Application of particle filtering in river flow modelling

Data Assimilation for Geosciences AESM5120C

David Haasnoot 4897900

A graph of a graph

Description automatically generated with medium confidence



Introduction

Different processes in geosciences can be modelled in order to obtain a better understanding of process and to make predictions. Often times these can be chaotic and difficult to model correctly. For this research project in the course Data Assimilation for Geosciences (AESM5120C) the flow of water in a river catchment is modelled. This is due to its importance in understanding the behaviour of water through a natural water catchment and the use in predicting the impact it can have on the surroundings. The model used is the ‘*Hydrologiska Byråns Vattenbalansavdelning*’ (HBV) model developed originally in Scandinavia but is applied word wide. It is also known as a bucket model as it models the different processes in a catchment as different buckets. It is schematised below in figure 1, the parameters are explained further on in depth.

A diagram of a flowchart

Description automatically generatedA group of blue graphs

Description automatically generated

*Figure 1: Schematisation of the HBV model. Figure 2: Results of brute force method*

Model background:

This model works well conceptually, however the processes at each stage are controlled by parameters which are unknown and can’t directly be measured. The infiltration capacity of section of soil can be measured, but the catchment as whole is not possible. The common, easier, approach is to do a Montecarlo simulation of the bulk process and from this choose a set of parameter values. In this brute force method, every parameter value is chosen randomly per iteration and then the model is evaluated. This can be very expensive to obtain the ‘best’ parameter set. This results in the image shown above on the right in figure 2. In some cases there is a clear pattern such as for and . Whilst in other cases there is little trend to be observed for which is the most optimal parameter.

Research scope:

Data assimilation can evaluate the best parameter at each time step and chose the best to continue with. This could be used to approximate the parameters better than using a bulk method described previously. The bulk method is fairly time intensive as many runs are needed to obtain meaningful parameters. As a comparison, figure 2 required 5000 runs. Even with this amount of runs it doesn’t guarantee that the optimal parameter set has been found. By using a particle filter every time step parameter inference can be used to better approximate the parameters, adjusting them to best fit with new data. In reality P, E and Q are known fairly quickly, thus this method could also be used to quickly model water systems. The question this research aims to answer is “*Can sequential particle filtering improve the calibration of HBV models in hydrology & thus predict river flows better?”*.

States & parameters:

The model takes as input precipitation (P) and potential evaporation (Ep) (or temperature as Ep is a function of temperature in most cases). The output is the river discharge Q. Observations of Q are needed in order to calibrate the model, which can thus also be used for data assimilation. Output of the model is the modelled discharge, which is dependent on the parameters . These all control flows or storage volumes of the different ‘buckets’ schematised in the model above. is the main state value of the state vector as this is compared to the observations. The different storages at each time interval should also be considered in the state vector as the next step depends on the storage in the previous step. Thus are also to be considered as elements of the state vector of the system. In hydrology some processes can take longer than the model timesteps. To deal with this a lag term is introduced. represents the time between water falling and it reaching the river. It’s modelled as a Weibull distribution where the flow is distributed over a the days. To still keep the Markov-Chain model structured required for data assimilation, the vector containing the lagged discharge is also part of the state vector. The parameters can be explained in more detail as follows:

* is the maximum amount of interception, under the assumption that all interception evaporates
* is a parameter used to describe what factor actually evaporates from the ground (Unsaturated root zone: ) and is
* Is the size of the reservoir of the unsaturated root zone () i.e. the amount of water the top layer of soil can hold. This parameter is used in a few other calculations.
* is a factor controlling the split between fast and slow flow (overland vs groundwater). Some water will be held by the soil whilst some flows over it and straight to the river. which is used to determine the water infiltrating: where is the actual precipitation reaching the ground. The rest flow into the fast reservoir – which is the ground water.
* is the maximum amount of percolation which can occur from the ground to the deeper ground water flow:
* is the lag time between water falling on a given day and all of it reaching the river weighted by a Weibull distribution.
* the fast flow is modelled as a linear reservoir thus a fraction of the volume stored leaves to the river
* Similarly the slow flow is also modelled as .

Methodology

Particle filtering & Likelihood:

To implement the particle filtering the model was first rewritten into a Markov chain structure. After each timestep the state vector of the model can be extracted. This is important for any data assimilation technique in order to apply the update. In the case of particle filtering, the state vector is resampled according to a likelihood function. Particles closer to the observed value are given a higher weight compared to particles further from the observed value and thus are more likely to be chosen when resampling. Using the residuals (observation minus modelled), weights are generated using a log function which are then normalised. This is illustrated in figure 3 below for the 50th day of the model.

A graph of a number of particles

Description automatically generated

*Figure 3: Likelihood function on day 50 of the model.*

Resampling:

Using the likelihoods, the particles are then resampled. This is done using the built in random package from python which chooses based on the provided weights.

Noise:

To avoid the ensemble collapsing to a single state, noise is added to the states of the particles. To generate noise, 0.005 is taken as the standard deviation. The noise is then sampled from a normal distribution and added to the state vector. Higher standard deviation yields a larger spread of particles whilst a smaller standard deviation introduces the chance of collapse. The parameters, the storage and the lag vector all receive this noise according to the same random gaussian noise function.

Observations:

3 years of real daily data obtained from the course ENVM1502 is available to initially compare to and test. From contact with Dr. M. Hrachowitz has been confirmed to be real data however the origin in currently unknown.

