Reducing non-linear PDEs using a reduced integration proper orthogonal decomposition method

Elisa Schenone, **Jack S. Hale**, Lars Beex
Stéphane P. A. Bordas

Ongoing work with Davide Baroli

UNIVERSITÉ DU LUXEMBOURG
Fonds National de la Recherche Luxembourg
COFUNDED BY MARIE CURIE ACTIONS
European Research Council
Established by the European Commission

Overview

- Model order reduction and POD.
- The problem with POD for non-linear problems. Assembly of non-linear operators dominates solution time.
- Novel reduced integration POD method.
    - DOLFIN.
    - UFL-based specification of weak forms.
    - SLEPc.
- Main alternative: Discrete Empirical Interpolation Method [S. Chaturantabut and Sorensen 09].
    - Difference: DEIM selects significant points to evaluate strong form, our method selects significant regions to evaluate weak forms.
    - Straightforward implementation in any finite element method code.
- Results.

Model Order Reduction

- Any method that reduces the runtime complexity of a model.
- Why?
  - Interactive rate simulation.
  - Multi-level Monte Carlo methods (Giles, Schwab).
  - Bayesian inference problems (Stuart).

Proper Orthogonal Decomposition (POD)

- Find a new set of *global* basis functions that optimally represent the data in a set of snapshots.

- Dimension of POD basis significantly smaller than original FEM basis.

- Snapshots, POD basis, projections - (a lot of work) *offline*.

- Small linear system solve - (a tiny bit of work) *online.*

Proper Orthogonal Decomposition (POD)

Input: Solution snapshots

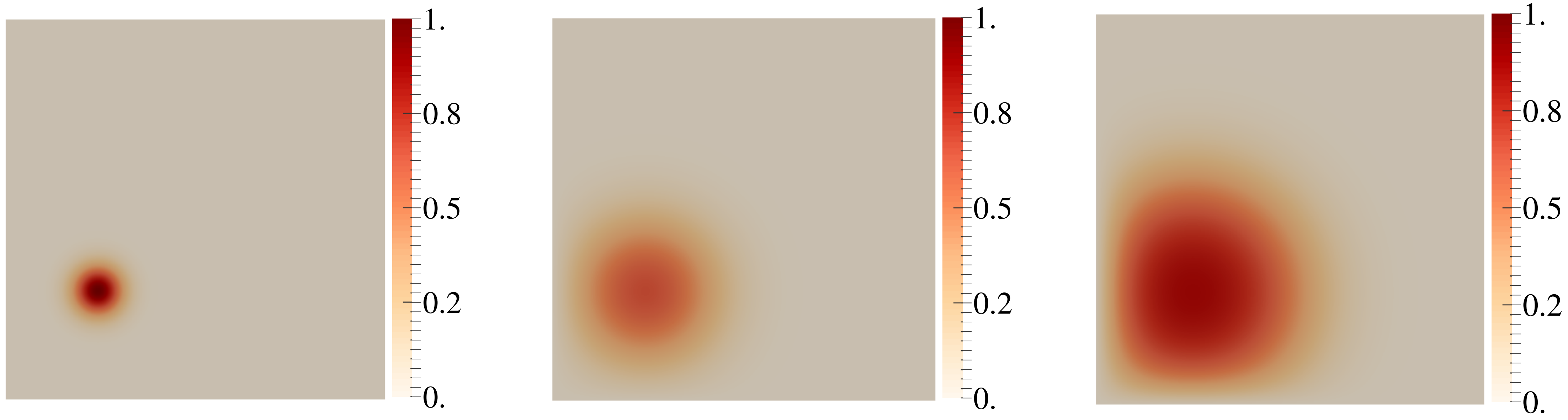
$$u_i \in V_h \quad dim(V_h) = m \quad \{u_1, u_2, u_3, ..., u_i\}$$

Output: POD basis functions.

