Projet de finances quantitatives

Oscar CLIVIO et Jean-Christophe CORVISIER

7 juin 2017

Introduction

Le but de notre projet est de calculer le prix d'une option de payoff $(K-\sum_{n=1}^{N} a_n S_t^n)_+$ et de maturité T avec les S_n des actifs risqués suivant le modèle de Black-Scholes et les a_n des réels positifs dont la somme vaut 1. Pour calculer le prix, nous utilisons la méthode de Monte-Carlo. Enfin, pour effectuer de la réduction de variance dans le calcul de ce prix, nous allons calculer de manière explicite le prix de l'option en changeant la moyenne arithmétique par la moyenne géométrique (c'est-à-dire remplacer $\sum_{n=1}^{N} a_n S_t^n$ par $\prod_{n=1}^{N}(S_t^n)^{a_n}$). Les prix seront donnés en euros.

1 Calcul théorique des prix des options

1.1 Modélisation théorique des actifs

Dans notre projet, nous supposons que les actifs risqués S_t^n sont générés par des modèles de black-Scholes avec des Browniens éventuellement corrélés. La formule de Black-Scholes nous donne que pour tout n allant de 1 à N, nous avons :

$$S_t^n = S_0^n \exp(\sigma_n W_t^n + (r - \frac{\sigma_n^2}{2})t)$$

avec les W_t^i des mouvements browniens que nous supposerons indépendants.

La filtration considérée $(\mathcal{F}_t), t \geq 0$ pour les calculs d'espérances conditionnelles, est la filtration générée par les $(W_t^1, ..., W_t^N), t \geq 0$. On rappelle alors que la tribu $\mathcal{F}_0$ est alors la tribu triviale.

1.2 Moyenne arithmétique

Comme nous l'avons vu dans le cours de finance, le prix dans l'univers risque-neutre de l'option de payoff $(K-\sum_{n=1}^{N} a_n S_n)_+$ et de maturité T est :

$$e^{-rT}\mathbb{E}[(K-\sum_{n=1}^{N} a_n S_T^n)_+]$$

Une formule explicite n'est pas exigée dans ce cas-là (a priori non triviale), nous calculerons donc le prix d'une telle option en utilisant la méthode de Monte-Carlo

1.3 Moyenne géométrique

Le prix de l'option dans l'univers risque-neutre en remplaçant la moyenne arithémtique par la moyenne géométrique dans la fonction de payoff est :

$$e^{-rT}\mathbb{E}[(K - \prod_{n=1}^{N}(S_T^n)^{a_n})_+]$$

Nous allons donner et démontrer une formule explicite pour le prix d'une telle option, ce qui nous permettra d'une part de comparer cela aux résultats obtenus par la méthode de Monte-Carlo, et d'autre part d'utiliser la moyenne géométrique comme variable de contrôle dans la simulation de Monte-Carlo pour le calcul de l'option avec moyenne arithmétique.

Le prix de l'option avec moyenne géométrique dans l'univers risque-neutre est donnée par la formule suivante :

$$\boxed{P = K\mathcal{N}(d) - e^{\mathbb{E}(Z_T)+\frac{Var(Z_T)}{2}}\mathcal{N}(d - \sqrt{Var(Z_T)})}$$

avec $\mathcal{N}$ la fonction de répartition de la loi normale $\mathcal{N}(0,1)$, $d = \frac{\log(K)-\mathbb{E}(Z)}{Var(Z_T)}$ et $Z_T = \sum_n^N a_n(\log(S_0^n) + \sigma_n W_T^n + (r - \frac{\sigma_n^2}{2})T))$

Démonstration :

Tout d'abord, calculons le terme $\prod_{n=1}^{N}(S_T^n)^{a_n}$.

$\prod_{n=1}^{N}(S_T^n)^{a_n} = \prod_{n=1}^{N}(S_0^n)^{a_n}\exp(\sum_n^N a_n(\sigma_n W_T^n + (r - \frac{\sigma_n^2}{2})T)) = \exp(Z_T)$

Or par indépendance des Browniens, on sait que Z_T suit une loi gaussienne, de paramètre $\mathbb{E}(Z_T)$ et $Var(Z_T)$.

On a les valeurs :

$$\mathbb{E}(Z_T) = \sum_n^N a_n(\log(S_0^n) - \frac{\sigma_n^2}{2})T) + rT$$

$$Var(Z_T) = \sum(a_n\sigma_n)^2 T$$

Par conséquent le calcul du prix revient à déterminer la valeur de $\mathbb{E}[(K - \exp(Z))_+]$ avec Z variable aléatoire suivant une loi gaussienne.

