Multiscale solvers for flow and transport in porous media

Hamdi Tchelepi[1] Sergey Klevtsov[1] Nicola Castelletto[2]

[1]Stanford University

[2]Lawrence Livermore National Laboratory

2021 CRM Summer School
Jun 07, 2021

Acknowledgements:

- Xiaochen Wang
- Marc Hesse
- Hui Zhou
- Yixuan Wang
- Abdul-Rahman Manea
- Arthur Moncorge
- Sebastian Bosma
- Sergey Klevtsov
- Patrick Jenny
- Seong Lee
- Pavel Tomin
- Nicola Castalletto
- Hadi Hajibeygi
- Joshua White
- Olav Moyner
- Brad Mallison

Outline

- Introduction
- MultiScale Formulations
- Flow: MSFE & MSFV
- Algebraic MultiScale (AMS)
- Two-Stage Algebraic Multi-Scale
- Mechanics
- Poro-Mechanics
- Multi-Level MultiScale

Governing equations

- Conservation laws:

$$\frac{\partial}{\partial t}\Big(\phi \sum_{\alpha} x_{\alpha}^{i} \rho_{\alpha} S_{\alpha}\Big) + \nabla \cdot \Big(\sum_{\alpha} x_{\alpha}^{i} \rho_{\alpha} \boldsymbol{v}_{\alpha}\Big) - \sum_{\alpha} x_{\alpha}^{i} \rho_{\alpha} q_{\alpha} = 0 \quad \text{(mass balance of comp. } i\text{)}$$

$$\nabla \cdot (\boldsymbol{\sigma}' - b\bar{p}\mathbf{1}) + \boldsymbol{f} = \mathbf{0} \quad \text{(momentum balance)}$$

- Constitutive models:

$$\boldsymbol{v}_{\alpha} = -\frac{k_{r\alpha}(S_{\alpha})}{\mu_{\alpha}} \boldsymbol{\kappa} \cdot (\nabla p_{\alpha} - \rho_{\alpha} \boldsymbol{g}) \qquad \text{(multiphase Darcy's law)}$$

$$\rho_{\alpha} = \rho_{\alpha}(p_{\alpha}, x_{\alpha}^{i}),\ \mu_{\alpha} = \mu_{\alpha}(p_{\alpha}, x_{\alpha}^{i}), \qquad \text{(fluid properties)}$$

$$p_{\alpha} = p + p_{\alpha}^{c}(S_{\alpha}), \qquad \text{(capillary pressure)}$$

$$\Delta\phi = b\Delta\epsilon_{v}(\boldsymbol{u}) + \frac{(b - \phi_{0})}{K_{s}} \Delta\bar{p}, \qquad \text{(porosity model)}$$

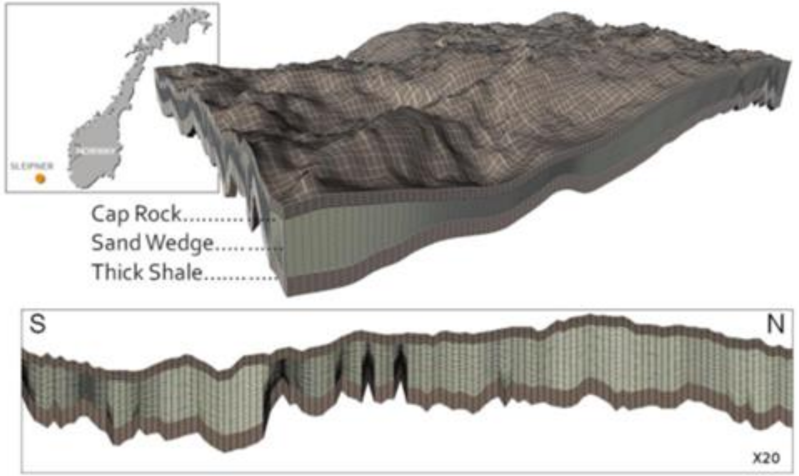
$$\boldsymbol{\sigma}' = \mathbf{C}_{\mathrm{dr}} : \nabla^{s} \boldsymbol{u} \qquad \text{(linear elasticity)}$$

Multiscale methods

Why multiscale solvers?

- Tightly coupled PDEs
- Strongly discontinuous properties
- High fidelity/resolution requirements
- Multiscale nature of the problems

Robust and scalable solvers are in demand!



Sleipner CO_2 storage problem [Cavanagh (2013)].

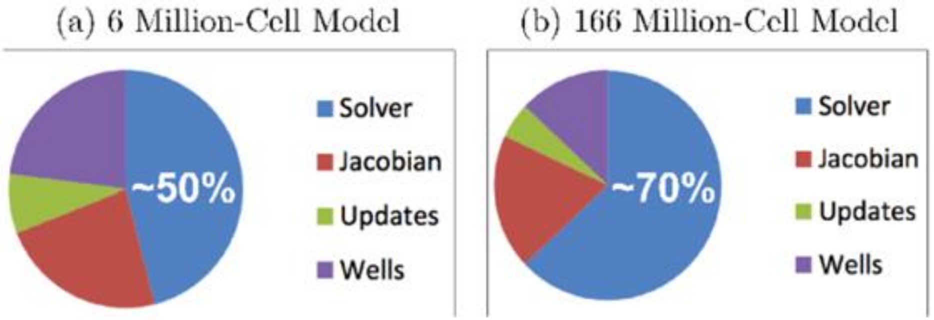
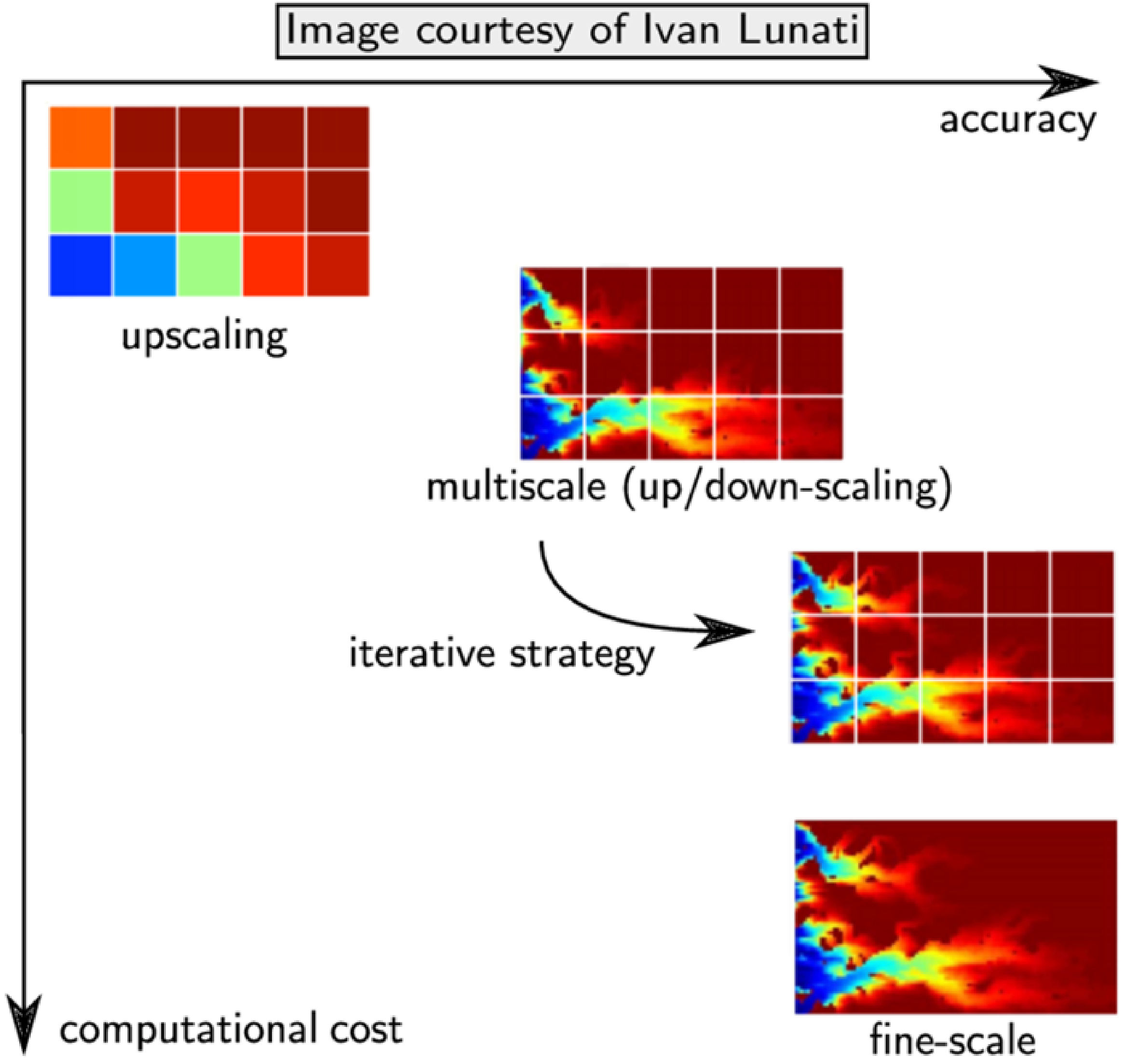


Figure 1.1: Performance breakdown for a simulation of 37 years of production using a compositional model on 720 Westmere cores and 5,640 Westmere cores, performed by Saudi ARAMCO's in-house reservoir simulator, GigaPOWERS^{TM}.

Manea, PhD Thesis [2015]

Multiscale methods



Upscaling Methods

- Effective Coefficients
- Pseudo Functions
- Volume Averaging
- Homogenization methods

Multiscale Methods

- Heterogeneous Multiscale Methods
- Variational Multiscale Method
- Multiscale Finite Element Method
- Multiscale Finite Volume Method
- Multiscale Mixed Finite Element Method
- ...

Governing equations

- Conservation laws:

$$\frac{\partial}{\partial t}\left(\phi \sum_\alpha x^i_\alpha \rho_\alpha S_\alpha\right) + \nabla \cdot \left(\sum_\alpha x^i_\alpha \rho_\alpha \boldsymbol{v}_\alpha\right) - \sum_\alpha x^i_\alpha \rho_\alpha q_\alpha = 0 \quad \text{(mass balance of comp. } i\text{)}$$

$$\nabla \cdot (\boldsymbol{\sigma}' - b\bar{p}\mathbf{1}) + \boldsymbol{f} = \mathbf{0} \quad \text{(momentum balance)}$$

- Constitutive models:

$$\boldsymbol{v}_\alpha = -\frac{k_{r\alpha}(S_\alpha)}{\mu_\alpha}\boldsymbol{\kappa} \cdot (\nabla p_\alpha - \rho_\alpha \boldsymbol{g}) \quad \text{(multiphase Darcy's law)}$$

$$\rho_\alpha = \rho_\alpha(p_\alpha, x^i_\alpha),\ \mu_\alpha = \mu_\alpha(p_\alpha, x^i_\alpha), \quad \text{(fluid properties)}$$

$$p_\alpha = p + p^c_\alpha(S_\alpha), \quad \text{(capillary pressure)}$$

$$\Delta\phi = b\Delta\epsilon_v(\boldsymbol{u}) + \frac{(b-\phi_0)}{K_s}\Delta\bar{p}, \quad \text{(porosity model)}$$

$$\boldsymbol{\sigma}' = \mathbf{C}_{\mathrm{dr}} : \nabla^s \boldsymbol{u} \quad \text{(linear elasticity)}$$

MultiScale Methods for Flow & Transport:

- MSFV: Flow Problem
- MSFV: IMPES: Adaptivity of Flow & Transport
- Three-Phase Flow with Interphase Mass Transfer
- Sequential Fully Implicit (SFI)
- Wells
- Compressible Multi-Phase Flow
- Nonlinear Transport: Adaptivity
- Compositional Flow & Transport
- Poro-Mechanics

Multiscale Finite Element Method (MSFE)

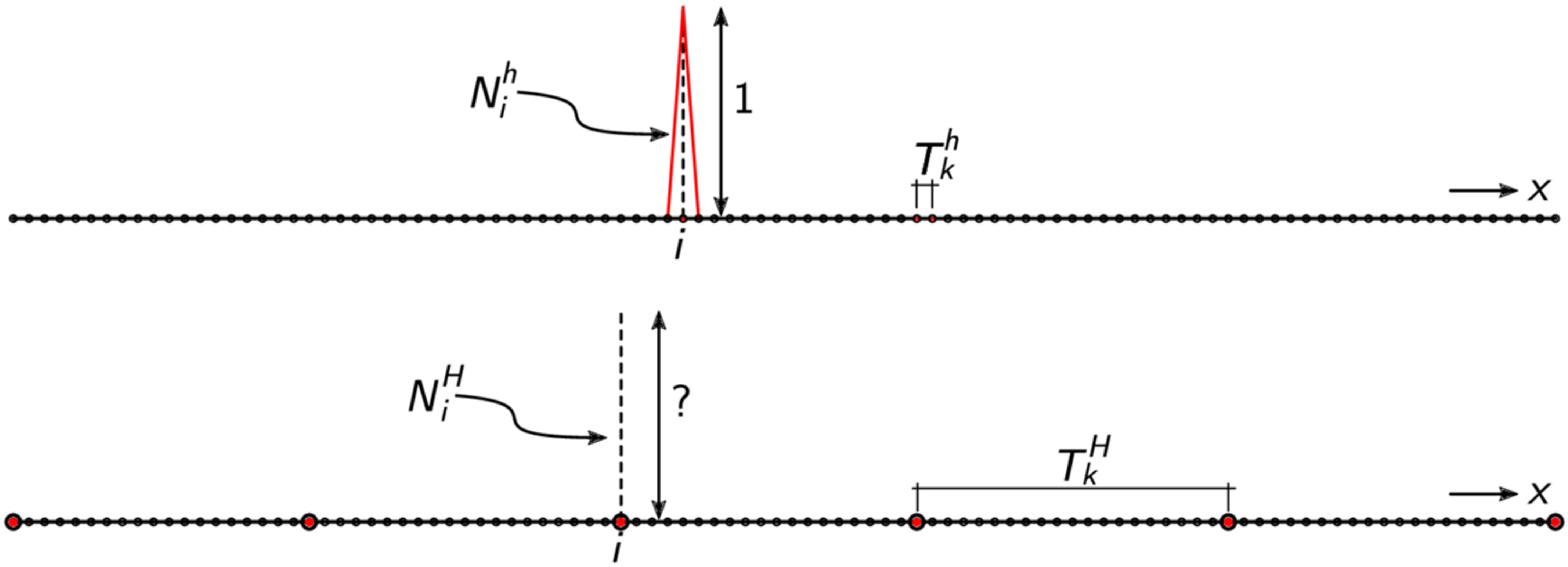
- Super-element method: Fedorenko (1976)
- Special/generalized finite element method: Babuska (1983, 1994)

- Multiscale finite element method: Hou & Wu (1997)

- Extensions and related work:
  - ▷ Multiscale mixed finite element method (Chen & Hou, 2003)
  - ▷ Multiscale mortar method (Arbogast et al., 2007)
  - ▷ Use of limited global information (Efendiev & Hou, 2007)
  - ▷ Application to vector problems (Efendiev & Hou, 2009; Buck et al. 2013)
  - ▷ ...and many other

Multiscale Finite Element Method (MSFE)

A coarse mesh is introduced ($\mathcal{T}^H$) along with new basis functions (N_i^H):

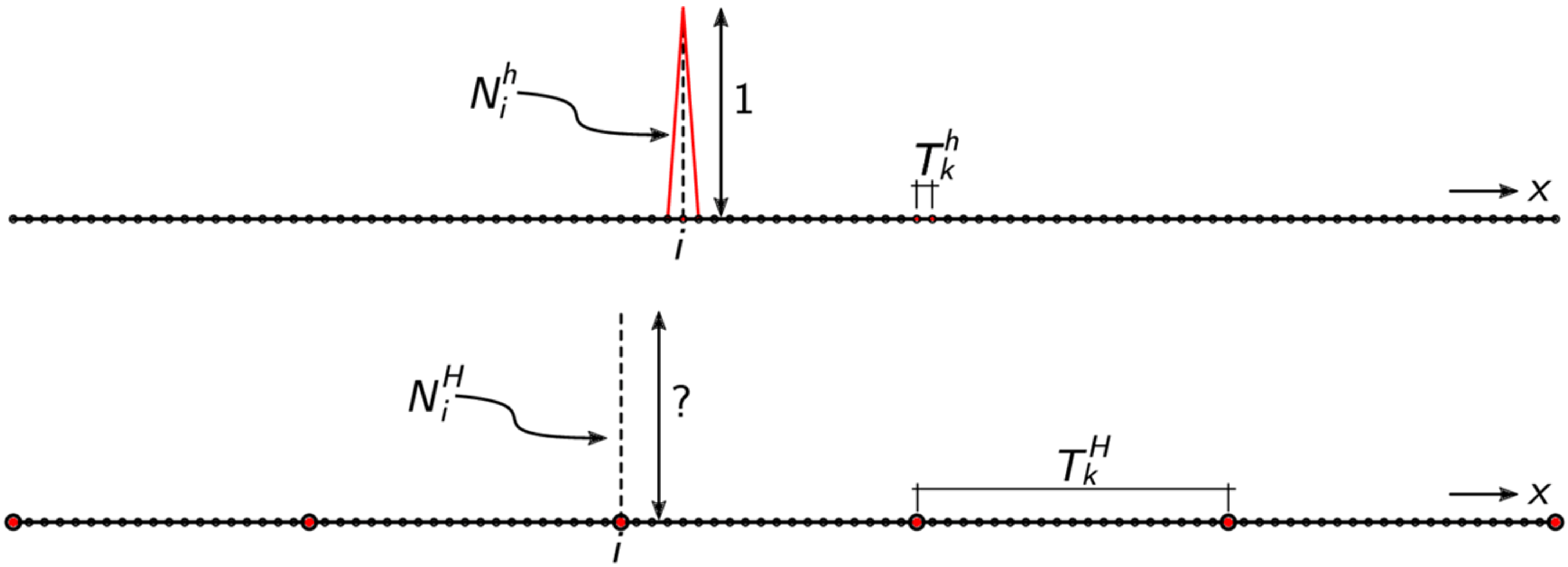


- ▷ We want to solve:

$$\nabla \cdot [\boldsymbol{\kappa}(\boldsymbol{x})\nabla\phi] + f = 0, \quad \text{in } \Omega$$
$$\phi = g, \quad \text{on } \partial\Omega$$

Multiscale Finite Element Method (MSFE)

