Non-intrusive ROM of linear poroelasticity in porous media
(https://arxiv.org/abs/2101.11810)

Teeratorn Kadeethum, Francesco Ballarin, and Nikolaos Bouklas

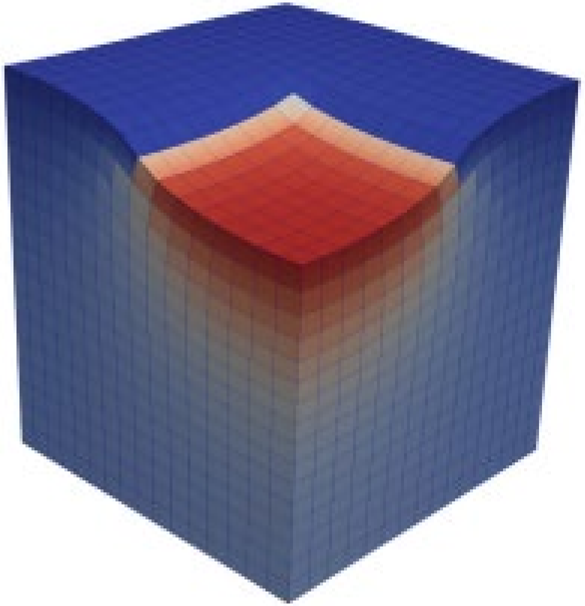
Governing equations

Momentum balance equation

$$
\begin{aligned}
\nabla \cdot \boldsymbol{\sigma}'(\boldsymbol{u}) - \alpha \nabla \cdot (p\mathbf{I}) + \boldsymbol{f} = 0 \quad &\text{in} \quad \Omega \times \mathbb{T}, \\
\boldsymbol{u} = \boldsymbol{u}_D \quad &\text{on} \quad \partial\Omega_u \times \mathbb{T}, \\
\boldsymbol{\sigma}(\boldsymbol{u}) \cdot \mathbf{n} = \boldsymbol{t}_D \quad &\text{on} \quad \partial\Omega_t \times \mathbb{T}, \\
\boldsymbol{u} = \boldsymbol{u}_0 \quad &\text{in} \quad \Omega \text{ at } t = 0,
\end{aligned}
$$

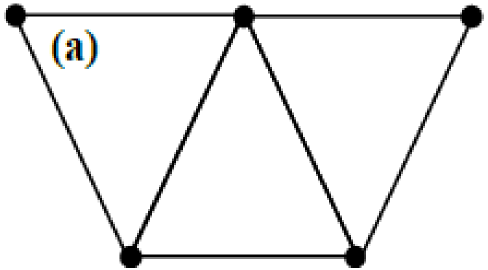
Mass balance equation

$$
\begin{aligned}
\left(\frac{1}{M} + \frac{\alpha^2}{K}\right)\frac{\partial p}{\partial t} + \frac{\alpha}{K}\frac{\partial \sigma_v}{\partial t} - \boldsymbol{\kappa}\nabla p = g \quad &\text{in} \quad \Omega \times \mathbb{T}, \\
p = p_D \quad &\text{on} \quad \partial\Omega_p \times \mathbb{T}, \\
-\boldsymbol{\kappa}\nabla p \cdot \mathbf{n} = q_D \quad &\text{on} \quad \partial\Omega_q \times \mathbb{T}, \\
p = p_0 \quad &\text{in} \quad \Omega \text{ at } t = 0,
\end{aligned}
$$

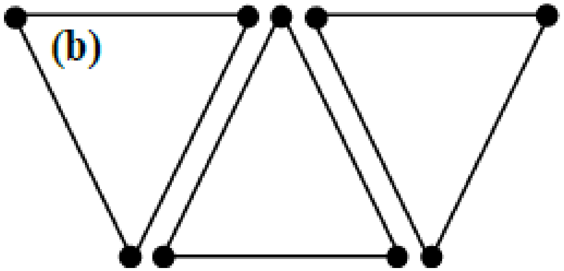


Pic from: J.Choo. Stabilized mixed continuous/enriched Galerkin formulations for locally mass conservative poromechanics. CMAME. 2019.