Calcul de $\mathbb{E}[(K - \exp(Z))_+]$ *:*

$\mathbb{E}[(K - \exp(Z))_+] = \frac{1}{\sqrt{2\pi Var(Z)}}\displaystyle\int_{-\infty}^{+\infty}\exp(-\frac{(x-\mathbb{E}(Z))^2}{2Var(Z)})(K - \exp(x))\mathbb{1}_{K\geq exp(x)}\,\mathrm{d}x$

En notant $A = \mathbb{E}[(K - e^Z)_+]$, on a alors :

$$\begin{aligned} A &= \frac{1}{\sqrt{2\pi Var(Z)}}\int_{-\infty}^{\log(K)} e^{-\frac{(x-\mathbb{E}(Z))^2}{2Var(Z)}}(K - e^x)\,\mathrm{d}x \\ &= \frac{1}{\sqrt{2\pi Var(Z)}}K\int_{-\infty}^{\log(K)}\exp(-\frac{(x-\mathbb{E}(Z))^2}{2Var(Z)})\,\mathrm{d}x - \frac{1}{\sqrt{2\pi Var(Z)}}\int_{-\infty}^{\log(K)}\exp(-\frac{(x-\mathbb{E}(Z))^2}{2Var(Z)})\exp(x) \end{aligned}$$

Calculons la valeur de

$$B = \frac{1}{\sqrt{2\pi Var(Z)}}\int_{-\infty}^{\log(K)}\exp(-\frac{(x-\mathbb{E}(Z))^2}{2Var(Z)})\,\mathrm{d}x$$

En posant $u := \frac{x-\mathbb{E}(Z)}{\sqrt{Var(Z}}$ et $d := \frac{\log K - \mathbb{E}(Z)}{\sqrt{Var(Z)}}$, on a

$B = \frac{1}{\sqrt{2\pi Var(Z)}} \displaystyle\int_{-\infty}^{\log(K)} \exp(-\frac{(x-\mathbb{E}(Z))^2}{2Var(Z)})\,\mathrm{d}x = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{d} \exp(-\frac{u^2}{2})\,\mathrm{d}u$

Soit $B = \frac{1}{\sqrt{2\pi Var(Z)}} \displaystyle\int_{-\infty}^{\log(K)} \exp(-\frac{(x-\mathbb{E}(Z))^2}{2Var(Z)})\,\mathrm{d}x = \mathcal{N}(d)$

Calcul du terme :

$$\frac{1}{\sqrt{2\pi Var(Z)}} \int_{-\infty}^{\log(K)} \exp(-\frac{(x-\mathbb{E}(Z))^2}{2Var(Z)}) \exp(x)\,\mathrm{d}x$$

que nous notons C

$$\begin{aligned}
C &= \frac{1}{\sqrt{2\pi Var(Z)}} \int_{-\infty}^{\log(K)} \exp(-\frac{(x-\mathbb{E}(Z))^2}{2Var(Z)} + x)\,\mathrm{d}x \\
&= \frac{1}{\sqrt{2\pi Var(Z)}} \int_{-\infty}^{\log(K)} \exp(-\frac{(x-\mathbb{E}(Z))^2 + 2Var(Z)x}{2Var(Z)})dx \\
&\text{avec } \frac{-(x-\mathbb{E}(Z))^2 + 2Var(Z)x}{2Var(Z)} = \frac{-(x-(\mathbb{E}(Z)+Var(Z)))^2 + Var(Z)^2 + 2Var(Z)\mathbb{E}(Z)}{2Var(Z)} \\
&= \frac{-(x-(\mathbb{E}(Z)+Var(Z)))^2}{2Var(Z)} + \mathbb{E}(Z) + \frac{Var(Z)}{2}
\end{aligned}$$

D'où $C = \frac{\exp(\mathbb{E}(Z)+\frac{Var(Z)}{2})}{\sqrt{2\pi Var(Z)}} \displaystyle\int_{-\infty}^{\log(K)} \exp(-\frac{(x-(\mathbb{E}(Z)+Var(Z)))^2}{2Var(Z)})\,\mathrm{d}x$

En effectuant le changement de variable $u = \frac{x-(\mathbb{E}(Z)+Var(Z))}{2Var(Z)}$, on a alors :

