CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

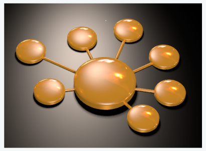
UNIVERSITÉ DU LUXEMBOURG

Legato Team
Computational Mechanics

Multiscale Fracture, CAD and Image as a Model

RealTCut erc

Stéphane P.A. BORDAS (1,2),
Pierre KERFRIDEN (2), Lars BEEX (1),
Elena ATROSHCHENKO (3), Karol MILLER (4)
(1) University of Luxembourg
(2) Cardiff University
(3) University of Chile
(4) University of Western Australia



The Royal Academy of Engineering

The Leverhulme Trust

Inria
INVENTEURS DU MONDE NUMÉRIQUE

(short!) history of the Legato research group

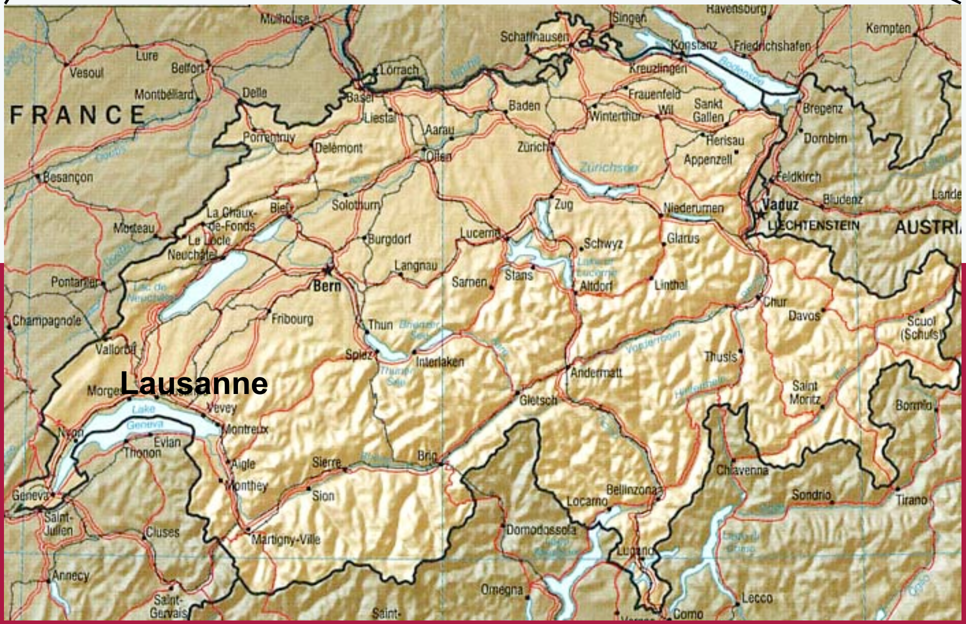
Chicago, Illinois

uni.lu
UNIVERSITÉ DU LUXEMBOURG

Advisor: Brian Moran

MSc Geotechnical Engineering (time domain reflectometry)

PhD. Damage Tolerance of Aerospace Structures (XFEM) and Biofilm Growth



UNIVERSITÉ DU LUXEMBOURG

Post-doc 2003-2006 -

Meshless/XFEM Geomechanics

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

ATLANTIC OCEAN
Orkney
Thurso
Wick
NORTH SEA
Harris
Benbecula
Inverness
Peterhead
SCOTLAND
Aberdeen
Fort William
Dundee
Edinburgh
Glasgow
Lanark
Ayr
Newcastle Upon Tyne
Dumfries
NORTHERN IRELAND
ENGLAND
Belfast
©World Guides

uni.lu
UNIVERSITÉ DU LUXEMBOURG

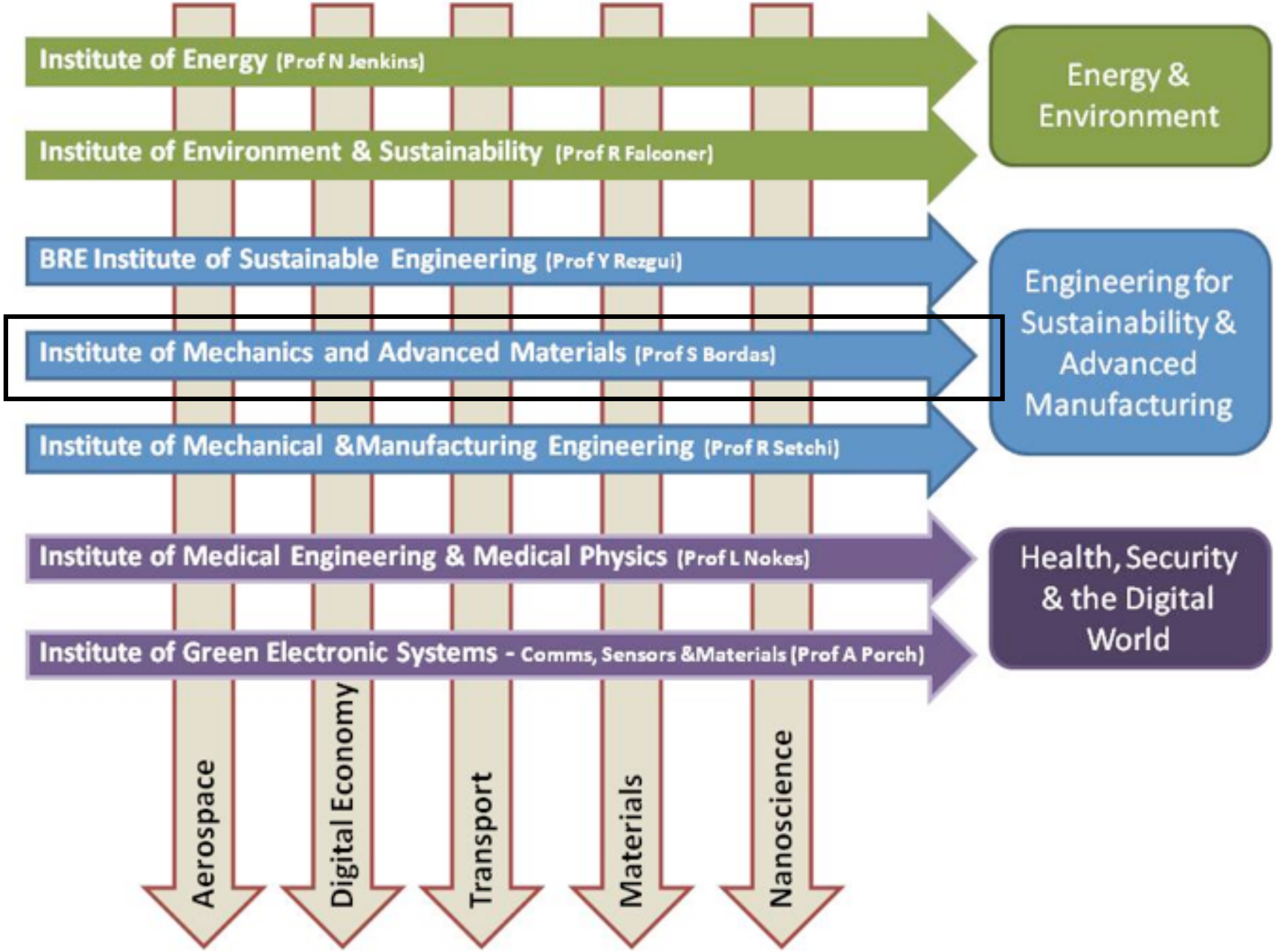
Lecturer in civil engineering /Assistant Professor

VIA VERITAS VITA
UNIVERSITY *of* GLASGOW

CARDIFF
UNIVERSITY
PRIFYSGOL
CAERDYDD

uni.lu
UNIVERSITÉ DU
LUXEMBOURG

CARDIFF
BLUES



Institute of Energy (Prof N Jenkins)

Institute of Environment & Sustainability (Prof R Falconer)

BRE Institute of Sustainable Engineering (Prof Y Rezgui)

Institute of Mechanics and Advanced Materials (Prof S Bordas)

Institute of Mechanical &Manufacturing Engineering (Prof R Setchi)

Institute of Medical Engineering & Medical Physics (Prof L Nokes)

Institute of Green Electronic Systems - Comms, Sensors &Materials (Prof A Porch)

Energy & Environment

Engineering for Sustainability & Advanced Manufacturing

Health, Security & the Digital World

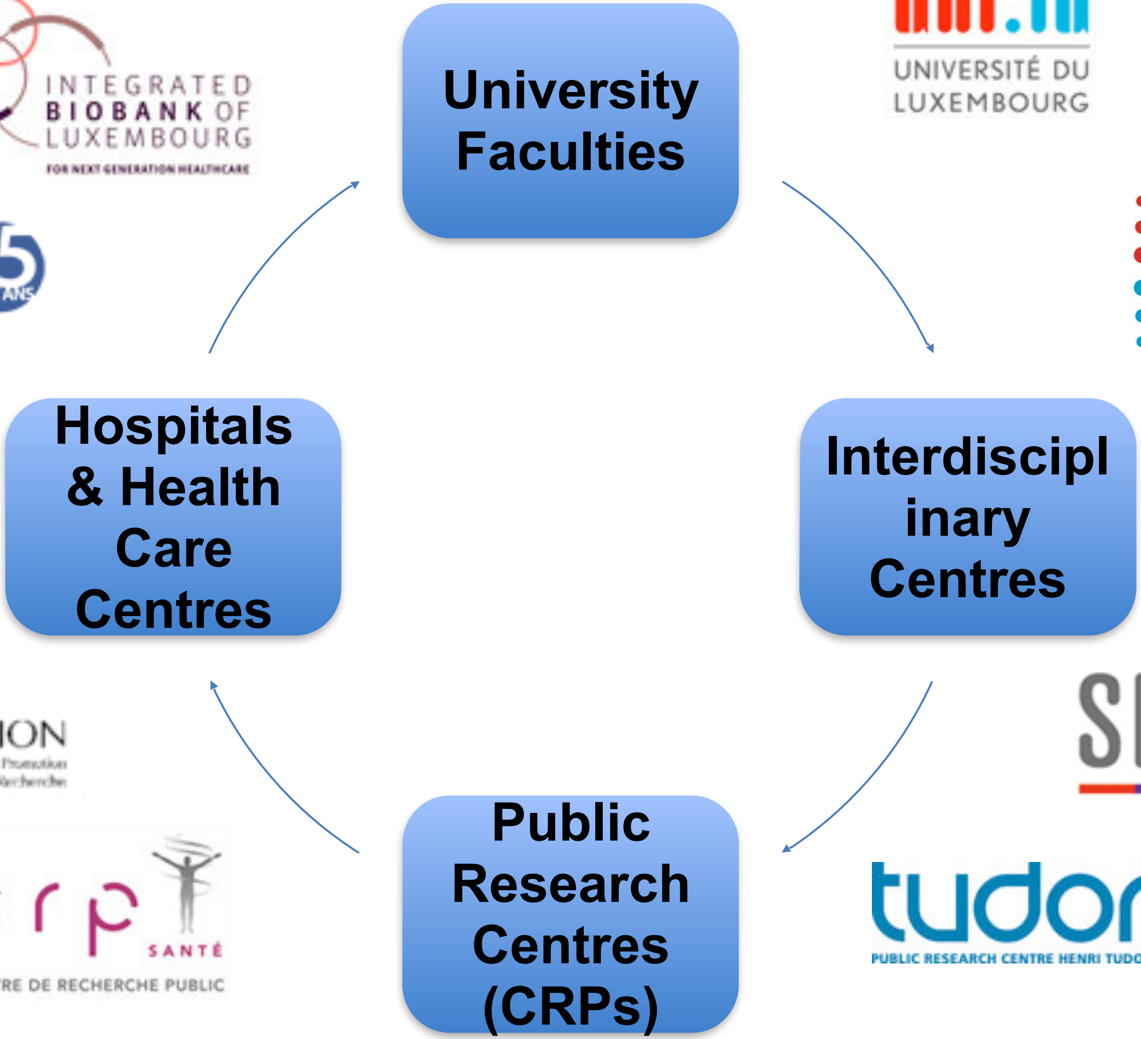
CANADA
RUSSIA
ICELAND
NORWAY
SWEDEN
FINLAND
BELARUS
POLAND
GERMANY
UKRAINE
FRANCE
TED STATES OF AMERICA
SPAIN
ITALY
TURKEY
KAZAKHSTAN
MONGOLIA
CHINA
IRAN
IRAQ
SYRIA
MEXICO
ALGERIA
LIBYA
EGYPT
SAUDI ARABIA
INDIA
MALI
NIGER
CHAD
SUDAN
ERITREA
YEMEN
GUINEA
NIGERIA
CENTRAL AFRICAN REPUBLIC
ETHIOPIA
SOMALIA
UGANDA
KENYA
DEMOCRATIC REPUBLIC OF THE CONGO
TANZANIA
THAILAND
VENEZUELA
COLOMBIA
PERU
BRAZIL
BOLIVIA
ANGOLA
ZAMBIA
NAMIBIA
BOTSWANA
SOUTH AFRICA
ARGENTINA

The Scientific Infrastructure in Luxembourg

uni.lu

UNIVERSITÉ DU LUXEMBOURG

Scientific landscape in Luxembourg


University Faculties

Interdisciplinary Centres

Public Research Centres (CRPs)

Hospitals & Health Care Centres

INTEGRATED BIOBANK OF LUXEMBOURG
FOR NEXT GENERATION HEALTHCARE

35 ANS

LUXINNOVATION
Agence Nationale pour la Promotion de l'Innovation et de la Recherche

crp SANTÉ
CENTRE DE RECHERCHE PUBLIC

uni.lu
UNIVERSITÉ DU LUXEMBOURG

LCSB

SnT

tudor
PUBLIC RESEARCH CENTRE HENRI TUDOR

Centre de Recherche Public
Gabriel Lippmann

The University of Luxembourg (UL)

The Faculties

Faculty of Science, Technology and Communication

- focuses on informatics, engineering, mathematics, life sciences, physics and material science

Faculty of Law, Economics and Finance

- Exceptional academic programs in Law, Economics and Finance
- Established as cooperation of CRP-Gabriel Lippmann and Luxembourg School of Finance

Faculty of Language and Literature, Humanities, Arts and Education

- Mission to study and accompany the development of the society in its social, economic, cultural, political and educational aspects

Computational mechanics & computational materials sciences Multiscale/field interface problems

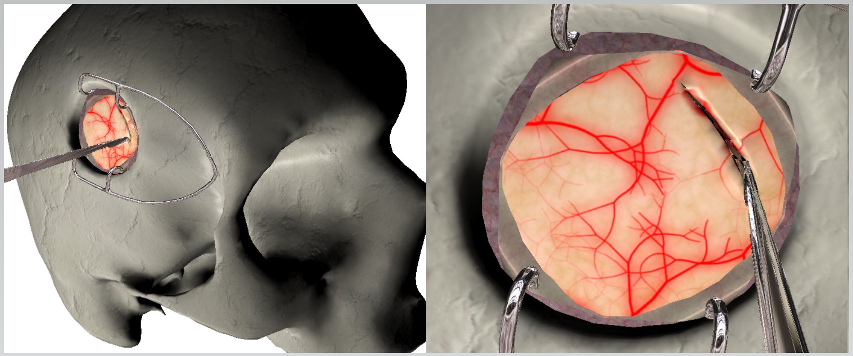
COMPETENCES

DISCRETISATION
discrete and continuum approaches

MULTI-SCALE FRACTURE
aerospace composites, polycrystalline materials

COUPLED PROBLEMS
biofilms, liquid crystals, fluid-structure, batteries

QUALITY & ERROR CONTROL
optimise computational time given an accuracy level



Real-time simulation of cutting during brain surgery

INTERACTIVITY
Reduce computational costs by several orders of magnitude

APPLICATIONS

PERSONALISED MEDICINE

Computer-aided surgery

Computer-aided diagnostics

ENGINEERING

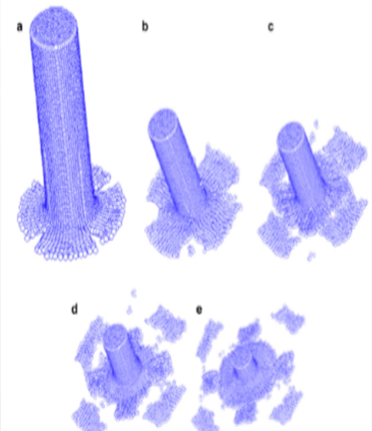
Durability & Sustainability

Energy

Aerospace

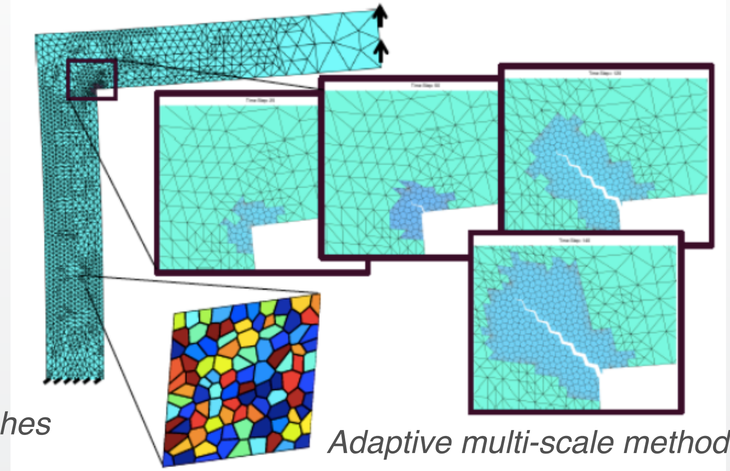
http://legato-team.eu — stephane.bordas@uni.lu S. Bordas, B. Peters, A. Zilian and S. Belouettar (LIST)

Discretisation



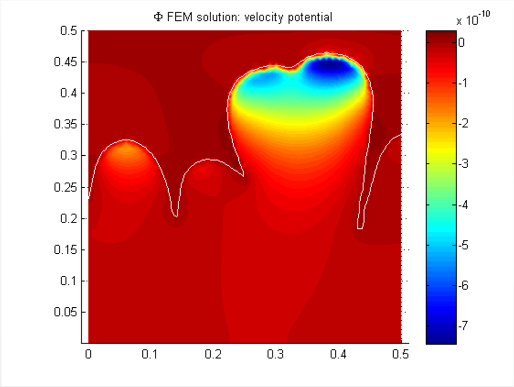
Mesh-free and discrete approaches to fracture

Fracture over multiple scales



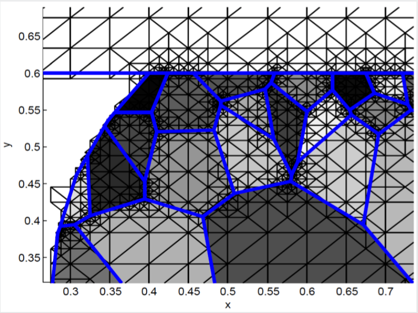
Adaptive multi-scale methods

Coupled problems

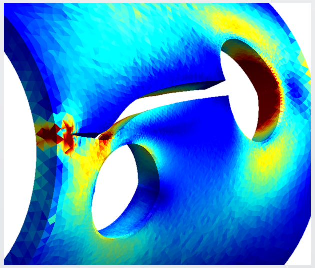


Biofilm growth

Quality and error control

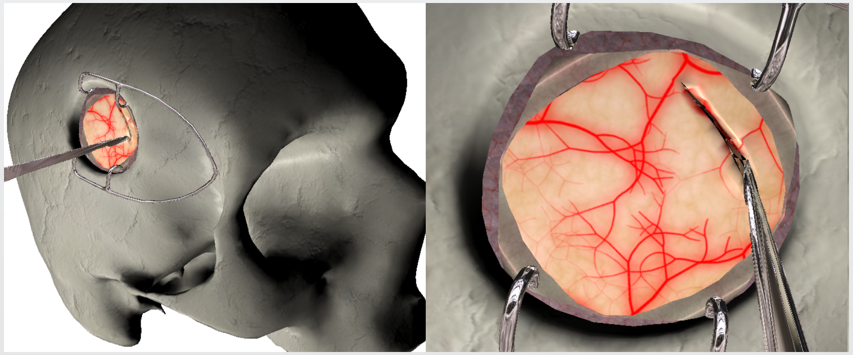


Durability of Pb-free solders



Error estimates for fracture

Interactivity and model order reduction



APPLICATIONS

Personalised Medicine

- *Computer-aided surgery*
- *Computer-aided diagnostics*

Engineering

- *Durability & Sustainability*
- *Energy*
- *Aerospace*

Research Strategy of the Legato Computational Mechanics Group

TOOLS

BAYESIAN INFERENCE

STATISTICAL INVERSE PROBLEMS

MODEL LEARNING

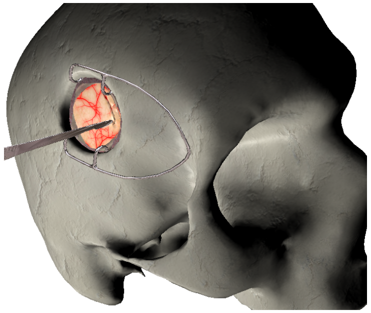
Leverage the commonality of mathematical formulations across various problem domains

APPLICATIONS

Personalised Medicine

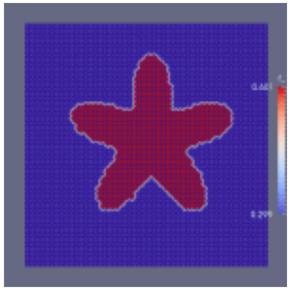
Engineering

Computer-aided surgery



Patient-specific simulations

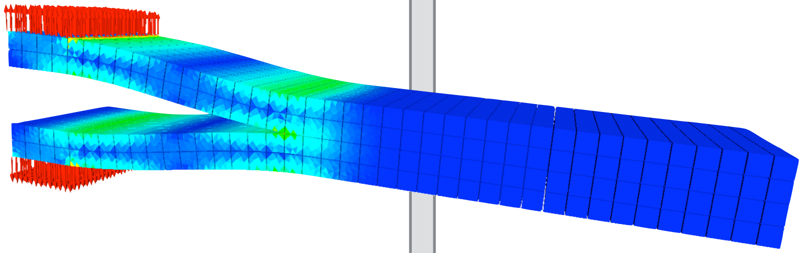
Computer-aided diagnostics



Tumour detection - inverse problem

Structural health monitoring

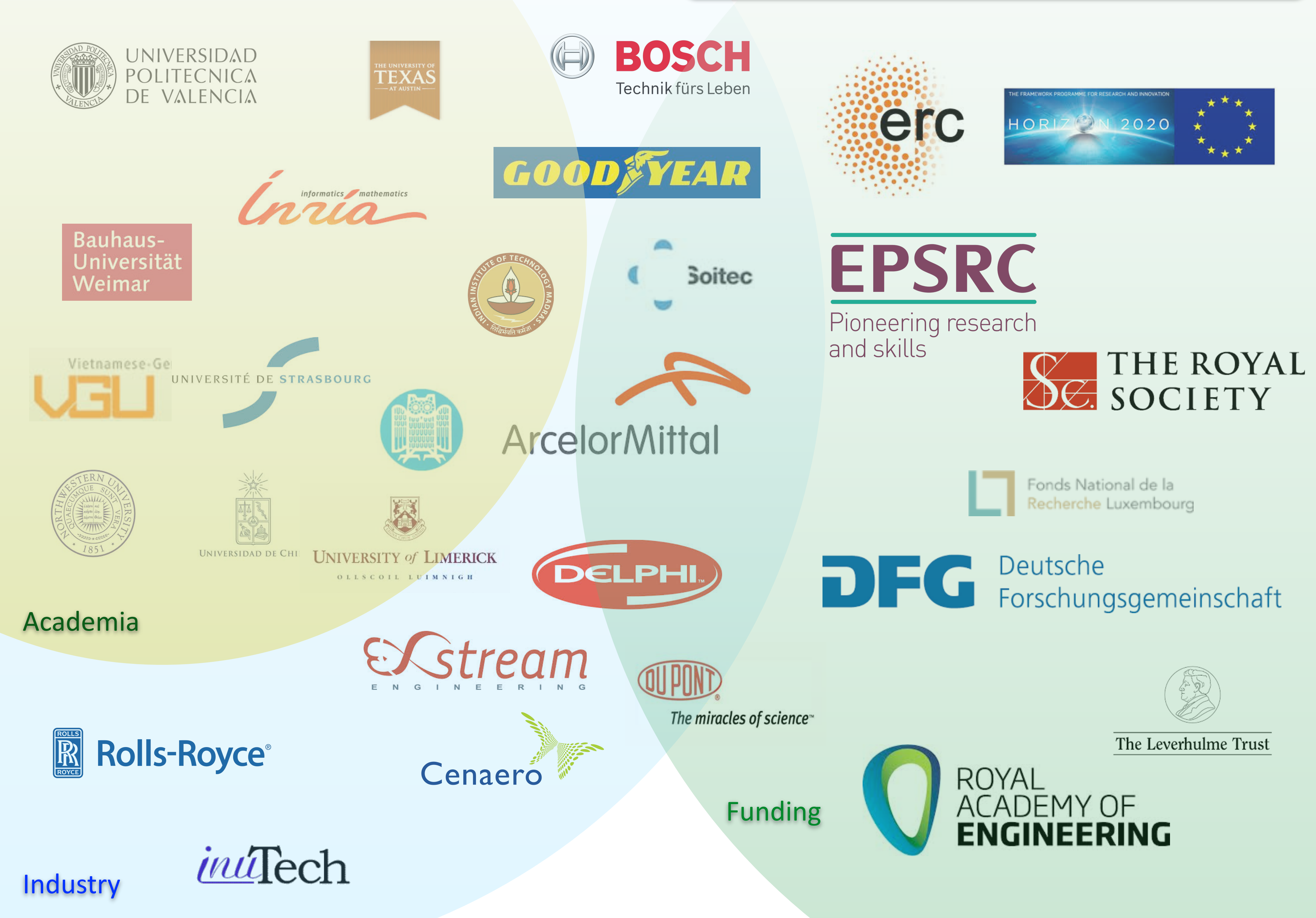
Adaptive "smart" structures



Crack detection and self-healing structures

PARTNERS

FUNDERS 8 million euros since 2006



UNIVERSIDAD POLITECNICA DE VALENCIA

THE UNIVERSITY OF TEXAS AT AUSTIN

BOSCH Technik fürs Leben

erc

THE FRAMEWORK PROGRAMME FOR RESEARCH AND INNOVATION HORIZON 2020

Inria informatics mathematics

GOODYEAR

Bauhaus-Universität Weimar

Soitec

Indian Institute of Technology Madras

EPSRC Pioneering research and skills

Vietnamese-Ge VGU

UNIVERSITÉ DE STRASBOURG

ArcelorMittal

THE ROYAL SOCIETY

Fonds National de la Recherche Luxembourg

Northwestern University 1851

Universidad de Chi

University of Limerick Ollscoil Luimnigh

DELPHI

DFG Deutsche Forschungsgemeinschaft

Academia

Xstream ENGINEERING

DUPONT The miracles of science

The Leverhulme Trust

Rolls-Royce

Cenaero

Funding

ROYAL ACADEMY OF ENGINEERING

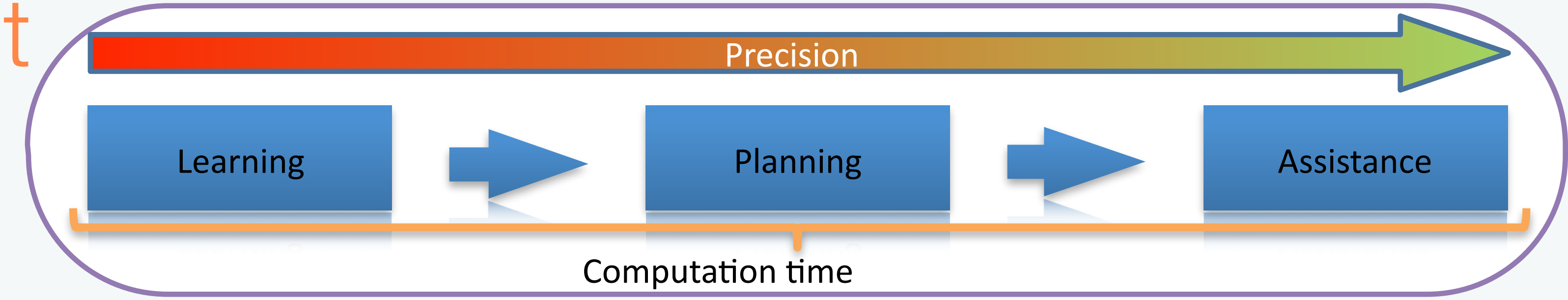
Industry

iniTech

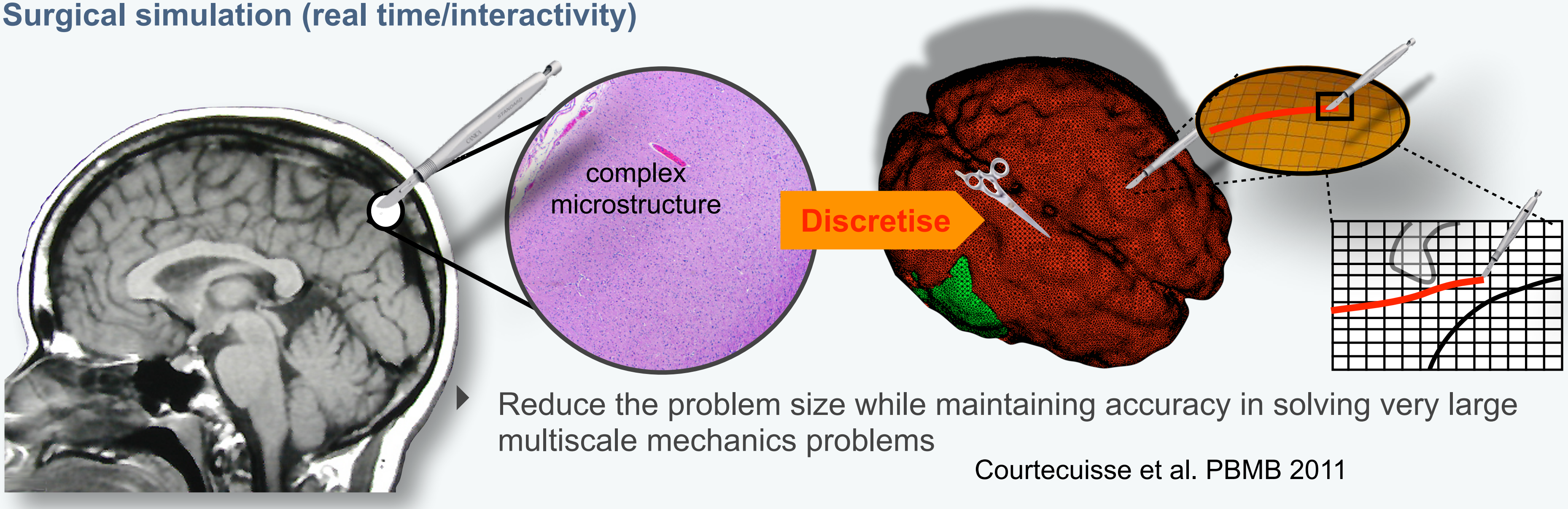
Some motivation

Motivation (1/2) - generate and solve models of patients: FAST!

RealTcut

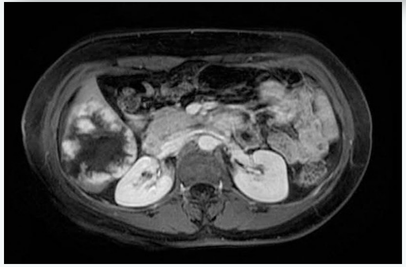
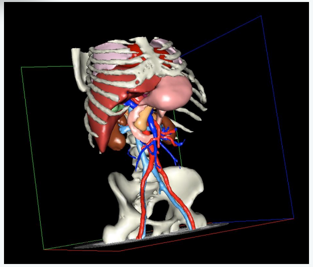
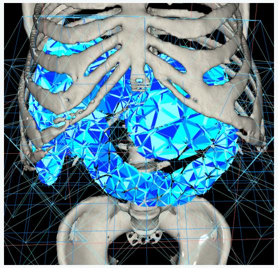
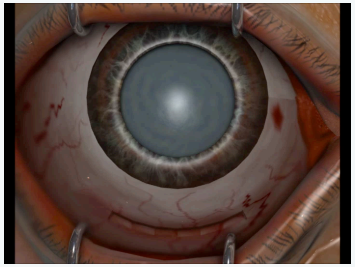
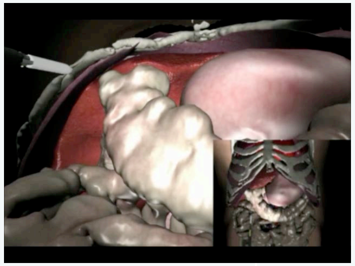


Surgical simulation (real time/interactivity)

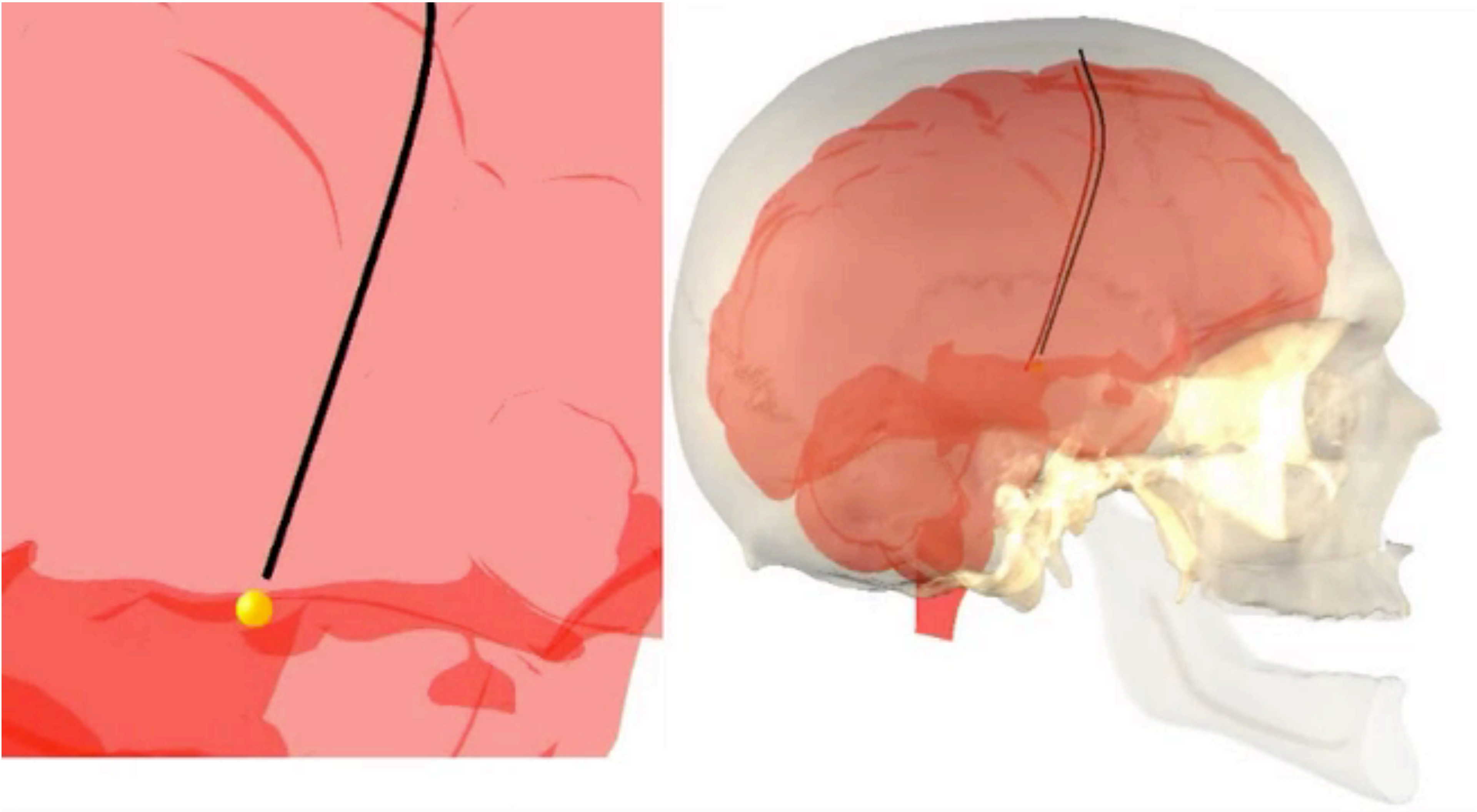


- Reduce the problem size while maintaining accuracy in solving very large multiscale mechanics problems

