SIAM/ASA J. UNCERTAINTY QUANTIFICATION
Vol. 6, No. 1, pp. 339–373

Parallel Local Approximation MCMC for Expensive Models*

Patrick R. Conrad†, Andrew D. Davis†, Youssef M. Marzouk†, Natesh S. Pillai‡, and Aaron Smith§

Abstract. Performing Bayesian inference via Markov chain Monte Carlo (MCMC) can be exceedingly expensive when posterior evaluations invoke the evaluation of a computationally expensive model, such as a system of PDEs. In recent work [*J. Amer. Statist. Assoc.*, 111 (2016), pp. 1591–1607] we described a framework for constructing and refining *local approximations* of such models during an MCMC simulation. These posterior-adapted approximations harness regularity of the model to reduce the computational cost of inference while preserving asymptotic exactness of the Markov chain. Here we describe two extensions of that work. First, we prove that samplers running in parallel can collaboratively construct a shared posterior approximation while ensuring ergodicity of each associated chain, providing a novel opportunity for exploiting parallel computation in MCMC. Second, focusing on the Metropolis-adjusted Langevin algorithm, we describe how a proposal distribution can successfully employ gradients and other relevant information extracted from the approximation. We investigate the practical performance of our approach using two challenging inference problems, the first in subsurface hydrology and the second in glaciology. Using local approximations constructed via parallel chains, we successfully reduce the run time needed to characterize the posterior distributions in these problems from days to hours and from months to days, respectively, dramatically improving the tractability of Bayesian inference.

Key words. Markov chain Monte Carlo, parallel computing, Metropolis-adjusted Langevin algorithm, Bayesian inference, approximation theory, local regression, surrogate modeling

AMS subject classifications. 65C40, 62F15, 60J22

DOI. 10.1137/16M1084080

1. Introduction.

Markov chain Monte Carlo (MCMC) is a powerful tool for performing Bayesian inference, but can be computationally prohibitive in many settings, especially when posterior density evaluations involve a computationally expensive step. For instance, applications in the physical sciences often require PDE *forward models*, evaluated using numerical solvers with nontrivial run times. When these solvers must be invoked with each posterior evaluation, direct sampling with MCMC can become intractable.

*Received by the editors July 11, 2016; accepted for publication (in revised form) December 4, 2017; published electronically March 27, 2018.

http://www.siam.org/journals/juq/6-1/M108408.html

Funding: The work of the first, second, and third authors was supported in part by the Scientific Discovery through Advanced Computing (SciDAC) program of the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research under award DE-SC0007099. The work of the fifth author was supported in part by the National Sciences and Engineering Research Council of Canada. The work of the fourth author was supported in part by the Office of Naval Research.

†Center for Computational Engineering and Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139 (prconrad@mit.edu, davisad@mit.edu, ymarz@mit.edu).

‡Department of Statistics, Harvard University, Cambridge, MA 02138 (pillai@stat.harvard.edu).

§Department of Mathematics and Statistics, University of Ottawa, Ottawa, ON, K1N 7N5, Canada (asmi28@uottawa.ca).

To reduce this computational burden, a standard approach is to construct an approximation or "surrogate" of the forward model or likelihood function, and then to sample from (or otherwise characterize) the posterior distribution induced by this approximation [45, 27, 41, 46, 36, 35, 2, 25, 29, 6, 15]. Although such approaches can be quite effective at reducing computational cost, they may be difficult to use in practice—in part because they separate the construction of the surrogate from the subsequent inference procedure. Approximation of the forward model biases posterior expectations [11] in a way that cannot easily be quantified. It is then difficult to decide how much computational effort should be devoted to surrogate construction, and how to balance the resulting biases with the statistical errors of posterior sampling. Alternatives such as delayed-acceptance MCMC [7, 14] yield asymptotically exact sampling but surrender potential speedups by requiring at least one evaluation of the forward model for each accepted sample. In recent work [9], we demonstrated that surrogate construction and posterior exploration can instead be *joined*, yielding a framework for incrementally and infinitely refining a surrogate during MCMC sampling. This framework allows the approximation to be tailored to the problem—e.g., made most accurate in regions of high posterior probability—while guaranteeing that the associated Markov chain asymptotically samples from the *exact* posterior distribution of interest. Empirical studies on problems of moderate dimension showed that the number of expensive posterior evaluations per MCMC step can be reduced by orders of magnitude, with no discernible loss of accuracy in posterior expectations.

This work describes two key extensions of the framework in [9]. First, we show that our approximation scheme enables a novel type of MCMC parallelism: concurrent chains can collaboratively develop a *shared* approximation. Effectively exploiting parallel computation in MCMC is often challenging because the core algorithm is inherently sequential, but our strategy directly deploys parallel resources to address the key performance bottleneck: the cost of repeatedly running the forward model.

Second, while our previous work showed how to build a convergent approximation of the target probability density, it did not support the idea of using this approximation to construct a proposal distribution. MCMC performance is highly dependent on the choice of proposal, but sophisticated proposals, such as the Metropolis-adjusted Langevin algorithm (MALA) and its manifold variants [20], can be expensive to apply because they require gradients (and possibly higher derivatives) of the forward model. This derivative information is often expensive or impossible to compute directly, but is trivial to extract from an approximation. Intuitively, it should then be possible to use our approximation framework to greatly reduce the costs of such proposals. Here we do exactly that, extending our previous theoretical results to show that the Monte Carlo estimates obtained by our algorithm converge to the correct value, as long as the convergence of our approximation to the target distribution yields convergence of the associated approximate Markov transition kernel in a suitably strong norm. As an example, we show how to use simplified manifold MALA within our local approximation scheme, and we prove that the resulting stochastic process is convergent in a representative case.

Finally, we construct two inference problems that are representative of interesting scientific queries, that involve computationally expensive forward models (such that naive use of the model in sampling would take days or months), and that have nontrivial posterior structure which must be characterized using MCMC. The first is a problem in groundwater hydrology,

where a subsurface conductivity field is inferred from observations of tracer transport; it is a more complex and realistic version of the linear elliptic PDE inverse problem [16], combining an elliptic equation for the hydraulic head with another PDE governing tracer dispersion [17, 37]. The second problem is drawn from glaciology: here we infer the basal friction parameters of a shallow-shelf ice stream model [31, 32, 33] from observations of surface ice velocity. Our numerical experiments evaluate MCMC efficiency, accuracy, and wall clock time, and benchmark the parallel performance of our algorithms. Results demonstrate the strong performance of our approach; for example, inference in the ice stream model becomes tractable, with the time needed to characterize the posterior reduced from roughly two months to just over a day.

The remainder of this paper is organized as follows. Section 2 reviews the basic algorithmic framework of local approximation MCMC. Section 3 presents and analyzes the shared construction of approximations for parallel MCMC. Section 4 describes the use of local approximations in the proposal scheme, and section 5 describes our numerical experiments. Proofs of the main convergence results, along with certain algorithmic details, are deferred to the appendices.

2. Review of local approximation MCMC.

We are interested in Bayesian inference problems with posterior densities of the form

$$p(\theta|\mathbf{d}) \propto \ell(\theta|\mathbf{d},\mathbf{f})p(\theta)$$

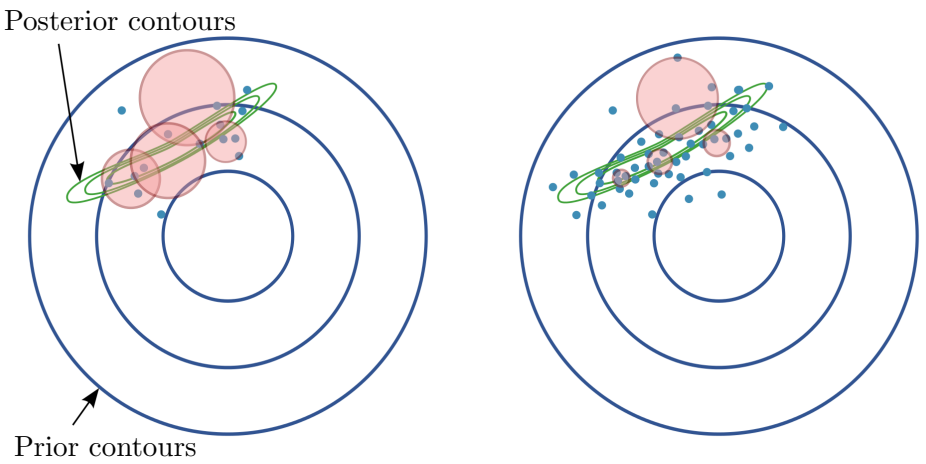
for parameters $\theta \in \Theta \subseteq \mathbb{R}^d$, data $\mathbf{d} \in \mathbb{R}^n$, a forward model $\mathbf{f} : \Theta \to \mathbb{R}^n$, and probability densities specifying the prior $p(\theta)$ and likelihood function ℓ. The forward model may enter the likelihood function in various ways. For instance, if $\mathbf{d} = \mathbf{f}(\theta) + \eta$, where η represents some measurement or model error with probability density p_η, then $\ell(\theta|\mathbf{d},\mathbf{f}) = p_\eta(\mathbf{d} - \mathbf{f}(\theta))$.

Assume that the forward model is both *computationally expensive* and a *black box*, so that we cannot inspect or modify it. In this setting, standard approaches to MCMC are likely to be limited by the computational expense of evaluating the forward model at every step of the chain. Our approach addresses this cost by storing the results of each model evaluation in a set $\mathcal{S}_t := \{(\theta_i, \mathbf{f}(\theta_i))\}_{i=1}^{n_t}$ and reusing them. The stochastic process $\{\theta_t\}_{t\geq 0}$ proposed in [9] evolves by drawing new points from some proposal kernel q and accepting or rejecting the proposed move according to an approximation of the forward model, $\tilde{\mathbf{f}}_t$, constructed from the set $\mathcal{S}_t$. During the simulation of this process, the algorithm carefully chooses new points at which to run the forward model, enlarging $\mathcal{S}_t$ and thus improving $\tilde{\mathbf{f}}_t$; we refer to enlargement of $\mathcal{S}_t$ as "refinement." Intuitively, it would seem that if $\tilde{\mathbf{f}}_t$ converges to $\mathbf{f}$ in an appropriate sense, then the sequence $\{\theta_t\}_{t\geq 0}$ might asymptotically behave like the usual Metropolis–Hastings chain with proposal q and forward model $\mathbf{f}$. Indeed, the algorithm we constructed in [9] has these properties.

We obtain a converging sequence of approximations $\tilde{\mathbf{f}}_t$ by constructing the approximation locally—that is, constructing $\tilde{\mathbf{f}}_t(\theta)$ using only the elements of $\mathcal{S}_t$ whose input values θ_i lie within a distance R of θ. The radius R is selected so that this subset contains a fixed number of points N. The value of N depends on the functional form of the approximation; for instance, if $\tilde{\mathbf{f}}_t$ is a local quadratic approximation, we need at least $(d+1)(d+2)/2 =: N_{\text{def}}$ points to fully

determine its coefficients.[1] Local approximations are relatively straightforward to analyze in that they typically converge whenever the sample set $\mathcal{S}_t$ becomes denser, thus allowing $R \to 0$. (Regularity conditions on $\mathbf{f}$ sufficient for convergence in the case of local polynomial approximations, for example, are given in [8].) These general conditions for convergence allow us to promote efficiency by aggressively tailoring $\mathcal{S}_t$ during sampling, while still maintaining asymptotic exactness of the overall MCMC. The resulting algorithm is straightforward to use, since its adaptivity allows users to treat it much like standard adaptive MCMC algorithms: the behavior of the chain can be monitored for convergence, which in our case reflects both the exploration of the posterior and the convergence of the approximation. Our work thus differs from previous efforts using global approximations to accelerate inference [36, 2, 25, 46], where the entire set $\mathcal{S}_t = \mathcal{S}_0$ is constructed as a preprocessing step and is used to build a single high-order approximation. In these methods it is difficult to choose how many samples $\mathcal{S}_0$ should contain or how to monitor the accuracy of the overall sampling.

An illustration of the algorithm is given in Figure 1. At early times, the samples are sparse, leading to local models constructed over large regions, depicted by large balls, rendering them relatively inaccurate. As MCMC progresses, refinements increase the density of the sample set in regions of high posterior probability, shrinking the local neighborhoods and increasing the quality of approximations. Model runs do not lie on any structured grid and are generally contained within regions of the parameter space that are relevant to the inference problem, thus enhancing efficiency whenever the posterior is concentrated.



(a) Early times. (b) Late times.

Figure 1. *Schematic of the behavior of local approximation MCMC. The balls are centered at locations where local approximations might be evaluated, and their radii are chosen to contain the N nearest points, used to build the approximation. The accuracy of a local approximation generally increases as this ball size shrinks. At early times, the sample set $\mathcal{S}_t$ is sparse, and thus local approximations are built over relatively large balls, such that their accuracy is limited. At later times, refinements enrich the sample set in regions of high posterior probability, allowing the balls to shrink and the approximations to become more accurate.*

[1] In practice, we often select $N = \sqrt{d}N_{\text{def}}$ to improve the conditioning of the associated least squares system. More details are given in [9].

We now review a sketch of our approximate MCMC algorithm, given in Algorithm 1. Please see Appendix A and Algorithm 3 for a more complete description of the algorithm; additional details can be found in our previous work [9]. The stochastic process $\{\theta_t\}_{t\geq 0}$ is produced by the method RunChain, which applies the transition kernel K_t repeatedly. The transition kernel is provided with the current state of the chain θ_t; the current set of samples $\mathcal{S}_t$; the inference problem, as defined by ℓ, $\mathbf{d}$, p, and $\mathbf{f}$; and a symmetric translation-invariant proposal distribution q. The kernel uses the current point of the chain, θ^-, to draw a proposal, θ^+. It forms local approximations near these points, $\tilde{\mathbf{f}}^+$ and $\tilde{\mathbf{f}}^-$, respectively, based on nearby samples contained in $\mathcal{S}_t$. Next, it computes the acceptance probability α in the usual way, substituting the approximations for the true forward model. Then, the algorithm optionally refines the sample set by choosing a new point θ^* and running the forward model at that location.

Algorithm 1. Sketch of approximate Metropolis–Hastings algorithm.

1: **procedure** RunChain($\theta_1, \mathcal{S}_1, \ell, \mathbf{d}, p, \mathbf{f}, q, T$)
2: **for** $t = 1, \ldots, T$ **do**
3: $(\theta_{t+1}, \mathcal{S}_{t+1}) \leftarrow K_t(\theta_t, \mathcal{S}_t, \ell, \mathbf{d}, p, \mathbf{f}, q)$
4: **end for**
5: **end procedure**

6: **procedure** $K_t(\theta^-, \mathcal{S}, \ell, \mathbf{d}, p, \mathbf{f}, q)$
7: Draw proposal $\theta^+ \sim q(\theta^-, \cdot)$.
8: Compute approximate models $\tilde{\mathbf{f}}^+$ and $\tilde{\mathbf{f}}^-$, valid near θ^+ and θ^-, respectively.
9: Compute acceptance probability $\alpha \leftarrow \min\left(1, \frac{\ell(\theta|\mathbf{d},\tilde{\mathbf{f}}^+)p(\theta^+)}{\ell(\theta|\mathbf{d},\tilde{\mathbf{f}}^-)p(\theta^-)}\right)$.
10: **if** approximation needs refinement near θ^- or θ^+ **then**
11: Select new point θ^* and grow $\mathcal{S} \leftarrow \mathcal{S} \cup (\theta^*, \mathbf{f}(\theta^*))$. Repeat from line 8.
12: **else**
13: Draw $u \sim \text{Uniform}(0, 1)$. If $u < \alpha$, **return** $(\theta^+, \mathcal{S})$; else **return** $(\theta^-, \mathcal{S})$.
14: **end if**
15: **end procedure**

Choosing when and where to refine $\mathcal{S}_t$ is critical to the performance of the overall algorithm. We combine two criteria to decide when to refine the approximation. First, the approximation is refined near θ^- or θ^+ with equal probabilities β_t, such that the expected number of refinements diverges as $t \to \infty$. This criterion is sufficient for convergence of the algorithm, as detailed in [9]. The sequence (β_t) may be difficult to tune in practice, however. Thus we complement the random refinement criterion with a cross-validation strategy that triggers refinement whenever the estimated error in the acceptance probability α (due to the approximation of $\mathbf{f}$) appears too large. This latter threshold for refinement is tightened with increasing t, pushing the approximation to improve as the chain lengthens. Although the cross-validation criterion is not sufficient for convergence of the algorithm, it appears efficient in practice, and we use it in conjunction with the random refinement strategy. When refinement is needed near either θ^- or θ^+, we do not simply run the model at that point, since doing

so would introduce clusters into $\mathcal{S}_t$, degrading the quality of local approximations. Instead, we use a local space-filling design strategy to choose a distinct but nearby point θ^* at which to run the model.