$C = \frac{1}{\sqrt{2\pi}} \displaystyle\int_{-\infty}^{d-\sqrt{Var(Z)}} \exp(-\frac{u^2}{2})\,\mathrm{d}u = \mathcal{N}(d-\sqrt{Var(Z)})$

En combinant B et C, on trouve alors que $\mathbb{E}[(K-\exp(Z))_+] = K\mathcal{N}(d) - \exp(\mathbb{E}(Z) + \frac{Var(Z)}{2})\mathcal{N}(d-\sqrt{Var(Z)})$

On obtient donc la formule du prix pour l'option avec moyenne géométrique :

$$P = K\mathcal{N}(d) - \exp(\mathbb{E}(Z_T) + \frac{Var(Z_T)}{2})\mathcal{N}(d-\sqrt{Var(Z_T)})$$

avec $d = \frac{\log(K)-\mathbb{E}(Z)_T}{Var(Z_T)}$, $\mathbb{E}(Z_T) = \sum_n^N a_n(\log(S_0^n) - \frac{\sigma_n^2}{2})T) + rT$, $Var(Z_T) = \sum(a_n\sigma_n)^2 T$ et $\mathcal{N}$ fonction de répartition de la loi normale centrée réduite.

2 Simulation numérique et résultats

Nous avons ensuite chercher à calculer les deux prix d'option (avec moyenne arithmétique et moyenne géométrique) en utilisant une méthode de Monte Carlo, c'est à dire utiliser la loi forte des grands nombres pour estimer les espérances que l'on cherche à déterminer. Nous avons procédé de la manière suivante : nous avons calculé par une méthode de Monte-Carlo la valeur de l'option avec moyenne géométrique, et avons comparé cette valeur à celle donnée par la formule que nous avons calculé précédemment, puis nous avons calculé à nouveau par une méthode de Monte-Carlo la valeur de l'option avec moyenne arithmétique, en utilisant l'estimateur du prix de l'option de moyenne géométrique comme variable de contrôle. Enfin nous avons comparé ces deux valeurs de prix.

2.1 Valeurs numériques prises pour la simulation

Nous avons pris les valeurs numériques suivantes pour nos simulations :

- $r = 0.05$
- $N = 5$
- $a_n = \frac{1}{5}, n = 1..5$
- $S_0^n = (10, 20, 30, 40, 50)$
- $\sigma_n = (0.1, 0.05, 0.06, 0.07, 0.1)$
- $T = 1.0$
- $K = 30$

2.2 Méthode de Monte Carlo sur option avec moyenne géométrique

L'idée de la méthode de Monte Carlo est la suivante : on simule un grand nombre de variables aléatoires M_i $i.i.d$ suivant la même loi que $(K - \prod_{n=1}^{N}(S_T^n)^{a_n})_+$, nous prenons leur valeurs, puis nous effectuons leur moyenne. La loi forte des grands nombres nous assure la convergence de $\frac{1}{N}\sum_{i=1}^{N} M_i$ vers $\mathbb{E}[(K - \prod_{n=1}^{N}(S_T^n)^{a_n})_+]$, et le théorème centrale limite nous donne des informations quant à la vitesse de convergence. (On multiplie évidemment par e^{-rT} la valeur trouvée par l'estimateur pour trouver le prix).

La valeur du prix théorique trouvée avec la formule établie en première partie est : $P_{geo} = 3.50096936047$

Pour une simulation de Monte Carlo de 50000 itérations, la valeur trouvée pour le prix de l'option avec moyenne géométrique est : $P_{MC} = 3.49717333538$

L'erreur relative de l'estimation de Monte Carlo (c'est à dire $\frac{\|P_{geo} - P_{MC}\|}{P_{geo}}$) est de 0.0011%.

La courbe suivante donne le tracé de la valeur du prix de l'option trouvée en utilisant la méthode de Monte Carlo, en fonction du nombre d'itérations effectué dans la simulation :