$$\{\phi_1, \phi_2, \ldots, \phi_N\}$$

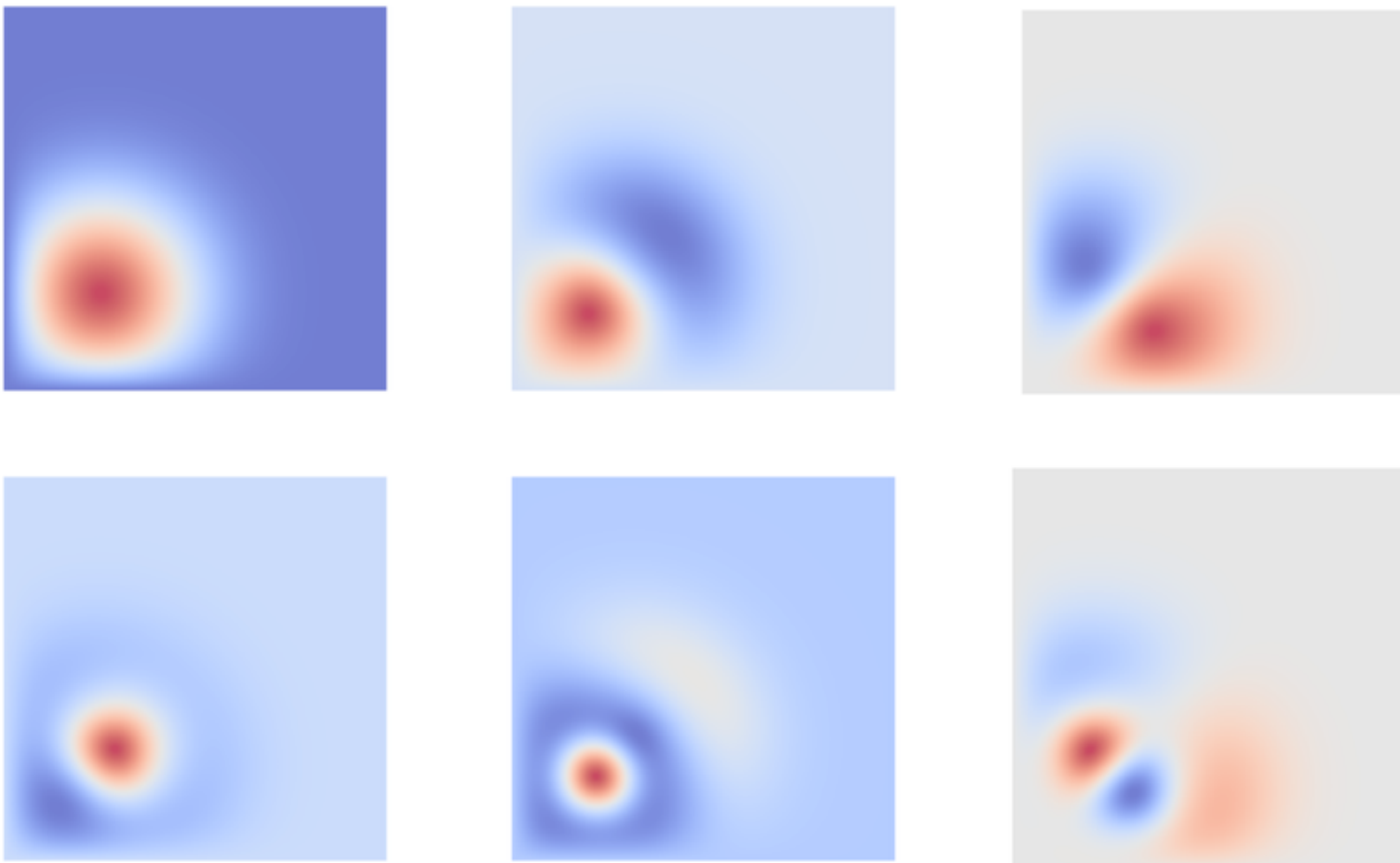
$$N \ll m$$

Non-linear reaction-diffusion

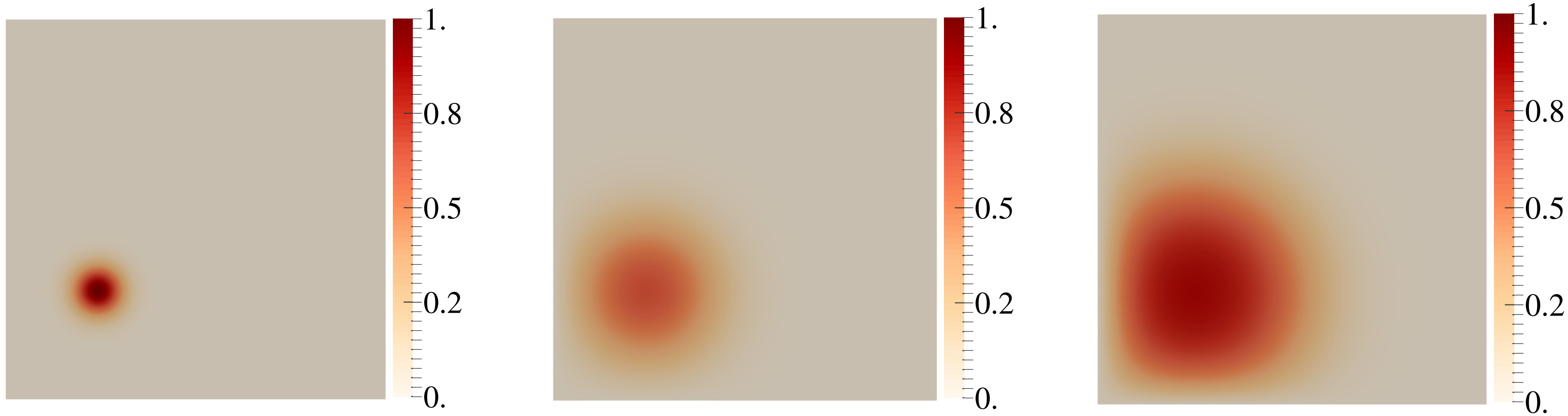


$$\left(\left(\frac{1}{\Delta t}+\frac{c}{2}\right)\mathbf{M}+\mathbf{K}\right)\mathbf{u}^{k+1}=\left(\frac{1}{\Delta t}-\frac{c}{2}\right)\mathbf{M}\mathbf{u}^{k}+\mathbf{L}(\mathbf{u}^{k})$$

