Zig-Zag Sampler

A MCMC Game-Changer

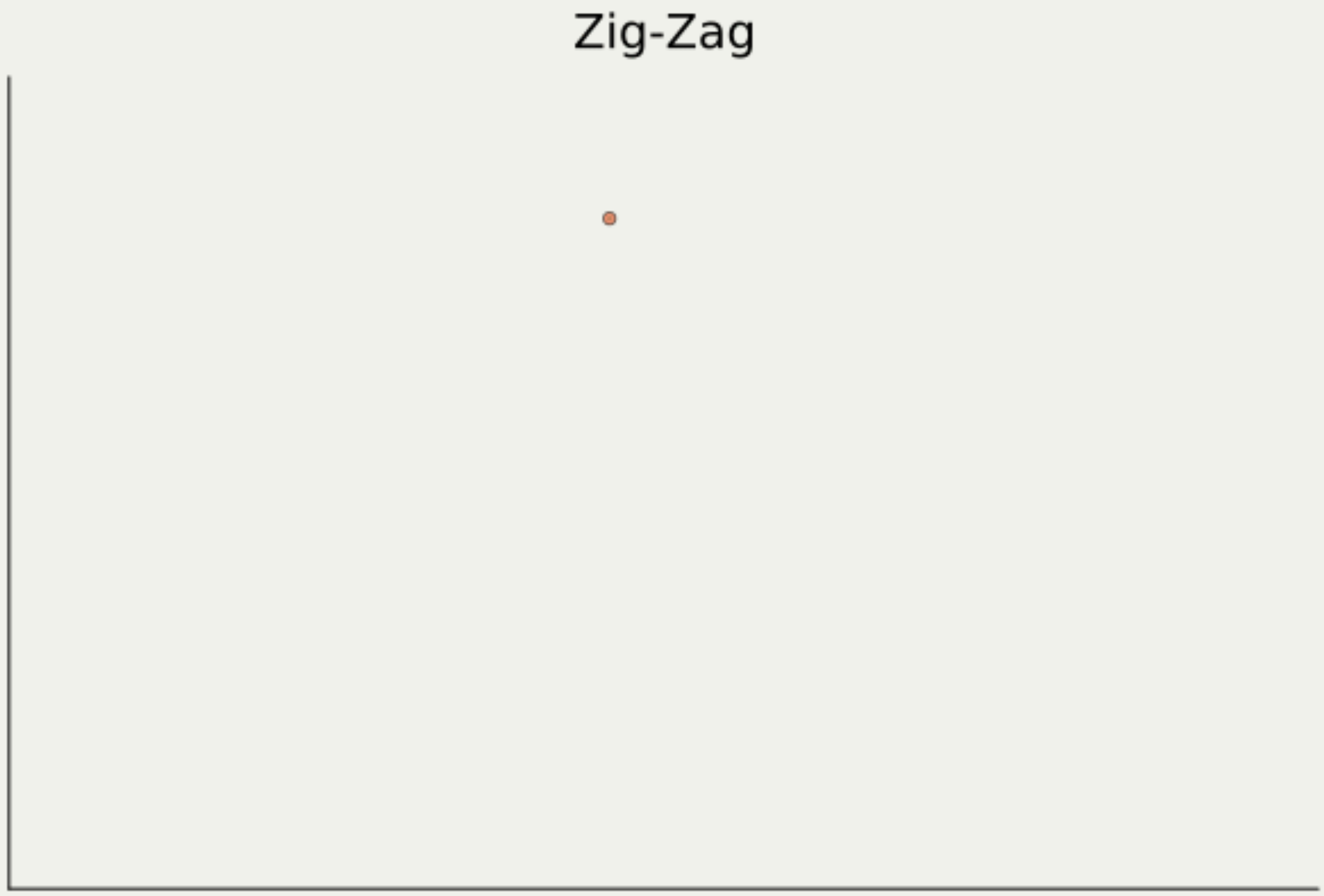
Hirofumi Shiba
Institute of Statistical Mathematics
the University of Tokyo

9/10/2024

Today's Menu

1 The Zig-Zag Sampler: What Is It?

A continuous-time variant of MCMC algorithms



Trajectory for Zig-Zag Sampler. Please attribute Hirofumi Shiba.

1.1 Keywords: PDMP (1/2)

PDMP (Piecewise Deterministic[1] Markov Process[2]) (Davis, 1984)

1. Mostly deterministic with the exception of random jumps happens at random times
2. Continuous-time, instead of discrete-time processes

→ Plays a complementary role to SDEs / Diffusions

Property	PDMP	SDE
Exactly simulatable?	✓	✗
Subject to discretization errors?	✗	✓
Driving noise	Poisson	Gauss

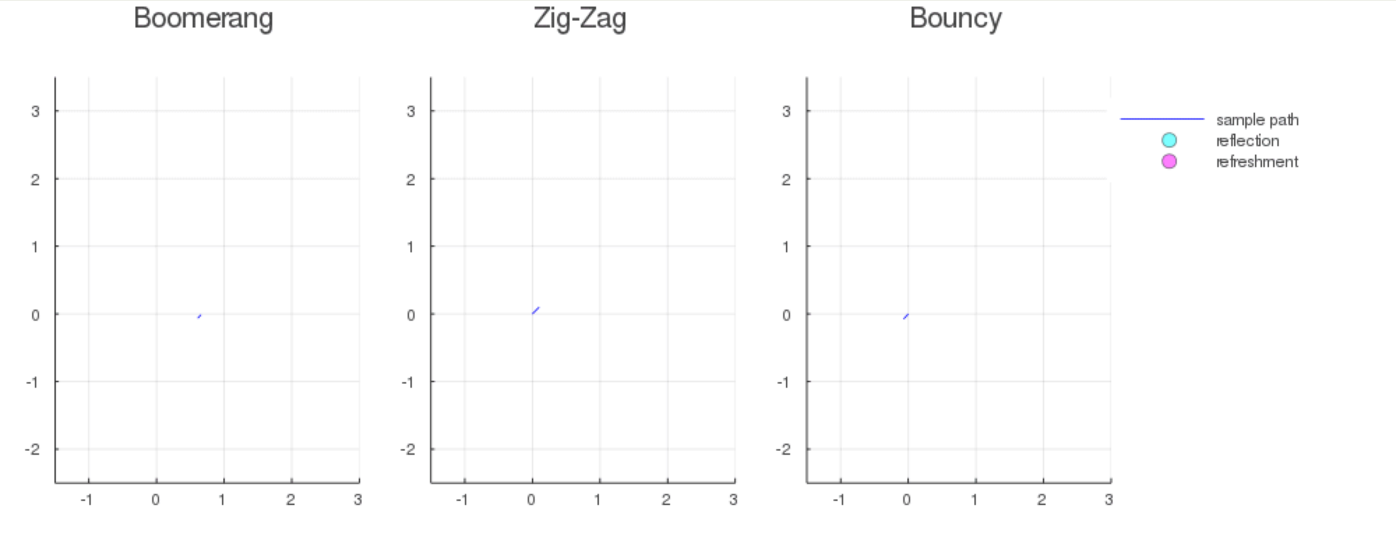
History of PDMP Applications

1. First applications: control theory, operations research, etc. (Davis, 1993)
2. Second applications: Monte Carlo simulation in material sciences (Peters and de With, 2012)
3. Third applications: Bayesian statistics (Bouchard-Côté et al., 2018)

1. Mostly deterministic with the exception of random jumps happens at random times
2. Continuous-time, instead of discrete-time processes

1.2 Keywords: PDMP (2/2)

- We will concentrate on Zig-Zag sampler (Bierkens, Fearnhead, et al., 2019)
- Other PDMPs: Bouncy sampler (Bouchard-Côté et al., 2018) , Boomerang sampler (Bierkens et al., 2020)



The most famous three PDMPs. Animated by (Grazzi, 2020)

1.3 Menu

What We've Learned

The new algorithm 'Zig-Zag Sampler' is based on comtinuous-time process called PDMP.

What We'll Learn in the Rest of this Section 1

We will review 3 instances of the standard (discrete-time) MCMC algorithm: MH, Lifted MH, and MALA.

1. Review: MH (Metropolis-Hastings) algorithm
2. Review: Lifted MH, A method bridging MH and Zig-Zag
3. Comparison: MH vs. Lifted MH vs. Zig-Zag
4. Review: MALA (Metropolis Adjusted Langevin Algorithm)
5. Comparison: Zig-Zag vs. MALA

1.4 Review: Metropolis-Hastings (1/2)

(Metropolis et al., 1953)-(Hastings, 1970)

Input: Target distribution p, (symmetric) proposal distribution q

1. Draw a $X_t \sim q(-|X_{t-1})$
2. Compute

$$\alpha(X_{t-1}, X_t) = \frac{p(X_t)}{p(X_{t-1})}$$

3. Draw a uniform random number $U \sim \mathrm{U}([0,1])$.
4. If $\alpha(X_{t-1}, X_t) \le U$, then $X_t \leftarrow X_{t-1}$. Do nothing otherwise.
5. Return to Step 1.

