Zig-Zag Sampl

A MCMC Game-Change

Hirofumi Shiba
Institute of Statistical Mathematic
the University of Tokyo

9/10/2024

Today's Menu

- 1 The Zig-Zag Sampler: What Is It?
- 2 The Algorithm: How to Use It?
- 3 Proof of Concept: How Good Is It?

1 The Zig-Zag Sampler: V

A continuous-time variant of MCMC algorit

Trajectory for Zig-Zag Sampler. Please attribute Hirofumi Shiba

1.1 Keywords: PDMP (1/2)

<u>PDMP</u> (Piecewise Deterministic[1] Markov Pr

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

→ Plays a complementary role to SDEs / Di

Property	PD
Exactly simulatable?	✓
Subject to discretization errors?	✗
Driving noise	Pois

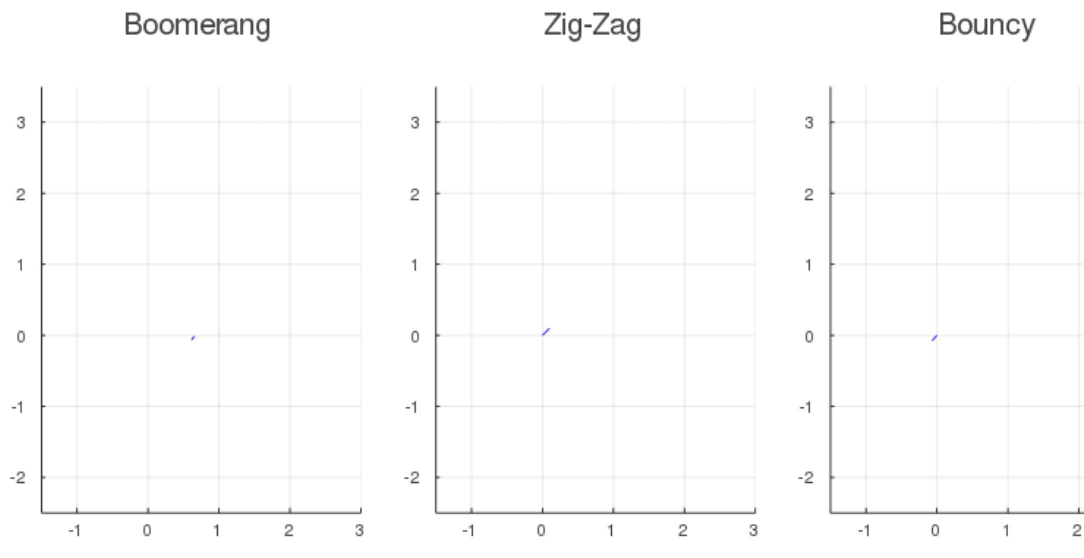
History of PDMP Applications

1. First applications: control theory, operations research,
2. Second applications: Monte Carlo simulation in materia
   de With, 2012)
3. Third applications: Bayesian statistics (Bouchard-Côté e

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

1.2 Keywords: PDMP (2/2)

- We will concentrate on Zig-Zag sampler Fearnhead, et al., 2019)
- Other PDMPs: Bouncy sampler (Bouchar Boomerang sampler (Bierkens et al., 202



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

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

We will review 3 instances of the standard (discrete-time) M
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 distributio

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 $\alpha(X_{t-1}, X_t) \leq U$, then $X_t \leftarrow X_{t-1}$. Do nothing otherwis
5. Return to Step 1.

MH algorithm works even without p's normalizing constant. He

1.5 Review: Metropolis-Hastings (2/2)

Alternative View: MH is a generic procedure to turn a

into a Markov chain converging to p.

The Choise of Proposal q

- Random Walk Metropolis (Metropolis et al., 1953): Unifor

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

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

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

- Metropolis-adjusted Langevin algorithm (MALA) (Besag, 19

$$q(-|X_t) := \text{ the transition probability of } X_t \text{ where } dX_t =$$

Hirofumi Shiba

1.6 Problem: Reversibility

Reversibility (a.k.a detailed balance):

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

In words:

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

→ Harder to explore the entire space

→ Slow mixing of MH

From the beginning of 21th century, many efforts have been ma

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

Lifted MH (Turitsyn et al., 2011)

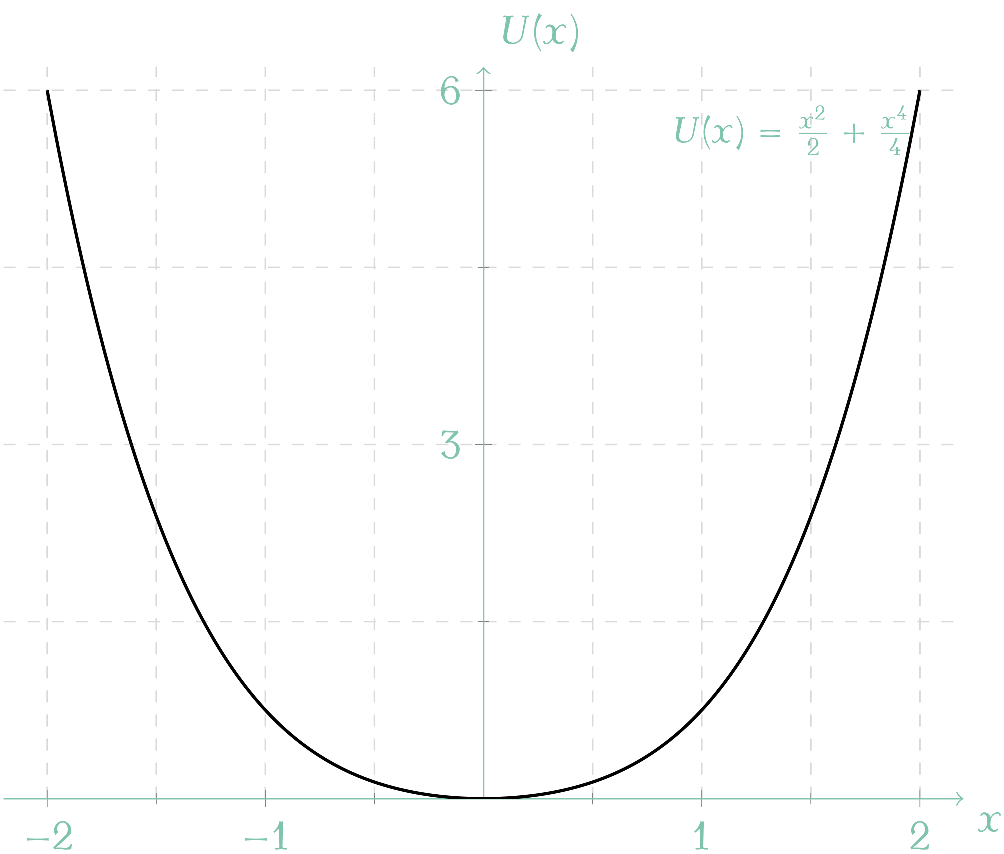
Input: Target p, two proposals $q^{(+1)}, q^{(-1)}$, and momentum σ

