Use of Multiscale Methods to Bypass Upscaling

Or as a Means to Provide Fast and Approximate Flow Responses in Optimization Workflows?

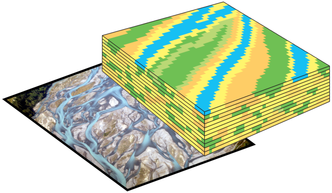
Knut–Andreas Lie

SINTEF ICT, Dept. Applied Mathematics

Workshop on Numerical Discretization and Upscaling Methods, Princeton, November 1-2

Research group

- 3 researchers
- 4 postdocs
- 2–3 PhD students
- 3 programmers



Collaboration with national and international partners in industry and academia

Research vision:

Direct simulation of complex grid models of highly heterogeneous and fractured porous media - a technology that bypasses the need for upscaling.

http://www.math.sintef.no/GeoScale/

Applications:

- Validation during development of geomodels
- Fast simulations of multiple realizations
- Optimization of production, well placement, etc
- History matching
- Geological storage of CO_2

Funding:

- Strategic research grant and PhD/postdoc grants
- Research grants with end-user involvement
- Industry projects

Direct Simulation on Geomodels

How to approach this vision ...

Research vision:

Direct simulation of complex grid models of highly heterogeneous and fractured porous media - a technology that bypasses the need for upscaling.

Geological models as direct input to ....

.... efficient multiscale simulation techniques

Research vision:

Direct simulation of complex grid models of highly heterogeneous and fractured porous media - a technology that bypasses the need for upscaling.

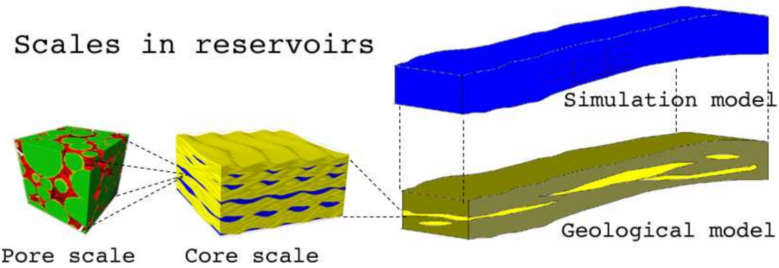
Geological models as direct input to ....

.... efficient multiscale simulation techniques

Physical Scales in Porous Media Flow
... one cannot resolve them all at once

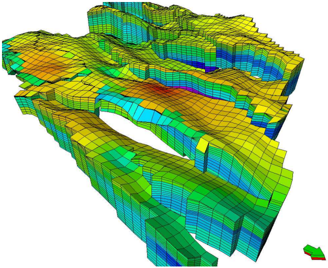
The scales that impact fluid flow in oil reservoirs range from

- the micrometer scale of pores and pore channels
- via dm–m scale of well bores and laminae sediments
- to sedimentary structures that stretch across entire reservoirs.



Geomodels:

- are articulations of the experts perception of the reservoir
- describe the reservoir geometry (horizons, faults, etc)
- give rock parameters (e.g., permeability $\mathbf{K}$ and porosity ϕ) that determine flow



In the following: the term "geomodel" will designate a grid model where rock properties have been assigned to each cell

Geological Models as Direct Input to Simulation
The impact of rock properties

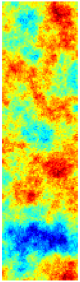
Rock properties are used as parameters in flow models

- Permeability $\mathbf{K}$ spans many length scales and have multiscale structure

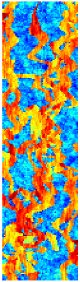
$$\max \mathbf{K} / \min \mathbf{K} \sim 10^3\text{–}10^{10}$$

- Details on all scales impact flow

Ex: Brent sequence



Tarbert



Upper Ness

Challenges:

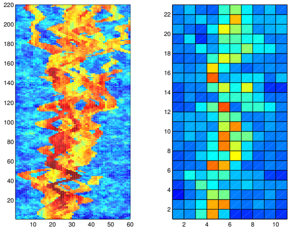
- How much details should one use?
- Need for good linear solvers, preconditioners, etc.

Gap in resolution:

- High-resolution geomodels may have $10^7 - 10^9$ cells
- Conventional simulators are capable of about $10^5 - 10^6$ cells

Traditional solution: upscaling of parameters

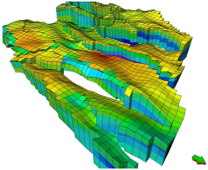
- Upscaling the geomodel is not always the answer
  - Loss of details and lack of robustness
  - Bottleneck in the workflow
- Need for fine-scale computations?
- In the future: need for multiphysics on multiple scales?



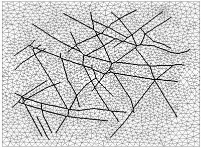
Challenges:

- 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

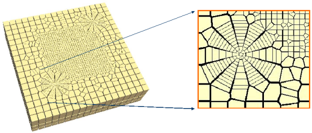
Corner point:



Tetrahedral:



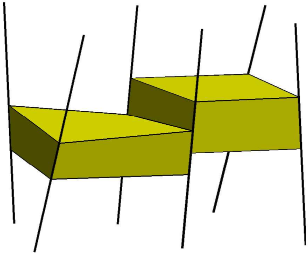
PEBI:

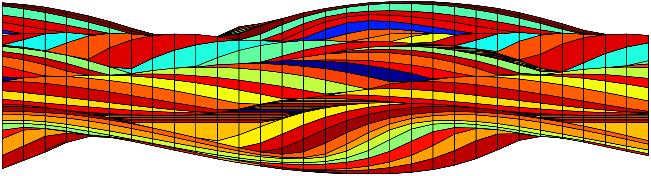


Corner-Point Grids
Industry standard for modelling complex reservoir geology

Specified in terms of:

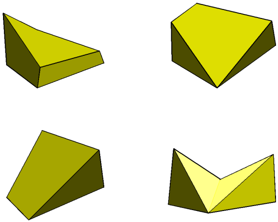
- areal 2D mesh of vertical or inclined pillars
- each volumetric cell is restriced by four pillars
- each cell is defined by eight corner points, two on each pillar



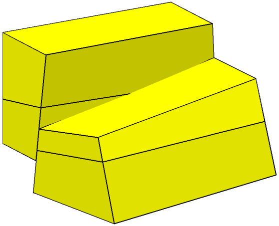


Discretisation on Corner-Point Grids
Cell geometries are challenging from a discretization point-of-view

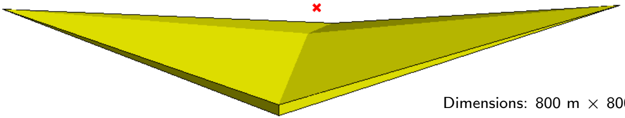
Skewed and deformed grid blocks:



Non-matching cells:



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



Dimensions: 800 m $\times$ 800 m $\times$ 0.25 m

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

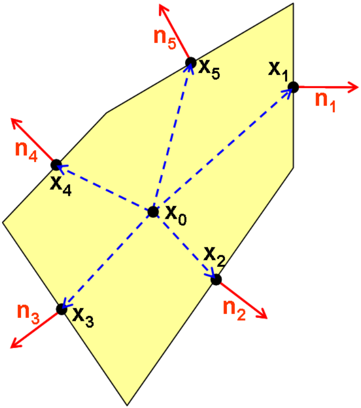
Key features:

- 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 $\mathbf{v} = (v_1, v_2, \dots, v_n)^T$ as:

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

where $\mathbf{p} = (p_1, p_2, \dots, p_n)^T$.

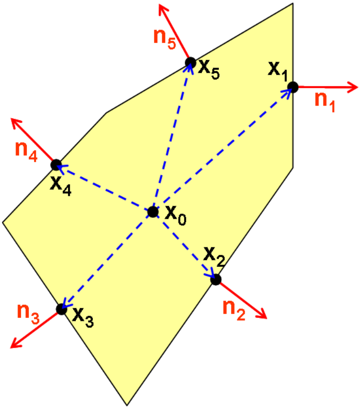


Express fluxes $\mathbf{v} = (v_1, v_2, \ldots, v_n)^T$ as:

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

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

$$\begin{aligned} v_i &= -A_i \mathbf{n}_i^T \mathbf{K}\mathbf{a} \\ p_i - p_0 &= (\mathbf{x}_i - \mathbf{x_0})^T \mathbf{a}. \end{aligned}$$



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

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

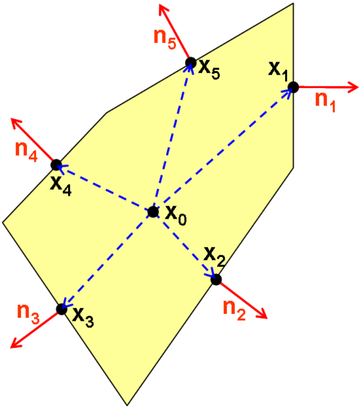
where $\mathbf{p} = (p_1, p_2, \ldots, p_n)^T$.
Impose exactness for any *linear* pressure field $p = \mathbf{x}^T\mathbf{a} + c$ (which gives velocity equal $-\mathbf{K}\mathbf{a}$):

$$\begin{aligned} v_i &= -A_i \mathbf{n}_i^T \mathbf{K}\mathbf{a} \\ p_i - p_0 &= (\mathbf{x}_i - \mathbf{x}_0)^T \mathbf{a}. \end{aligned}$$

