Thèse Master 1 : Pricing d'Options

Victor Mennesson, Alexandre Goldman, Baptiste Borne.

Encadrant : M. Rubenthaler

Introduction

Problématique Nous étudirons dans le cadre de ce mémoire différentes méthodes de pricing d'options :

- Evaluation d'options européennes et américaines par la méthode des arbres binomiaux
- Evaluation d'options européennes par la méthode de Monte Carlo
- Réduction de variance par fonction d'importance et variance de contrôle

Nous implémenterons ces méthodes numériques en C++, Python et R et en analyseront les résultats.

Références

[1] Cours "Méthode de Monte-Carlo", S. Rubenthaler

[2] Martingales pour la finance, C. Giraud

[3] TD "Méthode de Monte-Carlo pour le pricing d'Option", B. Lapeyre

[4] Options, Futures et autres actif dérivés, J.Hull

[5] Introduction aux processus stochastiques appliqués à la finance, Lamberton & Lapeyre

1 Evaluation d'options européennes et américaines par la méthode des arbres binomiaux :

1.1 Arbre binomiale à une période

Définition 1. Produits Dérivés Ce sont des titres qui dont la valeur dépend du cours d'un autre actif. Ce dernier est désigné par le terme sous-jacent.

Ici nous intéresserons aux options européennes et américaines d'achat et de vente.

Exemple. Option américaine L'option américaine d'achat (de vente) est un contrat qui donne à son détenteur le droit, et non l'obligation, d'acheter (de vendre) l'actif sous-jacent au contrat à toute date avant la date d'échéance N, au prix d'exercice fixé à l'avance.

Exemple 2. Option européenne L'option européenne d'achat (de vente) est un contrat qui donne à son détenteur le droit (et non l'obligation) d'acheter (de vendre) l'actif sous-jacent au contrat à la date d'échéance N, au prix d'exercice fixé à l'avance. Notons$S_n$le cours de l'actif sous-jacent au temps n. A l'échance N, nous avons deux cas :

- Soit $S_n <$ K : Alors le détenteur d'une option d'achat a le droit d'acheter l'actif sous-jacent au prix K, alors que cet actif sous jacent est au prix $S_n$sur le marché. Il n'exercera pas l'option.
  - Le détenteur de l'option de vente lui, a le droit de vendre cet actif au prix K. Le vendeur de l'option (la banque) va donc devoir acheter l'actif au prix K pour la revendre sur le marché au prix S_n. Cela revient à payer K-$S_n$au détenteur de l'option de vente.

— Soit $S_n \geqslant$ K Le détenteur de l'option d'achat, a le droit d'acheter l'actif au prix K. Le vendeur de l'option (la banque) va donc devoir lui vendre l'actif au prix K après l'avoir acheter sur le marché au prix S_n. Cela revient à payerS_n-K au détenteur de l'option d'achat.
  — Le détenteur d'une option de vente a le droit de vendre l'actif sous-jacent au prix K, alors que cet actif sous jacent est au prix $S_n$sur le marché. Il n'exercera pas l'option.

Fonction de paiment :

La valeur de l'option à l'échéance est appelée fonction de paiement (ou payoff). Etant le maximum de 0 ouS_n- K pour une option d'achat, son payoff est f =$(S_n-\text{K})_+$
De même, le payoff d'une option de vente est f = (K-$S_n)_+$

Une banque vend ces options contre une prime C et s'engage à payer la fonction de paiment (payoff). Pour se couvrir, elle ne peut s'assurer par diversification (principe des assurances) du fait de la forte corrélation des cours sur le marché financier. Ainsi elle doit couvrir chaque contrat individuellement. Nous allons détailler ici la méthode qui consiste à créer un portefeuille dont la valeur de départ est C et la valeur à l'échéance est égale à la fonction de paiement.

Nous allons d'abord devoir modéliser les marchés financiers. Les cours des sous-jacents des produits dérivés varie aléatoirement. Nous les modéliserons par une marche aléatoire à temps discret. Nous allons aussi émettre d'autres hypothèses, qui sont les hypothèses du modèle de Black et Scholes :

— L'absence des coûts de transaction,
— L'absence de dividende sur les titres,
— La possibilité d'emprunt illimité à un taux sans risque, constant et connu à l'avance,
— Les actifs sont liquides. Il y a toujours des vendeurs et des acheteurs pour tout produits,
— L'absence d'opportunité d'arbitrage

Définition 3. On dit qu'il existe une opportunité d'arbitrage s'il existe un portefeuille autofinancé Π tel que :

$X_0^{\Pi} = 0, \forall n \leq N, X_n^{\Pi} \geq 0$ et $\mathbb{P}(X_N^{\Pi} > 0) > 0$ avec$X_n^{\prod}$la valeur du portefeuille à la date n.

On peut traduire cette définition en disant qu'il existe une opportunité d'arbitrage dès lors qu'il est possible, en gérant un portefeuille autofinancé, de gagner de l'argent avec probabilité positive, tout en ayant aucun risque d'en perdre. Cette hypothèse est une hypothèse assez faible, en effet, s'il émerge une opportunité d'arbitrage, alors il y aura des investisseurs qui profiteront de cette arbitrage, jusqu'à ce que la possibilité d'arbitrage disparaisse.

Exemple. Option européenne avec un arbre binomiale à une période

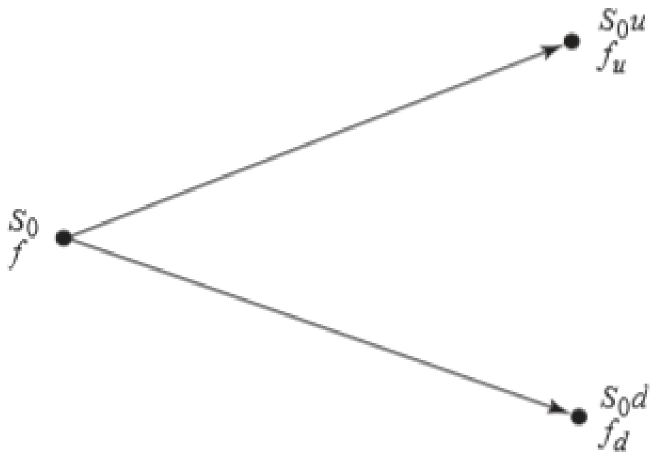

Figure 1.1 – Arbre Binomial à une période

Dans ce modèle, il y a deux dates, t=0 et t=N=1. Le cours de l'actif sous jacent est S_0à l'origine et peut prendre 2 valeurs à l'échéance,S_-ou S_+. On peut écrire la fonction de paiement comme f=g(S_N). Il est également possible d'emprunter ou d'investir dans l'actif sans risque au taux r.

On souhaite donc calculer le prix de l'option qui permettrait de couvrir l'option. Pour faire face au paiement de la fonction de paiement (f =(S_n−K)$_+$pour une option d'achat), le vendeur va acheter δ actifs sous-jacents S et β actifs sans risques de valeur B=1. Il constitue ainsi un portefeuille$\prod = (\beta, \delta)$ de valeur $X_0^{\prod} = \beta + \delta S_0$ à t=0 et$X_N^{\prod} = \beta(1+r) + \delta S_N$à t=N=1.