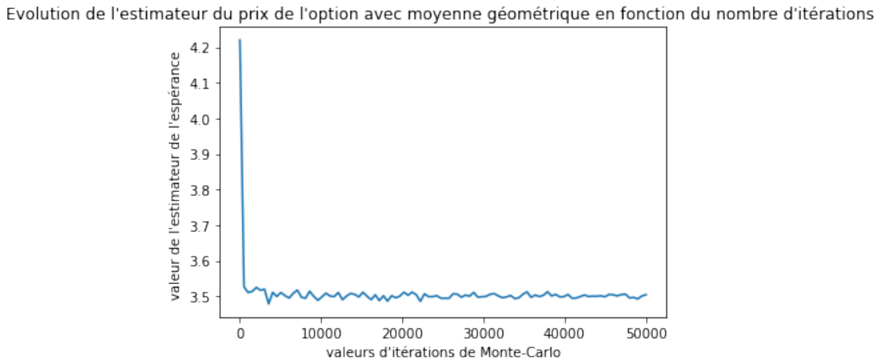

FIGURE 1 – Evolution de l'estimation du prix de l'option avec moyenne géométrique en fonction du nombre d'itérations de la simulation de Monte Carlo

Comme nous pouvons le voir sur la courbe, notre estimateur de Monte Carlo converge assez lentement vers la valeur théorique, au bout de 10000 itérations seulement les fluctuations autours de la valeur réelle devienne "raisonnables". La valeur obtenue à la fin est à 0.0011% de la valeur exacte, ce qui montre la puissance d'une telle méthode de simulation numérique.

2.3 Méthode de Monte Carlo sur option avec moyenne arithmétique

Le calcul de la valeur théorique est plus compliqué pour cette option, nous calculons donc cette valeur par la méthode de Monte Carlo, en utilisant la variable $(K - \prod_{n=1}^{N}(S_T^n)^{a_n})_+$ comme variable de contrôle.

On note N_{iter} le nombre d'itérations de la méthode, on pose $X := e^{-rT}(K - \prod_{n=1}^{N} a_n S_T^n)_+$ et $Y := e^{-rT}(K - \sum_{n=1}^{N} a_n S_T^n)_+$, et $(X_n)_{n=1..N_{iter}}$ et $(Y_n)_{n=1..N_{iter}}$ deux familles de variables aléatoires i.i.d. de même que loi que X et Y respectivement et telles que les réalisations $((X_n, Y_n))_{n=1..N_{iter}}$ sont indépendantes. La méthode de Monte-Carlo sans réduction de variance consiste à déterminer l'estimateur $\mu_n = \frac{1}{n}\sum_{k=1..n} Y_k$ pour $n = N_{iter}$, et avec réduction de variance à déterminer l'estimateur $\mu'_n = \frac{1}{n}\sum_{k=1..n} Y_k - b(\frac{1}{n}\sum_{k=1..n} X_k - \mathbb{E}(X))$ pour $n = N_{iter}$. En effet, $\mathbb{E}(X)$ est le prix de l'option avec moyenne géométrique et on cherche à estimer $\mathbb{E}(Y)$ qui est le prix de l'option avec moyenne arithmétique.

L'intérêt de la méthode ici est que si $Cov(XY) > 0$ alors $Var(\mu'_{N_{iter}}) < Var(\mu_{N_{iter}})$. En effet, à n fixé :

$$
\begin{aligned}
Var(\mu'_n) &= Var(\frac{1}{n}\sum_{k=1..n} Y_k - b(\frac{1}{n}\sum_{k=1..n} X_k - \mathbb{E}(X))) \\
&= Var(\mu_n) + \frac{b^2}{n}Var(X) - 2bCov(\frac{1}{n}\sum_{k=1..n} Y_k, \frac{1}{n}\sum_{k=1..n} X_k) \\
&\quad \text{où } Cov(\frac{1}{n}\sum_{k=1..n} Y_k, \frac{1}{n}\sum_{k=1..n} X_k) = \frac{1}{n^2}Cov(\sum_{k=1..n} Y_k, \sum_{k=1..n} X_k) = \frac{1}{n}Cov(Y, X) \\
&\quad \text{par indépendance des couples} (X_k, Y_k) \\
&= Var(\mu_n) + \frac{b^2}{n}Var(X) - \frac{2b}{n}Cov(Y, X)
\end{aligned}
$$

On remarque que $Var(\mu'_n)$ est un trinôme en b, minimal pour $b = b^* := \frac{Cov(X,Y)}{Var(X)}$.

$$\begin{aligned}
\text{Alors : } Var(\mu_n') &= Var(\mu_n) - \frac{Cov(X,Y)^2 Var(X)}{nVar(X)^2} \\
&= Var(\mu_n) - \frac{Cov(X,Y)^2 \frac{Var(Y)}{n}}{Var(X)Var(Y)} \\
&\quad \text{où } Var(\mu_n) = \frac{Var(X)}{n} \text{ et } \rho_{XY}^2 = \frac{Cov(X,Y)^2}{Var(X)Var(Y)} \\
&= Var(\mu_n)(1-\rho_{XY}^2)
\end{aligned}$$

En particulier, $Var(\mu'_{N_{iter}}) = Var(\mu_{N_{iter}})(1-\rho_{XY}^2)$

Or nous ne savons calculer ni $Cov(X,Y)$ ni $Var(Y)$, ce qui rend difficile le calcul exact de b^* et de $Var(\mu'_{N_{iter}})$.

