Data Assimilation for Fluid Dynamic Models

Amy Thompson

01 August 2013

Abstract

Data assimilation is the merging of observational data into a mathematical model. Thus, data assimilation is particularly important in the field of fluid dynamics and, given data taken from the trajectory of an instrument flowing in an unknown velocity field, can be used to recreate the velocity fields inducing the motion. This study examines the efficacy of a strategy which constructs a posterior distribution to represent the underlying velocity field. The strategy, based on Bayes Theorem, allows for the construction of the posterior distribution from the merging of the data, taken from a known function, and a mathematical model based on Fourier Series, with the random sampling of the coefficients according to Metropolis-Hastings within Markov chain Monte Carlo simulation. The results from the study, for 1-dimensional sampling in Eulerian assimilation, produce a posterior distribution that appropriately induces the observed data and therefore motivate the 2 dimensional sampling, in Eulerian assimilation, as well as further study with Lagrangian assimilation using lab data.

Contents

List of Figures

1 Introduction

The motion of an object through a fluid is related to the flow of the fluid in which it rests. This is a question in fluid dynamics that has been answered in the past using differential equations. To model this flow, or velocity field, using differential equations, there is a need for functions that express the motion of the fluid. These functions are often unknown or difficult to work with. Data assimilation provides an alternate approach to modeling an unknown velocity field using observational data and merging it with an emergent mathematical model. The goal of this research is to recreate the velocity field inducing the motion of observational data taken from the trajectory of an instrument flowing in an unknown velocity field. The strategy of this research is to merge observational data with its Fourier Series representation using Metropolis-Hastings random sampling of the Fourier Series coefficients, within a Markov chain Monte Carlo simulation.

2 Background and related work

Data assimilation is a process that merges observational data with a mathematical model. The result is an objective estimate of the state, which can be propagated through the model to obtain a prediction [2]. There have been many forms of data assimilation but the one used for this research is based on the Kalman filter. The Kalman filter was developed by Rudolf Emil Kalman and is where most data assimilation techniques come from. Using the Kalman filter for a linear system with Gaussian observation error, provides an exact mean and covariance of Gaussian distributions in these states, which is the prior distribution and posterior distribution respectively [1]. The use of the Kalman filter, or derivatives of the Kalman filter have been used in data assimilation to address fluid dynamic models; one such example is the Ph.D. thesis by Damon McDougall. This research is based off the work done by McDougall [2].

2.1 Bayes Theorem

Bayes Theorem gives the posterior distribution-the probability of an unknown observation conditional on the observed data-as proportionate to the prior distribution times the likelihood of the data. Remembering from probability theory, that $0 \leq p(x) \leq 1$ and $\int_{-\infty}^{\infty} p(x)dx = 1$. Given a velocity field v and data θ,

$$p(v|\theta) \propto p(v)p(\theta|v).$$

2.2 Types of data assimilation

As there are many different forms that data can be collected in, and also many different methods to model data, data assimilation has various types, many of which will not be discussed in this paper. The types of data assimilation used in this research are smoothing, Eulerian, and Lagrangian data assimilation.

2.2.1 Smoothing

Smoothing data assimilation is the process of estimating the state using all possible data [2]. For a given model and given observations, smoothing data assimilation makes a prediction about the model based upon all of the observations.

2.2.2 Eulerian data assimilation

There are two types of data assimilation that we will look at: Eulerian and Lagrangian. Eulerian data assimilation is when the path of an object is traced across some fluid. We observe a velocity field with noise where the observations are fixed. We assume a steady flow and so velocity does not change with time. For a velocity field v, of a fluid at fixed points in space and time t,

$$y = v(x,t) + n$$

where n is from an i.i.d. standard normal distribution [2]

2.2.3 Lagrangian data assimilation

Lagrangian data assimilation is when the object moves with the fluid,

$$y = z(t) + n$$

where n is from an i.i.d. standard normal distribution and z is the position of the object and the derivative of z equals $v(z,t)$ [2].

