Computers & Fluids 100 (2014) 291–307

Contents lists available at ScienceDirect

Computers & Fluids

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

Adjoint-accelerated statistical and deterministic inversion of atmospheric contaminant transport

Devina Sanjaya [a], Krzysztof Fidkowski [a,*], Ian Tobasco [b]

[a] Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, United States
[b] Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, United States

ARTICLE INFO

Article history:
Received 17 August 2012
Received in revised form 25 February 2014
Accepted 17 May 2014
Available online 29 May 2014

Keywords:
Inverse problems
Bayesian inference
Adjoint method
Markov chain Monte Carlo

ABSTRACT

In this work we present and compare deterministic and statistical algorithms for efficiently solving large-scale contaminant source inversion problems. The underlying equations of contaminant transport are assumed linear but unsteady and defined over complex geometries. The algorithms presented are accelerated through discrete adjoint solutions that are pre-computed efficiently in an offline stage, yielding savings in the time-critical online stage of several orders of magnitude in computational time. In the deterministic case, adjoints accelerate the application of the Hessian matrix, while in the statistical case, adjoints are used to directly evaluate samples. To address deterioration of statistical sampling efficiency for anisotropic posteriors, we present an application of a recently developed ensemble Markov chain Monte Carlo method. Results for two- and three-dimensional problems demonstrate the feasibility of statistical inversion for large-scale problems and show the advantage of statistical results over single-point deterministic results.

1. Introduction

Real-time modeling and inversion of contaminant release events is crucial in applications ranging from environmental safety monitoring to homeland security. Such events include unintentional industrial or transport accidents, as well as intentional biological or chemical attacks in urban environments. A successful response to these scenarios requires rapid and accurate identification of the extent of the contamination, inversion of the data to find the source of the release, and prediction of the subsequent path of contamination for evacuation and countermeasures. In this work we address the problem of inversion of the contaminant source for large-scale calculations under some simplifying assumptions.

The contaminant source inversion problem is inherently complex: the geometry is often intricate, the flow conditions are uncertain, and the available measurements are limited and noisy. Computationally, the evolution of the contaminant spread is characterized by a system of partial differential equations that must be discretized on complex geometries, resulting in millions of unknowns. Moreover, the problem is generally ill-conditioned in the sense that small changes in the outputs can cause large changes in the calculated inputs [1]. This ill-conditioning makes single-point deterministic calculations, ones that seek the “best” possible answer, not robust, where robustness is measured by the level of risk associated with using computation in a broader context such as design or decision making. The lack of robustness is due to the fact that the “best” input may be only one of many different inputs that produce nearly the same outputs, especially when measurement error is taken into account.

A statistical approach to the inverse problem, in which probability distributions instead of single-point estimates are calculated for the inputs, can be more robust. Probability distributions convey much more information than a most likely value, and they allow for informed decisions that make full use of the available data and associated uncertainties. Statistical approaches often characterize the probability distributions through sampling, which requires numerous forward simulations. However, sampling can be expensive, especially when each forward simulation already taxes computational resources. Thus, for large-scale contaminant inversions, sampling-based statistical approaches quickly become prohibitively expensive.

Multiple previous studies have investigated large-scale inverse problems [2,3], including the societally-relevant application of contaminant transport [4–9]. The high computational cost of practical contaminant transport simulations prevents their direct use for inversion calculations during real-time events. Reducing this

* Corresponding author. Tel.: +1 7346157247.
E-mail address: kfid@umich.edu (K. Fidkowski).

http://dx.doi.org/10.1016/j.compfluid.2014.05.021
0045-7930/© 2014 Elsevier Ltd. All rights reserved.

cost has motivated research into inversion approaches that employ approximate solution techniques, such as grid coarsening [2,10], reduced-order modeling [9,11], and stochastic expansions [12], but the utility and accuracy of these approaches for generating real-time results in practical, large-scale simulations is yet to be demonstrated.

To some extent, previous studies have also considered applying adjoint solutions, or at least output gradients, to uncertainty quantification algorithms. These include gradient-enhanced response surface construction via least-squares [13] and gradient-enhanced Kriging [14]; although presented in the context of optimization, these ideas extend to propagation of input uncertainties [15]. For statistical inverse problems, the use of output gradients has been proposed to improve the acceptance ratio in Markov chain Monte Carlo sampling of anisotropic posteriors [16]. In recent work, these gradients are computed using finite differences based on the large-scale model [11] and on cheaper coarse-scale models [17].

2. Forward problem

The forward problem consists of determining system output quantities through the solution of governing equations given prescribed inputs. In the present case of contaminant transport, the inputs are parameters that describe the initial condition and the outputs are contaminant concentration measurements at various points in space and time.

2.1. Governing equations

Transport of the scalar contaminant field is assumed to be governed by a convection–diffusion equation,

$$\frac{\partial u}{\partial t} + \nabla \cdot (\vec{V} u) - \nu \nabla^2 u = 0, \tag{1}$$

$$u(\vec{x}, 0) = u_0(\vec{x}), \tag{2}$$

$$y_k = u(\vec{x}_k, t_k), \quad 1 \leqslant k \leqslant n_{\text{out}} \tag{3}$$

where $u(\vec{x}, t)$ is the contaminant concentration, $\vec{V}(\vec{x}, t)$ is the velocity field, ν is the diffusion coefficient, $t = 0$ is the time of release, $u_0(\vec{x})$ is the initial condition, and y_k are scalar outputs. The outputs are calculated at spatial positions $\vec{x}_k$ and times t_k. We allow for multiple sensors to be located at the same position $\vec{x}_k$ and for multiple outputs to be recorded at the same time t_k. In solving the forward problem, (1), $\vec{V}$, ν, and $u_0(\vec{x})$ are assumed known, and of interest are the outputs y_k, which require the primal solution $u(\vec{x}, t)$ for $t > 0$.

2.2. Discretization

We discretize (1) using a high-order discontinuous Galerkin (DG) finite element method. DG is chosen for this problem because it offers several attractive features including straightforward extension to high order interpolation, a compact stencil, and stability for convection-dominated flows. Details of the DG spatial discretization can be found in previous works [11,18], and here we only mention that we use pure upwinding for the inviscid flux and the second form of Bassi and Rebay (BR2) [19] for the viscous flux. Order p polynomials are used for the spatial approximation on each element, and in time we employ second-order backward differencing (BDF2) with BDF1 on the first time step. Once discretized, (1) takes the following form,

$$\mathbf{R}^i(\mathbf{u}^j) \equiv \mathbf{M}\left(\frac{3}{2}\mathbf{u}^{j+1} - 2\mathbf{u}^j + \frac{1}{2}\mathbf{u}^{j-1}\right) + \mathbf{R}_s(\mathbf{u}^{j+1}) = 0, \tag{4}$$

where $\mathbf{u}^j \in \mathbb{R}^N$ is the discrete state vector at time $t_j, 1 \leqslant j \leqslant N_T, \mathbf{u}^0$ is the discrete initial condition, $\mathbf{M} \in \mathbb{R}^{N \times N}$ is the block-diagonal mass matrix, $\mathbf{R}^i \in \mathbb{R}^N$ is an unsteady residual vector at time $t_i, 1 \leqslant i \leqslant N_T, \mathbf{R}_s \in \mathbb{R}^N$ is the spatial residual, and N is the number of spatial degrees of freedom.

We denote by $\mathbf{u}$, without any superscripts, all the unknowns in the entire time history rolled into one vector, i.e. $\mathbf{u} \in \mathbb{R}^{N \cdot N_t}$. The system in (4) can then be written as

$$\mathbf{A}\mathbf{u} = \mathbf{F}\mathbf{u}^0, \tag{5}$$

where $\mathbf{F}$ contains the dependence of the unsteady residual on the initial condition, only nonzero for the first two unsteady residuals, and $\mathbf{A}$ is a sparse constant matrix that contains the dependence of the unsteady residual on the state. The outputs in (3) can also be expressed in terms of the discrete state vector according to

$$\mathbf{y} = \mathbf{C}\mathbf{u} = \mathbf{C}\mathbf{A}^{-1}\mathbf{F}\mathbf{u}^0, \tag{6}$$

where $\mathbf{y} \in \mathbb{R}^{n_{\text{out}}}$ and where the matrix $\mathbf{C}$ consists of the spatial approximation functions evaluated at the desired sensor locations and time nodes.

2.3. Initial and boundary conditions

We assume a spatially-Gaussian distribution for the contaminant concentration at $t = 0, u_0(\vec{x})$. Thus the initial condition is described by three parameters: the distribution center, the standard deviation, and the amplitude. These parameters are rolled into one vector, $\boldsymbol{\mu} \in \mathbb{R}^{n_{\text{par}}}$, where n_{par} is at most 4 in two dimensions and 5 in three dimensions. This simplified model is relevant for many single-point release scenarios and allows us to compare statistical and deterministic inversion approaches.

No conditions are imposed at outflow boundaries of the domain, while $u = 0$ is imposed at inflow boundaries. Our cases will include geometrical objects simulating buildings, and on their boundaries a zero flux condition is imposed. Finally, we assume a spatially-varying but temporally constant velocity field, $\vec{V}(\vec{x})$, derived from potential flow. Specifically, we solve Laplace's equation for the velocity potential, $\phi(\vec{x})$, approximated with $p = 4$ polynomials in space on the same mesh. We then differentiate the potential element-wise to obtain the velocity field, $\vec{V} = \nabla \phi$ on each element.

3. Inverse problem

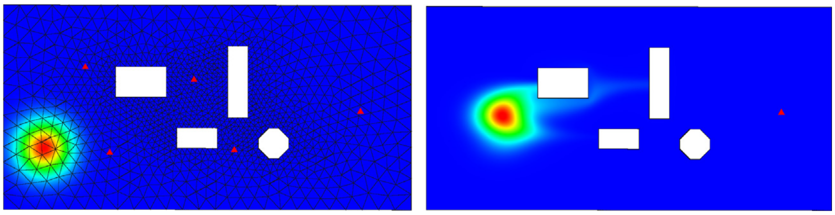
Whereas the forward problem concerns calculation of outputs of a system for given inputs, the inverse problem reverses this relationship and seeks unknown inputs from measured outputs. In the present contaminant transport problem, the inputs $\boldsymbol{\mu}$ are taken as parameters that describe the initial contaminant distribution, and the observed outputs $\bar{\mathbf{y}}$ consist of noisy contaminant concentration measurements at a limited number of sensors. In this section we present two approaches for obtaining $\boldsymbol{\mu}$ given $\bar{\mathbf{y}}$: one deterministic and one statistical.

3.1. Deterministic solution

The inverse problem can be formulated as a deterministic optimization problem of minimizing a cost function that incorporates the error between observed and simulated outputs along with a regularization term that is chosen to penalize certain features of the inputs:

$$\boldsymbol{\mu}^* = \arg\min_{\boldsymbol{\mu}} \mathcal{J}(\boldsymbol{\mu}), \quad \mathcal{J}(\boldsymbol{\mu}) = (\mathbf{y}(\boldsymbol{\mu}) - \bar{\mathbf{y}})^T(\mathbf{y}(\boldsymbol{\mu}) - \bar{\mathbf{y}}) + T(\mathbf{u}^0(\boldsymbol{\mu})). \tag{7}$$

The expression $\mathbf{y}(\boldsymbol{\mu})$ is shorthand for the chain of operations expressed in (6). $T(\boldsymbol{\mu})$ is a regularization term that alleviates ill-conditioning of the inverse problem – the use of parametrized initial



(a) Domain, mesh, outputs, and initial condition

(b) Sample adjoint solution

Fig. 1. Setup for the two-dimensional contaminant transport problem, showing the computational domain, mesh, sensor locations (red triangles), and contours of the initial condition. The prevailing advection velocity is from left to right. Shown on the right is a sample $t = 0$ adjoint solution for one of the outputs, indicated by the triangle in the figure. The red contours represent areas for which the output is most sensitive to $t = 0$ residual source perturbations, e.g. those arising from initial conditions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

conditions alleviates but does not eliminate the possibility of (7) being ill-posed. In this work we use continuous Tikhonov regularization, which effectively penalizes large amounts of initial contaminant,

$$T(\mathbf{u}^0) = \beta_M (\mathbf{u}^0)^T \mathbf{M}\mathbf{u}^0. \tag{8}$$

β_M is a heuristic constant that we tuned for each case by choosing the smallest value that still produced a well-conditioned system.

