Geophysical Journal International

Geophys. J. Int. (2021) **227,** 1523–1553
Advance Access publication 2021 July 22
GJI General Geophysical Methods

https://doi.org/10.1093/gji/ggab287

Bayesian seismic inversion: a fast sampling Langevin dynamics Markov chain Monte Carlo method

Muhammad Izzatullah,[1] Tristan van Leeuwen[2,3] and Daniel Peter[1]

[1] *Division of Physical Sciences and Engineering, King Abdullah University of Science and Technology (KAUST), 23955, Thuwal, Saudi Arabia. E-mail: muhammad.izzatullah@kaust.edu.sa*
[2] *Centrum Wiskunde & Informatica (CWI), Science Park 123, 1098 XG Amsterdam, The Netherlands*
[3] *Utrecht University, Mathematical Institute, Utrecht, The Netherlands*

Accepted 2021 July 20. Received 2021 June 6; in original form 2020 October 26

SUMMARY

In this study, we aim to solve the seismic inversion in the Bayesian framework by generating samples from the posterior distribution. This distribution incorporates the uncertainties in the seismic data, forward model, and prior information about the subsurface model parameters; thus, we obtain more information through sampling than through a point estimate (e.g. maximum *a posteriori* method). Based on the numerical cost of solving the forward problem and the dimensions of the subsurface model parameters and observed data, sampling with Markov chain Monte Carlo (MCMC) algorithms can be prohibitively expensive. Herein, we consider the promising Langevin dynamics MCMC algorithm. However, this algorithm has two central challenges: (1) the step size requires prior tuning to achieve optimal performance and (2) the Metropolis–Hastings acceptance step is computationally demanding. We approach these challenges by proposing an adaptive step-size rule and considering the suppression of the Metropolis–Hastings acceptance step. We highlight the proposed method's potential through several numerical examples and rigorously validate it via qualitative and quantitative evaluation of the sample quality based on the kernelized Stein discrepancy (KSD) and other MCMC diagnostics such as trace and autocorrelation function plots. We conclude that, by suppressing the Metropolis–Hastings step, the proposed method provides fast sampling at efficient computational costs for large-scale seismic Bayesian inference; however, this inflates the second statistical moment (variance) due to asymptotic bias. Nevertheless, the proposed method reliably recovers important aspects of the posterior, including means, variances, skewness and 1-D and 2-D marginals. With larger computational budget, exact MCMC methods (i.e. with a Metropolis–Hastings step) should be favoured. The results thus obtained can be considered a feasibility study for promoting the approximate Langevin dynamics MCMC method for Bayesian seismic inversion on limited computational resources.

Key words: Numerical approximations and analysis; Probability distributions; Statistical methods; Waveform inversion; Seismic tomography.

1 INTRODUCTION

Seismic inversion addresses the geophysical inverse problem of estimating subsurface model parameters from observed seismic data. Typically, seismic inversion is introduced as an iterative local-optimization problem that attempts to minimize the least-squares residuals between observed and synthetic seismic data. Mathematically, the inverse problem is ill-posed, resulting in non-unique solutions. Practically solving inverse problems is challenging owing to limitations in data acquisition, uncertainties in the observed seismic data and forward modelling and the non-uniqueness of the solution. Several geophysical methods have been proposed to tackle such challenges, which can be broadly classified into two main categories: deterministic methods, such as optimization-based approaches (e.g. Tarantola 1984, 1986; Gauthier *et al.* 1986; Virieux & Operto 2009; Métivier *et al.* 2017; Liu *et al.* 2019a), and statistical methods such as Bayesian inference (e.g. Tarantola & Valette 1982a; Mosegaard & Tarantola 1995; Martin *et al.* 2012; Bui-Thanh *et al.* 2013; Biswas & Sen 2017; Fang *et al.* 2018;

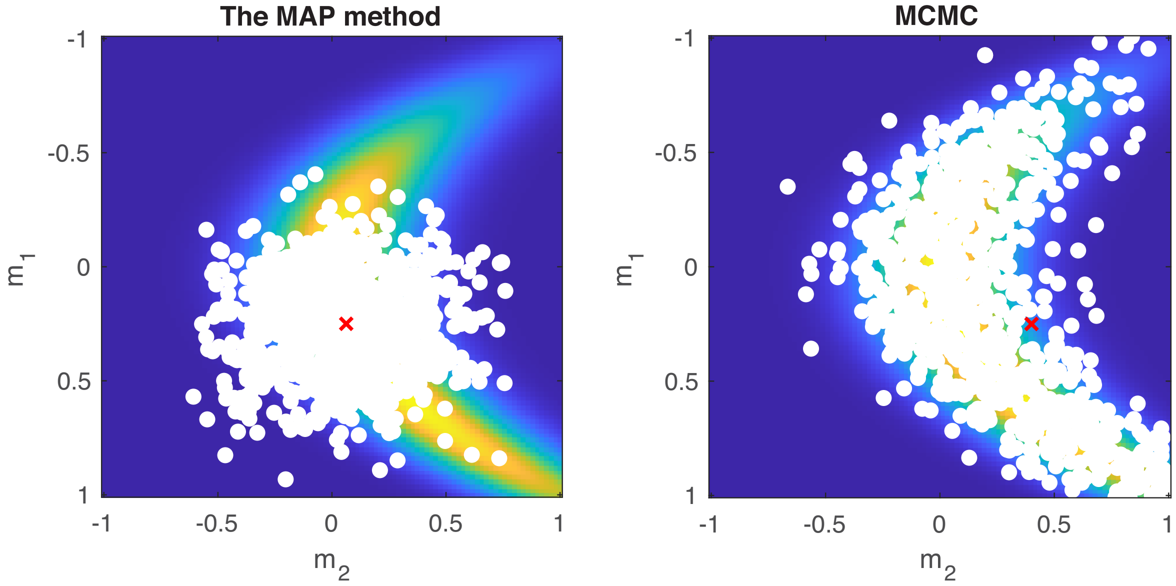

Figure 1. Samples drawn from the bivariate Rosenbrock density by the maximum a posteriori (MAP) method (left-hand panel), and MCMC (right-hand panel), respectively. The red cross in each figure represents the MAP point and the true mean, $\boldsymbol{\mu}$, respectively.

Fichtner & Zunino 2019; Fichtner *et al.* 2019; Izzatullah *et al.* 2019; Zhang & Curtis 2019; Gebraad *et al.* 2020; Koch *et al.* 2020; Zhang & Curtis 2020).

In this study, we focus on the statistical approach within the Bayesian framework, wherein seismic inversion problems are reformulated as statistical inference problems, incorporating uncertainties in the seismic data, forward model and prior information about the subsurface model parameters; this results in a posterior distribution that is the answer to the statistical inference problems. However, direct sampling from this distribution is infeasible; thus, we generate a finite number of samples from it using a Markov chain Monte Carlo (MCMC) algorithm. Based on the numerical cost of solving the forward problem and the dimensions of the subsurface model parameters and observed data, sampling with MCMC methods can be prohibitively expensive. Owing to computational limitations in the context of seismic inversion, the fully Bayesian approach commonly reduces to the maximum *a posteriori* (MAP) method; a method to estimate uncertainties only at the mode of the posterior (e.g. Bui-Thanh *et al.* 2013; Zhu *et al.* 2016; Fang *et al.* 2018; Izzatullah *et al.* 2019; Liu & Peter 2019b). The MAP method or the Laplace's approximation of the MAP model (Tzikas *et al.* 2008), is a technique for incorporating prior knowledge without evaluating the full posterior probability density. However, such a point estimate contains a limited amount of information; hence, it only infers uncertainty about a point, as shown in Fig. 1.

Apart from the MAP method, the full Bayesian inference approach recently received popularity in geophysics for uncertainty quantification. The potential use of sampling methods for seismic inversion has a long history in the geophysics community. Mosegaard & Tarantola (1995) introduced the Monte Carlo sampling method into the geophysics community. Shortly thereafter, Sambridge (1999a) introduced the widely used *Neighbourhood algorithm* which used the concept of so-called natural neighbours of samples in model parameter space to perform approximate importance sampling and to resample the resulting ensemble in a proper probabilistic manner (Sambridge 1999b). Malinverno (2002) introduced the transdimensional Monte Carlo methods into the geophysics community, based on the formalism detailed in (Green 1995), and later introduced the hierarchical and empirical Bayes approaches (Malinverno & Briggs 2004). Bodin & Sambridge (2009) introduced another significant advancement in the sampling methods for geophysical problems by extending the transdimensional MCMC to 2-D geophysical tomography by introducing parametrizations based on the Voronoi tesselations; this opens up opportunities for solving high-dimensional geophysical tomographic problems which previously intractable to solve (Rawlinson *et al.* 2014; Galetti *et al.* 2015). This work later extended to fully solve the 3-D geophysical tomographic problems by Piana Agostinetti *et al.* (2015) and Zhang *et al.* (2018). Recent trend in sampling methods gradually gained recognition through the Hamiltonian Monte Carlo (HMC) algorithm (e.g. Fichtner & Simutė 2018; Fichtner & Zunino 2019; Fichtner *et al.* 2019; Gebraad *et al.* 2020; Koch *et al.* 2020). The HMC algorithms uses Hamilton's equations to evolve a continuous-time Markov process, corresponding to a dynamic system with potential and kinetic energies. In practice, this dynamic is approximated using multiple leapfrog integrator steps, and the HMC is obtained by combining the above dynamic with the Metropolis–Hastings acceptance step. One particular case of the HMC algorithm yields the Metropolis-adjusted Langevin algorithm (MALA) when only one integration step is used between the sampling proposals. MALA belongs to the family of Langevin Monte Carlo (LMC) algorithms, which are derived from the Langevin dynamics (Lemons & Gythiel 1997; Brooks *et al.* 2011). Additionally, approximate MCMC algorithms (Welling & Teh 2011; Ahn *et al.* 2012; Teh *et al.* 2016; Dalalyan 2017; Wibisono 2018; Durmus *et al.* 2019; Durmus & Moulines 2019; Izzatullah *et al.* 2020b; Dalalyan & Riou-Durand 2020; Nemeth & Fearnhead 2020) that suppress the Metropolis–Hastings acceptance steps to improve the mixing rate of sampling algorithms for large-scale problems (e.g. Bayesian seismic inversion problems) are derived from the Langevin dynamics. However, these approximate MCMC algorithms exchange asymptotic correctness for an increased sampling speed resulting in biased inference (Gorham & Mackey 2015; Liu & Wang 2016; Gorham & Mackey 2017; Gorham *et al.* 2019). Both HMC and LMC may be deemed hybrids between the gradient-based optimization approach and derivative-free MCMC methods (e.g. random walks).

Furthermore, we want to highlight another extension of Langevin dynamics that has received research attention from the geophysics community, the *Stein* Variational Gradient Descent (SVGD, Zhang & Curtis 2019, 2020) as an alternative to the MCMC methods. Another extension of the Langevin dynamics is the kernelized Stein discrepancy (KSD) test—a novel MCMC diagnostic test for measuring the asymptotically biased approximation of the posterior distribution (Gorham & Mackey 2015, 2017; Gorham *et al.* 2019; Izzatullah *et al.* 2020a).

Herein, we focus on introducing LMC into the geophysics community, together with adaptation mechanisms to automate LMC step size selection in optimally exploring the target distribution. The contributions of this work to the field under study are as listed below.

(i) Inspired by the work of Dalalyan (2017), Durmus *et al.* (2019) and Nemeth & Fearnhead (2020), we aim to bridge the fields of optimization and Bayesian inference through an approximate LMC algorithm, that is algorithm that excludes the Metropolis–Hastings acceptance step. This approximate LMC algorithm enables geophysicists to quantify the uncertainties in large-scale problems with limited computational resources; however, this approach inflates the second statistical moment (variance) owing to its biasedness.

(ii) We incorporate an adaptive step-size rule into the LMC that adapts the sampling space's geometry through the Lipschitz condition and only introduces minimal computational overhead compared with heuristically searching for the optimal step size; it also provides the algorithm with an optimal sampling speed in exploring the target distribution.

(iii) We rigorously validate the algorithms and measure the MCMC samples' quality through the kernelized Stein discrepancy (KSD) and additional MCMC diagnostics such as trace and autocorrelation function (ACF) plots. KSD measures how well the sample approximates a target distribution and is essential for approximate LMC.

These contributions result in two LMC algorithms with an adaptive step-size rule that alleviates the difficulties inherent to choosing the optimal step size for LMC and provides an optimal sampling speed. These algorithms are called Lip-MALA, and represent an extension of the MALA and Lip-ULA, an algorithm belonging to the approximate MCMC family.

The rest of the manuscript is organized as follows: in Section 2, we describe the problem formulation, focusing on the Bayesian seismic inversion framework. In Section 3, we introduce the Langevin dynamics and their discrete-time approximation as the basis for the LMC. In Section 4, we present our proposed concept, an adaptive step-size rule based on the local smoothness of the probability log-density, to provide an algorithm with optimal sampling speed. We subsequently implement this adaptive step size within the LMC. In Section 5, we highlight the proposed methods through several numerical examples and evaluate the MCMC samples' quality based on kernelized Stein discrepancy (KSD, as described in Appendix A) and other MCMC diagnostics. Finally, we discuss the potential of the proposed methods and its limitations in Section 6.

2 PROBLEM FORMULATION

In seismic inversion, we seek to estimate the subsurface model parameters (e.g. seismic velocities or slowness) $\boldsymbol{m} \in \mathbb{R}^d$ from the observed seismic data $\boldsymbol{D} \in \mathbb{R}^l$, where d and l are the dimension of the model parameters and the observed seismic data, respectively. Following the Bayesian framework, seismic inversion is reformulated as a statistical inference problem that incorporates the uncertainties in the data misfit measurements, forward modelling, and prior information about the subsurface model parameters. The Bayesian framework's solution is the posterior probability density of the subsurface model parameters $\boldsymbol{m}$ given the observed seismic data $\boldsymbol{D}$, which encodes the degree of confidence of their estimate. Thus, we can quantify the subsurface parameters' resulting uncertainty by considering the uncertainties in the seismic data, model and priors. However, to complete such a probability density characterization, typically, we must evaluate numerous samples over the model parameter space. Therefore, the feasibility of Bayesian inference for practical problems strongly depends on the dimensionalities of the model parameters and the data spaces. In this section, we formulate a general framework for seismic Bayesian inference.

2.1 Seismic Bayesian inference framework

Suppose that the relationship between observed the seismic data $\boldsymbol{D}$ and the uncertain subsurface model parameters $\boldsymbol{m}$ is described by

$$\boldsymbol{D} = \boldsymbol{F}(\boldsymbol{m}) + \boldsymbol{\epsilon}, \tag{1}$$

where $\boldsymbol{\epsilon}$ represents noise due to data and/or modelling errors. Given the subsurface model parameters $\boldsymbol{m}$ and the noise $\boldsymbol{\epsilon}$, the forward modelling operator $\boldsymbol{F}$ solves the forward problem to yield $\boldsymbol{D}$; for example, in the context of full-waveform inversion (FWI), the forward modelling operator $\boldsymbol{F}(\boldsymbol{m})$ represents the numerical solution of the seismic wave equation (Tarantola & Valette 1982a; Tarantola 1984; Gauthier *et al.* 1986; Tarantola 1986; Virieux & Operto 2009; Métivier *et al.* 2017; Fang *et al.* 2018; Liu *et al.* 2019a).

To solve Bayesian seismic inversion, we must introduce the notion of prior probability density and the likelihood function. The prior probability density $\boldsymbol{\pi}_{prior}(\boldsymbol{m})$ encodes the confidence of the prior information on the unknown subsurface model parameters $\boldsymbol{m}$, whereas the likelihood function $\boldsymbol{\pi}_{like}(\boldsymbol{D}|\boldsymbol{m})$ describes the conditional probability density for the subsurface model parameters to generate actual seismic data $\boldsymbol{D}$. Based on Bayes' theorem, we obtain the posterior probability density $\boldsymbol{\pi}_{post}(\boldsymbol{m}|\boldsymbol{D})$ by combining the prior probability density and the likelihood function as follows:

$$\boldsymbol{\pi}_{post}(\boldsymbol{m}|\boldsymbol{D}) \propto \boldsymbol{\pi}_{like}(\boldsymbol{D}|\boldsymbol{m})\boldsymbol{\pi}_{prior}(\boldsymbol{m}), \tag{2}$$

where the posterior probability density $\boldsymbol{\pi}_{post}(\boldsymbol{m}|\boldsymbol{D})$ can be evaluated up to its normalizing constant. Herein, we focus on the posterior probability density, which can be written as

$$\boldsymbol{\pi}_{post}(\boldsymbol{m}|\boldsymbol{D}) \propto \exp\big(-J(\boldsymbol{m}, \boldsymbol{D})\big), \tag{3}$$

where $J(\boldsymbol{m}, \boldsymbol{D})$ is the negative log-posterior density, and it can be written as $J(\boldsymbol{m}, \boldsymbol{D}) = -\log \boldsymbol{\pi}_{like}(\boldsymbol{D}|\boldsymbol{m}) - \log \boldsymbol{\pi}_{prior}(\boldsymbol{m})$. $J(\boldsymbol{m}, \boldsymbol{D})$ can be considered as the misfit function of any deterministic seismic inversion. We assume that $J(\boldsymbol{m}, \boldsymbol{D})$ is continuously differentiable (Tarantola & Valette 1982a, b; Mora 1987; Plessix 2006) and this implies that its gradient $\partial J(\boldsymbol{m}, \boldsymbol{D})/\partial \boldsymbol{m}$ is Lipschitz-continuous with Lipschitz constant L_J (Kantorovich & Akilov 1982), that is

$$\|\nabla_{\boldsymbol{m}} J(\boldsymbol{m}, \boldsymbol{D}) - \nabla_{\boldsymbol{m}'} J(\boldsymbol{m}', \boldsymbol{D})\| \leq L_J \|\boldsymbol{m} - \boldsymbol{m}'\|. \tag{4}$$

Note that these assumptions are significant for ensuring all gradient-based algorithms (e.g. optimization or sampling algorithms) are well defined (Nocedal & Wright 2006), including HMC and LMC. For instance, when $\nabla_{\boldsymbol{m}} J(\boldsymbol{m}, \boldsymbol{D})$ is not Lipschitz continuous (which is commonly observed for the ℓ_1 or total-variation priors), the approximate LMC is generally explosive and MALA is not geometrically ergodic (Roberts & Tweedie 1996; Pereyra 2016). In addition, the complete evaluation of eqs (2) or (3), particularly in high dimensions, may be intractable to compute and impossible to interpret. Thus, we refer to MCMC algorithms to evaluate the posterior distributions.

MCMC algorithms are the standard technique for generating samples from a probability density. In particular, the Metropolis–Hastings method is an MCMC algorithm used for generating a sequence of random samples from a probability density for which direct sampling is infeasible. It applies a given proposal probability density $\mathcal{Q}(\boldsymbol{m}_t, \boldsymbol{y})$ at each sample point in the model parameter space $\boldsymbol{m}_t$ to generate a proposed sample point $\boldsymbol{y}$; once generated, the algorithm chooses to either accept or reject the proposed sample point and repeats from the new point, thereby generating a chain of samples from the posterior probability density $\boldsymbol{\pi}_{post}(\boldsymbol{m}|\boldsymbol{D})$. This process is well-known as the Metropolis–Hastings acceptance step (Brooks *et al.* 2011).

Theoretically, the Metropolis–Hastings acceptance step guarantees that the generated chain of samples comprises asymptotically unbiased samples generated from the target posterior probability density $\boldsymbol{\pi}_{post}(\boldsymbol{m}|\boldsymbol{D})$ as the number of sample points tends to infinity. However, an MCMC algorithm's success critically depends on the design of its proposal probability density. A properly designed proposal probability density yields an optimal acceptance rate and good sample quality.

3 LANGEVIN DYNAMICS MCMC

3.1 The Langevin dynamics

Langevin dynamics are named for the French physicist Paul Langevin, who developed them in 1908. Langevin dynamics apply Newton's second law to a representative Brownian motion to simplify Albert Einstein's approach to Brownian motion (Lemons & Gythiel 1997). However, in the LMC context, we consider the overdamped Langevin dynamics (i.e. Langevin dynamics without acceleration) to sample the posterior distribution. These dynamics are defined by

$$\mathrm{d}\boldsymbol{m}(t) = -\boldsymbol{\Sigma}\,\nabla \log \boldsymbol{\pi}(\boldsymbol{m}(t))\mathrm{d}t + \sqrt{2}\,\boldsymbol{\Sigma}^{\frac{1}{2}}\,\mathrm{d}\boldsymbol{W}(t), \tag{5}$$

where $\nabla \log \boldsymbol{\pi}(\boldsymbol{m}(t))$ is the drift term pushing the Brownian particle $\boldsymbol{m}(t)$ around in velocity space $\mathrm{d}\boldsymbol{m}(t)$; $\boldsymbol{W}(t)|_{t\geq 0}$ is a standard d-dimensional Brownian motion, with $\boldsymbol{\pi}$ as its stationary distribution; and $\boldsymbol{\Sigma}$ is a symmetric positive definite operator.

3.2 Approximate Langevin dynamics MCMC

The overdamped Langevin dynamics are a suitable candidate for an MCMC algorithm with $\boldsymbol{\pi}$ as their stationary distribution, and in our context, $\boldsymbol{\pi}$ refers to the posterior distribution as defined by eqs (2) or (3). In practice, we simulate eq. (5) by discretizing it using the Euler–Maruyama scheme (Stuart *et al.* 2004). This discretization produces the discrete-time Markov chain given by

$$\boldsymbol{m}_{t+1} = \boldsymbol{m}_t + \tau_t\,\boldsymbol{\Sigma}\,\nabla \log \boldsymbol{\pi}(\boldsymbol{m}_t) + \sqrt{2\tau_t}\,\boldsymbol{\Sigma}^{\frac{1}{2}}\,\boldsymbol{\xi}_t, \tag{6}$$

where $\boldsymbol{\xi}_t \sim \mathcal{N}(0, \boldsymbol{I}_{d\times d})$ is a vector of d-dimensional standard Gaussian random variables; $\boldsymbol{\Sigma}$ is a symmetric positive definite matrix; and τ_t is a step size, which can be fixed or varied for all steps. This scheme is equivalent to a gradient ascent procedure with injected Gaussian noise, wherein the gradient term $\nabla \log \boldsymbol{\pi}(\boldsymbol{m}_t)$ drives $\boldsymbol{m}_t$ towards points at which $\boldsymbol{\pi}$ has a high probability with $\boldsymbol{\Sigma}$ as a pre-conditioner. Whereas, the injected noise $\boldsymbol{\xi}_t$ prevents the chain from collapsing to the (local) maximum.

This scheme was first introduced in the field of molecular dynamics (e.g. Ermak 1975; Parisi 1981) and then popularized in artificial intelligence with several variations (e.g. Welling & Teh 2011; Ahn *et al.* 2012; Korattikara *et al.* 2014; Raginsky *et al.* 2017). Following

Table 1. Summary of the Langevin dynamics MCMC algorithms based on eq. (6).

