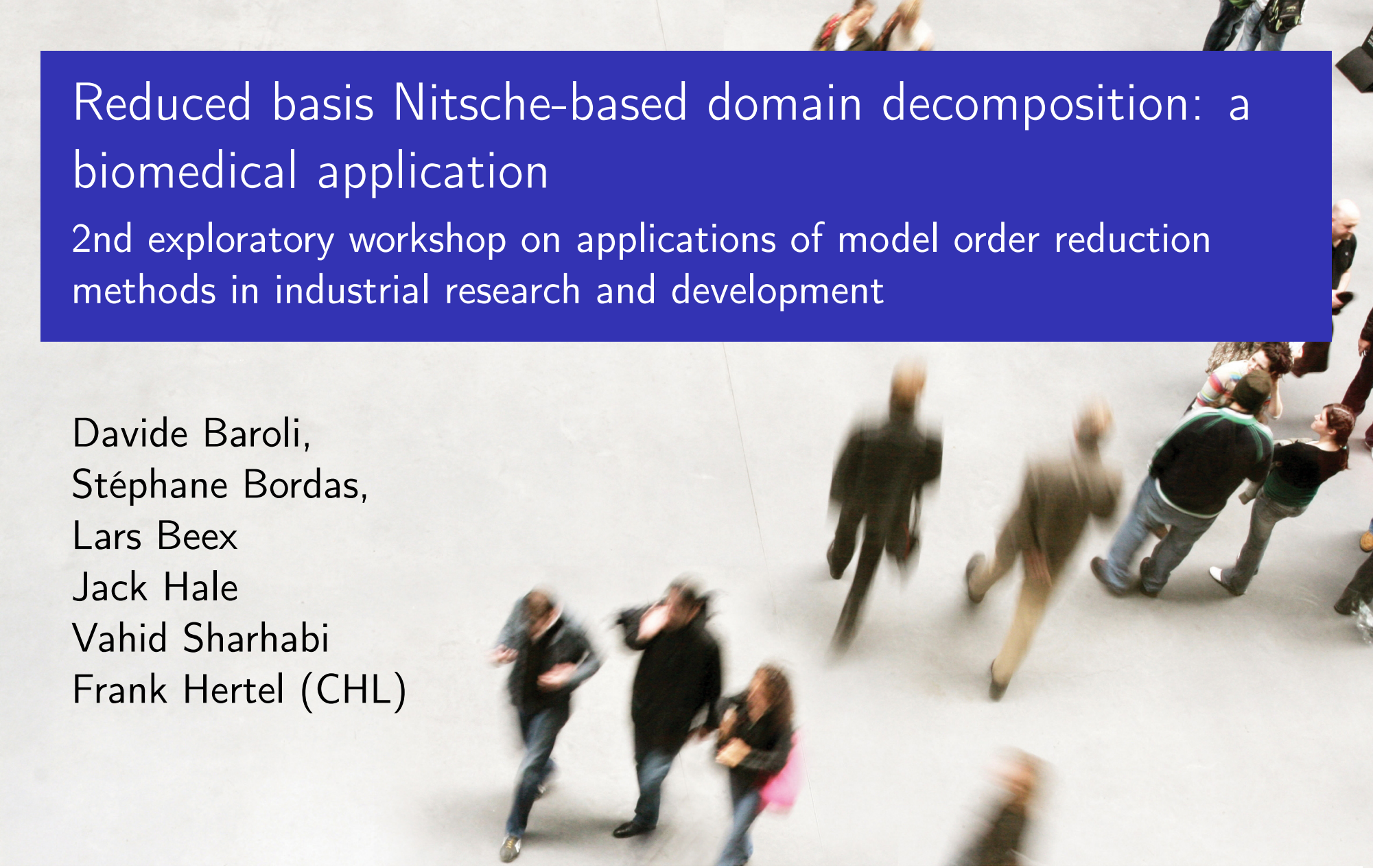
Reduced basis Nitsche-based domain decomposition: a biomedical application

2nd exploratory workshop on applications of model order reduction methods in industrial research and development

Davide Baroli,
Stéphane Bordas,
Lars Beex
Jack Hale
Vahid Sharhabi
Frank Hertel (CHL)



RUES | RESEARCH UNIT IN ENGINEERING SCIENCES

UNIVERSITÉ DU LUXEMBOURG

Outline

Case of study: real-time simulation for Kyphoplasty

In personalized healtcare application, a current challenge is to provide fast and reliable simulation to support decisions in surgery planning.

