

Contrôle Continu

Rappels

- Il sera tenu compte du soin apporté à la présentation et à la rédaction.
- Si vous n'arrivez pas à démontrer un résultat, vous pouvez l'admettre pour la suite de l'exercice.
- On suppose pouvoir simuler selon toutes les lois usuelles.

EXERCICE 1 (5 points) On considère la fonction de répartition

$$F(x) = \frac{1}{2} (x + x^2) \mathbf{1}_{0 \leq x < 1} + \mathbf{1}_{x \geq 1}.$$

On souhaite simuler X de fonction de répartition F .

1. Expliquer comment simuler X par la méthode d'inversion ?

Correction. Pour tout $0 < u < 1$, un calcul élémentaire donne

$$F^{-1}(u) = \sqrt{\frac{1}{4} + 2u} - \frac{1}{2}.$$

Donc il suffit de simuler $U \sim \text{Unif}([0, 1])$ puis de poser $X = F^{-1}(U) = \sqrt{\frac{1}{4} + 2U} - \frac{1}{2}$.

2. Calculer la densité f associée à la fonction de répartition F . Majorer f par une fonction constante sur $[0, 1]$ et décrire une méthode de rejet pour simuler X .

Correction. La densité est

$$f(x) = \left(x + \frac{1}{2}\right) \mathbf{1}_{0 \leq x < 1} \leq \frac{3}{2} \mathbf{1}_{0 \leq x \leq 1}.$$

Avec les notations du cours, on peut donc prendre $g(x) = \mathbf{1}_{0 \leq x \leq 1}$ et $m = 3/2$. Puisque g est la densité de la loi uniforme sur $[0, 1]$, la méthode de rejet fonctionne comme suit :

- (i) Simuler $Y \sim \text{Unif}([0, 1])$ et calculer

$$r(Y) = \frac{f(Y)}{mg(Y)} = \frac{2}{3} \left(Y + \frac{1}{2}\right).$$

- (ii) Simuler $U \sim \text{Unif}([0, 1])$: si $U \leq r(Y)$, poser $X = Y$, sinon retourner en (i).

3. Montrer qu'on peut voir la loi de X comme un mélange de deux lois classiques que l'on précisera et décrire une nouvelle méthode pour simuler X .

Correction. La formule de f montre que la loi de X est un mélange équiprobable entre une loi uniforme sur $[0, 1]$ et une loi Beta(2, 1) donc pour simuler X , il suffit de commencer par simuler $B \sim \text{Ber}(1/2)$, puis : si $B = 0$, on simule $X \sim \text{Unif}([0, 1])$; si $B = 1$, on simule $X \sim \text{Beta}(2, 1)$. Ceci s'écrit encore $X = (1 - B)U + BY$ avec $B \sim \text{Ber}(1/2)$, $U \sim \text{Unif}([0, 1])$, $Y \sim \text{Beta}(2, 1)$, les 3 variables étant indépendantes.

4. Des trois méthodes, laquelle choisissez-vous ?

Correction. On choisit la première, qui nécessite uniquement la simulation d'une variable (de loi uniforme) et l'appel à une fonction élémentaire. La seconde méthode nécessite à chaque fois la simulation de deux variables, avec une probabilité d'acceptation de $\frac{1}{m} = \frac{2}{3}$ donc elle est à proscrire. La simulation selon la loi de mélange est meilleure que la méthode de rejet mais un peu plus coûteuse que la première.

EXERCICE 2 (5 points) On considère un couple (X, Y) dont la loi admet la densité

$$f(x, y) = xe^{-(1+y)x} \mathbf{1}_{x>0, y>0}.$$

1. Soit $y > 0$, quelle est la loi de X sachant $Y = y$? Soit $x > 0$, quelle est la loi de Y sachant $X = x$? Reconnaître des lois usuelles.

Correction. Pour $y > 0$ fixé, on a

$$f(x|y) \propto f(x, y) \propto xe^{-(1+y)x} \mathbf{1}_{x>0},$$

donc $\text{Loi}(X|Y = y) = \text{Gamma}(2, 1 + y)$. De même, pour $x > 0$ fixé,

$$f(y|x) \propto f(x, y) \propto e^{-xy} \mathbf{1}_{y>0},$$

donc $\text{Loi}(Y|X = x) = \text{Exp}(x)$.

