École Polytechnique Fédérale de Lausanne

MATH-414 - Stochastic Simulation

Monte Carlo estimation of Sensitivities in Finance

Students :

- Charles Gendreau
- Eliott Van Dieren

Professor : Prof. Fabio Nobile
Teaching Assistant : Sundar Ganesh

Academic Year 2022-2023

Contents

1 Introduction

1.1 General definition of studied sensitivities in finance

Studying sensitivities in finance, also known as “Greeks”, can be useful for financial actors to assess the risk of a given derivative contract. Sensitivities are defined as partial derivative of quantities of interest such as option prices to understand how those quantities are impacted by a change of different parameters such as the price of the underlying S_0, its volatility σ or the risk-free interest rate r. For this project, the focus will be set on different ways of using the Monte-Carlo method to estimate three Greeks: Delta (δ), Vega (ν) and Gamma (γ), for various kind of derivative contracts. Given an option on an asset A, the Delta metric measures how much a \$1 change in the price of the underlying A will affect the price of the option. For an European call option, $\delta \in [0, 1]$ and if the price of A increases by \$1, then the call option on A will increase by δ. The Gamma is the derivative of Delta and so it measures how much a \$1 change in the price of A will change the value of Delta. Finally, the Vega measures how much the price of an option on A will change after a given change in the volatility of A.

In Section 1.2, we define the models and mathematical definitions of sensitivities. In Section 2, three methods to differentiate an option's price are defined while in Section 3, those methods are applied to various derivative contracts to estimate their sensitivities to model's parameters. Section 4 summarizes the report into take-home messages and discusses the next steps related to sensitivities estimation. Appendix A contains the functions used to estimate the various sensitivities.

1.2 Mathematical notations and models

The model used for the underlying prices for this project is the usual Black-Scholes model, which gives us the closed form for the stock price at time T:

$$S_T = S_0 \exp\left((r - \frac{1}{2}\sigma^2)T + \sigma W_T\right) \tag{1}$$

where S_0 is today's underlying price, r is the risk-free interest rate, σ is the volatility of the stock and $\{W_t\}_{t\in[0,T]}$ is a standard Brownian motion such that $W_T \sim \mathcal{N}(0, T)$. We assume both r and σ to remain constant in the period $[0, T]$. The price of an option expiring at time T can be expressed as $\mathbb{E}[f(S_T)]$ for some payoff function f.

Hence, given a stock with price S_t and a corresponding option payoff $f(S_T)$, the mathematical definition of Delta, Vega and Gamma are respectively

$$\delta := \frac{\partial \mathbb{E}[f(S_T)]}{\partial S_0}, \quad \nu := \frac{\partial \mathbb{E}[f(S_T)]}{\partial \sigma}, \quad \gamma := \frac{\partial^2 \mathbb{E}[f(S_T)]}{\partial S_0^2}$$

For clarity purposes, let $I := \mathbb{E}[f(S_T)]$. To estimate I and more generally any expected value in this project, we use the Crude Monte-Carlo (CMC) estimator which will be discussed in the next section.

2 Estimating sensitivities

We are interested into estimating how a change in the initial stock price S_0 or its volatility σ affects the price of the option $I := \mathbb{E}[f(S_T)]$. Our goal is thus to find $\frac{\partial I}{\partial \theta}$ for $\theta = S_0, \sigma$ and $\frac{\partial^2 I}{\partial S_0^2}$.

2.1 Finite differences

If I is sufficiently differentiable with respect to the parameter θ, we can write

$$\frac{\partial I}{\partial \theta} = \lim_{\Delta\theta \to 0} \frac{I(\theta + \Delta\theta) - I(\theta - \Delta\theta)}{2\Delta\theta}, \tag{2}$$

$$\text{and} \quad \frac{\partial^2 I}{\partial \theta^2} = \lim_{\Delta\theta \to 0} \frac{I(\theta + \Delta\theta) - 2I(\theta) + I(\theta - \Delta\theta)}{(\Delta\theta)^2}. \tag{3}$$

Hence, we can give an approximation of the sensitivities evaluating the right hand terms at a sufficiently small $\Delta\theta$. Once this parameter is fixed, we observe that the only source of randomness in $f(S_T)$ is $W_T \sim \mathcal{N}(0,T)$ due to the underlying model (1). Hence, for different parameter θ', one can simulate N independent underlying prices at time T, $(S^{(i)}_{T,\theta'})_{i\in[0,\dots,N]}$ and then estimate $I(\theta') = \mathbb{E}[f(S_{T,\theta'})]$ with the Crude Monte-Carlo estimator given by equation (4) as well as the empirical standard deviation of $f(S_{T,\theta'})$.

$$I \approx \hat{\mu}_{CMC} = \frac{1}{N}\sum_{i=1}^{N} f(S^{(i)}_{T,\theta'}) \quad \text{and} \quad \hat{\sigma}^2_{CMC} = \frac{1}{N-1}\sum_{i=1}^{N}(f(S^{(i)}_{T,\theta'}) - \hat{\mu}_{CMC})^2 \,. \tag{4}$$

2.2 Pathwise derivatives

In some cases related to continuity of $f(S_T)$, one can interchange the derivative and the expectation, such that:

$$\frac{\partial I}{\partial\theta} = \frac{\partial}{\partial\theta}\mathbb{E}[f(S_T)] = \mathbb{E}\left[\frac{\partial}{\partial\theta}f(S_T)\right] = \mathbb{E}\left[f'(S_T)\frac{\partial S_T}{\partial\theta}\right]. \tag{5}$$

For $\theta = S_0$ we have from (1)

$$\frac{\partial S_T}{\partial S_0} = e^{(r-\frac{1}{2}\sigma^2)T+\sigma W_T} = \frac{S_T}{S_0}, \tag{6}$$

and for $\theta = \sigma$

$$\frac{\partial S_T}{\partial\sigma} = (W_T - \sigma T)S_0 e^{(r-\frac{1}{2}\sigma^2)T+\sigma W_T} = (W_T - \sigma T)S_T. \tag{7}$$

Once the parameters are fixed, $f'(S_T)\frac{\partial S_T}{\partial\theta}$ can be easily simulated as it only depends on W_T by using the Crude Monte-Carlo estimator.
Similarly, if we can interchange the derivative and the expectation twice, we get :

$$\frac{\partial^2 I}{\partial\theta^2} = \frac{\partial^2}{\partial\theta^2}\mathbb{E}[f(S_T)] = \frac{\partial}{\partial\theta}\mathbb{E}\left[f'(S_T)\frac{\partial S_T}{\partial\theta}\right] = \mathbb{E}\left[f''(S_T)\left(\frac{\partial S_T}{\partial\theta}\right)^2 + f'(S_T)\left(\frac{\partial^2 S_T}{\partial\theta^2}\right)\right]$$

which can be estimated using the same aforementioned procedure.

We now discuss when one is allowed to interchange the derivative w.r.t. θ and the expectation. A general rule of thumb discussed in [1, p. 396] could be used to say that we are allowed to interchange the derivative and the expectation (usually) only if $\theta \mapsto f(S_T)$ is continuous and piecewise differentiable. We try to give here an argument that could give an intuition for this rule. One could approximate f by a increasing sequence of smooth functions $\{f_n\}_{n\geq 1}$. Applying the monotone convergence theorem, we would get that $\lim_{n\to\infty}\mathbb{E}[f_n(S_T)] = \mathbb{E}[f(S_T)]$. If f is continuous, we could choose a sequence of f_n that converges uniformly to f allowing us to interchange the limit in n and the derivative in θ. Note that we cannot do this if f is not continuous.

$$\frac{\partial}{\partial\theta}\mathbb{E}[f(S_T)] = \frac{\partial}{\partial\theta}\lim_{n\to\infty}\mathbb{E}[f_n(S_T)] = \lim_{n\to\infty}\frac{\partial}{\partial\theta}\mathbb{E}[f_n(S_T)]$$

Now as f_n is smooth enough, we can interchange the limit of the derivative with the expectation and then again the limit and the derivative to get

$$\frac{\partial}{\partial\theta}\mathbb{E}[f(S_T)] = \lim_{n\to\infty}\frac{\partial}{\partial\theta}\mathbb{E}[f_n(S_T)] = \mathbb{E}\left[\lim_{n\to\infty}\frac{\partial}{\partial\theta}f_n(S_T)\right] = \mathbb{E}\left[\frac{\partial}{\partial\theta}f(S_T)\right] \,.$$

Sufficient conditions to apply the pathwise derivative method are described in [1] (chapter 7.2.2). Using only the Black-Scholes model (1) which is very smooth w.r.t to $\theta = S_0, \sigma$, these sufficient conditions summarize as:

Theorem 1 *Let $f : \Omega \to \mathbb{R}$, ($\Omega \subseteq \mathbb{R}$ or $\mathbb{R}^m$ a domain) and suppose that the following conditions hold:*

1. *$\mathbb{P}(S_T \in D_f) = 1$, where $D_f := \{s : s \mapsto f(s)$ is differentiable$\}$.*
2. *$s \mapsto f(s)$ is Lipschitz.*

Then, $\frac{\partial}{\partial\theta}\mathbb{E}[f(S_T)] = \mathbb{E}\left[\frac{\partial}{\partial\theta}f(S_T)\right]$ (in the multivariate case, S_T would be a random vector as seen in section 3.3).

