Accelerating MCMC for imaging science by using an implicit Langevin algorithm

Teresa Klatzer[1,4]

with

P. Dobson[1,4], K. Zygalakis[1,4], M. Pereyra[2,4], Y. Altmann[2], J.-M. Sanz-Serna[3]

[1]University of Edinburgh [2]Heriot-Watt University [3]Universidad Carlos III de Madrid

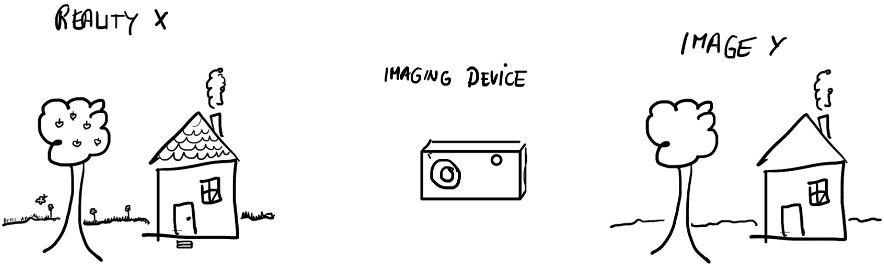
[4]Maxwell Institute for Mathematical Sciences

Contributed talks: CT07

Overview

Overview

Inverse problem in imaging



- Estimate the unknown image $x \in \mathbb{R}^d$ from an observation $y \in \mathbb{R}^p$
- Forward statistical model: $y \sim \mathcal{N}(Ax, \sigma^2 \mathbb{I}_p)$ or $y \sim \mathcal{P}(Ax, \beta)$
- Observation operator $A \in \mathbb{R}^{p \times d}$ is often rank-deficient or $A^T A$ has a poor condition number $\kappa \gg 1$
- Introduce regularisation $\rightarrow$ well-posed inverse problem

$$x^* = \underset{x \in \mathbb{R}^d}{\operatorname{argmin}} \, f_y(Ax) + \lambda g(x)$$

Bayesian methodology for solving inverse problems

Using a prior distribution $p(x)$ for regularisation and the likelihood $p(y|x)$ we can define the posterior distribution

$$\pi(x) := p(x|y) = \frac{p(y|x)p(x)}{\int_{\mathbb{R}^d} p(y|x)p(x)dx}$$

We consider the unnormalised posterior

$$\pi(x) \propto p(y|x)p(x) = e^{-U(x)} = e^{-f_y(x)-g(x)}$$

Options for regularisation g

- Assumption-driven (e.g. TV, TGV, ...)
- Data-driven (e.g. convex neural networks)

Goals for this work

- We want to use Bayesian methodology for scientific imaging applications where **uncertainty quantification** is required
- E.g. in low photon imaging, challenging noise and limited ground truth available
- Design an algorithm that converges at the order of $\sqrt{\kappa}$ iterations similar to **accelerated** optimisation schemes

Requirements

We require convexity of f_y and g (and therefore log-concavity of $\pi(x)$), because

1. Convexity guarantees the well-posedness of $\pi(x) := p(x|y)$ and of posterior moments of interest
2. Convexity allows convergence analysis of the Bayesian computation methods: faster convergence rates, tighter non-asymptotic bounds, and **stronger guarantees** on the delivered solutions.

Example: Poisson deconvolution

The posterior distribution $\pi(x)$ is modelled using a Poisson likelihood $y \sim \mathcal{P}(Ax, \beta)$ and a Moreau-Yosida smoothed TV regulariser (see Melidonis et al. (2023)):

$$\pi^{\lambda}(x) \propto \exp\left(-\sum_{i=1}^{d} \left[(Ax)_i + \beta - y_i \log((Ax)_i + \beta) + \iota_{(Ax)_i \geq 0} \right] \right.$$
$$\left. -\theta_{\mathrm{TV}^{\lambda}} \mathrm{TV}^{\lambda}(x) \right) .$$

Figure: True image x and observation y, MIV=10 Poisson noise and 5×5 box blur

Quantities of interest

Assumption

We have a model for the posterior distribution $\pi(x)$ for our imaging problem, and now want to compute it.

Quantities of interest

Assumption

We have a model for the posterior distribution $\pi(x)$ for our imaging problem, and now want to compute it.

We are interested in calculating the following quantities

Quantities of interest

Assumption

We have a model for the posterior distribution $\pi(x)$ for our imaging problem, and now want to compute it.