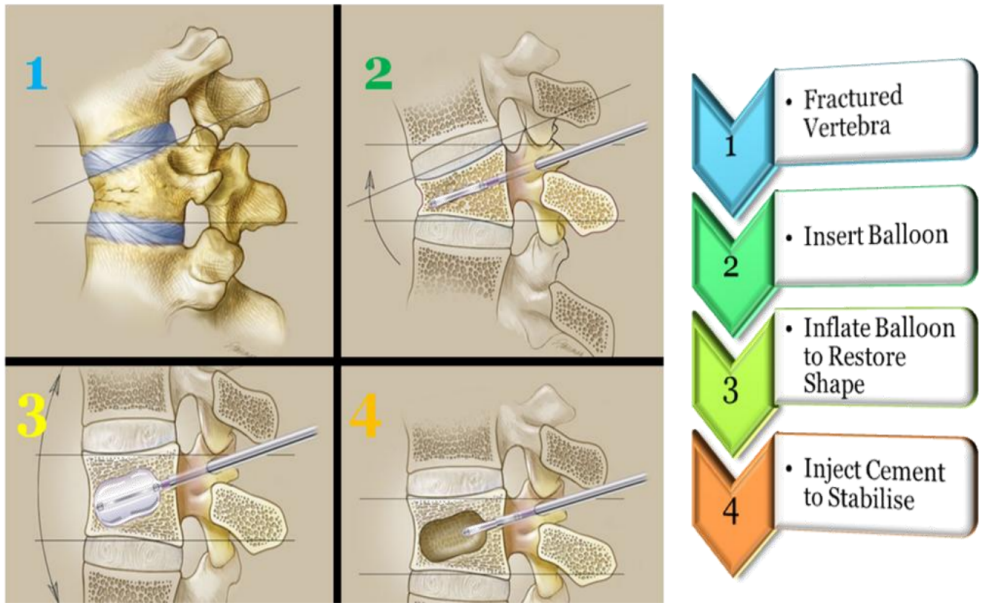


Image Segmentation

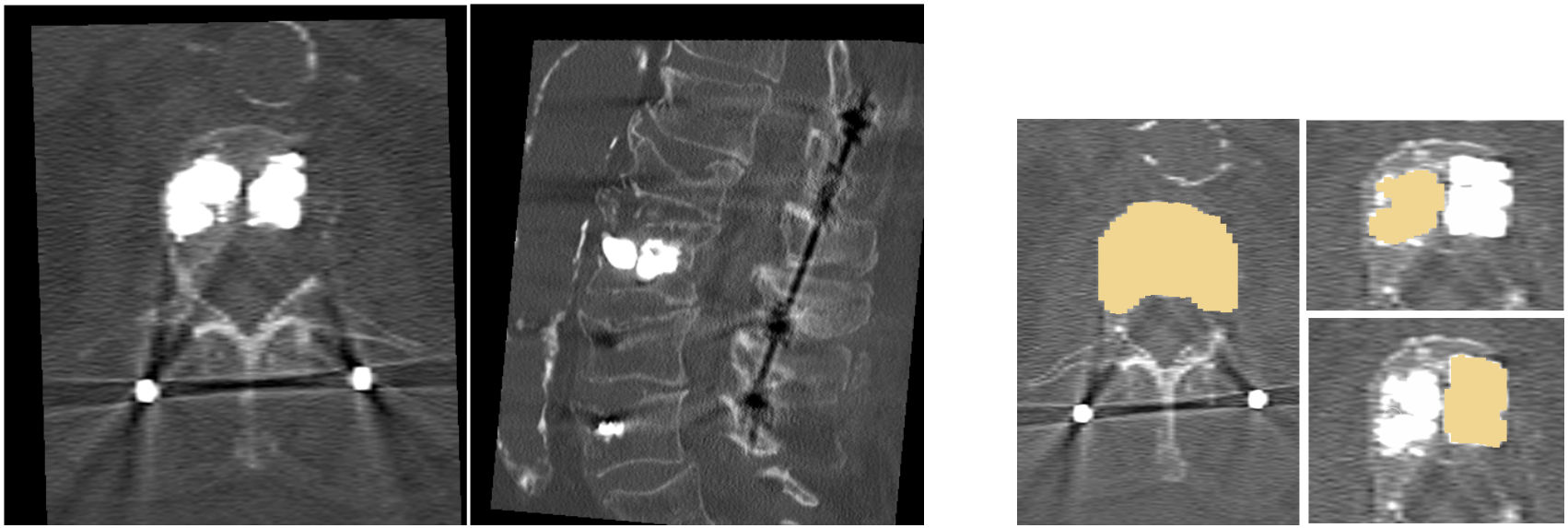


Figure: CT-Scan image provided by Dr. Hertel (CHL);Slicer segmentation of a layer

Computation model

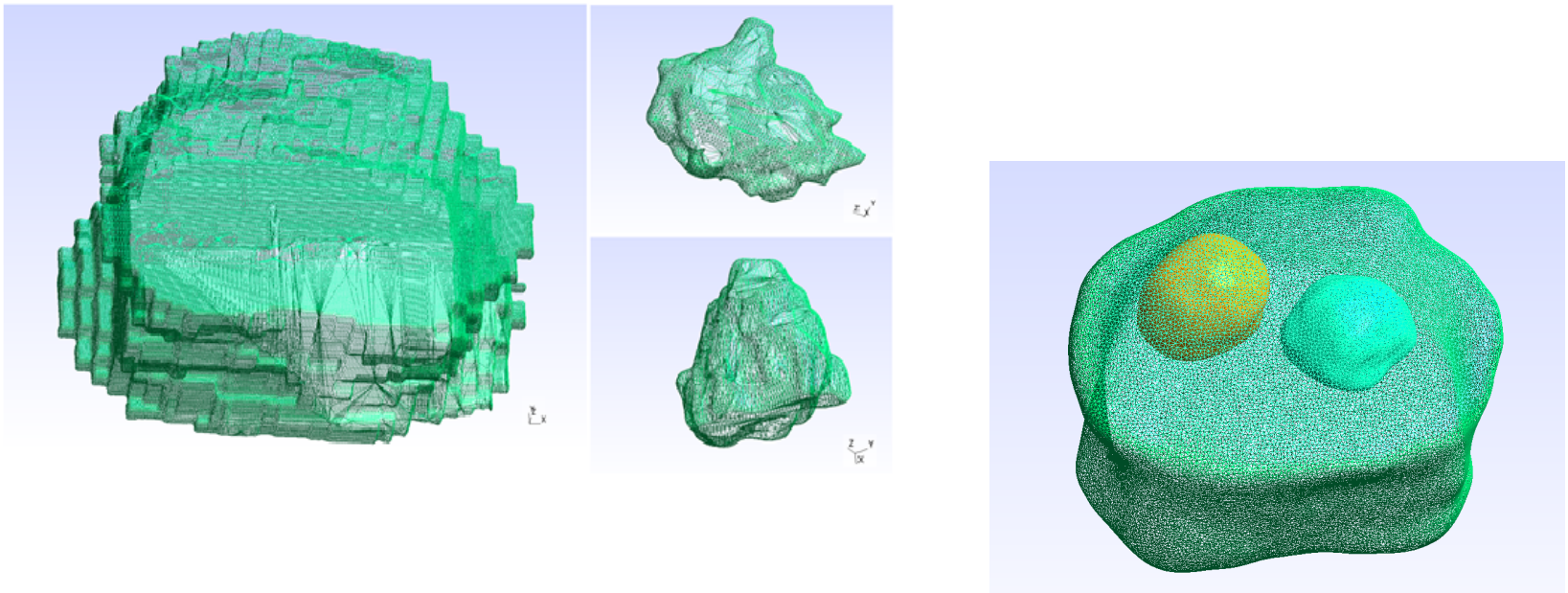


Figure: 3D surface reconstruction; geometrical processing: smoothing

Model: compressible linear eleasticity

The vertebra are considered to undergo small-deformations. The linear compressible elasticity problem reads: find the displacement u such that

$$\begin{aligned} -div(\sigma(\mathbf{u}, \mu, \lambda)) &= f(\beta, \rho, c_i) \quad \text{in } \Omega \\ \mathbf{u} &= 0 \quad \text{on } \partial\Gamma_D \\ \sigma(\mathbf{u}, \mu, \lambda) \cdot \mathbf{n} &= \mathbf{g} \quad \text{on } \partial\Gamma_N \end{aligned}$$

Here the stress tensor σ is related to the displacement by Hooke's law:

$$\sigma(\mathbf{u}, \mu, \lambda) = 2\mu\epsilon(\mathbf{u}) + \lambda tr(\epsilon(\mathbf{u}))\mathbf{I} \quad \text{in } \Omega$$