2.3 Likelihood ratio (LR)

One has that, $f(S_T) = f\left(S_0 \exp\left((r - \frac{1}{2}\sigma^2)T + \sigma W_T\right)\right)$ where $W_T \sim \mathcal{N}(0,T)$, hence $f(S_T)$ is a function of W_T and

$$I = \mathbb{E}[f(S_T)] = \int_{\mathbb{R}} f\left(S_0 \exp\left\{(r - \frac{1}{2}\sigma^2)T + \sigma w\right\}\right) p_{W_T}(w)dw$$

where p_{W_T} is the probability density function of $\mathcal{N}(0,T)$. Applying the change of variable $w = (\log(s/S_0) - (r - \sigma^2/2)T)/\sigma$, one gets

$$p_{S_T}(s) = \frac{1}{s\sigma\sqrt{2\pi T}} \exp\left\{-\frac{1}{2}\left(\frac{\log(s/S_0) - (r - \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}\right)^2\right\}. \tag{8}$$

If $s \mapsto f(s)$ does not depend on θ and if we can interchange the derivative and the integral we get:

$$\frac{\partial I}{\partial \theta} = \int_0^\infty f(s) \frac{\partial p_{S_T}(s)}{\partial \theta} ds = \mathbb{E}\left[f(S_T) \frac{\dot{p}_\theta(S_T)}{p_{S_T}(S_T)}\right] \tag{9}$$

where $\dot{p}_\theta(s) = \frac{\partial p_{S_T}(s)}{\partial \theta}$ and $\frac{\partial}{\partial \theta} \log p_{S_T}(s) = \frac{\dot{p}_\theta(s)}{p_{S_T}(s)}$, also known as the score function.
The following theorem is useful to prove that we can interchange the derivative and the integral (its proof can be found in multiple analysis books on Lebesgue integration).

Theorem 2 *Let $\Omega \subset \mathbb{R}^n$ open and $\varphi : \Omega \times \mathbb{R} \to \mathbb{R}$ a function such that*

- *For every $\theta \in [\theta_1, \theta_2]$, the function $s \mapsto \varphi(s,\theta)$ is integrable;*
- *for a.e. s, the function $s \mapsto \varphi(s,\theta)$ is differentiable on (θ_1, θ_2);*
- *there exists $\psi \in \mathcal{L}^1(\Omega, \mathbb{R})$ such that $|\frac{\partial \varphi(s,\theta)}{\partial \theta}| \leq \psi(s)$.*

Then $\theta \mapsto \int_\Omega \varphi(s,\theta)ds$ is differentiable on (θ_1, θ_2) and

$$\frac{\partial}{\partial \theta} \int_\Omega \varphi(s,\theta)ds = \int_\Omega \frac{\partial}{\partial \theta}\varphi(s,\theta)ds\ .$$

One then applies this theorem to $\varphi(s,\theta) = f(s)p_{S_T}(s,\theta)$ to verify if one can use the LR method. Observe that thanks to the smoothness w.r.t. $\theta = S_0, \sigma$ of p_{S_T} (cf. (8)), it is very likely that this theorem is verified, it is for example the case for a f not depending on θ satisfying the integrability condition.

Generally speaking, the derivative of the score function for $\theta = S_0, \sigma$ is of the following form :

$$\frac{\dot{p_\theta}(s)}{p_{S_T}(s)} = \frac{\partial}{\partial \theta}(\log p_{S_T}(s)) = \frac{\partial}{\partial \theta}(-\log(s\sigma\sqrt{2\pi T})) - \frac{1}{2}\frac{\partial}{\partial \theta}\left(\frac{\log(s/S_0) - (r - \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}\right)^2$$

we then have

$$\frac{\dot{p_{S_0}}(s)}{p_{S_T}(s)} = -\left(\frac{\log(s/S_0) - (r - \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}\right)\frac{1}{\sigma\sqrt{T}}\frac{\partial}{\partial S_0}\log(\frac{s}{S_0}) = \frac{1}{S_0\sigma\sqrt{T}}\left(\frac{\log(s/S_0) - (r - \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}\right) \tag{10}$$

and after some calculations

$$\frac{\dot{p_\sigma}(s)}{p_{S_T}(s)} = \frac{-1}{\sigma} - \left(\frac{\log(s/S_0) - (r - \frac{1}{2}\sigma^2)T)}{\sigma\sqrt{T}}\right)\left(\frac{rT - \log(s/S_0)}{\sigma^2\sqrt{T}} + \frac{\sqrt{T}}{2}\right). \tag{11}$$

We can now estimate the quantity $\frac{\partial I}{\partial \theta}$ simulating various values of S_T and then using the Crude Monte-Carlo estimator given in (4) as in the previous cases.

3 Estimation of Greeks for different derivative contracts

3.1 European Call Option

We consider the discounted price $f(S_T) = e^{-rT}[S_T - K]^+$ of a european call option on an underlying with price S_t, $t \in [0, T]$ with strike price $K = 120$ and maturity $T = 1$. We also fix the current underlying price $S_0 = 100$, the risk-free interest rate $r = 0.05$ and the volatility of the underlying $\sigma = 0.25$.
Under the Black-Scholes model assumption, one finds $\mathbb{E}[f(S_T)] = S_0\Phi(d_1) - e^{-rT}K\Phi(d_1 - \sigma\sqrt{T})$ as done in [2] and differentiating w.r.t $\theta = S_0, \sigma$ gives the following exact formulae

$$\delta = \Phi(d_1), \quad \nu = S_0\sqrt{T}\phi(d_1), \quad \gamma = \frac{\phi(d_1)}{\sigma S_0\sqrt{T}}, \quad \text{where } d_1 := \frac{\log(S_0/K) + (r + \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}$$

and where $\Phi(\cdot)$ and $\phi(\cdot)$ are respectively the cdf and the pdf of the standard normal distribution.
We now estimate δ and ν, meaning that we estimate $\frac{\partial I}{\partial \theta}$ for $\theta = S_0, \sigma$. We use the finite difference, pathwise derivative and likelihood ratio (LR) methods and then the Crude Monte-Carlo estimator given in equation (4) (with a function that could and often will be different than the payoff f), and so with different sample sizes $N \in [10^3, 10^6]$.

3.1.1 Finite difference method

For the finite difference method, we estimate using (2)

$$\delta, \nu = \frac{\partial I}{\partial \theta} \approx \frac{I(\theta + \Delta\theta) - I(\theta - \Delta\theta)}{2\Delta\theta} = \mathbb{E}\left[\frac{f(S_T^{\theta+\Delta\theta}) - f(S_T^{\theta-\Delta\theta})}{2\Delta\theta}\right]$$

with $\Delta\theta = 10^{-6}$ and $\theta = S_0, \sigma$ for the Delta and Vega estimation respectively.

