



Machine Learning pour la finance

Valorisation d'option US par deep learning et regression

Enseignant : Elie Romuald

Étudiant : Samuel Guilluy - Samer Said - Cédric Nogha

Formation M2 Statistique et Finance



Introduction



Partie 1

Utilisation de la méthode de Longstaff Schwartz pour le pricing des options American

Plan de la partie 1

Afin de pouvoir interpréter nos résultats, il nous est nécessaire de les comparer à d'autres méthodes de pricing d'option Américaine.

Nous allons donc comme dans l'article implémenter la méthode des éléments finies afin de calculer le prix d'un put Americain.

Nous utiliserons trois méthodes de pricing du put américain/bermudéen dans cette première partie :

- La méthode binomiale par Cox, Ross et Rubinstein (CRR).
- La méthodes des éléments finies avec le schéma d'Euler implicite en se basant l'EDP du prix La méthode Longstaff Schwartz présentée dans l'article.

Puis pour aller plus loin dans cette étude nous détaillerons les points suivants :

- Application de la méthode LSM au cas d'une option American-Bermudan-Asiatique
- Application de la méthode LSM au cas d'une option American avec présence de saut

Présentation de la méthode CRR

Il s'agit ici de faire un pricing Backward avec un arbre binomial.

Tout d'abord, on discrétise notre temps avec un pas : $\Delta T = \frac{T}{N}$

A tout pas de temps $n=0, \dots, N-1$, le sous-jacent traité peut ainsi prendre deux valeurs :

$$S_{n+1}^N = \begin{cases} uS_n^N & \text{avec probabilité } p_u \\ dS_n^N & \text{avec probabilité } p_d \end{cases}$$

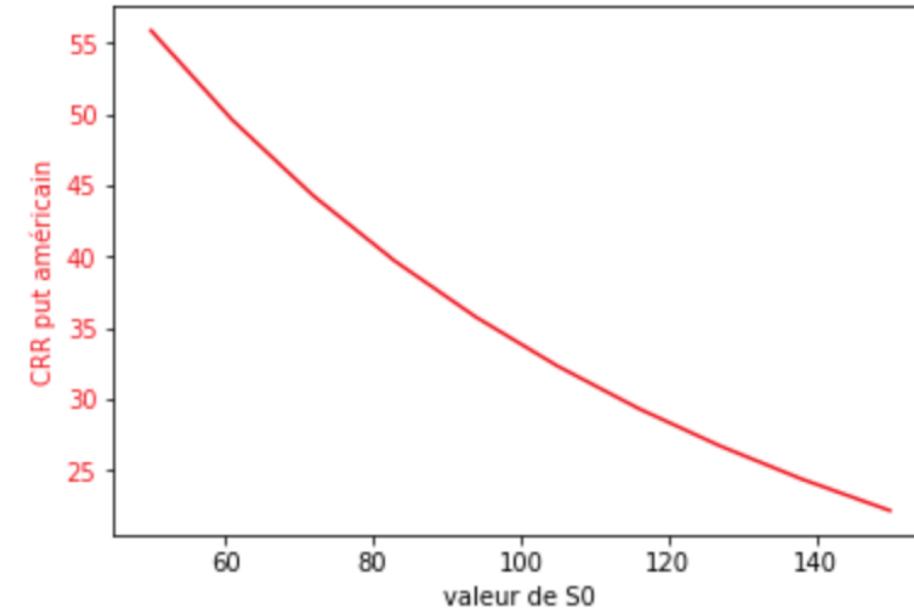
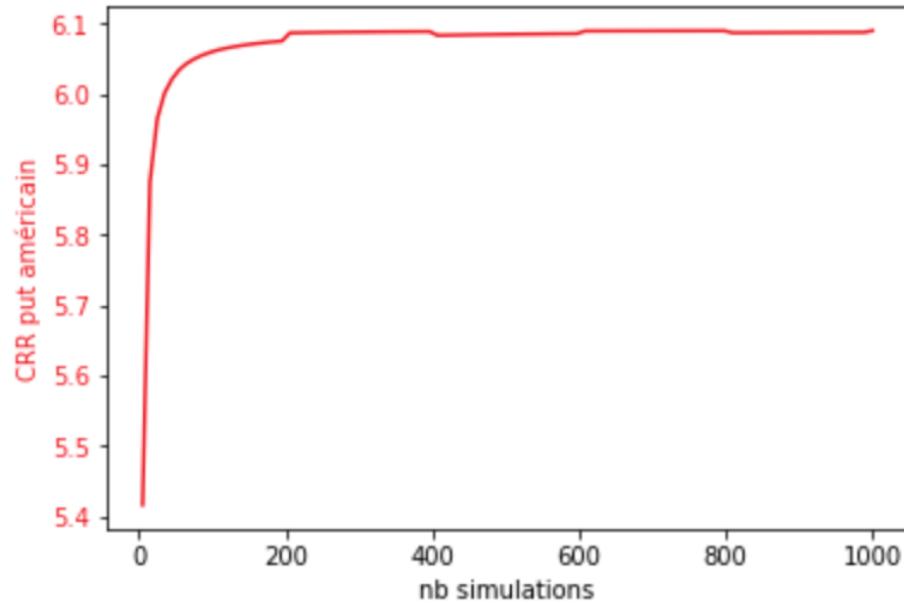
$$u = \frac{1}{d} = e^{\sigma\sqrt{\Delta T}}$$
$$p_u = 1 - p_d = \frac{e^{\sigma\sqrt{\Delta T}} - d}{u - d}$$

Pricing d'un put bermudéen/Américain par le mécanisme backward suivant :

$$V_T(S_0, \dots, S_T) = h(S_T)$$
$$V_t(S_0, \dots, S_t) = \max(h(S_t), e^{-r\Delta T} [p_u V_{t+1}(S_0, \dots, S_t, uS_t) + p_d V_{t+1}(S_0, \dots, S_t, dS_t)])$$

Présentation des résultats

Nous obtenons la convergence suivante pour le put américain ainsi que pour le put bermudéen:



EPD de Black & Scholes Présentation de la méthode

Afin de trouver une formulation de l'EDP de Black Scholes facile à résoudre par les méthodes classiques de différences finies, on repart de l'EDP de Black Scholes et on applique un renversement du temps puis un passage aux variables logarithmique.

L'équation revient alors à :

$$\frac{\partial u}{\partial \tau}(\tau, y) - \frac{\sigma^2}{2} \frac{\partial^2 u}{\partial y^2}(\tau, y) - \left(r - \frac{\sigma^2}{2}\right) \frac{\partial y}{\partial x}(\tau, y) + ru(\tau, y) = 0$$

Avec comme condition initiale : $u(0, y) = (e^y - K)_+$

La formulation en différence finie revient alors à :

$$\frac{du_i}{d\tau}(\tau) - \frac{\sigma^2}{2} \frac{u_{i+1}(\tau) - 2u_i(\tau) + u_{i-1}(\tau)}{h^2} - \left(r - \frac{\sigma^2}{2}\right) \frac{u_{i+1}(\tau) - u_{i-1}(\tau)}{h} + ru_i(\tau) = 0$$

Ce qui donne sous forme matricielle : $\frac{dU}{d\tau}(\tau) = AU(\tau)$, avec A tridiagonale.

