Accelerating Monte Carlo estimation with derivatives of high-level finite element models

Paul Hauseux[a], Jack S. Hale[a], Stéphane P. A. Bordas[a,b,*]

[a] *Research Unit in Engineering Science, University of Luxembourg, Luxembourg.*
[b] *Intelligent Systems for Medicine Laboratory (ISML), School of Mechanical and Chemical Engineering, The University of Western Australia, Australia.*

Abstract

In this paper we demonstrate the ability of a derivative-driven Monte Carlo estimator to accelerate the propagation of uncertainty through two high-level non-linear finite element models. The use of derivative information amounts to a correction to the standard Monte Carlo estimation procedure that reduces the variance under certain conditions. We express the finite element models in variational form using the high-level Unified Form Language (UFL). We derive the tangent linear model automatically from this high-level description and use it to efficiently calculate the required derivative information. To study the effectiveness of the derivative-driven method we consider two stochastic PDEs; a one-dimensional Burgers equation with stochastic viscosity and a three-dimensional geometrically non-linear Mooney-Rivlin hyperelastic equation with stochastic density and volumetric material parameter. Our results show that for these problems the first-order derivative-driven Monte Carlo method is around one order of magnitude faster than the standard Monte Carlo method and at the cost of only one extra tangent linear solution per estimation problem. We find similar trends when comparing with a modern non-intrusive multi-level polynomial chaos expansion method. We parallelise the task of the repeated forward model evaluations across a cluster using the ipyparallel and mpi4py software tools. A complete working example showing the solution of the stochastic viscous Burgers equation is included as supplementary material.

1. Introduction

The importance of modelling the effect of uncertainty in parameters on the output of a system is widely accepted in diverse fields such as geophysics [1], dynamical systems [2], statistical physics [3] and mathematical finance [4, 5]. Uncertainties can be modelled by both random variables or random fields [6, 7]

*Corresponding author.
Email addresses: paul.hauseux@uni.lu (Paul Hauseux), mail@jackhale.co.uk (Jack S. Hale), stephane.bordas@uni.lu (Stéphane P. A. Bordas)

and a significant quantity of research e.g. [8] has been performed to develop efficient numerical techniques to propagate this uncertainty through models and provide a statistical solution.

Commonly used methods to propagate these random variables or fields through models include the large family of Monte Carlo methods [9]. Monte Carlo methods lead to a collection of independent realisations of the forward model to solve. One of the main reasons that these methods are so popular in practice is that they are completely non-intrusive, requiring only an existing deterministic solver. They are also relatively straightforward to implement. The embarrassingly parallel nature of the integral summation problem is relatively easy to exploit. The convergence of the basic Monte Carlo method is guaranteed under very weak assumptions and independent of the dimension M of the parameter space. This latter property is particularly attractive for high-dimensional stochastic problems. However the rate of convergence is slow, on the order of $Z^{-1/2}$ where Z is the number of realisations [9]. Therefore minimising the number of realisations to achieve convergence is critical for achieving acceptable performance.

The core computational kernel of most stochastic algorithms, e.g. Monte Carlo, is the repeated evaluation of the forward model with different parameters. In the context of models that can be described by a system of partial differential equations (PDEs), solved using the finite element method (FEM) [10], it is usually the solution of the deterministic forward model, the map between a single realisation of the stochastic parameter and a quantity of interest in the output of a model, that dominates the overall cost of solving the statistical problem. A key goal then, assuming reasonable efforts have been expended to accelerate the deterministic forward model, is to develop methods that minimise the number of evaluations of the forward model. This is the goal of this paper and many others.

Variance reduction Monte Carlo methods attempt to minimise the number of realisations required to achieve an accurate result. Because the variance of the error of the basic Monte Carlo estimator is proportional to the variance of the quantity of interest, if we can construct enhanced estimators with reduced variance with respect to the basic Monte Carlo, we can achieve lower error with fewer forward model realisations. Note that the underlying $Z^{-1/2}$ convergence rate of the basic Monte Carlo estimator is not improved, but the overall error of the estimator is reduced. Methods in the wide class of variance reduction methods include multi-level techniques [8, 11, 12] and sensitivity derivative techniques [13, 14, 15]. Multi-level Monte Carlo uses a hierarchy of forward models of increasing computational complexity (e.g. uniform spatial grid refinement) and calculates the expectation as being that of the coarsest level, plus a correction based on the difference in expectation between consecutive levels [16]. Sensitivity derivative methods calculate a correction based on the sensitivity of the forward model with respect to the stochastic parameters about the mean parameters. Under certain conditions outlined in the paper of Jimenez et al. [15], and discussed in section 3.2.1 of this paper, the sensitivity-derivative method can lead to variance reduction, and thus improved performance. It is this class

of sensitivity derivative Monte Carlo methods that we study in this paper in the context of propagating uncertainty through non-linear finite element models.

We briefly note the recent interest in using derivative information to accelerate the convergence of Markov Chain Monte Carlo (MCMC) see e.g. [17, 18]. Distinctly from these works, we use derivative information to reduce the variance of the Monte Carlo estimation problem, rather than using the derivative information to improve the quality of the Markov chain that can then be used for Monte Carlo estimation.

We also remark on the difference of the sensitivity derivative Monte Carlo method and local stochastic perturbation methods, e.g. [19]. Local stochastic perturbation methods compute a local approximation to the stochastic problem via a Taylor series expansion around the mean value of the parameters. This perturbation problem is then used as an approximation to the original stochastic problem. In this work we use precisely the same Taylor series expansion but instead of using it to supplant the original stochastic problem we use the local perturbation as a correction to the global stochastic sampling problem.

An alternative to non-intrusive methods are intrusive methods, such as stochastic Galerkin finite element methods (SGFEM) [20, 21, 22, 23]. These methods lead to a directly accessible approximation of the stochastic problem, in contrast with the sampling methods that require the user to specify quantities of interest a priori. These methods are very efficient for linear PDEs with smaller stochastic dimension, but are more complex to implement into existing models due to their intrusive nature [24, 25]. These difficulties are compounded in the case of non-linear forward problems [22].

We do not use this intrusive type of stochastic method in this paper, instead, we advocate a compromise between non-intrusive and intrusive methods; we consider the sensitivity derivative Monte Carlo method to be *partially* intrusive, in the sense that in addition to the usual forward model evaluations we also require that the user has a (preferably efficient) way to compute derivatives with respect to the stochastic parameters. We also note that it is possible to implement SGFEM in a partially intrusive manner with iterative solution techniques, see e.g. [26]. Clearly the whether these requirements actually are intrusive depends on whether the user has models that can provide the required derivative information [14] or matrix-vector actions [26], respectively. In the past, easily computing derivatives of complex forward models either by the tangent linear (forward mode) or adjoint (reverse mode) of differentiation required the use of automatic differentiation tools, e.g. [27] operating on the low-level implementation of the problem. Even with automatic tools, this task can be difficult and error prone. The alternative to automatic differentiation was a complete bottom up implementation of the adjoint model. Recently however, automatic differentiation tools operating on the high-level algebraic description of models, such as dolfin-adjoint [28], have opened up the range of models that can be easily differentiated. Furthermore, unlike the output of many low-level automatic tools, the models produced by dolfin-adjoint are computationally optimal [28]. Because we do not consider time-dependant models in this work we only use the algebraic manipulation tools in the Unified Form Language (UFL) [29] alone, rather than

the complete set of automatic tools for differentiating time-dependant models available in dolfin-adjoint [28].

The key contributions of this paper are as follows; we present a first-order sensitivity derivative driven Monte Carlo method [13, 30, 15] to propagate uncertainty through two typical PDEs from the field of computational mechanics; a simple one-dimensional viscous Burgers equation with uncertain viscosity, and a complex three-dimensional Mooney-Rivlin hyperelastic equation. We concentrate on the low-dimensional case, when the stochastic parameters are real numbers, rather than random fields. In this setting the tangent linear or forward mode of automatic differentiation provides efficient computation of derivatives of the solution field with respect to the stochastic parameters. We automatically derive the tangent linear model from the high-level expression of the variational forms of the finite element models in UFL [29]. We show the superior performance of this method with respect to both the standard Monte Carlo method and a more modern non-intrusive multi-level polynomial chaos expansion method [25, 31] implemented using tools from the Chaospy toolbox [32]. We use the standard Monte Carlo method as a commonly understood baseline for performance comparison. However we remark that the sensitivity derivative method could be used in conjunction with other variance reduction techniques, such as multi-level Monte Carlo [12]. It can also be used with quasi-random low-discrepancy sequences [33], e.g. Sobol sequences, to improve the rate of convergence to $\mathcal{O}\left((\log Z)^M Z^{-1}\right)$ [14].

An outline of this paper is as follows; in section 2 we give the general stochastic problem setting, before continuing to section 3 to specifics on the three stochastic solution methods, namely standard Monte Carlo, sensitivity derivative Monte Carlo and the multi-level polynomial chaos expansion method. In section 4 we discuss the derivation of the tangent linear model from the high-level UFL [29] description of our finite element models. Finally we examine the effectiveness of the sensitivity derivative Monte Carlo method on two examples in section 5.

2. Problem setting

In general we are interested in the propagation of uncertainty through the following canonical form of a non-linear system of equations. Find the solution $u \in \mathcal{U}$ such that:

$$F(u, \omega) = 0, \tag{1}$$

where $\omega \in \mathcal{P}$ are some parameters of the system of equations. For a given realisation of the parameter ω_z we can obtain a solution u to the non-linear system of equations. So we can then write the solution as pure function of the parameters:

$$u := u(\omega). \tag{2}$$

Within the context of this work we restrict ourselves to non-linear systems of partial differential equation discretised using the finite element method. In this case the above non-linear system will be the discrete Galerkin weak formulation

of the residual equation [10]. Typically then the deterministic solution will be in some suitable Hilbert space, e.g. $u \in H^1(\Omega_s)$, where Ω_s is the domain of the PDE, typically space but possibly also time. The deterministic parameters could be a vector of constant scalar coefficients of the PDE, e.g. $\omega \in \mathbb{R}^4$ or a spatially-varying coefficient in some Hilbert space e.g. $\omega \in L^2(\Omega_s)$.

