

Estimation efficace du risque par simulations séquentielles imbriquées

Par Arthur Guerlais et Jérôme Chiche

Plan:

- I. Introduction**
 - A. Problématique
 - B. Présentation de la méthode
- II. Estimateur à échantillonnage uniforme**
 - A. La méthode
 - B. Optimisation
- III. Estimateur séquentiel**
 - A. La méthode
 - B. L'algorithme associé
- IV. Simulation avec seuil de fiabilité**
 - A. La méthode
 - B. L'algorithme associé
 - C. Comparaison avec l'estimateur séquentiel
 - D. Optimisation
- V. Allocation dynamique**
 - A. Problématique
 - B. L'algorithme associé
- VI. Explication du code et étude d'exemples**
 - A. Le code
 - B. Sorties pour l'exemple Gaussien
 - C. Sorties pour l'exemple du Put sous Black-Scholes
- VII. Conclusion**

I. Introduction

A. Problématique

La mesure et la gestion du risque sont au coeur des préoccupations des institutions financières. Il est nécessaire d'améliorer la modélisation et le calcul de ce risque. Nous nous intéressons ici à une méthode de simulation pour estimer un risque particulier, celui d'une grande perte. Il s'agit de l'estimation efficace du risque par une méthode de simulation séquentielle imbriquée.

B. Présentation de la méthode

On se place à l'instant $t=0$, et on cherche à évaluer la perte en $t=\tau$ d'un portefeuille constitué d'un "échangeable" à échéance $t=T$.

Notons S_t la valeur du sous-jacent sur lequel porte notre "échangeable", et X_t la valeur de notre portefeuille à la date t . S_0 et X_0 sont déterministes, et X_τ est une variable aléatoire que nous cherchons à estimer en simulant des trajectoires de S entre 0 et T .

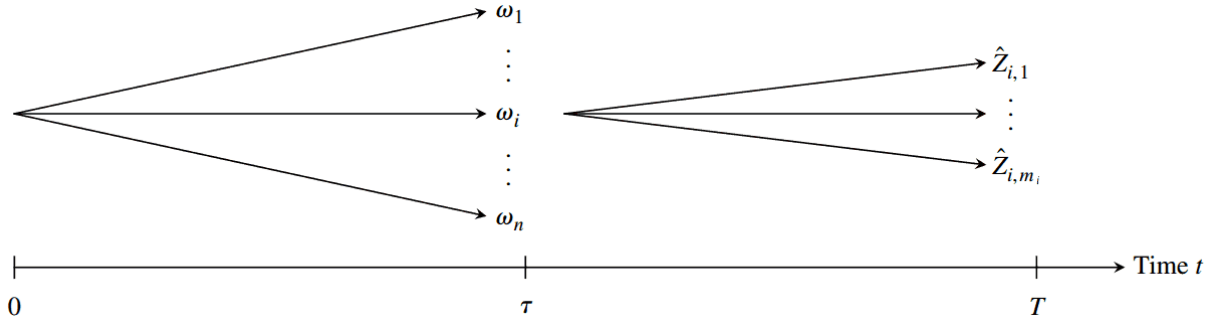
Plus précisément, nous voulons estimer la perte au temps τ donnée par : $L := X_0 - X_\tau$. Nous cherchons la probabilité d'une grande perte i.e. la probabilité que $L(\omega) \geq c$ où c est un seuil réel donné. Notons cette quantité $\alpha := P(L \geq c)$.

Le but de la méthode est d'estimer le plus précisément et efficacement possible cette probabilité α .

La méthode se décompose en deux étapes.

- Étape 1, ou "étape externe": Simulation d'un certain nombre n de scénarios ω_i , $i \in [1, n]$, qui représentent l'évolution du sous-jacent entre les temps 0 et τ .
- Étape 2, ou "étape interne": Pour chaque scénario ω_i , nous simulons m_i trajectoires du sous-jacent $S_T(\omega_i)_j$, $j \leq m_i$ entre les temps τ et T sachant $S_\tau = S_\tau(\omega_i)$. On calcule la perte associée $\hat{Z}_{i,j} = X_0 - \text{Payoff}_{actu}(S_T(\omega_i)_j)$. Puis, par une méthode de Monte-Carlo, on estime la perte associée au scénario externe i par

$$\hat{L}_i = \frac{1}{m_i} \sum_{j=1}^{m_i} \hat{Z}_{i,j}.$$



Le but de la méthode étant d'avoir pour chacun des n scénarios une estimation de la perte à l'instant τ du portefeuille, puis de déterminer le nombre de fois où cette estimation dépasse le seuil c représentant la grande perte. On approxime α par:

$$\hat{\alpha} = \frac{1}{n} \sum_{i=1}^n 1_{L(\omega_i) \geq c}$$

Nous allons étudier quatre méthodes d'échantillonnage et comparer ces méthodes. Nous commencerons par la plus simple: l'échantillonnage uniforme, qui consiste à prendre $m_i = m$, $\forall i \leq n$. Nous nous poserons la question de la répartition optimale entre les m et les n . Les trois autres sont des méthodes non-uniformes (m_i non constant) que nous détaillerons par la suite, avec leurs avantages et leurs inconvénients.

II. Estimateur à échantillonnage uniforme

A. La méthode

Le premier estimateur que nous considérons est l'estimateur à échantillonnage uniforme.

La méthode consiste ici à prendre m_i constant égal à m . On simule donc n scénarios ω_i , pour $1 \leq i \leq n$ pour l'étape 1, dite étape externe. Puis pour chaque scénario ω_i , m trajectoires du sous-jacent sont simulées entre les instants τ et T pour l'étape 2, dite interne.

