Algebraic coarse-graining methods in fracture mechanics: tackling local lack of correlation using domain decomposition

O. Goury, P. Kerfriden, S. P. A. Bordas
iMAM, Cardiff University

Outline

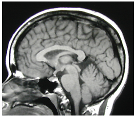
1. Introduction
   - Why model order reduction?
   - A straigthforward solution?
2. Partitioned POD method
   - Domain decomposition methods
   - System approximation
3. Results & Conclusion

Outline

Outline

Non-linear expensive simulations

- Problems depending on microscale phenomena $\Longrightarrow$ requires very fine mesh: expensive simulations
- **Surgical simulation**: real-time brain surgery simulation



- **Aeronautics**: advanced early-stage design

Projection-based model order reduction

We want to solve a parametrised mechanical problem:

$$\underbrace{\underline{\mathbf{F}}_{\text{int}}(\underline{\mathbf{U}}(\lambda),\lambda)}_{\text{Non-linear}} + \underline{\mathbf{F}}_{\text{ext}}(\lambda) = \underline{\mathbf{0}} \quad (1)$$

We are interested in the solution $\underline{\mathbf{U}}(\lambda)$ for many different values of λ.

Projection-based model order reduction assumption:

Solutions $\underline{\mathbf{U}}(\lambda)$ for different parameters λ are contained in a space of small dimension $span((\underline{\mathbf{C}}_i)_{i\in[\![1,n_c]\!]})$

Proper Orthogonal Decomposition (POD)

Look for $\underline{\mathbf{U}}$ as $\underline{\mathbf{U}} = \underline{\underline{\mathbf{C}}}\,\underline{\alpha}$. Where does the basis $\underline{\underline{\mathbf{C}}}$ comes from?

Proper Orthogonal Decomposition (POD)

Look for $\underline{\mathbf{U}}$ as $\underline{\mathbf{U}} = \underline{\underline{\mathbf{C}}}\,\underline{\alpha}$. Where does the basis $\underline{\underline{\mathbf{C}}}$ comes from?

- Solve the full problem a certain number of times varying the input parameter λ

Proper Orthogonal Decomposition (POD)

Look for $\underline{\mathbf{U}}$ as $\underline{\mathbf{U}} = \underline{\underline{\mathbf{C}}}\,\underline{\alpha}$. Where does the basis $\underline{\underline{\mathbf{C}}}$ comes from?

- Solve the full problem a certain number of times varying the input parameter λ
- You obtain a base of solutions (the snapshot): $(\underline{\mathbf{U}}_1, \underline{\mathbf{U}}_2, ..., \underline{\mathbf{U}}_{n_S}) = \underline{\underline{\mathbf{S}}}$

Proper Orthogonal Decomposition (POD)

Look for $\underline{\mathbf{U}}$ as $\underline{\mathbf{U}} = \underline{\underline{\mathbf{C}}}\,\underline{\alpha}$. Where does the basis $\underline{\underline{\mathbf{C}}}$ comes from?

- Solve the full problem a certain number of times varying the input parameter λ
- You obtain a base of solutions (the snapshot): $(\underline{\mathbf{U}}_1, \underline{\mathbf{U}}_2, ..., \underline{\mathbf{U}}_{n_S}) = \underline{\underline{\mathbf{S}}}$
- Extract the essence of this snapshot space using the singular value decomposition: gives the POD basis $\underline{\underline{\mathbf{C}}}$

Proper Orthogonal Decomposition (POD)

Look for $\underline{\mathbf{U}}$ as $\underline{\mathbf{U}} = \underline{\underline{\mathbf{C}}}\,\underline{\alpha}$. Where does the basis $\underline{\underline{\mathbf{C}}}$ comes from?

- Solve the full problem a certain number of times varying the input parameter λ
- You obtain a base of solutions (the snapshot): $(\underline{\mathbf{U}}_1, \underline{\mathbf{U}}_2, ..., \underline{\mathbf{U}}_{n_S}) = \underline{\underline{\mathbf{S}}}$
- Extract the essence of this snapshot space using the singular value decomposition: gives the POD basis $\underline{\underline{\mathbf{C}}}$
- Reduced system: $\min_{\underline{\alpha}} \quad \left\| \underline{\mathbf{F}}_{\mathbf{int}} \left(\underline{\underline{\mathbf{C}}}\,\underline{\alpha} \right) + \underline{\mathbf{F}}_{\mathbf{ext}} \right\|$

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

Proper Orthogonal Decomposition (POD)

