Parameter estimation by implicit sampling

Matthias Morzfeld[1,2,*], Xuemin Tu[3], Jon Wilkening[1,2] and Alexandre J. Chorin[1,2]

[1]Department of Mathematics, University of California, Berkeley, CA 94720, USA.
[2]Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA.
[3]Department of Mathematics, University of Kansas, Lawrence, KS 66045, USA.

Abstract

Implicit sampling is a weighted sampling method that is used in data assimilation to sequentially update state estimates of a stochastic model based on noisy and incomplete data. Here we apply implicit sampling to sample the posterior probability density of parameter estimation problems. The posterior probability combines prior information about the parameter with information from a numerical model, e.g. a partial differential equation (PDE), and noisy data. The result of our computations are parameters that lead to simulations that are compatible with the data. We demonstrate the usefulness of our implicit sampling algorithm with an example from subsurface flow. For an efficient implementation we make use of multiple grids, BFGS optimization coupled to adjoint equations, and Karhunen-Loève expansions for dimensional reduction. Moreover, several difficulties of Markov Chain Monte Carlo methods, e.g. estimation of burn-in times or correlations among the samples, are avoided because the implicit samples are independent.

1. Introduction

We wish to compute a set of parameters θ, an m-dimensional vector, so that simulations with a numerical model that requires these parameters are compatible with data z (a k-dimensional vector) we have collected. We assume that some information about the parameter is available before we collect the data and this information is summarized in a prior probability density function (pdf) $p(\theta)$. For example, one may know a priori that some of the parameters are positive. The numerical model, e.g. a partial differential equation (PDE), defines the likelihood $p(z|\theta)$, which describes how the parameters are connected with the data. Bayes' rule combines the prior and likelihood to find the posterior density

$$p(\theta|z) \propto p(\theta)p(z|\theta),$$

see, e.g. [41]. This posterior pdf defines which parameters of the numerical model are compatible with the data z. The goal in parameter estimation is to compute the posterior pdf.

If the prior and likelihood are Gaussian, then the posterior is also Gaussian, and it is sufficient to compute the mean and covariance of $\theta|z$ (because the mean and covariance define the Gaussian). The posterior mean and covariance are the minimizer and the inverse of the Hessian of the negative logarithm of a Gaussian posterior pdf. In nonlinear and non-Gaussian problems one can compute the posterior mode, often called the maximum a posteriori (MAP) point, by minimizing the negative logarithm of the posterior, and use the MAP point (instead of the mean) as an approximation of the parameter θ. The inverse of the Hessian of the negative logarithm of the posterior can be used to measure the uncertainty of this approximation. This method is sometimes called linearization about the MAP point (LMAP) or the Laplace approximation [6, 23, 35, 36].

One can also use Markov Chain Monte Carlo (MCMC) to solve a parameter estimation problem. In MCMC, one generates a collection of samples from the posterior pdf see, e.g. [13, 16, 28, 37]. The samples form an empirical estimate of the posterior, and statistics, e.g. the mean or mode, can be computed from this empirical estimate by averaging over the samples. Under mild assumptions, the averages one computes from the samples converge to the expected values with respect to the

* Corresponding author. Tel: +1 510 486 6335. Email address: mmo@math.lbl.gov. Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, California 94720, USA.

posterior pdf as the number of samples goes to infinity. In practice a finite number of samples is used and successful MCMC sampling requires that one can test if the chain has converged to the posterior pdf. The convergence can be slow due to correlations among the samples.

An alternative to MCMC is to use importance sampling. The idea is to draw samples from an importance function and to attach a weight to each sample such that the weighted samples form an empirical estimate of the posterior (see, e.g. [8]). The efficiency of importance sampling depends on the importance function which in turn defines the weights. Specifically, if the variance of the weights is large, then the weighted samples are a poor empirical estimate of the posterior and the number of samples required can increase quickly with the dimension of the problem [4,5,11,40]. For this reason, importance sampling has not been used for parameter estimation problems in which the dimension is usually large. We investigate if implicit sampling which has been used before in online-filtering/data assimilation [2, 9, 10, 12, 30, 31, 43], can overcome this issue.

We will describe how to apply implicit sampling to parameter estimation problems and it will become clear that an important step in implicit sampling is to minimize the negative logarithm of the posterior pdf, i.e. to find the MAP point. This optimization step identifies the region where the posterior probability is large, i.e. the region where the high-probability samples are. Starting from the MAP point, implicit sampling generates samples in its vicinity to explore the regions of high posterior probability. The optimization in implicit sampling represents the link between implicit sampling and MAP. In fact, the optimization methods used in MAP codes can be used for implicit sampling, however implicit sampling captures non-Gaussian characteristics of the posterior, which are usually missed by LMAP.

We illustrate the efficiency of our implicit sampling algorithm with numerical experiments using a problem from subsurface flow [3,35]. This problem is a common test problem for MCMC algorithms, and the conditions for the existence of a posterior measure and its continuity are well understood [13]. Earlier work on this problem includes [16], where Metropolis-Hastings MC sampling is used, and [32], which uses optimal maps and is further discussed below.

The remainder of this paper is organized as follows. In section 2 we explain how to use implicit sampling for parameter estimation and discuss an efficient implementation. Numerical examples are provided in section 3. Conclusions are offered in section 4.

2. Implicit sampling for parameter estimation

We wish to estimate an m-dimensional parameter vector θ from data which are obtained as follows. One measures a function of the parameters $h(\theta)$, where h is a given k-dimensional function; the measurements are noisy, so that the data z satisfy the relation:

$$z = h(\theta) + r, \tag{1}$$

where r is a random variable with a known distribution and the function h maps the parameters onto the data. Often, the function h involves solving a PDE. In a Bayesian approach, one obtains the pdf $p(\theta|z)$ of the conditional random variable $\theta|z$ by Bayes' rule:

$$p(\theta|z) \propto p(\theta)p(z|\theta), \tag{2}$$

where the likelihood $p(z|\theta)$ is given by (1) and the prior $p(\theta)$ is assumed to be known.