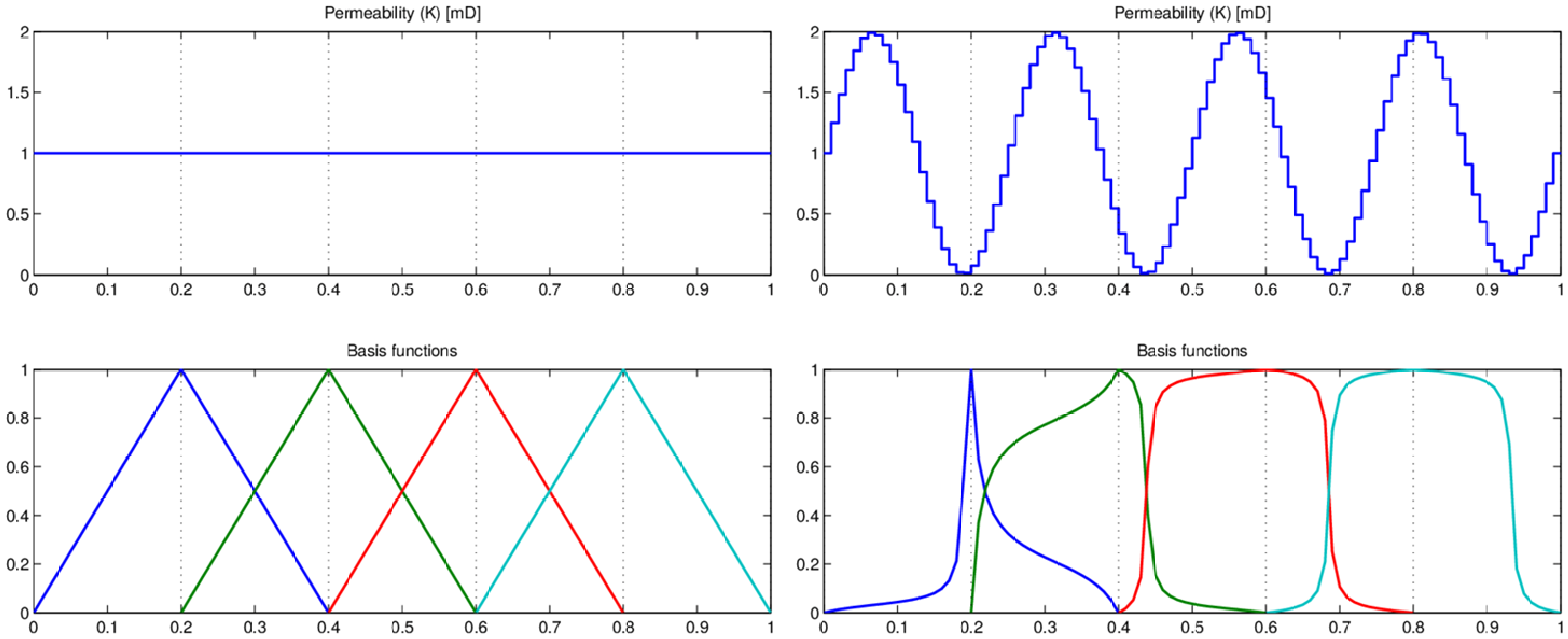
A coarse mesh is introduced ($\mathcal{T}^H$) along with new basis functions (N_i^H):



▷ The coarse basis functions satisfy:

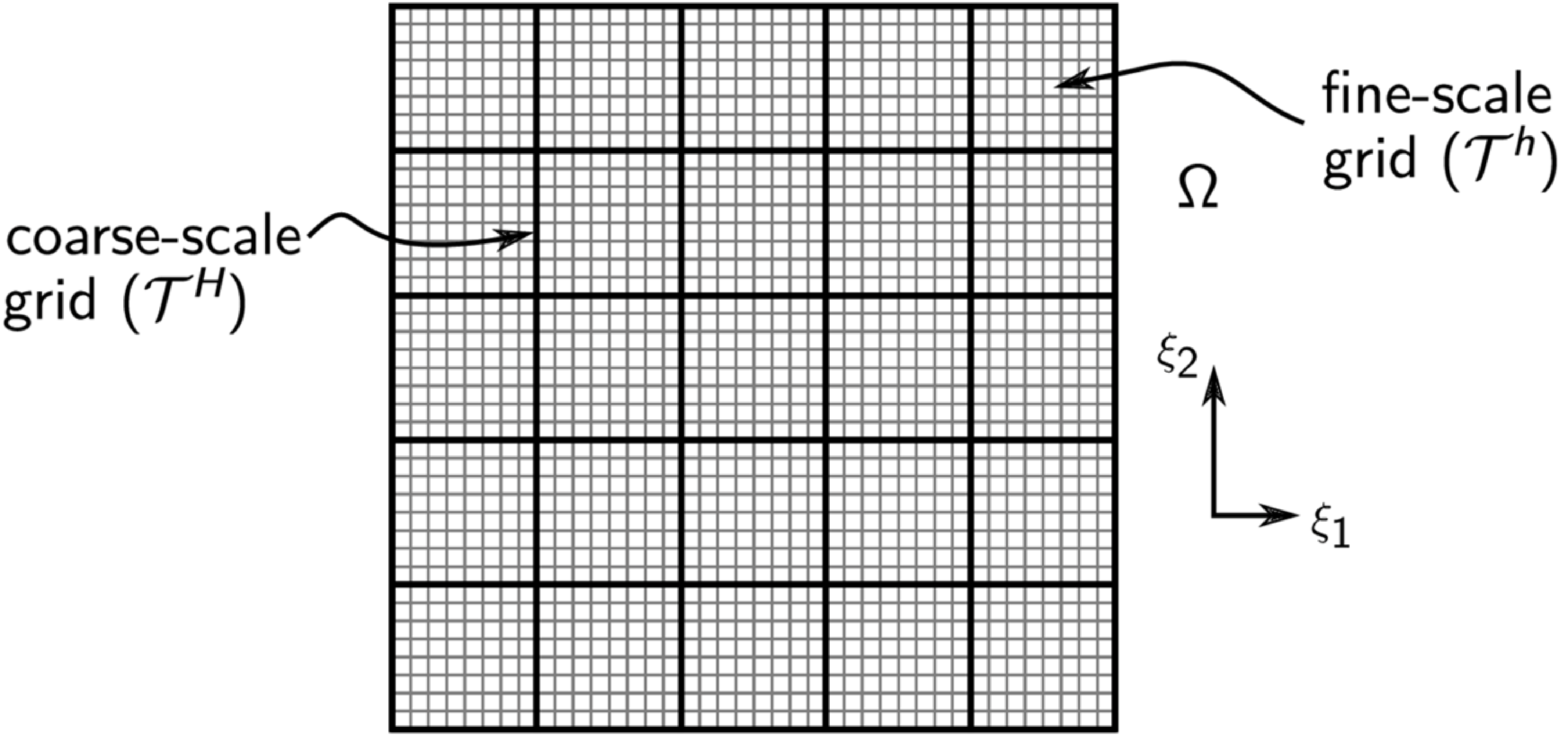
$$\nabla \cdot \left[\boldsymbol{\kappa}(\boldsymbol{x}) \nabla N_i^H \right] = 0, \quad \text{in } T_k^H$$
$$N_i^H(\boldsymbol{x}_j) = \delta_{ij}$$

Multiscale Finite Element Method (MSFE)

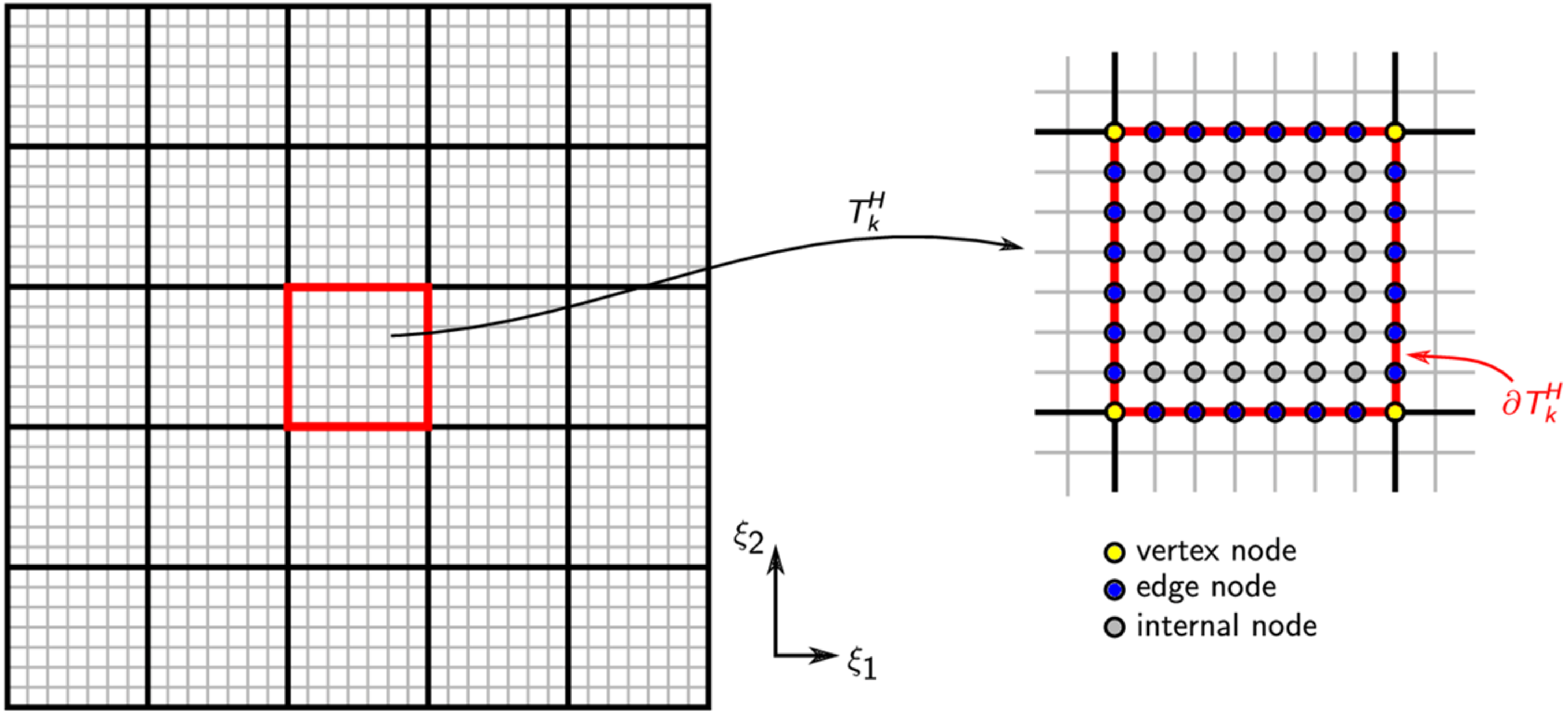


[Møyner & Lie, 2016]

MSFE basis functions



MSFE basis functions



MSFE basis functions

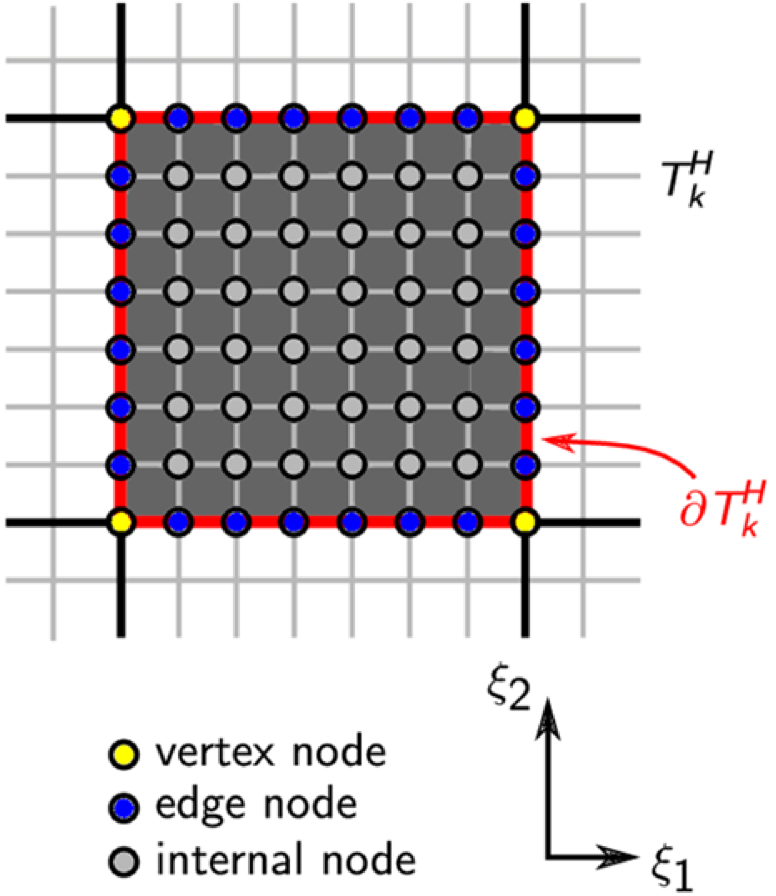
- Computed solving local problems on each coarse element
- Solutions of reduced-dimension problems on boundaries used as boundary conditions (localization assumption)

Given $\mathcal{S}^h = \text{span}\left\{\boldsymbol{N}_j^h(\Omega), j = 1, \dots, n^h\right\}$, find $\boldsymbol{N}_i^H : \mathcal{S}^h \rightarrow \mathbb{R}^2$ such that:

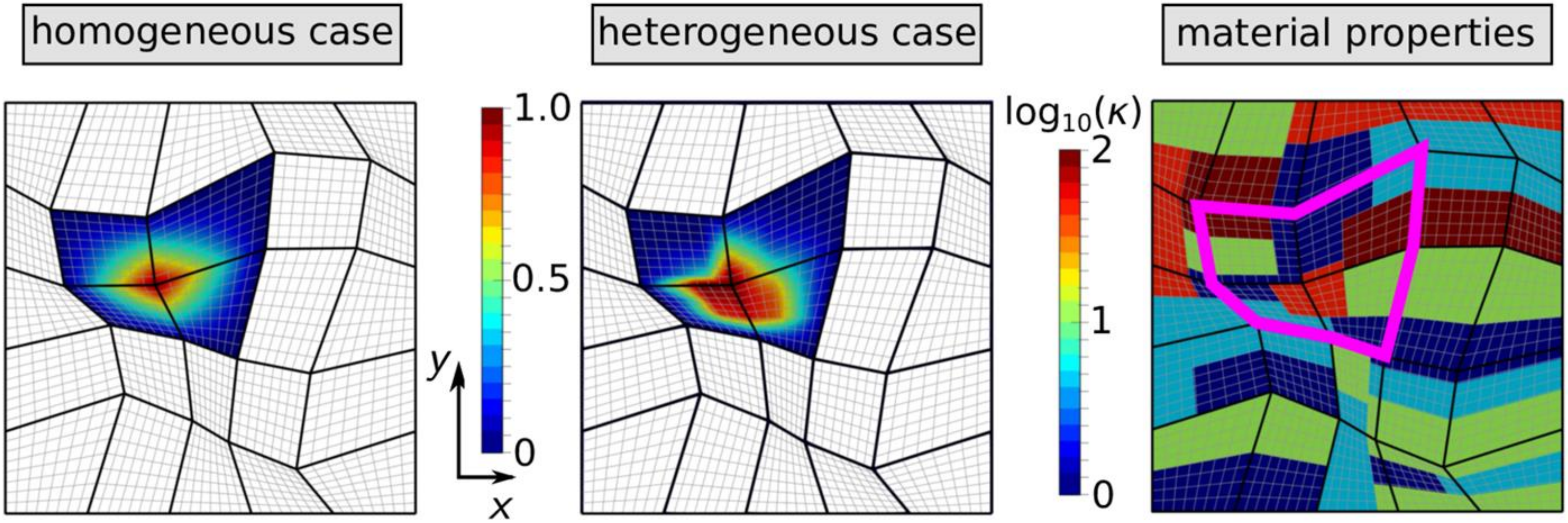
$$\nabla \cdot \left(\boldsymbol{\kappa}(\boldsymbol{x}) \nabla \boldsymbol{N}_i^H\right) = \boldsymbol{0} \qquad \text{in } T_k^H$$

$$\nabla_{\parallel} \cdot \left(\boldsymbol{\kappa}(\boldsymbol{x}) \nabla_{\parallel} \boldsymbol{N}_i^H\right) = \boldsymbol{0} \qquad \text{on } \partial T_k^H$$

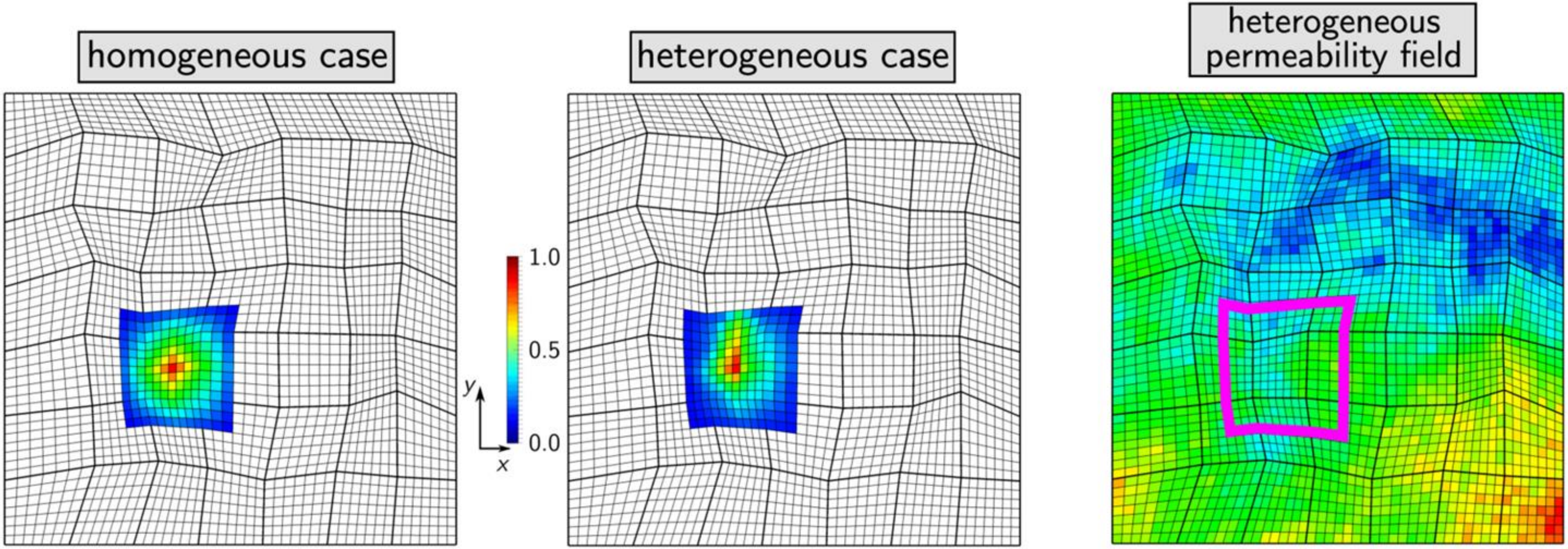
$$\boldsymbol{N}_i^H(\boldsymbol{\xi}_j^V) = \delta_{ij} \qquad \forall j \in \left\{1, \dots, n^H\right\}$$