Nous simulons donc au total:

- n scénarios pour l'étape externe
- $n \times m$ trajectoires pour l'étape interne

La méthode consiste alors, comme indiqué précédemment, à estimer dans chaque scénario la perte à l'aide des m trajectoires simulées.

Nous obtenons, pour chaque scénario ω_i un estimateur de la perte:

$$\hat{L}_i = \frac{1}{m} \times \sum_{j=1}^m \hat{Z}_{i,j}$$

Puis nous estimons la quantité α par $\hat{\alpha}_{n,m} = \frac{1}{n} \sum_{i=1}^n 1_{\hat{L}_i \geq c}$.

Le problème évident qui se pose est alors le suivant: comment choisir n et m pour optimiser cet estimateur ?

B. Optimisation

Optimiser un estimateur revient à minimiser l'erreur quadratique moyenne. Comme précisé ci-dessus, $n \times m$ simulations internes sont réalisées lors de la seconde étape pour obtenir $\hat{\alpha}_{n,m}$. Si l'on impose un coût de calcul de k , notre problème se ramène à chercher n^* et m^* tels que $E[(\hat{\alpha}_{n^*,m^*} - \alpha)^2] = \min_{n,m} E[(\hat{\alpha}_{n,m} - \alpha)^2]$, en respectant la contrainte $n \times m \leq k$.

Dans un premier temps , décomposons l'erreur quadratique moyenne en terme de biais et de variance:

$$E [(\hat{\alpha}_{n,m} - \alpha)^2] = E [(\hat{\alpha}_{n,m} - E [\hat{\alpha}_{n,m}])^2] + (E [\hat{\alpha}_{n,m} - \alpha])^2 = \text{Variance} + (\text{Biais})^2$$

Un théorème et son corollaire nous donnent une caractérisation asymptotique du biais et de la variance. Nous donnons ci-dessous leur résultat:

Théorème 1:

On suppose connue la densité jointe de (L, \hat{L}) : $p_m(l, \hat{l})$, et que ses dérivés partielles $\frac{\partial p_m(l, \hat{l})}{\partial l}$ et $\frac{\partial^2 p_m(l, \hat{l})}{\partial l^2}$ existent pour tout l et \hat{l} .

On conjecture en outre que: $\forall m \geq 1, \exists f_{0,m}(\cdot), f_{1,m}(\cdot), f_{2,m}(\cdot)$ telles que:

$$p_m(l, \hat{l}) \leq f_{0,m}(\cdot), \left| \frac{\partial p_m(l, \hat{l})}{\partial l} \right| \leq f_{1,m}(\cdot), \left| \frac{\partial^2 p_m(l, \hat{l})}{\partial l^2} \right| \leq f_{2,m}(\cdot)$$

Alors,

$$E [\hat{\alpha}_{n,m} - \alpha] = \frac{\theta_c}{m} + O(m^{-3/2})$$

où:

- $\theta_c = -Y'(c)$
- $Y(c) = \frac{1}{2} \times f(c) \times E [\sigma^2(\omega) | L(\omega) = c]$
- $\sigma^2(\omega)$ représente la variance des échantillons internes dans le scénario ω_i .

Corollaire 1 :

Sous les mêmes conditions , lorsque $m \rightarrow +\infty$

$$\text{Var} (\hat{\alpha}_{n,m}) = \frac{\alpha(1-\alpha)}{n} + O(m^{-1}n^{-1})$$

La variance asymptotique est déterminée par n et décroît en n^{-1} , tandis que le biais asymptotique est déterminé par m et décroît en m^{-1} .

Un choix naïf pour n et m serait de choisir $n = m = k^{1/2}$.

Le carré du biais asymptotique serait alors d'ordre k^{-1} et la variance asymptotique serait d'ordre $k^{-1/2}$. Ce choix naïf n'est clairement pas optimal car la variance domine asymptotiquement le carré du biais et détermine l'erreur quadratique moyenne.

Un meilleur choix serait d'augmenter n et de réduire m .

En utilisant le théorème et son corollaire précédemment cités, on peut se ramener au problème d'optimisation suivant:

trouver n et m pour lesquels on obtient $\min_{0 \leq n, m; n*m \leq k} \frac{\alpha(1-\alpha)}{n} + \frac{\theta_c^2}{m^2}$.

Les quantités n et m répondant au problème sont alors :

- $m^* = \frac{k^{1/3}}{\beta^*}$
- $n^* = \beta^* k^{2/3}$

avec $\beta^* := \left(\frac{\alpha(1-\alpha)}{2\theta_c^2}\right)^2$

Et l'erreur quadratique moyenne asymptotique optimale est :

$$E [(\hat{\alpha}_{n,m} - \alpha)^2] = 3(\beta^*)^2 k^{-2/3} + o(k^{-2/3})$$

La limite de tout ce raisonnement est qu'il est difficile d'estimer θ_c que nous avons utilisé pour exprimer m^* et n^* .

