

Prof. Pierre L'Ecuyer

DEVOIR 3

Devoir à remettre le *mercredi 14 octobre 2020, 13h30, avant le début du cours*, par courriel au professeur.

Les devoirs doivent être faits individuellement ou en équipe de deux. Il est très important de bien expliquer tout ce que vous faites et d'expliquer le sens de vos résultats (discussion), pour les expériences numériques et aussi pour les preuves. Dans la correction, on accordera davantage d'importance à la clarté des explications qu'aux programmes informatiques et aux résultats comme tels. Dans ce sens, remettre uniquement un programme qui donne les bons résultats ne suffit pas et vous donnera peu ou pas de points (nous ne corrigeons pas les programmes comme tels). Vous devez quand même envoyer une copie de vos programmes par courriel avec votre devoir, pour que le professeur puisse les vérifier au besoin.

Attention au plagiat: il n'est pas permis de copier et/ou modifier les programmes ou les solutions venant de quelqu'un d'autre.

1. (10 points) Aucune programmation requise pour cette question. Aucun calcul demandé.

Consider the discretely-observed Asian option of Example 1.8 of the class notes, but suppose that instead of estimating the price $v(s_0, T)$, we want to estimate the *derivative* of $v(s_0, T) = v(s_0, r, \sigma, T)$ with respect to σ ,

$$v_\sigma(s_0, r, \sigma, T) = \frac{\partial v(s_0, r, \sigma, T)}{\partial \sigma},$$

which is known as the *vega of the option*. Here we added the parameters r and σ , which were implicit in the notation in Example 1.8. One possible estimator of this derivative is the finite difference:

$$X(\delta) = \frac{Y(s_0, r, \sigma + \delta, T) - Y(s_0, r, \sigma, T)}{\delta},$$

where $Y(s_0, r, \sigma, T) = e^{-rT}g(S(t_1), \dots, S(t_d))$ is the sample discounted payoff and $\delta > 0$ is a small constant. We could simulate n independent realizations of $X(\delta)$, take the average as our estimator of the vega, and compute a confidence interval.

(a) Explain why this is a biased estimator when $\delta > 0$ is fixed. What happens to the bias when $n \rightarrow \infty$ while δ is fixed? And when $\delta \rightarrow 0$?

(b) To simulate $X(\delta)$, we must perform two simulations of the payoff, one at σ and one at $\sigma + \delta$. These two simulations can be performed with independent random numbers (IRN) or with common random numbers (CRNs). Explain the difference. Explain how you would implement the simulations with CRNs in this particular case. What happens with the variance of $X(\delta)$ when $\delta \rightarrow 0$, in each case?

(c) Is it possible to obtain an unbiased estimator of the vega by using the stochastic derivative in this case? If yes, explain how, derive the estimator mathematically, and prove that it is unbiased.

2. (10 points)

Exercice 2.34 des notes.

3. (30 points)

Une succursale d'une banque ouvre à 10h et ferme à 16h. On a deux types de clients qui viennent à la banque: ceux qui arrivent sans rendez-vous et se présentent à un guichet pour une transaction (type A), et ceux qui prennent un rendez-vous pour une affaire plus complexe comme une hypothèque, emprunt, REER, etc. (type B). Il y a aussi deux types d'employés qui servent les clients: les caissiers (type C) qui ne servent que les clients qui viennent aux guichets sans rendez-vous, et les conseillers (type D), qui peuvent servir tous les types de clients. Ces derniers servent en priorité les clients qui ont rendez-vous avec eux, mais quand ils sont libres et qu'il y a des clients en attente aux guichets, ils peuvent aussi servir ces derniers.

La journée est divisée en trois périodes: 10h à 12h (période 1), 12h à 14h (période 2), et 14h à 16h (période 3). Durant la période j , il y a n_j caissiers et m_j conseillers au travail. Quand le nombre de caissiers ou conseillers augmente d'une période à l'autre, ce sont les mêmes employés auxquels s'ajoutent des nouveaux. Quand il diminue, il y en a qui quittent et d'autres qui restent, mais pas de nouveaux. Après 16h, on ne laisse plus entrer de nouveaux clients, mais on va servir tous ceux qui sont déjà entrés, et chaque employé va quitter dès qu'il n'a plus personne à servir. Idem pour un employé qui quitte à la fin d'une période: il va toujours terminer le service en cours avant de quitter.

On suppose que les clients de type A arrivent selon un processus de Poisson de taux λ_j durant la période j , pour $j = 1, 2, 3$. Leurs durées de service sont indépendantes et suivent une loi lognormale de moyenne μ_a et d'écart-type σ_a , peu importe qui les sert. (Attention: ce sont la moyenne et l'écart-type, et non pas les paramètres de la loi.) Ils sont servis par ordre d'arrivée. On suppose qu'il n'y a pas d'abandons.