Les coefficient de A sont :

- Pour la valeur du terme en « i-1 », la diagonale inférieure $a = \frac{1}{2} \left(\frac{\sigma}{h}\right)^2 - \frac{(r-\sigma^2)}{2h}$
- Pour la valeur du terme en « i », la diagonale inférieure $b = \left(\frac{\sigma}{h}\right)^2 - r$
- Pour la valeur du terme en « i+1 », la diagonale supérieure $c = \frac{1}{2} \left(\frac{\sigma}{h}\right)^2 + \frac{(r-\sigma^2)}{2h}$ où $h = \frac{2L}{N}$

EDP de Black & Scholes : Choix de la Résolution numériques par différences finies

Par la suite, nous allons utiliser 3 schémas de résolutions numériques différents afin de résoudre ce problème de EDP.

En posant : $\Delta T = \frac{T}{M}$ le pas de discrétisation de l'intervalle de temps $[0, T]$ en M partie.

- Schéma de Euler explicite

$$U^{j+1} = (\mathbf{I} + \Delta T \mathbf{A}) U^j$$

- Schéma de Euler Implicite

$$(\mathbf{I} - \Delta T \mathbf{A}) U^{j+1} = U^j$$

- Schéma de Crank Nicholson

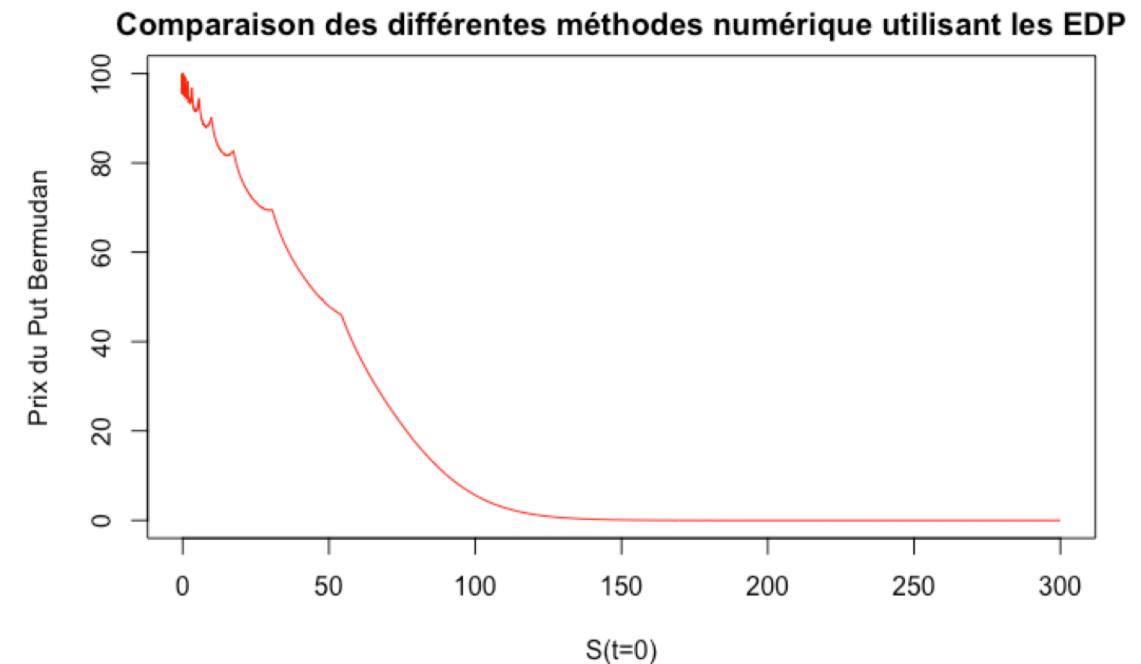
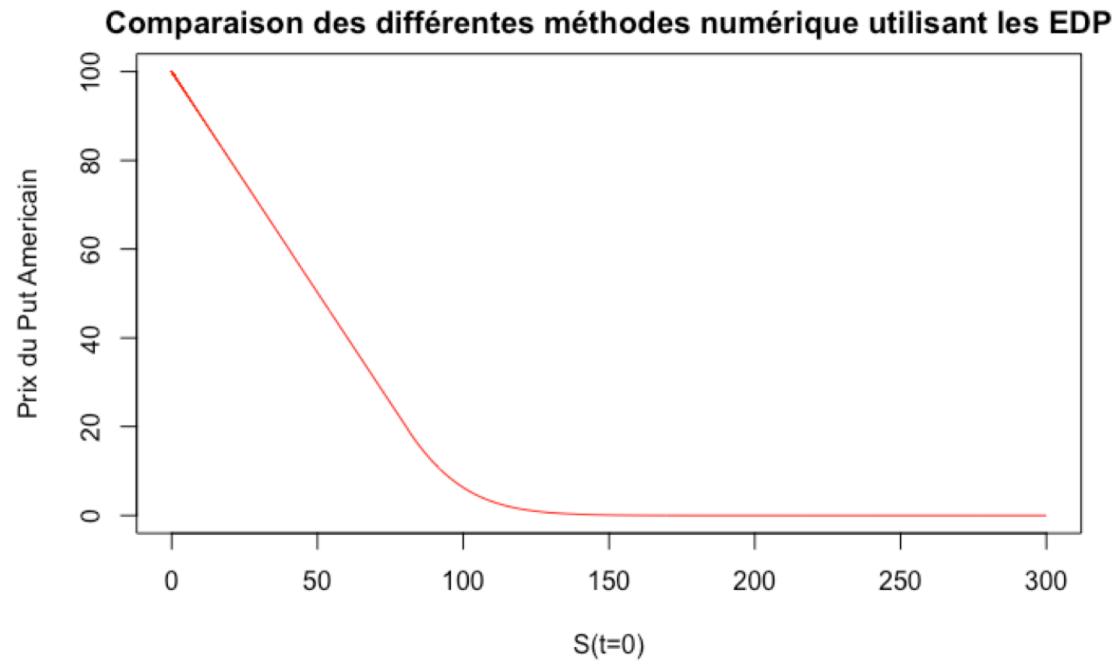
$$(\mathbf{I} - \theta \Delta T \mathbf{A}) U^{j+1} = (\mathbf{I} + (1 - \theta) \Delta T \mathbf{A}) U^j$$

Les Schémas d'Euler Implicite et de Crank Nicholson impose une inversion de matrice afin de calculer le terme $j+1$. En remarquant que la matrice à inverser est tridiagonale à coefficient égaux par diagonale, on peut appliquer une méthode d'inversion de complexité temporelle inférieur aux méthodes classiques telles que Gauss-Jordan.