Methods	Step size, τ	Pre-conditioning, $\boldsymbol{\Sigma}$	Metropolis–Hastings
Unadjusted Langevin Algorithm (ULA)	Constant	$\boldsymbol{\Sigma} = \boldsymbol{I}_{d\times d}$ or $\boldsymbol{\Sigma} = \boldsymbol{P}_{d\times d}$	No
Lipschitz-ULA (Lip-ULA)	Adaptive	$\boldsymbol{\Sigma} = \boldsymbol{I}_{d\times d}$ or $\boldsymbol{\Sigma} = \boldsymbol{P}_{d\times d}$	No
Stochastic Gradient Langevin Dynamics (SGLD)	Constant	$\boldsymbol{\Sigma} = \boldsymbol{I}_{d\times d}$ or $\boldsymbol{\Sigma} = \boldsymbol{P}_{d\times d}$	No
Metropolis-Adjusted Langevin algorithm (MALA)	Constant	$\boldsymbol{\Sigma} = \boldsymbol{I}_{d\times d}$ or $\boldsymbol{\Sigma} = \boldsymbol{P}_{d\times d}$	Yes
Lipschitz-MALA (Lip-MALA)	Adaptive	$\boldsymbol{\Sigma} = \boldsymbol{I}_{d\times d}$ or $\boldsymbol{\Sigma} = \boldsymbol{P}_{d\times d}$	Yes
Stochastic Newton MCMC (SN-MCMC)	$\tau = 1$	$\boldsymbol{\Sigma} = \boldsymbol{P}_{d\times d}$	Yes

Roberts & Tweedie (1996), the algorithm is referred to as an unadjusted Langevin algorithm (ULA) and it makes direct use of $\boldsymbol{m}_t$ as MCMC samples to approximate the probabilities and expectations with respect to $\boldsymbol{\pi}$. Whereas, if at each iteration $\boldsymbol{m}_t$ is being used as a proposal sample which is then accepted or rejected based on the Metropolis–Hastings criterion then, it is known as the Metropolis-adjusted Langevin algorithm (MALA). ULA is a simpler version of MALA, which does not use a Metropolis–Hastings acceptance step; thus, the MCMC samples produce a biased approximation of $\boldsymbol{\pi}$ (Roberts & Tweedie 1996; Dwivedi *et al.* 2018; Nemeth & Fearnhead 2020).

Recently, ULA has attracted significant research attention, particularly for high-dimensional inference problems on which most MCMC methods struggle. ULA is computationally efficient for sampling high-dimensional probabilities owing to the absence of the Metropolis–Hastings acceptance step. Note that ULA has traditionally been regarded as unreliable because of its asymptotic bias, which stems from the discretization error. According to Durmus & Moulines (2017), for constant step sizes, ULA converges to a unique stationary distribution that differs from $\boldsymbol{\pi}$ in most cases. However, to reduce the asymptotic bias, ULA often requires a sufficiently small step size, thus necessitating several iterations to correctly sample the target distribution (Wibisono 2018; Durmus & Moulines 2019). Moreover, because of this asymptotic bias, even for an appropriately implemented ULA algorithm, the distribution from which one samples will asymptotically have the correct mean or mode but will inflate the variance (Brosse *et al.* 2018; Nemeth & Fearnhead 2020). Nevertheless, ULA has advantageous computational costs because of its fast sampling. Despite ULA's advantages, the algorithms should be preferred according to the limitations of computational budget; if a large budget is available, one should favour exact methods such as MALA and HMC over the approximate Langevin dynamics MCMC method. As the computing budget increases, the exact MCMC methods will eventually become more accurate.

3.3 General challenges of the Langevin dynamics MCMC algorithm

Typically, MCMC methods derived from Langevin dynamics encounter two primary challenges: (i) the step size requires tuning and (ii) introducing the Metropolis–Hastings acceptance step substantially raises computational costs with the increasing rejection rate. The first challenge shares a similar issue with the gradient descent algorithm used for optimization. The step size governs the extent to which the drift of the Langevin dynamics can change, and according to eq. (4), the step size should be less than $1/L_J$ to avoid instability in the Euler–Maruyama discretization. Introducing the Metropolis–Hastings acceptance step into ULA to correct the asymptotic bias resulting from the discretization error leads to the second challenge. This approach, known as MALA, corresponds to using a Gaussian proposal distribution with mean $\boldsymbol{m}_t + \tau_t\,\boldsymbol{\Sigma}\,\nabla\log\boldsymbol{\pi}(\boldsymbol{m}_t)$ and covariance matrix $2\tau_t\boldsymbol{\Sigma}$:

$$\mathcal{Q}_{MALA}(\boldsymbol{m}_t, \boldsymbol{y}) = \mathcal{N}\big(\boldsymbol{y}; \boldsymbol{m}_t + \tau_t\,\boldsymbol{\Sigma}\,\nabla\log\boldsymbol{\pi}(\boldsymbol{m}_t), 2\tau_t\boldsymbol{\Sigma}\big). \tag{7}$$

However, choosing an appropriate proposal density for the Metropolis–Hastings acceptance step is non-trivial. As the rejection rate increases, the computational cost for accepting a proposal sample increases. Despite these challenges, LMC is superior compared with conventional random-walk MCMC methods (Brooks *et al.* 2011), whereby the assistance of the gradient of log-density $\nabla\log\boldsymbol{\pi}$ directs the proposed samples toward areas of high probability in density $\boldsymbol{\pi}$. We briefly summarize the LMC in Table 1, including two additional algorithms proposed in the next section, Lip-ULA and Lip-MALA.

4 LOCALLY LIPSCHITZ ADAPTIVE STEP SIZE

Step size τ can be tuned such that the MCMC methods achieve better mixing performance; however, this is non-trivial and problem-dependent. Theoretically, the optimal scaling of τ for MALA with dimension d is $\tau_{\mathrm{opt}}\propto d^{-1/3}$, which has an algorithmic complexity $\mathcal{O}(d^{1/3})$ (Roberts & Rosenthal 1998). Thus, τ must decrease with dimension d. Fixing the acceptance step with a small τ value provides to a large acceptance ratio; however, all accepted steps are small (on the order of τ on average), such that the sampling algorithm moves often, but slowly. Typically, approximately $\mathcal{O}(1/\tau)$ iterations are required to move through the support of the target probability density after the burn-in period (Roberts & Rosenthal 1998; Neal & Roberts 2006).

As mentioned earlier in Section 3, approximate MCMC methods offer the advantage of fast sampling owing to the exclusion of the Metropolis–Hastings acceptance step; however, the resulting samples are biased. We also understand (as will be demonstrated in Section 5) that the samples from approximate MCMC methods asymptotically have the correct mean or mode but will inflate the variance (Brosse *et al.* 2018; Nemeth & Fearnhead 2020). To control the bias and the variance, the step sizes need to decrease to zero; this will decelerate the algorithm's mixing rate and generate an increasing number of iterations to guarantee that the algorithm samples from the correct target probability density (Welling & Teh 2011; Teh *et al.* 2016; Durmus & Moulines 2017, 2019). We can also introduce a proximal gradient step as an alternative approach (Wibisono 2018); however, these approaches all potentially require several iterations to correctly sample the target distribution.

Here, we address the challenge of step size tuning by proposing an adaptive step size based on the local smoothness of the log-probability density $\boldsymbol{\pi}$. We borrow this idea from optimization algorithms (Polyak 1963, 1969; Drori & Teboulle 2014; Kim & Fessler 2016; Malitsky & Mishchenko 2019) in which the step size τ_t is chosen at each iteration as a particular approximation of the inverse of the local Lipschitz constant, L_J^{-1}. Consequently, the step size τ adapts to the local geometry of the misfit function. Motivated by the work of Dalalyan (2017) and Durmus *et al.* (2019), we incorporate adaptive step size τ based on the local geometry of the target probability log-density into the LMC algorithms.

To compute the adaptive step size at each iteration, we consider the following two inequalities:

$$\begin{cases} \tau_t^2 \leq (1+\alpha_t)\tau_{t-1}^2 \\ \tau_t \leq L_C \frac{\|\boldsymbol{m}_t - \boldsymbol{m}_{t-1}\|}{\left\|\boldsymbol{\Sigma}\nabla \log \boldsymbol{\pi}(\boldsymbol{m}_t) - \boldsymbol{\Sigma}\nabla \log \boldsymbol{\pi}(\boldsymbol{m}_{t-1})\right\|} \end{cases}, \tag{8}$$

where α_t represents the ratio between two consecutive step sizes. In this context, we use the convention $\frac{1}{0} = +\infty$; thus if $\nabla \log \boldsymbol{\pi}(\boldsymbol{m}_{t+1}) - \nabla \log \boldsymbol{\pi}(\boldsymbol{m}_{t+1}) = 0$, the second inequality can be ignored. Intuitively, these inequalities limit the speed at which the drift of Langevin dynamics changes according to the geometry of the log-probability density and it is being controlled by the coefficient L_C. Theoretically, this is important for specifying the appropriate step size, which should be less than the presented equalities to ensure that the proposed scheme is stable. Furthermore, by introducing the pre-condition $\boldsymbol{\Sigma}$ to the gradient of log-density, we obtain further information on the local geometry of the target probability density.

Incorporating the local geometric information regarding the target probability log-density into a step size is an alternative to choosing an appropriate proposal density for the Metropolis–Hastings acceptance step, as mentioned in the original work of Hastings (1970). According to Hastings (1970), the proposal density $\mathcal{Q}(\boldsymbol{m}_t, \boldsymbol{y})$ should be selected such that the proposed sample point $\boldsymbol{y}$ is not too far from $\boldsymbol{m}_t$; otherwise, the Metropolis–Hastings criterion will be small and the proposed sample point is likely to be rejected. Thus, we substitute the proposal density concept with the locally Lipschitz adaptive step size. Through this adaptive step size, we ensure that the next sample point $\boldsymbol{m}_{t+1}$ will be close enough to the current sample point $\boldsymbol{m}_t$ (i.e. within the local neighbourhood of $\boldsymbol{m}_t$) depending on the Lipschitz constant. In addition, the inequalities described above suggest that a proper step size should be used for ensuring that ULA and MALA will behave accordingly, that is the scheme will be stable and geometrically ergodic (Roberts & Tweedie 1996; Pereyra 2016).

We allow the constant factor L_C in the second inequality to be a free parameter, which penalizes the inverse of the Lipschitz constant, L_J^{-1}. According to Roberts & Rosenthal (1998), the step size τ should be tuned to approximately $\ell^2\, d^{-1/3}$, where ℓ^2 is a constant that controls the asymptotic acceptance probability. Through direct comparison, herein, we consider $L_C = d^{-1/3}$ according to $\tau_{opt} \propto d^{-1/3}$, which is the optimal scaling of MALA. This will further penalize the inverse of the Lipschitz constant as the dimension of the model parameters increases.

As we implemented this locally Lipschitz adaptive step size within MALA, we name the algorithm *Lipschitz-MALA (Lip-MALA)*. Algorithm 1 presents the pseudocode for implementing Lip-MALA. Furthermore, we incorporate this adaptive step size into ULA, creating *Lipschitz-ULA (Lip-ULA)*, which is presented in Algorithm 2. Compared with ULA and MALA, Lip-ULA and Lip-MALA can evaluate the posterior density by naturally exploiting the local geometry of the target probability log-density. This adaptive step size algorithm is generally universal and can be implemented within any LMC algorithm under the assumptions that $\log \boldsymbol{\pi}$ is continuously differentiable and that its gradient $\nabla \log \boldsymbol{\pi}$ is Lipschitz, as described by eq. (4) and explained in Section 3. To corroborate this idea, we focus on the implementation of Lip-ULA and Lip-MALA and study their performance in a Bayesian seismic inversion framework.

5 NUMERICAL EXAMPLES

In this section, we highlight the potentials of Lip-ULA and Lip-MALA in comparison to MALA on four numerical examples: (1) a bivariate Gaussian density, (2) a bivariate Rosenbrock density, (3) a low-dimensional model space FWI based on the Camembert model (Gauthier *et al.* 1986) and (4) a high-dimensional model space FWI based on the Marmousi model (Brougois *et al.* 1990). The posterior density in numerical example (1) is Gaussian, whereas the others are not.

For MALA, reasonable acceptance rate lies within the interval of [40,80] per cent. In providing consistent comparisons, we consider a step size that approximately provides an asymptotically optimal acceptance rate of 57.4 per cent for MALA (Roberts & Rosenthal 1998; Brooks *et al.* 2011). We use similar initial step sizes for Lip-ULA and Lip-MALA, allowing slightly higher or lower acceptance rates for MALA and Lip-MALA. However, the acceptance rate alone provides little information concerning the performance of MCMC algorithms; thus, we conduct additional MCMC diagnostics such as trace and ACF plots.

We validate the simulations via qualitative and quantitative evaluation of the sample quality based on conventional MCMC diagnostics for specifically chosen parameters and KSD. The KSD is a novel MCMC diagnostic that is similar in spirit to the central limit theorem (CLT). KSD measures an asymptotically biased approximation of the posterior distribution and monitors the sample convergence to the target density. Readers are referred to Appendix A for the details of the KSD calculations.

Algorithm 1 Metropolis-adjusted Langevin algorithm with locally Lipschitz adaptive step size (Lip-MALA MCMC)

```
Require: $\boldsymbol{m}_0, \tau_0 > 0, \alpha_0 = +\infty, L_C = d^{-1/3}, \boldsymbol{\Sigma} = \boldsymbol{I}_{d\times d}$ or $\boldsymbol{\Sigma} = \boldsymbol{P}_{d\times d}$
Ensure: $\boldsymbol{m}_0, \dots, \boldsymbol{m}_N$
  for $t = 1$ to $N$ do
    Draw the diffusion vector $\xi_t \sim \mathcal{N}(0, \boldsymbol{I}_{d\times d})$.
    Propose $\boldsymbol{m}' = \boldsymbol{m}_{t-1} - \tau_{t-1}\boldsymbol{\Sigma}\nabla \log \boldsymbol{\pi}(\boldsymbol{m}_{t-1}) + \sqrt{2\tau_{t-1}}\boldsymbol{\Sigma}^{1/2}\xi_{t-1}$
    Compute the accept–reject probability
    $\gamma(\boldsymbol{m}'|\boldsymbol{m}_{t-1}) = \min\left(1, \frac{\boldsymbol{\pi}(\boldsymbol{m}')\mathcal{N}(\boldsymbol{m}'|\boldsymbol{m}_{t-1}-\tau_{t-1}\boldsymbol{\Sigma}\nabla \log \boldsymbol{\pi}(\boldsymbol{m}_{t-1}),2\tau_{t-1}\boldsymbol{\Sigma})}{\boldsymbol{\pi}(\boldsymbol{m}_{t-1})\mathcal{N}(\boldsymbol{m}_{t-1}|\boldsymbol{m}'-\tau_{t-1}\boldsymbol{\Sigma}\nabla \log \boldsymbol{\pi}(\boldsymbol{m}'),2\tau_{t-1}\boldsymbol{\Sigma})}\right)$
    Draw $u \sim \mathcal{U}(0, 1)$.
    if $u < \gamma(\boldsymbol{m}'|\boldsymbol{m}_t)$ then
      Accept: $\boldsymbol{m}_t = \boldsymbol{m}'$
      Update $\tau_t = \min\left\{\sqrt{(1+\alpha_{t-1})}\tau_{t-1}, L_C \frac{\|\boldsymbol{m}_t - \boldsymbol{m}_{t-1}\|}{\left\|\boldsymbol{\Sigma}\nabla \log \boldsymbol{\pi}(\boldsymbol{m}_t) - \boldsymbol{\Sigma}\nabla \log \boldsymbol{\pi}(\boldsymbol{m}_{t-1})\right\|}\right\}$
      Update $\alpha_t = \tau_t/\tau_{t-1}$
    else
      Reject: $\boldsymbol{m}_t = \boldsymbol{m}_{t-1}$
    end if
  end for
```

Algorithm 2 An unadjusted Langevin algorithm with locally Lipschitz adaptive step size (Lip-ULA MCMC)

```
Require: $\boldsymbol{m}_0, \tau_0 > 0, \alpha_0 = +\infty, L_C = d^{-1/3}, \boldsymbol{\Sigma} = \boldsymbol{I}_{d\times d}$ or $\boldsymbol{\Sigma} = \boldsymbol{P}_{d\times d}, \boldsymbol{m}_1 = \boldsymbol{m}_0 - \tau_0\boldsymbol{\Sigma}\nabla \log \boldsymbol{\pi}(\boldsymbol{m}_0) + \sqrt{2\tau_0\boldsymbol{\Sigma}}\boldsymbol{\xi}_0$
Ensure: $\boldsymbol{m}_0, \dots, \boldsymbol{m}_N$
  for $t = 1$ to $N$ do
    Compute the step size $\tau_t = \min\left\{\sqrt{(1+\alpha_{t-1})}\tau_{t-1}, L_C \frac{\|\boldsymbol{m}_t - \boldsymbol{m}_{t-1}\|}{\left\|\boldsymbol{\Sigma}\nabla \log \boldsymbol{\pi}(\boldsymbol{m}_t) - \boldsymbol{\Sigma}\nabla \log \boldsymbol{\pi}(\boldsymbol{m}_{t-1})\right\|}\right\}$
    Compute $\alpha_t = \tau_t/\tau_{t-1}$
    Draw the diffusion vector $\xi_t \sim \mathcal{N}(0, \boldsymbol{I}_{d\times d})$
    Compute $\boldsymbol{m}_{t+1} = \boldsymbol{m}_t - \tau_t\boldsymbol{\Sigma}\nabla \log \boldsymbol{\pi}(\boldsymbol{m}_t) + \sqrt{2\tau_t}\boldsymbol{\Sigma}^{1/2}\boldsymbol{\xi}_t$
  end for
```

Table 2. Summary of the statistical results for Numerical Example 1 with bivariate Gaussian posterior density.

Methods	Mean, $(\boldsymbol{\mu_1}, \boldsymbol{\mu_2})$	Variance, $(\boldsymbol{\sigma_1^2}, \boldsymbol{\sigma_2^2})$	Acceptance rate, per cent
Truth	(0.4000, 0.4000)	(0.3022, 0.3022)	-
Lip-ULA	(0.3914, 0.4004)	(0.4544, 0.4528)	100.00 per cent
Lip-MALA	(0.3969, 0.3994)	(0.2929, 0.2969)	69.88 per cent
MALA	(0.3902, 0.4099)	(0.3004, 0.3089)	57.43 per cent

5.1 Bivariate Gaussian density

We consider a bivariate Gaussian posterior density as follows:

$$\boldsymbol{\pi}(\boldsymbol{m}|\boldsymbol{D}) \propto \exp\left(-\frac{1}{2}||\boldsymbol{A}\boldsymbol{m} - \boldsymbol{D}||_2^2 - \frac{1}{2}||\boldsymbol{L}\boldsymbol{m}||_2^2\right), \tag{9}$$

with $\boldsymbol{A} = \begin{pmatrix} 2 & 0.5 \\ 0.5 & 2 \end{pmatrix}$, $\boldsymbol{D} = [1, 1]^T$, and $\boldsymbol{L} = 10^{-3} \times \begin{pmatrix} 0.5 & 0 \\ 2 & 0 \end{pmatrix}$. For this example, we use a step size of $\tau = 2.6 \times 10^{-1}$. For all algorithms, we consider no pre-conditioning and set $\boldsymbol{\Sigma} = \boldsymbol{I}_{2\times 2}$. We sample the posterior with the initial model $\boldsymbol{m}_0 = [0, 0]^T$ for $N = 30\,000$ samples and discard the first half from the total number of samples as burn-in. The statistical results for this example are tabulated in Table 2.

Fig. 2 displays the samples drawn from the bivariate Gaussian posterior density in eq. (9) using the three different LMC algorithms described above. All of these algorithms approximately converge to the true mean $\boldsymbol{\mu}$, and their samples are accordingly distributed within the target density. Based on the results in Table 2, all three algorithms approximate the mean well; however, we observe that Lip-ULA records a higher variance than the other two. As described previously, this situation occurs owing to Lip-ULA being an asymptotically biased algorithm, that is without Metropolis–Hastings correction steps.

To further assess the results of this example, we perform MCMC diagnostics to evaluate the MCMC sample quality. We generate individual trace plots for each parameter of the bivariate Gaussian density in Fig. 3 to assess the chain convergence. Therein, we observe that the algorithms are mixing well and have reached the stationary region of the target density $\boldsymbol{\pi}(\boldsymbol{m}|\boldsymbol{D})$; the ACF is plotted in the left of Fig. 4.

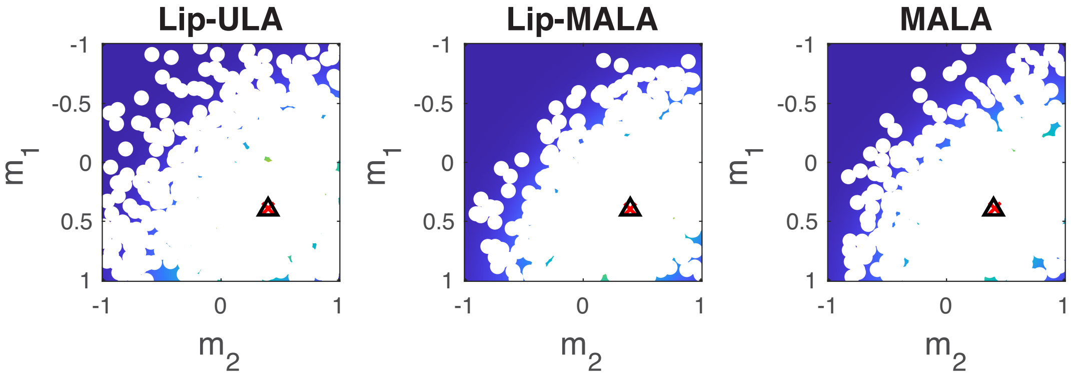

Figure 2. Samples drawn from the bivariate Gaussian density are plotted for three different Langevin dynamics MCMC algorithms: (a) Lip-ULA, (b) Lip-MALA and (c) MALA. The black triangle in each figure represents the true mean, $\boldsymbol{\mu}$, whereas the red cross represents the sample mean, $\overline{\boldsymbol{m}}$.

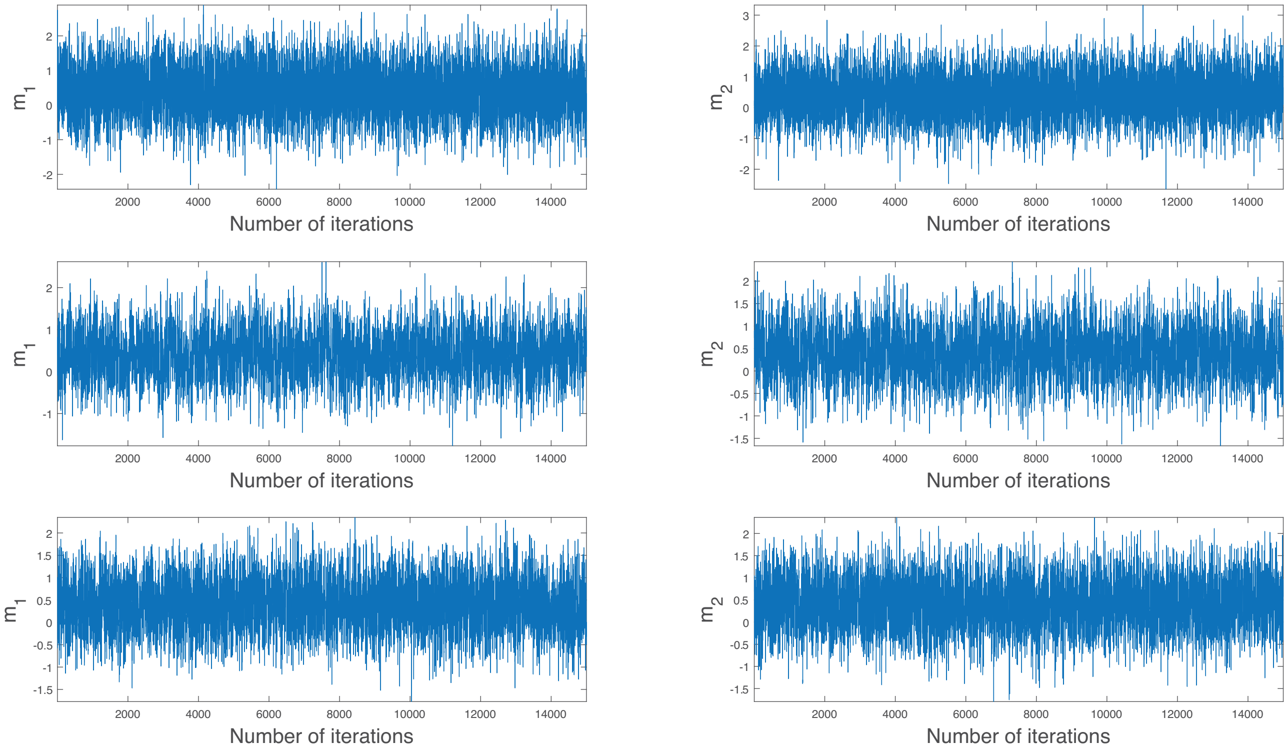

Figure 3. The trace plots for each bivariate Gaussian density parameter, $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$. First row: Lip-ULA; second row: Lip-MALA and third row: MALA.