MSFE basis functions



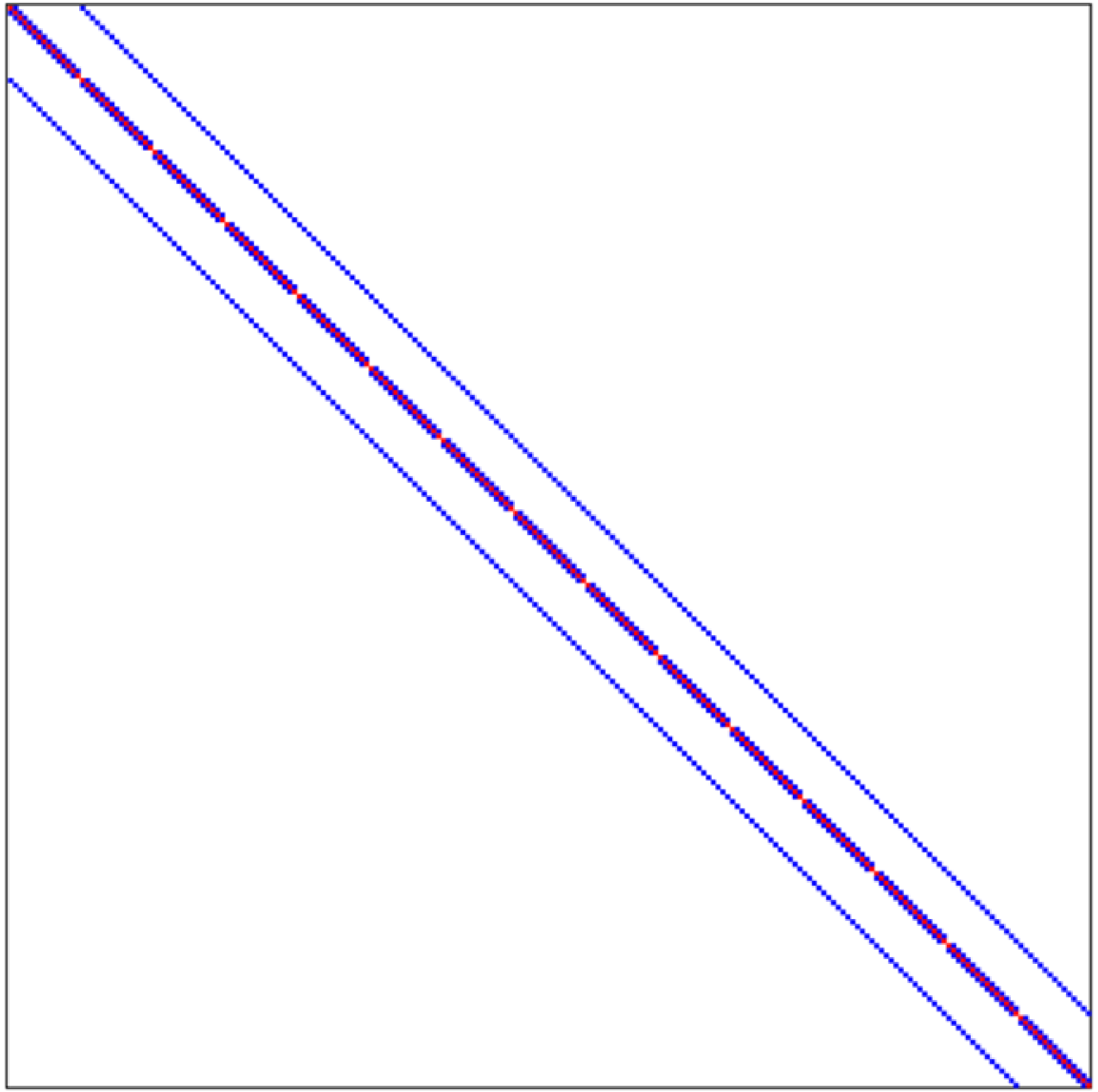
MSFV flow dual basis functions

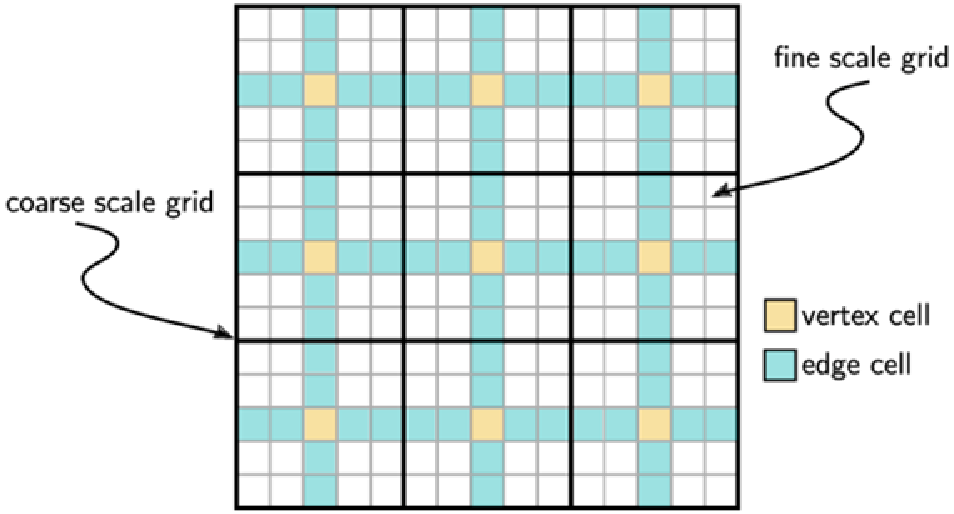


AMS

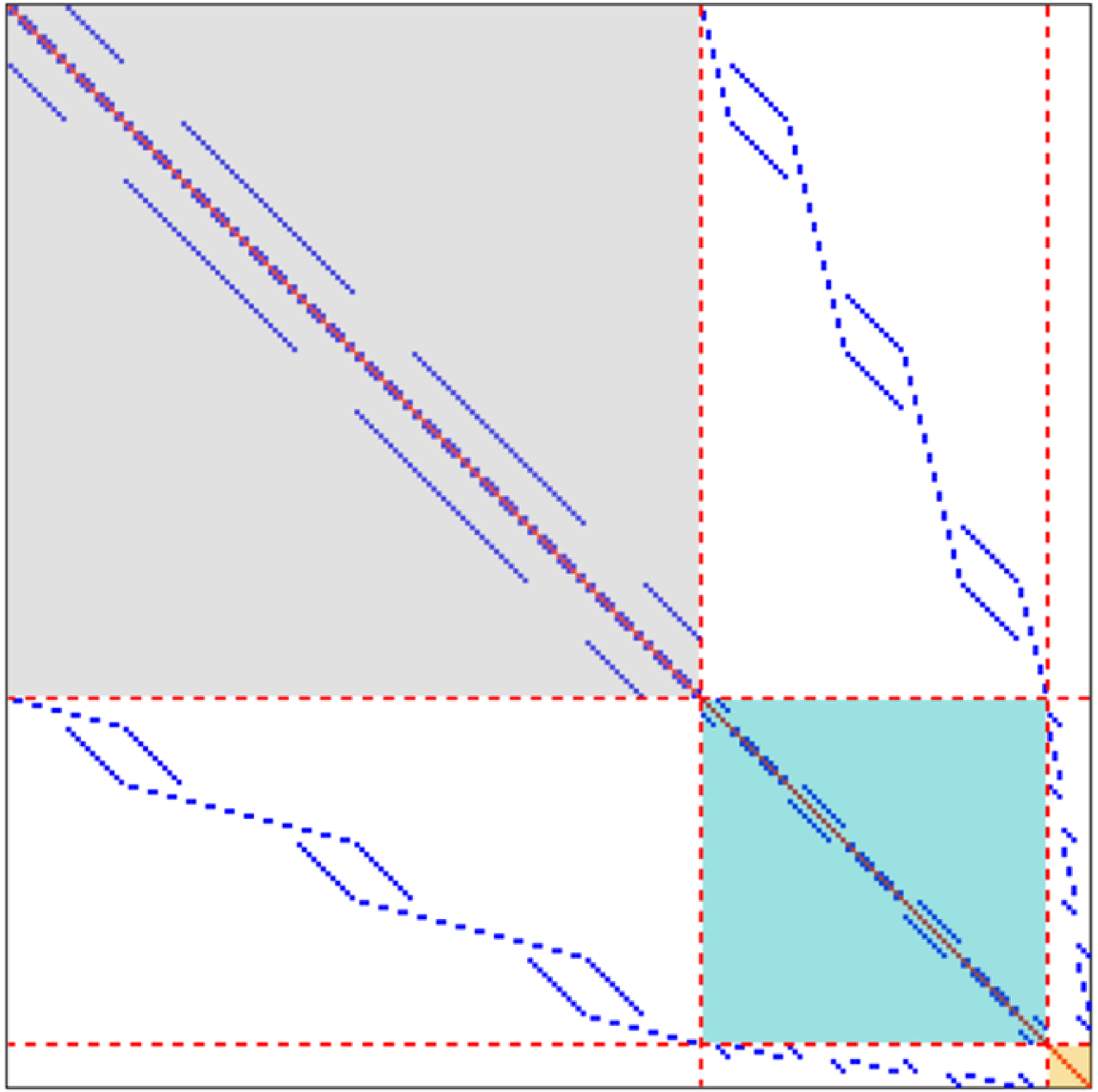
Algebraic MultiScale

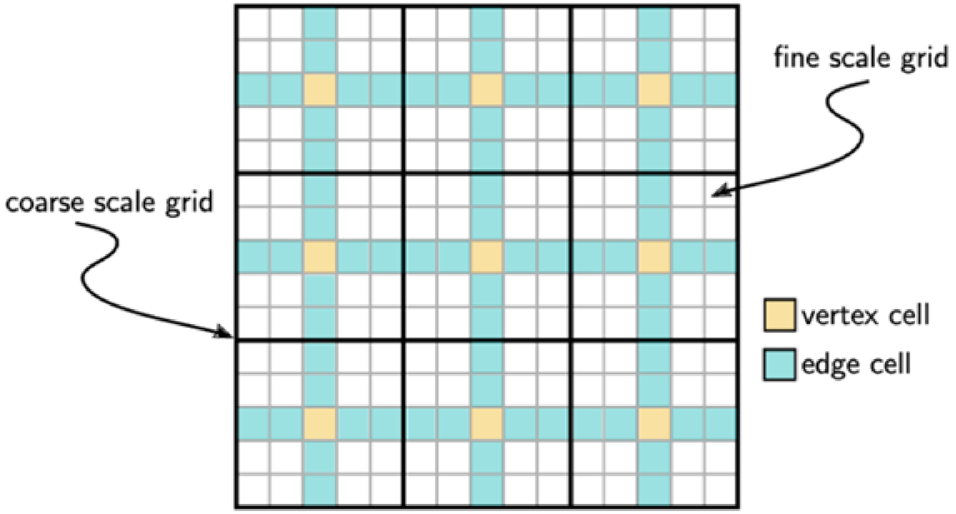
flow dual basis functions - algebraic construction





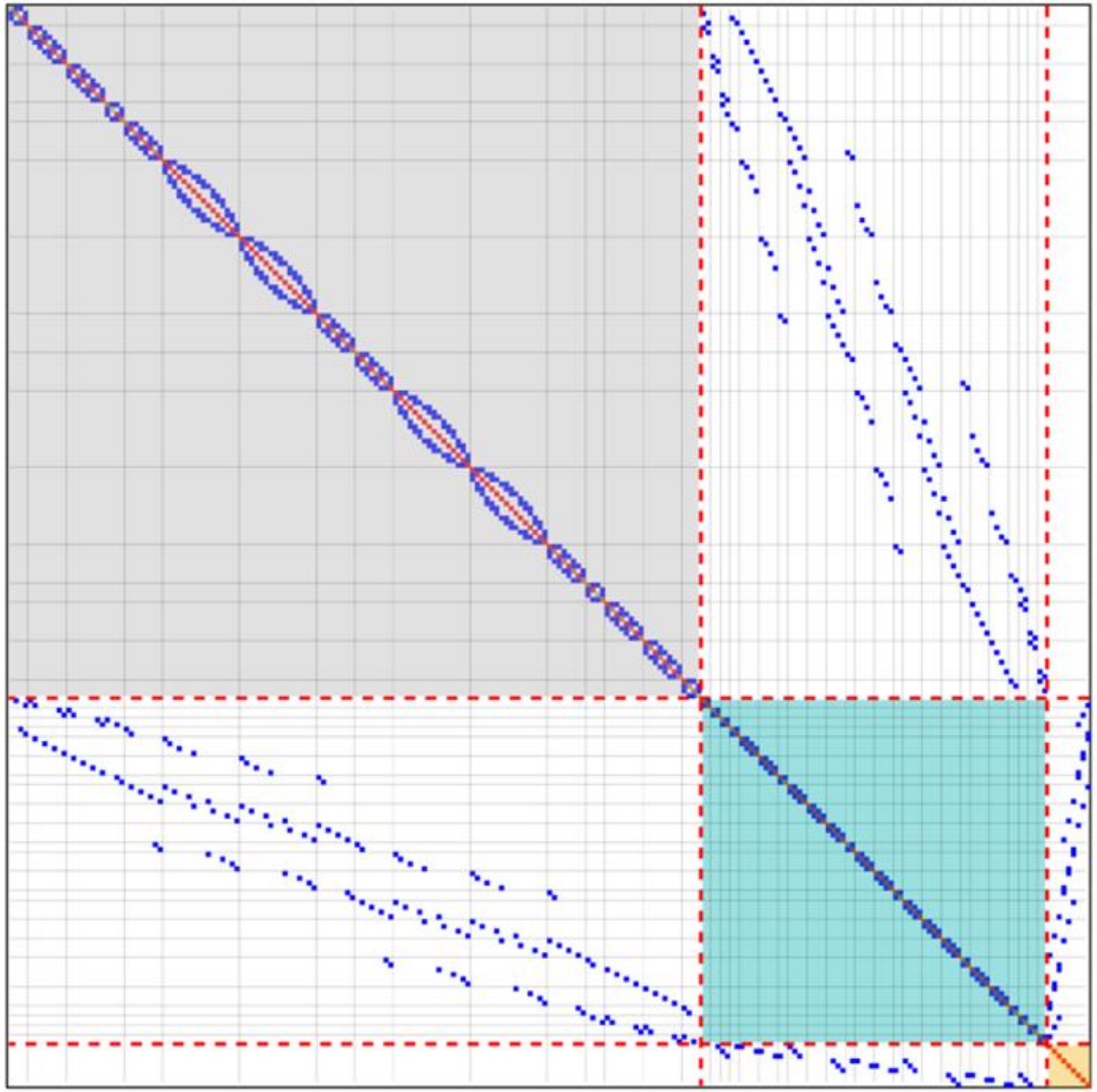
flow dual basis functions - algebraic construction

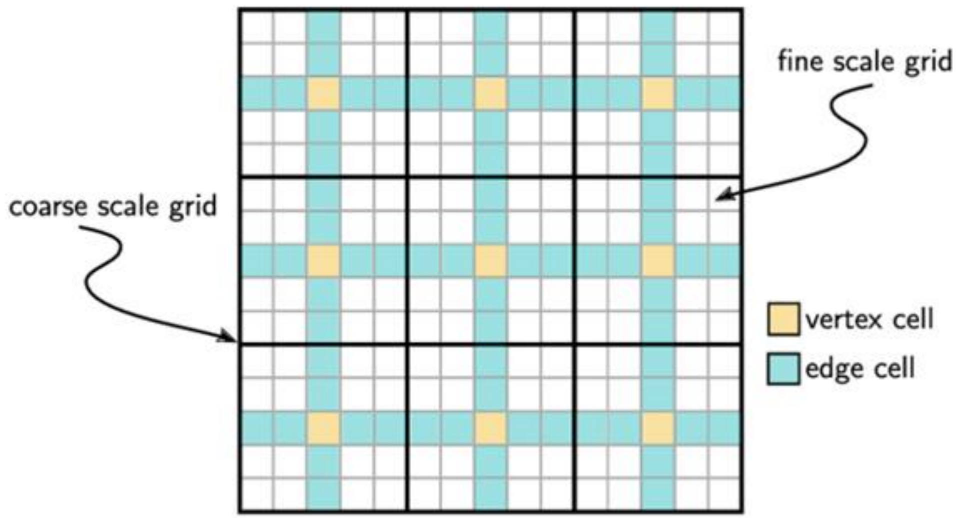




$$\begin{bmatrix} A_{II} & A_{IE} & \\ A_{EI} & A_{EE} & A_{EV} \\ & A_{VE} & A_{VV} \end{bmatrix}$$

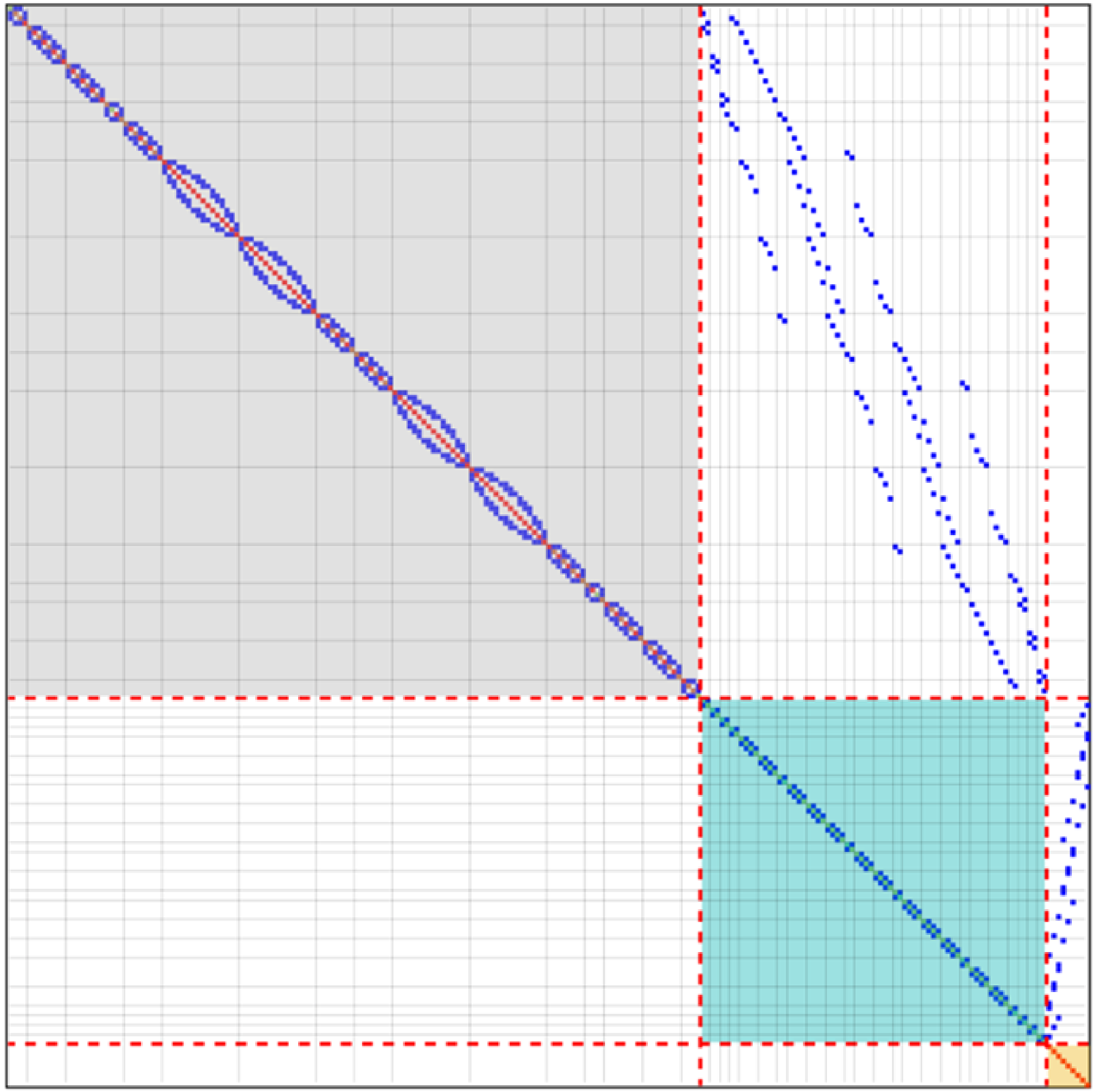
flow dual basis functions - algebraic construction

