Un petit tutoriel sur les méthodes de simulation stochastique pilotées par l'optimisation

Marcelo PEREYRA[1], Jean-Yves TOURNERET[2]

[1] Maxwell Institute for Mathematical Sciences & School of Mathematical and Computer Sciences
Heriot-Watt University, EH14 4AS, UK

[2]Université de Toulouse, INP-ENSEEIHT/IRIT/TéSA
ENSEEIHT, 2 Rue Camichel, France

m.pereyra@hw.ac.uk, jean-yves.tourneret@enseeiht.fr

Résumé – Devant l'intérêt croissant pour l'analyse de données en grande dimension et le "big data", il est intéressant de disposer de méthodes de simulation stochastique qui permettent de générer efficacement des vecteurs de grande dimension distribués suivant une loi a posteriori d'intérêt. Les vecteurs ainsi générés permettent par exemple de construire des estimateurs Bayésiens associés à des modèles statistiques très évolués. Cet article résume les principales méthodes de simulation stochastique qui se sont inspirées d'outils issus de la théorie de l'optimisation. Ces méthodes sont par exemple basées sur la discrétisation d'équations différentielles stochastiques issues d'un processus de diffusion de Langevin ou de dynamiques Hamiltoniennes. Elles peuvent également utiliser des outils puissants d'optimisation tels que les opérateurs proximaux ou des itérations de gradient conjugué.

Abstract – Stochastic simulation methods are efficient tools that allow samples to be generated according to a distribution of interest such as a posterior distribution. The generated samples can then be used to build Bayesian estimators of the unknown parameters of a statistical model. In order to respond to the recent increasing interest for high-dimensional data or big data, it is interesting to develop efficient stochastic simulation methods that can be used to sample high-dimensional distributions. This paper summarizes some methods that have inspired by results from optimization theory. Some of these methods are based on the discretization of stochastic differential equations resulting from a Langevin diffusion process or from Hamiltonian dynamics. Other methods use optimization tools such as proximal operators or conjugate gradient iterations.

1 Introduction

Les méthodes Bayésiennes ont montré leur intérêt dans de nombreuses applications du traitement du signal et des images. Dans un contexte d'estimation statistique, ces méthodes exploitent les informations contenues dans les données, exprimées via un terme de vraisemblance, et les connaissances a priori disponibles sur les paramètres inconnus d'un modèle statistique, pour déterminer la loi a posteriori de ce modèle qui est l'élément clé de toute inférence Bayésienne. Plus précisément, on note $\boldsymbol{x} = [x_1, \ldots, x_N]^T$ le vecteur inconnu du modèle statistique considéré, $\boldsymbol{y} = [y_1, \ldots, y_M]^T$ le vecteur des observations associé à $\boldsymbol{x}$ et $p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})$ la vraisemblance du modèle statistique considéré, où $\boldsymbol{\theta}$ résume les paramètres déterministes du modèle. Dans toute inférence Bayésienne, le vecteur $\boldsymbol{x}$ est muni d'une loi a priori notée $p(\boldsymbol{x};\boldsymbol{\theta})$, qui permet de déterminer la loi a posteriori de $\boldsymbol{x}$ sachant $\boldsymbol{y}$ notée $\pi(\boldsymbol{x})$ et définie par

$$\pi(\boldsymbol{x}) \triangleq p(\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\theta}) = \frac{p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})p(\boldsymbol{x};\boldsymbol{\theta})}{p(\boldsymbol{y};\boldsymbol{\theta})} \tag{1}$$

où la vraisemblance marginale

$$p(\boldsymbol{y};\boldsymbol{\theta}) = \int p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})p(\boldsymbol{x};\boldsymbol{\theta})d\boldsymbol{x} \tag{2}$$

est classiquement appelée "évidence" du modèle. Dans de nombreuses applications, il est important de pouvoir déterminer cette loi a posteriori $p(\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\theta})$ ou de construire des estimateurs de $\boldsymbol{x}$ basés par exemple sur la moyenne ou sur le maximum de cette loi a posteriori.

