$

**Multiscale MFE method**

- Seek solution in $\mathbf{V}_{H,h} \times W_{H,h} \subset H_0^{1,\text{div}} \times L^2$
- Approximation spaces: local numerical solutions



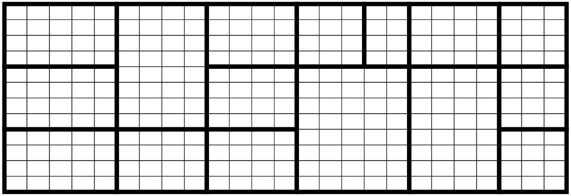
Fine grid with petrophysical parameters cell



# Multiscale mixed finite elements

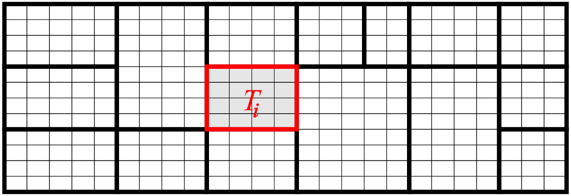
## Grids and basis functions in general

Fine grid with petrophysical parameters cell



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

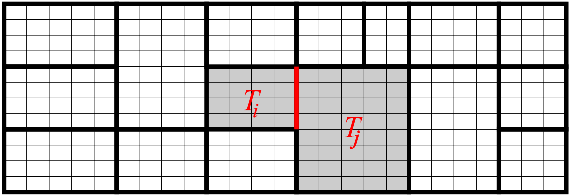
Fine grid with petrophysical parameters cell




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 $\phi_i \in V$.

Fine grid with petrophysical parameters cell




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 $\psi_{ij} \in U^{ms}$.

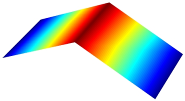
Decomposition:

- $p(x,y) = \sum_i p_i \phi_i(x,y)$ – sum over all coarse blocks
- $v(x,y) = \sum_{ij} v_{ij} \psi_{ij}(x,y)$ – sum over all block faces

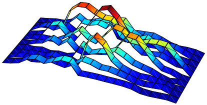
**Basis $\phi_i$ for pressure:**

$$\phi_i = \begin{cases} 1 & \text{in } T_i, \\ 0 & \text{otherwise.} \end{cases}$$

**Basis $\psi_{ij}$ for velocity:**



homogeneous (RT0)



heterogeneous

Velocity basis function $\psi_{ij}$ solves a local system of equations in $\Omega_{ij}$:

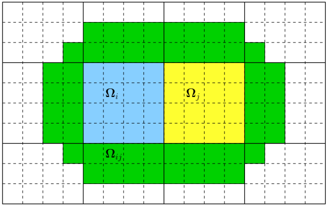
$$\vec{\psi}_{ij} = -\mu^{-1}\mathbf{K}\nabla\varphi_{ij}$$

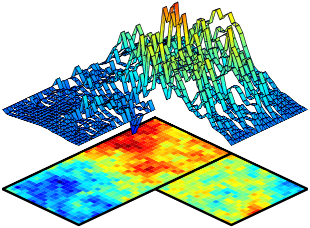
$$\nabla \cdot \vec{\psi}_{ij} = \begin{cases} w_i(\vec{x}), & \text{if } \vec{x} \in \Omega_i, \\ -w_j(\vec{x}), & \text{if } \vec{x} \in \Omega_j, \\ 0, & \text{otherwise.} \end{cases}$$

with no-flow conditions on $\partial\Omega_{ij}$

Source term: $w_i \propto \mathrm{trace}\,(K_i)$ drives a unit flow through $\Gamma_{ij}$.

If there is a sink/source in $T_i$, then $w_i \propto q_i$.



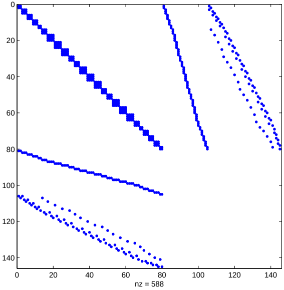
Split the basis functions, $\psi_{ij} = \psi_{ij}^H - \psi_{ji}^H$, to decouple across coarse faces.
*Hybrid* basis functions $\psi_{ij}^H$ as columns in a matrix $\mathbf{\Psi}$