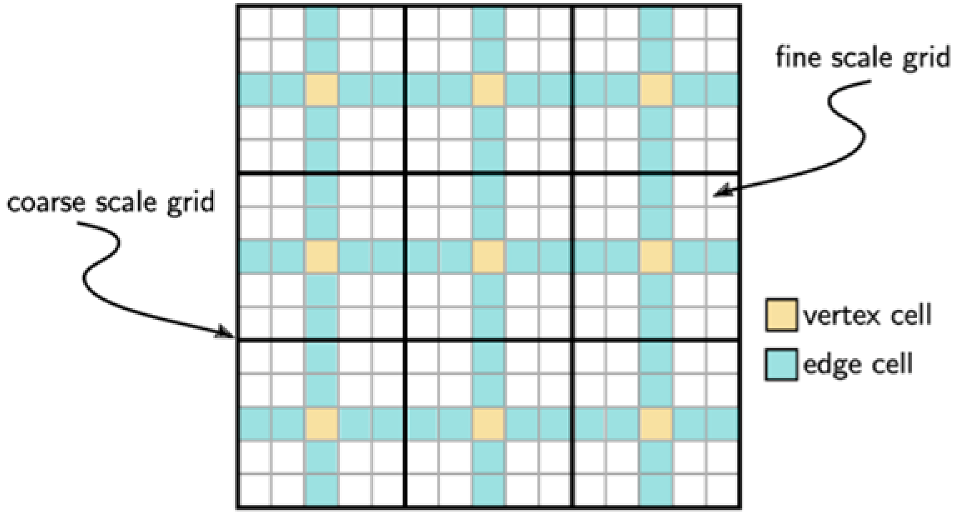




$$\begin{bmatrix} \tilde{A}_{II} & \tilde{A}_{IE} & \\ \tilde{A}_{EI} & \tilde{A}_{EE} & \tilde{A}_{EV} \\ & \tilde{A}_{VE} & \tilde{A}_{VV} \end{bmatrix}$$

flow dual basis functions - algebraic construction





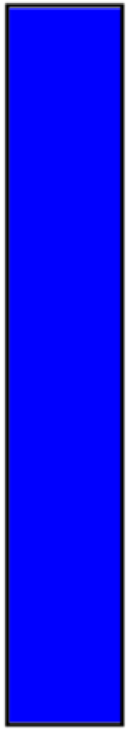
$$\begin{bmatrix} \tilde{A}_{II} & \tilde{A}_{IE} & \\ & \tilde{\tilde{A}}_{EE} & \tilde{A}_{EV} \\ & & I_{VV} \end{bmatrix} \Longrightarrow$$

$$\Longrightarrow P = \begin{bmatrix} P_{IV} \\ P_{EV} \\ I_{VV} \end{bmatrix} = \begin{bmatrix} -\tilde{A}_{II}^{-1}\tilde{A}_{IE}P_{EV} \\ -\tilde{\tilde{A}}_{EE}^{-1}\tilde{A}_{EV} \\ I_{VV} \end{bmatrix}$$

Prolongation and restriction operators

Downscaling

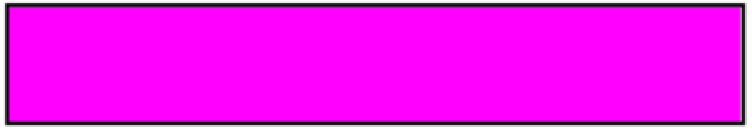
$\mathcal{P} =$



prolongation operator

Upscaling

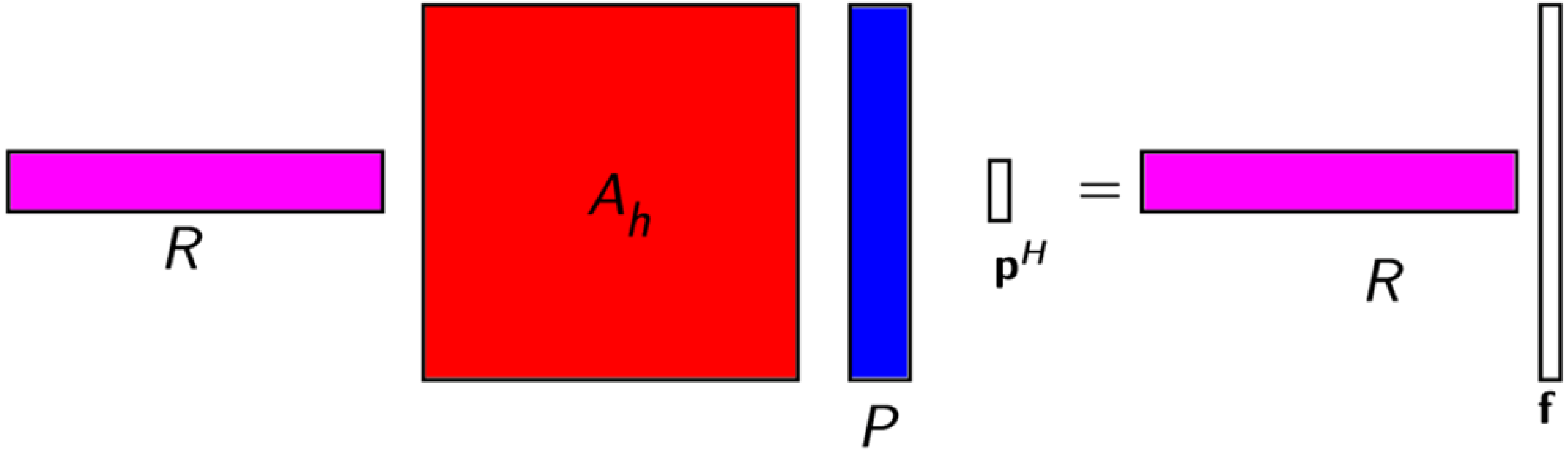
$\mathcal{R} =$



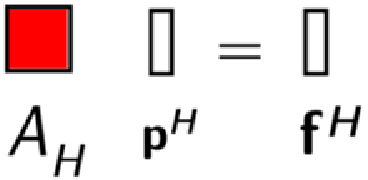
restriction operator

- $\mathcal{P}$ takes vectors on $\mathcal{T}^H$ and defines their analogue in $\mathcal{T}^h$
- $\mathcal{R}$ takes vectors on $\mathcal{T}^h$and defines their analogue in $\mathcal{T}^H$

Prolongation and restriction operators



$$\underbrace{\mathcal{R} A_h \mathcal{P}}_{A_H} \boldsymbol{p}^H = \underbrace{\mathcal{R}\boldsymbol{b}}_{\boldsymbol{b}^H}$$

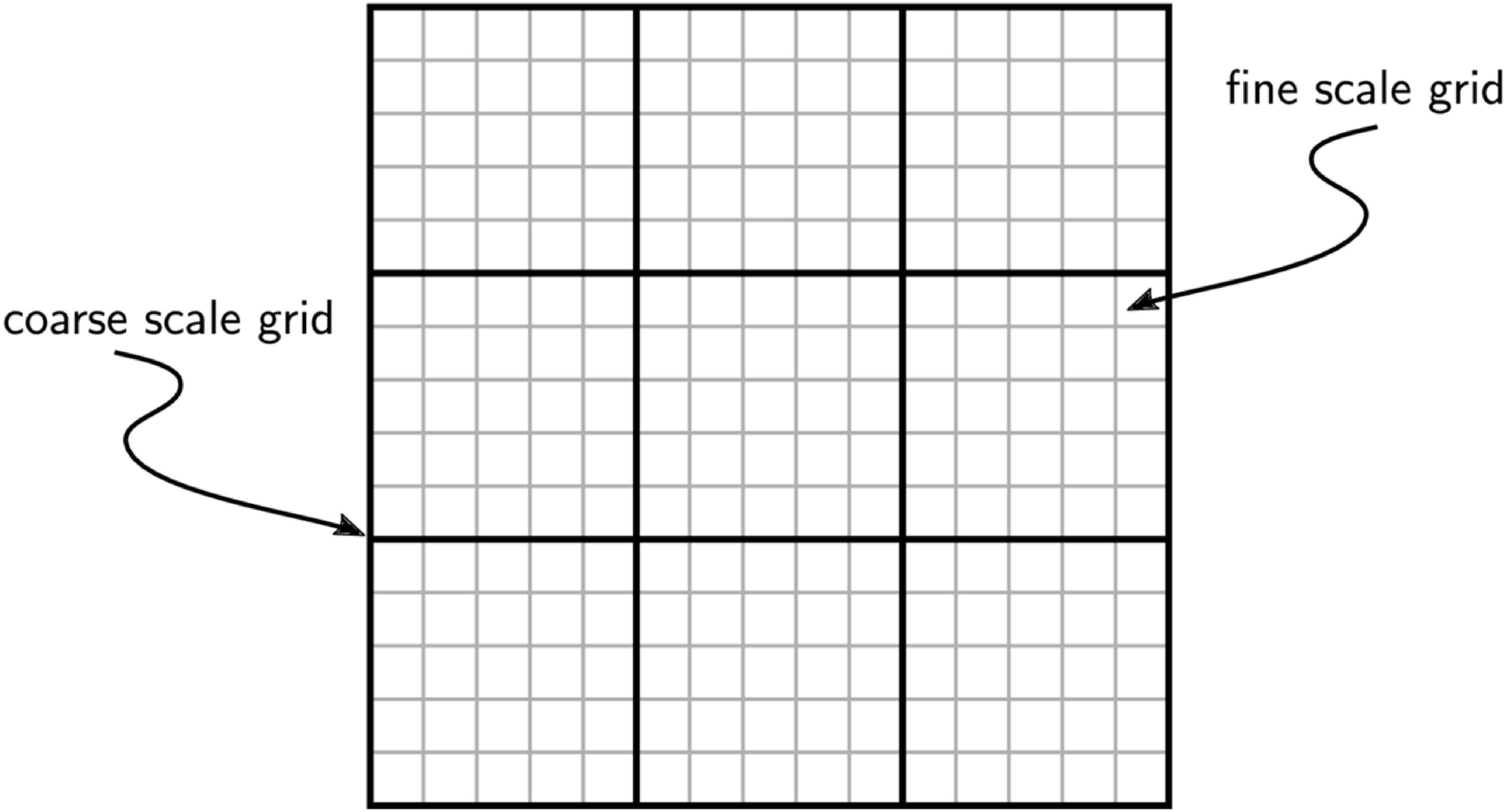


Solve for $\boldsymbol{p}^H$, then compute $\boldsymbol{p}^h_{MS} = \mathcal{P}\boldsymbol{p}^H$

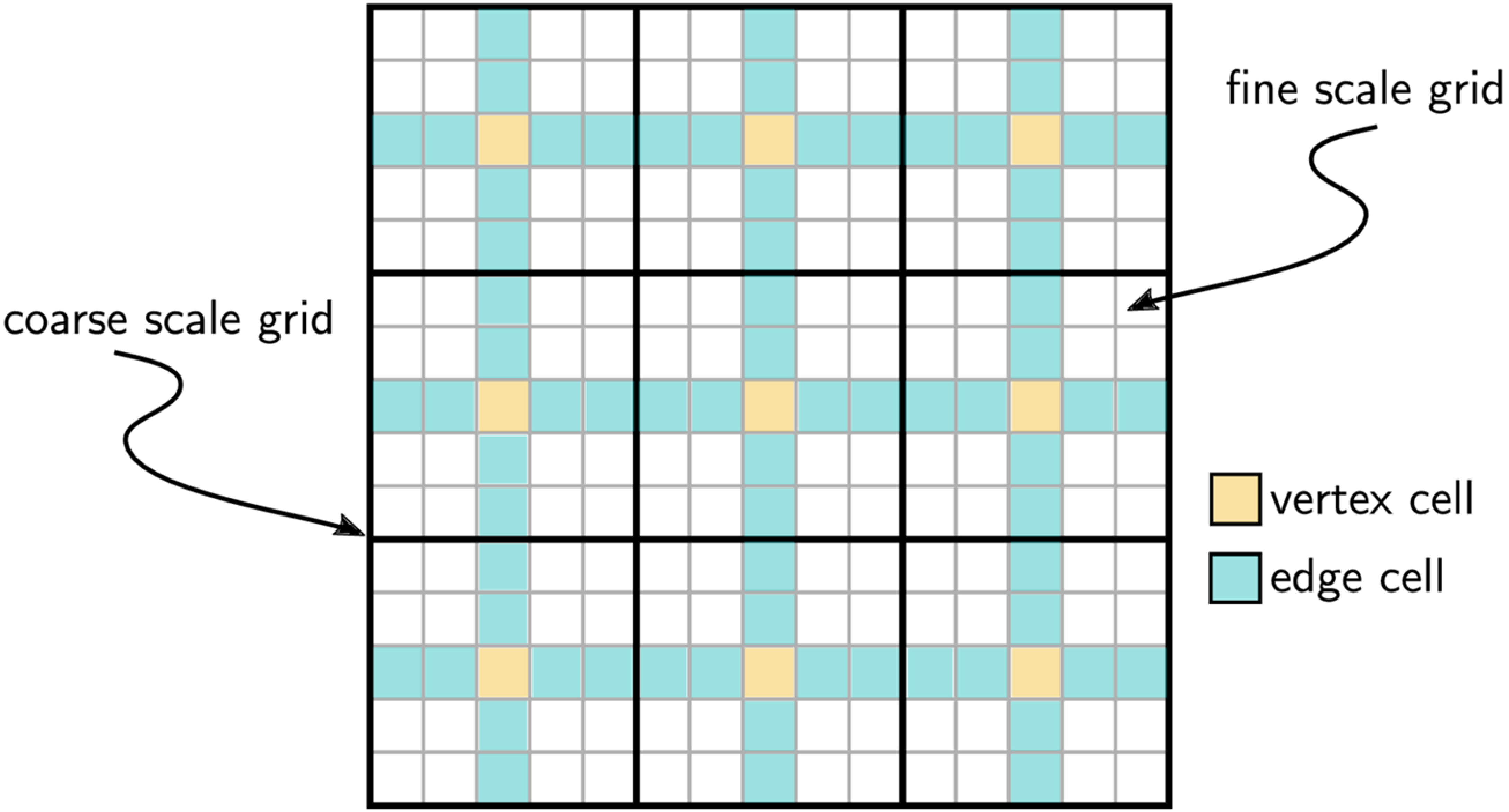
Approximate fine-scale solution

MSFV
Conservative Fine-Scale Velocity

MSFV flow dual coarse grid



MSFV flow dual coarse grid



MSFV flow dual basis functions

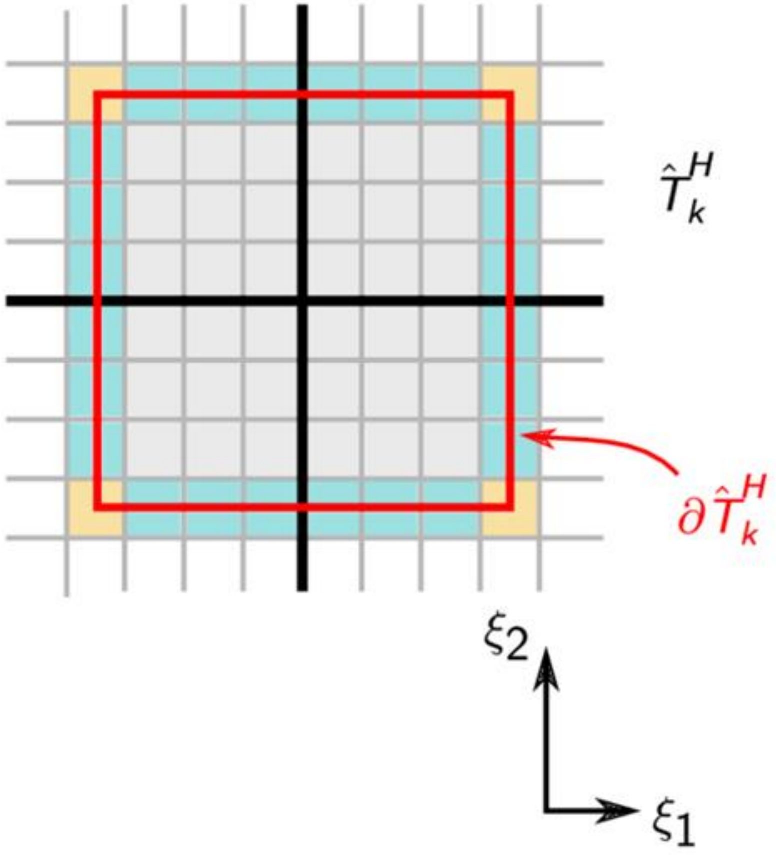
- Computed solving local problems (e.g. mass balance) on each dual coarse element
- Solutions of reduced-dimension problems on boundaries used as boundary conditions (localization assumption)

Given $\mathcal{S}^h = \text{span}\left\{N_j^h(\Omega), j = 1, \ldots, n^h\right\}$, find $N_i^H : \mathcal{S}^h \to \mathbb{R}$ such that:

$$\nabla \cdot \left(\boldsymbol{\kappa}(\boldsymbol{x})\nabla N_i^H\right) = 0 \qquad \text{in } \hat{T}_k$$

$$\nabla_{\parallel} \cdot \left(\boldsymbol{\kappa}(\boldsymbol{x})\nabla N_i^H\right)_{\parallel} = 0 \qquad \text{on } \partial\hat{T}_k$$

$$N_i^H(\boldsymbol{\xi}_j^V) = \delta_{ij} \qquad \forall j \in \left\{1, \ldots, n^H\right\}$$