POD Basis Functions



Non-linear reaction-diffusion



$$\left(\left(\frac{1}{\Delta t} + \frac{c}{2}\right)\mathbf{M} + \mathbf{K}\right)\mathbf{u}^{k+1} = \left(\frac{1}{\Delta t} - \frac{c}{2}\right)\mathbf{M}\mathbf{u}^{k} + \mathbf{L}(\mathbf{u}^{k})$$

$$((\frac{1}{\Delta t} + \frac{c}{2})\boldsymbol{\Phi}^{\mathbf{T}}\mathbf{M}\boldsymbol{\Phi} + \boldsymbol{\Phi}^{\mathbf{T}}\mathbf{K}\boldsymbol{\Phi})\tilde{\mathbf{u}}^{k+1} = (\frac{1}{\Delta t} - \frac{c}{2})\boldsymbol{\Phi}^{\mathbf{T}}\mathbf{M}\boldsymbol{\Phi}\tilde{\mathbf{u}}^{k} + \boldsymbol{\Phi}^{\mathbf{T}}\mathbf{L}(\tilde{\mathbf{u}}^{k})$$

once once once every timestep

Must re-assemble linear form for every timestep

Newton System

Linearisation about current solution

$$a(m; u\_; \delta u; v) = -F(m; u\_; v)$$

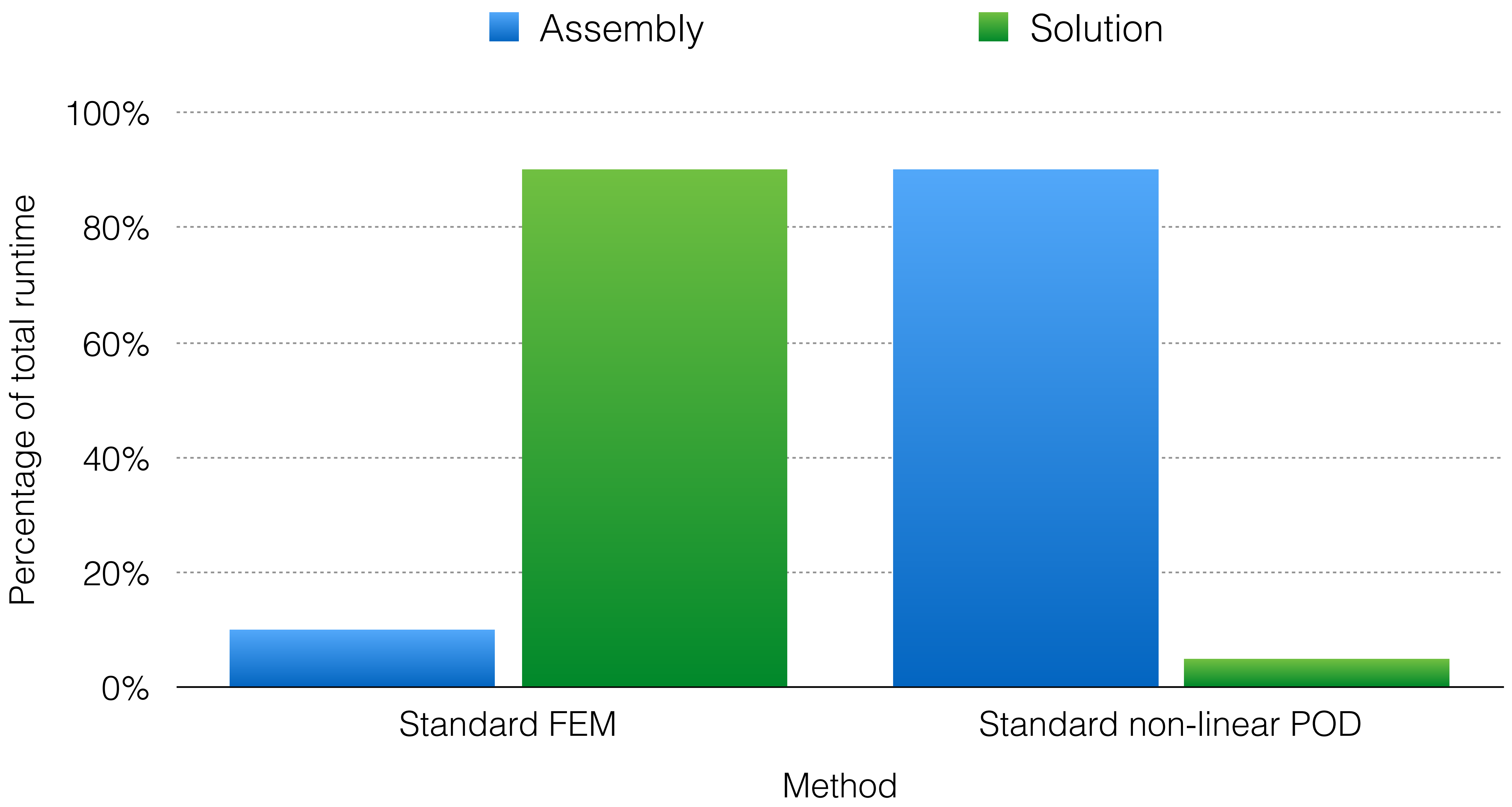
Newton system

$$\mathbf{A}(u\_)\,\delta u = \mathbf{b}(u\_)$$

every Newton step every Newton step

Must re-assemble linear and bilinear operators for every Newton step

Split of time



Same observation as [S. Chaturantabut and Sorensen 09]

Observations

- *Observation 1:* POD basis function optimally represents data in snapshots.
- *Observation 2:* Original finite element mesh is *not optimal* for POD basis functions.

Solution

- Construct a new mesh for *online assembly* of all non-linear operators, that is *optimal for the POD basis functions*.

- Hopefully, this new mesh will have significantly fewer cells.

- Fewer cells means less computational costs during assembly of operators associated with non-linear part of PDE.

Offline Algorithm

- Greedy algorithm on sequence of POD basis functions.

$$\{\phi_1, \phi_2, \ldots, \phi_N\}$$

- Start with coarse mesh and first POD basis function.