The associated uncertainty propagation problem can then be stated as follows. Consider a probability space $(\Omega_p, \mathcal{F}, P)$ where Ω_p is the sample space, $\mathcal{F}$ is a σ-algebra of subsets of Ω_p and P is a probability measure. Given a quantity of interest on the solution $\psi : \mathcal{U} \to \mathcal{V}$ and a probability density function (pdf) p_ω on the parameter, find the expected value of the quantity of interest:

$$\mathbb{E}\left[\psi\left(u\left(\omega\right)\right)\right] := \int_{\Omega_p} \psi\left(u(\omega)\right)\ dP(\omega). \tag{3}$$

Possible quantities of interest include the expected value of a scalar field solution $u \in H^1(\Omega)$ at a particular point in the domain $x_p \in \Omega_s$, in which case we have $\psi_m : \mathcal{U} \to \mathbb{R}$ given by:

$$\psi_m(u(\omega)) := u(x_p), \tag{4}$$

or for the variance of the solution u at a particular point $x_p \in \Omega_s$:

$$\psi_v(u(\omega)) := (u(x_p) - \mathbb{E}[u(x_p)])^2, \tag{5}$$

$$\mathbb{V}[f] := \mathbb{E}(\psi_v(f)), \tag{6}$$

where we have explicitly denoted the dependence of the solution field $u \in \mathcal{U}$ on the position in space x. For notational convenience, we drop the explicit dependence of the functional ψ on the solution u, and in turn drop the explicit dependence of the solution u on the position x_p, allowing us to write:

$$\psi := \psi(\omega). \tag{7}$$

In the vast majority of cases eq. (3) cannot be solved in closed form and we must resort to numerical methods to evaluate the integral.

3. Stochastic solution methods

In this section we give the specifics on the three stochastic solution methods that we use in this paper.

3.1. Standard Monte Carlo method

We give a brief overview of the standard Monte Carlo method which provides a baseline comparison for the other two methods we consider, see sections 3.2 and 3.3.

3.1.1. Mathematical development

The classical Monte Carlo [33] estimator can be used to evaluate an approximation of the expected value of the quantity of interest in eq. (3):

$$\mathbb{E}\left[\psi(\omega)\right] \approx \mathbb{E}^{\mathrm{MC}}\left[\psi(\omega)\right] := \frac{1}{Z}\sum_{z=1}^{Z}\psi\left(\omega_z\right), \tag{8}$$

where $\{\omega_z\}_{z=1}^{Z}$ is a sequence of independent and identically distributed realisations drawn from the distribution p_ω. Using the Central Limit Theorem [9] it can be shown that the above estimator converges under very weak assumptions to the exact expectation:

$$\left\|\mathbb{E}^{\mathrm{MC}}\left[\psi(\omega)\right] - \mathbb{E}\left[\psi(\omega)\right]\right\|_{L^2(\Omega_p)} \sim \nu\sqrt{\frac{\mathbb{V}[\psi(\omega)]}{Z}}, \tag{9}$$

where ν is a standard normal random variable $N(0,1)$. The above result is a *probabilistic* result on the error induced by the standard Monte Carlo estimator.

3.1.2. Implementation

The classical Monte Carlo method simply involves many evaluations of the quantity of interest for given realisations of the random variables and then taking the mean of those realisations. This is a totally non-intrusive method because it is possible to compute each $\psi\left(\omega_z\right)$ using an existing solver. However, convergence is slow, and often a very large number of realisations are required to obtain an accurate solution.

3.2. Variance reduction Monte Carlo method using sensitivity derivatives

Cao et al. [13] introduced an improved variance reduction estimator based on the use of sensitivity derivatives, or more precisely, the derivatives of the quantity of interest with respect to the parameters evaluated at the mean of the parameters.

3.2.1. Mathematical development

In this section we show the construction of this estimator from first principles and show its variance reduction properties with reference to Theorems 2.1 and 2.3 of [15]. For simplicity of notation, we assume that the quantity of interest is a functional $\psi : \mathcal{U} \to \mathbb{R}$, however, the result can be extended to vector valued functionals reasonably straightforwardly.

Consider a quantity of interest functional $\psi : \mathcal{U} \to \mathbb{R}$ which is at least C^n times differentiable. We write the n-th order Taylor series expansion of the functional ψ about the mean parameter $\bar{\omega} := \mathbb{E}(\omega)$ as:

$$T_n(\omega) = \sum_{k=0}^{n}\frac{1}{k!}D_\omega^k[\psi(\bar{\omega})](\omega-\bar{\omega})^k, \tag{10}$$

where $D^k_\omega[\cdot]$ denotes the usual k-th order derivative with respect to the parameter ω.

Then the sensitivity derivative enhanced Monte Carlo estimator of order n can be defined as [15]:

$$\mathbb{E}_n^{\mathrm{SD-MC}}[\psi(\omega)] := \mathbb{E}[T_n(\omega)] + \frac{1}{Z}\sum_{z=1}^{Z}[\psi(\omega_z) - T_n(\omega_z)]. \tag{11}$$

In this paper we only consider the case $n = 1$, the first-order sensitivity derivative enhanced Monte Carlo method [13]. In that case the Taylor expansion is:

$$T_1(\omega) = \psi(\bar{\omega}) + D^1_\omega[\psi(\bar{\omega})](\omega - \bar{\omega}), \tag{12}$$

and the first order sensitivity derivative estimator (SD-MC) [13] can be written without simplification as:

$$\begin{aligned}\mathbb{E}_1^{\mathrm{SD-MC}}[\psi(\omega)] = \mathbb{E}[\psi(\bar{\omega}) + D^1_\omega[\psi(\bar{\omega})](\omega - \bar{\omega})] \\ + \frac{1}{Z}\sum_{z=1}^{Z}\left[\psi(\omega_z) - \psi(\bar{\omega}) - D^1_\omega[\psi(\bar{\omega})](\omega_z - \bar{\omega})\right].\end{aligned} \tag{13}$$

From this point on we drop the explicit specification of the $k = 1$ on the derivative, e.g. D_ω.

By noting that the term $\mathbb{E}[D_\omega[\psi(\bar{\omega})](\omega - \bar{\omega})]$ is equal to zero, and that the term $\mathbb{E}[\psi(\bar{\omega})] = \psi(\bar{\omega})$ the above equation can be simplified to:

$$\mathbb{E}_1^{\mathrm{SD-MC}}[\psi(\omega)] := \frac{1}{Z}\sum_{z=1}^{Z}\left[\psi(\omega_z) - D_\omega[\psi(\bar{\omega})](\omega_z - \bar{\omega})\right]. \tag{14}$$

It is then possible to show [13, 15] that as $Z \to \infty$ the n-th order estimator is unbiased, that is $\mathbb{E}[\mathbb{E}_n^{\mathrm{SD-MC}}[\psi]] = \mathbb{E}[\psi]$, and also that it converges to the standard Monte Carlo estimator eq. (8).

The key theoretical estimate required is proof that the use of sensitivity derivatives does indeed reduce the variance of the standard Monte Carlo estimator. Theorem 2.1 of Jimenez et al. [15] shows that a bound on the L^2 absolute error of the first-order sensitivity derivative estimator $\mathbb{E}_1^{\mathrm{SD-MC}}$ can be written in terms of the variance of the first-order Taylor remainder of the functional ψ. Defining the first-order Taylor remainder as:

$$R_1(\omega) := \psi(\omega) - T_1(\omega), \tag{15}$$

it is possible to show that [15]:

$$\|\mathbb{E}_1^{\mathrm{SD-MC}}[\psi(\omega)] - \mathbb{E}[\psi(\omega)]\|_{L^2(\Omega_p)} \sim \nu\sqrt{\frac{\mathbb{V}[R_1(\psi(\omega))]}{Z}}. \tag{16}$$

The convergence rate remains at $\mathcal{O}(Z^{-1/2})$. However, comparing eq. (16) with the result for the classical Monte Carlo estimator eq. (9) we can see that we will

have a variance reduction iff:

$$\mathbb{V}[R_1(\psi(\omega))] \leq \mathbb{V}[\psi(\omega)]. \tag{17}$$

Theorem 2.3 of [15] provides a further criterion for such a condition to hold. Specifically, if:

$$\mathbb{V}[T_1(\psi)] \leq 2\mathrm{Cov}(\psi, T_1(\psi)), \tag{18}$$

then the first-order sensitivity derivative estimator will reduce the variance of the standard Monte Carlo method. Intuitively, these results tell us that the sensitivity derivative correction will only improve on the standard Monte Carlo method if the Taylor approximation to the function about the mean is a sufficiently good.

3.2.2. Implementation

The sensitivity derivative estimator eq. (16) can be seen as a simple post-processing treatment of the standard Monte Carlo estimator eq. (8). Just as in the standard Monte Carlo method, a sequence $\{\omega_z\}_{z=1}^{Z}$ of Z realisations of the stochastic variables ω must be drawn from the input distribution p_ω. Then for each realisation ω_z the quantity of interest $\psi_z := \psi(\omega_z)$ must be evaluated, each of which involves a forward model evaluation $u_z := u(\omega_z)$.

In addition to the usual realisations, it is necessary to calculate the derivative of the quantity of interest functional with respect to the parameter, evaluated at the mean of the parameter $D_\omega[\psi(\bar{\omega})]$. We leave the discussion precisely how we calculate the derivatives to section 4. Regardless of how the derivative is calculated, the critical thing to notice is that it is evaluated only once, at the expected value of the parameter $\bar{\omega}$. The derivative can then be cached and used throughout the sampling procedure. We note in advance of the results in section 5 that the cost of running the Z non-linear forward models massively dominates the single evaluation of the derivative at the mean required by the sensitivity derivative method. If the forward model provides the derivative information, the use of the sensitivity derivative estimator comes almost for free.

3.3. Multi-level collocation polynomial chaos expansion collocation method

It is well-known that the standard Monte Carlo method estimator described in section 3.1 is not the most efficient or modern method for solving uncertainty propagation problems of the type given in eq. (3). In this section we give an overview of the construction of a multi-level collocation polynomial chaos expansion (ML-PCE), collocation method as shown in [31, 25] which should be a tougher benchmark for the sensitivity derivative method to match. The multi-level aspect of this method can be seen as a close cousin of the multi-level Monte Carlo methods described in e.g. [8, 12, 16].

3.3.1. Mathematical development

We first discuss the collocation polynomial chaos expansion (PCE) approach before adding in the multi-level (ML) extension.

