UCRL-TR-229417

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Dynamic Data-Driven Event Reconstruction for Atmospheric Releases

B. Kosovic, R. Belles, F. K. Chow, L. D. Monache, K. Dyer, L. Glascoe, W. Hanley, G. Johannesson, S. Larsen, G. Loosmore, J. K. Lundquist, A. Mirin, S. Neuman, J. Nitao, R. Serban, G. Sugiyama, R. Aines

March 26, 2007

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

Dynamic Data-Driven Event Reconstruction for Atmospheric Releases

Branko Kosović, Richard Belles, Fotini K. Chow,
Luca Delle Monache, Kathleen Dyer,
Lee Glascoe, William Hanley, Gardar Johannesson,
Shawn Larsen, Gwendolen Loosmore,
Julie K. Lundquist, Arthur Mirin, Sthephanie Neuman, John Nitao,
Gardar Johannesson, Radu Serban, Gayle Sugiyama, and Roger Aines

March 26, 2007

Lawrence Livermore National Laboratory

UCRL-TR-229417

Disclaimer and Auspices

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48. The project 04-ERD-037 was funded by the Laboratory Directed Research and Development Program at LLNL.

Abstract

Accidental or terrorist releases of hazardous materials into the atmosphere can impact large populations and cause significant loss of life or property damage. Plume predictions have been shown to be extremely valuable in guiding an effective and timely response. The two greatest sources of uncertainty in the prediction of the consequences of hazardous atmospheric releases result from poorly characterized source terms and lack of knowledge about the state of the atmosphere as reflected in the available meteorological data. In this report, we discuss the development of a new event reconstruction methodology that provides probabilistic source term estimates from field measurement data for both accidental and clandestine releases.

Accurate plume dispersion prediction requires the following questions to be answered: What was released? When was it released? How much material was released? Where was it released? We have developed a dynamic data-driven event reconstruction capability which couples data and predictive models through Bayesian inference to obtain a solution to this inverse problem. The solution consists of a probability distribution of unknown source term parameters. For consequence assessment, we then use this probability distribution to construct a "composite" forward plume prediction which accounts for the uncertainties in the source term. Since in most cases of practical significance it is impossible to find a closed form solution, Bayesian inference is accomplished by utilizing stochastic sampling methods. This approach takes into consideration both measurement and forward model errors and thus incorporates all the sources of uncertainty in the solution to the inverse problem. Stochastic sampling methods have the additional advantage of being suitable for problems characterized by a non-Gaussian distribution of source term parameters and for cases in which the underlying dynamical system is nonlinear.

We initially developed a Markov Chain Monte Carlo (MCMC) stochastic methodology and demonstrated its effectiveness by reconstructing a wide range of release scenarios, using synthetic as well as real-world data. Data for evaluation of our event reconstruction capability were drawn from the short-range Prairie Grass, Copenhagen, and Joint Urban 2003 field experiments and a continental-scale real-world accidental release in Algeciras, Spain. The method was tested using a variety of forward models, including a Gaussian puff dispersion model INPUFF, the regional-to-continental scale Lagrangian dispersion model LODI (the work-horse real-time operational dispersion model used by the National Atmospheric Release Advisory Center), the empirical urban model UDM, and the building-scale computational computational fluid dynamics code FEM3MP. The robustness of the Bayesian methodology was demonstrated via the use of subsets of the available concentration data and by introducing error into some of the measurements. These tests showed that the Bayesian approach is capable of providing reliable estimates of source characteristics even in cases of limited or significantly corrupted data.

For more effective treatment of strongly time-dependent problems, we developed a Sequential Monte Carlo (SMC) approach. To achieve the best performance under a wide range of conditions we combined SMC and MCMC sampling into a hybrid methodology. We compared the effectiveness and advantages of this approach relative to MCMC using a set of synthetic data examples.

Our dynamic data-driven event reconstruction capability seamlessly integrates observational data streams with predictive models, in order to provide the best possible estimates of unknown source term parameters, as well as optimal and timely situation analyses consistent with both models and data.

This new methodology is shown to be both flexible and robust, adaptable for use with any atmospheric dispersion model, and suitable for use in operational emergency response applications.

1 Introduction

Accidental or terrorist atmospheric releases of hazardous material pose great risks to human health and the environment. Examples range from the continental-scale Chernobyl nuclear power plant accident in 1985 to the recent anthrax attacks in the United States, as well as relatively common toxic industrial chemical plant and transportation accidents. In the event of an atmospheric release of radioactive, chemical, or biological materials, emergency responders need timely transport-and-fate predictions, which predict the current and future locations and concentrations of material in the atmosphere and deposited on the ground. Such predictions help responders to make time-critical decisions regarding precautions for their own safety, plans for evacuation or shelter-in-place, the design of efficient field measurement sampling plans, treatment of affected populations, attribution, and decontamination of the affected area.

Accurate dispersion calculations require proper characterization of the release source term (e.g., the location, time, type, and quantity of material released) and knowledge of the relevant meteorological parameters representing the state of the atmosphere derived from observations and/or numerical weather prediction model results. In this report, we discuss the development and application of a new event reconstruction method that couples field measurements with dispersion model predictions in order to solve the inverse problem, estimate unknown source term parameters, and reduce plume prediction uncertainties. Our goal is the creation of an efficient, robust and reliable approach, which is applicable to a wide range of possible release scenarios and models and is suitable for operational use in emergency response.

In real-world events, the first indication that a release has occurred may come from sensors, visual observations, or even casualties. To accurately predict the resulting impacts, we first need to reconstruct the event by answering the following critical questions: What was released? How much material was released? When and where was it released? Inaccurate estimation of the source term can lead to gross errors and/or time delays during a crisis, with an associated loss of lives. Current emergency response methods rely on first responders or analysts to estimate source characteristics. As measurement data become available, modelers conduct trial-and-error simulations in order to match model output and data in order to improve plume prediction results.

An example of such a manual event reconstruction was performed for the Algeciras Cesium-137 release by operational staff at the National Atmospheric Release Advisory Center (NARAC). During 1998 May-June, elevated radiation readings of more than 1000 times background were reported in Switzerland, France, and Italy. The time resolution of this data were very poor, with many of the sensors reporting only one-to-two week integrated quantities. NARAC expert operations staff performed repeated manual simulations over several days in order to localize the possible release site to an area encompassing southern Spain and a nearby area in northern Africa. At approximately the same time, traces of Cesium-137 were discovered in the smoke stack filters of a Spanish steel mill at Algeciras, Spain just north of the straits of Gibraltar. Using this location, NARAC analysts then spent several additional days to manually estimate the time and quantity of the release via repeated model simulations and comparisons to data. The NARAC source estimate was later found to agree with the final Spanish estimates of an 8-80 Curie release from a medical source accidentally melted during a six-hour time window (Vogt et al., 1999).

Although successful, the Algeciras event reconstruction was difficult, time-consuming, labor intensive, and heavily dependent on expert analyst judgment. For complex events, manual and other traditional approaches (e.g., regression, inversion, optimization) have serious limitations. Typically these methods yield only a single "best" solution and provide no information on the full possible range of solutions and uncertainties. They are difficult to solve for applications involving large non-linear systems and often function poorly for sparse, high-volume, or high-frequency data streams and problems requiring the integration of disparate data types.

The coupling of Bayesian inference with stochastic sampling methodologies provides a powerful alternative approach. Bayesian methods reformulate the inverse event reconstruction problem into a solution based on efficient sampling of an ensemble of simulations, guided by statistical comparisons with data. We have developed such an event reconstruction capability that seamlessly couples a time-dependent data stream with predictive models in a manner that allows dynamic improvement in estimates of the source term parameters as data become available. This methodology provides probabilistic estimates, which in turn are used to produce a composite plume prediction, with inherent uncertainty quantification.

Our event reconstruction capability can be used to more efficiently and effectively respond to hazardous atmospheric releases and to derive the maximum possible information from expensive detection, warning, and incident characterization systems. The atmospheric release inverse problem also provides an ideal application for the development of general data-driven simulation methods, because the full solution requires the treatment of time-dependent, non-linear, high-dimensional, multi-scale, and turbulent (natural variability) behavior.

2 Inverse Methods

Given a complete description of a physical system, we can predict the outcome of a measurement via a ”forward” simulation which calculates the future state of a physical system by solving a set of equations that govern its evolution from an initial state, based on forcing and boundary conditions. The inverse problem consists of using the result of some observations(s) / measurement(s) to infer the values that characterize the initial state and forcing of the system. The particular atmospheric dispersion problem of interest in this report is the determination of unknown release source characteristics, which are often the primary source of uncertainty in real-world hazardous airborne release events.

Inverse methods have a long history in geophysical applications. The use of inverse methods in studies of global biogeochemical cycles is addressed by Kasibhatla et al. (2000). More general summaries of inverse methodologies and their application to constituent transport in oceans and atmosphere are offered by Wunsch (1996); Bennett (2002) and Enting (2002) in recently published books. An exhaustive review of optimization methods for parameter estimation has recently been presented by Aster et al. (2005).

A variety of approaches to solving the atmospheric dispersion inverse problem have been explored including non-linear optimization, back-trajectory, Green’s function, adjoint, and Kalman filter methods. However, these methods often fail due to the inherent complexities, high-dimensionality, and/or non-linearity of the underlying physical system. For example, Bennett (2002) points out that the adjoint approach introduces errors due to linearization. Similarly, if empirical algorithms are used in a model (e.g., Gaussian puff splitting), construction of an appropriate adjoint may be problematic (Errico, 1997). Traditional approaches also typically yield only a single “best” answer, even though the inverse problem often does not have a unique solution (Tarantola, 2005). They cannot account for non-Gaussian error distributions and have particular weaknesses for sparse (poorly-constrained) data problems, as well as the high-volume (potentially over-constrained) and diverse data streams anticipated in the near future.

The coupling of Bayesian inference with stochastic sampling methodologies provides a powerful alternative solution to the inverse problem. Bayesian inference uses data to infer the probability of a proposed hypothesis. Good general references are “Bayesian Theory” by Bernardo & Smith (1994) and “Bayesian Data Analysis” by Gelman et al. (2003). The application of such methods to geophysical (seismic) inverse problems was pioneered by Tarantola (1989). Advances in high-performance computing architectures and wider availability of such platforms, make the use of Bayesian methods tractable for atmospheric applications (Kandlikar, 1997; Rodgers, 2000).

We have developed a Bayesian event reconstruction methodology that seamlessly integrates observational data streams with predictive models. This approach is based on the generation of an ensemble of predictive simulations, derived by efficient sampling of unknown input or forcing parameters guided by statistical comparisons with concentration measurements. The new methodology also provides the means to assign measures of confidence to plume predictions. Our approach is robust, flexible, and able to deal with complex non-linear systems and to solve for parameters that exhibit non-Gaussian distributions. Multi-scale (e.g., continental, regional, urban, building) applications and a wide range of source types (e.g., single or multiple release locations; point, area, or volume sources; moving vehicles) are supported, as appropriate to terrorist, accidental, or battlefield dispersal of radiological/nuclear materials, industrial chemicals, and biological or chemical agents.

3 Bayesian Inference with Markov Chain Monte Carlo

We first introduce notation and definitions that are used in this report. Let θ and y be two random variables and define:

$p(y)$ = the probability distribution of y

$p(\theta, y)$ = the joint probability distribution of θ and y

$p(\theta \,|\, y)$ = the probability distribution of θ conditional on y

Then the following relationships are known to hold:

(1) If the random variables θ and y are independent, then $p(\theta, y) = p(\theta)p(y)$.

(2) Given the joint distribution of θ and y, the *marginal* distribution of y is given by integrating over θ,

$$p(y) = \int_\theta p(d\theta, y)$$

If θ is a discrete random variable with possible values $\theta_1, \ldots, \theta_n$, then $p(y) = \sum_{i=1}^n p(\theta = \theta_i, y)$.

(3) The joint distribution, the conditional distribution, and the marginal distribution are related by:

$$p(\theta, y) = p(\theta \,|\, y)p(y) = p(y \,|\, \theta)p(\theta).$$

3.1 Bayes' Theorem

Reverend Thomas Bayes' (1702–1761) theorem relates the conditional probability of an event θ occurring, conditioned on the fact that an another event y has occurred, to the probability of event y occurring conditioned on the fact that event θ has occurred. Bayes' theorem can be written as

$$p(\theta \,|\, y) = \frac{p(y \,|\, \theta)p(\theta)}{p(y)} \propto p(y \,|\, \theta)p(\theta). \tag{1}$$

One can think of θ as representing possible model configurations (parameters) and y as the observed data. Then $p(y \,|\, \theta)$ is the probabilistic relationship between the observed data y and a model configuration θ, and is referred to as the *likelihood*. The distribution $p(\theta)$ is called the *prior distribution* representing the *a priori* distribution of model parameters θ. The desired end result is the *posterior distribution* of θ given the data y, $p(\theta \,|\, y)$, which represents the possible set of model configurations given (conditional on) the observed data.

Numerical evaluation of the denominator of Equation (1)

$$p(y) = \int p(y \,|\, \theta)p(d\theta).$$

can be prohibitively expensive, especially if the forward model is computationally intensive and the dimensionality of $\boldsymbol{\theta}$ is high. Fortunately, sample-based inferencing makes this computation unnecessary. Instead realizations are generated from the (unscaled) posterior distribution which in turn can be used to compute quantities of interest such as the mean, variance, and mode according to

$$\begin{aligned}
&\text{Mean:} \quad E(\theta \,|\, y) = \int_\theta \theta p(d\theta \,|\, y). \\
&\text{Variance:} \quad \mathrm{var}(\theta \,|\, y) = \int_\theta (\theta - E(\theta \,|\, y))^2 p(d\theta \,|\, y). and \\
&\text{Mode:} \quad \arg\max_\theta p(\theta \,|\, y).
\end{aligned}$$

3.2 Stochastic sampling for Bayesian inference

We use a Markov Chain Monte Carlo (MCMC) procedure based on the Metropolis-Hastings sampling algorithm to calculate the posterior distribution . A good practical introduction to MCMC is the volume ”Markov Chain Monte Carlo in Practice”, edited by Gilks et al. (1995), the book ”Monte Carlo Strategies in Scientific Computing” by Liu (2001), and the overview paper by Andrieu et al. (2003).

The Markov chains are initialized by taking samples from the prior distribution, which is generally developed from any *a priori* knowledge of the event. A forward dispersion calculation is then performed to provide an initial prediction for comparison with observed data at sensors. The Metroplis-Hastings procedure then generates a Markov chain of possible samples in order to obtain the probability distributions for the unknown parameters of interest. As laid out in Table 1, a new sample (choice of unknown source parameters) is drawn from a Gaussian distribution centered at the current chain location. A forward calculation is then performed using the proposed source term parameters and results are compared to concentration measurements at the sensor locations. If the likelihood of the proposed set of source term parameters is greater than that corresponding to the previous chain location, the proposal is accepted and the Markov chain then advances to the new location. If the comparison is worse, a Bernoulli random variable (a “coin flip”) is used to decide whether or not to accept the new state. This step is critical in order to prevent chains from becoming trapped in local minima, where comparisons are more favorable than values in the local sampling area but where the chain has not converged on the ”true” global minima.

The posterior probability distribution in Equation 1 can be computed discretely from the resulting Markov chain paths as

$$p(\boldsymbol{\theta}|\mathbf{y}) \approx \pi(\boldsymbol{\theta}) = \sum_{i=1}^{n} (1/n)\delta(\boldsymbol{\theta}_i - \boldsymbol{\theta}) \tag{2}$$

which represents the probability of a particular model configuration ($\boldsymbol{\theta}$) giving results that match the observations at sensor locations ($\mathbf{y}$). Equation (2) is a sum over the entire Markov chain of length n of all sampled values $\boldsymbol{\theta}_i$ which fall within a certain “bin”, e.g., $\delta(\boldsymbol{\theta}_i - \boldsymbol{\theta}) = 1$ when $\boldsymbol{\theta}_i = \boldsymbol{\theta}$ and 0 otherwise. If a Markov chain spends several iterations sampling the same sampled values $\boldsymbol{\theta}_i$ (e.g., multiple new proposals are rejected because the given sample is more favorable than the new proposals), then the value of $p(\boldsymbol{\theta}|\mathbf{y})$ includes multiple contributions from that sample value in the summation in Equation (2).

There are two aspects of MCMC sampling that affect the overall statistical efficiency of the process (and the effective sample size) - the burn-in period and the chain’s autocorrelation. The burn-in period represents the number of samples needed for the Markov chain to relax from its initial condition and reach the stage in which it is sampling from the target distribution $\pi(\boldsymbol{\theta})$. These initial samples must be discarded and not used for inferencing in the the summation in Equation (2).

Table 1: Metropolis-Hastings algorithm used in Markov Chain Monte Carlo

- Given a current state $\boldsymbol{\theta}_i$, draw a new candidate state $\tilde{\boldsymbol{\theta}}$ from the proposal distribution, $T(\tilde{\boldsymbol{\theta}}, \boldsymbol{\theta}_i)$, which can be taken to be a Gaussian distribution centered at the previous accepted state.
- Compute the acceptance ratio as

$$\rho(\tilde{\boldsymbol{\theta}}, \boldsymbol{\theta}_i) = \frac{\pi(\tilde{\boldsymbol{\theta}})T(\boldsymbol{\theta}_i|\tilde{\boldsymbol{\theta}})}{\pi(\boldsymbol{\theta}_i)T(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta}_i)}$$

  where $\pi(\boldsymbol{\theta})$ is the discrete approximation of the posterior distribution defined in Equation (2).
- Compute an acceptance probability $\alpha(\tilde{\boldsymbol{\theta}}, \boldsymbol{\theta}_i)$

$$\alpha(\tilde{\boldsymbol{\theta}}, \boldsymbol{\theta}_i) = \min\Big(\rho(\tilde{\boldsymbol{\theta}}, \boldsymbol{\theta}_i), 1\Big)$$

- Draw u from the uniform distribution $U[0, 1]$ and update the state to $\boldsymbol{\theta}_{i+1}$ according to the following procedure

$$\boldsymbol{\theta}_{i+1} = \begin{cases} \tilde{\boldsymbol{\theta}} & \text{if} \quad \mathrm{u} \leq \alpha(\tilde{\boldsymbol{\theta}}; \boldsymbol{\theta}_{\mathrm{i}}) \\ \boldsymbol{\theta}_i & \text{otherwise} \end{cases}$$

The Markov chain realizations $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(N)}$ do not form an independent set of samples, since sequential realizations of $\pi(\boldsymbol{\theta})$ are obviously correlated. The degree of auto-correlation depends on how well the proposal distribution is able to "mix" the sample as well as the acceptance rate associated with the proposal distribution. A proposal distribution that alters the chain too little at each step ($\tilde{\boldsymbol{\theta}}$ too close to $\boldsymbol{\theta}$) results in a MCMC sample that is highly auto-correlated. On the other hand, if the proposal distribution makes large changes at each step, a low acceptance ratio typically results and the chain remains trapped in the same state for a long period of time, again resulting in high degree of auto-correlation in the final sample. The optimal proposal distribution is somewhere in between these two extremes. As a rule of thumb, an acceptance rate around 25% is thought to be appropriate for multi-dimensional problems (Gelman et al., 2003, page 306).

3.3 Likelihood Function

The likelihood function

$$p(\mathbf{y}|\boldsymbol{\theta}) \propto \mathcal{L}(\boldsymbol{\theta}) \tag{3}$$

is used to quantify the agreement between the model configuration and the data. The likelihood function is typically defined by

$$\ln\left[\mathcal{L}(\boldsymbol{\theta})\right] = \frac{\sum_i^N((C_i^M - C_i^E)^2)}{2\sigma_{rel}^2} \tag{4}$$

where C_i^M represent concentrations predicted by a dispersion model at locations i, C_i^E are the experimentally observed data, and σ_{rel} is the standard deviation of the combined forward model and measurement errors. The squared difference is summed over the N sensor locations. For the event reconstruction application, we use the natural logarithm of the model and data values in this formula. This prevents large concentration values from dominating the likelihood calculation, since the range of concentration data typically spans many orders of magnitude.

The likelihood function is calculated as the forward model for each proposed new state (sample source location x,y, and rate q values) is computed. As per the MCMC process described above, the proposed state is accepted if either

$$\mathcal{L}_{prop} > \mathcal{L} \quad \text{or} \quad \mathcal{L}_{prop} - \mathcal{L} \geq rand(0, 1] \tag{5}$$

where $\mathcal{L}_{prop}$ is the likelihood of the proposed state, $\mathcal{L}$ is the previous likelihood value, and $rand$ denotes a random number generated from a uniform distribution.

3.4 Necessary Statistical Convergence Criterion

Multiple chains are used in order to obtain better statistical sampling of the parameter space and to enable convergence monitoring (thus Equation (2) is overly simplified). Following Gelman et al. (2003), statistical convergence is monitored by computing the ratio of the variance of sampled parameters within one chain to the variance among different chains.

If there are m Markov chains of length n we can compute the between-chain variance B as

$$B = \frac{n}{m-1} \sum_{j=1}^{m} (\overline{\boldsymbol{\theta}}_j - \overline{\boldsymbol{\theta}})^2 \tag{6}$$

where

$$\overline{\boldsymbol{\theta}}_j = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\theta}_{ij} \tag{7}$$

is the average value along each Markov chain (a sample from a given chain is denoted by $\boldsymbol{\theta}_{ij}$) and

$$\overline{\boldsymbol{\theta}} = \frac{1}{m} \sum_{j=1}^{m} \overline{\boldsymbol{\theta}}_j \tag{8}$$

is the average of the values from all Markov chains. The within-chain variance W is given by

$$W = \frac{1}{m} \sum_{j=1}^{m} s_j^2 \tag{9}$$

where

$$s_j^2 = \frac{1}{n-1} \sum_{i=1}^{n} (\boldsymbol{\theta}_{ij} - \overline{\boldsymbol{\theta}}_i)^2 \ . \tag{10}$$

An estimate of the variance of $\boldsymbol{\theta}$ is then derived from B and W according to

$$\text{var}(\boldsymbol{\theta}) = \frac{n-1}{n} W + \frac{1}{n} B \tag{11}$$

and the convergence parameter, R, is computed as

$$R = \frac{\text{var}(\boldsymbol{\theta})}{W} \ . \tag{12}$$

The necessary condition for statistical convergence to the posterior distribution is that R approaches unity (Gelman et al., 2003). In practice, this is not always a sufficient condition as will be seen later (Section 4.

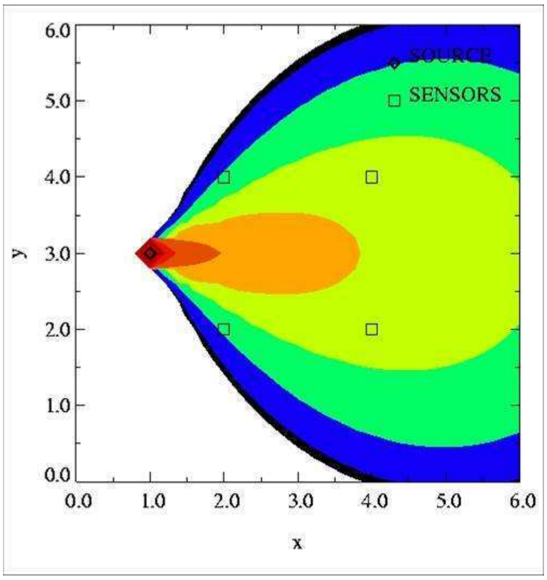


Figure 1: Plume generated with the INPUFF code and used to create synthetic sensor data. Red color contours represent higher concentration levels and blue colors reflect low concentrations.

4 MCMC Atmospheric Release Event Reconstruction

We developed a flexible and computationally efficient event reconstruction framework that enables rapid integration of a wide range of forward dispersion models and stochastic sampling methods. The computational framework was built up and tested in stages. For rapid prototyping, we implemented a relatively simple short-range dispersion model and verified our implementation using synthetic (model-generated) data and homogeneous conditions. We then proceeded with the integration of a more complex operational dispersion Lagrangian particle model and conducted tests using regional-scale tracer experiments and a real-world accidental atmospheric release. For complex urban environments, data from urban field studies were used with an empirical urban model and a full-physics building-resolving computational fluid dynamics model.

4.1 Synthetic Data Examples

For our initial development we used a Gaussian puff dispersion model INPUFF and model-generated synthetic data (for simplicity, the synthetic data was generated by INPUFF). These examples demonstrate the ability of the Bayesian inference stochastic sampling approach to simultaneously estimate both source locations and release rates and provide probabilistic solutions consistent with the available data.

The example domain was taken to be a flat $6km \times 6km$ square domain. The wind flows from left to right and is uniform over the domain. The "ground-truth" scenario releases material from a point location for one hour at a constant release rate. Synthetic data were generated for each sensor as six ten-minute averages covering the hour release period. The scenario plume at the end of the one-hour period is shown in Figure 1.

In the first application of our MCMC methodology, a 2×2 square sensor array was

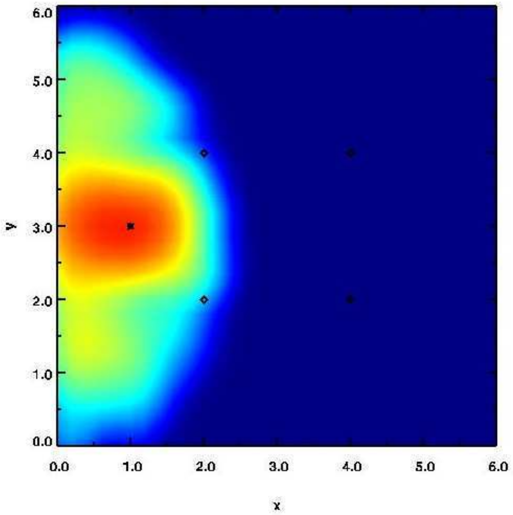


Figure 2: Posterior probability distribution of the source location for a synthetic data case using a square sensor array (four sensors, represented by diamonds). The source location is indicated by the black dot at the coordinates (1.0, 3.0). Red color contours represent high probability density and blue color contours represent low probability density.

used to reconstruct the event. An uninformed prior distribution (a flat distribution covering the entire simulation domain) was chosen as the starting point. The posterior distributions for the source location and release rate are presented in Figures 2 and 3.

Our second example demonstrates the ability of the methodology to tackle problems that do not have a unique solution and detect all possible source term parameter combinations consistent with the observed data. Synthetic data were generated for three sensors located on a line parallel to the wind direction with the source offset with respect to that line. By symmetry, it is obvious that the sensor measurements are consistent with two possible source locations. This non-unique solution is clearly seen in the posterior probability distribution presented in Figure 4, which has two modes corresponding to the two equally probable source locations. Figure 5 shows the four Markov Chains used in reconstructing the source. All the chains are well-mixed, i.e., both chains thoroughly explore both equally probable source locations.

The final synthetic example is based on synthetic data from three sensors arranged in a triangular array Figure 6. The impact of different choices of sensor array configurations on the posterior probability distribution can be compared to the inverse solutions based on the square sensor array (Figure 2). The smaller three-sensor array results in a broader posterior probability distribution for the location indicating greater uncertainty. More detailed study of the effects of sensor array size and configuration were pursued using the Copenhagen tracer study data set discussed in Section 4.3.

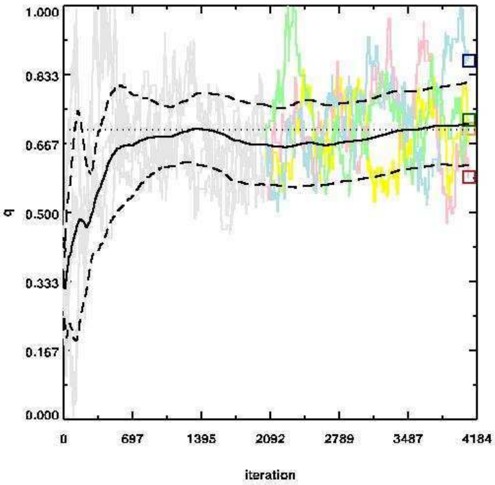


Figure 3: Source release rates for a synthetic case based on concentrations from a four-sensor square array using four Markov chains. Dotted line - actual release rate; solid line - mean of reconstructed release rates from four Markov chains; colored lines - release rates corresponding to each Markov chains; dashed lines - +/− one standard deviation from the mean value (solid line).

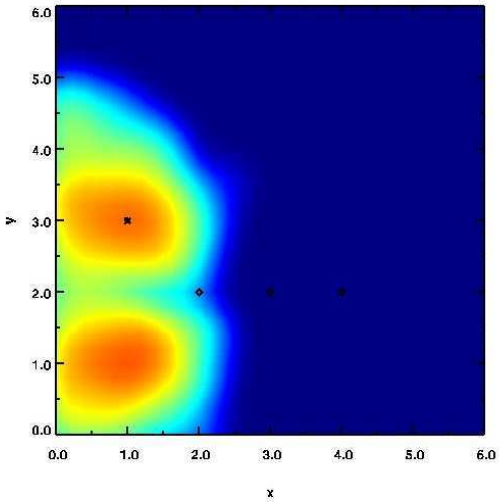


Figure 4: Bimodal posterior probability distribution of the source location for a synthetic data case using concentrations from a linear array of three sensor (black diamonds). The source location is indicated by the black dot at the coordinates (1.0, 3.0). Red color contours represent high probability density and blue color contours represent low probability density.

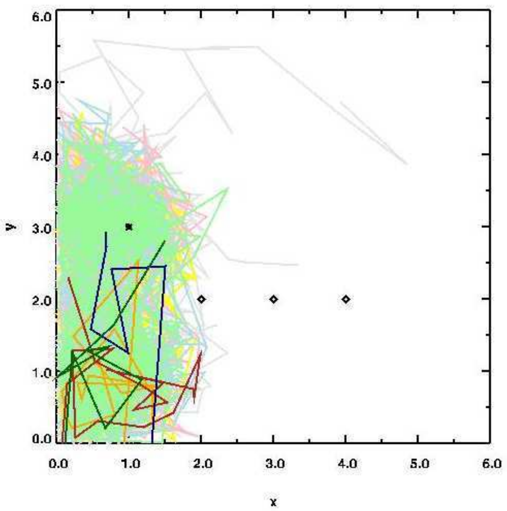


Figure 5: Markov chains explore two posterior probability distribution modes for the source location for a synthetic data case using concentrations from a linear array of three sensors. Light colors represent early stages of MCMC iteration procedure and dark colors represent late stages.

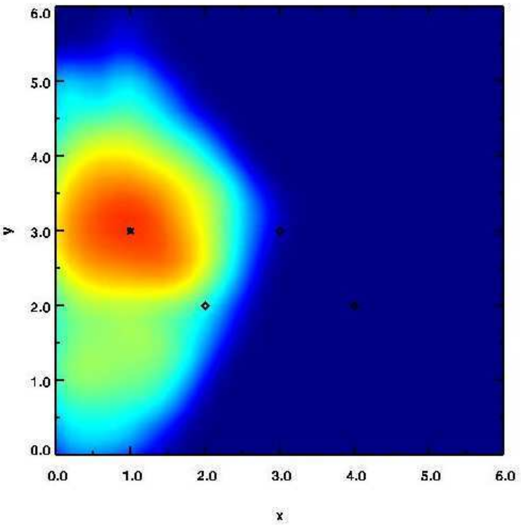


Figure 6: Posterior probability distribution of the source location for a synthetic data case using concentrations from a triangular sensor array. Red color contours represent high probability density and blue color contours represent low probability density.

4.2 Prairie Grass Field Study

As the next step, we extended the MCMC system to incorporate a three-dimensional Lagrangian particle dispersion code (LODI), which is a core operational model of the National Atmospheric Release Advisory Center (NARAC) used for emergency response applications. An analysis of LODI numerical errors (discretization and spatial smoothing) was carried out in order to account for the uncertainties involved in the inversion procedure. We first tested the MCMC-LODI system using data from a local-scale atmospheric tracer study, Project Prairie Grass.

The Prairie Grass field experiment (Barad, 1958, Ed.) has become the standard dataset for the evaluation of short-range (1 *km*) atmospheric dispersion. Multiple near-surface releases were carried out under a wide range of atmospheric conditions (ASTM, 2000). The site consisted of a flat agricultural field with short dry stubble grass. Neutrally buoyant sulfur dioxide was released for ten minutes from a small tube at a height of 0.46 m and integrated concentrations were then measured downwind at a height of 1.5m on five monitoring arcs located 50, 100, 200, 400, and 800 m from the source. Observations were reported as one ten-minute average per sensor. The lack of time resolution makes this dataset particularly challenging for event reconstruction. In particular, we expect larger uncertainties in the estimated source location, especially for the streamwise direction.

A test of the event reconstruction capability was performed using release 5 of the Prairie Grass field study. Experiment 5 is typical of slightly unstable atmospheric boundary layer conditions corresponding to Pasquill-Gifford stability class B. Winds were from the south (176 degrees) at 6.5 m/s, as measured at 2m. A wind field for the domain was constructed using this observation, a roughness length of 0.08, an inverse Monin-Obukhov length of -0.03, a boundary-layer height of 1100m, a surface-layer height of 16m, and a friction velocity of 0.39 using ADAPT (Sugiyama & Chan, 1998), the NARAC meteorological data preprocessor. This wind field was provided to LODI as a basis for LODI's forward runs, which used 10000 particles per simulation.

Sensor characteristics were considered in the handling of data in the reconstruction process. The minimum sensitivity level of the sensors is 10^{-8} kg/m^3 and the saturation level is 10^{-3} kg/m^3. The measurements C_{ij}^E and modeled concentrations C_{ij}^M were first compared to the detection range of the instruments used. Any measurements or modeled concentrations above the saturation level of the instrument were set to the saturation level; any measurements or modeled concentrations below the detection limit were set to the detection limit or sensor sensitivity threshold.

Reconstruction of the source location and release rate were based on comparison of sensor data with predictions from LODI. Although 138 of the 545 sensors reported measurements of the plume, we only used data from eight randomly-selected sensors on the two outer-arcs (400m and 800m from the source). Four of these sensors reported no measurements, which helped to bound the plume. The sensors which detected the plume had values ranging from 3.6×10^{-8} to 1.074×10^{-6} kg/m^3. The event reconstruction used a likelihood function based on the base ten logarithmic differences of the measured and predicted concentrations.

The reconstruction of source location and release rate was carried out with four Markov chains and 2000 iterations per chain. The reconstruction calculates a probability distribution for the source region that includes the actual source location, but the contours of high probability extend in the direction of the mean flow due to the lack of time resolution in the sensor data (Figure 7) and there is a wide range in possible source strengths (Figure 8). Other studies (Lundquist et al., 2005) show that this "smearing" in the direction of the flow does not occur if sensor data at a higher time resolution is used in the reconstruction.

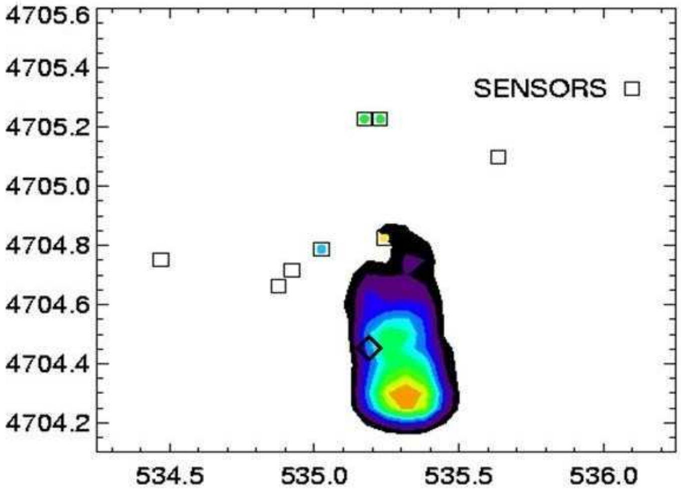


Figure 7: Probability distribution for the source location for Prairie Grass release 5. The x and y axes denote west-east and south-north location in km. Squares indicate the locations of the sensors, while the color filling the squares denotes the observed concentration (white indicates a zero observation, and blue, green, and yellow indicate low, medium, and high concentrations. The colored contours indicate the posterior probability distribution for the source location, with orange indicating the highest probability area. The actual source location, marked by a triangle, lies within the 50th percentile confidence contour.

The estimated source release rate shown in (Figure 8) is bimodal - the lower range of release rates includes the actual source release rate, while the higher release rate mode corresponds to a possible source upwind of the actual release. We demonstrate this by estimating the source location assuming a known release rate. The resulting conditional probability for the source location (Figure 9) shows the expected reduction in uncertainty of the reconstructed source location, with the actual release site lying within the 90th percentile confidence interval.

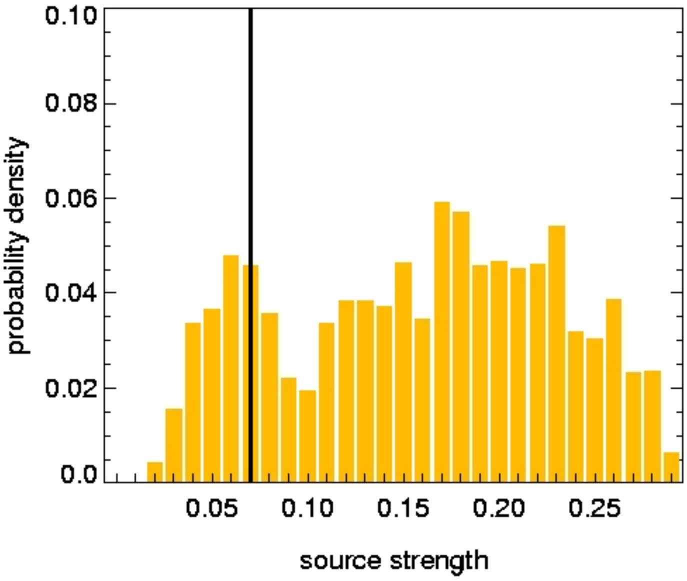


Figure 8: Probability distribution of source release rate for Prairie Grass release 5. The actual release rate was $q = 0.05$ g/s.

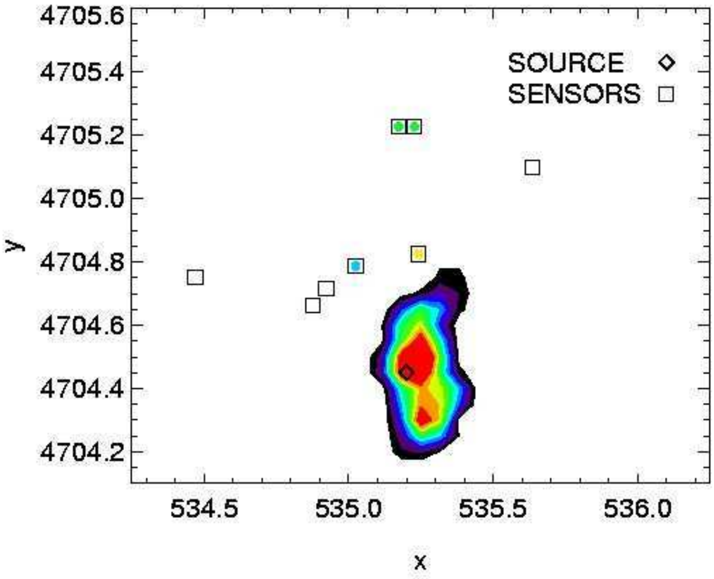


Figure 9: Conditional probability of the source location conditioned on the actual release rate for the Prairie Grass release 5. Axes and symbols are as in Figure 7. Instead of including all accepted locations, the probability contours only depict the locations from accepted proposals with source strengths close to the actual value. Warmer colors indicate higher probability; the real source location lies within the 90th percentile confidence interval.

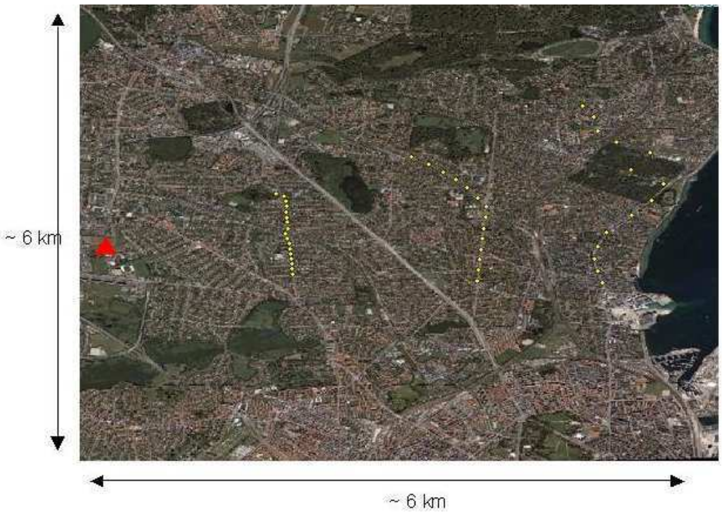


Figure 10: Domain of the Copenhagen experiment. The red triangle denotes the actual source location and the yellow dots indicate the three arcs of SF_6 sensor locations. The most distant sensors are approximately 6km from the source.

4.3 Copenhagen Tracer Experiment

The Copenhagen experiment, carried out in Denmark in 1978 and 1979, provides both a larger domain (6km from the source to the most distant sensor) and time resolution in the sensor data. The complete Copenhagen dataset documents eleven 1-hr duration releases of SF6 from a height of 115 m. SF6 is measured downwind along arcs 2 km, 4 km, and 6 km from the source. Three 20-minute averages of SF6 concentration are available from each sensor. A detailed technical description of the experiment and experimental methods is given in Gryning (1981) and Gryning & Lyck (1984).

The upwind area included some industrial, farmland, and moor areas, but was characterized as mostly residential, while the downwind sector is residential (see Figure 10). The roughness length for the area is estimated to be about 0.6m, based on vertical wind variances and mean wind profiles, which were obtained from the same tower from which the tracer was released. All releases occurred during the day, and stability estimates indicate neutral to moderately convective conditions. Release rates varied from 2.4 to 4.7 g s^{-1}. Wind speed, wind direction, and temperature were measured on the release tower at heights ranging between 2 and 200 m; turbulent quantities were measured at 115m. Winds were typically from the west or north-west. Boundary-layer height was estimated from a radiosonde released at Jægersborg, 5 km NE of the release tower.

The SF6 detectors exhibited a 20% systematic error and a 2% random error. The sensitivity level of the sensors was given in Gryning (1981) as 2×10^{-12} ppb, or 12.5 ng m^{-3}. Accordingly, the lowest concentration level used for the reconstruction was set to 10 ng m^{-3}. No information on instrument saturation was provided, so the ceiling for measured concentrations was set just above the highest observed level (7669 ng m^{-3}) at 10^5 ng m^{-3}.

We chose Intensive Operational Period 10 (IOP10) for our event reconstruction example, because it utilizes the largest number of sensors (45) and represents a case in which the plume spread is entirely encompassed by the sensor arcs. For IOP10, 39 sensors reported hits and 6 sensors reported misses. Winds varied slightly throughout the three hours of IOP10, veering from south-south-westerly to south-westerly and increasing from $4.6 m/s$ to $5.7 m/s$ at the surface, and from $9.9 m/s$ to $11.45 m/s$ at $200 m$.

The source term parameters were estimated using the 20-minute integrated concentration data collected by the sensors commencing one-half to one hour after the release began, as well as hourly-averaged wind fields developed using the ADAPT meteorological data assimilation model. Fixed values included the release start time and duration, the release height, and the knowledge that the release rate was constant throughout the duration of the release. Because the domain is relatively small, we assumed that the source started releasing in the time period in which the first measurements are available. The source term parameter reconstruction determined horizontal, x and y coordinates of the static point source and the release rate q.

The actual source was located at [2,5] km with the center of the domain [5,5] km located just upwind of the second arc. A flat, uninformed prior distribution was used for both the location and the release rate. The prior for the source location was limited to the domain of interest, a $6km$ by $6km$ square, that includes the source and all of the sensors. The prior distribution for the release rate was bounded from below by the sensitivity of the sensors to ($10\ ng\ m^{-3}$) and from above by the assumed saturation level ($10^5\ ng\ m^{-3}$). The release rate was allowed to range (unreasonably widely) between 10 $\mu g\ s^{-1}$ and 1000 $kg\ s^{-1}$ for each forward run.

Each forward run of the NARAC Lagrangian particle dispersion model, LODI, used 10000 particles. Concentrations in each 100 $m\times$ 100 m grid cell extending from the surface to 10m above the surface were calculated for each forward run. Predicted values were compared with observed concentrations using the following weighting method. First, the LODI grid cell containing the sensor was identified (e.g., $x = m$ and $y = n$). Then the sensor concentration ($\hat{C}$) at (m, n) was calculated by considering the concentrations (C) in neighboring cells:

$$\begin{aligned} C(m,n) = &\quad 2C(m,n) + C(m-1,n) + C(m+1,n) + C(m,n-1) + C(m,n+1) \\ + &\quad \frac{1}{2}\left(C(m-1,n-1) + C(m+1,n-1)\right. \\ &\quad \left. + C(m-1,n+1) + C(m+1,n+1)\right) \end{aligned}$$

The standard deviation of the combined forward model and measurements error, σ_{rel}, (Equation 4) was set at 0.2, to ensure a high rejection rate, implying a high confidence in both observations and model predictions and resulting in a narrower final probability distribution.

4.3.1 Base case using all available sensors

We first carried out the inversion using the data from all of the 45 sensors that reported hits during IOP10. The posterior probability distribution is presented in Figure 11. The actual source is contained within the 80 percentile confidence level, even though errors in meteorology were not explicitly considered. The estimate of the source release rate as a function of the iteration number is given in Figure 12. The thick red line represents the actual release rate and the thick black line is the mean of four Markov Chains shown by the thin colored lines. The uncertainty in reconstructed release rate is within approximately a factor of two of the actual release rate.

The measure of statistical convergence R (Equation 12) is shown in Figure 13 as a function of the iteration number. The location coordinates converge rapidly to their respective marginal posterior probability distributions ($R \approx 1$), while the release rate converges considerably slowly. This slower convergence is related to the relatively larger uncertainty in the source location in the along-wind direction, which occurs because

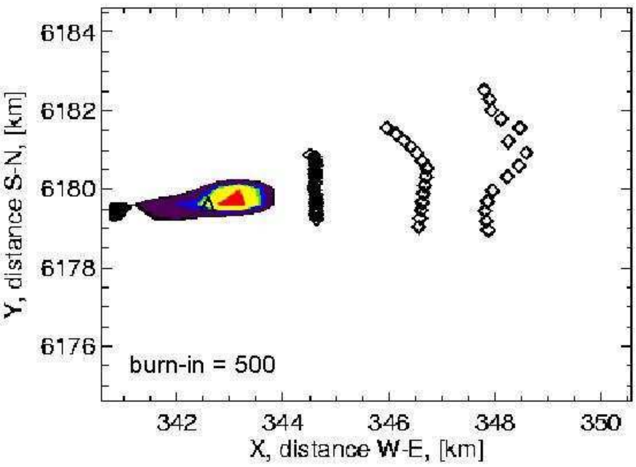


Figure 11: Probability distribution for the source location using the data from all 45 available sensors. Diamonds indicate the locations of sensors used in the reconstruction. The triangle indicates the actual source location. Colored contours show the posterior distribution for the source location, with warmer colors indicating higher confidence.

point measurements do not clearly distinguish between a high release rate by a distant source and a low release rate from a nearby location.

4.3.2 Effect of using fewer sensors

In an actual event, it is unlikely that 40+ sensors would be available. To stress our method, a reconstruction was carried out for the same IOP using only nine of the 45 available sensors. Since our Prairie Grass and synthetic data investigations revealed the importance of including some zero measurements in order to reach convergence quickly, the nine chosen sensors bounded the plume: three sensors reported data and six reported no data. The posterior probability distribution successfully predicts the actual release location to within the 50-percentile confidence level (Figure 14).

In Figure 15, the statistical convergence parameter R is presented. We observe that the location coordinates again converge rapidly to their respective marginal posterior probability distributions ($R \approx 1$), while the release rate converges more slowly and levels off at a value of $R = 3$.

4.3.3 Robustness of the source reconstruction procedure

To demonstrate the robustness of the Bayesian stochastic methodology in cases involving potentially conflicting data, we also reconstructed IOP10 using ”broken” sensors. Data from 15 randomly selected sensors (one-third of the total) were replaced with new values. Five sensors that reported zero concentrations were given false positive values; five sensors that reported above threshold data were replaced with zero concentrations (false negative values); and the third group of five sensors were given different (but possible) concentration values. Figure 16 shows the posterior probability distribution

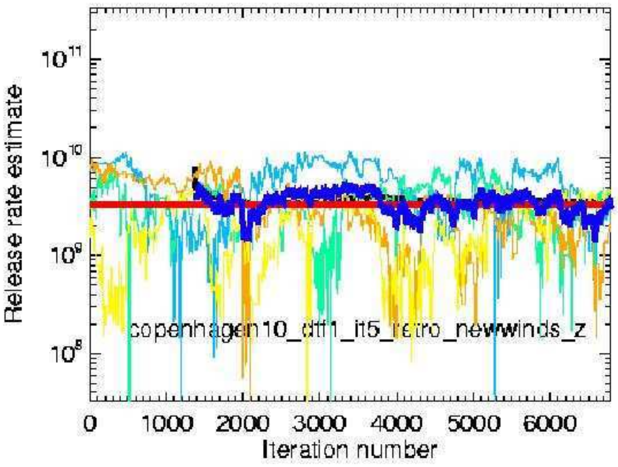


Figure 12: Reconstruction of the release rate using the data from all 45 available sensors. The thick red line is the actual release rate; the thick blue line is the mean of four Markov Chains; and the thin colored lines shows the individual chains.

for the source location. It can be seen that the "broken" sensors introduce bias in the source term inversion, such that the calculated probability distribution is translated in space in comparison to the solution based on the actual measurements. There also is a larger uncertainty in the source location as evidenced by the flatter and wider posterior probability distribution. Despite the degree of error introduced into the data, the actual source is still contained within the 10 percentile confidence level. Interestingly, the reconstruction of the release rate yields results qualitatively and quantitatively similar to that when all the available sensors are used (Figure 17).

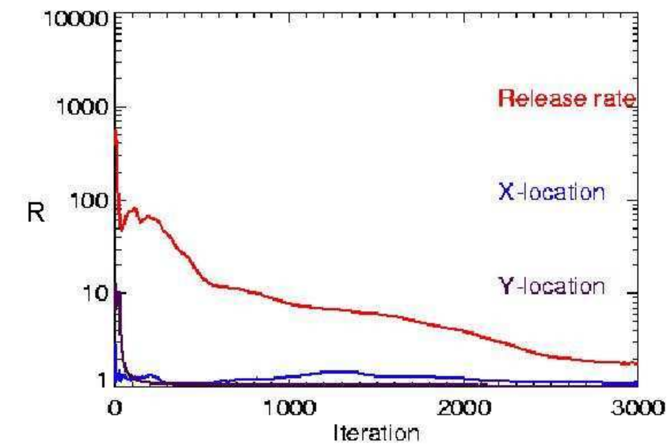


Figure 13: Convergence of the reconstruction of Copenhagen IOP10 using measurements from all 45 available sensors. The y-location parameter (purple line) converges most rapidly, while the release rate parameter (red line) only converges after 2900 iterations.

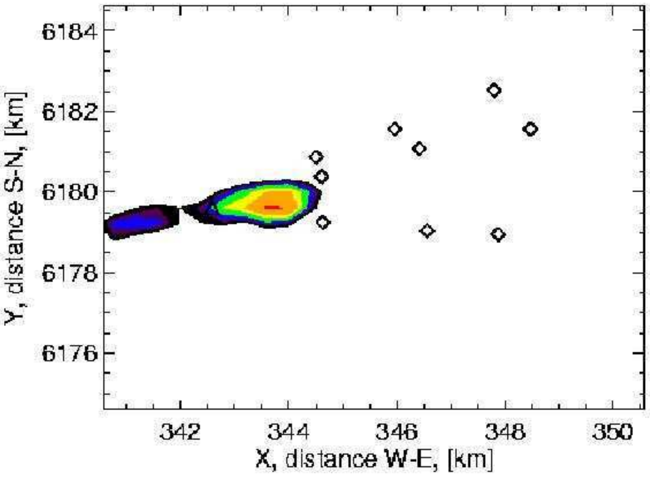


Figure 14: Probability distribution of source location, as in Figure 11, but using measurements from a reduced set of 9 sensors out of 45 available sensors.

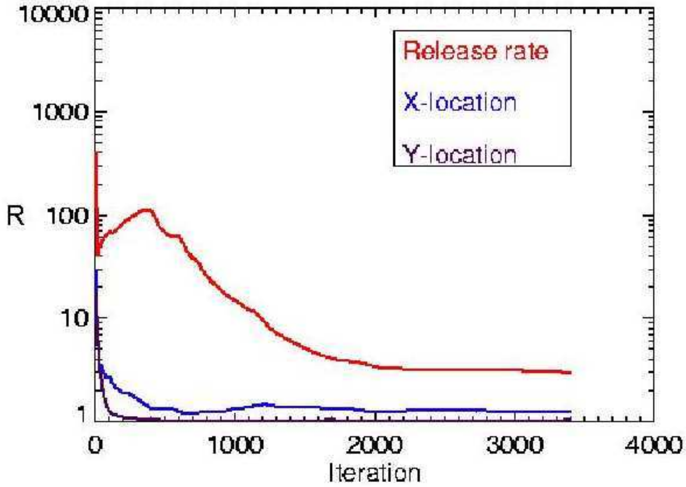


Figure 15: Statistical convergence measure, R, of stochastic procedure as a function of the number of iterations.

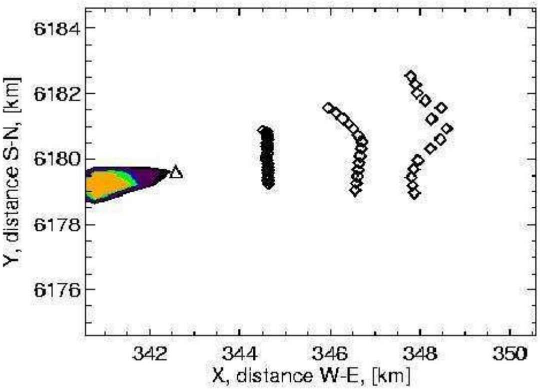


Figure 16: Probability distribution of the source location. The robustness of the methodology based on Bayesian inference with stochastic sampling is tested by solving the inverse problem using data in which 15 of the 45 sensors are assigned incorrect values.

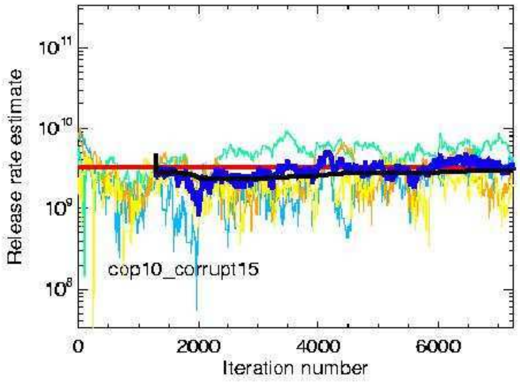


Figure 17: As in Figure 12, but for the reconstruction using data from 45 sensors, 15 of which were altered.

4.4 Continental Scale Real World Event Reconstruction

The Bayesian event reconstruction methodology was successfully applied to a continental-scale accidental release of Cesium-137 (Cs-137) from a steel mill at Algeciras, Spain in late May, 1998 (Delle Monache et al., 2007). The Bayesian MCMC approach was combined with simulated annealing and adaptive procedures to assure a robust and efficient exploration of parameter space. The simulation set-up was chosen to reflect an emergency response scenario in which only the source geometry and release time are assumed to be known. The event reconstruction process was able to estimate the likely source locations to within 100 km of the actual site, after exploring a domain covering an area of approximately 1800 by 3600 km. The source strength is reconstructed with a distribution of values of the same order of magnitude as the upper end of the range as reported by the Spanish Nuclear Security Agency. By running on a large parallel cluster, the inversion results was completed in a few hours, well within the necessary time scale for continental-scale emergency response applications. The complete text of a submitted paper (Delle Monache, L., Lundquist, J.K., Kosović, B., Johannesson, G., Dyer, K.M., Aines, R.D., Belles, R.D., Hanley, W.G., Larsen, S.C., Loosmore, G.A., Nitao, J.J., Sugiyama, G.A., Vogt, P.J., 2007: Bayesian inference and Markov Chain Monte Carlo sampling to reconstruct a contaminant source at continental scale, UCRL-JRNL-226644-DRAFT) is contained in Appendix A.

4.5 Event Reconstruction for Urban Releases

The ability to determine the source of a contaminant plume in urban environments is crucial for emergency response applications. This is an extremely difficult problem, since the transport of material is affected by the presence of individual buildings, which can divert flow in unexpected directions. However, high-resolution flow simulations are now possible for predicting plume evolution in complex urban geometries, which make it possible to apply event reconstruction to this critical application.

We integrated two building resolving urban dispersion models into the the Bayesian MCMC event reconstruction framework - a full-physics computational fluid dynamics (CFD) model FEM3MP (Finite Element Model 3 Massively Parallel) and a computationally-inexpensive empirical Gaussian puff Urban Dispersion Model (UDM). We conducted tests of the performance of the source inversion methodology for both a protoype isolated building and for tracer releases conducted during the Joint Urban 2003 field experiment in Oklahoma City.

CFD models are computationally intensive models capable of resolving details of flows and dispersion in complex urban environments. To enhance perfomance, steady-state flow fields generated by the FEM3MP model were used to drive thousands of forward dispersion simulations to create a database for the inversion procedure. We then used a Green's function approach to successfully solve the inverse dispersion problem and simultaneously determine the source location and release rate to within narrow confidence intervals. Green's function methods can be applied to dispersion problems characterized by dispersion coefficients that are not dependent on concentration, which enables the decoupling of source location and release rate inversion. Further computational performance enhancements can be obtained through the use of a reciprocal Green's function approach as described in Nitao (2004).

Using FEM3MP we were able to estimate the source location to within a block and the release rate to within a factor of two for the IOP3 tracer release of the Joint Urban 2003 field study (Chow et al., 2007). The results of the inversion indicate the probability of a source being found at a particular location with a particular release rate. A composite plume showing concentrations at the 90% confidence level can then be constructed using the realizations from the reconstructed probability distribution. The compositve plume contours can be interpreted as the likelihood of the concentration at a particular location being above or below a specified threshold value. Appendix B contains a copy of a submitted journal paper showing these and other results.

High-resolution computational fluid dynamics (CFD) models are computatoinally expensive. We therefore tested our event reconstruction method with an urban puff model developed by the United Kingdom's Defence Science Technology Laboratory's (Dstl) Urban Dispersion Model (UDM). The UDM provides rapid urban dispersion simulations by combining traditional Gaussian puff modeling with empirically deduced mixing and entrainment approximations for urban areas (Hall et al., 2003). Our event reconstuction results showed significantly reduction in the initial uncertainty in source term parameters, although model approximations resulted in some bias in the reconstructed parameters (Neuman et al., 2006). Appendix C contains a fuller discussion of this results of this work.

Although developed and used independently, event reconstruction using the UDM and the finite-element CFD code can be complementary in emergency response applications. It is possible to improve both the speed of execution and obtain high-resolution accuracy by using a combination of the two models and a staged sampling approach (Aines et al., 2002, e.g.,). The faster but less accurate model is applied to reduce the initial uncertainty. The posterior distribution from this model, then becomes the prior distribution to be used with the computationally-intensive high-resolution model to refine the final event reconstruction.

4.6 Event Reconstruction Sensor Siting

Our event reconstruction capability can be used to help design sensor networks networks for detecting and responding to atmospheric releases of hazardous materials (Lundquist et al., 2005). A quantitative measure of the reduction in uncertainty can be utilized by policy makers to determine the cost/benefit of deploying sensors.

Two numerical experiments were performed to demonstrate the utility of the event reconstruction methodology for sensor network design. In the first set of studies, only the time resolution of the sensors varied between three candidate networks. The most "expensive" sensor network offered few advantages over the moderately-priced network for the selected release. The second set of studes explored the implications of sensor detection limits, which can have a significant impact on costs. In this experiment, the expensive network most clearly defined the source location and release rate. The other networks provided insufficient data to distinguish between two possible clusters of source locations. Aggregation of the results into a composite plume can be used by decision-makers to distinguish the utility of the expensive sensor network in enhancing situation awareness.

Full details about these sensor siting studies can be found in the report (Lundquist, J.K., Kosović, B., Belles, R., 2005: Synthetic Event Reconstruction Experiments for Defining Sensor Network Characteristics. LLNL Technical Report UCRL-TR-217762) included in Appendix D.

5 Sequential and Hybrid Monte Carlo Sampling

In previous sections, we presented the development of our Bayesian methodology with MCMC stochastic sampling and demonstrated its effectiveness in atmospheric release event reconstruction. MCMC sampling was shown to be efficient for problems that are not strongly time-dependent and therefore require only limited time resolution. However, for dynamic events characterized by high frequency data streams, the MCMC algorithm becomes computationally prohibitive as it must re-incorporate all of the available data at every step of the sampling procedure.

Sequential Monte Carlo (SMC) was originally designed and developed to address the problem of sampling from a time-dependent, dynamically-evolving posterior distribution. SMC is inherently parallel and therefore suitable for efficient application on massively parallel platforms. For optimal performance, we created a hybrid methodology that takes advantage of the strengths of MCMC during the initial phase of a release when only limited amount of data is available and uses SMC at later stages as data streams increase. Technical report UCRL-TR-207173 (Johannesson et al., 2004) presented in Appendix E covers in detail the development, verification, and effectiveness of the hybrid SMC-MCMC approach.

We also explored the capabilities of the SMC methodology when applied to the reconstruction of events characterized by complex sources. Our preliminary results for cases involving multiple simultaneous point sources or moving point (vehicular) sources have shown that the hybrid sampling methodology is a promising approach to solving these complex inverse problems.

6 Summary

We have developed an atmospheric release event reconstruction methodology based on Bayesian inference combined with stochastic sampling procedures (MCMC, SMC, and hybrid). Although this approach can be computationally intensive, it is completely general and can be used for time-varying release rates, complex urban flow conditions, non-linear problems, and problems characterized by non-Gaussian distributions. The results of the inversion, specifically the posterior probability distribution, indicate the probability of a source being found at a particular location with a particular release rate. These results inherently reflect any lack of quality or spatial/temporal resolution in the observed data. For each reconstruction, a composite plume can be calculated which contains probabilistic information from the iterative inversion procedure. This plume shows the the likelihood of the concentration at a particular location being above (or below) specified threshold values.

We created a modular, scalable computational framework to accommodate the full set of stochastic methodologies (e.g., MCMC, SMC, hybrid stochastic algorithms, ”Green’s function”, ”reciprocal” methods), as well as a selection of key classes of dispersion models. This design provides a clear separation of stochastic algorithms from predictive models and supports parallelization at both the stochastic algorithm and individual model level.

After demonstrating of the feasibility of the MCMC stochastic approach using a Gaussian puff model INPUFF, we incorporated a three-dimensional Lagrangian particle dispersion code (LODI), which is a core operational NARAC dispersion model, into the stochastic inversion framework. We first tested the event reconstruction suite using integrated concentration measurements from the Prairie Grass field experiment. We sub-sampled the data by reducing the number of sensors used in the reconstruction process to quantify system performance and demonstrated the ability to reconstruct a release event with high confidence using only a few data points.

We then carried out an extensive investigation of the MCMC event reconstruction capability using an intermediate-scale field study, the Copenhagen tracer experiment. We tested the robustness of the source inversion capability even given sparse, contradictory, and/or inaccurate data. The Copenhagen dataset was then used as a starting point for a study into the design of sensor networks by using the event reconstruction capability to define the minimum necessary requirements for useful network architectures. We also demonstrated the applicability of our Bayesian methodology for continental-scale atmospheric releases by reconstructing a real-world accident in Algeciras Spain.

To address urban environments, we integrated two dispersion models into the MCMC event reconstruction framework: an empirical urban Gaussian puff model UDM and a full-physics CFD solver FEM3MP. We demonstrated the successful inversion of a prototype problem involving flow around an isolated building using the building-resolving FEM3MP model. Application of FEM3MP to the complex conditions present during IOP3 and IOP9 of the Joint URBAN 2003 experiment in Oklahoma City also proved successful despite the complex urban conditions. The advantages and limitations of the computationally efficient UDM were demonstrated using data from IOP3 of the Joint Urban 2003 experiment.

We initiated the development of an SMC methodology and demonstrated its performance advantages for treating time-dependent (dynamic) systems and high-frequency data streams. We also developed hybrid stochastic algorithms that combine the advantages of MCMC for relatively limited amounts of data with the efficiency of SMC in the later stages of an event when high-volume data streams become available.

