IMPERIAL

IMPERIAL COLLEGE LONDON

DEPARTMENT OF MATHEMATICS

BSC RESEARCH PROJECT

Sampling from log-concave distributions through Markov Chain Monte Carlo methods

Analysis and evaluation of Unadjusted and underdamped Langevin algorithms

---

Author:
Peiyi Zhou

Supervisor(s):
Dr. Ömer Deniz Akyildiz

Submitted in partial fulfillment of the requirements for the BSc in Mathematics at Imperial College London

June 10, 2024

Abstract

This report aims to discuss the behaviour of Markov Chain Monte Carlo (MCMC) methods for dealing with sampling problems, involving (i) the implementation of unadjusted Langevin algorithm derived from overdamped Langevin stochastic differential equation, and (ii) underdamped Langevin samplers derived from underdamped Langevin stochastic differential equation, with two different discretisations. For visualisation and evaluation of each algorithm's sampling performance, we focus on analysing the behaviour for sampling from the 1D Gaussian distributions, as an example of a log-concave distribution, which will be used as the target distribution throughout this report. Particularly, we will derive analytic expressions for the stationary distributions of these discretisations, some of which are known in the literature whereas some are not. Of particular importance are the stationary distributions of underdamped Langevin algorithms, which are not given as examples in the literature before, which we find important to derive. We will also discuss the convergence properties of these algorithms.

Acknowledgments

I would like to show my deepest gratitude to my supervisor, Dr. Deniz Akyildiz, for his unwavering support and insightful feedback throughout the duration of this research. My initial interest in stochastic simulation and computational statistics was sparked by his comprehensive in-class lectures, and this interest was further solidified through his thoughtfully designed extracurricular exercises and engaging office-hour discussions. His expertise and encouragement were essential and crucial to the completion of this work.

I also want to extend my heartfelt thanks to all the authors, whose papers and works I referenced in the fields of Markov chain Monte Carlo methods and various other statistical areas have been invaluable and greatly benefited my research. Their diligent research and groundbreaking contributions have provided solutions to many of the challenges I encountered, and have greatly enriched the depth and breadth of my own research. Their persistent dedication to advancing the field has been an inspiration and a crucial resource throughout my study.

Finally, I am deeply indebted to my family and friends for their unconditional support and encouragement throughout this journey. Their patience and understanding were my greatest source of strength.

Plagiarism statement

The work contained in this thesis is my own work unless otherwise stated.

Signature: Peiyi Zhou

Date: June 10, 2024

Contents

1 Introduction **5**

1.1 Background and history 6

1.2 Study Objectives and Contributions 7

1.3 Report Outline 8

2 Unadjusted Langevin MCMC method **9**

2.1 The rise of unadjusted Langevin algorithm (ULA) 10

2.2 Stationarity analysis 11

2.3 ULA performance analysis 13

2.3.1 An introduction of Distance measure 13

2.3.2 ULA performance: stationary variance control 15

2.3.3 ULA performance: rate of convergence 17

2.3.4 Summary 21

3 Underdamped Langevin MCMC method **23**

3.1 Euler-Maruyama discretisation on underdamped Langevin SDE 24

3.2 Stationary analysis 26

3.2.1 Vector autoregressions 26

3.2.2 Mean of $\{\mathbf{Z}_t\}_{t\geq 0}$ 28

3.2.3 Autocovariance sequence (acv) of $\{\mathbf{Z}_t\}_{t\geq 0}$ 29

3.2.4 Variance of $\{\mathbf{Z}_t\}_{t\geq 0}$ 30

3.3 Performance of underdamped Langevin diffusion (ULD) 33

3.4 New discretisation on underdamped Langevin SDE 38

4 Conclusion **43**

4.1 Future perspectives 43

Chapter 1

Introduction

Markov chain Monte Carlo (MCMC) methods are a set of algorithms aiming at sampling from a target probability distribution. They are called as 'Markov chain' since such algorithms would construct a Markov chain, or more commonly, a stochastic process, that simulates sampling from the target distribution. Although correlated samples would be generated at first, after burnin process, one would obtain samples which are approximately uncorrelated and behave approximately according to the sampling distribution.

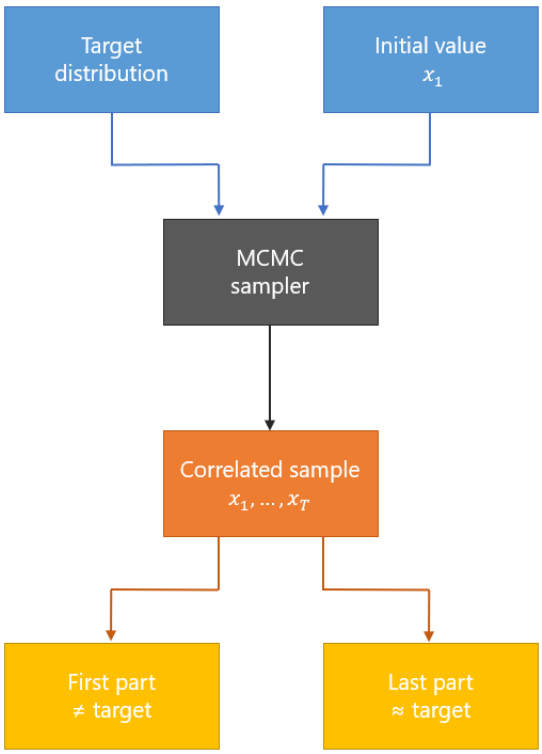


Figure 1.1 Flowchart showing how MCMC works, with burnin samples as the 'first part'. [1]

In this report, sampling problems on log-concave distributions by using MCMC methods will be discussed and evaluated, where 'log-concave' means our target sampling distribution would be in the form of $\exp(-U(x))$, with $U(x)$ being as a convex function (so leads $-U(x)$ being concave). For visualisation, a common choice for convex function $U(x)$ would be the quadratic function with a positive leading coefficient, which would further lead our log-concave sampling distribution be a 1D Gaussian distribution, denoted as $\mathcal{N}(x; \mu, \sigma^2)$ throughout this report. Meanwhile, since we would sample from stochastic processes, such MCMC algorithms should also be based on stationarity, which means the constructed sequence must be stationary.

1.1 Background and history

Langevin Monte Carlo (LMC), inspired by the Langevin dynamics, is widely recognised as a set of sampling algorithms that behaves similarly to the 'gradient descent optimisation algorithms'. For the gradient descent optimisation, we minimise an objective function $U : \mathbb{R}^d \to \mathbb{R}$ by taking steps in the direction of $-\nabla U$; whereas for Langevin Monte Carlo samplers, samples are generated from a target distribution $p_\star \propto \exp(-U)$ by moving, on average, in the direction of $-\nabla U$ [2]. These types of sampling algorithms are essential in various computational methods, because both gradient descent and Langevin Monte Carlo work well for tasks like global optimisation of convex objective functions and for sampling from log-concave distributions (so both requires U being as a convex function) [2]. The concept of using continuous-time Langevin dynamics for sampling a specified target distribution is discussed in [3] and [4]. However, it was fully developed in [5], where the Metropolis adjusted Langevin algorithm (MALA) was initially proposed and analysed. Specifically, the MALA leverages the Langevin stochastic differential equation (SDE), which converges to the target distribution exponentially fast, ensuring accurate sampling result, even in high-dimensional spaces.

The unadjusted Langevin algorithm (ULA), which is derived from the overdamped Langevin SDE [6], builds on the same principle as MALA, but just skips the Metropolis Hastings acceptance step to reduce computational cost, making it suitable for complex high dimensional sampling problems, where applying the Metropolis step can lead a great computation cost. However, even in a Gaussian setting, such simplification can also introduce a bias [2], that must be managed carefully to maintain the algorithm's effectiveness. Thus, we will use cer-

tain distance measures introduced in information theory for measuring the performance and convergence of such algorithms quantitatively.

Besides just as ULA is seen as an MCMC sampling algorithm from the overdamped Langevin SDE, the underdamped Langevin MCMC scheme, derived from the underdamped Langevin SDE system can be viewed as 'a version of Hamiltonian Monte Carlo (HMC) which has been observed to outperform overdamped Langevin MCMC methods' [7], and used as a more common sampling algorithm in many areas. In some sampling problems, the underdamped Langevin algorithm serves as an 'accelerated' version of the overdamped Langevin system, which typically achieves better convergence rates by incorporating a momentum process $\mathbf{V}_t$. Moreover, this improvement applies whether the target distribution is log-concave or not [6], [7], [8].

Therefore, evaluations of the MCMC methods mentioned previously are valuable and important for practitioners dealing with certain sampling problems. The analysis of both the unadjusted and underdamped Langevin algorithms reveals their specific advantages and limitations. As stated, while ULA offers computational efficiency compared with MALA, it requires careful handling to mitigate bias, especially in high-dimensional (Gaussian) settings. Meanwhile, although the underdamped Langevin algorithm typically achieves faster convergence for both log-concave and non-log-concave target distributions, examining its various discretisations (which will shown in Chapter 3 later) is also worthwhile. By analysing and evaluating the nuances of these MCMC methods, practitioners are able to choose and implement the most appropriate and efficient algorithm, with certain parameters, to achieve more accurate sampling results from the target distribution, hence advancing computational methods in various fields.

1.2 Study Objectives and Contributions

We will analyse sampling algorithms from theoretical and empirical perspectives in this report. Specifically, we (i) derive analytical expressions of the stationary distributions of discretised algorithms, which are approximately sampling from the target distribution, but have a bias due to the lack of Metropolis correction; (ii) analyse the asymptotic limit of discretisations of both unadjusted and underdamped Langevin algorithms, where the latter one has not been given in the literature before, to the best of our knowledge; (iii) compare the sampling performance and convergence behaviour of unadjusted and underdamped Langevin samplers, and also the

performance and bias with respect to the different stepsizes of the algorithms. In particular, we implement such comparisons and discussions by using distance measures in information theory: Kullback-Leibler (KL) divergence and Wasserstein distance, and computing these distances between the target distribution and the stationary, marginal distribution to make justifications.

1.3 Report Outline

In Chapter 2, we would first introduce and discuss unadjusted Langevin algorithm (ULA) from overdamped stochastic differential equations by applying Euler-Maruyama discretisation. Evaluations on the sampling performance of ULA would be carried through using Kullback-Leibler divergence and Wasserstein distance, under certain stationarity conditions. Then, in Chapter 3, we will further investigate the underdamped Langevin diffusion (ULD), or called as the underdamped Langevin algorithm, in short as underdamped LA in this report, from underdamped stochastic differential equations. We would discuss two different discretisations on such system of SDEs, with the stationarity analysis and the evaluations on the sampling performance.

Chapter 2

Unadjusted Langevin MCMC method

There are many different algorithms for implementing MCMC methods. Let us assume that we want to sample a target distribution $p_\star$ where

$$p_\star(x)\mathrm{d}x \propto \exp(-U(x))\mathrm{d}x \tag{2.1}$$

An efficient method for obtaining a sample from (2.1) is simulating the overdamped Langevin stochastic differential equation (SDE) [9] given by

$$\mathrm{d}X_t = -h(X_t)\mathrm{d}t + \sqrt{2}\mathrm{d}B_t = -\nabla U(X_t)\mathrm{d}t + \sqrt{2}\mathrm{d}B_t \tag{2.2}$$

with the settings that

- A random initial condition, or a starting point, $X_0 = x_0$.
- The function $h = \nabla U$, as the gradient of U.
- U is strongly convex and Lipschitz-smooth (L-smooth). This means $U : \mathbb{R}^d \to \mathbb{R}$ should be continuously differentiable, and its gradient ∇U should be Lipschitz continuous, with Lipschitz constant L [9]
$$\|\nabla U(x) - \nabla U(y)\| \leq L\|x - y\|$$
- $\{B_t\}_{t\geq 0}$ is a d-dimensional (standard) Brownian motion.
- $\{X_t\}_{t\geq 0}$ is the Markov chain generated for sampling the unnormalised density of the target distribution, i.e. $\exp(-U(x))$. Each state in the chain represents a sample from (2.1).

Remark 2.0.1. *Under these conditions, the convergence to the target distribution is guaranteed and exponentially fast.*

2.1 The rise of unadjusted Langevin algorithm (ULA)

With the overdamped SDE (2.2), the discrete time Markov chain associated with the unadjusted Langevin algorithm (ULA) is obtained by the Euler–Maruyama discretisation scheme of the Langevin SDE defined for $t \in \mathbb{N}$ by [10]

$$X_{t+1} = X_t - \gamma \nabla U(X_t) + \sqrt{2\gamma} Z_{t+1}, \qquad X_0 = x_0 \tag{2.3}$$

where $x_0 \in \mathbb{R}^d$ as the starting point, the stepsize $\gamma > 0$ and $(Z_t)_{t\in\mathbb{N}}$ are i.i.d. standard d-dimensional Gaussian variables, i.e. $Z_t \sim \mathcal{N}(0, \mathbf{I}_d)$. Another way for writing this is to set $v_t = \sqrt{2\gamma} Z_t \sim \mathcal{N}(0, 2\gamma \mathbf{I}_d)$,

$$X_{t+1} = X_t - \gamma \nabla U(X_t) + v_{t+1}, \qquad X_0 = x_0 \tag{2.4}$$

Note that with the initial value x_0 and the noise term in Gaussian distribution $\mathcal{N}(0, 2\gamma \mathbf{I}_d)$, our chain generated would also be in the Gaussian distribution. Therefore, in computational statistics, ULA works as the Algorithm 1.

Algorithm 1: Unadjusted Langevin algorithm (ULA)

Input: N, the number of samples we want to take
$X_0 = x_0$, as the initial value
n, the number of burnin samples need to discard

Output: Approximately uncorrelated samples of the target distribution $p_\star$

for $t = 1, 2, \cdots, N$ **do**
  Take a random sample x' from the proposal

$$X' \sim q(x'|x_{t-1}) = \mathcal{N}(x'; x_{t-1} - \gamma \nabla U(x_{t-1}), 2\gamma \mathbf{I}_d)$$

  which is same as $x' = x_{t-1} - \gamma \nabla U(x_{t-1}) + v_t$, for $v_t \sim \mathcal{N}(\mathbf{0}, 2\gamma \mathbf{I}_d)$;
  Update the new state by $x_t = x'$.
end
return Discard first n burnin samples and return the remaining samples.

As discussed above, our $U(x)$ in ULA should also satisfy being as L-smooth. In this report, we may just consider the one-dimensional case, and deal $U(x)$ in the form of a polynomial, as

polynomials are easier for analysis. In order to let polynomial $U(x)$ be L-smooth, the highest order of $U(x)$ would be 2, so that we may write $U(x) = \frac{1}{2}ax^2 + bx + c$, so $\nabla U(x) = ax + b$ in a linear form. Meanwhile, we also need $a > 0$ to keep the strong convexity.

Remark 2.1.1 (Extension: higher order sampling problem)**.** *Note that if we deal with higher dimensions, i.e. sampling on $\mathbb{R}^p$, with $p > 1$, then such quadratic polynomial setting would be in the form as $U(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{b}^T\mathbf{x} + \mathbf{c}$, with $\mathbf{A}$ being positive definite. Intuitively, our results in one dimension can be extended into multidimensions in a similar way.*

Setting $U(x)$ as such quadratic polynomial also leads our target density be a normal distribution as well, since

$$p_\star(x) \propto \exp\left(-\left(\frac{1}{2}ax^2 + bx + c\right)\right) \propto \exp\left(-\frac{a\left(x + \frac{b}{a}\right)^2}{2}\right) \implies p_\star(x) = \mathcal{N}\left(x; -\frac{b}{a}, \frac{1}{a}\right) \quad (2.5)$$

Also, put this gradient of $U(x)$ back to our ULA chain in (2.4), we would have

$$X_{t+1} = X_t - \gamma(aX_t + b) + v_{t+1} = (1 - a\gamma)X_t - \gamma b + v_{t+1}, \qquad X_0 = x_0 \quad (2.6)$$

This chain generated by ULA, as a discrete time series, is indeed in the form of an autoregressive process with order 1 (AR(1) process), and we may be interested in the stationary distribution for such process. The next section carries the stationarity analysis on this process.

2.2 Stationarity analysis

We may first recall the notion of stationary process.