We are interested in calculating the following quantities

1. $\operatorname{argmax}_{x \in \mathbb{R}^d} \pi(x)$ (can be computed efficiently by convex optimisation) $\rightarrow$ **Maximum-a-posteriori (MAP)**

Quantities of interest

Assumption

We have a model for the posterior distribution $\pi(x)$ for our imaging problem, and now want to compute it.

We are interested in calculating the following quantities

1. $\text{argmax}_{x \in \mathbb{R}^d} \pi(x)$ (can be computed efficiently by convex optimisation) $\rightarrow$ **Maximum-a-posteriori (MAP)**
2. $\mathbb{E}_\pi(f) := \int_{\mathbb{R}^d} f(x)\pi(x)dx$ (can be computed using a Markov Chain Monte Carlo sampling algorithm)

Quantities of interest

Assumption

We have a model for the posterior distribution $\pi(x)$ for our imaging problem, and now want to compute it.

We are interested in calculating the following quantities

1. $\operatorname{argmax}_{x \in \mathbb{R}^d} \pi(x)$ (can be computed efficiently by convex optimisation) $\rightarrow$ **Maximum-a-posteriori (MAP)**
2. $\mathbb{E}_\pi(f) := \int_{\mathbb{R}^d} f(x)\pi(x)dx$ (can be computed using a Markov Chain Monte Carlo sampling algorithm)

$\mathbb{E}_\pi(x) := \int_{\mathbb{R}^d} x\pi(x)dx \approxeq \frac{1}{N}\sum_{i=1}^{N} X_i$
$\rightarrow$ **Minimum mean square error (MMSE)**

Example: Poisson deconvolution

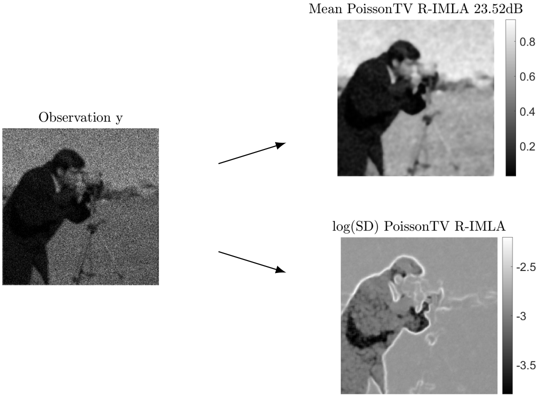


Figure: Quantities of interest: x_{MMSE} and standard deviation

Reminder: Acceleration in convex optimisation[1]

Let f a L-smooth and m-strongly convex function with conditioning number $\kappa = L/m$

[1] Some inspiration drawn from Sébastien Bubeck's blog

Reminder: Acceleration in convex optimisation[1]

Let f a L-smooth and m-strongly convex function with conditioning number $\kappa = L/m$

Accelerated Gradient Descent (AGD), Nesterov 2004

Set a point $x_0 = y_0$, iterate for $k \geq 0$

$$y_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$$

$$x_{k+1} = y_{k+1} + \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\left(y_{k+1} - y_k\right)$$

[1]Some inspiration drawn from Sébastien Bubeck's blog

Reminder: Acceleration in convex optimisation[1]

Let f a L-smooth and m-strongly convex function with conditioning number $\kappa = L/m$

Accelerated Gradient Descent (AGD), Nesterov 2004

Set a point $x_0 = y_0$, iterate for $k \geq 0$

$$y_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$$

$$x_{k+1} = y_{k+1} + \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}(y_{k+1} - y_k)$$

To reach ϵ accuracy, GD requires $\mathcal{O}(\kappa \log(1/\epsilon))$ iterations while AGD needs $\mathcal{O}(\sqrt{\kappa}\log(1/\epsilon)) \rightarrow$ **acceleration**!