These two types of data assimilation have techniques similar to each other. The distinction being that Eulerian data assimilation focuses on the object that does not move with time, while Lagrangian data assimilation focuses on the flow on an object with time. We begin our strategy working with Eulerian data assimilation.

2.3 Fourier Series

Fourier Series is a method of approximating a function very similar to Taylor Series. Fourier Series uses a summation of sines and cosines as an orthogonal basis over the given function space. The Fourier Series representation of a function f is:

$$\text{f(x)} = \tfrac{1}{2}a_0 + \sum_{n=1}^{\infty} a_n \cos(nx) + \sum_{n=1}^{\infty} b_n \sin(nx)$$

where

$$a_0 = \tfrac{1}{2\pi}\int_{-\pi}^{\pi} f(x)dx$$

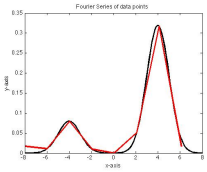
$$a_n = \tfrac{1}{\pi}\int_{-\pi}^{\pi} f(x)cos(nx)dx$$

$$b_n = \tfrac{1}{\pi}\int_{-\pi}^{\pi} f(x)sin(nx)dx \text{ [3]}$$

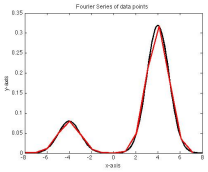
An example of approximating a function with a Fourier Series can be found in Figure 1; graph(a) shows the function as a black line and the red line as a loose fit using 8 data points; graph (b) shows the same function being approximated by a Fourier Series using 16 data points; graph (c) shows the same function being approximated by a Fourier Series using 32 data points. As the data points double, the approximation better fits the original function.

2.4 Markov chain Monte Carlo Simulation

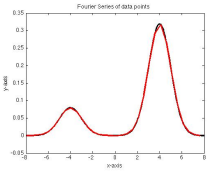
Markov chain Monte Carlo (MCMC) simulations provide a way to draw samples from an unknown target distribution. Using MCMC simulations when the target distribution is a posterior distribution, these simulations can be used for data assimilation [2]. We now provide a brief description of the components of MCMC used for this research: Monte Carlo methods, Markov chains, and Metropolis-Hastings sampling.



(a) Fourier Approximation with 8 data points



(b) Fourier Approximation with 16 data points



(c) Fourier Approximation with 32 data points

Figure 1: Fourier Series Approximation of a Function

2.4.1 Monte Carlo Methods

Monte Carlo Methods are a way to approximate an integral. Given the function *f*, integral I is:

$$I = \int_0^1 f(x)dx$$

Using Monte Carlo methods, the integral can be approximated by:

$$I \approx \int_0^1 \frac{f(x)}{g(x)} g(x)dx$$

Where $g(x)$ is a density on $[0, 1]$ such that if $f(x) \neq 0$ then $g(x) > 0$ [4]. From statistics we know that

$$I = E_g(\tfrac{f(X)}{g(X)})$$

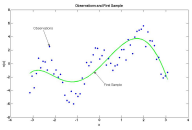
Where E_g is the expections with respect to the distribution of g. We then produce an i.i.d. sample from the distribution for g and set

$$\hat{I}_k = \tfrac{1}{k} \sum_{i=1}^{k} \frac{f(x^i)}{g(x^i)}$$

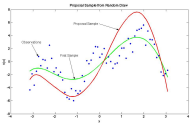
The law of large numbers tells us that $\hat{I}_k$ converges to I with probability 1 as k approaches infinity [4].

2.4.2 Markov chains

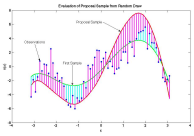
A Markov chain is a collection of random variables $X = \{X_k, k = 0, 1, 2, ...\}$ where the next step in the chain is independent of any steps in the chain prior to the current one. A random variable can be generated by a random number generator. There are many random number generators available to be used in computations, in all actuality, these are really pseudo-random number generators as true randomness is impossible to attain. Many of these pseudo-random number generators use algorithms based on prime numbers to give the impression of "randomness." For this research, Matlab's randn and rand funtions are used. The rand function provides a random number between 0 and 1 based on the uniform distribution. The randn function provides a random number based on the standard normal distribution.



