Journal of Computational Physics (2021)

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Randomized approaches to accelerate MCMC algorithms for Bayesian Inverse Problems

Arvind K. Saibaba[a,*], Pranjal Prasad[b], Eric de Sturler[b], Eric Miller[c], Misha E. Kilmer[d]

[a]*Department of Mathematics, North Carolina State University*
[b]*Department of Mathematics, Virginia Tech University*
[c]*Department of Electrical and Computer Engineering, Tufts University*
[d]*Department of Mathematics, Tufts University*

ARTICLE INFO

Article history:

ABSTRACT

Markov chain Monte Carlo (MCMC) approaches are traditionally used for uncertainty quantification in inverse problems where the physics of the underlying sensor modality is described by a partial differential equation (PDE). However, the use of MCMC algorithms is prohibitively expensive in applications where each log-likelihood evaluation may require hundreds to thousands of PDE solves corresponding to multiple sensors; i.e., spatially distributed sources and receivers perhaps operating at different frequencies or wavelengths depending on the precise application. We show how to mitigate the computational cost of each log-likelihood evaluation by using several randomized techniques and embed these randomized approximations within MCMC algorithms. These MCMC algorithms are computationally efficient methods for quantifying the uncertainty associated with the reconstructed parameters. We demonstrate the accuracy and computational benefits of our proposed algorithms on a model application from diffuse optical tomography where we invert for the spatial distribution of optical absorption.

1. Introduction

Inverse problems arise in many imaging applications; for example, to visualize the subsurface of the earth in geophysical applications, to visualize the internal structure of human or animal bodies in biomedical applications, and to detect materials of interest such as explosive or illegal narcotics in luggage scanning and other security applications. Across all these problems, the goal is to generate detailed reconstructions of the spatial distribution of one or more material properties. Inverse problems share a common structure: the availability of limited measurements collected

*Corresponding author: Tel.: +1-(919)-513-2299; Email: asaibab@ncsu.edu

on the boundary or outside of the medium and a physical, or forward, model mapping the unknown properties of the medium onto the data. Given the (often nonlinear) forward model that involves partial differential equations (PDEs), noisy data, and available computing resources, there is a need for efficient algorithms for solving the inverse problem and quantifying the uncertainty associated with the reconstructed parameters.

To perform uncertainty quantification (UQ), we adopt the Bayesian approach which combines the likelihood of the data given the parameters and the prior distribution to produce the posterior distribution. In the applications we consider, the posterior is non-Gaussian and the prevalent approach is to use Markov chain Monte Carlo (MCMC) techniques to produce samples from this distribution. These samples can then be used to estimate the conditional mean, conditional covariance, and other summary statistics associated with the posterior distribution. There are several computational challenges associated with inverse problems and UQ: modern imaging devices have the ability to collect a large volume of data using sensors, and the underlying physics has to be resolved on finely discretized grids. A dominant cost associated with a typical MCMC algorithm is the need to repeatedly evaluate the log-likelihood where each evaluation requires hundreds to thousands of expensive PDE solves. This cost can be prohibitive in the context of our applications and, therefore, efficient methods are necessary. In this work, we focus on randomized techniques to reduce the computational cost associated with the MCMC algorithm. Alternative techniques such as multilevel algorithms, model reduction, active subspaces, etc., can also be used; a survey of these techniques can be found in [45].

In recent years, there have been several approaches that use randomization to accelerate the computational cost associated with inverse problems. One line of inquiry employs randomization as a computation tool for reducing the effective number of partial differential equations (PDEs) to be solved when optimization-based techniques are use to solve inverse problems [31, 30, 49, 34, 7, 6]. These methods use the Monte Carlo trace estimator to approximate the objective function and the gradient to reformulate the optimization problem in the stochastic optimization framework. Other papers have used randomized techniques for computing an efficient representation of the posterior covariance matrix as a low-rank perturbation of the prior covariance matrix. This efficient representation can then be used for uncertainty quantification [52, 13]. However, these approximations have limitations for non-Gaussian posterior distributions, which we consider in this paper. Recent work in [37] developed theoretical results for bounding the distance between the exact and the approximate posterior distributions, when randomized methods are used to approximate the posterior distributions. However, there are very few works that use randomization in the context of MCMC algorithms, which is the focus of the present work. The use of Monte Carlo approximations to reduce the computational cost of evaluating the log-likelihood has been explored in [4, 9, 44, 14]. Randomization techniques were also used for efficient sampling in hierarchical Bayesian inverse problems [12, 51]. Our current work differs both in the kind of randomized techniques used and in how they are employed in the context of the MCMC algorithms.

An approach that is alternative to the randomized approaches proposed here is to use surrogate models to mitigate the cost of the log-likelihood evaluations. Here we only briefly mention a few approaches such as polynomial chaos [41, 57], local polynomial approximations [17], Gaussian process surrogates [55], reduced order models [20], sparse grid interpolation [15], active subspaces [18], neural network based surrogates [58], etc. An important point

here is that the accuracy of the surrogate approaches and the resulting speedups obtained depend to an extent on the choice of the parametization and the dimensionality of the parameter spaces, which is in contrast to the randomization techniques proposed here.

Overview of contributions and contents. This paper develops efficient MCMC algorithms, powered by randomized approximations, for quantifying the uncertainty in the reconstruction of the unknown parameters. The cost of the MCMC algorithms is dominated (in terms of the number of PDE solves) by repeated expensive evaluations of the log-likelihood. In Section 2, we explain the computational costs in terms of a model ill-posed problem, Diffuse Optical Tomography, for which the forward model is described by a Helmholtz equation with a purely imaginary wavenumber. Motivated by this fact, our first major contribution (Section 4) is developing several computationally efficient randomized approximations to the log-likelihood. The first approach uses Monte Carlo approximations to the log-likelihood and are *geometry independent* in that they do not depend on the geometric distribution of the sources and receivers. The second set of methods proposed are *geometry dependent* in that they exploit the specific geometric distribution of the sources and receivers. The geometry dependent methods are only applicable to a class of imaging problems arising from elliptic PDEs (e.g., diffuse optical tomography, hydraulic tomography, electrical resistivity tomography); however, the geometry independent methods are applicable across a broad class of imaging applications. Second, we use the randomized algorithms for approximating the log-likelihood to accelerate two well-known MCMC algorithms—standard Metropolis Hastings and a two-stage approach based on delayed acceptance (Section 3). Using the standard Metropolis Hastings with randomized approximations results in efficient but approximate Bayesian inference but using the two-stage approach we can draw samples from the posterior distribution with the correct statistics. Third, we demonstrate the benefits of our proposed approaches on a synthetic problem from Diffuse Optical Tomography (Section 5). The unknown parameters characterized the absorption field, which is assumed to be piecewise constant and is represented efficiently using the parametric level set approach. The numerical experiments give insight into the accuracy and the computational costs of the randomized algorithms and the performance of the MCMC methods accelerated by randomization.

2. Motivating application: Diffuse Optical Tomography

In this section, we consider the basic setup of Diffuse Optical Tomography (DOT). We describe the forward problem, the parameterization of the inverse problem and the computational challenges involved in solving the inverse problem. While DOT is a prototypical application that we consider, our approach is more general and is applicable to a wide-range of inverse problems such as electrical resistivity tomography, electrical impedance tomography, and hydraulic tomography.

2.1. Forward problem

In DOT, the medium of interest is illuminated by several light sources in the near-infrared portion of the electromagnetic spectrum (600nm-1000nm). Measurement sensors placed on the exterior of the medium measure the optical

fluence and this information is used to invert for spatial maps of optical quantities such as diffusion and absorption coefficients. For a large class of very practical problems, the radiative transport physics associated with the propagation of light through a medium (represented by the domain Ω) can be approximated by the diffusion model [5]

$$-\nabla \cdot D(\mathbf{x})\nabla\phi(\mathbf{x}) + \nu\mu_a(\mathbf{x})\phi(\mathbf{x}) = S_0\delta(\mathbf{x}-\mathbf{x}_j) \qquad \mathbf{x} \in \Omega, \tag{1}$$

$$\phi(\mathbf{x}) + 2A_n D(\mathbf{x})\frac{\partial\phi(\mathbf{x})}{\partial n} = 0 \qquad \mathbf{x} \in \partial\Omega, \tag{2}$$

where $D(\mathbf{x})$ is the diffusion coefficient. The diffusion coefficient is related to the reduced scattering coefficient μ_s' through the relation $D(\mathbf{x}) = \nu/3\mu_s'$; here we assume that the scattering coefficient, and therefore diffusion, is constant throughout the domain. The coefficient A_n depends on the refractive index of the interface of the two media. We denote by $\phi(\mathbf{x})$ the photon fluence, and ν is the electromagnetic propagation velocity within the medium. The quantity S_0 is the source power strength at wavelength and $\mathbf{x}_j$ is the position of source location of source numbered j. Following [53], we use the collimated source approach and use sources in the PDE term rather than the boundary condition. Following [52] we take $S_0 = 1$. Furthermore, $\mu_a(\mathbf{x})$ is the absorption coefficient, which is considered unknown in our problem, and we seek to recover this spatially dependent quantity.

To solve the systems of equations Eq. (1) along with boundary conditions Eq. (2), we use the standard linear Galerkin finite element approach. We denote the discretized photon fluence $\boldsymbol{\phi}$. The resulting system of equations can be summarized as

$$(\mathbf{K} + \mathbf{M}(\mathbf{p}))\,\boldsymbol{\phi}_j = \mathbf{s}_j, \qquad j = 1, \ldots, N_s, \tag{3}$$

where N_s is the number of sources, matrices $\mathbf{K}$ and $\mathbf{M}(\cdot)$ represent the discretized diffusion and absorption terms, and $\mathbf{p}$ represents the parameters to be recovered. The specific choice of parameterization will be explained in Section 2.2.

Data acquisition. We explain the data used for solving the inverse problem. Let $\mathbf{S} = \begin{bmatrix}\mathbf{s}_1 & \ldots & \mathbf{s}_{N_s}\end{bmatrix}$ denote the matrix corresponding to the discretized source terms. Similarly, denote the discretized delta functions $\mathbf{r}_j$ with the centers at the receiver locations $\mathbf{y}_j$ for $j = 1, \ldots, N_r$ and collected in the receiver matrix $\mathbf{R} = \begin{bmatrix}\mathbf{r}_1 & \ldots & \mathbf{r}_{N_r}\end{bmatrix}$. Then the measurement equation takes the form

$$\mathbf{D} = \mathbf{X}(\mathbf{p}) + \mathbf{N} \qquad \mathbf{N}_{ij} \sim \mathcal{N}(0, \sigma^2), \tag{4}$$

where σ is the standard deviation of the measurement noise, $\mathbf{D} \in \mathbb{R}^{N_r \times N_s}$ is the observed data and the forward model can be represented as

$$\mathbf{X}(\mathbf{p}) \equiv \mathbf{R}^\top \mathbf{A}(\mathbf{p})^{-1}\mathbf{S}, \tag{5}$$

where $\mathbf{A}(\mathbf{p}) = \mathbf{K} + \mathbf{M}(\mathbf{p})$.

In practice, the measurements from all the source-receiver pairs are not collected because the data collection process is either limited by physical or economical constraints or because the process is laborious and time consuming. To model this partial observation case, we denote by $\mathbf{E} \in \mathbb{R}^{N_r \times N_s}$, the matrix (with entries in $\{0, 1\}$) that encodes which source-receiver pairs are active. The matrix $\mathbf{E}$ has a nonzero entry if a receiver located at $\mathbf{y}_i$ records information due

to a source located at $\mathbf{x}_j$. Therefore, using our notation, the measurements that are actually collected are represented by $\mathbf{E} \odot \mathbf{D}$, where $\odot$ is the elementwise, or Hadamard, product. We briefly comment on the various possibilities for the entries of $\mathbf{E}$. If the matrix $\mathbf{E}$ is dense, then all the source-receiver pairs are active, i.e., all the receivers record information from all the sources. In most applications, $\mathbf{E}$ is a sparse matrix; for example, in our previous work [52], $\mathbf{E}$ is a sparse matrix with 4 nonzero entries per row, corresponding to four receivers per source location.