[1] Some inspiration drawn from Sébastien Bubeck's blog

Overview

MCMC methods to sample from $\pi(x)$

To construct our sampling algorithm, we consider the overdamped Langevin stochastic differential equation (SDE)

$$dX_t = \nabla \log \pi(X_t) dt + \sqrt{2} dW_t$$

MCMC methods to sample from $\pi(x)$

To construct our sampling algorithm, we consider the overdamped Langevin stochastic differential equation (SDE)

$$dX_t = \nabla \log \pi(X_t)dt + \sqrt{2}dW_t$$

We aim for a discretisation that

- Allows us take time-step arbitrary large (*stability*)
- The numerical invariant distribution $\tilde{\pi}$ is close to the posterior π (*asymptotic bias*)

Discretisations of the Langevin equation: θ-method

We consider a θ-method to solve the Langevin SDE for $\theta \in [0, 1]$

$$X_{n+1} = X_n + \delta \nabla \log \pi(\theta X_{n+1} + (1 - \theta) X_n) + \sqrt{2\delta}\xi_{n+1}$$

[2] Durmus et al. (2018)

Discretisations of the Langevin equation: θ-method

We consider a θ-method to solve the Langevin SDE for $\theta \in [0, 1]$

$$X_{n+1} = X_n + \delta \nabla \log \pi(\theta X_{n+1} + (1 - \theta) X_n) + \sqrt{2\delta}\xi_{n+1}$$

A simple method is the unadjusted Langevin algorithm (ULA)[2], when $\theta = 0$

$$X_{n+1} = X_n + \delta \nabla \log \pi(X_n) + \sqrt{2\delta}\xi_{n+1}$$

The step size for ULA is bounded by $\delta \leq 2/L$ (same order as GD!).

[2] Durmus et al. (2018)

Accelerating using explicit vs implicit methods

Possible solutions

- Use an **implicit method** (time step δ can be *arbitrarily large*)
- Explicit numerical schemes to improve on effective δ

Accelerating using explicit vs implicit methods

Possible solutions

- Use an **implicit method** (time step δ can be *arbitrarily large*)
- Explicit numerical schemes to improve on effective δ

Related work:

- Accelerating MCMC using an explicit stabilised method (SK-ROCK) Pereyra et al. (2020) can can overcome the stability limit $2/L$ by factor s, authors recommended $s \in \{2, \dots, 15\}$.

Accelerating using explicit vs implicit methods

Possible solutions

- Use an **implicit method** (time step δ can be *arbitrarily large*)
- Explicit numerical schemes to improve on effective δ

Related work:

- Accelerating MCMC using an explicit stabilised method (SK-ROCK) Pereyra et al. (2020) can can overcome the stability limit $2/L$ by factor s, authors recommended $s \in \{2, \ldots, 15\}$.
- The proposed method has similarities to the implicit schemes proposed in Hodgkinson et al. (2021); Wibisono (2018). Our approach is well-posed for models that are not smooth, and has better non-asymptotic convergence bounds.

Proposed algorithm: Implicit Langevin Algorithm

Recall the θ-method

$$X_{n+1} = X_n + \delta \nabla \log \pi(\theta X_{n+1} + (1-\theta) X_n) + \sqrt{2\delta}\xi_{n+1}$$

Proposed algorithm: Implicit Langevin Algorithm

Recall the θ-method

$$X_{n+1} = X_n + \delta \nabla \log \pi(\theta X_{n+1} + (1-\theta) X_n) + \sqrt{2\delta}\xi_{n+1}$$

We find that this equation is equivalent to calculating the proximal operator of $x \mapsto \theta^{-1} \log \pi(\theta x + (1-\theta) X_n)$ for $\theta \in (0,1]$

Proposed algorithm: Implicit Langevin Algorithm

Recall the θ-method

$$X_{n+1} = X_n + \delta\nabla \log \pi(\theta X_{n+1} + (1-\theta)X_n) + \sqrt{2\delta}\xi_{n+1}$$

We find that this equation is equivalent to calculating the proximal operator of $x \mapsto \theta^{-1} \log \pi(\theta x + (1-\theta)X_n)$ for $\theta \in (0, 1]$

$$X_{n+1} = \operatorname{argmin}_{x\in\mathbb{R}^d} F(x; X_n; \xi_{n+1}),$$

$$F(x; X_n; \xi_{n+1}) = \theta^{-1} \log \pi(\theta x + (1-\theta)X_n + (2\delta)^{-1}||x - X_n - \sqrt{2\delta}\xi_{n+1}||^2.$$

