Uncertainty Quantification in Finite Element Models: Application to Soft Tissue Biomechanics

Paul Hauseux[1], Jack S. Hale[1], Raphaël Bulle[1], Franz Chouly[2], Alexei Lozinski[2] and Stéphane P. A. Bordas[1]

July 23, 2018

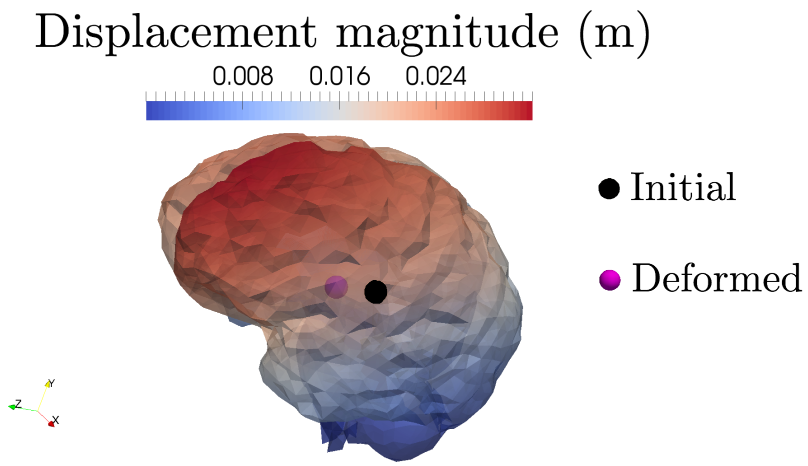
[1]University of Luxembourg, [2]Bourgogne France Comté University

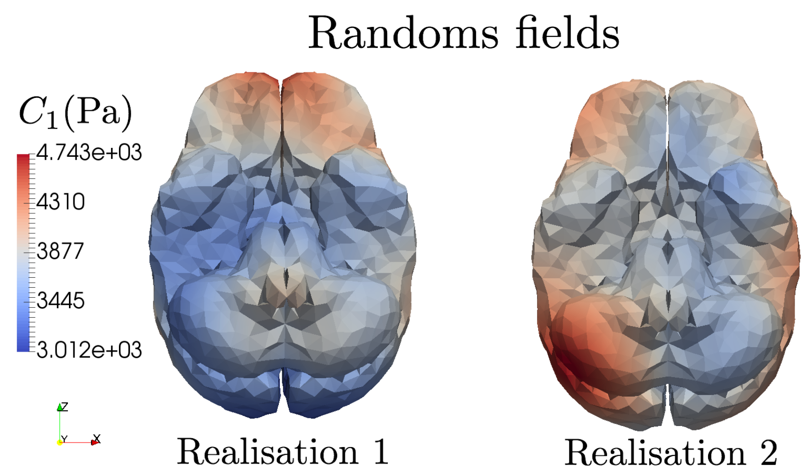
13th World Congress in Computational Mechanics (WCCM XIII) and 2nd Pan American Congress on Computational Mechanics (PANACM II)
July 22-27, 2018 Marriott Marquis

Context: Soft-tissue biomechanics simulations with uncertainty

- Uncertainty in parameters (material properties, loading, geometry, etc.) in biomechanics problems can influence the outcome of simulation results.

- Assessing the effects of uncertainty in material parameters in soft tissue models.
- Stochastic FE analysis. Random variables/fields.





- **Objective: propagate and visualise this uncertainty** with *non* or *partially-intrusive* methods (Forward UQ).

- Parameter identification (Inverse Problems in a Bayesian Setting).

General framework

- Stochastic non-linear system: $F(\boldsymbol{u}, \boldsymbol{\omega}) = \mathbf{0}$
- Probability space: $(\Omega, \mathcal{F}, P)$
- Random parameters: $\boldsymbol{\omega} = (\omega_1, \omega_2, \ldots, \omega_M)$

- Objective: provide statistical data for the solution of the problem.
- Integration (to determine the expected value of a quantity of interest):

$$E[\Psi(u(\omega))] = \int_{\Omega} \Psi(u(\omega)) dP(\omega)$$

Direct integration

Monte-Carlo method [Caflisch 1998]:

$$E[\Psi(u(\omega))] = \int_{\Omega} \Psi(u(\omega)) dP(\omega) \simeq \sum_{z=1}^{Z} p_z \Psi(u(\omega_z))$$

Algorithm:

while $z < Z$:

- choose randomly ω_z.
- evaluate $\Psi(u(\omega_z))$.
- add the contribution to the sum.

Convergence

- Converge «in law»: 1% for 10000 realisations, slow but independent of the dimension !

$$||\mathbb{E}^{\mathrm{MC}}\left[\psi(\omega)\right] - \mathbb{E}\left[\psi(\omega)\right]||_{L^2(\Omega_p)} \sim \mathrm{N}(0,1)\sqrt{\frac{\mathbb{V}[\psi(\omega)]}{Z}}$$

- Necessity to improve the convergence.

Work done:

- Low discrepancy sequences (Sobol, Hamilton, …): quasi MCM [Caflisch 1998].
- **Multi Level Monte-Carlo** techniques [Giles 2015], **Polynomial Chaos Expansion** [Matthies 2008] and **non-intrusive SGFEM** methods that only require access to a deterministic residual [Giraldi et al. 2014].
- MC methods by using **sensitivity information** (SD-MC) [Cao et. al 2004, Liu et al. 2013].