Courtecuisse et al. PBMB 2011

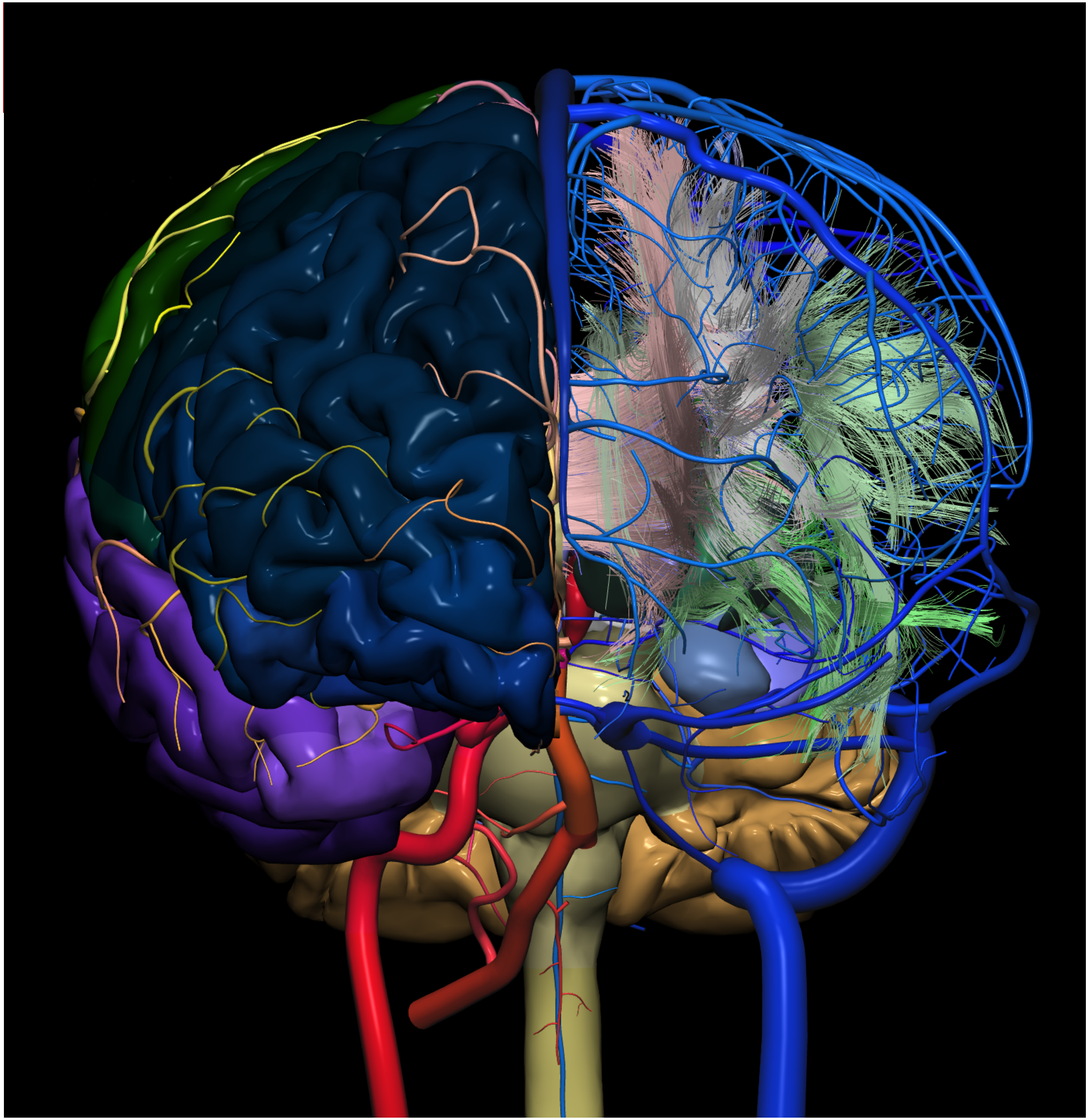






…not exactly brain surgery



Deep-brain stimulation

Courtesy Alexandre Bilger, PhD thesis, Inria, 2014



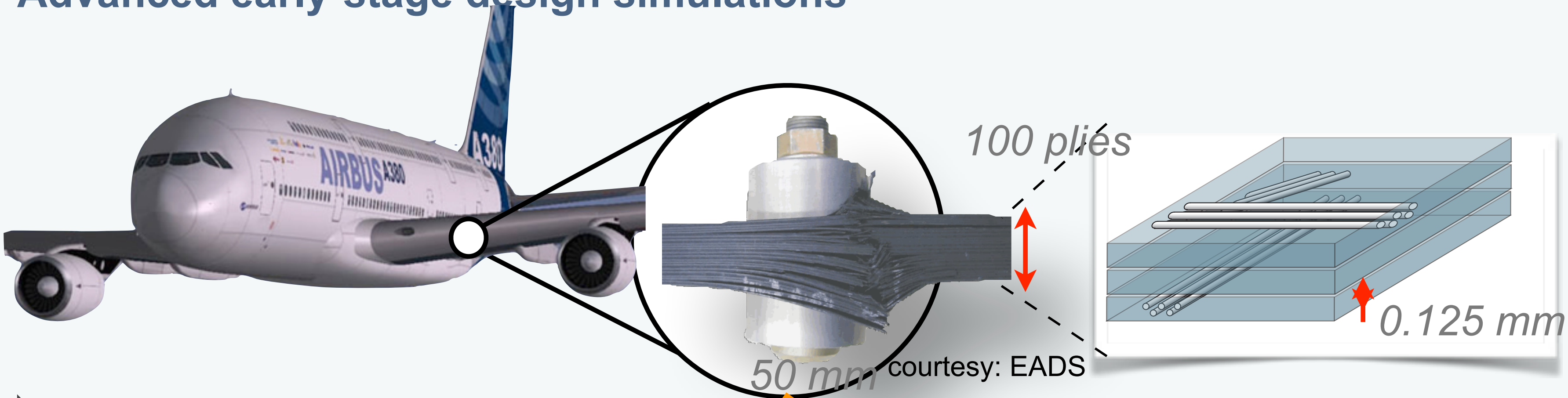
The brain is complicated…

But we only wish to compute displacements

Courtesy
Prof. Wies Nowinski,
A-Star, Singapore

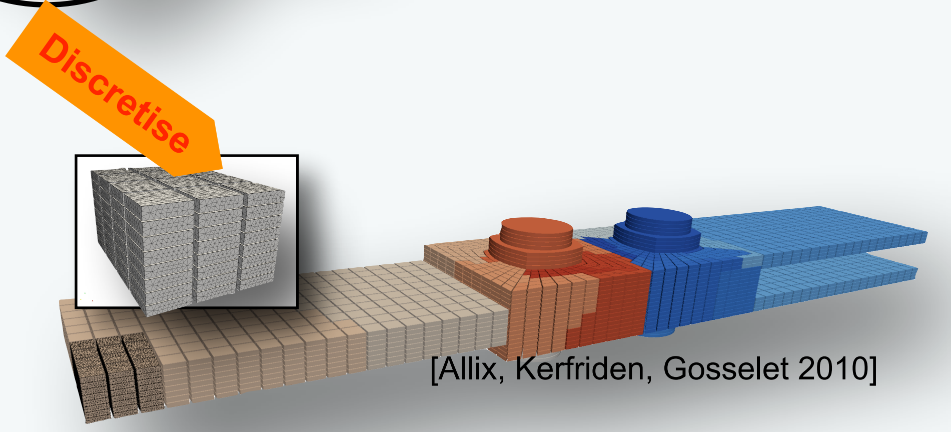
Motivation (2/2) - similar problems occur in aerospace

Advanced early-stage design simulations



courtesy: EADS

- Large gradients
- Explicit mesostructure description



[Allix, Kerfriden, Gosselet 2010]

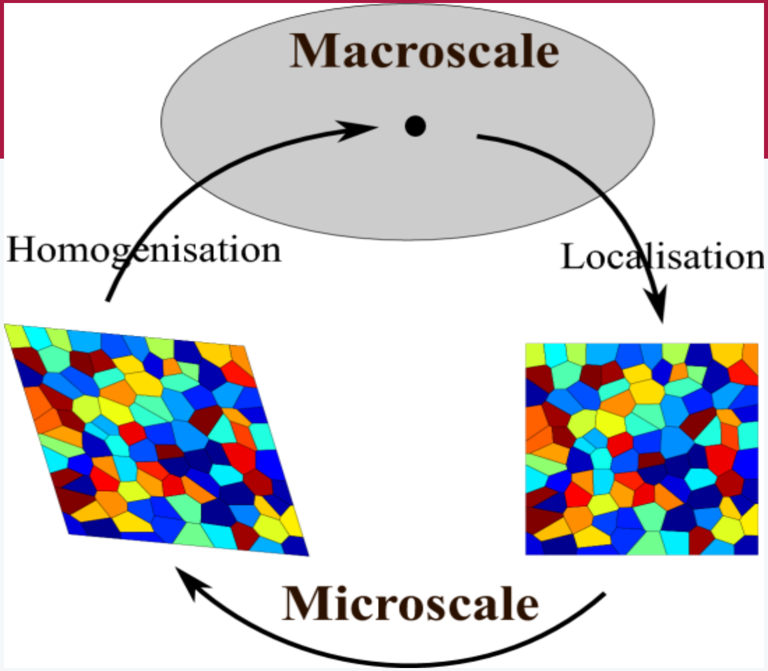
- Large number of parametric studies, e.g. load cases
- Account for the variability of the material

➡ Models and discretisations must be reduced

Ways to reduce the models

- Homogenisation (FE^2, etc.) - Hierarchical
- Concurrent and hybrid (bridging domain, ARLEQUIN, etc.)
- Enrichment (PUFEM, XFEM, GFEM)
- Model reduction (algebraic)

Reduction methods based on homogenisation



Definition of an RVE

$$l^c \gg l^f \gg l^g$$

Coupling of macroscopic and microscopic levels

The volume averaging theorem is postulated for:

1) Strain tensor:

$$\boldsymbol{\epsilon}^c = \frac{1}{|\Omega(\mathbf{x}^c)|} \int_{\partial\Omega(\mathbf{x}^c)} \mathbf{u}^f \otimes_s \mathbf{n} \, d\Gamma$$

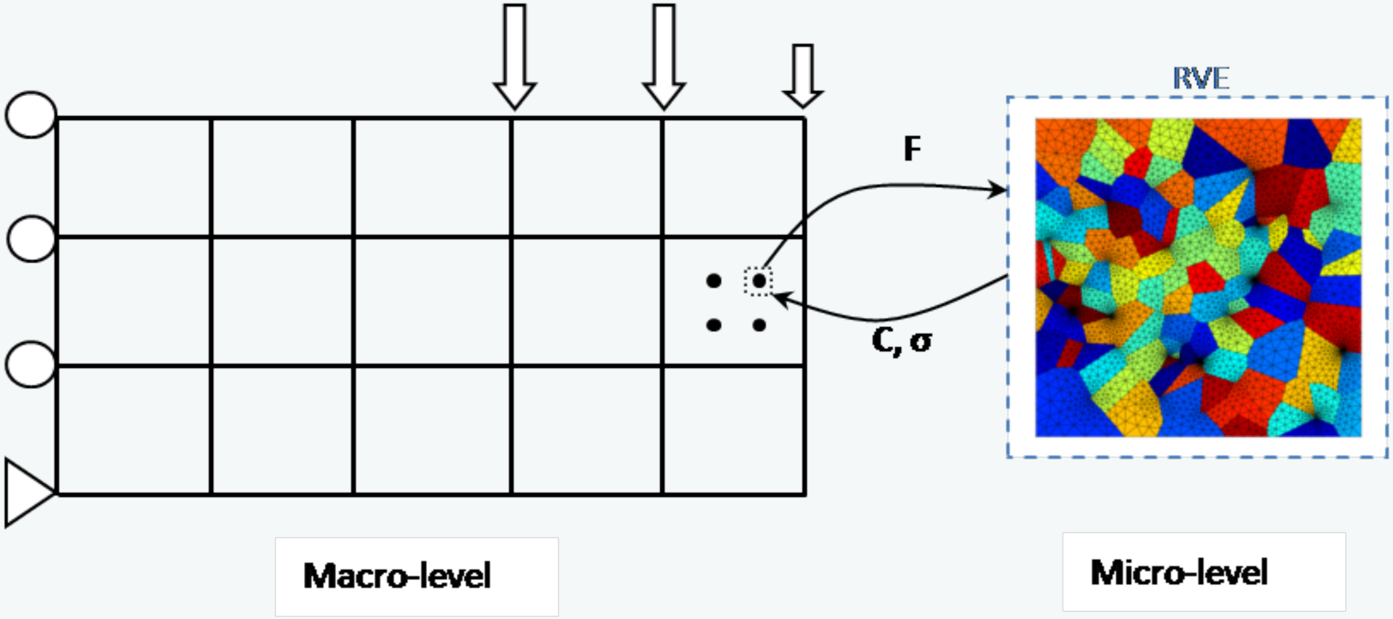
2) Virtual work (Hill-Mandel condition):

$$\boldsymbol{\sigma}^c : \boldsymbol{\delta\epsilon}^c = \frac{1}{|\Omega(\mathbf{x}^c)|} \int_{\partial\Omega(\mathbf{x}^c)} \mathbf{t}^f \cdot \delta\mathbf{u}^f \, d\Gamma$$

3) Stress tensor:

$$\boldsymbol{\sigma}^c = \frac{1}{|\Omega(\mathbf{x}^c)|} \int_{\partial\Omega(\mathbf{x}^c)} \mathbf{t}^f \otimes \mathbf{x}^f \, d\Gamma$$

Hierarchical multi-scale approaches (FE^2)



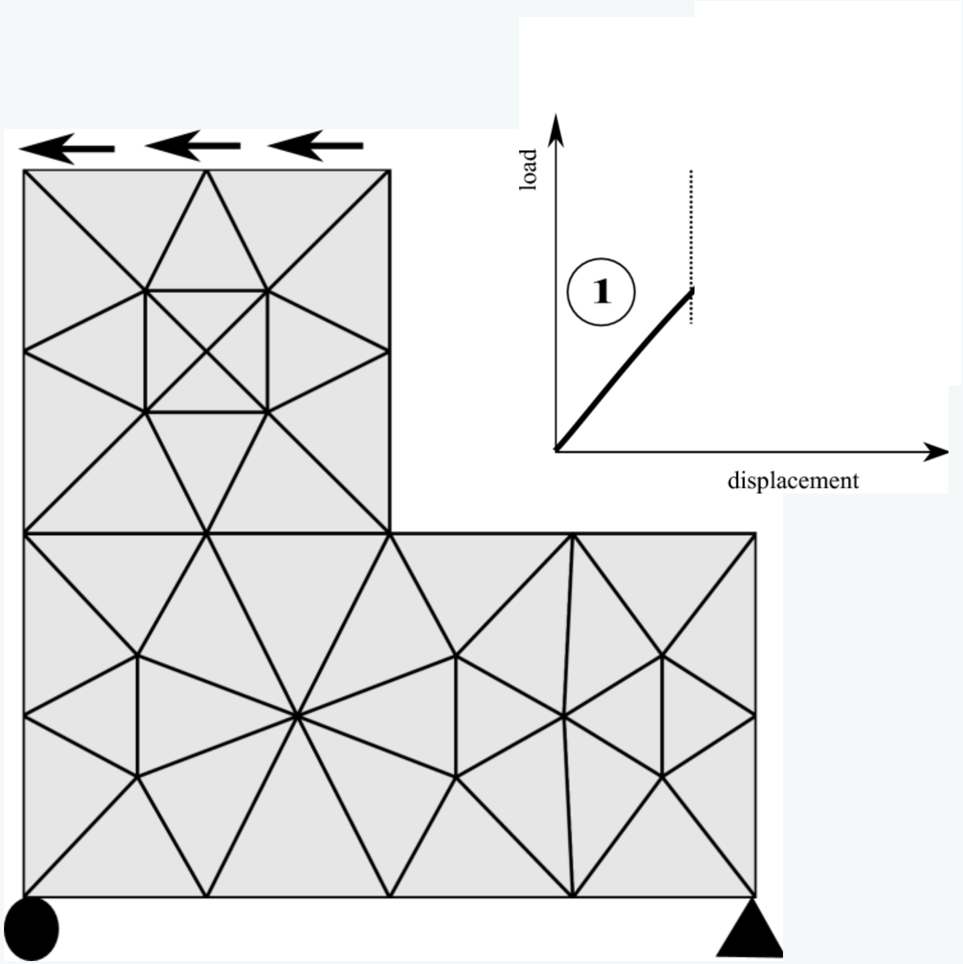
Advantages and abilities:

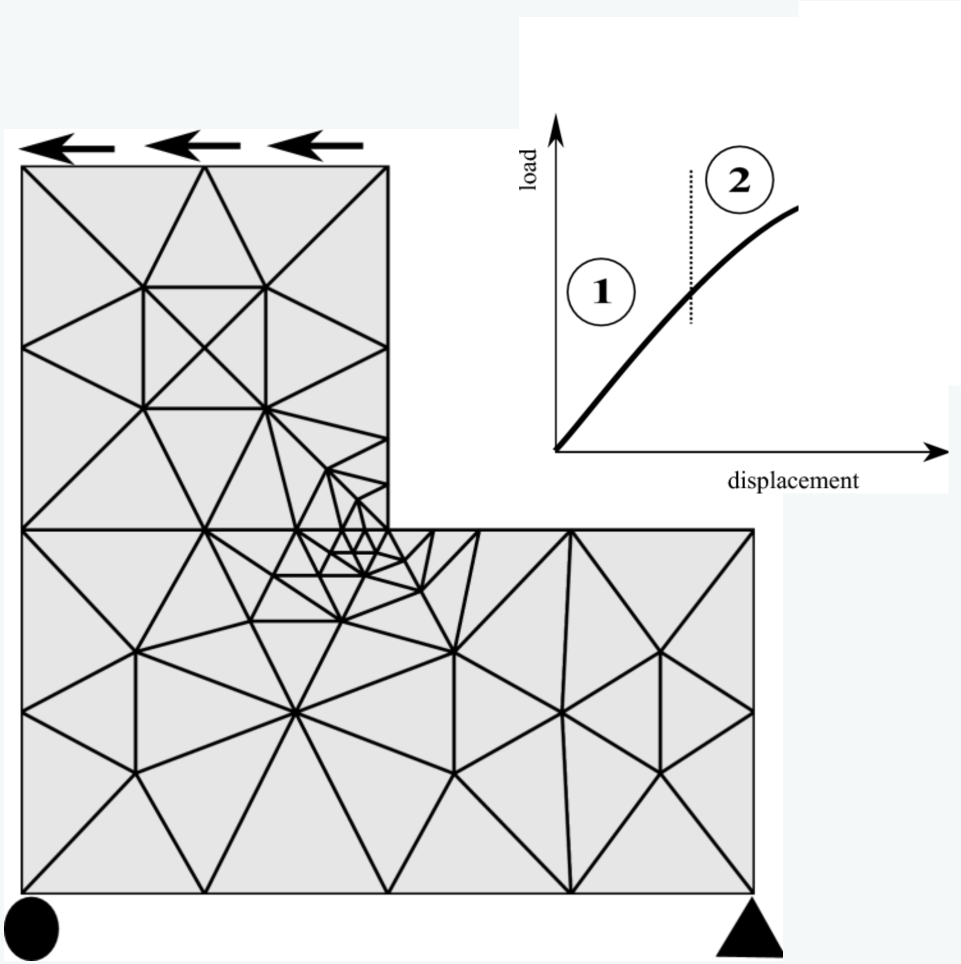
The macroscopic constitutive law is not required
Non-linear material behaviour can be simulated
Microscale behaviour of material is monitored at each load step

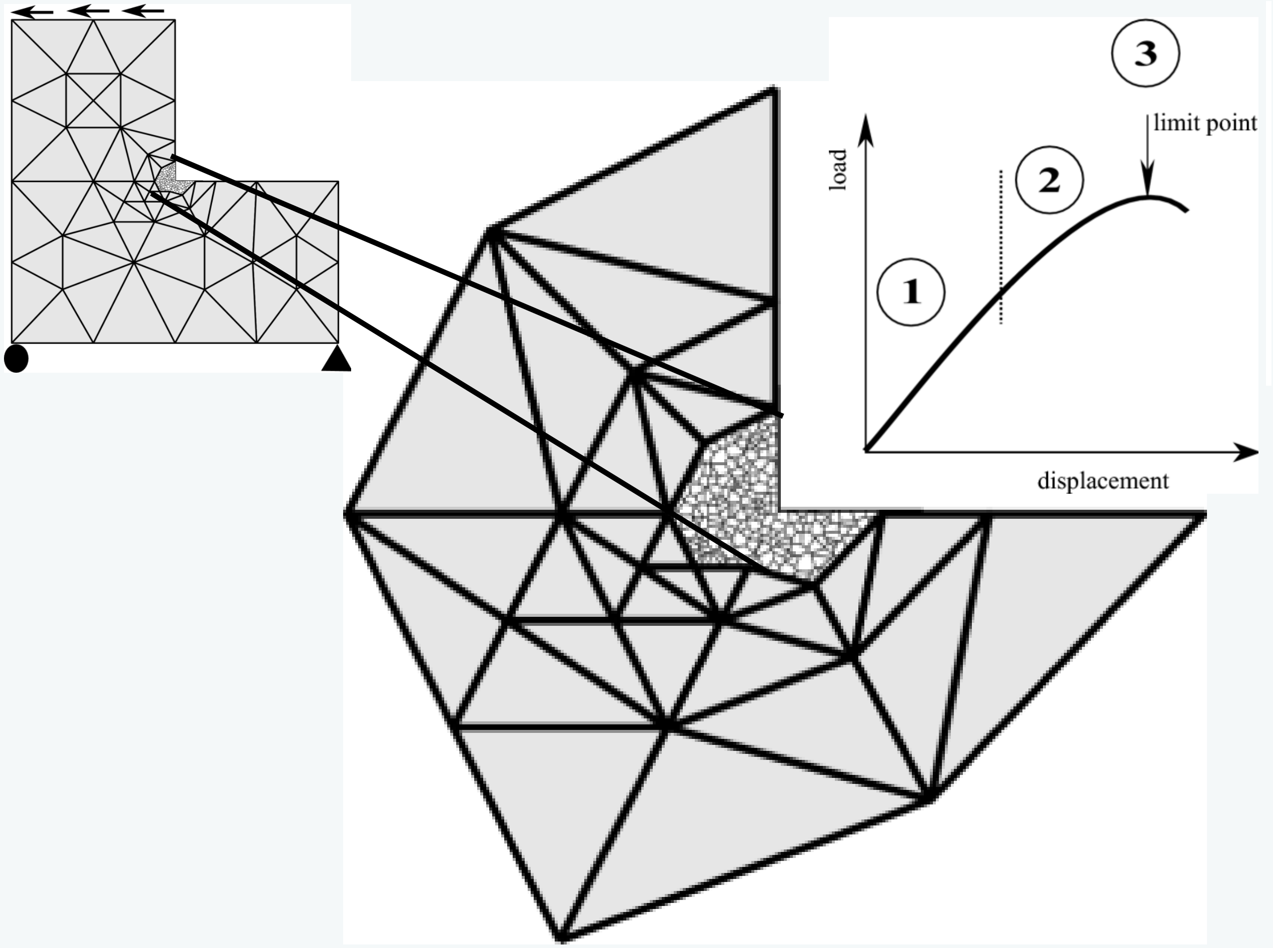
Drawbacks:

In softening regime:

- Lack of scale separation
- Macroscale mesh dependence


load
1
displacement


load
displacement
1
2


3
limit point
load
2
1
displacement

Reduction methods based on algebraic reduction

Illustration of the method of separated representation

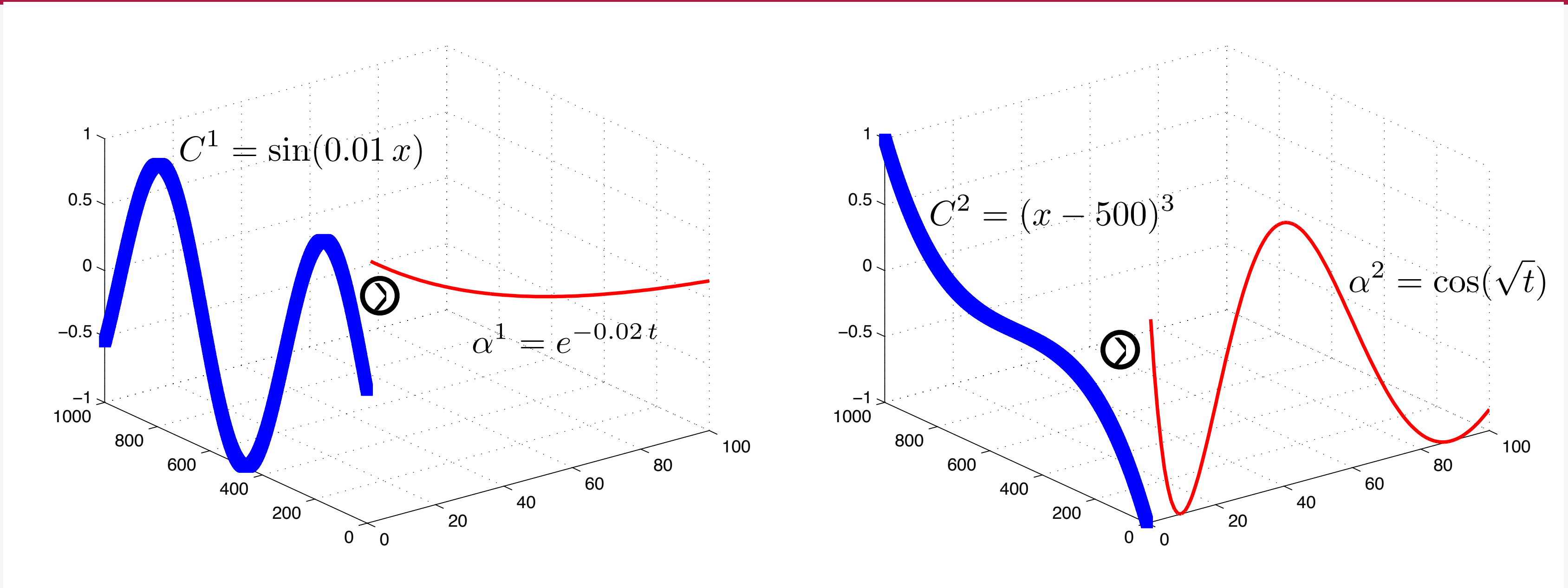


Illustration of the method of separated representation

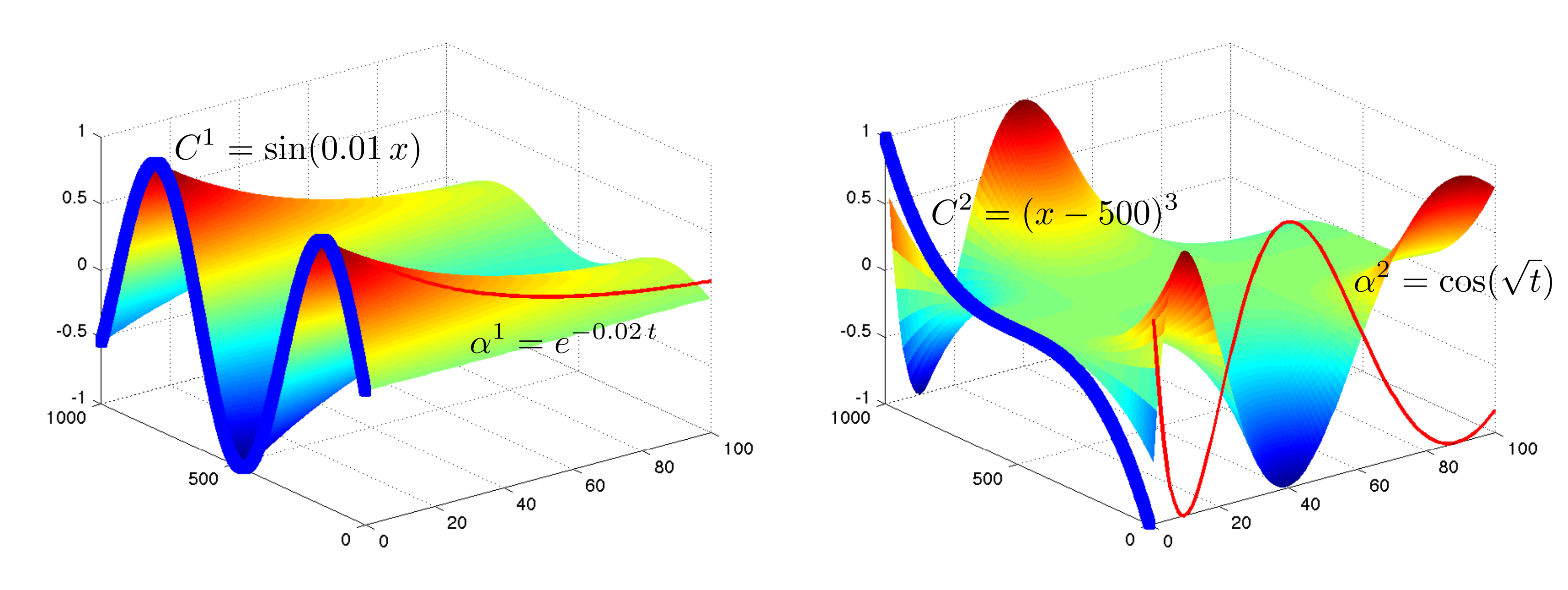
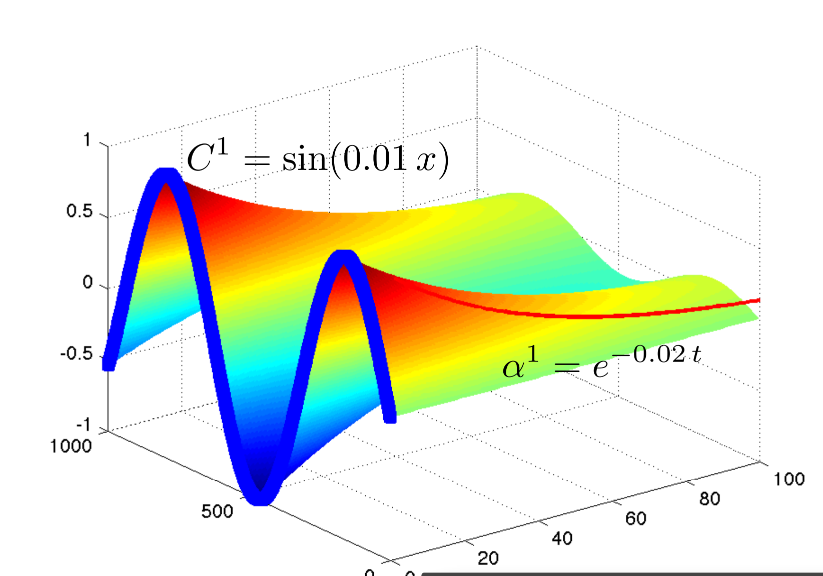
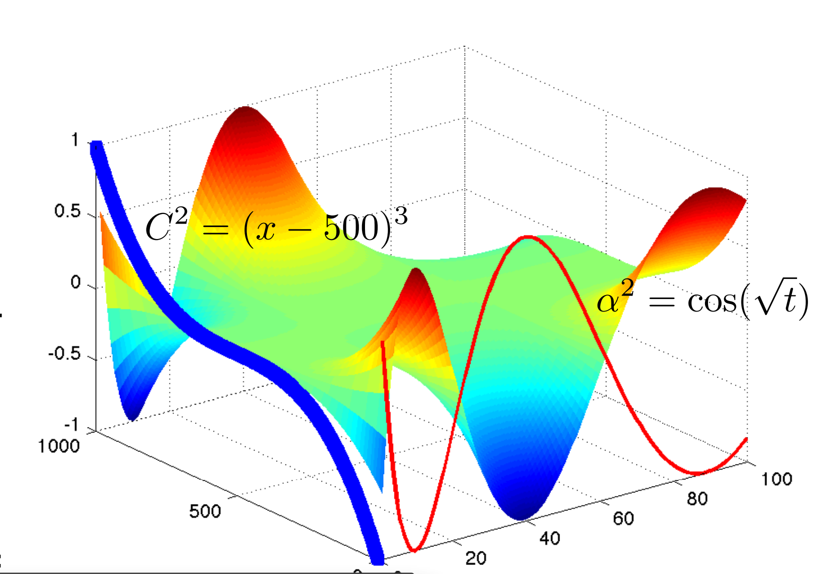


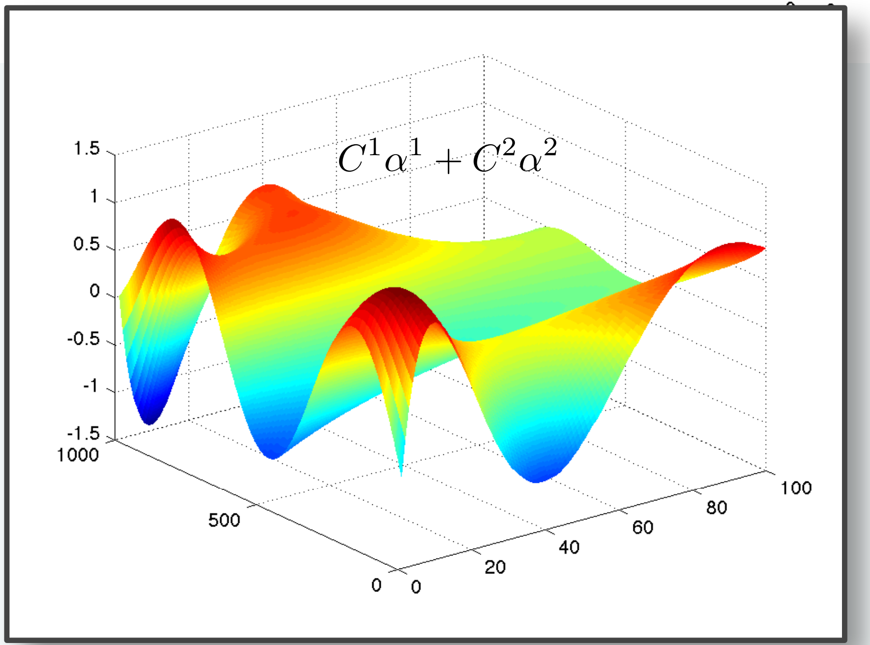
Illustration of the method of separated representation



+

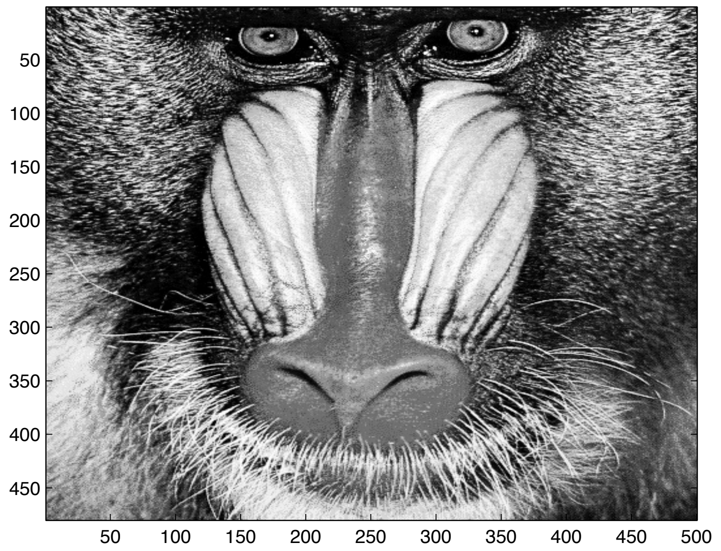


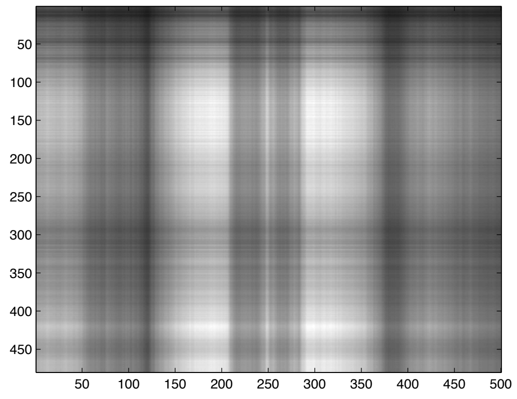
=



Very rich approximations!

Data compression: get the nose with the POD!



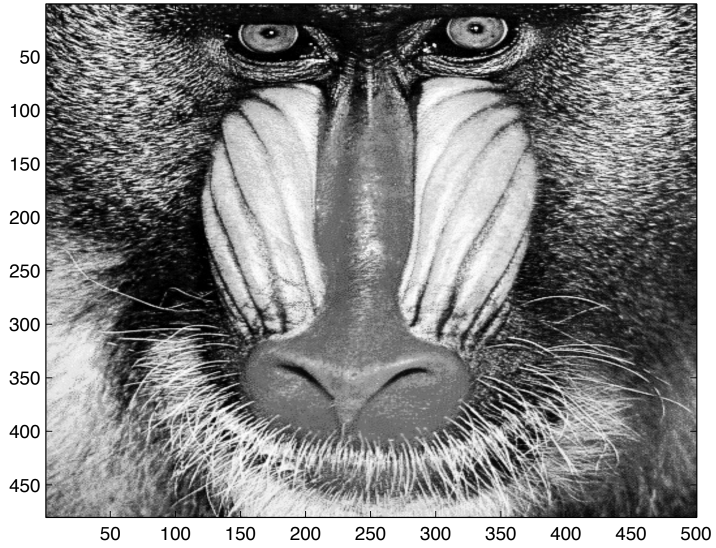


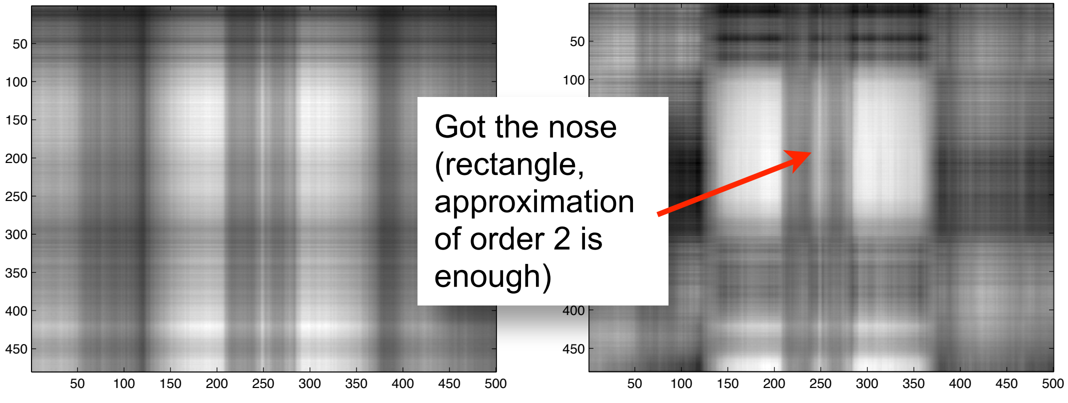
$n_c = 1$

$$\bar{u}(x_i, y_i) = \sum_{i=1}^{n_c} \underline{C_x^i}(x_i)\, \underline{C_y^i}(y_i)$$

$$(C_x^i, C_y^i)_{i\in[\![1,n_c]\!]} = \operatorname{argmin} \sum_{x_i} \sum_{y_j} (u(x_i, y_j) - \bar{u}(x_i, y_j))^2$$

Data compression: get the nose with the POD!





