

**ensemblekfilter.m –**

**Ensemble Kalman Filter Function in MATLAB**

**Joseph Squeo**

**Advisor: Xinyu Zhao**

**9/14/2019**

**Overview:**

The ensemble Kalman filter is a model optimization technique used to estimate a state, subjected to Gaussian noise, forward in time [1]. Given a state of a system, measurements and random noise, the ensemble Kalman filter provides a future estimate of the state at time An ensemble of the state is generated, updated in an analysis and advanced in time in a forecast step [1].

**Function Explanation:**

The following equations and explanations refer to the MATLAB function “*ensemblekfilter.m”*. The code contains numbered steps commented out to split the code into sections for easier understanding. All code is referenced from *Gillijns et. al.*, 2006 [1] and *Evensen et. al.*, 2010 [2].

\*Note the inputs and outputs of the function are listed in detail within the function itself.

1.ITERATE THROUGH TIME

Starts the *for* loop to iterate through each time index-. The loop starts at because is treated as the initial condition. Additionally, the truth state is given as:

Note that noise is added to the truth state during each time iteration, so the random Gaussian distributed noise is compounded with time. Matrix contains the discretized governing equations for each of the states (or cells) and is of size , where is the number of states. is the current truth state and is initialized with the initial condition for . is a column vector of size containing the mapping of cells or states that the source, affects. is a row vector containing the source term for each time interval, and of size . Lastly, is the random Gaussian distributed noise with standard deviation specified by the user input and an expected value of zero.

2.ITERATE THROUGH ENSEMBLE MEMBERS

Within the time loop, another loop iterates through from 1 to the total number of ensembles. A new random noise column vector is generated for each individual state for each ensemble for the model and measurement noise, and . Standard deviations and control the deviation of the random noise sample. Additionally, sample error is generated to initialize the first ensemble of the state when , with its associated standard deviation, .

The forecasted measurement is generated for each individual state and ensemble and stored in a matrix, along with the measurement with random noise.

Here, is the forecasted measurement, the measurement with random noise and the ensemble of the state with the random sample error when . When , the current estimated state is used to generate the new forecasted measurements. Since the model operator matrix contains the boundary conditions, noise is removed from the first and last rows of the noise matrices to discard noise from the boundaries when the state is updated.

Recall that must have a measurement for each state at each time, . If not, the user must use “ as a place holder. There is a few lines of code [151-156] that finds the elements of that are and replaces these values with the corresponding forecasted measurement values . This way, when the Kalman gain matrix is calculated below, the state will not be updated from a measurement if the value in .

3.AVERAGE ALL ENSEMBLE MEMBERS

The average value of each state for forecasted measurement is obtained from the ensemble. Since random Gaussian noise is added to the state and measurement in the previous steps, the larger the ensemble, the closer the expected value of the ensemble will be to the actual value.

4.ENSEMBLE ERROR MATRICES

The ensemble error matrices are formed for the estimated state and forecasted measurement to approximate the respective covariance matrices in step 5. These error matrices are controlled by the random noise.

5.ESTIMATE COVARIANCE MATRICES

Covariance matrices are approximated. These matrices are symmetrical since they are generated by a matrix times its transpose. When is the identity matrix to map the state estimate to the forecasted measurement, the matrices in (5,6) are equivalent making the following matrices equivalent.

6.KALMAN GAIN MATRIX

Kalman gain matrix is calculated to update the state in the analysis step.

\* is often nearly singular, so singular value decomposition (SVD) is used to find a pseudo-inverse of the matrix by removing unnecessary “noise” in the matrix causing singularity issues.

7.ANALYSIS STEP

The analysis step estimates the state at the current time by performing an ensemble of parallel data assimilation cycles. The resulting matrix is of size where is the number of states (or cells) and is the number of ensemble members.

8. FORECAST STEP

Lastly, the state is updated forward in time ( Note in the code, the previous step and current step are stored as the same variable, . When the loop finishes, this is used as the new best estimate of the state. Also, used here is from the previous step, equation (10). The random noise is unique from the random noise added to the true state.

9.STORE INTO OUTPUT VARIABLES

The mean-square error (MSE) is quantifed as the squared difference between the truth state and the average value of the ensemble members of the predicted state. As the number of ensemble members increases, the noise in the EnKF state prediction decreases along with the mean-squared error. The average of the ensemble members of equation (11) is found and stored as the EnKF state output, . Lastly, the first row of the matrix is initialized at time using the initial condition, , specified by the user-input. A function called “*ensemblekfilter\_plots*” is called to plot the MSE and EnKF truth state, measurements and EnKF output.

References

[1], [2]

[1] S. Gillijns, O. B. Mendoza, J. Chandrasekar, B. L. R. De Moor, D. S. Bernstein, and A. Ridley, “What is the ensemble Kalman filter and how well does it work?,” 2006, p. 6 pp.

[2] G. Evensen, *Data assimilation: The ensemble kalman filter*. 2010.