MultiScale Method for Flow in Deformable Porous Media: Poro-Elastic

Hadi Hajibeygi[1]

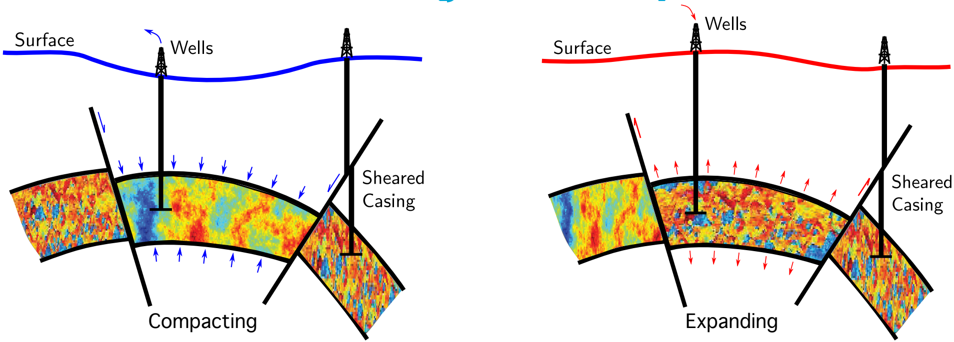
Irina Sokolova[1] Nicola Castelletto[2,3] Hamdi A. Tchelepi[2]

[1]Civil Engineering and Geosciences, Delft University of Technology
[2]Energy Resources Engineering, Stanford University
[3]Lawrence Livermore National Laboratory

*Journal of Computational Physics 331 (2017) 337-356
+ ECMOR 2018, Spain

[2]H.Hajibeygi@tudelft.nl

Poromechanics: effective design and safety assessment

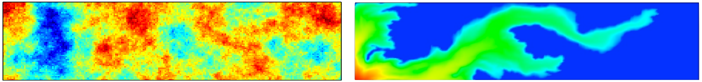


[modified after Mike Bruno, *SPE Drill Eng* (1992)]

- large scales (space-time)
- heterogeneous coefficients (no scale separation), e.g. K in $-\nabla \cdot (\frac{K}{\mu} \cdot \nabla p) = q$
- uncertain (parameters, geometries, B.C., ...)
- nonlinear coupled PDE's (mixed-type)

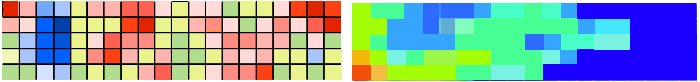
multiscale methods

Reference

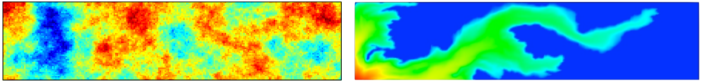


multiscale methods

Upscaling



Reference

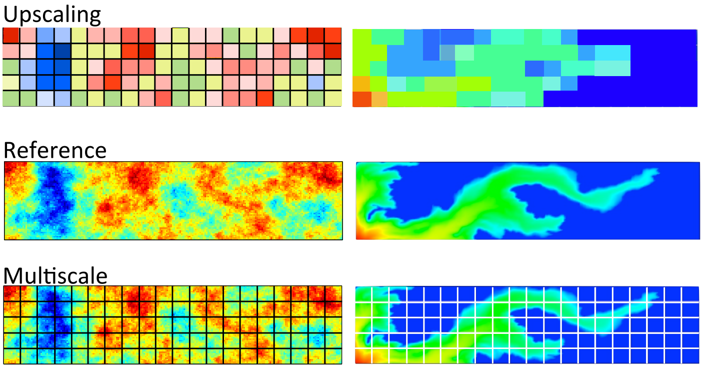


Upscaling Methods

"Effective Coefficients"

- Pseudo Functions
- Homogenization
- ...

multiscale methods



Upscaling Methods

"Effective Coefficients"

- Pseudo Functions
- Homogenization
- ...

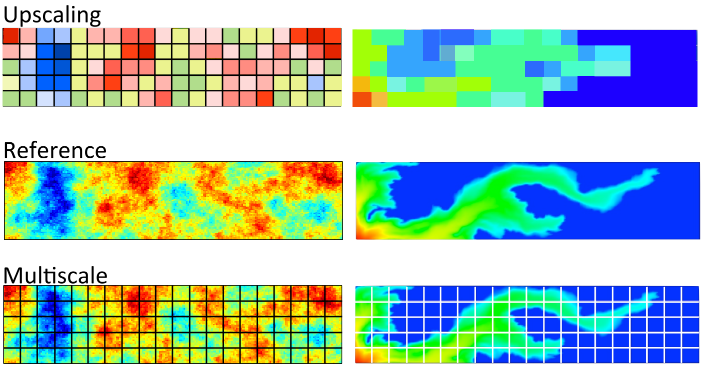
Multiscale Methods

"Map the Unknowns"

- Multiscale Finite-Element
- Multiscale Finite-Volume
- ...

R. Helmig et al., Ch. 15 in *Handbook of Geomathematics* (2010)
W. E, Principles of Multiscale Modeling, Cambridge University (2011)

multiscale methods



Upscaling Methods

"Effective Coefficients"

- Pseudo Functions
- Homogenization
- ...

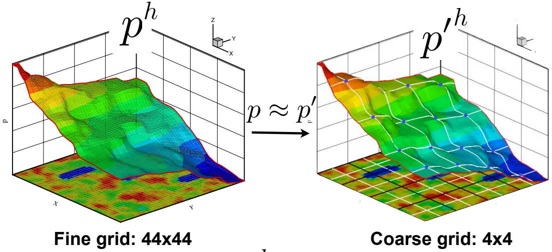
Multiscale Methods

"Map the Unknowns"

- Multiscale Finite-Element
- Multiscale Finite-Volume
- ...

Hou & Wu *JCP* ('97); Efendiev et al. SIAM ('00); Jenny et al. *JCP* ('03); Hajibeygi et al. *JCP* ('08) ...
Jenny, Tchelepi, Lee, Hajibeygi, Zhou, Wang, Tene, Manea, Cusini, Lie, Moyner, ...