3. Sharing local approximations for parallel MCMC. The naive approach to parallelizing MCMC is simply to run several independent chains in parallel. Although running parallel chains facilitates useful convergence diagnostics [12, 3], practical scaling in highly parallel environments is limited because of the serial nature of MCMC and the replication of transient behavior across multiple chains [44].

More sophisticated strategies for parallel MCMC exchange information between the chains, for example, by proposing moves to states discovered by other chains [13]. Population MCMC algorithms explore a family of tempered distributions with parallel chains, so that swapping states between the chains can provide long-range moves [5]. These techniques attempt to improve the mixing time of the Markov chain and when successful, may provide superior performance to the naive parallelization [22]. Other constructions, e.g., [4], propose multiple points in parallel and try to make use of all these points in determining subsequent steps of a single chain.

Any of these parallel approaches requires repeated evaluations of the forward model, however, which can dominate the overall cost of the algorithm. If multiple copies of Algorithm 1 are run in parallel, a natural idea is to allow them to collaborate by sharing a common set of evaluations $\mathcal{S}_t$. That is, whenever one chains performs refinement, the result is shared asynchronously with all the chains; hence each chain receives additional model evaluations "for free." Since the limiting computational cost in our context lies in constructing $\mathcal{S}_t$, parallelizing this process should directly impact the real-world performance of the sampler during the stationary and even the transient phases of the chains. With regard to the latter point, we note that parallelizing $\mathcal{S}_t$ can reduce the number of model evaluations that are triggered by each individual chain during its initial transient phase.

Although it should be straightforward to combine the parallel construction of $\mathcal{S}_t$ with the other parallelization strategies described above, we leave that to future work.

3.1. Convergence of the parallel algorithm. Recall that our local approximation MCMC algorithm is detailed in Appendix A; see, in particular, Algorithm 3. Below we will show that the sufficient conditions for convergence of a single-chain version of Algorithm 3, as described in [9] and reproduced below in Definition 3.1, are also sufficient conditions for the convergence of the parallel version. The arguments given in [9] are straightforward to extend because we have chosen conditions where enlarging the sample set $\mathcal{S}_t$ is always helpful; thus the additional refinements contributed by parallel chains cannot hinder convergence. Rather than repeating the entire discussion of convergence from that paper, here we merely extend the simplest and weakest convergence result—for a single chain on a compact state space [9, Theorem 3.4]—to the case of parallel chains. We refer the reader to [9, Theorem 3.3] for related conditions and a treatment of noncompact state spaces that can similarly be extended to the parallel case.

We require some notation before stating the result. Let $\mathcal{L}(X)$ denote the distribution of a random variable X. For fixed $\epsilon > 0$, we say that $\mathcal{S} \subset \Theta$ is an ϵ-cover of Θ if $\sup_{\theta\in\Theta} \min_{s\in\mathcal{S}} \|\theta - s\|_2 < \epsilon$. We note that if lines 13–21 and line 23 are removed from Algorithm 3, and all references to $\mathcal{S}_t$ are replaced by a reference to a single set $\mathcal{S}$, then the sequence $\{\theta_t\}_{t\geq 0}$

constructed by running the modified algorithm is a Markov chain. We use the $\mathcal{S}$ subscript to denote all approximate objects associated with this Markov chain (e.g., $K_{\mathcal{S}}$ is the associated transition kernel, $r_{\mathcal{S}}$ is the proposal function from line 9 of Algorithm 3 and $q_{\mathcal{S}}$ is the associated proposal density, $p_{\mathcal{S}} := \ell(\theta|\mathbf{d}, \tilde{\mathbf{f}})p(\theta)$ is the approximation to $\ell(\theta|\mathbf{d}, \mathbf{f})p(\theta)$ used in line 11 of Algorithm 3, and $\alpha_{\mathcal{S}}$ is the associated acceptance probability). Similarly, K_∞, r_∞, q_∞, p_∞, and α_∞ are the values of these objects for the Markov chain with the same proposal kernel as in Algorithm 3 and with the *correct* posterior distribution as its target distribution. Finally, define $\pi(\theta) := p(\theta)\ell(\theta|\mathbf{d}, \mathbf{f})/Z$, where Z is a normalization constant. Our simple result makes the following assumptions.

Definition 3.1 (sufficient conditions for convergence).

1. *The state space* Θ *is compact.*
2. *The proposal* $q(\theta, \cdot \,|\, \mathbf{f}\,) = q(\theta, \cdot)$ *does not depend on* $\mathbf{f}$*, and both the proposal distribution* $q(\theta, \cdot)$ *and target distribution* $p(\cdot|\,\mathbf{d}\,)$ *have* C^∞ *densities that are bounded away from zero uniformly in* θ*.*
3. *The sequence of parameters* $\{\beta_t\}_{t\in\mathbb{N}}$ *used in Algorithm 3 are of the form* $\beta_t \equiv \beta > 0$*; any sequence* $\{\gamma_t\}_{t\geq 0}$ *is allowed.*
4. *The approximation of* $\log p(\theta|\mathbf{d})$ *is made via quadratic interpolation on the* $N = (d+1)(d+2)/2$ *nearest points.*
5. *The subalgorithm* RefineNear *is replaced with*

$$\textsc{RefineNear}(\theta, \mathcal{S}) = \textbf{return}(\mathcal{S} \cup \{(\theta, f(\theta))\}).$$

6. *We fix a constant* $0 < \lambda < \infty$*. In line* 15 *of Algorithm 3, immediately before the word* **then***, we add "****or****, for* $\mathcal{B}(\theta^+, R)$ *as defined in the subalgorithm* LocApprox$(\theta^+, \mathcal{S}, \emptyset)$ *used in line* 10*, the collection of points* $\mathcal{B}(\theta^+, R) \cap \mathcal{S}$ *is not* λ*-poised." We add the same check, with* θ^- *replacing* θ^+ *and "line* 8*" replacing "line* 10*," in line* 17 *of Algorithm 3. The concept of poisedness is defined in* [8]*.*

The following result extends Theorem 3.4 of [9] to parallel chains.

Theorem 1 (convergence with parallel chains). *Let* $\{X_t^{(i)}\}_{t\geq 0, 1\leq i\leq n}$ *be the* n *stochastic processes obtained in a parallel run of Algorithm 3 with* n *chains, and assume that the algorithm parameters satisfy Definition 3.1. Then, for all* $1 \leq i \leq n$*,*

$$\lim_{t\to\infty} \|\mathcal{L}(X_t^{(i)}) - \pi\|_{TV} = 0.$$

Proof. The proof of Lemma B.3 of [9] holds exactly as stated, with the proof as given. The remainder of the proof of Theorem 3.4 from [9] holds for $\{X_t^{(i)}\}_{t\geq 0}$, for each fixed $1 \leq i \leq n$, with the following modifications:

(i) the chain $\{X_t\}_{t\geq 0}$ should be replaced by $\{X_t^{(i)}\}_{t\geq 0}$ wherever it appears, and

(ii) the auxiliary process associated with $\{X_t^{(i)}\}_{t\geq 0}$ is $\{(\mathcal{S}_t, X_t^{(1)}, \ldots, X_t^{(i-1)}, X_t^{(i+1)}, \ldots, X_t^{(n)})\}_{t\geq 0}$, rather than $\{\mathcal{S}_t\}_{t\geq 0}$. ∎

We emphasize that this proof of the convergence of $\{X_t^{(i)}\}_{t\geq 0}$ is completely indifferent to the points that are added to $\mathcal{S}_t$ by the other chains $\{X_t^{(j)}\}_{t\geq 0}$, $j \neq i$.

Remark 3.2 (Do parallel chains always work?). Although our sufficient conditions for convergence carry over to the parallel case, it is natural to ask whether there are any problems that are not covered by our current theory—i.e., where, having departed from the sufficient conditions of Definition 3.1, the single-chain algorithm still converges, but the parallel algorithm does not. We conjecture that the answer is no, but are unable to prove it.

To explain the difficulty in proving this conjecture, note that all the proofs of sufficient conditions for convergence given in [9] apply as stated to the parallel version of Algorithm 3 because they proceed by proving the following critical steps:

1. Due to minorization conditions (e.g., the second condition of Definition 3.1), for any $\epsilon > 0$ the set $\mathcal{S}_t$ will be an ϵ-cover of Θ for all t sufficiently large.
2. The distance $\| K_\infty(\theta_t, \cdot) - K_{\mathcal{S}_t}(\theta_t, \cdot) \|_{\mathrm{TV}}$ between a single "step" of Algorithm 3 and the step that would be made by the true transition kernel K_∞ can be made arbitrarily small by making $\mathcal{S}_t$ an ϵ-cover of Θ for ϵ sufficiently small.

In particular, under Definition 3.1, adding points to $\mathcal{S}_t$ cannot hurt the convergence of $\{X_t^{(i)}\}_{t \geq 0}$ very much, because adding points to an ϵ-cover always results in a set that is still an ϵ-cover. For a sufficiently broader class of Metropolis–Hastings chains, however, it is not true that $K_{\mathcal{S}}$ is close to K_∞ whenever $\mathcal{S}$ is an ϵ-cover, and in particular it is possible to add points to $\mathcal{S}$ while simultaneously making an approximation worse. This possibility of maladaption is what makes adaptive algorithms difficult to study and prevents us from making the stronger claim that the parallel algorithm is convergent under *every* possible condition where the single-chain algorithm is.

4. Local approximations and approximating the proposal.

We now show how the transition kernel of our approximate MCMC scheme can use the current approximation not only to evaluate the acceptance probability, but also to construct a proposal distribution. This development enables a much wider range of Metropolis–Hastings proposals to be used with expensive models, and in particular allows gradient- and Hessian-driven proposals to be used in a setting where derivatives of $\mathbf{f}$ cannot be directly evaluated. We proceed by recalling the Metropolis-adjusted Langevin algorithm (MALA) algorithm and explaining how to adapt local approximations to this proposal scheme. Next, we prove a general result that our modified algorithm is still convergent as long as the good properties of the approximation are transferred into good approximation of the overall kernel. We conclude by showing that the result applies in the representative case of manifold MALA.

4.1. Simplified manifold Metropolis-adjusted Langevin algorithm (mMALA).

The simplified mMALA [20] is a recent method for constructing proposals adapted to the local geometry of the target distribution. This method is also closely related to the preconditioning performed in the stochastic Newton method [34]. The mMALA proposal is derived by explicitly discretizing a Langevin diffusion with stationary distribution $p(\theta|\mathbf{d})$, leading to

$$q(\theta, \theta'|\mathbf{f}) = \mathcal{N}\left(\theta'; \theta + \frac{\epsilon}{2} M(\theta) \nabla_\theta \log\left(\ell(\theta|\mathbf{d}, \mathbf{f}) p(\theta)\right), \epsilon M(\theta)\right), \tag{4.1}$$

for integration step size ϵ and position-dependent symmetric positive definite (SPD) mass matrix $M(\theta)$, which we may treat as a preconditioner. "Preconditioning" in this context amounts to rescaling the parameter space, e.g., to make the distribution (locally) more isotropic. We

use the notation $q(\theta, \theta'|\mathbf{f})$ to emphasize the dependence of the proposal on the forward model. The corresponding acceptance ratio is

$$\alpha(\theta, \theta') = \min\left(1, \frac{\ell(\theta'|\mathbf{d}, \mathbf{f})p(\theta')q(\theta', \theta|\mathbf{f})}{\ell(\theta|\mathbf{d}, \mathbf{f})p(\theta)q(\theta, \theta'|\mathbf{f})}\right).$$

We are relatively unconstrained in our choice of preconditioner, as long as it is SPD. Standard MALA corresponds to choosing the identity matrix, $M(\theta) = I$. Simplified manifold MALA (mMALA) [20], on the other hand, chooses the mass matrix to reflect a Riemannian metric induced by the posterior distribution:

$$M(\theta) = \left[-\mathbb{E}_{\mathbf{d}|\theta}\left(\nabla_\theta^2 \log \ell(\theta|\mathbf{d}, \mathbf{f})\right) - \nabla_\theta^2 \log p(\theta)\right]^{-1}.$$

The inverse of this matrix is the expected Fisher information plus the negative Hessian of the log-prior density. In general, computing the expected Fisher information is not trivial, but it is relatively simple for Gaussian likelihoods, e.g.,

$$\ell(\theta|\mathbf{d}, \mathbf{f}) = \mathcal{N}(\mathbf{d}; \mathbf{f}(\theta), \Sigma_\ell),$$

with some prescribed covariance matrix $\Sigma_\ell \in \mathbb{R}^{n\times n}$. If we also have a Gaussian prior, $p(\theta) = \mathcal{N}(\theta; \mu, \Sigma_p)$, with covariance $\Sigma_p \in \mathbb{R}^{d\times d}$ and mean vector $\mu \in \mathbb{R}^d$, then

$$M^{-1}(\theta) = J(\theta)^\top \Sigma_\ell^{-1} J(\theta) + \Sigma_p^{-1},$$

where $J(\theta) := \nabla_\theta \mathbf{f}(\theta) \in \mathbb{R}^{n\times d}$. Girolami and Calderhead [20] observe that choosing the preconditioner in this manner can dramatically improve the performance of MALA. Yet even standard MALA can be difficult to apply in practice because the necessary derivatives must be computable and inexpensive; the manifold variant uses Jacobians of the forward model, which are typically even more challenging to obtain. Adapting mMALA and similar proposals to use local approximation is therefore particularly interesting, as approximations can cheaply provide these derivatives.

4.2. Modifying the algorithm. The key challenge in extending Algorithm 1 to mMALA (and similar proposals) is to allow simultaneous use of the approximation within the proposal and the acceptance probability. Algorithm 2 shows the three required changes. Two modifications are trivial: we restore the proposal distribution to its usual place in the acceptance probability, to account for the nonsymmetric proposal, and we provide the proposal with the approximate forward model $\tilde{\mathbf{f}}$.

The third step is more subtle, introducing a coupling construction to allow model refinement to proceed safely. Note that in Algorithm 1, refinement only recomputes the acceptance probability; the proposed point is held fixed. Hence, exactly one proposal is made per step, even though an inaccurate approximation might cause the algorithm to seek further information before deciding whether that proposal can be accepted. Allowing a new proposal to be generated upon refinement would bias the chain away from regions with inaccurate approximations (equivalently, towards regions where the approximation appears accurate), which is clearly undesirable.

This difficulty can be resolved by coupling the approximate kernel K_t to the kernel associated with the true model, K_∞. We accomplish this coupling by fixing the realization of the random variable used to generate the proposal, but allowing the proposal to be recomputed if the model is refined. (See [40, 18] for other algorithms that reuse randomness to avoid bias, and [48] for a typical use of this idea in a theoretical paper.) Specifically, construct a deterministic function $r(\theta, \mathbf{z}, \mathbf{f})$ such that drawing a random vector $\mathbf{z} \sim \mathcal{N}(0, I)$[2] and then computing $\theta' = r(\theta, \mathbf{z}, \mathbf{f})$ is equivalent to drawing $\theta' \sim q(\theta, \cdot|\mathbf{f})$. The modified algorithm holds $\mathbf{z}$ fixed under refinement, recomputing θ^+ as needed. In the case of standard Metropolis–Hastings proposals, this coupling strategy reduces to our original approach. This coupling construction ensures that the magnitude of any perturbation to the proposed point θ^+ induced by refinement vanishes as $t \to \infty$.

In the case of simplified mMALA, the proposal will be a Gaussian distribution, $q(\theta, \theta'|\mathbf{f}) = \mathcal{N}(\mu_q(\theta, \mathbf{f}), \Sigma_q(\theta, \mathbf{f}))$, for some position- and model-dependent mean μ_q and covariance Σ_q, and hence $r = \mu_q(\theta, \mathbf{f}) + \Sigma_q^{1/2}(\theta, \mathbf{f})\mathbf{z}$. The rest of the algorithm is updated naturally, including the inclusion of the proposal into the cross-validation criterion. The resulting approach is summarized in Algorithm 2. For brevity, we defer precise pseudocode to Algorithm 3 in Appendix A.