Définition 4. Marché Complet Le marché B-S est dit complet si pour toute variable aléatoire $f : \Omega \to \mathbb{R}^+$, il existe un portefeuille autofinancée Π qui réplique f à l'échéance N, i.e tel que $\forall \omega \in \Omega, f(\omega) = X_n^{\Pi}(\omega)$

Pour couvrir l'option, il faut que ce portefeuille est une valeur supérieure à f à l'échéance (Le cas $X_N^{\prod} > f$entraîne une opportunité d'arbitrage, que nous avons éliminés dans les hypothèses). Ainsi les deux valeurs que peut prendre l'actif sous-jacent S amène au système suivant :

$$\begin{aligned} g(S_-) &= \beta(1+r) + \delta S_- \\ g(S_+) &= \beta(1+r) + \delta S_+ \end{aligned}$$

dont la solution est $\delta = \frac{g(S_+) - g(S_-)}{S_+ - S_-}$ et $\beta = (1+r)^{-N} \frac{S_+ g(S_-) - S_- g(S_+)}{S_+ - S_-}$.

Ainsi le prix de la prime de l'option doit correspondre à la valeur du portefeuille initiale$X_0^{\prod} = \beta + \delta S_0$.

On a pu observer que l'évaluation d'une option à partir de des valeurs possibles du sous-jacent à la période suivante ne dépend pas de la probabilité que le sous-jacent atteigne ces valeurs. Soit $\mathbb{P}^*$la probabilité appelée risque-neutre définie comme :

$\mathbb{P}^*(S_N = S_+) = p^* = \frac{(1+r)^N S_0 - S_-}{S_+ - S_-}$ et $\mathbb{P}^*(S_N = S_-) = 1 - p^* = \frac{S_+ - (1+r)^N S_0}{S_+ - S_-}$.

Pour que $\mathbb{P}^*$soit bien une probabilité, il faut $S_-< S_0(1+r)^N< S_+$. On a choisi $\mathbb{P}^*$pour que $S_0= E^*[(1+r)^{-N}S^N]$

$$\begin{aligned}
E^*[(1+r)^{-N}S^N] &= (1+r)^{-N}S_+p^* + (1+r)^{-N}S_(1-p^*) \\
&= \frac{(1+r)^{-N}S_+((1+r)^N S_0-S_-) + 1+r)^{-N}S_-(S_+-1+r)^N S_0)}{S_+-S_-} \\
&= S_0
\end{aligned}$$

Faisons le lien entre la probabilité risque-neutre et l'évaluation du prix d'une option. Pour une option d'achat européenne, la fonction de paiement est g$(S_N) = (S_N - K)_+$. Si$S_- < K < S_+$on a $g(S_+) = S_+-K$et$g(S_-) = 0$, donc le prix de l'option est

$$\begin{aligned}
C &= \beta + \gamma S_0 \\
&= (1+r)^{-N}\frac{(S_+-\mathrm{K})}{S_+-S_-}S_- + \frac{(S_+-\mathrm{K})}{S_+-S_-}S_0 \\
&= \frac{(S_0 - (1+r)^{-N}S_-)(S_+-\mathrm{K})}{S_+-S_-}
\end{aligned}$$

Or la valeur moyenne de la fonction de paiement réactualisée$(1+r)^{-N}(S^N-K)_+$sous la probabilité $\mathbb{P}^*$

$$\begin{aligned}
E^*[(1+r)^{-N}(S^N - K)_+] &= (1+r)^{-N}(S^N - K)_+p^* \\
&= \frac{(S_0 - (1+r)^{-N}S_-)(S_+-\mathrm{K})}{S_+-S_-}
\end{aligned}$$

Ainsi le prix C de l'option est $C = E^*[(1+r)^{-N}(S^N - K)_+]$.

1.2 Le modèle de Cox, Ross et Rubinstein

Pour créer notre arbre binomiale, partons de la date t=0 et jusqu'à la maturité de l'option. À chaque période, le sous-jacent croit ou décroit en fonction d'un facteur u ou d avec 0<d<1<u . Ainsi, si le période à une période est S, il sera $S_u= S \times u$ ou $S_d= S \times d$ à la période suivante. Ces facteurs u et d sont calculés en prenant en compte la volatilité σ du sous-jacent, la durée ΔT de chaque étape . En partant du principe que la variance du log est $\sigma^2\Delta T$, on obtient :

$u = e^{\sigma\sqrt{\Delta T}}$ et $d = e^{-\sigma\sqrt{\Delta T}}$

A chaque dernier noeud, la valeur de l'option est $(S_n-\mathrm{K})_+$pour une option d'achat et $(K-S_n)_+$pour une option de vente. En partant des valeurs de l'option à maturité, on peut remonter un arbre binomiale jusqu'à la valeur de l'option à T=0.

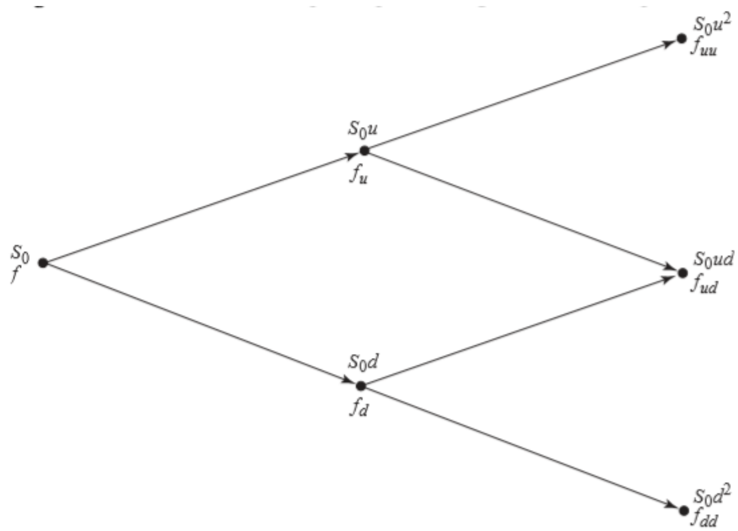

FIGURE 1.2 – Arbre Binomial à deux périodes

On appelle ici f_u et f_d la valeur de l'option quand le sous-jacent vaut $S_u$et S_d.

Construction du portefeuille de couverture et probabilité risque neutre

On considère une option d'achat européenne, dont la valeur à maturité (le payoff) est $(S_n$−K$)_+$. Pour calculer à la valeur de l'option à l'instant initiale, on reprend le raisonnement précédent qui consiste à prendre pour prime C de l'option la valeur initiale du portefeuille qui couvre l'option $X_0^{\prod}$. Nous procéderons par récurrence à chaque étape, pour trouver les valeurs de δ et β à chaque étape. Rappelons qu'à chaque étape, on peut trouver ces valeurs en fonction des valeurs de l'option et du sous jacent à la période suivante $\delta = \frac{X^{\Pi_+} - X^{\Pi_-}}{S_+ - S_-}$ et $\beta = (1+r)^{-N} \frac{S_+ X^{\Pi_-} - S_- X^{\Pi_+}}{S_+ - S_-}$. Si on applique cette méthode au modèle cox, ross, rubinstein on obtient $\delta = \frac{X^{\prod u} - X^{\prod d}}{S_u - S_d}$ et $\beta = e^{-r\Delta T} \frac{uX^{\prod d} - dX^{\prod u}}{S_u - S_d}$Si on calcule la valeur du portefeuille $X^{\prod} = \beta + \delta S$ en remplaçant β et δ on obtient alors $X^{\prod} = e^{-r\Delta T}(p^* X^{\prod u} + (1-p^*) X^{\prod d})$ avec $p^* = \frac{\exp(r\varDelta T) - d}{(u-d)} = \mathbb{P}^*(S_{n+1} = S_u)$

$[f_u \times p^* + f_d \times (1-p^*)]$ est l'espérance de la valeur future de l'option ce qui donne la valeur actuelle de l'option : $\exp(-r\varDelta T)[C_{uu} \times p^* + C_{ud} \times (1-p^*)]$ ce qui est la valeur actualisée au taux sans risque de l'espérance de la valeur future de l'option.