And the source is defined as : $f(\beta, \rho, c_i) = (0.0,\ -\rho \cdot 9.8 \cdot e^{-\sum_{i=1}^{3} c_i \cdot (x_i - \beta_i)^2})$

Finite element simulation: FEniCS

On simulation performance perspective: We are using FEniCS:

- Flexible code style with respect different code languages
- DSEL based-high level programming (easy to simulate complex physical-mathematical model)
- High scalability performance in assembly and solver based on parallel distributed linear algebra
- Universal installation on different operating system using light containers (Docker)
- Modifying the constitutive material law is done using one line of code and differentiation is automatic

Reliable and fast method: model reduction method

Given a solution set $\mathcal{M}_h = \{u_h(\mu) : \mu \in \mathcal{P}\}$ of high-fidelity problem, the RB method finds a few functions $\{\xi_i\}_{i=1}^N$ such that:

$$u_h(\mu) = \sum_{i=1}^{N} u_N(\mu)^i \xi_i$$

where the coefficients $u_N(\mu)$ could be obtained via Galerkin reduced basis approximation:

$$[\xi_1|\ldots|\xi_N]^T \mathbb{A}_h(\mu)[\xi_1|\ldots|\xi_N] u_N = [\xi_1|\ldots|\xi_N]^T \mathbf{f}_h(\mu)$$

with $\mathbb{A}_h$ and $\mathbf{f}_h$ are the discrete counterpart of the differential operator.

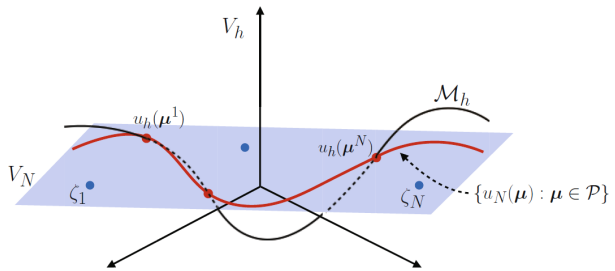


Figure: From "Reduced Basis Method for PDEs"(A.Quarteroni et al.)

RUES | RESEARCH UNIT IN ENGINEERING SCIENCES

UNIVERSITÉ DU LUXEMBOURG

Reduced Basis technique: Greedy algorithm using SLEPc

Given tol and set of parameters $\{\mu_1, \ldots, \mu_n\}$.
Initialize the Reduced Basis space:

```
reducedbasis = SLEPc.BV().create(Comm)
```

Reduced Basis technique: Greedy algorithm using SLEPc

Given tol and set of parameters $\{\mu_1, \ldots, \mu_n\}$.

1. Compute high-fidelity solution $u_h(\mu_1)$ for μ_1

Reduced Basis technique: Greedy algorithm using SLEPc

Given tol and set of parameters $\{\mu_1, \ldots, \mu_n\}$.

1. Compute high-fidelity solution $u_h(\mu_1)$ for μ_1
2. Set $\mathbf{V}_{rb} = span\{u_h(\mu_1)\}$.
   `reducedbasis.insertVec(index, function)`

Reduced Basis technique: Greedy algorithm using SLEPc

Given tol and set of parameters $\{\mu_1, \ldots, \mu_n\}$.

1. Compute high-fidelity solution $u_h(\mu_1)$ for μ_1
2. Set $\mathbf{V}_{rb} = span\{u_h(\mu_1)\}$.
3. For each $\mu \in \mathcal{P}$
   Compute $u_{RB} = \sum_{i=1}^{N} u_N^i \xi_i(\mu)$ where ξ_i are the basis of $\mathbf{V}_{rb}$.
   Evaluate the error estimator $\eta(\mu) = \frac{(res_{RB}, M^{-1} res_{RB})}{\alpha_{LB}}$

Reduced Basis technique: Greedy algorithm using SLEPc

Given tol and set of parameters $\{\mu_1, \ldots, \mu_n\}$.

1. Compute high-fidelity solution $u_h(\mu_1)$ for μ_1
2. Set $\mathbf{V}_{rb} = span\{u_h(\mu_1)\}$.
3. For each $\mu \in \mathcal{P}$
   Compute $u_{RB} = \sum_{i=1}^{N} u_N^i \xi_i(\mu)$ where ξ_i are the basis of $\mathbf{V}_{rb}$.
   Evaluate the error estimator $\eta(\mu) = \frac{(res_{RB}, M^{-1} res_{RB})}{\alpha_{LB}}$
4. Choose $\tilde{\mu} = arg\ max_{\mu \in \mathcal{P}} \eta(\mu)$

Reduced Basis technique: Greedy algorithm using SLEPc

Given tol and set of parameters $\{\mu_1, \ldots, \mu_n\}$.

1. Compute high-fidelity solution $u_h(\mu_1)$ for μ_1
2. Set $\mathbf{V}_{rb} = span\{u_h(\mu_1)\}$.
3. For each $\mu \in \mathcal{P}$
   Compute $u_{RB} = \sum_{i=1}^{N} u_N^i \xi_i(\mu)$ where ξ_i are the basis of $\mathbf{V}_{rb}$.
   Evaluate the error estimator $\eta(\mu) = \frac{(res_{RB}, M^{-1} res_{RB})}{\alpha_{LB}}$
4. Choose $\tilde{\mu} = arg\ max_{\mu \in \mathcal{P}} \eta(\mu)$
5. If $\eta(\tilde{\mu}) > tol$ and $V_{RB} = V_{RB} \cup \{u(\tilde{\mu})\}$
   `reducedbasis.orthogonalizeVec(function)`
   `reducedbasis.scaleColumn(norm)`

Online Galerkin Project with SLEPc

$$[\xi_1|\dots|\xi_N]^T \mathbb{A}_h(\mu)[\xi_1|\dots|\xi_N]u_N = [\xi_1|\dots|\xi_N]^T \mathbf{f}_h(\mu)$$

with $\mathbb{A}_h$ and $\mathbf{f}_h$ are the discrete counterpart of the differential operator.

```
reducedbasis.matProject(A, reducedbasis)
reducedbasis.dotVec(f)
```