The goal is to sample the posterior and use the samples to calculate useful statistics. This can be done with importance sampling as follows [8, 24]. One can represent the posterior by M weighted samples. The samples θ_j, $j = 1, \ldots, M$ are obtained from an importance function $\pi(\theta)$ (which is chosen such that it is easy to sample from), and the jth sample is assigned the weight

$$w_j \propto \frac{p(\theta_j)p(z|\theta_j)}{\pi(\theta_j)}.$$

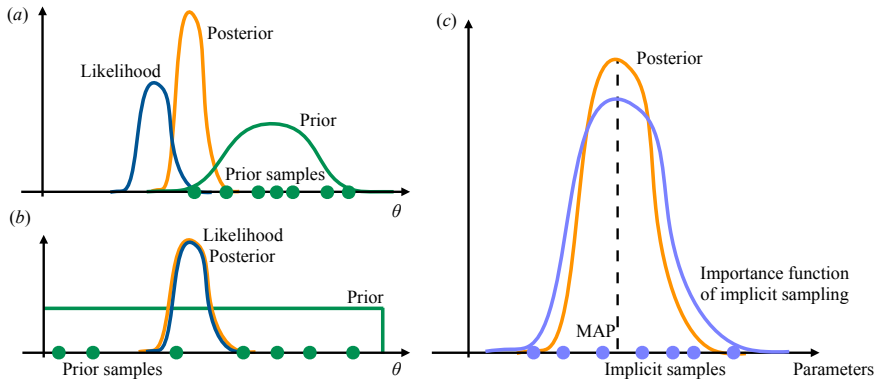

FIGURE 1. (a) The prior and likelihood are nearly mutually singular so that prior samples receive a small posterior probability. (b) The prior is broad and the likelihood is sharply peaked, so that the majority of prior samples receives a small posterior probability. (c) The importance function of implicit sampling assigns probability mass to the neighborhood of the MAP point, so that its overlap with the posterior pdf is significant, which leads to implicit samples that receive a large posterior probability.

A sample corresponds to a set of possible parameter values and the weight describes how likely this set is in view of the posterior. The weighted samples $\{\theta_j, w_j\}$ form an empirical estimate of $p(\theta|z)$, so that for a smooth function u, the sum

$$E_M(u) = \sum_{j=0}^{M} u(\theta_j)\hat{w}_j,$$

where $\hat{w}_j = w_j / \sum_{j=0}^{M} w_j$, converges almost surely to the expected value of u with respect to $p(\theta|z)$ as $M \to \infty$, provided that the support of π includes the support of $p(\theta|z)$ [8, 24].

The importance function must be chosen carefully or else sampling can become inefficient. For example, suppose you choose the prior as the importance function. In this case, the weights are proportional to the likelihood. Thus, one first draws samples from the prior and then determines their posterior probability by comparing them with the data. However, the samples one draws from the prior lie in the region where the prior probability is high and this region may not overlap with the region where the posterior probability is high. Two important scenarios in which this happens are (i) the prior and likelihood have (almost) disjoint support, i.e. the prior assigns probability mass in a small region of the (parameter) space in which the likelihood is small and vice versa, see figure 1a; (ii) the prior is broad, however the likelihood is sharply peaked, see figure 1b.

In either scenario, the samples we draw from the prior typically receive a low posterior probability so that the resulting empirical estimate of the posterior is inaccurate. An accurate empirical estimate requires samples with a high posterior probability, and a large number of prior samples may be required to obtain a few samples with high posterior probability. In fact, the number of samples required can increase catastrophically with the dimension of the problem, so that this importance sampling algorithm cannot be applied to high-dimensional problems [4, 5, 11, 40].

2.1. **Basic ideas of implicit sampling.** The idea in implicit sampling is to construct a data-informed importance function, which has a significant overlap with the posterior pdf (see figure 1c). This requires in particular that the importance function be large where the posterior pdf is large. We can find one point where the posterior pdf is large by minimizing its negative logarithm, i.e. we

find the MAP point as in MAP methods. To set up the notation, let

$$F(\theta) = -\log\left(p(\theta)p(z|\theta)\right), \tag{3}$$

so that the MAP point is the minimizer of F,

$$\mu = \arg\min_{\theta} F(\theta).$$

Our goal is to construct an importance function that assigns probability to the neighborhood the MAP point. For the construction, we first use a random variable ξ with pdf $p(\xi) \propto \exp\left(G(\xi)\right)$, which is easy to sample (e.g. a Gaussian). The variable ξ assigns probability mass to the neighborhood of its mode, the minimizer of G. Next we define a new random variable, x, implicitly by the solutions of the algebraic equations

$$F(x) - \phi = G(\xi) - \gamma, \tag{4}$$

where $\phi = \min F$ and $\gamma = \min G$. The pdf of x can be calculated by a change of variables

$$\pi(x) = p(\xi(x)) \left|\det\left(\frac{\partial \xi}{\partial x}\right)\right|,$$

provided the map $\xi \to x$ is one-to-one and onto. Note that several mappings $\xi \to x$ exist because (4) is underdetermined, it is a scalar equation in m variables. The pdf $\pi(x)$ is the importance function of implicit sampling and samples are drawn by solving (4). The samples are computed by solving (4), and their weights are

$$w_j \propto \frac{p(\theta|z)}{\pi(x)} \propto \underbrace{\exp\left(G(\xi(x)) - F(\theta)\right)}_{=\exp(\gamma-\phi)=\text{const.}} \left|\det\left(\frac{\partial \xi}{\partial x}\right)\right| \propto \left|\det\left(\frac{\partial \xi}{\partial x}\right)\right|, \tag{5}$$

proportional to the Jacobian of the map from x to ξ.

Note that a typical draw from the variable ξ is close to mode of ξ so that G evaluated at a typical sample of ξ is close to its minimum γ. Thus, the left-hand-side of equation (4) is likely to be small. A small left-hand-side implies a small right-hand-side so that the function F evaluated at the solution of (4) is close to its minimum ϕ. This forces the solutions of (4) to lie near the MAP point μ. Thus, by repeatedly solving (4) for several draws of the variable ξ, we explore the neighborhood of the MAP point.

2.2. **Solving the implicit equation.** We describe and implement two strategies for solving (4) for a Gaussian ξ with mean 0 and covariance matrix H^{-1}, where H is the Hessian of the function F at the minimum. With this ξ, equation (4) becomes

$$F(\theta) - \phi = \frac{1}{2}\xi^T H \xi. \tag{6}$$

Both algorithms are affine invariant and, therefore, capable of sampling within flat and narrow valleys of F, see [20] for a discussion of the importance of affine invariance in Monte Carlo sampling.

