Stochastic FE analysis of brain deformation with different hyper-elastic models

Paul Hauseux, Jack S. Hale and Stéphane P.A. Bordas

Thursday, September 22, 2016

UNIVERSITÉ DU LUXEMBOURG

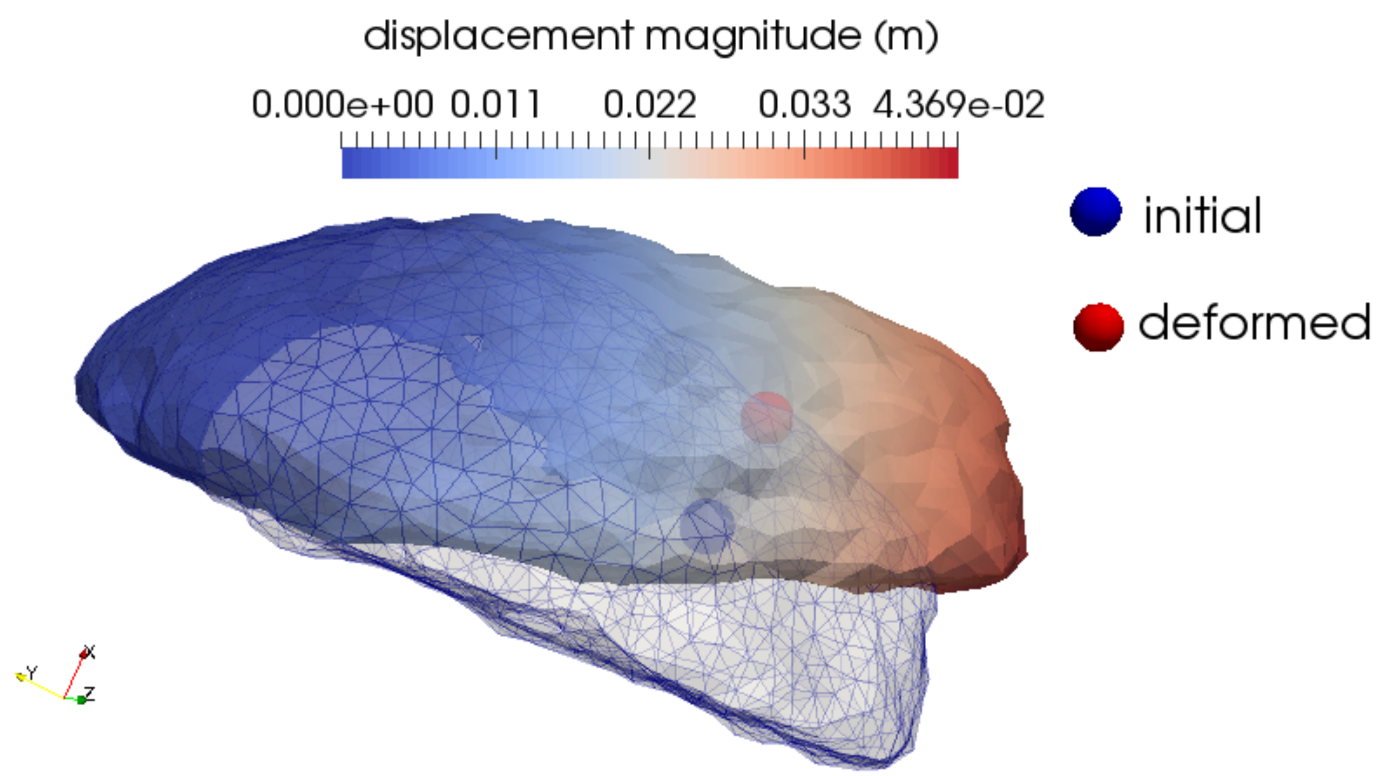
European Research Council
Established by the European Commission

Stg. No. 279578 RealTCut

CMBBE September 20-22 2016, Tel Aviv, Israel

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.



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

General framework

- Stochastic non-linear system: $\boldsymbol{F(u, \omega) = 0}$
- 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, Matthies 2008].

- 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].
- Complex models with only few lines of Python code.

Parallel computing:

- Ipyparallel and mpi4py software tools to massively parallelise individual forward model runs across a cluster and to reduce the workload.

Python package for uncertainty quantification:

- Chaospy [Feinberg and Langtangen 2015] to provide different stochastic objects.

