

Exercices sur la méthode de Monte-Carlo

Rémi PEYRE

Février 2016

Fondements de la méthode de Monte-Carlo

EXERCICE 1 — Le moment quatrième

Soit P la loi normale standard. Le but de cet exercice est de déterminer le moment quatrième de P , c.à.d. la quantité $\int_{\mathbf{R}} x^4 dP(x)$, notée M_4 dans la suite. Pour ce faire, nous allons nous laisser guider par des considérations “expérimentales” reposant sur la méthode de Monte-Carlo.

1. Reformuler M_4 comme l’espérance d’une variables aléatoire Y que l’on sait simuler facilement (en supposant qu’on sait déjà simuler P). Écrire une fonction `loi_de_Y` simulant la loi de Y .

2. Écrire un programme `MC_simple` évaluant M_4 par la méthode de Monte-Carlo (à partir de l’écriture comme espérance vue à la question 1), en utilisant 5 000 000 simulations. Ce programme ne prendra aucun argument, affichera la valeur estimée avec une précision de 6 chiffres après la virgule, et ne renverra rien.

Indication : La loi normale standard peut être simulée à l’aide de la fonction `randn` de MATLAB. Consulter « doc `randn` » pour plus de détails.

3. Justifier que la loi de la variable aléatoire Y de la question 1 est de classe L^2 .

4. Modifier le programme `MC_simple` en un programme `MC_IC` qui, outre l’estimation de M_4 , calcule et affiche aussi la marge d’erreur (au risque 1 %) associée à cette estimation. Lancer `MC_IC`; et au vu du résultat obtenu, émettre une conjecture sur la valeur exacte de M_4 .

5. Écrire aussi une variante du programme `MC_IC` appelée `MC_sup` dans laquelle, au lieu de calculer un intervalle de confiance pour M_4 , on cherche à en donner un majorant aussi fin que possible, sachant qu’on s’autorise là encore un risque d’erreur (asymptotique) de 1 %.

6. Écrire aussi une autre variante du programme `MC_IC` appelée `MC_homogene_IC` dans laquelle, au lieu de donner un intervalle de confiance pour M_4 , on donne un intervalle de confiance pour $M_4^{1/4}$.

7. Chronométrer le temps pris par l’exécution du programme `MC_IC`. En déduire combien de simulations vous pouvez demander si vous disposez de 2 minutes pour faire tourner votre programme; et modifier `MC_IC` en une variante `MC_IC_2min` procédant à ce nombre de simulations. Prévoir la marge d’erreur qu’on devrait obtenir avec cette variante. Lancer et chronométrer `MC_IC_2min` et vérifier que le temps mis et la marge d’erreur trouvée sont cohérentes avec vos prédictions. Les résultats de `MC_IC_2min` renforcent-ils la conjecture émise à la question 4?

8. (★) Démontrer rigoureusement la conjecture émise à la question 4.

EXERCICE 2 — Autour de la génération pseudo-aléatoire

Le but de cet exercice est de procéder à quelques expérimentations concrètes pour mieux appréhender les problématiques liées à la génération pseudo-aléatoire. On s’appuiera dans cet exercice sur un programme de base `MC` procédant à une estimation par Monte-Carlo assortie

de sa marge d'erreur : il pourra s'agir, par exemple, du programme `MC_IC` de l'exercice 1 ; mais tout autre programme de même nature conviendrait aussi bien.

1. Lancer trois fois de suite le programme `MC`. Les résultats obtenus sont-ils identiques ? Était-ce prévisible ? Pourquoi ce comportement est-il essentiel vis-à-vis de la méthode de Monte-Carlo ?

2. Redémarrer `MATLAB` ; lancer le programme `MC` ; noter le résultat sur un papier ; redémarrer encore `MATLAB` ; lancer à nouveau le programme `MC`. Que constate-t-on ? Comment cela s'explique-t-il ?

3. Reprendre la question 2, à ceci près qu'avant de lancer `MC` la première fois on exécutera la commande `rng ('tartempion')`, alors qu'avant de lancer `MC` la deuxième fois on exécutera plutôt la commande `rng ('trucmuche')`. Que constate-t-on ? Expliquer.

4. Expliquer comment on peut forcer `MATLAB` à adopter un comportement pseudo-aléatoire prévisible sans avoir à redémarrer le logiciel ; et tester expérimentalement votre méthode.

5. Expliquer quel peut être l'intérêt d'avoir un comportement pseudo-aléatoire prévisible pour un enseignant qui souhaite illustrer expérimentalement auprès de ses élèves qu'il peut arriver qu'un intervalle de confiance à 99,99 % ne contienne pas la bonne valeur.

6. Trouver un autre intérêt que peut avoir un comportement pseudo-aléatoire prévisible.

Indication : On pourra penser, par exemple, à la problématique suivante : « Comment se défendre d'une accusation de fraude lorsqu'on réalise une expérience aléatoire ? ». Une autre application serait pour transférer des données de simulation aléatoire très volumineuses malgré une connexion de faible débit...

On se pose maintenant la question de savoir quelle est la relation entre les suites pseudo-aléatoires générées depuis deux graines différentes.

7. Écrire une fonction `rand999` qui simule un nombre pseudo-aléatoire entier uniformément réparti entre 0 et 999. Écrire une fonction `echantillon_pseudoaleatoire` qui appelle 3 000 fois la fonction `rand999` et renvoie un tableau contenant le nombre d'occurrences de chaque valeur entre 0 et 999. Appeler deux fois la fonction `echantillon_pseudoaleatoire`, une fois après avoir invoqué `rng (0)`, et la seconde fois après avoir invoqué `rng (1)`. Les résultats obtenus vous semblent-ils compatibles avec l'idée que les suites pseudo-aléatoires issues des graines 0 et 1 seraient indépendantes ? En quoi cela est-il intéressant ?

EXERCICE 3 — Parallélisation

Le but de cet exercice est s'exercer à l'utilisation des procédés de parallélisation que permet la méthode de Monte-Carlo. On s'appuiera dans cet exercice sur un programme de base `MC` procédant à une estimation par Monte-Carlo assortie de sa marge d'erreur : il pourra s'agir, par exemple, du programme `MC_IC` de l'exercice 1 ; mais tout autre programme de même nature conviendrait aussi bien.

1. En chronométrant le temps que met votre programme de base `MC`, estimez la précision que vous pouvez espérer obtenir pour votre estimation si vous disposez d'une heure pour faire tourner votre calcul.

2. Supposons que vous soyez 20 élèves dans la classe, chacun muni de son propre ordinateur. Comment mettre en commun vos capacités de calcul pour obtenir un résultat plus précis qu'à la question précédent, tout en respectant la limite de temps impartie d'une heure ?

3. Quel piège convient-il d'éviter lors de cette procédure de parallélisation ? Comment y remédier ?

Indication : Confer exercice 2...

4. La procédure de parallélisation mise en place à la question 2 améliore-t-elle l'efficacité du calcul si l'on considère que le cout du calcul correspond au temps mis ? Et si on considère que le cout de calcul correspond au nombre d'instructions exécutées par le ou les processeurs ?

5. Imaginons maintenant qu'on vous demande de faire un calcul de haute précision par la méthode de Monte-Carlo, qui prendrait une journée entière à votre processeur. Nous imaginerons que vous ne puissiez pas vous permettre de faire tourner votre processeur pendant 24 heures d'affilée ; mais qu'en revanche, comme le résultat demandé n'est exigé que pour dans un mois, vous pouvez sans souci le faire tourner une heure par jour chaque jour. Expliquer comment vous pouvez utiliser une procédure de parallélisation "dans le temps" pour atteindre votre objectif. Mettre cette technique en pratique sur un exemple simplifié.

EXERCICE 4 — Vérification des propriétés énoncées en cours

Dans cet exercice, nous allons vérifier expérimentalement quelques-unes des propriétés énoncées dans le cours au sujet de la méthode de Monte-Carlo. On s'appuiera dans cet exercice sur un programme de base MC procédant à une estimation par Monte-Carlo assortie de sa marge d'erreur, interfacé de sorte que le programme prenne en entrée le nombre de simulations demandées et le niveau de risque demandé, et renvoie en sortie l'estimation calculée, la variance efficace (estimée) de l'estimateur, la marge d'erreur associée, les limites inférieure et supérieure de l'intervalle de confiance, le temps mis par le calcul, et le produit temps-variance.

1. Lorsqu'on exécute indépendamment plusieurs fois le programme MC avec les mêmes jeux de paramètres, sommes-nous censés trouver (à peu près) les mêmes estimateurs ? la même variance de l'estimateur ? les mêmes marges d'erreur ? les mêmes intervalles de confiance ? le même temps de calcul ? le même produit temps-variance ? Vérifier expérimentalement vos prévisions.

2. Lorsqu'on fait varier le nombre de simulations demandé (à niveau de risque égal), lesquelles des quantités ci-dessus sont-elles censées rester (à peu près) les mêmes ? Lesquelles sont censées varier, et de quelle façon en fonction du nombre de simulations ? Vérifier expérimentalement vos prévisions.

3. Lorsqu'on fait varier le niveau de risque demandé (à nombre de simulations égal), lesquelles des quantités ci-dessus sont-elles censées rester (à peu près) les mêmes ? Lesquelles sont censées varier, et de quelle façon en fonction du niveau de risque ? Vérifier expérimentalement vos prévisions.

4. Vérifier la compatibilité des intervalles de confiance d'un calcul à l'autre, qu'on conserve les mêmes jeux de paramètres ou non.

5. En supposant connue la valeur exacte de la quantité qu'on estime, vérifier par la méthode de Monte-Carlo que l'intervalle de confiance à 80 % de la méthode de Monte-Carlo contient effectivement la valeur exacte environ 80 % du temps.

EXERCICE 5 — Probabilité d'un brelan

Au poker (variante dite « fermée »), le joueur reçoit 5 cartes au sein d'un sabot contenant 4 cartes de chacun des 13 « hauteurs » possibles. Puis il décide d'échanger entre 0 et 3

cartes en vue d'améliorer sa main. Une combinaison intéressante est d'obtenir un brelan, c'est-à-dire (au moins) trois cartes de même hauteur. Le but de cet exercice est d'évaluer la probabilité qu'un joueur dont le seul but serait d'obtenir un brelan atteigne son objectif. Dans cet exercice, l'identité de chaque carte sera codée par un nombre entier compris entre 0 et 51, les hauteurs des cartes seront codées par un nombre entier compris entre 2 et 14 (de façons que les cartes 0, 1, 2 et 3 aient la hauteur 2, les cartes 4, 5, 6 et 7 aient la hauteur 3, etc.), et la main d'un joueur par un 5-vecteur de telles hauteurs.

1. Écrire une fonction `hauteur` qui prenne en entrée l'identité d'une carte et renvoie en sortie la hauteur associée.

2. Écrire une fonction `sabot` qui simule l'ordre des cartes dans un sabot parfaitement mélangé, c.à.d. qui renvoie un 52-vecteur contenant une fois exactement l'identité de chaque carte, distribué aléatoirement uniformément selon toutes les combinaisons possibles.

Indication : Il est demandé aux élèves de répondre à cette question par eux-mêmes, sans s'appuyer sur des recherches sur internet... Je mentionne néanmoins que, pour implémenter la fonction `sabot` d'une façon particulièrement efficace, on peut utiliser l'algorithme de mélange de Fisher-Yates, sur lequel vous trouverez facilement de la documentation sur le web.

3. (★) Démontrer que lorsqu'un joueur se voit servir une double paire (c.à.d. deux cartes de même hauteur, deux cartes d'une même deuxième hauteur, et une troisième carte d'une troisième hauteur), son intérêt est de ne garder qu'une des deux paires et de jeter les trois autres cartes.

4. Écrire une fonction `decision` qui prend en entrée le 5-vecteur des hauteurs des cartes reçues par le joueur, et qui renvoie en sortie un autre 5-vecteur de valeurs booléennes, le i^e booléen valant *vrai* si le joueur choisit de changer cette carte, et *faux* sinon. Rappelons qu'en vertu de la question précédente, la décision du joueur doit suivre la règle suivante :

- Si trois (au moins) des cartes dans la main du joueur sont de hauteurs identiques, il en change pas sa main ;
- Sinon, si le joueur a en main une (ou deux) paire de cartes de hauteurs identiques, il conserve cette paire (attention ; il ne doit conserver qu'une seule paire dans tous les cas) et change toutes ses autres cartes ;
- Sinon, le joueur change trois cartes arbitraires de sa main.

5. Écrire une fonction `simulation_bavarde` qui décrit l'ensemble du processus par lequel notre joueur tente d'atteindre son brelan. Cette fonction affichera successivement les hauteurs des huit cartes du haut du sabot, la main initialement reçue par le joueur, la sélection de cartes que notre joueur décide de changer, la main qu'il a à l'issue de son changement, et le fait que cette main corresponde ou pas à un brelan ; et renverra *vrai* si le joueur a obtenu un brelan, et *faux* sinon. On écrit aussi une variante `simulation` de cette fonction qui fait la même chose, mais sans rien afficher.

6. Évaluer la probabilité qu'un joueur dont le seul but est d'obtenir un brelan atteigne son objectif. On exprimera le résultat sous la forme d'un intervalle de confiance à 95 %.

EXERCICE 6 — Volume d'une boule

1. Évaluer le volume de la boule unité de dimension 6^[*] par la méthode de Monte-Carlo (en prenant la loi d'échantillonnage la plus évidente), en précisant un intervalle de confiance à 90 %. Comparer avec la valeur théorique, égale à $\pi^3/6$.

[*]. C.à.d. l'ensemble des points de \mathbf{R}^6 dont la distance euclidienne à l'origine est inférieure ou égale à 1.

On veut maintenant utiliser la méthode de Monte-Carlo avec une autre loi d'échantillonnage. Soit Q la loi sur \mathbf{R}^6 caractérisée par la densité

$$\frac{dQ}{dx}(x) = Z^{-1} e^{-4\|x\|_{\ell^1}},$$

où $\|x\|_{\ell^1} := |x_1| + |x_2| + \dots + |x_6|$, et où Z est la constante qui fait de (6) une densité de probabilité.

2. Écrire une fonction simulant Q .
3. Calculer la valeur de Z .
4. Reprendre la question 1 en utilisant Q comme loi d'échantillonnage.
5. Comparer l'efficacité de la méthode de Monte-Carlo selon qu'on utilise la méthode de base où l'échantillonnage selon Q .

EXERCICE 7 — Manque d'intégrabilité L^2

On considère une loi \mathbb{P} sous laquelle la variable X suit une loi Pareto(1,5), c'est-à-dire que X est une variable aléatoire sur $[1, \infty)$ telle que pour tout $x \geq 1$, $\mathbb{P}(X \geq x) = x^{-1,5}$.

1. Calculer la densité de la loi Pareto(1,5) par rapport à la mesure de Lebesgue. En déduire que X est L^1 mais pas L^2 , et calculer (théoriquement) $\mathbb{E}(X)$.
2. Écrire un programme MATLAB qui tente d'évaluer un intervalle de confiance à 80 % de X en faisant comme si celle-ci était de classe L^2 . À l'aide de la méthode de Monte-Carlo, établit dans quelle proportion de cas est-ce que cet intervalle contient effectivement $\mathbb{E}(X)$. Qu'en concluez-vous ?
3. En faisant varier le nombre de simulations, regarder comment évolue la variance inverse par simulation. Que remarque-t-on ? Commenter.

EXERCICE 8 — Contrôle de la fonction rand

La documentation de MATLAB nous dit que la fonction `rand` est censée fournir des nombres pseudo-aléatoires indépendants uniformément répartis entre 0 et 1. Mais est-ce bien vrai?... C'est ce que nous allons tester dans cet exercice !

1. Écrire un programme `test_rand_1` qui lance 10 000 000 fois la fonction `rand` et qui compte le nombre de fois où le résultat obtenu est tombé entre 0 et 0,1, resp. entre 0,1 et 0,2, etc. (Le résultat sera renvoyé sous la forme du 10-vecteur des nombres d'occurrences de chaque cas).
2. En admettant que les productions successives de `rand` sont bien indépendantes, en déduire les intervalles de confiance respectifs de la probabilité réelle que la fonction `rand` fournisse un résultat compris entre 0 et 0,1, resp. entre 0,1 et 0,2, etc. Ces résultats sont-ils compatibles avec ce qu'annonce la documentation de MATLAB ?
3. Maintenant, on souhaite tester si les valeurs successives fournies par MATLAB se comportent bien de façon indépendante. Si r_1, r_2, r_3 sont trois valeurs successives de la fonction `rand` fournies par MATLAB, quelle est censée être la probabilité que deux ou plus de ces valeurs soient distinctes ? Que $r_1 > r_2 > r_3$? que $r_1 > r_3 > r_2$? etc.
4. Écrire un programme `test_rand_2` qui simule 1 000 000 triplets successif de variables gaussiennes, regarde pour chacun de ces triplets dans quel ordre les valeurs se classent, en

compte le nombre d'occurrences de chaque cas. En déduire l'intervalle de confiance pour la vraie probabilité de chaque classement. Ces résultats sont-ils compatibles avec ce qu'affirme la documentation ?