2.2. Inverse Problem

We are interested in recovering the absorption coefficient which is spatially varying and is represented by the parameters $\mathbf{p} \in \mathbb{R}^p$. In this paper, we adopt the parametric level set approach [2] (PaLS) to represent the absorption coefficient as a piecewise constant function. The absorption field is represented as

$$\mu_a(\mathbf{x}) = \mu_a^p \chi_{\mathcal{D}}(\mathbf{x}) + \mu_a^b (1 - \chi_{\mathcal{D}}(\mathbf{x})),$$

where μ_a^p and μ_a^b represents the absorption coefficients of the perturbation and the background respectively, and $\chi_{\mathcal{D}}(\mathbf{x})$ is a characteristic function that takes the value 1 when $\mathbf{x} \in \mathcal{D}$ and 0 when $\mathbf{x} \in \Omega \backslash \mathcal{D}$. In the PaLS approach, we express the characteristic function $\chi_{\mathcal{D}}(\mathbf{x})$ as the τ-level set of a Lipschitz continuous function $\varphi : \mathcal{D} \to \mathbb{R}$ as follows

$$\chi_{\mathcal{D}}(\mathbf{x}) = H(\varphi(\mathbf{x}) - \tau) \qquad \varphi(\mathbf{x}) = \sum_{k=1}^{n_p} \alpha_k \psi\left(\beta_k \|\mathbf{x} - \chi_k\|^{\dagger}\right), \tag{6}$$

where $H(\cdot)$ is the Heaviside function. In practice, we use a smooth approximation H_ϵ of the Heaviside function $H(\cdot)$; see [2, Equation (5.12)]. We represent the function $\varphi(\mathbf{x})$ as weighted combinations of basis functions $\psi(\cdot)$ and we have $\|\mathbf{x}\|^{\dagger} = \sqrt{\|\mathbf{x}\|_2^2 + \zeta^2}$ and $\zeta > 0$ is a small parameter chosen in order to ensure that φ is differentiable. We choose $\psi(\cdot)$ as compactly supported radial basis functions; however, other choices include polynomials and general radial basis functions. The coefficients $\{\alpha_k\}$ control the magnitude of the radial basis functions, $\{\beta_k\}$ control the width, and $\{\chi_k\}$ control the centers. While having a large number of basis functions will be beneficial in representing and reconstructing fine scale features, it increases the number of the parameters to be estimated. Additional details of the implementation are given in Section 5.

Posterior distribution. We adopt the Bayesian approach for solving the inverse problem. From (4), since the measurement noise has a Gaussian distribution, we can write the data-misfit part of the likelihood as

$$\pi(\mathbf{d} \mid \mathbf{p}) \propto \exp\left(-\Phi(\mathbf{p})\right) \qquad \Phi(\mathbf{p}) \equiv \frac{1}{2\sigma^2} \|\mathbf{E} \odot (\mathbf{D} - \mathbf{X}(\mathbf{p}))\|_F^2, \tag{7}$$

where $\mathbf{d} = \mathrm{vec}(\mathbf{D})$ and $\propto$ denotes 'proportional to.' The Frobenius norm of a matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$ is defined as $\|\mathbf{M}\|_F^2 = \mathrm{trace}(\mathbf{M}^\top \mathbf{M}) = \sum_{i=1}^m \sum_{j=1}^n |m_{ij}|^2$. The symbol $\Phi(\mathbf{p})$ is sometimes called the data-misfit functional, and quantifies the misfit between the measured data $\mathbf{D}$ and the forward model $\mathbf{X}(\mathbf{p})$. The prior information regarding the parameters of interest is encapsulated in the prior distribution $\pi(\mathbf{p})$. Using Bayes' rule, we can combine the prior information and the data-misfit likelihood, to produce the posterior distribution

$$\pi(\mathbf{p}|\mathbf{d}) \propto \pi(\mathbf{d}|\mathbf{p})\pi(\mathbf{p}). \tag{8}$$

Note that the proportionality constant which depends only on the data $\mathbf{d}$ can be omitted since it does not affect the inference of the unknown parameters $\mathbf{p}$. Since the forward operator is nonlinear, the posterior distribution is non-Gaussian. To quantify the uncertainty in the reconstructed parameters, we use samples from the posterior distribution. For example, suppose we want to compute $\mathbb{E}\left[h(\mathbf{p})\right]$, where $h(\cdot)$ is a quantity of interest and the expectation is computed with respect to the posterior distribution. Given samples from the posterior distribution $\{\mathbf{p}_j\}$, and under suitable assumptions on $h(\cdot)$ and the samples, the Monte Carlo process converges as

$$\lim_{N\to\infty} \frac{1}{N} \sum_{j=1}^{N} h(\mathbf{p}_j) = \int h(\mathbf{p})\pi(\mathbf{p}|\mathbf{d})d\mathbf{p}, \tag{9}$$

almost surely. In order to generate samples from the posterior distribution, we adopt the Markov Chain Monte Carlo (MCMC) approach.

Computational cost. A typical MCMC algorithm such as the Metropolis-Hastings algorithm requires repeated evaluation of the log-likelihood, which is dominated by the cost of computing $\mathbf{X}(\mathbf{p})$. One approach to compute $\mathbf{X}(\mathbf{p})$ is to first compute N_s forward PDE solves $\mathbf{U}(\mathbf{p}) = \mathbf{A}(\mathbf{p})^{-1}\mathbf{S}$ and then compute $\mathbf{X}(\mathbf{p}) = \mathbf{R}^\top\mathbf{U}(\mathbf{p})$. Alternatively, if the number of receivers N_r is smaller than the number of source N_s, it may be computationally beneficial to first compute $\mathbf{V}(\mathbf{p}) = \mathbf{A}(\mathbf{p})^{-\top}\mathbf{R}$ and then compute $\mathbf{X}(\mathbf{p}) = \mathbf{V}(\mathbf{p})^\top\mathbf{S}$, which requires N_r adjoint PDE solves. Note that for DOT the forward operator is self-adjoint, but our present discussion of computational costs includes the more general case. Therefore, at each iteration of a typical MCMC approach, the dominant cost involves $\min\{N_s, N_r\}$ PDE solves. This computational cost can be overwhelmingly large for two reasons. On finely discretized grids, the number of degrees of freedom can be large, therefore, each PDE solve can be computationally expensive. Furthermore, for DOT and other related applications, the number of sources and receivers can range from hundreds to thousands. A typical MCMC solver can run for $10^5 - 10^6$ iterations; therefore, assuming the upper end of the estimates given above, we need 10^9 PDE solves. In this paper, we use several randomization methods to reduce this computational cost.

3. MCMC algorithms

To quantify the uncertainty associated with the reconstructions, we need samples from the posterior distribution $\pi(\mathbf{p}|\mathbf{d})$; the prevalent approach is to use Markov Chain Monte Carlo (MCMC) methods. We review the standard Metropolis-Hastings algorithm and the two-stage MCMC algorithm in Sections 3.1 and 3.2 respectively, and discuss the computational efficiency in Section 3.3.

3.1. Metropolis-Hastings approach

The main idea is to generate a Markov chain with $\pi(\mathbf{p}|\mathbf{d})$ as its stationary distribution. A key step to this approach is to construct the desired transition kernel for the Markov chain. A popular choice of transition kernel is the Metropolis-Hastings (MH) approach which is described in Algorithm 1.

Starting with an arbitrary initial vector $\mathbf{p}_0$, the MCMC algorithm is run for N_{chain} steps to generate the Markov chain $\{\mathbf{p}_k\}_{k=N_b+1}^{N_{\text{chain}}}$, where N_b is the number of samples discarded in the burn-in stage. At each iteration, the probability

Algorithm 1 Metropolis-Hastings algorithm

Require: A proposal distribution $q(\cdot|\cdot)$, maximum iterations $N_{\text{chain}} \geq 1$, burn-in period $1 \leq N_b < N_{\text{chain}}$.

1: **for** $k = 1, \ldots, N_{\text{chain}}$ **do**
2: Generate $\mathbf{p}_*$ from $q(\cdot|\mathbf{p}_k)$.
3: Evaluate the acceptance ratio

$$\alpha(\mathbf{p}_*, \mathbf{p}_k) = \min\left\{1, \frac{\pi(\mathbf{p}_*|\mathbf{d})q(\mathbf{p}_k|\mathbf{p}_*)}{\pi(\mathbf{p}_k|\mathbf{d})q(\mathbf{p}_*|\mathbf{p}_k)}\right\}.$$

4: Generate a random number $u \sim \mathcal{U}[0, 1]$. If $u \leq \alpha(\mathbf{p}_*, \mathbf{p}_k)$, accept this proposed iterate so that $\mathbf{p}_{k+1} = \mathbf{p}_*$; else set $\mathbf{p}_{k+1} = \mathbf{p}_k$.
5: **end for**
6: **return** The chain $\{\mathbf{p}_k\}_{k=N_b+1}^{N_{\text{chain}}}$.

of moving from state $\mathbf{p}_k$ to a next state $\mathbf{p}_{k+1}$ is $q(\mathbf{p}_*|\mathbf{p}_k)\alpha(\mathbf{p}_*, \mathbf{p}_k)$, so the transition kernel for the Markov chain is

$$K(\mathbf{p}_k, \mathbf{p}_*) = q(\mathbf{p}_*|\mathbf{p}_k)\alpha(\mathbf{p}_*, \mathbf{p}_k) + (1 - r(\mathbf{p}_k))\delta_{\mathbf{p}_k}(\mathbf{p}_*),$$

where $r(\mathbf{p}_k) = \int q(\mathbf{p}|\mathbf{p}_k)\alpha(\mathbf{p}, \mathbf{p}_k)d\mathbf{p}$. The chain obtained after discarding the iterates in the burn-in stage has the same stationary distribution as the posterior distribution $\pi(\mathbf{p}|\mathbf{d})$, and therefore can be used as empirical Monte Carlo estimators similar to (9), see [47, Section 7.3].

As argued earlier, the standard MH algorithm is computationally expensive because each log-likelihood evaluation is expensive, and several such evaluations need to be performed. To tackle this computational challenge, Christen and Fox [16] proposed a two-stage MCMC algorithm to address the computational cost of sampling from the posterior distribution. We briefly review this approach and discuss its computational efficiency.

3.2. Two-stage MCMC algorithm

Assume that we have a surrogate approximation for the true posterior distribution $\pi(\mathbf{p}|\mathbf{d})$ that is accurate and is cheap to evaluate. Denote the iterate of the MCMC chain at the current step as $\mathbf{p}_k$; the approximation of the likelihood at the current step $\mathbf{p}_k$ is denoted by $\widehat{\pi}_{\mathbf{p}_k}(\mathbf{p}|\mathbf{d})$. At each step of the algorithm, a proposed sample $\mathbf{p}_*$ is generated from $q(\cdot|\mathbf{p}_k)$; then subjected to an accept-reject step based on the approximate likelihood $\widehat{\pi}_{\mathbf{p}_k}(\mathbf{p}|\mathbf{d})$. This results in a new proposal distribution

$$q^*(\mathbf{p}_*|\mathbf{p}_k) = q(\mathbf{p}_*|\mathbf{p}_k)\alpha(\mathbf{p}_*, \mathbf{p}_k) + (1 - r(\mathbf{p}_k))\delta_{\mathbf{p}_k}(\mathbf{p}_*), \tag{10}$$

where $r(\mathbf{p}_k) = \int q(\mathbf{p}|\mathbf{p}_k)\alpha(\mathbf{p}, \mathbf{p}_k)d\mathbf{p}$, and $\delta_{\mathbf{p}}(\cdot)$ is the Dirac mass at $\mathbf{p}$. If the proposed sample is accepted then it is promoted to the next stage, where it is subject to a second round of accept-reject based on the full posterior distribution $\pi(\mathbf{p}|\mathbf{d})$. If the proposed sample is, instead, rejected at the first stage, the second stage is skipped entirely and we set $\mathbf{p}_{k+1} = \mathbf{p}_k$. The details of this algorithm are given in Algorithm 2.

The potential reduction in the computational cost in the two-stage approach occurs because the full, expensive, likelihood $\pi(\mathbf{p}|\mathbf{d})$ is only evaluated when the proposed sample promoted to the second stage is "good," in the sense that is likely to be accepted at the second stage. Note that in the two-stage approach, although $r(\cdot)$ appears in the definition of $q^*(\cdot|\cdot)$ there is no need to evaluate $r(\cdot)$. To see this, consider the following two cases. If $\mathbf{p}_* = \mathbf{p}_k$, then

$\rho(\mathbf{p}_*, \mathbf{p}_k) = 1$ and the algorithm does not enter the second stage. On the other hand, if the proposed sample is promoted to the second stage, then

$$q^*(\mathbf{p}_*|\mathbf{p}_k) = q(\mathbf{p}_*|\mathbf{p}_k)\alpha(\mathbf{p}_*, \mathbf{p}_k).$$

As in the case of the standard MH algorithm, Algorithm 2 is run for N_{chain} steps to generate the Markov chain $\{\mathbf{p}_k\}_{k=N_b+1}^{N_{\text{chain}}}$, where N_b is the number of samples discarded in the burn-in stage. Under the assumptions of [16, Theorem 1], which we assume to hold for our problem as well, the transition kernel is irreducible and strongly aperiodic. With this assumption, Christen and Fox argue that standard ergodic results can be used to prove that Algorithm 2 produces samples from the posterior $\pi(\mathbf{p}|\mathbf{d})$ (see Section 1 of [16], in particular Theorem 1 and the surrounding discussion).

Algorithm 2 Two-stage MCMC algorithm

Require: A proposal distribution $q(\cdot|\cdot)$, maximum iterations $N_{\text{chain}} \geq 1$, burn-in period $1 \leq N_b < N_{\text{chain}}$.

