River Flow Modeling with Stochastic Jump Diffusion

Scott Hanna – shanna7@jh.edu

Final Paper – 625.714: Stochastic Differential Equations– Johns Hopkins University

# Abstract

A stochastic model of river discharge is utilized for the purposes of parameter estimation and forecasting of the Potomac River flow rates. Flow rates, also called discharge, are modeled by a jump diffusion stochastic differential equation where jumps represent precipitation events and are assumed to be independent in time. Flow relaxation is accounted for as well as Brownian turbulence in the flow rate. In addition, a latent variable for rest flow rate is utilized to account for water saturation. A Bayesian particle filter approach with importance sampling of the posteriors is taken to update the model with discrete observations of USGS discharge data.

# Introduction

In 2016 the town of Ellicott city, MD experienced a devastating flood after heavy rains caused the streams flowing into the Potomac River to overflow. The flood was called a 1000-year flood because it was expected that the river would only flood to that extent, on average, once every 1000 years. That is, on any given year there is a 0.1% chance of such a flood occurring. However, only two years later, in 2018, a second flood of equivalent magnitude rushed through the shops and households of the town which were still rebuilding from the 2016 flood.

The question arises whether these two floods occurred by chance, unlikely as it was, or if conditions have changed to increase the likelihood of these floods occurring. Modeling river flow, or discharge (measured in ), could help us develop predictive abilities to forewarn of possible flood events. Fortunately, the United States Geological Survey (USGS) tracks discharge at hundreds of locations across the country. These gauges which take discharge measurements every 5-30 seconds provide a rich time series of observations.

In this paper, a stochastic modeling technique is utilized to use discharge measurements of the Potomac River provided by the USGS to help forecast high flow events. Additionally, the model could help learn about river conditions that affect discharge rate that might be used to determine flood likelihood. This information could help inform municipalities when the right time might be to construct flood mitigation measures. Such models could also help predict when high sediment and debris deposits might occur downstream (Packman, 2013).

River discharge is dependent on random fluctuations due to turbulent fluid dynamics that can be modeled as a Gaussian process (Lawrance, 1977). However, it is highly dependent on non-Gaussian precipitation events that occur randomly in time and in magnitude. To capture these random events in multi-year river flow time series observations, one approach has been Poisson jump models (Lefebvre, 2015).

There is a rich precedent in financial literature of the use of jump diffusion models that account for both Brownian motion and discrete Poisson jumps (Johannes,2009; Privault, 2021; Ntemi, 2021). These models are used for price forecasting, but we can use the methods, with some modification, for river flow forecasting. In the next section we outline the details of the jump diffusion model of river discharge and describe how it differs from price forecasting.

# Data and Stochastic Model

Data was collected by USGS stream gauge sensors. I used daily means of discharge in cubic feet per second over the period from 1/1/2015 to present. A graph of the collected data is shown in figure 1.

Chart, histogram

Description automatically generated

Figure : Daily mean discharge from 2015 to Present.

It can be observed that discharge is highly dependent on precipitation events that occur randomly in time and magnitude. Each precipitation event will increase the flow rate by some magnitude, but the flow levels quickly settle to some rest level. It also appears that the rest level, that is, the level at which the flow rate recedes to, changes over time as well. During times when many large precipitation events occur the rest level rises. I initially sought to model the rise and fall of the rest level as a periodic or seasonal adjustment. However, it does not appear this occurs at set intervals and modeling this data to a seasonal model using dynamic harmonic regression was unsuccessful.

A model I found that works better is a stochastic jump diffusion model with a latent factor for rest level that I call saturation. The stochastic differential equations for this model are given below.

The random variable is the flow rate or discharge in cubic feet per second, is represents saturation, which models the resting level of the water, and is Brownian motion which models turbulent flow. is a poisson distributed number of precipitation events in each time span with rate , and is an exponentially distributed precipitation with factor which represents the magnitude of a given precipitation event.