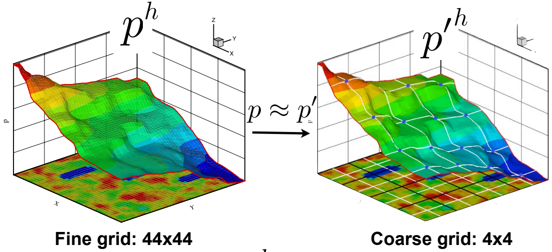
Multiscale & Multigrid:



$$p^h \approx p'^h = P\ p^H$$

- Algebraic: for A x = b, both construct $x^h \approx Px^H$, $(RAP)x^H = Rb$

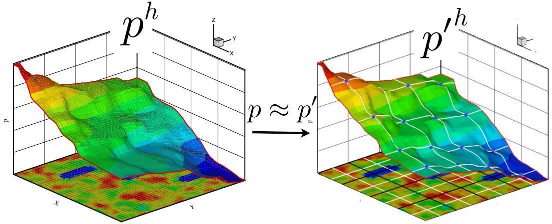
Multiscale & Multigrid:



$$p^h \approx p'^h = P\ p^H$$

- Algebraic: for A x = b, both construct $x^h \approx P x^H$, $(RAP)x^H = Rb$
- Coarse Grid: MS (few aggressive) – MG (many moderate) $\Rightarrow$ special R & P

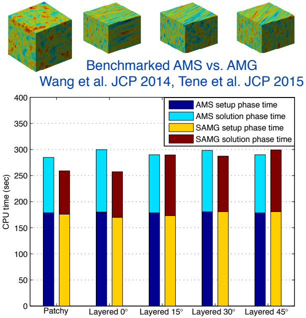
Multiscale & Multigrid:

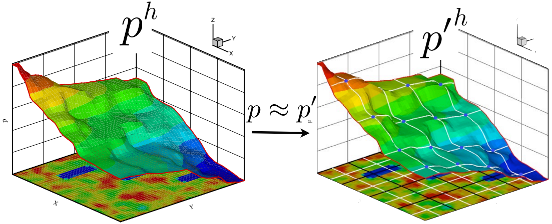


$$p^h \approx p'^h = P\ p^H$$

- Algebraic: for A x = b, both construct $x^h \approx P x^H$, $(RAP)x^H = Rb$
- Coarse Grid: MS (few aggressive) – MG (many moderate) $\Rightarrow$ special R & P
- Purpose: MS (good approximate solution) – MG (scalable solver)

Multiscale & Multigrid:





Fine grid: 44x44 **Coarse grid: 4x4**

$$p^h \approx p'^h = P \, p^H$$

- Algebraic: for A x = b, both construct $x^h \approx P x^H$, $(RAP)x^H = Rb$
- Coarse Grid: MS (few aggressive) – MG (many moderate) $\Rightarrow$ special R & P
- Purpose: MS (good approximate solution) – MG (scalable solver)

Model problem: Biot's poroelasticity equations

Find the displacement vector, $\boldsymbol{u}(\xi, t)$, and pressure, $p(\xi, t)$, such that:

$$\nabla \cdot \underbrace{(\mathbf{C}_{dr} : \nabla^s \boldsymbol{u} - bp\mathbf{1})}_{\text{Stress tensor}} = \boldsymbol{f} \qquad \text{(linear momentum balance)}$$

$$\frac{\partial}{\partial t} \underbrace{\left(b\nabla \cdot \boldsymbol{u} + \frac{1}{M_b} p \right)}_{\text{accumulation}} + \nabla \cdot \underbrace{\left(-\frac{\boldsymbol{\kappa}}{\mu} \nabla p \right)}_{\text{Darcy's velocity}} = g \qquad \text{(mass balance)}$$

H. F. Wang, Theory of Linear Poroelasticity, Princeton University (2000)

Model problem: Biot's poroelasticity equations

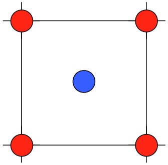
Find the displacement vector, $\boldsymbol{u}(\xi, t)$, and pressure, $p(\xi, t)$, such that:

$$\nabla \cdot \underbrace{\left(\mathbf{C}_{dr} : \nabla^s \boldsymbol{u} - bp\mathbf{1}\right)}_{\text{Stress tensor}} = \boldsymbol{f} \qquad \text{(linear momentum balance)}$$

$$\frac{\partial}{\partial t} \underbrace{\left(b\nabla \cdot \boldsymbol{u} + \frac{1}{M_b} p\right)}_{\text{accumulation}} + \nabla \cdot \underbrace{\left(-\frac{\boldsymbol{\kappa}}{\mu} \nabla p\right)}_{\text{Darcy's velocity}} = g \qquad \text{(mass balance)}$$

Fine-scale discrete fully-implicit FE-FV(displacement-pressure):

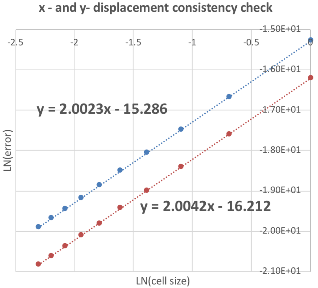
$$\underbrace{\begin{bmatrix} A_{uu} & A_{up} \\ A_{pu} & A_{pp} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \boldsymbol{u}^h_{n+1} \\ \boldsymbol{p}^h_{n+1} \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} \boldsymbol{f}_{n+1} \\ \boldsymbol{g}_{n+1} \end{bmatrix}}_{b}$$



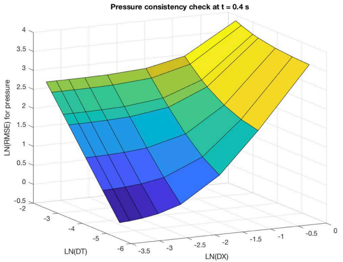
White et al. (2016); Deb & Jenny (2017) & Wheeler et al. & Luo, Gaspar, Oosterlee et al. ...

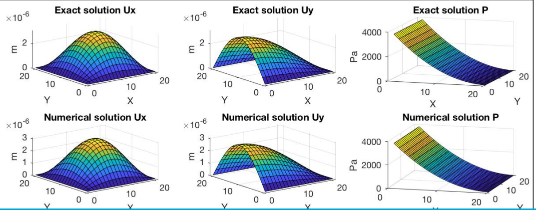
Fine-scale System Consistency Check (FV-FV)

- Mechanics RMS error for displacement:



- Poromechanics RMS error for pressure





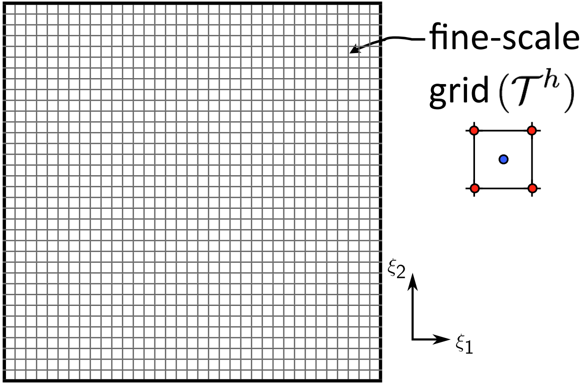
Hybrid MultiScale Finite Element-Finite Volume (1/2)

$$Ax = b, \textit{where} \quad x = \begin{bmatrix} \mathbf{u}^h \\ \mathbf{p}^h \end{bmatrix} \approx \begin{bmatrix} \mathbf{u}'^h \\ \mathbf{p}'^h \end{bmatrix} = \begin{bmatrix} P^{(u,u)} & 0 \\ 0 & P^{(p,p)} \end{bmatrix} \begin{bmatrix} \mathbf{u}^H \\ \mathbf{p}^H \end{bmatrix}$$