1: **for** $k = 1, \ldots, N_{\text{chain}}$ **do**
2: Generate $\mathbf{p}_*$ from $q(\cdot|\mathbf{p}_k)$.
3: Evaluate the acceptance ratio

$$\rho_1(\mathbf{p}_*, \mathbf{p}_k) = \min\left\{1, \frac{\widehat{\pi}_{\mathbf{p}_k}(\mathbf{p}_*|\mathbf{d})q(\mathbf{p}_k|\mathbf{p}_*)}{\widehat{\pi}_{\mathbf{p}_k}(\mathbf{p}_k|\mathbf{d})q(\mathbf{p}_*|\mathbf{p}_k)}\right\}.$$

4: Generate a random number $u \sim \mathcal{U}[0, 1]$. If $u < \rho_1(\mathbf{p}_*, \mathbf{p}_k)$ promote sample $\mathbf{p}_*$ to the next stage; else use $\mathbf{p}_* = \mathbf{p}_k$.
5: Evaluate the acceptance ratio

$$\rho_2(\mathbf{p}_*, \mathbf{p}_k) = \min\left\{1, \frac{\pi(\mathbf{p}_*|\mathbf{d})q^*(\mathbf{p}_k|\mathbf{p}_*)}{\pi(\mathbf{p}_k|\mathbf{d})q^*(\mathbf{p}_*|\mathbf{p}_k)}\right\},$$

where $q^*(\mathbf{p}_*|\mathbf{p}_k)$, is defined in (10).
6: Generate a random number $u \sim \mathcal{U}[0, 1]$. If $u < \rho_2(\mathbf{p}_*, \mathbf{p}_k)$, accept this sample so that $\mathbf{p}_{k+1} = \mathbf{p}_*$; else set $\mathbf{p}_{k+1} = \mathbf{p}_k$.
7: **end for**
8: **return** The chain $\{\mathbf{p}_k\}_{k=N_b+1}^{N_{\text{chain}}}$.

Note that if a state-independent approximation is used, that is $\widehat{\pi}_{\mathbf{p}}(\mathbf{p}|\mathbf{d}) = \widehat{\pi}(\mathbf{p}|\mathbf{d})$, then this algorithm is related to the surrogate transition algorithm proposed in [37, Section 9.4.3]. We provide a simplified representation of the acceptance ratio that will be useful in analyzing the computational efficiency of the proposed sampler. At iteration k, the acceptance ratio at the second stage is

$$\rho_2(\mathbf{p}_*, \mathbf{p}_k) = \min\left\{1, \frac{\pi(\mathbf{p}_*|\mathbf{d})\widehat{\pi}(\mathbf{p}_k|\mathbf{d})}{\pi(\mathbf{p}_k|\mathbf{d})\widehat{\pi}(\mathbf{p}_*|\mathbf{d})}\right\}.$$

The proof is available in [20]. From this equation it is clear that the acceptance rate at the second stage is high when $\widehat{\pi}(\mathbf{p}|\mathbf{d})$ is a good approximation $\pi(\mathbf{p}|\mathbf{d})$. In the extreme case that $\widehat{\pi}(\mathbf{p}|\mathbf{d}) = \pi(\mathbf{p}|\mathbf{d})$, the acceptance at the second stage is 1, implying that all the promoted iterates are accepted. If a state-dependent approximation is used, then the acceptance ratio at the second stage takes the form [33, Equation (A6)]

$$\rho_2(\mathbf{p}_*, \mathbf{p}_k) = \min\left\{1, \frac{\pi(\mathbf{p}_*|\mathbf{d}) \min\left\{1, \frac{\pi_{\mathbf{p}_*}(\mathbf{p}_k|\mathbf{d})}{\pi_{\mathbf{p}_*}(\mathbf{p}_*|\mathbf{d})}\right\}}{\pi(\mathbf{p}_k|\mathbf{d}) \min\left\{1, \frac{\pi_{\mathbf{p}_k}(\mathbf{p}_*|\mathbf{d})}{\pi_{\mathbf{p}_k}(\mathbf{p}_k|\mathbf{d})}\right\}}\right\}.$$

Here, for simplicity, we have assumed that the proposal distribution $q(\cdot|\cdot)$ is symmetric in its arguments, which is the case in the proposal distributions we use in the numerical experiments.

3.3. Computational efficiency and choice of proposal distributions

We review the statistical and computational efficiency of Algorithm 2. Let t_f be the computational time for the exact likelihood and let t_1 and t_2 denote the computational cost of the first and the second stages of the two-stage algorithm. In the first stage of the algorithm, the only cost is in evaluating the approximate likelihood, but if an iterate is accepted at the second stage, then it entails a full likelihood evaluation as well as two approximate likelihood evaluations in computing the reverse transition $q^*(\mathbf{p}_k|\mathbf{p}_*)$. Let $\eta \in [0, 1]$ be the average acceptance rate of the first stage. The cost of the two-stage approach is $N_{\text{chain}}(\eta t_2 + t_1)$, whereas the cost of the standard MH approach is $t_f N_{\text{chain}}$. Then the speedup of the two-stage MCMC approach is

$$\text{Speedup}_{\text{comp}} = \frac{t_f}{t_1 + \eta t_2}.$$

The speedup is maximal when both $\eta, t_1/t_f \ll 1$. That is the acceptance rate at the first stage, and the ratio of the inexpensive to the expensive model is as close to zero as possible. However, this formula does not account for the quality of the samples generated.

Let $\tau \geq 1$ be the integrated autocorrelation time (IACT) for the standard MH algorithm. It is computed as

$$\tau \equiv 1 + 2\sum_{k=1}^{\infty} \text{corr}(q(\mathbf{p}_0), q(\mathbf{p}_k)),$$

where $q(\cdot)$ is a quantity of interest and $\{\mathbf{p}_k\}$ is the MCMC chain. Similarly, let $\tau_t \geq 1$ be the IACT for the two-stage approach. The effective sample size measures the number of statistically independent samples and is computed as $\text{CES} = N_{\text{chain}}/\tau$. Accounting for the number of statistically independent samples, the speedup is obtained by replacing N_{chain} with CES to obtain

$$\text{Speedup}_{\text{stat}} = \frac{\tau_t}{\tau_f}\frac{t_f}{t_1 + \eta t_2},$$

where τ_f is the IACT for the exact likelihood approach. The computational efficiency comes from the fact that the exact likelihood is only evaluated when a good candidate sample is available; the quality of the sample then depends on the accuracy of how well the approximate distribution $\widehat{\pi}_{\mathbf{p}_k}(\mathbf{p}|\mathbf{d})$ matches the target distribution $\pi(\mathbf{p}|\mathbf{d})$. Therefore, there is a trade-off between computational efficiency on the one hand, and statistical efficiency (measured as CES) on the other.

We briefly mention the various choices of the proposal distribution $q(\cdot|\mathbf{p}_k)$ that we employed. A standard choice uses the random-walk proposal $\mathcal{N}(\mathbf{p}_k, \sigma_d^2\mathbf{I})$, where $\sigma_d^2 = 0.1^2/d$ and d is the number of unknown parameters. However, in some cases, we did not observe the acceptance rates that satisfied the rule of thumb $30 - 40\%$. A simple variant of this algorithm adaptively scales the random-walk variance in the first 20% of the iterations until the acceptance rate was within the desired rule of thumb [56, Algorithm 2]. The second adaptive approach that we use is the Adaptive Metropolis approach for generating proposal distribution. In this approach, the proposal distribution $q(\cdot|\mathbf{p}_0, \ldots, \mathbf{p}_{k-1})$ is Gaussian with mean $\mathbf{p}_k$ and the covariance

$$\mathbf{\Sigma}_k = \begin{cases} \mathbf{\Sigma}_0 & k \leq k_0 \\ s_d\left(\widetilde{\mathbf{\Sigma}}_k + \epsilon\mathbf{I}_d\right) & k > k_0 \end{cases},$$

where $\mathbf{\Sigma}_0$ is a fixed covariance matrix for the first k_0 iterations, $\widetilde{\mathbf{\Sigma}}_k$ is the sample covariance matrix, $s_d = 2.38^2/d$ and $\epsilon > 0$ is a small number of ensure that $\mathbf{\Sigma}_k$ is positive definite [29]. Although the resulting chains in both the adaptations are non-Markovian, they satisfy the so-called diminishing adaptation condition which ensures that the resulting chain has the correct ergodic properties and samples from the target distribution [11, Chapter 4].

4. Randomized strategies for efficient likelihood evaluation

In this section, we discuss several different strategies for efficiently evaluating the likelihood. The methods are split into two different cases: geometry-independent ('MCTrace') and geometry-dependent ('RandLR', 'RandTrace', and 'RandHODLR'). For simplicity, we first consider the case $\mathbf{E} = \mathbf{1}$, where $\mathbf{1} \in \mathbb{R}^{N_s \times N_r}$ is the matrix of ones. The case $\mathbf{E} \neq \mathbf{1}$ will be discussed in Section 4.3. Finally, in Section 4.4, we discuss ways of incorporating the randomized approximations for the log-likelihood into the MCMC algorithms discussed in Section 3.

4.1. MCTrace: Monte Carlo trace approximation

We develop an estimator for $\pi(\mathbf{p}|\mathbf{d})$ based on the Monte Carlo estimator for the trace of a matrix. Let $\mathbf{G} \in \mathbb{R}^{M \times N}$, and consider the computation of its squared Frobenius norm $\|\mathbf{G}\|_F^2$. This can be expressed in terms of the matrix trace as $\|\mathbf{G}\|_F^2 = \mathsf{trace}(\mathbf{G}^\top \mathbf{G})$. The Monte Carlo trace estimator approximates the trace by

$$\|\mathbf{G}\|_F^2 = \mathsf{trace}(\mathbf{G}^\top \mathbf{G}) = \mathbb{E}\left[\boldsymbol{\omega}^\top \mathbf{G}^\top \mathbf{G} \boldsymbol{\omega}\right], \tag{11}$$

where $\boldsymbol{\omega}$ is a random vector drawn from any multivariate distribution which has mean zero and identity as the covariance matrix. The Monte Carlo trace estimator replaces the expectation by a sample average, i.e.,

$$\|\mathbf{G}\|_F^2 \approx \frac{1}{\ell} \sum_{j=1}^{\ell} \boldsymbol{\omega}_j{}^\top \mathbf{G}^\top \mathbf{G} \boldsymbol{\omega}_j \;=\; \frac{1}{\ell} \mathsf{trace}(\mathbf{\Omega}^\top \mathbf{G}^\top \mathbf{G} \mathbf{\Omega}),$$

where $\boldsymbol{\omega}_j$ are independent draws and $\mathbf{\Omega} = \begin{bmatrix} \boldsymbol{\omega}_1 & \cdots & \boldsymbol{\omega}_\ell \end{bmatrix}$. The Monte Carlo estimator is an unbiased estimator for the matrix trace and the variance of this estimator depends on the particular choice of the distribution of the random vectors $\boldsymbol{\omega}$. We mention a few options for the distributions:

1. Gaussian: the entries of $\boldsymbol{\omega}$ are independent and identically distributed (i.i.d.) $\mathcal{N}(0, 1)$) random variables;
2. Rademacher: the entries of $\boldsymbol{\omega}$ are independently chosen from $\{-1, 1\}$ and with probability $1/2$;
3. Sparse Rademacher: the entries of $\boldsymbol{\omega}$ are independently chosen from $\{-\sqrt{s}, 0, \sqrt{s}\}$ and with probability $1/2s$, $1 - 1/s$, $1/2s$ respectively (here, $s > 0$ is a user-defined parameter);
4. Unit orthogonal: the vector $\boldsymbol{\omega}$ is drawn uniformly from the columns of an orthogonal matrix (e.g., the identity matrix).

The Sparse Rademacher has a few special cases worth pointing out: when $s = 1$ this is the Rademacher distribution, and when $s = 3$ this is the Achlioptas distribution [1]. A large value of s ensures that $\boldsymbol{\omega}$ is sparse and is beneficial when $\mathbf{G}$ is sparse. An analysis of various choices has been given in [8, 48, 34, 27, 19]. In this paper, we use the

Gaussian random vector as the distribution for $\boldsymbol{\omega}$ since Gaussian random vectors will also play a role in the geometry dependent methods.