EXERCICE 9 — Estimation du kurtosis

Supposons qu'on dispose d'une fonction $WXYZ$ permettant de simuler la loi jointe d'un quadruplet de variables aléatoires réelles (W, X, Y, Z) (de façon indépendante à chaque appel de la fonction). On suppose qu'on a effectué une boucle de la simulation au cours de laquelle on a simulé N quadruplets indépendants $(W_i, X_i, Y_i, Z_i)_{0 \leq i < N}$ suivant cette loi.

1. Pour $\alpha, \beta, \gamma, \delta$ quatre coefficients réels, exprimer l'espérance et la variance de la variable $\alpha W + \beta X + \gamma Y + \delta Z$ en fonction de $\alpha, \beta, \gamma, \delta$ et de $\mathbb{E}(W), \mathbb{E}(X), \mathbb{E}(Y), \mathbb{E}(Z), \mathbb{E}(W^2), \mathbb{E}(X^2), \mathbb{E}(Y^2), \mathbb{E}(Z^2), \mathbb{E}(WX), \mathbb{E}(WY), \mathbb{E}(WZ), \mathbb{E}(XY), \mathbb{E}(XZ)$ et $\mathbb{E}(YZ)$.

Indication : Il est conseillé d'introduire les notations vectorielles $\vec{X} := (W, X, Y, Z)$ et $\vec{\alpha} := (\alpha, \beta, \gamma, \delta)$, et de procéder aux calculs à l'aide du formalisme matriciel.

2. Écrire un programme `calculs_preliminaires` dans lequel, en recourant à N simulations du quadruplet (X, Y, Z, T) (N étant le paramètre d'entrée de notre programme), on estime toutes les espérances listées ci-dessus.

Indication : Là encore, le formalisme matriciel pourra alléger les notations, même s'il rend les choses un peu plus obscures...

3. Écrire le développement limité au premier ordre des fonctions suivantes au voisinage de $(\bar{w}, \bar{x}, \bar{y}, \bar{z}) = (\mathbb{E}(W), \mathbb{E}(X), \mathbb{E}(Y), \mathbb{E}(Z))$:

$$\begin{aligned} f_2(w, x, y, z) &= x - w^2, \\ f_3(w, x, y, z) &= \frac{y - 3wx + 2w^3}{(x - w^2)^{3/2}}, \\ f_4(w, x, y, z) &= \frac{z - 3x^2 - 4wy + 12w^2x - 6w^4}{(x - w^2)^2}. \end{aligned}$$

Pour $(\hat{w}, \hat{x}, \hat{y}, \hat{z})$ proche de $(\bar{w}, \bar{x}, \bar{y}, \bar{z})$, en déduire un développement limité au premier ordre de $(f_2(\hat{w}, \hat{x}, \hat{y}, \hat{z}) - f_2(\bar{w}, \bar{x}, \bar{y}, \bar{z}))$, resp. des expressions analogues pour f_3 et f_4 , comme fonction linéaire de $(\hat{w} - \bar{w}, \hat{x} - \bar{x}, \hat{y} - \bar{y}, \hat{z} - \bar{z})$ — les coefficients de ce développement étant autorisés à être des fonctions de $(\hat{w}, \hat{x}, \hat{y}, \hat{z})$.

4. Notant $\hat{w}, \hat{x}, \hat{y}, \hat{z}$ les estimateurs de Monte-Carlo respectifs de $\mathbb{E}(W), \mathbb{E}(X), \mathbb{E}(Y), \mathbb{E}(Z)$ obtenus à partir de N simulations du quadruplet (W, X, Y, Z) , en déduire les intervalles de confiance lorsqu'on cherche à estimer $f_2(\bar{w}, \bar{x}, \bar{y}, \bar{z})$ à partir de $f_2(\hat{w}, \hat{x}, \hat{y}, \hat{z})$, resp. les estimations analogues pour f_3 et f_4 .

5. Application : si X est une variable aléatoire de classe L^8 qu'on sait simuler, écrire un programme calculant, par la méthode de Monte-Carlo, les estimations et intervalles de confiance des quantités suivantes :

$$\begin{aligned} \text{Var}(X) & \qquad \qquad \qquad (\text{variance}); \\ \frac{\mathbb{E}((X - \mathbb{E}(X))^3)}{\text{Var}(X)^{3/2}} & \qquad \qquad \qquad (\text{coefficient d'asymétrie}); \\ \frac{\mathbb{E}((X - \mathbb{E}(X))^4)}{\text{Var}(X)^2} - 3 & \qquad \qquad \qquad (\text{kurtosis normalisé}). \end{aligned}$$

EXERCICE 10 — Fluctuations d'un estimateur

Cet exercice concerne un champ d'application essentiel de la méthode de Monte-Carlo, à savoir la détermination d'intervalles de confiance en statistique dans des contextes où les lois exactes sont trop compliquées à calculer. Nous ferons en l'occurrence cette application dans le cadre d'un exemple précis, mais les idées évoquées sont très générales et marchent dans n'importe quel cadre.

On suppose qu'on est en train d'observer des réalisations indépendantes successives X_1, X_2, \dots, X_N de la loi P_θ uniforme sur $[0, \theta]$, où le paramètre θ est inconnu ; et que notre but est de fournir une estimation ainsi qu'un intervalle de confiance pour θ à partir de la grandeur $X_* := \max_{1 \leq i \leq n} X_i$.

1. (★) Justifier que X_* est un estimateur convergent de θ .

Dans la suite de l'exercice, nous admettrons (ce qui sera bien le cas ici) que la loi \mathbb{P}_θ change de façon suffisamment régulière lorsque θ varie pour qu'on puisse assimiler les éléments du comportement de \mathbb{P}_θ qui nous intéresseront aux éléments correspondants du comportement de $\mathbb{P}_{\hat{\theta}}$, où $\hat{\theta}$ est un estimateur convergent de θ .

2. Écrire un programme `fluctuations` qui prend en entrée les valeurs de θ et N , ainsi qu'un nombre de simulations N_{sim} au choix de l'utilisateur, et affiche en sortie un histogramme (estimé) de la loi de $(X_* - \theta)$, cet histogramme étant calculé par une méthode de Monte-Carlo.

Indication : La fonction `histogram` de MATLAB permet de tracer des histogrammes.

3. Écrire une fonction `fluctuations_compact`, variante du programme `fluctuations` qui, plutôt que d'afficher l'historgramme de la variable aléatoire $(X_* - \theta)$, renvoie (une estimation de) les quantiles à 5 % et 95 % de sa loi, ainsi que son espérance.

4. À l'aide de la fonction `fluctuations_compact`, écrire un programme `estimateur` qui, à partir de la donnée de X_* et θ , calcule un estimateur $\hat{\theta}$ essentiellement sans biais de θ ^[†], ainsi qu'un intervalle de confiance à 90 % pour θ , le risque étant réparti à 5 % de chaque côté de l'intervalle.

EXERCICE 11 — Une petite astuce

Le programme ci-dessous est destiné à estimer par la méthode de Monte-Carlo l'espérance d'une variable aléatoire que l'on sait simuler par la fonction externe `simulation`, ainsi qu'à calculer un intervalle de confiance à 95 % pour cette estimation. Son code est le suivant :

```
function MonteCarlo (N)
    Qsomme = zeros (1, 4);
    Qsomme2 = zeros (1, 4);
    calculer_variance = true;
    for i = 1 : (N / 400)
        for j = 1 : 100
            for k = 1 : 4
                x = simulation ();
                Qsomme (k) = Qsomme (k) + x;
                Qsomme2 (k) = Qsomme2 (k) + x * x;
            end
        end
        Qvariance_estimee = Qsomme2 / (100 * i) - (Qsomme / (100 * i)) .^ 2;
```

[†]. Par « essentiellement sans biais », j'entends que l'estimateur $\hat{\theta}$ se comporte asymptotiquement comme un estimateur sans biais (quand N tend vers l'infini, uniformément en la valeur de N_{sim}), c.-à-d. que l'espérance de la variable aléatoire $(\hat{\theta} - \theta)$ est négligeable devant les valeurs typiques que cette variable prend.

```
varmax = max (Qvariance_estimee);
varmin = min (Qvariance_estimee);
if (varmax / varmin) < 1.001
    calculer_variance = false;
    break;
end
end
somme = sum (Qsomme);
somme2 = sum (Qsomme2);
if (calculer_variance)
    for j = (400 * i + 1) : N
        x = simulation ();
        somme = somme + x;
        somme2 = somme2 + x * x;
    end
    N_variance = N;
else
    N_variance = 400 * i
    for j = (400 * i + 1) : N
        x = simulation ();
        somme = somme + x;
    end
end
estimateur = somme / N;
variance_estimee = somme2 / N_variance - (somme / N_variance) .^ 2;
variance_estimateur = variance_estimee / N;
fprintf ('L'esperance de la loi est estimee a %f, ');
fprintf ('avec une marge d'erreur a 95 %% de ');
fprintf ('+/- %f\n', estimateur, 1.96 * sqrt (variance_estimateur));
end
```

Réduction de la variance

EXERCICE 12 — Grandes déviations d'un tirage à pile ou face

Dans cet exercice, on considère 100 lancers successifs d'une pièce de monnaie équilibrée ; la probabilité correspondant à cette situation aléatoire est notée \mathbb{P} . On note X la variable aléatoire correspondant au nombre de « pile » obtenus ; et $f := (X - 60)_+^{[*]}$. Le but de cet exercice est d'estimer $\mathbb{E}(f)$ aussi précisément que possible.

1. Écrire un algorithme “naïf” de Monte-Carlo pour estimer $\mathbb{E}(f)$, appelé `PF60_naif`. Cette fonction prendra en argument le nombre de simulations demandées ; affichera la valeur estimée ainsi que sa marge d'erreur à 95 % de confiance ; et ne renverra rien.

On peut imaginer la même expérience, mais où la pièce de monnaie serait pipée et aurait 60 % de chances de tomber sur « pile » à chaque fois. La probabilité correspondant à cette situation est notée \mathbb{P}' .

2. Justifier qu'il est légitime de considérer que les probabilités \mathbb{P} et \mathbb{P}' portent sur un même espace mesurable Ω ; proposer ce que pourrait être cet Ω ainsi la façon d'interpréter ses éventualités^[†].

Indication : Pour la suite de l'exercice, le compositeur de la question a en tête qu'on propose un choix de Ω qui soit naturel, et qui constitue en un ensemble fini de grand cardinal.

3. Quelle est la densité de \mathbb{P} par rapport à \mathbb{P}' ?

Indication : La réponse est $(5/6)^X (5/4)^{100-X}$; encore faut-il le justifier...

4. En déduire un algorithme d'échantillonnage préférentiel `PF60_pref` pour le calcul de $\mathbb{E}(f)$. (Cet algorithme utilisera la même interface que `PF60_naif` ; seule la structure interne sera différente).

EXERCICE 13 — La grande suite (I)

Nous considérons dans cet exercice la situation d'un joueur de yahtzee^[‡] qui cherche à obtenir une grande suite, c.à.d. la séquence des chiffres de '2' à '6'^[§]. Notant A cet évènement et p sa probabilité, p peut être estimée par la méthode de Monte-Carlo à l'aide du programme “naïf” suivant :

```
% "Nsim" est le nombre de simulations demandees.
function GS_naif (Nsim)
succes = 0;
% Boucle de Monte-Carlo.
```

[*]. Pour $a \in \mathbf{R}$, a_+ désigne la partie positive de a , c.à.d. $a_+ := \max(a, 0)$.

[†]. C.à.d. de dire à quels phénomène correspondrait chaque $\omega \in \Omega$.

[‡]. Le yahtzee est un jeu très connu, qui se joue avec 5 dés. Après avoir lancés les dés une première fois, le joueur choisit les dés qu'il souhaite relancer (cela peut éventuellement être aucun, ou tous) et les relance ; puis recommence une fois cette étape de relancer partiel (en changeant, s'il le veut, le choix des dés à relancer). Le but est d'obtenir une certaine combinaison sur les chiffres affichés par les dés à l'issue de ce protocole. (Les dés sont interchangeables : savoir quel dé est tombé sur quel chiffre n'importe pas pour la combinaison).

[§]. À noter que cette définition de la « grande suite » n'est pas celle de la variante standard du yahtzee (pour laquelle la séquence des chiffres de '1' à '5' compte aussi).

```

for i = 1: Nsim
    % "lesdes" est ce qu'affichent les des.
    lesdes = ceil (6 * rand (1, 5));
    % Première étape de relancer.
    dejavu = zeros (1, 5);
    for j = 1: 5
        if lesdes(j) == 1 || dejavu(lesdes(j) - 1)
            lesdes(j) = ceil (6 * rand ());
        else
            dejavu(lesdes(j) - 1) = true;
        end
    end
    % Seconde étape de relancer.
    dejavu = zeros (1, 5);
    for j = 1: 5
        if lesdes(j) == 1 || dejavu(lesdes(j) - 1)
            lesdes(j) = ceil (6 * rand ());
        else
            dejavu(lesdes(j) - 1) = true;
        end
    end
    % L'objectif est-il atteint ?
    dejavu = zeros (1, 6);
    for j = 1:5
        dejavu(lesdes(j)) = true;
    end
    if (all (dejavu(2: 6)))
        succes = succes + 1;
    end
end
p = succes / Nsim;
fprintf ('Probabilite de succes : ')
fprintf ('%f +/- %f.\n', p, 1.96 * sqrt (p * (1 - p) / Nsim));
return
end

```

1. Expliquer à quelle stratégie du joueur correspond le code ci-dessus. (Nous admettrons que cette stratégie est effectivement la meilleure possible).

Indication : La stratégie en question n'a rien que de très intuitif; la question est plutôt de comprendre comment fonctionne le programme et en quoi il suit bien la stratégie naturelle...

2. Notons $n_i^{(k)}$ le nombre de fois que le chiffre ' i ' apparaît parmi les dés à l'issue de k étapes de relancers [¶]. Montrer que, conditionnellement à tout ce qui s'est passé jusqu'à l'issue de la première relance, la probabilité que le joueur atteigne finalement son objectif est égale à

$$\frac{(5 - Z_1)!}{6^{5-Z_1}},$$

où

$$Z_1 := \sum_{i=2}^6 \mathbf{1}_{n_i^{(1)} \geq 1}.$$

3. En déduire un algorithme `GS_cond` (ayant la même interface que `GS_naif`) évaluant la probabilité de succès à l'aide d'une technique de conditionnement.

EXERCICE 14 — Histoire grecque

[¶]. Avec cette notation, le joueur atteint son objectif lorsque $n_i^{(2)} = \mathbf{1}_{2 \leq i \leq 6} \quad \forall i \in \{1, \dots, 6\}$.

On considère un actif financier dont le cours vaut 0 € au jour 0. Chaque jour, le cours de l'actif financier augmente ou diminue indépendamment, d'une quantité aléatoire $\mathcal{N}(-0,2;1)$ (exprimée en €). On note \mathbb{P}_0 la loi de probabilité correspondant à cette situation. On appelle p_0 la probabilité, sous \mathbb{P}_0 , qu'il existe au moins un jour $j \in \{1, \dots, 5\}$ pour lequel le cours de l'actif dépasse 2 €.

1. Écrire un programme `grecque_base` pour évaluer p_0 par la méthode de Monte-Carlo [||].

La question qu'on se pose est maintenant de savoir comment changerait la valeur p_0 si on s'était trompé dans notre modélisation, et qu'en fait, la loi que suit la variation quotidienne de notre actif financier était $\mathcal{N}(-0,18;1,1)$ [**]. On note \mathbb{P}_1 la loi associée à cette nouvelle situation aléatoire, et p_1 la probabilité correspondante que l'actif dépasse 2 €.

2. Comment réaliser un couplage naturel entre \mathbb{P}_0 et \mathbb{P}_1 [††] ?

3. À l'aide du couplage trouvé à la question précédente, écrire un algorithme `grecque_coup1` [††] qui évalue la différence $p_1 - p_0$ par la méthode de Monte-Carlo avec technique de couplage.