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"

$$\begin{array}{ccl} & \nearrow & \mathbb{R} \times \{+1\} \\ \mathbb{R} & & \\ & \searrow & \mathbb{R} \times \{-1\} \end{array}$$



$q^{(+1)}$: Only

$q^{(-1)}$: Only

→ Once g

continues

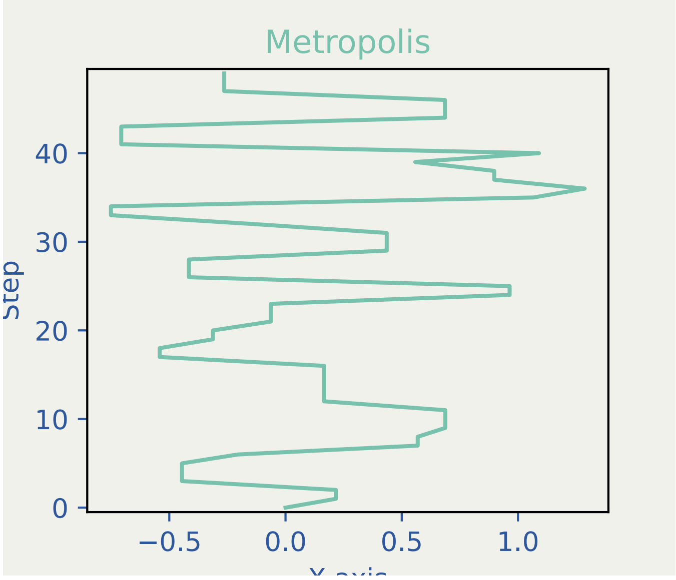
→ This is i

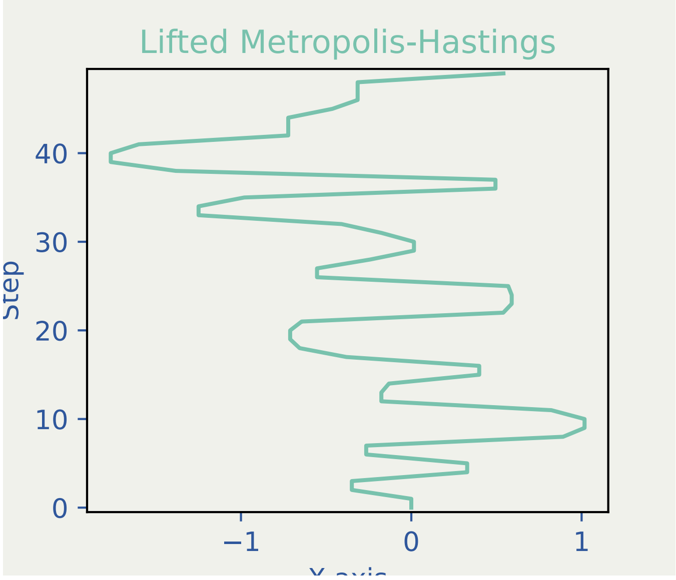
Probabil

$\neq$ F

1.9 Lifting (3/3)

Reversible dynamic of MH has ‘irreversified’



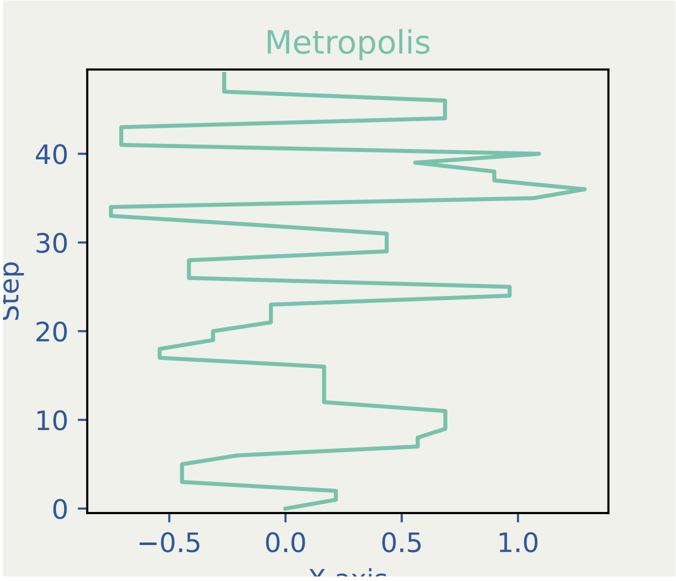


MH

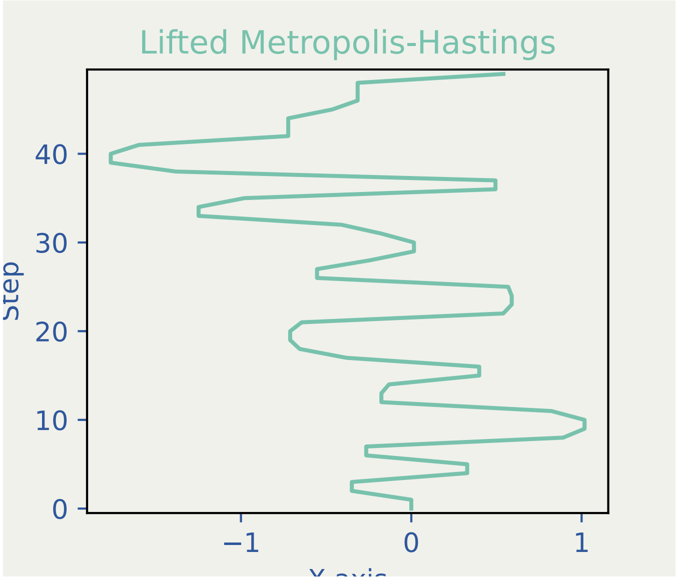
Lifted MH

*Irreversibility actually improves the efficiency of MCMC, as we

1.10 Comparison: MH vs. LMH vs. Zig-



MH



Lifted MH

Zig-Zag corresponds to the limiting case of
size of proposal q goes to zero, as we'll lear

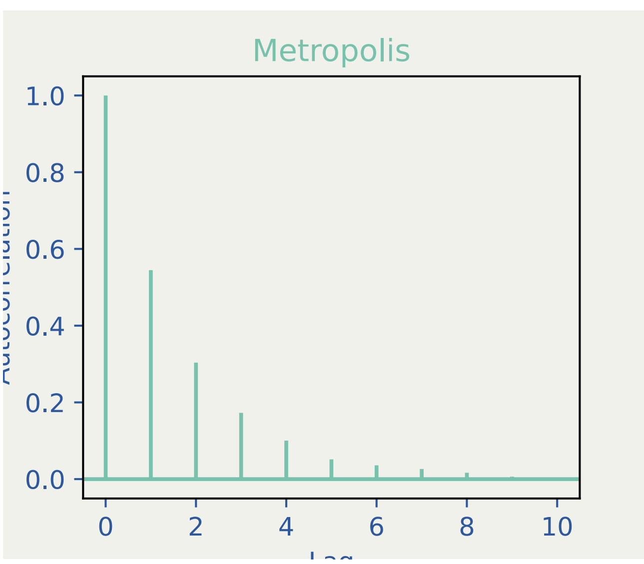
→ Zig-Zag has a maximum irreversibility.