By inspecting the chain and ACF plots, we can obtain a sense of the samples' degree of serial correlation and measure it precisely. The left of Fig. 4 shows that the samples' ACFs quickly decay to zero, indicating that all three algorithms only need few iterations to traverse the whole sample space of model parameters.

The trace and ACF plots measure convergence to the stationary distribution but do not consider the asymptotic bias. To validate our proposed methods (particularly Lip-ULA), we use KSD to measure sample quality. We compute and plot the KSD values at the right of Fig. 4. For the exact MCMC algorithms (i.e. Lip-MALA and MALA), convergence in KSD is expected; however, this is not the case for Lip-ULA, which we expect to reach a plateau as the number of finite samples N increases because of high variance. This situation is common for approximate MCMC algorithms owing to their asymptotic bias, but we would expect the mean to be more accurate than the variance. However, better correction to compensate for the inflation of the variance could be performed by introducing pre-conditioning or reducing the injection of Gaussian noise in the algorithm. Based on these MCMC diagnostics, all algorithms have reached the steady-state with good sample quality as indicated by a KSD convergence rate of $\mathcal{O}(1/\sqrt{N})$.

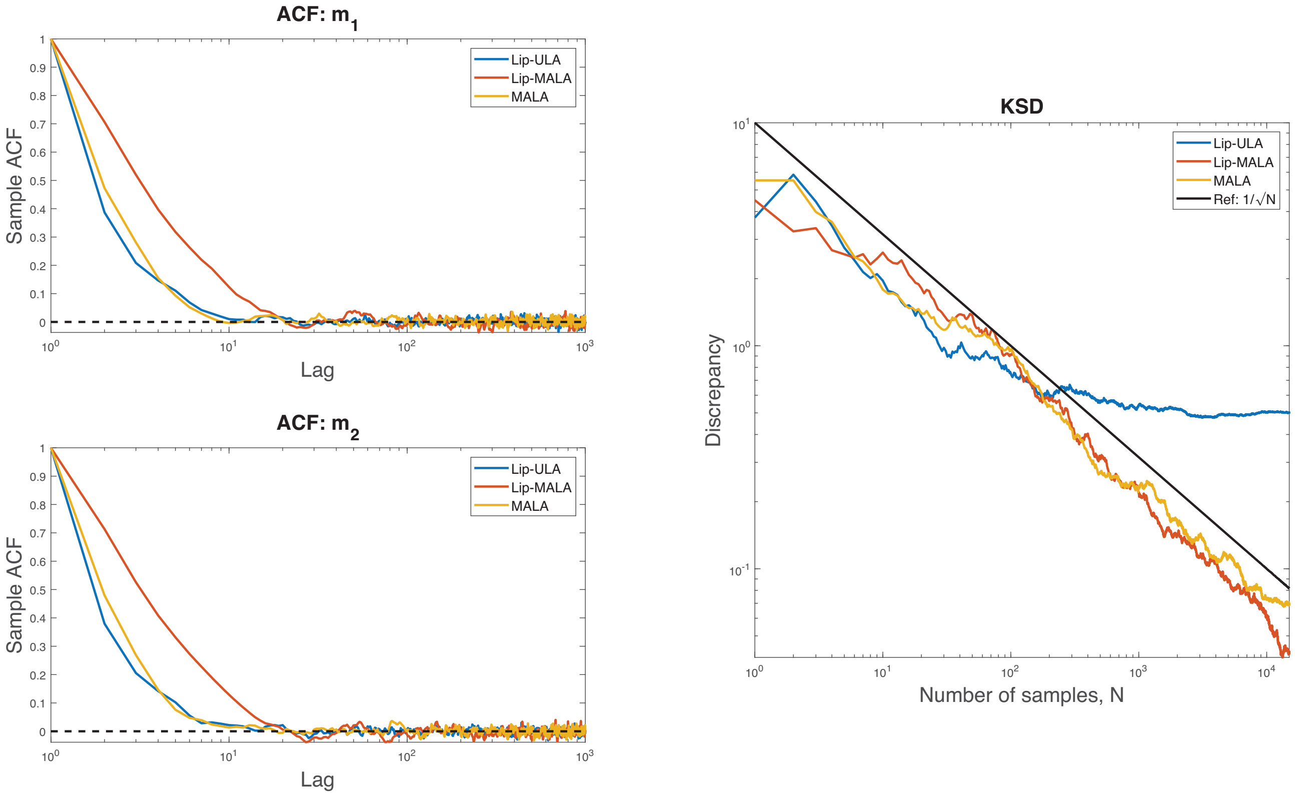

Figure 4. Left-hand panels: ACF of each bivariate Gaussian density parameter, $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$, respectively. Right-hand panel: KSD in log–log scale for samples from three different samplers: Lip-ULA, Lip-MALA and MALA.

Table 3. Summary of the statistical results for Numerical Example 2 with bivariate Rosenbrock density.

Methods	Mean, $(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2)$	Variance, $(\boldsymbol{\sigma}_1^2, \boldsymbol{\sigma}_2^2)$	Acceptance rate, per cent
Truth	(0.2500, 0.4005)	(0.3380, 0.2703)	-
Lip-ULA	(0.2527, 0.4652)	(0.4213, 0.3040)	100.00 per cent
Lip-MALA	(0.2383, 0.4270)	(0.3607, 0.2469)	58.24 per cent
MALA	(0.2785, 0.4218)	(0.3399, 0.2526)	58.38 per cent

5.2 Bivariate Rosenbrock density

In this numerical example, we consider a bivariate Rosenbrock density given by

$$\pi(\boldsymbol{m}) \propto \exp\left(\alpha(\boldsymbol{m}_1^2 - \boldsymbol{m}_2)^2 + (\boldsymbol{m}_1 - \beta)^4\right), \tag{10}$$

with $\alpha = 10$ and $\beta = 0.25$. Similar to its counterpart in the optimization literature, the bivariate Rosenbrock density is non-linear with respect to its model parameters $\boldsymbol{m}$. Here, we use a step size $\tau = 3.61 \times 10^{-2}$. Similarly, we consider no pre-conditioning and set $\boldsymbol{\Sigma} = \boldsymbol{I}_{2\times2}$, similar to the previous example. The sampling of the posterior starts from $\boldsymbol{m}_0 = [0, 0]^T$ for $N = 30\,000$ samples, again discarding the first half from the total number of samples as burn-in. We tabulate the statistical results for this example in Table 3.

Sampling performed by all three algorithms from the bivariate Rosenbrock density in eq. (10) is illustrated in Fig. 5. We observe that the samples from all algorithms are distributed accordingly within the support of the target density. Based on results recorded in Table 3, the algorithms satisfactorily approximate the true mean, although a slight deviation is caused by the non-linearity of the problem. We observed a similar variance result in the previous example, where Lip-ULA records higher variances than the other two algorithms. Being an approximate MCMC algorithm, the injected Gaussian noise in Lip-ULA drives the samples to extensively explore the low probability region, thereby inflating the variance.

To assess sample quality, we again perform MCMC diagnostics and show the individual trace plots for each parameter in the bivariate Rosenbrock density in Fig. 6. We find that the algorithms are mixing well and have reached the stationary region of target density $\boldsymbol{\pi}(\boldsymbol{m})$ with little sparsity in the chain. Additionally, the ACF plots are shown on the left of Fig. 7. Based on these ACF curves, we generally observe that all algorithms show slightly large autocorrelation with short lags; however, the ACF slowly stabilizes after ∼200 lags. We also observe that Lip-ULA shows slightly faster-decaying ACFs than the exact MCMC algorithms.

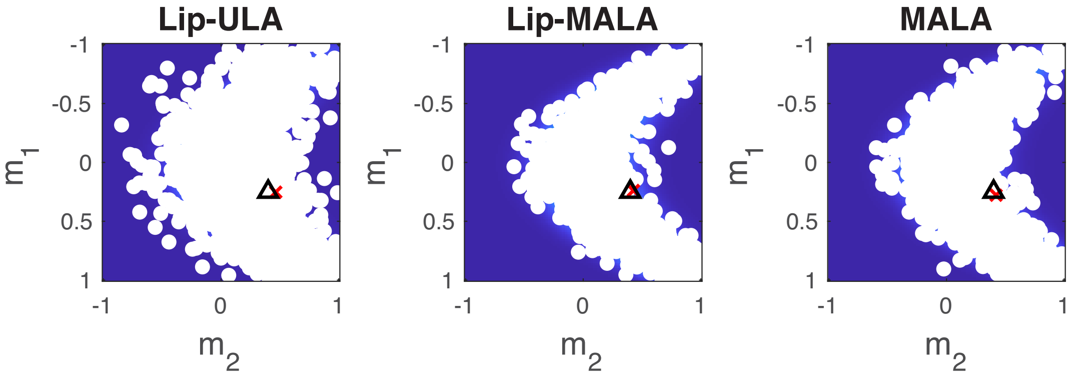

Figure 5. Samples drawn from the bivariate Rosenbrock density are plotted for three different Langevin dynamics MCMC algorithms: (a) Lip-ULA, (b) Lip-MALA and (c) MALA. The black triangle in each figure represents the true mean, $\boldsymbol{\mu}$, whereas the red cross represents the sample mean, $\overline{\boldsymbol{m}}$.

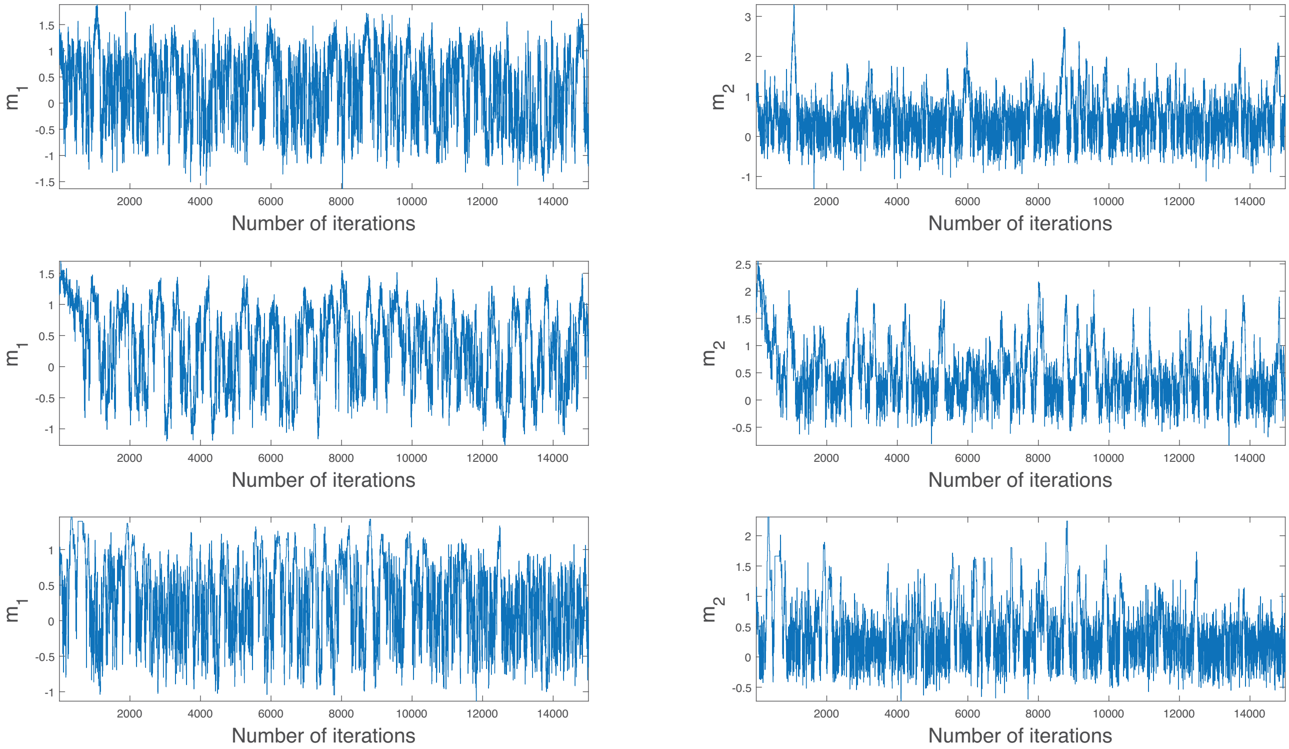

Figure 6. Trace plots for each parameter in the bivariate Rosenbrock density, $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$, respectively. First row: Lip-ULA; second row: Lip-MALA and third row: MALA.

The associated KSD values are plotted on the right of Fig. 7. For this example, MALA exhibits larger KSD fluctuations, which we attribute to the non-linearity of the problem. However, both Lip-MALA and MALA show convergence in KSD at the rate of $\mathcal{O}(1/\sqrt{N})$ as the number of samples increases. For Lip-ULA, we observe convergence in KSD from the early period of sampling until about $N = 1000$. However, as the number of samples increases, Lip-ULA reaches a plateau owing to the influence of high variance, which is typical for an approximate MCMC algorithm.

Based on these MCMC diagnostics, large sample sizes might be necessary for all algorithms to achieve low ACF values and well-mixed samples in the context of the non-linear bivariate Rosenbrock density. However, this may not compensate for the variance's inflation in Lip-ULA. Alternatively, one may propose introducing pre-conditioners into the algorithms; by pre-conditioning the algorithms, we may attain an accelerated MCMC convergence and mixing rate, particularly in non-linear problems.

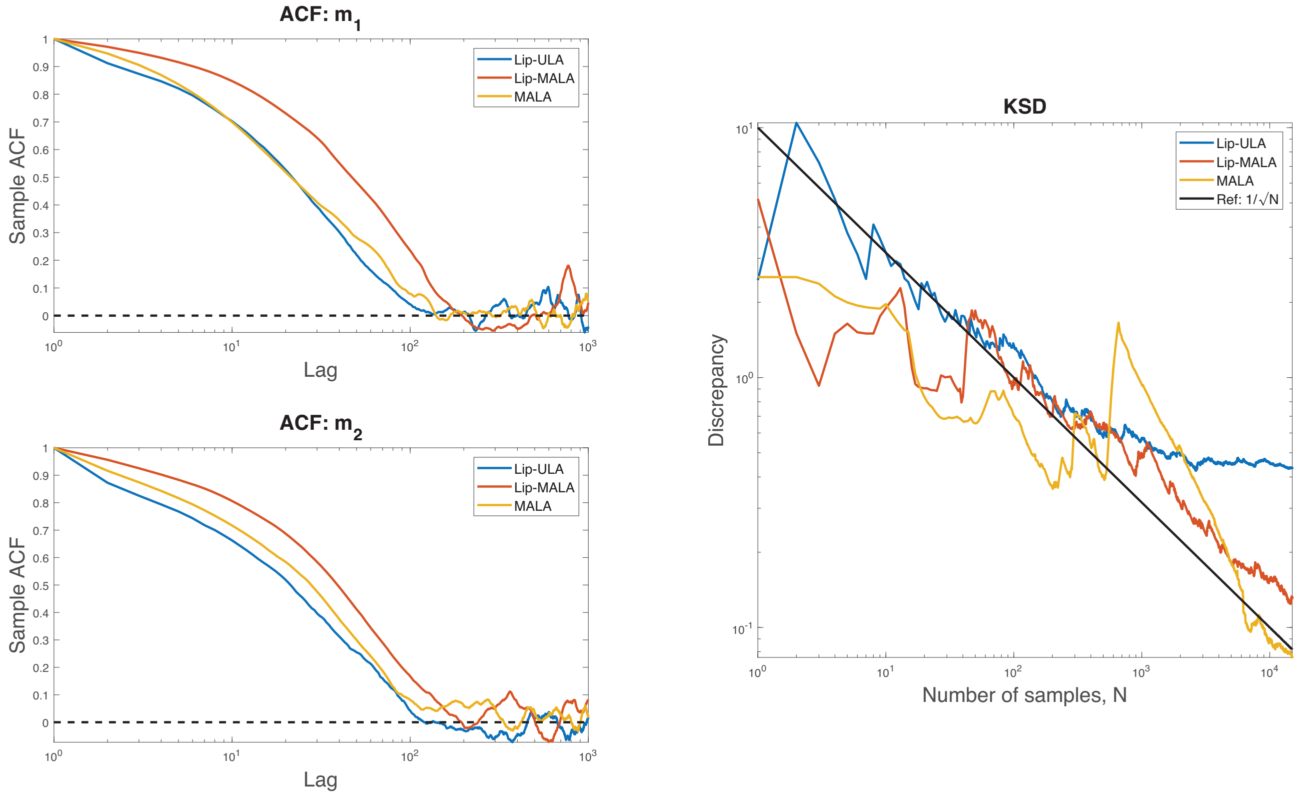

Figure 7. Left-hand panels: ACF of each parameter in the bivariate Rosenbrock density, $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$, respectively. Right-hand panel: KSD in log–log scale for the samples from three different samplers: Lip-ULA, Lip-MALA and MALA.

5.3 Low-dimensional model space FWI: the Camembert model

In this example, we consider a seismic FWI problem in the frequency domain with the following posterior density,

$$\pi(\boldsymbol{m}|\boldsymbol{D}) \propto \exp\left(-\frac{1}{2}||\boldsymbol{F}(\boldsymbol{m}) - \boldsymbol{D}||^2_{\boldsymbol{C_D}}\right)\pi(\boldsymbol{m}), \tag{11}$$

in low-dimensional model space. Similar to the previous example, the posterior density in eq. (11) admits a non-Gaussian distribution owing to the non-linearity of the forward modelling operator. Here, $\boldsymbol{F}(\boldsymbol{m})$ is a non-linear forward operator that maps the velocity model $\boldsymbol{m} \in \mathbb{R}^d$ onto the observed seismic data $\boldsymbol{D} \in \mathbb{R}^l$. The forward operator $\boldsymbol{F}(\boldsymbol{m})$ is strongly non-linear with respect to the model parameters $\boldsymbol{m}$; for this numerical example, we consider the velocity of the Camembert model with a domain size of 1000 × 1000 m as shown in Fig. 8(a). We discretize the model with a grid spacing of 20 m, yielding 2601 unknown parameters.

In this numerical example, we mimic a transmission cross-well experiment. We place 5 sources and 10 receivers at either side of the model with vertical sampling interval of 200 and 100 m, respectively, as displayed in Fig. 8(c). The signal-to-noise ratio in the data is 0.039 dB, and the relative standard deviation of the observation noise is 5 per cent. We use a frequency content from 3 to 12 Hz with a uniform frequency sampling of 3 Hz. All frequencies are used simultaneously in the sampling procedure, and no multiscale strategy is applied.

We set the data error covariance matrix $\boldsymbol{C_D} = \sigma_D^2 \boldsymbol{I_D}$ with $\sigma_D = 0.002$ and $\boldsymbol{I_D}$ being the identity matrix. For the model's prior $\boldsymbol{\pi}(\boldsymbol{m})$, we use uniform distributions with lower and upper bounds of 2.0 and 2.25 km s^{-1}, thereby encompassing the true velocity model. We started sampling with an initial model $\boldsymbol{m}_0$ as illustrated in in Fig. 8(b). Here, we set the step size to $\tau = 2 \times 10^{-6}$ with no pre-conditioning. We run the MCMC algorithms for $N = 200\,000$ iterations and consider the first 10 000 samples as the burn-in period. The simulations take approximately 3 d for each algorithm to complete when run in series on an Intel Xeon CPU E5-2680 v4 workstation. In this problem, Lip-ULA accepts the sample in each iteration with probability 1, whereas Lip-MALA and MALA accept the samples with an acceptance rate of 67.354 per cent, respectively.

We perform MCMC diagnostics to evaluate the sample quality and measure the algorithm's performance. We show the trace plots in Fig. 9 for three neighbouring model parameters, m_{229}, m_{1249} and m_{2269}. All respective chains reach the stationary region of the target probability density. This indicates that the chains have extensively explored the sample space, according to the prior information provided. We also evaluate the ACF for the same model parameters to complement the trace plots shown in the left-hand column of Fig. 10. Lip-ULA records low sample ACF values compared to Lip-MALA and MALA, which only drop to zero after ∼1000 lags. This indicates that Lip-ULA has high mixing rate in this setting. The right-hand column of Fig. 10 displays the computed KSD values. Referring to Izzatullah *et al.* (2020a), we thinned the samples to $N = 1 \times 10^3$ to reduce the computational costs, and the KSD converged for all algorithms. We also observe

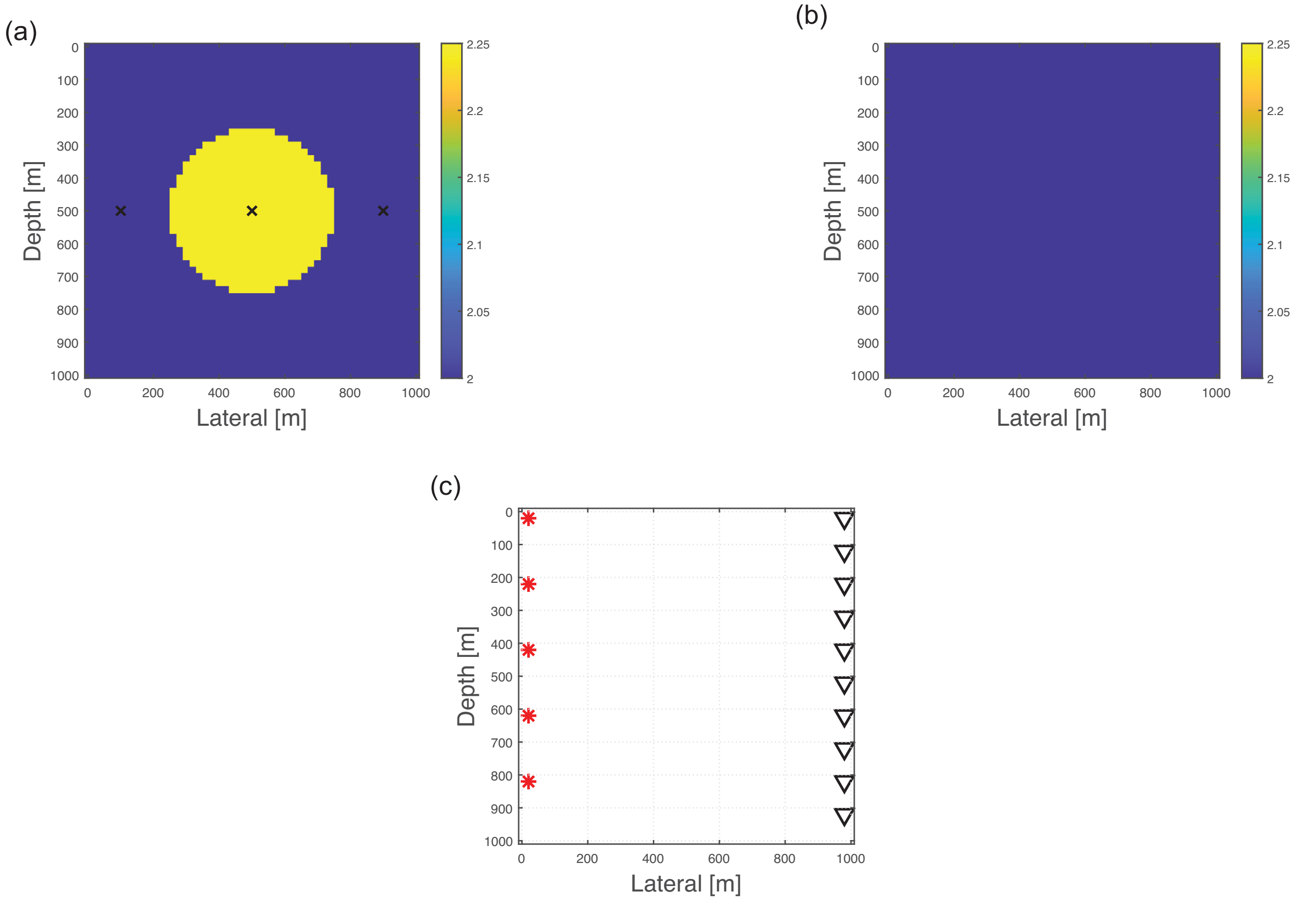

Figure 8. The Camembert model: (a) true model with the black crosses representing the chosen model parameters for MCMC diagnostics, namely the 229th, 1249th and 2269th parameters; (b) the initial model and (c) the model domain with sources (red stars) and receivers (black triangles).