MH algorithm works even without p's normalizing constant. Hence, its ubiquity.

1.5 Review: Metropolis-Hastings (2/2)

Alternative View: MH is a generic procedure to turn a simple q-Markov chain into a Markov chain converging to p.

The Choise of Proposal q

- Random Walk Metropolis (Metropolis et al., 1953): Uniform / Gaussian

$$q(y|x) = q(y-x) \in \left\{ \frac{d\mathrm{U}([0,1])}{d\lambda}(y-x), \frac{d\mathrm{N}(0,\Sigma)}{d\lambda}(y-x) \right\}$$

- Hybrid / Hamiltonian Monte Carlo (Duane et al., 1987): Hamiltonian dynamics

$$q(y|x) = \delta_{x+\epsilon\rho}, \qquad \epsilon > 0,\ \ \rho : \text{momentum defined via Hamiltonian}$$

- Metropolis-adjusted Langevin algorithm (MALA) (Besag, 1994): Langevin diffusion

$$q(-|X_t) := \text{ the transition probability of } X_t \text{ where } dX_t = \nabla \log p(X_t)\,dt + \sqrt{2\beta^{-1}}dB_t.$$

1.6 Problem: Reversibility

Reversibility (a.k.a detailed balance):

$$p(x)q(x|y) = p(y)q(y|x).$$

In words:

$$\text{Probability}[\text{Going } x \to y] = \text{Probability}[\text{Going } y \to x].$$

→ Harder to explore the entire space

→ Slow mixing of MH

From the beginning of 21th century, many efforts have been made to make MH irreversible

1.7 Lifting (1/3)

Lifting: A method to make MH's dynamics irreversible

How?: By adding an auxiliary variable $\sigma \in \{\pm 1\}$, called momentum

Lifted MH (Turitsyn et al., 2011)

Input: Target p, two proposals $q^{(+1)}, q^{(-1)}$, and momentum $\sigma \in \{\pm 1\}$

1. Draw X_t from $q^{(\sigma)}$
2. Do a MH step
3. If accepted, go back to Step 1.
4. If rejected, flip the momentum and go back to Step 1.

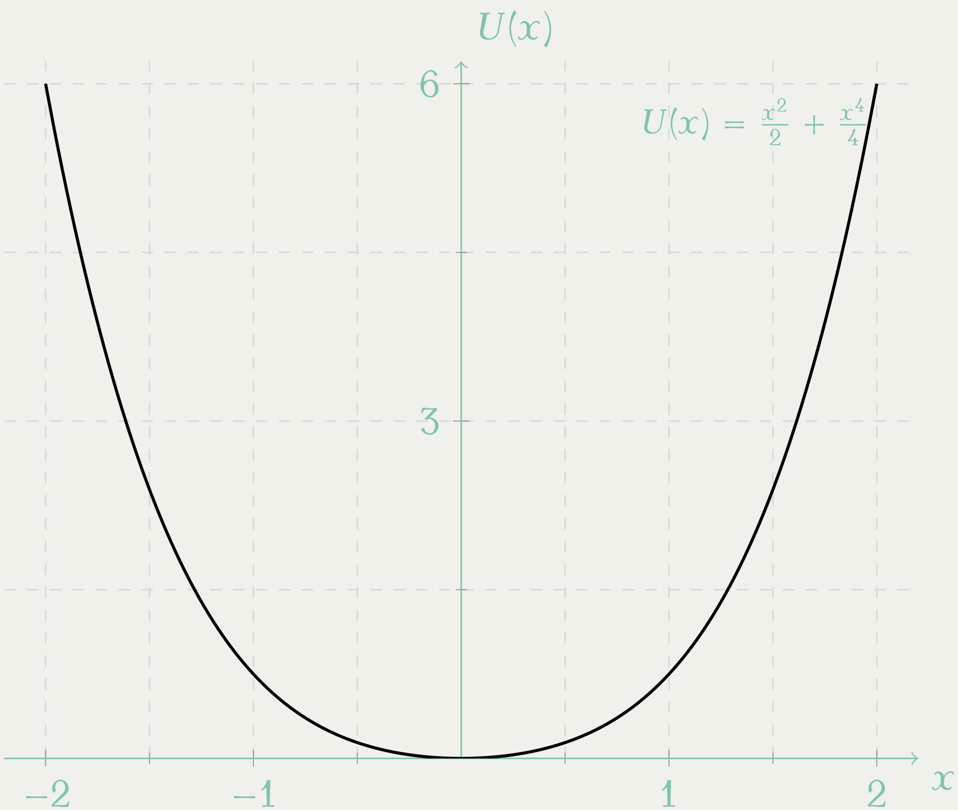
1.8 Lifting (2/3)

"Lifting"

$$\mathbb{R} \to \mathbb{R} \times \{+1\}, \quad \mathbb{R} \to \mathbb{R} \times \{-1\}$$

$q^{(+1)}$: Only propose $\rightarrow$ moves

$q^{(-1)}$: Only propose $\leftarrow$ moves



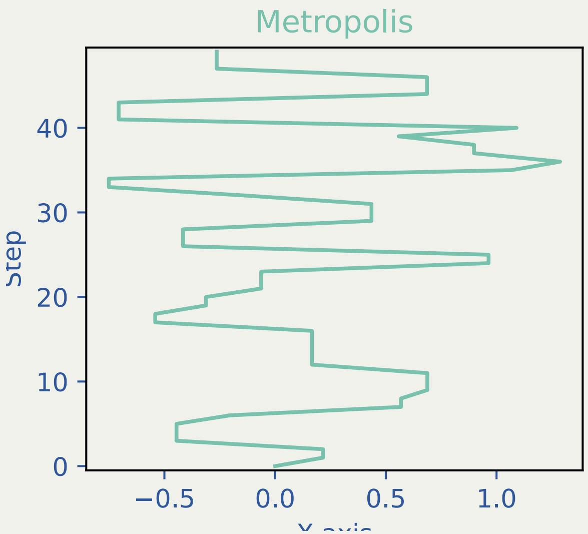
→ Once going uphill, it continues to go uphill.

→ This is irreversible, since

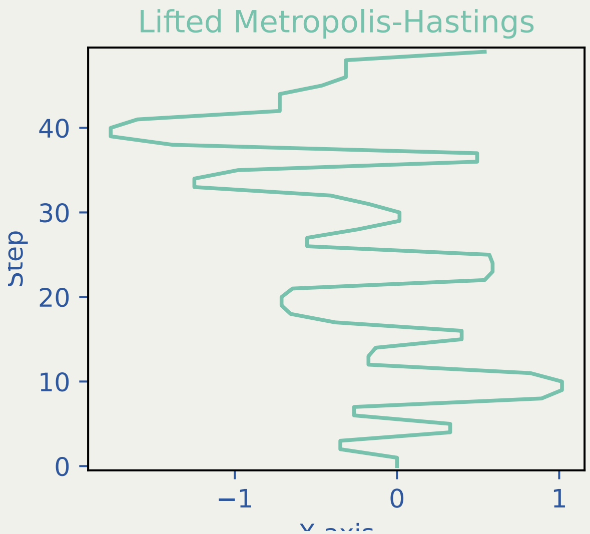
$$\text{Probability}[x \to y] \neq \text{Probability}[y \to x].$$

1.9 Lifting (3/3)

Reversible dynamic of MH has 'irreversified'



MH



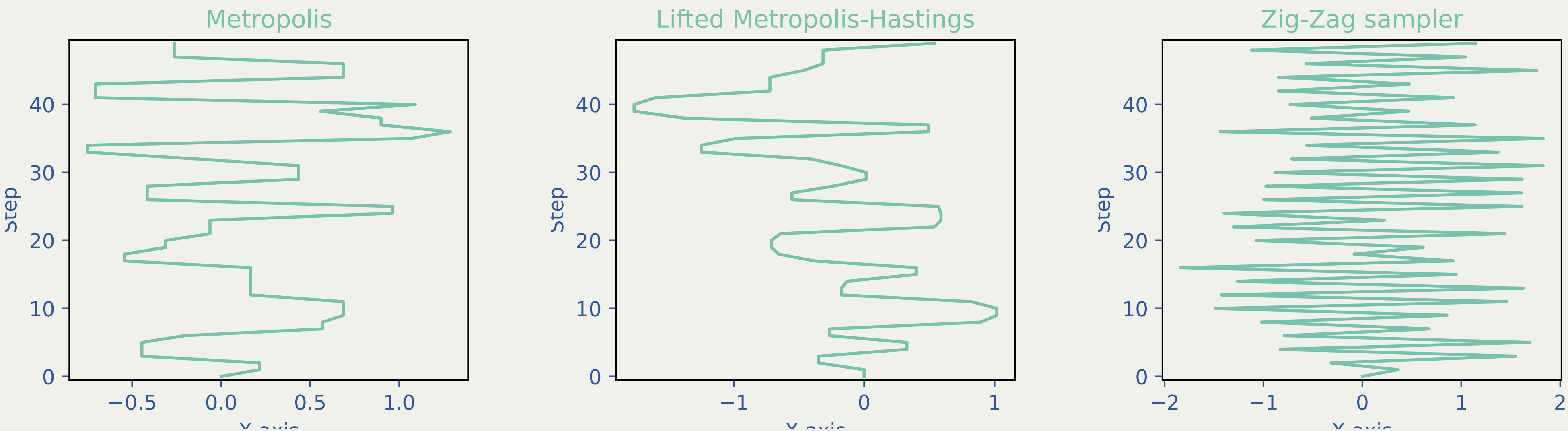
Lifted MH

> **Caution**
>
> **Scale is different**
>
> in the vertical axis!