We solve (7) by seeking local optima of $\mathcal{J}(\boldsymbol{\mu})$ via a Newton–Raphson method starting from several initial guesses for $\boldsymbol{\mu}$. We note that the nonlinear nature of the optimization problem arises here solely from the functional dependence of $\mathbf{u}^0$ on $\boldsymbol{\mu}$. Differentiating $\mathcal{J}(\boldsymbol{\mu})$ with respect to $\boldsymbol{\mu}$ and setting the result equal to zero yields,

$$\mathbf{R}^{\mu} \equiv \frac{\partial \mathbf{u}^0}{\partial \boldsymbol{\mu}} \left[\mathbf{H}\mathbf{u}^0(\boldsymbol{\mu}) - \mathbf{F}^T\mathbf{A}^{-T}\mathbf{C}^T\bar{\mathbf{y}} \right] = 0, \tag{9}$$

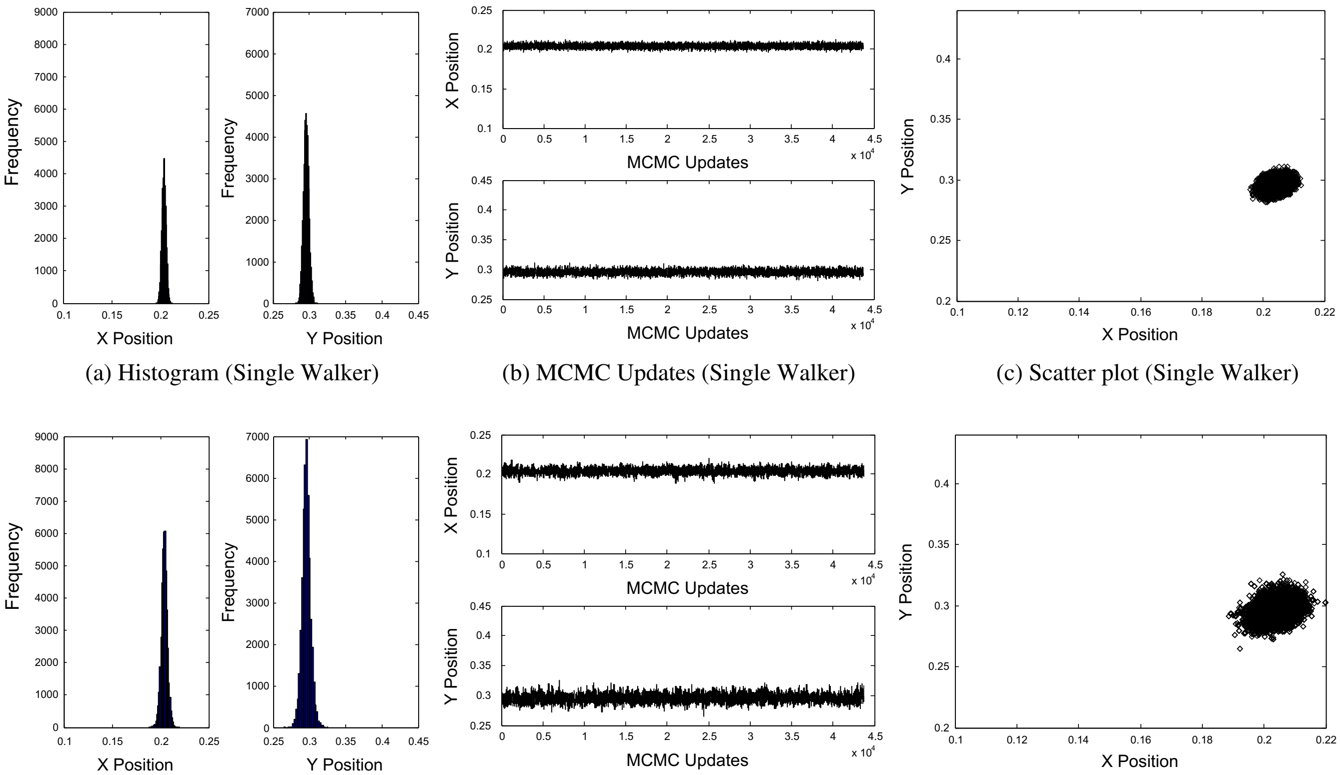
where the Hessian matrix, $\mathbf{H}$, is

$$\mathbf{H} \equiv \mathbf{F}^T\mathbf{A}^{-T}\mathbf{C}^T\mathbf{C}\mathbf{A}^{-1}\mathbf{F} + \beta_M\mathbf{M}. \tag{10}$$

$\mathbf{H}$ is not computed explicitly; rather only its action on a vector is computed through forward and adjoint solves. The Newton–Raphson method is applied to (9). The required Jacobian matrix,

$$\frac{\partial \mathbf{R}^{\mu}}{\partial \boldsymbol{\mu}} = \left(\frac{\partial \mathbf{u}^0}{\partial \boldsymbol{\mu}}\right)^T \mathbf{H} \frac{\partial \mathbf{u}^0}{\partial \boldsymbol{\mu}},$$

is computed by applying the Hessian matrix to the n_{par} vectors $\frac{\partial \mathbf{u}^0}{\partial \boldsymbol{\mu}}$ and then taking inner products with these same vectors. For robustness, Newton updates are limited to a fraction $f^{\text{update}} = 0.2$ of the a priori parameter domain extent. Finally, to account for the possibility of local optima, we solve the deterministic optimization problem a large number of times (500 in the results presented here) with randomly chosen starting parameter guesses.



(a) Histogram (Single Walker)

(b) MCMC Updates (Single Walker)

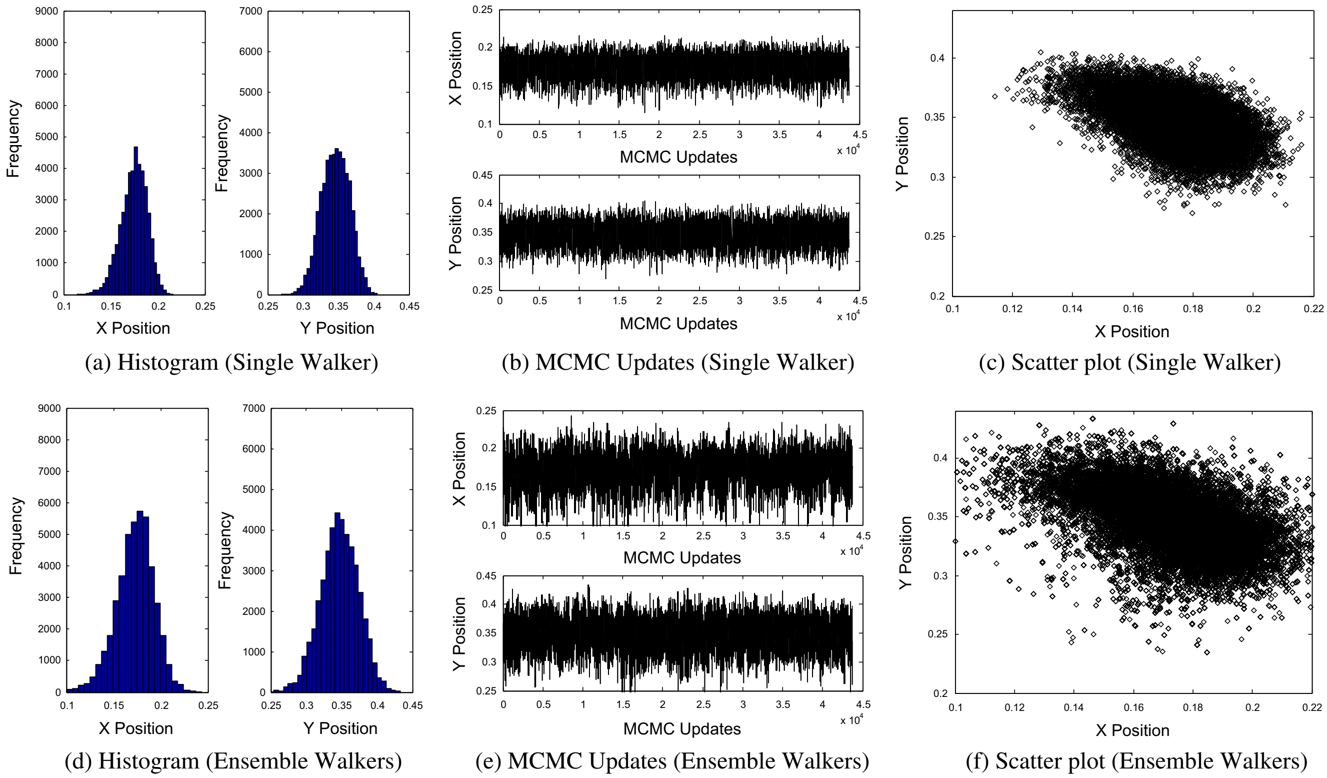
(c) Scatter plot (Single Walker)

(d) Histogram (Ensemble Walkers)

(e) MCMC Updates (Ensemble Walkers)

(f) Scatter plot (Ensemble Walkers)

Fig. 2. Statistical inversion results for the two-dimensional problem with an isotropic posterior, $\epsilon = 1 \times 10^{-2}$.



(a) Histogram (Single Walker) (b) MCMC Updates (Single Walker) (c) Scatter plot (Single Walker)

(d) Histogram (Ensemble Walkers) (e) MCMC Updates (Ensemble Walkers) (f) Scatter plot (Ensemble Walkers)

Fig. 3. Statistical inversion results for the two-dimensional problem with an isotropic posterior, $\epsilon = 5 \times 10^{-2}$.

3.2. Statistical solution

An important source of uncertainty in inverse calculations, especially in contaminant source inversion, is the uncertainty in the output measurements. Deterministic approaches that calculate best-fit inputs, such as the optimization method of Section 3.1, neglect this uncertainty. More robust in such situations are statistical approaches, which provide a probabilistic description of the inputs. The solution of the statistical inverse problem is a posterior joint probability density of the inputs, which reflects the degree of confidence in their values [20,21].

3.2.1. Posterior probability distribution

Statistical inversion relies on a known model for the measurement error. In this work we assume an additive error, $\boldsymbol{\varepsilon}$,

$$\bar{\mathbf{y}} = \mathbf{y}(\boldsymbol{\mu}^e) + \boldsymbol{\varepsilon}, \tag{11}$$

where $\boldsymbol{\mu}^e$ are the "exact" parameters that we could only hope to identify if there were no measurement errors. Furthermore, we assume that the errors are independent and normally distributed, with each component of $\boldsymbol{\varepsilon}$ being $N(0, \sigma^2)$, where $\sigma = \epsilon C_y$ and C_y is a reference contaminant concentration, here taken as unity. Values of ϵ are given in the results. With this error model, the likelihood of observing outputs $\bar{\mathbf{y}}$ given inputs $\boldsymbol{\mu}$ is

$$p(\bar{\mathbf{y}}|\boldsymbol{\mu}) \propto \exp\left[-\frac{1}{2\sigma^2}(\mathbf{y}(\boldsymbol{\mu}) - \bar{\mathbf{y}})^T(\mathbf{y}(\boldsymbol{\mu}) - \bar{\mathbf{y}})\right].$$

The posterior probability distribution, $p(\boldsymbol{\mu}|\bar{\mathbf{y}})$ is then obtained by Bayes' theorem,

$$p(\boldsymbol{\mu}|\bar{\mathbf{y}}) = p(\bar{\mathbf{y}}|\boldsymbol{\mu})\frac{p(\boldsymbol{\mu})}{p(\bar{\mathbf{y}})} \propto \begin{cases} \exp\left[-\frac{1}{2\sigma^2}(\mathbf{y}(\boldsymbol{\mu}) - \bar{\mathbf{y}})^T(\mathbf{y}(\boldsymbol{\mu}) - \bar{\mathbf{y}})\right] & \text{if } \boldsymbol{\mu} \in \mathcal{D} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where $\mathcal{D}$ is a parameter domain that is assumed a priori to contain the desired $\boldsymbol{\mu}$. Specifically, $p(\boldsymbol{\mu})$ is a prior distribution given by $p(\boldsymbol{\mu}) = \text{constant}$ for $\boldsymbol{\mu} \in \mathcal{D}$ and 0 otherwise.

3.2.2. Markov chain Monte Carlo (MCMC)

For practical problems the desired probability distribution cannot be found analytically, leaving numerical sampling as the next best alternative. In this work we use Markov chain Monte Carlo (MCMC) sampling, in which a chain of samples is constructed through a likelihood-based random walk in input space [22]. Denote by $\{\boldsymbol{\mu}^i\}_{i=1}^{N_m}$ a Markov chain of N_m parameter samples. Given a parameter sample $\boldsymbol{\mu} = \boldsymbol{\mu}^i$, the Metropolis–Hastings algorithm generates a new sample $\boldsymbol{\mu}^{i+1}$ by the following two steps:

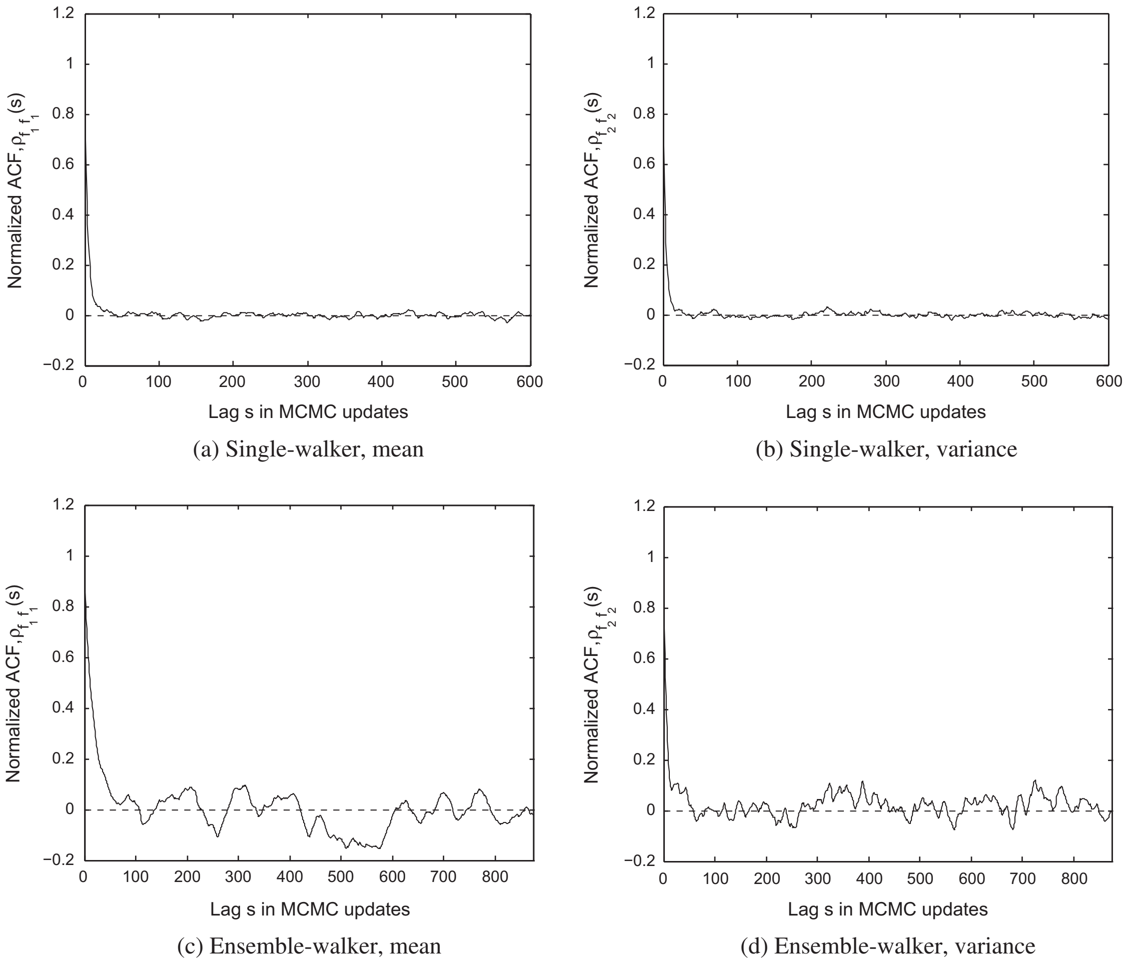
1. **Generation Step**: Sample $\boldsymbol{\mu}'$ from a proposal distribution, $q(\boldsymbol{\mu}'|\boldsymbol{\mu})$. Also calculate $q(\boldsymbol{\mu}|\boldsymbol{\mu}')$.
2. **Acceptance Step**: With probability

$$\alpha(\boldsymbol{\mu}'|\boldsymbol{\mu}) = \min\left[1, \frac{p(\boldsymbol{\mu}'|\bar{\mathbf{y}})q(\boldsymbol{\mu}'|\boldsymbol{\mu})}{p(\boldsymbol{\mu}|\bar{\mathbf{y}})q(\boldsymbol{\mu}|\boldsymbol{\mu}')}\right], \tag{13}$$

accept the proposal $(\boldsymbol{\mu}^{i+1} = \boldsymbol{\mu}')$. Otherwise, reject the proposal $(\boldsymbol{\mu}^{i+1} = \boldsymbol{\mu})$.