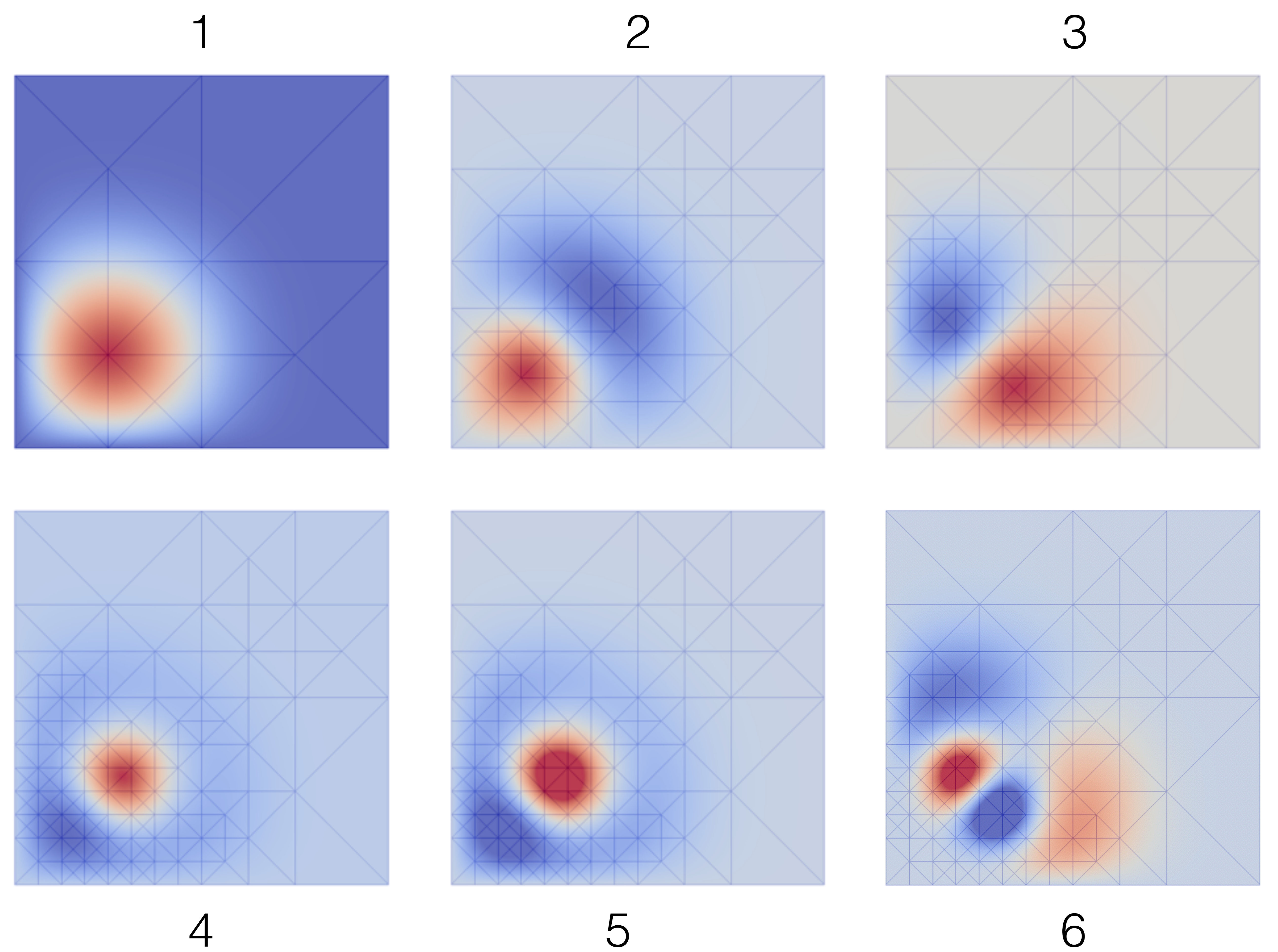
$$\mathcal{T}_h^1$$

- Calculate local error indicator.

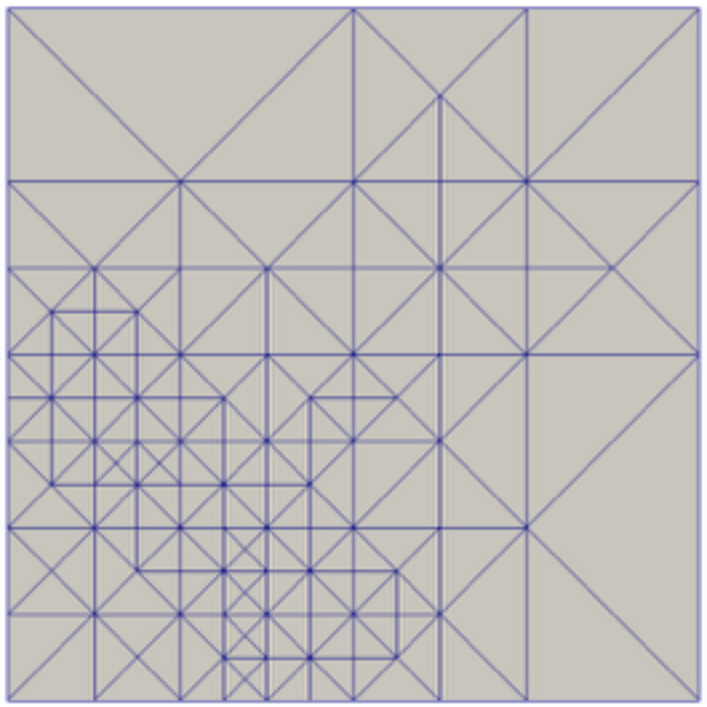
$$\eta_i = \int_{T_i^1(Q_n)} |\phi_1| \, \mathrm{d}x - \int_{T_i^1(Q_{n+1})} |\phi_1| \, \mathrm{d}x$$

- Refine until tolerance met (e.g. 1%, 5%, 10%).

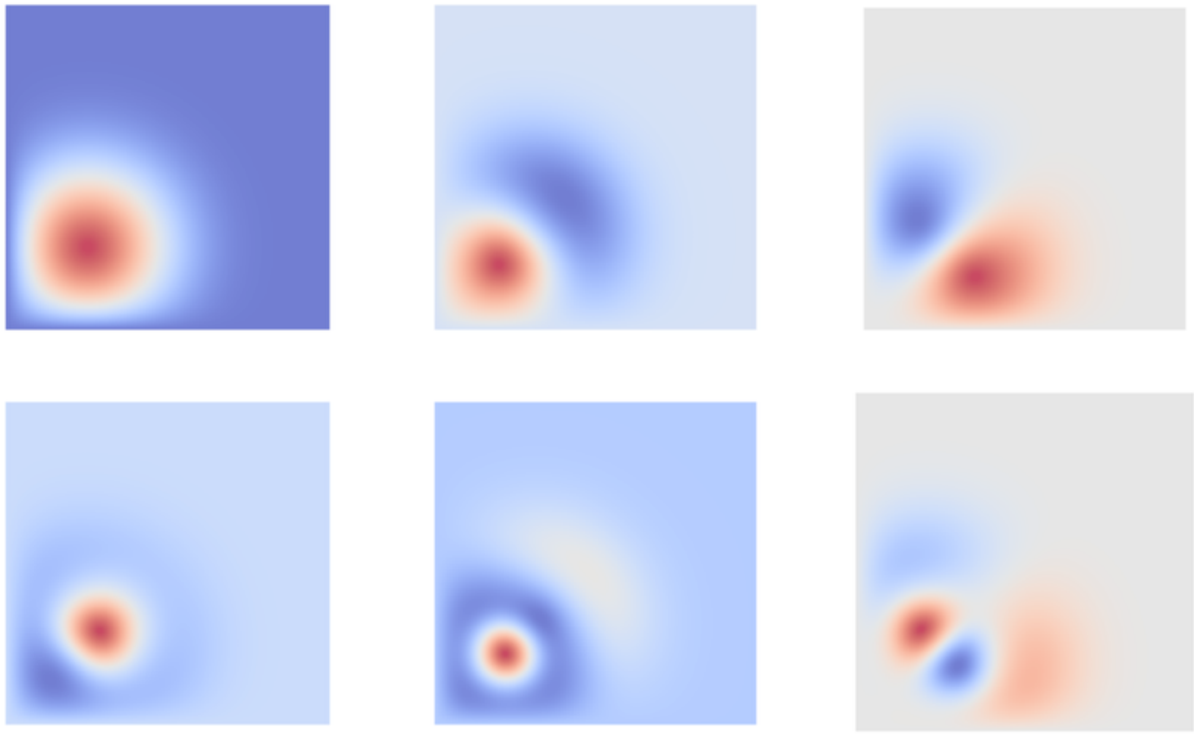
- Starting with new mesh, take second POD basis function, repeat.