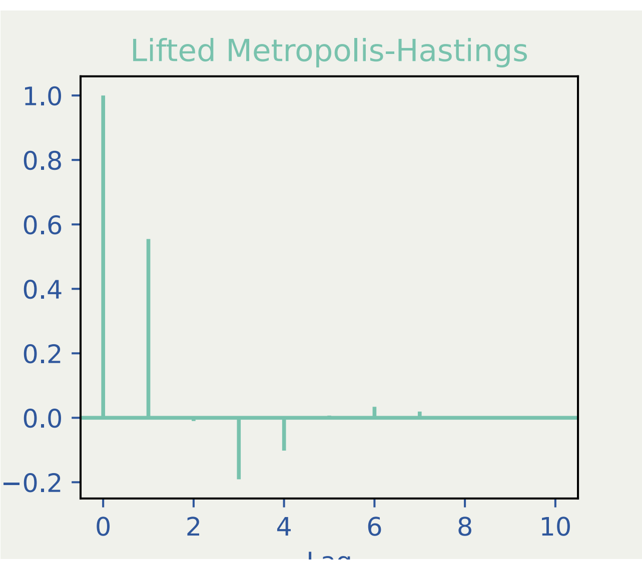
1.11 Comparison: MH vs. LMH vs. Zig-

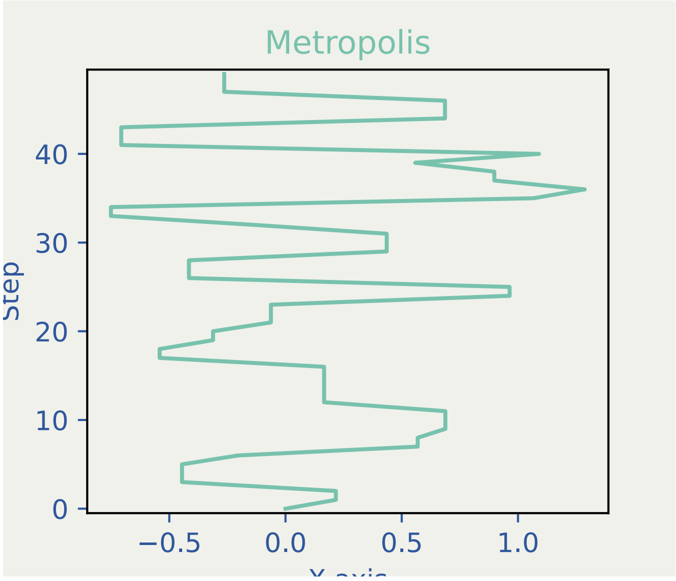
Irreversibility actually improves the efficienc

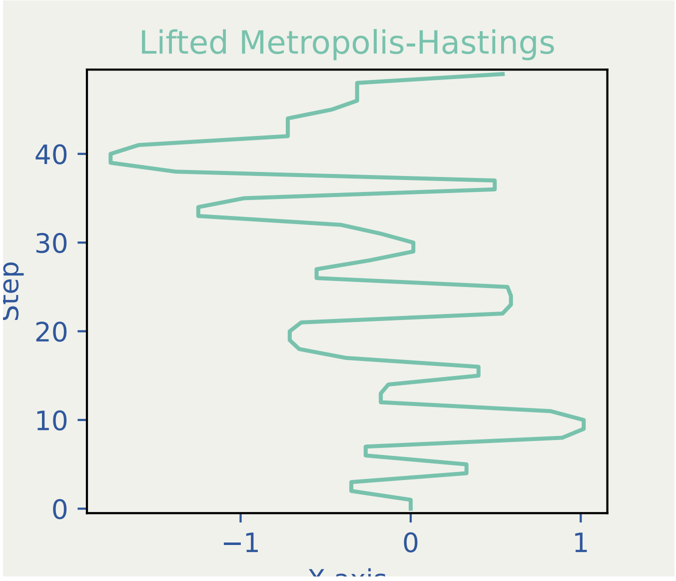
Faster decay of **autocorrelation** $\rho_t \approx \mathrm{Co}$

1. faster mixing of MCMC
2. lower variance of Monte Carlo estimates






Metropolis
Step
0
10
20
30
40
−0.5
0.0
0.5
1.0


Lifted Metropolis-Hastings
Step
0
10
20
30
40
−1
0
1

MH

Lifted MH

1.12 Review: MALA

Langevin diffusion: A diffusion process defined by the fo

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

Langevin diffusion itself converges to the target distribut

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

Two MCMC algorithms derived from **Lang**

ULA (Unadjusted Langevin Algorithm)

Use the discretization of (X_t). Discretization errors

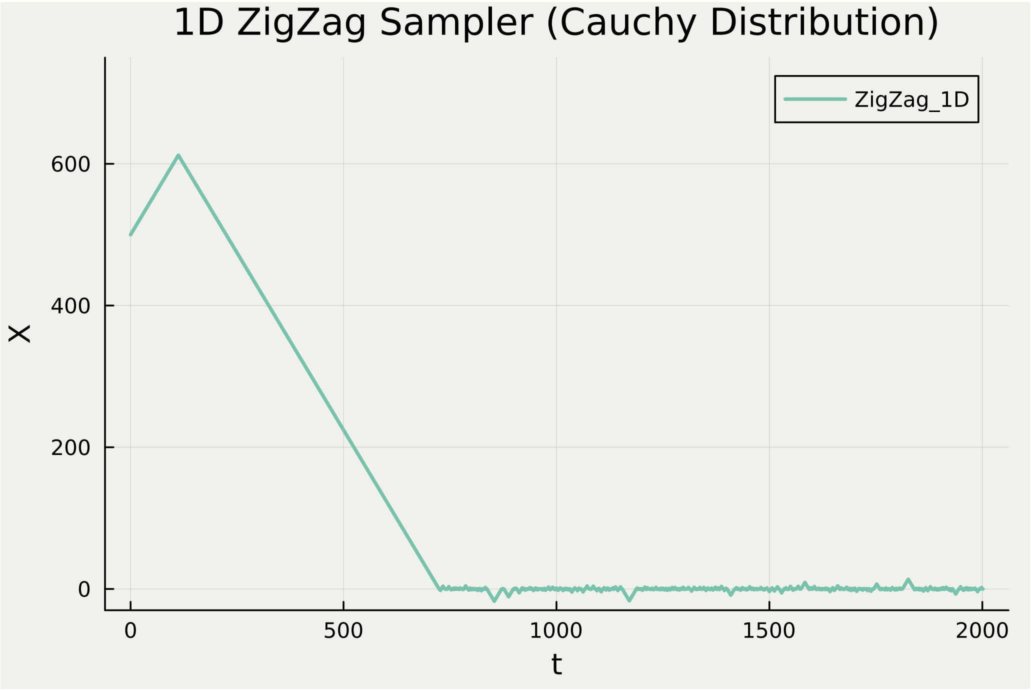
MALA (Metropolis Adjusted Langevin Algorithm)