Got the nose (rectangle, approximation of order 2 is enough)

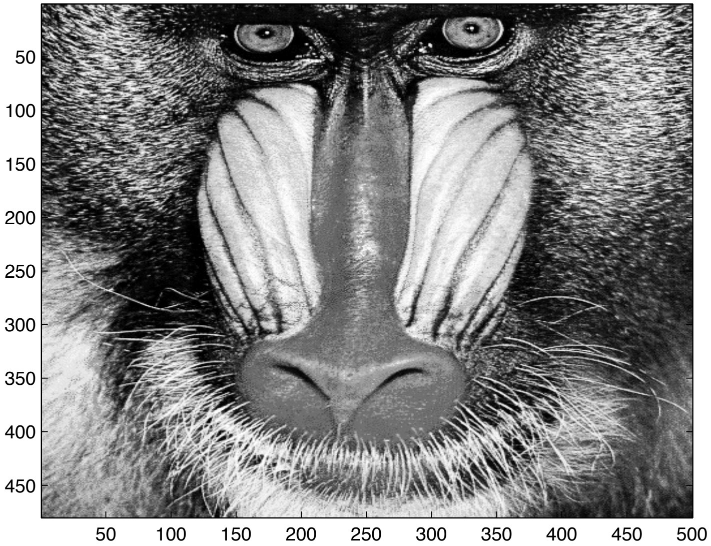
$n_c = 1$

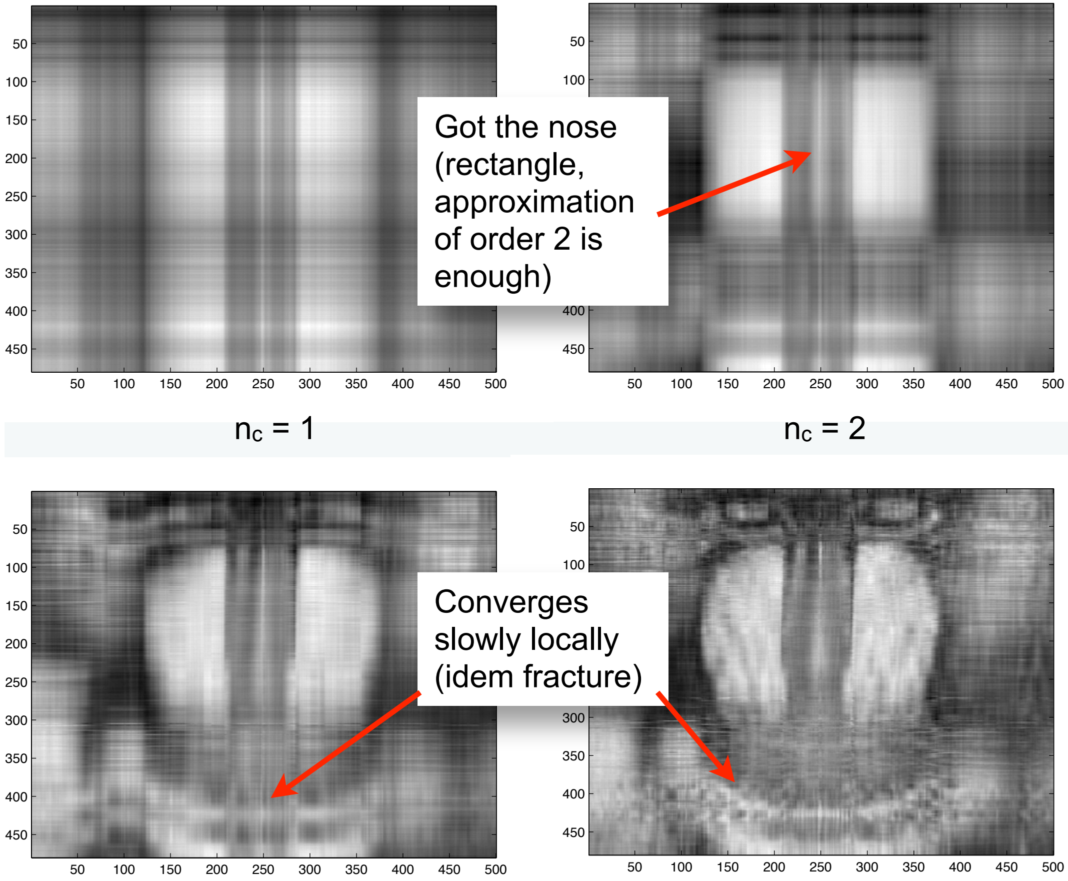
$n_c = 2$

$$\bar{u}(x_i, y_i) = \sum_{i=1}^{n_c} \underline{C_x^i}(x_i)\, \underline{C_y^i}(y_i)$$

$$(C_x^i, C_y^i)_{i \in [\![1, n_c]\!]} = \operatorname{argmin} \sum_{x_i} \sum_{y_j} (u(x_i, y_j) - \bar{u}(x_i, y_j))^2$$

Data compression: get the nose with the POD!





Got the nose (rectangle, approximation of order 2 is enough)

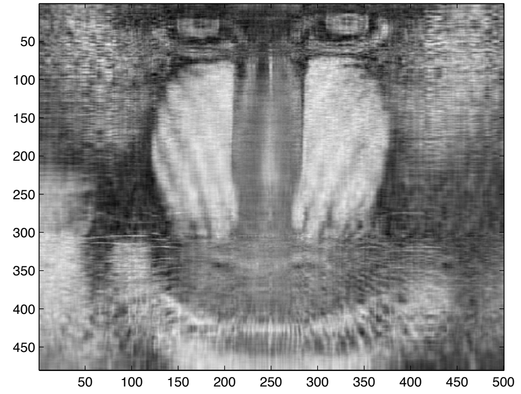
Converges slowly locally (idem fracture)

n_c = 1

n_c = 2

n_c = 5

n_c = 10



n_c = 20

$$\bar{u}(x_i, y_i) = \sum_{i=1}^{n_c} \underline{C_x^i}(x_i)\, \underline{C_y^i}(y_i)$$

$$(C_x^i, C_y^i)_{i\in[\![1,n_c]\!]} = \operatorname{argmin} \sum_{x_i} \sum_{y_j} (u(x_i, y_j) - \bar{u}(x_i, y_j))^2$$

Method of separated representation

- Search for the solution in space / time / parameter in a product space:

$$\boxed{\begin{aligned} &\underline{\bar{\mathbf{U}}} : \quad \mathcal{U}_{\text{sep}} = \mathbb{R}^n \times \mathcal{T} \times \mathcal{P} \to \mathbb{R}^n \\ &\qquad\qquad \underline{\bar{\mathbf{U}}}(t,\mu) = \sum_{i=1}^{n_C} \underline{\mathbf{C}}_i \, \beta_i(t)\gamma_i(\mu) \,, \end{aligned}}$$

$$\begin{array}{ll} \underline{\mathbf{C}}^i \in \mathbb{R}^n & \\ \beta^i : \mathcal{T} \to \mathbb{R}, & \forall i \in [\![1, n_C]\!] \,, \\ \gamma^i : \mathcal{P} \to \mathbb{R}, & \forall i \in [\![1, n_C]\!] \,, \end{array}$$

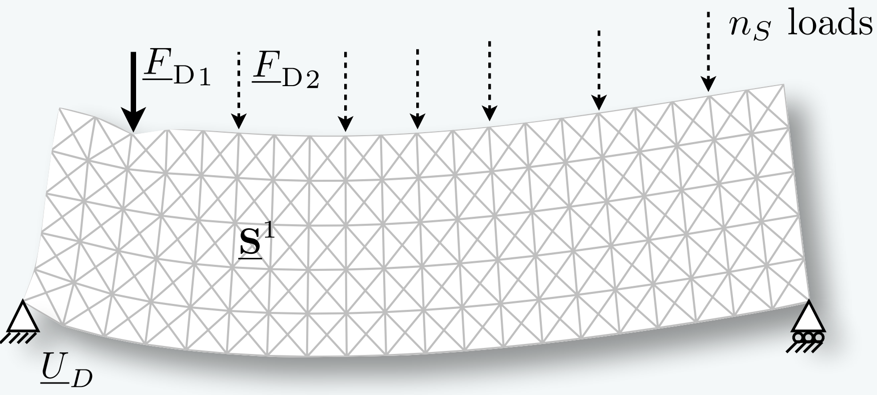
- Optimality of an expansion of order n_c with respect to a particular metric defined on $\mathcal{U}_{\text{sep}}$

  ➡ different metrics lead to different methods, which have their pro/cons

  ➡ Choice strongly dependent on the context

  - Data compression: **POD** (Proper Orthogonal Decomposition) is a classical choice in dimension 2
  - Data compression in many dimensions: **multilinear POD**
  - Solver in many dimensions without *a priori* knowledge of the solution: **PGD**
  - Model order reduction: **Snapshot POD, Snapshot PGD**
  - Initialiser, preconditioners: **low-order POD**, **low-order PGD**, **Snapshot POD**

a posteriori model order reduction. **Idea**: search for the **solution as a linear combination** of a set of **pre-calculated representative solutions**



(1) Solve FINE for n_S parameters (EXPENSIVE!)

$$\underline{\underline{S}} = \left(\; \underline{S}^1 \quad \underline{S}^2 \quad \ldots \quad \underline{S}^{n_S} \; \right)$$

(2) Singular value decomposition

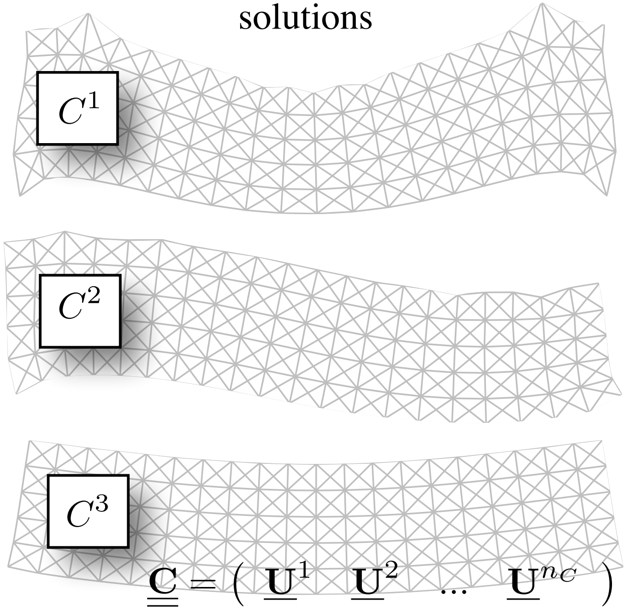
$$\underline{\underline{S}} = \underline{\underline{U}}\,\underline{\underline{\Sigma}}\,\underline{\underline{V}}^T = \sum_{k=1}^{n_S} \Sigma^k \, \underline{U}^k \, \underline{V}^{k\,T}$$

n_S solutions, sorted by relevance

where $(\Sigma^k)_{k \in [\![1 \;\, n_S]\!]}$ in decreasing order

(3) Truncation

Reduced basis: family of representative solutions



$$\underline{\underline{C}} = \left(\; \underline{U}^1 \quad \underline{U}^2 \quad \ldots \quad \underline{U}^{n_C} \; \right)$$

Family of representative solutions

$$\underline{U} = \underline{\underline{C}}\,\underline{\alpha}$$

Solution

Coefficients

Initial set of equations

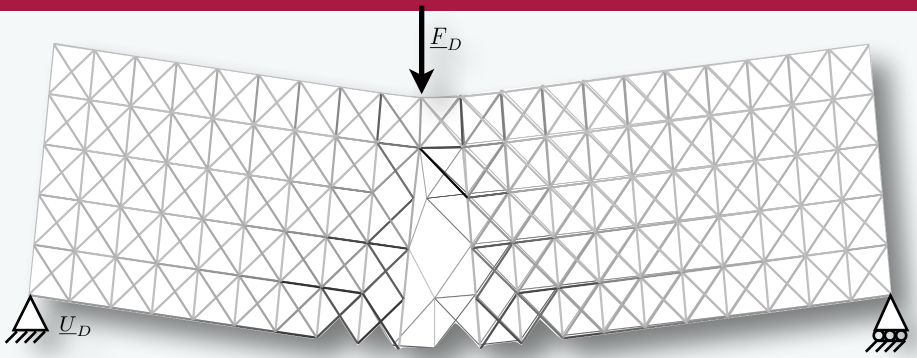
$$\underline{F}_{\mathrm{Int}}\left(\underline{U}\right) + \underline{F}_{\mathrm{Ext}} = 0$$

(4) Galerkin orthogonality

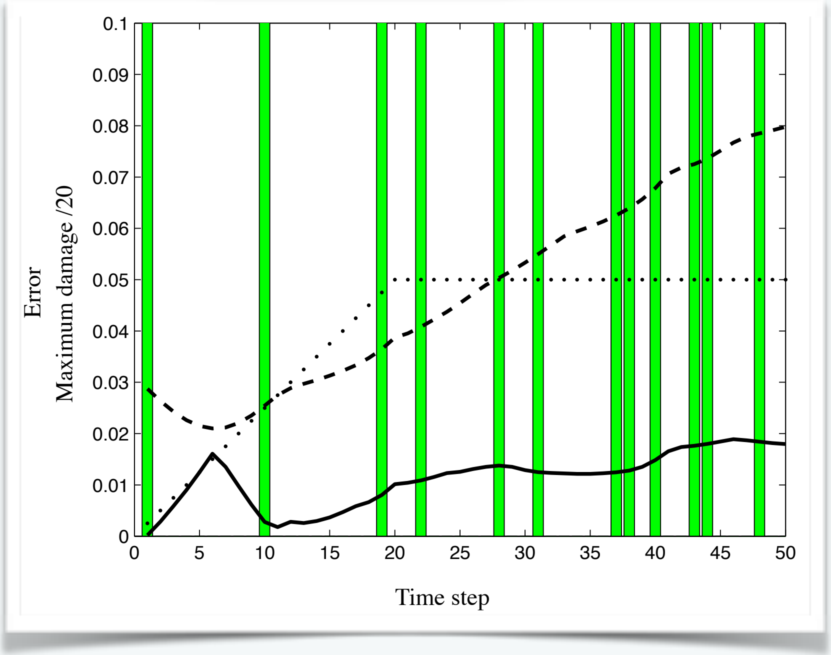
$$\underline{\underline{\mathbf{C}}}^T\,\underline{\mathbf{F}}_{\mathbf{int}}\left(\underline{\underline{\mathbf{C}}}\,\underline{\alpha}\right) + \underline{\underline{\mathbf{C}}}^T\,\underline{\mathbf{F}}_{\mathbf{ext}} = 0$$

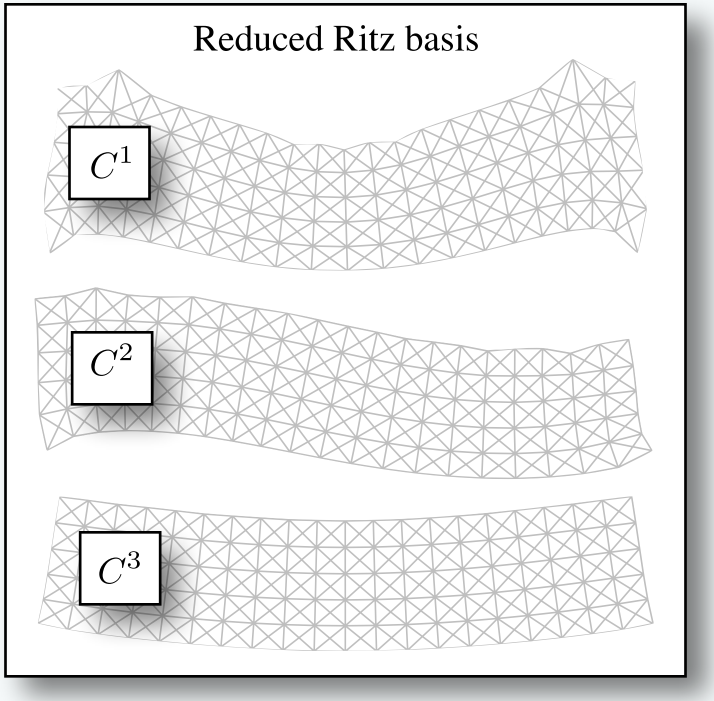
Approximation of the solution in a space of small dimension (n$_c$)

Limitations: case of highly non-linear fracture mechanics



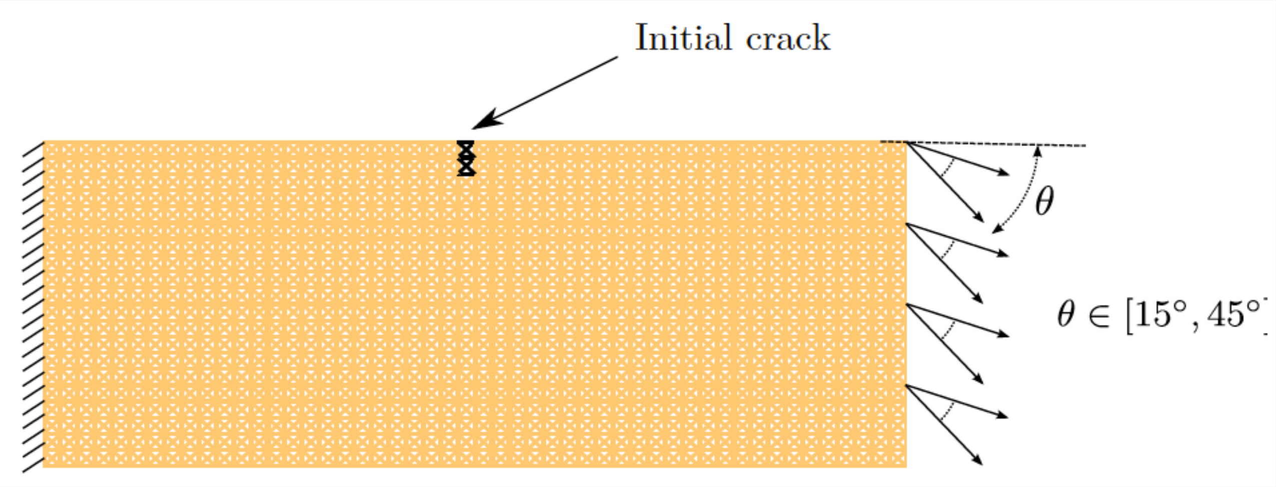
This solution is not in the snapshot !



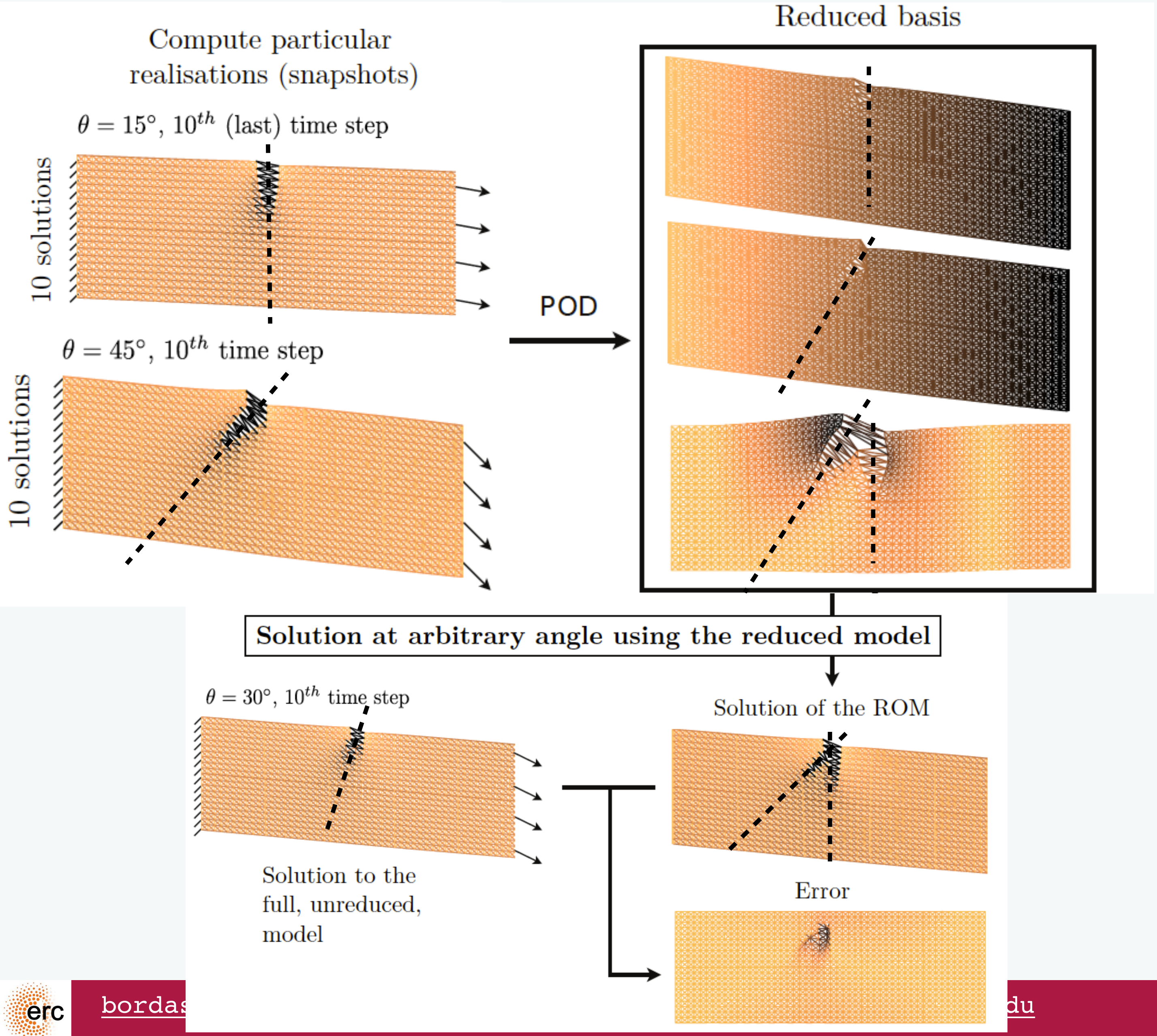


- P. Kerfriden, P. Gosselet, S. Adhikari, and S. Bordas. *Bridging proper orthogonal decomposition methods and augmented Newton-Krylov algorithms: an adaptive model order reduction for highly nonlinear mechanical problems.* Computer Methods in Applied Mechanics and Engineering, 200(5-8): 850-866, 2011.

Application to a parametric fracture problem

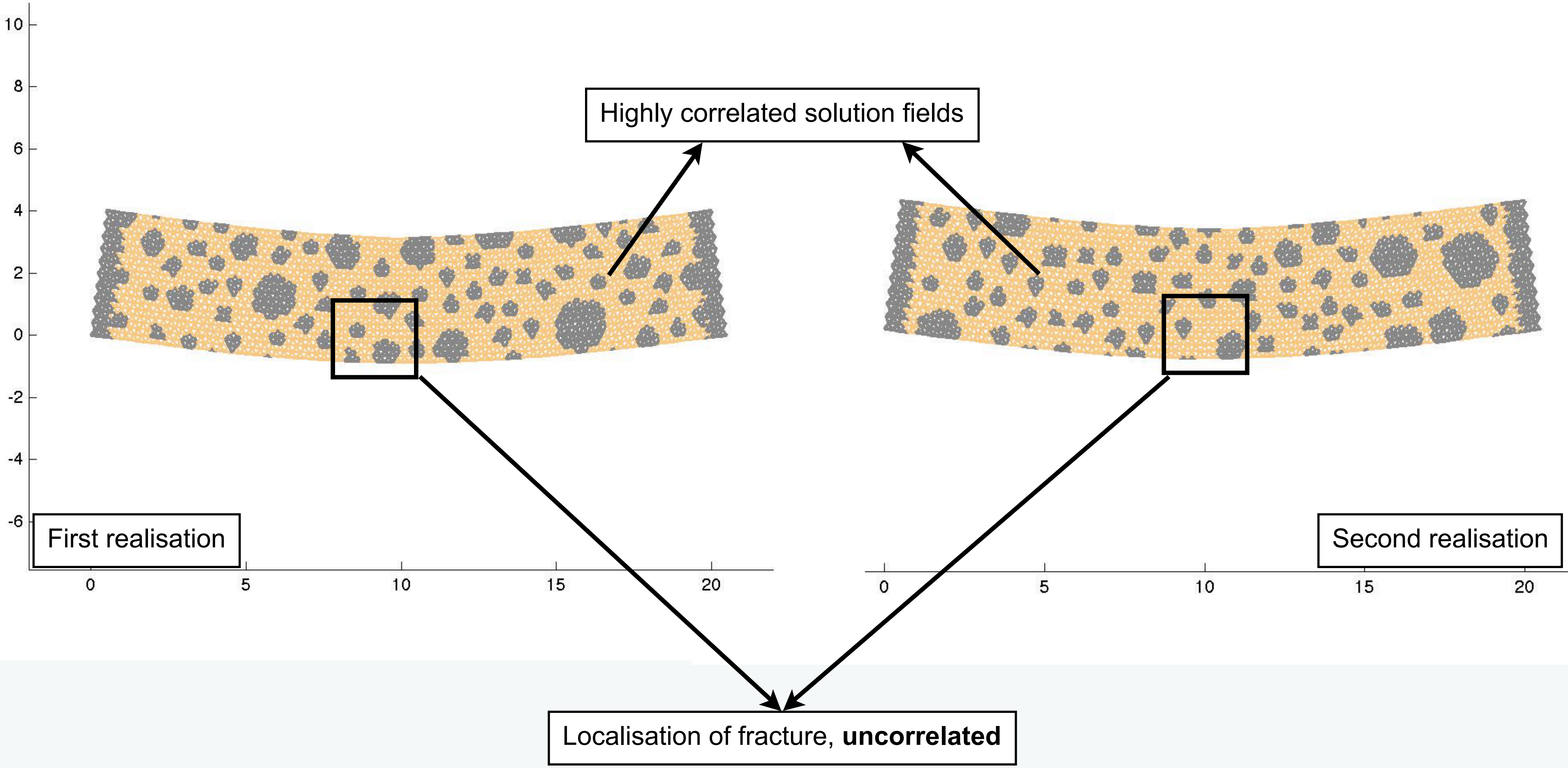


Application to a parametric fracture problem


Compute particular realisations (snapshots)
$\theta = 15°$, 10^{th} (last) time step
10 solutions
$\theta = 45°$, 10^{th} time step
10 solutions
POD
Reduced basis
Solution at arbitrary angle using the reduced model
$\theta = 30°$, 10^{th} time step
Solution to the full, unreduced, model
Solution of the ROM
Error

- The POD solution is not able to reproduce the solution in the cracked area
- Due to lack of correlation introduced by crack growth
- Leads to a local projection error

Parametric / stochastic multiscale fracture mechanics



Highly correlated solution fields

First realisation

Second realisation

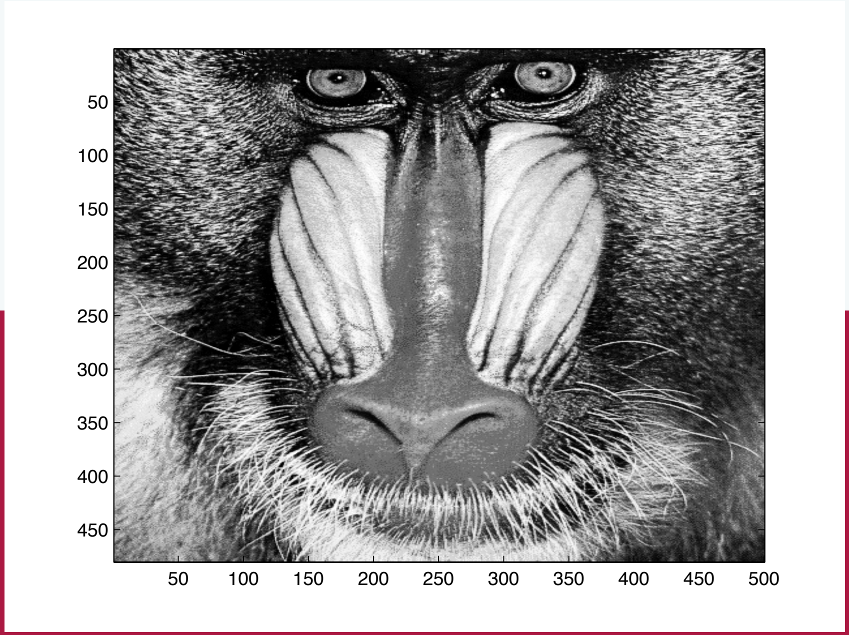
Localisation of fracture, **uncorrelated**

➡ Direct numerical simulation: efficient preconditioner?

➡ Reduced order modelling?

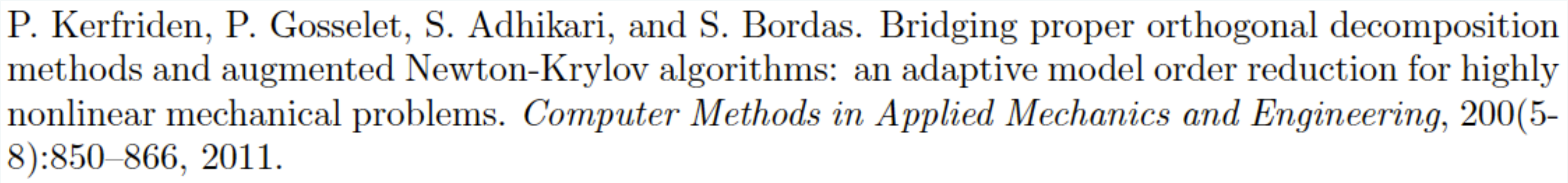
➡ Adaptive coupling?

THE RETURN OF THE MONKEY!

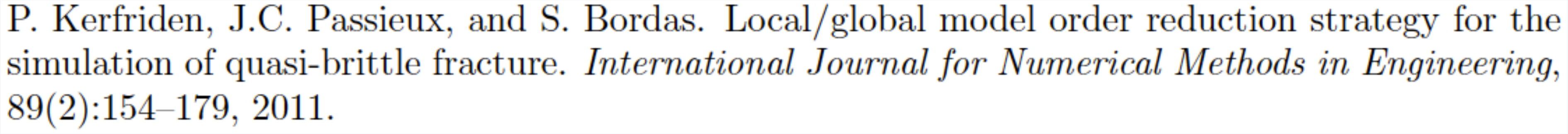


What can we do to address this lack of separation of scales/reducibility?

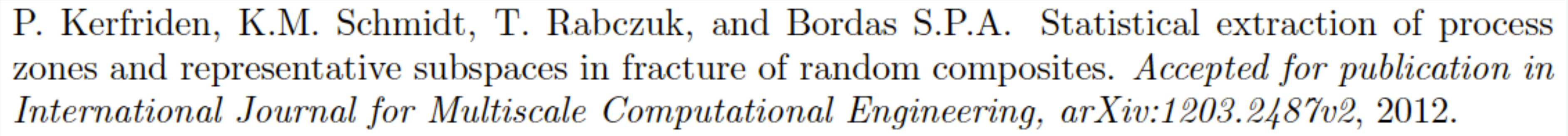
How we got to this point...



P. Kerfriden, P. Gosselet, S. Adhikari, and S. Bordas. Bridging proper orthogonal decomposition methods and augmented Newton-Krylov algorithms: an adaptive model order reduction for highly nonlinear mechanical problems. *Computer Methods in Applied Mechanics and Engineering*, 200(5-8):850–866, 2011.



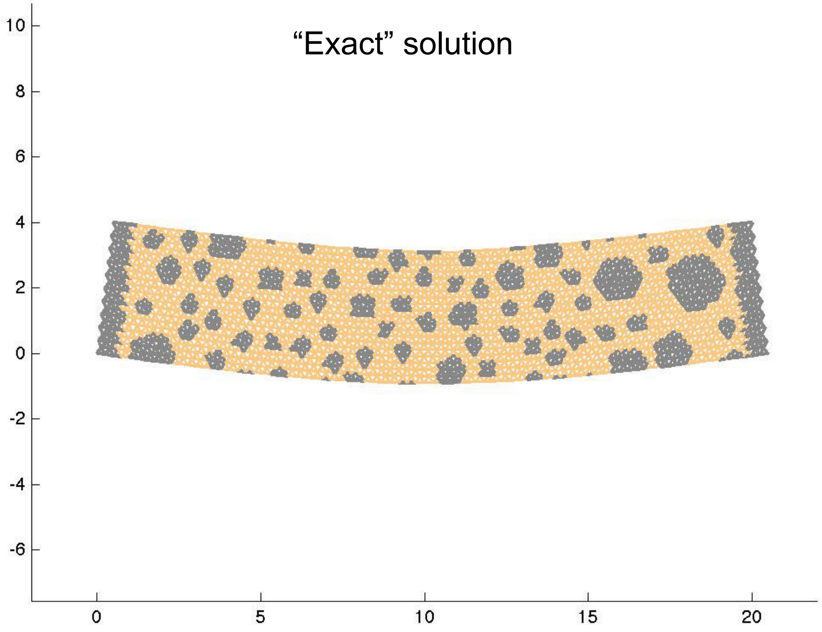
P. Kerfriden, J.C. Passieux, and S. Bordas. Local/global model order reduction strategy for the simulation of quasi-brittle fracture. *International Journal for Numerical Methods in Engineering*, 89(2):154–179, 2011.

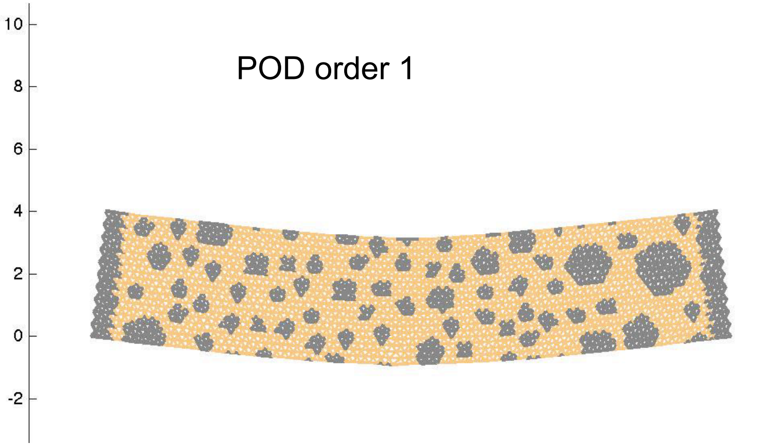


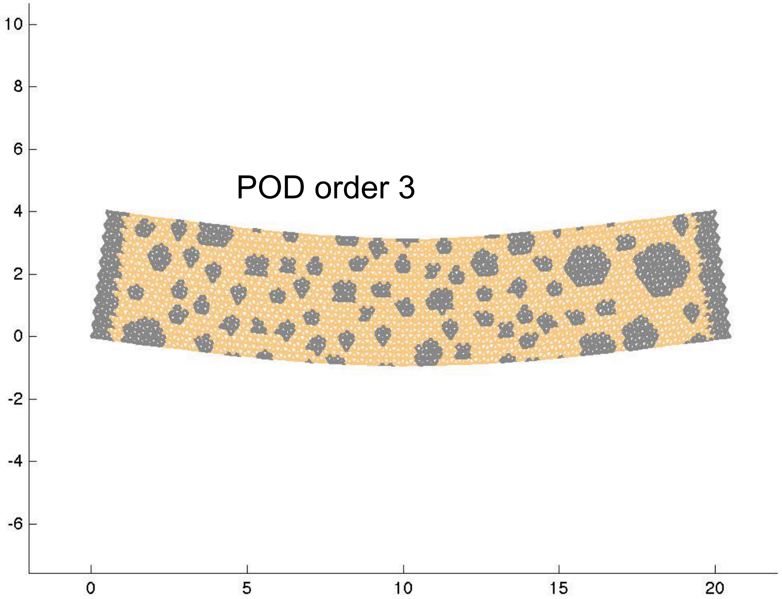
P. Kerfriden, K.M. Schmidt, T. Rabczuk, and Bordas S.P.A. Statistical extraction of process zones and representative subspaces in fracture of random composites. *Accepted for publication in International Journal for Multiscale Computational Engineering, arXiv:1203.2487v2*, 2012.

