DURET Guillaume
BRALET Antoine

5ETI majeure IMI

Compte Rendu du TP d'Estimation

Inférence Bayésienne et Monte-Carlo Markov Chain

Année 2020-2021

I. Introduction

Ce rapport vise à rapporter les résultats des expérimentations menées en Travaux Pratique (TP) le 7 octobre 2020. L'objectif de ce TP était de mettre en place l'algorithme de Metropolis-Hastings à l'aide d'une chaine de Monte-Carlo Markov permettant de simuler une marche aléatoire. En effet, en considérant une loi cible f, on cherche ici à trouver une série $(x^{(t)})$ identiquement distribuée selon cette loi cible. Ne sachant pas mettre en place une telle série, nous utilisons alors une loi dite de proposition g dont une série identiquement distribuée est facile à obtenir et telle que pour un x donné, $g(x) \geq f(x)$. Ainsi puisque la fonction g englobe complètement la fonction f, si l'on considère la série $(x^{(t)})$ issue de g et identiquement distribuée, n'importe quelle sous-série est également identiquement distribuée dans la zone quelle occupe. C'est pourquoi si l'on ne considère que la sous-série dont les valeurs sont comprises dans la zone occupée par f, cette sous-série peut être considérée comme identiquement distribuée. La figure suivante permet d'illustrer ces propos :

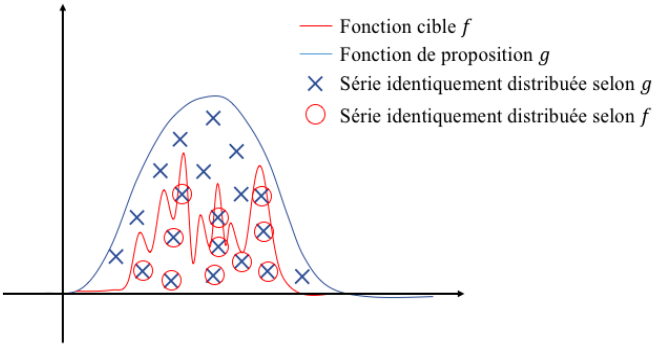


Illustration de la création d'une série identiquement distribuée selon f

Afin de créer cette série intelligemment, nous mettons en place une marche aléatoire de type Monte-Carlo Markov Chain (MCMC) qui consiste à prendre en compte la position de la série au temps t afin de pouvoir déterminer sa position au temps $t + 1$. Ainsi si notre point se trouve bien dans la zone couverte par f, le prochain point créé le sera dans ses environs. Si au contraire, il ne s'y trouve pas, alors on cherchera à faire un plus grand pas de sorte à revenir dans la zone couverte par f. De plus, l'algorithme de Metropolis-Hastings utilise un rapport d'acceptation rejet permettant de sélectionner ou non si un point est conservé pour la série identiquement distribuée selon f. Ce terme permet de ne pas rester bloquer dans des maxima locaux de la fonction f mais de bien couvrir l'ensemble de cette fonction en acceptant parfois des points aberrants qui vont permettre de découvrir potentiellement d'autres maxima.

II. Commentaires et explications algorithmiques

L'algorithme mis en place lors de ce TP se trouve essentiellement dans la fonction MetroHast.m. Néanmoins, pour pouvoir exécuter l'algorithme il faut choisir les paramètres dans le fichier mainEstimation.m et exécuter ce dernier.

Dans un premier temps nous considérons la loi BETA de paramètre a = 0.5 et b = 0.5 définie sur l'intervalle]0,1[comme étant la loi cible et une loi normale centrée réduite comme loi de proposition, ce qui se modélise par :

$$f_{BETA}(x\,;a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\, x^{a-1}\,(1-x)^{b-1} \quad \text{avec} \quad x_{cand} \sim x^t + \sigma_q \mathcal{N}(0,1)$$