Lifted MH successfully explores the edges of the target distribution.

*Irreversibility actually improves the efficiency of MCMC, as we observe in two slides later.

1.10 Comparison: MH vs. LMH vs. Zig-Zag (1/2)



MH

Lifted MH

Zig-Zag

Zig-Zag corresponds to the limiting case of lifted MH as the step size of proposal q goes to zero, as we'll learn later.

→ Zig-Zag has a maximum irreversibility.

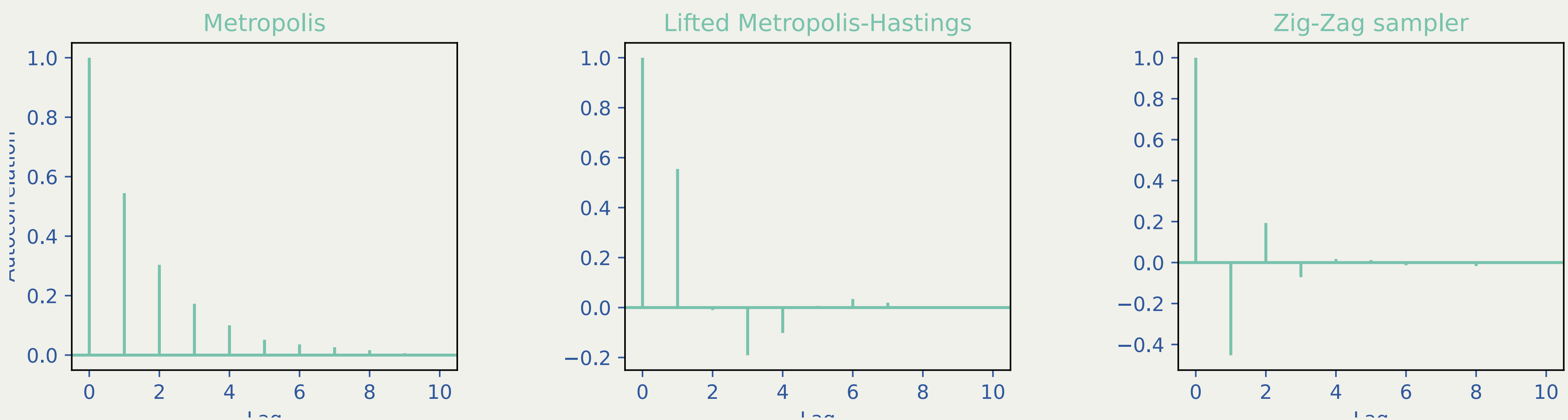
1.11 Comparison: MH vs. LMH vs. Zig-Zag (2/2)

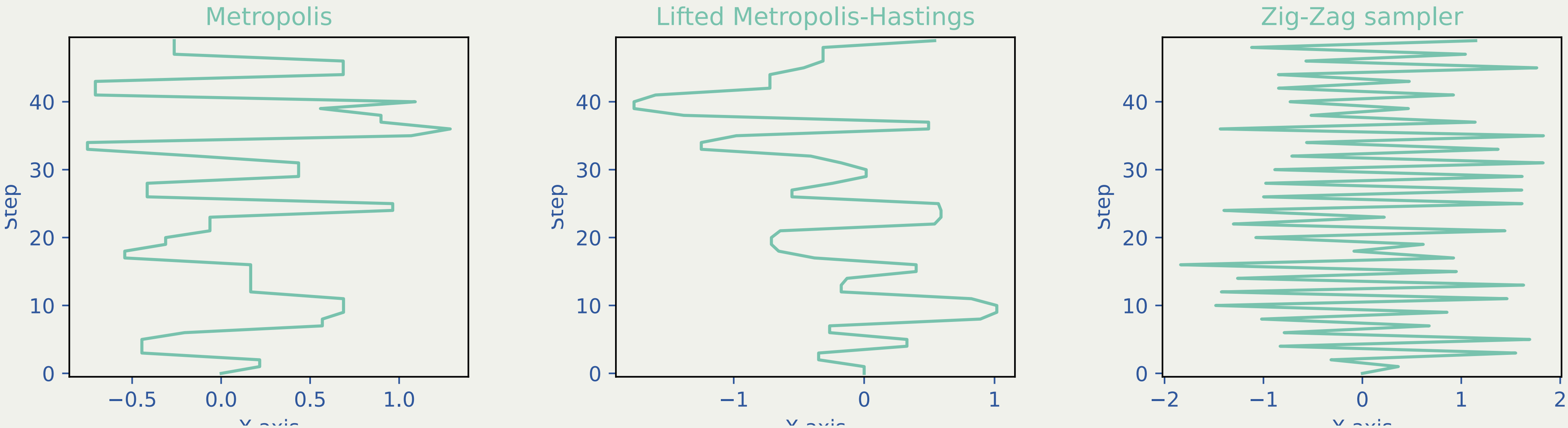
Irreversibility actually improves the efficiency of MCMC.

Faster decay of **autocorrelation** $\rho_t \approx \mathrm{Corr}[X_0, X_t]$ implies

1. faster mixing of MCMC

2. lower variance of Monte Carlo estimates





MH

Lifted MH

Zig-Zag

1.12 Review: MALA

Langevin diffusion: A diffusion process defined by the following SDE:

$$dX_t = \nabla \log p(X_t)\, dt + \sqrt{2\beta^{-1}} dB_t.$$

Langevin diffusion itself converges to the target distribution p in the sense that [1]

$$\|p_t - p\|_{L^1} \to 0, \qquad t \to \infty.$$

Two MCMC algorithms derived from **Langevin diffusion**:

ULA (Unadjusted Langevin Algorithm)

Use the discretization of (X_t). Discretization errors accumulate.

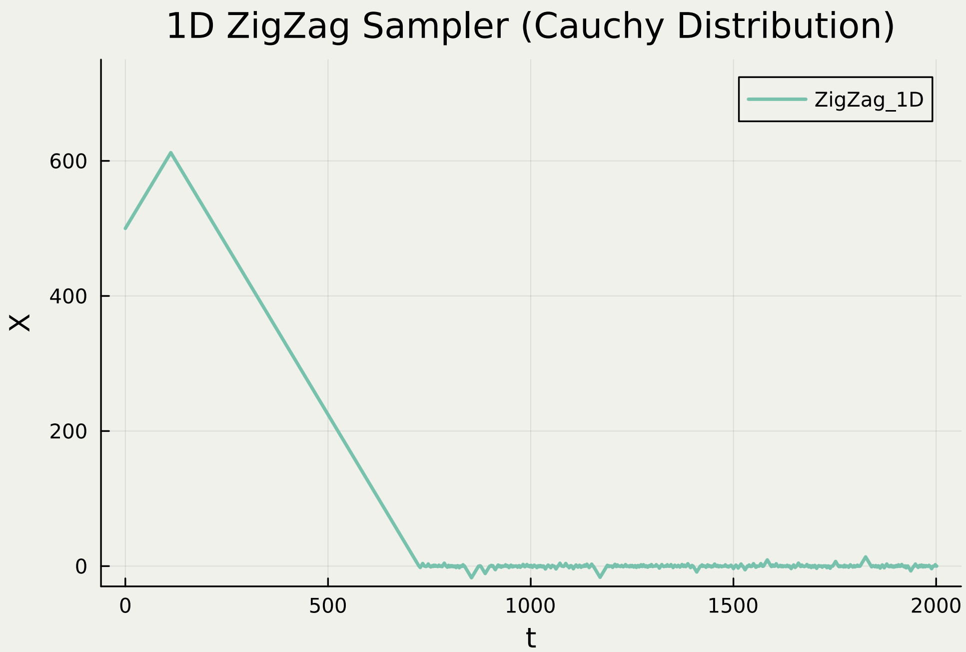
MALA (Metropolis Adjusted Langevin Algorithm)