Full-order model (FOM) – finite element



CG: for displacement field



DG: for pressure field

Monolithic

$$\begin{bmatrix} \mathscr{J}_{uu}^{\mathrm{CG}_2\times\mathrm{CG}_2} & \mathscr{J}_{up}^{\mathrm{CG}_2\times\mathrm{DG}_1} \\ \mathscr{J}_{pu}^{\mathrm{DG}_1\times\mathrm{CG}_2} & \mathscr{J}_{pp}^{\mathrm{DG}_1\times\mathrm{DG}_1} \end{bmatrix} \left\{ \begin{array}{c} \left(\delta u_h^n\right)^{\mathrm{CG}_2} \\ \left(\delta p_h^n\right)^{\mathrm{DG}_1} \end{array} \right\} = - \left\{ \begin{array}{c} R_u^{\mathrm{CG}_2} \\ R_p^{\mathrm{DG}_1} \end{array} \right\}$$

Time-stepping

$$\mathrm{BDF}_1\left(\varphi^n\right) := \frac{1}{\Delta t^n}\left(\varphi^n - \varphi^{n-1}\right)$$

https://www.sciencedirect.com/science/article/pii/S0309170819312576
https://link.springer.com/article/10.1007/s11004-020-09893-y
https://www.sciencedirect.com/science/article/pii/S0021999120308044

Reduced-order model (ROM) – data driven

1. Initialization of training set

$$[\boldsymbol{\mu}^{(1)}, \boldsymbol{\mu}^{(2)}, \cdots, \boldsymbol{\mu}^{(M-1)}, \boldsymbol{\mu}^{(M)}]$$

2. Full order model (FOM)

$$\text{FOM} = \begin{matrix} \boldsymbol{u}_h(\boldsymbol{\mu}^{(1)}), p_h(\boldsymbol{\mu}^{(1)}) \\ \vdots \\ \boldsymbol{u}_h(\boldsymbol{\mu}^{(M)}), p_h(\boldsymbol{\mu}^{(M)}) \end{matrix}$$

3. Proper orthogonal decomposition (POD)

$$\left.\begin{matrix} \boldsymbol{u}_h(\boldsymbol{\mu}^{(1)}), \cdots, \boldsymbol{u}_h(\boldsymbol{\mu}^{(M)}) \\ \text{Same goes for } p_h \end{matrix}\right\} \boldsymbol{w}_1^{\boldsymbol{u}}, \cdots, \boldsymbol{w}_{\mathrm{N}}^{\boldsymbol{u}}$$

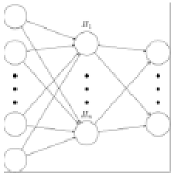
4. L^2 projection

$$\left.\begin{matrix} \boldsymbol{u}_h(\boldsymbol{\mu}^{(1)}), \cdots, \boldsymbol{u}_h(\boldsymbol{\mu}^{(M)}) \\ \boldsymbol{w}_1^{\boldsymbol{u}}, \cdots, \boldsymbol{w}_{\mathrm{N}}^{\boldsymbol{u}} \end{matrix}\right\} \boldsymbol{\theta}^{\boldsymbol{u}} = [\theta_1^u, \cdots, \theta_{\mathrm{N}}^u]$$

