Simulating topological changes in real time

error estimation & adaptivity

model order reduction

uncertainty quantification & material model selection

Pierre Kerfriden[a], Hadrien Courtecuisse[b], Christian Duriez[c],
Stéphane Cotin[d], Stéphane P. A. Bordas[a,e,f]

[a] institute of Mechanics and Advanced Materials, Cardiff University, *UK*
[b] CNRS, iCube, Strasbourg, *France*
[c] DEFROST Project, INRIA, Lille, *France*
[d] MIMESIS Project, INRIA, Strasbourg, *France*
[e] University of Luxembourg, Computational Sciences, *Luxembourg*
[f] University of Western Australia, *Australia*

* **Joint work with**
Olivier Goury, *DEFROST Project, INRIA, Lille, France*
Phuoc Huu Bui, *iCube, Strasbourg, USIAS*
Satyendra Tomar, Paul Hauseux, Jack Hale,
University of Luxembourg, Computational Sciences, Luxembourg

UNIVERSITÉ DU LUXEMBOURG

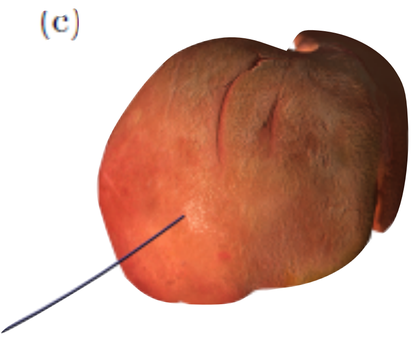
European Research Council
Established by the European Commission
Stg. No. 279578 RealTCut

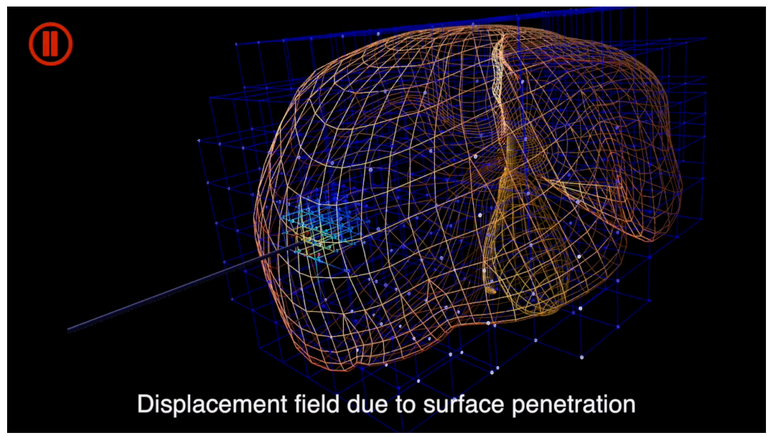
legato-team.eu


dic Table of the Finite Elements
O'NEILL

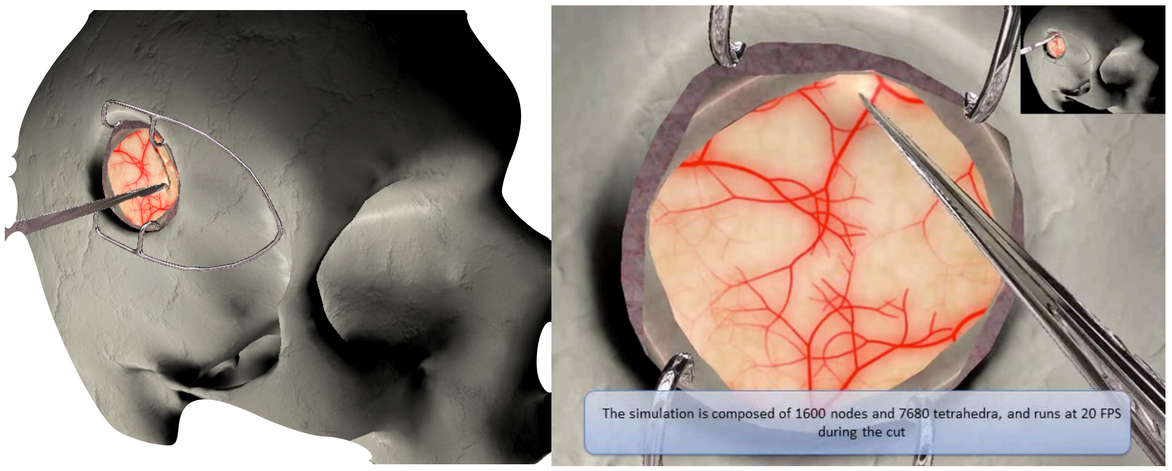
Reliable Simulations for Computer Integrated Surgery

ERC RealTCut





Real-time error-estimator driven mesh adaptation
Bui, Cotin, Bordas et al., 2016



Real-time implicit simulation of cutting in heterogeneous soft tissues
Courtecuisse, Cotin, Duriez, Bordas et al., Medical image analysis 2014

Long-term VISION

Provide real-time guidance to surgeons in uncertain conditions

- *Predict the risks of surgical steps*
- *Explore various courses of action*
- *Warning mechanisms during surgery*

Control errors
Control uncertainty
Choose the right material model

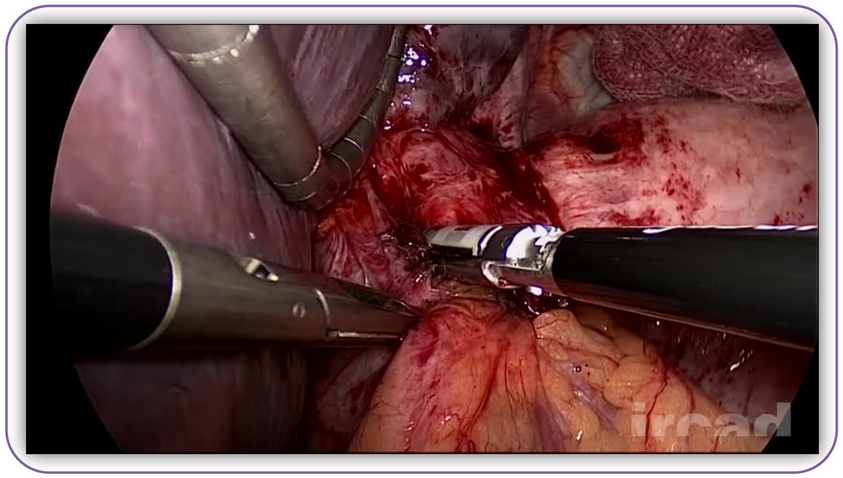
In a patient specific way
In "real time"

AIM **Build reliable simulations for patient-specific surgical assistance and planning.**

Medical Simulations

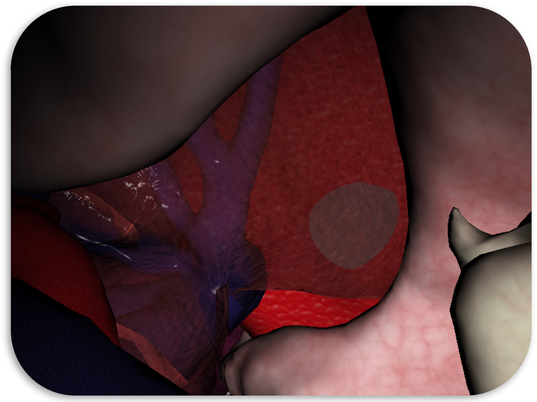
- Surgery : complex practice
  - Require a long training
  - Involves many risks
- New technologies :
  - Mini-invasive surgery, interventional radiology
  - Interface the surgeon and the patient
  - Video control, no haptic sensations
- Reproduce a similar surgical field

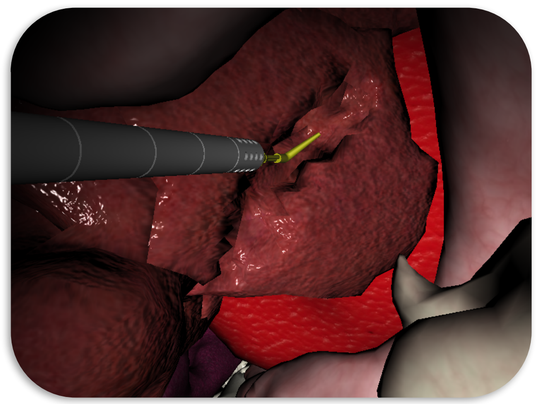
- Create a virtual representation of the patient :
  - Give a bio-mechanical behavior



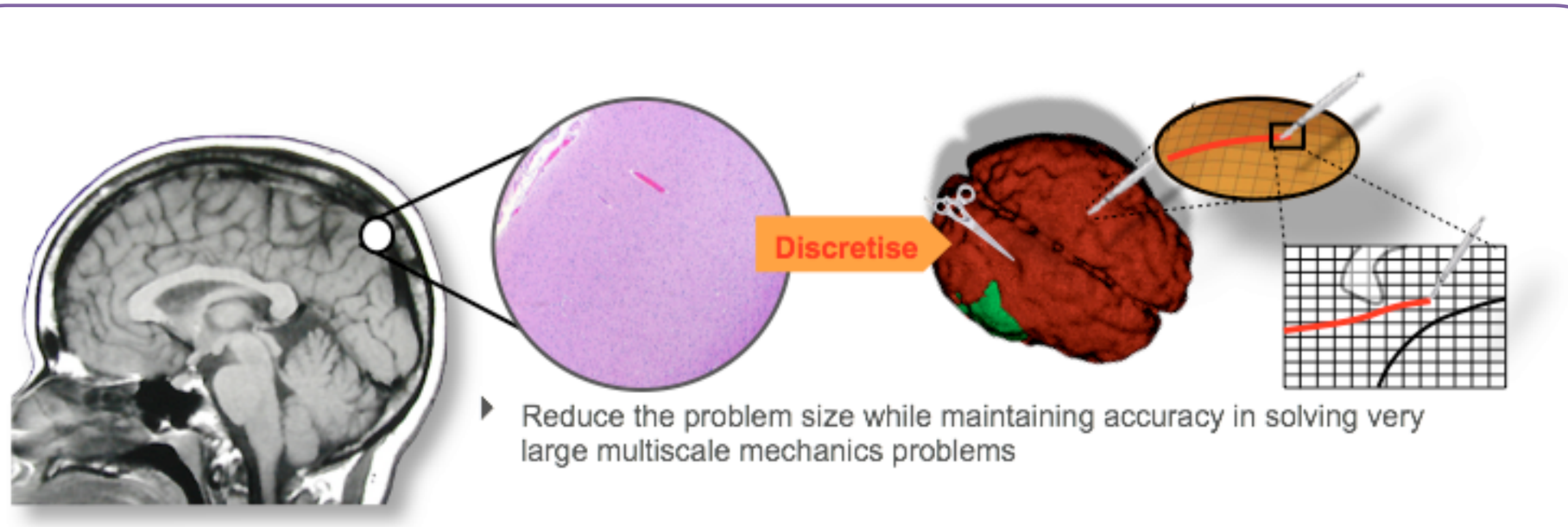
Medical Simulations

- Medical simulations implies :
  - User Interaction / Real-time computation / Stability
  - Computation based on physics law of deformable bodies
  - Accurate contact response (friction, mechanical coupling)
  - Heterogeneous materials
  - Patient specific / validation





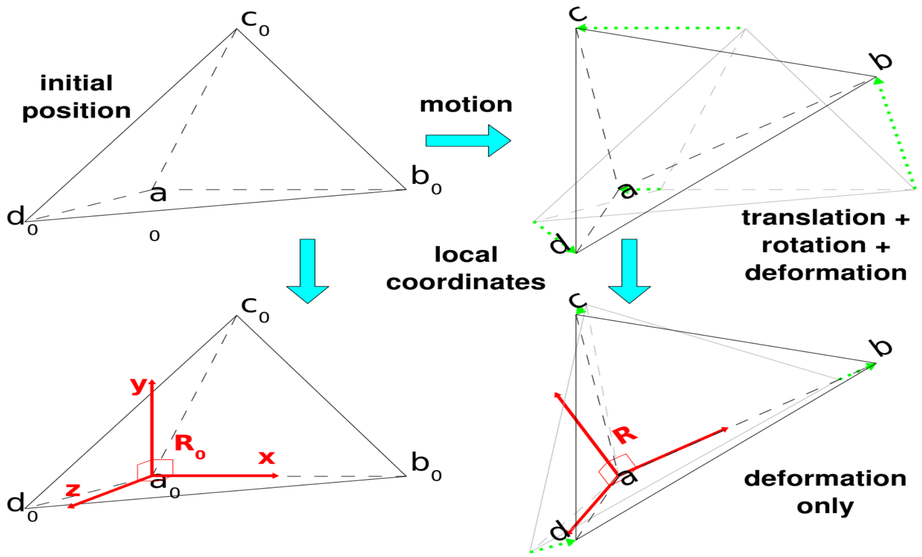
- Most of surgical procedures involve cutting
  - Topological modifications
  - Makes pre-computations difficult to use

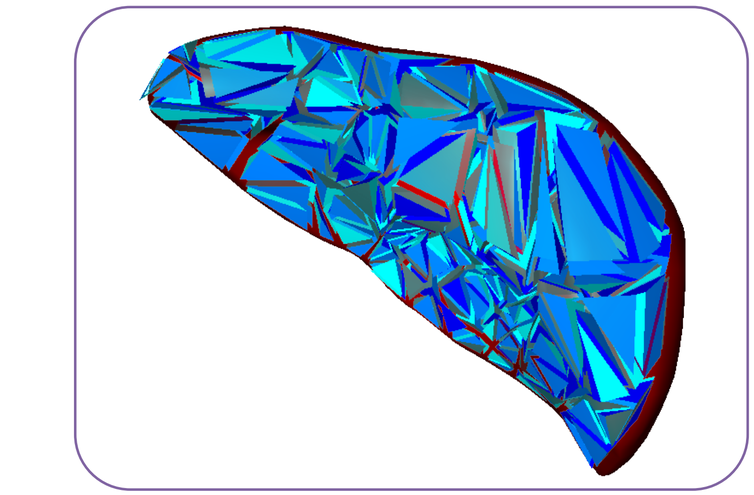


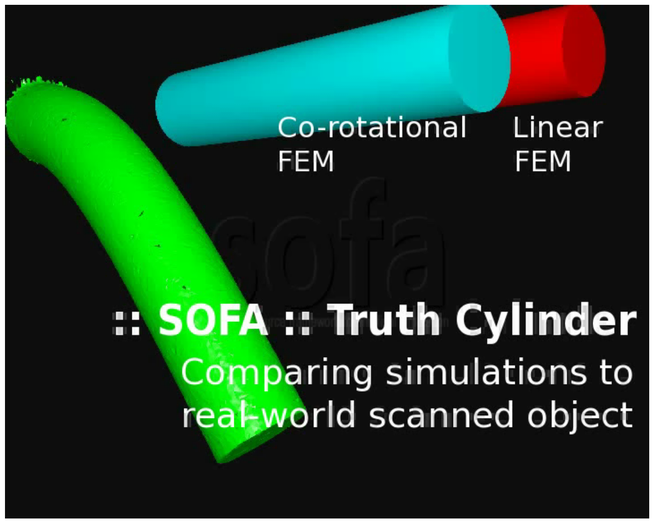
Equations formulation

- Continuum mechanics
  - Assumption of continuity of matter
  - Based on physics
  - Define a deformation model
- Use Finite Element Method
  - Discretization in small elements
  - Integrate and solve equations by elements
  - Approximation which depends on the size of elements
- We use the co-rotational formulation:
  - Linear material
  - Geometrical non linearities

$$K = \sum G_e * R_e * K_e * K_e^T * G_e^T$$



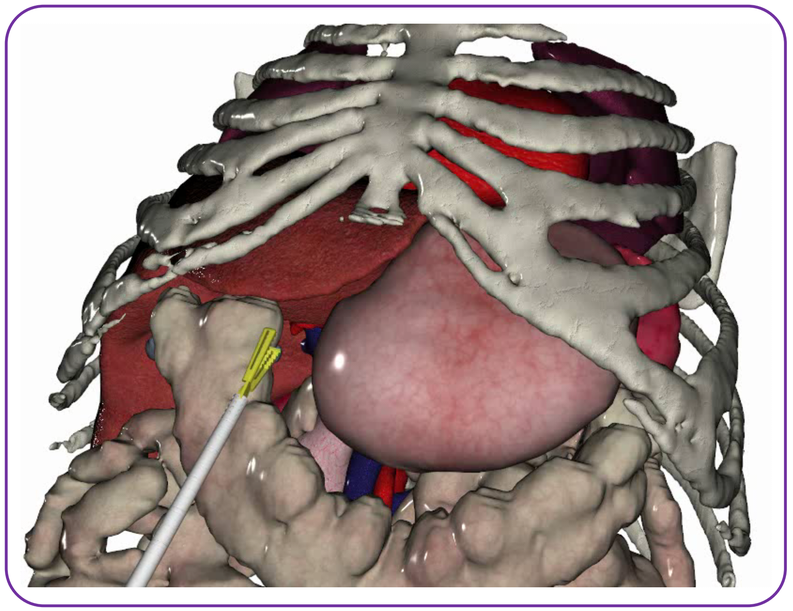




Temporal system evolution

- Dynamical system => Choose time integrator
- Explicit integration : TLED [Miller et al. 2006]
  - Local resolution, Fast and parallelizable
  - Enable topological modifications (cutting)
  - Stability implies to use small time steps

$$\begin{aligned} \boldsymbol{M}a &= \boldsymbol{f}(p_{t+h}, v_{t+h}) \\ v_{t+h} &= v_t + ha \\ p_{t+h} &= p_t + hv_{t+h} \end{aligned}$$

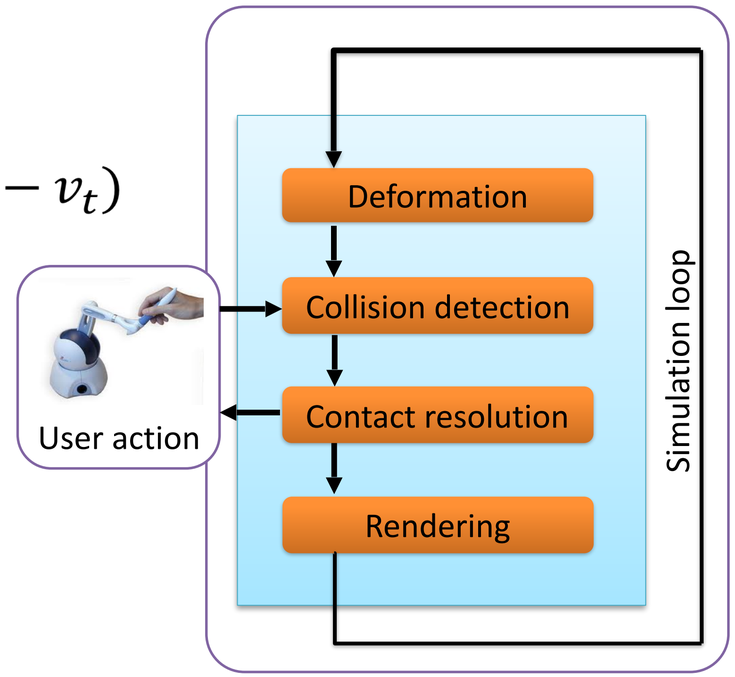


- Implicit integration :
  - Update the acceleration from position and velocity at the end of the time step
  - Stable with large time steps
    - Linearization of f around the initial position [Baraff et al. 1998]

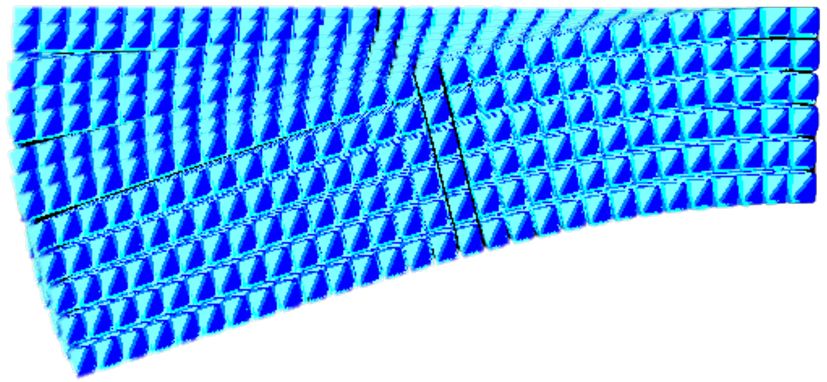
$$\boldsymbol{f}(p_{t+h}, v_{t+h}) = f(p_t, v_t) + K \cdot (p_{t+h} - p_t) + B \cdot (v_{t+h} - v_t)$$

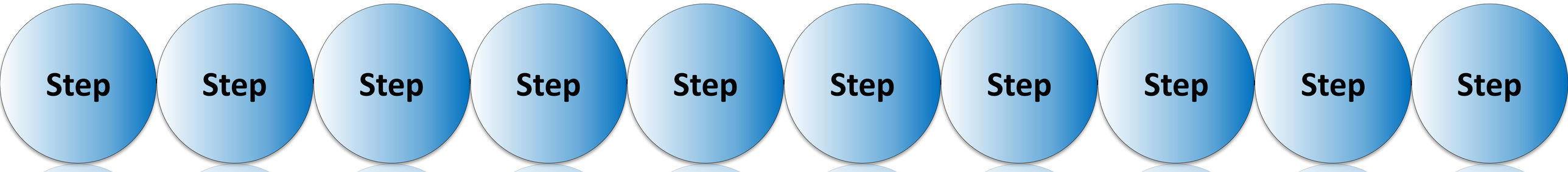
    - Rearranging equations gives the linear system :

$$\underbrace{(M - hB - h^2K)}_{A} \quad \underbrace{dv}_{x} = \underbrace{hf(p_t, v_t) + h^2Kv_t}_{b}$$