Afin de passer du pricing d'un call Européen au pricing d'un call Americain il suffit que à chaque pas de temps de calcul de notre option le prix soit $\max(U, K-S)$ où U est le prix du calcul par la méthode des éléments finies et K-S l'argent reçu si on sort décide d'exercé le produit immédiatement

```
for(int i=N-2; i>0; i--){
    Uj[i] = dprime[i] - cprime[i]*Uj[i+1] ;
    // Compare to immediate exercise when allowed
    if(j%nStepsPerDate==0){
        double S = exp(-L+i*h) ;
        if(Uj[i]<K-S){
            Uj[i] = K-S ;
        }
    }
}
```

EPD de Black & Scholes Présentation des résultats



Méthode LSM : Présentation de la méthode sur un exemple

Lorsque le strike est à 1,1

	Stock price paths			
Path	$t = 0$	$t = 1$	$t = 2$	$t = 3$
1	1.00	1.09	1.08	1.34
2	1.00	1.16	1.26	1.54
3	1.00	1.22	1.07	1.03
4	1.00	.93	.97	.92
5	1.00	1.11	1.56	1.52
6	1.00	.76	.77	.90
7	1.00	.92	.84	1.01
8	1.00	.88	1.22	1.34

	Cash-flow matrix at time 3		
Path	$t = 1$	$t = 2$	$t = 3$
1	—	—	.00
2	—	—	.00
3	—	—	.07
4	—	—	.18
5	—	—	.00
6	—	—	.20
7	—	—	.09
8	—	—	.00

	Regression at time 2	
Path	Y	X
1	.00 × .94176	1.08
2	—	—
3	.07 × .94176	1.07
4	.18 × .94176	.97
5	—	—
6	.20 × .94176	.77
7	.09 × .94176	.84
8	—	—

	Optimal early exercise decision at time 2		
Path	Exercise	Continuation	
1	.02	.0369	
2	—	—	
3	.03	.0461	
4	.13	.1176	
5	—	—	
6	.33	.1520	
7	.26	.1565	
8	—	—	

	Option cash flow matrix		
Path	$t = 1$	$t = 2$	$t = 3$
1	.00	.00	.00
2	.00	.00	.00
3	.00	.00	.07
4	.17	.00	.00
5	.00	.00	.00
6	.34	.00	.00
7	.18	.00	.00
8	.22	.00	.00

	Cash-flow matrix at time 2		
Path	$t = 1$	$t = 2$	$t = 3$
1	—	.00	.00
2	—	.00	.00
3	—	.00	.07
4	—	.13	.00
5	—	.00	.00
6	—	.33	.00
7	—	.26	.00
8	—	.00	.00

Méthode LSM : Présentation de l'algorithme

Nous avons pu extraire la méthode de calcul suivante à partir de la lecture de l'article :

Notations : Soit N le nombre de chemin de la simulation, M le nombre de date d'exercice, on les supposera équidistante : $t_j = j\Delta$ avec $\Delta = \frac{T}{M}$
h le payoff de l'option, $g=(g_1, g_2, \dots, g_d)$ le vecteur des fonction de base pour la régression

- 1) Initialisation d'un vecteur de taille N des prix à maturité du sous-jacent: $S_M = (S_T^0, S_T^1, \dots, S_T^{N-1})$
- 2) Initialisation d'un vecteur de taille N du payoff associé à S_M : $P_M = (h(S_T^0), h(S_T^1), \dots, h(S_T^{N-1}))$
- 3) Pour $j = M-1$ à 1 (boucle décroissante)

- Simulation du vecteur des prix : $S_{j\Delta} = (S_{j\Delta}^0, S_{j\Delta}^1, \dots, S_{j\Delta}^{N-1})$ conditionnellement à $S_{(j+1)\Delta}$

Pour calculer ce terme, nous allons utiliser la formule suivante :

$$(B_{t_j} \mid B_0 = 0, B_{t_{j+1}} = x) \sim N\left(\frac{t_j}{t_{j+1}}x, \frac{t_j}{t_{j+1}}(t_j - t_{j+1})I_N\right)$$

- Calcul du coefficient permettant de conserver uniquement les chemins ITM (In the Money)

$$\alpha_j = (\sum_{n \in \{ITM\ paths\}} (g(S^n)^t g(S^n))^{-1} \sum_{n \in \{ITM\ paths\}} P_j^n g(S^n))$$

L'approximation des valeurs continue à l'instant j de l'état x est : $\langle \alpha_j, g(x) \rangle$

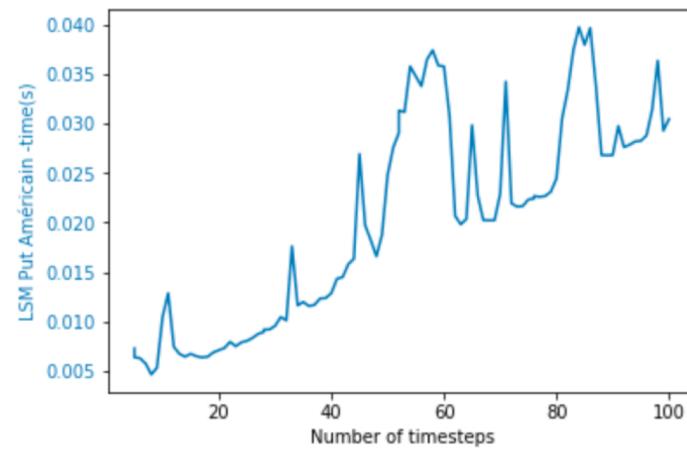
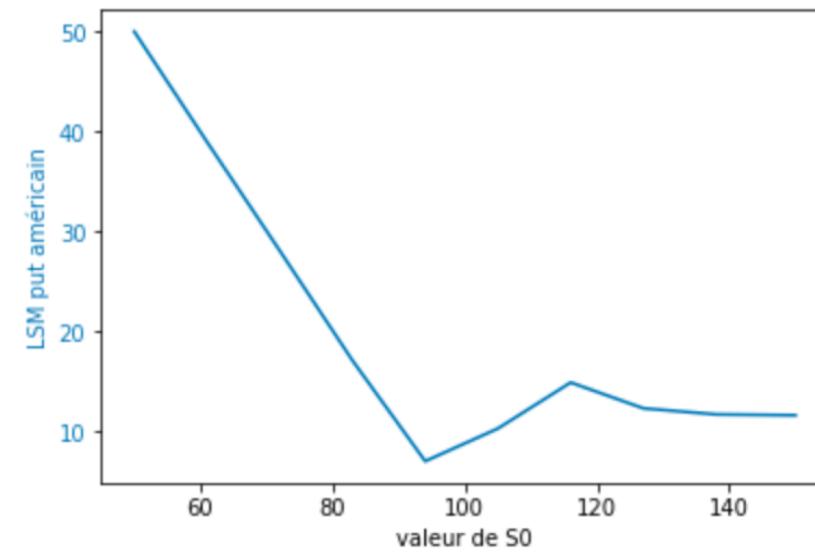
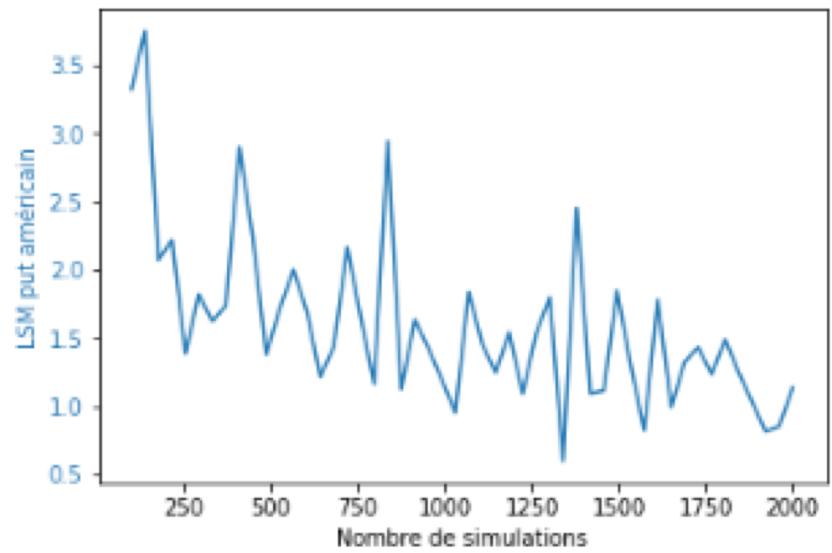
- On met ensuite à jour les prix en comparant la valeur continue et d'exercice des chemins ITM

