Multiscale Methods for Reservoir Simulation

Knut–Andreas Lie

Department of Mathematics and Cybernetics, SINTEF Digital/
Department of Mathematical Sciences, NTNU, Norway

Multiscale Methods Summer School
June 26–29, 2017, Hasselt, Belgium
1  Introduction

2  Multiscale finite-element methods

3  Multiscale mixed finite-element methods

4  Multiscale finite-volume methods

5  Examples with state-of-the-art method
Multiscale methods

Numerical methods that attempt to model physical phenomena on coarse grids while honoring small-scale features in an appropriate way consistent with the local property of the differential operator.
Multiscale methods versus upscaling

Coarse partitioning:

Localized flow problems:

Compute effective parameters:
Multiscale methods versus upscaling

Coarse partitioning:

Localized flow problems:

Flow field with subresolution:

Flow solution $\rightarrow$ basis functions:
From Poisson’s equation to reservoir simulation

Flow physics

\[ -\nabla(K\nabla p) = q \]

MsFV, MsMFE, 2003

Geology

0 = \partial_t (\phi b_o S_o) + \nabla \cdot (b_o \vec{v}_o) - b_o q_o
0 = \partial_t (\phi b_w S_w) + \nabla \cdot (b_w \vec{v}_w) - b_w q_w
0 = \partial_t [\phi(b_g S_b + b_o r_s o S_o)] + \nabla \cdot (b_g \vec{v}_g) + \nabla \cdot (b_o r_s o \vec{v}_o) - b_g q_g - b_o r_s o q_o
Two main tracks for commercial simulation: multiscale finite-volume (MsFV) and multiscale mixed finite-element (MsMFE) methods

<table>
<thead>
<tr>
<th>Property</th>
<th>MsFE</th>
<th>MsMFE</th>
<th>MsFV</th>
<th>MsRSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conservative velocity field</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Applicable to unstructured grids</td>
<td>×</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Robustness: aspect ratio / high contrast</td>
<td>✓</td>
<td>✓</td>
<td>?</td>
<td>✓</td>
</tr>
<tr>
<td>Compressible flow</td>
<td>×</td>
<td>?</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Systematic error control</td>
<td>✓</td>
<td>?</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Locally smooth</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Partition of unity</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Efficient</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

_Disclaimer_: many methods and a lot of academic research will not be covered in the following.
1. Introduction

2. Multiscale finite-element methods

3. Multiscale mixed finite-element methods

4. Multiscale finite-volume methods

5. Examples with state-of-the-art method
The multiscale finite-element (MsFE) method

Model problem
Variable-coefficient Poisson problem in 1D

\[(K(x)p')' = f, \quad x \in \Omega = [0, 1], \quad p(0) = p(1) = 0,\]

where \(f, k \in L^2(\Omega)\) and \(0 < \alpha < K(x) < \beta\) for all \(x \in \Omega\)

Variational formulation
Find \(p \in H^1_0(\Omega)\) such that

\[a(p, \varphi) = (f, \varphi)\quad \text{for all } \varphi \in H^1_0(\Omega),\]

where \((\cdot, \cdot)\) is the \(L^2\) inner-product and

\[a(p, \varphi) = \int_\Omega K(x) \partial_x p \partial_x \varphi \, dx\]
The MsFE method

\[ x_{i-1} \quad x_i \quad x_{i+1} \]
The MsFE method

For $i = 1, \ldots, n - 1$, we define a basis function $\phi_i \in H^1_0(\Omega)$ by

$$a(\phi_i, \varphi) = 0 \quad \text{for all } \varphi \in H^1_0(B_i \cup B_{i+1}), \quad \phi_i(x_j) = \delta_{ij},$$
For $i = 1, \ldots, n - 1$, we define a basis function $\phi_i \in H^1_0(\Omega)$ by

$$a(\phi_i, \varphi) = 0 \quad \text{for all } \varphi \in H^1_0(B_i \cup B_{i+1}), \quad \phi_i(x_j) = \delta_{ij},$$

Multiscale basis function associated with node $x_i$ is given as

$$-(K(x) \partial_x \phi'_i(x))' = 0, \quad x \in [x_{i-1}, x_{i+1}] = B_i \cup B_{i+1}$$

Obviously, $K(x) \phi'_i = C$, for some constant $C$
The MsFE method

Integrating over $B_i$ and using the prescribed values $\phi_i(x_{i-1}) = 0$ and $\phi_i(x_i) = 1$ gives

$$\int_{x_{i-1}}^{x_i} \phi_i'(x) \, dx = \phi_i(x_i) - \phi_i(x_{i-1}) = 1 = \int_{x_{i-1}}^{x_i} \frac{C}{K(x)} \, dx$$
The MsFE method

Integrating over $B_i$ and using the prescribed values $\phi_i(x_{i-1}) = 0$ and $\phi_i(x_i) = 1$ gives

$$\int_{x_{i-1}}^{x_i} \phi'_i(x) \, dx = \phi_i(x_i) - \phi_i(x_{i-1}) = 1 = \int_{x_{i-1}}^{x_i} \frac{C}{K(x)} \, dx$$

From which it follows that for $x \in B_i = [x_{i-1}, x_i]$

$$\phi'_i(x) = \frac{1/K(x)}{\int_{x_{i-1}}^{x_i} 1/K(x) \, dx} \quad \Rightarrow \quad \phi_i(x) = \frac{\int_{x_{i-1}}^{x} 1/K(x) \, dx}{\int_{x_{i-1}}^{x_i} 1/K(x) \, dx}$$
The MsFE method: basis functions
The MsFE method: basis functions
The MsFE method: basis functions
The MsFE method: basis functions
The MsFE method

Find the unique function $p_0$ in

$$V^{ms} = \text{span}\{\phi_i\} = \{u \in H^1_0(\Omega) : a(u, \varphi) = 0 \text{ for all } \varphi \in H^1_0(\bigcup_i B_i)\}$$

satisfying

$$a(p_0, \varphi) = (f, \varphi) \text{ for all } \varphi \in V^{ms}$$

Theorem

Assume that $p$ solves the variational formulation. Then $p = p_0 + \sum_{i=1}^{n} p_i$, where $p_i \in H^1_0(B_i)$ is defined by

$$a(p_i, \varphi) = (f, \varphi) \text{ for all } \varphi \in H^1_0(B_i)$$
Theorem

Assume that $p$ solves the variational formulation. Then $p = p_0 + \sum_{i=1}^{n} p_i$, where $p_i \in H_0^1(B_i)$ is defined by

$$a(p_i, \varphi) = (f, \varphi) \quad \text{for all } \varphi \in H_0^1(B_i)$$
Theorem

