

Simuler une distribution normale à support positif à partir de plusieurs lois candidates

Vincent MAZET¹, David BRIE¹, Jérôme IDIER²

¹Université Henri Poincaré, Nancy 1, CRAN, CNRS UMR 7039,
BP 239, F-54506 Vandœuvre-lès-Nancy Cedex

²École Centrale de Nantes, IRCCyN, CNRS UMR 6597,
BP 92101, F-44321 Nantes Cedex 3

{vincent.mazet,david.brie}@cran.uhp-nancy.fr, jerome.idier@irccyn.ec-nantes.fr

Résumé – Dans cet article, nous présentons une méthodologie pour générer des échantillons suivant une loi normale à support positif dans le cas scalaire grâce à l’algorithme d’acceptation-rejet. Nous proposons plusieurs lois candidates, et choisissons celle qui donnera le meilleur taux d’acceptation moyen pour des paramètres de la loi cible donnés. Nous obtenons alors une méthode très rapide puisque générant très peu de rejet.

Abstract – In this article, we provide an accept-reject algorithm to simulate positive normal variables in the univariate case. The main idea consists in determining among different proposal distributions a priori chosen the one which gets the highest average probability of acceptance, depending on the shape of the target distribution. It yields a fast method since it generates low reject.

1 Introduction

En déconvolution impulsionnelle, le signal à restaurer est souvent modélisé par un processus Bernoulli-gaussien. Or, dans certaines applications (telles que la spectroscopie [6, 9], ou la génétique [7]), les pics sont tous positifs. Ils peuvent donc être modélisés par un a priori Bernoulli-gaussien à support dans \mathbb{R}^+ , la loi a posteriori devient alors le mélange d’un pic de Dirac et d’une gaussienne de moyenne et de variance quelconques tronquée en zéro (c’est-à-dire à support positif). Dans le cas où l’on souhaite résoudre le problème en utilisant une méthode MCMC (notamment lorsque le nombre de variables est très important), il est nécessaire de générer des variables gaussiennes positives. Un autre type d’application est la séparation myope de sources positives dont les coefficients de mélanges sont également positifs en analyse de mélanges chimiques [8].

Nous présentons donc dans cet article une approche pour simuler la loi normale à support positif qui a pour expression :

$$f(x) = \frac{1}{C} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \mathbb{1}_{\mathbb{R}^+} \quad (1)$$

où

$$C = \sqrt{\frac{\pi\sigma^2}{2}} \left[1 + \operatorname{erf}\left(\frac{\mu}{\sqrt{2\sigma^2}}\right)\right].$$

Notons que la valeur de C n’est pas nécessaire pour la mise en œuvre de l’algorithme. On peut calculer facilement la moyenne et la variance de cette loi :

$$\mathbb{E}[x] = \mu + \sqrt{\frac{2\sigma^2}{\pi}} \frac{\exp(-\mu^2/2\sigma^2)}{1 + \operatorname{erf}(\mu/\sqrt{2\sigma^2})},$$
$$\operatorname{Var}[x] = \sigma^2 + \frac{\mu^2}{4} - \left[\frac{\mu}{2} + \sqrt{\frac{2\sigma^2}{\pi}} \frac{\exp(-\mu^2/2\sigma^2)}{1 + \operatorname{erf}(\mu/\sqrt{2\sigma^2})}\right]^2.$$

La technique d’inversion de la fonction de répartition [3, 4] consiste à générer $u \sim \mathcal{U}_{[0,1]}$, puis à calculer :

$$x = \mu + \sqrt{2\sigma^2} \operatorname{erf}^{-1}\left(u + \operatorname{erf}(\mu/\sqrt{2\sigma^2})(u-1)\right)$$

où erf est la fonction erreur. Cette méthode a l’avantage de fournir une solution explicite mais suppose que les fonctions erf et erf^{-1} sont parfaitement connues. En pratique, l’utilisation de ces fonctions peut se révéler inefficace car elles ne peuvent être qu’approchées numériquement et l’erreur d’approximation devient importante lorsque $|\mu|$ est trop grand (cf. section 4) [5, 10].

