49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition
4 - 7 January 2011, Orlando, Florida
AIAA 2011-824

A Probabilistic Approach to Inverse Convection-Diffusion

Ian S. Tobasco* and Krzysztof J. Fidkowski†

University of Michigan Dept. of Aerospace Engineering, Ann Arbor, MI, 48109

Initial condition inverse problems are ill-posed and computationally expensive to solve. We present a computational approach for solving inverse problems in the realm of one-dimensional contaminant transport. The approach employs finite differencing as a forward solver and probabilistic methods for inversion. Markov Chain Monte Carlo sampling is used to efficiently recover posterior probabilities. The results show that the Bayesian framework is a robust approach for initial condition inversion.

I. Introduction

Real-time modeling of accidental or intentional contaminant spread in urban settings is critical to homeland security and environmental safety. Knowing with good confidence where and when to evacuate prior to loss of health or even loss of life is of the utmost importance to decision making organizations. However, before such organizations can take action, knowledge must be obtained concerning both the origin and the evolution of the pollutants. The associated computational problem is inherently ill-posed, as many different inputs can give the same or very similar outputs. This problem is also computationally expensive: just modeling a forward dispersion problem in a realistic urban environment can take several hours using state-of-the-art computational fluid dynamics, and the proposed inverse problem is significantly more expensive. This paper presents a computational framework for probabilistic inversion, in which solutions are given as probability distributions for initial conditions.

This paper first presents the theory behind our approach. One-dimensional contaminant transport physics is discussed briefly and finite difference schemes used to numerically solve the linear convection-diffusion equation are detailed. The corresponding inverse problem is then posed in a Bayesian framework, and a Monte Carlo Markov Chain (MCMC) algorithm for performing efficient inversions is discussed. Results follow, in an effort to demonstrate the applicability of a probabilistic approach for inverting convective-diffusive processes.

II. Approach and Methodology

Our problem consists of solving both the forward and inverse convection-diffusion problems. While the forward problem is relatively easy to solve, the inverse problem requires finding the initial conditions which led to a set of known contaminant sensor readings from some current time. This is inherently ill-posed and computationally expensive to implement. Previous work took a fully deterministic approach, yielding a single best-fit solution to the inverse problem.[1,2] However, this can be both time-consuming and misleading, and does not account for uncertainties such as sensor measurement error. Instead, we take a probabilistic approach, in which probability distributions of the initial concentration profile are sought instead of a single solution. Our solution relies on an application of Bayesian statistical methods and utilizes MCMC algorithms. (Another approach would be to solve for the moments of the probability distributions, but this does not work well when the distributions are far from normal.) Before we can solve the inverse problem, we must develop an efficient method of producing solutions to the associated forward problem. Efficiency is critical in this first step as tens of thousands of forward solutions are typically required for generating accurate MCMC statistics.

*Undergraduate, UofM Dept. of Aerospace Engineering, 808 E Kingsley St, Ann Arbor, MI, 48104, AIAA Student Member
†Assistant Professor, UofM Dept. of Aerospace Engineering, 3029 FXB, 1320 Beal, Ann Arbor, MI, 48109, AIAA Member

A. Forward Convection-Diffusion

Consider a city, in which sensors capable of registering concentrations of contaminants have been placed strategically. We pose the following question: given the contaminant's initial concentration profile, i.e. the initial conditions, what concentrations will the sensors read after a finite amount of time? In order to solve this "forward" problem, we must first understand the physics of the contaminant transport.

In this work, we assume that the contaminant transport is governed by a linear convection-diffusion equation. Specifically, we have

$$\frac{Du}{Dt} - d\nabla^2 u = 0, \tag{1}$$

where d is the carrying medium's diffusivity and $u(\mathbf{x}, t)$ represents the concentration profile being transported. For simplicity, we'll discuss the implications of (1) in one dimension only; thus, we'll consider the following restatement of the convection-diffusion equation in the $\hat{\mathbf{x}}$ direction:

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} - d\frac{\partial^2 u}{\partial x^2} = 0, \tag{2}$$

with a being the carrying medium's velocity. Furthermore, we'll assume that both the convection velocity and the medium's diffusivity remain constant throughout the domain. The non-dimensionalized ratio of convective to diffusive transport is given by the Péclet number,

$$\mathrm{Pe} = \frac{aL}{d}, \tag{3}$$

where L is the characteristic domain length. In this work, we assume that convection dominates the contaminant transport, so that $\mathrm{Pe} \gg 1$.

To obtain numerical solutions to (2), we employ finite differencing methods, discretizing space and time into meshes. Specifically, we have

$$u(x, t) = u(i\Delta x, n\Delta t) = u_{i,n}, \tag{4}$$

where i and n belong to the natural numbers and Δx and Δt are the respective sizes of the space and time intervals. To solve (2) in this context, we chose a second order accurate discretization in both space and time, along with a second order backwards Euler integration scheme. (The $\frac{\partial u}{\partial x}$ term was discretized via the upwind method for stability purposes.) Thus, for any given (i, n) pair, equation (2) becomes

$$\frac{3u_{i,n+1} - 4u_{i,n} + u_{i,n-1}}{2\Delta t} + a\frac{3u_{i-1,n+1} - 4u_{i,n+1} + u_{i+1,n+1}}{2\Delta x} - d\frac{u_{i-1,n+1} - 2u_{i,n+1} + u_{i+1,n+1}}{\Delta x^2} = 0. \tag{5}$$

When this equation is enforced throughout the space-time domain, the problem becomes an algebraic one, which can be easily solved via software packages such as LAPACK.

Two sets of conditions must be specified in order to solve the algebraic equations produced by (5). First, we give boundary conditions, i.e. conditions to be enforced on the edges of the spatial domain. We chose periodic boundary conditions for our implementation, i.e. we have

$$u_{0,n} = u_{i_{\mathrm{tot}},n} \tag{6}$$

for all n, where i_{tot} is a prespecified number. For simplicity, we assumed that the initial contaminant concentration profile took on a Gaussian shape, such that

$$u(x, t_0) = \frac{A}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \tag{7}$$

where A is the cloud's initial amplitude, σ is the initial spread, and μ is the initial mean location. Specifying these free parameters, as well as the time-since-release (T), completely determines a solution. By time-since-release, we mean the amount of time elapsed from when the contaminants started to convect and diffuse to the unique time at which all the sensors take readings. In this manner, we can say that $t_0 = 0$, so that the time at which the sensors read values is exactly the time-since-release.

