

Prof. Pierre L'Ecuyer

DEVOIR 2

Devoir à remettre le *mardi 7 octobre 2014, 9h30, avant le début du cours*. Les numéros indiqués correspondent aux *exercices* des notes de cours. Les devoirs doivent être faits individuellement: un devoir par étudiant. Pour les expériences numériques, il est très important de bien expliquer tout ce que vous faites et d'expliquer le sens de vos résultats (discussion). 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, écrire un programme qui donne les bons résultats ne suffit pas. Veuillez placer les exercices dans le bon ordre sur vos copies. Le jour de la remise, envoyez aussi une copie de vos programmes par courriel au correcteur, pour qu'il puisse les essayer au besoin: . Attention au plagiat: il n'est pas permis de copier et/ou modifier les programmes ou les solutions d'un(e) autre étudiant(e).

— 1 —

Exercice 1.6 des notes de cours.

— 2 —

Consider the discretely-observed Asian option of Example 1.8, but suppose that instead of estimating $v(s_0, T)$, we want to estimate the *derivative* of $v(s_0, T)$ with respect to s_0 ,

$$\Delta(s_0, T) = \frac{\partial v(s_0, T)}{\partial s_0},$$

which is called the *delta of the option*. Our estimator of this derivative will be the finite difference:

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

where $Y(s_0, T) = e^{-rT}g(S(t_1), \dots, S(t_d))$ and $\delta > 0$ is a small constant. As usual, we will simulate n independent copies of $X(\delta)$, take the average as our estimator of $\Delta(s_0, T)$, and compute a confidence interval.

(a) Explain why this is a biased estimator of $\Delta(s_0, T)$ when $\delta > 0$ is fixed. What happens to the bias when $n \rightarrow \infty$ while δ is fixed? And when $\delta \rightarrow 0$? How should we choose δ in practice if we want a small bias?

(b) Implement a program that can perform this simulation experiment in the two following ways: (1) $Y(s_0, T)$ and $Y(s_0 + \delta, T)$ are simulated with independent random numbers (IRN) for each realization of $X(\delta)$; (2) they are simulated with common (and well-synchronized) random numbers (CRN). You can start from the program `Asian.java` available in the examples of SSJ, and seek some inspiration from the program `InventoryCRN.java` also available there.

- (c) Perform an experiment to compare your derivative estimators with IRN and CRN, with various values of δ , with the same option parameters as in the program `Asian.java`, and with $n = 10000$. You have the freedom (and the task) of selecting and comparing the values of δ that you will try. Compare the variances and discuss your results. What happens to the variance when δ becomes very small, with IRN and with CRN? What are your recommendations?
- (d) For the estimator with IRN, derive (mathematically) an exact expression for the limit $\lim_{\delta \rightarrow 0} \delta^2 \text{Var}[X(\delta)]$, as a function of $\sigma = \text{Var}[Y(s_0, T)]$. Does this explain your results for IRN in (c)?
- (e) Finally, perform an experiment as in (c), but using the stochastic derivative as an estimator. Compare and discuss.

— 3 —

Vous allez resimuler une file en tandem comme à la question 4 du devoir 1, mais cette fois il y aura trois serveurs à chaque station au lieu d'un seul, et dans les expériences numériques, le taux de service de chaque serveur sera $\mu_1 = 0.5$ à la station 1, et $\mu_2 = \mu_3 = 0.4$ aux deux autres stations. Les durées de service suivent donc une loi exponentielle, avec la même moyenne pour tous les serveurs d'une même station.

Dans ce cas-ci, les récurrences données dans l'exemple 1.7 des notes ne tiennent plus, et il est plus difficile d'écrire des récurrences, parce que les clients peuvent se dépasser aux stations. On vous demande d'implanter une classe `TandemQueueEv` qui va effectuer la simulation en utilisant des événements. Votre classe devra pouvoir prendre un nombre arbitraire (≥ 1) de serveurs à chaque station. Elle doit être implantée de manière simple et élégante (autant que possible). Le reste sera comme dans la question 4 du devoir 1, sauf qu'il suffit de fournir une méthode `simulateFixedTime` pour simuler le système sur un horizon de temps T fixé (pas besoin d'implanter une méthode pour un nombre de clients fixé).

Faites les mêmes expériences que dans la question 4 du devoir 1 et comparez vos résultats.