(a) Initial sample from polyfit of observations



(b) Sample from prior draw



(c) Evaluation of sample for step K+1

Figure 2: Metropolis-Hastings sampling with MCMC

For the purposes of this research, we assume that the Markov chain is time homogeneous. This allows for greater ease when dealing with the Markov chain. We also assume our Markov chain is ergodic, meaning that for a discrete time Markov chain on a discrete state space, the Markov chain is irreducible, aperiodic, and positive recurrent. An ergodic Markov chain has a limit distribution and guarantees a full sampling of the state space [4].

Another feature of working with Markov chains is the burn-in. For any Markov chain, there is an initial period of steps where the values are far from the truth. These initial steps are discarded before evaluation of the Markov chain. The size of the burn-in period is left to the discretion of the user; generally determined by trial and error. We chose to discard a burn-in period of the first 20% of the overall sample size.

2.4.3 Metropolis-Hastings Sampling

Within the MCMC simulations there are multiple ways to sample. We use the Metropolis-Hastings sampling method. This method performs as follows: Let $f(x)$ be the target density. Let $q(x), x \in S$ be the proposal distribution. Suppose $X_n = x \in S$.

$$\text{Sample } Z = z \text{ from } q(z), z \in S.$$
$$\text{Accept } Z = z \text{ with probability:}$$

$$\alpha(x, z) = min\{1, \tfrac{f(z)}{f(x)}\}$$

If $Z = z$ is accepted, set $X_{n+1} = z$. Otherwise, set $X_{n+1} = x$ [4]. An example of Metropolis-Hastings within MCMC can be found in Figure 2; graph (a) shows the observational data with the initial sample taken from a low order polynomial fit from the data points; graph (b) shows the proposal sample as it relates to the initial sample and the observations data; graph (c) shows the evaluation of the proposal sample versus the initial sample using the sum of linear least squares.

3 Preliminary

For this study, we solve for a velocity field as the posterior distribution. We solve for the posterior distribution using Bayes Theorem, with Fourier Series taking the role of the prior

and Metropolis-Hastings random sampling of the Fourier Series coefficients as the likelihood. Our resulting method is as follows:

$$p(velocityfield|data) \propto p(FourierSeries)p(data|FourierSeries)$$

To set up our simulation, we use the following map:

$$y = G(x) + n$$

Where G is our model, x is the state, y is the observation, and n is the noise. We found the likelihood, Φ using $\Phi = \frac{1}{2}||G(\hat{x}) - y||^2$ [2]. The 2-norm was used for this likelihood.

We construct a draw from the prior using the Fourier Transform. The random draw is of the Fourier coefficients. We construct the vector of coefficients in the frequency domain with the following guidelines:

1. The vector is of an odd size in length, accounting for a_0 all a_n and all b_n
2. The value for $a_0 = 0$
3. The number of random values of a_n are equal to the number of b_n
4. The values in b_n are the complex conjugates of the random values in a_n
5. The vector is constructed in accordance with Matlab's layout for the fft vector
6. The number of random values for each a_n chosen represent the dimension of the Fourier Series
7. The resulting functions after the ifft function is called must be real-valued functions (no imaginary parts)

We normalize the draws from the prior, ξ, according to the following algorithm:

$$z = (1 - \beta^2)^{\frac{1}{2}} u[i] + \beta\xi$$

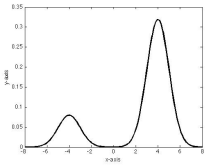
Where β is set to 1 and $u[i]$ is the current sample [2].

4 Methods

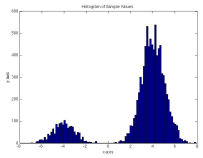
The methods used for this research were to begin with a 1D point sampling, move on to a 1D Eulerian function sampling within MCMC. The following sections summarize the work done to date for this research.

