LANGEVIN DIFFUSION FOR POPULATION BASED SAMPLING WITH AN APPLICATION IN BAYESIAN INFERENCE FOR PHARMACODYNAMICS

GEORGIOS ARAMPATZIS*, DANIEL WÄLCHLI†, PANAGIOTIS ANGELIKOPOULOS†‡, STEPHEN WU†§, PANAGIOTIS HADJIDOUKAS†, AND PETROS KOUMOUTSAKOS†¶

Abstract. We propose an algorithm for the efficient and robust sampling of the posterior probability distribution in Bayesian inference problems. The algorithm combines the local search capabilities of the manifold Metropolis adjusted Langevin transition kernels with the advantages of global exploration by a population based sampling algorithm, the transitional Markov chain Monte Carlo (TMCMC). The Langevin diffusion process is determined by either the Hessian or the Fisher information of the target distribution with appropriate modifications for non-positive definiteness. The present method is shown to be superior to other population based algorithms, in sampling probability distributions for which gradients are available, and is shown to handle otherwise unidentifiable models. We demonstrate the capabilities and advantages of the method in computing the posterior distribution of the parameters in a Pharmacodynamics model, for glioma growth and its drug induced inhibition, using clinical data.

Key words. Bayesian inference, Langevin diffusion, population based sampling, Pharmacodynamics

AMS subject classifications. 62F15, 62D05

DOI. 10.1137/16M1107401

1. Introduction. The unprecedented availability of experimental and observational data and increased computational power have fueled the re-emergence of Bayesian inference [23] for quantifying the uncertainty in the predictions of mathematical models. Bayesian inference is revolutionizing simulation science by integrating mathematical models with data and prior knowledge to enable robust predictions [22, 23, 24] in data rich domains such as medicine and drug discovery [37].

The Bayesian framework amounts to adjusting a subjective belief of a computational model and its parameters using data from the underlying physical process. The resulting posterior probability can then be used to robustly quantify the uncertainty in the model predictions. The practical value and computational cost of the Bayesian framework is largely determined by the effective sampling of the associated probability distributions. In the last decade a number of algorithms have been proposed to enhance this sampling by improving on the fundamental concept of Markov chain Monte Carlo (MCMC) [4, 9, 18, 28, 41]. Of particular interest are Monte Carlo algorithms [13] that may exploit massively parallel computer architectures. While

*Submitted to the journal’s Computational Methods in Science and Engineering section December 13, 2016; accepted for publication (in revised form) March 14, 2018; published electronically May 10, 2018.

http://www.siam.org/journals/sisc/40-3/M110740.html

Funding: This work was supported by the Swiss National Supercomputing Center (CSCS) under the project s659 and by the European Research Council (Advanced Investigator Award 341117).

†ETH-Zürich, CH-8092, Zurich, Switzerland (garampat@ethz.ch, wadaniel@student.ethz.ch, panagiotis.chatzidoukas@mavt.ethz.ch).

‡ETH-Zürich, CH-8092, Zurich, Switzerland. Current address: D. E. Shaw Research L.L.C., New York, NY 10036 (pangelikopoulos@gmail.com).

§ETH-Zürich, CH-8092, Zurich, Switzerland. Current address: Institute of Statistical Mathematics, 190-8562 Tokyo, Japan (stephenwu5@gmail.com).

¶Corresponding author. ETH-Zürich, CH-8092, Zurich, Switzerland (petros@ethz.ch).
MCMC algorithms are fundamentally “sequential,” several algorithms have been proposed in the last 20 years to address their parallel implementation. Among these algorithms are the annealed importance sampling [30], the power posterior algorithm [15], the parallel tempering algorithm [21], the equi-energy sampler [26], and parallel adaptive Metropolis [39]. The transitional MCMC algorithm (TMCMC) [10] and its improved version (BASIS) [40] are among the most prominent population (or particle) based MCMCs. They exploit information accumulated in the population to escape local modes in order to explore effectively multimodal or highly peaked probability distributions [17]. In TMCMC, a partial MCMC is performed for each member of the population via the Metropolis–Hastings (MH) algorithm with Gaussian proposal distribution [33]. The MH algorithm with Gaussian proposal distribution and constant covariance matrix corresponds to the discretization of an isotropic diffusion in the parameter space [34]. In TMCMC the matrix in the proposal distribution is the scaled sample covariance matrix of the population. However, in many situations, for instance, in multimodal or close to unidentifiable posterior distributions, the assumption of an isotropic covariance matrix, inherent to TMCMC, is inadequate. We note that sampling based on a nonisotropic covariance has been reported in the Metropolis adjusted Langevin algorithm (MALA) [34, 35]. In MALA, the proposal distribution is obtained from the discretization of a Langevin diffusion with a drift term related to the gradient of the target distribution and an adjustable constant diffusion coefficient. The optimal selection of the respective covariance matrix remains an open problem. In [16] the MALA algorithm is combined with a discretization of the Langevin diffusion on a general manifold. The authors proposed a covariance matrix that is connected with the Hessian or the Fisher information of the target distribution. The use of a covariance matrix is related to the Hessian [7] pertaining to the Newton method and to population based optimization algorithms such as CMA-ES [20]. Another class of potent sampling algorithms is the hybrid or Hamiltonian Monte Carlo (HMC) algorithms [14, 16] and their extensions on a general manifold [6]. We refer the reader to [17] for an extensive review in sampling algorithms for Bayesian computations.

In this article we augment the capabilities of the TMCMC algorithm by substituting the isotropic coefficient with a drift term and a diffusion coefficient that reflect the local geometry of the posterior distribution. We combine the TMCMC algorithm with Langevin diffusion transition kernels by following the manifold approach presented in [16]. We explore the Hessian and/or the Fisher information of the target distribution as local metrics to construct the covariance matrix of the proposal distribution. The proposed manifold TMCMC (mTMCMC) has been implemented for single core\(^1\) and multicore clusters\(^2\) [19]. Positive definiteness of the local metric is a key feature of MALA. However, this property is not ensured a priori for every probability distribution. Moreover, in the presence of nonidentifiable manifolds in the target distribution, the eigenvalues of the metric become arbitrarily small, leading to proposal distributions with ill-conditioned covariance matrices, resulting in sampling with low acceptance rate. In this paper, the noninvertibility is handled by disregarding the metric and using the covariance matrix usually employed by the TMCMC algorithm. The cases of non–positive definiteness and nonidentifiability are treated by decomposing the matrix associated with the particular metric and appropriately scaling the problematic eigenvectors. This technique is discussed in detail in subsection 3.1.

The proposed algorithm is first tested in a collection of multivariate Gaussian

\(^{1}\)MATLAB code can be downloaded from http://cse-lab.ethz.ch/software/smtmcmc.

\(^{2}\)Pi4U can be downloaded from http://www.cse-lab.ethz.ch/software/Pi4U.
distributions to showcase the advantages of the proposed metric correction scheme. The effectiveness of mTMCMC to sample challenging posterior distributions is further demonstrated in the Bayesian inference of a Pharmacodynamics problem. The Pharmacodynamics model describes the evolution of the mean diameter of low grade gliomas [32] under different drug therapies. Clinical data obtained from MRIs of different patients are used to infer the model parameters. The posterior probability distribution of the parameters is not readily invertible. The TMCMC algorithm [10] was unable to reproduce the posterior probability. The presence of at least one nonidentifiable manifold in the posterior distribution is being exposed by the use of the profile likelihood technique [31]. We find that while the TMCMC algorithm exhibited inability to sample the parameter space, the mTMCMC was well capable of exploring the parameter space. Finally, we demonstrate the ability of mTMCMC to sample areas of high probability by comparing its results with those from CMA-ES, a state-of-the-art population based optimization algorithm [20].

The paper is organized as follows: in section 2 we present background information on Bayesian inference and sampling using TMCMC and manifold MCMC. In section 3 we present the proposed mTMCMC algorithm and discuss the implementation details, and in section 4 we showcase the ability of the proposed algorithm to sample multimodal distributions in the presence of unidentifiable manifolds in a Pharmacodynamics model.

2. Background. We begin with an overview of Bayesian inference with sampling by the transitional Markov chain Monte Carlo (TMCMC) algorithm [10] and the manifold Metropolis adjusted Langevin algorithm (mMALA) [16].