Once we have specified all free parameters, we can easily apply (5) to find $u(x, T)$ for any x within our periodic domain. To finish the forward problem, we must specify how the sensors read values at time T. Real-world sensors take data with some error, which we model as being Gaussian. If y_k represents the exact

measurement that the k^{th} sensor would recover in a perfect world, we model each recovered measurement as being given by

$$\bar{Y}_k = y_k + \mathcal{N}(0, \epsilon). \tag{8}$$

Hence, we are considering each measurement to be a random variable distributed normally about y_k with standard deviation ϵ. (For simplicity, we'll take ϵ to be constant amongst sensors.) In order to determine what values y_k obtain, we only need to decide where each sensor is placed within the domain. Several different placements are discussed in section B.

B. Inverse Convection-Diffusion

Inverse problems form a unique branch of mathematics and engineering, as each problem is inherently ill-posed and generally results in large computational requirements. Furthermore, solving each problem requires sufficient knowledge of the associated forward problem. In this paper, we'll investigate the following: given knowledge of sensor readings only, what can we say about the conditions that led to our measurements? This is a more likely scenario than the forward problem, as contaminant spread in urban settings usually restricts itself to detailed knowledge of the current time only.

We take a Bayesian approach, accepting that the inverse problem does not give us much prior knowledge. Not only is there uncertainty about the contaminants' originating circumstances, but there is also uncertainty built into the sensor readings. As characterized in (8), each sensor has some associated measurement error, which will be built into our method. A Bayesian approach is a natural choice in this situation, as Bayesian statistics yields itself to problems which require procuring information about multiple unknowns from limited prior knowledge.

We follow a similar approach as outlined in Galbally et al.[3] Treating the initial conditions as a random variable in four dimensions, so that a realization would be $\mathbf{b} = (\mu, T, \sigma, A)$, we look for the conditional distribution of the initial conditions given a vector of realized sensor measurements ($\bar{\mathbf{y}}$). According to Bayes' Theorem, we have

$$p(\mathbf{b}|\bar{\mathbf{y}}) = \frac{p(\bar{\mathbf{y}}|\mathbf{b})p(\mathbf{b})}{p(\bar{\mathbf{y}})}, \tag{9}$$

commonly referred to as the posterior probability distribution. The posterior is made of three parts: the likelihood function ($p(\bar{\mathbf{y}}|\mathbf{b})$), the prior probability ($p(\mathbf{b})$), and a normalizing constant. Our sensor model provides a Gaussian likelihood function; i.e.

$$p(\bar{\mathbf{y}}|\mathbf{b}) \ \alpha \ \exp\left(\frac{-1}{2\epsilon^2}(\bar{\mathbf{y}} - \mathbf{y}(\mathbf{b}))^T(\bar{\mathbf{y}} - \mathbf{y}(\mathbf{b}))\right). \tag{10}$$

Generally, the prior distribution is uniform over the allowable domain, such that the posterior is zero wherever prior knowledge tells us that the initial conditions could not possibly exist. Thus, taking $\mathcal{A}$ to be the allowable domain, we wish to find the following posterior:

$$p(\mathbf{b}|\bar{\mathbf{y}}) \ \alpha \ \begin{cases} \exp\left(\frac{-1}{2\epsilon^2}(\bar{\mathbf{y}} - \mathbf{y}(\mathbf{b}))^T(\bar{\mathbf{y}} - \mathbf{y}(\mathbf{b}))\right), & \mathbf{b} \in \mathcal{A} \\ 0\,, & \text{otherwise} \end{cases}. \tag{11}$$

Visualizing the posterior given above is not a simple task, as each different set of initial conditions requires solving the forward problem to find $\mathbf{y}(\mathbf{b})$. A first approach would be to fix two elements of $\mathbf{b}$ (e.g. T and A), and evaluate the further conditioned (11) on a fine mesh over the remaining initial conditions. An iso-probability plot could be constructed, and the most likely point would be the best guess of the remaining initial conditions. This "brute-force" approach is costly to perform, but would undoubtedly reveal the conditioned posterior.

A more elegant approach, which is less costly for more than two unknowns, involves using an MCMC algorithm known as the Metropolis-Hastings algorithm. In this approach, we explore the posterior probability distribution function via a Markov Chain. Following the method outlined by both Calvetti and Somersalo[4] and Galbally et al.,[3] we start with an arbitrary $\mathbf{b}_0 \in \mathcal{A}$, and proceed with the following:

1. Given $\mathbf{b}_i$, randomly select $\mathbf{b}_p$ via $q(\mathbf{b}_p|\mathbf{b}_i)$, a uniform distribution over $\mathcal{A}$.

2. Calculate the acceptance ratio, given by

$$\alpha(\mathbf{b}_p|\mathbf{b}_i) = \min\left[1,\ \frac{p(\mathbf{b}_p|\bar{\mathbf{y}})q(\mathbf{b}_p|\mathbf{b}_i)}{p(\mathbf{b}_i|\bar{\mathbf{y}})q(\mathbf{b}_i|\mathbf{b}_p)}\right]. \tag{12}$$

3. Flip the "α-coin":

   Draw t from the unit uniform distribution; If $t < \alpha$, $\mathbf{b}_{i+1} = \mathbf{b}_p$, otherwise $\mathbf{b}_{i+1} = \mathbf{b}_i$.

This algorithm will generate a Markov Chain which converges to the posterior distribution. That is, as the number of samples increases, the distribution of samples represents the posterior more and more closely. (For general distributions, a finite number of samples will only give an approximation to the posterior.) Once we have generated enough samples via the Markov Chain, we can visualize the marginal posterior distributions by plotting histograms of the random walk. Again, the set of most likely values in each marginal distribution would be the best guess of the initial conditions given the sensor measurements. Although this method still requires solving the forward problem for each $\mathbf{b}_i$, the overall computing time should be less than that of the brute-force approach, especially as the number of unknown parameters grows.