Assume that $p$ solves the variational formulation. Then $p = p_0 + \sum_{i=1}^{n} p_i$, where $p_i \in H_0^1(B_i)$ is defined by

$$a(p_i, \varphi) = (f, \varphi) \quad \text{for all } \varphi \in H_0^1(B_i)$$

Assume that $p$ solves the variational formulation and that $\varphi \in V^{ms}$. Then

$$a(p - p_0, \varphi) = a(p, \varphi) - a(p_0, \varphi) = (f, \varphi) - (f, \varphi) = 0$$

Hence, $p_0$ is the orthogonal projection of $p$ onto $V^{ms}$.
The MsFE method: patch refinement property

**Theorem**

Assume that \( p \) solves the variational formulation. Then \( p = p_0 + \sum_{i=1}^{n} p_i \), where \( p_i \in H^1_0(B_i) \) is defined by

\[
a(p_i, \varphi) = (f, \varphi) \quad \text{for all} \; \varphi \in H^1_0(B_i)
\]

Assume that \( p \) solves the variational formulation and that \( \varphi \in V^{ms} \). Then

\[
a(p - p_0, \varphi) = a(p, \varphi) - a(p_0, \varphi) = (f, \varphi) - (f, \varphi) = 0
\]

Hence, \( p_0 \) is the orthogonal projection of \( p \) onto \( V^{ms} \).

Since \( H^1_0(\Omega) = V^{ms} \otimes H^1_0(\cup_i B_i) \) it follows that

\[
p_0(x_i) = p(x_i) \quad \text{for all} \; i
\]

In other words, \( p_0 \) is the interpolant of \( p \) in \( V^{ms} \).
The MsFE method: patch refinement property

**Theorem**

Assume that $p$ solves the variational formulation. Then $p = p_0 + \sum_{i=1}^{n} p_i$, where $p_i \in H_0^1(B_i)$ is defined by

\[ a(p_i, \varphi) = (f, \varphi) \quad \text{for all } \varphi \in H_0^1(B_i) \]

Let $p_I$ be the interpolant of $p$ in $V^{ms}$. Then $p - p_I \in H_0^1(\bigcup_i B_i)$ and it follows from the mutual orthogonality of $V^{ms}$ and $H_0^1(\bigcup_i B_i)$ with respect to $a(\cdot, \cdot)$ that

\[ a(p - p_I, \varphi) = 0 \quad \text{for all } \varphi \in V^{ms} \]
Theorem

Assume that $p$ solves the variational formulation. Then $p = p_0 + \sum_{i=1}^{n} p_i$, where $p_i \in H^1_0(B_i)$ is defined by

$$a(p_i, \varphi) = (f, \varphi) \quad \text{for all } \varphi \in H^1_0(B_i)$$

Let $p_I$ be the interpolant of $p$ in $V^{ms}$. Then $p - p_I \in H^1_0(\cup_i B_i)$ and it follows from the mutual orthogonality of $V^{ms}$ and $H^1_0(\cup_i B_i)$ with respect to $a(\cdot, \cdot)$ that

$$a(p - p_I, \varphi) = 0 \quad \text{for all } \varphi \in V^{ms}$$

Hence, for all $\varphi \in V^{ms}$

$$a(p_I, \varphi) = a(p, \varphi) = (f, \varphi) = a(p_0, \varphi) \quad \implies \quad a(p_I - p_0, \varphi) = 0$$
The MsFE method: patch refinement property

**Theorem**

Assume that $p$ solves the variational formulation. Then $p = p_0 + \sum_{i=1}^{n} p_i$, where $p_i \in H^1_0(B_i)$ is defined by

$$a(p_i, \phi) = (f, \phi) \quad \text{for all } \phi \in H^1_0(B_i)$$

Let $p_I$ be the interpolant of $p$ in $V^{ms}$. Then $p - p_I \in H^1_0(\bigcup_i B_i)$ and it follows from the mutual orthogonality of $V^{ms}$ and $H^1_0(\bigcup_i B_i)$ with respect to $a(\cdot, \cdot)$ that

$$a(p - p_I, \phi) = 0 \quad \text{for all } \phi \in V^{ms}$$

Hence, for all $\phi \in V^{ms}$

$$a(p_I, \phi) = a(p, \phi) = (f, \phi) = a(p_0, \phi) \quad \implies a(p_I - p_0, \phi) = 0$$

Thus, in particular, by choosing $\phi = p_I - p_0$ we obtain

$$a(p_I - p_0, p_I - p_0) = 0,$$

which implies that $p_0 = p_I$
The MsFE method: patch refinement property

**Theorem**

Assume that $p$ solves the variational formulation. Then $p = p_0 + \sum_{i=1}^{n} p_i$, where $p_i \in H^1_0(B_i)$ is defined by

$$a(p_i, \varphi) = (f, \varphi) \quad \text{for all } \varphi \in H^1_0(B_i)$$

In other words: the solution of the variational problem is decomposed into the MsFE solution and solutions of independent local subgrid problems.

This result does not extend to higher dimensions, but the basic construction applies and helps us understand how subgrid features of the solution can be embodied into a coarse grid approximation space.
The MsFE method in 2D

\[ p \in V^{ms} \text{ implies that } \nabla \cdot K \nabla \phi_{ij}^{m,n} = 0 \]

in all coarse blocks \( B_m \)
The MsFE method in 2D

\[ p \in V^{ms} \text{ implies that } \nabla \cdot \mathbf{K} \nabla \phi_{ij} = 0 \]
in all coarse blocks \( B_m \)

\[ \phi_{ij} = 0 \text{ on block interface not emanating from } x_{i,j} \]
The MsFE method in 2D

\[ p \in V^{ms} \text{ implies that } \nabla \cdot \mathbf{K} \nabla \phi^{ij} = 0 \text{ in all coarse blocks } B_m \]

\[ \phi^{ij} = 0 \text{ on block interface not emanating from } x_{i,j} \]

\[ \phi^{ij}(x_{m,n}) = \delta_{i,m} \delta_{j,n} \]
The MsFE method in 2D

\[ p \in V^{ms} \text{ implies that } \nabla \cdot \mathbf{K} \nabla \phi_{ij} = 0 \]

in all coarse blocks \( B_m \)

\[ \phi_{ij} = 0 \text{ on block interface not emanating from } x_{i,j} \]

\[ \phi_{ij}(x_{m,n}) = \delta_{i,m} \delta_{j,n} \]

Boundary conditions on edges emanating from \( x_{i,j} \)?

