SIAM/ASA J. Uncertainty Quantification
Vol. 11, No. 1, pp. 1–30

Multilevel Delayed Acceptance MCMC*

M. B. Lykkegaard†, T. J. Dodwell‡, C. Fox§, G. Mingas¶, and R. Scheichl‖

Abstract. We develop a novel Markov chain Monte Carlo (MCMC) method that exploits a hierarchy of models of increasing complexity to efficiently generate samples from an unnormalized target distribution. Broadly, the method rewrites the multilevel MCMC approach of Dodwell et al. [*SIAM/ASA J. Uncertain. Quantif.*, 3 (2015), pp. 1075–1108] in terms of the delayed acceptance MCMC of Christen and Fox [*J. Comput. Graph. Statist.*, 14 (2005), pp. 795–810]. In particular, delayed acceptance is extended to use a hierarchy of models of arbitrary depth and allow subchains of arbitrary length. We show that the algorithm satisfies detailed balance and hence is ergodic for the target distribution. Furthermore, multilevel variance reduction is derived that exploits the multiple levels and subchains, and an adaptive multilevel correction to coarse-level biases is developed. Three numerical examples of Bayesian inverse problems are presented that demonstrate the advantages of these novel methods. The software and examples are available in `PyMC3`.

Key words. Markov chain Monte Carlo, Bayesian inverse problems, multilevel methods, model hierarchies, detailed balance, variance reduction, adaptive error model

MSC codes. 62F15, 62M05, 65C05, 65C40

DOI. 10.1137/22M1476770

1. Introduction.

Sampling from an unnormalized posterior distribution $\pi(\cdot)$ using Markov chain Monte Carlo (MCMC) methods is a central task in computational statistics. This can be a particularly challenging problem when the evaluation of $\pi(\cdot)$ is computationally expensive and the parameters θ and/or data $\mathbf{d}$ defining $\pi(\cdot)$ are high-dimensional. The sequential (highly) correlated nature of a Markov chain and the slow converge rates of MCMC sampling mean that often many MCMC samples are required to obtain a sufficient representation

* Received by the editors February 8, 2022; accepted for publication (in revised form) August 19, 2022; published electronically January 25, 2023.
https://doi.org/10.1137/22M1476770
Funding: The first author was funded as part of the Water Informatics Science and Engineering Centre for Doctoral Training (WISE CDT) under a grant from the Engineering and Physical Sciences Research Council (EPSRC), grant EP/L016214/1. The second and fourth authors were funded by a Turing AI Fellowship (2TAFFP\100007). The third author was partially funded by MBIE contract UOOX2106. The work of the fifth author is supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2181/1 - 390900948 (the Heidelberg STRUCTURES Excellence Cluster).

† Centre for Water Systems and Institute for Data Science and AI University of Exeter, Exeter EX4 4QF, UK (m.lykkegaard@exeter.ac.uk).
‡ Alan Turing Institute and Institute for Data Science and AI, University of Exeter, Exeter EX4 4QF, UK (t.dodwell@exeter.ac.uk).
§ Department of Physics, University of Otago, Dunedin 9016, New Zealand (colin.fox@otago.ac.nz).
¶ Alan Turing Institute, British Library, 96 Euston Road, London NW1 2DB, UK (gmingas@turing.ac.uk).
‖ Institute for Applied Mathematics and Interdisciplinary Center for Scientific Computing, Heidelberg University, 69120 Heidelberg, Germany (r.scheichl@uni-heidelberg.de).

of a posterior distribution $\pi(\cdot)$. Examples of such challenging problems frequently occur in Bayesian inverse problems, image reconstruction, and probabilistic machine learning, where simulations of the measurements (required to calculate a likelihood function) depend on the evaluation of complex mathematical models (e.g., a system of partial differential equations) or the evaluation of prohibitively large data sets.

The topic of MCMC methods is a rich and active field of research. While the basic idea of the original Metropolis–Hastings algorithm [37, 25] is almost embarrassingly simple, it has given rise to a wide variety of algorithms tailored to different applications, most notably the Gibbs sampler [18], which samples each variable conditional on the other variables, the Metropolis adjusted Langevin algorithm (MALA) [43, 39], Hamiltonian Monte Carlo (HMC) [16], and the No-U-Turn Sampler (NUTS) [27], which all exploit gradient information to improve the MCMC proposals. We would also like to highlight the seminal work of Haario, Saksman, and Tamminen [22] on the adaptive Metropolis sampler that launched a new paradigm of adaptive MCMC algorithms (see, e.g., [2, 1, 42, 50, 51, 14]).

The most efficient MCMC methods cheaply generate candidate proposals, which have a high probability of being accepted while being almost independent from the previous sample. In this paper, we define an MCMC approach capable of accelerating existing sampling methods, where a hierarchy (or sequence) $\pi_0(\cdot), \ldots, \pi_{L-1}(\cdot)$ of computationally cheaper approximations to the exact posterior density $\pi(\cdot) \equiv \pi_L(\cdot)$ is available. As with the original delayed acceptance (DA) algorithm, proposed by Christen and Fox [8], short runs of MCMC subchains, generated using a computationally cheaper, approximate density $\pi_{\ell-1}(\cdot)$, are used to generate proposals for the Markov chain targeting $\pi_\ell(\cdot)$. The original DA method formulated the approach for just two levels and a single step on the coarse level. In this paper we extend the method by recursively applying DA across a hierarchy of model approximations for an arbitrary number of steps on the coarse levels—a method we term *multilevel delayed acceptance* (MLDA). There are clear similarities with multilevel Monte Carlo sampling methods, first proposed by Heinrich [26] and later by Giles [19], which have been widely studied for forward uncertainty propagation problems (see, e.g., [9, 4, 7, 47]) and importantly have been extended to Bayesian inverse problems in the multilevel Markov chain Monte Carlo (MLMCMC) approach by Dodwell et al. [15] as well as to the multi-index setting [23, 28].

The fundamental idea of multilevel methods is simple: We let the cheaper (or *coarse*) models do most of the work. In the context of sampling, be it Monte Carlo or MCMC, this entails drawing more samples on the coarser levels than on the finer, and we use the entirety of samples across all model levels to improve our Monte Carlo estimates. Additionally, in the context of MCMC, the samplers on the coarser levels inform the samplers on the finer levels by filtering out poor MCMC proposals, effectively boosting the acceptance rate and hence computational efficiency on the finer levels.

The multilevel MCMC algorithm of Dodwell et al. [15] achieves these goals and, importantly, provides a multilevel estimator for quantities of interest, utilizing the output of all chains, to allow tuning of work at each level to maximize variance reduction per compute effort. MLMCMC also allows parallelization across levels by running chains at the coarser levels independently of the finer. However, a by-product of the latter property is that MLMCMC only produces provably unbiased estimates in the limit of infinitely long coarse chains; see

section 2.2.4. This is a potential problem as computational efficiency may require quite short coarse chains.

One of our main motivations for reworking MLMCMC was to develop an MCMC that could operate with multiple levels of approximation for which we can write a multilevel estimator and that is provably unbiased for finite-length coarse chains. This paper reports the resulting algorithm, that extends the delayed acceptance MCMC to a multilevel setting with finite-length coarse chains. Those extensions pose several challenges requiring novel solutions: (1) As mentioned above, DA evaluates proposals using a single step on the coarse level; the extension to finite-length subchains is presented in section 2.2.1. (2) A less-obvious challenge is that MLMCMC operates with a different state variable at each level, with fewer components in the state at coarser levels, whereas DA uses the same state at both levels; extension of DA to using embedded state spaces is presented in section 2.2.2, where the extra "modes" at the fine level are proposed using an additional kernel. The extension to a multilevel DA is then straightforward by recursion on levels, as presented in section 2.2.3. (3) A further challenge is deriving a multilevel estimator for MLDA since the coarse chains in MLDA do not converge to known approximate posterior distributions, unlike MLMCMC, where the independence of chains means that, after burn-in, each chain samples from a known approximate distribution. In contrast, the short coarse chains in MLDA are, in a sense, always in burn-in. We overcome this difficulty by randomizing subchain length for proposals, as shown in section 2.2.1, and using a fixed subchain length for fine-level estimates to ensure that estimates of equivalent terms in the telescoping sums converge to the same value. That multilevel estimator is presented in section 2.3. The *adaptive* DA algorithm introduced in [13] increases significantly the statistical efficiency by constructing a posteriori error models that improve the approximate posterior distributions at coarse levels; see [14, 17]. Adaptive error models (AEMs) for MLDA are presented in section 2.4.

Finally, a further challenge is that DA MCMC is inherently sequential and fine-level proposals must be evaluated on the coarse levels, which precludes parallelization across levels. Whether MLDA can be effectively parallelized remains an outstanding question, which we discuss in section 4.

The paper is structured as follows. In the following section we present the MLDA algorithm, proving detailed balance of each extension of DA. In this process, we develop two additional algorithms, namely *randomized-length-subchain surrogate transition* (RST) in section 2.2.1 and *two-level delayed acceptance* (TLDA) in section 2.2.2, each of which is a valid MCMC sampler in its own respect. Throughout these sections we develop algorithms for two levels only, denoted C for "coarse" (the approximate chain) and F for "fine" (the exact chain). In section 2.2.2 we introduce different states at coarse and fine levels, also denoted (with a slight abuse of notation) by subscripts C and F, respectively. A recursive, multilevel DA algorithm is defined in section 2.2.3 with detailed balance following from previous sections. A comparison of MLDA and MLMCMC is presented in section 2.2.4 to provide some intuition on similarities and differences of the two algorithms. MLDA then provides a provably convergent multilevel algorithm for which we develop a multilevel estimator in section 2.3 that can be exploited for variance reduction. AEMs are developed in section 2.4. In section 3, we demonstrate the algorithm using three examples of Bayesian inverse problems. First, we show that extended subchains on the coarse level can significantly increase the effective sample size

compared to an equivalent single-level sampler on the fine level, using an example from gravitational surveying. Second, we demonstrate multilevel variance reduction on a predator-prey model, where coarse models are constructed by restricting the length of the time window over which the differential equation model is fitted to data. Third, we demonstrate the multilevel error model in the context of a subsurface flow problem. We show that when we utilize the error model, we can achieve high effective sample sizes on the finest level, even when a very crude approximation is employed as the coarsest model. Conclusions and future work are discussed in section 4.

2. Multilevel delayed acceptance. In this section we first outline the theoretical foundations of vanilla Metropolis–Hastings based MCMC [37, 25] and the DA method proposed by Christen and Fox [8]. We extend DA in two ways: horizontally, by allowing the coarse sampler to construct subchains of multiple coarse samples before proposing a sample on the fine level, and vertically, by recursively using DA on an entire hierarchy of models with increasing resolution/accuracy. This constitutes the MLDA sampler. From this foundation we further develop a multilevel estimator to exploit variance reduction and a multilevel AEM which improves the statistical efficiency of the algorithm.

2.1. Basic MCMC, ergodic theorems, and delayed acceptance. To show that MLDA correctly generates samples from the unnormalized target density $\pi(\cdot)$ we will build on standard ergodicity results for Markov chains (see [40] and references therein). Each algorithm considered here defines a stochastic iteration on a well-defined state and so defines a Markov chain. Hence, we can apply classical ergodic theorems for Markov chains.

The ergodic theorems for Markov chains (see [40] and references therein) state that the chain is π-ergodic if the chain is π-irreducible, aperiodic, and reversible with respect to π. Essentially, irreducibility and aperiodicity guarantee that the Markov chain has a unique equilibrium distribution, while reversibility with respect to π ensures that π is that unique distribution. The condition of π-irreducibility is satisfied when the proposal distribution is chosen such that the standard Metropolis–Hastings algorithm is π-irreducible. For algorithms based on DA, it is also necessary that the coarse-level approximation is chosen to maintain irreducibility; see [8, Thm. 1] for precise conditions on the approximation. Aperiodicity is a mild condition that is satisfied by any Metropolis–Hastings algorithm with a nonzero probability of rejection on any π-positive set; again see [8, Thm. 1]. We will assume that the proposal and approximations are chosen so that these conditions hold. Accordingly, we focus on establishing reversibility of algorithms, which is equivalent to the stochastic iteration being in detailed balance with the target density π; see [32].

2.1.1. Metropolis–Hastings MCMC. Consider first the plain vanilla Metropolis–Hastings algorithm for sampling from target density π_t. Given an initial state θ^0 and a proposal distribution with density function $q(\cdot|\theta)$, the Metropolis–Hastings algorithm for generating a chain of length N is given in Algorithm 1.

For each j, Algorithm 1 simulates a fixed stochastic iteration with θ^{j+1} being conditionally dependent only on θ^j, the state at step j, which can be represented by a fixed (stationary) transition kernel $K(y|x)$ that generates a (homogeneous) Markov chain. For target density π_t, the detailed balance may be written

Algorithm 1. Metropolis–Hastings (MH).

function: $\left[\theta^1, \ldots, \theta^N\right] = \mathbf{MH}\left(\pi_{\mathrm{t}}(\cdot), q(\cdot|\cdot), \theta^0, N\right)$
input: density of target distribution $\pi_{\mathrm{t}}(\cdot)$, density of proposal distribution $q(\cdot|\cdot)$, initial state θ^0, number of steps N
output: ordered list of states $\left[\theta^1, \ldots, \theta^N\right]$ (or just the final state θ^N)
for $j = 0$ to $N - 1$:

- Given θ^j, generate a proposal ψ distributed as $q(\psi|\theta^j)$,
- Accept proposal ψ as the next state, i.e., set $\theta^{j+1} = \psi$, with probability

$$\alpha(\psi|\theta^j) = \min\left\{1, \frac{\pi_{\mathrm{t}}(\psi)q(\theta^j|\psi)}{\pi_{\mathrm{t}}(\theta^j)q(\psi|\theta^j)}\right\}, \tag{2.1}$$

otherwise reject ψ and set $\theta^{j+1} = \theta^j$.

$$\pi_{\mathrm{t}}(x)\, K(y|x) = \pi_{\mathrm{t}}(y)\, K(x|y),$$

which, in general, is the property that K is self-adjoint in the measure π_{t}. See [32, sect. 5.3] for a nice method for showing that K simulated by **MH** Algorithm 1 is in detailed balance with π_{t}, and also for a more general class of acceptance probabilities.