**Coarse-scale hybrid mixed system**

$$\begin{bmatrix} \mathbf{\Psi}^\mathsf{T} \boldsymbol{B} \mathbf{\Psi} & \mathbf{\Psi}^\mathsf{T} \boldsymbol{C} \boldsymbol{\mathcal{I}} & \mathbf{\Psi}^\mathsf{T} \boldsymbol{D} \boldsymbol{\mathcal{J}} \\ \boldsymbol{\mathcal{I}}^\mathsf{T} \boldsymbol{C}^\mathsf{T} \mathbf{\Psi} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{\mathcal{J}}^\mathsf{T} \boldsymbol{D}^\mathsf{T} \mathbf{\Psi} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}^c \\ -\boldsymbol{p}^c \\ \boldsymbol{\lambda}^c \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{g}^c \\ \mathbf{0} \end{bmatrix}$$

$\mathbf{\Psi}$ – matrix with basis functions
$\boldsymbol{\mathcal{I}}$ – prolongation from blocks to cells
$\boldsymbol{\mathcal{J}}$ – prolongation from block faces to cell faces

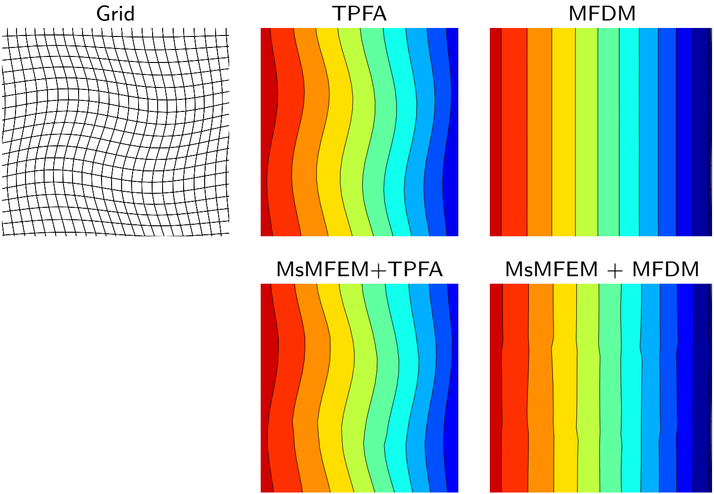
Reconstruction of fine-scale velocity $\boldsymbol{v}^f = \mathbf{\Psi} \boldsymbol{v}^c$
(Pressure bases may also have fine-scale structure if necessary)



# Multiscale mixed finite elements

## Multiscale method inherits properties of fine-scale solver

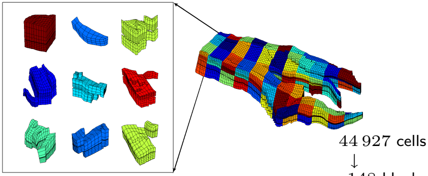
Single-phase flow, homogeneous **K**, linear pressure drop



# Generation of coarse grids

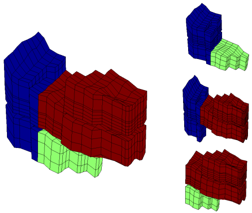
## Workflow with automated upgridding in 3D (here for logically Cartesian grids)