Data compression: fracture

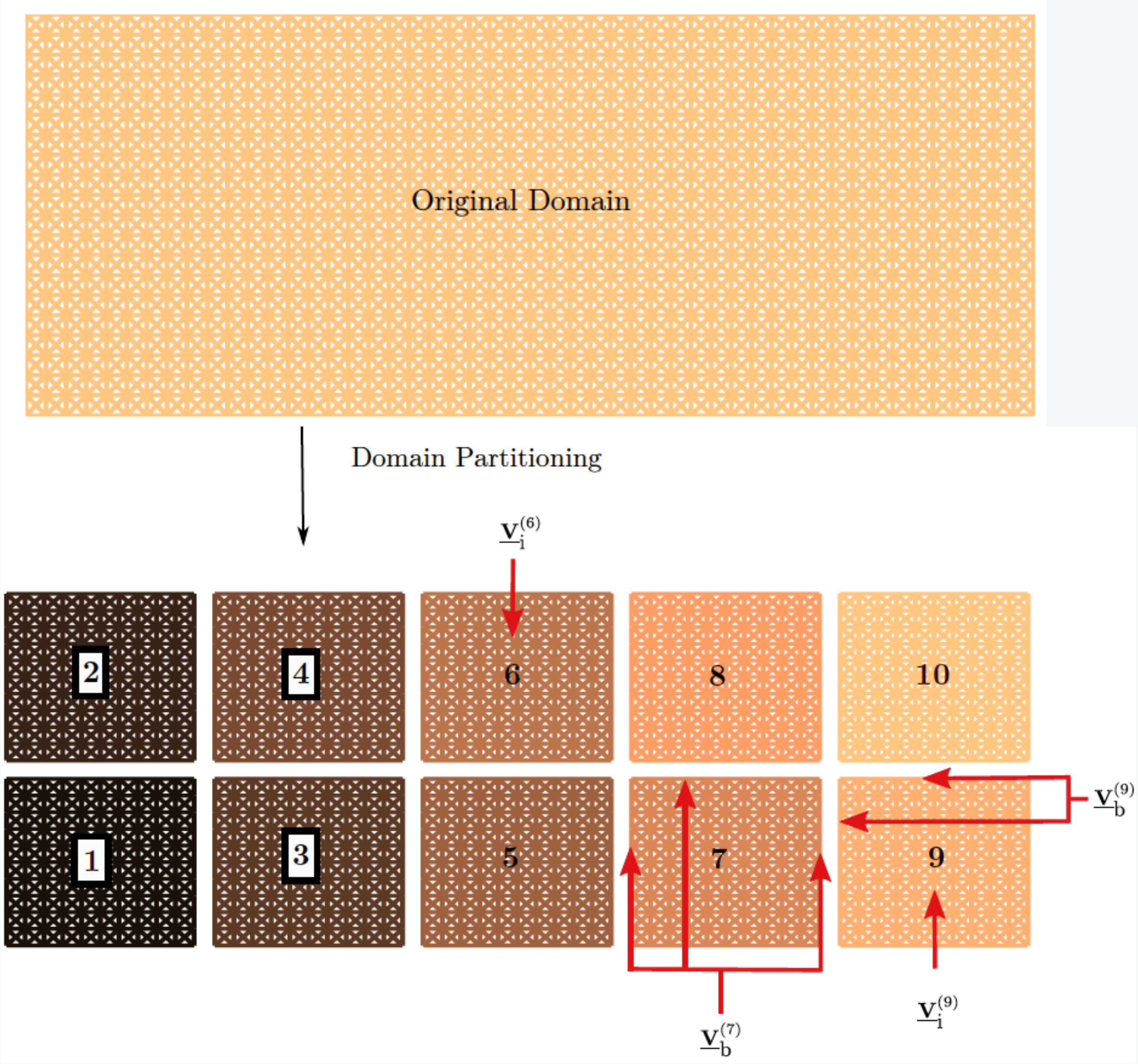
Snapshot POD (snapshot space is spanned by the ensemble of solutions at all time steps)


“Exact” solution

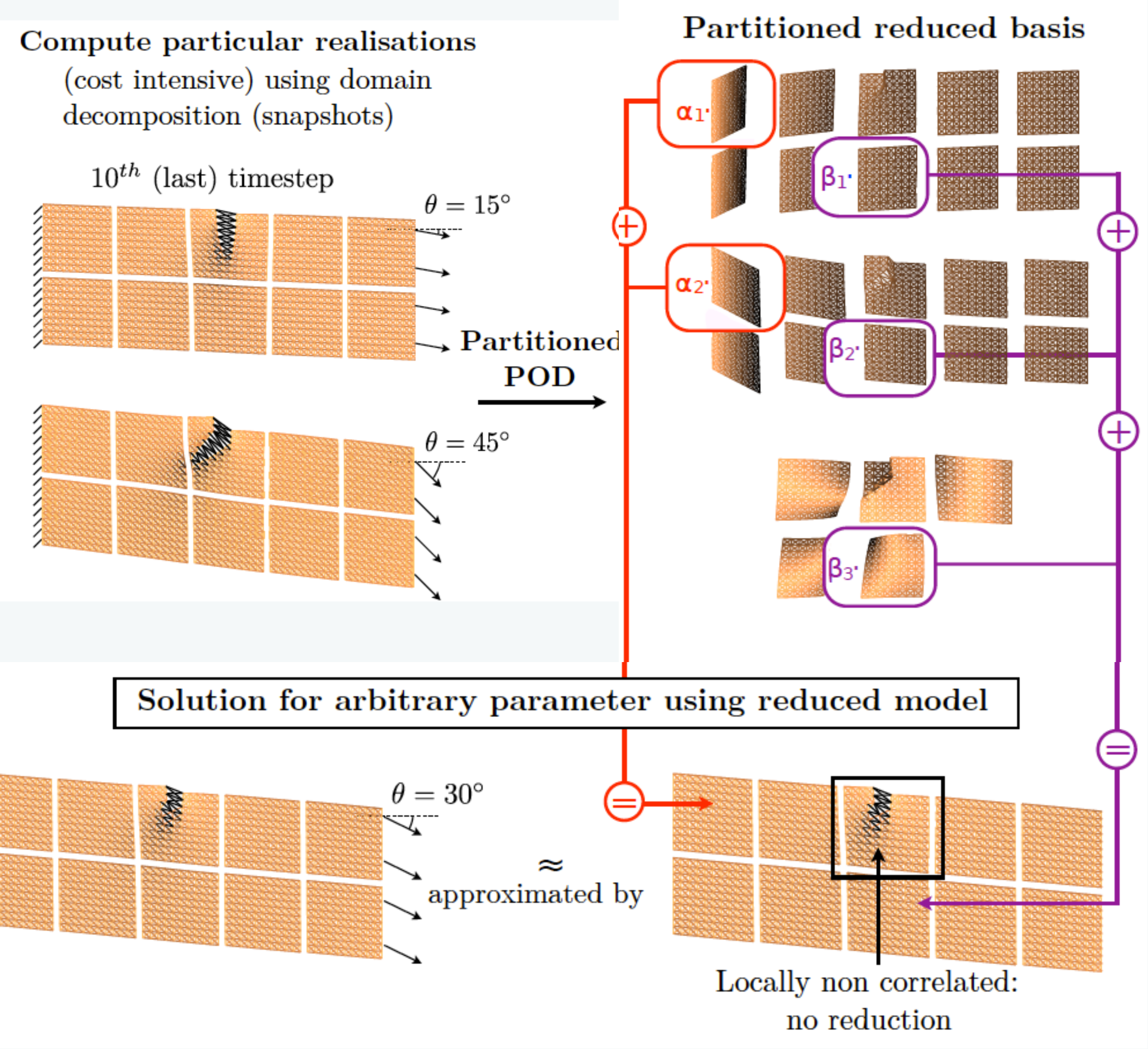

POD order 1


POD order 3

Partitioned POD/DDM


Original Domain
Domain Partitioning
$\underline{\mathbf{V}}_{\mathrm{i}}^{(6)}$
2
4
6
8
10
1
3
5
7
9
$\underline{\mathbf{V}}_{\mathrm{b}}^{(9)}$
$\underline{\mathbf{V}}_{\mathrm{i}}^{(9)}$
$\underline{\mathbf{V}}_{\mathrm{b}}^{(7)}$

Reduced DDM-POD



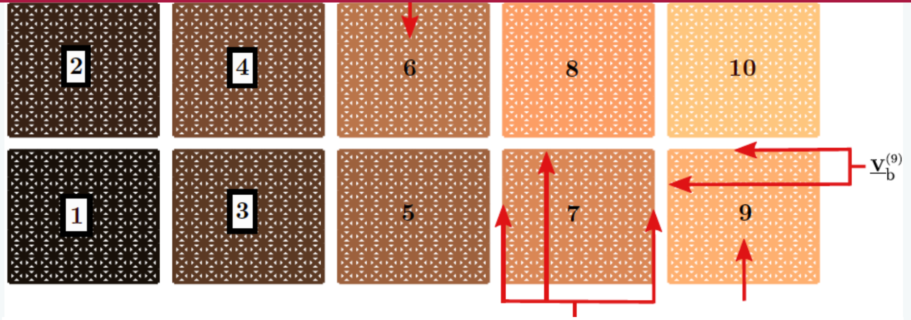
Compute particular realisations (cost intensive) using domain decomposition (snapshots)

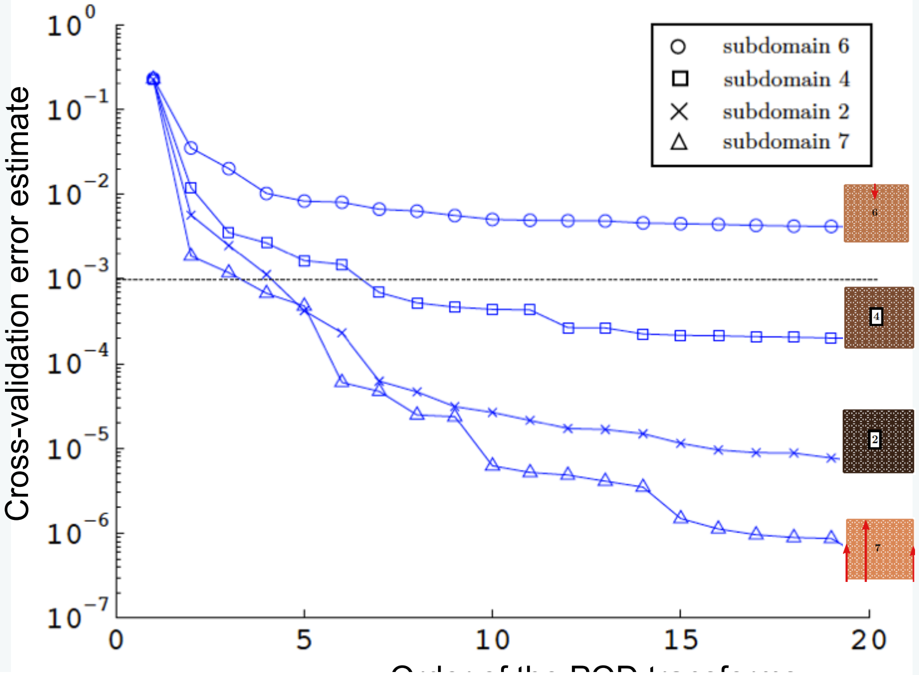
Partitioned reduced basis

Solution for arbitrary parameter using reduced model

- Decompose the structure into subdomains
- Perform a reduction in the highly correlated region
- Couple the reduced to the non-reduced region by a primal Schur complement

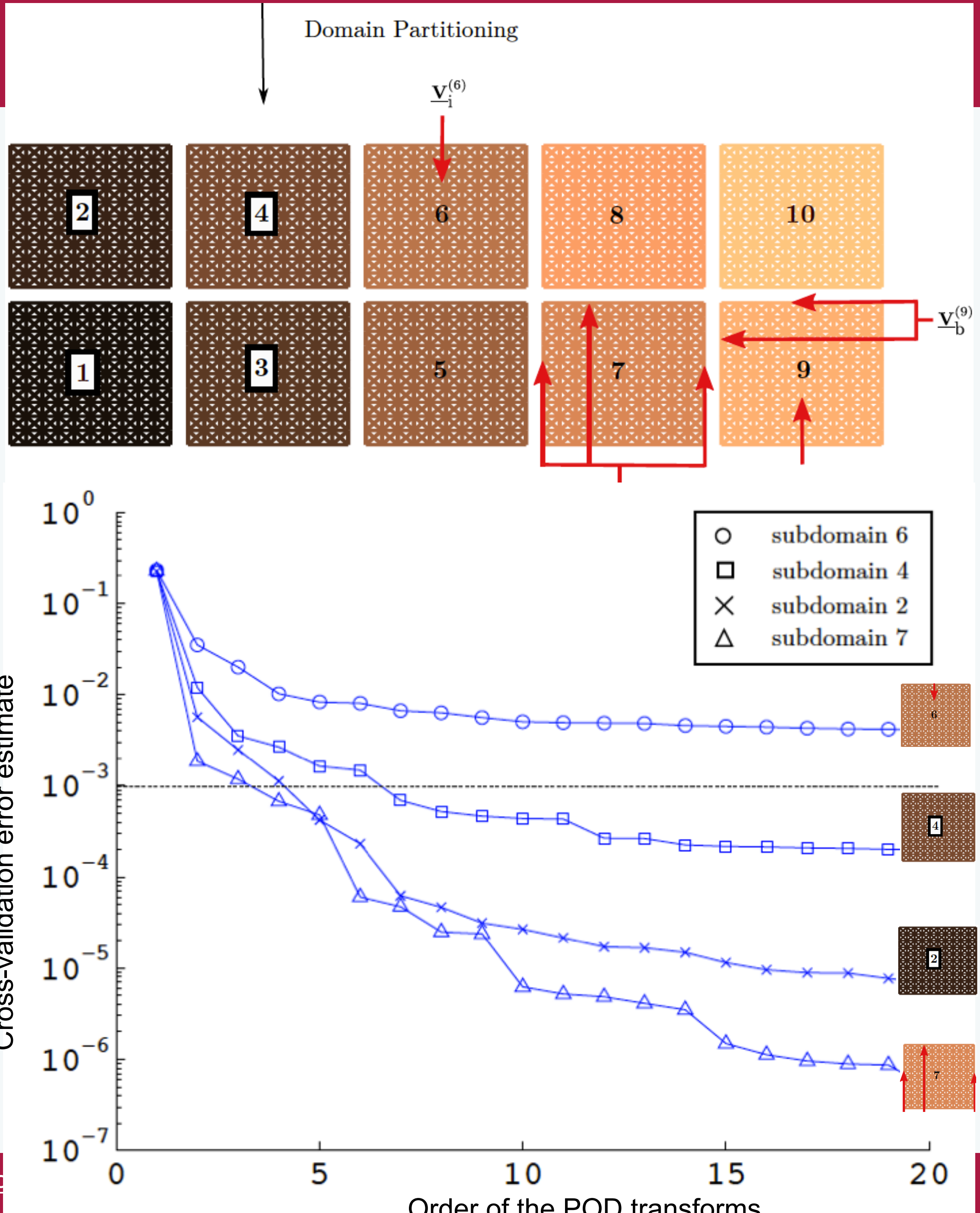
Choice of the reduced subdomains: local error estimation by “leave one out cross-validation” (LOOCV)





$$\left(\tilde{\nu}^{(e)}_{\mathrm{snap}}\right)^2 = \frac{\displaystyle\sum_{\mu\in\mathcal{P}^{\mathrm{s}}}\sum_{t_n\in\mathcal{T}^{\mathrm{h}}}\left\|\underline{\mathbf{U}}_{\mathrm{i}}(t_n,\mu) - \sum_{j=1}^{n_c^{(e)}}\left(\tilde{\underline{\mathbf{C}}}_{\mathrm{i},j}^{(e),(\mu)\,T}\,\underline{\mathbf{U}}_{\mathrm{i}}(t_n,\mu)\right)\tilde{\underline{\mathbf{C}}}_{\mathrm{i},j}^{(e),(\mu)}\right\|_2^2}{\displaystyle\sum_{t_n\in\mathcal{T}^{\mathrm{h}}}\sum_{\mu\in\mathcal{P}^{\mathrm{s}}}\|\underline{\mathbf{U}}_{\mathrm{i}}(t_n,\mu)\|_2^2}$$

- Reduced subspaces are independent and we assume a snapshot is *a priori* available
  - (1) Dimension of the local space for each subdomain?
  - (2) Is a given subdomain is reducible?
- (1) and (2) will be treated by cross-validation (e.g. W. J. Krzanowski. Cross-validation in principal component analysis. Biometrics, 43(3):575-584, 1987.)
  - **Training set**: snapshot
  - **Validation set**: set of additional finescale solutions
  - Independent training/validation avoids overfitting
  - Cross validation **emulates independence**. Error calculated using the local reduced basis obtained by a snapshot POD transform of all the available snapshot solutions except the one corresponding to the value of the summation variable.
- **NOTE:** If the snapshot is not assumed *a priori* then
  - Assess whether the snapshot contains sufficient information, and generate additional, suitable, data if required
  - Most analysis (mostly by statisticians) assume the snapshot is known *a priori*. Recent review: Hervé Abdi and Lynne J. Williams. Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics, 2(4):433{459, 2010.

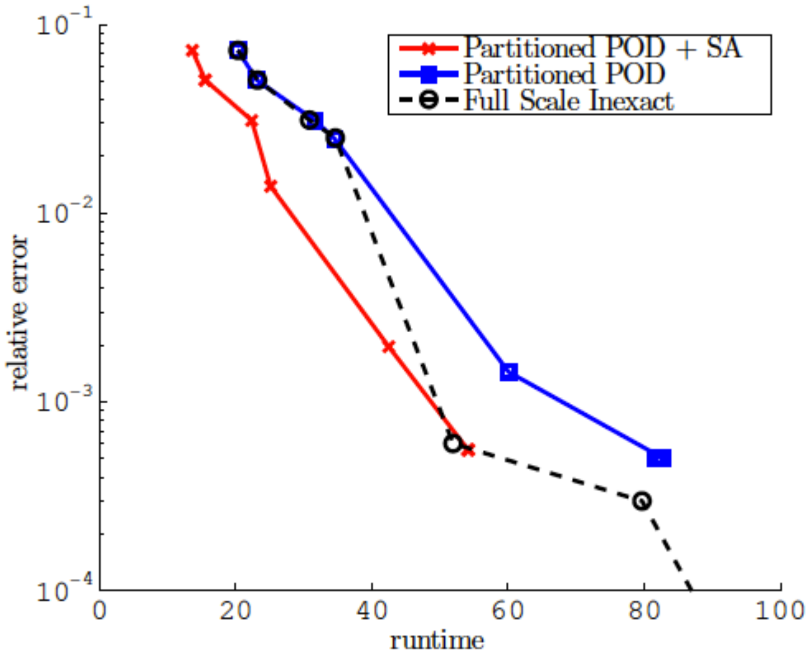

Domain Partitioning
$\underline{v}_i^{(6)}$
$\underline{v}_b^{(9)}$
2
4
6
8
10
1
3
5
7
9
Cross-validation error estimate
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
0
5
10
15
20
Order of the POD transforms
○ subdomain 6
□ subdomain 4
× subdomain 2
△ subdomain 7
6
4
2
7

Performance: load angle 40 | 27 - 121 nodes

- Relative error

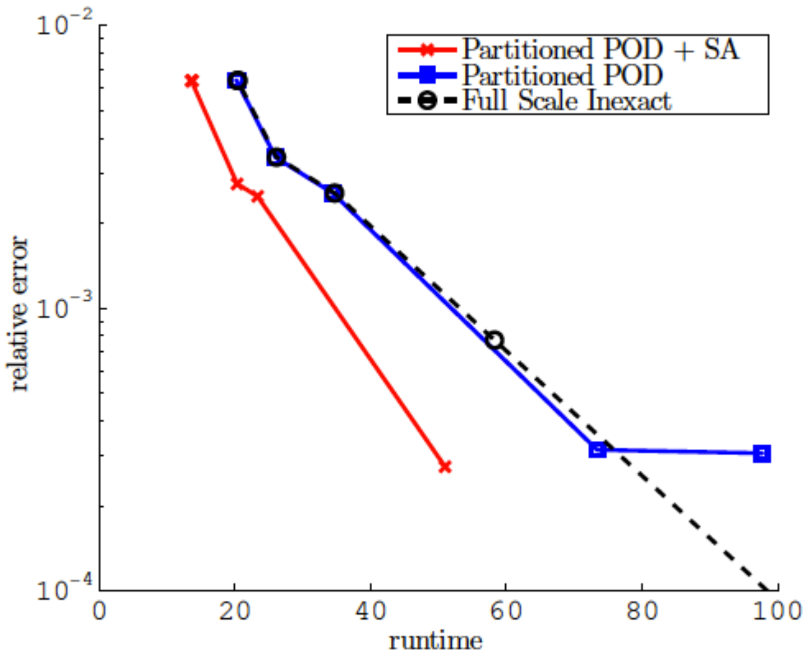
$$\nu^{\mathrm{app},(\mu)}\left(\underline{\mathbf{U}}^{\mathrm{app}}\right)^2 = \frac{\displaystyle\sum_{t_n\in\mathcal{T}^h}\|\underline{\mathbf{U}}^{\mathrm{app}}(t_n,\mu)-\underline{\mathbf{U}}^{\mathrm{ex}}(t_n,\mu)\|_2^2}{\displaystyle\sum_{t_n\in\mathcal{T}^h}\|\underline{\mathbf{U}}^{\mathrm{ex}}(t_n,\mu)\|_2^2}$$

40^o



(a) Relative error for the different models using 121 nodes per subdomain

27^o



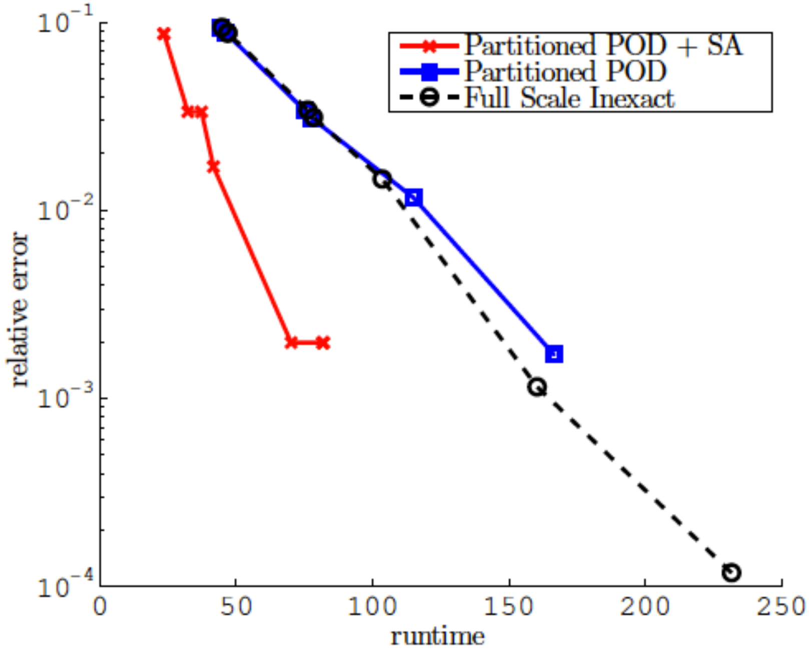
(a) Relative error for the different models using 121 no per subdomain

Performance: load angle 40 | 27 - 256 nodes

- Relative error

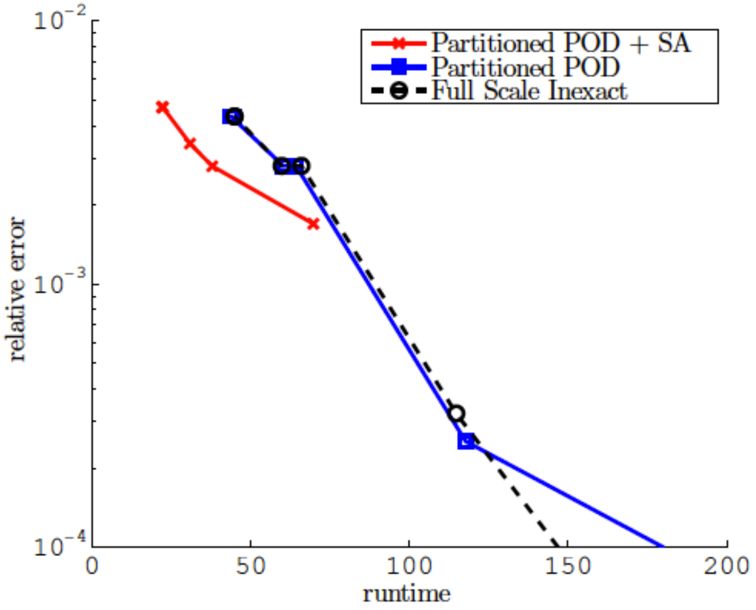
$$\nu^{\mathrm{app},(\mu)}\left(\underline{\mathbf{U}}^{\mathrm{app}}\right)^2 = \frac{\sum\limits_{t_n\in\mathcal{T}^h} \|\underline{\mathbf{U}}^{\mathrm{app}}(t_n,\mu)-\underline{\mathbf{U}}^{\mathrm{ex}}(t_n,\mu)\|_2^2}{\sum\limits_{t_n\in\mathcal{T}^h}\|\underline{\mathbf{U}}^{\mathrm{ex}}(t_n,\mu)\|_2^2}$$

40^o



(b) Relative error for the different models using 256 nodes per subdomain

27^o



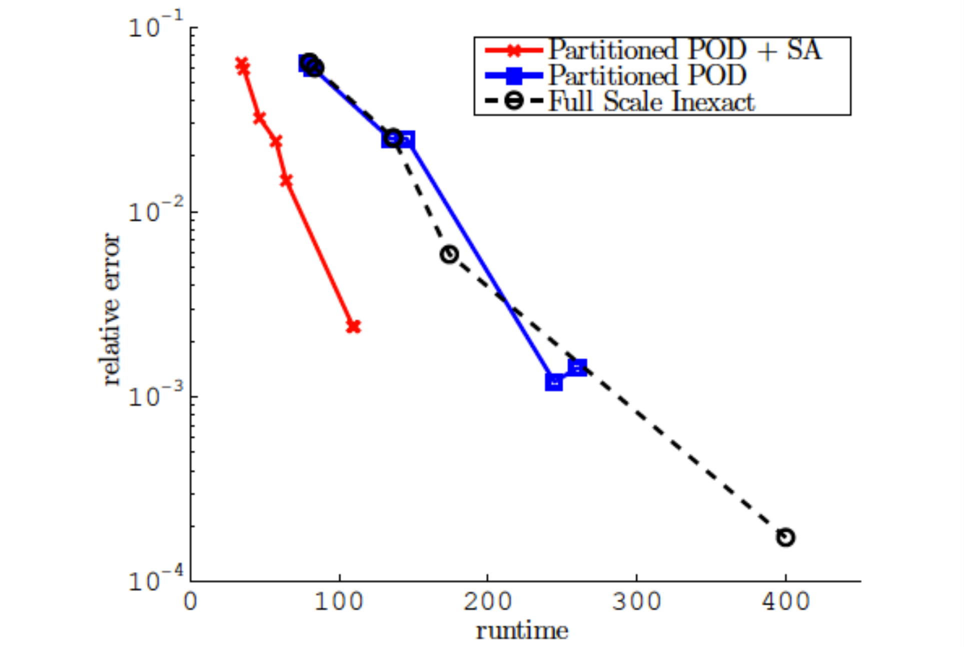
(b) Relative error for the different models using 256 nodes per subdomain

Performance: load angle 40 | 27 - 441 nodes

- Relative error

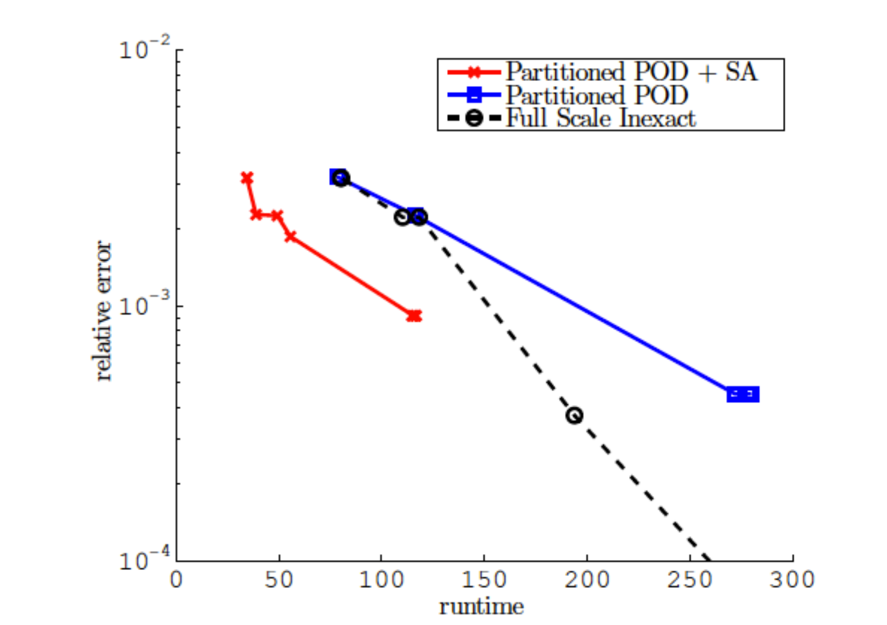
$$\nu^{\mathrm{app},(\mu)}\left(\underline{\mathbf{U}}^{\mathrm{app}}\right)^2 = \frac{\displaystyle\sum_{t_n\in\mathcal{T}^h} \|\underline{\mathbf{U}}^{\mathrm{app}}(t_n,\mu) - \underline{\mathbf{U}}^{\mathrm{ex}}(t_n,\mu)\|_2^2}{\displaystyle\sum_{t_n\in\mathcal{T}^h} \|\underline{\mathbf{U}}^{\mathrm{ex}}(t_n,\mu)\|_2^2}$$

40^o



(c) Relative error for the different models using 441 nodes per subdomain

27^o



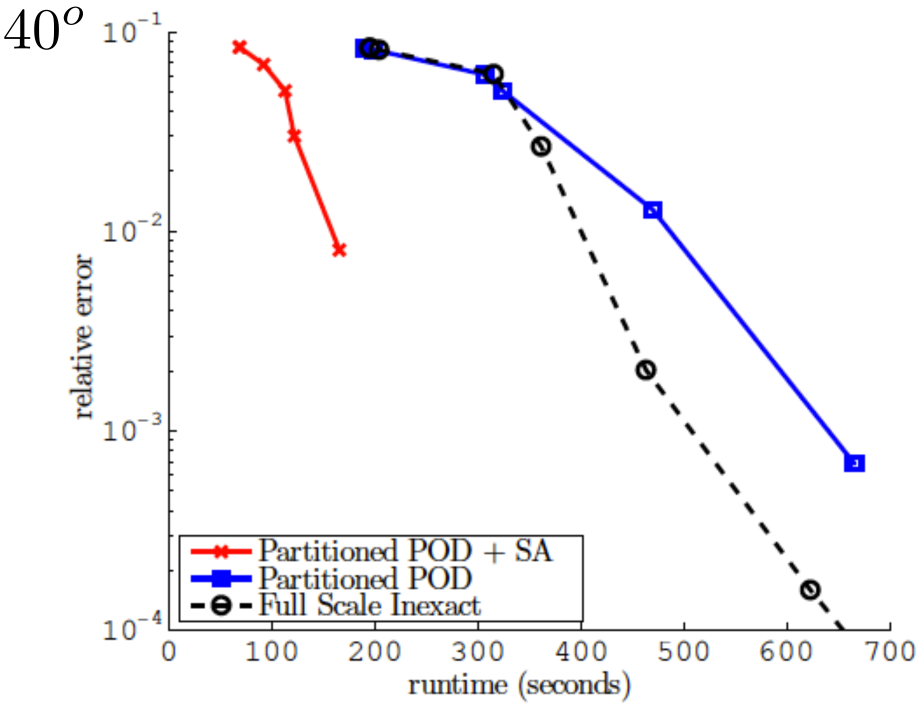
(c) Relative error for the different models using 441 nodes per subdomain

Performance: load angle 40 | 27 - 961 nodes

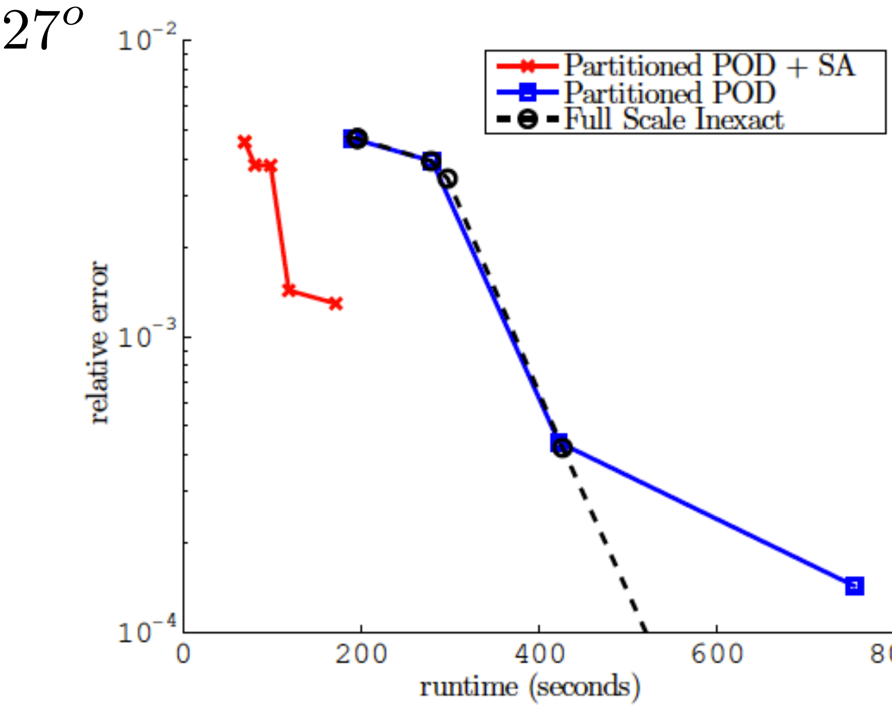
elative error

- Relative error

$$\nu^{\text{app},(\mu)}\left(\underline{\mathbf{U}}^{\text{app}}\right)^2 = \frac{\displaystyle\sum_{t_n\in\mathcal{T}^h} \left\|\underline{\mathbf{U}}^{\text{app}}(t_n,\mu)-\underline{\mathbf{U}}^{\text{ex}}(t_n,\mu)\right\|_2^2}{\displaystyle\sum_{t_n\in\mathcal{T}^h}\left\|\underline{\mathbf{U}}^{\text{ex}}(t_n,\mu)\right\|_2^2}$$



(d) Relative error for the different models using 961 nodes per subdomain



(d) Relative error for the different models using 961 per subdomain

Conclusions and perspectives

- Domain coupling using the primal Schur-complement domain decomposition method.
- Local subproblems have been reduced by projection in low-dimensional subspaces obtained by the snapshot POD.
- This approach permits to flexibly reduce the computational cost associated with highly nonlinear problems. In particular:
  - the **local reduced spaces are generated independently**, and have independent dimensions, which allows us to focus the numerical effort where it is most needed.
  - subdomains that are close to highly damaged zones need a richer model to account for the effect of topological changes. The local **POD transforms automatically generate local reduced spaces of larger dimension in these zones**.
  - the domain decomposition framework enables us to **switch from reduced local solvers to full local solvers** in a transparent manner. This is particularly useful for the subdomains that contain process zones, as a solution obtained by projection would be more expensive than a direct solution for a desirable accuracy.
  - the transition between ``offline'' and ``online'' computations becomes flexible. The **reduced models can be used in the zones where the local reduced spaces converge quickly** when enriching the snapshot space, while still computing snapshots and refining the reduced models via a direct local solver in the remaining subdomains.

Perspectives

- Further work related to domain decomposition
  - **load balancing** mismatch would occur when using such a strategy in parallel. CPUs which support domains that are not reduced, or domains for which the corresponding subproblems need to be projected in a space of relatively high dimension, would require to perform more operations. The domain partitioning itself should be performed jointly with the model reduction in order to distribute the load evenly.
  - **the interface problem itself was not reduced** here, to guarantee the interface kinematic compatibility.
    - ➡ Suboptimal reduced order model. Would generate expensive communications in parallel
    - ➡ A reduction of the interface problem using the POD can be done but is neither elegant nor easy
    - ➡ Dual Schur-complement domain decomposition method would allow the kinematic approximation of the subproblems to include the interface. However, this would only deflect the difficulty to the necessary reduction of the interface Lagrange multiplier space. This issue is our current direction of research.

Cutting in real time on GPU

- Dr Hadrien Courtecuisse

How can we build models fast?

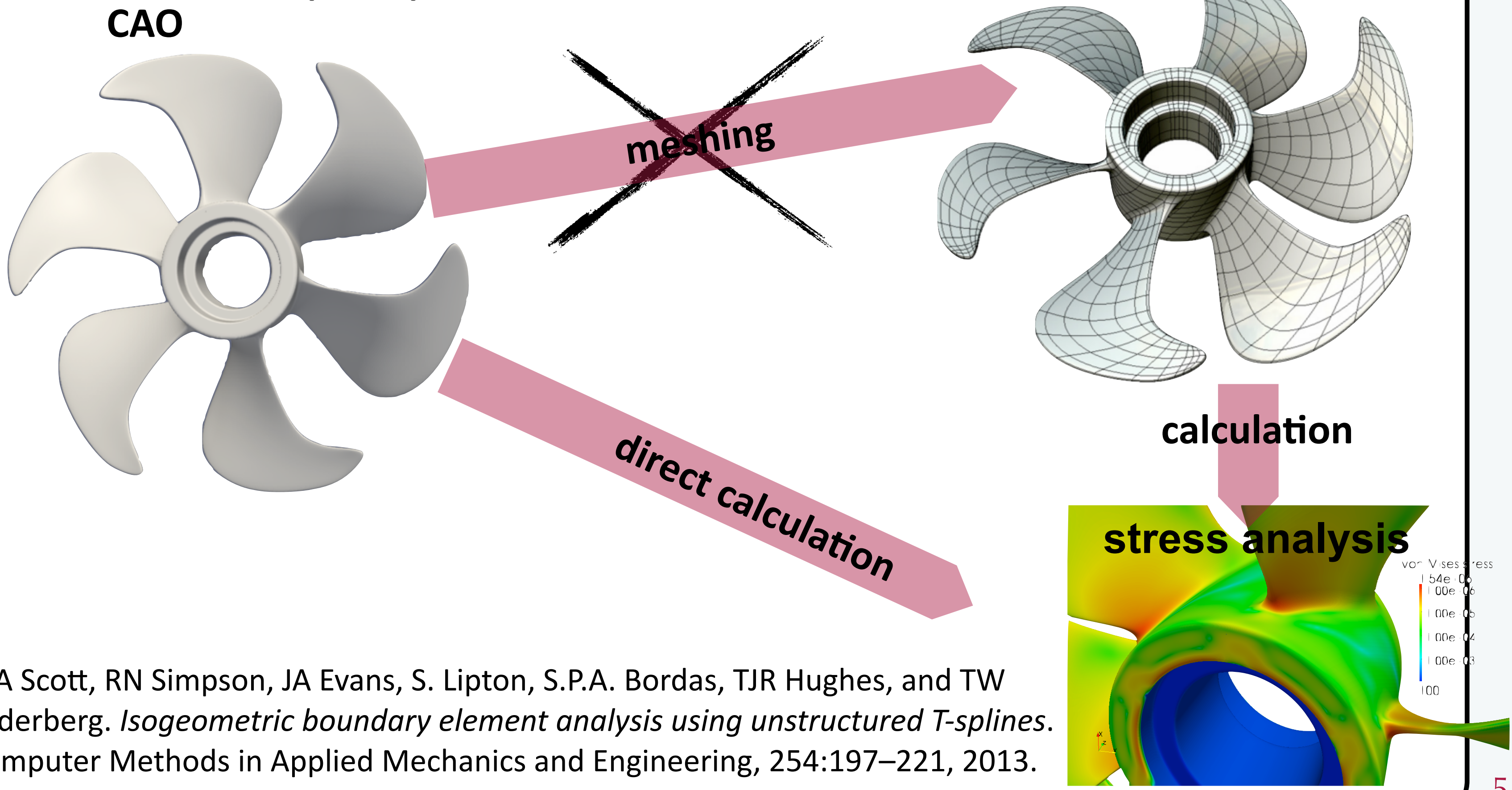
Theme: simplify the CAD->analysis transition (1/2)

Advanced discretization

NEW!