Ainsi pour une option d'achat sur 2 périodes, en partant des valeurs C_{uu}, C_{ud}, C_{dd} à T=2 on a à T=1 :

$$\begin{cases} C_u = & \exp(-r\varDelta T)[C_{uu} * p^* + C_{ud} * (1-p^*)] \\ C_d = & \exp(-r\varDelta T)[C_{uu} * p^* + C_{ud} * (1-p^*)] \end{cases} \text{avec } p^* = \frac{\exp(r\varDelta T) - d}{(u-d)}$$

$$C = \exp(-r\varDelta T)[C_u * p^* + C_d * (1-p^*)]$$

1.3 Implémentation de la méthode de l'arbre binomial

Nous avons structuré notre code C++ en utilisant deux classes, l'une pour les options de type américaine, l'autre pour les options de type européenne, héritant toutes les deux d'une classe "mère" abstraite : la méthode price est une méthode différente pour les deux types d'options, nous l'avons défini uniquement au niveau des classes filles. Pour la classe mère, cette méthode est une méthodes dite virtuelle pure. Par conséquent la classe mère est dite abstraite, on ne peut pas l'instancer, seulement ses classes héritées. La classe a pour utilité de structurer le programme et les objets en les rassemblant. On retrouve beaucoup de classes abstraites également dans la bilbiothèque standard.

La classe mère "BinomialTree" comporte 5 attributs, tout comme ces deux classes filles :

— So : le prix du sous-jacent à t=0
— K : le strike
— r : le taux sans risque
— T : la maturité
— σ : la volatilité
— N : le nombre de périodes

Ces attributs sont placés en portée "protected", cette portée n'a de sens que pour les classes qui se font hériter ; les éléments qui suivent ne sont pas accessibles depuis l'extérieur de la classe, sauf si c'est dans une classe fille. Plusieurs méthodes sont sont utilisées : des accesseurs pour les attributs, la méthode "price" (qui est la principale méthode, celle qui renvoie le prix de l'option à t=0), la méthode display qui renvoie les informations sur l'option choisie par l'utilisateur et le prix calculé, un constructeur, et des méthodes annexes comme par ex la méthode riskNeutralProb qui calcule simplement la proba risque neutre selon le modèle de Cox-Rox-Rubinstein en utilisant les données. Ci-dessous l'implémentation de la fonction price pour la classe "EuropeanOption"

```cpp
double EuropeanOption::price(){
    double prob = riskNeutralProb();

    vector<double> table(N+1);

    for (int compteur=0; compteur <=N; compteur++){
        table[compteur]=payOff(getStock(N-compteur,N),K,type);
    }
    for (int col=1;col<=N;col++){
        for (int index=0; index<=N-col; index++){
            table[index]=(prob*table[index]+(1-prob)*table[index+1])*exp(-r*T/N);
        }
        table.pop_back();
    }
    return table[0];
}
```

Figure 1.3 – Fonction Price pour la classe EuropeanOption sous C++

On déclare dans un premier temps un vector (tableau dynamique) de "double" de taille N+1. Ce tableau correspond à la dernière colonne de l'arbre binomial. On remplit alors ce tableau avec les valeurs des différents payoffs potentiels. Le premier élément tableau[0] correspond au payoff s'il n'y a eu que des hausses (N hausses et 0 baisses dans la trajectoire sur l'arbre), le dernier élément tableau[N] correspond au cas de N baisses. Ensuite on utilise deux boucles imbriquées pour "revenir en arrière" et calculer les colonnes (des payoffs) précédentes de l'arbre en utilisant pour chaque calcul la formule démontrée précédemment. Comme à chaque itération, on se place sur la colonne précédente, qui comporte 1 éléments de moins, on garde à chaque fois des informations inutiles pour les élements d'indices élevés. A chaque itération on garde une information inutile. On élimine alors le dernier élément du tableau à chaque itération, on réduit de 1 élément la taille du tableau.Ainsi, à la fin du calcul, on obtient un tableau à un seul élément : il correspond à l'unique élément de la première colonne de l'arbre. C'est le résultat de notre calcul pour le prix de l'option.

Notons ici que, si nous modélisions une évolution discrète du cours, le "compounding" se fait de manière continue à chaque calcul des payoffs. (? ? ? pas sur ...)

Dans la classe "AmericanOption" la méthode est similaire, mais pour les calculs des payoffs, il nous faut prendre la valeur maximale entre le payoff calculé avec la méthode précédente et le payoff si l'on décidait d'exercer l'option à ce moment là.

```cpp
double AmericanOption::price(){
    double  *v= new double[N+1]; // ptr vers un tableau de 'doubles' de N+1 valeurs
    // pointeur vers un tableau statique (unidimensionel)
    double Sij;                                   // cours option
    double p = riskNeutralProb();     // proba risque neutre

    // on calcule les payoffs finaux (dernière colonne de l'arbre)
    // on calcule en premier le payoff avec que des baisses puis on remonte jusqu'au payoff avec que des ups
    for(int j=0;j<=N;++j)  {      // boucle -> N+1 itérations (de 0 up à N up)
        *(v+j)= payOff(getStock(j,N), // getStock -> val de l'action après j up et n-j down
                       K,type);       // et on calcule le payoff final (fin de l'arbre) pour le cas j up n-j down
    }

    // Calcul  de la V.A. des pay-off récursivement en commençant du dernier élément du tableau au premier. (de i=n-1 à
        i=0)
    // on se place d'abord sur la colonne précédente, en diminuant i (i correspond au numéro de la période sur laquelle on
        se place, donc à la somme "dynamique" des ups and downs)
    // deuxième boucle : on commence à j=O (i.e 0 ups, que des downs -> dernière case de la colonne sur laquelle on se
        place)

    for(int i= N-1;i>=0;--i){
        for(int j=0;j<=i;++j){
            *(v+j)=(p*(*(v+j+1))+(1-p)*(*(v+j)))*exp(-r*T/N);
            Sij = payOff(getStock(j,i),K,type); // 2 lignes propres aux options américaines
            *(v+j)=(*(v+j)> Sij)?(*(v+j)): Sij; // on compare la valeur en utilisant la formule avec la valeur si on
                exerce maintenant (val intinsèque).
        }
    }
    return *v;
}
```

Figure 1.4 – Fonction Price pour la classe AmericanOption sous C++

On peut bien voir cette nouvelle étape dans le calcul ci-dessus, elle correspond aux deux dernières lignes. De plus, nous avons ici déclaré et manipulé nos variables à l'aide de pointeurs sur notre tableau. Remarquons que, lorsque l'on déclare un pointeur qui stockera l'adresse d'un tableau, on crée en fait des cases mémoires pour chaque élément du tableau. Le pointeur que l'on crée va en fait pointer sur l'adresse du premier élément (indice 0). Pour obtenir la valeur de celui-ci, il faudra donc écrire "*ptr", pour obtenir la valeur du $i^{ème}$élément du tableau (donc d'indice i-1), on écrira "*(ptr+i-1)" (cette syntaxe propre au tableau permet en fait de se déplacer de multiples du nombre d'octets correspondant à la taille d'un élement.)

Pour finir, l'utilisateur peut instancer dans le main la classe AmericanOption par exemple avec des paramètres choisis, et afficher, à l'aide de la méthode display, le prix calculé de l'option en rappelant ses caractéristiques.