MSFE-mech. mainly: Castalletto et al. JCP (16); Buck et al. Eur. J Math (13); Zhang et al. AWR (09)

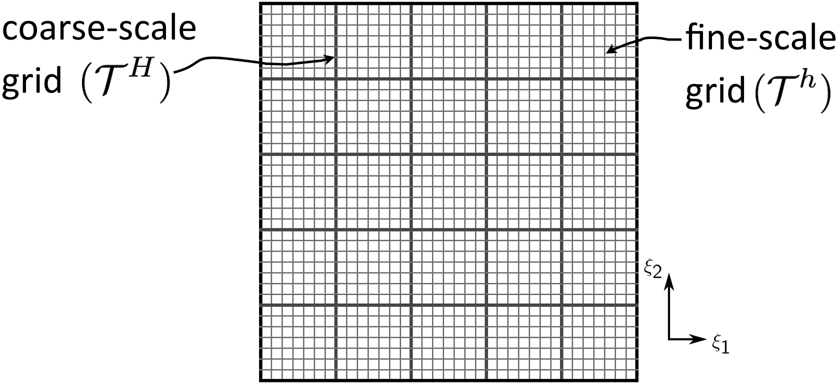
Hybrid MultiScale Finite Element-Finite Volume (1/2)

$$Ax = b, \textit{ where } \quad x = \begin{bmatrix} \mathbf{u}^h \\ \mathbf{p}^h \end{bmatrix} \approx \begin{bmatrix} \mathbf{u}'^h \\ \mathbf{p}'^h \end{bmatrix} = \begin{bmatrix} P^{(u,u)} & 0 \\ 0 & P^{(p,p)} \end{bmatrix} \begin{bmatrix} \mathbf{u}^H \\ \mathbf{p}^H \end{bmatrix}$$



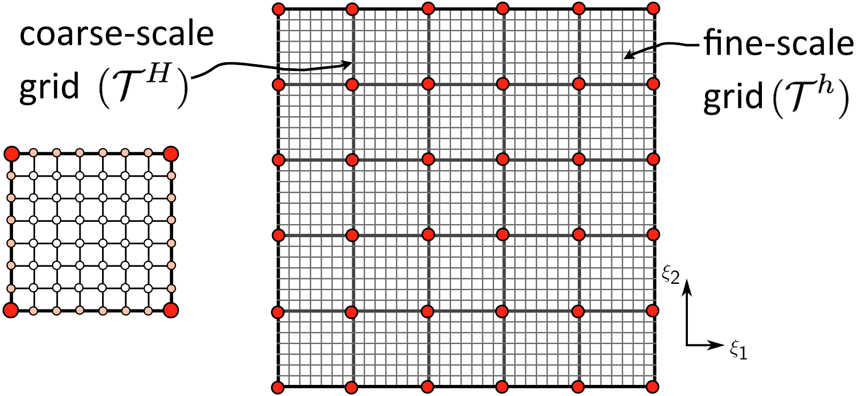
Hybrid MultiScale Finite Element-Finite Volume (1/2)

$$Ax = b, \textit{where} \quad x = \begin{bmatrix} \mathbf{d}^h \\ \mathbf{p}^h \end{bmatrix} \approx \begin{bmatrix} \mathbf{d}'^h \\ \mathbf{p}'^h \end{bmatrix} = \begin{bmatrix} P^{(d,d)} & 0 \\ 0 & P^{(p,p)} \end{bmatrix} \begin{bmatrix} \mathbf{d}^H \\ \mathbf{p}^H \end{bmatrix}$$



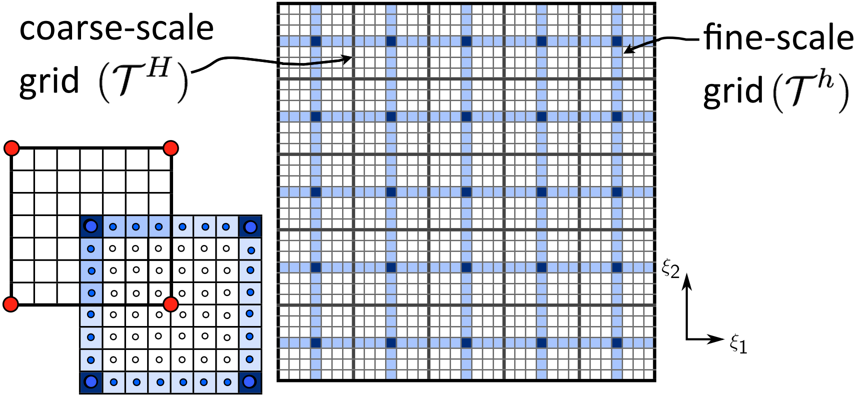
Hybrid MultiScale Finite Element-Finite Volume (1/2)

$$Ax = b, \textit{ where } \quad x = \begin{bmatrix} \mathbf{d}^h \\ \mathbf{p}^h \end{bmatrix} \approx \begin{bmatrix} \mathbf{d}'^h \\ \mathbf{p}'^h \end{bmatrix} = \begin{bmatrix} P^{(d,d)} & 0 \\ 0 & P^{(p,p)} \end{bmatrix} \begin{bmatrix} \mathbf{d}^H \\ \mathbf{p}^H \end{bmatrix}$$

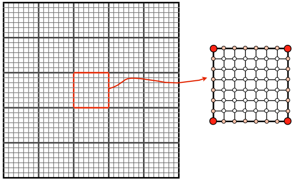
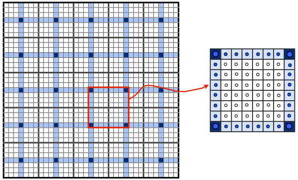


Hybrid MultiScale Finite Element-Finite Volume (1/2)

$$Ax = b, \textit{ where } \quad x = \begin{bmatrix} \mathbf{d}^h \\ \mathbf{p}^h \end{bmatrix} \approx \begin{bmatrix} \mathbf{d}'^h \\ \mathbf{p}'^h \end{bmatrix} = \begin{bmatrix} P^{(d,d)} & 0 \\ 0 & P^{(p,p)} \end{bmatrix} \begin{bmatrix} \mathbf{d}^H \\ \mathbf{p}^H \end{bmatrix}$$



