A model order reduction technique for speeding up computational homogenisation

Olivier Goury, Pierre Kerfriden, Wing Kam Liu, Stéphane Bordas
Cardiff University



CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Outline

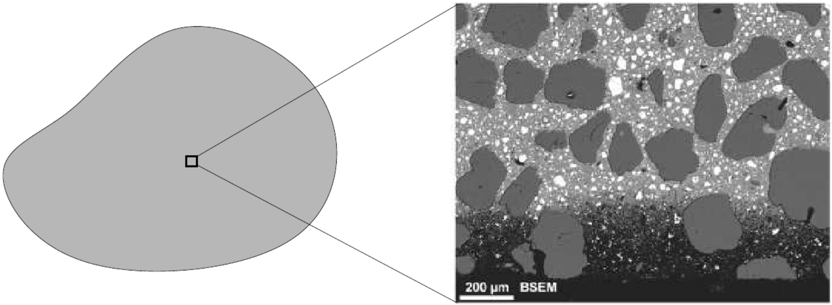
- Introduction
  - Heterogeneous materials
  - Computational Homogenisation
- Model order reduction in Computational Homogenisation
  - Proper Orthogonal Decomposition (POD)
  - Optimal snapshot selection
  - System approximation
  - Results
- Conclusion

Outline

Outline

Heterogeneous materials

Many natural or engineered materials are **heterogeneous**



- Homogeneous at the macroscopic length scale
- Heterogeneous at the microscopic length scale

Heterogeneous materials

Need to model the macro-structure while taking the micro-structures into account

$\Longrightarrow$ better understanding of material behaviour, design, etc..

Two choices:

- Direct numerical simulation: brute force!
- Multiscale methods: when modelling a non-linear materials
  $\Longrightarrow$ Computational Homogenisation

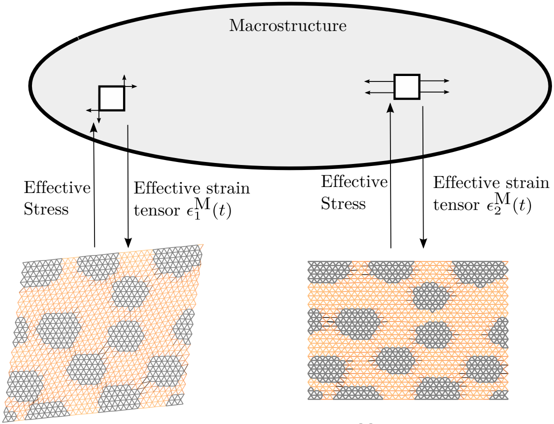
Outline

Introduction
- Heterogeneous materials
- Computational Homogenisation

Model order reduction in Computational Homogenisation
- Proper Orthogonal Decomposition (POD)
- Optimal snapshot selection
- System approximation
- Results

Conclusion

Semi-concurrent Computational Homogenisation (FE^2, ...)



$$\mathbf{u_1}(t) = \epsilon_1^M(t)(\mathbf{x} - \bar{\mathbf{x}}) + \tilde{\mathbf{u_1}}(t)$$

$$\mathbf{u_2}(t) = \epsilon_2^M(t)(\mathbf{x} - \bar{\mathbf{x}}) + \tilde{\mathbf{u_2}}(t)$$

Problem

- For non-linear materials: Have to solve a RVE boundary value problem at each point of the macro-mesh where it is needed. Still expensive!
- Need parallel programming

Outline

Outline

Strategy

- Use model order reduction to make the solving of the RVE boundary value problems computationally achievable
- Linear displacement:

$$\boldsymbol{\epsilon}^{\mathrm{M}}(t) = \begin{pmatrix} \epsilon_{xx}(t) & \epsilon_{xy}(t) \\ \epsilon_{xy}(t) & \epsilon_{yy}(t) \end{pmatrix}$$

$$\mathbf{u}(t) = \boldsymbol{\epsilon}^{\mathrm{M}}(t)(\mathbf{x} - \bar{\mathbf{x}}) + \tilde{\mathbf{u}} \quad \text{with} \quad \tilde{\mathbf{u}}_{|\Gamma} = \mathbf{0}$$