2.1. Bayesian inference. Given a model $f(x; \varphi)$, with $x \in \mathbb{R}^N_x$ the input vector and $\varphi \in \mathbb{R}^N_\varphi$ the parameter vector, and a data set $D = \{d_i | i = 1, \ldots, N_D\}$, our goal is to find parameters $\varphi$ such that $f(x; \varphi)$ is a good approximation to the observations $D$. The Bayes theorem provides a distribution of the model parameters conditioned on the data according to

$$p(\varphi | D) = \frac{p(D | \varphi) p(\varphi)}{p(D)}.$$  

The prior distribution $p(\varphi)$ encodes the available information on the parameters prior to observing any data. The denominator, $p(D)$, is referred to as the evidence of the data and is used for model selection [2]. Under the assumption that the data are independent and normally distributed around the output of the model, we postulate that

$$d_i = f(x_i; \varphi) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_n),$$

the likelihood function $p(D | \vartheta)$ takes the form

$$p(D | \vartheta) = \mathcal{N}(D | F(X, \varphi), \sigma_n I),$$

where $\vartheta = (\varphi^T, \sigma_n)^T$ is the parameter vector that contains both the model and the noise parameters, and $F(X, \varphi) = (f(x_1; \varphi), \ldots, f(x_{N_D}; \varphi))$. The assumption that the observations are independent is used here in order to simplify the presentation, and correlations between the observations can be included without changing the general formulation presented here.
2.2. Transitional Markov chain Monte Carlo (TMCMC). The TMCMC algorithm is a population based algorithm for sampling from a sequence of intermediate distributions controlled by the annealing scheme

\[ p_j(\vartheta) \propto p(D|\vartheta)^{\zeta_j} p(\vartheta) \]

for \( j = 1, \ldots, m \) and \( 0 = \zeta_1 < \cdots < \zeta_m = 1 \) that converges to the posterior distribution \( p(\vartheta|D) \propto p(D|\vartheta)p(\vartheta) \) when \( \zeta_m = 1 \).

The algorithm first draws \( N_1 \) samples from the prior distribution. At the \( j + 1 \) stage it uses \( N_j \) samples from the distribution \( p_j \) to obtain \( N_{j+1} \) samples from the distribution \( p_{j+1} \). Let \( \Theta_j = \{ \vartheta_{j,k}|k = 1, \ldots, N_j \} \) be the samples obtained at the \( j \)th step from \( p_j \). The following procedure gives samples from \( p_{j+1} \):

1. Draw \( N_{j+1} \) samples from the set \( \Theta_j \) with probability of the sample \( \vartheta_{j,k} \) to be selected equal to

\[ \hat{w}_{j,k} = \frac{w_{j,k}}{\sum_{k=1}^{N_j} w_{j,k}} , \]

where \( w_{j,k} = p(D|\vartheta_{j,k})^{\zeta_{j+1} - \zeta_j} \). Put the new samples in the set \( \tilde{\Theta}_{j+1} \) and set

\[ S_j = \frac{1}{N_j} \sum_{k=1}^{N_j} w_{j,k} . \]

2. For each sample in \( \tilde{\Theta}_{j+1} \) perform MCMC with Gaussian proposal distribution and covariance matrix \( \varepsilon^2 \Sigma_s^{(j)} \). Here, \( \varepsilon \) is a scaling parameter and \( \Sigma_s^{(j)} \) is the sample covariance at the \( j \)th stage given by

\[ \Sigma_s^{(j)} = \sum_{k=1}^{N_j} \hat{w}_{j,k}(\vartheta_{j,k} - \bar{\vartheta}_j)(\vartheta_{j,k} - \bar{\vartheta}_j)^\top , \]

where \( \bar{\vartheta}_j = \sum_{k=1}^{N_j} \hat{w}_{j,k} \vartheta_{j,k} \). Set the chain length equal to a predefined parameter \( \ell_{\text{max}} \).

The pseudocode of the algorithm is presented in Algorithm 1. In [40] it was suggested that the MCMC step, described in lines 11–14 of Algorithm 1, results in a bias accumulated in each stage. In the original TMCMC the MCMC is performed for each unique sample in \( \tilde{\Theta}_{j+1} \) with chain length equal to the number of occurrences of the sample. In [40] it is shown that in order for the bias to be reduced, all samples in \( \tilde{\Theta}_{j+1} \) should perform an MCMC step with chain length equal to a parameter \( \ell_{\text{max}} \), which is usually set to 1. The improved algorithm is called BASIS, and an efficient implementation can be found in the II4U framework [19].

A key advantage of TMCMC is that it can be efficiently parallelized since the likelihood evaluation is independent for each sample. An additional computational benefit introduced by the BASIS algorithm is that all MCMC chains have equal length and thus the work load can be balanced among the processors. Moreover, an important byproduct of the algorithm is that the evidence of the data is estimated by

\[ p(D) \approx \prod_{j=1}^{m-1} S_j , \]

where \( m \) is the total number of stages in the TMCMC algorithm [10]. In fact, (8) is an unbiased estimator of the evidence [13]. The idea of estimating the evidence of the
Algorithm 1 BASIS (TMCMC)

1: **Input:** Likelihood function \( p(D|\theta) \), prior distribution \( p(\theta) \)

\( N_j, N_{max} \) – number of samples per stage, maximum number of stages

\( \gamma, \varepsilon \) – threshold parameter, scale parameter

2: **Output:** \( \Theta_{final} \) – a set of samples from \( p(\theta|D) \)

\( S \) – estimation for the evidence \( p(D) \)

3: Draw initial sample set \( \Theta_1 = \{ \theta_{1,k} | k = 1, \ldots, N_1 \} \) from prior

4: Initialize \( j \leftarrow 1, \zeta_1 \leftarrow 0, S \leftarrow 1 \)

5: **repeat**

6: Choose \( \zeta_{j+1} \) such that the coefficient of variation of \( w_{j,k} < \gamma \) and \( \zeta_{j+1} \leq 1 \)

7: Calculate \( w_{j,k} \) with the chosen \( \zeta_{j+1} \)

8: \( S \leftarrow S \cdot \frac{1}{N_j} \sum_{j=1}^{N_j} w_{j,k} \)

9: Obtain \( \tilde{\Theta}_{j+1} \) by drawing \( N_{j+1} \) samples from the set \( \Theta_j \) with probabilities \( \propto w_{j,k} \)

10: Set \( \Sigma \) the weighted covariance given by (7)

11: for each sample in \( \tilde{\Theta}_{j+1} \) do

12: Perform MCMC with length equal to \( \ell_{max} \) and proposal distribution \( q(\cdot|\theta) = \mathcal{N}(\cdot|\theta, \varepsilon^2 \Sigma) \)

13: Add resulting samples in \( \Theta_{j+1} \)

14: end for

15: \( j \leftarrow j + 1 \)

16: **until** \( \zeta_j = 1 \) or \( j > N_{max} \)

17: \( \Theta_{final} \leftarrow \Theta_j \)

data by (8) can also be found in [8, 27] in the context of thermodynamic integration.
The main difference between these algorithms and TMCMC is that in TMCMC the annealing schedule \( \{ \zeta_j | j = 1, \ldots, m \} \) is estimated adaptively according to the scheme described in line 6 of Algorithm 1. In the thermodynamic integration the annealing schedule is chosen a priori.

Finally, we note that the parameter \( \varepsilon^2 \) was proposed in [10] to be set equal to 0.04. In [5] the authors propose an adaptive choice of \( \varepsilon^2 \) such that a predefined acceptance rate is achieved. We return to the issue of choosing \( \varepsilon^2 \) in subsection 3.2.

### 2.3. Manifold Metropolis adjusted Langevin algorithms.

Let \( \pi: \Theta \rightarrow \mathbb{R} \) be a probability distribution function where \( \Theta \subset \mathbb{R}^{N_\theta} \). The Metropolis–Hastings (MH) algorithm is a Markov chain Monte Carlo (MCMC) method for obtaining samples from \( \pi \) according to the following iterative scheme, starting from sample \( \theta_0 \):

1. propose a sample \( \theta^* \) according to a probability function \( q(\theta^*|\theta_j) \),
2. set \( \theta_{j+1} = \theta^* \) with probability

\[
\alpha(\theta^*|\theta_j) = \min \left( 1, \frac{\pi(\theta^*)q(\theta_j|\theta^*)}{\pi(\theta_j)q(\theta^*|\theta_j)} \right),
\]

and \( \theta_{j+1} = \theta_j \) with probability \( 1 - \alpha(\theta^*|\theta_j) \).