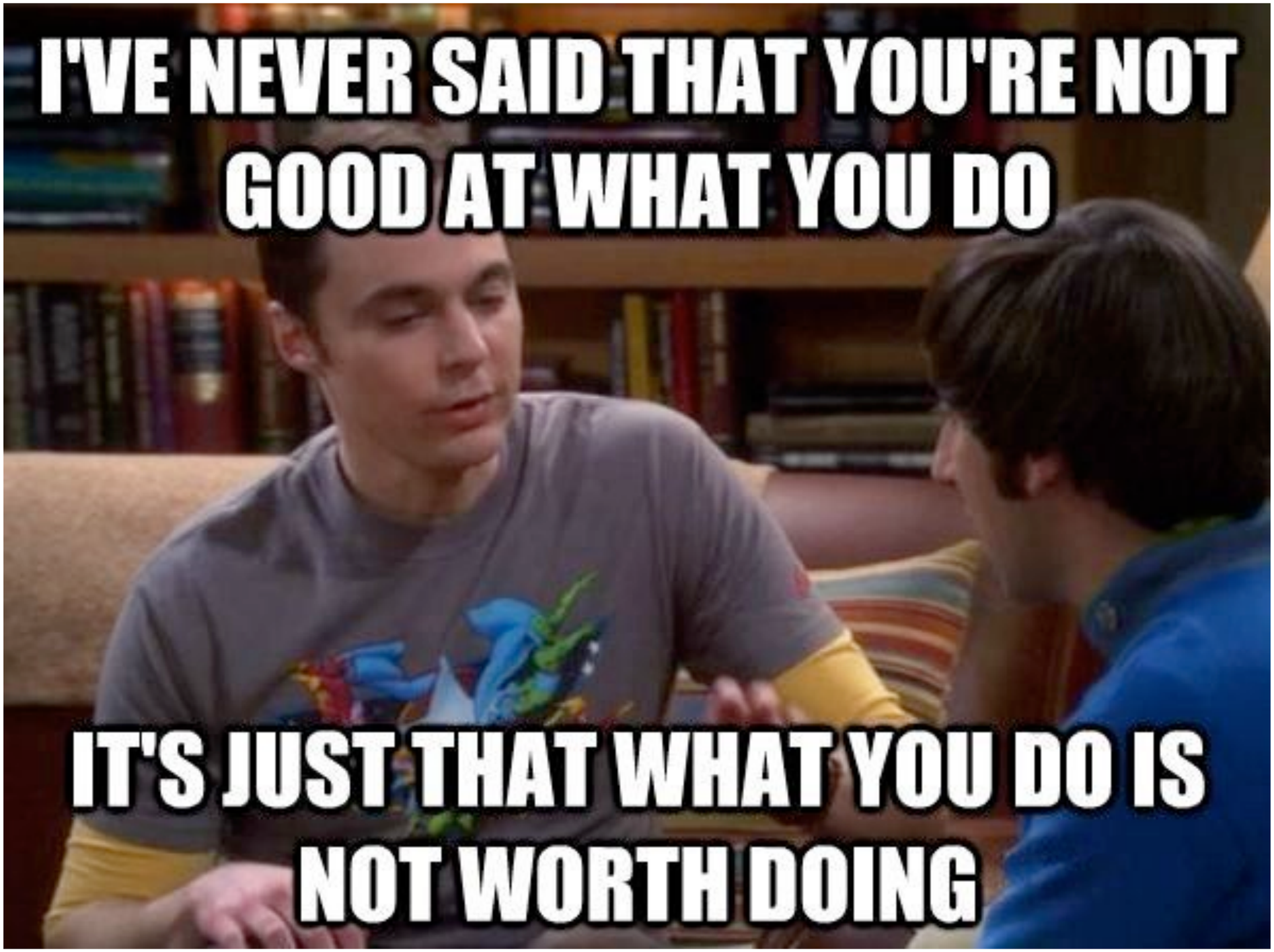
Temporal system evolution





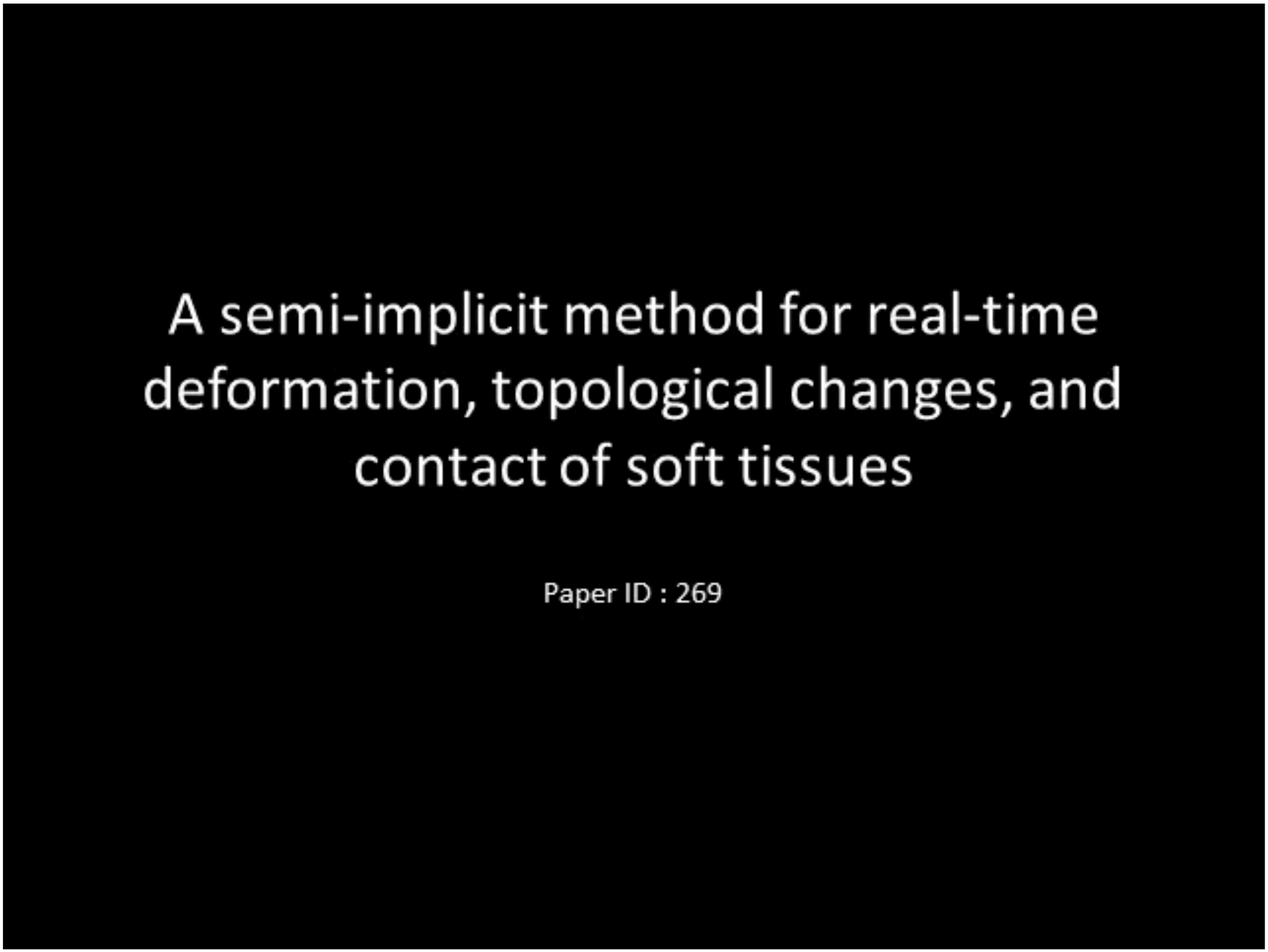
Due to nonlinearities, we need to solve a linear system $Ax = b$ at each time step

- Problem : How to solve $A\ x = b$ in real time ?

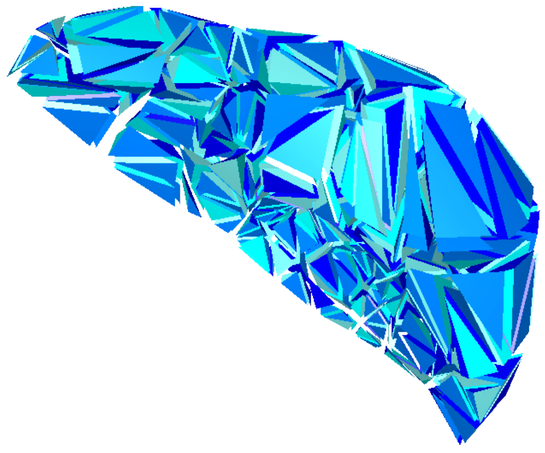

I'VE NEVER SAID THAT YOU'RE NOT GOOD AT WHAT YOU DO
IT'S JUST THAT WHAT YOU DO IS NOT WORTH DOING

Applications, Limitations, Open Questions

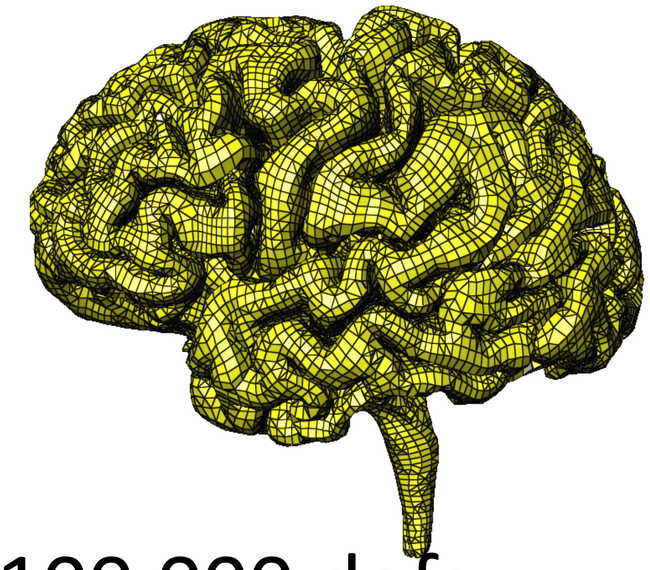
Conclusions



Real-time implicit simulation of cutting in heterogeneous soft tissues *Courtecuisse et al., Medical image analysis 2014*

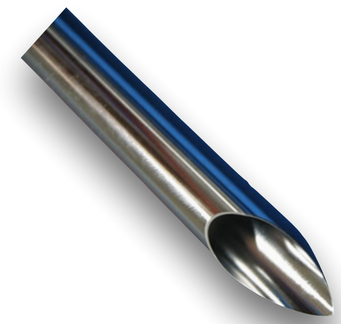


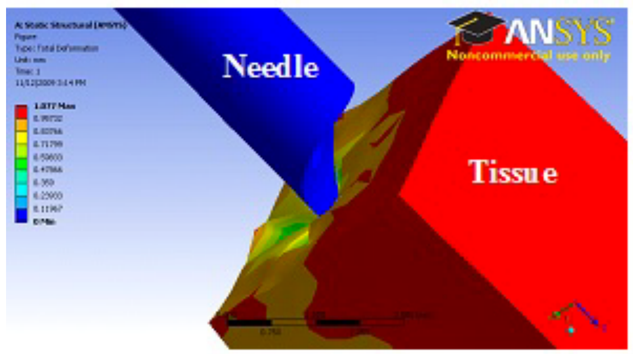
x100

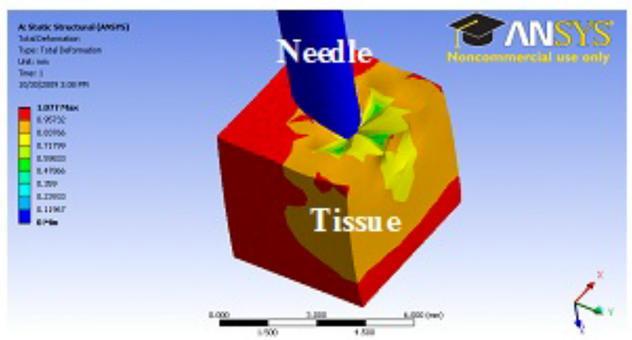


a few 1,000 dofs

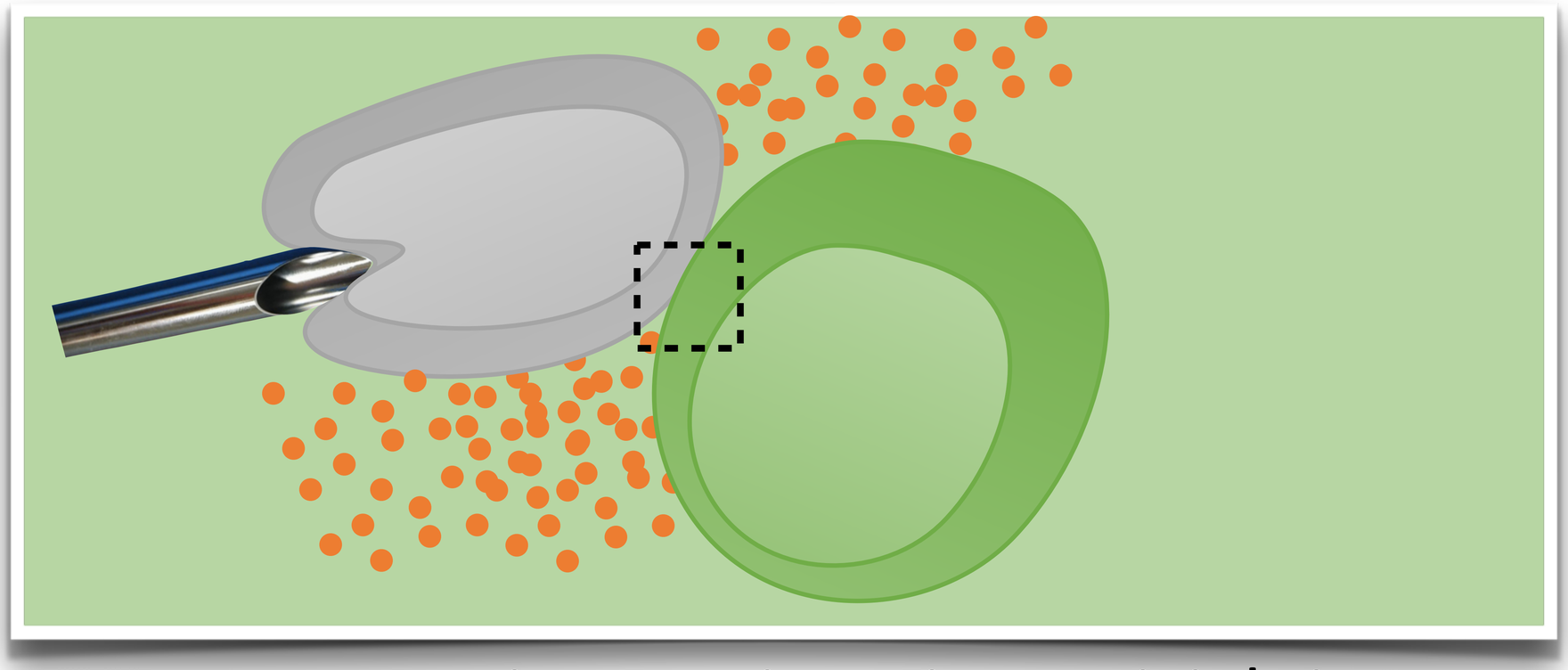
a few 100,000 dofs



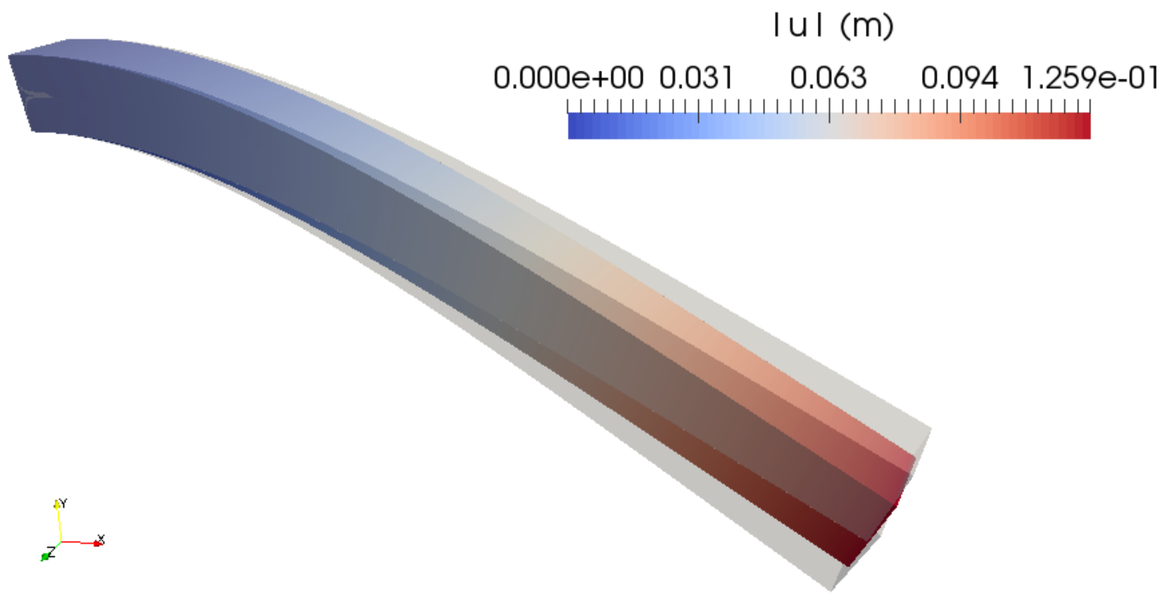




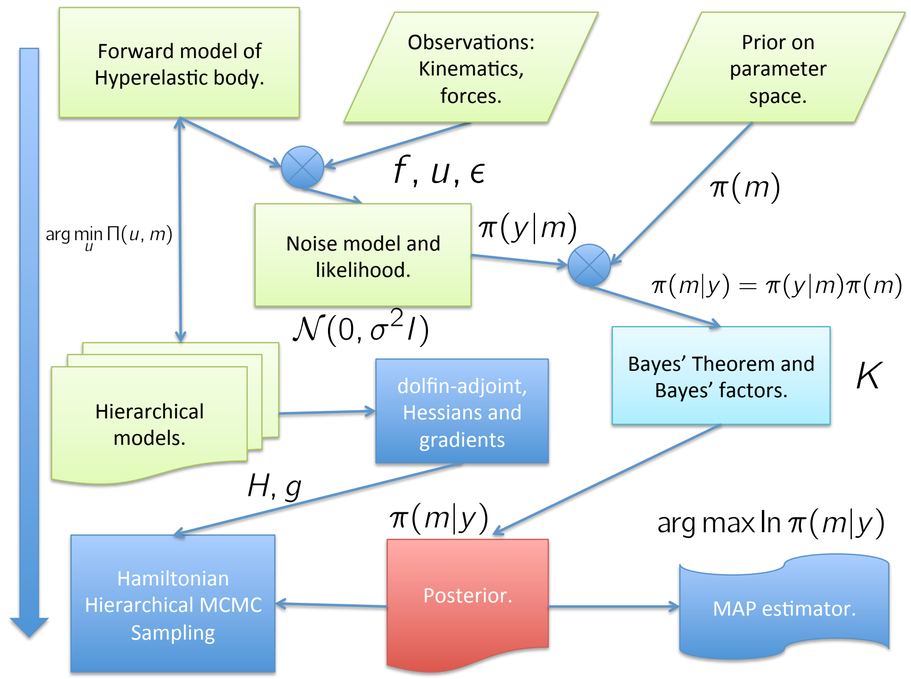
tool-tissue interaction models?



organ-organ interaction - intersticial tissue



uncertainties in geometry and material



model selection

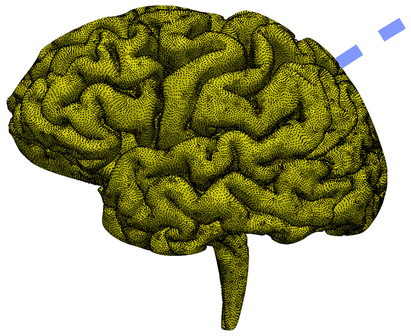
Fast patient specific solutions

Three-pronged approach

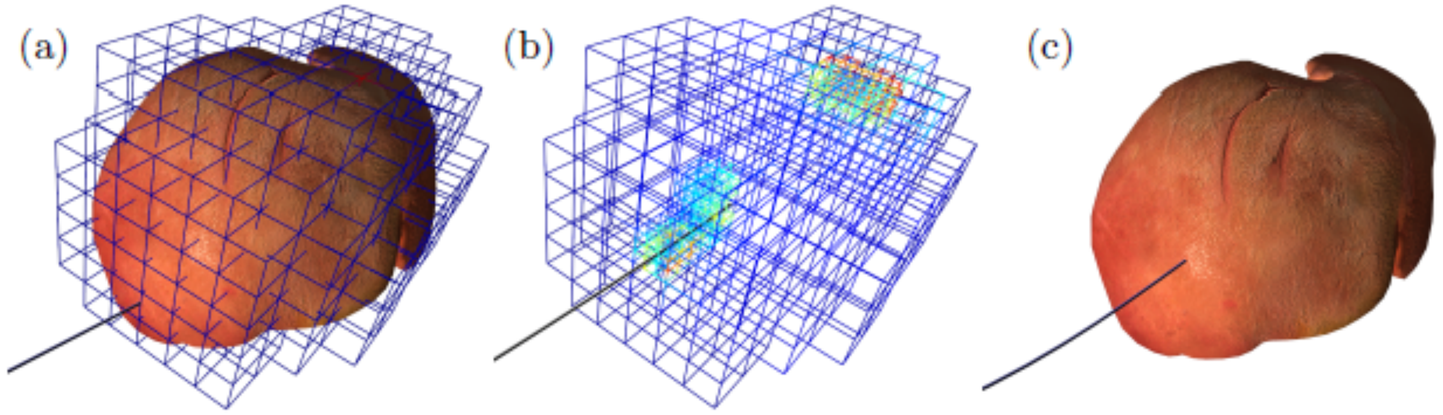
- Patient specific discretisation of the organs
- Model order reduction
- Error estimation and adaptivity

Fast patient specific solutions

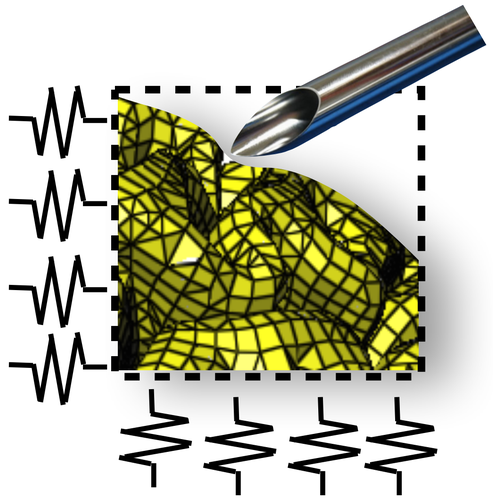
Three-pronged approach



Patient-specific MODEL



Error estimation and adaptivity

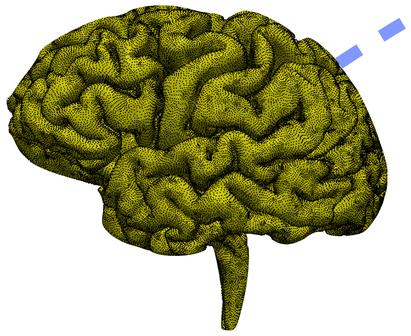


Spring equivalent to the Offline Lego Blocks

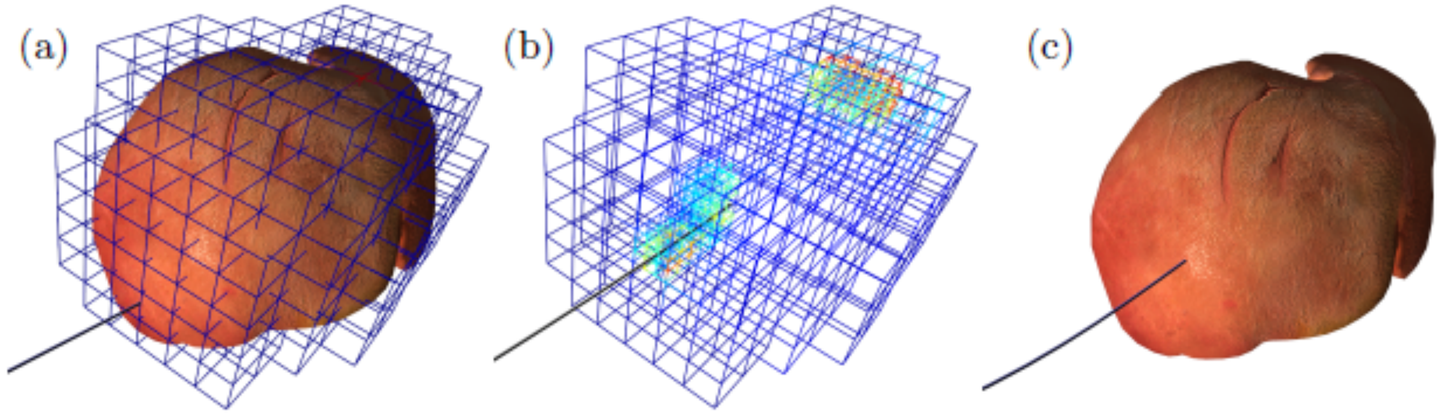
Model order reduction

Fast patient specific solutions

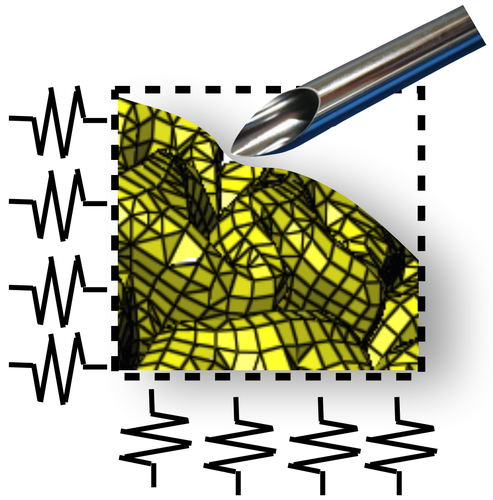
Three-pronged approach



Patient-specific MODEL



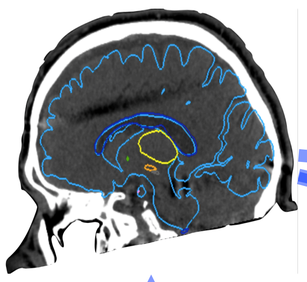
Error estimation and adaptivity



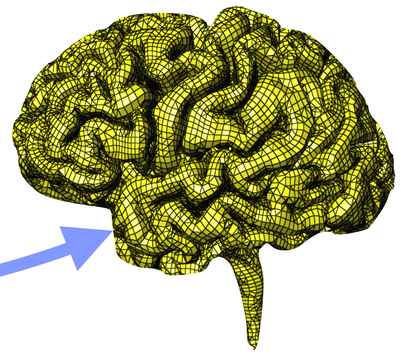
Spring equivalent to the Offline Lego Blocks

Model order reduction

Segmented image

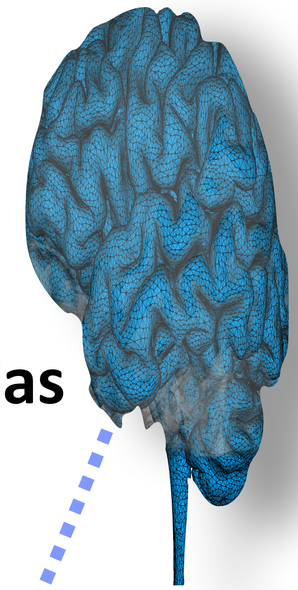


Polyhedral meshing virtual elements (Lévy)



Meshless collocation (Bordas, Bourantas)

Atlas



Scanner

Image as a model (Miller)

Segmentation-free meshing (Payan, Bucki)

Diffeomorphic maps (Payan)

Questions

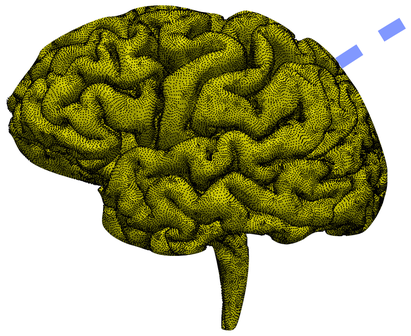
- patient-specificity?
- generation speed?
- automation?
- easy adaptivity?
- incompressibility?
- mesh distortion?