Hence, under mild conditions on the proposal density q and the initial state θ^0, the ergodic theorem for Markov chains applies, which guarantees that the j-step density converges to π_{t}, asymptotically as $j \to \infty$. Hence, the Markov chain is π_{t}-ergodic.

A common choice of proposal distributions for inverse problems in multiple dimensions is random-walk proposals, though these typically lead to adjacent states of the chain being highly correlated, resulting in high computational cost to estimate posterior expectations with a desired accuracy. In the following we do not discuss the choice of proposal q, though in some sense our primary concern is how to improve a proposal once chosen. We also do not discuss the choice of initial state.

The following lemma gives an alternative form of the acceptance probability in (2.1) used later.

Lemma 2.1. *If the proposal transition kernel $q(\cdot|\cdot)$ in Algorithm 1 is in detailed balance with some distribution π^*, then the acceptance probability (2.1) may be written*

$$\alpha(\psi|\theta^j) = \min\left\{1, \frac{\pi_{\mathrm{t}}(\psi)\pi^*(\theta^j))}{\pi_{\mathrm{t}}(\theta^j)\pi^*(\psi)}\right\}. \tag{2.2}$$

Proof. Substitute the detailed balance statement $\pi^*(\psi)q(\theta^j|\psi) = \pi^*(\theta^j))q(\psi|\theta^j)$ into (2.1) to get (2.2) almost everywhere. ∎

2.1.2. MCMC for hierarchical Bayesian models. A hierarchical Bayesian model of some problem, including inverse problems, leads to the posterior distribution for unknown parameters θ conditioned on measured data $\mathbf{d}$, given by Bayes' rule

$$\pi(\theta|\mathbf{d}) = \frac{\pi(\mathbf{d}|\theta)\pi_{\mathrm{p}}(\theta)}{\pi(\mathbf{d})}. \tag{2.3}$$

In the language of Bayesian analysis, $\pi_{\mathrm{p}}(\theta)$ is the *prior* distribution, $\pi(\mathbf{d}|\theta)$ as a function of θ is the *likelihood* function, and $\pi(\mathbf{d})$ is a normalizing constant commonly referred to as the *evidence*. The likelihood function is induced by the data-generating model

$$\mathbf{d} = \mathcal{F}(\theta) + \epsilon, \tag{2.4}$$

where $\mathcal{F}(\theta)$ is the forward model and ϵ is the measurement error. When the measurement error is Gaussian, i.e., $\epsilon \sim \mathcal{N}(0, \Sigma_\epsilon)$, the particular likelihood function is proportional to

$$\mathcal{L}(\mathbf{d}|\theta) = \exp\left(-\frac{1}{2}(\mathcal{F}(\theta) - \mathbf{d})^T \mathbf{\Sigma}_\epsilon^{-1}(\mathcal{F}(\theta) - \mathbf{d})\right). \tag{2.5}$$

In the Bayesian framework, solving the inverse problem is performed by exploring the posterior distribution $\pi(\theta|\mathbf{d})$ defined by (2.3) and evaluating statistics with respect to that distribution. Sample-based inference does this by drawing samples from the posterior distribution to evaluate sample-based Monte Carlo estimates of expected values. The plain vanilla route to drawing samples from $\pi(\theta|\mathbf{d})$ is to invoke the MH algorithm, Algorithm 1, with $\pi_{\mathrm{t}}(\cdot) = \pi(\cdot|\mathbf{d})$ such that

$$[\theta^1, \ldots, \theta^N] = \mathbf{MH}\left(\pi(\theta|\mathbf{d}), q(\cdot|\cdot), \theta^0, N\right).$$

Asymptotically, the density of the jth state θ^j converges to the posterior density $\pi(\cdot|\mathbf{d})$ and averages over this chain converge to expectations with respect to $\pi(\cdot|\mathbf{d})$, asymptotically in N.

Remark 1. When $\pi(\mathbf{d})$ in (2.3) is finite, the Metropolis ratio $\pi_{\mathrm{t}}(\psi)/\pi_{\mathrm{t}}(\theta^j)$ in Algorithm 1, equation (2.1), may be evaluated as a ratio of unnormalized densities

$$\frac{\pi(\mathbf{d}|\psi)\pi_p(\psi)}{\pi(\mathbf{d}|\theta^j)\pi_p(\theta^j)}. \tag{2.6}$$

Substitute $\pi_{\mathrm{t}}(\cdot) = \pi(\cdot|\mathbf{d})$ from (2.3) into the Metropolis ratio and note that the normalization constants $1/\pi(\mathbf{d})$ in the numerator and in the denominator cancel. Hereafter, for brevity we typically write the acceptance probability using the ratio of normalized posterior densities, as in (2.1) but actually compute with unnormalized densities, as in (2.6).

2.1.3. Delayed acceptance MCMC. The DA algorithm was introduced by Christen and Fox in [8], with the goal of reducing the computational cost per iteration by utilizing a computationally cheaper approximation of the forward map, and thus also of the posterior density, for evaluating the acceptance probability in Algorithm 1. One may also view DA as a way to improve the proposal kernel q, since DA modifies the proposal kernel using a Metropolis–Hastings accept-reject step to give an effective proposal that is in detailed balance with an (approximate) distribution that is hopefully closer to the target than is the equilibrium distribution of the original proposal kernel.

The DA algorithm is given in Algorithm 2 for target (fine) density π_{F} and approximate (coarse) density π_{C}. DA first performs a standard Metropolis–Hastings accept/reject step

Algorithm 2. Delayed acceptance (DA).

function: $\left[\theta^1, \ldots, \theta^N\right] = \mathbf{DA}\left(\pi_{\mathrm{F}}(\cdot), \pi_{\mathrm{C}}(\cdot), q(\cdot|\cdot), \theta^0, N\right)$
input: target (fine) density $\pi_{\mathrm{F}}(\cdot)$, approximate (coarse) density $\pi_{\mathrm{C}}(\cdot)$, proposal kernel $q(\cdot|\cdot)$, initial state θ^0, number of steps N
output: ordered list of states $\left[\theta^1, \ldots, \theta^N\right]$ (or just the final state θ^N)
for $j = 0$ to $N - 1$:

- Given θ^j, generate proposal ψ by invoking one step of the MH algorithm, Algorithm 1, for coarse target π_{C}:

$$\psi = \mathbf{MH}\left(\pi_{\mathrm{C}}(\cdot), q(\cdot|\cdot), \theta^j, 1\right). \tag{2.7}$$

- Accept proposal ψ as the next state, i.e., set $\theta^{j+1} = \psi$, with probability

$$\alpha(\psi|\theta^j) = \min\left\{1, \frac{\pi_{\mathrm{F}}(\psi) q_{\mathrm{C}}(\theta^j|\psi)}{\pi_{\mathrm{F}}(\theta^j) q_{\mathrm{C}}(\psi|\theta^j)}\right\}, \tag{2.8}$$

otherwise reject proposal ψ and set $\theta^{j+1} = \theta^j$.

(as given in Algorithm 1) with the approximate/coarse density π_{C}. If accepted, a second accept/reject step is used, with acceptance probability chosen such that the composite iteration satisfies detailed balance with respect to the desired target π_{F}.

In Algorithm 2, equation (2.8), $q_{\mathrm{C}}(\cdot|\cdot)$ is the effective proposal density from the first Metropolis–Hastings step with coarse density $\pi_{\mathrm{C}}(\cdot)$ as target; see [8] for details. The acceptance probability in (2.8) is the standard Metropolis–Hastings rule for proposal density q_{C}, targeting $\pi_{\mathrm{F}}(\cdot)$, hence Algorithm 2 simulates a kernel in detailed balance with $\pi_{\mathrm{F}}(\cdot)$ and produces a chain that is ergodic with respect to $\pi_{\mathrm{F}}(\cdot)$; see [8] for conditions on the approximation that ensure that the ergodic theorem applies. Computational cost per iteration is reduced because for proposals that are rejected in the first MH step in (2.7), and thus result in $\psi = \theta^j$, the second acceptance ratio in (2.8) involving the more expensive, fine target density $\pi_{\mathrm{F}}(\cdot)$ does not need to be evaluated again.

In the multilevel context with levels indexed by ℓ, the original DA algorithm, Algorithm 2, is a two-level method. Denote the more accurate forward map that defines the fine posterior distribution $\pi_\ell(\theta_\ell|\mathbf{d}_\ell)$ by $\mathcal{F}_\ell$ and the less accurate forward map that defines the approximate (coarse) posterior distribution $\pi_{\ell-1}(\theta_\ell|\mathbf{d}_{\ell-1})$ by $\mathcal{F}_{\ell-1}$. Note that we also allow a possibly altered or reduced data set $\mathbf{d}_{\ell-1}$ on level $\ell-1$ but that the states in the two forward maps and in the two distributions are the same. Then setting $\pi_{\mathrm{F}}(\cdot) = \pi_\ell(\cdot|\mathbf{d}_\ell)$ and $\pi_{\mathrm{C}}(\cdot) = \pi_{\ell-1}(\cdot|\mathbf{d}_{\ell-1})$ in the call to the DA algorithm, Algorithm 2, such that

$$[\theta_\ell^1, \ldots, \theta_\ell^N] = \mathbf{DA}\left(\pi_\ell(\cdot|\mathbf{d}_\ell), \pi_{\ell-1}(\cdot|\mathbf{d}_{\ell-1}), q(\cdot|\cdot), \theta^0, N\right),$$

computes a chain that is ergodic with respect to $\pi_\ell(\cdot|\mathbf{d}_\ell)$, asymptotically as $N \to \infty$.

The DA algorithm, Algorithm 2, actually allows for the approximate, coarse posterior distribution to depend on the state of the chain. Denote the state-dependent, approximate forward map at state θ by $\mathcal{F}_{\ell-1,\theta}$ and the resulting approximate posterior density by $\pi_{\ell-1,\theta}(\cdot|\mathbf{d}_{\ell-1})$. For state-dependent approximations it is always desirable and easy to achieve (see [14]) that $\mathcal{F}_{\ell-1,\theta}(\theta) = \mathcal{F}_\ell(\theta)$, so that $\pi_{\ell-1,\theta}(\theta|\mathbf{d}_{\ell-1}) = k\pi_\ell(\theta|\mathbf{d}_\ell)$ with the normalizing constant k independent of state θ. The acceptance probability (2.8) then has the explicit form

$$\alpha(\psi|\theta^j) = \min\left\{1, \frac{\min\left\{\pi_{\mathrm{F}}(\psi)q(\theta^j|\psi), \pi_{\mathrm{C},\psi}(\theta^j)q(\psi|\theta^j)\right\}}{\min\left\{\pi_{\mathrm{F}}(\theta^j)q(\psi|\theta^j), \pi_{\mathrm{C},\theta^j_\ell}(\psi)q(\theta^j|\psi)\right\}}\right\}. \tag{2.9}$$

For technical reasons, as explained in Remark 3 below, we will not use state-dependent approximations, but rather restrict ourselves to fixed approximate forward maps that do not depend on the current state.

2.2. Detailed balance beyond two levels. We will now extend DA to randomized-length subchains, to embedded state spaces at the coarser level, and finally to multiple levels. The resulting Markov chain on the finest level is shown to be in detailed balance with the target density.

2.2.1. Randomized-length-subchain surrogate transition MCMC. When the approximate forward map does not depend on the current state—for example, when using a fixed coarse discretization for a PDE—the resulting approximate posterior density is a fixed *surrogate* for the true posterior density, and Algorithm 2 coincides with the surrogate transition method introduced by Liu [32]. Lemma 2.1 then implies that the acceptance probability in (2.8) is

$$\alpha(\psi|\theta^j) = \min\left\{1, \frac{\pi_{\mathrm{F}}(\psi)\pi_{\mathrm{C}}(\theta^j)}{\pi_{\mathrm{F}}(\theta^j)\pi_{\mathrm{C}}(\psi)}\right\} \tag{2.10}$$

since the Metropolis–Hastings step in (2.7) ensures that the effective proposal kernel $q_{\mathrm{C}}(\cdot|\cdot)$ is in detailed balance with the approximate density $\pi_{\mathrm{C}}(\cdot)$.

We extend the surrogate transition method in two ways. As noted by Liu [32], multiple steps can be made with the surrogate, i.e., iterating the proposal and first accept/reject step in (2.7) before performing the second accept/reject step with acceptance probability in (2.10). We call the sequence of states generated by multiple steps of (2.7) a *subchain*. Further, we consider subchains of random length, set according to a probability mass function $p(\cdot)$ on the positive integers. In practice we set $J \in \mathbb{Z}^+$ and then set $p = \mathcal{U}(\{1, 2, \ldots, J\})$, though note that a deterministic choice of subchain length is another special case. The utility of randomizing the subchain length will become apparent in section 2.3. These extensions are included in Algorithm 3.

We will show that Algorithm 3 satisfies detailed balance using Lemma 2.3, needed also later.

Definition 2.2. *We define composition of Markov kernels K_1 and K_2 in the usual way* [21] *by*

$$(K_1 \circ K_2)(\theta|\psi) = \int K_1(\theta|\phi)K_2(\phi|\psi)\mathrm{d}\phi.$$

Algorithm 3. Randomized-length-subchain surrogate transition (RST).

function: $\left[\theta^1, \ldots, \theta^N\right] = \mathbf{RST}\left(\pi_{\mathrm{F}}(\cdot), \pi_{\mathrm{C}}(\cdot), q(\cdot|\cdot), p(\cdot), \theta^0, N\right)$
input: target (fine) density $\pi_{\mathrm{F}}(\cdot)$, surrogate (coarse) density $\pi_{\mathrm{C}}(\cdot)$, proposal kernel $q(\cdot|\cdot)$, probability mass function $p(\cdot)$ over subchain length, initial state θ^0, number of steps N
output: ordered list of states $\left[\theta^1, \ldots, \theta^N\right]$ (or just the final state θ^N)
for $j = 0$ to $N - 1$:

- Draw the subchain length $n \sim p(\cdot)$.
- Starting at θ^j, generate subchain of length n using the MH algorithm, Algorithm 1, to target $\pi_{\mathrm{C}}(\cdot)$:

$$\psi = \mathbf{MH}\left(\pi_{\mathrm{C}}(\cdot), q(\cdot|\cdot), \theta^j, n\right) \tag{2.11}$$

- Accept the proposal ψ as the next sample, i.e., set $\theta^{j+1} = \psi$, with probability

$$\alpha(\psi|\theta^j) = \min\left\{1, \frac{\pi_{\mathrm{F}}(\psi)\pi_{\mathrm{C}}(\theta^j)}{\pi_{\mathrm{F}}(\theta^j)\pi_{\mathrm{C}}(\psi)}\right\}, \tag{2.12}$$

otherwise reject and set $\theta^{j+1} = \theta^j$.