1
2
3
4
5
6

Interpolate modes



Mesh 6

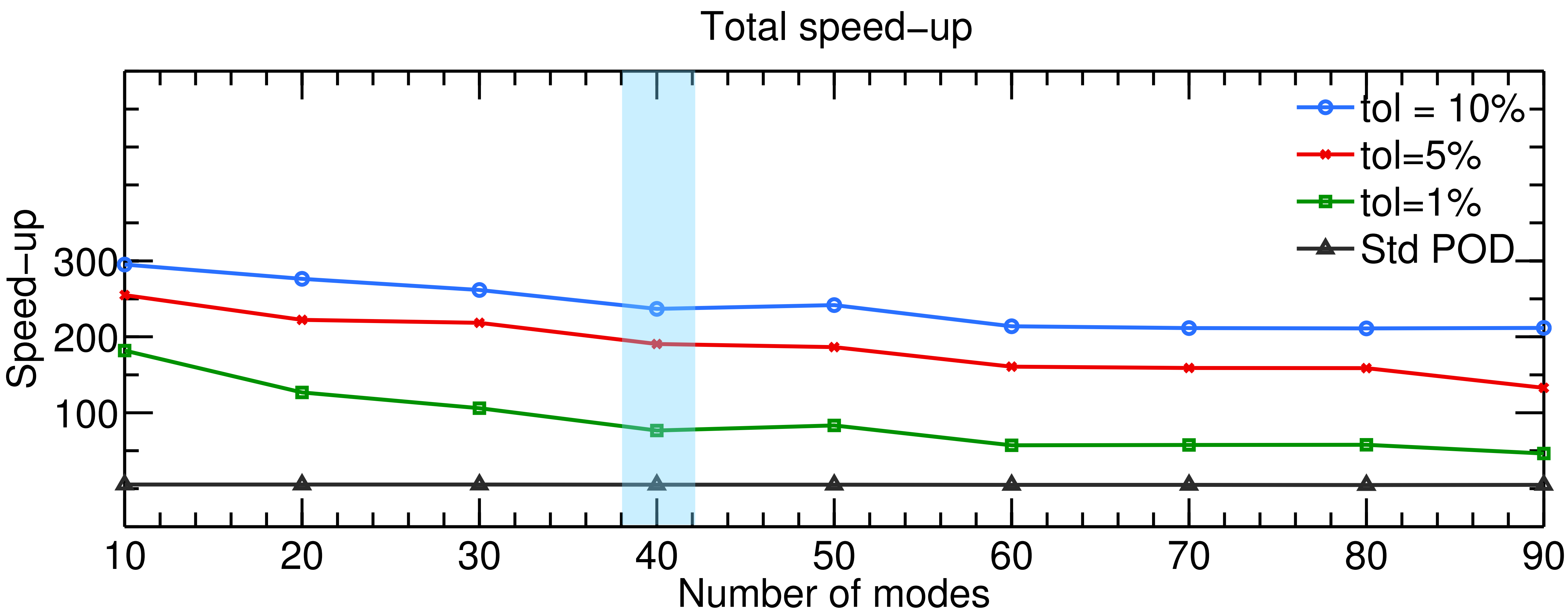


$$((\frac{1}{\Delta t} + \frac{c}{2})\mathbf{\Phi}^\mathbf{T}\mathbf{M}\mathbf{\Phi} + \mathbf{\Phi}^\mathbf{T}\mathbf{K}\mathbf{\Phi})\tilde{\mathbf{u}}^{k+1} = (\frac{1}{\Delta t} - \frac{c}{2})\mathbf{\Phi}^\mathbf{T}\mathbf{M}\mathbf{\Phi}\tilde{\mathbf{u}}^{k} + \mathbf{\Phi}^\mathbf{T}\mathbf{L}(\tilde{\mathbf{u}}^{k})$$

once once once everytime, but faster!