MC methods by using sensitivity information

Estimator [Cao et. al 2004, Liu et al. 2013]:

$$\mathbb{E}_1^{\mathrm{SD-MC}}[\psi(\omega)] := \frac{1}{Z}\sum_{z=1}^{Z}[\psi(\omega_z) - D[\psi(\bar{\omega})](\omega_z - \bar{\omega})]$$

This variance reduction method increases the accuracy of sampling methods. Here we only consider the case of the first-order sensitivity derivative enhanced Monte-Carlo method. By using sensitivity information computational workload can be reduced by one order of magnitude over commonly used schemes.

Main difficulty:

$$D[\psi(\bar{\omega})] \quad \textbf{??}$$

Numerical implementation

Implementation (DOLFIN/FEniCS) [Logg et al. 2012], advantages:

- UFL (Unified Form Language).

- Most existing FEM codes are not able to compute the tangent linear model and the sensitivity derivatives. However, it is possible with DOLFIN for a wide range of models with very little effort [Alnæs 2012, Farrell et al. 2013].

- We also use dolfin-adjoint to automatically derive the adjoint equations (first and second order) and their FE discretisation from UFL description. This gives us access to routines for calculating the gradient and Hessian-vector action of the QoI with respect to the parameter(s).

- Complex models with only few lines of Python code.
- Parallel computing (Ipyparallel and mpi4py).

Python package for uncertainty quantification:

- Chaospy [Feinberg and Langtangen 2015] to provide different stochastic objects (global sensitivity analysis, polynomial chaos exapnsion, etc.)

DOLFIN/FEniCS implementation: an example

- **Forward problem,** generalized Burgers equation with stochastic viscosity:

$$F(\nu, u; \tilde{u}) := \int_{\Omega_s} \nu \nabla u \cdot \nabla \tilde{u} - \frac{1}{2} \nabla u^2 \cdot \tilde{u} + \frac{1}{2} \nabla u \cdot \tilde{u} \, dx = 0 \quad \forall \tilde{u} \in H_0^1(\Omega_s)$$

```
nu_var = variable(Constant(omega))
F = nu_var*u_.dx(0)*u_t.dx(0)*dx + 0.5*u_.dx(0)*u_t*dx\
      - 0.5*(u_**2).dx(0)*u_t*dx
```

- The standard Newton method:

$$J(\nu, u^k; \delta u; \tilde{u}) = -F(\nu, u^k; \tilde{u}) \quad \forall \tilde{u} \in H_0^1(\Omega_s)$$

$$u^{k+1} = u^k + \delta u$$

```
J = derivative(F, u_, u)
solve(F == 0, u_, bcs, J=J)
```

DOLFIN/FEniCs implementation: an example

- **The tangent linear system:**

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

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

```
Fu = derivative(F, u, du)
Fd = - diff(F, omega)
dudomega = Function(V)
A, b = assemble_system(Fu, Fd, bcs=bcs)
solve(A, dudomega.vector(), b, “lu")
```

linear system to solve to evaluate du/dm !

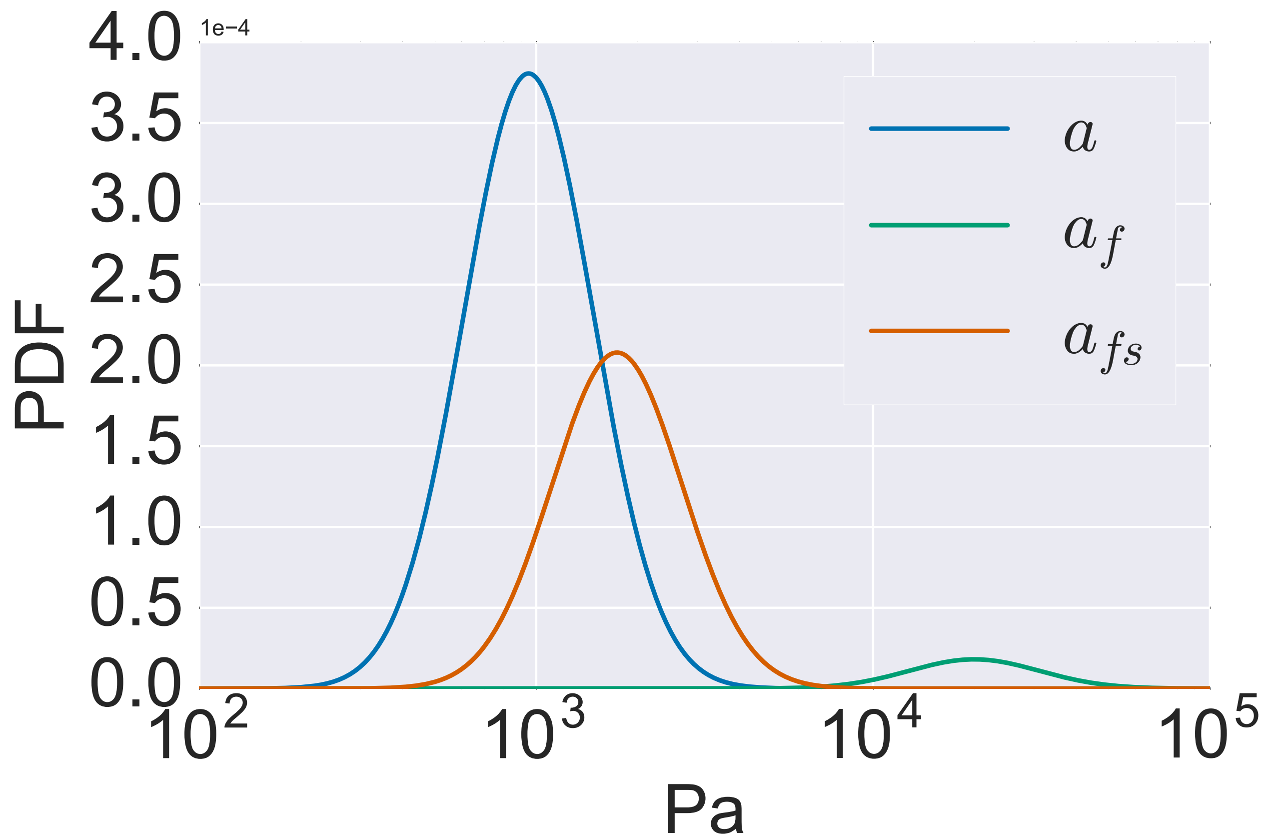
The complete implementation is only around 130 lines and the Docker image with the full software environment is included in: https://dx.doi.org/10.6084/m9.figshare.3561306 [Hauseux, P. and Hale, J.S. and Bordas, S. 2016]