Definition 2.2.1 (Stationary/covariance-stationary/weakly stationary)**.** *A process is said to be covariance stationary, or weakly stationary, if its first and second moments are time invariant. i.e. for a covariance-stationary process Y_t, we would have*

$$\begin{aligned} &\mathbb{E}(Y_t) = \mathbb{E}(Y_{t-1}) \qquad \forall t \\ &\operatorname{cov}(Y_t, Y_{t+\tau}) = s_\tau \qquad \forall t, \tau > 0 \\ &\operatorname{var}(Y_t) = s_0 < \infty \qquad \forall t \end{aligned}$$

where s_τ ($\tau \geq 0$) is called the autocovariance sequence (acvs) that is independent of t. [11]

Remark 2.2.2. *Throughout this report, if not necessarily stated, the word 'stationary' or 'stationarity' always indicate the covariance-stationarity or weakly stationarity.*

Theorem 2.2.3 (Conditions for stationarity, AR(p) process)**.** *For an AR(p) process written in the form as*

$$X_t = \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + \mu + \epsilon_t$$

where $\epsilon_t \sim \mathcal{N}(0, \sigma_\epsilon^2)$ *as the white noise process. We say this process is stationary/stable if the roots of the lag polynomial, also called the characteristic polynomial*

$$1 - \phi_1 z - \cdots - \phi_p z^p = 0$$

lie outside the unit circle/disk $|z| < 1$*. [11]*

Therefore, for the sequence (2.6) generated by ULA, we have the characteristic equation

$$1 - (1 - a\gamma)\, z = 0 \implies z = \frac{1}{1 - a\gamma}$$

Thus our Markov chain (2.6) generated by ULA is stationary if we have appropriate stepsize γ to satisfy

$$\left| \frac{1}{1 - a\gamma} \right| > 1 \tag{2.7}$$

Since the ULA works based on the recursion of the chain (2.6), it is really important to choose the appropriate stepsize γ that satisfies (2.6) to hold the stationarity, otherwise as time increases we always sample from different distribution at each time.

After choosing γ satisfying (2.7) to hold the stationarity of the chain, we may now work on its mean and variance. Assume that we now work on the one-dimensional distributions. By Definition 2.2.1, we can derive its mean by applying expectation on both sides of (2.6). If we set $\mathbb{E}(X_t) = \mu$, then

$$\mu = (1 - a\gamma)\, \mu - \gamma b \implies \mathbb{E}(X_t) = \mu = -\frac{b}{a} \tag{2.8}$$

Again, by Definition (2.2.1), we know that this stationary process (2.6) would have a time invariant variance. Therefore, intuitively, we can apply the variance on both sides. If we assume

$\mathrm{var}(X_t) = \sigma^2$, then

$$\sigma^2 = (1-a\gamma)^2\sigma^2 + 2\gamma \implies \mathrm{var}(X_t) = \sigma^2 = \frac{2\gamma}{1-(1-a\gamma)^2} = \frac{2}{a(2-a\gamma)} \tag{2.9}$$

Therefore, for the chain $\{X_t\}_{t\geq 0}$ generated by ULA, it would have a stationary distribution of

$$X \sim \mathcal{N}\left(\mu, \sigma^2\right) = \mathcal{N}\left(-\frac{b}{a}, \frac{2}{a(2-a\gamma)}\right) \tag{2.10}$$

This is biased due to the absence of Metropolis step, but compared with the target distribution $p_\star$ as in (2.5), this distribution (2.10) of chain generated by ULA is close when γ is not too big.

2.3 ULA performance analysis

Intuitively, not all settings in ULA works well for our sampling problem with respect to the target distribution $p_\star$ in (2.5). The choice of stepsize γ would affect the performance of ULA, and in this section we would investigate this.

2.3.1 An introduction of Distance measure

To start this section, we first introduce two measures that quantitatively determine 'the difference between one probability distribution to another'. These measures are

- Kullback-Leibler divergence/relative entropy
- 2-Wasserstein distance

Definition 2.3.1 (Kullback-Leibler divergence/relative entropy)**.** *Consider some unknown distribution $p(x)$, and suppose that we have modelled this using an approximating distribution $q(x)$. The Kullback-Leibler divergence (KL divergence), or known as the relative entropy, between the distributions $p(x)$ and $q(x)$, is calculated as [12]*

$$\mathrm{KL}(p||q) = -\int p(x)\log\frac{q(x)}{p(x)}\mathrm{d}x = \int p(x)\log\frac{p(x)}{q(x)}\mathrm{d}x \tag{2.11}$$

Lemma 2.3.2 (KL divergence on two univariate normal distributions p, q)**.** *Assume that $p(x) = \mathcal{N}(x;\mu_p,\sigma_p^2)$ and $q(x) = \mathcal{N}(x;\mu_q,\sigma_q^2)$ as two normal probability density functions, then their KL*

divergence defined in (2.11) can be written as [13]

$$\mathrm{KL}(p||q) = \frac{1}{2}\left(\log\frac{\sigma_q^2}{\sigma_p^2} + \frac{\sigma_p^2}{\sigma_q^2} + \frac{(\mu_p-\mu_q)^2}{\sigma_q^2} - 1\right) = \log\frac{\sigma_q}{\sigma_p} + \frac{\sigma_p^2+(\mu_p-\mu_q)^2}{2\sigma_q^2} - \frac{1}{2} \tag{2.12}$$

Definition 2.3.3 (2-Wasserstein distance, on two normal distributions)**.** *For two non-degenerate, d-dimensional normal distributions $\mu = \mathcal{N}(m_0, \Sigma_0)$ and $\nu = \mathcal{N}(m_1, \Sigma_1)$ (so Σ_0, Σ_1 are positive definite matrices), their 2-Wasserstein distance is defined as [14]*

$$W_2^2(\mu,\nu) = \|m_1 - m_0\|^2 + \mathrm{tr}\left(\Sigma_0 + \Sigma_1 - 2(\Sigma_0^{\frac{1}{2}}\Sigma_1\Sigma_0^{\frac{1}{2}})^{\frac{1}{2}}\right) \tag{2.13}$$

Lemma 2.3.4 (2-Wasserstein distance on two univariate normal distributions)**.** *More specifically, if we treat univariate normal distributions $p(x) = \mathcal{N}(x; \mu_p, \sigma_p^2)$ and $q(x) = \mathcal{N}(x; \mu_q, \sigma_q^2)$, then the 2-Wasserstein distance becomes*

$$W_2^2(p,q) = (\mu_q - \mu_p)^2 + (\sigma_p - \sigma_q)^2 \tag{2.14}$$

which implies the symmetric behaviour of 2-Wasserstein metric for the univariate normal case.

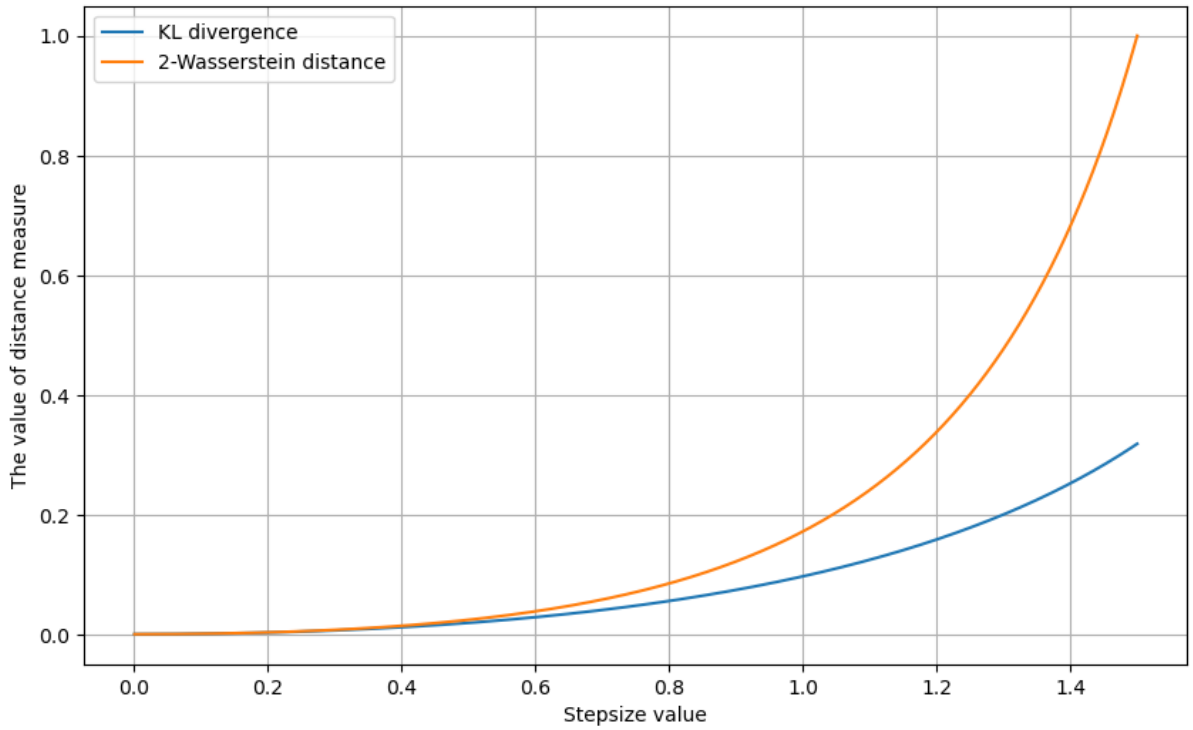


Figure 2.1 Graph for plotting the KL divergence (blue) and 2-Wasserstein distance (orange) on the special target $p_\star = \mathcal{N}(\mu, 1)$ (i.e. $a = 1$) and stationary distribution generated by ULA q as in (2.10).

2.3.2 ULA performance: stationary variance control

We are now able to calculate the KL divergence and 2-Wasserstein distance between the target distribution (2.5) and the stationary distribution generated from the ULA (2.10) as the approximation. With the notation of $p_\star(x) = \mathcal{N}\left(x; -\frac{b}{a}, \frac{1}{a}\right)$ as the unknown target distribution, and $q(x) = \mathcal{N}\left(x; -\frac{b}{a}, \frac{2}{a(2-a\gamma)}\right)$ as the stationary distribution generated by ULA as in (2.10), we would have

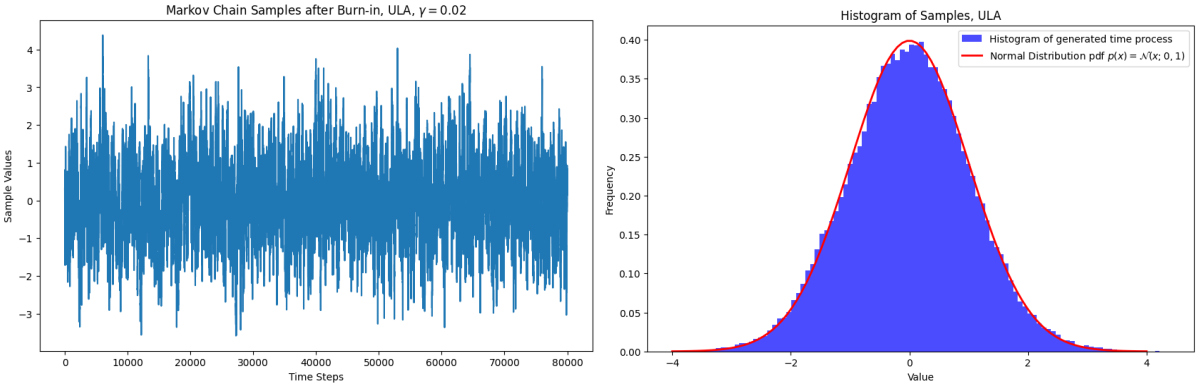
$$\mathrm{KL}(p_\star||q) = \frac{1}{2}\log\left(\frac{2}{a(2-a\gamma)}\right) - \frac{1}{2}\log\frac{1}{a} - \frac{a\gamma}{4} \tag{2.15}$$

$$W_2^2(p_\star, q) = \left(\frac{1}{a} - \sqrt{\frac{2}{a(2-a\gamma)}}\right)^2 \tag{2.16}$$

Notice that both metrics are independent of the mean, and we are able to plot these two measures with $p_\star(x) = \mathcal{N}(x; \mu, 1)$, see Figure 2.1.

Quantitatively, the larger the value of distance measure, the stationary distribution would act more different compared with the target distribution, leading a worse performance of ULA for this stepsize. Therefore, we would like to set a stepsize so that the distance measure is close to 0.

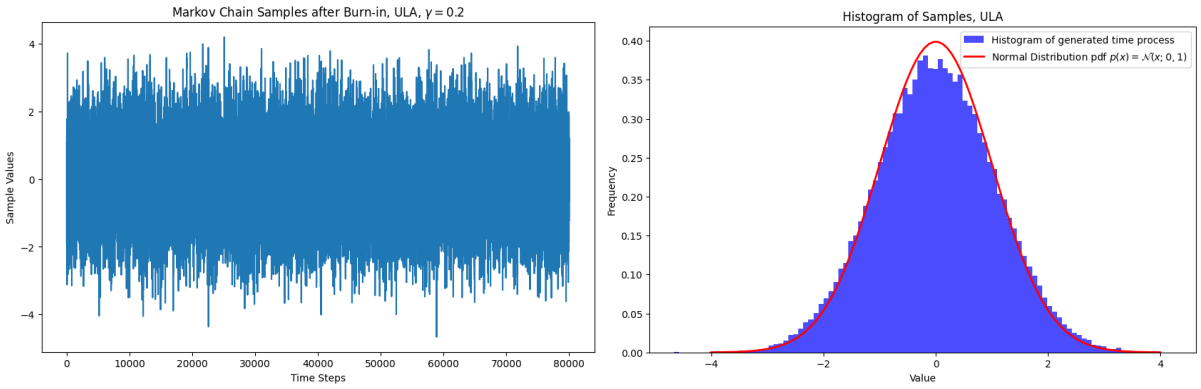
Based on the setting that the target distribution is $p_\star(x) = \mathcal{N}(x; 0, 1)$, we would first try a small stepsize $\gamma = 0.02$ to test its performance. This would be illustrated in Figure 2.2.



(a) Markov chain samples for ULA, $\gamma = 0.02$ **(b)** Histogram and density plot, $\gamma = 0.02$

Figure 2.2 The above figures shows using ULA to sample the target distribution $p_\star(x) = \mathcal{N}(x; 0, 1)$ with an appropriate stepsize $\gamma = 0.02$. This stepsize helps to hold the stationarity condition as in (2.7), so the chain converge to the stationary distribution as in (2.10). With this stepsize, the KL divergence and 2-Wasserstein distance are with the values of 2.517×10^{-5} and 2.538×10^{-5} respectively, which are negligible values. Note that the the histogram of the chain (Figure 2.2b) roughly matches the true density function, showing a great similarity between the target distribution (red curve) and the stationary distribution achieved by ULA.

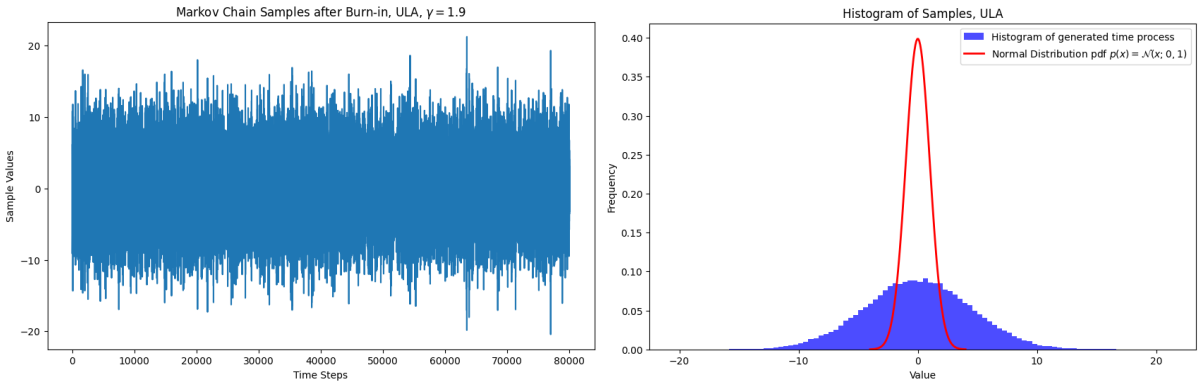
Dissimilarity increases when the stepsize rises, so the performance of ULA becomes worse. However, once the KL divergence and 2-Wasserstein distance are not too 'big', we may still perform ULA for sampling, *although such 'bigness' of measures really depends on the actual problem.*



(a) Markov chain samples for ULA, $\gamma = 0.2$ **(b)** Histogram and density plot, $\gamma = 0.2$

Figure 2.3 The above figures shows using ULA to sample the target distribution $p_\star(x) = \mathcal{N}(x; 0, 1)$ with a stepsize $\gamma = 0.2$, so the stationarity condition in (2.7) holds, and the chain would still converge to the stationary distribution (2.10). With this stepsize, the KL divergence and 2-Wasserstein distance are with the values of 2.680×10^{-3} and 2.926×10^{-3} respectively. Although they are still small, we notice that the sampling performance is worse than the previous case $\gamma = 0.02$, as the histogram in (2.3b) distorts the actual target density more compared with histogram in (2.2b).

A too large stepsize can drastically impair ULA performance, with a great dissimilarity between the target $p_\star$ and stationary distributions q, and a large variance for q from (2.10).