2.2.1. *Random maps.* One can look for solutions of (6) in a random direction, ξ,

$$\theta = \mu + \lambda(\xi)\,\xi. \tag{7}$$

The stretch factor λ can be computed by substituting (7) into (6), and solving the resulting equation for the scalar $\lambda(\xi)$ with Newton's method. A formula for the Jacobian of the random map defined by (6) and (7) was derived in [21, 31],

$$w \propto |J(\xi)| = \left|\lambda^{m-1}\, \frac{\xi^T H \xi}{\nabla_\theta F \cdot \xi}\right| \tag{8}$$

where m is the number of non-zero eigenvalues of H. The Jacobian is easy to evaluate if the gradient of F is easy to compute, e.g. using the adjoint method (see below).

2.2.2. *Linear maps.* An alternative strategy is to approximate F by its Taylor expansion to second order

$$F_0(\theta) = \phi + \frac{1}{2}(\theta - \mu)^T H(\theta - \mu),$$

where $\mu = \arg\min F$ is the minimizer of F (the MAP point) and H is the Hessian at the minimum. This strategy is called "implicit sampling with linear maps" and requires that one solves the quadratic equation

$$F_0(\theta) - \phi = \frac{1}{2}\xi^T H \xi, \tag{9}$$

instead of (6). This can be done by simply shifting ξ by the mode, $\theta = \mu + \xi$. The bias created by solving the quadratic equation (9) instead of (6) can be removed by the weights [2, 9]

$$w \propto \exp\left(F_0(\theta) - F(\theta)\right). \tag{10}$$

A comparison of the linear and random map methods is given in [21], where it is found that the random map loses its advantages as the dimension of the problem increases if the posterior is a small perturbation of a Gaussian. We will confirm this theory with our numerical examples below.

2.2.3. *Connections with optimal maps.* An interesting construction, related to implicit sampling, has been proposed in [32,39]. Suppose one wants to generate samples with the pdf $p(\theta|z)$, and have θ be a function of a variable ξ with pdf g, as above. If the samples are all to have equal weights, one must have, in the notations above,

$$p(\theta|z) = g(\xi)/J(\xi),$$

where, as above, J is the Jacobian of a map $\theta \to \xi$. Taking logs, one finds

$$F(\theta) + \log\beta = G(\xi) - \log\left(J(\xi)\right), \tag{11}$$

where $\beta = \int p(z|\theta)p(\theta)d\theta$ is the proportionality constant that has been elided in (2). If one can find a one-to-one mapping from ξ to θ that satisfies this equation, one obtains an optimal sampling strategy, where the pdf of the samples matches exactly the posterior pdf. In [32], this map is found globally by choosing $g = p(\theta)$ (the prior), rather than sample-by-sample as in implicit sampling. The main differences between the implicit sampling equation (4) and equation (11) are the presence of the Jacobian J and of the normalizing constant β in the latter; J has shifted from being a weight to being a term in the equation that picks the samples, and the optimization that finds the probability mass has shifted to the computation of the map.

If ξ is Gaussian and the problem is linear, equation (11) can be solved by a linear map with a constant Jacobian and this map also solves (4), so that one recovers implicit sampling. In particular, in a linear Gaussian problem, the local (sample-by-sample) map (4) of implicit sampling also solves the global equation (11), which, for the linear problem, is a change of variables from one Gaussian to another. If the problem is not linear, the task of finding a global map that satisfies (11) is difficult (see also [15, 27, 39, 44]). The determination of optimal maps in [32], based on nonlinear transport theory, is elegant but can be computationally intensive, and requires approximations that reintroduce non-uniform weights. Using (simplified) optimal maps and re-weighting the samples from approximate maps is discussed in [39]. In [34], further optimal transport maps from prior to posterior are discussed. These maps are exact in linear Gaussian problems, however in general they are approximate, due to neglecting the Jacobian, when the problem is nonlinear.

2.3. **Adjoint based optimization with multiple grids.** The first step in implicit sampling is to find the MAP point by minimizing F in (3). This can be done numerically by Newton, quasi-Newton, or Gauss-Newton methods (see, e.g. [33]). The minimization requires derivatives of the function F.

We consider parameter estimation problems in which the function h in (1) typically involves solving a PDE. In this case, adjoints are efficient for computing the gradient of F. The reason is

that the complexity of solving the adjoint equation is similar to that of solving the original "forward" model. Thus, the gradient can be computed at the cost of (roughly) two forward solutions. Adjoint methods are used widely in LMAP methods and can be used in connection with a quasi-Newton method, e.g. BFGS, or with Gauss-Newton methods. We illustrate how to use the adjoint method for BFGS optimization in the example below.

During the optimization one can make use of multiple grids. This idea first appeared in the context of online state estimation in [2], and is similar to a multi-grid finite difference method [17] and multi-grid Monte Carlo [19]. However, the idea is different from the usual "multi-grid" method (which is why we call it optimization with multiple-grids). The idea is as follows. First, initialize the parameters and pick a coarse grid. Then perform the minimization on the coarse grid and use the minimizer to initialize a minimization on a finer grid. The minimization on the finer grid should require only a few steps, since the initial guess is informed by the computations on the coarser grid, so that the number of fine-grid forward and adjoint solves is small. This procedure can be generalized to use more than two grids (see the example below).

3. Application to subsurface flow

We illustrate the applicability of our implicit sampling method by a numerical example from subsurface flow, where we estimate subsurface structures from pressure measurements of flow through a porous medium. This is a common test problem for MCMC and has applications in reservoir simulation/management (see e.g. [35]) and groundwater pollution modeling (see e.g. [3]).

We consider Darcy's law

$$u = -\frac{\kappa}{\mu}\nabla p,$$

where ∇p is the pressure gradient across the porous medium, μ is the viscosity and u is the average flow velocity; κ is the permeability and describes the subsurface structures we are interested in. Assuming, for simplicity, that the viscosity is constant, we obtain, from conservation of mass, the elliptic problem

$$-\nabla \cdot (\kappa \nabla p) = g, \tag{12}$$

on a domain Ω, with Dirichlet boundary conditions, and where the source term g represents externally prescribed inward or outward flow rates. For example, if a well were drilled and a constant inflow were applied through this well, g would be a delta function with support at the well.

