Comparison of Numerical Methods in Uncertainty Quantification

Ionut-Gabriel Farcas
M.Sc. student in Computational Science and Engineering

Institut für Informatik
Technische Universität München
Boltzmannstraße 3
85748 Garching
ionut.farcas@tum.de

Abstract: This paper presents and compares the results obtained using several methods for Stochastic Computations, used in Uncertainty Quantification. We practically present the methods using a simple ODE model. The focus is on both intrusive and non-intrusive methods, namely the Monte Carlo method, along with methods based on generalized polynomial chaos(gPC) methodology. Moreover, we asses the obtained results by comparing them with the analytical solution of the system.

1 Introduction

The purpose of this paper is to present the results and a comparison between numerical methods used in *Uncertainty Quantification* (UQ). This is motivated by the fact that Uncertainty Quantification is a new and rapidly growing domain, at the edge of applied mathematics and computational science, that receives an increased amount of attention, due to its necessity in analyzing every physical event or engineering system.

Furthermore, the reason for choosing to work with a simple model resides in the fact that our desire is to fully understand the UQ methods in a practical way, that will provide a basis for the further extension of the current work. Our interest is to have a fully functional UQ example that can be used as a one-dimensional stochastic system and that can be easily extended to several dimensions.

As most of the practical problems are of a complex kind, it requires the usage of efficient and fast numerical methods, making uncertainty quantification also very interesting from the high performance computing point of view. An increased number of dimensions introduces an overhead, caused by the large number of points used in the space discretization, for instance, via generalized tensor product, as in [Xiu08] and in this case, the dimensionality of the space has an exponential growth. There are different methods to brake the curse of dimensionality, like sparse grids [BG04] and tensor trains [DKO14]. This idea is a major research topic in UQ and its implementation is very suited for high performance architectures. Moreover, all modern UQ methods can be easily parallelized, making them

suitable for implementation on high performance architectures.

Much related work was done in comparison of UQ methods. In [Xiu08], the author compared the underlying methods using the *Burgers equation* and the *Navier Stokes equations*, in [XLS01] the used equation was again the *Navier Stokes equations* and in [WIC06], the author compared the UQ methods using the *1 d heat equation*. For us, it is also interesting to compare the results obtained with the UQ methods with the ones obtained via the analytical solution.

For the future work, it is in our great interest to extend the current version of the application code and make it usable in the case of more complex systems and at the end, to have a working UQ API.

2 Mathematical description

The used system is a nonhomogenous second order ordinary differential equation (ODE), representing the mathematical description of a second order harmonic oscillator with zero initial conditions. The reason for choosing this model resides in the fact that it has a computable analytical solution, that will be used to compare to the results obtained with the stochastic methods. Moreover, from a stochastic point of view, it can be treated both as a one and multi-dimensional system, property that makes it very suitable for understanding uncertainty quantification in both one and multi dimensions.

2.1 Deterministic model

A second order oscillator (*Figure 1*) is modeled using Newton's second law of motion and we do not insist on it, as it is beyond the purpose of this paper and there is an extensive documentation regarding this topic available in the literature.

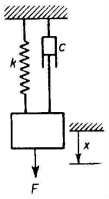

Figure 1: Second order oscillator

The model used in this paper is defined as $x : \mathbb{R}^+ \to \mathbb{R}$, by

$$\begin{cases} x'(t) = y(t) \\ y'(t) + c \cdot y(t) + k \cdot x(t) = f(t) \\ x(0) = y(0) = 0 \end{cases} \tag{1}$$

where

$$f(t) = F \cdot cos(\omega_a \cdot t) \tag{2}$$

and c represents the viscous damping coefficient $[N \cdot s/m]$, k the elasticity constant, F is the amplitude of the oscillations $[N/m^2]$ and ω_a is the angular frequency of the oscillations $[rad/s]$. Throughout this paper, the value of ω_a is set to 1.05 rad/s.

Figure 2 depicts the analytical solution for different values of k. As the focus is to analyze the results of uncertainty in one parameter of the system, we fix c = 5 $[N \cdot s/m]$ and F = 0.1 $[N/m^2]$. Thus, the only varying parameter remains k, the elasticity constant.