Note that if the proposal $\boldsymbol{\mu}'$ falls outside the parameter domain, $p(\boldsymbol{\mu}'|\bar{\mathbf{y}}) = 0$, and $\boldsymbol{\mu}'$ is immediately rejected. Otherwise, the calculation of the posterior $p(\boldsymbol{\mu}'|\bar{\mathbf{y}})$ requires a forward solve to determine $\mathbf{y}(\boldsymbol{\mu}')$. This forward solve at every step is the most expensive part of the MCMC algorithm, often making MCMC impractical for problems governed by large-scale models.

In this work we use a hypercube of dimensions $\boldsymbol{\Delta} \in \mathbb{R}^{n_{\text{par}}}$ to define our proposal distribution: $q(\boldsymbol{\mu}'|\boldsymbol{\mu})$ is a constant for $\boldsymbol{\mu}'$ inside the hypercube centered around $\boldsymbol{\mu}$ and 0 otherwise. The hypercube dimensions are taken simply as the parameter domain dimensions



(a) Single-walker, mean

(b) Single-walker, variance

(c) Ensemble-walker, mean

(d) Ensemble-walker, variance

Fig. 4. Normalized autocovariance functions for mean and variance estimators of the *X* position for the two-dimensional problem with an isotropic posterior and $\epsilon = 5 \times 10^{-2}$.

scaled by a factor $f^{\text{prop}} < 1$. In our results we tune f^{prop} to yield an MCMC acceptance ratio of approximately 30% [21].

3.2.3. Ensemble-walker MCMC

We refer to the algorithm presented in Section 3.2.2 as a “single-walker” MCMC, as only one sample is generated per MCMC iteration. One problem with this algorithm is that it can be very inefficient when exploring posterior distributions that exhibit anisotropy, i.e. that are stretched in parameter space, specifically when using an isotropic proposal distribution. Prior work [11] presented an anisotropic proposal in the context of single-walker MCMC for addressing this problem. In this work we employ an alternative method that we have observed to be more robust and efficient for anisotropic and curved posterior distributions. This method, introduced in [23], relies on an *ensemble* of walkers exploring the parameters space in a coordinated fashion.

Specifically, we use an ensemble-MCMC algorithm with a “stretch” move. In this algorithm, L walkers, $\boldsymbol{\mu}_l, 1 \leqslant l \leqslant L$, explore the parameter space. During each MCMC iteration one walker, $\boldsymbol{\mu}_j$ is the pivot and does not move. The other $L-1$ walkers, indexed by $l \neq j$, each consider the following proposed step about the pivot,

$$\boldsymbol{\mu}'_l = \boldsymbol{\mu}_j + Z_l(\boldsymbol{\mu}_l - \boldsymbol{\mu}_j) \tag{14}$$

Table 1
Comparison of deterministic and statistical inversion results for the two-dimensional problem with an isotropic posterior. The true initial position is $(X, Y) = (0.2, 0.3)$. Note that μ_X and μ_Y are mean-values and σ_X and σ_Y are standard deviations of X and Y.

ϵ	Value	Deterministic	Single walker	Ensemble walkers
1×10^{-2}	μ_X	0.2041	$0.2041 \pm 2.85 \times 10^{-5}$	$0.2040 \pm 9.96 \times 10^{-5}$
	σ_X	–	$0.0022 \pm 8.21 \times 10^{-8}$	$0.0033 \pm 4.46 \times 10^{-7}$
	μ_Y	0.2956	$0.2958 \pm 6.44 \times 10^{-5}$	$0.2958 \pm 1.62 \times 10^{-4}$
	σ_Y	–	$0.0037 \pm 2.91 \times 10^{-7}$	$0.0059 \pm 1.24 \times 10^{-6}$
	Acceptance ratio		0.314	0.314
5×10^{-2}	μ_X	0.1775	$0.1748 \pm 2.17 \times 10^{-4}$	$0.1729 \pm 6.67 \times 10^{-4}$
	σ_X	–	$0.0137 \pm 3.98 \times 10^{-6}$	$0.0201 \pm 1.54 \times 10^{-5}$
	μ_Y	0.3443	$0.3459 \pm 3.66 \times 10^{-4}$	$0.3470 \pm 2.33 \times 10^{-4}$
	σ_Y	–	$0.0200 \pm 7.52 \times 10^{-6}$	$0.0268 \pm 2.20 \times 10^{-5}$
	Acceptance ratio		0.338	0.321

Table 2
Computational time consumed by different components of the solver for the two-dimensional test case. The computational times have been normalized by the forward solve time. We expect the forward and adjoint solve times to be comparable for this linear problem – the larger forward solve time is due to overhead associated with a solver that is generally-applicable to nonlinear problems.

Forward solve	Adjoint solve	Adjoint-accelerated	
		Output calculation	Hessian application
1	0.36	3.7×10^{-5}	5.9×10^{-4}

where Z_l is a random number, independent for each walker, drawn from the probability distribution

$$p_z(z) \propto \begin{cases} \frac{1}{\sqrt{z}} & \text{if } z \in [1/a, a] \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

Here, a is a constant that we tune to obtain approximately 30% acceptance ratio. Each walker l accepts the proposal $\boldsymbol{\mu}'_l$ with probability

$$\alpha(\boldsymbol{\mu}'_l|\boldsymbol{\mu}_l) = \min\left[1, Z_l^{n_{\mathrm{par}}-1}\frac{p(\boldsymbol{\mu}'_l|\bar{\mathbf{y}})}{p(\boldsymbol{\mu}_l|\bar{\mathbf{y}})}\right]. \tag{16}$$

We denote by N_m^e the number of iterations taken by the ensemble MCMC algorithm. Note that the costs of the single and ensemble MCMC algorithms are approximately the same when $N_m = LN_m^e$.

3.2.4. Sampling statistics

For a large number of samples, we expect the above MCMC algorithms to converge to the underlying posterior probability distribution. In practice we will truncate the MCMC iterations at some reasonable number and calculate estimators of statistical quantities according to

$$\hat{f} = \frac{1}{N_m}\sum_{i=1}^{N_m} f(\boldsymbol{\mu}^i) \quad \text{(single walker)},$$

$$\hat{f} = \frac{1}{N_m^e}\sum_{i=1}^{N_m^e} \underbrace{\frac{1}{L}\sum\nolimits_{l=1}^{L} f(\boldsymbol{\mu}_l^i)}_{F(\vec{\boldsymbol{\mu}}^i)} \quad \text{(ensemble walkers)}.$$

In these expressions $f(\boldsymbol{\mu})$ is the kernel of the estimator, e.g. $f(\boldsymbol{\mu}) = \boldsymbol{\mu}$ for the mean. $F(\vec{\boldsymbol{\mu}}^i)$ is an estimator over the walkers at one MCMC iteration, $\vec{\boldsymbol{\mu}}^i = \{\boldsymbol{\mu}_l^i\}$. One measure of whether $\hat{f}$ is sufficiently converged with respect to different MCMC realizations is the estimator variance,

$$\mathrm{var}(\hat{f}) = \frac{\tau_f \mathrm{var}(f(\boldsymbol{\mu}^i))}{N_m} \quad \text{(single walker)}, \tag{17}$$

$$\mathrm{var}(\hat{f}) = \frac{\tau_f^e \mathrm{var}(F(\vec{\boldsymbol{\mu}}^i))}{N_m^e} \quad \text{(ensemble walkers)}, \tag{18}$$

where τ_f and τ_f^e are integrated autocorrelation times,

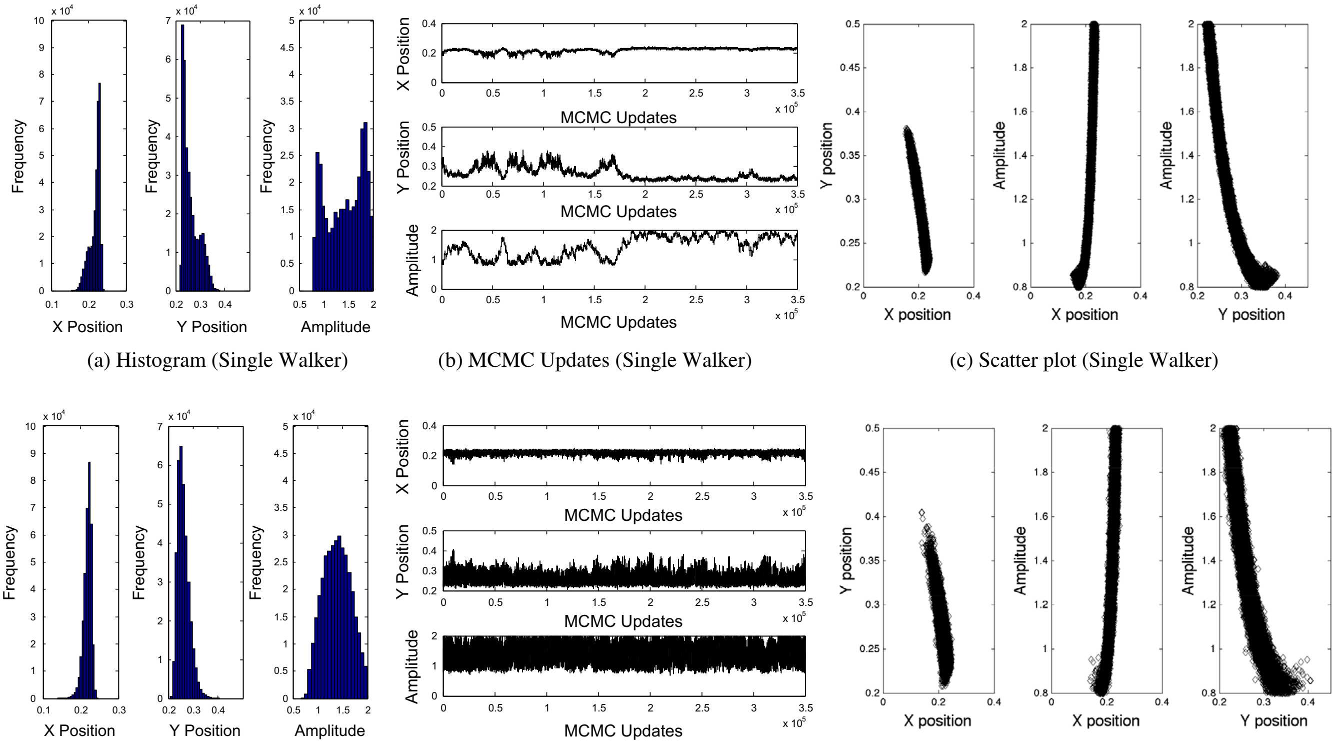
$$\tau_f = 1 + 2\sum_{s=1}^{\infty}\frac{\mathrm{cov}(f(\boldsymbol{\mu}^i), f(\boldsymbol{\mu}^{i+s}))}{\mathrm{var}(f(\boldsymbol{\mu}^i))}, \quad \text{(single walker)}, \tag{19}$$

$$\tau_f^e = 1 + 2\sum_{s=1}^{\infty}\frac{\mathrm{cov}(F(\vec{\boldsymbol{\mu}}^i), F(\vec{\boldsymbol{\mu}}^{i+s}))}{\mathrm{var}(F(\vec{\boldsymbol{\mu}}^i))}, \quad \text{(ensemble walkers)}. \tag{20}$$

In practice, we truncate the infinite sums to end at $s = \sqrt{N_m}$, or $s = \sqrt{N_m^e}$ in the ensemble case, to avoid noise in the covariance calculations for large s.