Une autre approche consiste à utiliser un algorithme d’acceptation-rejet [3, 10, 11]. La loi candidate la plus simple est évidemment la loi normale, mais elle n’est adaptée que dans le cas où μ est suffisamment grand (cf. figure 2). Au contraire, la loi exponentielle proposée par Robert [10] n’est bien adaptée que lorsque μ est suffisamment petit (cf. figure 2). En effet, lorsque $\mu \rightarrow -\infty$, la loi normale tronquée tend à ressembler à une loi exponentielle [5].

Comme la forme de la loi varie en fonction de μ et σ^2 , nous proposons d’utiliser une méthode basée sur l’algorithme d’acceptation-rejet en utilisant plusieurs lois candidates g , l’idée étant que chacune soit adaptée aux formes particulières de la loi cible. Cette idée d’un *algorithme d’acceptation-rejet mixte* a déjà été utilisée par Geweke [5], qui propose deux lois candidates (les lois normale et exponentielle) pour simuler une loi normale tronquée unilatéralement et détermine les intervalles d’utilisation de ces lois de manière empirique. Or, une détermination empirique donne des valeurs d’intervalles dépendant du code utilisé. Pour éviter cela, nous déterminons les valeurs des intervalles en fonction du taux d’acceptation théorique (lié au temps de calcul), ce qui permet d’obtenir des valeurs indépendantes du codage. De plus, afin d’améliorer les performances

de l'algorithme, nous proposons d'utiliser quatre lois candidates. Cette idée d'algorithme d'acceptation-rejet mixte peut s'adapter sur toute loi cible qui n'est pas directement échantillonnable par les techniques classiques, et dont la forme varie en fonction de ses paramètres.

Dans la section 2, nous présentons la méthodologie de manière générale puis nous l'appliquons au cas particulier de la simulation d'une loi normale à support positif dans la section 3. La section 4 propose des comparaisons avec des méthodes existantes et montre que l'approche proposée simule correctement une loi normale tronquée. Enfin, la section 5 conclue ce travail en y présentant quelques perspectives de recherche.

2 Méthodologie générale

L'algorithme d'acceptation-rejet requiert la détermination d'une loi candidate g et d'une constante M telles que

$$\forall x \in S, \quad M \geq f(x)/g(x) \quad (2)$$

où S est le support de f . La loi candidate $g(x)$ est nécessairement non nulle sur S afin que M soit fini et donc que le taux d'acceptation moyen soit non nul (cf. équation (4)). L'algorithme d'acceptation-rejet découle du lemme suivant :

Lemme 1 [3, 10, 11] La variable aléatoire x résultante de l'algorithme suivant est distribuée suivant f :

1. générer $z \sim g(z)$ et $u \sim \mathcal{U}_{[0,1]}$,
2. calculer $\rho(z) = f(z)/Mg(z)$,
3. si $u \leq \rho(z) : x = z$ (acceptation),
sinon : retour en 1 (rejet).

Le choix des lois candidates est déterminant pour les performances de la méthode. Tout d'abord, elles doivent être facilement échantillonnables, sinon l'algorithme d'acceptation-rejet perd son intérêt. En particulier, elles doivent être simulables avec un taux d'acceptation de 1, sinon le taux d'acceptation moyen correspondant doit être pris en compte dans le calcul du taux d'acceptation moyen global. Précisons également que le choix des lois candidates doit prendre en compte la complexité de l'algorithme : certaines lois intéressantes a priori s'avéreront inadéquates car la détermination de M ou le calcul de ses paramètres peut être difficile, gourmand en temps de calcul, voire impossible ! Ainsi, on peut choisir des lois courantes (dans notre exemple : les lois normale et exponentielle) ou construire des lois particulières afin d'obtenir des lois candidates convenables (une loi normale couplée à la loi uniforme pour notre exemple).

Pour le choix de M , n'importe quelle constante vérifiant l'équation (2) convient ; cependant, il est nécessaire que M soit la plus petite possible pour avoir un taux d'acceptation moyen élevé (équation (4)). La valeur optimale de M est donc :

$$M = \max_{x \in S} f(x)/g(x).$$

Malheureusement, cette valeur n'est pas toujours calculable. Cependant, si elle l'est, alors on peut déterminer le taux d'acceptation ρ :

$$\rho(x) = f(x)/Mg(x). \quad (3)$$

Enfin, le taux d'acceptation moyen $\bar{\rho} = \mathbb{E}[\rho(X)]$ nous permettra de déterminer les zones d'utilisation des lois candidates :

$$\bar{\rho} \triangleq \int \rho(x)g(x)dx = \frac{1}{M} \int f(x)dx = \frac{1}{M}. \quad (4)$$

Remarquons que si $g(x)$ est proche de zéro, alors M devient très grand et le taux d'acceptation moyen diminue. L'efficacité de l'algorithme dépend donc de l'adéquation entre f et g ; en particulier, pour que M reste borné, il faut que g ait des queues lourdes que celle de f , sans toutefois que leur rapport soit trop grand, au risque d'augmenter M [11].