Patient-specific MODEL

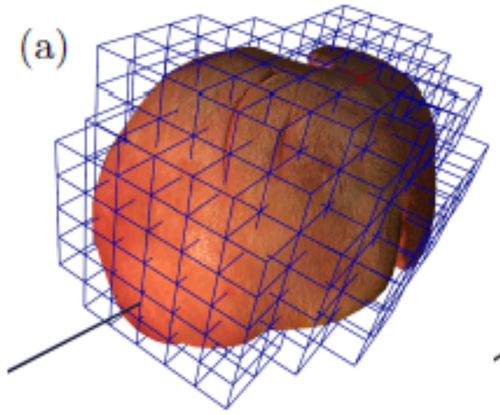
with Bruno Lévy, Yohan Payan, Karol Miller, Marek Bucki

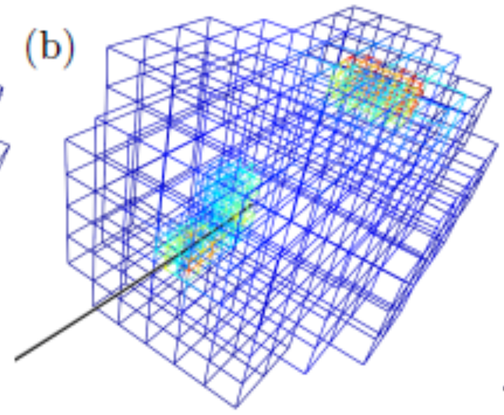
Fast patient specific solutions

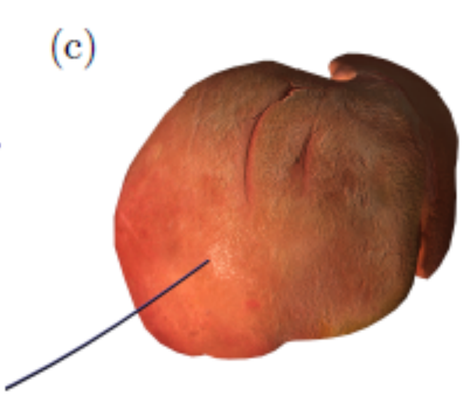
Three-pronged approach



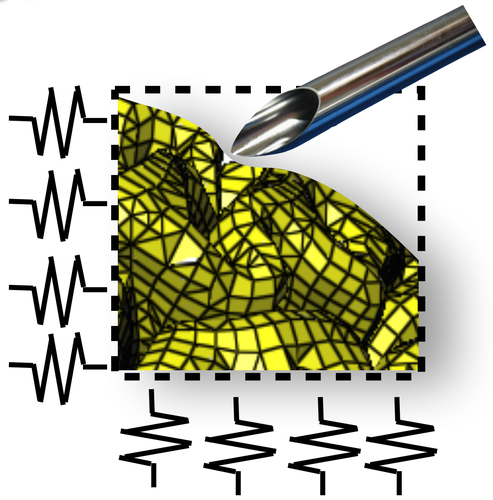
Patient-specific MODEL





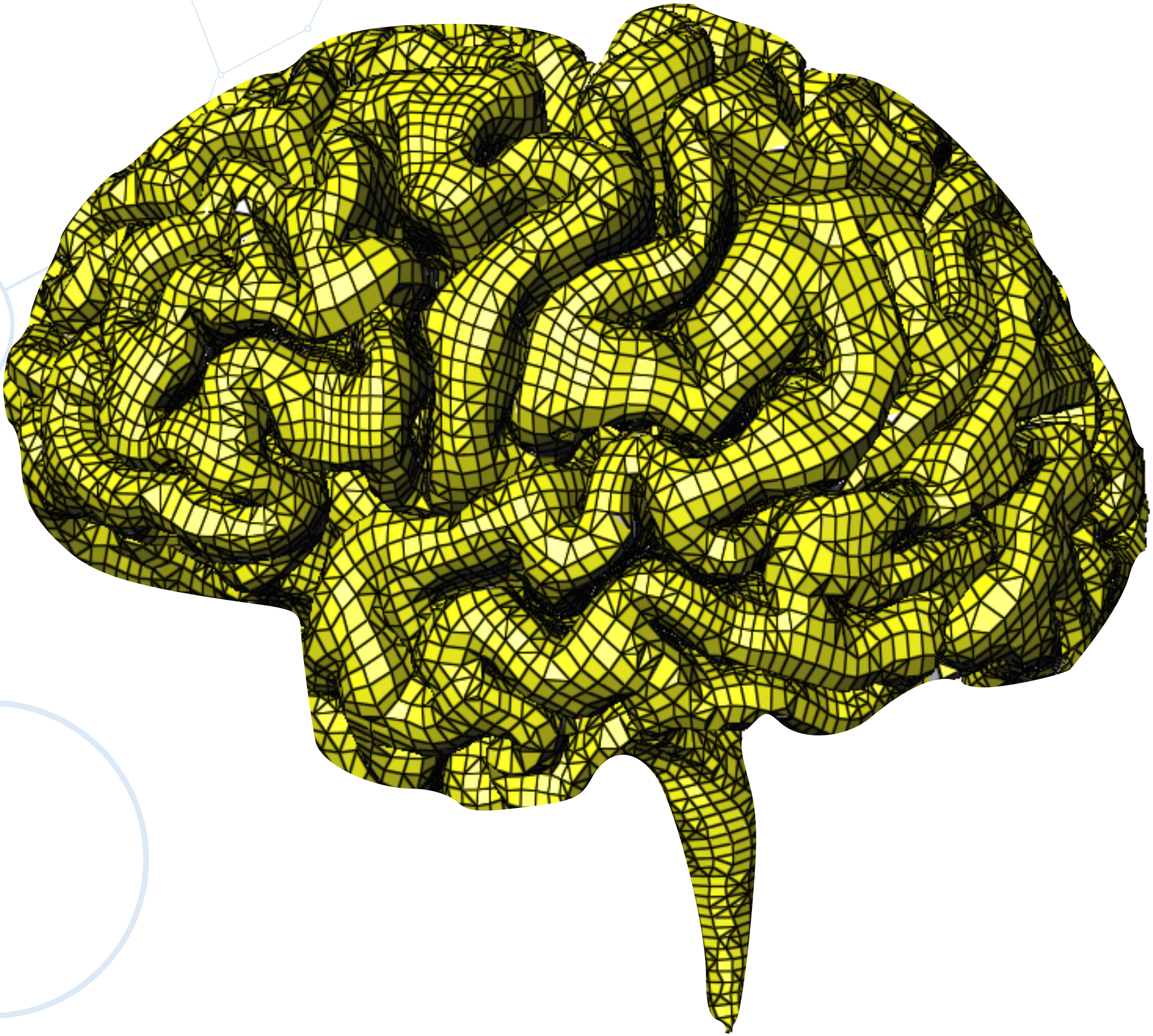


Error estimation and adaptivity

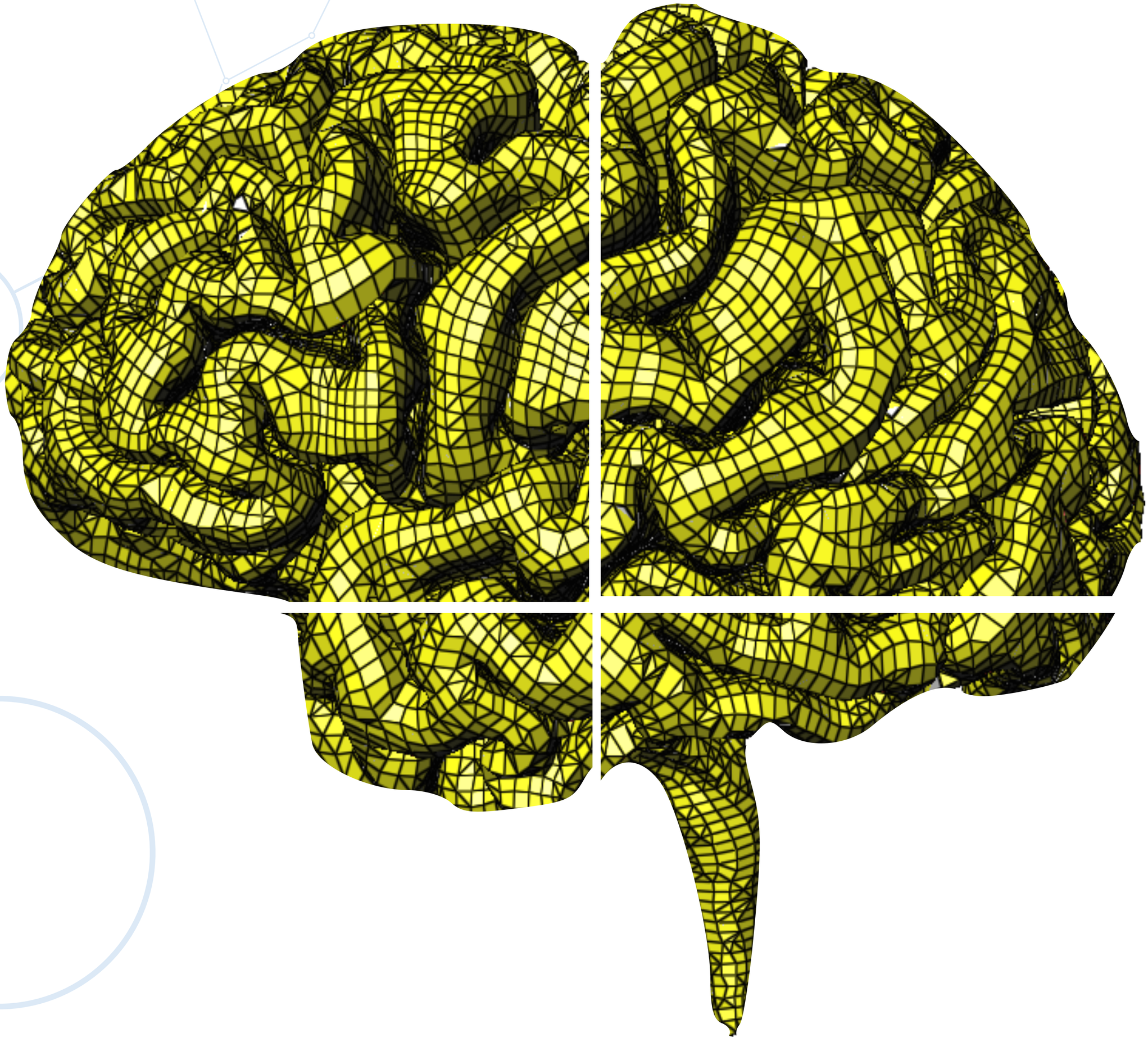


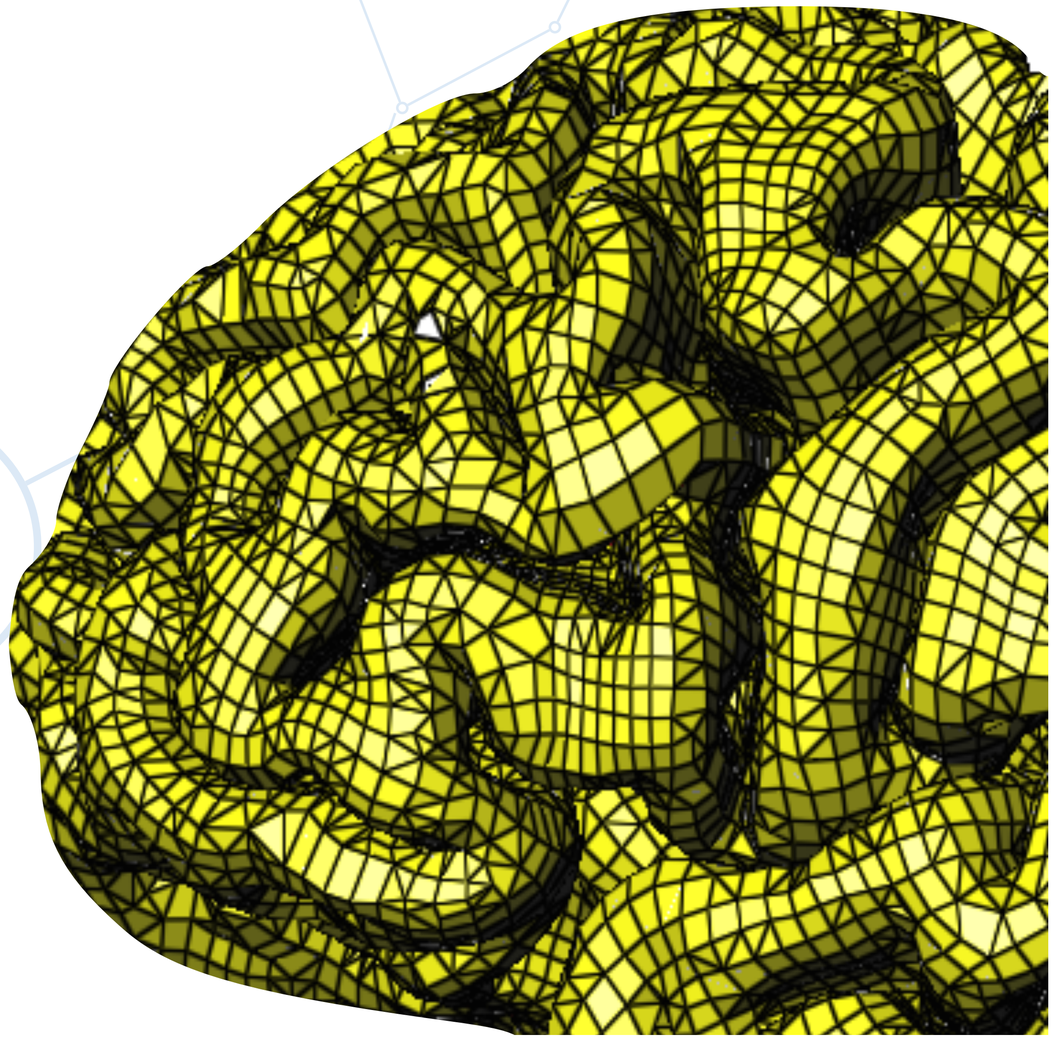
Spring equivalent to the Offline Lego Blocks

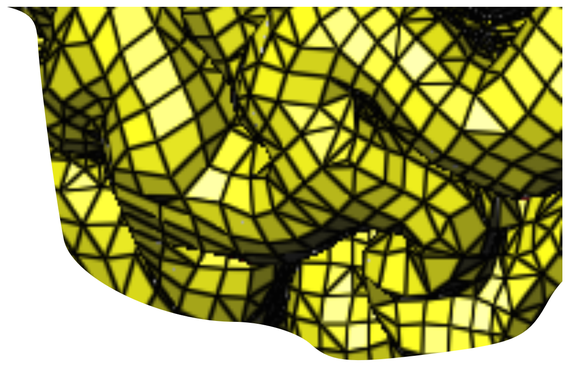
Model order reduction

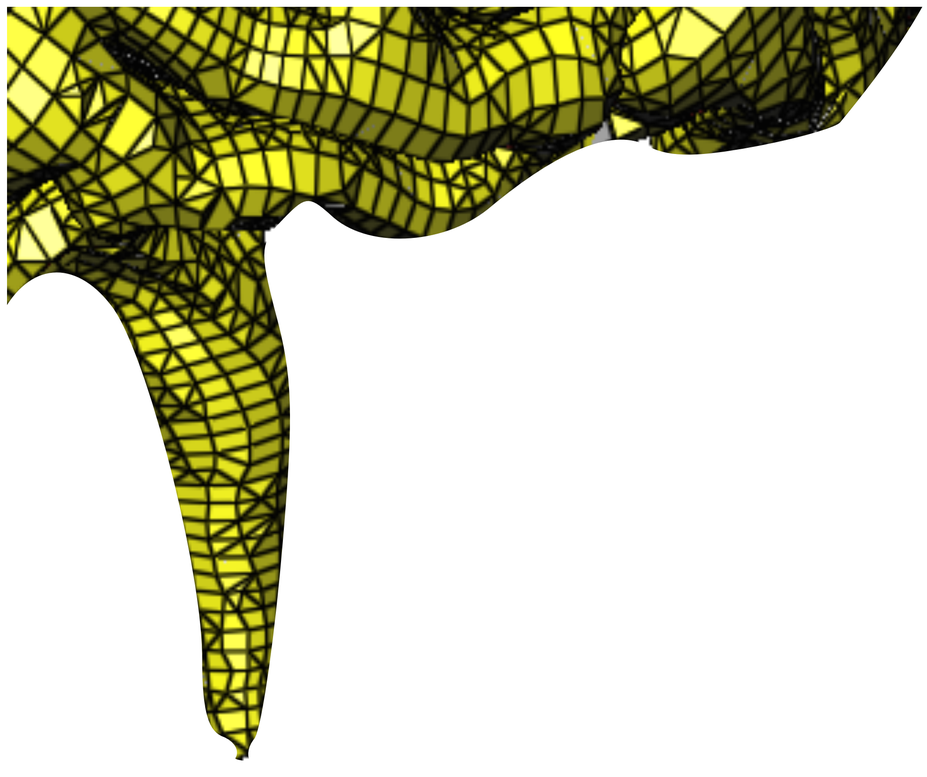


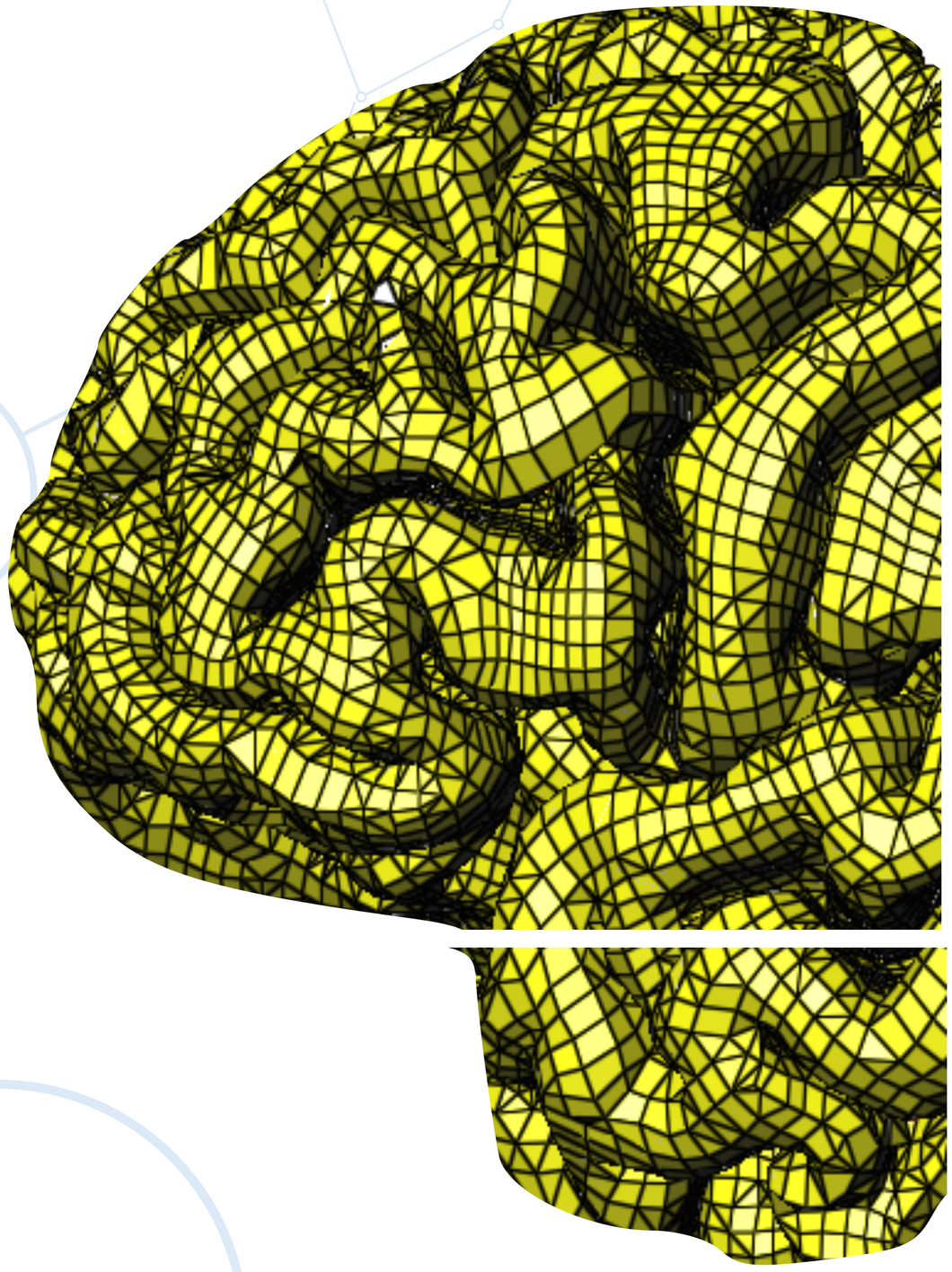

"What computer do you have?
And please don't say a white one."
BACONWRAPPEDMEDIA.COM