2. En déduire un échantillonneur de Gibbs séquentiel admettant f pour loi stationnaire.

Correction. On part d'un point $(X_0, Y_0) = (x_0, y_0) \in (\mathbb{R}_+^*)^2$. Sachant (X_n, Y_n) :

— On simule $X_{n+1} \sim \text{Gamma}(2, 1 + Y_n)$;

— On simule $Y_{n+1} \sim \text{Exp}(X_{n+1})$.

3. La chaîne de Markov ainsi construite est-elle irréductible ? apériodique ? S'ils sont vérifiés, formuler précisément la convergence en variation totale et le Théorème ergodique.

Correction. Sans faire de calculs, la marginale en x de $f(x, y)$, c'est-à-dire $f(x) = \int_{\mathbb{R}} f(x, y) dy$, est positive sur $\mathbb{R}_+^*$, qui est donc son support. Il en va de même pour la marginale en y . Quant au support de la loi jointe $f(x, y)$, c'est clairement $(\mathbb{R}_+^*)^2$, qui est donc le produit cartésien des supports des marginales. Par un résultat du cours, si on note π la loi de densité $f(x, y)$, la chaîne est donc π -irréductible et apériodique. De plus, π est équivalente à la mesure de Lebesgue sur $(\mathbb{R}_+^*)^2$. On en déduit que pour presque toute condition initiale $(x_0, y_0) \in (\mathbb{R}_+^*)^2$, on a

$$\|\text{Loi}(X_n, Y_n) - \pi\| \xrightarrow[n \rightarrow \infty]{} 0,$$

et

$$\frac{1}{n} \sum_{k=1}^n \varphi(X_k, Y_k) \xrightarrow[n \rightarrow \infty]{\text{p.s.}} \iint_{(\mathbb{R}_+^*)^2} \varphi(x, y) f(x, y) dx dy,$$

pour toute fonction $\varphi : (\mathbb{R}_+^*)^2 \rightarrow \mathbb{R}$ telle que

$$\iint_{(\mathbb{R}_+^*)^2} |\varphi(x, y)| f(x, y) dx dy < \infty.$$

4. Quelle est la loi de X ? En déduire une méthode pour générer un couple (X, Y) de densité f .

Correction. Pour tout $x > 0$, la densité marginale de X en x vaut

$$f(x) = \int_{\mathbb{R}} f(x, y) dy = e^{-x} \int_0^{\infty} x e^{-xy} dy = e^{-x} [e^{-xy}]_{\infty}^0 = e^{-x},$$

donc $X \sim \text{Exp}(1)$. Ainsi pour simuler directement (X, Y) de densité f , il suffit de simuler $X \sim \text{Exp}(1)$ puis, sachant $X = x$, simuler $Y \sim \text{Exp}(x)$.

5. Entre cette méthode et l'échantillonneur de Gibbs, laquelle choisissez-vous ?

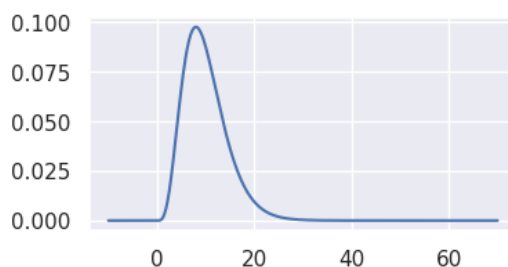
Correction. On choisit bien sûr cette dernière méthode de simulation : elle permet de simuler en un coup un couple de loi exactement π , tandis que les résultats assurés par l'échantillonneur de Gibbs sont seulement asymptotiques.

EXERCICE 3 (5 points)

1. Que représente le script suivant ? Quelle est la loi considérée ?

```
[3]: dist = stats.gamma(a=5, scale=2)

t = np.linspace(-10, 70, num=500)
plt.figure(figsize=(4, 2))
plt.plot(t, dist.pdf(t));
```



Correction. La fonction de densité de la loi $\text{Gamma}(5, 1/2)$.