4. Comparer l'efficacité à celle qu'on aurait obtenue en l'absence de couplage.

EXERCICE 15 — La grande suite (II)

On reprend le paradigme de la grande suite vu dans l'exercice 13; mais cette fois-ci, c'est une technique d'échantillonnage préférentiel qu'on va essayer d'utiliser pour améliorer l'efficacité. Notre nouvelle loi d'échantillonnage est définie en considérant que les dés sont pipés, et ce de la façon suivante : le premier dé a (à chaque lancer) 1/3 chance de tomber sur '2' et 2/15 chance de tomber sur n'importe quel autre chiffre ; le deuxième dé a 1/3 chance de tomber sur '3' et 2/15 chance de tomber sur n'importe quel autre chiffre ; etc. On appelle \mathbb{P}' la probabilité correspondant à cette situation de dés pipés.

1. Modifier le code présenté à l'exercice 13 pour que :
 - Le tirage des dés ait lieu selon la loi \mathbb{P}' ;
 - Outre l'obtention ou non d'une grande suite, on calcule aussi la densité de probabilité $d\mathbb{P}/d\mathbb{P}'$;
 - Mais c'est toujours $p := \mathbb{P}(A)$ qu'on évalue (en utilisant donc la technique d'échantillonnage préférentiel).

Indication : Attention ! La stratégie du joueur dans ce nouveau code doit-elle tenir compte du fait que les dés sont pipés, et s'adapter en conséquence ; ou le joueur doit-il continuer à jouer comme s'il ignorait que les dés sont pipés ?...

2. (★) Exécutez le code obtenu à la question précédente. Normalement vous constaterez que le résultat est mauvais, voire incompatible avec le précédent... Les soupçons se portent naturellement un problème de sous-échantillonnage, et c'est effectivement le cas. D'où vient ce sous-échantillonnage, à votre avis... ?

[||]. La fonction `grecque_base` prendra en argument le nombre de simulations demandées, affichera la valeur estimée de p_0 ainsi que la marge d'erreur (au risque 5 %), et ne renverra rien.

[**]. Attention : le second paramètre d'une loi normale exprime sa *variance*, pas son écart-type... !

[††]. Rappelons qu'un tel « couplage » consiste à construire une situation aléatoire dans laquelle chaque éventualité fournit deux trajectoires $(X_i^{(0)})_{i \in \{1, \dots, 5\}}$ et $(X_i^{(1)})_{i \in \{1, \dots, 5\}}$, qui soient telles que $(X_i^{(0)})_i$ suive la loi \mathbb{P}_0 et que $(X_i^{(1)})_i$ suive la loi \mathbb{P}_1 , et que les trajectoires de $X^{(0)}$ et $X^{(1)}$ "se ressemblent" dans la mesure du possible.

[††]. Dont l'interface sera essentiellement la même que pour `grecque_base`.

Pour éviter le problème de sous-échantillonnage, on considère un nouvel évènement de succès, noté B , défini par « Le premier dé finit sur '2' (conformément à son pipage), le seconde dé finit sur '3', etc. ». On note q la probabilité de cet évènement sous \mathbb{P} .

3. Justifier sommairement qu'on devrait avoir $q = p / 120$.

4. En déduire un algorithme d'échantillonnage préférentiel pour évaluer p , mais qui ne présente pas le problème de sous-échantillonnage.

Indication : Notre algorithme calculera de prime abord un estimateur de q ; mais puisqu'on sait que $p = 120q$, on peut en déduire un estimateur de p . Attention! Il faudra penser à tenir compte de la multiplication par 120 pour transporter la marge d'erreur de l'estimateur de q vers l'estimateur de p ...

Indication : Attention! Le joueur doit-il tenir compte du fait que le succès consistera en fait à obtenir l'évènement B , et donc adapter sa stratégie en conséquence; ou devra-t-il continuer à jouer comme s'il cherchait à obtenir l'évènement A ?...

5. (★) Exécuter cet algorithme. Normalement ça ne marche toujours pas; et même, c'est pire encore : l'algorithme converge clairement vers une valeur erronée...! Sauriez-vous expliquer ce qui s'est passé; et comment y remédier?

Indication : En fait, le problème est qu'avec la stratégie suivie par notre joueur, on n'a pas $q = p / 120$...! Un élément essentiel manque en effet à la stratégie pour que cette identité soit valide : lequel...? Rajoutez cet élément pour réparer l'erreur!

EXERCICE 16 — Le retour de la finance

On revient dans cet exercice sur le contexte de l'exercice 14. Notre objectif est d'utiliser la technique de conditionnement pour améliorer l'estimation de p_0 . Pour ce faire, nous considérons la sous-tribu engendrée par le couple (X_2, X_4) , notée \mathcal{A} .

1. Justifier que, conditionnellement à la tribu \mathcal{A} , les v.a. X_1, X_3, X_5 sont indépendantes, et suivent respectivement les lois

$$\mathcal{N}(X_2 / 2, 1/2), \quad \mathcal{N}((X_2 + X_4) / 2, 1/2), \quad \mathcal{N}(X_4, 1).$$

2. Calculer la probabilité que, conditionnellement à la tribu \mathcal{A} , le cours de l'actif dépasse 2 €.

Indication : Dans cette question, on considère que nous avons à notre disposition la fonction d'erreur erf^[*].

3. En déduire un algorithme de Monte-Carlo utilisant la technique de conditionnement pour évaluer p_0 plus finement.

EXERCICE 17 — Épidémie

On considère une population de 625 personnes, représentées par un carré de dimensions 25×25 (qu'on représentera dans MATLAB sous la forme d'une matrice), parmi lesquelles une épidémie est susceptible de se propager. Les règles de propagation sont les suivantes^[†] :

— Initialement, seule la personne centrale de coordonnées $(13, 13)$ (que nous appellerons patient zéro) est exposée (on représentera cet état par le nombre 1); les autres sont pures (état représenté par le nombre 0).

[*]. Qui est implémentée (sous ce nom) dans MATLAB.

[†]. On peut démontrer que l'ordre dans lequel on effectue les différentes opérations importe peu, pourvu qu'on aille jusqu'au bout de la simulation en s'assurant qu'il ne reste plus aucune personne exposée à la fin.

- Lorsqu'une personne est exposée, on tire une variable aléatoire de Bernoulli de paramètre 0,55 pour savoir ce qu'elle devient. Si la variable aléatoire tirée est positive, la personne devient infectée (état représenté par le nombre 2); si elle est nulle, la personne devient immunisée (état représenté par le nombre -1).
- Lorsqu'une personne est infectée, chacune de ses quatre voisines (ou trois, resp. deux, si on est sur le bord du carré, resp. dans un coin), si elle était pure, devient exposée.

1. (★) Simuler la propagation de l'épidémie.

Indication : Une façon de simuler rapidement l'épidémie est de maintenir à jour la liste des personnes exposées, et de traiter^[‡] les personnes *exposées* dans l'ordre de cette liste. Techniquement, on utilisera trois variables : une matrice `exposees` de taille 2×625 destinée à contenir les coordonnées des personnes exposées (initialisée de sorte que sa première colonne soit $(13, 13)^T$); et deux indices indiquant quelle sont les colonnes de `exposees` correspondant à la première, resp. à la dernière personne exposée encore à traiter (initialisés tous les deux à 1). Chaque fois qu'on tire la variable de Bernoulli d'une personne exposée, on incrémente le premier compteur, puis si la personne s'avère infectée, on rend exposés ses voisins encore purs; et chaque fois qu'une nouvelle personne devient exposée, on incrémente le second compteur et on met à jour la colonne correspondante de la matrice `exposees`.

2. Déterminer par la méthode de Monte-Carlo le nombre moyen de personnes qui seront infectées par l'épidémie (en incluant le patient zéro le cas échéant).

On s'intéresse maintenant à la possibilité de vacciner une personne prise au hasard. Cela signifie que cette personne démarrera la simulation avec le statut immunisée au lieu de pure (ou, si cette personne se trouve être le patient zéro, que l'épidémie ne démarrera jamais). La question est de savoir « combien d'infections cela évite-t-il, en moyenne ? ».

3. Utiliser une technique de couplage pour répondre à la question posée.

4. Comparer le nombre moyen d'infections évitées avec la probabilité (dans la situation initiale) qu'une personne prise au hasard finisse infectée. Expliquer pourquoi les épidémiologistes affirment que le choix d'être vacciné ou non n'est pas une question purement individuelle, mais engage aussi autrui.

EXERCICE 17bis — Les risques du levier

Un boursicoteur projette un investissement très risqué : il emprunterait une forte somme d'argent à sa banque pour l'investir dans deux entreprises, et rembourserait ensuite sa banque avec le bénéfice généré par ses placements. Plus exactement, on note S_{banque} la somme qu'il compte emprunter à sa banque, et r la taux d'intérêt exigé par sa banque sur la période concernée. Notre boursicoteur compte investir la somme d'argent S_{invest} entre les deux entreprises A et B , à raison d'une proportion π_A investie dans l'entreprise A , et $\pi_B = 1 - \pi_A$ investie dans l'entreprise B . Les cours actuels des actions des entreprises A et B sont respectivement notés X_A et X_B . Au terme de son prêt, on peut modéliser la valeur actualisée des cours par les variables aléatoires X'_A et X'_B , avec

$$(\log X'_A, \log X'_B) = (\log X_A, \log X_B) + (\mu_A, \mu_B) + \mathcal{N} \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{AA} & \sigma_{AB} \\ \sigma_{AB} & \sigma_{BB} \end{pmatrix} \right)^T.$$

Le risque que craint le plus notre boursicoteur est de se retrouver en défaut de paiement s'il ne peut pas rembourser sa banque. Notez toutefois qu'en cas de mauvaise conjoncture financière, il aura aussi la ressource de puiser dans sa cassette personnelle, d'un montant S_{perso} .

[‡]. Par « traiter » une personne exposée, je n'entends pas la *soigner* (cela n'aurait d'ailleurs pas de sens ici...), mais déterminer si elle finit par basculer dans l'état infectée (rendant alors exposées ses voisines encore purs) ou dans l'état immunisée.

On donne les valeurs suivantes pour les paramètres : $S_{invest} = S_{banque} = 100\,000\text{ €}$, $S_{perso} = 30\,000\text{ €}$, $\pi_A = 50\%$, $\mu_A = 11\%$, $\mu_B = 8\%$, $\sigma_{AA} = 0,25$, $\sigma_{BB} = 0,06$, $\sigma_{AB} = 0,06$, $r = 5\%$.

1. Justifier que, si \vec{X} est un n -vecteur gaussien standard et \mathbf{P} une matrice de taille $m \times n$, alors $\mathbf{P}\vec{X}$ est un m -vecteur gaussien centrée de matrice de covariance $\mathbf{P}\mathbf{P}^\top$. En déduire comment, à l'aide de la fonction `chol` de MATLAB, on peut simuler un vecteur gaussien d'espérance et de covariance (supposée non dégénérée) données.

2. Écrire un programme estimant la probabilité de défaut par la méthode de Monte-Carlo. (nom : `defaut_simple` entrée : nombre de simulations demandé ; affiche : probabilité de défaut calculée et son intervalle de confiance au risque 5 % ; sortie : rien).

3. Justifier que si \vec{X} est un vecteur gaussien de paramètres $(\vec{\mu}, \Sigma)$ et \vec{X}^* un vecteur gaussien centré de paramètres $(\vec{\mu}^*, \Sigma)$, alors la densité relative de la loi de \vec{X}^* par rapport à la loi de \vec{X} , au point \vec{x} , est égale à

$$\exp((\vec{\mu}^* - \vec{\mu})^\top \Sigma^{-1} (\vec{x} - (\vec{\mu} + \vec{\mu}^*)/2)).$$

4. En déduire une méthode d'échantillonnage préférentiel pour estimer la probabilité de défaut. (nom : `defaut_pref`, même interface que `defaut_simple`). On prendra comme loi d'échantillonnage préférentiel une loi où X'_A et X'_B se comportent comme si on avait $(\mu_A, \mu_B) = (-43\%, -15\%)$.

Notre boursicotier se dit que le risque pris est peut-être un peu trop élevé... Il envisage maintenant de prendre plutôt $\pi_A = 45\%$ (et donc $\pi_B = 55\%$), tous les autres paramètres restant inchangés. Il se demande de combien cela diminuerait pour lui le risque de défaut (en valeur absolue).

5. Écrire une fonction `diffdefaut` utilisant la méthode de couplage pour estimer le risque évité par cette nouvelle allocation des ressources. (même interface que `defaut_simple`).

6. (★) Au fait, et si la question consistait à évaluer le risque *relatif* évité par la nouvelle allocation?... Écrire le programme `diffreldefaut` correspondant.

7. Combiner les méthodes d'échantillonnage préférentiel et de couplage pour estimer plus finement le risque évité par la nouvelle allocation des ressources, dans un programme `diffdefaut_pref`. (même interface que pour `defaut_simple`).

Simulation de processus aléatoires

EXERCICE 18 — Processus stable

Un processus stable (normalisé) d'exposant $\alpha \in (0, 2)$ est un processus à valeurs dans \mathbf{R} , markovien, évoluant par sauts, défini de la façon suivante : pendant l'intervalle de temps dt , la probabilité que survienne un saut d'amplitude dans $[x, x + dx]$ vaut (indépendamment de tout ce qui s'est passé avant) $|x|^{-1-\alpha} dx dt$, pour tout $x \in \mathbf{R}^{[*]}$. (On suppose par ailleurs le processus issu de 0 au temps 0). Ici nous allons considérer une approximation de ce processus, où on ne retient que les sauts d'amplitude (en valeur absolue) supérieure à ε , où ε est une certaine valeur de troncature. Dans cet exercice, nous prendrons $\alpha = 1,5$ et $\varepsilon = 10^{-2}$ — dans les codes qu'on écrira, il est demandé de traiter ces paramètres comme des macros^[†], afin de pouvoir les modifier facilement si on le voulait.

1. Justifier que les instants auxquels se produisent les sauts (indépendamment de leur intensité) forment un « processus ponctuel de Poisson » d'intensité constante, c.à.d. que la probabilité qu'un saut se produise dans un intervalle de temps $[t, t+dt]$ vaut (indépendamment de tout ce qui s'est passé avant) λdt , où λ , appelée l'« intensité » du processus ponctuel de Poisson, vaut ici

$$\lambda := 1\,333,333.$$

(On calculera l'expression littérale de λ dans le cas général).

2. Écrire une fonction `Poisson_ponct` qui représente les instants des sauts. Cette fonction prendra en argument un temps T , ne renverra rien, et affichera l'ensemble des instants de saut, en traçant un point sur l'axe des abscisses à chaque instant correspondant. (On testera la fonction pour $T = 0,05$ p. ex.).

3. Justifier que, conditionnellement au fait qu'il se produise un saut à un instant donné, la loi de l'amplitude de ce saut est (indépendamment de tout ce qui s'est passé avant cet instant) une loi à densité sur \mathbf{R} notée μ , définie par :

$$d\mu(x) := \lambda^{-1} \mathbf{1}_{|x| \geq \varepsilon} |x|^{-1-\alpha} dx.$$

4. Écrire un programme `mu` qui simule la loi μ . Ce programme ne prendra aucun argument et retournera la valeur simulée.

5. Écrire une fonction `proc_stable` qui trace la trajectoire du processus stable. Cette fonction prendra en argument un temps T , ne renverra rien, et affichera la trajectoire du processus sur $[0, T]$. Je vous demande de tracer cette trajectoire, non pas en traçant la courbe exacte tenant compte de *chaque* saut, mais en calculant les valeurs de la courbe pour les temps qT avec $q \in \{0, 1/1\,024, \dots, 1\,023/1\,024, 1\}$.

EXERCICE 19 — Mouvement brownien avec rappel

[*]. En toute rigueur, pour un tel processus dont la densité de sauts est infinie, cette définition laisserait encore deux paramètres implicites non fixés, mais nous ne nous en préoccupons pas ici, dans la mesure où la troncature évite ce problème.

[†]. Sous MATLAB, cela se fera par le biais de variables globales définies une unique fois au début du code.

1. Écrire une fonction MATLAB d'arguments X_0 et T qui trace sur $[0, T]$ l'évolution de l'équation différentielle stochastique suivante :

$$dX_t = \sigma dW_t - kF \operatorname{sgn}(X_t) |X_t|^{k-1} dt, \quad (*)$$

où les paramètres sont $\{k = 6; F = 1; \sigma = 1\}$, qu'on traitera comme autant de macros. On utilisera la méthode d'Euler avec un découpage en 2 048 pas de temps homogène, ce paramètre de programmation étant lui aussi traité comme une macro. Tester le programme pour $X_0 = 0$ et $T = 16$.

