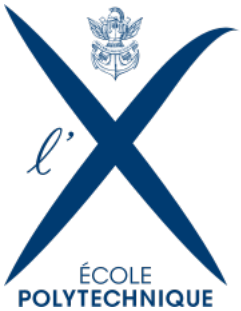
École Polytechnique



Report - Rapid covariance-based sampling of linear SPDE approximations in the multilevel Monte Carlo method

Andreas Petersson

J. Bellet, G. Klajer

9 décembre 2022

Table des matières

1 Introduction

The aim of this paper is to work on an efficient way to compute ineteresting quantities of a stochastic partial differential equation's approximation. Indeed, as analytical solutions are unvailable in general, numerical approximation of quantities such as covariance, expectation at a precise time t are sought-after.

The importance of this paper relies on the fact that sthocastic partial differential equations (SPDE) are really important in many fields such as physics (diffusion phenomenon) or in finances. Since then, having access to covariance, and other quantities of interest seems to be important in order to be able caracterize some concrete phenomenon.

The litterature all ready provides a lot of those approximations and the author A.Petterson decides to use the approach of one of them [1] . However, the author focus on, how, given this approximation, we could have access to interesting values. Indeed, the capital gain of this paper is the adaptation of the Monte Carlo algorithm to a co-variance based one able to give quicker access to basic parameters of $X(T)$. Moreover, this paper provides us a comparison between those two mehtods involving different assumptions and computation ressources.

This paper provides a covariance-based variants of the Monte-Carlo Mehtod algorithm. And it adapts it to Multi Level Monte Carlo as a second step. Those algorithms have the specificity to be particularly cheap in computational cost (developped in 3.2).

2 Problem Definition, methodology and theoritical analysis

2.1 Task Definition

Lets consider the linear SPDE :

$$dX(t) = AX(t) + F(t, X(t))dt + G(t)dW(t) \tag{1}$$

$$X(t = 0) = x_0$$

Where $t \in [0, T < \infty]$, $(H, \langle \cdot, \cdot \rangle, \| \cdot \|)$ a Hilbert ,$-A$ a positive definite, self-adjoint operator, W is an H-valued cylindrical Q-Wienner process, x_0 is a random member of a subspace of H. Finally F and G are mappigns that follows assumptions 2.0.1.

The spectral theorem applied to $(-A)^{-1}$ lets us introduce the orthogonal eigenbasis $(e_i)_{i\in\mathbb{N}}$ of H with $(\lambda_i)_{i\in\mathbb{N}}$ eigenvalues of $-A$

We are then able to define the separable Hilbert space :

$$\dot{H}^r = \left\{ f = \sum_{i=1}^{\infty} f_i e_i : f_i \in \mathbb{R}, \forall i \in \mathbb{N}, \|f\|_r^2 = \sum_{i=1}^{\infty} \lambda_i^r f_i^2 < \infty \right\}$$

With the inner product : $\langle \cdot, \cdot \rangle_r = \langle (-A)^{\frac{r}{2}} ., (-A)^{\frac{r}{2}} . \rangle$

For the spatial discretisation of (1), the authors follows the setting of [1] and introduces the $(V_h)_{h\in(0;1]}$, a family of subspaces of $\dot{H}^1$ equiped with $\langle ., . \rangle$ such that $N_h = dim(V_h) < \infty$. . In order to do so, two projectors are needed :

- $P_h : H^{-1} \to V_h$ the generalized orthognal projector onto V_h defined by : $\langle P_h f, \Phi \rangle_h = {}_{\dot{H}^{-1}}\langle f, \phi_h \rangle_{\dot{H}^1}$, where ${}_{\dot{H}^{-1}}\langle ., . \rangle_{\dot{H}^1}$ denotes the dual pairing.
- $R_h : R_h : \dot{H}^1 \to V_h$ the Ritz projector onto V_h : defined as the orthogonal projector onto V_h with respect to : $\langle \cdot, \cdot \rangle_1$

For the time discretization, the author uses the drift-implicit Euler Method. The time grid is obtained by the uniform discretization : $t_j = j\Delta t$ for $j = 0, 1, \ldots, N_{\Delta t} \in \mathbb{N}$

To avoid any confusions, the fully discrete approxiamtion (1) will be written : $(X_{h,\Delta t}^{t_j})_{j=0}^{N_{\Delta t}}$