Composition is associative, by Tonelli's theorem, so, by induction, the composition of multiple Markov kernels is well defined. The composition of a kernel K with itself will be denoted K^2, while the composition of n lots of the kernel K is denoted K^n, so the notation is the same as for composition of transition matrices defining Markov processes with a finite state space.

Lemma 2.3. *Let $K_1(x|y)$ and $K_2(x|y)$ be two transition kernels that are in detailed balance with a density π and that commute. Then their composition $(K_1 \circ K_2)$ is also in detailed balance with π.*

Proof.

$$\begin{aligned}
\pi(\psi)(K_1 \circ K_2)(\theta|\psi) &= \pi(\psi)\int K_1(\theta|\phi)K_2(\phi|\psi)\mathrm{d}\phi = \pi(\psi)\int K_2(\theta|\phi)K_1(\phi|\psi)\mathrm{d}\phi \\
&= \pi(\psi)\int K_2(\phi|\theta)\frac{\pi(\theta)}{\pi(\phi)}K_1(\psi|\phi)\frac{\pi(\phi)}{\pi(\psi)}\mathrm{d}\phi \\
&= \pi(\theta)\int K_2(\phi|\theta)K_1(\psi|\phi)\mathrm{d}\phi = \pi(\theta)(K_1 \circ K_2)(\psi|\theta).
\end{aligned}$$

■

Lemma 2.4. *Algorithm 3 simulates a Markov chain that is in detailed balance with $\pi_{\mathrm{F}}(\cdot)$.*

Proof. Recall that the effective density $q_{\mathrm{C}}(\cdot|\cdot)$ for proposals drawn according to Algorithm 2, equation (2.7), is in detailed balance with $\pi_{\mathrm{C}}(\cdot)$. Since q_{C} clearly commutes with itself, using Lemma 2.3, it follows by induction that $q_{\mathrm{C}}^n(\cdot|\cdot)$ (i.e., q_{C} composed n times with itself) is in detailed balance with $\pi_{\mathrm{C}}(\cdot)$ for any n. Hence, the effective proposal density in-

duced by Algorithm 3, equation (2.11), namely the mixture kernel $\sum_{n\in\mathbb{Z}^+} p(n)q_{\mathrm{C}}^n(\cdot|\cdot)$, is also in detailed balance with $\pi_{\mathrm{C}}(\cdot)$.

Finally, the acceptance probability in Algorithm 3, equation (2.12), for target density $\pi_{\mathrm{F}}(\cdot)$ follows from Lemma 2.1, since the proposal kernel is in detailed balance with $\pi_{\mathrm{C}}(\cdot)$. Consequently, Algorithm 3 produces a chain in detailed balance with $\pi_{\mathrm{F}}(\cdot)$. ∎

Remark 2. Choosing a multinomial probability mass function over the subchain length, with $p(J) = 1$ and $p(\neg J) = 0$, implies that Lemma 2.4 is also valid for the special case of a fixed subchain length J_C.

Remark 3. We do not yet have a version of Lemma 2.4 for fully state-dependent approximations, which is why we restrict here to state-independent surrogates.

Remark 4. If the densities of the coarse and fine posterior distributions in Algorithm 3 are with respect to the same prior distribution, i.e., $\pi_{\mathrm{F}}(\theta) = \pi_\ell(\theta|\mathbf{d}_\ell) \propto \pi_\ell(\mathbf{d}_\ell|\theta)\pi_{\mathrm{p}}(\theta)$ and $\pi_{\mathrm{C}}(\theta) = \pi_{\ell-1}(\theta|\mathbf{d}_{\ell-1}) \propto \pi_{\ell-1}(\mathbf{d}_{\ell-1}|\theta)\pi_{\mathrm{p}}(\theta)$, the acceptance probability in Algorithm 3, equation (2.12), is equal to

$$\alpha(\psi|\theta^j) = \min\left\{1, \frac{\pi_\ell(\mathbf{d}_\ell|\psi)\pi_{\ell-1}(\mathbf{d}_{\ell-1}|\theta^j)}{\pi_\ell(\mathbf{d}_\ell|\theta^j)\pi_{\ell-1}(\mathbf{d}_{\ell-1}|\psi)}\right\}. \tag{2.13}$$

2.2.2. Different fine and coarse states. In the DA algorithm, Algorithm 2, and hence also in the RST algorithm, Algorithm 3, the state in the fine and coarse target distributions is the same. In the MLMCMC of Dodwell et al. [15] different levels can have different states, which is natural when using, e.g., a hierarchy of FEM discretizations with different levels of mesh refinement. In this context, the states at different levels form a hierarchy of embedded spaces, where the state vector at any given level is part of the state vector at the next finer level. Hence, in a two-level hierarchy as described above, the (fine) state θ can be partitioned into "coarse modes" (or "components") denoted θ_{C} and "fine modes" θ_{F}, so that $\theta = (\theta_{\mathrm{F}}, \theta_{\mathrm{C}})$. The coarse modes θ_{C} are the components of the state vector on the coarse, approximate level targeted by π_{C}, while the fine target distribution π_{F} also depends on the fine modes θ_{F}.

The RST algorithm, Algorithm 3, is easily extended to allow this structure, as shown in Algorithm 4 below, where surrogate transition is only used to propose the states of the coarse modes, while the fine modes are drawn from some additional proposal distribution. The composite of the fine and coarse proposals then forms the proposed state at the fine level. For this extension it is important that the fine modes are proposed independently of the coarse modes to ensure detailed balance, as shown below.

Lemma 2.5. *TLDA in Algorithm 4 generates a chain in detailed balance with* π_{F}.

Proof. As noted in the proof of Lemma 2.4, the proposal density q_{C} induced by the surrogate transition step in Algorithm 4, equation (2.14), is in detailed balance with the coarse target density $\pi_{\mathrm{C}}(\cdot)$ over θ_{C}. As a kernel on the composite state $\theta = (\theta_{\mathrm{F}}, \theta_{\mathrm{C}})$ we can write the coarse proposal as

$$K_{\mathrm{C}} = \left[\begin{array}{c|c} I & 0 \\ \hline 0 & q_{\mathrm{C}} \end{array}\right],$$

where I denotes the identity of appropriate dimension. Similarly, the fine proposal (2.15) on the composite state has kernel

$$K_{\mathrm{F}} = \left[\begin{array}{c|c} q_{\mathrm{F}} & 0 \\ \hline 0 & I \end{array}\right].$$

Since K_{F} does not change the coarse modes, it trivially is in detailed balance with $\pi_{\mathrm{C}}(\cdot)$. Further, it is easy to check that K_{C} and K_{F} commute. Hence, by Lemma 2.3 the composition $(K_{\mathrm{F}} \circ K_{\mathrm{C}}^n)$ is also in detailed balance with $\pi_{\mathrm{C}}(\cdot)$ and so is the effective proposal kernel $\sum_{n\in\mathbb{Z}^+} p(n)(K_{\mathrm{F}} \circ K_{\mathrm{C}}^n)$ for drawing $\psi = (\psi_{\mathrm{F}}, \psi_{\mathrm{C}})$ according to Algorithm 4, equations (2.14) and (2.15). The acceptance probability in Algorithm 4, equation (2.16), then follows again from Lemma 2.1 and the chain produced by Algorithm 4 is in detailed balance with $\pi_{\mathrm{F}}(\cdot)$, as desired. ∎

Note that the RST algorithm, Algorithm 3, is a special case of Algorithm 4 with $\theta^j = \theta^j_{\mathrm{C}}$, i.e., θ^j_{F} is empty, and correspondingly $q_{\mathrm{F}}(\cdot|\cdot)$ is the (trivial) proposal on the empty space.

Algorithm 4. Two-level delayed acceptance (TLDA).

function: $\left[\theta^1, \ldots, \theta^N\right] = \mathbf{TLDA}\left(\pi_{\mathrm{F}}(\cdot), \pi_{\mathrm{C}}(\cdot), q(\cdot|\cdot), q_{\mathrm{F}}(\cdot|\cdot), p(\cdot), \theta^0, N\right)$
input: target (fine) density $\pi_{\mathrm{F}}(\cdot)$, surrogate (coarse) density $\pi_{\mathrm{C}}(\cdot)$, proposal kernel $q(\cdot|\cdot)$ on coarse modes, proposal kernel $q_{\mathrm{F}}(\cdot|\cdot)$ on fine modes, probability mass function $p(\cdot)$ over subchain length, initial state θ^0, number of steps N
output: ordered list of states $\left[\theta^1, \ldots, \theta^N\right]$ (or just the final state θ^N)
for $j = 0$ to $N - 1$:

- Draw the subchain length $n \sim p(\cdot)$.
- Starting at θ^j_{C}, generate subchain of length n using the Metropolis–Hastings algorithm, Algorithm 1, to target $\pi_{\mathrm{C}}(\cdot)$:

$$\psi_{\mathrm{C}} = \mathbf{MH}\left(\pi_{\mathrm{C}}(\cdot), q(\cdot|\cdot), \theta^j_{\mathrm{C}}, n\right) \tag{2.14}$$

- Draw the fine-mode proposal

$$\psi_{\mathrm{F}} \sim q_{\mathrm{F}}(\cdot|\theta^j_{\mathrm{F}}) \tag{2.15}$$

- Accept proposal $\psi = (\psi_{\mathrm{F}}, \psi_{\mathrm{C}})$ as next sample, i.e., set $\theta^{j+1} = \psi$, with probability

$$\alpha(\psi|\theta^j) = \min\left\{1, \frac{\pi_{\mathrm{F}}(\psi)\pi_{\mathrm{C}}(\theta^j_{\mathrm{C}})}{\pi_{\mathrm{F}}(\theta^j)\pi_{\mathrm{C}}(\psi_{\mathrm{C}})}\right\}, \tag{2.16}$$

otherwise reject and set $\theta^{j+1} = \theta^j$.

2.2.3. Multilevel delayed acceptance. The MLDA algorithm is a recursive version of TLDA in which instead of invoking Metropolis–Hastings to generate a subchain at the coarser levels, the algorithm is recursively invoked again (except for the coarsest level $\ell = 0$), leading to a *hierarchical* MLDA algorithm, which admits an arbitrary number of model levels L. The flexibility with respect to the depth of the model hierarchy and the subchain lengths allows for tailoring the algorithm to various objectives, including the reduction of variance (see section 2.3) or increasing the effective sample size (see section 3.1).

To be more precise, the MLDA algorithm, Algorithm 5 below, is called on the most accurate, finest level L. Then, for levels $1 \leq \ell \leq L$ it generates a subchain at level $\ell - 1$ as in TLDA, by recursively invoking MLDA on level $\ell - 1$, until the coarsest level $\ell = 0$ is reached, where plain MH in invoked. Required for MLDA are the hierarchy of density functions $\pi_0(\cdot), \ldots, \pi_L(\cdot)$ along with a coarsest level proposal q_0, partitions into coarse and fine modes at each level, fine-mode proposals $q_{1,\mathrm{F}}, \ldots, q_{L,\mathrm{F}}$, and probability mass functions

Algorithm 5. Multilevel delayed acceptance (MLDA).

function: $\left[\theta_\ell^1, \ldots, \theta_\ell^N\right] = \mathbf{MLDA}\left(\{\pi_k\}_{k=0}^{\ell}, q_0, \{q_{k,\mathrm{F}}\}_{k=1}^{\ell}, \{p_k\}_{k=1}^{\ell}, \theta_\ell^0, \ell, N\right)$

input: target densities $\pi_0(\cdot), \ldots \pi_\ell(\cdot)$, proposal densities $q_0(\cdot|\cdot)$ and $q_{1,\mathrm{F}}(\cdot|\cdot), \ldots, q_{\ell,\mathrm{F}}$, probability mass functions $p_1(\cdot), \ldots, p_\ell(\cdot)$ over subchain lengths on levels 0 to $\ell - 1$, initial state θ_ℓ^0, current level index ℓ, number of steps N

output: ordered list of states $[\theta_\ell^1, \ldots, \theta_\ell^N]$ at level ℓ (or just the final state θ_ℓ^N)

for $j = 0$ to $N - 1$:

- Draw the subchain length $n_\ell \sim p_\ell(\cdot)$ for level $\ell - 1$.
- Starting at $\theta_{\ell,\mathrm{C}}^j$, generate a subchain of length n_ℓ on level $\ell - 1$:
  - If $\ell = 1$, use the Metropolis–Hastings algorithm to generate the subchain,

$$\psi_{\mathrm{C}} = \mathbf{MH}\left(\pi_0(\cdot), q_0(\cdot, \cdot), \theta_{1,\mathrm{C}}^j, n_1\right).$$

  - If $\ell > 1$, generate the subchain by (recursively) calling MLDA,

$$\psi_{\mathrm{C}} = \mathbf{MLDA}\left(\{\pi_k(\cdot)\}_{k=0}^{\ell-1}, q_0(\cdot|\cdot), \{q_{k,\mathrm{F}}\}_{k=1}^{\ell-1}, \{p_k\}_{k=1}^{\ell-1}, \theta_{\ell,\mathrm{C}}^j, \ell - 1, n_\ell\right).$$

- Draw the fine-mode proposal $\psi_{\mathrm{F}} \sim q_{\ell,\mathrm{F}}(\,\cdot\,|\theta_{\ell,\mathrm{F}}^j)$.
- Accept proposal $\psi = (\psi_{\mathrm{F}}, \psi_{\mathrm{C}})$ as next sample, i.e., set $\theta_\ell^{j+1} = \psi$, with probability

$$\alpha(\psi|\theta^j) = \min\left\{1, \frac{\pi_\ell(\psi)\pi_{\ell-1}(\theta_{\ell,\mathrm{C}}^j)}{\pi_\ell(\theta_\ell^j)\pi_{\ell-1}(\psi_{\mathrm{C}})}\right\}, \tag{2.17}$$

otherwise reject and set $\theta_\ell^{j+1} = \theta_\ell^j$.