Unfortunately, the MsFE method is not locally mass-conservative in higher dimensions
1. Introduction

2. Multiscale finite-element methods

3. Multiscale mixed finite-element methods

4. Multiscale finite-volume methods

5. Examples with state-of-the-art method
The multiscale mixed finite-element method

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

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

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

**Standard MFE method**

- Seek solution in \(V_h \times W_h \subset H_0^{1, \text{div}} \times L^2\)
- Approximation spaces: piecewise polynomials (e.g., RT0)

\[
H_0^{1, \text{div}} = \{ \vec{v} \in L^2(\Omega)^d : \nabla \cdot \vec{v} \in L^2(\Omega) \text{ and } \vec{v} \cdot \vec{n} = 0 \text{ on } \partial \Omega \}.
\]
The multiscale mixed finite-element method

Find \((u, p) \in H_{0, \text{div}}^1 \times L^2\) such that

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

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

Multiscale MFE method

- Seek solution in \(V_{H,h} \times W_{H,h} \subset H_{0, \text{div}}^1 \times L^2\)
- Approximation spaces: local numerical solutions

\[
H_{0, \text{div}}^1 = \{ \mathbf{v} \in L^2(\Omega)^d : \nabla \cdot \mathbf{v} \in L^2(\Omega) \text{ and } \mathbf{v} \cdot \mathbf{n} = 0 \text{ on } \partial \Omega \}\]
Hierarchical grids and basis functions

Fine grid with petrophysical parameters cell

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

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

Basis functions $\phi_i(\mathbf{x},y)$ and $\mathbf{\psi}_{ij}(\mathbf{x},y)$ are computed numerically by solving a local flow problem, using an artificial source term to drive a unit flow over the interface between two pairs of blocks.
Hierarchical grids and basis functions

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 at least one basis function $\phi_i \in V$
Hierarchical grids and basis functions

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

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

Basis functions $\phi_i(x, y)$ and $\psi_{ij}(x, y)$ are computed numerically by solving a local flow problem, using an artificial source term to drive a unit flow over the interface between two pairs of blocks.
The coarse-scale system can be derived algebraically from a fine-scale discretization. Here, we will use a mixed formulation.

**Fine-scale system:**

\[
\begin{bmatrix}
B & C \\
C^T & 0
\end{bmatrix}
\begin{bmatrix}
u \\
-p
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
q
\end{bmatrix},
\]

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

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

Alternatively – mixed hybrid form:

\[
\begin{bmatrix}
B & C & D \\
C^T & 0 & 0 \\
D^T & 0 & 0
\end{bmatrix}
\begin{bmatrix}
u \\
-p \\
\pi
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
q \\
0
\end{bmatrix}
\]

\[
d_{ik} = \int_{\partial \Omega} |\psi_i \cdot n_i| \, dx
\]

**Multipoint method:**

- Darcy: \( u_i = T_i (e_i p_i - \pi_i) \)
- Mass conservation for all cells
- Continuity of fluxes across faces
Coarse-scale mixed system

Make the following assumption

\[ u = \Psi u_c + \tilde{u} \]
\[ p = \mathcal{I} p_c + \tilde{p} \]

\( \Psi \) – matrix with basis functions
\( \mathcal{I} \) – prolongation from blocks to cells
Coarse-scale mixed system

Make the following assumption

\[ u = \Psi u_c + \tilde{u} \]
\[ p = \mathcal{I} p_c + \tilde{p} \]

Reduction to coarse-scale system:

\[
\begin{bmatrix}
\Psi^T & 0 \\
0 & \mathcal{I}^T
\end{bmatrix}
\begin{bmatrix}
B & C \\
C^T & 0
\end{bmatrix}
\begin{bmatrix}
\Psi u_c + \tilde{u} \\
-\mathcal{I} p_c - \tilde{p}
\end{bmatrix}
= \begin{bmatrix}
0 \\
\mathcal{I}^T q
\end{bmatrix}
\]

- \( \Psi \) – matrix with basis functions
- \( \mathcal{I} \) – prolongation from blocks to cells
Coarse-scale mixed system

Make the following assumption

\[ u = \Psi u_c + \tilde{u} \]
\[ p = \mathcal{I} p_c + \tilde{p} \]

Reduction to coarse-scale system:

\[
\begin{bmatrix}
\Psi^T & 0 \\
0 & \mathcal{I}^T
\end{bmatrix}
\begin{bmatrix}
B & C \\
C^T & 0
\end{bmatrix}
\begin{bmatrix}
\Psi u_c + \tilde{u} \\
-\mathcal{I} p_c - \tilde{p}
\end{bmatrix} =
\begin{bmatrix}
0 \\
\mathcal{I}^T q
\end{bmatrix}
\]

\[
\begin{bmatrix}
\Psi^T B \Psi & \Psi^T C \mathcal{I} \\
\mathcal{I}^T C^T \Psi & 0
\end{bmatrix}
\begin{bmatrix}
u_c \\
-p_c
\end{bmatrix} =
\begin{bmatrix}
-\Psi^T B \tilde{u} + \Psi^T C \tilde{p} \\
q_c - \mathcal{I}^T C^T \tilde{u}
\end{bmatrix}
\]

\[ \Psi \] – matrix with basis functions
\[ \mathcal{I} \] – prolongation from blocks to cells
Coarse-scale mixed system

Make the following assumption

\[ u = \Psi u_c + \tilde{u} \]
\[ p = \mathcal{I} p_c + \tilde{p} \]

Reduction to coarse-scale system:

\[
\begin{bmatrix}
  \Psi^T & 0 \\
  0 & \mathcal{I}^T
\end{bmatrix}
\begin{bmatrix}
  B & C \\
  C^T & 0
\end{bmatrix}
\begin{bmatrix}
  \Psi u_c + \tilde{u} \\
  -\mathcal{I} p_c - \tilde{p}
\end{bmatrix} =
\begin{bmatrix}
  0 \\
  \mathcal{I}^T q
\end{bmatrix}
\]

Multiscale basis function:

\[
\begin{bmatrix}
  B \\
  C^T
\end{bmatrix}
\begin{bmatrix}
  \Psi \\
  -\Phi
\end{bmatrix} =
\begin{bmatrix}
  0 \\
  w
\end{bmatrix}
\]

Set of equations located to coarse blocks. Flow driven by weight \( w \)

Additional assumptions:

Since \( p \) is immaterial, assume \( w^T \tilde{p} = 0 \). Hence, \( p_i = \int_{\Omega} w p dx \).