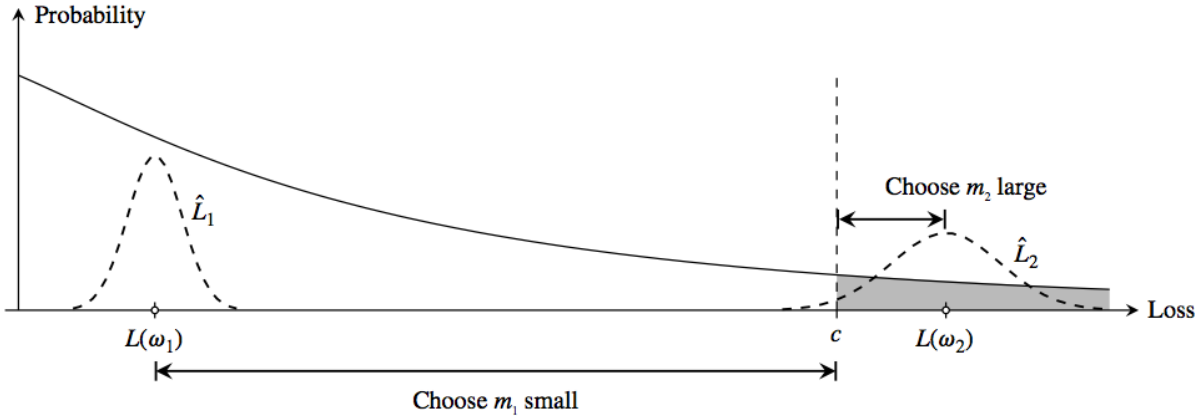
Remarquons de plus que si la perte $L(\omega_i)$ peut être exactement calculée à partir de la simulation ω_i , en faisant un Monte Carlo simple sur les scénarios externes, avec n simulations, on aurait un estimateur qui décroît asymptotiquement en n^{-1} , alors qu'avec notre méthode uniforme optimale, pour k scénarios internes simulés, on a un ordre de seulement $k^{-2/3}$.

III. Estimateur séquentiel

A. La méthode

L'estimateur défini au chapitre II. utilise un nombre constant de simulations internes pour chaque scénario externe. Or la formule pour estimer α ne prend en compte que les $1_{\hat{L}_i \geq c}$. Il paraît alors assez clair que cette stratégie n'est pas la plus efficace, comme le montre le schéma ci-dessous. On y voit deux scénarios possibles:

- Le premier montre un scénario où la perte $L(\omega_1)$ en τ est loin de c . On peut alors dire de manière assez fiable que la perte passera le seuil avec une probabilité très proche de 0 même en ajoutant d'autres simulations internes. Dans ce cas, \hat{L}_1 sera plus petit que c , et cela pourra être estimé sans utiliser un grand nombre d'étapes internes.
- Dans le second cas par contre, $L(\omega_2)$ est proche de c . On aurait donc envie de simuler plus d'étapes internes, et ce afin d'avoir une plus grande précision et savoir si oui ou non ω_2 est un scénario avec une perte qui dépasse c .



En fait, cela se voit dans la manière dont on estime α : $\hat{\alpha} = \frac{1}{n} \sum_{i=1}^n 1_{\hat{L}_i \geq c}$. On ne s'intéresse qu'à savoir si \hat{L}_i est supérieur à c ou non. On va donc faire pour chaque scénario ω_i une première estimation \hat{L}_i avec un nombre réduit m_i^0 de simulations internes $\hat{Z}_{i,j}, j \leq m_i^0$, et ensuite décider des scénarios pour lesquels on observe des pertes proche de c pour y ajouter des simulations et savoir plus précisément si la perte est plus importante que c ou non.

Cette simulation supplémentaire n'impactera $\hat{\alpha}$ que si le signe de $\hat{L}_i - c$ change après

avoir rajouté une simulation interne au scénario ω_i . Nous cherchons donc le scénario qui maximise la probabilité de ce changement de signe. Des calculs utilisant la formule de Chebychev montrent qu'il faut ajouter une simulation à un scénario qui vérifie:

$$i^* \in \arg \min_i \frac{m_i |\hat{L}_i - c|}{\sigma_i}$$

On remarque ici l'apparition de σ_i qui est la variance des $Z_{i,\cdot}$.

Ceci est cohérent avec l'intuition qu'il faut ajouter des simulations aux scénarios qui sont soit proches du seuil, i.e. $|\hat{L}_i - c|$ petit, soit à ceux auxquels on a pour l'instant attribué peu de réalisations, i.e. m_i petit. Aussi, si la variance σ_i de ce scénario est grande, il faudra plus simuler pour être plus précis, car nos simulations seront plus dilatées.

Par la suite, nous désignerons par "fiabilité" du scénario ω_i la quantité $\frac{m_i |\hat{L}_i - c|}{\sigma_i}$.

B. L'algorithme associé

Maintenant que nous avons expliqué la méthode proposée, nous allons expliquer comment nous avons implémenté cet algorithme, ainsi que ses inconvénients et ses avantages en termes de calcul.

Dans la pratique, pour utiliser cette méthode, il faut donc fixer au préalable trois valeurs: le nombre de scénarios n que l'on va utiliser, le nombre minimal m_0 de simulation interne par scénario, et le nombre moyen \bar{m} de simulations souhaitées par scénario (ce qui revient à choisir un nombre total de simulations internes à réaliser).

L'algorithme procède comme suit : nous simulons d'abord n scénarios externes, avec dans chacun m_0 réalisations internes. Nous trions ensuite le vecteur des scénarios ainsi obtenu en fonction de leur fiabilité respective définie ci-dessus. Ensuite, nous ajoutons une simulation au premier scénario du vecteur, et le remettons à sa place par rapport à sa nouvelle fiabilité. Nous répétons cette étape jusqu'à ce qu'on atteigne le nombre de simulations total voulu.

Comme dit précédemment, cela requiert de connaître les variances de chaque scénario qui peuvent être inconnues et difficilement estimables. De plus il y a la fiabilité qu'il faut calculer après chaque ajout de simulation. Cela représente un coût de calcul non négligeable, et nécessite de stocker les scénarios.

Les deux méthodes décrites par la suite héritent de celle-ci.

IV. Simulation avec seuil de fiabilité

A. La méthode

L'estimateur précédent nous permet de fixer le nombre de simulation à réaliser, et donc dans une certaine mesure le temps de calcul par un ordinateur, mais ne permet pas de nous assurer de la fiabilité de chaque scénario.