Stochastic FE analysis of brain deformation

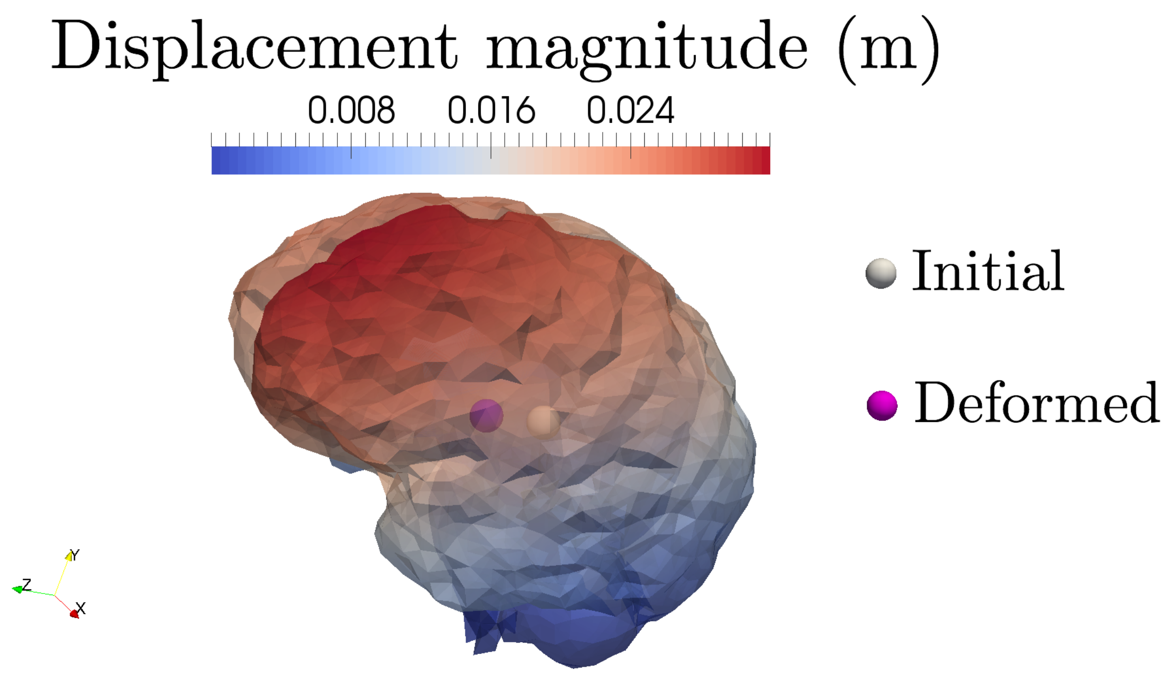
- Different hyper-elastic models implemented (Mooney-Rivlin, Neo-Hookean, Holzapfel and Ogden [Holzapfel and Ogden 2009]).
- Random variables/fields to model parameters [Adler 2007].
- Strain energy function for the Holzapfel and Ogden model:

$$\mathcal{W}_{iso} = \frac{a}{2b} \exp\left[b(\mathrm{I}_1 - 3)\right] + \sum_{i=f,s} \frac{a_i}{2b_i} \exp\left[b_i(\mathrm{I}_{4i} - 1)^2\right] + \frac{a_{fs}}{2b_{fs}} \left(\exp\left[b_{fs}\mathrm{I}_{8fs}^2\right] - 1\right)$$