The uncertain quantity in this problem is the permeability, i.e. κ is a random variable, whose realizations we assume to be smooth enough so that for each realization of κ, a unique solution of (12) exists. We would like to update our knowledge about κ on the basis of noisy measurements of the pressure at k locations within the domain Ω so that (1) becomes

$$z = h(p(\kappa), x, y) + r, \tag{13}$$

where r is a random variable.

In the numerical experiments below we consider a 2D-problem on a square domain $\Omega = [0,1] \times [0,1]$, and discretize (12) with a (standard) piecewise linear finite element method on a uniform $(N+1) \times (N+1)$ mesh of triangular elements [7]. We use the balancing domain decomposition by constraints method [14] to solve the resulting symmetric linear systems, i.e. we first decompose the computational domain into smaller subdomains and then solve a subdomain interface problem. The right hand side g are four delta distributions in the center of the domain (see figure 2).

Our finest grid is 64×64 and the pressure measurements and forcing g are arranged such that they align with grid points of our fine and coarse grids (which we use in the multiple-grid approach). The 49 pressure measurement are collected in the center of the domain (see figure 2).

The pressure measurements are perturbed with a Gaussian random variable $r \sim \mathcal{N}(0, R)$, with a diagonal covariance matrix R (i.e. we assume that measurement errors are uncorrelated). The

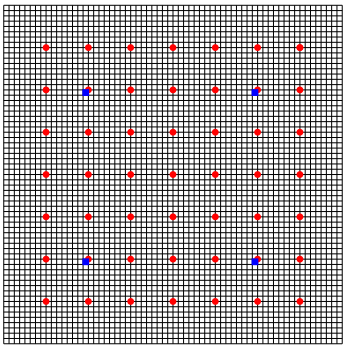

FIGURE 2. Mesh of the square domain (gray lines), pressure measurements (red dots), and forcing locations (delta distributions, blue squares)

variance at each measurement location is set to 30% of the reference solution. This relatively large variance brings about significant non-Gaussian features in the posterior pdf.

3.1. **The log-normal prior, its discretization and dimensional reduction.** The prior for permeability fields is often assumed to be log-normal and we follow suit. Specifically, the continuous permeability field is assumed log-normal with a squared exponential covariance function [38],

$$K(x_1, x_2, y_1, y_2) = \exp\left(-\frac{(x_1 - x_2)^2}{l_x^2} - \frac{(y_1 - y_2)^2}{l_y^2}\right), \tag{14}$$

where (x_1, y_1), (x_2, y_2) are two points in the domain Ω, and where the correlation lengths are equal $l_x = l_y = 0.5$. This prior models the (log-) permeability as a smooth function of x and y, so that solutions of the PDE (12) uniquely exist. Moreover, the theory presented in [13, 41] applies and a well defined posterior also exists for the continuous problem.

The random permeability field is discretized on our uniform grid by a finite-dimensional random variable with a log-normal distribution. The elements of the covariance matrix Σ are obtained from the continuous correlation function (14)

$$\Sigma(i, j) = K(x_i, x_j, y_i, y_j), \quad i, j = 1, \ldots, N,$$

where N is the number of grid points in each direction. We perform a dimension reduction via Karhunan-Loève (KL) expansions [18, 25], and use the resulting low-rank approximation of the covariance matrix Σ for all subsequent computations. Specifically, the factorization of the covariance function $K(x_1, x_2, y_1, y_2)$ into the x and y directions allows us to compute the covariance matrices in each direction separately, i.e. we compute the matrices Σ_x and Σ_y with elements

$$\Sigma_x(i, j) = \sigma_x^2 \exp\left(-\frac{(x_i - x_j)^2}{l_x^2}\right), \quad \Sigma_y(i, j) = \sigma_y^2 \exp\left(-\frac{(y_i - y_j)^2}{l_y^2}\right).$$

We then compute singular value decompositions (SVD) in each direction to form low-rank approximations $\hat{\Sigma}_x \approx \Sigma_x$ and $\hat{\Sigma}_y \approx \Sigma_y$ by neglecting small eigenvalues. These low rank approximations define a low rank approximation of the covariance matrix

$$\Sigma \approx \hat{\Sigma}_x \otimes \hat{\Sigma}_y,$$

where $\otimes$ is the Kronecker product. Thus, the eigenvalues and eigenvectors of $\hat{\Sigma}$ are the products of the eigenvalues and eigenvectors of $\hat{\Sigma}_x$ and $\hat{\Sigma}_y$. We obtain the low-rank approximation for the covariance matrix on the grid from the SVD of the covariance in each direction:

$$\hat{\Sigma} = V^T \Lambda V,$$

where Λ is a diagonal matrix whose diagonal elements are the m largest eigenvalues of Σ and V is an $m \times N$ matrix whose columns are the corresponding eigenvectors. Our approximate covariance $\hat{\Sigma}$ is optimal in the sense that the difference of the Frobenius norms of Σ and $\hat{\Sigma}$ is minimized. With $m = 30$ eigenvalues, we capture 99.9% of the variance (in the sense that the sum of the first 30 eigenvalues is 99% of the sum of all eigenvalues).

Thus, in reduced coordinates on the grid, the prior is

$$\hat{K} \sim \ln \mathcal{N}\left(\hat{\mu}, \hat{\Sigma}\right).$$

Exponentiating followed by the linear change of variables

$$\theta = V^T \Lambda^{-0.5} \hat{K},$$

gives a prior for the "effective parameters" θ,

$$p(\theta) = \mathcal{N}\left(\mu, I_m\right), \tag{15}$$

where $\mu = V^T \Lambda^{-0.5} \hat{\mu}$. We will carry out the computations in the reduced coordinates θ. This reduces the effective dimension of the problem from N^2 (4096 for our finest grid), to $m = 30$. The model reduction follows naturally from assuming that the permeability is smooth, so that the prior is correlated, and the probability mass localizes in parameter space. A similar observation, in connection with data assimilation, was made in [11].