(a) Markov chain samples for ULA, $\gamma = 1.9$ **(b)** Histogram and density plot, $\gamma = 1.9$

Figure 2.4 The above figures shows using ULA to sample the target distribution $p_\star(x) = \mathcal{N}(x; 0, 1)$ with a 'too large' stepsize $\gamma = 1.9$. Although the stationarity condition (2.7) still holds with this stepsize, the values of KL divergence and 2-Wasserstein distance for this stationary distribution is 1.023 and 12.06 respectively. Values for states in the Markov chain for (2.4a) are too extreme (reaches ± 20) that almost cannot be samples from $p_\star(x)$, whereas through the histogram (2.4b), such extreme values occur many times in such Markov chain generated by ULA, therefore indicates the stationary distribution with this big stepsize would have too large variance so that the density of 'extreme values' are non-negligible.

In conclusion, in order to let ULA performs better in sampling problem, we would set the stepsize γ that has small values of KL divergence and 2-Wasserstein distance, so the variance of stationary distribution can be controlled.

2.3.3 ULA performance: rate of convergence

In the last section we investigate the performance of ULA by measuring the 'difference' between the stationary distribution and the target distribution $p_\star$ with varying stepsize γ. It seems that the smaller stepsize, the two distributions resemble better. However, a too small stepsize would also lower the performance of ULA, since it would result a slow rate of convergence for the chain generated by ULA to reach the stationary distribution.

In other words, what we've discussed in the previous section is the similarity between the target distribution and the stationary distribution of the chain by ULA, but we should first reach, or converge to, the stationary distribution. If the chain with a certain stepsize has a stationary distribution that resembles the target distribution very well, but it converges slowly, then our samples taken may not follow the target distribution as well, unless a huge amount of samples are taken, which significantly increases our cost.

In this section, we would discuss the rate of convergence of ULA with a range of stepsizes. This means we focus on the distribution at each time and measure the KL divergence and 2-Wasserstein distance to the target distribution $p_\star$ in the form of (2.5), with certain stepsize γ. To start with, recall the chain generated by ULA is in the form of (2.6), with the substitution $m = 1 - a\gamma$ and $n = -\gamma b$, we would have

$$\begin{aligned} X_t = mX_{t-1} + n + v_t &= m(mX_{t-2} + n + v_{t-1}) + n + v_t \\ &= m^2 X_{t-2} + n(1+m) + (v_t + mv_{t-1}) \\ &= \cdots \\ &= m^t x_0 + n(1 + m + \cdots + m^{t-1}) + (v_t + mv_{t-1} + \cdots + m^{t-1}v_1) \\ &\sim \mathcal{N}\left(m^t x_0 + n\sum_{i=0}^{t-1} m^i, 2\gamma \sum_{i=0}^{t-1} m^{2i} \right) \end{aligned}$$

where we initialise our ULA by setting $X_0 = x_0$, and $v_t \sim \mathcal{N}(0, 2\gamma)$ as white noises, being independent and identically distributed.

Therefore, the distribution of the chain generated by ULA at time t is written as

$$X_t \sim \mathcal{N}\left((1-a\gamma)^t x_0 - \gamma b \sum_{i=0}^{t-1} (1-a\gamma)^i , 2\gamma \sum_{i=0}^{t-1} (1-a\gamma)^{2i} \right) \tag{2.17}$$

Followingly, if we denote this distribution as q_t, then the KL divergence between this distribution and the target distribution would be

$$\begin{aligned} \mathrm{KL}(p_\star || q_t) = \frac{1}{2} \log \left(2a\gamma \sum_{i=0}^{t-1} (1-a\gamma)^{2i} \right) + \frac{1}{2} \frac{1}{2a\gamma \sum_{i=0}^{t-1} (1-a\gamma)^{2i}} \\ + \frac{1}{2} \frac{\left[-\frac{b}{a} - \left((1-a\gamma)^t x_0 - \gamma b \sum_{i=0}^{t-1} (1-a\gamma)^i \right) \right]^2}{2\gamma \sum_{i=0}^{t-1} (1-a\gamma)^{2i}} - \frac{1}{2} \end{aligned} \tag{2.18}$$

and the 2-Wasserstein distance between this distribution and the target distribution is

$$W_2^2(p_\star, q_t) = \left(-\frac{b}{a} - \left[(1-a\gamma)^t x_0 - \gamma b \sum_{i=0}^{t-1} (1-a\gamma)^i \right] \right)^2 + \left(\sqrt{\frac{1}{a}} - \sqrt{2\gamma \sum_{i=0}^{t-1} (1-a\gamma)^{2i}} \right)^2 \tag{2.19}$$

Next, we may use the simpler 2-Wasserstein distance for our following discussion. Since both metrics measures the similarity of distributions, they would share the similar results as well. For illustration, we again use the example by setting the target distribution be $p_\star(x) = \mathcal{N}(x; 0, 1)$ with $a = 1, b = 0$, and we initialise the ULA chain by setting $x_0 = 1$.

We start with stepsize $\gamma = 0.02$. From last section we already know that this stepsize results a really good ULA performance on sampling. The decreasing trend of 2-Wasserstein distance could be seen in Figure 2.5.

Notice that, in Figure (2.5), at around $t = 400$, the distribution at time starts to converge to the stationary distribution, overlapping with the 2-Wasserstein distance of the stationary distribution with respect to the target distribution (which is 2.538×10^{-5} as deduced in Figure 2.2). This means when sampling by ULA with $\gamma = 0.02$, after $t = 400$ the distribution (2.17) starts to behave as the stationary distribution (2.10).

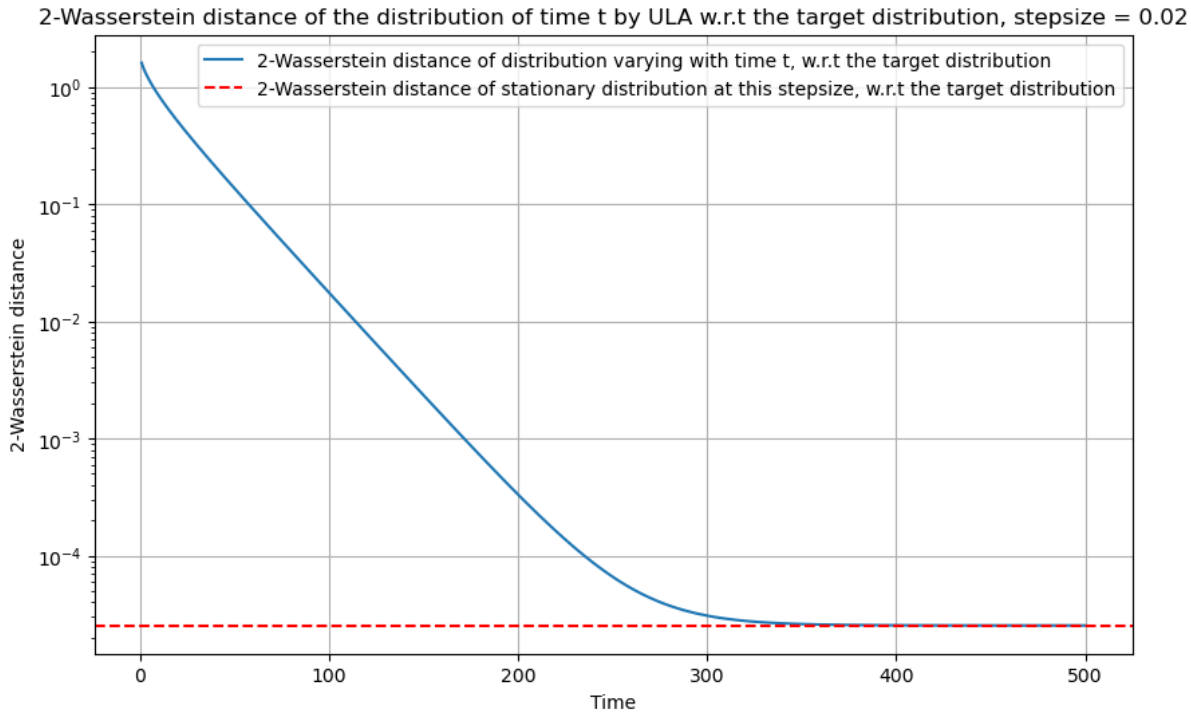


Figure 2.5 2-Wasserstein distance of the distribution (2.17) with respect to the target $p_\star$, with $\gamma = 0.02$.

We then increase our stepsize to $\gamma = 0.2$ to see the rate of convergence to the stationary distribution. This would be shown in Figure 2.6.

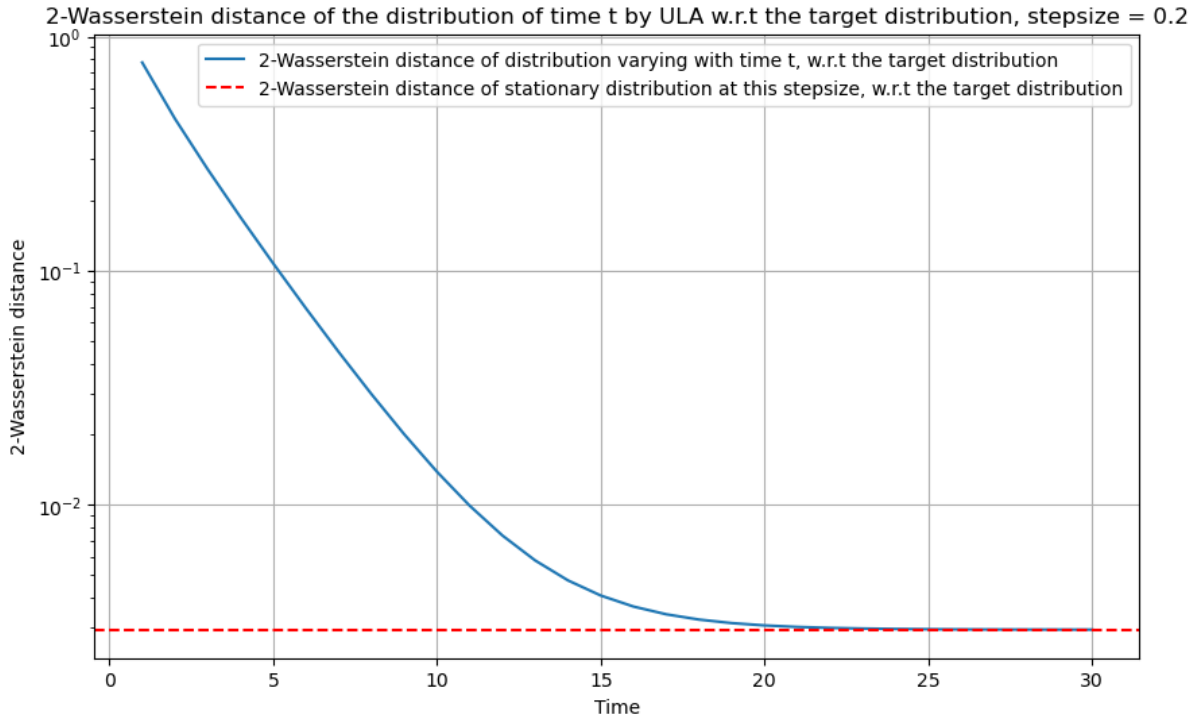


Figure 2.6 2-Wasserstein distance of the distribution (2.17) with respect to the target $p_\star$, with $\gamma = 0.2$.

From the above figure, obviously the rate of convergence increases when the stepsize is 0.2 - the 2-Wasserstein distance of distribution at each time overlaps with the stationary distribution (2.926×10^{-3}, as deduced in Figure 2.3) at around $t = 25$. As the sample cost increases when we need to take more samples to ensure the distribution at each time converges to the stationary distribution, ULA with a larger stepsize $\gamma = 0.2$ implies that we can have a lower sampling cost than ULA with $\gamma = 0.02$.

Finally, we work with a very small stepsize $\gamma = 0.001$, and the result is shown in Figure 2.7. The rate of convergence is significantly slower for $\gamma = 0.001$, as after $t = 10000$ the distribution at each time starts to converge to the stationary distribution, with an overlap of 2-Wasserstein distances to 6.255×10^{-8}, which is the 2-Wasserstein distance of the stationary distribution to the target $p_\star$ with this stepsize. The sampling cost for this ULA would be remarkably higher than the previous two, as it results a slow convergence to the stationary distribution.

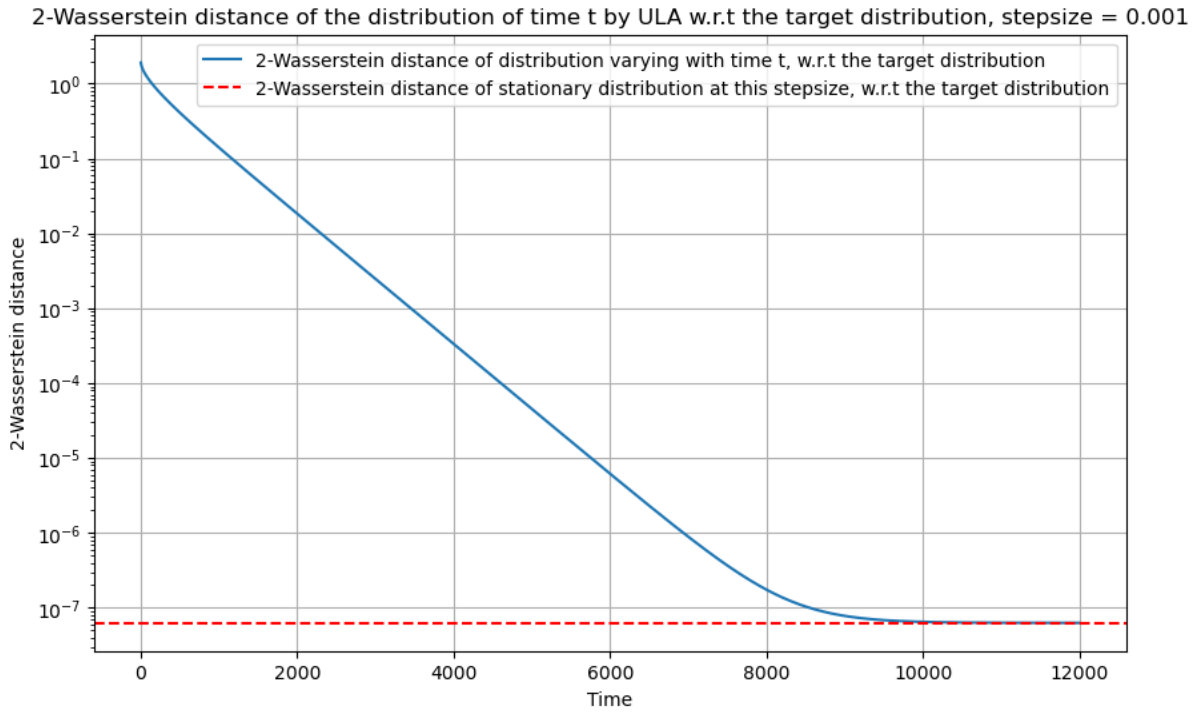
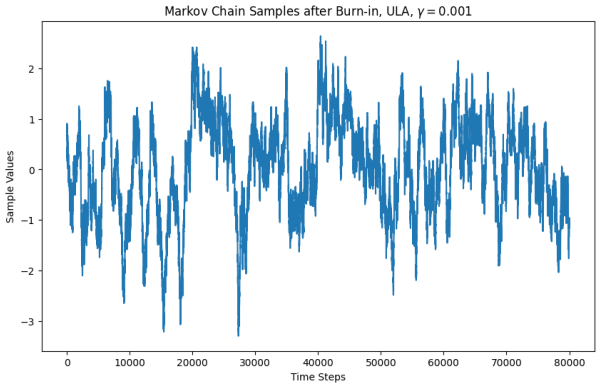
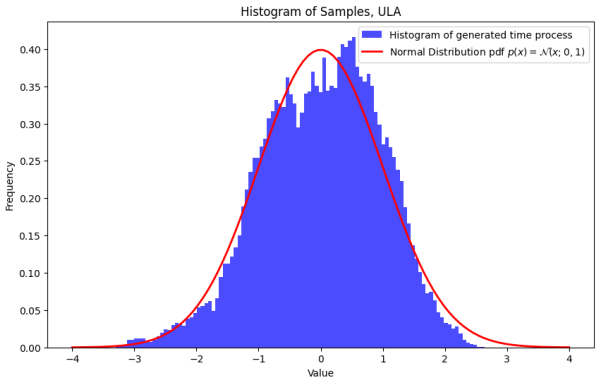


Figure 2.7 2-Wasserstein distance of the distribution (2.17) with respect to the target $p_\star$, with $\gamma = 0.001$.

We could show such slow convergence and high sampling cost by using Markov chain samples and histogram graphs again, which are illustrated in Figure 2.8 and 2.9.