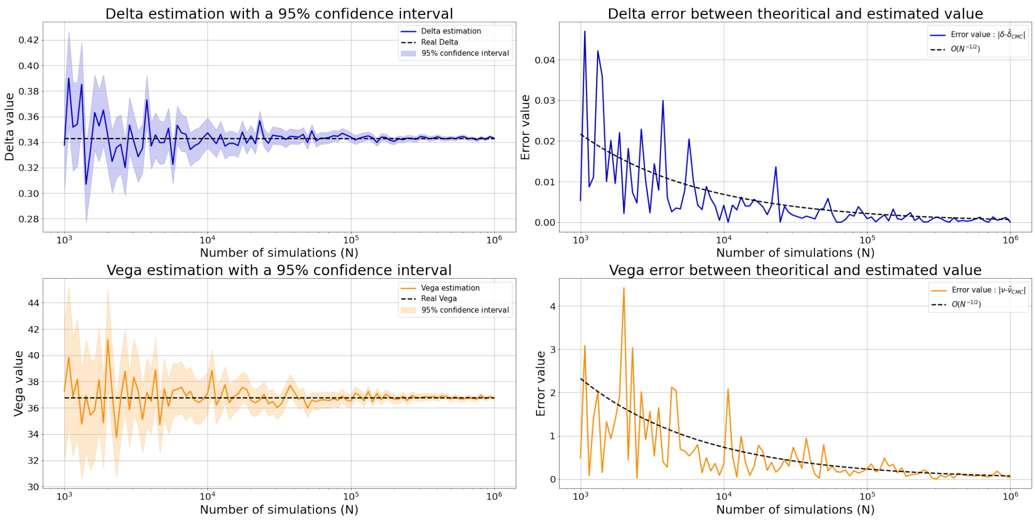

Figure 1: Delta and Vega estimation w.r.t. number of simulation (left column), and error bounds (right) with the finite differences method

As seen in Fig. 1, as the number of simulations (N) increases, both sensitivities converge to their theoretical B-S values. The convergence rate is of $\mathcal{O}(N^{-1/2})$ as expected for the Crude Monte-Carlo estimator (right column) [3].

One can also study how $\Delta\theta$ impact the estimation quality of the sensitivities by running simulations for every value of $\Delta\theta$ and comparing the bias for every estimation, as seen in Fig. 2. We note that the bias does not really change with respect to the chosen $\Delta\theta \in [10^{-12}, 10^{-1}]$ expect for the Vega bias which grows quite quickly for bigger values of $\Delta\theta$. However, the bias should theoretically grow in $\mathcal{O}(\Delta\theta)$. One possible explanation is that fresh samples are used for each $\Delta\theta$ value, but as we repeated the estimation

50 times and averaged the results, we do not think that this is the cause of the problem.

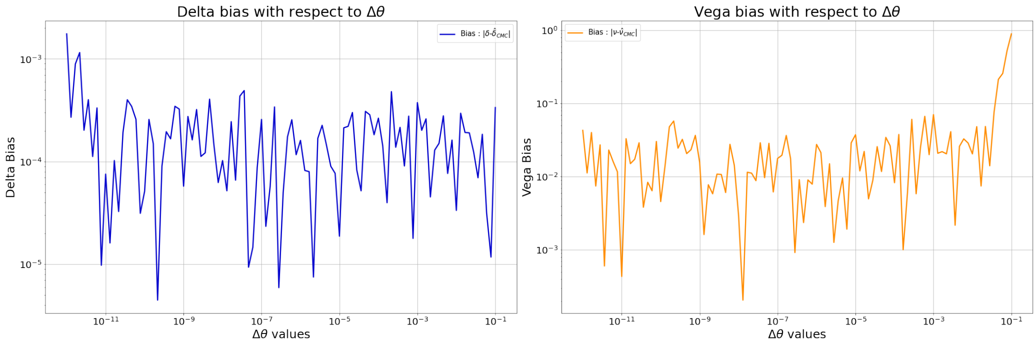

Figure 2: Evolution of Bias w.r.t $\Delta\theta$

Finally, as one has to compute samples for $S_{T,\theta\pm\Delta\theta}$ for the central finite difference, the question of creating two independant iid samples for W_T or using the same iid sample W_T for both final time underlying prices comes in place. A few arguments against the first case are described here after. Firstly, simulating two iid samples takes more computational power and therefore takes longer to run. For example, when computing the sensitivities Delta and Vega for $N \in [10^3, 10^6]$, the method with one idd sample took 1.5 seconds, while the other one took 5 seconds. Secondly, it does not mathematically make a lot of sense, as the whole idea of finite difference method is to fix any parameter except the one of interest, i.e S_0, σ for Delta and Vega respectively. This means that fixing W_T for $S_{T,\theta\pm\Delta\theta}$ makes more sense. Lastly, as described in [1, p. 380], independancy of W_T impacts the CMC estimator's variance. Indeed, one can write the variance of the estimator $\hat{\mu}_{CMC}$ as

$$\mathrm{Var}(\hat{\mu}_{CMC}) = \frac{\mathrm{Var}(f(S_{T,\theta+\Delta\theta}) - f(S_{T,\theta-\Delta\theta}))}{4N(\Delta\theta)^2} \tag{12}$$

with N being the number of simulations, f the payoff function and $S_{T,\theta}$ being the simulated underlying price at time T with a model using θ as one of its parameter, with all its other parameters fixed.
Three cases emerge from this variance:

1. The first one is assuming that $S_{T,\theta+\Delta\theta}$ is independent from $S_{T,\theta-\Delta\theta}$, i.e. that two independent samples W_T are drawn for each finite difference estimation. In this case, $\mathrm{Var}(f(S_{T,\theta+\Delta\theta})-f(S_{T,\theta-\Delta\theta}))$ is in $\mathcal{O}(1)$ with respect to $\Delta\theta$. This means that the estimator's variance $\mathrm{Var}(\hat{\mu}_{CMC}) \in \mathcal{O}\left(\Delta\theta^{-2}\right)$, which might grow quite quickly when $\Delta\theta \ll 1$.

2. The second case is taking the same iid sample W_T, which lands that $\mathrm{Var}(f(S_{T,\theta+\Delta\theta})-f(S_{T,\theta-\Delta\theta}))$ is in $\mathcal{O}(\Delta\theta)$. This gives us $\mathrm{Var}(\hat{\mu}_{CMC}) \in \mathcal{O}(\Delta\theta)$.

3. The third one is stricter than the second one, as it needs $f(S_{T,\theta})$ to be continuous a.s. in θ. If this is the case, then we get that $\mathrm{Var}(f(S_{T,\theta+\Delta\theta}) - f(S_{T,\theta-\Delta\theta}))$ is in $\mathcal{O}(\Delta\theta^2)$, and therefore that $\mathrm{Var}(\hat{\mu}_{CMC}) \in \mathcal{O}(1)$. In this project, one can observe that $f(S_T)$ is continuous a.s. with respect to S_0 and σ, meaning that we should get a variance of magnitude in $\mathcal{O}(1/N)$ when taking the number of simulations (N) into account.

Delta	Indep. W_T	Same W_T
Estimated variance	$7.8 \cdot 10^7$	$3.5 \cdot 10^{-7}$
Theoretical variance	10^6	10^{-6}

Vega	Indep. W_T	Same W_T
Estimated variance	$7.7 \cdot 10^7$	$6 \cdot 10^{-3}$
Theoretical variance	10^6	10^{-6}

Table 1: Finite difference estimator's variance w.r.t. independence of W_T, $\Delta\theta = 10^{-6}$, $N = 10^6$

As seen in Table 1, when using the same W_T for $S_{T,\theta\pm\Delta\theta}$ we match quite well the theoretical results for the Delta estimation while the Vega estimation variance is higher than expected. One could argue that as the Big-Oh notation is up to a constant, we could perfectly match the observed variance but we chose to keep it without any multiplicative constant. As for the case with two independent iid samples W_T, we observe a higher estimator's variance, even slightly bigger than the theoretically expected one, again up to a multiplicative constant. To wrap it up, using two independant iid samples for $S_{T,\theta\pm\Delta\theta}$ is not recommended and one should first sample a unique iid sample W_T and use it to simulate the prices S_T with variable θ parameters.

3.1.2 Pathwise derivative method

To apply the pathwise derivatives method, we observe that $f(s) = e^{-rT}[s-K]^+ = e^{-rT}(s-K)\mathbb{1}_{\{s>K\}}$ is differentiable everywhere but in $s = K$. Hence, $D_f = \mathbb{R} \setminus \{K\}$ and therefore $\mathbb{P}(S_T \in D_f) = 1$. Moreover, note that $\forall s, s' \in \mathbb{R} \quad |f(s) - f(s')| \leq e^{-rT}|s - s'|$ which means that $s \mapsto f(s)$ is Lipschitz. Hence by Theorem 1, one can interchange the derivative in θ and the expectation and thus use the pathwise derivatives method. The derivative of $f(s)$ is given by $f'(s) = e^{-rT}\mathbb{1}_{\{s>K\}}$ for $s \neq K$ and it is not defined otherwise.