3.2. **Multiple grids and adjoint based BFGS optimization.** Implicit sampling requires minimization of F in (3) which in reduced coordinates of this problem takes the form

$$F(\theta) = \frac{1}{2}\theta^T\theta + \frac{1}{2}\left(z - MP(\theta)\right)^T R^{-1}\left(z - MP(\theta)\right),$$

where M is a $k \times N^2$ matrix that defines at which locations on the (fine) grid we collect the pressure. We solve the optimization problem using BFGS coupled to an adjoint code to compute the gradient of F with respect to θ (see also, e.g. [22, 36]).

The adjoint calculations are as follows. The gradient of F with respect to θ is

$$\nabla_\theta F(\theta) = \theta + \left(\nabla_\theta P(\theta)\right)^T W,$$

P is a N^2 vector that contains the pressure on the grid and where $W = -M^T R^{-1}(z - MP(\theta))$. We use the chain rule to derive $\left(\nabla_\theta P(\theta)\right)^T W$ as follows:

$$(\nabla_\theta P(\theta))^T W = \left(\nabla_K P(\theta)\frac{\partial K}{\partial \hat{K}}\frac{\partial \hat{K}}{\partial \theta}\right)^T W = \left(\nabla_K P(\theta) e^{\hat{K}} V \Lambda^{0.5}\right)^T W = \left(V\Lambda^{0.5}\right)^T \left(\nabla_K P(\theta) e^{\hat{K}}\right)^T W,$$

where $e^{\hat{K}}$ is a $N^2 \times N^2$ diagonal matrix whose elements are the exponentials of the components of $\hat{K}$. The gradient $\nabla_K P(\theta)$ can be obtained directly from our finite element discretization. Let $P = P(\theta)$ and let K_l be the lth component of K, and take the derivative with respect to K_l of our finite element discretization to obtain

$$\frac{\partial P}{\partial K_l} = -A^{-1}\frac{\partial A}{\partial K_l}P,$$

where A is the $N^2 \times N^2$ matrix that defines the linear system we solve and where $\partial A/\partial K_l$ are component-wise derivatives. We use this result to obtain the following expression

$$\left(\nabla_K P(\theta) e^{\hat{K}}\right)^T W = -\left(e^{\hat{K}}\right)^T \begin{bmatrix} P^T \frac{\partial A}{\partial K_1}\left(A^{-T}W\right) \\ \vdots \\ P^T \frac{\partial A}{\partial K_{N^2}}\left(A^{-T}W\right) \end{bmatrix}. \tag{16}$$

When P is available, the most expensive part in (16) is to evaluate $A^{-T}W$, which is equivalent to solving the adjoint problem (which is equal to itself for this self-adjoint problem). The rest can be

Grid	Iterations	Forward solves
16×16	9	32
32×32	6	14
64×64	5	12

TABLE 1. Required iterations and function evaluations for multiple-grid optimization.

computed element-wise by the definition of A. Note that there are only a fixed number of nonzeros in each $\frac{\partial A}{\partial K_l}$, so that the additional work for solving the adjoint problem in (16) is about $O(N^2)$, which is small compared to the work required for the adjoint solve.

Collecting terms we finally obtain the gradient

$$\nabla_\theta F(\theta) = \theta + \left(V\Lambda^{0.5}\right)^T \left(\nabla_K P(\theta) e^{\hat{K}}\right)^T W \quad = \quad \theta - \left(V\Lambda^{0.5}\right)^T \left(e^{\hat{K}}\right)^T \begin{bmatrix} P^T \frac{\partial A}{\partial K_1} \left(A^{-T} W\right) \\ \vdots \\ P^T \frac{\partial A}{\partial K_{N^2}} \left(A^{-T} W\right) \end{bmatrix}.$$

Multiplying by $\left(V\Lambda^{0.5}\right)^T$ to go back to physical coordinates will require additional work of $O(mN^2)$. Note that the adjoint calculations for the gradient require only one adjoint solve because the forward solve (required for P) has already been done before the gradient calculation in the BFGS algorithm. In summary, our adjoint solves are only slightly more expensive than the forward solves. This concludes our derivation of an adjoint method for gradient computations.

We use this adjoint based gradient computations in a BFGS method with a cubic interpolation line search (see [33, Chapter 3]). We use the multiple-grids approach to reduce the number of fine-grid solves. We use three grids, 16×16, 32×32,and 64×64. The required number of iterations on each grid and the number of forward solves are summarized in table 1. After converting the cost of coarse/medium-grid solves to the cost of fine-grid solves, we estimate the cost of the multiple-grids optimization with 17 fine grid solves. Without multiple-grids, 36 fine grid solves are needed to find the same minimum.

3.3. **Implementation of the random and linear maps.** We generate samples using the linear map and random map methods described above. Both require the Hessian of F at the minimum. A direct finite difference method for the Hessian would require $m(m+1) = 930$ forward solves, which is too expensive (infeasible if m becomes larger). For that reason, we approximate the Hessian by

$$H \approx \hat{H} = I - Q^T (QQ^T + R)^{-1} Q, \tag{17}$$

where $Q = M \nabla_\theta P$, as is standard in LMAP methods [23]. Here the gradient of the pressure (or the Jacobian) is computed with finite differences, which requires $m + 1$ forward solves.

With this approximate Hessian generating samples with the random map method requires solving (6) with the ansatz (7). We use a Newton method for solving these equations and observe that it usually converges quickly (within 1-4 iterations). Each iteration requires a derivative of F with respect to λ, which we implement using the adjoint method, so that each iteration requires two forward solutions. In summary, the random map method requires between 2-8 forward solutions per sample. The linear map method requires generating a Gaussian sample and weighting it by (10), so that only one forward solve is required per sample.

The quality of the weighted ensembles of the random and linear map methods can be assessed by the variance of the weights. A well-distributed ensemble has a small variance of the weights. The variance of the weights is equal to $R - 1$, where

$$R = \frac{E(w^2)}{E(w)^2}.$$

In fact, R itself can be used to measure the quality of the samples [1, 42]. If the variance of the weights is small, then $R \approx 1$. Moreover, the effective sample size, i.e. the number of unweighted samples that would be equivalent in terms of statistical accuracy to the set of weighted samples, is about M/R [42], where M is the number of samples we draw. In summary, an R close to one indicates a well-distributed ensemble.