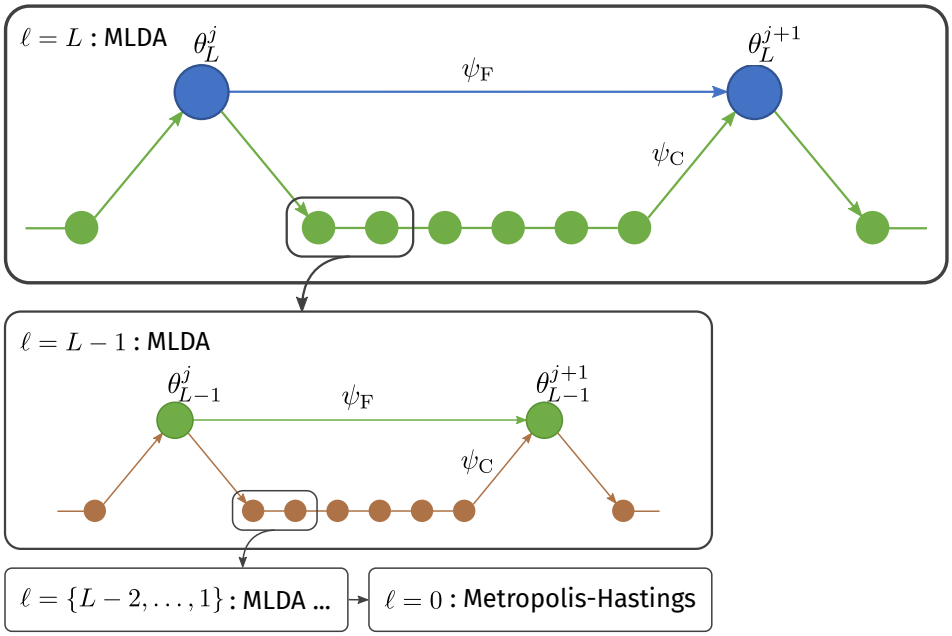

Figure 1. *The MLDA algorithm sampling with a model hierarchy with L levels. The MLDA sampler is employed recursively on each level $\ell > 0$, while on level $\ell = 0$, any Metropolis–Hastings algorithm can be used. On level $\ell = L$, the MLDA sampler generates a Markov chain in detailed balance with π_L, according to Theorem 2.6. On each level $\ell < L$, the respective samplers generate proposals for the coarse modes ψ_{C} of the next finer level.*

$p_1(\cdot), \ldots, p_L(\cdot)$ over the subchain lengths on levels 0 to $L-1$. Note that the fine-mode proposals are used to draw the *additional* finer modes on each level $1 \leq \ell \leq L$, to construct a hierarchy of embedded spaces as explained in section 2.2.2. The algorithm is illustrated conceptually in Figure 1.

A chain of length N at level L is then produced by calling

$$\left[\theta_L^1, \ldots, \theta_L^N\right] = \mathbf{MLDA}\left(\{\pi_k\}_{k=0}^L, q_0, \{q_{k,\mathrm{F}}\}_{k=1}^L, \{p_k\}_{k=1}^L, \theta_L^0, L, N\right). \tag{2.18}$$

We can now state the main theoretical result of paper.

Theorem 2.6. *MLDA in Algorithm 5, invoked as in (2.18), generates a Markov chain that is in detailed balance with π_L.*

Proof. The proof follows essentially by induction on the level ℓ from the proof of Lemma 2.5. At level $\ell = 1$, MLDA is equivalent to TLDA, and so the base step follows immediately from Lemma 2.5. Let us now assume that the proposal kernel for $\psi = (\psi_{\mathrm{F}}, \psi_{\mathrm{C}})$ on level ℓ simulated using MLDA on level $\ell-1$ is in detailed balance with $\pi_{\ell-1}$. Then it follows from Lemma 2.1 that the acceptance probability in Algorithm 5, equation (2.17), produces a Markov chain that is in detailed balance with $\pi_\ell(\cdot)$, which concludes the induction step. ∎

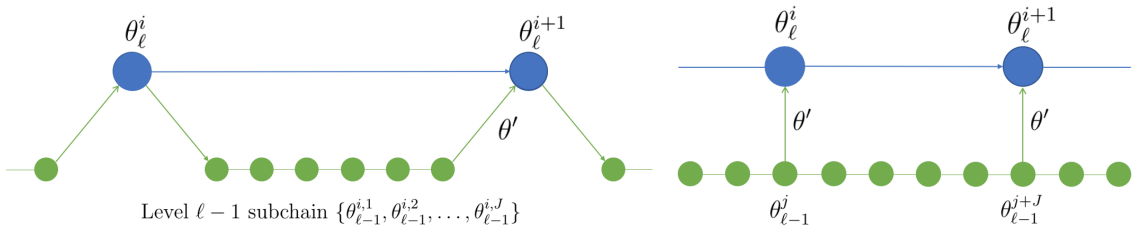

Figure 2. *Schematic for generating a proposal θ' on level ℓ for MLDA (left) and MLMCMC (right) using a fixed length subchain of length J. The key difference is that for MLMCMC the coarse chain on level $\ell-1$ is generated independently of the chain on level ℓ.*

2.2.4. Comparison with MLMCMC. The generalization of DA to an extended multilevel setting leads to clear similarities with the MLMCMC method proposed by Dodwell et al. [15]. The more subtle difference between the two approaches is illustrated in Figure 2.

The MLDA algorithm can be seen as a recursive application of the surrogate transition method over multiple levels. If a proposal ψ from level $\ell-1$ for level ℓ at state θ_ℓ^j is rejected, the initial state for the coarse subchain $\theta_{\ell-1}^0$ is set back to θ_ℓ^j. Hence, the new coarse subchain, which will generate the next proposal for level ℓ, is initialized from the same state as the previous subchain.

For MLMCMC [15], even if the coarse proposal is rejected, the coarse chain continues independently of the fine chain. In analogy to the subchain picture in MLDA, this corresponds to initializing the subchain on level $\ell-1$ with the coarse state ψ_{C} that has just been rejected on level ℓ. As a result, coarse and fine chains will separate and only recoalesce once a coarse proposal is accepted at the fine level. This choice provides better mixing at coarse levels and allows for efficient parallelization of the MLMCMC algorithm [45], but it does entail one important caveat: The practical algorithm in [15, Algorithm 3] does not necessarily define a Markov process unless coarse proposals passed to the next finer level are independent, as in [15, Algorithm 2]. The practical implication of violating this requirement is that we do not have a proof of convergence of MLMCMC with finite subchains because we cannot apply the theorems that guarantee convergence for homogeneous Markov chains. Indeed, numerical experiments (not shown) indicate that estimates using MLMCMC with finite subchains are biased and that the underlying chains do not converge to the desired target distributions.

Accordingly, in theory the practical multilevel estimator proposed by Dodwell et al. [15, Algorithm 3] is unbiased only if the coarse proposal is an independent sample from $\pi_{\ell-1}$, therefore only at infinite computational cost (i.e., when the subchain length goes to infinity). However, if the fixed subchain length is chosen to be greater than twice the integrated autocorrelation length of the chain at that level, in practice this bias disappears. This imposes the constraint that the subchain length might have to be fairly long. If the acceptance rate is also relatively low, the method becomes computationally inefficient, i.e., a lot of computational effort has to be put into generating independent proposals from a coarse distribution, which are then rejected with high probability.

2.3. A multilevel estimator and variance reduction. Using the MLDA sampler proposed above, it is in fact possible to define an asymptotically unbiased multilevel estimator that

retains most of the computational benefits of both multilevel Monte Carlo [19] and MLMCMC [15]. Let $Q_\ell(\theta_\ell)$ define some quantity of interest computed on level $\ell = 0, \ldots, L$. The aim is to estimate $\mathbb{E}_{\pi_L}[Q_L]$—the expectation of Q_L with respect to the posterior distribution π_L on the finest level L—using as little computational effort as possible.

The idea of multilevel Monte Carlo is, at its heart, very simple. The key is to avoid estimating the expected value $\mathbb{E}_\ell[Q_\ell]$ directly on level ℓ, but instead to estimate the correction with respect to the next lower level. Under the assumption that samples on level $\ell - 1$ are cheaper to compute than on level ℓ and that the variance of the correction term is smaller than the variance of Q_ℓ itself, the cost of computing this estimator is much lower than an estimator defined solely on samples from level ℓ. In the context of MLDA and MLMCMC, the target density π_ℓ depends on ℓ, so that we write

$$\mathbb{E}_{\pi_L}[Q_L] = \mathbb{E}_{\pi_0}[Q_0] + \sum_{\ell=1}^{L} \Big(\mathbb{E}_{\pi_\ell}[Q_\ell] - \mathbb{E}_{\pi_{\ell-1}}[Q_{\ell-1}] \Big), \tag{2.19}$$

which is achieved by adding and subtracting $\mathbb{E}_{\pi_\ell}[Q_\ell]$ for all levels $\ell = 0, \ldots, L-1$. Note that for the particular case where the densities $\{\pi_\ell\}_{\ell=0}^{L}$ are all equal, this reduces to the simple telescoping sum forming the basis of standard multilevel Monte Carlo [19].

The practical MLMCMC algorithm in [15, Algorithm 3] now proceeds by estimating the first term in (2.19) using the MCMC estimator $E_{\pi_0}[Q_0] \approx \frac{1}{N_0} \sum_{i=1}^{N_0} Q_0(\theta_0^i)$ with a Markov chain $[\theta_0^1, \ldots, \theta_0^{N_0}]$ produced with a standard Metropolis–Hastings on the coarsest level. Each of the correction terms for $\ell \geq 1$ is estimated by

$$\mathbb{E}_{\pi_\ell}[Q_\ell] - \mathbb{E}_{\pi_{\ell-1}}[Q_{\ell-1}] \approx \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} Q_\ell(\theta_\ell^i) - Q_{\ell-1}(\theta_{\ell-1}^{J_\ell i}), \tag{2.20}$$

where N_ℓ is the total number of samples on level ℓ after subtracting burn-in, J_ℓ is the subchain length on level $\ell - 1$, and $\theta_{\ell-1}^{J_\ell i}$ is the state of the coarse chain used as the proposal for the ith state of the fine chain in the MLMCMC algorithm. As mentioned in section 2.2.4, this multilevel estimator is unbiased only for MLMCMC as $J_\ell \to \infty$ or, in practice, for coarse subchains with J_ℓ greater than twice the integrated autocorrelation length.

An unbiased multilevel estimator can be produced using MLDA, without this constraint on the subchain lengths. However, since the levels of MLDA are strongly coupled and the coarse levels are consecutively realigned with the next finer, this is nontrivial. We achieve it by employing a particular form of the RST algorithm, Algorithm 3, in the MLDA algorithm, Algorithm 5. For all $\ell = 1, \ldots, L$, we set the probability mass function over the subchain length on level $\ell - 1$ to the discrete uniform distribution $p_\ell = \mathcal{U}(\{1, 2, \ldots, J_\ell\})$, where J_ℓ is the maximum subchain length. Hence, the jth proposal $\psi_C = \psi_{\ell-1}^j$ for the coarse modes on level ℓ in this version of MLDA constitutes an independent, uniformly-at-random draw from a subchain of length J_ℓ on level $\ell - 1$. Crucially, we let the coarse sampler continue sampling beyond the proposed state to produce subchains of fixed length J_ℓ for each state of the fine chain. Moreover, we also evaluate and store the quantity of interest at each state of each of those subchains on level $\ell - 1$.

Thus, using MLDA in this way to compute a chain $[\theta_L^1, \ldots, \theta_L^N]$ on the finest level L, in addition to the

$$N_L = N \text{ samples } \quad Q_L(\theta_L^1), \ldots, Q_L(\theta_L^{N_L}) \quad \text{on level } L,$$

we obtain also

$$N_\ell = N \times \prod_{k=\ell}^{L-1} J_{k+1} \text{ samples } \quad Q_\ell(\theta_\ell^1), \ldots, Q_\ell(\theta_\ell^{N_\ell}) \quad \text{on levels } \ell = 0, \ldots, L-1.$$

Using those samples the following asymptotically unbiased MLDA estimator of the posterior expectation $\mathbb{E}_{\pi_L}[Q_L]$ can be defined:

$$\widehat{Q}_L := \frac{1}{N_0} \sum_{i=1}^{N_0} Q_0(\theta_0^i) + \sum_{\ell=1}^{L} \frac{1}{N_\ell} \sum_{j=1}^{N_\ell} Q_\ell(\theta_\ell^j) - Q_{\ell-1}(\psi_{\ell-1}^j). \tag{2.21}$$

Here, $\psi_{\ell-1}^j$ denotes the proposal ψ_C for the coarse modes of the jth state θ_ℓ^j of the Markov chain on level ℓ produced by MLDA in Algorithm 5.

Let us first discuss why this estimator is asymptotically unbiased. For each j, the proposals ψ_{l-1}^j are independently and uniformly drawn from the subchain $[\theta_{l-1}^k : (j-1)J_\ell < k \leq jJ_\ell]$. Thus, the ensemble $\{\psi_{l-1}^1, \ldots, \psi_{l-1}^{N_\ell}\}$ is a random draw from $\{\theta_{l-1}^1, \ldots, \theta_{l-1}^{N_{\ell-1}}\}$ and thus identically distributed. As a consequence, in the limit as $N_\ell \to \infty$ for all ℓ, most terms on the right-hand side of (2.21) cancel. What remains is $\sum_{j=1}^{N_L} Q_L(\theta_L^j)$, which due to Theorem 2.6 is an unbiased estimator for $\mathbb{E}_{\pi_L}[Q_L]$ in the limit as $N_L \to \infty$.