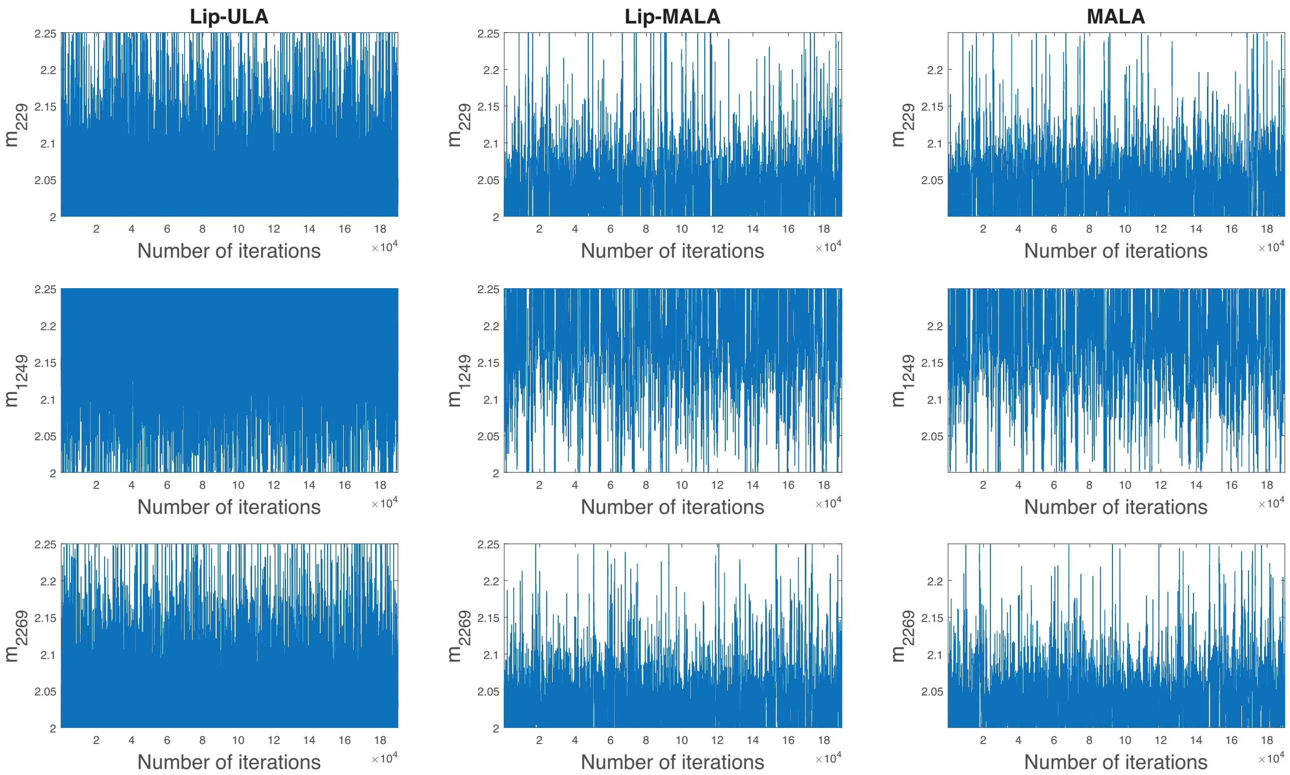

Figure 9. The Camembert model: trace plots for all three algorithms from the selected model parameters. First row: m_{229}; second row: m_{1249} and third row: m_{2269}.

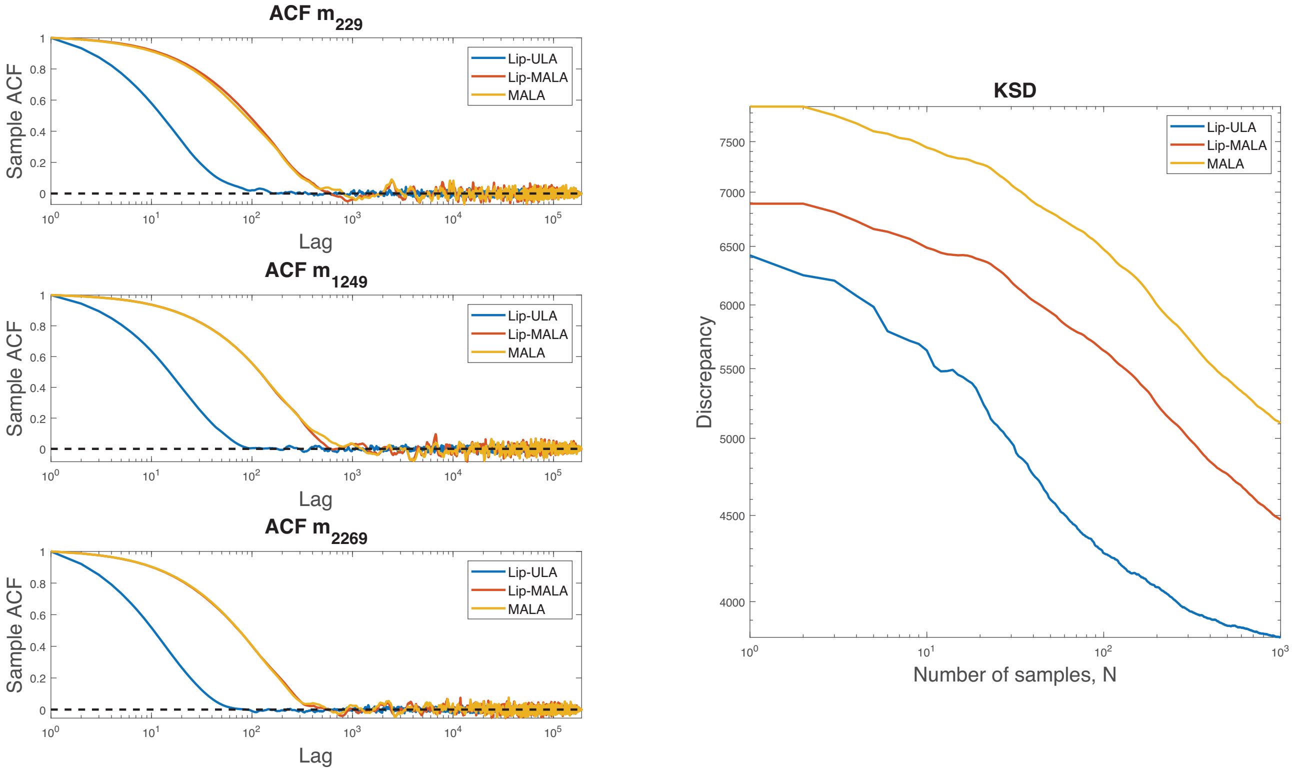

Figure 10. The Camembert model: Left-hand side: the ACF for all three algorithms from the selected model parameters, m_{229}, m_{1249} and m_{2269}, respectively. Right-hand side: the KSD in log–log scale for samples from three different samplers: Lip-ULA, Lip-MALA and MALA.

that the KSD for Lip-ULA starts to reach a plateau because of the influence of high variance, which is typical for an approximate MCMC algorithm, as discussed earlier. Overall, these diagnostics indicate that all chains have reached the stationary and high-posterior probability regions of the target probability density.

We continue the statistical analysis by displaying the sample mean, variance, and skewness models for all three algorithms in Fig. 11. Overall, the sample mean models for all algorithms are close to each other and resemble the true model after 200 000 MCMC iterations. This indicates that we have good data coverage and good illumination from the source–receiver geometry; additionally, this indicates that all algorithms have reached the high-posterior probability region. The sample variance models overall suggest small variations in the model parameters ($\sim$0.0 to -6×10^{-3} km s^{-1}). We observe high variations around the Camembert body and at the corners of the velocity model. Notably, owing to the influence of biasedness in Lip-ULA, its variance model is more pronounced than that of the others; in general, all sample variance models are close to each other, and we might suggest leveraging the qualitative–quantitative trade-off in interpreting the statistical results. However, this suggestion is outside of the scope of this paper.

The information from the sample mean and variance models alone are insufficient for characterizing the posterior distribution, particularly in the context of FWI; thus, we rely upon the skewness models to characterize the non-Gaussianity. We observe non-zero values for the model's skewness for many model parameters and records are highly skewed in regions with low variances. This indicates a non-Gaussian posterior. We speculate that the high skewness and non-Gaussian behaviours are contributed by wave-scattering and reflection events in those regions. However, further research is required to prove this speculation. We also visualize the 1-D and 2-D pairwise marginal posterior probability distributions for three neighbouring model parameters—m_{229}, m_{1249} and m_{2269}—for all three algorithms in Figs 12, 13 and 14, respectively. The marginals show that the algorithms explored unimodal posterior distributions with similar characteristics. Although the marginals show a unimodal distribution for at least three locations, the posterior is non-Gaussian owing to the non-linearity of the problem. In this context, an uncertainty analysis based on a Gaussian or a Laplace approximation might not be an alternative worth pursuing and it may have limited meaning in the FWI context. Fig. 1 explains this situation well.

5.4 High-dimensional model space: the Marmousi model

Using this numerical example, we demonstrate the proposed algorithm's capabilities for sampling a seismic FWI problem in a high-dimensional model space. We consider a posterior density similar to that in eq. (11) for the Marmousi model with a domain size of 3000 × 11 000 m, as shown in Fig. 15(a). We discretize the velocity model with a grid spacing of 50 m, yielding 13 420 unknown parameters; at the surface, we place 55 sources and 110 receivers with horizontal sampling intervals of 100 and 50 m, respectively, as displayed in Fig. 15(c). The signal-to-noise ratio in the data is 0.059 dB, and the relative standard deviation of the observation noise is 5 per cent. We use a frequency

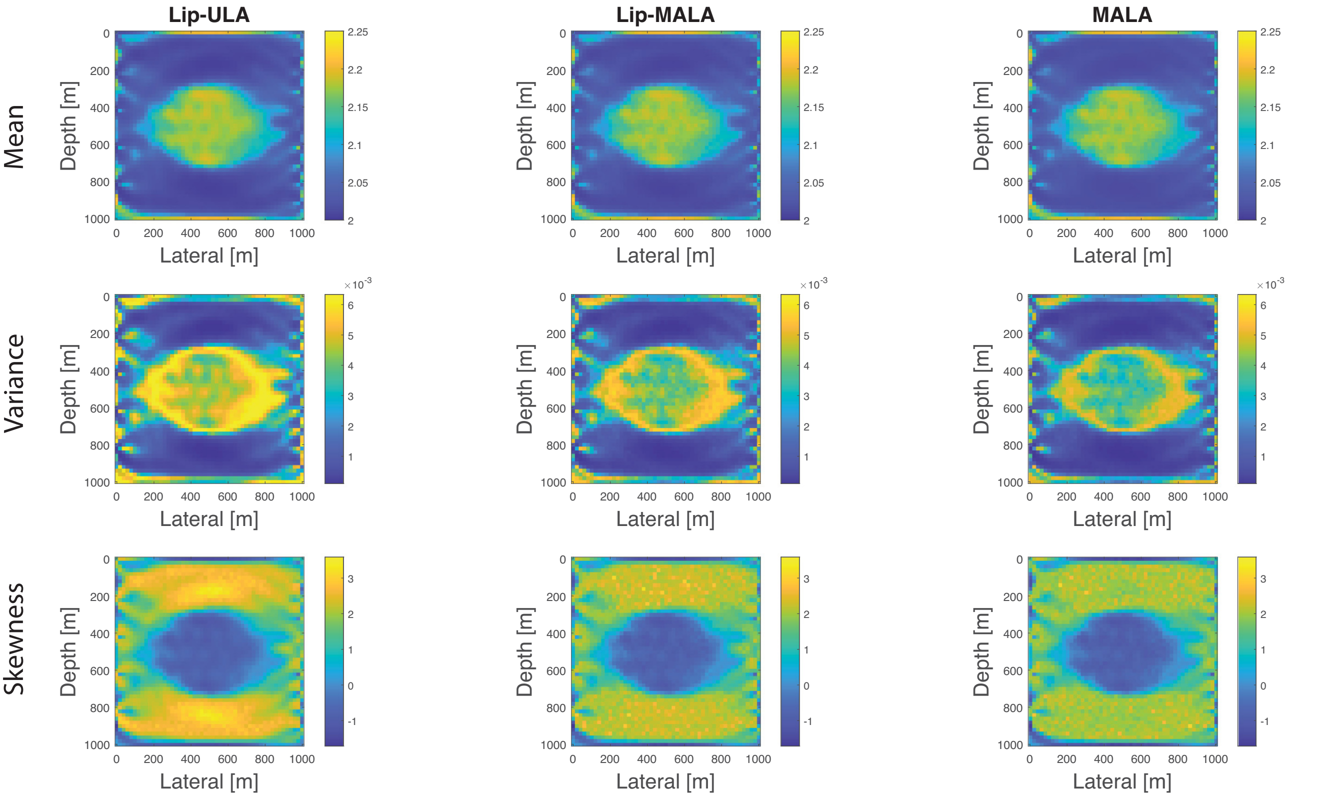

Figure 11. The Camembert model. Summary of the sample mean, variance, and skewness models of the posterior distribution for the three algorithms. First row: mean; second row: variance and third row: skewness.

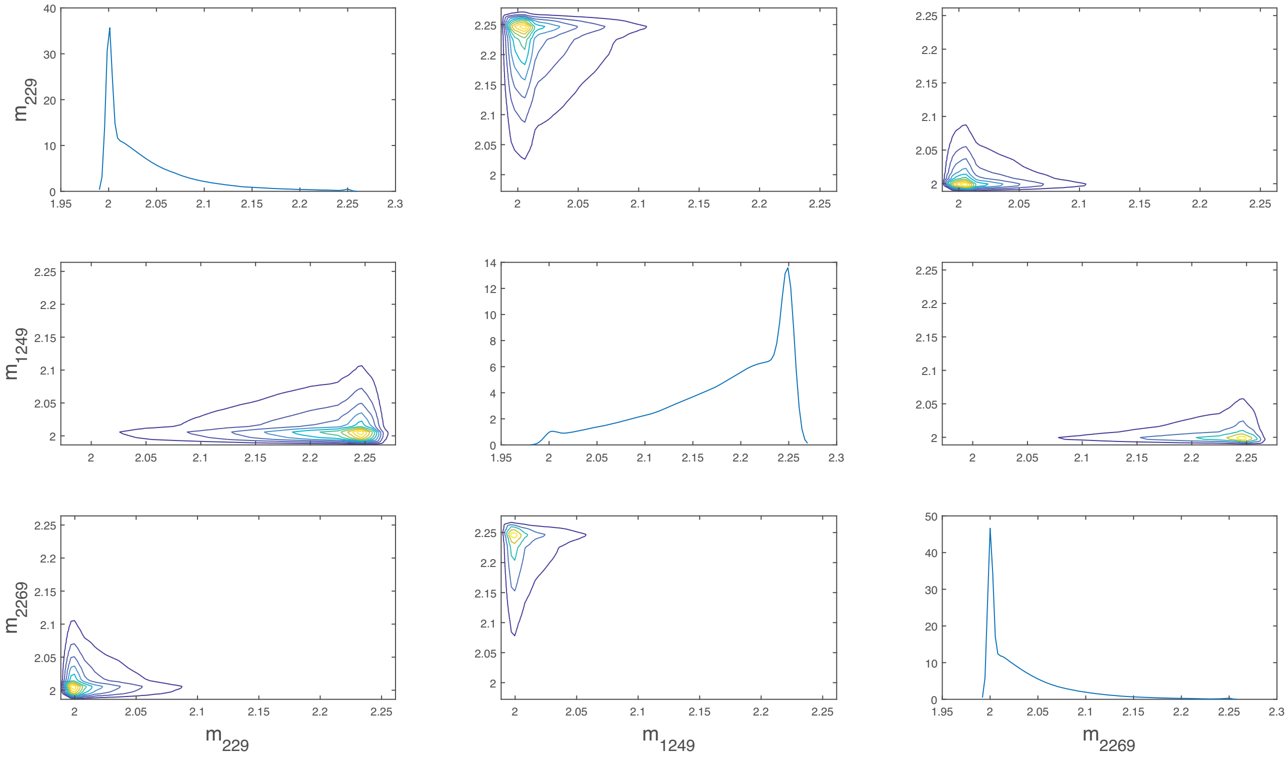

Figure 12. The Camembert model: The 1-D and 2-D pairwise marginal posterior probability distributions for three neighbouring model parameters, m_{229}, m_{1249} and m_{2269} for Lip-ULA.

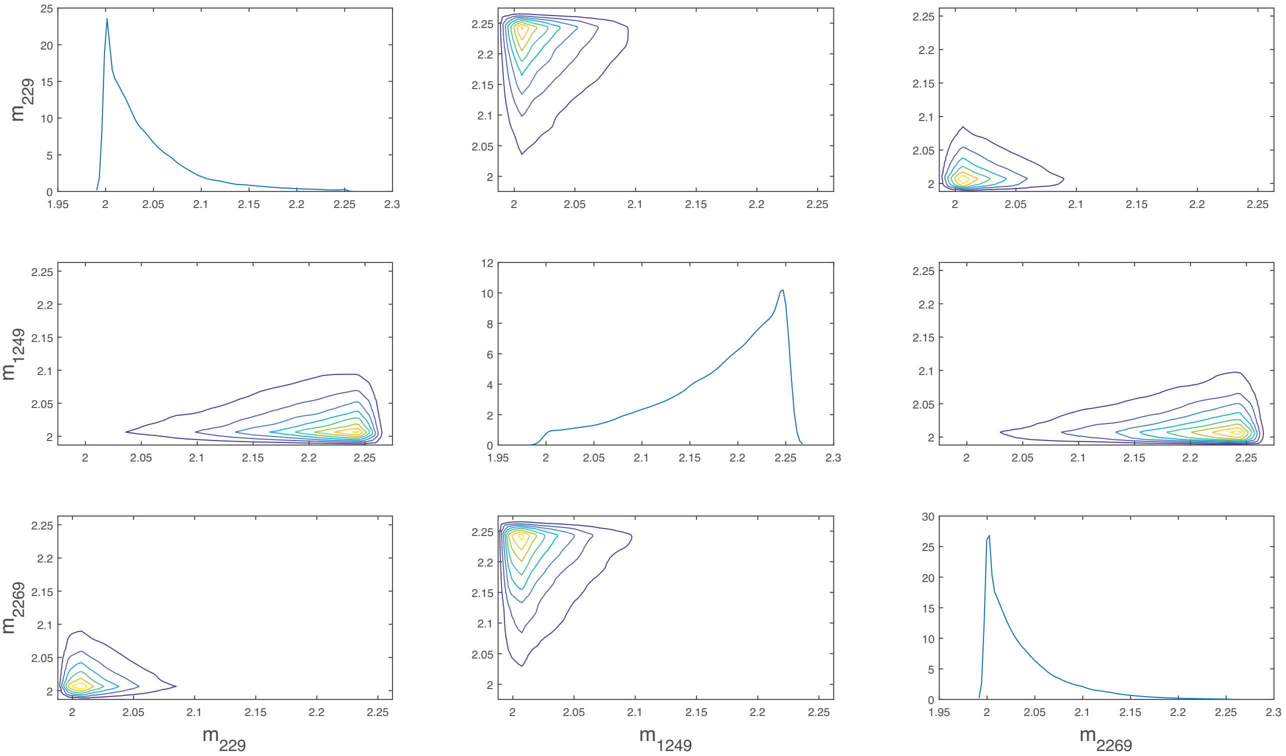

Figure 13. The Camembert model: The 1-D and 2-D pairwise marginal posterior probability distributions for three neighbouring model parameters, m_{229}, m_{1249} and m_{2269} for Lip-MALA.

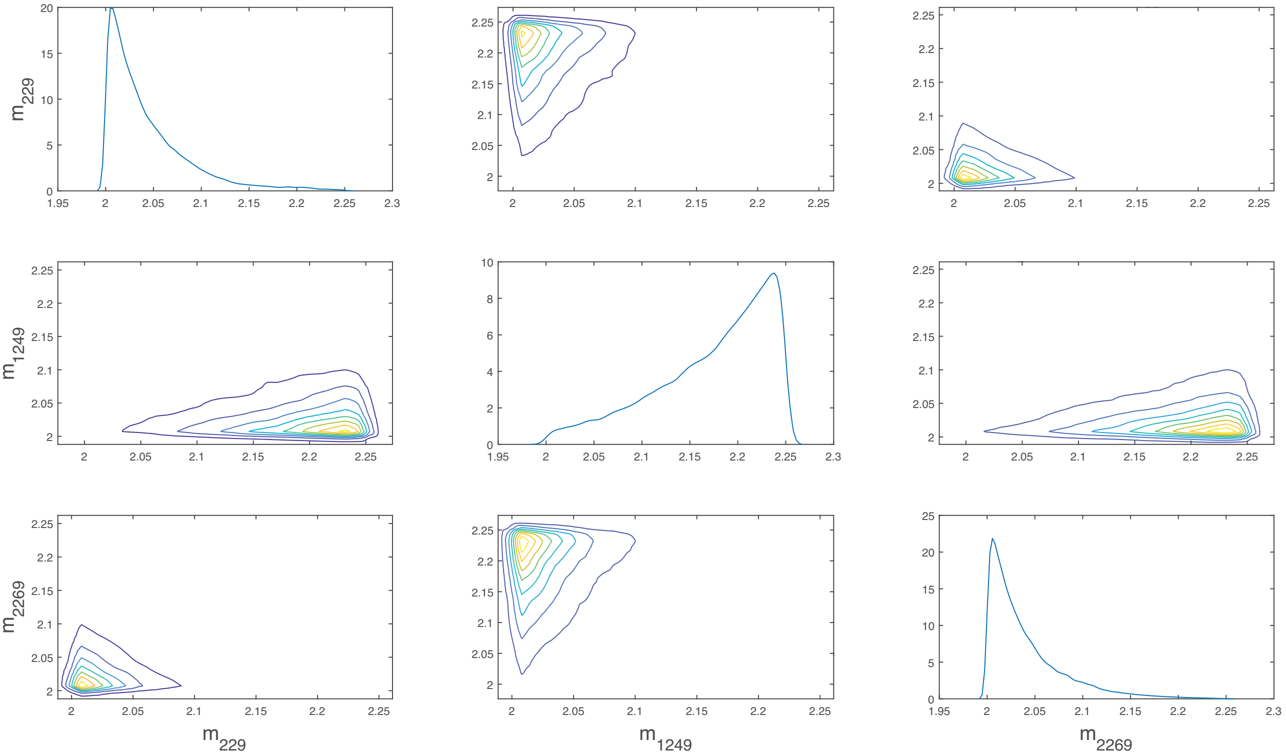

Figure 14. The Camembert model: The 1-D and 2-D pairwise marginal posterior probability distributions for three neighbouring model parameters, m_{229}, m_{1249} and m_{2269} for MALA.