We compute a value of R of about 1.6 for both methods. In fact, we generate 10 synthetic data sets, run implicit sampling with random and linear maps on each set and estimate R based on 10^4 samples for each numerical experiment. We compute a $R = 1.68 \pm 0.10$, for the linear map method and $R = 1.63 \pm 0.066$ for the random map method. The random map method thus performs slightly better, however the cost per sample is also slightly larger (because generating a sample requires solving (6), which in turn requires solving the forward problem). Since the linear map method is less expensive and easier to program, it is a more appropriate technique for this problem.

We have also experimented with symmetrization of implicit sampling [21], which is similar in spirit to the classic trick of antithetic variates [24]. The symmetrization of the linear map method is as follows. Sample ξ and compute a sample $x^+ = \mu + \xi$. Use the same ξ to compute $x^- = \mu - \xi$. Then pick x^+ with probability $p^+ = w(x^+)/(w(x^+) + w(x^-))$ and pick x^- with probability $p^- = w(x^-)/(w(x^+) + w(x^-))$, and assign the weight $w^s = (w(x^+) + w(x^-))/2$. This symmetrization can lead to a smaller R, i.e. a better distributed ensemble, in the small noise limit. In our example, we compute the quality measure $R = 1.4$. While this R is smaller than for the non-symmetrized methods, the symmetrization does not pay off in this example, since each sample of the symmetrized method requires two forward solves (to evaluate the weights).

3.4. **Comparisons with other methods.** The MAP and LMAP methods estimate parameters by computing the MAP point, i.e. the most likely parameters in view of the data, and estimate the uncertainty by a Gaussian whose covariance is the inverse of the Hessian of F at the minimum [6, 23, 35, 36]. In our example, LMAP overestimates the uncertainty since the Gaussian approximation has a standard deviation of 0.93 for the first parameter θ_1, whereas we compute 0.64 with the linear map and random map methods. The reason for the over-estimation of the uncertainty with LMAP is that the posterior is not Gaussian. This effect is illustrated in figure 3 where we show histograms of the marginals of the posterior for the first four parameters $\theta_1, \theta_2, \theta_3, \theta_4$, along with their Gaussian approximation as in LMAP. We also compute the skewness and excess kurtosis for these marginal densities. While the marginals for the parameters may become "more Gaussian" for the higher order coefficients of the KL expansion, the joint posterior exhibits significant non-Gaussian behavior. Since implicit sampling (with random or linear maps) does not require linearizations or Gaussian assumptions, it can correctly capture these non-Gaussian features. In the present example, accounting for the non-Gaussian effects brings about a significant reduction of the uncertainty.

Note that code for LMAP, can be converted into an implicit sampling code. In particular, implicit sampling with linear maps requires the MAP point and an approximation of the Hessian at the minimum. Both can be computed with LMAP codes. Non-Gaussian features of the posterior can then be captured by weighted sampling with linear maps, where each sample comes at a cost of a single forward simulation.

Another important class of methods for solving Bayesian parameter estimation problems is MCMC. We compare implicit sampling with Metropolis MCMC [26], where we use an isotropic Gaussian proposal density, for which we tuned the variance to achieve an acceptance rate of about 30%. This method requires one forward solution per step (to compute the acceptance probability). We start the chain at the MAP (to reduce burn-in time). In figure 4 we show the approximation of the conditional mean of the variables θ_1, θ_2, and θ_3, as a function of the number of steps in the chain (left). We observe that, even after 10^4 steps, the chain has not settled, in particular for the parameter θ_3 (see bottom left).

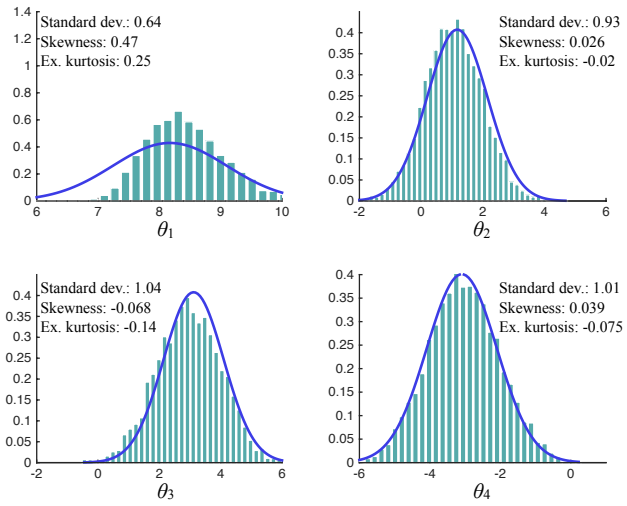

FIGURE 3. Marginals of the posterior computed with implicit sampling with random maps and their Gaussian approximation obtained via LMAP. Top left: $p(\theta_1|z)$. Top right: $p(\theta_2|z)$. Bottom left: $p(\theta_3|z)$. Bottom right: $p(\theta_4|z)$.