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



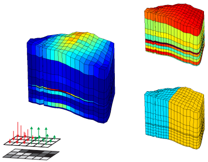

9 different coarse blocks

2) Detect all adjacent blocks



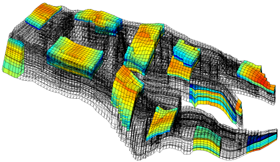
3) Compute basis functions

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



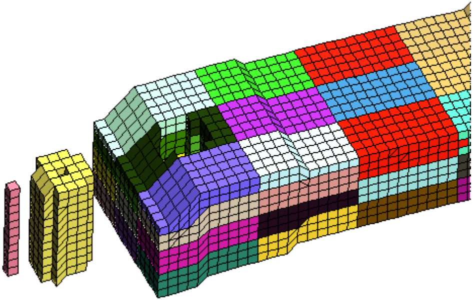
for all pairs of blocks

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



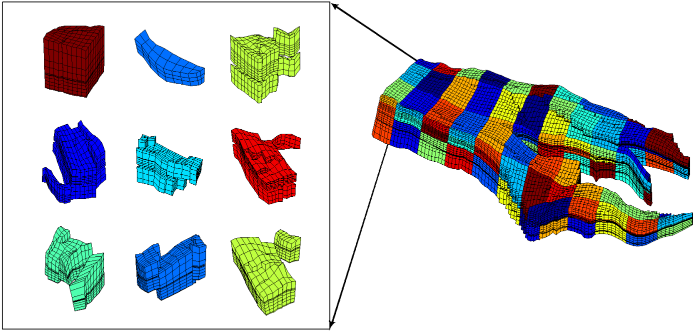
**(Unique) grid flexibility:**

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



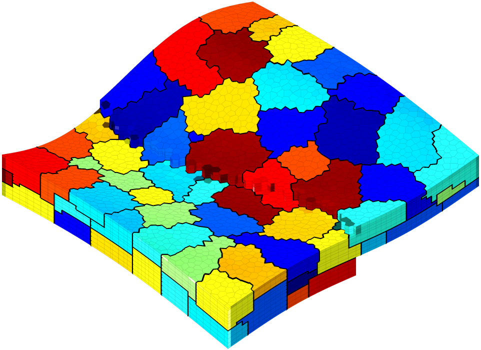
**(Unique) grid flexibility:**

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



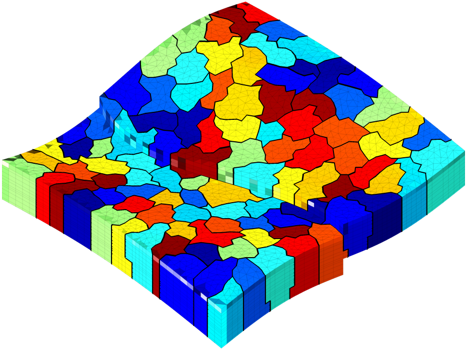
**(Unique) grid flexibility:**

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



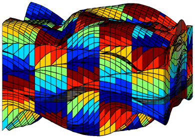
# Generation of coarse grids

Simple idea: follow geological structures for improved accuracy!

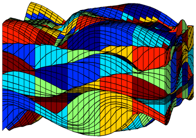
### A depositional bed

Eroded layers gives a large number of degenerate and inactive cells.
Relative error in saturation at $0.5$PVI:

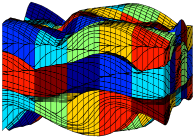
| Coarse grid | Isotropic | Anisotropic | Heterogeneous |
|---:|:---:|:---:|:---:|
| Physical | $0.1339$ | $0.2743$ | $0.2000$ |
| Logical | $0.0604$ | $0.1381$ | $0.1415$ |
| Constrained | $0.0573$ | $0.1479$ | $0.0993$ |



physical



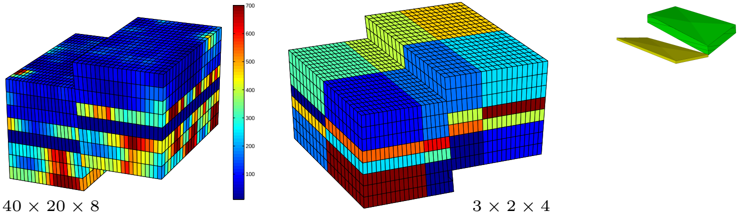
logical

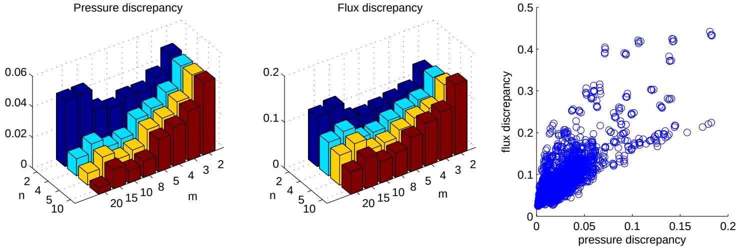


constrained

# Generation of coarse grids

## A fully robust method will require post-processing



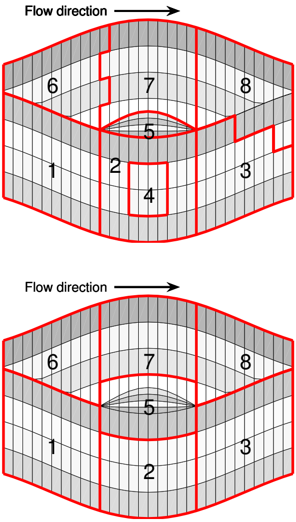




100 realizations: fault throw normally distributed with standard deviation equal 1/5 of the total reservoir height. Each realization has a new permeability field

# Generation of coarse grids

## Simple guidelines for choosing good coarse grids

1. Minimize bidirectional flow over interfaces:
    - Avoid unnecessary irregularity ($\Gamma_{6,7}$ and $\Gamma_{3,8}$)
    - Avoid single neighbors ($\Omega_4$)
    - Ensure that there are faces transverse to flow direction ($\Omega_5$)
2. Blocks and faces should follow geological layers ($\Omega_3$ and $\Omega_8$)
3. Blocks should adapt to flow obstacles whenever possible
4. For efficiency: minimize the number of connections
5. Avoid having too many small blocks



Semi-discretized and linearized pressure equation:

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

Hybrid system:

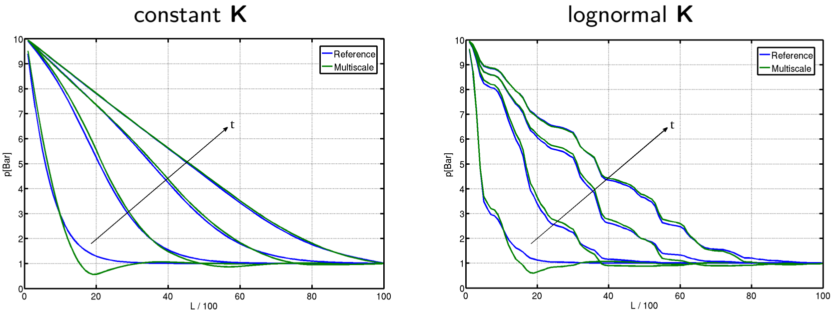
$$\begin{bmatrix} \boldsymbol{B} & \boldsymbol{C} & \boldsymbol{D} \\ \boldsymbol{C}^{\mathsf{T}}-\boldsymbol{V}_{\nu-1}^{\mathsf{T}} & \boldsymbol{P}_{\nu-1} & \boldsymbol{0} \\ \boldsymbol{D}^{\mathsf{T}} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}\begin{bmatrix} \boldsymbol{u}_{\nu} \\ -\boldsymbol{p}_{\nu}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

Coarse-grid formulation:

$$\begin{bmatrix} \boldsymbol{\Psi}^{\mathsf{T}}\boldsymbol{B}\boldsymbol{\Psi} & \boldsymbol{\Psi}^{\mathsf{T}}\boldsymbol{C}\boldsymbol{\mathcal{I}} & \boldsymbol{\Psi}^{\mathsf{T}}\boldsymbol{D}\boldsymbol{\mathcal{J}} \\ \boldsymbol{\Psi}^{\mathsf{T}}(\boldsymbol{C}-\boldsymbol{V})\boldsymbol{\mathcal{I}}-\boldsymbol{D}_{\lambda}\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{P}\boldsymbol{\mathcal{I}} & \boldsymbol{\mathcal{I}}^{\mathsf{T}}\boldsymbol{P}\boldsymbol{\mathcal{I}} & \boldsymbol{0} \\ \boldsymbol{\mathcal{J}}^{\mathsf{T}}\boldsymbol{D}^{\mathsf{T}}\boldsymbol{\Psi} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}\begin{bmatrix} \boldsymbol{u} \\ -\boldsymbol{p} \\ \boldsymbol{\pi} \end{bmatrix}=\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{\mathcal{I}}^{\mathsf{T}}\boldsymbol{P}\boldsymbol{p}_{f}^{n} \\ \boldsymbol{0} \end{bmatrix}$$

# Compressible black-oil models

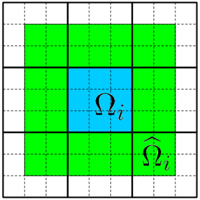
## Example 1: tracer transport in ideal gas (Lunati&Jenny 2006)




$p(0,t) = 1$ bar, $p(x,0) = 10$ bar, coarse grid: 5 blocks, fine grid: 100 cells

Residual equation:

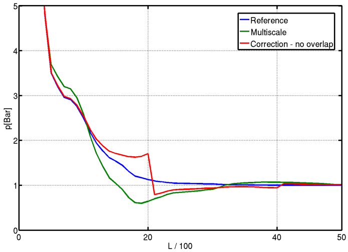
$$\begin{bmatrix} \boldsymbol{B} & \boldsymbol{C} \\ \boldsymbol{C}^\mathsf{T} & \boldsymbol{P} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_{\mathrm{ms}} + \hat{\boldsymbol{u}}^{\nu+1} \\ \boldsymbol{p}_{\mathrm{ms}} + \hat{\boldsymbol{p}}^{\nu+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{P}\boldsymbol{p}^n + \boldsymbol{V}\boldsymbol{u}^\nu \end{bmatrix}$$



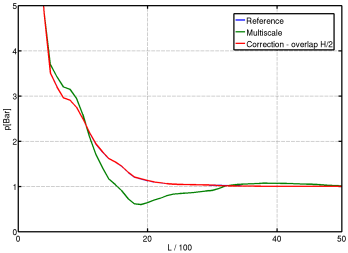

Localization: $\hat{u} = \sum_i \hat{u}_i$, $\hat{p} = \sum_i \hat{p}_i$

- zero right-hand-side in $\widehat{\Omega}_i \setminus \Omega_i$
- zero flux BCs on $\partial\widehat{\Omega}_i$