Hybrid MultiScale Finite Element-Finite Volume (2/2)

Fine-scale	Multiscale	
$\boldsymbol{u} = N_{\boldsymbol{u}}^h \boldsymbol{u}^h$	$\boldsymbol{u}' = N_{\boldsymbol{u}}^H \boldsymbol{u}^H$ $N_{\boldsymbol{u}}^H = N_{\boldsymbol{u}}^h P^{(\boldsymbol{u},\boldsymbol{u})}$ $\boldsymbol{u}^h \approx P^{(u,u)} \boldsymbol{u}^H$	
$p = N_p^h \boldsymbol{p}^h$	$p' = N_p^H \boldsymbol{p}^H$ $N_p^H = N_p^h P^{(p,p)}$ $\boldsymbol{p}^h \approx P^{(p,p)} \boldsymbol{p}^H$	

Note: full prolongation is considered if the flow-mechanics coupling is strong (Tene et al. JCP, 2016).

Pressure multiscale basis functions

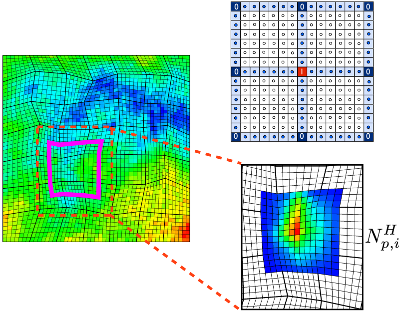
Local flow on dual-coarse elements:

$$\frac{\partial}{\partial t}\left(b\nabla\cdot\boldsymbol{u}+\frac{1}{M_b}p\right)+\nabla\cdot\left(-\frac{\boldsymbol{\kappa}}{\mu}\nabla p\right)=g$$

$$\nabla\cdot\left(-\frac{\boldsymbol{\kappa}}{\mu}\cdot\nabla\boldsymbol{N}^H_{p,i}\right)=0 \qquad \text{in } \hat{T}^H_i$$

$$\nabla_{\parallel}\cdot\left(-\frac{\boldsymbol{\kappa}}{\mu}:\nabla^s_{\parallel}\boldsymbol{N}^H_{p,i}\right)=0 \qquad \text{on } \partial\hat{T}^H_i$$

$$\boldsymbol{N}^H_{p,i}(\boldsymbol{\xi}^V_{p,j})=\delta_{ij}$$

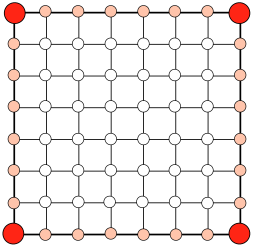


Reminder: $N^H_p = N^h_p P^{(p,p)}$

Displacement multiscale basis functions:

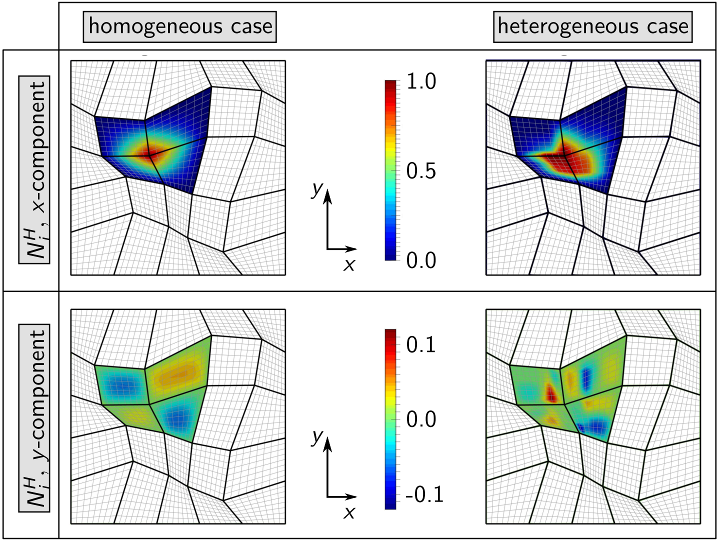
Local deformation (no pore pressure), based on: $\nabla \cdot (\mathbf{C}_{dr} : \nabla^s \boldsymbol{u} - bp\mathbf{1}) = \boldsymbol{f}$

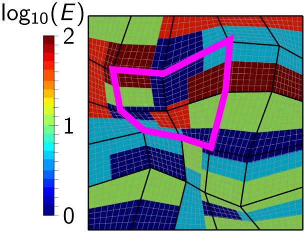
$$
\begin{aligned}
\nabla \cdot \left(\mathbf{C}_{dr} : \nabla^s \boldsymbol{N}^H_{\boldsymbol{u},i}\right) &= \mathbf{0} \qquad \text{in } T^H_k \\
\nabla_{\parallel} \cdot \left(\mathbf{C}_{dr} : \nabla^s_{\parallel} \boldsymbol{N}^H_{\boldsymbol{u},i}\right) &= \mathbf{0} \qquad \text{on } \partial T^H_k \\
\boldsymbol{N}^H_{\boldsymbol{u},i}(\boldsymbol{\xi}^V_{\boldsymbol{u},j}) &= \delta_{ij}\ \boldsymbol{e}
\end{aligned}
$$



Reminder: $N^H_{\boldsymbol{u}} = N^h_{\boldsymbol{u}} P^{(\boldsymbol{u},\boldsymbol{u})}$

Displacement multiscale basis functions





heterogeneous Young's modulus (E) distribution

Prolongation (H → h) and Restriction (h → H)

Prolongation operator

$$P = \begin{bmatrix} P^{(\boldsymbol{u},\boldsymbol{u})} & 0 \\ 0 & P^{(\boldsymbol{p},\boldsymbol{p})} \end{bmatrix}$$

Restriction operator

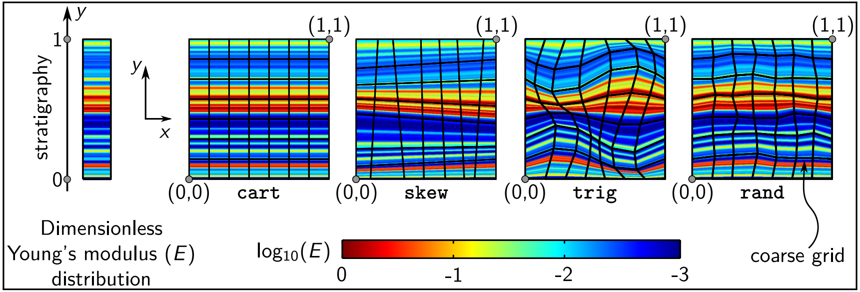
$$R = \begin{bmatrix} R^{(\boldsymbol{u},\boldsymbol{u})} & 0 \\ 0 & R^{(\boldsymbol{p},\boldsymbol{p})} \end{bmatrix}$$