Our stochastic methodology for dynamic data-driven airborne release event reconstruction is flexible and robust. Future extensions of this capability will incorporate time-varying releases, unsteady flow conditions, and elevated sources. Meteorological uncertainty will be incorporated to allow for errors induced by limited observational data or errors in numerical weather prediction forecasts. We will also extend the event reconstruction methodology to examine other source parameters such as the particle-size distribution, isotopic or chemical composition, and the initial source geometry. A further possible extension would be to combine our event reconstruction methodology with other inversion approaches.

Atmospheric release event reconstruction addresses immediate critical homeland and national security needs for counter-terrorism, consequence management, emergency response, attribution and attribution applications. This capability directly leverages the enormous investments being made at LLNL and other institutions to develop sensors, real-time data acquisition and communication systems, predictive models, and high performance computing. An operational event reconstruction tool will transform the way we respond to terrorist attacks, industrial and transportation accidents, and military engagements, by reducing situational awareness uncertainties and facilitating informed decision-making.

Acknowledgements

This project was funded by Lawrence Livermore National Laboratory's Laboratory Directed Research and Development Program. This program is mandated by Congress to fund director-initiated, long-term research and development (R&D) projects in support of the DOE and national laboratories mission areas. The Director's Office LDRD Program at LLNL funds creative and innovative R&D to ensure the scientific vitality of the Laboratory in mission-related scientific disciplines.

References

Aines, R., Nitao, J., Newmark, R., Carle, S., Ramirez, A., & Hanley, W. (2002). The Stochastic Engine Initiative: Improving Prediction of Behavior in Geologic Environments We Cannot Directly Observe Publication. Technical Report UCRL-ID-148221, Lawrence Livermore National Laboratory.

Andrieu, C., De Freitas, N., Doucent, A., & Jordan, M. I. (2003). An introduction to MCMC for machine learning. *Machine Learning*, *50*, 5–43.

Aster, R., Borchers, B., & Thurber, C. (2005). *Parameter Estimation and Inverse Problems*. Elsevier Academic Press.

Barad, M. (1958). Project Prairie Grass. A field program in diffusion. Geophys. Res. Paper No. 59, Vols. I and II. Technical Report AFCRF-TR-58-235, Air Force Cambridge Research Center, Bedford, MA.

Bennett, A. (2002). *Inverse Modeling of the Ocean and Atmosphere*. Cambridge Univ. Press.

Bernardo, J. M. & Smith, A. F. M. (1994). *Bayesian Theory*. Wiley.

Chow, F., Kosovic, B., & Chan, S. (2007). Source inversion for contaminant plume dispersion in urban environments using building-resolving simulations. Submitted to *Atmos. Environ.*, UCRL-JRNL-228011.

Delle Monache, L., Lundquist, J. K., Kosovic, B., Johannesson, G., Dyer, K. M., Aines, R., Belles, R. D., Hanley, W. G., Larsen, S. C., Loosmore, G. A., Mirin, A. A., Nitao, J. J., Sugiyama, G. A., & Vogt, P. J. (2007). Bayesian inference and Markov Chain Monte Carlo sampling to reconstruct a contaminant source at continental scale. Submitted to *J. Appl. Meteorol.*, UCRL-JRNL-226644-DRAFT.

Enting, I. (2002). *Inverse Problems in Atmospheric Constituent Transport*. Cambridge Univ. Press.

Errico, R. (1997). What is an adjoint. *Bull. Amer. Meteorol. Soc.*, *78*, 2577–2591.

Gelman, A., Carlin, J., Stern, H., & Rubin, D. (2003). *Bayesian Data Analysis*. Chapman & Hall/CRC.

Gilks, W., Richardson, S., & Spiegelhalter, D. (1995). *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC.

Gryning, S.-E. (1981). *Elevated Source SF$_6$-Tracer Dispersion Experiments in the Copenhagen Area*. Risø-R-446, RisøNational Laboratory. 187 pp.

Gryning, S.-E. & Lyck, E. (1984). Atmospheric dispersion from elevated sources in an urban area: Comparison between tracer experiments and model calculations. *J. Clim. Appl. Meteorol.*, *23*, 651–660.

Hall, D., Spanton, A., Griffiths, I., Hargrave, M., & Walker, S. (2003). The Urban Dispersion Model (UDM): Version 2.2. Technical Report TR04774, Defence Science and Technology Laboratory, Porton Down, Salisbury, UK.

Johannesson, G., Hanley, W., & Nitao, J. (2004). Dynamic Bayesian Models via Monte Carlo - An Introduction with Examples. Technical Report UCRL-TR-207173, Lawrence Livermore National Laboratory, Livermore, CA.

Kandlikar, M. (1997). Bayesian inversion for reconciling uncertainties in global mass balances. *Tellus*, *49B*.

Kasibhatla, P., Heimann, M., Rayner, P., Mahowald, N., Prinn, R., & Hartley, D. (2000). *Inverse Methods in Global Biogeochemical Cycles*. Washington, DC: American Geophyical Union.

Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing*. New York: Springer.

Lundquist, J., Kosovic, B., & Belles, R. (2005). Synthetic Event Reconstruction Experiments for Defining Sensor Network Characteristics. Technical Report UCRL-TR-217762, Lawrence Livermore National Laboratory, Livermore, CA.

Neuman, S., Glascoe, L., Kosovic, B., Dyer, K., Hanley, W., Nitao, J., & Gordon, R. (2006). Event Reconstruction for Atmospheric Releases Employing Urban Puff Model UDM with Stochastic Inversion Methodology. Paper J4.6, Sixth Symposium on Urban Environment, 86th American Meteorological Society Annual Meeting. Atlanta, GA. American Meteorological Society. UCRL-PROC-216842.

Nitao, J. (2004). The Use of Reciprocity in Atmospheric Source Inversion Problems. Technical Report UCRL-TR-312826, Lawrence Livermore National Laboratory.

Rodgers, C. (2000). *Inverse Methods for Atmospheric Sounding: Theory and Practice*. World Scientific.

Sugiyama, G. & Chan, S. (1998). A New Meteorological Data Assimilation Model for Real-Time Emergency Response. In *10th Joint Conference on the Applications of Air Pollution Meteorology*, Phoenix, AZ.

Tarantola, A. (1989). *Inverse Problem Theory: Methods for Data Fitting and Model Parameter Estimation*. Elsevier.

Tarantola, A. (2005). *Inverse Problem Theory: and Methods for Model Parameter Estimation*. SIAM.

Vogt, P., Pobanz, B., Aluzzi, F., Baskett, R., & Sullivan, T. (1999). ARAC simulation of the Algeciras, Spain steel mill Cs-137 release. In *Amer. Nucl. Soc. 7th Topical Meeting on Emergency Preparedness & Response*, Santa Fe, NM.

Wunsch, C. (1996). *The Ocean Circulation Inverse Problem*. Cambridge Univ. Press.

Appendix A

Bayesian inference and Markov Chain Monte Carlo sampling to reconstruct a contaminant source at continental scale

UCRL-JRNL-226644-DRAFT

UCRL-JRNL-226644-DRAFT

Bayesian inference and Markov Chain Monte Carlo sampling to reconstruct a contaminant source at continental scale

Delle Monache Luca, Julie K. Lundquist, Branko Kosovic, Gardar Johannesson, Kathleen M. Dyer, Roger D. Aines, Fotini Katopodes Chow, Rich D. Belles, William G. Hanley, Shawn C. Larsen, Gwen A. Loosmore, John J. Nitao, Gayle A. Sugiyama, Philip J. Vogt

December 8, 2006

Atmospheric Environment

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

Bayesian inference and Markov Chain Monte Carlo sampling to reconstruct a contaminant source at continental scale

Luca Delle Monache[1], Julie K. Lundquist[1], Branko Kosovic[1], Gardar Johannesson[1], Kathleen M. Dyer[1], Roger D. Aines[1], Fotini Katopodes Chow[2], Rich D. Belles[1], William G. Hanley[1], Shawn C. Larsen[1], Gwen A. Loosmore[1], John J. Nitao[1], Gayle A. Sugiyama[1], Philip J. Vogt[1]

[1] *Lawrence Livermore National Laboratory, Livermore, California, USA*

[2] *University of California, Berkeley, California, USA*

Corresponding author address: Luca Delle Monache, Lawrence Livermore National Laboratory, 7000 East Avenue, L-103, Livermore, CA 94550, USA

E-mail address: ldm@llnl.gov

Abstract

A methodology to reconstruct a source from a limited set of measurements and prior knowledge of the release is applied to a real accidental radioactive release at the continental scale, that occurred at the end of May, 1998, near Algeciras, Spain. The methodology combines Bayesian inference with Markov Chain Monte Carlo sampling (MCMC). Annealing and adaptive procedures are implemented to assure a robust and efficient parameter-space exploration. The simulation set-up reflects an emergency response scenario where only the source geometry and release time are assumed to be known, with the latter still highly uncertain at this date. The Bayesian stochastic algorithm is able to provide likely source locations within 100 km from the true source, after exploring a domain covering an area of approximately 1800 by 3600 km. The source strength, whose true value is uncertain as well, is reconstructed with a distribution of values of the same order of magnitude as the upper end of the range reported by the Spanish Nuclear Security Agency. By running the Bayesian MCMC algorithm on a large parallel cluster (with less than a thousand processors) the inversion results could be obtained in few hours as required for emergency response applications at the continental scale.

Keywords: Bayesian inference, Markov Chain Monte Carlo sampling, inverse modeling, source reconstruction, emergency response.

1. Introduction

Knowledge of the temporal and spatial evolution of a contaminant released into the atmosphere accidentally or deliberately is fundamental to adopting efficient strategies to protect the public health, and to mitigate the harmful effects of the dispersed material. In emergency response situations the source parameters may not be known. Typically a source is assumed, and assessment of the trajectory, spread, and ultimate fate of a contaminant plume is based on predictions from atmospheric dispersion models. The accuracy of these predictions is affected by uncertainties in several components of the plume prediction, including the atmospheric dispersion models themselves, the meteorological models used to drive the dispersion models, the atmospheric data assimilated by the meteorological models, and uncertainties in the parameters describing the contaminant source.

Among these sources of uncertainty, those comprising the initial state of the contaminant are often the most significant, and as such provide the central focus of this study. A methodology to solve the “inverse problem” is proposed to reconstruct unknown source parameters given a set of downwind measurements at a time after the release. The inversion algorithm is based on Bayesian inference combined with a Markov Chain Monte Carlo (MCMC) procedure (Gilks et al., 1996; Gelman et al., 2003). This methodology has been used to reconstruct atmospheric releases at local (~1 km) and regional (~10 km) scale using data from the Prairie Grass and Copenhagen tracer experiments (Lundquist et al., 2005). It also has been successfully applied in urban settings using building-resolving computational fluid dynamics simulations (Chow et al., 2006). The algorithm is applied here for the first time to a continental-scale accidental release of radioactive material from near Algeciras, Spain during May of 1998. This event affected for several days much of continental Europe including locations few thousands of kilometers downwind. The methodology provides a skillful, robust statistical characterization of the reconstructed source parameters in the presence of a complex atmospheric flow field using only crude measurements.

Algorithms based on integrating the adjoint dispersion model backward in time (e.g., Pudykiewicz 1998; Keats et al., 2006) have been implemented to solve reconstruction problems. While such approaches can substantially increase the speed and efficiency of the inversion process, there are several limitations. Foremost among these drawbacks, from the perspective of emergency response, is their lack of flexibility. Methods based on the adjoint model require extensive modification and refinements for every change to the framework components (e.g., forward models, datasets or dispersion scenarios) limit their usefulness especially during the initial stages following a release, when different characterizations using different datasets and component models are desirable. Additionally, adjoint-based methods are limited to processes that can be described by linear equations (Enting, 2002). For instance, this constraint precludes their applicability to cases during which chemical reactions are an important component of the dispersion process, or to the reconstruction of unknown meteorological parameters.

The following section provides a detailed description of the reconstruction algorithm. Section 3 discusses the Algeciras release incident, the reconstruction results of which are detailed in Section 4. Section 5 presents the conclusions and follows with a discussion of computational issues related to using our methodology as an emergency response tool.

2. Methodology

The stochastic event reconstruction algorithm is based on Bayes' theorem and a MCMC procedure to sample the unknown parameter space (Gilks et al., 1996; Gelman et al., 2003). The following subsections briefly summarize the theory on which the methodology is constructed and outline the main steps of the procedure.

2.1 Theoretical Framework

Bayes' theorem, as applied to a source reconstruction problem, can be stated as follows:

$$p(\mathbf{S} \mid \mathbf{M}) = \frac{p(\mathbf{M} \mid \mathbf{S})p(\mathbf{S})}{p(\mathbf{M})} \tag{1}$$

Here $p()$ is a probability distribution, $\mathbf{S} = (x, y, q)$ is the state vector formed by the point source parameters (x and y are the source horizontal coordinates and q is the emission rate), and $\mathbf{M}$ is an array formed by the measurements. Bayes' theorem relates the posterior distribution, $p(\mathbf{S} \mid \mathbf{M})$, to the product of the probability of the measurements given the source parameters, $p(\mathbf{M} \mid \mathbf{S})$, also called the likelihood function, and the probability of the source parameters prior to any knowledge of the measurements, $p(\mathbf{S})$, also called the prior. Here x, y, and q are assumed to be the unknown parameters, but in general the Bayes' theorem can be applied with $\mathbf{S}$ including also other parameters, e.g, z (the vertical coordinate), or the release time and duration.

Bayes' theorem is often expressed alternatively as

$$p(\mathbf{S} \mid \mathbf{M}) \propto p(\mathbf{M} \mid \mathbf{S})p(\mathbf{S}) \tag{2}$$

This form preserves the most important information contained in (1), the spatial distribution of the posterior probability, and avoids evaluation of the marginal distribution of **M,** $p(\mathbf{M}) = \int p(\mathbf{S})p(\mathbf{M} \mid \mathbf{S})\, d\mathbf{S}$, for which analytical solutions are rare and computation is expensive.

In this study Bayes' theorem is applied to describe the conditional probability $p(\mathbf{S} \mid \mathbf{M})$ of a source described by x, y and q, given the observed sensor measurements. To estimate the unknown source parameters, i.e., to reconstruct the source using (2), the posterior distribution must be sampled. Sampling the posterior distribution (left-hand side of Equation (2)) involves computing the probability distribution $p(\mathbf{M} \mid \mathbf{S})$ for any proposed **S** realization. $p(\mathbf{M} \mid \mathbf{S})$ quantifies the likelihood of a set of measurements **M** given the source parameters **S**. This likelihood is computed by running a forward dispersion model with the given source parameters **S** and comparing the model predicted concentrations with the sensor measurements (Section 2.2.1). The closer the prediction to the measurements, the higher the likelihood of the source parameters.

2.2 The Algorithm

The forward dispersion simulation is conducted using the Lagrangian Operational Dispersion Integrator (LODI) model (Ermak and Nasstrom, 2000; Nasstrom et al., 2000) developed at the National Atmospheric Release Advisory Center (NARAC) at Lawrence Livermore National Laboratory (LLNL). LODI is driven by meteorological data that can be obtained from a variety of sources including real-time observations, atmospheric forecast models, atmospheric analysis fields, or any combination of the above. The posterior distribution is sampled with a MCMC procedure via a Metropolis-Hasting algorithm (Gilks et al., 1996; Gelman et al., 2003). Here only the main steps of the procedure are discussed. These steps are shown in Figure 1.

Before executing the reconstruction procedure, the source term parameters, x, y and q are assigned prior distributions ($p(\mathbf{S})$) based on the limited information available about the release. Each parameter's prior distribution is bounded by a range specified from prior knowledge of the circumstances relevant to the problem being solved. The width of the prior distribution reflects the confidence in the initial estimate of the source parameters. Following setup, an initial value for each parameter is sampled from its prior and a number of iterations involving source-term sampling and dispersion simulation are executed until convergence to the posterior distribution is reached (See Section 2.3.2 and

4.3.2 for definition and a discussion on convergence criteria). At each iteration the forward run is input with a realization of **S**, i.e., a value for *x*, *y* and *q*. For each parameter, the values corresponding to each iteration forms a "Markov chain", that can be defined as a finite number of values in which the probability of a future value depends only upon the current value (Gilks et al. 1996). Once initiated, each iteration of the solution procedure consists of the following four steps:

1) A new value for *x* is proposed (x_{prop}) according to: $x_{prop} = x + dx$.

   Here *x* is the current value and *dx* is the vector displacement from that value. The displacement is modeled as a random-walk sampled from a Gaussian distribution with zero mean and a standard deviation specified from the current stepsize (discussed below). Hence, the prior distribution $p(\mathbf{S})$ is utilized as a target distribution to estimate a "prior likelihood" of x_{prop}. If the prior likelihood of x_{prop} exceeds that of *x*, the proposed value x_{prop} replaces *x*. If not, a random (Bernoulli) "coin flip" (Section 2.2.1) determines whether the new proposal, even with its lower prior likelihood, will be accepted. This ensures that the sampled parameters reflect the prior likelihood.
2) Step 1) is repeated independently for *y* and *q*.
3) The forward dispersion simulation is conducted using the current values of *x*, *y*, and *q*.
4) The likelihood of the current values of *x*, *y* and *q* is evaluated by comparing the agreement between the predicted data, using the current source parameters, and observed data at the sensor locations. This new likelihood is compared to that resulting from the previous forward simulation. The proposed state likelihood $p(\mathbf{M} \mid \mathbf{S})$ should not be confused with the prior likelihood as defined in step 1) that is based only on the prior distributions, i.e. $p(\mathbf{S})$, without considering the measurements (**M**). If the proposed state likelihood is higher than the likelihood of the previous state, it is accepted. If not, then a random (Bernoulli) "coin flip" (Section 2.2.1) is used to ensure the search explores the entire posterior state space. Occasional acceptance of new proposals with lower likelihoods ensures that the reconstruction procedure continues to search the complete space of

proposed states, preventing the procedure from remaining indefinitely within the neighborhood of a local extrema.

2.2.1 The likelihood function and acceptance condition

The quality of agreement between the predicted and observed data at the sensor locations is expressed in terms of a likelihood function (L). The present study utilizes a likelihood function of the form

$$\ln\left[p(\mathbf{M} \mid \mathbf{S})\right] \equiv \ln\left[L(\sigma, \mathbf{P}, \mathbf{M})\right] = -\frac{\sum_{i}^{N}[\log_{10}(P_i) - \log_{10}(M_i)]^2}{2\sigma^2} + \alpha \tag{3}$$

where L is the likelihood function, P_i are the elements of the array $\mathbf{P}$ of the predicted values at the sensor locations, M_i are the elements of the array $\mathbf{M}$ of the sensor measurements, σ is an error parameter chosen accordingly to expected errors in the observations and predictions, and α is a constant.

After LODI is run with the new proposed state (i.e, a new set of x, y and q), the proposed state is retained if

$$\underset{\text{(a)}}{\ln\left(L_{prop}\right) \geq \ln(L)} \quad or \quad \underset{\text{(b)}}{\ln\left(L_{prop}\right) - \ln(L) \geq \ln\left[rnd(0, 1]\right]} \tag{4}$$

where L_{prop} is the likelihood value of the proposal, L is the previous likelihood value, and $rnd(0, 1]$ is a random number generated from a uniform distribution in the interval (0, 1].

It is important to note that condition (4b) is more likely to be satisfied if the likelihood of the proposal is only slightly lower than the previous likelihood value. This aspect has important implications for improving the Bayesian event reconstruction algorithm efficiency, as explained in the next subsection. If large errors are expected in the prediction and/or measurements, large values of σ should be assumed, resulting in a flatter distribution of likelihood values (as evident from equation (3)). This in turn would increase the acceptance rate induced by a positive test of (4b). Therefore, the larger σ, the broader the posterior distribution. In this simulation $\sigma = 0.8$, that resulted in a rejection rate of approximately 80 %.

2.2.2 Cooling-off and adaptive step procedures

A cooling-off procedure is used (Neal, 2001) to assure at the initial stages of the sampler a broader exploratory sampling of the posterior distribution values. The procedure consists of normalizing the log-likelihood of the proposed values with a "temperature" constant which decreases ("cools-off") linearly with the number of iterations, to unity. Initially, the normalization of the likelihood values increases the number of times condition (4b) is met, since the normalization reduces the relative differences between different likelihood values, allowing the acceptance of less likely proposals. As the "temperature" is progressively decreased, the acceptance rate resulting from condition (4b) decreases accordingly. This procedure allows the Markov chain to initially include a broader subset of the parameter range as accepted values. This study employed a cooling-off period consisting of the first one hundred iterations of each Markov chain.

An adaptive stepsize algorithm has also been implemented to ensure that the magnitude of the displacement between current and proposed parameters is appropriate for the current stage of the search. Following Haario et al. (1999), after an initial transient during which the stepsize is held constant, the stepsize is computed as directly proportional to the variance of the values sampled up to the current iteration. This approach assures in the initial stages of the search a large stepsize, because the large variance of the sampled value, encouraging a broad exploration of parameter space and identifying high likelihood regions more efficiently than a smaller stepsize would. During the latter stages of a search, after the Markov chains have converged to within small neighborhoods of likelihood extrema, the sampled variance is smaller, and the resulting smaller stepsize encourages a chain to explore the contours of likelihood within the neighborhood of its current extrema. It should be noted that those stepsize adjustments affect only the rates of convergence and do not affect the posterior distribution (i.e., the shape of the returned sample).

It should be emphasized that the purpose of the algorithm is not to identify likelihood extrema, but to reveal the probabilistic landscape of likelihood, throughout the entire domain (bounded by the prior probability distributions). The combination of cooling-off and adaptive stepsize results in chains that explore the entire parameter space yet repeatedly converge to the same subset, yielding probabilistic estimates of source parameter values.

2.3 Burn-in and Convergence Definitions

In this section two important aspects of the Bayesian MCMC stochastic algorithm are defined, the burn-in and the convergence criteria.

2.3.1 Burn-in

The burn-in is an important phase represented by an initial subset of the total iterations that can be defined as follows. It can be seen as the number of iterations needed for the current parameter distribution to relax from the initial state (Gilks et al., 1996). Burn-in samples are usually discarded to construct the parameters posterior distributions. Burn-in is further discussed in Section 4.31.

2.3.2 Convergence

Convergence is reached in practice when more samples would not modify the resulting posterior distribution. Statistically, convergence to the posterior distribution can be estimated by computing between-chain variance and within-chain variance (Gelman et al., 2003). If there are m Markov chains of length n for a source parameter S, whose values are denoted by s, then we can compute between-chain variance B as

$$B = \frac{n}{m-1}\sum_{j=1}^{m}\left(\overline{S}_j - \overline{S}\right)^2 \tag{4}$$

where

$$\overline{S}_j = \frac{1}{n}\sum_{i=1}^{n} s_{ij} \tag{5}$$

and

$$\overline{S} = \frac{1}{m}\sum_{j=1}^{m}\overline{S}_j \tag{6}$$

and within-chain variances W as

$$W = \frac{1}{m}\sum_{j=1}^{m} w_j^2 \tag{7}$$

where

$$w_j^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(s_{ij} - \overline{S}_i\right)^2 \tag{8}$$

The convergence value ($\hat{R}$) is given by

$$\hat{R} = \frac{\mathrm{var}(S)}{W} \tag{9}$$

Where the variance of S is defined to be

$$\mathrm{var}(S) = \frac{n-1}{n}W + \frac{1}{n}B \tag{10}$$

The necessary condition for statistical convergence to the posterior distribution is that $\hat{R}$ approaches unity (Gelman et al., 2003). In practice, simulation should be run until $\hat{R}$ is smaller than 1.1 or 1.2 (Gilks et al., 1996). Convergence results are discussed in Section 4.3.2.

3. Algeciras Accidental Release

The Algeciras accident remained undetected for three days after the release and the source was unknown for 10 more days afterward. On May 30, 1998, a piece of medical equipment containing Cesium-137 (Cs-137) was accidentally melted in one of the furnaces at the Acerinox stainless steel production plant near Algeciras, Spain. The plume of Cs-137 fallout marked the most significant nuclear contamination on the European continent since the 1986 Chernobyl power plant disaster.

The release parameters are still not well known. According to Spanish Nuclear Security Agency (CSN) (Vogt et al., 1999, hereinafter referred as V99), the radioactivity of the material released was between 8 and 80 Ci (1 Ci = 3.7 x 10^{10} Bq) and the release occurred between 0100 and 0300 UTC on 30 May. None of the detected radioactivity levels were sufficient to adversely affect the human health or the environment. This event provides a real case to evaluate the source reconstruction methodology described in Section 2, on a continental scale.

A detailed analysis of the synoptic weather patterns at the time of the release can be found in V99. Figure 2 shows qualitatively the radioactive plume transport as reproduced by LODI given realistic source parameters (location: 5° 26' W, 36° 10' N; release time: 0130 UTC, May 30; release duration: 30 min.; release amount: 50 Ci). In each panel the source location is indicated by the white square; white circles represent location of sensors used in this study. Initially (top-left panel) the plume was caught in a westerly flow that advected the radioactive cloud over the Mediterranean Sea, leaving the accident virtually undetected. Between May 31 and June 1 (top-right panel) the plume shifted to the north and was detected by sensors located in southeast France and northwest Italy. After June 3, radioactivity was detected at a number of European stations; concentrations were reported from 24-hour to 1-week averages (V99).

4. Results

This section describes the details of the problem set-up and assumptions made to perform the simulations, followed by analysis of the results. The goal is to demonstrate the ability of the proposed method to reconstruct the source parameters for a contaminant release from available measurements in conditions similar to a real emergency response scenario occurring at the continental scale.

4.1 Simulation Set-up and Assumptions

The first measurements of contamination from the Algeciras release (24-hour average concentrations of radioactive material) were obtained on June 2, three days after the accident. The measurement stations, located in northwest Italy and southeast France (approximately 1600 km northeast of the source location), are shown by the white circles in Figure 3. From June 3 to June 14 a variety of measurements (from 24-hour averages to 1-week averages) were reported in several locations further downwind, mostly in central and eastern Europe. For the present study, only the first set of available 24-hour average measurements (17 values reported from 11 stations June 2-3) are used to reconstruct the unknown source. These data are chosen to evaluate the ability of the proposed methodology to function in an emergency response situation, using only the first available data.

The sensor locations used for the source reconstruction simulation cover a very small portion of the European continent (Figure 3). The compact area covered by the measurements presents a challenge to the reconstruction algorithm, as source parameters that are displaced relatively small distances from the true source may not encounter the sensors at all. Adding to this difficulty is the absence of “zero” concentration readings. Concentrations near the threshold detection levels of the instruments were not reported, rather than being reported as “zero” readings. Zero concentration readings are useful information for the algorithm, information that is lost when the locations fail to report those values. Such data likely would aid the reconstruction algorithm.

The source geometry (surface point source) and duration (between 0130 and 0200 UTC on May 30, 1998, as in V99) were assumed known. CSN reported the released to have happened between 0100 and 0300 UTC on May 30.

The meteorological data used to drive the dispersion model LODI was provided by the National Centers for Environmental Protection Aviation model (AVN) (Kanamitsu et al., 1991). The 6-hour AVN analysis fields at 1-degree horizontal spatial resolution were used as input for the Atmospheric Data Assimilation Parameterization Techniques (ADAPT) model (Sugiyama and Chan, 1998), which generated the 6-hourly spaced meteorological fields at 22-km horizontal resolution input to LODI. Qualitatively this meteorological field was in fair agreement with the observed flow, particularly over the Mediterranean Sea where the plume traveled before hitting the sensors. The dispersion simulation was conducted using 100 000 marker particles – a number sufficient to provide statistical resolution for this case.

The prior distribution of the source location is indicated by the dashed box in Figure 3. This domain (about 1800 km in the east-west direction and 3600 km in the north-south direction) was selected based on the predominantly westerly flow pattern over most of the sensor locations from May 30 (the day of the accident) through June 2 (when the first measurements were available) which implied that the source must have been located to the west of the sensors (western Europe or northwestern Africa). Further, given the wind speeds during the period, locations within 1800 km to the west would likely have passed beyond the sensors by the beginning of June 2. The meridional extent of the prior reflects similar confidence in potential values of the source latitude. Initially, each location within this box was given approximately the same probability to be sampled, with the *x* and *y* values generated from two Gaussian distributions with mean values taken to be the box center coordinates and standard deviations equal to the box dimension in the east-west direction.

4.2 Source Reconstruction

The stochastic engine was run with three independent Markov chains. For a given number of sampled values, a higher number of chains would produce the same number of samples more rapidly, if a large cluster is available to run the independent chains all at the same time. Additional comments on the computational costs of the procedure can be found at the end of Section 5.

Figure 3 depicts the locations explored by the three Markov chains. Given the assumption of a surface point source, the chains were allowed to explore only values at the surface. The gray dots represent accepted states during the burn-in phase (first 500 iterations, as defined in Section 2.3.1 and discussed in Section 4.3.1), whereas the black dots represent those states accepted afterwards. The chains efficiently explore the given sampling box, assuring the construction of a posterior distribution for the sampled parameters likely to include all of the dominant modes.

Figure 4 shows an expanded view of the region including the locations of the states accepted after burn-in. Different symbols (circle, five-point star, and triangle) represent the three chains. The posterior distribution reveals two distinct modes of high relative probability indicating likely source locations; one over land within 60 km north of the real source location (white square), and one over the Mediterranean Sea about 80 km west and 20 km south of Algeciras. The Markov chains sample both within and between the two modes providing evidence of chain convergence (as defined in Section 2.3.2 and discussed in Section 4.3.2).

The probabilistic aspect of the answer provides a useful tool for a real emergency-response scenario. The algorithm finds among all the possible solutions the few ones that are more consistent with the data available and its uncertainties. The information in Figure 4 provides guidance for decision makers formulating an appropriate strategy. It would have suggested looking for potential source locations (e.g., steel mill) just north of Algeciras, and this would have led to rapid identification of the real source. Moreover, the mode downwind of Algeciras would have prompted first searching for possible ships or any other floating or submerged body releasing radioactive material in that localized stretch of waters. The posterior distribution in Figure 4 could also be used to construct an updated prior for a more detailed search. Repeated application of such an approach could yield successively more precise source location information.

Figure 5 shows contours of location probabilities based on the accepted states spatial distribution showed in Figure 4. Also shown in Figure 5 are the surface wind fields driving LODI at the time of the release. The probability distributions clearly show the location of the two modes of high relative probability and likewise indicate how the distributions are influenced by changes in the wind field. For instance the distributions are stretched in approximately the along-wind direction, and the amount of the stretching is roughly proportional to the wind speed.

One explanation for this feature is the following. Measurements that are averaged over a given time interval are relatively insensitive to the plume's exact arrival and transit time, provided that the majority of the plume passes over the sensors within the given averaging interval. Since the flow pattern formed an arc between Algeciras and the sensors that was nearly stationary, sources spread over as much as 100 km along the streamline represented by the arc would have produced very similar 24-hour averaged predictions at the sensor locations. In fact, the along-wind length of the distribution shown in Figures 4-5 is roughly the distance a parcel would traversed given the model wind speed in 24 hours. Similar features in the posterior distributions for source location have been observed in studies utilizing similar inversion algorithms (although applied at much smaller spatial scales) where similar explanations hold (Lundquist et al., 2005; Chow et al., 2006).

More insights about the methodology performance can be inferred from Figures 6-8. Figure 6 shows the surface location (*x* and *y*) and the release rate (*q*) values versus the iteration number. The range adopted in each panel for the vertical axes reflect the actual ranges spanned by the sampler. In the top and central panel the horizontal grey line represents the true value of the source location coordinates, whereas the two horizontal grey lines in the bottom panel show the range of the true source emission, as reported by CSN. In each panel, there are three black lines representing the values the Markov chains assume during the iterations.

The first 100 iterations show the effect of the cooling-off procedure. Initially the chains span a wide portion of the given range for each parameter, but the acceptance rate decreases as the temperature linearly decreases with the iteration number. After 100 iterations, the chain values span only a limited subset of the possible range constrained by the likelihood of the proposed values as explained in Section 2.2.2.

The effects of the adaptive step procedure are less apparent than the cooling-off, but they can be noticed in the top and bottom panel of Figure 6. After the cooling-off period (i.e., after iteration 100), the values of *x* or *q* have a tendency to slowly change. This causes the variance of the iteration series to rapidly diminish, that in turn results in a decrease of the stepsize for subsequent sampling (Section 2.2.2). With a smaller stepsize, the Bayesian event reconstruction algorithm proposes states with higher likelihood that are more frequently accepted (approximately between iterations 200-250) than for the previous iterations (between 100-200). Without the implementation of the adaptive step procedure, the proposed states' rejection rate would have been inefficiently high, requiring a much higher number of iterations to collect the same number of samples than when the adaptive step procedure is adopted.

As the Markov chains in Figure 6 converge, the sampled parameter assumes the same values throughout consecutive iterations (horizontal black lines) with new proposed states or the individual proposed values (for *x*, *y*, and *q*) repeatedly rejected. All of the values for *x* (Figure 6 top panel) tend to be greater than the x-coordinate of the real source, towards downwind locations. After about 500 iterations the chains start to explore a small subset of the initial range, corresponding to the values with the highest likelihood for this parameter. Using accepted values of *x* a histogram can be constructed as shown in the top panel of Figure 7 (vertical grey line represents the true source x-coordinate value). The majority of the values are within 100 km from the real value, with the dominant mode about 80 km downwind of it.

The central panel of Figure 7 shows the quality of the prediction of the *y* values posteriori distribution (vertical grey line represents the true source y-coordinate). Two modes can be recognized, just few tens of km north and south of the true source location, being the latter the dominant one. The *y* values converge faster to a much smaller subset of the initial range (Figure 6, central panel). A higher rejection rate seems to apply to *y* as shown by more frequent intervals where *y* candidates are repeated, as compared to *x*. This behavior is due to the clustering and close spacing of the sensor locations – even a small (few tens of km) deviation from the real *y* value towards the south or towards the north leads to a plume that misses the sensor locations, which in turn results in a low likelihood value for the proposed state. Finally, this behavior indicates the quality of the meteorological field used to drive LODI. A less accurate meteorological field would lead to a plume hitting the sensor locations even if released from a source not close to Algeciras.

The bottom panel of Figure 6 shows the *q* values of the accepted states. The ordinate is represented with a logarithmic scale spanning the wide range of values sampled for *q* (from 10^{13} to 10^{17} μBq s^{-1}). There is a large uncertainty on what the real values of the radioactive material released was, with values ranging from 8 to 80 Ci (1 Ci = 3.7 x 10^{10} Bq) as shown by the two grey lines in the panel. From Figure 6 it can be noted that *q* is the slowest parameter to reach a phase where the accepted states span only a subset of the provided range. The *q* values also have the highest variability reflecting higher uncertainty than the determination of the location.

There is a tendency to over-predict the 8-80 Ci range. The majority of the sampled values after the burn-in (defined in Section 2.3.1 and discussed in Section 4.3.1) are of the same order of magnitude as the upper limit of 80 Ci. There are several possible explanations. The release duration assumed in these simulations (0130-0200 UTC on May 30, as in V99) is within the values suggested by CSN, but if this period is too short the algorithm would calculate higher emission rates in order to compensate for the total radioactivity released. Moreover, since the observed concentrations are 24-hour averages, it is extremely challenging to obtain a tight posterior distribution and a limited number of modes for q, since these observed values could result from a variety of different choices of x, y, and q. These uncertainties also affect the rapidity with which convergence is reached, q being the slowest converging parameter as discussed in the next section.

4.3 Burn-in and Convergence

In this section two important aspects of the stochastic procedures are briefly discussed, the burn-in phase and the convergence criteria.

4.3.1 Burn-in

Burn-in (as defined in Section 2.3.1) has been applied to the plots shown in Figure 4 and 7. The burn-in phase was chosen as the first 500 iterations from a visual inspection of the parameter sampled values variation with the number of iterations (Figure 6) and the convergence criteria ($\hat{R} < 1.5$ in Figure 8 for all parameters). Tests with burn-in ranging from the first 300 to the first 1500 iterations lead to posteriori distributions close to the one depicted in Figure 7 (not shown).

4.3.2 Convergence

Figure 8 shows the convergence values $\hat{R}$ (Section 2.3.2, Equation 9) for the source parameter x, y and q versus the number of iterations. After 2000 iterations the necessary statistical convergence condition of $\hat{R} < 1.2$ is met for all the parameters, with y being the fastest parameter reaching this condition, and q the slowest.

Note that after the first 100 iterations (i.e., during the cooling-off procedure), all three parameters appear to meet the convergence condition. Nevertheless, by visual examination of the parameters values versus iteration number (Figures 3 and 6) it is clear that convergence has not been reached after 100 iterations. Indeed, also an expert judgment is needed to assess the convergence of the simulation (Gilks et al., 1996). Practically, convergence is reached when is clear that more samples would not modify the resulting posterior distribution.

5. Conclusions

A methodology to reconstruct a source given a set of measurements has been presented. It combines Bayesian inference with Markov Chain Monte Carlo (MCMC) sampling, and produces posterior probability distributions of the parameters describing the unknown source. The methodology has been applied for the first time to a real accidental radioactive release at the continental scale occurred in May 1998, near Algeciras, Spain.

The parameters sampled are the source location and strength. The source duration has been assumed to be 0130-0200 UTC, May 28, which falls within the time interval 0100-0300 reported by the Spanish Nuclear Security Agency (CSN) (Vogt et al., 1999). The release was also assumed to be a surface point source.

The source location is reconstructed as a roughly bimodal distribution, with modes located a few tens of km north of Algeciras and about 80 km downwind of the real source location. The source strength is represented by a wider posterior distribution (reflecting the higher uncertainty of this parameter with respect to the source location) with a tendency of the Bayesian MCMC algorithm to over-predict the reported source strength. The majority of the sampled values of this parameter have the same order of magnitude of the estimated release. The over-prediction can be caused by the assumption of a shorter duration than in the real release.

The probabilistic aspect of the solution optimally combines a likely answer with the uncertainties of the available data. From several possible solutions, the Bayesian event reconstruction algorithm is able to find only the few ones that are more consistent with the data available and its uncertainties. The source reconstruction performed in this study would have provided a decision maker with accurate information about the accident, soon after the first measurements were available. This would have led to timely and efficient actions to preserve the public health, in case of harmful radioactive material concentrations were released. Moreover, the results presented show that the methodology have skills even with a limited number of observations available with a coarse time resolution (24-hour averages).

To demonstrate the efficiency of the methodology presented the stochastic engine has been run with three Markov chains. Each Markov chain can be run independently at the same time. Moreover, each chain can be run in parallel on multiple processors. By using 30 chains the same results can be obtained in less than six hours machine time. This could be accomplished by independently running the chains over a cluster with less than a thousand processors, each having a 2.4 GHz CPU speed. Furthermore, no effort has been done here to improve the efficiency and to reduce the computational load per Markov Chain.

The uncertainties in the meteorology directly affect the quality of the source reconstruction. Both the mean and turbulent components of the wind strongly affect the performance of the dispersion model and the quality of the predicted concentrations. These uncertainties could be taken into account by combining Bayesian inference and MCMC sampling with ensemble techniques, an approach widely used in the weather forecast community (e.g., Palmer and Hagedorn, 2006), that will be considered for future applications. Future work will also focus on algorithm optimizations to improve its efficiency and to reduce the overall computational cost. Finally, future tests will also include other parameters in the sampling process, e.g., the source release time and duration, to replicate closely real-time emergency response scenarios.

Acknowledgements

We thank Jeffrey Mirocha (LLNL) for carefully reviewing the manuscript and Steve Homann (LLNL) for discussing technical implications of accidental releases of radioactive material. We are thankful to Art Mirin (LLNL) for providing software engineering improvements to the stochastic code. We also thank Ernest Arnold, Robert Shectman, and Hoyt Walker (LLNL) for helping with the simulations set up.

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

References

Chow, T., Kosovic, B., Chan, S., 2006. Source inversion for contaminant plume dispersion in urban environments using building resolving simulations. 86th American Meteorological Society Annual Meeting, Atlanta, Georgia, USA <http://ams.confex.com/ams/Annual2006/techprogram/paper_100455.htm>.

Enting, I., 2002. Inverse Problems in Atmospheric Constituent Transport. Cambridge Univ. Press, New York, 392 pp.

Ermak, D., Nasstrom, J., 2000. A Lagrangian stochastic diffusion method for inhomogeneous turbulence. Atmospheric Environment 34, 7, 1059-1068.

Gelman, A., Carlin, J., Stern, H., Rubin, D., 2003. Bayesian Data Analysis. Chapman & Hall/CRC, London, 668 pp.

Gilks, W., Richardson, S., Spiegelhalter, D., 1996. Markov Chain Monte Carlo in Practice. Chapman & Hall/CRC, London, 486 pp.

Haario, H., Saksman, E., Tamminen, J., 1999. Adaptive proposal distribution for random walk Metropolis algorithm. Computational Statistics 14, 375-395.

Kanamitsu, M., Alpert, J., Campana, K., Caplan, P., Deaven, D., Iredell, M., Katz, B., Pan, H.-L., Sela, J., White, G., 1991. Recent changes implemented into the global forecast system at NMC. Weather and Forecasting 6, 425–435.

Keats, A., Yee, E., Lien, F.-S., 2006. Bayesian inference for source determination with applications to a complex urban environment. Accepted to appear on Atmospheric Environment.

Lundquist, J., Kosovic, B., Belles, R., 2005. Synthetic event reconstruction experiments for defining sensors network characteristics. Technical Report UCRL-TR-217762, Lawrence Livermore National Laboratory, Livermore, California, USA <http://www.llnl.gov/tid/lof/documents/pdf/328798.pdf>.

Nasstrom, J. S., Sugiyama, G., Leone, J. M. Jr., Ermak, D. L., 2000: A real-time atmospheric dispersion modeling system, Preprint, Eleventh Joint Conference on the Applications of Air Pollution Meteorology, Long Beach, California, USA, Jan. 9-14, 2000. American Meteorological Society, Boston, MA, 84-89.

Neal, R., 2001. Annealed importance sampling. Statistics and Computing 11, 125–139.

Palmer, T., Hagedorn, R., 2006. Predictability of Weather and Climate. Cambridge University Press, New York, 718 pp.

Pudykiewicz, J. A., 1998. Application of adjoint tracer transport equations for evaluating source parameters. Atmospheric Environment 32, 3039–3050.

Sugiyama, G., Chan, S., 1998. A new meteorological data assimilation model for real-time emergency response. 10th Joint American Meteorological Society Conference on the Applications of Air Pollution Meteorology, Phoenix, Arizona, USA <http://www.llnl.gov/tid/lof/documents/pdf/232515.pdf>.

Vogt, P., Pobanz, B., Aluzzi, F., Baskett, R., Sullivan, T., 1999. ARAC simulation of the Algeciras, Spain steel mill CS-137 release. Technical Report UCRL-JC-131330, Lawrence Livermore National Laboratory, Livermore, California, USA <http:// www.llnl.gov/tid/lof/documents/pdf/235247.pdf>.

Figure Captions

Figure 1. Flow diagram of the algorithm.

Figure 2. Qualitative illustration of the plume fate the four days following the release (May 30 – June 3, 1998), as simulated with the Lagrangian Operational Dispersion Integrator (LODI) model. The white square represents the real source location (5° 26' W, 36° 10' N), whereas the filled white circles are the sensors locations. Contoured concentrations are 24-hour averages.

Figure 3. Proposed state locations accepted. The dashed box is the starting sampling area, provided as input (i.e., the location prior distribution), covering western Europe and northwestern Africa. The grey dots are the pre burn-in states (iterations 1-500), while the black dots represent the post burn-in states (iterations > 500). White square and circles as in Figure 2.

Figure 4. Figure 3 zoom-in nearby the real-source location. Each point represents an accepted state post burn-in (iterations > 500), where different symbols (circle, five-point star, and triangle) correspond to different chains.

Figure 5. Location probability spatial distribution build with post burn-in (iterations > 500) accepted states.

Figure 6. The three chains values versus the number of iterations, for *x* (km), *y* (km) and *q* (μBq s^{-1}) (black lines). Values of *q* are reported on a logarithmic scale. Horizontal grey lines represent the true values of *x* and *y* in the top and central panel, respectively. The two horizontal grey lines in the bottom panel are the estimated likely range of the source strength (accordingly to the Spanish Nuclear Security Agency, Vogt et al. (1999)).

Figure 7. Posterior distribution as inferred by the Bayesian event reconstruction algorithm for *x* (km), *y* (km) and log(*q*) (log(μBq s^{-1})). Vertical grey lines represent the true values for *x* and *y* (top two panels) and the reported (accordingly to the Spanish Nuclear Security Agency, Vogt et al. (1999)) range for *q* (bottom panel).

Figure 8. Convergence values versus the number of iterations for *x* (dashed line), *y* (dot-dashed line), and *q* (solid line).

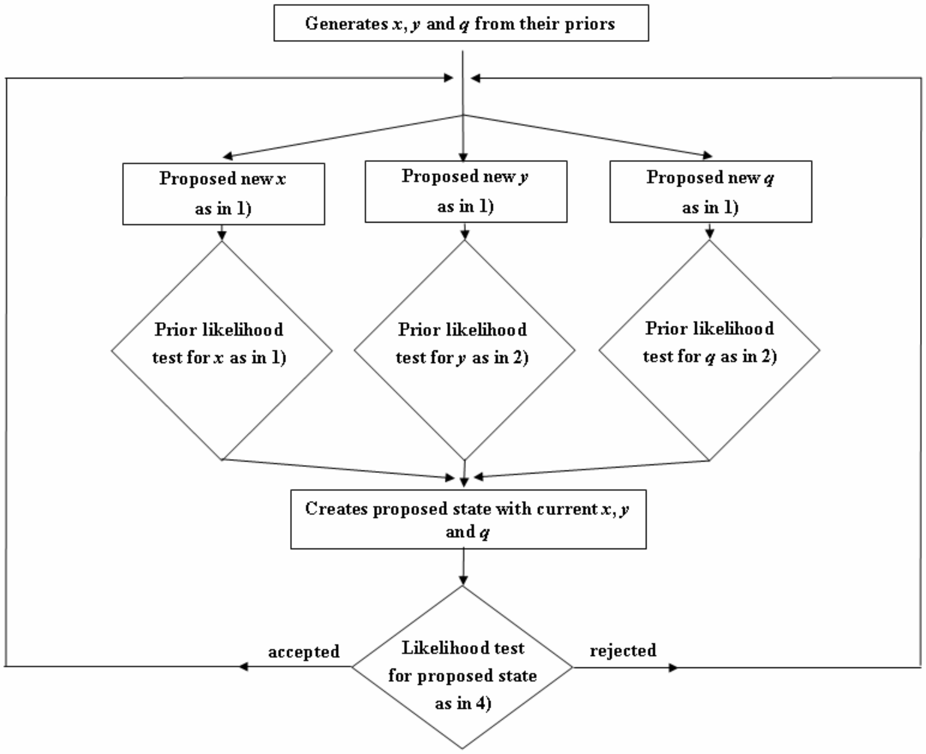

Generates x, y and q from their priors
Proposed new x as in 1)
Proposed new y as in 1)
Proposed new q as in 1)
Prior likelihood test for x as in 1)
Prior likelihood test for y as in 2)
Prior likelihood test for q as in 2)
Creates proposed state with current x, y and q
accepted
Likelihood test for proposed state as in 4)
rejected

Figure 1.

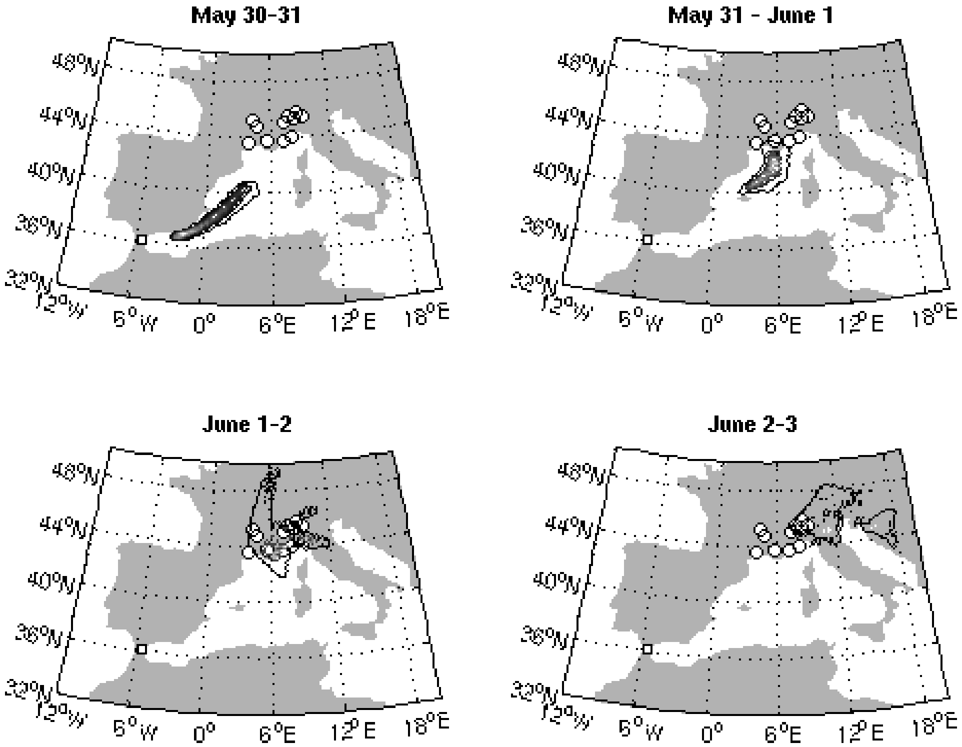

May 30-31
May 31 - June 1
June 1-2
June 2-3
48°N
44°N
40°N
36°N
32°N
12°W
6°W
0°
6°E
12°E
18°E

Figure 2.

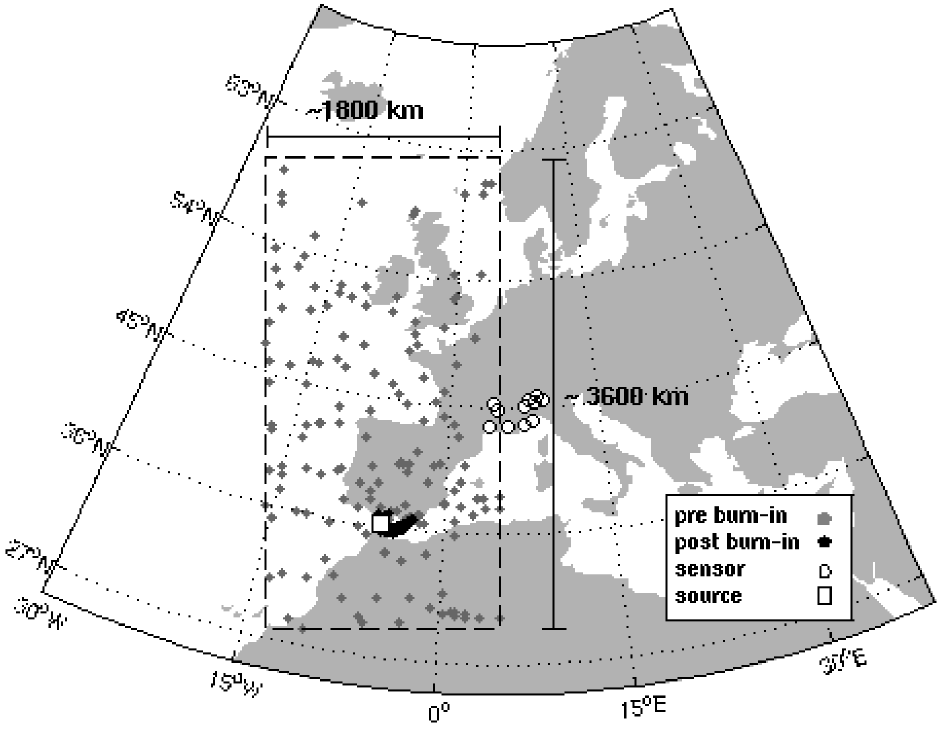

63°N
~1800 km
54°N
45°N
~ 3600 km
36°N
27°N
30°W
15°W
0°
15°E
30°E
pre burn-in
post burn-in
sensor
source

Figure 3.

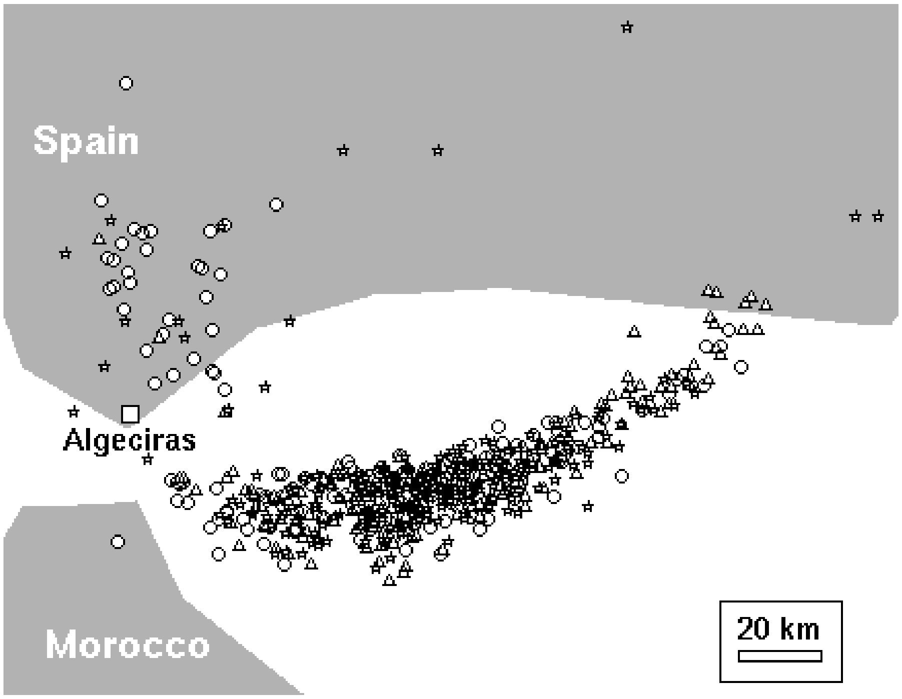
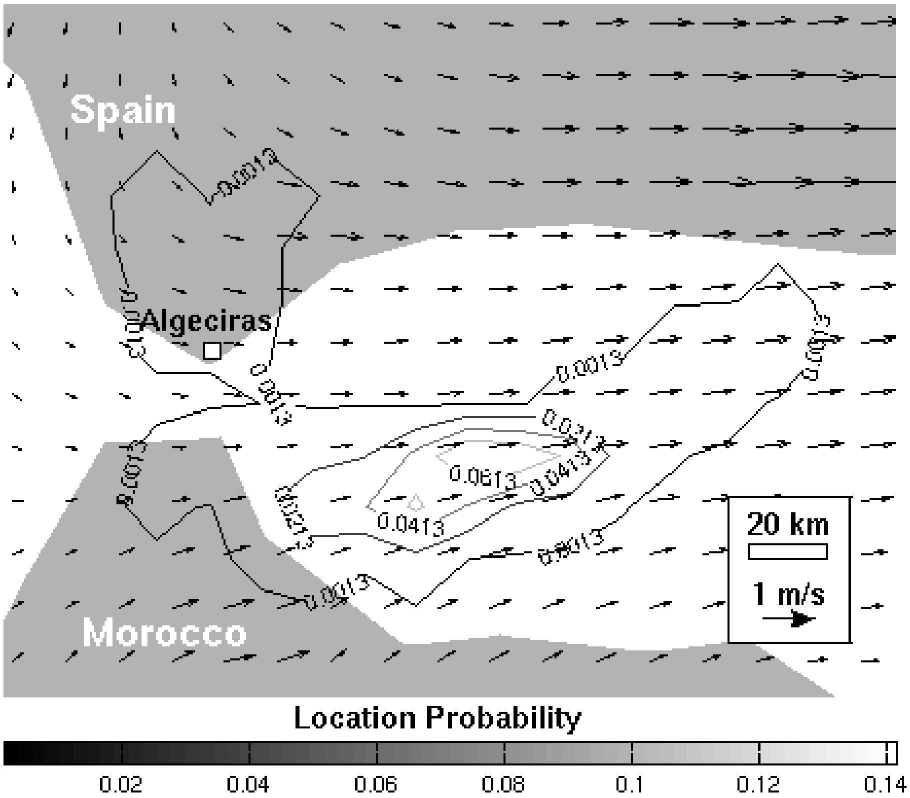


Figure 4.


Spain
Algeciras
Morocco
0.0013
0.0213
0.0413
0.0613
20 km
1 m/s
Location Probability
0.02
0.04
0.06
0.08
0.1
0.12
0.14

700 **Figure 5.**

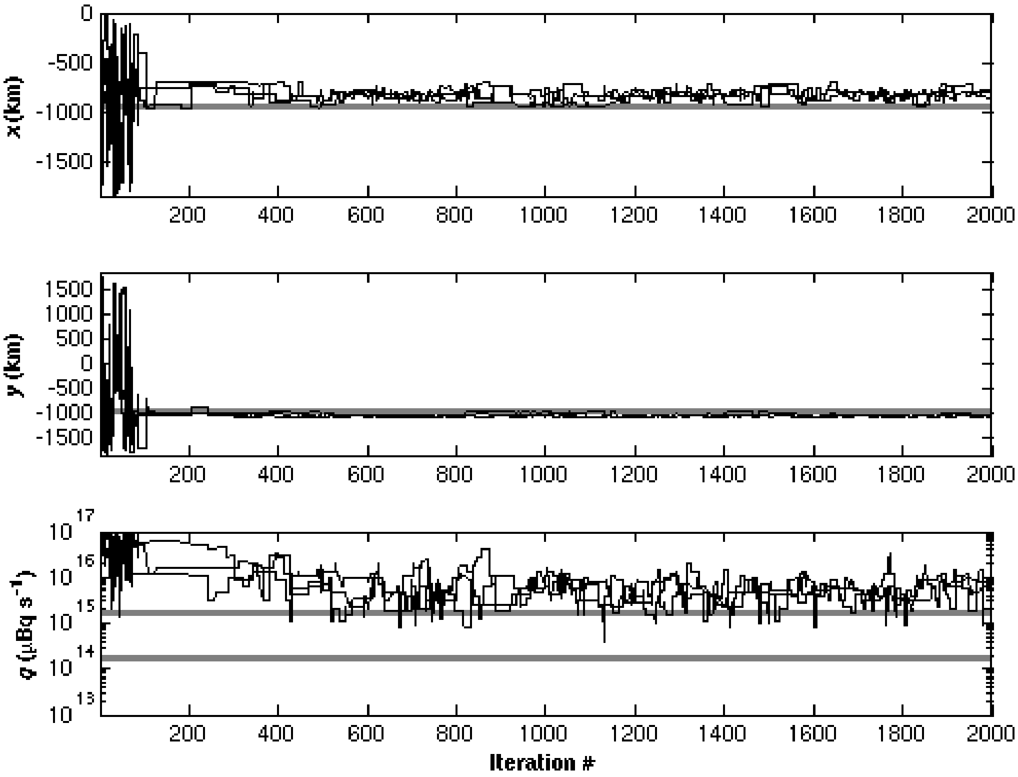

0
-500
-1000
-1500
x (km)
200
400
600
800
1000
1200
1400
1600
1800
2000
1500
1000
500
0
-500
-1000
-1500
y (km)
200
400
600
800
1000
1200
1400
1600
1800
2000
10^{17}
10^{16}
10^{15}
10^{14}
10^{13}
q (μBq s^{-1})
200
400
600
800
1000
1200
1400
1600
1800
2000
Iteration #

Figure 6.

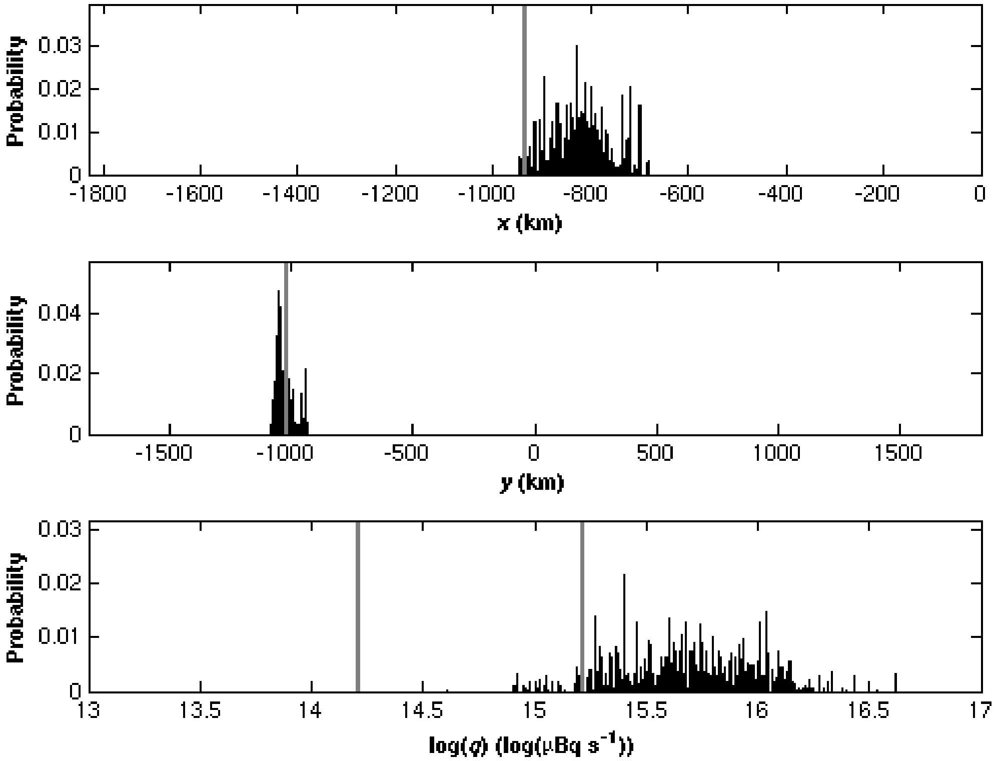

Probability
0.03
0.02
0.01
0
-1800
-1600
-1400
-1200
-1000
-800
-600
-400
-200
0
x (km)
Probability
0.04
0.02
0
-1500
-1000
-500
0
500
1000
1500
y (km)
Probability
0.03
0.02
0.01
0
13
13.5
14
14.5
15
15.5
16
16.5
17
log(q) (log(μBq s^{-1}))

Figure 7.

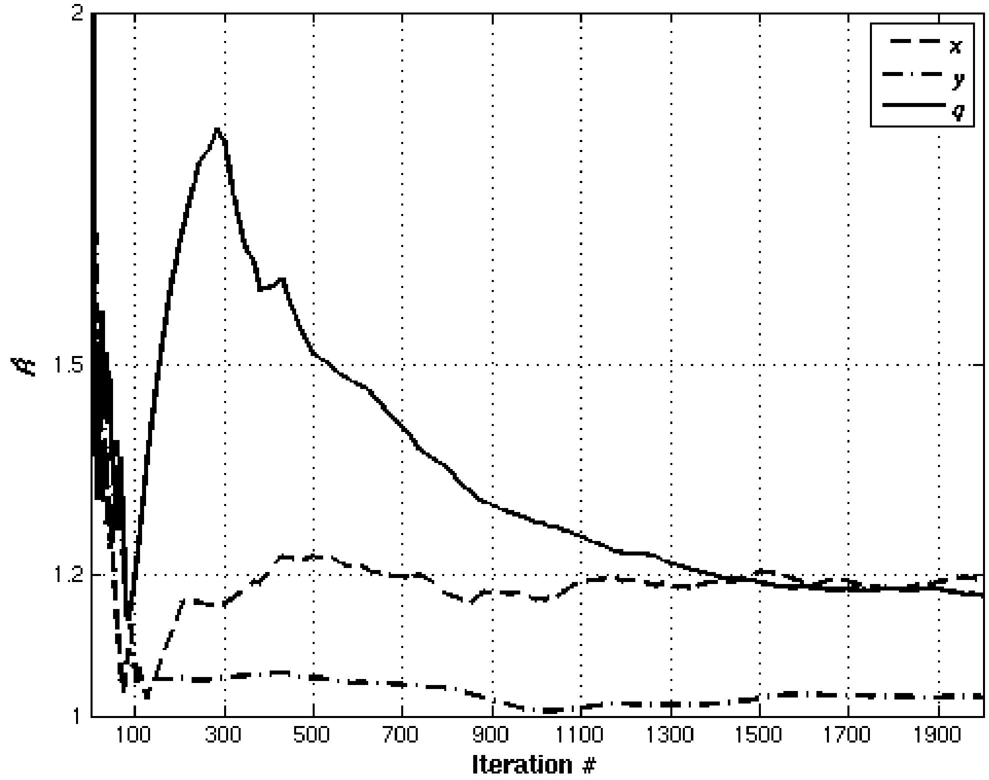

2
1.5
1.2
1
β
100
300
500
700
900
1100
1300
1500
1700
1900
Iteration #
x
y
q

Figure 8.