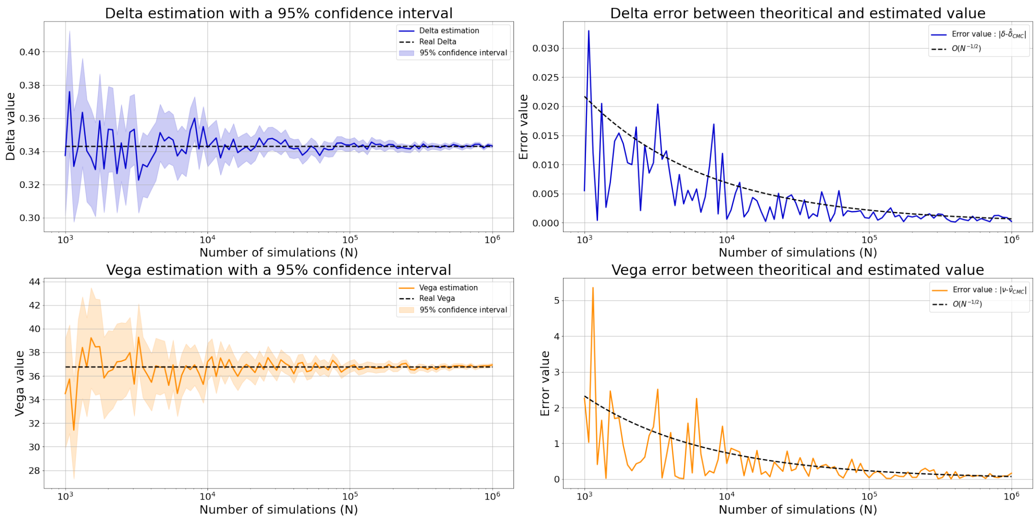

Figure 3: Delta and Vega estimation w.r.t. number of simulation (left column), and error bounds (right) with the pathwise derivative method

As seen in Fig. 3 and similarly to the finite difference method, when the number of simulations increases, both sensitivities converge to their theoretical B-S values. The convergence rate is again of $\mathcal{O}(N^{-\frac{1}{2}})$ as expected and the same error magnitude is observed than with the FD method.

3.1.3 Likelihood ratio method

As $f(s) = e^{-rT}[s-K]^+$ is independent of $\theta = S_0, \sigma$, Theorem 2 is verified and we can therefore apply the LR method estimating δ and ν using (9) with the score function given by (10) for δ and (11) for ν. As seen in Fig. 4, we observe the same trend as in Fig. 1 and 3. However, one can note that the scale of the δ and ν error is larger with the LR estimation than the two aforementioned methods. This shows that the LR method is less efficient when N is small.

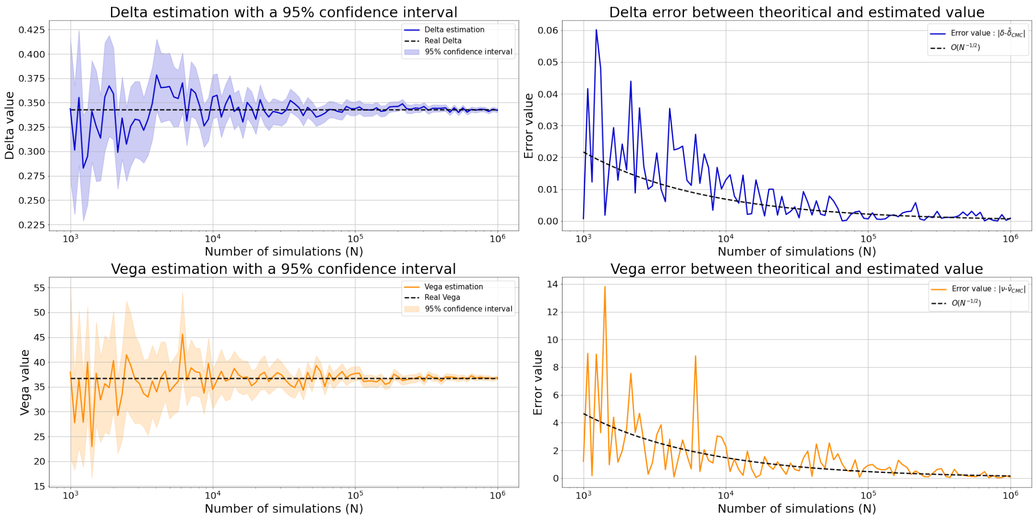

Figure 4: Delta and Vega estimation w.r.t. number of simulation (left column), and error bounds (right) with the likelihood ratio method

3.1.4 Estimation of γ

We are now interested in the estimation of $\gamma := \frac{\partial^2}{\partial S_0^2}\mathbb{E}[f(S_T)]$. We can apply the various combinations of the Likelihood ratio (LR) and pathwise derivative (PD) methods to write $\gamma = \mathbb{E}[\psi(W_T)]$ for some function ψ so that we can simulate $\psi(W_T)$ and then estimate γ with the Crude Monte-Carlo estimator. However, one could also use the finite differences method to estimate γ (with a first order derivative on $f'(S_T)$) by combining the FD method (2) with the LR or PD method.

LRLR: First, let us work using twice the LR method as follows. We recall that $f(s) = e^{-rT}[s-K]^+$ does not depend on S_0, also as $\frac{\dot{p}_\theta}{p_{S_T}}$ given in (10) is smooth enough, one can check that we can apply Theorem 2 to (9) and get

$$\frac{\partial^2 I}{\partial S_0^2} = \int_0^\infty f(s)\underbrace{\frac{\partial^2 p_{S_T}(s)}{\partial S_0^2}}_{:=\ddot{p}_{S_0}(s)}ds = \mathbb{E}\left[f(S_T)\frac{\ddot{p}_{S_0}(s)}{p_{S_T}(s)}\right] \tag{13}$$

where

$$\frac{\ddot{p}_{S_0}(s)}{p_{S_T}(s)} = -\frac{1}{S_0^2\sigma^2 T}\left(1+\log(\frac{s}{S_0}) - (r-\frac{1}{2}\sigma^2)T\right) + \left(\frac{1}{S_0\sigma\sqrt{T}}\left(\frac{\log(s/S_0)-(r-\frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}\right)\right)^2 \tag{14}$$

We can therefore estimate γ using the likelihood ratio method of order 1 with the new score function $\frac{\ddot{p}_{S_0}(s)}{p_{S_T}(s)}$. We obtain an estimation that is very close to the real value as we can see in Fig. 5.

LRPD: One can also apply the LR method followed by the PD one and vice-versa. Let us first apply the LR method:

$$\frac{\partial^2 I}{\partial S_0^2} = \frac{\partial}{\partial S_0}\mathbb{E}\left[f(S_T)\frac{\dot{p}_{S_0}(S_T)}{p_{S_T}(S_T)}\right]$$

Observing that the function $S_0 \mapsto f(S_T)\frac{\dot{p}_{S_0}(S_T)}{p_{S_T}(S_T)}$ is continuous and piece-wise differentiable, one can apply the rule of thumb to interchange the derivative and the expectation to get

$$\frac{\partial^2 I}{\partial S_0^2} = \mathbb{E}\left[\frac{\partial}{\partial S_0}\left(f(S_T)\frac{\dot{p}_{S_0}(S_T)}{p_{S_T}(S_T)}\right)\right] = \mathbb{E}\left[f'(S_T)\frac{\partial S_T}{\partial S_0}\frac{\dot{p}_{S_0}(S_T)}{p_{S_T}(S_T)} + f(S_T)\frac{\partial}{\partial S_0}\left(\frac{\dot{p}_{S_0}(S_T)}{p_{S_T}(S_T)}\right)\right].$$

With some basics computations one finds

$$\frac{\partial}{\partial S_0}\left(\frac{\dot{p}_{S_0}(S_T)}{p_{S_T}(S_T)}\right) = \frac{-1}{S_0}\frac{\dot{p}_{S_0}(S_T)}{p_{S_T}(S_T)}$$

which gives us a formula for γ that we can easily estimate:

$$\frac{\partial^2 I}{\partial S_0^2} = \mathbb{E}\left[\left(\frac{f'(S_T)S_T - f(S_T)}{S_0}\right)\frac{\dot{p}_{S_0}(S_T)}{p_{S_T}(S_T)}\right]. \tag{15}$$