Fluctuation $\tilde{\mathbf{u}}$ approximated by: $\tilde{\mathbf{u}} \approx \sum_i \phi_{\boldsymbol{i}} \alpha_i$

Projection-based model order reduction

The RVE problem can be written:

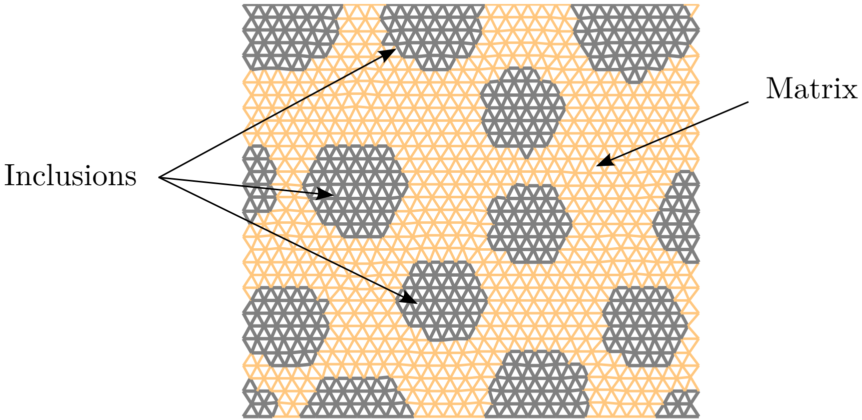
$$\underbrace{\underline{\mathbf{F}}_{\mathrm{int}}(\tilde{\underline{\mathbf{u}}}(\epsilon^{\mathrm{M}}(t)), \epsilon^{\mathrm{M}}(t))}_{\text{Non-linear}} + \underline{\mathbf{F}}_{\mathrm{ext}}(\epsilon^{\mathrm{M}}(t)) = \underline{\mathbf{0}} \quad (1)$$

We are interested in the solution $\tilde{\underline{\mathbf{u}}}(\epsilon^{\mathrm{M}})$ for many different values of $\epsilon^{\mathrm{M}}(t \in [0, T]) \equiv \epsilon_{xx}, \epsilon_{xy}, \epsilon_{yy}$.

Projection-based model order reduction assumption:

Solutions $\tilde{\underline{\mathbf{u}}}(\epsilon^{\mathrm{M}})$ for different parameters ϵ^{M} are contained in a space of small dimension $span((\phi_{\boldsymbol{i}})_{i \in [\![1,n]\!]})$

RVE boundary value problem



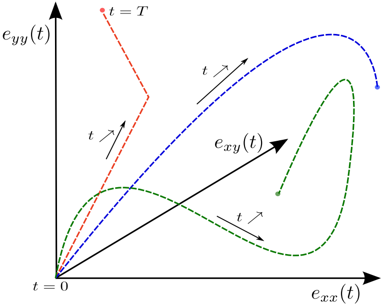
Proper Orthogonal Decomposition (POD)

How to choose the basis $[\phi_{\mathbf{1}}, \phi_{\mathbf{2}}, \ldots] = \mathbf{\Phi}$?

Proper Orthogonal Decomposition (POD)

How to choose the basis $[\phi_1, \phi_2, \ldots] = \mathbf{\Phi}$?

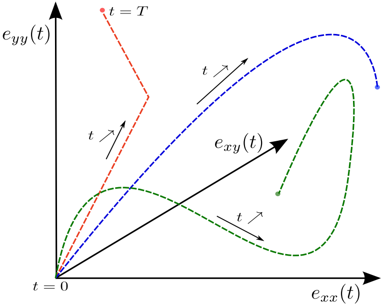
- “Offline“ Stage $\equiv$ Learning stage : Solve the RVE problem for a certain number of chosen values of ϵ^M



Proper Orthogonal Decomposition (POD)

How to choose the basis $[\phi_1, \phi_2, \ldots] = \mathbf{\Phi}$?