Nous allons donc présenter une méthode très similaire à la précédente, plus simple à analyser, qui permet de fixer un seuil de fiabilité pour chaque scénario, et sans utiliser d'espace mémoire, mais avec un coût important que nous expliquerons.

Comme précédemment, nous fixons au préalable le nombre de scénarios n sur lesquels nous allons travailler. Mais nous ne fixerons pas de minimum ou de maximum pour les nombres de simulations internes. Ici, pour chaque scénario externe, nous allons continuer à simuler des réalisations internes $\hat{Z}_{i,j}$ dans le scénario ω_i tant que la fiabilité de ce scénario n'aura pas dépassé un seuil γ qui sera, lui, fixé en même temps que n .

B. L'algorithme associé

Nous allons donc simuler un scénario externe, puis simuler des scénarios internes tant que la fiabilité n'aura pas dépassé le seuil voulu. Nous regardons ensuite la perte estimée pour ce scénario externe. Nous réitérons ce procédé autant de fois que le nombre de scénarios souhaités. Enfin, nous moyennons les estimations de pertes obtenues.

Cela évite de stocker tous les scénarios externes, en sommant les pertes au fur et à mesure que nous les obtenons, et en divisant à la fin par le nombre de scénarios. En revanche, étant donné que nous simulons tant que le seuil γ n'est pas atteint, il n'est pas possible de prévoir exactement le nombre d'étapes internes à réaliser, et donc le temps de calcul de l'algorithme.

C. Comparaison avec l'estimateur séquentiel

Ces deux estimateurs sont assez similaires dans l'idée : ils cherchent à construire un échantillonnage interne en maximisant la fiabilité de chaque scénario interne. En revanche, si le premier algorithme faisait intervenir le nombre moyen d'échantillons internes souhaités, le second algorithme se base lui sur le seuil γ . La paramétrisation est donc différente.

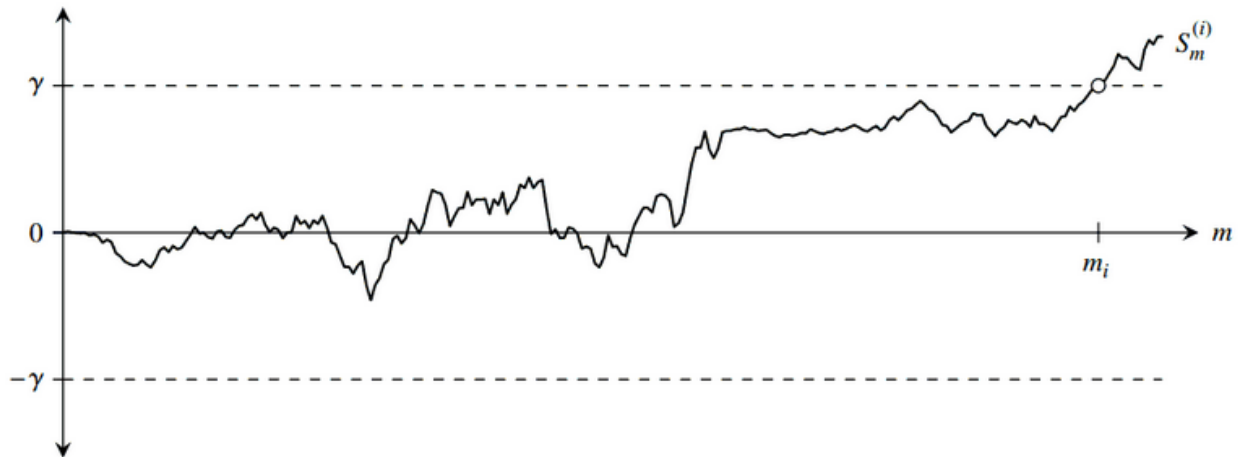
Comme dit précédemment, l'algorithme séquentiel permet de fixer le nombre de simulations souhaitées et paraît ainsi plus naturel, en fixant un temps de calcul. En revanche, le seuil ne permet pas de contrôler le coût de calcul. En effet, bien que le seuil définisse implicitement le coût, il n'est pas possible, à partir d'un coût imposé, de choisir le seuil.

Par contre, d'un point de vue théorique, l'estimateur à seuil de fiabilité est plus intéressant, car plus simple à analyser. Cela est dû au fait que les estimations des pertes $\hat{L}_1, \dots, \hat{L}_n$ sont indépendantes. On ne choisit pas d'ajouter une simulation interne à un scénario en fonction d'un paramètre des autres scénarios, mais seulement en fonction de la fiabilité de celui-ci. Cela permet d'analyser l'algorithme avec un seul scénario. Ce qui n'était pas le cas pour l'estimateur séquentiel qui comportait des dépendances entre les estimations de perte, et donc entre les scénarios.

Nous allons donc étudier un scénario ω_i avec un seuil γ donné. Cette étude se généralisera ensuite à tous les scénarios par indépendance (et distribution identique). On cherche le nombre de simulations m_i tel que:

$$m_i = \inf\{m > 0 : |S_m^{(i)}| \geq \gamma\} \text{ avec } S_m^{(i)} = \sum_{j=1}^m \frac{1}{\sigma_i} (\hat{Z}_{i,j} - c)$$

En effet, ce sera le nombre de simulations à effectuer pour dépasser le seuil. On remarque que $\{S_m^{(i)}, m \geq 0\}$ est une marche aléatoire et que m_i est le premier instant de sortie de l'intervalle $[-\gamma, \gamma]$ comme le montre la figure ci-dessous.