Reduced Integration

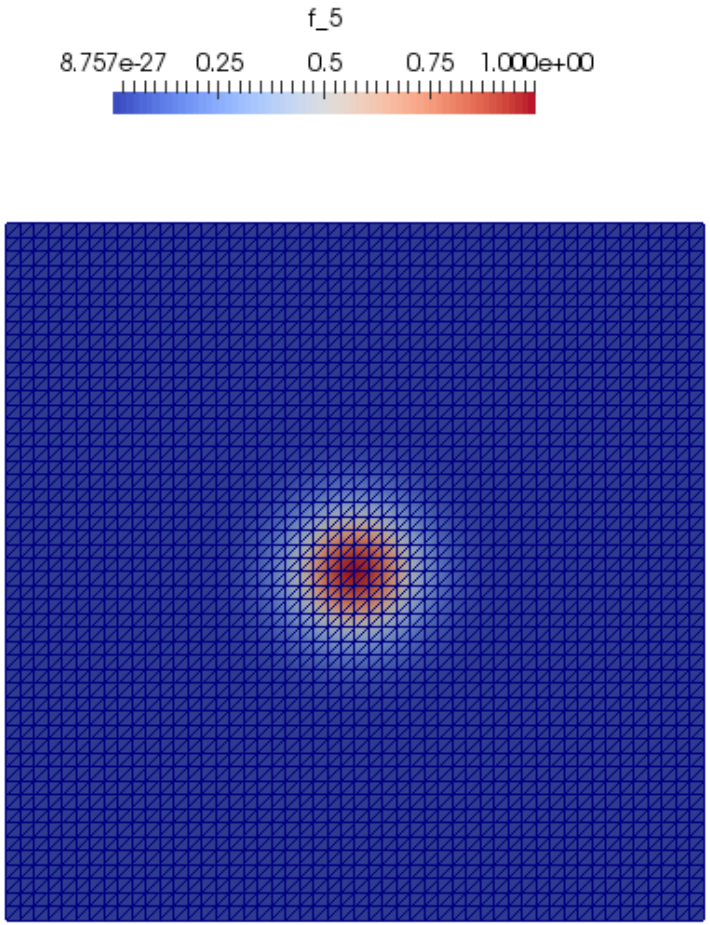
One of the major computational efforts in application of standard model reduction to linear (non-linear) PDEs is the assemble the finite element space operators and project onto the ROM basis.
We employ the novel technique "reduced integration“ which consists in: Given a subspace of reduced basis $V_{RB} = \{\xi_1, \ldots, \xi_N\}$, quadrature rule order m and a coarse mesh $\Omega_H := \mathcal{T}_H(\Omega)$: For each basis ξ_i and a tolerance *tol*.

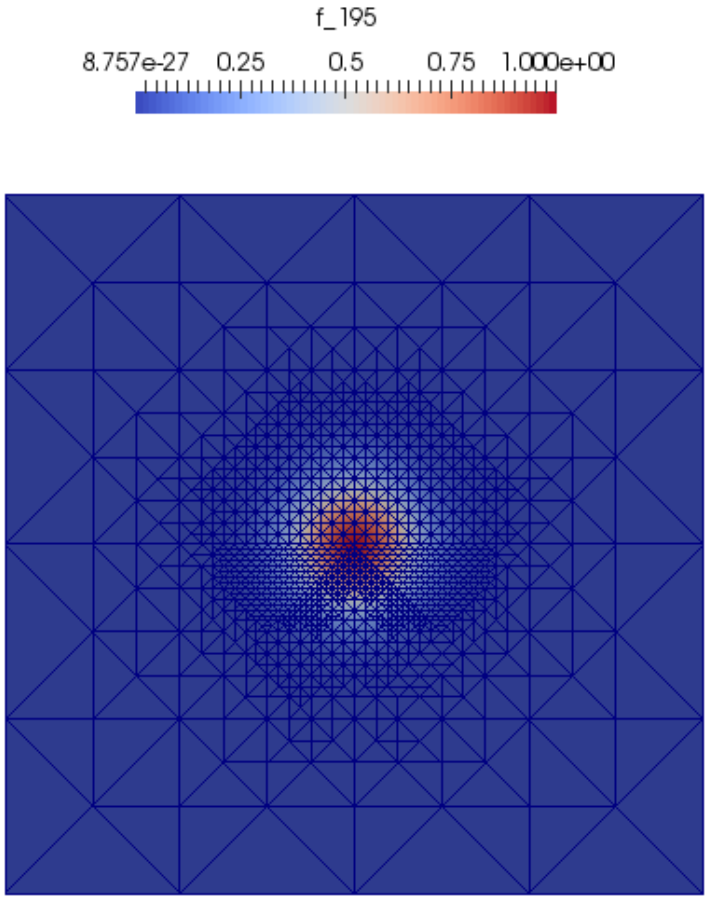
1. Compute the estimator

$$\eta(\xi_i, \Omega_H) = \int_{\Omega_H, m} |\xi_i| - \int_{\Omega_H, m+1} |\xi_i| \tag{1}$$

2. If $\eta(\xi_i, \Omega_H) > tol$ then $\Omega_H = \mathrm{REFINE}(\Omega_H)$ and go to **1**..

Reduced Integration test



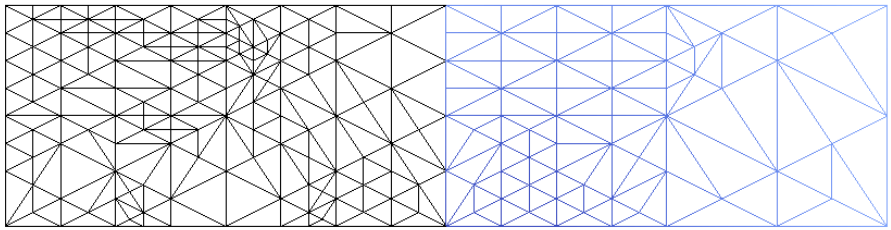


Locally reduced integration

Due to local behaviour of modes/basis which is sensitive to the zone of application of the external force/moments, we combine the domain-decomposition paradigm "divide-et-impera" with reduced integration technique.
The algorithm consist in

- Decompose the physical domain in part (regular - ad hoc)
- Restrict each reduced basis on each subdomain, constructing the space $V_{RB,i}$ for $i = 1, \ldots, \mathrm{NbSubDomains}$
- Perform in each subdomain the reduced-integration
- Re-orthonormalization of each local space.

Non-conforming impera algorithm



Some domain decomposition technique: FETI-DP, BDDC, Mortar, Discontinuous Galerkin, InterNODES, Nitsche, ...
ROM + domain decomposition: Reduced Basis Element [RBE], DG-RBE, static condensation reduced basis element, Latin Multiscale

Why we choose Nitsche technique?

Pros.

- do not require any additional unknown
- has the same accurancy property of the underling discretization
- avoid of mesh burden near interface
- could treat also overlapping subdomains

Cons.
The stability of Nitsche method depends on the choose of a penalty scalar value. However, it is possible to estimate this constant using trace inequality (S. Claus et al.).