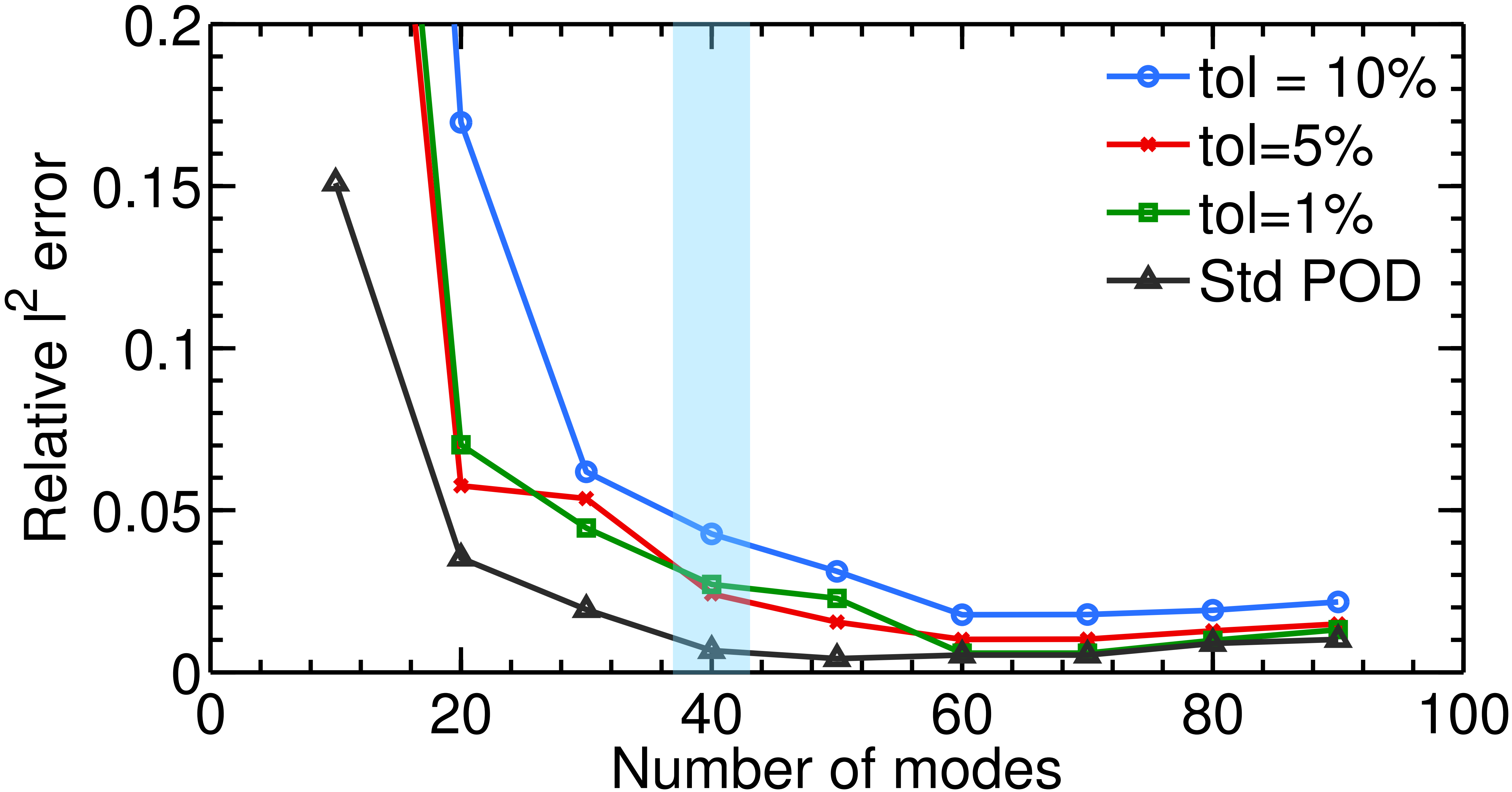
We assemble the operators where the modes are important.