Il paraît alors logique de s'intéresser à l'espérance \bar{m} de ce m_i qui pourra servir à approximer dans une certaine mesure le \bar{m} rencontré dans l'algorithme séquentiel.

D. Optimisation

Commençons par dire quelques mots sur le temps d'exécution de l'algorithme.

Celui-ci est proportionnel au nombre moyen de simulations internes, qui, comme nous l'avons vu ci-dessus est l'espérance d'une aléatoire. On note alors:

$$\bar{m}(\gamma) = E[\inf\{m > 0 : \frac{m}{\sigma(\omega)} |\hat{L}(\omega) - c| \geq \gamma\}].$$

Théorème 2:

On suppose que pour un scénario ω_i , les échantillons internes $\hat{Z}_{i,1}, \hat{Z}_{i,2}, \dots$ sont des variables aléatoires i.i.d., et en posant: $\mu(\omega_i) := \frac{L(\omega_i) - c}{\sigma(\omega_i)}$, que la densité de μ existe et est continûment dérivable au voisinage de 0.

Alors, lorsque $\gamma \rightarrow \infty$, le nombre d'échantillons internes espéré pour chaque scénario satisfait : $\bar{m}(\gamma) = O(\gamma \log(\gamma))$.

Le théorème nous donne une proportionnalité en $n \times \log(\gamma) \times \gamma$ du temps d'exécution de l'algorithme.

Notons, à présent, $\tilde{\alpha}_{\gamma,n}$ l'estimateur à seuil de fiabilité.

Le but de ce paragraphe est d'optimiser $\tilde{\alpha}_{\gamma,n}$ à l'aide du choix de n et de γ . Nous allons, pour cela, décomposer l'erreur quadratique moyenne en un terme de variance et un terme de biais, comme nous l'avons fait pour l'estimateur uniforme.

$$E[(\tilde{\alpha}_{\gamma,n} - \alpha)^2] = E[(\tilde{\alpha}_{\gamma,n} - E[\tilde{\alpha}_{\gamma,n}])^2] + (E[\tilde{\alpha}_{\gamma,n} - \alpha])^2$$

Sous les mêmes hypothèses que celles du théorème 2, le théorème suivant et son corollaire nous donnent une caractérisation asymptotique du biais et de la variance.

Théorème 3:

Le biais asymptotique de l'estimateur à seuil de fiabilité satisfait :

$$E[\tilde{\alpha}_{\gamma,n} - \alpha] = O(\gamma^{-2})$$

Corollaire 2:

La variance asymptotique de l'estimateur à seuil de fiabilité satisfait :

$$Var(\tilde{\alpha}_{\gamma,n}) = \frac{\alpha(1-\alpha)}{n} + O(\gamma^{-2}n^{-1})$$

D'après le théorème 2, le nombre moyen d'échantillons internes utilisés dans chaque scénario vérifie: $\bar{m}(\gamma) = O(\gamma^{1+\varepsilon})$, $\forall \varepsilon > 0$. Nous pouvons alors exprimer le biais et la variance

ci-dessus comme des fonctions de \bar{m} :

- $E[\tilde{\alpha}_{\gamma, n} - \alpha] = O(\bar{m}^{-2+\varepsilon})$
- $Var(\tilde{\alpha}_{\gamma, n}) = \frac{\alpha(1-\alpha)}{n} + O(\bar{m}^{-2+\varepsilon} n^{-1})$

La variance asymptotique est donc d'ordre n^{-1} , comme l'était la variance asymptotique de l'estimateur uniforme. Ce qui est cohérent avec ce nous avons dit précédemment, la variance est déterminée par l'étape externe. En revanche, si nous comparons le biais asymptotique de l'estimateur à seuil et celui de l'estimateur uniforme, tous deux fonctions du nombre moyen de scénarios internes simulés, on constate que le premier décroît aussi vite que le carré du second. Ce qui donne rend l'estimateur de seuil plus précis.

Nous majorons l'erreur quadratique comme suit :

$$E[(\hat{\alpha}_{\gamma, n} - \alpha)^2] \leq \frac{\alpha(1-\alpha)}{n} + \frac{C}{\gamma^4}$$

Nous pouvons réduire l'erreur quadratique moyenne en minimisant le terme de droite en fonction des paramètres n et γ , avec un choix approprié pour la constante C .

Nous cherchons alors : $\min_{\gamma, n} \frac{\alpha(1-\alpha)}{n} + \frac{C}{\gamma^4}$, en respectant toujours une contrainte de coût:

$$\bar{m}(\gamma)n \leq k$$

Supposons que l'on choisisse, pour un certain ε , γ^* proportionnel à $k^{1/5}$ et n^* proportionnel à $k^{4/5-\varepsilon}$.

Alors : $\bar{m}(\gamma^*)n^* = O(k^{1-\varepsilon} \log(k)) = o(k)$. La contrainte est donc respectée pour un k suffisamment grand.

L'erreur quadratique moyenne est alors: $E[(\hat{\alpha}_{\gamma, n} - \alpha)^2] = O(k^{-4/5+\varepsilon})$

Ainsi, l'estimateur à seuil optimal converge plus rapidement que tout estimateur uniforme. Ce résultat est dû à une plus grande simulation de scénarios externes ($k^{-4/5+\varepsilon}$ contre $k^{-2/3}$ pour l'uniforme) et à une plus petite simulation de scénarios internes ($k^{1/5}$ en moyenne par scénario contre $k^{1/3}$ pour l'uniforme). En concentrant nos simulations internes autour des scénarios dont la fiabilité était initialement plus faible, on construit une méthode plus efficace que la méthode par échantillonnage uniforme.