Same goes for p_h

5. Artificial neural networks (ANN)

$t, \boldsymbol{\mu}$



$\widehat{\boldsymbol{\theta}^{u}}$

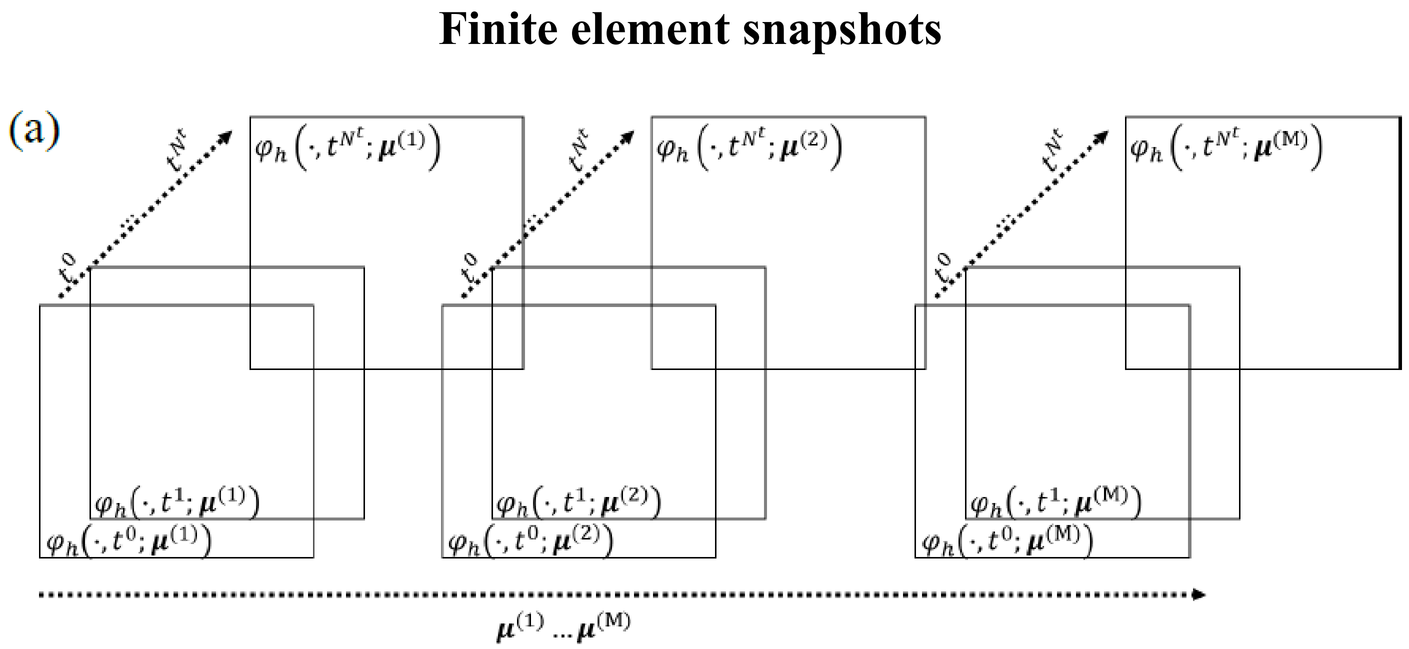
$\widehat{\boldsymbol{\theta}^{p}}$

6. Online phase

$$\widehat{\boldsymbol{u}_h}(t, \boldsymbol{\mu}) = \sum_{k=1}^{N} \widehat{\boldsymbol{\theta}_k^{\boldsymbol{u}}}(t, \boldsymbol{\mu})\, \boldsymbol{w}_k^{\boldsymbol{u}}$$

Same goes for $\widehat{p_h}$

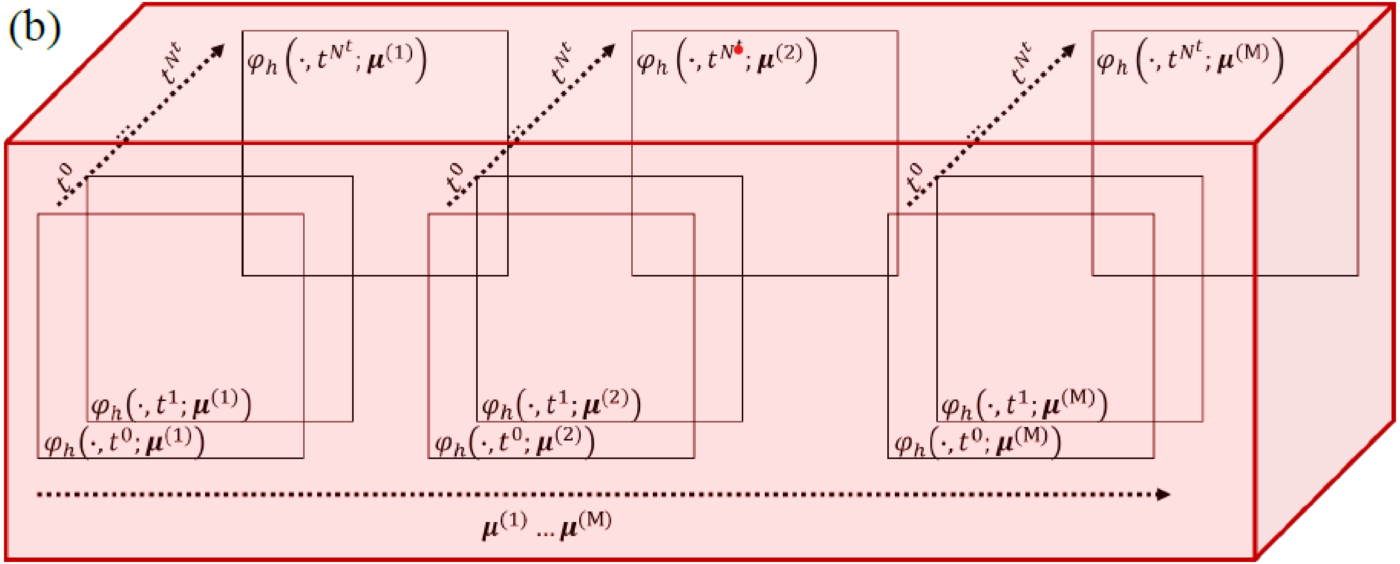
Proper orthogonal decomposition (POD)