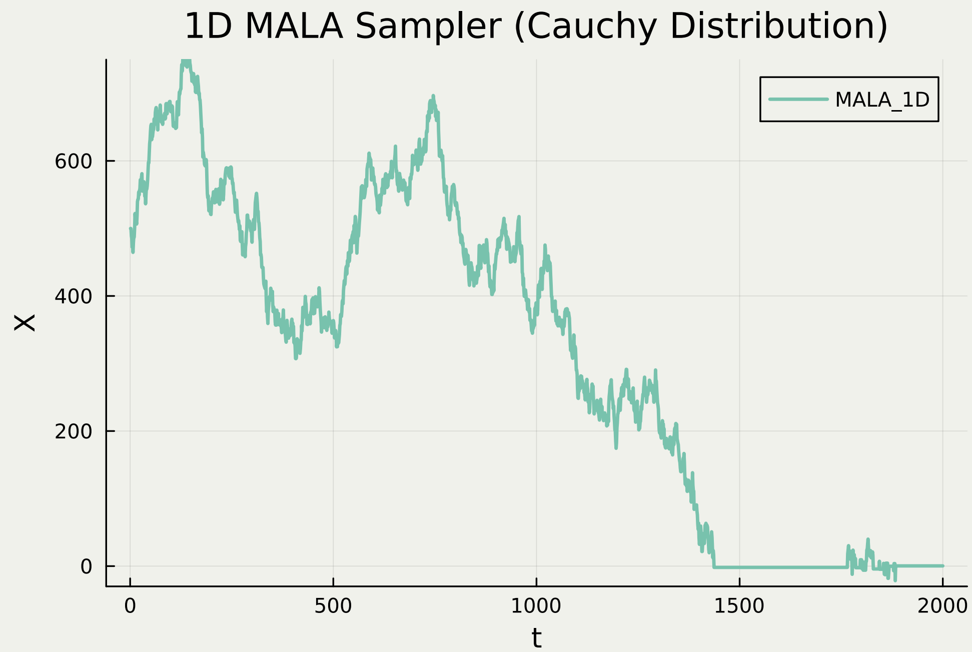
Use ULA as a proposal in MH, erasing the errors by MH steps.

1. under fairly general conditions on p.

1.13 Comparison: Zig-Zag vs. MALA (1/3)

How fast do they go back to high-probability regions? [1]





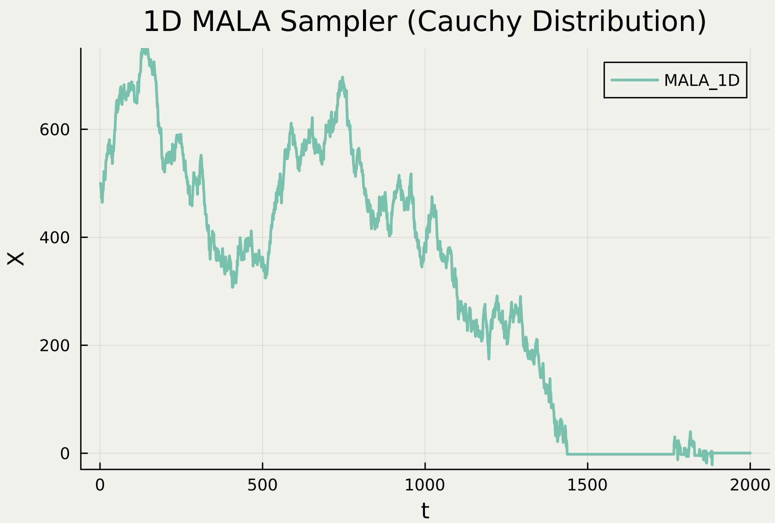
Zig-Zag

MALA

Irreversibility of Zig-Zag accelerates its convergence.

1. The target here is the standard Cauchy distribution $\mathrm{C}(0,1)$, equivalent to $\mathrm{t}(1)$ distribution. Its heavy tails hinder the convergence of MCMC.

1.14 Comparison: Zig-Zag vs. MALA (2/3)



MALA trajectory

Caution: Fake Continuity

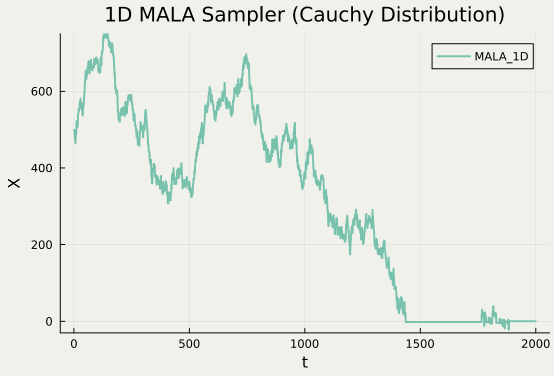
The left plot looks continuous, but **it actually is not**.

MH, including MALA, is actually a <u>discrete-time process</u>.

The plot is obtained by <u>connecting the points</u> by line segments.

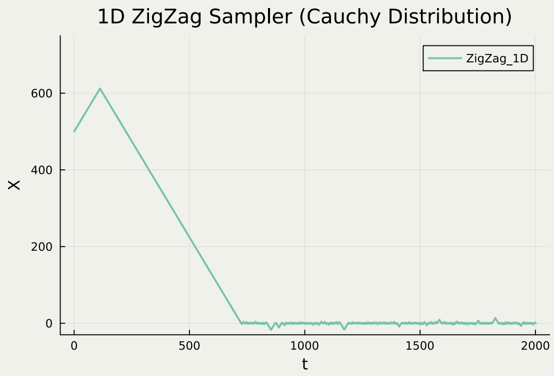
1.15 Comparison: Zig-Zag vs. MALA (3/3)

Monte Carlo estimation is also done differently:



MALA outputs $(X_n)_{n\in[N]}$ defines

$$\frac{1}{N}\sum_{n=1}^N f(X_n)\xrightarrow{N\to\infty}\int_{\mathbb{R}^d}f(x)p(x)\,dx.$$

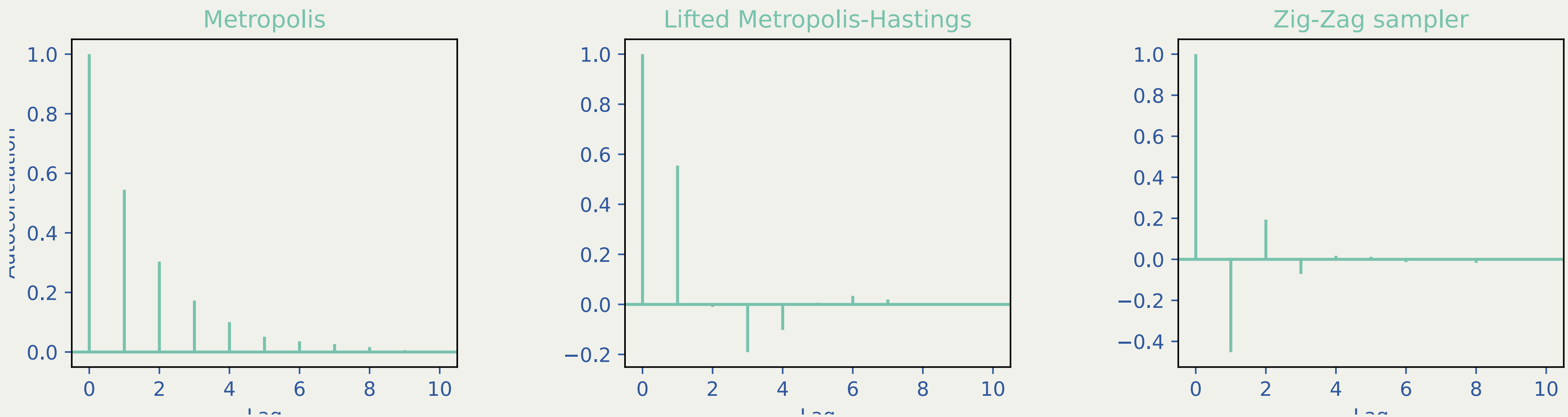


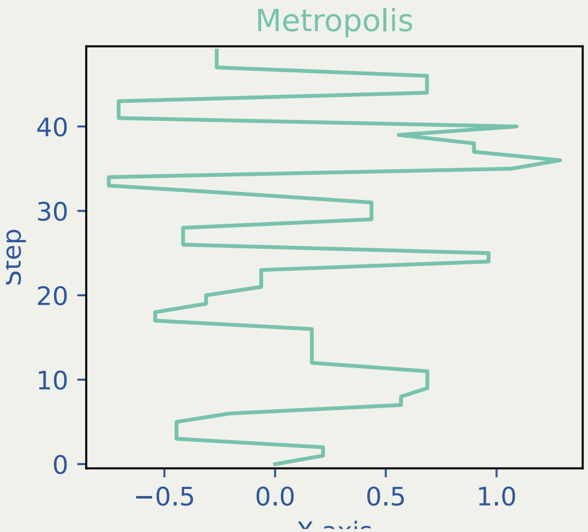
Zig-Zag outputs $(X_t)_{t\in[0,T]}$ defines