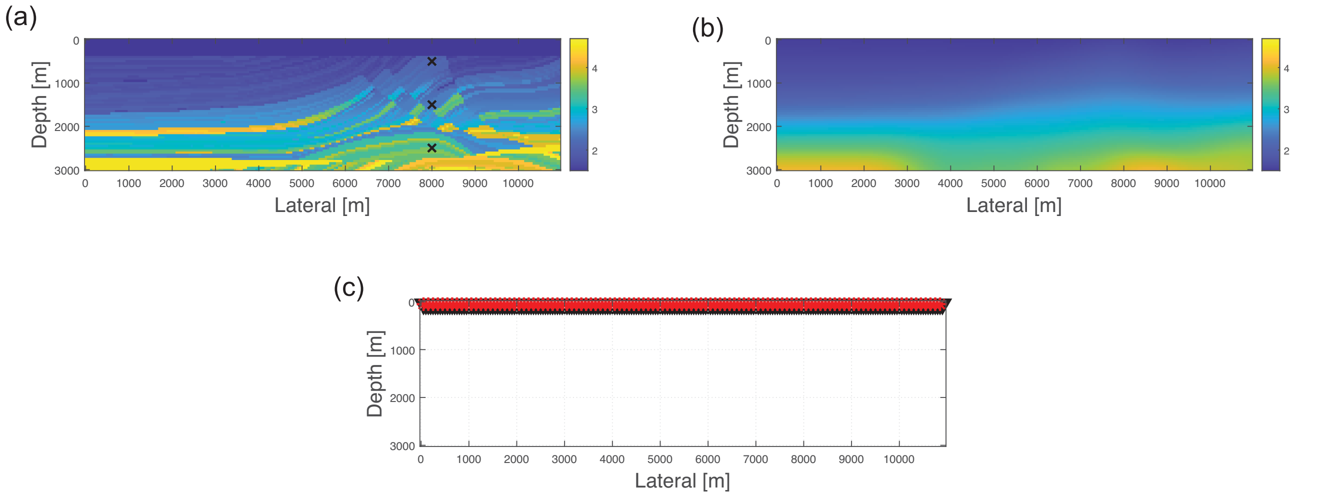

Figure 15. The Marmousi model. (a) True model with the black crosses representing the chosen model parameters for MCMC diagnostics (the 9709th, 9729th and 9749th parameters, respectively). (b) The initial model. (c) The model domain with sources (red stars) and receivers (black triangles).

content from 1 to 4 Hz with a uniform frequency sampling of 1 Hz; all frequencies are used simultaneously in the sampling procedure, no multiscale strategy is applied.

For this problem, we set the data error covariance matrix $\boldsymbol{C_D} = \sigma_D^2 \boldsymbol{I_D}$ with $\sigma_D = 0.05$ and $\boldsymbol{I_D}$ being the identity matrix. For the model's prior $\boldsymbol{\pi}(\boldsymbol{m})$, we use uniform distributions within certain bounds. The width of the prior reflects the minimum and maximum velocities of the Marmousi model. We started sampling with an initial model $\boldsymbol{m}_0$ obtained by smoothing the true model with a Gaussian kernel, as illustrated in in Fig. 15(b). We consider the relationship $\tau_{\mathrm{opt}} \propto d^{-1/3}$ as our guideline for setting the step size to $\tau = 1 \times 10^{-4}$.

We introduce the pseudo-Hessian matrix (Choi *et al.* 2007) as the pre-conditioner at each iteration to accelerate the MCMC convergence and mixing rate and overcome the computational challenges introduced by full-matrix pre-conditioning. The pseudo-Hessian matrix approximates the Hessian matrix's diagonal through the virtual sources used to compute partial-derivative wave fields, and it is calculated using a forward modelling operator; however, this ignores the correlation between parameters. In FWI, this is equivalent to assuming that changes in one location's velocity do not affect the velocity at other locations. In this context, the full covariance matrix is infeasible to compute as a pre-conditioner for the MCMC algorithms; explicitly computing the Hessian matrix of the negative log-posterior $-\log \boldsymbol{\pi}(\boldsymbol{m}|\boldsymbol{D})$, (i.e. the inverse of the covariance matrix) requires a forward-PDE problem for each of its columns, and thus, the same number of forward-PDE solves are required as the number of parameters. Martin *et al.* (2012) and Bui-Thanh *et al.* (2013) introduced a low-rank covariance approximation based on the Lanczos algorithm, which in the context of MCMC for a large-scale and high-dimensional data problem is still prohibitive to compute. Indeed, Bui-Thanh *et al.* (2013) implemented the mentioned algorithm for a large-scale global seismic inversion problem; however, their work focuses on Laplace's approximation for the MAP model instead of sampling the posterior density with MCMC algorithms. We continue this discussion on the choice of the pre-conditioner matrix in Appendix C based on our intuitive Gaussian examples.

We run the MCMC algorithms for $N = 60\,000$ samples: the first 5000 samples are considered as the burn-in period, and the following 55 000 samples are used for statistical analysis. The simulations took approximately 11 d to complete for each algorithm (run in serial on an Intel Xeon CPU E5-2680 v4 workstation). In this problem, Lip-ULA accepts the sample in each iteration with probability 1, whereas Lip-MALA and MALA accept the samples with acceptance rates of 50.64 and 46.45 per cent, respectively.

To further assess the results in this example, we perform MCMC diagnostics to evaluate the sample quality. The trace plots in Fig. 16 for three neighbouring model parameters, m_{9709}, m_{9729} and m_{9749}, exhibit slow mixing and high correlation for the first two parameters, indicating that the chains do not extensively explore the sample space compared to the third chain. We also evaluate the ACF for the same model parameters to complement the trace plots, as shown in the left column of Fig. 17. The sample ACF values for all algorithms at the respective model parameters drop to zero after ~10 000 lags. This is consistent with the trace plot diagnosis, which indicates that one needs more than 60 000 samples to reach the stationary region of the target probability density. The right column of Fig. 17 displays the computed KSD values. Similar to the previous example, we thinned the samples to $N = 1 \times 10^3$ to save computational costs. We observe that all algorithms have a relatively similar trends to KSD values and only start to converge at later iterations; this indicates that there are many serial correlations between successive draws and that the algorithms only explore a small region within the sample space (i.e. do not perform an extensive search). The information obtained from the KSD plots is consistent with the trace and ACF plots.

Fig. 18 displays the sample mean, variance and skewness models for all three algorithms. Although the chains have not formally reached the stationary region (at least not at all locations), we observe that the sample mean models for all algorithms are close to one another overall and that all models resemble the true model after 60 000 MCMC iterations. This indicates that all algorithms have reached the high-posterior probability regions. The sample variance models suggest small variations (~0.0−0.3 km s^{-1}) in the shallow regions of the model, particularly near the source–receiver level. This indicates that we have good data coverage and good illumination in the model's shallow regions. In contrast, the variances in the deeper region and at the corners are more pronounced (~0.8− 1.0 km s^{-1}). This is consistent

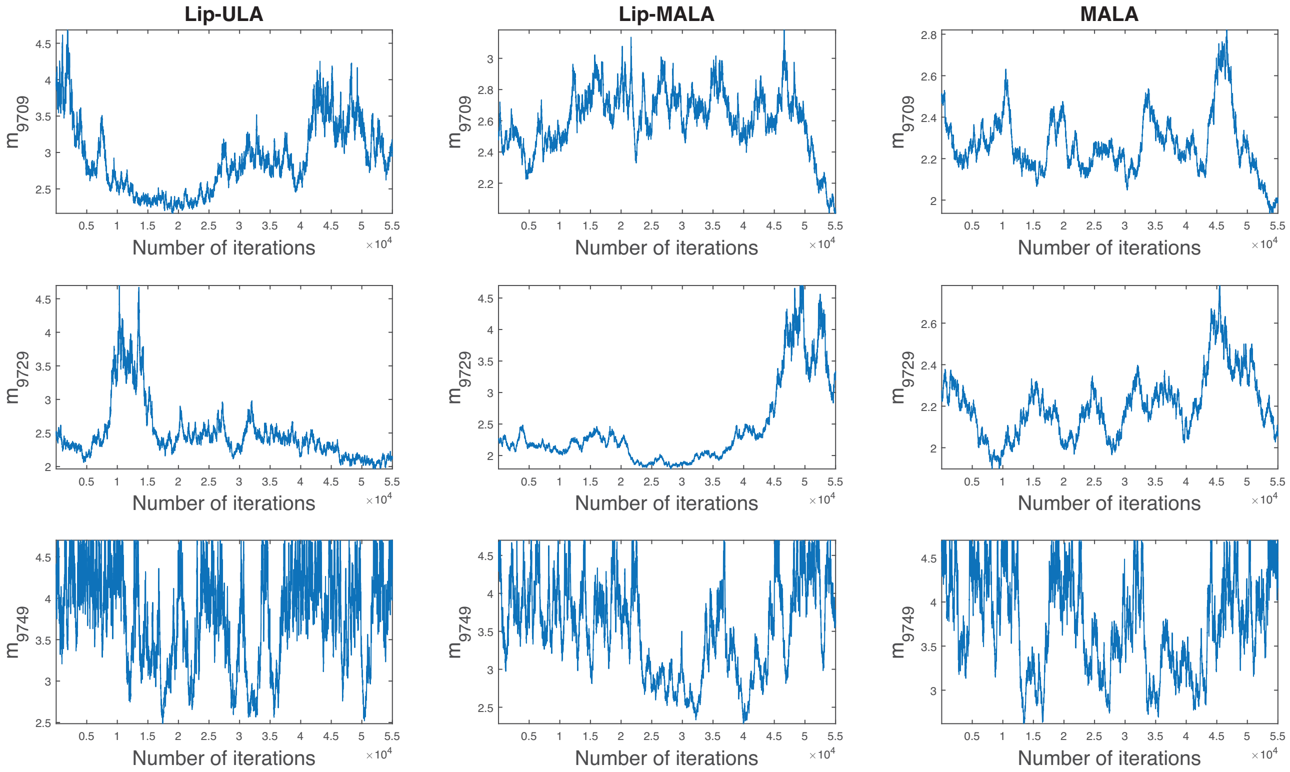

Figure 16. The Marmousi model. Trace plots for all three algorithms from the selected model parameters. First row: m_{9709}; second row: m_{9729} and third row: m_{9749}.

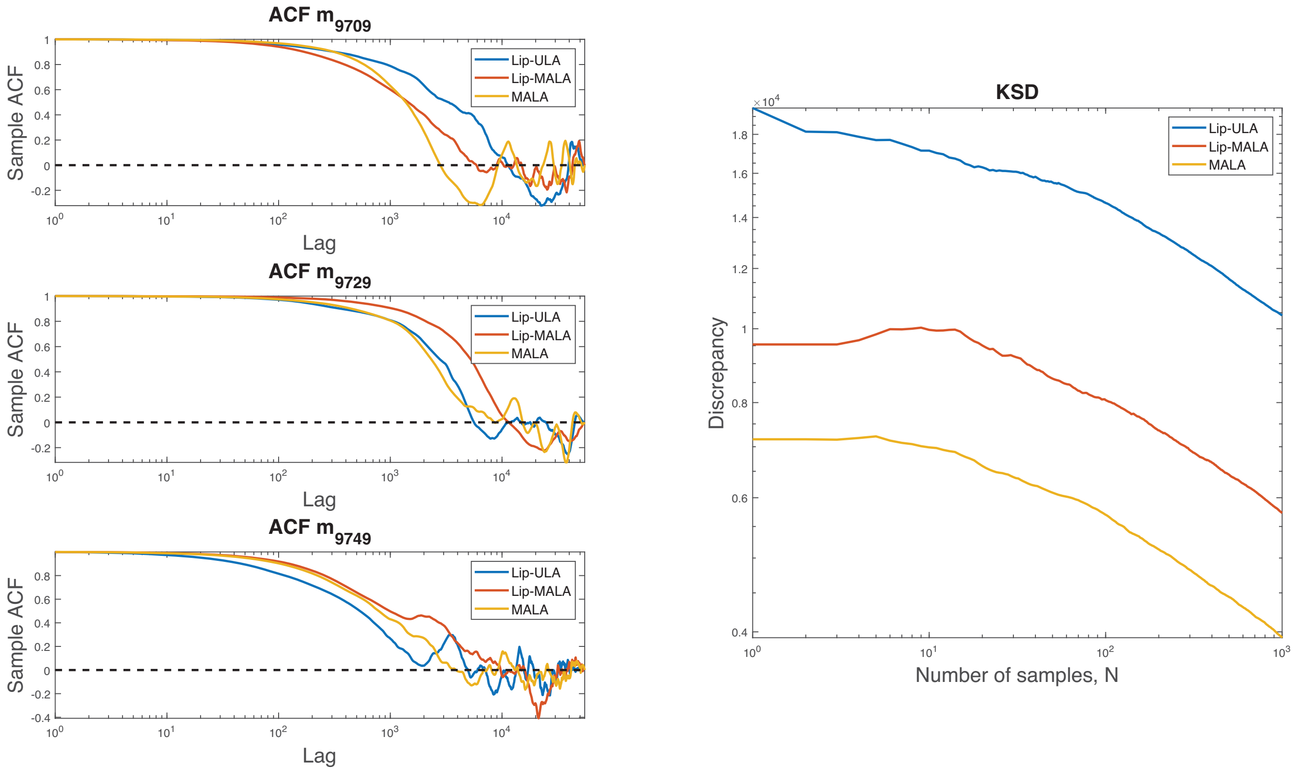

Figure 17. The Marmousi model. Left-hand panels: the ACF for all three algorithms from the selected model parameters, m_{9709}, m_{9729} and m_{9749}, respectively. Right-hand panel: the KSD in log–log scale for the samples from three different samplers: Lip-ULA, Lip-MALA and MALA.

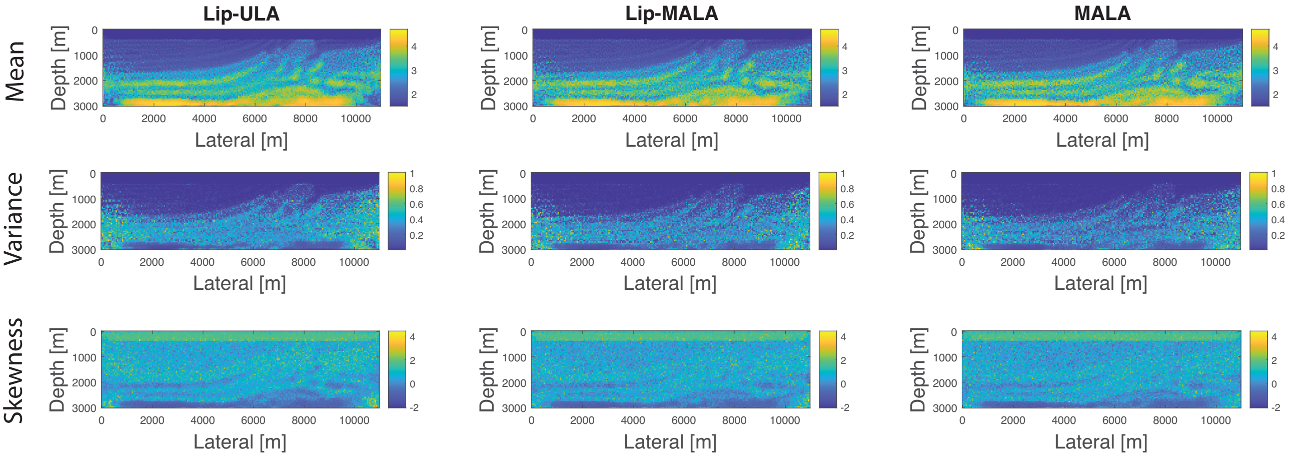

Figure 18. The Marmousi model. Summary of the posterior distribution, the sample mean, the variance, and the skewness models for three algorithms. First row: mean; second row: variance and third row: skewness.

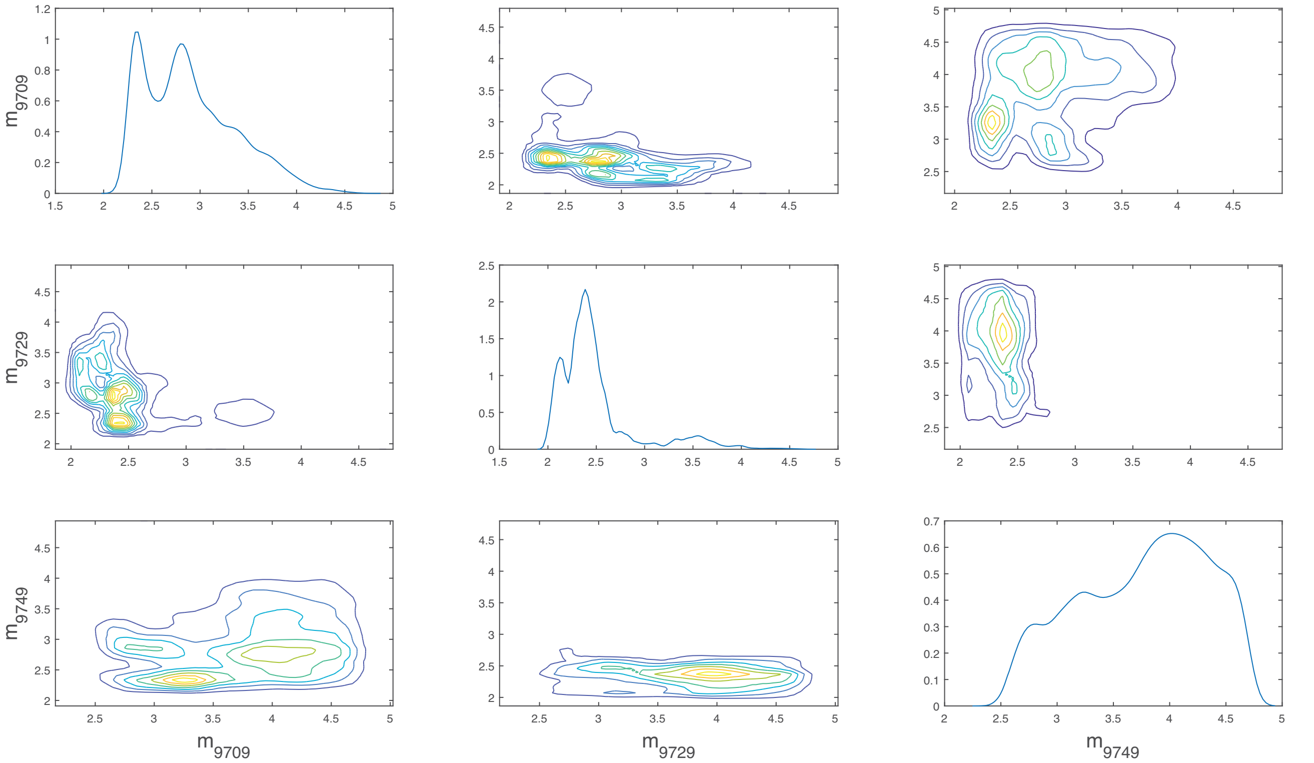

Figure 19. The Marmousi model. The 1-D and 2-D pairwise marginal posterior probability distributions for three neighbouring model parameters, m_{9709}, m_{9729} and m_{9749} for Lip-ULA.

in that the data are more informative in the shallow area of the model compared with the deep areas. Those regions are poorly illuminated, and the inferred velocities are spread out over a wide range of values. We observe non-zero values of skewness for many model parameters and find large skewness in the shallowest and deepest regions; this indicates a non-Gaussian posterior distribution and immediately suggests that the Gaussian/Laplace approximation for uncertainty quantification in FWI may be insufficient. In addition, we visualize the 1-D and 2-D pairwise marginal posterior probability distributions for three neighbouring model parameters, m_{9709}, m_{9729} and m_{9749} for all three algorithms in Figs 19, 20 and 21, respectively. The marginals show that the algorithms were exploring multimodal posterior distributions, indicating significantly non-Gaussian posterior distribution as a consequence of non-linearity; this, again, highlights that uncertainty analysis based on a Gaussian/Laplace approximation may have limited meaning in the context of FWI.

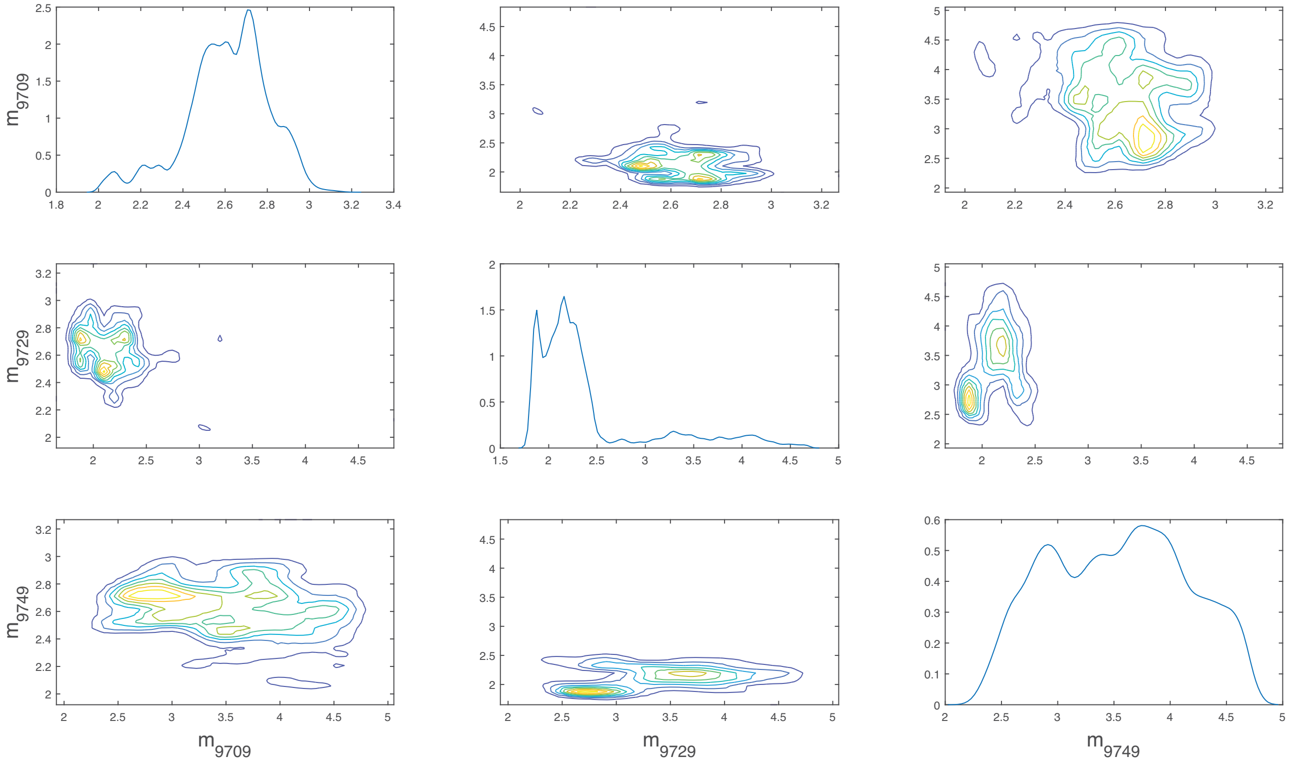

Figure 20. The Marmousi model. The 1-D and 2-D pairwise marginal posterior probability distributions for three neighbouring model parameters, m_{9709}, m_{9729} and m_{9749} for Lip-MALA.

