# Propagating uncertainty through a non-linear hyperelastic model using advanced Monte-Carlo methods

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

Wednesday, June 8 2016

UNIVERSITÉ DU LUXEMBOURG

European Research Council
Established by the European Commission

Stg. No. 279578 RealTCut

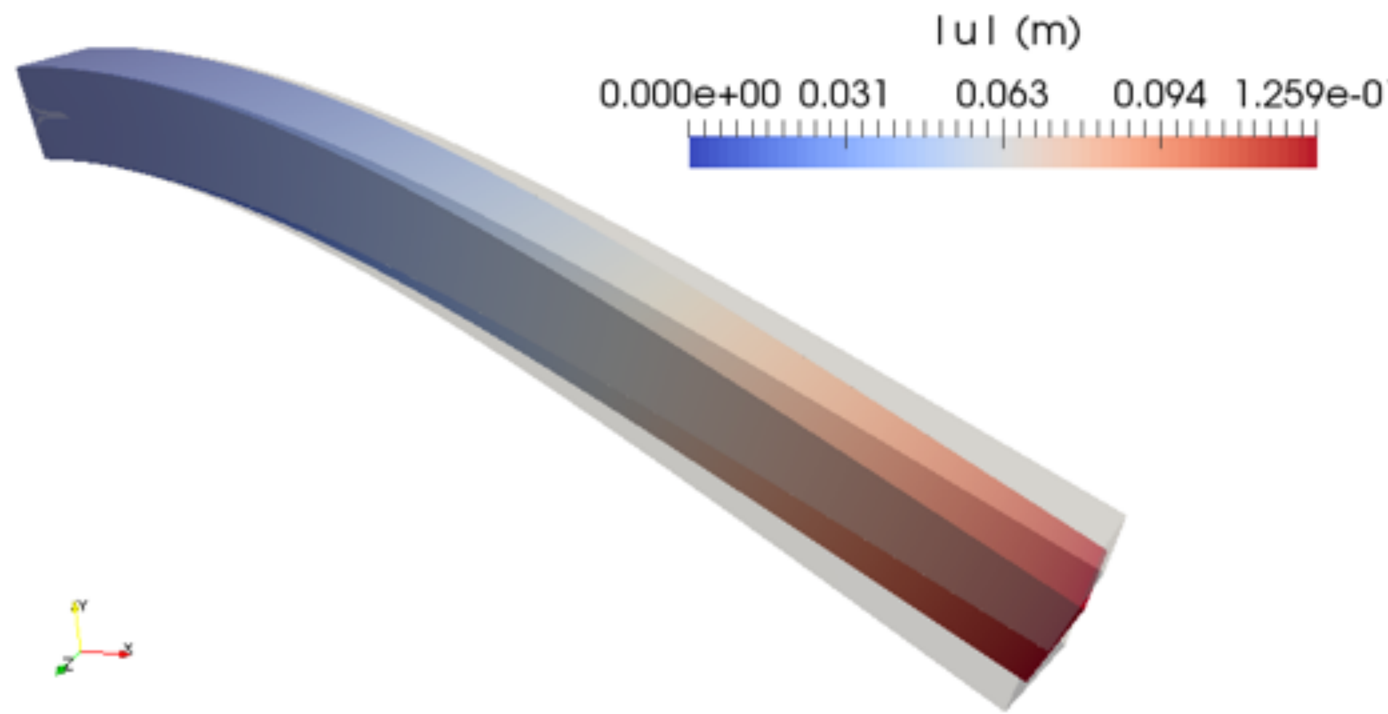
European Congress on Computational Methods in Applied Sciences and Engineering

5-10 JUNE 2016 Crete Island, Greece

# Context

## Soft-tissue biomechanics simulations with uncertainty

- 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

# 1) **Monte-Carlo method**

- A non-linear stochastic system:

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

- 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:

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

# 2) **MC method with use of sensitivity information**

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

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

- 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:

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

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

# 3) **Multi-level MC method with use of PCE**

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

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

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

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

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

```
Solve the deterministic system with average parameters to obtain u^d
k ⟵ 1
while no convergence do
    for  z = 1 to Z do
        Generate ω_z = (ω_1^z, ω_2^z, ..., ω_M^z)
        Generate u^k(ω_z) = F_pce(u^{k-1}(ω_z)) or u^d if k == 1
        Call to deterministic solver to do d (1 or more) iterations with starting values u^k(ω_z)
        and all random parameter function of ω_z
        output: u^k(ω_z) after d iterations
    end for
    Calculate F_pce, the PCE of u^k from Z values of ω_z and u^k(ω_z)
    k = k + 1
end while
```

# 4) **3D Numerical simulations**

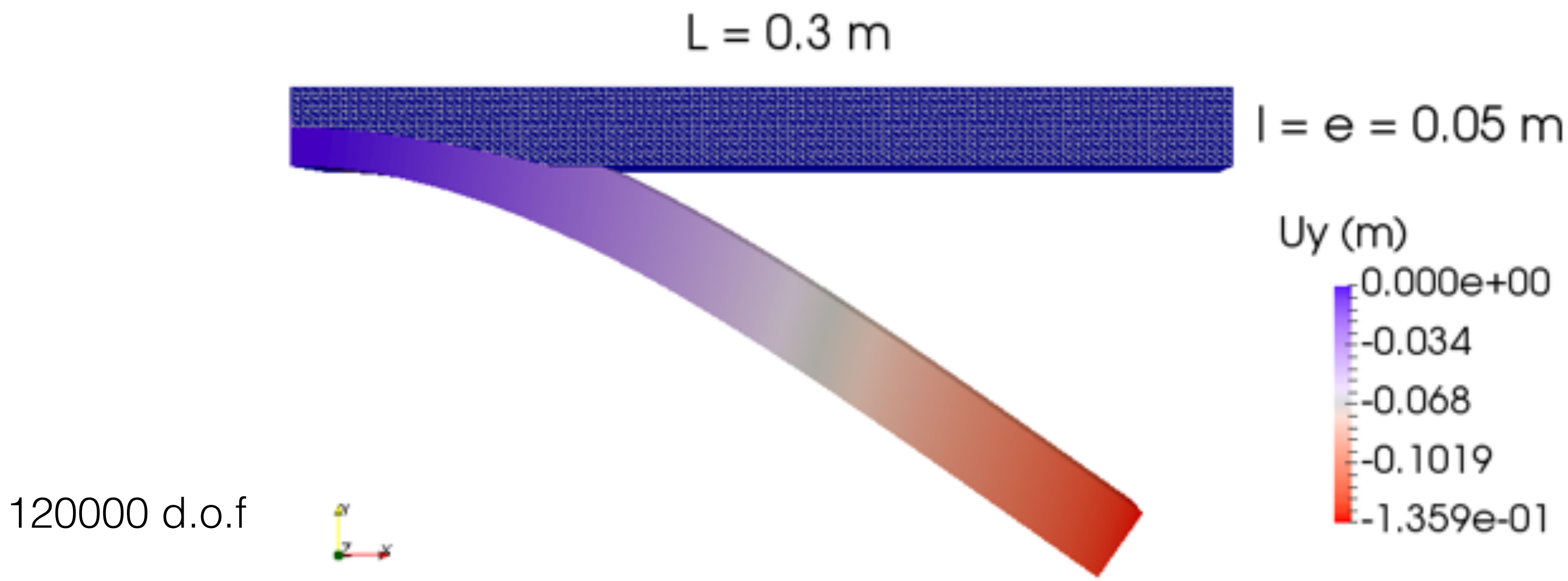



**Fig: Mesh, initial configuration and deformed configuration.**

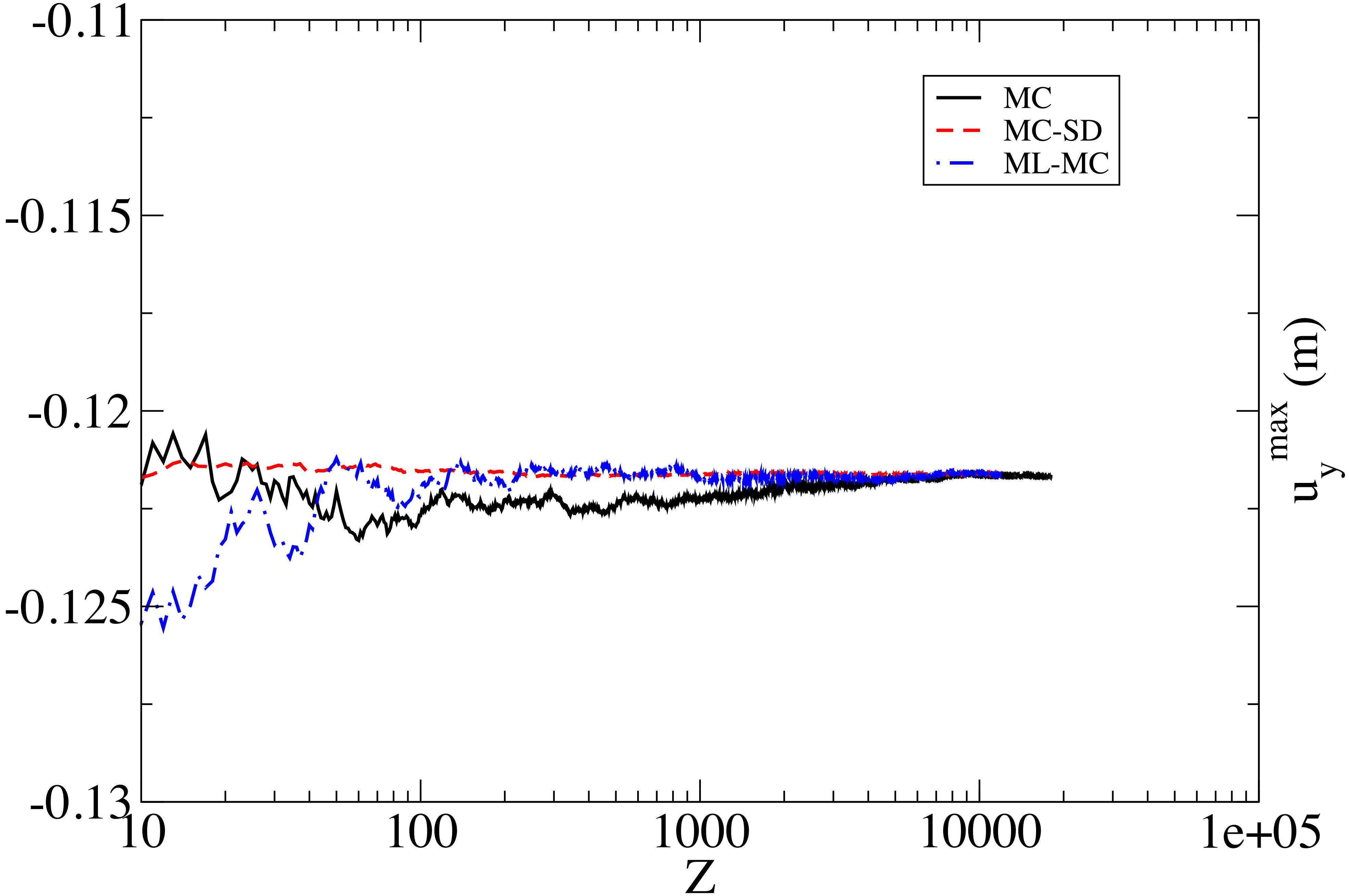
- The stored strain energy density function for a compressible Mooney–Rivlin material:

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

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

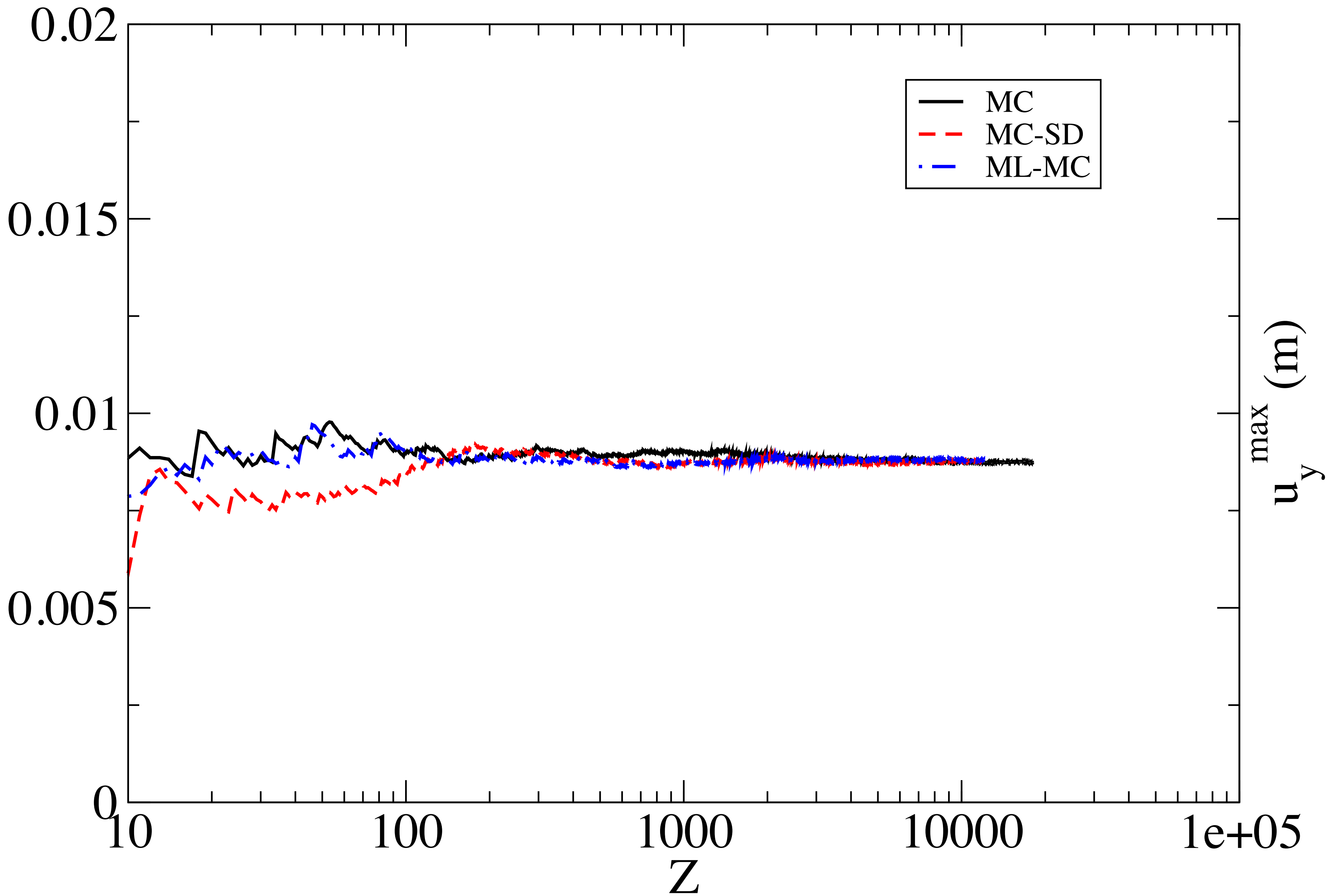
- 2 RV with beta(2,2) distribution: $\begin{aligned} \rho(\omega_1) &= \rho^0(1 + \omega_1/2) \\ D_1(\omega_2) &= D_1^0(1 + \omega_2) \end{aligned}$ $\left\{ \begin{aligned} D_1^0 &= 2 \cdot 10^5 \ \text{Pa} \\ C_2 &= 2 \cdot 10^5 \ \text{Pa} \\ C_1 &= 10^4 \ \text{Pa} \\ \rho^0 &= 600 \ kg/m^3 \end{aligned} \right.$