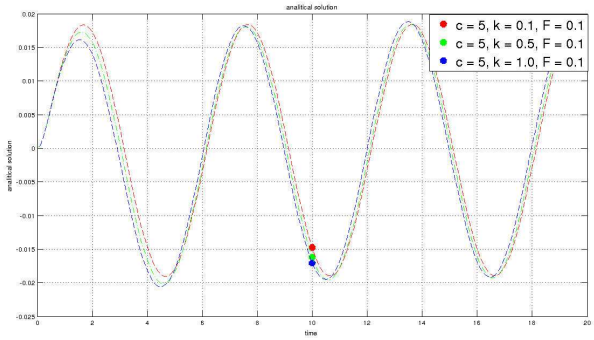

Figure 2: Analytical solution for different values of the parameters

2.2 Stochastic model

In the following, we consider a probabilistic framework and model the parameter k as a 1-variate random variable in a properly defined probability space $(\Omega, \mathfrak{A}, P)$, where Ω is the event space, $\mathfrak{A}$ is a σ-algebra and P is a probability measure, as in [Ras06].

In this paper, we model the uncertainty in (1) as a $Gaussian$ random variable $K : \Omega \to \mathbb{R}$ with the density $\rho : \mathbb{R} \to \mathbb{R}^+$, $(\overline{k},\ \sigma_k)$ the mean and the standard deviation and support $\Gamma = \mathbb{R}$. This choice is motivated by the fact that the majority of real life processes are of $Gaussian$ nature and this setup is the most favorable one to fully understand all the methods and concepts. Also, according to the authors in [Xiu08] and [Ras06], a $Gaussian$ process is fully charaterized by the first two statistical moments, i.e., the $expectation(mean)$ and the $variance$. The probability distribution function of K is:

$$\rho : \mathbb{R} \to \mathbb{R}^+, \rho(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}} \tag{3}$$

Thus, the stochastic model is defined as follows:

$$\begin{aligned} x : (\mathbb{R}^+ \times (\Omega, \mathfrak{A}, P)) \to (\mathbb{R}, B), x''(t,\omega) + 5 \cdot x'(t,\omega) + K(\omega) \cdot x(t,\omega) = \\ = 0.1 \cdot \cos(1.05 \cdot t), \text{with } x(0,\omega) = x'(0,\omega) = 0 \end{aligned} \tag{4}$$

2.3 Analytical solution

As it was specified in the beginning of this section, the choice of this system is motivated by the possibility to find its analytical solution and to use it for the assesment of the results obtained with the uncertainty quantification methods. With the setup presented in Section 2.1, the analytical solution will be a function that depends on time t and the elasticity constant k.

In order to compute the analytical solution, we employ the *Laplace transform* and obtain:

$$\begin{aligned} x(t,k) = c_1(k) \cdot \cos(\omega_a \cdot t) + c_2(k) \cdot \sin(\omega_a \cdot t) + \\ c_3(k) \cdot e^{\lambda_1(k) \cdot t} + c_4(k) \cdot e^{\lambda_2(k) \cdot t}, \end{aligned} \tag{5}$$

where $c_1(k)$, $c_2(k)$, $c_3(k)$, $c_4(k)$ are constants that depend on k and $\lambda_1(k)$ and $\lambda_2(k)$ are the zeros of $\lambda^2 + c \cdot \lambda + k$, depending on k as well. This second order polynomial is the associated algebraic polynomial of (1).

It is important to stress out that the form of analytical solution is determined by the nature of $\lambda_1(k)$ and $\lambda_2(k)$. With the setup presented in Section 2.1 together with the fact that $k \sim N(0,1)$ (*c.f. Section 2.2*), the zeros of $\lambda_1(k)$ and $\lambda_2(k)$ are $real$ and $distinct$, regardless of the value of k.

3 Uncertainty Quantification methods

In this section, the focus is on the presentation of the theoretical background of the stochastic methods. Moreover, for a better understaing of the methods from a computational point of view, we will present an overview of the numerical error for each method. And at the end, we will present a comparison of the methods based on convergence properties, implementation characteristics and scalability.

3.1 Sampling methods

Monte Carlo sampling, often abreviated as MCS, is one of the most commonly used methods in uncertainty quantification. According to [Xiu08] and [Iac11], the core idea of the algorithm is that one generates (independent) realizations of random inputs based on a prescribed probability distribution function and solves the problem for each generated sample. Thus, for each realization, the data is fixed and hence the problem becomes *deterministic*.