Algorithm 2. Sketch of approximate Metropolis–Hastings algorithm with general proposals.

1: **procedure** $K_t(\theta^-, \mathcal{S}, \ell, \mathbf{d}, p, \mathbf{f}, r, q)$
2: Draw $\mathbf{z}_t \sim \mathcal{N}(0, I)$.
3: Construct $\tilde{\mathbf{f}}^-$.
4: Compute $\theta^+ = r(\theta^-, \mathbf{z}_t, \tilde{\mathbf{f}}^-)$.
5: Construct $\tilde{\mathbf{f}}^+$.
6: Compute acceptance probability $\alpha \leftarrow \min\left(1, \frac{\ell(\theta^+|\mathbf{d}, \tilde{\mathbf{f}}^+)p(\theta^+)q(\theta^+, \theta^-|\tilde{\mathbf{f}}^+)}{\ell(\theta^-|\mathbf{d}, \tilde{\mathbf{f}}^-)p(\theta^-)q(\theta^-, \theta^+|\tilde{\mathbf{f}}^-)}\right)$.
7: **if** approximation needs refinement near θ^- or θ^+ **then**
8: Select new point θ^* and grow $\mathcal{S} \leftarrow \mathcal{S} \cup (\theta^*, \mathbf{f}(\theta^*))$. Repeat from line 3.
9: **else**
10: Draw $u \sim \text{Uniform}(0, 1)$. If $u < \alpha$, **return** $(\theta^+, \mathcal{S})$, else **return** $(\theta^-, \mathcal{S})$.
11: **end if**
12: **end procedure**

4.3. Convergence analysis. We now provide a convergence result for Algorithm 3. Some technical definitions and the proofs from this section may be found in Appendix B.

The general idea is to show that as the sample set $\mathcal{S}_t$ becomes dense, the approximate kernel $K_\mathcal{S}$ converges to the kernel using the true model, K_∞, and that MCMC converges as a result. We begin by stating our assumptions precisely. Below W_2 denotes the 2-Wasserstein metric, defined in Appendix B.

[2]We choose a vector of independent standard Gaussians for convenience and without loss of generality, but in practice other distributions for $\mathbf{z}$ may be more convenient.

Definition 4.1 (convergence assumptions). *Assume that the following hold:*

1. *For any $\delta > 0$, there is an $\epsilon = \epsilon(\delta) > 0$ so that any ϵ-cover $\mathcal{S}$ satisfies*

$$\sup_{\theta \in \Theta} W_2(K_{\mathcal{S}}(\theta, \cdot), K_\infty(\theta, \cdot)) < \delta. \tag{4.2}$$

2. *There exist constants $0 < \eta_0 < \infty$ and $1 < C < \infty$ such that for any $0 < \eta < \eta_0$,*

$$\sup_{\theta, \theta' \in \Theta, \, \|\theta - \theta'\| < \eta} W_2(K_\infty(\theta, \cdot), K_\infty(\theta', \cdot)) < C\eta. \tag{4.3}$$

3. *For any $\varphi_0 < \infty$ and $\delta > 0$, there exists $\epsilon > 0$ so that any ϵ-covers $\mathcal{S}$, $\mathcal{S}'$ satisfy*

$$\sup_{\|z\| \le \varphi_0} \sup_{\theta \in \Theta} \|r_{\mathcal{S}}(\theta, z) - r_{\mathcal{S}'}(\theta, z)\| \le \delta. \tag{4.4}$$

4. *The assumptions in Definition 3.1 hold.*

The following theorem states that these assumptions, which we will have to check, are sufficient for convergence of the approximate Markov chain.

Theorem 2 (convergence of Algorithm 3 on compact state space). *Let the assumptions in Definition 4.1 hold, and let $\{X_t\}_{t \ge 0}$ be the sequence drawn from a run of Algorithm 3. Then*

$$\lim_{t \to \infty} W_2(\mathcal{L}(X_t), \pi) = 0. \tag{4.5}$$

Remark 4.2. The proof proceeds by coupling each step of the output of Algorithm 3. Our coupling construction gives us the important estimate (B.3), which would not hold if the randomness at each step were resampled upon model refinement. In most cases, including our application to mMALA, this proof can be extended to give convergence in total variation distance by using a "one-shot" coupling (see [43]).

Finally, we observe that mMALA often satisfies Definition 4.1. Although our convergence results apply only to some uses of mMALA, we believe they are representative of the more general case and suggest the feasibility of analytically transferring the good properties of the approximation onto the kernel.

Theorem 3 (convergence of approximate mMALA). *We consider running Algorithm 3 with proposal kernel q (equivalently, r) given by the mMALA algorithm. Assume that the following hold:*

- *The state space Ω is the d-dimensional hypercube $[0, 1]^d$ for some $d \in \mathbb{N}$.*
- *The mass matrix $M(\theta)$ and likelihood $\ell(\theta | \mathbf{d}, \mathbf{f})$ are both C^∞ functions on Ω. Furthermore, the smallest singular value of $M(\theta)$ is uniformly bounded away from zero by some $c > 0$.*
- *The posterior density $p(\cdot | \mathbf{d})$ is C^∞ and bounded away from zero uniformly on Ω.*
- *Assumptions 3–6 of Definition 3.1 hold.*

Then the output $\{X_t\}_{t \ge 0}$ of Algorithm 3 satisfies

$$\lim_{t \to \infty} \|\mathcal{L}(X_t) - \pi\|_{TV} = 0.$$

The proof of Theorem 3, given in Appendix B, merely checks the assumptions in Definition 4.1. Essentially, these assumptions hold because mMALA uses approximations of the derivatives of $\mathbf{f}$ to construct a Gaussian proposal; the derivative approximations improve as $\mathcal{S}$ grows and the Gaussian proposal is not too sensitive to errors in these approximations, and hence the entire kernel converges in the necessary sense.

5. Numerical experiments.

We present three numerical examples to explore the algorithmic ideas developed in the preceding sections. First, we use a simple example to demonstrate how the improved mixing properties of MALA can successfully be paired with our local approximation scheme. Then, we turn to two more computationally intensive inference problems, with forward models drawn from realistic applications. The first of these, a groundwater tracer transport problem, is the focus of our parallel MCMC explorations. Though posterior evaluations are quite expensive in this problem, we can still compare results with standard MCMC chains that employ no approximation, and thus verify the accuracy of posterior expectations. The second application example is even more expensive—such that MCMC is essentially intractable without the use of approximations. Here, our goal is simply to show that with a *particular instantiation* of parallel local approximation MCMC, fully Bayesian inference that previously would not have been feasible (given reasonable computational resources) is now feasible.

5.1. Quartic example.

Consider a target distribution with the following log-quartic density:

$$\log \pi(x_1, x_2) = -x_1^4 - \frac{(2x_2 - x_1^2)^2}{2}, \tag{5.1}$$

also illustrated in Figure 2. We simulate from this target distribution in four ways: using (i) adaptive Metropolis (AM) [23] and (ii) mMALA, each paired with either (a) evaluations of the exact target density or (b) our local approximation (LA) scheme. In other words, the combinations (a+i) and (a+ii) are standard MCMC algorithms with two different proposal schemes, and the combinations (b+i) and (b+ii) pair local approximation MCMC with the same proposal schemes. We call these simulation approaches "exact+AM," "exact+mMALA," "LA+AM," and "LA+mMALA," respectively.

For each of the simulation approaches defined above, we run 20 independent chains, each of length 6×10^5 steps. (No parallelism is employed in this example.) We then evaluate an *expected squared relative error* $\bar{\varepsilon}^2$, as a function of the number of target density evaluations, for each approach. The quantity $\bar{\varepsilon}^2$ is defined as follows. Before computing expectations with respect to the target density, we discard the first 10^4 samples of each chain as burn-in. Then we obtain a reference estimate C_0 for the target covariance matrix by pooling post-burn-in samples from all 40 chains that employ exact target density evaluations. Next, for each independent chain (indexed by i) associated with a given simulation approach, we compute a running t-sample estimate of the target covariance $\hat{C}_t^{(i)}$ and define a relative squared error as

$$\varepsilon_t^{2,(i)} := \frac{\|\hat{C}_t^{(i)} - C_0\|_F^2}{\|C_0\|_F^2}, \tag{5.2}$$

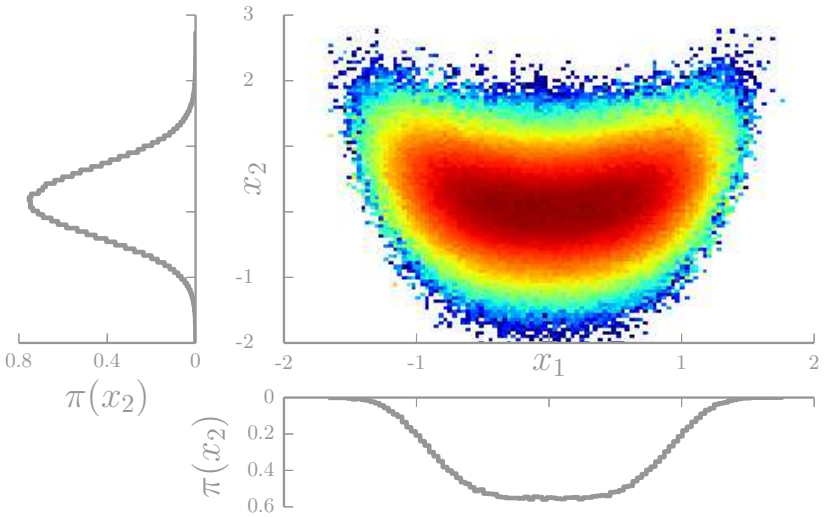


Figure 2. *Joint and marginal densities for the quartic target (5.1). We characterize this density with both exact density evaluations and local approximations, paired with adaptive Metropolis and mMALA.*

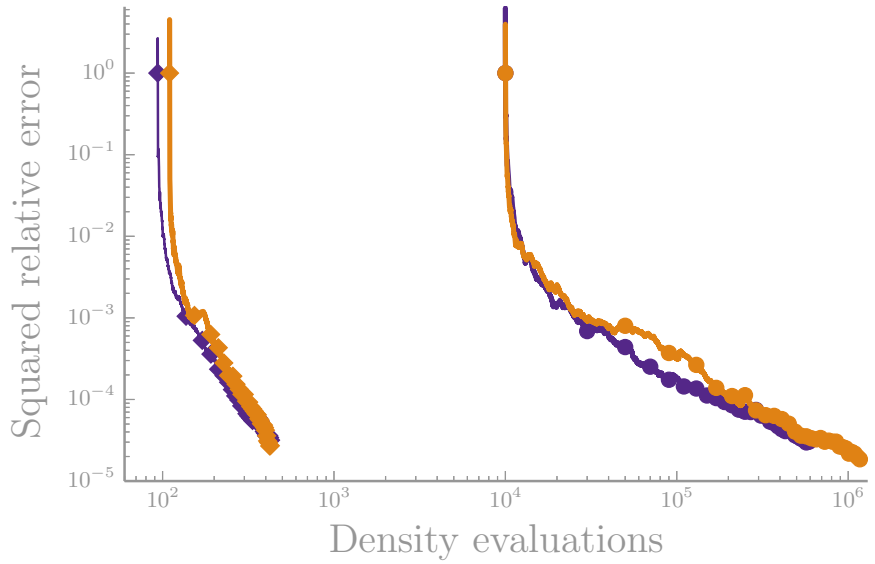


Figure 3. *Quartic example of section 5.1: Expected squared relative error $\bar{\varepsilon}_t^2$ as a function of the number of target density evaluations. Purple lines correspond to AM chains, while gold lines correspond to mMALA. The circles mark chains that employ exact evaluations of the target density, while diamonds mark chains using local approximation. In the exact case, mMALA requires evaluations of the target density and its gradient. We assume gradient evaluations are comparable in cost to density evaluations and, therefore, count them as density evaluations. Errors are obtained by averaging over 20 independent chains from each simulation approach, each of length 6×10^5 steps.*

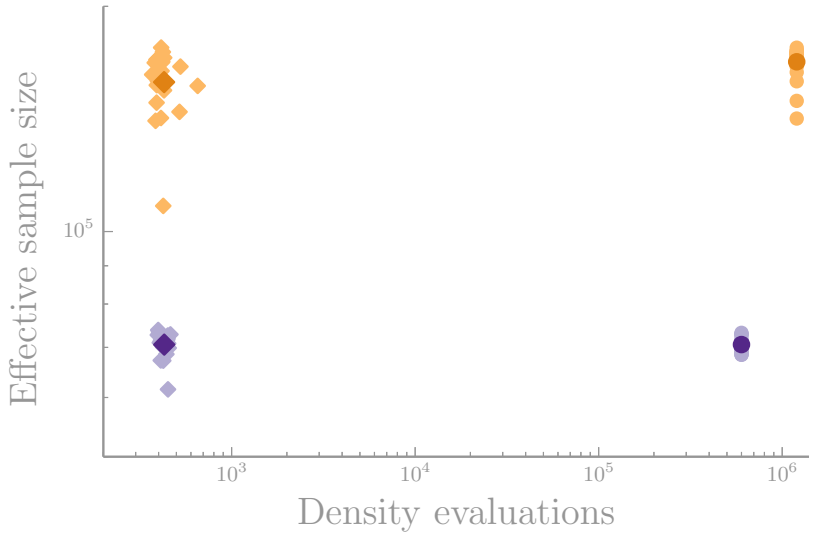


Figure 4. *Quartic example of section 5.1: Effective sample size for independent MCMC chains, each of length 6×10^5. As in Figure 3, purple symbols correspond to AM chains, and gold symbols correspond to mMALA chains. Circles indicate chains using exact target density evaluations, while diamonds indicate the use of local approximations. The darker dot in each cluster is its expected value. In the mMALA case with exact evaluations, we count target gradient evaluations as density evaluations.*

where $\|\cdot\|_F$ denotes the Frobenius norm. Then we average over the 20 independent chains to obtain $\bar{\varepsilon}_t^2 = \frac{1}{20}\sum_{i=1}^{20} \varepsilon_t^{2,(i)}$. Figure 3 plots $\bar{\varepsilon}_t^2$ versus the number of target *density evaluations*, for each simulation approach. For the exact+mMALA chains, which require direct evaluation of the gradients of $\log \pi$, we count each gradient evaluation as an additional density evaluation. For large-scale models, gradient evaluations (e.g., via an adjoint solve) might be more expensive than density evaluations, so this accounting is a conservative estimate of computational cost.

Several trends are apparent in this figure. First, comparing the exact and local approximation chains, we see that the same level of accuracy is achieved with significantly fewer density evaluations when using approximations. When target density evaluations are expensive, this translates to computational savings. We also note that the exact chains show a squared error decaying at roughly the standard Monte Carlo rate of $1/n$, where n is the number of density evaluations. But the error decays more quickly when using local approximation MCMC. This is because MCMC steps that do not require refinement of $\mathcal{S}_t$ can still reduce estimator variance—and thus the overall error—without using a target density evaluation. Since we expect the refinement frequency to decay as the chain progresses, we also expect the error decay rate, in terms of the number of target density evaluations, to accelerate.

Another useful measure of sample quality is the effective sample size (ESS) of each chain, which we compute from each chain's integrated autocorrelation time [50]. ESS is a measure of how many "effectively independent" samples have been generated from the target distribution. In Figure 4, we plot the ESS for each independently realized chain, using each of the four simulation approaches. In general, the mMALA chains have larger ESS than the AM chains,

reflecting their improved mixing for this target distribution. Also, the local approximation chains achieve nearly the same ESS as their exact counterparts, but with nearly three orders of magnitude fewer density evaluations. ESS of course varies from realization to realization; the dark symbols in the middle of the scatter plots illustrate the *average* ESS and cost of each set of 20 chains. In general, we do not expect that introducing an approximation will improve mixing, and in this example ESS with exact evaluations (exact+AM or exact+mMALA) provides an upper bound on sampling performance. Indeed, Figure 4 shows that the ESS is very slightly lower using local approximations; this is apparent in the MALA cases. Nonetheless, the local approximation chains achieve nearly the same ESS as their exact counterparts, but with nearly three orders of magnitude fewer density evaluations. Moreover, the improved mixing of mMALA in the exact case is preserved when using local approximations.