Appendix B

Source inversion for contaminant plume dispersion in urban environments using building-resolving simulations

UCRL-JRNL-228011

UCRL-JRNL-228011

Source inversion for contaminant plume dispersion in urban environement using building-resolving simulations

Fotini Katopodes Chow, Branko Kosovic, Stevens Chan

February 13, 2007

Atmospheric Environment

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

Source inversion for contaminant plume dispersion in urban environments using building-resolving simulations

Fotini Katopodes Chow[a,*] Branko Kosović[b] Stevens T. Chan[b]

[a]*Civil and Environmental Engineering, University of California, Berkeley, CA, 94720-1710, USA*

[b]*Atmospheric, Earth and Energy Department, Lawrence Livermore National Laboratory, Livermore, CA, 94550, USA*

Abstract

The ability to determine the source of a contaminant plume in urban environments is crucial for emergency response applications. Locating the source and determining its strength based on downwind concentration measurements, however, is complicated by the presence of buildings which can divert flow in unexpected directions. High-resolution flow simulations are now possible for predicting plume evolution in complex urban geometries, where contaminant dispersion is affected by the flow around individual buildings. Using Bayesian inference via stochastic sampling algorithms with a high-resolution CFD model, we can reconstruct an atmospheric release event to determine the plume source and release rate based on point measurements of concentration.

Event reconstruction algorithms are applied first for flow around a prototype isolated building (a cube), and then using observations and flow conditions from Oklahoma City during the Joint URBAN 2003 field campaign. Stochastic sampling

Preprint submitted to Elsevier *26 January 2007*

methods (Markov Chain Monte Carlo) are used to extract likely source parameters, taking into consideration measurement and forward model errors. In all cases the steady-state flow field generated by a 3D Navier-Stokes finite-element code (FEM3MP) is used to drive thousands of forward dispersion simulations. To enhance computational performance in the inversion procedure, a reusable database of dispersion simulation results is created. We are able to successfully invert the dispersion problems to determine the source location and release rate to within narrow confidence intervals even with such complex geometries.

Our stochastic methodology is general and can be used for time-varying release rates and reactive flow conditions. The results of inversion indicate the probability of a source being found at a particular location with a particular release rate, thus inherently reflecting uncertainty in observed data or the lack of enough data in the shape and size of the probability distribution. A composite plume showing concentrations at the desired confidence level can also be constructed using the realizations from the reconstructed probability distribution. This can be used by emergency responders as a tool to determine the likelihood of concentration at a particular location being above or below a threshold value.

Key words: Bayesian inference, computational fluid dynamics, Markov-Chain Monte Carlo, inverse problem, urban dispersion

* Corresponding author address: Civil and Environmental Engineering, MC 1710, University of California, Berkeley, CA, 94720-1710, Tel: 1-510-643-4405, Fax: 1-510-642-7483

Email addresses: `chow@ce.berkeley.edu` (Fotini Katopodes Chow), `kosovic1@llnl.gov` (Branko Kosović), `chan3@llnl.gov` (Stevens T. Chan).

URL: `http://www.ce.berkeley.edu/c̃how` (Fotini Katopodes Chow).

1 Introduction and background

Flow in urban environments is complicated by the presence of buildings, which divert the flow into often unexpected directions. Contaminants released at ground level are easily lofted above tall (~ 100 m) buildings and channeled through urban canyons that are perpendicular to the wind direction (see e.g., IOP 9 in Chan and Leach, 2007). The path of wind and scalars in urban environments is difficult to predict even with building-resolving computational fluid dynamics codes, due to the uncertainty in the synoptic wind and boundary conditions and errors in parameterizations of different physical processes such as turbulence.

Given the difficulties due to the complexity of urban flows, solving an inverse problem is quite challenging. That is, given measurements of concentration at sensors scattered throughout a city, is it possible to detect the source and strength of a contaminant release, and if so, can the uncertainty in source characteristics be estimated? The ability to determine source location and strength in a complex environment is necessary for emergency response for accidental or intentional releases of contaminants in densely-populated urban areas. The goal of this work is to demonstrate a robust statistical inversion procedure that performs well even under the complex flow conditions and uncertainty present in urban environments.

Much work has previously focused on direct inversion procedures, where an inverse solution is obtained using an adjoint advection-diffusion equation. The exact direct inversion approaches are strictly limited to processes governed by linear equations and also generally assume the system is steady-state (Enting, 2002; Keats et al., 2007). In addition to adjoint models, optimization techniques are also employed to obtain solutions to inverse problems. These techniques often give only a single best answer, or assume a Gaussian distribution to account for uncertainties. General dispersion related inverse problems, however, often include non-linear processes (e.g., dispersion of chemically reacting substances) or are characterized by non-Gaussian probability distributions (Bennett, 2002). Traditional methods also have particular weaknesses for sparse, poorly-constrained data problems, as well as in the case of high-volume, potentially over-constrained and diverse data streams.

We have developed a more general and powerful inverse methodology based on Bayesian inference coupled with stochastic sampling (Chow et al., 2006). Bayesian methods reformulate the inverse problem into a solution based on efficient sampling of an ensemble of predictive simulations, guided by statistical comparisons with observed data (see e.g. Ramirez et al., 2005). Predicted values from simulations are used to estimate the likelihoods of available measurements; these likelihoods in turn are used to improve the estimates of the unknown input parameters. Bayesian methods impose no restrictions on the types of models or data that can be used. Thus, highly non-linear systems and disparate types of concentration, meteorological and other data can be simultaneously incorporated into an analysis.

In this work we have implemented stochastic models based on Markov Chain Monte Carlo sampling for use with a high-resolution building-resolving computational fluid dynamics code, FEM3MP. The inversion procedure is first applied to flow around an isolated building (a cube) and then to flow in Oklahoma City (OKC) using data collected from SF_6 tracer gas releases during the Joint URBAN 2003 field experiment (Allwine, 2004). While we consider steady-state flows in this first demonstration, the approach used is entirely general and is also capable of dealing with unsteady, nonlinear governing equations. Our stochastic approach has been applied to other dispersion cases (Delle Monache et al., 2007), but never before to urban environments as done here.

2 Reconstruction procedure

2.1 Bayesian inference and Markov Chain Monte Carlo

The inversion or reconstruction algorithm uses Bayes' theorem combined with a Markov Chain Monte Carlo (MCMC) approach for stochastic sampling of unknown parameters (see e.g., Gelman et al., 2003). A brief description is given here; more details can be found in Johannesson et al. (2004, 2005). Bayes theorem is written

$$p(M|D) = \frac{p(D|M)p(M)}{p(D)} \propto p(D|M)p(M) \tag{1}$$

where M represents possible model configurations or parameters and D is observed data. For our application, Bayes theorem therefore describes the conditional probability ($p(M|D)$) of certain source parameters (the model configuration, M, including e.g. source location and release rate) given observed measurements of concentration at sensor locations (D). This conditional probability $p(M|D)$ is also known as the posterior distribution and is related to $p(D|M)$, the probability of the data conforming to a given model configuration, and $p(M)$, the possible model configurations before taking into account the measurements. $p(D|M)$, for fixed D, is called the likelihood function, while $p(M)$ is the prior distribution. In this application, we assume at the outset that the source could be located anywhere in the whole domain, so the prior distribution is uniform over the chosen domain. The probability $p(D)$ distribution is called the prior predictive distribution (Gelman et al., 2003) and represents a marginal distribution of D. $p(D)$ is a normalizing factor and is not needed when computing the posterior distribution. For a general problem where analytical solutions are not possible, the challenge lies in computing the likelihood function. For that purpose we use a stochastic sampling procedure and approximate the posterior distribution ($p(M|D)$) by the empirical distribution function described below.

2.2 Sampling procedure

We use a Markov Chain Monte Carlo (MCMC) procedure with the Metropolis-Hastings algorithm to obtain the posterior distribution of the source term parameters given the concentration measurements at sensor locations (Gelman et al., 2003; Gilks et al., 1996). We thus completely replace the Bayesian formulation with a stochastic sampling procedure to explore the model parameter space and obtain a probability distribution for the source location and strength. The Markov chains are initialized by taking samples from the prior distribution. To lower the computational cost, we limit the prior distribution to the ground surface (thus ignoring the possibility of elevated sources). All grid cells associated with the footprints of buildings are also excluded from the prior distribution for the Oklahoma City runs.

A forward dispersion calculation is first performed to provide the initial data for comparison with observed data at sensors at the initial locations of the Markov chains. Each Markov chain path is determined using the Metropolis-Hastings algorithm at each step (Delle Monache et al., 2007, see their Fig. 1). A sample is taken from a specified Gaussian proposal distribution centered at the current chain location and likewise from a Gaussian proposal distribution for the source strength. A forward calculation is performed for the proposal with these new parameters and results are compared to measurements at the concentration sensors. If the comparison is more favorable than the previous chain location, the proposal is accepted, and the Markov chain advances to the new location. If the comparison is worse, the proposal is not automatically rejected. Instead, a Bernoulli random variable (a "coin flip") is used to decide whether or not to accept the new state. This random component is important because it prevents the chain from becoming trapped in a local minimum where comparisons are more favorable than values in the local sampling area but where the chain has not converged on the true source location or release rate.

A log likelihood function is used to quantify the agreement between the model configuration and the data; it is defined as

$$\ln(\mathcal{L}(M)) = -\frac{\sum\limits_{i}^{N}(C_i^M - C_i^E)^2}{2\sigma_{rel}^2} \tag{2}$$

where C_i^M are model values at the sensor locations, C_i^E are the experimentally observed sensor values, and σ_{rel} is the standard deviation of the combined forward model and measurement errors. The squared difference is summed over the N sensor locations. In this work, the logarithm of the model and data values is taken before using this formula. This prevents large concentration values from dominating the likelihood calculation when the range of concentrations spreads over several orders of magnitude. The likelihood function is calculated as the forward model for each proposed new state (sample x,y, and q values) is computed. As described above, the proposed state is accepted if either

$$\ln(\mathcal{L}_{prop}) \geq \ln(\mathcal{L}) \quad \text{or} \quad \mathcal{L}_{prop}/\mathcal{L} \geq rnd(0, 1] \tag{3}$$

where $\ln(\mathcal{L}_{prop})$ is the log likelihood of the proposed state, $\ln(\mathcal{L})$ is the previous likelihood value, and rnd denotes a random number generated from a uniform distribution to represent the “coin flip”.

The posterior probability distribution in Eq. 1 is computed discretely from the resulting Markov chain paths defined by the algorithm described above and is estimated with

$$p(M|D) = \sum_{i=1}^{n}(1/n)\delta(M_i - M) \tag{4}$$

which represents the probability of a particular model configuration (M, including parameters such as source location and strength) giving results that match the observations at sensor locations (D). Equation (4) is a sum over the entire Markov chain of length n of all the sampled values M_i which fall within a certain “bin”. Thus $\delta(M_i - M) = 1$ when $M_i = M$

and 0 otherwise. If a Markov chain spends several iterations at the same location, meaning that multiple proposals were rejected because the given location was more favorable than the proposals, the value of $p(M|D)$ increases through the summation in Eq. (4), indicating a higher probability for those source parameters.

Multiple chains are used (typically four) to allow for better statistical sampling of the parameter space and to enable convergence monitoring (thus Eq. (4) is overly simplified). Statistical convergence to the posterior distribution is monitored by computing between-chain variance and within-chain variance (Gelman et al., 2003). If there are m Markov chains of length n then we can compute between-chain variance B with

$$B = \frac{n}{m-1} \sum_{j=1}^{m} (\overline{M}_j - \overline{M})^2 \tag{5}$$

where

$$\overline{M}_j = \frac{1}{n} \sum_{i=1}^{n} M_{ij} \tag{6}$$

is the average value along each Markov chain (a sample from a given chain is denoted by M_{ij}) and

$$\overline{M} = \frac{1}{m} \sum_{j=1}^{m} \overline{M}_j \tag{7}$$

is the average of the values from all Markov chains. The within-chain variance W is

$$W = \frac{1}{m} \sum_{i=1}^{m} s_i^2 \tag{8}$$

where

$$s_i^2 = \frac{1}{n-1} \sum_{j=1}^{n} (M_{ij} - \overline{M}_i)^2 \ . \tag{9}$$

An estimate of the variance of M is computed as

$$\mathrm{var}(M) = \frac{n-1}{n}W + \frac{1}{n}B \tag{10}$$

The convergence parameter, R, is then computed as

$$R = \frac{\mathrm{var}(M)}{W} \ . \tag{11}$$

The necessary condition for statistical convergence to the posterior distribution is that R approaches unity (Gelman et al., 2003). In practice, this is not always a sufficient condition for convergence, as seen below and in other studies (Delle Monache et al., 2007).

2.3 Source strength scaling

Typically the MCMC sampling requires thousands of iterations (samples) to converge to the posterior distribution, thus requiring thousands of forward dispersion model calculations. With simple Gaussian puff models (Johannesson et al., 2004) or Lagrangian particle tracking (Delle Monache et al., 2007) it is possible to calculate the forward models on the fly. With a three-dimensional CFD model, the computational cost quickly becomes prohibitive even for the simplest cases. For the current applications, we have simplified the situation for demonstration purposes by considering only steady-state flow conditions. (The chosen methodology remains completely general and can handle unsteady and reactive flows.) By assuming that the advection-diffusion problem is linear (e.g., no chemical reactions) we can use the precomputed steady flow field and Green's funcions to carry out one forward simulation at each of the thousands of locations in our prior distribution using a unit source strength and storing the resulting values at the sensor locations in a database. The source is modeled as a steady flux from one surface grid element. The stored concentrations can be rescaled depending on the proposed source release rate for a particular source location. Thus, during the inversion process, the sampled x and y locations are mapped to the corresponding grid element and dispersion results from each possible source location are obtained from the database and rescaled according to the current sampled value for the source strength. In this way, 20 000 iterations for each of four Markov chains can be performed in about ten minutes of computational time on four Xeon 2.4 GHz processors.

2.4 Forward model description - FEM3MP

The stochastic inversion procedure relies on a forward model to calculate instances of predicted sensor measurements, D, for given source term parameters, M. Here we use FEM3MP (Gresho and Chan, 1998; Chan and Stevens, 2000), a three-dimensional, incompressible Navier-Stokes finite-element code able to represent complex geometries and simulate flows in urban environments (Chan and Leach, 2004, 2007). Here FEM3MP is used in a Reynolds-Averaged Navier-Stokes approach.

For the example of flow around an isolated building, the model is driven by a steady logarithmic inflow profile at the upstream (west) boundary. Natural (i.e. zero tangential and normal stress) outflow boundary conditions are applied at the other boundaries. The steady-state flow field is pre-computed and is used to drive dispersion from a source with a constant release rate until a steady-state concentration field is obtained. The grid resolution is uniform far from the building, and is doubly fine near the corners of the building (see Fig. 6 later).

For the Oklahoma City simulations, we use setups similar to Chan and Leach (2007) for the third and ninth intensive observing periods (IOP3 and IOP9) from Joint URBAN 2003. Again, the flow field is assumed steady, with a logarithmic inflow profile on the southern and western boundary for IOP3 and on the southern boundary for IOP9. The wind speed is set to 6.5 m/s at $z = 50$ m with a wind direction of 185° (south-southwest) for IOP3 and similarly 7.2 m/s and 180° (south) for IOP9. The inflow profiles are based on upwind field observations near the computational domain (Chan and Leach, 2007). The flow field is pre-computed using FEM3MP. The release rate is constant (0.005 kg/s for IOP3 and 0.002 kg/s for IOP9) and simulations are performed until steady-state concentration fields are achieved (after about 10 minutes of simulation time). The atmosphere is assumed to be neutrally stratified since shear production of turbulence due to buildings is significantly larger than buoyant production (Lundquist and Chan, 2007). A standard eddy viscosity RANS turbulence model was used for IOP3, and a non-linear model was used for IOP9 (Chan and Leach, 2007). Buildings near the source are explicitly resolved, i.e., velocities and concentration within the buildings are set equal to zero. Far from the source, "virtual buildings" are used to reduce the computational cost. In this region, a drag force of very large value is added to the momentum equations for grid cells falling within the building boundaries. Previous work has shown that this approach produces satisfactory dispersion estimates far from the source (Chan and Leach, 2007).

3 Isolated building example

We have developed a prototype example of event reconstruction for a flow around an isolated building (a cube) with a source located upwind from the building (see Fig. 1). Four sensors are placed in a diamond-shaped array in the lee of the building. Data at the sensor locations is collected using a forward simulation from the true source location. The data is thus "synthetic" and used in this case only to test the inversion algorithm. Artificial measurement error with a standard log-normal distribution is also added to the synthetic data (in this case with mean $\mu = 0$ and standard deviation $\sigma_{rel} = 0.05$).

The source release rate was set to 0.1 (nondimensional units). As can be seen from Fig. 1 the actual source is located just above the symmetry line. Because the symmetry line is also the separatrix of this flow, this small deviation of the source location from the line of symmetry results in significant asymmetry in the resulting plume (Fig. 1). This example, while simple in geometry, thus incorporates complexities due to its three-dimensional nature that were not accounted for in previous inversion studies. The asymmetry of the plume is generated purely by the presence of the building. More simplistic dispersion models do not explicitly resolve buildings and hence cannot capture such features (Britter and Hanna, 2003).

The domain is discretized using about 19 000 elements (42×32×14). Forward runs are computed for all possible locations (on $z = 0$) and concentrations values at the sensors are stored in a database for each grid location. Total computation time for generation of the database was 6 hours using 64 2.4 GHz Xeon processors. The reconstruction or inversion algorithm proceeds as usual, but instead of running a new simulation for each proposed Markov chain step, the results are drawn from the concentration database, as described previously. This avoids repeated computations of releases at the same x, y locations by simply scaling the release rate as dictated by the sampling algorithm.

3.1 Source inversion

Figure 2 shows the points sampled by the four Markov chains. The chains quickly converge on the source location, sampling more frequently in the northern half of the domain as expected due to the asymmetry of the actual plume. The probability distribution for the source location is given in Fig. 3, which also reflects the asymmetry of flow. The peak of the distribution occurs just upwind of the actual source location. If the error from the measurements is set to zero (i.e. $\sigma_{rel} = 0$), the inversion procedure accurately predicts the source location as expected (i.e. the peak of the probability distribution matches the true source; not shown). The probability distribution is constructed using the second half of the MCMC iterations (i.e. 10 000 to 20 000), to allow the Markov chains to “mix” adequately to improve the statistical distribution and to exclude the random initialization from the final statistics. Thus, the so-called “burn-in” time is 10 000 iterations. The corresponding probability distribution for the source release rate is shown in Fig. 4. The peak of the histogram coincides with the actual release rate of 0.1.

Convergence rates for the x, y and q inversions are shown in Fig. 5. All convergence measures reach a value near 1.1 after about 10 000 iterations, indicating that the sampling procedure was thorough and adequate to generate a meaningful posterior probability distribution. Note that the convergence rate is independent of the spread in the distribution, and merely indicates that further sampling will not likely change the results. We are thus able to successfully invert this idealized three-dimensional dispersion problem and determine the source location and release rate to within a tight confidence region.

3.2 Composite plume

In addition to probabilistic predictions of the source location, emergency responders need predictions of concentrations over the entire plume area. A “most likely plume” could easily be constructed by performing a forward simulation from the peak of the probability distribution for the source location. This, however, would be one realization and would not contain the probabilistic information inherent in the reconstruction procedure.

We therefore construct a probabilistic, composite plume, from the plume realizations corresponding to all the samples from the posterior probability distribution of source term parameters. The composite plume is obtained by first creating histograms of concentration values at each spatial location in the domain using results from all iterations beyond the “burn-in” time. This step is followed by determining the concentration value at each location for which a certain pre-specified probability is exceeded. Contours of the 90% confidence interval are shown in Fig. 6. For values above the threshold (chosen to be 0.03), the plot shows 90% confidence that the concentration at a given location is higher than the contoured value. For values below the threshold, the contours indicate 90% confidence that the concentration is less than the contoured value.

The shape of this composite plume is quite different from that of the actual plume (Fig. 1). The composite plume represents a probabilistic estimate of concentrations and could aid in emergency response decisions for evacuation or sheltering in place depending on a chosen confidence interval and whether an area lies above or below a threshold value for toxicity.

4 Oklahoma City - Joint URBAN 2003 IOP3

The OKC domain for IOP3 includes the central business district, with a maximum building height of 120 m and an average building height of 30 m. Figure 7 shows the complexity of the wind flow in the downtown area during IOP3 generated using FEM3MP with steady inflow boundary conditions on the southern and western edges of the domain. Comparisons of dispersion results are made to 30-min averages of concentration measured at fifteen sensors within this domain. The domain is discretized using about 580 000 elements (132,146,30) covering a region of approximately $x = [-260, 346], y = [-68, 590]$. The prior distribution is limited to a somewhat smaller domain ($x = [-150, 130], y = [80, 410]$) to reduce computation time. The source strength was allowed to vary from 0.00001 to 1.0 kg/s with a mean of 0.5 and standard deviation $q_\sigma = 0.5$. Standard deviations for the location sampling were set to $x_\sigma = y_\sigma = 100$ m with means near the center of the sample domain at $x = 0$ m and $y = 80$ m. Standard deviations for source location and strength were determined by the problem domain size and refined with a trial and error procedure to ensure that the Markov chains had access to realistic ranges with minimal occurrences of “stuck” chains. Stuck chains can occur when the standard deviations chosen for the next iteration lead to a large number of rejected samples such that the chain remains in a given position for many iterations.

In addition, the cell spacing was effectively doubled by only considering sources in every other grid cell in a checkerboard pattern. Total computation time for 2560 forward runs (from each possible source location in the concentration database) was over 12 hours using 1024 2.4 GHz Xeon processors (equivalent to 17 days on 32 processors). Each forward run of FEM3MP simulataneously calculated 20 different source locations, requiring 128 different launches of the model. Each instance of the model used 32 processors. After generation of the database, the inversion process itself requires less than ten minutes of computation time on four processors.

4.1 Source inversion

Figure 8 shows the location of the buildings and 15 sensors in the downtown OKC area, together with four Markov chain paths. The chains quickly converge from four random initial locations to the general vicinity of the actual source location where they spend the remainder of their time sampling the parameter space and refining the probability distribution. Using the Markov chain paths, we construct the probability distribution for the source location, as shown in Fig. 9. The peak of the distribution is located approximately 70 meters south of the actual source location. Reasons for this will be discussed below. The accompanying release rate histogram is given in Fig. 10. The peak of the distributions falls near 0.001 kg/s, while the actual source strength was 0.005 kg/s.

Figure 11 shows convergence rates for x, y and q during the 20 000 iterations of the inversion procedure for OKC IOP3. The values for x, y and q converge after 10 000 iterations and only change slightly after that. The value for y is sometimes more difficult to pinpoint in the inversion process. Here y is the stream-wise direction, where a change in the distance to the source can sometimes be accommodated by a corresponding change in release rate. That is, a weaker source closer the sensor can sometimes produce similar results to a stronger source further away. Therefore, a value of $R = 2$ for the y location of the source can be considered acceptable.

A closer look at the individual plumes predicted by different source locations gives insight into the location of the peak of the x, y probability distribution. Figure 12 shows the plume predicted by FEM3MP for a source at the actual source location for IOP3 with the actual release rate. Contours of concentrations predicted by FEM3MP are shown together with small squares at the sensor locations colored according to the 30-min averaged observed concentrations during IOP3. Figure 13 shows the plume from the inverted source location, i.e. the peak of the x, y probability distribution for the source location. While the plumes predicted by the code seem reasonable, there are clearly discrepancies between the predicted concentrations and observations for both simulated plumes. These can be seen more clearly in a comparison of observed and modeled values at the 15 sensor concentrations. The inverted source location was determined by the stochastic inversion algorithm which minimizes the absolute error between modeled and observed values. Indeed, the sum of the absolute errors (Fig. 14) at the sensor locations is smaller using the inverted source location ($\sim$ 986 ppb total) than the true source location ($\sim$ 2733 ppb total). A discussion of model errors is given below.

4.2 Composite plume

We again construct a probabilistic, composite plume, representing the probability of concentration at a specific location being higher or lower than a certain value. Contours of the 90% confidence interval are shown in Fig. 15 with the threshold chosen at 10 ppb. Again we note that the shape of this composite plume is quite different from any individual realization or plume prediction such as those shown in Figs. 12 and 13. The white region indicates a lack of information and the inability to specify a 90% confidence interval at those locations (this region is dependent on the choice of the threshold value). The dark blue region envelopes the composite plume, indicating regions where there is 90% confidence that the concentrations are less than 0.01 ppb.

4.3 Treatment of model errors

The inversion procedure clearly relies heavily on the accuracy of the sensor measurements as well as the accuracy of the forward model used for dispersion simulations. While the FEM3MP code has been validated for many urban flows, there are several possible sources of error. To obtain a good probabilistic distribution for the source location and strength, all sources of error must be included a priori.

There are several reasons for the mismatch in predicted and observed concentrations. First of all, most of the observed concentrations are averaged values from a 30-min release, whereas model predictions are steady-state results. Additionally, there are uncertainties in the lateral boundary conditions prescribed in the simulation. Steady inflow has been specified for the inflow boundary, whereas in reality the wind at the domain boundary has fluctuations in space and time. A slight change in mean wind direction can greatly affect dispersion results. Chan and Leach (2004) demonstrated that time-varying inflow boundary conditions significantly changed the concentration plume in simulations of dispersion in Salt Lake City. In addition, to save computation time, the domain size used for the IOP3 simulations is smaller than for those performed by Chan and Leach (2007) for OKC, which perhaps increases the influence of the boundaries. We also use a simplified linear eddy-viscosity turbulence model for computational cost reasons, whereas Chan and Leach (2007) used a non-linear eddy-viscosity model which gives better agreement with the data but at a much higher computational cost.

The non-linear eddy-viscosity model often better represents dispersion in regions of building-induced turbulence, hence giving better agreement with observed concentrations as in Chan and Leach (2007). This eddy-viscosity model is used for IOP9 below.

Another potential source of error is in the specification of the source term in the simulation. While the tracer gas was released from a point source in the experiment, the model distributes the source over a grid cell, where the vertical injection velocity and concentration are specified at the boundary to match the release rate from the experiment. This yields a nearly steady concentration flux over the grid cell but with numerical oscillations (see region near the source in Fig. 12) in the neighboring cells due to the strong concentration gradients and insufficient grid resolution in the source area.

It is difficult to quantify the individual contributions of the multiple sources of error in FEM3MP. Model errors are therefore incorporated into the inversion process in a simple, lump sum fashion by adjusting σ_{rel} of the standard log-normal distribution, the relative error allowed in the comparison between different realizations of the simulation and the observed values. For the OKC simulations, σ_{rel} was set to the relatively high value of 0.5.

5 Oklahoma City - Joint URBAN 2003 IOP 9

As a further example of the building-resolving inversion procedure, we have also applied the source inversion to observations obtained during IOP9. The IOP9 simulations use the full-sized domain as well as the more sophisticated three-equation non-linear eddy viscosity closure of Chan and Leach (2007) to eliminate the modeling compromises made in the IOP3 case. The experiment conditions (in particular the true source location), however, lead to a much more challenging situation for the inversion procedure and demonstrate a case where the procedure is much more sensitive to the data and forward model errors.

The IOP9 domain covers approximately $x = [-498, 530], y = [-430, 2580]$ using a grid of 201×303×45 (approximately 2.75 million grid points). Grid spacing in the horizontal is as fine as 1-2 meters in the vicinity of resolved buildings and 1 m near the surface in the vertical direction. Again, for computational reasons the prior distribution is restricted to a slightly smaller domain ($x = [-180, 200], y = [310, 525]$) and a checkerboard pattern is used to limit the total number of forward runs to 3360. The forward simulations required about 100 hours of wall clock time using 1024 processors. Standard deviations for the location sampling were again set to $x_\sigma = y_\sigma = 100$ m with means near the center of the sample domain at $x = 0$ m and $y = 400$ m. The source strength was allowed to vary from 0.00001 to 1 kg/s. The inversion procedure for IOP9 required approximately ten minutes of computation time on four processors. The inversion time does depend on the choice of x_σ and y_σ and the proximity to buildings; the density of buildings in the IOP9 source region slows down the the sampling procedure since samples which fall within buildings are not allowed. Thus, a chain located in a narrow gap between buildings is limited in its choices for the next iteration; this requirement that samples not be located within buildings does not count toward a sample rejection or acceptance and only slightly slows the algorithm.

5.1 Source inversion

The resulting Markov chain paths and x-y probability distribution are shown in Figs. 16 and 17, respectively. The IOP9 experiment collected data from only 8 sensors (compared to 15 in IOP3), as shown in Fig. 16; model results are compared to 15-min averages of concentration at the sensor locations from 15-30 minutes after the release. The availability of fewer sensors, combined with the location of the source between two buildings, creates difficulties for the inversion procedure. Figure 17 indicates three probability peaks, clustered between different sets of buildings. The four Markov Chains converge to locations far south of the true source location, though time series of the x, y, q values do not clearly identify a single final source choice but continue to jump within the three peak regions of the probability distribution (not shown), contrary to the convergence rate plots shown in Fig. 20 which do indicate a trend of convergence (values of R less than 2) (Delle Monache et al., 2007). Probability distributions of the x, y, and q values are shown independently in Fig. 19. The y distribution shows three distinct peaks corresponding to the gaps between the buildings. The q distribution indicates good agreement in source strength; the inversion is able to pinpoint the source strength to within a narrow range from the original distribution of 0 to 1 kg/s: the peak indicates values of between 0.001 and 0.004 kg/s, which compares reasonable well with the true source strength for IOP9 of 0.002 kg/s.

The peak of the full probability distribution (from Fig. 17) near $x = 5m, y = 385m$ and the diffuse peak in the region of $x = 5m, y = 320m$ do give smaller errors (as indicated by the reconstruction) compared to simulation results from the actual source location at $x = 30m, y = 435m$. When restricting the x, y probability distribution to source strength values q within 50% of the true source strength (0.001-0.003 kg/s), the resulting conditional probability distribution shows a peak (near $y = 440$ m) within 50 m west of the actual source location (see Fig. 18).

5.2 Composite plume

Figure 22 shows the resulting composite plume using the IOP9 inversion data. The shape of the plume is entirely different than any single realization. The broadness of the compositie plume shapes reflects the uncertainty in the inversion procedure, a natural property of our stochastic inversion procedure. The composite plume is unable to indicate concentration levels with any specificity in this case, it merely delineates regions where it is 90% likely that the concentration will be greater than 10 ppb (green) and 90 % likely that it will be less than 0.1 ppb (blue).

5.3 Discussion of model errors

The complexity added by the presence of the source location between two buildings perpendicular to the flow direction appears to challenge the inversion procedure more than in the case of IOP3, but this is largely due to errors in the model predictions in comparison with the sensor observations due to reasons mentioned above. A test of this hypothesis was performed using synthetic sensor data. With the source placed at the true location and using the true source strength, the forward model results were collected at the 8 sensors to create a synthetic observation dataset. Inversion results using the synthetic data are shown in Fig. 21. All four Markov chains used in the inversion correctly identified the true source under these circumstances, using the same inversion parameters used for the standard IOP9 run. Several values of σ_{rel}, x_σ, and y_σ were tested to determine sensitivity to these choices. One chain occasionally converged to a location far from the source (depending on x_σ and y_σ, not shown) indicating that the building geometry also adds complexity to the flow so that multiple source locations are possible even when model error is largely removed by using synthetic data. The choice of inversion parameters is also important in determining the rate and accuracy of convergence, as discussed further below.

6 Effect of sensor density

A common question for urban planners to consider is the placement of chemical detecting sensors in regions of high interest, for example near dense or high occupancy buildings in urban areas. Sensor network design is easily evaluated using our stochastic algorithm which can be used to indicate the importance of a sensor to source inversion in a particular region

(Lundquist, 2005). A related question exists with regard to the number of sensors required for accurate source inversion. The appropriate number depends generally on the complexity of building geometries and ambient wind conditions.

We have evaluated the latter question using IOP3 as our test case. Of the available 15 sensors, inversion procedures were carried out using 8, 4, 2 and then just 1 of these sensors. The corresponding probability distributions are shown in Figure 23. As expected, the probability distribution broadens significantly as the number of sensors is reduced, reflecting the increased uncertainty due to the fewer data points involved. Nevertheless, results with 2 sensors are still able to identify the general region of the source, thus indicating that even as few as 2 sensors may be useful in an urban environment, provided they are deployed at the appropriate locations. With one sensor, the probability distribution becomes much more sensitive to model and observation errors. Figure 23d shows that if a sensor is used with a zero reading, it simply outlines the region where the source cannot be located. In this case, however, this outline is incorrect since the model is not able to reproduce the zero reading even when the source is in the correct location. Choosing another sensor (Fig. 23e) with a non-zero reading produces better results.

7 Discussion and conclusions

Our stochastic methodology for source inversion is based on Bayesian inference combined with a Markov Chain Monte Carlo sampling procedure. The stochastic approach used in this work is computationally intensive but the method is completely general and can be used for time-varying release rates and flow conditions, non-linear problems, and problems characterized by non-Gaussian distributions. The results of the inversion, specifically the shape and size of the posterior probability distribution, indicate the probability of a source being found at a particular location with a particular release rate, thereby inherently reflecting uncertainty in observed data or the data's insufficiency with respect to quality, or spatial or temporal resolution.

We have demonstrated successful inversion of a prototype problem with flow around an isolated building. Application to the complex conditions present during IOP3 and IOP9 of the Joint URBAN 2003 experiment in Oklahoma City also proved successful. Despite the many sources of error present in the comparison of model predictions with observed data during the inversion procedure, the peak of the probability distribution for the source location was within 70 m of the true source location for IOP3, and the actual source location was contained within the top percentiles of the probability distribution. For IOP9, model errors and other uncertainties limited the ability of the inversion procedure to exactly pinpoint the true source, though the source was contained within the broader distribution. A composite plume showing concentrations at the 90% confidence level was created for all three cases using plume predictions from the realizations given by the reconstructed probability distribution. This composite plume contains probabilistic information from the iterative inversion procedure and can be used by emergency responders as a tool to determine the likelihood of concentration at a particular location being above or below a threshold value. The effect of sensor density was also evaluated for IOP3 and found to give expected increases in the spread of the source probability distribution with a decrease in the number of sensors available.

Uncertainties in the inversion procedure increase with the complexity of the domain, paralleling the errors in the forward model. Because the probability distributions are able to reflect the uncertainty in source location, the source inversion procedure demonstrated here indicates high potential to be a useful tool for emergency responders regardless of model limitations. The building-resolving capability introduced here will enable source locations to be pin-pointed with high-resolution. The limiting factor to real-time response situations is currently the large computation time required. It is, however, conceivable that the building-resolving CFD model could be coupled with a simpler forward model (e.g. LODI, the Lagrangian particle model used by Delle Monache et al. (2007)) to reduce the prior distribution to a reasonable size. Thus, LODI can be used over a large urban region with our stochastic inversion algorithm and inversion with FEM3MP could follow to pin-point the source within the subregion identified by the LODI inversion. It is also conceivable that databases could be generated in advance for specific urban areas for plume predictions from CFD simulations, similar to the databases generated for IOP3 and IOP9 in this work. Forward runs could be performed for a variety of prevailing wind directions. In an emergency situation, the inversion procedure could be performed in a very reasonable time frame, on the order of ten minutes.

Efforts to reduce forward model errors are underway in parallel research programs; forward model capabilities do not inhibit the stochastic algorithm in any way, though practical applications of course depend on both. Further experience with inversion procedures in urban areas will lead to a better grasp of the range of the inversion parameter space most suitable for dense urban areas with complex building geometries. Future work will also include investigation of unsteady releases, unsteady flow conditions, and elevated sources. Meteorological uncertainty will also be incorporated to allow for errors induced by lack of sufficient information at the lateral boundaries such as errors in the specified mean wind direction.

Acknowledgements

Thanks are extended to Roger Aines, Luca Delle Monache, Kathy Dyer, William Hanley, and John Nitao for their contributions to this work. Computations were performed on Linux clusters at LLNL's computing center. This work was performed under the auspices of the U. S. Department of Energy by the University of California, Lawrence Livermore National Laboratory (LLNL) under Contract No. W-7405-Eng-48. The project (04-ERD-037) was funded by the Laboratory Directed Research and Development Program at LLNL.

References

Allwine, K. J., 2004. Joint Urban 2003 field study and urban mesonets. Fourth Symposium on Planning, Nowcasting, and Forecasting in the Urban Zone, American Meteorological Society.

Bennett, A., 2002. Inverse Modeling of the Ocean and Atmosphere. Cambridge Univ. Press.

Britter, R., Hanna, S., 2003. Flow and Dispersion in Urban Areas. Ann. Rev. Fluid Mech. 35, 469–496.

Chan, S. T., Leach, M., 2007. A validation of FEM3MP with Joint Urban 2003 data. J. Appl. Meteor. Joint Urban 2003 Special Issue, in press.

Chan, S. T., Leach, M. J., 2004. Large eddy simulation of an Urban 2000 experiment with various time-dependent forcing. Paper 13.3. Fifth Symposium on the Urban Environment, American Meteorological Society.

Chan, S. T., Stevens, D. E., 2000. An evaluation of two advanced turbulence models for simulating flow and dispersion around buildings. The Millen. NATO/CCMS International Technical Meeting on Air Pollution Modeling and its Application, 355–362.

Chow, F. K., Kosović, B., Chan, S. T., 2006. Source inversion for dispersion in urban environments using building-resolving simulations and bayesian inference with stochastic sampling. Paper J4.4. 6th Symposium on the Urban Environment, American Meteorological Society, 9 pages.

Delle Monache, L., Lundquist, J. K., Kosović, B., Johannesson, G., Dyer, K. M., Aines, R. D., Belles, R. D., Chow, F. K., Hanley, W. G., Larsen, S. C., Loosmore, G. A., Mirin, A. A., Nitao, J. J., Sugiyama, G. A., Vogt, P. J., 2007. Bayesian inference and Markov Chain Monte Carlo sampling to reconstruct a contaminant source at continental scale. Atmospheric Environment submitted.

Enting, I., 2002. Inverse Problems in Atmospheric Constituent Transport. Cambridge Univ. Press.

Gelman, A., Carlin, J., Stern, H., Rubin, D., 2003. Bayesian Data Analysis. Chapman & Hall/CRC.

Gilks, W. R., Richardson, S., Spiegelhalter, D. J., 1996. Markov Chain Monte Carlo in Practice. Chapman & Hall/CRC.

Gresho, P., Chan, S., 1998. Projection 2 goes turbulent and fully implicit. International Journal of Computational Fluid Dynamics 9, 249–272.

Johannesson, G., Chow, F. K., Glascoe, L., Glaser, R. E., Hanley, W. G., Kosović, B., Krnjajić, M., Larsen, S. C., Lundquist, J. K., Mirin, A. A., Nitao, J. J., Sugiyama, G. A., 2005. Sequential Monte-Carlo based framework for dynamic data-driven event reconstruction for atmospheric release. Proceedings of the Joint Statistical Meeting, 8 pages.

Johannesson, G., Hanley, W., Nitao, J., 2004. Dynamic Bayesian models via Monte Carlo - an introduction with examples. Tech. Rep. UCRL-TR-207173, Lawrence Livermore National Laboratory, Livermore, CA.

Keats, A., Yee, E., Lien, F.-S., 2007. Bayesian inference for source determination with applications to a complex urban environment. Atmospheric Environment 41, 465–479.

Lundquist, J. K., 2005. Synthetic event reconstruction experiments for defining sensor network characteristics. LLNL technical report, UCRL-TR-217762, Lawrence Livermore National Laboratory.

Lundquist, J. K., Chan, S. T., 2007. Consequences of urban stability conditions for computational fluid dynamics simulations of urban dispersion. J. Appl. Meteor. and Climatol. in press.

Ramirez, A. L., Nitao, J. J., Hanley, W. G., Aines, R., Glaser, R. E., Sengupta, S. K., Dyer, K. M., Hickling, T. L., Daily, W. D., 2005. Stochastic inversion of electrical resistivity changes using a Markov Chain Monte Carlo approach. Journal of Geophysical Research-Solid Earth 110 (B2), Art. No. B02101.

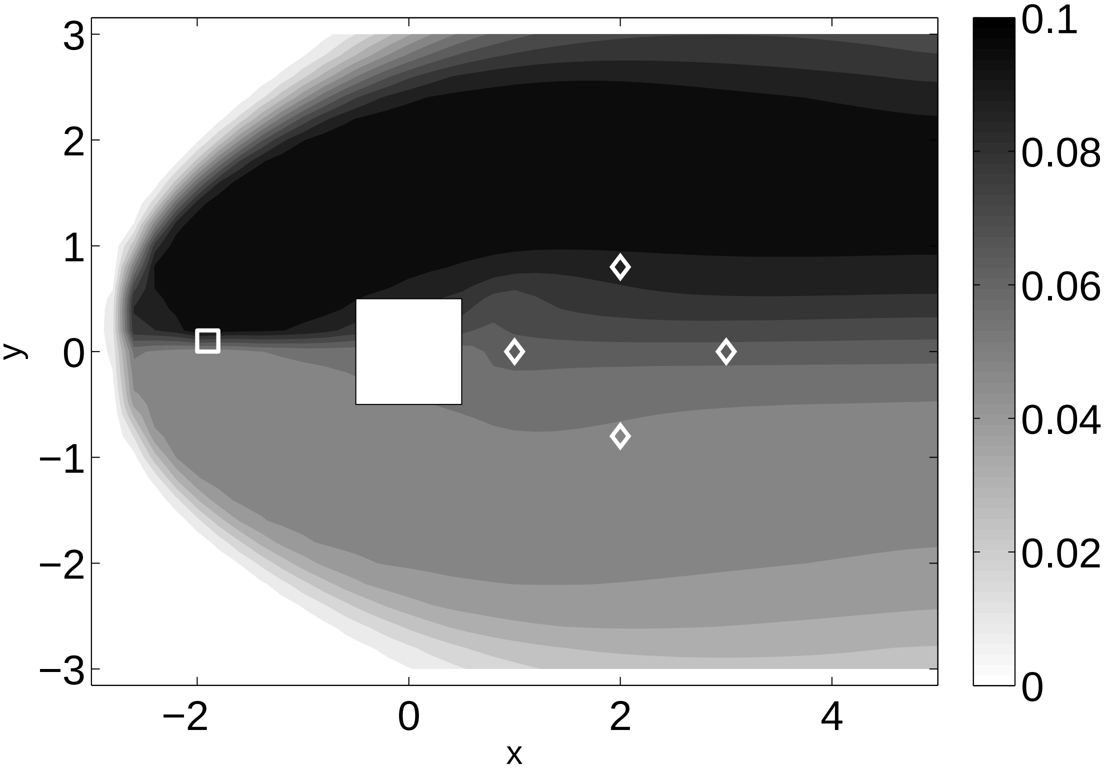


Fig. 1. Horizontal concentration contours at the first vertical level generated by forward simulation with FEM3MP for flow around an isolated building (white box). Four sensors are placed in the lee of the building (white diamonds). The source location is indicated by the white square.

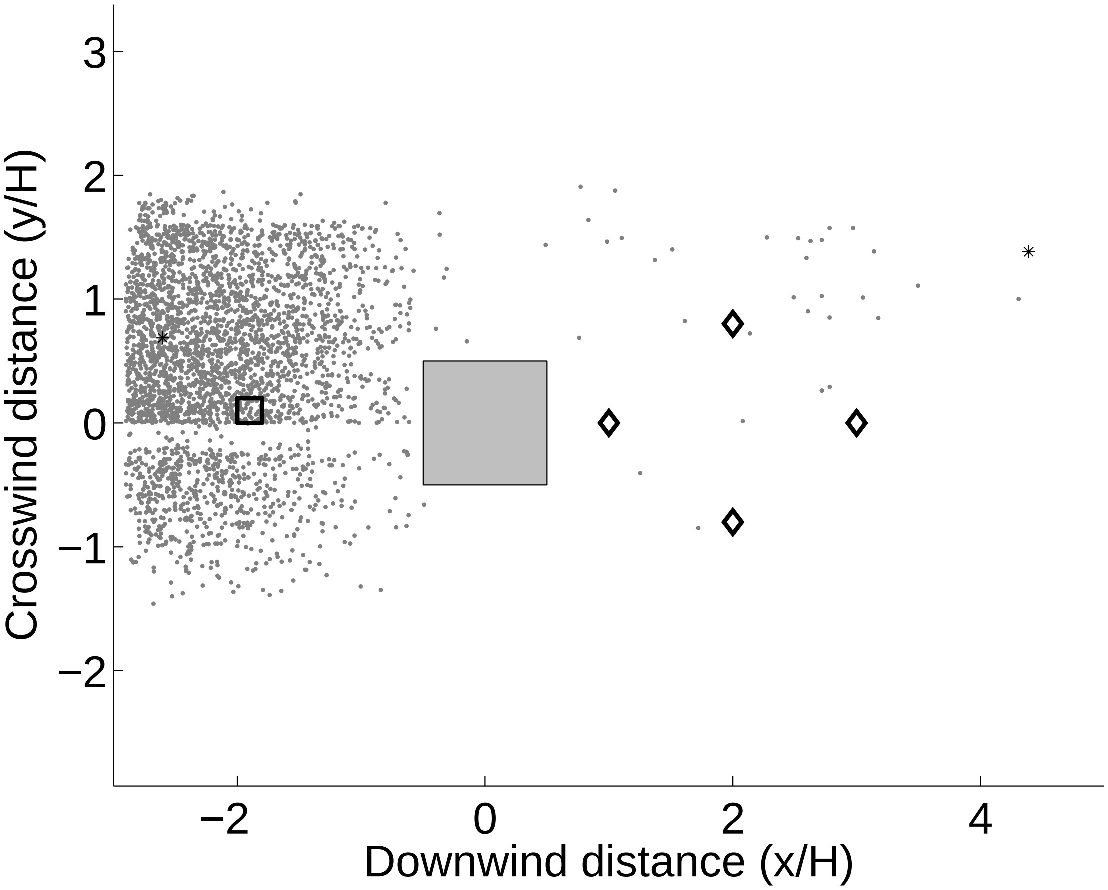


Fig. 2. Gray dots show locations sampled by the four Markov chain used for source inversion for flow around an isolated building (gray square at origin). Black diamonds indicate the four sensor locations. Black stars show the random starting points of the Markov chains (two are co-located at the origin). Small black square shows true source location.

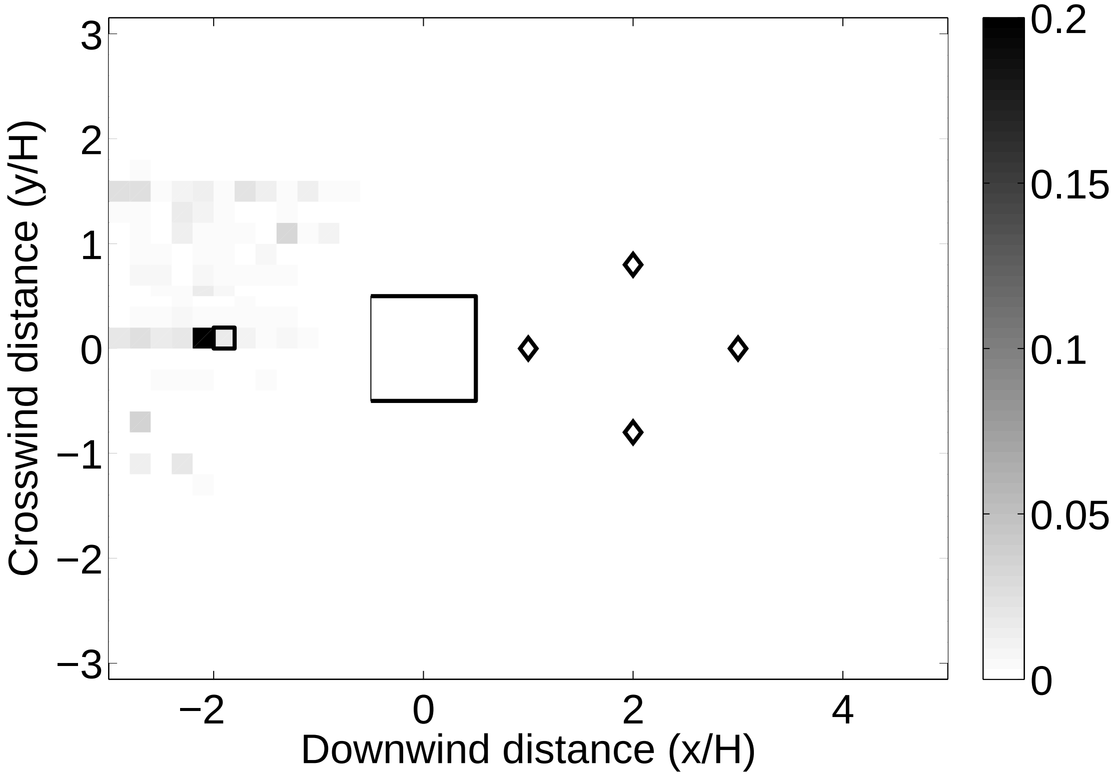


Fig. 3. Probability distribution of source location for flow around an isolated building. Black diamonds indicate the four sensor locations. Small black square shows true source location adjacent to peak of probability distribution.

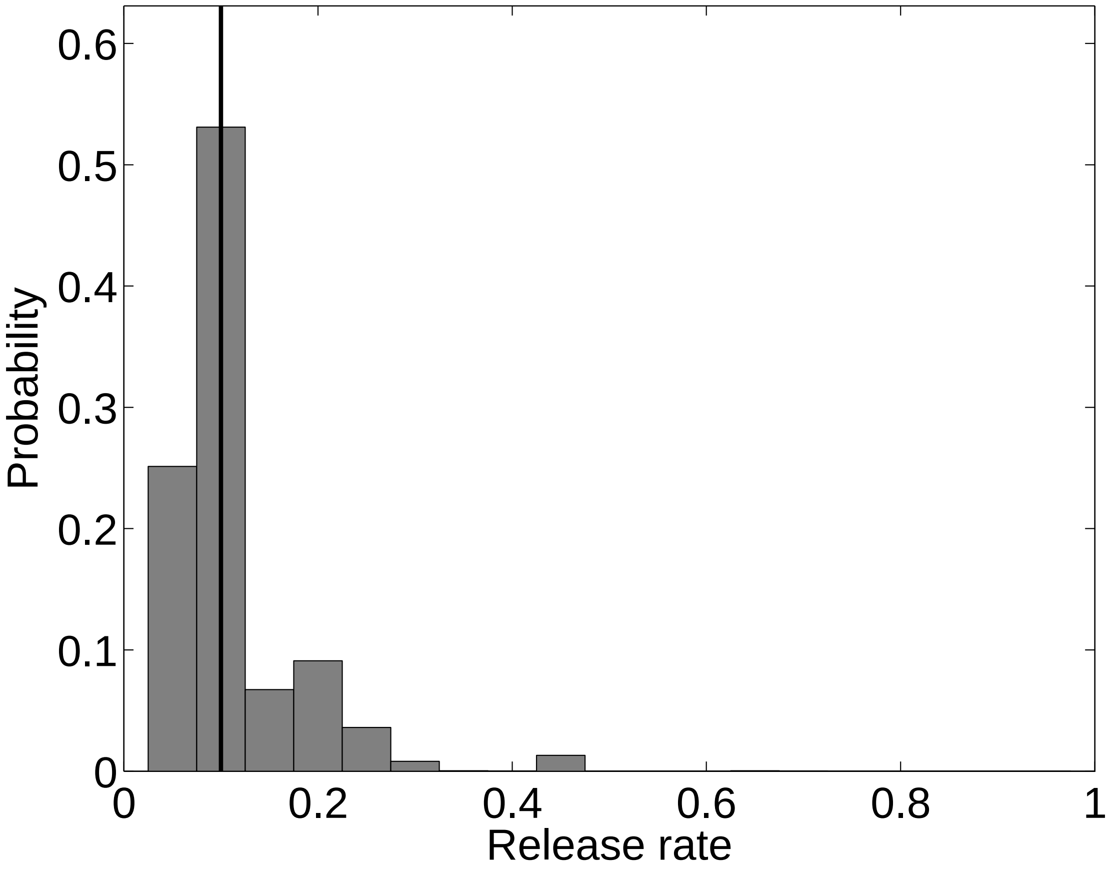


Fig. 4. Histogram of source strengths for flow around an isolated building. Solid vertical line shows actual release rate magnitude.

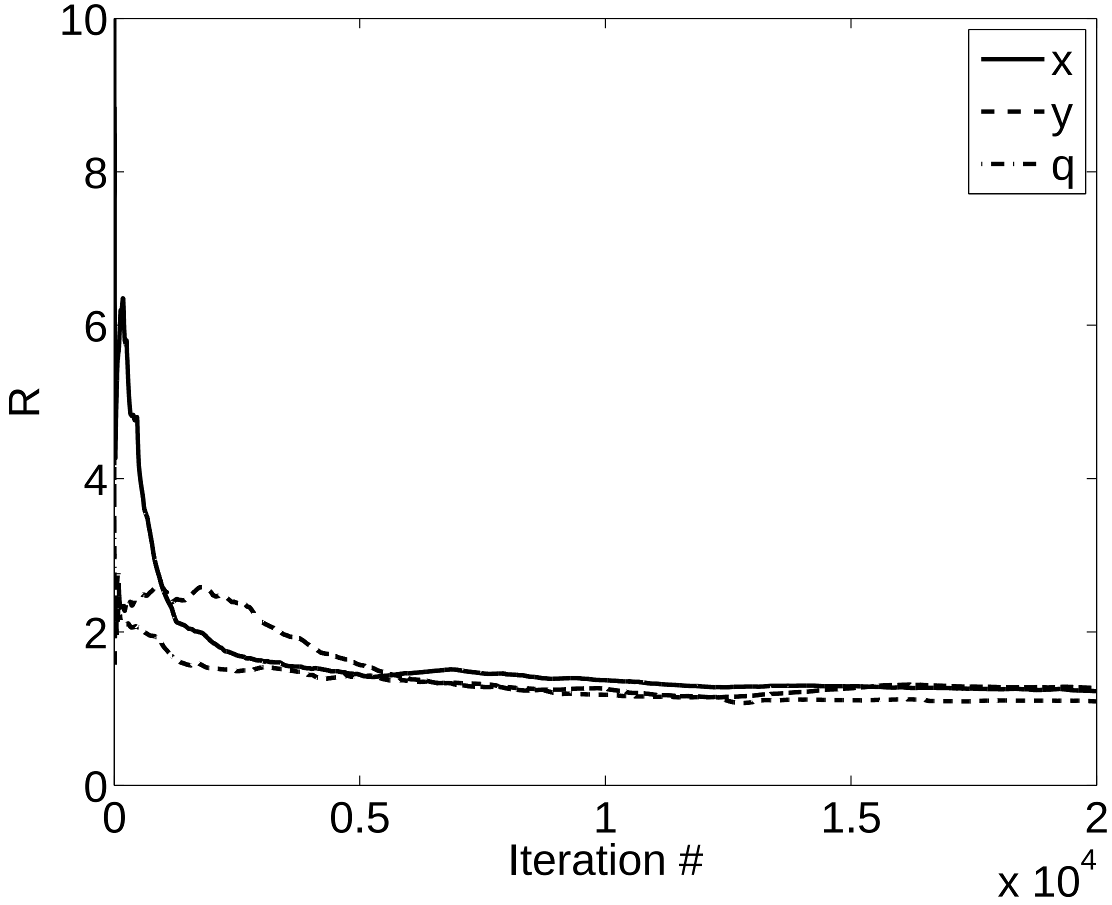


Fig. 5. Convergence rates for horizontal position (x, y) and source strength q for flow around an isolated building.

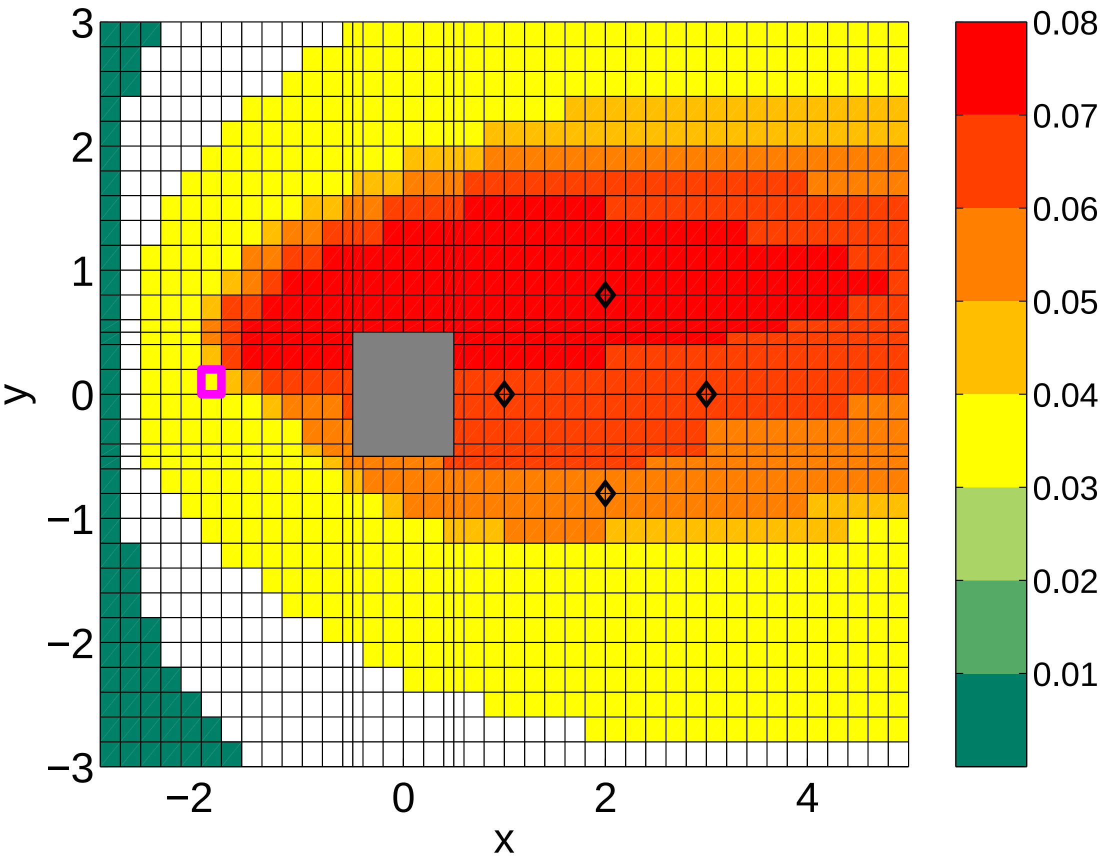


Fig. 6. Composite plume showing 90% confidence intervals for concentration levels for flow around an isolated building (white box). The threshold is set at 0.03. For concentrations above the threshold, there is 90% confidence that the concentration is higher than the contoured value. For values below the threshold, there is 90% confidence that the concentration is less than the contoured value. White region indicates that a 90% confidence interval cannot be established. Black diamonds indicate the four sensor locations. Small magenta square shows true source location.

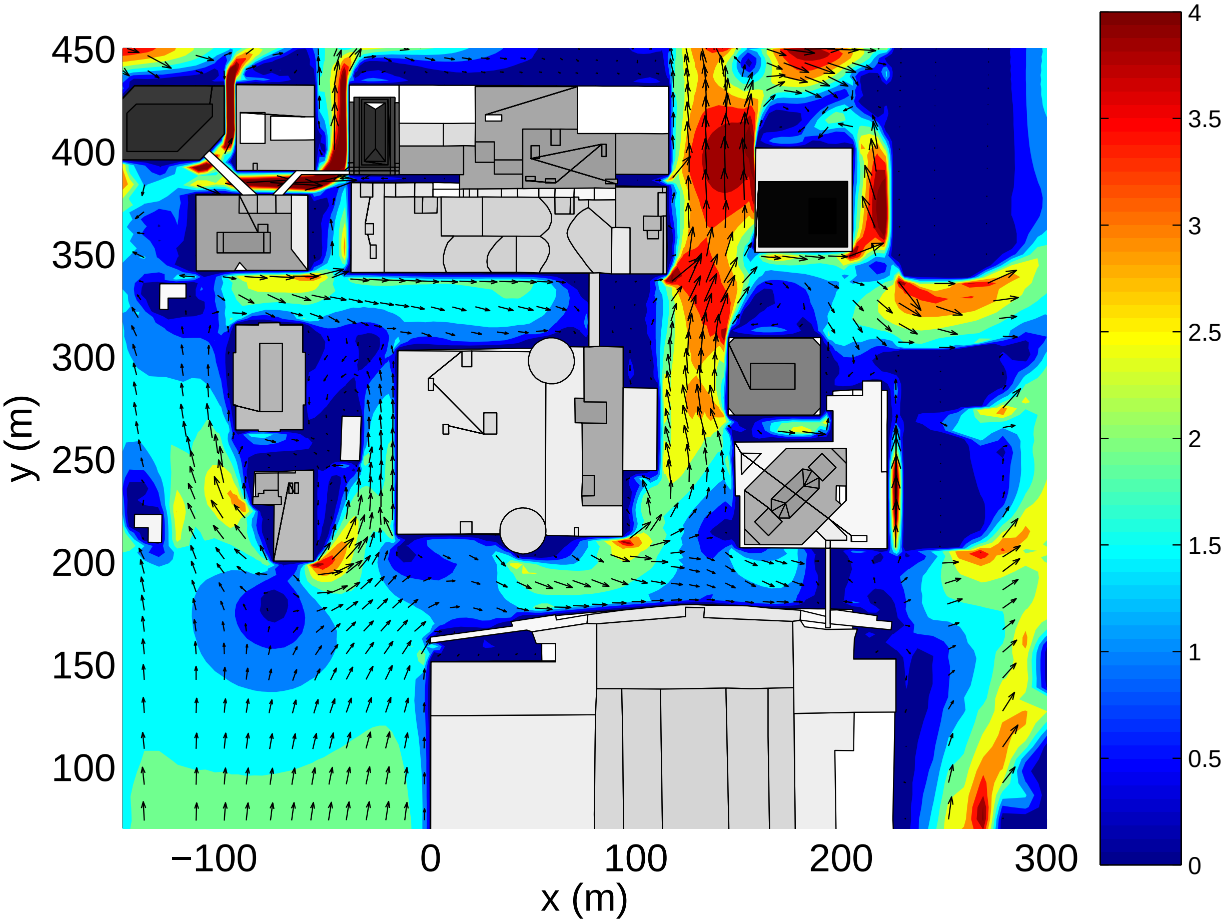


Fig. 7. Surface wind vectors (every third point shown in each direction) and contours of velocity magnitude (m/s) predicted by FEM3MP for flow in the central business district of Oklahoma City during IOP 3 of the Joint URBAN 2003 field experiment. Buildings are indicated with various shades of gray.

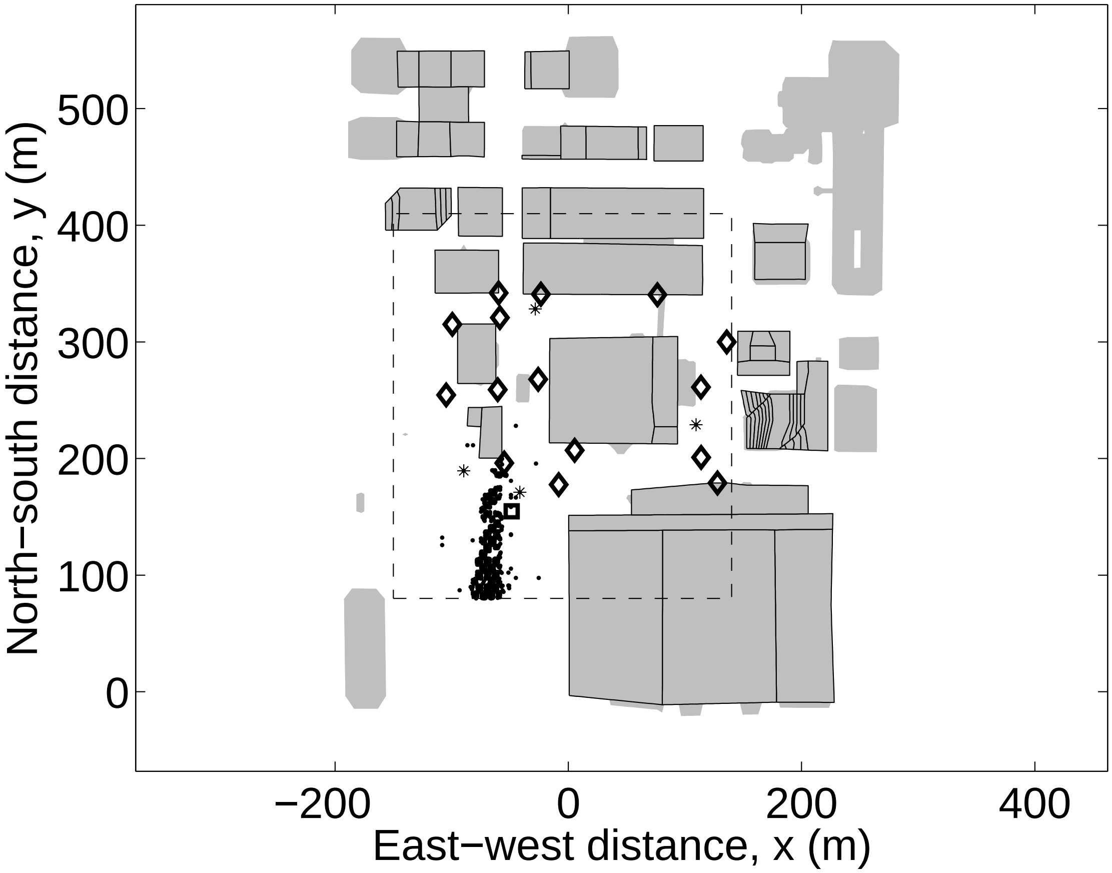


Fig. 8. Black dots show locations sampled by the four Markov chain used for source inversion for flow in Oklahoma City during IOP3. Black diamonds indicate sensor locations. Black stars show the random starting points of the Markov chains. Small black square shows true source location. Buildings that are treated explicitly are outlined in black in addition to shading; others are treated as virtual buildings. Dashed line shows zoomed-in region near the source used for later figures.