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

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

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



The Mimetic Finite Difference Method

Brezzi *et al.*, 2005

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

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

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

$$\begin{aligned} v_i &= -A_i \mathbf{n}_i^T \mathbf{K}\mathbf{a} \\ p_i - p_0 &= (\mathbf{x}_i - \mathbf{x_0})^T \mathbf{a}. \end{aligned}$$

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

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

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

Family of valid solutions:

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

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

The Mimetic Finite Difference Method
Brezzi *et al.*, 2005

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

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

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

$$\begin{aligned} v_i &= -A_i \mathbf{n}_i^T \mathbf{K}\mathbf{a} \\ p_i - p_0 &= (\mathbf{x}_i - \mathbf{x}_0)^T \mathbf{a}. \end{aligned}$$

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

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

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

Family of valid solutions:

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

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

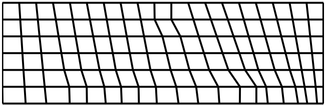
Imposing continuity across edges/faces and conservation yields a *hybrid* system:

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} & \mathbf{C} \\ \mathbf{B}^T & \mathbf{O} & \mathbf{O} \\ \mathbf{C}^T & \mathbf{O} & \mathbf{O} \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ \mathbf{p}_{\text{cells}} \\ \mathbf{p}_{\text{faces}} \end{pmatrix} = \text{RHS}$$

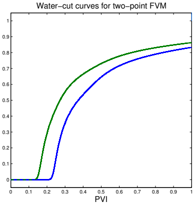
$$\Downarrow$$

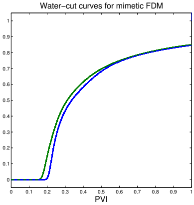
Reduces to s.p.d. system for $\mathbf{p}_{\text{faces}}$.

Standard method + skew grids = grid-orientation effects

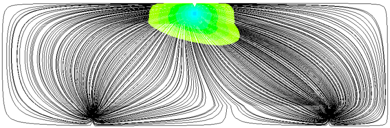


$\mathbf{K}$: homogeneous and isotropic, symmetric well pattern
$\longrightarrow$ symmteric flow

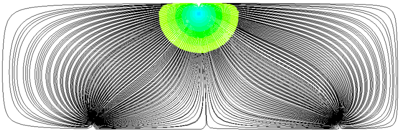




Streamlines with two-point method



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.

Direct Simulation on Geomodels

How to approach this vision ...

Research vision:

Direct simulation of complex grid models of highly heterogeneous and fractured porous media - a technology that bypasses the need for upscaling.

Geological models as direct input to ....

.... efficient multiscale simulation techniques

Research vision:

Direct simulation of complex grid models of highly heterogeneous and fractured porous media - a technology that bypasses the need for upscaling.

Geological models as direct input to ....

.... efficient multiscale simulation techniques

multiscale pressure solver

fast transport solvers

Key Technology: Multiscale Pressure Solvers

Efficient flow solution on complex grids – without upscaling

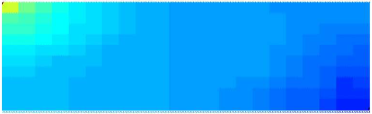
Basic idea:

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

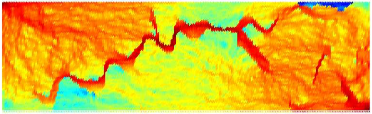
Example: Layer 36 from SPE 10

Pressure field computed with mimetic FDM

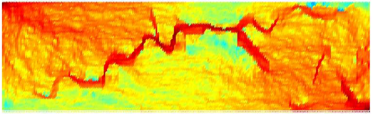
Pressure field computed with 4M



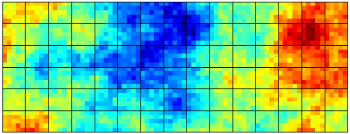
Velocity field computed with mimetic FDM



Velocity field computed with 4M

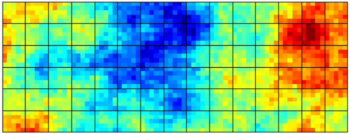


Standard upscaling:



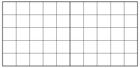
From Upscaling to Multiscale Methods

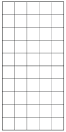
Standard upscaling:



$\Downarrow$

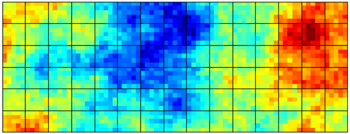
Coarse grid blocks:





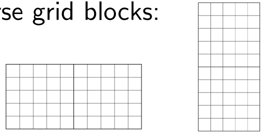
From Upscaling to Multiscale Methods

Standard upscaling:



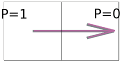
⇓

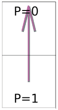
Coarse grid blocks:



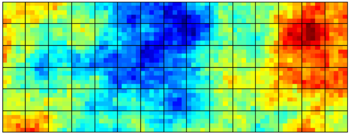
⇓

Flow problems:



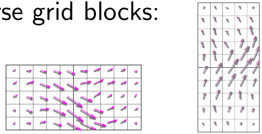


Standard upscaling:



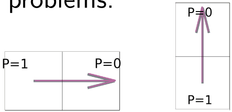
⇓

Coarse grid blocks:

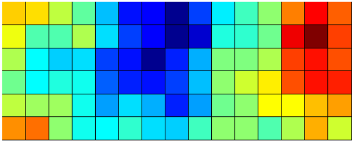


⇓ ⇑

Flow problems:

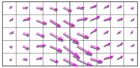


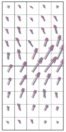
Standard upscaling:



$\Downarrow \quad \Uparrow$

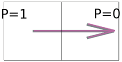
Coarse grid blocks:

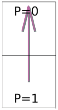




$\Downarrow \quad \Uparrow$

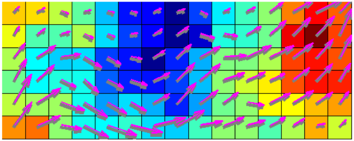
Flow problems:





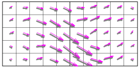
From Upscaling to Multiscale Methods

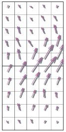
Standard upscaling:



⇓ ⇑

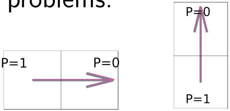
Coarse grid blocks:





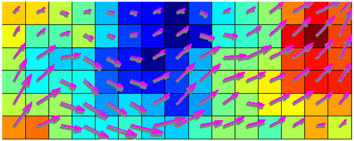
⇓ ⇑

Flow problems:



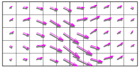
From Upscaling to Multiscale Methods

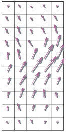
Standard upscaling:



⇓ ⇑

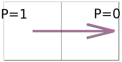
Coarse grid blocks:

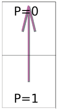




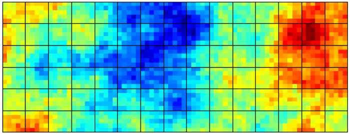
⇓ ⇑

Flow problems:



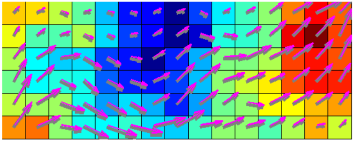


Multiscale method:



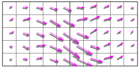
From Upscaling to Multiscale Methods

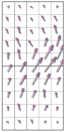
Standard upscaling:



⇓ ⇑

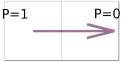
Coarse grid blocks:

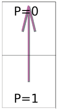




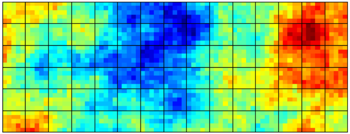
⇓ ⇑

Flow problems:



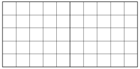


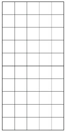
Multiscale method:



⇓

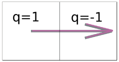
Coarse grid blocks:

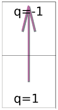




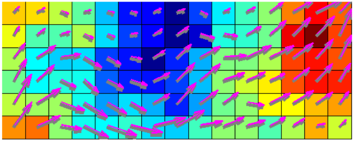
⇓

Flow problems:



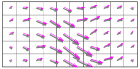


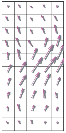
Standard upscaling:



$\Downarrow$ $\Uparrow$

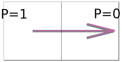
Coarse grid blocks:

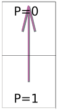




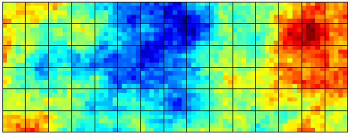
$\Downarrow$ $\Uparrow$

Flow problems:



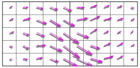


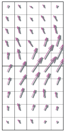
Multiscale method:



$\Downarrow$

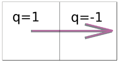
Coarse grid blocks:

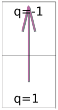




$\Downarrow$ $\Uparrow$

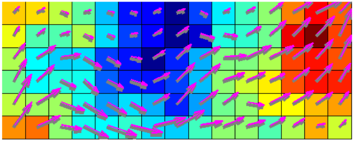
Flow problems:





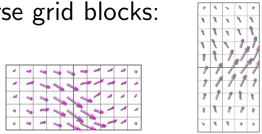
From Upscaling to Multiscale Methods

Standard upscaling:



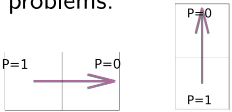
$\Downarrow$ $\Uparrow$

Coarse grid blocks:

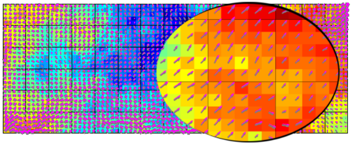


$\Downarrow$ $\Uparrow$

Flow problems:

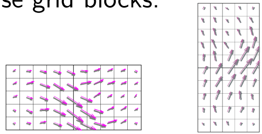


Multiscale method:



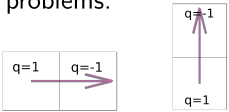
$\Downarrow$ $\Uparrow$

Coarse grid blocks:



$\Downarrow$ $\Uparrow$

Flow problems:



Multiscale Mixed/Mimetic Pressure Solvers

Advantages

Ability to handle industry-standard grids

- highly skewed and degenerate cells
- non-matching cells and unstructured connectivities

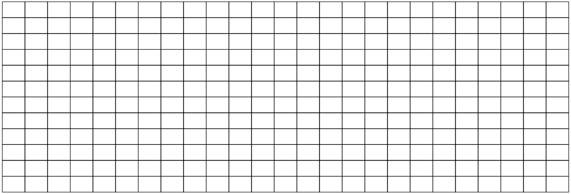
Compatible with current solvers

- can be built on top of commercial/inhouse solvers
- can utilize existing linear solvers

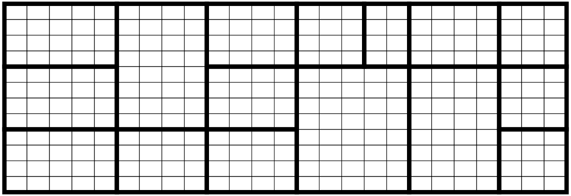
More efficient than standard solvers

- automated generation of coarse simulation grids
- easy to parallelize
- less memory requirements than fine-grid solvers

Assume we are given a *fine* grid with permeability and porosity attached to each fine-grid block:

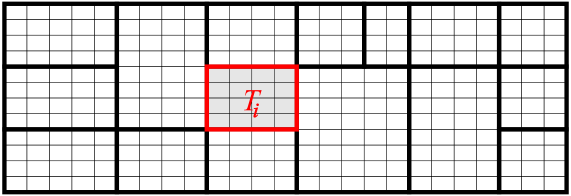


Assume we are given a *fine* grid with permeability and porosity attached to each fine-grid block:



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

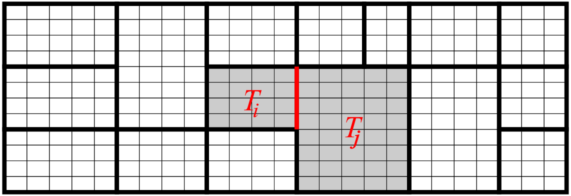
Assume we are given a *fine* grid with permeability and porosity attached to each fine-grid block:



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

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

Assume we are given a *fine* grid with permeability and porosity attached to each fine-grid block:



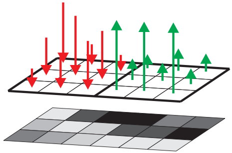
We construct a *coarse* grid, and choose the discretisation spaces U and V^{ms} such that:

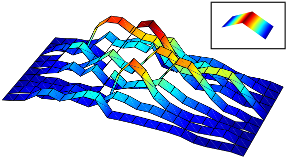
- For each coarse block T_i, there is a basis function $\phi_i \in U$.
- For each coarse edge Γ_{ij}, there is a basis function $\psi_{ij} \in V^{ms}$.

Velocity basis function ψ_{ij}: unit flow through Γ_{ij} defined as

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

and no flow $\psi_{ij} \cdot n = 0$ on $\partial(T_i \cup T_j)$.





Multiscale space:
$V^{ms} = \text{span}\{\psi_{ij} = -\lambda K \nabla \phi_{ij}\}$

Global velocity:

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

Discretisation matrices:

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

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

$$c_{ij} = \int_\Omega \phi_j \nabla \cdot \psi_i \, dx$$

Subgrid solvers on corner-point grids

- MFEM on tetrahedral subdivision of hexahedral cells
- TPFA or MPFA finite-volume methods
- mimetic finite-difference methods

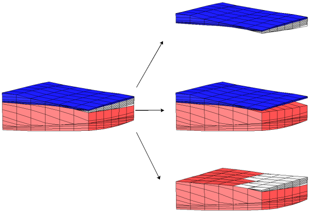
can all be recast in mixed form as a *discrete* approximation of the bilinear form

$$\int_\Omega u^T (\lambda \mathbf{K})^{-1} v \approx \sum_{\Gamma_i} \mathbf{u}_i \mathbf{M}_i \mathbf{v}_i,$$

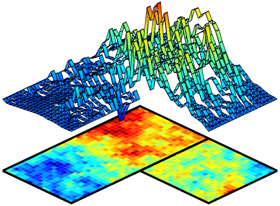
using fluxes $\mathbf{u}_i$ and $\mathbf{v}_i$ over cell-faces Γ_i

At initial time:

Detect all adjacent blocks



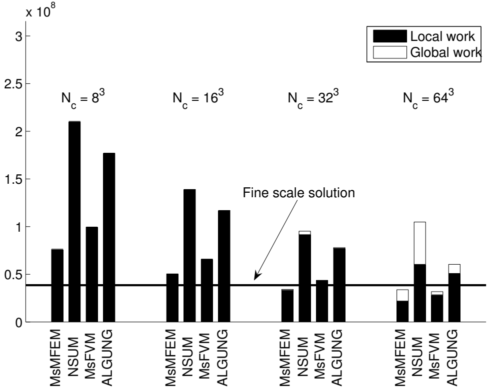
Compute ψ for each domain



For each time step:

- (Recompute basis functions)
- Assemble and solve coarse-grid system
- Recover fine-grid velocity
- Solve fluid-transport equations

Computational efficiency on a $128 \times 128 \times 128$ example



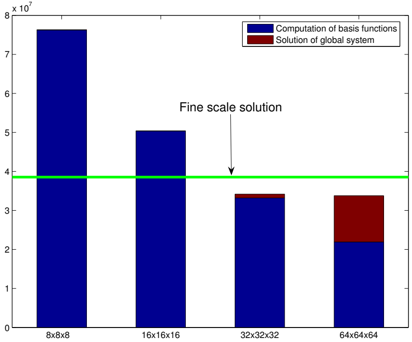
Multiscale solvers are not necessarily faster than a good direct solver for a *single* pressure solution

Direct solution may be more efficient, so why bother with multiscale?

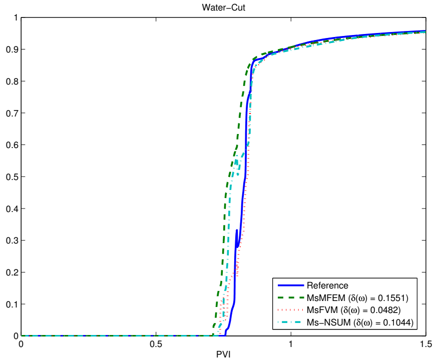
- Full simulation: $\mathcal{O}(10^2)$ time steps.
- Basis functions need not be recomputed

Also:

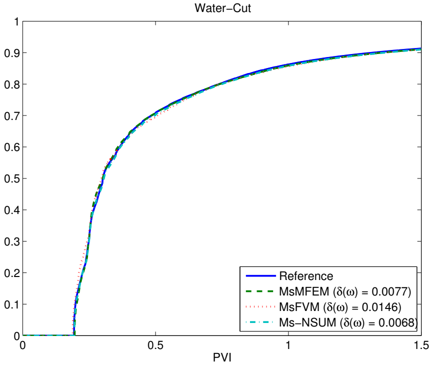
- Possible to solve very large problems
- Easy parallelization



Water cuts obtained by *never* updating basis functions:

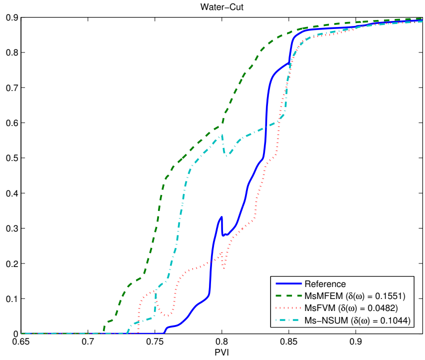


favorable ($M = 0.1$)

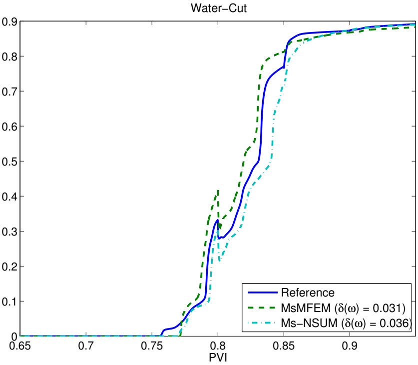


unfavorable ($M = 10.0$)

Improved accuracy by *adaptive* updating of basis functions:

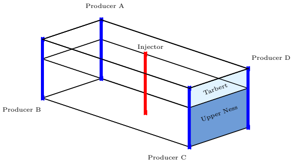


no updating



adaptive updating

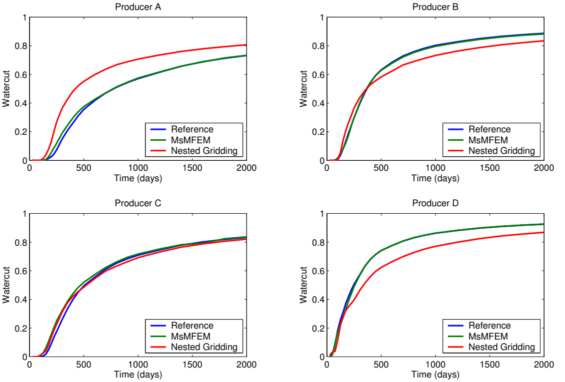
SPE 10, Model 2:



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

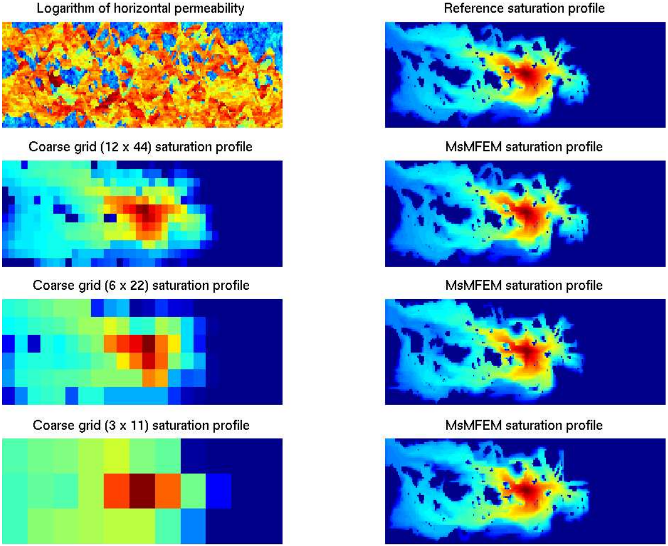
Water-cut curves at the four producers

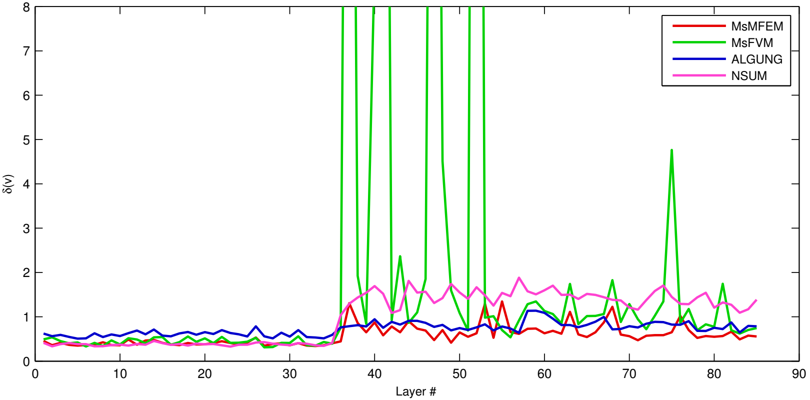


upscaling/downscaling, 4M/streamlines, fine grid

Application 1: Fast Reservoir Simulation on Geomodels

Robustness wrt coarse grid on Layer 85

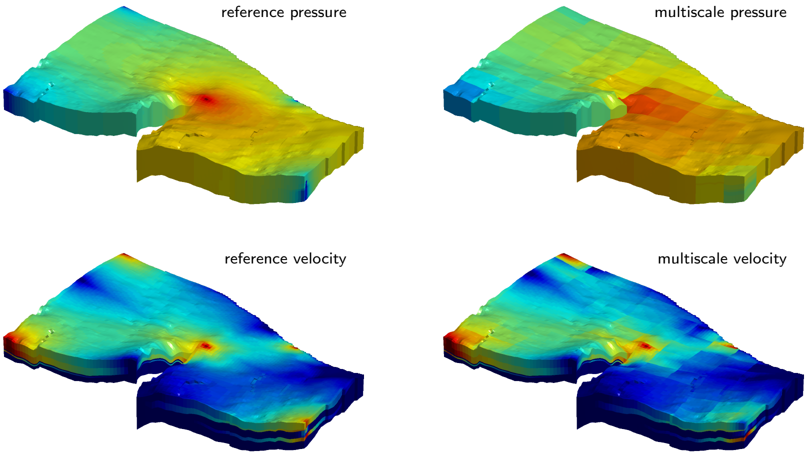




MsMFEM
MsFVM
ALGUNG
NSUM
δ(v)
Layer #
0
1
2
3
4
5
6
7
8
10
20
30
40
50
60
70
80
90

Application 1: Fast Reservoir Simulation on Geomodels

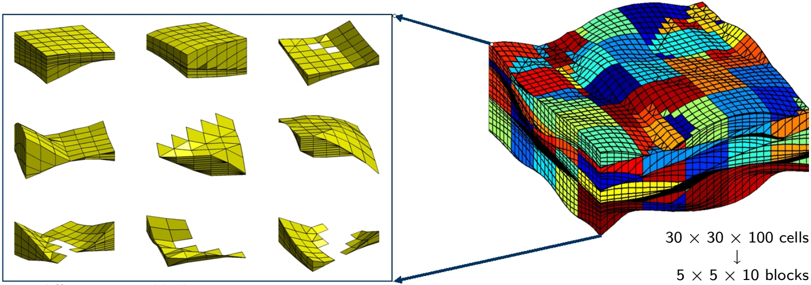
Three-phase black-oil simulation on real-field model



Application 2: Automated Generation of Coarse Grids

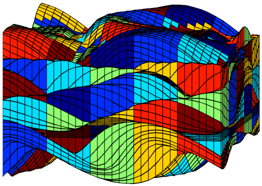
Block in coarse grid: connected set of cells from geomodel

Coarse grid = uniform partitioning in index space

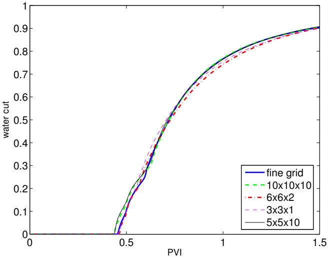


9 different coarse blocks

Coarse grid	Isotropic	Anisotropic	Heterogeneous
$10 \times 10 \times 10$	0.026	0.143	0.094
$6 \times 6 \times 2$	0.042	0.169	0.141
$3 \times 3 \times 1$	0.065	0.127	0.106
$5 \times 5 \times 10$	0.060	0.138	0.142

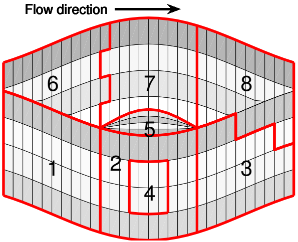


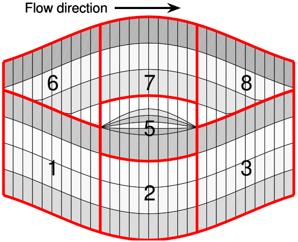
Logically $5 \times 5 \times 10$



Water cut

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

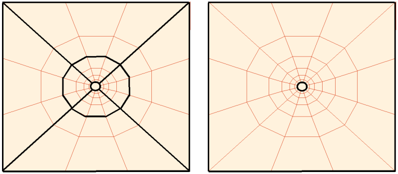


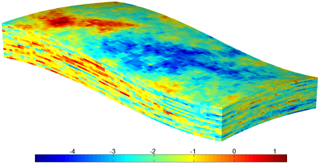


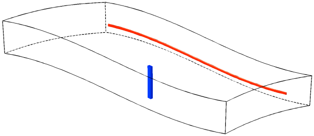
Application 3: Near-Well Modelling / Improved Well-Model

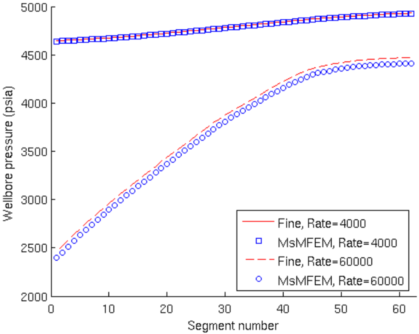
Fine grid to annulus,
block for each well segment

- No well model needed.
- Drift-flux wellbore flow.





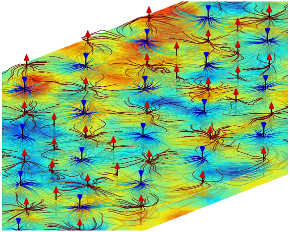


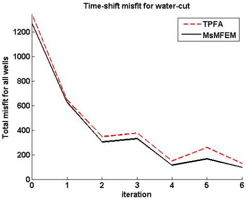


Assimilation of production data to calibrate model

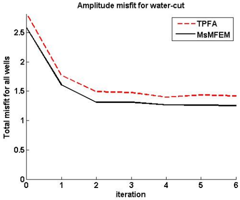
- 1 million cells, 32 injectors, and 69 producers
- 2475 days $\approx$ 7 years of water-cut data

Analytical sensitivities along streamlines + travel-time inversion (quasi-linearization of misfit functional)





Time-residual



Amplitude-residual

Computation time: $\sim$ 17 min on a desktop PC (6 iterations).