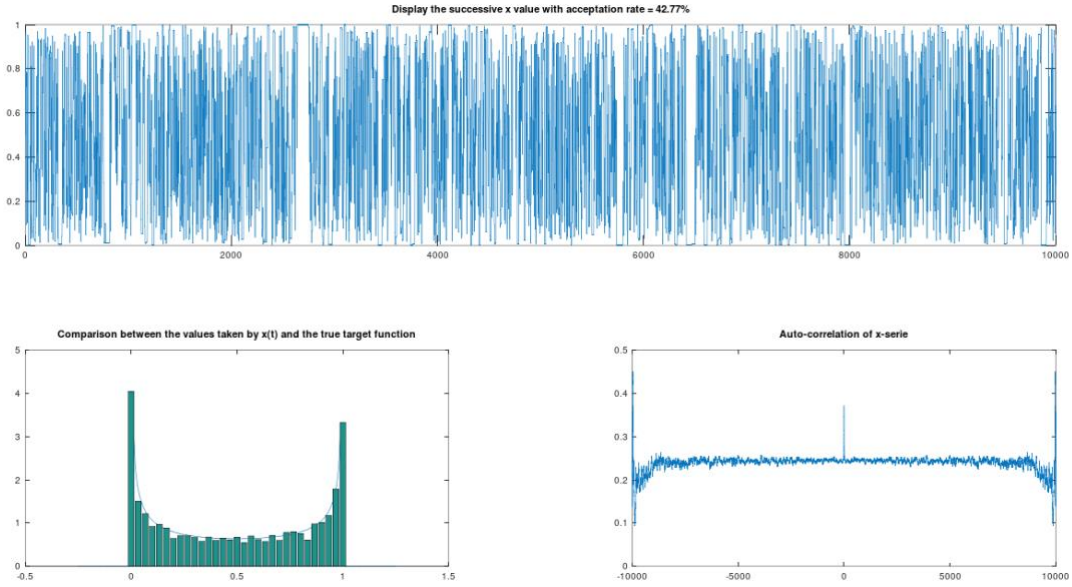
Pour obtenir les résultats de cette expérience, il est important que la variable « loiProp » soit placée à 0 et « loiTarg » doit valoir @cibleBeta.

Il est notable que lors du calcul du rapport d'acceptation-rejet :

$$\alpha(x^t, x^{cand)} = \frac{f(x^{cand})}{f(x^t)} . \frac{g(x^t|x^{cand})}{g(x^{cand}|x^t)}$$

en considérant x^t la valeur de la série au temps t et x^{cand} la valeur « candidate » pour la série au temps $t+1$, $g(x^{cand}|x^t)$ se substitue par $g(x^{cand} - x^t)$ car nous mettons en place une marche aléatoire. De plus sachant que la fonction g est symétrique, on a : $g(x^{cand} - x^t) = g(x^t - x^{cand})$.

Ainsi, en exécutant l'algorithme avec les paramètres initiaux (« sigq » = 0.5 et « T » = 10000), il est possible d'observer les figures suivantes :