L'expression du taux d'acceptation dépend dans certains cas des paramètres de la loi candidate (par exemple, le paramètre α de la loi exponentielle dans notre cas particulier : cf. section 3). Il faut alors trouver les valeurs de ces paramètres qui maximisent le taux d'acceptation.

Parmi l'ensemble des lois candidates utilisées, une seule fournit le meilleur taux d'acceptation moyen pour des paramètres de la loi cible donnés. Il faut donc déterminer les intervalles sur ces paramètres pour déterminer quelle est la meilleure loi candidate à utiliser sur l'intervalle considéré. L'algorithme proposé est donc le même que l'algorithme d'acceptation-rejet avec une étape préliminaire qui consiste à choisir la meilleure loi candidate g_i en fonction des paramètres de f , puis à calculer la constante M correspondante :

1. sélectionner la loi candidate g_i la plus appropriée en fonction des paramètres de f ,
2. calculer M_i ,
3. générer $z \sim g_i(z)$ et $u \sim \mathcal{U}_{[0,1]}$,
4. calculer $\rho(z) = f(z)/M_i g_i(z)$,
5. si $u \leq \rho(z) : x = z$ (acceptation),
sinon : retour en 3 (rejet).

Notons que si f est tronquée à $t \neq 0$, la méthode peut être adaptée par simple translation de la variable aléatoire générée $X : Y = X + t$. De même, sans perdre en généralité, on peut se restreindre au cas $\sigma^2 = 1$, puisque tous les autres cas se déduisent par changement d'échelle $Y = X\sigma$: nous supposons dans la suite que $\sigma^2 = 1$.

3 Simulation d'une loi normale tronquée

Quatre lois candidates (représentées figure 1) sont proposées pour simuler la loi cible f :

- ① La loi normale :

$$g_1(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) ;$$

- ② La loi normale couplée à la loi uniforme (c'est une distribution définie sur \mathbb{R}^+ , uniforme sur $[0, \mu[$ et qui suit la loi normale $\mathcal{N}(\mu, \sigma^2)$ sur $[\mu, +\infty[$, $\mu \geq 0$) :

$$g_2(x) = \frac{\mathbb{1}_{\mathbb{R}^+}}{\mu + \sqrt{\frac{\pi\sigma^2}{2}}} \begin{cases} 1 & \text{si } 0 \leq x < \mu, \\ \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) & \text{si } x \geq \mu ; \end{cases}$$

- ③ La loi normale tronquée en la moyenne ($\mu \leq 0$) :

$$g_3(x) = \frac{2}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \mathbb{1}_{[\mu, +\infty[} ;$$

- ④ La loi exponentielle [10] :

$$g_4(x) = \alpha \exp(-\alpha x) \mathbb{1}_{\mathbb{R}^+}$$

où la valeur de α correspond à la valeur qui maximise le taux d'acceptation moyen :

$$\alpha = \left(\sqrt{\mu^2 + 4\sigma^2} - \mu\right) / 2\sigma^2. \quad (5)$$

Des techniques de génération d'échantillons distribués suivant ces lois candidates sont présentées dans l'annexe A.