2. (★) À quelles équations plus communes correspond l'équation (*) :

1. Lorsque $F = 0$?
2. Lorsque $\sigma = 0$?
3. Lorsque $k = 2$? ^[‡]
4. Lorsque $k = \infty$? (asymptotiquement s'entend ?). ^[§]

3. On considère maintenant la valeur du paramètre $\sigma = 10$. Tester à nouveau le programme, pour les mêmes valeurs du paramètre. Comparer ce qui se passe si on prend 16 384 pas de temps. Remarquer que les tracés présentent des comportements complètement différents en fonction du nombre de pas de temps. Pourquoi n'est-ce pas « normal » ? Pour quel choix de discrétisation le comportement est-il cohérent avec celui attendu pour la véritable équation différentielle ? En raison de quoi, selon vous, l'autre choix ne marche-t-il pas ?

EXERCICE 20 — Volatilité variable

On considère une processus $(V_t)_{t \geq 0}$ (appelé processus d'Ornstein-Uhlenbeck géométrique défini par l'équation différentielle stochastique suivante :

$$dV_t = \sigma V_t dW_t + \left(\frac{1}{2} \sigma^2 - \lambda \log \frac{V_t}{V^*} \right) V_t dt,$$

où les paramètres sont $V^* = 1$, $\sigma^2 = 2$ et $\lambda = 1$.

1. Simuler le processus décrit ci-dessus sur 6 unités de temps, par la méthode d'Euler stochastique, pour la condition initiale $V_0 = 0,5$. On prendra un pas de temps de 2×10^{-3} .

On considère maintenant un processus $(X_t)_{t \geq 0}$ couplé au processus V_t . Le processus X évolue purement par sauts, mais la loi de densité des sauts dépend de V : la probabilité que, pendant un intervalle de temps infinitésimal $[t, t + dt]$, X fasse un saut d'amplitude dans $[x, x + dx]$ vaut

$$\mathbf{1}_{x \geq \varepsilon} \frac{\alpha V_t}{2} |x|^{-\alpha-1} dx dt,$$

où les paramètres sont $\alpha = 2$ et $\varepsilon = 10^{-2}$.

2. Montrer que la densité (totale) de probabilité de saut par unité de temps à un instant donné est égale à $\varepsilon^{-\alpha} V_t$.

3. Montrer que, conditionnellement au fait qu'il y ait un saut à un moment donné, indépendamment de la valeur de V_t , l'amplitude du saut suit une loi qui peut être simulée

[‡]. La réponse est hors-programme.

[§]. Question particulièrement difficile.

par $\varepsilon SU^{-1/\alpha}$, où S est un signe uniforme sur $\{-1, 1\}$ et U une variable uniforme sur $[0, 1]$, indépendante de S .

4. (★) Simuler l'évolution du couple (V_t, X_t) , avec la condition initiale $X_0 = 0$. (On affichera les deux trajectoires sur des graphiques différents, car les échelles ne sont pas les mêmes).

EXERCICE 21 — Mouvement brownien sur la sphère

Le but de cet exercice est de simuler une évolution aléatoire d'un point sur la sphère unité de \mathbf{R}^3 . J'affirme qu'une telle évolution peut être décrite par l'équation différentielle stochastique suivante, pour un paramètre α à choisir judicieusement :

$$d\vec{X}_t = \mathbf{P}(\vec{X}_t) \cdot d\vec{W}_t - \alpha \vec{X}_t dt,$$

où \vec{W}_t est un mouvement brownien standard tridimensionnel et

$$\mathbf{P}(\vec{X}) := \mathbf{I}_3 - \|\vec{X}\|_2^{-2} \begin{pmatrix} X_1^2 & X_1 X_2 & X_1 X_3 \\ X_1 X_2 & X_2^2 & X_2 X_3 \\ X_1 X_3 & X_2 X_3 & X_3^2 \end{pmatrix}.$$

(\mathbf{I}_3 est la matrice identité de dimension 3 ; X_1, X_2, X_3 sont les coordonnées de \vec{X} ; et $\|\vec{X}\|_2$ est sa norme euclidienne).

1. Supposons dans cette question qu'on puisse dans les calculs traiter \vec{W}_t comme un processus de classe \mathcal{C}^1 . Montrer qu'alors, pour $\alpha = 0$, $\|\vec{X}_t\|_2$ devrait être conservée au cours de l'évolution du processus.

Indication : Les calculs sont assez fastidieux si on y procède trop naïvement... Il vaut mieux vérifier que c'est la quantité $\|\vec{X}_t\|_2^2$ qui est conservée, car cela correspond à une fonction plus facile à différentier. D'autre part, on remarque qu'en termes vectoriels, on a $\|\vec{X}\|_2 = (\vec{X}^\top \vec{X})^{1/2}$ et $\mathbf{P}(\vec{X}) = \mathbf{I}_3 - \|\vec{X}\|_2^{-2} \vec{X} \vec{X}^\top$, ce qui permet de grandement simplifier les calculs.

2. Simuler le processus pour $\alpha = 0$ depuis la condition initiale $\vec{X}_0 = (1, 0, 0)$ sur 2 unités de temps avec un pas de discrétisation de taille 1/10 000. Même question pour $\alpha = 1$ et pour $\alpha = 2$. Dans quel cas la simulation vous semble-t-elle bien rester sur la sphère unité ?

Indication : On peut réaliser un tracé tridimensionnel à l'aide de la commande `plot3` de MATLAB. Plus simplement, vous pouvez aussi tracer $\|\vec{X}_t\|_2$ en fonction du temps.

3. (★) En utilisant la formule d'Itô (hors-programme), justifier le phénomène observé à la question précédente, et vérifier que la bonne valeur de α est bien celle trouvée.

EXERCICE 21bis — De l'importance de l'ordre

On considère le système d'équations différentielles stochastiques couplées suivant :

$$\begin{cases} dX_t = Y_t dW_t^1 + dW_t^2; \\ dY_t = X_t dW_t^3 + dW_t^4, \end{cases}$$

où $(W_t^1, W_t^3)_{t \geq 0}$ est un mouvement brownien bidimensionnel de variance par unité de temps

$$\begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix},$$

et $(W_t^2)_{t \geq 0}$ et $(W_t^4)_{t \geq 0}$ sont les mouvements browniens standard indépendants de (W^1, W^3) et indépendants entre eux. Lorsque nous voulons simuler un tel processus par le schéma d'Euler, une ambiguïté se pose : puisque la façon dont X évolue entre t et $t + dt$ dépend de la valeur de Y , faut-il utiliser la valeur de Y au début ou à la fin du pas de temps concerné ?... On envisagera donc trois variantes possibles pour simuler cette évolution :

- Une variante '01' dans laquelle on met à jour X puis Y : c.-à-d. que X_t évolue selon la valeur de Y au début du pas de temps, mais que Y_t évolue selon la valeur de X à la fin du pas de temps.
- Une variante '10' dans laquelle on met à jour Y puis X : c.-à-d. que Y_t évolue selon la valeur de X au début du pas de temps, mais que X_t évolue selon la valeur de Y à la fin du pas de temps.
- Une variante '00' dans laquelle on met à jour X et Y "simultanément" : c.-à-d. que X_t évolue selon la valeur de Y au début du pas de temps et Y_t évolue aussi selon la valeur de X au début du pas de temps.

1. Laquelle de ces trois variantes correspond à celle enseignée dans le cours ?

2. Écrire trois programmes `simulation_01`, `simulation_10` et `simulation_00` implémentant respectivement chacune des trois idées listées ci-dessus pour la simulation du processus. Ces programmes utiliseront l'interface suivante : ils prendront en argument les valeurs X_0 et Y_0 que doivent prendre X et Y au début de la simulation, ainsi que la durée T de la simulation, et traceront sur le même graphique les courbes de l'évolution de X_t et Y_t pour t entre 0 et T (et ne renverront rien). On prendra un pas de discrétisation de 2^{-12} .

3. Tester les trois programmes pour $(X_0, Y_0) = (2, 2)$ et $T = 1$. Au jugé, avez-vous l'impression d'une différence significative de comportement selon le cas ?

4. Pour affiner la réponse à la question précédente, on se propose d'estimer, selon le mode de simulation retenu, la probabilité que X_1 soit négatif (cette estimation se faisant bien entendu par la méthode de Monte-Carlo). Procéder à ces estimations (on prendra par exemple 2 000 simulations pour chaque cas), en précisant les intervalles de confiance à 99 %. En conclure que les trois méthodes n'aboutissent pas à la simulation des mêmes processus.

5. (★) En vous appuyant sur l'idée de développements limités formels expliquée en cours, dire quelles sont normalement les ÉDS *réellement* simulées pour les deux mauvaises méthodes. Vérifier par quelques tests que les résultats des mauvaises méthodes correspondent bien à ceux de ces ÉDS "corrigées".

Méthode de Monte-Carlo appliquée aux processus

EXERCICE 22 — Processus stable, suite

Cet exercice se place dans la continuité de l'exercice 18 vu précédemment, qu'on suppose avoir déjà été traité. Ainsi, on reprend ici les notations de cet exercice.

1. Évaluer, par la méthode de Monte-Carlo, la probabilité que le processus stable ne dépasse pas (en valeur absolue) le seuil $y_1 := 1$ avant l'instant $T_1 := 4$.

On considère dorénavant un autre processus (qu'on différenciera du précédent à l'aide du symbole $\tilde{\cdot}$), qui évolue également poissonniennement par sauts, la densité des sauts valant également λ , mais où la loi des sauts est maintenant la loi $\tilde{\mu}$ définie par :

$$d\tilde{\mu}(x) := Z^{-1} \mathbf{1}_{|x| \geq \varepsilon} e^{-(x/y_1)^2} |x|^{-1-\alpha},$$

où Z est la constante qui fait de $\tilde{\mu}$ une mesure de probabilité. Nous admettrons qu'il est possible de calculer numériquement la valeur de Z avec une grande précision, et qu'on alors celle-ci égale (pour les paramètres décrits ci-dessus) à

$$Z = 1\,328,899.$$

2. Écrire un programme `mutilde` simulant $\tilde{\mu}$.

Indication : Penser à la méthode de rejet, avec pour mesure de référence μ .

3. (★) Quelle est la densité de μ par rapport à $\tilde{\mu}$? En déduire comment calculer la densité de probabilité, pour une trajectoire donnée, de la loi du processus non tildé par rapport à la loi du processus tildé.

Indication : On calculera la densité relative comme un produit qu'on maintiendra au cours de la boucle de calcul de chaque simulation; où à chaque saut d'intensité x , on multipliera la densité relative par $d\mu/d\tilde{\mu}(x)$.

4. Appliquer une méthode d'échantillonnage préférentiel pour améliorer la précision de l'estimation précédente.

EXERCICE 23 — Intermittences

Une usine produit des boulons tout au cours de la journée, de 08 h 00 =: t_{deb} à 18 h 00 =: t_{fin} . En fonctionnement normal, elle produit 100 000 =: D boulons par heure. Toutefois, il faut exactement 1 h =: T_{pre} de préchauffage du système avant que la production puisse démarrer (y compris le matin à t_{deb}). En outre, l'approvisionnement électrique de l'usine est vétuste : l'électricité tombe en panne (en moyenne) une fois toutes les 3 h =: τ_{pan} , avec des pannes qui durent de l'ordre de 1 h =: τ_{rep} . Plus précisément, on modélise les pannes électriques ainsi :

— À un instant donné (donc en particulier le matin à t_{deb}), l'électricité a $\tau_{\text{rep}}/(\tau_{\text{pan}} + \tau_{\text{rep}})$ risque d'être en panne^[*].

[*]. Cela vient de ce qu'on pourrait démontrer que, lorsque cela fait suffisamment longtemps que le processus « réseau électrique » a démarré, la probabilité que l'électricité soit en panne à un instant donné converge vers $\tau_{\text{rep}} / (\tau_{\text{pan}} + \tau_{\text{rep}})$.

- Quand l'électricité est en panne, il faut attendre une loi exponentielle d'espérance τ_{rep} avant qu'elle soit réparée.
- Quand l'électricité fonctionne, elle tombe en panne au bout d'une loi exponentielle d'espérance τ_{pan} .

1. Simuler l'évolution de l'état du réseau électrique au cours de la journée. On présentera le résultat sous la forme d'un graphique, où « en panne » sera représenté par -2 et « en service » par -1 . (On présentera la réponse à cette question sous la forme d'un programme `elec_graph`, qui ne prendra aucun argument et ne renverra aucune valeur, et affichera le graphique souhaité). Vérifier la cohérence des résultats obtenus.

2. Modifier le programme précédent (en un programme `elec_usine_graph`) pour faire en sorte de tracer aussi sur la graphique l'évolution de l'état de l'usine au cours de la journée. (On représentera l'état « à l'arrêt » par 0, l'état « en préchauffage » par 1, et « en fonctionnement » par 2). Vérifier la cohérence des résultats obtenus.

3. Justifier que le processus « état du réseau électrique » est markovien, mais que le processus « état de l'usine » n'est *pas* markovien. Justifier qu'en revanche, si l'état de l'usine au cours du préchauffage n'est plus représenté par une constante, mais par le temps écoulé depuis le début du préchauffage^[†], alors le processus « (état du réseau électrique, état de l'usine) » est markovien, et même le processus « état de l'usine » tout seul.

4. Modifier la programme précédent (en un programme `elec_usine_graph_prod`) pour qu'il ait maintenant une valeur de sortie, correspondant au nombre de boulons total produit sur la journée^[‡]. Vérifier la cohérence des résultats obtenus.

5. Évaluer par la méthode de Monte-Carlo la production journalière moyenne de l'usine. Le programme, appelé `MC_prod`, prendra en argument le nombre de simulations demandées, ne renverra rien, et affichera une estimation de la production journalière moyenne, assortie d'un intervalle de confiance.

6. Démontrer que, à 17 h 00 (ou plus généralement à l'heure $t_{\text{fin}} - T_{\text{pre}}$), le nombre moyen de boulons que l'usine continuera de produire d'ici la fin de la journée vaut :

- 0 si l'usine est à l'arrêt ;
- $D\tau_{\text{pan}}(1 - e^{-T_{\text{pre}}/\tau_{\text{pan}}})$ si l'usine est en fonctionnement ;
- $D\tau_{\text{pan}}e^{-T_{\text{pre}}/\tau_{\text{pan}}}(e^{t/\tau_{\text{pan}}} - 1)$ si cela fait un temps t que le préchauffage a commencé.

7. En déduire une méthode de conditionnement pour le calcul du nombre de boulons moyens produit, implémentée dans un programme `MC_prod_cond`. Vérifier la cohérence des résultats obtenus. L'amélioration de l'efficacité est-elle intéressante ? Pourrait-on s'en douter ?

Exaspérée par les pannes à répétition de l'électricité, sur lesquelles elle n'a pas de pouvoir, la directrice de l'usine envisage de changer la chaîne de production : le rendement en fonctionnement serait le même, mais le préchauffage du système ne durerait plus qu'une demi-heure (valeur notée T'_{pre}). Comme cette nouvelle chaîne serait onéreuse à mettre en place, la directrice cherche à savoir quelle amélioration en résulterait sur la production journalière moyenne.

8. Estimer l'amélioration apportée, en utilisant une méthode de couplage. (Programme `MC_prod_coupl`). Vérifier la cohérence des résultats obtenus.

9. (★) Employer le conditionnement *en plus* du couplage pour améliorer l'efficacité du résultat. Vérifier la cohérence des résultats obtenus.

[†]. Mais n'utilisez pas cette représentation du préchauffage dans vos réponses.

[‡]. On ne se préoccupera pas ici du fait que les formules donnent des nombres de boulons non entiers.

EXERCICE 24 — File d'attente

Le but de cet exercice est de simuler un phénomène de file d'attente, par exemple à un péage sur l'autoroute, sous une hypothèse d'arrivée poissonnienne des voitures et de traitement poissonnien de leurs requêtes^[§].

Ce qu'on va simuler est l'évolution au cours du temps (noté t , compté en minutes) du nombre V_t de voitures attendant au péage. Ce nombre de voitures évolue markovienement, avec les règles suivantes. Au cours d'un intervalle de temps dt :

- S'il y a au moins une voiture qui attend, la probabilité qu'une voiture passe (et donc que V_t diminue d'une unité) est αdt , avec $\alpha = 4 \text{ min}^{-1}$;
- La probabilité qu'une nouvelle voiture arrive (et donc que V_t augmente d'une unité) est βdt , où β est la densité du trafic.