Since the coarse subsamplers in MLDA are repeatedly realigned with the next finer distribution by way of the MLDA transition kernel, the samples on the coarse levels are in fact not distributed according to the "vanilla" densities $\{\pi_\ell\}_{\ell=0}^{L-1}$, but come from some "hybrid" mixture distributions. With the particular choice for p_ℓ, the density of the mixture distribution arising from subsampling the coarse density on level $\ell - 1 < L$ can be written

$$\tilde{\pi}_{\ell-1} = \frac{1}{J_\ell} \sum_{n=1}^{J_\ell} K_{\ell-1}^n \, \tilde{\pi}_{\ell,C}, \tag{2.22}$$

where $\tilde{\pi}_{\ell,C}$ is the marginal density of the coarse modes of the next finer density, $K_{\ell-1}$ is the transition kernel simulated by each step of subsampling on level $\ell - 1$, and $K_{\ell-1}^n$ is that kernel composed with itself n times. Recall again that according to Theorem 2.6 the finest sampler targets the exact posterior, so that $\tilde{\pi}_L = \pi_L$. Thus, the MLDA estimator in (2.21) approximates the following telescoping sum:

$$\mathbb{E}_{\pi_L}[Q_L] = \mathbb{E}_{\tilde{\pi}_0}[Q_0] + \sum_{\ell=1}^{L} \Big(\mathbb{E}_{\tilde{\pi}_\ell}[Q_\ell] - \mathbb{E}_{\tilde{\pi}_{\ell-1}}[Q_{\ell-1}] \Big), \tag{2.23}$$

which is a small but crucial difference to the sum in (2.19) that forms the basis of MLMCMC [15].

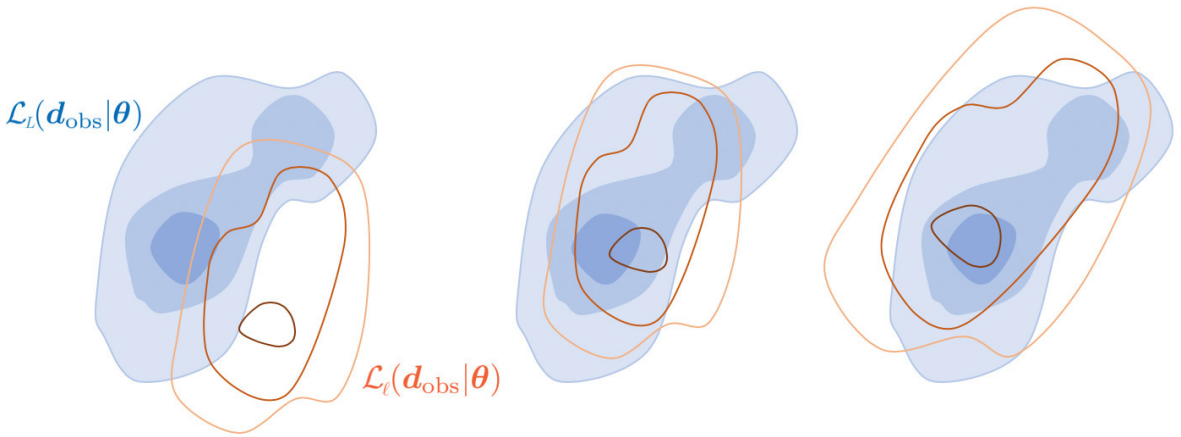

Figure 3. *Effect of applying the Gaussian AEM. The first panel shows the initial state before adaptation, where the coarse likelihood function ($\mathcal{L}_\ell(\mathbf{d}_{obs}|\theta)$, red isolines) approximates the fine likelihood function ($\mathcal{L}_L(\mathbf{d}_{obs}|\theta)$, blue contours) poorly. The second panel shows the effect of adding the mean of the bias to the likelihood functional, resulting in an offset of the coarse model likelihood function. The third panel shows the effect of also adding the covariance of the bias to the likelihood functional, resulting in a scaling and rotation of the coarse likelihood function. Adapted from* [34].

The computational gains due to multilevel variance reduction remain. In fact, since the mixture densities $\tilde{\pi}_{\ell-1}$ are conditioned every J_ℓ steps on the next finer chain, they are even closer and thus the variances of the correction terms in (2.21) will be further reduced compared to the variances of the estimates in (2.20). The fixed subchain lengths J_ℓ and thus the numbers of samples N_ℓ on the coarser levels can then be chosen as usual in multilevel Monte Carlo approaches to minimize the total variance for a fixed computational budget, or to minimize the cost to achieve the smallest variance. We are not going to go into more depth with respect to this estimator in this paper, but refer to, e.g., [9, 15, 20] for detailed analyses of multilevel (Markov chain) Monte Carlo estimators.

2.4. Adaptive correction of the approximate posteriors to improve efficiency. While the algorithm outlined in section 2.2 does guarantee sampling from the exact posterior, there are situations where convergence can be prohibitively slow. When the coarse model approximations are poor, the second-stage acceptance probability can be low, and many proposals will be rejected. This will result in suboptimal acceptance rates, poor mixing, and low effective sample sizes. The leftmost panel in Figure 3 shows a contrived example where the approximate likelihood function (red isolines) is offset from the exact likelihood function (blue contours) and its scale, shape, and orientation are incorrect.

One way to alleviate this problem is through *tempering*, where the variance in the likelihood function $\mathbf{\Sigma}_\epsilon$ on levels $\ell < L$ is inflated, resulting in a wider approximate posterior distribution. While this approach would allow the approximate posterior to encapsulate the exact posterior, it does not tackle the challenge in an intelligent fashion, and the inflation factor introduces an additional tuning parameter.

In place of tempering, an enhanced AEM can be employed to account for discrepancies between model levels. Let $\mathcal{F}_\ell$ denote the coarse forward map on level ℓ and let $\mathcal{F}_L$ denote the

forward map on the finest level L. To obtain a better approximation of the data d using $\mathcal{F}_\ell$, the two-level AEM suggested in [13] and analyzed in [14, 17] is extended here by adding a telescopic sum of the differences in the forward model output across all levels from ℓ to L,

$$d = \mathcal{F}_L(\theta) + \epsilon = \mathcal{F}_\ell(\theta) + \mathcal{B}_\ell(\theta) + \epsilon \quad \text{with} \quad \mathcal{B}_\ell(\theta) := \sum_{k=\ell}^{L-1} \underbrace{\mathcal{F}_{k+1}(\theta) - \mathcal{F}_k(\theta)}_{:=B_k(\theta)}, \tag{2.24}$$

denoting the bias on level ℓ at θ. The trick in the context of MLDA is that since $\mathcal{B}_\ell$ is just a simple sum, the individual bias terms B_k from pairs of adjacent model levels can be estimated independently, so that new information can be exploited each time *any* set of adjacent levels are evaluated for the same parameter value θ.

Approximating each individual bias term $B_k = \mathcal{F}_{k+1} - \mathcal{F}_k$ with a multivariate Gaussian $B_k^* \sim \mathcal{N}(\mu_k, \mathbf{\Sigma}_k)$, the total bias $\mathcal{B}_\ell$ can be approximated by the Gaussian $\mathcal{B}_\ell^* \sim \mathcal{N}(\mu_{\mathcal{B},\ell}, \mathbf{\Sigma}_{\mathcal{B},\ell})$ with $\mu_{\mathcal{B},\ell} = \sum_{k=\ell}^{L-1} \mu_k$ and $\mathbf{\Sigma}_{\mathcal{B},\ell} = \sum_{k=\ell}^{L-1} \mathbf{\Sigma}_k$.

The bias-corrected likelihood function for level ℓ is then proportional to

$$\mathcal{L}_\ell(\mathbf{d}|\theta) = \exp\left(-\frac{1}{2}(\mathcal{F}_\ell(\theta) + \mu_{\mathcal{B},\ell} - \mathbf{d})^T(\mathbf{\Sigma}_{\mathcal{B},\ell} + \mathbf{\Sigma}_e)^{-1}(\mathcal{F}_\ell(\theta) + \mu_{\mathcal{B},\ell} - \mathbf{d})\right). \tag{2.25}$$

The AEM, suggested by [29], is constructed offline, by sampling from the prior distribution before running the MCMC; we simply sample N parameter sets from the prior and compute the sample moments according to

$$\mu_k = \frac{1}{N}\sum_{i=1}^{N} B_k(\theta^{(i)}) \quad \text{and} \quad \mathbf{\Sigma}_k = \frac{1}{N-1}\sum_{i=1}^{N}(B_k(\theta^{(i)}) - \mu_k)(B_k(\theta^{(i)}) - \mu_k)^T. \tag{2.26}$$

However, this approach requires significant investment prior to sampling and may result in a suboptimal error model, since the bias in the posterior distribution is very different from the bias in the prior when the data is informative. Instead, as suggested in [13], an estimate for B_k can be constructed iteratively during sampling, using the following recursive formulae for sample means and sample covariances [22]:

$$\mu_{k,i+1} = \frac{1}{i+1}\Big(i\mu_{k,i} + B_k(\theta^{i+1})\Big) \quad \text{and} \tag{2.27}$$

$$\mathbf{\Sigma}_{k,i+1} = \frac{i-1}{i}\mathbf{\Sigma}_{k,i} + \frac{1}{i}\Big(i\mu_{k,i}\,\mu_{k,i}^T - (i+1)\mu_{k,i+1}\,\mu_{k,i+1}^T + B_k(\theta^{i+1})\,B_k(\theta^{i+1})^T\Big). \tag{2.28}$$

While this approach in theory results in an MCMC algorithm that is not Markov, the recursively constructed sample moments converge as sampling proceeds and hence the approach exhibits *diminishing adaptation* and *bounded convergence,* which is sufficient to ensure ergodicity for adaptive MCMC schemes [41, 42]. As shown in [14], it is also possible to construct a *state-dependent* AEM, where the coarse samples are corrected only according to the bias of the state of the MCMC, rather than the mean of the bias. This approach, however, may require a different form of the multilevel acceptance probability (equation (2.17)), which we have not

yet established, as discussed in section 2.2. We remark that while the simple Gaussian error model described here does suffer from a limited expressiveness, it is robust. Any coarse-level bias that is nonlinear in the model parameters will be absorbed by the respective covariance term, which will allow the coarse levels to sample "broader" and certainly encapsulate the true posterior. The general bias modeling framework described by (2.24) allows for the bias terms to be modeled by any functions of the model parameters, including Gaussian processes, artificial neural networks, polynomial chaos expansions, etc., as long as they either are constructed a priori or exhibit diminishing adaptation and bounded convergence. However, the Gaussian model proposed here does not require any tuning or caching of the bias history and is both computationally cheap and numerically robust. Hence, unless a particular problem strongly favors a different bias modeling approach, we recommend the Gaussian model described above.

3. Examples.

In this section, we consider three inverse problems which demonstrate the efficiency gains obtained by using MLDA, as well as by the extensions outlined above. The algorithm has been included in the free and open source probabilistic programming library `PyMC3`[1] as the `MLDA` step method since version 3.10.0, and the examples below were all completed using this implementation.

3.1. Gravitational survey.

In this example, we consider a two-dimensional gravity surveying problem, adapted from the one-dimensional problem presented in [24]. Our aim is to recover an unknown two-dimensional mass density distribution $f(\mathbf{t})$ at a known depth d below the surface from measurements $g(\mathbf{s})$ of the vertical component of the gravitational field at the surface. The contribution to $g(\mathbf{s})$ from infinitesimally small areas of the subsurface mass distribution are given by

$$\mathrm{d}g(\mathbf{s}) = \frac{\sin\theta}{r^2} f(\mathbf{t})\,\mathrm{d}\mathbf{t}, \tag{3.1}$$

where θ is the angle between the vertical plane and a straight line between two points $\mathbf{t}$ and $\mathbf{s}$, and $r = \|\mathbf{s}-\mathbf{t}\|_2$ is the Euclidean distance between the points. We exploit that $\sin\theta = d/r$, so that

$$\frac{\sin\theta}{r^2} f(\mathbf{t})\,\mathrm{d}\mathbf{t} = \frac{d}{r^3} f(\mathbf{t})\,\mathrm{d}\mathbf{t} = \frac{d}{\|\mathbf{s}-\mathbf{t}\|_2^3} f(\mathbf{t})\,\mathrm{d}\mathbf{t}. \tag{3.2}$$

This yields the integral equation

$$g(\mathbf{s}) = \int\int_T \frac{d}{\|\mathbf{s}-\mathbf{t}\|_2^3} f(\mathbf{t})\,\mathrm{d}\mathbf{t}, \tag{3.3}$$

where $T = [0,1]^2$ is the domain of the function $f(\mathbf{t})$. This constitutes our forward model. We solve the integral numerically using midpoint quadrature. For simplicity, we use m quadrature points along each dimension, so that in discrete form our forward model becomes

$$g(\mathbf{s}_i) = \sum_{l=1}^{m} \omega_l \sum_{k=1}^{m} \omega_k \frac{d}{\|\mathbf{s}_i - \mathbf{t}_{k,l}\|_2^3} \hat{f}(\mathbf{t}_{k,l}) = \sum_{j=1}^{m^2} \omega_j \frac{d}{\|\mathbf{s}_i - \mathbf{t}_j\|_2^3} \hat{f}(\mathbf{t}_j), \tag{3.4}$$

[1] https://docs.pymc.io/en/v3/index.html.