PDLR: If we first apply the the PD method using (5) and (6), we have

$$\frac{\partial^2}{\partial S_0^2}\mathbb{E}[f(S_T)] = \frac{\partial}{\partial S_0}\mathbb{E}\left[f'(S_T)\frac{S_T}{S_0}\right] = \frac{\partial}{\partial S_0}\mathbb{E}[g(S_T)]$$

where $g(S_T) = f'(S_T)\frac{\partial S_T}{\partial S_0} = e^{-rT}1_{\{S_T \geq K\}}\frac{S_T}{S_0}$ satisfies the conditions of theorem 2. Hence, we apply the likelihood ratio to g instead of f:

$$\begin{aligned}\frac{\partial}{\partial S_0}\mathbb{E}[g(S_T)] &= \int_0^\infty se^{-rT}1_{\{s\geq K\}}\frac{\partial}{\partial S_0}\frac{p_{S_T}(s)}{S_0}ds = \int_0^\infty se^{-rT}1_{\{s\geq K\}}\frac{\dot{p}_{S_0}(s)S_0 - p_{S_T}(s)}{S_0^2}ds \\ &= \int_0^\infty g(s)\dot{p}_{S_0}(s)ds - \frac{1}{S_0}\int_0^\infty g(s)p_{S_T}(s)ds = \mathbb{E}\left[g(S_T)\left(\frac{\dot{p}_{S_0}(S_T)}{p_{S_T}(S_T)} - \frac{1}{S_0}\right)\right] \end{aligned} \tag{16}$$

which we can easily simulate with the CMC estimator.

PDPD: One could also think about applying the Pathwise derivative method twice. However, we cannot interchange the derivative and the expectation two times for the European Call option. Indeed, one has

$$\frac{\partial^2}{\partial S_0^2}\mathbb{E}[f(S_T)] = \frac{\partial}{\partial S_0}\mathbb{E}\left[f'(S_T)\frac{S_T}{S_0}\right] = \frac{\partial}{\partial S_0}\mathbb{E}[g(S_T)]$$

where $g(s) = f'(s)\frac{s}{S_0} = e^{-rT}\mathbb{1}_{\{s>K\}}\frac{s}{S_0}$ is defined for $s \neq K$. As g is not continuous w.r.t. s, the rule of thumb states that one cannot interchange the derivative in S_0 and the expectation. One can also see that $\frac{\partial}{\partial S_0}\mathbb{E}[g(S_T)] \neq \mathbb{E}[\frac{\partial}{\partial S_0}g(S_T)]$. Indeed, by estimating an approximation of the left-hand term with the finite differences method we see that it is non zero, while $\mathbb{E}\left[\frac{\partial}{\partial S_0}g(S_T)\right] = \mathbb{E}\left[f''(S_T)(\frac{\partial S_T}{\partial S_0})^2 + f'(S_T)\frac{\partial^2 S_T}{\partial S_0^2}\right] = 0.$

As seen in Fig. 5, all three methods described above converge to the theorietical B-S value of Gamma, with an order of convergence of $\mathcal{O}(N^{-1/2})$.

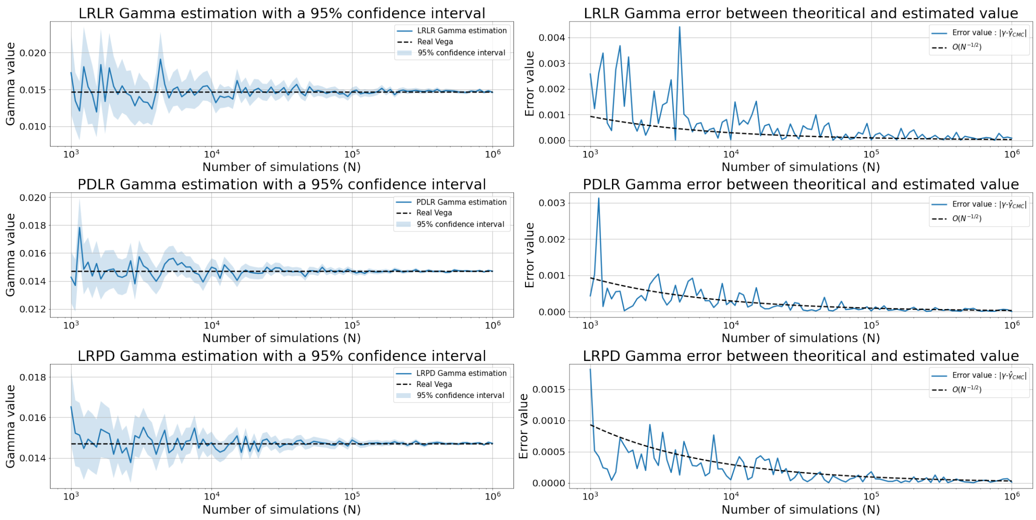

Figure 5: Gamma estimations w.r.t. number of simulation (left column), and error bounds (right) for different combinations of the LR and PD methods

Table 2 presents our estimates of gamma for the different combinations of the likelihood ratio (LR) and pathwise derivative (PD) methods. One can observe that LRPD and the PDLR give very close results.

Real value : $14.7 \cdot 10^{-3}$	**Estimation**	**Relative error**
LR-LR	$14.66 \cdot 10^{-3}$	0.9%
PD-LR	$14.73 \cdot 10^{-3}$	0.09%
LR-PD	$14.73 \cdot 10^{-3}$	0.09%

Table 2: Estimations of γ

3.2 Mixed estimators for Digital call option

We now consider the price of a Digital call option

$$f(S_T) = e^{-rT}\mathbb{1}_{\{S_T>K\}} \tag{17}$$

and want to find an estimate for δ. As the digital option payoff has a discontinuity at $S_T = K$, it is not differentiable at this point, and therefore one might have problems using the rule of thumb previously described to compute its sensitivities. To counter this problem, one can write

$$\mathbb{1}_{\{S_T>K\}} = f_\epsilon(S_T) + h_\epsilon(S_T) \text{ where } h_\epsilon(s) := (\mathbb{1}_{\{s>K\}} - f_\epsilon(s))$$

where $f_\epsilon(s) = \min\{1, \max\{0, \frac{s-K+\epsilon}{2\epsilon}\}\}$ is a continuous approximation of $\mathbb{1}_{\{s>K\}}$, for more clarity, we can also write

$$f_\epsilon(s) = \begin{cases} 0 & \text{if } s \le K-\epsilon \\ \frac{s-K+\epsilon}{2\epsilon} & \text{if } K-\epsilon \le s \le K+\epsilon \\ 1 & \text{if } s > K+\epsilon \end{cases} \qquad f'_\epsilon(s) = \begin{cases} 0 & \text{if } s \le K-\epsilon \\ \frac{1}{2\epsilon} & \text{if } K-\epsilon \le s \le K+\epsilon \\ 0 & \text{if } s > K+\epsilon \end{cases}$$

We can now express the Delta as

$$\delta = \frac{\partial}{\partial S_0}\mathbb{E}[f(S_T)] = e^{-rT}\left(\frac{\partial}{\partial S_0}\mathbb{E}[f_\epsilon(S_T))] + \frac{\partial}{\partial S_0}\mathbb{E}[h_\epsilon(S_T))]\right).$$

As f_e is continuous and piece-wise differentiable w.r.t. S_0, from the rule of thumb we can apply the pathwise derivative method to compute $\frac{\partial}{\partial S_0}\mathbb{E}[f_\epsilon(S_T)]$:

$$\frac{\partial}{\partial S_0}\mathbb{E}[f_\epsilon(S_T)] = \mathbb{E}\left[f'_\epsilon(S_T)\frac{\partial S_T}{\partial S_0}\right]$$

and as in the previous section, Theorem 2 applies and the LR method can be used to compute $\frac{\partial}{\partial S_0}\mathbb{E}[h_\epsilon(S_T)]$:

$$\frac{\partial}{\partial S_0}\mathbb{E}[h_\epsilon(S_T)] = \int_0^\infty h_\epsilon(s)\frac{\partial p_{S_T}(s)}{\partial S_0}ds = \mathbb{E}\left[h_\epsilon(S_T)\frac{\dot{p}_{S_0}(S_T)}{p_{S_T}(S_T)}\right].$$

We now have a expression of δ that we can estimate with two independent samples and the CMC estimator:

$$\delta = e^{-rT}\left(\mathbb{E}\left[f'_\epsilon(S_T)\frac{\partial S_T}{\partial S_0}\right] + \mathbb{E}\left[h_\epsilon(s)\frac{\dot{p}_{S_0}}{p_{S_T}}\right]\right) \tag{18}$$

with the following parameters : $\epsilon = 20$, $S_0 = K = 100$, $T = 0.25$, $\sigma = 0.25$ and $r = 0.05$.