$$P_j^n = \begin{cases} h(S^n) & \text{if } h(S^n) > \langle \alpha_j, g(S^n) \rangle \\ \text{inchangé sinon} & \end{cases}$$

- Actualisation de tous les prix : $P_{j-1}^n = e^{-\Delta r} P_j^n$ pour $n = 1, \dots, N$

- 4) Le prix final de l'option est : $\max(h(S_0), \frac{1}{N} \sum_{n=1}^N P_0^n)$

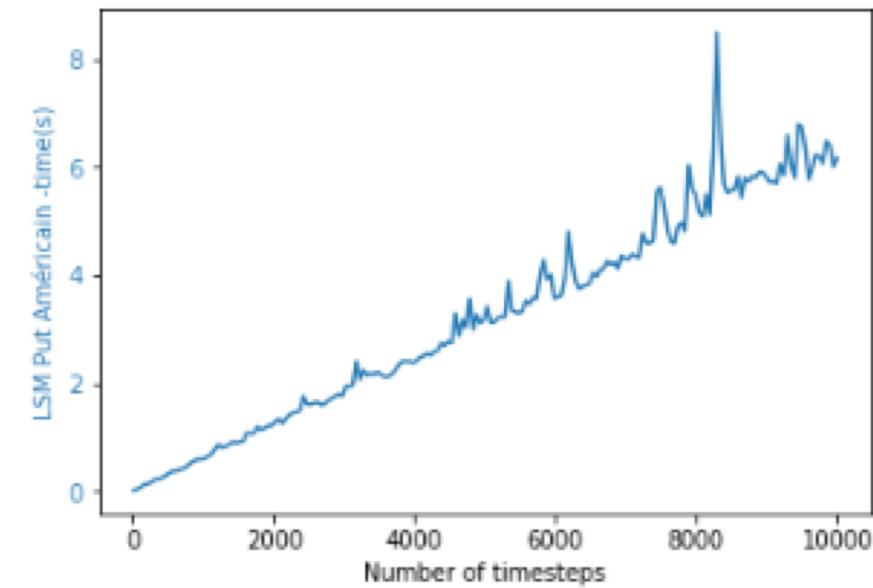
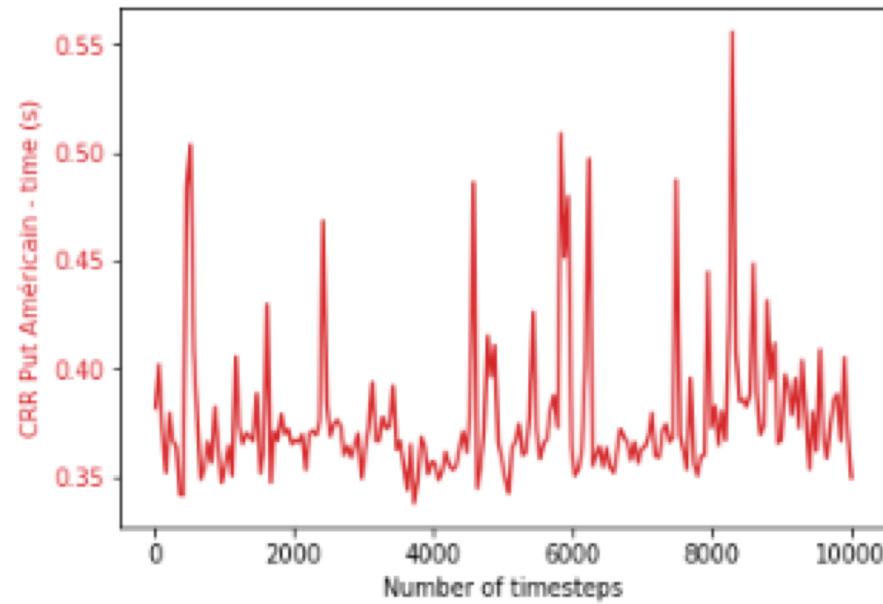
Méthode LSM : Présentation des résultats



LSM : Comparaison des temps d'exécution des méthodes

Ci-dessous, deux graphs afin de comparer les deux méthodes étudiées, à savoir CRR et LSM pour un put américain.

Nous remarquons que la méthode LSM prend nettement plus de temps que la méthode CRR, du aux simulations Monte Carlo ainsi qu'aux régressions linéaires.