Three-phase black-oil:

- Up and running on real-field models from industry
- More work is needed with respect to accuracy (strong pressure gradients, adaptivitiy, etc)

Modelling of wells:

- Several solutions (one block per perforation, wells created as inner boundary conditions, etc)
- Will investigate adaptivity to increase robustness/accuracy

Fractures and faults:

- Using mimetic: mostly a question of grid preprocessing
- Inclusion of capillary forces
- Extentions to Stokes–Brinkman using Taylor–Hood elements

Direct Simulation on Geomodels

How to approach this vision ...

Research vision:

Direct simulation of complex grid models of highly heterogeneous and fractured porous media - a technology that bypasses the need for upscaling.

Geological models as direct input to ....

.... efficient multiscale simulation techniques

multiscale pressure solver

fast transport solvers

Research vision:

Direct simulation of complex grid models of highly heterogeneous and fractured porous media - a technology that bypasses the need for upscaling.

Geological models as direct input to ....

.... efficient multiscale simulation techniques

multiscale pressure solver

fast transport solvers

nonunform coarsening

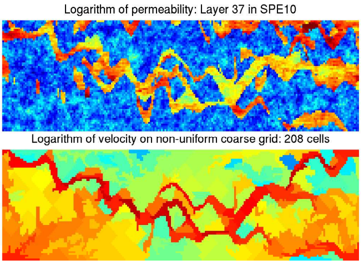
reordering

streamlines

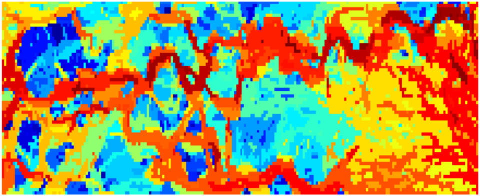
Task

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 accuracy loss.

SPE 10, Layer 37



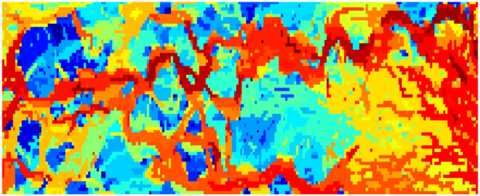
Step 1: Segment ln $|v|$ into N level sets



Robust choice: $N = 10$

Step 1: 1411 cells

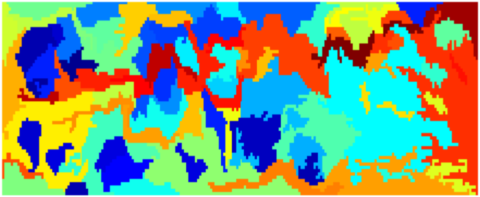
Step 1: Segment ln $|v|$ into N level sets



Robust choice: $N = 10$

Step 1: 1411 cells

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

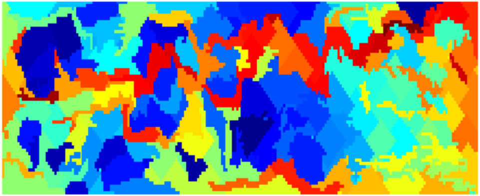


Merge B and B' if

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

Step 2: 94 cells

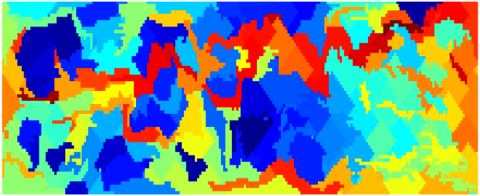
Step 3: Refine blocks with too much flow ($\int_B \ln|v|dx > C$)



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

Step 3: 249 cells

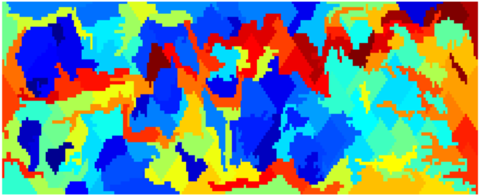
Step 3: Refine blocks with too much flow ($\int_B \ln |v| dx > C$)



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

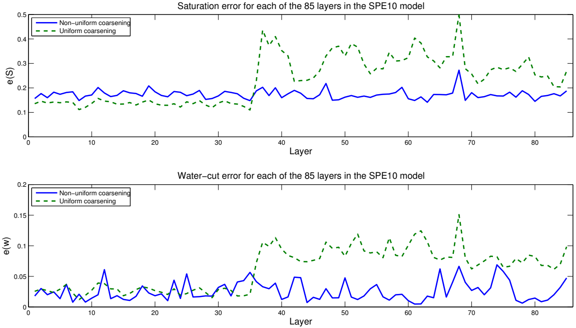
Step 3: 249 cells

Step 4: Combine small blocks with a neighbouring block



Step 2 repeated

Step 4: 160 cells

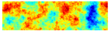
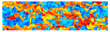


Geomodel:
$60 \times 220 = 13\,200$

Uniform:
$15 \times 44 = 660$

Non-uniform:
619–734 blocks

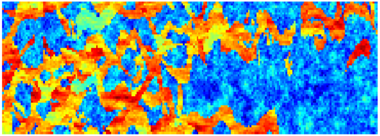
Observations:

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

Adaptive Model Reduction of Transport Grids

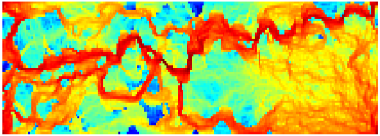
Example 1: Layer 68, SPE10, 5-spot well pattern

Logarithm of permeability: Layer 68

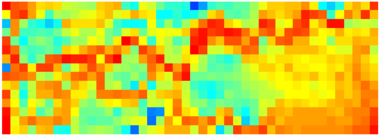


Geomodel: 13200 cells

Logarithm of velocity on geomodel

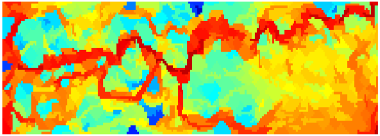


Logarithm of velocity on Cartesian coarse grid



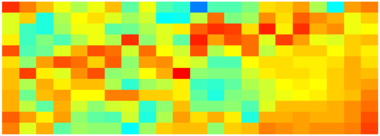
Coarse grid: 660 cells

Logarithm of velocity on non-uniform coarse grid



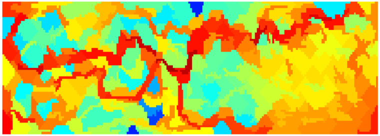
Coarse grid: 649 cells

Logarithm of velocity on Cartesian coarse grid



Coarse grid: 264 cells

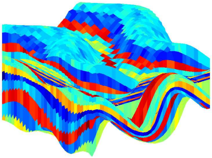
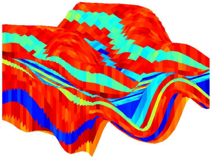
Logarithm of velocity on non-uniform coarse grid



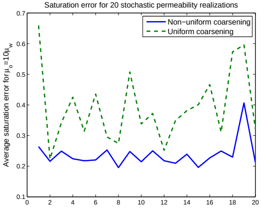
Coarse grid: 257 cells

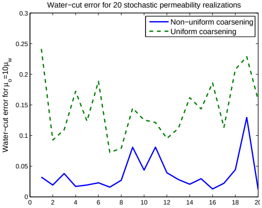
Adaptive Model Reduction of Transport Grids

Example 2: Depositional bed, 20 lognormal realizations, q5-spot

⇐ 2 realizations
15206 cells





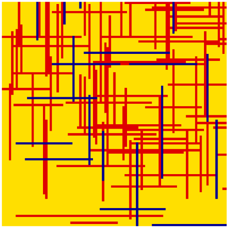
Uniform:
838 blocks

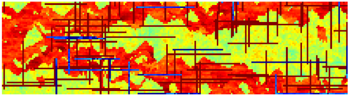
Non-uniform:
647–704 blocks

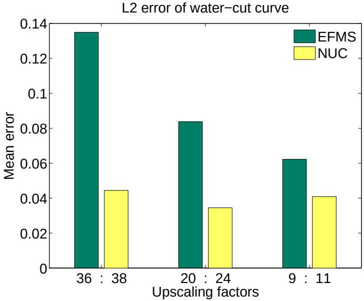
Observations:

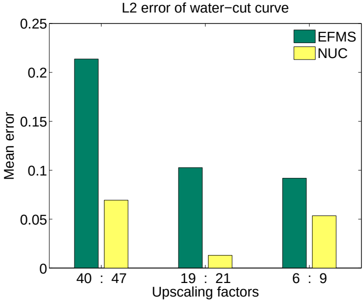
- Coarsening algorithm applicable to unstructured grids
  — accuracy consistent with observations for SPE10 models.
- Results obtained with uniform grid (in index space) inaccurate.

Adaptive Model Reduction of Transport Grids
Example 3: Fracture networks









Opportunities

- Utilization within optimization and data integration workflows?
- Adaptive model reduction as alternative to proxy models?

Unresolved questions

- Capillary forces – initial ideas are promising
- Three-phase black oil – not tested yet
- Applicability to grids with large differences in cell sizes

Research vision:

Direct simulation of complex grid models of highly heterogeneous and fractured porous media - a technology that bypasses the need for upscaling.

Geological models as direct input to ....

.... efficient multiscale simulation techniques

multiscale pressure solver

fast transport solvers

nonunform coarsening

reordering

streamlines

Direct Simulation on Geomodels

How to approach this vision ...

Research vision:

Direct simulation of complex grid models of highly heterogeneous and fractured porous media - a technology that bypasses the need for upscaling.

Geological models as direct input to ....