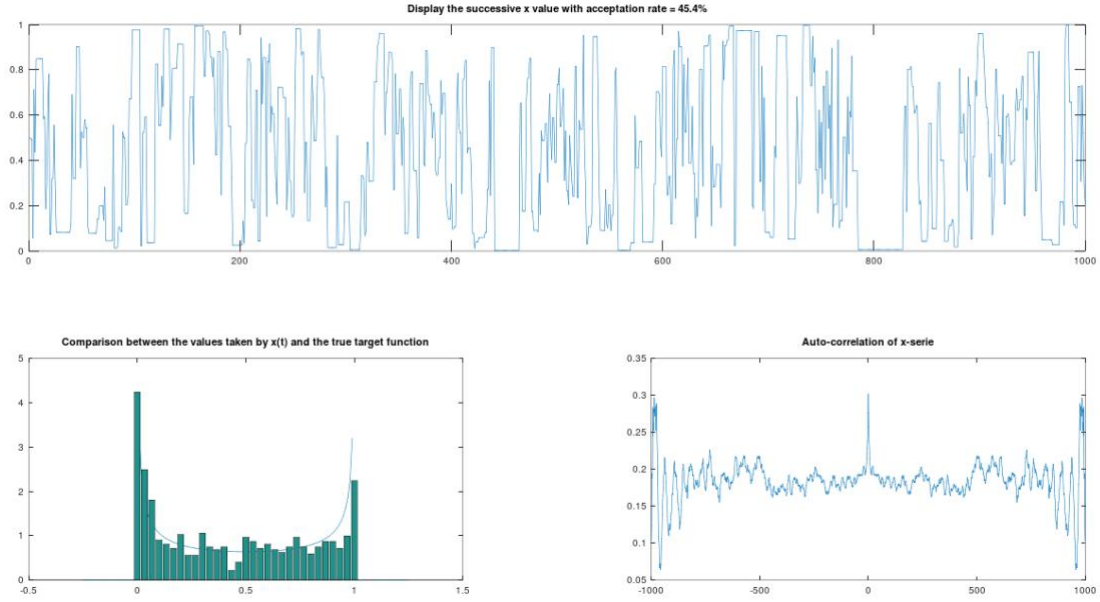


Il est possible ici de vérifier sur la figure en bas à gauche que la série créée permet bien de représenter la fonction cible statistiquement parlant. Il est également notable que le taux d'acceptation-rejet est de 42,77% ce qui signifie que l'on conserve approximativement autant de valeurs candidates que l'on en rejette. Cette balance est importante car si l'on acceptait trop de valeurs, dans ce cas, on suivrait la loi de proposition de beaucoup trop près et l'on ne se rapprocherait pas suffisamment de la loi cible. Au contraire, si l'on rejetait trop de valeurs, on risquerait d'être bloqué dans un maximum local de la fonction cible et l'on ne pourrait pas étudier l'ensemble de la fonction cible. Cette balance est donc particulièrement importante afin d'avoir une étude raisonnable et fiable de la fonction cible. Ce phénomène peut également être étudié à l'aide de la fonction d'autocorrélation de la série $(x^{(t)})$. On peut noter ici (en bas à droite de la figure) que cette fonction d'autocorrélation connait un pic significatif en 0, ce qui signifie que la série n'est pas du tout corrélée et donc que ses valeurs sont indépendantes les unes des autres. Cette indépendance est particulièrement importante car cela signifie que notre série est bien identiquement distribuée et que son évolution ne dépend en aucun cas de ses valeurs précédentes, phénomène qui permet de rassurer quant à l'étude de l'ensemble de la fonction cible et non pas seulement une partie de celle-ci. Il est important de rappeler ici que seule la partie centrale de la fonction d'autocorrélation doit être prise en compte car, l'autocorrélation étant un calcul d'intégrale de la superposition entre la série et sa translation temporelle, les premières valeurs ne contiennent que très peu d'échantillons et sont particulièrement bruitées.

III. Expérimentations et variation de paramètres

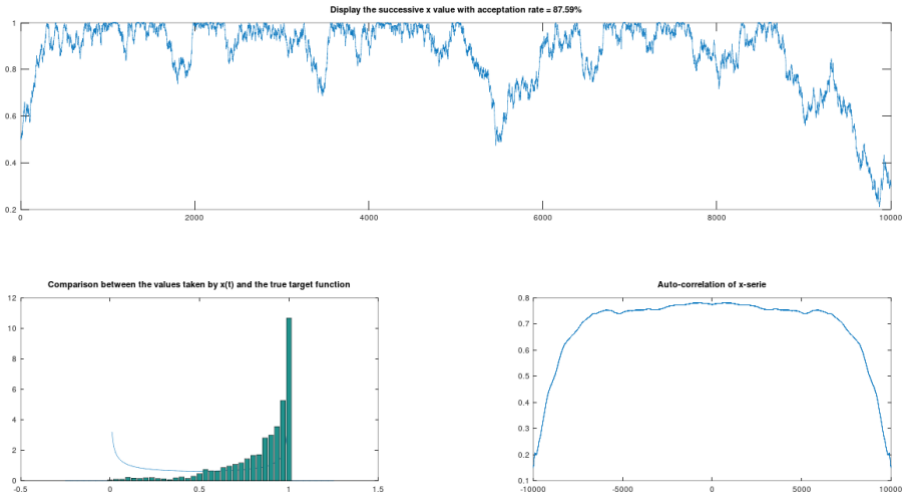
1. Modification du nombre d'échantillons