1. Sachant la valeur de V_t à un instant donné, quelle est la loi du temps d'attente avant que V_t évolue, et la loi du saut que V_t fera à cet instant ?

2. Écrire un programme simulant $(V_t)_{0 \leq t \leq T}$ (en partant de $V_0 = 0$), les arguments du programme étant T et β . Lancer les simulations pour $T = 720$ et respectivement $\beta = 0,5\alpha$, $\beta = 0,95\alpha$ et $\beta = 1,05\alpha$. Commenter.

3. Pour une simulation donnée, exprimer le temps d'attente moyen (sur cette simulation) τ des automobilistes en fonction de $(V_t)_t$. (Si $V_T > 0$, on fera comme si les voitures attendant encore au temps T étaient alors traitées instantanément). Modifier le programme pour qu'il retourne également cette valeur.

Indication : La réponse est que le temps d'attente moyen est l'intégrale de V , divisée par le nombre total de sauts vers le haut. Voyez-vous pourquoi ?...

4. (★) Expliquer pourquoi est-ce que, lorsqu'on réalise plusieurs simulations, le temps moyen d'attente des automobilistes sur l'ensemble des simulations n'est *pas* la moyenne des temps d'attente moyens sur chaque simulation.

5. Montrer que le nombre moyen d'automobilistes arrivant au cours d'une journée de durée T vaut exactement αT .

6. (★) Déterminer le temps d'attente moyen d'un automobiliste par la méthode de Monte-Carlo.

Indication : Le temps d'attente moyen des automobilistes n'est pas l'espérance du temps d'attente moyen par journée, mais on peut toutefois le déterminer à partir de deux autres espérances... Lesquelles ?

EXERCICE 25 — Le théorème de Girsanov

Le théorème de Girsanov énonce (entre autres) la chose suivante : si $(X_t)_{t \geq 0}$ est un mouvement brownien (issu de 0) de variance par unité de temps σ^2 soumis à une dérive $b(t)$ dépendant du temps, c.à.d. une solution de

$$dX_t = \sigma dW_t + b(t)dt,$$

alors la loi P_b des trajectoires de X sur $[0, T]$ est à densité par rapport à la loi P_0 des trajectoires du même mouvement brownien sans dérive, avec

$$\frac{dP_b}{dP_0}((f_t)_{0 \leq t \leq T}) = \exp\left(\sigma^{-2} \int_0^T (b(t)f'_t - \frac{1}{2}b(t)^2) dt\right). \quad (1)$$

[§]. Cette seconde hypothèse est évidemment peu réaliste concernant le cas d'un péage, mais ce n'est qu'un exercice...

(En général f ne sera pas dérivable, mais on pourra quand même donner un sens à (1) en interprétant « $\int_0^T b(t) f'_t dt$ » comme $\int_0^T b(t) df_t$).

1. (★) En interprétant la formule définissant X en termes de schéma d'Euler, “démontrer” le théorème de Girsanov.

2. Notons Q la loi de la trajectoire sur $[0, 1]$ d'un mouvement brownien standard avec dérive $b(t) = \mathbf{1}_{t < 1/2} \times 3/2 + \mathbf{1}_{t > 1/2} \times (-3/2)$. Exprimer la densité de Q par rapport à la loi P de la trajectoire du mouvement brownien sans dérive.

3. On note A l'événement « la trajectoire passe au-dessus de $1/2$, puis redescend en dessous de 0, tout cela avant l'instant $t = 1$ ». Comment peut-on évaluer $P(A)$ à l'aide de simulations de la loi Q ? Implémenter cette technique.

4. Comparer les résultats obtenus avec ceux de la méthode de Monte-Carlo “naïve” (càd. en échantillonnant selon P).

EXERCICE 26 — Option d'échange

Le but de cet exercice est de valoriser une option d'échange européenne par la méthode de Monte-Carlo. On considère deux actifs financiers appelés respectivement A et B , et on note A_t le cours de l'actif A au temps t (compté en années), resp. B_t le cours de B .

On suppose que le couple (A_t, B_t) évolue selon l'équation différentielle stochastique suivante :

$$\begin{cases} dA_t = A_t \times (\sigma_{A1} dW_t^{(1)} + \sigma_{A2} dW_t^{(2)} + \tau / (1 + A_t / A^*) \times dW_t^{(3)} + \beta_A dt); \\ dB_t = B_t \times (\sigma_{B1} dW_t^{(1)} + \sigma_{B2} dW_t^{(2)} + \beta_B dt), \end{cases}$$

avec

$$\begin{pmatrix} \sigma_{A1} & \sigma_{B1} \\ \sigma_{A2} & \sigma_{B2} \end{pmatrix} = \begin{pmatrix} 0,10 & 0,10 \\ 0,10 & -0,05 \end{pmatrix} \text{ an}^{-1/2}, \quad \begin{pmatrix} \beta_A \\ \beta_B \end{pmatrix} = \begin{pmatrix} 0,06 \\ 0,04 \end{pmatrix} \text{ an}^{-1}, \\ \tau = 0,20 \text{ an}^{-1/2}, \quad A^* = 100 \text{ €},$$

où $(W_t^{(1)})_t, (W_t^{(2)})_t, (W_t^{(3)})_t$ désignent trois mouvements browniens standard indépendants. On a en outre les conditions initiales $A_0 = 100 \text{ €}, B_0 = 120 \text{ €}$.

1. Écrire un programme qui simule l'évolution des cours des actifs A et B sur une durée de 2 an par la méthode d'Euler stochastique. (On prendra un pas de temps égal à la demi-journée, càd. 730 pas de temps par an).

2. (★) Justifier que sous la mesure de probabilité risque-neutre, (A_t, B_t) évolue selon l'équation différentielle stochastique suivante :

$$\begin{cases} dA_t = A_t \times (\sigma_{A1} d\tilde{W}_t^{(1)} + \sigma_{A2} d\tilde{W}_t^{(2)} + \tau / (1 + A_t / A^*) \times d\tilde{W}_t^{(3)} + r dt); \\ dB_t = B_t \times (\sigma_{B1} d\tilde{W}_t^{(1)} + \sigma_{B2} d\tilde{W}_t^{(2)} + r dt), \end{cases}$$

où $(\tilde{W}_t^{(1)})_t, (\tilde{W}_t^{(2)})_t, (\tilde{W}_t^{(3)})_t$ désignent trois (autres) mouvements browniens standard indépendants, et où r désigne le taux d'intérêt sans risque, ici supposé constant égal à $0,02 \text{ an}^{-1}$.

On considère maintenant une option dont la possession confère le droit suivant : à l'échéance $T = 2$ an, le détenteur peut s'il le souhaite acquérir une unité de l'actif A en

payant en échange une unité de l'actif B . La question qui va nous intéresser dans cet exercice est de savoir quel juste prix attribuer à cette option.

3. Justifier que, au temps T (ou, si vous préférez, juste avant le temps T), le juste prix de l'option est $(A_T - B_T)^+$.

4. Estimer l'espérance de $(A_T - B_T)^+$ sous la mesure de probabilité risque-neutre. (On pourra par curiosité comparer avec la valeur de cette espérance sous la vraie mesure de probabilité...). En déduire le juste prix de l'option (au temps initial).

EXERCICE 26bis — Processus de naissance & mort

Sur une petite île de l'Océan Pacifique vit une espèce de suidés endémique : le jambonnax grouikans. Suite à la découverte de l'île par l'homme, l'écosystème de l'île a été modifié par l'introduction involontaire d'espèces allogènes, de sorte que les écologistes s'alarment aujourd'hui d'un risque de disparition de l'espèce. Mais à quel point l'espèce est-elle réellement en danger ? Pour tenter répondre à cette question, les biologistes ont mis au point un modèle simple que nous allons étudier.

Dans ce modèle, le nombre de femelles jambonnax grouikans à l'instant t est noté X_t , et le nombre de mâles par Y_t . Le couple (X_t, Y_t) ne peut évoluer que par sauts de $(\pm 1, 0)$ ou $(0, \pm 1)$, c'est-à-dire que les naissances et morts d'individus ne peuvent avoir lieu qu'individuellement, et jamais simultanément. En outre, on modélise l'évolution du couple (X_t, Y_t) par un processus markovien [¶]. La natalité de l'espèce est telle que de nouveaux individus naissent au taux $\min\{\alpha_X X_t, \alpha_Y Y_t\}$, pour certains paramètres α_X et α_Y ; en outre, chaque naissance donne lieu indépendamment à un mâle ou une femelle avec probabilité $1/2$. Le taux de mortalité pour chaque individu, quant à lui, est de $\beta_0 + (X_t + Y_t)\beta_1$ chez les femelles, resp. de $\beta_0 + (X_t + Y_t)\beta_1 + (Y_t - 1)\beta_Y$ chez les mâles. On donne $\alpha_X = 0,5 \text{ an}^{-1}$, $\alpha_Y = 1 \text{ an}^{-1}$, $\beta_0 = 0,1 \text{ an}^{-1}$, $\beta_1 = 5 \cdot 10^{-3} \text{ an}^{-1}$, $\beta_Y = 5 \cdot 10^{-3} \text{ an}^{-1}$. En outre, un recensement de la population a montré qu'à la date actuelle ($t = 2016 \text{ an}$), il y avait 40 femelles et 30 mâles.

1. Écrire une fonction `simulation` qui simule l'évolution, à partir d'aujourd'hui, du nombre de mâles et de femelles de l'espèce. Cette fonction prendra en argument le nombre d'années sur lequel doit s'effectuer la simulation, et affichera un graphique (elle ne renverra rien) portant, en abscisse, la date (exprimée en années), et en ordonnées, le nombre d'individus. Trois courbes seront tracées, avec la même échelle : nombre de femelles (en magenta), nombre de mâles (en cyan), en nombre total d'individus (en noir).

2. Écrire une fonction `proba_ES` prenant en arguments (dans cet ordre) le nombre actuel de femelles et de mâles, une échelle de temps T considérée et un nombre de simulations N_{sim} , qui utilise la méthode de Monte-Carlo avec N_{sim} simulations pour calculer et afficher (elle ne renverra rien) les probabilités respectives que l'espèce s'éteigne ou survive à échelle de T années (en précisant les marges d'erreur à 5 %).

3. (★) Considérons la loi d'échantillonnage préférentiel où la population évolue selon la même dynamique, à ceci près que β_0 a été remplacé par une autre valeur β'_0 . Calculer la densité relative entre la vraie loi et la loi d'échantillonnage préférentiel.

Indication : Une façon de procéder qui s'avère plus pratique pour les calculs est de voir le processus comme une chaîne de Markov, en considérant des instants de temps espacés infinitésimalement.

[¶]. C'est une simplification assez grossière, car cela signifie qu'on ne tient pas compte de l'âge des suidés : ils sont sexuellement matures dès la naissance, ne peuvent pas mourir de vieillesse, et la grossesse est instantanée... !

4. Écrire une fonction `proba_ES_pref` prenant, outre les arguments T et N_{sim} , le paramètre β'_0 à utiliser pour la loi d'échantillonnage préférentiel, dans laquelle on estime le taux de survie pour l'espèce en utilisant la méthode d'échantillonnage préférentiel.

5. Tester la fonction `proba_ES_pref` avec une population initiale de 2 individus de chaque sexe, sur une échelle de temps de 10 ans, en prenant $\beta'_0 = 0,15$. Que remarque-t-on quant aux intervalles de confiance respectifs de la probabilité d'extinction et de survie ? Commenter.

Problèmes de synthèse

EXERCICE 27 — Mouvement brownien fractionnaire

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

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

$$\text{Cov}(H_{t_1}, H_{t_2}) = \frac{1}{2}(t_1^{7/4} + t_2^{7/4} - |t_2 - t_1|^{7/4}).$$

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.

EXERCICE 28 — Comment brille le Soleil ?

La question qui nous intéresse dans ce problème est de savoir combien de lumière la surface du Soleil émet en fonction de la direction. Pour ce faire, on considère une petite portion du Soleil proche de sa surface ; à cette échelle, on peut considérer que le Soleil occupe le demi-espace $\{z < 0\}$ de notre espace euclidien tridimensionnel usuel $\{(x, y, z) \in \mathbf{R}^3\}$.

Un modèle^[*] pour l'émission de la lumière par le Soleil est le suivant :

[*]. Ce modèle est assez simpliste, mais qualitativement correct.

1. Un photon^[†] est créé loin de la surface du Soleil. Il part alors en ligne droite dans une direction aléatoire uniformément distribuée.
2. Au bout d'une distance aléatoire dont la loi est exponentielle, si le photon est encore à l'intérieur du Soleil, il est absorbé par le plasma dont le Soleil est constitué, puis réémis de l'endroit même où il a été absorbé, mais dans une nouvelle direction indépendante de sa direction incidente, laquelle est à nouveau uniformément distribuée. On répète cette étape jusqu'à ce qu'éventuellement le photon traverse la surface du Soleil.
3. Une fois que le photon est sorti du Soleil, il continue sa course en ligne droite dans le vide indéfiniment.

Dans un premier temps, notre objectif est de simuler l'émission des photons par le Soleil. L'échelle des distances est supposée prise de sorte que la loi exponentielle qui intervient dans le modèle est de paramètre 1. On supposera pour simplifier (car cela ne change guère les résultats) que tous les photons sont émis depuis le point $(0, 0, -8)$.

1. Comment simuler une direction uniforme de l'espace, c'est-à-dire un point uniformément distribué sur la sphère unité ?

Indication : On pourra se rappeler qu'une certaine loi gaussienne est invariante par rotation...

2. Écrire une fonction MATLAB qui simule l'émission de N photons par le Soleil (on testera la fonction pour une dizaine de N tout au plus). Cette fonction prendra N en argument, dessinera un *plot* tridimensionnel de la trajectoire des N photons (on fera également apparaître sur le graphique la surface du Soleil, et on dessinera la longueur de trajectoire des photons une fois sortis du Soleil sur 32 unités de longueur), d'autre part les angles (exprimés en degrés) que font les photons par rapport à la surface au moment où ils émergent. Si, au bout de 3 000 simulations, un des photons simulés n'est toujours pas sorti du Soleil, on s'autorisera à laisser tomber la trajectoire de celui-ci et à tirer à la place un nouveau photon depuis le point de départ.

Indication : La fonction `plot3` permet de tracer des courbes tridimensionnelles linéaires par morceaux : vous obtiendrez l'aide en ligne sur cette fonction par la commande `doc plot3`.

Indication : La commande `hold on` permet de tracer plusieurs *plots* sur la même figure.

Indication : En ce qui concerne le tracé de la surface du Soleil, recopiez simplement le code suivant :

```
Xsurface = (-32:2:32) * ones(1, 33);
Ysurface = ones(33, 1) * (-32:2:32);
Zsurface = zeros(33);
mesh(Xsurface, Ysurface, Zsurface);
```

3. Écrire une fonction MATLAB qui utilise la méthode de Monte-Carlo pour évaluer la proportion de photons qui sortent avec un angle d'émergence compris entre 0° et 5° par rapport à la surface, entre 5° et 10° , ..., et entre 85° et 90° .

On imagine maintenant que trois physiciens tentent de calculer théoriquement les proportions que nous venons de simuler. À cause de diverses fautes de calcul, ils sont arrivés à trois conclusions différentes... Ainsi, selon le premier physicien, la probabilité qu'un rayon sorte du Soleil avec l'angle θ est proportionnelle à $\sin \theta d\theta$; selon la seconde, elle est proportionnelle à $\sin(2\theta) d\theta$; et selon la troisième, à $\sin \theta \sin(2\theta) d\theta$.

4. Au vu des simulations effectuées, qui a raison ?

5. Comment feriez-vous pour donner un sens quantifiable à la façon dont l'expérience numérique confirme ou infirme le calcul de tel ou tel physicien ? Développez.

[†]. C'est-à-dire un « grain » de lumière

EXERCICE 29 — Dudo

Le but de cet exercice est de répondre à la question : « Quand on lance 25 dés, quelle est la probabilité, notée p , qu'un des chiffres au moins apparaisse 7 fois ou plus ? ».^[‡] (Pour ceux qui ne souhaitent pas utiliser de paramètres explicites dans leurs formules, je suggère de noter $25 =: Z$ le nombre de dés, $6 =: F$ le nombre de faces d'un dé, et $7 =: D$ le nombre de dés identiques qui intervient dans la définition de p).