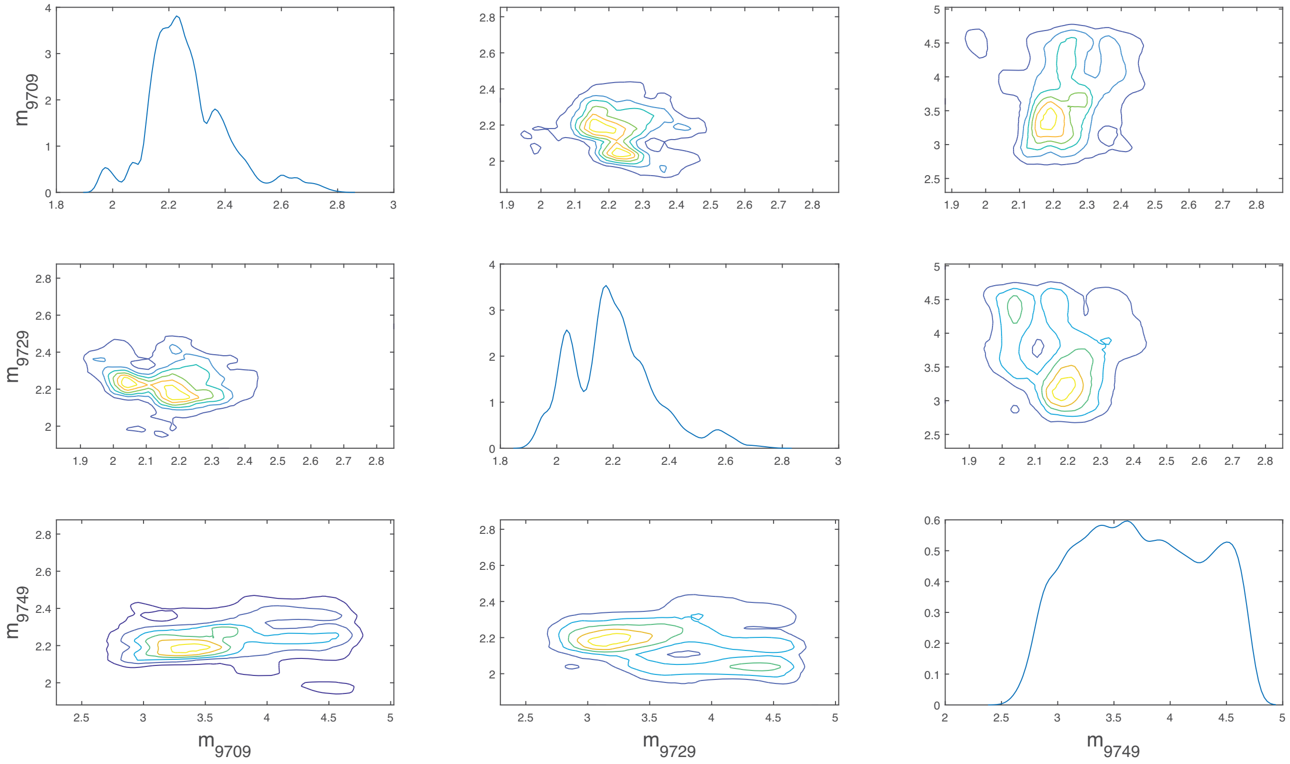

Figure 21. The Marmousi model. The 1-D and 2-D pairwise marginal posterior probability distributions for three neighbouring model parameters, m_{9709}, m_{9729} and m_{9749} for MALA.

6 DISCUSSION

Herein, we investigated different LMC algorithms and introduced a locally Lipschitz adaptive step size modification based on the local smoothness of the probability log-density. We highlighted the potential and limitations of the proposed algorithms (i.e. Lip-ULA and Lip-MALA) in comparison with MALA for various numerical examples and presented a qualitative and quantitative analysis of the sample quality based on KSD, ACF and trace plot evaluations.

In practice, for Bayesian FWI, we require extensive computational resources to perform numerous MCMC iterations. We observe in the numerical examples that all algorithms recovered the important subsurface structures similar to the true model. However, this information alone does not accurately represent the important aspects of the posterior distribution. Based on the results shown, one may consider Lip-ULA or a similar approximate MCMC algorithm to efficiently sample the posterior target density at high sampling speed with computational saving; notably, this approach shown an improvement in the computational cost of about 42.6 per cent from the optimal acceptance rate 57.4 per cent of MALA, by accepting all the samples in each iteration with probability one. Regarding the inflation of second statistical moment owing to the asymptotic biasedness, we recommend that geophysicists leverage the qualitative–quantitative trade-off in interpreting statistical results. With larger computational budgets, one should favour exact methods such as MALA and HMC over approximate LMC; as the computing budget increases, the accuracy of the exact MCMC methods increases. At this stage, we illustrated the LMC algorithms' potentials for performing Bayesian FWI; the presented results can be considered a feasibility study for further improvements.

In the following subsections, we discuss several issues concerning the Langevin dynamics MCMC algorithms.

6.1 Computational aspects

For a large-scale inference problem with high-dimensional data (e.g. Bayesian FWI), at least four main factors exist that can cause computational bottlenecks in Langevin dynamics MCMC algorithms:

(1)PDE forward solutions.
(2)PDE adjoint solutions (for computing the gradient of a log-posterior density).
(3)Matrix–vector multiplications in the presence of a pre-conditioner $\mathbf{\Sigma}$.
(4)The choice of prior distribution.

These factors can incur considerable computational costs; for example in example 5.4 simulations took approximately 11 d to completely execute a single chain LMC algorithm. At each iteration, we need to solve one PDE forward and one adjoint problem, which scales with the dimensions of the model space, and to some extent, with the number of observed seismic data points. These computational costs quickly become prohibitive as model dimensions and data size increase and as the Metropolis–Hastings rejection rate increases. However, promising advances have been made in reducing computational costs, such as dimensional (Cui *et al.* 2014, 2016; Zahm *et al.* 2018) and model-order (Borcea *et al.* 2019, 2020) reduction techniques. To avoid the third computational bottleneck, we use the pseudo-Hessian (Choi *et al.* 2007) as a pre-conditioner providing a numerical advantage by using vector–vector instead of matrix–vector multiplication. The choice of a prior distribution also has a significant impact on the algorithm's performance. A poorly chosen prior distribution or too general or less-significant prior information can critically increase the computational effort.

Furthermore, for the exact MCMC algorithms (i.e. Lip-MALA and MALA), computational costs increase if the sample rejection rate is high. Commonly, this is due to a poor choice of prior information, pre-conditioning matrix $\mathbf{\Sigma}$ and step size τ. In such situations, we observe that Lip-ULA can be more efficient when all samples are accepted with probability 1, effectively using computational resources. However, as we have mentioned repeatedly, one should favour exact methods such as MALA and HMC over approximate Langevin dynamics MCMCs for superior accuracy if larger computational budgets are available.

6.2 Algorithmic bias of ULA

Owing to its asymptotic bias, ULA is regarded as unreliable, meaning that it may converge to a limit different from its target density π. This bias is typically attributed to the discretization error in the continuous-time Langevin dynamics. The standard proposal for correcting it is to introduce the Metropolis–Hastings correction step (Roberts & Tweedie 1996; Dwivedi *et al.* 2018); however, this introduces additional complexity into the algorithm and increases the requirement of computational resources, as discussed previously.

Nevertheless, ULA has been shown to perform well theoretically and practically for high-dimensional Bayesian inference (Dalalyan 2017; Dalalyan & Karagulyan 2017; Durmus & Moulines 2017; Durmus *et al.* 2019; Durmus & Moulines 2019; Dalalyan & Riou-Durand 2020). Moreover, Wibisono (2018) showed that one can reduce the bias at the price of implementing a proximal gradient step. In general, this is an open problem subject to active research by the artificial intelligence and statistics communities (Dalalyan & Karagulyan 2019; Dalalyan & Riou-Durand 2020; Nemeth & Fearnhead 2020).

6.3 Adaptive step-size hyperparameters

In Algorithms 1 and 2, we introduced the locally Lipschitz adaptive step size to automate step size tuning; however, to obtain optimal performance, these algorithms depend on three hyperparameters: (i) the initial step size τ_0, (ii) the constant factor L_C and (iii) the pre-conditioning matrix $\mathbf{\Sigma}$. Based on our numerical examples, we find that the algorithms are insensitive to the initial step size τ_0. We used a relatively high initial step size τ_0 and still obtained satisfactory results comparable to MALA with an optimal step size τ.

For the constant factor L_C, we consider $L_C = d^{-1/3}$ according to $\tau_{\text{opt}} \propto d^{-1/3}$, the optimal scaling of MALA (Roberts & Rosenthal 1998), where d is the dimension of the model parameters. Herein, we did not further explore the sensitivity toward L_C because it is outside of the scope of this paper; however, it may affect the performance of the algorithms, as it did for optimization algorithms. In future work, we plan to explore this dependence on the constant factor L_C, for example by using neural networks (Sun & Alkhalifah 2019) to improve the performance of our current algorithms by training the recurrent neural network (RNN) using the history information of the iterates and the gradient of the misfit function to learn the best L_C.

In the case of a pre-conditioning matrix $\mathbf{\Sigma}$, we observed that our algorithms, and MALA are sensitive to the choice of a pre-conditioner, as demonstrated in example 5.4 and Appendix C. Moreover, Hamiltonian MCMC algorithms suffer from a similar situation as reported in Fichtner *et al.* (2019). Thus, finding and choosing an appropriate pre-conditioning matrix $\mathbf{\Sigma}$ provides an exciting research area in the context of Langevin and Hamiltonian dynamics MCMC algorithms; however, this pre-conditioning matrix $\mathbf{\Sigma}$ needs to be computationally feasible and highly informative in providing proper trajectories for efficiently exploring the posterior density.

6.4 MCMC diagnostics

In our numerical examples, we validated simulations through qualitative and quantitative evaluations of sample quality based on KSD, ACF and trace plots. These MCMC diagnostics are important because they provide insights on the convergence to stationary distribution and sample quality. We believe that performing these MCMC diagnostics and displaying the results is necessary. The statistical expectations alone (e.g. mean and variance, misfit plots and acceptance rates) are usually insufficient to study and display the MCMC algorithmic performances.

7 CONCLUSION

In this work, we focus on Langevin dynamics MCMC algorithms. By adding an adaptive step-size rule based on the local smoothness of the log-probability density to increase the sampling speed, we introduce Lip-MALA, a MALA extension as an exact MCMC algorithm (i.e. with a Metropolis–Hastings acceptance step). We also study the possibility of suppressing the computationally demanding Metropolis–Hastings acceptance step and proposed Lip-ULA. Lip-ULA belongs to the family of approximate MCMC algorithms, which are asymptotically biased owing to the discretization error.

We have compared the performance of the proposed algorithms with MALA with an optimal step size via several numerical examples. The proposed algorithms reliably recover important aspects of the posterior distribution, including means, variances, skewness and 1-D and 2-D marginals. We rigorously validated the algorithms by measuring MCMC sample quality through KSD, ACFs, and trace plots. For large-scale inference problems, particularly with limited computational resources, we highlight and recommend the approximate MCMC algorithm (e.g. Lip-ULA) as an alternative to the exact MCMC algorithms. However, this algorithm results in inflation of second statistical moment (variance) because of the asymptotic bias. We recommend that the geophysicists leverage the qualitative–quantitative trade-off when interpreting the statistical results. Despite this fact, approximate MCMC algorithms can provide fast sampling at efficient computational costs, especially on limited computational resources. Future work will further explore the potentials for computationally sophisticated approximate MCMC algorithms for Bayesian FWI.

DATA AND CODE AVAILABILITY

No new data were generated or analysed in support of this research. The programming codes to implement the algorithms underlying this article are available at https://github.com/izzatum/Langevin_GJI_2020

ACKNOWLEDGEMENTS

The first author would like to thank Tristan van Leeuwen at Utrecht University and Youssef M. Marzouk at MIT for visiting their research labs, which led to this work, and their continuous support. We thank Lester Mackey from Microsoft Research for providing guidance in the kernelized Stein discrepancy (KSD). We also gratefully acknowledge fruitful discussions with our colleagues Ricardo Baptista, Christopher Nemeth, Andrew Duncan and Felix J. Herrmann. Additionally, we thankfully acknowledge the constructive feedback from Andreas Fichtner and one anonymous reviewer. The research visits and the work reported herein were supported by funding from King Abdullah University of Science and Technology (KAUST).

REFERENCES

Ahn, S., Korattikara, A. & Welling, M., 2012. Bayesian posterior sampling via stochastic gradient Fisher scoring, in *Proceedings of the 29th International Coference on International Conference on Machine Learning,* ICML'12, pp. 1771–1778, Omnipress, Madison, WI, USA.

Biswas, R. & Sen, M., 2017. , 2D Full-Waveform Inversion and Uncertainty Estimation using the Reversible Jump Hamiltonian Monte Carlo, in *Proceedings of the SEG Technical Program Expanded Abstracts 2017*, Society of Exploration Geophysicists, pp. 1280–1285.

Bodin, T. & Sambridge, M., 2009. Seismic tomography with the reversible jump algorithm, *Geophys. J. Int.,* **178**(3), 1411–1436.

Borcea, L., Druskin, V., Mamonov, A.V. & Zaslavsky, M., 2019. Robust nonlinear processing of active array data in inverse scattering via truncated reduced order models, *J. Comput. Phys.,* **381,** 1–26.

Borcea, L., Druskin, V., Mamonov, A.V., Zaslavsky, M. & Zimmerling, J., 2020. Reduced order model approach to inverse scattering, *SIAM J. Imaging Sci.,* **13**(2), 685–723.

Brooks, S., Gelman, A., Jones, G. & Meng, X.-L., 2011. *Handbook of Markov chain Monte Carlo,* CRC Press.

Brosse, N., Moulines, E. & Durmus, A., 2018. The promises and pitfalls of stochastic gradient Langevin dynamics, in *Proceedings of the 32nd International Conference on Neural Information Processing Systems,* NIPS'18, pp. 8278–8288, Curran Associates Inc., Red Hook, NY, USA.

Brougois, A., Bourget, M., Lailly, P., Poulet, M., Ricarte, P. & Versteeg, R., 1990. Marmousi, model and data, in *Proceedings of the EAEG Workshop - Practical Aspects of Seismic Data Inversion*.

Bui-Thanh, T., Ghattas, O., Martin, J. & Stadler, G., 2013. A computational framework for infinite-dimensional Bayesian inverse problems. Part I: the linearized case, with application to global seismic inversion, *SIAM J. Sci. Comput.,* **35**(6), A2494–A2523.

Chen, W.Y., Barp, A., Briol, F.-X., Gorham, J., Girolami, M., Mackey, L. & Oates, C., 2019. Stein point Markov chain Monte Carlo, in *Proceedings of the 36th International Conference on Machine Learning,* Vol. 97 of Proceedings of Machine Learning Research, pp. 1011–1021, PMLR, Long Beach, CA, USA.

Choi, Y., Shin, C. & Min, D., 2007. *Frequency-Domain Elastic Full-Waveform Inversion Using the New Pseudo-Hessian Matrix: Elastic Marmousi-2 Synthetic Test,* pp. 1908–1912.

Chwialkowski, K., Strathmann, H. & Gretton, A., 2016. A kernel test of goodness of fit, in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - vol. 48,* ICML'16, pp. 2606–2615, JMLR.org.

Cui, T., Martin, J., Marzouk, Y.M., Solonen, A. & Spantini, A., 2014. Likelihood-informed dimension reduction for nonlinear inverse problems, *Inverse Problems,* **30**(11), 114015.

Cui, T., Law, K.J. & Marzouk, Y.M., 2016. Dimension-independent likelihood-informed MCMC, *J. Comput. Phys.,* **304,** 109–137.

Dalalyan, A., 2017. Further and stronger analogy between sampling and optimization: Langevin Monte Carlo and gradient descent, in *Proceedings of the 2017 Conference on Learning Theory,* Vol. 65 of Proceedings of Machine Learning Research, pp. 678–689, PMLR, Amsterdam, Netherlands.

Dalalyan, A.S. & Karagulyan, A.G., 2019. User-friendly guarantees for the Langevin Monte Carlo with inaccurate gradient.**129**(12), 5278–5311.

Dalalyan, A.S. & Riou-Durand, L., 2020. On sampling from a log-concave density using kinetic Langevin diffusions, *Bernoulli,* **26**(3), 1956–1988.

Drori, Y. & Teboulle, M., 2014. Performance of first-order methods for smooth convex minimization: a novel approach, *Math. Program.,* **145**(1–2), 451–482.

Durmus, A. & Moulines, E., 2017. Nonasymptotic convergence analysis for the unadjusted Langevin algorithm, *Ann. Appl. Probab.,* **27**(3), 1551–1587.

Durmus, A. & Moulines, E., 2019. High-dimensional Bayesian inference via the unadjusted Langevin algorithm, *Bernoulli,* **25**(4A), 2854–2882.

Durmus, A., Majewski, S. & Miasojedow, B., 2019. Analysis of Langevin Monte Carlo via convex optimization, *J. Mach. Learn. Res.,* **20**(73), 1–46.

Dwivedi, R., Chen, Y., Wainwright, M.J. & Yu, B., 2018. Log-concave sampling: Metropolis-Hastings algorithms are fast!, Vol. 75 of Proceedings of Machine Learning Research, pp. 793–797, PMLR.

Ermak, D.L., 1975. A computer simulation of charged particles in solution. I. Technique and equilibrium properties, *J. Chem. Phys.,* **62**(10), 4189–4196.

Fang, Z., Da Silva, C., Kuske, R. & Herrmann, F.J., 2018. Uncertainty quantification for inverse problems with weak partial-differential-equation constraints, *Geophysics*.**83**(6), R629–R647.

Fichtner, A. & Simutė, S., 2018. Hamiltonian monte carlo inversion of seismic sources in complex media, *J. geophys. Res.,* **123**(4), 2984–2999.

Fichtner, A. & Zunino, A., 2019. Hamiltonian nullspace shuttles, *Geophys. Res. Lett.,* **46**(2), 644–651.

Fichtner, A., Zunino, A. & Gebraad, L., 2019. Hamiltonian Monte Carlo solution of tomographic inverse problems, *Geophys. J. Int.,* **216**(2), 1344–1363.

Galetti, E., Curtis, A., Meles, G.A. & Baptie, B., 2015. Uncertainty loops in travel-time tomography from nonlinear wave physics, *Phys. Rev. Lett.,* **114,** 148501.

Gauthier, O., Virieux, J. & Tarantola, A., 1986. Two-dimensional nonlinear inversion of seismic waveforms: numerical results, *Geophysics*.**51**(7), 1387–1403.

Gebraad, L., Boehm, C. & Fichtner, A., 2020. Bayesian elastic full-waveform inversion using Hamiltonian Monte Carlo, *J. geophys. Res.,* **125**(3), e2019JB018428.

Girolami, M. & Calderhead, B., 2011. Riemann manifold Langevin and Hamiltonian Monte Carlo methods, *J. R. Stat. Soc., B,* **73**(2), 123–214.

Gorham, J. & Mackey, L., 2015. Measuring sample quality with Stein's method, in *Advances in Neural Information Processing Systems 28,* pp. 226–234, eds Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M. & Garnett, R., Curran Associates, Inc.

Gorham, J. & Mackey, L., 2017. Measuring sample quality with kernels, in *Proceedings of the 34th ICML'17: International Conference on Machine Learning,* Vol. 70, pp. 1292–1301, JMLR.org.

Gorham, J., Duncan, A.B., Vollmer, S.J. & Mackey, L., 2019. Measuring sample quality with diffusions, *Ann. Appl. Probab.,* **29**(5), 2884–2928.

Green, P.J., 1995. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination, *Biometrika,* **82**(4), 711–732.

Hansen, P.C. & Jørgensen, J.S., 2018. AIR Tools II: algebraic iterative reconstruction methods, improved implementation, *Numer. Algorithms,* **79**(1), 107–137.

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

Izzatullah, M., van Leeuwen, T. & Peter, D., 2019. Bayesian uncertainty estimation for full waveform inversion: a numerical study, in *SEG Technical Program Expanded Abstracts 2019*, pp. 1685–1689, Society of Exploration Geophysicists.

Izzatullah, M., Baptista, R., Mackey, L., Marzouk, Y. & Peter, D., 2020a. Bayesian seismic inversion: Measuring Langevin MCMC sample quality with kernels, in SEG Technical Program Expanded Abstracts 2020. pp. 295-299, Society of Exploration Geophysicists.

Izzatullah, M., Van Leeuwen, T. & Peter, D., 2020b. Langevin dynamics Markov Chain Monte Carlo solution for seismic inversion, **2020**(1), 1–5.

Kantorovich, L. & Akilov, G., eds, 1982. *Functional Analysis,* 2nd edn, Pergamon.

Kim, D. & Fessler, J.A., 2016. Optimized first-order methods for smooth convex minimization, *Math. Program.,* **159**(1–2), 81–107.

Koch, M.C., Fujisawa, K. & Murakami, A., 2020. Adjoint Hamiltonian Monte Carlo algorithm for the estimation of elastic modulus through the inversion of elastic wave propagation data, *Int. J. Numer. Methods Eng.,* **121**(6), 1037–1067.

Korattikara, A., Chen, Y. & Welling, M., 2014. Austerity in MCMC land: cutting the Metropolis-Hastings budget, in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32,* ICML'14, pp. I–181-I-189, JMLR.org.

Lemons, D.S. & Gythiel, A., 1997. Paul langevin's 1908 paper "on the theory of brownian motion" ["sur la théorie du mouvement brownien," c. r. acad. sci. (paris) 146, 530–533 (1908)], *Am. J. Phys.,* **65**(11), 1079–1081.

Liu, Q. & Peter, D., 2019b. Square-root variable metric based elastic full-waveform inversion - Part 2: uncertainty estimation, *Geophys. J. Int.*.**218**(2), 1100–1120.

Liu, Q. & Wang, D., 2016. Stein variational gradient descent: a general purpose Bayesian inference algorithm, in *Advances in Neural Information Processing Systems 29,* pp. 2378–2386, eds Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I. & Garnett, R., Curran Associates, Inc.

Liu, Q., Peter, D. & Tape, C., 2019a. Square-root variable metric based elastic full-waveform inversion - Part 1: theory and validation, *Geophys. J. Int.,* **218**(2), 1121–1135.

Malinverno, A., 2002. Parsimonious Bayesian Markov chain Monte Carlo inversion in a nonlinear geophysical problem, *Geophys. J. Int.,* **151**(3), 675–688.

Malinverno, A. & Briggs, V.A., 2004. Expanded uncertainty quantification in inverse problems: Hierarchical Bayes and empirical Bayes, *Geophysics,* **69**(4), 1005–1016.

Malitsky, Y. & Mishchenko, K., 2019. Adaptive gradient descent without descent, *arXiv preprint, arXiv:1910.09529*.

Martin, J., Wilcox, L.C., Burstedde, C. & Ghattas, O., 2012. A stochastic Newton MCMC method for large-scale statistical inverse problems with application to seismic inversion, *SIAM J. Sci. Comput.,* **34**(3), A1460–A1487.

Métivier, L., Brossier, R., Operto, S. & Virieux, J., 2017. Full waveform inversion and the truncated Newton method, *SIAM Rev.,* **59**(1), 153–195.

Mora, P., 1987. Nonlinear two-dimensional elastic inversion of multioffset seismic data, *Geophysics,* **52**(9), 1211–1228.

Mosegaard, K. & Tarantola, A., 1995. Monte Carlo sampling of solutions to inverse problems, *J. geophys. Res.,* **100**(B7), 12 431–12 447.

Neal, P. & Roberts, G., 2006. Optimal scaling for partially updating MCMC algorithms, *Ann. Appl. Probab.,* **16**(2), 475–515.

Nemeth, C. & Fearnhead, P., 2020. Stochastic gradient Markov Chain Monte Carlo, *J. Am. Stat. Assoc.,* **116**(533), 433–450 .

Nocedal, J. & Wright, S., 2006. *Numerical Optimization: Springer Series in Operations Research and Financial Engineering,* Springer.

Parisi, G., 1981. Correlation functions and computer simulations, *Nuclear Phys. B,* **180**(3), 378–384.