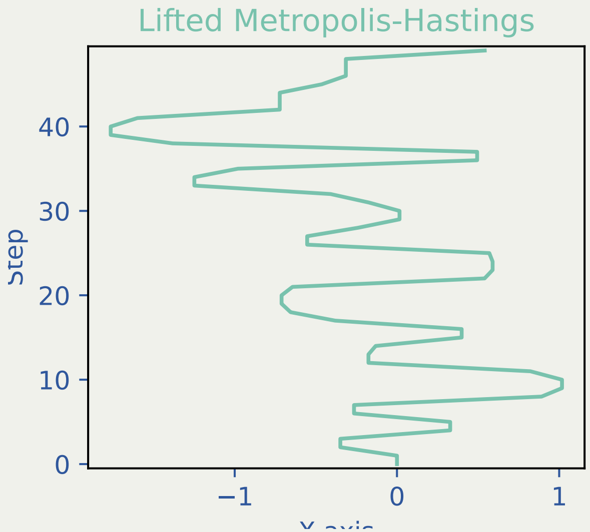
$$\int_0^T f(X_t)\,dt\xrightarrow{T\to\infty}\int_{\mathbb{R}^d}f(x)p(x)\,dx.$$

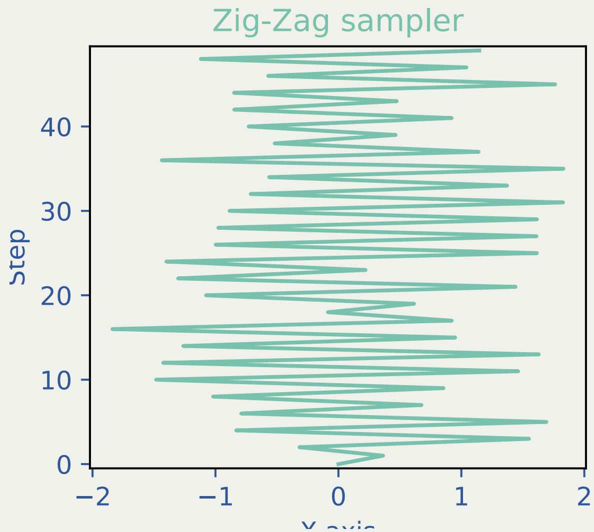
1.16 Recap of Section 1

- Zig-Zag Sampler's trajectory is a PDMP.
- PDMP, by design, has maximum irreversibility.
- Irreversibility leads to faster convergence of Zig-Zag in comparisons against MH, Lifted MH, and especially MALA.









MH

Lifted MH

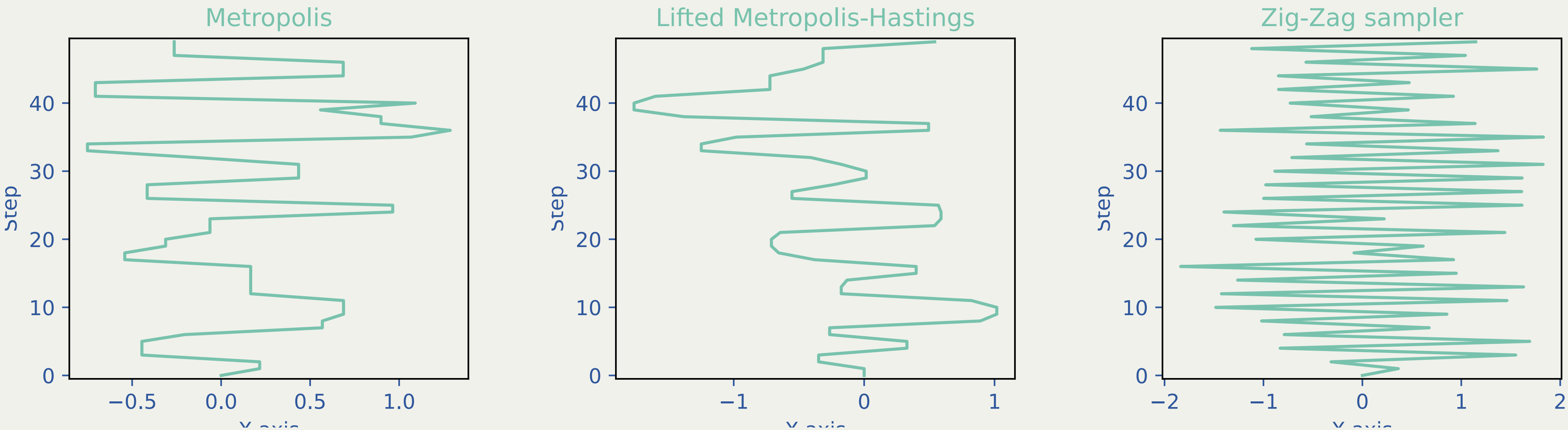
Zig-Zag

2 The Algorithm: How to Use It?

Fast and exact simulation of continuous trajectory.

2.1 Review: MH vs. LMH vs. Zig-Zag (1/2)

As we've learned before, Zig-Zag corresponds to the limiting case of lifted MH as the step size of proposal q goes to zero.



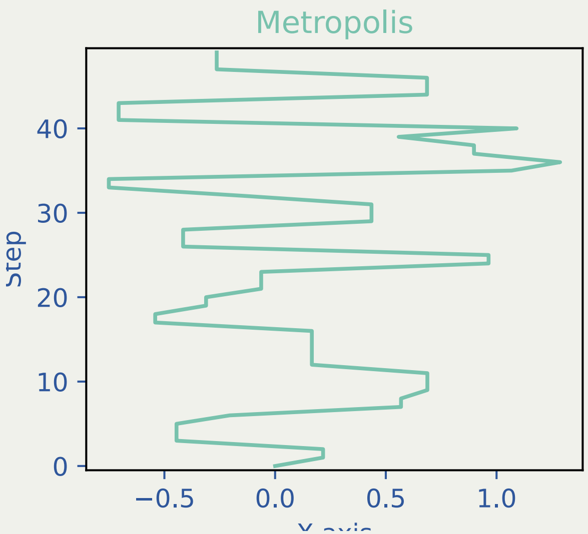
MH

Lifted MH

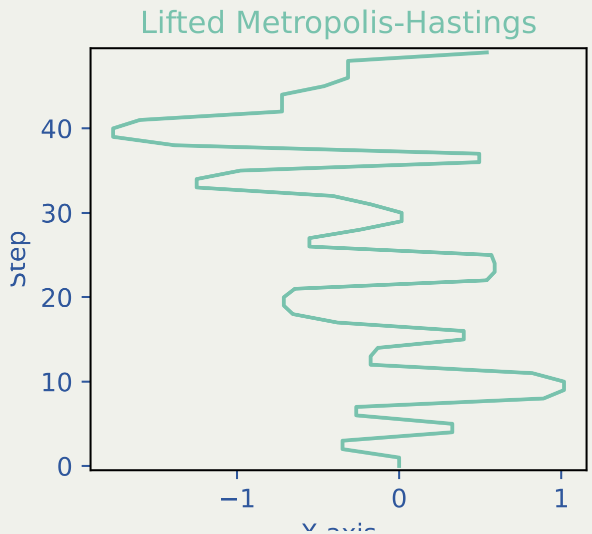
Zig-Zag

2.2 Review: MH vs. LMH vs. Zig-Zag (2/2)

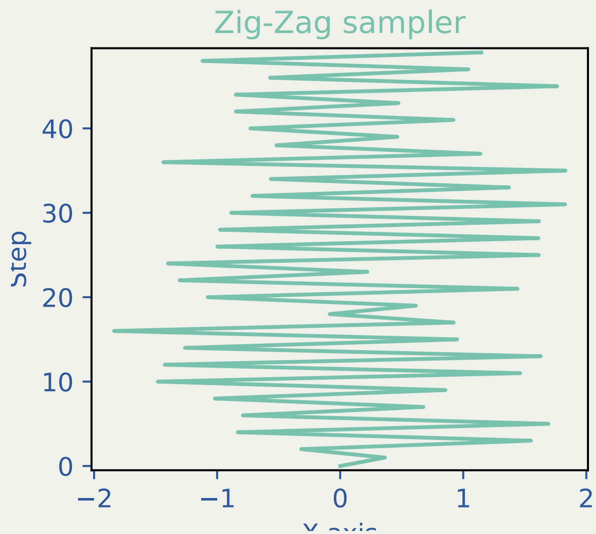
'Limiting case of lifted MH' means that we only simulate **where we should flip the momentum** $\sigma \in \{\pm 1\}$ in Lifted MH.