Implicit Midpoint Langevin Algorithm (IMLA): $\theta = 1/2$

Recall $\log \pi(x) \propto -U(x)$

Algorithm IMLA

Require: $N \geq 0$, $\delta > 0$ and $X_0 \in \mathbb{R}^d$.
  for n=0 : N-1 **do**
    Draw

$$\xi_n \sim \mathcal{N}(0, I_d)$$

    Set

$$X_{n+1} \leftarrow \arg\min_{x \in \mathbb{R}^d} 2U\left(\frac{1}{2}x + \frac{1}{2}X_n\right) + \frac{1}{2\delta}\|x - X_n - \sqrt{2\delta}\xi_n\|^2$$

  end for $=0$

Convergence analysis for $\pi(x) \propto \exp\left(-\frac{1}{2}x^T\Sigma^{-1}x\right)$

$$W_2(\pi; Q_n) \leq W_2(\pi; \tilde{\pi}) + C^n W_2(\tilde{\pi}, Q_0)$$

Convergence analysis for $\pi(x) \propto \exp\left(-\frac{1}{2}x^T\Sigma^{-1}x\right)$

$$W_2(\pi; Q_n) \leq W_2(\pi; \tilde{\pi}) + C^n W_2(\tilde{\pi}, Q_0)$$

Proposition (simplified)

Let Q_n be the probability distribution associated with the n-th iteration of the method starting at X_0.

Convergence analysis for $\pi(x) \propto \exp\left(-\frac{1}{2}x^T\Sigma^{-1}x\right)$

$$W_2(\pi; Q_n) \leq W_2(\pi; \tilde{\pi}) + C^n W_2(\tilde{\pi}, Q_0)$$

Proposition (simplified)

Let Q_n be the probability distribution associated with the n-th iteration of the method starting at X_0. Then the number of steps n required such that $W_2(\pi, Q_n)^2$ reaches ϵ accuracy is given by

$$n \approx \frac{\sqrt{\kappa}}{2}\left[\log\left(W_2(\pi, Q_0)\right) - \log(\epsilon)\right] \quad \text{for } \theta = \frac{1}{2}$$

Convergence analysis for $\pi(x) \propto \exp\left(-\frac{1}{2}x^T\Sigma^{-1}x\right)$

$$W_2(\pi; Q_n) \leq W_2(\pi; \tilde{\pi}) + C^n W_2(\tilde{\pi}, Q_0)$$

Proposition (simplified)

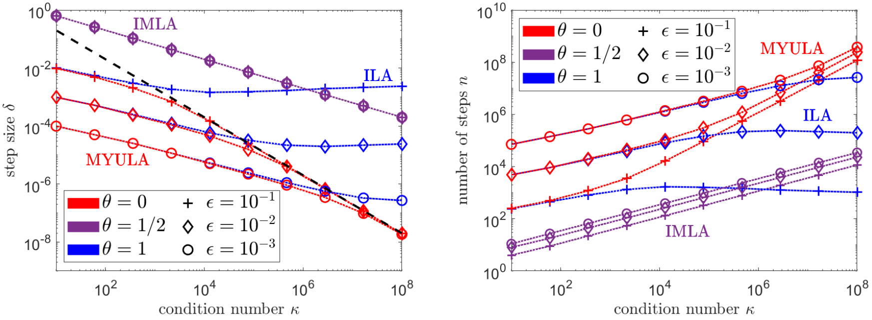
Let Q_n be the probability distribution associated with the n-th iteration of the method starting at X_0. Then the number of steps n required such that $W_2(\pi, Q_n)^2$ reaches ϵ accuracy is given by

$$n \approx \frac{\sqrt{\kappa}}{2}\left[\log\left(W_2(\pi, Q_0)\right) - \log(\epsilon)\right] \quad \text{for } \theta = \frac{1}{2}$$

with δ given by

$$\delta = \delta_* = \frac{2}{\sqrt{Lm}} = 2\sigma_{\min}\sigma_{\max} \quad \text{for } \theta = \frac{1}{2}$$

IMLA: Algorithmic recommendations



(a) step size δ against κ (b) number of steps n against κ

- Choosing $\theta = 1/2$ leads to order of $\sqrt{\kappa}$ iterations for convergence
- For strongly log-concave posteriors $\delta = \delta^* = 2/\sqrt{Lm}$ gives the optimal contraction rate

Overview

Recall: Poisson deconvolution

The posterior distribution $\pi(x)$ is modelled using a Poisson likelihood $y \sim \mathcal{P}(Ax, \beta)$ and a Moreau-Yosida smoothed TV regulariser (see Melidonis et al. (2023)):

$$\pi^{\lambda}(x) \propto \exp\Bigg(-\sum_{i=1}^{d} \left[(Ax)_i + \beta - y_i \log((Ax)_i + \beta) + \iota_{(Ax)_i \geq 0}\right] - \theta_{\mathrm{TV}^{\lambda}} \mathrm{TV}^{\lambda}(x) \Bigg).$$