Assume that \( \Psi \) spans velocity space, i.e., \( \tilde{u} \equiv 0 \).
Coarse-scale mixed system

Make the following assumption

\[ u = \Psi u_c + \tilde{u} \]
\[ p = \mathcal{I}p_c + \tilde{p} \]

Multiscale basis function:

\[
\begin{bmatrix}
B \\
C^T \\
0
\end{bmatrix}
\begin{bmatrix}
-\Phi
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
w
\end{bmatrix}
\]

Set of equations located to coarse blocks. Flow driven by weight \( w \)

Reduction to coarse-scale system:

\[
\begin{bmatrix}
\Psi^T & 0 \\
0 & \mathcal{I}^T
\end{bmatrix}
\begin{bmatrix}
B & C \\
C^T & 0
\end{bmatrix}
\begin{bmatrix}
\Psi u_c + \tilde{u} \\
-\mathcal{I}p_c - \tilde{p}
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
\mathcal{I}^T q
\end{bmatrix}
\]

Additional assumptions:

Since \( p \) is immaterial, assume \( w^T \tilde{p} = 0 \).
Hence, \( p_c = \int_{\Omega} w \rho \, dx \)
Coarse-scale mixed system

Make the following assumption

\[ u = \Psi u_c + \tilde{u} \]
\[ p = \mathcal{I} p_c + \tilde{p} \]

Reduction to coarse-scale system:

\[
\begin{bmatrix}
\Psi^T & 0 \\
0 & \mathcal{I}^T
\end{bmatrix}
\begin{bmatrix}
B & C \\
C^T & 0
\end{bmatrix}
\begin{bmatrix}
\Psi u_c + \tilde{u} \\
-\mathcal{I} p_c - \tilde{p}
\end{bmatrix} = 
\begin{bmatrix}
0 \\
\mathcal{I}^T q
\end{bmatrix}
\]

Multiscale basis function:

\[
\begin{bmatrix}
B \\
C^T \\
0
\end{bmatrix}
\begin{bmatrix}
\Psi \\
-\Phi
\end{bmatrix} = 
\begin{bmatrix}
0 \\
w
\end{bmatrix}
\]

Set of equations located to coarse blocks. Flow driven by weight \( w \)

Additional assumptions:

Since \( p \) is immaterial, assume \( w^T \tilde{p} = 0 \).

Hence, \( p_c^i = \int_{\Omega_i} wp \, dx \)

Assume that \( \Psi \) spans velocity space, i.e., \( \tilde{u} \equiv 0 \).
Subresolution in pressure

Why not also use the basis functions for pressure?

Pressure is immaterial, but still we need to scale the pressure basis functions. From the definition of the basis functions we have that

\[ B\Psi - C\Phi = 0 \implies B\Psi u_c - C\Phi u_c = 0 \]

which implies that \( \Phi \) and \( \Psi \) should scale similarly.
Why not also use the basis functions for pressure?

Pressure is immaterial, but still we need to scale the pressure basis functions. From the definition of the basis functions we have that

$$B\Psi - C\Phi = 0 \implies B\Psi u_c - C\Phi u_c = 0$$

which implies that $\Phi$ and $\Psi$ should scale similarly.

Hence, the starting-point for the algebraic reduction should be

$$\begin{bmatrix} B & C \\ C^T & 0 \end{bmatrix} \begin{bmatrix} \Psi u_c \\ -\mathcal{I}p_c - D_\lambda \Phi u_c \end{bmatrix} = \begin{bmatrix} 0 \\ q \end{bmatrix}$$

where $D_\lambda = \text{diag}(\lambda_i^0/\lambda_i)$ accounts for saturation variations.
Simple flow problem:

Flux given on left boundary, \( p = 0 \) on right, no-flow elsewhere
Fine grid: \( 10 \times 10 \times 4 \). Coarse grid: \( 5 \times 5 \times 2 \)
Example: linear systems

Hybrid fine–scale system

\[
\begin{bmatrix}
B & C & D \\
C^T & 0 & 0 \\
D^T & 0 & 0
\end{bmatrix}
\]

Hybrid coarse–scale system

\[
\begin{bmatrix}
\Psi^T B \Psi & \Psi^T C \mathcal{I} & \Psi^T D \mathcal{J} \\
\mathcal{I}^T C^T \Psi & 0 & 0 \\
\mathcal{J}^T D^T \Psi & 0 & 0
\end{bmatrix}
\]
Schur complement (block-wise Gauss elimination):

\[
(D^T B^{-1} D - F^T L^{-1} F) \pi = F^T L^{-1} g, \\
F = C^T B^{-1} D, \quad L = C^T B^{-1} C.
\]
**Basis functions**

In $\Omega_i$:

\[ \vec{\psi}_{ij} = -K \nabla p \]
\[ \nabla \cdot \vec{\psi}_{ij} = \omega_i \]

In $\Omega_j$:

\[ \vec{\psi}_{ij} = -K \nabla p \]
\[ \nabla \cdot \vec{\psi}_{ij} = -\omega_j \]

**Source $\omega_i$:**

\[ \omega_i(x) = \frac{K(x)}{\int_{\Omega_i} K(x) \, dx} \]

$x$-component of $\vec{\psi}_{ij}$
Basis functions

One-block approach:

\[
\begin{align*}
\tilde{\psi}_{ij} &= -\lambda K \nabla p \\
\nabla \cdot \tilde{\psi}_{ij} &= \omega_i \\
\Omega_i \\
\tilde{\psi}_{ij} &= -\lambda K \nabla p \\
\nabla \cdot \tilde{\psi}_{ij} &= -\omega_j \\
\Omega_j
\end{align*}
\]

Boundary condition

\[
\tilde{\psi}_{ij} \cdot \vec{n}_i = \nu_{ij} \quad \text{on} \quad \Gamma_{ij}, \quad \tilde{\psi}_{ij} \cdot \vec{n}_i = 0 \quad \text{on} \quad \partial B_i \setminus \Gamma_{ij}.
\]

\(\nu_{ij}\) determined by petrophysical properties (local) or flow solution (global)

Two-block approach:

\[
\begin{align*}
\tilde{\psi}_{ij} &= -\lambda K \nabla p \\
\nabla \cdot \tilde{\psi}_{ij} &= \omega_i \\
\Omega_i \\
\tilde{\psi}_{ij} &= -\lambda K \nabla p \\
\nabla \cdot \tilde{\psi}_{ij} &= -\omega_j \\
\Omega_j
\end{align*}
\]

No boundary condition on inner boundary. Not consistent, but accurate in practice. Can also use overlap if desired
Comparison with upscaling methods

Model equations:

\[
\nabla \cdot \vec{u} = q, \quad \vec{u} = -K \nabla p \\
S_t + \nabla \cdot (S \vec{u}) = \max(q, 0) + S \min(q, 0)
\]

Simulation setup: classical five-spot pattern on layers of SPE10

Layer 1, 400 days

Layer 85, 400 days
Comparison with upscaling methods

Model equations:

\[ \nabla \cdot \vec{u} = q, \quad \vec{u} = -K \nabla p \]

\[ S_t + \nabla \cdot (S \vec{u}) = \max(q, 0) + S \min(q, 0) \]

Simulation setup: classical five-spot pattern on layers of SPE10

Layer 1, 1200 days

Layer 85, 1200 days
Example: layers of SPE10

**Cartesian coarse grids:**
Multiscale methods give enhanced accuracy only when subgrid information is exploited

![Cartesian coarse grids diagram](image)

**Saturation error:**
\[ e(S) = \frac{\| S - S^{ref} \|_2}{\| S^{ref} \|_2} \]
**Example: layers of SPE10**

**Cartesian coarse grids:**
Multiscale methods give enhanced accuracy only when subgrid information is exploited

Saturation error: \( e(S) = \frac{\|S - S^{\text{ref}}\|_2}{\|S^{\text{ref}}\|_2} \)
Example: a dense system of fracture corridors
Computational complexity

Assume a uniform grid on a subset of $\mathbb{R}^d$:

- Grid model with $N = n_f \times N_c$ cells:
  - $N_c$ number of coarse blocks
  - $n_f$ number of fine cells in each coarse cell
- Linear solver of complexity $O(m^\alpha)$ for $m \times m$ system
- Negligible work for determining local b.c., numerical quadrature, and assembly (can be important for some methods)

**Direct solution**

$N^\alpha$ operations for a two-point finite volume method

**MsMFE**

Computing basis functions: $d \cdot N_c \cdot (2n_f)^\alpha$ operations
Solving coarse-scale system: $(d \cdot N_c)^\alpha$ operations
Example: $128 \times 128 \times 128$ fine grid

Comparison with algebraic multigrid, $\alpha = 1.2$
Direct solution may be more efficient, so why bother with multiscale?