Finally, the paper provides an expression of $(X_{h,\Delta t}^{t_j})_{j=0}^{N_{\Delta t}}$:

$$X_{h,\Delta t}^{t_{j+1}} - X_{h,\Delta t}^{t_j} = (A_h X_{h,\Delta t}^{t_{j+1}} + P_h F(t_j, X_{h,\Delta t}^{t_j}))\Delta t + P_h G(t_j)\Delta W^j \tag{2}$$

The aim of this paper is to compute an approximation of $\mathbb{E}\phi(X(T)]$ where $\phi(X(T))$ is a basic function of the SPDE's solution. In practice $E[\phi(X)]$ may represent the mean, the expectation, the covariance ...

2.2 Presentation of the Method

In order to compute an apporximation of $\mathbb{E}[\phi(X(T))]$, the paper propose to use the Monte-Carlo estimator, reminded by :

$$E_N[Y] = \frac{1}{N}\sum_{i=1}^{N} Y^{(i)}$$

Nevertheless, to be able to approximate $\mathbb{E}\phi(X(T))$ by the Monte-Carlo estimator, we use the discrete approximation $X^T_{h,\Delta t}$ of $X(T)$. In order to do so, we need to generate a large number of samples of $X^T_{h,\Delta t}$.

In practice the author explains that we sample the vector $\overline{\mathbf{x}}^T_h = [x_1, x_2, \dots, x_{N_h}]'$, Where the $(x_i)_{i=1,\dots,N_h}$ are the coefficients of the expansion : $X^T_{h,\Delta t} = \sum_{k=1}^{N_h} x_k \Phi^h_k$, where $\mathbf{\Phi}^h = (\Phi^h_k)_{k=1}^{N_h}$ is a basis of V_h.

Starting from this setting the paper compares two approaches : the path-base sampling and the covariance-based sampling.

2.2.1 Path-based sampling :

The path-based sampling is the classical approach to solve our problem. To do so, the method is to solve the $N_{\Delta t}$ matrix equations corresponding to (2) once for each sample $i = 1, 2, \dots, N_h$. $N_{\Delta t}$ is obtained by expanding (2) on $\mathbf{\Phi}^h$ and applying $\langle \Phi^h_i, . \rangle$ to each side of this equality for $i = 1, 2, \dots, N_h$. We then have access to $\overline{\mathbf{x}}^T_h$ that gives us acces to $\phi(X^T_{h,\Delta t})^{(i)}$.

It can be resumed by the pseudo-code Algorithm :

Algorithm 1 Path-based MC method of computingan estimate $E_N[\phi(X^T_{h,\Delta t})]$ of $\mathbb{E}[\phi(X(T))]$

Ensure: $E_N[\phi(X^T_{h,\Delta t})]$
  $result = 0$
  for $i = 1$ to N **do**
    Sample increments of a realization W^i of the Q-Wiener Process W
    Compute the vector $\overline{\mathbf{x}}^T_h = [x_1, x_2, \dots, x_{N_h}]'$ directly by solving the matrix equation corresponding to the drift Euler-Maruyama system (2) driven by $W^{(i)}$
    Compute $\phi(X^T_{h,\Delta t}) = \phi(\sum_{k=1}^{N_h} x_k \Phi^h_k)$
    $result = result + \frac{\phi(X^T_{h,\Delta t})^{(i)}}{N}$
  end for
  $E_N[\phi(X^T_{h,\Delta t})] = result$

2.2.2 Covariance-based sampling :

This method, is the alternative proposed by the author in order to improve computation costs of the approximation of $\mathbb{E}[\phi(X(T))]$.

To apply it, assumptions 2.0.1 need to be verified by (1).