The Nitsche-type weak formulation associated to this problem is: Find a $u_h \in V_h^k$ such that

$$a_h(u_h, v_h) - b_h(u_h, v_h) - b_h(v_h, u_h) + j_h(u_h, v_h) = L_h(v_h) \quad \forall\, v_h \in V_h^k$$

The bilinear form a_h and b_h are defined as

$$a(u_h, v_h) = \sum_{i=1}^{2} \int_{\Omega_i} \sigma(u_h^i, mu_i, \lambda_i) : \epsilon(v_h^i) dX$$

$$b(u_h, v_h) = \int_{\Gamma} \{(\sigma(u_h, mu, \lambda))\}_{\kappa}, [\![(v \cdot n)]\!] d\Gamma$$

and the penalty-ghost form is

$$j(u_h, v_h) = \int_{\Gamma} \frac{\alpha}{\{h\}} [\![u^h]\!][\![v^h]\!] d\Gamma$$

Create decomposed mesh:

```
multimesh=dolfin.MultiMesh()
multimesh.add(mesh)
multimesh.build()
```

The MultiMeshForm for the first subdomain:

```
a = inner(sigma(u,mu0,lmbda0), grad(v))*dX
b_u= - inner(w_avg(sigma(u,mu0,lmbda0),kappa0,kappa1),
          tensor_jump(v,n))*dI
b_v=- inner(w_avg(sigma(v,mu0,lmbda0),kappa0,kappa1),
          tensor_jump(u,n))*dI
j= alpha/avg(h)*inner(jump(u), jump(v))*dI
```

The bilinear form a_h and b_h are defined as

$$a(u_h, v_h) = \sum_{i=1}^{2} \int_{\Omega_i} \sigma(u_h^i, mu_i, \lambda_i) : \epsilon(v_h^i) dX$$

$$b(u_h, v_h) = \int_{\Gamma} \{(\sigma(u_h, mu, \lambda))\}_{\kappa}, [\![(v \cdot n)]\!] d\Gamma$$

and the penalty-ghost form is

$$j(u_h, v_h) = \int_{\Gamma} \frac{\alpha}{\{h\}} [\![u^h]\!][\![v^h]\!] d\Gamma$$

where the penalty parameter is

$$\alpha = \frac{C_I(d, p) \cdot (E_1 \cdot E_2)}{E_1 + E_2}$$

with d is physical dimension and p the order of the polynomial Lagrange discretization space.

The bilinear form a_h and b_h are defined as

$$a(u_h, v_h) = \sum_{i=1}^{2} \int_{\Omega_i} \sigma(u_h^i, mu_i, \lambda_i) : \epsilon(v_h^i) dX$$

$$b(u_h, v_h) = \int_{\Gamma} \{(\sigma(u_h, mu, \lambda))\}_{\kappa}, [\![(v \cdot n)]\!] d\Gamma$$

and the penalty-ghost form is

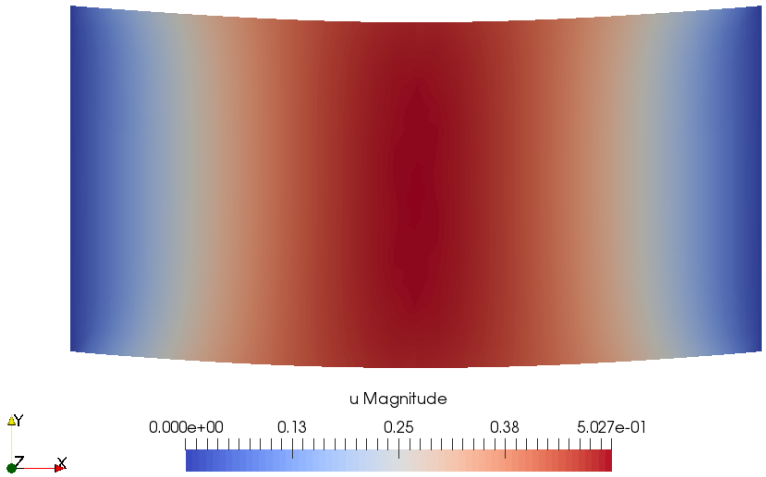
$$j(u_h, v_h) = \int_{\Gamma} \frac{\alpha}{\{h\}} [\![u^h]\!][\![v^h]\!] d\Gamma$$

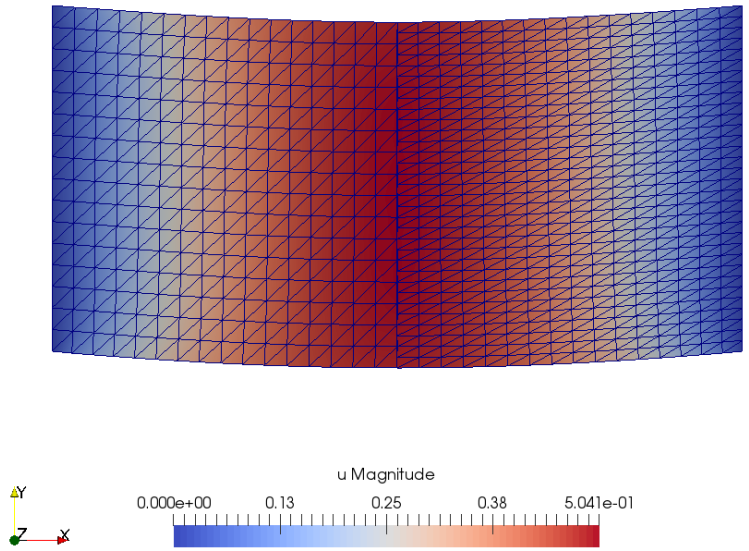
$$\{\mu u\}_{\kappa} = (\kappa_1 \mu_1 u^1 + \kappa_2 \mu_2 u^2)_{|\Gamma}$$

where the weights are

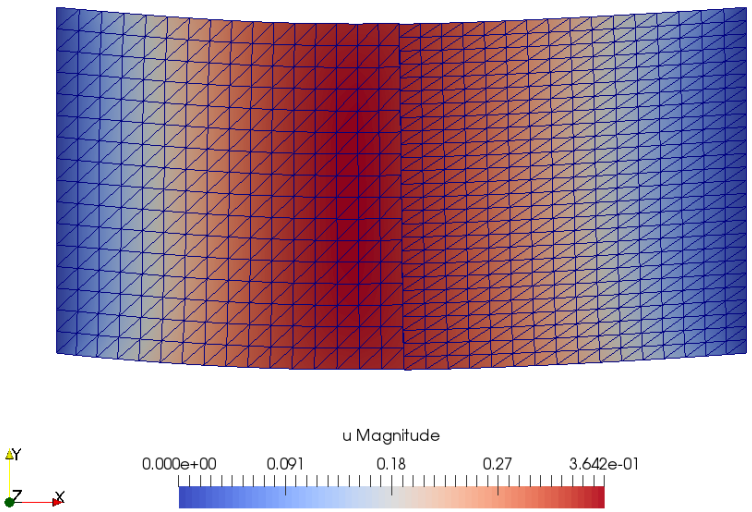
$$\kappa_1 = \frac{E_1}{E_1 + E_2}, \quad \kappa_2 = \frac{E_2}{E_1 + E_2}$$