III. Implementation and Results

The approach detailed above provides a way to solve both the forward and inverse convection-diffusion problems. Although the inverse problem is inherently ill-posed, our Bayesian framework allowed us to accept the uncertainty as an integral part of the problem. In order to test our methods, we prototyped the method first in Matlab, and then implemented our solution in C for both speed and portability. Algebraic manipulation was performed via calls to LAPACK, and several tests were run concerning mesh size and sensor placement. Each test was run serially on a desktop computer with 4 GB of RAM and a 2.80 GHz Pentium Dual-Core CPU (E6300).

A. Finite Differencing

As outlined in section A, we discretized the one-dimensional convection-diffusion equation with second order accuracy on a periodic domain. For simplicity, we took the domain to have unit length, assumed that the carrying medium was convecting with a constant unit velocity, and chose a high enough Péclet number to simulate atmospheric conditions ($\mathrm{Pe} = 10^3$). Our choices of a, L, and Pe determined the medium's diffusivity via (3). As will be explained later, we determined that, for a second order accurate discretization, the space mesh could have $i_{\mathrm{tot}} = 2^8 + 1$ points and still remain accurate enough for testing. (Tests in upcoming sections contain meshes with different values of i_{tot}, but we'll treat $2^8 + 1$ as our standard mesh.) Again, our choices of L and i_{tot} determined that $\Delta x = L/i_{\mathrm{tot}} = 2^{-8}$ m.

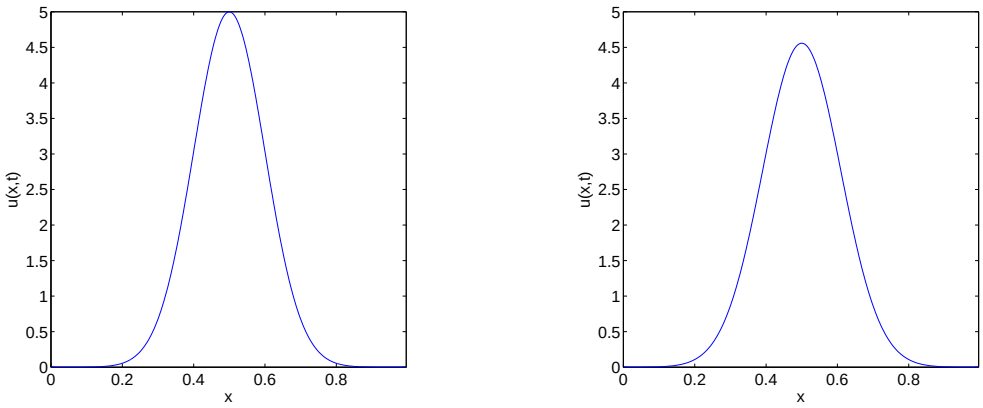
In earlier work, we employed a first order forward Euler integrator, so choosing a value for Δt required satisfying the CFL condition and ensuring that the heat number was less-than-or-equal-to a half. The most straightforward solution was to take $\Delta t = \Delta x$. For the high Péclet number in this work, both conditions are easily satisfied with this choice. Upon switching to a second order backward Euler integrator, we kept our choice of Δt purely for convenience.

While our choice of i_{tot} may seem arbitrary, it follows from our sensor model. We chose a contaminant concentration of $u = 1$ unit to represent a lethal dose and took $\epsilon = 0.01$ in (8). A quick test showed that approximately 0.0049886 units of concentration were lost due to numerical diffusion only. As the numerical diffusion was captured well within one standard deviation of the mean sensor error, we determined that $2^8 + 1$ points would present trustworthy results.

Figure 1, on the next page, shows an initial contaminant concentration profile and the corresponding forward solution after one second. To create the initial concentration profile, we used $\mu = 0.5$ m, $\sigma = 0.1$ m, and $A = 5$ units of concentration. As one second passes between Figure 1(a) and Figure 1(b), we could say that $T = 1$ s for this test. The solution presented here took 0.03 s to compute, and was found with $2^8 + 1$ points – our standard number of mesh points.

B. Sensor Numbers and Placement

The first inverse convection-diffusion tests looked at the effects on the posterior due to the number and arrangement of sensors within the periodic domain. To do so, we took a two-step approach. For any given



(a) Initial concentration profile

(b) Solution concentration profile

Figure 1. Convection-diffusion forward solution with $i_{\mathbf{tot}} = 2^8 + 1$.

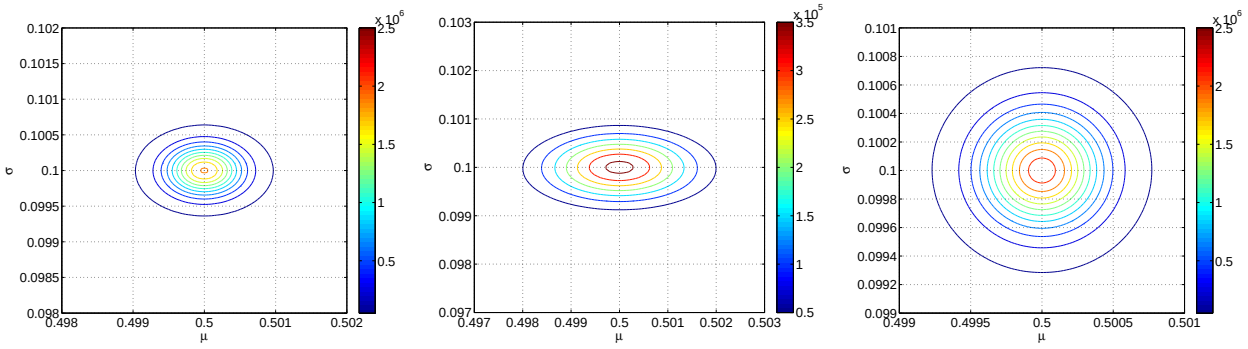
number and arrangement of sensors, we picked a set of "true" initial conditions ($\mathbf{b}_t$) and solved for $\mathbf{y}(\mathbf{b}_t)$. (From here on, we used $\mathbf{y}$ in place of $\bar{\mathbf{y}}$ to eliminate a confounding variable.) Then, we used the brute-force method outlined in section B in order to recover the posterior. This required finding $\mathbf{y}(\mathbf{b})$ over a mesh of two initial conditions: we chose μ and σ as the unknowns, fixing $T = 1$ s and $A = 5$ units of concentration.