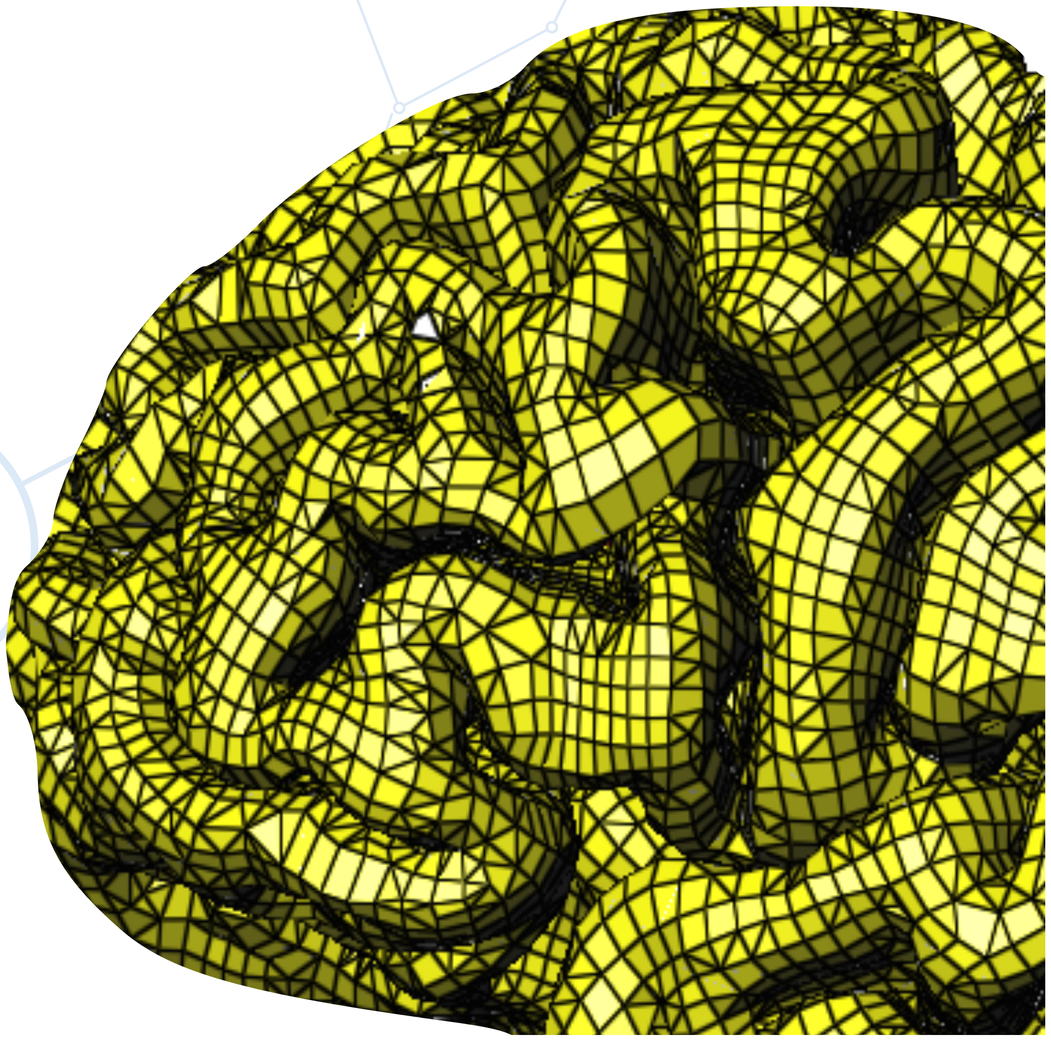


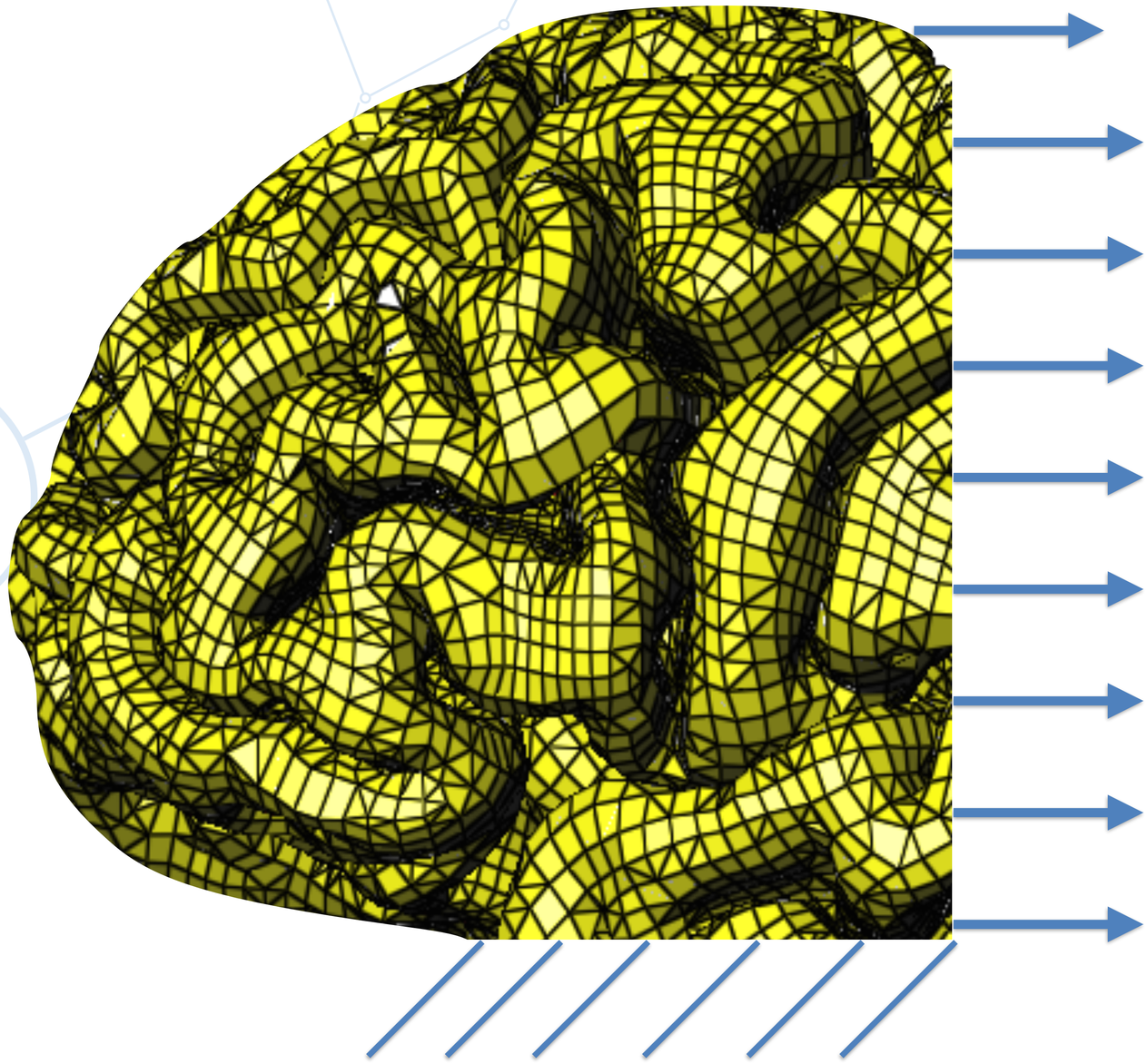






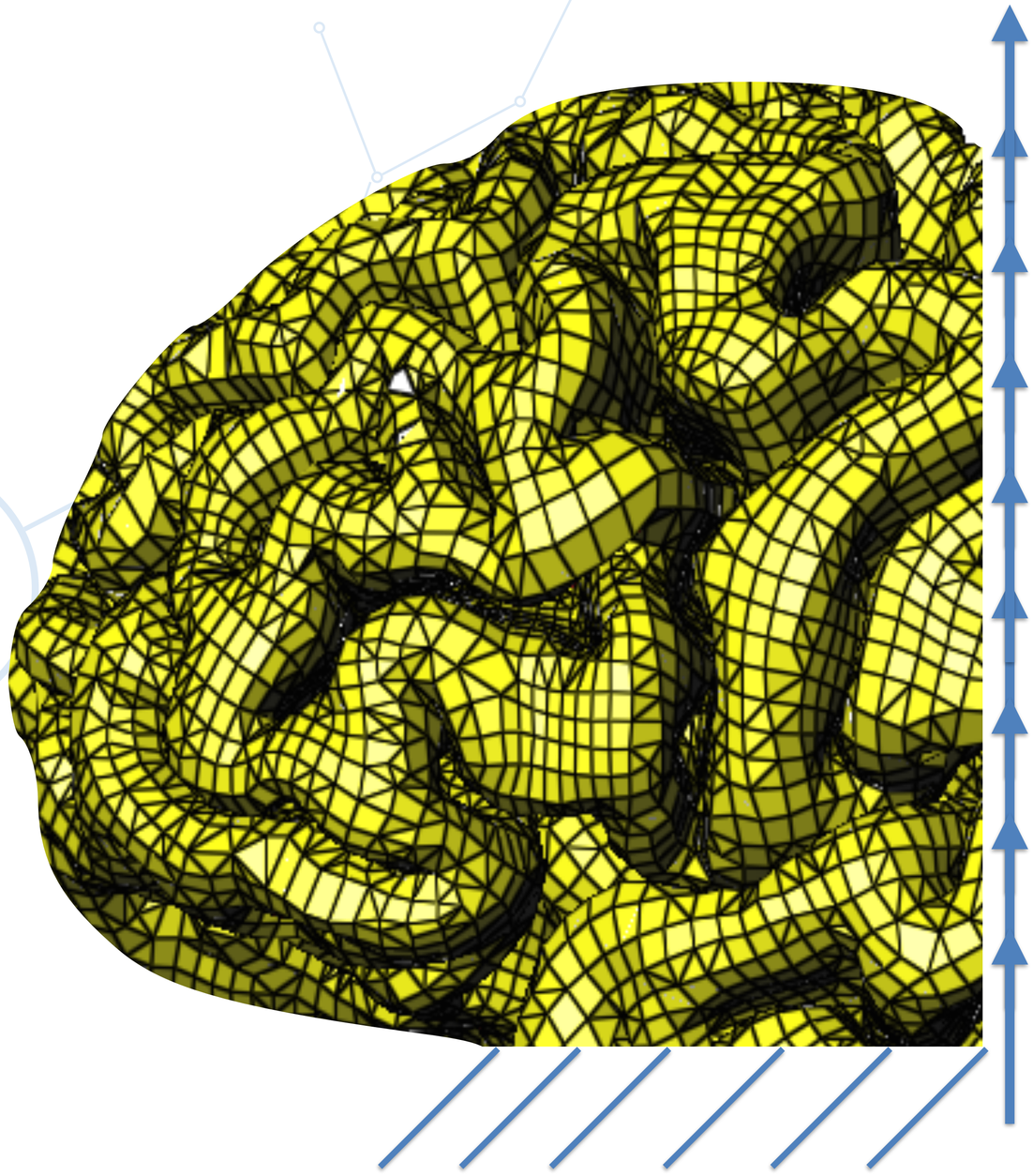






Deformation mode 1

Store in a vector U1

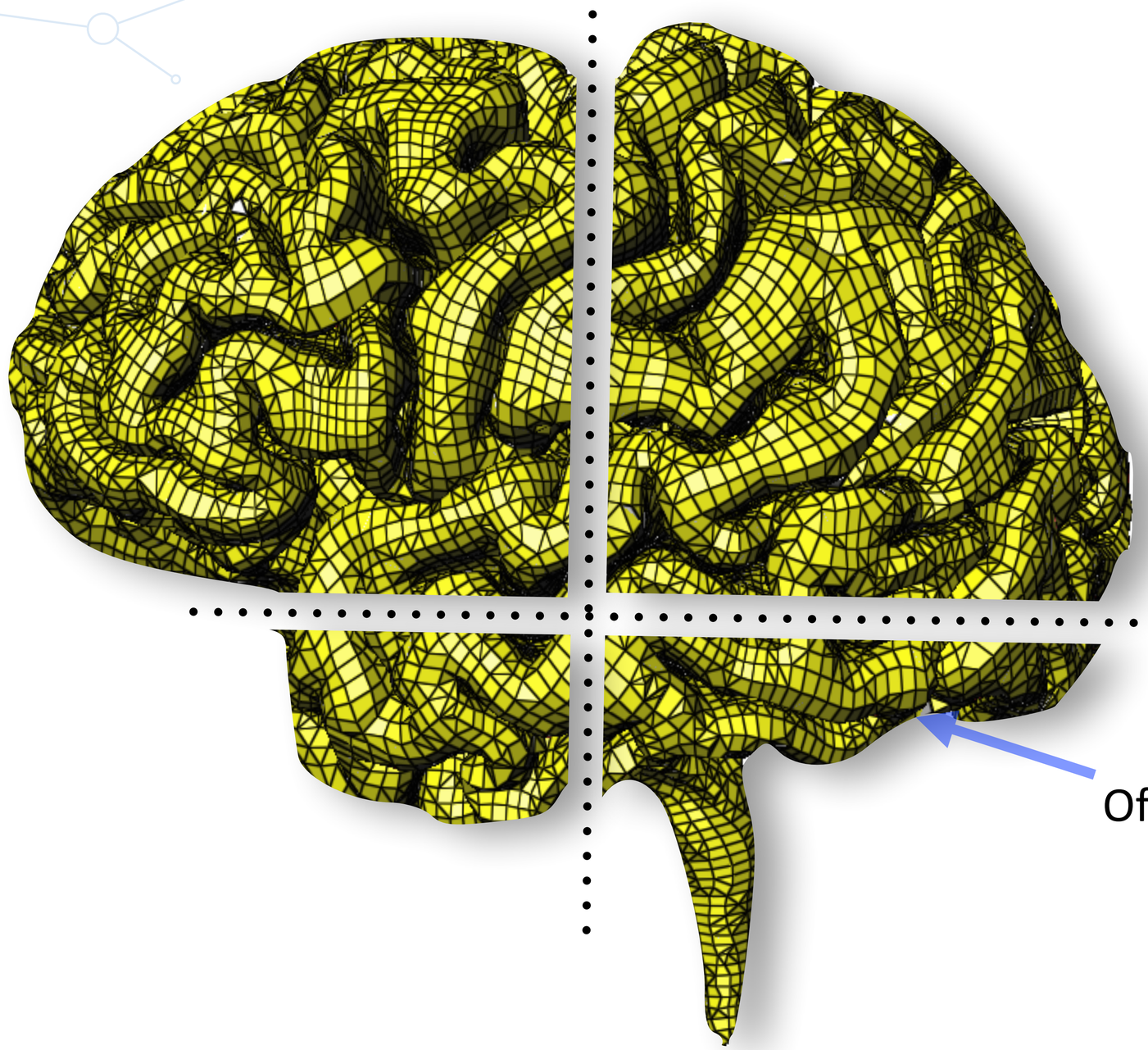


Deformation mode 2

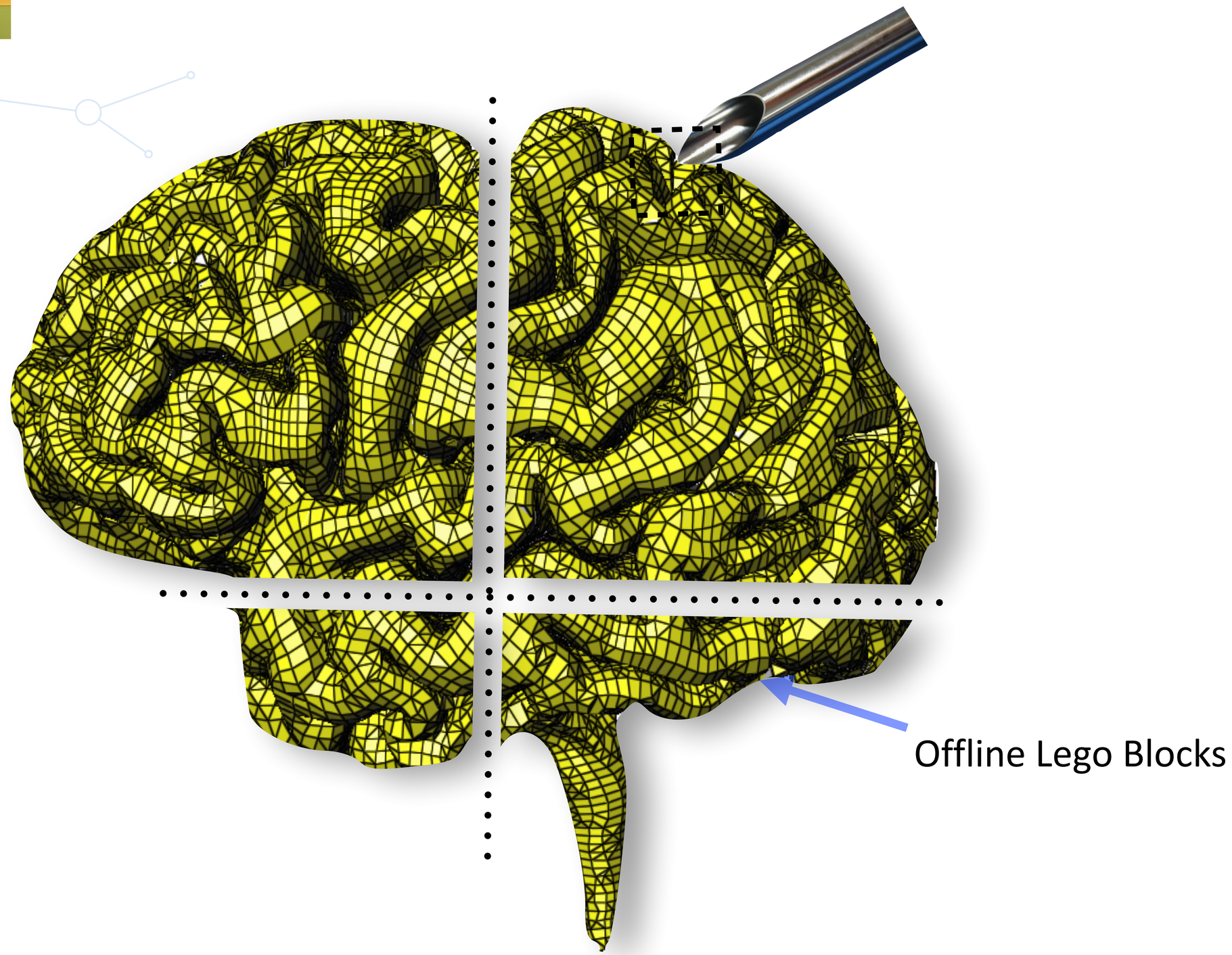
Store in a vector U2

Assemble the “Lego” blocks

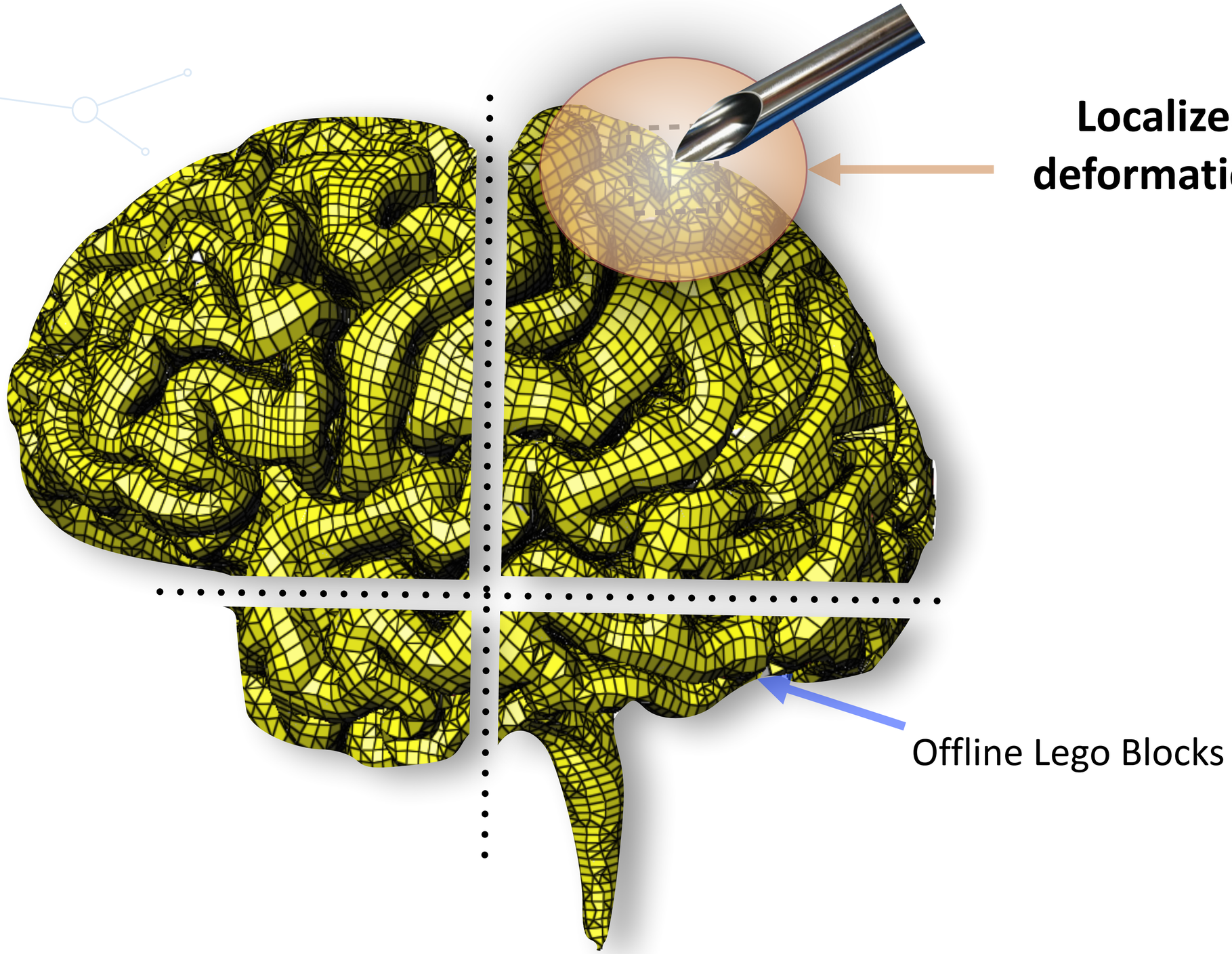
Assemble the “Lego” blocks



Insert needle/scalpel



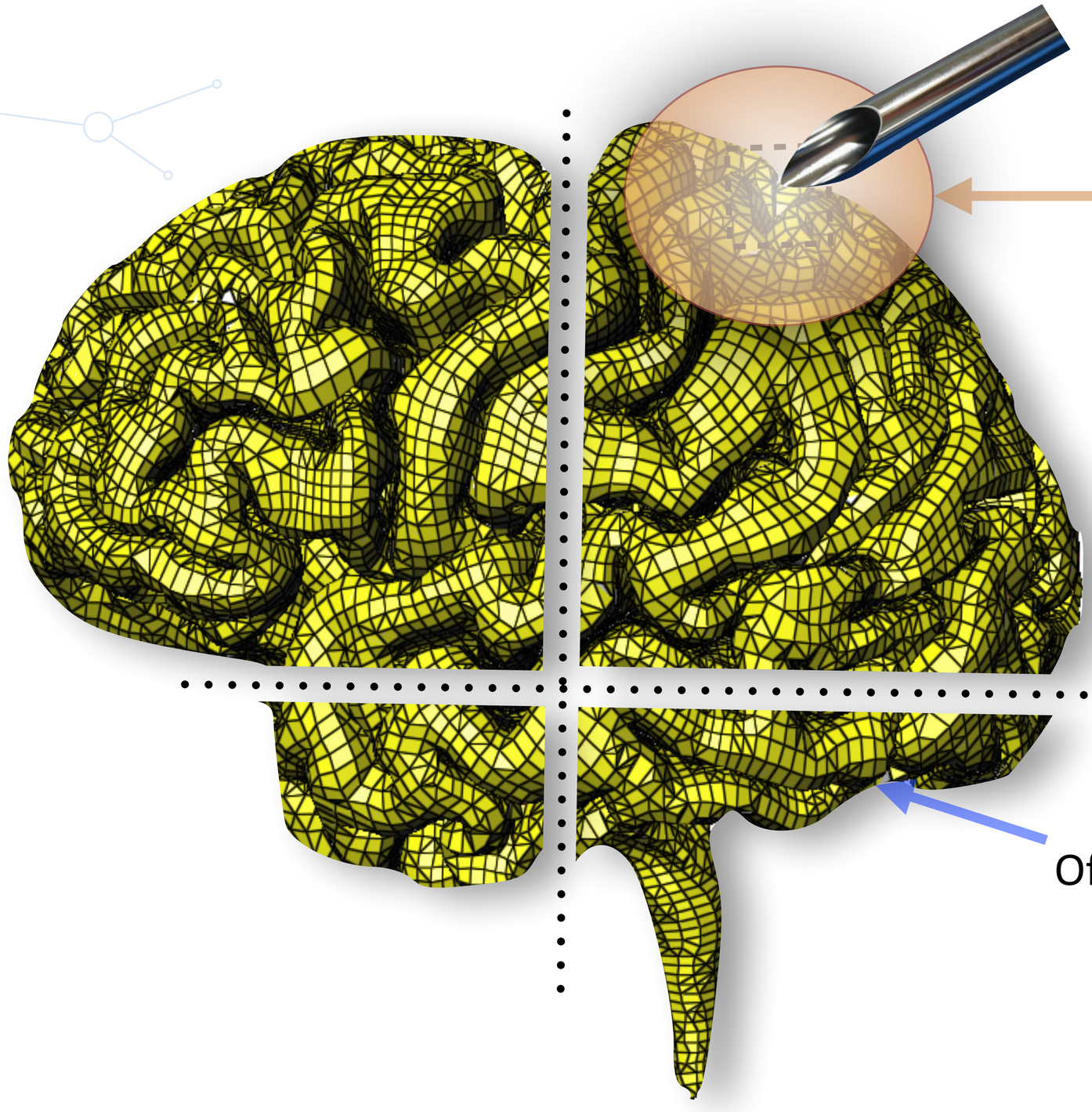
Insert needle/scalpel



Localized deformations

Offline Lego Blocks

Insert needle/scalpel

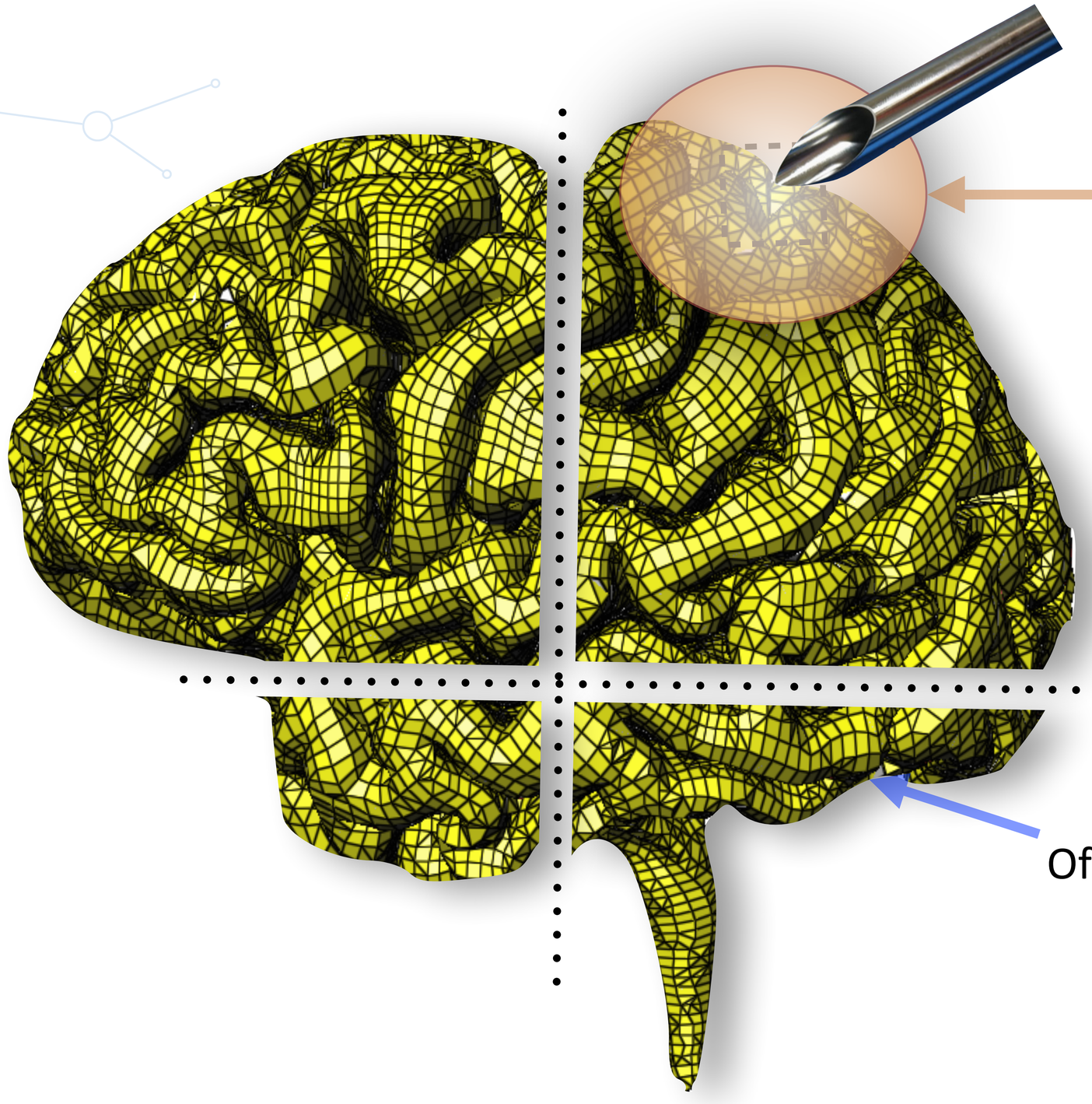


Localized deformations

Zone is not reducible

Offline Lego Blocks

Insert needle/scalpel



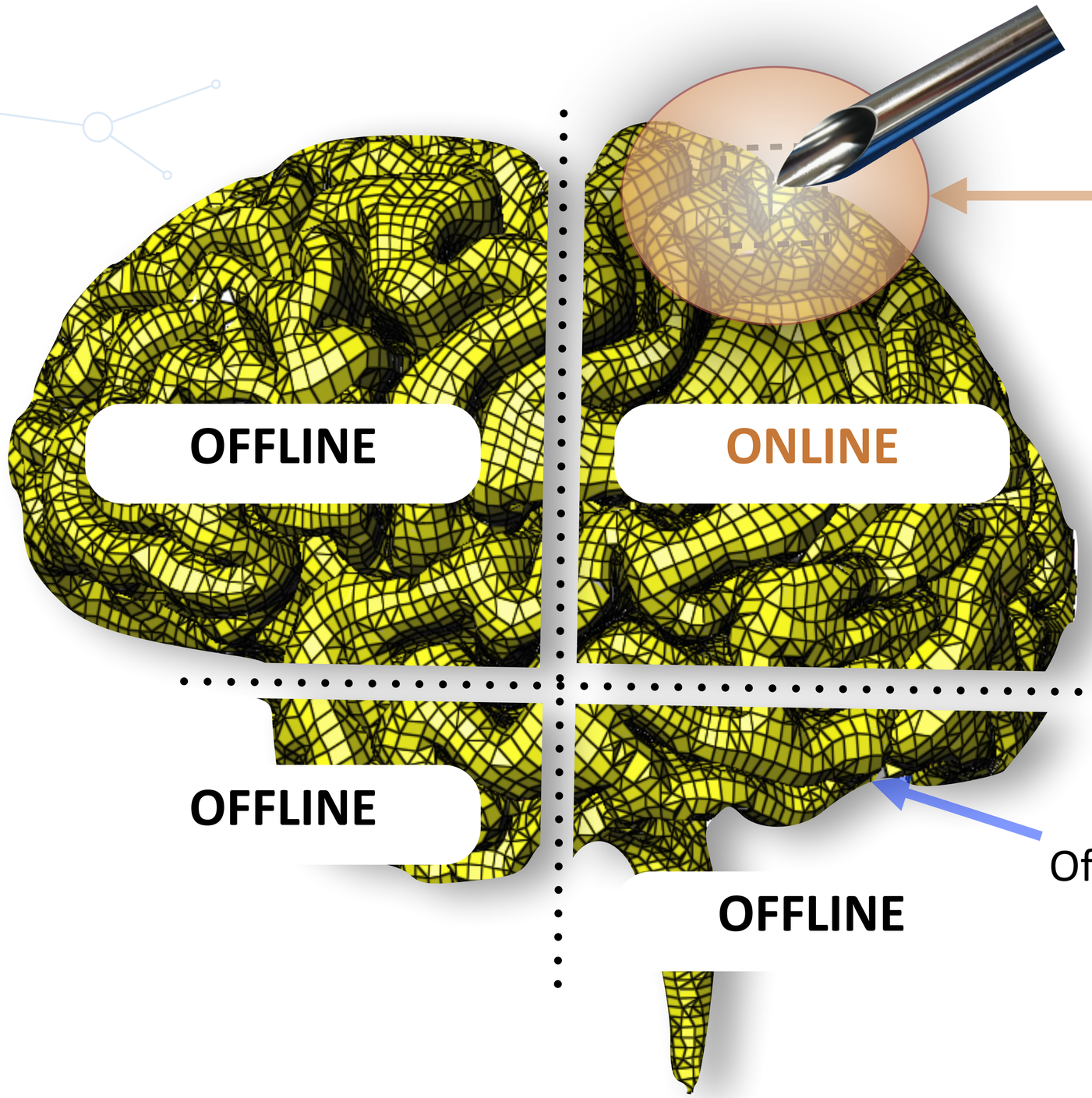
Localized deformations

Zone is not reducible

Full, direct simulation

Offline Lego Blocks

Insert needle/scalpel



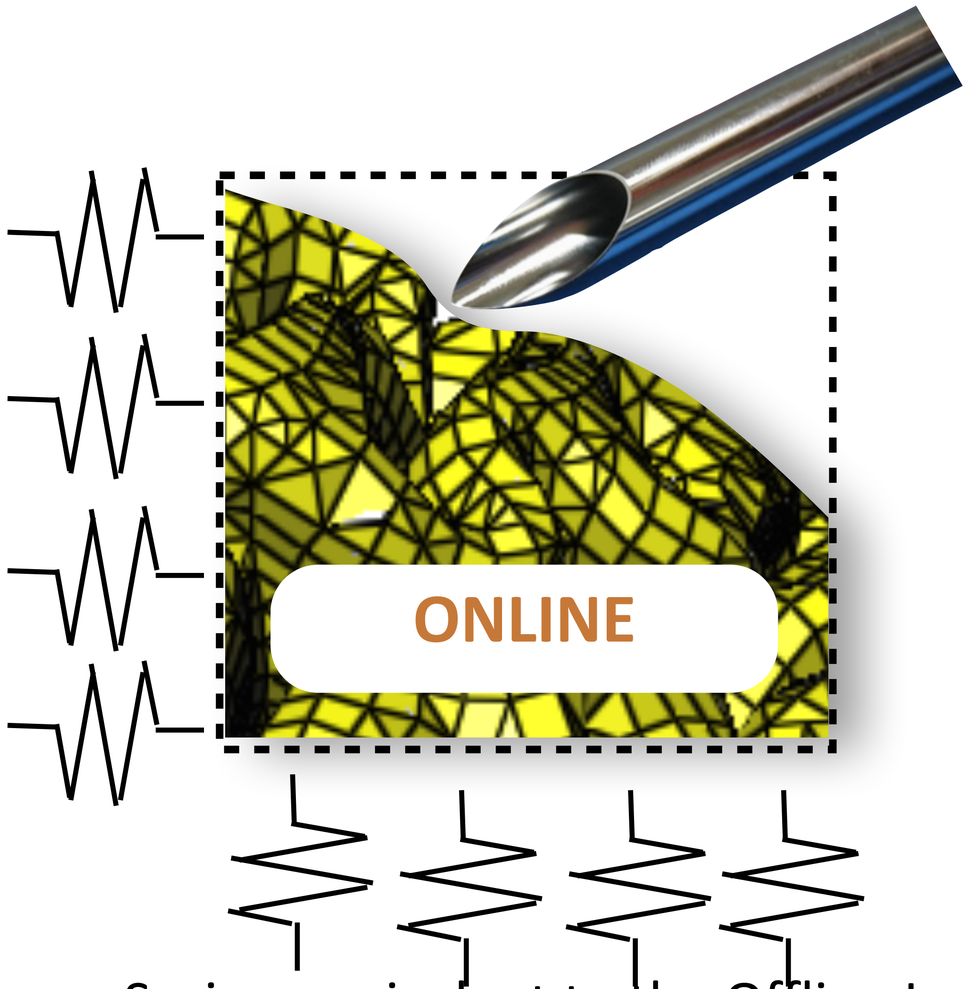
Localized deformations

Zone is not reducible

Full, direct simulation

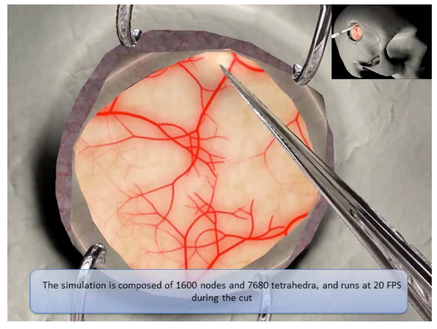
Offline Lego Blocks

Superimpose OFFLINE with ONLINE Lego Blocks in non-linear regions



Spring equivalent to the Offline Lego Blocks

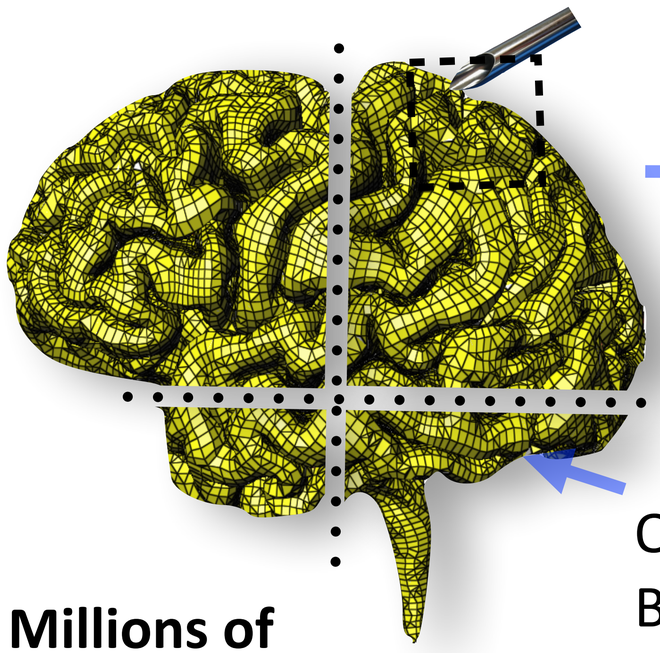
RealTCut
real-time
for ~5,000
unknowns



Fast patient specific solutions

Model order reduction

Precompute the OFFLINE Lego Blocks

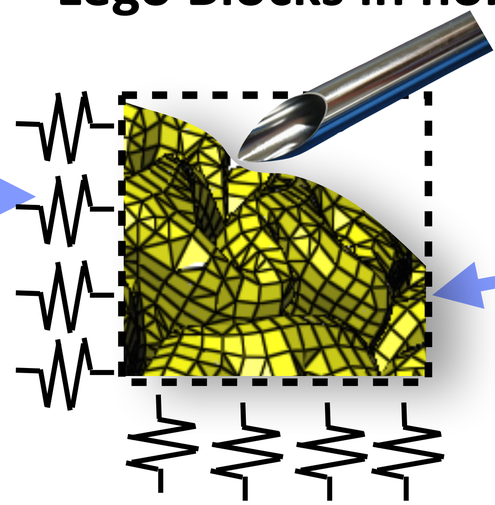