5.2. Tracer transport problem. Predicting the evolution of groundwater contaminant concentrations over time is vital to many monitoring and remediation efforts [37]. A contaminant is typically modeled as a nonreactive tracer that diffuses and is advected by groundwater flow. Here we construct an inverse problem that simulates a monitoring configuration: the tracer concentration is observed at a small number of wells over a short period of time, and the subsurface conductivity field must be inferred given these data.

The conductivity field is assumed to be piecewise constant in six irregularly shaped areas, reflecting different subsurface features (e.g., sand, clay, gravel) each with constant but unknown conductivities. We consider a problem domain with two horizontal coordinates $x, y \in [0,1]^2$. The true log-conductivity is depicted in Figure 5. The conductivity is parameterized as

$$\kappa(x,y) = \exp \theta_{\underline{j}(x,y)},$$

where $\underline{j}(x,y) \in \{1,2,\ldots,6\}$ is the smallest integer j such that $x_0^j \le x \le x_1^j$ and $y_0^j \le y \le y_1^j$, where the bounds $(x_0^j, x_1^j, y_0^j, y_1^j)$ are given in Table 1. The parameters θ_i are endowed with uniform priors; the upper and lower bounds for each prior are also given in Table 1.

Modeling tracer evolution requires first computing the hydraulic head, which determines the groundwater velocity. Under the Dupuit approximation [17], the hydraulic head h obeys the elliptic equation

$$\nabla \cdot (\kappa h \nabla h) = -f_h, \tag{5.3}$$

where $\kappa(x,y)$ is the conductivity field and $f_h(x,y)$ is the hydraulic head forcing. In the setup of our problem, the forcing is created by pumping at four well locations, $(a_i, b_i) \in \{(0.15, 0.15), (0.85, 0.15), (0.85, 0.85), (0.15, 0.85)\}$, such that

$$f_h(x,y) = \sum_{i=1}^{4} p_i \exp\left(\frac{(a_i - x)^2 + (b_i - y)^2}{0.02}\right),$$

where $p_i \in (10, 50, 150, 50)$. The model (5.3) assumes homogeneous Dirichlet boundary conditions at $y = 0$ and $y = 1$ and homogeneous Neumann conditions at $x = 0$ and $x = 1$. The Darcy velocity is determined by the hydraulic head gradient

$$\begin{bmatrix} u \\ v \end{bmatrix} = -h\kappa\nabla h. \tag{5.4}$$

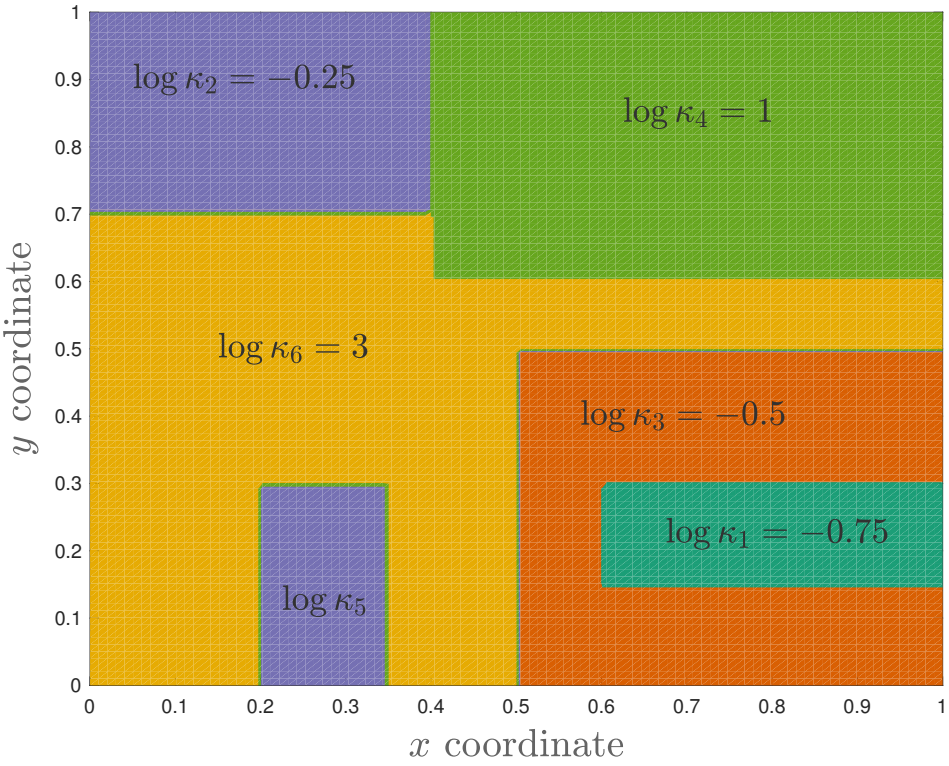


Figure 5. *The "true" log-conductivity field.*

Table 1

True values of the parameters for the tracer problem. The log-conductivity at location (x, y) is $\theta_{\underline{j}(x,y)}$, where $\underline{j}(x, y)$ is the smallest integer j such that $x_0^j \le x \le x_1^j$ and $y_0^j \le y \le y_1^j$; each parameter value $\theta_{\underline{j}}$ corresponds to $\log \kappa_{\underline{j}}$ in Figure 5.

Parameter	x_0^j	x_1^j	y_0^j	y_1^j	True value	Prior lower	Prior upper
θ_1	0.6	1	0.15	0.3	−0.75	−1	0
θ_2	0	0.4	0.7	0.1	−0.25	−1	1
θ_3	0.5	1	0	0.5	−0.5	−1	0
θ_4	0.4	1	0.6	1	1	0	2
θ_5	0.2	0.25	0	0.3	−0.25	−1	0
θ_6	0	1	0	1	3	2	5

The time-dependent tracer concentration $c(x, y, t)$ then evolves, given a flow-dependent dispersion tensor, via

$$\frac{\partial c}{\partial t} + \nabla \cdot \left(\left(d_m \mathbf{I} + d_l \begin{bmatrix} u^2 & uv \\ uv & v^2 \end{bmatrix} \right) \nabla c \right) - \begin{bmatrix} u \\ v \end{bmatrix} \cdot \nabla c = -f_t, \tag{5.5}$$

where $d_m = 2.5 \times 10^{-3}$ and $d_l = 2.5 \times 10^{-3}$ are dispersion coefficients and $f_t(x, y)$ is the tracer

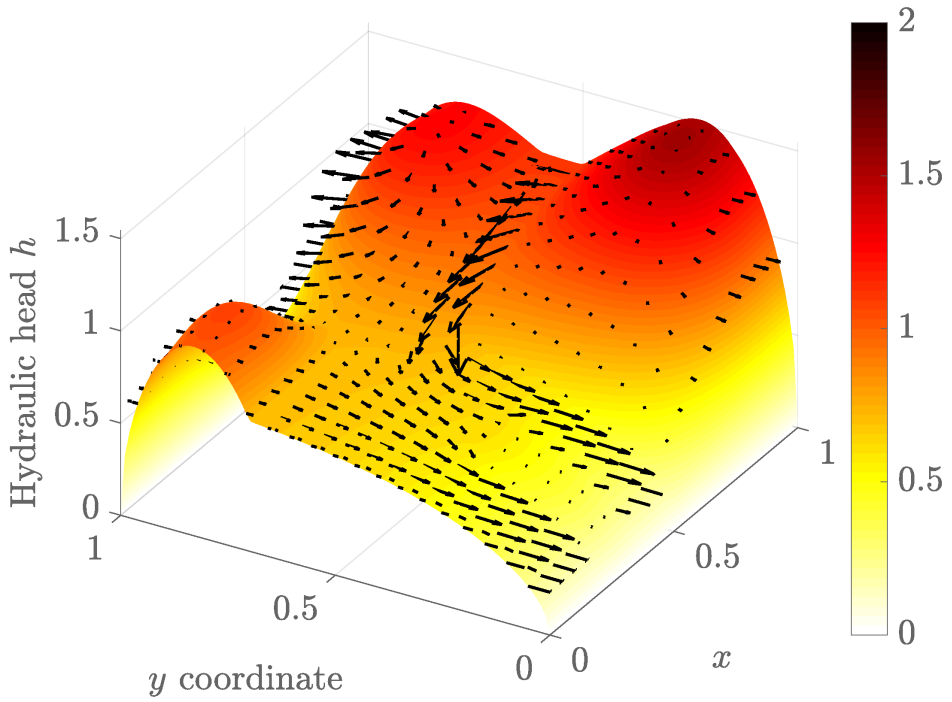


Figure 6. *Hydraulic head (colormap) $h(x, y)$ computed via (5.3) and corresponding velocities (5.4) (arrows), given the conductivity field in Figure 5.*

forcing. The tracer is forced by injection at each well location. The source term is similar to the one forcing the hydraulic head

$$f_t(x, y) = \sum_{i=1}^{4} r_i \exp\left(\frac{(a_i - x)^2 + (b_i - y)^2}{0.005}\right),$$

where $r_i \in (10, 5, 10, 5)$. The tracer has initial condition $c(x, y, 0) = 0$, and homogeneous Neumann conditions are enforced at all spatial boundaries. Since the hydraulic head forcing, tracer forcing, and dispersion coefficients are known, the forward model simply maps the conductivity to a time-evolving concentration field. Tracer observations are taken at 25 well locations: (x_i, y_j) such that $x_i = 0.1 + \frac{i-1}{5}$ and $y_j = 0.1 + \frac{j-1}{5}$ for $i, j \in \{1, \ldots, 5\}$ at successive times $t \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$.

The forward solver computes the steady state pressure and velocity fields, then simulates the tracer advection/diffusion. Figure 6 shows the hydraulic head and velocity fields resulting from the true log-conductivity, and Figure 7 shows the associated tracer concentration field at $t = 0.4$. Overall, the parameter-to-observable map, from the log-conductivities to the time-dependent tracer concentrations, is strongly nonlinear and challenging to approximate. Data for inversion are generated using a standard finite element scheme on a 200×200 mesh. The solver used for inversion (i.e., to evaluate the posterior density at a candidate value of θ) uses a coarser 100×100 mesh. In both cases (generating the data and within the inversion), time

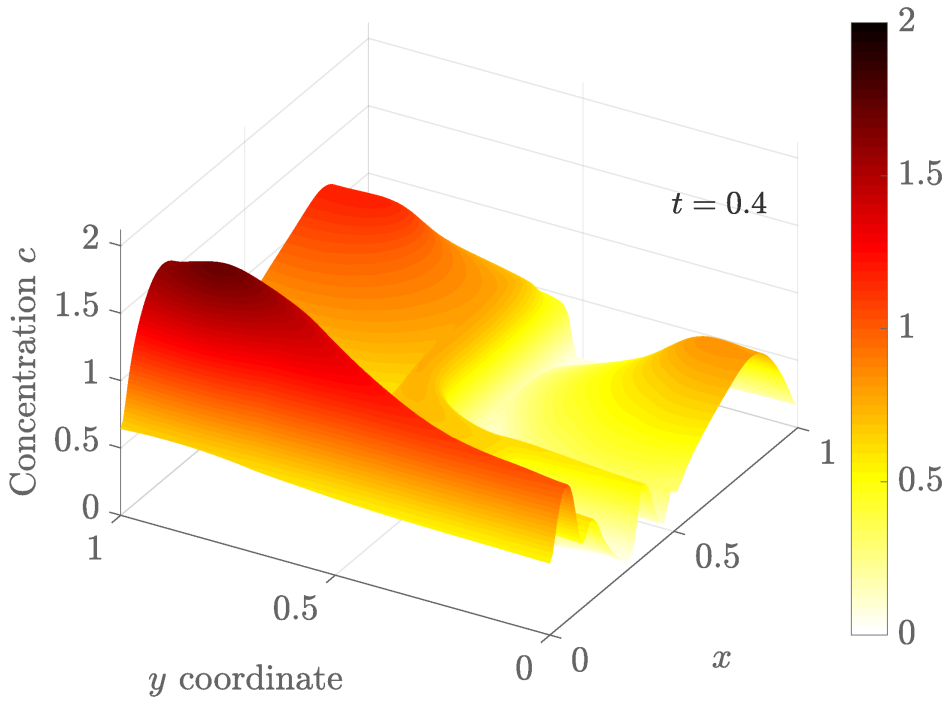
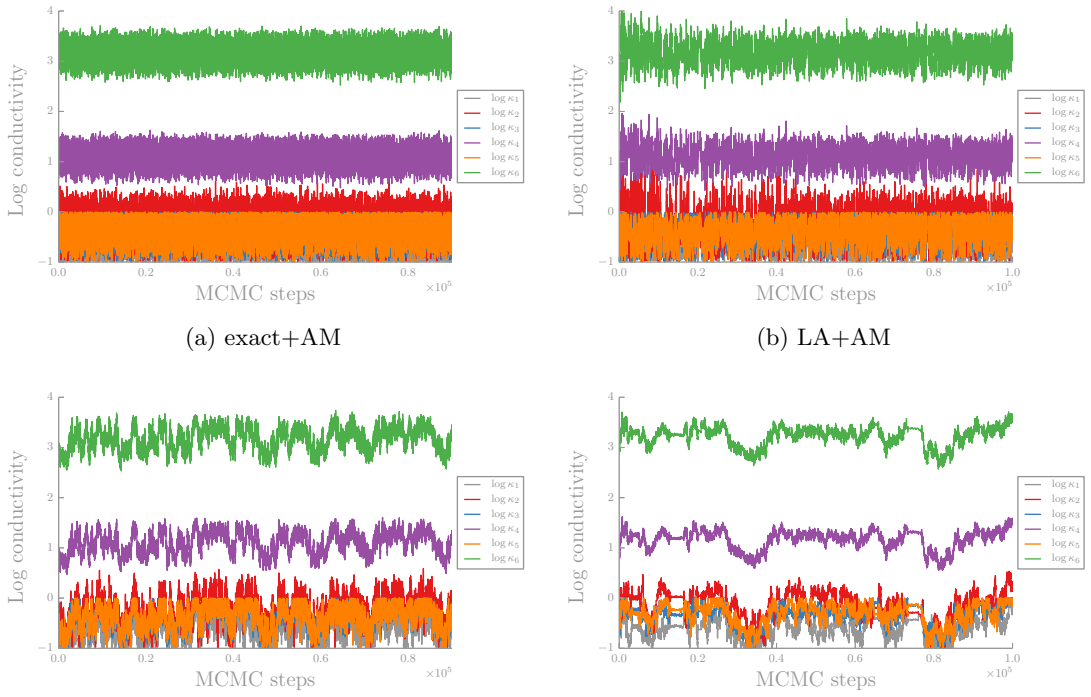


Figure 7. *The tracer concentration $c(x, y, t = 0.4)$, given the conductivity field in Figure 5. The tracer is injected from a well in each corner.*

integration of the contaminant concentration field uses a Crank–Nicolson scheme. The likelihood assumes additive and i.i.d. errors for each observation of tracer concentration, Gaussian with mean zero and variance 10^{-2}.

In a serial implementation, each evaluation of the forward model and hence the likelihood requires roughly 13 seconds of computation. Though we will mitigate this cost using local approximations, we also wish to compare our approach with chains that employ exact evaluations of the forward model. To make such comparisons feasible—and also to reflect computational practice for complex PDE models—we parallelize each forward model evaluation. We use four processors, which reduces the forward model's run time to roughly 4 seconds of computation. Thus our parallel MCMC scheme actually employs two levels of parallelism: an outer level involving parallel chains, as described in section 3, and an inner level within each forward model evaluation.

The posterior distribution in this problem has no standard analytical form. To establish a baseline for accuracy comparisons, we instead run 31 independent exact+AM chains. Each chain is 10^5 steps long, which requires several days (per chain) of computation. After discarding the first 10^4 samples of each chain as burn-in, the remaining samples are pooled and used to characterize the posterior distribution. Figure 8a shows a trace plot of one such exact+AM chain, for all six components of the state. Visually, the transient behavior of the chain appears exhausted well before 10^4 steps, justifying our choice of burn-in. One- and two-dimensional marginals of the posterior distribution, computed using the pooled exact+AM



(a) exact+AM (b) LA+AM

(c) exact+MALA (d) LA+MALA

Figure 8. *Tracer transport problem: Trace plots for a single MCMC chain (state versus MCMC iteration) either using exact density evaluations or employing a local approximation (LA), paired with either an AM or mMALA proposal.*

chains, are shown in Figure 9. The distribution has distinctly non-Gaussian structures, and the regions of high posterior probability seem to concentrate around the "true" parameters given in Table 1.