- "Offline" Stage $\equiv$ Learning stage : Solve the RVE problem for a certain number of chosen values of ϵ^M



- We obtain a base of solutions (the snapshot): $(\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_{n_S}) = \mathbf{S}$

- Find the basis $[\phi_{\mathbf{1}}, \phi_{\mathbf{2}}, \ldots] = \mathbf{\Phi}$ that minimises the cost function:

$$J(\mathbf{\Phi}) = \sum_{\mu \in \mathcal{P}^{\mathrm{s}}} \left\| \mathbf{u}_i - \sum_{k}^{n_{\mathrm{POD}}} \phi_{\mathbf{k}} . \langle \phi_{\mathbf{k}}, \mathbf{u}_i \rangle \right\|^2 \qquad (2)$$

  with the constraint $\left\langle \phi_{\mathbf{i}}, \phi_{\mathbf{j}} \right\rangle = \delta_{ij}$
- Use SVD (Singular Value Decomposition)

Reduced equations

- Reduced system after linearisation: $\min\limits_{\underline{\boldsymbol{\alpha}}} \quad \|\mathbf{K\Phi}\,\alpha + \mathbf{F_{ext}}\|$

Reduced equations

- Reduced system after linearisation: $\min_{\underline{\boldsymbol{\alpha}}} \quad \|\mathbf{K}\boldsymbol{\Phi}\,\alpha + \mathbf{F_{ext}}\|$
- In the Galerkin framework: $\boldsymbol{\Phi}^T\,\mathbf{K}\boldsymbol{\Phi}\,\alpha + \boldsymbol{\Phi}^T\,\mathbf{F_{ext}} = 0$

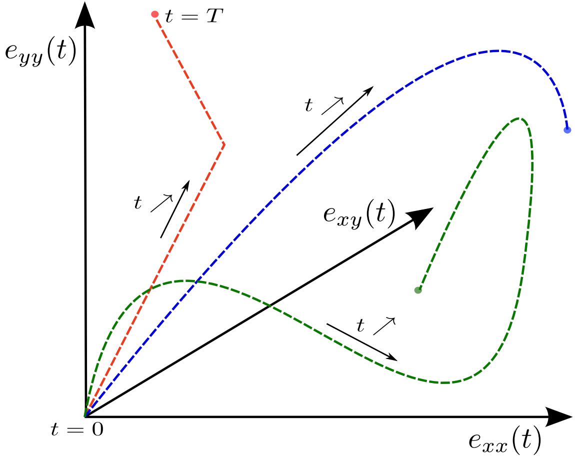
Reduced equations

- Reduced system after linearisation: $\min_{\underline{\boldsymbol{\alpha}}} \quad \|\mathbf{K}\boldsymbol{\Phi}\,\alpha + \mathbf{F_{ext}}\|$
- In the Galerkin framework: $\boldsymbol{\Phi}^T\,\mathbf{K}\boldsymbol{\Phi}\,\alpha + \boldsymbol{\Phi}^T\,\mathbf{F_{ext}} = 0$
- That's it! In the online stage, this much smaller system will be solved.

Outline

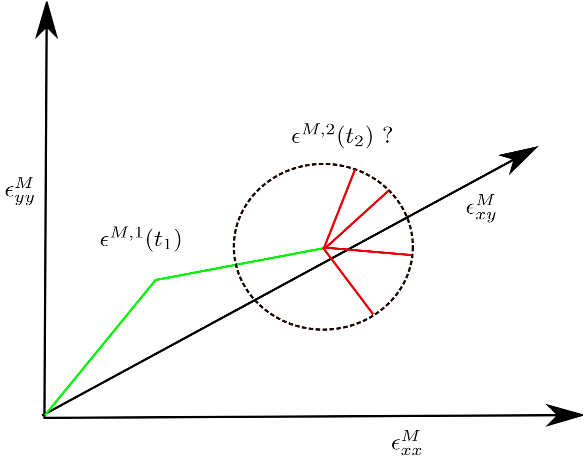
Arbitrary sampling unsatisfactory

Problem: parameter space is HUGE!
No guarantee that the arbitrary sampling ”explores“ the parameter space well enough



Load of worst prediction

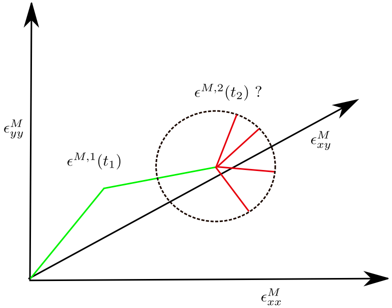
Rather than an arbitrary sampling, iteratively add the path of worst prediction:



Load of worst prediction

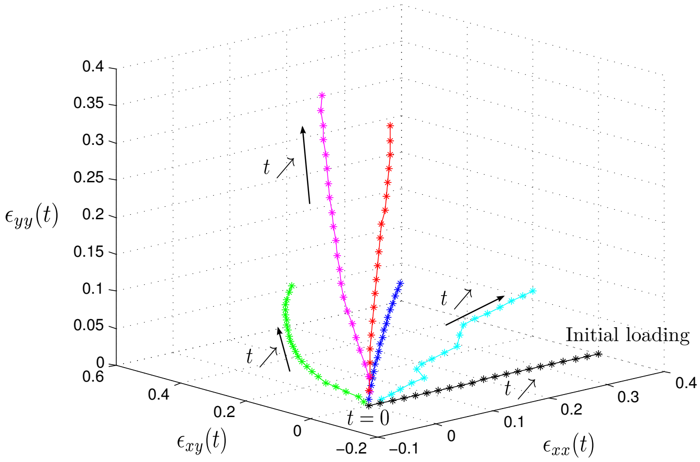
find the step increment $\Delta\epsilon^i$ that maximises:

$$\|u_{\text{exact}}(t_i, \epsilon(t_i) + \Delta\epsilon^i) - u_{\text{approx}}(t_i, \epsilon(t_i) + \Delta\epsilon^i)\|$$



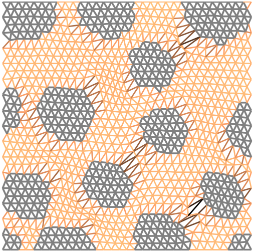
$$\epsilon(t_{i+1}) = \epsilon(t_i) + \Delta\epsilon^i_{\max}$$

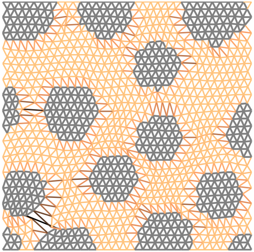
First paths generated

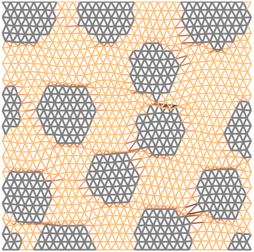


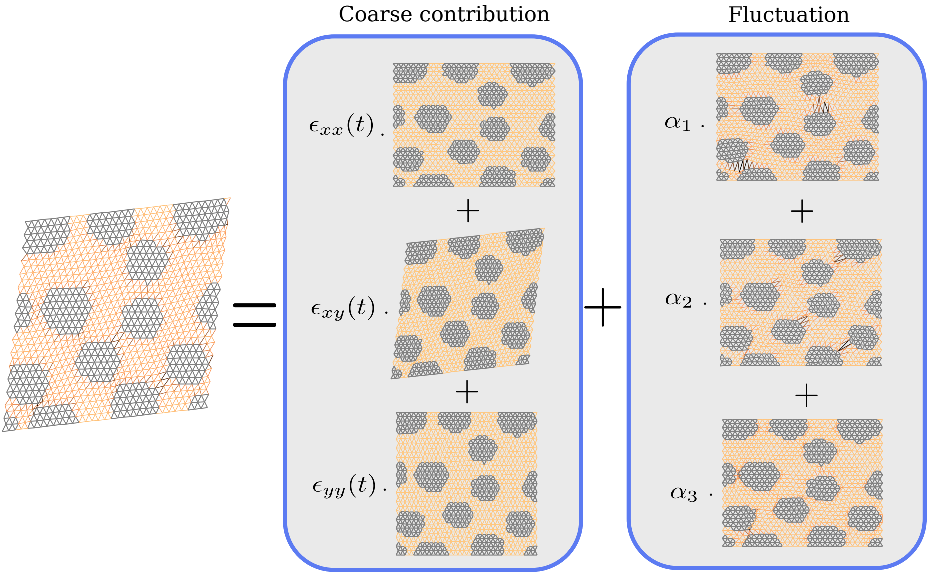
Example

Snapshot selection using the load of worst prediction algorithm
(36 load paths generated)
First 3 modes:









Coarse contribution
Fluctuation
$\epsilon_{xx}(t)$.
$\epsilon_{xy}(t)$.
$\epsilon_{yy}(t)$.
α_1 .
α_2 .
α_3 .