As one might want to decrease the estimator's variance, we describe below the method to optimally choose the number of samples used respectively for the pathwise derivative and likelihood ratio method:

1. Given N random samples from S_T, one initially starts a pilot run with $m = \frac{N}{2}$ samples for the PD method, and $N - m$ for the LR method (note that this implies that the two estimators are independent), which gives us the total estimator's variance

$$\begin{aligned}\mathrm{Var}_{tot} &= \mathrm{Var}\left(e^{-rT}\left(\hat{\mu}_{PD}(f_\epsilon) + \hat{\mu}_{LR}(h_\epsilon)\right)\right) \\ &= e^{-2rT}\left[\frac{1}{m}\mathrm{Var}\left(f'_\epsilon(S_T)\frac{\partial S_T}{\partial S_0}\right) + \frac{1}{N-m}\mathrm{Var}\left(h_\epsilon(S_T)\frac{\dot{p}_{S_0}(S_T)}{p_{S_T}(S_T)}\right)\right].\end{aligned} \tag{19}$$

2. By differentiating Var_{tot} w.r.t. m, one can get the following optimal value for m:

$$m_{1,2}^* = N\frac{\text{Var}\left(f'_\epsilon(S_T)\frac{\partial S_T}{\partial S_0}\right) \pm \sqrt{\text{Var}\left(f'_\epsilon(S_T)\frac{\partial S_T}{\partial S_0}\right)\text{Var}\left(h_\epsilon(S_T)\frac{\dot{p}_{S_0}(S_T)}{p_{S_T}(S_T)}\right)}}{\text{Var}\left(f'_\epsilon(S_T)\frac{\partial S_T}{\partial S_0}\right) - \text{Var}\left(h_\epsilon(S_T)\frac{\dot{p}_{S_0}(S_T)}{p_{S_T}(S_T)}\right)} \tag{20}$$

then one can select optimal m which is eligible i.e. $m \in (0, N)$ and if both are eligible, one selects m such that it minimizes the original Var_{tot} function. Please note that $m_{1,2}^*$ will be minima of the function as Var_{tot} is convex $\forall m \in (0, N)$ as $\frac{\partial^2}{\partial m^2}\text{Var}_{tot} = \frac{2}{m^3}\text{Var}(f'_\epsilon(S_T)\frac{\partial S_T}{\partial S_0}) + \frac{2}{(N-m)^3}\text{Var}(h_\epsilon(S_T))\frac{\dot{p}_{S_0}(S_T)}{p_{S_T}(S_T)}) \geq 0$ for all $m \in (0, N)$.

As seen in Fig. 6, we observe a slow decrease from $\epsilon = 1$ to the minimum $\epsilon = 26$, then a slow increase as ϵ gets bigger. It is also worth noting that until $\epsilon = 8$, a lower total variance is obtained by only estimating Delta for the digital option with the LR method than doing the approximation with ϵ. For $\epsilon = 26$, we get the estimated δ value to be : 0.0315 and an estimator's variance in $\mathcal{O}(10^{-10})$, which is 30% lower than the one obtained with the initial $\epsilon = 20$.

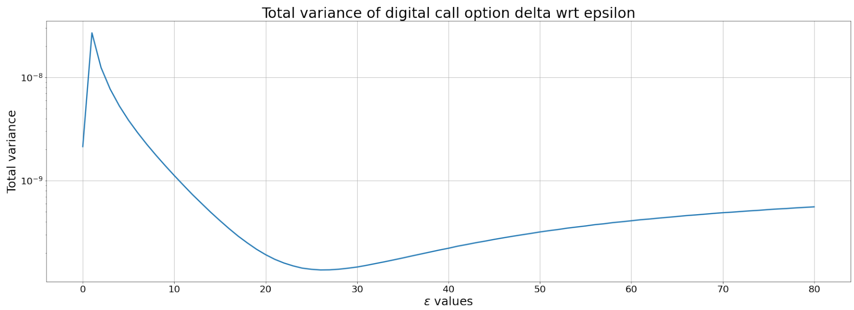

Figure 6: Total variance for digital option with respect to ϵ

3.3 Path-dependent option

We now consider an Asian option with payoff

$$f(\mathbf{S}) = f(S_{t_1}, ..., S_{t_m}) = \tilde{f}(\bar{S}_T) = e^{-rT}[\bar{S}_T - K]^+ \tag{21}$$

where $\mathbf{S} = (S_{t_1}, ..., S_{t_m})$ and $\bar{S}_T = \frac{1}{m}\sum_{i=1}^m S_{t_i}$ for $t_i = \frac{iT}{m}$, $i = 1, \dots m \in \mathbb{N}$, is called the discrete monitoring average in the time interval $[0, T]$. Observe that the payoff $f(\mathbf{S})$ depends on the path of the price S_t of the underlying stock. We are interested into applying the pathwise derivative and the LR methods to estimate $\delta = \frac{\partial}{\partial S_0}\mathbb{E}[f(\bar{S}_T)]$ and $\nu = \frac{\partial}{\partial \sigma}\mathbb{E}[\tilde{f}(\bar{S}_T)]$.

3.3.1 Pathwise derivative

Let us first work using the pathwise derivative method. We observe that $\bar{S}_T$ is also very smooth w.r.t. θ and one could easily show that Theorem 1 also applies for $\tilde{f}$ as a function of $\bar{S}_T$. This function verifies the conditions of Theorem 1 as shown in Section 3.1.2, hence we can interchange the expectation and the derivative as follow:

$$\frac{\partial}{\partial\theta}\mathbb{E}[f(\mathbf{S})] = \mathbb{E}\left[\frac{\partial}{\partial\theta}\tilde{f}(\bar{S}_T)\right] = \mathbb{E}\left[\tilde{f}'(\bar{S}_T)\frac{\partial\bar{S}_T}{\partial\theta}\right] \tag{22}$$

where $\tilde{f}'(x) = e^{-rT}\mathbb{1}_{\{x>K\}}$. We also find for $\theta = S_0$:

$$\frac{\partial\bar{S}_T}{\partial S_0} = \frac{\partial}{\partial S_0}\frac{1}{m}\sum_{i=1}^m S_{t_i} = \frac{1}{m}\sum_{i=1}^m\frac{\partial}{\partial S_0}S_{t_i} \overset{(6)}{=} \frac{1}{m}\sum_{i=1}^m\frac{S_{t_i}}{S_0} = \frac{\bar{S}_T}{S_0}\ . \tag{23}$$

For $\theta = \sigma$, we want to find a expression for $\frac{\partial \bar{S}_T}{\partial \sigma}$ that we can easily simulate. Using (7) for each $t \in [0, T]$ we can compute:

$$\frac{\partial S_t}{\partial \sigma} = (W_t - \sigma t) S_t = \frac{\log(S_t/S_0) - (r + \frac{1}{2}\sigma^2)t}{\sigma} S_t \ .$$

We can therefore write

$$\delta = e^{-rT}\mathbb{E}\left[\mathbb{1}_{\{\bar{S}_T > K\}} \frac{\bar{S}_T}{S_0}\right] \qquad \text{and} \qquad \nu = e^{-rT}\mathbb{E}\left[\mathbb{1}_{\{\bar{S}_T > K\}} \frac{1}{m}\sum_{i=1}^{m} \frac{\log(S_{t_i}/S_0) - (r + \frac{1}{2}\sigma^2)t_i}{\sigma} S_{t_i}\right] \quad (24)$$

which we can both easily estimate by simulating samples $W_{t_1}, \dots, W_{t_m}$ of a Brownian motion $\{W_t\}_{t \in [0,T]}$ giving us a simulation of each S_{t_i} with the formula (1) and then the CMC estimator.