Il existe un grand nombre d'algorithmes d'échantillonnage qui permettent de générer des vecteurs distribués suivant une loi d'intérêt. Ces algorithmes sont par exemple basés sur la méthode de Metropolis-Hastings (MH) ou sur un cas particulier de cette méthode appelée échantillonneur de Gibbs (le lecteur pourra se référer aux chapitres 7 et 8 de [1] pour une description complète de ces méthodes). Malheureusement, la plupart de ces algorithmes perdent leur efficacité lorsque $\boldsymbol{x}$ est de grande dimension. Pour faire face à ce problème, des algorithmes efficaces d'échantillonnage ont été récemment proposés s'inspirant d'outils issus de la théorie de l'optimisation. Cet article résume les plus populaires de ces algorithmes qui sont basés sur la discrétisation d'équations différentielles stochastiques issues d'un processus de diffusion de Langevin ou de dynamiques Hamiltoniennes, ou utilisent des outils issus de la théorie de l'optimisation tels que les opérateurs proximaux ou des itérations de gradient conjugué. Il est organisé en quatre parties s'intéressant respectivement aux algorithmes de

type Langevin MCMC (Section 2), aux méthodes MCMC Hamiltoniennes (Section 3), aux algorithmes MCMC proximaux (Section 4) et aux méthodes de simulation adaptées aux lois Gaussiennes en grande dimension (Section 5).

2 Algorithmes de type Langevin

L'algorithme MALA (pour "Metropolis adjusted Langevin algorithm") est un algorithm évolué de type MH qui s'inspire d'un processus de diffusion défini sur $\mathbb{R}^N$ défini comme la solution de l'équation différentielle stochastique (EDS) suivante [2]

$$dX(t) = \frac{1}{2}\nabla \log \pi\left[X(t)\right] dt + dW(t), \quad X(0) = \boldsymbol{x}_0 \quad (3)$$

où W un mouvement Brownien défini sur $\mathbb{R}^N$, $\boldsymbol{x}_0 \in \mathbb{R}^N$ dénote la condition initiale de cette EDS et ∇ désigne l'opérateur gradient. Sous certaines conditions de stabilité, la solution de l'EDS (3) $X(t)$ converge en loi vers π lorsque $t \to \infty$, et peut donc définir un schéma de simulation intéressant pour générer des vecteurs de loi π. Puisque la simulation directe de $X(t)$ suivant (3) n'est possible que dans des cas très particuliers, on peut discrétiser l'EDS (3), par exemple à l'aide d'une méthode d'Euler, ce qui conduit à la relation suivante

$$X^{(t+1)} \sim \mathcal{N}\left(X^{(t)} + \frac{\delta}{2}\nabla \log \pi\left(X^{(t)}\right), \delta \mathbb{I}_N\right) \quad (4)$$

où le paramètre δ est un pas de discrétisation. Sous certaines conditions sur π et δ, (4) produit une bonne approximation de $X(t)$ qui converge vers une loi stationnaire proche de π [3]. Dans l'algorithme MALA, l'erreur d'approximation est corrigée en introduisant une étape d'acceptation-rejet qui garantit la convergence de l'algorithme vers la densité cible d'intérêt π (voir [4] pour plus de détails).