Let us focus our attention on the computation of the data-misfit term $\Phi(\mathbf{p}) = \|\mathbf{D} - \mathbf{X}(\mathbf{p})\|_F^2/2\sigma^2$ where $\mathbf{X}(\mathbf{p}) = \mathbf{R}^\top \mathbf{A}^{-1}(\mathbf{p})\mathbf{S}$. Applying the Monte Carlo trace estimator to the matrix $\mathbf{G}(\mathbf{p}) \equiv \mathbf{D} - \mathbf{X}(\mathbf{p})$, it follows that

$$\Phi(\mathbf{p}) = \frac{1}{2\sigma^2}\|\mathbf{D} - \mathbf{X}(\mathbf{p})\|_F^2 \approx \frac{1}{2\ell\sigma^2}\sum_{j=1}^{\ell}\|\mathbf{D}\omega_j - \mathbf{X}(\mathbf{p})\omega_j\|_2^2 = \frac{1}{2\ell\sigma^2}\|(\mathbf{D} - \mathbf{X}(\mathbf{p}))\,\boldsymbol{\Omega}\|_F^2.$$

The last term is denoted as $\widehat{\Phi}(\mathbf{p})$. The reduction in computational costs is evident from the observation that

$$\mathbf{X}(\mathbf{p})\boldsymbol{\Omega} \;=\; \mathbf{R}^T \underbrace{\mathbf{A}^{-1}(\mathbf{p})\mathbf{S}}_{N_s-\text{solves}}\boldsymbol{\Omega} \;=\; \mathbf{R}^\top \underbrace{\mathbf{A}^{-1}(\mathbf{p})(\mathbf{S}\boldsymbol{\Omega})}_{\ell-\text{solves}}.$$

Here, if $\ell \ll N_s$, then the number of solves to be performed with the matrix $\mathbf{A}(\mathbf{p})$ is reduced from N_s to ℓ. Each column of the matrix $\mathbf{S}\boldsymbol{\Omega}$ is a random combination of the sources represented by the columns of $\mathbf{S}$ in different proportions; each column of $\mathbf{S}\boldsymbol{\Omega}$ can therefore be interpreted as a "simultaneous" random source. Similarly, the columns of $\mathbf{D}\boldsymbol{\Omega}$ are random combinations of the columns of the data matrix $\mathbf{D}$.

We briefly discuss the accuracy of the Monte Carlo trace estimators. Let $\boldsymbol{\Omega}$ is a standard Gaussian random matrix, and let $0 < \epsilon, \delta < 1$. If the number of columns of $\boldsymbol{\Omega}$ satisfies $\ell_G \geq 8\epsilon^{-2}\log(2/\delta)$, then a straightforward application of Theorem 3 in [48] gives

$$\text{Prob}\left[\left|\Phi(\mathbf{p}) - \widehat{\Phi}(\mathbf{p})\right| \leq \epsilon\Phi(\mathbf{p})\right] \geq 1 - \delta. \tag{12}$$

If instead of Gaussian random variables, Rademacher random variables are used, then for the same inequality to hold, the number of samples ℓ_R is required satisfy $\ell_R \geq 6\epsilon^{-2}\log(2/\delta)$. These results give insight into the accuracy of the log-likelihood. To quantify the accuracy of the resulting posterior distribution, we can use the approach in [35] to bound the Hellinger distance between the exact and approximate posterior distributions based on the moments of the difference between the true and approximate log-likelihoods.

4.2. Geometry exploiting randomized methods

We propose three different randomized estimators for the likelihood, depending on the geometry of the data-acquisition scheme, i.e., the geometric locations of the sources and receivers. To make matters concrete, we consider three different scenarios that are commonly used in DOT applications [46]. In the first scenario, called transmittance geometry, the sources and receivers are on either side of the domain and are well separated from each other. In the second scenario, called annular geometry, the sources and receivers surround the domain, but there is no clear separation between them. Finally, we also consider the sub-surface geometry, where the sources and receivers are on the same side of the domain and are not well-separated. The first two methods proposed are applicable when the sources and receivers are well-separated from each other (e.g., transmittance geometry) but the third method is capable of handling situations where the sources and receivers are not well-separated from each other (e.g., annular or reflectance geometry).

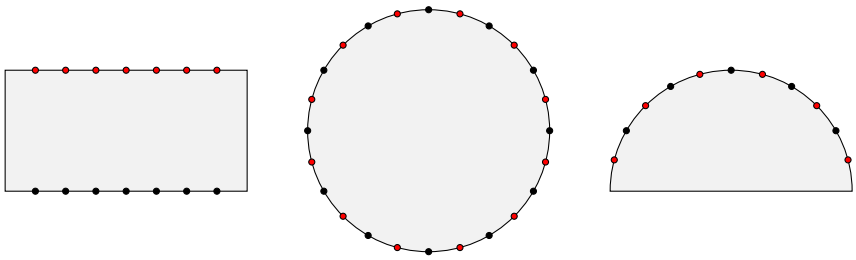

Fig. 1. Visualization of the three different scenarios of source receiver geometries: (left) transmittance, (center) annular, (right) sub-surface reflectance. The sources are denoted by red circles, and receivers by black circles, and the domain of interest is shaded.

4.2.1. RandLR: Randomized Low-rank approximation

Consider again the computation of the data-misfit part of the likelihood $\Phi(\mathbf{p}) = \|\mathbf{D} - \mathbf{X}(\mathbf{p})\|_F^2/2\sigma^2$. We first observe that the entries of the model $\mathbf{X}_{ij}(\mathbf{p}) = \mathbf{r}_i^\top \mathbf{A}(\mathbf{p})^{-1}\mathbf{s}_j$ can be seen as the discrete representation of the Green's function $G(\mathbf{x}_j, \cdot)$ due to source $\mathbf{s}_j$ (with source location $\mathbf{x}_j$) and evaluated at the receiver location, defined by $\mathbf{r}_i$. Note that in the case of DOT, the PDE corresponds to a uniformly elliptic operator; therefore, based on the arguments of [10], when the sources and receivers are well-separated, we can approximate the Green's function using a separable representation. This can be used to approximate the matrix $\mathbf{X}(\mathbf{p})$ using a low-rank representation. Numerical verification of the accuracy of the low-rank approximation has been demonstrated for the DOT application in Section 5.1. Note that the results in [10] are applicable to any uniformly elliptic PDE and, therefore, the low-rank approximation of $\mathbf{X}(\mathbf{p})$ is a feature of many imaging problems with similar structure (e.g., electrical resistivity tomography, electrical impedance tomography, hydraulic tomography, etc).

Although the results from [10] guarantee the existence of such a low-rank approximation of $\mathbf{X}(\mathbf{p})$, there are no algorithms provided to explicitly construct such a low-rank approximation. Traditional algorithms for low-rank representation, such as the singular value decomposition (SVD), rely on explicit access to the entries of the matrix. However, forming $\mathbf{X}(\mathbf{p})$ requires $\min\{N_s, N_r\}$ PDE solves, which we want to avoid. Our main idea to compute this low-rank approximation is to use the randomized SVD algorithm, previously reviewed and analyzed in [32]. Randomized SVD has the following crucial advantage compared to traditional algorithms: Computing the low-rank approximation does not require access to the entries of $\mathbf{X}(\mathbf{p})$ explicitly, but only requires matrix-vector products (henceforth, referred to as matvecs) involving $\mathbf{X}(\mathbf{p})$ and $\mathbf{X}(\mathbf{p})^\top$. We show how this can be exploited to lead to an efficient algorithm. Before proceeding, we note that other low-rank approximations can also be used in place of randomized SVD; examples include Golub-Kahan bidiagonalization [54] and sampling based methods [39].

To explain the randomized SVD approach, consider a matrix $\mathbf{G} \in \mathbb{R}^{M\times N}$, that is known to be approximately low-rank or to have rapidly decaying singular values. Let $\ell \leq \min\{M, N\}$ be the target rank of the low-rank approximation. The algorithm first samples a random matrix $\mathbf{\Omega} \in \mathbb{R}^{N\times \ell}$ with i.i.d. entries drawn from the standard normal distribution $\mathcal{N}(0, 1)$ entries. A random combination of the columns of $\mathbf{G}$, denoted as $\mathbf{Y} = \mathbf{G\Omega}$ is computed. A thin-QR factorization of $\mathbf{Y}$ is computed $\mathbf{Y} = \mathbf{QZ}$, whose columns contain an orthonormal basis for the range of $\mathbf{Y}$. The main idea behind the low-rank approach is that, if $\mathbf{G}$ has rapidly decaying singular values, then $\mathbf{Q}$ forms an approximate basis for the range of $\mathbf{G}$. This gives us the low-rank approximation $\mathbf{G} \approx \mathbf{QQ}^\top\mathbf{G}$. If the approximate SVD of $\mathbf{G}$ is needed,

compute $\mathbf{B} = \mathbf{Q}^\top \mathbf{G}$ and its thin-SVD $\mathbf{B} = \mathbf{U}_B \widetilde{\mathbf{\Sigma}}\widetilde{\mathbf{V}}^\top$; then

$$\mathbf{G} \approx \widehat{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\widetilde{\mathbf{V}}^\top \qquad \widehat{\mathbf{U}} \equiv \mathbf{Q}\mathbf{U}_B.$$

The accuracy of the low-rank approximation depends on the decay of the singular values. Precise results have been derived in [32, 28].

In the context of the MCMC approach, suppose that the current step is $\mathbf{p}_k$ and the proposed step is $\mathbf{p}_*$; we develop an approximation to the data misfit term $\Phi(\mathbf{p})$ that makes it computationally efficient to evaluate the posterior density (up to a proportionality constant) at a point $\mathbf{p}_*$ that is close to the current step $\mathbf{p}_k$. We show how to adapt the randomized SVD approach to estimate $\Phi(\mathbf{p})$. Let $\ell \leq \min\{N_s, N_r\}$ denote the target rank of the low-rank approximation we seek. As described in the general case for a matrix $\mathbf{G}$, we draw the standard Gaussian random matrix $\mathbf{\Omega} \in \mathbb{R}^{N_s \times \ell}$ and we compute $\mathbf{Y}_k = \mathbf{X}(\mathbf{p}_k)\mathbf{\Omega}$. A thin-QR factorization of $\mathbf{Y}_k$ is computed to obtain $\mathbf{Y}_k = \mathbf{Q}_k\mathbf{Z}$. Based on the previous arguments, the range of $\mathbf{Q}_k$ approximates the range of $\mathbf{X}(\mathbf{p}_k)$. For a proposed point $\mathbf{p}_*$ that is close to $\mathbf{p}_k$, we use the following rank-ℓ approximation to $\mathbf{X}(\mathbf{p}_*)$: namely, $\mathbf{X}(\mathbf{p}_*) \approx \mathbf{Q}_k\mathbf{Q}_k^\top\mathbf{X}(\mathbf{p}_*)$. We then have the local approximation of the data misfit $\Phi_{\mathbf{p}_k}(\mathbf{p}_*)$

$$\Phi_{\mathbf{p}_k}(\mathbf{p}_*) = \frac{1}{2\sigma^2}\|\mathbf{D} - \mathbf{Q}_k\mathbf{Q}_k^\top\mathbf{X}(\mathbf{p}_*)\|_F^2. \tag{13}$$

The procedure is explained in detail in Algorithm 3. Henceforth, we will refer to this approach as RandLR.

Algorithm 3 RandLR approach for estimating $\Phi(\mathbf{p}_*) \approx \widehat{\Phi}_{\mathbf{p}_k}(\mathbf{p}_*)$

Require: Parameter $1 \leq \ell \leq \min\{N_s, N_r\}$.
Matrices $\mathbf{R} \in \mathbb{R}^{n \times N_r}$, $\mathbf{S} \in \mathbb{R}^{n \times N_s}$, data $\mathbf{D} \in \mathbb{R}^{N_r \times N_s}$.
Parameters $\mathbf{p}_k$ (current step), $\mathbf{p}_*$ (proposed step), and noise variance σ^2.
{// Stage 1: Computing the basis $\mathbf{Q}_k$ at the current step $\mathbf{p}_k$}
Sample $\mathbf{\Omega} \in \mathbb{R}^{N_s \times \ell}$ with entries sampled from i.i.d. $\mathcal{N}(0, 1)$.
Compute $\mathbf{Y}_k = \mathbf{R}^\top\mathbf{A}(\mathbf{p}_k)^{-1}(\mathbf{S}\mathbf{\Omega})$ {Cost: ℓ forward solves.}
Compute thin-QR factorization $\mathbf{Y}_k = \mathbf{Q}_k\mathbf{Z}$.
{// Stage 2: Estimate of the negative log-likelihood at the proposed step $\mathbf{p}_*$}
Compute $\mathbf{B}_* = \mathbf{S}^\top\mathbf{A}^{-\top}(\mathbf{p}_*)(\mathbf{R}\mathbf{Q}_k)$ {Cost: ℓ adjoint solves.}

$$\widehat{\Phi}_{\mathbf{p}_k}(\mathbf{p}_*) = \frac{1}{2\sigma^2}\|\mathbf{D} - \mathbf{Q}_k\mathbf{B}_*\|_F^2$$

return $\widehat{\Phi}_{\mathbf{p}_k}(\mathbf{p}_*)$

We assume that the cost of this algorithm is dominated by the cost of solving the PDEs. Hence, the dominant cost of this algorithm is 2ℓ PDE solves: ℓ PDE solves to compute the basis $\mathbf{Q}_k$ and ℓ PDE solves to compute $\mathbf{B}_*$. Clearly the randomized approach is computationally effective if $2\ell \ll \min\{N_s, N_r\}$. The accuracy of the approximation depends on two factors: the proximity of $\mathbf{p}_k$ to $\mathbf{p}_*$, and the decay of the singular values of $\mathbf{X}(\mathbf{p})$. When $\mathbf{p}_* = \mathbf{p}_k$, the approximation $\mathbf{X}(\mathbf{p}_k) \approx \mathbf{Q}_k\mathbf{Q}_k^\top\mathbf{X}(\mathbf{p}_k)$ is accurate if the singular values of $\mathbf{X}(\mathbf{p}_k)$ are rapidly decaying. However, even when $\mathbf{p}_* \approx \mathbf{p}_k$, we expect the low-rank approximation $\mathbf{X}(\mathbf{p}_*) \approx \mathbf{Q}_k\mathbf{Q}_k^\top\mathbf{X}(\mathbf{p}_*)$ to be accurate. We will investigate the accuracy of the RandLR approach more carefully in the numerical experiments.