Isogeometric boundary element analysis

- complete suppression of mesh generation
- exact treatment of the geometry
- facilitates shape optimization

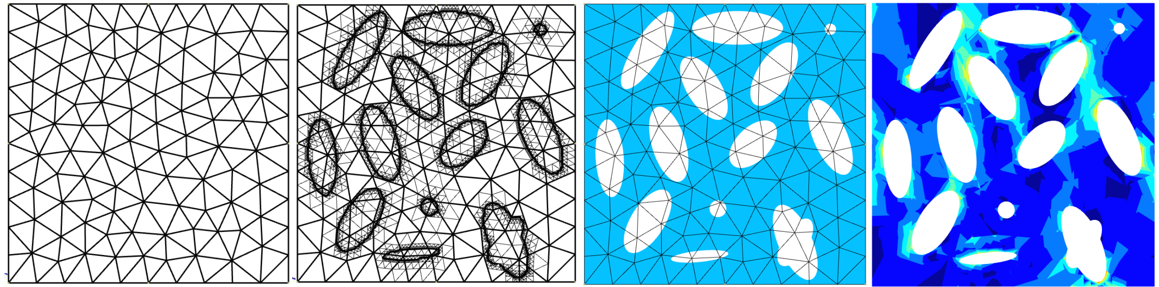


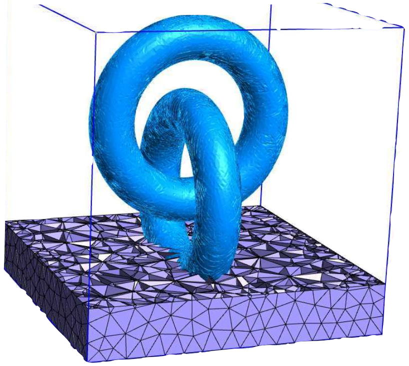
MA Scott, RN Simpson, JA Evans, S. Lipton, S.P.A. Bordas, TJR Hughes, and TW Sederberg. *Isogeometric boundary element analysis using unstructured T-splines*. Computer Methods in Applied Mechanics and Engineering, 254:197–221, 2013.

Simplify the CAD->analysis transition (2/2)

Implicit surface representation

- from CAD parameterization
- including vertices and sharp edges
- with error control



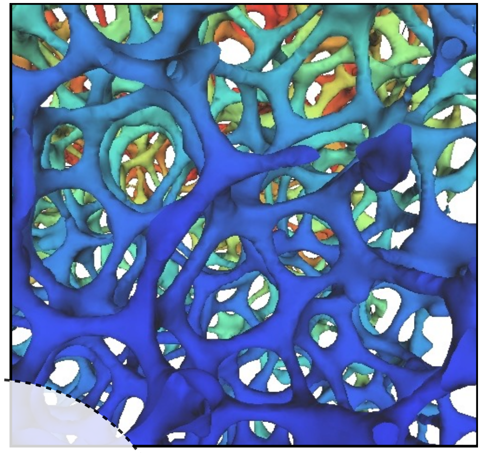


theses Moumnassi (w/ CRP)
et Nadal (w/ UPV)

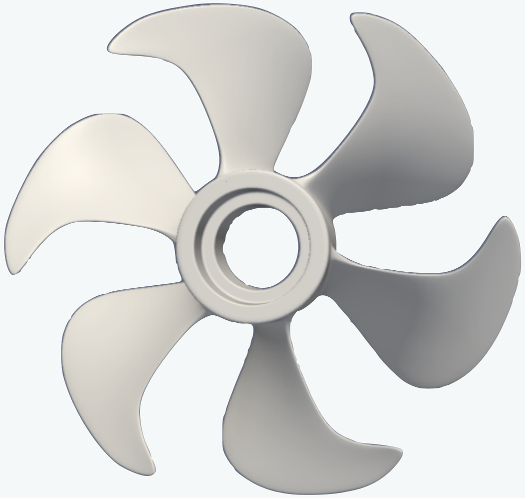
Image->num

NEW!

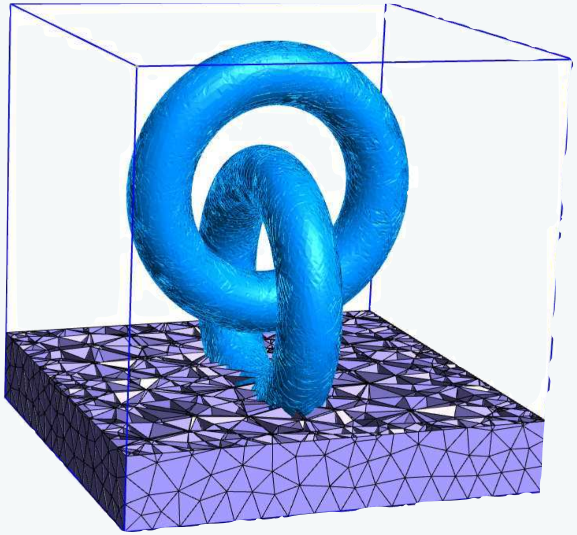
- CT image
- discretization
- computation



Advanced discretization

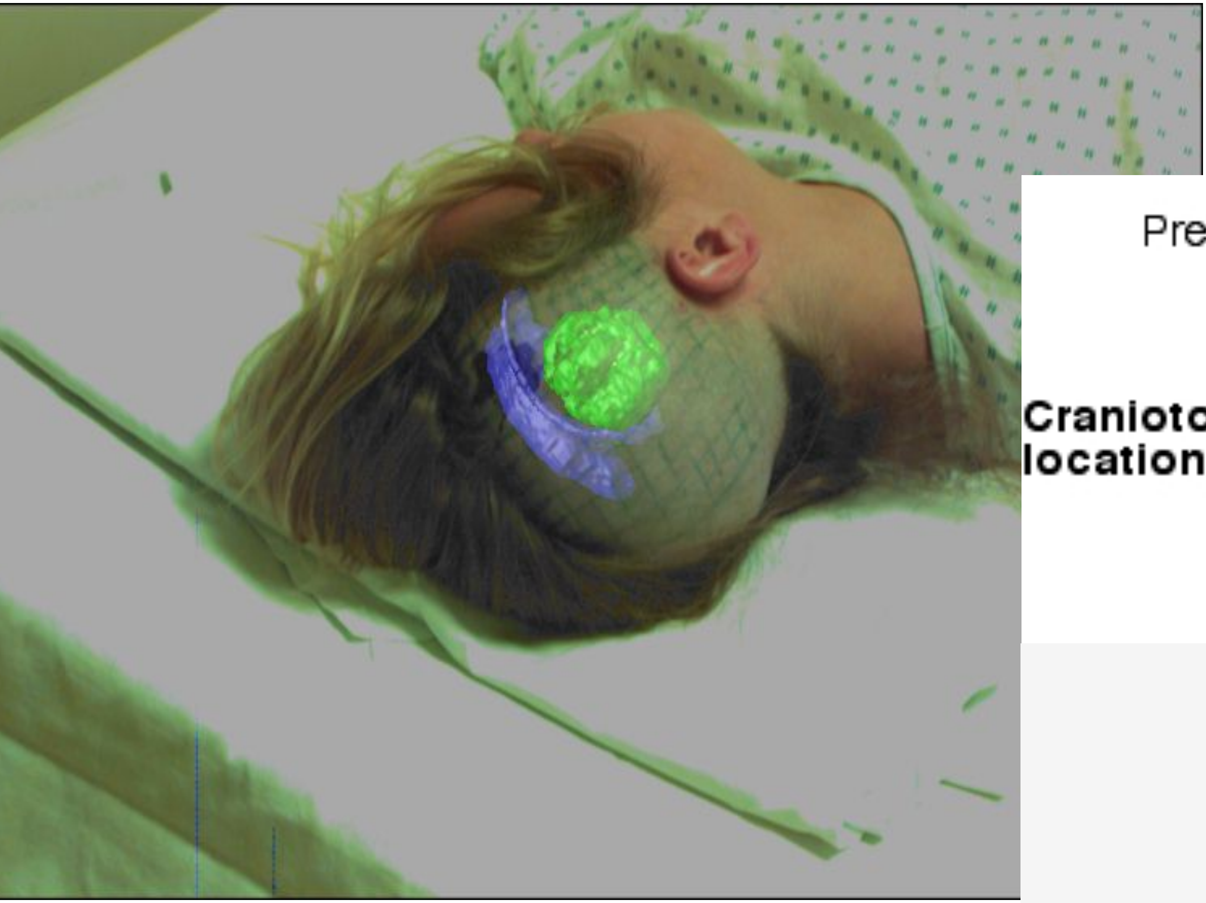


In other words, use the CAD as a model…

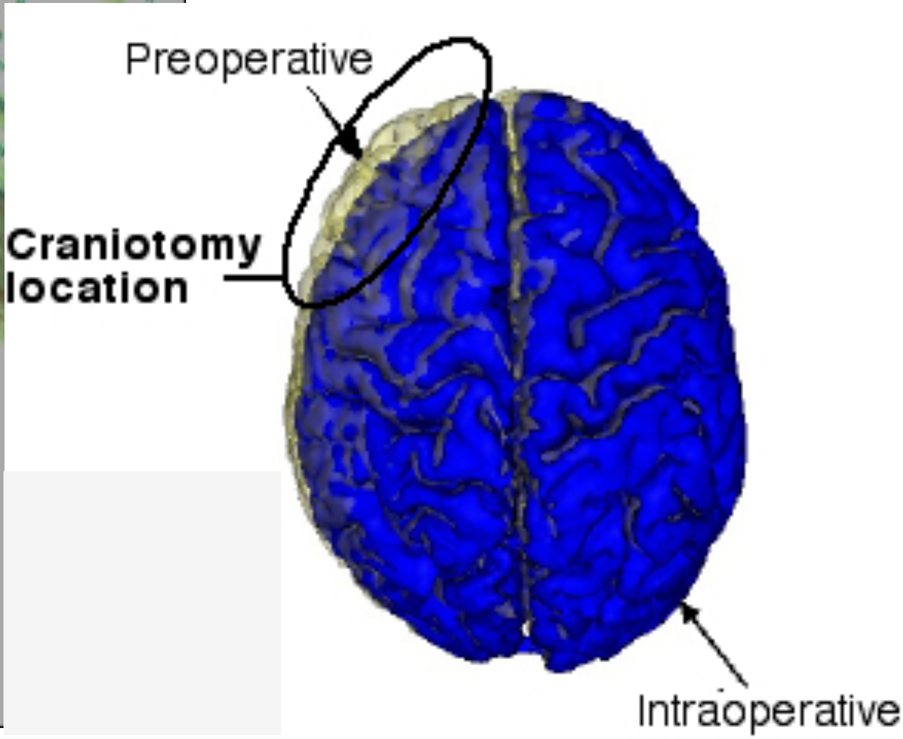


Our main motivation - image-guided neurosurgery

Image of brain tumour (green) is superimposed on patient as an aid to surgical planning and navigation

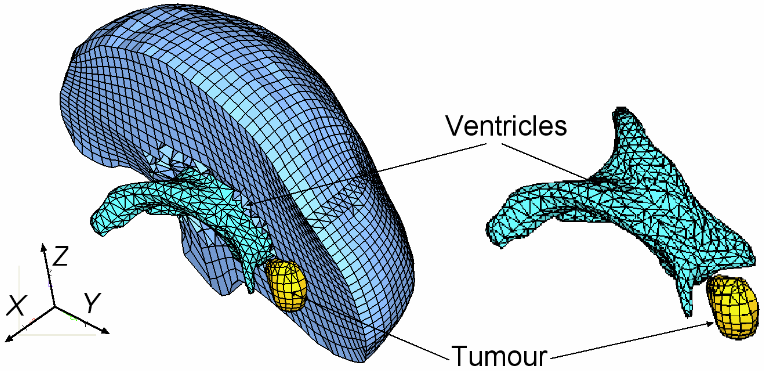


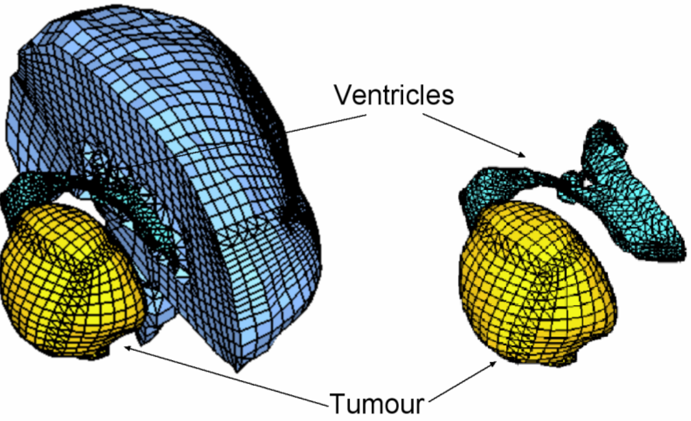
Courtesy of SPL, Harvard

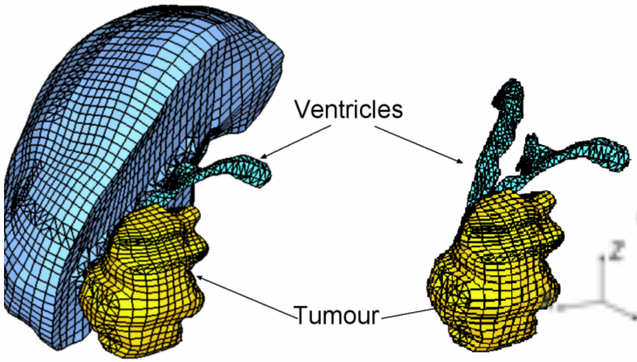


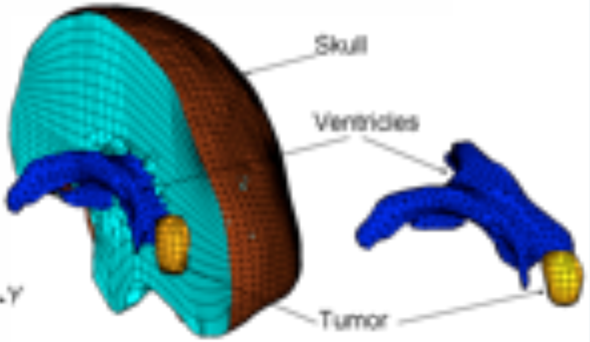
Patient-Specific Finite Element Meshes

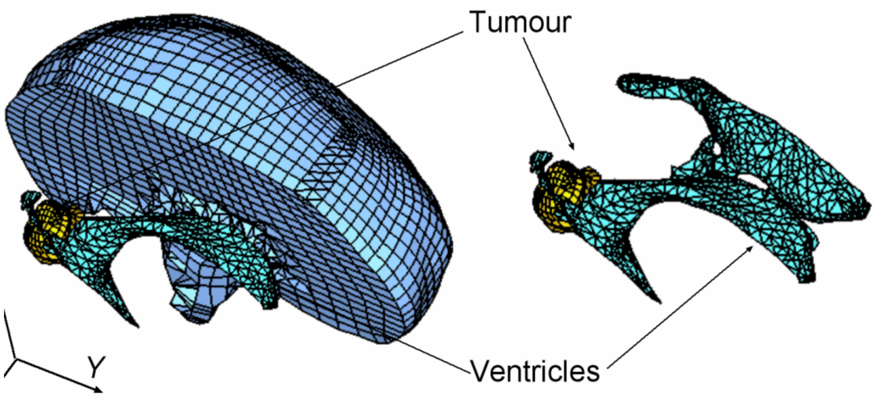
Joldes et al. (2009), *MICCAI 2009, Part II, LNCS 5762, pp. 300-307*

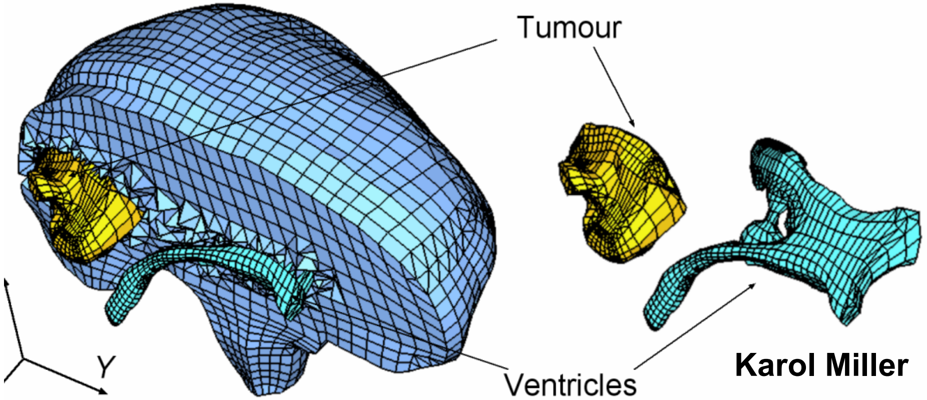




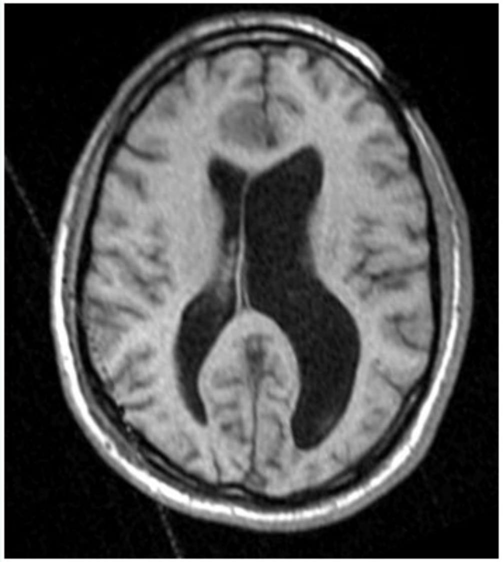




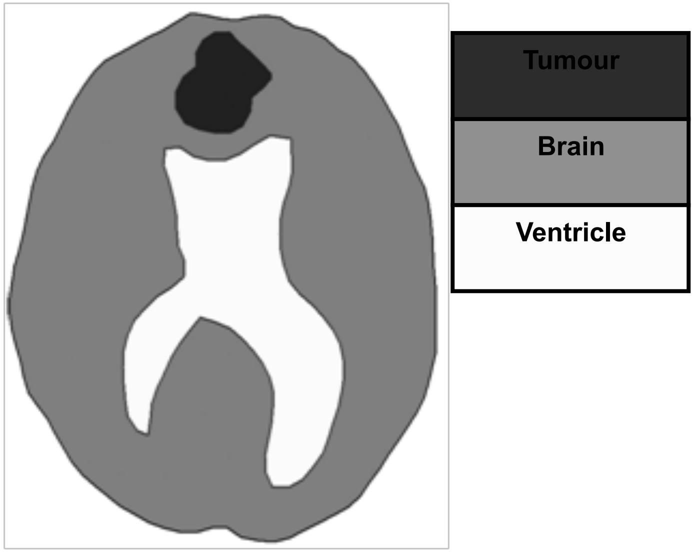




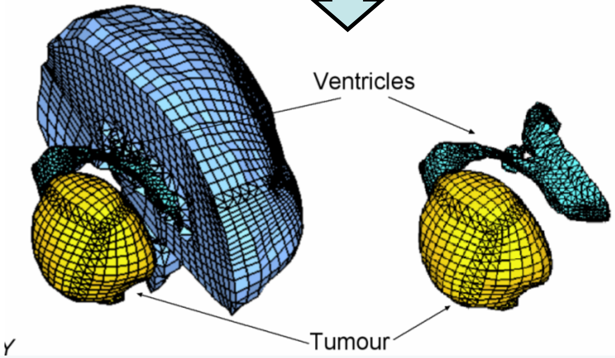
Neuroimage as a computational model?

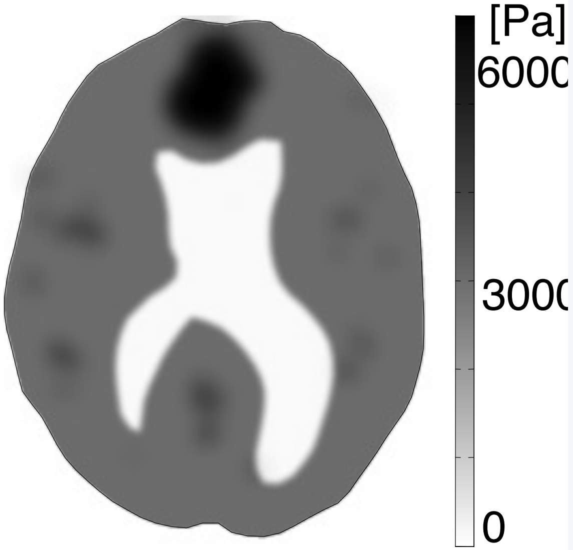


2D MRI slice



“Hard” segmentation

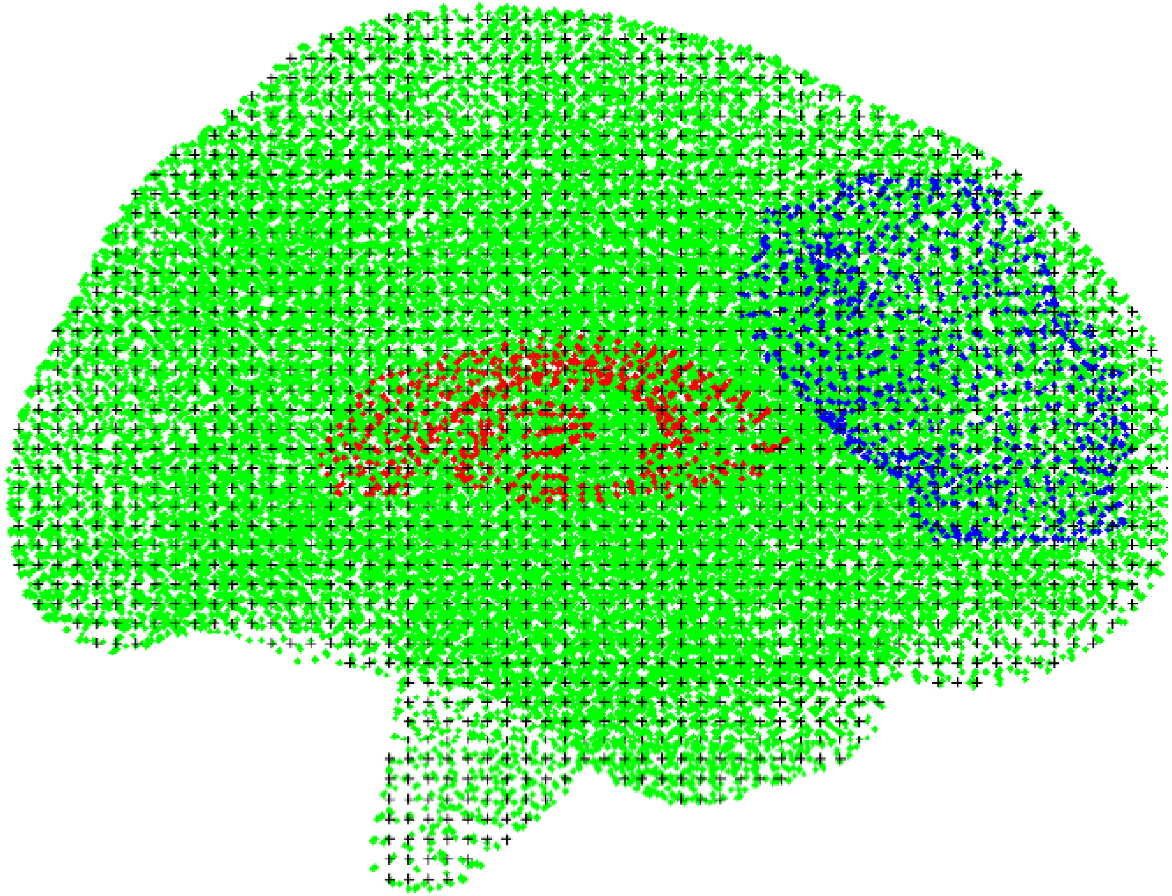




Assignment of mechanical properties based on statistical tissue classification

Karol Miller

3D patient-specific meshless computational grid of the brain

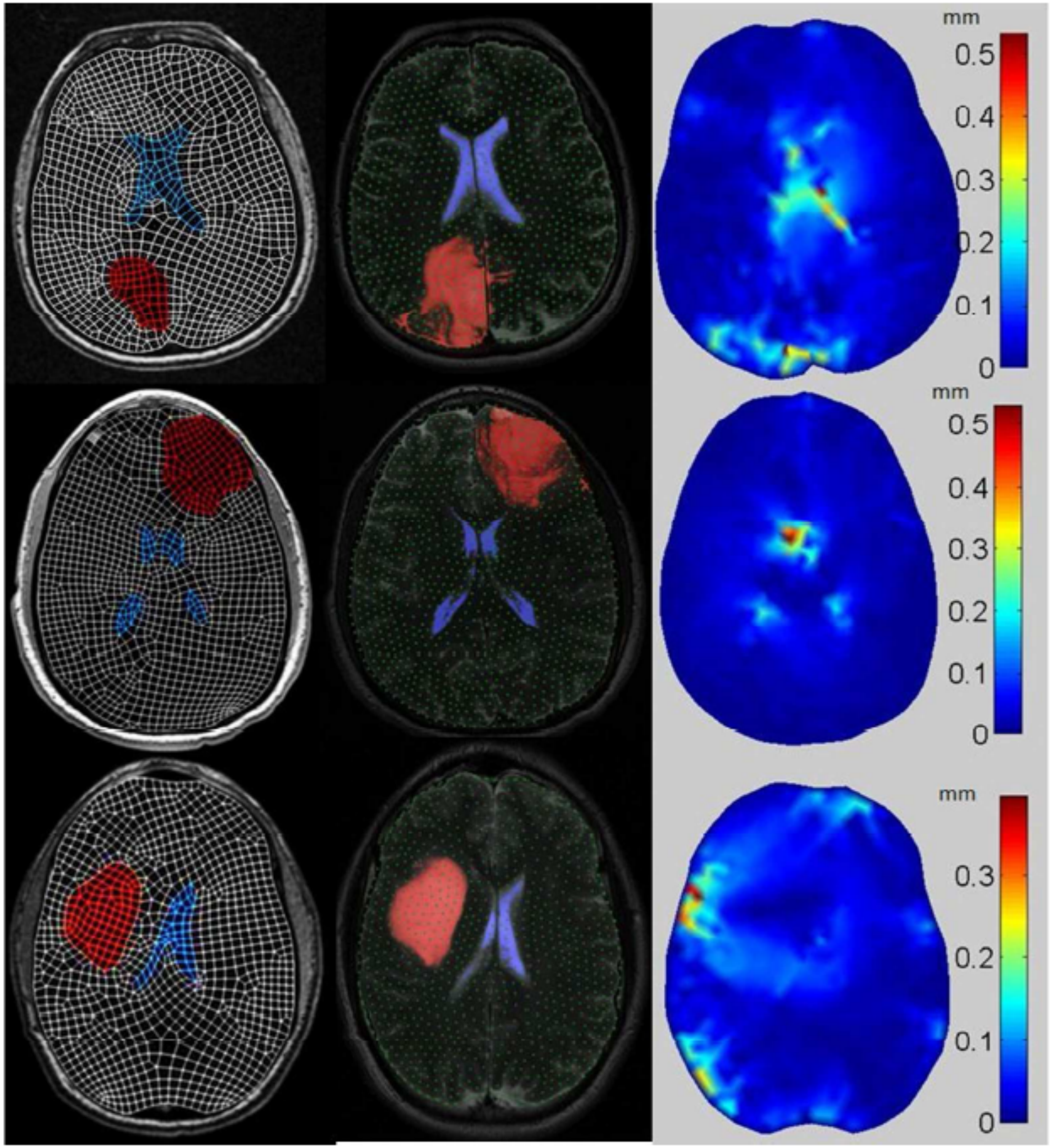


Green – parenchyma

Red – ventricles

Blue - tumour

Miller et al. (2012), *J. Biomech.* 45(15), pp. 2698-2701



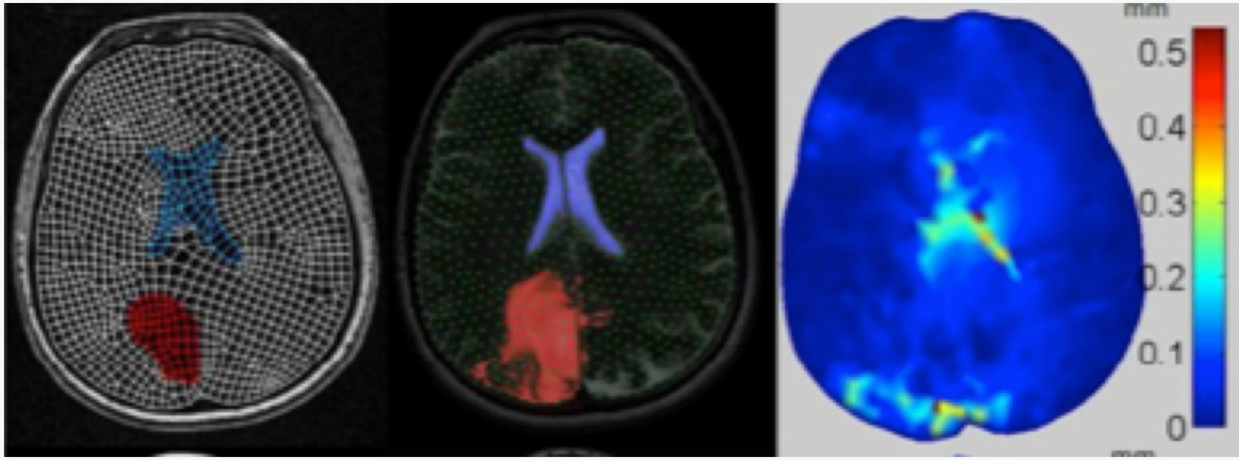
Evaluation of accuracy for three cases

Left column: Finite Element Models, with parenchyma, tumour (red) and ventricle (blue) modelled separately.

Middle column: Fuzzy Mesh-free Model without explicitly separating the tumour and ventricles, fuzzy tissue classifications of tumour (red), and ventricle (blue) are shown as cloud superimposed on the image; Nodes are shown as green dots.

Right column: Difference of the simulation results (computed deformation field) from the two models over the whole problem domain [mm].

Karol Miller



In other words, use the image as a model…

Yousef Ghaffari Motlagh

Chi Hoang

Hadrien Courtecuisse

Daniel Paladim

Xuan Peng

Danas Sutula

Dr. Nguyen Vinh Phu

Haojie Liang

Dr. Rob Simpson

Olivier Goury

Dr. Sundararajan Natarajan

Ahmad Akbari

Nguyen-Tanh Nhon

Chang-Kye Lee

Courtesy
PhD Com

Dr. Robert Simpson

Dr. Pierre Kerfriden

Research Area

Computational Mechanics & Computational Materials Science

UNIVERSITÉ DU LUXEMBOURG

Professor of Computational Mechanics

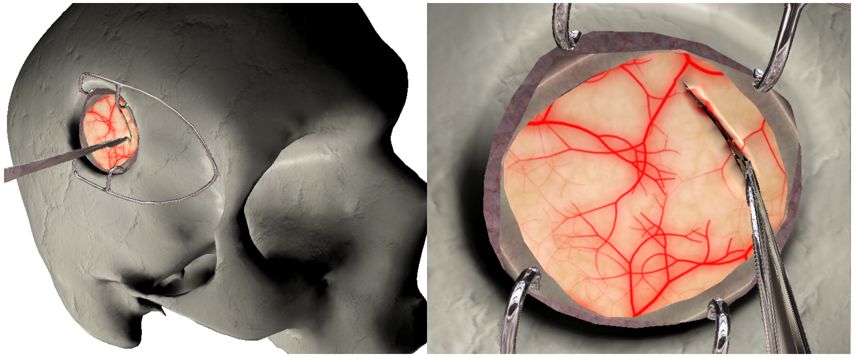
University of Luxembourg & Cardiff University

stephane.bordas@uni.lu, *bordasS@cardiff.ac.uk*

http://legato-team.eu

Principal Interests & Expertise

- Free and moving interfaces
- Multi-scale methods for fracture
- Multi-field coupled problems
- Inverse problems (Bayesian)
- Real-time methods, model order reduction
- Error estimation and simulation quality
- Meshless methods
- Extended Finite Element Methods



Real-time simulation of cutting during brain surgery

Representative Papers

- A partitioned model order reduction approach to **rationalise computational expenses** in nonlinear fracture mechanics, Computer Methods in Applied Mechanics and Engineering, (2013)
- **Isogeometric boundary** element analysis using unstructured T-splines, *Computer Methods in Applied Mechanics & Engineering* (2013)
- Effects of elastic strain energy and interfacial stress on the equilibrium morphology of **misfit particles** in heterogeneous solids, *J. Journal of the Mechanics & Physics of Solids* (2013)
- Real-time simulation of contact and **cutting of heterogeneous soft-tissues**, *Medical Image Analysis* (2014)
- A combined extended finite element and level set method for **biofilm** growth, Int. J. Num. Meth. Engng. (2008)

Thank you very much for your kind attention.

CISM school on Theoretical

Computational and Experimental Fracture Mechanics

in Udine, Italy, September 2015

Contact

stephane.bordas@uni.lu / gmail.com / alum.northwestern.edu

http://www.legato-team.eu

Model

- Lattice model
  - Euler-Bernoulli beams
  - Elastic damageable constitutive law
    - Helmholtz free energy:

$$e_d = \frac{1}{2}\left(ES\left(1 - d_1\frac{< u_{,s} >_+}{u_{,s}}\right)u_{,s}^2 + EI\left(1 - d_2\right)\theta_{,s}^2\right)$$

    - Generalised stress and Driving forces

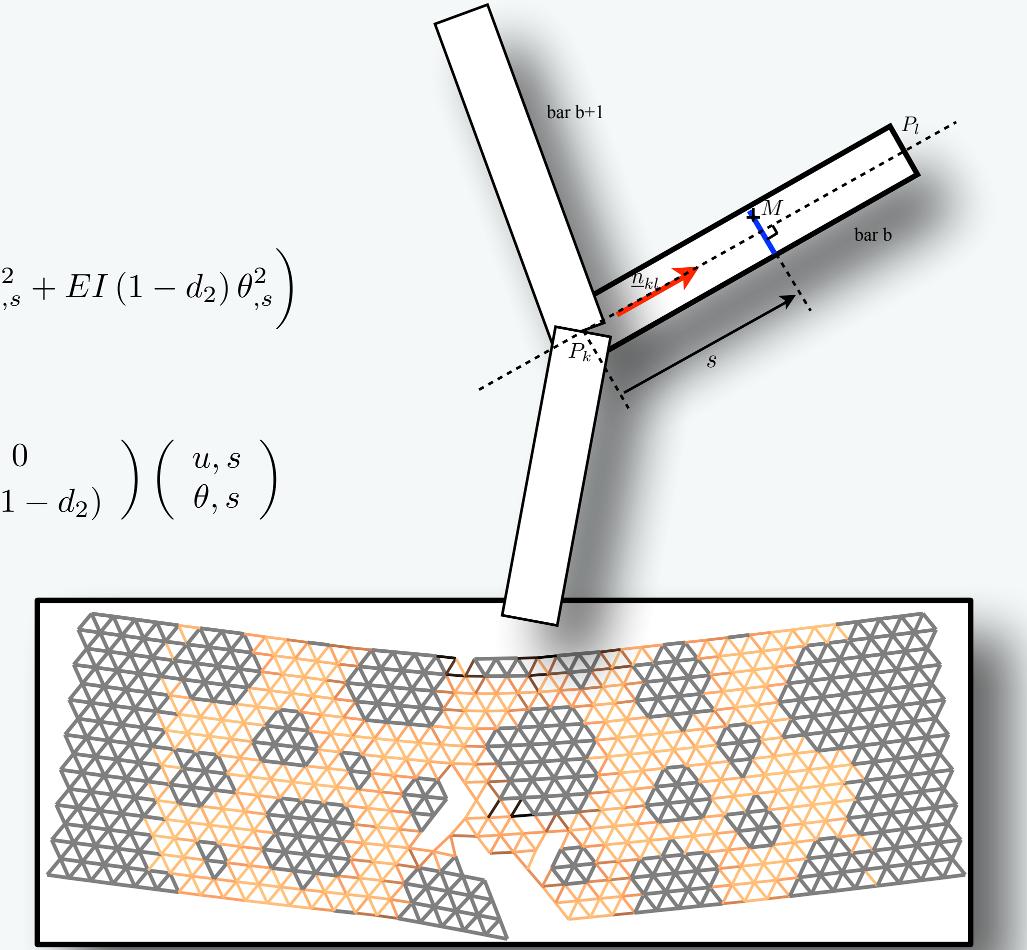
$$\begin{pmatrix} N \\ M \end{pmatrix} = \begin{pmatrix} ES(1 - d_1\frac{<u_{,s}>_+}{u_{,s}}) & 0 \\ 0 & EI\left(1 - d_2\right) \end{pmatrix}\begin{pmatrix} u,s \\ \theta,s \end{pmatrix}$$

$$\begin{cases} Y_1 = -\frac{\partial e_d}{\partial d_1} = ES\,\frac{<u_{,s}>_+}{u_{,s}}\,u_{,s}^2 \\ Y_2 = -\frac{\partial e_d}{\partial d_2} = EI\,\theta_{,s}^2 \end{cases}$$

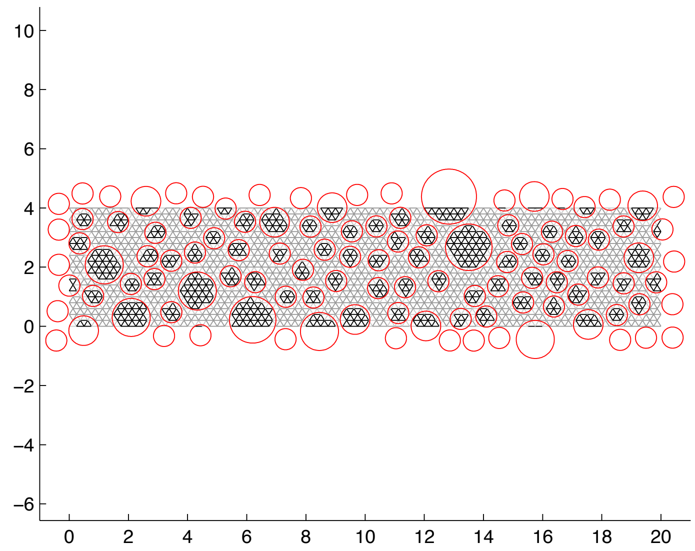
    - Evolution law (non-associated model)