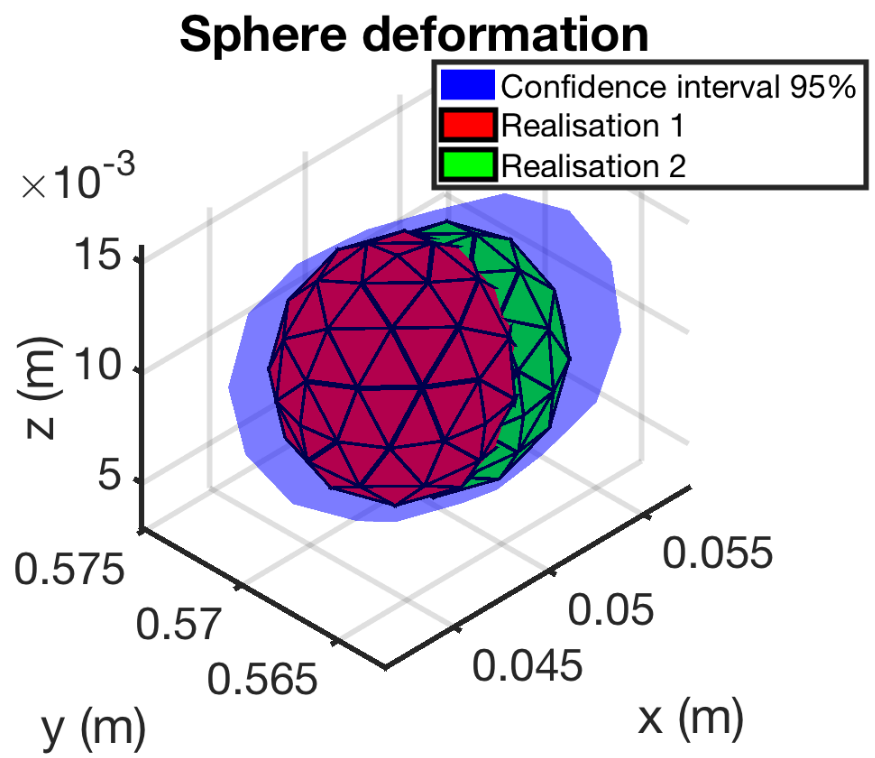
- for example 3RV:



Stochastic FE analysis of brain deformation Numerical results (8 RV, Holzapfel model)



Brain deformation with random parameters
1 MC realisation.



Confidence interval 95%
MC simulations.

The complete FEniCS implementation, the Docker image with the full software environment, the benchmarks problems and all associated data are available: http://bitbucket.org/unilucompmech/stochastic-hyperelasticity, http://doi.org/10.6084/m9.figshare.4900298 [Hauseux et al. 2018]