4. Adjoint acceleration

The MCMC algorithms presented in Section 3.2 for statistical inversion will require tens or hundreds of thousands of samples for accurate statistics. Since the calculation of the acceptance ratio



(a) Histogram (Single Walker)

(b) MCMC Updates (Single Walker)

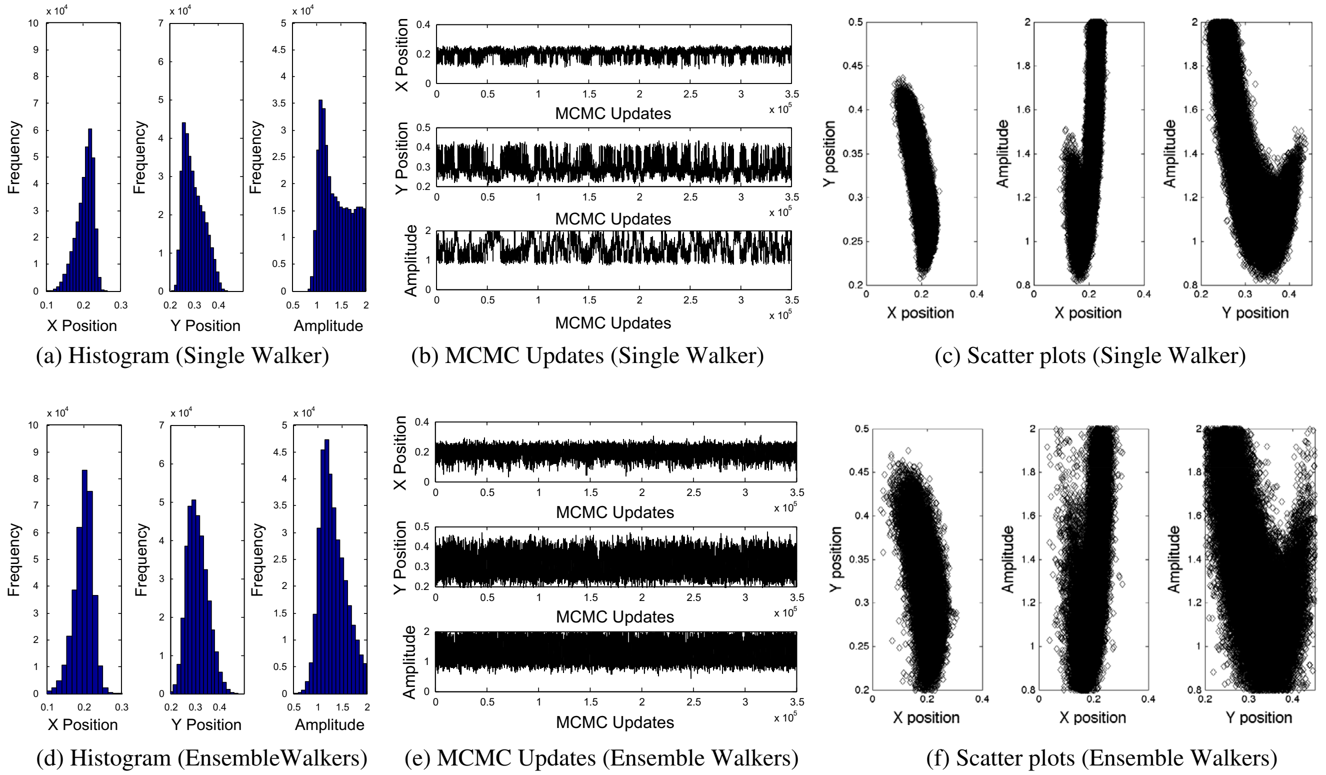
(c) Scatter plot (Single Walker)

(d) Histogram (Ensemble Walkers)

(e) MCMC Updates (Ensemble Walkers)

(f) Scatter plot (Ensemble Walkers)

Fig. 5. Statistical inversion results for the two-dimensional problem with an anisotropic posterior, $\epsilon = 1 \times 10^{-2}$.



(a) Histogram (Single Walker)

(b) MCMC Updates (Single Walker)

(c) Scatter plots (Single Walker)

(d) Histogram (EnsembleWalkers)

(e) MCMC Updates (Ensemble Walkers)

(f) Scatter plots (Ensemble Walkers)

Fig. 6. Statistical inversion results for the two-dimensional problem with an anisotropic posterior, $\epsilon = 5 \times 10^{-2}$.

for a sample requires output values, and hence individual forward solutions, the cost of performing MCMC directly will be prohibitive for large-scale problems. Prior work has addressed this issue with approximate output evaluations based on model reduction [11] or forward problem emulators. In this work we take advantage of the linearity of the forward problem to calculate the outputs exactly and efficiently using adjoint solutions. We note that adjoint solutions are already incorporated into the deterministic inversion through the presence of $\mathbf{A}^{-T}$ in the Hessian matrix.

Consider one of the outputs, y_k, which is a scalar contaminant measurement. Denoting by $\mathbf{c}_k$ the k^{th} row of $\mathbf{C}, y_k$ is given by

$$y_k = \underbrace{\mathbf{c}_k\mathbf{A}^{-1}}_{\psi_k^T}\mathbf{F}\mathbf{u}^0(\boldsymbol{\mu}) = \underbrace{\psi_k^T\mathbf{F}}_{\mathbf{s}_k^T}\mathbf{u}^0(\boldsymbol{\mu}) = \mathbf{s}_k^T\mathbf{u}^0(\boldsymbol{\mu}). \tag{21}$$

ψ_k is the unsteady adjoint solution corresponding to output y_k, and $\mathbf{s}_k$ is the associated output sensitivity vector. From (21), the adjoint is obtained by solving

$$\mathbf{A}^T\psi_k = \mathbf{c}_k^T, \tag{22}$$

which is a linear, unsteady equation. Due to the presence of the transpose on $\mathbf{A}$, the solution strategy for the adjoint equation is to march backwards in time. The forcing term, $\mathbf{c}_k^T$, is only nonzero at the time of the measurement reading, t_k, and hence the unsteady adjoint solution is marched from $t = t_k$ to $t = 0$. As the problem is linear, no state storage is required for the adjoint solution, which means that the cost of the adjoint is about the same as that of the forward solution. Finally, the initial-condition sensitivity vector for y_k is obtained by post-processing the adjoint according to $\mathbf{s}_k = \mathbf{F}^T\psi_k$.

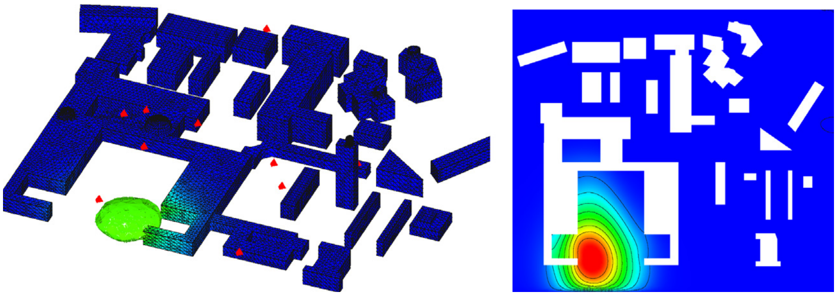
Once the sensitivity vectors $\mathbf{s}_k$ are calculated, the evaluation of an output for a given $\boldsymbol{\mu}$ is trivial. From (21), we see that y_k is just an inner product between the sensitivity vector and the initial condition obtained from $\boldsymbol{\mu}$, both of which are vectors of size N. By pre-computing the $\mathbf{s}_k$ for all outputs and using (21) for all output

Table 3
Comparison of integrated autocorrelation times for the two-dimensional problem with an anisotropic posterior.

ϵ	τ_f for	Single walker	Ensemble walkers
1×10^{-2}	μ_X	Unconverged	1.7303×10^3
	σ_X^2	Unconverged	186.8158
	μ_Y	Unconverged	1.8971×10^3
	σ_Y^2	Unconverged	198.0392
	μ_A	Unconverged	1.8492×10^3
	σ_A^2	Unconverged	124.5625
5×10^{-2}	μ_X	1.2338×10^3	312.9655
	σ_X^2	83.1260	55.8530
	μ_Y	1.4241×10^3	475.0969
	σ_Y^2	114.8945	66.2328
	μ_A	1.9566×10^3	329.9641
	σ_A^2	827.6326	77.5918

Table 4
Comparison of deterministic and statistical results for the two-dimensional problem with an anisotropic posterior. The true parameter values are $(X, Y, A) = (0.2, 0.3, 1.0)$.

ϵ	Parameter	Deterministic	Single walker	Ensemble walkers
1×10^{-2}	μ_X	0.173	–	$0.2181 \pm 7.38 \times 10^{-4}$
	σ_X	–	–	$0.0106 \pm 9.00 \times 10^{-6}$
	μ_Y	0.347	–	$0.2613 \pm 1.76 \times 10^{-3}$
	σ_Y	–	–	$0.0239 \pm 4.20 \times 10^{-5}$
	μ_A	0.787	–	$1.3854 \pm 2.00 \times 10^{-2}$
	σ_A	–	–	$0.2750 \pm 2.45 \times 10^{-3}$
	Acceptance ratio		0.317	0.286
5×10^{-2}	μ_X	0.156	$0.2049 \pm 1.40 \times 10^{-3}$	$0.1983 \pm 7.61 \times 10^{-4}$
	σ_X	–	$0.0229 \pm 1.21 \times 10^{-5}$	$0.0254 \pm 1.90 \times 10^{-5}$
	μ_Y	0.341	$0.2998 \pm 2.70 \times 10^{-3}$	$0.3104 \pm 1.54 \times 10^{-3}$
	σ_Y	–	$0.0419 \pm 3.79 \times 10^{-5}$	$0.0419 \pm 4.40 \times 10^{-5}$
	μ_A	0.717	$1.4021 \pm 2.29 \times 10^{-2}$	$1.3146 \pm 7.91 \times 10^{-3}$
	σ_A	–	$0.3057 \pm 4.20 \times 10^{-3}$	$0.2576 \pm 1.63 \times 10^{-3}$
	Acceptance ratio		0.294	0.304



(a) Domain, outputs, and initial condition

(b) Sample adjoint solution

Fig. 7. Setup for the three-dimensional contaminant transport problem, showing the computational domain, mesh, sensor locations (red triangles), and an iso-surface (40% of maximum value) of the initial condition. The prevailing velocity is to the up and right in the left diagram. Shown on the right is a $t = 0$ adjoint solution for the sensor closest to the main dome. The red contours represent areas for which the output is most sensitive to $t = 0$ residual source perturbations, e.g. effected through initial conditions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

evaluations, the MCMC algorithms become tractable for large-scale problems.

Furthermore, in the deterministic case, the adjoint solutions simplify the application of the Hessian matrix. Using (10) and (21), we have

$$\mathbf{H}\mathbf{v} = \underbrace{\mathbf{F}^T\mathbf{A}^{-T}\mathbf{C}^T}_{\boldsymbol{\Psi}}\underbrace{\mathbf{C}\mathbf{A}^{-1}\mathbf{F}}_{\mathbf{S}^T}\mathbf{v} + \beta_M\mathbf{M}\mathbf{v} = \sum_k \psi_k \mathbf{s}_k^T\mathbf{v} + \beta_M\mathbf{M}\mathbf{v}, \tag{23}$$

where $\boldsymbol{\Psi}$ is a matrix whose columns are ψ_k, and $\mathbf{S}$ is a matrix whose columns are $\mathbf{s}_k$. Therefore, application of the Hessian matrix reduces to a mass matrix multiplication combined with a weighted sum of adjoint vectors, where the weights are output values obtained with $\mathbf{v}$ as the initial condition.

5. Kriging forward emulator

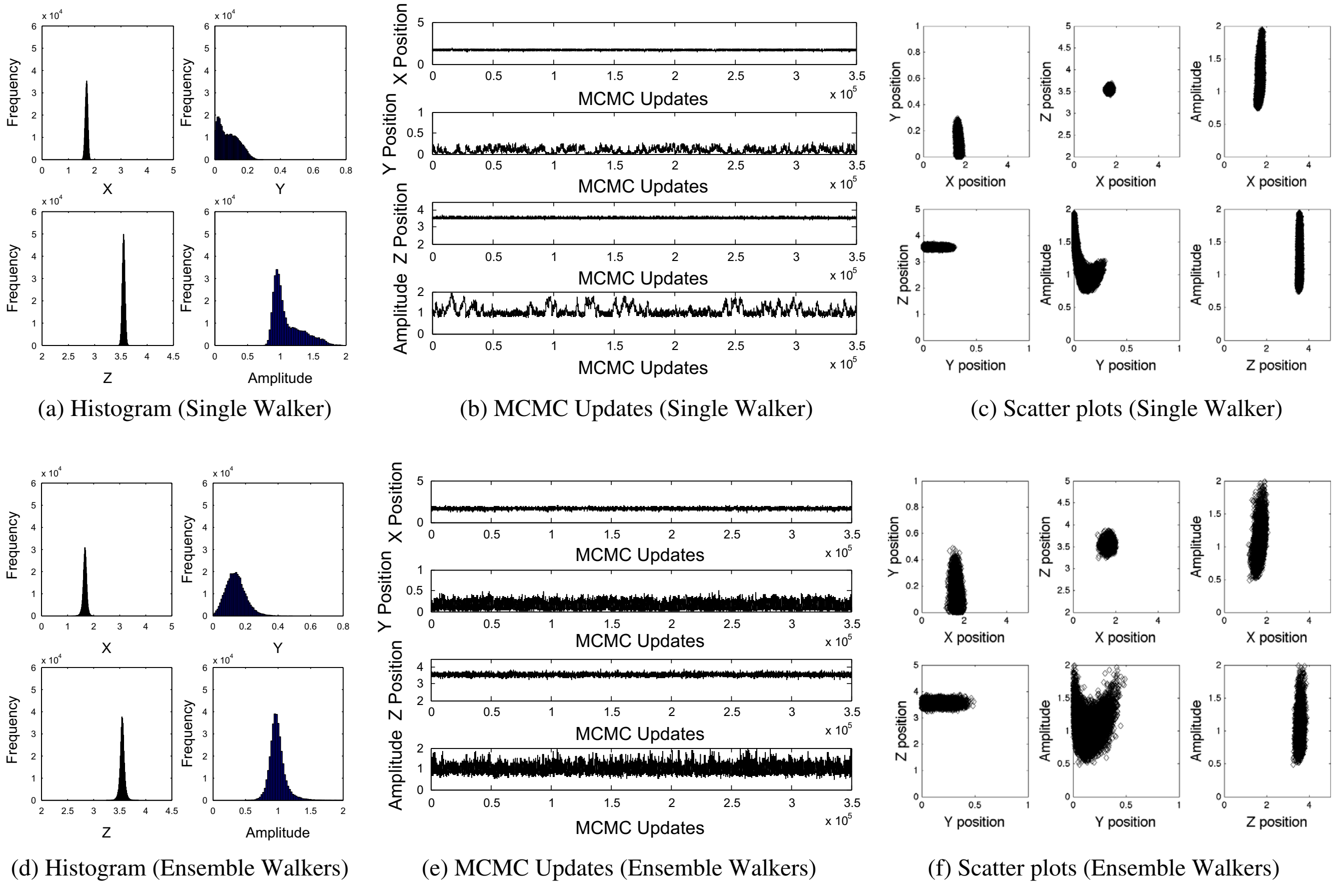
The adjoint acceleration of output calculations presented above is just one of many methods that can be applied to this atmospheric contaminant transport problem. Other methods, such as output response surrogates or polynomial chaos expansions, can also be used to solve the inverse problem. In this work, to provide a comparison, we also create a Kriging forward emulator to calculate the individual forward solutions during the MCMC process. Consider a design matrix $\mathbf{D} \in \mathbb{R}^{n_d \times n_{par}}$ and a test matrix $\mathbf{T} \in \mathbb{R}^{n_t \times n_{par}}$. We denote the number of design points by n_d and the number of test points by n_t. Here, we define a set of design points, $\{\boldsymbol{\mu}_i^d\}$ to be those parameter vectors at which the forward solution is accurately computed, and a set of test points, $\{\boldsymbol{\mu}_i^t\}$, to be the parameter vectors at which an approximate but quick forward solution is desired. During the MCMC process, the test matrix only consists of one test point since there is only one proposed point at a time, $n_t = 1$. The Kriging emulator is obtained by solving the following system,