Numerical test with single material

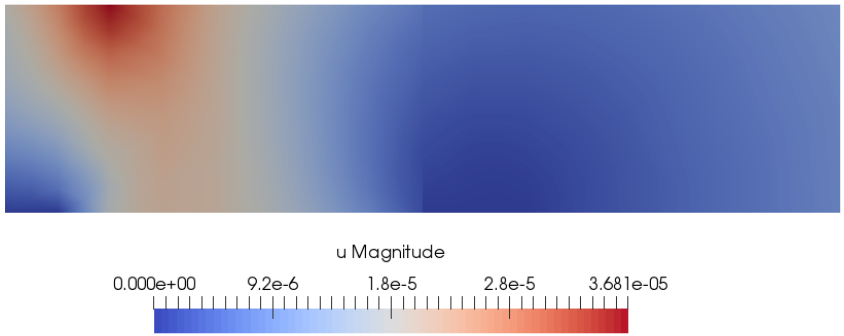


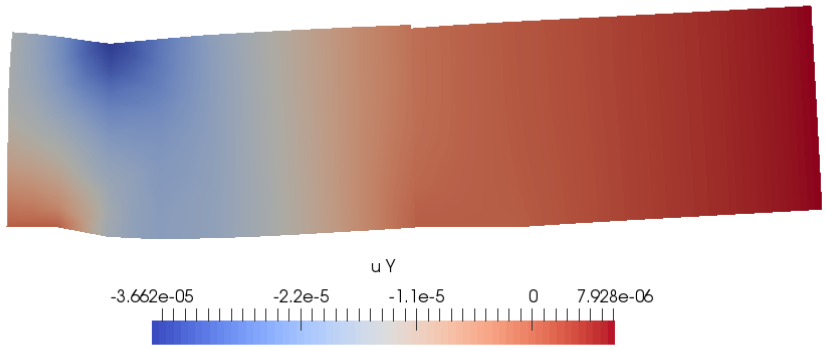


Numerical test with two materials



Numerical Validation Test:2D



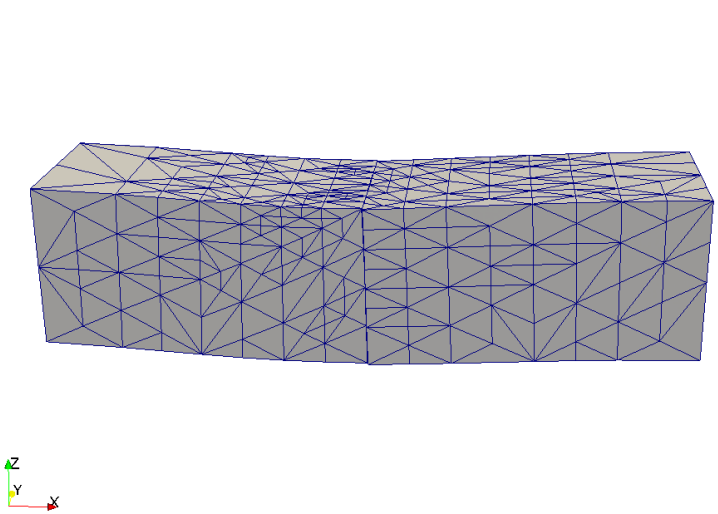


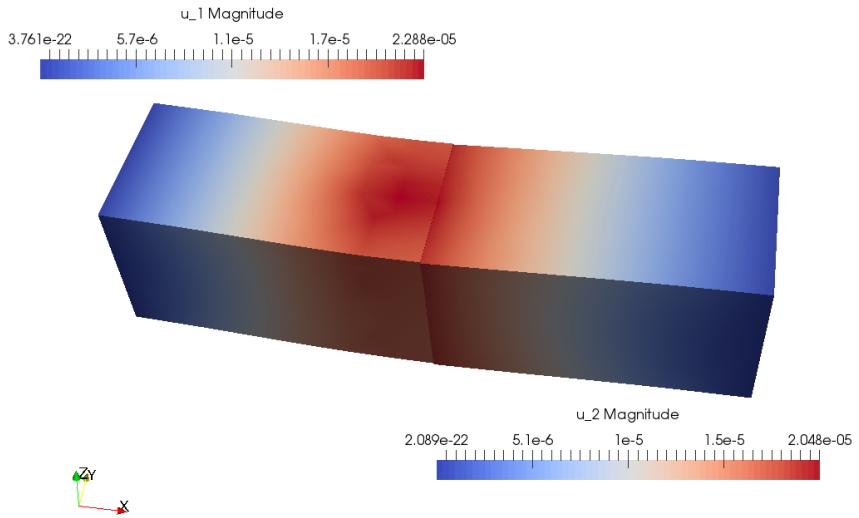
Number of dof of high-fidelity discretization:7442
Number of dof of reduced integration discretization: 2796
Speed-up: x40 time with iRB+Nitsche
Speed-up: x10 time with reduced integration in assemble. Error in first subdomain L_2-norm:=3.92e-07, H_1-norm=2.03e-06
Error in first subdomain L_2-norm:=2.23e-08, H_1-norm=5.47e-08