In order to only test the effects of the number and arrangement of sensors, we knowingly commited an "inverse-crime" by having $i_{\text{tot}} = 2^8 + 1$ in both of the above steps. This would be a bad choice in any regular situation, but the results in the following section will show how much of a confounding effect using two different meshes can create. The following sections will also include tests that combine both sensor numbers and placement with a refined "exact" mesh.

We have yet to discuss the different arrangements of sensors that were used during testing. We chose four types of sensor arrangement, which we believed represent a variety of one-dimensional arrangements. These arrangements are as follows:

- Uniform – the sensors are placed equal lengths apart
- Random – the sensors are placed via the unit uniform distribution
- Left Uniform – the sensors are placed equal lengths apart within the left half of the domain
- Left Skew – the sensors are placed via an exponential distribution, with mean 0.3.

Figures 2-5 on this page and the next show the results of placing different numbers of sensors via the above distributions.

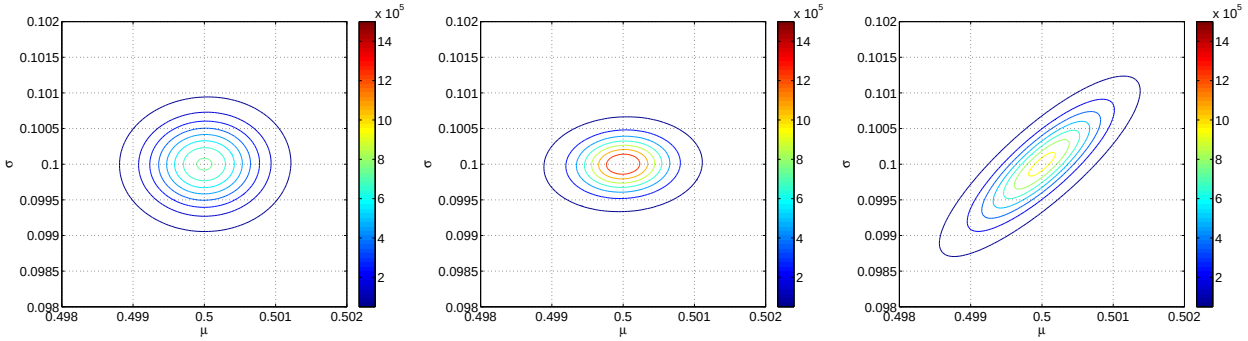


(a) 3 sensors

(b) 4 sensors

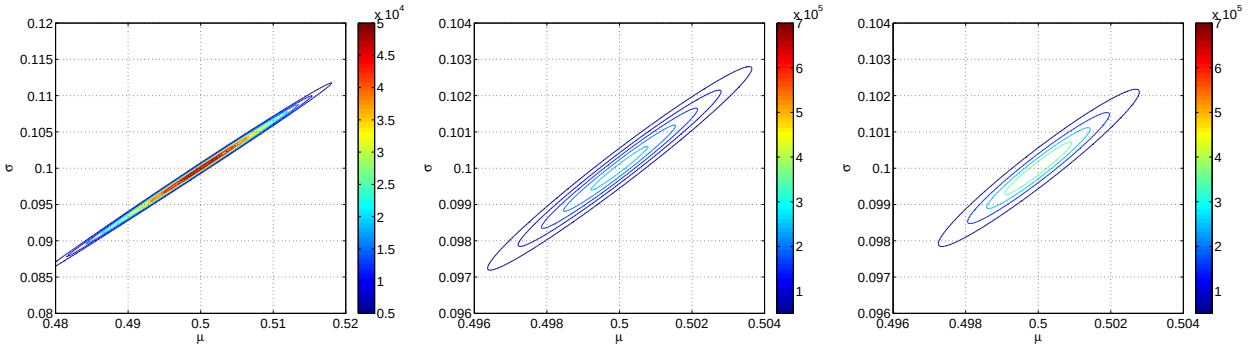
(c) 5 sensors

Figure 2. Effect of uniform sensor placement on posterior PDF.



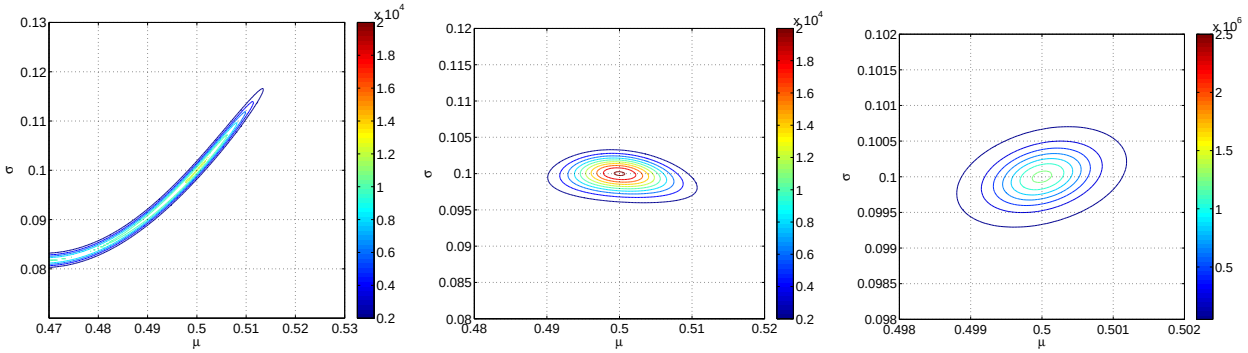
(a) 3 sensors (b) 4 sensors (c) 5 sensors

Figure 3. Effect of random sensor placement on posterior PDF.