Assumptions 2.0.1 *Assumptions, under wich, $\overline{\boldsymbol{x}}^T_h$ is gaussian and guarantees the existence of a mild solution to (1)*

- *$W = (W(t))_{t\in[0,T]}$ is an $(\mathcal{F})_{t\in[0,T]}$-adapted cylindrical Q-Wiener process where the operator $Q \in \mathcal{L}(H)$ is self-adjoint and positive demidefinite.*
- *There is a constant $C > 0$ such that $G : [0,T] \to \mathcal{L}^0_2$ satisfies :*

$$\|G(t_1) - G(t_2)\|_{\mathcal{L}^0_2} \leqslant C \mid t_1 - t_2 \mid^{1/2}$$

- *The function $F : [0,T] \times H \to \dot{H}^{-1}$ is affine in H and,*

$$\exists C > 0, \forall f \in H, \|F(T_1, f) - F(t_2, f)\|_{-1} \leqslant C(1 + \|f\|) \mid t_1 - t_2 \mid^{1/2}$$

- *The initial value x_0 is a $\mathcal{F}_0$-measurable $\dot{H}^1$-valued Gaussian random variable.*

Some notations need to be introduced :

- $R_{h,\Delta t} = (I_H - \Delta t A_h)$
- $F^{1,j}_{h,\Delta t} = (I_H - \Delta t P_h) F^1_{t_j}$

— $F^{2,j}_{h,\Delta t} = \Delta t P_h F^2_{t_j}$

The capital gain of the paper lies in the theorem 2.1, that gives access to $\mu^T = \mathbb{E}[X^T_{h,\Delta t}] \in V_h$ and $\Sigma^T = Cov(X^T_{h,\Delta t}) \in V_h^{\otimes 2}$

Theorem 2.1 (Recursions giving access to expectation and covariance of $X^T_{h,\Delta t}$) *Under assumptions 2.0.1 and if $(X^T_{h,\Delta t})_{j=0}^{N_{\Delta t}}$ is obtained by (2) We have access to the recursions :*

$$R_{h,\delta t}\mu^{t_{j+1}} = F^{1,j}_{h,\Delta t}\mu^{t_j} + F^{2,j}_{h,\Delta t} \tag{3}$$

$$(R_{h,\delta t})^{\otimes 2}\Sigma^{t_{j+1}} = (F^{1,j}_{h,\Delta t})^{\otimes 2}\Sigma^{t_j} + \mathbb{E}[(P_h G(t_j)\Delta W^j)^{\otimes 2}] \tag{4}$$

The idea of this method is to compute $Cov(X_{h,\Delta t})$. The covariance matrix of $\overline{\mathbf{x}}^T_h$ is then used to generate samples of $\overline{\mathbf{x}}^T_h$.

Algorithm 2 Covariance-based MC method of computing an estimate $E_N[\phi(X^T_{h,\Delta t})]$ of $\mathbb{E}[\phi(X(T))]$

Ensure: $E_N[\phi(X^T_{h,\Delta t})]$

From the mean vector μ and covariance matrix Σ of $\overline{\mathbf{x}}^T_h$ by solving the matrix equation corresponding to (3) and (4)

$result = 0$

for $i = 1$ to N **do**

Sample $\overline{\mathbf{x}}^T_h = [x_1, x_2, \ldots, x_{N_h}]' \sim N(\mu, \Sigma)$

Compute $\phi(X^T_{h,\Delta t}) = \phi(\sum_{k=1}^{N_h} x_k \mathbf{\Phi}^h_k)$

$result = result + \frac{\phi(X^T_{h,\Delta t})^{(i)}}{N}$

end for

$E_N[\phi(X^T_{h,\Delta t})] = result$

2.2.3 Covariance-based sampling in a multilevel Monte Carlo setting

For our aim to compute $\mathbb{E}[\phi(X(T))]$, the MLMC algorithm, can be a more efficient. In the paper, the author, propose to adapt the mehtod developped in 2.2.2 with multi level MonteCarlo estimator :

$$E^L[Y_L] = E_{N_0}[Y_0] + \sum_{l=1}^{L} E_{N_l}[Y_l - Y_{l-1}]$$

Where $(Y_l)_{l\in\mathbb{N}_0}$ a sequence of random variables in $L^2(\Omega, \mathbb{R})$. In this setting, l can be interpreted as the level of the MLMC estimator.

Under some stronger assumptions, the paper also exposes how to adapt the two previous methods 2.2.1 and 2.2.2 to the MLMC estimator in order to have better computational cost.

2.3 Theoretical justification and guarantees

The paper provide two kind of convergence guarentees.