While the AM chains appear to mix well for this problem, mMALA proves far less effective. Figure 8c shows trace plots of an exact+mMALA chain targeting the same posterior. This calculation is rather laborious (over 415 hours), as direct evaluations of the gradient of the forward model are not available; instead we compute the gradients using finite differences. This simulation is not intended as a practical approach, but rather to assess the performance of mMALA in the absence of local approximations. We find that the chain mixes quite poorly; the ESS after 10^5 MCMC steps is only 80. Based on the results of section 5.1, we do not expect mMALA paired with local approximations to fare any better and, indeed, Figure 8d shows that mixing is poor for an LA+mMALA chain. Given these results, we focus the rest of this section on AM chains, with a goal of exploring the performance of parallel LA schemes. More broadly, we note that there is no guarantee that MALA schemes should improve over adaptive Metropolis (or even simple random-walk Metropolis) in low-dimensional problems such as those considered here. The potential for such improvements is problem-dependent and sometimes rather delicate, as was recognized almost immediately when MALA was introduced [42].

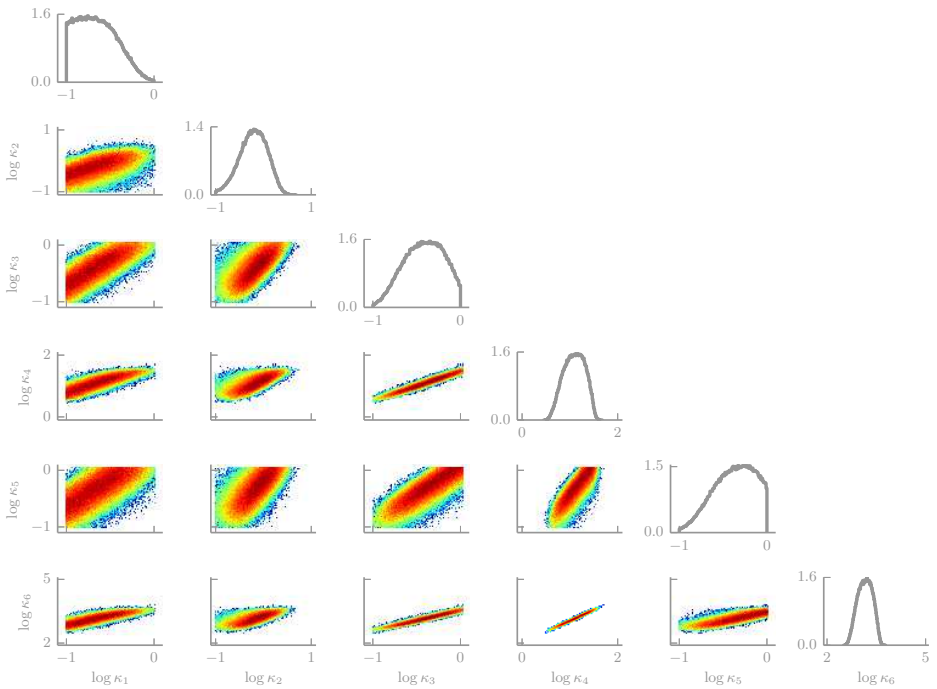


Figure 9. *One- and two-dimensional posterior marginals of the parameters in the hydrologic tracer transport problem. Bounds on each subplot axis are the upper and lower bounds for the uniform prior on the corresponding parameter (Table 1).*

We first examine the convergence of estimates produced by *single* LA+AM chains. Algorithm settings are given in Appendix A, and code for this example is provided in the supplementary material (M108408_01.zip [local/web 39.4KB]). We run 51 independent chains, again discarding the first 10^4 samples of each chain as burn-in. For consistency, we simply choose the same burn-in period for the exact chains and LA chains. If anything, this choice is less favorable to LA—though asymptotically it is immaterial. The mixing of a single LA+AM chain is visualized by the trace plot in Figure 8b. Initially, the chain does not mix as quickly as in the exact+AM case, but mixing improves as the approximation is refined, and overall the chain appears to explore the posterior quite efficiently. We also emphasize that the horizontal axis in Figure 8b does not reflect computational cost, since the latter is dominated by target density evaluations rather than MCMC steps.

To assess error versus computational cost, Figure 10 shows, for each individual chain, the squared relative error in a running posterior covariance estimate versus wall clock time. The squared relative error $\varepsilon_t^{2,(i)}$ is defined in (5.2), where the reference value C_0 of the posterior covariance is computed by pooling all 2.79×10^6 available exact+AM samples. For comparison, we also plot error versus run time for 31 exact+AM chains. When reporting wall clock times here and below, we include the computational cost of the entire chain, including the cost of portions discarded as burn-in. Error in the LA chains decreases steadily and reaches an accuracy comparable to the exact chains, but with significantly shorter run times. We also

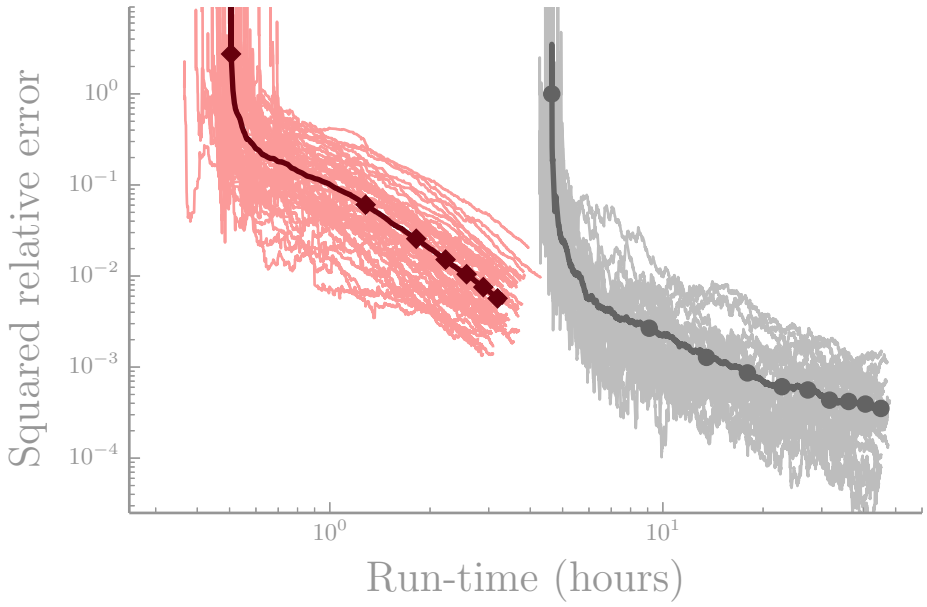


Figure 10. *Tracer transport problem: Relative squared error in the posterior covariance estimates produced by independent single (i.e., not parallel) AM chains, versus run time. The light gray lines correspond to* 31 *independent exact+AM chains, each of length* 10^5. *The dark gray line shows the expected error for this exact case. The light red lines correspond to* 51 *single LA+AM chains, each of length* 2×10^5. *The dark red line shows the expected error in the approximate case.*

notice that decay rate of the expected error (bold red line in Figure 10) in the LA case seems to accelerate. As noted in the quartic example (where longer chains accentuated this trend), this acceleration is due to the fact that refinements happen less frequently as the chain progresses, while additional MCMC steps continue to reduce the error.

The local approximation sampler becomes even more effective in a parallel chain setting, where concurrent chains are allowed to share posterior density evaluations by building a common $\mathcal{S}_t$. The colored lines in Figure 11 show error versus run time for increasing levels of parallelism k, from 1 to 30 chains. To assess the variability of the error, each k-chain simulation is repeated several times; each such realization is shown in the figure. Each individual LA+AM chain (within a group of k) has a fixed length of 10^5 steps and, as before, the first 10^4 samples of each chain are discarded as burn-in. The error plotted on the vertical axis is again the squared relative error in the posterior covariance. Two trends are visible in the colored lines. First, as the number of chains increases, the error decreases. In and of itself, this is not surprising: summing across the chains, we accumulate more MCMC samples and, along the way, seek more model evaluations to refine the local approximations (this will be quantified precisely in subsequent figures). But the colored lines *also* move to the left as the number of parallel chains increases; in other words, both the error *and* the run time are reduced. This trend contrasts with that obtained by simply running exact+AM chains in parallel, an exercise depicted by the gray lines in Figure 11. Using this naive parallelization, adding more chains decreases the sampling error but does not affect the run time. Moreover, the run times

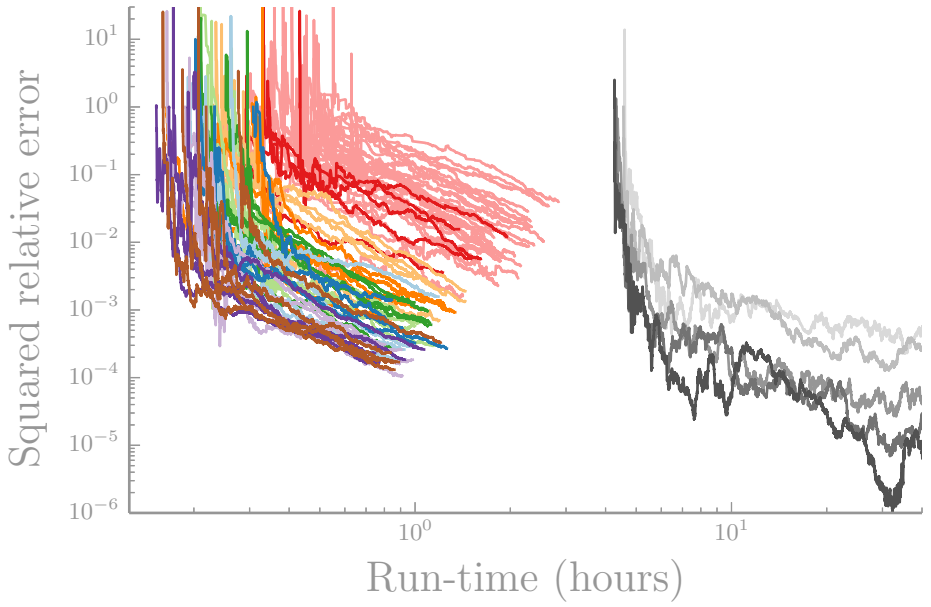


Figure 11. *Tracer transport problem: Relative squared error in the posterior covariance estimates obtained from parallel MCMC chains. The gray lines are computed using exact target density evaluations for* $k \in \{1, 2, 4, 8, \text{ or } 16\}$ *chains. Darker shades correspond to simulations with more parallel chains. The colored lines are computed using local approximation MCMC. We use* $k \in \{1, 2, 4, 6, 8, 10, 13, 16, 20, 25, \text{or } 30\}$ *chains corresponding to light red, red, light orange, orange, light green, green, light blue, blue, light purple, purple, and brown, respectively. The error is that of a running covariance estimate obtained by pooling samples from the* k *concurrent chains. Sharing posterior density evaluations shortens the run time* and *reduces the error.*

of LA+AM are one to two orders of magnitude smaller for comparable errors.

We can also characterize the behavior of parallel local approximations by evaluating ESS as a function of computational effort. Figure 12 shows ESS as a function of wall clock time. First, as a baseline, consider again running exact+AM chains of length 10^5 in parallel, depicted by gray and black circles. We certainly expect parallel chains to yield a larger ESS once their samples are pooled, and indeed the circles jump upwards as we increase the number of concurrent chains from 1 to 30. Increasing the number of chains in the exact case does not, however, change the time it takes to simulate each chain; thus the gray and black dots are vertically aligned at the same run times. In the parallel LA+AM cases, depicted by colored diamonds, the story is more interesting. As the number of parallel chains increases, the symbols move upwards *and* to the left, reflecting decreased run times. Several independent realizations of each parallel case are presented, since the simulations are not deterministic. Note that the ESS of a single LA+AM chain (light red) is lower than that of an exact+AM chain of the same length; this is expected, given the mixing comparison at the top of Figure 8. Similarly, 30 parallel exact+AM chains have a higher combined ESS than 30 parallel LA+AM chains (the brown diamonds of Figure 12). But the latter entail a vastly smaller computational effort. Because of the collaboration among chains, we can compute a larger

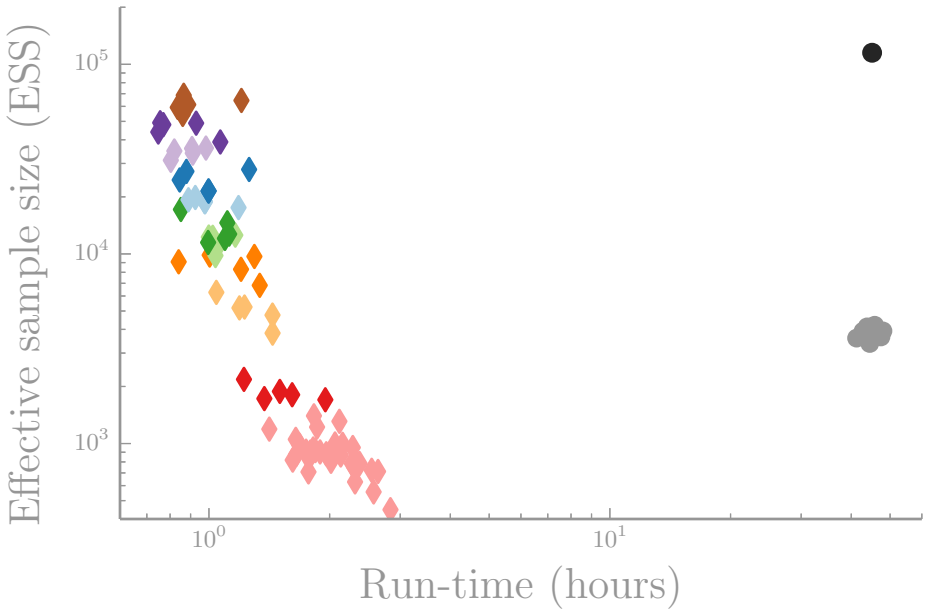


Figure 12. *Results of the parallel efficiency study on the tracer transport problem, comparing run time to the total ESS across parallel chains. Each symbol represents one (parallel) experiment. Light gray circles and light red diamonds correspond to single chains of length* 10^5 *(with* 10^4 *burn-in) using exact evaluations and local approximation, respectively. Each colored diamond represents a different number* k *of parallel LA+AM chains,* $k \in \{1, 2, 4, 6, 8, 10, 13, 16, 20, 25, \text{or } 30\}$*, and the colors are as in Figure 11. The black circle corresponds to* 30 *parallel exact+AM chains. Using local approximations, running more chains increases ESS* and *decreases run time.*

number of independent samples in less time.

Our second comparison uses a more stringent measure of parallel efficiency: ESS per chain-hour, i.e., the total ESS divided by the number of chains and the wall clock time. This measure removes the intrinsic advantage of having multiple chains. A naive MCMC parallelization yields no improvement in efficiency according to this metric: the number of independent samples might grow linearly with the number of chains, but this growth is normalized away. Figure 13 shows this behavior for exact+AM chains using gray circles. In contrast, the results of parallel local approximation, depicted by colored diamonds, show steady gains in ESS/(chain-hour) with additional parallel chains. This gain is the result of collaboration among the chains in the most computationally expensive element of the inference problem—evaluating the posterior density—by sharing evaluations from which we construct a shared surrogate model. We note that the total number of model evaluations performed during the parallel experiments is still higher than in a single-chain case, but since the additional evaluations are parallelized, the run time is shorter.