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

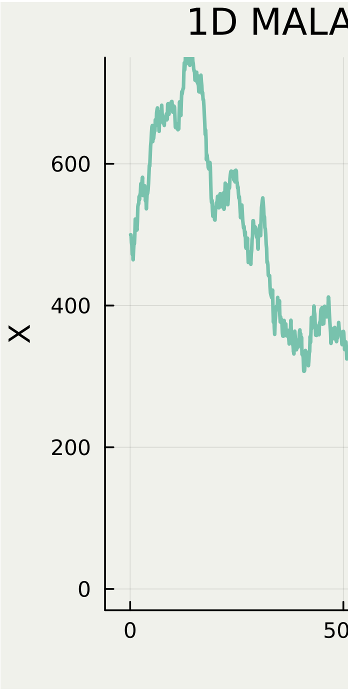
1. under fairly general conditions on p.

1.13 Comparison: Zig-Zag vs. MALA (

How fast do they go back to high-probability



Zig-Zag

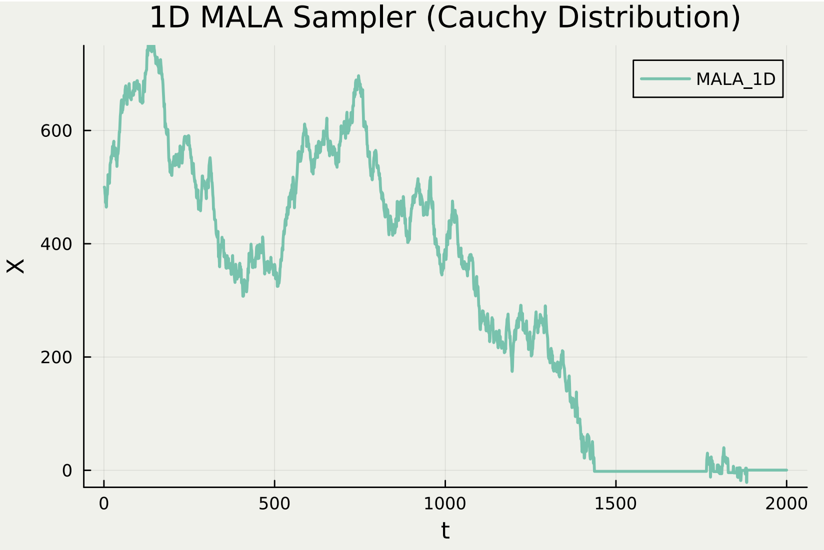


MALA

Irreversibility of Zig-Zag accelerates its conv

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

1.14 Comparison: Zig-Zag vs. MALA (



MALA trajectory

Caution: Fake C

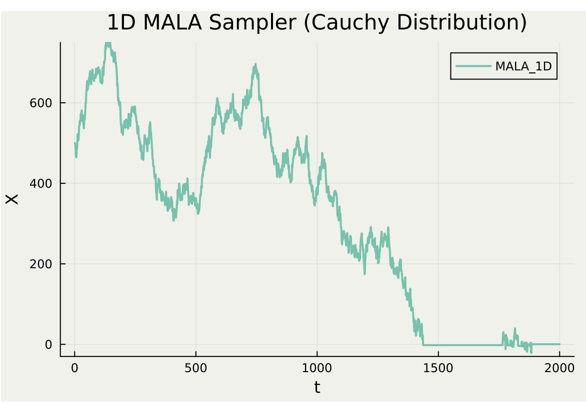
The left plot looks
actually is not.