- Flow: FE ($R^{(p,p)} = {P^{(p,p)}}^T$) or FV (integration over coarse volumes)
- Displacement: FE ($R^{(u,u)} = {P^{(u,u)}}^T$) or FV (integration over coarse volumes)
- **Finally**, instead of $Ax^h = b$, solve $(RAP)x^H = R\ b$
  Then, $x'^h = Px^H$ is the approximate solution

Iterative Strategy: (1) Solve CS + (2) Smooth FS

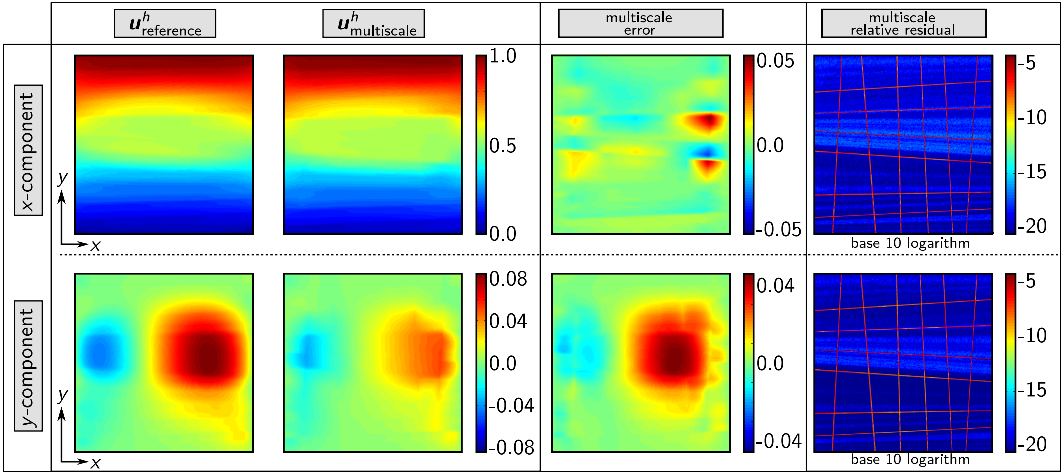
- Employ coarse grids (only once)
- Solve local P & R (only once with our formulation)
  1. Solve coarse-scale correction: $(RAP)\delta x^H = R(b - Ax^\nu)$
  2. Find approximate fine-scale solution: $x^{\nu+1/2} = x^\nu + P\delta x^H$
  3. If $||b - Ax^{\nu+1/2}||_2 < \epsilon$, done!
     Else, (smooth $x^{\nu+1/2}$) $\rightarrow x^\nu$ & repeat from 1

Heterogeneous synthetic example: Setup

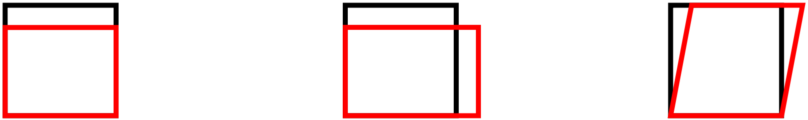


- Fine-scale grid: 224×224 elements $(101, 250)$
- Coarse-scale grid: 7×7 elements (128)

Skewed mesh, simple shear



Heterogeneous synthetic example: Results (no iterations)

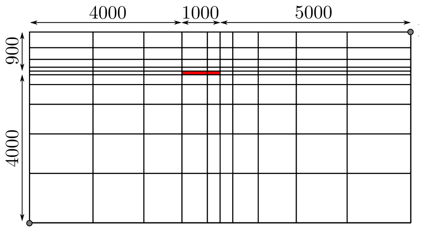


mesh	laterally constrained		laterally unconstrained		simple shear	
	ε	ρ	ε	ρ	ε	ρ
`cart`	4.35×10^{-13}	8.63×10^{-16}	4.24×10^{-02}	1.27×10^{-05}	4.27×10^{-02}	2.97×10^{-04}
`skew`	8.70×10^{-03}	1.84×10^{-05}	5.20×10^{-02}	2.43×10^{-05}	5.23×10^{-02}	2.91×10^{-04}
`trig`	2.79×10^{-02}	6.28×10^{-05}	5.77×10^{-02}	7.98×10^{-05}	5.69×10^{-02}	3.67×10^{-04}
`rand`	1.30×10^{-02}	1.78×10^{-05}	3.75×10^{-02}	2.51×10^{-05}	4.80×10^{-02}	3.13×10^{-04}

- ε: relative error L^{∞}-norm
- ρ: relative residual L^2-norm

Plane-strain subsidence

- depletion Δp = -100 bar
- Fine grid: 320×320 $(206, 082)$
- MSFE grid: 10×10 (242)



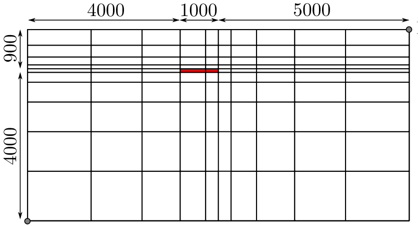
- $E = \frac{(1-2\nu)(1+\nu)}{(1-\nu)c_M}$, with $\nu = 0.3$
- $c_M = 0.01241 \left|\sigma'_y\right|^{-1.1342}$ [bar^{-1}]
- $\sigma'_y = -0.12218 \left|y\right|^{1.0766} + 0.1 \left|y\right|$

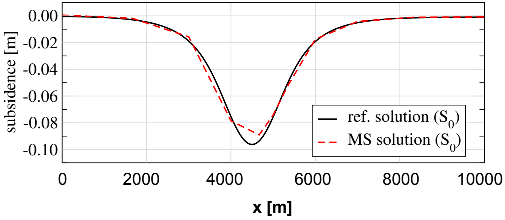
Quantities for E represent the Northern Adriatic basin, Italy [Baù et al., Geotechnique (2002)]

Plane-strain subsidence

- depletion Δp = -100 bar
- Fine grid: 320×320 $(206,082)$
- MSFE grid: 10×10 (242)