In a typical simulation of multiphase flow:

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

Also:

- Possible to solve very large problems
- Easy parallelization
Example: 10th SPE Comparative Solution Project

SPE 10, Model 2:

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

Streamline solver 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
Workflow with automated upgridding in 3D

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

2) Detect all adjacent blocks

3) Compute basis functions

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

for all pairs of blocks

4) Block in coarse grid: component for building global solution
Multiscale method inherits properties of fine-scale solver

Single-phase flow, homogeneous $K$, linear pressure drop
The method so far:

- resolves viscous forces on fine-scale using elliptic basis functions
- resolves other physical forces like gravity, capillary pressure, compressibility, etc on the coarse scale
More physics

The method so far:

- resolves viscous forces on fine-scale using elliptic basis functions
- resolves other physical forces like gravity, capillary pressure, compressibility, etc on the coarse scale

**Why is this so?**

Think of the MsMFE method as a means for computing a homogeneous solution of an equation of the form

$$-\nabla \cdot (\lambda \mathbf{K} \nabla p) = q - h(x, p)$$

In a multiphase setting:

$$-\nabla \cdot (\lambda \mathbf{K} \nabla p) = q - \nabla \cdot (g \mathbf{K} \sum_{\alpha} \rho_{\alpha} \lambda_{\alpha} \nabla z)$$

Since $\lambda$ and $\lambda_{\alpha}$ depend upon $S$, the balance of viscous and gravity forces will depend upon $S \rightarrow$ basis functions would depend strongly upon $S$
To get a convergent method, we need to also account for variations that are not captured by the basis functions → solve a residual equation

\[
\begin{bmatrix}
B & C \\
C^T & 0
\end{bmatrix}
\begin{bmatrix}
\Psi u_c + \tilde{u} \\
-\mathcal{I} p_c - D\lambda \Phi u_c - \tilde{p}
\end{bmatrix} = \begin{bmatrix}
0 \\
q
\end{bmatrix}
\]
Residual correction

To get a convergent method, we need to also account for variations that are not captured by the basis functions. Solve a residual equation

$$\begin{bmatrix}
B & C \\
C^T & 0
\end{bmatrix}
\begin{bmatrix}
\Psi u_c + \tilde{u} \\
-\mathcal{L}p_c - D\lambda \Phi u_c - \tilde{p}
\end{bmatrix}
= \begin{bmatrix}
0 \\
q
\end{bmatrix}$$

$$\begin{bmatrix}
B & C \\
C^T & 0
\end{bmatrix}
\begin{bmatrix}
\tilde{u} \\
-\tilde{p}
\end{bmatrix}
= \begin{bmatrix}
(CD\lambda \Phi - B\Psi)u_c + C\mathcal{L}p_c \\
q - C^T\Psi u_c
\end{bmatrix}$$

To solve this equation, we will typically use a (non)overlapping domain-decomposition method.
Compressible flow

*Parabolic* pressure equation

\[
\vec{v} = -\lambda \mathbf{K} (\nabla p - \sum_j \rho_j f_j \vec{g})
\]

\[
\nabla \cdot \vec{v} = q - c_t \frac{\partial p}{\partial t} + \left( \sum_j c_j f_j \vec{v} + \alpha(p) \mathbf{K} \vec{g} \right) \cdot \nabla p
\]

Iterative mixed formulation:

\[
\begin{bmatrix}
B(s^n) & C \\
C^\top & P(s^n, p_{\nu+1}^{n+1})
\end{bmatrix}
\begin{bmatrix}
\nu_{\nu+1}^{n+1} \\
-p_{\nu+1}^{n+1}
\end{bmatrix}
= \begin{bmatrix}
f(s^n, p_{\nu}^{n+1}) \\
g(s^n, p^n, p_{\nu+1}^{n+1})
\end{bmatrix}
\]

\(n\) denotes time step and \(\nu\) denotes iteration step
Iterative MsMFE for compressible flow

Compute elliptic basis functions, constructed with $w(x) \propto \phi(x)$

For $t=0: \Delta t:T$

1. Solve coarse-scale system iteratively until convergence

   $\begin{bmatrix}
   \Psi^T B \Psi & \Psi^T C \mathcal{I} \\
   \mathcal{I}^T (C^T \Psi - P_\nu D_\lambda \Phi) & \mathcal{I}^T P_\nu \mathcal{I}
   \end{bmatrix}
   \begin{bmatrix}
   u_{\nu+1}^c \\
   -p_{\nu+1}^c
   \end{bmatrix}
   =
   \begin{bmatrix}
   \Psi^T f_\nu \\
   \mathcal{I}^T g_\nu
   \end{bmatrix}$

2. Compute residual equation by domain decomposition

   $\begin{bmatrix}
   B & C^T \\
   C^T & P
   \end{bmatrix}
   \begin{bmatrix}
   \hat{u}_{\nu+1}^c \\
   -\hat{p}_{\nu+1}
   \end{bmatrix}
   =
   \begin{bmatrix}
   f_c - \Psi^T B \Psi u_c + \Psi^T C \mathcal{I} p_c \\
   g_c - \mathcal{I}^T(C^T \Psi - P_\nu D_\lambda \Phi) u_c + \mathcal{I}^T P_\nu \mathcal{I} p_c
   \end{bmatrix}$

3. If fine-scale residual is not below tolerance, go to Step 1
Development towards industry deployment

Grid

Physics

black-oil
industry

black-oil
academic

2-phase
incompr.
industry

2-phase
incompr.

1-phase

With streamlines: 2 min 20 sec!

2009: MsMFE for Stokes–Brinkman

Adjoint MsMFE

Prototypes in FrontSim & MoReS

2003: MsMFE

2004–2005

2006

2008

2008–2010

2012

2013

MsMFE, 2012

iMsMFE, 2012

Corner point w/o fault

Corner point

Unstructured blocks

Adaptive to geology

Unstructured

With streamlines: 2 min 20 sec!

With streamlines: 2 min 20 sec!