RUES | RESEARCH UNIT IN ENGINEERING SCIENCES

UNIVERSITÉ DU LUXEMBOURG

Numerical Validation Test:3D



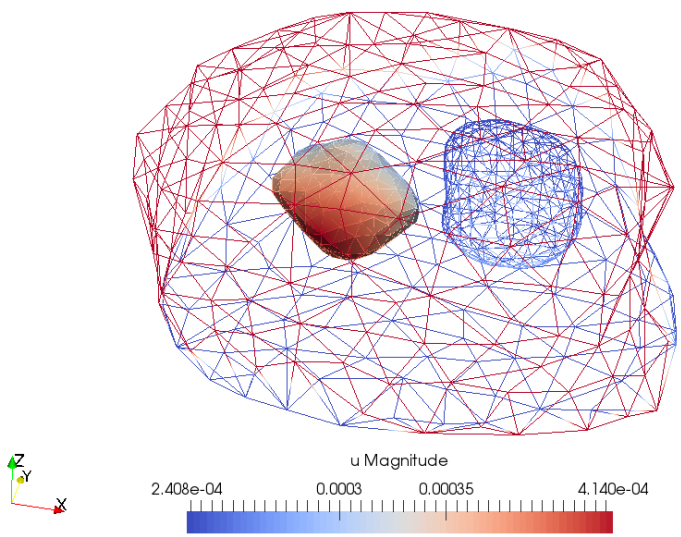


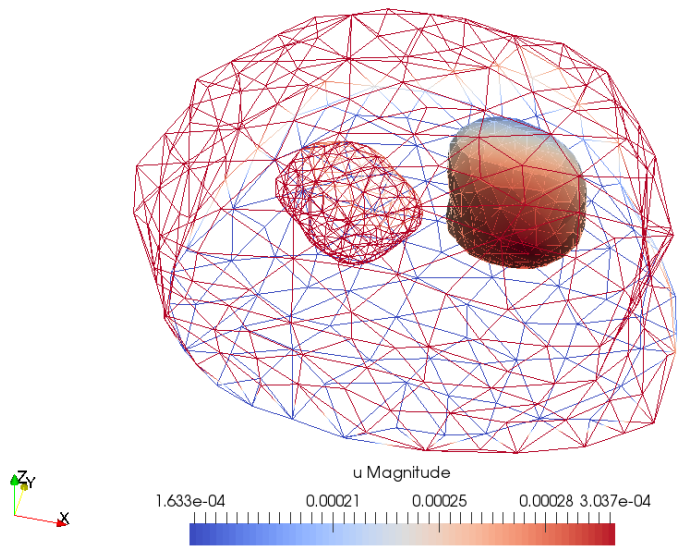
Number of dof of high-fidelity discretization:9600
Number of dof of reduced integration discretization: 6953
Speed-up: x1000 time with iRB+Nitsche
Speed-up: x8 time with reduced integration in assemble.