In the limit, the samples obtained using the MH algorithm will be distributed according to \( \pi \). The proposal distribution \( q \) is usually chosen to be a Gaussian distribution centered at \( \theta_j \) with covariance matrix \( \sigma^2 I \), i.e., \( q(\cdot|\theta_j) = \mathcal{N}(\cdot|\theta_j, \sigma^2 I) \) and \( I \) the identity matrix in \( \mathbb{R}^{N_\theta} \). The parameter \( \sigma \) needs to be tuned depending on the probability...
If $\sigma$ is too small, the proposals will be local and the chain will not be able to efficiently explore the parameter space, and if $\sigma$ is too big, there will be many rejected samples leading to slow convergence.

An improved proposal scheme is based on the observation that the random variable that satisfies the stochastic differential equation (SDE)

$$d\theta_t = \frac{1}{2} \nabla \log \pi(\theta) \, dt + dW_t,$$

where $W_t$ is an $N_\theta$ dimensional Wiener process, has $\pi$ as stationary distribution. Then, the Euler–Maruyama discretization is given by

$$\theta_{n+1} = \theta_n + \varepsilon \frac{1}{2} \nabla \log \pi(\theta_n) + \sqrt{\varepsilon} W_n, \quad W_n \sim \mathcal{N}(0, I),$$

where $\varepsilon$ is the time step of the discretization. Since $\varepsilon$ introduces error, $\pi$ is no longer the equilibrium distribution of $\theta_n$ and thus cannot be sampled directly by solving (10). Instead, the sample $\theta_{n+1}$ is used as a proposal in the MH algorithm. In other words, the proposal distribution in the MH algorithm is given by

$$q(\cdot|\theta_n) = \mathcal{N}\left(\cdot | \theta_n + \varepsilon \frac{1}{2} \nabla \log \pi(\theta_n), \varepsilon I\right).$$

This scheme is known as the Metropolis adjusted Langevin algorithm (MALA). Notice that the original MH algorithm with Gaussian proposal distribution corresponds to the MALA algorithm for the SDE

$$d\theta_t = \varepsilon dW_t,$$

which corresponds to an isotropic diffusion in $\mathbb{R}^{N_\theta}$ and is usually referred to as the random walk MH algorithm (RWMH).

Although the proposals based on (10) follow the direction of the maximum change, guiding $\theta$ to regions of high probability, the isotropic diffusion may be inappropriate in the presence of highly correlated random variables. The following SDE offers a better proposal scheme:

$$d\theta_t = \frac{1}{2} \Sigma \nabla \log \pi(\theta) \, dt + \sqrt{\Sigma} dW_t,$$

where the correlation of the variables is encoded in the constant and positive definite matrix $\Sigma$. The algorithm is known as the preconditioned MALA [34]. A variation of the preconditioned MALA [16] is based on the following SDE with position dependent covariance matrix:

$$d\theta_t = \frac{1}{2} G^{-1}(\theta_t) \nabla \log \pi(\theta_t) dt + \Omega(\theta_t) dt + G^{-\frac{1}{2}}(\theta_t) dW_t,$$

where

$$\Omega_t(\theta_t) = |G(\theta_t)|^{-\frac{1}{2}} \sum_j \frac{\partial}{\partial \theta_j} \left[ G_{i,j}^{-1}(\theta_t) |G(\theta_t)|^\frac{1}{2} \right],$$

and $G$ is a positive definite matrix. Equation (15) describes diffusion in a manifold, defined in local coordinates by $G$ [16], and the resulting MCMC algorithm is called manifold MALA (mMALA). In [41] it is shown that the correct form of $\Omega$ should be

$$\Omega_t(\theta_t) = \frac{1}{2} \sum_j \frac{\partial}{\partial \theta_j} \left[ G_{i,j}^{-1}(\theta_t) \right] = \frac{1}{2} \sum_j \left[ G^{-1}(\theta_t) \frac{\partial G(\theta_t)}{\partial \theta_j} G^{-1}(\theta_t) \right]_{i,j}.$$
where (16) and (17) describe equivalent diffusions under the condition
\[ \partial_{\theta_j} G_{k,m}(\theta) = \partial_{\theta_k} G_{j,m}(\theta). \]
The authors refer to the resulting sampling scheme as position dependent MALA (pMALA). A simplified version, under the assumption that the manifold has constant curvature, can be obtained by setting \( \Omega = 0 \) leading to proposals,
\[
q(\cdot | \theta) = N\left( \cdot | \theta + \frac{\varepsilon}{2} G^{-1}(\theta) \nabla \log \pi(\theta), \varepsilon G^{-1}(\theta) \right).
\]
The resulting scheme is known as simplified manifold MALA (smMALA) [16].

However, there are two remaining questions:
1. What is the optimal choice for \( G \)?
2. How should the parameter \( \varepsilon \) be chosen?

An optimal scaling of \( \varepsilon \) with respect to the dimension of \( \theta \) for various MH schemes (MALA included) has been derived [34], while an optimal scaling for a class of mMALA algorithms has been provided for a wide class of distributions [7]. We return to this issue in subsection 3.2, where we discuss a heuristic procedure for the automatic tuning of \( \varepsilon \) in the framework of TMCMC.

In order to answer the first question, we consider a specific form for \( \pi \) as a posterior of the Bayesian inference problem, i.e., \( \pi(\theta) = p(\theta|D) \propto p(D|\theta) p(\theta) \). Then there are two widely used choices: the negative of the Hessian of \( \log p(D, \theta) \),
\[
G(\theta) = H(\theta) := -\frac{\partial^2}{\partial \theta^2} \log p(D, \theta)
\]
(19)
and the Fisher information of \( p(D, \theta) \),
\[
G(\theta) = I(\theta) := -\mathbb{E}_{D|\theta} \left[ \frac{\partial^2}{\partial \theta^2} \log p(D, \theta) \right]
\]
(20)
which is the Fisher information of the likelihood function minus the Hessian of the prior distribution. In subsection 3.1 we discuss the most suitable choice of \( G \).

Finally, in [29] a similar proposal distribution to (18) is proposed, where a new point is proposed according to
\[
\theta_{n+1} = \theta_n + H^{-1}(\theta_n) \nabla \log \pi(\theta_n) + H^{-\frac{1}{2}}(\theta_n) W_n, \quad W_n \sim \mathcal{N}(0, I) .
\]
The method is called stochastic Newton MCMC and compared to (18) \( G = H, \varepsilon = 1, \) and the \( \frac{1}{2} \) factor has been dropped.

3. Manifold transitional Markov chain Monte Carlo. In the TMCMC algorithm, the proposal distribution at the \( j \)th stage at point \( \theta \) is a Gaussian density function centered at \( \theta \) with a constant covariance matrix, i.e., \( \theta' \sim \mathcal{N}(\theta, \varepsilon^2 \Sigma_s^{(j)}) \), where \( \Sigma_s^{(j)} \) is the weighted sample covariance matrix at stage \( j \) given by (7).

Here, we improve the quality of TMCMC samples by using a proposal scheme based on the mMALA algorithms discussed in subsection 2.3.