Exemple : si on écrit dans le main : "AmericanOption optionA(50,52,2,0.05,0.7,"put",10000) ; optionA.display() ;", on obtient :

```
----------------------------
Current price    : 50 $
Strike           : 52 $
Maturity         : 2 years
Risk-free rate   : 0.05
Volatility       : 0.7
Number of steps  : 10000
============================
----------------------------
Value of American put : 17.7574 $
----------------------------
..................................
Calculation time: 9.17549 secondes
..................................
Program ended with exit code: 0
```

2 Evaluation d'Options Européennes par Monte Carlo

2.1 Prix d'une option européenne sous forme d'espérance

Marché Bond-Stock :

On se toujours dans le cas discret et on s'intéresse à un portefeuille constitué de deux actifs :

1. un actif sans rique B (d'évolution deterministique)
2. un actif risqué S (d'évolution stochastique)

On obtient alors le système suivant décrivant le passage d'un état à un autre pour les 2 actifs du portefeuille

$$\begin{cases} B_N = & (1+r_N)B_{N-1} \\ \Delta S_N = & (1+\rho_N)S_{N-1} \end{cases}$$

avec r_N le taux d'intérêt de B et ρ_N le rendement de S deux variables aléatoires.

Nous noterons $\mathcal{F}_N$ l'information dont on dispose au temps N. Alors $\mathcal{F}_N$=$\sigma(B_0,...,B_N,B_{N+1},S_0,...,S_N)$.

En effet, à l'instant N on connait B$_{N+1}$car on connait la valeur du taux d'interet r_N.

De façon immédiate, on obtient de la relation de récurrence la formule explicite suivante :

$$\begin{cases} B_N = & (1+r_N)\times...\times(1+r1)B_0 \\ S_N = & (1+\rho_N)\times...\times(1+\rho_1)S_0 \end{cases}$$

Définition 5 (Exponentielle stochastique)**.** Pour un processus $(X_N)_{N\geq 0}$,on associe l'exponentielle stochastique $\varepsilon_N(X)$ définie par $\varepsilon_0(X)=1$ et $\forall N>1, \varepsilon_N(X)=(1+\Delta X_1)\times...\times(1+\Delta X_N)$

En posant $O_N=\sum_{k=1}^{n} r_k$ et $A_N=\sum_{k=1}^{n}\rho_k$ alors d'après la définition ci-dessus il vient que :

$\varepsilon_N(O)=(1+\Delta O_1)\times...\times(1+\Delta O_N)=(1+r_1)\times...\times(1+r_N)$

et$\varepsilon_N(A)=(1+\Delta A_1)\times...\times(1+\Delta A_N)=(1+\rho_1)\times...\times(1+\rho_N)$

Dans la suite de notre mémoire nous utiliserons l'écriture suivante pour d'écrire l'évolution de nos actifs :

$$\begin{cases} B_N = & B_0\varepsilon_N(O) \\ S_N = & S_0\varepsilon_N(X) \end{cases}$$

Nous allons appliquer la méthode de Monte Carlo en temps discret, on modélise alors le cours du sous-jacent par une trajectoire binomiale.

Afin de pouvoir utiliser la méthode de Monte Carlo pour évaluer les options européennes, nous allons utiliser des raisonnements d'arbitrage et de couverture, en utiisant nottament le marché Bond-Stock, afin d'aboutir à l'écriture du prix d'une option européenne sous forme d'une espérance. $C = \mathbb{E}^*[e^{-rT}f(S_N)] = e^{-rT}E^*[f(S_N)]$ où la fonction payoff $f(S_T)$ vaut $(S_T - K)_+$pour un Call Européen et $(K - S_T)_+$ pour un Put Européen. En particulier, elle ne dépend que de la valeur de l'action à maturité.

—

Nous allons démontrer que le prix d'une option européenne peut s'écrire sous la forme $C = \mathbb{E}^*[e^{-rT}f(S_N)]$ où E^*est l'espérance sous la probabilité risque neutre $\mathbb{P}^*$que nous définirons. Cette méthode nous permettra d'exploiter la méthode de Monte Carlo pour simuler numériquement le prix d'une option européenne.

Définition 6. Probabilité Risque-Neutre On dit que P^* est une probabilité risque-neutre si :
- P^* est équivalente à la probabilité réelle, i.e. $\forall A \subset \Omega, P^*(A) = 0$ ssi $P(A) = 0$
- $(S_n/\varepsilon_n(U))_{n\leq N}$ est une martingale sous P^*

La proabilité risque-neutre est un pure outil de calcul, elle n'est pas observée sur les marchés. Le marché, cependant, évolue en moyenne de la même manière dans la réalité et sous cette probabilité risque-neutre.

Portfeuilles autofinancés :

On définit $\prod = (\beta_N, \delta_N)$ le portefeuille composé au temps N de β_N unités de la bond B et de δ_N unités du stock S. La valeur du portefeuille est alors définie de la manière suivante :

$X_N^{\prod} = \beta_N B_N + \delta_N S_N$

C'est au temps N que nous déterminons les nouveaux coefficients $\prod = (\beta_{N+1}, \delta_{N+1})$ du portfeuille . On considère que la valeurs du portefeuille avant et après réinvestisssent est la même, c'est à dire que :

$\beta_N B_N + \delta_N S_N = \beta_{N+1}B_N + \delta_{N+1}S_N$

Le portfeuille définit est un portfeuille dit autofinancé.

On vient de voir que la valeur d'un portfeuille autofinancé ne varie pas lors du réinvestissement,néanmoins celle ci varie entre deux instants consécutifs :

$$\begin{aligned}\Delta X_{N+1}^{\prod} = X_{N+1}^{\prod} - X_N^{\prod} = \beta_{N+1}B_{N+1} + \delta_{N+1}S_{N+1} - (\beta_N B_N + \delta_N S_N) &= \beta_{N+1}B_{N+1} + \delta_{N+1}S_{N+1} - (\beta_{N+1}B_N + \delta_N S_N) \\ &= \beta_{N+1}\Delta B_{N+1} + \Delta S_{N+1}\delta_{N+1}\end{aligned}$$

Proposition 7 (Autofinancement et Martingales)**.** *Supposons qu'il existe une probabilité risque neutre P^*.Alors, la valeur réactualisée $(\varepsilon_N(U)^{-1}X_N^{\prod})_{N\geq 0}$ d'un portfeuille autofinancé $\prod$est une martingales sous P^*.*

Démonstration. Réécrivons la valeur de notre portefeuille :

$$\varepsilon_N(O)^{-1}X_N^{\prod} = \beta_N\frac{B_N}{\varepsilon_N(O)} + \frac{S_N}{\varepsilon_N(O)}\delta_N = B_0\beta_N + \frac{S_N}{\varepsilon_N(O)}\delta_N$$

$$\begin{aligned}
E^*(\varepsilon_N(O)^{-1}X_N^{\prod}|F_{N-1}) &= E^*(B_0\beta_N + \frac{S_N}{\varepsilon_N(O)}\delta_N|F_{N-1}) \\
&= B_0\beta_N + \delta_N E_N^*(\frac{S_N}{\varepsilon_N(O)} \mid F_{N-1}) \text{ (car } \beta_N \text{ est } \delta_N \text{ sont } F_{N-1}\text{-mesurable)} \\
&= B_0\beta_N + \delta_N \frac{S_{N-1}}{\varepsilon_{N-1}(O)} \text{ (car } (\frac{S_N}{\varepsilon_N(O)})_{n<N} \text{ est une martingale sous } \mathbb{P}^*) \\
&= (B_0\varepsilon_{N-1}(O)\beta_N + \delta_N S_{N-1})\varepsilon_{N-1}(O)^{-1} \\
&= \varepsilon_N(O)^{-1}(\beta_N B_{N-1} + \delta_N S_{N-1}) \text{ (car } B_0\varepsilon_{N-1}(\mathrm{O}){=}B_{N-1})
\end{aligned}$$

Enfin, puisque l'on travaille avec un portefeuille autofinancé on sait que lors du réinvestissement :

$$\beta_N B_{N-1} + \delta_N S_{N-1} = \beta_{N-1}B_{N-1} + \delta_{N-1}S_{N-1}$$

