Sujets Python - M2 280

Théo Lopès-Quintas

2021-2022

Consignes générales

Les sujets proposés ici sont à l'image des cours : la difficulté n'est pas dans les notions traitées, mais dans la capacité à produire un code de grande qualité. La notation suivra ce principe : un code qui répond parfaitement au sujet et qui va au delà ne pourra avoir une note élevée si le code n'est pas simple, modulaire et lisible.
On portera un intérêt particulier au noms des différentes variables et des fonctions. Les sujets étant volontairement simple, il est attendu pour avoir une meilleure note d'aller au delà du sujet : faites preuve d'initiative.

Il n'est attendu que le ou les scripts Python de code, mais si vous jugez nécessaire d'accompagner le code avec un fichier texte et/ou des captures d'écran de ce que produit le code. Mais il n'y a rien d'obligatoire.

1 Coder et décoder le code RSA

On souhaite crypter un message avec le code RSA et être capable de le décrypter en connaissant toute les informations, mais également sans connaître la clé secrète.

Prenons l'exemple de Bob qui veut échanger un message avec Alice en utilisant le code RSA. Commençons par la création de la clé publique et secrète.

1. Bob choisit deux nombres premiers p et q et calcule $n = p \times q$.
2. Alice calcule $\phi(n) = (p-1) \times (q-1)$ et choisit e tel que e et $\phi(n)$ soient premier entre eux.
3. Alice calcule d définit par : $d \times e \equiv 1[\phi(n)]$
4. Alice partage (n, e) la clé publique et d est appelé la clé secrète.

Ainsi, Alice donne à tout le monde la clé publique, mais n'échange qu'avec Bob la clé secrète. A partir de maintenant, Bob peut crypter son message et le transmettre à tout le monde. Seulement Alice sera cependant capable de le décrypter. Pour crypter et décrypter un message, on suit la procédure suivante.

1. Bob écrit un message et le transforme en un nombre m.
2. Bob calcule $c \equiv m^e[n]$ et transmet le message crypté c.
3. Alice calcule $c^d[n] = m$ puis traduit le message en texte.

Exemple 1. *On prend $p = 11$ et $q = 59$. On a donc que la clé publique est $(n, e) = (649, 3)$ et la clé secrète est $d = 387$.*
Bob souhaite transmettre le message m =theo. Pour ce faire, il utilise la fonction ord *qui renvoie un code pour chaque caractère. Ici, par exemple* ord$(t) = 116$. *Puis, en utilisant le processus décrit au-dessus, on obtient la valeur 51.*

La grande faiblesse de la méthode RSA est quelle est très *simple* à décoder pour un attaquant, donc quelqu'un qui ne connait pas la clé secrète, mais uniquement la clé publique (n, e) et message crypté c. En effet, si l'on est capable de calculer la décomposition en nombre premier de n, on peut retrouver p et q et donc reconstruire d. La raison qui fait que le code est encore utilisé, c'est qu'en utilisant de très grand nombre premiers, il est très difficile de trouver cette décomposition en nombre premier. Par exemple, avec une implémentation naïve comme nous l'avons fait dans la séance 1 de la décomposition en facteur premier, il faut au moins plus de 20 minutes pour retrouver la décomposition en facteur premier de $n = 991 \times 997 = 988027$: j'ai arrêté le chronomètre à 20 minutes.

Il est donc demander dans ce projet d'être capable de coder et de décoder n'importe quel message texte. On demande aussi d'être capable de décoder un message pour lequel on ne connait que la clé publique.

Il faut donc être capable d'avoir un code efficace, très lisible et modulaire. On ne demande cependant pas le code le plus optimal possible, l'accent doit être mit sur la qualité du code.

2 Le jeu de la vie de Conway

Un automate cellulaire est une grille régulière de cellules qui ont chacune un état (parmi un nombre fini de possibilitées). L'état d'une cellule au temps $t + 1$ dépend de l'état de son voisinage au temps t. A chaque temps on applique les mêmes règles à la totalité de la grille.