In the polynomial chaos expansion approach we expand the stochastic solution $u(x,\omega)$ with $\omega \in \mathbb{R}^M$ in a basis $H_\alpha(\omega)$ of polynomials of dimension M and order p [34, 7]:

$$u(x,\omega) = \sum_{\alpha \in \mathcal{J}_{M,p}} c_\alpha(x) H_\alpha(\omega), \tag{19}$$

with $\alpha \in \mathcal{J}_{M,p}$. $\mathcal{J}_{M,p}$ is the set of multi-indices given by $\{\alpha \in \mathbb{N}_0^{(\mathbb{N})} \mid \alpha = \{\alpha_1, \ldots, \alpha_j, \ldots, \alpha_M\}, \alpha_j \in \mathbb{N}_0, |\alpha| = \sum_{j=1}^{M} \alpha_j \leq p\}$. The dimension of $\mathcal{J}_{M,p}$ increases rapidly with M and p:

$$N = \dim(\mathcal{J}_{M,p}) = (M+p)!/(M!\ p!), \tag{20}$$

making this technique suitable for problems of low to moderate stochastic dimension [25].

Different methods exist to calculate the coefficients $c_\alpha(x)$ of the polynomial chaos expansion. We choose to solve for the coefficients using a collocation approach and the solution of a least-squares minimisation problem. This method consists of minimising the square of the Euclidean norm between Z random samples (collocation points) of the solution $U = \left\{\{u(x,\omega_z)\}_{z=1}^{Z}\right\}^T$ and the PCE. With:

$$P = \begin{bmatrix} H_0(\omega_1) & \ldots & H_{N-1}(\omega_1) \\ . & & . \\ . & & . \\ . & & . \\ H_0(\omega_Z) & \ldots & H_{N-1}(\omega_Z) \end{bmatrix} \text{ and } c = \begin{Bmatrix} c_0(x) \\ . \\ . \\ . \\ c_{N-1}(x) \end{Bmatrix}, \tag{21}$$

the objective is to compute the PCE coefficients contained in c and the minimisation leads to the following least-squares problem:

$$c = (P^T P)^{-1} P U. \tag{22}$$

Note that unlike Monte Carlo methods we have a complete approximation of the full stochastic solution and can compute quantities of interest quickly and directly from the PCE expansion.

We now describe the extension of the PCE method with multiple levels (ML-PCE). Let n be the typical number of Newton iterations required to acheive a given tolerance in the Newton method used to solve eq. (1). For the computation of the collocation realisations of the first level of the ML-PCE method, instead of using n non-linear iterations, we choose to compute the solution U using $d_1 < n$ iterations. The output of this procedure is the coefficients of the PCE at the first level. We then proceed to the second level. We now choose to compute

the collocation realisations of the second level using $d_2 < n$ Newton iterations at a *new set* of collocation points. The crucial step is that we initialise the Newton solver of the realisations on the second-level using the output of the PCE algorithm at the first level. We repeat this procedure for K levels. A formal description of the multilevel algorithm is given in algorithm 1.

Algorithm 1 Algorithm for the multi-level polynomial chaos expansion collocation method.

K is the number of levels.
$d = \{d_1, \ldots, d_K\}$ are the number of Newton iterations at each level.
Z is the number of collocation points.
for $k = 1$ to K **do**
  Draw Z collocation points $\omega_z \sim p_\omega$.
  for $z = 1$ to Z **do**
    if k = 1 **then**
      Generate $u^1(\omega_z)$ by solving forward problem with d_1 iterations.
    else
      Generate initial guess for forward problem $u_0^k(\omega_z)$ from PCE calculated at level $k-1$.
      Generate $u^k(\omega_z)$ by solving forward problem with d_k iterations using $u_0^k(\omega_z)$ as initial guess.
    end if
  end for
  Calculate the PCE coefficients for level k from ω_z and $u_k(\omega_z)$.
end for
Output: Coefficients of the multi-level PCE expansion c for level K.

3.3.2. Implementation

We use the methods in the Chaospy toolbox [32] to implement the PCE method and collocation least-squares problem.

4. Automatically deriving tangent linear models

In this section we describe the process by which we derive the tangent linear model using the automatic differentiation features in the Unified Form Language (UFL) [29] of the FEniCS Project [35]. For a full review of techniques that can be used to derive possibly time-dependant tangent linear and adjoint models using the DOLFIN automated finite element solver [36] we again refer the reader to the paper on dolfin-adjoint [28].

We begin with the canonical form of the non-linear system of equations given in eq. (1):

$$F(u, \omega) = 0. \tag{23}$$

Then, taking the total derivative of the above equation with respect to the stochastic parameter ω gives:

$$\frac{\mathrm{d}}{\mathrm{d}\omega}F(u,\omega) = \frac{\partial F}{\partial u}\frac{\mathrm{d}u}{\mathrm{d}\omega} + \frac{\partial F}{\partial \omega}\frac{\mathrm{d}\omega}{\mathrm{d}\omega} = 0, \tag{24}$$

and re-arranging gives the tangent linear model associated with the forward model eq. (1):

$$\frac{\partial F(u,\omega)}{\partial u}\frac{\mathrm{d}u}{\mathrm{d}\omega} = -\frac{\partial F(u,\omega)}{\partial \omega}. \tag{25}$$

In the case that the solution u is discretised using finite elements, leading to a discrete space with size Q and the parameter is a scalar, $\omega \in \mathbb{R}$, the solution Jacobian $\frac{\mathrm{d}u}{\mathrm{d}\omega} := u_\omega$ will be a vector of size Q, the linearised solution operator about the solution u $\frac{\partial F(u,\omega)}{\partial u} := A_u$ will be a matrix size $Q \times Q$, and the right hand side source term $\frac{\partial F(u,\omega)}{\partial \omega} := f_\omega$ will be a vector of size Q. Having solved the non-linear problem eq. (1) at the mean parameter $\bar{\omega}$ to find $u(\bar{\omega})$ we can then solve the following linear system at $u(\bar{\omega})$:

$$A_u u_\omega = -f_\omega, \tag{26}$$

to directly obtain the required derivative information $u_\omega = D_\omega[u(\bar{\omega})]$ for the sensitivity-derivative driven Monte Carlo method. Once we have defined the finite element residual equation in variational form using UFL, we can derive the corresponding variational forms for the tangent linear model with only two function calls that invoke the automatic differentiation techniques implemented in UFL [29].

5. Results

In this section we demonstrate the effectiveness of the sensitivity derivative Monte Carlo method on two example problems; the first problem is a simple one-dimensional problem in fluid mechanics and the second problem a large scale three-dimensional problem from solid mechanics.

5.1. Generalised Burgers equation with stochastic viscosity

In this subsection we solve the stochastic generalised Burgers equation with uncertain viscosity. This example is very similar to the one presented in Liu et al. [30]. We have chosen to replicate a similar result for two reasons; firstly, we wish to independently demonstrate the effectiveness of the sensitivity derivative estimator, and secondly we want to publish a simple and complete working example of the method that can be modified in a relatively straightforward way to the user's time-independent problem if it can be expressed in the Unified Form Language (UFL). A key difference of our implementation compared to that in [30] is the use of the finite element method instead of the finite difference method to solve the forward problem, and the automatic symbolic calculation of the tangent linear problem from the UFL description. We also estimate the

second moment of the solution (and hence calculate the standard deviation) and show an interval estimate of the full stochastic solution. Using the finite element method to solve the forward problem is not required, in the sense that an analytical solution to the forward model is available (although, we might add, not the stochastic problem). Even calculating the finite element method solution takes fractions of a second. However, these properties make it an excellent candidate to verify the correctness of the method. In section 5.2 we present a more challenging three-dimensional hyperelastic problem.

5.1.1. Forward model

The forward model is described by the following weak residual formulation of the generalised Burgers equation with stochastic viscosity $\nu \sim p_\nu$:

Find $u \in H_D^1(\Omega_s)$ such that:

$$F(\nu, u; \tilde{u}) := \int_{\Omega_s} \nu \nabla u \cdot \nabla \tilde{u} - \frac{1}{2} \nabla (u^2 - u) \tilde{u} \, dx = 0 \quad \forall \tilde{u} \in H_0^1(\Omega_s), \tag{27}$$

where the space $H_D^1(\Omega_s)$ is the usual Sobolev space of square-integrable functions on the domain $\Omega_s := [-\frac{1}{2}, \frac{1}{2}]$ with square-integrable weak derivatives that satisfies the Dirichlet boundary conditions:

$$u\left(-\tfrac{1}{2}\right) = u^e\left(-\tfrac{1}{2}, \nu\right), \quad u\left(\tfrac{1}{2}\right) = u^e\left(\tfrac{1}{2}, \nu\right), \tag{28}$$

with u_e the exact analytical solution to the problem:

$$u^e(x, \nu) := \frac{1}{2}\left[1 + \tanh\left(\frac{x}{4\nu}\right)\right]. \tag{29}$$

We take the stochastic viscosity $\nu \sim p_\nu \sim N(0.25, 0.025)$ and the two quantities of interest functionals $\{\psi_1, \psi_2\} : \mathcal{U} \to \mathbb{R}$ to be:

$$\psi_1 := u(0.3), \quad \psi_2 := [u(0.3)]^2, \tag{30}$$

i.e. we estimate the first and second moments of the velocity at a point $x_p = 0.3$ in the domain.

We solve the forward model using a piecewise linear finite element method within DOLFIN [36] on a uniformly-refined mesh with 1024 cells using a standard Newton method with update from u^k to u^{k+1} given by the following linear problem:

$$J(\nu, u^k; \delta u; \tilde{u}) = -F(\nu, u^k; \tilde{u}) \quad \forall \tilde{u} \in H_0^1(\Omega_s), \tag{31a}$$

$$u^{k+1} = u^k + \delta u. \tag{31b}$$

In our implementation we let the automatic differentiation capabilities of UFL derive the Jacobian J from the weak residual eq. (27) on our behalf. We drive the residual equation of the forward problem to a relative error of $\sim 10^{-12}$. This results in a relative error in the forward model numerical solution with respect to the analytical solution of around $\sim 10^{-8}$ in the H^1-norm. The forward model solution can be viewed as exact with respect to the stochastic estimation procedure.