Finalement,

$$\varepsilon_N(O)^{-1}(\beta_N B_{N-1} + \delta_N S_{N-1}) = \varepsilon_N(O)^{-1}(\beta_{N-1}B_{N-1} + \delta_{N-1}S_{N-1}) = \varepsilon_N(O)^{-1}X_{N-1}^{\prod}$$

et la valeur réactualisée $(\varepsilon_N(U)^{-1}X_N^{\prod})_{N\geq 0}$ d'un portefeuille autofinancé $\prod$est une martingales sous P^* :

$$E^*(\varepsilon_N(O)^{-1}X_N^{\prod} \mid F_{N-1}) = \varepsilon_N(O)^{-1}(\beta_N B_{N-1} + \delta_N S_{N-1}) = \varepsilon_N(O)^{-1}(\beta_{N-1}B_{N-1} + \delta_{N-1}S_{N-1}) = \varepsilon_N(O)^{-1}X_{N-1}^{\prod}$$

□

Définition 8. (On s'intéresse ici à l'évolution du marché jusqu'à une date N fixée.) On dit qu'il existe une opportunité d'arbitrage s'il existe un portefeuille autofinancé Π tel que :

$X_0^{\Pi} = 0$,$\forall n \leq N, X_n^{\Pi} \geq 0$ et $\mathbb{P}(X_N^{\Pi} > 0) > 0$.

On peut traduire cette définition en disant qu'il existe une opportunité d'arbitrage dès lors qu'il est possible, en gérant un portefeuille autofinancé, de gagner de l'argent avec probabilité positive, tout en ayant aucun risque d'en perdre.

Théorème 9. *Il n'existe pas d'opportunité d'arbitrage si et seulement si il existe au moins une probabilité risque-neutre.*

Définition 10. Marché Complet Le marché B-S est dit complet si pour toute variable aléatoire $f : \Omega \to \mathbb{R}^+$, il existe un portefeuille autofinancée Π qui réplique f à l'échéance N, i.e tel que $\forall\omega \in \Omega, f(\omega) = X_n^{\Pi}(\omega)$

Théorème 11. *Considérons un marché B-S sans opportunité d'arbitrage. On a l'équivalence :*
Le marché est complet $\iff$ *Il existe une unique probabilité risque-neutre* $\mathbb{P}^*$.

Proposition 12. *Supposons qu'il existe une probabilité risque neutre P^*. Alors la valeur actualisée $(\varepsilon_N(U)^{-1}X_n^{\Pi})_{n\geq 0}$ d'un portefeuille autofinancée Π est une martingale sous P^*.*

Définition 13. Portefeuille de couverture Un portefeuille Π est un portefeuille de couverture, si sa valeur à l'échéance est supérieure ou égale à la fonction de paiement f, i.e. si $\forall w \in \Omega, X_N^{\Pi}(w) \geq f(w)$

Supposons que le marché B-S est sans opportunité d'arbitrage et complet, il existe alors une unique probabilité risque-neutre P^*.

Prix d'une Option Européenne :

Théorème 14. *Le prix d'une option européenne de fonction de paiement f à l'échéance N est* $C = \mathbb{E}^*[\varepsilon_N(U)^{-1} f(S_N)]$.

Démonstration. Soit Π un portefeuille de couverture autofinancé quelconque. On sait alors que $(X_n^{\Pi}/\epsilon_n(U))_{n\leq N}$ est une martingale, donc

$$\mathbb{E}^*(X_N^{\Pi}/\epsilon_N(U)) = X_0^{\Pi}\,.$$

Or, comme Π est un portefeuille de couverture, on a

$$\forall w \in \Omega, X_N^{\Pi}(w) \geq f(w)\,.$$

. Par conséquent, $X_0^{\Pi} = \mathbb{E}^*(X_N^{\Pi}/\epsilon_N(U)) \geq \mathbb{E}^*(f/\epsilon_N(U))$

Ceci étant vrai pour tout portefeuille de couverture autofinancé, on a : $C = inf\{X_0^{\Pi}$ tel que Π est autofinancé et $\forall w \in \Omega, X_N^{\Pi}(w) \geq f(w)\} \geq \mathbb{E}^*(f/\epsilon_N(U)$. (1)

De plus, le marché étant complet, il existe un portefeuille autofinancé tel que $\forall w \in \Omega, X_n^{\Pi^*}(w) = f(w)$.

Π^* est autofinancé donc $(X_N^{\Pi *}\varepsilon_N(U)^{-1})_{n\leq N}$ est une martingale.

Donc, $X_0^{\Pi^*} = \mathbb{E}^*(X_N^{\Pi^*}/\epsilon_N(U)) = \mathbb{E}^*(f/\epsilon_N(U))$.

Comme Π^* est un portefeuille de couverture autofinancé, $X_0^{\Pi^*} \geq C$, d'où $C \leq \mathbb{E}^*(f/\epsilon_N(U))$ (2)

D'après (1) et (2), $C = \mathbb{E}^*(f/\epsilon_N(U))$ □

2.2 Principe de la méthode de Monte Carlo

— La méthode de Monte Carlo permet d'approcher une quantité que l'on peut écrire sous fome d'espérance d'une certaine variable aléatoire, disons $I = \mathbb{E}[X]$ avec X une variable aléatoire. Si on sait simuler des variables aléatoires $X_1, X_2, ...X_n$ i.i.d., on peut approcher I par la moyenne empirique des X_i : $I \approx \frac{X_1+X_2+...X_n}{n}$, avec n "grand", d'après la loi des grands nombres.

— Ici, on a vu que l'on pouvait écrire le prix d'une option européenne sous forme d'une certaine espérance d'une fonction de la variable aléatoire S_N, que l'on peut simuler en temps discret à l'aide d'un arbre binomial et en temps continu avec un mouvement brownien géométrique.

2.3 Implémentation de la méthode de Monte Carlo de réduction de variance

2.3.1 Méthode de réduction de variance par variable de contrôle

Nous allons exploiter la parité call-put de manière à réécrire le prix d'un call sous la forme "$\mathbb{E}[f(X) - h(X)] + \mathbb{E}[h(X)]$" de telle sorte que l'on sache calculer $\mathbb{E}[h(X)]$ et que $\mathbb{V}(f(X) - h(X)) \leq \mathbb{V}(f(X))$.

D'après la parité call-put, nous avons :

$$\begin{aligned}
C - P &= \mathbb{E}^*(f(S_N)/\epsilon_N(U)) - \mathbb{E}^*(f_2(S_N)/\epsilon_N(U)) \\
&= \mathbb{E}^*((f(S_N) - f_2(S_N))/\epsilon_N(U)) \\
&= \mathbb{E}^*((S_N - K)/\epsilon_N(U)) = S_0 - K\epsilon_N(U)^{-1}
\end{aligned}$$

(car $(S_n\epsilon_n(U)^{-1})$ est une martingale sous $\mathbb{P}^*$) avec $f(S_N) = (S_N - K)_+$ et $f_2(S_N) = (K - S_N)_+$.

Autrement dit :

$$
\begin{aligned}
C &= P + \mathbb{E}^*(h(S_N)/\epsilon_N(U)) \text{ avec h}(S_N)\text{=}(S_N\text{-K}) \\
&= \mathbb{E}^*(f_2(S_N)/\epsilon_N(U)) + \mathbb{E}^*(h(S_N)) \\
&= \mathbb{E}^*((f(S_N) - h(S_N))/\epsilon_N(U)) + \mathbb{E}^*(h(S_N)/\epsilon_N(U)) \\
&\quad (\text{car } f(S_N) - h(S_N) = (S_N - K)_+ + (S_N - K) = (K - S_N)_+ = f_2(S_N)
\end{aligned}
$$

Comme on a vu que $\mathbb{E}^*(h(S_N)/\epsilon_N(U)) = S_0 - K\epsilon_N(U)^{-1}$, on a bien ré-écrit notre call de manière à pouvoir exploiter la méthode de variable de contrôle.