(a) 3 sensors (b) 4 sensors (c) 5 sensors

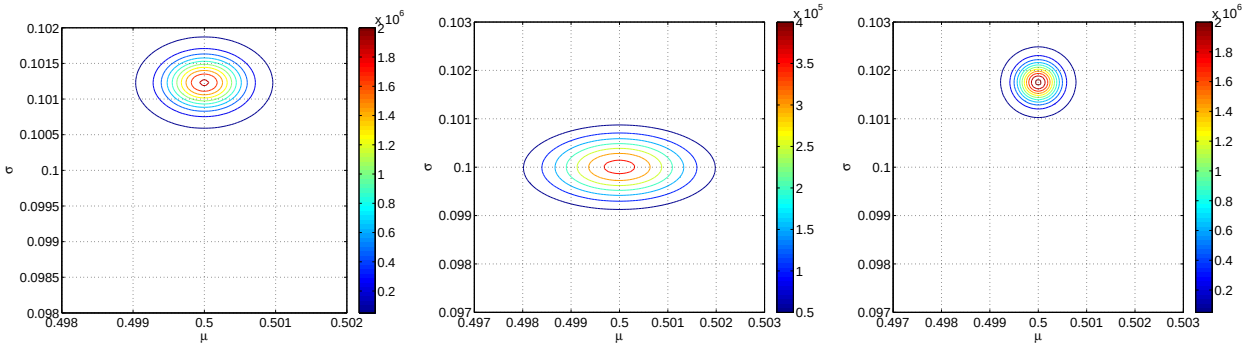
Figure 4. Effect of left uniform sensor placement on posterior PDF.



(a) 3 sensors (b) 4 sensors (c) 5 sensors

Figure 5. Effect of left skew sensor placement on posterior PDF.

Each plot was generated with true initial conditions $\mu_t = 0.5$ m, $T_t = 1$ s, $\sigma_t = 0.1$ m, and $A_t = 5$ units of concentration. Although the conditioned posteriors clearly differ from one sensor arrangement to another, each iso-probability plot (except 5(a)) is centered about the true initial conditions that we wished to recover. This is misleading, however, as we committed an inverse-crime in recovering the data. The main point of these plots is to show how much sensor arrangement can effect the posterior. We immediately concluded that the arrangements which favored one side would not work for this problem, and that uniform sensor placement may work best for performing inversions. We were not able to conclude anything about what number of sensors would work best at this point. It is interesting to note, however, that more sensors do not necessarily give better results – for an example, compare the changes in the posterior between Figures



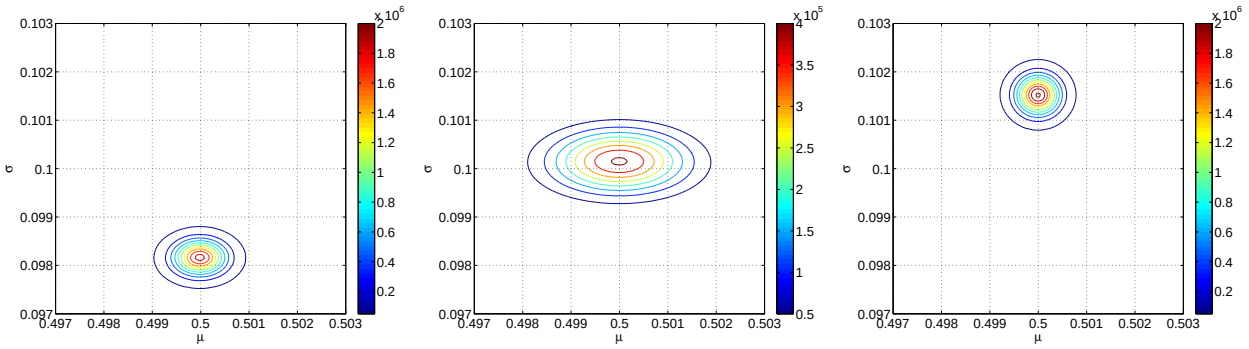
(a) 3 sensors (b) 4 sensors (c) 5 sensors

Figure 6. Effect of mesh size on posterior PDF, inversions performed with $i_{\mathbf{tot}} = 2^8 + 1$.

3(a), 3(b), and 3(c). It seems, from these figures, that arrangement plays a bigger role in yielding a reliable posterior than does the number of sensors. The brute-force results shown in this section took anywhere from 350 s to 1447 s to complete, depending on the coarseness of the grid that was used.

C. Mesh Tests

In the previous section, we knowingly committed an inverse-crime, in an attempt to isolate the effects of sensor numbers and placement. Next, we investigated the effects of using different numbers of points for the space mesh by changing i_{tot}. As previously stated, we originally found $2^8 + 1$ points to be adequate for solving the forward problem. For this group of tests, we used a refined mesh of $2^9 + 1$ points in order to procure the true sensor measurements ($\mathbf{y}(\mathbf{b}_t)$). Each posterior was then recovered by the brute-force method, using different numbers of mesh points to find $\mathbf{y}(\mathbf{b})$ for use in (11). As in the previous section, we fixed $T = 1$ s and $A = 5$ units in order to facilitate computations. Unless otherwise stated, sensors were arranged via the uniform distribution (see previous section) in each mesh-related test, although the number of sensors was changed between tests. Figures 6 and 7 show the effects on the posterior of using different meshes when performing inversions.

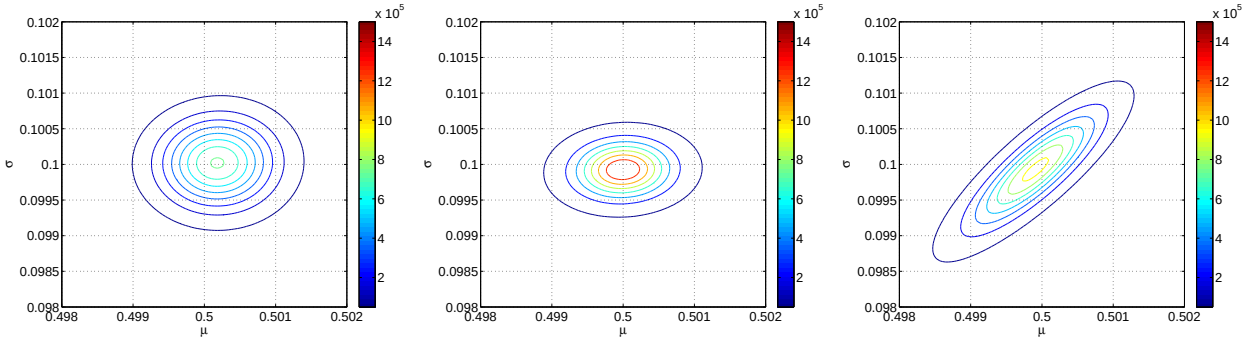


(a) 3 sensors (b) 4 sensors (c) 5 sensors

Figure 7. Effect of mesh size on posterior PDF, inversions performed with $i_{\mathbf{tot}} = 2^7 + 1$.