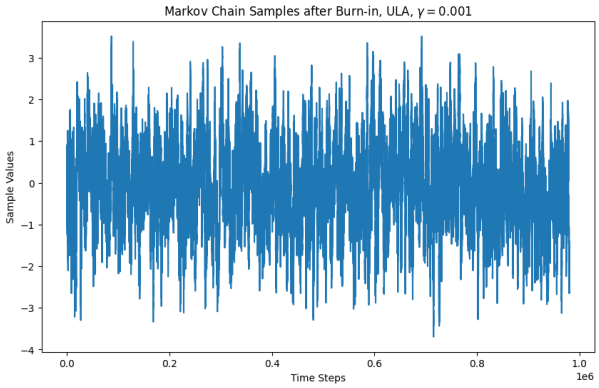
(a) Markov chain samples for ULA, $\gamma = 0.001$, with 10^5 samples, and burnin the first 20000 samples



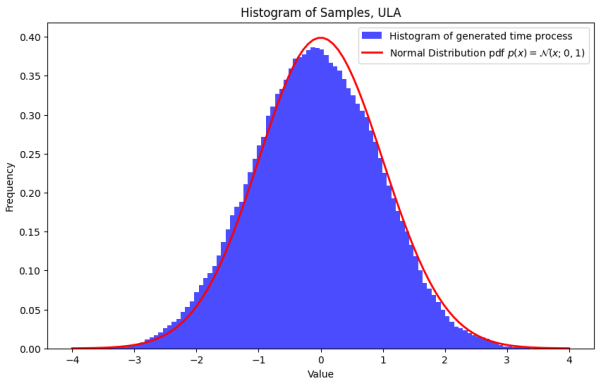
(b) Histogram and density plot, $\gamma = 0.001$, with 10^5 samples, and burnin the first 20000 samples

Figure 2.8 The above set of figures shows that the ULA with $\gamma = 0.001$ would perform badly for the sampling problem to the target $p_\star$. The chain shown in Figure (2.8a) implies the samples taken are highly correlated, and the histogram in Figure (2.8b) obviously does not match the target density well. As discussed, this is due to a slow convergence to the stationary distribution.

Once we increase our sample size, the sampling by this ULA would be better. However, this would also lead a significant rise of the sampling cost.



(a) Markov chain samples for ULA, $\gamma = 0.001$, with 10^6 samples, and burnin the first 20000 samples



(b) Histogram and density plot, $\gamma = 0.001$, with 10^6 samples, and burnin the first 20000 samples

Figure 2.9 Compared with Figure 2.8, when we increase our sample size to 10^6, this ULA again starts to perform well, with a smaller 2-Wasserstein distance (around 6.255×10^{-8}) to the target $p_\star$ and a closer histogram to match the target density.

2.3.4 Summary

To summarise, under stationarity condition (2.7), we would have a 'trade-off' between the stationary variance control and the rate of convergence.

- If we use larger stepsize γ to reach faster convergence to the stationary distribution to reduce the sampling cost, the stationary distribution's variance would increase to ruin the

performance of ULA. This can be seen in Figure 2.2 and Figure 2.3.

- If we use smaller stepsize γ to control the stationary distribution's variance closer to the target distribution's variance, reduce the KL divergence and 2-Wasserstein distance to make the stationary distribution of ULA more similar to the target distribution, the rate of convergence from the distribution at each time to the stationary distribution would be slower, which means we need to take more samples, so the sampling cost would be higher. This idea can be seen by using Figure 2.2, Figure 2.8 and Figure 2.9.

Therefore, in order to increase the performance of ULA on sampling problem, but meanwhile do not have too high sampling cost, the stepsize γ, satisfying the stationarity condition (2.7), would be chosen carefully to consider both stationary variance control and rate of convergence. An appropriate stepsize γ would both have a smaller KL divergence or 2-Wasserstein distance between the stationary distribution and the target to make these two distributions similar, and a faster rate of convergence to reduce the sampling cost.

Chapter 3

Underdamped Langevin MCMC method

An alternative to methods based on the overdamped Langevin SDE (2.2) for sampling $p_\star$ in (2.1) is the class of algorithms based on the underdamped Langevin SDE [6]. The underdamped Langevin SDE is given as a system of two SDEs

$$\begin{cases} \mathrm{d}V_t = \underbrace{-\eta V_t \mathrm{d}t}_{\text{friction}} \underbrace{-h(X_t)\mathrm{d}t}_{\text{acceleration}} + \sqrt{\frac{2\eta}{\beta}}\mathrm{d}B_t \\ \mathrm{d}X_t = V_t \mathrm{d}t \end{cases} \tag{3.1}$$

the above system of equations (3.1) is also known as the kinetic Langevin diffusion, or the second-order Langevin process [8], where $\{X_t, V_t\}_{t\geq 0}$ are called position and momentum process respectively, as its name, $\{X_t\}$ is the desired Markov chain for sampling target density $p_\star(x)$. As before, $h = \nabla U$, and U needs to be a strong convex and L-smooth function.

Similar to the overdamped SDE (2.2), this underdamped Langevin algorithm can be used as both an MCMC sampler and non-convex optimiser [6], since under appropriate conditions (with appropriate stepsize γ chosen) as the ULA, the underdamped Langevin algorithm is also stationary with an invariant distribution measure π that

$$\pi(v, x)\mathrm{d}v\mathrm{d}x \propto \exp\left(-\beta\left(\frac{1}{2}\|v\|^2 + U(x)\right)\right)\mathrm{d}v\mathrm{d}x \tag{3.2}$$

with the convergent diffusion. [6]

3.1 Euler-Maruyama discretisation on underdamped Langevin SDE

If we can know the exact form of $h = \nabla U$, then we can again apply the Euler-Maruyama discretisation to solve the above underdamped Langevin SDE (3.1) as

$$\begin{cases} V_{t+1} = V_t - \gamma(\eta V_t + h(X_t)) + \sqrt{\frac{2\gamma\eta}{\beta}}\epsilon_{t+1} \\ X_{t+1} = X_t + \gamma V_t \end{cases} \tag{3.3}$$

as the underdamped Langevin MCMC method, where as before, γ is the stepsize, $\epsilon_t \sim \mathcal{N}(0, \mathbf{I}_d)$ for all $t \geq 0$ as the d-dimensional noise under normality assumption. However, in real world sampling problem, $h = \nabla U$ is expensive or impossible to compute exactly, but we can obtain an unbiased estimate of it efficiently. Therefore, usually the underdamped Langevin MCMC is solved with stochastic gradients, and such MCMC method is called as Stochastic Gradient Hamiltonian Monte Carlo (SGHMC), which is given as [6]

$$\begin{cases} V_{t+1}^{\gamma} = V_t^{\gamma} - \gamma[\eta V_t^{\gamma} + H(X_t^{\gamma}, \Theta_{t+1})] + \sqrt{\frac{2\gamma\eta}{\beta}}\epsilon_{t+1} \\ X_{t+1}^{\gamma} = X_t^{\gamma} + \gamma V_t^{\gamma} \end{cases} \tag{3.4}$$

where $\gamma > 0$ is the stepsize, $V_0^{\gamma} = v_0, X_0^{\gamma} = x_0$, and $\mathbb{E}[H(x, \Theta_0)] = h(x) = \nabla U(x)$ for every $x \in \mathbb{R}^d$.

In this report, however, we will assume that $h = \nabla U$ can be derived directly and focus on (3.3). Without loss of generality, we can set β to be 1. As before, we rewrite (3.3) equivalently as

$$\begin{cases} V_{t+1} = (1 - \gamma\eta)V_t - \gamma\nabla U(X_t) + \sqrt{2\gamma\eta}\epsilon_{t+1} \\ X_{t+1} = X_t + \gamma V_t \end{cases} \tag{3.5}$$

As mentioned in the previous section, with the assumption of one-dimensional sampling problem, a very common example that can ensure U is strongly convex and be used for analysis is to set $U(x)$ as a quadratic polynomial with a positive leading coefficient (if we deal with multidimensional sampling problem, then such quadratic polynomial setting would be in the same form as mentioned in Remark 2.1.1).

Meanwhile, with such U, we already know that the target density $p_\star$ would be in Gaussian distribution, as stated in (2.5). Therefore, we may substitute the target distribution's mean

with $-\frac{b}{a} = \mu_\star$, and the variance with $\frac{1}{a} = \sigma^2$, i.e. we rewrite our target distribution in (2.5) as $p_\star(x) = \mathcal{N}(x; \mu_\star, \sigma^2)$.

With this target density, we can know $\nabla U(x) = \frac{x-\mu_\star}{\sigma^2}$, then (3.5) is equivalent to

$$\begin{pmatrix} V_{t+1} \\ X_{t+1} \end{pmatrix} = \begin{pmatrix} 1-\gamma\eta & -\frac{\gamma}{\sigma^2} \\ \gamma & 1 \end{pmatrix} \begin{pmatrix} V_t \\ X_t \end{pmatrix} + \begin{pmatrix} \frac{\gamma\mu_\star}{\sigma^2} \\ 0 \end{pmatrix} + \begin{pmatrix} \sqrt{2\gamma\eta} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \epsilon_{t+1} \\ \zeta_{t+1} \end{pmatrix} \implies \mathbf{Z}_{t+1} = \mathbf{A}\mathbf{Z}_t + \mathbf{b} + \mathbf{C}\mathbf{W}_{t+1} \tag{3.6}$$

by letting

$$\mathbf{Z}_t = (V_t, X_t)^T, \quad \mathbf{A} = \begin{pmatrix} 1-\gamma\eta & -\frac{\gamma}{\sigma^2} \\ \gamma & 1 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} \frac{\gamma\mu_\star}{\sigma^2} \\ 0 \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} \sqrt{2\gamma\eta} & 0 \\ 0 & 0 \end{pmatrix} \tag{3.7}$$

and under the normality assumption, we would assume $\epsilon_t \sim \mathcal{N}(0,1), \zeta_t \sim \mathcal{N}(0,1)$, so then

$$\mathbf{W}_t = \begin{pmatrix} \epsilon_t \\ \zeta_t \end{pmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_2) \tag{3.8}$$

Therefore, for the underdamped Langevin system with $U(x)$ set to be a quadratic polynomial, it is equivalent to consider and sample the joint time process $\{Z_t\}_{t\geq 0}$. The algorithm would be similar to ULA and proposed as:

Algorithm 2: Underdamped Langevin sampler algorithm (underdamped LA, or ULD)

Input: N, the number of samples we want to take
$\mathbf{Z}_0 = \mathbf{z}_0$, as the initial momentum-position process vector
n, the number of burnin samples need to discard

Output: Approximately uncorrelated samples of the target distribution $p_\star$

for $t = 1, 2, \cdots, N$ **do**
  Take a random sample z' from the proposal

$$\mathbf{Z}' \sim q(\mathbf{z}'|\mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{z}'; \mathbf{A}\mathbf{z}_{t-1} + \mathbf{b}, \mathbf{C}\mathbf{C}^T)$$

  which is equivalently to generate z' by

$$\mathbf{z}' = \mathbf{A}\mathbf{z}_{t-1} + \mathbf{b} + \mathbf{C}\mathbf{w}_t$$

  for $\mathbf{W}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_2)$;
  Update the new state by $\mathbf{z}_t = \mathbf{z}'$.
end
return Discard first n burnin samples and return the remaining samples. The second entry of $\{\mathbf{Z}_t\}_{t\geq 0}$ is the required position process $\{X_t\}_{t\geq 0}$ that for sampling the target distribution $p_\star(x)$.

3.2 Stationary analysis

We now discuss the stationarity for this underdamped Langevin algorithm, i.e. we need to check whether or not the chain generated by underdamped Langevin SDE

$$\mathbf{Z}_t = \mathbf{A}\mathbf{Z}_{t-1} + \mathbf{b} + \mathbf{C}\mathbf{W}_t \tag{3.9}$$

is a stationary process, with the same definition of $\mathbf{A}, \mathbf{b}, \mathbf{C}, \mathbf{Z}_t, \mathbf{W}_t$ defined in (3.7) and (3.8).

3.2.1 Vector autoregressions

The p-th order vector autoregression, also known as vector autoregressive process with an order p, denoted VAR(p), is in the form of

$$\mathbf{y}_t = \mathbf{c} + \mathbf{\Phi}_1\mathbf{y}_{t-1} + \mathbf{\Phi}_2\mathbf{y}_{t-2} + \cdots + \mathbf{\Phi}_p\mathbf{y}_{t-p} + \boldsymbol{\epsilon}_t \tag{3.10}$$

where $\mathbf{c}$ denotes an $(n \times 1)$ vector of constants and $\mathbf{\Phi}_j$ an $(n \times n)$ matrix of autoregressive coefficients for $j = 1, 2, \cdots, p$. The $(n \times 1)$ vector $\boldsymbol{\epsilon}_t$ is a vector of white noise that satisfies $\mathbb{E}(\boldsymbol{\epsilon}_t) = 0$ and $\mathbb{E}(\boldsymbol{\epsilon_i}\boldsymbol{\epsilon_j^T}) = \delta_{ij}\mathbf{\Omega}$, for δ_{ij} representing the Kronecker delta, and $\mathbf{\Omega}$ represents a covariance matrix for the noise $\boldsymbol{\epsilon}_t$. [15]

Again, a centering process can be taken on the process (3.10) to result an equivalent process by removing $\boldsymbol{\mu} = (\mathbf{I}_n - \mathbf{\Phi}_1 - \cdots - \mathbf{\Phi}_p)^{-1}\mathbf{c}$ on both sides. Such $\boldsymbol{\mu}$ satisfies $-(\mathbf{\Phi}_1 + \cdots + \mathbf{\Phi}_p)\boldsymbol{\mu} = \mathbf{c} - \boldsymbol{\mu}$ so that

$$(\mathbf{y}_t - \boldsymbol{\mu}) = \mathbf{\Phi}_1(\mathbf{y}_{t-1} - \boldsymbol{\mu}) + \mathbf{\Phi}_2(\mathbf{y}_{t-2} - \boldsymbol{\mu}) + \cdots + \mathbf{\Phi}_p(\mathbf{y}_{t-p} - \boldsymbol{\mu}) + \boldsymbol{\epsilon}_t \tag{3.11}$$

Then, we could write this process (3.11) into its reduced form as a VAR(1) process. To do this, we let

$$\boldsymbol{\xi}_t = \begin{pmatrix} \mathbf{y}_t - \boldsymbol{\mu} \\ \mathbf{y}_{t-1} - \boldsymbol{\mu} \\ \vdots \\ \mathbf{y}_{t-p+1} - \boldsymbol{\mu} \end{pmatrix} \in \mathbb{R}^{np\times 1}, \mathbf{F} = \begin{pmatrix} \mathbf{\Phi}_1 & \mathbf{\Phi}_2 & \cdots & \mathbf{\Phi}_{p-1} & \mathbf{\Phi}_p \\ \mathbf{I}_n & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_n & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I}_n & \mathbf{0} \end{pmatrix} \in \mathbb{R}^{np\times np}, \mathbf{v}_t = \begin{pmatrix} \boldsymbol{\epsilon}_t \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix} \in \mathbb{R}^{np\times 1} \tag{3.12}$$

so that (3.11) can be rewritten as a centered VAR(1) process

$$\boldsymbol{\xi}_t = \mathbf{F}\boldsymbol{\xi}_{t-1} + \mathbf{v}_t \tag{3.13}$$

so that the conditions for stationarity of process (3.10) is same with the process (3.13) [15].

Proposition 3.2.1 (Eigenvalues of matrix $\mathbf{F}$)**.** *The eigenvalues of the matrix* $\mathbf{F}$ *defined in (3.12) satisfy*

$$|\mathbf{I}_n\lambda^p - \mathbf{\Phi}_1\lambda^{p-1} - \mathbf{\Phi}_2\lambda^{p-2} - \cdots - \mathbf{\Phi}_p| = 0 \tag{3.14}$$

Theorem 3.2.2 (Conditions for stationarity, VAR(p) process)**.** *A VAR(p) process defined as (3.10) is stationary as long as all its eigenvalues of* $\mathbf{F}$ *have modulus less than 1. i.e. if* λ *is a solution of (3.14), then* $|\lambda| < 1$*. Equivalently, we could also say such VAR(p) process is stationary if all values of* z *satisfying*

$$|\mathbf{I}_n - \mathbf{\Phi}_1 z - \mathbf{\Phi}_2 z^2 - \cdots - \mathbf{\Phi}_p z^p| = 0$$

lie outside the unit disk $|z| < 1$ *[15].*

We then go back to our time process (3.9) generated by underdamped Langevin SDE. Note that such process looks similar to the a vector autoregressive process VAR(1) once we treat $\mathbf{CW_t} = \boldsymbol{\gamma}_t$, and we may also 'center' this process by removing $(\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}$ on both sides (later we would show this is the mean of this process), so that the process (3.9) can be equivalently written as the centered VAR(1) process

$$\boldsymbol{\xi}_t = \mathbf{A}\boldsymbol{\xi}_{t-1} + \boldsymbol{\gamma}_t \tag{3.15}$$

where $\boldsymbol{\xi}_t = \mathbf{Z}_t - (\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}$. Meanwhile, by (3.8), we have

$$\mathbb{E}(\boldsymbol{\gamma}_t) = \mathbf{0}, \qquad \operatorname{cov}(\boldsymbol{\gamma}_i, \boldsymbol{\gamma}_j) = \mathbb{E}(\boldsymbol{\gamma}_i\boldsymbol{\gamma}_j^T) = \delta_{ij}\mathbf{C}\mathbf{C}^T \tag{3.16}$$

where δ_{ij} is the Kronecker delta.