# 4) **3D Numerical simulations**



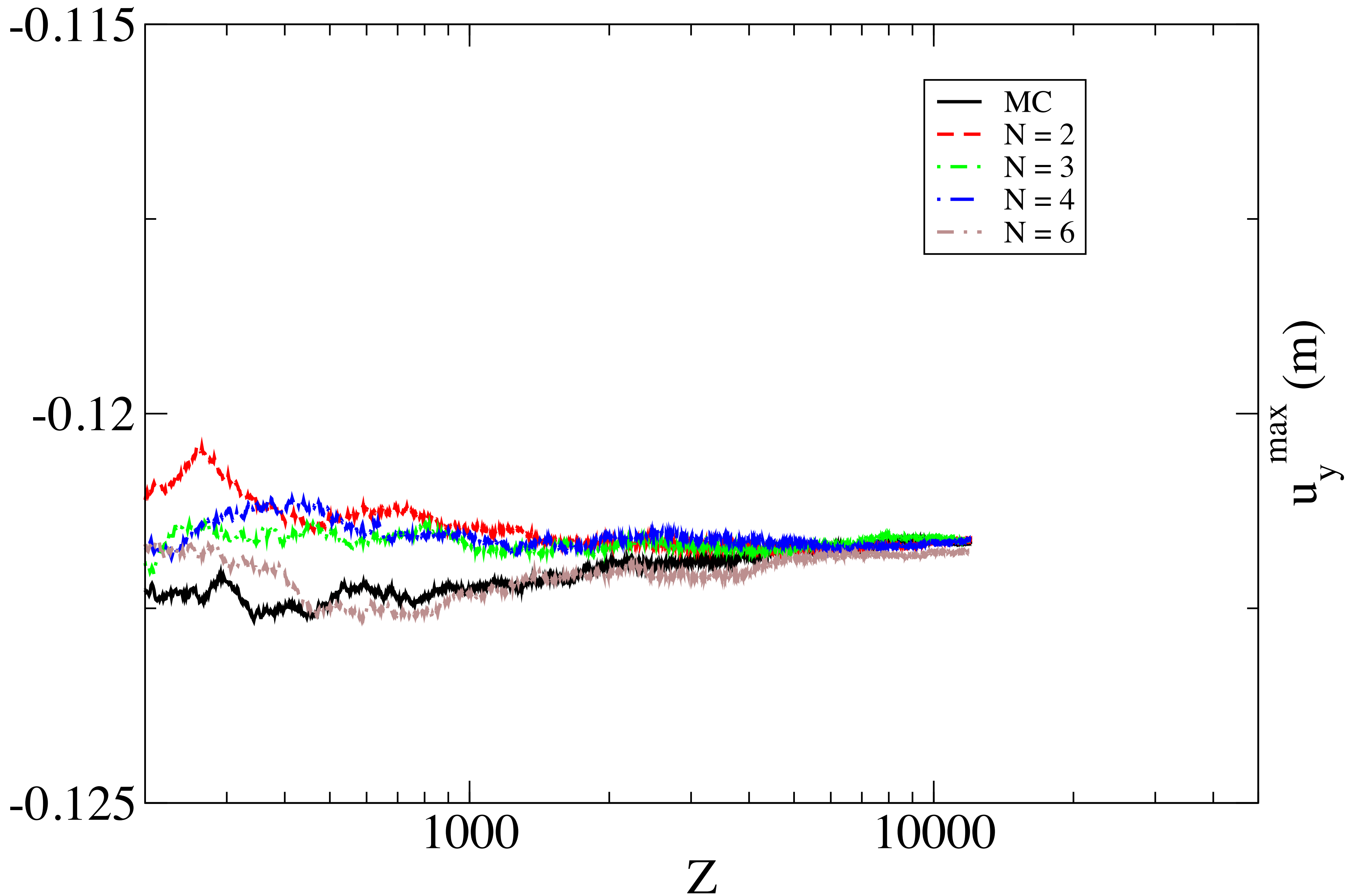

Mean

# 4) **3D Numerical simulations**




Std

# 4) **3D Numerical simulations**




Mean: influence of the number of levels

# 4) **3D Numerical simulations**

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