RUES | RESEARCH UNIT IN ENGINEERING SCIENCES

UNIVERSITÉ DU LUXEMBOURG

Patient-specific application: Vertebra augemented by kyphoplasty

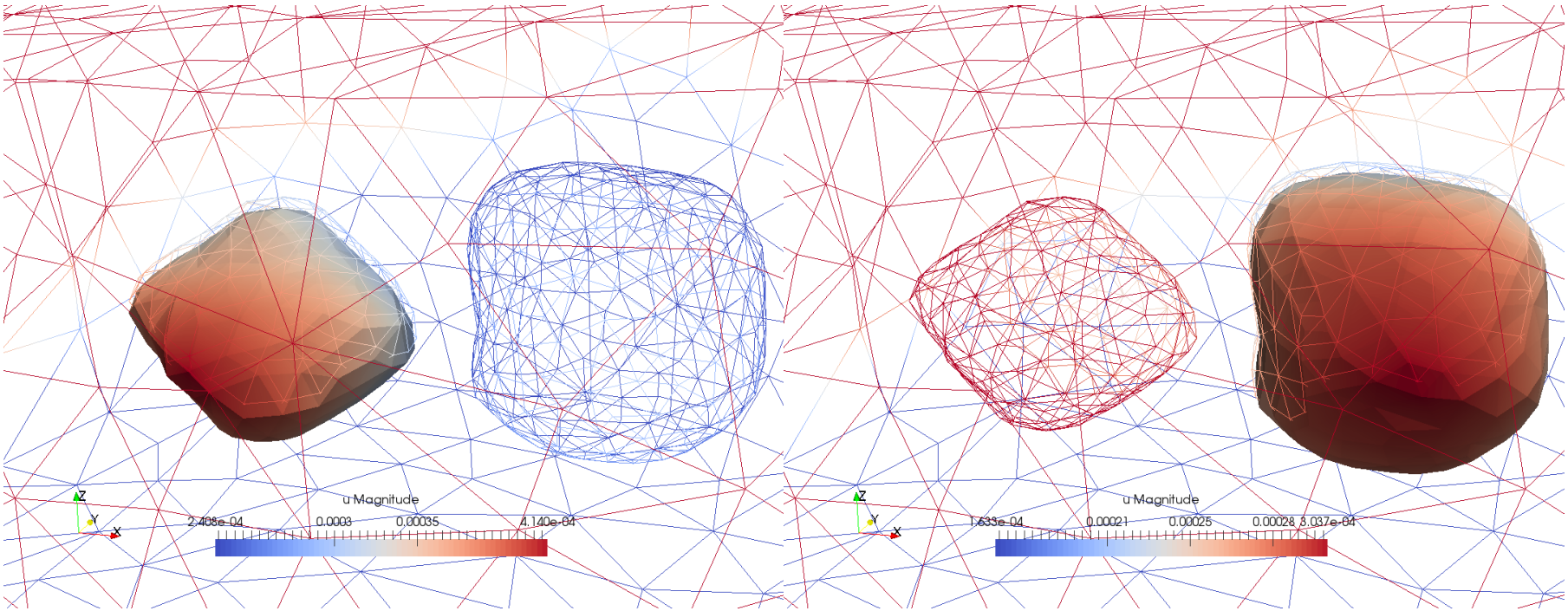




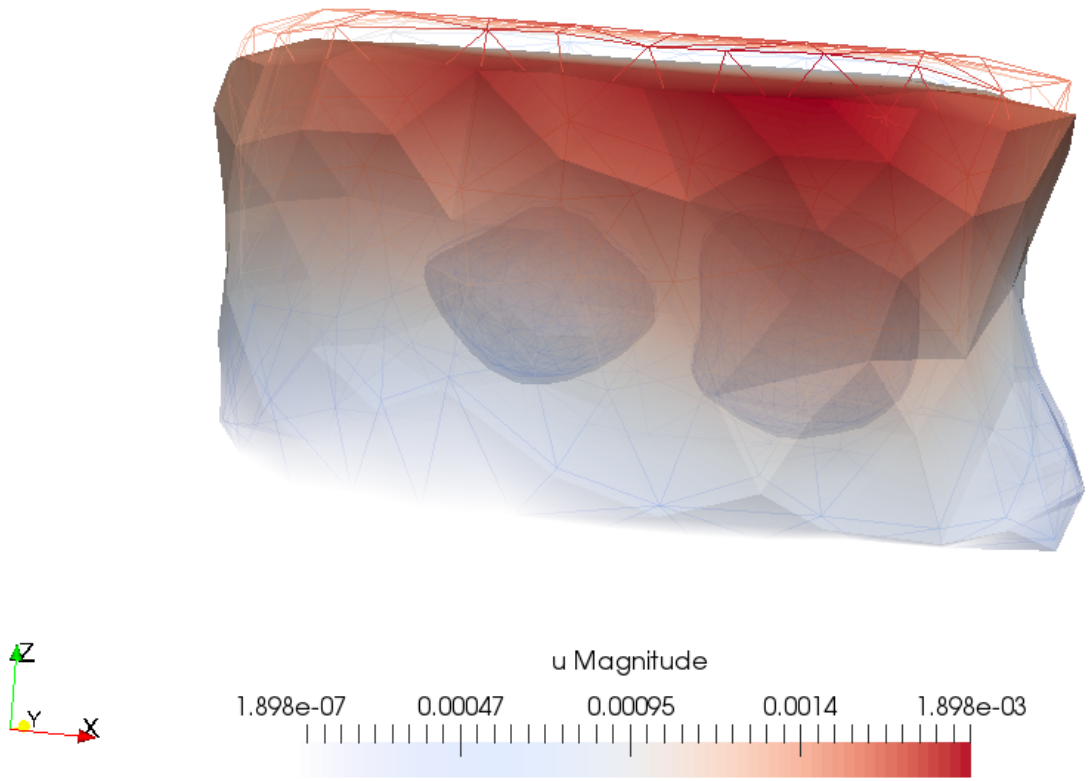
RUES | RESEARCH UNIT IN ENGINEERING SCIENCES

UNIVERSITÉ DU LUXEMBOURG

Reference vs Deformed Domain for each cemented domain



Deformation of the surface of Vertebra



Conclusion

Preliminary numerical comparison evidences:

- flexible approach in online gluing with Nitsche formulation;
- assemble speed-up by local reduced integration;
- iRB-Nitsche provides a fast and reliable simulation in patient-specific scenario;

Working in progress:

- integration of iRB-Nitsche in multi-query UQ algorithm.

Future perspective: Contact Bio-mechanics problem

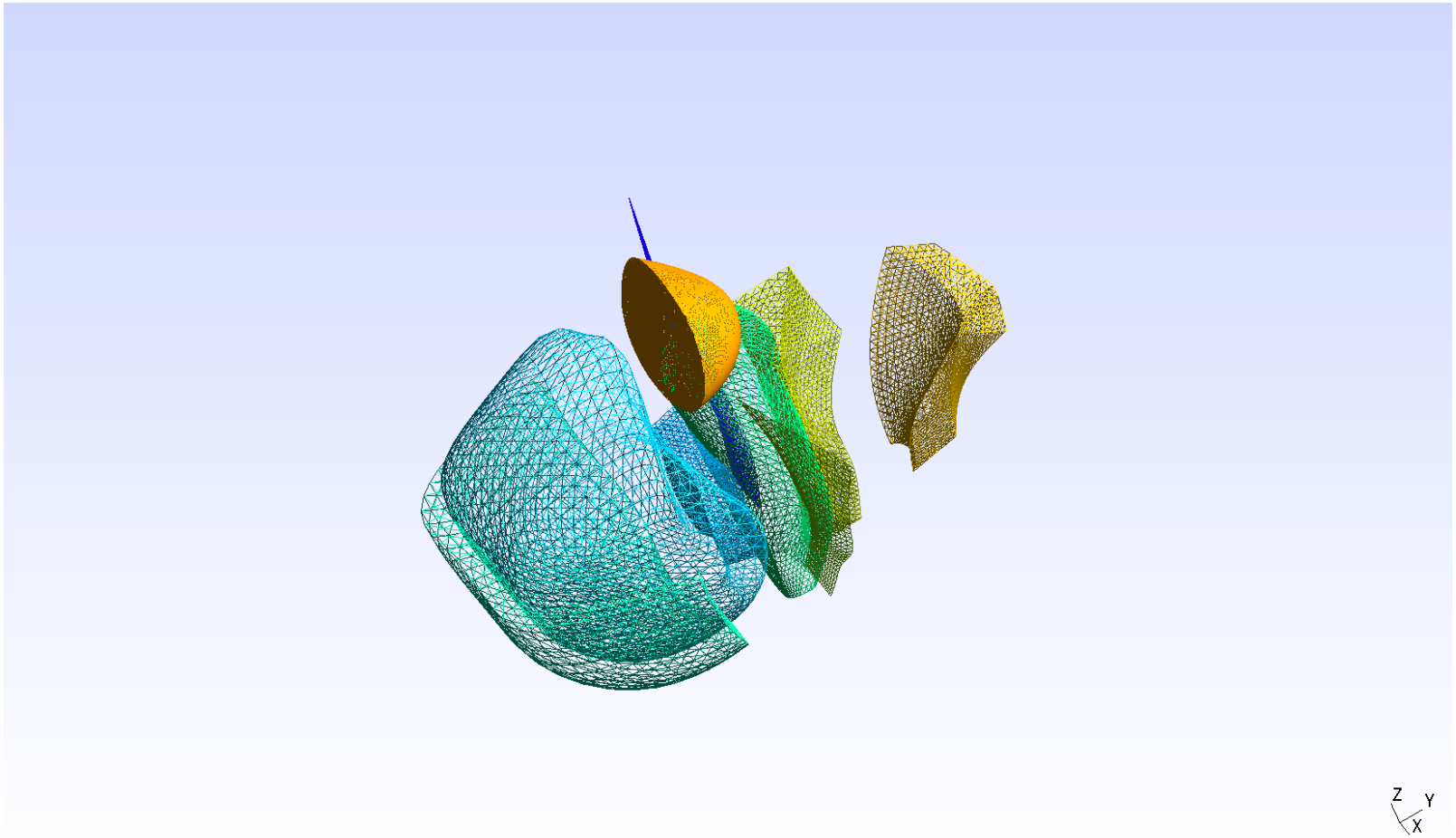


Figure: Computational mesh of pelvic organ surface provided by Prof. M.Brieu (Lille)

RUES | RESEARCH UNIT IN ENGINEERING SCIENCES

UNIVERSITÉ DU LUXEMBOURG

Acknowledgements

Computational sciences priority
European Research Council Starting Independent Research Grant (ERC Stg grant agreement No. 279578) entitled “Towards real time multiscale simulation of cutting in non-linear materials with applications to surgical simulation and computer guided surgery" led by Stéphane P. A. Bordas
Thanking: Dr.Hertel, S. Claus, V. Sharhabi.

RUES | RESEARCH UNIT IN ENGINEERING SCIENCES

UNIVERSITÉ DU LUXEMBOURG

Thank you for your attention