Proper orthogonal decomposition (POD)

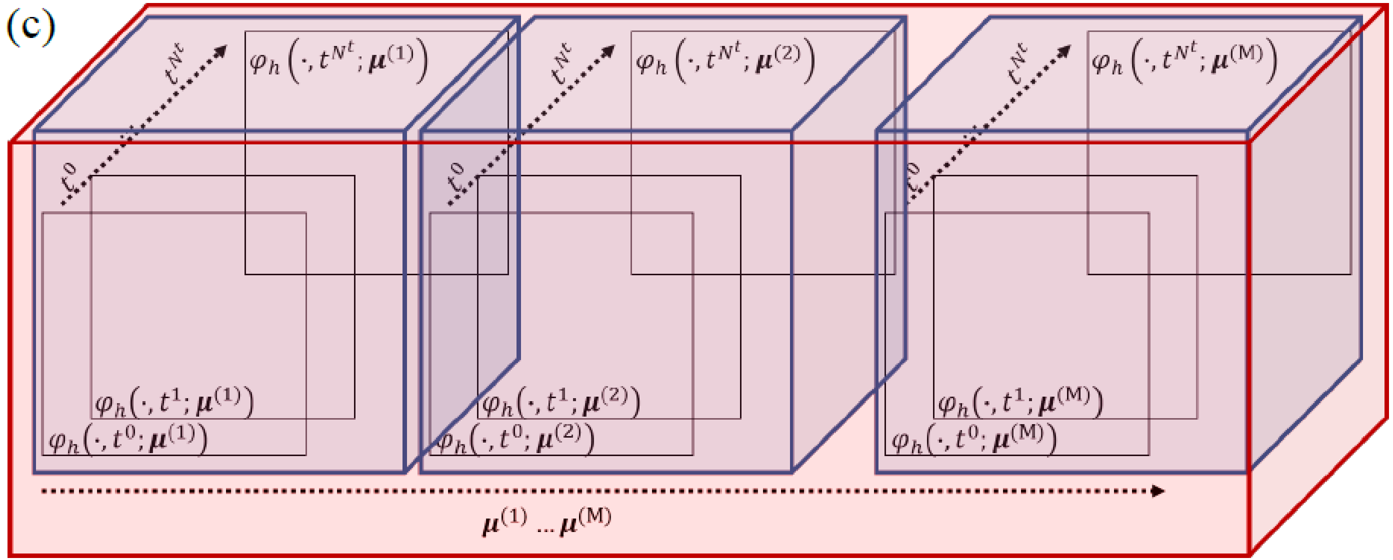
Single compression

(b)

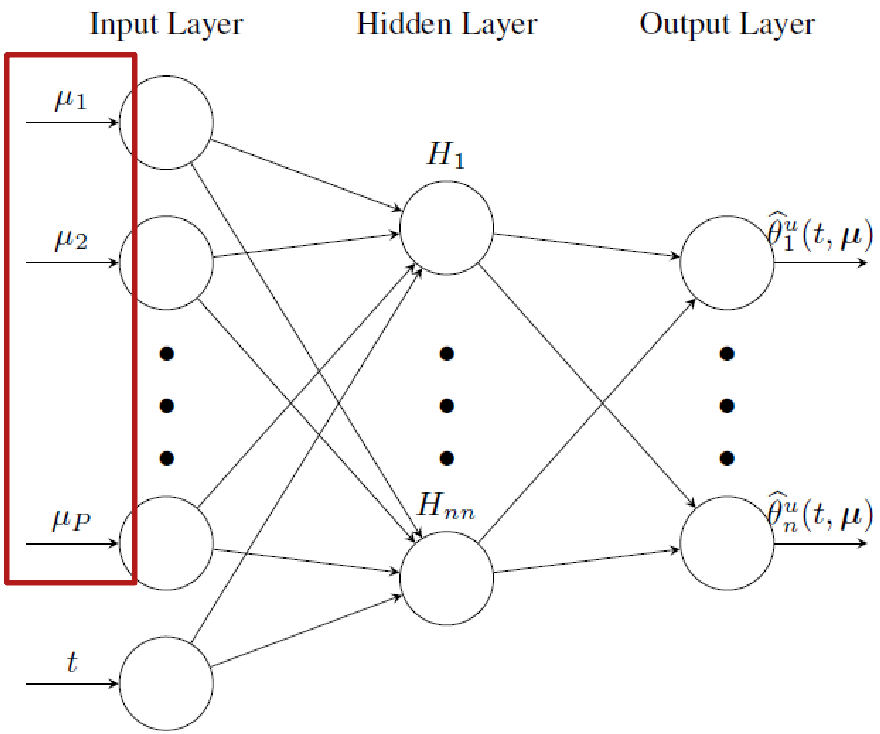


Proper orthogonal decomposition (POD)