MH, including MALA, is actually a <u>discrete-ti</u>

The plot is obtained by <u>connecting the poin</u>

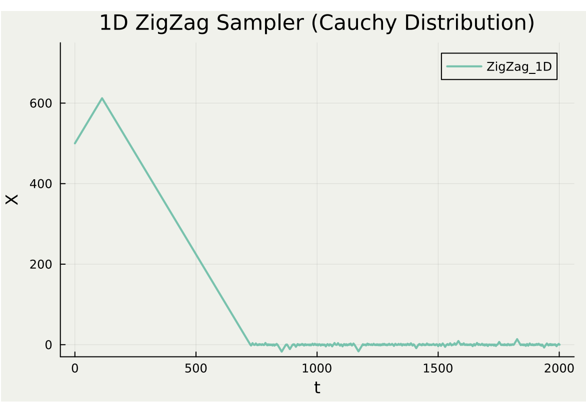
1.15 Comparison: Zig-Zag vs. MALA (

Monte Carlo estimation is also done differen



<u>MALA</u> outputs $(X_n)_{n\in[}$

$$\frac{1}{N}\sum_{n=1}^N f(X_n) \xrightarrow{N\to\infty}$$

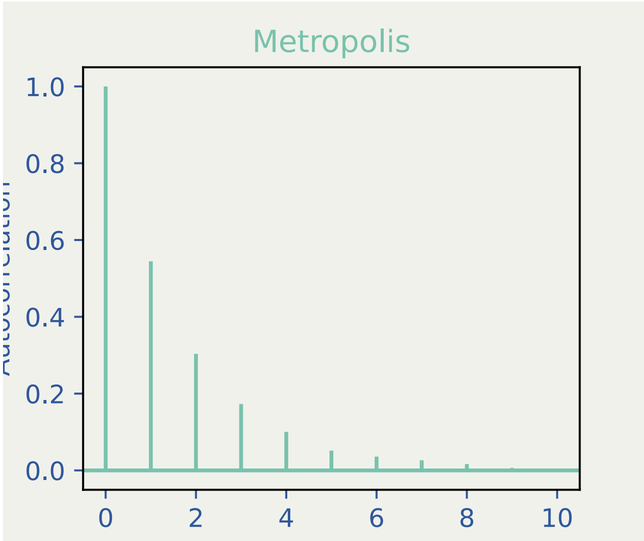


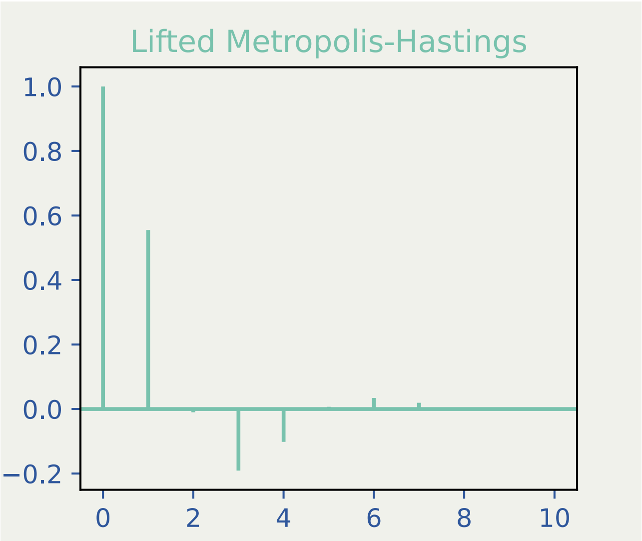
<u>Zig-Zag</u> outputs $(X_t)_{t\in}$

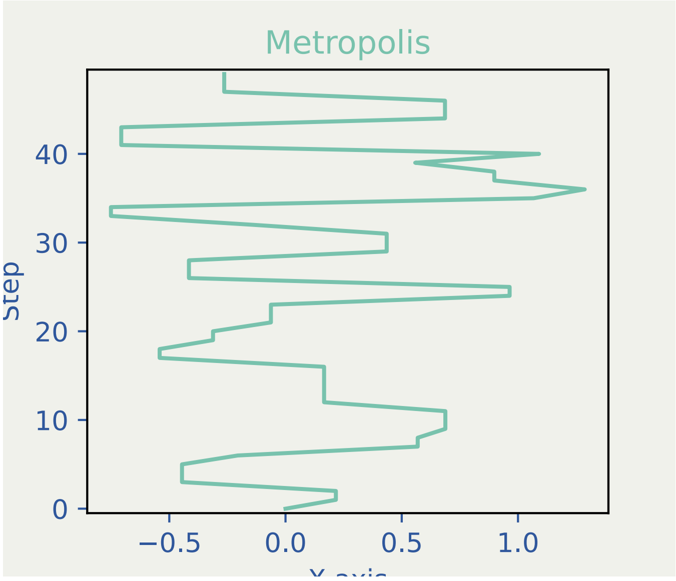
$$\int_0^T f(X_t)\,dt \xrightarrow{T\to\infty}$$

1.16 Recap of Section 1

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

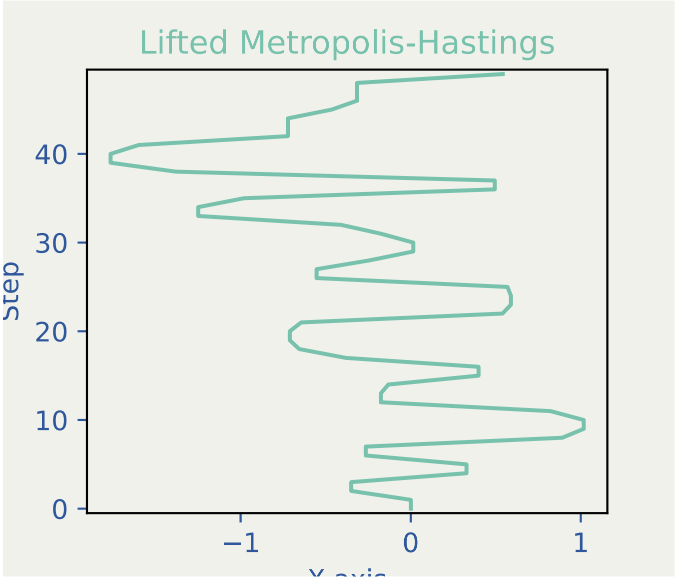






Metropolis
Step
X axis

MH



Lifted Metropolis-Hastings
Step
X axis

Lifted MH

2 The Algorithm: How t

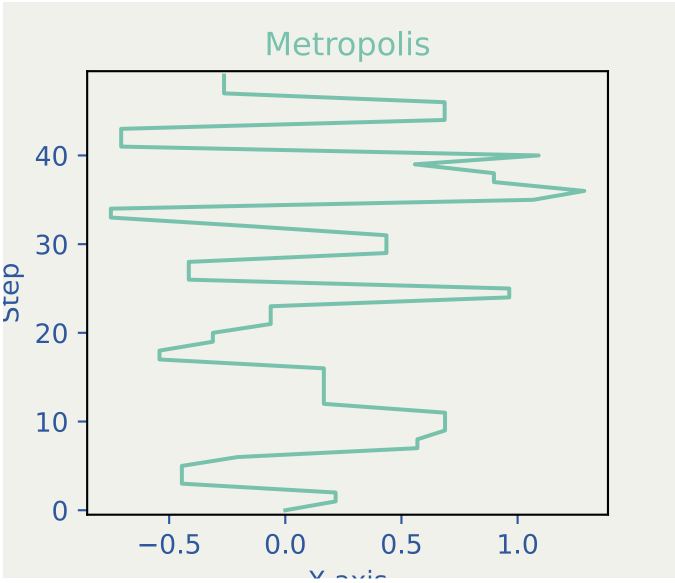
Fast and exact simulation of continuous traj