Les clients de type B doivent avoir pris un rendez-vous avant le début de la journée. Pour cela, la journée est divisée en 12 plages de 30 minutes, et pour chaque conseiller (de type D), chaque rendez-vous est pour l'une de ces 12 plages. Chaque rendez-vous est avec un conseiller particulier. Pour simplifier, on va supposer que pour chaque conseiller et chaque plage horaire, il y aura un rendez-vous prévu dans cette plage avec probabilité r , indépendamment d'une plage à l'autre. Chaque client qui a un rendez-vous peut ne pas se présenter, disons avec probabilité p , et sinon arrive avec un retard (délai par rapport à son heure de rendez-vous) R aléatoire de loi normale de moyenne μ_r et écart-type σ_r (R peut prendre des valeurs négatives). Lorsqu'un client ne se présente pas, le conseiller doit quand même attendre jusqu'à la fin de la plage de 30 minutes avant de faire autre chose.

Les durées de service des clients de type B sont de loi lognormale de moyenne μ_b et écart-type σ_b . Les clients ayant rendez-vous pour un conseiller donné doivent être servis par ce même conseiller, selon l'ordre des rendez-vous. Si ce dernier n'est pas libre à l'heure du rendez-vous, le client doit attendre son tour. Si un conseiller est libre et n'a pas de rendez-vous prévu avant au moins s secondes, et si au moins un client de type A est en attente, alors ce conseiller va servir un client de type A. Sinon, le conseiller attend qu'il y ait un client de type A en

attente, ou qu'il reste s secondes ou moins avant son prochain rendez-vous. Dans ce dernier cas, il attend son prochain rendez-vous. Pour simplifier, vous pouvez aussi supposer que l'on ne débute jamais le service avant l'heure du rendez-vous, et que l'on ne compte jamais le temps d'attente avant l'heure du rendez-vous (ce qui est logique, car on ne veut pas pénaliser le système si le client arrive trop en avance). Lorsqu'un client de type A arrive, il est servi en priorité par un caissier (type C) s'il y en a un de libre.

Soient w_a et w_b les temps d'attente moyens par client pour les types A et B, respectivement, à long terme (sur une infinité de jours). On veut estimer w_a et w_b en simulant n jours. Soient $W_{i,a}$ et $W_{i,b}$ la somme des temps d'attente des clients de types A et B pour le jour i . Votre programme va simuler ce système et calculer ces valeurs, qui sont des réalisations de variables aléatoires.

(a) On vous demande d'écrire un programme pour simuler ce modèle et calculer les valeurs des $W_{i,a}$ et $W_{i,b}$ pour les n jours, puis estimer w_a et w_b , et construire deux histogrammes, pour les valeurs des temps d'attente (individuels) des clients de chacun des deux type pour les n jours ensemble. Vous pouvez vous inspirer un peu du programme `CallCenter.java`, sauf que chaque conseiller devra ici être un objet, car on doit les distinguer. Les clients de type B doivent savoir avec quel conseiller ils ont rendez-vous, à quelle heure, et à quelle heure ils arrivent. Les conseillers doivent savoir quand ils ont des rendez-vous. C'est peut-être une bonne idée de générer au début de la journée tous les rendez-vous, et mêmes les heures d'arrivées des rendez-vous. Chaque client qui a un rendez-vous pourra être un objet créé dès le départ et contenant l'information qui lui est associée (heure d'arrivée, nom ou numéro du conseiller, etc.).

Si le temps est mesuré en secondes dans votre programme, vous pouvez définir les constantes `MINUTE = 60` et `HOURL = 3600` pour convertir toutes les unités en secondes. Par exemple, on mettra `35/HOURL` si 35 est un taux d'arrivée par heure, et `20 * MINUTES` pour une valeur de 20 minutes.

Pour le reste, je vous laisse l'occasion de réfléchir et faire vous même un design élégant pour votre programme.

(b) Vous allez utiliser votre programme pour simuler la banque avec les valeurs suivantes, où tous les taux d'arrivées sont *par heure*: $n_1 = 3$, $n_2 = 4$, $n_3 = 3$, $m_1 = 2$, $m_2 = 3$, $m_3 = 2$, $\lambda_1 = 30$, $\lambda_2 = 40$, $\lambda_2 = 25$, $\mu_a = 240$ secondes, $\sigma_a = 50$ secondes, $\mu_r = -100$ secondes, $\sigma_r = 60$ secondes, $\mu_b = 25$ minutes, $\sigma_b = 4$ minutes, $r = 0.8$, $p = 0.05$, $s = 10$ minutes, et $n = 1000$. Votre programme va lire ces valeurs dans un fichier. Discutez vos résultats.

(c) Supposons maintenant que l'on veut estimer les dérivées de w_a et w_b par rapport à s , en utilisant des différences finies avec des valeurs aléatoires communes. On peut simuler par exemple à s puis à $s + \delta$, un peu comme on a fait pour le centre d'appels dans la section 6.2.2 de mes notes. Vous pouvez essayer par exemple $\delta = 30$ secondes ou encore $\delta = 10$ secondes. Faites cela et expliquez comment vous faites pour synchroniser le mieux possible vos nombres aléatoires, i.e., pour qu'ils soient utilisés le plus possible pour la même chose pour les deux politiques. Vous pouvez essayer et comparer différentes possibilités, comme on a fait à la section 6.2.2. Le plus important est de bien expliquer ce que vous faites et ce que vous apprenez.