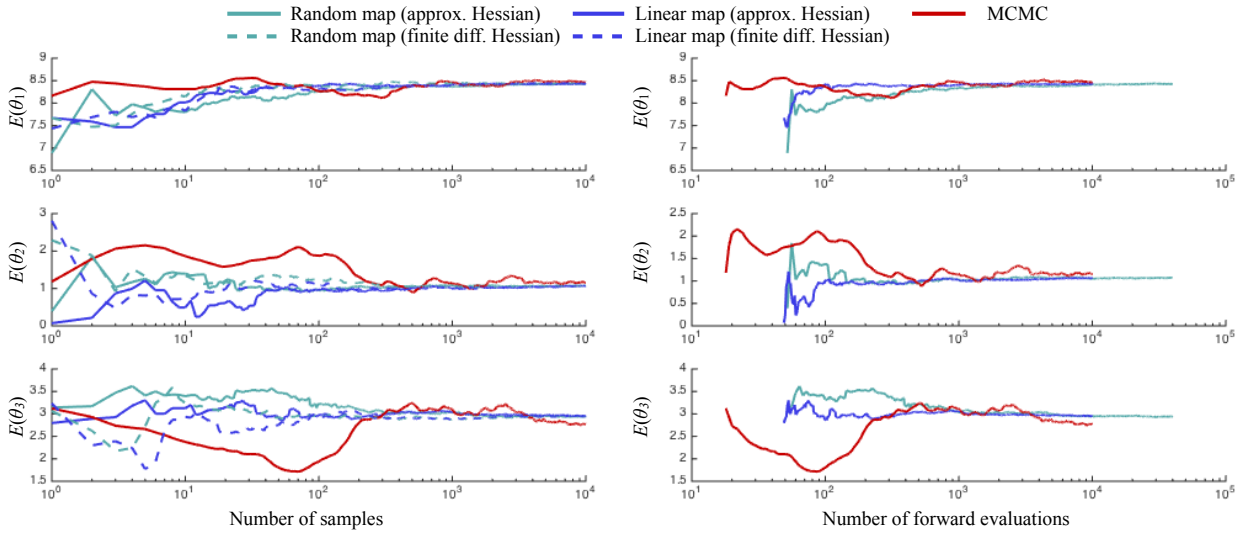

FIGURE 4. Expected value as a function of the number of samples (left), and as a function of required forward solves (right). Red: MCMC. Turquoise: implicit sampling with random maps and approximate Hessian (dashed) and finite difference Hessian (solid). Blue: implicit sampling with linear maps and approximate Hessian (dashed) and finite difference Hessian (solid).

With implicit sampling we observe a faster convergence, in the sense that the approximated conditional mean does not change significantly with the number of samples. In fact, about 10^2 samples are sufficient for accurate estimates of the conditional mean. As a reference solution, we also show results we obtained with implicit sampling (with both random and linear maps) for which we used a Hessian computed with finite differences (rather than with the approximation in equation (17)).

The cost per sample of implicit sampling and the cost per step of Metropolis MCMC are different, and a fair comparison of these methods should take these costs into account. In particular, the off-set cost of the minimization and computation of the Hessian, required for implicit sampling must be accounted for. We measure the cost of the algorithms by the number of forward solves required. The results are shown for the parameters θ_1, θ_2 and θ_3 in the right panels of figure 4.

We find that the fast convergence of implicit sampling makes up for the relatively large a priori cost (for minimization and Hessian computations). In fact, the figure suggests that the random methods requires only a few hundred samples, whereas Metropolis MCMC and the random map method require thousands of samples. The convergence of Metropolis MCMC can perhaps be increased by further tuning, or by choosing a more advanced transition density. Implicit sampling on the other hand requires little tuning other than deciding on standard tolerances for the optimization. Moreover, implicit sampling generates independent samples with a known distribution, so that issues such as determining burn-in times, auto-correlation times and acceptance ratios, do not arise. It should also be mentioned that implicit sampling is easy to parallelize. Parallelizing Metropolis MCMC on the other hand is not trivial, because it is a sequential technique.

Finally, we discuss connections of our proposed implicit sampling methods to stochastic Newton MCMC [28]. In stochastic Newton one first finds the MAP point (as in implicit sampling or LMAP) and then starts a number of MCMC chains from the MAP point. The transition probabilities are based on local information about F and make use of the Hessian of F, evaluated at the location of the chain. Thus, at each step, a Hessian computation is required which, with our finite difference scheme, requires 31 forward solves (see above) and, therefore, is expensive (compared to generating samples with implicit sampling, which requires computing the Hessian only once). Second-order adjoints (if they were available) do not reduce that cost significantly. We have experimented with stochastic Newton in our example and have used 10–50 chains and taking about 200 steps per chain. Without significant tuning, we find acceptance rates of only a few percent, leading to a slow convergence of the method. We also observe that the Hessian may not be positive definite at all locations of the chain and, therefore, can not be used for a local Gaussian transition probability. In summary, we find that stochastic Newton MCMC is impractical unless second order adjoints are available to speed up the Hessian computations. Variations of stochastic Newton were explained and compared to each other in [37].

4. Conclusions

We explained how to use implicit sampling to estimate the parameters in PDE from sparse and noisy data. The idea in implicit sampling is to find the most likely state, often called the maximum a posteriori (MAP) point, and generate samples that explore the neighborhood of the MAP point. This strategy can work well if the posterior probability mass localizes around the MAP point, which is often the case when the data constrain the parameters. We discussed how to implement these ideas efficiently in the context of parameter estimation problems using multiple grids and adjoints to speed up the required optimization.

Our implicit sampling approach has the advantage that it generates independent samples, so that issues connected with MCMC, e.g. estimation of burn-in times, auto-correlations of the samples, or tuning of acceptance ratios, are avoided. Our approach is also fully nonlinear and captures non-Gaussian features of the posterior (unlike linear methods such as the linearization about the MAP point) and is easy to parallelize.

We illustrated the efficiency of our approach in numerical experiments with an elliptic inverse problem that is of importance in applications to reservoir simulation/management and pollution modeling. The elliptic forward model is discretized using finite elements, and the linear equations are solved by balancing domain decomposition by constraints. The optimization required by implicit sampling is done with a BFGS method coupled to an adjoint code. We use the fact that the solutions are expected to be smooth for model order reduction based on Karhunan-Loève expansions, and

found that our implicit sampling approach can exploit this low-dimensional structure. Moreover, implicit sampling is about an order of magnitude faster than Metropolis MCMC sampling (in the example we consider). We also discussed connections and differences of our approach with linear/ Gaussian methods, such as linearization about the MAP, and with stochastic Newton MCMC methods. In particular, one can build an implicit sampling code starting from a MAP code by simply adding the Gaussian sampling and weighting step. At the cost of one additional forward solve per sample, the implicit sampling approach can reveal non-Gaussian features.

Acknowledgements

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under contract DE-AC02005CH11231, and by the National Science Foundation under grant DMS-0955078, DMS-1115759, DMS-1217065, and DMS-1419069.

References

[1] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/ non-Gaussian Bayesian tracking *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[2] E. Atkins, M. Morzfeld, and A.J. Chorin. Implicit particle methods and their connection with variational data assimilation. *Monthly Weather Review*, 141(6):1786–1803, 2013.

[3] J. Bear. *Modeling groundwater flow and pollution*. Kluwer, 1990.