Proper orthogonal decomposition modes for each subdomain
10’s of unknowns

Offline Lego Blocks

Millions of unknowns

Superimpose OFFLINE with ONLINE Lego Blocks in non-linear regions



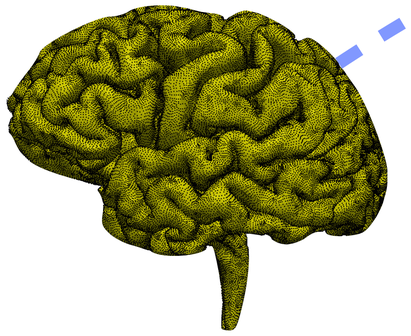
Online Lego Block **RealTCut** real-time for ~5,000 unknowns

Spring equivalent to the Offline Lego Blocks

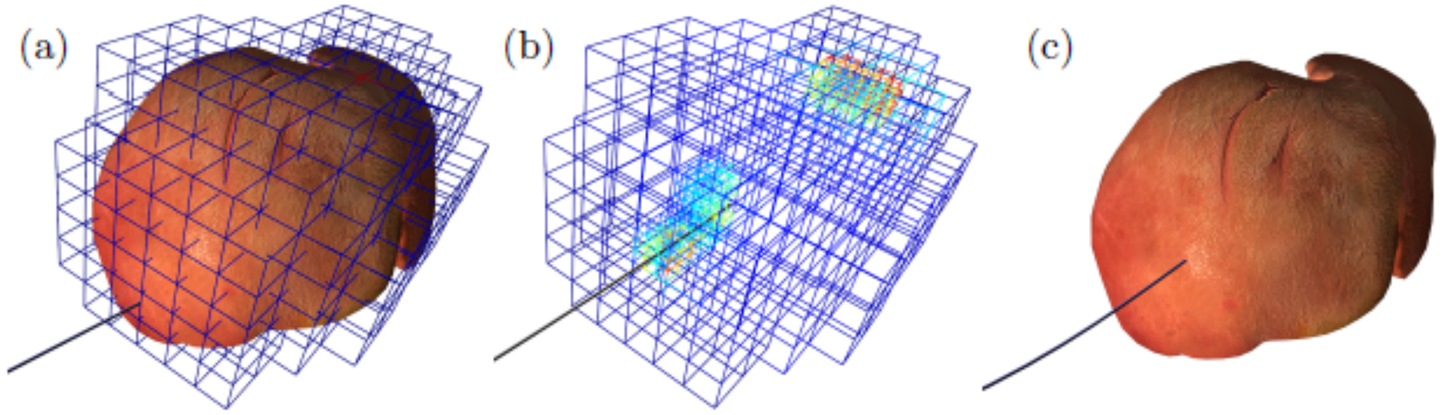
1000’s of unknowns

Fast patient specific solutions

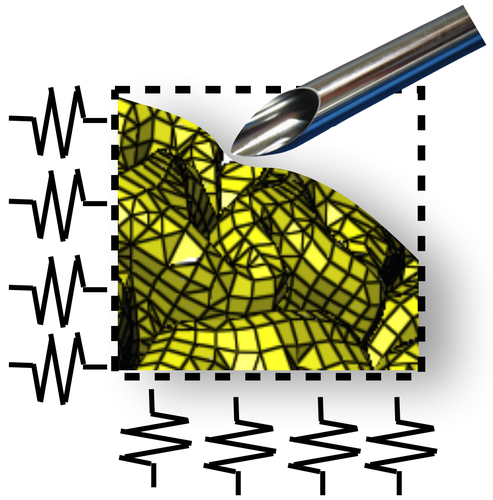
Three-pronged approach



Patient-specific MODEL



Error estimation and adaptivity



Spring equivalent to the Offline Lego Blocks

Model order reduction

Sources of error

physical problem

mathematical model

discretisation

numerical solution

model error

Model reduction error I

discretisation error

Model reduction error II

numerical error

Can we solve the problem fast enough?

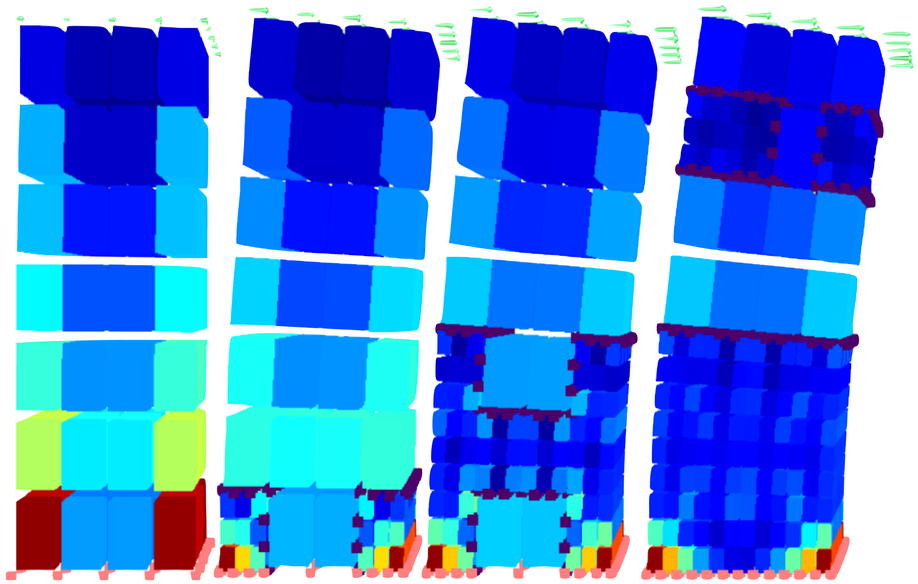
total error

validation: are we solving the **right** problem?

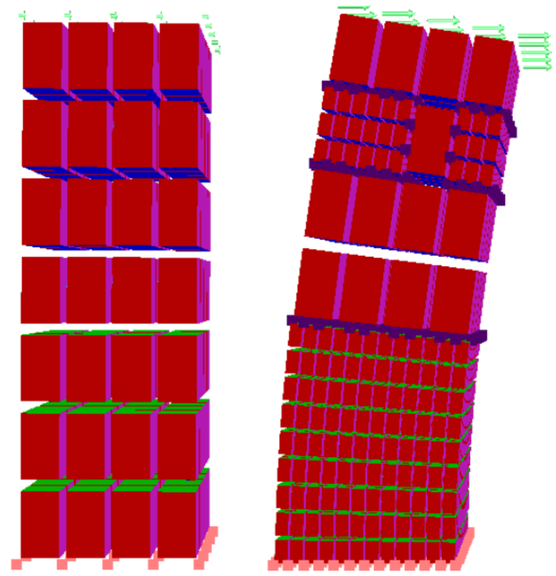
verification: are we solving the problem **right?**

Localise expenses where needed

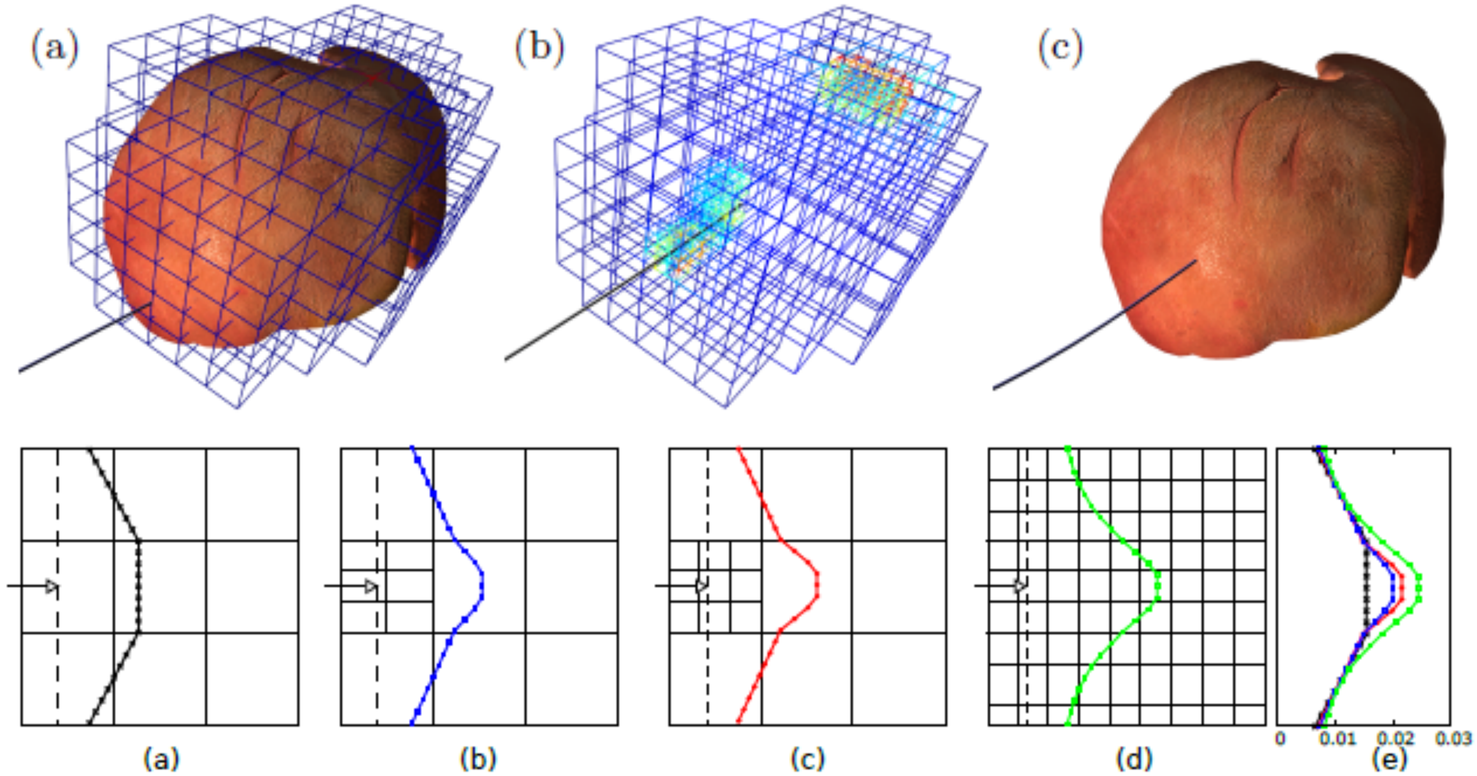
Error estimation and adaptivity



error maps Zienkiewicz-Zhu errors



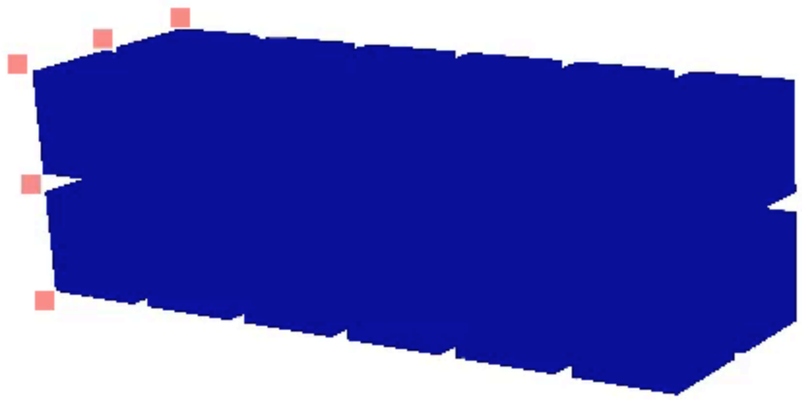
adapted oct-trees



adaptivity for real time needle insertion

with Bui Huu Phuoc, Hadrien Courtecuisse, Satyendra Tomar, Franz Chouly and Stéphane Cotin Franz Chouly