Is that good enough?

- Speed-up actually poor
- Equation "$\mathbf{\Phi}^T \mathbf{K}\mathbf{\Phi}\, \alpha + \mathbf{\Phi}^T \mathbf{F}_{\mathbf{ext}} = 0$" quicker to solve but $\mathbf{\Phi}^T \mathbf{K}\mathbf{\Phi}$ still expensive to evaluate
- Need to do something more $\Longrightarrow$ system approximation

Outline

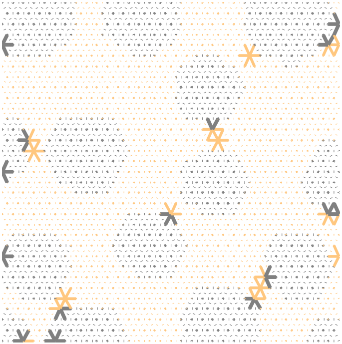
Introduction
- Heterogeneous materials
- Computational Homogenisation

Model order reduction in Computational Homogenisation
- Proper Orthogonal Decomposition (POD)
- Optimal snapshot selection
- System approximation
- Results

Conclusion

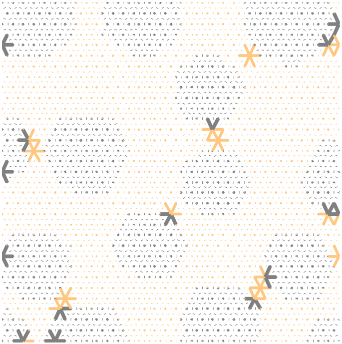
Idea

- Define a surrogate structure that retains only very few elements of the original one



Idea

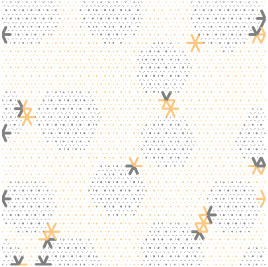
- Define a surrogate structure that retains only very few elements of the original one



- Reconstruct the operators using a second POD basis representing the internal forces

“Gappy“ technique

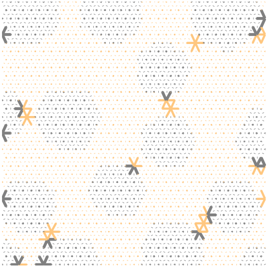
Originally used to reconstruct altered signals



- $\underline{\mathbf{F}}_{\mathbf{int}}(\mathbf{\Phi}\,\alpha)$ approximated by $\underline{\mathbf{F}}_{\mathbf{int}}(\mathbf{\Phi}\,\alpha) \approx \mathbf{\Psi}\,\beta$

"Gappy" technique

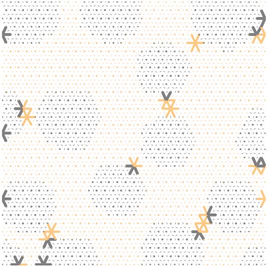
Originally used to reconstruct altered signals



- $\underline{\mathbf{F}}_{\text{int}}(\boldsymbol{\Phi}\,\alpha)$ approximated by $\underline{\mathbf{F}}_{\text{int}}(\boldsymbol{\Phi}\,\alpha) \approx \boldsymbol{\Psi}\,\boldsymbol{\beta}$
- $\underline{\mathbf{F}}_{\text{int}}(\boldsymbol{\Phi}\,\alpha)$ is evaluated exactly only on a few selected nodes: $\widehat{\underline{\mathbf{F}}_{\text{int}}(\boldsymbol{\Phi}\,\alpha)}$

“Gappy“ technique

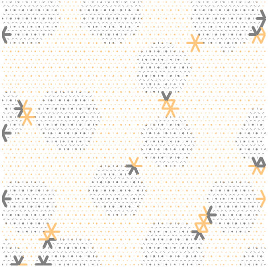
Originally used to reconstruct altered signals