On peut donc estimer le prix d'un call par $S_0 - K\epsilon_N(U)^{-1} + \frac{\epsilon_N(U)^{-1}}{n}\sum_{i=1}^{n}(f(S_N) - h(S_N)) \longrightarrow C$ p.s. d'après la loi forte des grands nombres.

Afin de s'assurer que cette méthode peut effectivement réduire la variance, prenons un exemple et comparons $\mathbb{V}((f(X) - h(X))\epsilon_N(U)^{-1})$ et $\mathbb{V}(f(X)\epsilon_N(U)^{-1})$. Pour des parmamètres donnés pour K, S_0, r ..., nous obtenons les histogrammes suivants (figure 2.1) pour 400 simulations du cours.

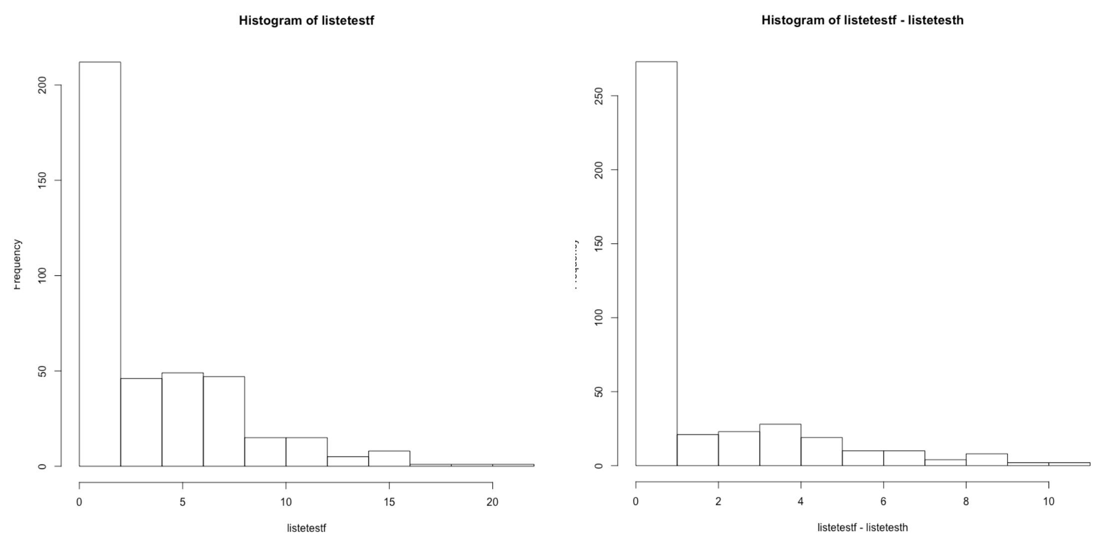

FIGURE 2.1 – Histogrammes obtenus pour 400 simulations des v.a. $f(S_N)$ et $f(S_N) - h(S_N)$ où S_N suit une trajectoire binomiale

On obtient des estimations des 2 variances de respectivement 16 et 5.1 pour et $\mathbb{V}(f(X))$ et $\mathbb{V}(f(X) - h(X))$.

En traçant l'estimateur d'un call(S=31, K=30, vol=0.2, r=0.05, nb s/-périodes=100,T=1) en fonction du nombre de variables simulés, on constate bien qu'en utilisant la variable de contrôle, on observe une erreur de convergence plus faible qu'avec l'estimateur de départ. L'erreur $\left|\frac{X_1 + \ldots X_n}{n} - \mathbb{E}[X_1]\right|$ étant d'ordre $\frac{\sigma}{\sqrt{n}}$ d'après le théorème central limite, comparer l'erreur de convergence entre les 2 méthodes revient à comparer $\mathbb{V}(f(X))$ et $\mathbb{V}(f(X) - h(X))$. Si cette dernière variance est plus faible, alors la convergence va se faire plus rapidement, comme on peut l'observer sur le graphe ci-dessous (rouge : avec variable de contrôle, bleu : sans réduction)

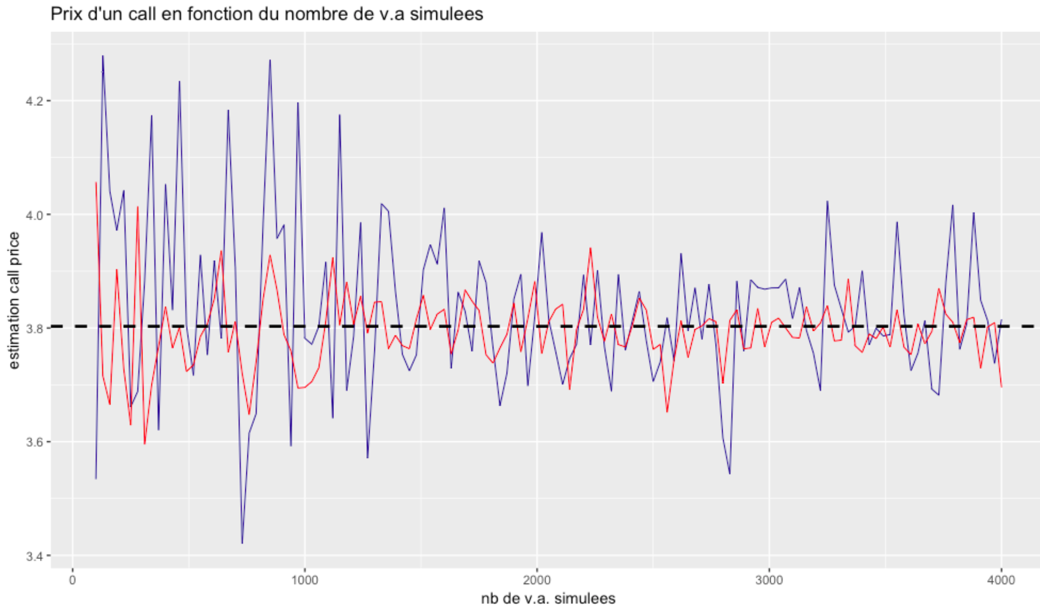

FIGURE 2.2 – Estimation du prix du Call(S=31, K=30, vol=0.2, r=0.05, nb s/-périodes=100,T=1) avec (rouge) et sans (bleu) réduction de variance sous R.

En fait, on ne réduira la variance que lorsque $f(S_N) = (S - K)_+$ et $h(S_N) = S - K$ sont suffisamment "proches". Intuitivement, cela se produit lorsque S_0 est suffisamment dans la monnaie (i.e. S_0 suffisamment au-dessus du strike pour un call). On obtiendra alors peu de valeurs nulles dans les calculs des "max" à partir des variables simulés et $f(S_N)$ et $h(S_N)$ seront alors souvent égaux. Par contre, lorsque S_0 est trop en-dessous du strike, les payoffs calculés à partir des simulations seront souvent négatifs et les deux fonctions ne seront plus "proches", on aura même tendance à avoir une variance plus élevées avec la méthode de la variable de contrôle, comme sur le graphe ci-dessous (en rouge le prix du call (S=25, K=30, vol=0.2, r=0.05, nb s/-périodes=100,T=1) estimé avec la variable de contrôle, en bleu sans réduction de variance). Cette méthode ne serra donc efficace que dans certains cas (S_0 assez dans la monnaie, faible volatilité ...). On obtient dans le cas ci-dessous, des estimations de la variance de 13.61 avec variance de contrôle et de 4.62 sans réduction de variance, donc une augmentation de variance en utilisant la variance de contrôle.