4.2.2. RandTrace: Randomized trace estimator based on subspace iteration

As before, we denote the misfit $\mathbf{G}(\mathbf{p}_k) \equiv \mathbf{D} - \mathbf{X}(\mathbf{p}_k)$ at the current iterate $\mathbf{p}_k$. In the transmittance geometry, where the sources and receivers are well-separated, we argued in Section 4.2.1 that the forward model $\mathbf{X}(\mathbf{p}_k)$ can be approximated by a low-rank matrix. From (4), the data matrix can be represented as $\mathbf{D} = \mathbf{X}(\mathbf{p}_{\text{true}}) + \mathbf{N}$ for a true parameter $\mathbf{p}_{\text{true}}$. When the noise level is sufficiently small, by the Weyl perturbation theorem [26, Corollary 8.6.2], it is reasonable to assume that the data matrix $\mathbf{D}$ is also approximately low-rank. Similarly, we argue that the misfit $\mathbf{D} - \mathbf{X}(\mathbf{p}_k)$ is also a low-rank matrix with a possibly higher numerical rank.

To estimate the misfit $\mathbf{\Phi}(\mathbf{p})$, we use the randomized trace estimator developed in [50] which is based on the randomized subspace iteration [50]. Let ℓ denote the number of samples. We first draw a standard Gaussian random matrix matrix $\mathbf{\Omega} \in \mathbb{R}^{N_s \times \ell}$ and compute $\mathbf{Y}_k = \mathbf{G}(\mathbf{p}_k)^\top \mathbf{G}(\mathbf{p}_k)\mathbf{\Omega}$ and the thin QR factorization $\mathbf{Y}_k = \mathbf{Q}_k \mathbf{Z}_k$. Then we approximate

$$\|\mathbf{D} - \mathbf{X}(\mathbf{p})\|_F^2 = \mathsf{trace}(\mathbf{G}(\mathbf{p})^\top \mathbf{G}(\mathbf{p})) \approx \|(\mathbf{D} - \mathbf{X}(\mathbf{p}))\mathbf{Q}_k\|_F^2.$$

This gives a local approximation to the posterior distribution at the current iterate $\mathbf{p}_k$; when we evaluate this at the proposed step $\mathbf{p}_*$

$$\widehat{\pi}_{\mathbf{p}_k}(\mathbf{p}_*|\mathbf{d}) \propto \exp\left(-\frac{1}{2\sigma^2}\|(\mathbf{D} - \mathbf{X}(\mathbf{p}_*))\mathbf{Q}_k\|_F^2\right)\pi(\mathbf{p}_*).$$

The procedure is summarized in Algorithm 4. The dominant computational cost of estimating the log-likelihood is ℓ forward solves and ℓ adjoint solves to compute $\mathbf{Q}_k$, and an additional cost of ℓ forward solves. Compared to RandLR this algorithm is more expensive since it requires ℓ additional forward solves.

Algorithm 4 RandTrace approach for estimating $\Phi(\mathbf{p}_*) \approx \widehat{\Phi}_{\mathbf{p}_k}(\mathbf{p}_*)$

Require: Parameter $1 \le \ell \le \min\{N_s, N_r\}$.
Matrices $\mathbf{R} \in \mathbb{R}^{n \times N_r}$, $\mathbf{S} \in \mathbb{R}^{n \times N_s}$, and data $\mathbf{D} \in \mathbb{R}^{N_r \times N_s}$.
Parameters $\mathbf{p}_k$ (current step), $\mathbf{p}_*$ (proposed step), and noise variance σ^2.

Sample a standard Gaussian random matrix $\mathbf{\Omega} \in \mathbb{R}^{N_s \times \ell}$.
{// Stage 1: Computing the basis $\mathbf{Q}_k$ at the current step $\mathbf{p}_k$}
Compute $\mathbf{Y}' = \mathbf{D}\mathbf{\Omega} - \mathbf{R}^\top \mathbf{A}(\mathbf{p}_k)^{-1}(\mathbf{S}\mathbf{\Omega})$ {Cost: ℓ forward solves.}
Compute $\mathbf{Y}_k = \mathbf{D}^\top \mathbf{Y}' - \mathbf{S}^\top \mathbf{A}(\mathbf{p}_k)^{-\top}(\mathbf{R}\mathbf{Y}')$ {Cost: ℓ forward solves.}
Compute thin-QR factorization $\mathbf{Y}_k = \mathbf{Q}_k \mathbf{Z}_k$.
{// Stage 2: Estimate of the negative log-likelihood at the proposed step $\mathbf{p}_*$}
Compute $\mathbf{B}_* = \mathbf{D}\mathbf{Q}_k - \mathbf{R}^\top \mathbf{A}^{-1}(\mathbf{p}_*)(\mathbf{S}\mathbf{Q}_k)$ {Cost: ℓ forward solves.}
return Estimate of the negative log-likelihood $\widehat{\Phi}_{\mathbf{p}_k}(\mathbf{p}_*) = \frac{1}{2\sigma^2}\|\mathbf{B}_*\|_F^2$

4.2.3. RandHODLR: randomized techniques for other geometries

The previous methods made the crucial assumption that the sources and the receivers are well-separated, which led to a low-rank approximation of the forward model $\mathbf{X}(\mathbf{p})$ and the misfit $\mathbf{D} - \mathbf{X}(\mathbf{p})$. The different randomized methods exploited this low-rank property in different ways. However, for certain source-receiver configurations, this well-separated assumption no longer holds since the sources and receivers are interspersed with each other. Examples of these configurations are the circular and the sub-surface geometry. In these circumstances, while the matrix $\mathbf{X}(\mathbf{p})$ is itself not approximately low-rank, it has a special structure which we describe and exploit.

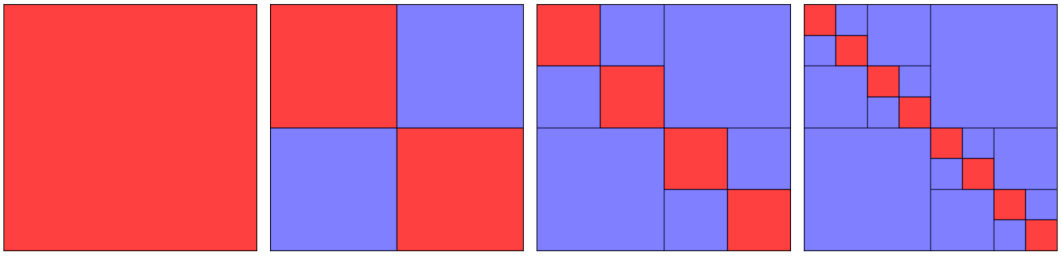

Fig. 2. A hierarchy of low-rank off diagonal blocks. Blue blocks refer to low-rank matrices, red blocks correspond to full-rank blocks.

For these geometries, the matrix $\mathbf{X}(\mathbf{p})$ can be approximated by a matrix with Hierarchical Off-Diagonal Low-Rank (HODLR) structure. A simple visualization of the hierarchy of low-rank approximations is available in Figure 2. In this structure, the diagonal blocks are stored in full-rank format, whereas the off-diagonal blocks are stored in low-rank format. We assume that there L levels, where $L \sim \log_2(N_s)$, and the off-diagonal blocks are approximated using low-rank matrices of target rank ℓ. At level 0, the matrix is effectively full-rank. At level 1, the $(1, 2)$ and $(2, 1)$ blocks can be compressed and stored in low-rank format with rank ℓ. At level 2, the $(1, 1)$ and $(2, 2)$ blocks are further compressed recursively, and so on. Most importantly, to approximate the matrix $\mathbf{X}(\mathbf{p})$ in HODLR format, the matrix need not be stored or computed explicitly but can be approximately reconstructed using a peeling algorithm originally proposed in [36]. The version of the algorithm that is used in this paper is based on [40, Section 3]. We will not reproduce this algorithm here since it is explained in great depth in that paper. We will refer to this approach as RandHODLR.

The number of matvecs involving $\mathbf{X}(\mathbf{p})$ required for approximating the matrix is $\mathcal{O}(\ell \log_2(N_s))$; since each matvec involves a forward or an adjoint PDE solve, this is also the dominant cost of the algorithm. Compared to the naive approach, RandHODLR is computationally efficient if $\ell \log_2(N_s) \ll \min\{N_s, N_r\}$. Let the current iterate be $\mathbf{p}_k$; once the approximation $\widehat{\mathbf{X}}(\mathbf{p}_k)$ has been computed in HODLR format, computing the approximation to $\Phi(\mathbf{p}_k)$ is straightforward, and is computed as $\widehat{\Phi}_{\mathbf{p}_k}(\mathbf{p}_k) = \|\mathbf{D} - \widehat{\mathbf{X}}(\mathbf{p}_k)\|_F^2/(2\sigma^2)$. In this case, we are able to approximate the misfit $\Phi(\mathbf{p}_k)$ at a given point $\mathbf{p}_k$ but cannot construct a local approximation as was done in RandLR and RandTrace approaches. This has implications in the choice of MCMC algorithms used in combination with the RandHODLR approach.

4.3. The case $\mathbf{E} \neq \mathbf{1}$ *and summary*

Thus far, we have assumed that all the source receivers pairs are active, i.e., $\mathbf{E} = \mathbf{1}$. This assumption need not hold for every application. For example, in the transmittance geometry, the sources and receivers are co-axial and for every source, only one receiver may be active. In this case $\mathbf{E} = \mathbf{I}$. For the RandLR and RandHODLR approaches, we first compute the approximation $\widehat{\mathbf{X}}(\mathbf{p})$ and then compute the approximation $\widehat{\Phi}(\mathbf{p})$. Therefore, when $\mathbf{E} \neq \mathbf{1}$, we can easily amend this algorithm to compute $\widehat{\Phi}(\mathbf{p})$ as

$$\widehat{\Phi}(\mathbf{p}) = \frac{1}{2\sigma^2}\|\mathbf{E} \odot (\mathbf{D} - \widehat{\mathbf{X}}(\mathbf{p}))\|_F^2.$$

For the RandTrace approach, we did not pursue the extension to the case $\mathbf{E} \neq \mathbf{1}$. For the MCTrace approach, when $\mathbf{E} \neq \mathbf{1}$, we can use develop stochastic approximations based on [31, 30]. For example, denoting $\mathbf{G}(\mathbf{p}) = \mathbf{D} - \mathbf{X}(\mathbf{p})$, we

can write

$$\mathbf{E} \odot \mathbf{G}(\mathbf{p}) = \mathbb{E}\left[\mathrm{diag}(\mathbf{G}(\mathbf{p})\mathbf{w})\,\mathbf{E}\,\mathrm{diag}(\mathbf{w})\right].$$

Here $\mathbf{w}$ is a random vector with zero mean and identity as the covariance matrix. Given this stochastic reformulation, we can derive the Monte Carlo approximation (see [31, Section 4])

$$\widehat{\Phi}(\mathbf{p}) = \frac{1}{2\sigma^2}\left\|\frac{1}{\ell}\sum_{j=1}^{\ell}\mathrm{diag}(\mathbf{G}(\mathbf{p})\mathbf{w}_j)\,\mathbf{E}\,\mathrm{diag}(\mathbf{w}_j)\right\|_F^2.$$

Finally, in Table 1 we summarize the various choices of randomized methods, their applicability, and associated computational costs.

	Geometry Dependent	$\mathbf{E} \neq \mathbf{1}$	Matvecs/ PDE solves
MC Trace	No	Yes	$\sim 8\epsilon^{-2}\log(2/\delta)$
RandLR	Yes	Yes	2ℓ
RandTrace	Yes	No	3ℓ
Rand HODLR	Yes	Yes	$\sim \ell\log_2(N_s)$

Table 1. A summary of the various randomized methods proposed in the paper. Here (ϵ, δ) are user-defined parameters. The number of matvecs for 'RandHOLDR' is the asymptotic cost.

4.4. Combining randomization with MCMC

In this section, we have proposed several randomized approximations $\widehat{\pi}_{\mathbf{p}}(\mathbf{p}|\mathbf{d})$. We discuss the various options of combining these randomization strategies with the MCMC algorithms. We have three different options:

Option I Use Metropolis-Hastings using the approximate likelihood $\widehat{\pi}_{\mathbf{p}}(\mathbf{p}|\mathbf{d})$. This is the approach proposed in [25],

Option II Use state-independent approximation within the delayed acceptance framework of Algorithm 2. This is essentially the surrogate transition MCMC [37, Section 9.4.3]. Note that this approach is only feasible with the MC approximation to the log-likelihood.