5.1.2. Tangent linear model

For the tangent linear model, which gives the derivative of the solution with respect to the parameter, we solve the following linear variational problem:

Find $u_\nu := D_\nu[u] \in H^1_{D_\nu}(\Omega_s)$ such as:

$$\frac{dF}{d\nu} = \frac{\partial F}{\partial \nu} + \frac{\partial F}{\partial u} \cdot \frac{du}{d\nu} = 0, \tag{32}$$

i.e.,

$$\int_{\Omega_s} \nu \nabla u_\nu \nabla \tilde{u}_\nu - \frac{1}{2} \nabla \left(2u_\nu u - u_\nu\right) \tilde{u}_\nu \, \mathrm{d}x = -\int_{\Omega_s} \nabla u \nabla \tilde{u}_\nu \, \mathrm{d}x \quad \forall \tilde{u}_\nu \in H^1_0(\Omega_s), \tag{33}$$

where $H^1_{D_\nu}(\Omega_s)$ satisfies the Dirichlet boundary conditions given by the derivative of the exact analytical solution with respect to the parameter:

$$u_\nu\left(-\tfrac{1}{2}\right) = u^e_\nu\left(-\tfrac{1}{2}, \nu\right), \quad u_\nu\left(\tfrac{1}{2}\right) = u^e_\nu\left(\tfrac{1}{2}, \nu\right), \tag{34}$$

with:

$$u^e_\nu(x, \nu) := \frac{1}{2} \frac{x}{4\nu^2} \left[\tanh^2\left(\frac{x}{4\nu}\right) - 1\right]. \tag{35}$$

We solve the tangent linear model on the same mesh and using the same linear finite element method as the forward model. Typically we see a relative error in the tangent linear model with respect to the analytical solution of around $\sim 10^{-8}$ in the H^1-norm. The example code contains a file that verifies the correct convergence rate of both the forward and tangent linear models to the exact solutions, but we do not show these results here.

Just like the derivation of the Jacobian for the Newton method solution of the forward model, we let the symbolic differentiation capabilities of UFL derive the tangent linear problem from the specification of the weak residual. This procedure is demonstrated in the example code.

We reiterate the point that the tangent linear model is only solved once at the mean parameter. To calculate the tangent linear solution $u_\nu(\bar{\nu})$, we first solve the forward model at the average parameter to give $u(\bar{\nu})$. Substituting the forward model solution at the mean parameter $u = u(\bar{\nu})$ into the tangent linear model and solving we obtain the desired derivative u_ν.

5.1.3. Estimation

We now proceed to the implementation of the estimation procedure. Unlike the forward and tangent linear problems, we do not have an exact analytical solution for the stochastic problem. So we must estimate the expected values of the quantities of interest $\{\psi_1, \psi_2\}$ using a numerical method. We use Chaospy [32] to draw a quasi-random Sobol sequence on a unit line before mapping the realisations of the sequence to the distribution p_ν using the inverse cumulative density function [32]. Drawing $Z = 200000$ samples from the Sobol sequence gives us the following highly accurate estimates:

$$\mathbb{E}[\psi_1] \simeq \mathbb{E}^S[\psi_1] = 0.646937696525, \tag{36a}$$

$$\mathbb{E}[\psi_2] \simeq \mathbb{E}^S[\psi_2] = 0.418729810487. \tag{36b}$$

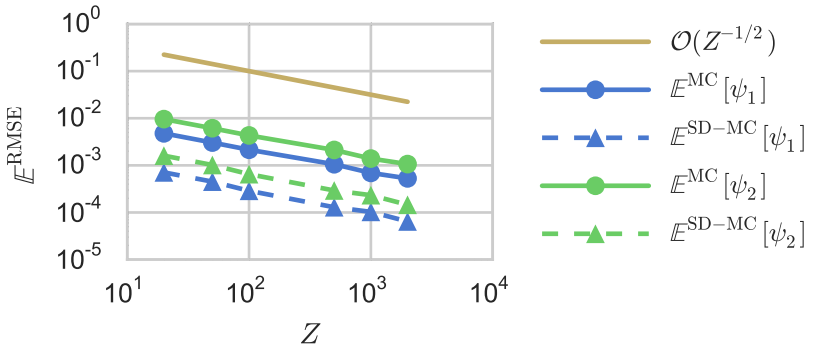


Figure 1: Log-log plot of relative root-mean-square error (RMSE) for standard Monte Carlo E^{MC} and sensitivity-derivative enhanced Monte Carlo methods $E^{\text{SD-MC}}$ on the quantity of interest functions ψ_1 (first moment) and ψ_2 (second moment) evaluated at a domain point $x_p = 0.3$ of the stochastic Burgers equation. Expected convergence rate line of $\mathcal{O}(Z^{-1/2})$ included for comparison. The sensitivity derivative Monte Carlo method provides almost one order of magnitude better convergence than the standard Monte Carlo method at the expense of only one extra linear solve.

The errors of the standard Monte Carlo method and the sensitivity derivative enhanced method are calculated with respect to these two values. We use the exact forward model solution, rather than the approximate finite element solution, to obtain our highly accurate estimates.

Because the output of an Monte Carlo estimator is itself random, to get sensible error estimates we must take an estimate of the output of the underlying estimator. We assess the performance of a given estimator on a quantity of interest by calculating the relative root-mean-squared error (RMSE):

$$\mathbb{E}^{\text{RMSE}}[\mathbb{E}[\psi]] = \sqrt{\frac{\mathbb{E}^{\text{MC}}[(\mathbb{E}[\psi] - \mathbb{E}^{S}[\psi])^2]}{\mathbb{E}^{S}[\psi]^2}}. \tag{37}$$

For a given estimator, set to use a fixed number of realisations (we select $Z = \{20, 50, 100, 500, 1000, 2000\}$), we choose to take $Z = 100$ realisations of said estimator to evaluate the RMSE. We show the results of the RMSE convergence study in fig. 1. The sensitivity derivative Monte Carlo method produces an estimator with an RMSE error one order of magnitude lower than the standard Monte Carlo method. Both estimators acheive the expected convergence rate of $\mathcal{O}(Z^{-1/2})$. Taking the estimate on the first moment as an example, $\mathbb{E}(\psi_1)$, to achieve an error of $\sim 10^{-3}$ we require $Z \sim 20$ non-linear forward model solves using the sensitivity derivative Monte Carlo method, versus around $Z \sim 1000$ non-linear forward model solves model evaluations for the standard Monte Carlo method.

To close up our discussion on the stochastic Burgers equation, we show the solution evaluated at 20 evenly spaced points in the domain in fig. 2 using the

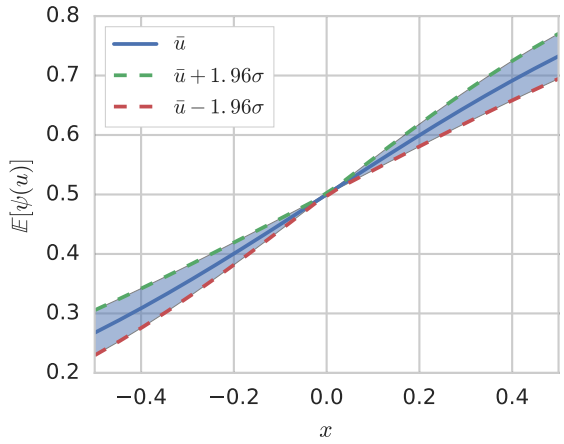


Figure 2: Solution of stochastic Burgers equation with stochastic viscosity $\nu \sim N(0.25, 0.025)$. Plot shows mean solution with $\pm 1.96\sigma$ standard deviation envelope denoted by shaded region. This figure was generated using the sensitivity derivative Monte Carlo estimator with $Z = 2000$ forward model evaluations and one tangent linear model evaluation.

sensitivity derivative Monte Carlo method. The quantity of interest functionals in this case are simply the ones shown in eq. (36) but evaluated at multiple points in the domain. The Monte Carlo method gives us the sampled mean and interval estimate that respects the $u(0, \nu) = 0 \ \forall \nu$ condition given by the analytical solution.

Note that in this example where we have quantities of interest in many positions in the domain we still only need to solve one tangent linear problem as we can get the sensitivities of the solution field $u \in H^1(\Omega_s)$ with respect to the scalar parameter $\nu \in \mathbb{R}$ with just one linear solve.

We remark that if we had multiple parameters, e.g. $\omega \in \mathbb{R}^M$ we would need to solve M tangent linear problems to have the required sensitivity information for the estimator. If the parameter was instead a random field $\omega \in L^2(\Omega_p)$, which when discretised can be viewed as $\omega \in \mathbb{R}^m$ with very high M, then clearly these M linear solves would be prohibitively expensive. In this high-dimensional case, it would make more sense to select a functional quantity of interest $\psi : \mathcal{U} \to \mathbb{R}$ and use the adjoint, or reverse mode of automatic differentiation, which scales excellently with increasing m [28], to calculate the sensitivity with respect to all M parameters with one just one linear solve. Current work includes the use of the adjoint mode of differentiation in the context of sensitivity derivative Monte Carlo estimators for random fields.

5.2. Hyperelasticity equation with stochastic material parameters

In this subsection we solve the stochastic Mooney-Rivlin hyperelasticity equation. The deterministic version of this model is often used in biomechanics

simulations to describe the behaviour of soft-tissue. However, in many soft-tissue biomechanics simulations the material parameters used in the definition of a given simulation often have a significant degree of uncertainty associated with them. This motivates the use of the stochastic version where uncertainty in the parameters can be taken rigorously into account. Taking into account uncertainties in biomechanics models is a relatively young field and little work has been done in particular for soft tissue with large deformation [37]. In a clinical environment, where safety-critical decisions must be made based on the output of such simulations, being able to propagate uncertainty efficiently through such models is of importance.

We repeat the essential features of the Mooney-Rivlin hyperelastic model here. Interested readers are referred to [38] for more details. In this section we use bold to denote vector fields, e.g $\mathbf{X} \in \mathbb{R}^3$ and capital sans-serif to denote tensors e.g. $\mathsf{F} \in \mathbb{R}^{3\times 3}$.

Consider a continuum body $\mathfrak{B}$ with reference configuration $\chi_0 : \mathfrak{B} \to \mathbb{R}^3$ and an unknown (deformed) configuration $\chi : \mathfrak{B} \to \mathbb{R}^3$. The domain occupied by the body in Euclidean space $\mathbb{R}^3$ is then $\Omega_0 := \chi_0(\mathfrak{B})$ and $\Omega := \chi(\mathfrak{B})$ in the reference and deformed configurations, respectively. Material points of the continuum body $\mathfrak{p} \in \mathfrak{B}$ are mapped similarly, giving $\mathbf{X} = \chi_0(\mathfrak{p})$ and $\mathbf{x} = \chi(\mathfrak{p})$ in the reference and deformed configurations, respectively.