$$\begin{bmatrix} \boldsymbol{G}^d & \boldsymbol{\Phi}^d \\ \boldsymbol{\Phi}^{d,T} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{g}(\boldsymbol{T}) \\ \boldsymbol{\phi}^t(\boldsymbol{T}) \end{bmatrix} \tag{24}$$

where $G_{ik}^d = \mathrm{Cov}(\boldsymbol{\mu}_i^d, \boldsymbol{\mu}_k^d; \ell)$ is the design kernel with correlation length ℓ, $\Phi_{ij}^d = \phi_j(\boldsymbol{\mu}_i^d)$ is the jth basis function evaluated at design point i, $g_{ik} = \mathrm{Cov}(\boldsymbol{\mu}_i^d, \boldsymbol{\mu}_k^t; l)$ is the test kernel with correlation length ℓ, $\phi_{ij}^t = \phi_j(\boldsymbol{\mu}_i^t)$ is the jth basis function evaluated at test point i, $\boldsymbol{w} \in \mathbb{R}^{n_d}$ is the weight vector, $\boldsymbol{\lambda} \in \mathbb{R}^{n_b}$ is a vector of auxiliary variables (Lagrange multipliers) and n^b is the number of basis functions.

In this work, we apply Kriging to a problem with a two-dimensional input space ($n_{\text{par}} = 2$), and we use tensor-product (bi-linear) basis functions. In computing design and test kernels, we normalized design and test points with respect to their domain, as shown in (25) and (26). Note that in these equations we set the domain of the design and test points to be the same.

$$G_{ik}^d = \exp\left[\frac{\sum_{j=1}^{n_{par}}\left(\frac{(\mu_{ij}^d - \mu_{kj}^d)}{\mu_{\max_j}^d - \mu_{\min_j}^d}\right)^2}{\ell}\right] \tag{25}$$



(a) Histogram (Single Walker)

(b) MCMC Updates (Single Walker)

(c) Scatter plots (Single Walker)

(d) Histogram (Ensemble Walkers)

(e) MCMC Updates (Ensemble Walkers)

(f) Scatter plots (Ensemble Walkers)

Fig. 8. Statistical inversion results for the three-dimensional problem, $\epsilon = 1 \times 10^{-2}$.

$$g_{ik} = \exp\left[\frac{\sum_{j=1}^{n_{par}}\left(\frac{(\mu_{ij}^{d} - \mu_{kj}^{t})}{\mu_{\max_j}^{d} - \mu_{\min_j}^{d}}\right)^2}{\ell}\right] \tag{26}$$

Finally, we can calculate the expected forward solution and its variance using (27) and (28), respectively.

$$E[\mathbf{y}] = \mathbf{w}^T \mathbf{y}^d \tag{27}$$

$$Var[\mathbf{y}] = 1 - \left(\mathbf{w}^T \mathbf{g} + \boldsymbol{\lambda}^T \boldsymbol{\phi}^t\right) \tag{28}$$

Therefore, using the expected-value formula in (27), we can compute approximate individual forward solutions for use in MCMC without running full simulations. We note that a key difference between an emulator such as Kriging and the adjoint acceleration approach is that the adjoint approach yields exact outputs, whereas the emulator yields approximate ones.

6. Implementation

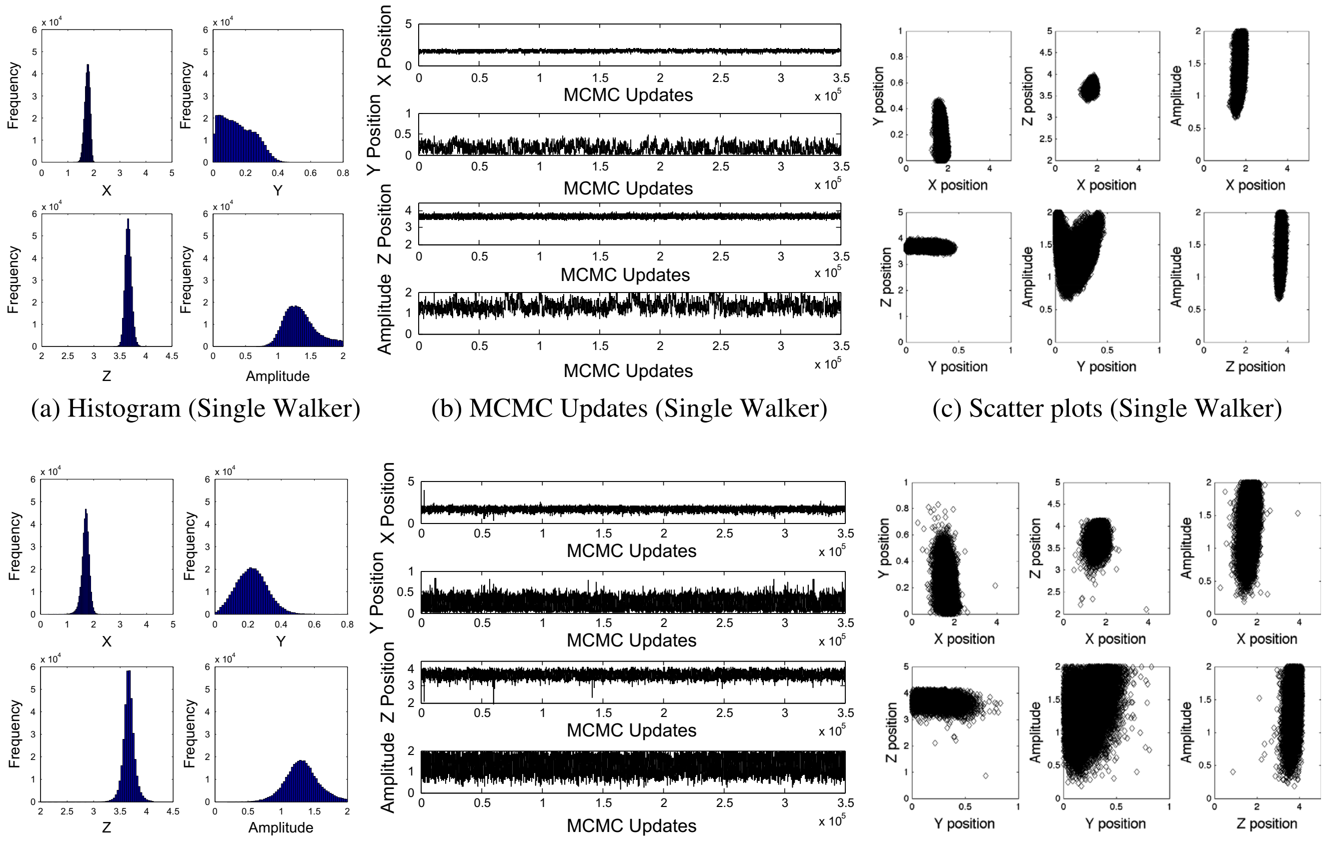
The deterministic and statistical inversion algorithms are implemented using the same software library as the forward code. Both algorithms take advantage of distributed-memory parallelization of the computational domain for large problems, which accelerates the application of the Hessian matrix in the deterministic case and the computation of vector products in the statistical case. Adjoint vectors required for the inversion acceleration are precomputed in parallel during an "offline" stage, as this calculation does not require any information about the measured outputs. These vectors are stored to disk and read during the "online" inversion runs. All calculations are performed on dual quad-core 2.3 GHz nodes with 2 GB RAM per node.

Output measurements for the inversion runs are simulated by running the forward code with the exact inputs to obtain $\mathbf{y}(\boldsymbol{\mu}^e)$, and then adding an additive Gaussian error according to (11). ϵ values of .01, .02 and .05 were considered, corresponding to between 1% and 5% measurement error.

7. Results

In this section, we present deterministic and statistical inversion results for contaminant transport in two and three dimensions. These results are obtained using the following three inversion approaches: deterministic inversion, statistical inversion with single-walker MCMC, and statistical inversion with ensemble-walker MCMC. For each case we verified the discrete adjoint implementation by comparing outputs computed using the adjoint-based sensitivity vectors to the results of an actual forward run for a single random initial condition. The agreement was within residual tolerance (approximately 10^{-10}) for all outputs.

The chosen test cases include both isotropic and anisotropic posteriors. Metrics of comparison are accuracy and efficiency of the inversion method in computing the unknown initial conditions of the inverse problem. Accuracy is measured relative to the "true" input values, $\boldsymbol{\mu}^e$, which are the initial conditions in solving the forward problem. As the solver has not been optimized, efficiency is measured in the number of outputs calculated instead of computational time; computing the outputs is the most expensive part of the inversion process. All results generated by MCMC inversions



(a) Histogram (Single Walker) (b) MCMC Updates (Single Walker) (c) Scatter plots (Single Walker)

(d) Histogram (Ensemble Walkers) (e) MCMC Updates (Ensemble Walkers) (f) Scatter plots (Ensemble Walkers)

Fig. 9. Statistical inversion results for the three-dimensional problem, $\epsilon = 2 \times 10^{-2}$.

Table 5
Comparison of integrated autocorrelation times for the three-dimensional problem.

ϵ	τ_f for	Single walker	Ensemble walkers
1×10^{-2}	μ_X	661.2314	237.0236
	σ_X^2	50.4890	36.3050
	μ_Y	3101.0774	676.3711
	σ_Y^2	533.0037	58.4876
	μ_Z	31.6192	128.0236
	σ_Z^2	9.6079	39.0362
	μ_A	4073.9410	235.0992
	σ_A^2	2559.8635	95.2444
2×10^{-2}	μ_X	594.7043	258.1075
	σ_X^2	50.1377	25.1242
	μ_Y	1186.1198	289.3373
	σ_Y^2	631.8149	47.5943
	μ_Z	53.3130	114.1064
	σ_Z^2	20.3977	22.5975
	μ_A	1575.1466	307.8960
	σ_A^2	945.0098	35.7277

do not include the first 12.5% of the generated samples to remove burn-in.

7.1. Contaminant transport in two dimensions

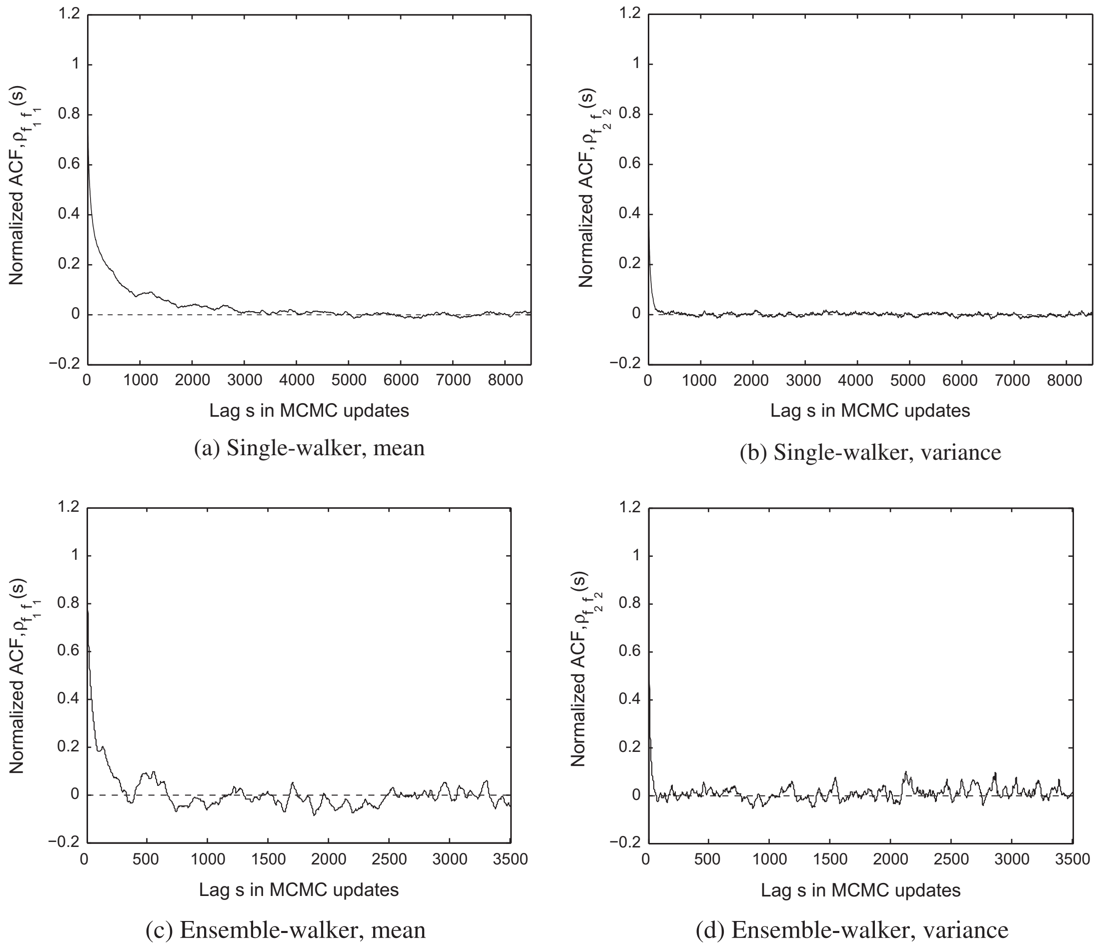
The computational domain for the two-dimensional example consists of a cross-section of a city with buildings, as shown in Fig. 1. Five sensor locations were chosen in a pseudorandom fashion by a user without iteration or tuning. Each sensor took three readings spaced equally in time, for a total of 15 outputs. A spatial approximation order of $p = 2$ was used to generate the baseline output values $\mathbf{y}(\boldsymbol{\mu}_e)$. The Peclet number for the simulations, based on the mean velocity and domain size in the x-direction, is $Pe = 100$.