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")
```

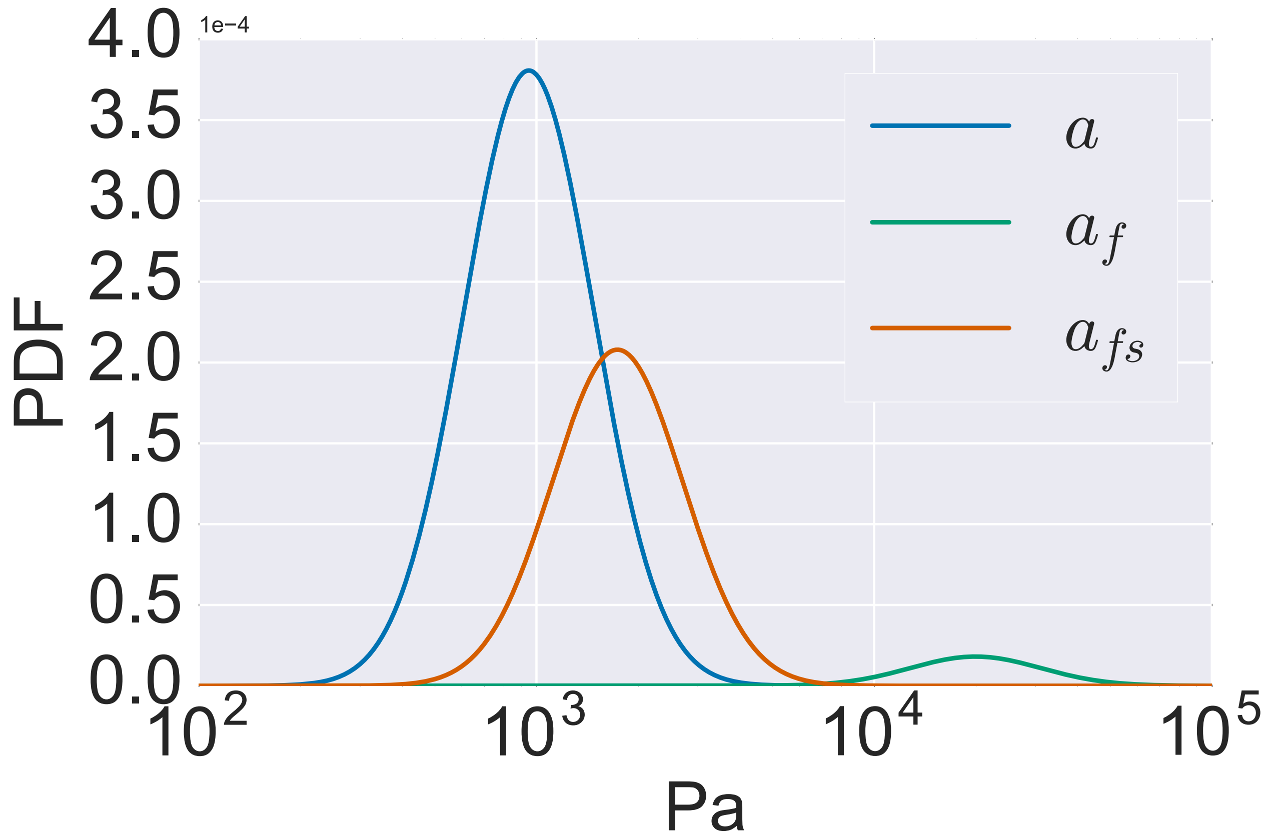
linear system to solve to evaluate du/dm !

Stochastic FE analysis of brain deformation

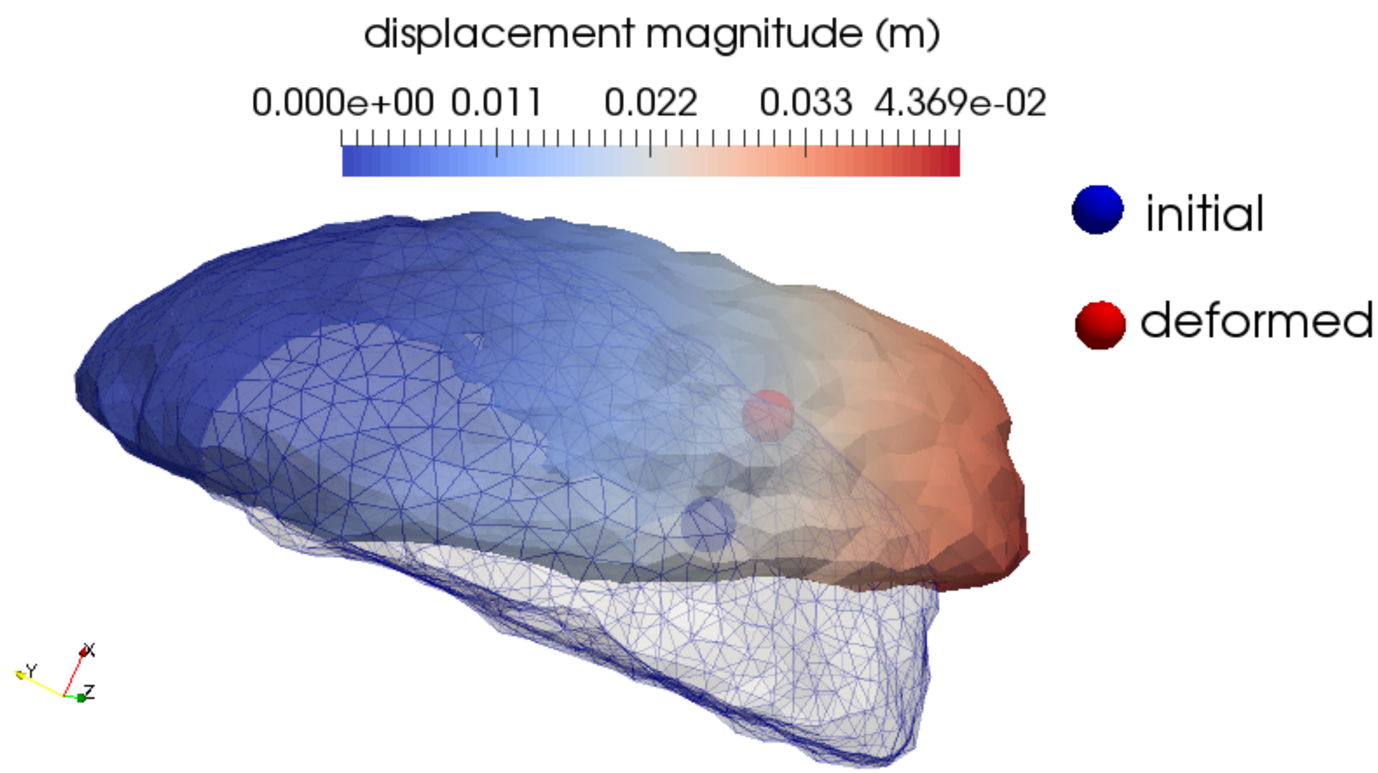