La modification de nombre d'échantillons dans la série influence directement les résultats obtenus ainsi que la précision de l'histogramme final des valeurs de la série comme en témoigne la figure ci-dessous où la variable « T » vaut 1000. On peut y constater aisément que la série a passé du temps sur le pic de gauche et n'a eu que très peu de « temps », donc d'itérations, pour étudier le pic de droite de la fonction cible. En contrepartie, plus le nombre d'échantillons est grand, plus l'algorithme sera précis mais sera lent. Le compromis temps de calcul - précision est toujours important à rechercher.

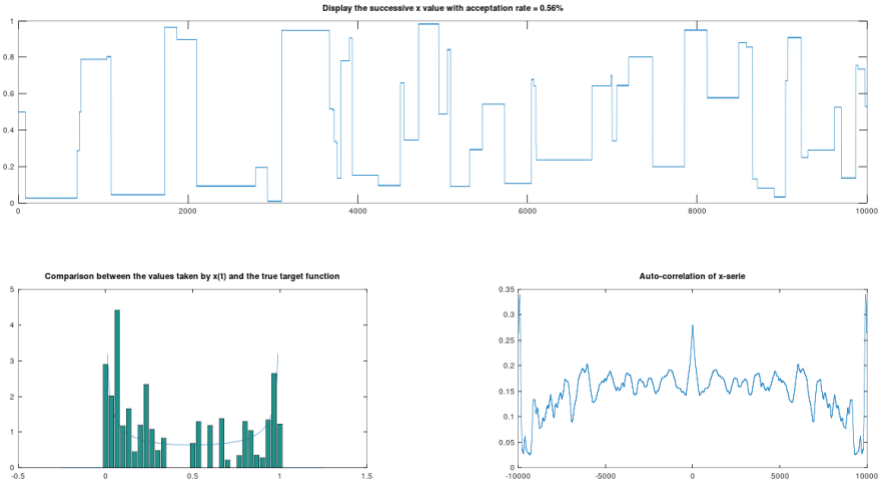


2. Modification du pas

Afin d'accélérer le processus tout en gardant une bonne précision, il est également possible de modifier un autre paramètre : le pas d'avancée. Mais une fois de plus des précautions sont à prendre avec ce paramètre « sigq » dans l'algorithme. En effet, si celui-ci est trop petit (0.001 sur la figure ci-dessous), alors on peut observer que l'algorithme aura tendance à rester toujours très proche d'un temps t à un temps $t+1$ ce qui ne permettra pas d'étudier l'ensemble de la fonction cible même avec un nombre d'échantillons très grand, ici 10000. On peut alors de plus constater que la fonction d'autocorrélation est beaucoup plus stable et statue donc sur une forte corrélation et donc dépendance d'un échantillon à l'autre de la série. En témoigne également le taux d'acceptation-rejet qui est particulièrement fort : 87.59%, ce qui signifie que la majeure partie des échantillons candidats sont acceptés et sont très proches les uns des autres (à cause du pas très faible).



Au contraire si l'on choisit un pas trop grand (50 sur la figure ci-dessous), l'on observe un tout autre phénomène : cette fois ci le taux d'acceptation rejet est de 0.56% ce qui signifie qu'une très grande majorité des échantillons candidats ont été rejetés ($\alpha = 0$ à chaque itération) et donc on observe une série très crénelée à cause de ces rejets. Par conséquent, l'autocorrélation est elle aussi beaucoup moins prononcée en 0 et beaucoup plus bruitée ce qui s'explique par cette succession d'échantillons identiques. Aussi, l'histogramme final est particulièrement perturbé car les valeurs les plus répétées ne sont pas nécessairement issues du bon fonctionnement de l'algorithme mais de la forte répétition des échantillons à cause d'un pas beaucoup trop grand.