Figure: True image x and observation y, MIV=10 Poisson noise and 5×5 box blur

Reflected IMLA

Algorithm R-IMLA

Require: $N \geq 0$, $\delta > 0$ and $X_0 \in \mathbb{R}^d$.

for n=0 : N-1 **do**

Draw

$$\xi_n \sim \mathcal{N}(0, I_d)$$

Set

$$X^i_{n+1} = |\tilde{X}^i_{n+1}|, \qquad \text{for all } i = 1, \dots, d,$$

$$\tilde{X}_{n+1} \leftarrow \arg\min_{x \in \mathbb{R}^d} 2U^\lambda\left(\frac{1}{2}x + \frac{1}{2}X_n\right) + \frac{1}{2\delta}\|x - X_n - \sqrt{2\delta}\xi_n\|^2$$

end for=0

Posterior mean, $s = 10$

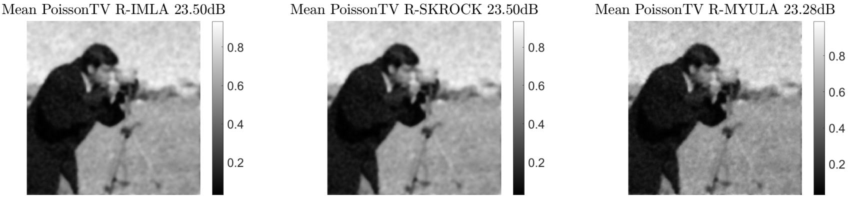


Figure: Posterior mean. ILA using $\delta = 6.65e^{-5}$ equivalent to step size when $s = 10$ for SKROCK, ULA using $\delta = 1/L$

Log of posterior standard deviation, $s = 10$

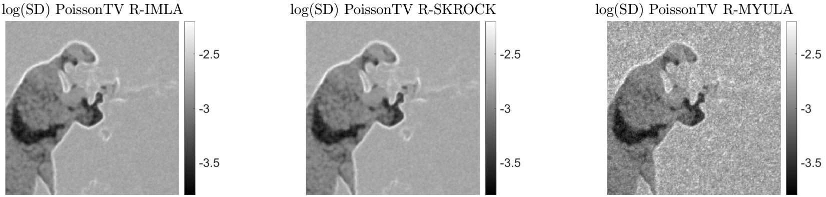


Figure: ILA using $\delta = 6.65e^{-5}$ equivalent to step size when $s = 10$ for SKROCK, ULA using $\delta = 1/L$

Autocorrelation comparison for $s = 20$ and $s = 40$

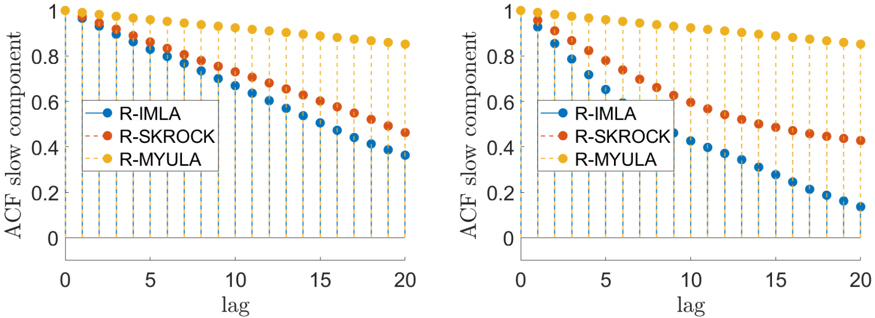


Figure: R-IMLA was run using $\delta = 2.82 \times 10^{-4}$ (left), and $\delta = 1.16 \times 10^{-3}$ (right) which is equivalent to the effective time step of R-SKROCK when $s = 20$ or $s = 40$, R-MYULA was run using $\delta = 1/L$.