Hirofumi Shiba

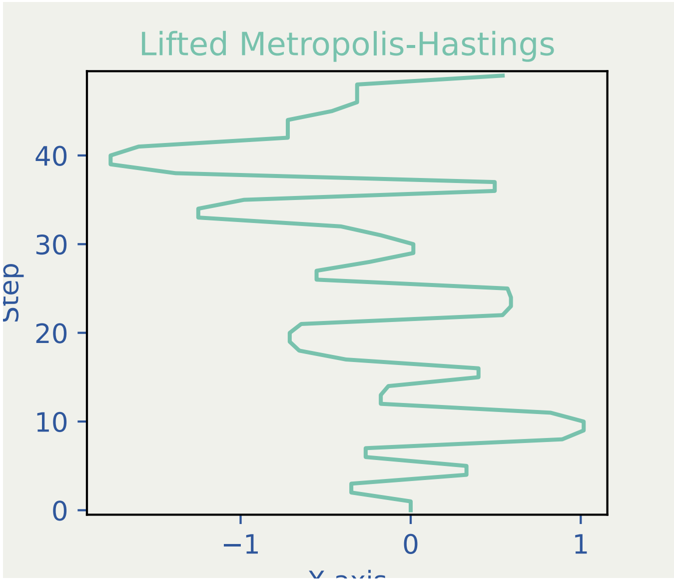
2.1 Review: MH vs. LMH vs. Zig-Zag (I

As we've learned before, Zig-Zag correspon

case of lifted MH as the step size of proposa



MH

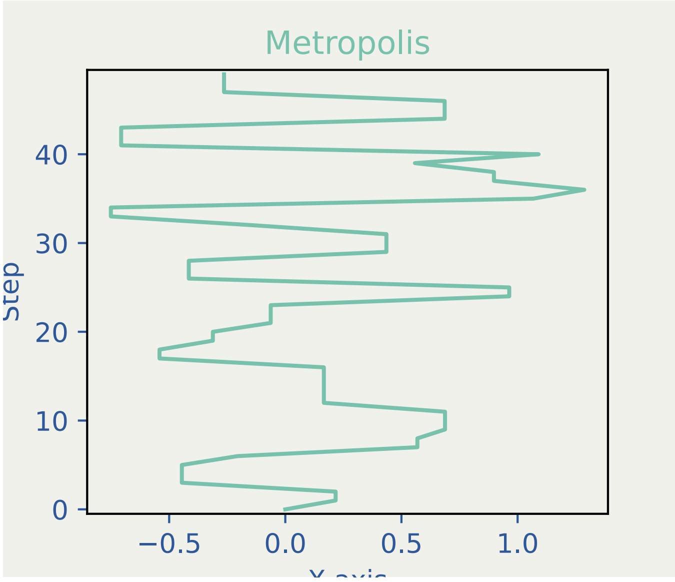


Lifted MH

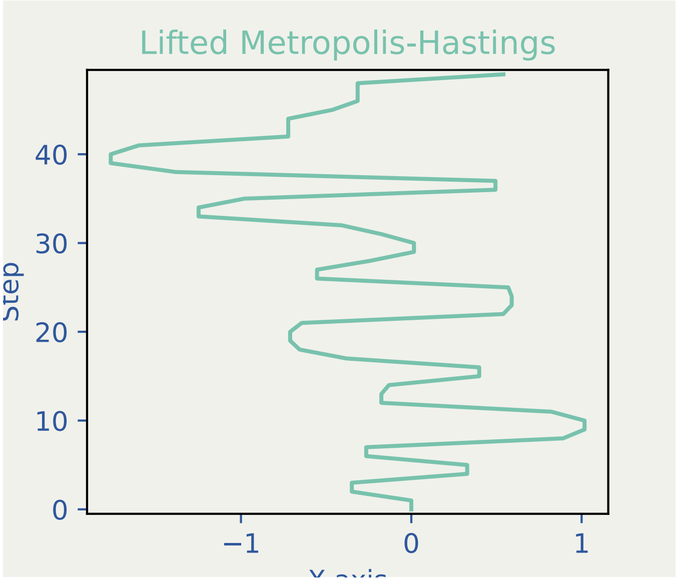
2.2 Review: MH vs. LMH vs. Zig-Zag (2

'Limiting case of lifted MH' means that we o

we should flip the momentum $\sigma \in \{$



MH



Lifted MH

2.3 Algorithm (1/2)

'Limiting case of lifted MH' means that we o

<u>we should flip the momentum</u> $\sigma \in \{$

(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
2. Linearly interpolate until time T_n:

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

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

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

2.4 Algorithm (2/2)

(Fundamental Property of Zig-Zag Sampler (1d)
2019**)**

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

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

ensures the Zig-Zag sampler converges to the target p, wher
negative function.

Its ergodicity is ensured as long as there exi
that[1]

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

1. With some regularity conditions on U. (See Bierkens, Rober

2.5 Core of the Algorithm

Given a rate function

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

how to simulate a corresponding Poisson po

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

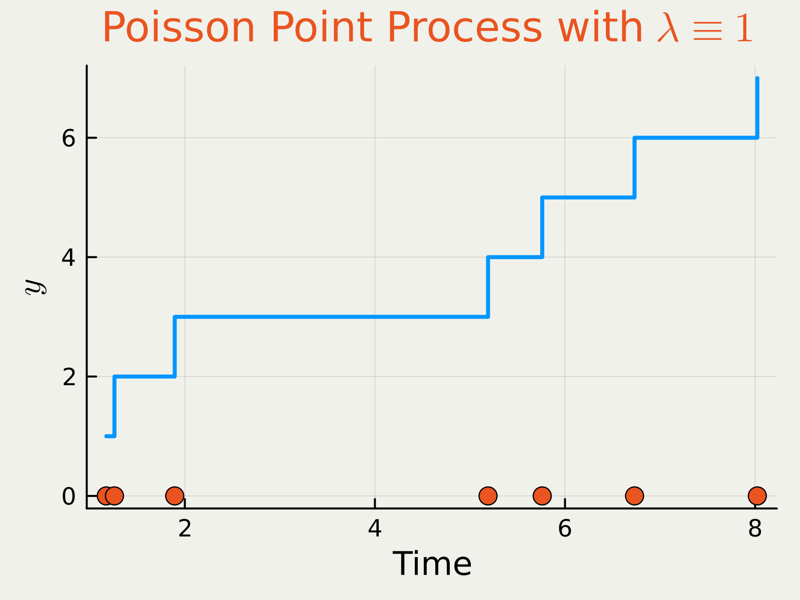
2.6 Simulating Poisson Point Process (

What is a Poisson Point Process with rate λ?

The number of points in $[0,t]$ follows a Poisson distribution v

$$N([0,t]) \sim \mathrm{Pois}\left(M(t)\right), \qquad M(t) := \int_0^t \lambda$$

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



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

- blue line: Poisso
- red dots: Poisso

satisfying $N_t = N$

2.7 Simulating Poisson Point Process (

Proposition (Simulation of Poisson Point Process

The first arrival time T_1 of a Poisson Point Process with rate

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

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

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

complicated.

→ Inverting M can be impossible.

→ We need more general techniques: Poiss

2.8 Poisson Thinning (1/2)

(Lewis and Shedler, 1979)

To obtain the first arrival time T_1 of a Poisson Point Process

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 int

3. Accept T with probability $\frac{m(T)}{M(T)}$.

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

2.9 Poisson Thinning (2/2)

In order to simulate a Poisson Point Process

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

we find a invertible upper bound M that sat

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

for all possible Zig-Zag trajectories $\{(x_s,\sigma_s$

2.10 Recap of Section 2

1. Continuous-time MCMC, based on PDMI
   different algorithm and strategy.
2. To simulate PDMP is to simulate Poisson
3. The core technology to simulate Poisson
   Poisson Thinning.
4. Poisson Thinning is about finding an uppe
   tractable inverse M^{-1}; Typically a polynor
5. The upper bound M has to be given on a

3 Proof of Concept: How It?

Quick demonstration of the state-of-the-art toy example.

