MÉTHODES DE MONTE-CARLO
« EXAMEN BLANC »
par Rémi Peyre

PROBLÈME 1 — La récompense de Mickey Thunk

Mickey THUNK, *génial informaticien, vient d'achever la programmation du logiciel* T$\bigcirc$C, *destiné à révolutionner la monde de la typographie.* [*] *Il est tellement fier de sa programmation qu'il défie quiconque de trouver le moindre bug dans son programme ! Une personne qui trouverait un tel bug remporterait ainsi un prix de 100 €. Mieux, une personne qui trouverait un second bug après le premier remporterait pour sa part 200 €, la troisième 400 €, et ainsi de suite en doublant la prime à chaque nouveau bug trouvé.*

En fait, Mickey Thunk ne pense pas que son programme soit totalement exempt de bugs, mais il estime qu'il ne doit plus en rester que « trois ou quatre ». Il modélise le nombre total de bugs dans son programme par une loi de Poisson de paramètre 3,5. Notons X *une variable aléatoire suivant, sous une probabilité* $\mathbb{P}$, *la loi* $\mathcal{P}oisson(3{,}5)$, *et* Y *la variable aléatoire correspondant au montant total de la récompense, qui est une fonction de* X. *Mickey Thunk se demande combien va coûter sa récompense en moyenne ; autrement dit, il souhaite calculer* $\mathbb{E}(Y)$ *(il estime que tous les bugs finiront par être trouvés).*

1. Écrire une fonction MATLAB appelée `Poisson` qui prend un argument λ et renvoie un nombre aléatoire suivant la loi $\mathcal{P}oisson(\lambda)$.
Indication : On rappelle que, si $T_1, T_2, \ldots$ sont des v.a. i.i.d. suivant une loi $\mathcal{E}xponentielle(\lambda)$, alors

$$\sup\left\{k \in \mathbb{N} : \sum_{i=1}^{k} T_i \leqslant 1\right\} \tag{1}$$

suit la loi $\mathcal{P}oisson(\lambda)$.

2. Écrire une fonction MATLAB appelée `Y` qui prend un argument X et renvoie la valeur de la récompense totale Y correspondant au nombre de bugs X. On essaiera de programmer cette fonction de façon qu'elle soit la plus rapide possible.

3. Évaluer $\mathbb{E}(Y)$ par une méthode de Monte-Carlo « naïve ». On écrira une fonction MATLAB appelée `MCnaif` qui prend en argument le nombre N de simulations demandées et renvoie un intervalle de confiance à 90 % (soit $\pm 1{,}65$ sigmas) pour $\mathbb{E}(Y)$. (On admettra que Y est L^2).

On cherche maintenant à évaluer $\mathbb{E}(Y)$ *à l'aide d'un échantillonnage préférentiel. La loi d'échantillonnage retenue est une loi de Poisson de paramètre* 5.

4. Quelle est la desité relative de la loi $\mathcal{P}oisson(5)$ par rapport à la loi $\mathcal{P}oisson(3{,}5)$?
Indication : On rappelle que si P est une loi de Poisson de paramètre λ, alors pour tout $n \in \mathbb{N}$

$$P(\{n\}) = \frac{\lambda^n}{n! e^{\lambda}}. \tag{2}$$

5. Évaluer $\mathbb{E}(Y)$ par la méthode de Monte-Carlo avec échantillonnage préférentiel suggerée ci-dessus. On écrira une fonction MATLAB appelée `MCpref` qui prend en argument le nombre N de simulations demandées et renvoie un intervalle de

[*]. Toute ressemblance avec une situation réelle serait purement volontaire.

confiance à 90 % pour $\mathbb{E}(Y)$. (On admettra qu'il n'y a toujours pas de souci avec les questions d'intégrabilité L^2).

PROBLÈME 2 — Mouvement brownien fractionnaire

On définit le mouvement brownien fractionnaire de paramètre $7/8$ *comme un processus* gaussien centré $(H_t)_{t\geqslant 0}$ *à valeurs réelles tel que, pour tous* $0 \leqslant s \leqslant t$, $(H_t - H_s)$ *ait pour écart-type* $(t-s)^{7/8}$*. On admettra qu'un tel processus existe avec des trajectoires continues.*

1. Montrer que pour tous $t_1, t_2 \in \mathbb{R}_+$,

$$\mathrm{Cov}(H_{t_1}, H_{t_2}) = \tfrac{1}{2}\big(t_1^{7/4} + t_2^{7/4} - |t_2 - t_1|^{7/4}\big). \tag{3}$$

2. Simuler ce mouvement brownien fractionnaire (en le plottant) sur l'intervalle $[0,1]$ par la méthode de Cholesky. Le code-source sera une fonction MATLAB appelée `MBf`, prenant en argument le nombre points retenus pour la discrétisation, ceux-ci étant régulièrement espacés. Lancer cette simulation pour un nombre de points au moins égal à 1 024, et faire une copie d'écran du diagramme obtenu (au format `.png` de préférence), la figure étant affichée en grande fenêtre.

3. Le mouvement brownien fractionnaire n'est pas un processus markovien. Sauriez-vous expliquer pourquoi (sans faire une preuve formelle), rien qu'en regardant l'allure de la courbe obtenue ?
Indication : Il pourra éventuellement être utile de zoomer sur le diagramme.

4. On définit la variable aléatoire $X = \sup_{t\in[0,1]} |H_t|$. En admettant que 1024 points de simulation constituent une approximation suffisante pour la question qui nous intéresse, évaluer $\mathbb{E}(X)$ par la méthode de Monte-Carlo. Le programme s'appellera `supMBf_a`, prendra comme seul argument le nombre de simulations demandé, et reverra un intervalle de confiance à 2 sigmas pour $\mathbb{E}(X)$.

5. Calculer exactement $\mathbb{E}(|H_1|)$.

6. En utilisant $|H_1|$ comme variable de contrôle, améliorer la méthode de Monte-Carlo précédente. Le programme s'appellera `supMBf_b`.

7. Modifier les fonctions `supMBf_a` et `supMBf_b` pour que, au cours de leur exécution, elles affichent aussi leur efficacité (inutile de changer les noms des fonctions). Comparer les efficacités et commenter.

PROBLÈME 3 — Rétroingénierie

Question. Voici un programme. Dites tout ce que vous trouverez d'intelligent à dire dessus.

```
function A = mystere(B)
C = min(B,4096);
D = B-C;
E = 0;
F = 0;
for G = 1:C
    H = sqrt(1-rand^2);
    E = E+H;
    F = F+H*H;
end
I = F/C-(E/C)*(E/C);
```

```
for J = 1:D
    K = sqrt(1-rand^2);
    E = E+K;
end
A = 4*(E/B+I/sqrt(B)*[-3,3]);
end
```