V. Allocation dynamique

A. Problématique

Nous avons vu précédemment, avec l'estimateur séquentiel, comment choisir et répartir le nombre de simulations internes à travers chaque scénario. Néanmoins, une question reste en suspend: comment savoir s'il est plus rentable de simuler un nouveau scénario ou ajouter une simulation dans un scénario existant et augmenter sa fiabilité ?

Notre analyse de l'estimateur à seuil de fiabilité nous a montré que le choix optimal pour le nombre de scénarios n , étant donné un budget k de simulations interne, était proportionnel à $k^{4/5-\varepsilon}$, et donc une moyenne \bar{m} de simulations internes proportionnel à $k^{1/5}$. Il reste un paramètre que nous n'avons pas abordé jusqu'ici: comment choisir les constantes qui apparaissent (C dans l'estimateur à seuil et θ_c dans l'estimateur uniforme optimal). Il ne faut pas négliger ces constantes qui pourraient avoir un réel impact sur la performance des algorithmes. Cependant il n'est pas évident de les déterminer. C'est là qu'intervient l'algorithme à allocation dynamique.

B. L'algorithme associé

Nous avons précédemment remarqué que, asymptotiquement parlant, la variance est déterminée par le nombre de scénarios et décroît en n^{-1} , et le biais par la moyenne du nombre d'échantillons internes décroît en \bar{m}^{-2} .

L'idée de l'algorithme est alors d'estimer le biais et la variance au fur et à mesure qu'on ajoute des simulations. On décidera en conséquence s'il vaut mieux créer un nouveau scénario ou ajouter une simulation au scénario avec la plus petite fiabilité.

Donc si l'estimation du carré du biais domine celle de la variance, on préférera ajouter des simulations internes, et inversement, si c'est la variance qui est plus grande, on voudra ajouter de nouveaux scénarios.

Un heuristique du théorème central limite nous donne les approximations suivantes pour le biais et la variance:

$$E[\hat{\alpha} - \alpha] \approx \hat{B} = \hat{\alpha} - \bar{\alpha}, \text{ et } Var(\hat{\alpha}) \approx \hat{V} = \frac{\bar{\alpha}(1-\bar{\alpha})}{n}$$

$$\text{où on a posé } \bar{\alpha} := \frac{1}{n} \sum_{i=1}^n \Phi\left(\frac{m_i(\hat{L}_i - c)}{\sigma_i}\right)$$

L'algorithme procède ainsi :

- On se donne un nombre n_0 de scénarios à générer, et m_0 le nombre initial d'échantillons internes par scénario.
- On se donne un budget k de simulations interne que l'on divise en intervalles $K = \frac{k}{\tau_e}$ de longueur τ_e . Par construction, le premier intervalle est de longueur $\tau_e - n_0 m_0$ à cause de la première étape.
- Puis, dans chaque intervalle:

On estime le biais et la variance par les formules données ci-dessus.

Si on a \bar{m} simulations internes en moyenne, et n scénarios externes, et qu'on suppose qu'on en aura \bar{m}' et n' respectivement à la fin de l'intervalle, on peut estimer le carré du biais et la variance qu'on aura par:

$$\hat{B}^2 \times (\frac{\bar{m}}{\bar{m}'})^4 \text{ et } \hat{V} \times \frac{n}{n'}$$

On cherche donc \bar{m}' et n' qui minimisent $\hat{B}^2 \times (\frac{\bar{m}}{\bar{m}'})^4 + \hat{V} \times \frac{n}{n'}$ (l'erreur quadratique estimée du prochain intervalle), sous les contraintes:

$$\bar{m}' n' = \bar{m} n + \tau_e \text{ (ajout de } \tau_e \text{ simulations internes)}$$

$$n \leq n' \leq n + \tau_e \text{ (on ne peut pas retirer de scénario, ni en ajouter plus de } \tau_e \text{)}$$

$$\bar{m}' \geq 0$$

Les paramètres optimaux sont donnés par:

$$n' = \min \left\{ \max \left\{ \left(\frac{\hat{V} n}{4 \hat{B}^2 \bar{m}^4} (\bar{m} n + \tau_e)^4 \right)^{1/5}, n \right\}, n + \tau_e \right\}$$

$$\bar{m}' = \frac{\bar{m} n + \tau_e}{n'}$$

On ajoute donc $n' - n$ scénarios pour en avoir le nombre voulu, et on ajoute les τ_e simulations supplémentaires comme dans l'algorithme séquentiel (en ajoutant d'abord des simulations aux scénarios qui ont moins de m_0 simulations).

VI. Explication du code et étude d'exemples

Nous entamons la dernière partie de notre projet qui porte sur l'application des méthodes étudiées précédemment à deux exemples bien précis:

- Un portefeuille constitué d'un unique actif suivant une distribution gaussienne
- Un portefeuille constitué d'un put dont le sous-jacent suit une distribution log-normale

Nous présentons le code, les applications numériques et les résultats obtenus.

A. Le code

Nous avons commencé par créer une classe abstraite `Modele` dont héritent les deux modèles que nous avons implémenté, le modèle gaussien et le modèle Black-Scholes (log-normal). Cette classe servira à avoir un code le plus générique et extensible possible, et à définir toutes les méthodes dont nous aurons besoin pour qu'un modèle soit utilisable dans le reste du code.

Nous avons ensuite créé une autre classe abstraite `Payoff` qui contient une méthode prenant un modèle en paramètre (plus exactement un pointeur sur un modèle), et renvoyant le payoff en T actualisé à la date τ . De celle-ci héritent des payoffs tels qu'un call, un put, ou une digitale.