Le jeu de la vie de Conway est le plus célèbre des automates cellulaires développé par John Horton Conway en 1970. A chaque itérations, il y a 4 règles qui s'appliquent pour décider de la vie ou de la mort d'une cellule:

- **Naissance** : si une cellule est actuellement morte mais qu'elle a exactement 3 cellules voisines vivantes, alors la cellule devient vivante.
- **Survie** : Toute cellule avec 2 ou 3 voisins vivants reste vivant.
- **Mort par asphyxie** : Toute cellule avec strictement plus de 3 voisins meurt.
- **Mort de solitude** : Toute cellule avec strictement moins de 2 voisins meurt.

Pour initialiser la grille, on lancera une distribution au hasard. Voici un exemple de ce qu'on peut obtenir :

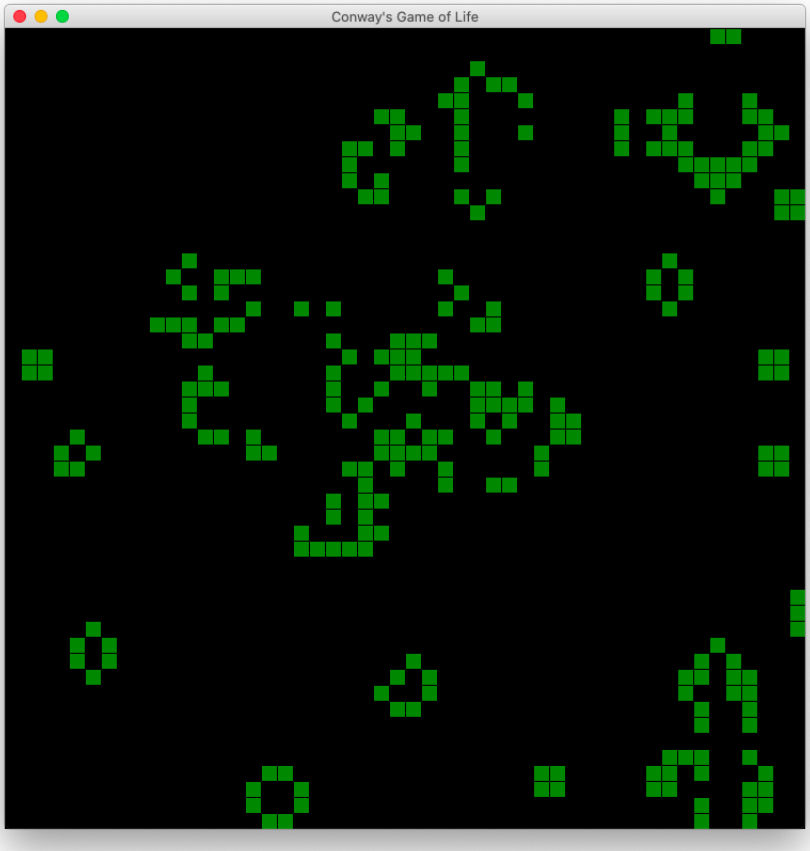

Figure 1: Exemple de ce qu'on peut obtenir à une itération donnée

Le but du projet est de prendre en main la librairie Tkinter et d'implémenter proprement le jeu de la vie de Conway. L'utilisateur doit pouvoir avoir la possibilité de modifier un peu la disposition de la grille (nombre de case en hauteur, largeur, taille de la cellule).

3 Amélioration du visualiseur de stratégie d'option

On veut continuer de travailler sur le visualiseur de stratégie d'options que l'on a travaillé en cours. Pour rappel voilà ce qu'on peut atteindre en utilisant les données de Yahoo Finance en temps réel :