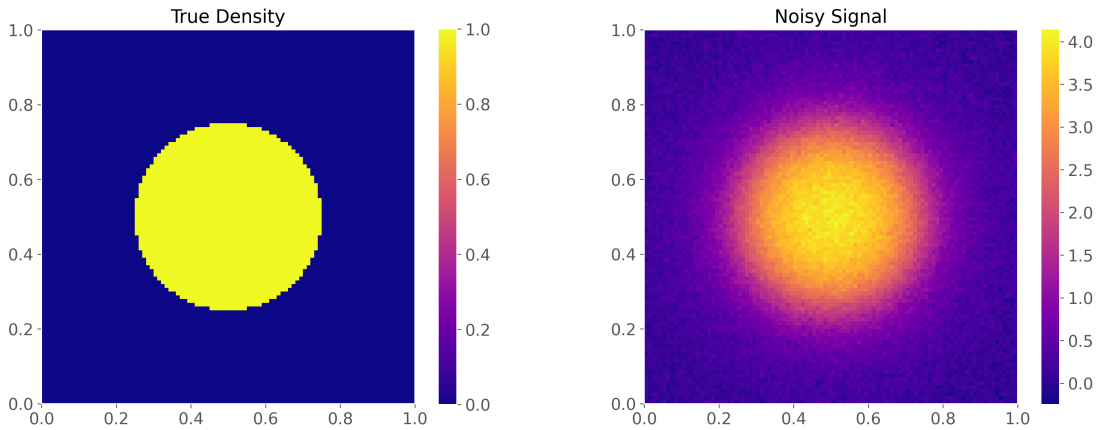

Figure 4. *(Left) The "true" mass density $f(t)$. (Right) The noisy signal at $d = 0.1$, with $\sigma_\epsilon = 0.1$.*

where $\omega_j = 1/m^2$ are the quadrature weights, $\hat{f}(\mathbf{t}_j)$ is the approximate subsurface mass at the quadrature points $\mathbf{t}_j,\ j = 1, \ldots, m^2$, and $g(\mathbf{s}_i)$ is the surface measurement at the collocation point $\mathbf{s}_i,\ i = 1, \ldots, n^2$. Hence, when $n > m$, we are dealing with an overdetermined problem and vice versa. This can be expressed as a linear system $\mathbf{Ax} = \mathbf{b}$, where

$$a_{ij} = \omega_j \frac{d}{\|\mathbf{s}_i - \mathbf{t}_j\|_2^3}, \quad x_j = \hat{f}(\mathbf{t}_j), \quad b_i = g(\mathbf{s}_i). \tag{3.5}$$

Due to the ill-posedness of the underlying, continuous inverse problem, the matrix $\mathbf{A}$ is very ill-conditioned, which entails numerical instability and spurious, often oscillatory, naive solutions for noisy right-hand sides. A problem of this type is traditionally solved by way of *regularization* such as Tikhonov regularization or truncated singular value decomposition, but it can also be handled in a more natural and elegant fashion as a Bayesian inverse problem.

For the exerimental set-up, a "true" mass density distribution $f(t)$ was assigned on T at a depth of $d = 0.1$ (Figure 4, left panel). The modeled signal was then discretized with $m = n = 100$ and perturbed with white noise with standard deviation $\sigma_\epsilon = 0.1$ (Figure 4, right panel) to be used as synthetic data in the numerical experiment.

The unknown mass density distribution was modeled as a Gaussian random process with a Matérn 3/2 covariance kernel [38]:

$$C_{3/2}(x, y) = \sigma^2 \left(1 + \frac{\sqrt{3}\|x - y\|_2}{\lambda}\right) \exp\left(-\frac{\sqrt{3}\|x - y\|_2}{\lambda}\right) \quad \text{for} \quad x, y \in D, \tag{3.6}$$

where λ is the covariance length scale and σ^2 is the variance. The random field was parametrized using a truncated Karhunen–Loève (KL) expansion of $f(t)$, i.e., an expansion in terms of a finite set of independent, standard Gaussian random variables $\theta_i \sim \mathcal{N}(0, 1),\ i = 1, \ldots, R$, given by

$$f(t, \omega) = \sum_{i=1}^{R} \sqrt{\mu_i}\phi_i(t)\theta_i(\omega). \tag{3.7}$$

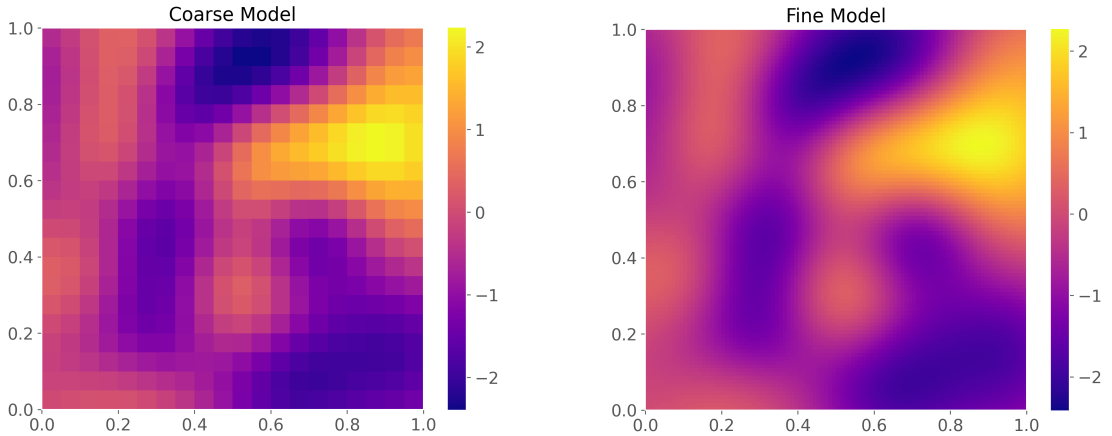

Figure 5. *Random realizations of the Matérn 3/2 random process prior, used to model the unknown mass density for the coarse model with $m = 20$ (left) and the fine model with $m = 100$ (right).*

Here, $\{\mu_i\}_{i\in\mathbb{N}}$ are the sequence of strictly decreasing real, positive eigenvalues, and $\{\phi_i\}_{i\in\mathbb{N}}$ the corresponding L^2-orthonormal eigenfunctions of the covariance operator with kernel $C_{3/2}(x, y)$.

A model hierarchy consisting of two model levels, with $m = 100$ and $m = 20$, respectively, was created. A Matern 3/2 random process with $l = 0.2$ and $\sigma^2 = 1$ was initialized on the fine model level and parametrized using KL decomposition, which was then truncated to encompass its $R = 32$ highest energy eigenmodes. It was then projected to the coarse model space (Figure 5).

Thus, the prior distribution of the model parameters $(\theta_i)_{i=1}^R$ is $\mathcal{N}(0, I_R)$. To sample from the posterior distribution of these parameters and thus to estimate the posterior mean conditioned on the synthetic data, we used the TLDA sampler with a random walk Metropolis–Hastings (RWMH) sampler on the coarse level. We ran two independent chains, each with 20000 draws, a burn-in of 5000, and a subchain length on the coarse level of 10. We also ran two chains using a single level RWMH sampler on the fine level with otherwise identical settings, but with no subchains. Each chain was initialized at the maximum a posteriori point.

While RWMH converged to the same parameter estimates as MLDA, RWMH exhibited inferior mixing (Figure 6) and fewer effective samples per second (Figure 7), particularly for the higher KL coefficients.

3.2. Predator-prey model. The Lotka–Volterra model describes the interaction between populations of prey (N) and predators (P) over time [44]. Their interaction is described by the system of nonlinear, first order, ordinary differential equations (ODEs)

$$\frac{dN}{dt} = aN - bNP \quad \text{and} \quad \frac{dP}{dt} = cNP - dP \quad \text{for } t > 0. \tag{3.8}$$

The model outputs are fully described by the parameters

$$\theta = \{N_0, P_0, a, b, c, d\},$$

which include the initial densities of prey and predators at time $t = 0$, and ecological parameters a, b, c, d, where broadly a is the birth rate of the prey, b is the encounter rate between prey

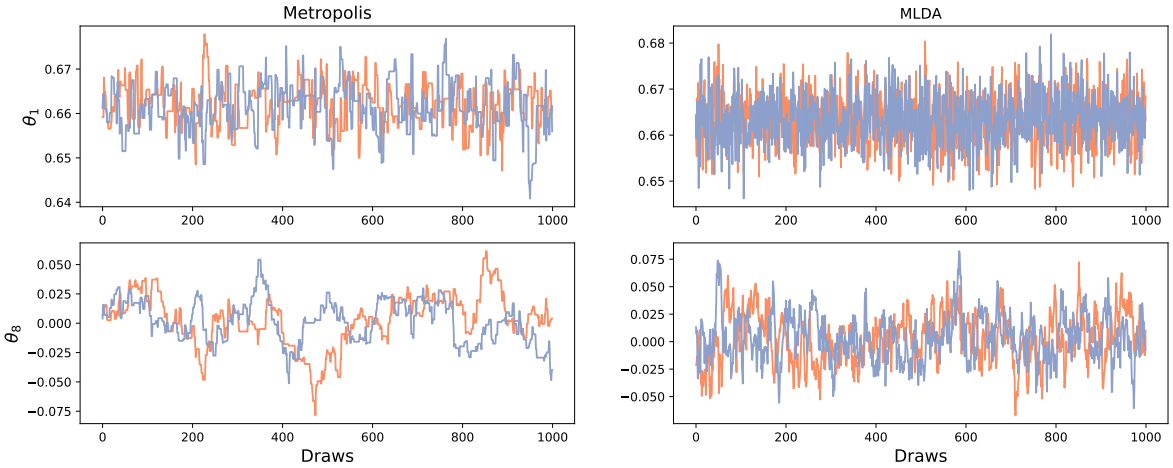

Figure 6. *Traces of θ_1 (top row) and θ_8 for RWMH (left column) and MLDA (right column), respectively. Different colors represent the independent chains.*

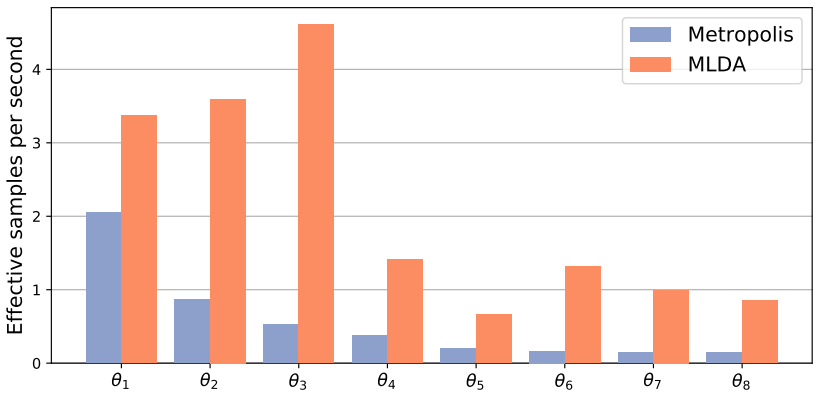

Figure 7. *Algorithmic performance measured in ES/s (effective samples per second) for the eight highest energy KL coefficients $\theta_k, k = 1, \ldots, 8$, for both RWMH (blue) and MLDA (red).*

and predators, c is the growth rate for the predators, and d is the death rate of the predators. For further details on their physical interpretation see, for example, [3].

In this example, we wish to infer the distribution of θ, given noisy observations of prey and predator densities at discrete time intervals, i.e., $N(t^\star)$ and $P(t^\star)$ for $t^\star \in \mathcal{T}$, where $\mathcal{T} = [0, 12]$ is the domain. The observations are again synthetically generated by solving (3.8) with the "true" parameters

$$\theta^\star = \{10.0, 5.0, 3.0, 0.7, 0.2, 1.0\}$$

and perturbing the calculated values $N(t^\star)$ and $P(t^\star)$ with independent Gaussian noise $\epsilon \sim \mathcal{N}(0, 1)$ (Figure 8). Our aim is to predict the mean density of predators $\mathbb{E}(P)$ over the same period.

The solutions of the ODE system in (3.8) can be approximated by a suitable numerical integration scheme. We use an explicit, adaptive Runge–Kutta method of order 5(4) [46]. For

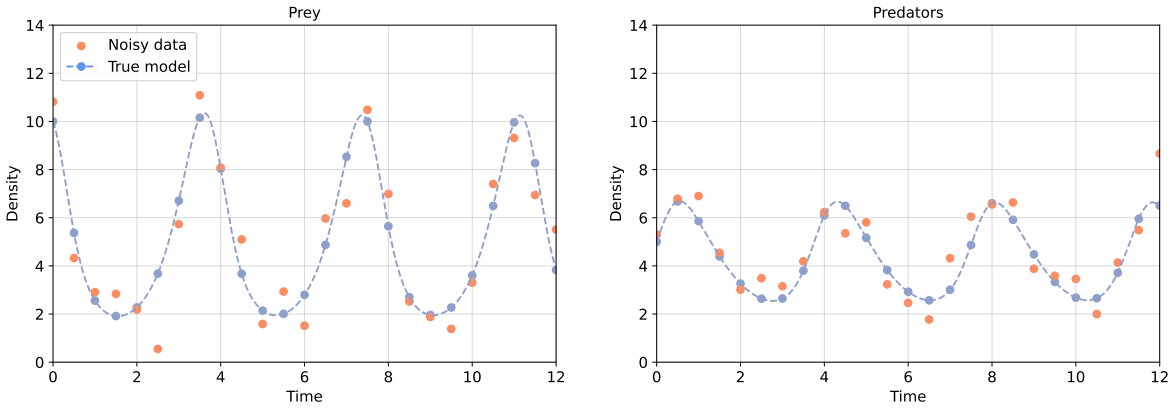

Figure 8. *The true (blue) and measured (red) densities of prey (left) and predators (right).*

the finest level $\ell = 2$, we integrate over the entire time domain $\mathcal{T}_2 = [0, 12]$ and use the entire data set to compute the likelihood function, while for the coarse levels, we stop integration early, so that $\mathcal{T}_1 = [0, 8]$ and $\mathcal{T}_0 = [0, 4]$, and use only the corresponding subsets of the data to compute the likelihood functions.

We assume that we possess some prior knowledge about the parameters and use informed priors $N_0 \sim \mathcal{N}(10.8, 1)$, $P_0 \sim \mathcal{N}(5.3, 1)$, $a \sim \mathcal{N}(2.5, 0.5)$, $b \sim$ Inv-Gamma$(1.0, 0.5)$, $c \sim$ Inv-Gamma$(1.0, 0.5)$, and $d \sim \mathcal{N}(1.2, 0.3)$.