[4] P. Bickel, B. Li, and T. Bengtsson. Sharp failure rates for the bootstrap particle filter in high dimensions. *IMS Collections: Pushing the Limits of Contemporary Statistics: Contributions in Honor of Jayanta K. Ghosh*, 3:318–329, 2008.

[5] T. Bengtsson, P. Bickel, and B. Li. Curse of dimensionality revisited: the collapse of importance sampling in very large scale systems. *MS Collections: Probability and Statistics: Essays in Honor of David A. Freedman*, 2:316–334, 2008.

[6] 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 J. Sci. Comput.*, 35(6):A2494-A2523, 2013.

[7] D. Braess. Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics. Cambridge University Press, 1997.

[8] A.J. Chorin and O.H. Hald. *Stochastic Tools in Mathematics and Science*. Springer, third edition, 2013.

[9] A.J. Chorin, M. Morzfeld, and X. Tu. Implicit particle filters for data assimilation. *Communications in Applied Mathematics Computational Sciences*, 5:221–240, 2010.

[10] A.J. Chorin, M. Morzfeld, and X. Tu. Implicit sampling, with applications to data assimilation. *Chinese Annals of Mathematics*, 34B:89–98, 2013.

[11] A.J. Chorin and M. Morzfeld. Condition for successful data assimilation. *Journal of Geophysical Research*, 118(20):11522–11533, 2013.

[12] A.J. Chorin and X. Tu. Implicit sampling for particle filters. *Proceedings of the National Academy Sciences USA*, 106:17249–17254, 2009.

[13] M. Dashti and A.M. Stuart. Uncertainty quantification and weak approximation of an elliptic inverse problem. *SIAM Journal on Numerical Analysis*, 49(6):2524–2542, 2011.

[14] C. R. Dohrmann. A preconditioner for substructuring based on constrained energy minimization. *SIAM Journal on Scientific Computing*, 25:246–258, 2003.

[15] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10:197–208, 2000.

[16] Y. Efendiev, T. Hou, and W. Luo. Preconditioning Markov chain Monte Carlo simulations using coarse-scale models. *SIAM Journal on Scientific Computing*, 28(2):776–803 (electronic), 2006.

[17] R.P. Fedorenko. A relaxation method for solving elliptic difference equations. *USSR Computational Mathematics and Mathematical Physics*,1, 1961.

[18] R. Ghanem and P. Spanos. *Stochastic Finite Elements: A Spectral Approach*. Dover, 2003.

[19] J. Goodman and A.D. Sokal. Multigrid Monte Carlo method. Conceptual foundations. *Physical Review D*, 40:2035–2071, 1989.

[20] J. Goodman and J. Weare. Small-noise analysis and symmetrization of implicit Monte Carlo samplers. *Communications on Pure and Applied Mathematics*, accepted, 2015.

[21] J. Goodman and J. Weare. Ensemble samplers with affine invariance. *Communications in Applied Mathematics and Computational Science*, 5(1):65–80, 2010.

[22] M. Hinze, R. Pinnau, M. Ulrbich, and S. Ulbrich. *Optimization with PDE Constraints*. Springer, 2009.

[23] M.A. Iglesias, K.J.H. Law and A.M. Stuart. Evaluation of Gaussian approximations for data assimilation in reservoir models. *Computational Geosciences*, 17: 851-885, 2013.

[24] M. Kalos and P. Whitlock. *Monte Carlo methods, volume 1*. John Wiley & Sons, 1 edition, 1986.

[25] O.P. LeMaitre and O.M. Knio. *Spectral Methods for Uncertainty Quantification: with Applications to Computational Fluid Dynamics*. Springer, 2010.

[26] J.S. Liu. *Monte Carlo Strategies for Scientific Computing*. Springer, 2008.

[27] J.S. Liu and R. Chen. Blind Deconvolution via Sequential Imputations. *Journal of the American Statistical Association*, 90(430):567–576, 1995.

[28] 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:A1460–A1487, 2012.

[29] Y.M. Marzouk, H.N. Najm, and L.A. Rahn. Stochastic spectral methods for efficient Bayesian solution of inverse problems. *Journal of Computational Physics*, 224(2):560–586, 2007.

[30] M. Morzfeld and A.J. Chorin. Implicit particle filtering for models with partial noise, and an application to geomagnetic data assimilation. *Nonlinear Processes in Geophysics*, 19:365–382, 2012.

[31] M. Morzfeld, X. Tu, E. Atkins, and A.J. Chorin. A random map implementation of implicit filters. *Journal of Computational Physics*, 231(4):2049–2066, 2012.

[32] T.A. Moselhy and Y.M. Marzouk. Bayesian inference with optimal maps. *Journal of Computational Physics*, 231:7815–7850, 2012.

[33] J. Nocedal and S.T. Wright. *Numerical Optimization*. Springer, second edition, 2006.

[34] D.S. Oliver. Minimization for conditional simulation. *Journal of Computational Physics*, 265:1–15, 2014.

[35] D.S. Oliver, A.C. Reynolds, and N. Liu. *Inverse theory for petroleum reservoir characterization and history matching*. Cambridge University Press, 2008.

[36] D.S. Oliver and Y. Chen. Recent progress on reservoir history matching: a review. *Computers and Geosciences*, 15, 185–221, 2011.

[37] 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(4):A1525-A1555, 2014.

[38] C.E. Rasmussen and C.K.I. Wiliams. *Gaussian processes for machine learning*. MIT Press, 2006.

[39] N. Recca. A new methodology for importance sampling. *Masters Thesis*, Courant Institute of Mathematical Sciences, New York University.

[40] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson. Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136:4629–4640, 2008.

[41] A. M. Stuart. Inverse problems: a Bayesian perspective. *Acta Numerica*, 19:451–559, 2010.

[42] E. Vanden-Eijnden and J. Weare. Data assimilation in the low-noise, accurate observation regime with application to the Kuroshio current. *Monthly Weather Review*, 141:1822–1841, 2012.

[43] B. Weir, R.N. Miller and Y.H. Spitz. Implicit estimation of ecological model parameters. *Bulletin of Mathematical Biology*, 75:223–257, 2013.

[44] V.S. Zaritskii and L.I. Shimelevich. Monte Carlo technique in problems of optimal data processing. *Automation and Remote Control*, 12:95–103, 1975.