Jump diffusion models in the literature (e.g., Johannes,2009) model the jump magnitudes with a gaussian term which is more apt for financial modeling. However, this wouldn’t make sense for river flows at the time scales we are observing. We only consider precipitation events that add water and therefore increase flow rate. Looking at a histogram of the discharge data in figure 2 gives a sense for the distribution of precipitation events and why I chose to model them as exponential.

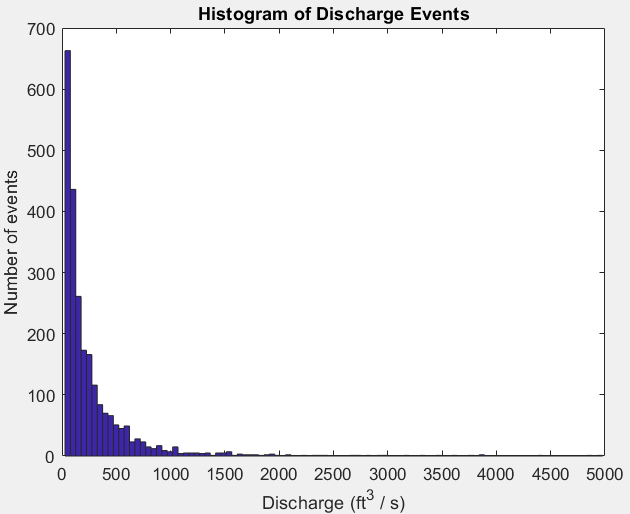


Figure : Histogram of discharge events

The value in the model represents the flow decay rate and models the rate at which high volumes of water will after precipitation events will discharge out of the waterway. The value is the turbulence factor, is saturation decay rate and is saturation percentage, or percentage of flow that will add to the saturation of the water way. The more saturated the waterway, the higher the rest level of flow rate.

I applied an Euler discretization algorithm to numerically simulate the model for flow rate. Applying the solution with no saturation (that is for all ) does not result in a rising and falling level of water flow as we see in the observations. Figure 3 shows an example solution with no saturation. In contrast, adding saturation to the model results that more closely match our observations as shown in Figure 4.

Chart, histogram

Description automatically generated

Figure : Modeled Discharge with no saturation.

Chart, histogram

Description automatically generated

Figure : Discharge (blue) with saturation. The saturation level is shown as the dotted red curve.

Our goal is to solve the filtering problem for this model using the USGS discharge measurements of the Potomac River. By filtering the data, we can hopefully perform parameter estimation of river saturation rate, precipitation rate and magnitude. This may allow us to profile different river systems to assess flood vulnerability.

# Filtering

Filtering will allow us to use the stochastic SDE model, along with collected discharge observations, to give forecasts and parameters estimations. Two recent papers on solving the filtering problem for jump diffusion SDEs with latent variables are (Johannes,2009; Privault, 2021; Ntemi, 2021). They take a Bayesian approach, where new state parameters, , and the observations, , are related by the Bayes rule:

Where is the predicted flow rate. Thus represents the posterior distribution. Unfortunately, given the complicated nature of the SDE which contains a continuous diffusion term and discrete jumps, the posterior distributions cannot be solve analytically. Additionally, Kalman filter techniques are not adequate as they use a linear Gaussian assumption that isn’t well suited for Poisson jumps.

The approach used to compute the posterior distributions is the particle filter algorithm. This algorithm approximates the distribution by producing several thousand particles that diffuse through the potential outcomes of the applied propagator function. It is important that enough particles are produced to properly distribute themselves such that they appropriately approximate the posterior distribution.

Particle filtering was carried out with the Control System Toolbox for MATLAB. This library contains a particle filter object which can be instantiated with two inputs: a link to a user-defined state function propagator, and a user defined likelihood function. The particle filter process is shown in the flow charge diagram in figure 5.

Diagram, schematic

Description automatically generated

Figure 5: Matlab control toolbox implementation of the Particle Filter. u and w represent the control and noise inputs respectively, y is the measurements and x-hat represents the estimated state of the system.