- Different hyper-elastic models implemented (Mooney-Rivlin, Neo-Hookean, Holzapfel and Ogden [Holzapfel and Ogden 2009]).
- Random variables/fields to model parameters [Adler 2007].
- Strain energy function for the Holzapfel and Ogden model:

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

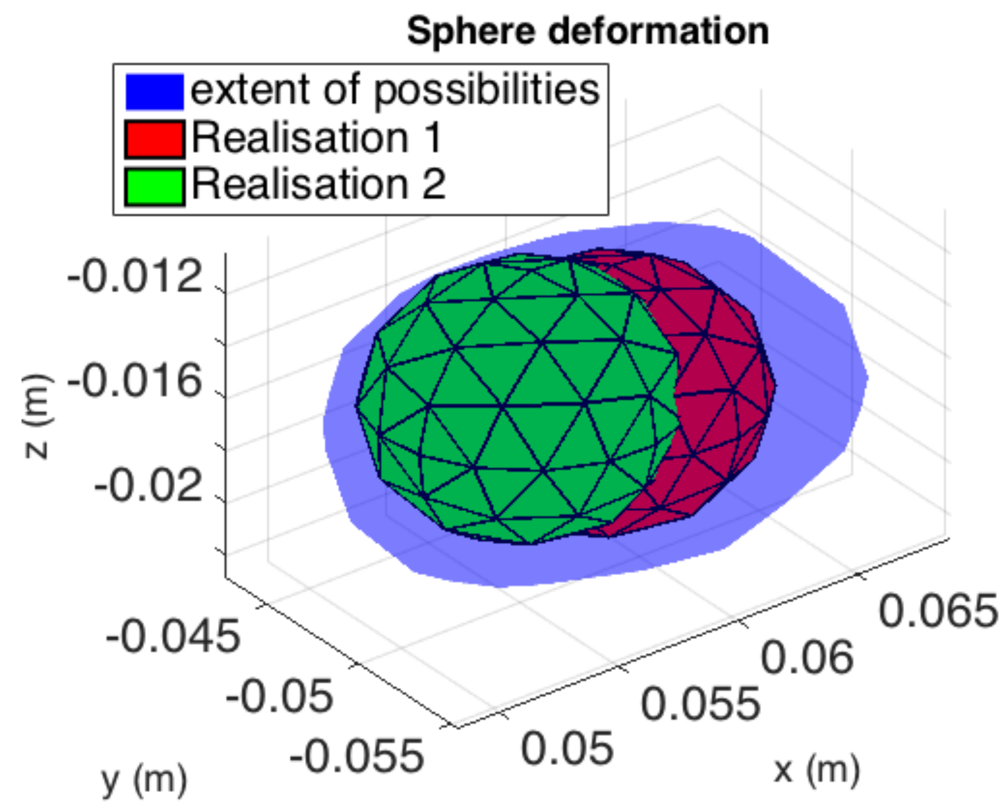
- for example 3RV:



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.

Numerical results: convergence

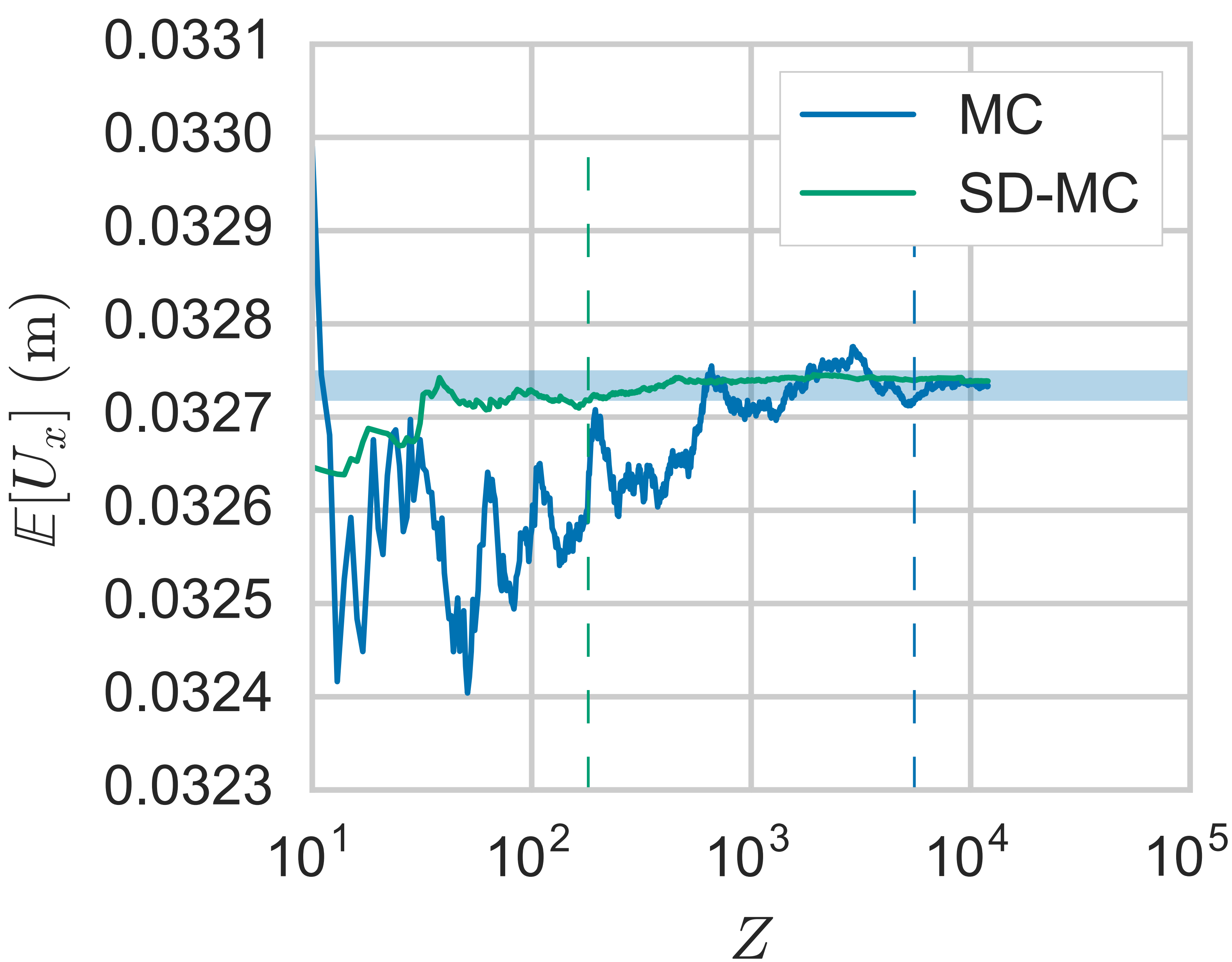
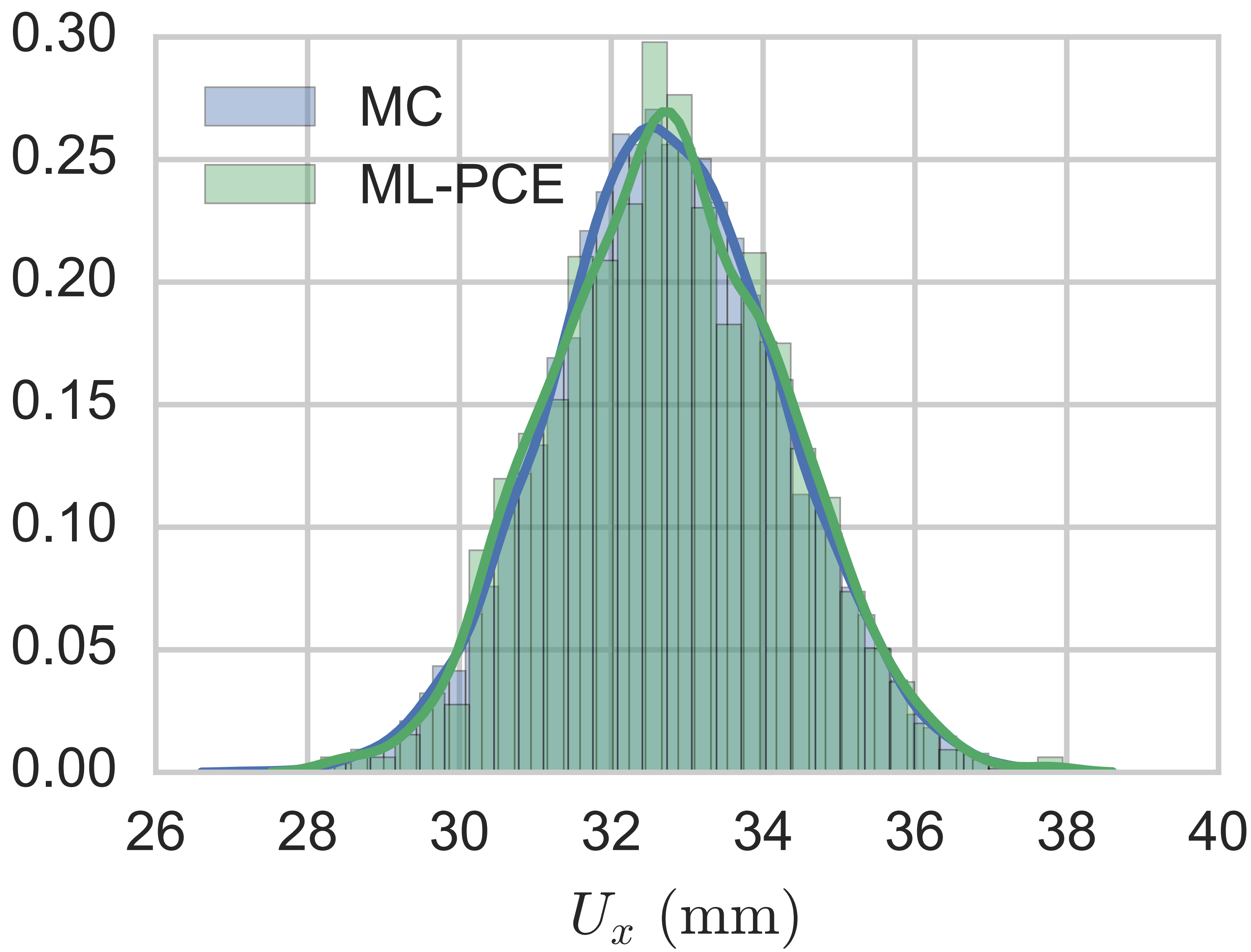