To demonstrate the multilevel variance reduction feature, we ran the MLDA sampler with randomization of the subchain length as described in section 2.3 and then compared the (multilevel) MLDA estimator in (2.21), which uses both the coarse and fine samples, with a standard MCMC estimator based only on the samples produced by MLDA on the fine level. In both cases, we used the three-level model hierarchy as described above and employed the differential evolution Markov chain (DE-MC$_Z$) proposal [48] on the coarsest level. The coarsest level proposal kernel was automatically tuned during burn-in to achieve an acceptance rate between 0.2 and 0.5. The subchain lengths of $J_2 = J_1 = 10$ were chosen to balance the variances of the two contributions to the multilevel estimator (equation (2.21)), as for MLMC and MLMCMC.

Figure 9 shows the development of the total sampling error as the sampling progresses, for the sampler with and without variance reduction. Employing variance reduction clearly leads to a lower sampling error than the standard approach. Figure 10 shows the true prey and predator densities along with samples from the posterior distribution, demonstrating that the true model is encapsulated by the posterior samples, as desired.

3.3. Subsurface flow. In this example, a simple model problem arising in subsurface flow modeling is considered. Probabilistic uncertainty quantification is of interest in various situations, for example, in risk assessment of radioactive waste repositories. Moreover, this simple PDE model is often used as a benchmark for MCMC algorithms in the applied mathematics literature [36, 35, 15, 11, 10, 5]. The classical equations which govern steady-state single-phase subsurface flow in a confined aquifer are Darcy's law coupled with an incompressibility constraint

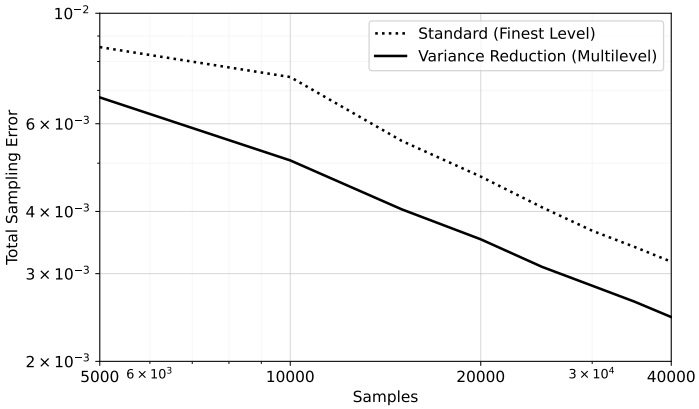

Figure 9. *Development of the total sampling error as sampling progresses for the sampler with (solid) and without (dashed) variance reduction.*

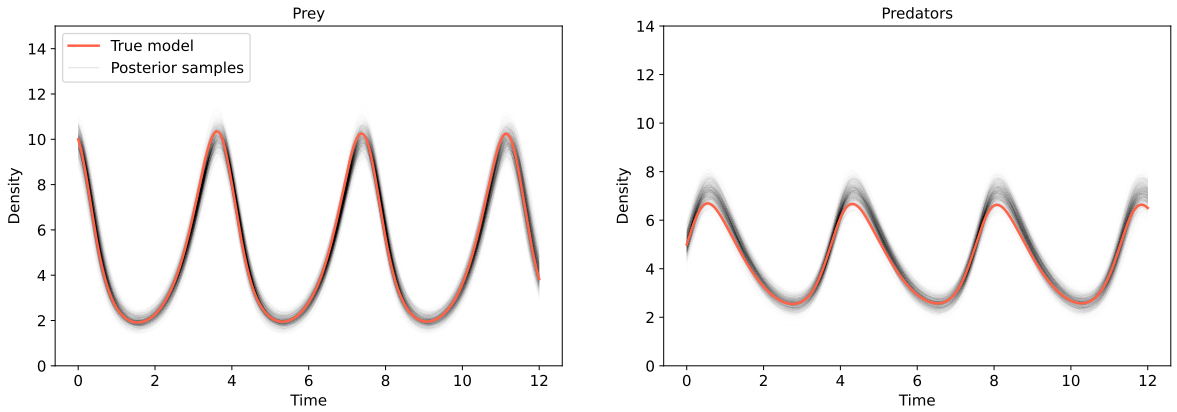

Figure 10. *True model (red) and posterior samples (black).*

$$w + k\nabla p = g \quad \text{and} \quad \nabla \cdot w = 0 \quad \text{in} \quad D \subset \mathbb{R}^d \tag{3.9}$$

for $d = 1, 2$, or 3, subject to suitable boundary conditions. Here p denotes the hydraulic head of the fluid, k the permeability tensor, w the flux, and g the source term.

A typical approach to treat the inherent uncertainty in this problem is to model the permeability as a random field $k = k(x, \omega)$ on $D \times \Omega$ for some probability space $(\Omega, \mathcal{A}, \mathbb{P})$. Therefore, (3.9) can be written as the following PDE with random coefficients:

$$-\nabla \cdot k(x, \omega)\nabla p(x, \omega) = f(x) \quad \text{for all} \quad x \in D, \tag{3.10}$$

where $f := -\nabla \cdot g$. As a synthetic example, consider the domain $D := [0, 1]^2$ with $f \equiv 0$ and deterministic boundary conditions

$$p|_{x_1=0} = 0, \quad p|_{x_1=1} = 1 \quad \text{and} \quad \partial_n p|_{x_2=0} = \partial_n p|_{x_2=1} = 0. \tag{3.11}$$

A widely used model for the prior distribution of the permeability in hydrology is a log-Gaussian random field [15, 12, 11, 5, 30], characterized by the mean of $\log k$, here chosen to be 0, and by its covariance function, here chosen to be

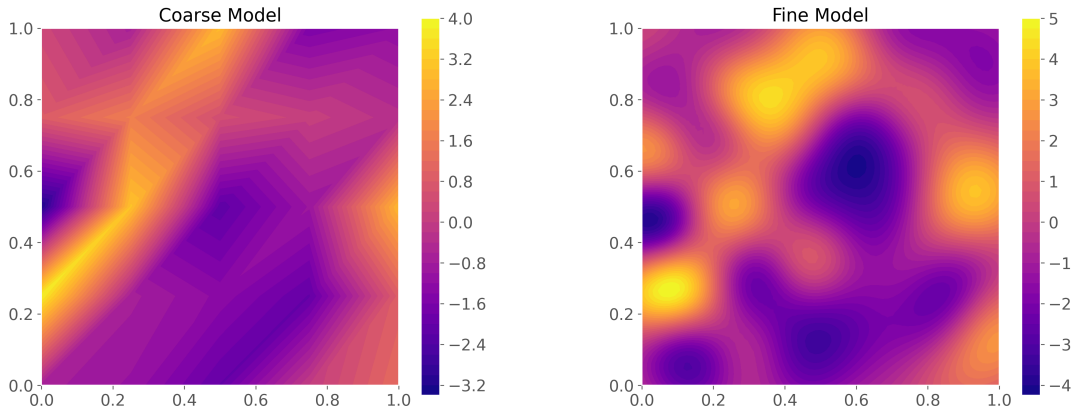

Figure 11. *True log-conductivity field of the coarsest model with m_0 grid points (left) and the finest model with m_2 grid points (right).*

$$C(x,y) := \sigma^2 \exp\left(-\frac{\|x-y\|_2^2}{2\lambda^2}\right), \quad \text{for} \quad x,y \in D, \tag{3.12}$$

with $\sigma = 2$ and $\lambda = 0.1$. Again, the log-Gaussian random field is parametrized using a truncated KL expansion of $\log k$, i.e., an expansion in terms of a finite set of independent, standard Gaussian random variables $\theta_i \sim \mathcal{N}(0,1)$, $i = 1, \ldots, R$, given by

$$\log k(x,\omega) = \sum_{i=1}^{R} \sqrt{\mu_i}\phi_i(x)\theta_i(\omega). \tag{3.13}$$

Again, $\{\mu_i\}_{i\in\mathbb{N}}$ are the sequence of strictly decreasing real, positive eigenvalues, and $\{\phi_i\}_{i\in\mathbb{N}}$ are the corresponding L^2-orthonormal eigenfunctions of the covariance operator with kernel $C(x,y)$. Thus, the prior distribution on the parameter $\theta = (\theta_i)_{i=1}^R$ in the stochastic PDE problem (equation (3.10)) is $\mathcal{N}(0, I_R)$. In this example we chose $R = 64$.

The aim is to infer the posterior distribution of θ, conditioned on measurements of p at $M = 25$ discrete locations $x^j \in D$, $j = 1, \ldots, M$, stored in the vector $d_{obs} \in \mathbb{R}^M$. Thus, the forward operator is $\mathcal{F} : \mathbb{R}^R \to \mathbb{R}^M$ with $\mathcal{F}_j(\theta_\omega) = p(x^j, \omega)$.

All finite element (FE) calculations were carried out with `FEniCS` [31], using piecewise linear FEs on a uniform triangular mesh. The coarsest mesh $\mathcal{T}_0$ consisted of $m_0 = 5$ grid points in each direction, while subsequent levels were constructed by two steps of uniform refinement of $\mathcal{T}_0$, leading to $m_\ell = 4^\ell(m_0 - 1) + 1$ grid points in each direction on the three grids $\mathcal{T}_\ell$, $\ell = 0, 1, 2$ (Figure 11).

To demonstrate the excellent performance of MLDA with the AEM, synthetic data was generated by drawing a sample from the prior distribution and solving (equation (3.10)) with the resulting realization of k on $\mathcal{T}_2$. To construct d_{obs}, the computed discrete hydraulic head values at $(x^j)_{j=1}^M$ were then perturbed by independent Gaussian noise, i.e., by a sample $\epsilon^* \sim \mathcal{N}(0, \Sigma_\epsilon)$ with $\Sigma_\epsilon = 0.01^2 I_M$.

To compare the "vanilla" MLDA approach to the AEM-enhanced version, we sampled the same model using identical sampling parameters, with and without AEM activated. For

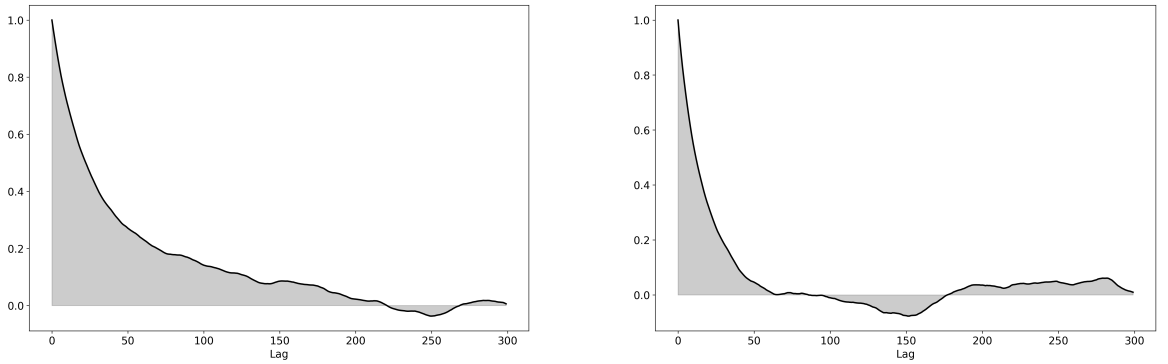

Figure 12. *Autocorrelation function for θ_1 for samples without AEM (left) and with AEM (right).*

each approach, we sampled two independent chains, each initialized at a random point from the prior. For each chain, we drew 20000 samples plus a burn-in of 5000. We used subchain lengths $J_0 = J_1 = 5$, since that produced the best trade-off between computation time and effective sample size for MLDA with the AEM. Note that the cost of computing the subchains on the coarser levels only leads to about a 50% increase in the total cost for drawing a sample on level L. The DE-MC$_Z$ proposal [48] was employed on the coarsest level with automatic step-size tuning during burn-in to achieve an acceptance rate between 0.2 and 0.5.

To assess the performance of the two approaches, the autocorrelation function (Figure 12) and the effective sample size for each parameter were computed [49]. Since the coarsest model was quite a poor approximation of the finest, running MLDA without the AEM yielded relatively poor results, with an average effective sample size of 326 out of 40000 samples, and strong autocorrelation. However, when the AEM was employed and otherwise using the exact same sampling parameters, we obtained an average effective sample size of 1012 out of 40000 samples, with correspondingly weaker autocorrelation.

Note that this particular numerical experiment was chosen to demonstrate the dramatic effect that employing the AEM can have in MLDA, thus making it possible to use multilevel sampling strategies with very crude approximate models. An FE mesh with 25 degrees of freedom is extremely coarse for a Gaussian random field with correlation length $\lambda = 0.1$, yet using the AEM it still provides an excellent surrogate for DA. Typically much finer models are used in real applications with longer subchains on the coarser levels (cf. [15]). The AEM will be less critical in that case and MLDA will also produce good ESS without the AEM.

4. Conclusions and future work. In this paper, we have presented an extension of state-independent DA MCMC [8], where a hierarchy of coarse MCMC samplers inform the finest sampler in a cascading fashion. If the models on the coarse levels are carefully designed, the approach can lead to significant computational savings, compared to standard single-level MCMC. A possible direction for future research would be to extend this approach further to the general DA context, where also state-dependent approximations are supported. We would like to highlight that the choice of proposal on the coarsest level is free, as long as it achieves irreducibility for the coarsest distribution. We have chosen relatively simple proposals for the

coarsest level, but if, e.g., the gradient of the likelihood function is available, one can also employ more advanced gradient-informed proposals, such as MALA, HMC, or NUTS.