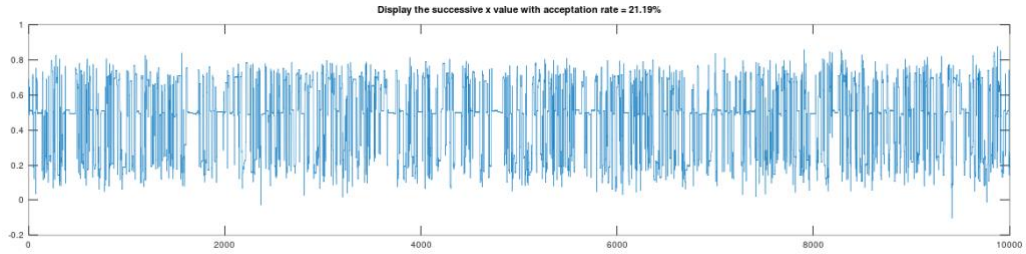
IV. Modification de loi cible et de proposition

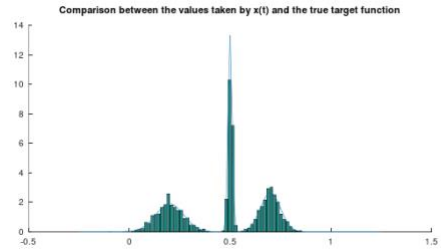
La modification de la loi cible permet de vérifier que notre algorithme est fonctionnel pour tout type de fonction cible et non pas seulement la fonction BETA précédemment mentionnée. Pour ce faire nous avons créé une fonction somme de trois gaussiennes $G(x;\ \mu, \sigma)$ de moyenne μ et d'écart-type σ :

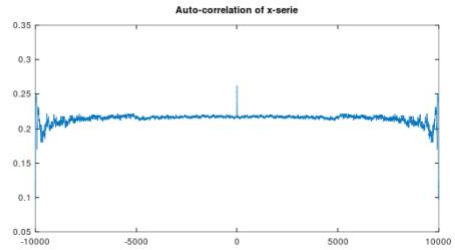
$$f(x) = \frac{1}{3}\left[\, G(x; 0.2, 0.07) + G(x; 0.5, 0.01) + G(x; 0.7, 0.05)\right]$$

Il est bien sûr possible de changer les valeurs des moyennes et écart-types des gaussiennes pour générer une nouvelle fonction à l'aide du fichier « gaussianTarget.m ».

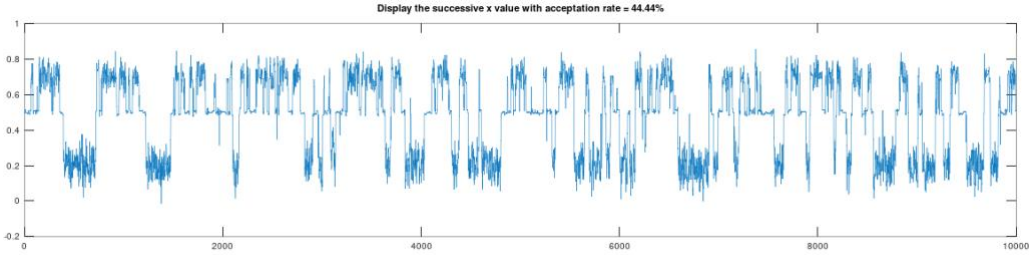
Ainsi, la modification de la loi cible (loiTarg = @gaussianTarget) en conservant 10000 échantillons et un pas de 0.5 permettent d'obtenir les résultats suivants :