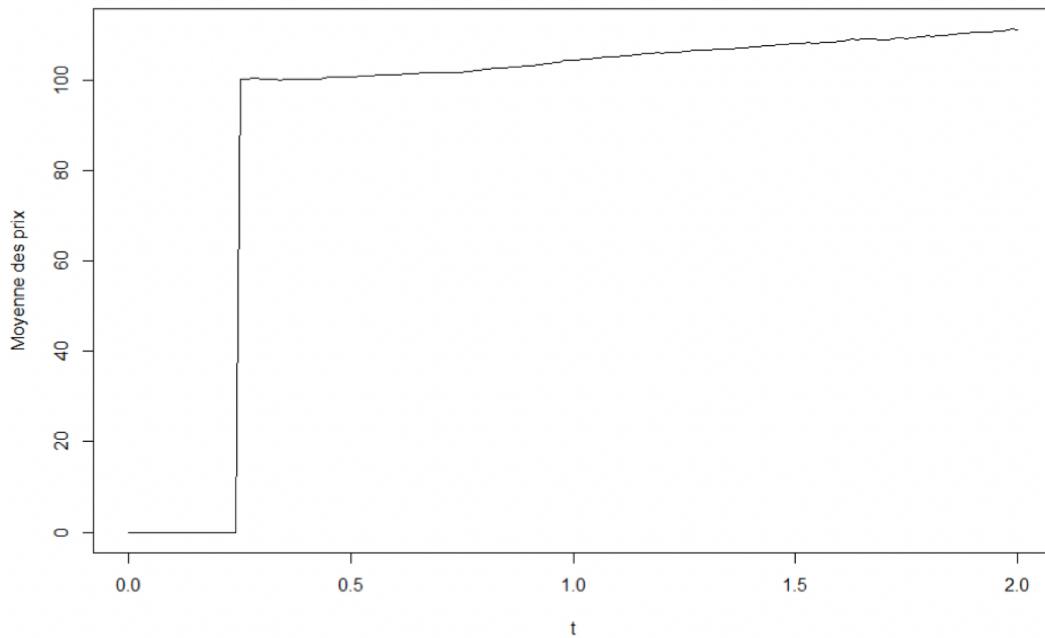


LSM : cas d'une option American-Bermudan-Asiatique

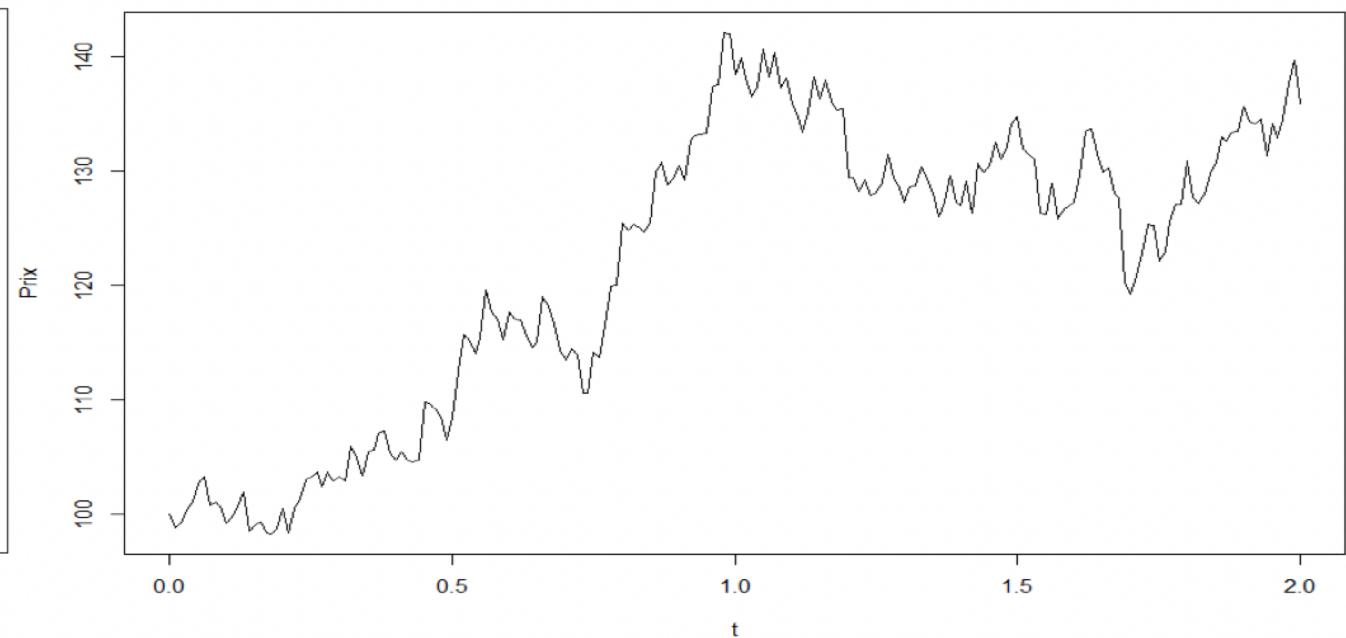
- Dans cette section, on applique la méthode LSM pour une option call exotique path-dependent.
- Une option sur la moyenne des prix sur un horizon prédéfini et ne peut être exercée qu'après une certaine période qui s'appelle « lockout period ».
- On a toujours les mêmes paramètres que la partie précédente, N : nombre de chemins de la simulation, K : nombre de dates d'exercice, etc. Mais cette fois, on ajoute t : La durée de la période après laquelle on a le droit à exercer l'option qu'on va prendre après t=3 mois.
- On a défini premièrement la matrice A, la matrice de moyenne des prix sur un horizon de 3 mois telle que à la date t, $0,25 \leq t \leq T$, A_t reçoit la moyenne de prix sur 3 mois avant la date t.
- On a défini après la matrice $\max(A-K, 0)$, la matrice de cash flow obtenue de la faite d'exercer l'option.

LSM : cas d'une option American-Bermudan-Asiatique

Evolution de la moyenne au cours du temps pour un chemin quelconque



Evolution de prix au cours du temps pour un chemin quelconque



LSM : cas d'une option American-Bermudan-Asiatique

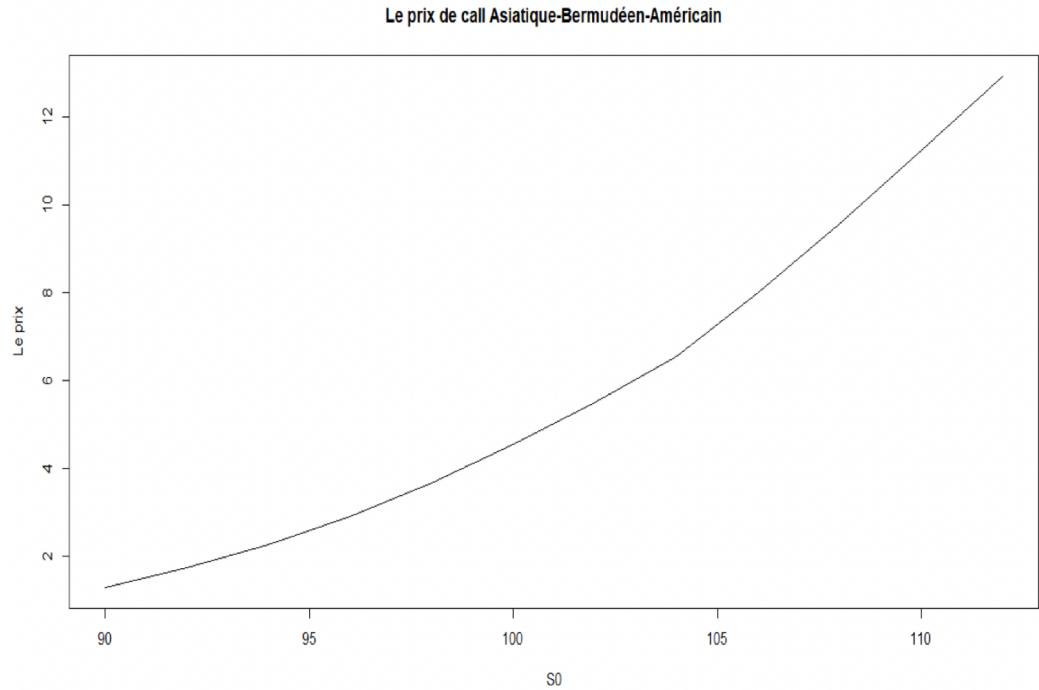
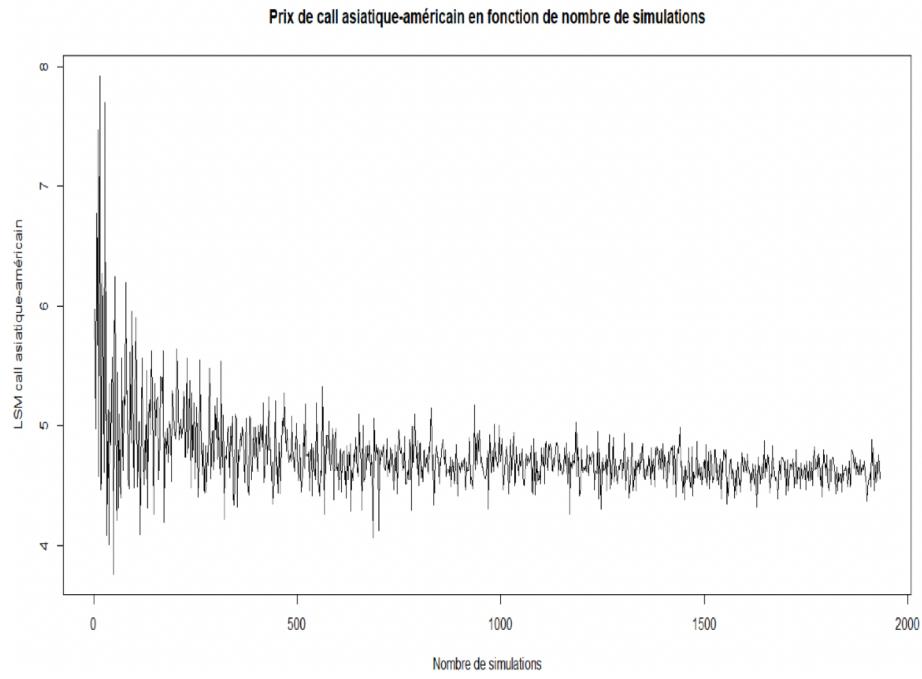
La complexité de l'option comme la dépendance de cash-flow à la trajectoire rend la résolution du problème compliquée comme pour les différences finies, on a transformé le problème à un markovien en prenant comme variables d'état, le prix et la moyenne.