After one solves the deterministic realizations of the problem and obtaines the corresponding solutions, the statistical information can be easily extracted, e.g. $expectation(mean)$, $variance$, etc. It is important to remark that the formulas employed for obtaining the statistical information (*c.f. Algorithm 1, Step 3*) are independent of the choice of the probability density function used to generate the random inputs. The algorithm for using Monte

Carlo sampling in Uncertainty Quantification can be summarized as follows:

Algorithm 1 Monte Carlo sampling

1. Generate n samples of the random inputs, where n is user defined;
2. For each generated sample, solve the associated stochastic equation and obtain the solution $x^{(i)}(t,\omega)$;
3. Estimate the required solution statistics, e.g.
expectation(mean): $\mathbb{E}[x^{(i)}(t,\omega)] = \int_{\mathbb{R}} x^{(i)}(t,\omega)\rho(\omega)d\omega \approx \frac{1}{n}\sum_{i=1}^{n} x^{(i)}(t,\omega^{(i)})$
and variance: $Var[x^{(i)}(t,\omega)] = \mathbb{E}[(x(t,\omega) - \mathbb{E}[x(t,\omega)])^2] \approx \frac{1}{n-1}\sum_{i=1}^{n}(x^{(i)}(t,\omega^{(i)}) - \mathbb{E}[x^{(i)}(t,\omega)])^2$

3.2 Stochastic Collocation methods

As the authors in [Iac11], [JCC12] and [Xiu08] present, in collocation methods, we seek to satisfy the governing equation at a discrete set of points ("nodes") in the corresponding random space. From this, we see that, similar to MCS, we work again with a *deterministic* system. But unlike the Monte Carlo method, where the approach is based on sampling, the focus in stochastic collocations is, as presented in [Xiu08], to utilize an expansion that seeks to approximate the random process via orthogonal polynomials of random variables. The used expansion is based on *generalized Polynomial Chaos*(gPC) and for a more detailed view on this topic, see [JCC12], [Xiu08].

Let us consider an ∞-*dimensional* space $\mathbb{W}$ of orthogonal polynomials with respect to the probability measure ρ in $\mathbb{R}$, i.e.

$$\int_{\mathbb{R}} \rho(x)\cdot\phi^m(x)\cdot\phi^n(x)\cdot dx = h_{mn}^2\cdot\delta_{mn},\ \forall m,n\in\mathbb{N}_0 \tag{6}$$

where $\phi^m,\phi^n \in \mathbb{W}$, δ_{mn} is the Kronecker delta function and $h_{mn}^2 \in \mathbb{R}$ is the normalization factor. Furthermore, one can always normalize the orthogonal bases such that $h_{mn}^2 \equiv 1$ and this will be adopted throughout this paper. As the considered stochastic model is of $Gaussian$ nature with the support $\Gamma = \mathbb{R}$, the used polynomials are the $Hermite$ polynomials ($\phi^0 = 1,\ \phi^1 = 1,\ \phi^2 = x^2 - 1, \phi^3 = x^3 - 3x$ etc.), having the normalization factor $\int_{\mathbb{R}} \rho(x)\cdot(\phi^m)^2(x)\cdot dx = m!,\ \forall m\in\mathbb{N}_0$

As it is presented in [Xiu08], the Q^{th}-order gPC approximation of $x(t,\omega)$ can be obtained by projecting x onto the space $\mathbb{W}_Q$, i.e.

$$x(t,\omega) \approx x^Q(t,\omega) = \sum_{j=1}^{Q} c^j(t)\cdot\phi^j(\omega), \tag{7}$$

where $c^j(t)$ are the expansion coefficiants computed as:

$$c^j(t) = \int_{\mathbb{R}} x(t,\omega) \cdot \phi^j(\omega) \cdot \rho(\omega) \cdot d\omega = \mathbb{E}[x(t,\omega) \cdot \phi^j(\omega)], j = 1 \dots \mathrm{Q}, \tag{8}$$

where $\mathbb{E}$ is the expectation operator. From this, the statistics can be obtained easily. The expectation is calculated as (using (6) and (7)):