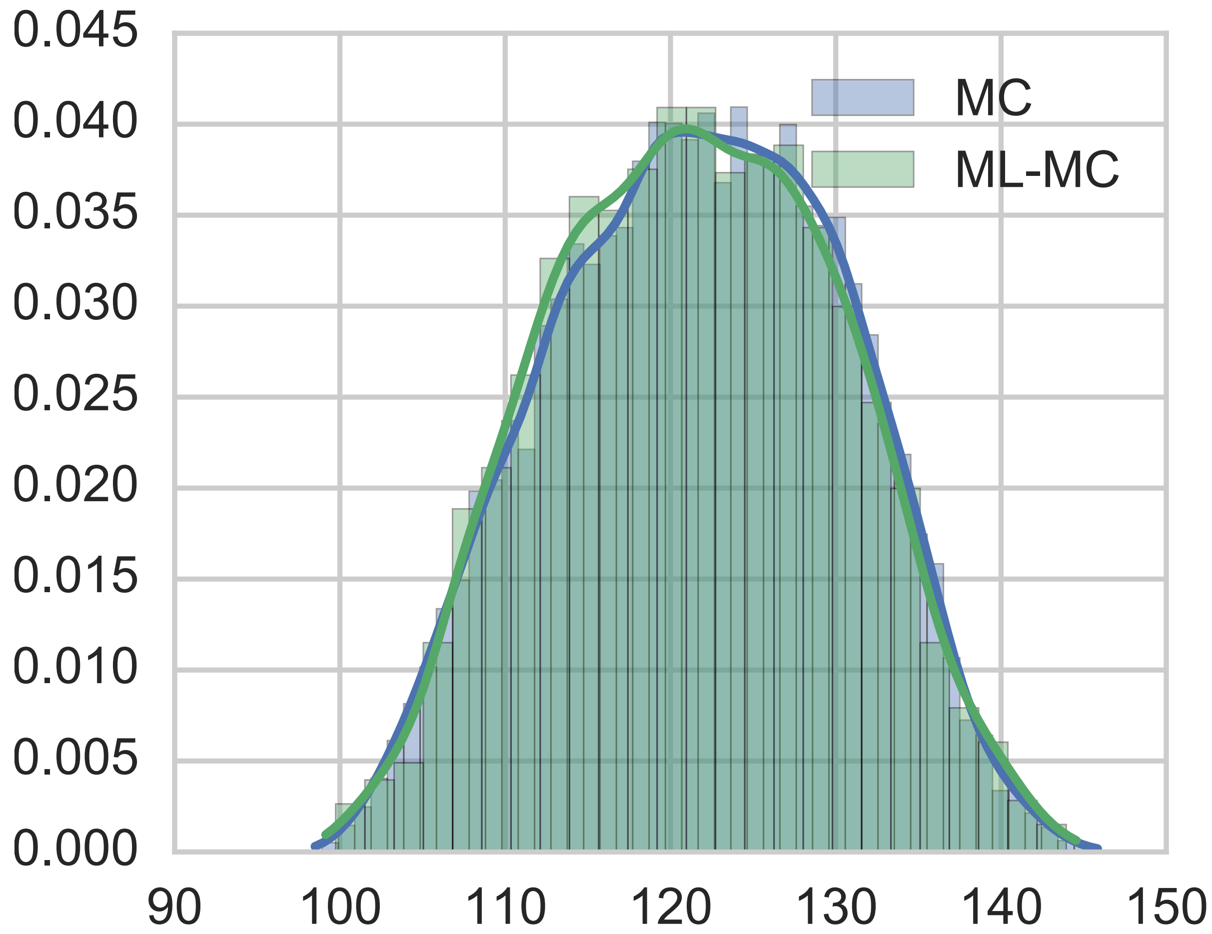
| | $\omega_2$ | $\omega_1$ |
|---|---|---|
| $\left\lVert \frac{d\boldsymbol{u}}{d\omega_i} \right\rVert$ | $1.64$ | $4.36$ |
| $\lvert \frac{du_y^{max}}{d\omega_i} \rvert$ | $0.014$ | $0.036$ |
| $\lvert \frac{du_x^{max}}{d\omega_i} \rvert$ | $0.0081$ | $0.021$ |