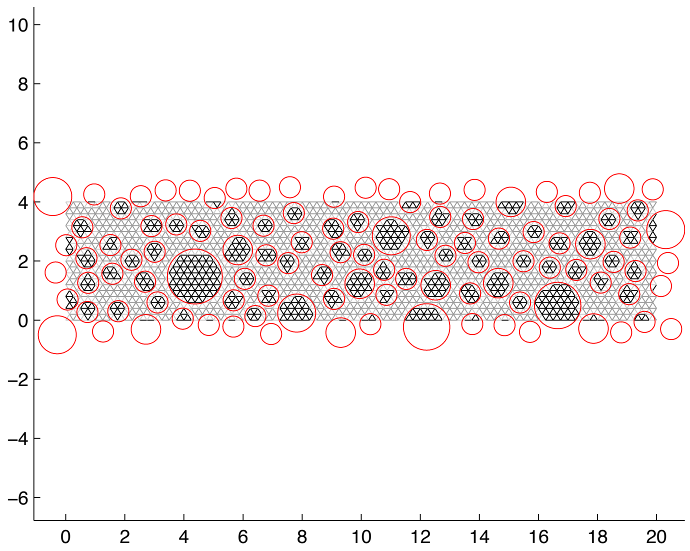
$$Y = \left(Y_1^\alpha + \gamma\,Y_2^\alpha\right)^{1/\alpha}$$

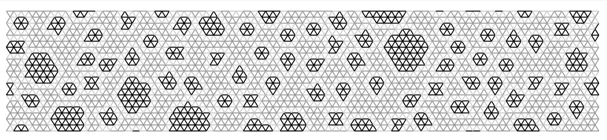
$$d = d_1 = d_2 = \left(\frac{Y - Y_0}{Y_c - Y_0}\right)^\beta$$

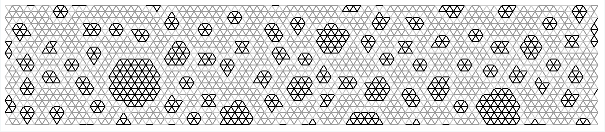


- Random packing following Fuller distribution curve
  - Particles 10 times stiffer than interface and matrix, no damage
  - fracture toughness of interface 4 times lower than matrix


10
8
6
4
2
0
−2
−4
−6
0
2
4
6
8
10
12
14
16
18
20


10
8
6
4
2
0
−2
−4
−6
0
2
4
6
8
10
12
14
16
18
20





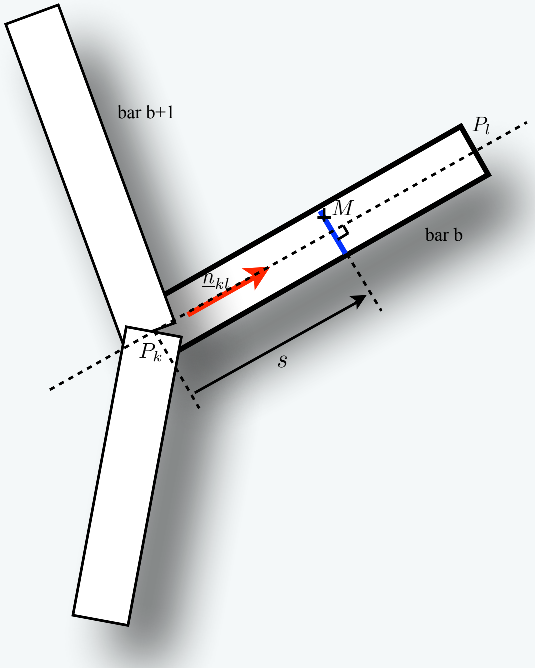
Nonlinear solver

- Discretisation
  - space: third order polynomial for deflection, first order for normal displacement

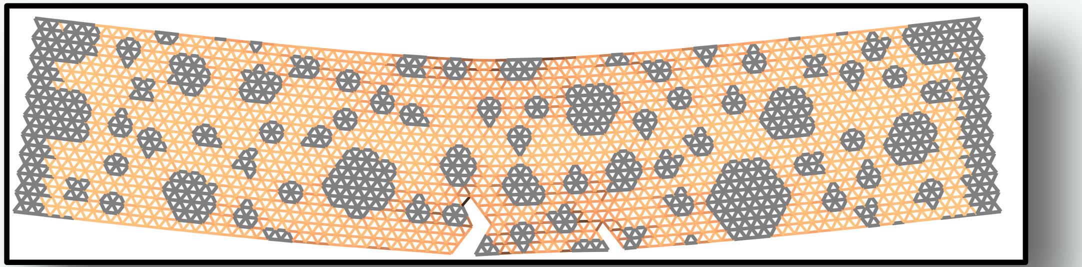
$$\mathcal{U}^h(\Omega) = \left\{ \underline{u}(\underline{x}) \mid \underline{u}(\underline{x}) = \sum_{i=1}^{n_n} N_i(\underline{x})\, \underline{u}_i \right\}$$

  - timefind $(\underline{u}_{|t_n})_{n\in[\![0,n_t]\!]}$

$$\boxed{\underline{\mathbf{F}}_{\text{Int}}\left(\underline{\mathbf{\Delta U}}, \left(\underline{\mathbf{U}}_{|t_m}\right)_{m\in[\![0,n]\!]}, \mu\right) + \underline{\mathbf{F}}_{\text{Ext}}\left(t_{n+1}, \mu\right) = \underline{\mathbf{0}}}$$



- Newton / arc-length / Line Search [Lorentz and Badel '04, Alfano and Crisfield '03]



Projection-based Model Order Reduction

- Non-linear, time dependant (rate-independent), parameter dependent set of balance equations

$$\boxed{\underline{\mathbf{F}}_{\text{Int}}\left(\underline{\Delta\mathbf{U}}, \left(\underline{\mathbf{U}}_{|t_m}\right)_{m\in[\![0,n]\!]}, \mu\right) + \underline{\mathbf{F}}_{\text{Ext}}\left(t_{n+1},\mu\right) = \underline{\mathbf{0}}}$$

- Approximation of the solution in a reduced space:

$$\boxed{\underline{\Delta\mathbf{U}}_{|t_{n+1}} = \sum_{i=1}^{n_s} \underline{\mathbf{C}}^i \alpha^i = \underline{\underline{\mathbf{C}}}\,\underline{\boldsymbol{\alpha}}}$$

where

$$\underline{\Delta\mathbf{U}}_{|t_n+1} = \underline{\mathbf{U}}_{|t_{n+1}} - \underline{\mathbf{U}}_{|t_n}$$

$$\underline{\underline{\mathbf{C}}} = \begin{pmatrix} \underline{\mathbf{C}}^1 & \underline{\mathbf{C}}^2 & \dots & \underline{\mathbf{C}}^{n_c} \end{pmatrix}$$

Approximation of the solution increment in a space of small dimension (n_c).

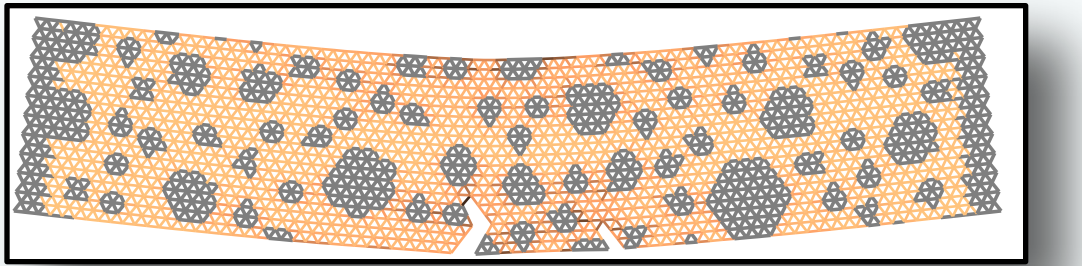
Global vectors of nodal values corresponding to piecewise polynomial global basis functions, as opposed to locally supported in finite element).

- Approximation of the balance equations (Galerkin for instance)

$$\underline{\mathbf{F}}_{\text{Ext}} + \underline{\mathbf{F}}_{\text{Int}}\left(\underline{\mathbf{U}}_{|t_n} + \underline{\underline{\mathbf{C}}}\,\underline{\boldsymbol{\alpha}}\right) = \underline{\mathbf{R}}_{|t_n} \xrightarrow{\text{Galerkin}} \underline{\underline{\mathbf{C}}}^T\left(\underline{\mathbf{F}}_{\text{Ext}} + \underline{\mathbf{F}}_{\text{Int}}\left(\underline{\mathbf{U}}_{|t_n} + \underline{\underline{\mathbf{C}}}\,\underline{\boldsymbol{\alpha}}\right)\right) = \underline{\mathbf{0}}$$

n_c balance equations

Model Order Reduction using Snapshot POD



$$\underline{\underline{\mathbf{C}}}^T \left(\underline{\mathbf{F}}_{\text{Ext}} + \underline{\mathbf{F}}_{\text{Int}}(\underline{\mathbf{U}}_{|t_n} + \underline{\underline{\mathbf{C}}}\,\underline{\boldsymbol{\alpha}})\right) = \underline{\mathbf{0}}$$

- **Solver at each time increment:** Newton / arc-length / Line Search (idem initial problem)

$$\boxed{\underline{\underline{\mathbf{K}}}^i_{\text{T,R}}\,\underline{\delta\boldsymbol{\alpha}}^{i+1} = -\underline{\mathbf{R}}^i_{\text{R}}}$$

- tangent of the reduced balance equations:

$$\underline{\underline{\mathbf{K}}}^i_{\text{T,R}} = \underline{\underline{\mathbf{C}}}^T \left.\frac{\partial \underline{\mathbf{F}}_{\text{Int}}(\underline{\mathbf{U}}_{|t_n} + \underline{\underline{\mathbf{C}}}\,\underline{\boldsymbol{\alpha}})}{\partial \underline{\boldsymbol{\alpha}}}\right|_{\underline{\boldsymbol{\alpha}}=\underline{\boldsymbol{\alpha}}^i}$$

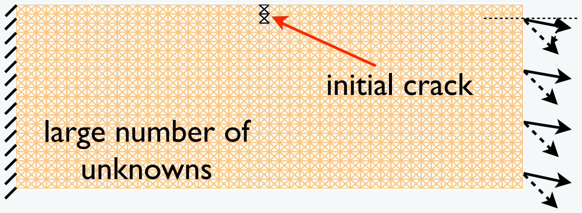
- residual of the reduced balance equations:

$$\underline{\mathbf{R}}^i_{\text{R}} = \underline{\underline{\mathbf{C}}}^T \left(\underline{\mathbf{F}}_{\text{Ext}} + \underline{\mathbf{F}}_{\text{Int}}(\underline{\mathbf{U}}_{|t_n} + \underline{\underline{\mathbf{C}}}\,\underline{\boldsymbol{\alpha}}^i)\right)$$

Projection-based Model Order Reduction

$\theta \in [15^o \; 45^o]$

Parametrised fracture model

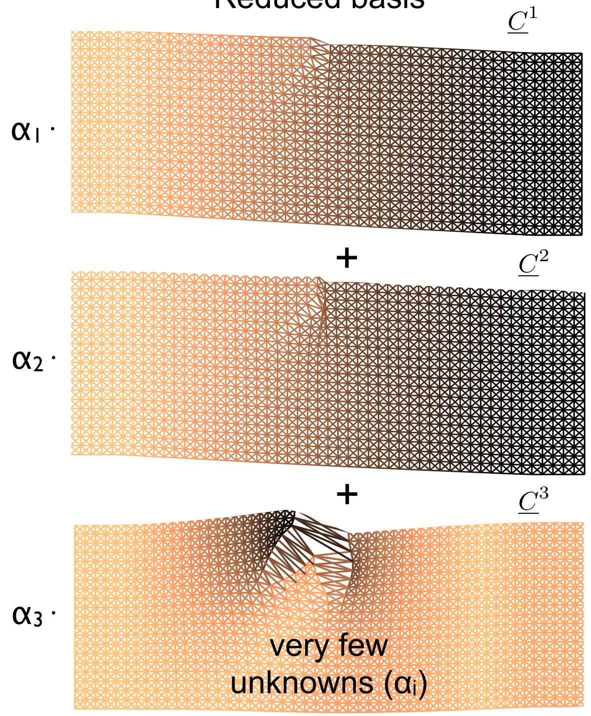


initial crack

large number of unknowns

Construction of reduced order model

Reduced basis



very few unknowns (α_i)

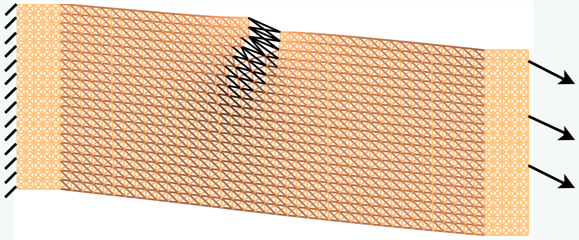
Reduced system

$$\underline{\underline{\mathbf{C}}}^T \left(\underline{\mathbf{F}}_{\text{Ext}} + \underline{\mathbf{F}}_{\text{Int}} (\underline{\mathbf{U}}_{|t_n} + \underline{\underline{\mathbf{C}}}\, \underline{\boldsymbol{\alpha}}) \right) = \underline{\mathbf{0}}$$

very few equations, **possibly** cheap solution

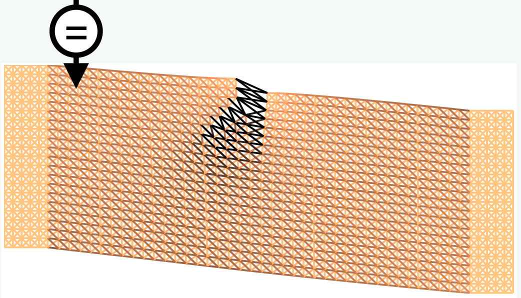
Solution at arbitrary angle using the reduced model

$\theta = 30^o$



approximated by

$\approx$



Questions

- **Choice of reduced space**
  - basis vectors?
  - dimension
- **Choice of system approximation**
- **Error control**

➡ Should be driven by **optimality**, consistency and stability considerations [Farhat ’09], in order to achieve a **significant speed-up**

Method of separated representation

- Search for the solution in space / time / parameter in a product space:

$$\boxed{\begin{aligned} \underline{\bar{\mathbf{U}}} : \quad & \mathcal{U}_{\text{sep}} = \mathbb{R}^n \times \mathcal{T} \times \mathcal{P} \to \mathbb{R}^n \\ & \underline{\bar{\mathbf{U}}}(t,\mu) = \sum_{i=1}^{n_C} \underline{\mathbf{C}}_i \, \beta_i(t) \gamma_i(\mu) \, , \end{aligned}}$$

$$\begin{aligned} & \underline{\mathbf{C}}^i \in \mathbb{R}^n \\ & \beta^i : \mathcal{T} \to \mathbb{R}, \quad \forall i \in [\![1, n_C]\!] \, , \\ & \gamma^i : \mathcal{P} \to \mathbb{R}, \quad \forall i \in [\![1, n_C]\!] \, , \end{aligned}$$

Proper Orthogonal Decomposition (POD)

- Search for the solution in a product space of **two** spaces (space / time):

$$\boxed{\begin{aligned} \underline{\bar{\mathbf{U}}} : \quad & \mathcal{V} = \mathbb{R}^n \times \mathcal{T} \to \mathbb{R}^n \\ & \underline{\bar{\mathbf{U}}}(t) = \sum_{i=1}^{n_C} \underline{\mathbf{C}}_i \, \alpha_i(t) = \underline{\underline{\mathbf{C}}} \, \underline{\boldsymbol{\alpha}}(t) \end{aligned}}$$

- that minimises a distance with respect to the exact solution (POD (Pearson ’01, Hotelling ’33, Karhunen ‘47, Loeve ’63))

$$\boxed{\tilde{J}_{\text{POD}}(\underline{\underline{\mathbf{C}}}, \alpha) = \| \underline{\mathbf{U}} - \underline{\bar{\mathbf{U}}} \|_{\mathcal{V}}}$$

  - with the norm on $\mathcal{V}$: $\| \underline{\mathbf{X}} \|_{\mathcal{V}} = \int_{t \in \mathcal{T}} \| \underline{\mathbf{X}}(t) \|_2^2 \, dt$
  - ill-posed, add the constraint $\underline{\underline{\mathbf{C}}}^T \underline{\underline{\mathbf{C}}} = \underline{\underline{\mathbf{I}}}_d$

Proper Orthogonal Decomposition (POD)

- Search for the solution in a product space of **two** spaces (space / time):

$$\boxed{\begin{aligned} &\underline{\bar{\mathbf{U}}}: \quad \mathcal{V} = \mathbb{R}^n \times \mathcal{T} \to \mathbb{R}^n \\ &\qquad \underline{\bar{\mathbf{U}}}(t) = \sum_{i=1}^{n_C} \underline{\mathbf{C}}_i \, \alpha_i(t) = \underline{\underline{\mathbf{C}}} \, \underline{\boldsymbol{\alpha}}(t) \end{aligned}}$$

- that minimises a distance with respect to the exact solution (POD (Pearson '01, Hotelling '33, Karhunen '47, Loeve '63))

$$\boxed{\tilde{J}_{\mathrm{POD}}(\underline{\underline{\mathbf{C}}}, \alpha) = \|\underline{\mathbf{U}} - \underline{\bar{\mathbf{U}}}\|_{\mathcal{V}}}$$

  - with the norm on $\mathcal{V}$: $\|\underline{\mathbf{X}}\|_{\mathcal{V}} = \int_{t \in \mathcal{T}} \|\underline{\mathbf{X}}(t)\|_2^2 \, dt$
  - ill-posed, add the constraint $\underline{\underline{\mathbf{C}}}^T \underline{\underline{\mathbf{C}}} = \underline{\underline{\mathbf{I}}}_d$

- Set the functional derivative $\dfrac{\delta \tilde{J}_{\mathrm{POD}}}{\delta \underline{\boldsymbol{\alpha}}(t)} = \underline{\mathbf{0}}$

$$\boxed{\underline{\boldsymbol{\alpha}} = \underline{\underline{\mathbf{C}}}^T \underline{\mathbf{U}}}$$

$\underline{\bar{\mathbf{U}}}$ is the orthogonal projection of $\underline{\mathbf{U}}$ onto Im(C)

Multi-dimension: GPOD and Multilinear POD / SVD

- Global POD (not very fancy but straightforward)

$$\boxed{\begin{aligned} &\bar{\underline{\mathbf{U}}} : \quad \mathcal{V} = \mathbb{R}^n \times (\mathcal{T} \times \mathcal{P}) \to \mathbb{R}^n \\ &\bar{\underline{\mathbf{U}}}(t,\mu) = \sum_{i=1}^{n_C} \underline{\mathbf{C}}_i\, \alpha_i(t,\mu) = \underline{\underline{\mathbf{C}}}\, \underline{\boldsymbol{\alpha}}(t,\mu) \end{aligned}}$$

- Miminise $J_{\mathrm{GPOD}}(\underline{\underline{\mathbf{C}}}) = \displaystyle\int_{\mu\in\mathcal{P}} \int_{t\in\mathcal{T}} \|\underline{\mathbf{U}}(t,\mu) - \underline{\underline{\mathbf{C}}}\,\underline{\underline{\mathbf{C}}}^T \underline{\mathbf{U}}(t,\mu)\|_2^2 \, dt\, d\mu$ $\qquad \underline{\underline{\mathbf{C}}}^T \underline{\underline{\mathbf{C}}} = \underline{\underline{\mathbf{I}}}_d$

$$\underline{\underline{\mathbf{C}}} = \begin{pmatrix} \underline{\phi}^1 & \underline{\phi}^2 & \dots & \underline{\phi}^{n_c} \end{pmatrix} \qquad \underline{\underline{\mathbf{K}}}\, \underline{\phi}^k = \lambda^k \underline{\phi}^k \qquad \underline{\underline{\mathbf{K}}} = \int_{\mu\in\mathcal{P}} \int_{t\in\mathcal{T}} \underline{\mathbf{U}}(t,\mu)\, \underline{\mathbf{U}}(t,\mu)\, dt\, d\mu$$

$(\lambda^k)_{k\in[\![0,n]\!]}$ in decreasing order

Multi-dimension: GPOD and Multilinear POD / SVD

- Global POD (not very fancy but straightforward)

$$\underline{\bar{\mathbf{U}}} : \quad \mathcal{V} = \mathbb{R}^n \times (\mathcal{T} \times \mathcal{P}) \to \mathbb{R}^n$$

$$\underline{\bar{\mathbf{U}}}(t,\mu) = \sum_{i=1}^{n_C} \underline{\mathbf{C}}_i\, \alpha_i(t,\mu) = \underline{\underline{\mathbf{C}}}\,\underline{\boldsymbol{\alpha}}(t,\mu)$$

  - Miminise $J_{\text{GPOD}}(\underline{\underline{\mathbf{C}}}) = \displaystyle\int_{\mu\in\mathcal{P}}\int_{t\in\mathcal{T}} \|\underline{\mathbf{U}}(t,\mu) - \underline{\underline{\mathbf{C}}}\,\underline{\underline{\mathbf{C}}}^T\underline{\mathbf{U}}(t,\mu)\|_2^2\, dt\, d\mu \qquad \underline{\underline{\mathbf{C}}}^T\underline{\underline{\mathbf{C}}} = \underline{\underline{\mathbf{I}}}_d$

➡ $\underline{\underline{\mathbf{C}}} = \begin{pmatrix} \underline{\phi}^1 & \underline{\phi}^2 & \dots & \underline{\phi}^{n_c} \end{pmatrix} \qquad \underline{\underline{\mathbf{K}}}\,\underline{\phi}^k = \lambda^k \underline{\phi}^k \qquad \underline{\underline{\mathbf{K}}} = \displaystyle\int_{\mu\in\mathcal{P}}\int_{t\in\mathcal{T}} \underline{\mathbf{U}}(t,\mu)\,\underline{\mathbf{U}}(t,\mu)\, dt\, d\mu$

$(\lambda^k)_{k\in[\![0,n]\!]}$ in decreasing order

- Multilinear POD: Search for the solution in space / time / parameter in a product space:

$$\underline{\bar{\mathbf{U}}} : \quad \mathcal{U}_{\text{sep}} = \mathbb{R}^n \times \mathcal{T} \times \mathcal{P} \to \mathbb{R}^n$$

$$\underline{\bar{\mathbf{U}}}(t,\mu) = \sum_{i=1}^{n_C} \underline{\mathbf{C}}_i\, \beta_i(t)\gamma_i(\mu)\,,$$

  - minimise $\tilde{J}_{\text{MLPOD}}(\underline{\underline{\mathbf{C}}}) = \displaystyle\int_{\mu\in\mathcal{P}}\int_{t\in\mathcal{T}} \|\underline{\mathbf{U}}(t,\mu) - \underline{\bar{\mathbf{U}}}(t,\mu)\|_2^2\, dt\, d\mu$

➡ interesting for separation time / parameter in fracture, TODO

One word about the Proper Generalized Decomposition

- If one does not want to calculate the exact solution, or cannot obtain it (intractable)

$$\boxed{\begin{aligned}\underline{\bar{\mathbf{U}}} : \quad & \mathcal{U}_{\text{sep}} = \mathbb{R}^n \times \mathcal{T} \times \mathcal{P} \rightarrow \mathbb{R}^n \\ & \qquad \underline{\bar{\mathbf{U}}}(t,\mu) = \sum_{i=1}^{n_C} \underline{\mathbf{C}}_i \, \beta_i(t) \gamma_i(\mu) \, ,\end{aligned}}$$

- Least-square PGD

$$J(\underline{\underline{\mathbf{C}}}, \underline{\beta}, \underline{\Gamma}) = \text{argmin} \|\underline{\mathbf{R}}\|_{\mathcal{V}}^2$$

$$J(\underline{\underline{\mathbf{C}}}, \underline{\beta}, \underline{\Gamma}) = \text{argmin} \int_{t \in \mathcal{T}, \, \mu \in \mathcal{P}} \underline{\mathbf{R}}(\underline{\bar{\mathbf{U}}}(t,\mu))^T \underline{\mathbf{R}}(\underline{\bar{\mathbf{U}}}(t,\mu)) \, dt \, d\mu$$

One word about the Proper Generalized Decomposition

- If one does not want to calculate the exact solution, or cannot obtain it (intractable)

$$\boxed{\begin{aligned}\underline{\bar{\mathbf{U}}}:\quad &\mathcal{U}_{\text{sep}} = \mathbb{R}^n \times \mathcal{T} \times \mathcal{P} \rightarrow \mathbb{R}^n \\ &\underline{\bar{\mathbf{U}}}(t,\mu) = \sum_{i=1}^{n_C} \underline{\mathbf{C}}_i\, \beta_i(t)\gamma_i(\mu)\,,\end{aligned}}$$

- Least-square PGD

$$J(\underline{\underline{\mathbf{C}}}, \underline{\beta}, \underline{\Gamma}) = \operatorname{argmin} \|\underline{\mathbf{R}}\|^2_{\mathcal{V}}$$

$$J(\underline{\underline{\mathbf{C}}}, \underline{\beta}, \underline{\Gamma}) = \operatorname{argmin} \int_{t\in\mathcal{T},\, \mu\in\mathcal{P}} \underline{\mathbf{R}}(\underline{\bar{\mathbf{U}}}(t,\mu))^T \underline{\mathbf{R}}(\underline{\bar{\mathbf{U}}}(t,\mu))\, dt\, d\mu$$

- Galerkin PGD: $\displaystyle\int_{t\in\mathcal{T},\, \mu\in\mathcal{P}} \delta\underline{\bar{\mathbf{U}}}^T \underline{\mathbf{R}}(\underline{\bar{\mathbf{U}}}(t,\mu))\, dt\, d\mu = 0 \qquad \forall\, \delta\underline{\bar{\mathbf{U}}} \in \mathcal{V}$

➡ Does not lead to a classical eigen value problem, needs to be associated with an algorithm to find a quasi-optimum (progressive PGD) [Ladeveze, Nouy]

➡ Fixed-point algorithms permits to obtain an iterative solution by solving separate problems in each dimension [Chinesta, Ammar]

Using the separation of variables in practice

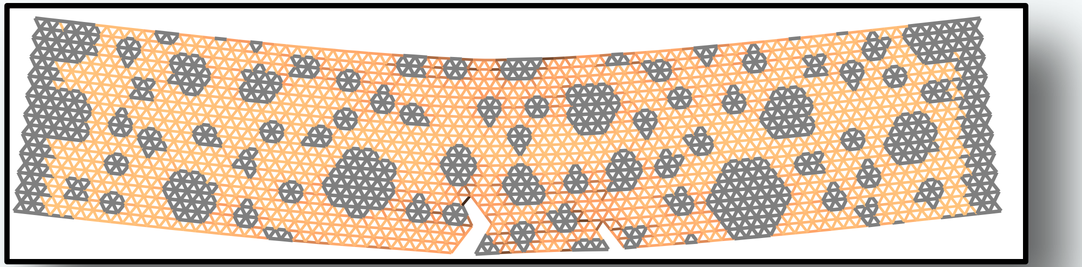
$$\boxed{\begin{aligned} \underline{\bar{\mathbf{U}}}: \quad & \mathcal{U}_{\text{sep}} = \mathbb{R}^n \times \mathcal{T} \times \mathcal{P} \to \mathbb{R}^n \\ & \underline{\bar{\mathbf{U}}}(t,\mu) = \sum_{i=1}^{n_C} \underline{\mathbf{C}}_i \, \beta_i(t)\gamma_i(\mu)\,, \end{aligned}}$$

- From the knowledge of such a decomposition one can derive:
  - Fast evaluation of solution by just “plugging in” the values of the parameters (data compression)
  - Initialisers / preconditioners of solvers for particular realisations of the parameters
    - ➡ The decomposition does not need to be “converged” or very accurate
    - ➡ Does not need to be complete (valid for all range of parameters). An availability for discrete values of the parameters can be enough (Snapshots). Interpolation can be performed (“gappy reconstructions”)
  - Model Order Reduction (closely linked to the previous bullet point)
    - ➡ Keep only one part of the decomposition an compute the remaining one for the desired realisation of the parameters (example: generate an optimal space basis in the snapshot space and use it in the framework of classical MOR)
    - ➡ Again, no need to have a very accurate / complete decomposition. The reduced model will “fill the gaps”

Model Order Reduction using the separation of variables

$$\underline{\bar{\mathbf{U}}} : \quad \mathcal{U}_{\text{sep}} = \mathbb{R}^n \times \mathcal{T} \times \mathcal{P} \rightarrow \mathbb{R}^n$$

$$\underline{\bar{\mathbf{U}}}(t,\mu) = \sum_{i=1}^{n_C} \underline{\mathbf{C}}_i \, \beta_i(t) \gamma_i(\mu) \, ,$$



$$\underline{\underline{\mathbf{C}}}^T \left(\underline{\mathbf{F}}_{\text{Ext}} + \underline{\mathbf{F}}_{\text{Int}} (\underline{\mathbf{U}}_{|t_n} + \underline{\underline{\mathbf{C}}}\, \underline{\boldsymbol{\alpha}}) \right) = \underline{\mathbf{0}}$$

Optimal state variable for a particular realisation,
compensate the inaccuracy in the global decomposition

Model order reduction using the separation of variables

$$\boxed{\begin{aligned} \underline{\bar{\mathbf{U}}}: \quad \mathcal{U}_{\text{sep}} &= \mathbb{R}^n \times \mathcal{T} \times \mathcal{P} \to \mathbb{R}^n \\ \underline{\bar{\mathbf{U}}}(t,\mu) &= \sum_{i=1}^{n_C} \underline{\mathbf{C}}_i \, \beta_i(t)\gamma_i(\mu) \, , \end{aligned}}$$

- One does not have the exact solution for all parameters / time steps, only for some ***snapshots.***

- Snapshot POD (Sirovich '87)

➡ generate an optimal space basis in the snapshot space and use it in the framework of classical MOR

➡ Same thing could be done with MLPOD or PGD

Snapshot Proper orthogonal decomposition

- Modified functional

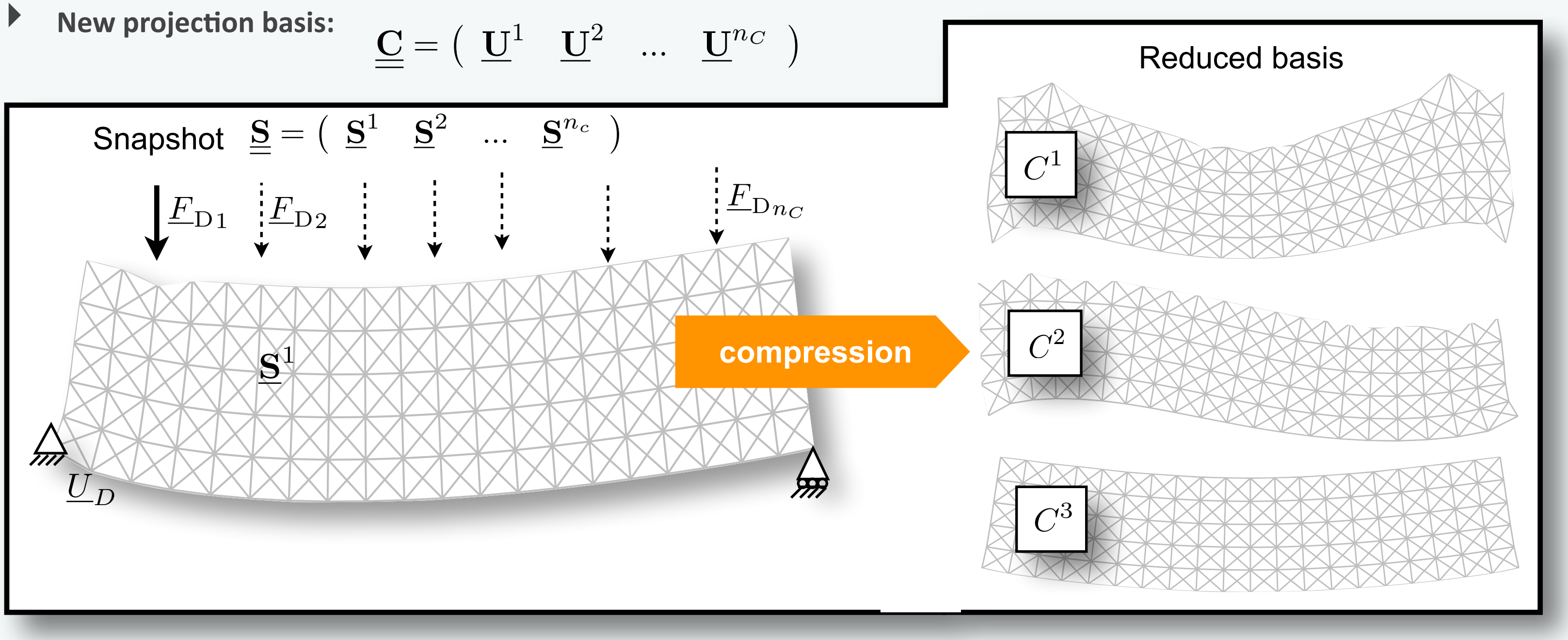
  - find an orthonormal family of vectors $(\underline{\mathbf{C}}^k)_{k\in[\![1,n_c]\!]}$ such that $n_c < n_s$

    and $\widehat{J}(\underline{\mathbf{C}}^1,\dots,\underline{\mathbf{C}}^{n_c}) = \sum_{j=1}^{n_s} \left\| \underline{\mathbf{S}}^j - \sum_{i=1}^{n_c} \left(\underline{\mathbf{C}}^{i^T} \underline{\mathbf{S}}^j \right) \underline{\mathbf{C}}^i \right\|_2^2$ is minimum

  with snapshots $\underline{\mathbf{S}}^j = \underline{\mathbf{U}}(t^j,\, \mu^j)$

  - SVD: $\underline{\underline{\mathbf{S}}} = \underline{\underline{\mathbf{U}}}\,\underline{\underline{\mathbf{\Sigma}}}\,\underline{\underline{\mathbf{V}}}^T = \sum_{k=1}^{n_S} \Sigma^k\, \underline{\mathbf{U}}^k\, \underline{\mathbf{V}}^{k^T}$ where $(\Sigma^k)_{k\in[\![1\ \ n_S]\!]}$ in decreasing order

  - **New projection basis:** $\underline{\underline{\mathbf{C}}} = \left(\underline{\mathbf{U}}^1 \quad \underline{\mathbf{U}}^2 \quad \dots \quad \underline{\mathbf{U}}^{n_C} \right)$


Reduced basis
Snapshot $\underline{\underline{\mathbf{S}}} = (\; \underline{\mathbf{S}}^1 \quad \underline{\mathbf{S}}^2 \quad \dots \quad \underline{\mathbf{S}}^{n_c} \;)$
$\underline{F}_{\mathrm{D}1}$
$\underline{F}_{\mathrm{D}2}$
$\underline{F}_{\mathrm{D}n_C}$
$\underline{\mathbf{S}}^1$
$\underline{U}_D$
compression
C^1
C^2
C^3

Snapshot Proper orthogonal decomposition

- Modified functional

> find an orthonormal family of vectors $(\underline{C}^k)_{k\in[\![1,n_c]\!]}$ such that $n_c < n_s$
>
> and $\widehat{J}(\underline{C}^1,\dots,\underline{C}^{n_c}) = \sum_{j=1}^{n_s} \left\| \underline{S}^j - \sum_{i=1}^{n_c} \left(\underline{C}^{i^T} \underline{S}^j\right) \underline{C}^i \right\|_2^2$ is minimum

with snapshots $\underline{S}^j = \underline{U}(t^j,\, \mu^j)$

- SVD: $\underline{\underline{S}} = \underline{\underline{U}}\,\underline{\underline{\Sigma}}\,\underline{\underline{V}}^T = \sum_{k=1}^{n_S} \Sigma^k\, \underline{U}^k\, \underline{V}^{k^T}$ where $(\Sigma^k)_{k\in[\![1\ n_S]\!]}$ in decreasing order

- **New projection basis:** $\underline{\underline{C}} = \left(\underline{U}^1 \quad \underline{U}^2 \quad \dots \quad \underline{U}^{n_C} \right)$

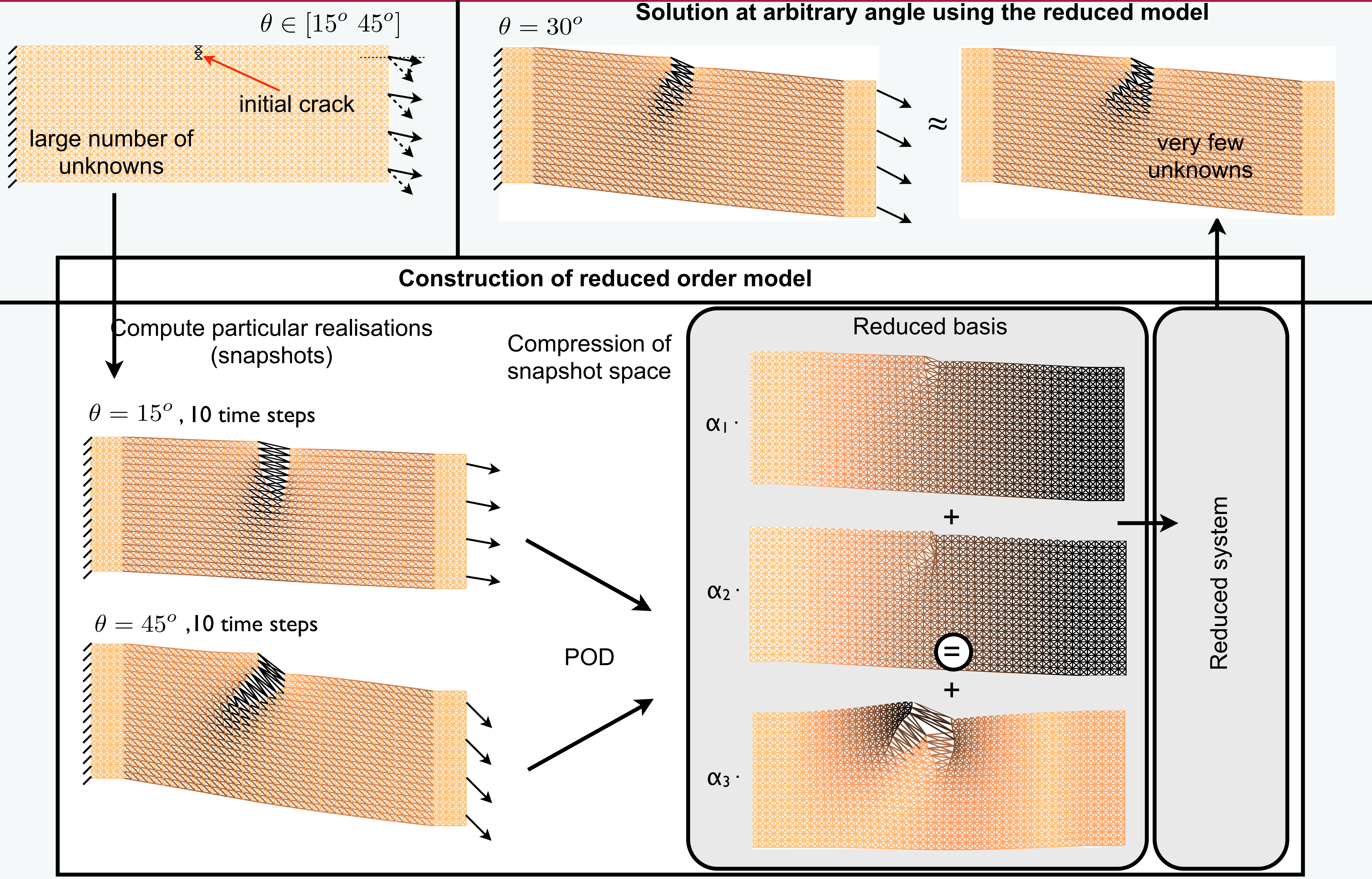
- Error in snapshot: $\widehat{J}(\underline{C}^1,\dots,\underline{C}^{n_c}) = \sum_{i=n_c+1}^{n_s} \sqrt{\Sigma^k} = \sum_{i=n_c+1}^{n_s} \lambda^k$

$$\nu_{\mathrm{SVD}} = \frac{\left(\sum_{j=1}^{n_s} \left\| \underline{S}^j - \sum_{i=1}^{n_c} \left(\underline{C}^{i^T} \underline{S}^j\right) \underline{C}^i \right\|^2\right)^{\frac{1}{2}}}{\left(\sum_{j=1}^{n_s} \left\|\underline{S}^j\right\|^2\right)^{\frac{1}{2}}} = \frac{\left(\sum_{i=n_c+1}^{n_s} \lambda_i\right)^{\frac{1}{2}}}{\left(\sum_{i=1}^{n_s} \lambda_i\right)^{\frac{1}{2}}}$$

thin SVD much cheaper than the eigenvalue problem of the POD (original idea of Sirovich)