$$\begin{aligned}
\mathbb{E}[x(t,\omega)] &= \int_R x(t,\omega) \cdot \rho(\omega) \cdot d\omega \approx \int_R \sum_{j=1}^{Q} c^j(t) \cdot \phi^j(\omega) \cdot \rho(\omega) \cdot d\omega \\
&= \sum_{j=0}^{Q} c^j \cdot \int_R \phi^j(\omega) \cdot \rho(\omega) \cdot d\omega \\
&= c^0 \cdot \int_R \phi^0(\omega) \cdot \rho(\omega) \cdot d\omega + \sum_{j=1}^{Q} c^j \cdot \int_R \phi^j(\omega) \cdot \rho(\omega) \cdot d\omega = c^0
\end{aligned} \tag{9}$$

and the variance:

$$\begin{aligned}
Var[x(t,\omega)] &= \mathbb{E}[(x(t,\omega) - \mathbb{E}[x(t,\omega)])^2 = \int_R \Big(x(t,\omega) - \mathbb{E}[x(t,\omega)]\Big)^2 \cdot \rho(\omega) \cdot d\omega \\
&\approx \int_R \Big(\sum_{j=0}^{Q} c^j(t) \cdot \phi^j(\omega) - c^0\Big)^2 \cdot \rho(\omega) \cdot d\omega \\
&= \int_R \Big(\sum_{j=1}^{Q} c^j(t) \cdot \phi^j(\omega)\Big)^2 \cdot \rho(\omega) \cdot d\omega = \sum_{j=1}^{Q} (c^j(t))^2.
\end{aligned} \tag{10}$$

The literature presents several approaches for stochastic collocations, for instance see [JCC12], but we focus on the one based on a pseudo-spectral approach as it is presented in [Xiu08] and [XLS01]. In this approach, we compute the coefficiants $c^j(t)$ in (8) by using numerical quadrature, in the following way:

$$c^j(t) = \sum_{i=0}^{N} x(t,\omega^{(i)}) \cdot \phi(\omega^i) \cdot \alpha^i, \tag{11}$$

where $\{\omega^i, \alpha^i\}_{i=1}^N$ are a set of nodes and weights chosen such that the error between (8) and (11) is as small as possible (also called "aliasing error"). In (11), as it is presented in [Xiu08], $x(t,\omega^i)$ represents the deterministic solution with ω^i fixed.

In a 1-dimensional stochastic space where the random variable is of *Gaussian* nature, the optimal choice for nodes and weights is via *Gauss* quadrature and because the used orthogonal polynomials are the *Hermite* polynomials, we employ the *Gauss-Hermite* quadrature.

3.3 Stochastic Galerkin method

As we have seen in the previous section, the key in using a gPC approximation (7) is the evaluation of the coefficiants $c^j(t)$. Formula (8), used to compute of the coefficiants of the gPC expansion, is of little use, as it involves the unknown solution $x(t, \omega)$. An alternative method to stochastic collocations is the *Stochastic Galerkin* method. The core idea of the method is that we write (1) in a stochastic *weak form* and then we perform a projection, to obtain a (coupled) system of ODEs, as the author in [Xiu08] specifies.

To emphasize this idea, we consider the gPC approximation (7) of order Q for $x(t, \omega)$. Furthermore, let us consider a similar expansion for the random parameter k, based on the fact that k is normally distributed:

$$k(\omega) = \mu + \sigma \cdot \omega = \mu \cdot \phi^0(\omega) + \sigma \cdot \phi^1(\omega) \tag{12}$$

Thus, by plugging (7) and (12) into (1), we obtain:

$$\begin{gathered} \sum_{j=0}^{Q} (c^j)''(t) \cdot \phi^j(\omega) + c \cdot \sum_{j=0}^{Q} (c^j)'(t) \cdot \phi^j(\omega) + \\ (\mu \cdot \phi^0(\omega) + \sigma \cdot \phi^1(\omega)) \cdot \sum_{j=0}^{Q} c^j(t) \cdot \phi^j(\omega) = f(t) \end{gathered} \tag{13}$$

Next, a *Galerkin projection* is made using $\phi^k(\omega), \forall k = 0 \dots$ Q. After performing the projection, (13) becomes:

$$(c^k)''(t) + c \cdot (c^k)'(t) + \mu \cdot c^k(t) + \frac{1}{(\phi^k(\omega))^2} \cdot \sum_{j=0}^{Q} (\sigma \cdot c^j(t) \cdot e_{1jk}) = f(t) \cdot \int_R \phi^k(\omega) \cdot \rho(\omega) \cdot d\omega \tag{14}$$

where $e_{1jk} = \mathbb{E}[\phi^1(\omega) \cdot \phi^j(\omega) \cdot \phi^k(\omega)]$. Together with $\frac{1}{(\phi^k(\omega))^2}$, e_{1jk} can be evaluated analytically from the definition of the polynomials ϕ^i, as the authors in [JCC12] specify. We point out that the initial conditions in the equations (13) and (14) remain zero.