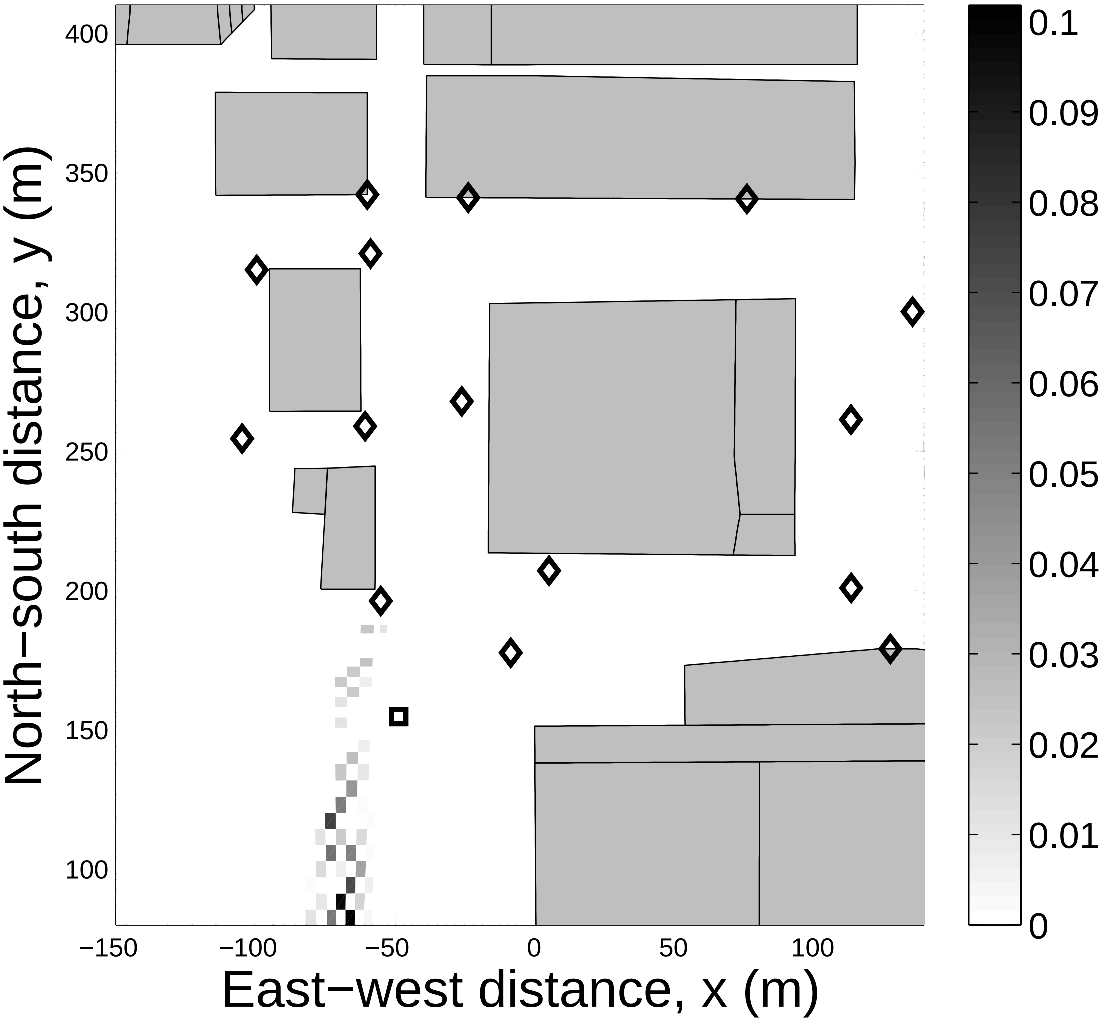


Fig. 9. Probability distribution of source location for flow in OKC during IOP 3. Only sub-domain indicated by dashed line in Fig. 8 is shown. Actual source location is shown by black square. Buildings are shaded in gray.

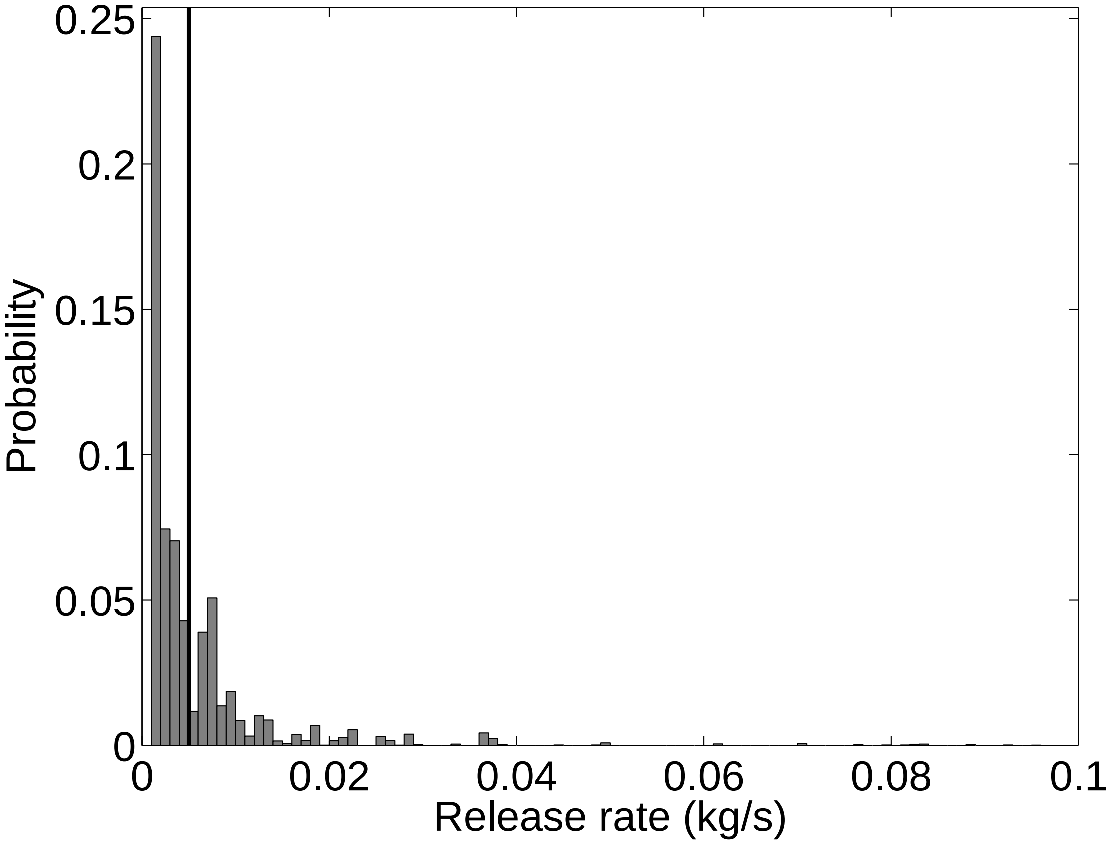


Fig. 10. Histogram of source strengths for flow in OKC during IOP 3. Solid vertical line shows magnitude of actual release rate.

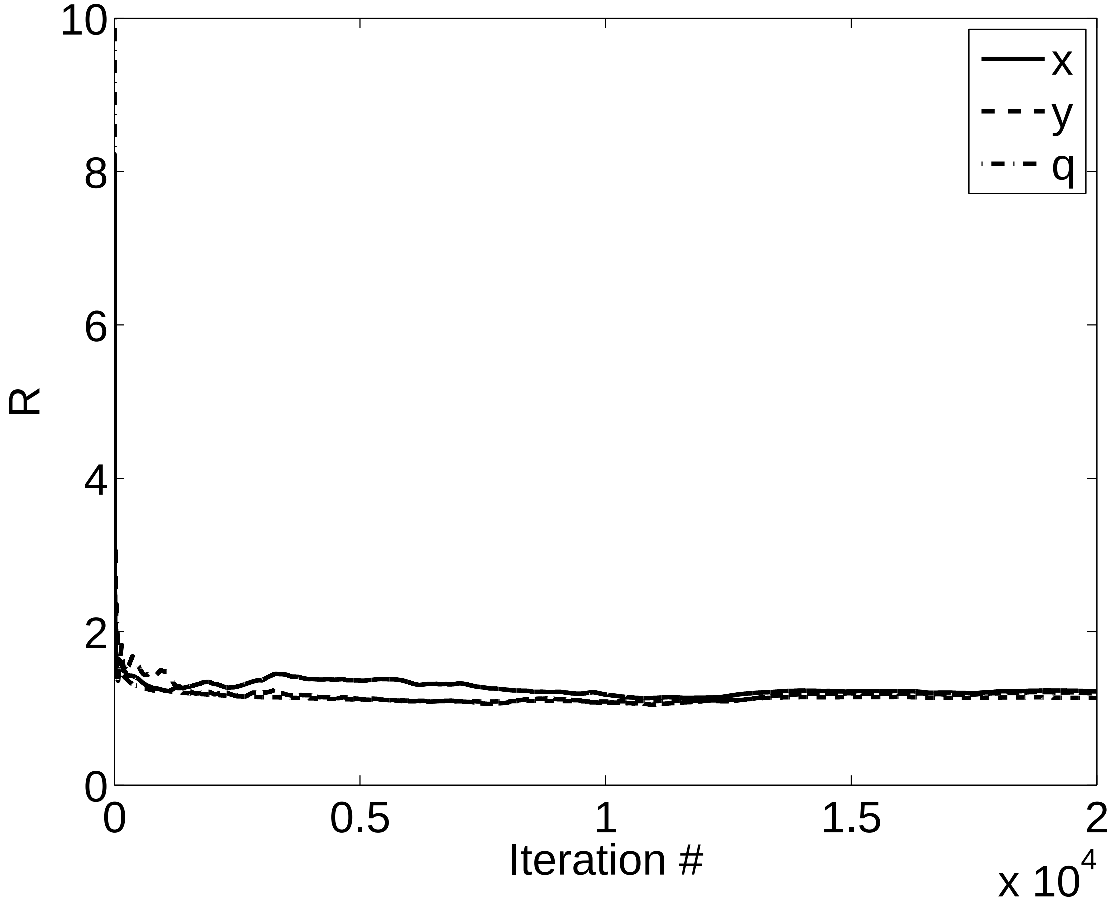


Fig. 11. Convergence rates for horizontal position (x, y) and source strength (q) for flow in OKC during IOP 3.

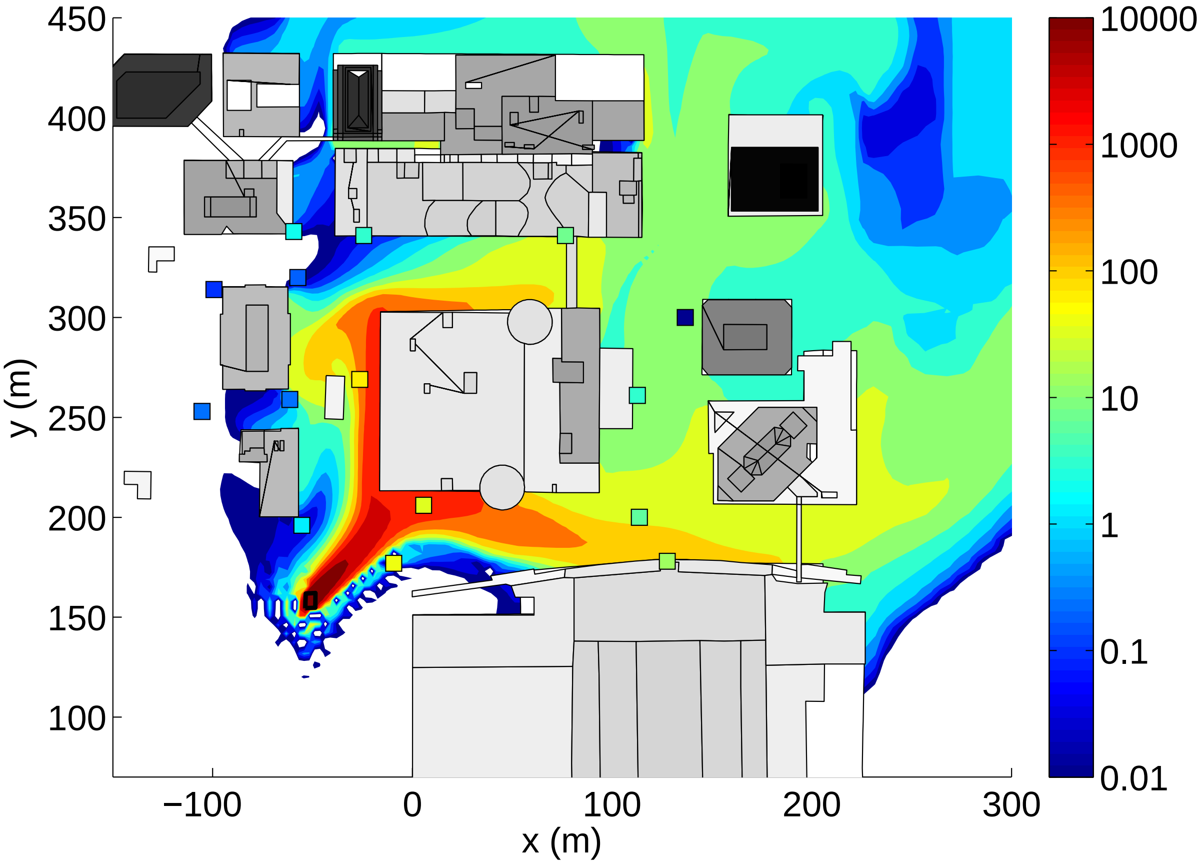


Fig. 12. Concentration plume predicted by FEM3MP with actual source location (small black square) and release rate for OKC IOP 3 compared to averaged concentration measurements (small squares colored by concentration value).

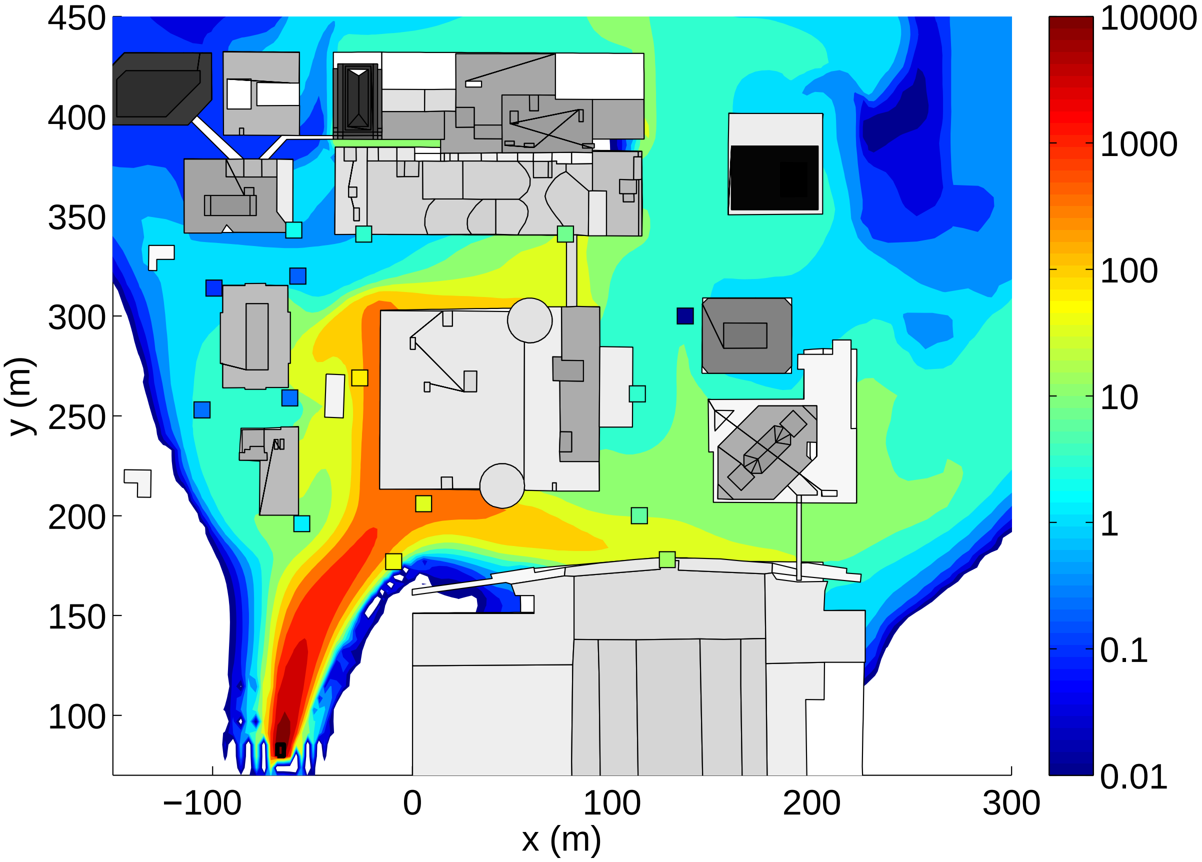


Fig. 13. As in Fig. 12 except source location and strength are from peak of reconstructed probability distribution.

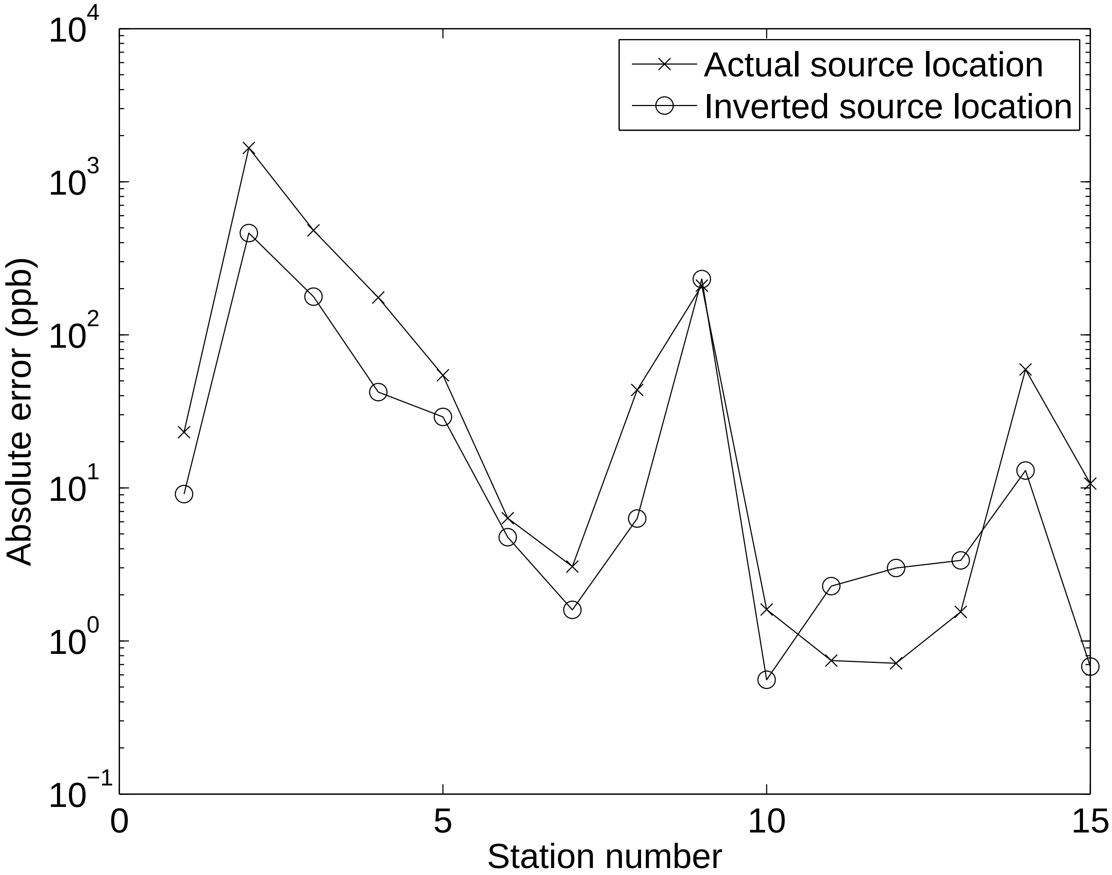


Fig. 14. Absolute error of FEM3MP predictions compared to observed concentrations at the 15 sensor locations for actual and inverted source locations.

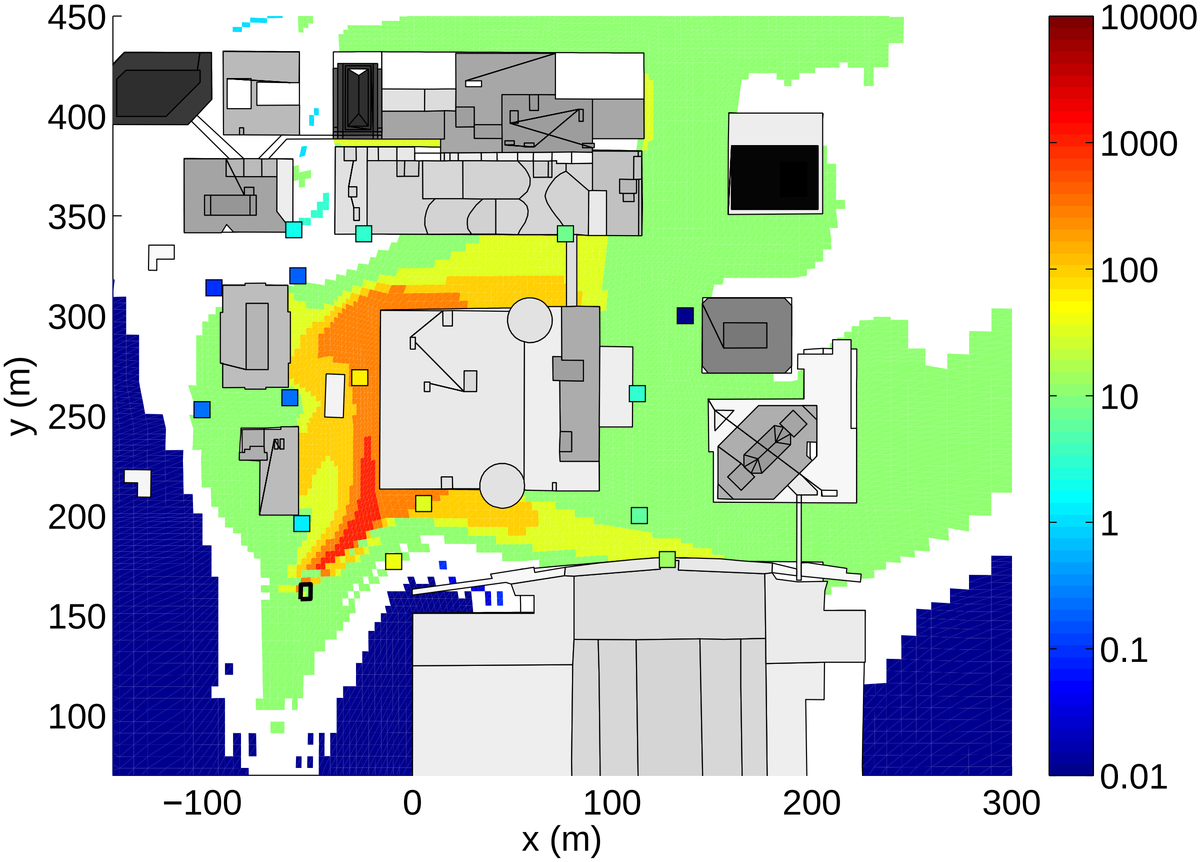


Fig. 15. Composite plume showing 90% confidence intervals for concentration levels for flow in OKC during IOP 3. Observed concentrations are also shown as small colored squares. The threshold is set at 10 ppb.

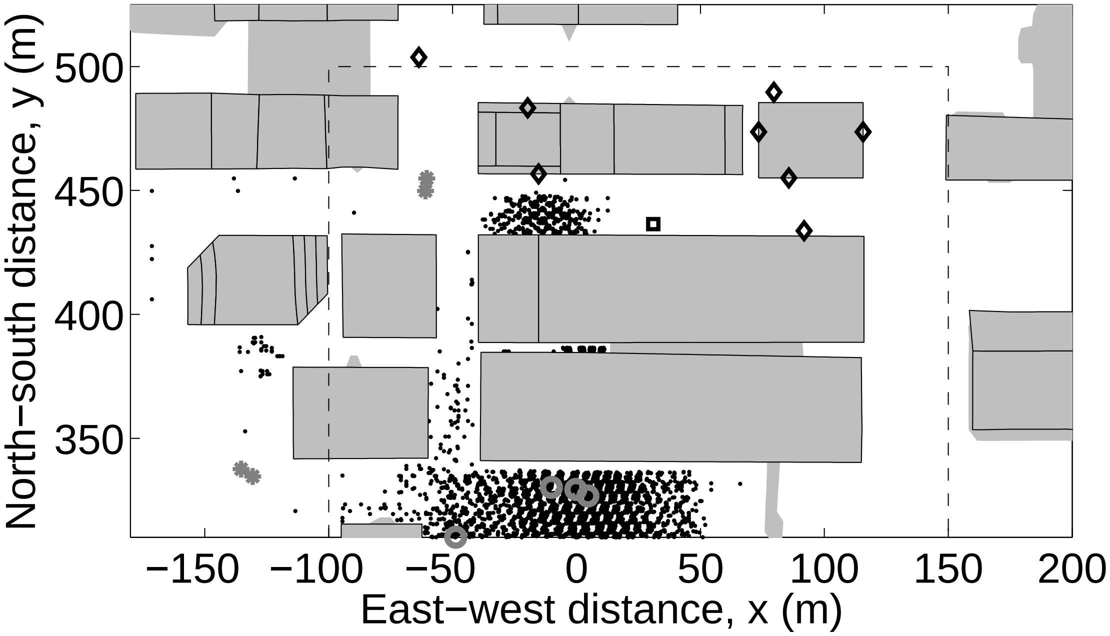


Fig. 16. Black dots show locations sampled by the four Markov chain used for source inversion for flow in Oklahoma City during IOP9. Black diamonds indicate sensor locations. Black stars show the random starting points of the Markov chains. Small black square shows true source location. Buildings that are treated explicitly are outlined in black in addition to shading; others are treated as virtual buildings. Dashed line shows zoomed-in region near the source used for later figures.

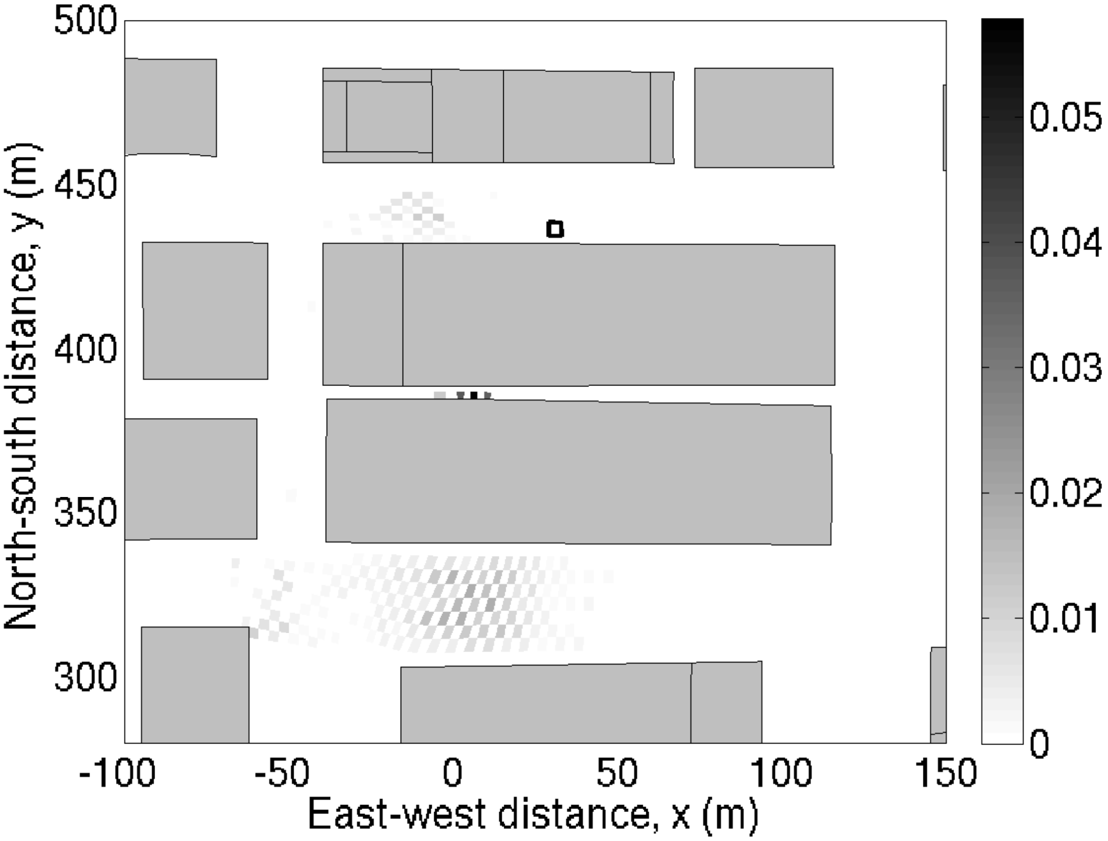


Fig. 17. Probability distribution of source location for flow in OKC during IOP 9. Only sub-domain indicated by dashed line in Fig. 16 is shown. Actual source location is shown by black square. Buildings are shaded in gray.

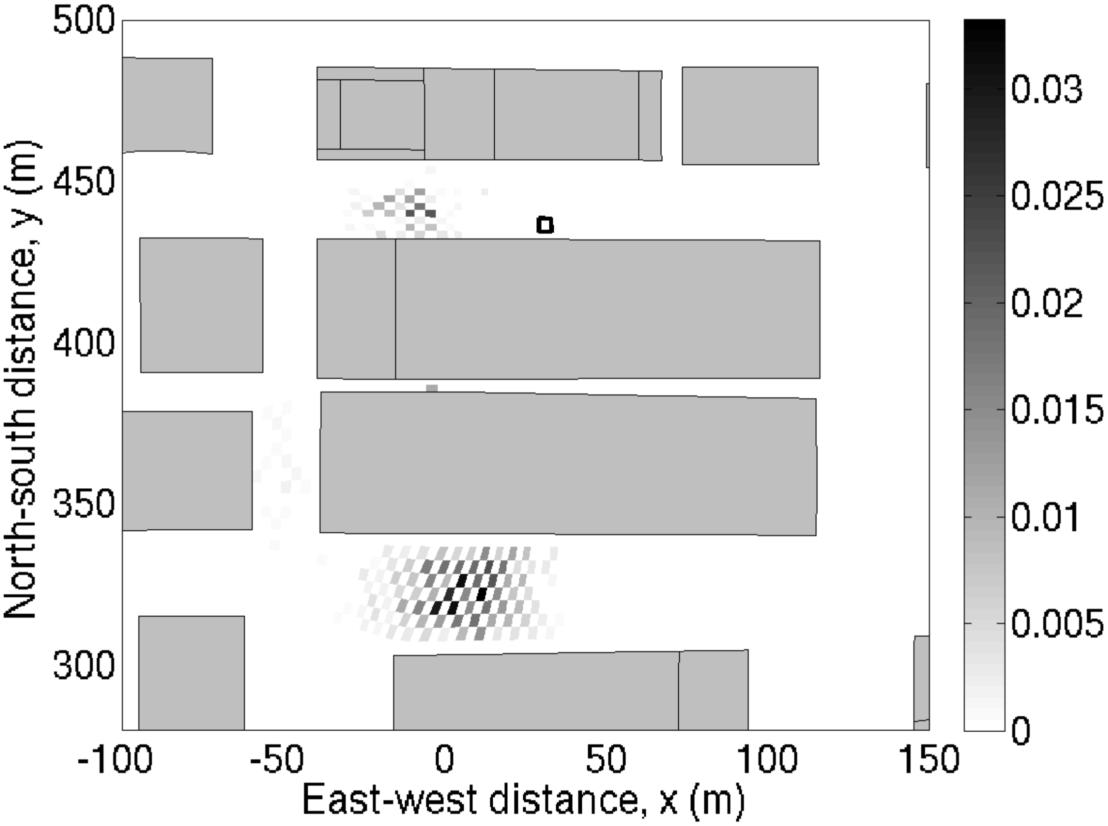


Fig. 18. Conditional probability distribution of source location for flow in OKC during IOP9 with source strength values within 50% of true value.

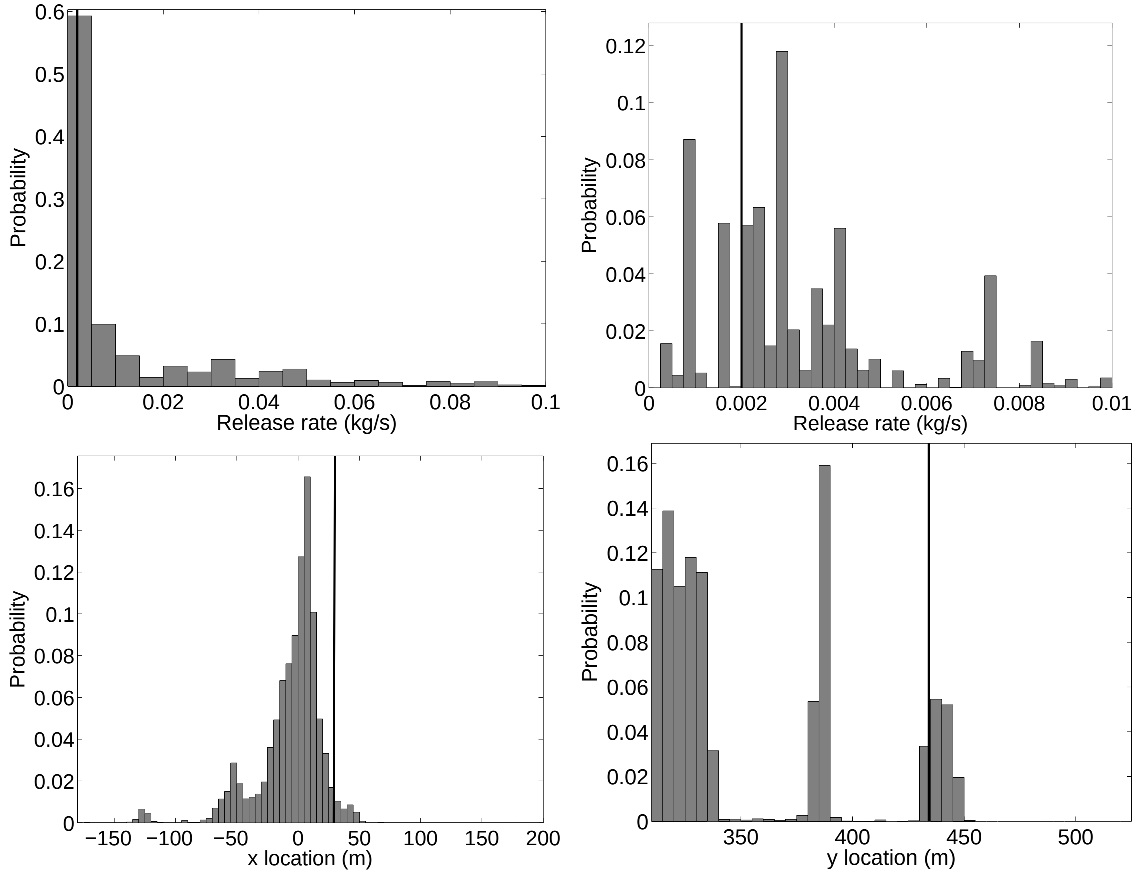


Fig. 19. Histogram of source strengths and x, y positions for flow in OKC during IOP 9. Vertical solid lines denote release rates and location coordinates of actual source.

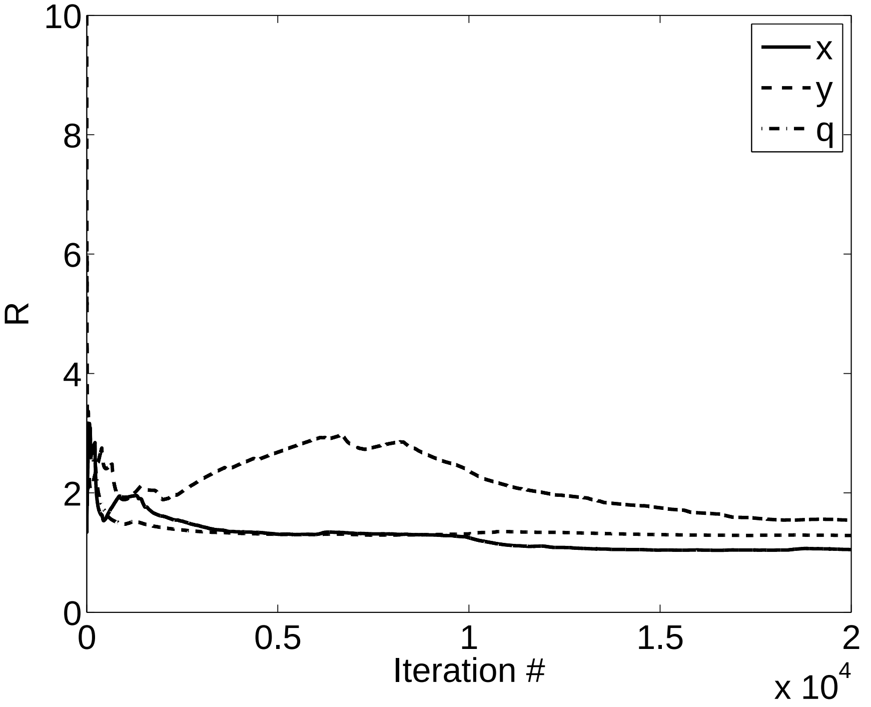


Fig. 20. Convergence rates for horizontal position (x, y) and source strength q for flow in OKC during IOP 9.

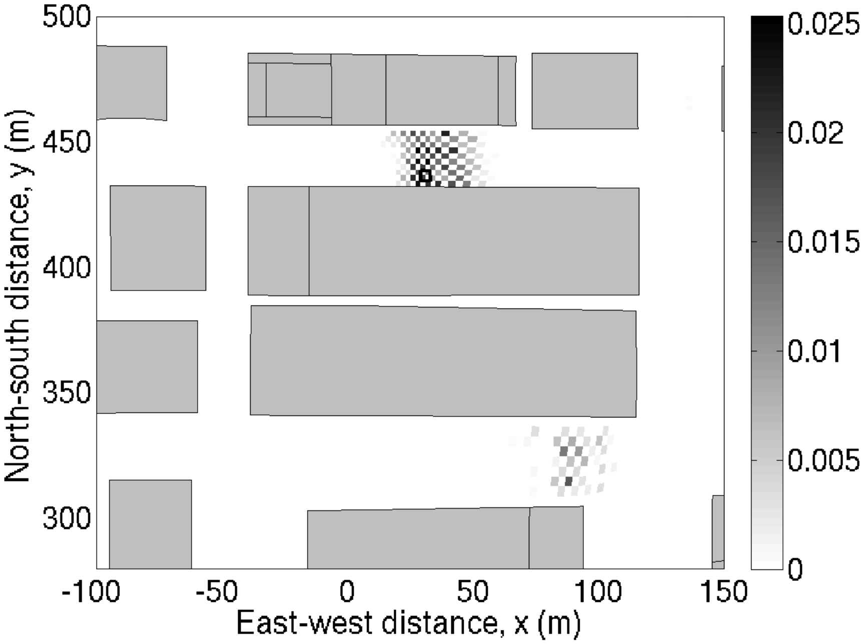


Fig. 21. Probability distribution of source location for flow in OKC during IOP 9 using synthetic sensor data generated from a forward simulation at the actual source location. As in Fig. 17.

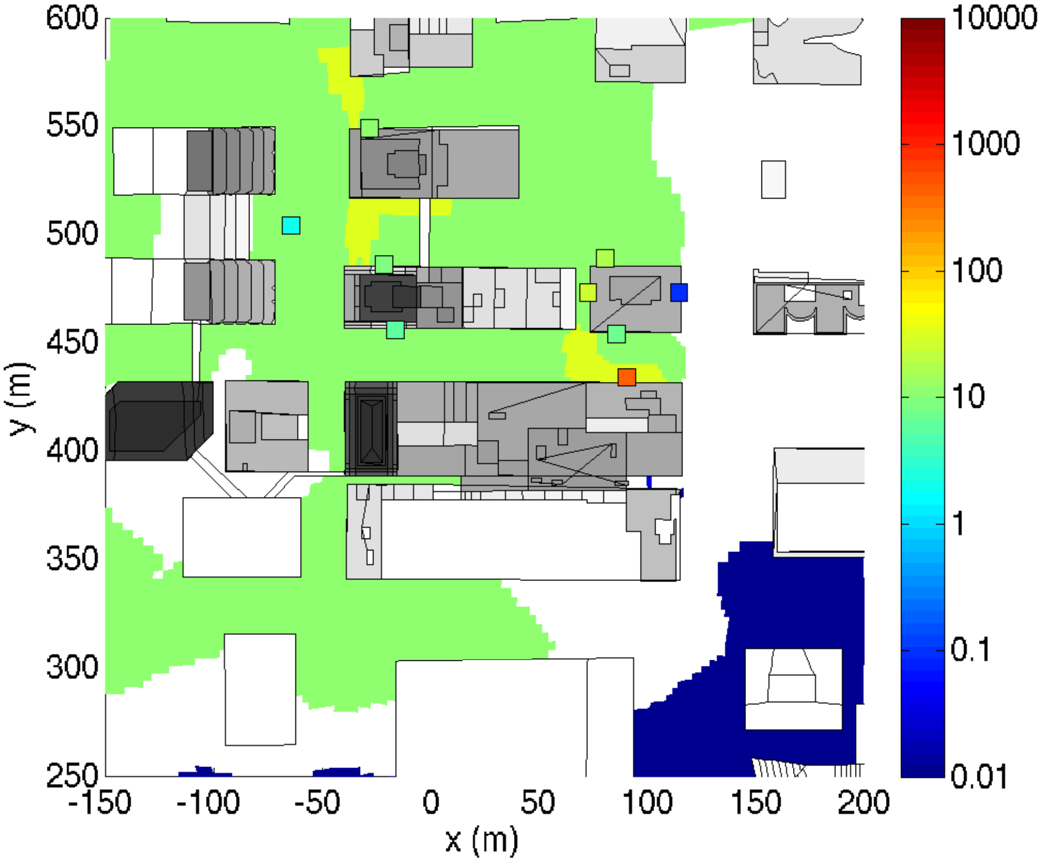
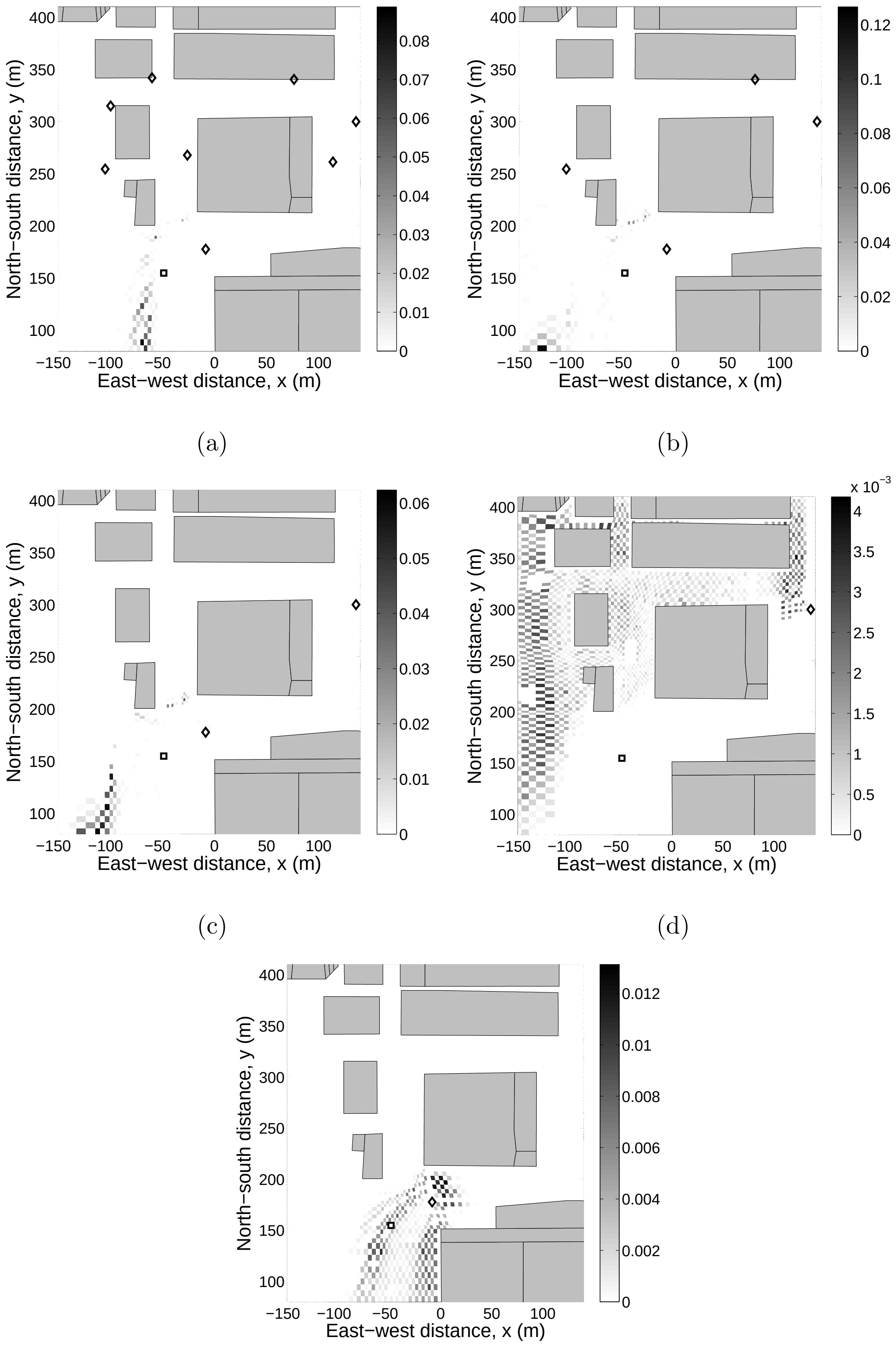


Fig. 22. Composite plume showing 90% confidence intervals for concentration levels for flow in OKC during IOP 9. Observed concentrations are also shown as small colored squares. The threshold is set at 10 ppb.



(a)

(b)

(c)

(d)

(e)

Fig. 23. Probability distribution of source location for flow in OKC during IOP 3 using 8, 4, and 2 sensors, and two different instances with 1 sensor of chosen from the original 15. Sensors chosen are shown with black diamonds. The true source is indicated by the black square.

Appendix C

Event Reconstruction for Atmospheric Releases Employing Urban Puff Model UDM with Stochastic Inversion Methodology

UCRL-PROC-216842

UCRL-PROC-216842

Event Reconstruction for Atmospheric Releases Employing Urban Puff Model UDM with Stochastic Inversion Methodology

S. Neuman, L. Glascoe, B. Kosovic, K. Dyer, W. Hanley, J. Nitao, R. Gordon

November 4, 2005

American Meteorological Society
Atlanta, GA, United States
January 29, 2005 through February 2, 2005

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

J4.6 EVENT RECONSTRUCTION FOR ATMOSPHERIC RELEASES EMPLOYING URBAN PUFF MODEL UDM WITH STOCHASTIC INVERSION METHODOLOGY

Stephanie Neuman, Lee Glascoe*, Branko Kosović, Kathy Dyer, William Hanley, John Nitao,
1. Lawrence Livermore National Laboratory, Livermore, CA 94551
and Robert Gordon
2. Riskaware, Ltd, Bristol, BS1 5BT, United Kingdom

ABSTRACT

The rapid identification of contaminant plume sources and their characteristics in urban environments can greatly enhance emergency response efforts. Source identification based on downwind concentration measurements is complicated by the presence of building obstacles that can cause flow diversion and entrainment. While high-resolution computational fluid dynamics (CFD) simulations are available for predicting plume evolution in complex urban geometries, such simulations require large computational effort. We make use of an urban puff model, the Defence Science Technology Laboratory's (Dstl) Urban Dispersion Model (UDM), which employs empirically based puff splitting techniques. UDM enables rapid urban dispersion simulations by combining traditional Gaussian puff modeling with empirically deduced mixing and entrainment approximations. Here we demonstrate the preliminary reconstruction of an atmospheric release event using stochastic sampling algorithms and Bayesian inference together with the rapid UDM urban puff model based on point measurements of concentration. We consider source inversions for both a prototype isolated building and for observations and flow conditions taken during the Joint URBAN 2003 field campaign at Oklahoma City.

The Markov Chain Monte Carlo (MCMC) stochastic sampling method is used to determine likely source term parameters and considers both measurement and forward model errors. It should be noted that the stochastic methodology is general and can be used for time-varying release rates and flow conditions as well as nonlinear dispersion problems. The results of inversion indicate the probability of a source being at a particular location with a particular release rate. Uncertainty in observed data, or lack of sufficient data, is inherently reflected in the shape and size of the probability distribution of source term parameters. Although developed and used independently, source inversion with both UDM and a finite-element CFD code can be complementary in determining proper emergency response to an urban release. Ideally, the urban puff model is used to approximate the source location and strength. The more accurate CFD model can then be used to refine the solution.

1. INTRODUCTION AND BACKGROUND

In the event of an atmospheric release, effective consequence management depends on how much is known about the release event and how quickly the problem can be analyzed to an operationally required degree of certainty. Accurate quantification of specific details of a release can greatly assist relief efforts and subsequent forensic analysis. Such quantification, rarely a straightforward task, becomes particularly complicated when the release occurs in the presence of building obstacles that can cause flow and dispersion complications. To assist the rapid analysis of atmospheric releases, the 'event reconstruction' (ER) methodology was developed to provide answers to the questions surrounding a release event: (1) what was released, (2) how much was released, and (3) when and where it occurred (Aines et al. 2002; Kosovic et al. 2005). The ER approach developed at Lawrence Livermore National Laboratory is a Bayesian inference methodology combining observed data with forward predictive models to determine unknown source characteristics. This capability can leverage from a large computational framework that supports multiple stochastic algorithms, forward models, and runs on a wide range of computational platforms. To analyze urban dispersion rapidly, the ER methodology was linked to the rapid urban puff splitting model, the UDM Version 2.2, developed by Dstl, a United Kingdom Ministry of Defence Lab located in Porton Down. For this study, the stochastic algorithm used in the Bayesian inference scheme is a Markov Chain Monte Carlo (MCMC) algorithm. All UDM and ER runs were processed for this effort using a single processor on an MS Windows operating system.

The Urban Dispersion Model (UDM) is an empirical puff model that estimates atmospheric dispersion in an urban environment by differentiating three different puff splitting regimes (open, urban, and long-range) based on empirical evidence. Different dispersion modeling procedures are applied for each regime in such a way to account for the effect of single building, building clusters, or an entire urban environment on the dispersion of Gaussian puffs (Hall et al. 2003).

In the open regime, the overall proportion of the surface covered by obstacles is less than five percent. The puffs arising in this regime travel across a largely open terrain over which single obstacles or groups of obstacles are distributed. Interaction with these obstacles changes the size and rate of travel of the puff. If the obstacle is of sufficient size in comparison to the puff, the puff will split: a portion of the material will become entrained in the wake of the building while the remainder proceeds largely unaffected. The fraction of the puff that is entrained will spread uniformly across the entrainment region and be delayed by a characteristic wake residence time. After interaction with the obstacle, puff spreading of both the unentrained puff and the entrained puff is increased due to turbulence in the recovery region.

In the urban regime, the plan area density of the obstacles is greater than five percent. The single obstacle interactions utilized in the open regime are no longer valid due to interference with multiple entrainment regions from densely distributed obstacles. Puffs quickly become large

* *Corresponding author address*: NARAC/IMAAC, L-103, Lawrence Livermore National Laboratory, Livermore, CA 94551 email: glascoe1@llnl.gov

enough to encompass obstacles resulting in a lateral dispersion that is effectively higher than the value given by traditional puff models due to puff interaction with surface obstacles. Atmospheric stratification is assumed neutral in this regime for UDM 2.2, a reasonable assumption as mechanically generated turbulence in the urban environment is likely to dominate dispersion near the ground. For the long-range regime the puff is large compared to any surface obstacles, and puffs are treated with conventional Gaussian dispersion modeling techniques.

The UDM was implemented into the existing ER framework to provide rapid results taking urban obstacles into consideration. The UDM implementation complements efforts employing the FEM3MP CFD model (see Chan et al. (2001); Chow et al. (2006)). The UDM is computationally expedient enough to run on a single processor, as a typical forward simulation with over 100 buildings requires less than a minute to complete for the most complicated case (a similar CFD run requires on the order of 100 CPU hours). Other advantages of using the UDM for rapid analysis are its relative ease-of-use with customizable buildings, source strength and location, and easily described sensor locations. Disadvantages of using a simple empirical model with ER include the fact that empirically based building-wake entrainment and detrainment methods inherent to such a model create inversion difficulties due to discontinuities. Also, the simple wind field and puff splitting techniques which allow for rapid dispersion calculations tend to lead to reduced accuracy in comparison to CFD modeling. As an example of lost detail, the UDM does not directly model channeling effects between buildings, a phenomenon typically observed in urban experiments, including the URBAN 2003 field campaign (Allwine 2004). However, depending on source location relative to important building obstacles, puff entrainment and detrainment can provide some compensation for the lack of channeling effects (Figure 1). The Oklahoma City example discussed below demonstrates the problems that this can cause in the final source characterization.

Two example ER scenarios using the UDM are discussed below. The first scenario is for simple flow around a cubic building; the second scenario is a release in downtown Oklahoma City for observations and flow conditions during the Joint URBAN 2003 field campaign. In both simulations, the event reconstruction code simultaneously samples both source location and source strength. In the UDM 2.2, source strength is represented by total mass released, and results of probable source strength are presented in this way.

2. RECONSTRUCTION METHODOLOGY

The ER framework for this study performs stochastic inversion using MCMC techniques (see, for example, Gelman et al. (2003)). The procedure is as follows: 1. estimates of source location and source strength are obtained from a defined prior distribution or proposal distribution of source term parameters; 2. the forward model (UDM) is run using these input values; 3. the output sensor data from the forward model is compared to the observed data using Bayes theorem; 4. the sampled source term configuration is either accepted or rejected following a Metropolis-Hastings algorithm; 5a. if accepted, the likelihood function is updated and the values used in the next iteration are sampled from the proposal distribution centered on the accepted value; 5b. if rejected, the next point is selected based on the last accepted value; 6. this process is repeated for a large number of iterations until the convergence to a posterior probability distribution of source term parameters (representing the solution to the inverse problem) is achieved. Effective reconstruction using Bayesian inference via stochastic sampling requires model and data error quantification. A single log-normal standard deviation distribution characterized by a single input parameter is used to represent both uncertainty in the sensor measurements and uncertainty in the forward model. The higher the input value of error, the broader the resulting probability distribution will be. More details on this methodology can be found in Johannesson et al. (2004) and in proceedings paper **J4.4** (Chow et al. 2006).

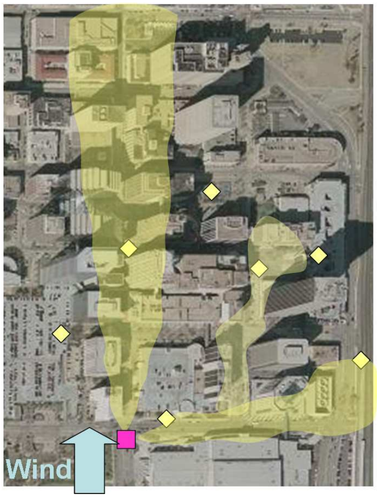


Figure 1: Given a highly complex domain, with buildings of various shapes and sizes, and concentration measurements at a few locations, is it possible to find the source of a contaminant plume with a fast urban puff model?

3. ISOLATED BUILDING

The first test of integrating UDM with the ER methodology is a simple cubic building, 10m to a side and is a follow-on study to the ‘Isolated Building’ of paper **J4.4** in these proceedings (Chow et al. 2006). Figure 2 shows a forward simulation using the UDM. The entrainment region is clearly visible in the figure. Also, the intentional slight asymmetry of the source location can be seen in the resulting plume. The ER was performed in comparison to synthetic data generated by the UDM for an ‘actual’ source location.

The resulting Markov chains for the source inversion are shown in Figure 3. The asterisks mark the initial location of each of the four chains. The diamonds represent the four sensors, and the actual source is shown as a magenta square. After some exploration of the domain space, the chains quickly converge to the area immediately surrounding the actual source location. Note that two of the Markov chains explored the entrainment re-

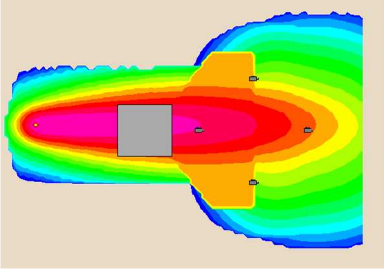


Figure 2: Horizontal concentration contours at the first vertical level generated by a UDM forward simulation for flow around an isolated cubic building. Four sensors are placed in the lee of the building.

gion. This result reflects how puffs arising in a building entrainment region are automatically fully entrained and how the detrainment process simulates a source. However, the resulting probability distribution, Figure 4, shows that the number of samples that the Markov chains sampled from the entrainment region is negligible compared to the number of samples in the vicinity of the actual source. Note the peak of the probability distribution is close to the actual source location.

In addition to the source location, release strength was stochastically sampled during this simulation. The resulting release strengths are displayed in a histogram in Figure 5. The distribution of total mass has a single, significant peak in very good agreement with the actual value, shown as a solid vertical line. When model predictions are compared to synthetic data, as in this example, the source inversion calculation is very accurate. However, to conduct source inversion for actual events, the model must be able to predict source characteristics using real data. Due to random and systematic differences between sensor measurements and model predictions, we expect that event reconstruction will be less accurate in this case.

4. OKLAHOMA CITY - JOINT URBAN 2003 IOP3

Given a highly complex domain, with buildings of various shapes and sizes, and concentration measurements at a few locations, the possibility of locating the source of a contaminant plume and determining its characteristics using a fast Gaussian puff model is of great interest (Figure 1). Event reconstruction with the UDM was applied to Oklahoma City in order to compare the model output to observations from the Joint URBAN 2003 field campaign. A standard shape file of downtown Oklahoma City was used to construct the buildings. Actual source and sensor locations were used to recreate the field experiment. An event reconstruction calculation was conducted using concentration measurements from the Intensive Observational Period 3 (IOP3) from the Joint Urban 2003 tracer field experiment in Oklahoma City with a southerly wind input. A UDM 3D puff simulation and the downtown of Oklahoma City is illustrated in Figure 6. During our simulations, one large building in the south of the modeled domain was found to play a key role. The entrainment region of this building will be shown to adjust for some deficiencies of the forward model, specifically the lack of

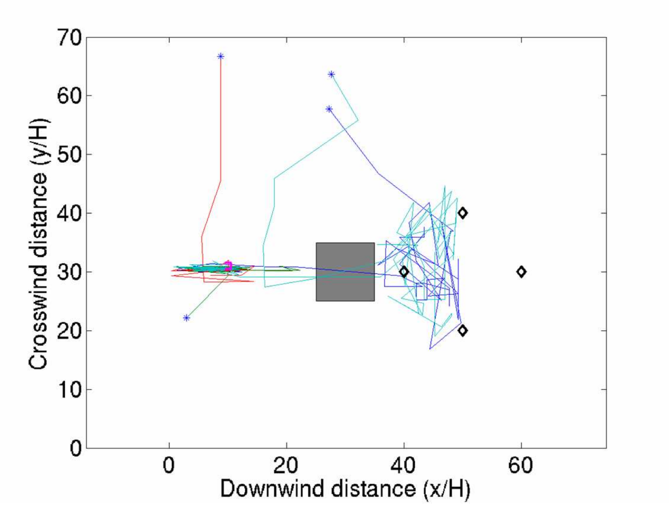


Figure 3: Paths of four Markov chains for flow around an isolated cubic building. Note that the magenta square indicates the source and the black diamonds indicate the four sensors.

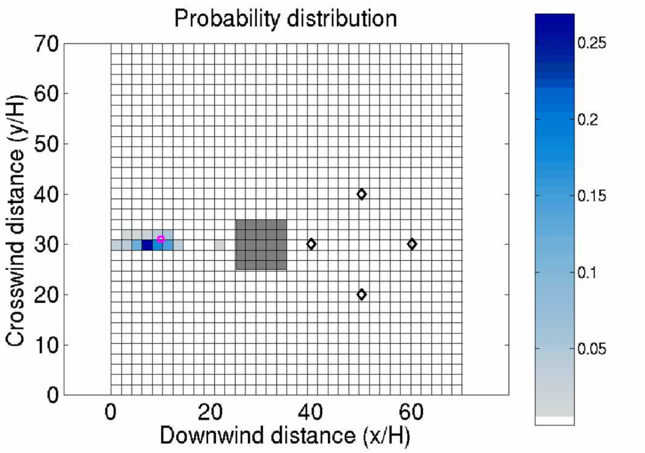


Figure 4: Probability distribution of source location for flow around an isolated cubic building.

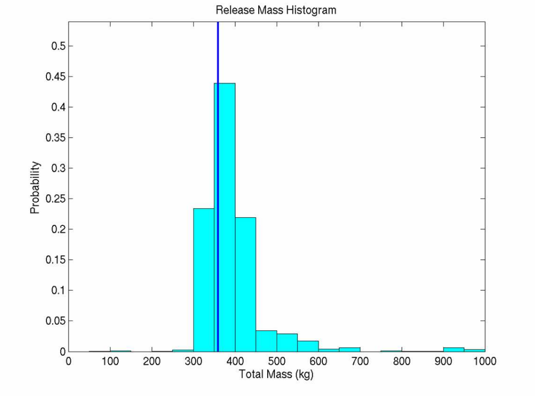


Figure 5: Histogram of source strengths for flow around an isolated cubic building. Vertical blue line indicates actual release rate.

channeling effects.

Puffs and 2D contours of ground-level concentration are displayed in Figures 6 and 7, respectively. Wind speed was 6.5m/s at 50m above ground. The number of iterations is 1700 and each of those iterations involved four Markov chains. The complete calculation took less than 33 hours on a single 857 MHz processor. Scaling linearly, if eight Markov chains are distributed to eight separate 857 MHz processors, the entire calculation, could be completed in approximately one hour.

The resulting Markov chains are illustrated in Figure 8. Note how the chains quickly converge to south of the domain. While there is good mixing by three of the chains, one chain becomes stuck in a local minimum, and remains at the northwest corner of the building. The resulting probability distribution is shown in Figure 9. There are three distinct peaks visible in the distribution. One peak is within 20m of the actual source location, which is shown as a triangle. Another peak is towards the bottom of the domain, and the third near the large building, part way between the other two locations. Three peaks are also noted in the release strength histogram, Figure 10. One peak is a very low value of release mass. The second, smallest, peak corresponds with the actual release mass, shown as a solid vertical line. The final peak is a higher value, between 8 kg and 9 kg of total mass released during the simulation.

In order to determine the probable locations that corresponded to each of the three most likely release rates, conditional probability for each was computed. Figure 11 illustrates the conditional probability of source location depending on release mass (low, mid and high) and Figure 12 illustrates the relatively rapid convergence on source location as opposed to source strength. The low peak, less than 1 kg, corresponded to the location very near the actual source location. The resulting probabilities for both location and strength are about 25%, indicating that one of the four chains spent much of its time in that location without being able to further explore the domain. This is confirmed by examining the details of the Markov chains: one chain spends the simulation in that location. The release strength is low because of the close proximity to the sensors.