Hence, by using Theorem 3.2.2, we claim that our process (3.9) is stationary if all roots for $|\mathbf{I}_2 - \mathbf{A}z| = 0$ lie outside the unit disk $|z| < 1$.

$$|\mathbf{I}_2 - \mathbf{A}z| = \begin{vmatrix} 1 - z(1-\gamma\eta) & \frac{\gamma}{\sigma^2}z \\ -\gamma z & 1 - z \end{vmatrix} = \left(1 - \gamma\eta + \frac{\gamma^2}{\sigma^2}\right) z^2 + (\gamma\eta - 2)z + 1 = 0$$

$$\Longrightarrow z = \frac{2 - \eta\gamma \pm \sqrt{(\gamma\eta - 2)^2 - 4\left(1 - \gamma\eta + \frac{\gamma^2}{\sigma^2}\right)}}{2\left(1 - \gamma\eta + \frac{\gamma^2}{\sigma^2}\right)}$$

hence our underdamped Langevin Markov chain (3.9) is stationary if we have appropriate η, γ, σ to satisfy

$$\left|\frac{2 - \eta\gamma + \sqrt{(\gamma\eta - 2)^2 - 4\left(1 - \gamma\eta + \frac{\gamma^2}{\sigma^2}\right)}}{2\left(1 - \gamma\eta + \frac{\gamma^2}{\sigma^2}\right)}\right| > 1, \quad \left|\frac{2 - \eta\gamma - \sqrt{(\gamma\eta - 2)^2 - 4\left(1 - \gamma\eta + \frac{\gamma^2}{\sigma^2}\right)}}{2\left(1 - \gamma\eta + \frac{\gamma^2}{\sigma^2}\right)}\right| > 1 \tag{3.17}$$

3.2.2 Mean of $\{\mathbf{Z}_t\}_{t\geq 0}$

After showing the process is stationary, we then investigate its mean. Assume

$$\boldsymbol{\mu} = \mathbb{E}(\mathbf{Z}_t) = \begin{pmatrix} \mathbb{E}(V_t) \\ \mathbb{E}(X_t) \end{pmatrix}$$

since the process (3.9) is stationary, we can take expectations of both sides [15] and derive this expectation by

$$\mathbb{E}(\mathbf{Z}_t) = \mathbf{A}\mathbb{E}(\mathbf{Z}_{t-1}) + \mathbf{b} \implies \boldsymbol{\mu} = \mathbf{A}\boldsymbol{\mu} + \mathbf{b} \implies \boldsymbol{\mu} = (\mathbf{I}_2 - \mathbf{A})^{-1}\mathbf{b} \tag{3.18}$$

note that $\mathbf{I}_2 - \mathbf{A}$ is always invertible for this case, since for stepsize $\gamma > 0$, we always have

$$|\mathbf{I}_2 - \mathbf{A}| = \left|\begin{pmatrix} \gamma\eta & \frac{\gamma}{\sigma^2} \\ -\gamma & 0 \end{pmatrix}\right| = \frac{\gamma^2}{\sigma^2} > 0$$

Therefore, we could explicitly compute the mean of the process

$$\boldsymbol{\mu} = (\mathbf{I}_2 - \mathbf{A})^{-1}\mathbf{b} = \frac{\sigma^2}{\gamma^2}\begin{pmatrix} 0 & -\frac{\gamma}{\sigma^2} \\ \gamma & \gamma\eta \end{pmatrix}\begin{pmatrix} \frac{\gamma\mu_\star}{\sigma^2} \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \mu_\star \end{pmatrix} = \begin{pmatrix} \mathbb{E}(V_t) \\ \mathbb{E}(X_t) \end{pmatrix} \tag{3.19}$$

3.2.3 Autocovariance sequence (acv) of $\{\mathbf{Z}_t\}_{t\geq 0}$

We then consider the autocovariance sequence (acv): $s_\tau = \mathrm{cov}(\mathbf{Z}_t, \mathbf{Z}_{t+\tau})$ for this process. As mentioned, we know that our target process (3.9) from underdamped Langevin SDE is equivalent to the centered VAR(1) process (3.15). Since $\boldsymbol{\xi}_t = \mathbf{Z}_t - (\mathbf{I}-\mathbf{A})^{-1}\mathbf{b} = \mathbf{Z}_t - \boldsymbol{\mu}$ and $\boldsymbol{\mu}$ is a constant vector, we shall directly have

$$\mathrm{cov}(\boldsymbol{\xi}_t, \boldsymbol{\xi}_{t+\tau}) = \mathrm{cov}(\mathbf{Z}_t, \mathbf{Z}_{t+\tau}) = s_\tau$$

Now by (3.15), we can easily expand $\boldsymbol{\xi}_{t+\tau}$ recursively as

$$\begin{aligned}
\boldsymbol{\xi}_{t+\tau} = \mathbf{A}\boldsymbol{\xi}_{t+\tau-1} + \boldsymbol{\gamma}_{t+\tau} &= \mathbf{A}(\mathbf{A}\boldsymbol{\xi}_{t+\tau-2} + \boldsymbol{\gamma}_{t+\tau-1}) + \boldsymbol{\gamma}_{t+\tau} \\
&= \mathbf{A}^2\boldsymbol{\xi}_{t+\tau-2} + (\mathbf{A}\boldsymbol{\gamma}_{t+\tau-1} + \boldsymbol{\gamma}_t) \\
&= \cdots \\
&= \mathbf{A}^\tau \boldsymbol{\xi}_t + \sum_{k=0}^{\tau-1} \mathbf{A}^k \boldsymbol{\gamma}_{t+\tau-k}
\end{aligned}$$

Therefore, the acv can be derived as

$$\begin{aligned}
s_\tau = \mathrm{cov}(\boldsymbol{\xi}_t, \boldsymbol{\xi}_{t+\tau}) &= \mathrm{cov}\left(\boldsymbol{\xi}_t, \mathbf{A}^\tau \boldsymbol{\xi}_t + \sum_{k=0}^{\tau-1} \mathbf{A}^k \boldsymbol{\gamma}_{t+\tau-k}\right) \\
&= \mathrm{cov}(\boldsymbol{\xi}_t, \boldsymbol{\xi}_t)(\mathbf{A}^\tau)^T + \mathrm{cov}\left(\boldsymbol{\xi}_t, \sum_{k=0}^{\tau-1} \mathbf{A}^k \boldsymbol{\gamma}_{t+\tau-k}\right)
\end{aligned}$$

Now, since $\boldsymbol{\xi}_t = \mathbf{A}\boldsymbol{\xi}_{t-1} + \boldsymbol{\gamma}_t$, by using the idea of general linear process and recursion we can know that $\mathbf{Z}_t$ can be represented as a series of all its past and present errors $\boldsymbol{\gamma}_t, \boldsymbol{\gamma}_{t-1}, \cdots$, i.e. a (vector) moving average MA(∞) process as

$$\boldsymbol{\xi}_t = \sum_{k=0}^{\infty} \mathbf{A}^k \boldsymbol{\gamma}_{t-k} \tag{3.20}$$

As mentioned in (3.16), together with the linearity of covariance, we would have

$$\mathrm{cov}\left(\boldsymbol{\xi}_t, \sum_{k=0}^{\tau-1} \mathbf{A}^k \boldsymbol{\gamma}_{t+\tau-k}\right) = \mathbf{0} \tag{3.21}$$

Therefore, finally s_τ can be simplified as,

$$s_\tau = \mathrm{cov}(\boldsymbol{\xi}_t, \boldsymbol{\xi}_t)(\mathbf{A}^\tau)^T = s_0(\mathbf{A}^\tau)^T \tag{3.22}$$

and by the same deduction, we also have

$$s_{-\tau} = \mathbf{A}^\tau s_0 \tag{3.23}$$

3.2.4 Variance of $\{\mathbf{Z}_t\}_{t\geq 0}$

We finally consider the variance of this time process, which is simply $s_0 = \mathrm{cov}(\mathbf{Z}_t, \mathbf{Z}_t) = \mathrm{cov}(\boldsymbol{\xi}_t, \boldsymbol{\xi}_t) = \mathbb{E}(\boldsymbol{\xi}_t\boldsymbol{\xi}_t^T)$, and for $\boldsymbol{\xi}_t\boldsymbol{\xi}_t^T$, we have

$$\begin{aligned}\boldsymbol{\xi}_t\boldsymbol{\xi}_t^T &= (\mathbf{A}\boldsymbol{\xi}_{t-1} + \boldsymbol{\gamma}_t)(\mathbf{A}\boldsymbol{\xi}_{t-1} + \boldsymbol{\gamma}_t)^T \\ &= \mathbf{A}\boldsymbol{\xi}_{t-1}\boldsymbol{\xi}_{t-1}^T\mathbf{A}^T + \mathbf{A}\boldsymbol{\xi}_{t-1}\boldsymbol{\gamma}_t^T + \boldsymbol{\gamma}_t\boldsymbol{\xi}_{t-1}^T\mathbf{A}^T + \boldsymbol{\gamma}_t\boldsymbol{\gamma}_t^T\end{aligned}$$

With the help of (3.20) and (3.16), we would have $\mathbb{E}(\boldsymbol{\xi}_{t-1}\boldsymbol{\gamma}_t^T) = \mathbb{E}(\boldsymbol{\gamma}_t\boldsymbol{\xi}_{t-1}^T) = \mathbf{0}$, hence by linearity of expectation,

$$\mathbb{E}(\boldsymbol{\xi}_t\boldsymbol{\xi}_t^T) = \mathbf{A}\mathbb{E}(\boldsymbol{\xi}_{t-1}\boldsymbol{\xi}_{t-1}^T)\mathbf{A}^T + \mathbb{E}(\boldsymbol{\gamma}_t\boldsymbol{\gamma}_t^T) \implies s_0 = \mathbf{A}s_0\mathbf{A}^T + \mathbf{C}\mathbf{C}^T \tag{3.24}$$

which is indeed the Lyapunov equation for the variance of time process $\mathbf{Z}_t$. Indeed, we would get the same result if we directly apply variance operator on both sides of (3.15). In order to solve s_0 explicitly, the **vec** operator can be used to obtain a closed-form solution to (3.24).

Definition 3.2.3 (**vec** operator)**.** *If* $\mathbf{A}$ *is an* $(m \times n)$ *matrix, then* ***vec***$(\mathbf{A})$ *is an* $(mn \times 1)$ *column vector obtained by stacking the columns of* $\mathbf{A}$*, one below the other, with the columns ordered from left to right. For example, if*

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix} \in \mathbb{R}^{3\times 2}$$

then $\mathbf{vec}(\mathbf{A}) = (a_{11}, a_{21}, a_{31}, a_{12}, a_{22}, a_{32})^T \in \mathbb{R}^6$*. [15]*

From its definition, it is obvious that we have the linearity property for **vec** operator, that is

$$\mathbf{vec}(\mathbf{A} + \mathbf{B}) = \mathbf{vec}(\mathbf{A}) + \mathbf{vec}(\mathbf{B}) \tag{3.25}$$

provided matrices $\mathbf{A}, \mathbf{B}$ having the same size.

Definition 3.2.4 (Kronecker product)**.** *Let $\mathbf{A} \in \mathbb{R}^{K\times L}$ and $\mathbf{B} \in \mathbb{R}^{M\times N}$ be two matrices, then the Kronecker product between $\mathbf{A}$ and $\mathbf{B}$ is the $(KM \times LN)$ block matrix that*

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & \cdots & a_{1L}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{K1}\mathbf{B} & \cdots & a_{KL}\mathbf{B} \end{pmatrix}$$

where a_{ij} represents the ij-th entry of matrix $\mathbf{A}$ as a real scalar, which means that $a_{ij}\mathbf{B}$ is a matrix that its pq-th entry is $a_{ij}b_{pq}$. [16]

Theorem 3.2.5 (Eigenvalues and eigenvectors of Kronecker product)**.** *Let $\mathbf{A} \in \mathbb{R}^{m\times m}$ and $\mathbf{B} \in \mathbb{R}^{n\times n}$ and we denote $\sigma(\cdot)$ as the set of eigenvalues of matrix $\cdot$ (note that we could have complex eigenvalues). Let $\lambda \in \sigma(\mathbf{A})$ being as an eigenvalue of $\mathbf{A}$ with corresponding eigenvector $\mathbf{x}$, $\mu \in \sigma(\mathbf{B})$ being as an eigenvalue of $\mathbf{B}$ with corresponding eigenvector $\mathbf{y}$, then $\lambda\mu$ is an eigenvalue of $\mathbf{A} \otimes \mathbf{B}$ with corresponding eigenvector $\mathbf{x} \otimes \mathbf{y}$. [17]*

Proposition 3.2.6. *Let $\mathbf{A}, \mathbf{B}, \mathbf{C}$ be matrices whose dimensions are such that the product $\mathbf{ABC}$ exists. Then*

$$\mathbf{vec}(\mathbf{ABC}) = (\mathbf{C}^T \otimes \mathbf{A})\mathbf{vec}(\mathbf{B})$$

where the symbol $\otimes$ denotes the Kronecker product. [15]

Recall that $s_0 = \mathrm{cov}(\mathbf{Z}_t, \mathbf{Z}_t)$ is a covariance matrix, hence if we apply the **vec** operator to both sides of (3.24), by using linearity of **vec** operator in (3.25) and proposition 3.2.6, the result is

$$\mathbf{vec}(s_0) = \mathbf{vec}(\mathbf{A}s_0\mathbf{A}^T + \mathbf{CC^T}) = (\mathbf{A} \otimes \mathbf{A})\mathbf{vec}(s_0) + \mathbf{vec}(\mathbf{CC}^T) = \boldsymbol{\mathcal{A}}\mathbf{vec}(s_0) + \mathbf{vec}(\mathbf{CC}^T) \tag{3.26}$$

where $\boldsymbol{\mathcal{A}} = \mathbf{A} \otimes \mathbf{A}$, and in this example, we have $\mathbf{A} \in \mathbb{R}^{2\times 2}$, then $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{4\times 4}$, and then equation (3.26) has the solution

$$(\mathbf{I}_4 - \boldsymbol{\mathcal{A}})\mathbf{vec}(s_0) = \mathbf{vec}(\mathbf{CC}^T) \implies \mathbf{vec}(s_0) = (\mathbf{I}_4 - \boldsymbol{\mathcal{A}})^{-1}\mathbf{vec}(\mathbf{CC}^T) \tag{3.27}$$

provided that $\mathbf{I}_4 - \boldsymbol{\mathcal{A}}$ is non-singular.

Lemma 3.2.7. *If all eigenvalues of* $\mathbf{A}$ *satisfy* $|\lambda| < 1$*, then the matrix* $\mathbf{I} - \mathbf{A}$ *is non-singular.*

Proof. First, note that if λ_i is an eigenvalue of $\mathbf{A}$ with a corresponding eigenvector $\mathbf{x}_i : \mathbf{A}\mathbf{x}_i = \lambda_i \mathbf{x}_i$, then for any number t, we have

$$(t\mathbf{I} - \mathbf{A})\mathbf{x}_i = (t - \lambda_i)\mathbf{x}_i$$

meaning that $t - \lambda_i$ is an eigenvalue of $t\mathbf{I} - \mathbf{A}$.

We assume that $\mathbf{A}$ has total n eigenvalues. Then, write the characteristic polynomial of $\mathbf{A}$

$$c_{\mathbf{A}}(t) = |t\mathbf{I} - \mathbf{A}| = \prod_{i=1}^{n}(t - \lambda_i)$$

since the product of eigenvalues of a matrix is the determinant of this matrix. We let $t = 1$ to obtain

$$|\mathbf{I} - \mathbf{A}| = \prod_{i=1}^{n}(1 - \lambda_i)$$

Finally, for all real eigenvalues λ_i, we directly have $1 - \lambda_i > 0$; if $\lambda_i \in \mathbb{C}$ as a complex eigenvalue, then λ_i^* is also an eigenvalue. We use $\Re(z)$ and $\Im(z)$ to represent the real part and imaginary part of z respectively, and for complex λ_i that $|\lambda_i| < 1$, it is equivalent to say that $0 < |\Re(\lambda_i)| < 1, 0 < |\Im(\lambda_i)| < 1$. Then

$$(1 - \lambda_i)(1 - \lambda_i^*) = 1 - 2\Re(\lambda_i) + |\lambda_i|^2 = 1 - 2\Re(\lambda_i) + \Re(\lambda_i)^2 + \Im(\lambda_i)^2 = (1 - \Re(\lambda_i))^2 + \Im(\lambda_i)^2 > 0$$

Therefore, we can claim that $|\mathbf{I} - \mathbf{A}| = \prod_{i=1}^{n}(1 - \lambda_i) > 0 \iff \mathbf{I} - \mathbf{A}$ is non-singular. □

By Theorem 3.2.5, if we denote λ_i as the i-th eigenvalue for $\mathbf{A}$, then the eigenvalues for $\boldsymbol{\mathcal{A}}$ are all of the form $\lambda_i\lambda_j$. For a stationary distribution, by Theorem 3.2.2, all λ_i satisfy $|\lambda_i| < 1$, which implies that all eigenvalues of $\boldsymbol{\mathcal{A}}$ are inside the unit disk as well: $|\lambda_i\lambda_j| = |\lambda_i||\lambda_j| < 1$, hence by Lemma 3.2.7, $\mathbf{I}_4 - \boldsymbol{\mathcal{A}}$ is indeed non-singular.