Cependant, on peut montrer qu'un estimateur de b^* est défini, pour tout $n \in \mathbb{N}_{iter}$, par :

$$b_n = \frac{\sum_{k=1..n}(X_k - \mathbb{E}(X))(Y_k - \mu_n)}{(X_k - \mathbb{E}(X))^2}$$

et à l'aide du théorème central limite que, en posant $\mu_n'' = \frac{1}{n}\sum_{k=1..n} Y_k - b_n(\frac{1}{n}\sum_{k=1..n} X_k - \mathbb{E}(X))$, l'estimateur μ_n'' converge pour $n \to +\inf$ vers la même espérance et la même variance que μ_n'.

On utilise donc $\mu''_{N_{iter}}$ à la place de μ_n'.

Il nous reste à estimer $Var(\mu'_{N_{iter}})$. On reprend la formule $Var(\mu'_{N_{iter}}) = Var(\mu_{N_{iter}})(1 - \rho_{XY}^2) = \frac{Var(Y)}{N} - b^{*2}Var(X)$.

Or $Var(X) = e^{-2rT}\mathbb{E}[((K - \prod_{n=1}^N (S_T^n)^{a_n})_+)^2] - P^2$ où P est le prix de l'option avec moyenne géométrique, déjà calculé exactement. Cette variance peut se calculer exactement, et de même que pour le calcul de P, on obtient :

$$\begin{aligned}
\mathbb{E}[((K - \prod_{n=1}^{N}(S_T^n)^{a_n})_+)^2] &= K^2\mathcal{N}(d) + e^{2\mathbb{E}(Z_T)+2Var(Z_T)}\mathcal{N}(d - 2\sqrt{Var(Z_T)} \\
&\quad - 2Ke^{\mathbb{E}(Z_T)+Var(Z_T)/2}\mathcal{N}(d - \sqrt{Var(Z_T)}f)
\end{aligned}$$

toujours avec $\mathcal{N}$ la fonction de répartition de la loi normale $\mathcal{N}(0,1)$, $d = \frac{\log(K)-\mathbb{E}(Z)}{Var(Z_T)}$ et $Z_T = \sum_n^N a_n(\log(S_0^n) + \sigma_n W_T^n + (r - \frac{\sigma_n^2}{2})T))$

Pour obtenir $Var(Y)$, nous estimons $\mathbb{E}(Y^2)$ par Monte-Carlo sans réduction de variance et nous lui retranchons le carré de l'estimation du prix avec moyenne arithmétique selon la même méthode. De plus, nous remplaçons b^* par son estimateur b_N.

Pour une simulation de $N_{iter} = 50000$ itérations, nous trouvons une valeur de prix avec moyenne arithmétique sans utiliser de variable de contrôle de 0.269825364896. L'utilisation de la variable de contrôle nous donne quant à elle une valeur de prix qui est de 0.269802723685. L'écart relatif entre le prix estimé sans variable de contrôle, et celui avec variable de contrôle ($\frac{\|P_{MCsansVControle} - P_{MCavecVControle}\|}{P_{MCavecVControle}}$) est alors de 0.0001%. La variance calculée pour l'estimateur du prix sans utilisation de la variable de contrôle est de 0.000005282 , celle obtenue en utilisant la variable de contrôle est de 0.000005288. Comme nous pouvons le constater, la variance en utilisant la variable de contrôle est bien inférieur à celle obtenue sans variable de contrôle, mais la différence est minime. Ces résultats nous permettent de constater que la réduction de variance en utilisant $(K - \prod_{n=1}^{N}(S_T^n)^{a_n})_+$ comme variable de contrôle a un apport supplémentaire assez faible ici par rapport à la simple méthode de Monte Carlo si nous simulons pour un nombre assez important.