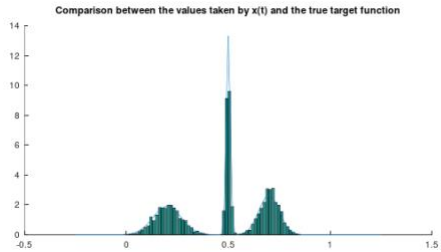


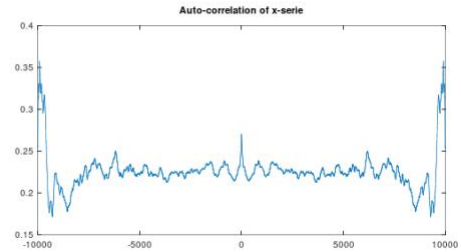




Il est alors remarquable que nous ayons ici un résultat particulièrement satisfaisant comme le montre la comparaison entre l'histogramme et la courbe théorique mais aussi la fonction d'autocorrélation qui statue définitivement sur une indépendance des valeurs. Néanmoins le taux d'acceptation-rejet semble assez faible ici (en l'occurrence 21.19%) ce qui signifierai que trop de valeurs ont été rejetée et potentiellement pas suffisamment acceptées. Pour pallier à ces rejet il est possible de réduire le pas à 0.1 et obtenir le résultat suivant :







Cette fois ci, le taux d'acceptation rejet et très similaire à celui obtenu lors de l'expérience avec la fonction BETA, à savoir 44.44% et l'on peut également constater que les valeurs de l'histogramme retracent de nouveau assez bien la courbe théorique. Néanmoins, au regard de la fonction d'autocorrélation, le résultat est beaucoup plus mitigé. En effet il semblerait que la série ne soit plus indépendante car le pic en 0 exigé est très largement rabaissé.

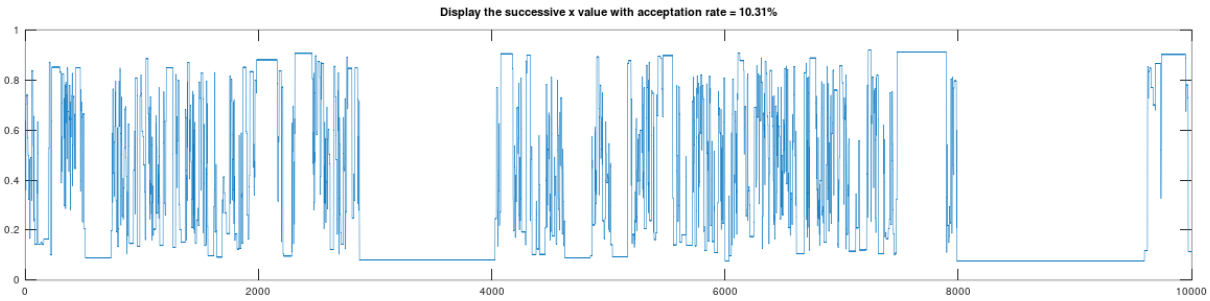
Ceci est dû au fait que notre pas est trop petit et comme cela est visible sur l'affichage de la série, elle semble assez crénelée ce qui provoque cette dépendance d'une valeur à l'autre.

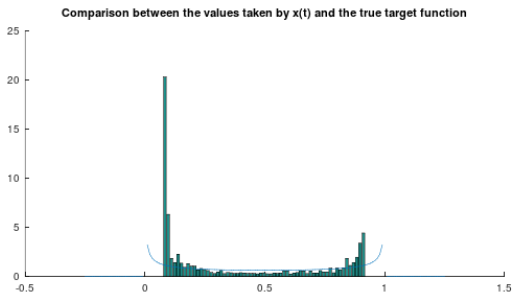
Cette expérience permet de mettre en lumière que seul le taux d'acceptation-rejet final n'est pas révélateur du comportement de l'algorithme et que celui-ci est très dépendant de la fonction cible choisie. C'est pourquoi, afin d'assurer le bon fonctionnement de la méthode, le plus sûr et de vérifier que la fonction d'autocorrélation possède un pic strict en 0, symbole d'une série dont les valeurs sont indépendantes les unes des autres. Ceci afin d'être certain que l'on a bien inspecté l'ensemble de l'intervalle de définition de la fonction.