The presented MLDA algorithm has clear similarities with MLMCMC [15], in that it allows for any number of coarse levels and extended subchains on the coarse levels, but unlike MLMCMC, it is Markov and asymptotically unbiased, also for finite-length subchains. To achieve this quality, the algorithm must be sequential, which complicates parallelization considerably. One remedy for this challenge, and a possible direction for future research, would be to employ prefetching of proposals [6]. The central idea of prefetching is to precompute proposal "branches" and evaluate those in parallel, since for each proposal there are only two options, namely *accept* or *reject*. Prefetching and evaluating entire proposal branches is significantly more computationally demanding than the strictly sequential approach and generates more waste, similar to multiple-try Metropolis [33], since entire branches will effectively be rejected at each step. Minimizing the waste of prefetching while maintaining the computational gains of parallelization constitutes a complex, probabilistic optimization problem. This could be addressed by controlling the prefetching length, e.g., using a reinforcement learning agent to learn an optimal policy, and to then hedge bets on valuable prefetching lengths, based on the latest sampling history.

A question that remains is the optimal choice of the subchain lengths $\{J_\ell\}_{\ell=1}^{L}$ for the coarse levels, which is essentially the only tuning parameter in the MLDA algorithm. A good rule of thumb may be to choose the length for any level such that the cost of creating the subchain corresponds to the cost of evaluating a single proposal on the next finer level, but this is not the most rigorous approach. The question has previously been studied in the context of MLMC [9] and MLMCMC [15] and involves either computing the optimal (effective) sample size for each level for a fixed acceptable sampling error or computing the sampling error corresponding to a fixed computational budget. A similar approach can be taken for MLDA, but with some caveats. First, the number of samples on each level is determined not only by the subchain length on that level, but by the number of samples on the next finer level. Hence, care must be taken when choosing the subchain lengths. Second, it is nontrivial to determine the effective sample size of a level a priori, because of the direct correspondence with the distribution on the next finer level by way of the MLDA acceptance criterion. One possible workaround would be to determine the optimal subchain lengths adaptively by empirically determining the effective sample sizes and variances on each level during burn-in. Similarly to the prefetching approach outlined above, these decisions could also be outsourced to a reinforcement learning agent that would adaptively learn the optimal policy for minimizing either cost or sampling error. We emphasize this question as a potential direction for future research.

Acknowledgments. MCMC sampling was completed using the `MLDA` sampler of the free and open source probabilistic programming library `PyMC3`. The `PyMC3` code is available at GitHub: https://github.com/pymc-devs/pymc. The examples shown in this paper are available at https://github.com/mikkelbue/MLDA_examples. The authors have no conflicts of interest to declare.

REFERENCES

[1] C. Andrieu and J. Thoms, *A tutorial on adaptive MCMC*, Statist. Comput., 18 (2008), pp. 343–373, https://doi.org/10.1007/s11222-008-9110-y.

[2] Y. F. Atchadé, *An adaptive version for the metropolis adjusted Langevin algorithm with a truncated drift*, Methodol. Comput. Appl., 8 (2006), pp. 235–254, https://doi.org/10.1007/s11009-006-8550-0.
[3] N. Bacaër, *A Short History of Mathematical Population Dynamics,* Springer, London, 2011, https://doi.org/10.1007/978-0-85729-115-8.
[4] A. Barth, C. Schwab, and N. Zollinger, *Multi-level Monte Carlo finite element method for elliptic PDEs with stochastic coefficients*, Numer. Math., 119 (2011), pp. 123–161, https://doi.org/10.1007/s00211-011-0377-0.
[5] A. Beskos, M. Girolami, S. Lan, P. E. Farrell, and A. M. Stuart, *Geometric MCMC for infinite-dimensional inverse problems*, J. Comput. Phys., 335 (2017), pp. 327–351, https://doi.org/10.1016/j.jcp.2016.12.041.
[6] A. E. Brockwell, *Parallel Markov chain Monte Carlo simulation by pre-fetching*, J. Comput. Graph. Statist., 15 (2006), pp. 246–261, https://doi.org/10.1198/106186006X100579.
[7] J. Charrier, R. Scheichl, and A. L. Teckentrup, *Finite element error analysis of elliptic PDEs with random coefficients and its application to multilevel Monte Carlo methods*, SIAM J. Numer. Anal., 51 (2013), pp. 322–352, https://doi.org/10.1137/110853054.
[8] J. A. Christen and C. Fox, *Markov chain Monte Carlo using an approximation*, J. Comput. Graph. Statist., 14 (2005), pp. 795–810, https://doi.org/10.1198/106186005X76983.
[9] K. A. Cliffe, M. B. Giles, R. Scheichl, and A. L. Teckentrup, *Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients*, Comput. Vis. Sci., 14 (2011), pp. 3–15, https://doi.org/10.1007/s00791-011-0160-x.
[10] P. R. Conrad, A. Davis, Y. M. Marzouk, N. S. Pillai, and A. Smith, *Parallel local approximation MCMC for expensive models*, SIAM/ASA J. Uncertain. Quantif., 6 (2018), pp. 339–373.
[11] P. R. Conrad, Y. M. Marzouk, N. S. Pillai, and A. Smith, *Accelerating asymptotically exact MCMC for computationally intensive models via local approximations*, J. Amer. Statist. Assoc., 111 (2016), pp. 1591–1607, https://doi.org/10.1080/01621459.2015.1096787.
[12] P. G. Constantine, C. Kent, and T. Bui-Thanh, *Accelerating Markov chain Monte Carlo with active subspaces*, SIAM J. Sci. Comput., 38 (2016), pp. A2779–A2805, https://doi.org/10.1137/15M1042127.
[13] T. Cui, C. Fox, and M. J. O'Sullivan, *Bayesian calibration of a large-scale geothermal reservoir model by a new adaptive delayed acceptance Metropolis Hastings algorithm: Adaptive delayed acceptance Metropolis-Hastings algorithm*, Water Resour. Res., 47 (2011), https://doi.org/10.1029/2010WR010352.
[14] T. Cui, C. Fox, and M. J. O'Sullivan, *A posteriori stochastic correction of reduced models in delayed-acceptance MCMC, with application to multiphase subsurface inverse problems: Stochastic correction of reduced models in delayed-acceptance MCMC*, Internat. J. Numer. Methods Engrg., 118 (2019), pp. 578–605, https://doi.org/10.1002/nme.6028.
[15] T. J. Dodwell, C. Ketelsen, R. Scheichl, and A. L. Teckentrup, *A hierarchical multilevel Markov chain Monte Carlo algorithm with applications to uncertainty quantification in subsurface flow*, SIAM/ASA J. Uncertain. Quantif., 3 (2015), pp. 1075–1108, https://doi.org/10.1137/130915005.
[16] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth, *Hybrid Monte Carlo*, Phys. Lett. B, 195 (1987), pp. 216–222, https://doi.org/10.1016/0370-2693(87)91197-X.
[17] C. Fox, T. Cui, and M. Neumayer, *Randomized reduced forward models for efficient Metropolis-Hastings MCMC, with application to subsurface fluid flow and capacitance tomography*, GEM Int. J. Geomath., 11 (2020), pp. 1–38.
[18] S. Geman and D. Geman, *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*, IEEE Trans. Pattern Anal. Mach. Intell., 6 (1984), pp. 721–741, https://doi.org/10.1109/TPAMI.1984.4767596.
[19] M. B. Giles, *Multilevel Monte Carlo path simulation*, Oper. Res., 56 (2008), pp. 607–617, https://doi.org/10.1287/opre.1070.0496.
[20] M. B. Giles, *Multilevel Monte Carlo methods*, Acta Numer., 24 (2015), pp. 259–328.
[21] M. Giry, *A categorical approach to probability theory*, in Categorical Aspects of Topology and Analysis, Springer, Berlin, 1982, pp. 68–85.
[22] H. Haario, E. Saksman, and J. Tamminen, *An adaptive metropolis algorithm*, Bernoulli, 7 (2001), pp. 223–242, https://doi.org/10.2307/3318737.

[23] A.-L. Haji-Ali, F. Nobile, L. Tamellini, and R. Tempone, *Multi-index stochastic collocation convergence rates for random PDEs with parametric regularity*, Found. Comput. Math., 16 (2016), pp. 1555–1605, https://doi.org/10.1007/s10208-016-9327-7.
[24] P. C. Hansen, *Discrete Inverse Problems: Insight and Algorithms*, SIAM, Philadelphia, 2010, https://doi.org/10.1137/1.9780898718836.
[25] W. K. Hastings, *Monte Carlo sampling methods using Markov chains and their applications*, Biometrika, 57 (1970), pp. 97–109.
[26] S. Heinrich, *Multilevel Monte Carlo methods*, in Proceedings of the Third International Conference on Large-Scale Scientific Computing-Revised Papers, London, 2001, Springer-Verlag, Berlin, pp. 58–67, http://dl.acm.org/citation.cfm?id=645740.666755.
[27] M. D. Hoffman and A. Gelman, *The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo*, J. Mach. Learn. Res., 15 (2014), pp. 1593–1623.
[28] A. Jasra, K. Kamatani, K. Law, and Y. Zhou, *A multi-index Markov chain Monte Carlo method*, Int. J. Uncertain. Quantif., 8 (2018), pp. 61–73.
[29] J. Kaipio and E. Somersalo, *Statistical inverse problems: Discretization, model reduction and inverse crimes*, J. Comput. Appl. Math., 198 (2007), pp. 493–504, https://doi.org/10.1016/j.cam.2005.09.027.
[30] S. Lan, *Adaptive dimension reduction to accelerate infinite-dimensional geometric Markov chain Monte Carlo*, J. Comput. Phys., 392 (2019), pp. 71–95, https://doi.org/10.1016/j.jcp.2019.04.043.
[31] H. P. Langtangen and A. Logg, *Solving PDEs in Python – The FEniCS Tutorial Volume* I, Simula SpringerBriefs on Computing, Springer, Cham, 2017.
[32] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*, Springer Ser. Statist., Springer, New York, 2004, https://doi.org/10.1007/978-0-387-76371-2.
[33] J. S. Liu, F. Liang, and W. H. Wong, *The multiple-try method and local optimization in metropolis sampling*, J. Amer. Statist. Assoc., 95 (2000), pp. 121–134, https://doi.org/10.1080/01621459.2000.10473908.
[34] M. B. Lykkegaard, T. J. Dodwell, and D. Moxey, *Accelerating uncertainty quantification of groundwater flow modelling using a deep neural network proxy*, Comput. Methods Appl. Mech. Engrg., 383 (2021), 113895, https://doi.org/10.1016/j.cma.2021.113895.
[35] Y. M. Marzouk and H. N. Najm, *Dimensionality reduction and polynomial chaos acceleration of Bayesian inference in inverse problems*, J. Comput. Phys., 228 (2009), pp. 1862–1902, https://doi.org/10.1016/j.jcp.2008.11.024.
[36] Y. M. Marzouk, H. N. Najm, and L. A. Rahn, *Stochastic spectral methods for efficient Bayesian solution of inverse problems*, J. Comput. Phys., 224 (2007), pp. 560–586, https://doi.org/10.1016/j.jcp.2006.10.010.
[37] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, *Equation of state calculations by fast computing machines*, J. Chem. Phys., 21 (1953), pp. 1087–1092, https://doi.org/10.1063/1.1699114.
[38] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning, Adaptive Computation and Machine Learning*, MIT Press, Cambridge, MA, 2006.
[39] G. O. Roberts and J. S. Rosenthal, *Optimal scaling of discrete approximations to Langevin diffusions*, J. R. Stat. Soc. Ser. B Statist. Methodol., 60 (1998), pp. 255–268, https://doi.org/10.1111/1467-9868.00123.
[40] G. O. Roberts and J. S. Rosenthal, *General state space Markov chains and MCMC algorithms*, Probab. Surv., 1 (2004), pp. 20–71, https://doi.org/10.1214/154957804100000024.
[41] G. O. Roberts and J. S. Rosenthal, *Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms*, J. Appl. Probab., 44 (2007), pp. 458–475, https://www.jstor.org/stable/27595854.
[42] G. O. Roberts and J. S. Rosenthal, *Examples of adaptive MCMC*, J. Comput. Graph. Statist., 18 (2009), pp. 349–367, https://doi.org/10.1198/jcgs.2009.06134.
[43] G. O. Roberts and R. L. Tweedie, *Exponential convergence of Langevin distributions and their discrete approximations*, Bernoulli, 2 (1996), pp. 341–363, https://doi.org/10.2307/3318418.
[44] L. L. Rockwood and J. W. Witt, *Introduction to Population Ecology*, 2nd ed., Wiley Blackwell, Chichester, UK, 2015.

[45] L. Seelinger, A. Reinarz, L. Rannabauer, M. Bader, P. Bastian, and R. Scheichl, *High performance uncertainty quantification with parallelized multilevel Markov chain Monte Carlo*, in Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, ACM, New York, 2021, https://doi.org/10.1145/3458817.3476150.

[46] S. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*, Studies in Nonlinearity, Westview Press, Cambridge, MA, 2007.

[47] A. L. Teckentrup, R. Scheichl, M. B. Giles, and E. Ullmann, *Further analysis of multilevel Monte Carlo methods for elliptic PDEs with random coefficients*, Numer. Math., 125 (2013), pp. 569–600.

[48] C. J. F. ter Braak and J. A. Vrugt, *Differential evolution Markov chain with snooker updater and fewer chains*, Statist. Comput., 18 (2008), pp. 435–446, https://doi.org/10.1007/s11222-008-9104-9.

[49] A. Vehtari, A. Gelman, D. Simpson, B. Carpenter, and P.-C. Bürkner, *Rank-normalization, folding, and localization: An improved $\widehat{R}$ for assessing convergence of MCMC*, Bayesian Anal., 16 (2020), pp. 667–718, https://doi.org/10.1214/20-BA1221.

[50] J. A. Vrugt, C. ter Braak, C. Diks, B. A. Robinson, J. M. Hyman, and D. Higdon, *Accelerating Markov chain Monte Carlo simulation by differential evolution with self-adaptive randomized subspace sampling*, Int. J. Nonlinear Sci. Numer. Simul., 10 (2009), pp. 273–290, https://doi.org/10.1515/IJNSNS.2009.10.3.273.

[51] Q. Zhou, Z. Hu, Z. Yao, and J. Li, *A hybrid adaptive MCMC algorithm in function spaces*, SIAM/ASA J. Uncertain. Quantif., 5 (2017), pp. 621–639, https://doi.org/10.1137/16M1082950.