The conditional probability corresponding to the actual mass, 3.1 kg < q < 4.1 kg, peaks toward the bottom of the domain, almost 200m south of the actual source location. When the source material is released in the model from the actual source location, the puffs are too narrow to hit the sensors channeling northeast of the source. When the source is located at the peak of the middle plot of Figure 11, the increased distance to the sensors and the interaction with the large building sufficiently enlarge the puff to better agree with the actual concentrations. The conditional probability corresponding to the highest release strength, 8.25kg < q < 9.25kg, is shown in the far-right plot of Figure 11. Due to its proximity to the large building, material released from this point is automatically entrained in the building's wake. Here, the entrainment region acts as a source, releasing material from the entrainment region over time. The entrainment creates a large, diffuse puff in the wake of the building, resulting in predicted source strengths for this location that are higher than the actual value.

5. DISCUSSION AND CONCLUSIONS

Event reconstruction calculations using the Urban Dispersion Model, UDM, can be performed very rapidly to provide a valid initial approximation for source location and release strength even in a complex urban environment. As an emergency response tool, event reconstruction with the UDM is more applicable than a CFD equivalent because of the speed at which a complete calculation can be completed. Ideally, results obtained from reconstruction with UDM can be used to significantly decrease the sampling domain needed to perform more accurate CFD calculations. That is, using independent data, posterior distributions obtained using ER with the UDM can be used as a prior distribution for ER with a CFD code. With a smaller domain, those subsequent calculations can be conducted much more expediently.

When conducting a source inversion calculation using the UDM as a forward model, it is important to have all Markov chains exploring the domain space in order to predict accurate source probability distributions. In order to obtain sufficient mixing, input parameters such as step size in x and y, step size in q, and quantified error should be specified carefully with attention to appropriate values relevant to the scale of the problem. Determining the correct input values for these parameters can take some trial and error. As illustrated in the Oklahoma City example, one of the main sources of error in the posterior probability distributions for complicated city examples is the lack of channeling effects in the forward model UDM. The wind field applied by UDM is very simplified and cannot reproduce complex urban flows beyond building entrainment. Channeling effects are somewhat compensated for by building entrainment effects, but the results of the reconstruction consequently may not reflect the actual source location.

The next step in this research is to test the UDM with a larger domain space and with more sensor data. Sensor data for the Oklahoma City example exists up to 4 km from the source. It is anticipated that with an extended domain, the lack of channeling producing error over the short range will have less impact on the results. Also, stochastic sampling of wind direction as well as source location and strength may give better results.

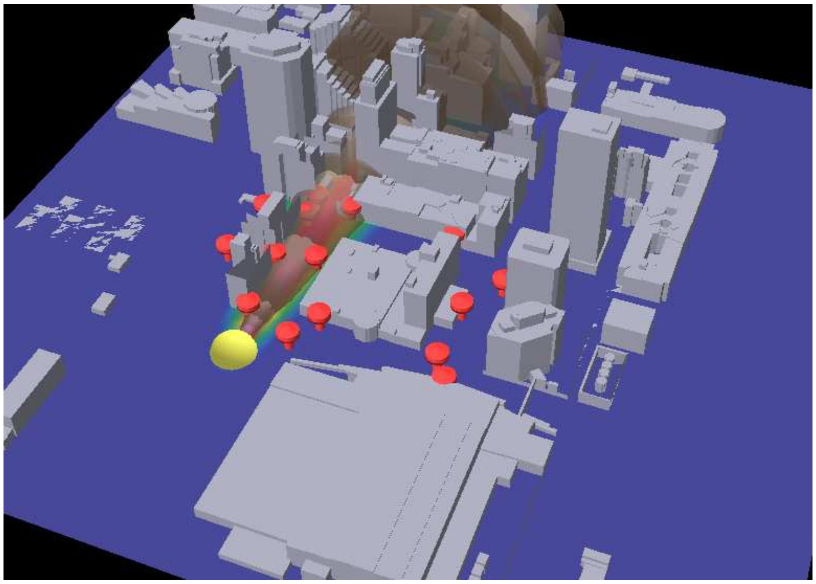


Figure 6: Three-dimensional puffs generated by a forward simulation with the UDM 2.2 for flow in and around the downtown business district of Oklahoma City. Note how the puffs expand and entrain behind the larger buildings.

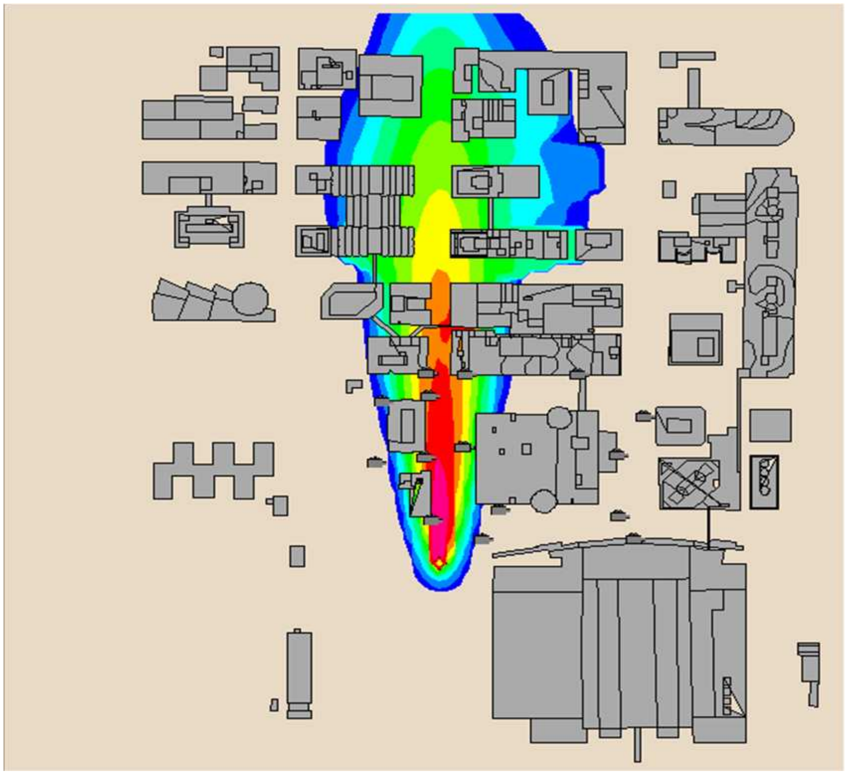


Figure 7: Horizontal concentration contours at the first vertical level generated by a single forward simulation with the UDM 2.2 for flow in and around the downtown business district of Oklahoma City.

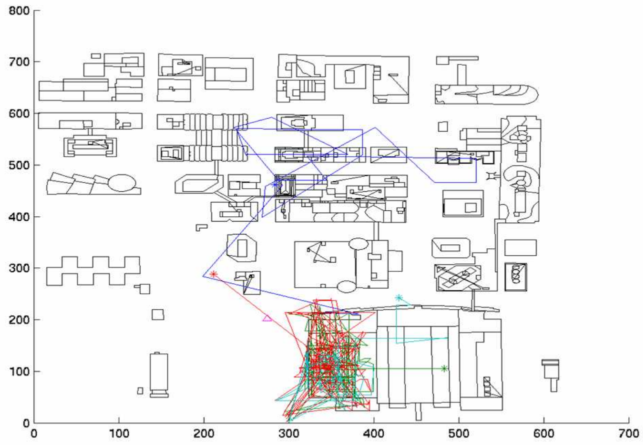


Figure 8: Paths of four Markov chains for flow in and around the downtown business district of Oklahoma City.

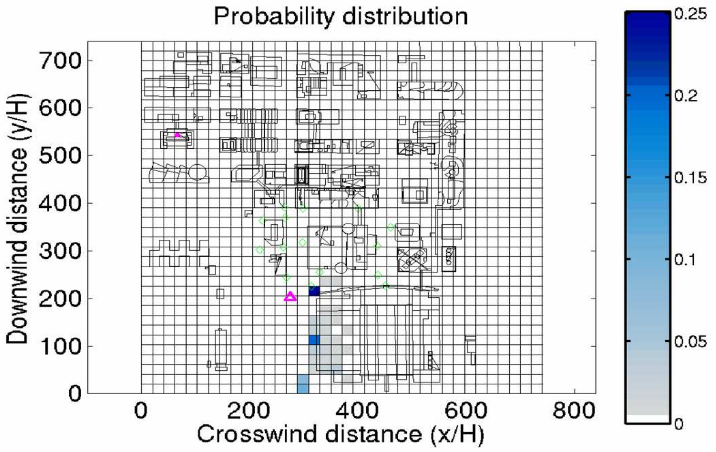


Figure 9: Probability distribution of source location for flow in and around the downtown business district of Oklahoma City. The magenta delta indicates actual release location.

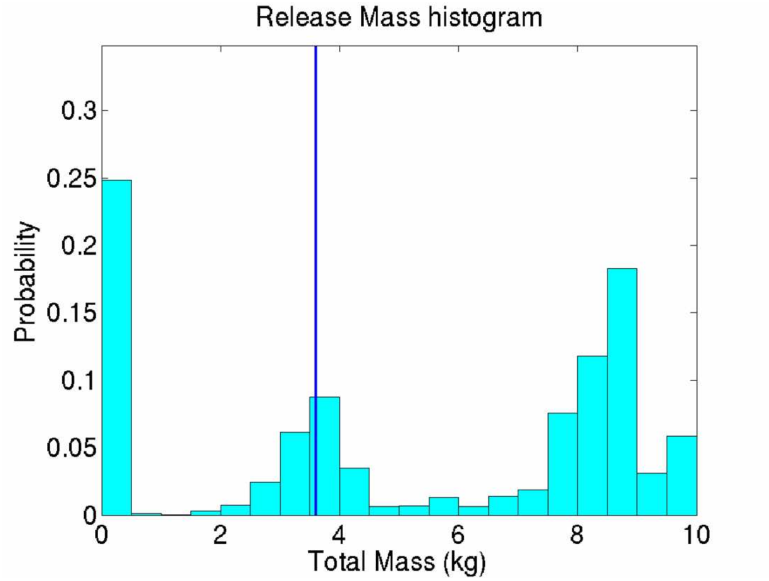


Figure 10: Histogram of source strengths, q, and the conditional probabilities for flow in and around the downtown business district of Oklahoma City. Vertical blue line indicates actual release rate.

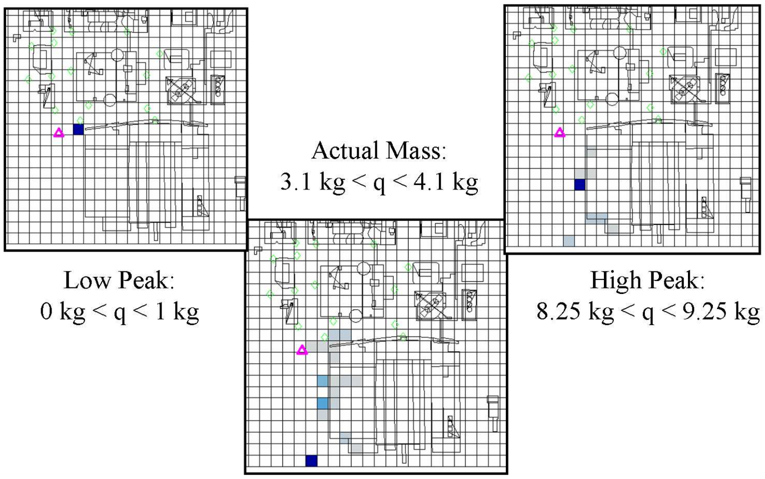


Figure 11: Conditional probability distribution of source location for flow in and around the downtown business district of Oklahoma City. The magenta delta indicates actual release location.

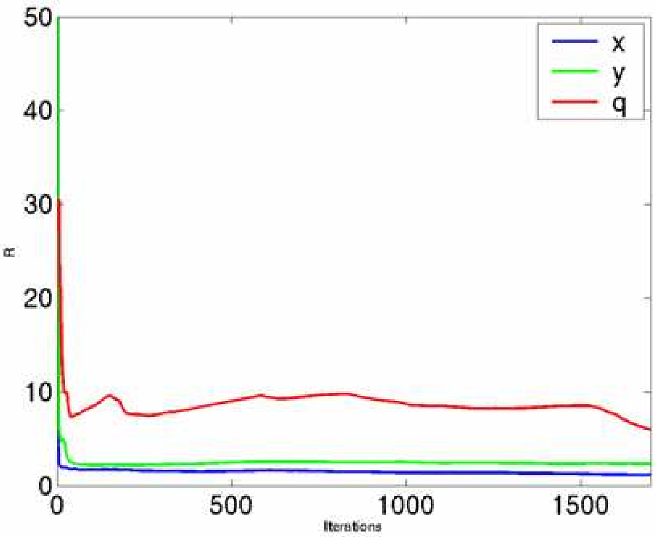


Figure 12: Convergence of x and y location, and slower conversion of strength q for the Oklahoma City example.

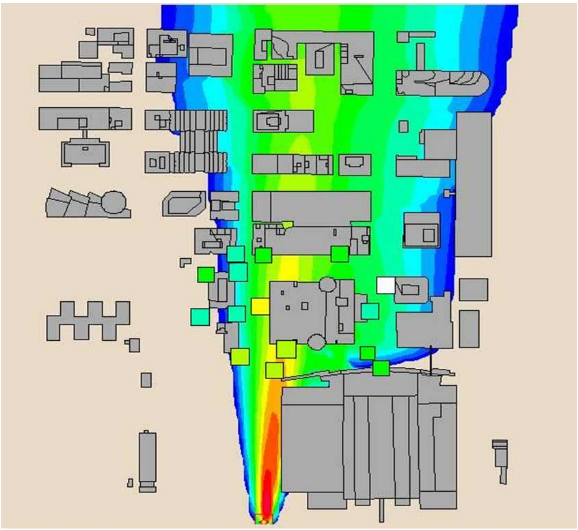


Figure 13: The building wake entrainment acts as a flow channeling effect within the UDM. Horizontal concentration contours at the first vertical level generated by forward simulation with UDM for flow in and around the downtown of Oklahoma City for the location associated with actual source strength.

6. ACKNOWLEDGMENTS

This work was funded by LDRD project number 04-ERD-037 and was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48. This paper is numbered UCRL-PROC-216842.

REFERENCES

Aines, R., J. Nitao, R. Newmark, S. Carle, A. Ramirez, and W. Hanley, 2002: The Stochastic Engine Initiative: Improving Prediction of Behavior in Geologic Environments We Cannot Directly Observe Publication. Technical Report UCRL-ID-148221, Lawrence Livermore National Laboratory.

Allwine, K., 2004: Joint Urban 2003 field study and urban mesonets. *Fourth Symposium on Planning, Nowcasting, and Forecasting in the Urban Zone*, American Meteorological Society.

Chan, S., D. Stevens, and W. Smith, 2001: Validation of CFD urban dispersion models using high resolution wind tunnel data. **107**.

Chow, F., B. Kosovic, and S. Chan, 2006: Source inversion for contaminant plume dispersion in urban environments using building-resolving simulations. *Annual American Meteorological Society Meeting, Atlanta*, S. Hanna, ed.

Gelman, A., J. Carlin, H. Stern, and D. Rubin, 2003: *Bayesian Data Analysis*. Chapman and Hall, CRC.

Hall, D., A. Spanton, I. Griffiths, M. Hargrave, and S. Walker, 2003: The Urban Dispersion Model (UDM): Version 2.2 technical documentation release 1.1. Technical Report DSTL/TR04774, Porton Down, UK.

Johannesson, G., W. Hanley, and J. Nitao, 2004: Dynamic Bayesian models via Monte Carlo - an introduction with examples. Technical Report UCRL-PRES-207173, Lawrence Livermore National Laboratory, Livermore, CA.

Kosovic, B., G. Sugiyama, S. Chan, T. Chow, K. Dyer, L. Glascoe, R. Glaser, W. Hanley, G. Johannesson, S. Larsen, G. Loosmore, J. Lundquist, A. Mirin, S. Neuman, J. Nitao, R. Serban, and C. Tong, 2005: Dynamic data-driven event reconstruction for atmospheric releases. *Aerosols and Safety, 2005 Conference*, Obninsk, Russia, number UCRL-PRES-215766.

Appendix D

Synthetic Event Reconstruction Experiments for Defining Sensor Network Characteristics

UCRL-TR-217762

UCRL-TR-217762

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Synthetic Event Reconstruction Experiments for Defining Sensor Network Characteristics

J. K. Lundquist, B. Kosovic, R. Belles

December 16, 2005

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

Synthetic Event Reconstruction Experiments for Defining Sensor Network Characteristics

Julie K. Lundquist, Branko Kosović, and Rich Belles

Abstract

An event reconstruction technology system has been designed and implemented at Lawrence Livermore National Laboratory (LLNL). This system integrates sensor observations, which may be sparse and/or conflicting, with transport and dispersion models via Bayesian stochastic sampling methodologies to characterize the sources of atmospheric releases of hazardous materials. We demonstrate the application of this event reconstruction technology system to designing sensor networks for detecting and responding to atmospheric releases of hazardous materials. The quantitative measure of the reduction in uncertainty, or benefit of a given network, can be utilized by policy makers to determine the cost/benefit of certain networks.

Herein we present two numerical experiments demonstrating the utility of the event reconstruction methodology for sensor network design. In the first set of experiments, only the time resolution of the sensors varies between three candidate networks. The most “expensive” sensor network offers few advantages over the moderately-priced network for reconstructing the release examined here. The second set of experiments explores the significance of the sensors’ detection limit, which can have a significant impact on sensor cost. In this experiment, the expensive network can most clearly define the source location and source release rate. The other networks provide data insufficient for distinguishing between two possible clusters of source locations. When the reconstructions from all networks are aggregated into a composite plume, a decision-maker can distinguish the utility of the expensive sensor network.

1. Introduction

To detect the atmospheric transport of hazardous materials, new and innovative sensor networks are currently being designed and deployed. These networks can serve one or more of several purposes: they can detect the spread of hazardous materials before large populations have been exposed so that emergency response officials can organize evacuations; they can identify the size of the release so that officials can respond during an event with evacuations or inoculations; they can be used in a forensic role, post-event, to describe the release so that decontamination efforts can be prescribed.

Many types of sensors and sensor networks for detecting atmospheric releases of hazardous materias have been designed; some networks have been deployed, such as the BioWatch network that is designed to provide early warning in the case of a mass pathogen release (Shea and Lister, 2003). These networks have varying degrees of detection sensitivities, false-alarm rates, and frequency of data collection. However, the utility of these networks in characterizing the sources of atmospheric releases of hazardous materials has not yet been demonstrated systematically to our knowledge.

Reconstructing the source of a detected atmospheric release is a crucial step in predicting the consequences of such a release. The primary source of uncertainty in prediction the consequence of an atmospheric release is determining the source term characteristics, such as location, magnitude, and duration of the release. The type of source is also a crucial component in determining consequences: sources may be instantaneous (like an explosion) or continuous (a long-term release), localized to one point or over a wide area, static or moving, at the surface or elevated. Even if the source is perfectly characterized, the complexity of atmospheric flow, especially in urban environments or in complex terrain, presents additional challenges for a dispersion model.

An event reconstruction technology system has been designed and implemented at Lawrence Livermore National Laboratory (LLNL). This system integrates sensor observations, which may be sparse and/or conflicting, with transport and dispersion models via Bayesian stochastic sampling methodologies to characterize the sources of atmospheric releases of hazardous materials. The event reconstruction methodology identifies source characteristics (such as location, magnitude, duration) that are most consistent with the observed data, given a quantification of the errors expected in the both the observations and in the forward dispersion model. Once the source is characterized, an ultimate prediction of likely affected areas is possible. This ultimate prediction is a composite of all the likely sources and can guide emergency responders more effectively than a single forward prediction from a single (possibly incorrect) estimate of source characteristics. The composite prediction provides a measure of uncertainty in source characterization and the result of the release of hazardous material.

Ideally, the observations describing an event would provide enough information about an event so that the uncertainty regarding the source location or magnitude is very low. To ensure this minimal uncertainty, sensor networks must be designed with that goal in mind. It is possible, using the event reconstruction system, to examine certain scenarios of interest using different sensor networks to determine which sensor network(s) will provide the greatest reduction of uncertainty in source characterization and response. This quantitative reduction in uncertainty can be provided to policy makers to determine the cost/benefit of certain networks. Questions such as the following can be addressed:

- Would a network consisting of fewer instruments that are more sensitive protect my facility better than a network with more instruments that are less sensitive?
- If I have time constraints on my response to the detection of a release, how often must I collect information from my sensors?
- To reduce costs while still protecting my city, is a dense network of instruments with a high false alarm rate a better choice than a sparse network of more reliable instruments?

To demonstrate the utility of the event reconstruction system for answering questions like these, we present a pair of numerical experiments designed to determine and quantify the advantages of using sensors of varying time resolutions and varying detection limits for identifying a source of a hazardous release that affects a suburban domain of size 6 km by 6 km. This experiment quantifies the importance of time resolution and sensor detection limit thresholds in otherwise identical instruments deployed to identify the source of a 1.5-hour-long release of a neutrally-buoyant gas in the suburban area. Although these experiments were loosely based on an actual atmospheric tracer experiment (the Copenhagen release, Grying and Lyck, 1984), such experiments using the LLNL event reconstruction methodology require only a definition of the region of interest, a climatology of atmospheric conditions for that region, some specification of the types of releases required to be considered, a transport-and-dispersion model suitable for simulating the dispersion of the material(s) of interest, and a measure of error for the sensor observations and the transport-and-dispersion model predictions.

2. Description of the event reconstruction methodology

LLNL's event reconstruction methodology (Kosovic et al., 2005) integrates Bayesian stochastic methodologies with "forward" atmospheric transport and dispersion models and observations of atmospheric concentrations due to an atmospheric release to determine unknown source parameters. The event reconstruction system can provide optimal characterization of unknown source term parameters, given a set of measurements of atmospheric concentrations M_{ij}

at locations i and times j, an atmospheric dispersion model that predicts concentrations C_{ij} at locations i and times j as a function of source term parameters, and quantification of the error in both model predictions of concentrations C and observed measurements M.

For example, event reconstruction is often used to determine probabilistic estimates of two unknown source terms parameters, source location X and source magnitude R. This final probabilistic estimate is known as the posterior distribution, which is calculated over many iterations in the following way. At an nth iteration of the reconstruction, a Markov chain samples unknown source term parameters Xn and Rn from a large set of possibilities $\boldsymbol{X}$ and $\boldsymbol{R}$. These source parameters Xn and Rn are provided to an atmospheric transport and dispersion model, which uses Xn and Rn to predict atmospheric concentrations at sensor locations i and times j. The measurements M_{ij} and the model predictions $C_{ij(XnRn)}$ are compared; details of that comparison are discussed below. Based on that comparison, the probability of source location Xn and source magnitude Rn are evaluated via comparison to previous guesses $Xn\text{-}1$, $Xn\text{-}2$, … and $Rn\text{-}1$, $Rn\text{-}2$, …. If the comparison is favorable for Xn and Rn, their values are retained for subsequent comparisons of $Xn+1$ and $Rn+1$. Eventually, convergence to a final posterior distribution is attained, and that final posterior distribution summarizes the most likely of source term parameters X and R given measurements and their error, prior knowledge about the characteristics of the source, and prior estimates of the transport and dispersion model error.

This process, known as Markov Chain Monte Carlo sampling for Bayesian inference, is discussed in detail in popular texts such as Robert and Casella (2005) and Liu (2001), e.g.. The sampling procedure used herein relies on a Metropolis-Hastings algorithm for generated samples Xn and Rn from a domain of possibilities $\boldsymbol{X}$ and $\boldsymbol{R}$. Multiple Markov Chains can proceed through the domains $\boldsymbol{X}$ and $\boldsymbol{R}$ simultaneously; four Markov Chains are used in the reconstructions presented here.

a. The “forward” atmospheric transport and dispersion model

A core component of an atmospheric event reconstruction is the efficient use of an atmospheric dispersion model. The present methodology has been used with a wide array of transport and dispersion models. These models include a relatively simple and fast 2D Gaussian puff model INPUFF (Petersen and Lavdas, 1986), a Lagrangian particle dispersion model LODI which is used at LLNL’s National Atmospheric Release Advisory Center (NARAC) (Ermak and Nasstrom, 2000, Larson and Nasstrom, 2001), and a building-resolving computational fluid dynamics code FEM3MP (Chan and Stevens, 2000; Chow et al., 2006). In each case, thousands of possible source term parameters Xn and Rn, are provided to the forward dispersion model, meaning that thousands of forward simulations are carried out. Computational efficiencies such as a Green’s function approach (Chow et al. 2006) are available to reduce the number of forward simulations, but have not been employed for the study presented herein.

The National Atmospheric Release Advisory Center's Lagrangian particle dispersion model, LODI, provided the forward atmospheric transport and dispersion simulations for this experiment. Based on a given source location and release rate, LODI generates a number of Lagrangian particles that disperse within its simulation domain based on meteorological and turbulence parameters calculated by LODI and provided to it by a meteorological data assimilation model, ADAPT (Sugiyama and Chan 1998), also developed at the National Atmospheric Release Advisory Center.

b. The likelihood function and assumed error

The quality of the reconstruction, or the precision of the final posterior distribution of the unknown source term parameters, is related to the error assumed or known in both the actual measurements used in the reconstruction and the forward model (as well as to the quality of the data used in the reconstruction). For the reconstructions discussed here, these two errors are incorporated into one error parameter, σ, which is utilized in the comparison described above.

The measurements M_{ij} and modeled concentrations C_{ij} are first compared to the detection range of the instruments used. Any measurements or modeled concentrations above the saturation level of the instrument are set to the saturation level; any measurements or modeled concentrations below the detection limit are set to the detection limit or sensor sensitivity threshold. The natural log of the likelihood function L for source Xn and Rn over all the sensor measurements N is a function of the difference between the measurements and the modeled concentrations assuming source parameters Xn and Rn:

$$\ln\left(L_{XnRn}\right) = \frac{\sum_{ij}\left(M_{ij} - C_{ij}\right)^2}{2N\sigma^2}$$

This likelihood value L_{XnRn} is compared to that of previously-tested values such as $L_{Xn-1Rn-1}$. Other likelihood functions are possible. Generally, values of source term parameters that lead to smaller values of L are retained, although some violations of that rule are allowed to ensure wide sampling of the source term domains $\boldsymbol{X}$ and $\boldsymbol{R}$ and to prevent any Markov chain from being caught in a local minimum of L. Successful likelihood values contribute to the final posterior distribution.

If larger errors in measurements or modeled concentrations are appropriate, larger values of σ should be assumed. For all reconstructions presented herein, σ=0.2. Larger values of the error range, or s, will generally lead to broader final posterior distributions. Possible bias in the measurements or in the model is not accounted for in the formulation of the likelihood function as presented here, but may be incorporated into the likelihood function.

3. Description of the numerical simulations

These simulations were roughly based on the Copenhagen tracer experiments (Gryning and Lyck, 1984), using a domain, terrain features, and meteorology from the 19 July 1979 release of sulfur hexafluoride tracer gas in the suburban Copenhagen area. Hourly averages of wind speed and wind direction from the TV tower from which the tracer gas was released were used to define a wind field in NARAC's ADAPT model; observations from four levels above the surface were available (10m, 60m, 120m, and 200m). Boundary-layer height (2090m) was estimated from a sounding released within 10km of the source. Friction velocity (0.77 m/s) was estimated from the mean wind profile. Surface roughness for the suburban domain was estimated at 0.6m. Although the meteorology was prescribed for this event reconstruction, the characteristics of meteorology could also be included in the set of unknowns that the event reconstruction system seeks to identify. Wind speed and wind direction profiles for hours 1000, 1100, and 1200 UTC (local time – 1 hour) appear in Figure 1. Note that wind direction was reported only at 10m, 120m, and 200m levels.

Two numerical studies are presented herein. The first study explores the role of time resolution of the instruments used in this study. The second study explores the role of sensor sensitivity or detection limit. Each study includes three reconstructions; all reconstructions attempt to identify the location (in the horizontal plane) and rate of release of the tracer gas. Four sensors with known, fixed locations are distributed within the domain, nominally 2-4 km from the source. (The locations of the sensors correspond to actual locations of sensors used during the 1979 experiments; these sensors are numbers 1-22, 1-38, 3-23, and 3-32 using Gryning's sensor identification system (Gryning 1981).) This situation is analogous to a scenario in which sensor locations are predetermined by logistical constraints, but sensor characteristics, such as averaging time or sensitivity, are flexible. All simulations use the same hourly meteorology and seek to characterize the same release, which commences at 1050 UTC and concludes at 1220 UTC, releasing material at a constant release rate of 3.2e+09 ng/sec.

Each simulation uses sensors from the same four locations for each simulation, as well as "synthetic" concentrations reported by LODI given the actual source location and release rate. The synthetic concentrations recorded at each location were based not only on LODI's predicted concentrations at the 100m x 100m (x 20m high) grid cell encompassing each sensor's location, but also including the eight nearest neighbors of that grid cell using the weighting scheme shown in Figure 3. Because sensors in the Copenhagen experiment were typically mounted on street poles at altitudes 0-20m above the surface, LODI concentrations for the lowest 20m are considered.

The reconstructions are summarized in Table 1. For the first study, the three reconstructions differ in the time resolution of the sensor providing data. Although all sensors recorded data from 1038-1238 UTC, the “60m resolution” sensors (where “m” indicates “minutes”) reported averaged concentrations for 1038-1138 and 1138-1238; the “10m resolution” sensors reported averaged concentrations for a total of twelve ten-minute intervals; and the “5m resolution” sensors reported averaged concentrations for a total of twenty-four five-minute intervals. In a domain of this size (6km length scale) and for wind speeds of this magnitude (average of 9.2 m/s at 60m altitude over the three relevant hours), ten minutes are required for material to be transported throughout the domain. Therefore, only an instrument with time resolution at or greater than this ten-minute time scale is expected to provide adequate information to characterize the source location and magnitude. All the sensors can detect atmospheric concentrations between 10 and 10000 ng/m3, or three orders of magnitude, based on the reported detection limits used in the original Copenhagen study (Gryning, 1981).

For the second study, the detection limit (or sensor sensitivity) varies. All the sensors report data at ten-minute intervals, as the “10m resolution” sensors in the first experiment. The “low-threshold” reconstruction uses data from instruments with an expanded lower detection limit, reporting in a range from 0.01 to 10000 ng/m3, or over six orders of magnitude. The moderate threshold sensor network reports data over three orders of magnitude, from 10 to 10000 ng/m3. The “high-threshold” reconstruction uses limited data from instruments reporting from 1000 to 10000 ng/m3.

Event reconstruction was carried out for 5000 iterations for each sensor set, searching over source x and y location and release rate. All simulations were carried out on Livermore Computing’s mcr platform, using less than 12 cpu hours on 68 2-processor nodes.

	Time Resolution of sensors (minutes)	Sensor detection range (orders of magnitude)	Qualitative description of network
four_05mres	5	3	Expensive
four_10mres	10	3	Moderate
four_60mres	60	3	Inexpensive
low_thresh	10	6	Expensive
four_10mres	10	3	Moderate
high_thresh	10	1	Inexpensive

Table 1: Summary of reconstructions discussed herein. The “four_10mres” reconstruction is utilized in both the time-resolution study and the detection limit study.

4. Assessment of the sensor networks tested

Several metrics can assess which type of network provides optimal information to users wishing to understand the source of material responsible for the observed data. Before considering these metrics, it is advisable to ensure that a reconstruction has converged. These metrics include histograms of source characteristics, probability contours of source location, and finally, composite plumes based on the posterior distribution from the reconstruction.

a. Convergence metrics

The posterior distribution can only be determined if a reconstruction has converged. Only information generated after convergence should be considered when evaluating a sensor network. Little information can be gleaned from convergence tests other than the fact that convergence has been attained, which is necessary for subsequent analysis of the posterior distribution to which the reconstruction has converged.

Convergence is typically defined (Gelman et al., 2004, p. 297) by a measure of the variation between the chains used in the reconstruction to variation within each chain. When this ratio, R_hat, approaches 1 (in practicality, is less than 2), convergence is said to be attained. Each of the three reconstructions converged rapidly, within the first 1000 iterations. Iterations 2000-5000 constitute the posterior distributions, presented here via histograms, probability contours, and composite plumes. The convergence metric R_hat for all three reconstructed parameters (x-location, y-location, and release rate) are shown in Figure 4 (for the time-resolution study) and in Figure 5 (for the sensor-threshold study).

b. Location histograms

After convergence has been attained, the posterior distribution of the reconstruction reveals the characterization of the source. For synthetic studies and reconstructions based on studies in which the exact characteristics of the source are known, as presented here, a comparison of the histogram to “truth” can generate confidence in the reconstruction. In cases for which “truth” is unknown and to be determined, the histogram’s nature (flat vs. sharp) can indicate how much information is attainable from available data.

The histograms for each of the three source characteristics reconstructed (y-location; x-location; and release rate), along with an indicator of “truth”, are shown below. The largest uncertainty in the reconstruction of a source location is typically in the direction along the mean wind, which in this case, is in the x-direction. Therefore, the histograms of y-locations should be narrowly focused and more correct.

In the time-resolution experiment, all sensor arrays identify the y-location of the source within a 3-km range, while the reconstruction explored a 10-km range (Figure 6). However, the 60-min (Figure 6c) resolution sensors provide a reconstruction with a bimodal distribution of y-location, with one peak at the correct location (indicated with the heavy vertical line) and one location 1.5 km north of the correct location. This bimodal probability distribution indicates a suboptimal network.

In the detection-threshold experiment, the “expensive” network with instruments with a low detection threshold provides a reconstruction that clearly and correctly identifies the y-location of the source (Figure 7a), indicating the utility of this type of instrument. The moderate-threshold instruments constitute a suboptimal network (Figure 7b), indicating a bimodal distribution of y-location, with one peak at the correct location (indicated with the heavy vertical line) and one location 1.5 km north of the correct location. Finally, the inexpensive network of instruments with a high detection limit provides no information on the y-location of this source, indicated by the flat distribution in Figure 7c.

As noted above, the greatest location uncertainty in a reconstruction is typically in the direction of the mean wind, which in this case was from the west. This large uncertainty is seen in the broad probability distributions for x-location.

In the time-resolution experiment, distributions of the x-locations include probable locations within a 5-km range around the correct source location (and would probably extend further upwind had the domain been large enough to include such locations). The superior time resolution of the 5-minute resolution sensor network did not provide the ability to reduce the uncertainty in x-location for this case (Figure 8a) over the uncertainty determined with the moderate network (Figure 8b). This failure disproves the hypothesis that the 5-minute resolution sensors would resolve the arrival time of the plume more precisely than the 10-minute resolution sensors, because the advection time from an upwind sensor to a downwind sensor (distance of nominally 4 km) with mean winds of 9.2 m/s is less than 8 minutes. The tracer gas release did last for ninety minutes, however; an instantaneous release of tracer gas would likely be reconstructed better by the higher time resolution sensors.

In the detection-threshold experiment, a more marked difference between the networks is evident. The low-detection threshold instruments do identify the correct x-location within 3 km, with a Gaussian distribution (Figure 9a) and a significant peak close to the real source. The moderate network identifies a broader range of possible x-locations (Figure 9b), while the high-detection limit instruments provide no information at all to reduce the range of possibilities (Figure 9c).

c. Joint location histograms

When a source characteristic, such as location, is defined by more than one parameter, such as x and y, more insight can be gleaned by the inspection of joint histograms. Presented in Figure 10 and Figure 11 are joint histograms of probability of source location for both x and y, superimposed on a map including the real source location (the red triangle) and the sensor locations (the four green diamonds), for the two experiments. Shading indicates the joint probability of a particular cell being the source location; the more intense blues represent higher probability.

In the time-resolution experiment (Figure 10), the joint probability distributions based on data from the 10-minute and 60-minute resolution sensor networks clearly illustrate that these networks cannot distinguish between two clusters of possible source locations: one band includes the correct location, and another band to the north includes a peak at the wrong location. The 10-minute resolution network has an especially strong peak at an incorrect location (x=343, y = 6481) that is weighted more strongly than the peak at the correct location. Only reconstruction based on the 5-minute resolution sensor network correctly emphasizes the southern band, which encompasses the true source location, although it does include both bands.

In the detection-limit threshold experiment (Figure 11), the advantage of the low-detection-limit network is obvious. In the joint histogram for the expensive network, only a few locations are highlighted, and those locations are very close to, within 500m of, the actual source, with some upwind uncertainty (Figure 11a). The network composed of sensors with a moderate detection limit cannot distinguish between two clusters of possible source locations: one band that includes the correct location, and another band to the north that includes a peak at the wrong location. The moderate network identifies an especially strong peak at an incorrect location (x=343, y = 6481) that is weighted more strongly than the peak at the correct location (Figure 11b). Finally, the high-detection-limit network (Figure 11c) cannot reduce the infinite range of possible source locations beyond identifying that the source location is not immediately upwind of the sensors, as noted by the white areas (indicating zero probability) upwind of the sensors. The rest of the domain consists of possible source locations, indicating that this sensor network provides no information at all to decision-makers seeking to understand the characteristics of the source of an atmospheric release of this magnitude.

d. Release rate histograms

When identifying the source of an unknown atmospheric release, the magnitude of the release is often an important parameter. Knowing the size of the source term can guide emergency-response actions, such as determining whether evacuation or sheltering-in-place is appropriate. The size of the source term is also important for post-release cleanup efforts. An ideal sensor network, coupled with event reconstruction methodologies, should therefore be able to quantify the size of a detected atmospheric release. Histograms of the release rate reconstructions for these two experiments appear in Figure 12 and Figure 13.

Inspection of the release rate histogram clearly indicates the problematic nature of reconstruction with very coarse time-resolution instruments (Figure 12). The histogram based on 60-min sensor data (Figure 12c) is very flat, filling almost the entire range of release rates considered. Both the 10-min (Figure 12b) and the 5-min (Figure 12a) sensor data narrow the field of possibilities to acceptable limits. The 10-min sensor data reconstruction indicates a slight peak in probability at the correct release rate, although the determination of the "best" sensor array should not be based on the histogram of one quantity alone, but on the aggregate evaluation of all desired source parameters, as available in the composite plume.

In the detection-limit experiment, the least expensive network again fails to provide useful information, filling almost the entire range of release rates considered (Figure 13c). The moderate network indicates a slight peak in probability at the correct release rate (Figure 13b), although this release rate peak corresponds to an incorrect location, as discussed above in Figure 11b. Surprisingly, the reconstruction based on data from the expensive network cannot precisely identify the strength of the source (Figure 13a) as well as it identifies the location of the source. Quantification of the utility of different sensor networks, via articulation of the networks' capabilities, would be helpful information for decision-makers.

e. Composite and aggregate plume predictions

A key piece of information for a decision-maker choosing between sensor networks is a composite plume, a reconstruction of the original plume based only on the data provided from the sensors and the forward model via the Bayesian stochastic inversion. This composite summarizes the posterior distribution of all possible source characteristics, weighting each source characteristic (here, X and R) by their probability of occurring as determined by the reconstruction (and seen in the histograms presented herein). Because the refinement of the reconstruction is determined by the characteristics of the sensor network, this measure of network performance provides a quantitative means of evaluating the utility of a given sensor network.

Composite plumes are generated by aggregating together all runs from the forward model, based on the Xn and Rn tested during the reconstruction. At each grid cell in the forward model's domain, at each time step simulated, there exists a distribution of atmospheric concentrations due to the dispersion predicted by the forward model based on each Xn and Rn. A plume dispersion plot consisting of the total concentration expected in each cell, normalized by the total number N of Xn and Rn that contributed to that concentration, would create an aggregate plume. This aggregate plume representation does not explicity incorporate the probabilistic information obtained via the event reconstruction. To incorporate the probabilistic information, a confidence threshold level is defined by the decision-maker. (In this study, the 90% confidence level is utilized.) The composite plume indicates, for each cell, that the reconstruction based on the sensor data is 90% confident that concentrations at that cell are above a certain threshold.

Composite plumes for the detection-limit experiment are shown in Figure 14. The composite plume from the reconstruction based on the low-detection-limit sensor network (Figure 14a) is able to reproduce both the finely-structured stochastic nature of the plume edges and a high-concentration contour (outlined in black) that is also evident in the original plume (Figure 14d). The reconstruction based on the moderate-detection-limit sensor network (Figure 14b) can also reproduce the general shape of the original plume. The reconstructed composite plume in Figure 14b is also broader than the original and includes the “alternate” source to the north of the real source, potentially providing misleading information. Finally, the reconstruction using the high-detection limit sensors, shown in Figure 14c, provides minimal useful information to a decision-maker, failing to include the real source location or identify a high-concentration-level contour (as represented by the black line seen in Figure 14a, b, or d). The only information provided in Figure 14c is that a release may have happened somewhere in the domain of interest and that it affected the two downwind sensors at this timestep; no resolution of the features of the plume is possible.

The composite plume can be very useful for a decision-maker seeking to determine how to utilize sensors to protect an asset. Numerical experiments such as these, which incorporate sensors with different characteristics (detection thresholds, time resolution, false-alarm frequency, and expense), along with likely release scenarios, can quantitatively illustrate what information different types of networks would provide in the case of an atmospheric release of hazardous materials. In the detection-limit experiment shown above, the decision-maker would weigh the utility of defining a high-concentration contour (outlined in black in Figure 14, which could correspond with a Protective Action Guideline), available from the low-detection-limit and moderate-detection-limit networks, against the higher cost of the low-detection-limit network.

5. Conclusions

Sensor networks must be designed to provide enough quantitative information about an event to reduce uncertainty in emergency response. We have demonstrated the application of an event reconstruction technology system to designing sensor networks for detecting and responding to atmospheric releases of hazardous materials. This system, developed at Lawrence Livermore National Laboratory, integrates observations with transport and dispersion models via Bayesian stochastic sampling methodologies to characterize the sources of atmospheric releases of hazardous materials. The event reconstruction methodology identifies source characteristics that are most consistent with the observed data, and then can provide an ultimate prediction, or composite plume, describing likely affected areas. This ultimate prediction is a composite of all the likely sources. In a real event, it can guide emergency responders more effectively than a single forward prediction from a single estimate of source characteristics as it provides a measure of uncertainty in source characterization and the result of the release of hazardous material. Before an event, this quantitative measure of the reduction in uncertainty, or benefit of a given network, can be utilized by policy makers to determine the cost/benefit of various networks.

Herein we present two numerical experiments demonstrating the utility of the event reconstruction methodology for sensor network design. Both experiments are loosely based on the Copenhagen tracer experiment (Gryning, 1981; Gryning and Lyck, 1984), but numerical sensor network design experiments require only climatological weather data, a dispersion model, and specifications of the types of sensors being considered – no actual tracer experiment data is required. In the present study, data from each network was provided to the event reconstruction system in order to identify the location and magnitude of a 1.5-hour release of a neutrally-buoyant gas in a 6km x 6 km suburban area.

In the first set of experiments, only the time resolution of the sensors varies between the three reconstructions. The most “expensive” sensor network, which provided data every five minutes (as compared to every ten minutes – “moderately-priced” or every sixty minutes – “inexpensive” – over this 1.5-hour release), offers only a few advantages over the moderately-priced network when attempting to reconstruct the location of the source explored here. Utilizing data from either the “moderate” or the “expensive” network, the event reconstruction methodology could identify the source location of the release within 5km, and could identify the magnitude of the source within 25%.

The second set of experiments presented herein explore the significance of the sensors’ detection limit, which can have a significant impact on sensor cost. All sensors report data every ten minutes. The “expensive” network had a very low detection limit, and could distinguish data within a range of six orders of magnitude. The “moderate” network could identify data within a range of three orders of magnitude, while the “inexpensive” network could identify data within one order of magnitude. The upper limit, or saturation level, of the instruments in all three networks was identical. In this set of experiments, the expensive network can most clearly define the source location and source release rate. The other networks provide data insufficient for distinguishing between two possible clusters of source locations. When the reconstructions from all networks are aggregated into a composite plume, a decision-maker can distinguish the network that best suits needs. Reconstructions from both the expensive network and the moderately-priced network can reproduce certain high-threshold contours of atmospheric concentrations from the release considered here.

However, because of the limited sensitivity of the moderately-priced network, reconstruction from that network incorrectly predicts effects of the release in regions that would not be affected. A decision-maker could thus weigh the potential false-positive risk against the cost savings of that network.

The experiments presented herein have explored only a single type of release and a single meteorological scenario, in order to demonstrate the application of event reconstruction to sensor network design. More complete sensor network studies would consider multiple climatological conditions (i.e. wind speed, wind direction, atmospheric stability) representative of the region of interest. A range of possible source magnitudes may also be explored to ensure that the network would provide useful composite plumes in most likely scenarios.

6. References

Chan, S. and D. Stevens, 2000, An Evaluation of Two Advanced Turbulence Models for Simulating the Flow and Dispersion Around Buildings, The Millennium NATO/CCMS Int. Tech. Meeting on Air Pollution Modeling and its Application, Boulder, CO, May 2000, 355-362.

Chow, K. K., B. Kosović, and S. T. Chan, 2006: Source Inversion for Contaminant Plume Dispersion in Urban Environments using Building-Resolving Simulations, American Meteorological Society's 6th Symposium on the Urban Environment, Atlanta, Georgia, Jan 29 – Feb 2, 2006.

Ermak, D.L., and J.S. Nasstrom, 2000: A Lagrangian Stochastic Diffusion Method for Inhomogeneous Turbulence, *Atmos. Environ*., 34, 7, 1059-1068.

Gelman, A., J. B. Carlin, H. S. Stern, and D. B. Rubin, 2004: *Bayesian Data Analaysis, Second Edition*. Chapman & Hall/CRC. 668 pp.

Gryning, S.-E., 1981: *Elevated Source SF6-Tracer Dispersion Experiments in the Copenhagen Area*. Risø-R-446. Risø National Laboratory, 187pp.

Gryning, S.-E. and E. Lyck,1984: Atmospheric Dispersion from Elevated Sources in an Urban Area: Comparison between Tracer Experiments and Model Calculations. *J. Clim. Appl. Meteorol.*, **23**, 651-660.

Kosović, B., and co-authors, 2005: Stochastic Source Inversion Methodology and Optimal Sensor Network Design. 9th Annual George Mason University Conference on "Atmospheric Transport and Dispersion Modeling", Fairfax, VA, July 18-20, 2005. UCRL-PRES-213633.

Larson, D.J., and J.S. Nasstrom, 2002: Shared and Distributed Memory Parallelization of a Lagrangian Atmospheric Dispersion Model, *Atmos. Environ*., **36**, 1559-1564.

Liu, J. S., 2001: *Monte Carlo Strategies in Scientific Computing*. Springer Series in Statistics, 343 pp.

Robert, C. P. and G. Casella, 2005: *Monte Carlo Statistical Methods, Second Edition*. Springer Texts in Statistics, 645 pp.

Petersen, W.B., and Lavdas L.G., 1986. INPUFF 2.0 - A Multiple Source Gaussian Puff Dispersion Algorithm. User's Guide. U.S. E.P.A., Research Triangle Park, NC.

Shea, D. A. and S. A. Lister, 2003: The BioWatch Program: Detection of Bioterrorism. Congressional Research Service Report No. RL 32152. Available at http://www.fas.org/sgp/crs/terror/RL32152.html, downloaded 12/15/2005.

Sugiyama, G., and S.T. Chan, 1998. A new meteorological data assimilation model for real-time emergency response. *10th Joint Conference on the Applications of Air Pollution Meteorology*, Phoenix, AZ, Jan. 11-16, 1998, American Meteorological Society, Boston, MA. 285-289.

7. Acknowledgements

Appreciation is extended to the entire Event Reconstruction team at LLNL, including Chuck Baldwin, Stevens T. Chan, Fotini K. Chow, Kathleen Dyer, Ron Glaser, William Hanley, Gardar Johannesson, Milovan Krnjajić, Shawn Larsen, Gwen Loosmore, Arthur Mirin, Stephanie Neuman, John Nitao, Radu Serban, Alan Sicherman, Gayle Sugiyama, and Charles Tong. The authors also appreciate discussions with Roger Aines and Jeffrey Mirocha.

8. Figures

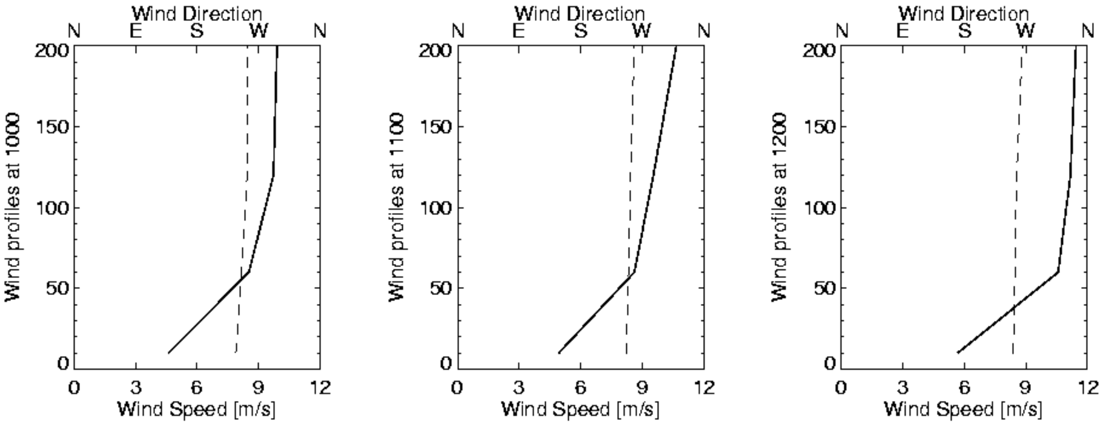


Figure 1: Vertical profiles of wind speed (solid line) and wind direction (dashed line) for the three hours relevant to these simulations

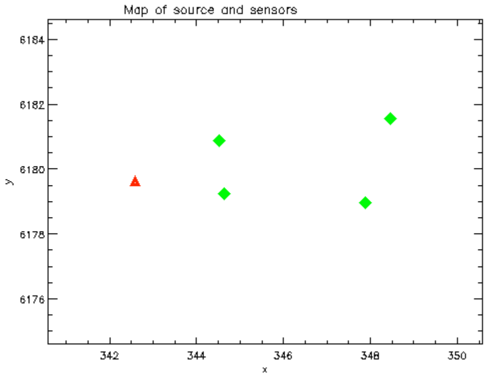


Figure 2: The source (red triangle) and sensors (green diamonds) used in this study. The winds, as noted in Figure 1, are from the west.

.5	1	.5
1	2 ✚	1
.5	1	.5

Figure 3: Weighting of concentrations calculated for cells including and around a sensor (indicated with cross).

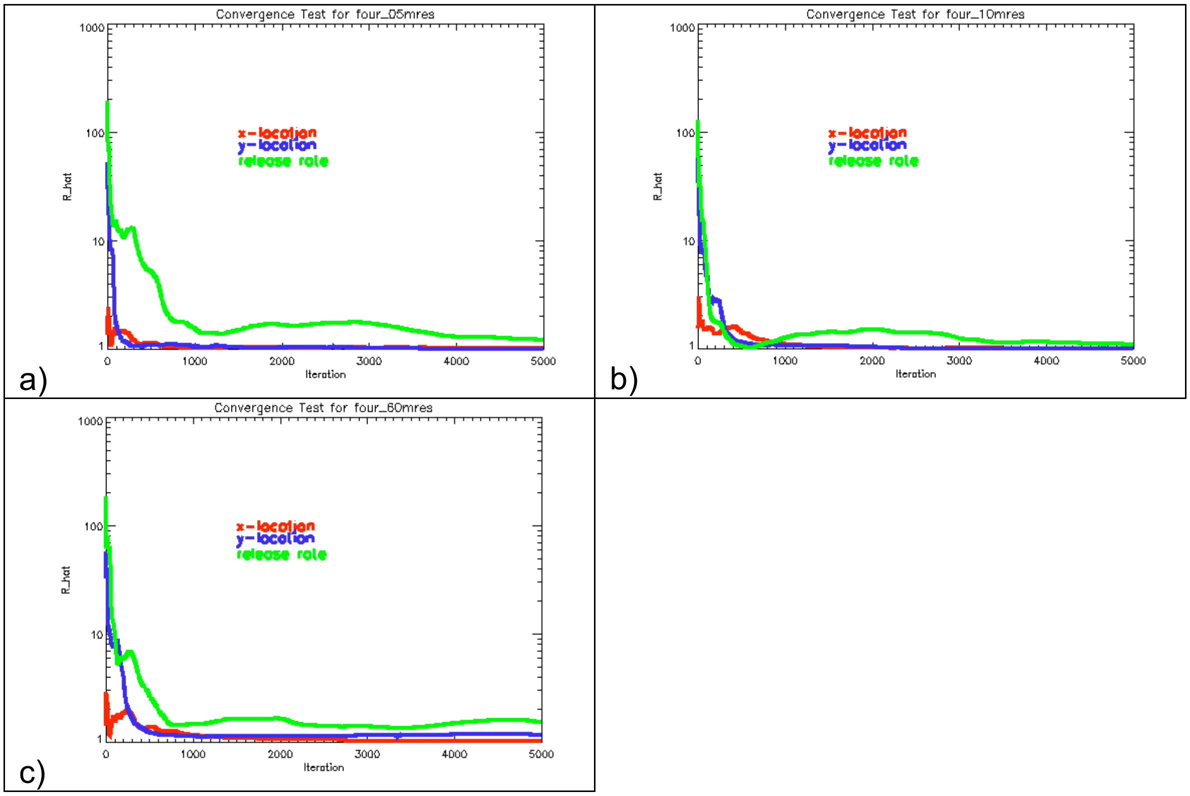


Figure 4: Convergence metrics for the time-resolution study using a) the five-minute network, b) the ten-minute network, c) the 60-minute network

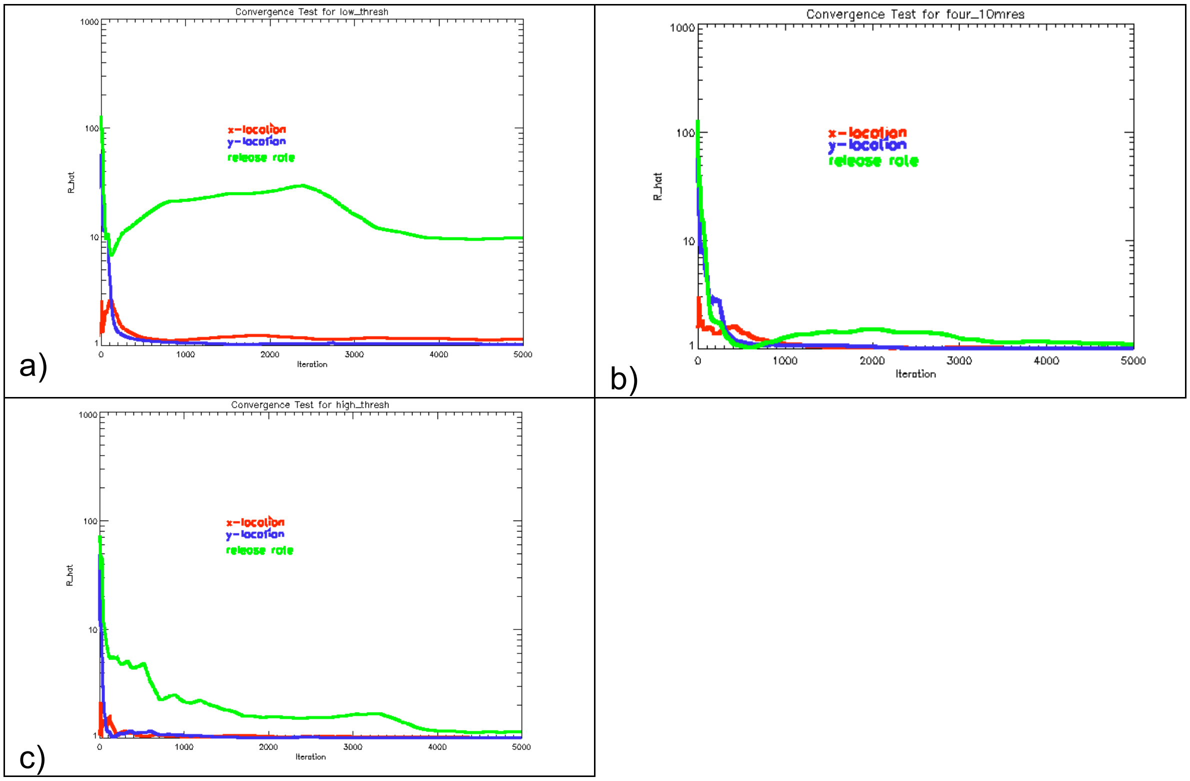


Figure 5: Convergence metrics for the sensor-sensitivity study using a) the low-threshold network, b) the moderate-threshold network, c) the high-threshold network

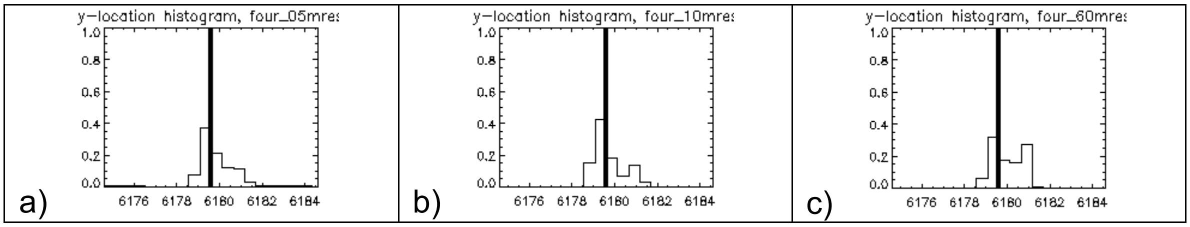


Figure 6: Histograms for y-location for simulations with a) 5-minute b) 10-minute, c) 60-minute resolution sensors.

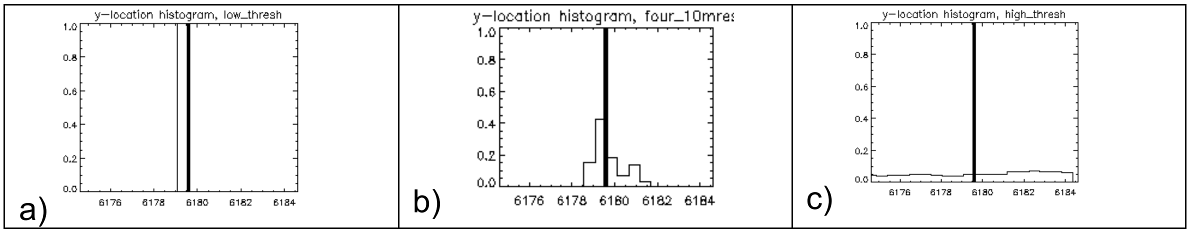


Figure 7: Histograms for y-location for simulations with a) low-threshold, b) moderate-threshold, and c) high-threshold sensors.

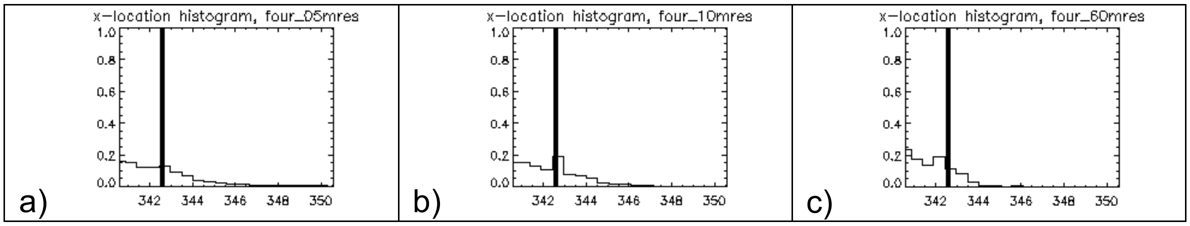


Figure 8: Histograms for x-location for simulations with a) 5-minute b) 10-minute, c) 60-minute resolution sensors.

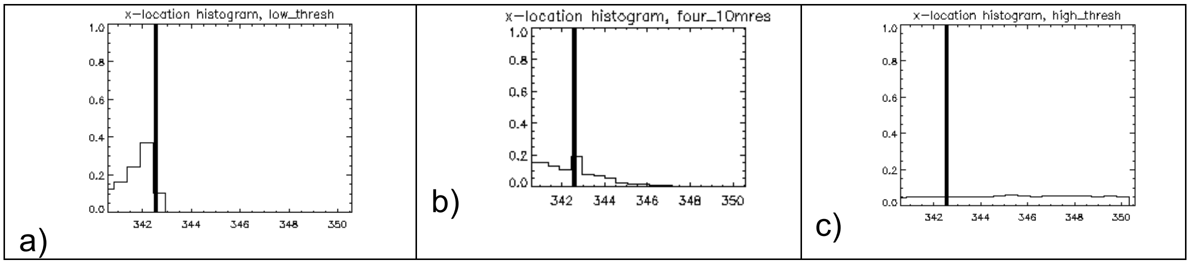


Figure 9: Histograms for x-location for simulations with a) low-threshold, b) moderate-threshold, and c) high-threshold sensors.

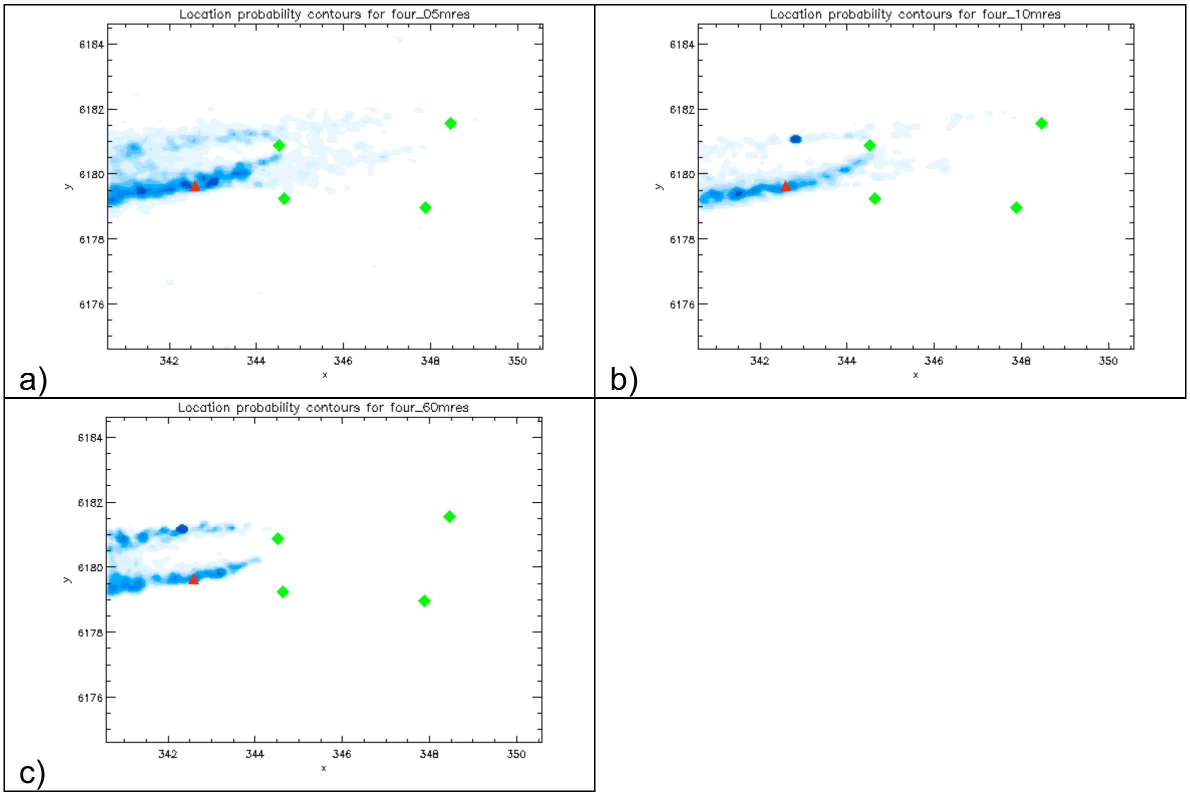


Figure 10: Joint histogram for x and y location for simulation with sensors with a) 5-minute, b) 10-minute, and c) 60-minute time resolution

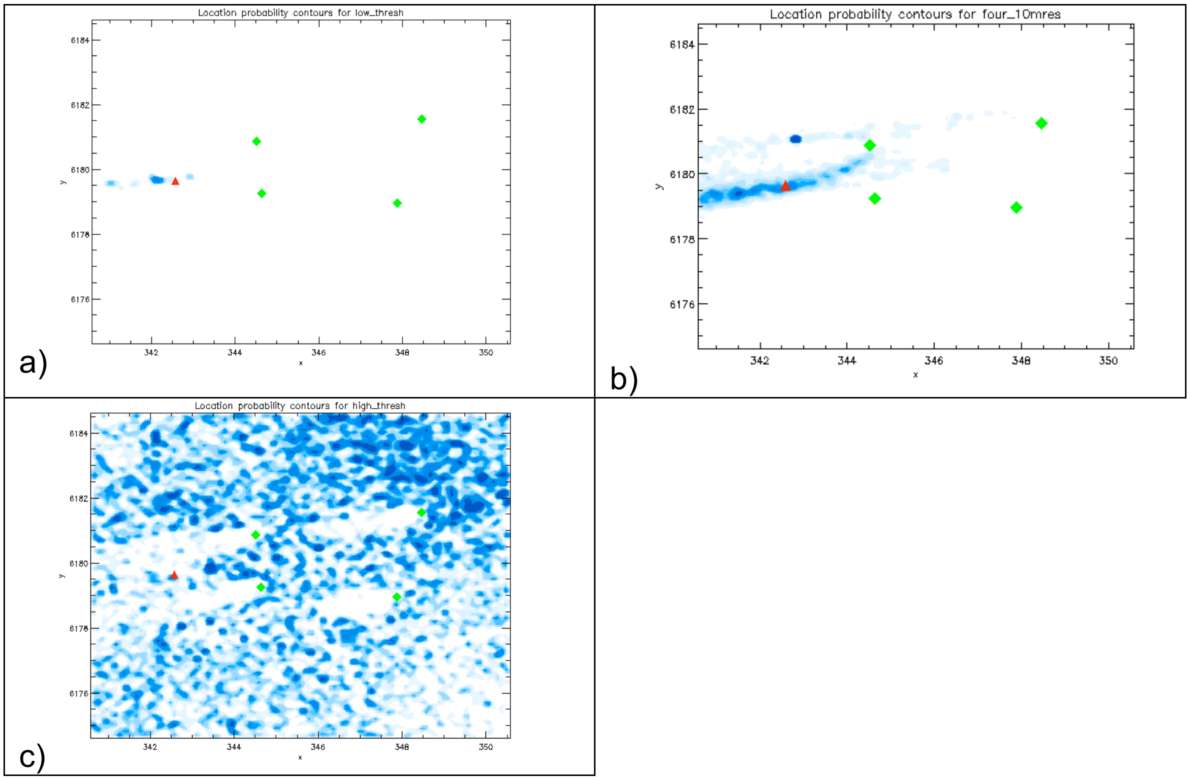


Figure 11: Joint histogram for x and y location for simulation with sensors with a) low, b) moderate, and c) high thresholds. Darker blues indicate higher probability locations.