Results & Discussion

Model without data assimilation:

As a comparison the model can be run forward 5000 times, each time sampling different set of parameters which are then kept constant throughout the model run. Here we see that sampling within the supplied minimum and maximum values provided (explained above) we can obtain almost all the observed values. This result can be found in figure 4. From all runs, the best is chosen by calculating the Nash-Sutcliffe Efficiency (NSE).NSE assess how well a model fits. NSE is given as one minus the error variance over the variance of the observation: . An NSE of 1 would mean the best possible fit whilst 0 indicates the mean would be a better fit. This best run has an NSE of 0.85 and is shown in orange on the graph. The actual best NSE varies per bulk run of the model as the parameters are chosen at random.   
A graph of a model

Description automatically generated

*Figure 4: 5000 runs of the model for different parameters without assimilating.*

Model with data assimilation:

Running the model for 50 particles yields the plot found in figure 5. We see that the envelope within which the model predicts is much closer to the observations.

 A graph of a graph

Description automatically generated with medium confidence

*Figure 5: Particle filtering with 50 particles assimilating the model at every timestep.*

The mean NSE of the ensemble shown in figure 5 is 0.85 which is lower than the optimal run mentioned above. However the spread of the NSE of the ensemble is between 0.847 and 0.853. This closeness of the minimum and maximum reflects how well the ensemble follows the observations. Increasing the standard deviation of the white noise added to the particles after resampling would also increase the envelope and possibly further reduce the chances of collapse of the ensemble which still occurs in some runs of the model. In testing the wider envelopes caused an NSE value of around 0.6, but always captured all observations in the envelope as shown in figure 6. The higher NSE values shows that signs where the ensemble would collapse into a very narrow interval as shown at the end in figure 7.

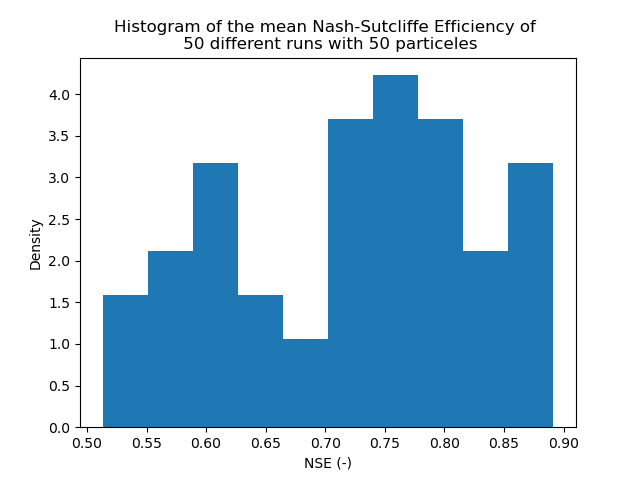
A graph of a graph

Description automatically generated with medium confidenceA graph of a graph

Description automatically generated with medium confidence

*Figure 6: NSE of 0.6 capturing all observations.*  *Figure 7: NSE of 0.87, sometimes collapsing at peaks*

To further investigate this natural variation in the model runs, the model with 50 particles was run 50 times. This shows the spread discussed above. The variation in models is natural as the variations in noise added will vary how the model behaves across different runs. Tweaking the resampling and noise methods could results in a better fit. Using more particles seemed to improve the performance of the model somewhat, however not enough to warrant the use of more particles for this report.



*Figure 8: spread of mean NSE of 50 model runs*

Spin up period

One main advantage of data assimilation is the removal of a spin up period. As the initial conditions in a catchment are unknown and very difficult to measure the first month or two are needed for the model to adjust and ensure that the storage terms are correct as seen in figure 9. In the assimilated particle filter model the state of the particle with the highest likelihood function are inherited by the other particles after a time step. This means that the ensemble quickly diverges to be closer to the observations as seen in figure 10, cutting the spin up period down to 2 weeks instead of 2 months.

A graph of a graph

Description automatically generated with medium confidenceA graph of a graph

Description automatically generated with medium confidence

*Figure 9 and 10: plots comparing behaviour of first half year of the model. Left bulk method, right assimilated.*

Reflection on parameters & Conclusion

The ‘best’ set of parameters from the classical model can be compared to the evolution of the parameters of the particles in the data assimilated model. Note even in the best model from 5000 runs there will be some variation, thus ‘best’ is used in quotes as this isn’t a truth but merely the best that the simple algorithm can produce. In figure 11 we see the two compared with the ‘best’ in orange and the ensemble in the coloured spectrum. Interesting is that the particle filter seems to remove the need for a unsaturated rootzone () and a lag period () and instead compensates for this by adjusting the percolation () and factors Kf and Ks which influence how quickly water moves through the subsurface (fast - ) and groundwater (slow – ). The adjustments in these parameters physically doesn’t make that much sense as these parameters should be more or less constant in a catchment. Having said that, the way moist soil reacts is very different from how dry soil reacts. Something which isn’t fully captured by the model, but is compensated for by the data assimilation technique. This shows that for a complex system which isn’t fully described by the model data assimilation can correct the state of the model in order to improve the modelling of the system. Further research could be done to see how the model would perform for predictions, for example using weekly periods of updating data to predict the discharge in the coming week given a rain scenario or given rain data when the discharge data hasn’t arrived yet. Rain data from radar observations can be received much quicker than the response of a catchment. Data assimilation could be used to quickly adjust the predictions for floods in a given region, similar to how weather forecasting uses observations to adjust forecasts.

A group of graphs with numbers

Description automatically generated

*Figure 11: spread of parameters over the model time*