Dans un troisième fichier, nous avons créé une classe `Scenario` qui stockera la valeur du sous-jacent en τ , le nombre de simulations internes, ainsi que l'estimation de la perte pour ce scénario. Vous trouverez ensuite une classe `Sample` qui stocke un échantillon de Scénarios, la valeur initiale du portefeuille, la loi externe suivie par l'actif, la loi interne (qui est faite pour être modifiée en fonction du scénario sur lequel on travaille en changeant sa valeur initiale), ainsi que le type de payoff de notre "tradable".

Vous trouverez aussi d'autres méthodes et fonctions diverses, documentées dans le code, et qui servent à implémenter nos estimateurs.

Enfin, nous avons implémenté dans un quatrième fichier nos estimateurs en nous servant de tout ce qui a été écrit précédemment.

Pour l'uniforme, nous générons simplement les n scénarios avec m simulations internes un par un en ajoutant 1 à α si l'estimation de la perte dépasse c , puis calculons $\hat{\alpha}_{n,m} = \frac{\alpha}{n}$ associé.

Pour le séquentiel, nous générons un échantillon de n scénarios et m_0 simulations

internes chacun, puis trions le vecteur des scénarios par fiabilité. Ensuite nous ajoutons une simulation au premier scénario en le remettant à sa place avec sa nouvelle fiabilité. Nous calculons, après avoir ajouté le nombre de simulations voulu, le $\hat{\alpha}$ associé.

Pour l'algorithme à seuil de fiabilité, nous créons les n scénarios un par un, en calculons la fiabilité, et ajoutons un scénario tant que le seuil n'est pas dépassé. Puis calculons le $\hat{\alpha}_{\gamma, n}$ associé.

Pour l'algorithme à allocation dynamique, nous créons d'abord un échantillon de n_0 scénarios, avec m_0 simulations chacun, puis trions cet échantillon par fiabilité. Par la suite, notre vecteur de scénario sera décomposé en deux parties en enregistrant le rang de la séparation virtuelle: les n_{Fiab} premiers éléments sont ceux qui ont déjà atteint les m_0 simulations internes, et sont triés par ordre de fiabilité croissante. Ceux qui suivent n'ont pas encore atteint ce seuil et sont triés par ordre de nombre de simulations décroissant. Ensuite, sur chaque intervalle (de taille τ_e), nous calculons la cible du nombre de scénarios à atteindre, nous ajoutons le nombre de scénarios manquant à la fin de l'échantillon jusqu'à ce que chaque scénario ait m_0 simulations internes, puis nous ajoutons les simulations restantes de l'intervalle au début de l'échantillon comme dans l'algorithme séquentiel.

Remarquons que ces algorithmes sont fait pour fonctionner avec notre code pour pouvoir ajouter des modèles d'évolutions de sous-jacents ou des payoffs différents. Nous n'utilisons pas la classe échantillon pour les algorithmes uniforme et à seuil, pour économiser la mémoire comme expliqué précédemment.

B. Sorties pour l'exemple Gaussien

Dans ce premier exemple, le portefeuille est constitué d'un unique actif suivant une distribution gaussienne.

En notant S_t la valeur de l'actif à l'instant t , on suppose donc que:

- entre 0 et τ , $S_t \sim N(X_0, \sigma_{outer})$
- entre τ et T , $S_t \sim N(X_\tau, \sigma_{inner})$

Et nous fixons des valeurs numériques: $S_0 = X_0 = 0$, $\sigma_{outer} = 1$, $\sigma_{inner} = 5$.

Le pay-off à l'instant T est simplement S_T .

Pour cet exemple, nous choisissons trois valeurs du seuil c : 1.282, 2.326 et 3.090 correspondant chacune respectivement à des probabilités de pertes α de 10%, 1% et 0.1%.

Disposant de toutes les informations nécessaires, nous avons implémenté cet exemple.

Les résultats sont présentés dans le tableau ci-dessous.

Gaussien		n	m	Espérance	Variance	Biais ²	Erreur quad.	EQ norm.
$\alpha = 10\%$	Uniforme 1/3:2/3	25 199	159	11,67%	4,23E-06	2,80E-04	2,84E-04	15,2
	Uniforme optimal	4 499	889	10,31%	1,99E-05	9,38E-06	2,92E-05	1,6
	Séquentiel	5 000	400	10,08%	1,79E-05	7,09E-07	1,86E-05	1,0
$\alpha = 1\%$	Uniforme 1/3:2/3	25 199	159	1,54%	6,04E-07	2,86E-05	2,92E-05	16,7
	Uniforme optimal	5 089	786	1,11%	2,29E-06	1,19E-06	3,48E-06	2,0
	Séquentiel	5 000	400	1,00%	1,75E-06	8,04E-10	1,75E-06	1,0
$\alpha = 0.1\%$	Uniforme 1/3:2/3	25 199	159	0,20%	7,71E-08	1,08E-06	1,16E-06	5,3
	Uniforme optimal	7 788	514	0,13%	1,58E-07	8,34E-08	2,42E-07	1,1
	Séquentiel	5 000	400	0,10%	2,17E-07	1,41E-09	2,19E-07	1,0

Ces sorties ont été produites par notre code en utilisant les valeurs données dans l'article, sur 1000 simulations, puis analysées et présentées à l'aide d'Excel.

La dernière colonne représente l'erreur quadratique normalisée par l'erreur quadratique de l'estimateur séquentiel.

Nous remarquons que dans nos algorithmes séquentiels, la variance domine largement le carré du biais. Cela est dû au fait du choix de n et m qui ne sont pas optimaux. Le programme étant trop lent pour le refaire tourner, nous affichons tout de même les sorties, qui nous permettent d'observer que même sans choisir ces paramètres de manière optimale, l'estimateur séquentiel reste bien mieux que l'uniforme.