**Without overlap:**



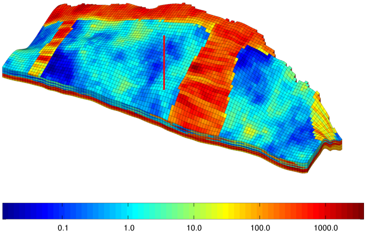

**With overlap:**



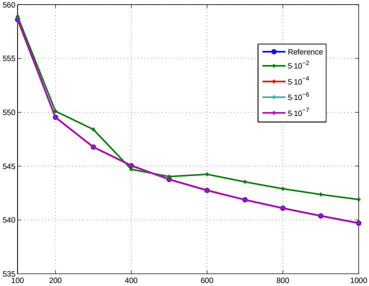
# Compressible black-oil models

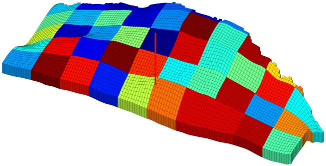
## Example 2: primary production

- Shallow-marine reservoir (realization from SAIGUP)
- Model size: $40 \times 120 \times 20$
- Initially filled with gas, 200 bar
- Single producer, bhp=150 bar
- Multiscale solution for different tolerences compared with fine-scale reference solution.




Rate in well perforation (m$^3$/day)



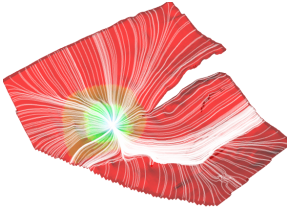



# Transport solvers

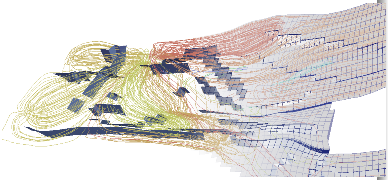
## Multiscale methods need efficient transport solvers

- Streamlines, time-of-flight, etc:
  - intuitive visualization + new data
  - subscale resolution
  - good scaling, known to be efficient

**Flow pattern (CO2 injection):**



**Connections across faults:**

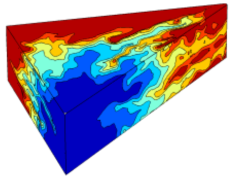


# Transport solvers

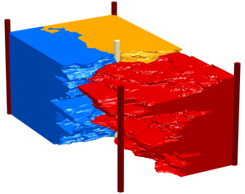
## Multiscale methods need efficient transport solvers

- Streamlines, time-of-flight, etc:
  - intuitive visualization + new data
  - subscale resolution
  - good scaling, known to be efficient

**Time-of-flight (timelines):**



**Flooded volumes (stationary tracer):**

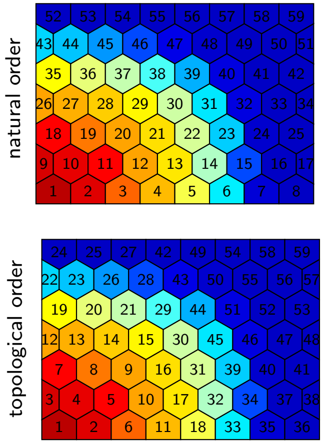


# Transport solvers

## Multiscale methods need efficient transport solvers

- Streamlines, time-of-flight, etc:
  - intuitive visualization + new data
  - subscale resolution
  - good scaling, known to be efficient
- Optimal ordering
  - same assumptions as for streamlines
  - utilize causality $\longrightarrow \mathcal{O}(n)$ algorithm, cell-by-cell solution
  - local control over nonlinear iterations

**Topological sorting**

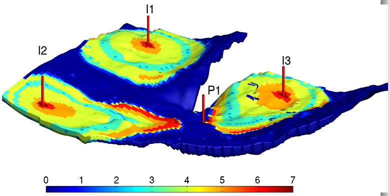


# Transport solvers

## Multiscale methods need efficient transport solvers

- Streamlines, time-of-flight, etc:
  - intuitive visualization + new data
  - subscale resolution
  - good scaling, known to be efficient
- Optimal ordering
  - same assumptions as for streamlines
  - utilize causality $\longrightarrow \mathcal{O}(n)$ algorithm, cell-by-cell solution
  - local control over nonlinear iterations

**Local iterations:**




Johansen formation: 27 437 active cells

Global vs local Newton–Raphson solver

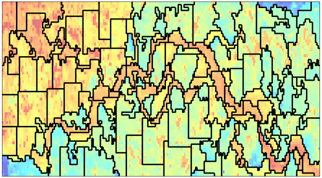
| $\Delta t$ | global | | local | |
|---|---|---|---|---|
| days | time | iter | time (sec) | iter |
| 125 | 2.26 | 12.69 | 0.044 | 0.93 |
| 250 | 2.35 | 12.62 | 0.047 | 1.10 |
| 500 | 2.38 | 13.25 | 0.042 | 1.41 |
| 1000 | 2.50 | 13.50 | 0.042 | 1.99 |