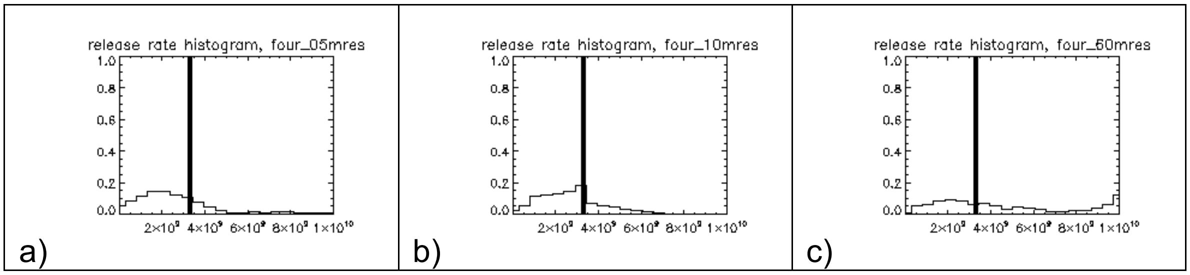


Figure 12: Histogram for release rate for simulation with sensors with a) 5-minute, b) 10-minute, and c) 60-minute time resolution

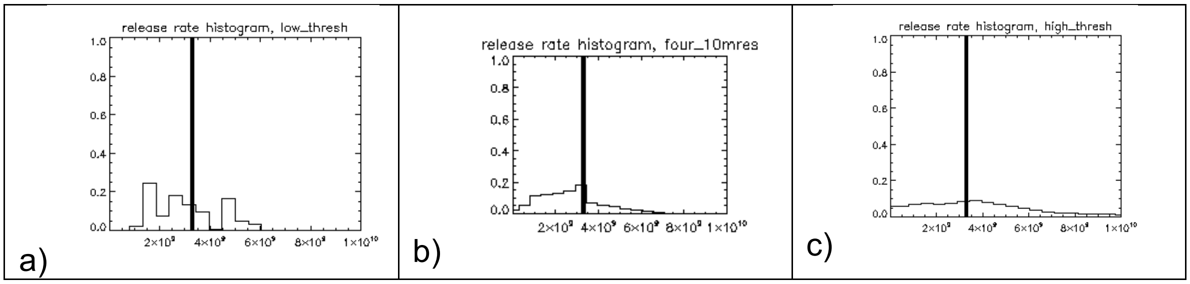


Figure 13: Histogram for release rate for simulation with sensors with a) low, b) moderate, and c) high detection limits

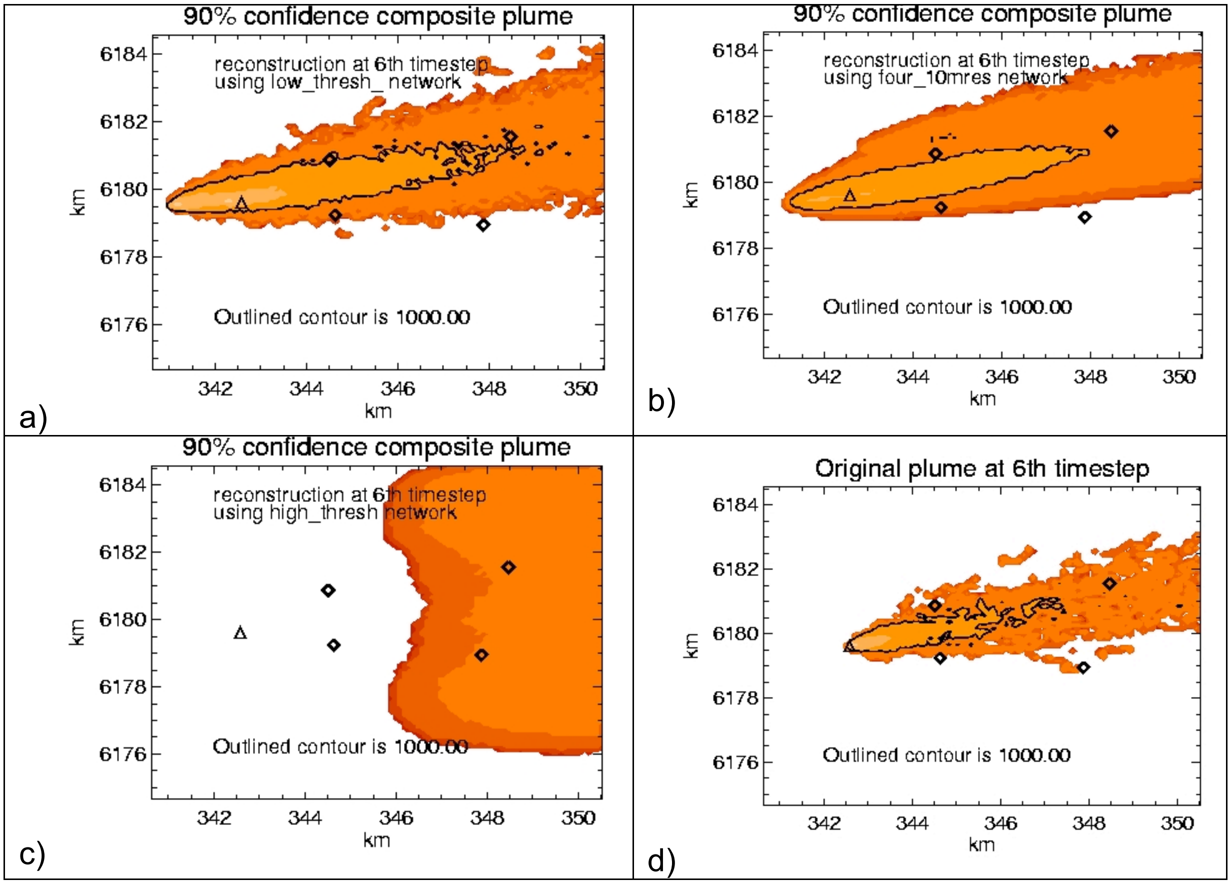


Figure 14: Composite plumes with logarithmic color contours, generated from a) the low-detection level , b) the moderate-detection level, and c) the high-detection level sensor networks. Colored contour levels indicate the magnitude of atmospheric concentrations in which a decision-maker can have 90% confidence. Highest concentrations are in yellow and light orange; lower concentrations are in red and dark red. Note that the reconstruction with the high-detection-level network c) provides confidence only that half of the domain will experience concentrations above a low level. Plume d) represents the original plume, from which the data for the reconstruction were generated. Note that both the reconstructions using the low-detection limit instruments, a), and the moderate-detection limit instruments, b), can reproduce the high-concentration contour of 1000 ng/m3, outlined in a), b), and d) with a solid black line. Note also in b) that the plume extends further in the north-south direction to encompass the probability that the source might be at a second alternate location to the north of the real location.

9. Appendix A – adapt.nml, observ.met, stnloc.met files for the meteorology used in this example

a. adapt.nml

```
 *** adapt.nml automatically generated on 19-Aug-2004 08:53:44 via code written by
Michael Dillon on 5-21-04


 &adapt_control
  flag_debug   = .true.
 /

 &adapt_grid
  file_met_grid     = "../main_grd_copenhagen.nc"
  opt_grid_file     = "gridgen"
 /

 *** Beginning adapt parameters for met time1979JUL19_100000

 &adapt_metdata
  file_met_field     = "met_field_1979JUL19_100000.nc"
  opt_src_obs        = "ascii2"
  opt_src_field      = "none"
  file_src_obs       = "observ.met"
  file_src_station   = "../stnloc.met"
  flag_station_km    = .true.
  nmethod            = 2
 /

 &adapt_field2D
  hgt_vert_coord         = "zAGL"
  hgt_boundary_layer     = 2090
  hgt_geostrophic_layer  = 2090
  z0                     = 0.6000000
  inv_monin_obukhov_len  = -2.617801e-003
  friction_velocity      = 7.700000e-001
 /

 &adapt_method
  opt_wind_horz              = "spddir"
  flag_use_missing_wind = .true.
  opt_met_type          = "wind2d"
  obs_date_time         = "1979JUL19_100000"
  blend_exp             = 0.100000
  flag_upr_in_sl        = .false.
  flag_twr_local_only   = .false.
  flag_mc_adjust        = .false.
  met_x_border          = 100000.0
  met_y_border          = 100000.0
  blend_max_veer        = 180
  sl_pwr_exp            = 0.1900000
 /

 &adapt_mc_adjust
 /

 &adapt_method
  opt_method = "turb"
 /

 &adapt_turbulence
  sigmav_tavg       = 3600
  sigmav_tavgo      = 3600
  sigmav_t_lagran_h = 400
  turb_param_h      = "sigmav_simthry"
  sigmav            = 1.71
  sigmav_meas_hgt   = 115
  turb_param_z      = "simthry"
  sim_kz_c          = 4
  sim_kz_trop       = 0.01
 /

 *** Beginning adapt parameters for met time1979JUL19_110000

 &adapt_metdata
  file_met_field     = "met_field_1979JUL19_110000.nc"
  opt_src_obs        = "ascii2"
  opt_src_field      = "none"
  file_src_obs       = "observ.met"
  file_src_station   = "../stnloc.met"
  flag_station_km    = .true.
  nmethod            = 2
 /

 &adapt_field2D
  hgt_vert_coord         = "zAGL"
  hgt_boundary_layer     = 2090
  hgt_geostrophic_layer  = 2090
  z0                     = 0.6000000
  inv_monin_obukhov_len  = -2.617801e-003
  friction_velocity      = 7.700000e-001
 /

 &adapt_method
  opt_wind_horz              = "spddir"
  flag_use_missing_wind = .true.
  opt_met_type          = "wind2d"
  obs_date_time         = "1979JUL19_110000"
  blend_exp             = 0.100000
  flag_upr_in_sl        = .false.
  flag_twr_local_only   = .false.
  flag_mc_adjust        = .false.
  met_x_border          = 100000.0
  met_y_border          = 100000.0
  blend_max_veer        = 180
  sl_pwr_exp            = 0.1900000
 /

 &adapt_mc_adjust
 /

 &adapt_method
  opt_method = "turb"
 /

 &adapt_turbulence
  sigmav_tavg       = 3600
  sigmav_tavgo      = 3600
  sigmav_t_lagran_h = 400
  turb_param_h      = "sigmav_simthry"
  sigmav            = 1.71
  sigmav_meas_hgt   = 115
  turb_param_z      = "simthry"
  sim_kz_c          = 4
  sim_kz_trop       = 0.01
 /

 *** Beginning adapt parameters for met time1979JUL19_120000

 &adapt_metdata
  file_met_field     = "met_field_1979JUL19_120000.nc"
  opt_src_obs        = "ascii2"
  opt_src_field      = "none"
  file_src_obs       = "observ.met"
  file_src_station   = "../stnloc.met"
  flag_station_km    = .true.
  nmethod            = 2
 /

 &adapt_field2D
  hgt_vert_coord         = "zAGL"
  hgt_boundary_layer     = 2090
  hgt_geostrophic_layer  = 2090
  z0                     = 0.6000000
  inv_monin_obukhov_len  = -2.617801e-003
  friction_velocity      = 7.700000e-001
 /

 &adapt_method
  opt_wind_horz              = "spddir"
  flag_use_missing_wind = .true.
  opt_met_type          = "wind2d"
  obs_date_time         = "1979JUL19_120000"
  blend_exp             = 0.100000
  flag_upr_in_sl        = .false.
  flag_twr_local_only   = .false.
  flag_mc_adjust        = .false.
  met_x_border          = 100000.0
  met_y_border          = 100000.0
  blend_max_veer        = 180
  sl_pwr_exp            = 0.1900000
 /

 &adapt_mc_adjust
 /

 &adapt_method
  opt_method = "turb"
 /

 &adapt_turbulence
  sigmav_tavg       = 3600
  sigmav_tavgo      = 3600
  sigmav_t_lagran_h = 400
  turb_param_h      = "sigmav_simthry"
  sigmav            = 1.71
  sigmav_meas_hgt   = 115
  turb_param_z      = "simthry"
  sim_kz_c          = 4
  sim_kz_trop       = 0.01
 /

 *** Beginning adapt parameters for met time1979JUL19_130000

 &adapt_metdata
  file_met_field     = "met_field_1979JUL19_130000.nc"
  opt_src_obs        = "ascii2"
  opt_src_field      = "none"
  file_src_obs       = "observ.met"
  file_src_station   = "../stnloc.met"
  flag_station_km    = .true.
  nmethod            = 2
 /

 &adapt_field2D
  hgt_vert_coord         = "zAGL"
  hgt_boundary_layer     = 2090
  hgt_geostrophic_layer  = 2090
  z0                     = 0.6000000
  inv_monin_obukhov_len  = -2.617801e-003
  friction_velocity      = 7.700000e-001
 /

 &adapt_method
  opt_wind_horz              = "spddir"
  flag_use_missing_wind = .true.
  opt_met_type          = "wind2d"
  obs_date_time         = "1979JUL19_130000"
  blend_exp             = 0.100000
  flag_upr_in_sl        = .false.
  flag_twr_local_only   = .false.
  flag_mc_adjust        = .false.
  met_x_border          = 100000.0
  met_y_border          = 100000.0
  blend_max_veer        = 180
```

```
  sl_pwr_exp            = 0.1900000
 /

 &adapt_mc_adjust
 /

 &adapt_method
  opt_method = "turb"
 /

 &adapt_turbulence
  sigmav_tavg       = 3600
  sigmav_tavgo      = 3600
  sigmav_t_lagran_h = 400
  turb_param_h      = "sigmav_simthry"
  sigmav            = 1.71
  sigmav_meas_hgt   = 115
  turb_param_z      = "simthry"
  sim_kz_c          = 4
  sim_kz_trop       = 0.01
 /
```

b. observ.met

```
METDATASET '1979JUL19_100000'
SFC
'TV TWR'     236.7     4.60
'TV TWR2'     -1.0     8.53
'TV TWR3'     253.3     9.73
'TV TWR4'     253.3     9.90
UPR
'TV TWR'  60    -1.0     8.53
'TV TWR'  120   253.3     9.73
'TV TWR'  200   253.3     9.90


METDATASET '1979JUL19_110000'
SFC
'TV TWR'     246.7     4.93
'TV TWR2'     -1.0     8.62
'TV TWR3'     253.3     9.55
'TV TWR4'     258.3     10.67
UPR
'TV TWR'  60    -1.0     8.62
'TV TWR'  120   253.3     9.55
'TV TWR'  200   258.3     10.67


METDATASET '1979JUL19_120000'
SFC
'TV TWR'     251.7     5.68
'TV TWR2'     -1.0     10.58
'TV TWR3'     256.7     11.18
'TV TWR4'     265.0     11.45
UPR
'TV TWR'  60    -1.0     10.58
'TV TWR'  120   256.7     11.18
'TV TWR'  200   265.0     11.45


METDATASET '1979JUL19_130000'
SFC
'TV TWR'     240.0     5.60
'TV TWR2'     -1.0     9.70
'TV TWR3'     260.0     10.80
'TV TWR4'     260.0     10.80
UPR
'TV TWR'  60    -1.0     9.70
'TV TWR'  120   260.0     10.80
'TV TWR'  200   260.0     10.80
```

c. stnloc.met

```
SFC
'TV TWR'      342.580     6179.610     10
'TV TWR2'     342.580     6179.610     60
'TV TWR3'     342.580     6179.610     120
'TV TWR4'     342.580     6179.610     200
UPR
'TV TWR'      342.580     6179.610
```

10. Appendix B – the .pyin file for the 60m resolution reconstruction

```
# File: four_60mres.pyin
#
# Input file for mcmc_app_copenhagen, generated automatically by Julie
#
################################################################################
# Tells python to look for .py files in the current (working) directory.
import os

from mcmc_app.mcmc_drivers import make_target_sample
from mcmc_app.misc         import make_proc_grp
from mcmc_app.seedmaker    import SeedMaker

# Maximum number of iterations
itermax = 5000

# Number of iterations for burn-in (used only for postprocessing
# or convergence monitoring purposes)
burn_in = 200

# Number of independent sequences
num_seqs = 4

# Number of processors per forward model
# ??? Problem: Allow this to be set to not-1, but degrade gracefully
# ??? in serial mode.
num_procs_per_mod = 32

# Number of processors per sequence
# (total number of processors will be equal to num_seqs times this value.
# (note: for mpi job, r.h.s. must be integer because it is parsed by mpi script)
num_procs_per_seq = 1

# Processor group for mpi jobs; can be set to None for non-mpi jobs
#proc_grp = main_driver.make_proc_grp(num_seqs, num_procs_per_seq)
proc_grp = make_proc_grp(num_seqs, num_procs_per_seq)


# Seed generator -----------------------
# Creates different seeds for different chains, even if they are running
# on different processors
seed = 38895
seed_generator = SeedMaker(seed, itermax, num_seqs, proc_grp=proc_grp)

# MCMC algorithm -----------------------

synthetic_data = {
    # required input:
    'class_name'  : 'LODI_mcmc_et.sampler.ExampleSampler',
    # application specific input:
    'step_size_xy' : 1.0,
    'step_size_z' : 1.0,
    'step_size_q' : 1.0,
    # for this example the base state consists of (x,y) with
    # that are gaussian with following means and sigma's
    'x_mean' : 342580.0,
    'x_sigma' : 1.,
    'x_min' : 342577.0,
    'x_max' : 342582.0,
    'y_mean' : 6179610.0,
    'y_sigma' : 1.,
    'y_min' : 6179607.0, #4704237.0,
    'y_max' : 6179612.0, #4705417.0,
    'q_mean' : 3.3e+09,
    'q_sigma' : 1.e+07,
    'q_min' : 3.28e+09, #0.07779,
    'q_max' : 3.32e+09 #0.07781
}


base_sampler_input = {
    # required input:
    'class_name'  : 'LODI_mcmc_et.sampler.ExampleSampler',
    # application specific input:
    'step_size_xy' : 0.1,
    'step_size_q' : 0.1,
    # for this example the base state consists of (x,y) with
    # that are gaussian with following means and sigmas
    'x_mean' : 345580.0,
    'x_sigma' : 10000.,
    'x_min' : 340580.0,
    'x_max' : 350580.0,
    'y_mean' : 6179610.0,
    'y_sigma' : 10000.,
    'y_min' : 6174610.0,
    'y_max' : 6184610.0,
    'q_mean' : 3.2e+09,
    'q_sigma' : 3.e+09,
    'q_min' : 1.0e+4,
    'q_max' : 1.0e+15
}

# Make the target state (synthetic truth) using the base sampler
# (not needed if not used by log_like_fun_input, below)
target_sample = make_target_sample(synthetic_data, seed_generator)

class LODINmlTemplate :
    def __init__(self) :

        self.template_lines = (
      '&prob_setup',
      '    title           = " Copenhagen Experiment - IOP10 " ',
      '    tstart_str      = "1979JUL19_103800"',
      '    tstop_str       = "1979JUL19_123800"',
      '    dt_part_str     = "02:00:00"',
      '    nbins           = 1',
      '    nsrc            = 1',
      '    num_met_times   = 3',
      "    met_time_strs = ",
               '              "1979JUL19_100000"',
               '              "1979JUL19_110000"',
               '              "1979JUL19_120000"',
      '     dt_dump_str    = "0::0:0:0"',
      '     dt_min         = 0',
      '     dt_fact_adv    = 1',
      '     dt_fact_dif    = 1',
      '     dt_limit       = 3600',
      '     dz_dep         = 20',
      '     met_format     = "arac"',
      '     out_bin_ascii  = .false.',
      '     out_part_ascii = .false.',
      '     rd_grid        = "gridgen"',
      '     rdm_dist       = "nongauss"',
      '     reflect        = "vertical"',
      '     solver_id      = "rk2"',
      ' /',
      '',
      '&thist_param',
      ' /',
      '',
      '&src_param',
      '    source_id      = "Source  1"',
      '    max_num_part   = 10000',
      '    species        = "SF6"',
      '    mass_distrib   = "table"',
      '    m_bin_fract    = 1.0',
      '    m_bin_diam_max = 0.0',
      '    m_bin_diam_min = 0.0',
      '    nset_dep_vel   = 0.0000000E+00',
      '    geom_time_strs = "1979JUL19_105000"',
      '    geom_type      = 2  ',
      'key_x_pt', #'    mean_x         =    342580.0',
      'key_y_pt', #    mean_y         =    6179610.',
      '    std_x          =    1.000000',
      '    std_y          =    1.000000',
      '    cutoff_dx_min  =    2.500000',
      '    cutoff_dy_min  =    2.500000',
      '    cutoff_dx_max  =    2.500000',
      '    cutoff_dy_max  =    2.500000',
      '    mean_z         =    10.00000',
      '    std_z          =    1.000000',
      '    cutoff_dz_min  =    2.500000',
      '    cutoff_dz_max  =    2.500000',
      '    er_time_strs   = "1979JUL19_105000      1979JUL19_122000"',
      'key_emiss_rates', #    emiss_rates    =    3.200000e+009      0.0000000E+00',
      '    er_units_type   = "mass"',
      '    decay_param     = "none"',
      '    half_life       = 0.0',
      '    lambda          = 0.0',
      '    decay_chain     = .false.',
      '    start_time_str  = "1979JUL19_105000"',
      '    stop_time_str   = "1979JUL19_122000"',
      '    dt_hold_str     = "0::0:0:0"',
      '    source_model    = "neutral"',
      '    src_generation_method = "new"',
      '    src_agl_flg     = .true.',
      ' /',
      '',
      '&bin_param',
      '    bin_id         = "Bin  1"',
      '    samp_type      = "average"',
      '    type           = "air"',
      '    orientation    = "xy"',
      '    bin_agl_flg    = .true.',
      '    position       = 10.0',
      '    width          = 20.0',
      '    dt_samp_str    = "0::01:00:00"',
      '    dt_bin_out_str = "0::01:00:00"',
      '    source_list    = "Source  1"',
      '    species_name   = "SF6"',
      ' /',
      '',
      '&turb_param',
      '    read_adapt_turb = .true.',
      ' /',
      '',
      '&met_param',
      ' /  ',
         '',
         '',
         '',
        )


    def create_nml(self, state, output_file_name) :
        # Create the LODI nml file, using a template and the new state
        # information.
        file_out = open(output_file_name, 'w')

        q=state.sampler_data['q']
        for line in self.template_lines:
            if line == 'key_x_pt':
                print >>file_out, '   mean_x =', state.sampler_data['x']
            elif line == 'key_y_pt':
                print >>file_out, '   mean_y =', state.sampler_data['y']
            elif line == 'key_emiss_rates':
                print >>file_out, '   emiss_rates =', state.sampler_data['q']
            else:
                print >>file_out, line

        file_out.flush()
        os.fsync(file_out.fileno())
        file_out.close

        return


LODI_nml_template = LODINmlTemplate()

# Lines to put in LODI_files.nml file
LODI_files_dir = os.getcwd() #+ '/../experiment5' #jkl

LODI_files_nml = (
    "&grid_name",
    "   num_m_grids = 1",
    "   m_grid_name = '" + LODI_files_dir + "/grid/main_grd_copenhagen.nc'",
    "   c_grid_name = '" + LODI_files_dir + "/grid/conc_grd_copenhagen.nc'",
    "/",
    "",
    "&metfiles",
    "   grid_num = 1",
    "   met_file_name = ",
    '             "' + LODI_files_dir + '/iop10/met_field_1979JUL19_100000.nc"',
    '             "' + LODI_files_dir + '/iop10/met_field_1979JUL19_110000.nc"',
    '             "' + LODI_files_dir + '/iop10/met_field_1979JUL19_120000.nc"',
    "/",
    "",
    "&decay_chains_file",
    '   decay_chains_file_name = "decaychains.dat"',
    "/",
    "",
    "",
    "",
)


# Likelihood function -----------------------
# there needs to be one likelihood function for each stage; for this
# example we have only a single stage
log_like_fun_1 = {
    # required input:
    'class_name'       : 'LODI_mcmc_et.likefun.LogLikeFunA',

    # following is information necessary for parallelization
    'num_seqs'          : num_seqs,
    'num_procs_per_mod' : num_procs_per_mod,

    # following is set for random synthetic truth measurements
    'target_sample'   : target_sample,

    # following is set for random synthetic truth measurements
    #'measurement_data'   : data,

    # the iteration at which the likelihood function is actually turned on
    # (for single stage, this should be set to 1; otherwise during staging
    # the likelihood is ignored until the iteration hits the following value)
    'start_iter'      : 1,

    # Settings for model_driver
    'LODI_files_nml' : LODI_files_nml,  # Lines to put in LODI_files.nml file
    'LODI_nml_template' : LODI_nml_template,

    'sensors' : [
                  [344629., 6179248., 3600.  ], # Arc 1-22
#                  [344607., 6180387., 3600.  ], # Arc 1-33
                  [344509., 6180871., 3600.  ], # Arc 1-38
#                  [346559., 6179040., 3600.  ], # Arc 2-23
#                  [346412., 6181080., 3600.  ], # Arc 2-33
#                  [345961., 6181562., 3600.  ], # Arc 2-36
                  [347873., 6178952., 3600.  ], # Arc 3-23
                  [348468., 6181563., 3600.  ], # Arc 3-32
#                  [347792., 6182526., 3600.  ], # Arc 3-36
                  [344629., 6179248., 7200.  ], # Arc 1-22
#                  [344607., 6180387., 7200.  ], # Arc 1-33
                  [344509., 6180871., 7200.  ], # Arc 1-38
#                  [346559., 6179040., 7200.  ], # Arc 2-23
#                  [346412., 6181080., 7200.  ], # Arc 2-33
#                  [345961., 6181562., 7200.  ], # Arc 2-36
                  [347873., 6178952., 7200.  ], # Arc 3-23
                  [348468., 6181563., 7200.  ] # Arc 3-32
#                  [347792., 6182526., 7200.  ]  # Arc 3-36
                 ], # Arc 3-36

    # application specific input (e.g. likelihood function parameters)
    'param_a' : 1.0,
    'param_b' : 0.0,
    'lowerbound' : +1,
    'upperbound' : +4,
    'sigma_rel' : 0.2,
    'option' : 1,
}

log_like_fun_input = [ log_like_fun_1 ]

# Datadumps, plots, monitoring --------------------

state_out = {
    'class_name'    : 'LODI_mcmc_et.dumpers.DumpTextC',

    'burn_in'       : burn_in,
    'starting_iter' : 0,
    'single_file'   : 1,   # Dump all output for a sequence to one file.
}
states_out = [state_out]

# Restart ------------------------

# Settings to write out restart files.
restart_write = {
    'class_name'    : 'mcmc_app.outputs.RestartOutput',
    'starting_iter' : 1000,
}

# Settings to read restart files.
restart_read = {
    # If following is present and is not 0 or None, use initial state from
    # restart file
    'use_restart_new'      : 1,
    'restart_iter_new'     : 4427,       # Iteration for reading restart
}
```

Appendix E

Dynamic Bayesian Models via Monte Carlo - An Introduction with Examples

UCRL-TR-207173

UCRL-TR-207173

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Dynamic Bayesian Models via Monte Carlo - An Introduction with Examples -

G. Johannesson, B. Hanley, J. Nitao

October 12, 2004

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

Dynamic Bayesian Models via Monte Carlo
— An Introduction with Examples —

Gardar Johannesson Bill Hanley John Nitao

September 29, 2004

Lawrence Livermore National Laboratory

UCRL-TR-207173

Abstract

This report gives an introduction to a Bayesian probabilistic approach to modeling a dynamic system, with emphasis on stochastic methods for posterior inference. The Bayesian paradigm is a powerful tool to combine observed data along with prior knowledge to gain a current (probabilistic) understanding of unknown model parameters. In particular, it provides a very natural framework for updating the state of knowledge in a dynamic system. For complex systems, such updating needs to be carried out via stochastic sampling of unknown model parameters. An overview is given of the well established Markov chain Monte Carlo (MCMC) approach to achieve this and of the more recent sequential Monte Carlo (SMC) approach, which is better suited for dynamic systems. Examples are provided, including an application to event reconstruction for an atmospheric release.

Contents

1 Short Introduction to Bayesian Modeling

We shall now give a brief introduction to the Bayesian paradigm to modeling and inference, along with examples. A good introduction to Bayesian theory and modeling is “Bayesian Theory” by Bernardo & Smith (1994) and “Bayesian Data Analysis” by Gelman et al. (2004).

1.1 Basic Notation

Let X and Y be two random variables and denote by:

$p(Y) =$ the probability distribution of Y.

$p(X, Y) =$ the joint probability distribution of X and Y.

$p(X \mid Y) =$ the probability distribution of X conditional on Y.

We shall use the same notation for a continuous random variable, in which case $p(\cdot)$ is referring to a continuous density function, and for a discrete random variable, in which case $p(\cdot)$ is referring to a probability mass function. In addition, we shall not in general distinguish between a (unknown) random variable and a particular value it can take; hence, we use $p(Y)$ to mean both the probability distribution of Y or if Y is known (observed) the probability distribution of Y evaluated at that particular observed value.[1] In the case where we need to distinguish between the two, we write $p(Y = y)$ to mean the probability distribution of the random variable Y evaluated at the value y. Hence, if Y is a discrete random variable, $p(Y = y)$ is the probability of $Y = y$, while if Y is a continuous random variable, $p(Y = y)$ is the probability density function of Y evaluated at y.

There are few basic principles that are used repeatedly in this document:

(1) If the random variables X and Y are independent, then $p(X, Y) = p(X)p(Y)$.

(2) Given the joint distribution of X and Y, the *marginal* distribution of Y is given by integrating over X,

$$p(Y) = \int_{\mathcal{X}} p(dX, Y), \quad \text{where } X \in \mathcal{X}.$$

If X is a discrete random variable with possible values $x_1, \ldots, x_n$, then $p(Y) = \sum_{i=1}^{n} p(X = x_i, Y)$.

(3) We have the following relationship between the joint distribution, the conditional distribution, and the marginal distribution:

$$p(X, Y) = p(X \mid Y)p(Y) = p(Y \mid X)p(X).$$

[1] This is a slight abuse of notation, but has become an accepted practice in statistical literature, particularly in Bayesian text.

1.2 Bayes' Theory

Reverend Thomas Bayes' (1702–1761) theory simply states how one can relate the probability of an event X occurring, conditionally on the fact that an another event Y has occurred, to the probability of event Y occurring, conditionally on the fact that event X has occurred. Bayes' theory can be written as

$$p(X \mid Y) = \frac{p(Y \mid X)p(X)}{p(Y)} \propto p(Y \mid X)p(X).$$

In above, one can think of X as representing possible model configurations (parameters) and Y as observed data. Then $p(Y \mid X)$ describes, in a probabilistic way how the observed data Y is linked to a given model configuration X, and is often referred to as the *likelihood* or the *data model.* The distribution $p(X)$ is referred to as the *prior distribution*, describing in a probabilistic way possible model configurations X prior to seeing the data Y. The end result is the *posterior distribution* of X given the data Y, $p(X \mid Y)$, which describes possible model configurations given (conditional on) the observed data. Given the posterior distribution, one can plot it (particularly if X is one or two dimensional variable) or compute summary statistics for the distribution. Popular statistics include:

$$\text{Mean:} \quad E(X \mid Y) = \int_{\mathcal{X}} X p(dX \mid Y).$$

$$\text{Variance:} \quad \operatorname{var}(X \mid Y) = \int_{\mathcal{X}} (X - E(X \mid Y))^2 p(dX \mid Y).$$

$$\text{Mode:} \quad \arg\max_X p(X \mid Y).$$

One can contrast Bayes' theory to the more classical approach for inference, where X is thought to be an unknown *deterministic* parameter and often *estimated* using, for example maximum likelihood;

$$\hat{X} = \text{the value of } X \text{ that maximizes } p(Y \mid X).$$

This gives a single best model configuration that is in compliance with the data (as judged by $p(Y \mid X)$), while the posterior distribution $p(X \mid Y)$ assigns a probability density over the different model configurations based on their compliance to the observed data and our prior knowledge of X.

1.3 Examples

We shall now give few examples contrasting the classical and Bayesian approach.

Discrete Probability Space

Assume that our (unknown) state-of-the-system parameter X can only take N different, but known values; say $x_1, \ldots, x_N$. From n independent experiments we observe the data $y_1, \ldots, y_n$. The data is assumed to be related to the unknown system parameter X through a (conditional) probabilistic data model,

$$p(Y_1 = y_i \,|\, X)\,; \quad i = 1, \ldots n,$$

where Y_i is a random variable representing the outcome of the i-th experiment. Due to the independence of the n experiments, the joint distribution of the data, given the state of the system, is

$$p(\mathbf{Y} = \mathbf{y} \,|\, X) = \prod_{i=1}^{n} p(Y_1 = y_1 \,|\, X)$$

where $\mathbf{Y} = (Y_1, \ldots, Y_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$. Given a prior distribution on X, $p(X = x_j)$; $j = 1, \ldots, N$, the posterior probability distribution of X is given by

$$p(X = x_j \,|\, \mathbf{Y} = \mathbf{y}) = \frac{p(\mathbf{Y} = \mathbf{y} \,|\, X = x_j) p(X = x_j)}{\sum_{k=1}^{N} p(\mathbf{Y} = \mathbf{y} \,|\, X = x_k) p(X = x_k)}\,; \quad j = 1, \ldots, N,$$

which is easily computed if one can evaluate $p(Y_i = y_i \,|\, X = x_j)$ and $p(X = x_j)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, N$. Further, if one has very little information a *priori* about which state the system is in, an ideal non-informative prior distribution for X is $p(X = x_j) = 1/N$; $j = 1, \ldots, N$. This prior distribution yields

$$p(X = x_j \,|\, \mathbf{Y} = \mathbf{y}) \propto p(\mathbf{Y} = \mathbf{y} \,|\, X = x_j)\,; \quad j = 1, \ldots, N.$$

The maximum likelihood (ML) estimator of X is given by the state

$$\hat{x} = \arg \max_{x \in \{x_1, \ldots, x_N\}} p(\mathbf{Y} = \mathbf{y} \,|\, X = x).$$

Hence, the ML estimator is the posterior mode (the value x that gives the highest posterior probability) when X is given a non-informative prior distribution.

Gaussian Distributed Measurements

Assume we have the data $y_1, \ldots, y_n$ that are independently distributed according to a Gaussian (normal) distribution with mean μ and variance σ^2;

$$y_i \sim \text{Gau}(\mu, \sigma^2), \text{ independently for } i = 1, \ldots, n,$$

where "$\sim$ Gau(μ, σ^2)" reads "distributed as Gaussian with mean μ and variance σ^2". Assume further that the variance σ^2 is known, but the mean parameter μ

is unknown and our goal is to conduct inference on μ given the data $y_1, \ldots, y_n$. Classical statistical analysis gives the ML estimator of μ as

$$\hat{\mu} = \bar{y}, \text{ where } \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i.$$

In the Bayesian framework, assume we assign μ the prior distribution

$$\mu \sim \text{Gau}(\xi, \tau^2), \; \xi \text{ and } \tau^2 \text{ known and given.}$$

The posterior distribution of μ, $p(\mu \,|\, \mathbf{y})$, can be shown to be $\text{Gau}(M, V)$ with

$$M = \left(\frac{\bar{y}}{\sigma^2/n} + \frac{\xi}{\tau^2}\right) V \quad \text{and} \quad V = \left(\frac{1}{\sigma^2/n} + \frac{1}{\tau^2}\right)^{-1}.$$

The posterior mean can be seen to be a weighted average of the empirical average $\bar{y}$ and the prior mean ξ. Note as n gets large (more data sampled), M gets closer to $\bar{y}$, the ML estimator of μ, and the posterior variance gets closer to σ^2/n (which is the variance of $\bar{y}$). Similarly, as one lets τ^2 grow larger (yielding effectively a non-informative prior for μ), the same effect is seen.

Numerical (Physical) Model

Assume we have a deterministic numerical (physical) model that predicts n different numerical quantities. Let

$$\big(F_1(\theta), \ldots, F_n(\theta)\big) = \mathbf{F}(\theta),$$

be the n predicted output quantities from the numerical model when configured according to the parameter θ. An experiment is conducted that gives (observed) measurements $y_1, \ldots, y_n$ of the quantities that the numerical model $F(\cdot)$ aims at predicting. The observed data is assumed to be related to the model predictions as follows,

$$y_i = F_i(\theta) + \varepsilon_i \,; \quad i = 1, \ldots, n,$$

where $\varepsilon_1, \ldots, \varepsilon_n$ are independent Gaussian distributed measurement errors with with zero mean and a known variance σ^2. The data model above can also be written as

$$y_i \sim \text{Gau}(F_i(\theta), \sigma^2)\,; \quad i = 1, \ldots, n,$$

yielding a data model $p(y_i \,|\, \theta)$ that is a Gaussian distribution with mean $F_i(\theta)$ and variance σ^2.

The ML estimator of θ is given by

$$\hat{\theta} = \arg\max_{\theta} p(\mathbf{y} \,|\, \theta),$$

where

$$p(\mathbf{y}\,|\,\theta) = \prod_{i=1}^{n} p(y_i\,|\,\theta).$$

Depending on how computationally involved the numerical model is, and on the dimension of θ, the above (global) optimization can be difficult to carry out.

Given a prior distribution on θ, $p(\theta)$, the posterior distribution of θ is given by

$$p(\theta\,|\,\mathbf{y}) = \frac{p(\mathbf{y}\,|\,\theta)p(\theta)}{p(\mathbf{y})},$$

where

$$p(\mathbf{y}) = \int p(\mathbf{y}\,|\,\theta)p(d\theta).$$

Again, depending on how computationally involved $F(\cdot)$ is and on the dimensionality of θ, evaluating (numerically) the above integral can be prohibitively expensive. Instead of trying to evaluate the integral, an alternative approach is to generate a collection of realizations from the posterior distribution and use these samples to conduct inference (i.e., compute the mean, variance, etc., of the posterior distribution of θ). Indeed, that is the focus of the remaining portion of this report for the case where the posterior distribution of interest is of a particular dynamic form.

2 Dynamic Bayesian Models

We shall now focus on a particular class of probability models that are dynamic by nature. For this class of models the parameter space of interest is *expanding* with time while more data is gathered. Hence, as each new batch of data arrives our goal is to carry out, or rather update our current probabilistic knowledge of the system, which at that point includes both "old" and "new" parameters. A good introduction to dynamic models is "Bayesian Forecasting and Dynamic Models" by West & Harrison (1997).

2.1 The Basic Definition

Denote by

$\boldsymbol{\theta}_t$ the collection of model parameters associated with time t.

$\mathbf{y}_t$ the collection of (potential) data available at time t.

The relationship between the data and the model parameters is described probabilistically by the time-evolving data-model (the likelihood),

$$p(\mathbf{y}_t \,|\, \boldsymbol{\theta}_{1:t})\,; \quad t = 1, 2, \ldots, \tag{1}$$

where we have, without loss of generality, assumed discrete and equal-spaced time points, and where

$$\boldsymbol{\theta}_{1:t} \equiv (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_t).$$

Note that the observed data $\mathbf{y}_t$ do not only depend on the parameters at time t, $\boldsymbol{\theta}_t$, but on the whole time history, $\boldsymbol{\theta}_{1:t}$. The joint distribution of all data observed up to and including time t is given by

$$p(\mathbf{y}_{1:t} \,|\, \boldsymbol{\theta}_{1:t}) = \prod_{t'=1}^{t} p(\mathbf{y}_{t'} \,|\, \boldsymbol{\theta}_{1:t'}), \tag{2}$$

where $\mathbf{y}_{1:t} \equiv (\mathbf{y}_1, \ldots, \mathbf{y}_t)$.

For Bayesian inference a prior distribution is specified for the model parameters $\boldsymbol{\theta}_1$, …, $\boldsymbol{\theta}_t$. Taking advantage of the dynamic nature of the model, the prior distribution can be written as

$$p(\boldsymbol{\theta}_{1:t}) = p(\boldsymbol{\theta}_1)p(\boldsymbol{\theta}_2 \,|\, \boldsymbol{\theta}_1) \cdots p(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{1:t-1}), \tag{3}$$

where the prior distribution of the model parameters at each time point is specified conditional on the model parameters from the previous time points.

We can summarize our dynamic model as:

$$\begin{aligned} \text{Data-Model:} &\quad p(\mathbf{y}_t \,|\, \boldsymbol{\theta}_{1:t}) \\ \text{Parameter-Model:} &\quad p(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{1:t-1}), \end{aligned} \tag{4}$$

along with the initial prior distribution $p(\boldsymbol{\theta}_1)$; $t = 1, 2, \ldots$. It should be noted that both the data model and the parameter model in (4) can also condition on past data, yielding the more general model:

$$\begin{aligned} \text{Data-Model:} \quad & p(\mathbf{y}_t \,|\, \boldsymbol{\theta}_{1:t}, \mathbf{y}_{1:t-1}) \\ \text{Parameter-Model:} \quad & p(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{1:t-1}, \mathbf{y}_{1:t-1}). \end{aligned}$$

However, for the remaining of this document we shall assume (4), but the results presented do apply to the more general model above.

Our goal is to conduct posterior inference on $\boldsymbol{\theta}_{1:t}$ as time evolves and more data is gathered. Bayes' theory gives the posterior distribution at time t as

$$\pi_t(\boldsymbol{\theta}_{1:t}) \equiv p(\boldsymbol{\theta}_{1:t} \,|\, \mathbf{y}_{1:t}) \propto p(\mathbf{y}_{1:t} \,|\, \boldsymbol{\theta}_{1:t}) p(\boldsymbol{\theta}_{1:t}). \tag{5}$$

Using the product form of the likelihood in (2) and the dynamic nature of the prior in (3), we can write the posterior as (or rather, proportional to) the following product,

$$\pi_t(\boldsymbol{\theta}_{1:t}) \propto \left(\prod_{t'=1}^{t} p(\mathbf{y}_{t'} \,|\, \boldsymbol{\theta}_{1:t'}) \right) \left(\prod_{t'=1}^{t} p(\boldsymbol{\theta}_{t'} \,|\, \boldsymbol{\theta}_{1:t'-1}) \right) = \prod_{t'=1}^{t} p(\mathbf{y}_{t'} \,|\, \boldsymbol{\theta}_{1:t'}) p(\boldsymbol{\theta}_{t'} \,|\, \boldsymbol{\theta}_{1:t'-1}), \tag{6}$$

where we for convenience define $\boldsymbol{\theta}_{1:0} = \emptyset$ (an empty set of parameters), so that $p(\boldsymbol{\theta}_1 \,|\, \boldsymbol{\theta}_{1:0}) = p(\boldsymbol{\theta}_1)$. The above expression for the posterior hints at an alternative, sequential expression for the posterior distribution at time t, that is based on "updating" the posterior distribution from the previous time point, $t-1$;

$$\pi_t(\boldsymbol{\theta}_{1:t}) \propto \big(p(\mathbf{y}_t \,|\, \boldsymbol{\theta}_{1:t}) p(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{1:t-1}) \big)\, \pi_{t-1}(\boldsymbol{\theta}_{1:t-1}). \tag{7}$$

One can also derive this posterior updating expression from a purely statistical argument as follows: Given $\pi_{t-1}(\boldsymbol{\theta}_{1:t-1})$, our prior knowledge of $\boldsymbol{\theta}_{1:t}$ at time t based on all data up to and including time $t-1$ is given by the distribution

$$\begin{aligned} \pi_{t-1}(\boldsymbol{\theta}_{1:t}) \equiv p(\boldsymbol{\theta}_{1:t} \,|\, \mathbf{y}_{1:t-1}) &= p(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{1:t-1}, \mathbf{y}_{1:t-1}) p(\boldsymbol{\theta}_{1:t-1} \,|\, \mathbf{y}_{1:t-1}) \\ &= p(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{1:t-1}) \pi_{t-1}(\boldsymbol{\theta}_{1:t-1}), \end{aligned}$$

where we used that $\boldsymbol{\theta}_t$ is independent of the data $\mathbf{y}_{1:t-1}$ given the parameter history $\boldsymbol{\theta}_{1:t-1}$ (i.e., $p(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{1:t-1}, \mathbf{y}_{1:t-1}) = p(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{1:t-1})$). Using this, we can write the posterior at time t as

$$\pi_t(\boldsymbol{\theta}_{1:t}) = p(\mathbf{y}_t \,|\, \boldsymbol{\theta}_{1:t}) \pi_{t-1}(\boldsymbol{\theta}_{1:t}).$$

Although one can write down the posterior distribution up to a proportionality constant at each given time point t, using it for inference is altogether another problem. Computing the proportionality constant can be prohibitively difficult as it involves a numerical multi-dimensional integral (integrating $p(\mathbf{y}_{1:t} \,|\, \boldsymbol{\theta}_{1:t}) p(\boldsymbol{\theta}_{1:t})$ with respect to $\boldsymbol{\theta}_{1:t}$). An alternative is to sample (i.e., generate) realizations from the (unscaled) posterior distribution and use them for inference (i.e., computing means, variances, quantiles, etc.). Even if we could compute the missing proportionality constant, sampling based inference is often the only viable option in summarizing the posterior distribution, especially in high dimensional settings. This is what we shall explore in Section 3 and 4, and in particular how one can construct a sampling procedure that samples from $\pi_1(\boldsymbol{\theta}_1)$, $\pi_2(\boldsymbol{\theta}_{1:2})$, $\ldots$, in a sequential effective way, taking advantage of the dynamic nature of the posterior in (7)

2.2 Example: Target Tracking

A classical example of a dynamic model is 2D target tracking. The goal is to track a moving target and report on its location, $\mathbf{x}_t = (x_{1t}, x_{2t})$, and velocity, $\mathbf{v}_t = (v_{1t}, v_{2t})$, at discrete time points $t = 1, 2, \ldots$. A simple dynamic model for $\boldsymbol{\theta}_t = (\mathbf{x}_t, \mathbf{v}_t)$ is given by

$$\begin{aligned}\mathbf{x}_t &= \mathbf{x}_{t-1} + 0.5(\mathbf{v}_{t-1} + \mathbf{v}_t)\\ \mathbf{v}_t &= \mathbf{v}_{t-1} + \boldsymbol{\delta}_t,\end{aligned}$$

which linearly interpolates the velocity vector between time $(t-1)$ and t and assumes an *auto-regressive* model for the velocity vector, where the rate-of-change (the acceleration) $\boldsymbol{\delta}_t$ is assumed Gaussian with mean zero and known variance-covariance matrix $\mathbf{W}$. The model can also be written in the matrix form

$$\begin{bmatrix}\mathbf{x}_t\\ \mathbf{v}_t\end{bmatrix} = \begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}\begin{bmatrix}\mathbf{x}_{t-1}\\ \mathbf{v}_{t-1}\end{bmatrix} + \begin{bmatrix}0.5\\ 1\end{bmatrix}\boldsymbol{\delta}_t.$$

For simplicity, assume that the target tracking data consists of (noise corrupted) position observations, $\mathbf{y}_1, \mathbf{y}_2, \ldots$, that are related to the actual location of the target via

$$\mathbf{y}_t = \mathbf{x}_t + \boldsymbol{\varepsilon}_t,$$

where $\boldsymbol{\varepsilon}_t$ is a zero mean Gaussian measurement error with variance-covariance matrix $\mathbf{V}$.

Given an initial Gaussian prior distribution for $(\mathbf{x}_1, \mathbf{v}_1)$, a closed-form solution (the *Kalman-filter*) exists for updating the posterior distribution at time $(t-1)$ to yield the posterior distribution at time t in the form of a Gaussian distribution (see e.g., West & Harrison, 1997, for an overview). This result depends on the linearity of both the measurement model and the dynamic model for $(\mathbf{x}_t, \mathbf{v}_t)$, along with the Gaussian assumption made about the measurement errors and the acceleration of the target.

It is relatively easy to extend the above simple target tracking scenario to a more complicated one, where the observed tracking data are not directly (linearly) related to the location of the target and the maneuvering model for the target is much more complicated. For such a general model, a closed-form solution for updating the posterior distribution at each time t is seldom available and one needs to resort to sampling-based methods for posterior inference.

2.3 Example: Atmospheric Dispersion Modeling with Unknown Source Characteristics

The goal here is to estimate (probabilistically) the location and release rate history of a contaminant into the atmosphere using a numerical atmospheric contaminant dispersion model and relatively few concentration measurements at given sensor locations. In this simple scenario let

$\mathbf{x}_t \in \mathbb{R}^3$ be the location of a point source in the time interval $(t-1, t]$.

$s_t \in \mathbb{R}^+$ be the source strength (release rate) in the time interval $(t-1, t]$.

$\boldsymbol{\theta}_t \equiv (\mathbf{x}_t, s_t)$.

Given the source history $\boldsymbol{\theta}_{1:t} = (\mathbf{x}_{1:t}, \mathbf{s}_{1:t})$ we use a rather simple Gaussian puff model, INPUFF (Petersen & Lavdas, 1986), to predict the resulting concentration of the contaminant. Let

$\hat{C}(\mathbf{x}', t') = \hat{C}(\mathbf{x}', t'; \boldsymbol{\theta}_{1:t'})$ be the model predicted contaminant average concentration in $(t'-1, t']$ at location $\mathbf{x}'$ due to a source with release history given by $\boldsymbol{\theta}_{1:t'}$.

For the dispersion model in question, the predicted concentration $\hat{C}(\mathbf{x}', t')$ can be broken down into additive contributions from each time interval,

$$\hat{C}(\mathbf{x}', t') = \sum_{t=1}^{t'} \hat{G}_{\mathbf{x}_t, t}(\mathbf{x}', t') s_t, \tag{8}$$

where

$\hat{G}_{\mathbf{x},t}(\mathbf{x}', t')$ gives the predicted average concentration in $(t'-1, t]$ at $\mathbf{x}'$ due to a source at location $\mathbf{x}$ with a release rate of 1 in $(t-1, t]$ (and zero outside $(t-1, t]$).

The observed data is assumed to consist of time-averaged concentration measurements at given sensor (monitor) sites. Assuming a network of M sensors at locations $\mathbf{m}_1, \ldots, \mathbf{m}_M$, let

$c_{j,t}$ = the average observed concentration from the j-th sensor in the time interval $(t-1, t]$.

The observed data is then assumed to be related to the predicted concentration via the simple data model

$$p(c_{j,t} \,|\, \hat{C}(\mathbf{m}_j, t)) = \text{Gau}(\hat{C}(\mathbf{m}_j, t), V(\hat{C}(\mathbf{m}_j, t))\big|_0^{\infty}, \tag{9}$$

where $\text{Gau}(\mu, \sigma^2)\big|_l^u$ denotes a Gaussian (Normal) density with mean μ and variance σ^2 and truncated between l and u ($l < u$), and $V(\cdot)$ is a known variance function.

The model is then fully specified by giving a prior distribution for the source location and the release rate history. We shall assume that the source is not moving, $\mathbf{x}_t = \mathbf{x}$, but little is know about it's location. We therefore assign a non-informative prior to the location,

$$p(\mathbf{x}) \propto 1 \text{ if } \mathbf{x} \in \mathcal{X}, \, 0 \text{ otherwise},$$

where $\mathcal{X}$ is the spatial domain of interest. The source release is assumed to start at an unknown time $t^* \geq 1$ with a vague information of the initial release rate, but is then assumed to change "smoothly" as time progresses. We formulate this prior information as following:

$$p(t^*) = \begin{cases} 1/t^*_{\max} & \text{if } t^* \in \{1, \ldots, t^*_{\max}\}, \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

That is, a flat prior on the initial start-time between $t^* = 1$ and $t^* = t^*_{\max}$. For the initial release rate, we assume that

$$p_{t^*}(s_{t^*}) = f_1(s_{t^*}) \tag{11}$$

where $f_1(\cdot)$ is a given prior distribution on positive release and note that $s_1 = \cdots = s_{t^*-1} = 0$. And finally for $t > t^*$, we assume that

$$p(s_t \,|\, s_{t-1}) = f_2(s_t \,|\, s_{t-1}) \tag{12}$$

where $f_2(\cdot \,|\, \cdot)$ is a conditional distribution. An example of f_1 and f_2 are:

$$\begin{aligned} f_1(\cdot) &= \text{Gau}(\mu_1, \sigma_1^2)\big|_0^{c^+} \\ f_2(\cdot \,|\, s_{t-1}) &= \text{Gau}(s_{t-1}, \sigma_2^2)\big|_0^{c^+}, \end{aligned}$$

where the parameters μ_1, σ_1^2, and σ_2^2 are assumed known and recall that $\text{Gau}(\cdot, \cdot)\big|_0^{c^+}$ denotes a truncated Gaussian distribution.

Due to how complicated the model is, particularly the dependence of the dispersion model on the location parameter $\mathbf{x}$, sampling-based methods need to be used for posterior inference at each time point t. And this is what we shall now study for the dynamic model in general.

3 Markov Chain Monte Carlo (MCMC)

We shall now give a review of the well established Markov chain Monte Carlo (MCMC) approach for generating realizations from the posterior distribution $\pi_t(\boldsymbol{\theta}_{1:t})$ in (7); $t = 1, 2, \ldots$. A good practical introduction to MCMC is the volume "Markov Chain Monte Carlo in Practice", edited by Gilks et al. (1996), the book "Monte Carlo Strategies in Scientific Computing" by Liu (2001), and the overview paper by Andrieu et al. (2003).

Our basic goal is to generate realizations, $\boldsymbol{\theta}_{1:t}^{(1)}, \ldots, \boldsymbol{\theta}_{1:t}^{(N)}$ from the posterior distribution $\pi_t(\boldsymbol{\theta}_{1:t})$ in (7) for $t = 1, 2, \ldots$. All inference are then conducted using these realizations. That is, for example if $Q(\boldsymbol{\theta}_{1:t})$ is a function of the unknown parameters, then its posterior expected value,

$$E(Q(\boldsymbol{\theta}_{1:t}) \,|\, \mathbf{y}_{1:t}) \equiv \int Q(\boldsymbol{\theta}_{1:t}) \pi_t(d\boldsymbol{\theta}_{1:t}),$$

is approximated by

$$\hat{E}(Q(\boldsymbol{\theta}_{1:t}) \,|\, \mathbf{y}_{1:t}) \equiv \sum_{i=1}^{N} (1/N) Q(\boldsymbol{\theta}_{1:t}^{(i)}).$$

Basically we have approximated the posterior distribution at time t, $\pi_t(\boldsymbol{\theta}_{1:t})$, by the empirical distribution function,

$$\hat{\pi}_t^N(\boldsymbol{\theta}_{1:t}) = \sum_{i=1}^{N} (1/N) \delta(\boldsymbol{\theta}_{1:t}^{(i)} - \boldsymbol{\theta}_{1:t}), \tag{13}$$

where $\delta(\boldsymbol{\theta}_{1:t}^{(i)} - \boldsymbol{\theta}_{1:t}) = 1$ if $\boldsymbol{\theta}_{1:t}^{(i)} = \boldsymbol{\theta}_{1:t}$, otherwise 0.

3.1 The Basics of MCMC

The MCMC approach has a long and successful history for non-dynamic models, but has been shown to be somewhat less appropriate for dynamic models (in its most general form). However, there are cases when MCMC is well suited for dynamic models, one being when the main interest is on a single time point given fixed set of data and as such the model can simply be treated as static.

The MCMC approach generates realization(s) from a Markov chain that has the posterior distribution $\pi_t(\boldsymbol{\theta}_{1:t})$ as its stationary distribution. This is accomplished by generating the realization $\boldsymbol{\theta}_{1:t}^{(i)}$ using the previous realization, $\boldsymbol{\theta}_{1:t}^{(i-1)}$ along with a probabilistic proposal mechanism that outlines how this is done. One of the most popularized MCMC algorithm to generate a chain of size N from $\pi_t(\boldsymbol{\theta}_{1:t})$ is given in Table 1 and is referred to as the *Metropolis-Hastings* (M-H) MCMC sampling algorithm. The proposal distribution $q_t(\tilde{\boldsymbol{\theta}}_{1:t} \,|\, \boldsymbol{\theta}_{1:t}^{(i)})$ in Step A of the M-H algorithm is specified by the user and can be very general (see Section 3.4). The acceptance ratio in Step B of the M-H algorithm is given by the posterior ratio multiplied by the proposal ratio (or rather divided by the proposal ratio). The inclusion of the proposal ratio is to correct for "bias" in the proposal distribution; note that if the proposal distribution is symmertric (unbiased), that is, $q_t(\tilde{\boldsymbol{\theta}}_{1:t} \,|\, \boldsymbol{\theta}^{(i)}_{1:t}) = q_t(\boldsymbol{\theta}^{(i)}_{1:t} \,|\, \tilde{\boldsymbol{\theta}}_{1:t})$, then the proposal ratio is just equal to 1 and does not enter the expression for the acceptance ratio.

Table 1: Markov Chain Monte Carlo (MCMC) Algorithm:

The following algorithm describes how to generate realizations from $\pi_t(\boldsymbol{\theta}_{1:t})$ for a given t (i.e., at a given time point).

Step 0 (Initialization): A starting value $\boldsymbol{\theta}^{(1)}_{1:t}$ for the Markov chain is proposed.

For $i = 1, \ldots, N-1$: Use Metropolis-Hasting (M-H) sampler:

Step A (Proposal) Given the i-th step of the Markov chain, $\boldsymbol{\theta}^{(i)}_{1:t}$, the next step is proposed via a *proposal distribution*;

$$\tilde{\boldsymbol{\theta}}_{1:t} \sim q_t(\tilde{\boldsymbol{\theta}}_{1:t} \,|\, \boldsymbol{\theta}^{(i)}_{1:t}). \tag{14}$$

Step B (M-H Acceptance Ratio): The acceptance ratio,

$$\rho_t(\tilde{\boldsymbol{\theta}}_{1:t}; \boldsymbol{\theta}^{(i)}_{1:t}) = \frac{\pi_t(\tilde{\boldsymbol{\theta}}_{1:t}) q_t(\boldsymbol{\theta}^{(i)}_{1:t} \,|\, \tilde{\boldsymbol{\theta}}_{1:t})}{\pi_t(\boldsymbol{\theta}^{(i)}_{1:t}) q_t(\tilde{\boldsymbol{\theta}}_{1:t} \,|\, \boldsymbol{\theta}^{(i)}_{1:t})} = \frac{p(\mathbf{y}_{1:t} \,|\, \tilde{\boldsymbol{\theta}}_{1:t}) p(\tilde{\boldsymbol{\theta}}_{1:t}) q_t(\boldsymbol{\theta}^{(i)}_{1:t} \,|\, \tilde{\boldsymbol{\theta}}_{1:t})}{p(\mathbf{y}_{1:t} \,|\, \boldsymbol{\theta}^{(i)}_{1:t}) p(\boldsymbol{\theta}^{(i)}_{1:t}) q_t(\tilde{\boldsymbol{\theta}}_{1:t} \,|\, \boldsymbol{\theta}^{(i)}_{1:t})} \tag{15}$$

is computed, along with the acceptance probability

$$\alpha_t(\boldsymbol{\theta}_{1:t}; \boldsymbol{\theta}^{(i)}_{1:t}) = \min\{\rho_t(\tilde{\boldsymbol{\theta}}_{1:t}; \boldsymbol{\theta}^{(i)}_{1:t}),\ 1\}.$$

Step C (Selection): Generate $u \sim \text{Uniform}[0,1]$ and let

$$\boldsymbol{\theta}^{(i+1)}_{1:t} = \begin{cases} \tilde{\boldsymbol{\theta}}_{1:t} & \text{if } u \leq \alpha_t(\tilde{\boldsymbol{\theta}}_{1:t}; \boldsymbol{\theta}^{(i)}_{1:t}), \\ \boldsymbol{\theta}^{(i)}_{1:t} & \text{otherwise.} \end{cases}$$

The efficiency of the M-H algorithm depends on the "quality" of the proposal distribution in Step A. The proposal distribution can be factored in a fashion similar to the prior distribution given in (3). That is (suppressing the chain index i),

$$q_t(\tilde{\boldsymbol{\theta}}_{1:t} \,|\, \boldsymbol{\theta}_{1:t}) = q_t(\tilde{\boldsymbol{\theta}}_1 \,|\, \boldsymbol{\theta}_{1:t}) q_t(\tilde{\boldsymbol{\theta}}_2 \,|\, \tilde{\boldsymbol{\theta}}_1, \boldsymbol{\theta}_{1:t}) \cdots q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t-1}, \boldsymbol{\theta}_{1:t}).$$

By restricting the proposal of $\tilde{\boldsymbol{\theta}}_{t'}$, $t' = 1, \ldots, t$, to condition only on parameters up to and including time t' (a rather natural restriction given the dynamics of the model), the above proposal distribution can be written as

$$q_t(\tilde{\boldsymbol{\theta}}_{1:t} \,|\, \boldsymbol{\theta}_{1:t}) = \prod_{t'=1}^{t} q_t(\tilde{\boldsymbol{\theta}}_{t'} \,|\, \tilde{\boldsymbol{\theta}}_{1:t'-1}, \boldsymbol{\theta}_{1:t'}), \tag{16}$$

where recall that $\boldsymbol{\theta}_{1:0} = \emptyset$.

A typical MCMC proposal algorithm alternates between different type of proposals in a systematic or random fashion with each proposal only modifying a subset of the parameters. For example, a (sub-)proposal distribution that only modifies the t-th parameter (the last parameter) can be written as

$$q_t(\tilde{\boldsymbol{\theta}}_{1:t} \,|\, \boldsymbol{\theta}_{1:t}) = q_t(\tilde{\boldsymbol{\theta}}_{t} \,|\, \boldsymbol{\theta}_{1:t})\delta(\tilde{\boldsymbol{\theta}}_{1:t-1} - \boldsymbol{\theta}_{1:t-1}).$$

Similar sub-proposal distributions can be created for the other components of the parameter vector, and the proposal step A in the MCMC Algorithm in Table 1 would alternate between different sub-proposals.

The acceptance ratio (15) can be written as the product,

$$\rho_t(\tilde{\boldsymbol{\theta}}_{1:t}; \boldsymbol{\theta}_{1:t}) = \prod_{t'=1}^{t} \left(\frac{p(\mathbf{y}_{t'} \,|\, \tilde{\boldsymbol{\theta}}_{1:t'}) p(\tilde{\boldsymbol{\theta}}_{t'} \,|\, \tilde{\boldsymbol{\theta}}_{1:t'-1}) q_t(\boldsymbol{\theta}_{t'} \,|\, \boldsymbol{\theta}_{1:t'-1}, \tilde{\boldsymbol{\theta}}_{1:t'})}{p(\mathbf{y}_{t'} \,|\, \boldsymbol{\theta}_{1:t'}) p(\boldsymbol{\theta}_{t'} \,|\, \boldsymbol{\theta}_{1:t'-1}) q_t(\tilde{\boldsymbol{\theta}}_{t'} \,|\, \tilde{\boldsymbol{\theta}}_{1:t'-1}, \boldsymbol{\theta}_{1:t'})} \right), \tag{17}$$

using the conditional format of the proposal distribution in (16) and the product format of the posterior in (6). Depending on the proposal distribution, it is not necessarily the case that all the components in the above expression need to evaluated. For example, if the new proposal $\tilde{\boldsymbol{\theta}}_{1:t}$ is such that only changes are made to $\tilde{\boldsymbol{\theta}}_{t''}$, where $t \geq t'' \geq 1$, then only terms with $t' \geq t''$ in the final product in (17) need to be evaluated, the other terms cancel out.