1. Écrire une fonction appelée `simulation` qui simule le lancer de 25 dés, et renvoie 1 si l'un des chiffres apparaît 7 fois ou plus, et 0 sinon.

Indication : Ceux qui travaillent sous MATLAB pourront regarder avec profit la documentation des commandes suivantes : `eq`, `max`, `randi`, `sum`.

2. À l'aide de la fonction `simulation` écrite à la question précédente, écrire un programme appelé `MC_p` pour évaluer la probabilité recherchée à l'aide de la méthode de Monte-Carlo, en précisant l'intervalle de confiance.

Dans la suite du problème, on appelle *chiffre record* le chiffre qui apparaît en plus grand nombre au tirage des 25 dés, avec la règle suivante : quand le plus grand nombre d'occurrences est partagé par plusieurs chiffres, on tire au hasard uniformément parmi ces chiffres l'un d'eux qui sera décrété être le *chiffre record*. On note q la probabilité de l'évènement « Le chiffre record est '1', et celui-ci apparaît au moins 7 fois ».

3. Exprimer p en fonction de q , en justifiant le résultat. Si l'intervalle de confiance pour q (à une certaine précision) est $\hat{q} \pm r$, quel est l'intervalle de confiance correspondant pour p à la même précision ?

Dans la suite du problème, nous n'allons pas chercher à évaluer directement p , mais plutôt q , étant donné que d'après la question précédente cela donne du même coup un estimateur pour p .

4. Écrire une méthode de Monte-Carlo "naïve" pour évaluer q . (le programme sera appelé `Pb1_MC_q`).

5. (★) À quelle technique de réduction de variance présentée dans le cours vous fait penser le lien entre p et q ? Pourquoi l'idée d'évaluer q plutôt que p est-elle à priori tout-à-fait stupide ? À votre avis, pourquoi avons-nous retenu cette idée malgré tout ?

Pour un lancer des 25 dés, on note A le nombre de '1' qui apparaissent.

6. Calculer exactement $\mathbb{E}(A)$, $\mathbb{E}(A^2)$, et [facultatif] $\mathbb{E}(A^3)$.

7. (★) Écrire une méthode de Monte-Carlo utilisant les variables de contrôle A et A^2 (et éventuellement A^3) pour évaluer q . (le programme sera appelé `Pb1_MC_q_ctrl`).

On définit maintenant la probabilité Q d'un tirage de 25 dés où les dés sont pipés : pour chaque dé, le chiffre '1' a 25 % de chances de sortir, tandis que les autres chiffres n'ont que 15 % de chances chacun. (Ceux qui ne souhaitent pas écrire les paramètres pourront poser $0,15 =: (1 - \varepsilon)/6$, resp. $0,25 =: (1 + 5\varepsilon)/6$).

8. Pour un tirage $\omega = (\omega_1, \dots, \omega_{25})$ des dés (les ω_i étant les chiffres affichés par les différents dés), exprimer la densité relative $(dP/dQ)(\omega)$ — où P désigne la loi du tirage de dés non pipés.

[‡]. La motivation pour calculer p est librement inspirée d'un jeu de dés chilien appelé *dudo* — mais il est inutile de connaître ce jeu pour aborder le problème.

9. En déduire une technique d'échantillonnage préférentiel pour évaluer q , et l'implémenter. (le programme sera appelé `Pb1_MC_q_pref`).

EXERCICE 30 — Croissance de fissures

Un ingénieur s'intéresse au risque d'éclatement d'un tube de générateur de vapeur dans une centrale nucléaire. Nous faisons ici la modélisation grossière suivante : il y a dans le tube une unique fissure dont la taille initiale est nulle ; et au cours du fonctionnement du générateur de vapeur, la fissure croît selon le processus markovien suivant :

- La longueur de la fissure croît par des sauts brutaux, entre lesquels elle reste constante ;
- Les sauts se produisent de façon poissonnienne, c'est-à-dire qu'à tout instant, indépendamment du passé, il y a la même probabilité qu'un saut se produise, avec un taux de $4,5 \cdot 10^{-4}$ par heure de fonctionnement (taux noté τ) que ce saut se produise ;
- Quand un saut se produit, l'amplitude de ce saut (c'est-à-dire l'accroissement de la longueur de la fissure) suit une loi exponentielle de longueur caractéristique $0,4 \text{ mm} =: \lambda$, c'est-à-dire que la loi de ce saut X est telle que $\mathbb{P}(X \geq x) = e^{-x/\lambda}$.

Un tube est destiné à être utilisé pendant un cycle de fonctionnement de durée $10\,000 \text{ h} =: T$, et éclate si la longueur de la fissure dépasse $10 \text{ mm} =: L^*$. On cherche à évaluer la probabilité p qu'un tube éclate avant la fin du cycle.

1. Écrire une fonction (qu'on appellera `Pb2_evolution`) qui simule l'évolution de la longueur $\ell(t)$ de la fissure au cours du temps sur la durée T , plotte la courbe correspondante, et renvoie la longueur finale de la fissure (éventuellement supérieure à L^* , auquel cas le tube aura en réalité éclaté avant).

Indication : On pourra utiliser la fonction `stairs` de MATLAB pour plotter une fonction évoluant par sauts.

2. Pourquoi sera-t-il pratiquement indispensable d'utiliser une technique de réduction de variance pour ce problème ?

3. Quelle est la loi du nombre de fois que la fissure s'allonge au cours du cycle de fonctionnement ? Coder une fonction appelée `probanbsauts` qui calcule *exactement*, pour $n \geq 1$, la probabilité que la fissure subisse au moins n allongements au cours du cycle.

Indication : Histoire que cette question ne risque pas de faire "barrage", je précise que la loi du nombre de fois que la fissure s'allonge appartient à une famille de lois classiques, ce qui vous devrait vous permettre de l'identifier facilement même si vous ne parvenez pas à le justifier.

4. Supposons qu'on simule l'allongement final de la fissure ainsi : *d'abord*, on simule la liste (infinie) $\delta_1, \delta_2, \dots$ des longueurs des allongements successifs de la fissure — sachant que tous ces accroissements n'auront pas forcément lieu effectivement — ; et *ensuite*, on simule le nombre d'allongements qui se sont produits au cours du cycle de fonctionnement pour en déduire la longueur totale atteinte. Conditionnellement à la première partie de la simulation, quelle est la probabilité que le tube éclate ?

5. En déduire une technique de conditionnement pour évaluer la probabilité d'éclatement d'un tube, et implémenter celle-ci (le programme s'appellera `MC_cond`). Pourquoi n'est-ce en fait pas un problème que, dans la première partie de simulation, on doive théoriquement simuler une suite infinie de valeurs (ce qui serait évidemment impossible) ? [Facultatif] À quelle technique de réduction de variance peut se rattacher l'argument qui dit qu'on n'a en fait pas à simuler une suite infinie de valeurs ?

Indication : Dans la mesure où la fonction `pbanbsauts` sera appelée très régulièrement par le programme `MC_cond`, on pourra gagner un temps considérable en tabulant à l'avance les valeurs de retour les plus fréquentes de `pbanbsauts`.

Un industriel propose à EDF d'utiliser un acier plus cher mais un peu plus résistant pour fabriquer les tubes, acier pour lequel le taux des sauts est de $\tau' = 4 \cdot 10^{-4}$ par heure de fonctionnement, avec une longueur caractéristique $\lambda' = 0,38$ mm pour les sauts (toujours supposés exponentiels). Les tubes éclatent toujours pour une longueur de fissure de 10 mm. EDF cherche à savoir si ce nouvel acier vaut le coup en évaluant avec précision la diminution de probabilité d'éclatement qu'il induit.

6. Toujours en utilisant la technique de conditionnement de la question précédente, utiliser la méthode des simulations communes pour évaluer la différence entre les probabilités d'éclatement selon qu'on utilise le premier ou le second acier (le programme s'appellera `MC_cond_CRN`).

EXERCICE 31 — Provision du casino

Ce problème concerne un casino qui se demande quel montant il doit provisionner dans ses coffres pour une soirée. On modélise ainsi le déroulement d'une soirée dans le casino : au début de la soirée, le casino a préparé dans ses coffres un certain montant en euros, appelé provision. Au cours de la soirée, de nombreuses parties de jeux de hasard sont jouées. On supposera ici que le seul jeu de ce casino est la roulette : dans une partie de roulette, le joueur mise une certaine somme d'argent et parie sur un numéro ; avec une chance sur 37, le numéro parié par le joueur sort et celui-ci récupère alors 36 fois sa mise ; dans le cas contraire, le casino garde le mise du joueur et ne lui rend rien. (C'est le déséquilibre entre le nombre 37 de paris possibles et le facteur 36 du gain qui permet au casino de faire un bénéfice sur le long terme).

On suppose pour simplifier que les parties sont jouées de façon poissonnienne, autrement dit qu'à un instant donné le fait qu'une partie soit jouée ou pas (ainsi que le montant misé et le résultat de la partie) est totalement indépendant de ce qui se passe aux autres instants. Nous supposons qu'il se joue en moyenne 10 parties par minute au cours de la soirée, la soirée durant 8 heures. Lorsqu'une partie est jouée, dans 50 % des cas la mise est de 5 €, qui correspond à la mise minimale autorisée ; dans 5 % des cas la mise est de 100 €, qui correspond à la mise maximale autorisée ; et dans les 45 % de cas restants, le montant misé m (exprimé en euros) suit une loi telle que $\log m$ soit uniforme sur l'intervalle $[\log 5, \log 100]$. Indication : Il pourra être judicieux d'introduire des notations littérales pour manipuler plus aisément les différentes quantités numériques introduites dans l'énoncé. Je propose que la probabilité de gain à la roulette soit notée `pgain` ; que le facteur de gain en cas de succès (càd. le ratio entre le montant récupéré et le montant misé) soit noté `fgain` ; que le nombre moyen de parties par unité de temps soit noté `lambda` ; que la durée totale de la soirée soit notée `Tsoiree` ; et que les montants minimal et maximal pour une mise soient notés respectivement `misemin` et `misemax` et que les probabilités qu'un joueur joue chacune de ces mises soient notées respectivement `pmmin` et `pmmax`. Je suggère également que les unités de base choisies soient l'euro pour les sommes d'argent, et la minute pour les durées.

1. Écrire une fonction `mise` qui simule la loi de la mise d'un joueur. Cette fonction ne prendra aucun argument et renverra la somme misée.

2. Écrire une fonction `partie` qui simule le déroulement d'une partie d'un joueur (la mise étant choisie aléatoirement comme précisé dans l'énoncé). Cette fonction ne prendra aucun argument et renverra le gain net de la partie pour le casino.

3. Écrire une fonction `attente` qui simule le temps qu'il faut attendre à un instant donné avant qu'un joueur ne joue une partie. La fonction ne prendra aucun argument et renverra le temps attendu.

4. Écrire une fonction `soiree` qui simule l'évolution de la réserve disponible dans les coffres du casino au cours d'une soirée, sans se soucier de la contrainte physique qui fait qu'en pratique, la réserve est obligée de rester positive. Cette fonction prendra en argument la provision initialement constituée par le casino et affichera une visualisation graphique de l'évolution de la réserve au cours du temps, sans retourner de valeur.

Le casino s'intéresse à la probabilité qu'un joueur fasse "sauter la banque" : on dit que la banque saute lorsque, à un instant donné, le casino se retrouve engagé à verser à un joueur une quantité d'argent dont il ne dispose pas dans ses coffres.

5. Par rapport à la simulation de la question précédente, à quoi correspond le fait que la banque saute au cours de la soirée ?

6. Écrire une fonction `sautage` qui simule l'évolution d'une soirée et nous dit si la banque a sauté. Cette fonction prendra en argument la provision initialement constituée par le casino, et renverra 1 si la banque saute et 0 sinon (vous pouvez aussi utiliser des booléens).

7. Écrire un programme `psautage` pour estimer par la méthode de Monte-Carlo la probabilité que la banque saute. Ce programme prendra en argument la provision initialement constituée par le casino et affichera la probabilité estimée que la banque saute, assortie d'un intervalle de confiance à 90 %, sans retourner de valeur.

8. Justifier que la probabilité (exacte) que la banque saute est une fonction décroissante de la provision initialement constituée par le casino.

En réalité, ce qui préoccupe le casino n'est pas uniquement le risque que la banque saute. Un autre risque est celui de braquage... Or, en cas de braquage, si les voleurs parviennent à vider la caisse, le préjudice subi par le casino sera évidemment d'autant plus sévère que la provision était élevée : il n'est donc pas forcément judicieux de constituer une provision trop importante ! Le casino estime qu'un sautage de la banque lui coûte en pratique 100 000 € (à cause de la perte de réputation que cela entraîne) ; quant au risque de braquage, il est estimé à une malchance sur 10 000 par soirée, et le cas échéant on suppose pour simplifier que le montant perdu correspond exactement à celui de la provision. Ce que la banque cherche à minimiser en réalité, c'est l'espérance du préjudice total subi.

Indication : Je suggère d'appeler `csautage` le coût d'un sautage de la banque, et `pbraquage` la probabilité de braquage pour une soirée.

9. Écrire un programme `Eprejudice` estimant l'espérance du préjudice subi (par soirée), avec un intervalle de confiance à 90 %. Ce programme prendra en argument la provision initialement constituée par le casino, et affichera l'estimation de l'espérance du préjudice subi, assortie d'un intervalle de confiance à 90 %, sans retourner de valeur.

10. Notre casino hésite entre deux montants possibles pour provisionner ses coffres, mais compte tenu de la puissance de calcul dont il dispose, le programme `Eprejudice` ne lui permet pas d'avoir des résultats suffisamment fins pour trancher sur l'identité de la meilleure option... Quelle technique de réduction de la variance peut-il utiliser pour pallier ce souci ? Implémenter cette technique dans un programme appelé `comparaison`, qui prendra en arguments les deux montants de provisions envisagés, et qui répondra si on peut affirmer que le premier montant est le meilleur choix au risque de 10 %, si on peut affirmer que le second montant est le meilleur choix au risque de 10 %, ou si on ne peut pas trancher au niveau de risque considéré (la fonction ne retournera par ailleurs aucune valeur).

Pour rendre les calculs plus rapides, on peut envisager de modéliser certains aspects du processus de l'évolution de la réserve $(R_t)_{t \in [0, T_{soirée}]}$ par une équation différentielle stochas-

tique. Par exemple, on pourrait souhaiter approximer la contribution des petites parties (parties dont les mises sont de montant mise_{\min}) sous la forme d'un incrément de mouvement brownien avec dérive, de sorte que

$$dR_t \simeq adW_t + bdt + (\text{contribution des moyennes et grosses mises}). \quad (2)$$

11. Pourquoi l'équation (2) est-elle une approximation raisonnable pour la contribution des petites parties ?

12. Calculer théoriquement les valeurs de a et b adaptées à l'approximation de la contribution des moyennes et grosses mises.

Indication : Les valeurs de a et b doivent être choisies de sorte que le mouvement brownien avec dérive ait la même moyenne et la même variance que le véritable processus ; il faut donc calculer la moyenne et la variance de la contribution des petites mises pour déterminer a et b .

EXERCICE 32 — L'or de la rivière

Une compagnie minière s'intéresse à l'achat d'une vallée dans une région réputée pour ses cours d'eau aurifères, afin de prospecter de l'or dans la rivière qui y coule. On modélisera le cours de la rivière par le segment $[0, L]$, où L est la longueur de la vallée ($L = 50$ km). Les ingénieurs géologues ont établi que dans cette région, quand on parcourt un segment, le profil de la densité d'or le long de ce segment (en masse d'or par unité de longueur) peut être modélisée de la façon suivante :

- L'or est présent sous la forme de "gisements" indépendants, dont les contributions se superposent les unes aux autres. (Un même point peut très bien appartenir à plusieurs gisements différents).
- Chaque gisement G est caractérisé par l'emplacement de son centre (z_G), la quantité d'or qu'il contient (q_G) et l'étalement spatial du gisement autour de son centre (r_G). La contribution du gisement au profil de densité d'or le long de la rivière est

$$d_G(x) = (\alpha_G - \beta_G(x - z_G)^2)_+,$$

où $d_G(x)$ désigne la densité d'or en x qui est due au gisement G , et où « \cdot_+ » est l'opération "partie positive" (càd. $a_+ := \max(a, 0)$). Les paramètres α_G et β_G sont reliés à q_G et r_G de sorte que

$$\int_{\mathbf{R}} d_G(x) dx = q_G$$

et

$$\text{supp } d_G = [z_G - r_G, z_G + r_G],$$

où 'supp' désigne le support d'une fonction (càd. [l'adhérence topologique de] l'ensemble des points où cette fonction est non nulle).