Model Order Reduction using Snapshot POD

Parametrised fracture model



$\theta \in [15^o \; 45^o]$

initial crack

large number of unknowns

$\theta = 30^o$

Solution at arbitrary angle using the reduced model

$\approx$

very few unknowns

Construction of reduced order model

Compute particular realisations (snapshots)

Compression of snapshot space

$\theta = 15^o$, 10 time steps

$\theta = 45^o$,10 time steps

POD

Reduced basis

$\alpha_1 \cdot$

+

$\alpha_2 \cdot$

=

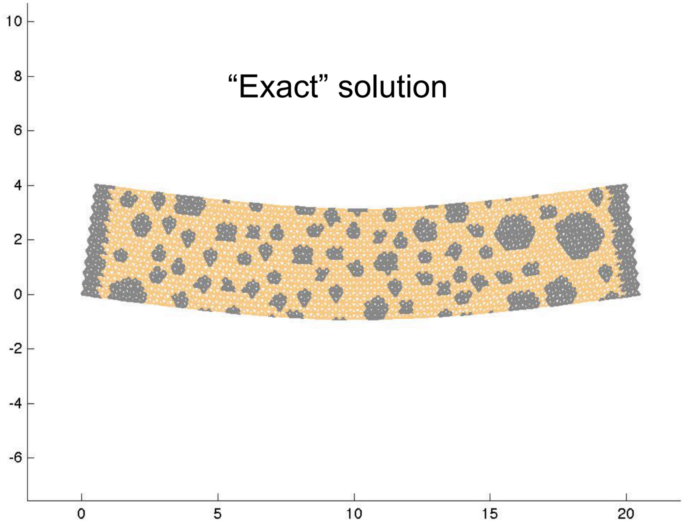
+

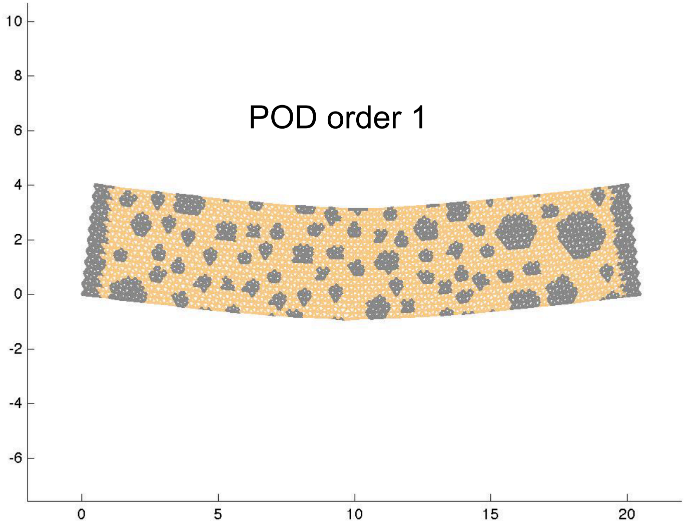
$\alpha_3 \cdot$

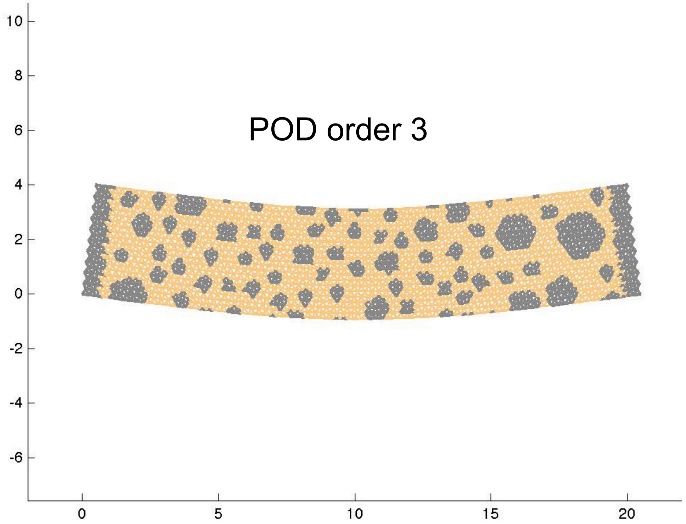
Reduced system

Data compression: fracture

Snapshot POD (snapshot space is spanned by the ensemble of solutions at all time steps)

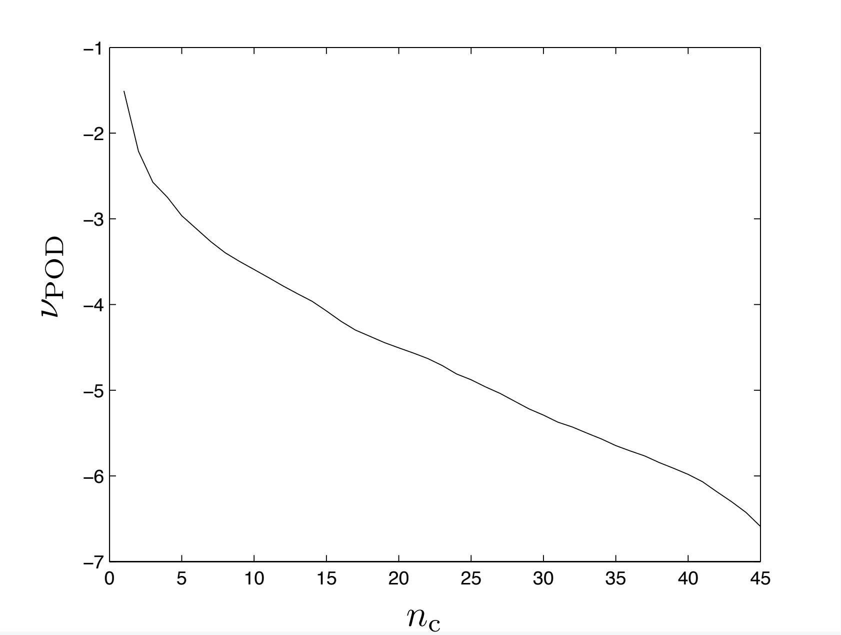






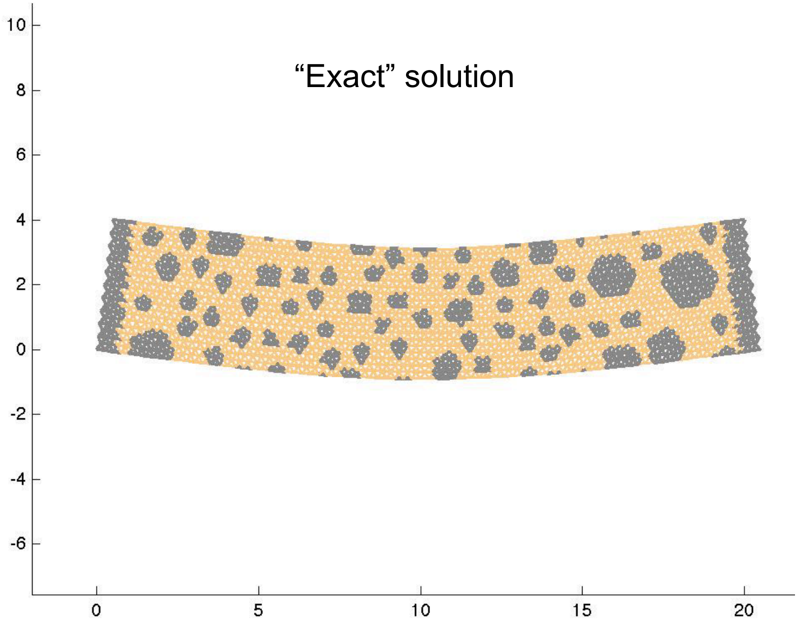
Data compression: fracture

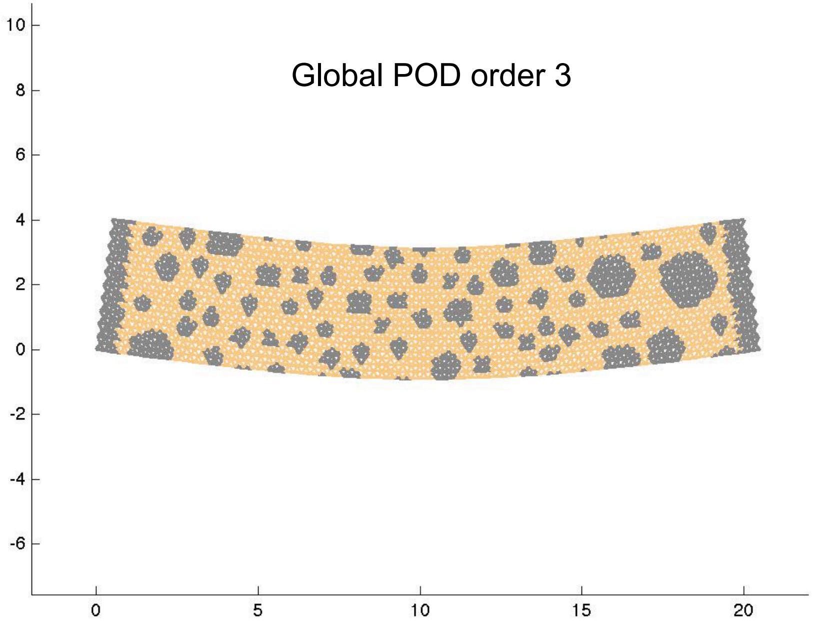
$$\nu_{\mathrm{POD}} = \frac{\left(\sum_{j=1}^{n_s}\left\|\underline{\mathbf{S}}^j - \sum_{i=1}^{n_c}\left(\underline{\mathbf{C}}^{i^T}\underline{\mathbf{S}}^j\right)\underline{\mathbf{C}}^i\right\|^2\right)^{\frac{1}{2}}}{\left(\sum_{j=1}^{n_s}\left\|\underline{\mathbf{S}}^j\right\|^2\right)^{\frac{1}{2}}} = \frac{\left(\sum_{i=n_c+1}^{n_s}\lambda_i\right)^{\frac{1}{2}}}{\left(\sum_{i=1}^{n_s}\lambda_i\right)^{\frac{1}{2}}}$$



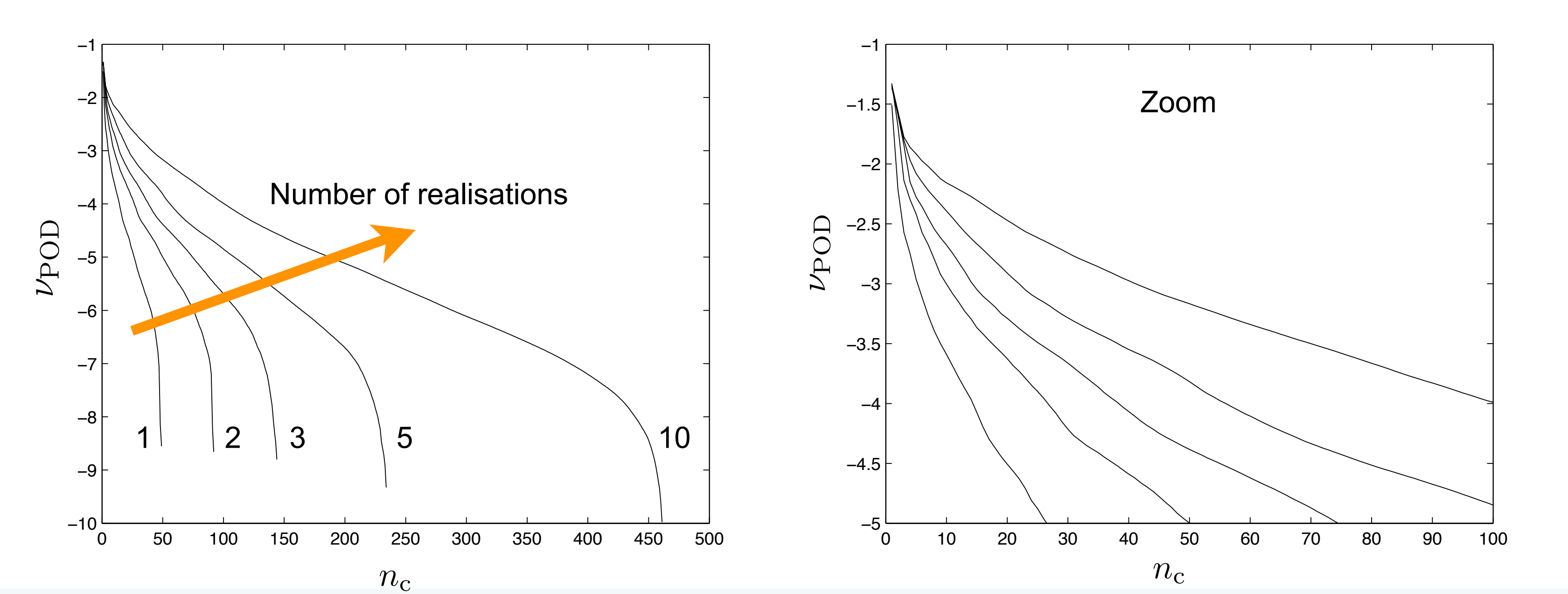
Data compression: parametric (random) fracture

Global Snapshot POD (snapshot space is spanned by the ensemble of solutions at all time steps for successive realisations)





Data compression: parametric (random) fracture

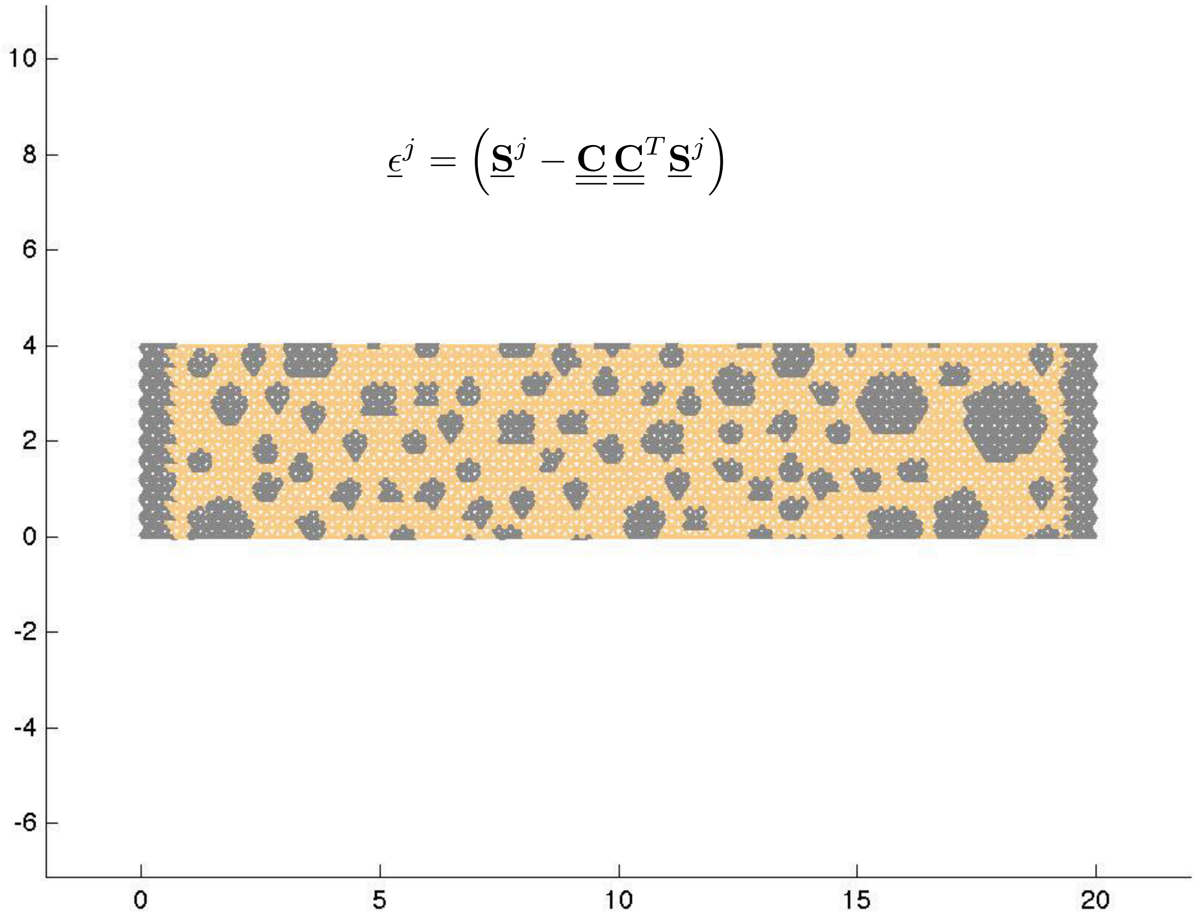


- Number of modes required to achieve a certain error increases linearly with the number of realisations.
  - Uncorrelated!

Can we do something about it?

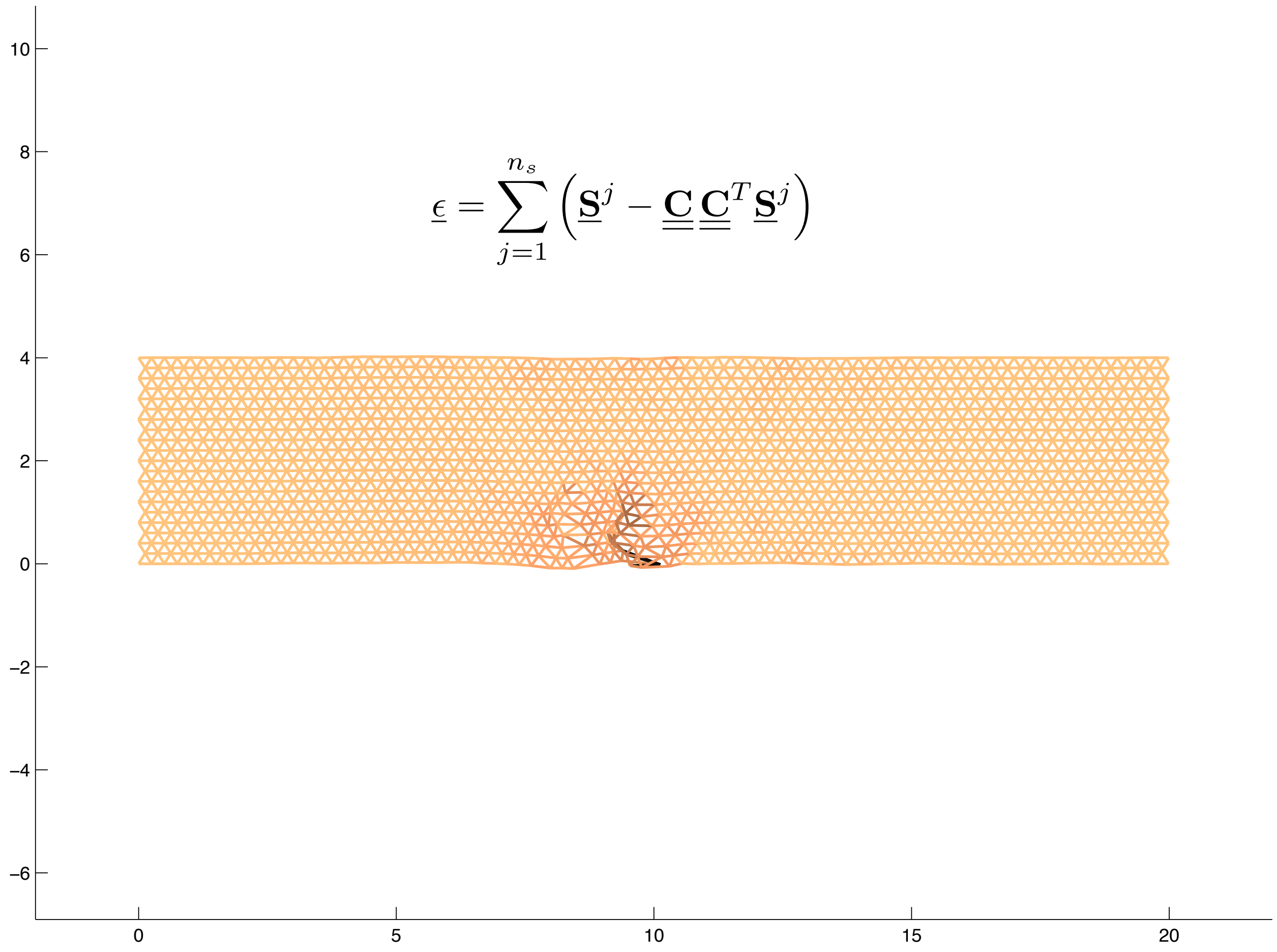
Error in one particular realisation

- Yes (ouf...), the error is local!
  - Error in the First of ten realisations used to build the reduced model


$$\underline{\epsilon}^j = \left(\underline{\mathbf{S}}^j - \underline{\underline{\mathbf{C}}}\,\underline{\underline{\mathbf{C}}}^T\underline{\mathbf{S}}^j\right)$$

Global error

- 10 realisations



$$\underline{\epsilon} = \sum_{j=1}^{n_s} \left(\underline{\mathbf{S}}^j - \underline{\underline{\mathbf{C}}}\,\underline{\underline{\mathbf{C}}}^T \underline{\mathbf{S}}^j \right)$$

Master/slave reduced order modelling

- Solution increment decomposed into two contributions

$$\underline{\mathbf{\Delta U}} = \underline{\mathbf{\Delta U}}^{(r)} + \underline{\mathbf{\Delta U}}^{(f)} \qquad \text{where} \qquad \begin{cases} \underline{\mathbf{\Delta U}}^{(r)} = \underline{\underline{\mathbf{E}}}^{(r)^T} \widetilde{\underline{\mathbf{\Delta U}}}^{(r)} \\ \underline{\mathbf{\Delta U}}^{(f)} = \underline{\underline{\mathbf{E}}}^{(f)^T} \widetilde{\underline{\mathbf{\Delta U}}}^{(f)} \end{cases}$$

extractor

“fully resolved” (master): cannot or should not be approximated in a Ritz basis

- Approximation of the slave part: $\widetilde{\underline{\mathbf{\Delta U}}}^{(r)} = \left(\underline{\underline{\mathbf{E}}}^{(r)} \underline{\underline{\mathbf{C}}}\right) \underline{\boldsymbol{\alpha}}$

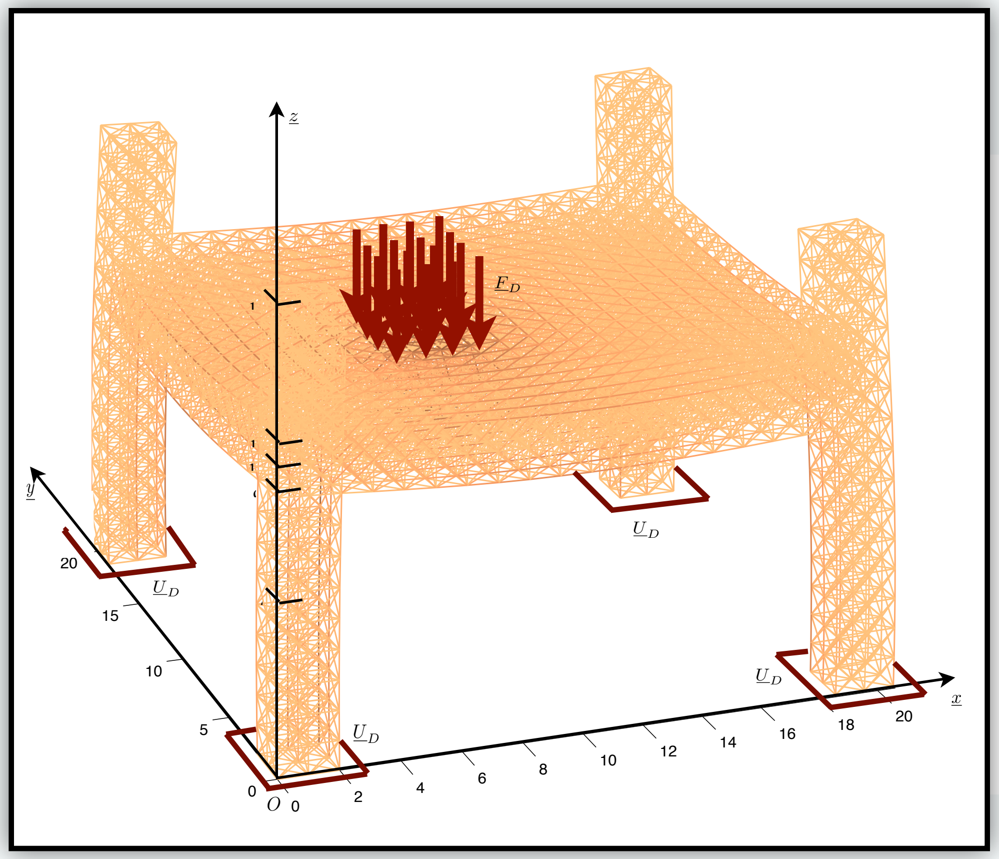
➡ $\underline{\mathbf{\Delta U}} = \underline{\underline{\mathbf{P}}}^{(r)} \underline{\underline{\mathbf{C}}}\, \underline{\boldsymbol{\alpha}} + \underline{\underline{\mathbf{E}}}^{(f)^T} \widetilde{\underline{\mathbf{\Delta U}}}^{(f)}$ where $\underline{\underline{\mathbf{P}}}^{(r)} = \underline{\underline{\mathbf{E}}}^{(r)^T} \underline{\underline{\mathbf{E}}}^{(r)}$

➡ Reduced state variables: $\underline{\mathbf{X}} = \begin{pmatrix} \underline{\boldsymbol{\alpha}} \\ \widetilde{\underline{\mathbf{\Delta U}}}^{(f)} \end{pmatrix}$

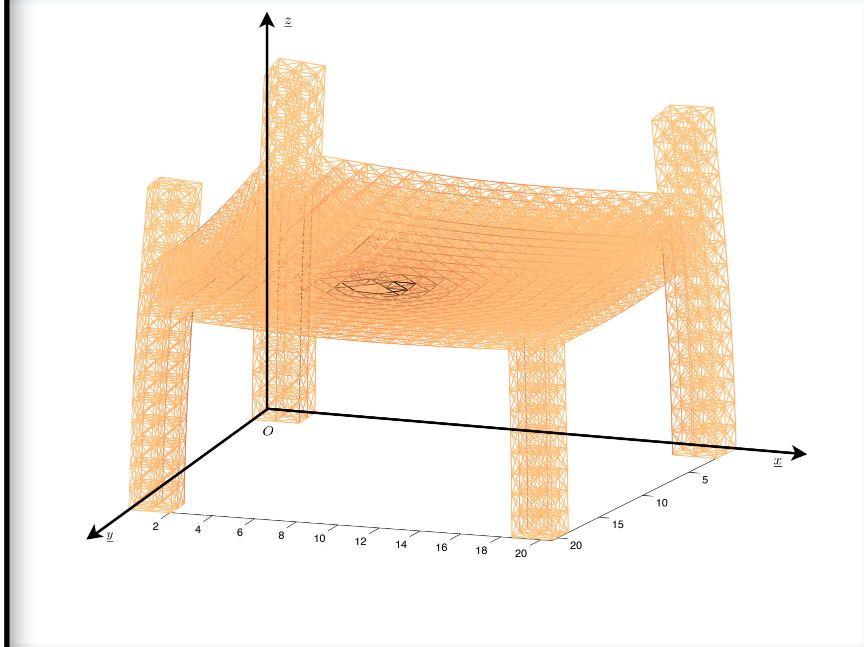
- Galerkin orthogonality

$$\boxed{\underline{\underline{\mathbf{A}}}^T \left(\underline{\mathbf{F}}_{\mathrm{Int}} \left(\underline{\mathbf{\Delta U}}(\underline{\mathbf{X}}) + \underline{\mathbf{U}}_{|t_n}\right) + \underline{\mathbf{F}}_{\mathrm{Ext}}\right) = \underline{\mathbf{0}}} \qquad \text{where} \qquad \underline{\underline{\mathbf{A}}} = \left(\underline{\underline{\mathbf{P}}}^{(r)} \underline{\underline{\mathbf{C}}} \quad \underline{\underline{\mathbf{E}}}^{(f)^T}\right)$$

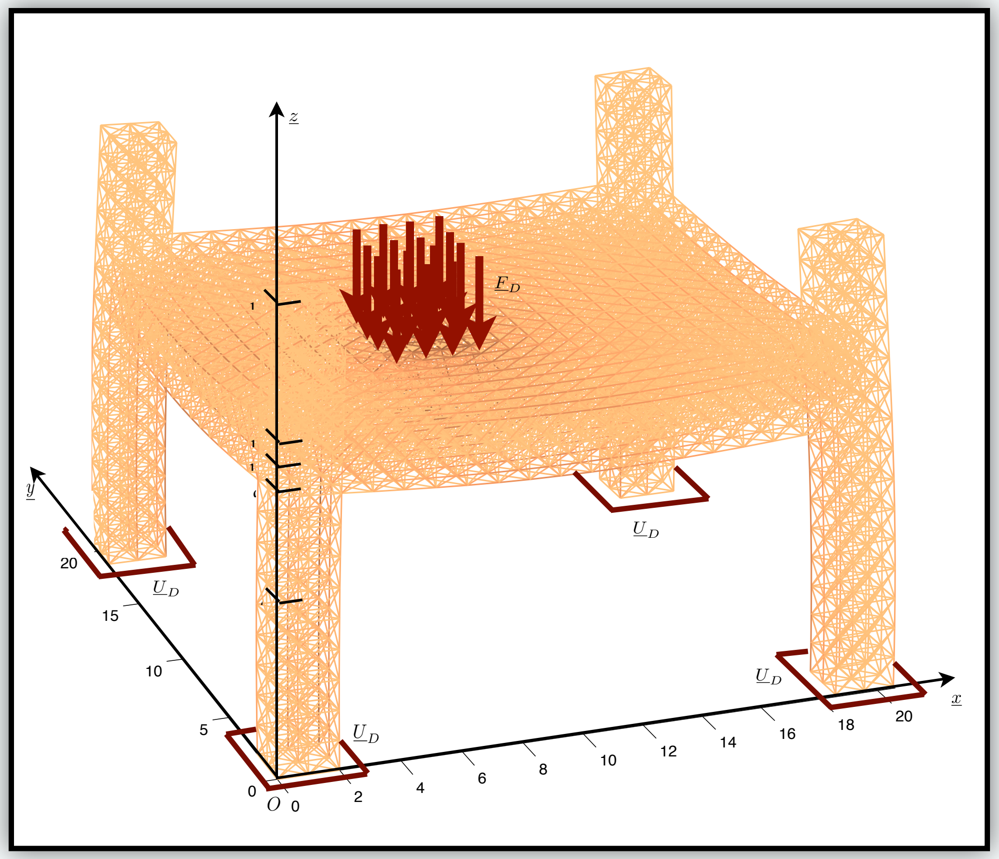
Example



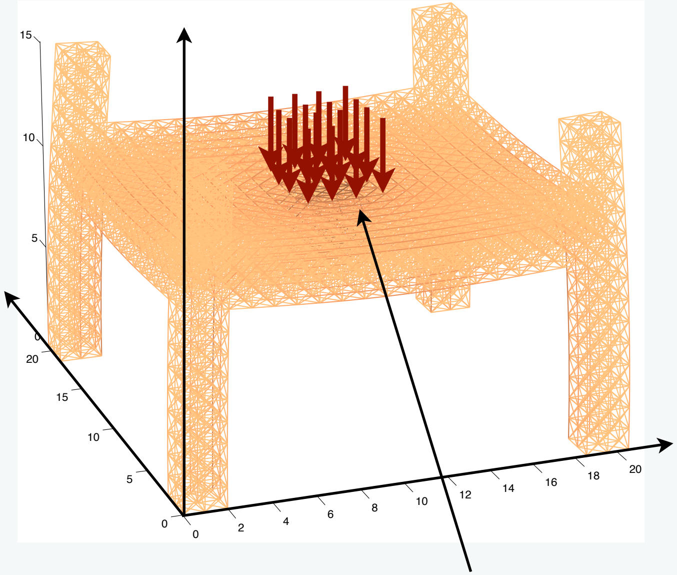
final damage steps
obtained in 30 time steps



Example: snapshot



Snapshot: solution of a nearby problem
(30 time steps)



Different location of the applied forces

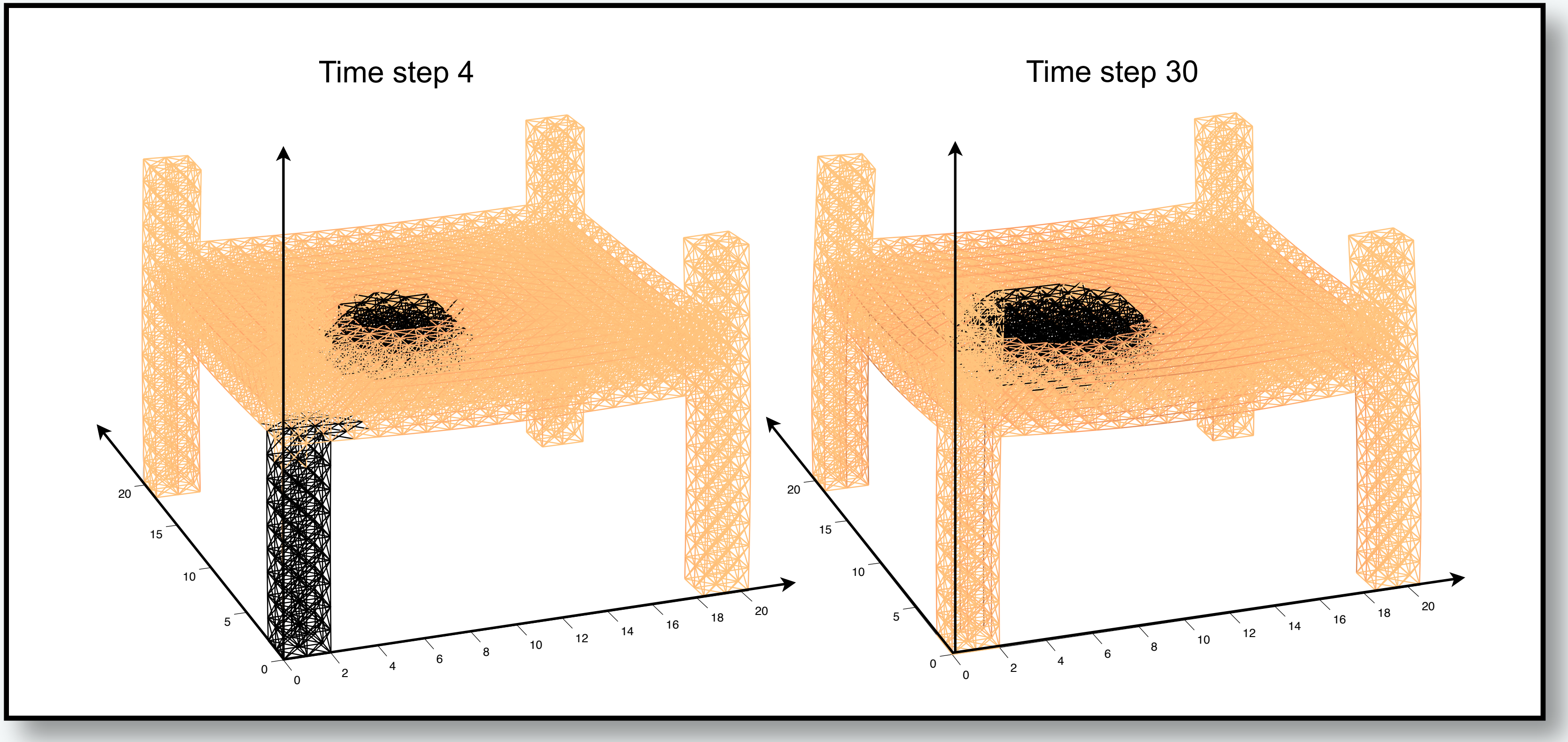
➡ extraction of a coarse approximation space by a SVD (3 basis vectors)