5.3. Shallow-shelf ice stream model. Continental ice sheets are divided into basins that are drained by fast-flowing river-like ice streams. These ice streams regulate the discharge of ice mass into the ocean, and hence play a key role in determining the overall behavior of the ice sheet. The Intergovernmental Panel on Climate Change has identified the Antarctic

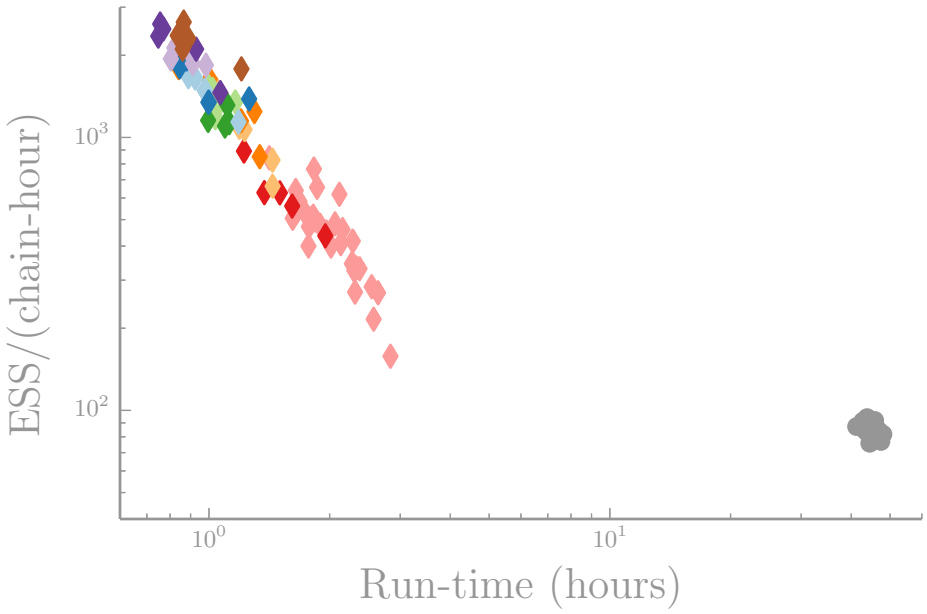


Figure 13. *Results of the parallel efficiency study on the tracer transport problem, comparing run time to the effective number of samples produced per chain-hour. Each symbol represents one (parallel) experiment. Again, colors from light red to brown correspond to more parallel chains, $k \in \{1, 2, 4, 6, 8, 10, 13, 16, 20, 25, \text{or } 30\}$. Using parallel local approximations, ESS per chain-hour increases with the number of chains.*

contribution to sea-level rise as an important source of uncertainty in climate projections, and ice streams have become a widespread topic of study [10, 24].

Ice stream dynamics are not completely understood, nor are the factors governing their dynamics. Although satellite data provide plentiful observations of topology and surface velocities [19, 30, 47], basal properties such as the friction between the base of the ice and the underlying ground—the *basal friction*—are difficult or impossible to observe directly. The basal friction varies widely and may be higher if the ice is scraping directly against rough bedrock or lower if the ice rests on till, a mixture of mud and rock that lubricates the interface. The basal friction also parameterizes basal lubrication caused by melting basal ice (possibly due to geothermal or frictional heating). Previous work infers basal friction given surface velocity observations [32, 39]; quantifying uncertainty in the basal friction, however, requires considerable computational expense and/or posterior approximations [38]. In this example, we explore the problem of inferring the basal friction from surface velocities, employing local approximations to reduce the computational cost of MCMC.

Ice is often modeled as a highly viscous, non-Newtonian, and incompressible fluid. In particular, the shallow-shelf approximation [31, 32, 33] describes ice stream velocity assuming that (i) the horizontal extent ($\mathcal{O}(100\,\text{km})$) is much larger than the vertical extent ($\mathcal{O}(1\,\text{km})$), and (ii) the vertical velocity is zero. The nondimensionalized shallow-shelf equations for a

two-dimensional horizontal domain $[0,1]^2 \ni (x,y)$ are

$$\frac{\partial}{\partial x}\left(2\nu h\left(2\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)\right)+\frac{\partial}{\partial y}\left(\nu h\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\right)-\beta\left|u\right|^{m-1}u=h\frac{\partial s}{\partial x},$$
$$\frac{\partial}{\partial x}\left(2\nu h\left(2\frac{\partial v}{\partial y}+\frac{\partial u}{\partial x}\right)\right)+\frac{\partial}{\partial y}\left(\nu h\left(\frac{\partial v}{\partial x}+\frac{\partial u}{\partial y}\right)\right)-\beta\left|v\right|^{m-1}v=h\frac{\partial s}{\partial y},$$

with boundary conditions

$$u=0 \quad \text{and} \quad v=-1 \quad \text{at} \quad x=0 \quad \text{and} \quad x=1,$$
$$\tfrac{\partial u}{\partial y}+\tfrac{\partial v}{\partial x}=0 \quad \text{at} \quad y=0 \quad \text{and} \quad y=1, \quad \text{and} \quad 2\tfrac{\partial v}{\partial y}+\tfrac{\partial u}{\partial x}=0 \quad \text{at} \quad y=1,$$

where

$$\nu=\frac{1}{2}\left(\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial v}{\partial y}\right)^2+\frac{1}{4}\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)^2+\frac{\partial u}{\partial x}\frac{\partial v}{\partial y}\right)^{-\frac{n-1}{2n}}$$

is the velocity-dependent viscosity [31, 32, 33]. Assuming that the surface elevation $s(x,y)$ and ice thickness $h(x,y)$ are known and that $n=\frac{1}{m}=3$, the forward model maps realizations of the basal friction $\beta(x,y)$ to the horizontal velocities $u(x,y)$ and $v(x,y)$.

To define our Bayesian inference problem, we endow the log-basal friction field $\log\beta(x,y)$ with a Gaussian process prior, using an isotropic squared-exponential covariance kernel,

$$C\left((x_1,y_1),(x_2,y_2)\right)=\sigma^2\exp\left(-\frac{(x_1-x_2)^2+(y_1-y_2)^2}{2l^2}\right),$$

with correlation length $l=0.1$ and variance $\sigma^2=25$. This field is easily parameterized with a Karhunen–Loève expansion [1]:

$$\beta(x,y;\theta)\approx\exp\left(\sum_{i=1}^{d}\theta_i\sqrt{\lambda_i}\varphi_i(x,y)\right),$$

where λ_i and $\varphi_i(x,y)$ are the eigenvalues and eigenfunctions, respectively, of the integral operator on $[0,1]^2$ defined by the kernel C, and the parameters θ_i inherit independent standard normal priors, $\theta_i\sim\mathcal{N}(0,1)$. We truncate the Karhunen–Loève expansion at $d=12$ modes and infer the weights $(\theta_1,\ldots,\theta_{12})$ from data. The true basal diffusivity field is shown in Figure 14.

Data arise from observations of the velocity field on a uniform 10×10 grid covering the unit square, $(x_i,y_i)\in\{(.05,.05),\ldots,(.95,.95)\}$, as depicted in Figure 15. Both the u and v components of velocity are observed, and observational errors are taken to be independent, additive, and identically Gaussian, $\mathcal{N}(0,0.01^2)$. To avoid an "inverse crime" [26], data are generated with a 25×25 mesh, but inference uses a coarser 15×15 mesh.

The posterior distribution of the basal friction field is quite challenging to sample, as the forward model requires, on average, 26 seconds per evaluation. Using a direct MCMC approach, a numerical simulation comprising 10 parallel chains of 200,000 steps each would therefore take nearly two months to run. Using LA+AM on 10 parallel chains, we complete exactly the same simulation in just over one day, a nearly 60-fold improvement in the run

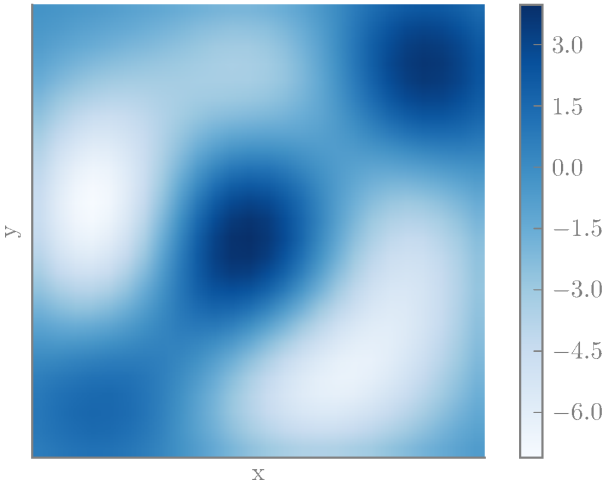


Figure 14. *Ice stream inference problem: The true log-basal friction field,* $\log \beta(x, y)$.

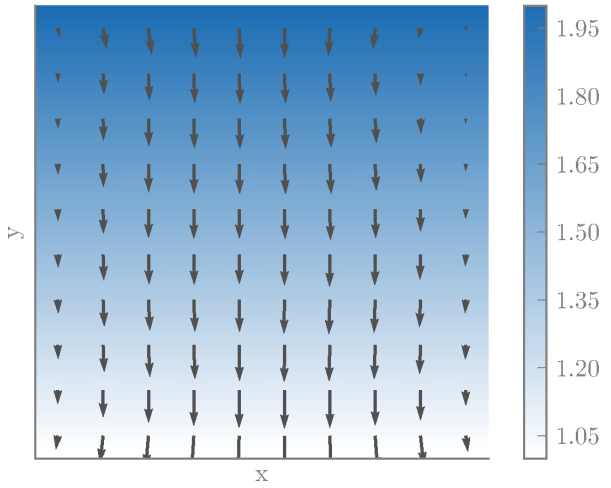


Figure 15. *Ice stream inference problem: The assumed ice height field* $h(x, y)$ *(shading) superimposed on the observed velocity field (vectors), given the basal friction in Figure 14. Note the left-right asymmetry in the velocity field at the top of the domain, induced by the high friction region at the top right.*

time. Representative one- and two-dimensional marginals of the posterior (focusing on only the first 6 of 12 dimensions) are shown in Figure 16. Note that several parameters are strongly correlated, and that many marginal distributions appear skewed and non-Gaussian. These 2 million samples were produced using only about 35,000 runs of the forward model.

6. Conclusions. This work has extended our previous development of asymptotically exact MCMC algorithms that employ local approximations of expensive models. We lifted restrictive assumptions on the type of MCMC kernel that could be used—in particular, allowing the proposal distribution to extract derivatives, and hence geometric information, from the approximation. Doing so enables a wide variety of more sophisticated proposal distributions,

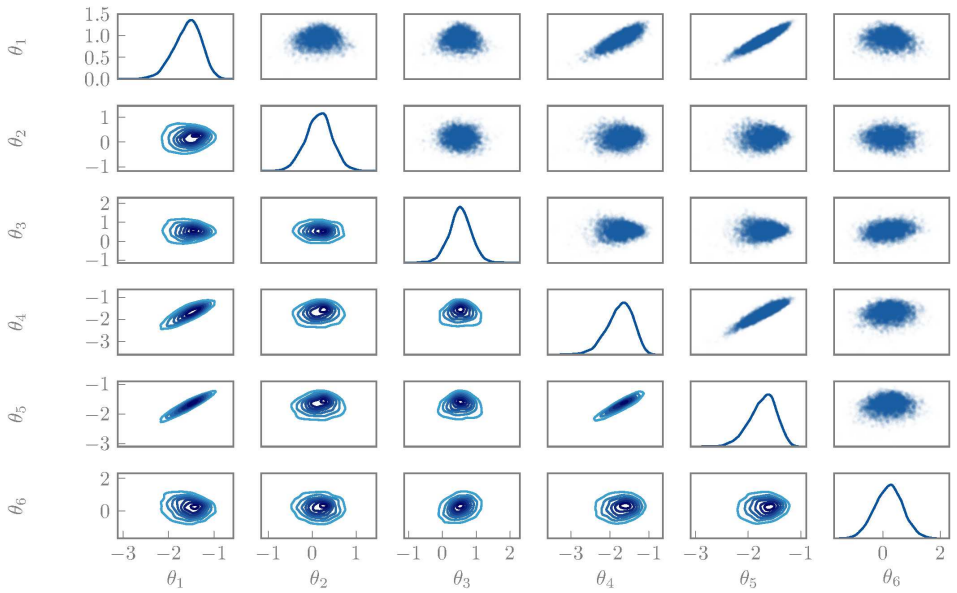


Figure 16. *One- and two-dimensional posterior marginals of the first six parameters in the ice stream inference problem.*

such as manifold MALA, to be applied in settings where they would otherwise be intractable (e.g., when forward model derivatives cannot be directly evaluated) or unaffordable. Additionally, we showed that using approximations allows the most computational intensive element of many MCMC simulations—the forward model or likelihood evaluations—to be directly parallelized through the shared and online construction of a posterior-adapted set of samples. Sharing this set of model evaluations among multiple MCMC chains drives the construction of local approximations on each chain, providing a novel and effective means of reducing the run time of MCMC simulations. Our shared local approximation scheme can readily be paired with other MCMC parallelization schemes, e.g., methods that use the presence of multiple chains to improve mixing; this is a natural avenue for future work.

To demonstrate the practical utility of these developments, we presented two challenging inference problems that we believe reflect scientifically interesting settings where forward models are necessarily expensive. Using parallel computing resources, we demonstrated a nearly two-orders-of-magnitude improvement in the run time of a groundwater hydrology inference problem, and a roughly 60-fold reduction in the run time of an ice stream inference problem. These results suggest that our approach may help make a range of challenging Bayesian inference problems feasible. A reusable and open source implementation of this algorithm is available as part of the MIT Uncertainty Quantification (MUQ) library (http://muq.mit.edu).

Appendix A. Complete algorithm description. This appendix provides a complete description of the local approximation MCMC algorithm from [9], extended here to MCMC proposals that also employ the approximation $\tilde{\mathbf{f}}$. We replicate necessary subroutines from [9]; for a full discussion and derivation of these methods, please see that paper. The sketch

given in Algorithm 2 of section 2 is expanded here into Algorithm 3, which takes additional parameters β_t and γ_t that determine when refinement is performed according to random or cross-validation criteria, respectively. The choice of γ_t is arbitrary, but $\sum_t \beta_t$ must diverge; based on the parameter study in [9], the numerical experiments in section 5 are performed with $\beta_t = 0.01t^{-0.2}$ and $\gamma_t = 0.1t^{-0.1}$. These numerical experiments employ local quadratic approximations, as described below. Code used to run the examples, in conjunction with MUQ, is provided in the supplementary material (M108408_01.zip [local/web 39.4KB]).

Algorithm 3. Metropolis–Hastings with local approximations and general proposals.

1: **procedure** RunChain($\mathbf{f}, r, q, \theta_1, \mathcal{S}_1, \ell, \mathbf{d}, p, T, \{\beta_t\}_{t=1}^T, \{\gamma_t\}_{t=1}^T$)
2: **for** $t = 1 \dots T$ **do**
3: $(\theta_{t+1}, \mathcal{S}_{t+1}) \leftarrow K_t(\theta_t, \mathcal{S}_t, \ell, \mathbf{d}, p, \mathbf{f}, r, q, \beta_t, \gamma_t)$
4: **end for**
5: **end procedure**

6: **procedure** $K_t(\theta^-, \mathcal{S}, \ell, \mathbf{d}, p, \mathbf{f}, r, q, \beta_t, \gamma_t)$
7: Draw proposal $\mathbf{z}_t \sim \mathcal{N}(0, I)$.
8: $\tilde{\mathbf{f}}^- \leftarrow$ LocApprox$(\theta^-, \mathcal{S}, \emptyset)$.
9: $\theta^+ \leftarrow r(\theta^-, \mathbf{z}_t, \tilde{\mathbf{f}}^-)$.
10: $\tilde{\mathbf{f}}^+ \leftarrow$ LocApprox$(\theta^+, \mathcal{S}, \emptyset)$.
11: $\alpha \leftarrow \min\left(1, \frac{\ell(\theta^+|\mathbf{d},\tilde{\mathbf{f}}^+)p(\theta^+)q(\theta^+,\theta^-|\tilde{\mathbf{f}}^+)}{\ell(\theta^-|\mathbf{d},\tilde{\mathbf{f}}^-)p(\theta^-)q(\theta^-,\theta^+|\tilde{\mathbf{f}}^-)}\right)$. ▷ Compute nominal acceptance ratio
12: Compute ϵ^+ and ϵ^- as in (A.1)–(A.2).
13: **if** $u \sim \text{Uniform}(0,1) < \beta_m$ **then** ▷ Refine with probability β_m
14: Randomly, $\mathcal{S} \leftarrow$ RefineNear$(\theta^+, \mathcal{S})$ or $\mathcal{S} \leftarrow$ RefineNear$(\theta^-, \mathcal{S})$.
15: **else if** $\epsilon^+ \geq \epsilon^-$ and $\epsilon^+ \geq \gamma_m$ **then** ▷ If needed, refine near the larger error
16: $\mathcal{S} \leftarrow$ RefineNear$(\theta^+, \mathcal{S})$
17: **else if** $\epsilon^- > \epsilon^+$ and $\epsilon^- \geq \gamma_m$ **then**
18: $\mathcal{S} \leftarrow$ RefineNear$(\theta^-, \mathcal{S})$
19: **end if**
20: **if** refinement occurred **then** repeat from line 8.
21: **else** ▷ Evolve chain using approximations
22: Draw $u \sim \text{Uniform}(0,1)$. If $u < \alpha$, **return** $(\theta^+, \mathcal{S})$, else **return** $(\theta^-, \mathcal{S})$.
23: **end if**
24: **end procedure**

Algorithm 4 provides several subroutines. The first, LocApprox, gathers the N nearest neighbors from $\mathcal{S}_t$ to use in constructing the approximation; for quadratics, $N = \frac{\sqrt{d}(d+1)(d+2)}{2}$. The operator $\mathcal{A}^{\sim j}_{\mathcal{B}(\theta,R)}$ constructs the local approximation; in this work, it fits a quadratic (a degree-two polynomial) with least squares. The input j facilitates cross-validation and, unless $j = \emptyset$, designates that the jth neighbor should be omitted. The second routine, RefineNear, solves a local optimization problem to choose a new point θ^* that is near θ but space-filling overall; this point is used to enrich $\mathcal{S}_t$.