Real-time bending and twisting of a beam



with Bui Huu Phuoc, Hadrien Courtecuisse, Satyendra Tomar, Franz Chouly and Stéphane Cotin

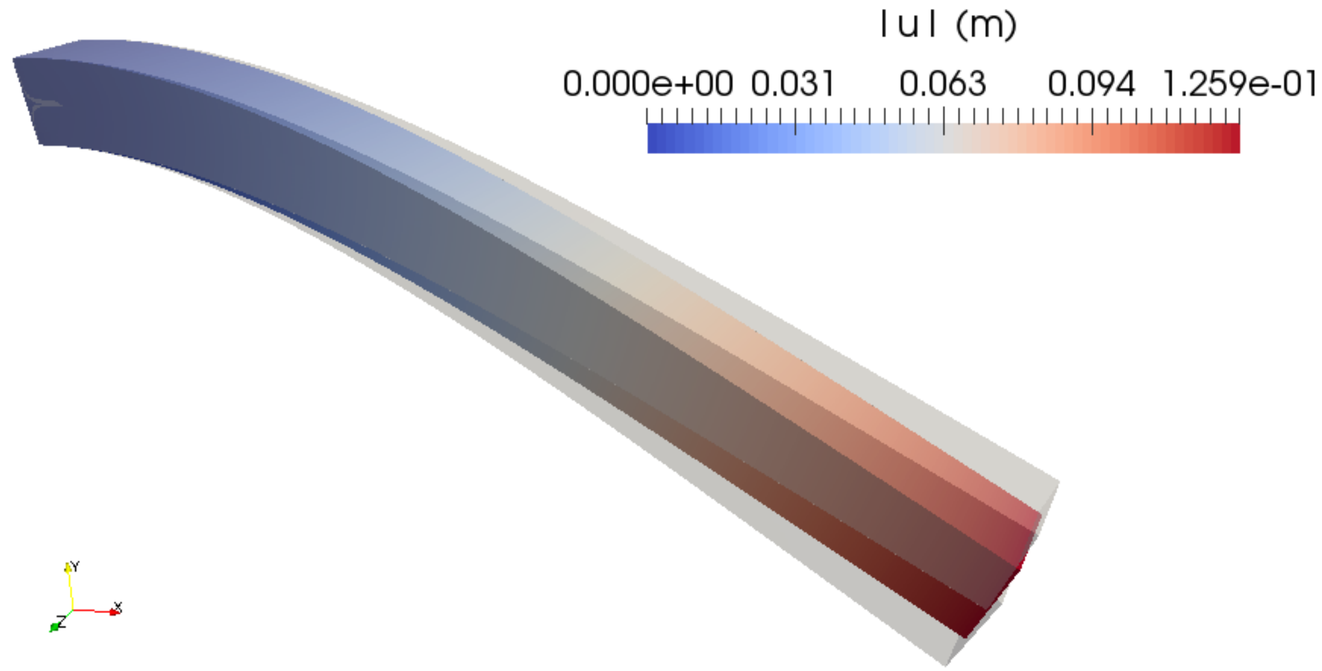
SOFA-based simulations

Real-time error-driven needle insertion

with Bui Huu Phuoc, Hadrien Courtecuisse, Satyendra Tomar, Franz Chouly and Stéphane Cotin

Patient-specific material models and parameters?

- Non-linear hyperelastic model as a stochastic PDE with random coefficients
- *Partially-intrusive* Monte-Carlo methods to propagate uncertainty



Deformation of the beam: mean +/- standard deviation

- Implementation: DOLFIN [Logg et al. 2012] and chaospy [Feinberg and Langtangen 2015]
- Ipyparallel and mpi4py to massively parallelise individual forward model runs across a cluster

Monte-Carlo method

- A non-linear stochastic system:

$$F(\boldsymbol{u}, \boldsymbol{\omega}) = \mathbf{0}$$

- Expected value of a quantity of interest [Caflisch 1998]:

$$E(\psi\left(\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{\omega})\right)) = \int_{\Omega} \psi\left(\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{\omega})\right)\ dP(\boldsymbol{\omega}) = \frac{1}{Z}\sum_{z=1}^{Z} \psi\left(\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{\omega_z})\right) + o\left(\frac{||\ \psi\ ||}{\sqrt{Z}}\right)$$

Probability space: $(\Omega, \mathcal{F}, P)$

Random parameters: $\boldsymbol{\omega} = \left(\omega_1, \omega_2, \ldots, \omega_M\right)$

- The classical Monte-Carlo approach:

$$E(\psi\left(\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{\omega})\right))^{MC} \approx \frac{1}{Z}\sum_{z=1}^{Z} \psi\left(\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{\omega_z})\right)$$

MC method with use of sensitivity information

- Expected value of a quantity of interest [Cao *et al.* 2004]:

$$E(\psi(\boldsymbol{u}(\boldsymbol{x},\boldsymbol{\omega})))^{SD-MC} \approx \frac{1}{Z}\sum_{z=1}^{Z}\left(\psi(\boldsymbol{u}(\boldsymbol{x},\boldsymbol{\omega_z})) - \sum_{i=1}^{M}\frac{d\psi}{d\omega_i}(\bar{\boldsymbol{\omega}}) \times (\omega_i - \bar{\omega}_i)\right)$$

- Tangent linear model to evaluate the sensitivity derivatives [Farrell *et al.* 2013]:

$$\underbrace{\frac{\partial F(\boldsymbol{u},\boldsymbol{\omega})}{\partial \boldsymbol{u}}}_{U\times U}\ \underbrace{\frac{d\boldsymbol{u}}{d\boldsymbol{\omega}}}_{U\times M} = -\underbrace{\frac{\partial F(\boldsymbol{u},\boldsymbol{\omega})}{\partial \boldsymbol{\omega}}}_{U\times M}$$

U: size of the deterministic problem
M: number of random parameters

- First and Second moments of the displacement:

$$\bar{\boldsymbol{u}} \approx \frac{1}{Z}\sum_{z=1}^{Z}\left(\boldsymbol{u}(\boldsymbol{x},\boldsymbol{\omega_z}) - \sum_{i=1}^{M}\frac{d\boldsymbol{u}}{d\omega_i}(\bar{\boldsymbol{\omega}}) \times (\omega_i - \bar{\omega}_i)\right)$$

$$\bar{\boldsymbol{u}}^2 \approx \frac{1}{Z}\sum_{z=1}^{Z}\left(\boldsymbol{u^2}(\boldsymbol{x},\boldsymbol{\omega_z}) - 2\bar{\boldsymbol{u}}\sum_{i=1}^{M}\frac{d\boldsymbol{u}}{d\omega_i}(\bar{\boldsymbol{\omega}}) \times (\omega_i - \bar{\omega}_i)\right)$$

Multi-level MC method with use of PCE

- Polynomial chaos expansion (PCE) [Wiener 1936]:

$$\boldsymbol{u^k}(\boldsymbol{x},\boldsymbol{\omega}) = \sum_{\alpha \in \mathcal{J}_{M,p}} \boldsymbol{u^k_\alpha}(\boldsymbol{x}) H_\alpha(\boldsymbol{\omega})$$

$$\dim(\mathcal{J}_{M,p}) = (M+p)!/(M!p!)$$

- ML-MC method [Matthies 2008, Giles 2015]:

Algorithm 1 Algorithm for the multilevel Polynomial Chaos Expansion Monte-Carlo method

1: Solve the deterministic system with average parameters to obtain $\boldsymbol{u}^d$
2: $k \longleftarrow 1$
3: **while** no convergence **do**
4: **for** $z = 1$ to Z **do**
5: Generate $\boldsymbol{\omega_z} = (\omega_1^z, \omega_2^z, \ldots, \omega_M^z)$
6: Generate $\boldsymbol{u^k}(\boldsymbol{\omega_z}) = F_{pce}\left(\boldsymbol{u^{k-1}}(\boldsymbol{\omega_z})\right)$ or $\boldsymbol{u}^d$ if $k == 1$
7: Call to deterministic solver to do d (1 or more) iterations with starting values $\boldsymbol{u^k}(\boldsymbol{\omega_z})$ and all random parameter function of $\boldsymbol{\omega_z}$
8: output: $\boldsymbol{u^k}(\boldsymbol{\omega_z})$ after d iterations
9: **end for**
10: Calculate F_{pce}, the PCE of $\boldsymbol{u^k}$ from Z values of $\boldsymbol{\omega_z}$ and $\boldsymbol{u^k}(\boldsymbol{\omega_z})$
11: $k = k + 1$
12: **end while**

3D Numerical simulations

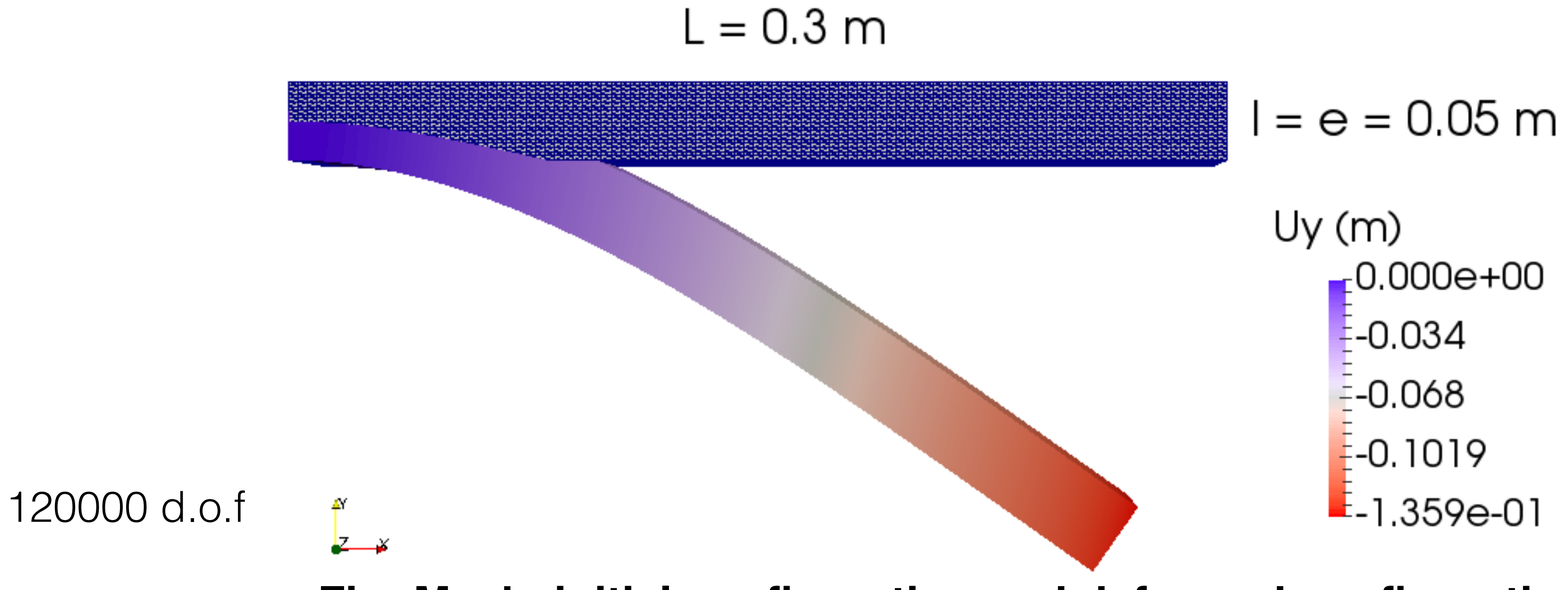


Fig: Mesh, initial configuration and deformed configuration.

- The stored strain energy density function for a compressible Mooney–Rivlin material:

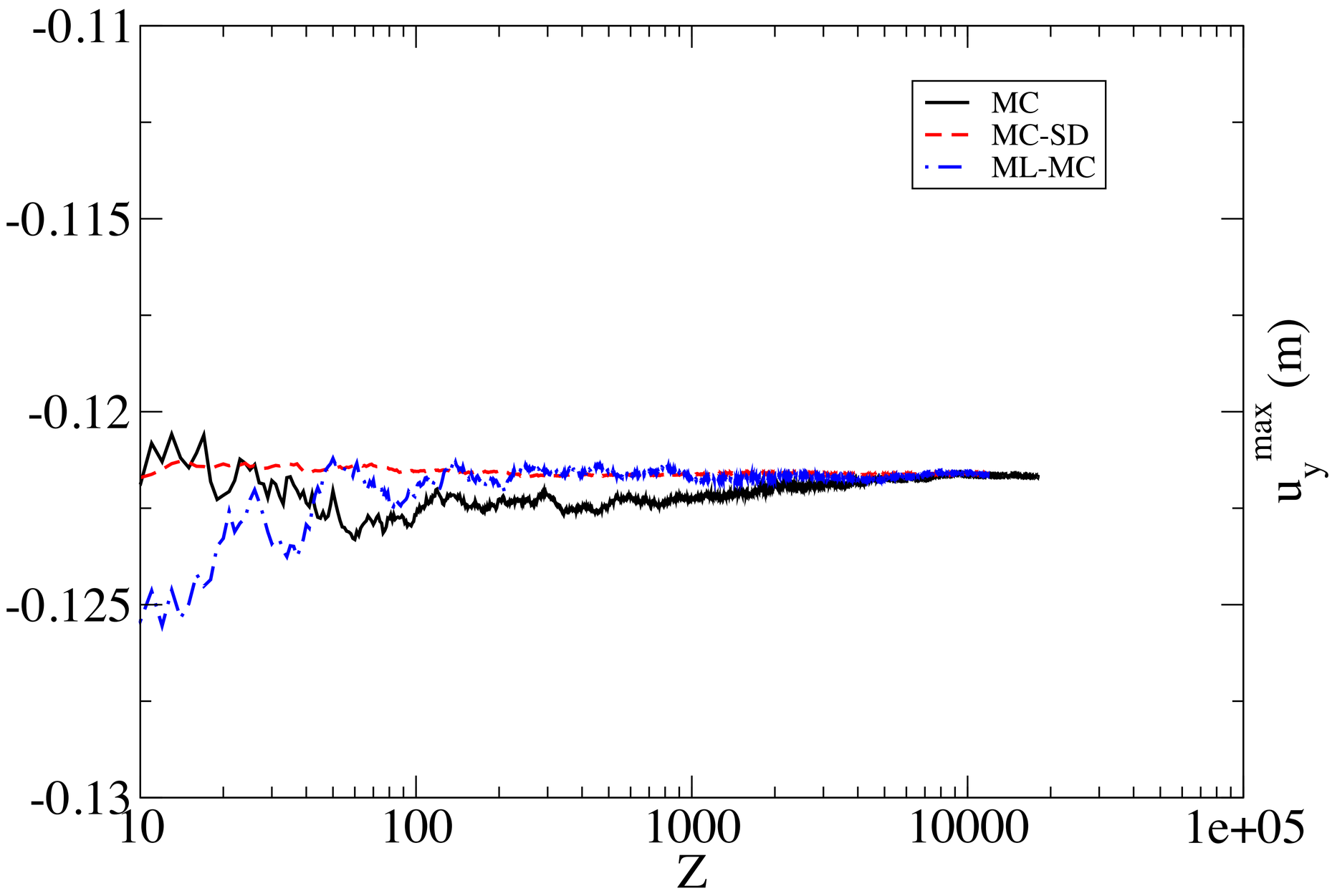
$$W = C_1(\overline{I}_1 - 3) + C_2(\overline{I}_2 - 3) + D_1(\det \mathbf{F} - 1)^2$$

- The total potential energy: $\Pi = W d\boldsymbol{x} - \rho \boldsymbol{g} d\boldsymbol{x}, \ \left(\boldsymbol{g} = g\vec{y}, g = 9.81 \ m.s^{-2}\right)$

- 2 RV with beta(2,2) distribution:

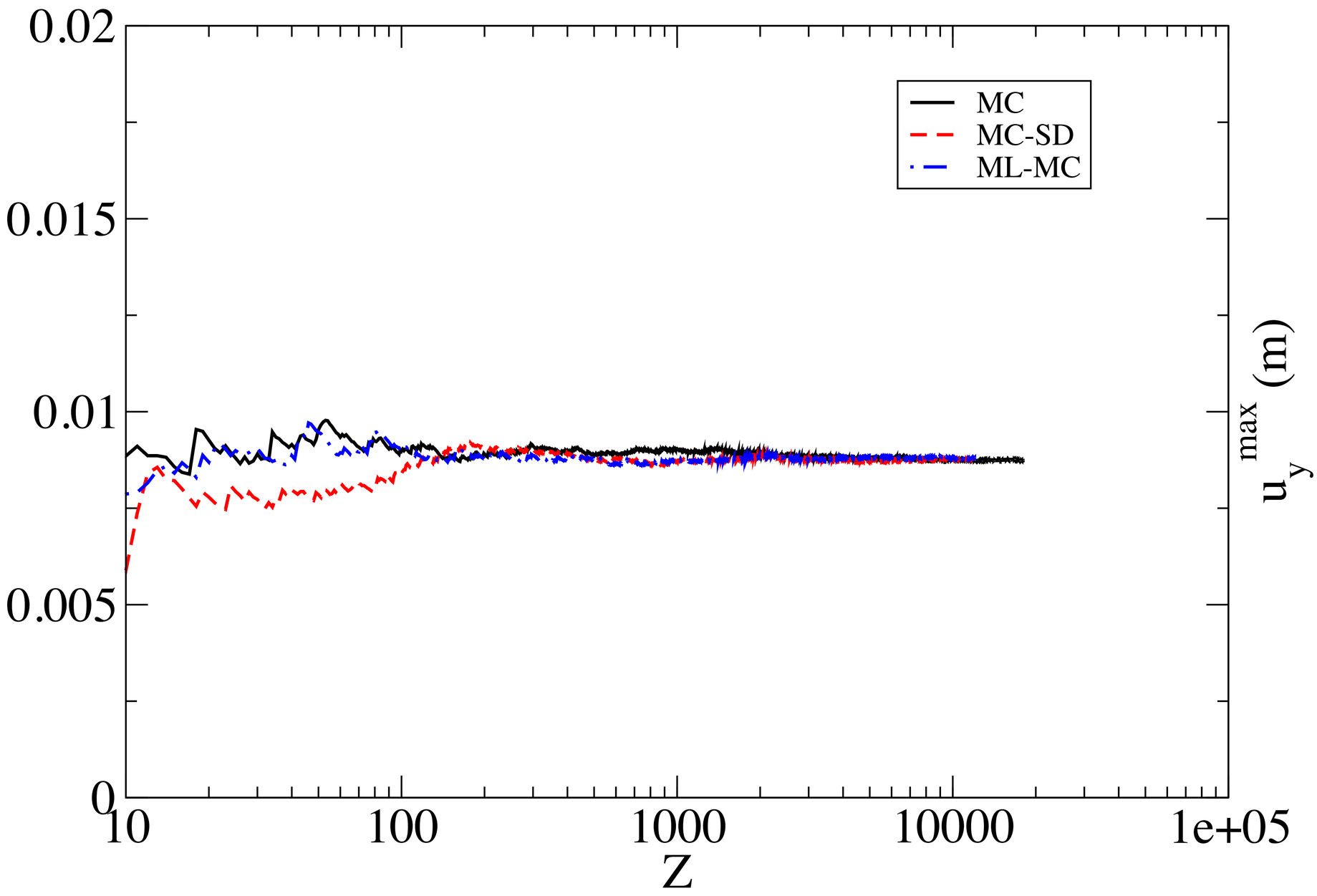
$$\begin{aligned} \rho(\omega_1) &= \rho^0(1 + \omega_1/2) \\ D_1(\omega_2) &= D_1^0(1 + \omega_2) \end{aligned} \quad \begin{cases} D_1^0 = 2 \cdot 10^5 \text{ Pa} \\ C_2 = 2 \cdot 10^5 \text{ Pa} \\ C_1 = 10^4 \text{ Pa} \\ \rho^0 = 600 \ kg/m^3 \end{cases}$$

3D Numerical simulations



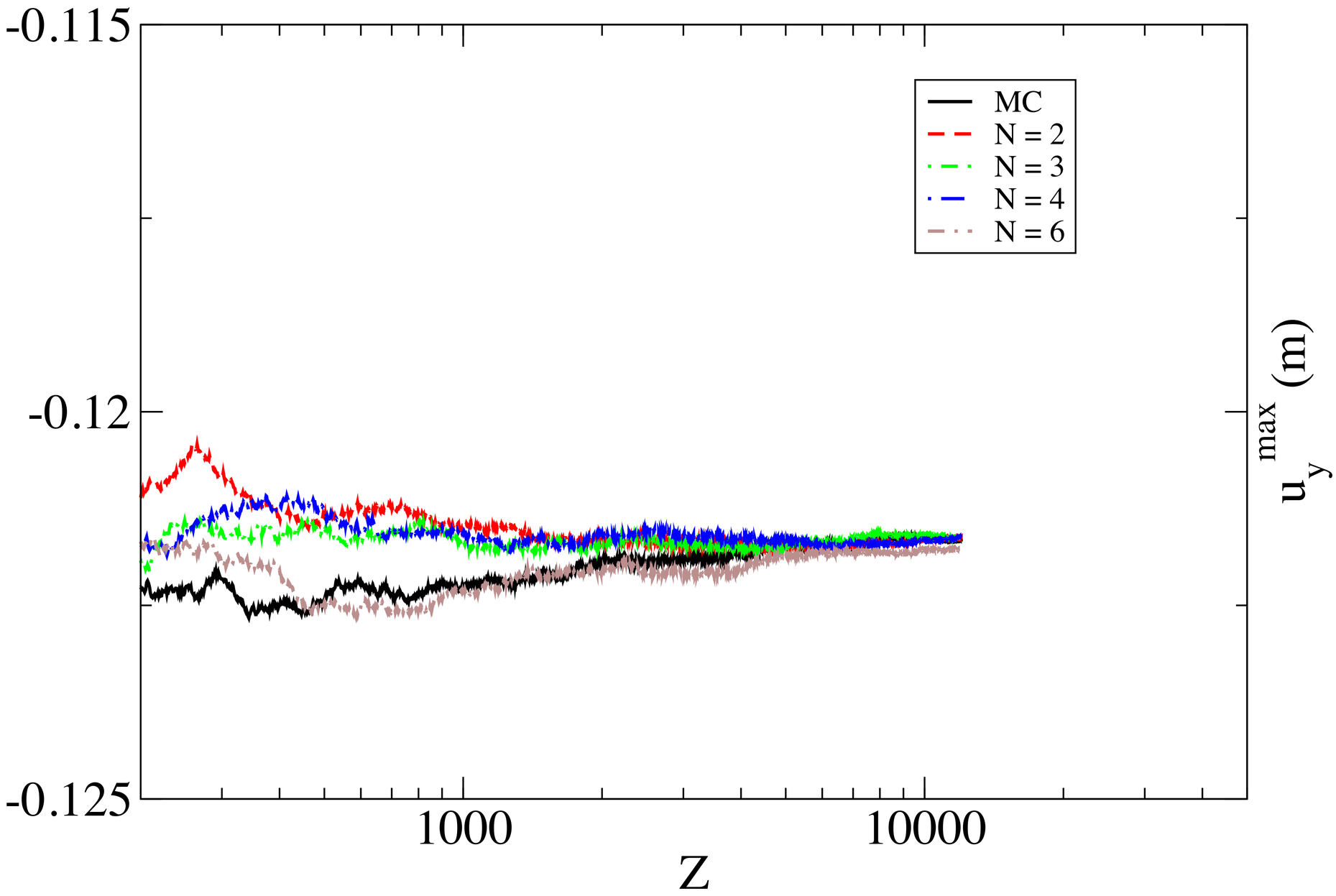
Mean

3D Numerical simulations



Std

3D Numerical simulations



Mean: influence of the number of levels

3D Numerical simulations

Global and local sensitivity analysis [Sobol 2001, Sudret 2008]