Just as before, each plot was generated with true initial conditions $\mu_t = 0.5$ m, $T_t = 1$ s, $\sigma_t = 0.1$ m, and $A_t = 5$ units of concentration. Eliminating the inverse-crime from our setup showed that performing inversions with coarser meshes causes the posterior to walk around parameter space. The only plot that seems unaffected by this can be seen in Figure 6(b). (Even with $i_{\text{tot}} = 2^7 + 1$ points, inverting with four uniform sensors results in a posterior that walks about much less than do the other two in Figure 7.) Upon comparing the posterior in Figure 6(b) to the similar posterior in Figure 2(b), we concluded that using four sensors in a uniform arrangement might give the best results. Before going on, however, we investigated the combined effect of inverting with a coarser mesh and arranging sensors in ways other than uniformly. Both

sensor arrangements which favored the left side gave wildly incorrect posteriors, so we have only included here plots in which sensors were arranged randomly via the unit uniform distribution.



(a) 3 sensors (b) 4 sensors (c) 5 sensors

Figure 8. Effect of mesh size and random sensor arrangement on posterior PDF, inversions performed with $i_{\mathbf{tot}} = 2^8 + 1$.

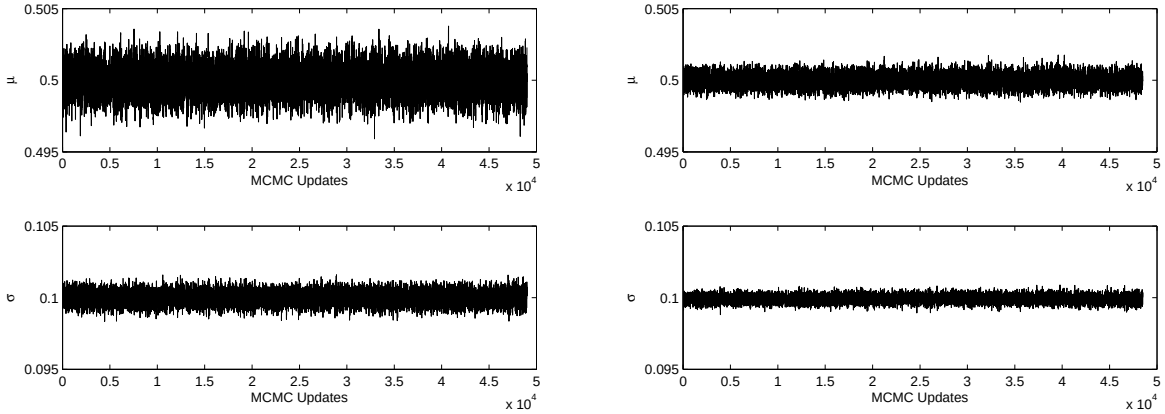
As throughout this section, true data for Figure 8 was generated with $2^9 + 1$ mesh points, along with initial conditions $\mu_t = 0.5$ m, $T_t = 1$ s, $\sigma_t = 0.1$ m, and $A_t = 5$ units of concentration. We chose to perform inversions using $2^8 + 1$ mesh points, discarding the $i_{\text{tot}} = 2^7 + 1$ case for accuracy's sake. Surprisingly, these results are on the whole better than those shown in Figure 6. The posterior means barely deviate from $(0.5, 0.1)$, and the iso-probability curves are similar to those shown in Figure 3. It seems that uniformly random sensor placement helps to compensate for any loss in information due to inverting with a coarser mesh. Perhaps a sensor arrangement which involves randomly perturbing a perfectly uniform arrangement would be closer to the optimal choice for solving inverse convection-diffusion. For the data presented in this section, the average computation time for posteriors which performed inversions using $2^8 + 1$ mesh points was 338 s, while the average time for posteriors using $2^7 + 1$ mesh points was 45 s. We do not, however, advocate inverting with less than $2^8 + 1$ points, due to accuracy issues.

D. MCMC Inversions

Thus far, we have recovered the posterior via the brute-force method only. This has limited us to viewing twice-conditioned posteriors, and has proved to be computationally expensive. Now, we investigate inverse convection-diffusion using Markov Chains. Following the Metropolis-Hastings algorithm, outlined at the end of section B, we take a random walk through parameter space which eventually should converge to the posterior distribution. As the first member of the random walk is a guess, there will most likely be some burn-in period associated with the completed Markov Chain. In this work, we discarded the burn-in period heuristically, focusing only on the part of the walk which appeared as random noise.

Validation of our MCMC approach was performed by gathering samples from conditioned posteriors, as in the previous section. As before, we set $A = 5$ units of concentration and $T = 1$ s prior to performing the inversion, so that only μ and σ would be altered in the random walk. The proposal distribution which we used to gather our samples ($q(\mathbf{b}_p|\mathbf{b}_i)$ from section B) was a uniform distribution over the allowable domain ($\mathcal{A}$). For these tests, we considered to be prior knowledge that $\mu_t \in [0, 1]$ and $\sigma_t \in [0, .5]$ – the Cartesian product of which formed the allowable domain. Furthermore, as is common practice,[4] we scaled the proposal distribution to achieve an optimal acceptance ratio of anywhere between 20% and 30%. Figures 9 and 10 on the next page show the MCMC walks and corresponding histograms after truncating the first 1,000-1,500 samples to remove burn-in.