.... efficient multiscale simulation techniques

multiscale pressure solver

fast transport solvers

nonunform coarsening

reordering

streamlines

Fast Methods Based on Topological Sorting

Flow models are typically on the form

$$au + \mathbf{v} \cdot \nabla f(u) = b(u), \qquad u \text{ given on } \partial\Omega^-$$

Examples:

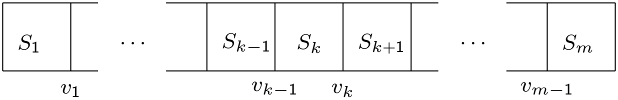
- Steady-state tracer: $\mathbf{v} \cdot \nabla c = 0$
- Time-of-flight: $\mathbf{v} \cdot \nabla \tau = \phi$
- Implicit schemes for multiphase/multicomponent transport:

$$S^{n+1} + \Delta t \mathbf{v} \cdot \nabla f(S^{n+1}) = \Delta t q(S^{n+1}) + S^n$$

Basic idea

1. Utilize the unidirectional flow property to solve cell by cell
2. High order: discontinous Galerkin + upwind flux to preserve unidirectional flow property

Fast Methods Based on Topological Sorting
Motivation: Implicit scheme in 1D

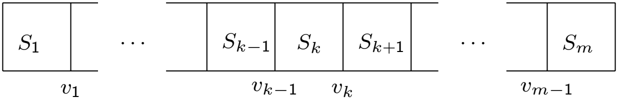


First-order upwind scheme ($v_k > 0$, $\forall k$):

$$\frac{\phi}{\Delta t}(S_k^{n+1} - S_k^n) + \frac{1}{\Delta x}\Big(v_{k-1} f(S_{k-1}^{n+1}) - v_k f(S_k^{n+1})\Big) = Q_k(S_k^{n+1})$$

Lower triangular matrix $\Longrightarrow$ equations can be solved in sequence

Fast Methods Based on Topological Sorting
Motivation: Implicit scheme in 1D



First-order upwind scheme ($v_k > 0$, $\forall k$):

$$\frac{\phi}{\Delta t}(S_k^{n+1} - S_k^n) + \frac{1}{\Delta x}\Big(v_{k-1} f(S_{k-1}^{n+1}) - v_k f(S_k^{n+1})\Big) = Q_k(S_k^{n+1})$$

Lower triangular matrix $\Longrightarrow$ equations can be solved in sequence

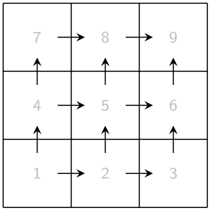
Multidimensions

Same idea applies by using *a topological sort* of the directed graph of fluxes

Fast Methods Based on Topological Sorting

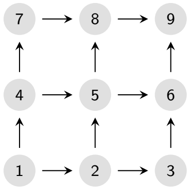
Motivation: Homogeneous quarter-five spot in 2D

Grid and flux matrix

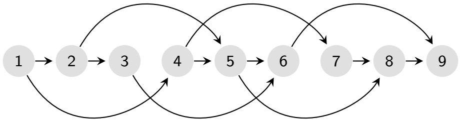


$$V = \begin{bmatrix} * & & & & & & & & \\ * & * & & & & & & & \\ & * & * & & & & & & \\ * & & & * & & & & & \\ & * & & * & * & & & & \\ & & * & & * & * & & & \\ & & & * & & & * & & \\ & & & & * & & * & * & \\ & & & & & * & & * & * \end{bmatrix}$$

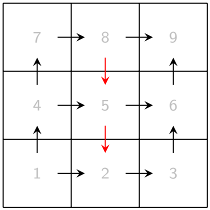
Graph interpretation



Flattened graph (unidirectional)

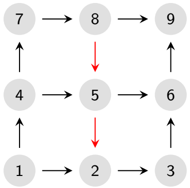


Grid and flux matrix

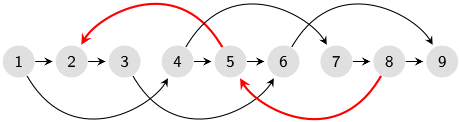


$$V = \begin{bmatrix} * & & & & & & & & \\ * & * & & & {\color{red}*} & & & & \\ & * & * & & & & & & \\ * & & & * & & & & & \\ & & & * & * & & & {\color{red}*} & \\ & & * & & * & * & & & \\ & & & * & & & * & & \\ & & & & & & * & * & \\ & & & & & * & & * & * \end{bmatrix}$$

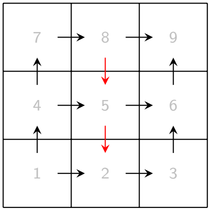
Graph interpretation



Flattened graph

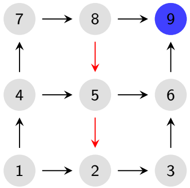


Grid and flux matrix



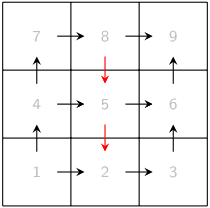
$$V = \begin{bmatrix} * & & & & & & & & \\ * & * & & & * & & & & \\ & * & * & & & & & & \\ * & & & * & & & & & \\ & & & * & * & & & * & \\ & & * & & * & * & & & \\ & & & * & & & * & & \\ & & & & & & * & * & \\ & & & & & * & & * & * \end{bmatrix}$$

Graph interpretation and topological sorting



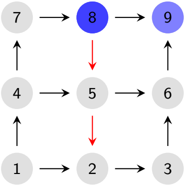
Flattened graph (unidirectional)

Grid and flux matrix



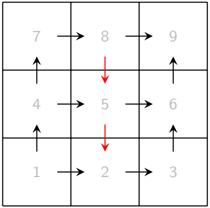
$$V = \begin{bmatrix} * & & & & & & & & \\ * & * & & & {\color{red}*} & & & & \\ & * & * & & & & & & \\ * & & & * & & & & & \\ & & & * & * & & & {\color{red}*} & \\ & & * & & * & * & & & \\ & & & * & & & * & & \\ & & & & & & * & * & \\ & & & & & * & & * & * \end{bmatrix}$$

Graph interpretation and topological sorting



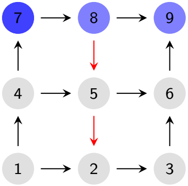
Flattened graph (unidirectional)

Grid and flux matrix



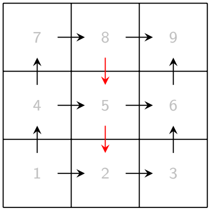
$$V = \begin{bmatrix} * & & & & & & & & \\ * & * & & & * & & & & \\ & * & * & & & & & & \\ * & & & * & & & & & \\ & & & * & * & & & * & \\ & & * & & * & * & & & \\ & & & * & & & * & & \\ & & & & & & * & * & \\ & & & & & * & & * & * \end{bmatrix}$$

Graph interpretation and topological sorting



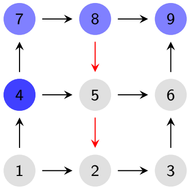
Flattened graph (unidirectional)

Grid and flux matrix



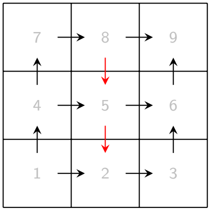
$$V = \begin{bmatrix} * & & & & & & & & \\ * & * & & & \color{red}{*} & & & & \\ & * & * & & & & & & \\ * & & & * & & & & & \\ & & & * & * & & & \color{red}{*} & \\ & & * & & * & * & & & \\ & & & * & & & * & & \\ & & & & & & * & * & \\ & & & & & * & & * & * \end{bmatrix}$$

Graph interpretation and topological sorting



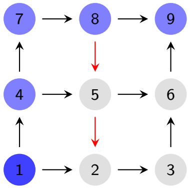
Flattened graph (unidirectional)

Grid and flux matrix



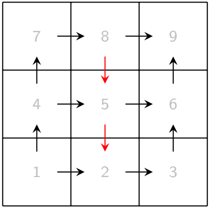
$$V = \begin{bmatrix} * & & & & & & & & \\ * & * & & & {\color{red}*} & & & & \\ & * & * & & & & & & \\ * & & & * & & & & & \\ & & & * & * & & & {\color{red}*} & \\ & & * & & * & * & & & \\ & & & * & & & * & & \\ & & & & & & * & * & \\ & & & & & * & & * & * \end{bmatrix}$$

Graph interpretation and topological sorting



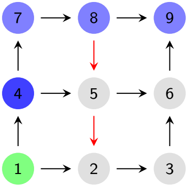
Flattened graph (unidirectional)

Grid and flux matrix



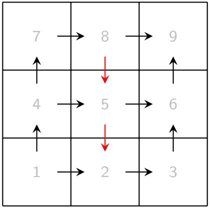
$$V = \begin{bmatrix} * & & & & & & & & \\ * & * & & & {\color{red}*} & & & & \\ & * & * & & & & & & \\ * & & & * & & & & & \\ & & & * & * & & & {\color{red}*} & \\ & & * & & * & * & & & \\ & & & * & & & * & & \\ & & & & & & * & * & \\ & & & & & * & & * & * \end{bmatrix}$$

Graph interpretation and topological sorting



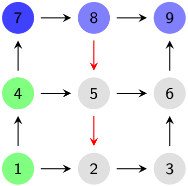
Flattened graph (unidirectional)