MH



Lifted MH



Zig-Zag

2.3 Algorithm (1/2)

'Limiting case of lifted MH' means that we only simulate **where we should flip the momentum** $\sigma \in \{\pm 1\}$ in Lifted MH.

(1d [1] Zig Zag sampler Bierkens, Fearnhead, et al., 2019**)**

Input: Gradient $\nabla \log p$ of log target density p

For $n \in \{1, 2, \cdots, N\}$:

1. Simulate an first arrival time T_n of a Poisson point process (described in the next slide)
2. Linearly interpolate until time T_n:

$$X_t = X_{T_{n-1}} + \sigma(t - T_{n-1}), \qquad t \in [T_{n-1}, T_n].$$

3. Go back to Step 1 with the momentum $\sigma \in \{\pm 1\}$ flipped

1. Multidimensional extension is straightforward, but we won't cover it today.

2.4 Algorithm (2/2)

(Fundamental Property of Zig-Zag Sampler (1d) Bierkens, Fearnhead, et al., 2019**)**

Let $U(x) := -\log p(x)$. Simluating a Poisson point process with a rate function

$$\lambda(x,\sigma) := \Big(\sigma U'(x)\Big)_+ + \gamma(x)$$

ensures the Zig-Zag sampler converges to the target p, where γ is an arbitrary non-negative function.

Its ergodicity is ensured as long as there exists $c, C > 0$ such that[1]

$$p(x) \le C|x|^{-c}.$$

1. With some regularity conditions on U. (See Bierkens, Roberts, et al., 2019).

2.5 Core of the Algorithm

Given a rate function

$$\lambda(x,\sigma) := \Big(\sigma U'(x)\Big)_+ + \gamma(x)$$

how to simulate a corresponding Poisson point process?

What We'll Learn in the Rest of this Section 2

1. What is Poisson Point Process?
2. How to Simulate It?
3. Core Technique: Poisson Thinning

Take Away: Zig-Zag sampling reduces to Poisson Thinning.

2.6 Simulating Poisson Point Process (1/2)

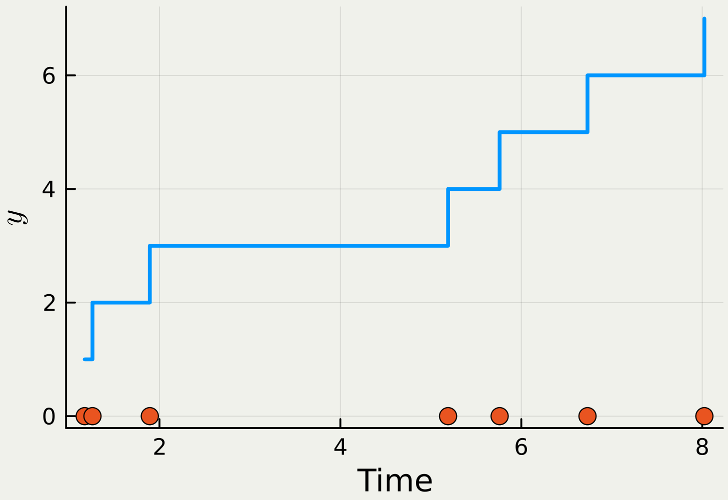
What is a Poisson Point Process with rate λ?

The number of points in $[0,t]$ follows a Poisson distribution with mean $\int_0^t \lambda(x_s,\sigma_s)\,ds$:

$$N([0,t]) \sim \mathrm{Pois}\,(M(t))\,, \qquad M(t) := \int_0^t \lambda(x_s,\sigma_s)\,ds.$$

We want to know when the first point T_1 falls on $[0,\infty)$.

Poisson Point Process with $\lambda \equiv 1$



When $\lambda(x,\sigma) \equiv c$ (constant),

- blue line: Poisson Process
- red dots: Poisson Point Process

satisfying $N_t = N([0,t]) \sim \mathrm{Pois}(ct)$.

2.7 Simulating Poisson Point Process (2/2)

Proposition (Simulation of Poisson Point Process)

The first arrival time T_1 of a Poisson Point Process with rate λ can be simulated by

$$T_1 \overset{\mathrm{d}}{=} M^{-1}(E), \qquad E \sim \mathrm{Exp}(1), M(t) := \int_0^t \lambda(x_s, \sigma_s)\, ds,$$

where $\mathrm{Exp}(1)$ denotes the exponential distribution with parameter 1.

Since $\lambda(x, \sigma) := \biggl(\sigma U'(x)\biggr)_+ + \gamma(x)$, M can be quite complicated.

→ Inverting M can be impossible.

→ We need more general techniques: Poisson Thinning.

2.8 Poisson Thinning (1/2)

(Lewis and Shedler, 1979)

To obtain the first arrival time T_1 of a Poisson Point Process with rate λ,

1. Find a bound M that satisfies

$$m(t) := \int_0^t \lambda(x_s, \sigma_s)\, ds \le M(t).$$

2. Simulate a point T from the Poisson Point Process with intensity M.
3. Accept T with probability $\frac{m(T)}{M(T)}$.

- $m(t)$: Defined via $\lambda(x,\sigma) := \left(\sigma U'(x)\right)_+ + \gamma(x)$.
- $M(t)$: Simple upper bound $m \le M$, such that M^{-1} is analytically tractable.

2.9 Poisson Thinning (2/2)

In order to simulate a Poisson Point Process with rate

$$\lambda(x,\sigma) := \Big(\sigma U'(x)\Big)_+ + \gamma(x),$$

we find a invertible upper bound M that satisfies

$$\int_0^t \lambda(x_s,\sigma_s)\,ds = m(t) \le M(t).$$

for all possible Zig-Zag trajectories $\{(x_s,\sigma_s)\}_{s\in[0,T]}$.

2.10 Recap of Section 2

1. Continuous-time MCMC, based on PDMP, has an entirely different algorithm and strategy.
2. To simulate PDMP is to simulate Poisson Point Process.
3. The core technology to simulate Poisson Point Process is Poisson Thinning.
4. Poisson Thinning is about finding an upper bound M, with tractable inverse M^{-1}; Typically a polynomial function.
5. The upper bound M has to be given on a case-by-case basis.

3 Proof of Concept: How Good Is It?

Quick demonstration of the state-of-the-art performance on a toy example.

3.1 Review: The 3 Steps of Zig-Zag Sampling

Given a target p,

1. Calculate the negative log-likelihood $U(x) := -\log p(x)$

2. Fix a refresh rate $\gamma(x)$ and compute the rate function

$$\lambda(x,\sigma) := \biggl(\sigma U'(x)\biggr)_+ + \gamma(x).$$

3. Find an invertible upper bound M that satisfies

$$\int_0^t \lambda(x_s,\sigma_s)\,ds =: m(t) \le M(t).$$

3.2 Model: 1d Gaussian Mean Reconstruction

Setting

- Data: $y_1,\cdots,y_n\in\mathbb{R}$ aquired by

$$y_i\overset{\text{iid}}{\sim}\mathrm{N}(x_0,\sigma^2),\qquad i\in[n],$$

  with $\sigma>0$ known, $x_0\in\mathbb{R}$ unknown.

- Prior: $\mathrm{N}(0,\rho^2)$ with known $\rho>0$.
- Goal: Sampling from the posterior

$$p(x)\;\propto\;\left(\prod_{i=1}^n\phi(x|y_i,\sigma^2)\right)\phi(x|0,\rho^2),$$

  where $\phi(x|y,\sigma^2)$ is the $\mathrm{N}(y,\sigma^2)$ density.