Numerical results: convergence

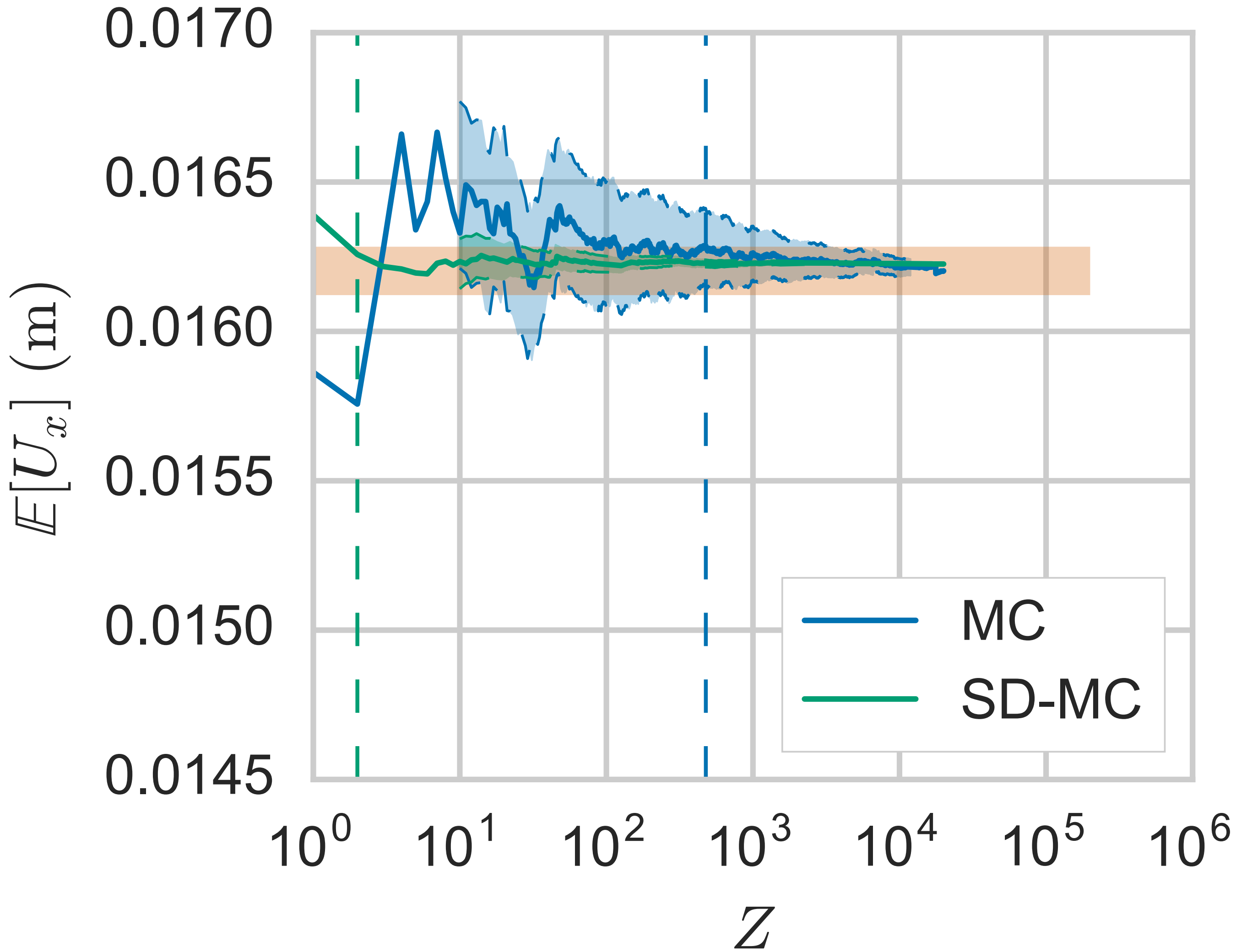
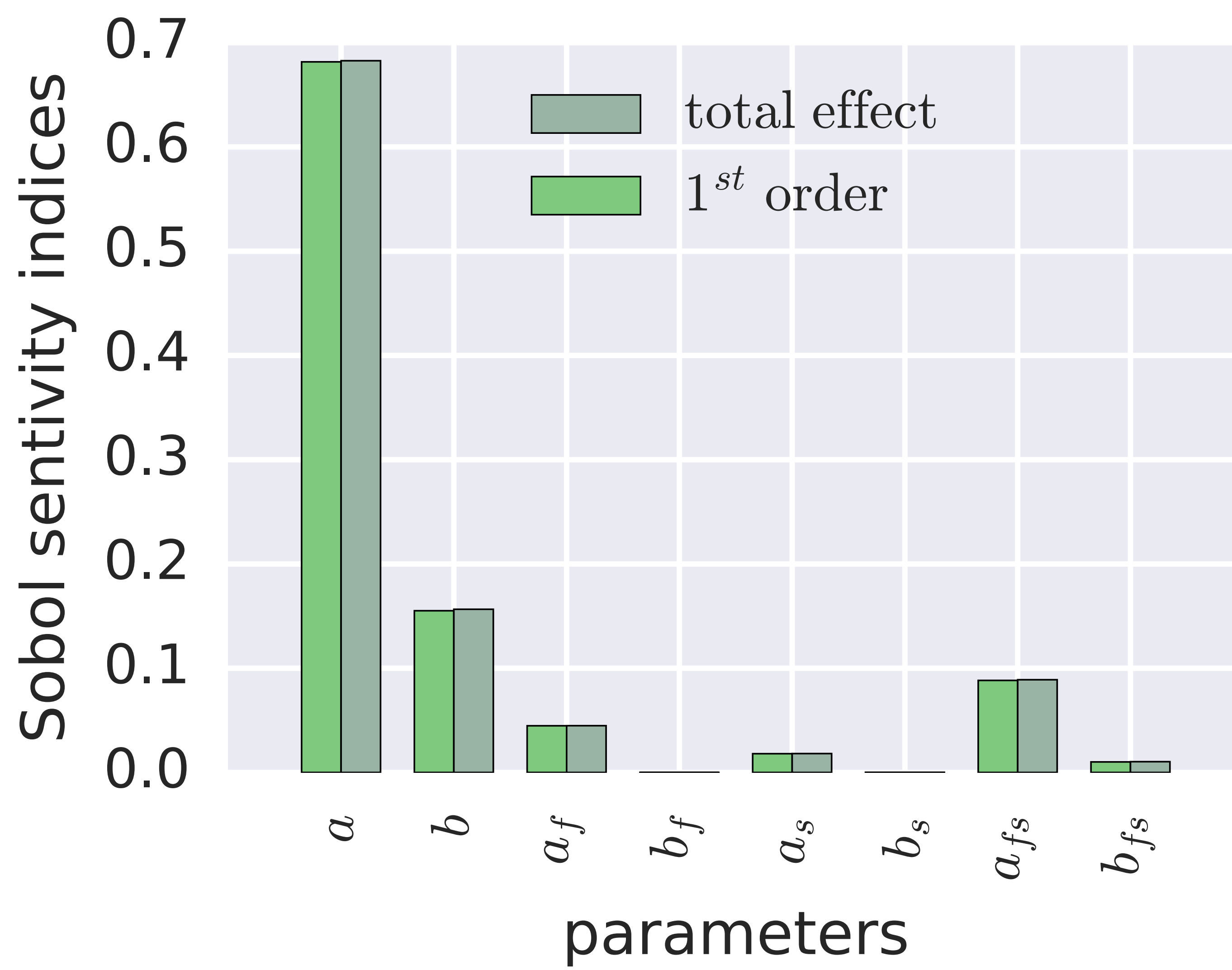


Fig. Center of the sphere: expected value of the displacement in the x direction as a function of Z with a confidence level at 95%.

Global sensitivity analysis (HO model)