In the \( j \)th stage of TMCMC, samples from the distribution
\[
p_j(\theta) = \frac{p(D|\theta)^{\gamma_j} p(\theta)}{p_j(D)}
\]
(22)
must be collected based on some proposal. In order to use the proposal scheme (15), the gradient of \( \log p_j \) should be computed:

\[
(23) \quad \nabla \log p_j(\vartheta) = \gamma \nabla \log p(\mathcal{D}|\vartheta) + \nabla \log p(\vartheta).
\]

For the rest of the presentation we restrict the discussion to uniform prior distributions; thus the last term in (23), as well as the last term in \( G(\vartheta) \) in (19) and (20), vanishes. In this case the diffusion (15) is written as

\[
(24) \quad d\vartheta_t = \frac{\gamma}{2} \Sigma \nabla \log p(\mathcal{D}|\vartheta_t) dt + \Omega(\vartheta_t) dt + \sqrt{\Sigma(\vartheta_t)} dW_t,
\]

where \( \Sigma = G^{-1} \) with \( G \), the metric corresponding to \( p^\vartheta(\mathcal{D}|\vartheta) \) and \( \Omega \), is given by (17) with \( G \) substituted by \( G_\gamma \). In general, the matrix \( G_\gamma \) may not be invertible or positive definite. We define \( \Sigma \) as the pseudo covariance matrix, and we discuss this issue in more detail in section 3.1. Note that under the modeling assumption (3) and with \( G \) given by (19) and (20), it can be shown that

\[
(25) \quad \Sigma(\vartheta) = \gamma^{-1} \Sigma(\vartheta) \quad \text{and} \quad \Omega(\vartheta) = \gamma^{-1} \Omega(\vartheta).
\]

Moreover, the gradient of the log-likelihood function is given by

\[
(26) \quad \frac{\partial}{\partial \vartheta_k} \log p(\mathcal{D}|\vartheta) = \left\{ \begin{array}{ll}
\sigma_n^{-2} \sum_{i=1}^{N_d} (d_i - f_i) \frac{\partial}{\partial \vartheta_k} f_i, & k = 1, \ldots, N_\theta - 1, \\
-N_d \sigma_n^{-1} + \sigma_n^{-3} \sum_{i=1}^{N_d} (d_i - f_i)^2, & k = N_\theta,
\end{array} \right.
\]

where \( f_i = f(x_i; \varphi) \) and the Hessian is given by

\[
(27) \quad \frac{\partial^2}{\partial \vartheta_k \partial \vartheta_\ell} \log p(\mathcal{D}|\vartheta) = \left\{ \begin{array}{ll}
\sigma_n^{-2} \sum_{i=1}^{N_d} (d_i - f_i) \left( \frac{\partial}{\partial \vartheta_k} \frac{\partial}{\partial \vartheta_\ell} f_i - \frac{\partial}{\partial \vartheta_k} f_i \frac{\partial}{\partial \vartheta_\ell} f_i, \right), & k, \ell = 1, \ldots, N_\theta - 1, \\
-2\sigma_n^{-3} \sum_{i=1}^{N_d} (d_i - f_i) \frac{\partial^2}{\partial \vartheta_k \partial \vartheta_\ell} f_i, & k = 1, \ldots, N_\theta - 1, \ell = N_\theta,
\end{array} \right.
\]

\[
-3\sigma_n^{-4} \sum_{i=1}^{N_d} (d_i - f_i)^2, & k = N_\theta.
\]

The computation of the gradient and the Hessian of \( p(\mathcal{D}|\vartheta) \) involves computation of the derivatives of the observable function \( f \). We discuss the details of this computation in Appendix A.

### 3.1. Choices and corrections for the pseudo covariance matrix.

We consider two possible choices of the metric \( G \)—the Hessian and the Fisher information, defined in (19) and (20), respectively. These choices are associated with three problems:

(a) \( G \) is not invertible.

(b) \( G \) is invertible with some negative eigenvalues; thus \( G^{-1} \) is not positive definite.

(c) \( G \) is invertible and \( G^{-1} \) is positive definite, but some eigenvalues of \( G \) are very small, respectively, some eigenvalues of \( G^{-1} \) are very large. This results in a Gaussian proposal distribution that lies largely outside the bounds specified in the prior distribution.
The first problem is addressed by setting \( G = \Sigma_s^{(j)} \), the sample covariance matrix at the \( j \)th stage of TMCMC (see (7)). The second problem emerges only when \( G = \mathcal{H} \) since the Fisher information is always positive semidefinite. The presence of close to zero eigenvalues is treated in the third case. One possible solution to this is to set \( G = \Sigma_s^{(j)} \) as in the noninvertible case. This fix disregards all the information that is contained in the directions with negative eigenvalues. Past proposals for fixing the non–positive definiteness of the covariance matrix include the SoftAbs method [4] and a modified Cholesky decomposition [25]. Here, we follow the approach used in [29] and substitute the negative eigenvalues of \( G^{-1} \) with a predefined positive number chosen to be equal to the smallest eigenvalue of \( \Sigma_s^{(j)} \).

The third problem appears in the case of unidentifiable parameters or parameter combinations. The likelihood of the data stays constant for these parameters and thus all derivatives are zero. In many practical situations, one may encounter the situation where computational unidentifiability appears [31]. In this case the problem is identifiable in a small region of the probability space but close to unidentifiable in a large region of the parameter space. Unidentifiable directions correspond to zero, or close to zero, eigenvalues in the Hessian or the Fisher information, leading to large eigenvectors in the Gaussian proposal distribution. It is important to note here that this entire discussion is particular to the case of a uniform prior; for an in-depth consideration of the role of noninformed directions in the case of a Gaussian prior the reader is referred to the works [3, 11, 12].

We treat the problems arising from the large eigenvalues as follows: Let \( \Sigma = Q \Lambda Q^\top \in \mathbb{R}^{N_\vartheta \times N_\vartheta} \) be the eigendecomposition of the covariance matrix of a normal distribution centered at the \( \vartheta^* \), with \( \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_{N_\vartheta}) \), \( \lambda_i \in \mathbb{R} \), the eigenvalues of \( \Sigma \) and \( Q \) a square matrix whose columns \( q_i \in \mathbb{R}^{N_\vartheta} \) are the eigenvectors corresponding to \( \lambda_i \). Then, for \( \eta \in (0,1) \) the \( 1 - \eta \) of the probability mass lies inside the ellipsoid described by the equation

\[
(\vartheta - \vartheta^*)^\top Q \Lambda^{-1} Q^\top (\vartheta - \vartheta^*) = \chi^2_{N_\vartheta}(\eta), \quad \vartheta \in \mathbb{R}^{N_\vartheta},
\]

where \( \chi^2_{N_\vartheta}(\eta) \) is the upper 100\( \eta \)th percentile of the \( \chi^2 \) distribution with \( N_\vartheta \) degrees of freedom [38]. The semiaxes of the ellipsoid (28) are given by \( \sqrt{\lambda_i \chi^2_{N_\vartheta}(\eta)} q_i \). The idea is that the eigenvalues that lead to large semiaxes will be adapted such that the ellipsoid will lie inside the prior domain. We have observed that this approach leads to small eigenvalues near the boundaries, and thus the proposal distribution may be concentrated near the boundaries. We resolve this issue by adapting the semiaxes to an extended boundary that is defined as a percentage \( \rho \in [0,1] \) of the length of the boundaries of the prior probability.

We adapt the large eigenvalues of \( \Sigma \) by finding constants \( c_i \in \mathbb{R} \) such that the points

\[
\hat{p}_i^\pm = \vartheta^* \pm \sqrt{\lambda_i \chi^2_{N_\vartheta}(\eta)} q_i, \quad \hat{\lambda}_i = c_i \lambda_i, \quad i = 1, \ldots, N_\vartheta,
\]

lie inside the extended bounds of the prior distribution. Let \( \alpha, \beta \in \mathbb{R}^{N_\vartheta} \) be the vectors that define the uniform prior distribution, i.e., \( \alpha_i \leq \vartheta_i \leq \beta_i \), and \( J_{i,\alpha}^\pm \) and \( J_{i,\beta}^\pm \) the sets of indices that violate the inequalities

\[
(1 - \rho) \alpha_j \leq p_{i,j}^\pm \quad \text{and} \quad p_{i,j}^\pm \leq (1 + \rho) \beta_j,
\]

respectively, where \( p_{i,j}^\pm = \vartheta^* \pm \sqrt{\lambda_i \chi^2_{N_\vartheta}(\eta)} q_i, \quad i = 1, \ldots, N_\vartheta \). Next, we define the con-
Fig. 1. Illustration of the eigenvalue adaptation using the extended boundary in order to treat large eigenvectors of the covariance matrix.

The advantages of the extended boundaries approach, as well as the selection of the parameter $\rho$, are presented in a truncated multivariate Gaussian in subsection 4.1.