La rapidité de convergence de l'algorithme MALA peut être accélérée de manière significative en remplaçant δ par une matrice judicieusement choisie notée $\boldsymbol{\Sigma}(\boldsymbol{x})$, qui dépend en particulier de l'état courant de la chaîne de Markov et capture les corrélations locales de π. La matrice $\boldsymbol{\Sigma}(\boldsymbol{x})$ peut être construite en exprimant la dynamique de Langevin sur une variété Riemannienne comme dans [5]. Dans ces algorithmes MALA *Riemanniens*, le choix de $\boldsymbol{\Sigma}(\boldsymbol{x})$ peut se ramener à la définition d'une distance ou métrique pour l'espace des paramètres d'intérêt [5]. Comme les notions de gradients Riemannien et Euclidiens sont liées par la relation $\tilde{\nabla} g(\boldsymbol{x}) = \boldsymbol{\Sigma}(\boldsymbol{x})\nabla g(\boldsymbol{x})$, on observe que ce problème est très étroitement lié au problème de pré-conditionnement du gradient en optimisation. Comme en optimisation, des choix standards de matrices $\boldsymbol{\Sigma}$ peuvent être obtenus. On peut par exemple choisir $\boldsymbol{\Sigma}$ comme une matrice définie-positive construite à partir de l'inverse de la matrice Hessienne en s'inspirant de l'algorithme BFGS [6] ou en utilisant la transformation SoftAbs [7]. Une autre stratégie consiste à utiliser comme matrice $\boldsymbol{\Sigma}(\boldsymbol{x})$ l'inverse de la matrice d'information de Fisher [5], qui est la métrique "naturelle" d'un point de vue de la théorie de l'information et est en lien avec la méthode de descente de gradient naturel [8]. Puisque l'utilisation exacte de la Hessienne ou de la matrice d'information de Fisher est généralement trop couteuse, un challenge dans la communauté du traitement du signal a été de trouver des approximations qui restent efficaces en grande dimension (voir [9, 10] pour quelques travaux récents). Une autre stratégie consiste à optimiser $\boldsymbol{\Sigma}$ en ligne, ce qui conduit naturellement à des versions adaptatives de l'algorithme MALA. Algorithmiquement, cela peut s'effectuer en interprétant l'algorithme comme une méthode de descente de gradient stochastique, ce qui montre une autre forme de synergie entre les méthodes de simulation stochastique et les algorithmes d'optimisation déterministes.

3 Méthodes MCMC Hamiltoniennes

La méthode MCMC Hamiltonienne (appelée méthode HMC dans cet article) résulte de l'application de l'algorithme MH à un problème défini à l'aide d'un vecteur de variables auxiliaires [11]. A partir d'un vecteur $\boldsymbol{w} \in \mathbb{R}^N$ et d'une matrice $\boldsymbol{\Sigma} \in \mathbb{R}^{N\times N}$ définie positive, considérons la loi augmentée $\pi(\boldsymbol{x}, \boldsymbol{w}) \propto \pi(\boldsymbol{x}) \exp(-\frac{1}{2}\boldsymbol{w}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{w})$, qui admet la loi cible d'intérêt $\pi(\boldsymbol{x})$ comme loi marginale. La méthode HMC est basée sur le fait que les trajectoires définies par les "dynamiques Hamiltoniennes" conservent les lignes de niveaux de $\pi(\boldsymbol{x}, \boldsymbol{w})$. Un point $(\boldsymbol{x}_0, \boldsymbol{w}_0) \in \mathbb{R}^{2N}$ qui évolue selon les équations différentielles (5) pendant un temps de simulation $(0, t]$

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} &= -\nabla_{\boldsymbol{w}} \log \pi(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{\Sigma}^{-1}\boldsymbol{w} \\ \frac{\mathrm{d}\boldsymbol{w}}{\mathrm{d}t} &= \nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{x}, \boldsymbol{w}) = \nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{x}) \end{aligned} \quad (5)$$

fournit alors un point $(\boldsymbol{x}_t, \boldsymbol{w}_t)$ tel que $\pi(\boldsymbol{x}_t, \boldsymbol{w}_t) = \pi(\boldsymbol{x}_0, \boldsymbol{w}_0)$. En d'autres termes, une loi de proposition vérifiant (5) admet $\pi(\boldsymbol{x}, \boldsymbol{w})$ comme loi invariante. En exploitant cette propriété pour la simulation, la méthode HMC combine (5) avec une étape d'échantillonnage aléatoire, $\boldsymbol{w} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$, qui admet également comme loi invariante $\pi(\boldsymbol{x}, \boldsymbol{w})$, et produit donc une chaîne de Markov ergodique. Finalement, comme pour la diffusion de Langevin (3), il n'est généralement pas possible de résoudre les équations Hamiltoniennes (5) exactement. On utilise alors une approximation comme l'approximation *leap-frog* (détaillée dans [11]) qui permet d'obtenir le schéma de simulation suivant

$$\begin{aligned} \boldsymbol{w}^{(t+\delta/2)} &= \boldsymbol{w}^{(t)} + \frac{\delta}{2}\nabla_{\boldsymbol{x}} \log \pi\left(\boldsymbol{x}^{(t)}\right) \\ \boldsymbol{x}^{(t+\delta)} &= \boldsymbol{x}^{(t)} + \delta\boldsymbol{\Sigma}^{-1}\boldsymbol{w}^{(t+\delta/2)} \\ \boldsymbol{w}^{(t+\delta)} &= \boldsymbol{w}^{(t+\delta/2)} + \frac{\delta}{2}\nabla_{\boldsymbol{x}} \log \pi\left(\boldsymbol{x}^{(t+\delta)}\right) \end{aligned} \quad (6)$$