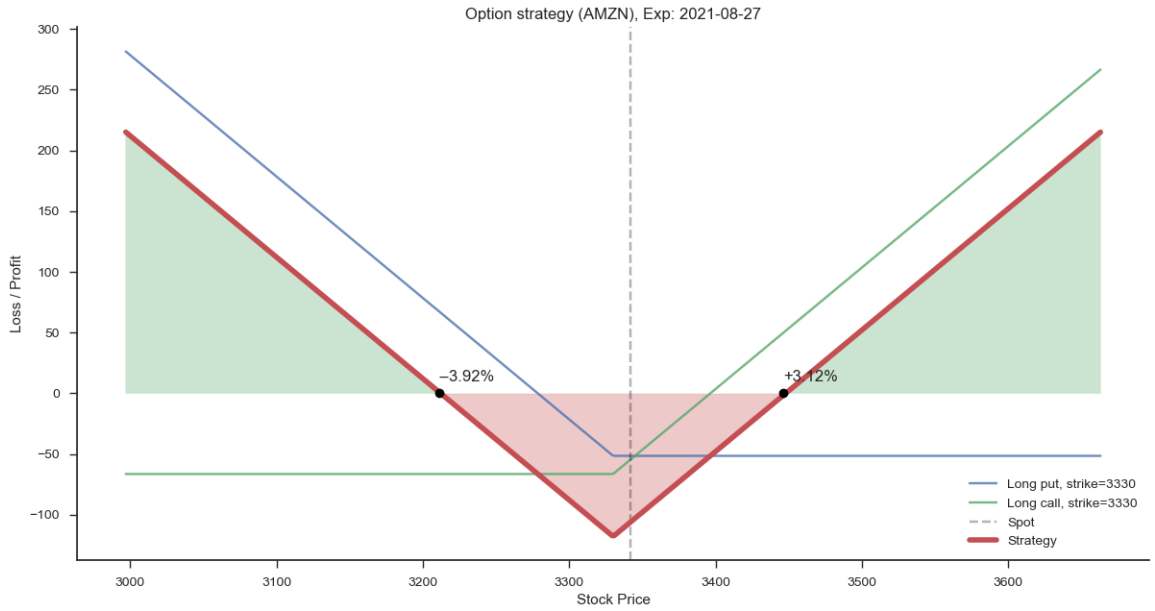

Figure 2: Proposition de ce que l'on a pu obtenir à la suite de la séance 4

Regarder uniquement le gain ou la perte reste faible en terme d'interprétation financière d'une stratégie.

On souhaite donc intégrer la vision des grecs pour juger de la qualité d'une stratégie d'options. Les différents grecs d'une stratégie d'options constituent une mesure du risque du portefeuille d'option qui compose la stratégie.

Ces indicateurs mesurent l'impact sur le prix d'une option de la variation des paramètres qui le forment :

- Le prix du sous-jacent S (le spot)
- Le strike K
- La volatilité implicite σ
- Le temps au bout duquel l'option peut être exécutée. Ainsi, si il reste 3 jours avant l'arrivé à maturité d'une option, $T = 3$
- Le taux d'intérêt r. On peut le prendre faible, pour mes simulations j'ai pris $r = 0.01\%$ par exemple

On rappelle que dans le cadre du modèle de Black-Scholes, le prix d'un call aujourd'hui de maturité dans T jours et de strike K est :

$$C = S\mathcal{N}(d_1) - Ke^{-rT}\mathcal{N}(d_2)$$

Avec :

$$\begin{aligned} d_1 &= \frac{1}{\sigma\sqrt{T}}\left[\ln\left(\frac{S}{K}\right) + \left(r + \frac{1}{2}\sigma^2\right)T\right] \\ d_2 &= d_1 - \sigma\sqrt{T} \\ \mathcal{N}(x) &= \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}\,\mathrm{d}t \end{aligned}$$

On rappelle que par la parité call-put, on a $C - P = S - Ke^{-rT}$. On peut maintenant définir le delta et le gamma :

$$
\begin{aligned}
\delta_{\text{Call}} &= \frac{\partial C}{\partial S} = \mathcal{N}(d_1) \\
\delta_{\text{Put}} &= \frac{\partial P}{\partial S} = \mathcal{N}(d_1) - 1 \\
\gamma_{\text{Call}} &= \gamma_{\text{Put}} = \frac{\partial^2 C}{\partial S^2} = \frac{\mathcal{N}'(d_1)}{S\sigma\sqrt{T}}
\end{aligned}
$$

Un grec du portefeuille est la somme des grecs de chacune des options qui le compose. Voici un exemple de ce qu'on peut souhaiter avoir :