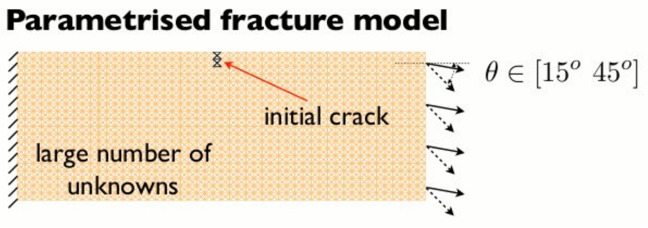
Look for $\underline{\mathbf{U}}$ as $\underline{\mathbf{U}} = \underline{\underline{\mathbf{C}}}\,\underline{\alpha}$. Where does the basis $\underline{\underline{\mathbf{C}}}$ comes from?

- Solve the full problem a certain number of times varying the input parameter λ
- You obtain a base of solutions (the snapshot): $(\underline{\mathbf{U}}_1, \underline{\mathbf{U}}_2, ..., \underline{\mathbf{U}}_{n_S}) = \underline{\underline{\mathbf{S}}}$
- Extract the essence of this snapshot space using the singular value decomposition: gives the POD basis $\underline{\underline{\mathbf{C}}}$
- Reduced system: $\min_{\underline{\alpha}} \quad \left\| \underline{\mathbf{F}}_{\mathbf{int}} \left(\underline{\underline{\mathbf{C}}}\,\underline{\alpha} \right) + \underline{\mathbf{F}}_{\mathbf{ext}} \right\|$
- In the Galerkin framework: $\underline{\underline{\mathbf{C}}}^T\, \underline{\mathbf{F}}_{\mathbf{int}} \left(\underline{\underline{\mathbf{C}}}\,\underline{\alpha} \right) + \underline{\underline{\mathbf{C}}}^T\, \underline{\mathbf{F}}_{\mathbf{ext}} = 0$

Outline

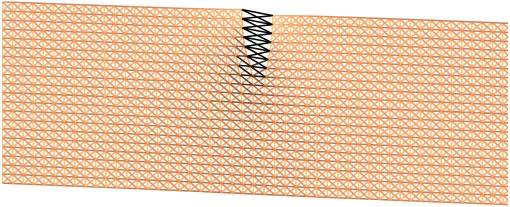
1. Introduction
   - Why model order reduction?
   - A straigthforward solution?
2. Partitioned POD method
   - Domain decomposition methods
   - System approximation
3. Results & Conclusion

Example

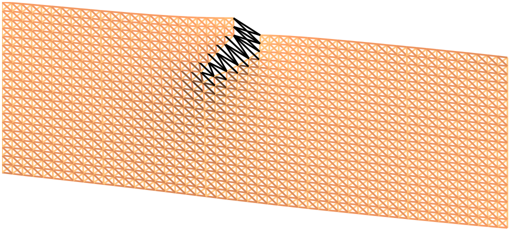

Parametrised fracture model
initial crack
large number of unknowns
$\theta \in [15^o \; 45^o]$

Snapshots

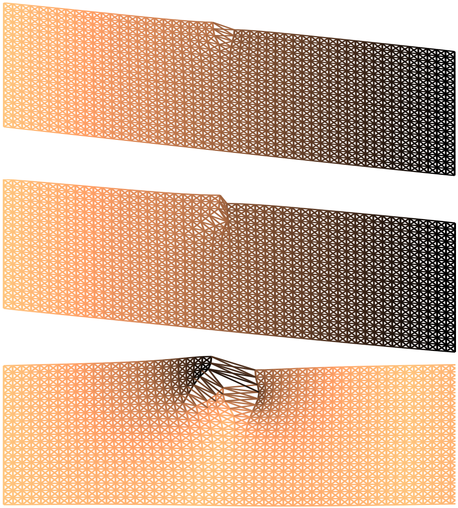
15 degrees:



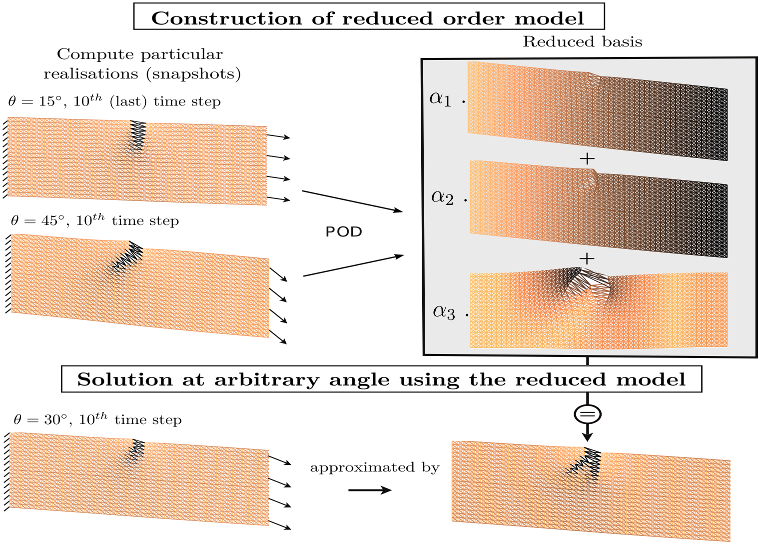
45 degrees:



First 3 modes of the POD basis



Fracture not well captured



What can we do?

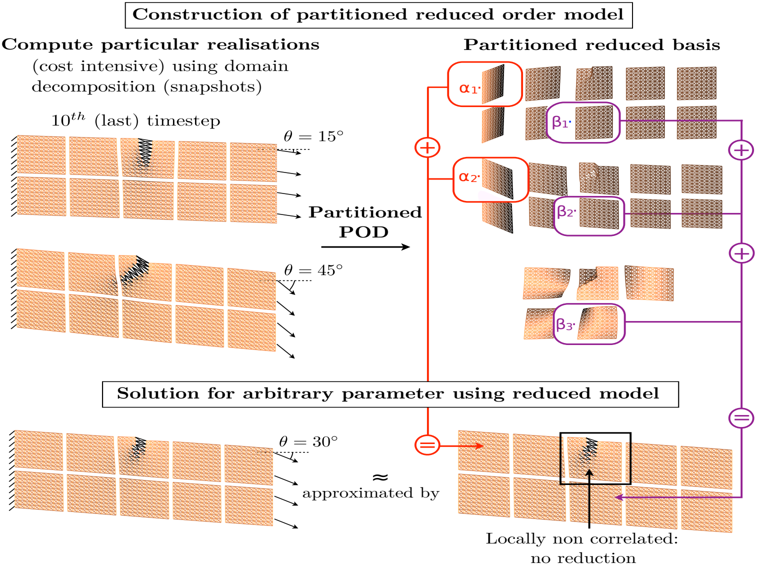
Idea: juste divide up the domain and select regions that are “reducible”

Outline

1. Introduction
   - Why model order reduction?
   - A straigthforward solution?
2. Partitioned POD method
   - Domain decomposition methods
   - System approximation
3. Results & Conclusion

Outline

1. Introduction
   - Why model order reduction?
   - A straigthforward solution?
2. Partitioned POD method
   - Domain decomposition methods
   - System approximation
3. Results & Conclusion


Construction of partitioned reduced order model
Compute particular realisations (cost intensive) using domain decomposition (snapshots)
10th (last) timestep
$\theta = 15°$
$\theta = 45°$
Partitioned POD
Partitioned reduced basis
$\alpha_1 \cdot$
$\beta_1 \cdot$
$\alpha_2 \cdot$
$\beta_2 \cdot$
$\beta_3 \cdot$
Solution for arbitrary parameter using reduced model
$\theta = 30°$
≈
approximated by
Locally non correlated: no reduction

Is that good enough?

- Speed-up actually poor
- Equation “$\underline{\underline{\mathbf{C}}}^T\, \underline{\mathbf{F}}_{\mathbf{int}}\left(\underline{\underline{\mathbf{C}}}\, \underline{\alpha}\right) + \underline{\underline{\mathbf{C}}}^T\, \underline{\mathbf{F}}_{\mathbf{ext}} = 0$“ quicker to solve but $\underline{\underline{\mathbf{C}}}^T\, \underline{\mathbf{F}}_{\mathbf{int}}\left(\underline{\underline{\mathbf{C}}}\, \underline{\alpha}\right)$ still expensive to evaluate
- Need to do something more $\Longrightarrow$ system approximation

Outline

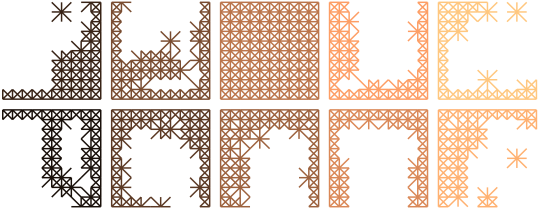
1. Introduction
   - Why model order reduction?
   - A straigthforward solution?
2. Partitioned POD method
   - Domain decomposition methods
   - System approximation
3. Results & Conclusion