Option III Use state-dependent delayed acceptance algorithm, i.e., Algorithm 2.

This list of options is arranged in increasing order of computational cost per iteration. When a Monte Carlo estimator is constructed using the chain resulting from Option I, in general, the resulting estimator is biased. However, the bias may be acceptable if the resulting Monte Carlo estimator has a smaller mean square error for a given amount of computational effort, compared with the standard Metropolis–Hasting algorithm. Further details on this trade-off are given in [21, Section 5]. On the other hand, the chains resulting from Options II and III result in chains with the correct stationary distribution, provided the other necessary conditions are satisfied.

A summary of the various combinations of randomization with the MCMC options is given in Table 2.

5. Numerical Results

The first set of numerical experiments we perform investigates the accuracy of the randomized methods for evaluating the log-likelihood proposed in Section 4. We then present numerical methods that investigate the performance of the randomized methods in combination of the MCMC methods presented in Section 3.

	Method	Option I	Option II	Option III
Geometry independent	MC Trace	Yes	Yes	Yes
Geometry independent	RandLR	Yes	No	Yes
	RandTrace	Yes	No	Yes
	RandHODLR	Yes	No	No

Table 2. Summary of the various combinations of randomized log-likelihood approximations with the MCMC algorithms.

5.1. Accuracy of log-likelihood

In this set of experiments, we test the accuracy of the various randomized algorithms for approximating the log-likelihood proposed in Section 4.

Experiment 1: Transmittance geometry. The domain is taken to be $\mathcal{D} = (0, L)^2$ where $L = 4$ cm. The grid size is fixed to be 201×201 with $40,401$ degrees of freedom. The PDE is discretized using standard linear finite elements using the FEniCS package [3, 38]. We take 256 sources evenly spaced in the interval $x \in [0.2L, 0.8L]$ with $y = 0$, the receivers have the same x values but with $y = L$. We take $\mu_a^p = 1.0$ cm^{-1} and $\mu_a^b = 0.06$ cm^{-1} and $\mu_s' = 9$ cm^{-1}.

The true image is chosen to be a PaLS representation with three basis functions and randomly chosen centers; the parameter $\mathbf{p}$ defining the point at which the log-likelihood is evaluated is chosen similarly but is different than the true image. We add 1% Gaussian noise to simulate measurement error.

The sources and the receivers are well separated, therefore, this setup satisfies the assumptions of the methods in Sections 4.2.1 and 4.2.2. We call these two methods RandLR and RandTrace respectively. We compare this with the Monte Carlo approach proposed in Section 4.1 and the Monte Carlo runs were averaged over 100 different runs. The results are reported in Fig. 3, where the relative error is plotted against the number of samples ℓ. Note that the cost of RandLR is 2ℓ PDE solves and RandTrace is 3ℓ PDE solves, whereas the cost of MCTrace is ℓ PDE solves. We see

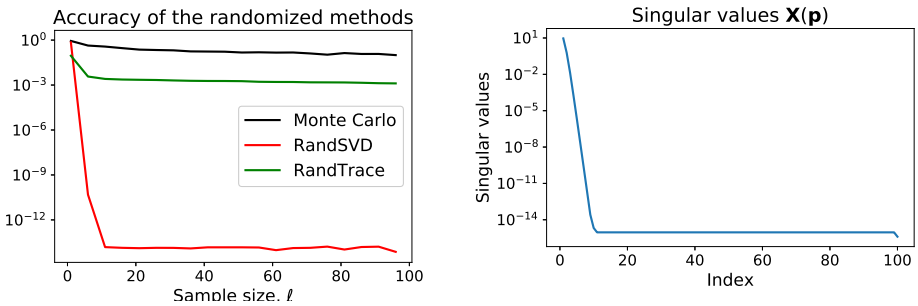

Fig. 3. Relative Error of the randomized methods to evaluate the misfit $\Phi(\mathbf{p})$. We compare the MCTrace, RandLR, and RandTrace approaches as a function of the parameter ℓ; note that the computational cost of each algorithm is different.

that the errors in the MCTrace and the RandTrace approach are comparable and acceptable if a coarse approximation is desired. On the other hand, the RandLR approximation is highly accurate (in terms of relative errors) and a sample size of $\ell = 5 - 10$ is sufficient for this application. A closer examination revealed that the singular values of the matrix $\mathbf{X}(\mathbf{p})$ decayed exponentially, which explains the accuracy of the RandLR approach. Using a sample size $\ell = 10$ gave relative error on the order of machine precision, with a speedup by a factor of $256/20 \approx 13$.

Experiment 2: Annular geometry. In this setup, the domain is taken to be a circle of radius $R = 4$ cm with $N_s = 256$ sources and receivers. The domain is discretized with $20,054$ degrees of freedom. The sources and receivers are co-located and evenly spaced on the circumference of the domain, but we neglect the source/source interaction. That is, we take $\mathbf{S} = \mathbf{R}$ and $\mathbf{E} = \mathbf{1} - \mathbf{I} \in \mathbb{R}^{N_s \times N_s}$; in other words, the matrix $\mathbf{E}$ has entries 1 except along the diagonals where it has zero entries. The true image is chosen to be a PaLS representation with 3 basis functions and randomly chosen centers; the parameter $\mathbf{p}$ defining the point at which the log-likelihood is evaluated is chosen similarly but is different than the true image. Once again, we use the same optical parameters and add 1% noise to the data to simulate measurement noise.

In this case, since the sources and receivers are not well-separated, so the RandLR approach is not suitable, and therefore, we use the RandHODLR approach to approximate $\mathbf{X}(\mathbf{p})$. We use the peeling algorithm as described in [40], with $\log_2(N_s/16) = 4$ levels, as outlined in Section 4.2.3. In Table 3, we report the computational cost (measured as number of PDE solves), speedup, and the relative error in the data-misfit $\Phi(\mathbf{p})$. As the parameter ℓ increases, the cost increases but the relative error decreases sharply. We see that we obtain speedups of ~ 2 using the RandHODLR algorithm for reasonable relative error $\sim 10^{-3}$. Additional speedups can be obtained by utilizing the symmetry of $\mathbf{X}(\mathbf{p})$, which we have ignored in the present implementation of the peeling algorithm.

Parameter	PDE solves	Speedup	Error
$\ell = 3$	68	3.8	1.83×10^{1}
$\ell = 5$	92	2.8	2.07×10^{-1}
$\ell = 7$	116	2.2	8.03×10^{-3}
$\ell = 9$	140	1.8	8.41×10^{-4}

Table 3. Relative error in the RandHODLR approach with varying parameter ℓ which controls target rank of the off-diagonal blocks. The number of sources and receivers are $N_s = N_r = 256$. Reported also are the number of PDE solves to compute using the RandHODLR approach, and the computational speedup. The last column reports the relative error in the misfit $\Phi(\mathbf{p})$.

5.2. MCMC results

In the previous experiment we investigated the accuracy of the randomized approximations to the log-likelihood. In the next set of experiments, we now investigate the use of randomized approximations within the MCMC algorithms. We focus on the transmission geometry case as in Experiment 1 of Section 5.1 and consider two different variations. We use the same domain size, distribution of sources and receivers and optical parameters as in that experiment. The number of sources and receivers are taken to be 100 each; since each source-receiver pair is assumed to be active, this results in 10^4 measurements. We use a grid with $51 \times 51 = 2601$ degrees of freedom to discretize the PDE. To simulate measurement error we add 1% additive Gaussian noise.

1. Setup 1 The true image is taken to be an amoeba shaped object (see Figure 4). Once again, we use $n_p = 3$ basis functions. We fix the parameters $\alpha_k = 1$ and $\beta_k = 1.7$ for $k = 1, \ldots, 3$ and only invert for the location of the radial basis functions; therefore, the inverse problem only has $p = 2n_p = 6$ unknown parameters. The prior distribution is taken to be the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Gamma})$ with mean $\boldsymbol{\mu} = 2\mathbf{e}$, where $\mathbf{e} \in \mathbb{R}^6$ is the vector of ones, and the covariance $\boldsymbol{\Gamma} = 10^2\mathbf{I}$. Note that the mean of the prior distribution is the center of the domain.

2. Setup 2 We consider a true image which consists of two horizontal bars of thickness 0.8 cm (see Figure 8) and use 1% additive Gaussian noise to simulate measurement error. We use $n_p = 4$ basis functions, fix the parameters $\alpha_k = 1$ and $\beta_k = 1$ for $k = 1, \ldots, 4$ and only invert for the locations of the radial basis functions; in this instance of the inverse problem we have $p = 2n_p = 8$ unknown parameters. The prior distribution is taken to be the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Gamma})$ with mean $\boldsymbol{\mu} = 2\mathbf{e}$, where $\mathbf{e} \in \mathbb{R}^8$ is the vector of ones, and the covariance $\boldsymbol{\Gamma} = 10^2\mathbf{I} \in \mathbb{R}^{8\times 8}$.

Note that in both cases, we are avoiding inverse crimes since the true image cannot be exactly represented using the parameterizations adopted in the paper. Note that only allowing the centers $\boldsymbol{\chi}$'s to move limits the type of images that can be represented using this parameterization. However, the emphasis of this paper is on the computational costs and an exploration of different parameterizations and prior distributions is being pursued in ongoing work.

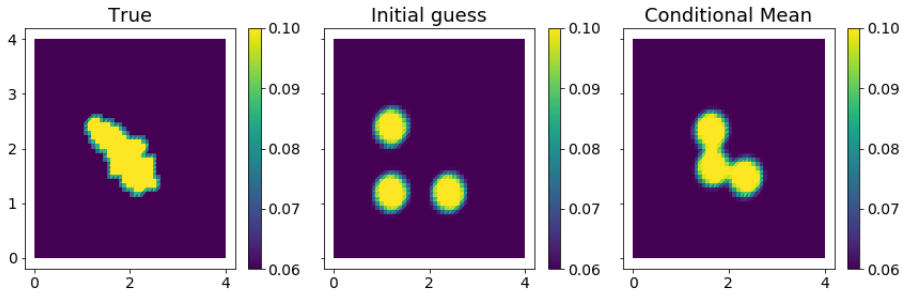

Fig. 4. (left) True image, (center) initial guess for all three methods, (right) and image obtained using the conditional mean (only the latter 50% of the chain was used).

5.2.1. MH algorithm

We first consider the MH described in Section 3, with a simple adaptation scheme to select the step size of the random walk. We consider two different randomized techniques for mitigating the computational cost of evaluating the log-likelihood: the MCTrace approach and the RandLR approach, and compare these approaches with exact log-likelihood evaluation. We choose a sample size $\ell = 20$ for the MCTrace approach and $\ell = 10$ for the RandLR approach. With this choice, both the approximate algorithms have the same cost of evaluating the likelihood. Furthermore, the random matrix $\boldsymbol{\Omega}$ was kept fixed throughout the simulation.

In each case, the MCMC algorithm is run for 5×10^5 iterations and the first half of the chain is discarded to mitigate the effects of burn in. For the full log-likelihood evaluation, we display the trace plots of all the variables in Figure 5. The trace plots show that the Markov chain appears to be mixing well. In Figure 4, we show the true image, the initial starting guess, and the conditional mean (the average is computed in parametric space). We omit the corresponding plots for the chains obtained using the approximate algorithms since they are very similar.

Some other statistics of the chains are given in Table 4. To compute the IACT, we used the implementation in the `emcee` package [24]. The essential sample size (ESS) is the total number of iterations divided by the integrated

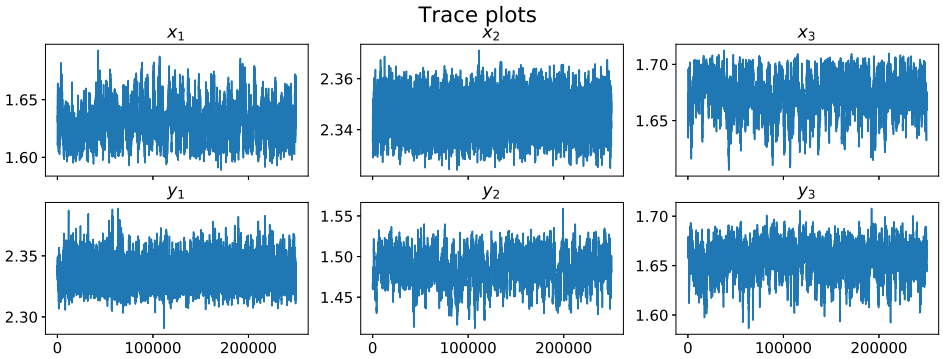

Fig. 5. Trace plots of the parameters p, which denote the centers χ's of the basis functions. The chain was generated using MH on the 'exact' log-likelihood evaluation. The first half of the chain was treated as the burn-in stage and are, therefore, discarded.