Speed up against FEM



40 modes: POD: **5x**, 1%: **90x**, 5%: **220x**, 10%: **300x**

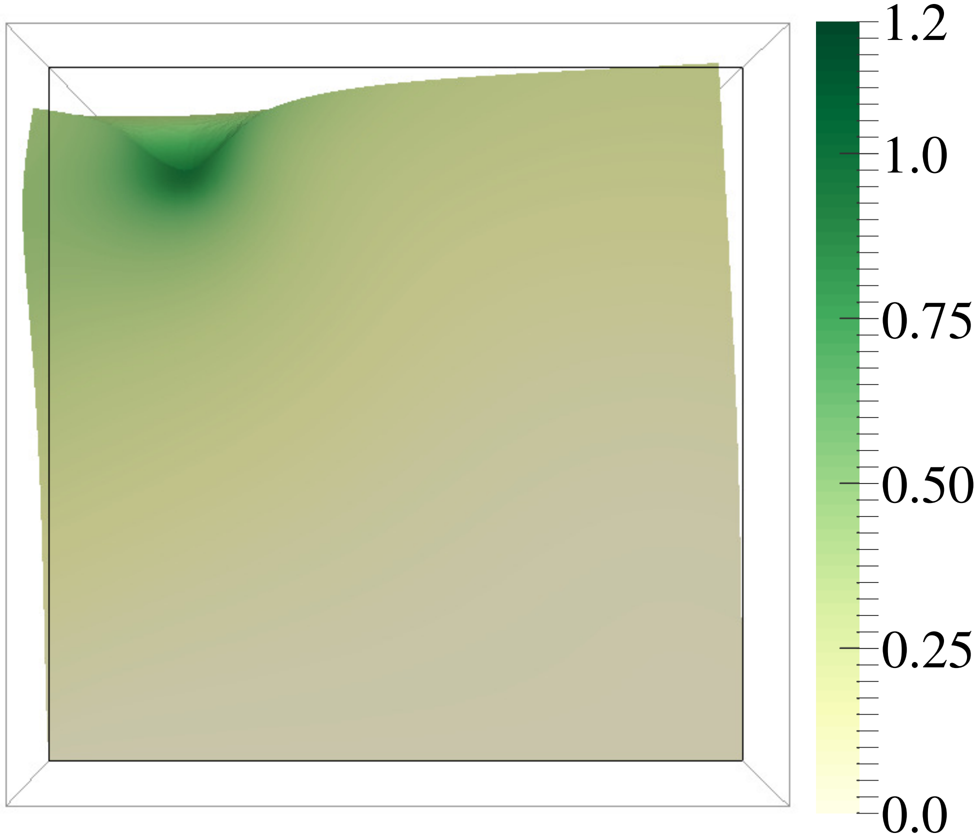
Error against FEM



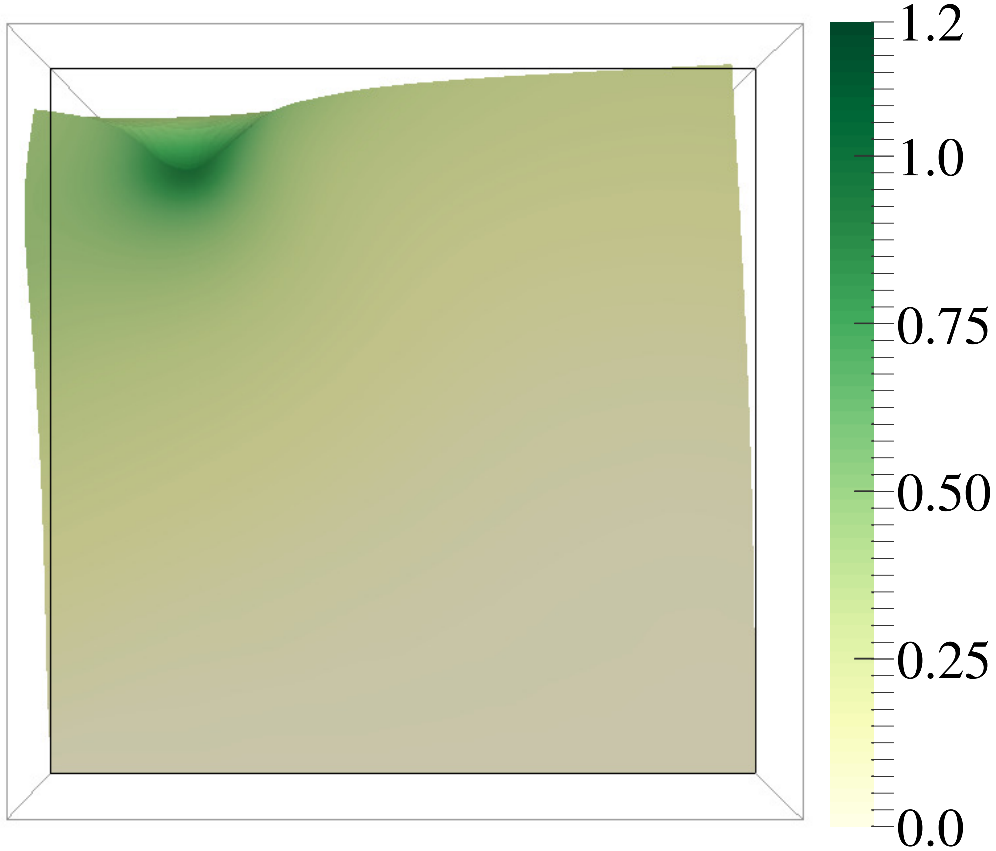
40 modes POD: **0.008**, 1%: **0.03**, 5%: **0.03**, 10%: **0.045**

Hyperelasticity (Newton)

FEM



Reduced integration POD



30x speed up over standard POD

Summary

- Simple method to overcome issue of operator construction cost in reduced order methods.
- Four ingredients:
  - Mesh refinement.
  - Error indicators.
  - Projections and interpolations.
  - Linear algebra operations.
- Key idea: generate optimal meshe for optimal POD basis functions.

Questions?