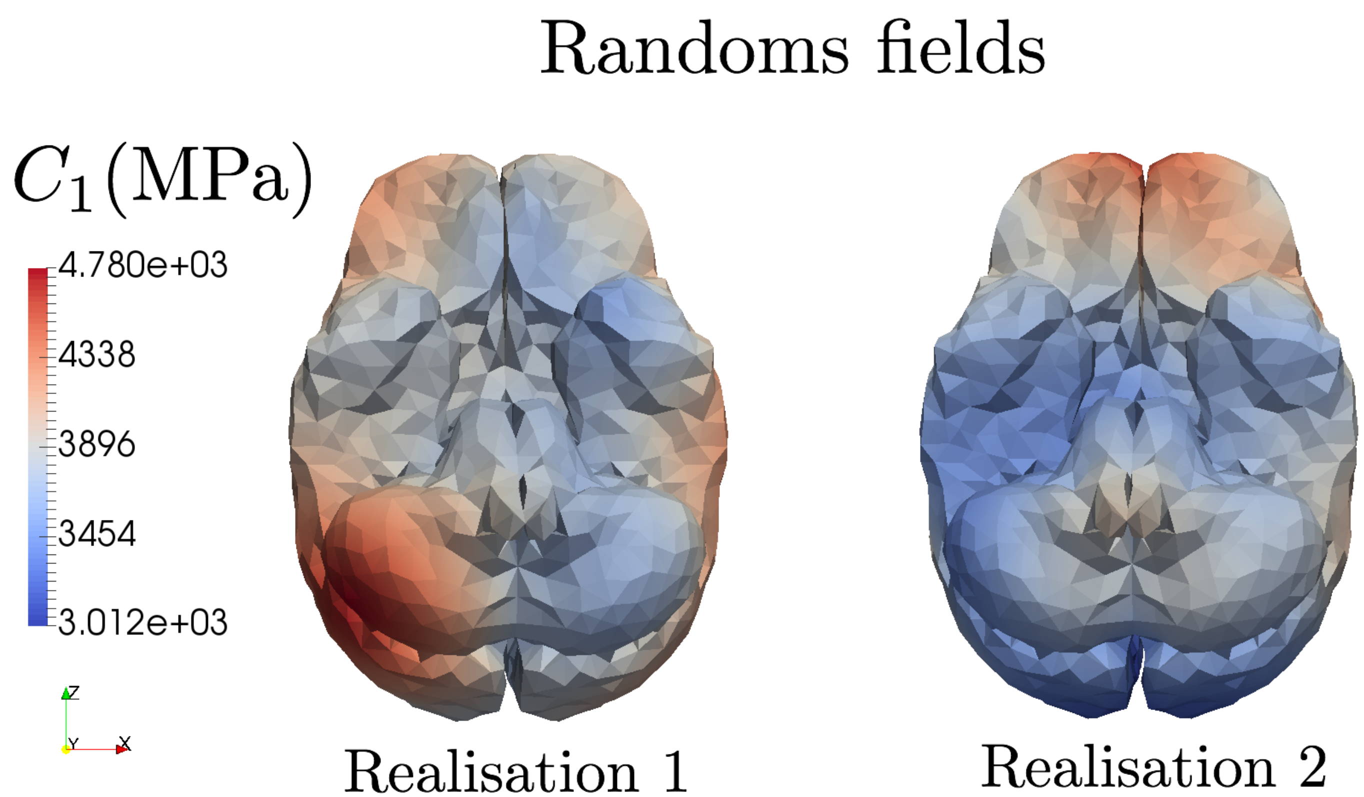
- Sobol sensitivity indices [Sobol 2015, Saltelli 2002]



Quantity of interest: displacement magnitude of the target.

Random Fields

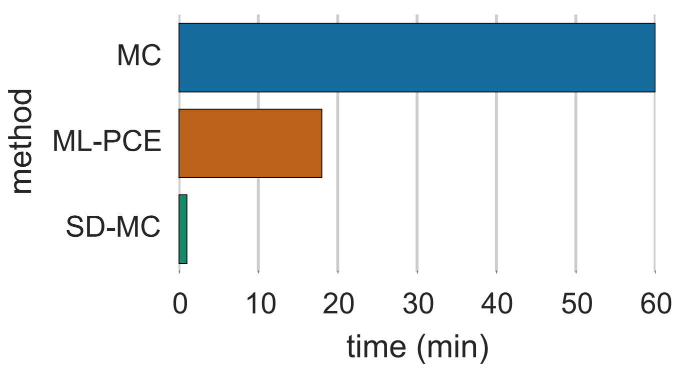
- Different methods: **Karhunen–Loève expansion** [Adler 2007], **Fast Fourier transform** [Nowak 2004], Gaussian random fields with Matérn covariance functions from the **solution of a stochastic PDE** [Lindgren et al. 2011].