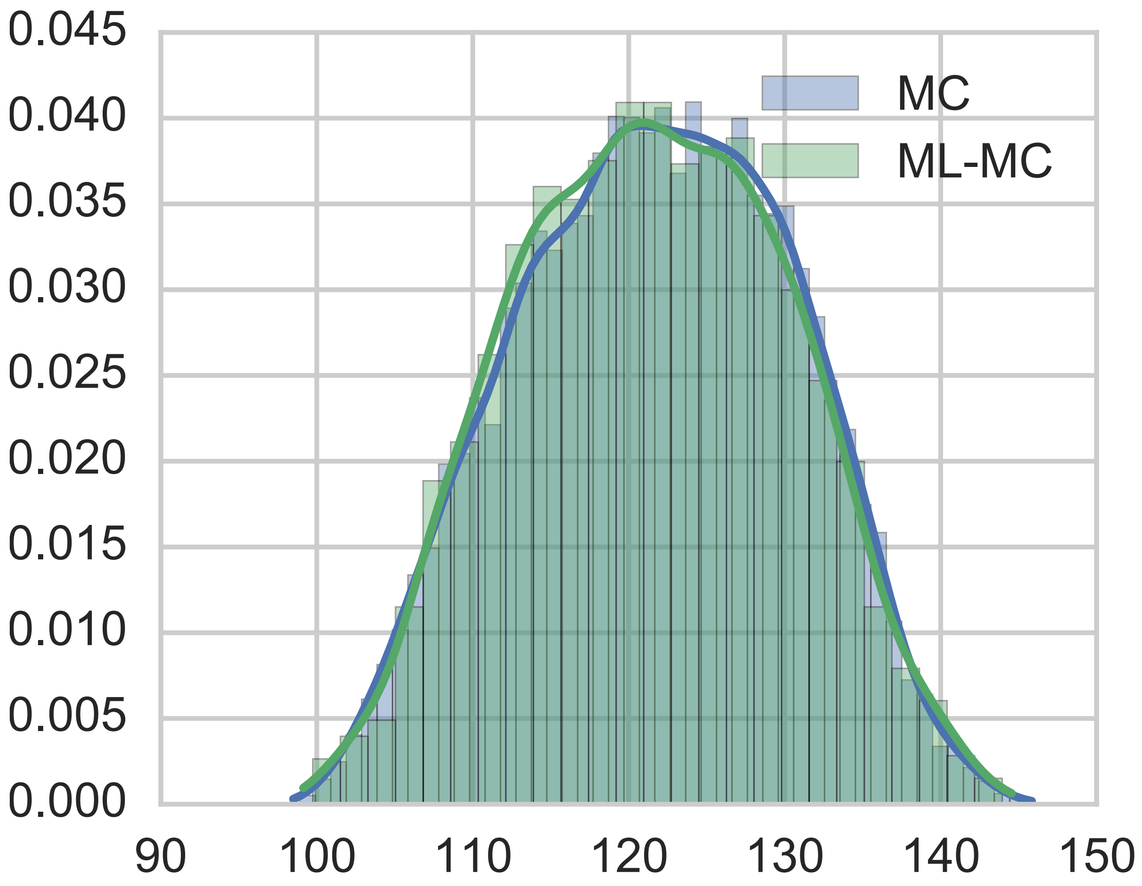
	ω_2	ω_1
$\left\| \frac{d\boldsymbol{u}}{d\omega_i} \right\|^{max}$	1.64	4.36
$\left\| \frac{du_y}{d\omega_i} \right\|^{max}$	0.014	0.036
$\left\| \frac{du_x}{d\omega_i} \right\|^{max}$	0.0081	0.021

Table 1: Local sensitivity around mean parameters

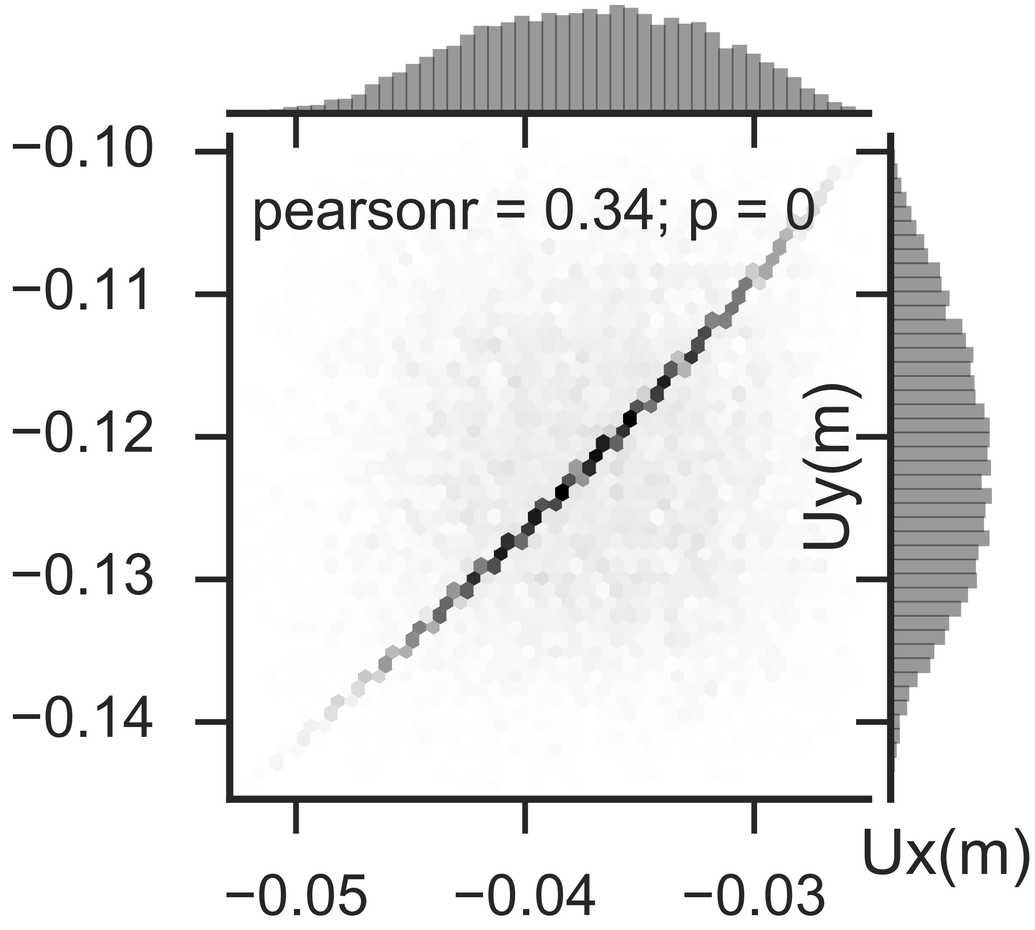
	u_x^{max}		u_y^{max}	
	ω_2	ω_1	ω_2	ω_1
First order	0.136	0.862	0.133	0.867
Total effect	0.138	0.862	0.133	0.868

Table 2: Sobol's sensitivity indices (global sensitivity) for the quantities of interest

3D Numerical simulations



$|u_y^{max}|(mm)$



MC-simulations (Z=18000)

	MC	MC-SD	ML-MC
T (min)	1100	65	225

Computational time with 120 engines running in parallel: comparison between the different methods with a number of realisations to have an accurate solution (MC with Z = 18000, MC-SD with Z = 1000 and ML-MC with Z = 4500).

Uncertainty Quantification: conclusions

- *Partially-intrusive* Monte-Carlo methods to propagate uncertainty

- By using sensitivity information and multi-level methods with polynomial chaos expansion we demonstrate that **computational workload can be reduced by one order of magnitude over commonly used schemes**

- Implementation: **DOLFIN** [Logg et al. 2012] and chaospy [Feinberg and Langtangen 2015]

- Ipyparallel and mpi4py to **massively parallelise individual forward model runs** across a cluster

The approach can be used ~immediately as a black box

Patient specific material models

Model selection and Bayesian updating

- What is the best model for the patient?
- What are the associated material parameters

Data-driven, Bayesian model selection and parameter identification

1. Prior

Prior knowledge
Material parameters inc. distribution from general population

Noise
- Model (e.g. additive)
- Distribution (Gauss)
- Characterisation

Constitutive model

$$\psi^{\mathrm{eq}}(\mathbf{F}) = \frac{\mu}{2}\left(\frac{(\mathrm{tr}\,\mathbf{B})}{J^{2/3}} - 3\right) + \frac{1}{2}K(J-1)^2$$

Data assimilator
Bayesian inference

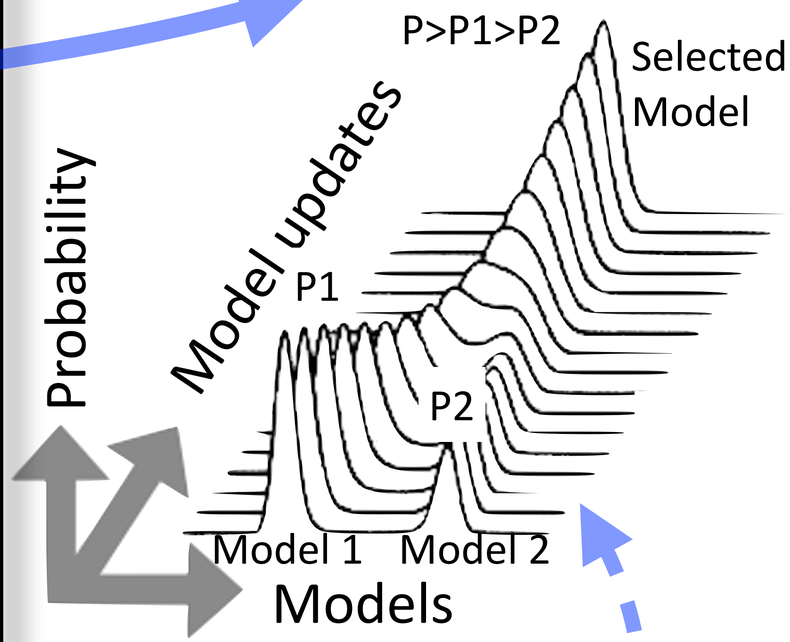
$$\Pr(M|D) = \frac{\Pr(D|M)\Pr(M)}{\Pr(D)}.$$

MRI

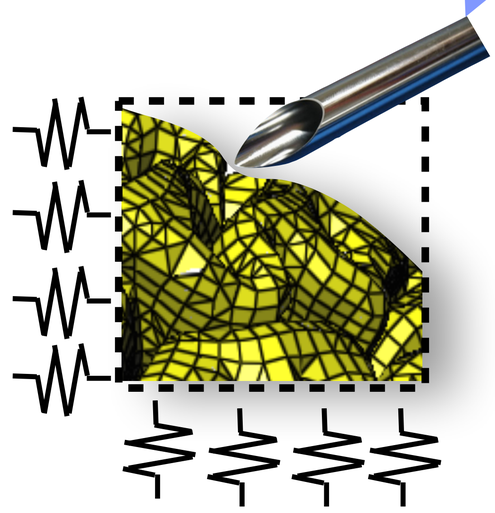
Stereo-cameras

Simulator

3. Model selector



Mechanical solver



2. Posterior data

CT

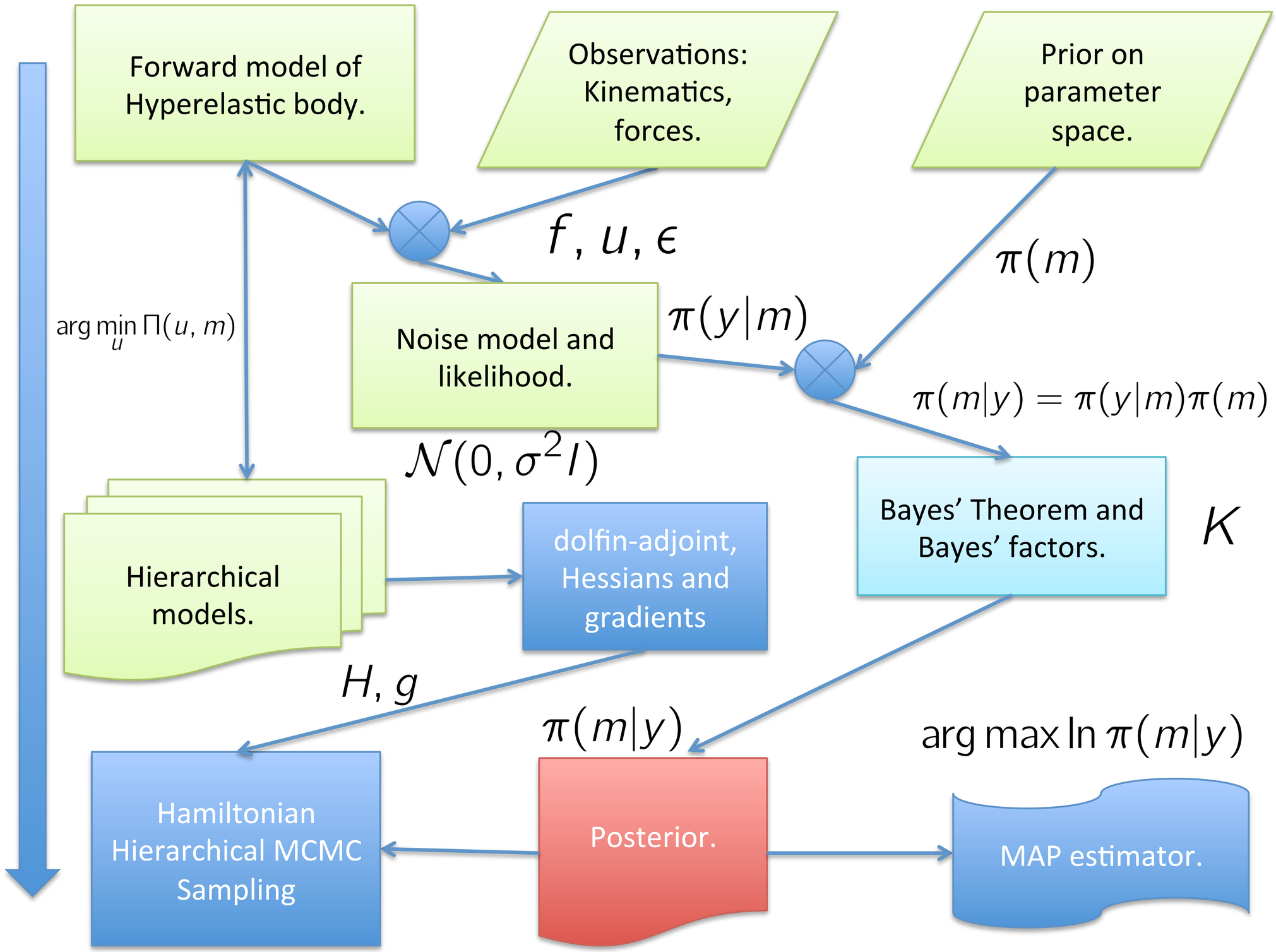
Ultrasound

5. Action

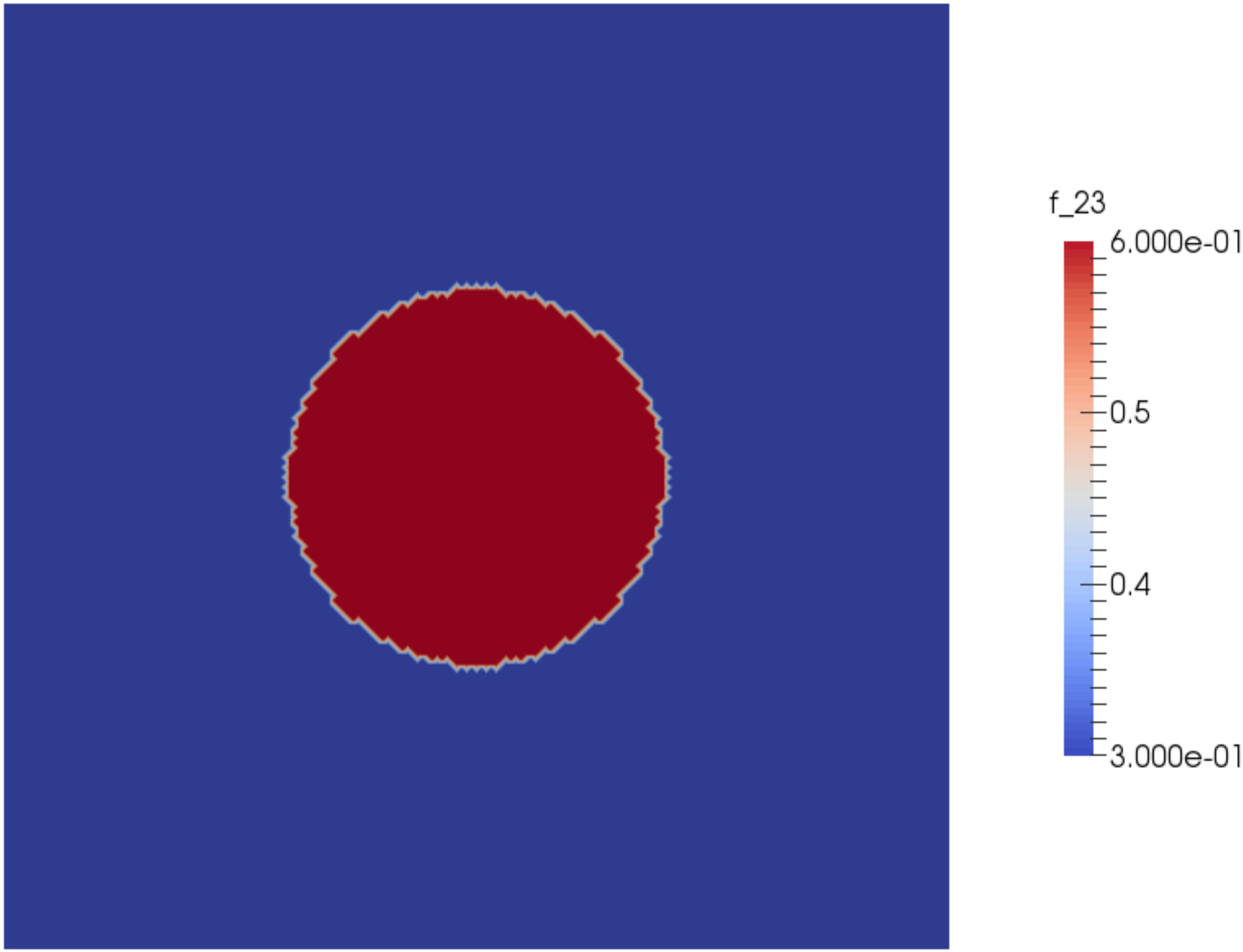
4. Feedback

What is the best model for the patient?