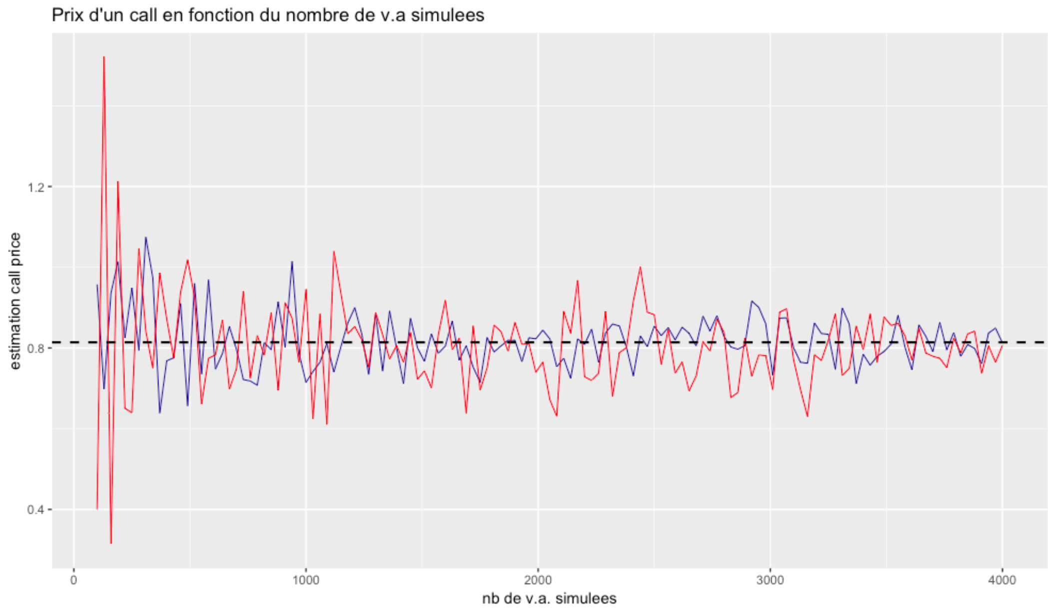

FIGURE 2.3 – Estimation du prix du Call(S=25, K=30, vol=0.2, r=0.05, nb s/-périodes=100,T=1) avec (rouge) et sans (bleu) utiliser la variable de contrôle sous R.

Dans le cas où on arrive effectivement à réduire la variance, on peut calculer un intervalle de confiance autour de l'estimateur du prix du call.

Calculons un intervalle de confiance à 95% autour de l'estimateur :

Onc herche $t \in \mathbb{R}$ tel que

$$\mathbb{P}(\left|\frac{X_1 + X_2 + ...X_n}{n} - \mathbb{E}(X_1)\right| \leq t) \geq 0.95$$

$$\mathbb{P}(\left|\frac{\sqrt{n}}{\sigma}(\frac{X_1 + X_2 + ...X_n}{n} - \mathbb{E}(X_1))\right| \leq t\frac{\sqrt{n}}{\sigma}) \geq 0.95$$

Donc par le théorème central limite,

$$\mathbb{P}(-t\frac{\sqrt{n}}{\sigma} \leq \mathcal{N}(0,1) \leq t\frac{\sqrt{n}}{\sigma}) \geq 0.95$$

$$\mathbb{P}(\mathcal{N}(0,1) \leq t\frac{\sqrt{n}}{\sigma}) \geq 0.975$$

On choisit alors $t\frac{\sqrt{n}}{\sigma} = 1.96$ i.e. $t = 1.96\frac{\sigma}{\sqrt{n}}$

On pourra plutôt choisir $\widetilde{t} = 1.96\frac{\hat{\sigma}}{\sqrt{n}}$avec $\hat{\sigma}$ l'estimateur déja présenté de la variance. On obtiendra toujours un intervalle de confiance à 95% (d'après le théorème de Slutsky.)

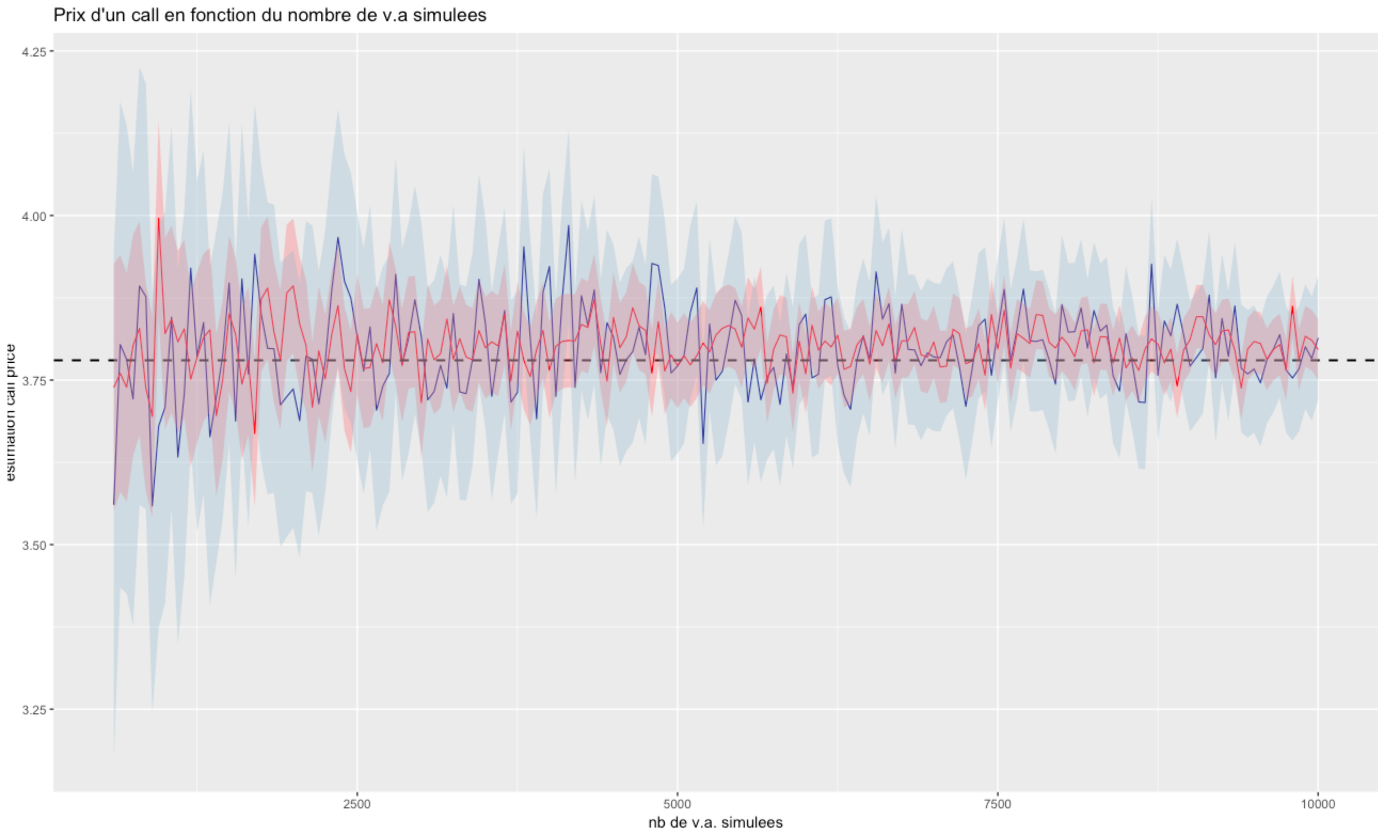

FIGURE 2.4 – Estimation du prix du Call(S=31, K=30, vol=0.2, r=0.05, nb s/-périodes=100,T=1) avec (rouge) et sans (bleu) réduction et intervalle de confiance à 95% sous R.