There are two characteristics that determine the effective sample size (the statistical efficiency) of the MCMC realizations $\boldsymbol{\theta}_{1:t}^{(1)}, \ldots, \boldsymbol{\theta}_{1:t}^{(N)}$: the burn-in period and the chain's auto-correlation. The burn-in period represents the number of samples needed at the beginning for the Markov chain to actually reach the state where it is sampling from the target distribution, $\pi_t(\boldsymbol{\theta}_{1:t})$. These initial samples are discarded and not used for inference; hence reducing the effective sample size. The second issue is auto-correlation. Due to the Markovian nature of the algorithm, the realizations $\boldsymbol{\theta}_{1:t}^{(1)}, \ldots, \boldsymbol{\theta}_{1:t}^{(N)}$ are not an independent sample from $\pi_t(\boldsymbol{\theta}_{1:t})$; nearby realizations can be highly correlated. The amount of auto-correlation in the sample depends on how well the proposal distribution is able to "mix" the sample and the acceptance rate associated with the proposal distribution. If the proposal distribution alters the chain too little at each step ($\tilde{\boldsymbol{\theta}}_{1:t}$ too close to $\boldsymbol{\theta}_{1:t}$), the resulting MCMC sample tends to show high auto-correlation. Similarly, a proposal distribution that makes large changes at each step typically has a low acceptance ratio and therefore stays in the same state for a long period of time, which causes high auto-correlation in the final sample. The optimal proposal distribution is somewhere in between, and as a rule of thumb, an acceptance rate around 25% is thought to be good in multi-dimensional problems (Gelman et al., 2004, page 306) (if higher, the proposal distribution is making changes that are too small while if lower, the proposal distribution is making changes that are too big).

The main drawback of the MCMC algorithm for dynamic models is it does not have a natural way of carrying the posterior information available from the sample $\boldsymbol{\theta}_{1:t}^{(1)}, \ldots, \boldsymbol{\theta}_{1:t}^{(N)}$ over to time $t+1$, to generate the sample $\boldsymbol{\theta}_{1:t+1}^{(1)}, \ldots, \boldsymbol{\theta}_{1:t+1}^{(N)}$. At time $t+1$ one would simply start a new Markov chain, with $\pi_{t+1}(\boldsymbol{\theta}_{1:t+1})$ as its targeting distribution, without taking any direct advantage of the sequential nature of the posterior distribution at time $t+1$, as given by (7). There is one exception to this that applies to a particular MCMC algorithm, an algorithm that rejuvenates and extends the MCMC realizations from the previous time point, which we shall now describe.

3.2 Sequential MCMC via Rejuvenation and Extension

We shall now give a short account of a particular MCMC algorithm that takes advantage of the MCMC realizations from previous time step. We shall see later that this MCMC algorithm mirrors (and in many ways inspires) a very similar Sequential Monte Carlo (SMC) algorithm; see Section 4.3.

Assume at time $t-1$ we have an MCMC sample $\boldsymbol{\theta}_{1:t-1}^{(1)}, \ldots, \boldsymbol{\theta}_{1:t-1}^{(N)}$ from $\pi_{t-1}(\boldsymbol{\theta}_{1:t-1})$[2]. Using this sample we derive the following approximation (as in (13)),

$$\pi_{t-1}(\boldsymbol{\theta}_{1:t-1}) \simeq \hat{\pi}_{t-1}^{N}(\boldsymbol{\theta}_{1:t-1}) \equiv \sum_{i=1}^{N} (1/N)\delta(\boldsymbol{\theta}_{1:t-1} - \boldsymbol{\theta}_{1:t-1}^{(i)}).$$

By plugging this approximation in place of $\pi_{t-1}(\boldsymbol{\theta}_{1:t-1})$ in (7), we derive the following approximation to the posterior at time t,

$$\begin{aligned}\pi_t(\boldsymbol{\theta}_{1:t}) &\simeq C \times p(\mathbf{y}_t \,|\, \boldsymbol{\theta}_{1:t}) p(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{1:t-1}) \sum_{i=1}^{N} (1/N)\delta(\boldsymbol{\theta}_{1:t-1} - \boldsymbol{\theta}_{1:t-1}^{(i)}) \\ &= C \times \sum_{i=1}^{N} p(\mathbf{y}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(i)}, \boldsymbol{\theta}_t) p(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(i)}) (1/N)\delta(\boldsymbol{\theta}_{1:t-1} - \boldsymbol{\theta}_{1:t-1}^{(i)}),\end{aligned} \tag{18}$$

where C is an unknown normalizing constant. The approach we take here is to generate samples from the approximation above instead of $\pi_t(\boldsymbol{\theta}_{1:t})$. By taking this approach, we have restricted the to-be-generated realizations from the posterior at time t to be of the form $\boldsymbol{\theta}_{1:t} = (\boldsymbol{\theta}_{1:t-1}^{(I)}, \boldsymbol{\theta}_t)$, where $\boldsymbol{\theta}_{1:t-1}^{(I)}$, $I \in \{1, \ldots, N\}$, is a

[2] Assume also that this sample has been corrected for a burn-in period

realization from the posterior at time $t-1$. Hence, we are simply rejuvenating and extending the past realizations based on the information content of the new data, $\mathbf{y}_t$. The drawback of this approach is that if the new data is highly informative and not very much in line with what the previous data have indicated, the past posterior sample might not be rich enough (e.g., not large enough) to include a sufficient number of past realizations that are in a good agreement with the new data. Hence, taking this approach usually requires a large number of MCMC realizations (a large N), and even if that is satisfied, it often yields an impoverished sample for conducting inference on $\boldsymbol{\theta}_{t'}$ when $t-t'$ is large.

A well known trick to sample from a *mixture* of distributions, like the one in (18), is to augment the parameter space to include the mixture index; work with $(\boldsymbol{\theta}_{1:t}, I)$ instead of only $\boldsymbol{\theta}_{1:t}$ where $I \in \{1, \ldots, N\}$ is the mixture component index. Define the following two distributions associated with the augmented parameter $(\boldsymbol{\theta}_{1:t}, I)$:

$$\begin{aligned}\pi_t^a(\boldsymbol{\theta}_{1:t} \,|\, I) &\equiv C \,\times\, p(\mathbf{y}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(I)}, \boldsymbol{\theta}_t) p(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(I)}) \delta(\boldsymbol{\theta}_{1:t-1} - \boldsymbol{\theta}_{1:t-1}^{(I)}),\\ \pi_t^a(I) &\equiv 1/N\,; \quad I = 1, \ldots, N.\end{aligned}$$

The joint distribution of the augmented parameter $(\boldsymbol{\theta}_{1:t}, I)$ is then

$$\begin{aligned}\pi_t^a(\boldsymbol{\theta}_{1:t}, I) &= \pi_t^a(\boldsymbol{\theta}_{1:t} \,|\, I)\pi_t^a(I)\\ &= C \,\times\, p(\mathbf{y}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(I)}, \boldsymbol{\theta}_t) p(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(I)}) \delta(\boldsymbol{\theta}_{1:t-1} - \boldsymbol{\theta}_{1:t-1}^{(I)})(1/N),\end{aligned}$$

and in particular, the marginal distribution of $\boldsymbol{\theta}_{1:t}$ with respect to $\pi_t^a(\boldsymbol{\theta}_{1:t}, I)$ is

$$\pi_t^a(\boldsymbol{\theta}_{1:t}) = \sum_{I=1}^{N} \pi_t^a(\boldsymbol{\theta}_{1:t} \,|\, I)\pi_t^a(I) = \text{the mixture in (18).}$$

This suggests that one could construct a MCMC algorithm to sample from $\pi_t^a(\boldsymbol{\theta}_{1:t}, I)$ and then simply drop the index I, yielding a sample from the above marginal distribution which is equal to the target mixture distribution in (18). The proposal for this augmented approach (i.e., Step A in Table 1) would be:

Step A (Augmented Proposal)

(1) Sample $\tilde{I} \sim \pi_t^a(\tilde{I})$ = a uniform distribution on $\{1, \ldots, N\}$.

(2) Sample $\tilde{\boldsymbol{\theta}}_t \sim q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}, \mathbf{y}_t)$.

(3) Let $\tilde{\boldsymbol{\theta}}_{1:t} \equiv (\boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}, \tilde{\boldsymbol{\theta}}_t)$, and the augmented proposal is $(\tilde{\boldsymbol{\theta}}_{1:t}, \tilde{I})$.

What is particularly noticeable about the above augmented proposal is it does not depend on $\boldsymbol{\theta}_{1:t}^{(i)}$, the previous realization from the Markov chain. Proposal distributions that have this feature are often referred to as independent M-H proposals.

Since the proposals are independent they can be made in a parallel fashion, as N independent processes. The augmented proposal can be made slightly more general by replacing step **(1)** by the following:

(1') Sample $\tilde{I} \sim q_t(\tilde{I} \,|\, \mathbf{y}_t)$, a discrete proposal distribution on $\{1, \ldots, N\}$.

Note that this proposal distribution depends on the new data $\mathbf{y}_t$, allowing for the possibility of using the new data to see which realizations from time $t-1$ are more fit to be extended to time t and which are not.

The acceptance ratio (Step B in Table 1) for the augmented proposal is given by:

Step B (Augmented M-H Acceptance Ratio) Let $(\boldsymbol{\theta}_{1:t}^{(i)}, I^{(i)}) = (\boldsymbol{\theta}_{1:t-1}^{(I^{(i)})}, \boldsymbol{\theta}_t^{(i)}, I^{(i)})$ be the previous sample from the Markov chain, then

$$\rho_t(\tilde{\boldsymbol{\theta}}_{1:t}, \tilde{I}; \boldsymbol{\theta}_{1:t}^{(i)}, I^{(i)}) = \frac{p(\mathbf{y}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}, \tilde{\boldsymbol{\theta}}_t) p(\tilde{\boldsymbol{\theta}}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})})\, q_t(\boldsymbol{\theta}_t^{(i)} \,|\, \boldsymbol{\theta}_{1:t-1}^{(I^{(i)})}, \mathbf{y}_t)}{p(\mathbf{y}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(I^{(i)})}, \boldsymbol{\theta}_t^{(i)}) p(\boldsymbol{\theta}_t^{(i)} \,|\, \boldsymbol{\theta}_{1:t-1}^{(I^{(i)})})\, q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}, \mathbf{y}_t)}.$$

Due to the augmented independent M-H sampler, the above acceptance ratio does not include any mixed terms, terms that include both components from the next proposed state, $(\tilde{\boldsymbol{\theta}}_{1:t}, \tilde{I})$, and the current state, $(\boldsymbol{\theta}_{1:t}^{(i)}, I^{(i)})$. Even though this is the case, the acceptance process can not made in parallel, as N independent processes, as in the proposal step[3]. This is due to what seems to be a rather random use of the i-th sample to compute the acceptance ratio for the new proposal, and therefore influencing if the new state will be accepted or not; recall that the i-th sample had no impact on how the new proposal was generated! One can therefore ask if it is possible to "adapt" this particular MCMC algorithm such that in can be easily conducted in parallel? The answer to that is Sequential Monte Carlo (SMC), which we shall review in Section 4.

3.3 Sequential MCMC via Rejuvenation, Modification, and Extension

What follows is an outline of how one could modify the above approach to also propose changes in the parameter history (i.e., propose changes to $\boldsymbol{\theta}_{1:t-1}$), not simply rejuvenate and extend the previous realizations to time t. However, this extension results in complications that might in some cases reduce its usefulness.

We replace the augmented proposal step from previous section with the following step:

[3] Although, one could compute in parallel $p(\mathbf{y}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}, \tilde{\boldsymbol{\theta}}_t)$, $p(\tilde{\boldsymbol{\theta}}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})})$, and $q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}, \mathbf{y}_t)$ for all the N different proposals that can be made in parallel (i.e., at the same time as the proposals are made), and then use to compute the acceptance ratio when needed.

Step A (Augmented Proposal 2)

(1) Sample $\tilde{I} \sim q_t(\tilde{I} \,|\, \mathbf{y}_t)$, a distribution on $\{1, \ldots, N\}$.

(2a) Sample $\tilde{\boldsymbol{\theta}}_{1:t-1} \sim q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} \,|\, \boldsymbol{\theta}^{(\tilde{I})}_{1:t-1}, \mathbf{y}_t)$.

(2b) Sample $\tilde{\boldsymbol{\theta}}_t \sim q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t-1}, \mathbf{y}_t)$.

(3) Let $\tilde{\boldsymbol{\theta}}_{1:t} \equiv (\tilde{\boldsymbol{\theta}}_{1:t-1}, \tilde{\boldsymbol{\theta}}_t)$.

This version of the augmented proposal step both modifies the past (selected) realization $\boldsymbol{\theta}^{(\tilde{I})}_{1:t-1}$ and extents it to time t. The proposal distribution in step (2a) can be taken to be of the sequential form,

$$q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} \,|\, \boldsymbol{\theta}^{(\tilde{I})}_{1:t-1}, \mathbf{y}_t) = \prod_{t'=1}^{t-1} q_t(\tilde{\boldsymbol{\theta}}_{t'} \,|\, \tilde{\boldsymbol{\theta}}_{1:t'-1}, \boldsymbol{\theta}^{(\tilde{I})}_{1:t'}, \mathbf{y}_t).$$

Typically we aim only at changing relatively few parameters associated with $\boldsymbol{\theta}^{(\tilde{I})}_{1:t-1}$ in the proposal (those parameters that are believed to have the largest impact on the newly observed data $\mathbf{y}_t$). As such, many of the sub-proposal distributions above put $\tilde{\boldsymbol{\theta}}_{t'} = \boldsymbol{\theta}^{(\tilde{I})}_{t'}$ with probability 1.

The main difference (and added complexity) of this approach versus the previous approach that did not modify the past, is in computing the acceptance ratio ρ. Instead of computing the acceptance ratio with respect to the mixture approximation in (18), yielding a approximate sample from the posterior, we compute the acceptance ratio with respect to the true posterior, hence yielding a sample from the exact posterior distribution. That is, the mixture approximation is only used to construct the proposal. We use therefore (15), or (17), to compute the acceptance ratio, with $q_t(\tilde{\boldsymbol{\theta}}_{1:t} \,|\, \boldsymbol{\theta}^{(i)}_{1:t})$ in (15) given by

$$q_t(\tilde{\boldsymbol{\theta}}_{1:t} \,|\, \boldsymbol{\theta}^{(i)}_{1:t}) = q_t(\tilde{\boldsymbol{\theta}}_{1:t}) = q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t-1}, \mathbf{y}_t) \sum_{\tilde{I}=1}^{N} q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} \,|\, \boldsymbol{\theta}^{(\tilde{I})}_{1:t-1}, \mathbf{y}_t) q_t(\tilde{I} \,|\, \mathbf{y}_t).$$

A few comments on evaluating the proposal ratio (15). Since the proposal is derived by modifying a realization from time $t-1$, some of the likelihood and prior calculations involved have already been carried out at time $t-1$. Secondly, evaluating the proposal distribution $q_t(\tilde{\boldsymbol{\theta}}_{1:t})$ involves summation from $\tilde{I} = 1$ to N over the realizations from time $t-1$, which can be computationally expensive. However, if only a few of the past parameters are modified, most (if not all except one) of the N terms in the sum are equal to zero, making it manageable to evaluate the mixture sum. (For example, if we only modify the component $\boldsymbol{\theta}^{(\tilde{I})}_{t-1}$ of the selected realization from time $t-1$, then only realizations from time $t-1$ which have identical parameter history from time 1 to $t-2$ yield non-zero probability in computing the summation associated with $q_t(\tilde{\boldsymbol{\theta}}_{1:t})$.)

3.4 MCMC Proposal Distributions

Nothing has been said so far on how the proposal distributions (14) of the MCMC algorithm are specified. In general, the only condition that $q_t(\tilde{\boldsymbol{\theta}}_{1:t} \,|\, \boldsymbol{\theta}_{1:t}^{(i)})$ in (14) needs to satisfy is the rather natural condition that

$$q_t(\tilde{\boldsymbol{\theta}}_{1:t} \,|\, \boldsymbol{\theta}_{1:t}^{(i)}) > 0 \;\text{ if and only if }\; q_t(\boldsymbol{\theta}_{1:t}^{(i)} \,|\, \tilde{\boldsymbol{\theta}}_{1:t}) > 0.$$

We shall now briefly mention few approaches that have been used for constructing proposal distributions.

The Gibbs Sampler.

The Gibbs-sampling approach partitions the parameter vector $\boldsymbol{\theta}_{1:t}$ into blocks of related parameters (e.g., into t blocks with each block given by $\boldsymbol{\theta}_{t'}$; $t' = 1, \ldots, t$). A proposal is then made by changing the parameters of a single block at a time using the *full conditional distribution* (see below) of the block's parameters as the proposal distribution. To demonstrate, let each parameter block consist of $\boldsymbol{\theta}_{t'}$; $t' = 1, \ldots, t$, and we wish to propose a change to the block indexed by $t' \in \{1, \ldots, t\}$. The new proposal, $\tilde{\boldsymbol{\theta}}_{1:t}$, is given by

$$\tilde{\boldsymbol{\theta}}_{1:t\backslash t'} = \boldsymbol{\theta}_{1:t\backslash t'} \;\text{ and }\; \tilde{\boldsymbol{\theta}}_{t'} \sim \pi_t(\tilde{\boldsymbol{\theta}}_{t'} \,|\, \boldsymbol{\theta}_{1:t\backslash t'}),$$

where $\boldsymbol{\theta}_{1:t\backslash t'} \equiv \{\boldsymbol{\theta}_\tau : \tau = 1, \ldots, t,\; \tau \neq t'\}$ and $\pi(\tilde{\boldsymbol{\theta}}_{t'} \,|\, \boldsymbol{\theta}_{1:t\backslash t'})$ is the full conditional distribution of $\tilde{\boldsymbol{\theta}}_{t'}$, given by

$$\pi_t(\tilde{\boldsymbol{\theta}}_{t'} \,|\, \boldsymbol{\theta}_{1:t\backslash t'}) = p(\tilde{\boldsymbol{\theta}}_{t'} \,|\, \mathbf{y}_{1:t}, \boldsymbol{\theta}_{1:t\backslash t'}).$$

The acceptance ratio (15) is then given by

$$\begin{aligned}\rho_t(\tilde{\boldsymbol{\theta}}_{1:t}; \boldsymbol{\theta}_{1:t}) &= \frac{\pi_t(\boldsymbol{\theta}_{1:t\backslash t'}, \tilde{\boldsymbol{\theta}}_{t'})\pi_t(\boldsymbol{\theta}_{t'} \,|\, \boldsymbol{\theta}_{1:t\backslash t'})}{\pi_t(\boldsymbol{\theta}_{1:t\backslash t'}, \boldsymbol{\theta}_{t'})\pi_t(\tilde{\boldsymbol{\theta}}_{t'} \,|\, \boldsymbol{\theta}_{1:t\backslash t'})} \\ &= \frac{\big(\pi_t(\boldsymbol{\theta}_{1:t\backslash t'})\pi_t(\tilde{\boldsymbol{\theta}}_{t'} \,|\, \boldsymbol{\theta}_{1:t\backslash t'})\big)\pi_t(\boldsymbol{\theta}_{t'} \,|\, \boldsymbol{\theta}_{1:t\backslash t'})}{\big(\pi_t(\boldsymbol{\theta}_{1:t\backslash t'})\pi_t(\boldsymbol{\theta}_{t'} \,|\, \boldsymbol{\theta}_{1:t\backslash t'})\big)\pi_t(\tilde{\boldsymbol{\theta}}_{t'} \,|\, \boldsymbol{\theta}_{1:t\backslash t'})} = 1.\end{aligned}$$

Hence, Gibbs-sampler moves are always accepted. The algorithm updates the different parameter blocks in a systematic order or a parameter block is selected randomly and updated.

For complex models, the full conditional proposal distributions needed are not always available in closed form or readily available for sampling. However, one can aim at constructing a proposal distribution q_t that is an approximation to the full conditional distribution (e.g., a Gaussian approximation). In that case, one would need to compute the acceptance ratio as it is not guaranteed to be equal to 1 (i.e., some of the proposal made by the approximation will most likely be rejected).

Random-Walk MCMC

One of the more common way to create a MCMC proposal distribution is via simple random walk. Let $\boldsymbol{\theta}_{1:t}^{(i)}$ be the current state of the Markov chain. A new proposal is generated as

$$\tilde{\boldsymbol{\theta}}_{1:t} = \boldsymbol{\theta}_{1:t}^{(i)} + \boldsymbol{\delta}_{1:t},$$

where $\boldsymbol{\delta}_{1:t} \sim q_t(\boldsymbol{\delta}_{1:t} \,|\, \boldsymbol{\theta}_{1:t}^{(i)})$. Hence, a perturbation is made to the current state of the chain. The new proposal is then accepted or rejected in the usual way.

Langevin Diffusion.

Langevin diffusion can be thought of as a special case of a more general hybrid (or rather 'Hamiltonian') Monte Carlo algorithms (see e.g., Liu, 2001, chapter 9) and yields a more effective random-walk procedure.

Let $\boldsymbol{\theta}_{1:t}^{(i)}$ be the current state of the Markov chain. A new proposal is given by

$$\tilde{\boldsymbol{\theta}}_{1:t} = \boldsymbol{\theta}_{1:t}^{(i)} + \frac{1}{2} \frac{\partial \log \pi_t(\boldsymbol{\theta}_{1:t})}{\partial \boldsymbol{\theta}_{1:t}} \bigg|_{\boldsymbol{\theta}_{1:t}^{(i)}} h + h^{1/2} \mathbf{Z}_t,$$

where $\mathbf{Z}_t \sim \text{Gau}(\mathbf{0}, \mathbf{I})$ and h is a provided step-size parameter. Note the use of the gradient of the log-posterior distribution in determine the proposal — the new proposal has a tendency to be closer to the (local) mode of the posterior distribution. The new proposal is then accepted or rejected in the usual way.

Table 2: Importance Sampling (IS) Algorithm.

(1) Generate a sample of size N from the *proposal distribution* $q(\boldsymbol{\theta})$;

$$\boldsymbol{\theta}^{(i)} \sim q(\boldsymbol{\theta}), \quad i = 1, \ldots, N.$$

(2) Compute the importance weights,

$$\tilde{w}^{(i)} \propto \frac{\pi(\boldsymbol{\theta}^{(i)})}{q(\boldsymbol{\theta}^{(i)})}, \quad i = 1, \ldots, N,$$

and define $w^{(i)} = \tilde{w}^{(i)} / \sum_{j=1}^{N} \tilde{w}^{(j)}$.

The distribution $\pi(\cdot)$ is then approximated by

$$\hat{\pi}^N(\boldsymbol{\theta}) \equiv \sum_{i=1}^{N} w^{(i)} \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^{(i)}),$$

which places the probability mass $w^{(1)}, \ldots, w^{(N)}$ on the support points $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(N)}$.

4 Sequential Monte Carlo (SMC)

Sequential Monte Carlo (SMC) is inherently designed to sample from dynamic posterior distributions, both in terms of leveraging the dynamic nature of the model and also in terms of reusing previous calculations. As SMC is not Markovian, it is inherently parallel; the different Monte Carlo proposals can be generated and evaluated in parallel. A good Introduction to SMC is "Sequential Monte Carlo Methods in Practice" by Doucet et al. (2001) and "Monte Carlo Strategies in Scientific Computing" by Liu (2001). The paper by Arulampalam et al. (2002) gives a tutorial focusing on Bayesian tracking.

4.1 Importance Sampling (IS)

At the core of the SMC approach is the generation of a weighted sample via importance sampling (IS). Suppose one wants to generate a sample of size N from the distribution $\pi(\boldsymbol{\theta})$ without having direct access to an algorithm to do so, but is able to evaluate $\pi(\boldsymbol{\theta})$ up to a proportionality constant. Importance sampling accomplishis this by using a proposal distribution $q(\boldsymbol{\theta})$, that is close to $\pi(\boldsymbol{\theta})$ and from which it is easy to generate samples. The basic algorithm is given in Table 2 on page 20.

The efficiency of the IS algorithm to generate a representative sample from the target distribution, $\pi(\boldsymbol{\theta})$, is judged by how evenly the importance weights $\{\tilde{w}^{(i)}\}$ are distributed. One measure on the efficiency is the *effective sample size*, defined as

$$\text{ESS} \equiv \frac{1}{\sum_{i=1}^{N}(w^{(i)})^2}.$$

If all the weights are equal, then ESS $= N$, and on the other side, if all the weights are equal to zero except one, then ESS $= 1$.

For posterior inference, where $\pi(\boldsymbol{\theta}) \propto p(\mathbf{y} \,|\, \boldsymbol{\theta})p(\boldsymbol{\theta})$ and $\mathbf{y}$ is the observed data, IS is particularly useful. For example, one could take the proposal distribution as the prior distribution, $q(\boldsymbol{\theta}) = p(\boldsymbol{\theta})$, which would result in

$$\tilde{w}^{(i)} = p(\mathbf{y} \,|\, \boldsymbol{\theta}^{(i)}), \quad \text{for } \boldsymbol{\theta}^{(i)} \sim p(\boldsymbol{\theta})\,;\; i = 1, \ldots, N.$$

Hence, the weights would be proportional to the likelihood. Note that this might not yield an effective posterior sample (in terms of ESS) and a better proposal distribution might be needed, that is, a distribution that is closer to $p(\mathbf{y} \,|\, \boldsymbol{\theta})p(\boldsymbol{\theta})$.

4.2 The Basics of SMC

Sequential Monte Carlo aims at using IS to generate samples from a sequence of distributions, $\pi_1(\boldsymbol{\theta}_1), \pi_2(\boldsymbol{\theta}_{1:2}), \ldots$, without needing to start from "scratch" with each new distribution. This makes SMC particularly efficient for dynamically evolving models. The basic steps of the SMC algorithm are given in Table 3 on page 22.

The SMC algorithm is relatively simple, but as in IS, its effectiveness is determined by how good the proposal distribution is in Step A, Table 3, and how computationally feasible it is to evaluate the resulting importance weights in Step B. By taking advantage of the dynamic nature of the model, the proposal distribution can be partitioned in the same sequential fashion as the prior distribution,

$$q_t(\tilde{\boldsymbol{\theta}}_{1:t}) = q_t(\tilde{\boldsymbol{\theta}}_1)q_t(\tilde{\boldsymbol{\theta}}_2 \,|\, \tilde{\boldsymbol{\theta}}_{1:1}) \cdots q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t-1}) = \prod_{t'=1}^{t} q_t(\tilde{\boldsymbol{\theta}}_{t'} \,|\, \tilde{\boldsymbol{\theta}}_{1:t'-1}), \tag{20}$$

where recall that $\tilde{\boldsymbol{\theta}}_{1:0} = \emptyset$, an empty set of parameters. For the proposal distribution (20), the IS weight (19) can be written as

$$\tilde{w}_{1:t} \propto \prod_{t'=1}^{t} \left(\frac{p(\mathbf{y}_{t'} \,|\, \tilde{\boldsymbol{\theta}}_{1:t'})p(\tilde{\boldsymbol{\theta}}_{t'} \,|\, \tilde{\boldsymbol{\theta}}_{1:t'-1})}{q_t(\tilde{\boldsymbol{\theta}}_{t'} \,|\, \tilde{\boldsymbol{\theta}}_{1:t'-1})} \right) \propto \tilde{w}_{1:t-1} \left(\frac{p(\mathbf{y}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t})p(\tilde{\boldsymbol{\theta}}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t-1})}{q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t-1})} \right)$$

using the product format of the posterior in (6).

Note, although not directly indicated, all the conditional distributions in (20) may take advantage of the IS from the previous time point, $\Theta_{1:t-1}$, and the new data, $\mathbf{y}_t$; along the lines of the sequential MCMC algorithms in Section 3.2 and 3.3. This is really the key to the success of SMC for dynamic problems.

Table 3: Sequential Monte Carlo (SMC) Algorithm:

Initialization: Assume at time $t = t_0 \in \{1, 2, \dots\}$ we have an importance sample

$$\Theta_{1:t_0} = \{\boldsymbol{\theta}^{(i)}_{1:t_0}, w^{(i)}_{1:t_0} : i = 1, \dots, N\}$$

from the posterior distribution $\pi_{t_0}(\boldsymbol{\theta}_{1:t_0})$

For $t = t_0 + 1, t_0 + 2, \dots$**:**

Step A (Proposal)

For $i = 1, \dots, N$, sample

$$\tilde{\boldsymbol{\theta}}^{(i)}_{1:t} \sim q_t(\tilde{\boldsymbol{\theta}}_{1:t}) = q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t-1}) q_t(\tilde{\boldsymbol{\theta}}_{1:t-1})$$

where $q_t(\tilde{\boldsymbol{\theta}}_{1:t})$ is a user-specified *proposal distribution*. Note how the proposal distribution is partitioned into two parts; first $\tilde{\boldsymbol{\theta}}_{1:t-1}$ is sampled from $q_t(\tilde{\boldsymbol{\theta}}_{1:t-1})$ and then $\tilde{\boldsymbol{\theta}}_t$ is sampled from $q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t-1})$.

The key to a good SMC proposal distribution is to leverage (condition on) $\Theta_{1:t-1}$ and the new data $\mathbf{y}_t$. That is, take

$$q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t-1}) q_t(\tilde{\boldsymbol{\theta}}_{1:t-1}) = q_t(\tilde{\boldsymbol{\theta}}_t | \tilde{\boldsymbol{\theta}}_{1:t-1}, \mathbf{y}_t) q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} \,|\, \Theta_{1:t-1}, \mathbf{y}_t).$$

Step B (Importance Weights)

For $i = 1, \dots, N$, evaluate the unscaled *importance weights*,

$$\tilde{w}^{(i)}_{1:t} \propto \frac{\pi_t(\tilde{\boldsymbol{\theta}}^{(i)}_{1:t})}{q_t(\tilde{\boldsymbol{\theta}}^{(i)}_{1:t})} \propto \frac{p(\mathbf{y}_t \,|\, \tilde{\boldsymbol{\theta}}^{(i)}_{1:t}) p(\tilde{\boldsymbol{\theta}}^{(i)}_t \,|\, \tilde{\boldsymbol{\theta}}^{(i)}_{1:t-1})}{q_t(\tilde{\boldsymbol{\theta}}^{(i)}_t \,|\, \tilde{\boldsymbol{\theta}}^{(i)}_{1:t-1})} \frac{\pi_{t-1}(\tilde{\boldsymbol{\theta}}^{(i)}_{1:t-1})}{q_t(\tilde{\boldsymbol{\theta}}^{(i)}_{1:t-1})} \tag{19}$$

Let,

$$\boldsymbol{\theta}^{(i)}_{1:t} = \tilde{\boldsymbol{\theta}}^{(i)}_{1:t} \quad \text{and} \quad w^{(i)}_{1:t} = \tilde{w}^{(i)}_{1:t} / \sum_{j=1}^{N} \tilde{w}^{(j)}_{1:t},$$

then, we have the approximation;

$$\pi_t(\boldsymbol{\theta}_{1:t}) \simeq \hat{\pi}^N_t(\boldsymbol{\theta}_{1:t}) \equiv \sum_{i=1}^{N} w^{(i)}_{1:t} \delta(\boldsymbol{\theta}_{1:t} - \boldsymbol{\theta}^{(i)}_{1:t}).$$

Note. Above, $\boldsymbol{\theta}^{(i)}_{1:t}$ is simply put equal to $\tilde{\boldsymbol{\theta}}^{(i)}_{1:t}$, however, often an additional perturbation step is introduced (e.g., a single MCMC step) yielding $\boldsymbol{\theta}^{(i)}_{1:t}$ different from $\tilde{\boldsymbol{\theta}}^{(i)}_{1:t}$.

A natural way to take advantage of the IS from $\pi_{t-1}(\boldsymbol{\theta}_{1:t-1})$ is to build a proposal distribution $q_t(\cdot)$ that conditions on a given realization from $\pi_{t-1}(\boldsymbol{\theta}_{1:t-1})$ (similar to Section 3.2). Such proposal distribution can be written as

$$\begin{aligned} q_t(\tilde{\boldsymbol{\theta}}_{1:t} \,|\, \boldsymbol{\theta}_{1:t-1}) &= q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t-1}) q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} \,|\, \boldsymbol{\theta}_{1:t-1}), \\ &\quad \text{where } \boldsymbol{\theta}_{1:t-1} \sim \pi_{t-1}(\boldsymbol{\theta}_{1:t-1}). \end{aligned} \tag{21}$$

The proposal distribution $q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} \,|\, \boldsymbol{\theta}_{1:t-1})$ can be taken to be of the sequentially form (see also (16)),

$$q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} \,|\, \boldsymbol{\theta}_{1:t-1}) = \prod_{t'=1}^{t-1} q_t(\tilde{\boldsymbol{\theta}}_{t'} \,|\, \tilde{\boldsymbol{\theta}}_{1:t'-1}, \boldsymbol{\theta}_{1:t'}).$$

Hence, the proposal can be considered to consist of three steps: (1) draw a realization from $\pi_{t-1}(\cdot)$, (2) perturbing the drawn realization via $q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} \,|\, \boldsymbol{\theta}_{1:t-1})$, and finally (3) extend the perturbed realization by drawing $\tilde{\boldsymbol{\theta}}_t$ from $q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t-1})$. The immediate drawback of this general conditional proposal approach above is that

$$q_t(\tilde{\boldsymbol{\theta}}_{1:t}) = q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t-1}) \int q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} \,|\, \boldsymbol{\theta}_{1:t-1}) \pi_{t-1}(\boldsymbol{\theta}_{1:t-1}) d\boldsymbol{\theta}_{1:t-1} \tag{22}$$

is needed for the evaluation of the importance weight in (19). This integral is rarely available in closed form and often difficult to evaluate directly. However, one has the approximation,

$$q_t(\tilde{\boldsymbol{\theta}}_{1:t}) \simeq q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t-1}) \sum_{i=1}^{N} q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} \,|\, \boldsymbol{\theta}_{1:t-1}^{(i)}) w_{1:t-1}^{(i)},$$

using the IS approximation $\hat{\pi}_{t-1}^{N}(\boldsymbol{\theta}_{1:t-1})$ of $\pi_{t-1}(\boldsymbol{\theta}_{1:t-1})$. Depending on how the proposal $q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} \,|\, \boldsymbol{\theta}_{1:t-1})$ is constructed, most of the terms in the summation above might be equal to zero, and only few a terms would need to be summed up (it is computationally expensive to loop through all N terms of the sum to generate a single proposal — recall there are N proposals to be made). We shall now outline SMC algorithms that take this approach and have been showed to be successful in number of cases; see Doucet et al. (2001).

4.3 SMC via Rejuvenation and Extension

This algorithm is the SMC version of the MCMC rejuvenation and extension algorithm in Section 3.2 — or vice versa. We shall first introduce it from a more classical view which is often attributed to Neil Gordon (Gordon et al., 1993) and referred to as Gordon's bootstrap filter or simply as a particle filter. We then follow up with a generalization due to Pitt and Shephard (Pitt & Shephard, 1999, 2001) which improves on its efficiency and robustness.

Gordon's Bootstrap Filter

Classical applications of the SMC algorithm often have a data model (a likelihood) where the data at time t only depends on the model parameters at time t,

$$p(\mathbf{y}_t \,|\, \boldsymbol{\theta}_{1:t}) = p(\mathbf{y}_t \,|\, \boldsymbol{\theta}_t).$$

An example is the target tracking model in Section 2.2. As such, the newly observed data $\mathbf{y}_t$ is mostly informative about $\boldsymbol{\theta}_t$ and carries less information about $\boldsymbol{\theta}_{t-1}, \ldots, \boldsymbol{\theta}_1$. In light of this, a good candidate for the conditional proposal distribution in (21) is

$$\begin{aligned} q_t(\tilde{\boldsymbol{\theta}}_{1:t} \,|\, \boldsymbol{\theta}_{1:t-1}) = q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t-1})\delta(\tilde{\boldsymbol{\theta}}_{1:t-1} - \boldsymbol{\theta}_{1:t-1}), \\ \text{where } \boldsymbol{\theta}_{1:t-1} \sim \pi_{t-1}(\boldsymbol{\theta}_{1:t-1}), \end{aligned} \tag{23}$$

which corresponds to taking $q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} \,|\, \boldsymbol{\theta}_{1:t-1}) = \delta(\tilde{\boldsymbol{\theta}}_{1:t-1} - \boldsymbol{\theta}_{1:t-1})$ in (21). That is, $\tilde{\boldsymbol{\theta}}_{1:t} = (\boldsymbol{\theta}_{1:t-1}, \tilde{\boldsymbol{\theta}}_t)$, and only the new addition, $\tilde{\boldsymbol{\theta}}_t$, is generated and the rest is kept identical to $\boldsymbol{\theta}_{1:t-1}$. Note, there is nothing in the above approach that prevents it from being used for the more general data model $p(\mathbf{y}_t \,|\, \boldsymbol{\theta}_{1:t})$. However, if the newly acquired data has information that is not very much in line with past data, this approach could yield a large number of SMC realizations with small weights (i.e., a small effective sample size); this issue was also raised in Section 3.2.

To generate a proposal from (23) one would use the IS from $\pi_{t-1}(\boldsymbol{\theta}_{1:t-1})$, and replace Step A in Table 3 with:

Step A (Rejuvenation and Extension Proposal)

(1) Sample $\tilde{I}$ from $\{1, \ldots, N\}$ with $p(\tilde{I} = j) = w^{(j)}_{1:t-1}$; $j = 1, \ldots, N$.

(2) Sample $\tilde{\boldsymbol{\theta}}_t \sim q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \boldsymbol{\theta}^{(\tilde{I})}_{1:t-1})$.

(3) Let $\tilde{\boldsymbol{\theta}}_{1:t} \equiv (\boldsymbol{\theta}^{(\tilde{I})}_{1:t-1}, \tilde{\boldsymbol{\theta}}_t)$.

Since the proposal distribution (23) does not modify the past, the integral in (22) does not need to be evaluated, and the marginal proposal distribution needed for the IS weights in (19) is simply given by

$$q_t(\tilde{\boldsymbol{\theta}}_{1:t}) = q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t-1})\pi_{t-1}(\tilde{\boldsymbol{\theta}}_{1:t-1}).$$

The resulting IS weights in Step B in Table 3 are then given by

$$\tilde{w}_{1:t} \propto \frac{\pi_t(\tilde{\boldsymbol{\theta}}_{1:t})}{q_t(\tilde{\boldsymbol{\theta}}_{1:t})} \propto \frac{p(\mathbf{y}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t})p(\tilde{\boldsymbol{\theta}}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t-1})\pi_{t-1}(\tilde{\boldsymbol{\theta}}_{1:t-1})}{q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t-1})\pi_{t-1}(\tilde{\boldsymbol{\theta}}_{1:t-1})} = \frac{p(\mathbf{y}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t})p(\tilde{\boldsymbol{\theta}}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t-1})}{q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t-1})}, \tag{24}$$

and note how the $\pi_{t-1}(\cdot)$ terms cancel out. Gordon et al. (1993) proposed taking $q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \boldsymbol{\theta}_{1:t-1})$ equal to $p(\tilde{\boldsymbol{\theta}}_t \,|\, \boldsymbol{\theta}_{1:t-1})$, yielding $\tilde{w}_{1:t} = p(\mathbf{y}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t})$.

The goal in importance sampling is always to construct a proposal distribution that results in weights of similar size, yielding a large effective sample size. For the case above, when we condition on the past, it translates into selecting a good proposal distribution for $\tilde{\boldsymbol{\theta}}_t$. It can be shown that the full conditional distribution $p(\tilde{\boldsymbol{\theta}}_t \,|\, \boldsymbol{\theta}_{1:t-1}, \mathbf{y}_t)$ is the "optimal" proposal distribution for Gordon's bootstrap filter, since

$$p(\tilde{\boldsymbol{\theta}}_t \,|\, \boldsymbol{\theta}_{1:t-1}\mathbf{y}_t) \propto p(\mathbf{y}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t}) p(\tilde{\boldsymbol{\theta}}_t \,|\, \boldsymbol{\theta}_{1:t-1}), \tag{25}$$

yielding $\tilde{w}_{1:t} \propto 1$ in (24).

Although one might be able to generate $\tilde{\boldsymbol{\theta}}_t$ using the optimal proposal distribution (25), the step before, proposing $\tilde{\boldsymbol{\theta}}_{1:t-1}$, is done without taking into account the new data — it is simple generated from the posterior at time $t-1$ using the IS. This can be particularly inefficient if the new data carries information that has large impact on the past. Pitt & Shephard (1999, 2001) improved upon the basic bootstrap filter, by taking a similar approach as discussed in Section 3.2 and 3.3 on sequential MCMC, which we shall outline now.

Pitt's and Shephard's Modification

We could have introduced the bootstrap filter by aiming at generating realizations from the following mixture approximation to $\pi_t(\boldsymbol{\theta}_{1:t})$,

$$\begin{aligned} \hat{\pi}_t(\boldsymbol{\theta}_{1:t}) &\equiv C \,\times\, p(\mathbf{y}_t \,|\, \boldsymbol{\theta}_{1:t}) p(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{1:t-1}) \sum_{i=1}^{N} w_{1:t-1}^{(i)} \delta(\boldsymbol{\theta}_{1:t-1} - \boldsymbol{\theta}_{1:t-1}^{(i)}) \\ &= C \,\times\, \sum_{i=1}^{N} p(\mathbf{y}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(i)}, \boldsymbol{\theta}_t) p(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(i)}) w_{1:t-1}^{(i)} \delta(\boldsymbol{\theta}_{1:t-1} - \boldsymbol{\theta}_{1:t-1}^{(i)}), \end{aligned} \tag{26}$$

which is derived from (7) by replacing $\pi_{t-1}(\boldsymbol{\theta}_{1:t-1})$ with its IS approximation $\hat{\pi}_{t-1}^{N}(\boldsymbol{\theta}_{1:t-1})$, and where C is an unknown normalizing constant. Then, similar to Section 3.2, we introduce the augmented parameter $(\boldsymbol{\theta}_{1:t}, I)$ with the joint distribution

$$\pi_t^a(\boldsymbol{\theta}_{1:t}, I) = C \,\times\, p(\mathbf{y}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(I)}, \boldsymbol{\theta}_t) p(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(I)}) w_{1:t-1}^{(I)} \delta(\boldsymbol{\theta}_{1:t-1} - \boldsymbol{\theta}_{1:t-1}^{(I)}), \tag{27}$$

and we note that for the marginal distribution of $\boldsymbol{\theta}_{1:t}$ we have that

$$\pi_t^a(\boldsymbol{\theta}_{1:t}) = \sum_{I=1}^{N} \pi_t^a(\boldsymbol{\theta}_{1:t}, I) = \text{the mixture in (26).}$$

Hence, as mentioned in Section 3.2, this suggests that we could sample from the joint augmented distribution in (27) and then simply drop the index I to derive a sample from (26).

Pitt and Shephard suggested basing the proposal on the augmented distribution

$$q_t^a(\tilde{\boldsymbol{\theta}}_{1:t}, \tilde{I}) = q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}) v_{1:t-1}^{(\tilde{I})} \delta(\tilde{\boldsymbol{\theta}}_{1:t-1} - \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}),$$

where the weighs $v_{1:t-1}^{(1)}, \dots, v_{1:t-1}^{(N)}$ are allowed to depend on the new data $\mathbf{y}_t$; if $v_{1:t-1}^{(i)} = w_{1:t-1}^{(i)}$ their approach yields the bootstrap filter. The marginal proposal distribution for $\tilde{\boldsymbol{\theta}}_{1:t}$ is then

$$q_t(\tilde{\boldsymbol{\theta}}_{1:t}) = \sum_{i=1}^{N} q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t-1}^{(i)}) v_{1:t-1}^{(i)} \delta(\tilde{\boldsymbol{\theta}}_{1:t-1} - \boldsymbol{\theta}_{1:t-1}^{(i)}), \tag{28}$$

which can be compared to (26). An augmented proposal is then simply generated using the following procedure (very similar to the previous one):

Step A (P & S Rejuvenation and Extension Proposal)

(1) Sample $\tilde{I}$ from $\{1, \dots, N\}$ with $p(\tilde{I} = j) = v_{1:t-1}^{(j)}$; $j = 1, \dots, N$.

(2) Sample $\tilde{\boldsymbol{\theta}}_t \sim q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})})$.

(3) Let $\tilde{\boldsymbol{\theta}}_{1:t} = (\boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}, \tilde{\boldsymbol{\theta}}_t)$, and the augmented proposal is $(\tilde{\boldsymbol{\theta}}_{1:t}, \tilde{I})$.

The IS weight associated with the proposal $(\tilde{\boldsymbol{\theta}}_{1:t}, \tilde{I})$ is given by

$$\tilde{w}_{1:t} \propto \frac{\pi_t^a(\tilde{\boldsymbol{\theta}}_{1:t}, \tilde{I})}{q_t^a(\tilde{\boldsymbol{\theta}}_{1:t}, \tilde{I})} \propto \frac{p(\mathbf{y}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}, \tilde{\boldsymbol{\theta}}_t) p(\tilde{\boldsymbol{\theta}}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}) \tilde{w}_{1:t-1}^{(\tilde{I})}}{q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}) v_{1:t-1}^{(\tilde{I})}} \tag{29}$$

In light of this, Pitt and Shephard proposed taking

$$v_{1:t-1}^{(i)} \propto w_{1:t-1}^{(i)} p(\mathbf{y}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(i)}, \hat{\boldsymbol{\theta}}_t) \quad \text{and} \quad q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(i)}) = p(\tilde{\boldsymbol{\theta}}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(i)}),$$

where $\hat{\boldsymbol{\theta}}_t = \hat{\boldsymbol{\theta}}_t(\boldsymbol{\theta}_{1:t-1}^{(i)})$ is some likely value of $\tilde{\boldsymbol{\theta}}_t$ conditional on $\boldsymbol{\theta}_{1:t-1}^{(i)}$ (i.e., the mode, the mean, or other likely value associated with $p(\tilde{\boldsymbol{\theta}}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(i)})$). Alternatively, one could use

$$v_{1:t-1}^{(i)} \propto w_{1:t-1}^{(i)} p(\mathbf{y}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(i)}, \hat{\boldsymbol{\theta}}_t) p(\hat{\boldsymbol{\theta}}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(i)}) \quad \text{and} \quad q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(i)}) = p(\tilde{\boldsymbol{\theta}}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(i)}, \mathbf{y}_t),$$

and recall that $p(\tilde{\boldsymbol{\theta}}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(i)}, \mathbf{y}_t) \propto p(\mathbf{y}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(i)}, \tilde{\boldsymbol{\theta}}_t) p(\tilde{\boldsymbol{\theta}}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(i)})$.

Note that the final weights are not all equal, but the newly acquired data impacted the weights $v_{1:t-1}^{(1)}, \dots, v_{1:t-1}^{(N)}$, and therefore which realizations from time $t-1$ were carried on to time t. This is particularly important when the distribution of $\mathbf{y}_t$ is given by $p(\mathbf{y}_t \,|\, \boldsymbol{\theta}_{1:t})$, but not by $p(\mathbf{y}_t \,|\, \boldsymbol{\theta}_t)$.

4.4 SMC via Rejuvenation, Modification and Extension

We shall now extend the SMC approach introduced in the previous section to the case where we not only extend previous IS realizations, but also modify them to some extent. (See Section 3.3 for a similar topic.)

One approach to extend previous SMC approaches, that only rejuvenate and extend previous IS realizations, is to consider a proposal distribution of the following form,

$$q_t(\tilde{\boldsymbol{\theta}}_{1:t}) = q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t-1}) \sum_{i=1}^{N} q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} \,|\, \boldsymbol{\theta}_{1:t-1}^{(i)}) q_t(i) \tag{30}$$

where $\{\boldsymbol{\theta}_{1:t-1}^{(i)}\}$ are the IS from time $t-1$. Note that through $q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} \,|\, \boldsymbol{\theta}_{1:t-1}^{(i)})$ a perturbation can be made to i-th IS realization from time $t-1$. A proposal from this distribution is generated by the following steps:

Step A (Rejuvenation, Modification, and Extension Proposal)

(1) Sample $\tilde{I}$ from $\{1, \dots, N\}$ with $p(\tilde{I} = i) = q_t(i)$.
(2a) Sample $\tilde{\boldsymbol{\theta}}_{1:t-1} \sim q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} \,|\, \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})})$.
(2b) Sample $\tilde{\boldsymbol{\theta}}_t \sim q_t(\tilde{\boldsymbol{\theta}}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t-1})$.
(3) Put $\tilde{\boldsymbol{\theta}}_{1:t} = (\tilde{\boldsymbol{\theta}}_{1:t-1}, \tilde{\boldsymbol{\theta}}_t)$.

As mentioned in Section 3.3, the conditional proposal distribution $q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} \,|\, \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})})$ can be taken to be of the sequential form,

$$q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} \,|\, \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}) = \prod_{t'=1}^{t-1} q_t(\tilde{\boldsymbol{\theta}}_{t'} \,|\, \tilde{\boldsymbol{\theta}}_{1:t'-1}, \boldsymbol{\theta}_{1:t'}^{(\tilde{I})}),$$

and note that each of sub-proposal distributions may depend on the newly acquired data, $\mathbf{y}_t$. If the new data is only informative for a small subset of the parameters, many of the sub-proposal distributions can simply keep the past value of the parameter intact (i.e., put $\tilde{\boldsymbol{\theta}}_{t'} = \tilde{\boldsymbol{\theta}}_{t'}^{(\tilde{I})}$ with probability 1 for some $t' \in \{1, \dots, t-1\}$).

The main difference between this approach and the previous approach, in which we did not perturb the past IS realizations, is that IS weight calculations can not be based on the IS approximation in (26). The weight calculations need to be based on the original expression for the posterior distribution; see (6). That makes this approach not as computationally efficient as the previous method, however, it is more flexible. The amount of extra computational effort needed depends on how extensively the proposal distribution $q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} \,|\, \boldsymbol{\theta}^{(\tilde{I})}_{1:t-1})$ modifies the IS realization $\boldsymbol{\theta}^{(\tilde{I})}_{1:t-1}$, generated at the previous time point. Recall, from (6), that the posterior distribution can be written as

$$\pi_t(\boldsymbol{\theta}_{1:t}) \propto \prod_{t'=1}^{t} p(\mathbf{y}_{t'} \,|\, \boldsymbol{\theta}_{1:t'}) p(\boldsymbol{\theta}_{t'} \,|\, \boldsymbol{\theta}_{1:t'-1}). \tag{31}$$

Then depending on how extensively $\boldsymbol{\theta}^{(\tilde{I})}_{1:t-1}$ is modified by the proposal process, most of the computations involved in computing the above posterior have already been carried out at the previous time point. A similar argument applies to the evaluation of the proposal distribution $q_t(\tilde{\boldsymbol{\theta}}_{1:t})$, given by (30) and needed in (19) to compute the IS weight; see also Section 3.3 on generalizing the rejuvenating and extending MCMC algorithm. We shall come back to the issue of modifying past IS realizations later in Section 4.5, where we combine SMC with MCMC to perturb past realizations.

Note. For most applications, it is reasonable to assume that the newly acquired data at time t, $\mathbf{y}_t$, has information content mostly relevant to parameters close to t in time. That is, $\mathbf{y}_t$ has no (or very small) information value for $\boldsymbol{\theta}$-parameters sufficiently far into the past; for $\boldsymbol{\theta}_{t'}$ where t' is considerably smaller than t. As such, in practice one does not carry around the whole time history of the $\boldsymbol{\theta}$ parameter, but rather a time window of a fixed size (i.e., $\boldsymbol{\theta}_{1:t}$ is replaced with $\boldsymbol{\theta}_{(t-k):t}$ for some k).

4.5 Hybrid Methods: MCMC within SMC and MCMC prior to SMC

There can be some benefits of mixing SMC and MCMC to generate realizations from the posterior. There are really two areas where MCMC could benefit SMC.

MCMC Within SMC

Some recent attempts have been made using a one or more MCMC steps within each SMC step to perturb the current IS (MacEachern et al., 1999; Gilks & Berzuini, 2001; Godsill & Clapp, 2001). For example, in the case where one adopts SMC based on rejuvenation and extension (i.e., a SMC that does not modify the past), one can at the end of each SMC time step apply one or more MCMC steps to each IS realization. The MCMC step can be very general, and in particular one could propose to modify the past, resulting in a SMC-MCMC hybrid algorithm that rejuvenates, extends, and modifies past IS realizations. The most basic algorithm is as follows:

```
Step 0: Assume at time $t-1$ we have the IS $\Theta_{t-1} = \{\boldsymbol{\theta}^{(i)}_{1:t-1}, w^{(i)}_{1:t-1} : i = 1, \dots, N\}$.
Step 1: Carry out a SMC step from time $t-1$ to $t$ (e.g., using Pitt's and Shephard's rejuvenation and extension algorithm from Section 4.3), yielding a new IS $\Theta_t = \{\boldsymbol{\theta}^{(i)}_{1:t}, w^{(i)}_{1:t} : i = 1, \dots, N\}$.
Step 2: Use $B$ MCMC steps to perturb the IS:
  For $i = 1, \dots, N$:
    Step 2.1: Draw $I_i \in \{1, \dots, N\}$ with $p(I_i = k) = w^{(k)}_{1:t}$; $k = 1, \dots, N$, and put $\boldsymbol{\theta}^{(i,0)}_{1:t} = \boldsymbol{\theta}^{(I_i)}_{1:t}$.
    For $j = 1, \dots, B$:
      Step 3.1: Make a MCMC proposal $\tilde{\boldsymbol{\theta}}_{1:t} \sim q(\tilde{\boldsymbol{\theta}}_{1:t} \mid \boldsymbol{\theta}^{(i,j-1)}_{1:t})$.
      Step 3.2: Compute the MCMC acceptance probability $\alpha(\tilde{\boldsymbol{\theta}}_{1:t}; \boldsymbol{\theta}^{(i,j-1)}_{1:t})$ and put $\boldsymbol{\theta}^{(i,j)}_{1:t} = \tilde{\boldsymbol{\theta}}_{1:t}$ with probability $\alpha(\tilde{\boldsymbol{\theta}}_{1:t}; \boldsymbol{\theta}^{(i,j-1)}_{1:t})$, else $\boldsymbol{\theta}^{(i,j)}_{1:t} = \boldsymbol{\theta}^{(i,j-1)}_{1:t}$.
Step 3 The new IS is given by $\boldsymbol{\theta}^{(i)}_{1:t} = \boldsymbol{\theta}^{(i,M)}_{1:t}$ with $w^{(i)}_{1:t} = 1/N$ — that is, the sample is equally weighted.
```

There is a variation to the algorithm above where the random draw in **Step 2.1** is simply replaced with $\boldsymbol{\theta}^{(i,0)}_{1:t} = \boldsymbol{\theta}^{(i)}_{1:t}$.

MCMC Prior to SMC

The SMC algorithm in Table 3 needs to be initialized with an IS at time t_0; the first time point of data processing. An ideal way to generate this initial sample is via MCMC using data from time $1, \dots, t_0$. The resulting, equally weighted MCMC sample can then be passed on to SMC for processing data from time $t_0+1, t_0+2, \dots$.

4.6 SMC Proposal Distributions

For a SMC algorithm that just rejuvenates and extends past realizations (the bootstrap filter and Pitt's and Shephard's modification), we have already mentioned two natural candidates for the proposal distribution $q_t(\tilde{\boldsymbol{\theta}}_t \mid \tilde{\boldsymbol{\theta}}_{1:t-1})$:

$$q_t(\tilde{\boldsymbol{\theta}}_t \mid \tilde{\boldsymbol{\theta}}_{1:t-1}) = p(\tilde{\boldsymbol{\theta}}_t \mid \tilde{\boldsymbol{\theta}}_{1:t-1}), \qquad \text{(the prior)}$$

$$q_t(\tilde{\boldsymbol{\theta}}_t \mid \tilde{\boldsymbol{\theta}}_{1:t-1}) = p(\tilde{\boldsymbol{\theta}}_t \mid \tilde{\boldsymbol{\theta}}_{1:t-1}, \mathbf{y}_t). \qquad \text{(the full conditional)}$$

In the case where the full conditional distribution is not available, one can aim at designing a proposal distribution that is an approximation to the full conditional (this mirrors the Gibbs proposal algorithm in MCMC). Popular approximations are multi-variate Gaussian or t distributions. If the prior distribution $p(\boldsymbol{\theta}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t-1})$ is informative (relatively narrow and well focused) it is often just sufficient to take the proposal distribution equal to the prior distribution, as suggested by Gordon.

In the case when the past SMC realizations are perturbed by carrying out one or more MCMC steps for each realization, as outlined in previous section, all the proposal methods suggested in Section 3.4 apply.

5 Applications

We shall demonstrate the use of MCMC and SMC for two applications. The first one is a linear Gaussian (Normal) model, a combination of the Gaussian example given in Section 1.3, page 3, and the Gaussian target-tracking setup given in Section 2.2. In this case there is an analytic, closed form expression for the posterior distributions of interest, which can be compared to the sample-derived (MCMC/SMC) posterior inference approach. The second application is the atmospheric event reconstruction problem described in Section 2.3. In this case there is no closed-form analytic expression available for posterior inference.

5.1 Bivariate Gaussian Distribution

Our setup is as follows: Assume at "time" 1 we have the unknown system parameter x_1 (e.g., a location of an object) and an observation y_1 that is assumed to be related to x_1 according to the additive measurement-error model

$$y_1 = x_1 + \varepsilon_1, \qquad \text{where } \varepsilon_1 \sim \text{Gau}(0, \sigma^2). \tag{32}$$

The measurement-error model can also be written as $y_1 \sim \text{Gau}(x_1, \sigma^2)$. A *priori*, we assume that

$$x_1 = \mu_1 + \delta_1, \quad \text{where } \delta_1 \sim \text{Gau}(0, \tau^2). \tag{33}$$

That is, $x_1 \sim \text{Gau}(\mu_1, \tau^2)$ where both μ_1 and τ^2 are known. Given this setup, Gaussian theory (e.g., West & Harrison, 1997, chapter 17.2) yields:

$$\begin{aligned} \big(y_1 \,|\, x_1\big) &\sim \text{Gau}(x_1, \tau^2 + \sigma^2) \\ \begin{bmatrix} y_1 \\ x_1 \end{bmatrix} &\sim \text{Gau}\left(\begin{bmatrix} \mu_1 \\ \mu_1 \end{bmatrix}, \begin{bmatrix} \tau^2 + \sigma^2 & \tau^2 \\ \tau^2 & \tau^2 \end{bmatrix} \right). \\ \big(x_1 \,|\, y_1\big) &\sim \text{Gau}(\mu_1 + \rho^2(y_1 - \mu_1), \tau^2(1 - \rho^2)) \end{aligned}$$

where $\rho^2 = \tau^2/(\tau^2 + \sigma^2)$. Hence, the posterior distribution of x_1 given y_1 is

$$p(x_1 \,|\, y_1) \text{ , and is } \text{Gau}(\mu_1 + \rho^2(y_1 - \mu_1), \tau^2(1 - \rho^2)). \tag{34}$$

For the setup above, we generated synthetic data. We assumed that $x_1 = 0$ and generated y_1 according to the measurement-error model in (32) with $\sigma^2 = 1$, yielding

$$y_1 = -0.626, \text{ drawn from } \text{Gau}(x_1 = 0, \sigma^2 = 1).$$

The parameters associated with the prior for x_1 in (33) were taken to be

$$\mu_1 = 0, \text{ and } \tau^2 = 10^2,$$

yielding a rather vague prior information. This yields a Gaussian posterior distribution for x_1 with mean equal to -0.620 and standard deviation equal to 0.990;

see (34). We shall now apply MCMC to sample from the posterior distribution and compare to the true distribution.

We applied MCMC using a Gaussian random-walk proposal distribution,

$$\tilde{x}_1 \sim q_1(\tilde{x}_1 \,|\, x_1^{(i)}) = \varphi(\tilde{x}_1; x_1^{(i)}, \xi^2),$$

where $\varphi(\tilde{x}_1; x_1^{(i)}, \xi^2)$ denotes the Gaussian density with mean $x_1^{(i)}$ and variance ξ^2 evaluated at $\tilde{x}_1$. Since the proposal distribution is symmetric ($q_1(\tilde{x}_1 \,|\, x_1^{(i)}) = q_1(x_1^{(i)} \,|\, \tilde{x}_1)$), the acceptance ratio is simply given by

$$\rho(\tilde{x}_1; x_1^{(i)}) = \frac{p(y_1 \,|\, \tilde{x}_1)}{p(y_1 \,|\, x_1^{(i)})} = \frac{\varphi(y_1; \tilde{x}_1, \sigma^2)}{\varphi(y_1; x_1^{(i)}, \sigma^2)}.$$

We generated three MCMC samples, each of size 2,000, using a different value for ξ in the proposal distribution for each sample; $\xi = 0.35, 2.5, 12$. With $\xi = 0.35$ the acceptance rate was around 90% (too high due to too small step-size), with $\xi = 2.5$ the acceptance rate was around 0.43% (which is close to optimal), while for $\xi = 12$ the acceptance rate was around 10% (too low due to too large step-size). Figure 1 summarizes the results from the three chains. We see how the MCMC sample corresponding to $\xi = 2.5$ mixes better than the other two samples, resulting in smaller auto-correlation (i.e., larger effective sample size). A histogram of the sample realizations is seen to match well the true posterior density.

We shall now extend the above example to “time” 2: At time 2 we have the unknown system variable x_2 (e.g., the object moved to a new location) and a new observation y_2 that is assumed to be related to x_2 according to the same additive measurement-error model as before;

$$y_2 = x_2 + \varepsilon_2, \quad \text{where } \varepsilon_2 \sim \text{Gau}(0, \sigma^2).$$

What we know a *priori* is that x_2 is not too far (different) from x_1. We therefore assume the following conditional prior distribution for x_2,

$$x_2 = x_1 + \eta_2, \quad \text{where } \eta_2 \sim \text{Gau}(0, 1).$$

That is, $x_2 \sim \text{Gau}(x_1, 1)$ a *priori*. To generate synthetic data at time 2, we let $x_2 = 1$ and generate the observation y_2 according to the measurement-error model, yielding $y_2 = 1.184$.

Given the new data y_2 we want to derive a sample from the posterior distribution of (x_1, x_2) given (y_1, y_2); that is, from $p(x_1, x_2 \,|\, y_1, y_2)$. It should be noted, since all the distributions involved are Gaussian, the posterior distribution is available in closed form as multivariate Gaussian (see e.g., West & Harrison, 1997, chapter 17.2). Hence, we can compare our sample to the true posterior, as before.

