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

Knut–Andreas Lie, SINTEF, Norway

Schlumberger/Chevron, 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**
- 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:

Flow field with subresolution:

⇓

Local flow problems:

Flow solutions → basis functions:
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.
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:

Many faces:

Difficult geometries:

Non-matching cells:

Small interfaces:

(Very) high aspect ratios:

800 × 800 × 0.25 m
Discretization of the fine-grid problem
General family of conservative methods

Basic formulation

\[ u_i = T_i (e_i p_i - \pi_i), \quad e_i = (1, \ldots, 1)^T \]

- \( p_i \) – the pressure at the center of cell \( i \)
- \( u_i \) – the vector of outward face fluxes
- \( \pi_i \) – the vector of face pressures
- \( T_i \) – the one-sided transmissibilities

Special cases:

- The standard two-point method: \( T_{ii} = \vec{n}_i \cdot \vec{K} \vec{c}_i / |\vec{c}_i|^2 \)
- 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,

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

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

Positive-definite system obtained by a Schur-complement reduction

\[
(D^T B^{-1} D - F^T L^{-1} F) \pi = F^T L^{-1} g,
\]

\[F = C^T B^{-1} D, \quad L = C^T B^{-1} C.\]

Reconstruct cell pressures and fluxes by back-substitution,

\[
Lp = q + F^T \pi, \quad Bv = C p - D \pi.
\]
Discretization of the fine-grid problem
Herein: a mimetic method, Brezzi et al., 2005

\[ M \mathbf{u} = (e p - \pi) \quad \leftrightarrow \quad \mathbf{u} = T(e p - \pi) \]

Requiring exact solution of linear flow \((p = x^T a + k)\):

\[ M N K = C \quad N K = T C \]

\(C\) – vectors from cell to face centroids. \(N\): area-weighted normal vectors

Family of schemes (given by explicit formulas):

\[
M = \frac{1}{|\Omega_i|} C K^{-1} C^T + Q_N^T S_M Q_N^T
\]

\[
T = \frac{1}{|\Omega_i|} N K N^T + Q_C^T S Q_C^T
\]

\(Q_N^\perp\) is an orthonormal basis for the null space of \(N^T\), and \(S_M\) is any positive definite matrix. Herein, we use null-space projection

\[
P_N^\perp = Q_N^\perp S_M Q_N^\perp = I - Q_N Q_N^T
\]
Find \((v, p) \in H_0^{1, \text{div}} \times L^2\) such that

\[
\int (\lambda K)^{-1} u \cdot v \, dx - \int p \nabla \cdot u \, dx = 0, \quad \forall u \in H_0^{1, \text{div}},
\]

\[
\int \ell \nabla \cdot v \, dx = \int q \ell \, dx, \quad \forall \ell \in L^2.
\]

**Standard MFE method**

- Seek solution in \(V_h \times W_h \subset H_0^{1, \text{div}} \times L^2\)
- 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

\[
\int (\lambda K)^{-1} u \cdot v \, dx - \int p \nabla \cdot u \, dx = 0, \quad \forall u \in H_0^{1, \text{div}},
\]

\[
\int \ell \nabla \cdot v \, dx = \int q \ell \, dx, \quad \forall \ell \in L^2.
\]

Multiscale MFE method

- Seek solution in \(V_{H,h} \times W_{H,h} \subset H_0^{1, \text{div}} \times L^2\)

- Approximation spaces: local numerical solutions
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 $\psi_{ij} \in U_{ms}$.
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 $\psi_{ij} \in U^{ms}$.
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 $\phi_i \in V$. 
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 $\Omega_{ij}$:

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

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

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

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

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

---

Multiscale mixed finite elements
Local flow problems

Homogeneous medium
Heterogeneous medium

Homogeneous medium
Heterogeneous medium
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 $\Psi$

**Coarse-scale hybrid mixed system**

\[
\begin{bmatrix}
\Psi^T B \Psi & \Psi^T C \mathcal{I} & \Psi^T D \mathcal{J} \\
\mathcal{I}^T C^T \Psi & 0 & 0 \\
\mathcal{J}^T D^T \Psi & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
\nu^c \\
-p^c \\
\lambda^c \\
\end{bmatrix} =
\begin{bmatrix}
0 \\
0 \\
0 \\
\end{bmatrix}
\]

$\Psi$ – matrix with basis functions

$\mathcal{I}$ – prolongation from blocks to cells

$\mathcal{J}$ – prolongation from block faces to cell faces

Reconstruction of fine-scale velocity $\nu^f = \Psi \nu^c$

(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

Grid

TPFA

MFDM

MsMFEM+TPFA

MsMFEM + MFDM
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

- 44,927 cells → 148 blocks
- 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.
A depositional bed

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

<table>
<thead>
<tr>
<th>Coarse grid</th>
<th>Isotropic</th>
<th>Anisotropic</th>
<th>Heterogeneous</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical</td>
<td>0.1339</td>
<td>0.2743</td>
<td>0.2000</td>
</tr>
<tr>
<td>Logical</td>
<td>0.0604</td>
<td>0.1381</td>
<td>0.1415</td>
</tr>
<tr>
<td>Constrained</td>
<td>0.0573</td>
<td>0.1479</td>
<td>0.0993</td>
</tr>
</tbody>
</table>
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
Compressible black-oil models
Fine-grid and coarse-grid formulation

Semi-discretized and linearized pressure equation:

\[ c_{\nu-1} \frac{p^n_\nu - p^{n-1}_\nu}{\Delta t} + \nabla \cdot \vec{u}^n_\nu - \zeta^n_{\nu-1} \vec{u}^n_{\nu-1} \cdot K^{-1} \vec{u}^n_\nu = q \]