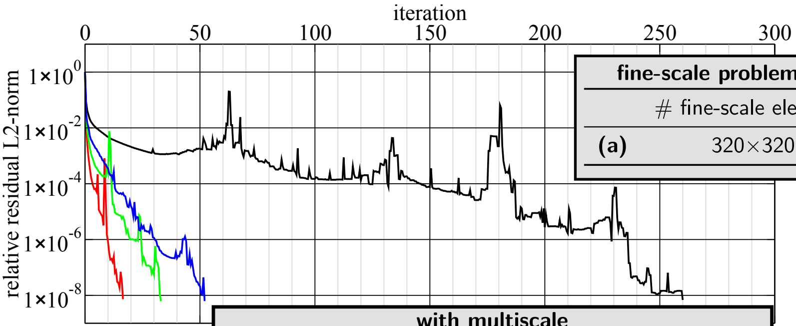
- $E = \frac{(1-2\nu)(1+\nu)}{(1-\nu)c_M}$, with $\nu = 0.3$
- $c_M = 0.01241 \left|\sigma'_y\right|^{-1.1342}$ $[\text{bar}^{-1}]$
- $\sigma'_y = -0.12218 \left|y\right|^{1.0766} + 0.1 \left|y\right|$





Quantities for E represent the Northern Adriatic basin, Italy [Baù et al., Geotechnique (2002)]

Plane-strain subsidence: ILU(0)-BiCGStab performance



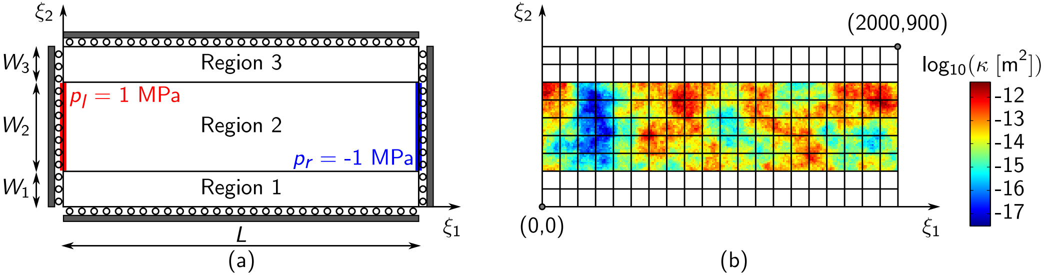
fine-scale problem without multiscale

	# fine-scale elements	# d.o.f.	iter
(a)	320×320	206,082	260

with multiscale

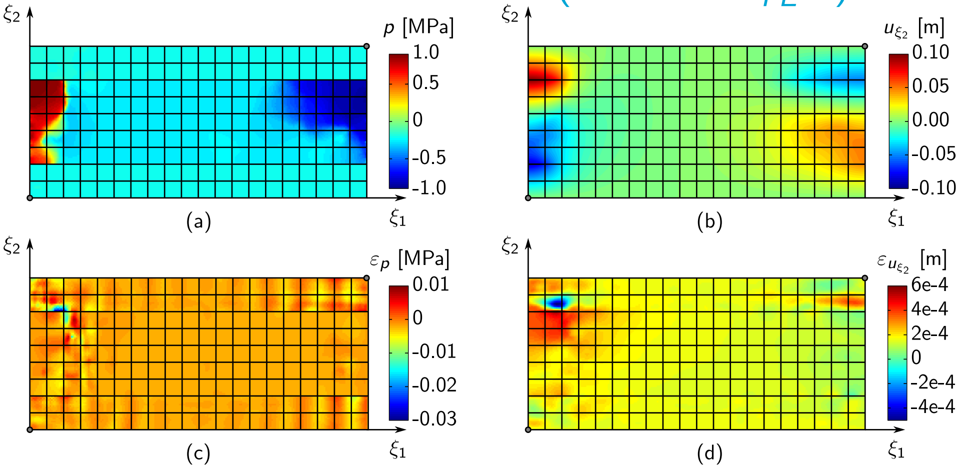
	# coarse-scale elements	# d.o.f.	C_r	iter
(b)	40×40	3,362	61.3	17
(c)	20×20	882	233.7	33
(d)	10×10	242	851.6	52

Heterogeneous synthetic reservoir



- fine-scale: 220×99 ($n_{\mathbf{d}}^h = 43,558$; $n_p^h = 21,780$)
- coarse-scale: 20×9 ($n_{\mathbf{d}}^H = 358$; $n_p^H = 180$)

$\mathcal{M}$-preconditioned GMRES solution ($R^{(p,p)} = R^{(p,p)}_{FE}$)



- time $t = 10$ days (constant timestep $\Delta t = 0.1$ day)
- relative residual tolerance, $\tau = 10^{-6}$; # avg. iter. per timestep 25

Summary

- Multiscale method for flow in elastic porous media
  - **Efficient:** $(RAP)^{-1}$ instead of A^{-1}, P & R local and computed once!
  - **Accurate:** basis functions (P) capture heterogeneity
  - **Reliable:** error reduction to any desired level (combined with a smoother)

$\Rightarrow$ Promising for field-scale (km) studies

Acknowledgments

- TOTAL S.A. (STEMS project)
- SUPRI-B (Stanford)
- Schlumberger
- DARSim (Delft Advanced Reservoir Simulation)

Thank you!

Model problem: Biot's poroelasticity equations

Find the displacement vector **u**(x, t) and pressure **p**(x, t):

$$\nabla \cdot \underbrace{(\mathbf{C}_{dr} : \nabla^s \boldsymbol{u} - bp\mathbf{1})}_{\text{Stress tensor}} = \boldsymbol{f} \qquad \text{(linear momentum balance)}$$

$$\frac{\partial}{\partial t}\underbrace{\left(b\nabla \cdot \boldsymbol{u} + \frac{1}{M_b}p\right)}_{\text{accumulation}} + \nabla \cdot \underbrace{\left(-\frac{\boldsymbol{\kappa}}{\mu}\nabla p\right)}_{\text{Darcy's velocity}} = g \qquad \text{(mass balance)}$$

Voigt notation:

$$\bar{\sigma} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} = \begin{bmatrix} (\lambda + 2\mu) & \lambda & 0 \\ \lambda & (\lambda + 2\mu) & 0 \\ 0 & 0 & \mu \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} = \begin{bmatrix} (\lambda + 2\mu)\frac{\partial u_x}{\partial x} + \lambda \frac{\partial u_y}{\partial y} \\ \lambda \frac{\partial u_x}{\partial x} + (\lambda + 2\mu)\frac{\partial u_y}{\partial y} \\ \mu \frac{\partial u_x}{\partial y} + \mu \frac{\partial u_y}{\partial x} \end{bmatrix}$$

$$\nabla \cdot \bar{\sigma} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & \frac{\partial}{\partial y} \\ 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \bar{\sigma}$$