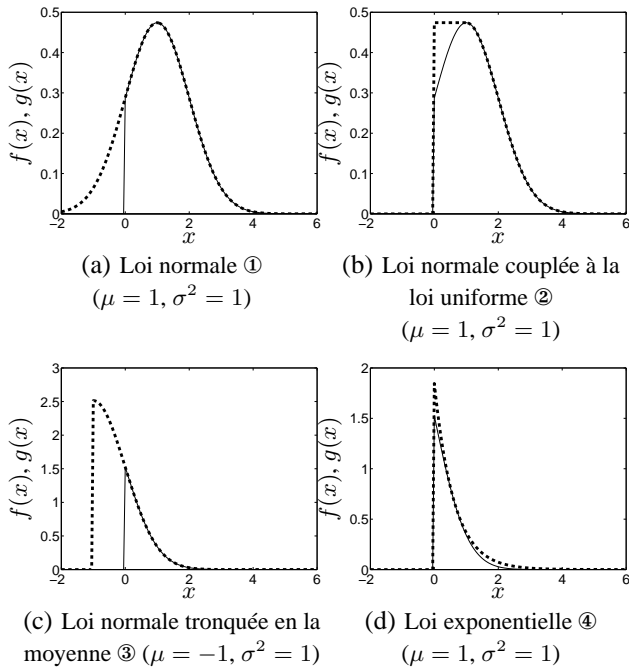


FIG. 1 – Lois candidates proposées

Notre choix est motivé par le fait que les lois ① et ④ permettent d'obtenir un taux d'acceptation moyen très important lorsque $|\mu| \gg 0$, et les lois ② et ③ permettent d'améliorer le taux d'acceptation moyen autour de zéro (cf. figure 2). Pour chacune de ces lois, nous pouvons calculer l'expression des constantes M (et ainsi les taux d'acceptation moyens) :

$$M_1 = \sqrt{2\pi\sigma^2}/C$$

$$M_2 = \left(\mu + \sqrt{\pi\sigma^2/2}\right)/C$$

$$M_3 = \sqrt{2\pi\sigma^2}/2C$$

$$M_4 = \exp\left(\frac{\alpha}{2}(2\mu + \alpha\sigma^2)\right)/\alpha C$$

Le calcul des taux d'acceptation $\rho(x)$ est alors direct.

Les taux d'acceptation moyens pour chacune des quatre lois candidates ont été représentés en fonction de μ figure 2. Nous pouvons alors trouver les expressions des trois points d'intersections qui délimitent les zones. Les calculs sont directs pour μ_B et μ_C , mais on ne peut que déterminer une valeur approchée de μ_A . En effet, d'après l'équation (5), on a :

$$\mu = (1 - \alpha^2\sigma^2)/\alpha,$$

et, en égalant les taux d'acceptation moyen des lois ③ et ④, on obtient :

$$\alpha\sigma \exp(\alpha^2\sigma^2/2) = e\sqrt{2/\pi}.$$

Cette dernière équation définit $\alpha\sigma$ sous une forme implicite. On peut en déduire que $\alpha\sigma$ s'exprime sous la forme

$$\alpha\sigma = \sqrt{W(2e^2/\pi)} \approx 1,1367,$$

où W est la fonction de Lambert définie comme l'inverse multivaluée de la fonction $f(w) = we^w$ [2]. Ainsi :

$$\mu_A \approx -0,257\sigma, \quad \mu_B = 0, \quad \mu_C = \sigma\sqrt{\pi/2}.$$

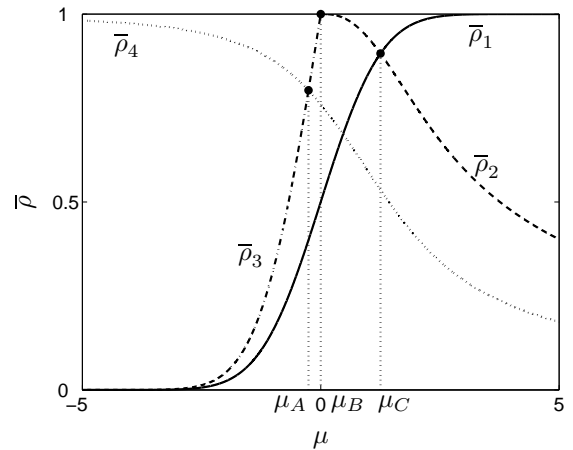


FIG. 2 – Taux d'acceptation moyens pour les quatre lois candidates en fonction de μ ($\sigma^2 = 1$)

Le taux d'acceptation moyen le plus faible correspond à $\mu = \mu_A$; il est égal à 0,797 environ, ce qui est un très bon taux d'acceptation. L'algorithme obtenu est donc très rapide car il génère très peu de rejet.