2 min 20 sec!

2 min 20 sec!

2 min 20 sec!

2 min 20 sec!

2 min 20 sec!

2 min 20 sec!

2 min 20 sec!

2 min 20 sec!

2 min 20 sec!

2 min 20 sec!

2 min 20 sec!

2 min 20 sec!

2 min 20 sec!

2 min 20 sec!

2 min 20 sec!

2 min 20 sec!
Outline

1. Introduction
2. Multiscale finite-element methods
3. Multiscale mixed finite-element methods
4. Multiscale finite-volume methods
5. Examples with state-of-the-art method
Extensive research over the past 15 years – more than 60 papers by Jenny, Lee, Tchelepi, Lunati, Hajibeygi, and others:

- correction functions to handle non-elliptic features
- extension to compressible flow
- adaptivity in updating of basis functions
- iterative formulation with smoothers (Jacobi, GMRES, ...)
- algebraic formulation
- fracture models (embedded/hierarchical, etc)

Strong focus on the ability to converge to a fine-scale solution has gradually made MsFV similar to multigrid methods
Multiscale finite-volume methods: the key concept

\[ -\nabla \cdot K \nabla p = q \]
\[ Ax = q \]

Initial fine-scale system, incorporating all details of geological model

\[ x = P x_c \]
\[ P = \text{basis}(A) \]
\[ A_{ms} = RAP \]
\[ q_c = Rq \]

Multiscale expansion:
generate basis functions, restrict fine-scale system and right-hand side

\[ x_c = A_{ms}^{-1} q_c \]
\[ x \approx P x_c \]

Solve reduced system, prolongate to obtain approximate pressure
Prolongation and restriction operators

R: 20 × 400

P: 400 × 20
Prolongation and restriction operators

R: 20 × 400

P: 400 × 20
Qualitatively correct \(\rightarrow\) small residual

\[ p^\ast = p^\nu + S(q - A p^\nu) \]

\[ p^\nu + 1 = p^\ast + A^{-1} ms (q - A p^\ast) \]

\(S\) is some inexpensive smoother, e.g., ILU(0)

Hajibeygi, Jenny, Tchelepi, Wang, ... (2008–2015)
Qualitatively correct \(\rightarrow\) small residual

\[ p^* = p^n + S(q - A p^n) \]

\[ p^n + 1 = p^* + A^{-1} m s (q - A p^*) \]

\( S \) is some inexpensive smoother, e.g., ILU(0)

Hajibeygi, Jenny, Tchelepi, Wang, ... (2008–2015)
Residual iteration:

\[ p^* = p^{\nu} + S(q - Ap^{\nu}) \]

\[ p^{\nu+1} = p^* + A_{ms}^{-1}(q - Ap^*) \]

S is some inexpensive smoother, e.g., ILU(0)

Iterative multiscale framework

Flow problem: ∇(K∇p) = q

Discretization: Ap = q

Fine-grid solution

Coarse partition: Bj = {Ci}
Prolongation: p = Ppc

APpc = q

Coarse solution pc

Dual grid/interaction region

Numerical basis function

Restriction: R(AP)pc = Acpc

Alternative iterative methods

1) Richardson iteration:

\[ p^{ν+1} = p^ν + \omega^ν A_{ms}^{-1}(q - A p^n u) \]

2) Two-level method:

\[ p^* = p^ν + S(q - A p^ν) \]
\[ p^{ν+1} = p^* + A_{ms}^{-1}(q - A p^*) \]

3) \( A_{ms}^{-1} \): preconditioner for GMRES
Iterative multiscale framework

Flow problem: $\nabla (K \nabla p) = q$

Discretization: $A p = q$

Fine-grid solution

Coarse partition: $B_j = \{C_i\}$

Prolongation: $p = P p_c$

$A p_c = q$

Coarse solution $p_c$

Dual grid/interaction region

Numerical basis function

Restriction: $R (AP) p_c = A_c p_c$

Alternative iterative methods

1) Richardson iteration:
   
   $$p^\nu + 1 = p^\nu + \omega^\nu A_{ms}^{-1} (q - A p^n u)$$

2) Two-level method:
   
   $$p^* = p^\nu + S (q - A p^\nu)$$
   $$p^{\nu+1} = p^* + A_{ms}^{-1} (q - A p^*)$$

3) $A_{ms}^{-1}$: preconditioner for GMRES

These can be modified
The MsFV prolongation operator

\[ \nabla_t \cdot (K \nabla_t p) = 0 \]

\[ \Phi = 1 \]

\[ \Phi = 0 \]

\[ \nabla \cdot (K \nabla p) = 0 \]

The MsFV prolongation operator

\[ \nabla \cdot (K \nabla p) = 0 \]

\[ \Phi = 1 \]

\[ \Phi = 0 \]

\[ \nabla \cdot (K \nabla p) = 0 \]

The MsFV prolongation operator

\[ \nabla \cdot (K \nabla p) = 0 \]

\[ \Phi = 1 \]

\[ \Phi = 0 \]

The MsFV prolongation operator

\[ \nabla_t \cdot (K \nabla_t p) = 0 \]

\[ \nabla \cdot (K \nabla p) = 0 \]

The MsFV prolongation operator
Permute system based on dual-grid ordering

\[ Qp_h = p = \begin{bmatrix} p_i \\ p_f \\ p_e \\ p_n \end{bmatrix}, \quad QA_hQ^T = A = \begin{bmatrix} A_{ii} & A_{if} & 0 & 0 \\ A_{fi} & A_{ff} & A_{fe} & 0 \\ 0 & A_{ef} & A_{ee} & A_{en} \\ 0 & 0 & A_{ne} & A_{nn} \end{bmatrix} \]

Matrix block \( A_{kl} \): influence from cells \( l \) to mass balance of cells \( k \)
The MsFV method: operator formulation

Permute system based on dual-grid ordering

\[ Qp_h = p = \begin{bmatrix} p_i \\ pf \\ pe \\ pn \end{bmatrix} , \quad QA_h Q^T = A = \begin{bmatrix} A_{ii} & A_{if} & 0 & 0 \\ A_{fi} & A_{ff} & A_{fe} & 0 \\ 0 & A_{ef} & A_{ee} & A_{en} \\ 0 & 0 & M_{ee} & A_{en} \end{bmatrix} \]

Matrix block \( A_{kl} \): influence from cells \( l \) to mass balance of cells \( k \)