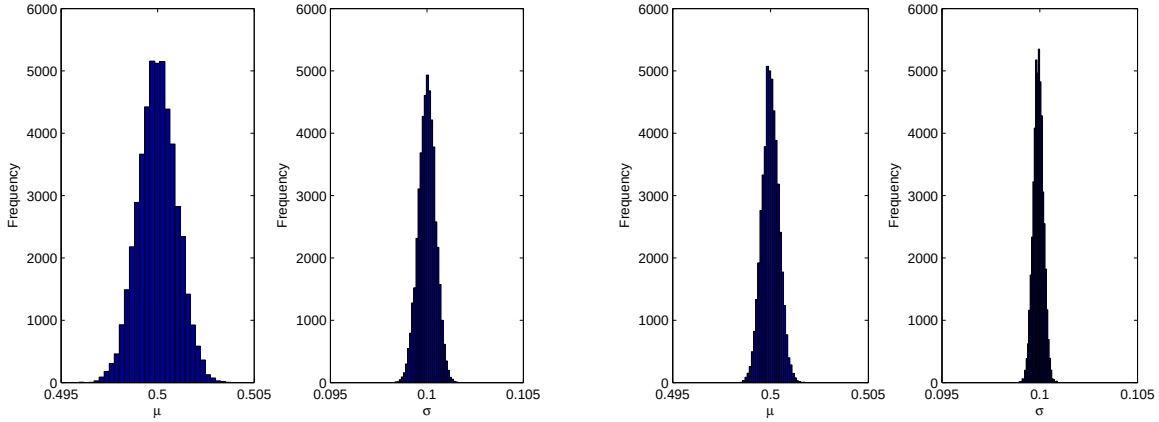
These figures show the ability of the MCMC approach to recover the same posterior as does the brute-force method (see Figures 6(b) and 8(b) for comparison). Again, we used the same true initial conditions for μ and σ as before, taking $\mu_t = 0.5$ and $\sigma_t = 0.1$. With 4 uniform sensors, our walk gave $\bar{\mu} = 0.4999$ and $\bar{\sigma} = 0.1$; the random setup walk gave $\bar{\mu} = 0.5$ and $\bar{\sigma} = 0.0999$. Note that no inverse-crime was committed in either case, so we expected the recovered means to deviate slightly from the correct values. Both two-parameter MCMC inversions performed here were 50,000 samples long, to ensure usable results. The first inversion (Figures 9(a) and 10(a)) had an acceptance rate of 0.2681 and took 1,654 s to run, whereas the second (Figures 9(b) and 10(b)) had an acceptance rate of 0.27052 and a total runtime of 1,652 s.



(a) 4 uniform sensors

(b) 4 random sensors

Figure 9. Two-parameter MCMC updates.



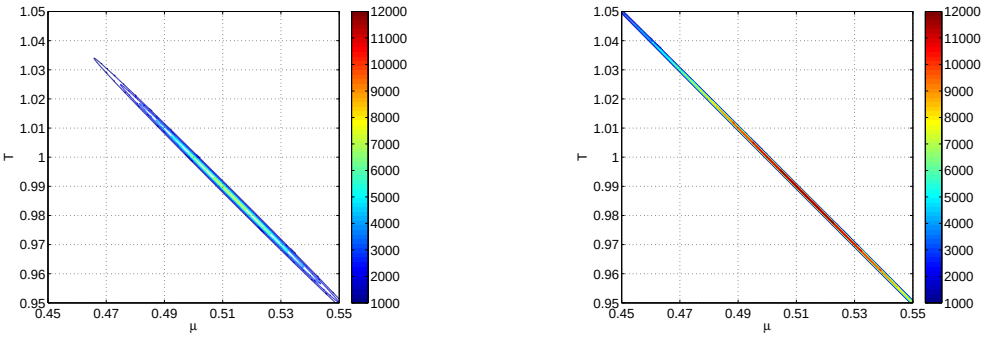
(a) 4 uniform sensors

(b) 4 random sensors

Figure 10. Two-parameter MCMC histograms.

After ensuring the Metropolis-Hastings algorithm would work for our problem, we wished to perform a complete inversion, considering all members of $\mathbf{b}_t$ to be unknown. However, we preface this last inversion with two more brute-force posteriors. Figure 11 shows posteriors resulting from both uniform and random placement of four sensors, where we've taken μ and T to be unknown. (Now, we've conditioned on $A = 5$ units of concentration and $\sigma = 0.1$ m.) As one would expect, there is a strong relationship between the initial location of the contaminants and the time-since-release. Diffusion prevents this from being overly problematic, but having not committed an inverse-crime has led to similarly shifted posteriors for both sensor arrangements. As the parameter grids used here were bigger than used before, we saw wall times of around 8,190 s while performing each inversion.

Now, we present the complete MCMC inversions, where we've allowed all members of $\mathbf{b}_t$ to change throughout. Again, we try to recover the true initial conditions, which are $\mu_t = 0.5$ m, $T_t = 1$ s, $\sigma_t = 0.1$ m, and $A_t = 5$ units of concentration. Figure 12 on the following page shows the resulting MCMC walks for both uniform and random sensor placements, which do not seem to converge to any recognizable posterior, even within 50,000 samples. This can be explained by the strong correlation between μ and T (see Figure 11), which causes the marginal posteriors to widen significantly. Even though we scaled the proposal distribution such that both inversions achieved acceptance rates between 20% and 30% – the optimal range – we could not converge quickly enough to guarantee usable results. In order for the random walks to accurately depict the

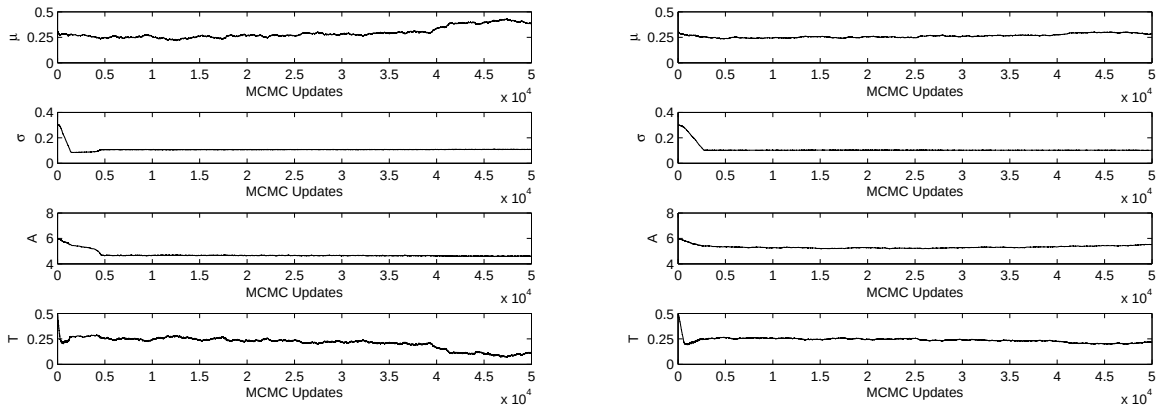


(a) 4 uniform sensors

(b) 4 random sensors

Figure 11. Posterior PDFs, T vs. μ.

posterior, we could, for example, dramatically increase the number of samples, but this would result in heavily increased computation times. Future work could also attempt to relieve this by sampling the posterior via an anisotropic proposal distribution, rather than the uniform one used here. Indeed, the updates in Figure 12 exhibit trend-like low-frequency behavior, meaning that the proposal scaling was too small to efficiently explore the posterior. With an anisotropic proposal, one could retain an optimal acceptance rate with a much larger proposal window.