C. Sorties pour l'exemple du Put sous Black-Scholes

Dans ce second exemple, le portefeuille se résume à une position longue sur un put de maturité T .

Fixons la valeur numérique des paramètres entrant en jeu:

- Valeur en 0 du sous-jacent: $S_0 = 100$
- Drift: $\mu = 8\%$
- Volatilité annualisée: $\sigma = 20\%$
- Taux sans risques: $r = 3\%$
- Strike du Put: $K = 95$
- Maturité: $T = \frac{1}{4}$, en années
- Horizon de risque: $\tau = \frac{1}{52}$, en années

Par la formule de Black-Scholes, nous déterminons la valeur en 0 du put: $X_0 = 1.669$

Le sous-jacent S , suit une loi log-normale.

Entre 0 et τ , il évolue sous la probabilité monde réel.

Sa valeur en τ est donc donnée par: $S_\tau = S_0 \exp((\mu - \frac{\sigma^2}{2})\tau + \sigma\sqrt{\tau}\omega)$, où ω est une gaussienne centrée réduite, et sa valeur initiale est S_0 .

Entre τ et T , le sous-jacent évolue sous la probabilité risque neutre, le drift est donc remplacé par le taux sans risque, et sa valeur initiale est S_τ .

Pour un scénario ω , la perte du portefeuille en τ , est donnée par:

$$L(\omega) = X_0 - X_\tau = X_0 - E[e^{-r(T-\tau)}(K - S_T(\omega, W))_+ | \omega].$$

De plus, la perte sur l'étape interne, prend la forme: $\hat{Z}_{i,j} = X_0 - e^{-r(T-\tau)}(K - S_T(\omega_i, W_{i,j}))_+$, pour un scénario externe ω_i .

Pour ce second exemple, nous choisissons trois valeurs du seuil c : 0.859, 1.221 et 1.390 correspondant chacune respectivement à des probabilités de pertes α de 10%, 1% et 0.1%.

Disposant ainsi de toutes les informations nécessaires, nous avons implémenté cet exemple. Les résultats sont présentés dans le tableau ci-dessous.

Put		n	m	Espérance	Variance	Biais ²	Erreur quad.	EQ norm.
$\alpha = 10\%$	Uniforme 1/3:2/3	25 199	159	12,25%	4,26E-06	5,07E-04	5,11E-04	26,5
	Uniforme optimal	5 095	785	10,52%	1,81E-05	2,67E-05	4,48E-05	2,3
	Séquentiel	5 000	400	10,13%	1,77E-05	1,62E-06	1,93E-05	1,0
$\alpha = 1\%$	Uniforme 1/3:2/3	25 199	159	1,96%	8,10E-07	9,18E-05	9,27E-05	42,6
	Uniforme optimal	3 143	1 273	1,12%	3,49E-06	1,39E-06	4,88E-06	2,2
	Séquentiel	5 000	400	1,00%	2,17E-06	5,57E-10	2,17E-06	1,0
$\alpha = 0.1\%$	Uniforme 1/3:2/3	25 199	159	0,38%	1,57E-07	7,87E-06	8,03E-06	39,5
	Uniforme optimal	2 570	1 556	0,12%	4,92E-07	5,65E-08	5,49E-07	2,7
	Séquentiel	5 000	400	0,10%	2,03E-07	2,19E-10	2,03E-07	1,0

Ces sorties ont été produites par notre code en utilisant les valeurs données dans l'article, sur 1000 simulations, puis analysées et présentées à l'aide d'Excel.

La dernière colonne représente l'erreur quadratique normalisée par l'erreur quadratique de l'estimateur séquentiel.

VII. Conclusion

Nous avons vu, au cours de notre étude, une méthode pour estimer la probabilité d'une forte perte à une date donnée dans le futur, pour un portefeuille ne dépendant que d'un actif. Elle repose sur l'idée de simulations séquentielles imbriquées, et nous avons alors étudié différentes versions de cette méthode.

Nous avons commencé par regarder la manière la plus naïve de l'implémenter, c'est à dire en allouant à chaque scénario externe le même nombre de simulations.

Nous avons ensuite cherché divers moyens pour l'optimiser. Tout d'abord, nous avons cherché à connaître le meilleur rapport entre le nombre de scénarios à générer et le nombre de simulations par scénario. Cela a abouti en une décroissance de l'erreur quadratique en $k^{-2/3}$ où k est le nombre totale de simulations internes que l'on se donne.

Enfin, nous avons remarqué qu'il n'était pas optimal d'allouer autant de simulations internes à chaque scénario. En introduisant la fiabilité de chacun des scénarios, avons exposé une version dans laquelle nous voulions que chaque scénario ait une fiabilité minimum fixée, et avons donc dû étudier le nombre moyen de simulations nécessaires pour dépasser ce seuil. Enfin, une dernière méthode est de fixer le nombre de simulations k et un nombre de scénarios minimum, et de décider de l'ajout de nouveaux scénarios au fur et à mesure de l'algorithme en estimant le biais et la variance.

Ces méthodes d'allocation séquentielle nous ont permis d'atteindre un ordre de $k^{-4/5}$ pour l'erreur quadratique, qui est bien meilleur que celui trouvé pour l'algorithme optimal uniforme.

Nous avons appliqué ces méthodes à deux exemples, et les résultats obtenus corroborent bien la théorie étudiée auparavant.

La méthode s'avère efficace mais coûteuse en temps de calcul. Ce coût peut néanmoins être optimalement utilisé et la méthode améliorée en procédant à un échantillonnage précis et adapté à chaque étape.