Method	AR	IACT	ESS	MSJ	Speedup
Full	0.31	685.43	364	1.77×10^{-5}	–
MCTrace	0.21	333.14	750	2.14×10^{-5}	5
RandLR	0.31	685.43	364	1.77×10^{-5}	5

Table 4. We report various MCMC diagnostic metrics for the three different methods. The MCMC algorithms were run using Option I in Section 3; 'Full' means that no randomization was used in the log-likelihood evaluations.

autocorrelation time (IACT). The mean squared jump (MSJ) is defined as

$$\text{MSJ} = \frac{1}{M} \sum_{j=1}^{M-1} \|\mathbf{p}_{j+1} - \mathbf{p}_j\|_2^2.$$

Here $M = N_{\text{chain}} - N_b$ is the number of iterations retained after burn-in. It is seen that the sampling statistics for the full log-likelihood evaluation is very similar to the results of RandLR approach, suggesting the high accuracy of the log-likelihood evaluations in the RandLR approach. Interestingly, MCTrace has a lower IACT suggesting that the chain has better mixing properties than without the approximation; a similar observation was made in [42, Section 3]. Finally, in Figure 6, we plot the empirical distribution of the various inversion parameters. The plot was made using the `corner.py` package [23].

Both the approximate log-likelihood evaluations are 5 times faster (cost measured in number of PDE solves) than the full log-likelihood evaluation. The gains in computational cost are more substantial when more sources are used but the same number of samples ℓ are used. The resulting bias due to the use of an approximate likelihood is likely to be small compared to the variance of the Monte Carlo estimator.

We now consider the Setup 2 and investigate the performance of the randomized algorithms on this setup. We use the RandLR approximation to the log-likelihood and use it with Metropolis-Hastings (Option I) with the parameter $\ell = 10$; once again we obtain a speedup of a factor of 5. We run the algorithm for 10^6 iterations and discard the first 25% of the chain due to burn-in. The trace plots are displayed in Figure 8 and provide a sanity check for the convergence. The acceptance rate was about 39%, the IACT was 3781.39 (corresponding ESS is approximately 198), and the MSJ was 2.21×10^{-6}. The empirical densities of the parameters is reported in Figure 9.

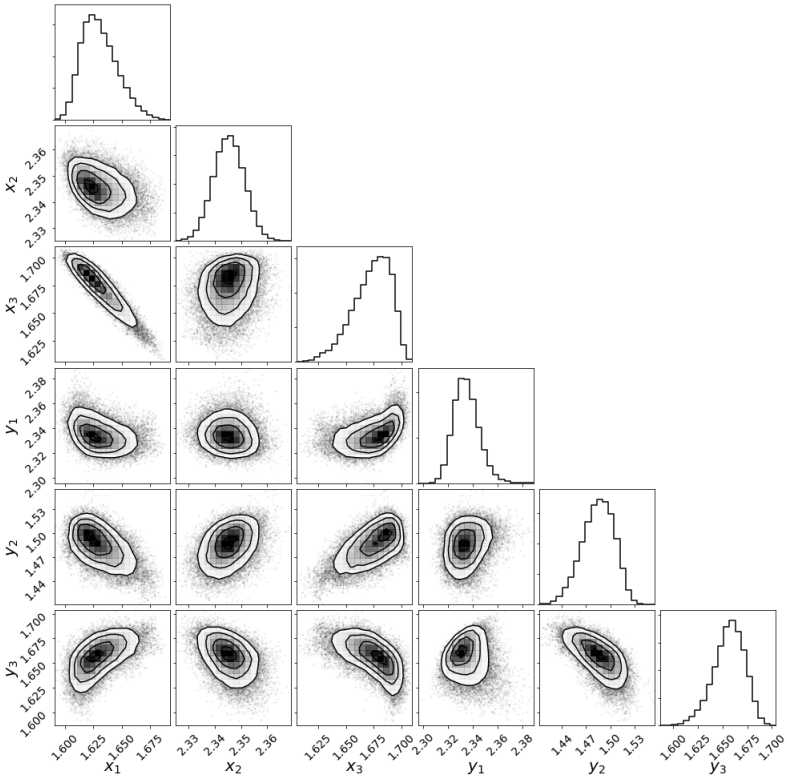

Fig. 6. Individual and pairwise marginal densities of the different inversion parameters corresponding to Setup 1 (see Figure 4). We used the last 25% of the chain and thinned it down further by retaining every 10-th element.

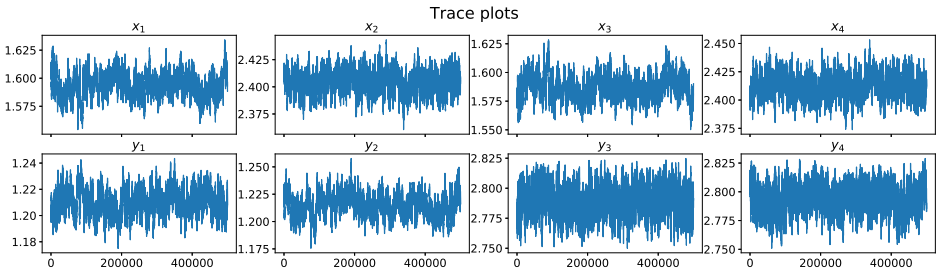

Fig. 7. Trace plots of entries of the parameters p, which denote the centers χ's of the basis functions. The chain was generated using the MH algorithm with the 'exact' log-likelihood evaluation. The first 25% of the chain was treated as the burn-in stage and therefore discarded.

5.2.2. Two-stage MCMC

Finally, we consider the two-stage approach discussed in Section 3 applied to Setup 1. We first consider the RandLR approach for the approximate log-likelihood evaluation, i.e., we use (13) as the local approximation—this corresponds to Option III in Section 4.4. In this experiment, we vary the parameter ℓ and report various MCMC diagnostic parameters in Table 5. Here 'AR1' and 'AR2' denote the ratio of number of accepted steps to the total number of iterations. We see that when $\ell = 3$, we have $(0.39/0.41) \times 100 \approx 95\%$ acceptance rate at the second stage, whereas when $\ell \geq 5$ we have 100% acceptance rate at the second stage. This is consistent with the observation in

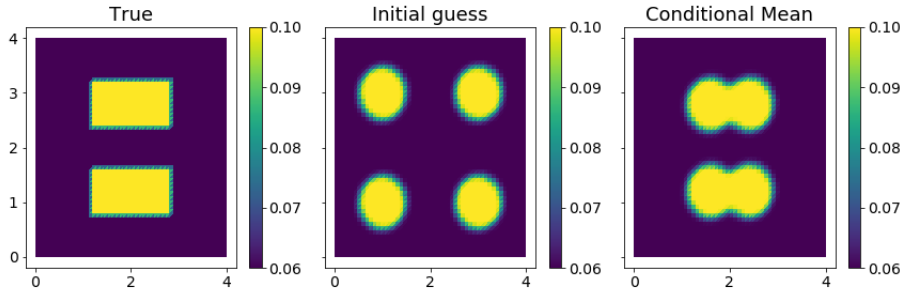

Fig. 8. Results for Setup 2: (left) True image, (center) initial guess for all three methods, (right) and image obtained using the conditional mean (the first 25% of the chain was discarded due to burn-in).

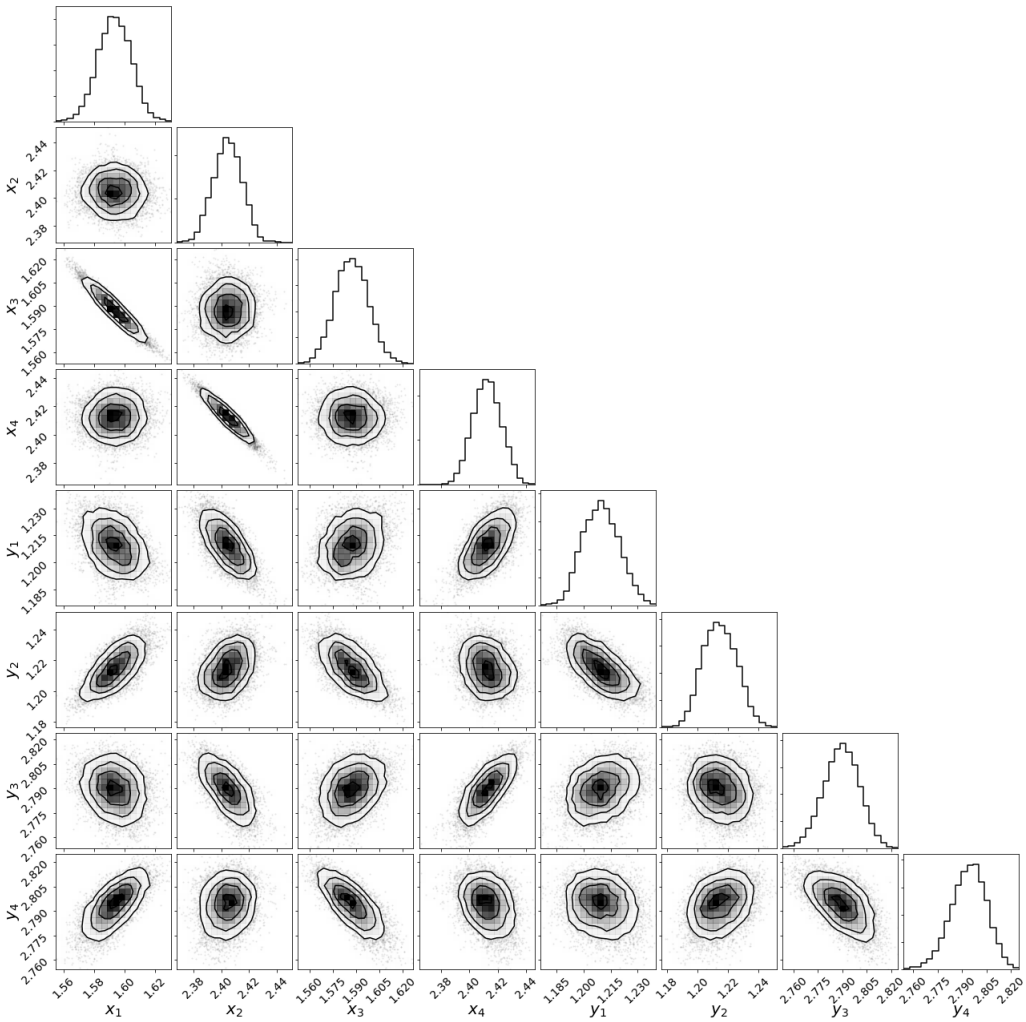

Fig. 9. Individual and pairwise marginal densities of the different inversion parameters corresponding to Setup 2 (see Figure 8). We discarded the first 25% of the chain to burn-in and thinned it down further by retaining every 100-th element.

Section 5.1, that the RandLR approach for evaluating the log-likelihood is highly accurate.

We discuss the speedup of the two-stage approach. The cost of the full log-likelihood evaluation measured in the

ℓ	AR1	AR2	IACT	ESS	MSJ	$\text{Speedup}_{\text{comp}}$	$\text{Speedup}_{\text{stat}}$
3	0.41	0.39	1736.4	143	1.38×10^{-5}	2.09	0.82
5	0.30	0.30	885.53	282	1.80×10^{-5}	2.53	1.95
10	0.31	0.31	788.57	317	1.77×10^{-5}	1.98	1.72

Table 5. Various MCMC diagnostic metrics using the Two-stage approach with RandLR approximation as the approximation which corresponds to Option III in Section 3. 'AR1' is the acceptance ratio at the first stage, and 'AR2' denotes the acceptance ratio at the second stage (fraction of proposed samples to the total number of iterations); AR2/AR1 gives the fraction of the number accepted to the number promoted.

number of PDE solves is $t_f = N_s = 100$. The cost at the first stage of the algorithm is $t_1 = \ell$ PDE solves, since the basis $\mathbf{Q}_k$ is available, and the cost at the second stage is $t_2 = n_2 + 3\ell$ PDE solves since it involves computing the full likelihood (N_s solves), computing the basis $\mathbf{Q}_{\mathbf{p}_*}$ (ℓ PDE solves) and $\pi_{\mathbf{p}_*}(\mathbf{p}_k|\mathbf{d})/\pi_{\mathbf{p}_*}(\mathbf{p}_*|\mathbf{d})$ (2ℓ PDE solves). Therefore, the computational and statistical speedups take the form

$$\text{Speedup}_{\text{comp}} = \frac{N_s}{\eta(3\ell + N_s) + \ell}, \qquad \text{and} \qquad \text{Speedup}_{\text{stat}} = \frac{\tau_f}{\tau_t}\frac{N_s}{\eta(3\ell + N_s) + \ell}.$$

From the table, we also see that the IACT for the two stage approach τ_t is greater than the IACT for the exact likelihood approach τ_f, which is to be expected since the Two-Stage approach is not statistically efficient compared to the MH algorithm. Therefore, while the computational speedup $\text{Speedup}_{\text{comp}} \approx 2$, the statistical speedup are smaller and can be smaller than 1. Based on these results the parameter $\ell = 5$ appears to strike the best balance computational cost as well as statistical efficiency. However, since the acceptance ratio at the second stage seems to be close to 100%, for maximum computational efficiency (5× speedup) we can use RandLR with MH (Option I).