7.1.1. Isotropic posterior

In this example, we fix the contaminant amplitude at unity and the contaminant spread, i.e. standard deviation, at 1/8. The inverse problem then consists of solving for the contaminant position $\boldsymbol{\mu} = (X, Y)$. The "true" initial position is $\boldsymbol{\mu}_e = (0.2, 0.3)$, and the parameter domain is $[0, 1]^2$, which defines the left half of the computational domain. We use a chain of $N_m = 50,000$ samples for the single-walker MCMC inversion, and a chain of $N_m^e = 5,000$ samples for an ensemble-walker MCMC inversion with $L = 10$ walkers.

Figs. 2 and 3 show sample scatter plots, histograms, and MCMC updates of single and ensemble walkers for measurement errors of $\epsilon = 1 \times 10^{-2}$ and 5×10^{-2}. "Fuzzy worms", one indication of chain convergence, are formed with MCMC updates of both the single and ensemble walker algorithms. As expected, the larger measurement error yields a larger spread in the inverted parameters.

To further assess convergence of the MCMC chains, the integrated autocorrelation times, τ_f and τ_f^e, were calculated for mean and variance estimators of both the X and Y positions. The values of τ_f (single walker) were between 6 and 15, and the values of τ_f^e (ensemble walker) were between 3 and 49. These values are significantly smaller than the total number of samples. Since each iteration of the ensemble MCMC algorithm is a factor of $L = 10$ more expensive than an iteration of a single walker MCMC, the fact that τ_f^e ranges from half to over three times as large as τ_f suggests that the ensemble walker MCMC algorithm is not necessary for this isotropic problem.



(a) Single-walker, mean

(b) Single-walker, variance

(c) Ensemble-walker, mean

(d) Ensemble-walker, variance

Fig. 10. Normalized autocovariance functions for mean and variance estimators of the X position for the three-dimensional problem with $\epsilon = 2 \times 10^{-2}$.

Table 6
Comparison of statistical results for the three-dimensional problem. The true parameters are $(X, Y, Z, A) = (1.7, 0.12, 3.6, 1.0)$. Note that μ and σ refer to a parameter mean and standard deviation.

ϵ	Parameter	Deterministic	Single walker	Ensemble walkers
1×10^{-2}	μ_X	1.6972	$1.6896 \pm 2.10 \times 10^{-3}$	$1.6657 \pm 1.87 \times 10^{-3}$
	σ_X	–	$0.0491 \pm 3.96 \times 10^{-5}$	$0.0720 \pm 1.21 \times 10^{-4}$
	μ_Y	0.0859	$0.0859 \pm 5.60 \times 10^{-3}$	$0.1401 \pm 2.64 \times 10^{-3}$
	σ_Y	–	$0.0593 \pm 1.54 \times 10^{-4}$	$0.0601 \pm 9.80 \times 10^{-5}$
	μ_Z	3.5518	$3.5517 \pm 2.48 \times 10^{-4}$	$3.5500 \pm 8.49 \times 10^{-4}$
	σ_Z	–	$0.0261 \pm 5.07 \times 10^{-6}$	$0.0444 \pm 5.20 \times 10^{-5}$
	μ_A	0.9584	$1.1020 \pm 2.39 \times 10^{-2}$	$0.9881 \pm 3.18 \times 10^{-3}$
	σ_A	–	$0.2216 \pm 6.60 \times 10^{-3}$	$0.1226 \pm 9.82 \times 10^{-4}$
	Acceptance ratio		0.313	0.280
2×10^{-2}	μ_X	Not converged	$1.7466 \pm 4.10 \times 10^{-3}$	$1.6936 \pm 3.78 \times 10^{-3}$
	σ_X	–	$0.0987 \pm 1.58 \times 10^{-4}$	$0.1393 \pm 4.59 \times 10^{-4}$
	μ_Y	Not converged	$0.1547 \pm 5.70 \times 10^{-3}$	$0.2230 \pm 2.67 \times 10^{-3}$
	σ_Y	–	$0.0975 \pm 4.39 \times 10^{-4}$	$0.0929 \pm 1.81 \times 10^{-4}$
	μ_Z	Not converged	$3.6619 \pm 7.47 \times 10^{-4}$	$3.6652 \pm 1.94 \times 10^{-3}$
	σ_Z	–	$0.0606 \pm 4.20 \times 10^{-5}$	$0.1073 \pm 3.77 \times 10^{-4}$
	μ_A	Not converged	$1.3392 \pm 1.61 \times 10^{-2}$	$1.3126 \pm 7.40 \times 10^{-3}$
	σ_A	–	$0.2401 \pm 4.20 \times 10^{-3}$	$0.2493 \pm 1.23 \times 10^{-3}$
	Acceptance ratio		0.322	0.282

Fig. 4 shows the normalized autocovariance functions versus sample lag s for the mean and variance estimators of the X position at $\epsilon = 5 \times 10^{-2}$. The autocovariance functions approach zero rapidly and then oscillate randomly due to noise, indicating adequate mixing of the MCMC chains in both the single and ensemble walker cases.

The inversion results obtained from all three methods are listed in Table 1. The deterministic results consist of only single values

Table 7
Computational time consumed by different components of the solver for the three-dimensional test case. The computational times have been normalized by the forward solve time. Again, the larger forward solve time relative to the adjoint is due to overhead associated with a solver that is generally-applicable to nonlinear problems.

Forward solve	Adjoint solve	Adjoint-accelerated	
		Output calculation	Hessian application
1	0.30	3.9×10^{-5}	2.1×10^{-3}

for X and Y, whereas for the statistical results we are able to show both the mean and standard deviation estimators and their respective variances. From this table we see that as the measurement error increases, the quality of the deterministic inversion deteriorates, as expected. While the mean estimates from the statistical results move further from the true values as well, and interestingly follow closely the single-point deterministic results, the standard deviation estimators increase correspondingly, informing the user that the precise location of the contaminant source is not well-known. Ultimately, the histograms of X and Y in the statistical case provide the most detailed information about the accuracy of the source inversion.

Another important question is the efficiency of the methods in obtaining these results. By design of the MCMC chains, the number of outputs evaluated by the single and ensemble-walker algorithms were roughly equal at 50,000 – in fact, slightly lower because some proposals fell outside the parameter domain and were immediately rejected. The cost of the deterministic inversion is more difficult to compare quantitatively given that the algorithm is not based on sample evaluations. Here we refer to the computational times in our implementation, as presented in Table 2. Including all restarts from different initial conditions, on the order of 2000 Hessian matrix applications were required for the deterministic results. In our implementation this made the computational cost of the deterministic inversion approximately 20 times faster than the statistical results. We expect the deterministic algorithm to be faster, but we are cautious about generalizing conclusions due to arbitrary choices (e.g., number of MCMC samples or the number of deterministic restarts) in both algorithms.

In Table 2 we also see the effect of adjoint acceleration on the output calculation. Accounting for the fact that a forward solve yields multiple outputs (15 in this case), we see that the speedup of the adjoint computation is still approximately 3.5 orders of magnitude. A similar speedup occurs in the application of the Hessian matrix, which would normally require one forward and one adjoint solve. For larger cases, this acceleration is critical to enabling real-time inversions.

7.1.2. Anisotropic posterior

In the next set of results we use the same two-dimensional problem as before but add contaminant amplitude, A, to the list of inversion parameters. This addition makes the posterior aniso-

Table 8
Comparison of deterministic and statistical results of the isotropic, two-dimensional problem at increased numerical resolution. The true parameters are $(X, Y) = (0.2, 0.3)$, and $\epsilon = 2 \times 10^{-2}$.

p, N_T	Parameter	Deterministic	Single walker	Ensemble walkers
2200	μ_X	2.034	$0.1995 \pm 2.57 \times 10^{-4}$	$0.1970 \pm 3.42 \times 10^{-4}$
	σ_X	–	$0.0083 \pm 4.37 \times 10^{-6}$	$0.0118 \pm 6.27 \times 10^{-6}$
	μ_Y	0.3347	$0.3428 \pm 5.42 \times 10^{-4}$	$0.3471 \pm 6.10 \times 10^{-4}$
	σ_Y	–	$0.0164 \pm 1.24 \times 10^{-5}$	$0.0212 \pm 1.41 \times 10^{-5}$
	Acceptance ratio		0.383	0.320
3400	μ_X	2.014	$0.1971 \pm 2.95 \times 10^{-4}$	$0.1936 \pm 4.08 \times 10^{-4}$
	σ_X	–	$0.0090 \pm 4.89 \times 10^{-6}$	$0.0127 \pm 8.40 \times 10^{-6}$
	μ_Y	0.3358	$0.3444 \pm 5.99 \times 10^{-4}$	$0.3499 \pm 7.68 \times 10^{-4}$
	σ_Y	–	$0.0171 \pm 1.29 \times 10^{-5}$	$0.0222 \pm 1.37 \times 10^{-5}$
	Acceptance ratio		0.389	0.316

Table 9
Comparison of statistical results for the three-dimensional problem. True values of the parameters are $(X, Y, Z, A) = (1, 7, 0.12, 3.6, 1.0)$, and $\epsilon = 1 \times 10^{-2}$.

p, N_T	Parameter	Deterministic	Single walker	Ensemble walkers
1200	μ_X	1.6972	$1.6896 \pm 2.10 \times 10^{-3}$	$1.6657 \pm 1.87 \times 10^{-3}$
	σ_X	–	$0.0491 \pm 3.96 \times 10^{-5}$	$0.0720 \pm 1.21 \times 10^{-4}$
	μ_Y	0.0859	$0.0859 \pm 5.60 \times 10^{-3}$	$0.1401 \pm 2.64 \times 10^{-3}$
	σ_Y	–	$0.0593 \pm 1.54 \times 10^{-4}$	$0.0601 \pm 9.80 \times 10^{-5}$
	μ_Z	3.5518	$3.5517 \pm 2.48 \times 10^{-4}$	$3.5500 \pm 8.49 \times 10^{-4}$
	σ_Z	–	$0.0261 \pm 5.07 \times 10^{-6}$	$0.0444 \pm 5.20 \times 10^{-5}$
	μ_A	0.9584	$1.1020 \pm 2.39 \times 10^{-2}$	$0.9881 \pm 3.18 \times 10^{-3}$
	σ_A	–	$0.2216 \pm 6.60 \times 10^{-3}$	$0.1226 \pm 9.82 \times 10^{-4}$
	Acceptance ratio		0.313	0.280
2400	μ_X	1.6940	$1.6895 \pm 3.30 \times 10^{-3}$	$1.6679 \pm 1.72 \times 10^{-3}$
	σ_X	–	$0.0486 \pm 3.91 \times 10^{-5}$	$0.0707 \pm 1.40 \times 10^{-4}$
	μ_Y	0.0866	$0.0813 \pm 7.40 \times 10^{-3}$	$0.1366 \pm 2.66 \times 10^{-3}$
	σ_Y	–	$0.0590 \pm 1.84 \times 10^{-4}$	$0.0584 \pm 8.70 \times 10^{-5}$
	μ_Z	3.5523	$3.5525 \pm 3.00 \times 10^{-4}$	$3.5509 \pm 1.12 \times 10^{-3}$
	σ_Z	–	$0.0260 \pm 4.99 \times 10^{-6}$	$0.0444 \pm 5.30 \times 10^{-5}$
	μ_A	0.9564	$1.1247 \pm 3.75 \times 10^{-2}$	$0.9875 \pm 3.16 \times 10^{-3}$
	σ_A	–	$0.2469 \pm 1.09 \times 10^{-2}$	$0.1230 \pm 8.07 \times 10^{-4}$
	Acceptance ratio		0.309	0.280

Table 10
Comparison of statistical results for the two-dimensional problem using Kriging with uniform grid. The true parameter values are $(X, Y, A) = (0.2, 0.3)$.

ϵ	Parameter	Single walker set 1	Single walker set 2	Ensemble walkers
1×10^{-2}	μ_X	$0.2005 \pm 8.69 \times 10^{-5}$	$0.3286 \pm 5.49 \times 10^{-5}$	$0.2011 \pm 2.38 \times 10^{-4}$
	σ_X	$0.0050 \pm 5.61 \times 10^{-7}$	$0.0041 \pm 3.17 \times 10^{-7}$	$0.0079 \pm 4.82 \times 10^{-6}$
	μ_Y	$0.2995 \pm 8.03 \times 10^{-5}$	$0.0041 \pm 3.17 \times 10^{-7}$	$0.2988 \pm 2.22 \times 10^{-4}$
	σ_Y	$0.0046 \pm 4.86 \times 10^{-7}$	$0.0048 \pm 4.30 \times 10^{-7}$	$0.0072 \pm 2.09 \times 10^{-6}$
	Acceptance ratio	0.323	0.327	0.316
5×10^{-2}	μ_X	$0.2351 \pm 3.24 \times 10^{-3}$	$0.3552 \pm 1.28 \times 10^{-2}$	$0.2550 \pm 2.78 \times 10^{-3}$
	σ_X	$0.0644 \pm 3.92 \times 10^{-4}$	$0.1117 \pm 5.73 \times 10^{-3}$	$0.0858 \pm 3.90 \times 10^{-4}$
	μ_Y	$0.2841 \pm 8.04 \times 10^{-4}$	$0.6712 \pm 5.00 \times 10^{-3}$	$0.2756 \pm 8.63 \times 10^{-4}$
	σ_Y	$0.0273 \pm 1.54 \times 10^{-5}$	$0.0573 \pm 9.57 \times 10^{-5}$	$0.0379 \pm 3.79 \times 10^{-2}$
	Acceptance ratio	0.340	0.331	0.319

tropic, as there can now be a trade-off between position and amplitude via diffusion. The same true X and Y values are used as in the previous example, and the true amplitude is set to $A = 1.0$ with an amplitude domain of $[0.1, 2.0]$.