MSFV conservative velocity reconstruction

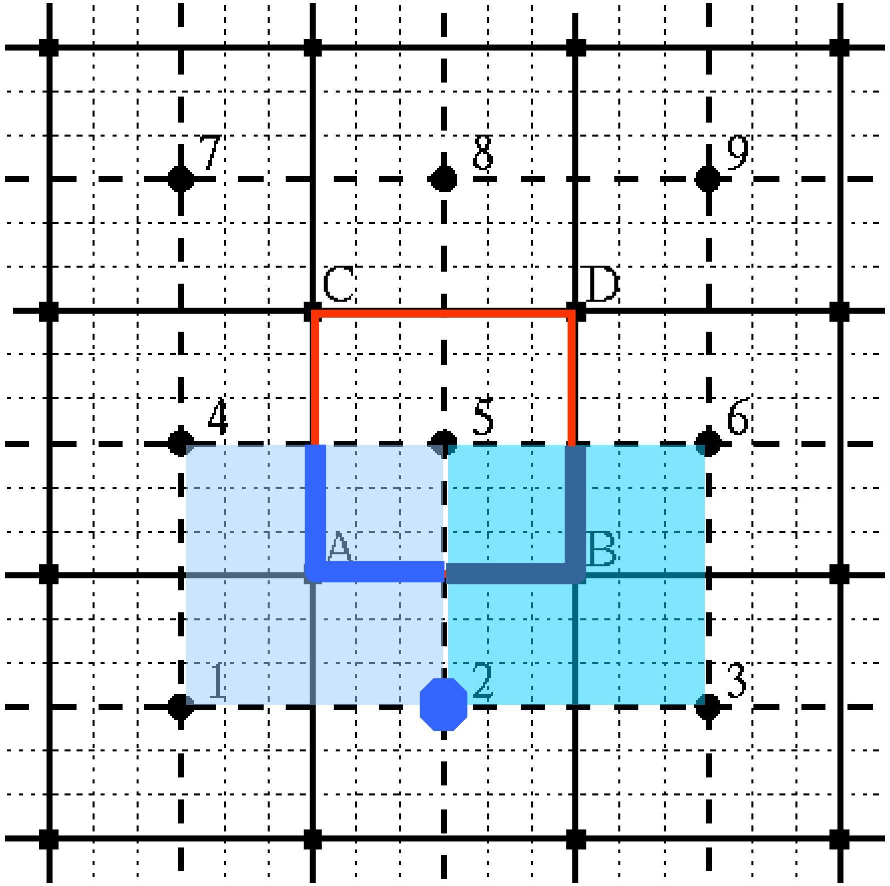
Dual basis functions are used to:

- Compute coarse-scale transmissibility coefficients
- Interpolate coarse-scale pressure solution

MSFV employs a second set of basis functions (primal):

- Constructed by solving local Neumann problems using fluxes obtained from interpolated pressure field as boundary conditions
- Used to reconstruct a locally conservative fine-scale velocity field (important for transport problems)

Conservative Fine-Scale Velocity Field



Superposition in Primal CV 5:

$$p = \sum_{j=1}^{9} p^j \Phi_5^j$$

Multiscale finite volume (MSFV)

- Multiscale finite volume method: Jenny, Lee, Tchelepi (2003)
- Extensions and related work
  - ▷ correction functions to handle non-elliptic features (Lee et al. 2008, Lunati & Jenny, 2008)
  - ▷ extension to compressible flow (Zhou & Tchelepi, 2008)
  - ▷ handling of wells (Wolfsteiner et al., 2006; Wang, 2015)
  - ▷ adaptive updating of basis functions (and transport equations)
  - ▷ iterative formulation with smoothers (Hajibeygi, 2008; Zhou & Tchelepi, 2012)
  - ▷ algebraic formulation (Zhou & Tchelepi, 2008; Wang et al. 2014)
  - ▷ ... and many more!

TAMS
Two-Stage Algebraic Multiscale Solver

Two-stage multiscale preconditioner

- Multiscale *global* operator: $\boxed{\mathcal{M}_G^{-1} = \mathcal{P}\left(\mathcal{R}A_h\mathcal{P}\right)^{-1}\mathcal{R}}$
- $\mathcal{M}_G^{-1}$ is rank-deficient by $n^h - n^H$!
- Complement with a *local* smoother $\mathcal{M}_L^{-1}$ (Jacobi, Gauss-Seidel, ILU, ...)
- Multiplicative two-stage preconditioner: $\mathcal{M}_{\mathrm{TAMS}}^{-1} : \boldsymbol{v} \longmapsto \boldsymbol{z}$

$$\boxed{\begin{aligned} \boldsymbol{z}_1 &= \mathcal{M}_G^{-1}\boldsymbol{v} && \textbf{(stage 1)} \\ \boldsymbol{z} &= \boldsymbol{z}_1 + \mathcal{M}_L^{-1}\left(\boldsymbol{v} - A_h\boldsymbol{z}_1\right) && \textbf{(stage 2)} \end{aligned}}$$

$$\boxed{\mathcal{M}_{\mathrm{TAMS}}^{-1} = \mathcal{M}_G^{-1} + \mathcal{M}_L^{-1}\left(I - A_h\mathcal{M}_G^{-1}\right)}$$

Two-stage multiscale preconditioner: scalability

Y. Wang et al. / Journal of Computational Physics 259 (2014) 284–303 301

Table 3
CPU time (sec) and iteration steps for SAMG on a patchy domain.

Problem size	32 × 32 × 32	64 × 64 × 64	128 × 128 × 128	256 × 256 × 256
Setup phase	0.12	1.14	12.33	157.41
Solution phase	0.14	0.76	6.33	73.02
Total	0.26	1.90	18.66	230.43
GMRES iterations	6	7	9	10

Table 4
CPU time (sec) and iteration steps for AMS on a patchy domain.

Problem size	32 × 32 × 32	64 × 64 × 64	128 × 128 × 128	256 × 256 × 256
Setup phase	0.29	2.12	17.00	147.96
Solution phase	0.10	0.87	8.01	55.91
Total	0.39	2.99	25.01	203.87
GMRES iterations	22	21	21	18

Two-stage multiscale preconditioner: parallel performance

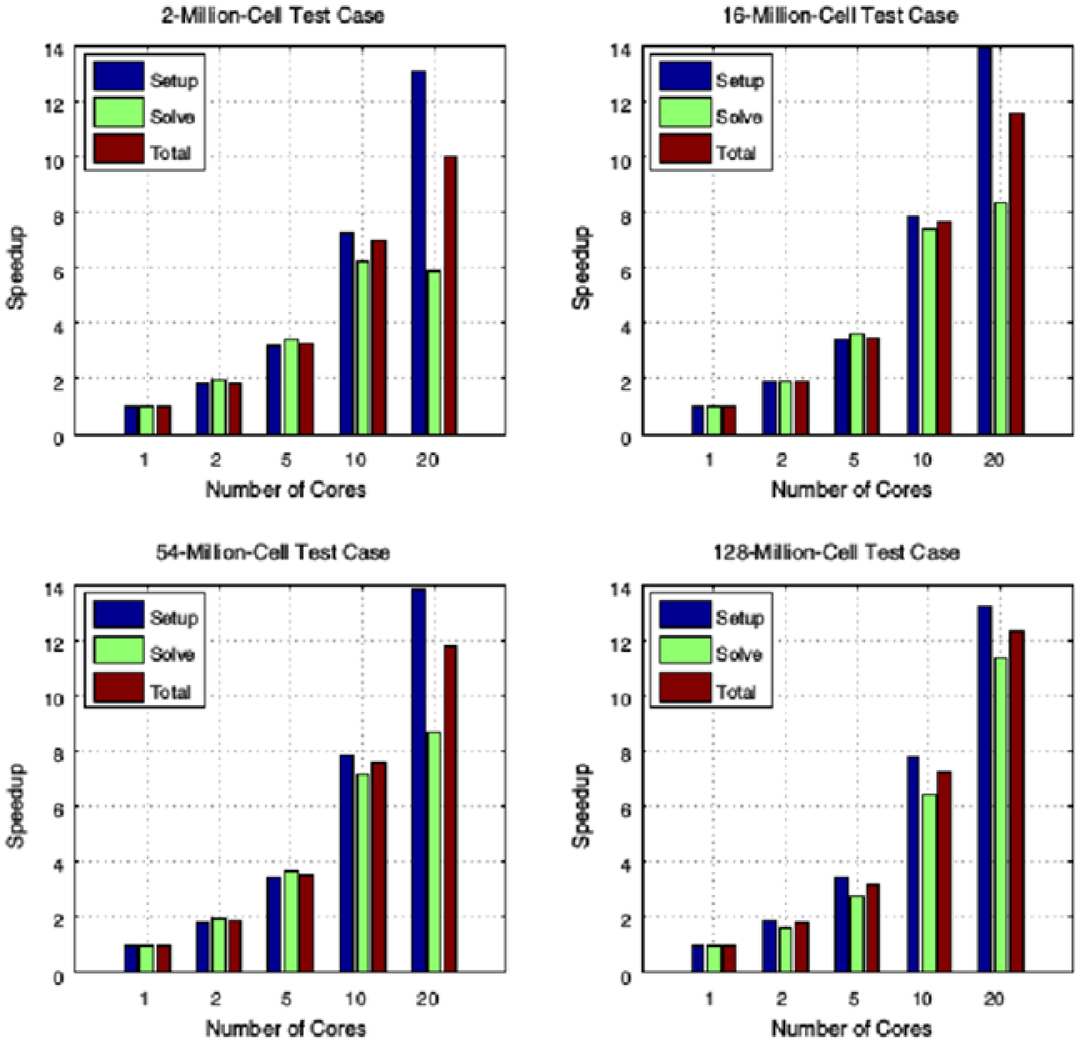


Fig. 11—Speedup of setup and solve kernels of AMS for several problem sizes running on two sockets of Intel Xeon Processor E5-2690-v2.

[Manea, Sewall, Tchelepi, 2016]

Multiscale poromechanics

Incompressible flow and geomechanics

Two-field fomulation: find the displacement vector, $\boldsymbol{u}$, and pressure, p, such that:

$$\nabla \cdot \underbrace{(\mathbf{C}_{dr} : \nabla^s \boldsymbol{u} - bp\mathbf{1})}_{\text{Cauchy stress tensor}} + \boldsymbol{f} = \mathbf{0} \qquad \text{(linear momentum balance)}$$

$$\underbrace{\frac{\partial}{\partial t}(b\nabla \cdot \boldsymbol{u})}_{\text{fluid volume change}} - \nabla \cdot \underbrace{\left(\frac{\boldsymbol{\kappa}}{\mu}\nabla p\right)}_{\text{Darcy's law}} = q \qquad \text{(mass balance)}$$

Discretization

- Linear nodal FE for $\boldsymbol{u}$
- Cell-centered FV for p
- Backward Euler time integration

Matrix form

$$\begin{bmatrix} K & B_1 \\ B_2 & C \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}^h_{n+1} \\ \boldsymbol{p}^h_{n+1} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f}_{n+1} \\ \boldsymbol{g}_{n+1} \end{Bmatrix}$$

MultiScale Mechanics

Mechanics multiscale basis functions

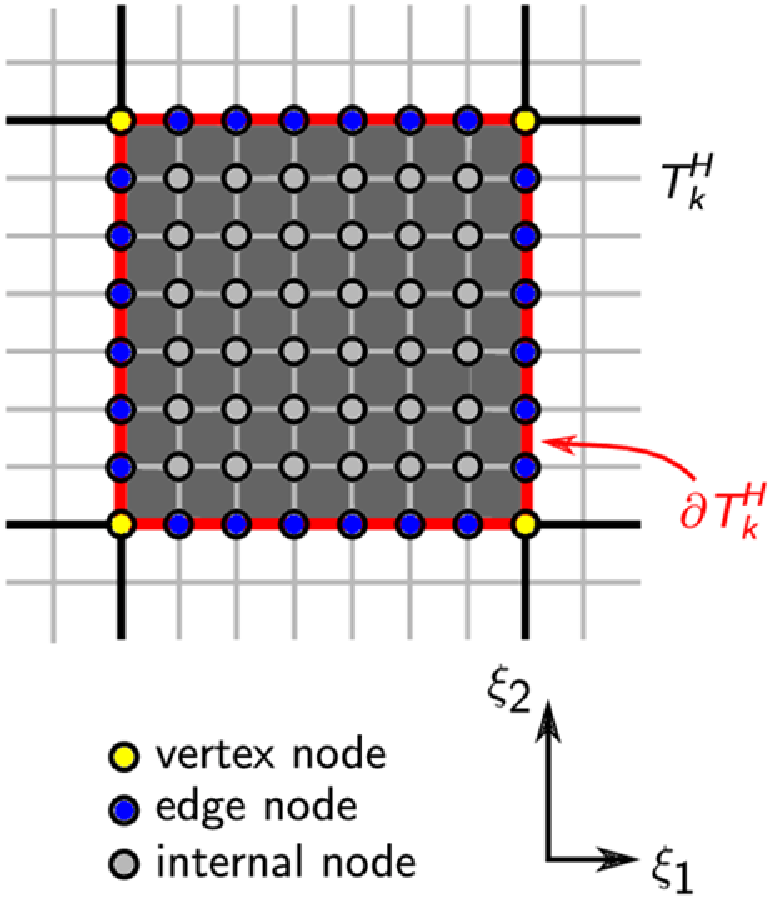
Computed solving local linear-momentum balance assuming drained conditions

Given $\mathcal{S}_{\boldsymbol{u}}^h = \text{span}\left\{\boldsymbol{N}_{\boldsymbol{u},j}^h(\Omega), j = 1, \dots, n_{\boldsymbol{u}}^h\right\}$, find $\boldsymbol{N}_{\boldsymbol{u},i}^H : \mathcal{S}_{\boldsymbol{u}}^h \to \mathbb{R}^2$ such that:

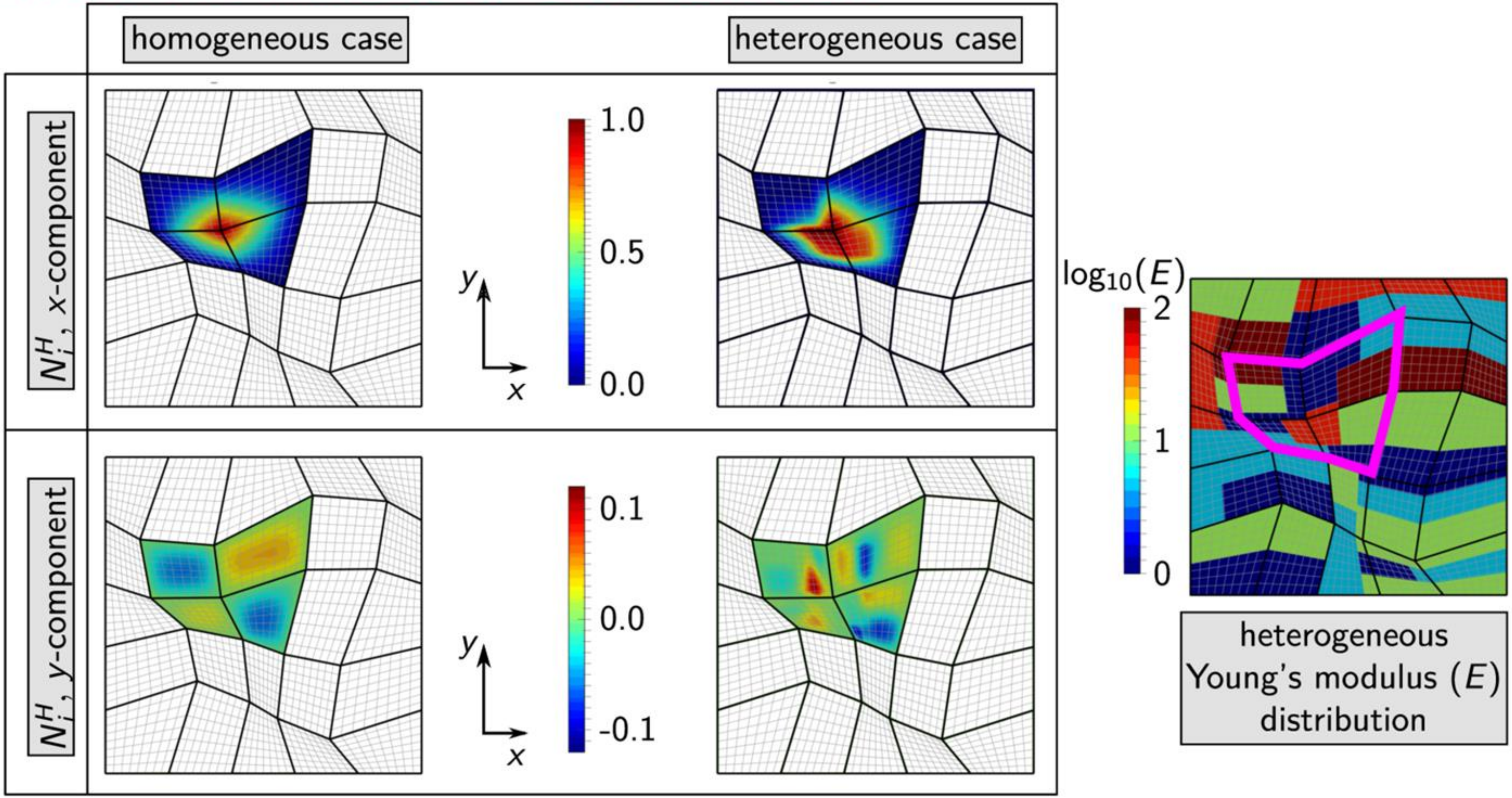
$$\nabla \cdot \left(\mathbf{C}_{dr} : \nabla^s \boldsymbol{N}_{\boldsymbol{u},i}^H\right) = \mathbf{0} \qquad \text{in } T_k^H$$

$$\nabla_{\parallel} \cdot \left(\mathbf{C}_{dr} : \nabla_{\parallel}^s \boldsymbol{N}_{\boldsymbol{u},i}^H\right) = \mathbf{0} \qquad \text{on } \partial T_k^H$$

$$\boldsymbol{N}_{\boldsymbol{u},i}^H(\boldsymbol{\xi}_{\boldsymbol{u},j}^V) = \delta_{ij}\, \boldsymbol{e} \quad \forall j \in \left\{1, \dots, n_{\boldsymbol{u}}^H\right\}$$