Remove lower-diagonal blocks and ensure mass balance is still enforced,

\[ (M_{kk})_{rr} = (A_{kk})_{rr} + \sum_s (A_{kl})_{rs} \rightarrow \begin{bmatrix} A_{ii} & A_{if} & 0 & 0 \\ 0 & M_{ff} & A_{fe} & 0 \\ 0 & 0 & M_{ee} & A_{en} \\ 0 & 0 & 0 & M_{nn} \end{bmatrix} \]
Assume nodal pressure $p_n$ to be known. This gives a solution

$$p = P p_n$$

where $B$ are the basis functions

$$P = \begin{bmatrix}
    A_{ii}^{-1} A_{if} M_{ff}^{-1} A_{fe} M_{ee}^{-1} A_{en} \\
    M_{ff}^{-1} A_{fe} M_{ee}^{-1} A_{en} \\
    M_{ee}^{-1} A_{en} \\
    I
\end{bmatrix}$$
The MsFV method: operator formulation

Assume nodal pressure $p_n$ to be known. This gives a solution

$$p = P p_n$$

where $B$ are the basis functions

$$P = \begin{bmatrix} A_{ii}^{-1} A_{if} M_{ff}^{-1} A_{fe} M_{ee}^{-1} A_{en} \\ M_{ff}^{-1} A_{fe} M_{ee}^{-1} A_{en} \\ M_{ee}^{-1} A_{en} \\ I \end{bmatrix}$$

Pressure in nodes $p_n$ found by enforcing mass balance on the coarse grid
The MsFV method: operator formulation

Categorization of cells

System matrix $A$

$$\begin{bmatrix} A_{ii} & A_{ei} & 0 \\ 0 & M_{ee} & A_{ne} \\ 0 & 0 & M_{nn} \end{bmatrix} \rightarrow \tilde{P} = \begin{bmatrix} A_{ii}^{-1}A_{ei}M_{ee}^{-1}A_{ne} \\ M_{ee}^{-1}A_{ne} \\ I \end{bmatrix}$$

$$(M_{ll})_{rr} = (A_{ll})_{rr} + \sum_s (A_{kl})_{rs}$$
The MsFV method: prominent shortcomings

Not working as well as you may get the impression of:

- Only applicable to relatively simple grids: Cartesian, simplexes, 'conceptual' fault models
- Localization procedure not robust → unstable multipoint coarse-scale stencil gives oscillatory solutions
- Test cases reported in literature use seemingly complex flow physics
- Use of iterations over-emphasized!

\[ \log(K) \]

SPE 10: MsFV solution \( p \notin [0, 1] \)
The MsFV method: wirebasket ordering

Requirement of consistent dual-primal partition makes coarsening difficult

- node/edges
- faces
- noncontiguous faces
- extra cells
The MsFV method: wirebasket ordering

Requirement of consistent dual-primal partition makes coarsening difficult

- node/edges
- faces
- noncontiguous faces
- extra cells

Algorithms for generating partitions on general grids:
- automated on rectilinear, curvilinear, triangular, and Voronoi grids
- semi-automated on (simple) stratigraphic grids non-matching faces
- no known algorithm for full industry-standard complexity
The MsFV method: wirebasket ordering

Requirement of consistent dual-primal partition makes coarsening difficult

Automated algorithms struggle with:
- dual block centers in low-permeable regions
- dual edges crossing strong permeability contrasts (twice)
- large number of cells categorized as edges
  → nonmonotonicity, poor decoupling, failure to reproduce linear flow
MsTPFA: improve monotonicity properties

Idea: make coarse-scale stencil be of two-point type

Approach:
- Move degrees-of-freedom to block faces (as in MsMFE)
- Compute flow solutions as in transmissibility upscaling
- Use additional partition-of-unity to define basis functions
MsTPFA: improve monotonicity properties

- Much more stable than MsFV, although not 100% perfect
- Applicable to stratigraphic and fully unstructured grids
- Can be used both as preconditioner and approximate solver
- Slightly less accurate than MsFV on simple rectangular grids
- Can likely be generalized to other MPFA-type methods
What are our requirements on the prolongation operator?

- Partition of unity to represent constant fields
  \[ \sum_j P_{ij} = 1 \rightarrow \text{Exact interpolation of constant modes} \]

- Algebraically smooth: minimize \( \|Ap\|_1 \) implies that \( A P p_c \approx Ap \) locally

- Localication: coarse system \( A_c = R A P \) becomes denser as the support of basis functions grows
Basis functions require **a coarse grid** and **a support region**

- Region constructed using triangulation of nodal coarse neighbors, resulting in a multipoint stencil on the coarse scale
- Avoid solving reduced flow problem along perimeter
- Main point: simple to implement in 3D for general polyhedral grids
MsRSB: restricted smoothing

Ideally, operators are both smooth and local

1. Start with constant functions on primal grid
2. Apply Jacobi-like iterations as in algebraic multigrid methods, $P^{n+1} = P^n - \omega D^{-1}(A P^n)$
3. Restrict each function to its support region
4. Repeat Steps 2 and 3 until convergence

Initial constant basis

After one pass

After 10 passes

Converged ($n \approx 100$)
MsRSB: restricted smoothing

Ideally, operators are both *smooth* and *local*

1. Start with constant functions on primal grid
2. Apply Jacobi-like iterations as in algebraic multigrid methods, $P^{n+1} = P^n - \omega D^{-1}(A P^n)$
3. Restrict each function to its support region
4. Repeat Steps 2 and 3 until convergence
Coarse grid: $3 \times 3$ partition

Set $P_j$ to one inside block $j$

Jacobi increment: $d_j = -\omega D^{-1} A P_j^n$

Localize update:

$$\hat{d}_{ij} = \begin{cases} 
  d_{ij} - P_{ij}^n \sum_k d_{ik} \\ 
  1 + \sum_k d_{ik} 
\end{cases}$$

Apply increment: $P_{ij}^{n+1} = P_{ij}^n + \hat{d}_{ij}$

Indices: $i=$cell, $j=$●, $k=$●
Outline

1. Introduction
2. Multiscale finite-element methods
3. Multiscale mixed finite-element methods
4. Multiscale finite-volume methods
5. Examples with state-of-the-art method
Example: validation on SPE10 layers

<table>
<thead>
<tr>
<th>Error</th>
<th>Grid</th>
<th>$p$ (L$^2$)</th>
<th>$p$ (L$^\infty$)</th>
<th>$v$ (L$^2$)</th>
<th>$v$ (L$^\infty$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MsFV</td>
<td>6 × 11</td>
<td>0.0313</td>
<td>0.0910</td>
<td>0.1138</td>
<td>0.4151</td>
</tr>
<tr>
<td>MsRSB</td>
<td>6 × 11</td>
<td>0.0204</td>
<td>0.0766</td>
<td>0.0880</td>
<td>0.4071</td>
</tr>
</tbody>
</table>
Example: validation on SPE10 layers

<table>
<thead>
<tr>
<th>Error</th>
<th>Grid</th>
<th>p (L^2)</th>
<th>p (L_\infty)</th>
<th>v (L^2)</th>
<th>v (L_\infty)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MsFV</td>
<td>6 × 11</td>
<td>0.2299</td>
<td>2.0725</td>
<td>0.4913</td>
<td>0.7124</td>
</tr>
<tr>
<td>MsRSB</td>
<td>6 × 11</td>
<td>0.0232</td>
<td>0.0801</td>
<td>0.1658</td>
<td>0.3240</td>
</tr>
</tbody>
</table>
Example: GMRES-MS-ILU(0) for full SPE10
Example: unstructured PEBI grid