4.1 1D Point Sampling

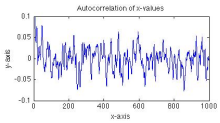
Suppose we have a simple target function, f(x), consisting of sines and cosines. Generate a random x-value within the support of the function and establish it as the first sample. Generate a second random x-value, $\hat{x}$, according to some proposal distribution that has the same support as the target distribution, both Gaussian and Uniform distributions would be appropriate proposal distributions. Evaluate $\hat{x}$ using the Metropolis-Hastings sampling method. Generate a distribution of $\hat{x}$-values and plot a histogram and compare to the target function (Figure 3). The histogram should fit the curve of the target function. Next, evaluate the correlation of the $\hat{x}$-values to check that the entire function space is being sampled.



(a) Target function



(b) Histogram of Samples



(c) Autocorrelation

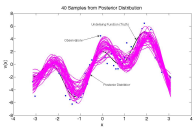
Figure 3: 1D Point Sampling

4.2 1D Eulerain Function Sampling within MCMC

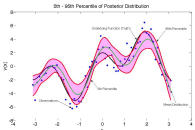
Suppose we have a simple target function, u, consisting of sines and cosines. Generate an evenly distributed subset of data points from the target function. The use of multiples of 2 for number of original points in the function as well as for the number of points in the subset allows for better functionality of the Matlab Fast Fourier Transform fft and its inverse ifft. An initial sample function, v, is created from a low order polynomial fit to the subset of data points and sampling from the frequency domain for the Fourier Series coefficients uses Metroplis-Hastings within MCMC simulation. The graphs in Figure 4 provide an example of the research work; graph (a) shows 40 samples taken from the posterior distribution generated from given observations. The distribution approximates both the underlying function and the given observations within an acceptable range; graph (b) shows the 5th through the 95th percentile of the posterior distribution and how well it approximates the truth and given observations, as well as showing the mean of the posterior distribution; graph (c) shows what happens when observations are given for only a portion of the support domain, the posterior distribution only approximates the truth and given observations well within the portion of the domain where the likelihood can be evaluated. The following code gives a overview of the code used for this simulation.

Pseudocode:

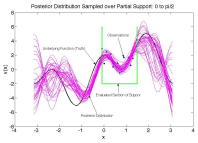
1. Choose target function
2. Select subset of x and y values and set as exact
3. Fit polynomial to subset of values
4. Generate observations from randn added to exact
5. Set initial sample to polynomial fit
6. Get next sample from RandomDraw
7. Normalize next sample as shown above
8. Add next sample to initial according to Metropolis-Hastings algorithm
9. Evaluate likelihood of initial and likelihood of next
10. Accept next sample according to Metropolis-Hastings algorithm



(a) 40 Samples from Posterior Distribution



(b) 5-95 Percent of Posterior Distribution with Mean



(c) Evaluation of Partial Support for Posterior Distribution

Figure 4: 1D Eulerian Function Sampling

5 Conclusion and Future Work

The research on 1D point sampling and 1D Eulerian function sampling is consistent with expectations from the methods used. The Metropolis-Hastings random sampling of the Fourier coefficients within the MCMC simulation provides a posterior distribution that approximates the given observational data. The current work for this research involves 2D Eulerian function sampling as well as a shift to Lagrangian data assimilation. Future work will be to gather observational data from tank in Dr. Ani Hsieh's lab.

The ability to recreate a velocity field from given observational data is important in the area of fluid dynamics. Using data assimilation to find this velocity field could provide a more cost effective means of creating models of fluid dynamics which is crucial to industries related to transportation services and weather prediction.

References

[1] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[2] D. McDougall. *Assimilating Eulerian and Lagrangian Data to Quantify Flow Uncertainty in Testbed Oceanography Models.* PhD thesis, University of Warwick, 2012.

[3] D.L. Powers. *Boundary Value Problems.* Prentice-Hall, Inc., Florida, USA, 1987.

[4] E Thönnes. Monte carlo methods. 2012.