3.1 Review: The 3 Steps of Zig-Zag Sa

Given a target p,

1. Calculate the negative log-likelihood $U(x$
2. Fix a refresh rate $\gamma(x)$ and compute the

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

3. Find an invertible upper bound M that sa

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

3.2 Model: 1d Gaussian Mean Reconstr

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

$$U(x)=-1$$

$$=\frac{x^2}{2\rho^2}$$

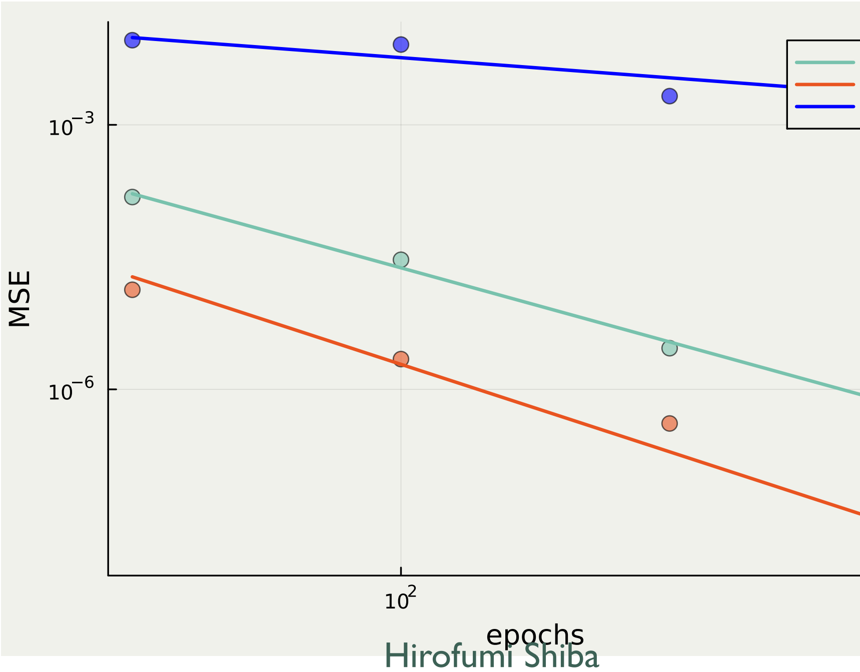
$$U'(x)=\frac{x}{\rho^2}$$

$$U''(x)=\frac{1}{\rho^2}$$

3.3 Menu

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

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



3.4 Simple Zig-Zag Sampler with $\gamma \equiv 0$

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

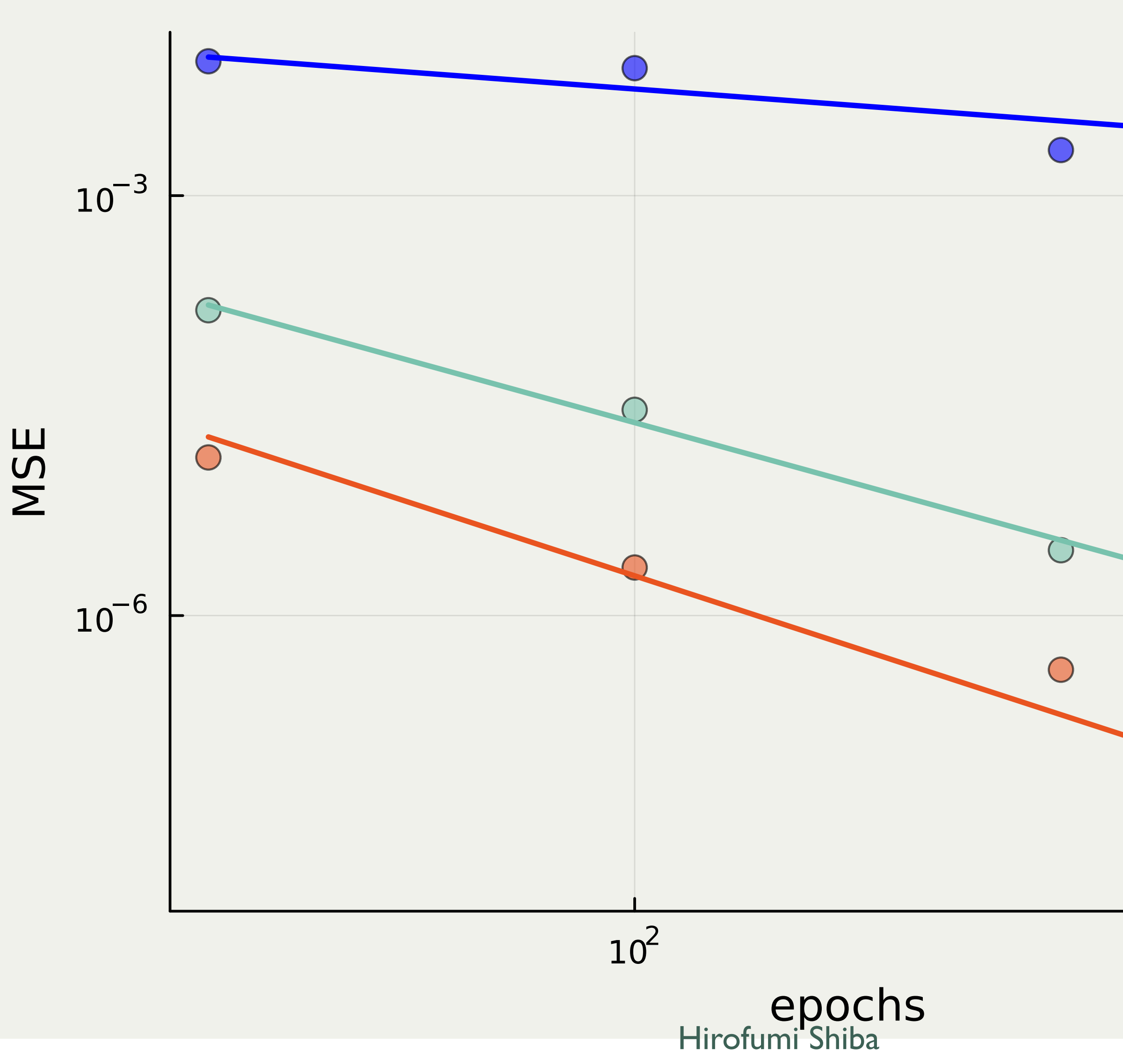
$$
\begin{aligned}
m(t) &= \int_0^t \lambda(x_s, \sigma_s)\, ds = \int_0^t \Bigg(\sigma U'(x \\
&\le \left(\frac{\sigma x}{\rho^2} + \frac{\sigma}{\sigma^2} \sum_{i=1}^n (x - y_i) + t \Bigg(\right. \\
&=: (a + bt)_+ = M(t),
\end{aligned}
$$

where

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

3.5 Result: 1d Gaussian Mean Reconstr

We generated 100 samples from $\mathrm{N}(x_0, \sigma^2)$ v



3.6 MSE per Epoch: The Vertical Axis

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

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

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 bour complicated / demanding.