2. Le script suivant affiche une réalisation d'un estimateur $\hat{I}_n$. Donner la formulation de cet estimateur et justifier sa convergence vers une quantité I à préciser (on ne demande pas de calculer I).

```
[4]: n = 10**5

x = dist.rvs(size=n)
est = np.mean( np.exp(-(x-50)**2/2) )

print(f"{est:.3e}")
```

6.395e-13

Correction. Soient $X_1, \dots, X_n$ i.i.d. selon la loi $\text{Gamma}(5, 1/2)$. L'estimateur considéré est $\hat{I}_n = \frac{1}{n} \sum_{i=1}^n e^{-\frac{(X_i-50)^2}{2}}$ qui, par la loi forte des grands nombres, converge vers $I = \mathbb{E} \left[e^{-\frac{(X_1-50)^2}{2}} \right]$. On pourra noter que cette espérance est bien définie puisque $0 \leq e^{-\frac{(X_1-50)^2}{2}} \leq 1$.

3. On souhaite afficher l'évolution de $\hat{I}_n$ en fonction de n . Par quoi faut-il remplacer ??? dans le script ci-dessous ?

```
[6]: n_range = np.arange(1, n+1)
     est_cum = ???
```

```
plt.plot(n_range, est_cum);
```

Correction. On peut écrire `??? = np.cumsum( np.exp(-(x-50)**2/2) ) / n_range`.

4. Proposer un estimateur Monte-Carlo par échantillonnage préférentiel de I , noté $\tilde{I}_n$.

Correction. Vu que la fonction $\varphi : x \mapsto e^{-\frac{(x-50)^2}{2}}$ à intégrer est maximale en 50, il est naturel (mais pas nécessairement optimal) de prendre une loi qui met beaucoup de masse en 50, par exemple $\mathcal{N}(50, 1)$. En appelant f la densité de la loi Gamma(5, 1/2), on peut en fait vérifier que la fonction φf est maximale pour une valeur proche de 50. Notons g la densité de $\mathcal{N}(50, 1)$. Puisque g ne s'annule pas, le rapport suivant est bien défini pour tout $x \in \mathbb{R}$ et vaut :

$$\frac{\varphi(x)f(x)}{g(x)} = \frac{e^{-\frac{(x-50)^2}{2}} f(x)}{e^{-\frac{(x-50)^2}{2}} / \sqrt{2\pi}} = \sqrt{2\pi} f(x).$$

Ainsi,

$$I = \int_{\mathbb{R}} \varphi(x)f(x) dx = \int_{\mathbb{R}} \frac{\varphi(x)f(x)}{g(x)} g(x) dx = \sqrt{2\pi} \int_{\mathbb{R}} f(x)g(x) dx = \sqrt{2\pi} \mathbb{E}[f(Y)],$$

avec $Y \sim \mathcal{N}(50, 1)$. Un estimateur Monte-Carlo par échantillonnage préférentiel de I est donc, considérant $Y_1, \dots, Y_n$ i.i.d. selon $\mathcal{N}(50, 1)$,

$$\tilde{I}_n = \frac{\sqrt{2\pi}}{n} \sum_{i=1}^n f(Y_i).$$

5. Par quoi faut-il remplacer `???` et `$$$` ci-dessous pour implémenter l'estimateur $\tilde{I}_n$?

```
[5]: aux = ???

y = aux.rvs(size=n)
estEP = $$$

print(f"{estEP:.3e}")
```

Correction. On peut écrire `??? = stats.norm(loc=50)`
et `$$$ = np.sqrt(2*np.pi) * np.mean( dist.pdf(y) )`.

6. L'estimateur $\tilde{I}_n$ correctement normalisé dispose-t-il d'une variance asymptotique finie ?

Rappel : La loi Gamma(a, b) (où $a > 0$ et $b > 0$) a une densité proportionnelle à $x \mapsto x^{a-1} e^{-bx} \mathbf{1}_{x \geq 0}$.

Correction. Il est clair que $f(Y) \in L^2$ puisque :

$$\mathbb{E}[f(Y)^2] \propto \int_0^\infty \left(y^4 e^{-y/2}\right)^2 e^{-\frac{(y-50)^2}{2}} dy < \infty.$$

Ainsi, le théorème central limite assure que

$$\sqrt{n} \left(\tilde{I}_n - I \right) \xrightarrow[n \rightarrow \infty]{\mathcal{L}} \mathcal{N} \left(0, \text{Var} \left(\sqrt{2\pi} f(Y) \right) \right),$$

avec $\text{Var}(\sqrt{2\pi} f(Y)) = 2\pi \mathbb{E}[f(Y)^2] - I^2$.