Pereyra, M., 2016. Proximal Markov Chain Monte Carlo algorithms, *Stat. Comput.,* **26**(4), 745–760.

Piana Agostinetti, N., Giacomuzzi, G. & Malinverno, A., 2015. Local three-dimensional earthquake tomography by trans-dimensional Monte Carlo sampling, *Geophys. J. Int.,* **201**(3), 1598–1617.

Plessix, R.-E., 2006. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications, *Geophys. J. Int.,* **167**(2), 495–503.

Polyak, B., 1963. Gradient methods for the minimisation of functionals, *USSR Comput. Math. Math. Phys.,* **3**(4), 864–878.

Polyak, B., 1969. Minimization of unsmooth functionals, *USSR Comput. Math. Math. Phys.,* **9**(3), 14–29.

Raginsky, M., Rakhlin, A. & Telgarsky, M., 2017. *Non-Convex Learning Via Stochastic Gradient Langevin Dynamics: A Nonasymptotic Analysis,* Vol. 65 of Proceedings of Machine Learning Research, pp. 1674–1703, PMLR, Amsterdam, Netherlands.

Rawlinson, N., Fichtner, A., Sambridge, M. & Young, M.K., 2014. Chapter one - Seismic tomography and the assessment of uncertainty, *Advances in Geophysics,* Vol. **55,** pp. 1–76, ed. Dmowska, R., Elsevier.

Roberts, G.O. & Rosenthal, J.S., 1998. Optimal scaling of discrete approximations to Langevin diffusions, *J. R. Stat. Soc., B,* **60**(1), 255–268.

Roberts, G.O. & Tweedie, R.L., 1996. Exponential convergence of Langevin distributions and their discrete approximations, *Bernoulli,* **2**(4), 341–363.

Sambridge, M., 1999a. Geophysical inversion with a neighbourhood algorithm–I. Searching a parameter space, *Geophys. J. Int.,* **138**(2), 479–494.

Sambridge, M., 1999b. Geophysical inversion with a neighbourhood algorithm–II. Appraising the ensemble, *Geophys. J. Int.,* **138**(3), 727–746.

Stuart, A.M., Voss, J. & Wilberg, P., 2004. Conditional path sampling of SDES and the Langevin MCMC method, *Commun. Math. Sci.,* **2**(4), 685–697.

Sun, B. & Alkhalifah, T., 2019. Ml-descent: an optimization algorithm for FWI using machine learning, *Geophysics,* **0**(ja), 1–135.

Tarantola, A., 1984. Inversion of seismic reflection data in the acoustic approximation, *Geophysics,* **49**(8), 1259–1266.

Tarantola, A., 1986. A strategy for nonlinear elastic inversion of seismic reflection data, *Geophysics,* **51**(10), 1893–1903.

Tarantola, A. & Valette, B., 1982a. Inverse problems = quest for information, *J. Geophys.,* **50**(1), 159–170.

Tarantola, A. & Valette, B., 1982b. Generalized nonlinear inverse problems solved using the least squares criterion, *Rev. Geophys.,* **20**(2), 219–232.

Teh, Y.W., Thiery, A.H. & Vollmer, S.J., 2016. Consistency and fluctuations for stochastic gradient Langevin dynamics, *J. Mach. Learn. Res.,* **17**(7), 1–33.

Tzikas, D.G., Likas, A.C. & Galatsanos, N.P., 2008. The variational approximation for Bayesian inference, *IEEE Signal Process. Mag.,* **25**(6), 131–146.

Virieux, J. & Operto, S., 2009. An overview of full-waveform inversion in exploration geophysics, *Geophysics,* **74**(6), WCC1–WCC26.

Welling, M. & Teh, Y.W., 2011. Bayesian learning via stochastic gradient Langevin dynamics, in *Proceedings of the 28th International Conference on International Conference on Machine Learning,* ICML'11, pp. 681–688, Omnipress, Madison, WI, USA.

Wibisono, A., 2018. Sampling as optimization in the space of measures: the Langevin dynamics as a composite optimization problem, *Proceedings of Machine Learning Research,* Vol. **75,** pp. 2093–3027, PMLR.

Zahm, O., Cui, T., Law, K., Spantini, A. & Marzouk, Y., 2018. Certified dimension reduction in nonlinear Bayesian inverse problems, arXiv preprint, arXiv:1807.03712.

Zhang, X. & Curtis, A., 2019. Seismic tomography using variational inference methods, *J. geophys. Res.,* **124,** doi:10.1029/2019JB018589.

Zhang, X. & Curtis, A., 2020. Variational full-waveform inversion, *Geophys. J. Int.,* **222**(1), 406–411.

Zhang, X., Curtis, A., Galetti, E. & de Ridder, S., 2018. 3-D Monte Carlo surface wave tomography, *Geophys. J. Int.,* **215**(3), 1644–1658.

Zhu, H., Li, S., Fomel, S., Stadler, G. & Ghattas, O., 2016. A Bayesian approach to estimate uncertainty for full-waveform inversion using a priori information from depth migration, *Geophysics,* **81**(5), R307–R323.

APPENDIX A: KERNELIZED STEIN DISCREPANCY

Approximate MCMC algorithms trade-off asymptotic correctness for increased sampling speed. In such cases, the stationary distribution does not match the target distribution $\boldsymbol{\pi}$, and the samples form a biased approximation of the distribution of interest.

The standard MCMC diagnostics include mean and trace plots, pooled and within-chain variance measures, effective sample size and asymptotic variance (Brooks *et al.* 2011). However, these statistics measure the convergence to the stationary distribution without considering asymptotic bias. Thus, these diagnostic criteria are not appropriate for either assessing convergence to a desired target or for tuning MCMC algorithms, in particular, for approximate Langevin dynamics MCMC algorithms. The design of the appropriate diagnostic tests for approximate MCMC algorithms is a relatively new research area. Recently, computable Stein discrepancies have been introduced as a new family of diagnostic criteria for comparing sample quality (Gorham & Mackey 2015, 2017; Gorham *et al.* 2019).

One class of measures that can be evaluated in closed-form based on the theory of the reproducing kernel Hilbert spaces (RKHS) is KSD, presented by Chwialkowski *et al.* (2016), Liu & Wang (2016), Gorham & Mackey (2017) and Izzatullah *et al.* (2020a).

The KSD between an approximate distribution $\tilde{\boldsymbol{\pi}}$ and the target distribution $\boldsymbol{\pi}$ is defined as

$$d_{KSD}(\tilde{\pi}, \pi) = \sqrt{\sum_{j=1}^{d} \frac{1}{N^2} \sum_{i,i'=1}^{N} k_0^j(m_i, m_{i'})}, \tag{A1}$$

where k_0^j is a Stein kernel. To measure the discrepancy between the distributions, KSD sums the Stein kernel k_0^j evaluations across all pairs of N samples for each dimension of the model parameter $\boldsymbol{m} \in \boldsymbol{R}^d$; the Stein kernel k_0^j for $j \in \{1, \dots, d\}$ is given by

$$\begin{aligned} k_0^j(\boldsymbol{m}, \boldsymbol{m}') = &(\nabla_{\boldsymbol{m}^{(j)}} \log \boldsymbol{\pi}(\boldsymbol{m})^T \nabla_{\boldsymbol{m}'^{(j)}} \log \boldsymbol{\pi}(\boldsymbol{m}'))k(\boldsymbol{m}, \boldsymbol{m}') \\ &+\nabla_{\boldsymbol{m}'^{(j)}} \log \boldsymbol{\pi}(\boldsymbol{m}')^T \nabla_{\boldsymbol{m}^{(j)}} k(\boldsymbol{m}, \boldsymbol{m}') \\ &+\nabla_{\boldsymbol{m}^{(j)}} \log \boldsymbol{\pi}(\boldsymbol{m})^T \nabla_{\boldsymbol{m}'^{(j)}} k(\boldsymbol{m}, \boldsymbol{m}') \\ &+\nabla_{\boldsymbol{m}^{(j)}} \nabla_{\boldsymbol{m}'^{(j)}} k(\boldsymbol{m}, \boldsymbol{m}'). \end{aligned} \tag{A2}$$

The kernel $k(\boldsymbol{m}, \boldsymbol{m}')$ plays a significant role in detecting non-convergence to the target distribution π, and must therefore be chosen appropriately. Gorham & Mackey (2017) and Chen *et al.* (2019) recommend using the pre-conditioned inverse multiquadric kernel, $k(\boldsymbol{m}, \boldsymbol{m}') = (c^2 + ||\boldsymbol{\Lambda}^{-1/2}(\boldsymbol{m} - \boldsymbol{m}')||_2^2)^\beta$, which was proven to detect non-convergence for a wide class of target densities π when $c > 0$ and $\beta \in (-1, 0)$ with $\boldsymbol{\Lambda}$ for some symmetric positive definite matrix. Note that the matrix $\boldsymbol{\Lambda}$ can also form part of an MCMC transition kernel, such as the pre-conditioner matrix in MALA (Girolami & Calderhead 2011). KSD has a computational complexity of $\mathcal{O}(N^2)$.

With this choice of kernel, we can use KSD to evaluate the bias that arises from using finite samples of the MCMC algorithms, (particularly approximate MCMC algorithms) to characterize the target distribution $\boldsymbol{\pi}$. The two factors contributing to the bias are: (i) the asymptotic bias from the choice of step size τ in the Langevin dynamics MCMC methods and (ii) the non-asymptotic bias from the correlation between MCMC samples.

We provide four simple demonstrations of KSD in detecting convergence, partial convergence, and non-convergence to a target density $\boldsymbol{\pi}$. We consider a multivariate Gaussian density with dimension 20, $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_{20\times 20})$ as the target density $\boldsymbol{\pi}$. We demonstrate the KSD evaluation for the following numerical examples.

(1)Detecting convergence: the samples are drawn exactly from the target density $\boldsymbol{\pi}$.
(2)Detecting partial convergence: the samples are drawn from the target density $\boldsymbol{\pi}$ with a perturbed mean, $\boldsymbol{\mu} = [1, 0, \dots, 0]^T$.
(3)Detecting partial convergence: The samples are drawn from the target density $\boldsymbol{\pi}$ with a perturbed variance, $\boldsymbol{\Sigma} = \mathrm{diag}\big([0.001, 1, \dots, 1]\big)$.
(4)Detecting non-convergence: The samples are drawn from a different density, in this case from Gamma density $X \sim \boldsymbol{\Gamma}(7.5, 1.0)$.

For all numerical examples mentioned above, we draw 10 000 i.i.d. samples from the respective mentioned densities for KSD evaluation with respect to the target density $\boldsymbol{\pi}$. The results are shown in Fig. A1, where the behaviours in examples 2 and 3 are commonly observed in MCMC methods, particularly in approximate MCMC algorithms. These phenomena are related to the rate of MCMC convergence in each dimension in reaching a certain precision in approximating the statistical moments of target density $\boldsymbol{\pi}$; they commonly depend on the chosen MCMC algorithm, density geometry, covariance matrix spectrum and other factors.

APPENDIX B: THE CHOICE OF STEP SIZE: ULA VERSUS MALA

To achieve an optimal acceptance rate of 57.4 per cent for MALA (Roberts & Rosenthal 1998), we need to tune the step size τ accordingly. Here, we demonstrate that the choice of τ plays a significant role in MALA achieving the optimal acceptance rate. This choice is also important for ULA in maintaining the discretization-bias trade-off owing to the absence of Metropolis–Hastings correction steps.

We perform three different simulations using a set of step sizes for both MALA and ULA with $N = 30\,000$ samples, and $\tau = 2.59 \times 10^{-2}$, 2.59×10^{-1} and 2.59, respectively. We discard the first 15 000 samples as burn-in; the step size $\tau = 2.59 \times 10^{-1}$ is the optimal step size for MALA that provides the optimal acceptance rate of ∼57.4 per cent according to Roberts & Rosenthal (1998). We assess the KSD convergence of MALA and ULA for the respective step sizes and demonstrate that these algorithms are sensitive to the choice of step size τ. A minimal value of τ leads to slow convergence while a big τ drives ULA to diverge and provides MALA with a low acceptance rate. We illustrate the findings in Figs A2– A3.

APPENDIX C: THE CHOICE OF PRE-CONDITIONING MATRIX

We divide this appendix into two subsections to demonstrate the importance and influence of the pre-conditioning matrix for the MCMC performance and the sample quality. We demonstrate the following cases.

(i) Full covariance: $\boldsymbol{\Sigma} = \boldsymbol{H}^{-1}$ with $\boldsymbol{H} = \boldsymbol{A}^T\boldsymbol{A} + \boldsymbol{L}^T\boldsymbol{L}$ as the Hessian with respect to the negative log-posterior $-\log \boldsymbol{\pi}(\boldsymbol{m}|\boldsymbol{D})$.
(ii) Inverse of the pseudo-Hessian: $\boldsymbol{\Sigma} = \mathrm{diag}\big(\boldsymbol{H}\big)^{-1}$ with $\mathrm{diag}\big(\boldsymbol{H}\big) = \mathrm{diag}\big(\boldsymbol{A}^T\boldsymbol{A} + \boldsymbol{L}^T\boldsymbol{L}\big)$ as the diagonal of the Hessian with respect to the negative log-posterior $-\log \boldsymbol{\pi}(\boldsymbol{m}|\boldsymbol{D})$.

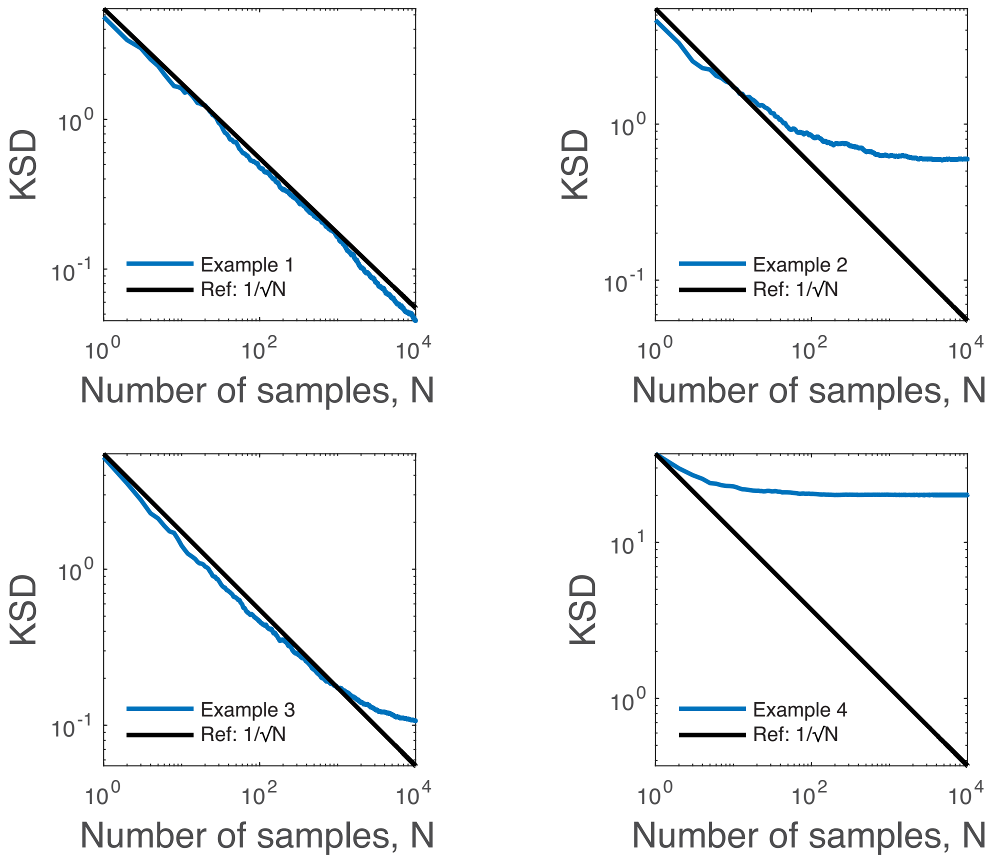

Figure A1. Top left-hand panel: the i.i.d. samples draw exactly from target density $\boldsymbol{\pi}$. Top right-hand panel: the i.i.d. samples draw from the target density $\boldsymbol{\pi}$ with perturbed mean, $\boldsymbol{\mu} = [1, 0, \ldots, 0]^T$. Bottom left-hand panel: the i.i.d. samples draw from the target density $\boldsymbol{\pi}$ with perturbed variance, $\boldsymbol{\Sigma} = \text{diag}\Big([0.001, 1, \ldots, 1]\Big)$. Bottom right-hand panel**:** the i.i.d. samples draw from Gamma density $X \sim \boldsymbol{\Gamma}(7.5, 1.0)$.

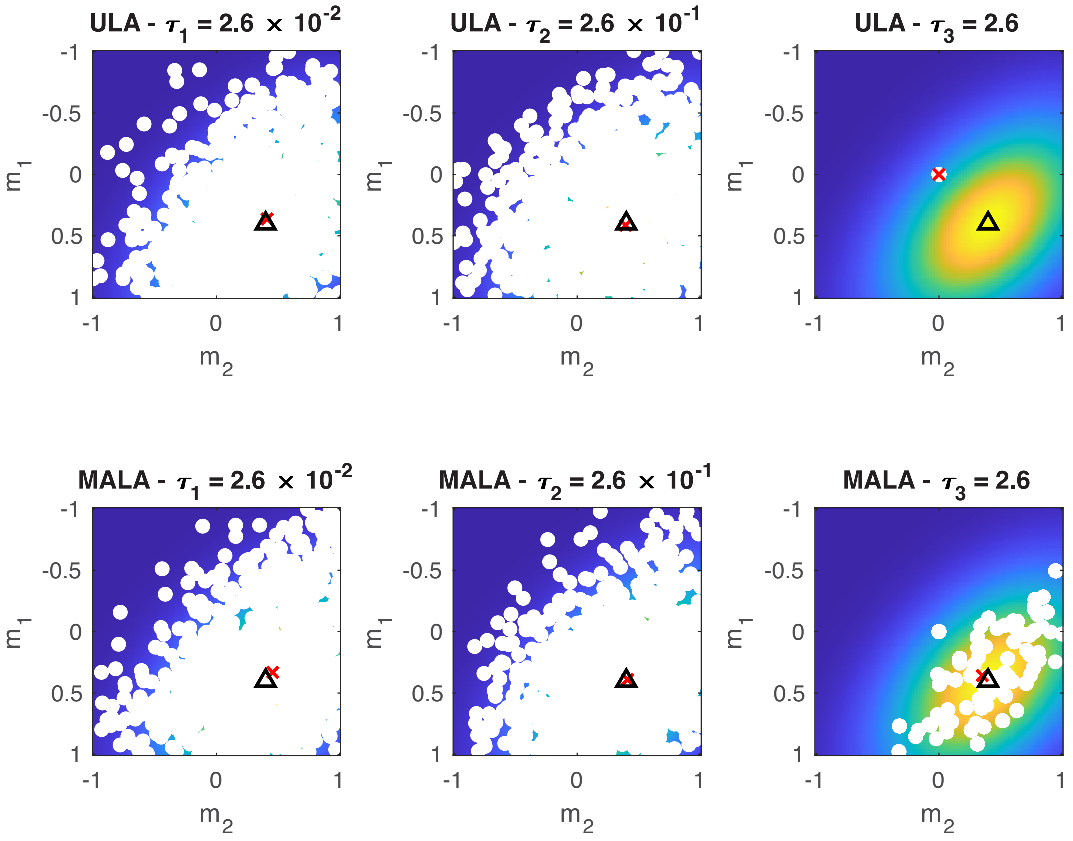

Figure A2. Samples of the bivariate Gaussian density are plotted from both ULA and MALA with respective step sizes, $\tau = 2.59 \times 10^{-2}$, 2.59×10^{-1} and 2.59. The black triangle in each figure represents the true mean, $\boldsymbol{\mu}$, whereas the red cross represents the sample mean, $\overline{\boldsymbol{m}}$.

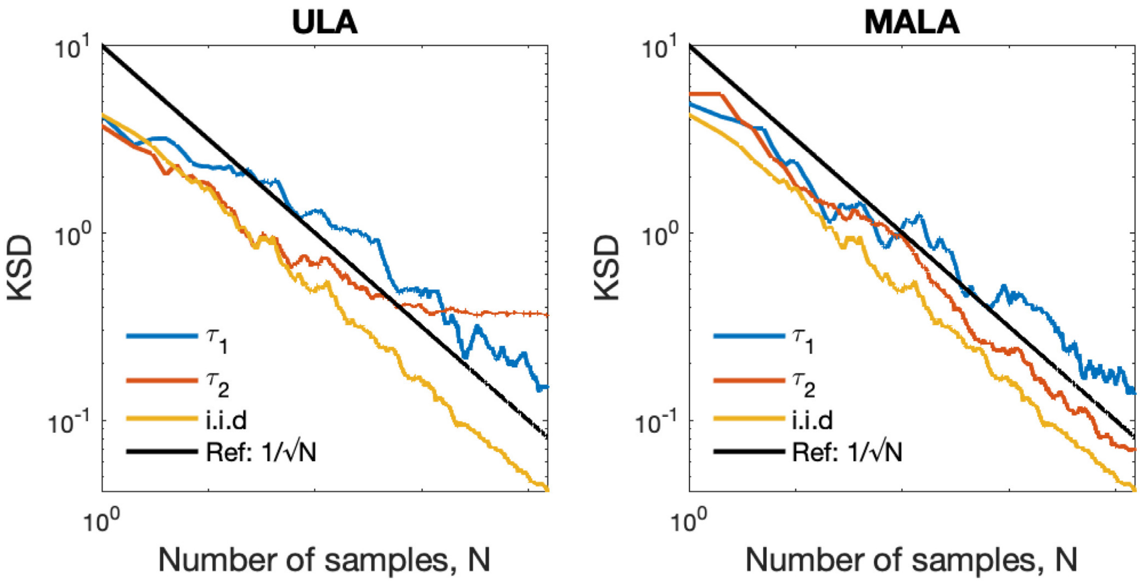

Figure A3. KSD in the log–log scale for the samples from two different algorithms for the bivariate Gaussian density: ULA and MALA.