Idea

- Integrate only over some nodes of the domain
- Reconstruct the operators using a second POD basis

“Gappy“ technique

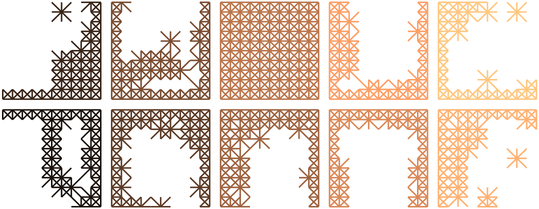
Originally used to reconstruct images



- $\underline{\mathbf{F}}_{\text{int}}\left(\underline{\underline{\mathbf{C}}}\,\underline{\alpha}\right)$ approximated by $\widetilde{\underline{\mathbf{F}}_{\text{int}}\left(\underline{\underline{\mathbf{C}}}\,\underline{\alpha}\right)} = \underline{\underline{\mathbf{D}}}\,\underline{\beta}$

"Gappy" technique

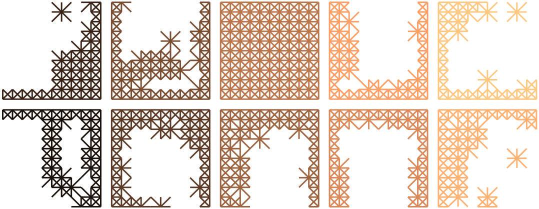
Originally used to reconstruct images



- $\underline{\mathbf{F}}_{\text{int}}\left(\underline{\underline{\mathbf{C}}}\,\underline{\alpha}\right)$ approximated by $\widetilde{\underline{\mathbf{F}}_{\text{int}}\left(\underline{\underline{\mathbf{C}}}\,\underline{\alpha}\right)} = \underline{\underline{\mathbf{D}}}\,\underline{\beta}$
- $\underline{\mathbf{F}}_{\text{int}}\left(\underline{\underline{\mathbf{C}}}\,\underline{\alpha}\right)$ is evaluated exactly only on a few selected nodes: $\widehat{\underline{\mathbf{F}}_{\text{int}}\left(\underline{\underline{\mathbf{C}}}\,\underline{\alpha}\right)}$