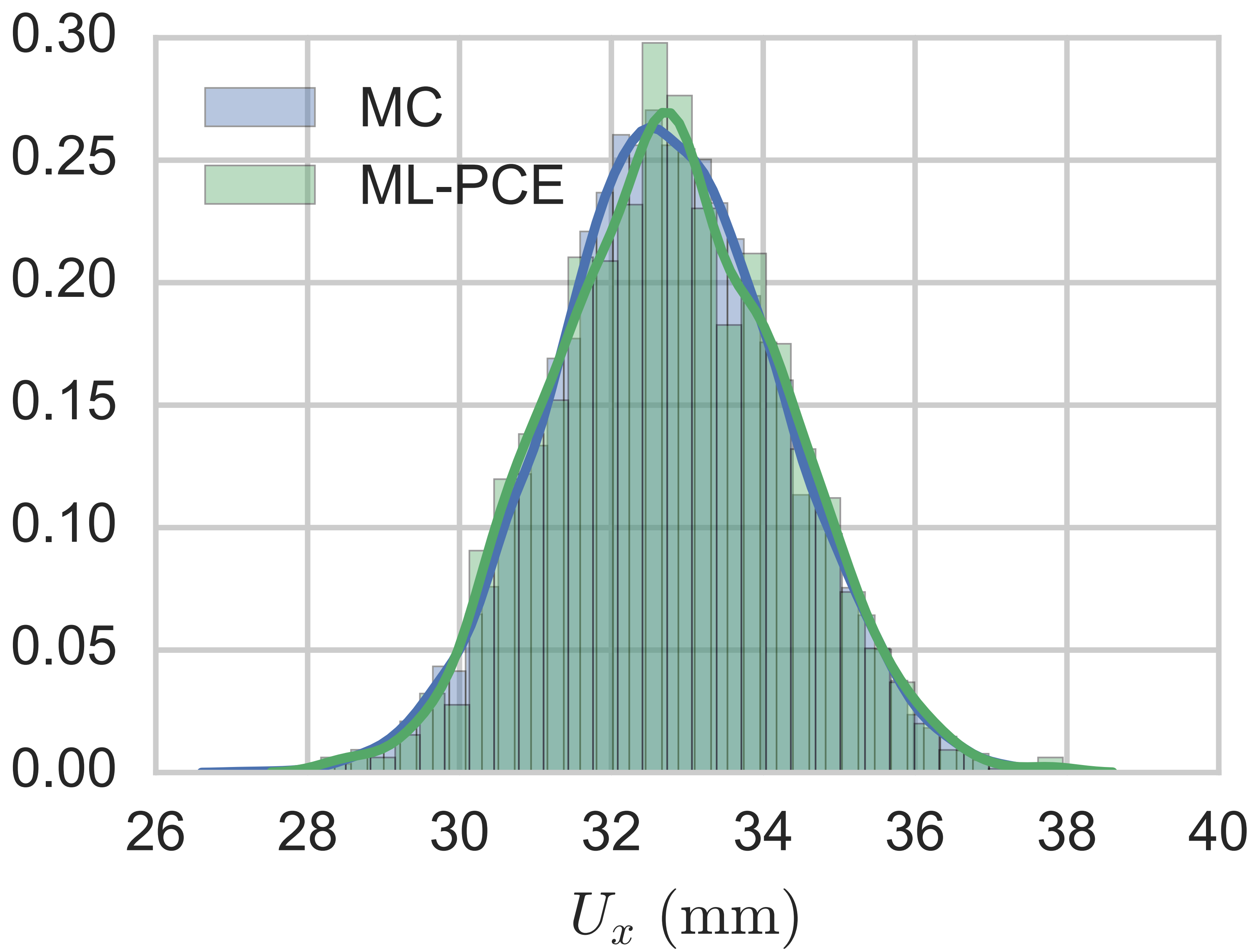


Two realisations of RF, with a log-normal distribution, for the parameter C_1 (in MPa).

Numerical results (Mooney-Rivlin solid)
ML Monte-Carlo technique: ML-PCE

- Monte Carlo method with use of Polynomial Chaos Expansion to improve the convergence [Matthies 2008, Hauseux 2016].





Histogram (MC and ML-PCE methods).

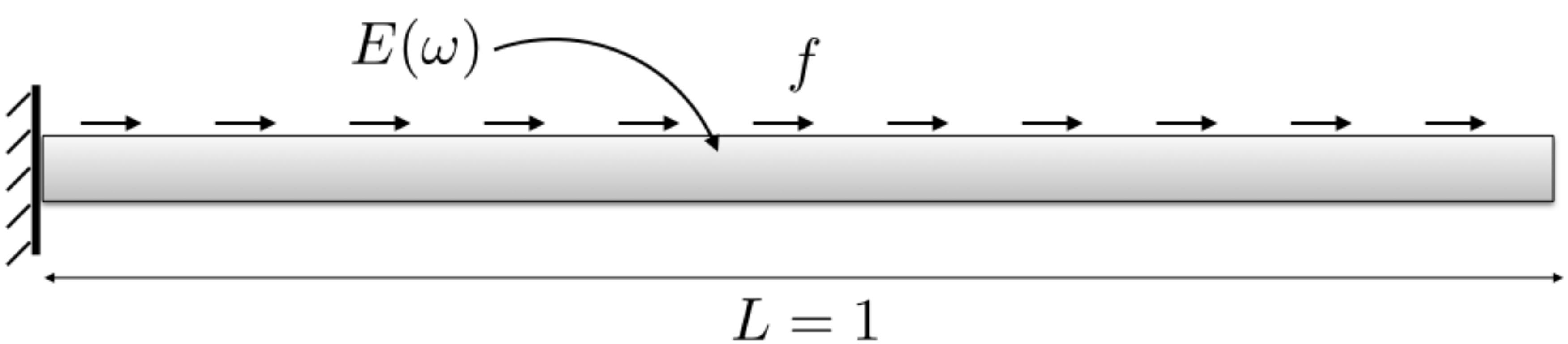
Parameter identification (Inverse Problems in a Bayesian Setting)

- Inverse and forward problems are strongly connected. In a bayesian setting [Matthies et al. 2017], developing methods that reduce the number of evaluations of the forward model to an absolute minimum to achieve convergence is crucial for tractable computations.

- Objective: identify a **PDF** in a Baysesian setting (and not a deterministic constant). We take into account the heterogeneity of the material.