La courbe d'évolution de l'estimation du prix avec moyenne arithmétique sans variable de contrôle est donnée dans la figure suivante :

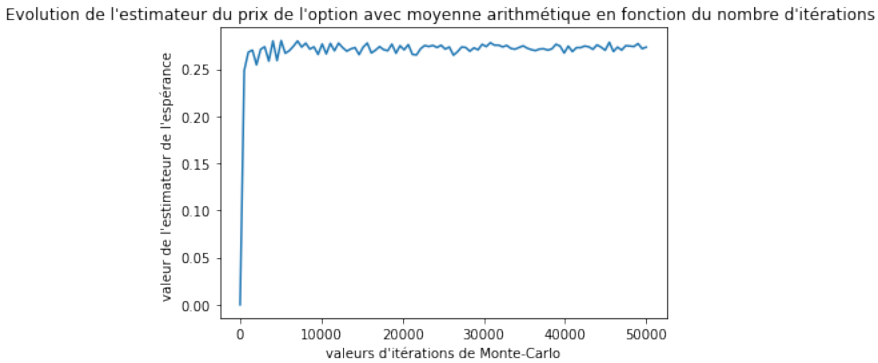

FIGURE 2 – Evolution de l'estimation du prix de l'option avec moyenne arithmétique en fonction du nombre d'itérations de la simulation de Monte Carlo sans réduction de variance

La courbe d'évolution de l'estimation du prix avec moyenne arithmétique en fonction du nombre d'itérations de la simulation de Monte Carlo avec réduction de variance est donnée dans la figure suivante :

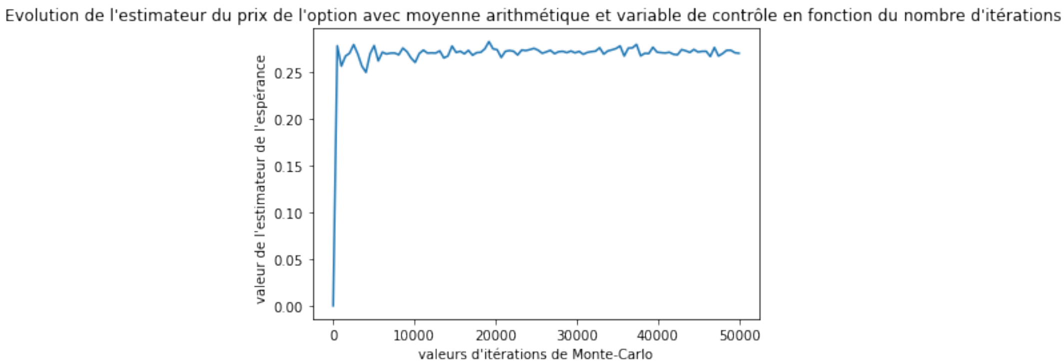

FIGURE 3 – Evolution de l'estimation du prix de l'option avec moyenne arithmétique en fonction du nombre d'itérations de la simulation de Monte Carlo avec réduction de variance

Nous pouvons que ces deux courbes sont extrêmement similaires, et qu'à partir de 10000 itérations, les estimateurs convergent vers le prix de l'option, fluctuant peu à partir de ce nombre d'itérations. On remarque également que la réduction de variance permet d'atteindre légèrement plus rapidement un encadrement restreint de la valeur du prix, par rapport à la méthode sans réduction de variance.

Ainsi, nous pouvons conclure que la méthode de Monte Carlo pour calculer la valeur du prix d'une option de payoff avec moyenne géométrique est assez adaptée et donne de bons résultats par rapport à la valeur théorique calculée. Cette même méthode nous permet de calculer la valeur du prix en remplaçant la moyenne géométrique par la moyenne arithmétique. On peut légèrement améliorer cet estimation en utilisant l'estimateur du prix de l'option avec moyenne géométrique des actifs comme variable de contrôle, cependant le gain apporté est assez faible. Notons également le prix plus important de l'option avec moyenne géométrique par rapport à l'option avec moyenne arithmétique.