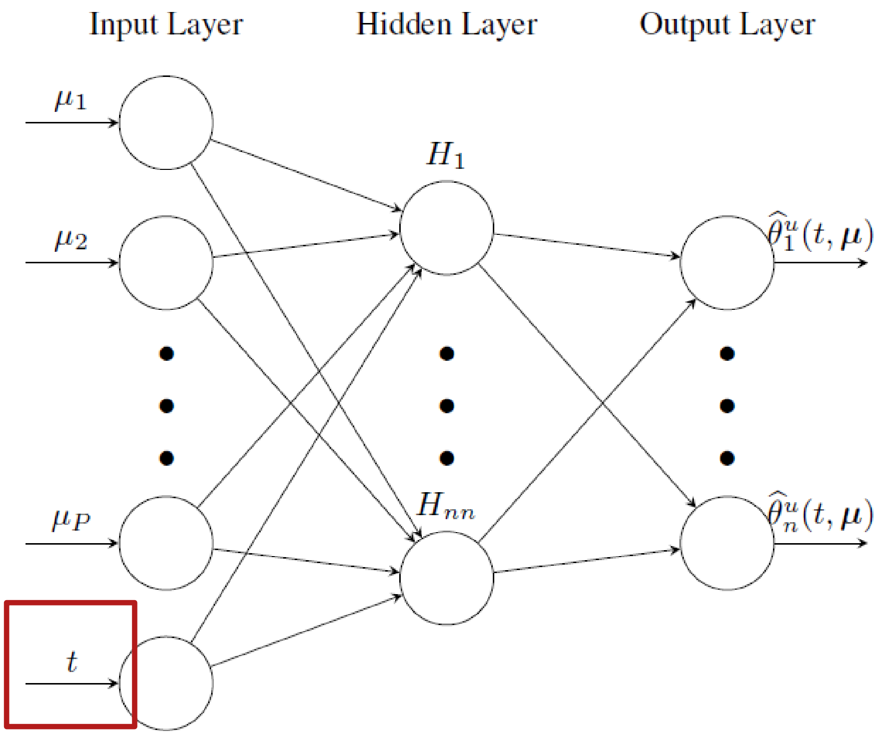
Nested compression



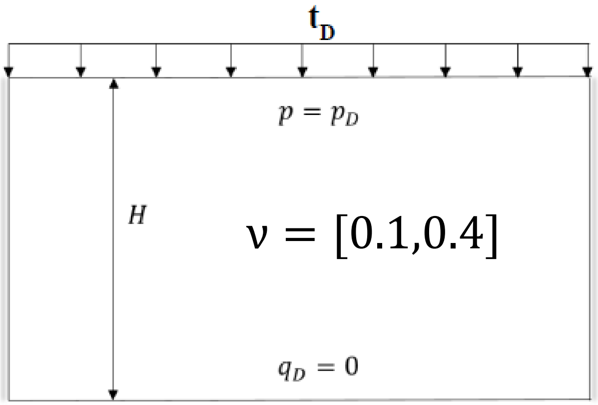
Artificial neural networks (ANN)

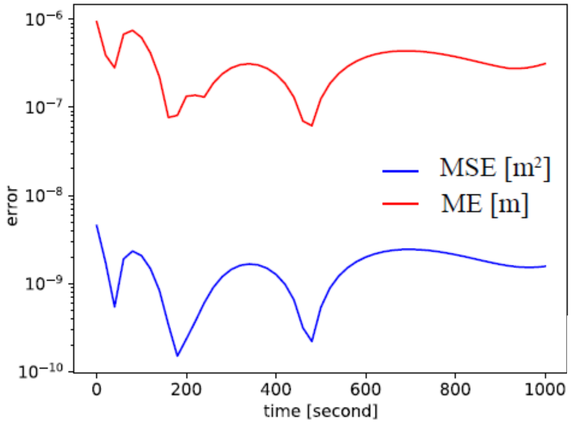


Artificial neural networks (ANN)