Therefore, we are trying to automate the w

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). Co
of Knowledge in
Grenander and M
Statistical Society.
56(4), 591–592.
Bierkens, J., Fearnhead
The Zig-Zag Proc
Sampling for Baye
Annals of Statistics
Bierkens, J., Grazzi, S.,
G. O. (2020). The
Proceedings of the
on Machine Learni
Bierkens, J., Roberts, G
Ergodicity of the
Applied Probability
Bouchard-Côté, A., Vo

(2018). The boun
nonreversible rej
monte carlo meth
Statistical Associat

Corbella, A., Spencer,
(2022). Automati
Statistics and Com

Dai, H., Pollock, M., ar
Carlo Fusion. *Jou*
56(1), 174–191.

Davis, M. H. A. (1984)
markov processe
diffusion stochast
Statistical Society.
46(3), 353–388.

Davis, M. H. A. (1993)
optimization, Vol.

Duane, S., Kennedy, A
Roweth, D. (1987
Letters B, *195*(2),

Fearnhead, P., Grazzi,
G. O. (2024). Sto
deterministic mo
Grazzi, S. (2020). Piec
carlo.
Hastings, W. K. (1970)
methods using m
applications. *Biom*
Lewis, P. A. W., and Sh
Simulation of non
processes by thin
Quarterly, *26*(3), 4
Metropolis, N., Rosen
N., Teller, A. H., a
of state calculatio
machines. *The Jou*
1087–1092.
Pagani, F., Chevallier, A
Cotter, S. (2024).
general models. S

61.

Peters, E. A. J. F., and

Rejection-free m

general potential

Scott, S. L., Blocker, A

H. A., George, E.

(2016). Bayes an

monte carlo algo

Management Scie

Management, *11*,

Srivastava, S., Cevher,

(2015). WASP: S

of subset posteri

N. Vishwanathan,

eighteenth interna

intelligence and st

San Diego, Califo

Sutton, M., and Fearn

convex PDMP-ba

Computational an

1425–1435.
Turitsyn, K. S., Chertk
Irreversible Mont
Efficient Sampling
Phenomena, *240*(5
Welling, M., and Teh, Y
via stochastic gra
Proceedings of the
on international co
pages 681–688. N
Omnipress.

Appendix: Scalability by Subsampling

Construction of ZZ-CV (Zig-Zag with Cont

Hirofumi Shiba

3.8 Review: 1d Gaussian Mean Reconst

U' has an alternative form:

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

where

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

→ We only need one sample y_i to evaluate

3.9 Randomized Rate Function

Instead of

$$\lambda_{\text{ZZ}}(x, \sigma) = \left(\sigma U'(x) \right)_+$$

we use

$$\lambda_{\text{ZZ-CV}}(x, \sigma) = \left(\sigma U'_I(x) \right)_+, \qquad I \sim$$

Then, the latter is an unbiased estimator of the former

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

3.10 Last Step: Poisson Thinning

Find an invertible upper bound M that satis

$$\int_0^t \lambda_{\text{ZZ-CV}}(x_s,\sigma_s)\,ds =: m_I(t) \le M(t),$$

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

3.11 Upper Bound M with Control Va

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, wit

paramete

$$m_i(t)$$

where

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

And m_i is redefined as

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

3.12 Subsampling with Control Variate

Zig-Zag sampler with the random rate funct

$$\lambda_{\text{ZZ-CV}}(x,\sigma) = \left(\sigma U_I'(x)\right)_+, \qquad I$$

and the upper bound

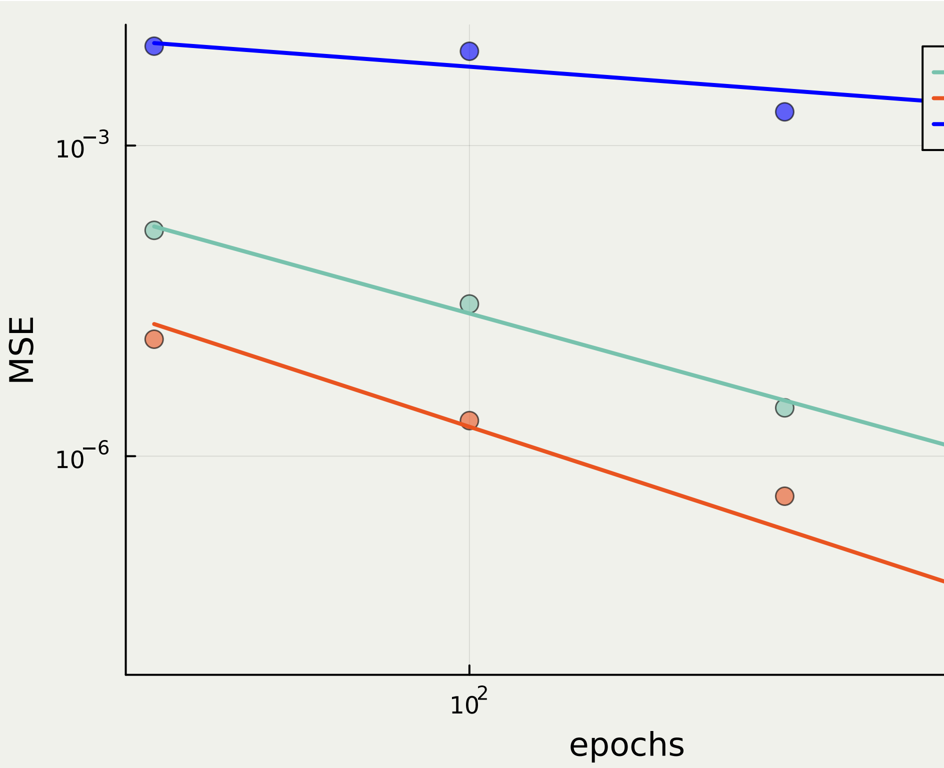
$$M(t) = a + bt$$

is called Zig-Zag with Control Variates (Bier al., 2019).

3.13 Zig-Zag with Control Variates

1. has $O(1)$ efficiency as the sample size n g

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
for large data.

1. Devide-and-conquer

   Devide the data into smaller **chunks** and
   each **chunk**.

2. Subsampling

   Use a subsampling estimate of the likeliho
   require the entire data.

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

Devide the data into smaller chunks and run
chunk.

Unbiased?	**Method**
✕	WASP
✕	Consensus Monte Carlo
✓	Monte Carlo Fusion

3.16 Scalability (3/3) by Subsampling

Use a subsampling estimate of the likelihooc
require the entire data.

Unbiased?	Method
✕	Stochastic Gadient MCMC
✓	Zig-Zag with Subsampling
✕	Stochastic Gradient PDMP