3.4 Comparison between Monte Carlo, Stochastic Collocations and Stochastic Galerkin methods

In order to efficiently implement and use the proposed methods, it is quintessential to thoroughly understand their particularities, advantages and disadvantages.

Regarding the $Monte\ Carlo$ method, its advantages are the fact that it is straightforward to use, thus easy to implement. Also, it is easily parallelizable and it is designed to work for arbitrary probability distribution functions. It features a major disadvantage, this is its accuracy depends on a large number of executions, making it computationally expensive. As the author in [Xiu08] specifies, the mean value typically converges as $\frac{1}{\sqrt{K}}$, where K is the number of realizations.

It is important to remark that the convergence of MCS is independent of the number of dimensions, thus, when the number of dimensions is high (typically ≥ 50), it is a very good choice. MCS requires the solution of the given equation, hence a numerical algorithm that automatically introduces a certain error. Moreover, as the method is based on sampling, this introduces an error as well. To summarize, the numerical error in MCS is:

$$\epsilon_{MC} = \epsilon_{sampling} + \epsilon_{num\_solution} \tag{15}$$

We remark that in the literature are proposed several methods to improve the convergence of MCS by e.g. using *quasi-Monte Carlo* method. For a more detailed view, see [Fox99].

The $Stochastic\ Collocation$ method is also straightforward to use and implement, as its main focus is to approximate the coefficiants $c^j(t)$ in (8) by (11). This requires a pair of nodes and weights used for numerical quadrature and a numerical solver for approximating the exact solution of (1). All solver calls are independent from each other and when working in multi-dimensional spaces (typically ≤ 50) there are proposed several approaches, as in [BG04] or [DKO14].

Unlike MCS, it converges very fast and it requires only a small number of coefficiants to efficiently determine the statistics. Regarding the existent numerical error, it is composed by the error of the approximation via gPC expansion (7), the numerical quadrature in (11) and the numerical solver for determining an approximation of the exact solution in (11). Thus,

$$\epsilon_{SC} = \epsilon_{gPC\_approx} + \epsilon_{num\_quadrature} + \epsilon_{num\_solution} \tag{16}$$

Stochastic Galerkin method is an intrusive method and its basic idea is that the projection is conducted in order to ensure that the error is orthogonal to the functional space spanned by the finite-dimensional basis. This is very efficient, as the error gets minimized, albeit it has an overhead when the form of (1) is complicated. Moreover, (14) is most of the time coupled, so this introduces an aditional overhead.

The main disadvantage of this method is the fact that the application code is complicated to implement and moreover, it is not reusable as in the case of the other presented methods (thus, the method is intrusive). Albeit, in the case of a "nice" form of (1), this method is very efficient, due to the fact that the convergence is exponential, as the author in [Xiu08] presents. We point out that (14) requires a numerical solver, that introduces a numerical error, along with the error caused by the gPC approximation (7) of $x(t, \omega)$.

$$\epsilon_{SC} = \epsilon_{gPC\_approx} + \epsilon_{num\_solution} \tag{17}$$

4 Numerical results

In this section we present the results obtained with the employed stochastic methods and to asses their quality by comparing them with the results obtained via the analytical solution. The quantities of interest are the *expectation(mean)* and the *variance* of the solutions, taken at time $t_{interest}$ = 10 s. The comparison will be done using the numerical values, the runtime, the relative errors and the used number of samples.