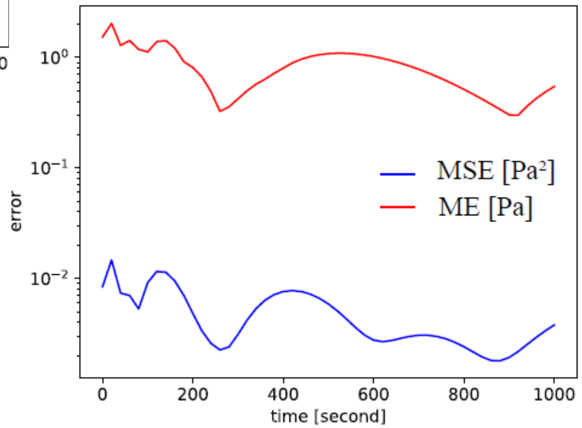
Terzaghi's consolidation problem





Displacement
(1 x 10^{-4})

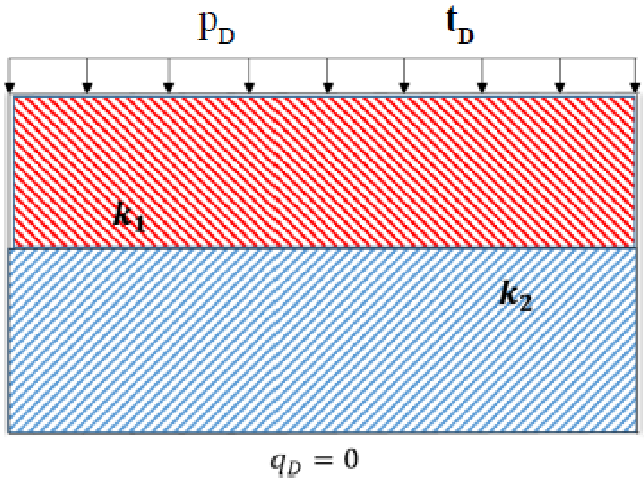
Pressure
(1 x 10^{3})

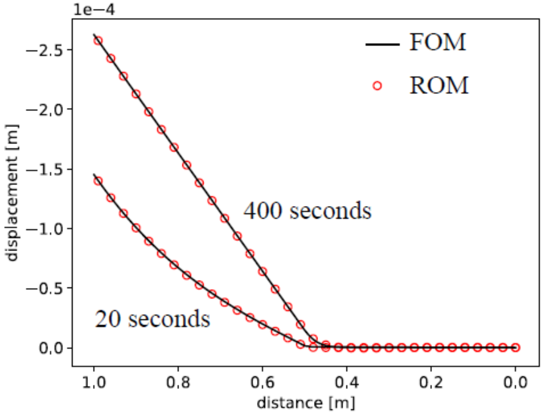


$$\mathrm{MSE}_{\varphi}(t,\boldsymbol{\mu}) := \|\varphi_h(\cdot;t,\boldsymbol{\mu}) - \widehat{\varphi}_h(\cdot;t,\boldsymbol{\mu})\|_{\varphi}^{2}$$

$$\mathrm{ME}_{\varphi}(t,\boldsymbol{\mu}) := \|\varphi_h(\cdot;t,\boldsymbol{\mu}) - \widehat{\varphi}_h(\cdot;t,\boldsymbol{\mu})\|_{\varphi}^{\infty}$$

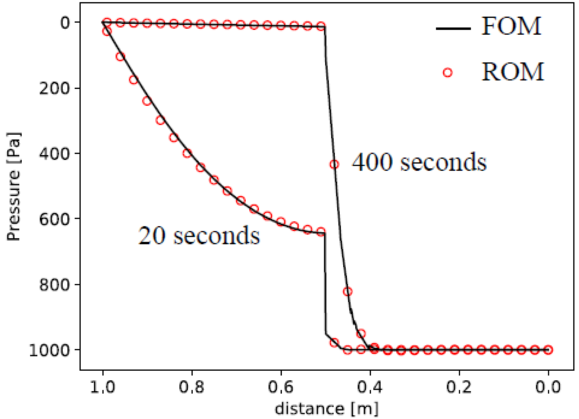
Consolidation problem with 2-layered material