We can define the deformation between the points in undeformed and deformed configurations as a map:

$$\boldsymbol{\varphi} = \chi \cdot \chi_0^{-1}, \qquad \boldsymbol{\varphi} : \Omega_0 \ni \mathbf{X} \to \mathbf{x} \in \Omega, \tag{38}$$

and then assuming the map is sufficiently differentiable, we define the deformation gradient tensor as:

$$\mathsf{F}(\mathbf{X}) := \frac{\partial \boldsymbol{\varphi}}{\partial \mathbf{X}}. \tag{39}$$

We furthermore assume that the boundary of the reference domain Ω_0 can be divided into two open subsets $\partial_D\Omega_0$ and $\partial_N\Omega_0$ on which Dirichlet (displacement) and Neumann (traction) boundary conditions are applied, respectively.

Based on the definition of the deformation gradient tensor F we define the left Cauchy-Green tensor as:

$$\mathsf{C} := \mathsf{F}^T\mathsf{F}. \tag{40}$$

We can then write the Mooney-Rivlin energy density functional $\mathcal{W} : \mathbb{R}^{3\times 3} \times \Omega_0 \to \mathbb{R}$ as [39, 40]:

$$\mathcal{W}(\mathsf{C}, \mathbf{X}) := C_1(\bar{I}_1 - 3) + C_2(\bar{I}_2 - 3) + D_1(I_3^{1/2} - 1)^2, \tag{41}$$

where $\{C_1, C_2, D_1\}$ are possibly random material parameters which we assume to be spatially homogeneous in this work, and the modified first and second invariants are:

$$\bar{I}_1 := J^{-2/3} I_1, \tag{42a}$$

$$\bar{I}_2 := J^{-4/3} I_2, \tag{42b}$$

where I_1, I_2 and I_3 are the unmodified invariants of C given by:

$$I_1(\mathsf{C}) := \operatorname{tr}(\mathsf{C}), \tag{43a}$$

$$I_2(\mathsf{C}) := \frac{1}{2}\left((\operatorname{tr}\mathsf{C})^2 - (\operatorname{tr}\mathsf{C}^2)\right), \tag{43b}$$

$$I_3(\mathsf{C}) := \det(\mathsf{F})^2 = J^2. \tag{43c}$$

Defining the displacement field as $\mathbf{u} := \boldsymbol{\varphi} - \mathbf{X}$ and a linear functional $\mathbf{f}$ that encodes the external loading (surface tractions, body forces etc.) we can characterise the elastic equilibrium displacement field $\mathbf{u}^*$ as the solution to the following minimisation problem:

$$\begin{aligned}\mathbf{u}^* &= \underset{\mathbf{u}\in[H^1_D(\Omega_0)]^3}{\arg\min}\ L(\mathbf{u}) \\ &= \underset{\mathbf{u}\in[H^1_D(\Omega_0)]^3}{\arg\min}\left\{\int_{\Omega_0}\mathcal{W}(\mathsf{C},\mathbf{X})\,\mathrm{d}x_0 - \langle\mathbf{f},\mathbf{u}\rangle\right\},\end{aligned} \tag{44}$$

where $[H^1_D(\Omega_0)]^3$ is the usual vector-valued Sobolev space of square integrable functions with square integrable derivatives that satisfies the given Dirichlet boundary conditions and $\mathrm{d}x_0$ is a measure on Ω_0.

Using standard arguments, see [41], a (possibly non-unique) minimum of the above problem exists. Taking the Fréchet derivative we can derive the residual equation:

$$F(\omega,\mathbf{u}^*;\tilde{\mathbf{u}}) = D_{\tilde{u}}L(u^*) = 0, \quad \forall\tilde{u}\in H^1_0(\Omega_0). \tag{45}$$

At this point we are in a position to the apply the general stochastic framework outlined in section 2.

We now describe the specific geometry, boundary conditions and probability density functions on the stochastic parameters that we use in this example. We consider the undeformed three-dimensional domain shown in fig. 3 given by $\Omega_0 = [0,L]\times[-l/2,l/2]\times[-e/2,e/2]$ with $e = l \ll L$. We define $\partial_D\Omega_0$ to be the portion of the boundary of the domain $\partial\Omega_0$ on which $X_x = 0$. We apply the Dirichlet condition $\mathbf{u} = \mathbf{0}$ on $\partial_D\Omega_0$. We summarise the preceding by saying that we have a beam built into a stiff wall on the left-hand end. No external boundary tractions are applied, however, the beam is subject to a body force due to gravity. We can then write the term with the linear functional $\mathbf{f}$ as:

$$\langle\mathbf{f},\mathbf{u}\rangle := \int_{\Omega_0}(-g)\rho u_z\,\mathrm{d}x_0, \tag{46}$$

where ρ is the possibly random density of the material and g is the standard acceleration due to gravity.

Out of the four material parameters $\{C_1, C_2, D_1, \rho\}$ in this example we select two to be subject to uncertainty, so:

$$\omega := \{\rho, D_1\} \in \mathbb{R}^{\star 2}_+. \tag{47}$$

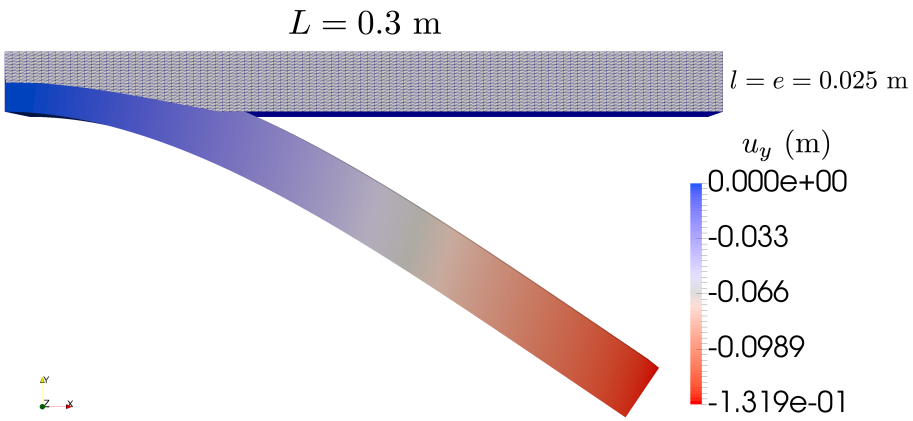


Figure 3: Undeformed domain Ω_0 shown with tetrahedral mesh. For a single realisation of the random variables $\omega := \{\rho, D_1\}$ we show the deformed domain Ω coloured with the y-component of the displacement u_y calculated using the deterministic forward model.

We specify the following distributions for the parameters:

$$\rho(\omega_1) := \rho^0(1 + \omega_1), \quad \omega_1 \sim \text{Beta}(2, 2), \tag{48a}$$

$$D_1(\omega_2) := D_1^0(1 + \omega_2), \quad \omega_2 \sim \text{Beta}(2, 2). \tag{48b}$$

With the Beta$(2, 2)$ distribution we have zero probability of drawing non-physical negative-valued parameters.

The numerical values for all of the parameters in the model are:

$$\begin{cases} C_1 = 10^4 \text{ Pa} \\ C_2 = 2 \times 10^5 \text{ Pa} \\ D_1^0 = 2 \times 10^5 \text{Pa} \\ L = 0.3 \text{ m} \\ l = 0.025 \text{ m} \\ g = 9.81 \text{ m/s}^2 \\ \rho^0 = 600 \text{ kg/m}^3 \end{cases} . \tag{49}$$

We now briefly describe the finite element method implementation of the forward model. We discretise the forward model using a piecewise linear finite element method on a uniform tetrahedral mesh leading to a discrete displacement solution with 120000 degrees of freedom. To solve the non-linear problem we use a Newton method from SNES [42] with continuation in the density parameter ρ

and a third-order backtracking line search. We let the symbolic differentiation capabilities of UFL derive the Jacobian of the forward model for the Newton solver. We solve the linear systems arising from the Newton iterations using a conjugate gradient method preconditioned using algebraic multigrid (Hypre BoomerAMG [43]) interfaced from PETSc [42]. We show an archetypal solution of the deterministic forward model for a single realisation of the stochastic parameters in fig. 3.

5.2.1. Tangent linear model

We let the symbolic differentiation capabilities of UFL derive the tangent linear models from the symbolic representation of the residual equation on our behalf. As we now have two stochastic parameters we derive two different tangent linear models so that we can calculate the required sensitivity derivatives, one with respect to each parameter $\{u_\rho, u_{D_1}\}$.

The tangent linear model is solved on the same mesh and using the same piecewise linear finite element method as the forward model. To solve the resulting linear system we use the MUMPS direct factorisation solver [44] interfaced from PETSc [42].

5.2.2. Parallelisation

We use the ipyparallel parallel computing toolbox [45] to distribute forward model evaluations across an 8 socket Intel Xeon E7-8880-based machine with 15 cores per socket, giving 120 cores in total. This machine is installed in the Gaia cluster at the University of Luxembourg [46]. We use the message passing interface (MPI) backend for communication between ipyparallel engines and the ZeroMQ distributed messaging system to communicate between engines and the controller. Each ipyparallel engine uses its own MPI communicator (`SELF`) to solve each realisation of the forward problem.

5.2.3. Estimation

Let's turn to the first experiment. We take four quantities of interest to be estimated using the standard Monte Carlo (MC), sensitivity derivative Monte Carlo (SD-MC) and multi-level polynomial chaos expansion (ML-PCE) methods:

$$\mathbf{X}_p := \{L, 0, 0\}^T, \tag{50a}$$
$$\psi_1 := \psi_m(u_x(\mathbf{X}_p)/L), \tag{50b}$$
$$\psi_2 := \psi_v(u_x(\mathbf{X}_p)/L), \tag{50c}$$
$$\psi_3 := \psi_m(u_y(\mathbf{X}_p)/l), \tag{50d}$$
$$\psi_4 := \psi_v(u_y(\mathbf{X}_p)/l). \tag{50e}$$

In words, the expectations of our quantities of interest are the mean and variance of the x and y-components of the displacement $\mathbf{u}$ evaluated at a point $\mathbf{X}_p$ located at the centroid of the $y-z$ plane at the free end of the beam.