The negative log-likelihood:

$$\begin{aligned}U(x)&=-\log p(x)\\&=\frac{x^2}{2\rho^2}+\frac{1}{2\sigma^2}\sum_{i=1}^n(x-y_i)^2\end{aligned}$$

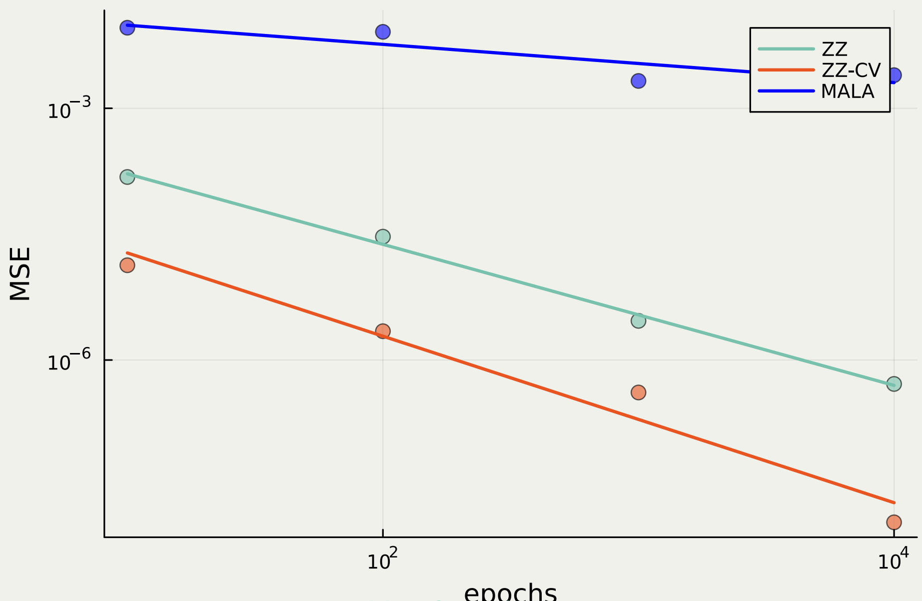
$$U'(x)=\frac{x}{\rho^2}+\frac{1}{\sigma^2}\sum_{i=1}^n(x-y_i),$$

$$U''(x)=\frac{1}{\rho^2}+\frac{n}{\sigma^2}.$$

3.3 Menu

In the rest of this Section 3, we'll learn:

1. Even a simple Zig-Zag Sampler with $\gamma \equiv 0$ surpasses MALA.
2. Incorporating sub-sampling, Zig-Zag with Control Variates further improves the efficiency.



3.4 Simple Zig-Zag Sampler with $\gamma \equiv 0$ (1/2)

Fixing $\gamma \equiv 0$, we obtain the upper bound M

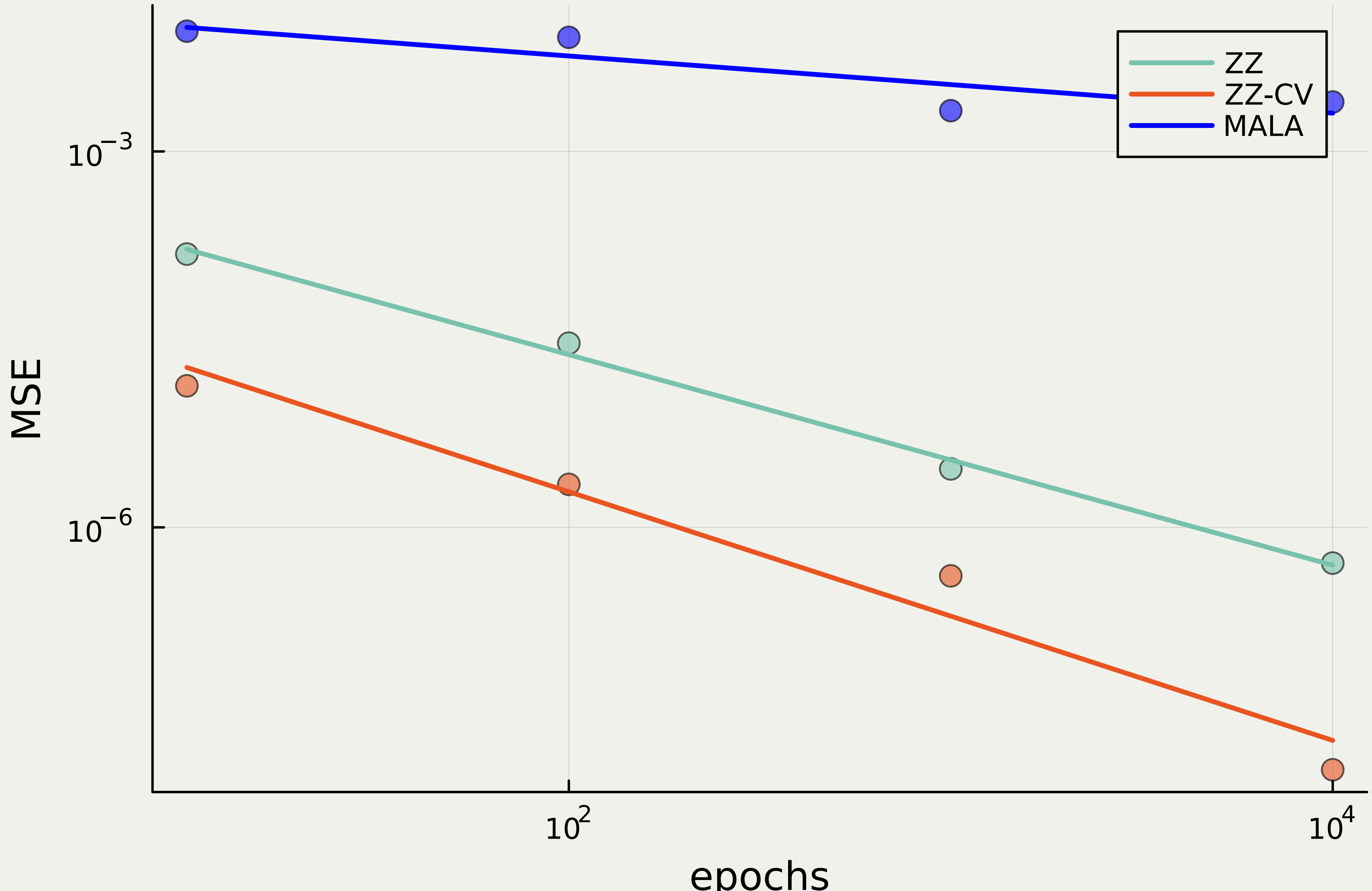
$$\begin{aligned} m(t) &= \int_0^t \lambda(x_s, \sigma_s)\,ds = \int_0^t \Big(\sigma U'(x_s)\Big)_+ ds \\ &\le \left(\frac{\sigma x}{\rho^2} + \frac{\sigma}{\sigma^2}\sum_{i=1}^n (x - y_i) + t\left(\frac{1}{\rho^2} + \frac{n}{\sigma^2}\right)\right)_+ \\ &=: (a + bt)_+ = M(t), \end{aligned}$$

where

$$a = \frac{\sigma x}{\rho^2} + \frac{\sigma}{\sigma^2}\sum_{i=1}^{n}(x - y_i), \quad b = \frac{1}{\rho^2} + \frac{n}{\sigma^2}.$$

3.5 Result: 1d Gaussian Mean Reconstruction

We generated 100 samples from $\mathrm{N}(x_0, \sigma^2)$ with $x_0 = 1$.



3.6 MSE per Epoch: The Vertical Axis

MSE (Mean Squared Error) of $\{X_i\}_{i=1}^n$ is defined as

$$\frac{1}{n}\sum_{i=1}^n (X_i - x_0)^2.$$

Epoch: Unit computational cost.

The following is considered as one epoch:

- One evaluation of a likelihood ratio

$$\frac{p(X_{n+1})}{p(X_n)}.$$

- One evaluation of a Poisson Point Process.

3.7 Good News!

Case-by-case construction of an upper bound M is too complicated / demanding.

Therefore, we are trying to automate the whole procedure.

Automatic Zig-Zag

1. Automatic Zig-Zag (Corbella et al., 2022)
2. Concave-Convex PDMP (Sutton and Fearnhead, 2023)
3. NuZZ (numerical Zig-Zag) (Pagani et al., 2024)

References

Slides and codes are available here

Besag, J. E. (1994). Comments on "Representations of Knowledge in Complex Systems" by U. Grenander and M. I. Miller. *Journal of the Royal Statistical Society. Series B (Methodological)*, *56*(4), 591–592.

Bierkens, J., Fearnhead, P., and Roberts, G. (2019). The Zig-Zag Process and Super-Efficient Sampling for Bayesian Analysis of Big Data. *The Annals of Statistics*, *47*(3), 1288–1320.

Bierkens, J., Grazzi, S., Kamatani, K., and Roberts, G. O. (2020). The boomerang sampler. *Proceedings of the 37th International Conference on Machine Learning*, *119*, 908–918.

Bierkens, J., Roberts, G. O., and Zitt, P.-A. (2019). Ergodicity of the zigzag process. *The Annals of Applied Probability*, *29*(4), 2266–2301.

Bouchard-Côté, A., Vollmer, S. J., and Doucet, A.

(2018). The bouncy particle sampler: A nonreversible rejection-free markov chain monte carlo method. *Journal of the American Statistical Association*, *113*(522), 855–867.

Corbella, A., Spencer, S. E. F., and Roberts, G. O. (2022). Automatic zig-zag sampling in practice. *Statistics and Computing*, *32*(6), 107.

Dai, H., Pollock, M., and Roberts, G. (2019). Monte Carlo Fusion. *Journal of Applied Probability*, *56*(1), 174–191.

Davis, M. H. A. (1984). Piecewise-deterministic markov processes: A general class of non-diffusion stochastic models. *Journal of the Royal Statistical Society. Series B (Methodological)*, *46*(3), 353–388.