3.3.2 Likelihood Ratio

To apply the likelihood ratio method, a little bit more work is required. We recall that we want to estimate $\frac{\partial}{\partial \theta}\mathbb{E}[\tilde{f}(\bar{S}_T)] = \frac{\partial}{\partial \theta}\mathbb{E}[f(S_{t_1}, \dots S_{t_m})]$.
First, observing that for any $t, l \geq 0$,

$$S_{t+l} = S_t e^{(r - \frac{1}{2}\sigma^2)l + \sigma(W_{t+l} - W_t)}$$

and using the fact that $\{W_t\}_t$ is a Brownian motion and so $W_{t+l} - W_t$ is independent of $\mathcal{F}_t = \sigma(W_\alpha \ : \ 0 \leq \alpha \leq t)$ (which is adapted to $\{S_t\}_{t \geq 0}$), one can see that $\{S_t\}_{t \geq 0}$ is a Markov Process. Not to overcharge the notation, we will now write $S_i = S_{t_i}$, $i = 1, ..., m$.
We can also write

$$\frac{\partial}{\partial \theta}\mathbb{E}[f(\mathbf{S})] = \frac{\partial}{\partial \theta}\int_{\mathbb{R}^m} f(s_1, \dots s_m) p_S(s_1, ..., s_m) ds_1 \cdots ds_m \quad (25)$$

where $p_S = p_{S_1,...,S_m}$ is the probability density function of the random vector $\mathbf{S}$. Using conditional densities, one has that

$$\begin{aligned} p_S(\mathbf{s}) &= p_{S_m|S_{m-1},...,S_1}(s_m|s_{m-1}, ..., s_1) p_{S_{m-1},...,S_1}(s_{m-1}, ..., s_1) \\ &= p_{S_m|S_{m-1}}(s_m|s_{m-1}) p_{S_{m-1},...,S_1}(s_{m-1}, ..., s_1) \\ &= p_{S_m|S_{m-1}}(s_m|s_{m-1}) \cdots p_{S_2|S_1}(s_2|s_1) \cdot p_{S_1|S_0}(s_1|s_0) \end{aligned}$$

where in the last two equalities we used the Markov property. We also recall that $S_0 \in \mathbb{R}$ is deterministic. With the same reasoning that gave us equation (8), we find that

$$p_{S_k|S_{k-1}}(s_k|s_{k-1}) = \frac{1}{s_k \sigma \sqrt{2\pi(t_k - t_{k-1})}} \exp\left\{-\frac{1}{2}\left(\frac{\log(s_k/S_{k-1}) - (r - \frac{1}{2}\sigma^2)(t_k - t_{k-1})}{\sigma\sqrt{t_k - t_{k-1}}}\right)^2\right\} \ . \quad (26)$$

Applying theorem Theorem 2 (the verification is similar to the one in section 3.1, we can rewrite equation (25) and write observing that $f(s_1, ..., s_m)$ is independent of θ

$$\frac{\partial}{\partial \theta}\mathbb{E}[f(\mathbf{S})] = \int_{\mathbb{R}} f(\mathbf{s}) \frac{\partial}{\partial \theta} p_{S_m|S_{m-1}}(s_m|s_{m-1}) \cdots p_{S_2|S_1}(s_2|s_1) \cdot p_{S_1|S_0}(s_1|s_0) d\mathbf{s} \ . \quad (27)$$

For $\theta = S_0$, one should note that $p_{S_k|S_{k-1}}(s_k|s_{k-1})$ is independent of S_0 for every $k = 2, ..., m$ and hence

$$\frac{\partial}{\partial \theta} p_{S_m|S_{m-1}}(s_m|s_{m-1}) \cdots p_{S_1|S_0}(s_1|s_0) = p_{S_m|S_{m-1}}(s_m|s_{m-1}) \cdots p_{S_2|S_1}(s_2|s_1) \frac{\partial}{\partial S_0} p_{S_1|S_0}(s_1|s_0) \ .$$

This gives

$$\delta = \frac{\partial}{\partial S_0}\mathbb{E}[f(\mathbf{S})] = \int_0^\infty f(\mathbf{s}) \frac{\partial}{\partial S_0} p_S(\mathbf{s}) d\mathbf{s} = \mathbb{E}\left[f(\mathbf{S}) \frac{\frac{\partial p_S}{\partial S_0}(\mathbf{S})}{p_S(\mathbf{S})}\right] = \mathbb{E}\left[f(\mathbf{S}) \frac{\dot{p}_{S_0}(S_1)}{p_{S_1}(S_1)}\right]$$

where $\dot{p}_{S_0}(s_1) = \frac{\partial}{\partial S_0} p_{S_1}(s_1)$ as S_0 is deterministic.

Using (10) with time t_1 and (21) we find

$$\delta = \mathbb{E}\left[e^{-rT}[\bar{S}_T - K]^+ \frac{1}{S_0\sigma\sqrt{T/m}}\left(\frac{\log(S_1/S_0) - (r - \frac{1}{2}\sigma^2)\frac{T}{m}}{\sigma\sqrt{T/m}}\right)\right] . \tag{28}$$

which can be estimated by simulating the samples $W_{t_1}, ..., W_{t_m}$ of a Brownian motion and then $\mathbf{S} = (S_{t_1}, ..., S_{t_m})$ with (1).

For $\theta = \sigma$, the evaluation of the score function follows the same principle, however every $p_{S_k|S_{k-1}}$ has a dependence on σ. We want to compute the score function

$$\frac{\dot{p}_\sigma(\mathbf{s})}{p_S(\mathbf{s})} = \frac{\partial}{\partial\sigma}\log p_S(\mathbf{s}) .$$

We compute:

$$\frac{\partial}{\partial\sigma}\log p_S(\mathbf{s}) = \sum_{k=1}^{m}\frac{\partial}{\partial\sigma}\log p_{S_k|S_{k-1}}(s_k|s_{k-1})$$
$$= \sum_{k=1}^{m}\frac{-1}{\sigma} - \left(\frac{\log(s_k/s_{k-1}) - (r - \frac{1}{2}\sigma^2)(t_k - t_{k-1}))}{\sigma\sqrt{t_k - t_{k-1}}}\right)\left(\frac{r(t_k - t_{k-1}) - \log(s_k/s_{k-1})}{\sigma^2\sqrt{t_k - t_{k-1}}} + \frac{\sqrt{t_k - t_{k-1}}}{2}\right) .$$

In the second equality we used equation (26) and (11). Finally we have:

$$\nu = \mathbb{E}\left[e^{-rT}[\bar{S}_T - K]^+ u(\mathbf{S})\right] \tag{29}$$

where the score function u is

$$u(\mathbf{S}) = \sum_{k=1}^{m}\frac{-1}{\sigma} - \left(\frac{\log(S_k/S_{k-1}) - (r - \frac{1}{2}\sigma^2)\frac{T}{m})}{\sigma\sqrt{T/m}}\right)\left(\frac{r\frac{T}{m} - \log(S_k/S_{k-1})}{\sigma^2\sqrt{T/m}} + \frac{\sqrt{T/m}}{2}\right) .$$

This expression can be estimated as we did for δ.

4 Conclusion

To conclude, we presented several methods to estimate financial sensitivities on various asset classes, i.e. the European Call option, the digital option and the asian option. Even though we got good numerical results, one could think of other computational methods to compute those sensitivites more efficiently. For instance, one could think about Control Variate methods to reduce the estimator's variance. One could also work on evaluating the performance of the different estimators for example analysing their variance, bias, convergence rate and computation cost. A quick numerical computation showed that with the same parameters as in section 3.1, the LR method estimators of δ and ν had a variance ten times higher that the estimators with the finite difference and pathwise derivative methods. A more detailed analysis would probably be interesting.

References

[1] Paul Glasserman. Estimating Sensitivities. In Paul Glasserman, editor, *Monte Carlo Methods in Financial Engineering*, Stochastic Modelling and Applied Probability, pages 377–420. Springer, New York, NY, 2003.

[2] EPFL FIN-415 Probability and Stochastic Calculus 2022-2023.

[3] Fabio Nobile. EPFL MATH-414 Stochastic Simulation Lecture Notes.

A Appendix - Code snippets

Hereafter are the main functions used in the notebook which can be found in the zip file attached in the submission. Each function has a self-explainatory docstring explaining its use, parameters and outputs. The other parts of the notebook are simply plotting cells and rewriting mathematical expressions which were discussed in the main report.