$$(\sigma^2 S^2 / 2) H_{SS} + r S H_S + \frac{1}{25+t} (S - A) H_A - r H + H_t = 0,$$

$$H(S, A, T) = \max(0, A_T - K).$$

Pour la méthode de LSM, on utilise 50000 (25,000 plus 25000 antithétiques) chemins et 100 dates d'exercice par an. Comme fonctions de base pour la régression, on utilise une constante, les deux premiers polynômes de Laguerre évalués au prix de stock et les deux premiers aussi évalués en moyenne des prix et le produit croisé des trois premiers polynômes de Laguerre, finalement on a 8 fonctions de base.

Option American-Bermudan-Asiatique : Présentation des résultats



LSM : cas d'une option American avec présence de saut – Présentation de la méthode

1) Initialisation d'un vecteur de taille N des prix à maturité du sous-jacent: $S_M = (S_T^0, S_T^1, \dots, S_T^{N-1})$

Soit U^i la variable égale à (Poisson($\lambda \times T$) > 0) multiplié par une loi Uniforme(0,T)

Si U^i est non nul 0 alors il y a eu un saut en U^i

donc $S_T^i = 0$

2) Initialisation d'un vecteur de taille N du payoff associé à S_M : $P_M = (h(S_T^0), h(S_T^1), \dots, h(S_T^{N-1}))$

3) Pour j = M-1 à 1 (boucle décroissante)

- Simulation du vecteur des prix : $S_{j\Delta} = (S_{j\Delta}^0, S_{j\Delta}^1, \dots, S_{j\Delta}^{N-1})$ conditionnellement à $S_{(j+1)\Delta}$

Pour calculer ce terme, nous allons utiliser la formule suivante :

$$(B_{t_j} \mid B_0 = 0, B_{t_{j+1}} = x) \sim N\left(\frac{t_j}{t_{j+1}}x, \frac{t_j}{t_{j+1}}(t_j - t_{j+1})I_N\right)$$

Et si U^i est après l'instant $i \times \Delta$ alors $S_{j\Delta}^i$ est nul

- Calcul du coefficient permettant de conserver uniquement les chemins ITM (In the Money)

$$\alpha_j = (\sum_{n \in \{ITM\ paths\}} (g(S^n)^t g(S^n)))^{-1} \sum_{n \in \{ITM\ paths\}} P_j^n g(S^n)$$

L'approximation des valeurs continues à l'instant j de l'état x est : $\langle \alpha_j, g(x) \rangle$