Hybrid system:

\[
\begin{bmatrix}
B & C & D \\
C^T - V^T_{\nu-1} & P_{\nu-1} & 0 \\
D^T & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\vec{u}_\nu \\
-\vec{p}_\nu \\
\pi_\nu
\end{bmatrix}
= \begin{bmatrix}
0 \\
P_{\nu-1} p^{n-1} + q
\end{bmatrix}
\]

Coarse-grid formulation:

\[
\begin{bmatrix}
\Psi^T B \Psi & \Psi^T C \mathcal{I} & \Psi^T D \mathcal{J} \\
\Psi^T (C - V) \mathcal{I} - D_\lambda \Phi^T P \mathcal{I} & \mathcal{I}^T P \mathcal{I} & 0 \\
\mathcal{J}^T D^T \Psi & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\vec{u} \\
-\vec{p} \\
\pi
\end{bmatrix}
= \begin{bmatrix}
0 \\
\mathcal{I}^T P p^n_f
\end{bmatrix}
\]
Compressible black-oil models
Example 1: tracer transport in ideal gas (Lunati & Jenny 2006)

\[ p(0, t) = 1 \text{ bar}, \; p(x, 0) = 10 \text{ bar}, \] coarse grid: 5 blocks, fine grid: 100 cells
Compressible black-oil models
Residuals by domain decomposition

Residual equation:

\[
\begin{bmatrix}
B & C \\
C^T & P
\end{bmatrix}
\begin{bmatrix}
u_{ms} + \hat{u}^{\nu+1} \\
p_{ms} + \hat{p}^{\nu+1}
\end{bmatrix} = 
\begin{bmatrix}
0 \\
PP^n + V\nu
\end{bmatrix}
\]

Localization: \( \hat{u} = \sum_i \hat{u}_i, \hat{p} = \sum_i \hat{p}_i \)
- zero right-hand-side in \( \hat{\Omega}_i \setminus \Omega_i \)
- zero flux BCs on \( \partial \hat{\Omega}_i \)

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 tolerances compared with fine-scale reference solution.
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 $\rightarrow O(n)$ algorithm, cell-by-cell solution
  - local control over nonlinear iterations
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 $\mathcal{O}(n)$ algorithm, cell-by-cell solution
  - local control over nonlinear iterations

Local iterations:

Johansen formation: 27 437 active cells

Global vs local Newton–Raphson solver

<table>
<thead>
<tr>
<th>$\Delta t$</th>
<th>global</th>
<th>local</th>
</tr>
</thead>
<tbody>
<tr>
<td>days</td>
<td>time</td>
<td>iter</td>
</tr>
<tr>
<td>125</td>
<td>2.26</td>
<td>12.69</td>
</tr>
<tr>
<td>250</td>
<td>2.35</td>
<td>12.62</td>
</tr>
<tr>
<td>500</td>
<td>2.38</td>
<td>13.25</td>
</tr>
<tr>
<td>1000</td>
<td>2.50</td>
<td>13.50</td>
</tr>
</tbody>
</table>
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 $\rightarrow O(n)$ 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 → \( O(n) \) 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
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 $\xrightarrow{} \mathcal{O}(n)$ 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
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 \(\mathcal{O}(n)\) 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:

![Water-cut curves](image)

Reference solution
1581 blocks
854 blocks
450 blocks
239 blocks
119 blocks

Pore volume injected

0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Pore volume injected

Reference solution
1581 blocks
854 blocks
450 blocks
239 blocks
119 blocks

0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Pore volume injected
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 $\rightarrow O(n)$ 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:**

- fine grid 11,864 cells
- flow-based 127 blocks
- METIS 175 blocks
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 $\rightarrow$ 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 → different grid for flow and transport
- Method for model reduction:
  - adjoint simulations → 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 → different grid for flow and transport
- Method for model reduction:
  - adjoint simulations → 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

<table>
<thead>
<tr>
<th>Solver</th>
<th>CPU-time (wall clock)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total</td>
</tr>
<tr>
<td>Multigrid</td>
<td>39 min</td>
</tr>
<tr>
<td>Multiscale</td>
<td>17 min</td>
</tr>
</tbody>
</table>
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 $\rightarrow$ different grid for flow and transport
- Method for model reduction:
  - adjoint simulations $\rightarrow$ gradients
  - ensemble simulations with representative basis functions
- Improved model fidelity:
  - subscale resolution
  - multiphysics applications

Fracture corridors

800 x 800

80 x 80 upscaled

80 x 80 multiscale

28 / 30
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 → different grid for flow and transport
- Method for model reduction:
  - adjoint simulations → gradients
  - ensemble simulations with representative basis functions
- Improved model fidelity:
  - subscale resolution
  - multiphysics applications

Stokes–Brinkmann:
Capabilities:

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

Resolved and unresolved questions

Not yet there:
▶ Compressible three-phase black-oil + non-Cartesian grids
▶ Parallelization
▶ Fully implicit formulation
▶ Compositional, thermal, ...
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

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

Flow Physics

Geological representation

Coarse

Detailed

Simple

Complex

Commercial simulators

Fast reservoir simulator

“GeoScale” technology

Large-scale simulation

- Support for time-critical processes
- Optimal model reduction for tradeoff between time and accuracy
- Parallelization
- Multimillion reservoir cells

- Split fine / coarse scales
- Very fast
- Near-well modeling

Flow Physics

Geological representation

Coarse

Detailed

Simple

Complex

Super-fast lightweight simulation

Lightweight simulation

Fast reservoir simulator

“GeoScale” technology

Large-scale simulation

- Support for time-critical processes
- Optimal model reduction for tradeoff between time and accuracy
- Parallelization
- Multimillion reservoir cells

- Split fine / coarse scales
- Very fast
- Near-well modeling