A.1 Prices simulation

```
def simul_S_T(params:list,n:int,W_T : np.array) -> np.array:
    """
    Returns a numpy array of size n with prices of the underlying at maturity

    Args :
        - params is a list of the following values :
            * T : Time of maturity in years (float)
            * S_0 : Underlying price at time 0 (float)
            * r : free-risk interest rate (float)
            * sig : Volatility of the underlying (float)
        - n : number of prices (int)
    Returns :
        - numpy array of S_T
    """
    T : float = params[0]; S_0 : float = params[1]
    r : float = params[2]; sig : float = params[3]
    return S_0*np.exp((r-0.5*sig**2)*T+sig*W_T)
```

A.2 Crude Monte Carlo Estimator

```
def CMC_estimator(func,X:np.array,params:list) -> list:
    """
    Crude Monte-Carlo estimator of E[func(X)]
    (Simulaton has to be done prior to using this function and stored in X)
    args:
        - func : function from R^N to R^N where N = length of X
        - X : numpy array
        - params : list of parameters for the func function
    returns :
        - Crude Monte-Carlo estimator of the mean
        - Standard deviation (for CI purposes)
    """
    return np.array([np.mean(func(X,params)),np.std(func(X,params))])
```

For the Crude Monte Carlo esitmator, we chose to not simulate the values of W_T inside the function as we would be using the same W_T for the finite difference function, and would therefore not want to simulate again the samples inside this function.

A.3 Finite Difference method

```
def finite_difference(func,params:list,dtheta:float,derivative:str, n:int, w_stud:bool,
    init_W_T=None) -> np.array:
    """
    Finite difference method to compute the derivative of I := E[func(S_T)] wrt theta

    I is computed with the CMC_estimator function which takes func, S_T and params as
    argument

    args:
        - func : function from R^N to R^N which takes S_T as parameter (payoff function)
        - params is a list of the following values :
            * T : Time of maturity in years (float)
            * S_0 : Underlying price at time 0 (float)
            * r : free-risk interest rate (float)
            * sig : Volatility of the underlying (float)
```

```
            * K : Strike price for the payoff
        - dtheta : small delta of theta to compute the finite difference (float)
        - derivative : either "delta", "vega" or "gamma"
        - n : number of simulations for Monte-Carlo
        - w_stud : True if study on diff W_T simul, false else
        - init_W_T : np array of W_T of size n
    returns : the estimated value of dI/dtheta by finite difference method and its standard deviation

    """

    if derivative not in ["delta","vega","gamma"]:
        print(f"Derivative not supported : {derivative}")
        raise ValueError

    idx_deriv : int = 3 if derivative == "vega"  else 1 # Useful to know which parameter we have to modify

    # Definition of new parameters
    params_pos : list = params.copy(); params_neg : list = params.copy()
    params_pos[idx_deriv] += dtheta ; params_neg[idx_deriv] -= dtheta

    # Generation of Brownian motion
    if w_stud:
        W_T_1 : np.array = st.norm.rvs(loc=0,scale=np.sqrt(params[0]),size=n)
        W_T_2 : np.array = st.norm.rvs(loc=0,scale=np.sqrt(params[0]),size=n)
        if derivative == "gamma":
            W_T_3 : np.array = st.norm.rvs(loc=0,scale=np.sqrt(params[0]),size=n)
            W_T : list = [W_T_1,W_T_2,W_T_3]
        else:
            W_T : list = [W_T_1,W_T_2]
    else:
        W_T_1: np.array = st.norm.rvs(loc=0,scale=np.sqrt(params[0]),size=n)
        W_T : list = [W_T_1,W_T_1]

    if init_W_T is not None: # Fixing the W_T for the delta theta question
        W_T : list = [init_W_T,init_W_T]

    diff_f = lambda x,p : func(x[0],p[0])-func(x[1],p[1]) # x = [S_T_pos,neg], p = [params_pos,neg]
    diff_f_2 = lambda x,p : func(x[0],p[0])-2*func(x[2],p[2])+func(x[1],p[1]) # x = [S_T_pos,neg,S_T], p = [params_pos,neg,params]

    # Generation of payoff estimates
    if derivative == "gamma": # Due to different equation for second order derivative estimates
        S_T_tot = [simul_S_T(params_pos,n,W_T[2]),simul_S_T(params_neg,n,W_T[0]),simul_S_T(params,n,W_T[1])]
        params_tot = [params_pos,params_neg,params]
        mu_std = CMC_estimator(diff_f_2,S_T_tot,params_tot)
        return mu_std/(dtheta**2)
    else: # For delta and vega
        S_T_tot = [simul_S_T(params_pos,n,W_T[0]),simul_S_T(params_neg,n,W_T[1])]
        params_tot = [params_pos,params_neg]
        mu_std = CMC_estimator(diff_f,S_T_tot,params_tot)
        return 0.5*mu_std/dtheta
```

A bit of explaination for the W_T's is necessary. First, if we want to simulate the original FD method, we have to set $w\_stud = False$, so that it generates only one sample W_T^1, which is then put into the array W_T for generalization purposes. If $w\_stud = True$, then we are generating two iid samples $W_T^{1,2}$ which was necessary to test the indepedancy of W_T for Section 3.1.1. This function returns the mean estimator as well as $\frac{\sqrt{\mathrm{Var}(I_{\theta+\Delta\theta}-I_{\theta+\Delta\theta})}}{2\Delta\theta}$ for Delta and Vega estimations. So to get the real standard deviation, one has to divide this result by $\sqrt{N}$ (which is done when computing confidence intervals for the plotting).

A.4 Pathwise derivative method

```
def pathwise_deriv(dfunc, params:list,dSdtheta,derivative:str, n:int) -> np.array:
```

```
    """
    1st order pathwise derivative method to compute the derivative of I := E[func(S_T)] wrt theta

    args:
        - dfunc : function from R^N to R^N which takes S_T as parameter (derivative of payoff function)
        - params is a list of the following values :
            * T : Time of maturity in years (float)
            * S_0 : Underlying price at time 0 (float)
            * r : free-risk interest rate (float)
            * sig : Volatility of the underlying (float)
            * K : Strike price for the payoff
        - dSdtheta : function of derivative of S_T wrt theta (takes W_T and parameters as input)
        - derivative : either "delta", "vega" or "combination". "combination" is used for LRPD for gamma estimation
        - n : number of simulations for Monte-Carlo

    returns : the estimated value of dI/dtheta by pathwise derivative and its standard deviation

    """

    if derivative not in ["delta","vega","combination"]:
        print(f"Derivative not supported : {derivative}")
        raise ValueError

    W_T : np.array = st.norm.rvs(loc=0,scale=np.sqrt(params[0]),size=n)
    S_T : np.array = simul_S_T(params,n,W_T)
    path_deriv = lambda x,params : dfunc(x,params)*dSdtheta(W_T,params)

    return CMC_estimator(path_deriv,S_T,params)
```

As the second order was not necessary in our project, we chose not to implement it for clarity purposes.

A.5 Likelihood ratio method

```
def likelihood_ratio(func,dp,params:list,derivative:str,n:int) -> np.array:
    """
    Likelihood ratio method to compute the derivative of I := E[func(S_T)] wrt theta

    args:
        - func : function from R^N to R^N which takes S_T as parameter (payoff function)
        - dp : function from R^N to R^N which takes S_T as parameter (partial deriv wrt theta of pdf of S_t)
        - params is a list of the following values :
            * T : Time of maturity in years (float)
            * S_0 : Underlying price at time 0 (float)
            * r : free-risk interest rate (float)
            * sig : Volatility of the underlying (float)
            * K : Strike price for the payoff
        - derivative : either "delta", "vega"
        - n : number of simulations for Monte-Carlo

    returns : the estimated value of dI/dtheta by pathwise derivative and its standard deviation

    """

    if derivative not in ["delta","vega", "gamma"]:
        print(f"Derivative not supported : {derivative}")
        raise ValueError

    W_T : np.array = st.norm.rvs(loc=0,scale=np.sqrt(params[0]),size=n)
    S_T : np.array = simul_S_T(params,n,W_T)
    f = lambda x,params : func(x,params)*dp(x,params)

    return CMC_estimator(f,S_T,params)
```