où le paramètre δ permet à nouveau de contrôler le pas de discrétisation. L'erreur d'approximation introduite par l'approximation (6) est ensuite corrigée par une étape de MH assurant que $\pi(\boldsymbol{x}, \boldsymbol{w})$ est la loi d'intérêt de la chaine de Markov. Pour obtenir des vecteurs distribués suivant la loi marginale $\pi(\boldsymbol{x})$, il suffit de projeter l'état augmenté $(\boldsymbol{x}^{(t)}, \boldsymbol{w}^{(t)})$ sur l'espace d'état d'origine en supprimant le vecteur $\boldsymbol{w}^{(t)}$. Cette méthode est résumée dans l'algorithme 1 ci-dessous.

Algorithme 1 : Méthode de Monte Carlo Hamiltonienne (avec leap-frog)

Choisir une valeur initiale pour $\boldsymbol{x}^{(0)}$, $\delta > 0$ et $L \in \mathbb{N}$.
for $t = 1$ to T **do**
  Mettre à jour la variable auxiliaire $\boldsymbol{w} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$.
  Générer un candidat $(\boldsymbol{x}^*, \boldsymbol{w}^*)$ en propageant l'état courant $(\boldsymbol{x}^{(t-1)}, \boldsymbol{w})$ avec L étapes de leap-frog de longueur δ définie dans (6).
  Calculer la probabilité d'acceptation

$$\rho^{(t)} = \min\left(1, \frac{\pi(\boldsymbol{x}^*, \boldsymbol{w}^*)}{\pi(\boldsymbol{x}^{(t-1)}, \boldsymbol{w})}\right).$$

  Générer $u_t \sim \mathcal{U}(0, 1)$.
  if $u_t \leq \rho^{(t)}$ **then**
    Accepter le candidat et poser $\boldsymbol{x}^{(t)} = \boldsymbol{x}^*$.
  else
    Rejeter le candidat et poser $\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t-1)}$.
  end if
end for
Sortie $\{\boldsymbol{x}^{(t)}\}_{t=1}^T$.

On peut noter que les développements effectués pour l'algorithme MALA s'appliquent aussi à la méthode HMC (l'algorithme MALA est équivalent à une méthode HMC avec une étape de leap-frog par itération). En particulier, si on exprime la dynamique Hamiltonienne dans une variété de Riemann, on obtient un algorithme HMC *Riemmanien*, où δ est remplacé par une matrice $\boldsymbol{\Sigma}(\boldsymbol{x})$ qui dépend de l'état courant $\boldsymbol{x}$ et qui peut améliorer la rapidité de convergence de manière significative [5]. De manière similaire, il y a des méthodes HMC adaptatives qui utilisent des pas de gradient stochastique pour optimiser les paramètres de manière séquentielle [11].

4 Algorithmes MCMC proximaux

Les algorithmes MCMC tels que l'algorithme MALA ou les méthodes HMC reposent sur des outils d'analyse différentielle qui permettent de se déplacer efficacement dans des espaces de grande dimension. Ces stratégies de simulation peuvent s'avérer inefficaces si la loi π n'est pas suffisamment régulière, par exemple si π n'est pas Lipchitzienne et continûment différentiable. En effet, les approximations à temps discret utilisées pour construire la chaîne de Markov de l'algorithme MCMC peuvent dans ce cas devenir divergentes. Ce problème interdit l'utilisation de l'algorithme MALA ou des méthodes HMC lorsque la fonction objectif n'est pas différentiable ou lorsque le vecteur $\boldsymbol{x}$ est soumis à des contraintes, par exemple de positivité, ou de norme inférieure à une valeur donnée.