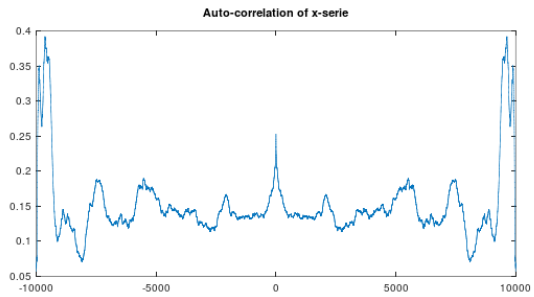
Il est également possible de choisir une autre loi de proposition afin de mettre en place la marche aléatoire. Il est à noter que se servir d'une loi uniforme aurait mené aux mêmes résultats et à des conclusions très similaires, c'est pourquoi nous n'avons pas cherché à implémenter cette loi dans ce TP. Néanmoins, nous avons essayé de mettre en place la marche aléatoire de Langevin qui se caractérise comme suit :

$$x_{cand} \sim x^t + \sigma.[\mathcal{N}(0,1) + \frac{1}{2}.\nabla \log(f(x^t))]$$

Cette loi n'est plus seulement une loi aléatoire, elle doit aussi permettre de tenir compte du gradient de telle sorte qu'à chaque sélection de candidat, l'on va chercher à se diriger dans le sens de la pente ascendante de la courbe. Ainsi, l'on devrait converger plus rapidement vers la solution finale. Après plusieurs tentatives de mise en place d'une telle marche nous n'avons pas réussi à l'implémenter. Il est cependant possible de voir dans le code les différentes sections réservées à la marche aléatoire de Langevin, de plus pour tester celle-ci et voir les résultats, il est possible de donner la valeur 1 à la variable « loiProp ». Voici ci-dessous le meilleur résultat que nous ayons pu obtenir en utilisant 10000 échantillons et un pas de 0.5.







On peut alors noter plusieurs problèmes ici :

- Une série crénelée, qui selon les expériences précédentes est dû à un pas trop grand, or en réduisant le pas le résultat empire.
- Un histogramme donc les valeurs ne couvrent pas tout l'intervalle de définition de f, ce qui est dû aux très fortes valeurs de gradient dans la loi de proposition qui une fois x^t arrivé à la borne inférieure (resp. supérieure), propose des valeurs x_{cand} aberrantes et donc sont systématiquement rejetées ($\alpha = 0$).
- Ce phénomène permet également d'expliquer l'effet créneaux ainsi que les grands pics aux bornes inférieure et supérieure.
- La fonction d'autocorrélation est également en accord avec les remarques précédentes et illustre bien la redondance de la série et donc sa dépendance aux valeurs précédentes.

Enfin, il est notable que lancer plusieurs fois l'algorithme de suite ne permet pas de retrouver des résultats similaires, au contraire, ce « meilleur » résultat est obtenu après avoir effectué plusieurs tests avec ces mêmes valeurs (de nombre d'échantillons et de pas), mais aussi avec d'autres valeurs. Il est donc certain que l'algorithme implémenté n'est pas une marche de Langevin, mais nous n'avons pas pu résoudre le problème d'implémentation.

V. Conclusion

Suite à cette étude il est alors possible de conclure en plusieurs points et en premier lieu au fait que l'algorithme de Metropolis-Hastings est particulièrement performant quand il s'agit de trouver une série de points identiquement distribuée selon une fonction dont il n'est pas aisé de générer une telle série. Néanmoins, il faut veiller au bon paramétrage de l'algorithme afin que sa convergence vise bien la fonction cible. En effet, il peut être très aisé de tomber dans un maximum local et de ne pas pouvoir en sortir à cause d'un pas trop petit, ou à l'inverse de ne proposer que des valeurs candidates aberrantes et donc d'avoir une répétition trop importante des valeurs de l'algorithme.

Il semble important de noter que le problème rencontré pour la mise en place de la marche aléatoire de Langevin est purement algorithmique, en effet en pratique, cette marche aléatoire devrait permettre d'améliorer sensiblement la précision et la vitesse de convergence de l'algorithme de Metropolis-Hastings.