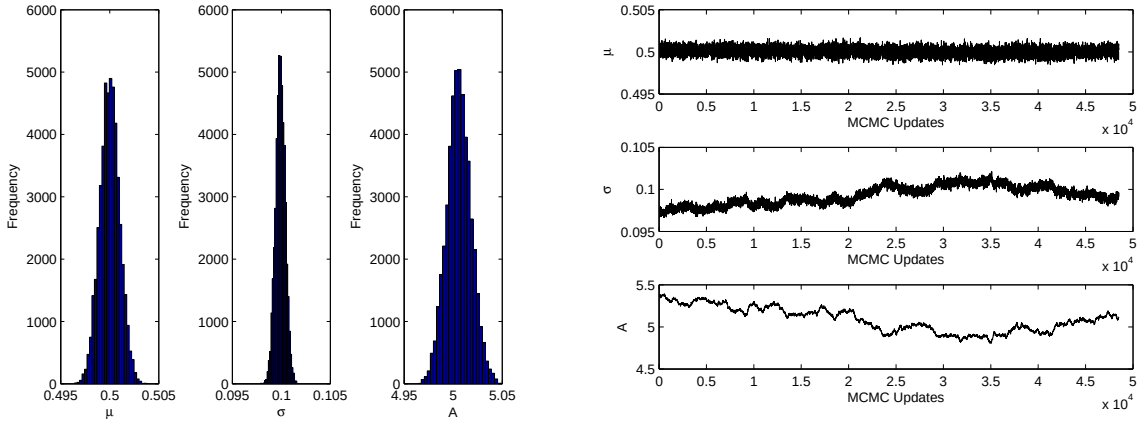
(a) 4 uniform sensors

(b) 4 random sensors

Figure 12. Four-parameter MCMC updates.

As it seems four-parameter MCMC inversions which treat time-since-release as unknown do not currently give efficient results, we investigated the feasibility of three-parameter inversions. For these final tests, we conditioned the posterior on time-since-release, taking $T = 1$ s throughout the random walk. Again, a successful inversion should present marginal distributions with mean values identical (or very close, as no inverse-crime was committed) to the true initial conditions. Results of inverting with both uniform and random sensor arrangements are shown in Figure 13 on the next page. For the uniform sensor setup, the MCMC updates were almost identical in appearance to those shown in Figure 9, so only the truncated histograms are included here. The random sensor setup, however, resulted in a walk that, as in the four-parameter inversion, did not explore enough of the posterior within 50,000 samples. As before, the updates in Figure 13(b), although now somewhat truncated for burn-in, show trend-like low-frequency behavior: an anisotropic proposal seems to be necessary in this case as well. The histograms in Figure 13(a) are encouraging, however, and the marginal distributions have means of $\bar{\mu} = 0.5$ m, $\bar{\sigma} = 0.0999$ m, and $\bar{A} = 5.006$ units of concentration. Furthermore, this random walk had an acceptance ratio of 0.2538 and took only 1,631 s to complete, making three-parameter inversion times comparable to the two-parameter times seen

before. It appears that when the "right" proposal distribution is selected for a given posterior, MCMC inversions can give accurate and efficient results over multiple unknowns.



(a) 4 uniform sensors

(b) 4 random sensors

Figure 13. Three-parameter MCMC inversions.

IV. Conclusion

A probabilistic approach for solving inverse convection-diffusion in one-dimension was presented, and a Monte Carlo Markov Chain algorithm for efficiently performing inversions over multiple unknowns was discussed. In order to facilitate inversions, a one-dimensional contaminant transport forward solver was implemented. The arrangement of sensors within the domain of interest was found to have a significant effect on the size and shape of the posterior PDFs, and arrangements favoring one side of the domain were shown to provide unusable results. Upon performing mesh tests to eliminate the effects of any inverse-crimes, four sensors in a uniform or random setup were found to provide posteriors which exhibited adequate resilience to changes in mesh size. The MCMC approach was then validated for two unknowns; computation times were found to be similar to those of the previously used brute-force method. MCMC inversions over three and four unknowns were also performed, yielding usable results only over reasonably isotropic posteriors. As optimal acceptance rates were always achieved, we concluded that anisotropic proposal distributions could help explore similarly anisotropic posteriors distributions. Multiple-parameter MCMC inversions showed promising computation times, and we believe the MCMC approach is a clear win over the brute-force method.

Future work will include introducing anisotropic proposal distributions to MCMC inversions, in hopes that time-since-release can be treated as an unknown. Other methods of increasing efficiency during inversions will be explored, and may include replacing our primal forward solver with an adjoint-based solver. This and other model reduction techniques will become essential when performing inversions in two or three dimensions. Once we reach higher dimensions, we believe that probabilistic methods such as those presented in this paper will become invaluable to providing efficient and usable results.

References

[1]Akcelilk, V., Biros, G., Draganescu, A., Ghattas, O., Hill, J., and van Bloemen Waanders, B., "Dynamic Data-Driven Inversion for Terascale Simulations: Real-time Identification of Airborne Contaminants," *2005 ACM/IEEE Conference on Supercomputing*, 2005.

[2]Bashir, O., *Hessian-Based Model Reduction with Applications to Initial-Condition Inverse Problems*, Master's thesis, Massachusetts Institute of Technology, 2007.

[3]Galbally, D., Fidkowski, K., Willcox, K., and Ghattas, O., "Nonlinear Model Reduction for Uncertainty Quantification in Large-Scale Inverse Problems," *International Journal for Numerical Methods in Engineering*, 2000.

[4]Calvetti, D. and Somersalo, E., *Introduction to Bayesian Scientific Computing: Ten Lectures on Subjective Computing*, chap. 9, Springer, 2007.