The first one is given by the theorem 2.2

Theorem 2.2 (Strong convergence) *Under the assumpitons 2.0.1 and, $X^T_{h,\Delta t}$ an approximation of $X(T)$. Then, $\forall p \geq 1, \sup_{h,\Delta t}(\|X^T_{h,\Delta t}\|_{L^p(\Omega;H)}) < \infty$, there is a constant $C > 0$ such that :*

$$\|X(T) - X^T_{h,\Delta t}\|_{L^p(\Omega;H)} \leq C(h + \Delta t^{1/2})$$

Under assumptions 2.0.1 and 2.2.1, we obtain the weak convergence of theorem 2.3

Assumptions 2.2.1 — *For some $\delta \in [1/2, 1]$ there is a constant $C > 0$ such that G and F satisify :*

$$\|G(t_1) - G(t_2)\|_{\mathcal{L}^0_2} \leq C \mid t_1 - t_2 \mid^{\delta}$$

and

$$\|F(t_1) - F(t_2)\|_{\mathcal{L}^0_2} \leq C \mid t_1 - t_2 \mid^{\delta}$$

— *The functional ϕ is a member of $C_p^2(H, \mathbb{R})$ the space of all continous mappigns from H to $\mathbb{R}$ wich are twice continously Fréchet-differentiable with at most polynomially growing derivatives.*
— *The initial value $x_0 \in \dot{H}^1$*

Theorem 2.3 (Weak convergence) *Under assumptions 2.0.1 and 2.2.1, there is a constant $C > 0$ such that :*

$$| \mathbb{E}[\phi(X(T)) - \phi(X^T_{h,\Delta t})] | \leq C(1+ | log(h) |)(h^2 + \Delta t^\delta), \forall h, \Delta t \in [0, 1]$$

3 Experimental evaluation

3.1 Methodology

To evaluate the method we use a Monte Carlo estimate of the quadratic error including a reference solution for a very fine mesh grid $h = 2^{-8}$. The numerical experiment makes comparison between the convergence rates of the different estimators returned by the algorithms and verifies if it corresponds to the expected theoritical results. To do so, we have built several algorithms, run them for a fix number of iterations, computed the quadratic error relative to a reference algorithm - that is valid in this specific setting -, and finally ploted the results. The data is entirely generated in the simulation of the SPDE. Moreover, we have collected and focused mostly on the performance data - except for the plot of the path of the stochastic process simulated.

3.2 Results

Assumptions 3.0.1 *There is a $d \in \mathbb{N}$ such that the cost of computing :*
— *one step of 2 is $\mathcal{O}(h^{\alpha d})$ where $\alpha \in [1, 2]$*
— *one step oh the tensorized system 4 is $\mathcal{O}(h^{-2d})$*
— *sampling a gaussian V_h-valued random variable with covariance given by 4 $\prime(h^{-2d})$*
— *any other operation of algorithm $i \in 1, 2$ corsts $\mathcal{O}(h^{-\omega_i d})$ where $\omega_i \in \mathbb{N}$*
$\alpha \in [1, 2]$

Under the assumptions 3.0.1, the cost of computing $E_N[H^T_{h,\Delta t}]$ with Algorithm 1 is bounded by $\prime(max(h^{-4-\alpha d-2/\delta}, h^{-\omega_1 d}))$ and with Algorithm 2 by $\mathcal{O}(max(h^{-2d-4}, h^{-\omega_2 d}))$

More over we obtained some convergences results summarized by the different plots :

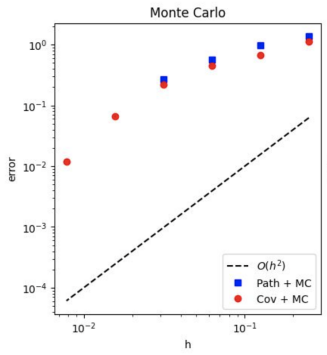


FIGURE 1 – Convergence, Monte Carlo from our implementation

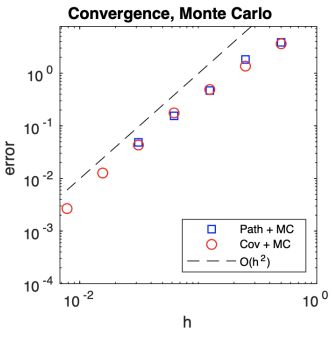


FIGURE 2 – Convergence, Monte Carlo from the paper

Some additionnal plots of our implementation are available in the Appendix 3.3.