Table 1: Results obtained using a numerical solver

MCS					
Samples	Expectation	Variance	runtime(s)	E.R.E. [1]	V.R.E. [2]
100	0.00716	0.00603	2.5113	30.0%	71.7%
1000	0.00868	0.01075	6.1123	15.1%	49.5%
2500	0.00948	0.01505	17.6021	7.2%	29.2%
10000	0.00957	0.01801	24.2412	6.4%	15.3%
50000	0.00996	0.01988	120.6654	2.5%	6.5%
Collocations					
Points	Expectation	Variance	runtime(s)	E.R.E.	V.R.E.
20x2	0.01023	0.00802	0.4300	$\approx 8 \cdot 10^{-5}$	86.0%
20x4	0.01023	0.01404	0.5502	$\approx 8 \cdot 10^{-5}$	34.1%
20x5	0.01023	0.01812	0.7000	$\approx 8 \cdot 10^{-5}$	14.7%
20x6	0.01023	0.02019	0.8370	$\approx 8 \cdot 10^{-5}$	4.9%
20x8	0.01023	0.02115	1.1340	$\approx 8 \cdot 10^{-5}$	0.28%
Galerkin					
Points	Expectation	Variance	runtime(s)	E.R.E.	V.R.E.
20x2	-0.00343	0.00019	0.7421	133.5 %	86.0%
20x4	0.00968	0.00784	2.2512	5.3 %	63.1%
20x5	0.01017	0.01418	3.3131	0.5 %	33.3%
20x6	0.01022	0.01847	4.6418	0.03 %	13.2%
20x8	0.01023	0.02107	7.7876	$\approx 2.47 \cdot 10^{-5}$	0.96%

Moreover, having the analytical solution, we can see how much does the numerical error influence the results, by comparing the outcomes via analytical solution and the numerical solver. All simulations were performed using an Intel Core i7 2630M microprocessor and Matlab R2013b.

From Figure 2 (*c.f. Section 2.1*), we see that different values of k between 0 and 1 induce a small variance in the analytical solution, thus, we can infer that the variance in the numerical results will be small, too.

When using the analytical solution, the statistical moments are calculated using the formulas $\mathbb{E}[x(t,\omega)] = \int_R x(t,\omega) \cdot \rho(\omega) \cdot d\omega$ and $Var[x(t,\omega)] = \mathbb{E}[(x(t,\omega) - \mathbb{E}[x(t,\omega)])^2$, where the integrals are computed numerically, using *Gauss-Hermite* quadrature. We remark that the quadrature will introduce a small numerical error in the results that is $\sim 10^{-15}$. Using 15 integration nodes and weights, we obtain:

$$\mathbb{E}[x(t_{interest},\omega)] = 0.01023,\ Var[x(t_{interest},\omega)] = 0.02128 \tag{18}$$

In Table 1 [3], we present the results obtained with the stochastic methods when we use a numerical solver.

[1] Expectation relative error

[2] Variance relative error

[3] In the first column, for Collocations and Galerkin methods, the first number represents the number of integration points(N); the second one, the maximal degree of ϕ when computing the coefficiants $c^j(t)$(Q)

Table 2: Results obtained using the analytical solution

MCS					
Samples	Expectation	Variance	runtime(s)	E.R.E.	V.R.E.
100	0.00671	0.00745	0.1123	34.3%	69.9%
1000	0.01137	0.00118	0.1653	11.0%	44.3%
2500	0.00954	0.01639	0.2075	6.4%	22.9%
10000	0.00973	0.01835	0.4412	4.0%	13.7%
50000	0.01021	0.02107	2.6098	0.01%	0.29%
Collocations					
Points	Expectation	Variance	runtime(s)	E.R.E.	V.R.E.
20x2	0.01023	0.00297	0.03418	$\approx 1.77 \cdot 10^{-15}$	84.3%
20x4	0.01023	0.01401	0.01725	$\approx 1.77 \cdot 10^{-15}$	33.7%
20x5	0.01023	0.01815	0.02050	$\approx 1.77 \cdot 10^{-15}$	14.5%
20x6	0.01023	0.02023	0.02823	$\approx 1.77 \cdot 10^{-15}$	4.7%
20x8	0.01023	0.02122	0.05532	$\approx 1.77 \cdot 10^{-15}$	0.02%
Galerkin					
Points	Expectation	Variance	runtime(s)	E.R.E.	V.R.E.
20x2	0.00214	0.00568	0.3912	82.7 %	75.2%
20x4	0.00995	0.00899	0.5131	3.7 %	54.1%
20x5	0.01019	0.01628	0.6352	0.34 %	24.7%
20x6	0.01022	0.01927	0.8768	0.03 %	11.6%
20x8	0.01023	0.02127	1.2419	$\approx 4.58 \cdot 10^{-15}$	0.01%

From this, we see that all three methods converge, albeit in different manners. MCS requires a large number of samples($\sim 10^4$) to achieve convergence and this causes an overhead, due to the need for a large number of FLOPS (by an increased runtime) and thus, making the method computationally expensive. Moreover, from a computational science point of view, the inefficiency of this method is also caused by the large storage requirements, because while the number of samples increases, we need to store all computed solutions.