**Table 1:** Local sensitivity around mean parameters

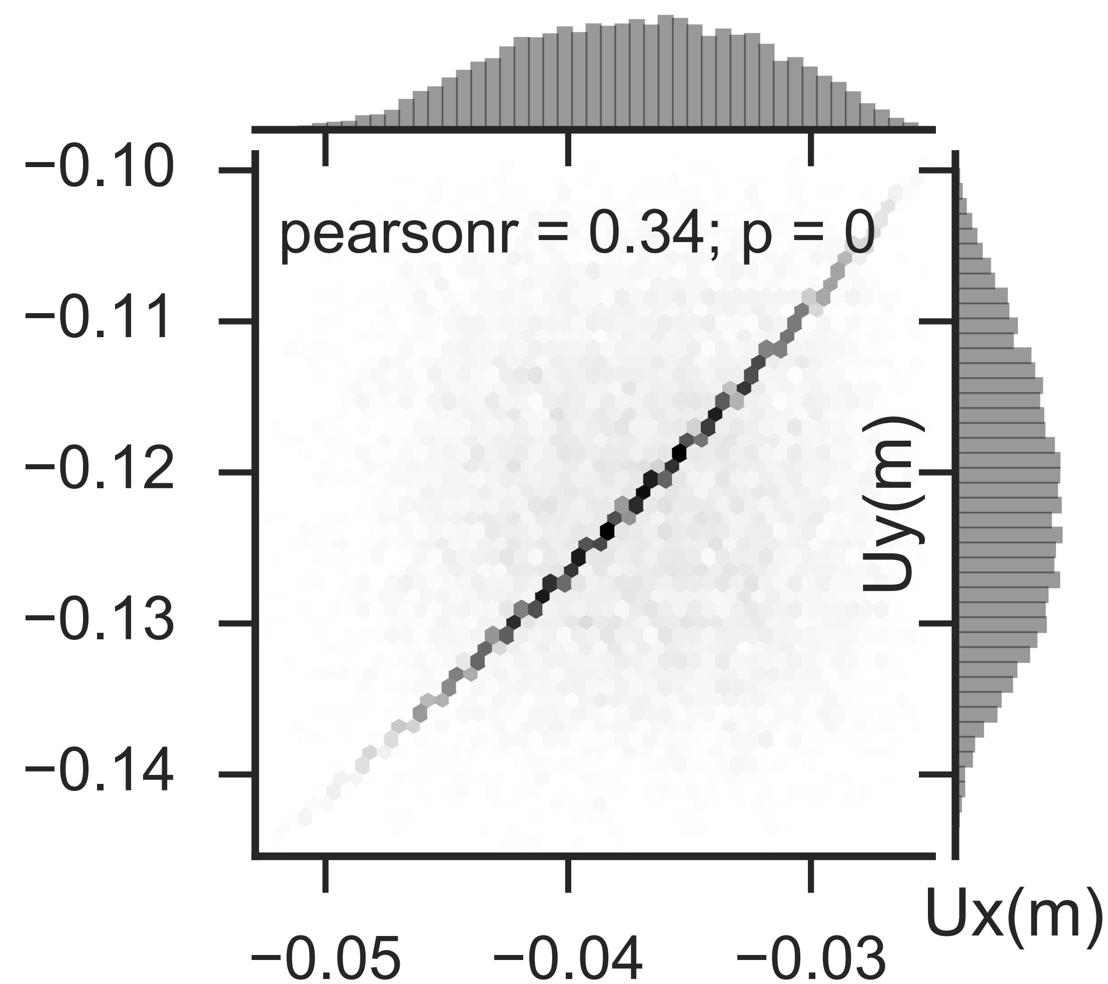
| | $u_x^{max}$ | | $u_y^{max}$ | |
|---|---|---|---|---|
| | $\omega_2$ | $\omega_1$ | $\omega_2$ | $\omega_1$ |
| First order | 0.136 | 0.862 | 0.133 | 0.867 |
| Total effect | 0.138 | 0.862 | 0.133 | 0.868 |

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

# 4) **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).

# Conclusion

- *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