Grid and flux matrix



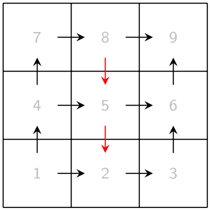
$$V = \begin{bmatrix} * & & & & & & & & \\ * & * & & & * & & & & \\ & * & * & & & & & & \\ * & & & * & & & & & \\ & & & * & * & & & * & \\ & & * & & * & * & & & \\ & & & * & & & * & & \\ & & & & & & * & * & \\ & & & & & * & & * & * \end{bmatrix}$$

Graph interpretation and topological sorting



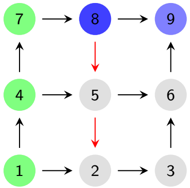
Flattened graph (unidirectional)

Grid and flux matrix



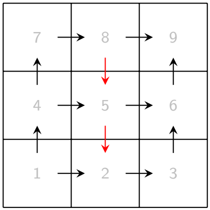
$$V = \begin{bmatrix} * & & & & & & & & \\ * & * & & & {\color{red}*} & & & & \\ & * & * & & & & & & \\ * & & & * & & & & & \\ & & & * & * & & & {\color{red}*} & \\ & & * & & * & * & & & \\ & & & * & & & * & & \\ & & & & & & * & * & \\ & & & & & * & & * & * \end{bmatrix}$$

Graph interpretation and topological sorting



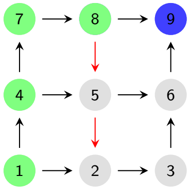
Flattened graph (unidirectional)

Grid and flux matrix



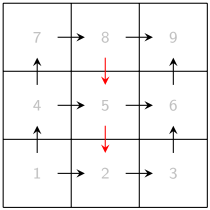
$$V = \begin{bmatrix} * & & & & & & & & \\ * & * & & & \color{red}{*} & & & & \\ & * & * & & & & & & \\ * & & & * & & & & & \\ & & & * & * & & & \color{red}{*} & \\ & & * & & * & * & & & \\ & & & * & & & * & & \\ & & & & & & * & * & \\ & & & & & * & & * & * \end{bmatrix}$$

Graph interpretation and topological sorting



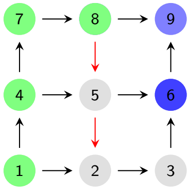
Flattened graph (unidirectional)

Grid and flux matrix



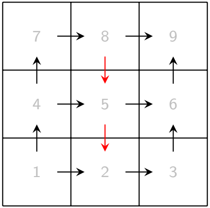
$$V = \begin{bmatrix} * & & & & & & & & \\ * & * & & & {\color{red}*} & & & & \\ & * & * & & & & & & \\ * & & & * & & & & & \\ & & & * & * & & & {\color{red}*} & \\ & & * & & * & * & & & \\ & & & * & & & * & & \\ & & & & & & * & * & \\ & & & & & * & & * & * \end{bmatrix}$$

Graph interpretation and topological sorting



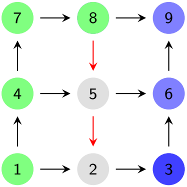
Flattened graph (unidirectional)

Grid and flux matrix



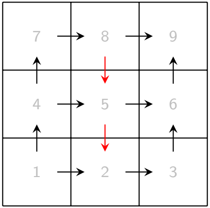
$$V = \begin{bmatrix} * & & & & & & & & \\ * & * & & & {\color{red}*} & & & & \\ & * & * & & & & & & \\ * & & & * & & & & & \\ & & & * & * & & & {\color{red}*} & \\ & & * & & * & * & & & \\ & & & * & & & * & & \\ & & & & & & * & * & \\ & & & & & * & & * & * \end{bmatrix}$$

Graph interpretation and topological sorting



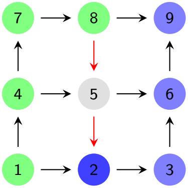
Flattened graph (unidirectional)

Grid and flux matrix



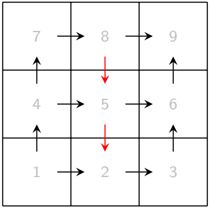
$$V = \begin{bmatrix} * & & & & & & & & \\ * & * & & & {\color{red}*} & & & & \\ & * & * & & & & & & \\ * & & & * & & & & & \\ & & & * & * & & & {\color{red}*} & \\ & & * & & * & * & & & \\ & & & * & & & * & & \\ & & & & & & * & * & \\ & & & & & * & & * & * \end{bmatrix}$$

Graph interpretation and topological sorting



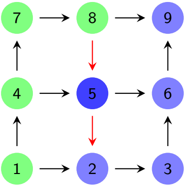
Flattened graph (unidirectional)

Grid and flux matrix



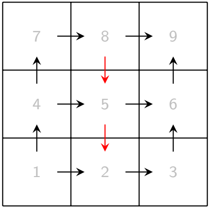
$$V = \begin{bmatrix} * & & & & & & & & \\ * & * & & & {\color{red}*} & & & & \\ & * & * & & & & & & \\ * & & & * & & & & & \\ & & & * & * & & & {\color{red}*} & \\ & & * & & * & * & & & \\ & & & * & & & * & & \\ & & & & & & * & * & \\ & & & & & * & & * & * \end{bmatrix}$$

Graph interpretation and topological sorting



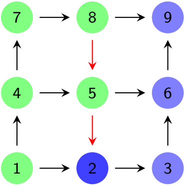
Flattened graph (unidirectional)

Grid and flux matrix

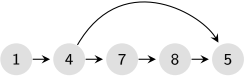


$$V = \begin{bmatrix} * & & & & & & & & \\ * & * & & & * & & & & \\ & * & * & & & & & & \\ * & & & * & & & & & \\ & & & * & * & & & * & \\ & & * & & * & * & & & \\ & & & * & & & * & & \\ & & & & & & * & * & \\ & & & & & * & & * & * \end{bmatrix}$$

Graph interpretation and topological sorting



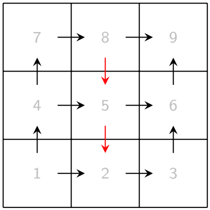
Flattened graph (unidirectional)



Fast Methods Based on Topological Sorting

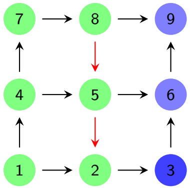
Motivation: Heterogeneous quarter-five spot in 2D

Grid and flux matrix

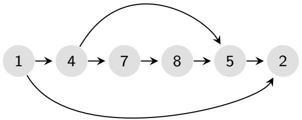


$$V = \begin{bmatrix} \ast & & & & & & & & \\ \ast & \ast & & & \color{red}{\ast} & & & & \\ & \ast & \ast & & & & & & \\ \ast & & & \ast & & & & & \\ & & & \ast & \ast & & & \color{red}{\ast} & \\ & & \ast & & \ast & \ast & & & \\ & & & \ast & & & \ast & & \\ & & & & & & \ast & \ast & \\ & & & & & \ast & & \ast & \ast \end{bmatrix}$$

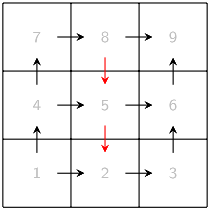
Graph interpretation and topological sorting



Flattened graph (unidirectional)

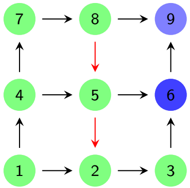


Grid and flux matrix

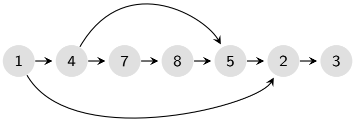


$$V = \begin{bmatrix} * & & & & & & & & \\ * & * & & & \color{red}{*} & & & & \\ & * & * & & & & & & \\ * & & & * & & & & & \\ & & & * & * & & & \color{red}{*} & \\ & & * & & * & * & & & \\ & & & * & & & * & & \\ & & & & & & * & * & \\ & & & & & * & & * & * \end{bmatrix}$$

Graph interpretation and topological sorting



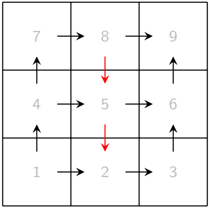
Flattened graph (unidirectional)



Fast Methods Based on Topological Sorting

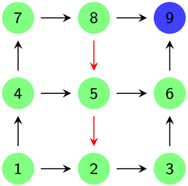
Motivation: Heterogeneous quarter-five spot in 2D

Grid and flux matrix

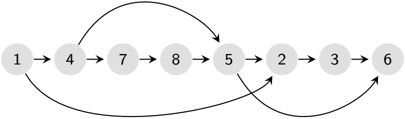


$$V = \begin{bmatrix} * & & & & & & & & \\ * & * & & & {\color{red}*} & & & & \\ & * & * & & & & & & \\ * & & & * & & & & & \\ & & & * & * & & & {\color{red}*} & \\ & & * & & * & * & & & \\ & & & * & & & * & & \\ & & & & & & * & * & \\ & & & & & * & & * & * \end{bmatrix}$$

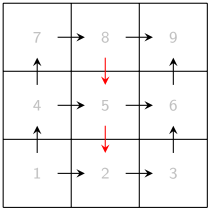
Graph interpretation and topological sorting



Flattened graph (unidirectional)