On the other hand, with much less used points, the Stochastic Collocations and Stochastic Galerkin methods converge very quickly and with much less computation power. Thus, we ascertain that the gPC based methods are more suitable for this application and although their overall numerical errors have three and respectively, two sources (*c.f. Section 3.4*), they produce very accurate results.

In the next simulation (*c.f. Table 2*), we use the analytical solution for the computations in MCS, Stochastic Collocations and Galerkin methods, thus eliminating the error caused by the numerical solver. In this way, we can better asses the obtained results.

As expected, the results for all methods are improved, proving that the error introduced by the numerical solver playes an important role in the final outcomes. In the case of MCS, for a large number of samples, the solution converges again, but this time, as we expect, the required number of FLOPS is drastically decreased (by approximately a factor of 60).

When using the analytical solution, the gPC based methods perform also much better, as the numerical error introduced by the numerical solver is eliminated. This leads to the conclusion that the gPC approximation (7) is very precise, even for a small number of points (i.e. a coarser truncation).

Considering a comparison between the Stochastic Collocations and Stochastic Galerkin methods, we can infer that although the Galerkin method converges slower, for a large number of points it performes better than the Collocations method. We should keep in mind that as the obtained results with these methods are similar, one is non-intrusive (Collocations) as the other is intrusive (Galerkin) - *c.f. Section 3*.

5 Conclusions

This paper underlined and compared results obtained with numerical methods used in uncertainty quantification, in a practical context. The focus was to use both intrusive and non-intrusive methods. As uncertainty is naturally present in every physical process or system, it is quintessential to consider it before any simulation is done.

The proposed example was a simple ODE model of a second order oscillator and the uncertainty was considered to be in one of the parameters. Being a simple model, it had a computable analytical solution that was used to asses the quality of the obtained results. Based on simulations, we assesed that both Stochastic Collocation method and the Stochastic Galerkin method are very suitable to treat the uncertainty in this system.

Moreover, as for the future work, we intend to transform the existing code into an UQ API and to also incorporate our sparse grids framework into the application code [4], this model can be used also as a complex system, as it is easily extendable to a higher chaos dimension.

References

[BG04] H.J. Bungartz and M. Griebel. Sparse Grids. *Acta Numerica*, pages 1–123, 2004.

[DKO14] S.V. Dolkov, B.N. Khoromskij, and I.V. Oseledets. Computation of Extreme Eigenvalues in Higher Dimensions using block tensor train format. *Computer Physics Communication*, 185:1207–1216, 2014.

[Fox99] B.P. Fox. *Strategies for Quasi-Monte Carlo*. Kluwer Academic Pub., 1999.

[Iac11] G. Iaccarino. Lectures on Uncertainty Quantification in Computational Science, Department of Mechanical Engineering Stanford University, 2011.

[JCC12] B. Jia, S. Cai, and Y. Cheng. Stochastic Collocation Method for Uncertainty Propagation. *Papers - American Institute of Aeronautics AND Astronautics*, 6:4770–4787, 2012.

[Ras06] I. Rasa. *Lectures on Probability Theory and Stochastic Processes*. U.T.PRES, 2006.

[4] see http://www5.in.tum.de/SGpp/releases/index.html

[WIC06] Q. Wang, G. Iaccarrino, and P. Constantine. Uncertainty quantification in simple linear and non-linear problems. *Center for Turbulence Reseach - Annual Research Briefs*, 2006.

[Xiu08] D. Xiu. Fast Numerical Methods for Stochastic Computations : A Review. *Communication in Computational Physics*, 5:242–272, 2008.

[XLS01] D. Xiu, D. Lucor, and C.H. Su. Stochastic Modeling of Flow-Structure Interactions using Generalized Polynomial Chaos. *Journal of Fluids Engineering*, 124:51–59, 2001.