Therefore, we can compute the variance s_0 by (3.27), and the result is

$$s_0 = \begin{pmatrix} -\frac{4\eta\sigma^4}{2\eta^2\gamma\sigma^4 - 3\eta\gamma^2\sigma^2 - 4\eta\sigma^4 + \gamma^3 + 4\gamma\sigma^2} & \frac{2\eta\gamma\sigma^4}{2\eta^2\gamma\sigma^4 - 3\eta\gamma^2\sigma^2 - 4\eta\sigma^4 + \gamma^3 + 4\gamma\sigma^2} \\ \frac{2\eta\gamma\sigma^4}{2\eta^2\gamma\sigma^4 - 3\eta\gamma^2\sigma^2 - 4\eta\sigma^4 + \gamma^3 + 4\gamma\sigma^2} & \frac{2\eta\sigma^4(\eta\gamma\sigma^2 - \gamma^2 - 2\sigma^2)}{2\eta^2\gamma\sigma^4 - 3\eta\gamma^2\sigma^2 - 4\eta\sigma^4 + \gamma^3 + 4\gamma\sigma^2} \end{pmatrix} \tag{3.28}$$

This would also finally indicate our stationary distribution is then

$$\mathbf{Z} = (\mathbf{V}, \mathbf{X}) \sim \mathcal{N}(\boldsymbol{\mu}, s_0) \tag{3.29}$$

with $\boldsymbol{\mu}$ defined in (3.19) and s_0 defined in (3.28).

In underdamped Langevin diffusion algorithm, we pick the second entry as our simulation for sampling. Therefore, it is necessary to consider the marginal distribution of stationary distribution $\mathbf{X}$ from the joint, bivariate stationary distribution (3.29).

Theorem 3.2.8 (Marginal distribution of multivariate normal distribution)**.** *Let x follows a multivariate normal distribution*

$$x \sim \mathcal{N}(\mu, \Sigma)$$

then the marginal distribution of any subset vector x_s is also a multivariate normal distribution

$$x_s \sim \mathcal{N}(\mu_s, \Sigma_s)$$

where μ_s drops the irrelevant variables (the ones not in the subset, i.e. marginalised out) from the mean vector μ, Σ_s drops the corresponding rows and columns from the covariance matrix Σ. [18]

By above theorem 3.2.8, we now have the marginal, stationary distribution $\mathbf{X}$ from (3.29)

$$\mathbf{X} \sim \mathcal{N}\left(\mu_\star, \frac{2\eta\sigma^4(\eta\gamma\sigma^2 - \gamma^2 - 2\sigma^2)}{2\eta^2\gamma\sigma^4 - 3\eta\gamma^2\sigma^2 - 4\eta\sigma^4 + \gamma^3 + 4\gamma\sigma^2}\right) \tag{3.30}$$

which is indeed the stationary marginal measure of the position chain from the Euler-Maruyama discretisation of underdamped Langevin SDE, which has not yet been discussed in the literature.

3.3 Performance of underdamped Langevin diffusion (ULD)

Similar with the evaluation of ULA performance in section 2.3, we may also be interested in the performance of underdamped Langevin MCMC, carried the same analysis as well.

Again, we are going to use KL divergence and 2-Wasserstein distance to measure the difference between the target distribution $p_\star(x) = \mathcal{N}(x; \mu, \sigma^2)$ and the marginal, stationary distribution q we've derived as in (3.30). By using the deduction in (2.12) for KL divergence and (2.14)

for 2-Wasserstein distance, if we set $\sigma_q^2 = \frac{2\eta\sigma^4(\eta\gamma\sigma^2-\gamma^2-2\sigma^2)}{2\eta^2\gamma\sigma^4-3\eta\gamma^2\sigma^2-4\eta\sigma^4+\gamma^3+4\gamma\sigma^2}$, then we would have

$$\mathrm{KL}(p_\star||q) = \log\left(\frac{\sigma_q}{\sigma}\right) + \frac{\sigma^2}{2\sigma_q^2} - \frac{1}{2} \tag{3.31}$$

$$W_2^2(p_\star, q) = (\sigma - \sigma_q)^2 \tag{3.32}$$

As before, both metrics are independent of the mean.

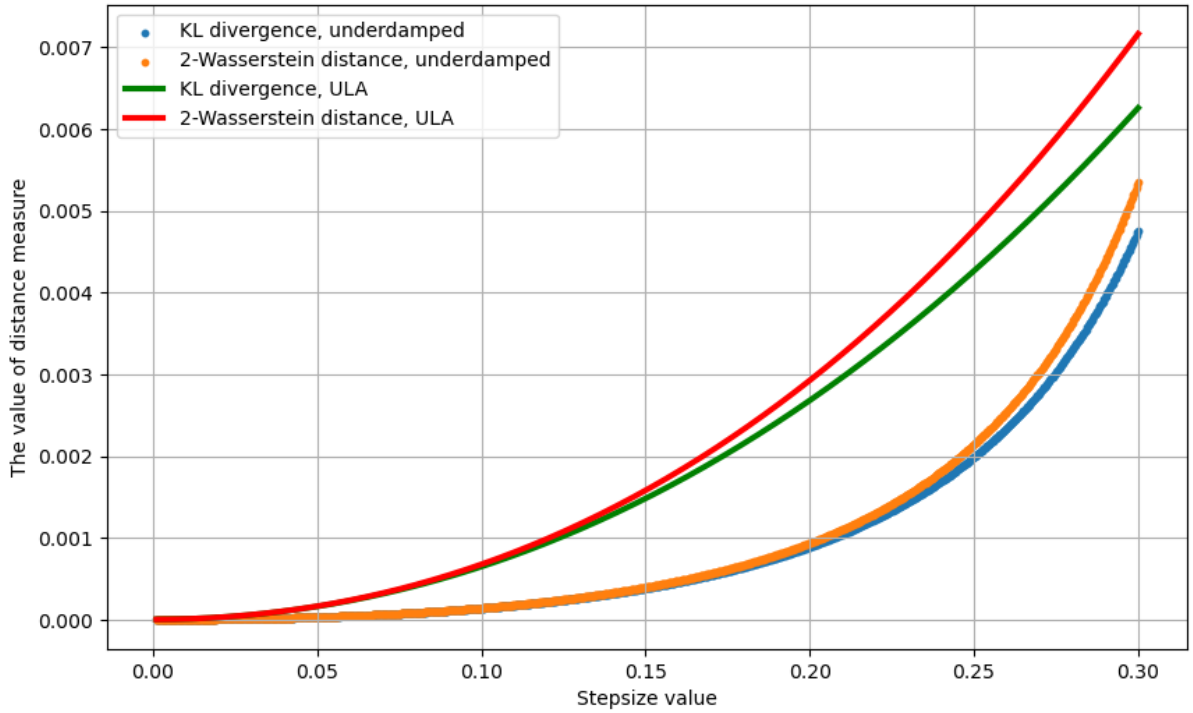
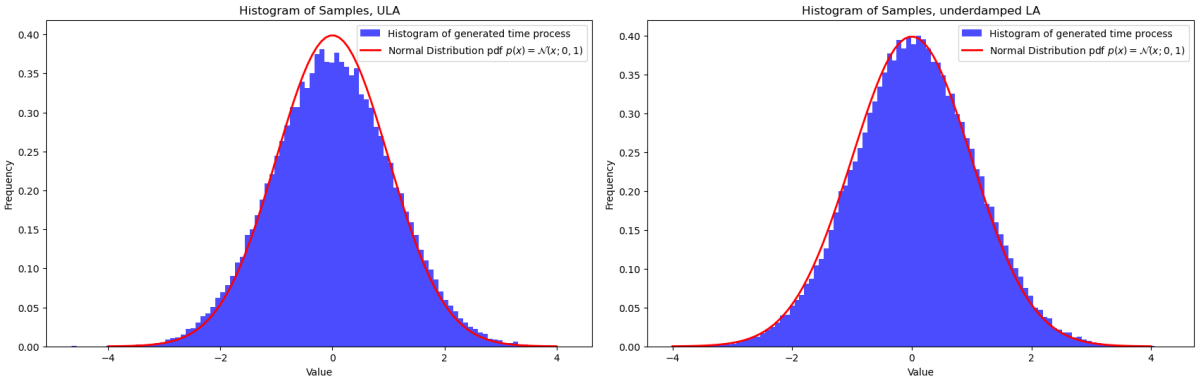


Figure 3.1 Graph for the KL divergence (blue) and 2-Wasserstein distance (orange) on the special target $p_\star = \mathcal{N}(\mu_\star, 1)$ and marginal distribution generated by underdamped Langevin algorithm q as in (3.30), with the friction coefficient $\eta = 5$, together with KL divergence (green) and 2-Wasserstein distance (red) on $p_\star$ and stationary distribution generated by ULA as in (2.10). Notice that, the blue and yellow 'curves' are scatter plots since not all stepsizes γ would satisfy the stationarity condition (3.17), and we would omit these stepsizes.

The above Figure 3.1 shows that at a same value of distance measure, the stepsize for underdamped Langevin algorithm would be larger than the corresponding stepsize for ULA. Meanwhile, for a same stepsize value, the distance measure for underdamped Langevin diffusion would be smaller than that by using ULA: at the same stepsize $\gamma = 0.2$, using underdamped Langevin diffusion would achieve a closer distribution to the target than ULA, since it would have a lower value of 2-Wasserstein distance. To illustrate this smaller dissimilarity, we may

apply the histograms from two different MCMC methods with $\gamma = 0.2$, see this in Figure 3.2.



(a) ULA at $\gamma = 0.2$ **(b)** ULD with $\eta = 5$ at $\gamma = 0.2$

Figure 3.2 The above set of histogram plots shows that the underdamped Langevin diffusion (ULD) would result a better sampling than the ULA with the same stepsize, for a closer match with respect to the true density curve.

It is also natural to think about the effect of friction coefficient (η) in the performance of the underdamped Langevin algorithm for the sampling. Previously we set $\eta = 5$, but what if we change this value of η? To investigate, we first fix our stepsize $\gamma = 0.2$, and apply the same step as above to plot the graph of KL divergence/2-Wasserstein distance against a range of friction coefficients η.

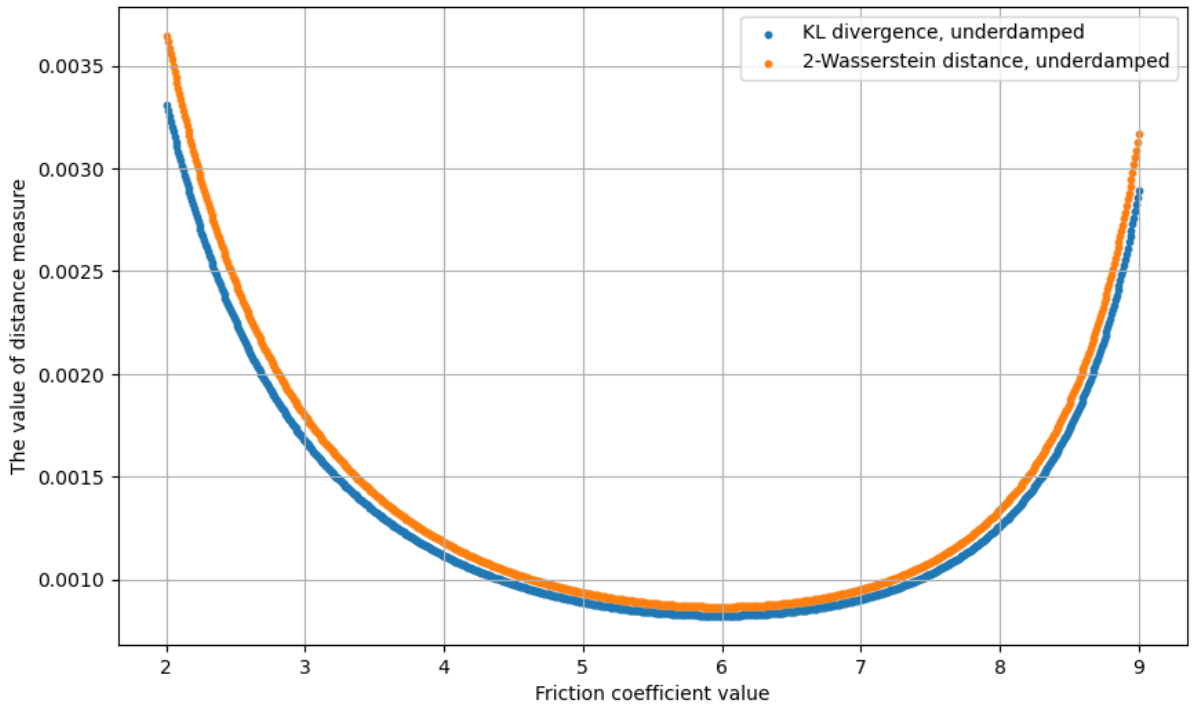
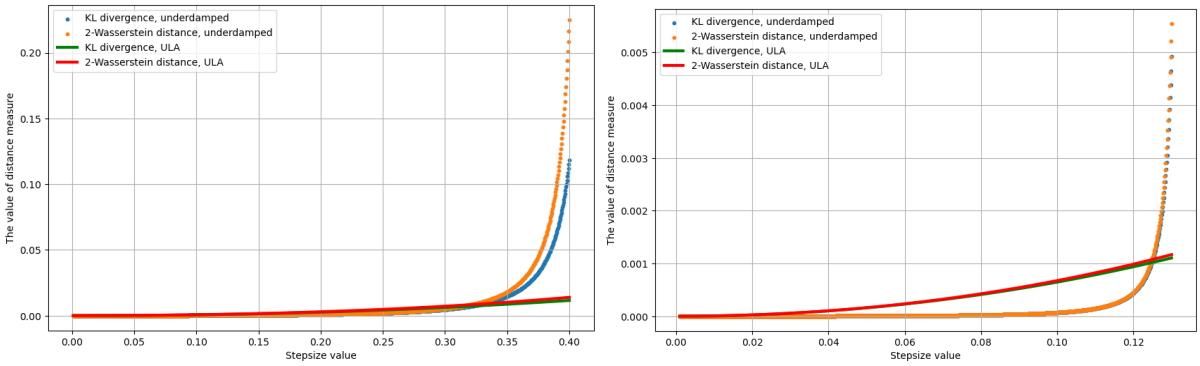


Figure 3.3 Graph for scatter plotting the KL divergence (blue) and 2-Wasserstein distance (orange) on the target $p_\star = \mathcal{N}(\mu_\star, 1)$ and stationary, marginal distribution generated by underdamped Langevin algorithm q as in (3.30), with fixed stepsize $\gamma = 0.2$.

From the above figure, it is obvious that the friction value η would affect our performance of sampling, when under a fixed stepsize $\gamma = 0.2$. Notice that the curve behaves like a U-shape, with the optimal value at around $\eta = 6$.

Remark 3.3.1. *If we take a larger value of stepsize, then the metrics for underdamped Langevin diffusion would diverge (so the metrics would rise again for greater dissimilarity), and it is much faster than that for ULA. Moreover, the larger the friction coefficient η taken, the smaller stepsize for metrics of underdamped Langevin diffusion causing divergence.*



(a) Quick divergence for $\gamma \in [0.001, 0.4]$, $\eta = 5$ **(b)** Quick divergence for $\gamma \in [0.001, 0.13]$, $\eta = 15$

Figure 3.4 The above set of figures explains the above remark, where Figure (3.4a) indicates faster divergence for metrics of underdamped Langevin diffusion when the stepsize increases to 0.40, and Figure (3.4b) shows a larger friction η would shrink the stepsize range that can be taken not cause a divergence of the metrics.

Similar to what is summarised in section 2.3.4 for the performance of ULA, we would have similar result for the performance of underdamped Langevin algorithm: a too large stepsize would ruin the similarity between the stationary distribution achieved as (3.30) by having a too large stationary variance, but a too small stepsize taken would cause a slow convergence for the distribution at each time to the stationary distribution.