3.2. Sampling schemes. In this section we summarize the proposal schemes that we implement within the TMCMC algorithm. For the proposal distribution we use the unified notation

$$
q(\vartheta | \vartheta^*) = \mathcal{N}(\vartheta | D(\vartheta^*; \varepsilon, \gamma), C(\vartheta^*; \varepsilon, \gamma))
$$

with $D$ and $C$ the mean and the covariance matrix of the Gaussian proposal distribution, respectively. In the original TMCMC we have $D(\vartheta^*; \varepsilon, \gamma) = \vartheta^*$ and $C(\vartheta^*; \varepsilon, \gamma) = \varepsilon \Sigma^{(j)}$; see (7). Since $\Sigma^{(j)}$ is a constant matrix, the proposal distribution is symmetric and the acceptance ratio is independent of $q$. 

$$
c_i = \min \left\{ c_{i,\alpha}^+, c_{i,\alpha}^-, c_{i,\beta}^+, c_{i,\beta}^- \right\}
$$

for $i = 1, \ldots, N_\vartheta$. Finally, the correction constant for the $i$th eigenvalue is given by

$$
c_i = \min \left\{ c_{i,\alpha}^+, c_{i,\alpha}^-, c_{i,\beta}^+, c_{i,\beta}^- \right\}.
$$

We will denote the corrected covariance by $\hat{\Sigma} = \hat{Q} \hat{\Lambda} \hat{Q}^T$ and $\hat{\Lambda} = C \Lambda$ with $C = \text{diag}(c_1, \ldots, c_{N_\vartheta})$. In all the numerical tests of section 4 we have set $\eta = 0.3$. For an illustration of this approach see Figure 1, where the eigenvalue in the direction of $p_1^+$ has been adapted such that $p_1^+$ will lie inside the extended boundary domain. Notice that the eigenvalues in the direction of $p_2^\pm$ have not been changed.

The advantages of the extended boundaries approach, as well as the selection of the parameter $\rho$, are presented in a truncated multivariate Gaussian in subsection 4.1.
The simplified manifold TMCMC (smTMCMC) (18) is obtained by assuming that the metric $G_\gamma$ is locally constant. This assumption leads to $\Sigma_\gamma(\vartheta) = \Sigma_\gamma(\vartheta^*)$ and $\Omega_\gamma(\vartheta) = 0$. The pseudo covariance matrix $\Sigma_\gamma(\vartheta)$ is substituted by $\hat{\Sigma}_\gamma(\vartheta)$, the corrected covariance according to the scheme presented in subsection 3.1. Finally, the mean and the variance of the proposal distribution are given by $D(\vartheta; \epsilon, \gamma) = \vartheta + 2\epsilon \hat{\Sigma}_\gamma(\vartheta) \nabla \log p_D(\vartheta)$ and $C(\vartheta; \epsilon, \gamma) = \epsilon \Sigma_\gamma(\vartheta)$, respectively. Notice that the proposal distribution is not symmetric. This asymmetry must be accounted for in the calculation of the acceptance ratio (see Table 2).

Without the assumption of a locally constant metric, the term $\Omega_\gamma$ is not necessarily zero. After correcting the pseudo covariance matrix $\Sigma_\gamma$, the $\Omega$ term is written as

\[
\hat{\Omega}_\gamma,i(\vartheta) = -\frac{1}{2} \sum_j \left[ \hat{\Sigma}_\gamma(\vartheta) \frac{\partial \hat{G}_\gamma(\vartheta)}{\partial \vartheta_j} \hat{\Sigma}_\gamma(\vartheta) \right]_{i,j},
\]

with $\hat{G}_\gamma = \hat{\Sigma}^{-1}$. In order to express the unknown derivative of $\hat{G}$ in terms of the known derivative of $G$, we follow the same procedure as in [4] (see also Appendix B):

\[
\frac{\partial \hat{G}_\gamma}{\partial \vartheta_j} = Q \left( J \circ (Q^{-1} \frac{\partial G_\gamma}{\partial \vartheta_j} Q) \right) Q^T,
\]

where $\circ$ denotes the Hadamard product and

\[
J_{i,j} = \begin{cases} 
\frac{\lambda_i - \lambda_j}{\lambda_i - \lambda_j}, & i \neq j, \\
\frac{\partial \lambda}{\partial x_i} = c_i, & i = j.
\end{cases}
\]

We call the resulting algorithm position dependent TMCMC (pTMCMC).

**Tuning the scale parameter.** The parameter $\epsilon$ is tuned depending on the sampling algorithms. A usual practice is to vary $\epsilon$ until the acceptance rate of the algorithm reaches a desired acceptance rate. Following the results in [36], we set the target acceptance rate for TMCMC, which is based on random walk MH (RWMH), equal to 0.234 and for the Langevin diffusion based TMCMC equal to 0.574. It can be shown that $\epsilon$ scales like $N_\gamma^{-1}$ for RWMH and like $N_\gamma^{-1/3}$ for MALA [36] as $N_\gamma \to \infty$. One way to find the optimal scaling is to run some preparatory, relatively small, TMCMC simulations with various $\epsilon$ and choose the one that gives the optimal acceptance rate.

An adaptive method to find the optimal scaling is proposed in [5] in the framework of TMCMC. In this method the sample sets at the $j$th stage of TMCMC are divided into subsets. These subsets are run sequentially, and information on the mean acceptance rate is passed to the next subset. The scaling parameter is tuned depending on the distance between the mean acceptance rate and the target acceptance rate.

**Choosing the metric.** In the aforementioned sampling schemes either the Hessian or the Fisher information is used as the underlying metric. Note that the Hessian needs the computation of second order derivatives, while the Fisher information requires only first order derivatives. Moreover, since the Fisher information is always positive semidefinite, we expect it to perform better than the Hessian. However, in some cases the Fisher information is not explicitly known, e.g., in the case of Gaussian mixtures, and thus the use of the Hessian is inevitable. Finally, in the pTMCMC scheme, where the computation of $\Omega$ is involved, the order of the needed derivatives is increased by one. Thus, for $G = \mathcal{H}$ and $G = \mathcal{I}$, third and second order derivatives,
respectively, should be computed. In order to avoid the computation of third order derivatives, we choose not to implement the Hessian in this sampling scheme.

### Table 1

**Mean and covariance of the proposal distribution;** \( q(\vartheta | \vartheta^*) = \mathcal{N}(\vartheta | D(\vartheta^*; \varepsilon, \gamma), C(\vartheta^*; \varepsilon, \gamma)) \) for the TMCMC algorithms described in subsection 3.2.

<table>
<thead>
<tr>
<th>Name</th>
<th>Mean, ( D(\vartheta^*; \varepsilon, \gamma) )</th>
<th>Covariance, ( C(\vartheta^*; \varepsilon, \gamma) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMCMC</td>
<td>( \vartheta^* )</td>
<td>( \varepsilon \Sigma^j )</td>
</tr>
<tr>
<td>smTMCMC</td>
<td>( \vartheta^* + \frac{\gamma}{2} \Sigma^j \nabla \log p(D</td>
<td>\vartheta^*) )</td>
</tr>
<tr>
<td>pTMCMC</td>
<td>( \vartheta^* + \frac{\gamma}{2} \Sigma^j \nabla \log p(D</td>
<td>\vartheta^*) + \varepsilon \Sigma^j \nabla \log p(D</td>
</tr>
</tbody>
</table>

### Table 2

**Acceptance ratio** \( \alpha(\vartheta|\vartheta^*) = \min(1, A(\vartheta|\vartheta^*)) \) for the proposal distribution of the TMCMC algorithms described in subsection 3.2.

| Name          | Acceptance ratio, \( A(\vartheta|\vartheta^*) \) |
|---------------|-----------------------------------------------|
| TMCMC         | \( \frac{p(D|\vartheta)}{p(D|\vartheta^*)} \) |
| smTMCMC       | \( \frac{p(D|\vartheta) \mathcal{N}(\vartheta^* | D(\vartheta), C(\vartheta))}{p(D|\vartheta^*) \mathcal{N}(\vartheta | D(\vartheta^*), C(\vartheta^*))} \) |
| pTMCMC        | \( \frac{p(D|\vartheta) \mathcal{N}(\vartheta^* | D(\vartheta), C(\vartheta))}{p(D|\vartheta^*) \mathcal{N}(\vartheta | D(\vartheta^*), C(\vartheta^*))} \) |

The performance of the proposed algorithms is discussed in detail in section 4. In Table 1 we present a summary of the proposal distribution and in Table 2 the resulting acceptance ratio for the various sampling schemes.

### 4. Applications.

We demonstrate the capabilities of the present algorithm on a number of benchmark problems and on a challenging Pharmacodynamics model that is calibrated using clinical data.