3 ANNEXE

Nous joignons en annexe le code que nous avons implémenté pour réaliser nos simulations numériques :

```
# Modules importes

import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import minimize
from math import erf
from scipy.stats import norm

#fonction qui simule un brownien
def MouvementBrownien(t):
    if t>0.0:
        valeur_proba=np.random.normal(0,np.sqrt(t))
        return valeur_proba
    else:
        return 0

#fonction de payoff avec moyenne arithmetique
def payoff(K,liste_sigma,r,liste_S0,T,liste_an):
    Sum=0
    for i in range(len(liste_sigma)):
        Sum=Sum+calcul_actif(liste_S0[i],liste_sigma[i],r,T)*liste_an[i]
    return max(K-Sum,0)


#fonction de payoff avec moyenne geometrique
def payoffProd(K,liste_sigma,r,liste_S0,T,liste_an):
    Prod=1
    for i in range(len(liste_sigma)):
        Prod=Prod*np.exp(liste_an[i]*np.log(calcul_actif(liste_S0[i],liste_sig
    return max(K-Prod,0)


#retourne la valeur de l'actif risque
def calcul_actif(S0,sigma,r,t):
    Wt=MouvementBrownien(t)
    valeur_actif=S0*np.exp(sigma*Wt+(r - sigma**2 / 2)*t)
    return valeur_actif

#fonction qui retourne le prix de l'option avec moyenne geometrique
def prixExpliciteGeometrique(K,liste_sigma,r,liste_S0,T,liste_an):

    def normal(x):
        return norm.cdf(x);

    Esp = liste_an.T.dot(np.log(liste_S0) - liste_sigma**2 * T/2) + r*T;
```

```
    Std = np.sqrt(T*(liste_an**2).dot(liste_sigma**2));

    d = (np.log(K) - Esp) / Std;

    return np.exp(-r*T)*(K*normal(d)-np.exp(Esp+Std**2 / 2.)*normal(d -Std));

#fonction retournant la valeur exacte de la variance de l'option avec
#moyenne geometrique
def varianceExpliciteGeometrique(K,liste_sigma,r,liste_S0,T,liste_an):

    def normal(x):
        return norm.cdf(x);

    Esp = liste_an.T.dot(np.log(liste_S0) - liste_sigma**2 * T/2) + r*T;
    Std = np.sqrt(T*(liste_an**2).dot(liste_sigma**2));

    d = (np.log(K) - Esp) / Std;

    return np.exp(-2*r*T)*(K**2*normal(d)+np.exp(2*Esp+2*Std**2)*
    \n normal(d-2*Std) \n
    -2*K*np.exp(Esp+Std**2/2.)*normal(d-Std)) -\n
     prixExpliciteGeometrique(K,liste_sigma,r,liste_S0,T,liste_an)**2;



#valeur numeriques pour les simulations
an=np.array([1./5.]*5);
S0=np.array([10,20,30,40,50]);
K=max(np.prod(np.exp(np.log(S0)/5.)),an.T.dot(S0))
print(np.prod(np.exp(np.log(S0)/5.))*1.2)
print(an.T.dot(S0)*0.8)
K=31
N=50000
r=0.05
sigma=np.array([0.1,0.05,0.06,0.07,0.1]);
T=1.0

#fonction qui retourne les valeur de N realisations de l'option avec
#moyenne arithmetique
def realisationsMonteCarloArith(K,sigma,r,S0,T,an,N):
    realisations = []
    for i in range(N):
        realisations.append(np.exp(-r*T)*payoff(K,sigma,r,S0,T,an))
    return np.array(realisations)

#fonction qui retourne les valeurs de N realisations de l'option avec
#moyenne geometrique
def realisationsMonteCarloGeo(K,sigma,r,S0,T,an,N):
    realisations = []
    for i in range(N):
```

```
        realisations.append(np.exp(-r*T)*payoffProd(K,sigma,r,S0,T,an))
    return np.array(realisations)

#retourne l'estimateur de Monte Carlo avec les fonctions de realisations
#precedentes
def estimateurMonteCarlo(realisations):
    return realisations.mean();


#estimation empirique de la variance par Monte Carlo
def estimateurVarianceMonteCarlo(realisations, estimateurEsperance):
    return (realisations**2).mean() - estimateurEsperance**2;


#estimateur de la variable de controle
def estimateur_bn(realisationsGeo,realisationsArith,prixExactGeo,\n
        estimateurArith):
    diffGeo = realisationsGeo - prixExactGeo;
    return diffGeo.T.dot(realisationsArith-estimateurArith) \n
    /np.sum(diffGeo**2)

#estimateur du prix de l'option avec moyenne arithmetique par
#reduction de variance
def estimateurTotal(realisationsGeo,realisationsArith,prixExactGeo):