Cross-validation is used to estimate the error in the acceptance probability evaluated using

Algorithm 4. Supporting algorithms.

procedure LocApprox($\theta, \mathcal{S}, j$)
  Select R so that $|\mathcal{B}(\theta, R)| = N$, where $\mathcal{B}(\theta, R) := \{(\theta_i, \mathbf{f}(\theta_i)) \in \mathcal{S} : \|\theta_i - \theta\|_2 \le R\}$. ▷ Select ball of points
  $\tilde{\mathbf{f}} \leftarrow \mathcal{A}^{\sim j}_{\mathcal{B}(\theta,R)}$. ▷ Fit local approximation
  return $\tilde{\mathbf{f}}$
end procedure

procedure RefineNear($\theta, \mathcal{S}$)
  Select R so that $|\mathcal{B}(\theta, R)| = N$. ▷ Select ball of points
  $\theta^* \leftarrow \arg\max_{\|\theta' - \theta\| \le R} \min_{\theta_i \in \mathcal{S}} \|\theta' - \theta_i\|$. ▷ Optimize near θ
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{\theta^*, \mathbf{f}(\theta^*)\}$. ▷ Grow the sample set
  return $\mathcal{S}$
end procedure

the approximations. Define the nominal and leave-one-out variants of the approximations, for $j = 1, \ldots, N$, as

$$\tilde{\mathbf{f}}^+ = \text{LocApprox}(\theta^+, \mathcal{S}, \emptyset), \qquad \tilde{\mathbf{f}}^+_{\sim_j} = \text{LocApprox}(\theta^+, \mathcal{S}, j),$$
$$\tilde{\mathbf{f}}^- = \text{LocApprox}(\theta^-, \mathcal{S}, \emptyset), \qquad \tilde{\mathbf{f}}^-_{\sim_j} = \text{LocApprox}(\theta^-, \mathcal{S}, j).$$

Then compute the approximate posterior ratio and all the leave-one-out variants (here, slightly modified from our original work to include the proposal densities):

$$\zeta := \frac{\ell(\theta^+|\mathbf{d}, \tilde{\mathbf{f}}^+) p(\theta^+) q(\theta^+, \theta^-|\tilde{\mathbf{f}}^+)}{\ell(\theta^-|\mathbf{d}, \tilde{\mathbf{f}}^-) p(\theta^-) q(\theta^-, \theta^+|\tilde{\mathbf{f}}^-)},$$
$$\zeta^{+,\sim j} := \frac{\ell(\theta^+|\mathbf{d}, \tilde{\mathbf{f}}^+_{\sim_j}) p(\theta^+) q(\theta^+, \theta^-|\tilde{\mathbf{f}}^+_{\sim_j})}{\ell(\theta^-|\mathbf{d}, \tilde{\mathbf{f}}^-) p(\theta^-) q(\theta^-, \theta^+|\tilde{\mathbf{f}}^-)},$$
$$\zeta^{-,\sim j} := \frac{\ell(\theta^+|\mathbf{d}, \tilde{\mathbf{f}}^+) p(\theta^+) q(\theta^+, \theta^-|\tilde{\mathbf{f}}^+)}{\ell(\theta^-|\mathbf{d}, \tilde{\mathbf{f}}^-_{\sim_j}) p(\theta^-) q(\theta^-, \theta^+|\tilde{\mathbf{f}}^-_{\sim_j})}.$$

Finally, find the maximum difference between the α values computed using ζ and those computed using the leave-one-out variants $\zeta^{+,\sim j}$ and $\zeta^{-,\sim j}$, averaging over the forward and reverse directions. These are the error indicators:

$$\epsilon^+ := \max_j \left(\left| \min(1, \zeta) - \min\left(1, \zeta^{+,\sim j}\right) \right| + \left| \min\left(1, \frac{1}{\zeta}\right) - \min\left(1, \frac{1}{\zeta^{+,\sim j}}\right) \right| \right), \tag{A.1}$$

$$\epsilon^- := \max_j \left(\left| \min(1, \zeta) - \min\left(1, \zeta^{-,\sim j}\right) \right| + \left| \min\left(1, \frac{1}{\zeta}\right) - \min\left(1, \frac{1}{\zeta^{-,\sim j}}\right) \right| \right). \tag{A.2}$$

Appendix B. Proofs of the main results. Throughout this section, we use the notation $f(x) = O(g(x))$ to mean that there exists some constant $0 < C < \infty$ so that $f(x) \le Cg(x)$. If the constant C depends on an important parameter, we sometimes use that parameter as a

subscript for emphasis; for example, $\frac{x^2}{p} = O_p(x^4)$ for all fixed $p > 0$, but there is no constant $C < \infty$ so that $\frac{x^2}{p} \le Cx^4$ uniformly in $p > 0$.

For any pair of measures μ, ν on a metric space $(\mathcal{X}, d)$, denote by $\Pi(\mu, \nu)$ the collection of all pairs of random variables $(X, Y) \in \mathcal{X}^2$ that have marginal distributions $\mathcal{L}(X) = \mu$, $\mathcal{L}(Y) = \nu$. Recall that the *Wasserstein metric* on measures on a metric space $(\mathcal{X}, d)$ is given by

$$W_d(\mu, \nu) = \inf_{(X,Y)\in\Pi(\mu,\nu)} \mathbb{E}[d(X, Y)].$$

We also use the shorthand $W_p \equiv W_{\|\cdot\|_p}$ when $1 \le p \le \infty$. The *total variation* distance between two probability measures μ, ν is given by $\|\mu - \nu\|_{\mathrm{TV}} = W_\rho(\mu, \nu)$, where $\rho(x, y) \equiv \mathbf{1}_{x\neq y}$. The *mixing time* of a Markov chain $\{Z_t\}_{t\ge0}$ with stationary distribution π on state space Ω is

$$\tau_{\mathrm{mix}} = \inf\left\{t : \sup_{Z_0=z\in\Omega} \|\mathcal{L}(Z_t) - \pi\|_{\mathrm{TV}} < \frac{1}{4}\right\}.$$

Proof of Theorem 2. Denote the diameter of Ω by D_Ω and the mixing time of K_∞ by τ_{mix}; by assumptions 1 and 2 of Definition 3.1, respectively, $D_\Omega, \tau_{\mathrm{mix}} < \infty$. For $\epsilon > 0$, let $\tau_\epsilon = \inf\{t > 0 : \mathcal{S}_t \text{ is an } \epsilon\text{-cover of } \Theta\}$. By substituting τ_ϵ for τ everywhere that it is used, the proof of Lemma B.4 of [9] shows that

$$\mathbb{P}[\tau_\epsilon < \infty] = 1 \tag{B.1}$$

for all $\epsilon > 0$.

Next, fix $S, T \in \mathbb{N}$ and $\psi, \delta, \varphi_0 > 0$, and let $\epsilon = \epsilon(\delta)$ be the smaller of the values of $\epsilon(\delta)$ from inequalities (4.2), (4.4). Let $\mathcal{F}_T$ be the σ-algebra $\sigma(\{X_t, \mathcal{S}_t\}_{0\le t\le T})$. We will let $\{Y_t\}_{t\ge T}$ be a Markov chain with transition kernel K_∞ started at $Y_T = X_T$, and we will let $\{Z_t\}_{t\ge T}$ be a Markov chain with transition kernel K_∞ started at the distribution $\mathcal{L}(Z_T) = \pi$. We now describe a coupling of the three stochastic processes $\{X_t\}_{T\le t\le T+S}$, $\{Y_t\}_{T\le t\le T+S}$, and $\{Z_t\}_{T\le t\le T+S}$. We couple $\{Y_t\}_{T\le t\le T+S}$, $\{Z_t\}_{T\le t\le T+S}$ so that

$$\mathbb{P}[Y_{T+S} = Z_{T+S}|Y_T, Z_T] = \|\mathcal{L}(Y_{T+S}|Y_T) - \mathcal{L}(Z_{T+S}|Z_T)\|_{\mathrm{TV}}. \tag{B.2}$$

At least one coupling with this property exists by the definition of the total variation distance; choose one such coupling arbitrarily. We then couple $\{X_t\}_{T\le t\le T+S}$ to $\{Y_t\}_{T\le t\le T+S}$ iteratively in t. Denote by $\tilde{X}$ the value that would be returned in the tth iteration of Algorithm 3 if line 21 were ignored, and let $\mathbf{z}$ be the value obtained in line 7. Then, (X_{t+1}, Y_{t+1}) can be coupled conditional on $(X_t, Y_t, \mathcal{S}_t)$ so that

$$\begin{aligned}\mathbb{E}[\|X_{t+1} - Y_{t+1}\|_2] \le \mathbb{E}[\|X_{t+1} - \tilde{X}\|] + \mathbb{E}[\|\tilde{X} - Y_{t+1}\|] \le \delta + D_\Omega \mathbb{P}[\|\mathbf{z}\| > \varphi_0] + \frac{\psi}{S+1} \\ + \sup_{\theta,\theta'\in\Theta,\, \|\theta-\theta'\|<\|X_t-Y_t\|} W_2(K_{\mathcal{S}_t}(\theta, \cdot), K_\infty(\theta', \cdot)).\end{aligned} \tag{B.3}$$

Such a coupling exists by inequality (4.4) and the definition of the Wasserstein distance. By the "gluing" lemma (Chapter 1 of [49]), it is possible to combine the couplings of $\{X_t, Y_t\}_{T\le t\le T+S}$

and $\{Y_t, Z_t\}_{T \le t \le T+S}$ into a single coupling $\{X_t, Y_t, Z_t\}_{T \le t \le T+S}$ that satisfies both inequality (B.2) and also inequality (B.3) for all $T \le t < T+S$. Under this coupling,

$$W_2(Y_{T+S}, Z_{T+S}) \le D_\Omega \mathbb{P}[Z_{T+S} \ne Y_{T+S}] \le D_\Omega 2^{-\lfloor \frac{S}{\tau_{\text{mix}}} \rfloor}. \tag{B.4}$$

Let η_0 be as in the requirements for (4.3). By inequalities (4.2) and (B.3), we have for $T \le t < T+S$ that

$$\begin{aligned}
\mathbb{E}[\|X_{t+1} - Y_{t+1}\|_2 \,|\mathcal{F}_T] &= \mathbb{E}[\|X_{t+1} - Y_{t+1}\|_2 \mathbf{1}_{T \ge \tau_\epsilon} |\mathcal{F}_T] + \mathbb{E}[\|X_{t+1} - Y_{t+1}\|_2 \mathbf{1}_{T < \tau_\epsilon} |\mathcal{F}_T] \\
&\le \delta + D_\Omega \mathbb{P}[\|\mathbf{z}\| > \varphi_0] + \frac{\psi}{S+1} + \mathbb{E}\Big[\sup_{\theta, \theta' \in \Theta,\, \|\theta - \theta'\| < \|X_t - Y_t\|} W_2(K_\infty(\theta, \cdot), K_\infty(\theta', \cdot)) |\mathcal{F}_T \Big] \\
&+ \mathbb{E}\Big[\sup_{\theta \in \Theta} W_2(K_{\mathcal{S}_t}(\theta, \cdot), K_\infty(\theta, \cdot)) \mathbf{1}_{T \ge \tau_\epsilon} |\mathcal{F}_T \Big] + D_\Omega \mathbf{1}_{T < \tau_\epsilon} \\
&\le \delta + D_\Omega \mathbb{P}[\|\mathbf{z}\| > \varphi_0] + \frac{\psi}{S+1} + C\mathbb{E}[\|X_t - Y_t\|_2 \,|\mathcal{F}_T] + D_\Omega \mathbb{P}[\|X_t - Y_t\| \ge \eta_0 |\mathcal{F}_T] \\
&+ \delta + D_\Omega \mathbf{1}_{T < \tau_\epsilon} \\
&\le D_\Omega \mathbb{P}[\|\mathbf{z}\| > \varphi_0] + \frac{\psi}{S+1} + 2\delta + \left(C + \frac{D_\Omega}{\eta_0} \right) \mathbb{E}[\|X_t - Y_t\|_2 \,|\mathcal{F}_T] + D_\Omega \mathbf{1}_{T < \tau_\epsilon}.
\end{aligned}$$

Iterating this inequality over $T \le t < T+S$ and recalling that $\|X_T - Y_T\|_2 = 0$,

$$\mathbb{E}[\|X_{T+S} - Y_{T+S}\|_2 \,|\mathcal{F}_T] \le \left(2\delta + \frac{\psi}{S+1} + D_\Omega \mathbb{P}[\|\mathbf{z}\| > \varphi_0] \right) \left(C + \frac{D_\Omega}{\eta_0} \right)^{S+1} + D_\Omega \mathbf{1}_{T < \tau_\epsilon}. \tag{B.5}$$

Combining inequalities (B.4) and (B.5),

$$\begin{aligned}
W_2(X_{T+S}, \pi) &\le \mathbb{E}[\|X_{T+S} - Z_{T+S}\|_2] \\
&\le D_\Omega 2^{-\lfloor \frac{S}{\tau_{\text{mix}}} \rfloor} + \left(2\delta + \frac{\psi}{S+1} + D_\Omega \mathbb{P}[\|\mathbf{z}\| > \varphi_0] \right) \left(C + \frac{D_\Omega}{\eta_0} \right)^{S+1} + D_\Omega \mathbb{P}[T < \tau_\epsilon].
\end{aligned}$$

Letting ψ go to 0,

$$W_2(X_{T+S}, \pi) \le D_\Omega 2^{-\lfloor \frac{S}{\tau_{\text{mix}}} \rfloor} + (2\delta + D_\Omega \mathbb{P}[\|\mathbf{z}\| > \varphi_0]) \left(C + \frac{D_\Omega}{\eta_0} \right)^{S+1} + D_\Omega \mathbb{P}[T < \tau_\epsilon]. \tag{B.6}$$

For $\alpha \in \mathbb{N}$, define $\delta(\alpha) = \frac{1}{\alpha^2}$, $\varphi_0(\alpha) = \inf\{\varphi : \mathbb{P}[\|\mathbf{z}\| > \varphi] \le \alpha^{-2}\}$, $S(\alpha) = \lfloor \frac{-\log(\alpha)}{\log(C + \frac{D_\Omega}{\eta_0})} \rfloor - 1$, and $T(\alpha)' = \inf\{t : \mathbb{P}[t < \tau_{\epsilon(\delta(\alpha))}] \le \frac{1}{\alpha}\}$. It is easy to check that $\lim_{\alpha \to \infty} S(\alpha) = \lim_{\alpha \to \infty} T(\alpha)' = \infty$, and so for any sequence $T(\alpha) > T(\alpha)'$, inequality (B.6) implies

$$\lim_{\alpha \to \infty} W_2(X_{T(\alpha)+S(\alpha)}, \pi) \le \lim_{\alpha \to \infty} \left(D_\Omega 2^{-\lfloor \frac{S(\alpha)}{\tau_{\text{mix}}} \rfloor} + \frac{4 D_\Omega}{\alpha} \right) = 0.$$

Since this holds for *any* sequence $T(\alpha) > T(\alpha)'$, inequality (4.5) follows.[3] ∎

Proof of Theorem 3. It is enough to check that the conditions of Theorem 2 hold. Go through the elements of Definition 4.1 in order as follows:

1. To check that inequality (4.2) holds, fix $\delta > 0$. By results in [8],[4] there exists a constant $\epsilon_1 = \epsilon_1(\delta, \lambda) > 0$ so that for all $\epsilon < \epsilon_1$,

$$\sup_{\theta \in \Theta} |p_{\mathcal{S}}(\theta) - p(\theta|\mathbf{d})| < \frac{\delta}{2D_\Omega} \tag{B.7}$$

if $\mathcal{S}$ is an ϵ-cover and the points $\mathcal{B}(\theta, R)$ chosen in line 2 of Algorithm 4 are λ-poised. The same discussion in [8] implies that there exists a constant $\epsilon_2 = \epsilon_2(\delta, \lambda) > 0$ so that for all $\epsilon < \epsilon_2$,

$$\sup_{\theta \in \Theta} |M_{\mathcal{S}}(\theta) - M_\infty(\theta)|, \, |\nabla_\theta(\pi_{\mathcal{S}}) - \nabla_\theta(\pi_\infty)| < \delta \tag{B.8}$$

if $\mathcal{S}$ is an ϵ-cover and the points $\mathcal{B}(\theta, R)$ chosen in line 2 of Algorithm 4 are λ-poised. Since the smallest singular value of $M(\theta)$ is bounded below uniformly in θ, this implies that there exists a constant $\epsilon_3 = \epsilon_3(\delta, \lambda) > 0$ so that for all $\epsilon < \epsilon_3$ (see [21, Prop. 7]),

$$\sup_{\theta \in \Theta} W_2(q_{\mathcal{S}}(\theta, \cdot), q_\infty(\theta, \cdot)) < \frac{\delta}{2} \tag{B.9}$$

as long as $\mathcal{S}$ is an ϵ-cover and the points $\mathcal{B}(\theta, R)$ chosen in line 2 of Algorithm 4 are λ-poised.