- À chaque point x , la probabilité qu'il y ait un gisement d'or centré sur ce point est indépendante, égale à λdx , où λ est un certain paramètre du modèle ($\lambda = 0,04 \text{ km}^{-1}$). En d'autres termes, l'ensemble des centres de gisements d'or est un processus ponctuel de poisson sur \mathbf{R} , d'intensité λ .
- Pour un gisement d'or donné, la loi des paramètres q_G et r_G , qui est la même indépendamment d'un gisement à l'autre, est telle que

$$d\mathbb{P}(q_G = q, r_G = r) = Z \mathbf{1}_{r < R} \frac{\exp(-q / \kappa(R - r))}{R - r} dq dr, \quad (3)$$

où R et κ sont des paramètres du modèle ($R = 25$ km, $\kappa = 1$ t · km⁻¹) ('t' est le symbole de la tonne), et où Z est la constante de renormalisation qui assure qu'on ait bien une densité de probabilité.

Indication : Dans tout ce problème, je suggère de prendre comme unités de base respectivement le kilomètre pour les distances et la tonne pour les quantités d'or.

1. Écrire une fonction `qetr` qui simule la loi décrite par (3). Ce programme ne prendra aucun argument et renverra les valeurs de q et r pour le gisement.

Indication : Pour arriver à simuler cette loi, la meilleure méthode en l'occurrence consiste à d'abord déterminer la loi (marginale) de r , puis la loi conditionnelle de q sachant la valeur de r , de façon qu'on simulera d'abord r , puis q .

Indication : Si vous ne parvenez pas à résoudre cette question, pour la suite du problème, je vous suggère de prendre q et r indépendants, avec q uniforme sur $[0, \kappa]$ et r uniforme sur $[0, R]$.

2. Déterminer α et β en fonction de q et r (j'enlève ici les indices ' \cdot_G '), et écrire une fonction `alphabeta` qui prend en argument q et r et renvoie α et β .

3. Écrire une fonction `gisement` qui prend en argument les paramètres (q_G, r_G, z_G) d'un gisement, et renvoie le profil de densité dû à ce gisement sur le cours d'eau de la rivière. Ce profil sera renvoyé sous la forme de deux vecteurs-lignes de même longueur : le vecteur des densités sur différents points de discrétisation de $[0, L]$ (on prendra un pas de discrétisation de 50 m au plus), et le vecteur des abscisses correspondantes.

On note $\bar{d}(x)$ la densité d'or en un point x de la rivière, qui est la somme des contributions $\bar{d}_G(x)$ des différents gisements.

4. Écrire une fonction `profil` qui simule le profil $(\bar{d}(x))_{x \in [0, L]}$ de densité de l'or sur le cours de la rivière et qui trace celui-ci. La fonction ne prendra pas d'argument et ne renverra rien.

Indication : Attention, il peut y avoir un effet de certains gisements dont le centre n'est pas sur le cours d'eau de la rivière, mais avant ou après celui-ci ! On remarquera toutefois qu'on dispose d'une borne a priori sur les valeurs possibles de r (laquelle ?), ce qui nous permet de déterminer un segment sur lequel il suffit de regarder les centres de gisements à prendre en compte qui sont susceptibles d'avoir un impact sur le cours d'eau.

5. Le processus $(\bar{d}(x))_{x \in [0, L]}$ est-il markovien ? Justifier.

6. Justifier que le processus $(\bar{d}(x))_{x \in [0, L]}$ est *stationnaire*, c.à.d. que la loi de (la restriction de) le profil d sur un segment $[a_0, a_1]$ est indentique à sa loi sur le segment $[a_0 + b, a_1 + b]$ pour tous a_0, a_1, b (sous réserve que ces segments soient contenus dans $[0, L]$, s'entend).

Pour savoir si la vallée vaut le coup d'être achetée, la compagnie procède à un sondage en deux points de la rivière d'abscisses $x_1 = 15$ km et $x_2 = 35$ km, qui lui permet d'y déterminer les valeurs de la densité \bar{d} . On note $\bar{\delta} := (\bar{d}(x_1) + \bar{d}(x_2))/2$. On note par ailleurs \bar{q} la quantité totale d'or dans le lit de la rivière.

7. Écrire une fonction `deltaqbarre` qui ne prend aucun argument et, après simulation d'un profil \bar{d} , renvoie les valeurs de $\bar{\delta}$ et \bar{q} associées.

8. Quelle est l'espérance de $\bar{\delta}$ conditionnellement à la valeur de \bar{q} ? Justifier. En déduire un estimateur sans biais de \bar{q} en fonction de $\bar{\delta}$.

Indication : Il y a un argument qui permet de trouver l'espérance conditionnelle de $\bar{\delta}$ sachant \bar{q} sans avoir à faire de calculs compliqués...

9. L'ingénieur mathématicien de la compagnie suggère à celle-ci de calculer la régression linéaire entre \bar{q} et $\bar{\delta}$, afin d'avoir une idée de ce que vaut l'espérance de \bar{q} conditionnellement

à $\bar{\delta}$. Autrement dit, on cherche une approximation de $\mathbb{E}(\bar{q}|\bar{\delta})$ sous la forme

$$\mathbb{E}(\bar{q}|\bar{\delta}) \approx A\bar{\delta} + B.$$

Comment l'estimateur des moindres carrés dit-il qu'il faut estimer A et B dans un tel cas ?

10. La question précédente nous amène à devoir estimer certaines espérances, disons les espérances de quantités que nous désignerons par f_1, f_2, \dots, f_k . Implémenter une méthode de Monte-Carlo pour estimer ces espérances (en précisant un intervalle de confiance). **Attention, pour cette question je vous demanderai de ne pas travailler sous la loi \mathbb{P} , mais sous la loi $\mathbb{P}(\cdot|\mathcal{E})$, où \mathcal{E} est l'évènement $\{\bar{\delta} \in [\bar{\delta}_-, \bar{\delta}_+]\}$. ($\bar{\delta}_- = 4 \text{ t} \cdot \text{km}^{-1}$, $\bar{\delta}_+ = 8 \text{ t} \cdot \text{km}^{-1}$).**

Indication : Pour simuler la loi $\mathbb{P}(\cdot|\mathcal{E})$, on procèdera simplement par rejet à partir de \mathbb{P} .

Indication : À défaut d'avoir trouvé la réponse à la question précédente, vous pouvez prendre $k = 1$ et $f_1 = \bar{q}\bar{\delta}$.

11. Implémenter une méthode d'échantillonnage préférentiel pour estimer $\mathbb{P}(\mathcal{E})$ et $\mathbb{E}(f_1 \mathbf{1}_{\mathcal{E}})$ (\mathcal{E} étant l'évènement défini à la question précédente), où la loi d'échantillonnage diffère de la loi vraie par sa valeur de γ (on pourra par exemple multiplier γ par deux). Commenter les résultats obtenus.

12. On observe le paradoxe suivant : pour les grandes valeurs \bar{q}_1 de \bar{q} , notant $\bar{\delta}_1 := \mathbb{E}(\bar{\delta}|\bar{q} = \bar{q}_1)$, on a $\mathbb{E}(\bar{q}|\bar{\delta} = \bar{\delta}_1) \ll \bar{q}_1$! Commenter : En quoi est-ce paradoxal ? Comment expliquer le phénomène ? Notre compagnie doit-elle plutôt s'intéresser à trouver un estimateur sans biais de \bar{q} ou à connaître l'espérance de \bar{q} conditionnellement à $\bar{\delta}$? Quelles seraient les conséquences possibles d'un mauvais choix ?

13. (★) À votre avis, à la question 10, pourquoi vous ai-je demandé de travailler sous la loi $\mathbb{P}(\cdot|\mathcal{E})$ plutôt que sous la loi \mathbb{P} ?

Indication : Il y a deux réponses possibles : l'une est une raison strictement mathématique ; l'autre concerne les implications pratiques de se restreindre à l'évènement \mathcal{E} pour la situation d'entreprise présentée par l'énoncé.

14. (★) À partir des résultats de la question 11, on peut obtenir un estimateur de $\mathbb{E}(f_1|\mathcal{E})$ (lequel ?). Comment calculeriez-vous un intervalle de confiance associé à cet estimateur, aussi finement que possible ? Dans le cas où on échantillonne sous la véritable loi, votre estimateur de $\mathbb{E}(f_1|\mathcal{E})$ donne-t-il (essentiellement) les mêmes résultats que l'estimateur "naïf" en échantillonnant sous $\mathbb{P}(\cdot|\mathcal{E})$, ou perd-on quelque chose à utiliser la nouvelle méthode ?

EXERCICE 33 — C'est chaud... !

Dans ce problème, on étudie l'échauffement d'un circuit électrique au cours d'un certain processus industriel. Le risque qu'on cherche à quantifier est celui de la fonte du circuit, qui aurait des conséquences graves.

On note θ_t la température du circuit à l'instant t . L'évolution de θ_t est donnée par l'équation différentielle stochastique suivante :

$$d\theta_t = \sigma dW_t - \lambda(\theta_t - \Theta_t)dt, \quad (4)$$

où W_t désigne un mouvement brownien standard, λ, σ sont des paramètres du modèle (voir le tableau 1 pour les valeurs de ces paramètres), et où $(\Theta_t)_t$ est une fonction déterministe définie par :

$$\Theta_t := \Theta_* - \alpha(t - T_2)^2/2, \quad (5)$$

Θ_*, T_2, α étant d'autres paramètres du modèle (voir encore tableau 1). On a en outre la condition initiale :

$$\theta_{t=0} \sim \mathcal{N}(\Theta_* - \alpha T_2^2 / 2 - \alpha T_2 / \lambda, \sigma^2 / 2\lambda). \quad (6)$$

1. Si la température se mesure en K et que le temps se mesure en min, déduire des équations (4) et (5) dans quelles unités se mesurent les paramètres $\Theta_*, T_2, \alpha, \lambda, \sigma$; puis vérifier l'homogénéité physique de (6).

Paramètre	Valeur
α	0,2
Δ	5
λ	0,2
σ	2,2
T_1	25
T_2	30
T_3	35
T_4	60
Θ_*	1 348
Θ_f	1 358

TABLE 1 – Valeurs des paramètres du problème 33

2. Simuler le processus $(\theta_t)_t$ sur l'intervalle de temps $[0, T_4]$, par la méthode d'Euler stochastique. (On prendra un pas de temps de 0,05). On répondra à cette question en écrivant une fonction `simul_temp` qui ne prendra pas d'argument et ne renverra pas de valeur, et affichera le graphique de θ en fonction du temps, ainsi que (sur le même graphique) la courbe constante égale à Θ_f (cf. tableau 1).

Indication : Pour tracer deux courbes sur le même graphique, il faut écrire les commandes `plot` correspondantes entre les instructions « `hold on` » et « `hold off` ».

Le circuit fond lorsque, à un moment donné entre 0 et T_4 , θ_t dépasse Θ_f .

3. Écrire une fonction `proba_fusion` estimant, par la méthode de Monte-Carlo, la probabilité que le circuit fonde, assortie d'un intervalle de confiance à 95 %. Cette fonction ne prendra aucun argument et ne renverra aucune valeur, se contentant d'afficher l'estimateur calculé et sa marge d'erreur.

4. Reprendre la question précédente avec un pas de temps de 0,01 pour la simulation, et comparer les résultats obtenus (avec, disons, 500 000 simulations pour le pas de temps originel et 200 000 simulations pour le nouveau pas^[§]). Vous devriez normalement observer que ces résultats ne sont pas compatibles... À quoi cela est-il dû, à votre avis ?

5. Expliquer le fonctionnement et l'intérêt du programme suivant :

```
function b = fusion_super ()
global % La fin de cette ligne est cachée.
initialiser_parametres (); % La fonction "initialiser_parametres"
                           % initialise les paramètres globaux.
b = false;
```

[§]. Pour un tel nombre de simulations, les calculs peuvent être très lents ; si vous ne pouvez pas monter jusqu'à un tel niveau, expliquez alors juste comment vous effectuez votre comparaison, quitte à ce que votre conclusion s'avère différente de celle prévue par l'énoncé.

```

t = 0;
temperature = temperature_initiale (); % La fonction "temperature initiale"
                                        % simule la loi de la temperature
                                        % initiale.
while t < T4
    if temperature > Tfusion
        b = true;
        break
    end
    if temperature > Tfusion - 4.
        pas = .01;
    else
        pas = .05;
    end
    temperature = temperature + % La fin de cette ligne est cachée
    t = t + pas;
end
return
end

```

On considère une nouvelle loi de probabilité \mathbb{P}' correspondant au même modèle que la probabilité \mathbb{P} qui nous intéresse, à ceci près que Θ_t y est remplacée par la fonction Θ'_t définie par :

$$\Theta'_t := \begin{cases} \Theta_t & \text{pour } t \notin [T_1, T_3), \\ \Theta_t + \Delta & \text{pour } t \in [T_1, T_3). \end{cases}$$

Nous admettons provisoirement que la loi du processus $(\theta_t)_t$ sous \mathbb{P}' est alors à densité par rapport à la loi de $(\theta_t)_t$ sous \mathbb{P} , avec

$$\frac{d\mathbb{P}'}{d\mathbb{P}} = \exp\left(\frac{\Delta\lambda}{\sigma^2}\left(\theta_{T_3} - \theta_{T_1} + \lambda \int_{T_1}^{T_3} (\theta_t - \Theta_t - \Delta/2) dt\right)\right), \quad (7)$$

dont on calculera la valeur numérique par la méthode des rectangles.

6. Améliorer la fonction `proba_fusion` en une fonction `proba_fusion_pref` utilisant une technique d'échantillonnage préférentiel, en prenant \mathbb{P}' pour loi d'échantillonnage (avec un pas de temps de 0,05).

7. Que se serait-il passé si on avait pris $\Delta = 20$? Expliquer.

8. Démontrer (informellement) la formule (7).

EXERCICE 34 — Le tournoi de tennis

16 joueurs de tennis s'affrontent dans un tournoi à élimination directe, le tableau étant composé aléatoirement. Ces joueurs sont numérotés de #1 à #16, #1 étant le joueur le plus fort et #16 le plus faible. Lorsque le joueur # n dispute un match, son niveau au cours de cet match est aléatoire (car il y a des jours avec et des jours sans), suivant la loi $\mathcal{N}(-\log n; 0,5)$ — on décrit ici le « niveau » d'un joueur par un paramètre dans \mathbf{R} , d'autant plus grand que le joueur joue bien — : de la sorte, le joueur le mieux classé est celui qui a le plus de chances de gagner un match, mais le joueur le moins bien classé a ses chances quand même. Un exemple (tiré aléatoirement) de ce que peut donner un tel modèle de tournoi est donné dans l'illustration ci-dessous — je précise que dans notre modèle, le niveau d'un même joueur d'un match à l'autre est indépendant.

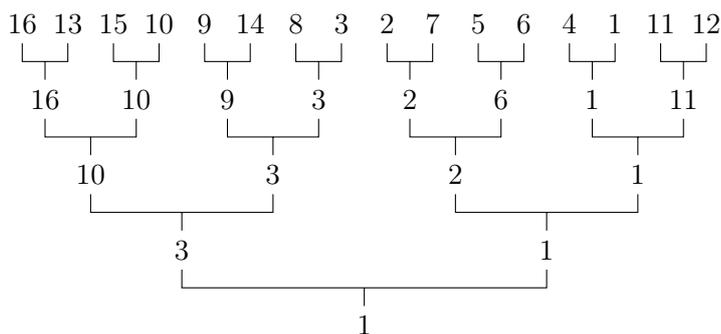


FIGURE 1 – Exemple de tournoi. Ici, il y a eu deux surprises au premier tour, où #16 a battu #13 et #6 a battu #5. Hélas pour #16 et #6, leur chance a ensuite tourné au second tour, où ils n’ont pas renouvelé leur performance (cela aurait d’ailleurs été d’autant plus difficile que leurs adversaires au second tour étaient plus redoutables qu’au premier). Plus généralement, à partir du second tour la hiérarchie a toujours été respectée, et en particulier c’est logiquement #1 qui s’est imposé. Notez que les hasards du tableau ont fait que #3 est arrivé en finale sans avoir eu à affronter d’adversaire mieux classé que lui, alors que #2 a été éliminé par #1 en demi-finale.

Un bookmaker souhaite établir la cote du joueur #16 pour ce tournoi (le tableau n’ayant pas encore été tiré); pour cela, il a besoin de déterminer la probabilité, notée p_{16} , que ce joueur gagne.