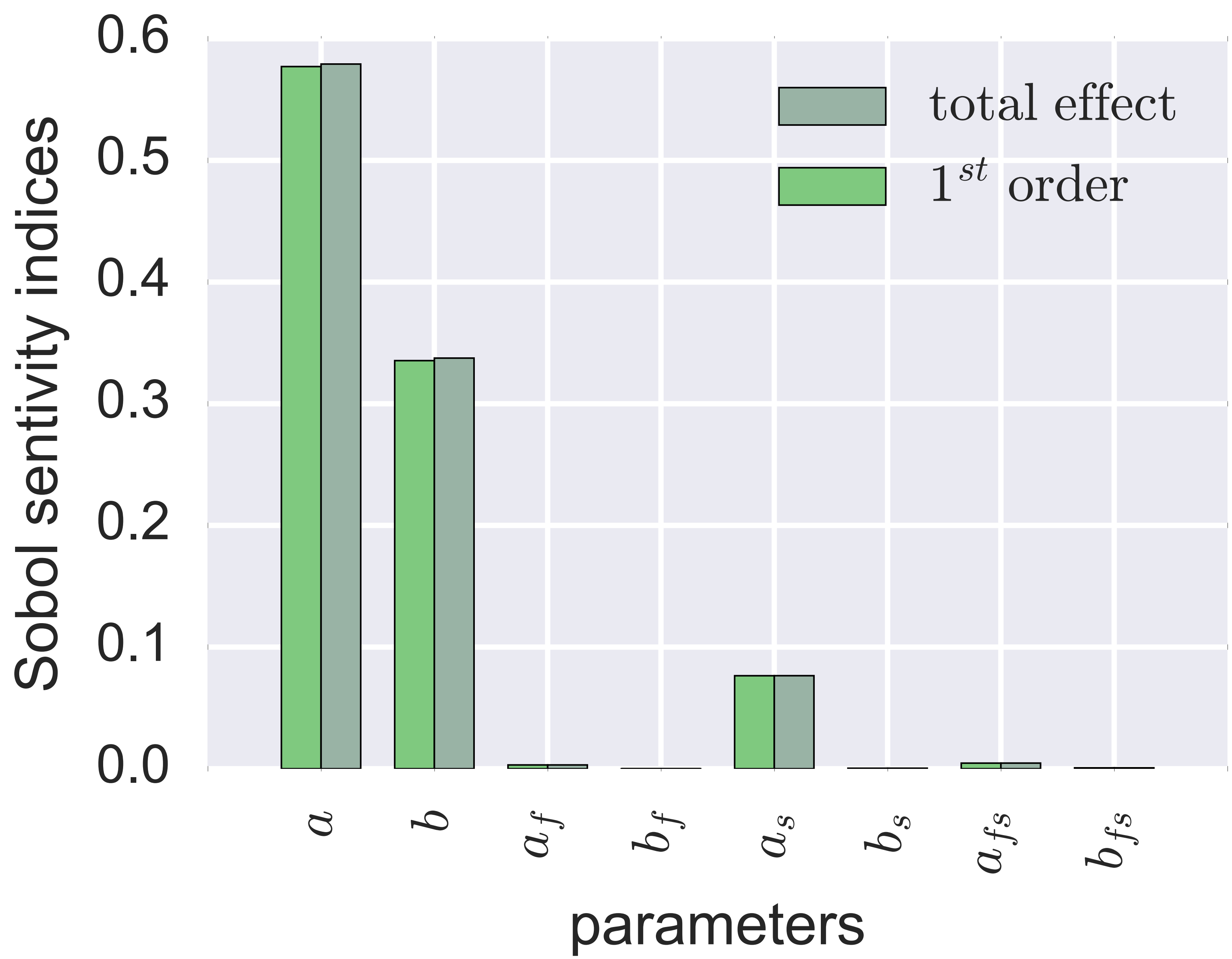
Fig. Center of the sphere: expected value of the displacement in the x direction as a function of Z.

Numerical results (8 RV, Holzapfel model)
ML Monte-Carlo technique: ML-PCE



Histogram (MC and MC-PCE methods).

Global sensitivity analysis



Quantity of interest: displacement of the sphere in the x direction.

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 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.

- 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.