In the case of our jump diffusion model, we can define a multivariate distribution that contains both the time evolution of and the latent saturation . Additionally, for the purposes of parameter estimation, we can include small particle diffusions in all parameters (). Table 1 provides a summary of the input for the state function propagator.

|  |  |  |
| --- | --- | --- |
| **Variable Description** | **Symbol** | **Representation** |
| *Discharge* |  |  |
| *Saturation* |  |  |
| *Discharge Relaxation Rate* |  |  |
| *Gaussian Turbulence Modifier* |  |  |
| *Saturation Percentage* |  |  |
| *Saturation Relaxation rate* |  |  |
| *Precipitation Frequency* |  |  |
| *Precipitation Magnitude* |  |  |

**Table 1**: Variables Descriptions for the propagation function used in the particle filter algorithm.

The full representation of the state propagator is therefore:

Where will be drawn from a distribution. Additionally, the parameters will have an additional gaussian diffusion term added every iteration. The purpose of this is to help the particles diffuse out to approximate the posterior distribution. Finally, the likelihood function was implemented assuming a gaussian normal measurement error with a mean of zero and a co-variance matrix of the identity matrix.

# Results

The number of particles was chosen to be 5000 as this provided precise enough distribution given the variability in the large jumps. Initial values were chosen based on the Euler simulations as follows: . The particle filter was set to resample particles every iteration using a multinomial resampling policy.

Figure 6 shows output for the first 500 days where red circle indicates filtered results, and the blue curve are actual measured values. The data was well modeled by the particle filtering algorithm. Figure 7 displays the residuals between the model and measurements for the entire time-period of observations. It appears the greatest errors occur in jumps. Meaning, either the model predicted a jump that did not occur (high positive error) or did not predict a jump that did occur (high negative error). Because jumps are large flow events, getting them wrong has a big impact on the filtering accuracy.

Chart, histogram

Description automatically generated

Figure 6: Filtered results (red circles) overlaid on top of the discharge data (blue). This plot shows the first 500 days of results.

Chart, box and whisker chart

Description automatically generated

Figure 7: Error in filtered results vs. Discharge measurements. Error primarily occurred due to missed jumps.

Table 2 shows the values of the resultant parameters. Time series graphs of two parameters of interest, saturation percentage (), and precipitation occurrence rate ( are shown in figure 7. I believed the saturation percentage (figure 8-left) would be much smaller, but in all simulations the value came out to be a significant portion of the overall flow rate. I was also surprised to see it change over time since I would have suspected this to be a static feature of a waterway. It makes more sense that the precipitation occurrence would be time variable as is shown in figure 8-right.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 1.617 | 0.770 | 0.781 | 0.251 | 0.788 | 222.2 |

Table 2: Resultant estimated parameters

|  |  |
| --- | --- |
| Graphical user interface, chart, line chart, scatter chart  Description automatically generated | Chart, line chart, scatter chart  Description automatically generated |
| Figure 8: Plots of saturation percentage ( [left] and precipitation occurrence rate () [right] are shown. | |

It took upward of 10 minutes to run the particle filtering algorithm with 5000 particles over the entire time span of 2413 days. This lengthy time makes sense given that at each iteration 5000 possible configurations are being sampled for this complicated equation. The full function is 8 dimensional when the parameters are considered. Therefore, many particles are required to fill the event space and give an accurate approximation of the posterior distribution. Running with fewer particles often resulted in divergence in the algorithm.

# Discussion

**Results discussion**

The jump diffusion saturation model with particle filtering was successfully able to model discharge measurements of the Potomac River. However, parameter results did not converge in a consistent manner. This could be indicative of two main things. First, the parameters representing river saturation or precipitation rate could change over time. For example, precipitation rate likely has seasonal variations. Second, the parameters may include interdependencies which were not accounted for in this model. Saturation, for example, may depend on the precipitation event rate and magnitude.