- $\underline{\mathbf{F}}_{\mathbf{int}}\,(\boldsymbol{\Phi}\,\alpha)$ approximated by $\underline{\mathbf{F}}_{\mathbf{int}}\,(\boldsymbol{\Phi}\,\alpha) \approx \boldsymbol{\Psi}\,\beta$
- $\underline{\mathbf{F}}_{\mathbf{int}}\,(\boldsymbol{\Phi}\,\alpha)$ is evaluated exactly only on a few selected nodes: $\widehat{\underline{\mathbf{F}}_{\mathbf{int}}\,(\boldsymbol{\Phi}\,\alpha)}$
- β found through: $\min_{\boldsymbol{\beta}} \left\| \widehat{\boldsymbol{\Psi}}\,\beta - \widehat{\underline{\mathbf{F}}_{\mathbf{int}}\,(\boldsymbol{\Phi}\,\alpha)} \right\|_2$

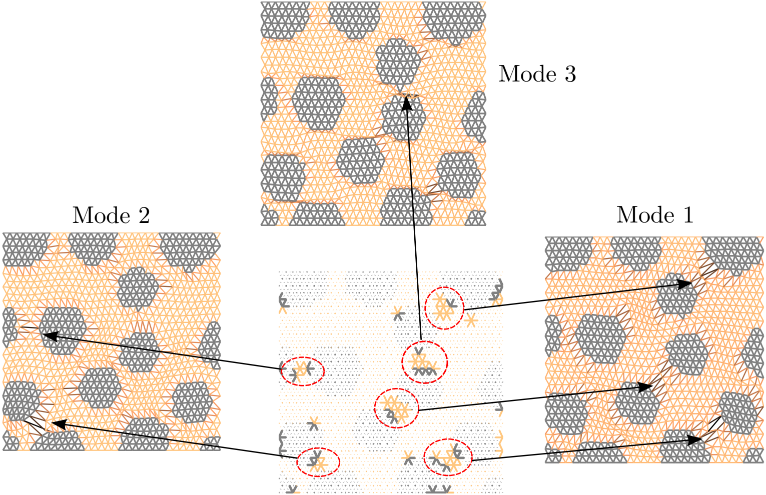
“Gappy“ technique

Originally used to reconstruct altered signals



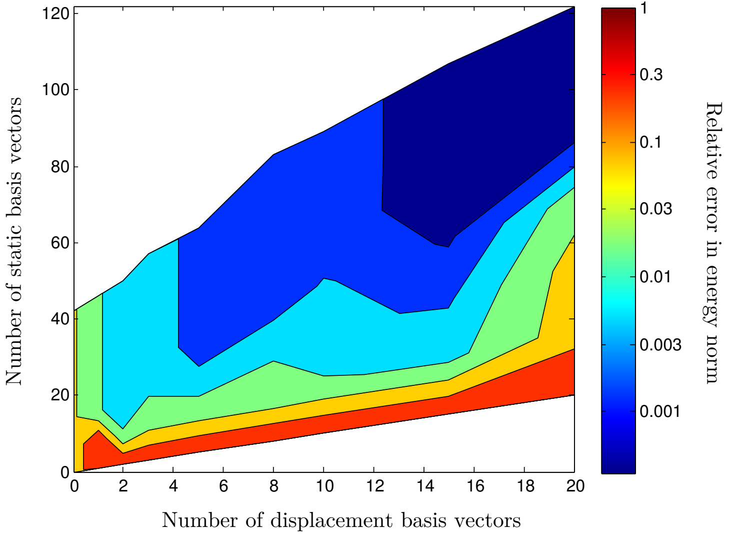
- $\underline{\mathbf{F}}_{\mathbf{int}}\,(\boldsymbol{\Phi}\,\boldsymbol{\alpha})$ approximated by $\underline{\mathbf{F}}_{\mathbf{int}}\,(\boldsymbol{\Phi}\,\boldsymbol{\alpha}) \approx \boldsymbol{\Psi}\,\boldsymbol{\beta}$
- $\underline{\mathbf{F}}_{\mathbf{int}}\,(\boldsymbol{\Phi}\,\boldsymbol{\alpha})$ is evaluated exactly only on a few selected nodes: $\widehat{\underline{\mathbf{F}}_{\mathbf{int}}\,(\boldsymbol{\Phi}\,\boldsymbol{\alpha})}$
- $\boldsymbol{\beta}$ found through: $\min_{\boldsymbol{\beta}} \left\| \widehat{\boldsymbol{\Psi}}\,\boldsymbol{\beta} - \widehat{\underline{\mathbf{F}}_{\mathbf{int}}\,(\boldsymbol{\Phi}\,\boldsymbol{\alpha})} \right\|_2$
- Selection of the controlled elements using DEIM