3.3 Discussion

The numerical results match indeed the theoretical ones. The method is quite powerful because it allows to compute efficiently the expected values by developing specific algorithms - in particular covariance-based scheme. Nevertheless, the method is still computationally expensive. The theoretical results encompass precisely the algorithmic implementation. This is why the results are coherent : the only source of variations/errors from one implementation to another may only be due to the choice of the elementary solvers used in the method. Furthermore, the setting of the paper is broad and can be applied to a lot of SPDEs.

Références

[1] Raphael Kruse. "Strong and Weak Approximation of Semilinear Stochastic Evolution Equations". In : *Springer Cham* 177 (2014). DOI : https://doi.org/10.1007/978-3-319-02231-4.

Appendix

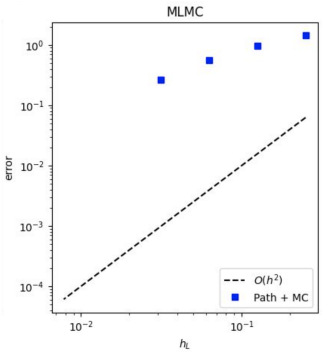


FIGURE 3 – Convergence, Monte Carlo Multi Level from our implementation

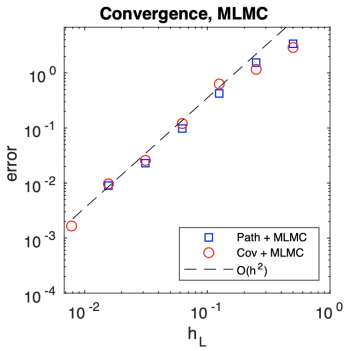


FIGURE 4 – Convergence, Monte Carlo Multi Level from the paper

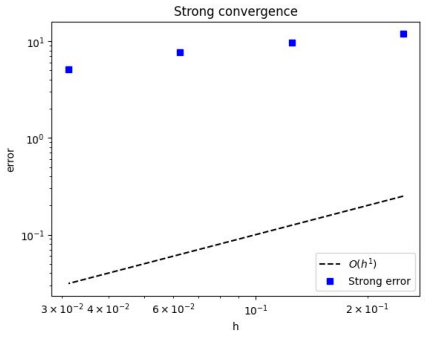


FIGURE 5 – Monte Carlo estimate of the strong error $\|X(T) - X^5_{h,\Delta t}\|_{L^2(\Omega;H)}$ from our implementation

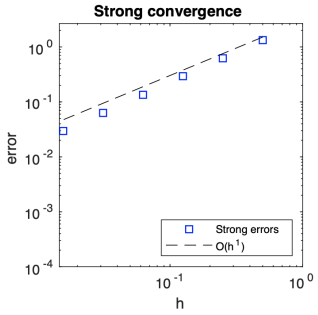


FIGURE 6 – Monte Carlo estimate of the strong error $\|X(T) - X^5_{h,\Delta t}\|_{L^2(\Omega;H)}$ from the paper

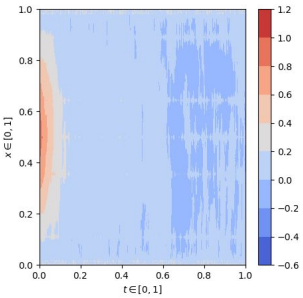


FIGURE 7 – Approximated path - $X^t_{h=2^{-8}}$

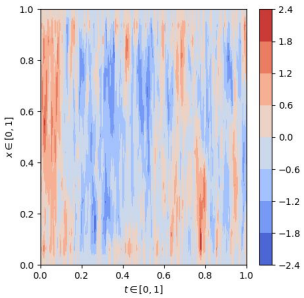


FIGURE 8 – Approximated path - $X^t_{h=2^{-4}}$

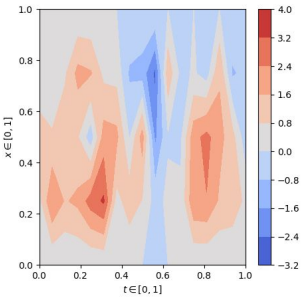


FIGURE 9 – Approximated path - $X^t_{h=2^{-2}}$

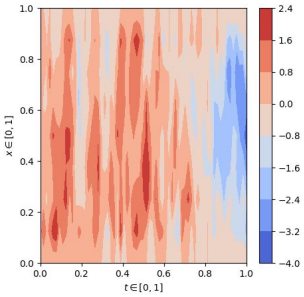


FIGURE 10 – Approximated path - $X^t_{h=2^{-3}}$