EXERCICE 4 (5 points)

1. Pour $\mu \geq 0$ et $\sigma > 0$, on note $\mathcal{N}_+(\mu, \sigma^2)$ la loi de densité f vérifiant :

$$\forall x \in \mathbb{R}, \quad f(x) \propto e^{-\frac{(x-\mu)^2}{2\sigma^2}} \mathbf{1}_{x \geq 0}.$$

Expliquer en quoi la fonction ci-dessous simule par rejet selon une loi de la forme $\mathcal{N}_+(\mu, 1)$.

```
[8]: def sample(loc=0, size=1):
    aux = stats.norm(loc=loc)
    TryAgain = True
    while TryAgain:
        x = aux.rvs(size=3*size)
        x = x[x >= 0]
        TryAgain = (x.size < size)
    return x[:size]
```

Correction. Soit $\mu \geq 0$. La loi $\mathcal{N}_+(\mu, 1)$ a pour densité $f(x) = \frac{c}{\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2}} \mathbf{1}_{x \geq 0}$ (pour tout $x \in \mathbb{R}$), avec $c > 0$ (on a même $c \geq 1$). Ici `aux` représente la loi de proposition, qui est $\mathcal{N}(\mu, 1)$, avec `loc` = μ . Appelons $g(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2}}$ sa densité.

Pour tout $y \in \mathbb{R}$, le rapport des densités vaut $\frac{f(y)}{g(y)} = c \mathbf{1}_{y \geq 0}$, qui est uniformément majoré par c . Ainsi, le rapport d'acceptation est $r(y) = \frac{f(y)}{cg(y)} = \mathbf{1}_{y \geq 0}$. Pour $U \sim \text{Unif}([0, 1])$, le test $U \leq r(y)$ est donc presque sûrement égal à tester $y \geq 0$. Ainsi, simuler $\mathcal{N}_+(\mu, 1)$ par rejet revient simplement à tirer une proposition suivant $\mathcal{N}(\mu, 1)$ puis à la conserver si elle est positive et à recommencer sinon.

Le fait est que, puisque $\mu \geq 0$, le taux moyen de rejet est inférieur à $1/2$. On tire donc d'emblée (suivant la loi de proposition) 3 fois plus d'observations qu'il n'en faut puis on ne conserve que les positives. Cela suffit généralement à obtenir le nombre voulu de tirages. Si ce n'est pas le cas, on recommence.

2. On considère le schéma bayésien suivant :

$$\begin{cases} \boldsymbol{\theta} \sim \chi_4^2 \\ \mathbf{X} = (X_1, \dots, X_N) \mid \boldsymbol{\theta} \sim (\mathcal{N}_+(\boldsymbol{\theta}, 1))^{\otimes N}. \end{cases}$$

Compléter le code suivant pour simuler une réalisation $\mathbf{X}$ du vecteur aléatoire $\mathbf{X}$.

```
[9]: N = 200

prior = stats.chi2(df=4)
```

Correction. Il suffit d'écrire :

```
theta = prior.rvs()
X = sample(loc=theta, size=N)
```

3. Expliciter, à une constante multiplicative près, la loi a posteriori $\Pi[\cdot \mid \mathbf{X}]$.

Rappel : Pour un entier non nul d , la loi χ_d^2 a une densité proportionnelle à $x \mapsto x^{d/2-1} e^{-x/2} \mathbf{1}_{x \geq 0}$.

Correction. La loi a posteriori est à densité par rapport à la mesure de Lebesgue sur $\mathbb{R}$, et vérifie :

$$\forall \theta \in \mathbb{R}, \quad \pi(\theta \mid \mathbf{X}) \propto \pi(\theta) p_{\theta}(\mathbf{X}) \propto \theta e^{-\frac{\theta + \sum_{i=1}^N (\theta - X_i)^2}{2}} \mathbf{1}_{\theta \geq 0},$$

qui n'est pas une loi usuelle.