- Unstructured grid designed to minimize grid orientation effects
- Two embedded radial grids near wells
- Fine grid adapts to faults
- The faults are sealed, i.e., allow no fluid flow through
Example: unstructured PEBI grid

Two-step preconditioner, ILU(0) as 2nd stage, Richardson iterations
Example: unstructured PEBI grid

Two-step preconditioner, ILU(0) as 2nd stage, Richardson iterations
Early field model of a giant reservoir from the Norwegian North Sea

- 216 000 cells with a large number of faults and eroded layers
- Very challenging anisotropic permeability and grid
- Model includes cells with nearly 40 faces
- Contrived well pattern: four vertical wells force flow through the whole model
Example: trade accuracy for computational efficiency

SPE 10 with two strongly compressible fluids

Iterated fine-scale solver:
- 0.001 pressure increment tolerance
- $10^{-6}$ tolerance for algebraic multigrid

Iterated multiscale solver:
- 0.005 pressure increment tolerance
- $10^{-2}$ tolerance for MsRSB solver

Approximate MsRSB solver is ten times faster than baseline sequential
Example: realistic waterflooding

Watt Field: water flooding
415 711 active cells, three rock types
7 injectors, 15 horizontal producers
Example: realistic waterflooding

Thin solid: fine-scale solution
Thick dashed: multiscale solution

Multiscale: 800 blocks, tolerance 0.05
Solver speedup: 9×
Example: 3-phase flow

- Synthetic model with fluid behavior based on SPE1 benchmark
- Gas is injected at constant rate into an undersaturated reservoir
- Producer at fixed bottom hole pressure
- Highly sensitive to pressure approximation

Gas saturation at breakthrough

![Graph showing gas saturation at breakthrough with time and production rate.](image)
Example: 3-phase flow

Gas production rate [m$^3$/s]

Reference
MsRSB (Tol = 0.1)
MsRSB (Tol = 1e-3)
MsRSB (Tol = 1e-6)

Time [year]

9.3 9.4 9.5 9.6 9.7
Example: water-based EOR

- Full Eclipse 100 polymer model with adsorption, Todd–Longstaff mixing, inaccessible pore volume, and permeability reduction
- Polymer concentration changes water viscosity to achieve better sweep
- Viscosity of water-polymer mixture depends on velocity (shear thinning)
- Non-Newtonian fluid rheology makes the pressure equation highly nonlinear
Example: compositional flow

- Carefully designed, sequentially-implicit method
- Challenging six-component fluid model from Mallison et al. (SPE 79691)
- Peng–Robison equation of state
- Heterogeneity sampled from the SPE 10 model

![Graph showing production vs. time for different components](image-url)
There are still issues that can be improved:

- Slow convergence in certain cases with strong contrasts and long correlation lengths
- Desire to adapt coarse grid to geological features
- Improved resolution of wells
- More efficient reconstruction of conservative fluxes

Previous work:

- generalized multiscale element methods (Efendiev et al)
- hybrid finite-volume/Galerkin method (Cortinovis and Jenny)
Assume $N$ prolongation operators $P^1, \ldots, P^N$ that may come from different coarse grids and support regions, or different multiscale methods (MsRSB, MsFV,\ldots)

Likewise, there are $N$ restriction operators $R^1, \ldots, R^N$
Assume $N$ prolongation operators $P^1, \ldots, P^N$ that may come from different coarse grids and support regions, or different multiscale methods (MsRSB, MsFV, \ldots)

Likewise, there are $N$ restriction operators $R^1, \ldots, R^N$

Multiplicative multistep method:

\[
p^* = p^{k+(\ell-1)/N} + S(q - Ap^{k+(\ell-1)/N}) \\
\]

\[
p_{k+\ell/N} = p^* + P^\ell \underbrace{(R^\ell A P^\ell)}_{A_{ms}^\ell}^{-1} R^\ell (q - Ap^*),
\]

Example setup: $P^1$ is general and covers domain evenly, whereas $P^2, \ldots, P^N$ are feature specific
Minimal assumptions on operators

Three requirements on pairs of prolongation/restriction operators:

1. $P^\ell$ and $R^\ell$ are constructed from a non-overlapping partition of the fine grid. Each column $j$ in $P^\ell$ is called a *basis function* and is associated with a coarse grid block $B^\ell_j$.

2. The support $S^\ell_j$ of each basis function is compact and contains $B^\ell_j$.

3. The columns of $P^\ell$ form a partition of unity, i.e., each row in $P^\ell$ has unit row sum.
Numerical example: SPE10

Layer 85: pressure drop from north to south end, linear relperms, unit viscosity

<table>
<thead>
<tr>
<th>Partition</th>
<th>L^2</th>
<th>L^∞</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rectangular</td>
<td>0.0307</td>
<td>0.1782</td>
</tr>
<tr>
<td>Metis</td>
<td>0.0791</td>
<td>0.5506</td>
</tr>
<tr>
<td>Combined</td>
<td>0.0293</td>
<td>0.2929</td>
</tr>
</tbody>
</table>
Numerical example: SPE10

Layer 85: pressure drop from north to south end, linear relperms, unit viscosity

<table>
<thead>
<tr>
<th>Partition</th>
<th>$L^2$</th>
<th>$L^\infty$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rectangular</td>
<td>0.0307</td>
<td>0.1782</td>
</tr>
<tr>
<td>Metis</td>
<td>0.0791</td>
<td>0.5506</td>
</tr>
<tr>
<td>Combined</td>
<td>0.0293</td>
<td>0.2929</td>
</tr>
</tbody>
</table>

- Local nonmonotonicity
Numerical example: SPE10

Layer 85: pressure drop from north to south end, linear relperms, unit viscosity

Local nonmonotonicity

Residual
Numerical example: SPE10

Layer 85: pressure drop from north to south end, linear relperms, unit viscosity

\[ A_{ms}^{-1} \]

four ms iterations
Numerical example: Gullfaks

Higher resolution: $80 \times 100 \times 52$ cells, 416 000 active
Partition: rectangular (upper) and by Metis (lower)
Numerical example: Gullfaks

![Graph showing residual iterations for different methods: Cart, Metis, Cart+Metis.](Image)
Summary

Presented a number of different multiscale methods:

- 15+ years of research with many detours/focus on unimportant issues
- MsRSB is probably the most simplistic found in the literature . . .
- Large number of tests — very encouraging results!
- Finally, we seem to have a method that is working as required
- Key to efficiency: reduce accuracy, but retain mass conservation
- MsRSB is implemented in the INTERSECT R&P simulator
- MsMFE, MsFV, and MsRSB all available in MRST