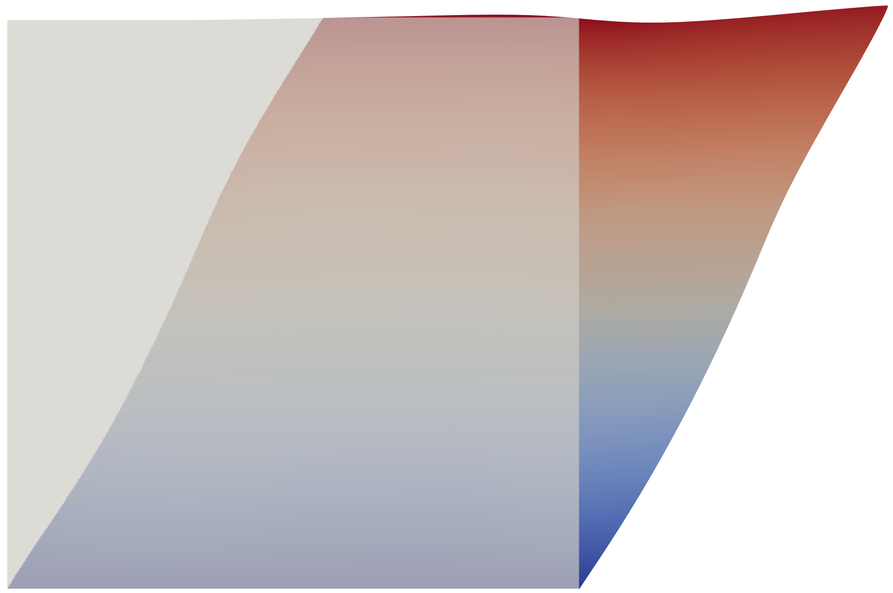
Bayesian model selection

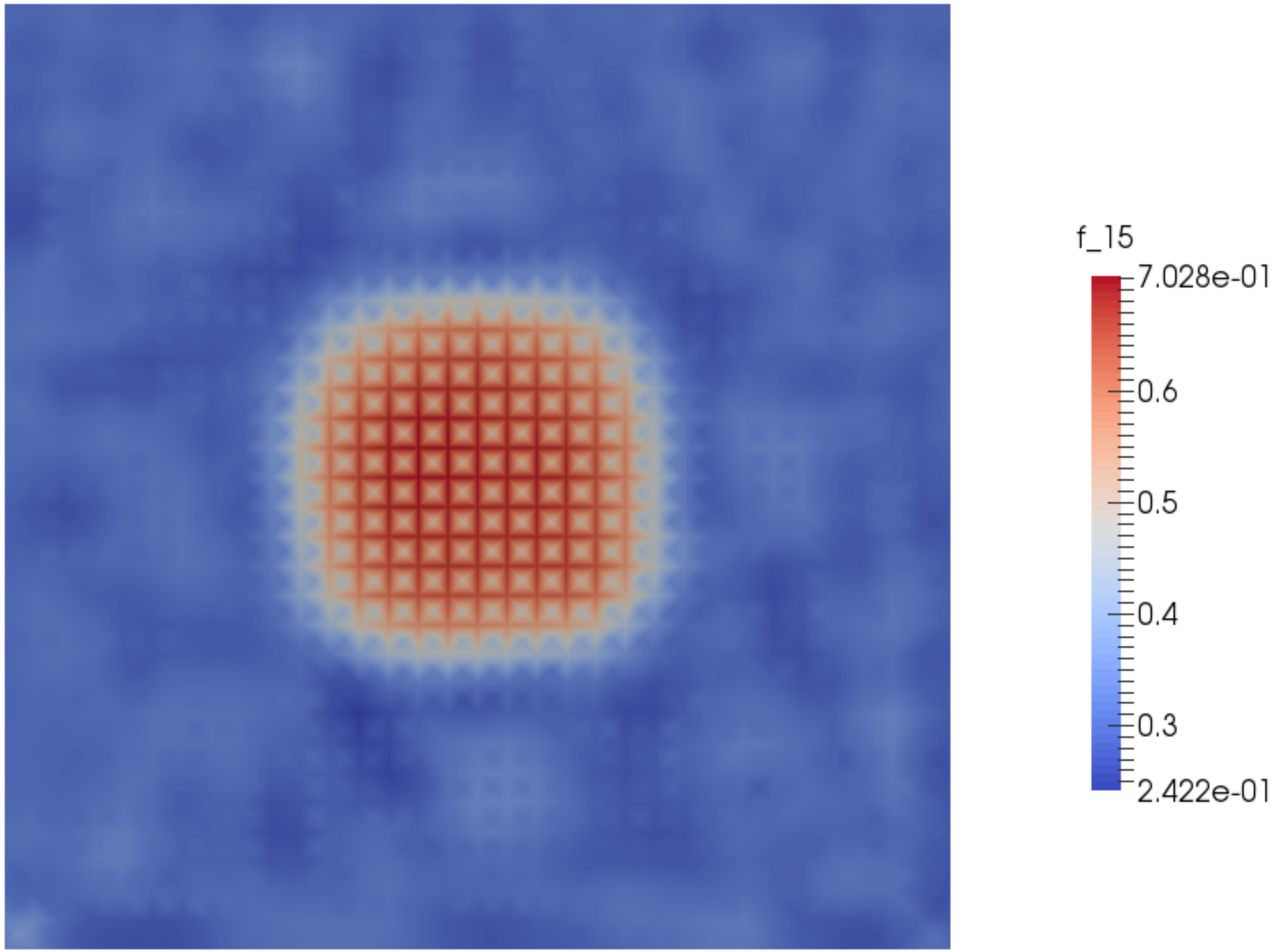


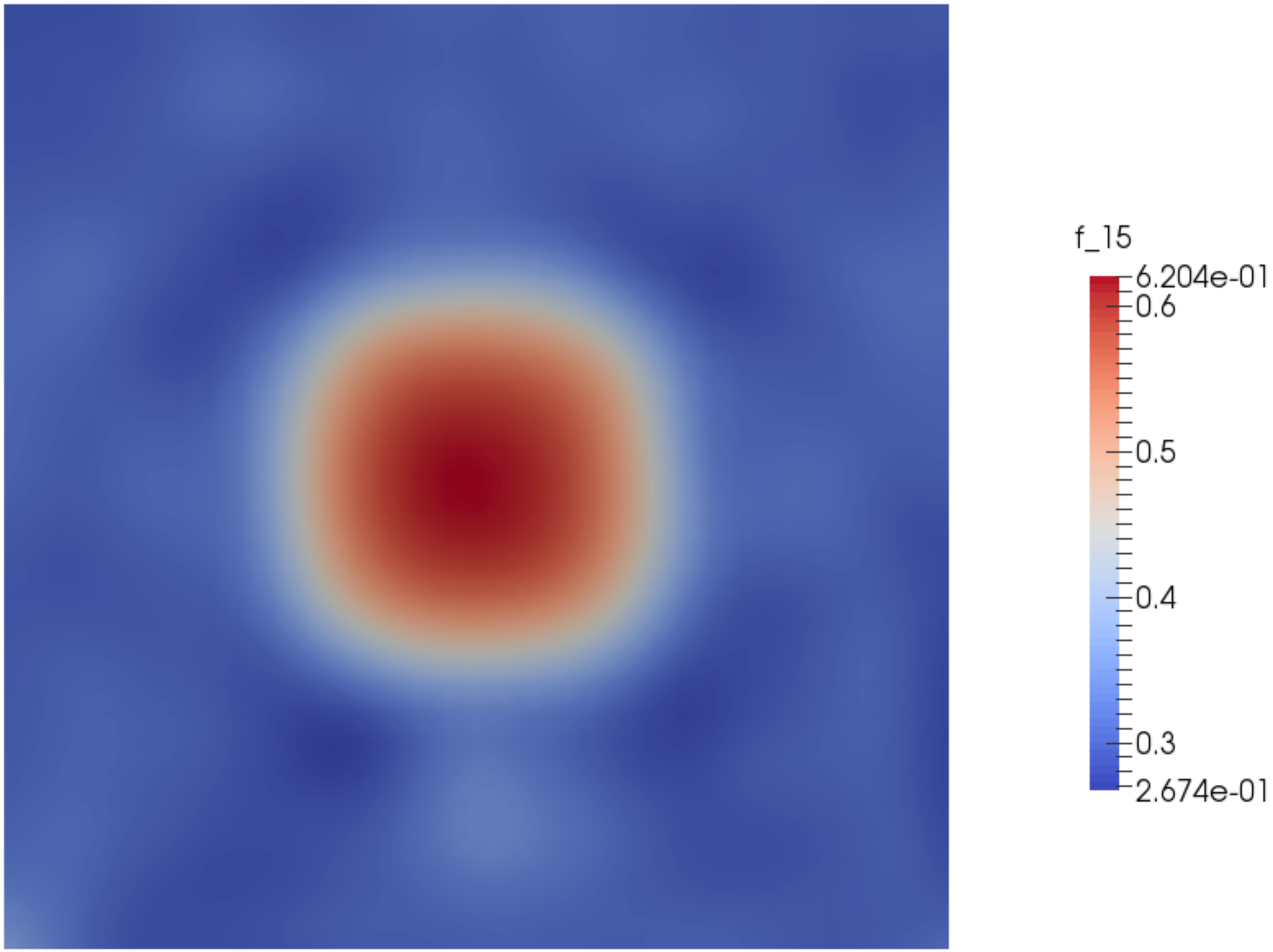
Identify inclusion geometry and parameters knowing only displacement field on the boundary

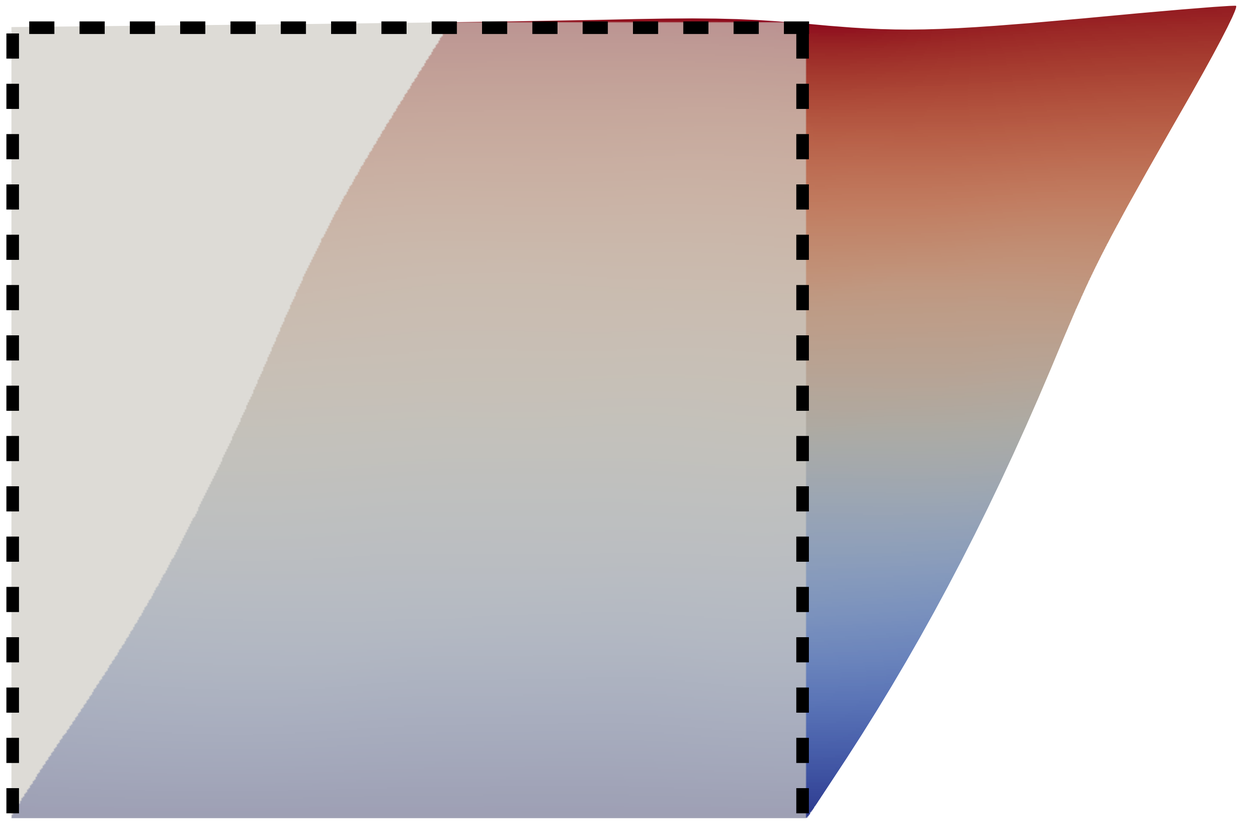


Impose shear, compression, extension loading




f_15
7.028e-01
0.6
0.5
0.4
0.3
2.422e-01


f_15
6.204e-01
0.6
0.5
0.4
0.3
2.674e-01



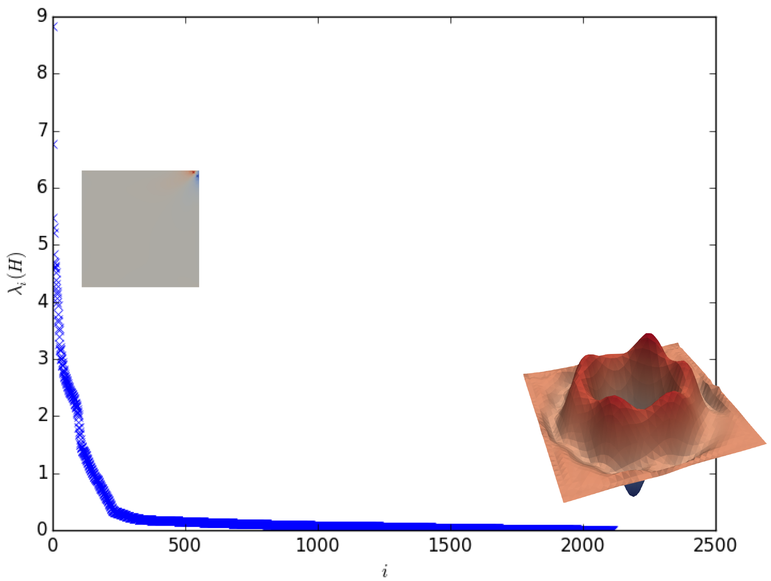
Q: What can we infer about the parameters inside the domain, just from displacement observations on the outside?

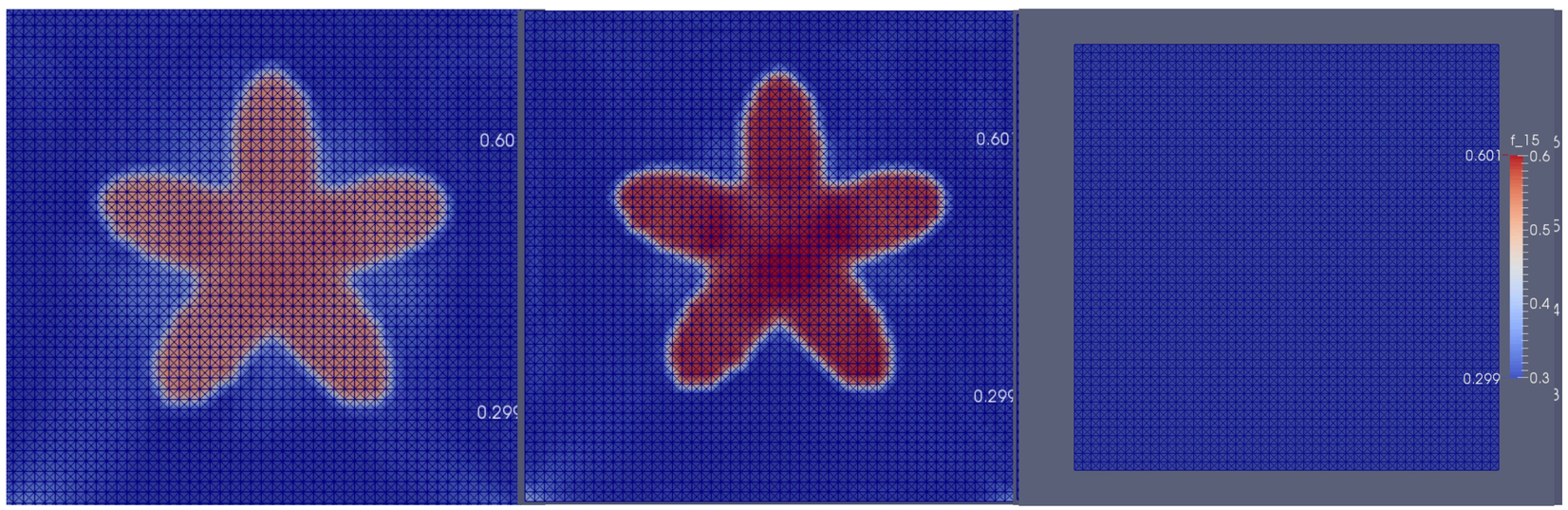
Q: Which parameters are we most uncertain about?

Large-scale statistical inverse pbs

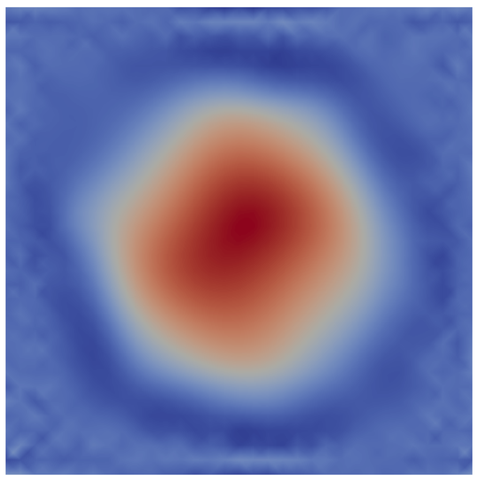
Proof of concept

Hyper-elastic statistical inversion 2.10^3 parameters



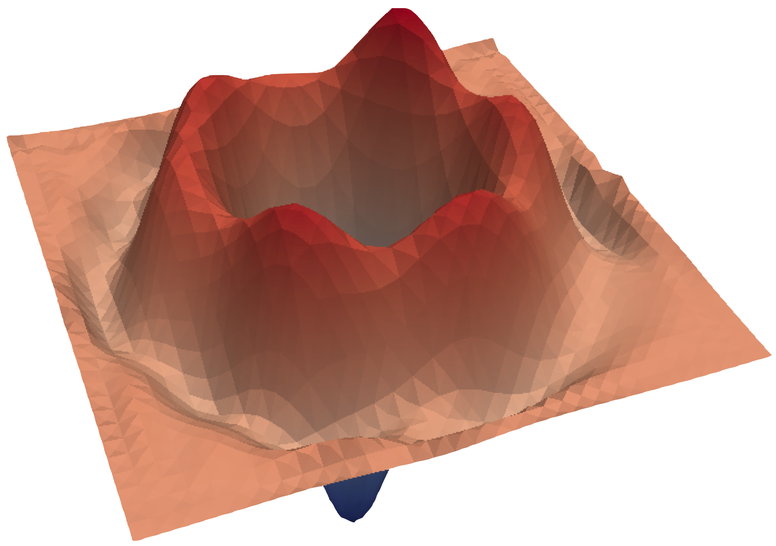


Low rank structure of posterior.



Recovered parameters under sparse observations

Direction most constrained by data

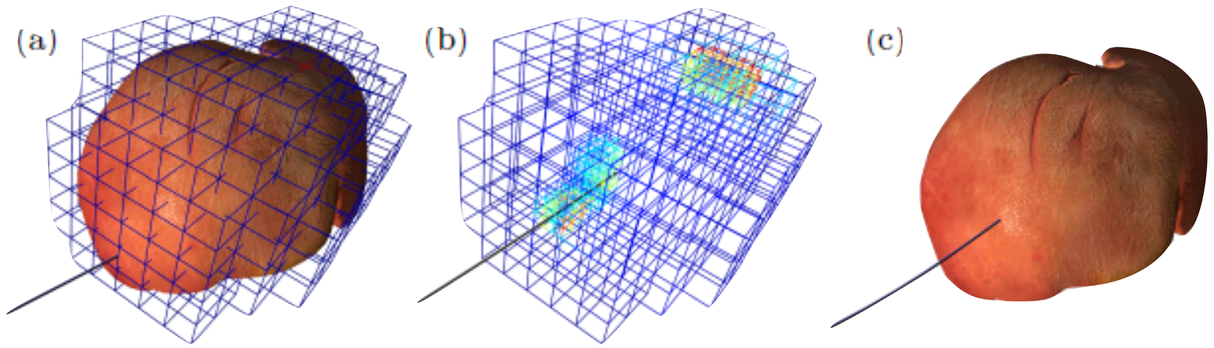


Direction least constrained by data (trailing eig.val)

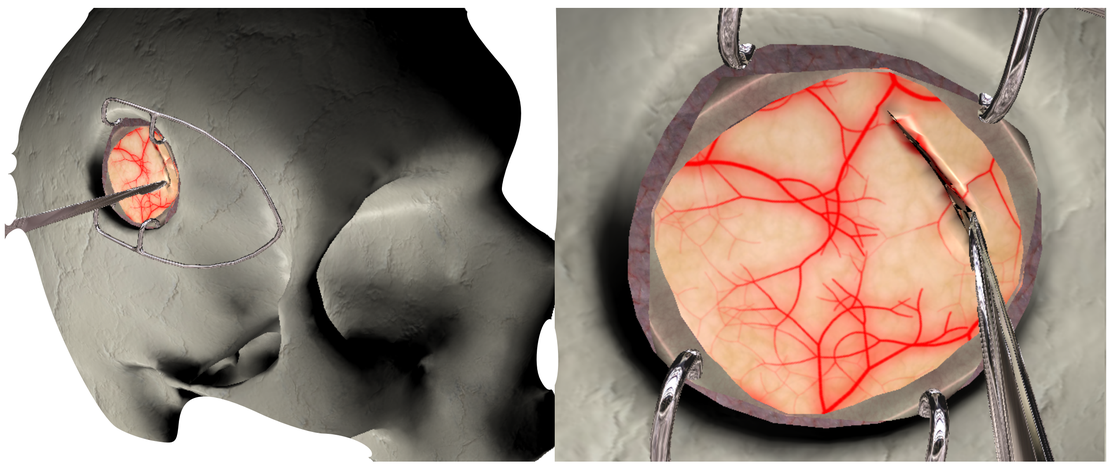

There's a fine line between
wrong and visionary.
Unfortunately,
you have to be a
visionary to see it.
Sheldon Cooper,
The Big Bang Theory: The Pirate Solution

Long-term VISION

Provide real-time guidance to surgeons in uncertain conditions



Real-time error-estimator driven mesh adaptation
Bui, Courtecuisse, Cotin, Bordas et al., 2016



Real-time implicit simulation of cutting in heterogeneous soft tissues
Courtecuisse, Cotin, Duriez, Bordas et al., Medical image analysis 2014

Open questions

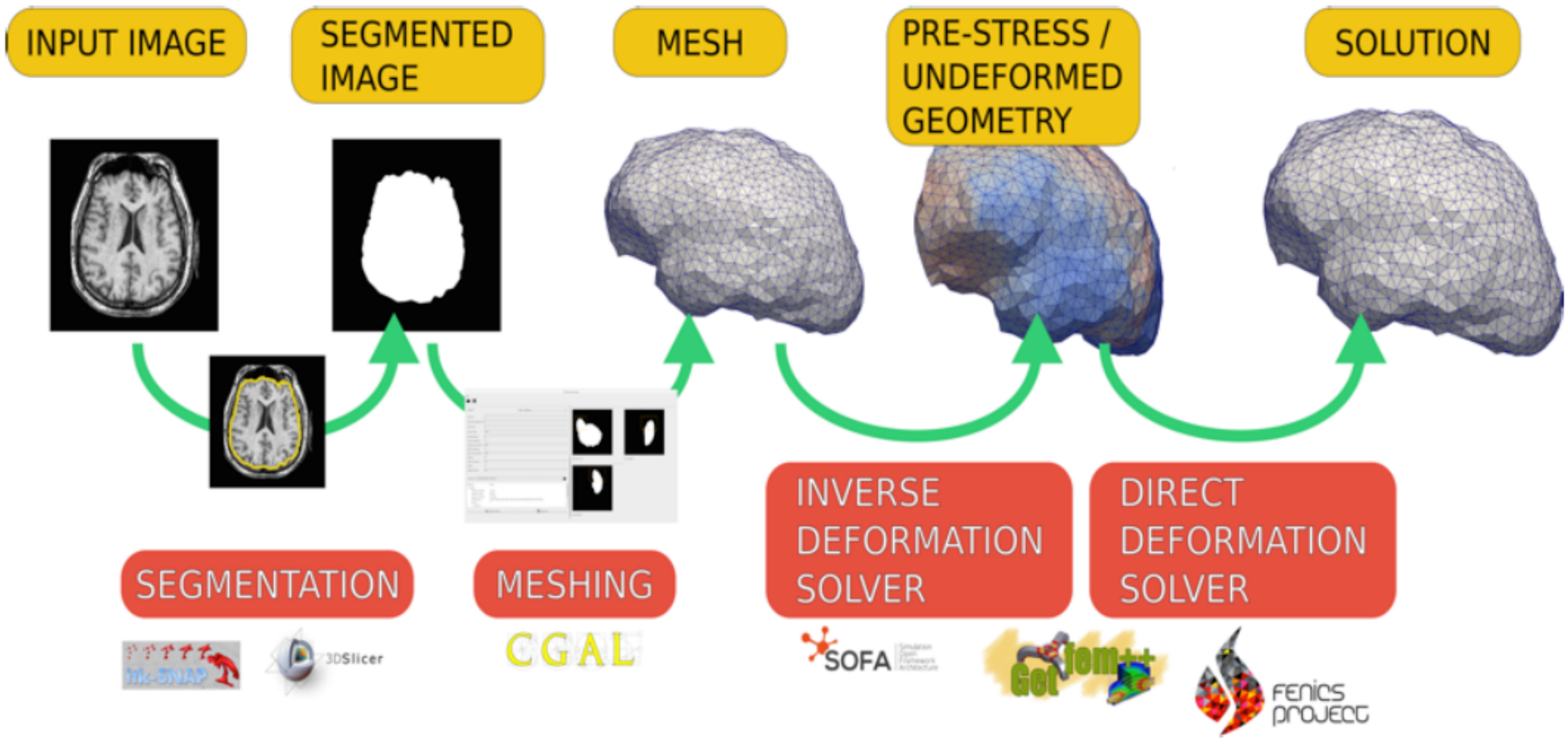
- How can we determine the importance of the material model? Machine learning sufficient?
- Of the associated material parameters?
- How important is it to model the geometry of the organs?
- Tool-tissue interaction models?
- Data-driven approaches that store known data in prior and learns from new data to construct posterior?
- Model order reduction approaches amenable to real-time simulations of "large" non-linear systems.

Our approach

Control errors vs. compute time
Control uncertainty
Choose the right material model
Domain-based model order reduction

From mesh to biomechanical solution: pipeline

Thank you for your attention



with Alex Bilger

Tools

- The FEniCS Project is collection of free software for the automated, efficient solution of differential equations using the finite element method.

- dolfin-adjoint automatically derives the discrete adjoint, tangent linear and higher-order adjoint models from a high-level description of the forward model.

http://fenicsproject.org

Wells, Logg, Rognes, Kirby and many, many others…

http://www.dolfin-adjoint.org

Farrell, Funke, Ham and Rognes.
2015 Wilkinson Prize for Numerical Software.

```
from dolfin import *
from dolfin_adjoint import *

mesh = UnitSquareMesh(32, 32)

U = VectorFunctionSpace(mesh, "CG", 1)
V = FunctionSpace(mesh, "CG", 1)
# solution
u = Function(U)
# test functions
v = TestFunction(U)
# incremental solution
du = TrialFunction(U)
mu = interpolate(Constant(1.0), V)
lmbda = interpolate(Constant(100.0), V)
```

```
dims = mesh.type().dim()
I = Identity(dims)
F = I + grad(u)
C = F.T*F
J = det(F)
Ic = tr(C)
```

```
dims = mesh.type().dim()
I = Identity(dims)
F = I + grad(u)
C = F.T*F
J = det(F)
Ic = tr(C)
phi = (mu/2.0)*(Ic - dims) -
mu*ln(J) + (lmbda/2.0)*(ln(J))**2
Pi = phi*dx
# Gateaux derivative with respect to
u in direction v
F = derivative(Pi, u, v)
# and with respect to u in direction
du
J = derivative(F, u, du)

u_h = Function(U)
F_h = replace(F, {u: u_h})
J_h = replace(J, {u: u_h})
solve(F_h == 0, u_h, bcs, J=J_h)
```

2016 Dec 12-14

1st Luxembourg Medical Simulation Workshop