1. Implémenter un algorithme de Monte-Carlo pour évaluer p_{16} .

Indication : Une méthode particulièrement rapide pour tirer le tableau aléatoirement est l’algorithme de Fisher-Yates, dont on trouvera la description sur Wikipédia par exemple. Toutefois dans un premier temps je vous conseille de mettre au point vous-mêmes une méthode que vous compreniez bien, quitte à comparer ensuite avec ce qu’aurait donné Fisher-Yates.

Indication : Bien entendu, dès que #16 perd un de ses matchs, on sait qu’il ne gagnera pas le tournoi, de sorte qu’il n’est pas nécessaire de simuler ce qui se passe après. Au fait, à quelle technique de réduction de la variance s’apparente cette observation ?

Indication : Notez que, quitte à changer la représentation des branches du tableau, on peut toujours supposer que #16 est le joueur situé tout à gauche du tableau.

La convergence de l’algorithme étant très lente, notre bookmaker souhaite améliorer sa précision.

2. Implémenter une technique de réduction de la variance par variables antithétiques. Évaluer le gain d’efficacité.

3. Implémenter un échantillonnage préférentiel dans lequel le joueur #16 est “survolté” et se comporte toute la semaine comme s’il avait le niveau ℓ . (on pourra prendre $\ell = -\log 8$ pour l’application). Évaluer le gain d’efficacité.

4. Implémenter une technique de conditionnement où on conditionne par rapport à tous les matchs dans lesquels on est sûr que #16 ne jouera pas. Pour calculer exactement la probabilité de victoire conditionnelle de #16, on utilisera la fonction `erfinv`. Évaluer le gain d’efficacité.

Indication : La fonction `erfinv` étant relativement gourmande en calcul, on s’efforcera de suivre une méthode d’implémentation qui minimise le nombre de recours à celle-ci.

5. Implémenter une technique de stratification par rapport à l’identité du plus fort joueur dans la moitié de tableau de #16, selon qu’il s’agit de #1, #2, #3, #4 ou d’un joueur plus faible. Évaluer le gain d’efficacité.

Indication : Vous pouvez choisir, soit de stratifier “à postériori” en procédant à un échantillonnage uniforme, soit d’optimiser l’échantillonnage inter-strates. Dans ce dernier cas, vous avez besoin de stratifier “à priori”, c.à.d. de savoir tirer un tableau uniformément à l’intérieur d’une strate donnée : comment feriez-vous pour cela ?

6. Implémenter une technique de réduction de la variance par variables de contrôle en utilisant pour variables de contrôle le niveau réel auquel #16 a joué lors de chacun de ses quatre matchs (ou aurait joué s’il avait été qualifié). Évaluer le gain d’efficacité.

EXERCICE 35 — Les aiguilles de Buffon

La formule de Buffon

L'expérience des aiguilles de Buffon est une méthode empirique de calcul de π . Dans cette expérience, on se situe dans une pièce dont le sol est revêtu d'un parquet avec des lattes de largeur ℓ , et on dispose d'un grand nombre Z d'aiguilles identiques de longueur a . On lance les aiguilles depuis une hauteur beaucoup plus grande que ℓ (en pratique, il suffit des les lâcher de la hauteur de son corps). Une fois que les aiguilles sont au sol, on compte le nombre de points où une aiguille croise une séparation entre deux lattes de rangées différentes (voir la figure 2), nombre qu'on appelle N . La formule de Buffon est alors un certain estimateur de π en fonction de Z , N , a et ℓ .

Pour une aiguille i , on note $y_i \in \mathbf{R}$ la coordonnée 'y' du milieu de l'aiguille ('y' étant l'axe perpendiculaire aux lattes), et note \tilde{y}_i la distance de ce point à la latte située immédiatement en dessous, de sorte que (nous prendrons ici $y = 0$ à une frontière entre deux lattes) \tilde{y}_i est l'unique $h \in [0, \ell)$ tel que $y_i = kh + \tilde{y}_i$ avec $k \in \mathbf{N}$. D'autre part, on note $\theta_i \in [0, \pi)$ l'angle formé par l'aiguille avec l'horizontale. (voir la figure 2).

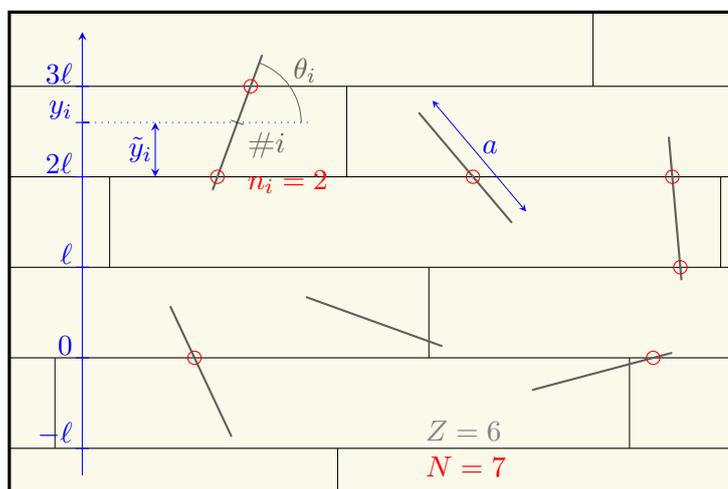


FIGURE 2 – Illustration de l'expérience de Buffon

1. À votre avis, à quoi devrait ressembler la loi de y_i ? Et celle de \tilde{y}_i ? Quelle sera selon vous la loi de θ_i ? Et celle du couple (\tilde{y}_i, θ_i) ?
2. Appelons n_i le nombre d'intersections de l'aiguille i avec les frontières entre lattes. Exprimer n_i en fonction de (\tilde{y}_i, θ_i) .
3. Calculer théoriquement $\mathbb{E}(n)$ (j'entends par là la valeur commune de $\mathbb{E}(n_i)$).
4. Expliquer en quoi on peut voir une méthode de Monte-Carlo dans la méthode des aiguilles de Buffon : quel est la quantité ν qu'on estime, et quel est son estimateur ? Dédire de l'estimateur de ν une formule pour estimer π .
5. Calculer un intervalle de confiance (dont je vous laisse choisir le niveau) pour l'estimateur de ν de la question précédente. En déduire un intervalle de confiance pour π .
Indication : On pourra considérer que la largeur de l'intervalle de confiance pour ν est suffisamment petite pour qu'on puisse linéariser les changements de variables.
6. Faire l'expérience en vrai (prendre une photo). Commenter le résultat et sa précision.

Indication : On pourra utiliser des allumettes à la place des aiguilles.

On s'intéresse maintenant à la question de simuler l'expérience de Buffon pour en obtenir une meilleure précision. Un des problèmes qui se pose est de simuler θ , ou plus exactement le couple $(\cos \theta, \sin \theta)$: en effet, on ne peut pas légitimement utiliser la connaissance de la valeur de π dans cette simulation, étant donné que notre but est précisément de calculer cette valeur ; et on ne peut pas non plus utiliser les fonctions trigonométriques, pour la même raison...

NOTA : En fait, on ne connaîtra jamais θ lui-même, mais seul le couple $(\cos \theta, \sin \theta)$ nous importe !

7. Proposer une méthode de simulation de $(\cos \theta, \sin \theta)$ satisfaisant les contraintes ci-dessus.

Indication : Si on sait simuler des variables aléatoires uniformes sur $[0, 1]$, [¶] on peut en déduire une méthode de simulation de la loi uniforme sur le disque unité par rejet sur le carré $[-1, 1] \times [0, 1]$, puis en tirer une simulation de $(\cos \theta, \sin \theta)$.

8. Implémenter une simulation de la méthode des aiguilles de Buffon avec un million d'aiguilles. (On prendra des paramètres a et ℓ à peu près égaux à ceux de l'expérience physique). Commenter le résultat obtenu ainsi que sa précision.

Réduction de la variance

9. On considère la loi du couple (\tilde{y}, θ) pour chaque aiguille. Conditionnellement à θ , quelle est la loi de \tilde{y} ? En déduire l'espérance conditionnelle de n sachant θ , qu'on notera $\bar{n}(\theta)$.

10. Grâce à la question précédente, utiliser une technique de conditionnement pour obtenir une nouvelle estimation de ν (et donc de π) par la méthode de Monte-Carlo. Implémenter cette technique (avec les mêmes paramètres qu'à l'implémentation de la question 8). Commenter la précision du résultat et la vitesse du calcul par rapport à la simulation de la question 8.

☛ Dans la suite du problème, on s'intéresse à des techniques de réduction de variance pour le problème d'évaluation de ν tel que posé lors de la question précédente, c'est-à-dire, on part de la méthode dans laquelle le conditionnement est déjà utilisé. Il s'agit donc d'évaluer ν à partir de simulations de $(\sin \theta, \cos \theta)$.

11. Que valent $\mathbb{E}(\sin^2 \theta)$, resp. $\mathbb{E}(\sin^4 \theta)$?

12. Utiliser les réponses à la question précédente pour réduire la variance dans l'évaluation de ν (donc de π) à l'aide de l'utilisation des variables de contrôle $\sin^2 \theta$ et $\sin^4 \theta$.

13. Soit θ' la valeur qui se déduit de θ par un quart de tour, c.à.d. $\theta' = \theta \pm \pi/2$ où le signe est choisi de sorte que $\theta' \in [0, \pi)$. Montrer que θ' a la même loi que θ .

14. Expliquer grossièrement pourquoi on peut s'attendre à ce que $\bar{n}(\theta)$ et $\bar{n}(\theta')$ soient négativement corrélées. En déduire une technique de réduction de variance pour l'évaluation de ν à l'aide de l'utilisation de variables antithétiques. Implémenter cette méthode et commenter.

15. Proposer une interprétation "physique" à la méthode de variables antithétiques des questions précédentes.

Le double effet *Kiss Cool*

[¶]. Les variables pseudo-aléatoires que fournissent les logiciels de calcul numérique, dans la quasi-totalité des cas, sont produites par la méthode du générateur congruentiel linéaire, méthode qui n'a absolument pas besoin de connaître la valeur de π pour simuler une loi uniforme sur $[0, 1]$: il n'y a donc pas d'escroquerie méthodologique à ce niveau.

16. Lorsque nous utilisons la méthode de rejet pour simuler θ , quelle est la loi du nombre d'essais nécessaires avant de tomber dans le disque unité ?

17. En déduire une autre méthode de Monte-Carlo pour évaluer π à partir des mêmes simulations. (On ne demande pas forcément d'implémenter la méthode dès cette question).

18. Expliquer pourquoi l'évaluation de π qu'on fait à la question 17 et celle qu'on faisait à la question 10, bien que provenant des mêmes simulations, sont indépendantes.

19. Soient deux variables aléatoires indépendantes $X_1 \sim m + \mathcal{N}(\sigma_1^2)$ et $X_2 \sim m + \mathcal{N}(\sigma_2^2)$. Pour $\alpha_1, \alpha_2 \in \mathbf{R}$, quelle est la variance de $\alpha_1 X_1 + \alpha_2 X_2$? En déduire pour quel choix de (α_1, α_2) la variance de $\alpha_1 X_1 + \alpha_2 X_2$ est minimale sous la contrainte « $\alpha_1 + \alpha_2 = 1$ ».

20. En déduire, lorsque \hat{m}_1 et \hat{m}_2 sont deux estimateurs non biaisés gaussiens indépendants d'une même quantité m , quelle est la meilleure façon de combiner des deux estimateurs pour évaluer plus finement m , et préciser que la variance du nouvel estimateur obtenu.

21. Implémenter la méthode de la question précédente pour combiner les deux estimations de π obtenues aux questions 10 et 17 ; commenter.

EXERCICE 36 — Arbitrage

L'arbitrage est une activité financière consistant à exploiter les petites incohérences du marché boursier pour gagner de l'argent par des transactions habiles sans avoir aucune connaissance des mécanismes économiques sous-jacents. C'est à une telle activité d'arbitrage que s'intéresse cet exercice, quoique dans un cas grossièrement simplifié et pas forcément réaliste.

Un cabinet d'arbitrage s'est rendu compte que la psychologie des traders faisait que l'évolution du cours des actifs financiers ne suivait pas exactement une martingale^{[[[} : il existe en fait un processus caché, appelé tendance, qui crée des périodes où les cours ont plutôt tendance à monter, ou plutôt tendance à descendre. Dans ce modèle simplifié, on supposera qu'il existe un seul actif financier, dont le prix au temps t est noté A_t ; et la « tendance » à l'instant t est notée T_t . A_t et T_t suivent l'évolution couplée suivante :

$$\begin{cases} dA_t = (\sigma_{AT}^2 |T_t| + \sigma_{A0}^2)^{1/2} A_t dW_t^A + T_t A_t dt, \\ dT_t = \sigma_T dW_t^T - \lambda_T T_t dt. \end{cases}$$

où $(W_t^T)_t$ et $(W_t^A)_t$ sont des mouvements browniens standard indépendants. Les paramètres sont les suivants : $\sigma_{AT} = 2$, $\sigma_{A0} = 1,5 \cdot 10^{-3} \text{ h}^{-1/2}$, $\sigma_T = 1,5 \cdot 10^{-5} \text{ h}^{-3/2}$, $\lambda_T = 0,6 \text{ h}^{-1}$.

La manœuvre à laquelle s'intéresse notre cabinet d'arbitrage est d'acheter un certain volume d'actions à l'heure actuelle ($t_0 := 9 \text{ h}$), et de les revendre au temps $\tau := 17\frac{1}{2} \text{ h}$. En moyenne, à l'issue de cette opération, on aura donc gagné $\mathbb{E}(A_\tau) - A_{t_0}$ par action : si cette valeur est positive, il y a donc opportunité d'arbitrage, car en répétant le même genre de manœuvre chaque jour où l'espérance est positive, le cabinet d'arbitrage aura la certitude de gagner à long terme !

1. Écrire une fonction `simulation` simulant l'évolution du processus couplé. Cette fonction prendra en entrée les valeurs de A_{t_0} et T_{t_0} , et affichera l'évolution de A et T sur l'intervalle de temps $[t_0, \tau]$. On écrira également une variante `simulation_muette` qui n'affiche rien, mais renvoie le prix final A_τ de l'actif.

Indication : Pour le tracé graphique, on utilisera la même échelle en abscisse pour les deux graphiques ; par contre les deux échelles en ordonnées pour A et T n'auront aucun rapport, puisque ce n'est même pas physiquement homogène... Une bonne idée pour ce faire est de tracer deux graphiques l'un sous l'autre à l'aide de la commande `subplot`.

2. Dans le cas $T_{t_0} = 4 \cdot 10^{-5} \text{ h}^{-1}$, $A_{t_0} = 100 \text{ €}$, estimer $E(A_\tau)$ par la méthode de Monte-Carlo, avec 10^4 simulations. Qu'en pensez-vous ?...

3. Calculer

$$\mathbb{E} \left(\int_{t=0}^{\tau} (\sigma_{AT}^2 |T_t| + \sigma_{A0}^2)^{1/2} A_t dW_t^A \right).$$

En déduire une variable de contrôle pertinente pour réduire la variance de l'estimation de $E(A_\tau)$, et implémenter la méthode de Monte-Carlo améliorée correspondante.

4. En réalité, T_{t_0} lui-même n'est pas connu avec certitude : notre cabinet d'arbitrage est simplement capable de le modéliser par une certaine loi normale. Disons que pour le jour qui nous intéresse, T_{t_0} suit la loi $\mathcal{N}(m, s^2)$, avec $m = 2 \cdot 10^{-5} \text{ h}^{-1}$ et $s = 10^{-5} \text{ h}^{-1}$. Écrire une variante du programme de la question précédente qui tienne compte aussi de l'incertitude sur T_{t_0} , et dans laquelle on introduit en prime la variable de contrôle $\lambda_T^{-1} A_{t_0} T_{t_0}$ pour réduire la variance supplémentaire engendrée par cette nouvelle incertitude.

[[[. Pour ceux qui connaissent la théorie financière, on supposera ici qu'on travaille sous la mesure risque-neutre (ou, de manière équivalente, que les investisseurs n'ont aucune aversion au risque). En outre, on supposera pour simplifier que le taux de rendement sans risque est nul.

5. (★) Nonobstant le caractère douteux de toutes les hypothèses de modélisation effectuées, pourquoi la méthode d'arbitrage étudiée dans cet exercice (pour les paramètres de l'énoncé) est-elle en fait inepte en pratique ?