4. Expliquer en termes mathématiques et en détail l'objectif de ce script :

```
[10]: n = 10**4
init = X.mean()

unif = stats.uniform()
kernel = stats.uniform(loc=-0.25, scale=0.5)
threshold = unif.rvs(size=n)
move = kernel.rvs(size=n)

sample = [init]
n_acc = 0
for i in range(n):
    x = sample[-1]
    y = x + move[i]
    crit = y * np.exp(-(y + np.sum((X-y)**2))/2) * (y >= 0) / ( x * np.exp(-(x +
    ↪np.sum((X-x)**2))/2) * (x >= 0) )
    dec = threshold[i] < crit
    sample.append(y if dec else x)
    n_acc += dec
sample = np.asarray(sample)
rate = n_acc / n
```

Correction. Il s'agit d'un algorithme itératif qui, à chaque itération, ajoute à la liste `sample` une nouvelle proposition ou la dernière valeur acceptée. Dans le cadre de ce cours, cela ne peut être qu'un algorithme de Metropolis. Par ailleurs, la proposition est donnée par $y = x + \text{move}[i]$, avec `move` un tableau de réalisations issues de $\text{Unif}([-1/4, 1/4])$, ce qui signifie que le noyau de proposition est, pour tout $x \in \mathbb{R}$, $Q(x, \cdot) = \text{Unif}([x - 1/4, x + 1/4])$. Ce noyau est à densité par rapport à la mesure de Lebesgue sur $\mathbb{R}$, laquelle s'écrit, pour tout $(x, y) \in \mathbb{R}^2$, $q(x, y) = 2\mathbf{1}_{|x-y| \leq 1/4}$. La fonction q étant symétrique, nous sommes dans le cas de l'algorithme de Metropolis symétrique, dont le rapport s'écrit :

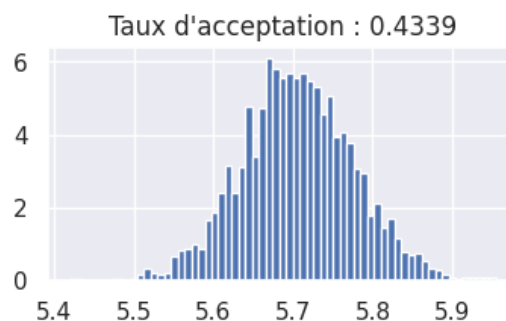
$$\rho(x, y) = \frac{g(y)q(y, x)}{g(x)q(x, y)} = \frac{g(y)}{g(x)},$$

où g est la densité cible. On reconnaît d'après le code ci-dessus pour g :

$$g(y) \propto y e^{-\frac{y + \sum_{i=1}^N (X_i - y)^2}{2}} \mathbf{1}_{y \geq 0} \propto \pi(y \mid \mathbf{X}).$$

Ainsi, le script implémente un algorithme de Metropolis symétrique pour simuler approximativement suivant la loi a posteriori $\Pi[\cdot \mid \mathbf{X}]$ avec le noyau de proposition $Q(x, \cdot) = \text{Unif}([x - 1/4, x + 1/4])$.

5. Un histogramme de `sample` donne le résultat suivant. Que pouvez-vous en dire ?



Correction. Sans avoir vérifié les hypothèses du théorème de Bernstein-von Mises, on peut imaginer que la loi a posteriori est proche ($N = 200$) d'une loi normale, ce qui se vérifie sur l'histogramme. Par ailleurs, le taux d'acceptation est compris entre 20% et 80%, ce qui donne confiance dans le résultat obtenu.

6. Qu'affiche le script suivant ?

```
[12]: alpha = 0.05

inter_low = np.quantile(sample, alpha/2, method='inverted_cdf')
inter_up = np.quantile(sample, 1-alpha/2, method='inverted_cdf')

print(f"[{inter_low:0.3f}, {inter_up:0.3f}]")
```

Correction. Puisqu'on a simulé approximativement suivant la loi a posteriori $\Pi[\cdot | \mathbf{X}]$, le script donne une approximation d'une réalisation d'un intervalle de crédibilité de niveau $1 - \alpha = 95\%$ pour θ .