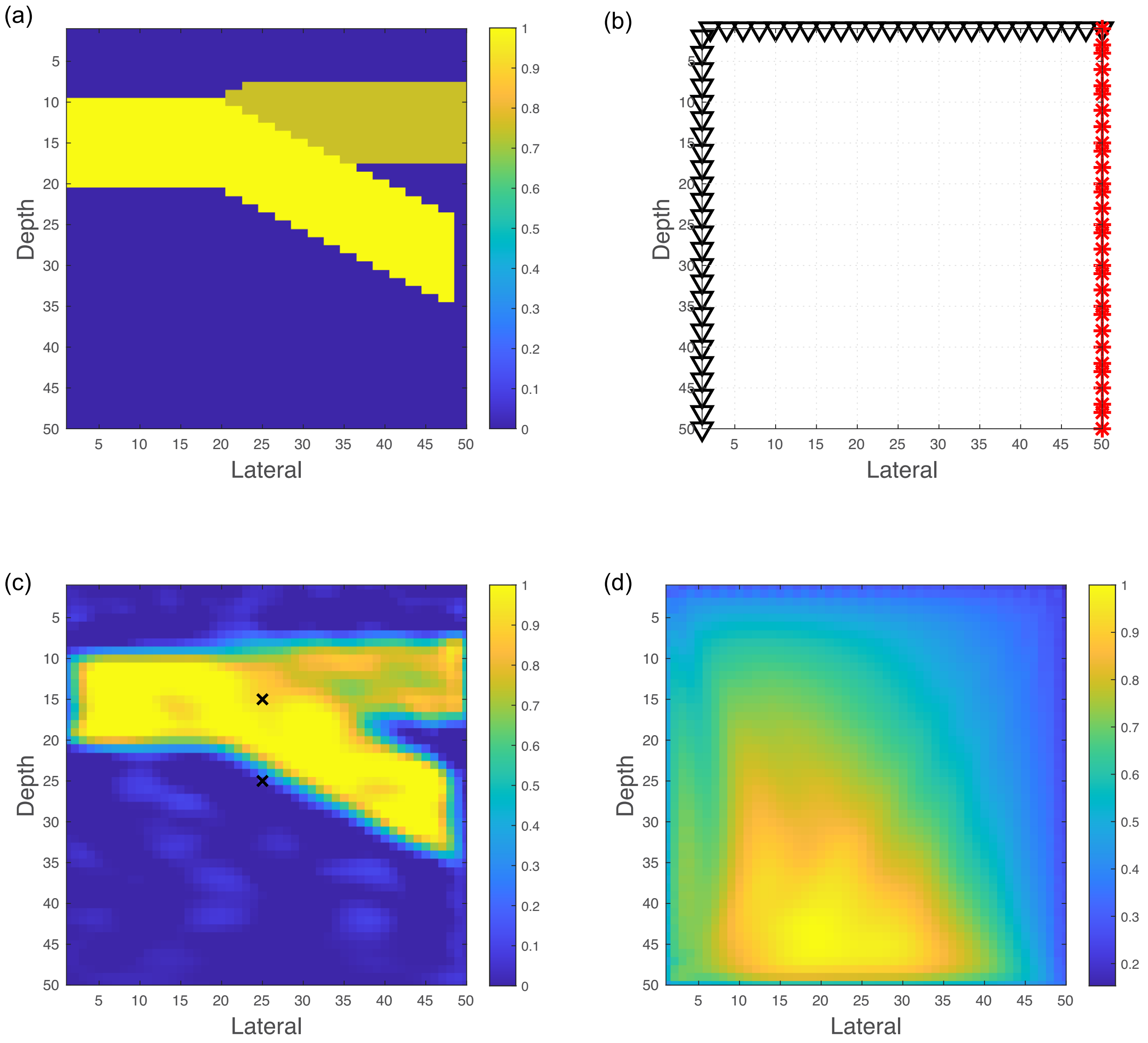

Figure A4. Full covariance pre-conditioner: (a) true model; (b) square domain with sources (red stars) and receivers (black triangles); (c) the posterior mean, $\boldsymbol{\mu}$, with black crosses representing the chosen elements for MCMC diagnostics, namely the 1215th and 1225th elements and (d) the posterior variance, $\boldsymbol{\sigma}^2$.

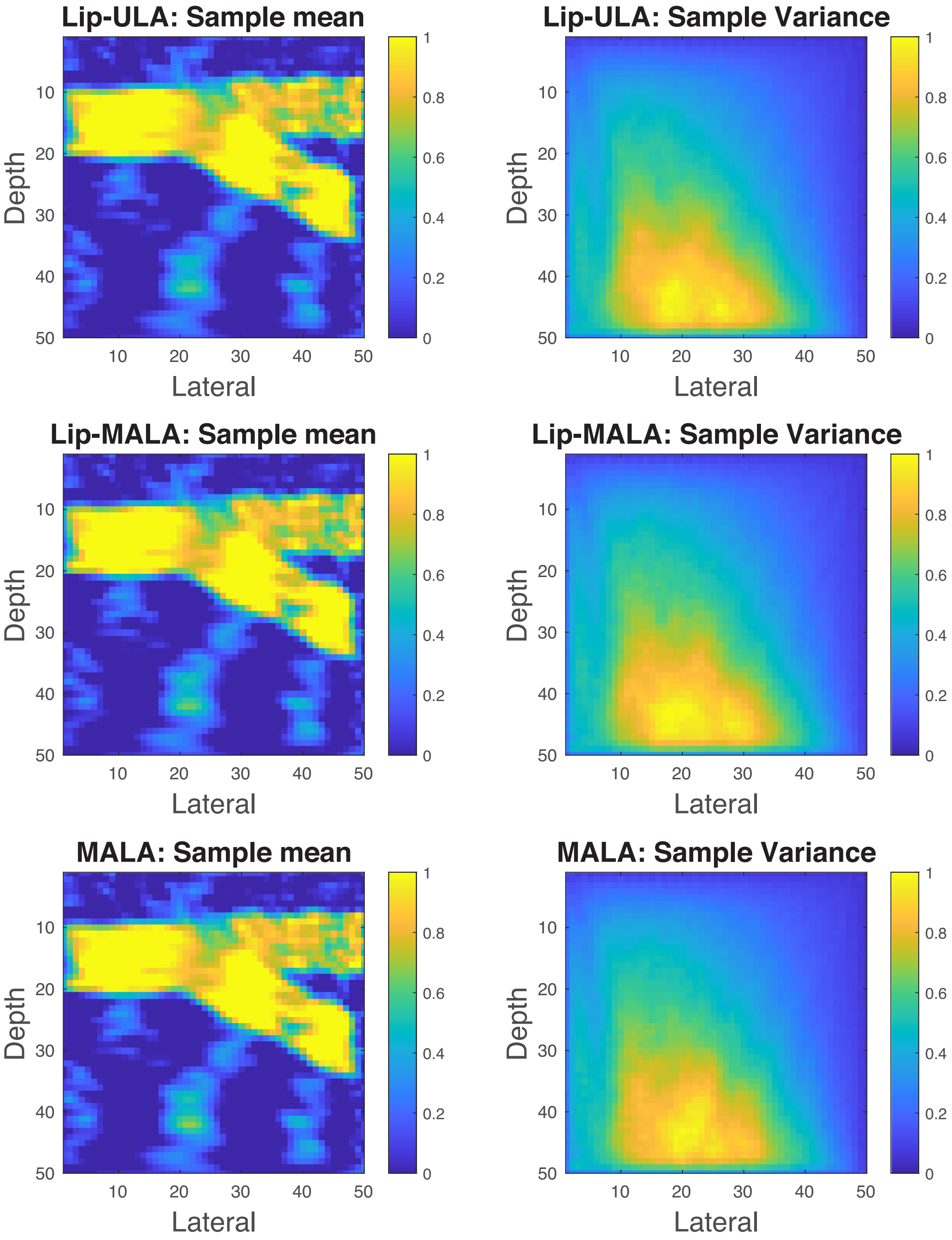

Figure A5. Full covariance pre-conditioner. The sample mean $\overline{m}$ and variance S^2 for three Langevin dynamics MCMC algorithms with $N = 5 \times 10^4$ samples. First row: Lip-ULA; second row: Lip-MALA and third row: MALA.

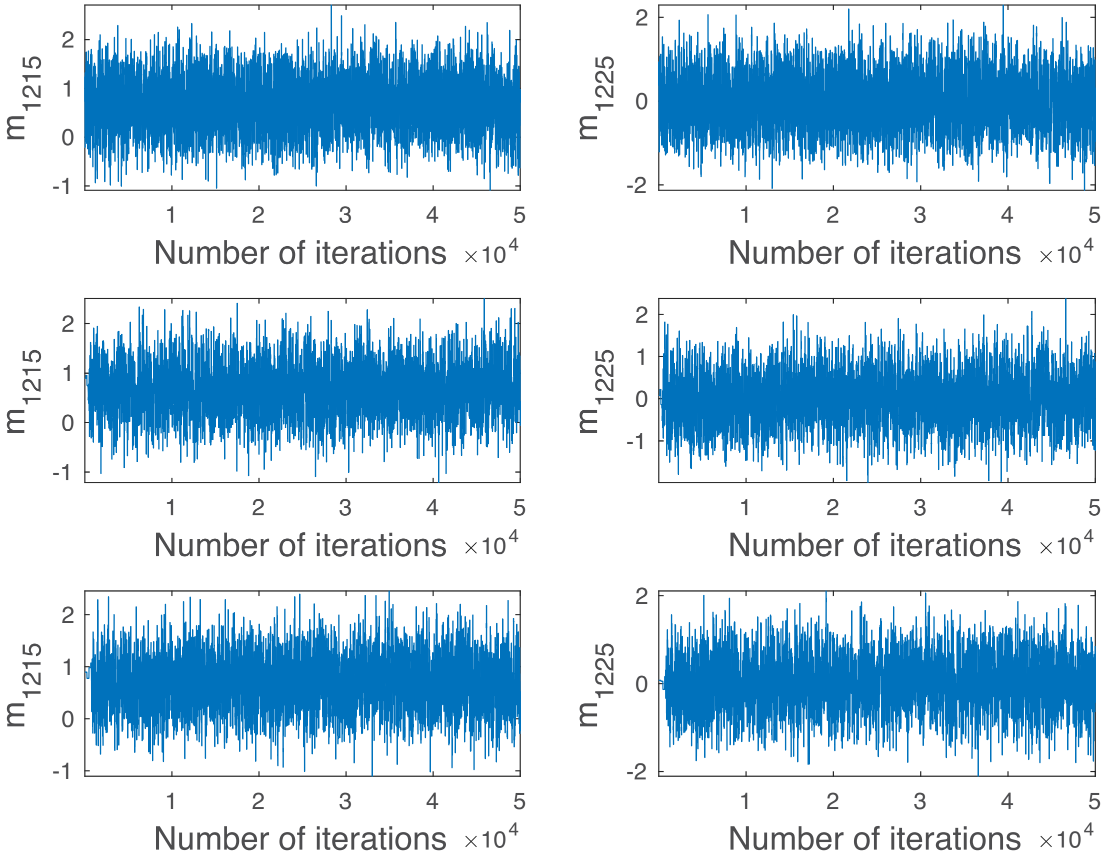

Figure A6. Full covariance pre-conditioner. The trace plots from two slowness model parameters, m_{1215} and m_{1225}. First row: Lip-ULA; second row: Lip-MALA and third row: MALA.

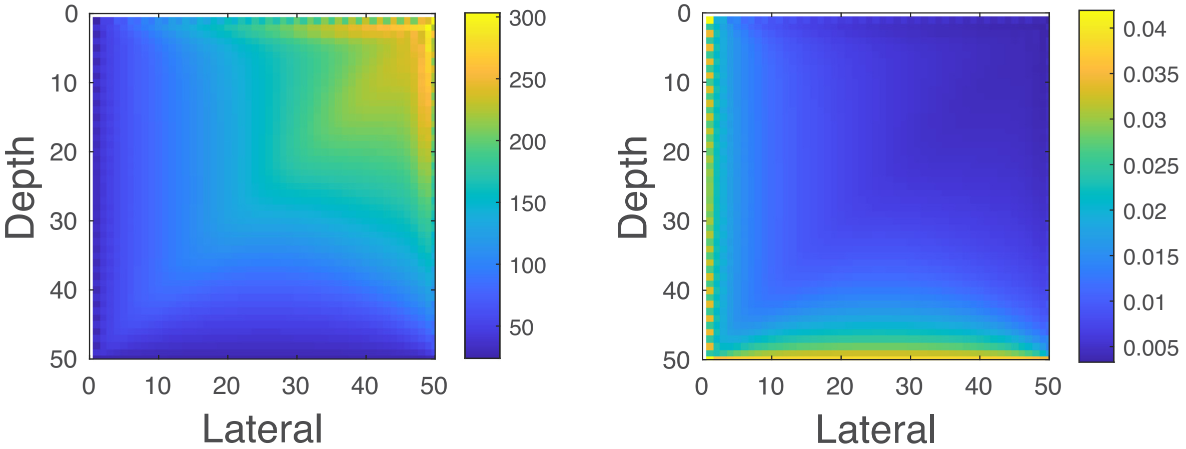

Figure A7. Inverse of the pseudo-Hessian pre-conditioner. Left-hand panel: the diagonal of the Hessian [$\mathrm{diag}\left(\boldsymbol{H}\right)$] and right-hand panel: the inverse of the Hessian [$\mathrm{diag}\left(\boldsymbol{H}\right)^{-1}$].

We consider a linear traveltime tomography problem, as presented in (Hansen & Jørgensen 2018) with the Gaussian posterior density as

$$\pi(\boldsymbol{m}|\boldsymbol{D}) \propto \exp\left(-\frac{1}{2}||\boldsymbol{A}\boldsymbol{m} - \boldsymbol{D}||^2_{\boldsymbol{C}_D} - \frac{1}{2}||\boldsymbol{L}\boldsymbol{m}||^2_2\right), \tag{C1}$$

where $\boldsymbol{A}$ is the linear forward problem operator which maps the slowness model $\boldsymbol{m}$ to the observed seismic traveltime data $\boldsymbol{D}$. The domain for the slowness model parameters $\boldsymbol{m}$ is a square of size [0, 50] × [0, 50] as in Fig. A4(a). The domain contains 30 equally spaced sources located

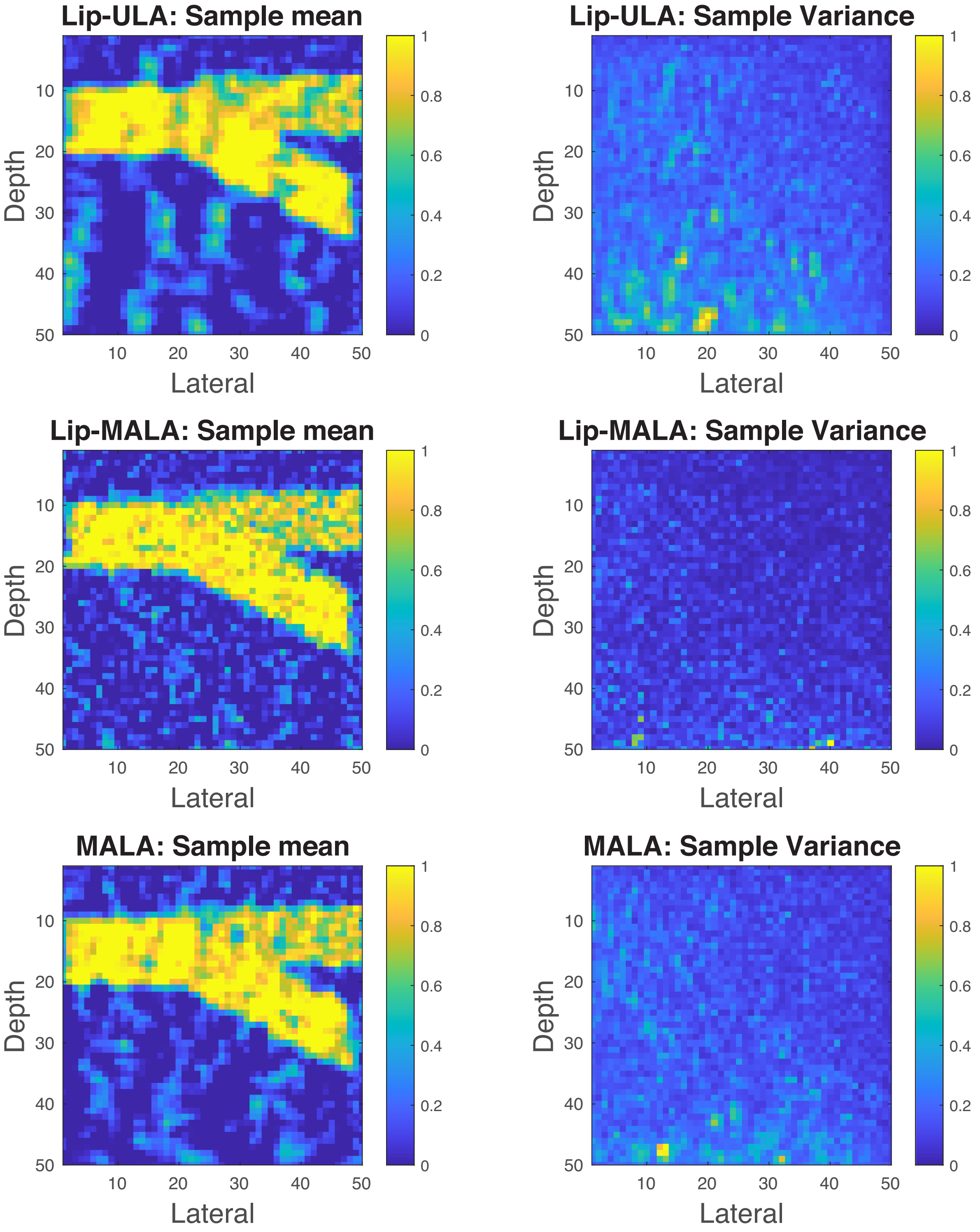

Figure A8. Inverse of the pseudo-Hessian pre-conditioner: the sample mean $\overline{m}$ and variance S^2 for three Langevin dynamics MCMC algorithms with $N = 5 \times 10^4$ samples. First row: Lip-ULA; second row: Lip-MALA and third row: MALA.

at the right-hand side, as well as 50 equally spaced receivers located on top of the surface and the left-hand side, as displayed in Fig. A4(b). The signal-to-noise ratio in the data is 22.6 dB and the relative standard deviation of the observation noise to the data is 1 per cent.

For this problem, the data error covariance matrix is set to $\boldsymbol{C_D} = \sigma_D^2 \boldsymbol{I_D}$ with $\sigma_D = 0.051$ and $\boldsymbol{I_D}$ being the identity matrix. We use a Gaussian prior density $\mathcal{N}(\mathbf{0}, \boldsymbol{C}_{\text{prior}})$ for the parameters, where $\boldsymbol{C}_{\text{prior}} = (\boldsymbol{L}^T\boldsymbol{L})^{-1}$ and $\boldsymbol{L}$ is a Laplacian matrix. In Figs A4(c) and (d), we plot the posterior mean and standard deviation, respectively. For this Gaussian posterior density, the posterior mean corresponds to the maximum *a posteriori* (MAP) model. We compute the MAP model by minimizing the negative log-posterior using 100 conjugate gradient

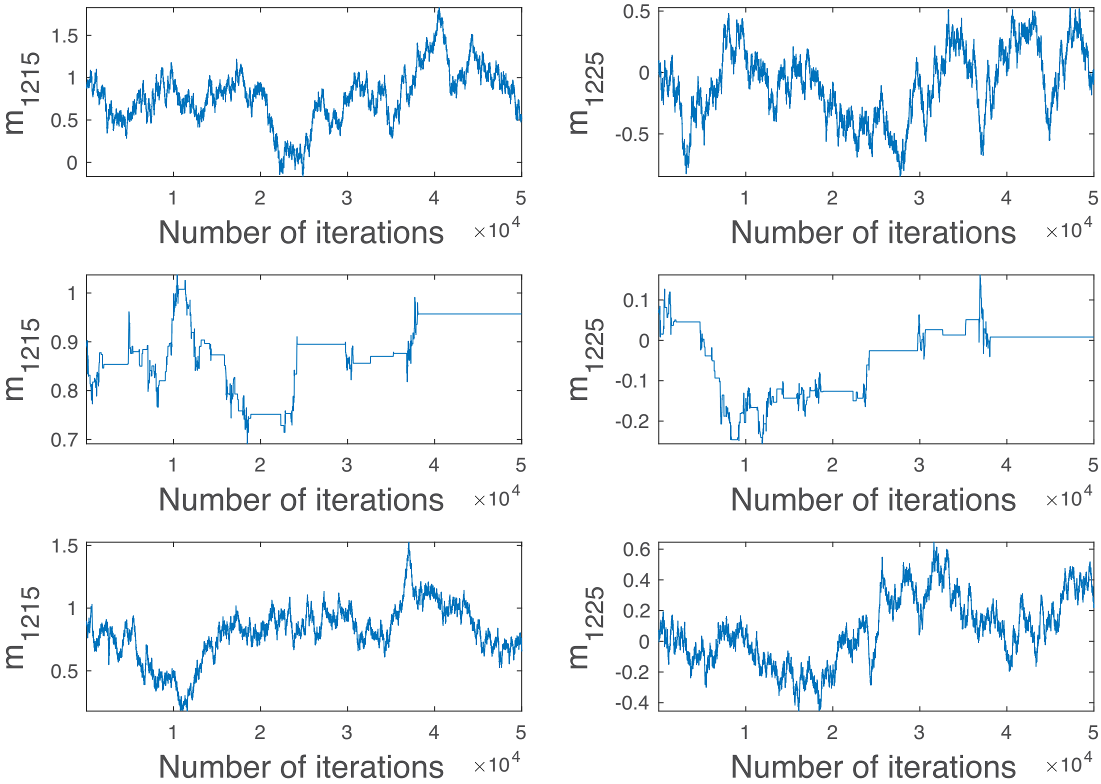

Figure A9. Inverse of the pseudo-Hessian pre-conditioner: The trace plots for two of the slowness model parameters, m_{1215} and m_{1225}. First row: Lip-ULA; second row: Lip-MALA and third row: MALA.

iterations (Hansen & Jørgensen 2018). The posterior standard deviation is obtained by inverting the Hessian matrix of the negative log-posterior $[-\log \boldsymbol{\pi}(\boldsymbol{m}|\boldsymbol{D})]$.

To sample from this target density, we initialize all algorithms from the MAP model in Fig. A4(c) as a warm start. Heuristically choosing an optimal step size for MALA in this numerical example is nontrivial; we consider the relationship $\tau_{\text{opt}} \propto d^{-1/3}$ as our guideline. We run the MCMC algorithms for $N = 5 \times 10^4$ steps without considering the burn-in period. In this example, we further introduce a pre-conditioning matrix to the respective algorithms.

C.1 Full covariance pre-conditioner

In this experiment, we perform the above-mentioned simulations with a step size of $\tau = 0.075$. Fig. A5 illustrates the resulting sample mean $\overline{\boldsymbol{m}}$ and variance $\boldsymbol{S}^2$ maps for all three algorithms. We observe that the sample mean $\overline{\boldsymbol{m}}$ and variance $\boldsymbol{S}^2$ for all algorithms match closely with the MAP model and the posterior variance in Fig. A4. In this problem, Lip-ULA accepts the sample in each iteration with probability 1, whereas Lip-MALA and MALA accept the samples with acceptance rates of 71.83 and 70.63 per cent, respectively.

We perform MCMC diagnostic to evaluate the sampling quality. Fig. A6 shows the trace plots for two model parameters, m_{1215} and m_{1225}, respectively. We observe that, by pre-conditioning the algorithms, the MCMC chains mix well and reach the stationary target density $\boldsymbol{\pi}(\boldsymbol{m}|\boldsymbol{D})$ with comparatively small sample sizes for such a large-scale problem.

C0.2 Inverse of the pseudo-Hessian pre-conditioner

Here, we use the same simulation setting as in the previous example, except that we pre-condition the algorithms with the inverse of a pseudo-Hessian (as displayed in Fig. A7) and set the step size $\tau = 0.0052$. Fig. A8 illustrates the sample mean $\overline{\boldsymbol{m}}$ and variance $\boldsymbol{S}^2$ for all three algorithms. We observe that the quality of these parameters deteriorate for all algorithms in comparison to the full covariance pre-conditioner case. We further observe that the sample mean $\overline{\boldsymbol{m}}$ of all algorithms is noisy with $\boldsymbol{S}^2$ converging towards the posterior variance σ^2; however, both illustrations indicate a slow convergence. Such situations can occur owing to a poor choice of the pre-conditioning matrix and a mismatch between the step-size scaling and the chosen pre-conditioner, which is the inverse of the pseudo-Hessian in this case. This leads to an acceptance rate of 1.49 per cent for Lip-MALA and 55.11 per cent for MALA.

We perform MCMC diagnostic to further analyse the sampling quality and show trace plots in Fig. A9 for two elements of the model parameters, m_{1215} and m_{1225}. We observe that the chains presented in Fig. A9 show a slow mixing, and there seem to be few independent observations in our samples.

In both subsections for this example, we observe that a poor choice of pre-conditioning matrix and a mismatch between the scaling of step size τ influences the MCMC performance and sample quality, even for the exact MCMC algorithms (i.e. Lip-MALA and MALA). In practice, multiplication and factorization of the full pre-conditioning matrix are computationally expensive, particularly for large-scale inference problems. A diagonal pre-conditioner may be an alternative, but it should be chosen wisely to ensure the best performance and MCMC sample quality.