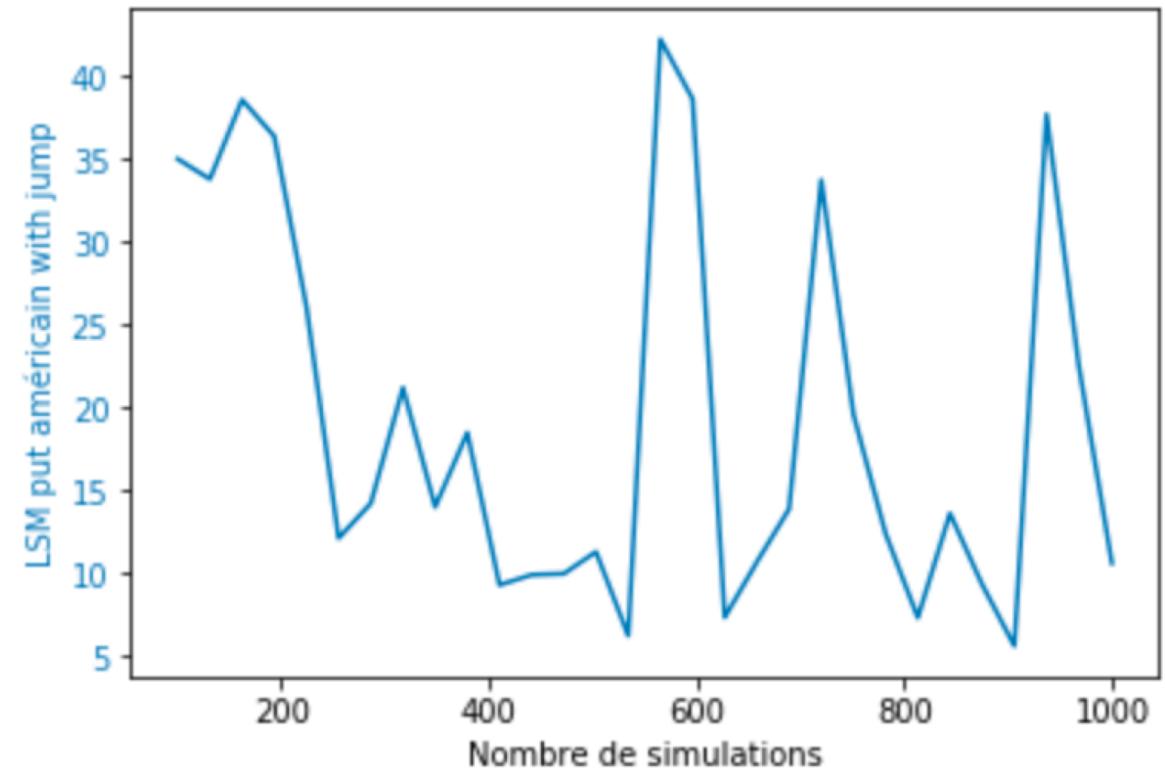
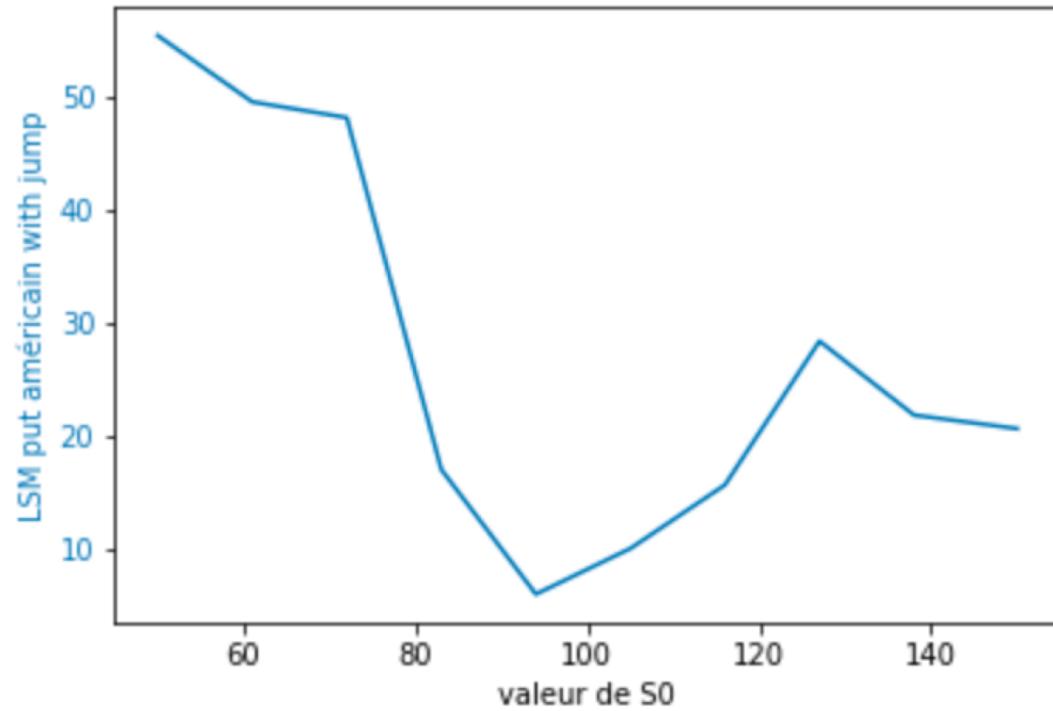
- On met ensuite à jour les prix en comparant la valeur continue et d'exercice des chemins ITM

$$P_j^n = \begin{cases} h(S^n) & \text{if } h(S^n) > \langle \alpha_j, g(S^n) \rangle \\ \text{inchangé sinon} & \end{cases}$$

- Actualisation de tous les prix : $P_{j-1}^n = e^{-\Delta r} P_j^n$ pour $n = 1, \dots, N$

4) Le prix final de l'option est : $\max(h(S_0), \frac{1}{N} \sum_{n=1}^N P_0^n)$

LSM : cas d'une option American avec présence de saut – Présentation des résultats





Partie 2

Pricing d'options américaines à grande dimension

Introduction de la partie 2

Framework

On considère le pricing d'options américaines en temps discret.

Le prix de ce type d'options peut être donnée par la solution du problème d'arrêt optimal :

$$V_0 = \sup_{\tau \in \mathcal{T}(0, \dots, T)} \mathbf{E}[f_\tau(X_\tau)]$$

f_t étant la fonction de payoff discounté et X_0, X_1, \dots, X_T décrivent les processus stochastiques des sous-jacents décrivant le prix des actifs et l'environnement du marché.

Introduction de la partie 2

Framework

Pour fixer les idées, considérons le prix d'une option américaine de strike K sur la moyenne arithmétique de plusieurs sous-jacents corrélés dont les stocks sont modélisés via Black-Scholes par :

$$X_{i,t} = x_{i,0} e^{rt} e^{\sum_{j=1}^m (\sigma_{i,j} W_j(t) - \frac{1}{2} \sigma_{i,j}^2 t)}$$

Où r est le taux sans risque, et $\sigma_i = (\sigma_{i,1}, \sigma_{i,2}, \dots, \sigma_{i,m})^T$ est la volatilité du ième stock, $x_{i,0}$ est la valeur du prix initial du ième stock and $W_j(t) : t \in R_+$ sont des processus de Wiener indépendants.

Introduction de la partie 2

Framework

Pour une fonction de payoff f_t donnée, déterminer son prix revient à trouver un temps d'arrêt optimal $\tau^* \in \mathcal{T}(0, \dots, T)$ satisfaisant

$$\text{Soit } q_t(x) = \sup_{\tau \in \mathcal{T}(t+1, \dots, T)} E[f_\tau(X_\tau | X_t = x)]$$

On peut alors montrer que $\tau^* = \inf \{s \geq 0 : q_s(X_s) \leq f_s(X_s)\}$

Pour déterminer τ^* on se ramène à résoudre le programme suivant :

$$q_T(x) = 0,$$

$$q_t(x) = E \left\{ \max \{f_{t+1}(X_{t+1}, q_{t+1}(X_{t+1}) | X_t = x\} \right\}$$
$$(t = 0, 1, \dots, T - 1)$$

Introduction de la partie 2

Framework

En effet, on a par similitude avec la définition de τ^*

$$q_t(x) = E \{ f_{\tau_t^*}(X_{\tau_t^*}) \}$$

, avec $\tau_t^* = \inf \{ s \geq t+1 \mid q_s(X_s) \leq f_s(X_s) \}$

Puis en utilisant le caractère markovien de $\{X_s\}$, on a :

$$\begin{aligned} q_t(X_t) &= \\ &= E \left\{ f_{t+1}(X_{t+1}) I_{q_{t+1}(X_{t+1}) \leq f_{t+1}(X_{t+1})} + f_{\tau_{t+1}^*}(X_{\tau_{t+1}^*}) I_{q_{t+1}(X_{t+1}) > f_{t+1}(X_{t+1})} \right\} \\ &= E \{ E \{ \dots | X_0, \dots, X_{t+1} \} | X_0, \dots, X_t \} \\ &= E \{ \max \{ f_{t+1}(X_{t+1}), q_{t+1}(X_{t+1}) \} | X_t \} \end{aligned}$$

Méthode numérique d'estimation des q_t

Méthode numérique d'estimation des q_t

Considérons la classe de réseaux de neurones suivante :

$$\mathcal{F}_k(\beta_n) = \left\{ \sum_{i=1}^k c_i \sigma(a_i^T x + b) + c_0 : a_i \in R^d, b_i \in R, \sum_{i=0}^k |c_i| \leq \beta_n \right\}$$

L'algorithme pour déterminer les q_t consiste à simuler des

processus de Markov indépendants

$$\left\{ X_{i,t}^{(l)} \right\}_{t=0,\dots,T} (l = 0, \dots, T-1, i = 1, 2, \dots, n) \text{ identiquement distribués à } \{X_t\}_{t=0,1,\dots,T}.$$

Ensuite on utilise les trajectoires Monte-Carlo ainsi créées pour estimer les q_t avec une régression et un réseau de neurones.