4 Simulations numériques

Toutes les simulations ont été faites avec Matlab. La fonction erf est définie dans [1], et les variables uniformes et normales sont générées avec rand et randn respectivement.

La méthode d'inversion [3, 4], bien qu'elle soit explicite et que le taux d'acceptation soit de 1, s'avère inefficace dans certains cas pratiques où $-\mu$ est trop grand [5]. Cela est dû à la fonction erf dont le calcul engendre des approximations créant alors des problèmes numériques. Pour illustrer ce propos, considérons la génération de 100 000 variables normales positives avec le paramètre $\mu = -7.5$ grâce à la méthode d'inversion et l'approche proposée. Les histogrammes correspondants sont représentés figure 3¹. Il est clair que la méthode

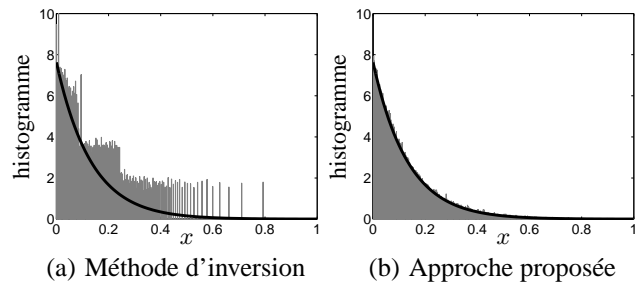


FIG. 3 – Histogrammes de 100 000 variables ($\mu = -7.5, \sigma^2 = 1$), et la distribution normale tronquée correspondante.

d'inversion s'avère inefficace alors que l'approche proposée fonctionne très bien. Évidemment, les performances de la méthode d'inversion pourraient être améliorées en utilisant une meilleure approximation de la fonction erf, mais cela entraînerait un coup calculatoire conséquent. De plus, pour certaines

¹Les éventuelles valeurs infinies générées par la méthode d'inversion n'ont pas été représentées sur l'histogramme.

	AP	①	②	④
$\mu = -2$	0,36 s	11,094 s	($\mu < 0!$)	0,39 s
$\mu = 0$	0,406 s	0,485 s	0,422 s	0,454 s
$\mu = 0.5$	0,406 s	0,375 s	0,422 s	0,515 s
$\mu = 2$	0,281 s	0,297 s	0,531 s	0,813 s

TAB. 1 – Comparaison des temps de calcul pour l’approche proposée (AP) et les algorithmes d’acceptation-rejet utilisant les lois candidates ①, ② et ④.

valeurs des paramètres, la méthode d’inversion peut générer des valeurs négatives ou infinies. Si elles restent relativement peu nombreuses, le problème peut être résolu en générant de nouvelles variables, mais, dans certains cas, la méthode peut ne donner que des valeurs inexploitable (par exemple avec $\mu = -8.5$) : la méthode d’inversion s’avère alors inappropriée.

Comparons maintenant l’approche proposée avec trois algorithmes d’acceptation-rejet classiques utilisant les lois candidates ①, ② et ④. Nous comparons le temps de calcul pour différents μ , ce qui, en général, est relativement proportionnel au taux d’acceptation moyen. Les temps de simulation de 10 000 variables sont présentés dans le tableau 1. Dans tous les cas, les temps de calcul sont cohérents avec les taux d’acceptation moyens de la figure 2, sauf pour $\mu = 0, 5$ où la loi normale ① est plus rapide que la loi normale couplée à la loi uniforme ②. Cela est dû au fait que générer une variable normale est beaucoup plus rapide que générer une variable suivant la loi ②. Mais comme cette différence dépend du programme utilisé, il est difficile de déterminer les différents intervalles à partir des temps de calcul réels. C’est pourquoi nous préférons le faire à partir des taux d’acceptation moyens théoriques, ce qui permet tout de même d’obtenir de bonnes valeurs pour les intervalles. De toute manière, la différence est faible et particulière au cas $\mu = 0, 5$.

5 Conclusion

Nous avons présenté une approche pour simuler des variables suivant une loi dont la forme varie avec ses paramètres. L’idée développée utilise un algorithme d’acceptation-rejet dont la loi candidate est sélectionnée parmi d’autres lois choisies a priori. L’approche est ensuite appliquée au cas particulier de la génération de variables aléatoires normales à support positif. Quatre lois candidates ont été présentées, et les intervalles sur lesquelles elles sont le plus performantes en terme de taux d’acceptation ont été déterminés. Des simulations ont illustré le fait que la méthode fonctionne bien et rapidement dans toutes les situations possibles.