Controlled elements locations



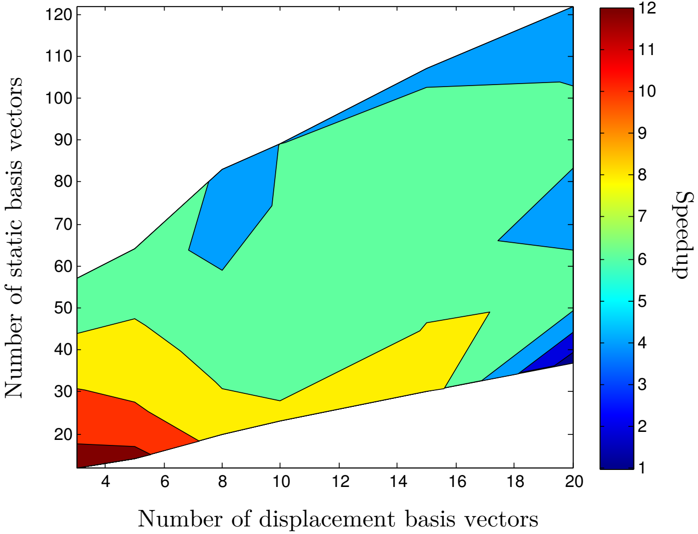
Outline

Error

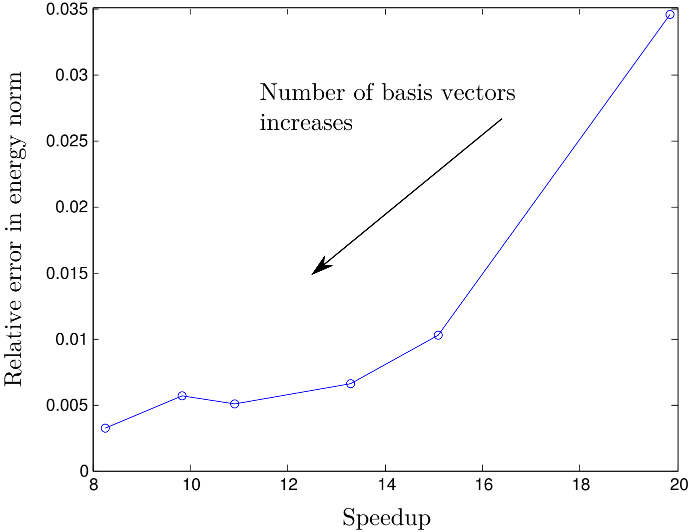


120
100
80
60
40
20
0
0
2
4
6
8
10
12
14
16
18
20
Number of static basis vectors
Number of displacement basis vectors
1
0.3
0.1
0.03
0.01
0.003
0.001
Relative error in energy norm

Speedup



Number of static basis vectors
120
110
100
90
80
70
60
50
40
30
20
4
6
8
10
12
14
16
18
20
Number of displacement basis vectors
12
11
10
9
8
7
6
5
4
3
2
1
Speedup



Number of basis vectors
increases
Relative error in energy norm
Speedup
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
8
10
12
14
16
18
20

Outline

Conclusion

- Model order reduction can be used to solved the RVE problem faster and with a reasonable accuracy
- An efficient snapshot selection algorithm can be developed when dealing with time-dependent parameters
- The controlled elements generated by the DEIM algorithm lie where damage is high
- Can be thought of as a bridge between analytical and computational homogenisation:
  the reduced bases are pseudo-analytical solutions of the RVE problem that is still computationally solved at very reduced cost

Thank you for your attention!