λ and μ are 1st and 2nd Lame parameters

H. F. Wang, Theory of Linear Poroelasticity, Princeton University (2000)

Reduced boundary problem

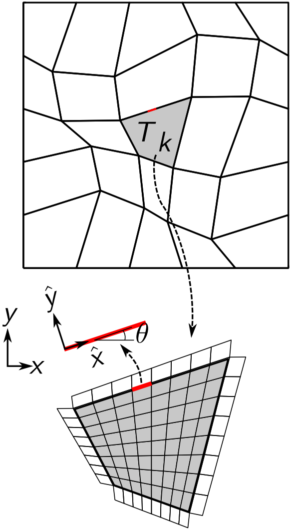
$$\hat{\mathbf{C}}_{dr} = \begin{bmatrix} \hat{K}_v & (\hat{K}_v - 2\hat{G}) & 0 \\ (\hat{K}_v - 2\hat{G}) & \hat{K}_v & 0 \\ 0 & 0 & \hat{G} \end{bmatrix}$$

$$\hat{\nabla}^s = \left(\hat{\nabla}\cdot\right)^T = \begin{bmatrix} \frac{\partial}{\partial \hat{x}} & 0 \\ 0 & \frac{\partial}{\partial \hat{y}} \\ \frac{\partial}{\partial \hat{y}} & \frac{\partial}{\partial \hat{x}} \end{bmatrix}$$

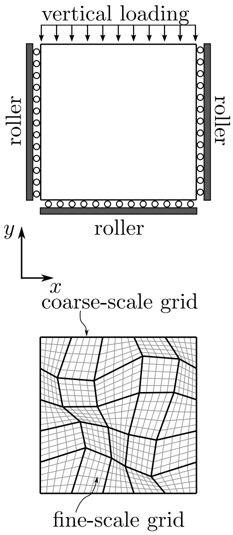
$$\hat{\nabla}^s_{\parallel} = \left(\hat{\nabla}_{\parallel}\right)^T = \begin{bmatrix} \frac{\partial}{\partial \hat{x}} & 0 \\ 0 & 0 \\ 0 & \frac{\partial}{\partial \hat{x}} \end{bmatrix}$$

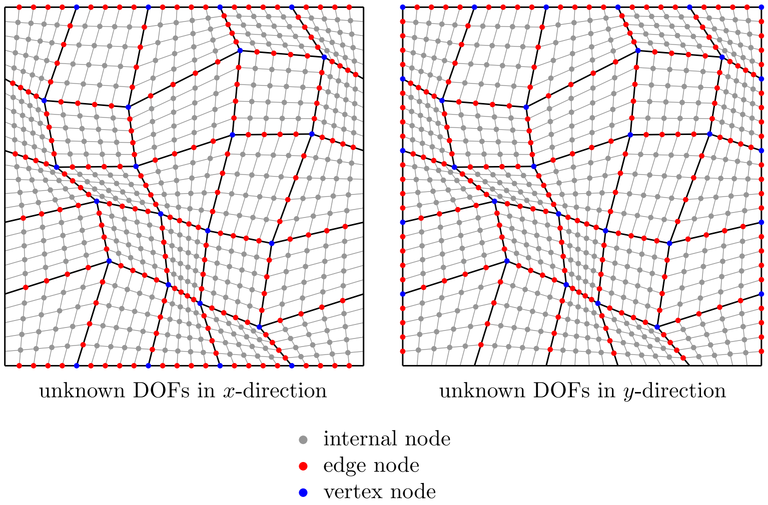
$$\begin{cases} \dfrac{\partial}{\partial \hat{x}} \left(\hat{K}_v \dfrac{\partial \hat{N}^H_{i_{\hat{x}}}}{\partial \hat{x}} \right) = 0 & \text{(axial equilibrium)} \\[2ex] \dfrac{\partial}{\partial \hat{x}} \left(\hat{G} \dfrac{\partial \hat{N}^H_{i_{\hat{y}}}}{\partial \hat{x}} \right) = 0 & \text{(transverse equilibrium)} \end{cases}$$

$\hat{K}_v$, $\hat{G}$: averaging procedure, e.g. max value [Buck et al., DDMSE XXI (2014)]



Wire-basket decomposition of the fine-scale problem





Wire-basket decomposition of the fine-scale problem

Given the permutation matrix W associated with the wire-basket permutation, the fine-scale system can be rewritten as

$$\boxed{\hat{K}_h \hat{\boldsymbol{d}}^h = \hat{f}^h}$$

with

$$\hat{K}_h = W^T K W = \begin{bmatrix} \hat{K}_{II} & \hat{K}_{IE} & \hat{K}_{IV} \\ \hat{K}_{EI} & \hat{K}_{EE} & \hat{K}_{EV} \\ \hat{K}_{VI} & \hat{K}_{VE} & \hat{K}_{VV} \end{bmatrix}, \quad \hat{\boldsymbol{d}}^h = W^T \boldsymbol{d}^h = \begin{bmatrix} \hat{\boldsymbol{d}}_I \\ \hat{\boldsymbol{d}}_E \\ \hat{\boldsymbol{d}}_V \end{bmatrix}, \quad \hat{\boldsymbol{f}}^h = W^T \boldsymbol{d}^h = \begin{bmatrix} \hat{\boldsymbol{f}}_I \\ \hat{\boldsymbol{f}}_E \\ \hat{\boldsymbol{f}}_V \end{bmatrix}$$

The subscripts I, E, and V denote the internal, edge, and vertex DOFs

Wire-basket decomposition of the fine-scale problem

Gaussian elimination of the first block row leads to:

$$\begin{bmatrix} \hat{K}_{II} & \hat{K}_{IE} & \hat{K}_{IV} \\ 0 & \hat{S}_{EE} & \hat{S}_{EV} \\ 0 & \hat{S}_{VE} & \hat{S}_{VV} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{d}}_I \\ \hat{\boldsymbol{d}}_E \\ \hat{\boldsymbol{d}}_V \end{bmatrix} = \begin{bmatrix} I & 0 & 0 \\ -\hat{K}_{EI}\hat{K}_{II}^{-1} & I & 0 \\ -\hat{K}_{VI}\hat{K}_{II}^{-1} & 0 & I \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{f}}_I \\ \hat{\boldsymbol{f}}_E \\ \hat{\boldsymbol{f}}_V \end{bmatrix},$$

where $\hat{S}_{ij}$ are the blocks of the Schur complement matrix $\hat{S}$, i.e., $\hat{S}_{ij} = \hat{K}_{ij} - \hat{K}_{iI}\hat{K}_{II}^{-1}\hat{K}_{Ij}, \forall(i,j) \in \{E,V\} \times \{E,V\}$. The reduced boundary condition is an approximation to the second block row, i.e.,

$$\tilde{K}_{EE}\hat{\boldsymbol{d}}_E + \tilde{K}_{EV}\hat{\boldsymbol{d}}_V = \boldsymbol{0},$$