There are two approaches we can take to generate realizations from $p(x_1, x_2 \,|\, y_1, y_2)$: (1) start a new MCMC to generate a sample from $p(x_1, x_2 \,|\, y_1, y_2)$ or (2) use the previous MCMC sample as a starting point for a SMC. The first option would be similar

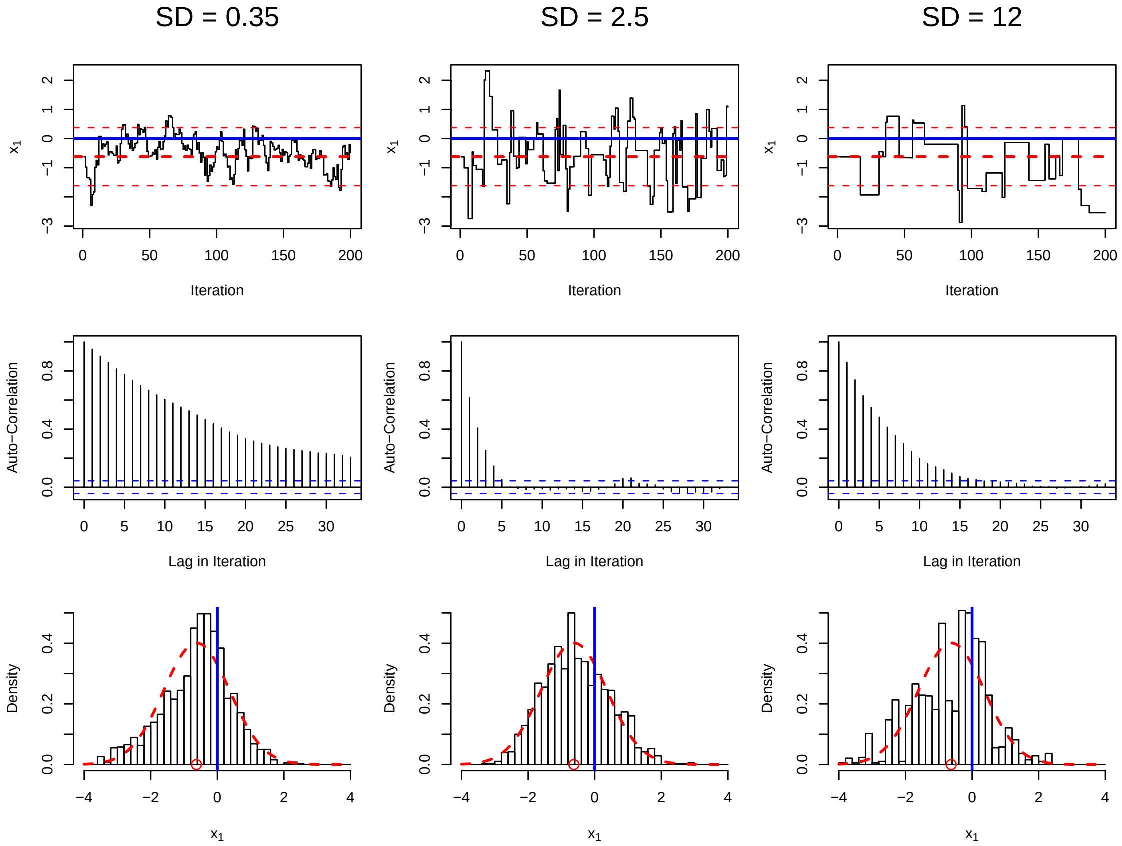


Figure 1: MCMC summary plots for x_1 for three different MCMC samples; left, a proposal distribution with small step-size ($\xi = 0.35$), middle, a proposal distribution with a good step-size ($\xi = 2.5$), and right, a proposal distribution with too big step-size ($\xi = 12$). The first row of plots shows the the first 200 realizations for each chain along with the true value of x_1 superimposed (blue, solid line) and the mean and plus/minus one standard deviation of the true posterior distribution (red,dashed). The middle row of plots shows the auto-correlation in each chain. The bottom row of plots show a histogram of the realizations along with the true value of x_1 (blue, solid) and the true posterior density (red, dashed). The red circles show the data point y_1.

to the previous MCMC approach for x_1 alone, except we would use, for example a 2D Gaussian random-walk proposal as we have to sample both x_1 and x_2. We shall therefore demonstrate the use of the second option, SMC, using Gordon's bootstrap filter, as outlined in Table 4 for this implementation.

Table 4: SMC Algorithm for Bivariate Gaussian Example.

Initial Sample: Start with the inital sample $\{x_1^{(i)} : i = 1, \ldots, N\}$ generated by MCMC. (Note, it is an equally weighted sample; $w_1^{(i)} = 1/N$.)

For $i = 1, \ldots, N$:

(1) Sample $x_2^{(i)} \sim q_2(x_2 \,|\, x_1^{(i)})$, where $q_2(\cdot \,|\, x_1^{(i)})$ is Gau$(x_1^{(i)}, 1)$, the conditional prior distribution.

(2) Compute the importance weights $\tilde{w}_{1:2}^{(i)} = \varphi(y_2; x_2^{(i)}, \sigma^2)$

The final sample is then given by $\{(x_1^{(i)}, x_2^{(i)}), w_{1:2}^{(i)} : i = 1, \ldots, N\}$, where $w_{1:2}^{(i)} = \tilde{w}_{1:2} / \sum_j \tilde{w}_{1:2}^{(j)}$.

To get a final, equally weighted sample at time 2, the final weighted SMC realizations were resampled; that is, 2,000 sample points were drawn from the final collection (with replacement), where the probability of drawing each point is proportional to its final weight. Hence, the resampled collection has multiple copies of realizations with high weights, but realization with low weights have small chance of being picked. (Of the 2,000 original realizations, 1,124 were selected by the resampling process and of those, 500 appeared once in the sample, 372 appeared twice, and 372 three times.) Figure 2 summarizes the results. It shows the marginal histograms of the samples for x_1 and x_2, along with the true posterior distribution, and the joint distribution of x_1 and x_2 along with true posterior contour lines.

5.2 Atmospheric Dispersion Modeling with Unknown Source Characteristics

We shall now apply MCMC and SMC to estimate an unknown release into the atmosphere using a computer dispersion simulation model as described in Section 2.3.

The Setup: Synthetic Truth and Data

To test the feasibility of using MCMC and SMC to conduct inference on the characteristics of an unknown release into the atmosphere, we generated a synthetic sensor

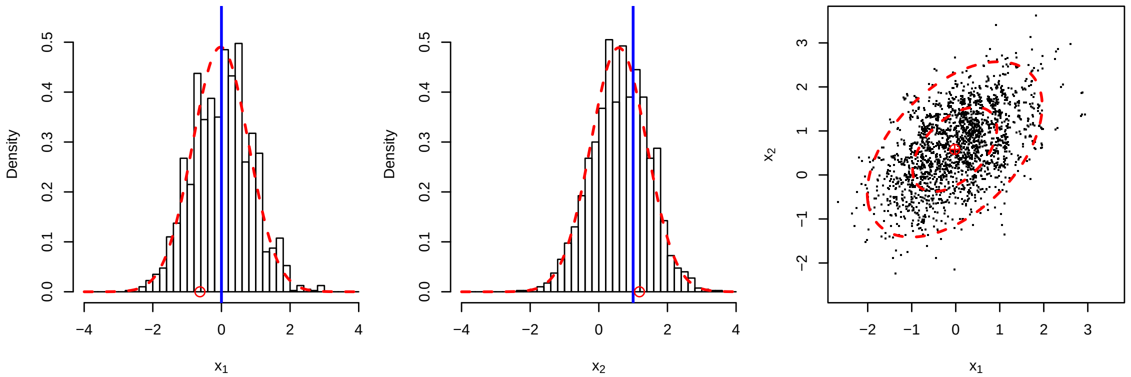


Figure 2: The left and the middle panels show the marginal distribution of x_1 and x_2, with the true marginal distribution shown as (red) dotted line and the true value of x_1 and x_2 given by a (blue) solid vertical line (a red circles show the observed data). The right panel shows the joint distribution of x_1 and x_2 as represented by the SMC realizations (via resampling). The true mean of the joint posterior distribution is shown along with the 50% and the 95% contour lines.

data from a given source. Our setup is shown in Figure 3 (left). It shows a single stationary source on the left side of the domain, with a constant wind blowing from the West and five sensors located downwind from the source. Our time domain is one hour and is splitted into six 10min intervals. In the first 10min interval the source is not emitting at all, it then emits at a (relative) rate of 1.0, 0.5, 0.25, 0.1, 0.0 in the remaining five 10min intervals. The five sensors report 10min average concentrations in the same six 10min intervals as the source is emitting at a constant rate (this is just for convenience and is not required). The atmospheric dispersion model INPUFF (Petersen & Lavdas, 1986) was used to simulate the dispersion of the release, which includes computing average concentrations at the five sensors sites in the six 10min time intervals. These values were taken as the true concentrations at the five sites in the six time periods; that is, in terms of the notation introduced in Section 2.3,

$C(\mathbf{m}_j, t) = \hat{C}(\mathbf{m}_j, t) =$ the INPUFF predicted contaminant average concentration in the t-th time period, $t = 1, \ldots, 6$, at sensor location $\mathbf{m}_j$, $j = 1, \ldots, 5$.

Sensor data $\{c_{j,t} : j = 1, \ldots, 5,\ t = 1, \ldots, 6\}$ was then generated according to the truncated Gaussian data-model in (9) with mean $\hat{C}(\mathbf{m}_j, t)$ and variance $V(\hat{C}(\mathbf{m}_j, t))$ given by

$$V(\hat{C}(\mathbf{m}_j, t)) = \left(1\text{E-}9 + 0.2 \times \hat{C}(\mathbf{m}_j, t)\right)^2. \tag{35}$$

Hence, the standard deviation is given by $1\text{E-}9 + 0.2 \times \hat{C}(\mathbf{m}_j, t)$, indicating that measured average 10min concentration of around 1E-9 and below are not distinguishable from zero, while higher concentration measurements have an approximated coeffi-

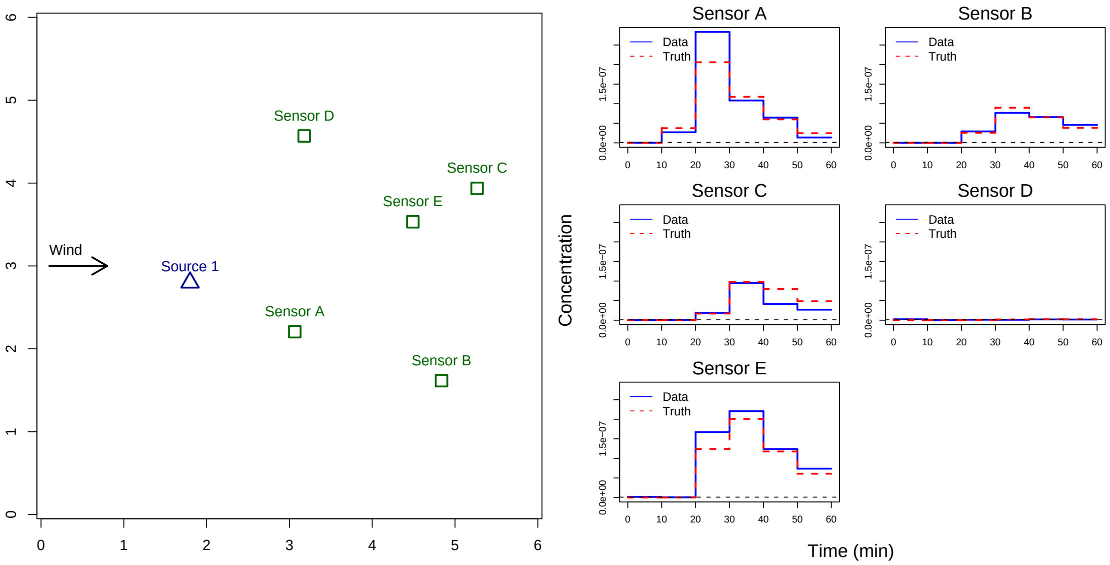


Figure 3: Left, the location of the stationary release source along with the five sensors. Right, the synthetic true 10min average concentration at the five sensor sites along with the synthetic observed concentrations.

cient of variation (CV) equal to 0.2 (20%). Figure 3 (right) shows the synthetic truth $\{\hat{C}(\mathbf{m}_j, t)\}$ at the five sensors along with the synthetic data $\{c_{j,t}\}$.

Finally, we note that the INPUFF model satisfies the additive factorization of the predictive concentration as given by (8). This leads to simplifications (and time-savings) in computations.

Initial MCMC at $t = 2$

From Figure 3 (right) we see that the first non-zero concentration is observed in the second 10min time interval at sensor A, a concentration of 2.7E-8, with the remaining four sensors reporting zero concentrations (or rather, concentrations below detection level).

We now seek to start an initial MCMC sampler to sample from the posterior distribution of the unknown source location, $\mathbf{x}$, and the release rate in the first two 10min time interval, $\mathbf{s}_{1:2}$; that is, we seek to sample from $\pi_2(\boldsymbol{\theta}_{1:2})$, $\boldsymbol{\theta}_{1:2} = (\mathbf{x}, \mathbf{s}_{1:2})$. We assume a flat prior on the location of the source, as outlined in Section 2.3, and a prior on the release rate that assumes an unknown start (i.e., either in the first or the second time period) and then truncated Gaussian distribution for a non-zero release; see (10)–(12). In terms of the notation in Section 2.3, we take the initial non-zero release prior to be given by

$$f_1(s_{t^*}) = f(s_{t^*}) \text{ is } \mathrm{Gau}(0, 20^2)\big|_0^\infty,$$

which is also the prior we use for subsequent releases; that is, $f_2(s_2 \mid s_1) = f(s_1)$.

Hence, we assume that knowing s_1 has no value in determining s_2 a *priori* (a rather vague assumption). To summarize, the prior on $\boldsymbol{\theta}_{1:2}$ is given by

$$p(\boldsymbol{\theta}_{1:2}) = \begin{cases} f(s_1)f(s_2) & \text{if } t^* = 1, \\ f(s_2) & \text{if } t^* = 2. \end{cases}$$

We take the data-model, the likelihood, to be given by the product of the individual distributions in (9), yielding

$$p(\mathbf{c}_{1:2}\,|\,\boldsymbol{\theta}_{1:2}) = \prod_{t=1}^{2}\prod_{j=1}^{5} \varphi(c_{j,t}; \hat{C}(\mathbf{m}_j,t), (2\text{E-}9 + 0.2 \times \hat{C}(\mathbf{m}_j,t))^2)\Big|_0^\infty, \tag{36}$$

where $\varphi(c;\mu,V)\big|_0^\infty$ is the density of a Gaussian distribution with mean μ and variance V, but restricted to the interval $(0,\infty]$. Note we have inflated the variance slightly by adding 1E-9 to the standard deviation used to generate the synthetic data; see (35). This mirrors reality, where the likelihood used in the MCMC sampler is just an approximation to the true (unknown) likelihood function.

The proposal distribution is a mixture of random-walk proposals and consist of either: (1) making a release rate change proposal, or (2) making a source location change proposal, or (3) making a joint release and location change proposal.

For the source location we use a random-walk on a lattice with a 0.1 horizontal/vertical distance between grid-locations:

Location Proposal

Let $\mathbf{x}$ be the current location of the Markov chain, then:

(1) Create the grid-point neighborhood set

$$\mathcal{N}_d(\mathbf{x}) \equiv \{\tilde{\mathbf{x}} : |x_1 - \tilde{x}_1| \le d,\ |x_2 - \tilde{x}_2| \le d,\ \text{and } \tilde{\mathbf{x}} \ne \mathbf{x}\},$$

where $d > 0$ is a given neighborhood-size parameter, and recall that $\mathbf{x} = (x_1, x_2)$ and $\tilde{\mathbf{x}} = (\tilde{x}_1, \tilde{x}_2)$.

(2) Generate the source location proposal $\tilde{\mathbf{x}} \sim q_2(\tilde{\mathbf{x}}\,|\,\mathcal{N}_d(\mathbf{x}))$, where

$$q_2(\tilde{\mathbf{x}}\,|\,\mathcal{N}_d(\mathbf{x})) = \frac{1}{|\mathcal{N}_d(\mathbf{x})|} I(\tilde{\mathbf{x}} \in \mathcal{N}_d(\mathbf{x})), \tag{37}$$

$|\mathcal{N}_d(\mathbf{x})|$ = the number of grid-points in $\mathcal{N}_d(\mathbf{x})$, and $I(\tilde{\mathbf{x}} \in \mathcal{N}_d(\mathbf{x})) = 1$ if $\mathbf{x} \in \mathcal{N}_d(\mathbf{x})$, otherwise equal to 0.

The size of the neighborhood $\mathcal{N}_d(\mathbf{x})$, given by d, affects the efficiency of the location proposal. If d is too small, the resulting chain does not mix well and in addition can also get "stuck" sampling in the vicinity of a local posterior mode. If d is too large, large number of proposals gets rejected, but the chain is less likely to get stuck around a local posterior mode. The approach we take is to select randomly the neighborhood size d among three values, $d = 0.1, 0.3, 2$, with probability of selecting each equal to $2/7, 4/7, 1/7$, respectively. Hence, if a source location proposal is made, a neighborhood-size parameters d is first drawn randomly, then a location is selected randomly from $\mathcal{N}_d(\mathbf{x})$.

The benefits of working with a source location lattice is in terms of reduced number of INPUFF runs needed, as one can store the results for each grid-location by storing the values $\{\hat{G}_{\mathbf{x},t}(\mathbf{m}_j, t')\}$ for each grid-location $\mathbf{x}$.[4] The drawback of the lattice approach is that we cannot distinguish between source locations within a 0.1×0.1 pixel. In practice, the resolution of the lattice can be linked to the accuracy of the dispersion simulation program; a less accurate dispersion simulator can work on a coarser grid.

The proposal distribution for the source release rates, $\mathbf{s}_{1:2}$, is slightly more involved and is a two-step mixture; either propose a change in the start time of the release or propose a change to the current non-zero release rates (or propose both at the same time).

A change in the start time (t^*) is simply accomplished via random-walk to a nearest neighbor. Since $\mathbf{s}_{1:2}$ is only of length two, it is just an issue if the release started in the first time interval or the second time interval. Let $\mathbf{s}_{1:2} = (s_1, s_2)$ be the current release rate. The change of start-time proposal is given by:

Release-Rate Start-Time Proposal

(1) If $t^* = 1$, that is if $s_1 > 0$, then $\tilde{t}^* = 2$ is proposed with

$$\tilde{\mathbf{s}}_{1:2} = (\tilde{s}_1 = 0, \tilde{s}_2 = s_2),$$

yielding $q_2(\tilde{t}^* = 2, \tilde{\mathbf{s}}_{1:2} \,|\, t^* = 1, \mathbf{s}_{1:2}) = 1$.

(2) If $t^* = 2$, that is if $s_1 = 0$, then $\tilde{t}^* = 1$ is proposed with

$$\tilde{\mathbf{s}}_{1:2} = (\tilde{s}_1, \tilde{s}_2 = s_2), \quad \text{where } \tilde{s}_1 \sim \text{Gau}(0, 5^2)\big|_0^\infty,$$

yielding $q_2(\tilde{t}^* = 1, \tilde{\mathbf{s}}_{1:2} \,|\, t^* = 2, \mathbf{s}_{1:2}) = \varphi(\tilde{s}_2; 0, 5^2)\big|_0^\infty$.

A change to the non-zero release rates is proposed via random-walk as follows:

[4] Actually, we are able to get $\{\hat{G}_{\mathbf{x},t}(\mathbf{m}_j, t') : \mathbf{x} = \text{all grid points},\ t \leq t'\}$ in a single 'reverse' INPUFF-run for each value of $(\mathbf{m}_j, t')$; $j = 1, \ldots, 5$, $t' = 1, 2$. Hence, this requires only a total of 10 INPUFF runs.

Non-Zero Release-Rate Proposal

(1) Propose to change n_c non-zero release rates, where

$$n_c = 1 + \Delta_c,\ \ \Delta_c \sim \mathrm{Bin}(N_c, \pi_c),$$

where $\mathrm{Bin}(N_c, \pi_c)$ denotes a binomial distribution on $\{0, \dots, N_c\}$, $N_c = t - t^*$ and $\pi_c \in (0, 1]$ is the rate parameter. Note, if $t = t^* = 2$, then $n_c = 1$, but if $t^* = 1$, either one or two release rates are changed according to the rate parameter π_c.

(2) Given the number of release rates to change (n_c), select randomly among the non-zero release rates which one to change and let $\{t_{c,j} : j = 1, \dots, n_c\}$ index the selected time periods.

(3) For $j = 1, \dots, n_c$, make the random-walk proposal,

$$\tilde{s}_{t_{c,j}} \sim \mathrm{Gau}(s_{t_{c,j}}, \tau_j^2)\big|_0^\infty,$$

where the standard deviation τ_j specifies the "step-size". The τ_j's are selected randomly from the set $\{1, 3, 9\}$ with probability $\{2/7, 4/7, 1/7\}$, respectively. Hence, each random-walk is carried out with different step-size.

The proposal density is then given by

$$q_2(\tilde{\mathbf{s}}_{1:2} \,|\, \mathbf{s}_{1:2}) = \prod_{j=1}^{n_c} \varphi(\tilde{s}_{t_{c,j}}; s_{t_{c,j}}, \tau_j^2)\big|_0^\infty,$$

and note that we consider n_c, $\{t_{c,j}\}$, and $\{\tau_j\}$ fixed; that is, the reverse proposal density $q_2(\mathbf{s}_{1:2} \,|\, \tilde{\mathbf{s}}_{1:2})$ is computed with the same numbers.

When a decision is made to make a source release change, a random draw is made as to: (1) make a change to the start-time, (2) make a change to the non-zero release rates, or (3) make a simultaneous change to the start-time and non-zero releases. The probability assigned to these three types of proposals is 1/12, 10/12 and 1/12. That is, most of the time a non-zero release rate proposal is made.

The MCMC proposal step then alternates in a random fashion between making (1) a source location proposal, (2) making a source release rate proposal, or (3) make both source location and release rate proposals. An equal probability was assigned to the three different types.

Six different MCMC samples, each of size 10,000, were generated using the above proposal process. All six chains were initialized with the release rate $\mathbf{s}_{1:2}^{(0)} = (0.1, 0.1)$, but at six different locations:

$$\mathbf{x} \in \{(4,1),\ (4,3),\ (4,5),\ (1,1),\ (1,2),\ (1,5)\}.$$

The acceptance rate for each chain was about 20% (this low acceptance rate is expected as the proposal process has a number of save-guard sub-proposals steps that have a very low change of being accepted when performed, but can potentially move the chain across low-probability barriers).

For the first 500 iterations ($i = 1, \ldots, 500$), the likelihood was taken to be given by

$$p(\mathbf{c}_{1:2} \,|\, \boldsymbol{\theta}_{1:2})^{1/T_i},$$

where $T_i = 1 + (10 - i \times (10/500))$ and often referred to as the annealing temperature; see, for example Liu (2001), chapter 10. This causes the true likelihood (and hence, the data) to be brought in "slowly" as a high value of T results in a "flatter" likelihood (heated likelihood). This annealing process is well known technique to escape from a bad initial values, for example, one that is located in the vicinity of a local posterior mode of a low probability mass.

Figure 4 summarizes the MCMC output from the chain initialized at location (4,3) and the chain initialized at location (1,2) — the first 500 iterations were discarded (recall those use the "heated" likelihood). Both chains quickly fixates on realizations with $s_1 = 0$ (which was how the synthetic data was generated), but the data seams to provide little information on the release rate in the second time period, as it is seen to vary widely. Most realizations for the source location form a half-circle upwind from the sensor reporting the only non-zero concentration. The second chain, initialized at source location (1,2), generates in the beginning source location realizations that are clustered together in the lower-left corner. This cluster of realizations has lower posterior probability compared to the main cluster, as can be seen from the trace plot of the log-posterior (a log-posterior difference of about 5 translates into posterior density ratio of about 150).

The six chains were combined to form a single posterior sample, with the first 1/3 of each chain discarded as a burn-in period. Figure 5 shows two maps of the marginal posterior distribution of the source location. It shows a half-circle shaped distribution upwind from the only sensor reporting a non-zero concentration. The true location of the source is at the edge of the posterior distribution.

Figure 6 shows the posterior marginal distribution of the release rate in the two time periods. The release rate for the first time period is estimated to have $p(s_1 = 0 \,|\, \mathbf{c}_{1:2}) = 0.89$; that is, most likely no release in the first time period (which is the case). However, the data is not very informative for the release in the second time interval, and the marginal posterior distribution for s_2 is very close to the prior distribution; Figure 6 (left). It is often more informative to look at the posterior release rate conditional on a given location. Figure 6 shows the expected (average) non-zero release rate in the second time period for different potential source locations (those with non-zero posterior probability). As can be seen, locations further away from the sensors are associated with higher release rates than those that are closer to the sensors.

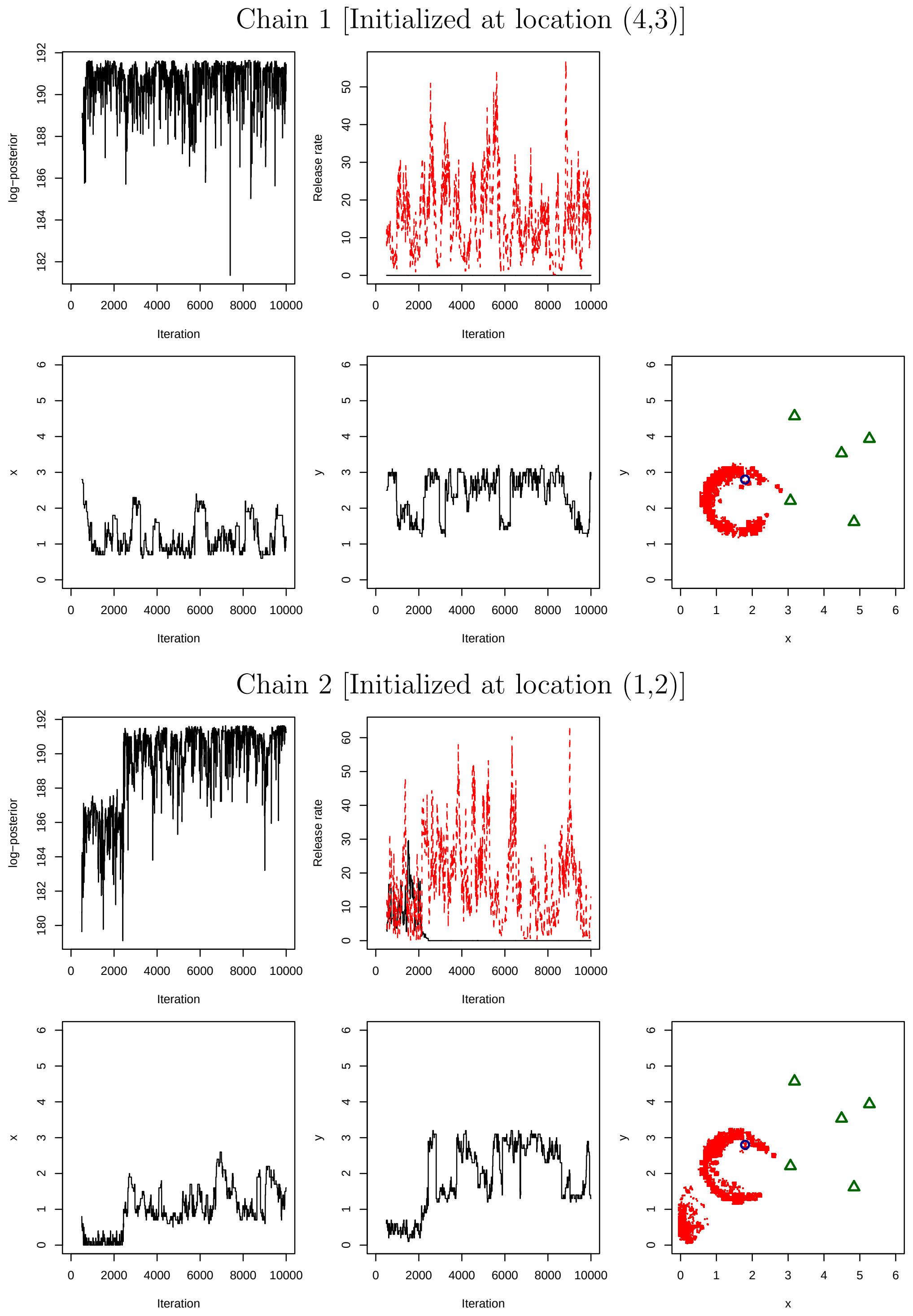


Figure 4: MCMC summary for two chains. Shown is the trace of the log-posterior distribution (up to an unknown additive constant), the trace of the release rate parameters, the trace of the x and y components of the source location, and finally a plot of the sampled source locations along with the location of the sensors and the true location of the source.

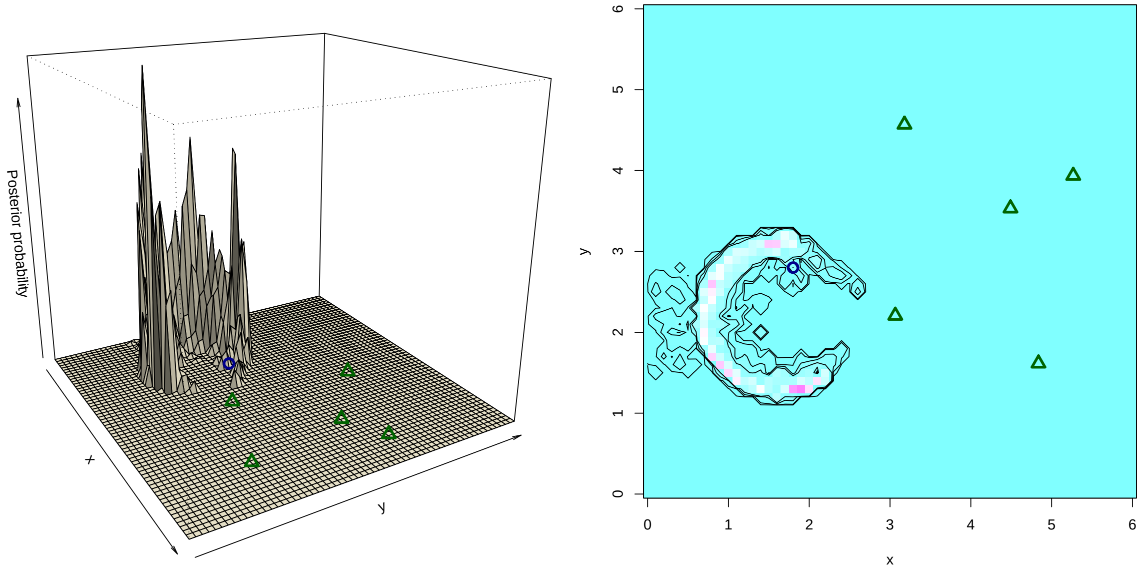


Figure 5: Left, a 3D perspective plot of the marginal posterior distribution of the source location. Right, a 2D level plot of the marginal posterior distribution of the source location (color scheme: cyan = low, magenta = high) along with contours showing the regions containing the 90%, 95%, 99%, and 99.9% of the highest posterior probability density (HPD credible sets).

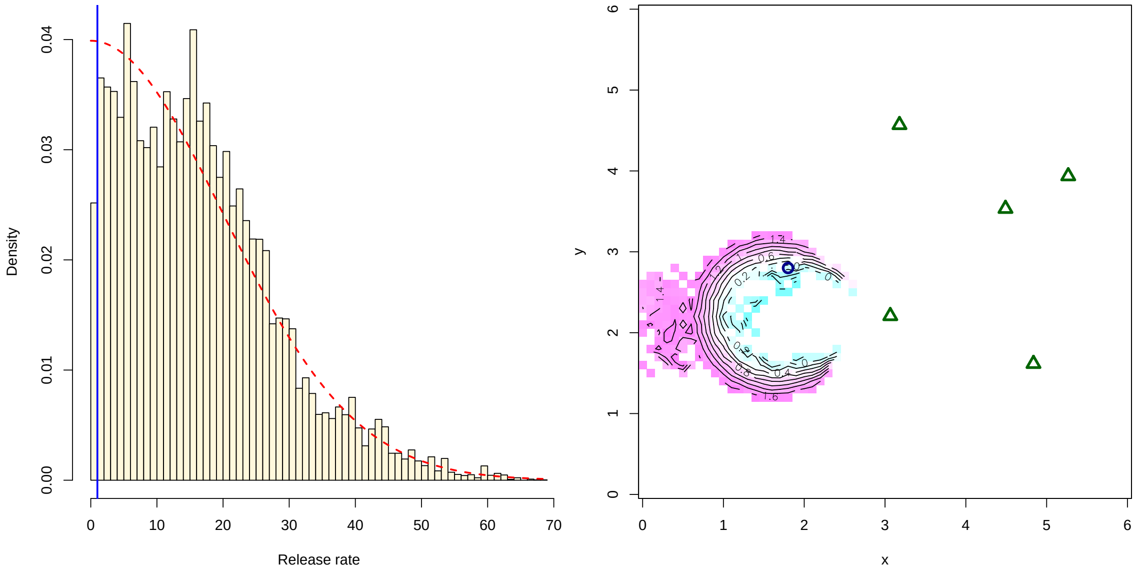


Figure 6: Left, a histogram of the posterior samples for the release rate in the second time period. The true release rate is indicated with a (blue) vertical line and the prior distribution is shown as a (red) dotted line. Right, the expected (average) non-zero release rate, $\log_{10}$-transformed, in the second time period conditional on location (i.e., $\log_{10}$ average release-rate at each location pixel).

MCMC at $t = 3$

As new data arrives in the third time period ($t = 3$), one can carry out a new MCMC for posterior inference or use SMC, using the MCMC sample from $t = 2$ as the initial sample. We shall now carry out a MCMC posterior sampling for $t = 3$ (starting from scratch), but later one we shall use SMC for the same purpose.

We applied the same proposal process at $t = 3$ as at $t = 2$, with obvious extensions to make it applicable for three time periods. A short initial run was carried out to fine-tune the proposal distributions (i.e., step-size of random-walk samplers, etc.), then six different chains were sampled, as in the case for $t = 2$.

Figure 7 summaries the result for two of the six chains in the same way as in Figure 4. There is considerable more non-zero concentration sensor-observations available at $t = 3$ that yield stronger posterior information. We see that one of the chains in Figure 7 quickly converges while the other one needs approximately 4,000 iterations to stabilize.

We combined the samples from the six chains after discarding the first half of each chain (a rather conservative approach). Figure 8 shows the marginal posterior distribution of the source location and the marginal distribution of the source release-rate in the second time period (s_2). There is a much stronger posterior knowledge about the source location at this time. Similarly, the marginal posterior distribution for the release rate at $t = 2$ is rather peeked with the true release rate close to the posterior peek. The release rate at $t = 1$ is estimated to be equal to 0 with 99.97% probability. However, not much is known about the release rate at $t = 3$ (as expected).

SMC

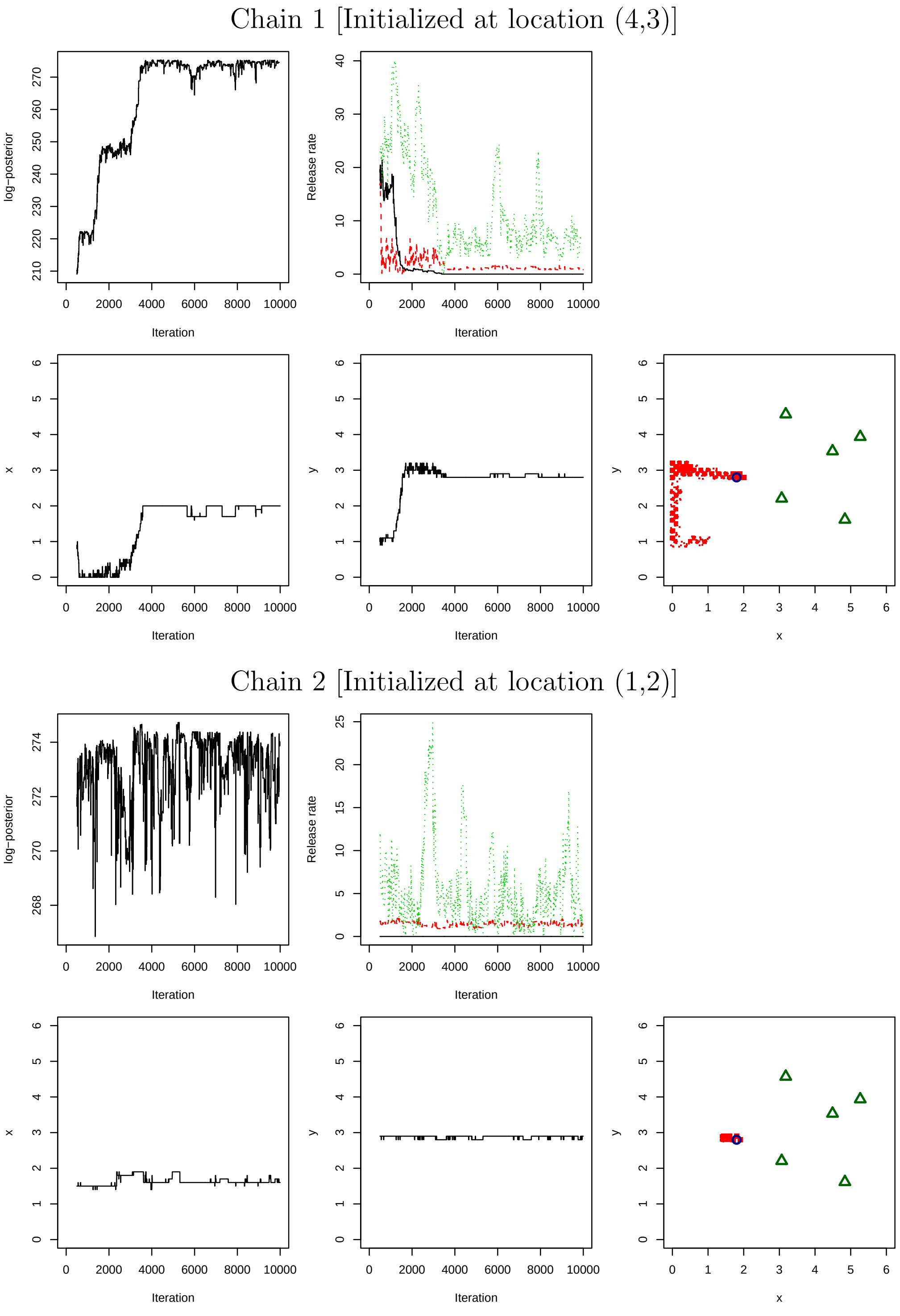
We shall now carry out SMC for $t = 3, \ldots, 6$ using the last 6,667 MCMC realizations (the first 3,333 discarded as a burn-in period) from each of the six chains at $t = 2$ as the initial posterior sample;

$$\Theta_{1:2} = \{\boldsymbol{\theta}_{1:2}^{(i)} : i = 1, \ldots, 40{,}002\},$$

where the realizations are all of equal weight. We shall use Pitt's and Shephard's (P&S) modification of Gordon's bootstrap filter, as outlined in Section 4.3, with the addition of performing MCMC perturbation within each SMC cycle, as outlined in Section 4.5 on hybrid methods. The SMC-MCMC algorithm applied is given in Table 5, with details on the various proposal distribution used to follow.

The likelihood, $p(\mathbf{c}_t \,|\, \boldsymbol{\theta}_{1:t})$, is as in (9) and (36);

$$p(\mathbf{c}_t \,|\, \boldsymbol{\theta}_{1:t}) = \prod_{j=1}^{5} \varphi(c_{j,t}; \hat{C}(\mathbf{m}_j, t), (2\text{E-}9 + 0.2 \times \hat{C}(\mathbf{m}_j, t))^2)\big|_0^\infty .$$



Chain 1 [Initialized at location (4,3)]

Chain 2 [Initialized at location (1,2)]

Figure 7: MCMC summary for two chains of six at $t = 3$; see Figure 4 for the content of each plot.

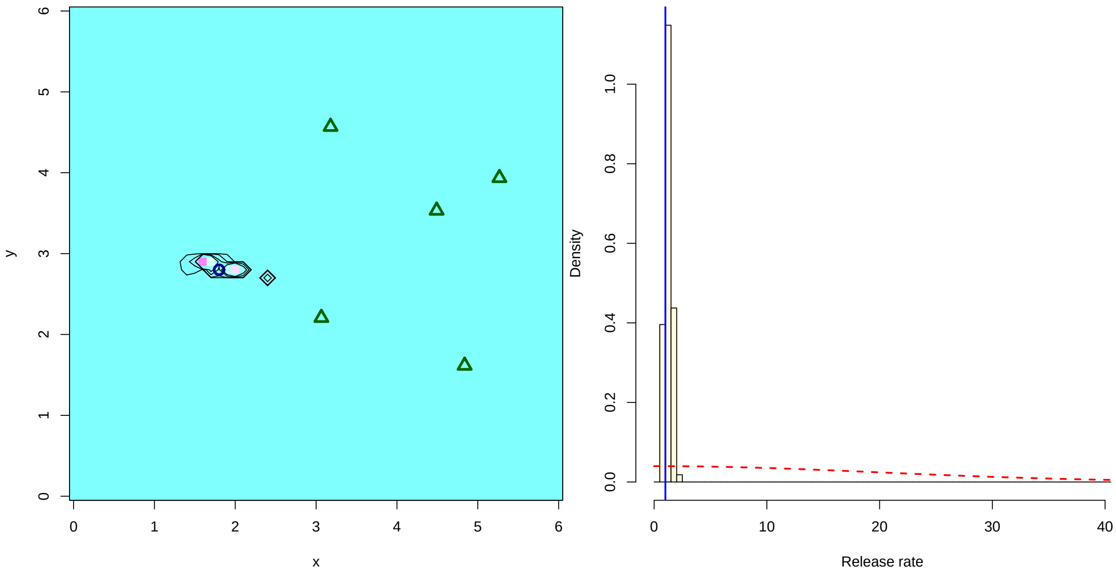


Figure 8: Left, a 2D level plot of the marginal posterior distribution of the source location (color scheme: cyan = low, magenta = high) along with contours showing the regions containing the 90%, 95%, 99%, and 99.9% of the highest posterior probability density (HPD credible sets). Right, a histogram of the posterior samples for the release rate in the second time period, with the true release rate shown as a (blue) vertical line and the prior distribution shown as a (red) dotted line.

Similarly, the conditional prior for $\boldsymbol{\theta}_t$, $p(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{1:t-1})$, is as in the MCMC case for non-zero releases, and given by

$$p(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{1:t-1}) = p(\mathbf{x}_t, s_t \,|\, \mathbf{x}_{t-1}) = \begin{cases} \varphi(s_t; 0, 20^2) & \text{if } \mathbf{x}_t = \mathbf{x}_{t-1}, \\ 0 & \text{otherwise.} \end{cases} \tag{38}$$

Note, this is a very vague conditional prior on the release rate as it does not depend on $\mathbf{s}_{1:t-1}$ at all.

The proposal distribution $q_t(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{1:t-1})$ in Table 5 is given by

$$q_t(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{1:t-1}) = q_t(\mathbf{x}_t, s_t \,|\, \mathbf{x}_{t-1}, s_{t-1}) = \begin{cases} \varphi(s_t; s_{t-1}, 10^2)\big|_0^\infty & \text{if } \mathbf{x}_t = \mathbf{x}_{t-1}, \\ 0 & \text{otherwise.} \end{cases}$$

Hence, it proposes no change in location (as expected and in accordance to our model) and then s_t is generated from a Gaussian distribution with mean s_{t-1} and standard deviation 10, and constrained to the interval $[0, \infty)$.

For the MCMC perturbation step in Table 5 we used a similar proposal process as in the MCMC-only application previously for $t = 2$ and $t = 3$. That is, a random choice is made to carry on: (1) a proposal to change the source release, (2) a proposal to change the source location, or (3) both. The probability assigned to these three proposals is 1/6, 4/6, and 1/4, respectively.

The source-release proposal consists of a random-walk proposal for a selected source-release time period. Let $\boldsymbol{\theta}_{1:t}^{(i,j)} = (\mathbf{x}^{(i,j)}, \mathbf{s}_{1:t}^{(i,j)})$ be the current value of the Markov chain, then:

Release-Rate Proposal

(1) Select a time period $\check{t}$ from $\{t-2, t-1, t\}$, with probability 1/4, 2/4, and 1/4 of selecting each period, respectively.

(2) Generate $\check{s}_{\check{t}} \sim \text{Gau}(s_{\check{t}}^{(i,j)}, \sigma_{\check{t}}^2)\big|_0^\infty$ and put the remaining release rates of $\check{\mathbf{s}}_{1:t}$ identical to those of $\mathbf{s}_{1:t}^{(i,j)}$.

The standard deviations (SD) used in the release-rate proposal above were given by,

$$\sigma_k = 0.75 \times \{\text{the empirical SD of } \{\check{s}_k^{(i,0)} : i = 1, \ldots, N\}\},$$

but never taken less than 0.1^2.

The proposal for the source location was taken to be a random-walk to a grid-point within a horizontal or vertical distance of 0.2 from the current location; that is, as the proposal given in (37).

The SMC results are summarized in Figures 9–11, for the time periods 1–3, 1–4, and 1–6, respectively. Each figure shows the marginal posterior distribution of the source location and the marginal posterior distribution the release rate for the three most recent time periods in each case. As expected, as more data is gathered, the marginal posterior distribution of the source locations narrows around the true location of the source. Similarly, as more data is processed, we gain better knowledge about the source release-rate history. Note how the posterior distribution of the release rate in the most recent time period in each case gets more informative (narrower) at later time periods. This is due to a narrower posterior distribution for the source location, which limits what the potential release rates in the latest periods could be, given the data.

MCMC versus SMC

Both MCMC and SMC samples were generated for posterior inference at $t = 3$; see Figure 8 and Figure 9, respectively. We notice a slight difference in the shape of the highest posterior density (HPD) regions constructed for the source location based on the two methods. However, the extent of the HPDs regions are very similar for both methods. The SMC-based posterior distribution of s_2 in Figure 9 seams to be slightly narrower than the one shown in Figure 8 and based on the MCMC sample. In general, we believe that the SMC sample gives a better representation of the posterior than the MCMC sample; the SMC sample consist of a slightly larger number of realizations (about 40,000 versus 30,000), but more importantly, it is a better mixed sample since *each* SMC realizations is independently perturbed via 10 MCMC iterations.

One might get the impression that the SMC algorithm needs considerable more computation time than the MCMC algorithm since, in addition to extending the past SMC realizations from time $t = 2$ to $t = 3$, it performs 10 MCMC iteration for each SMC realization (a total of 400,000 MCMC iterations). However, this is not the case. As the SMC is not a sequential algorithm (like the MCMC algorithm), one can take advantage of highly optimized vectorized computer operations that operate on all the realizations at once. In fact, our current (serial) prototype implementation of the SMC algorithm ran faster than the MCMC implementation. In addition, it is relatively easy to implement the SMC algorithm to effectively use a computer with a large number of CPUs (e.g., a Linux cluster), while this is not the case for the MCMC algorithm.

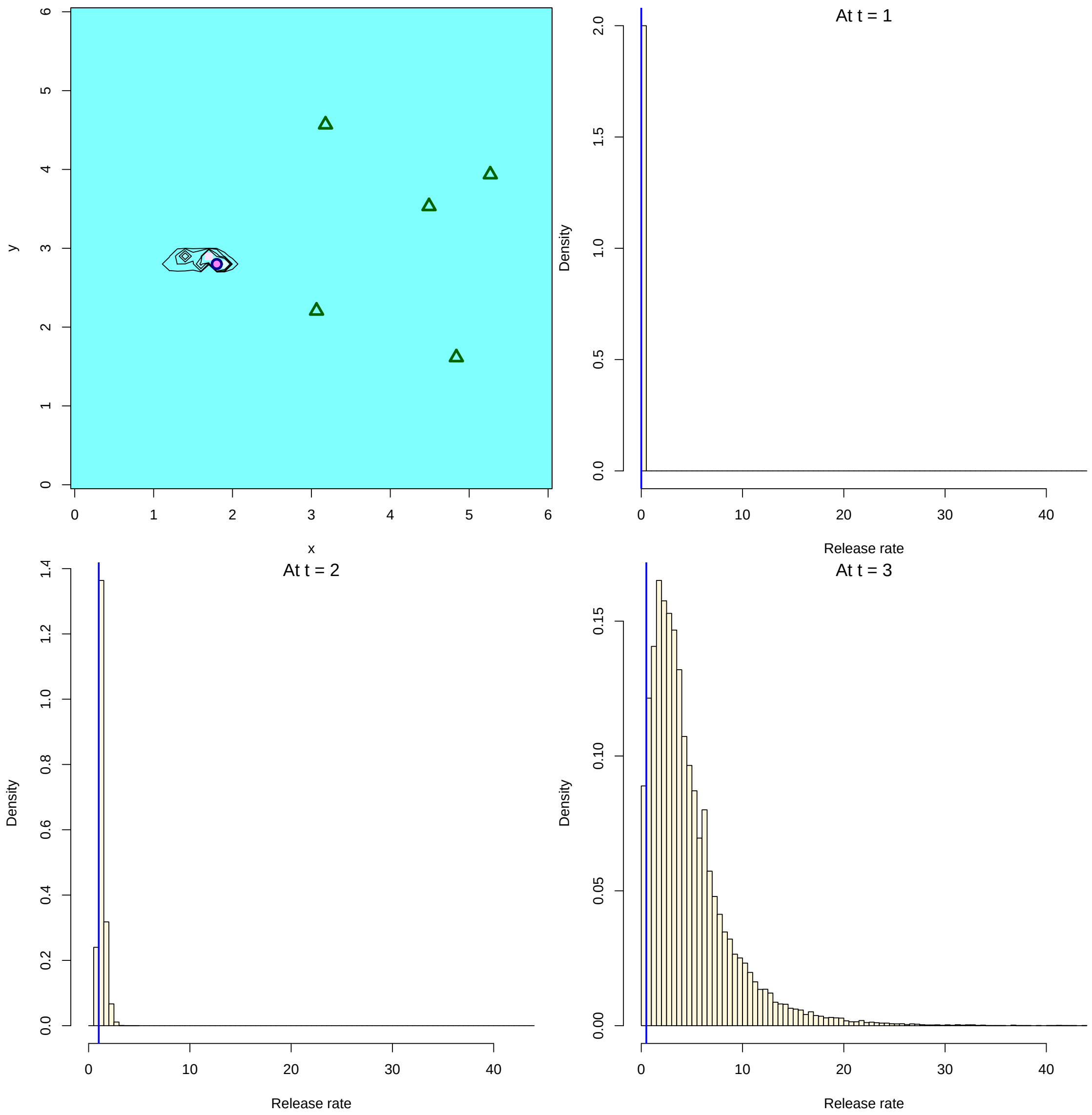


Figure 9: SMC posterior inference after processing data from time periods 1–3. Top-left, a 2D level plot of the marginal posterior distribution of the source location (color scheme: cyan = low, magenta = high) along with contours showing the regions containing the 90%, 95%, 99%, and 99.9% of the highest posterior probability density (HPD credible sets). The remaining plots show a histogram of the posterior samples for the release rate at $t = 1, 2, 3$, with the true release rate shown as a (blue) vertical line (note different horizontal scale).

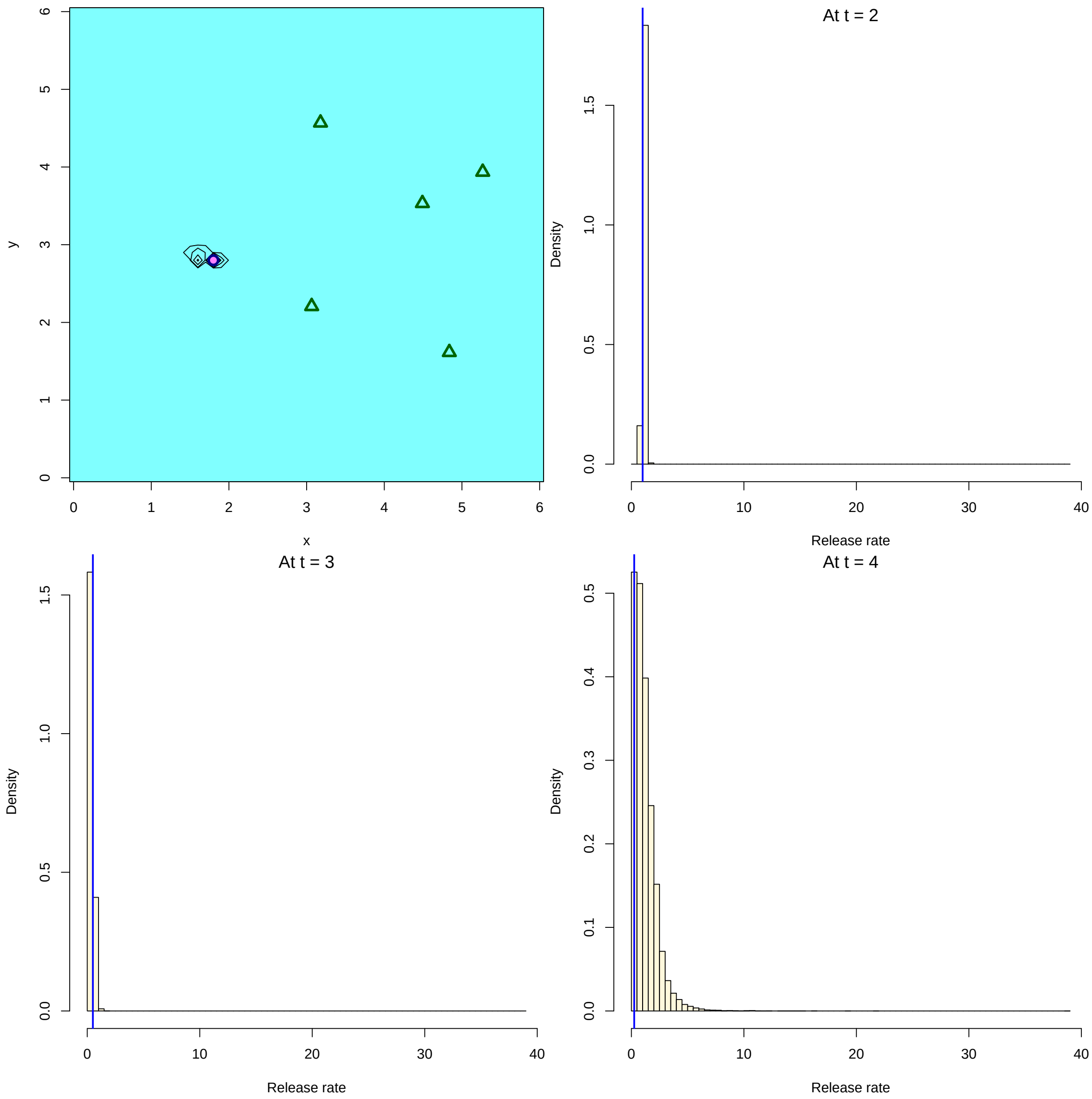


Figure 10: SMC posterior inference after processing data from time periods 1–4. Top-left, a 2D level plot of the marginal posterior distribution of the source location (color scheme: cyan = low, magenta = high) along with contours showing the regions containing the 90%, 95%, 99%, and 99.9% of the highest posterior probability density (HPD credible sets). The remaining plots show a histogram of the posterior samples for the release rate at $t = 2, 3, 4$, with the true release rate shown as a (blue) vertical line (note different horizontal scale).

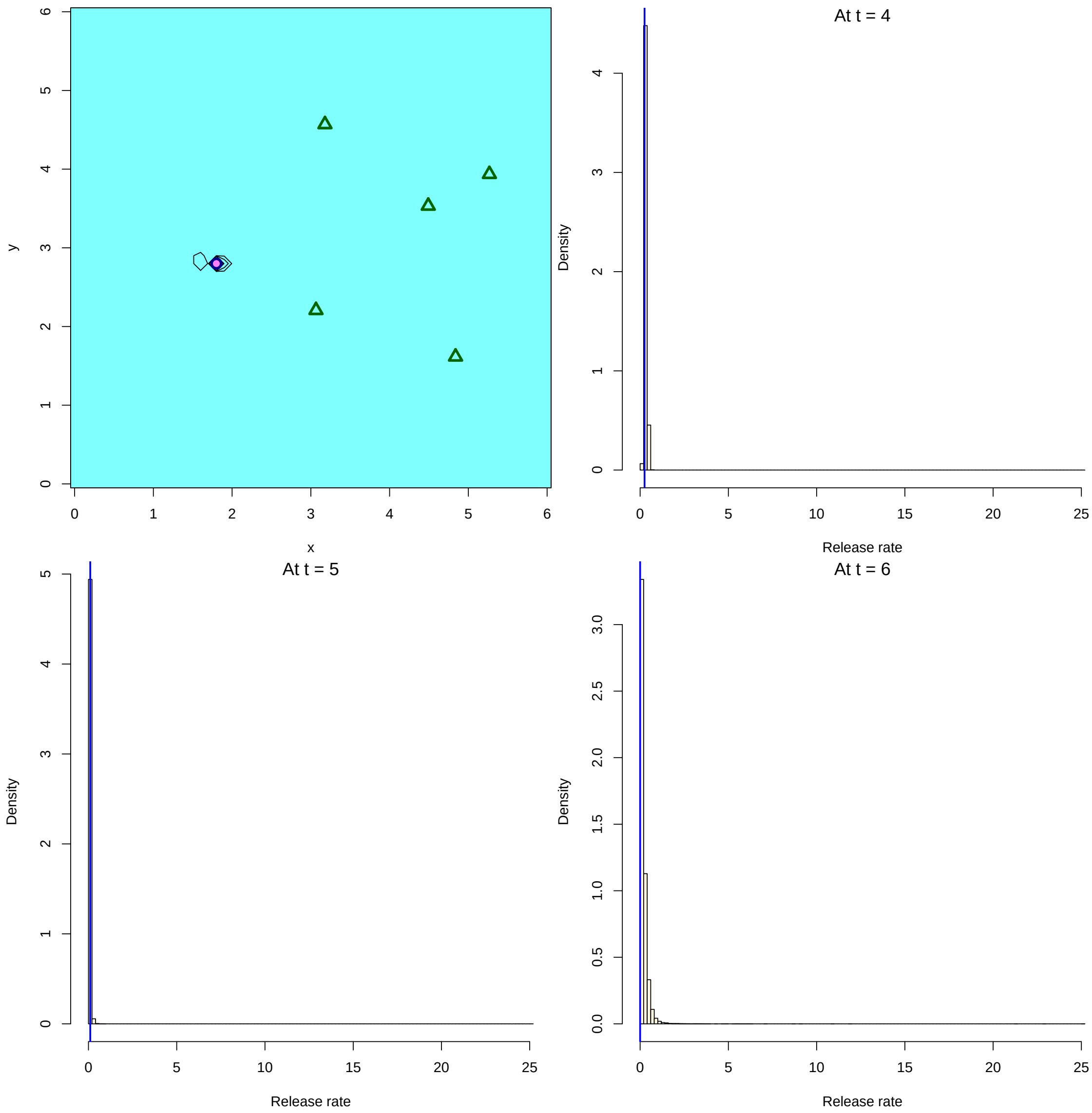


Figure 11: SMC posterior inference after processing data from time periods 1–6. Top-left, a 2D level plot of the marginal posterior distribution of the source location (color scheme: cyan = low, magenta = high) along with contours showing the regions containing the 90%, 95%, 99%, and 99.9% of the highest posterior probability density (HPD credible sets). The remaining plots show a histogram of the posterior samples for the release rate at $t = 4, 5, 6$, with the true release rate shown as a (blue) vertical line (note different horizontal scale).

Table 5: SMC-MCMC Algorithm for Atmospheric Event Reconstruction.

The hybrid SMC-MCMC algorithm used to generate samples from the posterior at times $t = 3, \ldots, 6$ in the atmospheric reconstruction application. Details on proposal distributions provided in text.

```
Initial Sample: Start with the initial, equal-weighted sample $\{\boldsymbol{\theta}_{1:2}^{(i)} : i = 1, \ldots, N\}$, $N =$ 40,002, derived from the initial MCMC samples.

For $t = 3, \ldots, 6$: (Looping through the time periods)

    Proposal Weights (P&S): For $i = 1, \ldots, N$:
        (1) Put $\hat{\boldsymbol{\theta}}^{(i)} = 0$.
        (2) Compute $v_{1:t-1}^{(i)} = p(\mathbf{c}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(i)}, \hat{\boldsymbol{\theta}}_t^{(i)})^2$. ["heated" likelihood.]
    Extending to time $t$: For $i = 1, \ldots, N$:
        (1) Sample $\tilde{I}_i \in \{1, \ldots, N\}$ with $p(\tilde{I}_i = j) \propto v_{1:t-1}^{(j)}$; $j = 1, \ldots, N$.
        (2) Generate $\tilde{\boldsymbol{\theta}}_t^{(i)} \sim q_t(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{1:t-1}^{(\tilde{I}_i)})$ and let $\tilde{\boldsymbol{\theta}}_{1:t}^{(i)} \equiv (\boldsymbol{\theta}_{1:t-1}^{(\tilde{I}_i)}, \tilde{\boldsymbol{\theta}}_t^{(i)})$.
        (3) Compute the importance-sample weight

            $$\tilde{w}_{1:t}^{(i)} = \frac{p(\mathbf{c}_t \,|\, \tilde{\boldsymbol{\theta}}_{1:t}^{(i)}) p(\tilde{\boldsymbol{\theta}}_t^{(i)} \,|\, \tilde{\boldsymbol{\theta}}_{1:t-1}^{(i)})}{q_t(\tilde{\boldsymbol{\theta}}_t^{(i)} \,|\, \boldsymbol{\theta}_{1:t-1}^{(i)})} \frac{1}{v_{1:t-1}^{(i)}}.$$ [recall, $w_{1:t-1}^{(i)} \propto 1$.]

    MCMC Perturbation: For $i = 1, \ldots, N$:
        Selection: Select $\tilde{I}_i \in \{1, \ldots, N\}$ with $p(\tilde{I}_i = j) \propto \tilde{w}_{1:t}^{(j)}$; $j = 1, \ldots, N$, and put $\boldsymbol{\theta}_{1:t}^{(i,0)} = \tilde{\boldsymbol{\theta}}_{1:t}^{(\tilde{I}_i)}$.
        MCMC Loop: For $j = 1, \ldots, B$: [$B = 10$ used.]
            (1) Propose $\check{\boldsymbol{\theta}}_{1:t} \sim q_t(\check{\boldsymbol{\theta}}_{1:t} \,|\, \boldsymbol{\theta}_{1:t}^{(i,j-1)})$.
            (2) Compute the M-H ratio

                $$\rho_t(\check{\boldsymbol{\theta}}_{1:t}; \boldsymbol{\theta}_{1:t}^{(i,j-1)}) = \frac{p(\mathbf{c}_{1:t} \,|\, \check{\boldsymbol{\theta}}_{1:t}) p(\check{\boldsymbol{\theta}}_{1:t}) q_t(\boldsymbol{\theta}_{1:t}^{(i,j-1)} \,|\, \check{\boldsymbol{\theta}}_{1:t})}{p(\mathbf{c}_{1:t} \,|\, \boldsymbol{\theta}_{1:t}^{(i,j-1)}) p(\boldsymbol{\theta}_{1:t}^{(i,j-1)}) q_t(\check{\boldsymbol{\theta}}_{1:t} \,|\, \boldsymbol{\theta}_{1:t}^{(i,j-1)})}.$$

            (3) Generate $u \sim$ Unif$[0,1]$ and put $\boldsymbol{\theta}_{1:t}^{(i,j)} = \check{\boldsymbol{\theta}}_{1:t}$ if $\rho_t(\check{\boldsymbol{\theta}}_{1:t}; \boldsymbol{\theta}_{1:t}^{(i,j-1)}) > u$, otherwise put $\boldsymbol{\theta}_{1:t}^{(i,j)} = \boldsymbol{\theta}_{1:t}^{(i,j-1)}$.
        Collect: Put $\boldsymbol{\theta}_{1:t}^{(i)} = \boldsymbol{\theta}_{1:t}^{(i,B)}$, then $\{\boldsymbol{\theta}_{1:t}^{(i)} : i = 1, \ldots, N\}$ is an equal-weighted sample from $\pi_t(\boldsymbol{\theta}_{1:t})$.
```

References

Andrieu, C., De Freitas, N., Doucent, A., & Jordan, M. I. (2003). An introduction to MCMC for machine learning. *Machine Learning*, *50*, 5–43.

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

Bernardo, J. M. & Smith, A. F. M. (1994). *Bayesian Theory*. Wiley.

Doucet, A., de Freitas, J. F. G., & Gordon, N. J. (2001). *Sequential Monte Carlo methods in practice*. New York: Springer-Verlag.

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2004). *Bayesian Data Analysis* (Second Edition ed.). Boca Raton, Florida: Hapman and Hall/CRC.

Gilks, W. R. & Berzuini, C. (2001). Following a moving target — Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society B*, *63*, 127–146.

Gilks, W. R., Richardson, S., & Spiegelhalter, D. J. E. (1996). *Markov Chain Monte Carlo in Practice*. Chapman & Hall.

Godsill, S. & Clapp, T. (2001). Improvement strategies for Monte Carlo particle filters. In A. Doucent, N. de Freitas, & N. Gordon (Eds.), *Sequential Monte Carlo Methods in Practice* (pp. 139–158). New York: Springer.

Gordon, N. J., Salmon, D. J., & Smith, A. F. M. (1993). A novel approach to nonlinear/non-gausian Bayesian state estimation. *IEEE Proceedings on Radar and Signal Processing*, *140*, 107–113.

Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing*. New York: Springer.

MacEachern, S. N., Clyde, M., & Liu, J. S. (1999). Sequential importance sampling for nonparametric Bayes models: The next generation. *Canadian Journal of Statistics*, *27*, 251–267.

Petersen, W. & Lavdas, L. (1986). Inpuff 2.0: A multiple source gaussian puff dispersion algorithm — user's guide. Technical Report EPA/600/8-86/024, EPA.

Pitt, M. K. & Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, *23*, 356–359.

Pitt, M. K. & Shephard, N. (2001). *Sequential Monte Carlo methods in practice*, chapter Auxiliary variable based particle filters. New-York: Springer-Verlag.

West, M. & Harrison, J. (1997). *Bayesian Forecasting and Dynamic Models (Second Edition)*. New York: Springer-Verlag.