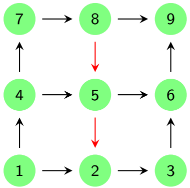


Grid and flux matrix

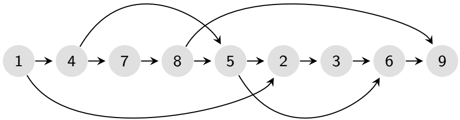


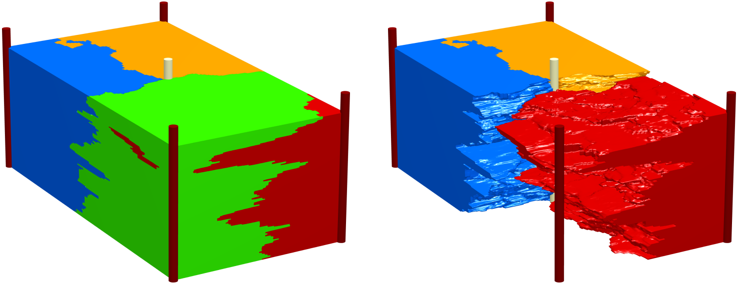
$$V = \begin{bmatrix} * & & & & & & & & \\ * & * & & & {\color{red}*} & & & & \\ & * & * & & & & & & \\ * & & & * & & & & & \\ & & & * & * & & & {\color{red}*} & \\ & & * & & * & * & & & \\ & & & * & & & * & & \\ & & & & & & * & * & \\ & & & & & * & & * & * \end{bmatrix}$$

Graph interpretation and topological sorting



Flattened graph (unidirectional)





SPE 10, Model 2, 60 × 220 × 85 (1.122 million grid blocks)

Stationary tracer:

Solve $\mathbf{v} \cdot \nabla c = q_i$ for i wells
Contour $c = 0.5$

Scheme: dG(n)+upwinding

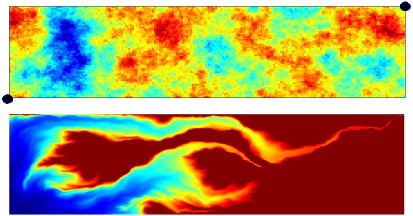
Timings:

order	dof's	time
0	1	3.1 sec
1	4	9.9 sec
2	10	86.8 sec

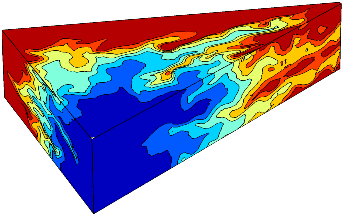
CPU: AMD Athlon X2 4400+

Time-of-flight

- Travel time for a neutral particle injected at boundary/well
- Timelines for single phase flow



Layer 1 of SPE 10



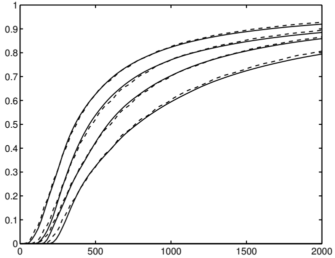
$64 \times 64 \times 16$ grid, vertical q5-spot

Key idea:

Implicit time discretization: element-wise Newton-Raphson solution gives high efficiency.

- $\mathcal{O}(n)$ operations for n unknowns
- Local control over Newton iteration.
- Small memory requirements.
- Small, simple code.
- Well-known *conservative* discretisation.
- Valid for general polyhedral grids.

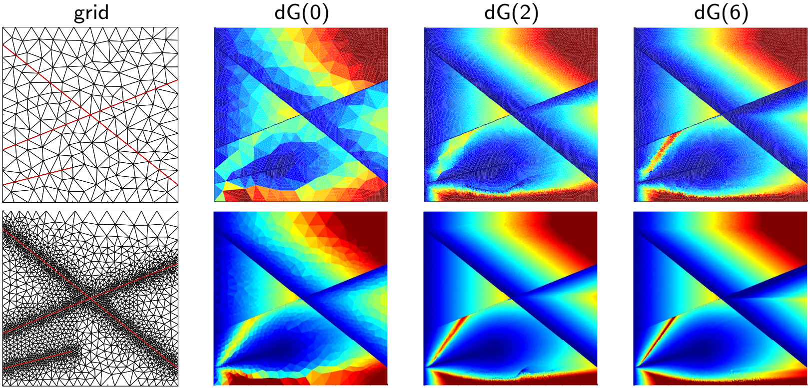
Water-cut, SPE 10, Model 2



Δt=20 days, 2 minutes

Fast Methods Based on Topological Sorting

Application 4: Fracture networks



Opportunities

- Simple way of generating streamline-type data
- Utilization within optimization and data integration workflows?

Unresolved questions

- Efficient linear solvers for various loop sizes
- Elimination/reduction of loops
- How to prevent oscillations for dG(n), $n > 1$
- Operator splitting (capillary/gravity forces)

Direct Simulation on Geomodels
How to approach this vision ...

Research vision:

Direct simulation of complex grid models of highly heterogeneous and fractured porous media - a technology that bypasses the need for upscaling.

Geological models as direct input to ....

.... efficient multiscale simulation techniques

multiscale pressure solver

fast transport solvers

nonunform coarsening

reordering

streamlines

Research vision:

Direct simulation of complex grid models of highly heterogeneous and fractured porous media - a technology that bypasses the need for upscaling.

Geological models as direct input to ....

.... efficient multiscale simulation techniques

multiscale pressure solver

fast transport solvers

nonunform coarsening

reordering

streamlines

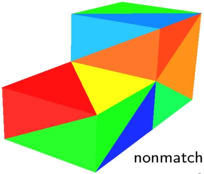
Streamline Simulation
Accurate tracing of streamlines

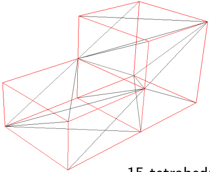
- Pollock (88) – analytic tracing of streamlines on Cartesian grids
  - linear interpolation of fluxes in each direction
  - analytical formula for increment within each cell
- Prevost et al. (02) – extension of Pollock's method to irregular grids
  - isoparametric transformation to unit reference cube
  - linear flux interpolation scaled by Jacobi determinant at element midpoint
  - analytic streamline path mapped to physical space
- Jimenez et al. (05,08)
  - pseudo-time of flight (improved Jacobi)
  - tracing across faults (collapsed cells)
- Matringe et al. (05,06) – higher-order MFEM velocity spaces
- Hægland et al. (07) – corner-velocity interpolation

To handle degenerate and nonmatching cells, we:

- Subdivide faces/edges to make matching corner-point grid
- Use a global Delaney triagularization $\longrightarrow$ each regular hexahedral cell is subdivided into six tetrahedra
- Reconstruct fluxes on tetrahedra
- Trace streamlines analytically within each tetrahedron (constant velocity)

This approach should preserve uniform flow


nonmatching cells


15 tetrahedra

Future apporach: Pollock (or similar) for regular cells, tetrahedral reconstruction for degenerate and nonmatching cells?

Streamline simulation for CO_2:

- efficient 1-D solvers for operator splitting
- transition from injection- to gravity-driven flow
- circular streamlines

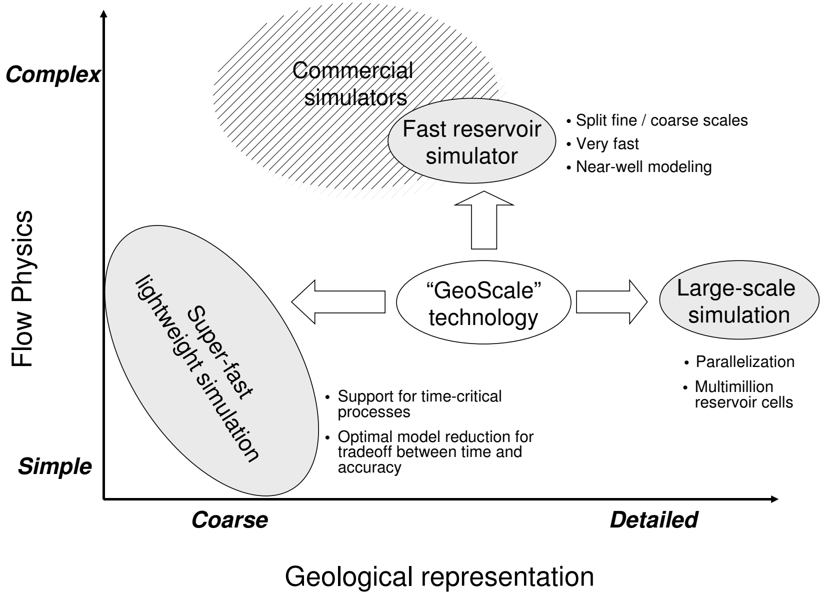
Summary

The research will enable simulator technology to better aid more work processes – by striking balances between reduced computational time, geological representation, and complexity of flow physics

A key to efficient simulation methods – operator splitting:

- Multiscale pressure solvers:
  - Upscaling and downscaling in one step
  - Robust and efficient alternative to upscaling
  - Flow field on coarse, intermediate, and fine grid
- Fast transport solvers, (coarse–intermediate–fine grids):
  - Adaptive nonuniform coarsening
  - Discontinuous Galerkin with topological sorting
  - Streamlines

Summary


Complex
Simple
Flow Physics
Commercial simulators
Fast reservoir simulator
• Split fine / coarse scales
• Very fast
• Near-well modeling
"GeoScale" technology
Large-scale simulation
• Parallelization
• Multimillion reservoir cells
Super-fast lightweight simulation
• Support for time-critical processes
• Optimal model reduction for tradeoff between time and accuracy
Coarse
Detailed
Geological representation