We first test our algorithm in a truncated Gaussian distribution. This test demonstrates the need of the extended boundary approach discussed in subsection 3.1. In the second example the scaling of the error of smTMCMC is compared to that of TMCMC in a multivariate Gaussian distribution. Finally, we sample the posterior distribution of a Bayesian inference problem on the parameters of the Pharmacodynamics model. The challenging part of sampling this distribution is that it has at least one large manifold of nonidentifiable parameters. In this example the TMCMC algorithm completely fails to explore the parameter space, while the smTMCMC algorithm provides good quality samples. We note that in our experiments the differences between smTMCMC and pTMCMC are indistinguishable, and thus only results from smTMCMC are reported. The provided code also includes the pTMCMC algorithm, which employs as diffusion metric the Fisher information.

#### 4.1. Truncated Gaussian distribution.

Here, we illustrate the ability of the proposed algorithm to correctly sample distributions with mass concentrated at the
Fig. 2. Marginal histograms of the sampled compared with the exact (red line) truncated Gaussian distribution (37). The sampling algorithm is the smMALA algorithm with no extended boundaries. Notice that the sampling algorithm incorrectly concentrates samples near the boundaries of the prior.

boundaries of the prior distribution. The strategy of adapting the eigenvalues of the covariance matrix of the proposal distribution by extending the boundaries of the prior distribution in subsection 3.1 is compared with the $\rho = 0$ adaptation where the eigenvalues are adapted to the boundaries of the prior. For $\rho = 0$, the covariance matrix after the correction, $\hat{\Sigma}$, at samples near the boundaries will have some very small eigenvalues, and the MCMC chain starting at these points will be locally trapped.

We consider the truncated Gaussian probability distribution

$$p(\vartheta) = \prod_{i=1}^{4} \mathcal{N}(\vartheta_i | \mu_i, \sigma_i^2) \mathcal{U}(\vartheta_i | 0, 10),$$

with $\mu = (0, 5, 10, 9)$ and $\sigma^2 = (0.05, 0.5, 2, 5)$. The marginal distribution of $\vartheta_i$ is depicted with a solid line in Figure 2. The histograms in Figure 2 are obtained with the smMALA algorithm, and $G$ is the Fisher information matrix of (37) using 500 samples and $\varepsilon = 1$. It is evident from this example that the samples are concentrated at the boundaries of the prior distribution where there is significant mass of the Gaussian distribution. Notice that the variable $\vartheta_2$, which is distributed away from the boundaries, is sampled correctly.

Next, we sample (37) using smMALA with extended boundaries, and we study the effect of the parameter $\rho$ on the accuracy of the sampling. Let $\tilde{p}$ be the estimated probability distribution using a sampling algorithm and $p$ the target distribution (37). We measure the accuracy using the relative entropy or Kullback–Leibler (KL) divergence of $p$ from $\tilde{p}$,

$$D_{KL}(\tilde{p} \| p) = \int_{-\infty}^{\infty} \tilde{p}(\vartheta) \log \frac{\tilde{p}(\vartheta)}{p(\vartheta)} \, d\vartheta,$$

with $p$ absolutely continuous with respect to $\tilde{p}$; i.e., $p(\vartheta) = 0$ implies $\tilde{p}(\vartheta) = 0$. The KL divergence is not a metric, since it is not symmetric and does not obey the triangle
inequality, but it is a measure of loss of information if \( p \) is used instead of \( \hat{p} \). The reason we consider \( D_{KL}(\hat{p} \parallel p) \), and not \( D_{KL}(p \parallel \hat{p}) \), is that \( \hat{p} \) (the estimated probability) is always absolutely continuous with respect to \( p \) (the target distribution). Since \( \vartheta_i \) are independent, the KL divergence reduces to the sum of KL divergence of the marginal distributions.

![Figure 3](image_url)

**Fig. 3.** (a) Estimated relative entropy (KL divergence) between the estimated and the truncated Gaussian distribution of (37) as a function of the parameter \( \rho \) for the smTMCMC algorithm. (b) The same quantity as a function of the sample size of the sampling algorithm for TMCMC, smTMCMC with \( \rho = 0, 0.2, 0.5, 1 \), and smTMCMC with no correction.

In Figure 3a the averaged KL divergence over 100 independent samplings (each with 500 samples) is plotted as a function of \( \rho \) and is depicted with bars and a scale that corresponds to the left y-axis. For \( \rho = 0 \) the divergence reaches its maximum, while for \( \rho \) near 0.2 the divergence is minimized. For \( \rho \) larger than 0.3 the divergence slightly increases but remains low. As expected, the mean acceptance rate over the stages of smTMCMC, depicted with filled dots in Figure 3a and a scale that corresponds to the right y-axis, drops as \( \rho \) increases.

In Figure 3b the averaged KL divergence over 200 independent samplings is plotted as a function of the number of samples of the sampling algorithm. The KL divergence of TMCMC and smTMCMC with \( \rho = 0 \) shows little improvement as the number of samples increases. This implies that the bias introduced by the eigenvalue adaptation scheme is not reduced with the size of the sample set. On the other hand, smTMCMC with \( \rho = 0.2, 0.5, \) and 1 has lower error than the TMCMC, with both slopes being equal to approximately -1. Finally, smTMCMC with no correction is shown here. As expected, the no correction scheme works well in this case as there are no nonidentifiable directions. Moreover, the performance of the correction scheme is similar to that with no correction. Confidence intervals are not presented because they are smaller than the size of the markers in the plot.

4.2. Multidimensional Gaussian distribution. In this section we present the scaling of the sampling error for the TMCMC and smTMCMC algorithms in a Gaussian distribution and a bimodal Gaussian mixture. In the first test case, the mean \( \mu \) is zero and the covariance matrix \( \Sigma \) of the target distribution is randomly generated with the MATLAB function `gallery('randcorr',d)`, where \( d \) is the number of dimensions. The error of the sampling algorithm is defined as \( E = \frac{1}{2}(e_1 + e_2) \),
where

\[
e_1 = \frac{1}{d} \sum_{i=1}^{d} |\bar{\mu}_i - \mu_i| \quad \text{and} \quad e_2 = \frac{1}{d^2} \sum_{i,j=1}^{d} |\bar{\Sigma}_{i,j} - \Sigma_{i,j}| ,
\]

and \(\bar{\mu}, \bar{\Sigma}\) are the estimated mean and covariance matrix, respectively. The reported sampling error is averaged over 100 independent simulations using 1000 samples. The initial distribution is a \(d\)-dimensional uniform distribution with support in \([-10, 10]\) for each dimension. We set the scaling parameters \(\varepsilon = 0.04\) for TMCMC and \(\varepsilon = 1\) for smTMCMC. Notice that in this case the Hessian matrix is equal to the Fisher information and equal to the inverse of the covariance matrix of the target distribution.

In Figure 4a the error for \(d = 2, 5, 10, 15, 20\) is presented. Both errors increase with the dimension of the target distribution, while the error of smTMCMC is always below the error of TMCMC. In Figure 5 the error for \(d = 5\) as a function of the number of samples is presented in logarithmic scale. The continuous lines correspond to fitted linear functions, and the rate of convergence is approximately \(-\frac{1}{2}\).

Next, we test the performance of the sampling algorithms in a \(d\)-dimensional bimodal Gaussian mixture distribution with the two modes centered at \(\mu_1 = (-5, \ldots, -5)\) and \(\mu_2 = -\mu_1\) and two equal covariance matrices \(\Sigma_1 = \Sigma_2\) randomly generated as in the previous example. The scaling parameters are the same as in the previous example. Note that the Fisher information is not explicitly known for Gaussian mixtures; hence the Hessian is the only metric we use for smTMCMC. To calculate the errors, we first assign each sample to either one of the modes depending on the shortest Euclidean distance. The total error is defined as the average of the local error on each mode, as defined in (39).

In Figure 4b the error \(E\) averaged over 100 independent simulations is presented. The size of the error bars is comparable to that of the markers and thus is not included in the graph. The sample size used in this example is chosen to be \(N_s = 5000\). The reason it is increased compared to the unimodal test case is that TMCMC is not able to detect both modes of the bimodal distribution with fewer samples. Moreover, with the number of samples fixed, TMCMC is able to detect both modes up to \(d = 8\), while smTMCMC can go up to 10. Lastly, the difference between the TMCMC and smTMCMC error is increased as a function of dimension.
The computation of the Hessian or Fisher information comes with an additional cost, but during our experiments no remarkable runtime differences have been observed due to the small computational intensity of the problem. The maximum relative runtime difference has been detected in high dimensions where smTMCMC runs approximately 10% slower than TMCMC.