Mechanics multiscale basis functions



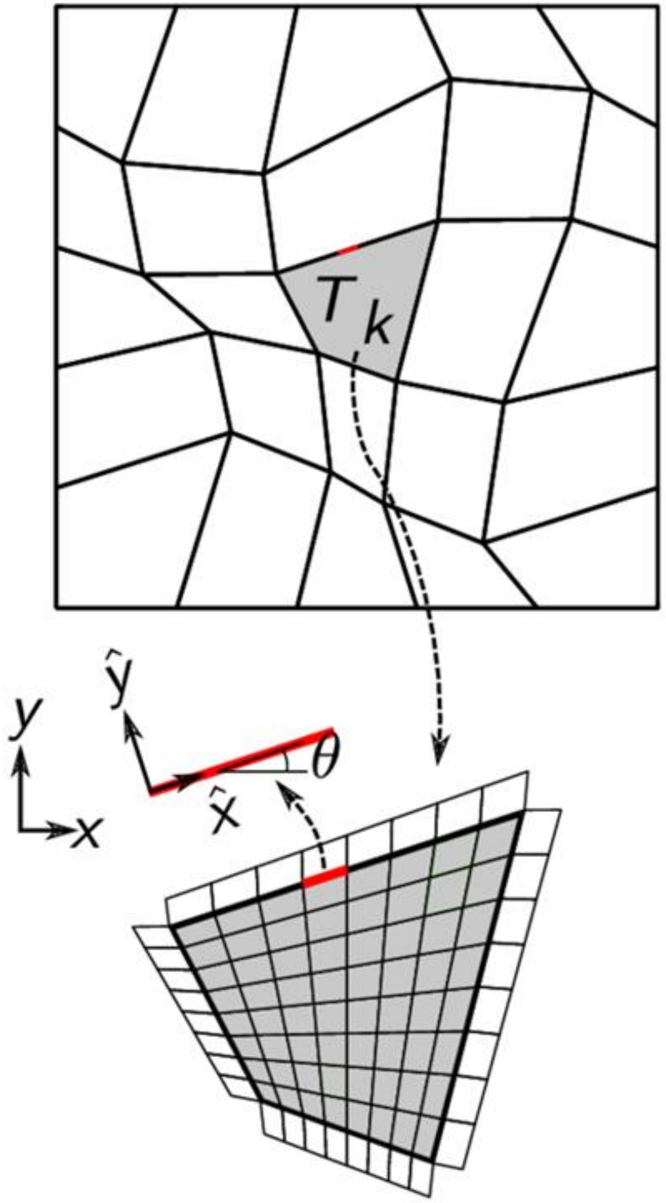
Mechanics reduced boundary problems

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

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

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

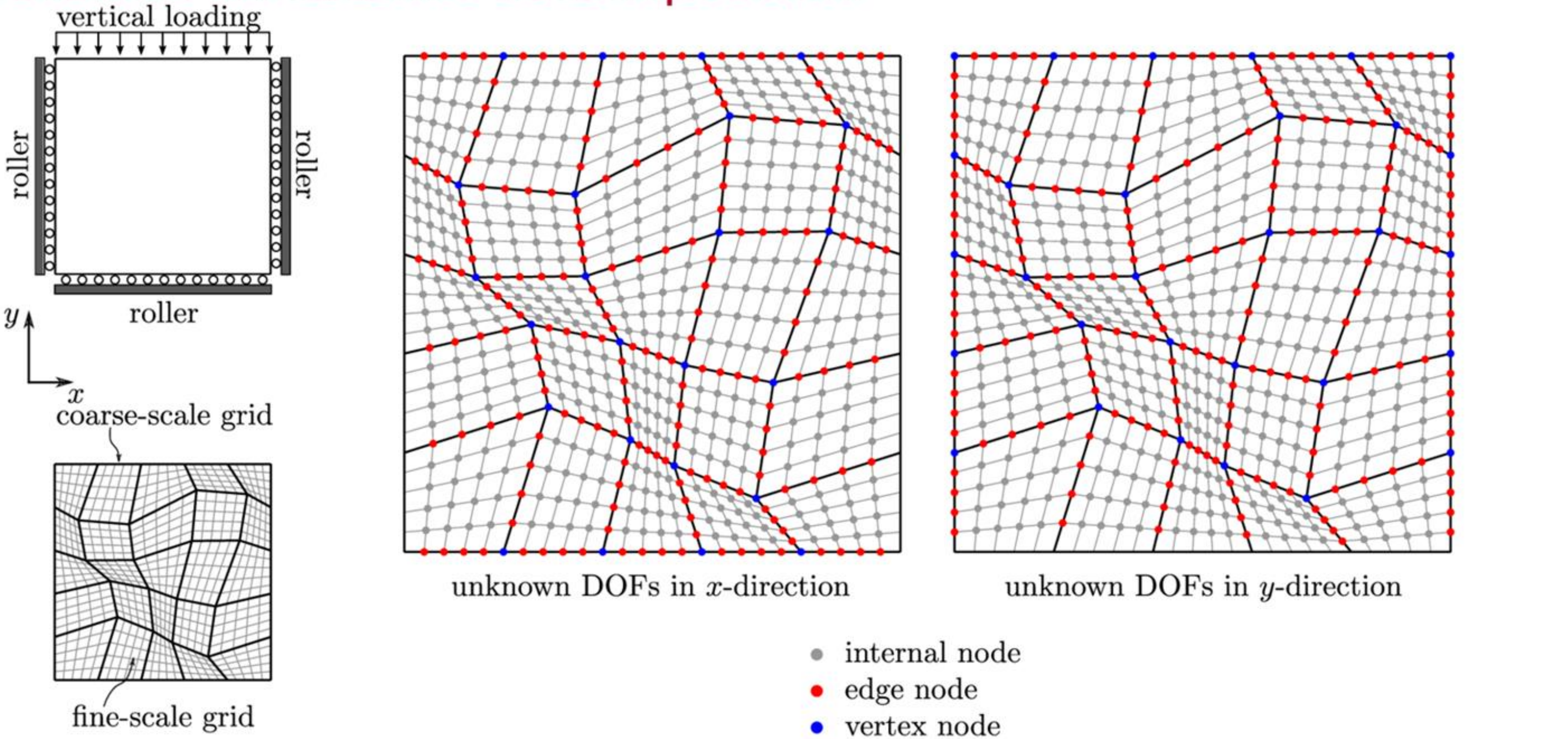
$$\begin{cases} \dfrac{\partial}{\partial \hat{x}} \left(\hat{K}_v \dfrac{\partial \hat{N}^H_{i_{\hat{x}}}}{\partial \hat{x}} \right) = 0 & \text{(axial equilibrium)} \\[2ex] \dfrac{\partial}{\partial \hat{x}} \left(\hat{G} \dfrac{\partial \hat{N}^H_{i_{\hat{y}}}}{\partial \hat{x}} \right) = 0 & \text{(transverse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



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

[Castelletto, Hajibeygi, Tchelepi. JCP (2017)]

Mechanics wirebasket decomposition



[Castelletto, Hajibeygi, Tchelepi. JCP (2017)]

Mechanics algebraic construction

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

$$\hat{K}_h = W^T K W = \begin{bmatrix} \hat{K}_{II} & \hat{K}_{IE} & \hat{K}_{IV} \\ \hat{K}_{EI} & \hat{K}_{EE} & \hat{K}_{EV} \\ \hat{K}_{VI} & \hat{K}_{VE} & \hat{K}_{VV} \end{bmatrix}$$

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

Mechanics algebraic construction

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.,

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

Algebraic MultiScale Solvers: Unstructured Grid

"Algebraic" multiscale

How to construct the coarse grid and supports only from the system matrix A?

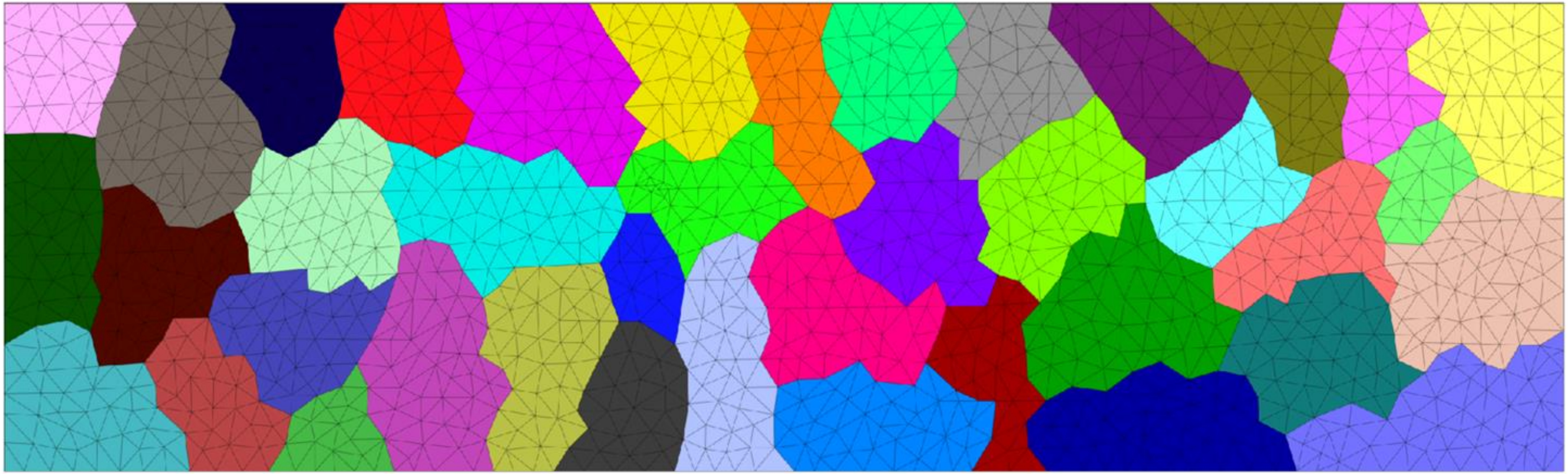
- Difficult to construct "wirebasket" in general case
- Just do AMG?

Add minimal grid information: element-to-node adjacency (topology)

- Available in any FEM code
- Also available "for free" from a coupled poroelastic system
- Keep this representation at the coarse level to allow >2 levels

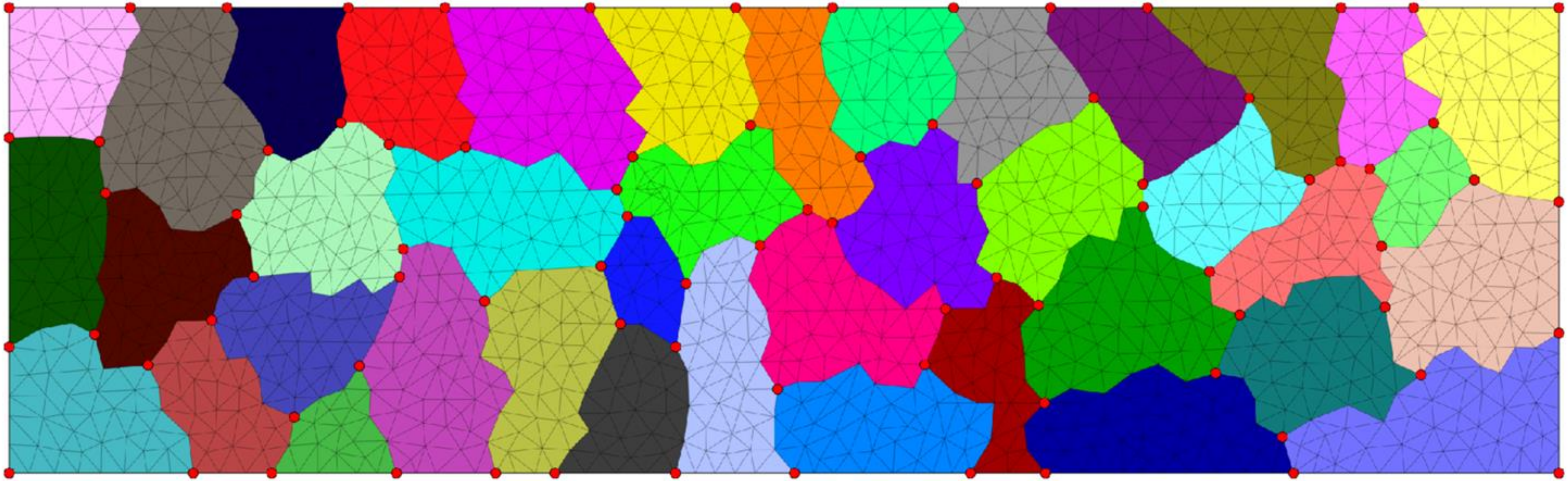
Mechanics - coarse grid

Step 1: generate the coarse grid from partition

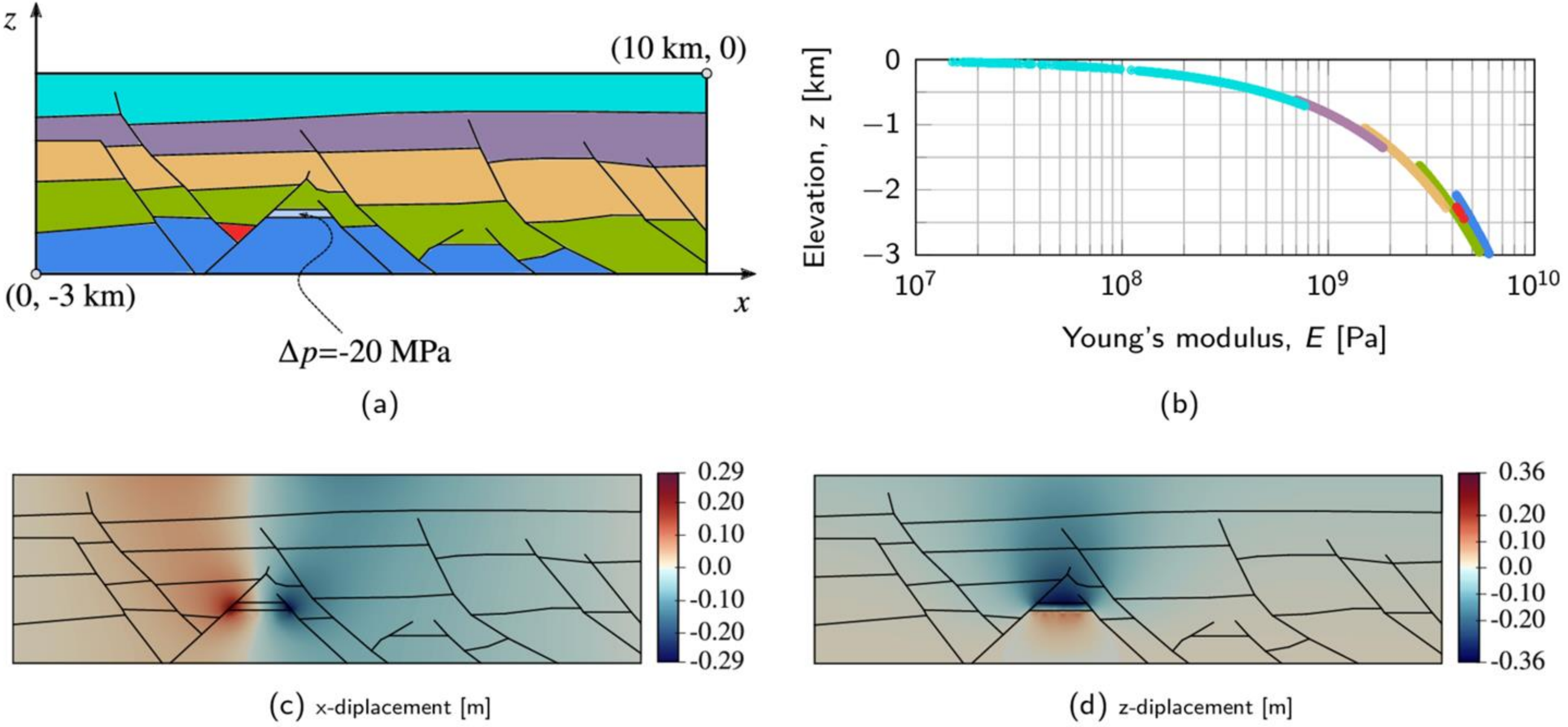


Mechanics - coarse grid

Step 1: generate the coarse grid from partition



Mechanics - 2D cross-section example



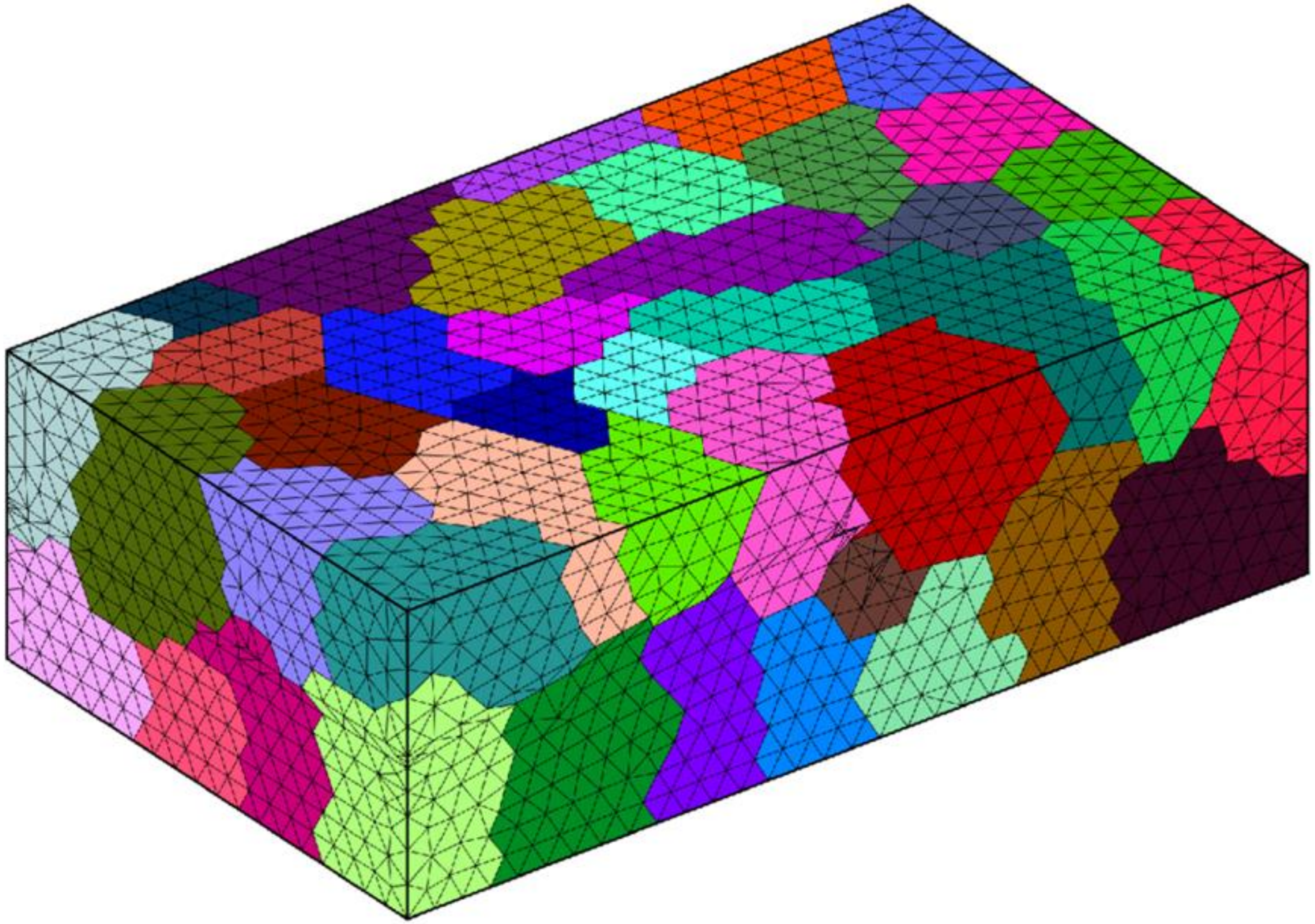
Mechanics - 2D cross-section example

ℓ	fine grid			coarse grid			coarsening ratio	
	# cell	# node	# dof	# cell	# node	# dof	cell	dof
1	23,390	11,947	23,894	64	128	256	365.5	93.3
2	46,932	23,817	47,634	128	256	512	366.7	93.0
3	93,129	47,085	94,170	256	512	1,024	363.8	92.0
4	186,940	94,165	188,330	512	1,021	2,042	365.1	92.2
5	372,359	187,210	374,420	1,024	2,038	4,076	363.6	91.9
6	745,904	374,346	748,692	2,048	4,078	8,156	364.2	91.8
7	1,490,288	747,199	1,494,398	4,096	8,162	16,324	363.8	91.5