The non-convergence of the estimated parameters limits conclusions that can be drawn from this model. For example, the two dates of flooding in Ellicott city, just downstream of the location of discharge gauge measurements, were on July 30, 2016 (day 577) and May 27, 2018 (day 1243). Although the overall discharge was high on both dates, there was no consistent pattern in the parameters. For example, neither saturation rate, nor precipitation rate or magnitude parameters were significantly high or low during these dates. Additionally, the variable nature of the parameter estimates limits the accuracy of forecasting.

The conclusion I can reach at this point is that a lot more work needs to be done to get consistent parameter estimates and accurate forecasting. The large jumps in discharge values that result from precipitation events make consistent trends in data difficult to resolve. In the next section I will cover some methods that can be used to improve the model.

**Further Research**

One difficulty related to the utilization of particle filtering to model jump diffusion with high magnitude jumps was related to algorithm divergence. Because large jumps can occur with varying regularity the posterior probability distribution swept out a very large possibly space. If, in the next iteration, there was a jump but only a small subset of the 5000 particles saw a jump, there won’t be enough to accurately approximate the magnitude of the jump. One solution to this is a method called particle injection (Kochenderfer,2021). This method adds particles from a broader distribution into each algorithm. This process can be adaptive by injecting more particles when the mean weight of the current iteration’s particles is low.

Another improvement we can make is to take into consideration seasonal effects on precipitation events. In fact, we could also pull in precipitation data as an additional dimension to the measurement space to reduce the variable parameters related to precipitation. Smoothing data may assist in determining longer term season trends.

If the non-convergence of the parameters can be solved, the particle filter method, as implemented in the control toolbox for MATLAB, allows for a control parameter . It may be useful to consider water flow controls from upstream reservoirs to limit flood potential in the downstream areas.

Finally, one important question of municipalities in flood zones is: when is the right time to construct a flood mitigation project such as a levy? Given a model of water flow we may be able to pose this as a stopping time problem. If an appropriate cost function can be produced for floods of a given magnitude, then we could analyze when the best time would be to no longer follow the status quo.

# References

Richard F. Bass. Stochastic differential equations with jumps. *Probability Surveys*, 1(none):1 – 19, 2004.

Michael S. Johannes, Nicholas G. Poison, and Jonathan R. Stroud. Optimal filtering of jump diffusions: Extracting latent states from asset prices. *Review of Financial Studies*, 22(7):2759 – 2799, 2009.

Mykel J. Kochenderfer, Tim A. Wheeler, and Kyle H. Wray. *Algorithms For Decision Making*. MIT Press. 2021.

A. J. Lawrance and N. T. Kottegoda. Stochastic modelling of river flow time series. *Journal of the Royal Statistical Society, Series A*, 140:1 – 31, 1977.

Mario Lefebvre and Fatima Bensalma. Modeling and forecasting river flows by means of filtered Poisson processes. *Applied Mathematical Modelling*, 39(1):230–243, 2015.

Binliang Lin and Roger A. Falconer. Integrated 1-d and 2-d models for flow and water quality modelling. *Proceedings of ... Congress of International Association for Hydraulic Research*, 31:543 – 552, 2005.

Myrsini Ntemi and Constantine Kotropoulos. A jump-diffusion particle filter for price prediction. *Signal Processing*, 183, 2021.

Aaron Packman, Jennifer Drummond, and Antoine Aubeneau. Stochastic modeling of fine particle deposition, resuspension, and hyporheic exchange in rivers. *Geophysical Research Abstracts*, 15:@Abstract EGU2013 – 6548, 2013.

Douglas Pender, Sandhya Patidar, Gareth Pender, and Heather Haynes. Stochastic simulation of daily stream flow sequences using a hidden Markov model. *Hydrology Research*, 47(1):75 – 88, 2016.

Nicolas Privault. Lecture notes - notes on statistical finance. *Nanyang Technological University, School of Physical and Mathematical Sciences*, pages 689 – 783, 2021.

Code for the project can be found at <https://github.com/sphanna/DischargeJumpDiffusion>