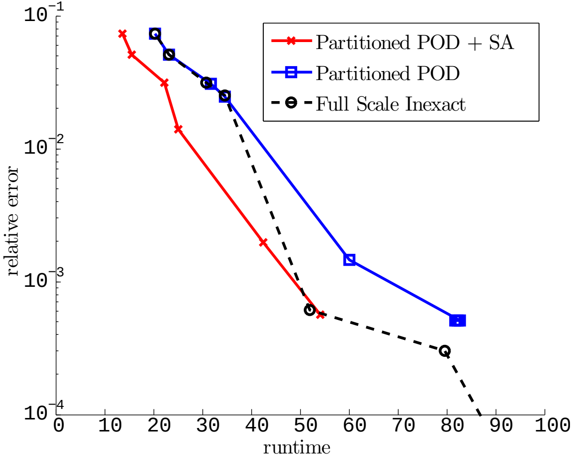
“Gappy“ technique

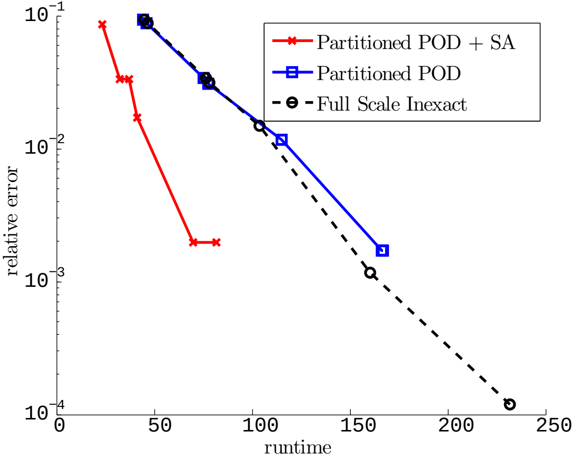
Originally used to reconstruct images

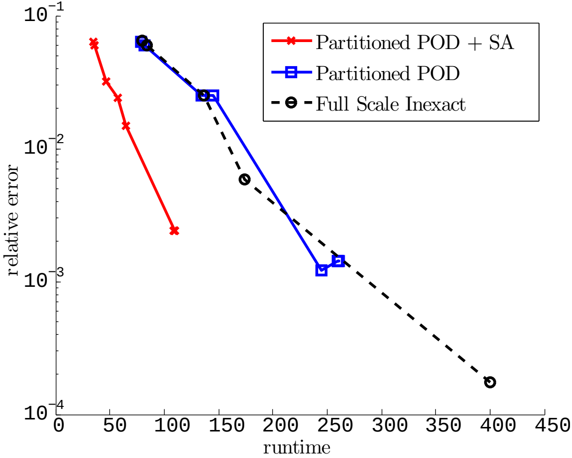


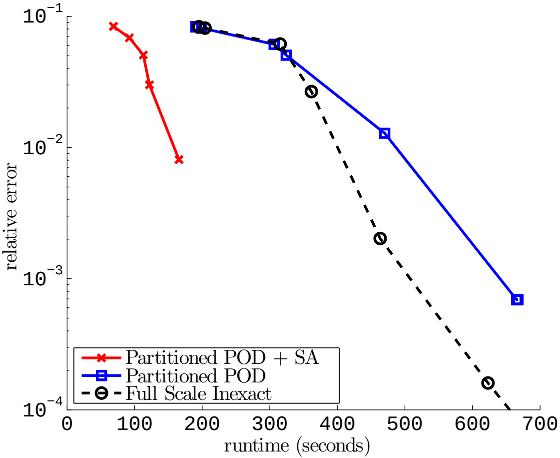
- $\underline{\mathbf{F}}_{\text{int}}\left(\underline{\underline{\mathbf{C}}}\,\underline{\alpha}\right)$ approximated by $\widetilde{\underline{\mathbf{F}}_{\text{int}}\left(\underline{\underline{\mathbf{C}}}\,\underline{\alpha}\right)} = \underline{\underline{\mathbf{D}}}\,\underline{\beta}$
- $\underline{\mathbf{F}}_{\text{int}}\left(\underline{\underline{\mathbf{C}}}\,\underline{\alpha}\right)$ is evaluated exactly only on a few selected nodes: $\widehat{\underline{\mathbf{F}}_{\text{int}}\left(\underline{\underline{\mathbf{C}}}\,\underline{\alpha}\right)}$
- $\underline{\beta}$ found through: $\min\limits_{\underline{\beta}} \left\| \widehat{\underline{\underline{\mathbf{D}}}}\,\underline{\beta} - \widehat{\underline{\mathbf{F}}_{\text{int}}\left(\underline{\underline{\mathbf{C}}}\,\underline{\alpha}\right)} \right\|_2$

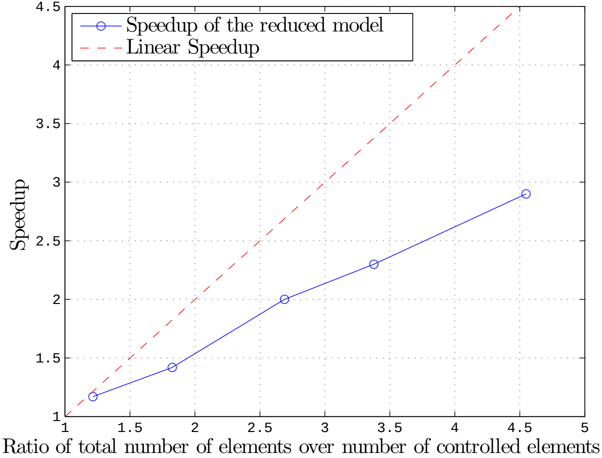
Outline


Partitioned POD + SA
Partitioned POD
Full Scale Inexact
relative error
runtime
10^{-1}
10^{-2}
10^{-3}
10^{-4}
0
10
20
30
40
50
60
70
80
90
100


Partitioned POD + SA
Partitioned POD
Full Scale Inexact
relative error
runtime
10^{-1}
10^{-2}
10^{-3}
10^{-4}
0
50
100
150
200
250


Partitioned POD + SA
Partitioned POD
Full Scale Inexact
relative error
runtime
10^{-1}
10^{-2}
10^{-3}
10^{-4}
0
50
100
150
200
250
300
350
400
450


relative error
10^{-1}
10^{-2}
10^{-3}
10^{-4}
0
100
200
300
400
500
600
700
runtime (seconds)
Partitioned POD + SA
Partitioned POD
Full Scale Inexact


Speedup of the reduced model
Linear Speedup
Speedup
Ratio of total number of elements over number of controlled elements
1
1.5
2
2.5
3
3.5
4
4.5
5

Thank you for your attention!