Displacement

Pressure



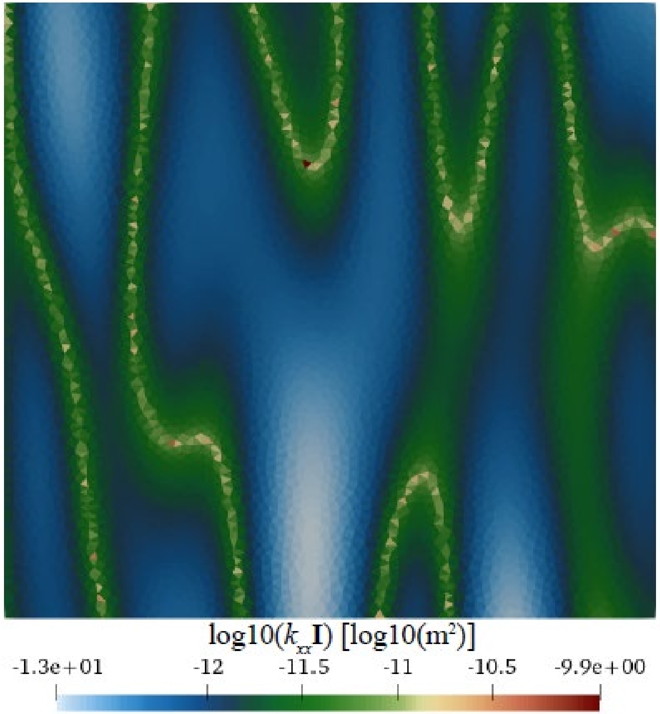
$$k_1 := \begin{bmatrix} 1.0 \times 10^{-12} & 0.0 \\ 0.0 & 1.0 \times 10^{-12} \end{bmatrix}$$

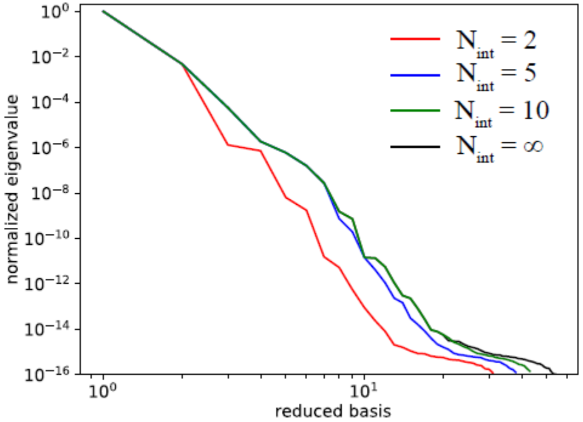
$$k_2 := \begin{bmatrix} k_{xx} & 0.0 \\ 0.0 & k_{xx} \end{bmatrix}.$$