    estimateurGeo = estimateurMonteCarlo(realisationsGeo);
    estimateurArith = estimateurMonteCarlo(realisationsArith);
    estimateurbn=estimateur_bn(realisationsGeo,prixExpliciteGeometrique(K,\n
    liste_sigma,r,liste_S0,T,liste_an))
    return estimateurArith + estimateurbn*(estimateurGeo - prixExactGeo)

#fonction d'affichage de l'evolution des estimateurs des prix avec et sans
#reduction de variance, ainsi que l'evolution des variances calculees
def reductionVariance(K,sigma,r,S0,T,an,N):
    realisationsArith = realisationsMonteCarloArith(K,sigma,r,S0,T,an,N)
    realisationsGeo = realisationsMonteCarloGeo(K,sigma,r,S0,T,an,N);
    prixExactGeo = prixExpliciteGeometrique(K,sigma,r,S0,T,an)
    varianceExacteGeo = varianceExpliciteGeometrique(K,sigma,r,S0,T,an)

    esperanceInitiale = estimateurMonteCarlo(realisationsArith);
    estimateurbn = estimateur_bn(realisationsGeo,realisationsArith,\n
    prixExactGeo,esperanceInitiale)
    esperanceReduite = esperanceInitiale + \n
    estimateurbn*(estimateurMonteCarlo(realisationsGeo) - prixExactGeo)


    #print(estimateurbn)
    #print(estimateurVarianceMonteCarlo(realisationsGeo, \n
    estimateurMonteCarlo(realisationsGeo)))
    #print(varianceExacteGeo)
```

```
    varianceInitiale = estimateurVarianceMonteCarlo(realisationsArith,\n
    esperanceInitiale) / N;
    varianceReduite = varianceInitiale*(1-\n
    estimateurbn**2*varianceExacteGeo/(N*varianceInitiale));

    #print("Esperance initiale: ", esperanceInitiale)
    #print("Esperance reduite: ", esperanceReduite)
    #print("Variance initiale: ", varianceInitiale)
    #print("Variance reduite: ", varianceReduite)
    return esperanceReduite


#trace de la courbe d'evolution de l'estimation
#du prix avec moyenne geometrique
#en fonction du nombre d'iterations de Monte Carlo
vecteur_x=np.linspace(1,50001,100)
vecteur_y=[]

for i in range(100):
    vecteur_y.append(estimateurMonteCarlo(\n
    realisationsMonteCarloGeo(K,sigma,r,S0,T,an,int(vecteur_x[i]))))
plt.plot(vecteur_x,vecteur_y,linewidth=1.5, linestyle="-", \n
label=("trajectoire de l'estimateur de l'esperance"))
plt.ylabel("valeur de l'estimateur de l'esperance ")
plt.xlabel("valeurs d'iterations de Monte-Carlo")
plt.title("Evolution de l'estimateur du prix de l'option \n
avec moyenne geometrique en fonction du nombre d'iterations ")
plt.show()


#trace de la courbe d'evolution de l'estimation
#du prix avec moyenne arithmetique et reduction
#de variance
#en fonction du nombre d'iterations de Monte Carlo
vecteur_x=np.linspace(1,50001,100)
vecteur_y=[]

for i in range(100):
    vecteur_y.append(reductionVariance(K,sigma,r,S0,T,an,int(vecteur_x[i])))
plt.plot(vecteur_x,vecteur_y,linewidth=1.5, linestyle="-", \n
label=("trajectoire de l'estimateur de l'esperance"))
plt.ylabel("valeur de l'estimateur de l'esperance  ")
plt.xlabel("valeurs d'iterations de Monte-Carlo")
plt.title("Evolution de l'estimateur du prix de l'option \n
avec moyenne arithmetique et variable de controle en fonction du nombre d'ite
plt.show()



#trace de la courbe d'evolution de l'estimation
```

```
#du prix avec moyenne arithmetique sans reduction
#de variance
#en fonction du nombre d'iterations de Monte Carlo

vecteur_x=np.linspace(1,50001,100)
vecteur_y=[]

for i in range(100):
    vecteur_y.append(estimateurMonteCarlo(\n
    realisationsMonteCarloArith(K,sigma,r,S0,T,an,int(vecteur_x[i]))))
plt.plot(vecteur_x,vecteur_y,linewidth=1.5, linestyle="-", \n
label=("trajectoire de l'estimateur de l'esperance"))
plt.ylabel("valeur de l'estimateur de l'esperance ")
plt.xlabel("valeurs d'iterations de Monte-Carlo")
plt.title("Evolution de l'estimateur du prix de l'option \n
avec moyenne arithmetique en \n
fonction du nombre d'iterations ")
plt.show()
```