Image deconvolution using a CRR-NN[3] prior

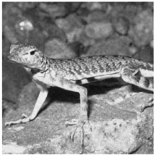
$$\pi(x) \propto \exp\left(-\frac{\|Ax-y\|^2}{2\sigma^2} - \frac{\lambda}{\mu} R_{\Theta}(\mu x)\right).$$

[3] Goujon et al. (2023)

Posterior means

Mean crr-nn IMLA 29.70dB

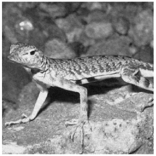
Mean crr-nn IMLA 30.04dB



Mean crr-nn IMLA 29.63dB

Mean crr-nn SKROCK 28.86dB

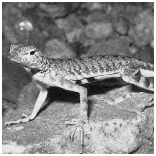
Mean crr-nn SKROCK 29.75dB



Mean crr-nn SKROCK 29.21dB

Mean crr-nn ULA 28.76dB

Mean crr-nn ULA 29.72dB



Mean crr-nn ULA 29.18dB

PSNR and convergence

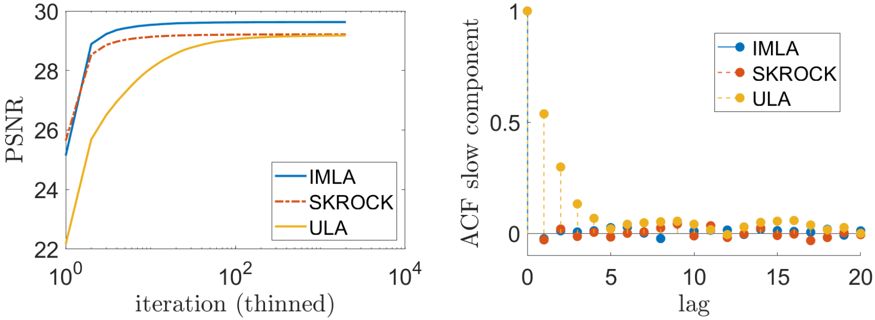
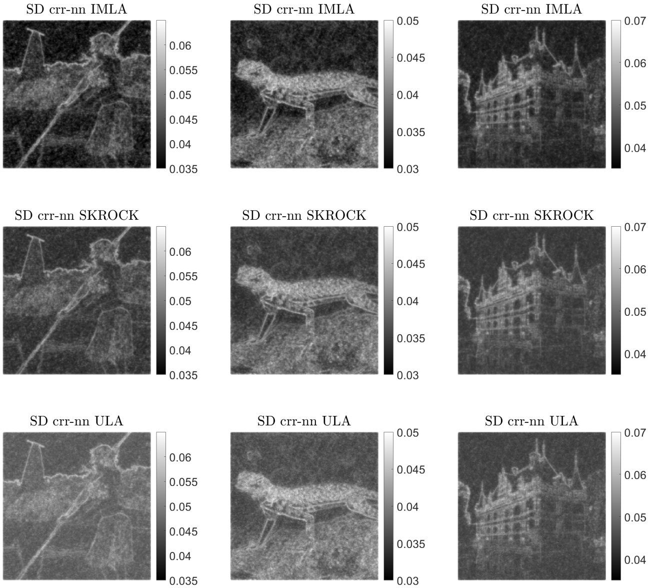


Figure: Results for the castle image

Pixel standard deviation



Overview

Summary

- Proposed new MCMC methodology for imaging inverse problems that provably accelerate the convergence to (numerical) equilibrium (similar behaviour to Nesterov method for optimisation)
- Identification of optimal time step to maximise convergence speed
- Non-smooth objectives and constraints can be dealt with
- The method involves an implicit step: This can be more computationally efficient that current state of the art SKROCK

Preprint available at https://teresa-klatzer.github.io

References

Durmus A., Moulines E., Pereyra M. Efficient Bayesian Computation by Proximal Markov Chain Monte Carlo: When Langevin Meets Moreau // SIAM J. Imaging Sci. 2018. 11, 1. 473–506.

Goujon A., Neumayer S., Bohra P., Ducotterd S., Unser M. A Neural-Network-Based Convex Regularizer for Inverse Problems // IEEE Trans. Comput. 2023. 9. 781–795.

Hodgkinson L., Salomone R., Roosta F. Implicit Langevin algorithms for sampling from log-concave densities // J Mach Learn Res. 2021. 22. 19–292.

Melidonis S., Dobson P., Altmann Y., Pereyra M., Zygalakis K. Efficient Bayesian Computation for Low-Photon Imaging Problems // SIAM J. Imaging Sci. 2023. 16, 3. 1195–1234.

Pereyra M., Vargas-Mieles L. A., Zygalakis K. C. Accelerating Proximal Markov Chain Monte Carlo by Using an Explicit Stabilized Method // SIAM J. Imaging Sci. 2020. 13, 2. 905–935.

Sampling as optimization in the space of measures: The Langevin dynamics as a composite optimization problem. // . 75. 2018. 2093–3027. (Proc. Mach. Learn. Res.).