Combining inequalities (B.7) and (B.9), we have for all $0 < \epsilon < \min(\epsilon_1, \epsilon_3)$ that

$$\begin{aligned} \sup_{\theta \in \Theta} W_2(K_{\mathcal{S}}(\theta, \cdot), K_\infty(\theta, \cdot)) &\le \sup_{\theta \in \Theta} W_2(q_{\mathcal{S}}(\theta, \cdot), q_\infty(\theta, \cdot)) + D_\Omega \sup_{\theta \in \Theta} |p_{\mathcal{S}}(\theta) - p(\theta|\mathbf{d})| \\ &\le \frac{\delta}{2} + \frac{\delta}{2} = \delta. \end{aligned}$$

This completes the proof of inequality (4.2).

2. By the assumption that the mass matrix $M(\theta)$ and likelihood $\ell(\theta|\mathbf{d}, \mathbf{f})$ are both C^∞ functions on Ω, and that the smallest singular value of M and the likelihood ℓ are both uniformly bounded away from zero, we have

$$\begin{aligned} \|q_\infty(\theta, \cdot) - q_\infty(\theta', \cdot)\|_{\mathrm{TV}} &= \left\| \mathcal{N}\left(\theta + \frac{\epsilon}{2} M(\theta) \nabla_\theta \log\left(\ell(\theta|\mathbf{d}, \mathbf{f}) p(\theta)\right), \epsilon M(\theta)\right) \right. \\ &\qquad \left. - \mathcal{N}\left(\theta' + \frac{\epsilon}{2} M(\theta') \nabla_\theta \log\left(\ell(\theta'|\mathbf{d}, \mathbf{f}) p(\theta')\right), \epsilon M(\theta')\right) \right\|_{\mathrm{TV}} \\ &\le \left\| \mathcal{N}\left(\theta + \frac{\epsilon}{2} M(\theta) \nabla_\theta \log\left(\ell(\theta|\mathbf{d}, \mathbf{f}) p(\theta)\right), \epsilon M(\theta)\right) \right. \end{aligned} \tag{B.10}$$

[3] Since the convergence to stationarity under the Wasserstein distance may not be monotone, this flexibility in the choice of $T(\alpha)$ is necessary to obtain the desired convergence result.

[4] The required result is a combination of Theorems 3.14 and 3.16, as discussed in the text after the proof of Theorem 3.16 in [8].

$$
\begin{aligned}
&\quad - \mathcal{N}\left(\theta' + \frac{\epsilon}{2}M(\theta')\nabla_\theta \log\left(\ell(\theta'|\mathbf{d},\mathbf{f})p(\theta')\right), \epsilon M(\theta)\right)\Big\|_{\mathrm{TV}} \\
&+ \Big\|\mathcal{N}\left(\theta' + \frac{\epsilon}{2}M(\theta')\nabla_\theta \log\left(\ell(\theta'|\mathbf{d},\mathbf{f})p(\theta')\right), \epsilon M(\theta)\right) \\
&\quad - \mathcal{N}\left(\theta' + \frac{\epsilon}{2}M(\theta')\nabla_\theta \log\left(\ell(\theta'|\mathbf{d},\mathbf{f})p(\theta')\right), \epsilon M(\theta')\right)\Big\|_{\mathrm{TV}} \\
&= O_c(\|\theta - \theta'\|),
\end{aligned}
$$

where the bound on the first term in the last line is standard, and the second term in the last line is bounded by an application of [28, Lem. 4.8]. By a similar calculation,

$$
|\alpha_\infty(\theta, z) - \alpha_\infty(\theta', z)| = O_c(\|\theta - \theta'\|). \tag{B.11}
$$

Inequalities (B.10) and (B.11) imply that

$$
\begin{aligned}
&\sup_{\theta,\theta'\in\Theta,\,\|\theta-\theta'\|<\eta} W_2(K_\infty(\theta,\cdot), K_\infty(\theta',\cdot)) \\
&\qquad \le D_\Omega \sup_{\theta,\theta'\in\Theta,\,\|\theta-\theta'\|<\eta} \|K_\infty(\theta,\cdot) - K_\infty(\theta',\cdot)\|_{\mathrm{TV}} \\
&\qquad \le D_\Omega\Bigg(\sup_{\theta,\theta'\in\Theta,\,\|\theta-\theta'\|<\eta} \|q_\infty(\theta,\cdot) - q_\infty(\theta',\cdot)\|_{\mathrm{TV}} \\
&\qquad\qquad + \sup_{\theta,\theta',z\in\Theta,\,\|\theta-\theta'\|<\eta} |\alpha_\infty(\theta,z) - \alpha_\infty(\theta',z)|\Bigg) \\
&\qquad = O(\|\theta - \theta'\|).
\end{aligned}
$$

This completes the proof of inequality (4.3).

3. Inequality (4.4) follows immediately from (B.7) and (B.8).
4. Assumption 1 in Definition 3.1 holds by our assumption that Θ is the d-dimensional hypercube.
5. Assumption 2 in Definition 3.1 has two parts. The first part, that $q(\theta,\cdot|\mathbf{f})$ has a C^∞ density that is bounded away from zero uniformly in $\theta, \mathbf{f}$, follows from the form of the mMALA proposal and the fact that the state space is compact. The second part, that $p(\cdot|\mathbf{d})$ has a C^∞ density that is bounded away from zero uniformly in θ, is an assumption of our theorem.
6. Assumptions 3–6 in Definition 3.1 are assumed in the statement of the theorem.

This completes the proof of the theorem. ∎

REFERENCES

[1] R. J. Adler, *The Geometry of Random Fields*, SIAM, Philadelphia, 2010, https://doi.org/10.1137/1.9780898718980.

[2] N. Bliznyuk, D. Ruppert, and C. A. Shoemaker, *Local derivative-free approximation of computationally expensive posterior densities*, J. Comput. Graph. Statist., 21 (2012), pp. 476–495.

[3] S. P. Brooks and G. O. Roberts, *Assessing convergence of Markov chain Monte Carlo algorithms*, Stat. Comput., 8 (1998), pp. 319–335.

[4] B. Calderhead, *A general construction for parallelizing Metropolis-Hastings algorithms*, Proc. Natl. Acad. Sci. USA, 111 (2014), pp. 17408–17413.

[5] O. Cappe, A. Guillin, J.-M. Marin, and C. P. Robert, *Population Monte Carlo*, J. Comput. Graph. Statist., 13 (2004), pp. 907–929.
[6] P. Chen and Ch. Schwab, *Sparse-grid, reduced-basis Bayesian inversion*, Comput. Methods Appl. Mech. Engrg., 297 (2015), pp. 84–115.
[7] J. A. Christen and C. Fox, *Markov chain Monte Carlo using an approximation*, J. Comput. Graph. Statist., 14 (2005), pp. 795–810.
[8] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*, SIAM, Philadelphia, 2009, https://doi.org/10.1137/1.9780898718768.
[9] 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.
[10] Core Writing Team, R. K. Pachauri, and L. A. Meyer, eds., *Climate Change* 2014: *Synthesis Report. Contribution of Working Groups* I, II *and* III *to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*, IPCC, Geneva, Switzerland, 2014.
[11] S. L. Cotter, M. Dashti, and A. M. Stuart, *Approximation of Bayesian inverse problems for PDEs*, SIAM J. Numer. Anal., 48 (2010), pp. 322–345, https://doi.org/10.1137/090770734.
[12] M. K. Cowles and B. P. Carlin, *Markov chain Monte Carlo convergence diagnostics: A comparative review*, J. Amer. Statist. Assoc., 91 (1996), pp. 883–904.
[13] R. V. Craiu, J. Rosenthal, and C. Yang, *Learn from thy neighbor: Parallel-chain and regional adaptive MCMC*, J. Amer. Statist. Assoc., 104 (2009), pp. 1454–1466.
[14] 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*, Water Resour. Res., 47 (2011), W10521.
[15] T. Cui, Y. M. Marzouk, and K. Willcox, *Scalable posterior approximations for large-scale Bayesian inverse problems via likelihood-informed parameter and state reduction*, J. Comput. Phys., 315 (2016), pp. 363–387.
[16] M. Dashti and A. M. Stuart, *Uncertainty quantification and weak approximation of an elliptic inverse problem*, SIAM J. Numer. Anal., 49 (2011), pp. 2524–2542, https://doi.org/10.1137/100814664.
[17] J. Dupuit, *Études théoriques et pratiques sur le mouvement des eaux dans les canaux découverts et a travers les terrains perméables*, Dunod, Paris, 1863.
[18] J. A. Fill and M. Huber, *The randomness recycler: A new technique for perfect sampling*, in Proceedings of the 41st Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 2000, pp. 503–511.
[19] P. Fretwell, H. D. Pritchard, D. G. Vaughan, J. L. Bamber, N. E. Barrand, R. Bell, C. Bianchi, R. G. Bingham, D. D. Blankenship, G. Casassa, et al., *Bedmap2: Improved ice bed, surface and thickness datasets for Antarctica*, The Cryosphere, 7 (2013), pp. 375–393.
[20] M. Girolami and B. Calderhead, *Riemann manifold Langevin and Hamiltonian Monte Carlo methods*, J. R. Stat. Soc. Ser. B Stat. Methodol., 73 (2011), pp. 123–214.
[21] C. R. Givens and R. M. Shortt, *A class of Wasserstein metrics for probability distributions*, Michigan Math. J., 31 (1984), pp. 231–240.
[22] P. J. Green, K. Łatuszyński, M. Pereyra, and C. P. Robert, *Bayesian computation: A summary of the current state, and samples backwards and forwards*, Stat. Comput., 25 (2015), pp. 835–862.
[23] H. Haario, E. Saksman, and J. Tamminen, *An adaptive Metropolis algorithm*, Bernoulli, 7 (2001), pp. 223–242.
[24] Intergovernmental Panel on Climate Change, *Climate Change* 2013*: The Physical Science Basis. Contribution of Working Group* I *to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*, Cambridge University Press, Cambridge, New York, 2013.
[25] V. R. Joseph, *Bayesian computation using design of experiments-based interpolation technique*, Technometrics, 54 (2012), pp. 209–225.
[26] J. P. Kaipio and E. Somersalo, *Statistical and Computational Inverse Problems*, Springer, New York, 2005.
[27] M. C. Kennedy and A. O'Hagan, *Bayesian calibration of computer models*, J. R. Stat. Soc. Ser. B Stat. Methodol., 63 (2001), pp. 425–464.
[28] B. Klartag, *A central limit theorem for convex sets*, Invent. Math., 168 (2007), pp. 91–131.

[29] J. Li and Y. M. Marzouk, *Adaptive construction of surrogates for the Bayesian solution of inverse problems*, SIAM J. Sci. Comput., 36 (2014), pp. A1163–A1186, https://doi.org/10.1137/130938189.
[30] J. Li, J. Paden, C. Leuschen, F. Rodriguez-Morales, R. D. Hale, E. J. Arnold, R. Crowe, D. Gomez-Garcia, and P. Gogineni, *High-altitude radar measurements of ice thickness over the Antarctic and Greenland ice sheets as a part of Operation IceBridge*, IEEE Trans. Geosci. Remote Sens., 51 (2013), pp. 742–754.
[31] D. R. MacAyeal, *Large-scale ice flow over a viscous basal sediment: Theory and application to ice stream B, Antarctica*, J. Geophys. Res. Solid Earth, 94 (1989), pp. 4071–4087.
[32] D. R. MacAyeal, *A tutorial on the use of control methods in ice-sheet modeling*, J. Glaciol., 39 (1993), pp. 91–98.
[33] D. R. MacAyeal, *EISMINT: Lessons in Ice-Sheet Modeling*, Department of Geophysical Sciences, University of Chicago, Chicago, IL, 1997.
[34] J. Martin, L. C. Wilcox, C. Burstedde, and O. Ghattas, *A stochastic Newton MCMC method for large-scale statistical inverse problems with application to seismic inversion*, SIAM J. Sci. Comput., 34 (2012), pp. 1460–1487, https://doi.org/10.1137/110845598.
[35] Y. Marzouk and D. Xiu, *A stochastic collocation approach to Bayesian inference in inverse problems*, Commun. Comput. Phys., 6 (2009), pp. 826–847.
[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.
[37] L. S. Matott, *Screening-level sensitivity analysis for the design of pump-and-treat systems*, Ground Water Monit. R., 32 (2012), pp. 66–80.
[38] N. Petra, J. Martin, G. Stadler, and O. Ghattas, *A computational framework for infinite-dimensional Bayesian inverse problems, Part* II: *Stochastic Newton MCMC with application to ice sheet flow inverse problems*, SIAM J. Sci. Comput., 36 (2014), pp. A1525–A1555, https://doi.org/10.1137/130934805.
[39] N. Petra, H. Zhu, G. Stadler, T. J. R. Hughes, and O. Ghattas, *An inexact Gauss–Newton method for inversion of basal sliding and rheology parameters in a nonlinear Stokes ice sheet model*, J. Glaciol., 58 (2012), pp. 889–903.
[40] J. G. Propp and D. B. Wilson, *Exact sampling with coupled Markov chains and applications to statistical mechanics*, Random Structures Algorithms, 9 (1996), pp. 223–252.
[41] C. E. Rasmussen, *Gaussian processes to speed up hybrid Monte Carlo for expensive Bayesian integrals*, in Bayesian Statistics 7, Oxford University Press, New York, 2003, pp. 651–660.
[42] G. O. Roberts and J. S. Rosenthal, *Optimal scaling of discrete approximations to Langevin diffusions*, J. R. Stat. Soc. Ser. B Stat. Methodol., 60 (1998), pp. 255–268.
[43] G. O. Roberts and J. S. Rosenthal, *One-shot coupling for certain stochastic recursive sequences*, Stochastic Process. Appl., 99 (2002), pp. 195–208.
[44] J. S. Rosenthal, *Parallel computing and Monte Carlo algorithms*, Far East J. Theor. Stat., 4 (2000), pp. 207–236.
[45] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, *Design and analysis of computer experiments*, Statist. Sci., 4 (1989), pp. 409–423.
[46] T. J. Santner, B. J. Williams, and W. I. Notz, *The Design and Analysis of Computer Experiments*, Springer, New York, 2003.
[47] B. E. Schutz, H. J. Zwally, C. A. Shuman, D. Hancock, and J. P. DiMarzio, *Overview of the ICESat mission*, Geophys. Res. Lett., 32 (2005), L21S01.
[48] D. Steinsaltz, *Locally contractive iterated function systems*, Ann. Probab., 27 (1999), pp. 1952–1979.
[49] C. Villani, *Optimal Transport: Old and New*, Grundlehren Math. Wiss. 338, Springer-Verlag, Berlin, 2009.
[50] U. Wolff, *Monte Carlo errors with less errors*, Comput. Phys. Comm., 156 (2004), pp. 143–153.