Méthode numérique d'estimation des qt

Méthode numérique d'estimation des qt

On pose ensuite :

$$\hat{q}_n T(x) = 0 \quad (x \in R^d)$$

Pour $t \in \{0, 1, \dots, T\}$ fixé, connaissant une estimation $\hat{q}_{n,t+1}$ de q_{t+1} on estime $q_t(x) = E \{ \max \{ f_{t+1}(X_{t+1}), q_{t+1}(X_{t+1}) \} | X_t = x \}$ •

Analyser le modèle de pricing utilisé et dégager les impacts de chaque paramètre.

En appliquant une régression par un réseau de neurones sur un échantillon $(X_t, \max \{ f_{t+1}(X_{t+1}), q_{t+1}(X_{t+1}) \})$

Méthode numérique d'estimation des qt

Méthode numérique d'estimation des qt

Pour choisir le paramètre k du réseau de neurones on divise notre échantillon simulé en un échantillon de données d'apprentissage et un échantillon de données de test et le choix de k se fait en résolvant :

$$\hat{q}_{n_l, t}(\cdot) = \operatorname{argmin}_{f \in \mathcal{F}_k(\beta_n)} \left(\frac{1}{n_l} \sum_{i=1}^{n_l} |f(X_{i,t}^{(l)}) - \hat{Y}_{i,t}^{(l)}|^2 \right)$$

Méthode numérique d'estimation des qt

Méthode numérique d'estimation du k optimal

Pour le k optimal on le choisit en minimisant le L_2 risque empirique sur les échantillons test et tel que :

$$\hat{k} = \operatorname{argmin}_{k \in \mathcal{P}} \frac{1}{n_t} \sum_{i=n_l+1}^n |\hat{q}_{n_l,t}^k(X_{i,t}^{(t)} - \hat{Y}_{i,t}^{(t)})|^2$$

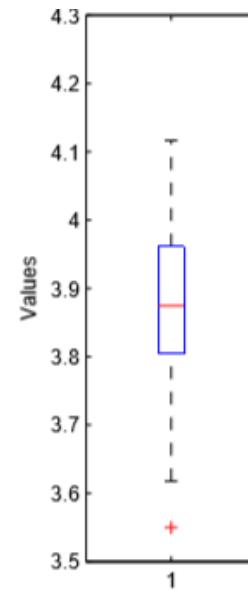
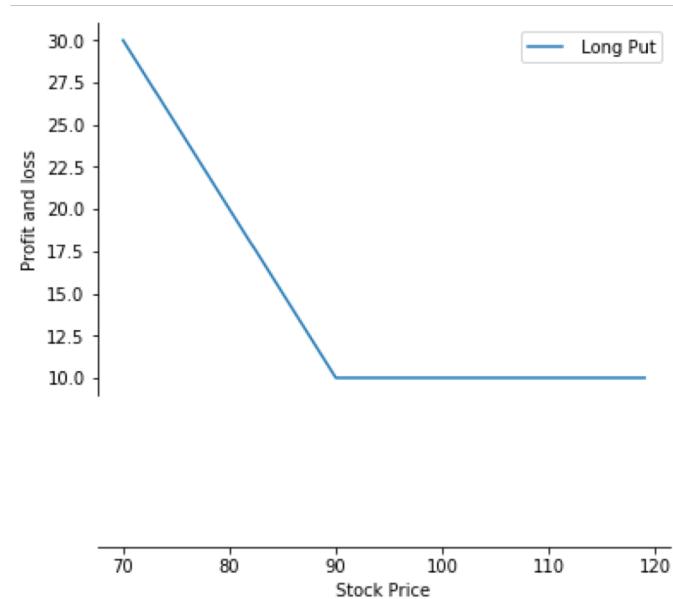
On pose enfin : $\hat{q}_{n,t}(x) = \hat{q}_{n_l,t}^{\hat{k}}(x)$

Présentation des résultats

On considère dans un premier temps comme fonction de payoff bornée un put de strike $K = 90$, de maturité $T=1$ an.

La discréétisation temporelle se fait en posant $t_0 = 0 < t_1 < \dots < t_m = 1$, avec $m=12$. On pose également comme fonction $r = 0.05$, x_0 valeur initiale des stocks à 100

En prenant des échantillons de taille $n=2000$, on obtient le boxplot suivant pour l'estimation du prix avec notre algorithme:

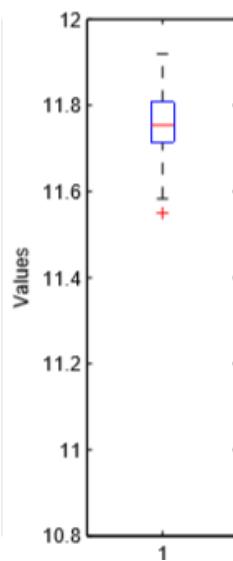


Présentation des résultats

Considérons maintenant un payoff sur 5 actifs sous-jacents dont la matrice de volatilité est donnée par:

$$\sigma = \begin{pmatrix} 0.3024 & 0.1354 & 0.0722 & 0.1367 & 0.1641 \\ 0.1354 & 0.2270 & 0.0613 & 0.1264 & 0.1610 \\ 0.0722 & 0.0613 & 0.0717 & 0.0884 & 0.0699 \\ 0.1367 & 0.1264 & 0.0884 & 0.2937 & 0.1394 \\ 0.1641 & 0.1610 & 0.0699 & 0.1394 & 0.2535 \end{pmatrix}$$

En prenant un nombre de simulation $n = 10\,000$, une discréttisation de pas $m=48$, on obtient:



Conclusion

Le pricing des options américaines revêt un caractère singulier car pour estimer sa valeur il faut utiliser les valeurs des sous-jacents en plusieurs points différents de leur trajectoire.

Dans le cas d'options dont le payoff est simple, et prend en compte un faible nombre de sous-jacents, les méthodes de différence finies et de type CRR nous permettent d'estimer des prix très satisfaisants

mais dans le cas où les sous-jacents sont de type markoviens et les payoffs plus complexes et intégrant plusieurs dimensions d'actifs corrélés, les méthodes de type KKT sont plus adaptées, mais les implémenter de façon efficiente demande beaucoup de temps de calcul à ce moment, et il est très difficile d'estimer les grecs ;