We briefly comment on the use MCTrace approach for approximating the log-likelihood in the two-stage approach. If the random matrix $\mathbf{\Omega}$ is considered fixed throughout the iterative process, we have a state-independent approximation of the log-likelihood. This corresponds to Option II as described in Section 4.4. For small sample sizes we found that the IACT of the two stage approach was sufficiently high that the statistical speedup $\text{Speedup}_{\text{stat}}$ were less than 1 for all the parameters of ℓ we tried $(10 - 60)$. Furthermore, the percentage of iterates at the second stage compared to those promoted was between $30 - 70\%$. Therefore, we found the cost benefit versus accuracy of the MCTrace approach did not make it suitable for the Two-Stage approach.

6. Conclusion

Motivated by the computational cost of MCMC algorithms involving repeated evaluations of the log-likelihood, which in turn can have hundreds of PDE evaluations, we proposed several randomized methods for mitigating this cost. The randomized methods come in two flavors depending on whether or not we take into account the geometric locations of the sources and receivers. Numerical experiments showed that the geometry-dependent methods are both computationally efficient and very accurate. However, the geometry-independent method, MCTrace, has benefits since it does not require significant assumptions making it widely applicable to many scenarios, and is very easy to implement but suffers from low accuracy. When the randomized approximations to the log-likelihood are used in the standard MH algorithm without any convergence guarantees, the resulting iterates have similar behavior to the

MH algorithm with exact log-likelihood evaluations. In the two-stage approach, the RandLR approximation had very high acceptance at the second stage, suggesting that RandLR can safely be used in the MH algorithm, which yields a dramatic speedup in computational costs.

It should be mentioned that the randomized techniques developed here can be used along with other techniques for computational cost reduction such as model reduction techniques [22], and recycling Krylov subspace methods [43] to produce more efficient MCMC algorithms. Another avenue for research appears to be using optimized sources/detectors in addition to randomized sources as was done in our previous work [7]. We are currently investigating the use of the parameteric level set approach for uncertainty quantification for piecewise constant reconstruction in Bayesian inverse problems, allowing for a richer class of target shapes than used in the numerical experiments in Section 5.

7. Acknowledgements

The authors would like to acknowledge support from the National Science Foundation through the grants: DMS-1720398 (A.K.S.); DMS-1720305 (P.P. and E.d.S.); DMS-1720291 and CCF-1934553 (E.M and M.K.).

References

[1] D. Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of computer and System Sciences*, 66(4):671–687, 2003.

[2] A. Aghasi, M. Kilmer, and E. L. Miller. Parametric level set methods for inverse problems. *SIAM Journal on Imaging Sciences*, 4(2):618–650, 2011.

[3] M. S. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells. The FEniCS project version 1.5. *Archive of Numerical Software*, 3(100), 2015.

[4] C. Andrieu and G. O. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725, 2009.

[5] S. R. Arridge. Optical tomography in medical imaging. *Inverse problems*, 15(2):R41, 1999.

[6] S. Aslan, E. de Sturler, and S. Gugercin. Randomization for the efficient computation of parametric reduced order models for inversion. *arXiv preprint arXiv:2007.06027*, 2020.

[7] S. S. Aslan, E. de Sturler, and M. E. Kilmer. Randomized approach to nonlinear inversion combining random and optimized simultaneous sources and detectors. *SIAM Journal on Scientific Computing*, 41(2):B229–B249, 2019.

[8] H. Avron and S. Toledo. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *Journal of the ACM (JACM)*, 58(2):8, 2011.

[9] G. Bal, I. Langmore, and Y. Marzouk. Bayesian inverse problems with Monte Carlo forward models. *Inverse Problems & Imaging*, 7:81, 2013.

[10] M. Bebendorf and W. Hackbusch. Existence of $\mathcal{H}$-matrix approximants to the inverse FE-matrix of elliptic operators with L^∞-coefficients. *Numerische Mathematik*, 95(1):1–28, 2003.

[11] S. Brooks, A. Gelman, G. L. Jones, and X.-L. Meng, editors. *Handbook of Markov chain Monte Carlo*. Chapman & Hall/CRC Handbooks of Modern Statistical Methods. CRC Press, Boca Raton, FL, 2011.

[12] D. A. Brown, A. Saibaba, and S. Vallélian. Low-rank independence samplers in hierarchical Bayesian inverse problems. *SIAM/ASA Journal on Uncertainty Quantification*, 6(3):1076–1100, 2018.

[13] T. Bui-Thanh, O. Ghattas, J. Martin, and G. Stadler. A computational framework for infinite-dimensional Bayesian inverse problems Part i: The linearized case, with application to global seismic inversion. *SIAM Journal on Scientific Computing*, 35(6):A2494–A2523, 2013.

[14] D. M. Ceperley and M. Dewing. The penalty method for random walks with uncertain energies. *The Journal of chemical physics*, 110(20):9812–9820, 1999.

[15] P. Chen and C. Schwab. Adaptive sparse grid model order reduction for fast Bayesian estimation and inversion. In *Sparse Grids and Applications-Stuttgart 2014*, pages 1–27. Springer, 2016.

[16] J. A. Christen and C. Fox. Markov chain Monte Carlo using an approximation. *Journal of Computational and Graphical statistics*, 14(4):795–810, 2005.

[17] P. R. Conrad, Y. M. Marzouk, N. S. Pillai, and A. Smith. Accelerating asymptotically exact MCMC for computationally intensive models via local approximations. *Journal of the American Statistical Association*, 111(516):1591–1607, 2016.

[18] P. G. Constantine, C. Kent, and T. Bui-Thanh. Accelerating markov chain monte carlo with active subspaces. *SIAM Journal on Scientific Computing*, 38(5):A2779–A2805, 2016.

[19] A. Cortinovis and D. Kressner. On randomized trace estimates for indefinite matrices with an application to determinants. *arXiv preprint arXiv:2005.10009*, 2020.

[20] T. Cui, C. Fox, and M. O'sullivan. Bayesian calibration of a large-scale geothermal reservoir model by a new adaptive delayed acceptance Metropolis Hastings algorithm. *Water Resources Research*, 47(10), 2011.

[21] T. Cui, Y. M. Marzouk, and K. E. Willcox. Data-driven model reduction for the Bayesian solution of inverse problems. *International Journal for Numerical Methods in Engineering*, 102(5):966–990, 2015.

[22] E. De Sturler, S. Gugercin, M. E. Kilmer, S. Chaturantabut, C. Beattie, and M. O'Connell. Nonlinear parametric inversion using interpolatory model reduction. *SIAM Journal on Scientific Computing*, 37(3):B495–B517, 2015.

[23] D. Foreman-Mackey. corner.py: Scatterplot matrices in Python. *The Journal of Open Source Software*, 24, 2016.

[24] D. Foreman-Mackey, D. W. Hogg, D. Lang, and J. Goodman. emcee: The MCMC hammer. *Publications of the Astronomical Society of the Pacific*, 125:306–312, 2013.

[25] C. Fox and G. Nicholls. Sampling conductivity images via MCMC. *The art and science of Bayesian image analysis*, pages 91–100, 1997.

[26] G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, fourth edition, 2013.

[27] S. Gratton and D. Titley-Peloquin. Improved bounds for small-sample estimation. *SIAM Journal on Matrix Analysis and Applications*, 39(2):922–931, 2018.

[28] M. Gu. Subspace iteration randomization and singular value problems. *SIAM Journal on Scientific Computing*, 37(3):A1139–A1173, 2015.

[29] H. Haario, E. Saksman, and J. Tamminen. An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223–242, 2001.

[30] E. Haber and M. Chung. Simultaneous source for non-uniform data variance and missing data. *arXiv preprint arXiv:1404.5254*, 2014.

[31] E. Haber, M. Chung, and F. Herrmann. An effective method for parameter estimation with PDE constraints with multiple right-hand sides. *SIAM Journal on Optimization*, 22(3):739–757, 2012.

[32] N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[33] E. Laloy, B. Rogiers, J. A. Vrugt, D. Mallants, and D. Jacques. Efficient posterior exploration of a high-dimensional groundwater model from two-stage Markov chain Monte Carlo simulation and polynomial chaos expansion. *Water Resources Research*, 49(5):2664–2682, 2013.

[34] E. B. Le, A. Myers, T. Bui-Thanh, and Q. P. Nguyen. A data-scalable randomized misfit approach for solving large-scale PDE-constrained inverse problems. *Inverse Problems*, 33(6):065003, 2017.

[35] H. C. Lie, T. Sullivan, and A. L. Teckentrup. Random forward models and log-likelihoods in Bayesian inverse problems. *SIAM/ASA Journal on Uncertainty Quantification*, 6(4):1600–1629, 2018.

[36] L. Lin, J. Lu, and L. Ying. Fast construction of hierarchical matrix representation from matrix–vector multiplication. *Journal of Computational Physics*, 230(10):4071–4087, 2011.

[37] J. S. Liu. *Monte Carlo strategies in scientific computing*. Springer Science & Business Media, 2008.

[38] A. Logg, K.-A. Mardal, and G. N. Wells. *Automated Solution of Differential Equations by the Finite Element Method*. Springer, 2012.

[39] M. W. Mahoney and P. Drineas. CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702, 2009.

[40] P.-G. Martinsson. Compressing rank-structured matrices via randomized sampling. *SIAM Journal on Scientific Computing*, 38(4):A1959–A1986, 2016.

[41] Y. M. Marzouk and H. N. Najm. Dimensionality reduction and polynomial chaos acceleration of Bayesian inference in inverse problems. *Journal of Computational Physics*, 228(6):1862–1902, 2009.

[42] G. K. Nicholls, C. Fox, and A. M. Watt. Coupled MCMC with a randomized acceptance probability. *arXiv preprint arXiv:1205.6857*, 2012.

[43] M. O'Connell, M. E. Kilmer, E. de Sturler, and S. Gugercin. Computing reduced order models via inner-outer Krylov recycling in diffuse optical tomography. *SIAM Journal on Scientific Computing*, 39(2):B272–B297, 2017.

[44] P. D. O'Neill, D. J. Balding, N. G. Becker, M. Eerola, and D. Mollison. Analyses of infectious disease data from household outbreaks by Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 49(4):517–542, 2000.

[45] B. Peherstorfer, K. Willcox, and M. Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *SIAM Review*, 60(3):550–591, 2018.

[46] B. W. Pogue, T. O. McBride, U. L. Osterberg, and K. D. Paulsen. Comparison of imaging geometries for diffuse optical tomography of tissue. *Optics Express*, 4(8):270–286, Apr 1999.

[47] C. Robert and G. Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[48] F. Roosta-Khorasani and U. Ascher. Improved bounds on sample size for implicit matrix trace estimators. *Foundations of Computational Mathematics*, 15(5):1187–1212, 2015.

[49] F. Roosta-Khorasani, K. Van Den Doel, and U. Ascher. Stochastic algorithms for inverse problems involving PDEs and many measurements. *SIAM Journal on Scientific Computing*, 36(5):S3–S22, 2014.

[50] A. K. Saibaba, A. Alexanderian, and I. C. Ipsen. Randomized matrix-free trace and log-determinant estimators. *Numerische Mathematik*, 137(2):353–395, 2017.

[51] A. K. Saibaba, J. Bardsley, D. A. Brown, and A. Alexanderian. Efficient marginalization-based MCMC methods for hierarchical Bayesian inverse problems. *SIAM/ASA Journal on Uncertainty Quantification*, 7(3):1105–1131, 2019.

[52] A. K. Saibaba and P. K. Kitanidis. Fast computation of uncertainty quantification measures in the geostatistical approach to solve inverse problems. *Advances in water resources*, 82:124–138, 2015.

[53] M. Schweiger, S. R. Arridge, M. Hiraoka, and D. T. Delpy. The finite element method for the propagation of light in scattering media: Boundary and source conditions. *Medical Physics*, 22(11):1779–1792, 1995.

[54] H. D. Simon and H. Zha. Low-rank matrix approximation using the Lanczos bidiagonalization process with applications. *SIAM Journal on Scientific Computing*, 21(6):2257–2274, 2000.

[55] A. Stuart and A. Teckentrup. Posterior consistency for Gaussian process approximations of bayesian posterior distributions. *Mathematics of Computation*, 87(310):721–753, 2018.

[56] J. A. Vrugt. Markov chain Monte Carlo simulation using the DREAM software package: Theory, concepts, and MATLAB implementation. *Environmental Modelling & Software*, 75:273–316, 2016.

[57] L. Yan and T. Zhou. Adaptive multi-fidelity polynomial chaos approach to Bayesian inference in inverse problems. *Journal of Computational Physics*, 381:110–128, 2019.

[58] L. Yan and T. Zhou. An adaptive surrogate modeling based on deep neural networks for large-scale bayesian inverse problems. *Communications in Computational Physics*, 28(5):2180–2205, 2020.