Il a été montré récemment que l'algorithme MALA ou les méthodes HMC peuvent être utilisés pour des modèles convexes non lisses en exploitant des outils d'optimisation convexe comme les opérateurs proximaux ou les envelopes de Moreau-Yoshida. Par exemple, une approximation proximale du processus de diffusion de Langevin (3) définie par [1]

$$X^{(t+1)} \sim \mathcal{N}\left(\mathrm{prox}_{\frac{-\delta}{2}\log\pi}\left(X^{(t)}\right), \delta\mathbb{I}_N\right) \tag{7}$$

a été récemment proposée dans [14] comme une alternative à l'approximation d'Euler

$$X^{(t+1)} \sim \mathcal{N}\left(X^{(t)} + \frac{\delta}{2}\nabla\log\pi\left(X^{(t)}\right), \delta\mathbb{I}_n\right)$$

utilisée dans l'algorithme MALA (d'autres versions plus avancées de (7) utilisent des techniques de splitting permettant de gérer des éléments lisses et non-lisses séparément, voir [15]). Il a été établi dans [14, 15] que si π est log-concave, (7) correspond à une discrétisation de (3) remarquablement stable avec des propriétés de convergence très intéressantes. De plus, l'algorithme MALA "proximal" résultant de l'utilisation de (7) avec une étape de MH possède également des propriétés de convergence remarquables [14] par rapport à d'autres méthodes MALA ou HMC. On peut noter que l'utilisation directe de (7) sans étape de MH permet généralement d'obtenir des algorithmes très efficaces qui sont uniquement limités par la présence d'un biais asymptotique [15]. Enfin, l'utilisation d'opérateurs proximaux dans les méthodes HMC a fait l'objet de travaux récents pour l'échantillonnage de densités log-concaves non continument différentiables [16]. Les expérimentations effectuées dans [16] montrent que l'utilisation d'opérateurs proximaux dans des algorithmes MCMC peut être très efficace, en particulier dans des applications où l'espace d'état est de grande dimension et où les lois a priori sont non lisses. Analyser les propriétés de convergence de cet algorithme est un problème intéressant qui mériterait une étude spécifique.

5 Lois gaussiennes en grande dimension

L'approche standard pour simuler suivant une loi gaussienne multivariée de matrice de précision $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ est d'effectuer une factorisation de Cholesky de cette matrice sous la forme $\boldsymbol{Q} = \boldsymbol{L}^T\boldsymbol{L}$, de générer un vecteur gaussien auxiliaire $\boldsymbol{w} \sim \mathcal{N}(0, \mathbb{I}_N)$ et d'obtenir un vecteur $\boldsymbol{x}$ de loi $\mathcal{N}(0, \boldsymbol{Q})$ par résolution du système linéaire $\boldsymbol{L}\boldsymbol{x} = \boldsymbol{w}$ [17]. La complexité calculatoire de cette approche est cependant trop couteuse pour la plupart des problèmes de grande dimension (on peut cependant noter qu'il y a des cas spécifiques qui peuvent se traiter comme lorsque $\boldsymbol{Q}$ est Toeplitz [18], circulante [19] ou parcimonieuse [17]).

Les méthodes de simulation de lois gaussiennes multivariées inspirées de méthodes d'optimisation utilisent la propriété qu'un vecteur gaussien peut être obtenu par minimisation d'une fonction de coût judicieusement choisie [20, 21]. Considérons un modèle Bayésien avec une vraisemblance gaussienne $\boldsymbol{y}|\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{H}\boldsymbol{x}, \boldsymbol{\Sigma}_{\boldsymbol{y}})$ et une loi a priori $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{x}_0, \boldsymbol{\Sigma}_{\boldsymbol{x}})$, où $\boldsymbol{H} \in \mathbb{R}^{N\times M}$ est un un opérateur linéaire, la moyenne de la loi a priori est $\boldsymbol{x}_0 \in \mathbb{R}^N$, et $\boldsymbol{\Sigma}_{\boldsymbol{x}} \in \mathbb{R}^{N\times N}$, $\boldsymbol{\Sigma}_{\boldsymbol{y}} \in \mathbb{R}^{M\times M}$ sont deux

1. $\mathrm{prox}_{\varphi}(\boldsymbol{v})$ désigne l'opérateur proximal de φ évalué au point $\boldsymbol{v} \in \mathbb{R}^N$ [12, 13].