Les travaux futurs se concentreront sur la recherche de lois candidates plus performantes, telle que la loi normale couplée à un polynôme, ou la loi normale tronquée en la moyenne dont les deux paramètres sont déterminés pour maximiser le taux d’acceptation moyen. De plus, les recherches pourront également être tournées vers le cas de la simulation d’une loi normale tronquée des deux côtés ce qui peut être utile dans le traitement de données censurées, ou pour simuler des lois normales multidimensionnelles à support positif.

A Simulation des lois candidates

Un échantillon x distribué suivant la loi ② est situé soit dans la partie uniforme (d’aire \mathcal{A}_u), soit dans la partie gaussienne (d’aire \mathcal{A}_g). Comme ces aires sont égales à :

$$\mathcal{A}_u = \frac{\mu}{\mu + \sqrt{\pi\sigma^2/2}} \quad \text{et} \quad \mathcal{A}_g = \frac{\sqrt{\pi\sigma^2/2}}{\mu + \sqrt{\pi\sigma^2/2}},$$

elles sont de somme unité. Ainsi, l’algorithme suivant permet de générer un échantillon x suivant la loi ② :

1. générer $u \sim \mathcal{U}_{[0,1]}$,
2. si $u < \mathcal{A}_u$, générer $v \sim \mathcal{U}_{[0,1]}$ puis calculer $x = \mu v$,
sinon, générer $v \sim \mathcal{N}(0, \sigma^2)$ puis calculer $x = |v| + \mu$.

Pour générer un échantillon x suivant la loi ③, il suffit de décaler de μ la valeur absolue d’une variable gaussienne centrée :

$$x = |y| + \mu \quad \text{où} \quad y \sim \mathcal{N}(0, \sigma^2).$$

Enfin, la méthode d’inversion de la fonction de répartition permet de simuler la loi exponentielle en générant $u \sim \mathcal{U}_{[0,1]}$, puis en calculant $x = -\ln(1 - u)/\alpha$.

Références

- [1] W.J. Cody, Rational Chebyshev approximations for the error function, *Math. Comp.* (1969) 631–638.
- [2] R.M. Corless, G.H. Gonnet, D.E.G. Hare, D.J. Jeffrey et D.E. Knuth, On the Lambert W Function, *Adv. Comp. Math.* **5** (1996) 329–359.
- [3] L. Devroye, *Non-uniform random variate generation*. Springer-Verlag, New York (1986). Disponible sur <http://jeff.cs.mcgill.ca/~luc/rnbookindex.html>.
- [4] A.E. Gelfand, A.F.M. Smith et T.-M. Lee, Bayesian Analysis of constrained parameter and truncated problems using Gibbs sampling, *J. Amer. Stat. Assoc.* **87** (1992) 523–532.
- [5] J. Geweke, Efficient Simulation from the Multivariate Normal and Student-t Distributions Subject to Linear Constraints, *Computing Science and Statistics : Proceedings of the 23rd Symposium on the Interface*, (1991) 571–578.
- [6] S. Gulam Razul, W.J. Fitzgerald et C. Andrieu, Bayesian model selection and parameter estimation of nuclear emission spectra using RJMCMC, *Nucl. Inst. Meth. A* **497** (2003) 492–510.
- [7] N.M. Haan et S.J. Godsill, Bayesian model for DNA sequencing, *ICASSP’02*, Orlando, Floride, USA, (2002) **4** 4020–4023.
- [8] S. Moussaoui, D. Brie, A. Mohammad-Djafari et C. Carteret, Separation of non-negative mixture of non-negative sources using a Bayesian approach and MCMC sampling, soumis à *IEEE Trans. Signal Process.* (2005)
- [9] V. Mazet, J. Idier et D. Brie, Déconvolution impulsionnelle positive myope, *20^e GRETSI*, Louvain-la-Neuve, Belgique, 6-9 septembre 2005.
- [10] C.P. Robert, Simulation of truncated normal variables, *Stat. Comp.* **5** (1995) 121–125.
- [11] C.P. Robert, *Méthodes de Monte Carlo par chaînes de Markov*, Economica (1996).