For the PCE method we use multi-dimensional Jacobi polynomials of order $p = 10$ for the basis of the polynomial chaos expansion of stochastic dimension $M = 2$. We use three levels for the multi-level process. The MC and SD-MC methods are precisely those described in section 3.

We note that we cannot examine the convergence of the estimators in the same rigorous way as we did for the stochastic Burgers equation in section 5.1. This is because the forward model takes too long to run to generate an RMSE plot like the one shown in fig. 1, which required 370000 forward model evaluations. The results in this section can be viewed as being indicative that the SD-MC estimator performs well for this more complex problem. To do this we show how the estimate of quantity of interest evolves with increasing number of samples Z. In all experiments we let the guaranteed and reliable MC estimate run until the estimate settles down. We then use the evolution of this estimate as a point of comparison for the estimates from the SD-MC and ML-PCE methods.

Two evolutions of the three estimators of the first-order quantities of interest ψ_1 and ψ_3 with increasing number of realisations Z are plotted in fig. 4. All evolutions eventually converge to the same value for both quantities of interest. For ψ_3 the SD-MC estimator settles down very rapidly to the converged answer, requiring only tens of samples versus around 1000 samples for the standard Monte Carlo method. The ML-PCE method also performs well requiring around 500 samples on the final level.

The two evolutions of the three estimators of the second-order quantities of interest ψ_2 and ψ_4 with increasing number of realisations Z are plotted in fig. 5. All three estimators eventually converge to the same value for both quantities of interest. Although the results are less emphatic than those shown for the first-order quantities in fig. 4, the SD-MC estimator does seem to settle down faster than both the ML-PCE and MC estimators. There is not much to choose between the ML-PCE and MC estimators, although we remark that our methodology of using the final estimate of the MC estimator to denote the $Z - \mathbb{E}$ region does produce a bias in favour of the MC estimator.

Finally, we show the mean and standard deviation of the complete stochastic solution in fig. 6. This image was computed using the SD-MC method with $Z = 3000$ samples. We again remark that only two extra tangent linear problem solves (one for each parameter) were needed to obtain the required sensitivity derivatives for the SD-MC method.

5.2.4. Efficiency

Clearly computational time is an important factor in assessing the three estimation methods. Comparing the MC and SD-MC estimators is quite straightforward; Z solves of the non-linear forward model are required for both, and two extra forward and tangent linear model solve for the SD-MC method. The cost of these four extra solves is computationally negligible compared with the Z non-linear forward model solves. The cost of computing the MC and SD-MC estimators themselves is also negligible compared with the time spent solving

the non-linear forward problems. The only real variable of bearing on computational time for the MC and SD-MC methods then is Z.

However, comparing the computational time for both the MC and SD-MC methods with the ML-PCE method is more difficult. In the ML-PCE method we end up computing ZK non-linear model evaluations with varying numbers of non-linear iterations d_k in the Newton method at each level. In total, the ML-PCE method requires more linear solves than the MC and SD-MC methods, although the lower level solves are relatively cheaper, and the output of the lower levels is used to initialise the Newton iterations for faster convergence at the higher levels. In addition, a linear least-squares problem must be solved at each level, although again the cost of this is small compared with the non-linear forward model solves.

With the above in mind we show the wall time to achieve an accurate solution using the three methods for the first order quantity ψ_3 and second-order quantity ψ_4 in fig. 7. We can see that the SD-MC method performs hugely better at estimating the first-order quantity, and significantly better for the second-order quantity, compared to both the ML-PCE and MC methods.

5.2.5. Sensitivity and correlation analysis

In this final section we present some auxiliary local and global sensitivity analysis results calculated using the tangent linear model and the MC method, respectively. We also examine the correlation between two functionals of interest. The global sensitivity analysis was performed using the methods included in the software package Chaospy [32].

We first present the local sensitivity analysis of the quantities of interest ψ_1 and ψ_3. Additionally, we introduce the further quantity of interest $\psi_5 : [H_D^1(\Omega_0)]^3 \rightarrow [H_D^1(\Omega_0)]^3$:

$$\psi_5 := \psi_m(\mathbf{u}(\mathbf{X})/L), \tag{51}$$

which gives the mean displacement field of the beam as:

$$\bar{\mathbf{u}}(\mathbf{X})/L = \mathbb{E}[\psi_5]. \tag{52}$$

The quantities of interest ψ_1 and ψ_3 are then simply the x and y-components, respectively, of the mean displacement field evaluated at the point $\mathbf{X}_p$.

The studied local sensitivities of the quantities of interest with respect to the random parameters are then:

$$D_{\omega_i}[\psi_j](\bar{\omega}_i), \quad i = \{1, 2\}\,,\, j = \{1, 3, 5\}\,. \tag{53}$$

The norms of these quantities are given in fig. 8. All quantities of interest are more sensitive with respect to the density parameter ρ than the volumetric material parameter D_1. This matches our intuitive expectations of the problem setup, because the denser the object is the more it will sag under its own weight.

We also calculate the global sensitivities using the method of Sobol [47] using the polynomial chaos expansion output of MC method, see e.g. [48]. This

variance-based sensitivity method gives an understanding of the behaviour of the quantities of interest across the entire probability space, rather than the crude local sensitivity method just demonstrated.

The first-order and total effect Sobol indices are shown in fig. 9. The global trends are similar to those of local sensitivity analysis results in fig. 8. The density variable ρ explains most of the variance in the both output functionals ψ_1 and ψ_3. The total sensitivity index is nearly equal to the first order-sensitivity index for all variables and all quantities of interest. This means that there is little interaction between the two random variables.

Finally, in fig. 10(a) we show the normalised histogram of ψ_3 calculated using the MC and ML-PCE methods, and in fig. 10(b) the hexagonal binning correlation between ψ_1 and ψ_3. The distributions in fig. 10(a) calculated using the standard MC and ML-PCE methods are similar. From the hexagonal binning in fig. 10(b) it appears that ψ_1 and ψ_3 are partially correlated, with Pearson coefficient correlation of 0.34 with a zero p-value.

6. Conclusions

In this paper we have explored the use of sensitivity derivative Monte Carlo methods, originally introduced by Cao et al. [13], to efficiently propogate uncertainty through two high-level finite element models. The required sensitivity derivative information was efficiently calculated with the tangent linear model derived through automatic differentiation of the high-level model.

We compared the sensitivity derivative method to two other non-intrusive methods; the standard Monte Carlo method and a multi-level polynomial chaos expansion collocation method. For the two mechanics stochastic problems we examined we were able to show the improved performance of the sensitivity derivative driven estimator. The benefits for first-order type quantities are largest, but the gains still appreciable for second-order type quantities.

We are currently investigating the extension of this method to solving high-dimensional stochastic problems arising from random fields, where the advantages of Monte Carlo method's asymptotic behaviour independent of the stochastic dimension are most beneficial. In this high-dimensional context we will use the adjoint method to calculate the required derivative information efficiently with respect to the stochastic field.

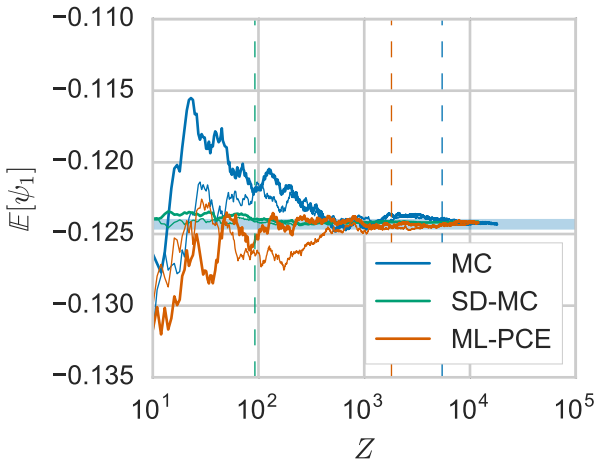
Supporting material

Complete code to produce the results in section 5.1 is given in [49].

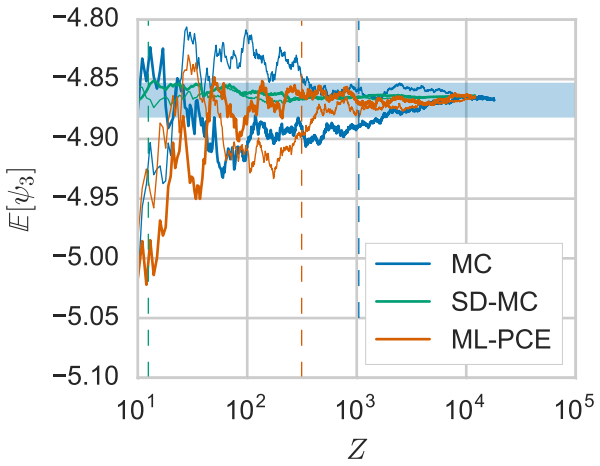
Acknowledgements

We thank the financial support of the 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.' We are also grateful for the support of the Fonds National de la Recherche Luxembourg FWO-FNR grant INTER/FWO/15/10318764. Jack S. Hale is supported by the National Research Fund, Luxembourg, and cofunded under the Marie Curie Actions of the European Commission (FP7-COFUND) Grant No. 6693582. The experiments presented in this paper were carried out using the HPC facilities of the University of Luxembourg [46].

