Data assimilation with HYPE

Introduction

The hydrological simulation system used to run the hydrological model HYPE can be used for data assimilation with the ensemble Kalman filter method. The ordinary HYPE executable does not contain this functionality, but compiled with a flag for assimilation you get an executable that can be used for data assimilation with HYPE. The base for the data assimilation is a propagation of an ensemble of models forward in time, one step at the time. Forcing data is perturbed (a little and differently) for the ensemble members by a special routine in the assimilation code to give the ensemble a little variance. The perturbations are spatially correlated with a certain correlation length.

After a time step has been simulated for the whole ensemble, and if there are some observations that are to be assimilated for that time step, a so called “analysis” is made. The analysis will use the deviations between observations and corresponding modelled predictor to adjust (or update) the states of the model. HYPE uses the ensemble Kalman filter (enKf) method for the analysis.

The base of the method is the Kalman filter:

$$X_{analysis} = X_{predictor} + K \times \left(Y_{predictor} - Y_{observation} \right)$$

where X is the model’s state variables and Y is its observation variables. The matrix K, the Kalman gain, is calculated as a function of the covariance between error in the model predictors and model states.

Thus, the Kalman filter can adjust all model states with a selected set of observations (including observations that are not of the model states) as long as the observation variable can be predicted from the model states. The function that predicts the observation variable is called an observation operator. For instance, discharge (COUT), which is not a model state but a result of other model states (primary the water stored in soil, rivers and lakes) is the predictor of observed discharge (ROUT).

For simple models the covariance between errors in a model predictor and the model states can be theoretically derived, but for more complex models like HYPE that is difficult. This is where the “ensemble” of the ensemble Kalman filter comes into play. Instead of calculating the covariance matrices needed for the analysis, they are estimated from the ensemble of models. The model errors of the predictors and the model states are sampled from the ensemble and the covariances are calculated for the samples.

Method

The classic ensemble Kalman filter method assumes a linear model, an infinite ensemble size and Gaussian error distribution. These assumptions are usually violated since the use of very many ensemble members quickly becomes a computational bottleneck in large domains or with long simulation runs. The normally-distributed error assumption has also been found to be invalid for physically bounded variables like soil moisture. The lognormal and logit-normal transformations have been successfully applied to semi-bounded e.g., precipitation and bounded e.g., soil moisture, respectively.

At each time step in the assimilation run, an ensemble of random but finite perturbations is introduced in the forcing (precipitation and temperature) and observations (state variables) that results into randomly generated model trajectories. The perturbations in the forcing are spatially correlated fields that propagate information to locations where it may be missing, with the spatial correlations themselves controlled by the so-called spatial localization, which suppresses superfluous covariances that can arise e.g., in regions with complex terrain.

In the next step, the ensemble of model states and fluxes is propagated through the dynamic model and transformed to the observation space, where the predicted observation and observation error terms mentioned above are obtained. The localization is included by modifying the error covariance matrix terms present in the Kalman gain matrix. These are multiplied by two scalar matrices that restrict the influence of the observations on updated variables using element-wise products. One ensures that the distance-dependent functions used to define the scalar matrices diminish to zero at appropriate separation lengths.

Input and output files

There is one additional input file AssimInfo.txt where the data assimilation settings are defined. A few settings are given in the ordinary information file. First, assimilation needs to be turned on in the info.txt file (`assimilation Y`). In addition, it is possible to restart an assimilation from a saved ensemble of states (`indaensstate`) and to save ensemble states for later restart (`outdaensdate`).

The assimilation may give additional output files in addition to the usual files requested for in info.txt. In the usual files, the assimilation routine can write the mean or median result of the ensemble. Whether you get the mean or median depends on the flag set in AssimInfo.txt. In addition, you can get duplicated files for minimum and maximum value of the ensemble, standard deviation or each ensemble member. The extra files are separated by a suffix (number) appended to the ordinary output file name. For example timeCOUT_002.txt could hold the ensemble minimum values.

An assimilation simulation may temporarily save the ensemble states in binary files instead of keeping them in memory. This function can be used to restart an assimilation simulation at previously saved states, but can also be used for large model setups to save memory during the simulation. The files are either one for each ensemble (nnnnnn.bin) or one for each of the three variable types; state ensemble, forcing ensemble and auxiliary ensemble (ensXstates.bin, ensFstates.bin, ensAstates.bin).