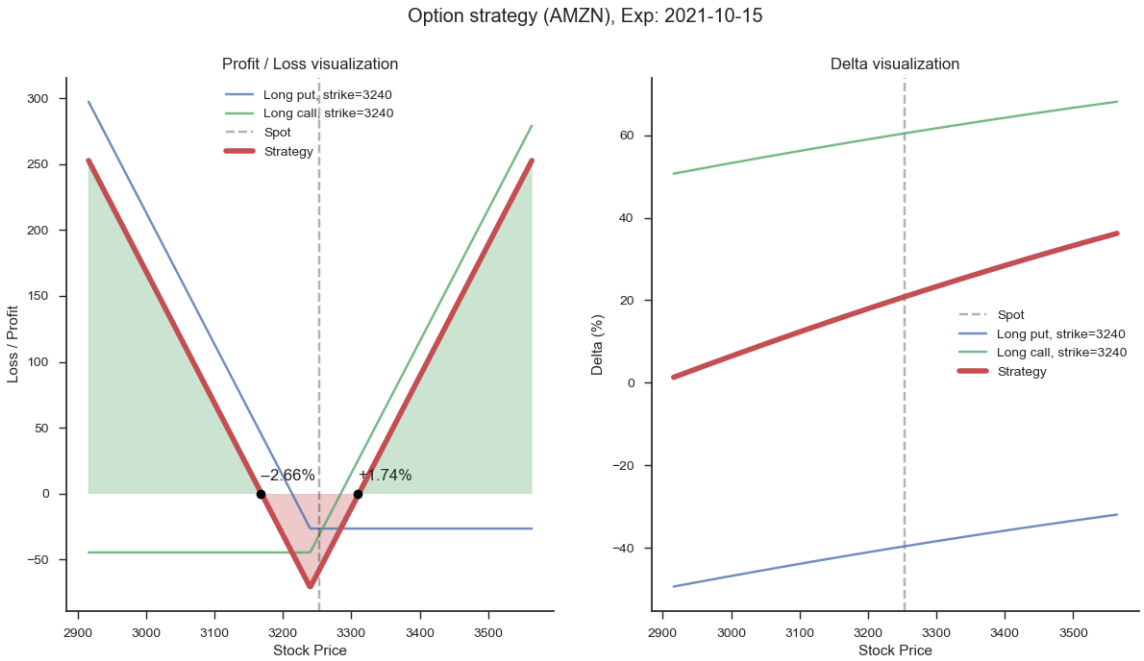

Figure 3: Proposition de ce que l'on peut vouloir pour le delta

On peut imaginer implémenter le calcul du delta et du gamma par exemple. Il est demandé d'utiliser les données de marcher, via yahoo finance par exemple, ou d'une autre manière. Il faut donc récupérer le prix le plus *frais*, un historique sur un an pour calculer la volatilité implicite annualisé et le prix des différentes options considérées par l'utilisateurs.
On rappelle que pour calculer la volatilité annualisé on calcule le log return quotidien puis qu'on calcule la volatilité quotidienne de ce log return. On multipliera le résultat par $\sqrt{252}$ car il y a 252 jours d'exercices.

A Justification de la procédure RSA

Soit p et q premiers, on définit leurs produit $n = p \times q$ et on définit la fonction ϕ qu'on appelle indicatrice d'Euler.

L'indicatrice d'Euler se définit comme suit : pour un nombre donné a, elle renvoie le nombre de nombre premier avec a inférieur ou égal à a. Donc $\phi(8) = 4$ parce que seulement 1, 3, 5 et 7 sont premiers avec 8.
Ainsi, $\phi(p) = p - 1$ pour p premier. Et la fonction a une propriété de multiplicativité pour a et b premiers entre eux : $\phi(ab) = \phi(a)\phi(b)$. Pour le prouver, il faut travailler avec le théorème des restes chinois et l'exploiter pour caractériser la correspondance entre $\{1, 2, \ldots, ab\}$ et le produit cartésien $\{1, 2, \ldots, a\} \times \{1, 2, \ldots, b\}$.

D'où la notation : $\phi(n) = (p-1)(q-1)$. On a choisit d comme l'inverse modulaire de e pour $\phi(n)$, autrement dit :

$$\exists k \in \mathbb{Z}, ed = 1 + k \times \phi(n)$$

On calcule le message m crypté c grâce à :

$$c = m^e[d]$$

Ainsi,

$$\begin{aligned} c^d[n] &= m^{de}[n] \\ &= m^{1+k\phi(n)}[n] \\ &= m \times \left(m^{\phi(n)}\right)^k [n] \end{aligned}$$

Or on sait par le théorème de Fermat généralisé que pour tout entier naturel n strictement positif et tout entier a premier avec n, on a :

$$a^{\phi(n)} \equiv 1[n]$$

D'où le résultat.