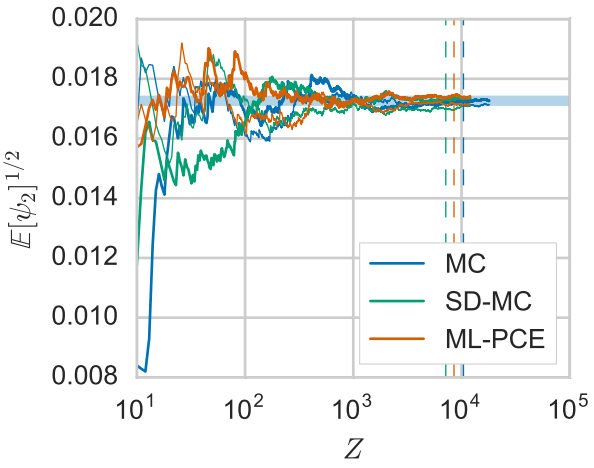


(a) Mean of $u_x(\mathbf{X}_p)/L$

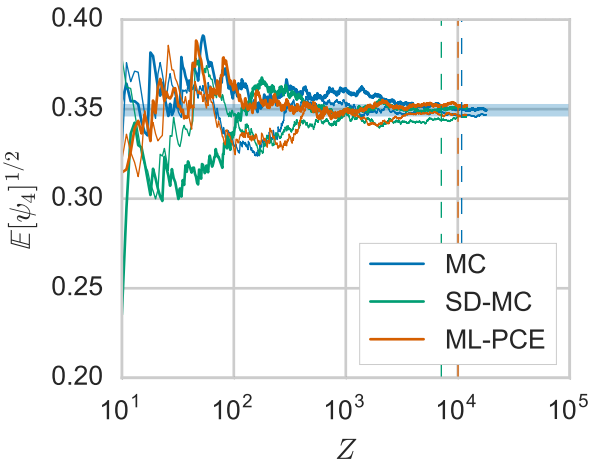


(b) Mean of $u_y(\mathbf{X}_p)/l$

Figure 4: Two evolutions of the MC, SD-MC and ML-PCE estimations of ψ_1 and ψ_3 as a function of the number of realisations Z. The blue shaded box indicates the region in $Z - \mathbb{E}$ space within $\pm 0.3\%$ of the final estimate predicted by the MC estimator. The vertical dashed lines, with colour corresponding to the estimator, shows the last mean Z for which the evolving curves leave this region. The SD-MC estimator settles down to the converged answer significantly faster than the MC or ML-PCE estimators.



(a) Standard deviation of $u_x(\mathbf{X}_p)/L$



(b) Standard deviation of $u_y(\mathbf{X}_p)/l$

Figure 5: Two evolutions of the square-root of the MC, SD-MC and ML-PCE estimations of ψ_2 and ψ_4 as a function of the number of realisations Z. The blue shaded box indicates the region in $Z - \mathbb{E}$ space within $\pm 1\%$ of the final estimate predicted by the MC estimator. The vertical dashed lines, with colour corresponding to the estimator evolution curves, shows the mean of the last Z for which the evolving curves leaves this region. The SD-MC estimator settles down to the converged answer faster than both the SD and ML-PCE estimators.

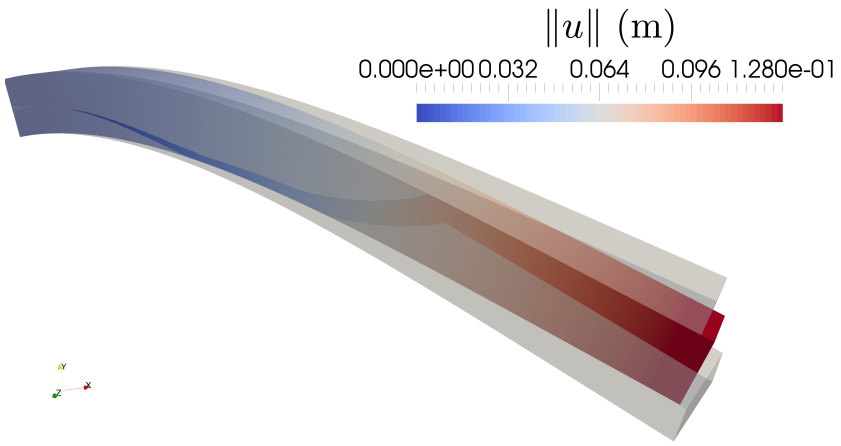
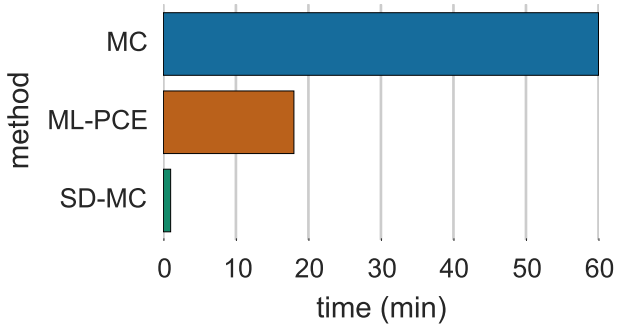
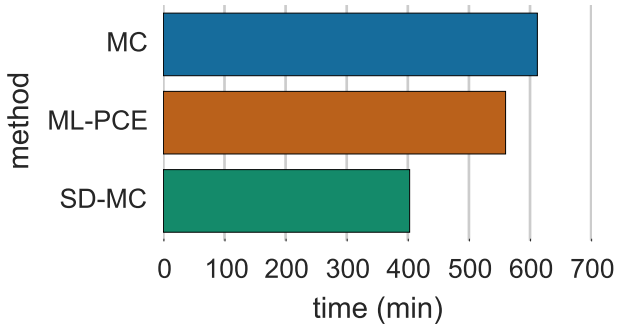


Figure 6: Solution of the beam governed by the stochastic Mooney-Rivlin hyperelastic equations with random density ρ and material parameter D_1. The mean deformed domain $\overline{\mathbf{x}} := \mathbf{X} + \bar{\mathbf{u}}$ is shown coloured with the pointwise magnitude of the mean displacement field $\|\bar{\mathbf{u}}(\mathbf{X})\|_{\mathbb{R}^3}$. The upper and lower transparent grey objects show the mean deformed boundary perturbed by the standard deviation of the displacement field $\bar{\mathbf{x}} \pm \boldsymbol{\sigma}(\mathbf{X})$, respectively.



(a) $\mathbb{E}[\psi_3]$



(b) $\mathbb{E}[\psi_4]$

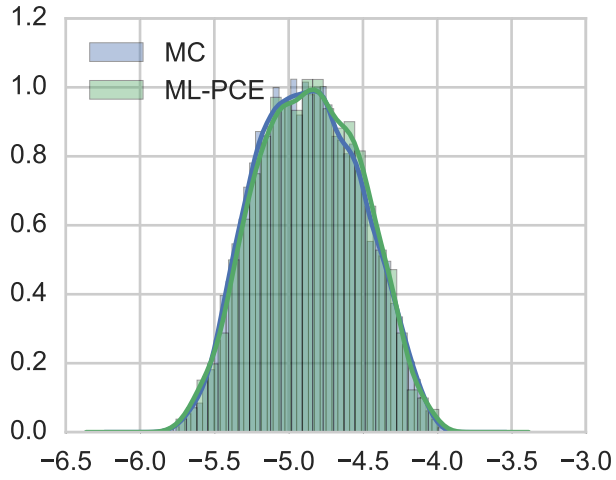
Figure 7: Bar plot of approximate elapsed wall time to generate an accurate estimate of the first-order quantity of interest ψ_3 (a) and second-order quantity of interest ψ_4 (b) using the three estimation methods. We define the Z to obtain an accurate solution for each method to be the corresponding dashed line given in fig. 4(b) and fig. 5(b), respectively. The timing is then calculated by multiplying the wall time by Z/Z_{max} where $Z_{\max}$ is the total number of fine-level forward model realisations at the end of the evolution.

	ρ	D_1
$\|D_{\omega_i}[\psi_1](\bar{\omega}_i)\|_{\mathbb{R}}$	0.07	0.016
$\|D_{\omega_i}[\psi_3](\bar{\omega}_i)\|_{\mathbb{R}}$	1.5	0.33
$\|D_{\omega_i}[\psi_5](\bar{\omega}_i)\|_{L^2(\Omega_0)}$	14.5	5.5

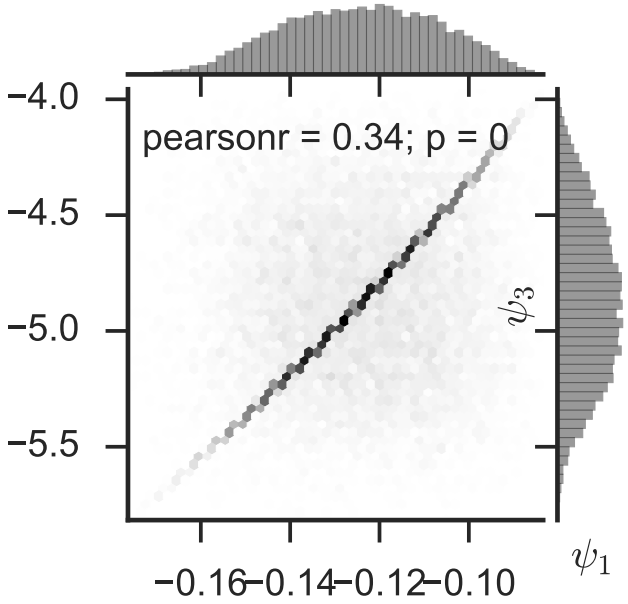
Figure 8: Local sensitivities of functionals of interest with respect to the parameters evaluated using the tangent linear model at the mean parameter. We can see in all cases that the problem is locally more sensitive to the density parameter ρ than the parameter related to the volumetric material behaviour D_1.

	$\psi_1 \sim u_x$		$\psi_3 \sim u_y$	
	ρ	D_1	ρ	D_1
First-order	0.862	0.136	0.867	0.133
Total	0.862	0.138	0.868	0.133

Figure 9: Global variance-based Sobol sensitivity indices calculated from the polynomial chaos expansion produced by the MC method. The first-order sensitivity indices are significantly larger for the density ρ than for the material parameter D_1 for both quantities of interest, signifying that the density explains most of the variance in the functionals of interest. The total sensitivity index is nearly equal to the first-order sensitivity index in all cases, signifying that there is little interaction between the two random variables.



(a) Normalised histogram of ψ_3 generated with MC and ML-PCE methods.



(b) Hexagonal binning histogram of $\psi_1 - \psi_3$ generated using MC method.

Figure 10: Various histogram plots.

References

[1] M. Sambridge, K. Mosegaard, Monte Carlo Methods in Geophysical Inverse Problems, Reviews of Geophysics 40 (3) (2002) 1009. doi:10.1029/2000RG000089.

[2] J. S. Liu, R. Chen, Sequential Monte Carlo Methods for Dynamic Systems, Journal of the American Statistical Association 93 (1998) 1032–1044. doi:10.2307/2669847.

[3] D. Landau, K. Binder, A Guide to Monte Carlo Simulations in Statistical Physics, Cambridge University Press, 2005.
URL http://www.cambridge.org/us/academic/subjects/physics/computational-science-and-modelling/guide-monte-carlo-simulations-statistical-physics-4th-edition

[4] P. Glasserman, P. Heidelberger, P. Shahabuddin, Variance Reduction Techniques for Estimating Value-at-Risk, Manage. Sci. 46 (10) (2000) 1349–1364. doi:10.1287/mnsc.46.10.1349.12274.