Preconditioner computation and application

Algorithm 1: M^{-1} PRECONDITIONER COMPUTATION

1: **Compute** P_H^h
2: **Compute** $R_h^H = (P_H^h)^T$
3: **Compute** $K_H = R_h^H K_h P_H^h$
4: **Factorize** $K_H = L_{K_H} U_{K_H}$ with a direct solver
5: **Compute** M_{sm}^{-1} as a local fine-scale smoothing preconditioner of K_h

Preconditioner computation and application

Algorithm 2: M_{MS}^{-1} PRECONDITIONER APPLICATION

1: **Input**: $\boldsymbol{v}^h$; **Output**: $\boldsymbol{z}^h = M_{MS}^{-1}\boldsymbol{v}^h$
2: **Compute** $\boldsymbol{v}^H = R_h^H \boldsymbol{v}^h$
3: **Compute** $\boldsymbol{z}^H = L_{K_H}^{-1} \boldsymbol{v}^H$
4: **Compute** $\boldsymbol{v}^H = U_{K_H}^{-1} \boldsymbol{z}^H$
5: **Compute** $\boldsymbol{z}^h = P_H^h \boldsymbol{v}^H$

Preconditioner computation and application

Algorithm 3: M^{-1} PRECONDITIONER APPLICATION

1: **Input**: $\boldsymbol{v}^h$; **Output**: $\boldsymbol{z}^h = M^{-1}\boldsymbol{v}^h$
2: **Apply** M_{MS}^{-1} to $\boldsymbol{v}^h$ to get $\boldsymbol{z}_0^h$
3: **Compute** $\boldsymbol{r}_0^h = K_h \boldsymbol{z}_0^h$
4: **Compute** $\boldsymbol{r}_0^h \leftarrow \boldsymbol{v}^h - \boldsymbol{r}_0^h$
5: **Apply** M_{sm}^{-1} to $\boldsymbol{r}_0^h$ to get $\boldsymbol{z}^h$
6: **Compute** $\boldsymbol{z}^h \leftarrow \boldsymbol{z}_0^h + \boldsymbol{z}^h$

Pressure multiscale basis functions

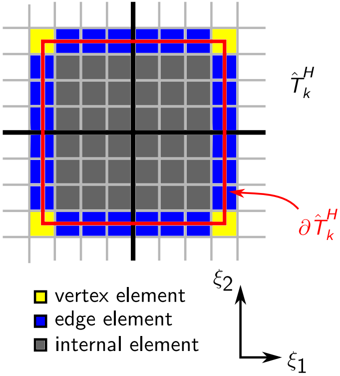
Computed solving incompressible mass balance over each coarse element

Given $\mathcal{S}_p^h = \text{span}\left\{N_{p,j}^h(\Omega), j = 1, \ldots, n_p^h\right\}$, find $N_{p,i}^H : \mathcal{S}_p^h \rightarrow \mathbb{R}$ such that:

$$\nabla \cdot \left(\boldsymbol{\lambda} \cdot \nabla N_{p,i}^H\right) = 0 \quad \text{in } \hat{T}_k$$

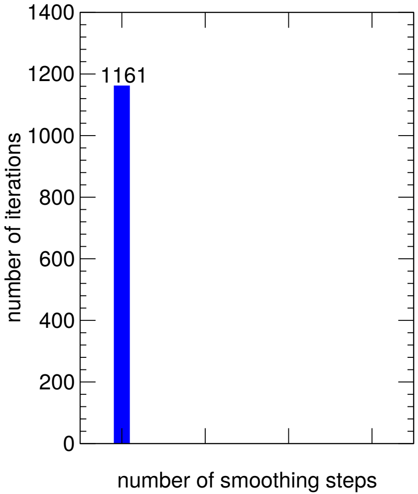
$$\nabla_{\parallel} \cdot \left(\boldsymbol{\lambda} \cdot \nabla N_{p,i}^H\right)_{\parallel} = 0 \quad \text{on } \partial \hat{T}_k$$

$$N_{p,i}^H(\boldsymbol{\xi}_{p,j}^V) = \delta_{ij} \quad \forall j \in \left\{1, \ldots, n_p^H\right\}$$

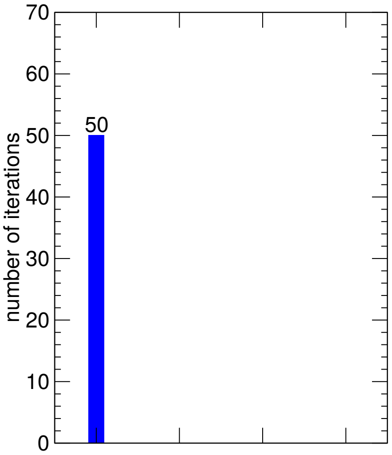


$$N_{p,i}^H = N_p^h \phi_i^p \implies P^{(p,p)} = [\phi_1^p, \ldots, \phi_{n_p^H}^p] \implies N_p^H = N_p^h P^{(p,p)}$$

Skewed mesh, simple shear: iterative

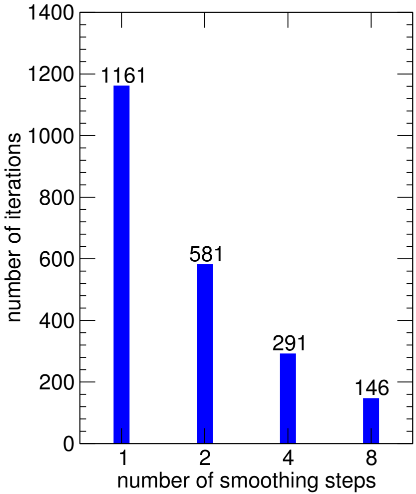


Richardson, ILU(0) smoother

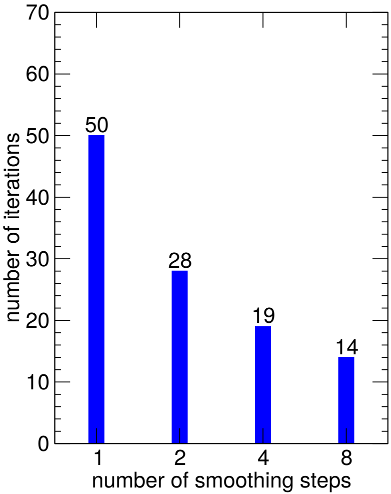


BiCG-Stab, ILU(0) smoother

Skewed mesh, simple shear: iterative



Richardson, ILU(0) smoother



BiCG-Stab, ILU(0) smoother