However, unlike ULA, it is really hard to derive the exact distribution of $\mathbf{Z}_t$ at each time t from the chain (3.6), along with the marginal distribution of position chain X_t - indeed, when $t = 3$, the variance of marginal distribution of X_t is already in a very complicated form as $\frac{2\eta\gamma^3(\sigma^4((\eta\gamma-2)^2+1)+(\eta^2\gamma^2\sigma^2-3\eta\gamma\sigma^2-\gamma^2+3\sigma^2)^2)}{\sigma^4}$, which makes us nearly not possible to work out the change of 2-Wasserstein distance along with time as we did for ULA. Therefore, we now instead using computational 1-Wasserstein distance for this purpose, which involves in-built function `wasserstein_distance` in `scipy.stats` in Python.

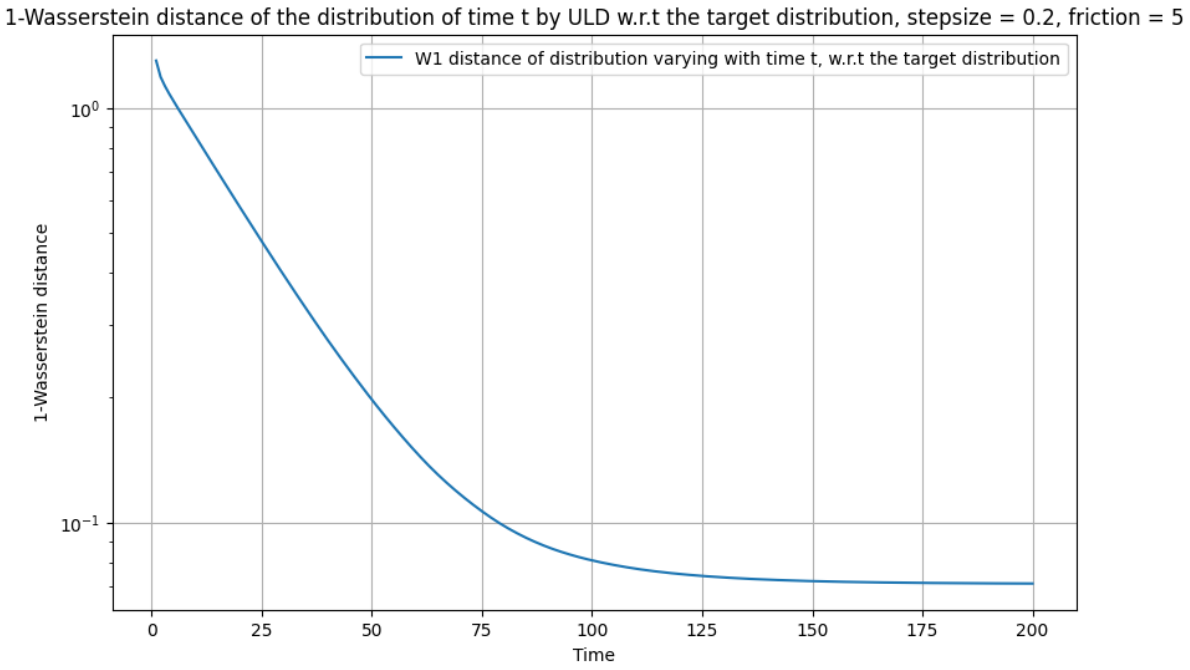


Figure 3.5 Computational 1-Wasserstein distance of the position chain X_t from the marginal distribution of ULD, with $\gamma = 0.2, \eta = 5$. Note the convergence of 1-Wasserstein distance approximately occurs after $t = 150$, meaning a convergence to the stationary marginal distribution (3.30).

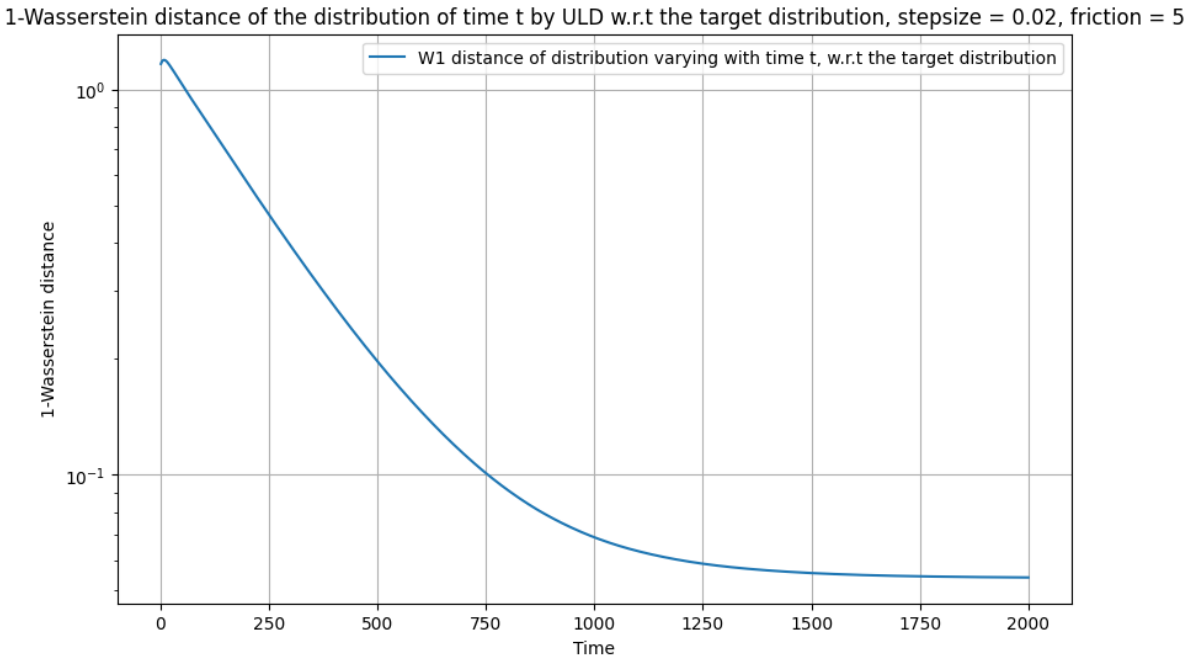


Figure 3.6 Computational 1-Wasserstein distance of the position chain X_t from the marginal distribution of ULD, with $\gamma = 0.02, \eta = 5$. Note this time the convergence of 1-Wasserstein distance approximately occurs after $t = 1500$, implying a ten times slower convergence to (3.30) than the previous one as in Figure 3.5.

Finally, with the form of autocovariance deduced in (3.22), it is also possible to plot the

autocorrelation graph on the position chain $\{X_t\}$ for our ULD process under Euler-Maruyama discretisation. A too small stepsize would also result a large and non-negligible autocorrelation, which means the samples taken are highly correlated.

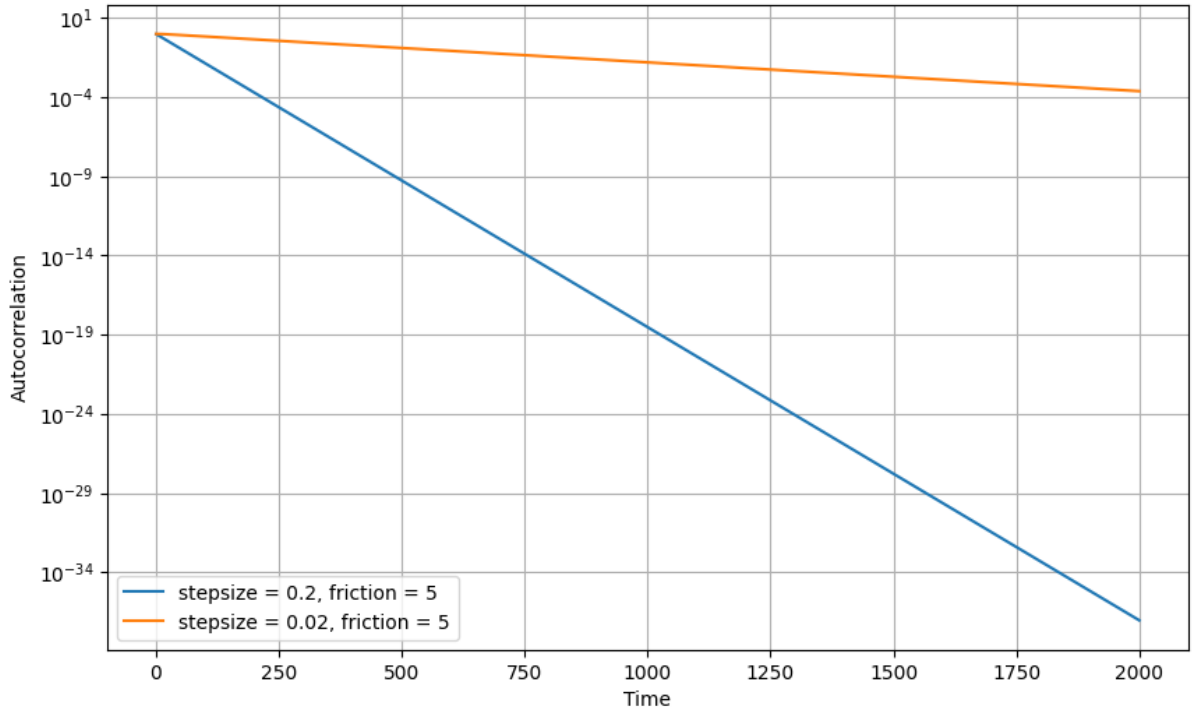


Figure 3.7 The autocorrelation of position chain X_t along with time with different parameters. The autocorrelation is in the log-scale.

3.4 New discretisation on underdamped Langevin SDE

Besides the Euler-Maruyama discretisation for solving the underdamped Langevin SDE system (3.1), Dalalyan and Riou-Durand have introduced another discretisation for solving this SDE system, which is achieved by defining a sequence of functions ψ_t recursively by [8]

$$\psi_0(x) = e^{-\eta x}, \qquad \psi_{t+1}(x) = \int_0^x \psi_t(s)ds \tag{3.33}$$

where as before, η represents the coefficient of friction. By using this sequence ψ_t, the discretisation of (3.1) with $\beta = 1$ involves a stepsize $\gamma > 0$ and is defined by the following recursion

$$\begin{pmatrix} V_{t+1} \\ X_{t+1} \end{pmatrix} = \begin{pmatrix} \psi_0(\gamma)V_t - \psi_1(\gamma)h(X_t) \\ X_t + \psi_1(\gamma)V_t - \psi_2(\gamma)h(X_t) \end{pmatrix} + \sqrt{2\eta} \begin{pmatrix} \xi_{t+1} \\ \xi'_{t+1} \end{pmatrix} \tag{3.34}$$

As before, we still have $h(X_t) = \nabla U(X_t)$, and (ξ_{t+1}, ξ'_{t+1}) is a $2d$-dimensional (i.e. we assume both momentum and position process $V_t, X_t \in \mathbb{R}^d$) centered Gaussian vector satisfying the following conditions:

- (ξ_j, ξ'_j)'s are i.i.d and independent of the initial condition (starting point) (V_0, X_0).
- For any fixed j, if $(\xi_j)_i$ represents the i-th component of vector ξ_j, then the random vectors $((\xi_j)_1, (\xi'_j)_1), ((\xi_j)_2, (\xi'_j)_2), \cdots, ((\xi_j)_d, (\xi'_j)_d)$ are i.i.d with the covariance matrix

$$\mathbf{C} = \int_0^\gamma (\psi_0(t), \psi_1(t))^T (\psi_0(t), \psi_1(t)) dt \tag{3.35}$$

  where $(\psi_0(t), \psi_1(t))^T \in \mathbb{R}^2$ as a 2D vector.

Similar to the previous cases, we would still investigate $d = 1$ for 1-dimensional case. Therefore, we would have

$$\psi_1(t) = \int_0^t e^{-\eta s} ds = \left[-\frac{1}{\eta} e^{-\eta s} \right]_{s=0}^{s=t} = -\frac{1}{\eta}(e^{-\eta t} - 1) \tag{3.36}$$

$$\psi_2(t) = \int_0^t -\frac{1}{\eta}(e^{-\eta s} - 1) ds = \left[\frac{1}{\eta^2} e^{-\eta s} + \frac{1}{\eta} s \right]_{s=0}^{s=t} = \frac{1}{\eta^2}(e^{-\eta t} - 1) + \frac{1}{\eta} t \tag{3.37}$$

Followingly, the covariance matrix is computed as

$$\begin{aligned}
\mathbf{C} = \int_0^\gamma (\psi_0(t), \psi_1(t))^T (\psi_0(t), \psi_1(t)) dt &= \int_0^\gamma \begin{pmatrix} e^{-\gamma t} \\ -\frac{1}{\eta}(e^{-\eta t} - 1) \end{pmatrix} \begin{pmatrix} e^{-\gamma t} & -\frac{1}{\eta}(e^{-\eta t} - 1) \end{pmatrix} dt \\
&= \begin{pmatrix} \int_0^\gamma e^{-2\gamma t} dt & -\int_0^\gamma \frac{1}{\gamma} e^{-\gamma t}(e^{-\gamma t} - 1) dt \\ -\int_0^\gamma \frac{1}{\gamma} e^{-\gamma t}(e^{-\gamma t} - 1) dt & \int_0^\gamma \frac{1}{\gamma^2}(e^{-\gamma t} - 1)^2 dt \end{pmatrix} \\
&= \begin{pmatrix} \frac{1}{2\gamma}\left(1 - e^{-2\gamma^2}\right) & \frac{1}{2\gamma^2}\left(1 - e^{-\gamma^2}\right)^2 \\ \frac{1}{2\gamma^2}\left(1 - e^{-\gamma^2}\right)^2 & \frac{1}{\gamma} - \frac{1}{2\gamma^3}\left(e^{-\gamma^2} - 3\right)\left(e^{-\gamma^2} - 1\right) \end{pmatrix}
\end{aligned}$$

As before, once we consider $U(x)$ be a strongly convex function in terms of quadratic functions, i.e. $U(x) = \frac{1}{2}ax^2 + bx + c$, the sampling density $p_\star$ would be in the Gaussian distribution as in (2.5), with mean $-\frac{b}{a} = \mu_\star$, and the variance with $\frac{1}{a} = \sigma^2$. Therefore, $\nabla U(x) = h(x) = \frac{x - \mu_\star}{\sigma^2}$, and (3.34) could be written as

$$\begin{pmatrix} V_{t+1} \\ X_{t+1} \end{pmatrix} = \begin{pmatrix} \psi_0(\gamma) & -\frac{\psi_1(\gamma)}{\sigma^2} \\ \psi_1(\gamma) & 1 - \frac{\psi_2(\gamma)}{\sigma^2} \end{pmatrix} \begin{pmatrix} V_t \\ X_t \end{pmatrix} + \begin{pmatrix} \psi_1(\gamma)\frac{\mu_\star}{\sigma^2} \\ \psi_2(\gamma)\frac{\mu_\star}{\sigma^2} \end{pmatrix} + \sqrt{2\eta} \begin{pmatrix} \xi_{t+1} \\ \xi'_{t+1} \end{pmatrix} \tag{3.38}$$

with $\psi_0(\gamma), \psi_1(\gamma), \psi_2(\gamma)$, and the covariance matrix $\mathbf{C}$ of (ξ_{t+1}, ξ'_{t+1}) as deduced above. For convenience, we will apply the same technique as the Euler-Maruyama discretisation, by rewriting (3.38) into

$$\mathbf{Z}_{t+1} = \mathbf{A}\mathbf{Z}_t + \mathbf{b} + \sqrt{2\eta}\boldsymbol{\xi}_{t+1} \tag{3.39}$$

but this time we let

$$\mathbf{Z}_t = (V_t, X_t)^T, \quad \mathbf{A} = \begin{pmatrix} \psi_0(\gamma) & -\frac{\psi_1(\gamma)}{\sigma^2} \\ \psi_1(\gamma) & 1 - \frac{\psi_2(\gamma)}{\sigma^2} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} \psi_1(\gamma)\frac{\mu_\star}{\sigma^2} \\ \psi_2(\gamma)\frac{\mu_\star}{\sigma^2} \end{pmatrix}, \quad \boldsymbol{\xi}_{t+1} = \begin{pmatrix} \xi_{t+1} \\ \xi'_{t+1} \end{pmatrix} \tag{3.40}$$

and since $\mathbf{Z}_t$ is assumed to be a centered Gaussian vector, then $\boldsymbol{\xi}_t \sim \mathcal{N}(0, \mathbf{C})$. Note that it is merely in the same form as (3.9), so we would carry same stationary analysis on this newly generated sequence (3.39).