For the present anisotropic posterior, longer MCMC chains are needed. We used $N_m = 400,000$ samples for the single-walker MCMC, and, to obtain a similar number of output evaluations, $N_m^e = 20,000$ steps of the ensemble-walker MCMC with $L = 20$ walkers. The larger number of walkers for this case was needed to efficiently sample the anisotropic posterior.

Figs. 5 and 6 show histograms, MCMC updates, and parameter scatter plots for the single and ensemble walkers. We see that in the ensemble-walker case, the MCMC updates form "fuzzy worms" indicative of well-mixed chains. On the other hand, the single-walker MCMC chains do not appear to be well-mixed, especially for $\epsilon = 1 \times 10^{-2}$.

Next, we calculate the integrated auto-correlation time τ_f for the mean and variance of all initial conditions to assess the MCMC convergence. Table 3 shows that MCMC inversion with ensemble walkers generally has a lower autocorrelation time by at least a factor of $L = 20$, which means that MCMC inversion with the ensemble-walker algorithm is more efficient in this case. Note that we could not calculate τ_f of single walker for $\epsilon = 1 \times 10^{-2}$ due to the lack of convergence in the MCMC chain.

Table 4 compares the results of the deterministic and statistical inversion algorithms for this problem. We note that we were unable to obtain data for single-walker MCMC inversion at $\epsilon = 1 \times 10^{-2}$ due to the lack of convergence of the single walker for such a highly-anisotropic posterior. At $\epsilon = 5 \times 10^{-2}$, both statistical inversion algorithms yield comparable estimators. However, the estimator variances are lower in the case of the ensemble walkers, indicating that the chain is better-mixed in this case.

The deterministic results only provide information about the mean values of the parameters of interest, and these are of similar accuracy as the statistical results. However, the deterministic results do not provide information about the variance of the inverted parameters, and hence no metric by which to judge the quality of the inversion.

In order to assess the efficiency of the statistical inversions, we calculate the number of output evaluations. The single-walker MCMC runs required roughly N_m output evaluations, i.e. 400,000. This is because few proposals fell outside the domain due to the use of a small proposal hypercube required to keep the acceptance ratio at the optimal 30%. In the ensemble-walker case, about 25% of the proposals fell outside the parameter domain, so that the number of output evaluations was smaller, at approximately 300,000. In this case, the deterministic results were obtained with roughly the same number of Hessian matrix applications as in the previous example, i.e. 2000. The computational time results for this case are the same as given in Table 2. This makes the deterministic solve about two orders of magnitude faster than the statistical methods. However, we again stress the presence of multiple tunable parameters affecting the accuracy and robustness of the deterministic approach, as well as the fact that the deterministic result provides only a single-point answer, as opposed to the entire posterior distribution obtained from statistical methods.

7.2. Contaminant transport in three dimensions

For our three-dimensional test case, we simulate contaminant transport in a realistic urban area. The domain, shown in Fig. 7, is the same as in Lieberman et al. [24]: a rectangular prism of dimensions 4.71 × 4.13 × 0.85, where one model unit corresponds to 100 m in the physical domain. Outputs are obtained from 10

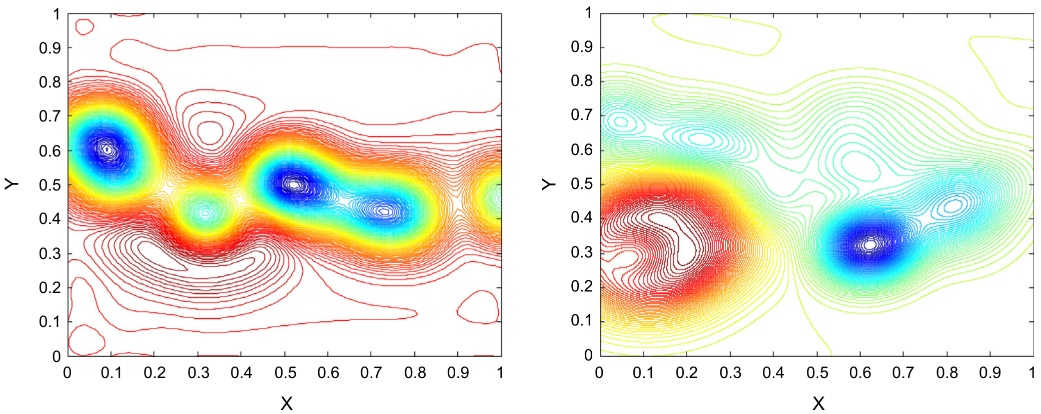


Fig. 11. The posterior generated using Kriging (left) compared to the exact posterior (right). There are more local maxima and minima when posterior is generated using Kriging. Anisotropy also occurs when Kriging is used.

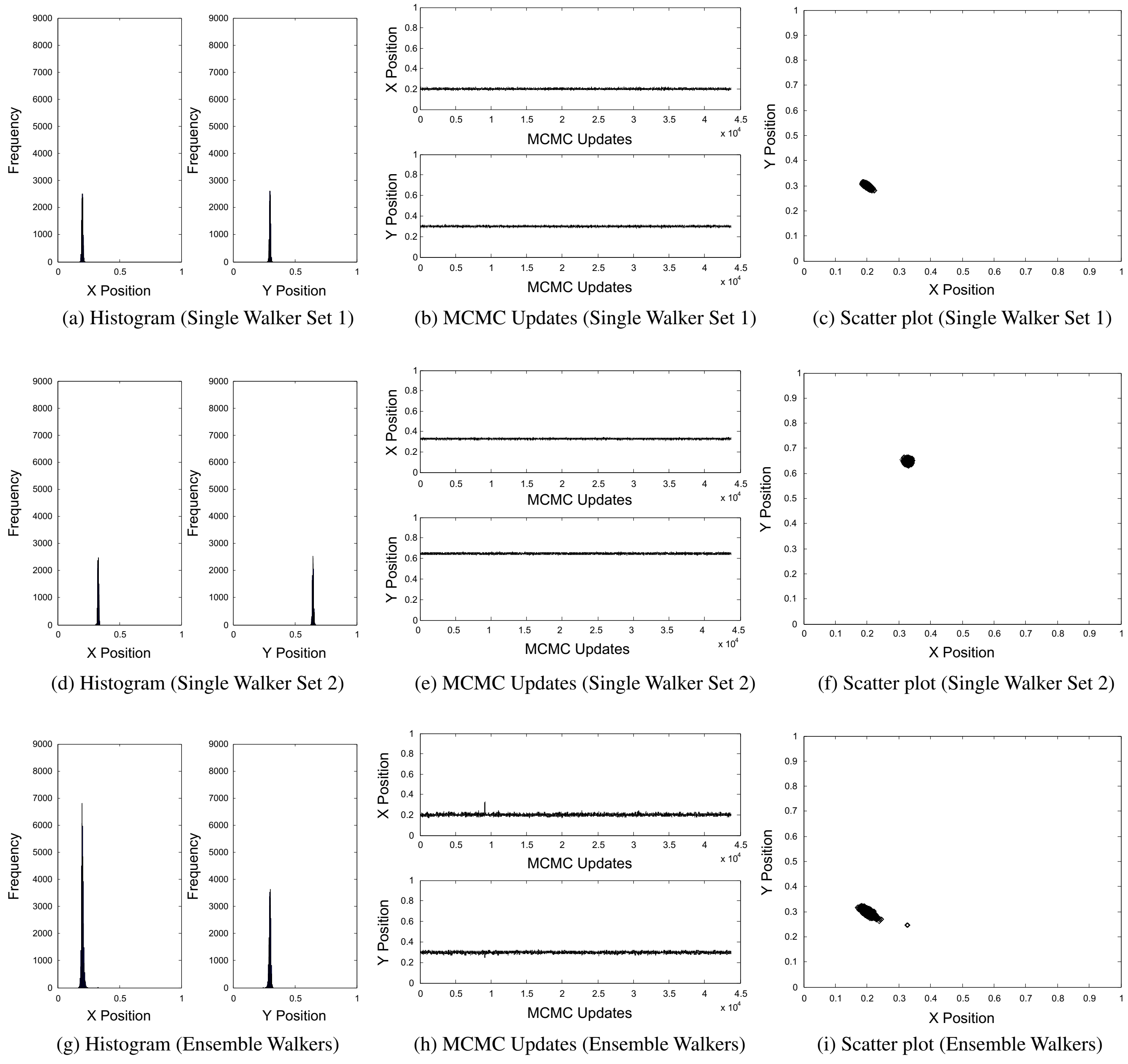
sensors with locations as shown in Fig. 7. Each sensor provides 4 readings for a total of 40 outputs. The sensors were chosen in a pseudorandom fashion by a user without iteration or tuning. The Peclet number for the simulations, based on the mean velocity and the domain extent in the direction of velocity, was $Pe = 50$. True output measurements were generated by running the forward simulation with finite element order $p = 1$ approximation and a prescribed initial condition as shown in Fig. 7.

For this test case, we set the "true" input parameters to $(X, Y, Z, A) = (1.7, 0.12, 3.6, 1.0)$. We used $N_m = 400,000$ samples for the single walker, and $N_m^e = 20,000$ samples of the ensemble-walker with $L = 20$ walkers. Again, we chose the number of walkers to get good results, although we did not find a high degree of sensitivity to the number of walkers, as long as this number was sufficiently high. Figs. 8 and 9 show statistical inversion results for $\epsilon = 1 \times 10^{-2}$ and 2×10^{-2}. Notice that while all MCMC chains of ensemble walkers formed "fuzzy worms", this was not the case for the single walker (e.g., the MCMC chains for Y position and Amplitude in Fig. 8). Note also the high degree of scatter for the single-walker samples with $\epsilon = 2 \times 10^{-2}$.

Table 5 compares the integrated autocorrelation times for both statistical inversions. The autocorrelation times of the ensemble walker algorithm are generally lower than those of the single factor, in some cases by up to a factor of 20.

Fig. 10 shows the normalized autocovariance functions for mean and variance estimators of the X position using $\epsilon = 2 \times 10^{-2}$. The autocovariances for both the single and the ensemble-walker cases are seen to approach zero rapidly and then to oscillate randomly due to noise, indicating well-mixed chains. We note that the oscillations in the ensemble-walker case are of larger amplitude than in the single-walker case.

Table 6 show deterministic and statistical results for the three-dimensional problem. Whereas the deterministic results provide only one value for each parameter, the statistical results yield addi-



(a) Histogram (Single Walker Set 1)

(b) MCMC Updates (Single Walker Set 1)

(c) Scatter plot (Single Walker Set 1)

(d) Histogram (Single Walker Set 2)

(e) MCMC Updates (Single Walker Set 2)

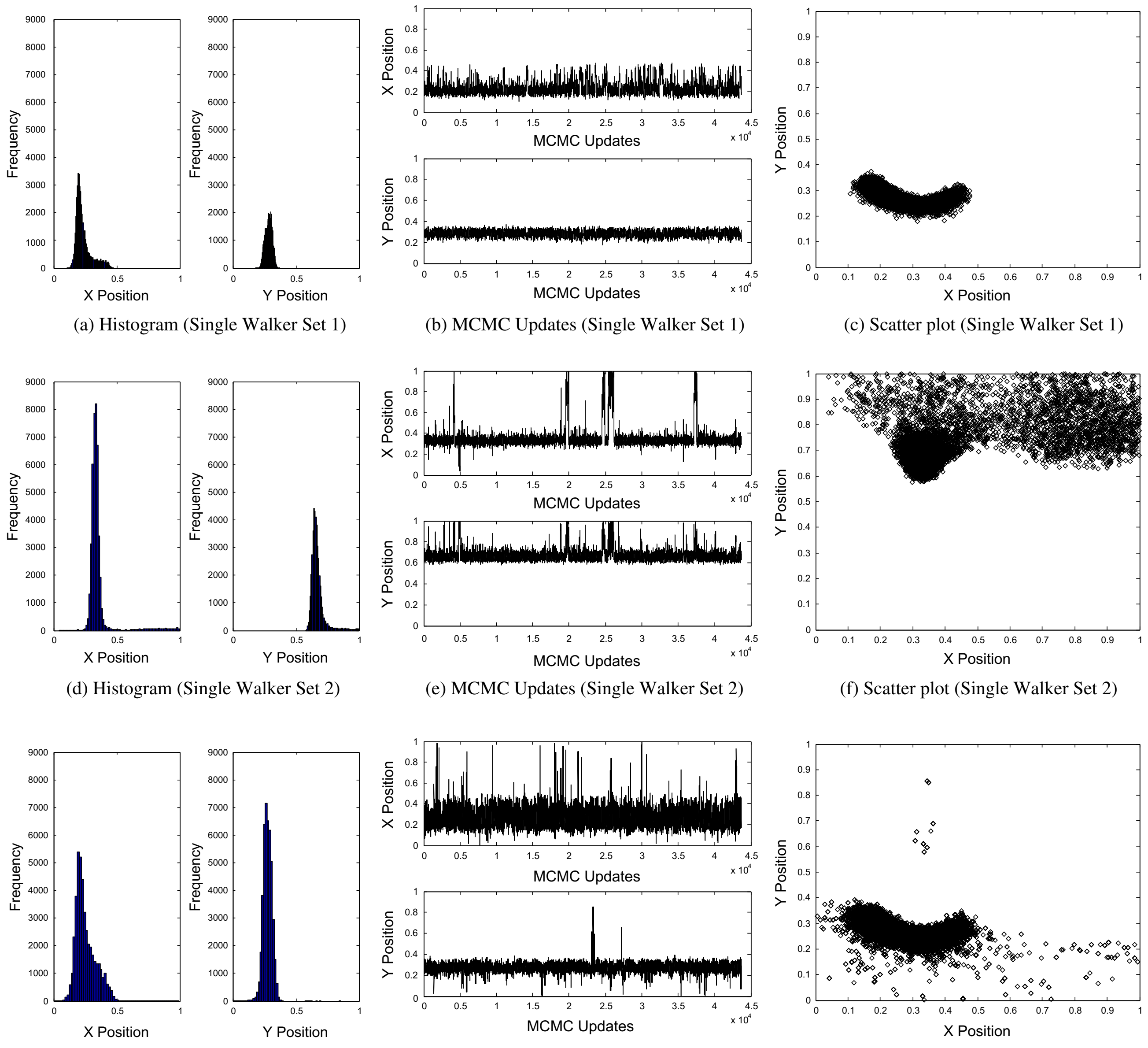
(f) Scatter plot (Single Walker Set 2)

(g) Histogram (Ensemble Walkers)

(h) MCMC Updates (Ensemble Walkers)

(i) Scatter plot (Ensemble Walkers)

Fig. 12. Statistical inversion results for the two-dimensional problem generated using Kriging with uniform grid, $\epsilon = 1 \times 10^{-2}$.