# Transport solvers

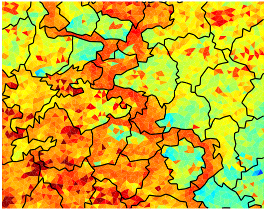
## Multiscale methods need efficient transport solvers

- Streamlines, time-of-flight, etc:
  - intuitive visualization + new data
  - subscale resolution
  - good scaling, known to be efficient
- Optimal ordering
  - same assumptions as for streamlines
  - utilize causality $\longrightarrow \mathcal{O}(n)$ algorithm, cell-by-cell solution
  - local control over nonlinear iterations
- Amalgamation of cells
  - flow-adapted grids
  - simple and flexible coarsening
  - adaptive gridding schemes
  - efficient model reduction

**Cartesian grid:**



**Triangular grids:**

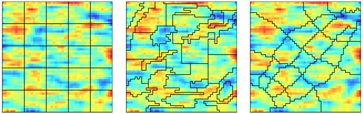


# Transport solvers

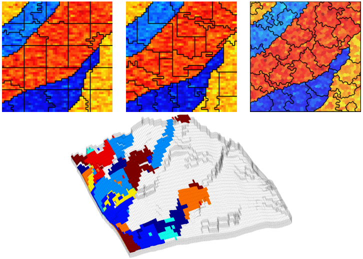
## Multiscale methods need efficient transport solvers

- Streamlines, time-of-flight, etc:
  - intuitive visualization + new data
  - subscale resolution
  - good scaling, known to be efficient
- Optimal ordering
  - same assumptions as for streamlines
  - utilize causality $\longrightarrow \mathcal{O}(n)$ algorithm, cell-by-cell solution
  - local control over nonlinear iterations
- Amalgamation of cells
  - flow-adapted grids
  - simple and flexible coarsening
  - adaptive gridding schemes
  - efficient model reduction

**Different partitioning:**



**Adapting to geology**

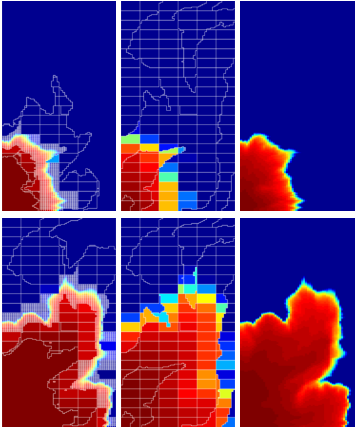


# Transport solvers

## Multiscale methods need efficient transport solvers

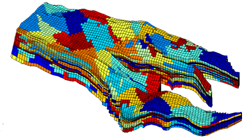
- Streamlines, time-of-flight, etc:
  - intuitive visualization + new data
  - subscale resolution
  - good scaling, known to be efficient
- Optimal ordering
  - same assumptions as for streamlines
  - utilize causality $\longrightarrow \mathcal{O}(n)$ algorithm, cell-by-cell solution
  - local control over nonlinear iterations
- Amalgamation of cells
  - flow-adapted grids
  - simple and flexible coarsening
  - adaptive gridding schemes
  - efficient model reduction

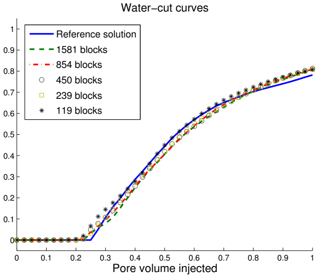
**Dynamic adaption**



- Streamlines, time-of-flight, etc:
  - intuitive visualization + new data
  - subscale resolution
  - good scaling, known to be efficient
- Optimal ordering
  - same assumptions as for streamlines
  - utilize causality $\longrightarrow \mathcal{O}(n)$ algorithm, cell-by-cell solution
  - local control over nonlinear iterations
- Amalgamation of cells
  - flow-adapted grids
  - simple and flexible coarsening
  - adaptive gridding schemes
  - efficient model reduction