For the mean of the sequence, as before, if we set $\boldsymbol{\mu} = \mathbb{E}(\mathbf{Z}_t)$, then (3.18) still holds here, and by putting $\psi_0(\gamma), \psi_1(\gamma), \psi_2(\gamma)$ into $\mathbf{A}, \mathbf{b}$, trivially we obtain the mean of the process as

$$\boldsymbol{\mu} = (\mathbf{I}_2 - \mathbf{A})^{-1}\mathbf{b} = \begin{pmatrix} \frac{\eta\gamma e^{\eta\gamma} - e^{\eta\gamma} + 1}{\eta\gamma(e^{\eta\gamma} - 1)} & -\frac{1}{\gamma} \\ \frac{\sigma^2}{\gamma} & \frac{\eta\sigma^2}{\gamma} \end{pmatrix} \begin{pmatrix} \frac{\mu_\star(e^{\eta\gamma} - 1)e^{-\eta\gamma}}{\eta\sigma^2} \\ \frac{\mu(\eta\gamma e^{\eta\gamma} - e^{\eta\gamma} + 1)e^{-\eta\gamma}}{\eta^2\sigma^2} \end{pmatrix} = \begin{pmatrix} 0 \\ \mu_\star \end{pmatrix} = \begin{pmatrix} \mathbb{E}(V_t) \\ \mathbb{E}(X_t) \end{pmatrix} \tag{3.41}$$

by using symbolic computations in Python. The above result is sensible, as sharing the same result as (3.19).

For the variance (s_0, for using the same notation) of the sequence, we would also apply the same process and obtain the corresponding Lyapunov equation.

$$s_0 = \mathbf{A}s_0\mathbf{A}^T + 2\eta\mathbf{C} \tag{3.42}$$

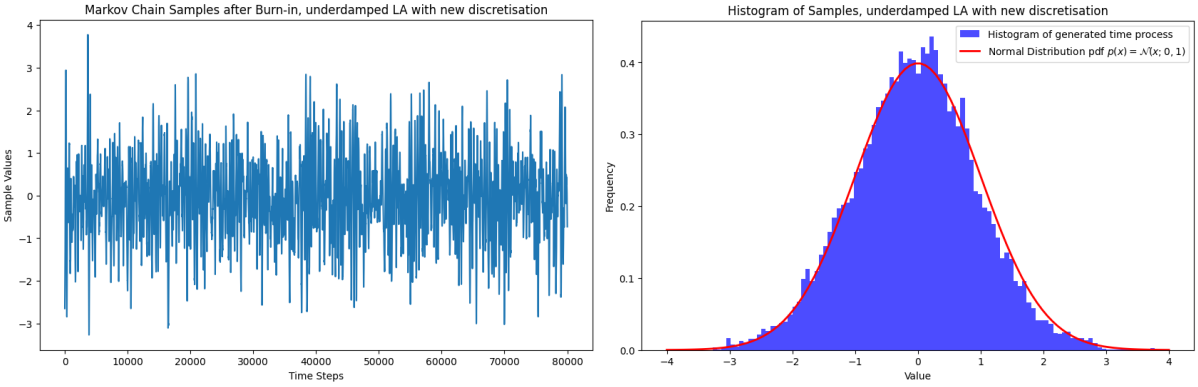
and with the help of **vec** operator, similar as (3.27), if we set $\mathcal{A} = \mathbf{A} \otimes \mathbf{A}$ as the Kronecker product, then the closed-form solution of the variance for chain (3.39) would be derived as

$$\mathbf{vec}(s_0) = (\mathbf{I}_4 - \boldsymbol{\mathcal{A}})^{-1}\mathbf{vec}(2\eta\mathbf{C}) \tag{3.43}$$

Theoretically we are able to derive the explicit form of the variance s_0, so the stationary, marginal distribution of the position chain $\mathbf{X}_t$ could also be deduced as before. However, due to heavy computation cost in the Kronecker product of $\mathbf{A}$, `MemoryError` will occur and Python

is unable to give me the explicit matrix form of s_0. Fortunately, we could simulate this new discretisation with certain values of stepsize γ and friction coefficient η, to see the sampling performance. As the computation of KL divergence or the Wasserstein distance would be rather complicated for this case, we will mainly implement qualitative analysis in this part.

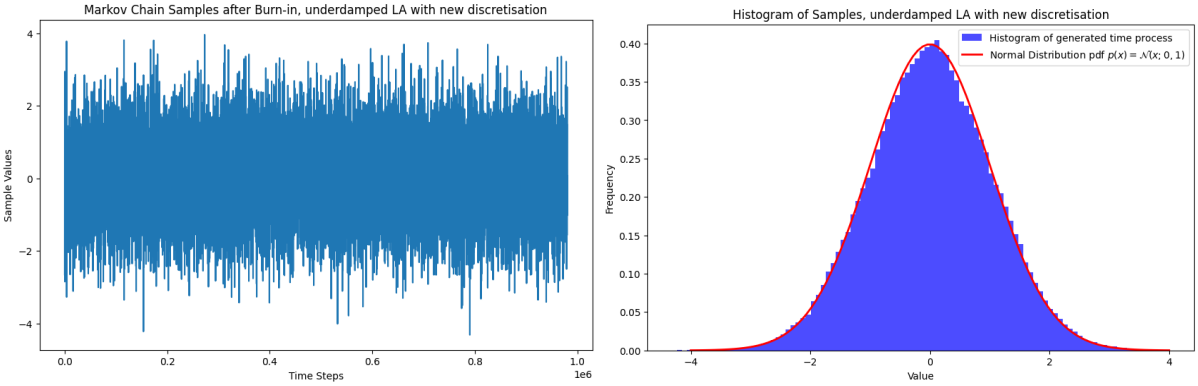
We first implement the sampling performance of this new discretisation with stepsize $\gamma = 0.02$ and $\eta = 0.5$, with total 10^5 samples and burnin the first 20000 samples, see Figure 3.8.



(a) Markov chain samples for the new discretisation, $\gamma = 0.02, \eta = 0.5$, with $N = 10^5$ and $n = 20000$.

(b) Histogram and density plot for new discretisation, $\gamma = 0.02, \eta = 0.5$, with $N = 10^5$ and $n = 20000$.

Figure 3.8 The above set of figures shows the sampling performance of ULD with the new discretisation with $\gamma = 0.02$ and $\eta = 0.5$, taken 10^5 samples and burnin the first 20000 samples. It is obvious that the sampling performance is not so desirable with respect to the target $p_\star$. Compare with good sampling performance for previous cases, the chain shown in Figure (3.8a) still implies the non-negligible correlation between taken samples, which is consistent with the histogram in Figure (3.8b) that does not match the target density well.



(a) Markov chain samples for the new discretisation, $\gamma = 0.02, \eta = 0.5$, with $N = 10^6$ and $n = 20000$.

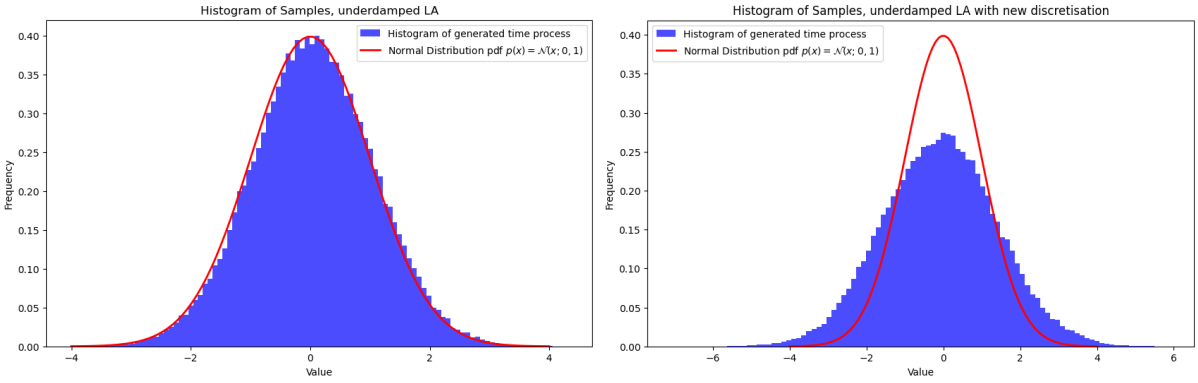
(b) Histogram and density plot for new discretisation, $\gamma = 0.02, \eta = 0.5$, with $N = 10^6$ and $n = 20000$.

Figure 3.9 The above set of figures shows the sampling performance of ULD with the new discretisation with $\gamma = 0.02$ and $\eta = 0.5$, taken 10^6 samples and burnin the first 20000 samples. The chain plot in Figure (3.9a) shows the uncorrelation of samples, which result a much better sampling performance showing in the Figure (3.9b), with a better match between the histogram and the target density plot.

We then increase the number of samples taken from 10^5 to 10^6, with the same amount of burnin samples, to see whether the new discretisation would perform better sampling with more random samples taken. See Figure 3.9.

From Figure (3.8) and (3.9), we may qualitatively state that such discretisation would have a slower rate of convergence to the stationary, marginal distribution of $\mathbf{X}_t$ compared with the Euler-Maruyama discretisation, for showing correlated behaviour of samples in Figure (3.8a).

It is also noticeable that, although they come from the same SDE system, the above new discretisation produces different chain compared with the Euler-Maruyama discretisation. Therefore, for same choice of parameters we would obtain different sampling performance. This idea is illustrated in Figure 3.10.



(a) Histogram and density plot for the Euler-Maruyama discretisation, $\gamma = 0.02, \eta = 0.5$

(b) Histogram and density plot for new discretisation, $\gamma = 0.2, \eta = 5$.

Figure 3.10 The above set of histogram plots shows the different sampling performance from different discretisations, with the same parameters $\gamma = 0.2, \eta = 5$, taken 10^5 samples and burnin the first 20000 samples. Obviously Euler-Maruyama discretisation will give better samples than the new discretisation.

From all above discussions, it is not difficult to see that for the new discretisation, if we aim to achieve a better sampling performance, the choice of stepsize γ should be small, and the friction coefficient η should also be close to γ: one possible implementation can be $\gamma = 0.02, \eta = 0.5$ as shown in Figure 3.8 and 3.9. This is fairly different than the Euler-Maruyama discretisation, where the difference between the γ and η could be rather big, but still produces good sampling: for example, $\gamma = 0.2, \eta = 5$ as mentioned before.

Chapter 4

Conclusion

In conclusion, this report presents a comprehensive analysis of sampling algorithms from both theoretical and empirical perspectives. We have derived analytical expressions for the stationary distributions of discretised algorithms, highlighting the bias introduced by the absence of Metropolis correction. Additionally, we have examined the asymptotic behaviours of discretisations for both unadjusted and underdamped Langevin algorithms, with a particular evaluation on the latter, which has not been extensively covered in existing literature.

Through a comparative study, we evaluated the sampling performance and convergence behavior to the stationary, marginal distribution of these algorithms, considering various step-sizes and friction coefficients (for underdamped Langevin algorithms particularly). By utilizing Kullback-Leibler (KL) divergence and Wasserstein distance as distance measures, we quantitatively assessed the discrepancies between the target distribution and the stationary, marginal distributions. These insights contribute to a deeper understanding of the effectiveness and limitations of different Langevin Monte Carlo sampling methods, guiding their applications in different sampling problems.

4.1 Future perspectives

For the new discretisation of the underdamped Langevin SDE system mentioned in (3.34), although complicated, theoretically it is still feasible to conduct a quantitative analysis of the new discretisation using KL divergence and Wasserstein distance, to measure the difference between the target distribution $p_\star$ and the stationary, marginal distribution of $\mathbf{X}_t$, which will be obtained

from solving (3.43). This approach enables us to examine the impact of different parameter choices, including the stepsize γ and the friction coefficient η, on sampling performance, similar to the analysis conducted for the Euler-Maruyama discretisation shown in Figure (3.1) and (3.3). Through such analysis, it will be possible to gain a better understanding of, and make more informed choices about, the parameters γ and η when using this new discretisation to solve sampling problems.

Furthermore, as Dalalyan and Riou-Durand further stated in their paper 'on sampling from a log-concave density using kinetic Langevin diffusions' [8], when $U(x)$ is also twice differentiable, another discretisation on the underdamped Langevin SDEs can be applied and lead to 'a provably better sampling error, under the condition that the Hessian matrix of $U(x)$ is Lipschitz-continuous with respect to the spectral norm' [8]. A similar analysis throughout this report can be applied to this discretisation to evaluate its sampling performance. This would allow us to compare various discretisations of the same underdamped Langevin SDE system in terms of their efficiency and effectiveness in handling the sampling problem for log-concave distributions.

Last but not least, we can extend our analysis and evaluation to multidimensional contexts. Instead of focusing solely on the 1D Gaussian distribution as our log-concave target, we will consider a multidimensional Gaussian distribution, which is more general and practical for real-life applications. Theorem 6.2 in [2] discusses the convergence of ULA for Gaussian targets in a multidimensional setting, and we may intend to apply the same analysis to the underdamped Langevin samplers.

Appendix

In this report, the Python `sympy` module is frequently employed to derive symbolic expressions for the mean and variance of the sequence generated by the underdamped Langevin sampler algorithm, for both discretization methods. More specifically, various `sympy` packages, including `symbols`, `Matrix`, `eye`, `simplify`, and `kronecker_product` from `sympy.matrices`, are also utilised for obtaining and checking the expressions as shown in (3.17), (3.19), (3.28) and (3.41), also the derivation of covariance matrix $\mathbf{C}$ of $\boldsymbol{\xi}_t = (\xi_t, \xi_t')^T$ for the new discretisation as stated in the last part.

Meanwhile, the Python `matplotlib.pyplot` and `scipy.stats` modules, along with package `norm`, are also used for generating plots. For generation of random numbers from normal distributions, `numpy.random.normal` package is used, and, if without further clarifications, in this report the default seed is set to be 42: `numpy.random.seed(42)`.

Bibliography

[1] M. Taboga. "*Markov Chain Monte Carlo (MCMC) methods*", Lectures on probability theory and mathematical statistics. https://www.statlect.com/fundamentals-of-statistics/Markov-Chain-Monte-Carlo, 2021. [Online; accessed June 2, 2024].

[2] D. Sanz-Alonso and O. Al-Ghattas. A first course in monte carlo methods. pages 61–70, May 25, 2024.

[3] J. Besag. 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, 1994.

[4] U. Grenander and M. I. Miller. Representations of knowledge in complex systems. *Journal of the Royal Statistical Society. Series B, Methodological*, 56(4):549–581, 1994.

[5] G. O. Roberts and R. L. Tweedie. Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli: official journal of the Bernoulli Society for Mathematical Statistics and Probability*, 2(4):341–363, 1996.

[6] Ö. D. Akyildiz and S. Sabanis. "*Nonasymptotic analysis of Stochastic Gradient Hamiltonian Monte Carlo under local conditions for nonconvex optimization*". 2020.

[7] X. Cheng, N. S. Chatterji, P. L. Bartlett, and M. I. Jordan. Underdamped langevin MCMC: A non-asymptotic analysis. January, 2018.

[8] A. S. Dalalyan and L. Riou-Durand. On sampling from a log-concave density using kinetic langevin diffusions. July 2018.

[9] A. Durmus and É. Moulines. "*The Langevin MCMC: Theory and Methods* -

Course 2". https://www.icts.res.in/sites/default/files/paap-2019-08-09-Eric%20Moulines.pdf, August 9, 2019. [Online; accessed February 27, 2024].

[10] N. Brosse, A. Durmus, É. Moulines, and S. Sabanis. The tamed unadjusted langevin algorithm. *Stochastic processes and their applications*, 129(10):3638–3663, 2019.

[11] M. Stigler. "*Stationary models MA, AR and ARMA*". http://matthieustigler.github.io/Lectures/Lect2ARMA.pdf, November 14, 2008. [Online; accessed February 8, 2024].

[12] C. M. Bishop. *Pattern recognition and machine learning*. Information science and statistics. Springer, New York, NY, 2006 - 2006.

[13] D. I. Belov and R. D. Armstrong. Distributions of the kullback-leibler divergence with applications. *British journal of mathematical & statistical psychology*, 64(2):291–309, 2011.

[14] A. Salmona, J. Delon, and A. Desolneux. Gromov-wasserstein distances between gaussian distributions. *Journal of Apllied Probability*, 59(4), 2022. hal-03197398v2.

[15] J. D. Hamilton. *Time series analysis*. Princeton University Press, Princeton, New Jersey, 1994. p. 257-265.

[16] M. Taboga. "*Kronecker product*", Lectures on matrix algebra. https://www.statlect.com/matrix-algebra/Kronecker-product, 2021. [Online; accessed January 30, 2024].

[17] K. Schäcke. "*On the Kronecker Product*". August 1, 2013.

[18] J. Soch et al. Statproofbook/statproofbook.github.io: Statproofbook 2023, chapter 4.1.14. https://zenodo.org/doi/10.5281/zenodo.4305949. [Online; accessed March 5, 2024].