(a) Histogram (Single Walker Set 1)

(b) MCMC Updates (Single Walker Set 1)

(c) Scatter plot (Single Walker Set 1)

(d) Histogram (Single Walker Set 2)

(e) MCMC Updates (Single Walker Set 2)

(f) Scatter plot (Single Walker Set 2)

(g) Histogram (Ensemble Walkers)

(h) MCMC Updates (Ensemble Walkers)

(i) Scatter plot (Ensemble Walkers)

Fig. 13. Statistical inversion results for the two-dimensional problem generated using Kriging with uniform grid, $\epsilon = 5 \times 10^{-2}$.

tional information through standard deviations and histograms. Moreover, for $\epsilon = 2 \times 10^{-2}$, the deterministic inversion failed to converge for any regularization parameter, β_M, attempted. In other words, at this level of noise, the output measurements yield a problem that is not suitable for solving via the deterministic approach. On the other hand, both the single-walker and the ensemble-walker statistical approaches produce reasonable inversions. We note that the estimator variances are more often lower for the ensemble-walker algorithm compared to the single-walker algorithm, indicating slightly better convergence of the ensemble-walker MCMC chains.

To assess efficiency, we look at the number of output calculations required by the MCMC algorithms. For the single-walker case, these were 392,798 and 393,041 for $\epsilon = 1 \times 10^{-2}$ and $\epsilon = 2 \times 10^{-2}$, respectively. The ensemble-walker algorithm required fewer output evaluations: 282,384 and 220,656 for $\epsilon = 1 \times 10^{-2}$ and $\epsilon = 2 \times 10^{-2}$, respectively. The lower number of output evaluations in the ensemble-walker case is caused by a greater percentage of samples falling outside of the parameter domain due to a larger allowable proposal. This effect was observed in the two-dimensional anisotropic case, and it is indicative of better exploration of the parameter space by the ensemble-walker algorithm. Finally, the cost of a deterministic inversion was again approximately 2000 Hessian-matrix applications. Based on the data in Table 7 this resulted in a time reduction of about two orders of magnitude compared to the statistical inversion in our implementation. We stress again, however, that this timing comparison is quite approximate and that the deterministic inversions can fail to converge, and if they do converge, they provide only single-point inversion results as opposed to complete histograms. Finally, we note from Table 7 that while the adjoint-accelerated output calculation consumes approximately the same relative time as in the two-dimensional case, the Hessian application is more expensive due to a larger number of input parameters.

7.3. Numerical accuracy

In the two and three-dimensional cases, we set the finite element order and number of time steps at reasonable values such that numerical accuracy considerations did not significantly influence the results. To demonstrate the effect of numerical discretization errors, in this section we repeat the inversion results with an incremented spatial approximation order, $p = 2 \rightarrow 3$ in two dimensions and $p = 1 \rightarrow 2$ in three dimensions, and with double the number of time steps. For conciseness, we only present here the mean values and standard deviations for the statistical inversions. In addition, we simulate the exact outputs by perturbing the lower-resolution exact data, $\mathbf{y}(\boldsymbol{\mu}_e)$; runs at both resolutions use the same simulated output measurements.

7.3.1. Isotropic, two-dimensional problem

Inversions for the two-dimensional isotropic problem were performed at both the original and augmented numerical resolutions using a measurement error of $\epsilon = 2 \times 10^{-2}$. In the statistical inversions, $N_m = 50,000$ samples were used in the single-walker case, and $N_m^e = 5000$ samples were used in the $L = 10$ ensemble-walker case. Table 8 compares the deterministic and statistical inversion results at the two levels of numerical resolution. As shown, the differences in the results are all less than 5%, indicating that for the discretization parameters chosen numerical resolution does not have a strong effect on the observed results.

7.3.2. Three-dimensional problem

Finally, inversions were also repeated for the three-dimensional problem at the original and augmented numerical resolutions using a measurement error of $\epsilon = 1 \times 10^{-2}$. In the statistical inversions, $N_m = 400,000$ samples were used in the single-walker case, and $N_m^e = 20,000$ samples were used in the $L = 20$ ensemble-walker case. Table 9 compares the deterministic and statistical inversion results at the two levels of numerical resolution. While some differences are larger compared to the two-dimensional case (the largest mean value difference is approximately 6%, and one standard deviation difference is 11%) most of the values agree very well. That is, numerical accuracy effects are unlikely to change any of the conclusions in the three-dimensional case.

7.4. Comparison with MCMC chain driven by Kriging

In this section, we present the two-dimensional, isotropic result using Kriging output calculations to drive the MCMC chain as comparison results. The set up of the problem is the same as the two-dimensional, isotropic result as presented in section 7.1.1. Our design points are chosen to form a uniform grid with 11 equally-spaced points in each dimension. A uniform grid is used to decrease the probability of facing a singularity problem in calculating the individual forward solutions during the MCMC process. The test point is the proposed location during the MCMC process. We set the kernel $\boldsymbol{G}^d$ and $\boldsymbol{g}(\boldsymbol{T})$ to be the covariance with correlation length of $\ell = 0.01$.

For MCMC chain with single walker, we set the initial location of the walker to be at (0.0258,0.0516) in the first set and at (1.0,1.0) in the second set. As we can see in Table 10, the first and second set of MCMC chain with single walker did not converge to the same value. The reason is because when Kriging is used to drive the MCMC chain, the number of local maxima and minima in the posterior increases as compared to when adjoint acceleration is used. The posterior generated using Kriging also introduces anisotropy. Fig. 11 shows the posterior generated using adjoint acceleration and Kriging. Therefore, as what we seen before, it is expected that the MCMC chain with single walker might not converge to the true value parameter values, while MCMC chain with ensemble walkers will converge to the true parameter values. Figs. 12 and 13 show the statistical inversion results for $\epsilon = 1 \times 10^{-2}$ and $\epsilon = 5 \times 10^{-2}$, respectively.

8. Conclusions

We presented both deterministic and statistical algorithms for solving linear but large-scale contaminant source inversion problems. At the heart of these algorithms is a separation of the inversion into an offline and online stage. The offline stage requires knowledge of the geometry, prevailing wind, and sensor locations, but not of the sensor readings. The online stage takes as input sensor readings and returns information about the initial location of the contaminant. There is no approximation or associated loss of accuracy in this separation since discrete adjoints are computed in the offline stage; that is, the outputs computed during the online stage via inner products with sensitivity vectors are exactly the same as if the forward simulation were performed.

In the statistical case, an ensemble-walker variant of the Markov chain Monte Carlo method is demonstrated to be more efficient at sampling anisotropic posterior distributions, which we have found to occur often in contaminant source inversion problems. This method has the additional advantage in that it can be parallelized across walkers as all but one walker move in each MCMC step. For example, in the three-dimensional case, an online inversion speedup of approximately 20 (the number of walkers) could be attained. We did not employ this parallelization in the present work, but ideal scaling is expected for this embarrassingly-parallel portion of the algorithm.

The present work requires the initial contaminant distribution to be parametrized by a small number of scalar inputs. This parametrization can be extended to initial conditions that are more complicated than single Gaussians through the addition of more parameters. The statistical inversion is expected to perform well for a larger number of parameters due to favorable scaling of the Markov chain Monte Carlo method. In addition, adjoint acceleration is not expected to deteriorate since the cost of each adjoint solution depends only on the number of outputs, not the number of inputs.

We expect tremendous benefits from the adjoint approach even for a larger number of sensor readings, or outputs. Note that additional outputs increase primarily the cost of the offline stage, in which an adjoint solution is calculated for each output. These can be pre-computed for any number of outputs since the offline stage is not time-critical. The online stage is affected to a lesser extent through additional inner products, one for each output evaluation. However, these calculations are trivially parallelizable over the outputs, and hence the cost of additional outputs can be perfectly compensated by additional compute cores.

We note also that the online stages of adjoint-accelerated deterministic and statistical inversion algorithms are not independent of the size of the large-scale problem. This contrasts with general reduced-order models, in which the number of original degrees of freedom no longer appears in the online stage. However, the adjoint approach does not introduce any additional approximation errors and still yields orders of magnitude speedup for an otherwise intractable inversion problem. This speedup can produce real-time inversion results on today's commodity multi-core architectures.

Linearity of the transport equation is an intrinsic component of the acceleration, and hence the approach would not apply directly to problems with, for example, nonlinear reactions involving the contaminant. Of course, this restriction does not preclude linear transport with a velocity field obtained from a nonlinear flow solution. Future directions include extending the presented algorithms to real-time-inferred velocity fields, multiple sources, and to including release time as a parameter.

References

[1] Hadamard J. Lectures on the cauchy problem in linear partial differential equations. Yale University Press; 1923.

[2] Higdon D, Lee H, Bi Z. A Bayesian approach to characterizing uncertainty in inverse problems using coarse and fine-scale information. IEEE Trans Signal Process 2002;50(2):389–99.

[3] Kaipio J, Somersalo E. Statistical inverse problems: discretization, model reduction, and inverse crimes. J Comput Appl Math 2007;198:493–504.

[4] Snodgrass MF, Kitanidis PK. A geostatistical approach to contaminant source identification. Water Resour Res 1997;33(4):537–46.

[5] Cybyk BZ, Boris JP, Y TR, Jr., Lind CA, Landsberg AM. A detailed contaminant transport model for facility hazard assessment in urban areas. AIAA Paper 1999-3441; 1991.

[6] Rödenbeck C, Houweling S, Gloor M, Heimann M. CO_2 flux history 1982–2001 inferred from atmospheric data using global inversion of atmospheric transport. Atmos Chem Phys 2003;3:1919–64.

[7] Michalak AM, Bruhwiler L, Tans PP. A geostatistical approach to surface flux estimation of atmospheric trace gases. J Geophys Res 2004;109.

[8] Akcelilk V, Biros G, Draganescu A, Ghattas O, Hill J, van Bloemen Waanders B. Dynamic data-driven inversion for terascale simulations: Real-time identification of airborne contaminants. In: 2005 ACM/IEEE conference on supercomputing; 2005.

[9] Bashir O, Willcox K, Ghattas O, van Bloemen Waanders B, Hill J. Hessian-based model reduction for large-scale systems with initial condition inputs. Int J Numer Meth Eng 2007;0(0). http://dx.doi.org/10.1002/nme.2100. 0-0.

[10] Efendiev Y, Hou T, Luo W. Preconditioned Markov chain Monte Carlo simulations using coarse-scale models. SIAM J Scient Comput 2007;28(2):776–803.

[11] Galbally D, Fidkowski K, Willcox K, Ghattas O. Nonlinear model reduction for uncertainty quantification in large-scale inverse problems. Int J Numer Meth Eng 2009;81:1581–608.

[12] Marzouk YM, Najm HN, Rahn LA. Stochastic spectral methods for efficient Bayesian solution of inverse problems. J Comput Phys 2007;224:560–86.

[13] Chung H-S, Alonso JJ. Using gradients to construct response surface models for high-dimensional design optimization problems. AIAA Paper 2001-0922; 2001.

[14] Chung H-S, Alonso JJ. Using gradients to construct cokriging approximation models for high-dimensional design optimization problems. AIAA Paper 2002-0317; 2002.

[15] Dwight RP, Han Z-H. Efficient uncertainty quantification using gradient-enhanced Kriging. AIAA Paper 2009-2276; 2009.

[16] Hanson KM, Cunningham GS. Posterior sampling with improved efficiency. Med Imag: Image Process 1998:371–82.

[17] Dostert P, Efendiev Y, Mohanty B. Efficient uncertainty quantification techniques in inverse problems for Richards' equation using coarse-scale simulation models. Advan Water Resour 2009;32:329–39.

[18] Fidkowski KJ, Roe PL. An entropy adjoint approach to mesh refinement. SIAM J Scient Comput 2010;32(3):1261–87.

[19] Bassi F, Rebay S. GMRES discontinuous Galerkin solution of the compressible Navier–Stokes equations. In: Cockburn K, Shu, editors. Discontinuous Galerkin methods: theory, computation and applications. Berlin: Springer; 2000. p. 197–208.

[20] Tarantola A. Inverse problem theory and methods for model parameter estimation. Philadelphia: SIAM; 2005.

[21] Calvetti D, Somersalo E. Introduction to Bayesian scientific computing: ten lectures on subjective computing. Springer; 2008.

[22] Hastings W. Monte Carlo sampling methods using Markov chains and their applications. Biometrika 1970;57(1):97–109.

[23] Goodman J, Weare J. Ensemble samplers with affine invariance. Commun Appl Math Comput Sci 2010;5(1):65–80.

[24] Lieberman C, Fidkowski K, Willcox K, van Bloemen Waanders B. Hessian-based model reduction: large-scale inversion and prediction. Int J Numer Meth Fluids, doi:http://dx.doi.org/10.1002/fld.3650.