A feature of the smTMCMC algorithm is that the number of corrections in the covariance matrix, i.e., the number of times a matrix $G$ was transformed to a matrix $\hat{G}$ as discussed in subsection 3.1, drops from a very high value (approximately 90%) for early stages to almost zero during the last stages of the algorithm. This is expected, since during the early TMCMC stages the target distribution is close to uniform. This leads to close to zero values for the Hessian or the Fisher information matrices, which in turn leads to quite wide proposal distributions that have to be corrected. As TMCMC evolves, the sampled distribution is getting closer to the target distribution and the Hessian or Fisher information has the potential to be improved, at least for locally log-concave functions.

4.3. A Pharmacodynamics model. We apply the proposed algorithms to the Bayesian inference of a model for the growth for adult, diffuse, low-grade gliomas and their drug induced inhibition. We employ the model proposed in [32] and infer its parameters using clinical data from MRI measurements of tumors from different patients. The tumor is composed of proliferative tissue ($P$) and quiescent tissue ($Q$). The treatment, due to the drug administration ($C$), aims to destroy proliferative cells or transform the quiescent cells into damaged quiescent cells ($Q_P$). The damaged quiescent cells may either die or repair their DNA and transform into proliferative
cells. The model consists of a system of ordinary differential equations,

\[
\begin{align*}
\frac{dC}{dt} &= -\varphi_1 C, \\
\frac{dP}{dt} &= \varphi_4 P \left( 1 - \frac{P + Q + Q_p}{K} \right) + \varphi_5 Q_p - \varphi_3 P - \varphi_1 \varphi_2 CP, \\
\frac{dQ}{dt} &= \varphi_3 P - \varphi_1 \varphi_2 CQ, \\
\frac{dQ_p}{dt} &= \varphi_1 \varphi_2 CQ - \varphi_5 Q_p - \varphi_6 Q_p, \\
C(0) &= 0, \quad P(0) = \varphi_7, \quad Q(0) = \varphi_8, \quad Q_p(0) = 0.
\end{align*}
\]

We set \( Y = (C, P, Q, Q_p)^T \), and the quantity of interest is \( f(Y; t; \varphi) = P(t; \varphi) + Q(t; \varphi) + Q_p(t; \varphi) \). Here the correspondence of the parameter vector \( \varphi \) to the parameters used in [32] is \( \varphi = (K D E, \gamma, k_{Q}, \lambda_{P}, k_{Q_p}, \delta_{Q_p}, P_0, Q_0) \).

We process a set of five measurements \( D_k = \{ (t_i, d_i) : i = 1, \ldots, N_d \}, \) \( k = 1, \ldots, 5 \), that correspond to the measurements of the mean tumor diameter \( d_i \) at time instances \( t_i \) of five patients, and a set of time instances of drug administration \( \{ \tau_i : i = 1, \ldots, N_\tau \} \) is provided [32]. At \( t = \tau_i \) we restart the simulation and set \( C(\tau_i) = 1 \). The differential equation system is solved using the MATLAB function \texttt{ode45}. In order to reduce the number of inferred parameters, we assume that at time \( t = 0 \) a measurement, \( d_1 \), is exactly known and thus \( \varphi_8 = d_1 - \varphi_7 \).

The likelihood function is based on the assumption (2) of a normally distributed model error, and the augmented parameter vector is defined as \( \vartheta = (\varphi^\top, \sigma_n)^\top \). We choose a uniform prior distribution for the parameters, \( p(\vartheta) = \prod_{i=1}^{8} U(\vartheta_i | a_i, b_i) \) with \( a = (10^{-2}, 10^{-2}, 10^{-5}, 10^{-5}, 10^{-5}, 10^{-5}, 10^{-5}, 10^{-5}) \) and \( b = (20, 20, 2.5, 0.3, 0.05, 0.6, 1, 33) \).

Since the prior distribution is uniform, the maximum a posteriori probability (MAP) estimate coincides with the maximum likelihood (ML) estimate,

\[
\vartheta_{ML} = \arg \max_{\vartheta} p(\vartheta | D) = \arg \max_{\vartheta} p(D | \vartheta),
\]

where the maximum is taken over all components of \( \vartheta \) (\( \vartheta_j > 0, \ j = 1, \ldots, 8 \)). We optimize the log-likelihood for the different data sets, corresponding to different patients, using the covariance matrix adaptation evolution strategy (CMA-ES) [20]. Then, we draw samples from the posterior distribution and compare the maximum log-likelihood in the set of samples with the ML estimate obtained using CMA-ES. We use this test as an indicator to check whether the sampling algorithm is able to sample the high probability area close to the ML point.

In Table 3 we report the estimated ML by the CMA algorithm and compare with the maximum log-likelihood in the sample set of TMCMC and smTMCMC algorithms. The results of pTMCMC are indistinguishable from those of smTMCMC and thus are not reported here. The TMCMC algorithm is unable to identify the ML, while smTMCMC gets closer to maximum likelihood. Increasing the sample size consistently improves the results of smTMCMC, while this is not true for TMCMC; see, for example, the decrease in the log-likelihood for \( 10^5 \) samples for patients 3 and 5 in Table 3.

The difficulty that the sampling algorithms face in trying to identify the high probability areas suggests that an unidentifiable manifold is present in the posterior...
The maximum log-likelihood estimate for the Pharmacodynamics model (40) obtained using the CMA-ES algorithm, compared with the maximum log-likelihood found in the sample set using the TMCMC and the smTMCMC algorithms.

<table>
<thead>
<tr>
<th>No</th>
<th>CMA-ES</th>
<th>TMCMC $N_s = 1e4$</th>
<th>TMCMC $N_s = 1e5$</th>
<th>smTMCMC $N_s = 1e4$</th>
<th>smTMCMC $N_s = 1e5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-33.06</td>
<td>-38.81</td>
<td>-36.99</td>
<td>-34.89</td>
<td>-33.66</td>
</tr>
<tr>
<td>2</td>
<td>-29.91</td>
<td>-39.94</td>
<td>-35.67</td>
<td>-32.11</td>
<td>-30.96</td>
</tr>
<tr>
<td>5</td>
<td>-15.68</td>
<td>-35.41</td>
<td>-36.23</td>
<td>-19.42</td>
<td>-17.87</td>
</tr>
</tbody>
</table>

probability space. Using the profile log-likelihood (PL) function [31],

$PL(\vartheta_i | D) = \max_{\vartheta \setminus \vartheta_i} p(\vartheta | D),$  

where the notation $\vartheta \setminus \vartheta_i$ implies that we fix the $i$th element and we maximize over the remaining elements of $\vartheta$. We verified this assumption for $i = 1$. Areas of constant PL indicate nonidentifiable parameters, and as shown in Figure 6 the PL function exhibits a large area of unidentifiability for $\vartheta_1 > 2$.

![Figure 6](image.png)

Fig. 6. The profile log-likelihood function (42) for the $\vartheta_1$ parameter for the Pharmacodynamics model (40) conditioned on data from patient 1.

In Figures 7, 8, and 9 we present the posterior samples conditioned on data from the first patient. In Figure 7 the sampling was done using the TMCMC algorithm and $10^4$ samples, and in Figures 8 and 9 the sampling was done using the smTMCMC and $10^4$ and $10^5$ samples, respectively. The histograms for the eight parameters are plotted along the diagonal. Pair samples and a smoothed version of the pair marginal histogram are plotted in the upper and lower triangular part of the figure, respectively. We note that the TMCMC is trapped in local maxima of the posterior distribution in all directions and it is not able to correctly populate the posterior sample space. This is evident from the spikes on the histograms on Figure 7. This unnatural local mass concentration does not appear in the smTMCMC sample set shown in Figure 8.

We have also observed that the TMCMC algorithm does not produce consistent samples. The shape of the estimated distribution varies significantly between individual runs of the algorithm. Moreover, this problem is not fixed by increasing the size...
of the sample set. In contrast, this is not the case for the smTMCMC algorithm. The distribution presented in Figure 9 does not change between individual runs of the algorithm or if the number of samples is increased. This becomes evident by comparing Figure 8 and Figure 9, where $10^4$ and $10^5$ samples were used, respectively.