[5] J. Cárdenas, E. Fruchard, J. F. Picron, C. Reyes, W. Walters, K. ans Yang, Monte carlo within a day, Risk 12 (2) (1999) 55–59.

[6] R. J. Adler, J. E. Taylor, Random fields and geometry, no. 115 in Springer Monographs in Mathematics, Springer, New York, 2007.

[7] P. Malliavin, Stochastic analysis, Grundlehren der mathematischen Wissenschaften, Springer, Berlin, New York, 1997.

[8] M. B. Giles, Multilevel Monte Carlo Path Simulation, Operations Research 56 (3) (2008) 607–617. doi:10.1287/opre.1070.0496.

[9] R. E. Caflisch, Monte Carlo and quasi-Monte Carlo methods, Acta Numerica 7 (1998) 1–49. doi:10.1017/S0962492900002804.

[10] S. C. Brenner, L. R. Scott, The mathematical theory of finite element methods, Texts in applied mathematics, Springer, New York, Berlin, Paris, 2002. doi:10.1007/978-0-387-75934-0.

[11] A. Barth, C. Schwab, N. Zollinger, Multi-level Monte Carlo Finite Element method for elliptic PDEs with stochastic coefficients, Numerische Mathematik 119 (1) (2011) 123–161. doi:10.1007/s00211-011-0377-0.

[12] K. A. Cliffe, M. B. Giles, R. Scheichl, A. L. Teckentrup, Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients, Computing and Visualization in Science 14 (1) (2011) 3–15. doi:10.1007/s00791-011-0160-x.

[13] Y. Cao, M. Y. Hussaini, T. A. Zhang, Exploitation of sensitivity derivates for improving sampling methods, AIAA Journal 42 (2) (2004) 815–822. doi:10.2514/1.2820.

[14] Y. Liu, M. Yousuff Hussaini, G. Ökten, Optimization of a monte carlo variance reduction method based on sensitivity derivatives, Applied Numerical Mathematics 72 (Complete) (2013) 160–171. doi:10.1016/j.apnum.2013.06.005.

[15] E. Jimenez, Y. Liu, M. Y. Hussaini, Variance Reduction Method Based on Sensitivity Derivatives, Part 2, Appl. Numer. Math. 74 (2013) 151–159. doi:10.1016/j.apnum.2012.07.010.

[16] M. B. Giles, Multilevel monte carlo methods, Acta Numerica 24 (2015) 259–328. doi:10.1017/S096249291500001X.

[17] M. Girolami, B. Calderhead, Riemann manifold Langevin and Hamiltonian Monte Carlo methods, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73 (2) (2011) 123–214. doi:10.1111/j.1467-9868.2010.00765.x.

[18] S. Lan, T. Bui-Thanh, M. Christie, M. Girolami, Emulation of higher-order tensors in manifold Monte Carlo methods for Bayesian Inverse Problems, Journal of Computational Physics 308 (2016) 81–101. doi:10.1016/j.jcp.2015.12.032.

[19] O. Çavdar, A. Bayraktar, A. Çavdar, Effects of random material and geometrical properties on structural safety of steel–concrete composite systems, International Journal for Numerical Methods in Biomedical Engineering 27 (9) (2011) 1473–1492. doi:10.1002/cnm.1377.

[20] R. G. Ghanem, A. Doostan, On the construction and analysis of stochastic models: Characterization and propagation of the errors associated with limited data, J. Comput. Phys. 217 (1) (2006) 63–81. doi:10.1016/j.jcp.2006.01.037.

[21] R. G. Ghanem, P. D. Spanos, Stochastic finite elements : a spectral approach, Dover publications 2003, Mineola, New York, 1991. doi:10.1007/978-1-4612-3094-6.

[22] G. Stefanou, The stochastic finite element method: Past, present and future, Computer Methods in Applied Mechanics and Engineering 198 (9–12) (2009) 1031 – 1051. doi:10.1016/j.cma.2008.11.007.

[23] P. Frauenfelder, C. Schwab, R. A. Todor, Finite elements for elliptic problems with stochastic coefficients, Computer Methods in Applied Mechanics and Engineering 194 (2–5) (2005) 205–228. doi:10.1016/j.cma.2004.04.008.

[24] H. Matthies, A. Keese, Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations, Comput. Methods Appl. Mech. Eng. 194 (12-16) (2005) 1295–1331. doi:10.1016/j.cma.2004.05.027.

[25] G. H. Matthies, Stochastic finite elements: Computational approaches to stochastic partial differential equations, Journal of Applied Mathematics and Mechanics 88 (2008) 849–873. doi:10.1002/zamm.200800095.

[26] M. Eigel, C. J. Gittelson, C. Schwab, E. Zander, Adaptive stochastic Galerkin FEM, Computer Methods in Applied Mechanics and Engineering 270 (2014) 247–269. doi:10.1016/j.cma.2013.11.015.

[27] U. Naumann, The Art of Differentiating Computer Programs, Software, Environments and Tools, Society for Industrial and Applied Mathematics, 2011. doi:10.1137/1.9781611972078.

[28] P. Farrell, D. Ham, S. Funke, M. Rognes, Automated Derivation of the Adjoint of High-Level Transient Finite Element Programs, SIAM Journal on Scientific Computing 35 (4) (2013) C369–C393. doi:10.1137/120873558.

[29] M. S. Alnæs, A. Logg, K. B. Ølgaard, M. E. Rognes, G. N. Wells, Unified Form Language: A Domain-specific Language for Weak Formulations of Partial Differential Equations, ACM Trans. Math. Softw. 40 (2) (2014) 9:1–9:37. doi:10.1145/2566630.

[30] Y. Liu, M. Y. Hussaini, G. Ökten, Optimization of a monte carlo variance reduction method based on sensitivity derivatives, Applied Numerical Mathematics 72 (2013) 160–171.

[31] L. Giraldi, D. Liu, H. G. Matthies, A. Nouy, To be or not to be intrusive? the solution of parametric and stochastic equations—the "plain vanilla" galerkin case, SIAM Journal on Scientific Computing 36 (2014) 2720–2744. doi:10.1137/140969063.

[32] J. Feinberg, H. P. Langtangen, Chaospy: An open source tool for designing methods of uncertainty quantification, J. Comput. Science 11 (2015) 46–57. doi:10.1016/j.jocs.2015.08.008.

[33] C. A. Caflisch, Monte carlo and quasi -monte carlo methods, Acta numerica 7 (1998) 1–49. doi:10.1017/S0962492900002804.

[34] N. Wiener, The homogeneous chaos, American Journal of Mathematics 60 (1938) 897–936. doi:10.2307/2371268.

[35] M. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, G. N. Wells, The FEniCS Project Version 1.5, Archive of Numerical Software 3 (100). doi:10.11588/ans.2015.100.20553.

[36] A. Logg, G. N. Wells, DOLFIN: Automated Finite Element Computing, ACM Trans. Math. Softw. 37 (2) (2010) 20:1–20:28. doi:10.1145/1731022.1731030.

[37] B. Staber, J. Guilleminot, Stochastic modeling of a class of stored energy functions for incompressible hyperelastic materials with uncertainties, Comptes Rendus Mécanique 343 (9) (2015) 503 – 514. doi:http://dx.doi.org/10.1016/j.crme.2015.07.008.

[38] P. Wriggers, Nonlinear Finite Element Methods, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. doi:10.1007/978-3-540-71001-1.

[39] P. Haupt, Continuum Mechanics and Theory of Materials, Springer, 2012. doi:10.1007/978-3-662-04775-0.

[40] J. Bonet, R. D. Wood, Nonlinear continuum mechanics for finite element analysis, Cambridge University Press, Cambridge (UK), New York, 2008.

[41] J. E. Marsden, T. J. Hughes, Mathematical foundations of elasticity, Courier Corporation, 1994.
URL https://www.google.com/books?hl=en&lr=&id=STyKAAAAQBAJ&oi=fnd&pg=PP1&ots=eP_PF-xGjs&sig=H7Nfn2yAfag7ahH_QGdTd8g32zU

[42] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, H. Zhang, H. Zhang, PETSc users manual, Tech. Rep. ANL-95/11 - Revision 3.7, Argonne National Laboratory (2016).
URL http://www.mcs.anl.gov/petsc

[43] R. D. Falgout, U. M. Yang, hypre: A Library of High Performance Preconditioners, in: P. M. A. Sloot, A. G. Hoekstra, C. J. K. Tan, J. J. Dongarra (Eds.), Computational Science — ICCS 2002, no. 2331 in Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2002, pp. 632–641. doi:10.1007/3-540-47789-6_66.

[44] P. Amestoy, C. Ashcraft, O. Boiteau, A. Buttari, J. L'Excellent, C. Weisbecker, Improving Multifrontal Methods by Means of Block Low-Rank Representations, SIAM Journal on Scientific Computing 37 (3) (2015) A1451–A1474. doi:10.1137/120903476.

[45] F. Pérez, B. E. Granger, IPython: a system for interactive scientific computing, Computing in Science and Engineering 9 (3) (2007) 21–29. doi:10.1109/MCSE.2007.53.

[46] S. Varrette, P. Bouvry, H. Cartiaux, F. Georgatos, Management of an Academic HPC Cluster: The UL Experience, in: Proc. of the 2014 Intl. Conf. on High Performance Computing & Simulation (HPCS 2014), IEEE, Bologna, Italy, 2014, pp. 959–967.

[47] I. M. Sobol´, Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates, Math. Comput. Simul. 55 (1-3) (2001) 271–280. doi:10.1016/S0378-4754(00)00270-6.

[48] B. Sudret, Global sensitivity analysis using polynomial chaos expansions., Reliability Engineering and System Safety 93 (7) (2008) 964–979. doi:10.1016/j.ress.2007.04.002.

[49] P. Hauseux, J. S. Hale, S. P. A. Bordas, Solving the stochastic Burgers equation with a sensitivity derivative-driven Monte Carlo method, 2016. doi:10.6084/m9.figshare.3561306.v2.
URL https://figshare.com/articles/Solving_the_stochastic_Burgers_equation_with_a_sensitivity_derivative-driven_Monte-Carlo_method/3561306