matrices de covariance définies positives. Il est facile de montrer que la loi a posteriori $p(\boldsymbol{x}|\boldsymbol{y})$ est aussi gaussienne de vecteur moyenne $\boldsymbol{\mu} \in \mathbb{R}^N$ et de matrice de précision $\boldsymbol{Q} \in \mathbb{R}^{N\times N}$ définis par

$$\begin{aligned}\boldsymbol{Q} &= \boldsymbol{H}^T\boldsymbol{\Sigma}_{\boldsymbol{y}}^{-1}\boldsymbol{H} + \boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1}\\ \boldsymbol{\mu} &= \boldsymbol{Q}^{-1}\left(\boldsymbol{H}^T\boldsymbol{\Sigma}_{\boldsymbol{y}}^{-1}\boldsymbol{y} + \boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1}\boldsymbol{x}_0\right).\end{aligned}$$

Simuler suivant la loi a posteriori de $\boldsymbol{x}|\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{Q}^{-1})$ par factorisation de Cholesky de $\boldsymbol{Q}$ peut se révéler couteux quand N est grand. Certaines méthodes s'inspirant de l'optimisation cherchent à résoudre le problème de minimisation "aléatoire"

$$\begin{aligned}\boldsymbol{x} = \underset{\boldsymbol{u}\in\mathbb{R}^N}{\operatorname{argmin}}\ &\left(\boldsymbol{w}_1 - \boldsymbol{H}\boldsymbol{u}\right)^T\boldsymbol{\Sigma}_{\boldsymbol{y}}^{-1}\left(\boldsymbol{w}_1 - \boldsymbol{H}\boldsymbol{u}\right)\\ &+ \left(\boldsymbol{w}_2 - \boldsymbol{u}\right)^T\boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1}\left(\boldsymbol{w}_2 - \boldsymbol{u}\right)\end{aligned} \tag{8}$$

avec $\boldsymbol{w}_1 \sim \mathcal{N}(\boldsymbol{y}, \boldsymbol{\Sigma}_y)$ et $\boldsymbol{w}_2 \sim \mathcal{N}(\boldsymbol{x}_0, \boldsymbol{\Sigma}_x)$. Il est facile de vérifier que si (8) est résolu de manière exacte, alors $\boldsymbol{x}$ est un vecteur distribué suivant la loi a posteriori $p(\boldsymbol{x}|\boldsymbol{y})$. D'un point de vue complexité calculatoire, il est cependant beaucoup plus simple de résoudre (8) approximativement, par exemple en utilisant quelques itérations de gradient conjugué [21]. L'erreur d'approximation peut alors être compensée en utilisant une étape de MH [22], au prix de l'introduction d'une certaine corrélation entre les vecteurs générés par la chaine, et donc d'une diminution de la taille effective de l'échantillon généré par la chaine. Heureusement il y a une manière élégante de déterminer automatiquement le nombre optimal d'itérations de gradient conjugué qui maximise l'efficacité de l'algorithme [22].

6 Conclusions

Cet article a résumé les principales méthodes de simulation stochastiques pilotées par l'optimisation qui peuvent s'organiser en quatre grandes classes : les méthodes de type Langevin MCMC, les méthodes MCMC Hamiltoniennes, les méthodes MCMC proximales et les méthodes spécifiquement adaptées aux lois normales en grande dimension. Nous renvoyons le lecteur à [4] pour un exposé plus détaillé de ces méthodes et pour quelques travaux futurs qui nous paraissent prometteurs. Nous pensons que l'optimisation jouera un rôle de plus en plus important pour la construction de méthodes de simulation stochastique adaptées aux problèmes de grande dimension.

Références

[1] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, 2nd ed. New York : Springer, 2004.

[2] G. Roberts and R. Tweedie, "Exponential convergence of Langevin distributions and their discrete approximations," *Bernoulli*, vol. 2, no. 4, pp. 341–363, 1996.

[3] A. Durmus and E. Moulines, "Non-asymptotic convergence analysis for the unadjusted langevin algorithm," arXiv, Accepted for publication in Ann. Appl. Probab. 1507.05021, July 2015. [Online]. Available : http ://arxiv.org/pdf/1507.05021v1.pdf

[4] M. Pereyra, P. Schniter, E. Chouzenoux, J.-C. Pesquet, J.-Y. Tourneret, A. Hero, and S. Mclaughlin, "A survey of stochastic simulation and optimization methods in signal processing," *IEEE. J. Selected Topics in Signal Process.*, vol. 10, no. 2, pp. 224–241, Mar. 2016.