**Model reduction by coarsening:**



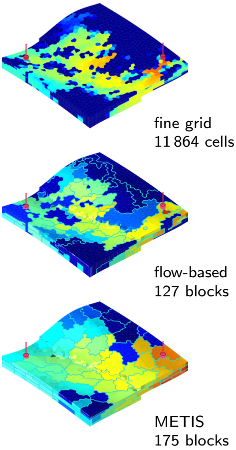


# Transport solvers

## Multiscale methods need efficient transport solvers

- Streamlines, time-of-flight, etc:
  - intuitive visualization + new data
  - subscale resolution
  - good scaling, known to be efficient
- Optimal ordering
  - same assumptions as for streamlines
  - utilize causality $\longrightarrow \mathcal{O}(n)$ algorithm, cell-by-cell solution
  - local control over nonlinear iterations
- Amalgamation of cells
  - flow-adapted grids
  - simple and flexible coarsening
  - adaptive gridding schemes
  - efficient model reduction

**Model reduction by coarsening:**



# Usage and outlook

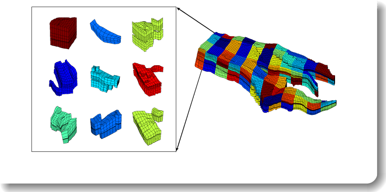
For what purposes are multiscale methods useful?

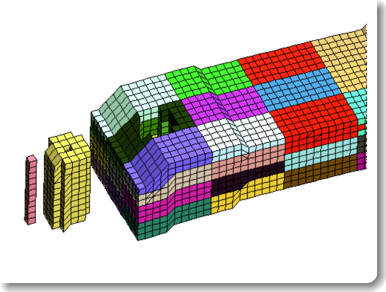
- As robust upscaling methods?
- As alternative to upscaling and fine-scale solution?
- To provide flow simulation earlier in the modelling loop?
- To get 90% of the answer in 10% of the time?
- *Fit-for-purpose solvers in workflows for ranking, history matching, planning, optimization, ...*

# Usage and outlook

## Success stories and unreaped potential

- More flexible wrt grids than standard upscaling methods: automatic coarsening



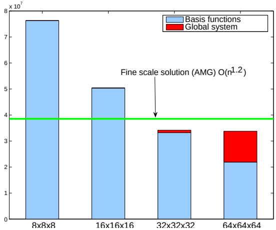


# Usage and outlook

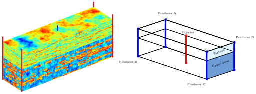
## Success stories and unreaped potential

- More flexible wrt grids than standard upscaling methods: automatic coarsening
- Reuse of computations, key to computational efficiency

**Operations vs. upscaling factor:**




**SPE10: 1.1 mill cells**



Inhouse code from 2005:

multiscale: 2 min and 20 sec
multigrid: 8 min and 36 sec

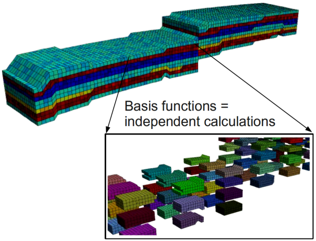
Fully unstructured Matlab/C code from 2010:

mimetic : 5–6 min

# Usage and outlook

## Success stories and unreaped potential

- More flexible wrt grids than standard upscaling methods: automatic coarsening
- Reuse of computations, key to computational efficiency
- Natural (elliptic) parallelism:
  - multicore and heterogeneous computing

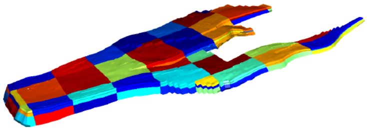



# Usage and outlook

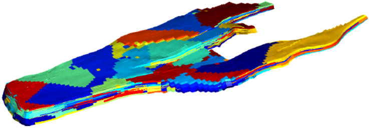
## Success stories and unreaped potential

- More flexible wrt grids than standard upscaling methods: automatic coarsening
- Reuse of computations, key to computational efficiency
- Natural (elliptic) parallelism:
  - multicore and heterogeneous computing
- Fine-scale velocity $\longrightarrow$ different grid for flow and transport

**Pressure grid:**



**Transport grid:**

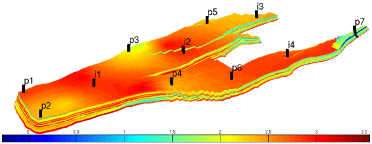


# Usage and outlook

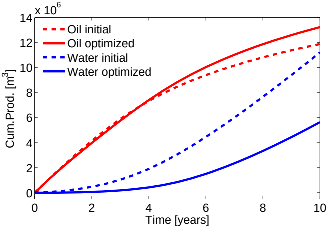
## Success stories and unreaped potential

- More flexible wrt grids than standard upscaling methods: automatic coarsening
- Reuse of computations, key to computational efficiency
- Natural (elliptic) parallelism:
  - multicore and heterogeneous computing
- Fine-scale velocity $\longrightarrow$ different grid for flow and transport
- Method for model reduction:
  - adjoint simulations $\longrightarrow$ gradients
  - ensemble simulations with representative basis functions