Local/global splitting

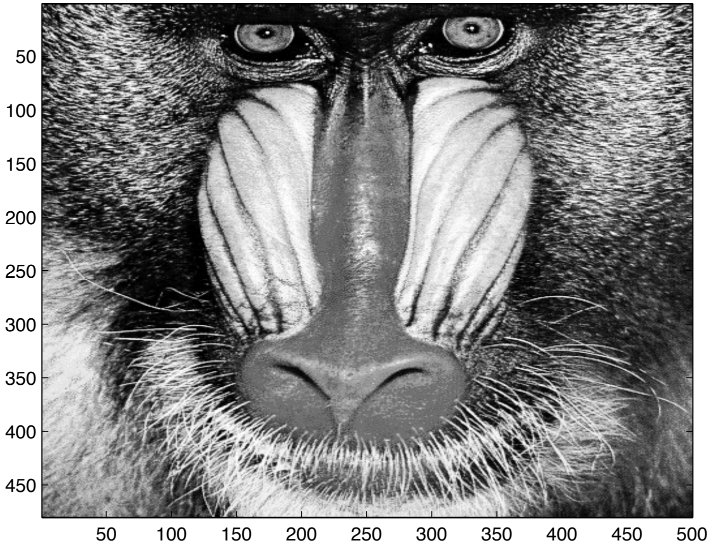
- master degrees of freedom corresponding to the nodes connected to elements undergoing significant damage increments

➡ requires some mechanical understanding of the behaviour of the model

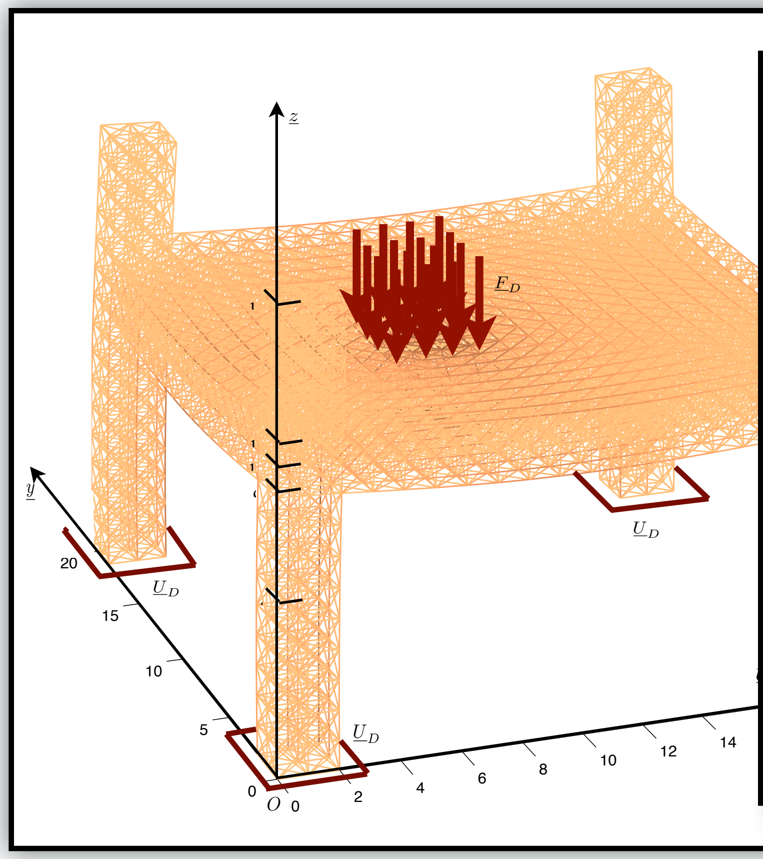
- Update at each time increment

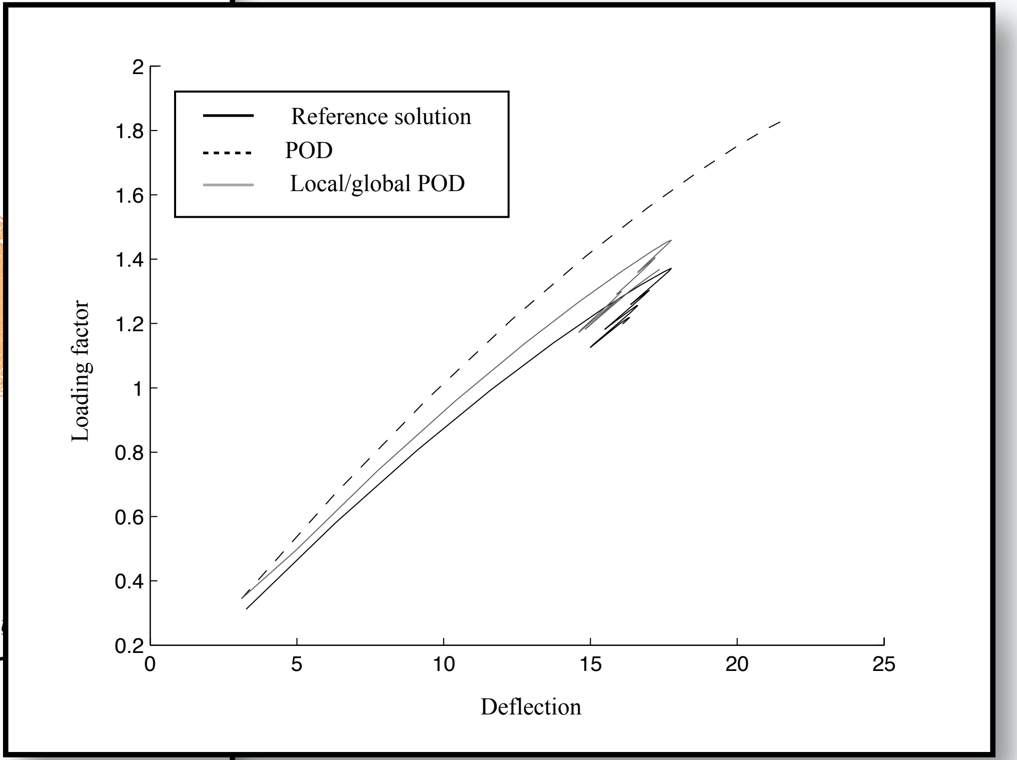


The monkey is back



Results: no global correction





“On-the-fly” enrichment via Krylov iterations

- **Idea:** if the norm of the residual of the full problem is too large, enrich the ROM before the next Newton iteration by a **coarse iterative solution of the full linearized problem**

- **Tool:** projected conjugate gradient [Dostal ‘88] on the linearized problem $\underline{\underline{\mathbf{K}}}_{\mathrm{T}}\,\underline{\boldsymbol{\delta}\mathbf{U}} = \underline{\mathbf{R}}$
  - splitting of the search space:

$$\boxed{\begin{array}{l} \mathbb{R}^{n_u} = \mathrm{Im}(\underline{\underline{\mathbf{C}}}) \oplus \mathrm{Im}(\underline{\underline{\mathbf{C}}})^{\perp_{\underline{\underline{\mathbf{K}}}_{\mathrm{T}}}} \\ \underline{\boldsymbol{\delta}\mathbf{U}} = \underline{\boldsymbol{\delta}\mathbf{U}}_{\mathrm{C}} + \underline{\boldsymbol{\delta}\mathbf{U}}_{\mathrm{K}} \quad \text{where} \quad \left\{ \begin{array}{r} \underline{\boldsymbol{\delta}\mathbf{U}}_{\mathrm{C}} = \underline{\underline{\mathbf{C}}}\,\underline{\boldsymbol{\delta}\boldsymbol{\alpha}} \in \mathrm{Im}(\underline{\underline{\mathbf{C}}}) \\ \underline{\boldsymbol{\delta}\mathbf{U}}_{\mathrm{K}} \in \mathrm{Im}(\underline{\underline{\mathbf{C}}})^{\perp_{\underline{\underline{\mathbf{K}}}_{\mathrm{T}}}} = \mathrm{Ker}(\underline{\underline{\mathbf{C}}}^T\underline{\underline{\mathbf{K}}}_{\mathrm{T}}) \end{array} \right. \end{array}}$$

  - Orthogonality insured by a projector

“On-the-fly” enrichment via Krylov iterations

- **Idea:** if the norm of the residual of the full problem is too large, enrich the ROM before the next Newton iteration by a **coarse iterative solution of the full linearized problem**

- **Tool:** projected conjugate gradient [Dostal ‘88] on the linearized problem $\underline{\underline{\mathbf{K}}}_{\mathrm{T}}\,\underline{\boldsymbol{\delta}\mathbf{U}} = \underline{\mathbf{R}}$

  - splitting of the search space:

$$\boxed{\begin{array}{l} \mathbb{R}^{n_u} = \mathrm{Im}(\underline{\underline{\mathbf{C}}}) \oplus \mathrm{Im}(\underline{\underline{\mathbf{C}}})^{\perp_{\underline{\underline{\mathbf{K}}}_{\mathrm{T}}}} \\ \underline{\boldsymbol{\delta}\mathbf{U}} = \underline{\boldsymbol{\delta}\mathbf{U}}_{\mathrm{C}} + \underline{\boldsymbol{\delta}\mathbf{U}}_{\mathrm{K}} \quad \text{where} \quad \left\{ \begin{array}{r} \underline{\boldsymbol{\delta}\mathbf{U}}_{\mathrm{C}} = \underline{\underline{\mathbf{C}}}\,\underline{\boldsymbol{\delta\alpha}} \in \mathrm{Im}(\underline{\underline{\mathbf{C}}}) \\ \underline{\boldsymbol{\delta}\mathbf{U}}_{\mathrm{K}} \in \mathrm{Im}(\underline{\underline{\mathbf{C}}})^{\perp_{\underline{\underline{\mathbf{K}}}_{\mathrm{T}}}} = \mathrm{Ker}(\underline{\underline{\mathbf{C}}}^T\underline{\underline{\mathbf{K}}}_{\mathrm{T}}) \end{array} \right. \end{array}}$$

  - Orthogonality insured by a projector $\underline{\boldsymbol{\delta}\mathbf{U}} = \underline{\underline{\mathbf{C}}}\,\underline{\boldsymbol{\delta\alpha}} + \underline{\underline{\mathbf{P}}}\,\underline{\boldsymbol{\delta}\mathbf{U}}_{\mathrm{K}}$ where $\underline{\underline{\mathbf{P}}} = \underline{\underline{\mathbf{I}}}_{\mathrm{d}} - \underline{\underline{\mathbf{C}}}(\underline{\underline{\mathbf{C}}}^T\underline{\underline{\mathbf{K}}}_{\mathrm{T}}\underline{\underline{\mathbf{C}}})^{-1}\underline{\underline{\mathbf{C}}}^T\underline{\underline{\mathbf{K}}}_{\mathrm{T}}$

  - Yields two uncoupled systems:

- coarse scale, solved by a direct solver:

  ➡ linear predictions performed on the reduced problem (i.e.: when solved by the basic snapshot-POD)

$$(\underline{\underline{\mathbf{C}}}^T\,\underline{\underline{\mathbf{K}}}_{\mathrm{T}}\,\underline{\underline{\mathbf{C}}})\,\underline{\boldsymbol{\delta\alpha}} = \underline{\underline{\mathbf{C}}}^T\,\underline{\mathbf{R}}$$

$$\Rightarrow \underline{\boldsymbol{\delta}\mathbf{U}}_{\mathrm{C}} = \underline{\underline{\mathbf{C}}}(\underline{\underline{\mathbf{C}}}^T\,\underline{\underline{\mathbf{K}}}_{\mathrm{T}}\,\underline{\underline{\mathbf{C}}})^{-1}\underline{\underline{\mathbf{C}}}^T\,\underline{\mathbf{R}}$$

- fine scale, solved approximately by the conjugate gradient:

  ➡ correction used to enrich the Ritz basis (K$_{\mathrm{T}}$-orthogonal to the previous Ritz basis by construction)

$$\left(\underline{\underline{\mathbf{P}}}^T\,\underline{\underline{\mathbf{K}}}_{\mathrm{T}}\,\underline{\underline{\mathbf{P}}}\right)\underline{\boldsymbol{\delta}\mathbf{U}}_{K} = \underline{\underline{\mathbf{P}}}^T\,\underline{\mathbf{R}}$$

“On-the-fly” enrichment via Krylov iterations

- **Idea:** if the norm of the residual of the full problem is too large, enrich the ROM before the next Newton iteration by a **coarse iterative solution of the full linearized problem**

- **Tool:** projected conjugate gradient [Dostal ‘88] on the linearized problem $\underline{\underline{\mathbf{K}}}_{\mathrm{T}}\,\underline{\boldsymbol{\delta}\mathbf{U}} = \underline{\mathbf{R}}$
  - splitting of the search space:

$$\boxed{\begin{array}{l} \mathbb{R}^{n_u} = \mathrm{Im}(\underline{\underline{\mathbf{C}}}) \oplus \mathrm{Im}(\underline{\underline{\mathbf{C}}})^{\perp_{\underline{\underline{\mathbf{K}}}_{\mathrm{T}}}} \\ \underline{\boldsymbol{\delta}\mathbf{U}} = \underline{\boldsymbol{\delta}\mathbf{U}}_{\mathrm{C}} + \underline{\boldsymbol{\delta}\mathbf{U}}_{\mathrm{K}} \quad \text{where} \quad \left\{ \begin{array}{r} \underline{\boldsymbol{\delta}\mathbf{U}}_{\mathrm{C}} = \underline{\underline{\mathbf{C}}}\,\underline{\boldsymbol{\delta}\boldsymbol{\alpha}} \in \mathrm{Im}(\underline{\underline{\mathbf{C}}}) \\ \underline{\boldsymbol{\delta}\mathbf{U}}_{\mathrm{K}} \in \mathrm{Im}(\underline{\underline{\mathbf{C}}})^{\perp_{\underline{\underline{\mathbf{K}}}_{\mathrm{T}}}} = \mathrm{Ker}(\underline{\underline{\mathbf{C}}}^T\underline{\underline{\mathbf{K}}}_{\mathrm{T}}) \end{array} \right. \end{array}}$$

  - Orthogonality insured by a projector $\underline{\boldsymbol{\delta}\mathbf{U}} = \underline{\underline{\mathbf{C}}}\,\underline{\boldsymbol{\delta}\boldsymbol{\alpha}} + \underline{\underline{\mathbf{P}}}\,\underline{\boldsymbol{\delta}\mathbf{U}}_{\mathrm{K}}$ where $\underline{\underline{\mathbf{P}}} = \underline{\underline{\mathbf{I}}}_{\mathrm{d}} - \underline{\underline{\mathbf{C}}}(\underline{\underline{\mathbf{C}}}^T\underline{\underline{\mathbf{K}}}_{\mathrm{T}}\underline{\underline{\mathbf{C}}})^{-1}\underline{\underline{\mathbf{C}}}^T\underline{\underline{\mathbf{K}}}_{\mathrm{T}}$
  - Yields two uncoupled systems:

– coarse scale, solved by a direct solver:

$$(\underline{\underline{\mathbf{C}}}^T\,\underline{\underline{\mathbf{K}}}_{\mathrm{T}}\,\underline{\underline{\mathbf{C}}})\,\underline{\boldsymbol{\delta}\boldsymbol{\alpha}} = \underline{\underline{\mathbf{C}}}^T\,\underline{\mathbf{R}}$$

$$\Rightarrow\ \underline{\boldsymbol{\delta}\mathbf{U}}_{\mathrm{C}} = \underline{\underline{\mathbf{C}}}(\underline{\underline{\mathbf{C}}}^T\,\underline{\underline{\mathbf{K}}}_{\mathrm{T}}\,\underline{\underline{\mathbf{C}}})^{-1}\underline{\underline{\mathbf{C}}}^T\,\underline{\mathbf{R}}$$

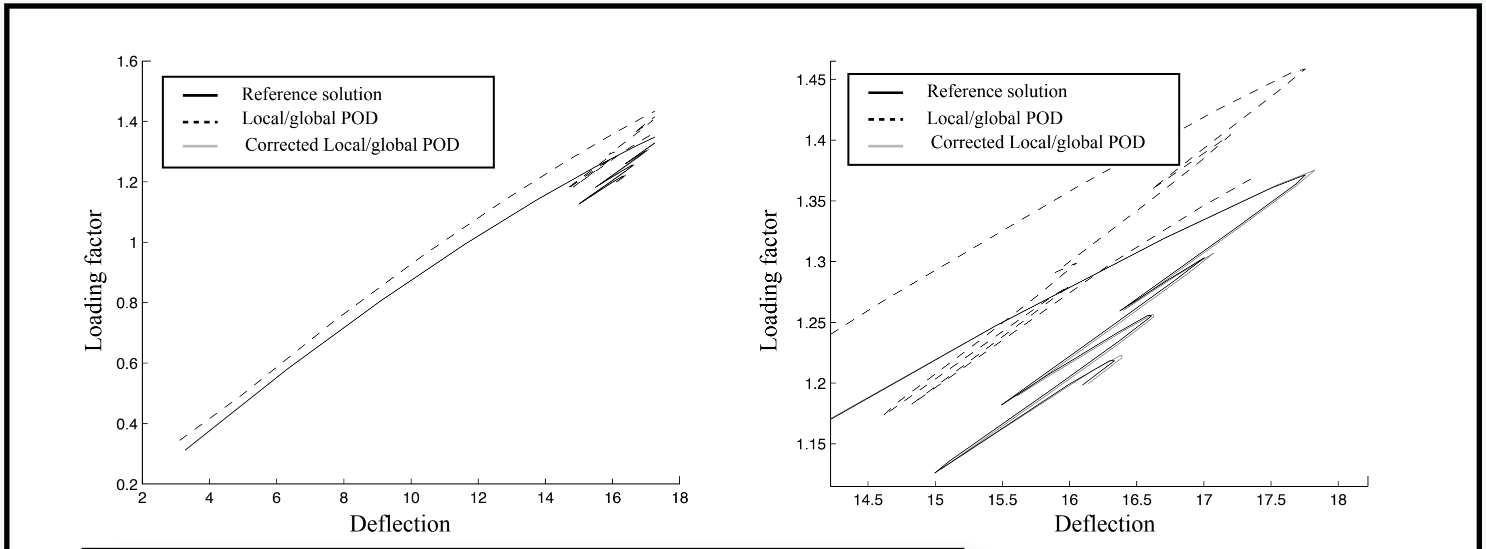
➡ linear predictions performed on the reduced problem
(i.e.: when solved by the basic snapshot-POD)

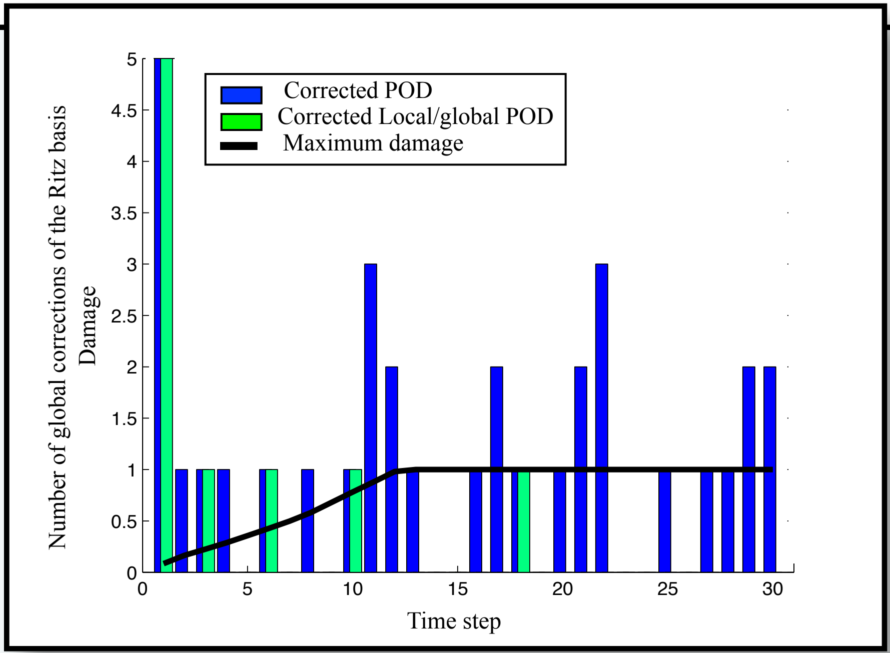
– fine scale, solved approximately by the conjugate gradient:

$$\left(\underline{\underline{\mathbf{P}}}^T\,\underline{\underline{\mathbf{K}}}_{\mathrm{T}}\,\underline{\underline{\mathbf{P}}}\right)\underline{\boldsymbol{\delta}\mathbf{U}}_K = \underline{\underline{\mathbf{P}}}^T\,\underline{\mathbf{R}}$$

➡ correction used to enrich the Ritz basis
(K$_{\mathrm{T}}$-orthogonal to the previous Ritz basis by construction)

Results: with global corrections





Solution algorithm

- Not a reduced order model anymore (ROM of only one part of the DOF)!

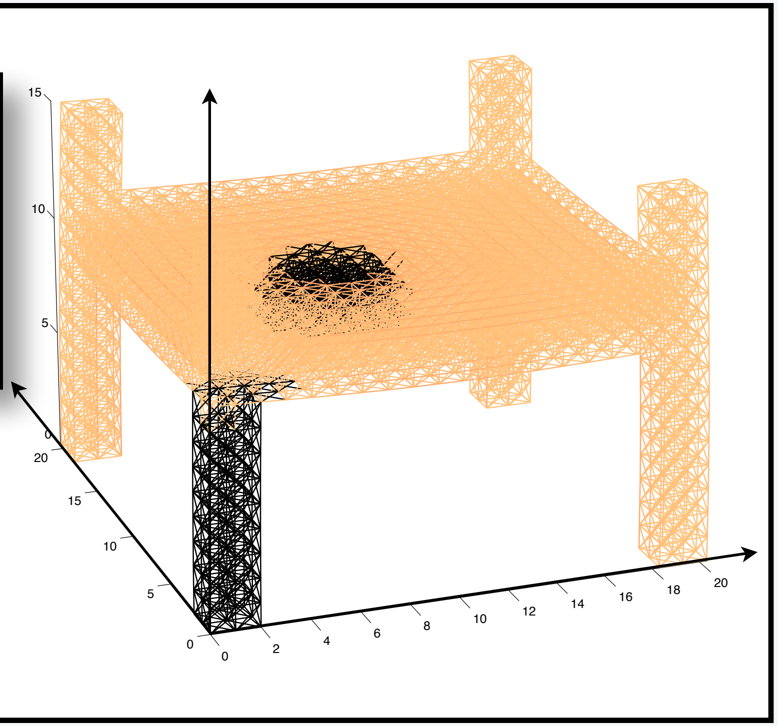
  ➡ solution algorithm might become inefficient (compare to basic POD) if not thought of carefully

➡ Idea: use the reduced order model to accelerate the solution algorithm.

- Successive linearisation in the Newton process:

$$\left.\frac{\partial \underline{\mathbf{R}}_{\mathrm{R}}(\underline{\mathbf{X}})}{\partial \underline{\mathbf{X}}}\right|_{\underline{\mathbf{X}}=\underline{\mathbf{X}}^i} \underline{\boldsymbol{\delta}\mathbf{X}}^{i+1} = -\underline{\mathbf{R}}_{\mathrm{R}}^i$$

$$\begin{pmatrix} \underline{\underline{\mathbf{K}}}_{\mathrm{T,R}}^{(rr),i} & \underline{\underline{\mathbf{K}}}_{\mathrm{T,R}}^{(rf),i} \\ \underline{\underline{\mathbf{K}}}_{\mathrm{T,R}}^{(fr),i} & \underline{\underline{\mathbf{K}}}_{\mathrm{T,R}}^{(ff),i} \end{pmatrix} \begin{pmatrix} \underline{\boldsymbol{\delta\alpha}}^i \\ \widetilde{\underline{\boldsymbol{\delta}\mathbf{U}}}^{(f),i} \end{pmatrix} = -\begin{pmatrix} \underline{\mathbf{R}}_{\mathrm{R}}^{(r),i} \\ \underline{\mathbf{R}}_{\mathrm{R}}^{(f),i} \end{pmatrix}$$



- Linear solver?
  - direct: the information from the snapshot is not used
  - **iterative with reduced order model as a preconditioner**

MOR-based preconditioner

- **Condensation** on the master degrees of freedom

$$\boxed{\underline{\underline{\mathbf{S}}}_{\mathrm{P}}^{(f)}\,\underline{\delta\mathbf{U}}^{(f)} = \underline{\mathbf{R}}_{\mathrm{C}}^{(f)}} \qquad \text{where} \qquad \begin{cases} \underline{\underline{\mathbf{S}}}_{\mathrm{P}}^{(f)} = \underline{\underline{\mathbf{K}}}_{\mathrm{T,R}}^{(ff)} - \underline{\underline{\mathbf{K}}}_{\mathrm{T,R}}^{(fr)} \left(\underline{\underline{\mathbf{K}}}_{\mathrm{T,R}}^{(rr)}\right)^{-1} \underline{\underline{\mathbf{K}}}_{\mathrm{T,R}}^{(rf)} \\ \underline{\mathbf{R}}_{\mathrm{C}}^{(f)} = \underline{\mathbf{R}}_{\mathrm{R}}^{(f)} - \underline{\underline{\mathbf{K}}}_{\mathrm{T,R}}^{(fr)} \left(\underline{\underline{\mathbf{K}}}_{\mathrm{T,R}}^{(rr)}\right)^{-1} \underline{\mathbf{R}}_{\mathrm{R}}^{(r)} \end{cases}$$

➡ Any solution satisfy the reduced balance equations of the “slave” degrees of freedom

$$\underline{\delta\mathbf{U}}^{(r)} = \underline{\underline{\mathbf{P}}}^{(r)}\underline{\underline{\mathbf{C}}}\,\underline{\delta\boldsymbol{\alpha}} \qquad \text{with} \qquad \underline{\delta\boldsymbol{\alpha}} = \left(\underline{\underline{\mathbf{K}}}_{\mathrm{T,R}}^{(rr)}\right)^{-1} \left(\underline{\mathbf{R}}_{\mathrm{R}}^{(r)} - \underline{\underline{\mathbf{K}}}_{\mathrm{T,R}}^{(rf)}\underline{\delta\mathbf{U}}^{(f)}\right)$$

MOR-based preconditioner

- **Condensation** on the master degrees of freedom

$$\boxed{\underline{\underline{\mathbf{S}}}_{\mathrm{P}}^{(f)}\,\underline{\boldsymbol{\delta}\mathbf{U}}^{(f)} = \underline{\mathbf{R}}_{\mathrm{C}}^{(f)}}\qquad \text{where}\qquad \begin{cases} \underline{\underline{\mathbf{S}}}_{\mathrm{P}}^{(f)} = \underline{\underline{\mathbf{K}}}_{\mathrm{T,R}}^{(ff)} - \underline{\underline{\mathbf{K}}}_{\mathrm{T,R}}^{(fr)}\left(\underline{\underline{\mathbf{K}}}_{\mathrm{T,R}}^{(rr)}\right)^{-1}\underline{\underline{\mathbf{K}}}_{\mathrm{T,R}}^{(rf)} \\ \underline{\mathbf{R}}_{\mathrm{C}}^{(f)} = \underline{\mathbf{R}}_{\mathrm{R}}^{(f)} - \underline{\underline{\mathbf{K}}}_{\mathrm{T,R}}^{(fr)}\left(\underline{\underline{\mathbf{K}}}_{\mathrm{T,R}}^{(rr)}\right)^{-1}\underline{\mathbf{R}}_{\mathrm{R}}^{(r)} \end{cases}$$

➡ Any solution satisfy the reduced balance equations of the “slave” degrees of freedom

$$\underline{\boldsymbol{\delta}\mathbf{U}}^{(r)} = \underline{\underline{\mathbf{P}}}^{(r)}\underline{\underline{\mathbf{C}}}\,\underline{\boldsymbol{\delta\alpha}} \qquad \text{with} \qquad \underline{\boldsymbol{\delta\alpha}} = \left(\underline{\underline{\mathbf{K}}}_{\mathrm{T,R}}^{(rr)}\right)^{-1}\left(\underline{\mathbf{R}}_{\mathrm{R}}^{(r)} - \underline{\underline{\mathbf{K}}}_{\mathrm{T,R}}^{(rf)}\underline{\boldsymbol{\delta}\mathbf{U}}^{(f)}\right)$$

- Projected conjugate gradient on the condensed problem

splitting of search space $\mathbb{R}^{n_f} = \mathrm{Im}(\underline{\underline{\mathbf{C}}}^{(f)}) \oplus \mathrm{Im}(\underline{\underline{\mathbf{C}}}^{(f)})^{\perp_{\underline{\underline{\mathbf{S}}}_{\mathrm{P}}^{(f)}}}$ where $\underline{\underline{\mathbf{C}}}^{(f)} = \underline{\underline{\mathbf{E}}}^{(f)}\underline{\underline{\mathbf{C}}}$

$$\underline{\boldsymbol{\delta}\mathbf{U}}^{(f)} = \underline{\boldsymbol{\delta}\mathbf{U}}_{\mathrm{C}}^{(f)} + \underline{\boldsymbol{\delta}\mathbf{U}}_{\mathrm{K}}^{(f)} \qquad \text{where} \qquad \begin{cases} \underline{\boldsymbol{\delta}\mathbf{U}}_{\mathrm{C}}^{(f)} = \underline{\underline{\mathbf{C}}}^{(f)}\underline{\boldsymbol{\delta\alpha}}^{(f)} \in \mathrm{Im}\left(\underline{\underline{\mathbf{C}}}^{(f)}\right) \\ \underline{\boldsymbol{\delta}\mathbf{U}}_{\mathrm{K}}^{(f)} \in \mathrm{Im}(\underline{\underline{\mathbf{C}}}^{(f)})^{\perp_{\underline{\underline{\mathbf{S}}}_{\mathrm{P}}^{(f)}}} = \mathrm{Ker}\left({\underline{\underline{\mathbf{C}}}^{(f)}}^{T}\underline{\underline{\mathbf{S}}}_{\mathrm{P}}^{(f)}\right) \end{cases}$$

- Projector $\underline{\underline{\mathbf{P}}} = \underline{\underline{\mathbf{I}}}_{\mathrm{d}} - \underline{\underline{\mathbf{C}}}^{(f)}({\underline{\underline{\mathbf{C}}}^{(f)}}^{T}\underline{\underline{\mathbf{K}}}_{\mathrm{T}}\underline{\underline{\mathbf{C}}}^{(f)})^{-1}{\underline{\underline{\mathbf{C}}}^{(f)}}^{T}\underline{\underline{\mathbf{K}}}_{\mathrm{T}}$

- Two uncoupled systems:

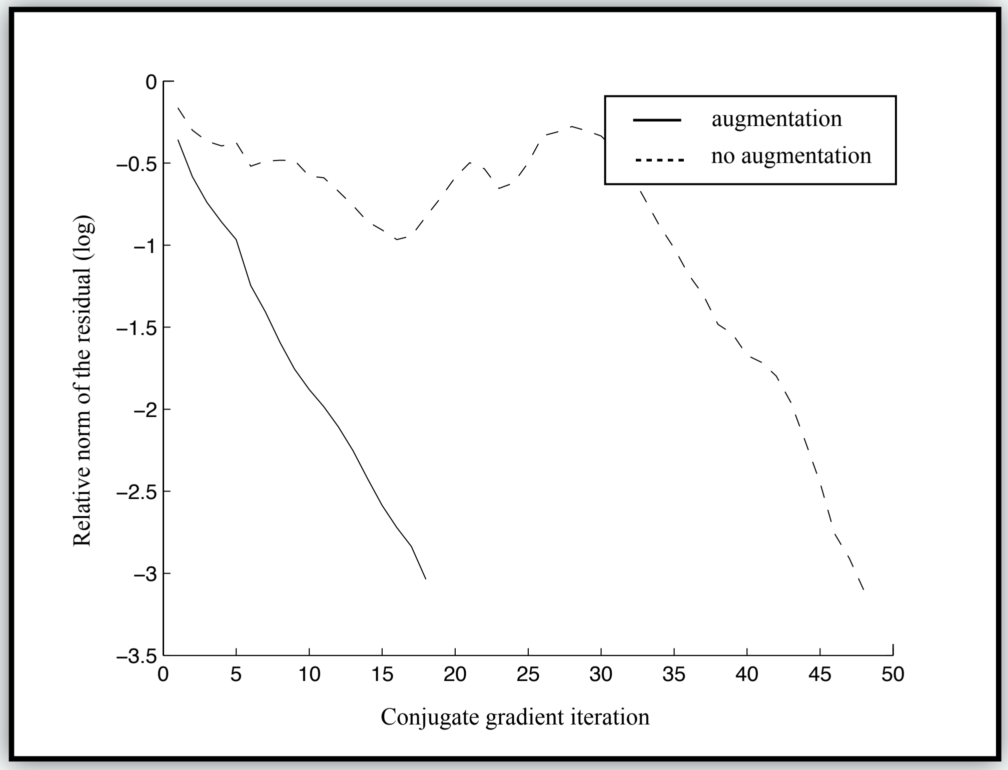
➡ Initiation by the reduced order model

$$\underline{\boldsymbol{\delta}\mathbf{U}}_{\mathrm{C}}^{(f)} = \underline{\underline{\mathbf{C}}}^{(f)}({\underline{\underline{\mathbf{C}}}^{(f)}}^{T}\underline{\underline{\mathbf{S}}}_{\mathrm{P}}^{(f)}\underline{\underline{\mathbf{C}}}^{(f)})^{-1}{\underline{\underline{\mathbf{C}}}^{(f)}}^{T}\underline{\mathbf{R}}_{\mathrm{C}}^{(f)}$$

➡ Iterative computation of a correction by a Krylov solver in a space orthogonal to Im(C)

$$\left(\underline{\underline{\mathbf{P}}}^{T}\underline{\underline{\mathbf{S}}}_{\mathrm{P}}^{(f)}\underline{\underline{\mathbf{P}}}\right)\underline{\boldsymbol{\delta}\mathbf{U}}_{\mathrm{K}}^{(f)} = \underline{\underline{\mathbf{P}}}^{T}\underline{\mathbf{R}}_{\mathrm{C}}^{(f)}$$

Results: MOR-based preconditioner



Weighted GPOD

$$J_{\mathrm{WPOD}}(\underline{\underline{\mathbf{C}}}, \underline{\boldsymbol{\alpha}}) = \int_{\mu\in\mathcal{T}} \int_{t\in\mathcal{T}} \Phi((t,\mu)) \|\underline{\mathbf{U}}(t) - \underline{\underline{\mathbf{C}}}\,\underline{\boldsymbol{\alpha}})(t,\mu)\|^2_{\underline{\underline{\mathbf{L}}}}\, dt\, d\mu$$

$$\|\underline{\mathbf{X}}\|^2_{\underline{\underline{\mathbf{L}}}} = \underline{\mathbf{X}}^T \underline{\underline{\mathbf{L}}}\, \underline{\mathbf{X}}$$

$\underline{\underline{\mathbf{L}}}$ diagonal operator with non-negative entries

$$J_{\mathrm{POD}}(\underline{\underline{\mathbf{C}}}, \underline{\boldsymbol{\alpha}}) = \int_{\mu\in\mathcal{T}} \int_{t\in\mathcal{T}} \Phi((t,\mu)) \|\underline{\mathbf{U}}(t) - \underline{\underline{\mathbf{C}}}\,\underline{\boldsymbol{\alpha}})(t,\mu)\|^2_{\underline{\underline{\mathbf{L}}}}\, dt\, d\mu$$

Progressive Weighted GPOD

- *Find* $(\underline{\underline{\mathbf{C}}}, \underline{\boldsymbol{\alpha}})$ *minimising*

$$J_{\mathrm{WPOD}}(\underline{\underline{\mathbf{C}}}, \underline{\boldsymbol{\alpha}}) = \int_{\mu \in \mathcal{T}} \int_{t \in \mathcal{T}} \Phi((t,\mu)) \| \underline{\mathbf{U}}(t) - \underline{\underline{\mathbf{C}}}\, \underline{\boldsymbol{\alpha}})(t,\mu) \|^2_{\underline{\underline{\mathbf{L}}}}\, dt\, d\mu$$

- *Update the weights* $(\Phi(t,\mu), \underline{\underline{\mathbf{L}}})$ *so that* $J_{\mathrm{POD}}(\underline{\underline{\mathbf{C}}}, \underline{\boldsymbol{\alpha}})$ *decreases sufficiently*

- *convergence check*

Progressive Restricted GPOD

- In our case:

$$J_{\mathrm{WPOD}}(\underline{\underline{\mathbf{C}}},\underline{\boldsymbol{\alpha}}) = \sum_{\mu\in\mathcal{P},\, t\in\mathcal{T}} \|\underline{\mathbf{U}}(t) - \underline{\underline{\mathbf{C}}}\,\underline{\boldsymbol{\alpha}}(t,\mu)\|^2_{\underline{\underline{\mathbf{P}}}_{\mathrm{R}}}$$

$$\underline{\underline{\mathbf{P}}}_{\mathrm{R}} = \underline{\underline{\mathbf{E}}}_{\mathrm{R}}\,\underline{\underline{\mathbf{E}}}_{\mathrm{R}}^{\;T}$$ diagonal boolean matrix, starting with identity

- Update:

$$\underline{\underline{\mathbf{P}}}_{\mathrm{R}} \leftarrow \underline{\underline{\mathbf{P}}}_{\mathrm{R}}\,\underline{\underline{\mathbf{P}}}_{\mathrm{Up}}$$

$$J_{\mathrm{WPOD}}(\underline{\underline{\mathbf{C}}},\underline{\boldsymbol{\alpha}}) = \sum_{\mu\in\mathcal{P},t\in\mathcal{T}} \underline{\epsilon}(t,\mu)^T\underline{\underline{\mathbf{P}}}_{\mathrm{R}}\underline{\epsilon}(t,\mu)$$

$$J_{\mathrm{WPOD}}(\underline{\underline{\mathbf{C}}},\underline{\boldsymbol{\alpha}}) = \sum_{\mu\in\mathcal{P},t\in\mathcal{T}}\sum_{i=1}^{n}\epsilon_i^2(t,\mu)\mathbf{P}_{\mathrm{R}ii} = \sum_{i=1}^{n}\sum_{\mu\in\mathcal{P},t\in\mathcal{T}}\epsilon_i^2(t,\mu)\mathbf{P}_{\mathrm{R}ii}$$

$$\mathbf{P}_{\mathrm{Up}ii} = (1-\delta_{ij}) \qquad j = \operatorname{argmax}_{i\in[\![1,n]\!]}\sum_{\mu\in\mathcal{P},t\in\mathcal{T}}\epsilon_i^2(t,\mu)\mathbf{P}_{\mathrm{R}ii}$$