$$k_{xx} = [1.0 \times 10^{-16}, 1.0 \times 10^{-15}]$$

$$\mu = (\nu, \alpha) \in [0.1, 0.4] \times [0.4, 1.0]$$

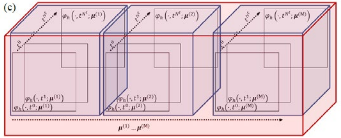
Heterogeneous media - POD

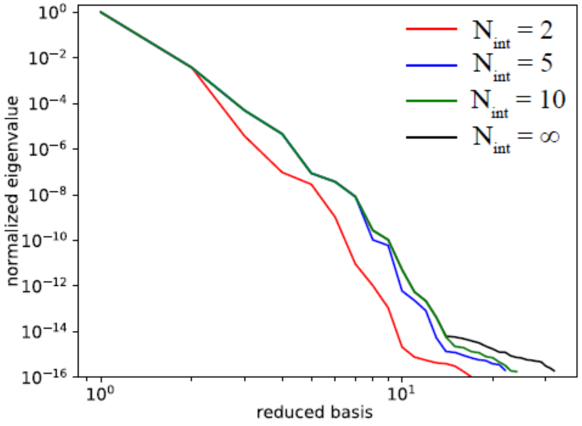




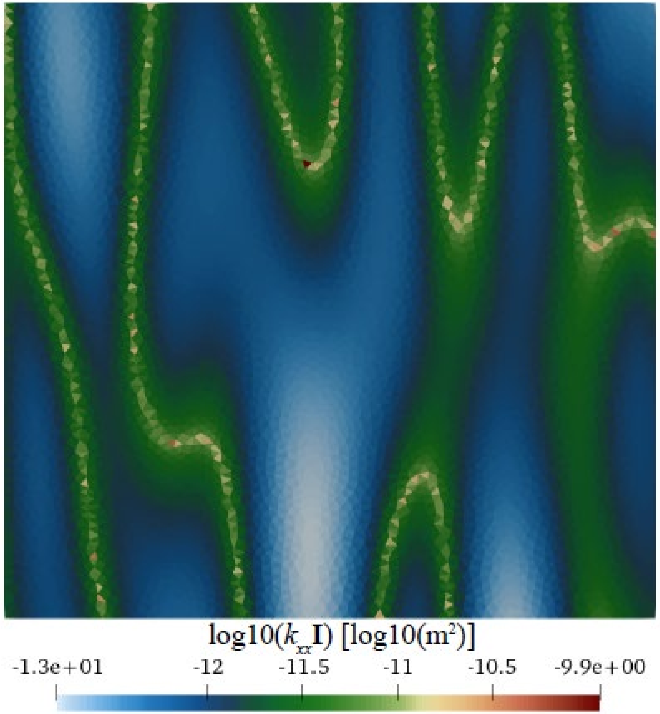
Displacement

Pressure





Heterogeneous media - POD

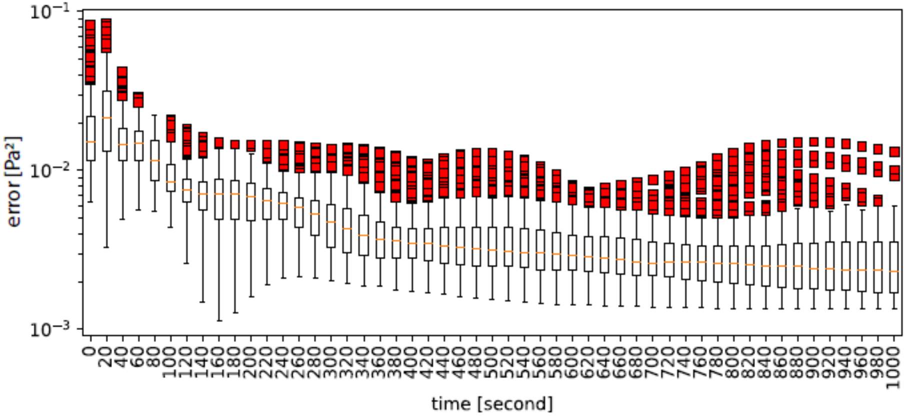


Wall time used to perform POD

	$N_{\text{int}} = 2$	$N_{\text{int}} = 5$	$N_{\text{int}} = 10$	$N_{\text{int}} = \infty$
$M = 100$	100	125	170	1574
$M = 400$	437	650	1437	36705
$M = 900$	1168	2319	6475	268754

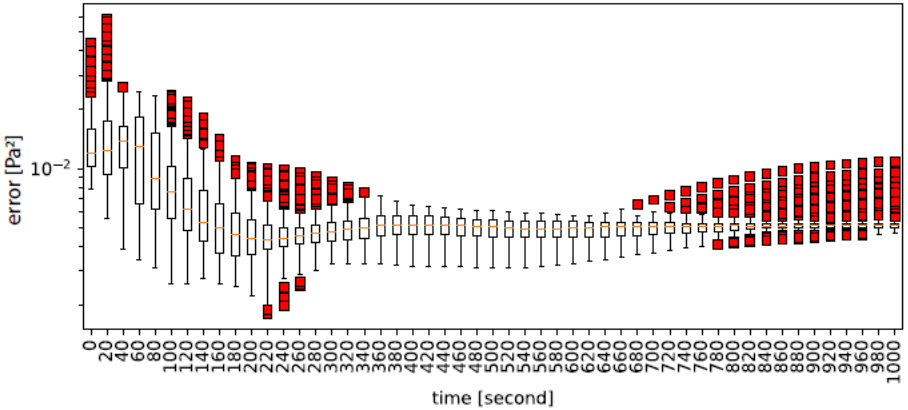
Nested compression could save a lot of time

Heterogeneous media – 1000 test cases



snapshot = 400, reduced basis = 10 (5), hidden layers = 3, and neurons = 7

snapshot = 900, reduced basis = 20 (10), hidden layers = 5, and neurons = 10



Heterogeneous media – costs

Table 9: Example 4: Comparison of the wall time (seconds) used for sensitivity analysis

	M = 400 (model 1)	M = 900 (model 2)	FOM
Train FOM snapshots	7160	16020	-
Perform POD	1437	6475	-
Train ANN	7064	18492	-
Prediction - 1000 testing μ	2895	3160	17790
Prediction - per testing μ	2.9	3.2	17.8

Taking the training time into account, we need to perform at least 1050 and 2850 inquiries (online phase) to have a break-even point for model 1 and model 2, respectively.

Current works

- Nonlinear compression – autoencoder and its variants
- Adaptive mesh and timestep
- Physics-informed neural networks

Thank you ☺

https://arxiv.org/abs/2101.11810

https://gitlab.com/multiphenics/multiphenics

https://gitlab.com/RBniCS/RBniCS