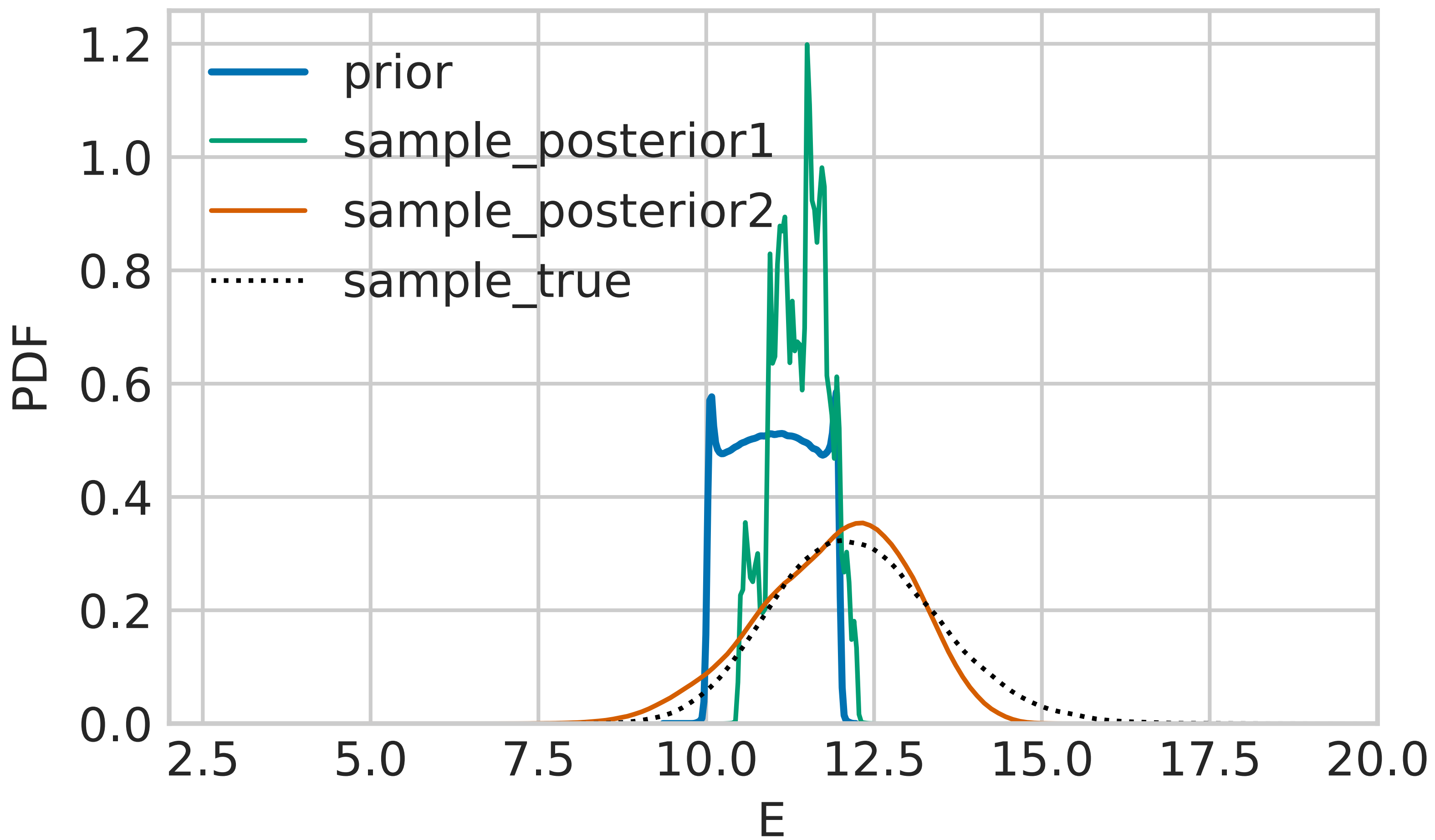
- Linear (LBU) and quadratic (QBU) Baysesian updates (similar to Kalman filter with PCE techniques).

- Didactic example : (1D beam)

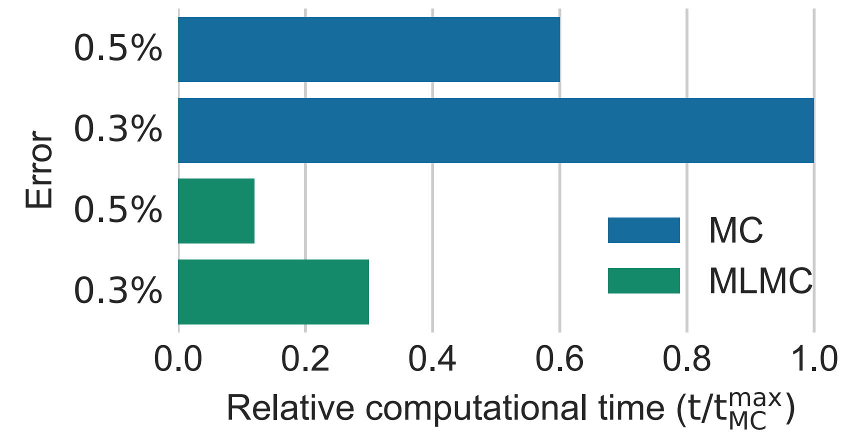


Data: 20 samples, one QoI per sample.

Numerical results: Parameter identification (Inverse Problems in a Bayesian Setting)



Numerical results for the 1D didactic example.
We want to identify the **PDF** of the Young's modulus.



Conclusion

Stochastic modelling:

- Random variables/fields to model parameters with a degree of uncertainty: application to brain deformation.

P*artially-intrusive* Monte-Carlo methods to propagate uncertainty:

- By using sensitivity information (tangent linear approach/adjoint approach) and multi-level Monte Carlo methods we demonstrate that computational workload can be reduced by at least one order of magnitude over commonly used schemes.

- Global and local sensitivity analysis.

Numerical implementation:

- Implementation: DOLFIN [Logg et al. 2012] and chaospy [Feinberg and Langtangen 2015].
- Non-linear hyper-elastic models (Mooney-Rivlin, Neo-Hookean, Holzapfel and Ogden [Holzapfel and Ogden 2009]).
- Ipyparallel and mpi4py to massively parallelise individual forward model runs accros a cluster.

Parameter identification (Inverse Problems in a Bayesian Setting).