[5] M. Girolami and B. Calderhead, "Riemann manifold Langevin and Hamiltonian Monte Carlo methods," *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, vol. 73, pp. 123–214, 2011.

[6] Y. Zhang and C. A. Sutton, "Quasi-Newton methods for Markov chain Monte Carlo," in *Advances in Neural Information Processing Systems 24 (NIPS 2011)*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds., 2011, pp. 2393–2401.

[7] M. Betancourt, "A general metric for Riemannian manifold Hamiltonian Monte Carlo," in *National Conference on the Geometric Science of Information*, ser. Lecture Notes in Computer Science 8085, F. Nielsen and F. Barbaresco, Eds. Springer, 2013, pp. 327–334.

[8] S.-I. Amari, "Natural gradient works efficiently in learning," *Neural Comput.*, vol. 10, no. 2, pp. 251–276, Feb. 1998.

[9] S. Allassonniere and E. Kuhn, "Convergent Stochastic Expectation Maximization algorithm with efficient sampling in high dimension. Application to deformable template model estimation," *ArXiv e-prints*, Jul. 2012.

[10] Y. Marnissi, A. Benazza-Benyahia, E. Chouzenoux, and J.-C. Pesquet, "Majorize-minimize adapted Metropolis Hastings algorithm. application to multichannel image recovery," in *Proc. European Signal Process. Conf. (EUSIPCO 2014)*, Lisbon, Portugal, Sep. 1-5 2014, pp. 1332–1336.

[11] R. Neal, "MCMC using Hamiltonian dynamics," in *Handbook of Markov Chain Monte Carlo*, S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, Eds. Chapman & Hall/CRC Press, 2013, pp. 113–162.

[12] J. J. Moreau, "Fonctions convexes duales et points proximaux dans un espace hilbertien," *C. R. Acad. Sci. Paris Sér. A*, vol. 255, pp. 2897–2899, 1962.

[13] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, H. H. Bauschke, R. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, Eds. New York : Springer-Verlag, 2010, pp. 185–212.

[14] M. Pereyra, "Proximal Markov chain Monte Carlo algorithms," *Statistics and Computing*, 2015, in press.

[15] A. Durmus, E. Moulines, and M. Pereyra, "Efficient Bayesian computation by proximal Markov chain Monte Carlo : when Langevin meets Moreau," *ArXiv e-prints*, Dec. 2016.

[16] L. Chaari, J.-Y. Tourneret, C. Chaux, and H. Batatia, "A Hamiltonian Monte Carlo method for non-smooth energy sampling," *IEEE. Trans. Signal Process.*, vol. 64, no. 21, pp. 5585–5594, 2016.

[17] H. Rue, "Fast sampling of Gaussian Markov random fields," *J. Royal Statist. Society Series B*, vol. 63, no. 2, pp. 325–338, 2001.

[18] W. F. Trench, "An algorithm for the inversion of finite Toeplitz matrices," *J. Soc. Ind. Appl. Math.*, vol. 12, no. 3, pp. 515–522, 1964.

[19] D. Geman and C. Yang, "Nonlinear image recovery with half-quadratic regularization," *IEEE Trans. Image Process.*, vol. 4, no. 7, pp. 932–946, Jul. 1995.

[20] G. Papandreou and A. L. Yuille, "Gaussian sampling by local perturbations," in *Advances in Neural Information Processing Systems 23 (NIPS 2010)*, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds., 2010, pp. 1858–1866.

[21] F. Orieux, O. Feron, and J.-F. Giovannelli, "Sampling high-dimensional Gaussian distributions for general linear inverse problems," *IEEE Signal Process. Lett.*, vol. 19, no. 5, pp. 251–254, May 2012.

[22] C. Gilavert, S. Moussaoui, and J. Idier, "Efficient Gaussian sampling for solving large-scale inverse problems using MCMC," *IEEE Trans. Signal Process.*, vol. 63, no. 1, pp. 70–80, Jan. 2015.