**Water-flood optimization:**




Reservoir geometry from a Norwegian Sea field



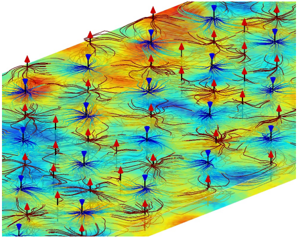

Forward simulations:
44 927 cells, 20 time steps, $<$ 5 sec in Matlab

# Usage and outlook

## Success stories and unreaped potential

- More flexible wrt grids than standard upscaling methods: automatic coarsening
- Reuse of computations, key to computational efficiency
- Natural (elliptic) parallelism:
  - multicore and heterogeneous computing
- Fine-scale velocity $\longrightarrow$ different grid for flow and transport
- Method for model reduction:
  - adjoint simulations $\longrightarrow$ gradients
  - ensemble simulations with representative basis functions

**History matching 1 million cells:**



7 years: 32 injectors, 69 producers

Generalized travel-time inversion + multiscale:
7 forward simulations, 6 inversions

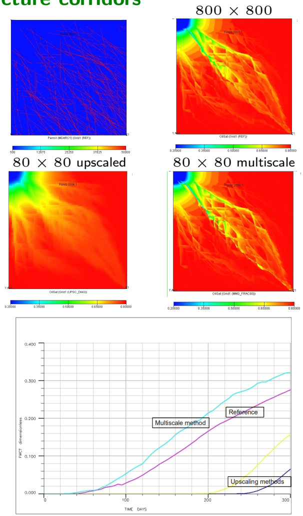
| | CPU-time (wall clock) | | |
|---|---|---|---|
| Solver | Total | Pres. | Transp. |
| Multigrid | 39 min | 30 min | 5 min |
| Multiscale | 17 min | 7 min | 6 min |

# Usage and outlook

## Success stories and unreaped potential

- More flexible wrt grids than standard upscaling methods: automatic coarsening
- Reuse of computations, key to computational efficiency
- Natural (elliptic) parallelism:
  - multicore and heterogeneous computing
- Fine-scale velocity $\longrightarrow$ different grid for flow and transport
- Method for model reduction:
  - adjoint simulations $\longrightarrow$ gradients
  - ensemble simulations with representative basis functions
- Improved model fidelity:
  - subscale resolution
  - multiphysics applications

**Fracture corridors**

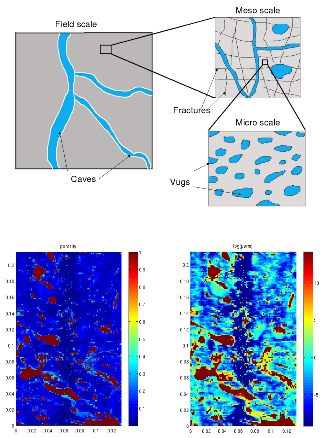


# Usage and outlook

## Success stories and unreaped potential

- More flexible wrt grids than standard upscaling methods: automatic coarsening
- Reuse of computations, key to computational efficiency
- Natural (elliptic) parallelism:
  - multicore and heterogeneous computing
- Fine-scale velocity $\longrightarrow$ different grid for flow and transport
- Method for model reduction:
  - adjoint simulations $\longrightarrow$ gradients
  - ensemble simulations with representative basis functions
- Improved model fidelity:
  - subscale resolution
  - multiphysics applications

**Stokes–Brinkmann:**



## Usage and outlook

### Resolved and unresolved questions

**Capabilities:**

- ✓ Two-phase flow
- ✓ Cartesian / unstructured grids
- ✓ Realistic flow physics ⇒ iterations
  - Correction functions + smoothing
  - Residual formulation + domain decomposition
- ✓ Pointwise accuracy ⇒ iterations

# Usage and outlook

## Resolved and unresolved questions

### Capabilities:

✓ Two-phase flow

✓ Cartesian / unstructured grids

✓ Realistic flow physics $\Rightarrow$ iterations

- Correction functions + smoothing
- Residual formulation + domain decomposition

✓ Pointwise accuracy $\Rightarrow$ iterations

### Not yet there:

- Compressible three-phase black-oil + non-Cartesian grids
- Parallelization
- Fully implicit formulation
- Compositional, thermal, ...

Other issues:

- How should unstructured grids be coarsened?
- Need for global information or iterative procedures?
- A posteriori error analysis (resolution or fine-scale junk)?
- More than two levels in hierarchical grid?
- How to include models from finer scales?

# Current and future research at SINTEF

## Three main directions