The location of the ML estimate using CMA-ES and the smTMCMC algorithm is marked in the diagonal histograms of Figure 8 and Figure 9 with $\times$ and $\circ$, respectively. It can be observed that there is a discrepancy from the CMA-ES estimate in Figure 8, where $10^4$ samples were used. In Figure 9 the discrepancy is alleviated by increasing the sample size to $10^5$. With this observation, it becomes evident that the high probability areas of the posterior distribution are populated. In summary, we find that smTMCMC encompasses the properties of TMCMC, and its extensions enable it to effectively sample the posterior distribution of a challenging Pharmacodynamics model using clinical data.

5. Conclusion. We have proposed a new population based sampling algorithm that combines the transitional MCMC (TMCMC) algorithm with Langevin diffusion transition kernels. Instead of using isotropic diffusion in the TMCMC algorithm, the proposal scheme is based on the time discretization of a Langevin diffusion that takes into consideration the geometry of the target distribution. Thus diffusion is taking place in a manifold that is defined through either the Hessian or the Fisher information matrix of the distribution.

The adaptation to the local geometry of the distribution enhances the sampling capabilities of the method. At the same time the requirement for a positive definite Hessian or Fisher information matrix cannot be guaranteed for general distributions. Even if the matrices are positive definite, some eigenvalues may be close to zero, severely affecting the efficiency of the algorithm. Such cases of nearly zero eigenvalues are present in systems with unidentifiable parameters. We alleviate the problems associated with these matrices by introducing the manifold TMCMC (mTMCMC). We successfully applied the proposed algorithm using a Gaussian distribution with probability mass concentrated in the boundaries of the prior distribution as well as with a bimodal Gaussian mixture distribution, greatly increasing the sampling quality. Finally, we showcased the ability of the proposed algorithm to sample challenging, multimodal distributions in the presence of unidentifiable manifolds using the posterior distribution in a Pharmacodynamics model, and we compared its efficiency to that of existing sampling methods.

The extension of the algorithm for priors other than uniform is also conditionally possible. If the prior distribution is not log-concave, then the proposed algorithm is not applicable. In the case of a log-concave prior distribution, one has to redefine the extended boundary technique, presented in subsection 3.1, in the absence of a prior with compact support.

Appendix A. Derivatives of ordinary differential equations. In this section we show how the derivatives of the model output, $f$, with respect to the model parameters, $\varphi$, can be computed. These quantities appear in the evaluation of the derivative of the log-likelihood function with respect to the parameters; see (26) and (27).

Let $f := f(Y, t; \varphi)$ be an observable function on the solution of the ODE system,

\[
\begin{align*}
\dot{Y}(t) &= G(Y; \varphi), & t &\in [0, T], \\
Y(0) &= g(\varphi),
\end{align*}
\]  

(43)
where $Y \in \mathbb{R}^{N_Y}$ and $\varphi \in \mathbb{R}^{N_\varphi}$ are the vector of parameters. Here, we assume that $G$ is a smooth function such that all the derivatives used below are well defined. In order to find the first and second order derivatives needed for the manifold algorithms, the following extended system must be solved:

\begin{equation}
\begin{split}
\dot{Y} &= G(Y; \varphi), \\
\dot{S}_k &= G^1_k(Y, S; \varphi), \quad k = 1, \ldots, N_\varphi, \\
\dot{H}_{k,\ell} &= G^2_{k,\ell}(Y, S, H; \varphi), \quad k, \ell = 1, \ldots, N_\varphi, \\
Y(0) &= g(\varphi), \quad S(0) = g^1_k(\varphi), \quad H(0) = g^2_{k,\ell}(\varphi),
\end{split}
\end{equation}

where $G^1_k$ and $G^2_{k,\ell}$ are the first total derivative of $G$ with respect to $\varphi_k$ and the second total derivative of $G$ with respect to $\varphi_k$ and $\varphi_\ell$, respectively, and $g^1_k = \frac{\partial}{\partial \varphi_k} g$ and $g^2_{k,\ell} = \frac{\partial^2}{\partial \varphi_k \partial \varphi_\ell} g$.

The function $G^1_k = D_{\varphi_k} G$ for $k = 1, \ldots, N_\varphi$ is given by

\begin{equation}
G^1_k = AS_k + B_k,
\end{equation}

where

\begin{equation}
A_{ij} := A_{ij}(Y; \varphi) = \frac{\partial}{\partial Y_j} G_i(Y; \varphi)
\end{equation}
Fig. 8. Samples from the posterior distribution of Pharmacodynamics model (40) conditioned on data from patient 1 and using the smTMCMC algorithm with $10^4$ samples. The $\times$ and $\circ$ correspond to the ML estimates obtained by the CMA-ES and the smTMCMC algorithm, respectively.

for $i, j = 1, \ldots, N_Y$ and

$$B_k := B_k(Y; \varphi) = \frac{\partial}{\partial \varphi_k} G(Y; \varphi).$$

The function $G^2_{k,\ell} = D_{\varphi_k} D_{\varphi\ell} G$ for $k, \ell = 1, \ldots, N_\varphi$ is given by

$$G^2_{k,\ell} = AH_{k,\ell} + (I \otimes S_k^T) C_{k,\ell} (I \otimes S_\ell) + D_k S_\ell + D_\ell S_k + J_{k,\ell},$$

where $A$ is defined in (46), $\otimes$ is the Kronecker product, $I \in \mathbb{R}^{N_Y \times N_Y}$ is the identity matrix, and $1 = (1, \ldots, 1) \in \mathbb{R}^{N_Y}$. The matrix $C$ is a block diagonal matrix $C_{k,\ell} = \text{diag}(C_{k,\ell,1}, \ldots, C_{k,\ell,N_Y})$ with the block matrices given by

$$C_{k,\ell,i} := C_{k,\ell,i}(Y; \varphi) = \frac{\partial^2}{\partial Y_k \partial Y_\ell} G_i(Y; \varphi).$$

The matrix $D_k$ and the vector $J_{k,\ell}$ are given by

$$D_{k,i,j} := D_{k,i,j}(Y; \varphi) = \frac{\partial^2}{\partial \varphi_k \partial Y_j} G_i(Y; \varphi).$$
Fig. 9. Samples from the posterior distribution of Pharmacodynamics model (40) conditioned on data from patient 1 and using the smTMCMC algorithm with $10^5$ samples. The × and ○ correspond to the ML estimates obtained by the CMA-ES and the smTMCMC algorithm, respectively.

\begin{equation}
J_{k,\ell} := J_{k,\ell}(Y; \varphi) = \frac{\partial^2}{\partial \varphi_k \partial \varphi_\ell} G(Y; \varphi)
\end{equation}

for $i,j = 1, \ldots, N_Y$.

A MATLAB function is provided which, given functions $G$ and $g$, by performing symbolic calculations, gives as output the functions $G_k^1$ and $G_{k,\ell}^2$ as MATLAB function handles.

**Appendix B. Derivative of the transformed matrix.**

**Proposition B.1.** Let $A \in \mathbb{R}^{n \times n}$ be a matrix that depends on a parameter $\vartheta$, and let $A = Q \Lambda Q^T$ be its eigendecomposition. Let $f(A) = Q f(\Lambda) Q^T$ be a transformation of the matrix $A$. Then, it holds that

\begin{equation}
\frac{\partial f(A)}{\partial \vartheta} = Q \left( J \circ \left( Q^T \frac{\partial A}{\partial \vartheta} Q \right) \right) Q^T,
\end{equation}

where $\circ$ denotes the Hadamard product and $J$ is given by

\begin{equation}
J_{i,j} = \begin{cases}
\frac{f(\lambda_i) - f(\lambda_j)}{\lambda_i - \lambda_j}, & i \neq j, \\
\frac{\partial f(\lambda_i)}{\partial \lambda_i}, & i = j.
\end{cases}
\end{equation}
Proof. For the proof see section 2 in [1].

Acknowledgments. We would like to thank Dr. Benjamin Ribba and Dr. François Ducray for permission to use the data in the Pharmacodynamics model of subsection 4.3. We would also like to thank the anonymous reviewers for their valuable comments.

REFERENCES


LANGEVIN DIFFUSION FOR POPULATION BASED SAMPLING  