Davis, M. H. A. (1993). Markov models and optimization,Vol. 49. Chapman & Hall.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid monte carlo. *Physics Letters B*, *195*(2), 216–222.

Fearnhead, P., Grazzi, S., Nemeth, C., and Roberts, G. O. (2024). Stochastic gradient piecewise deterministic monte carlo samplers.

Grazzi, S. (2020). Piecewise deterministic monte carlo.

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika*, *57*(1), 97–109.

Lewis, P. A. W., and Shedler, G. S. (1979). Simulation of nonhomogeneous poisson processes by thinning. *Naval Research Logistics Quarterly*, *26*(3), 403–413.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, *21*(6), 1087–1092.

Pagani, F., Chevallier, A., Power, S., House, T., and Cotter, S. (2024). NuZZ: Numerical zig-zag for general models. *Statistics and Computing*, *34*(1)

61.
Peters, E. A. J. F., and de With, G. (2012). Rejection-free monte carlo sampling for general potentials. *Physical Review E*, *85*(2).
Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I., and McCulloch, R. E. (2016). Bayes and big data: The consensus monte carlo algorithm. *International Journal of Management Science and Engineering Management*, *11*, 78–88.
Srivastava, S., Cevher, V., Dinh, Q., and Dunson, D. (2015). WASP: Scalable Bayes via barycenters of subset posteriors. In G. Lebanon and S. V. N. Vishwanathan, editors, *Proceedings of the eighteenth international conference on artificial intelligence and statistics*, Vol. 38, pages 912–920. San Diego, California, USA: PMLR.
Sutton, M., and Fearnhead, P. (2023). Concave-convex PDMP-based sampling. *Journal of Computational and Graphical Statistics*, *32*(4),

1425–1435.

Turitsyn, K. S., Chertkov, M., and Vucelja, M. (2011). Irreversible Monte Carlo algorithms for Efficient Sampling. *Physica D-Nonlinear Phenomena*, *240*(5-Apr), 410–414.

Welling, M., and Teh, Y. W. (2011). Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on international conference on machine learning*, pages 681–688. Madison, WI, USA: Omnipress.

Appendix: Scalability by Subsampling

Construction of ZZ-CV (Zig-Zag with Control Variates).

3.8 Review: 1d Gaussian Mean Reconstruction

U' has an alternative form:

$$U'(x) = \frac{x}{\rho^2} + \frac{1}{\sigma^2}\sum_{i=1}^n (x - y_i) =: \frac{1}{n}\sum_{i=1}^n U_i'(x),$$

where

$$U_i'(x) = \frac{x}{\rho^2} + \frac{n}{\sigma^2}(x - y_i).$$

→ We only need one sample y_i to evaluate U_i'.

3.9 Randomized Rate Function

Instead of

$$\lambda_{\mathrm{ZZ}}(x,\sigma) = \Big(\sigma U'(x)\Big)_+$$

we use

$$\lambda_{\mathrm{ZZ\text{-}CV}}(x,\sigma) = \Big(\sigma U_I'(x)\Big)_+, \qquad I \sim \mathrm{U}([n]).$$

Then, the latter is an unbiased estimator of the former:

$$\mathrm{E}_{I\sim\mathrm{U}([n])}\Big[\lambda_{\mathrm{ZZ\text{-}CV}}(x,\sigma)\Big] = \lambda_{\mathrm{ZZ}}(x,\sigma).$$

3.10 Last Step: Poisson Thinning

Find an invertible upper bound M that satisfies

$$\int_0^t \lambda_{\text{ZZ-CV}}(x_s, \sigma_s)\,ds =: m_I(t) \le M(t), \qquad I \sim \mathrm{U}([n]).$$

It is harder to bound $\lambda_{\text{ZZ-CV}}$, since it is now an estimator (random function).

3.11 Upper Bound M with Control Variates

Preprocessing (once and for all)

1. Find

$$x_* := \operatorname*{argmin}_{x\in\mathbb{R}} U(x)$$

2. Compute

$$U'(x_*) = \frac{x_*}{\rho^2} + \frac{1}{\sigma^2}\sum_{i=1}^n (x_* - y_i).$$

Then, with a re-parameterization of m_i,

$$m_i(t) \le M(t) := a + bt,$$

where

$$a = (\sigma U'(x_*))_+ + \|U'\|_{\mathrm{Lip}}\|x - x_*\|_p, \qquad b := \|U'\|_{\mathrm{Lip}}.$$

And m_i is redefined as

$$m_i(t) = U'(x_*) + U_i'(x) - U_i'(x_*).$$

3.12 Subsampling with Control Variates

Zig-Zag sampler with the random rate function

$$\lambda_{\text{ZZ-CV}}(x,\sigma)=\left(\sigma U_I'(x)\right)_+,\qquad I\sim\mathrm{U}([n]).$$

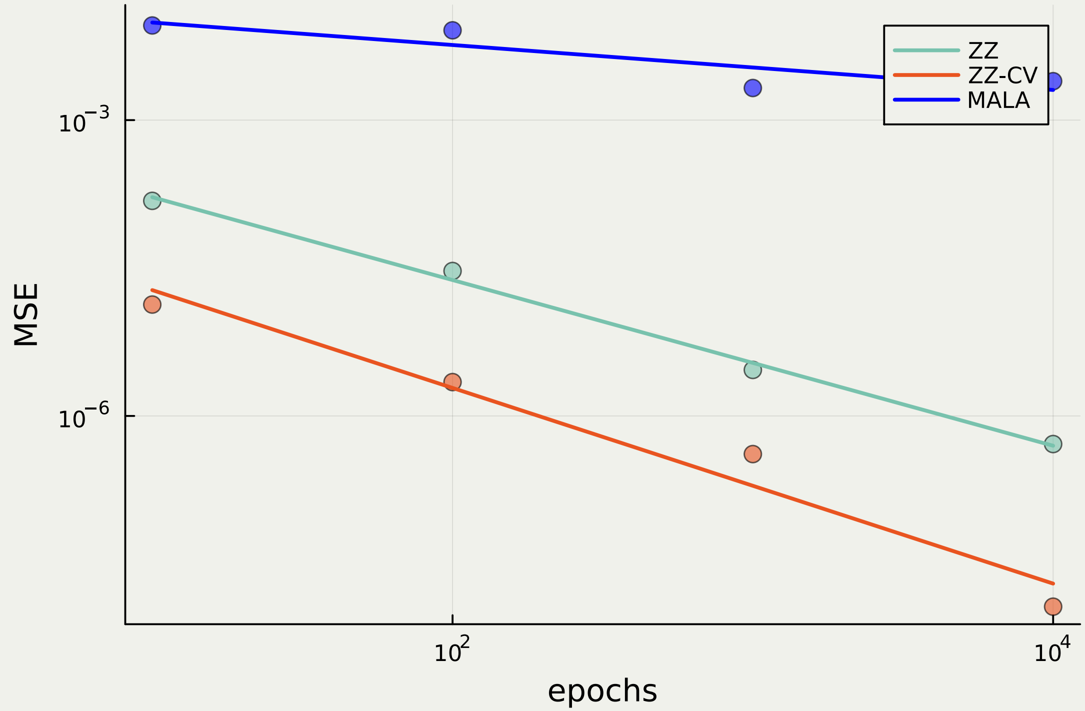
and the upper bound

$$M(t)=a+bt$$

is called Zig-Zag with Control Variates (Bierkens, Fearnhead, et al., 2019).

3.13 Zig-Zag with Control Variates

1. has $O(1)$ efficiency as the sample size n grows.[1]
2. is exact (no bias).



1. As long as the preprocessing step is properly done.

3.14 Scalability (1/3)

There are currently two main approaches to scaling up MCMC for large data.

1. Devide-and-conquer

   Devide the data into smaller **chunks** and run MCMC on each **chunk**.

2. Subsampling

   Use a subsampling estimate of the likelihood, which does not require the entire data.

3.15 Scalability (2/3) by Devide-and-conquer

Devide the data into smaller chunks and run MCMC on each chunk.

Unbiased?	**Method**	**Reference**
✗	WASP	(Srivastava et al., 2015)
✗	Consensus Monte Carlo	(Scott et al., 2016)
✓	Monte Carlo Fusion	(Dai et al., 2019)

3.16 Scalability (3/3) by Subsampling

Use a subsampling estimate of the likelihood, which does not require the entire data.

Unbiased?	Method	Reference
✗	Stochastic Gadient MCMC	(Welling and Teh, 2011)
✓	Zig-Zag with Subsampling	(Bierkens, Fearnhead, et al., 2019)
✗	Stochastic Gradient PDMP	(Fearnhead et al., 2024)

0 reactions

0 comments

Write Preview Aa

Sign in to comment

Sign in with GitHub