ℓ	IC(0)		Gauss-Seidel		Sym. Gauss-Seidel		ℓ_1-Jacobi ($\times 2$)	
	MsRSB	no MS	**MsRSB**	no MS	**MsRSB**	no MS	**MsRSB**	no MS
1	**27**	137	**54**	290	**39**	204	**69**	369
2	**27**	194	**55**	410	**39**	290	**70**	517
3	**28**	268	**57**	576	**40**	407	**72**	736
4	**26**	385	**54**	818	**38**	570	**68**	1046
5	**27**	544	**57**	1155	**40**	816	**72**	1478
6	**27**	769	**56**	1633	**40**	1153	**71**	—
7	**27**	1085	**56**	—	**40**	1629	**72**	—

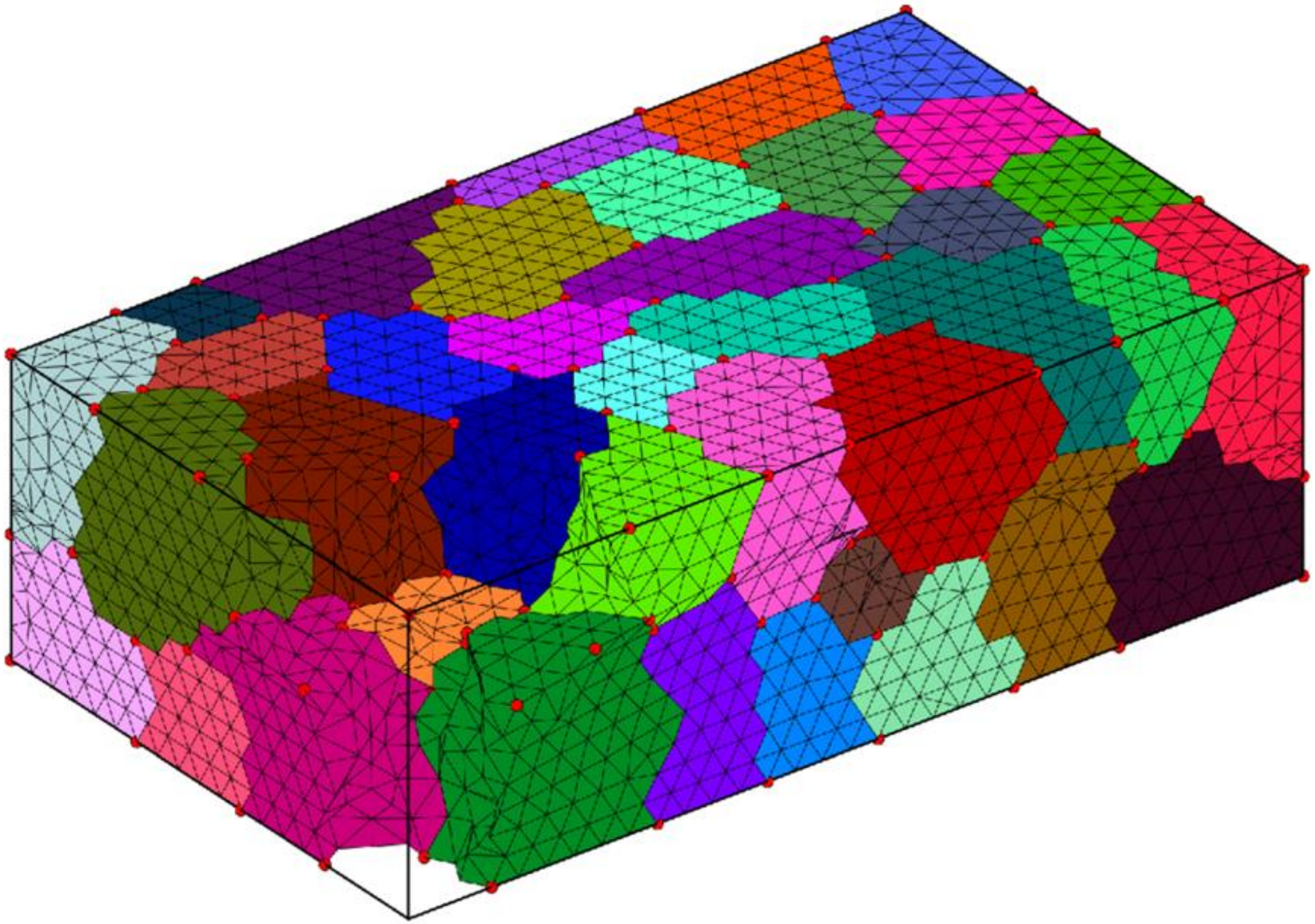
Mechanics - coarse grid

Step 1: generate the coarse grid from partition



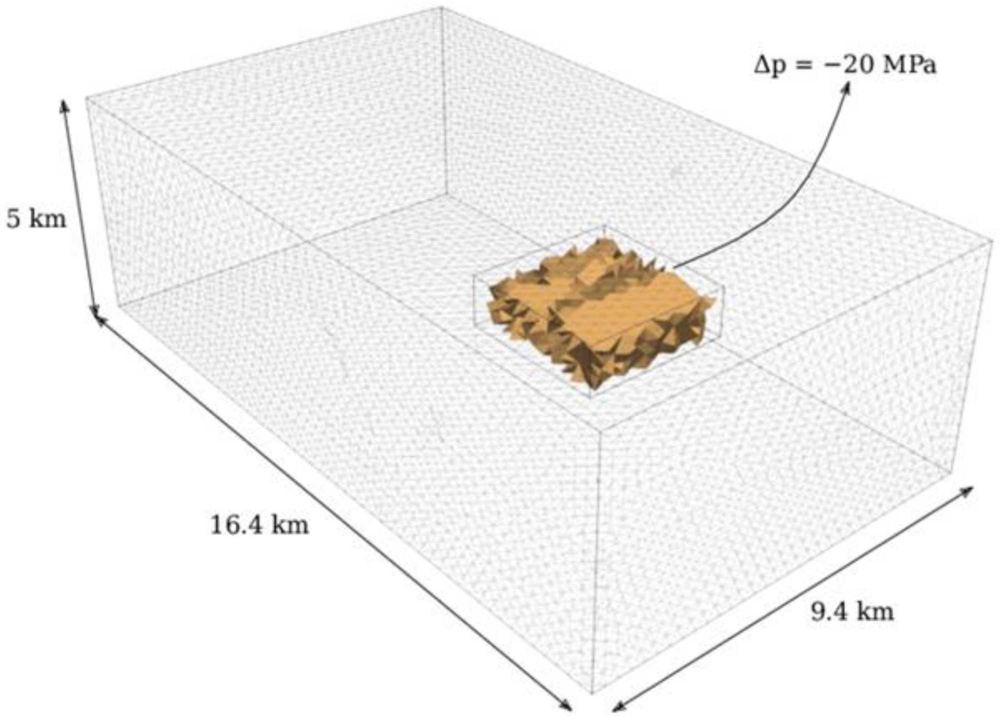
Mechanics - coarse grid

Step 1: generate the coarse grid from partition

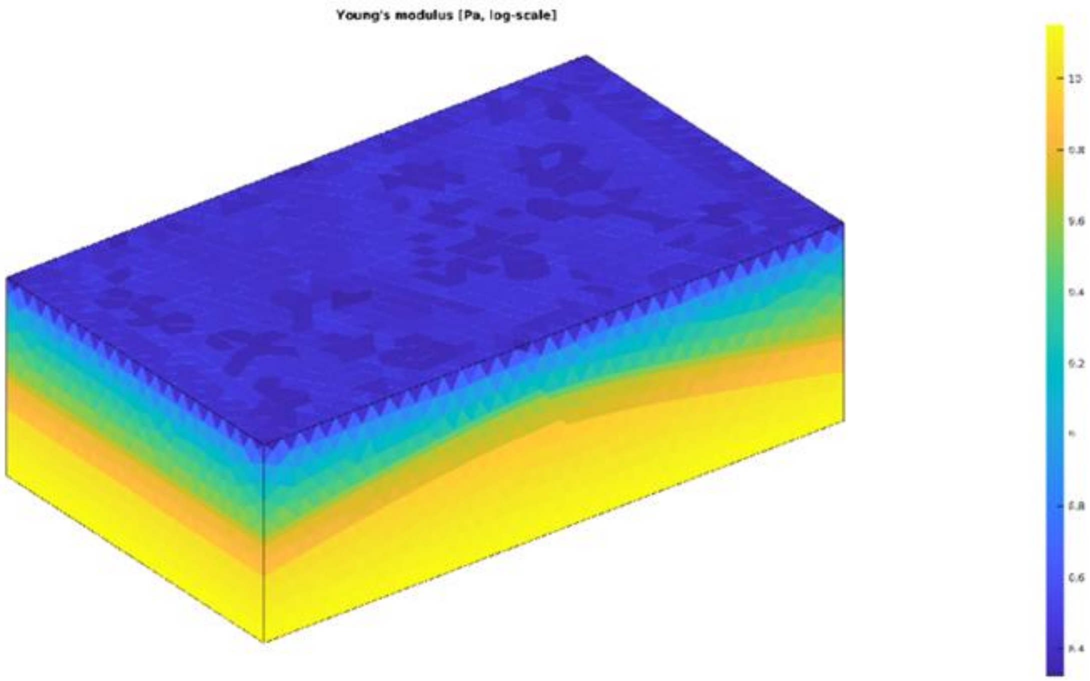


3D unstructured mesh - elasticity

Problem setup:



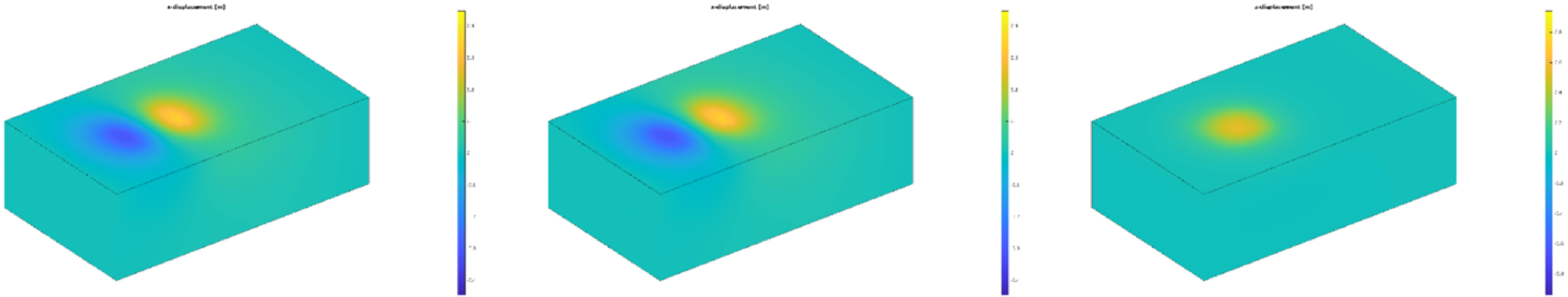
(a) Problem dimensions



(b) Young's modulus [Pa, log-scale]

3D unstructured mesh - elasticity

Solution:



(a) x-diplacement [m] (b) y-diplacement [m] (c) z-diplacement [m]

3D unstructured mesh - elasticity

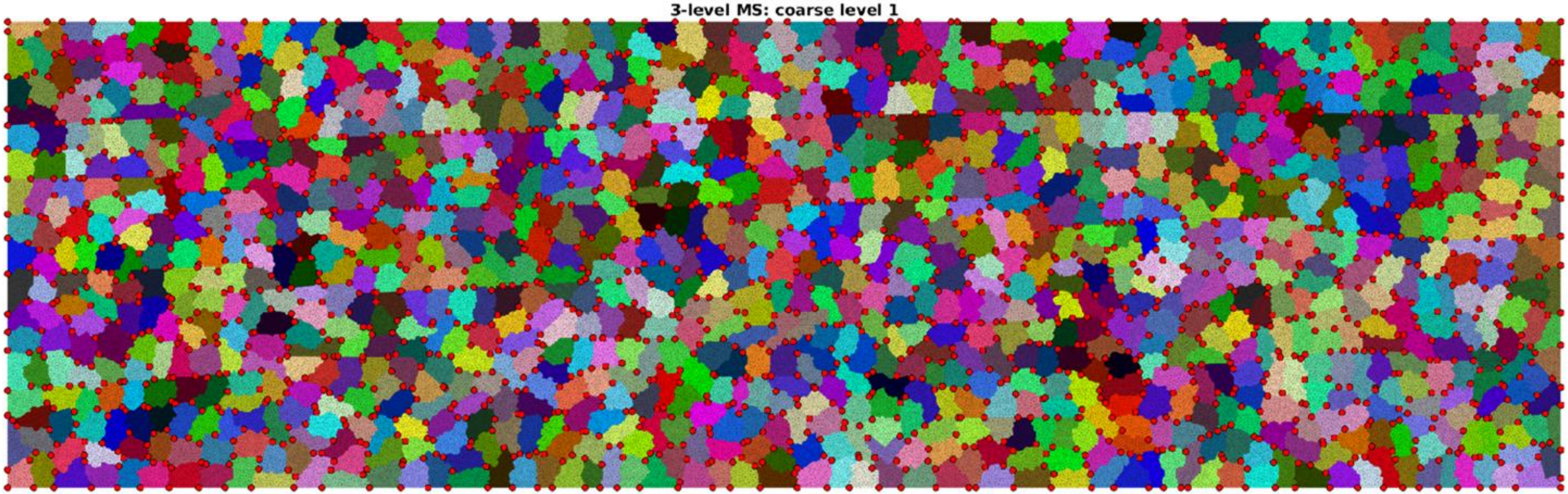
ℓ	fine grid			coarse grid			coarsening ratio	
	# cell	# node	# dof	# cell	# node	# dof	cell	dof
0	9,140	1,956	5,868	12	52	156	761.67	37.62
1	61,656	11,570	34,710	64	321	963	963.38	36.04
2	274,299	48,633	145,899	256	1,344	4,032	1071.48	36.19

ℓ	IC(0)		Sym. Gauss-Seidel		ℓ_1-Jacobi ($\times 2$)	
	MsRSB	no MS	**MsRSB**	no MS	**MsRSB**	no MS
1	**55**	73	**31**	49	**52**	90
2	**48**	101	**25**	60	**43**	132
3	**47**	159	**22**	92	**39**	206

Multi-Level Multi-Scale Poro-Mechanics

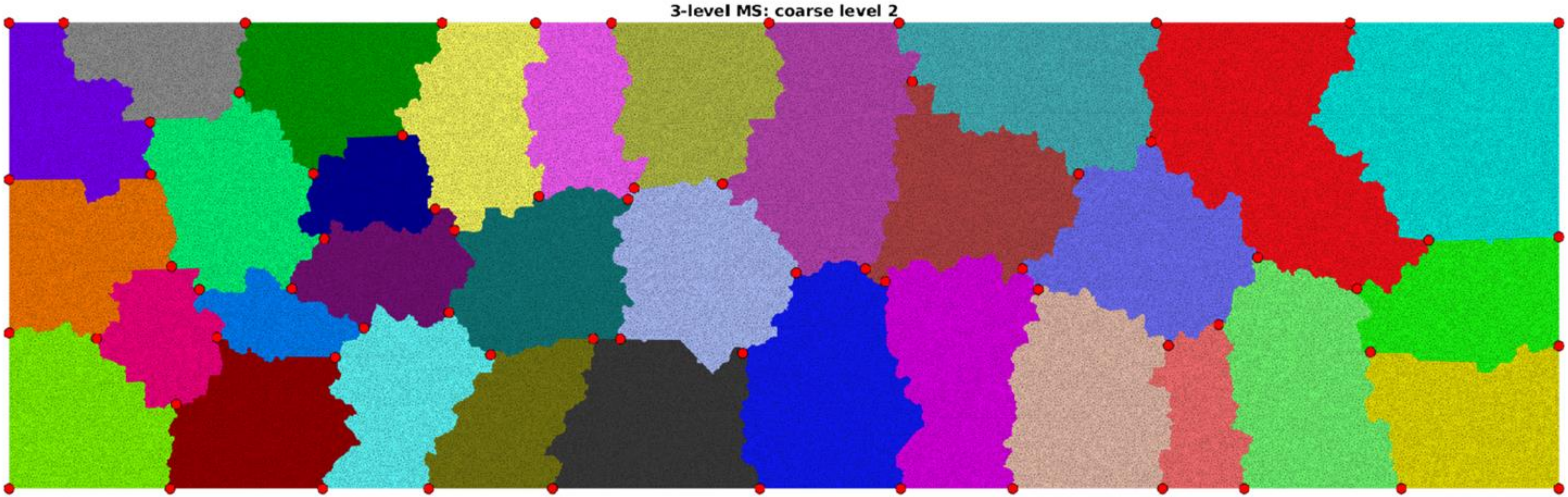
Mechanics - 3-level multiscale

Since mesh at every level is represented only via element-node adjacency, can apply idea recursively at coarse levels:



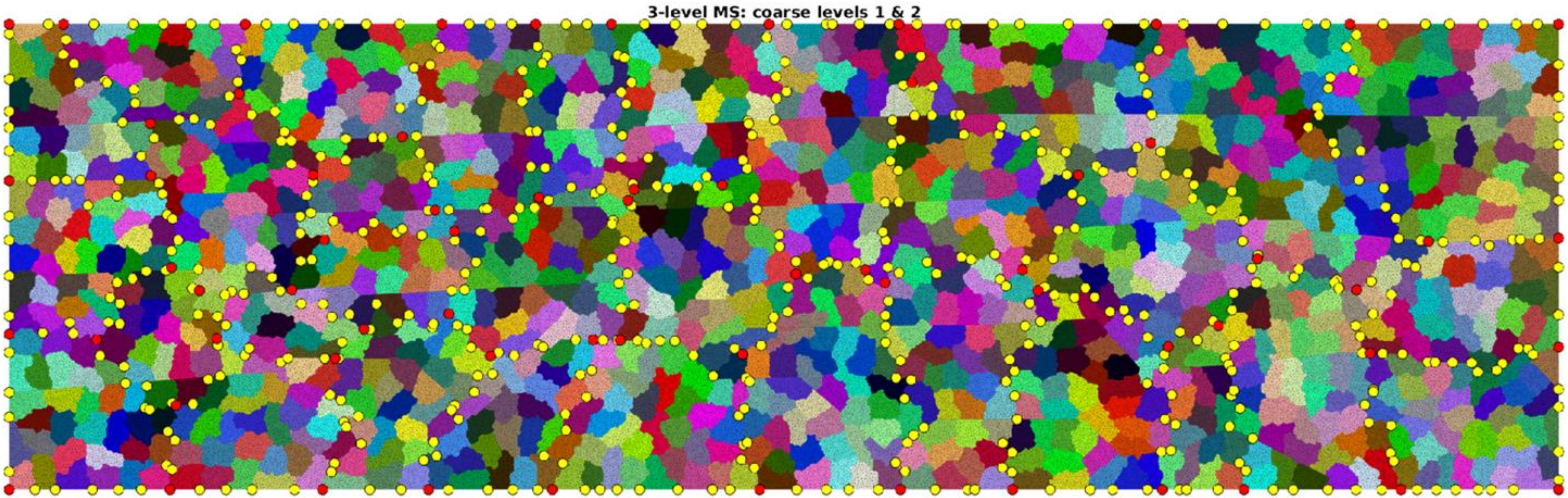
Mechanics - 3-level multiscale

Since mesh at every level is represented only via element-node adjacency, can apply idea recursively at coarse levels:



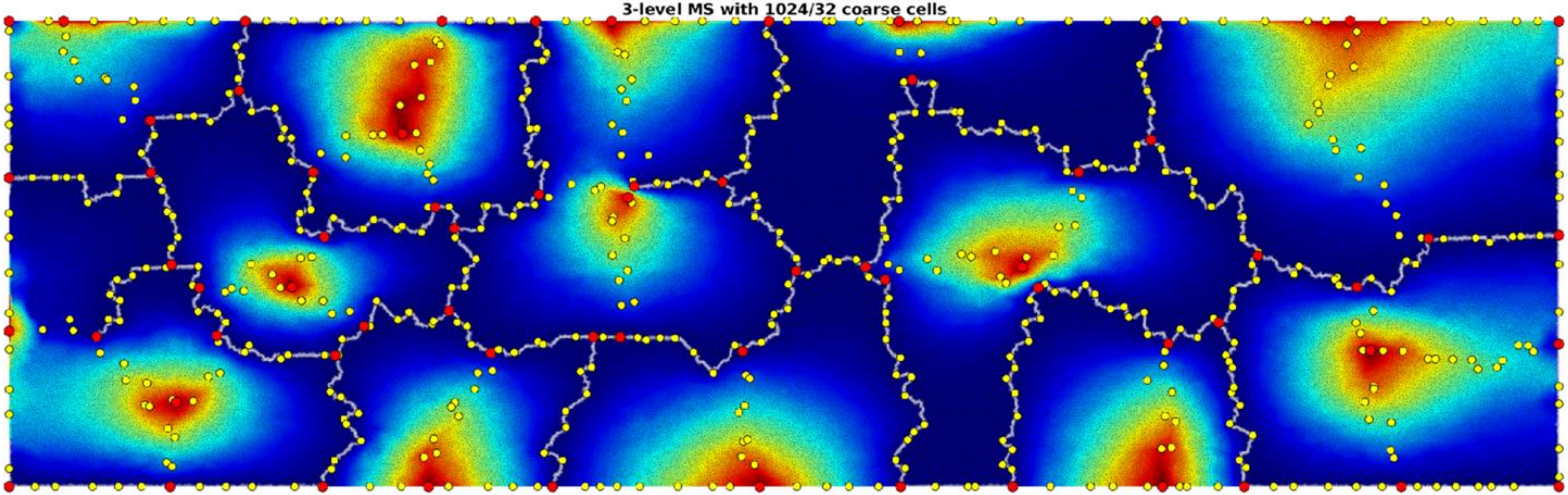
Mechanics - 3-level multiscale

Since mesh at every level is represented only via element-node adjacency, can apply idea recursively at coarse levels:



Mechanics - 3-level multiscale

Since mesh at every level is represented only via element-node adjacency, can apply idea recursively at coarse levels:



Mechanics - 2D cross-section example - 3-level multiscale

ℓ	fine grid		L1 coarse grid		L1 ratio		L2 coarse grid		L2 ratio	
	cell	dof	cell	dof	cell	dof	cell	dof	cell	dof
2	46,932	47,634	128	512	366.66	93.04	4	16	32	32.00
3	93,129	94,170	256	1,024	363.79	91.96	8	36	32	28.44
4	186,940	188,330	512	2,042	365.12	92.23	16	68	32	30.03
5	372,359	374,420	1,024	4,076	363.63	91.86	32	132	32	30.88
6	745,904	748,692	2,048	8,156	364.21	91.80	64	260	32	31.37
7	1,490,288	1,494,398	4,096	16,324	363.84	91.55	128	516	32	31.64

Table: 3-level vs 2-level multiscale, CG iterations

ℓ	MsRSB + IC(0)		MsRSB + SGS	
	3-level	2-level	3-level	2-level
2	27	27	40	39
3	28	28	41	40
4	26	26	38	38
5	27	27	40	40
6	27	27	40	40
7	27	27	40	40

Poroelasticity - preconditioner design

- Form and solve coarse-scale system:

$$\boldsymbol{x} \approx \underbrace{\begin{Bmatrix} \boldsymbol{u}^h_{MS} \\ \boldsymbol{p}^h_{MS} \end{Bmatrix}}_{\boldsymbol{x}_{MS}} = \underbrace{\begin{bmatrix} \mathcal{P}^{(u,u)} & \\ & \mathcal{P}^{(p,p)} \end{bmatrix}}_{\mathcal{P}} \underbrace{\begin{Bmatrix} \boldsymbol{u}^H \\ \boldsymbol{p}^H \end{Bmatrix}}_{\boldsymbol{x}^H} \quad \Rightarrow \quad \underbrace{\mathcal{R} A^h \mathcal{P}}_{A^H} \boldsymbol{x}^H = \underbrace{\mathcal{R} \boldsymbol{b}^h}_{\boldsymbol{b}^H}$$

- The coarse-scale system is fully coupled:

$$\begin{bmatrix} \mathcal{R}^{(u,u)} & \\ & \mathcal{R}^{(p,p)} \end{bmatrix} \begin{bmatrix} K^h & B_1^h \\ B_2^h & C^h \end{bmatrix} \begin{bmatrix} \mathcal{P}^{(u,u)} & \\ & \mathcal{P}^{(p,p)} \end{bmatrix} = \begin{bmatrix} K^H & B_1^H \\ B_2^H & C^H \end{bmatrix}$$

- Can use different coarse grids and/or resolutions for the two problems

[Castelletto, Klevtsov, Hajibeygi, Tchelepi. *CompGeo* (2019)]

Poroelasticity - preconditioner design

- Multiplicative preconditioner: $\mathcal{M}_{\text{mult}}^{-1} : \boldsymbol{v} \longmapsto \boldsymbol{z}$

$$\begin{aligned} \boldsymbol{z}_1 &= \mathcal{M}_{MS}^{-1}\boldsymbol{v} && \textbf{(stage 1)} \\ \boldsymbol{z} &= \boldsymbol{z}_1 + \mathcal{M}_L^{-1}(\boldsymbol{v} - A\boldsymbol{z}_1) && \textbf{(stage 2)} \end{aligned}$$

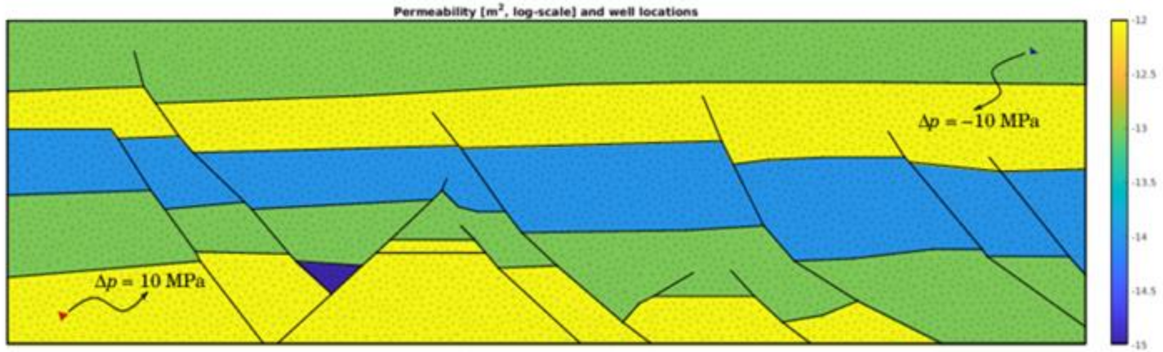
- **Multiscale** operator: $\mathcal{M}_G^{-1} = \mathcal{P}\left(\mathcal{R}A\mathcal{P}\right)^{-1}\mathcal{R}$
- **Post-smoothing** via block-triangular operator with *fixed-stress* split:

$$\mathcal{M}_L^{-1} = \begin{bmatrix} \tilde{K} & -B_1 \\ & \tilde{S} \end{bmatrix}^{-1} \qquad S \approx C - B_2 K^{-1} B_1$$

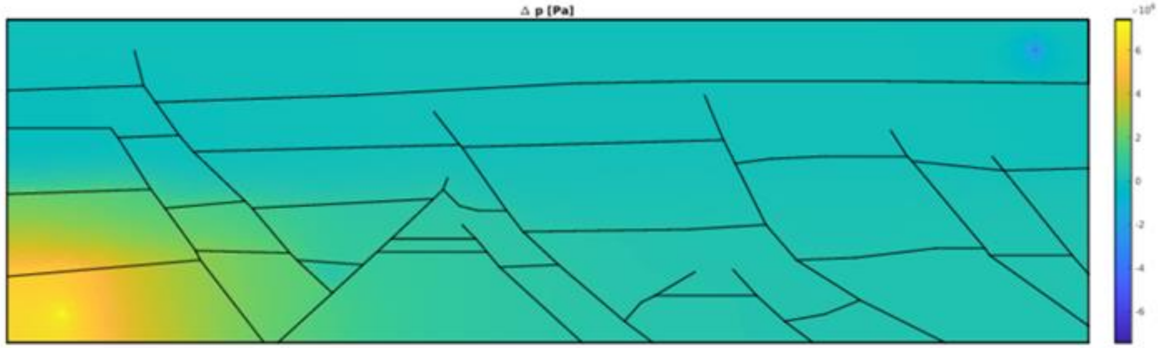
- Schur complement constructed algebraically via *probing*

[Castelletto, Klevtsov, Hajibeygi, Tchelepi. *CompGeo* (2019)]

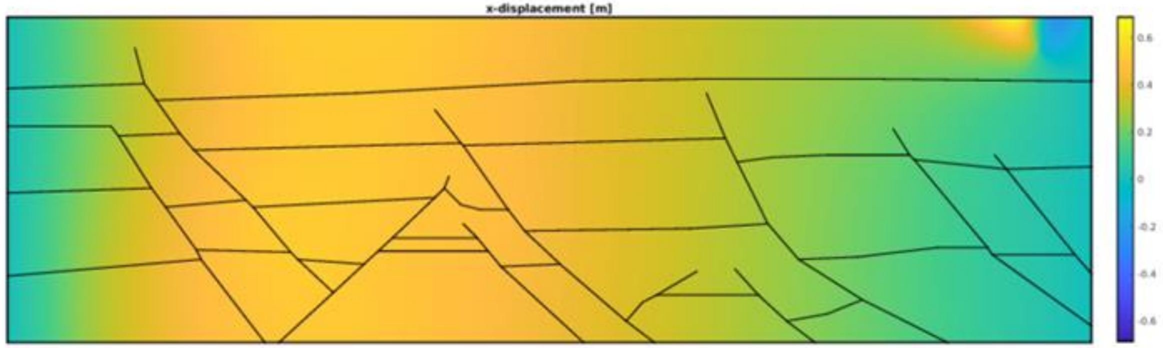
Poroelasticity - 2D cross-section example



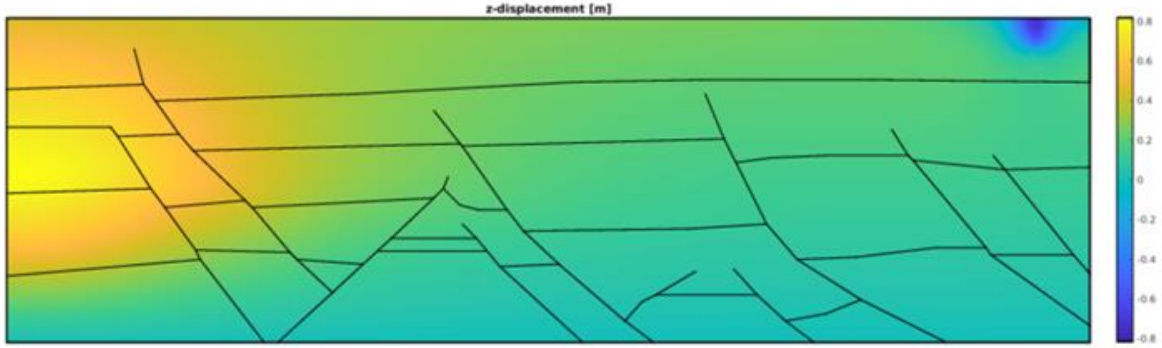
(a) Permeability and wells



(b) Pressure change [Pa]



(c) x-diplacement [m]



(d) z-diplacement [m]

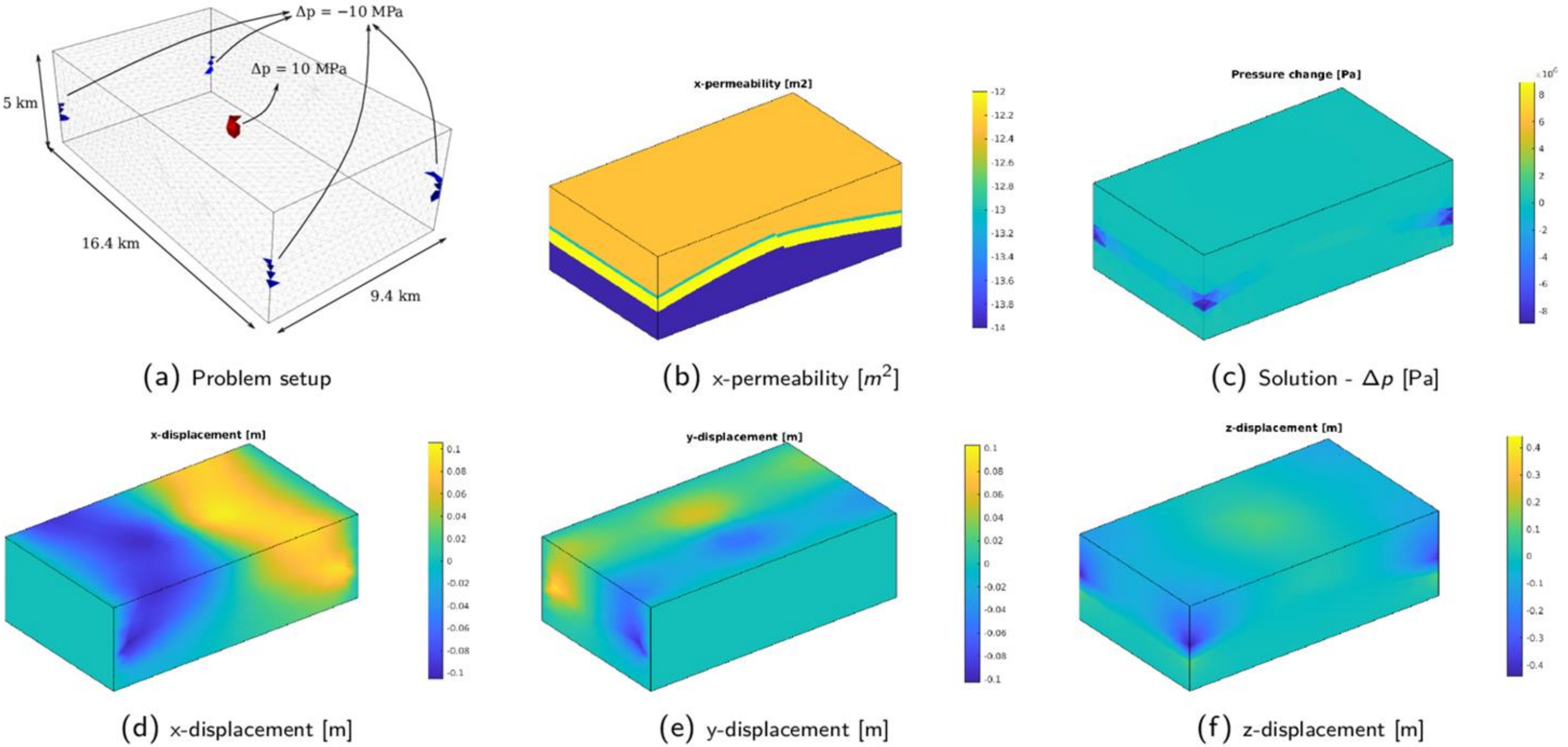
Poroelasticity - 2D cross-section example

ℓ	fine grid			c. grid (M)		c. grid (F)		coarsening ratio		
	# cell	# node	# dof	# cell	# dof	# cell	# dof	mech	flow	total
0	11,879	6,119	24,117	64	256	64	64	47.80	185.61	75.37
1	23,390	11,947	47,284	128	504	128	128	47.41	182.73	74.82
2	46,932	23,817	94,566	256	1,012	256	256	47.07	183.33	74.58
3	93,129	47,085	187,299	512	2,030	512	512	46.39	181.89	73.68
4	186,940	94,165	375,270	1,024	4,066	1,024	1,024	46.32	182.56	73.73
5	372,359	187,210	746,779	2,048	8,132	2,048	2,048	46.04	181.82	73.36

ℓ	IC(0)		Sym. Gauss-Seidel		ℓ_1-Jacobi ($\times 2$)	
	MsRSB	no MS	**MsRSB**	no MS	**MsRSB**	no MS
0	**71**	312	**82**	355	**131**	—
1	**67**	433	**78**	480	**117**	—
2	**66**	500	**78**	—	**115**	—
3	**62**	—	**76**	—	**104**	—
4	**72**	—	**86**	—	**117**	—
5	**67**	—	**82**	—	**109**	—

Note: GMRES iteration limit set to 500

Poroelasticity - 3D unstructured mesh



(a) Problem setup

(b) x-permeability [m^2]

(c) Solution - Δp [Pa]

(d) x-displacement [m]

(e) y-displacement [m]

(f) z-displacement [m]

Poroelasticity - 3D unstructured mesh

ℓ	fine grid			c. grid (M)		c. grid (F)		coarsening ratio		
	# cell	# node	# dof	# cell	# dof	# cell	# dof	mech	flow	total
0	9,140	1,956	15,008	12	156	29	29	37.62	315.17	81.12
1	61,656	11,570	96,366	64	963	156	156	36.04	395.23	86.12
2	274,299	48,633	420,198	256	4,032	677	677	36.19	405.17	89.23

ℓ	IC(0)		Sym. Gauss-Seidel		ℓ_1-Jacobi ($\times 2$)	
	MsRSB	no MS	**MsRSB**	no MS	**MsRSB**	no MS
0	**55**	82	**64**	103	**100**	270
1	**65**	144	**67**	159	**97**	419
2	**68**	217	**67**	235	**93**	642

Summary – MultiScale Formulations for Flow & Mechanics in Porous Media

Many acronyms:

- **MSFE**
- **MSFV**
- **AMS**
- **TAMS**
- **Multi-Level Multi-Scale**

Acknowledgements

- Industrial Consortium on Reservoir Simulation Research at Stanford (SUPRI-B)

- MAELSTROM Project Sponsored by TotalEnergies