2.3.2 Méthode de réduction de variance par fonction d'importance

Dans le cadre avec arbre binomial, le cours de notre action peut être simulé par une fonction d'une variable aléatoire Y suivant une loi binomiale $\mathscr{B}(N,p)$ et dénombrant le nombre de mouvements up dans l'arbre.

N représente le nombre de sous périodes, soit le nombre de possibilités d'avoir une augmentation ou un diminution du cours. Aussi pour k mouvements up on effectuera N-k mouvements down.

p est la probabilité associée à un mouvement up dans cet arbre. (p ne doit pas être confondu avec la probabilité de hausse dans le monde réel).

Ansi, si Y$\sim\mathscr{B}(N,p)$ alors le cours de notre action est définit par $S_N = S_0 \times u^Y \times d^{N-Y} = S_N(Y)$.

On propose une famille de fonction d'importance : $\mathscr{F} = \{f_p, p \in [0,1]\}$ avec f_p la fonction de masse d'une loi binomiale $\mathscr{B}(N,p)$.

Dans cette famille, on va chercher la fonction d'importance, donc le paramètre p, qui permet de réduire le plus la variance.

Par ailleurs, on a :

$$\begin{aligned} C &= \mathbb{E}^*[(f/\epsilon_N(U)] = E^*[\phi(S_N)] = \sum_{k=0}^{N} \mathbb{P}_{p^*}(\mathscr{B}(N,p)=k) \times \phi(S_N(k)) \\ &= \sum_{k=0}^{N} \frac{\mathbb{P}_{p*}(\mathscr{B}(N,p)=k) \times \phi(S_N(k))}{P_p(\mathscr{B}(N,p)=k)} \times \mathbb{P}_p(\mathscr{B}(N,p)=k) \\ &= \mathbb{E}[\frac{f_*(Y) * \phi(S_N(Y))}{f_p(Y)}] \end{aligned}$$

où Y suit une $\mathscr{B}(N,p)$et f_* fct de masse de $\mathscr{B}(N,p^*)$ et f_p fct de masse de $\mathscr{B}(N,p)$

p^*est la probabilité risque neutre, $p \in [0,1]$ et $\phi = \frac{max(0,S_N-K)}{\epsilon_N(U}$.

On a obtenu une nouvelle méthode de Monte Carlo.

Notre objectif est de simuler pour différentes valeurs de p$\in$ $[0,1]$ des variables aléatoires ayant pour densité les fonctions fp correspondantes et ensuite de déterminer la fonction d'importance f_p (donc le p) qui minimise la variance de la méthode de Monte Carlo.

Nous approcherons cette variance théorique par l'estimateur

$$Var_{estimateur} = \frac{(\frac{f_*(Y_1)\times\phi(Y_1)}{f_p(Y_1)} - \frac{f_*(Y_2)\times\phi(Y_2)}{f_p(Y_2)})^2 + ... + (\frac{f_*(Y_{2n-1})\times\phi(Y_{2n-1})}{f_p(Y_{2n-1})} - \frac{f_*(Y_{2n})\times\phi(Y_{2n})}{f_p(Y_{2n})})^2}{2n}$$

Nous implémentons cette méthode en Python (pour un $Call(S_0 = 29, K = 30, vol = 0.2, N = 365, n = 1000, r = 0.05)$ voici les différents graphes obtenus pour les estimations de l'espérance et de la variance.

Nous avons fait plusieurs constats lors de cette implémentation.

Tout d'abord, pour des valeurs de p faibles, peu d'observations tomberont au-dessus du strike, nous obtiendrons souvent des termes nuls pour l'estimation du payoff. Un grand nombre de simulations permettrait de compenser ces termes nuls par des termes extrêmement élevés (les cas où on tombe au-dessus du strike) de part la probabilité très faibles au dénominateur. Ob observe que, dans ces conditions, la convergence est très lente. Pour des valeurs de p élevées,

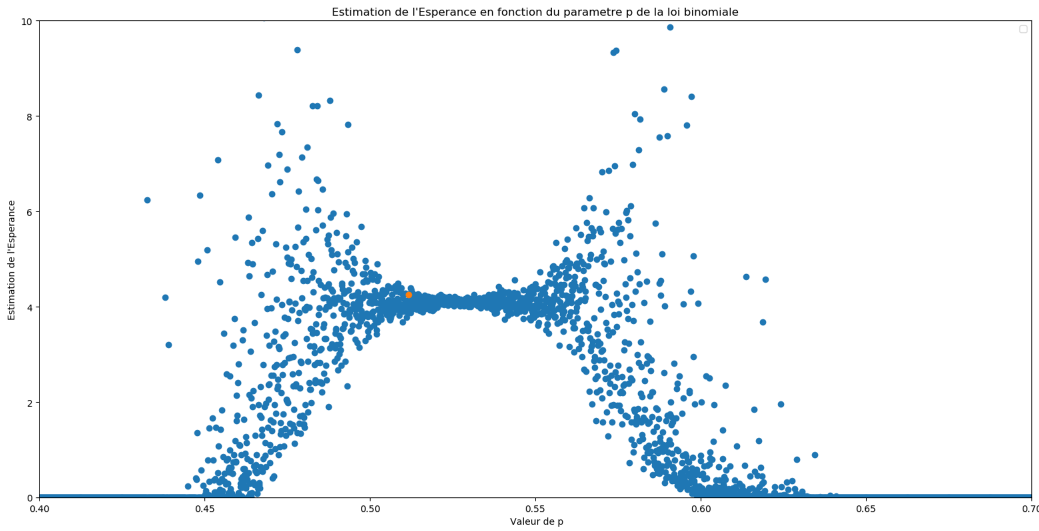

Figure 2.5 – Estimation de l'espérance (valeur du call) en fonction du paramètre p pour n=1000 et N=365

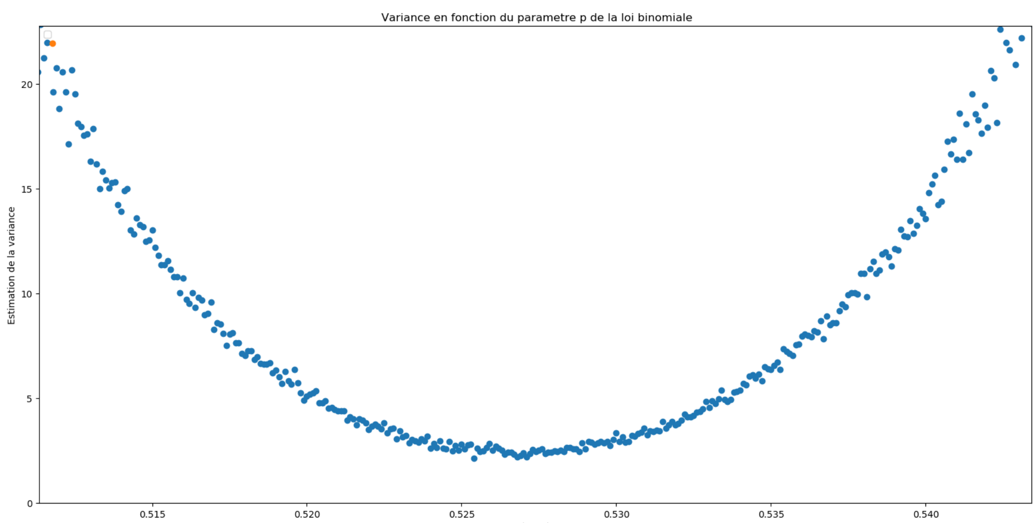

Figure 2.6 – Estimation de la variance en fonction du paramètre p (le point orange correspond à l'estimation de $\mathbb{V}^*[\varepsilon_N(U)^{-1}f(S_N)]$)