README

Dong-Jun (DJ) Seo
djseo@uta.edu
Aug 14, 2021

1. Introduction

This document describes the AEnKF_for_SACUH package for adaptive conditional bias-penalized ensemble Kalman filter (AEnKF) for the Sacramento (SAC) soil moisture accounting and unit hydrograph (UH) models. It also describes how to run the example for 4 forecast points and plot the results, and how one may use the AEnKF-specific subroutines for other possible applications. This self-contained package includes the source code, the hydrologic model parameter files, the hydrologic model input files, a script to run the example, and the R scripts to plot the results (see Table 1).

The source code, *AEnKF_for_SACUH_for_CandG.f90*, uses AEnKF to assimilate retrospectively the real-time observations of streamflow, precipitation and potential evapotranspiration into SACUH to improve streamflow prediction. Most of the subroutines in the source code is to support the hydrologic models, rather than the AEnKF algorithm. In the source code, only the 3 subroutines, *eval_aenkf_gain, eval_opt_alpha1* and *eval_opt_alpha2*, and lower-level subroutines *eval_aenkf_gain* calls comprise the AEnKF algorithm.

The AEnKF-specific subroutines are heavily commented. As such, it is expected that the user will be able to adapt the code for other applications with a minimal to modest amount of effort. Due to the legacy nature of the hydrologic models and the supporting subroutines, the source code is mostly in a Fortran 77 style with only a few exceptions, including dynamic memory allocation and matrix algebra.

2. Files included

The following files are included in the self-contained package.

Table 1. List of files included in the AEnKF_for_SACUH package.

Name	Description		Note
AEnKF_for_SACUH_for_CandG.f90	Source code		
SACSMA_COLI2_COLI2_UpdateStates.xml	Sacramento model (SAC) parameters for:	COLI2	
SACSMA_DLTC1H_DLTC1HLF_UpdateStates.xml		DLTC1	
SACSMA_GTBM3SNE_GTBM3SNE_UpdateStates.xml		GTBM3	
SACSMA_MONN7_MONN7_UpdateStates.xml		MONN7	
UNITHG_COLI2_COLI2_UpdateStates.xml		COLI2	See Seo et al.
UNITHG_DLTC1H_DLTC1_UpdateStates.xml		DLTC1	

UNITHG_GTBM3SNE_GTBM3SNE_UpdateStates.xml	Unit hydrograph model (UH) parameters for:	GTBM3	(2021) for details.
UNITHG_MONN7_MONN7_UpdateStates.xml		MONN7	
new_map06_COLI2	Mean areal precipitation (MAP) data for:	COLI2	
new_map06_DLTC1		DLTC1	
new_map06_GTBM3		GTBM3	
new_map06_MONN7		MONN7	
COLI2.qin	Streamflow data for:	COLI2	
DLTC1.qin		DLTC1	
GTBM3.qin		GTBM3	
MONN7.qin		MONN7	
run_AEnKF_for_SACUH	Script for running the executable for the 4 locations		
plot_mean_crps.R	R script for plotting the mean CRPS results for the 4 locations		See Seo et al. (2021) to verify the results.
plot_rmse.R	R script for plotting the RMSE results for the 4 locations		

3. Running the example

1) Compile the source code. The simplest way is to type *gfortran AEnKF_for_SACUH_for_CandG.f90* which will generate the executable *a.out*. If you have an Intel® compiler, which is significantly faster, type *ifort* instead of *gfortran*.

2) Once compiled, run the executable for the 4 locations by typing *./run_AEnKF_for_SACUH* > *capture* where *capture* is the file name of your choice to capture all output written to the screen.

3) Once the run is complete, run the R scripts, *plot_mean_crps.R* and *plot_rmse.R*, to generate jpeg files for mean continuous ranked probability score (CRPS) of ensemble prediction and root mean square error (RMSE) of ensemble mean prediction vs. lead time, respectively. The 4 jpeg files from *plot_mean_crps.R* should produce the 4 panels shown in Fig 1. Similarly, the 4 jpeg files from *plot_rmse.R* should produce the 4 panels shown in Fig 2.

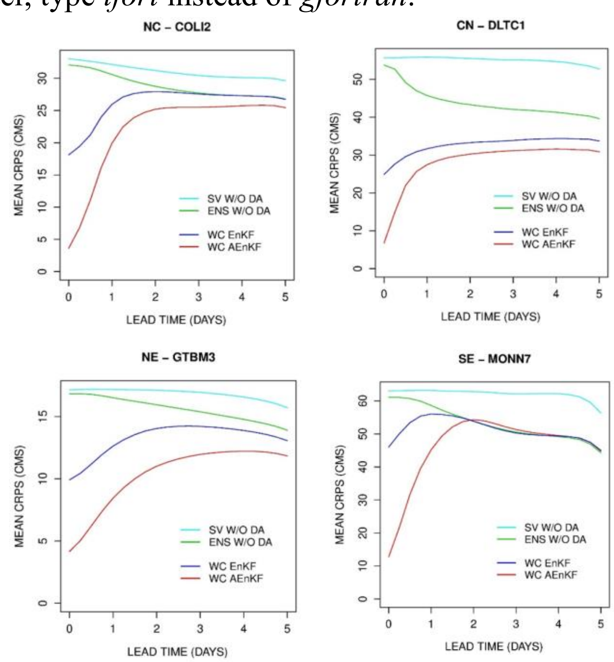

Fig 1. Mean CRPS results from running *plot_mean_crps.R.*

4. Adaptation for other applications

To use the AEnKF-specific parts of the source code for other applications, it is necessary to modify subroutine *get_h_and_r* in subroutine *eval_aenkf_gain*. Subroutine *get_h_and_r* specifies the observation structure matrix, H_k, the observation error covariance matrix, R_k, and its inverse, R_k^{-1}.

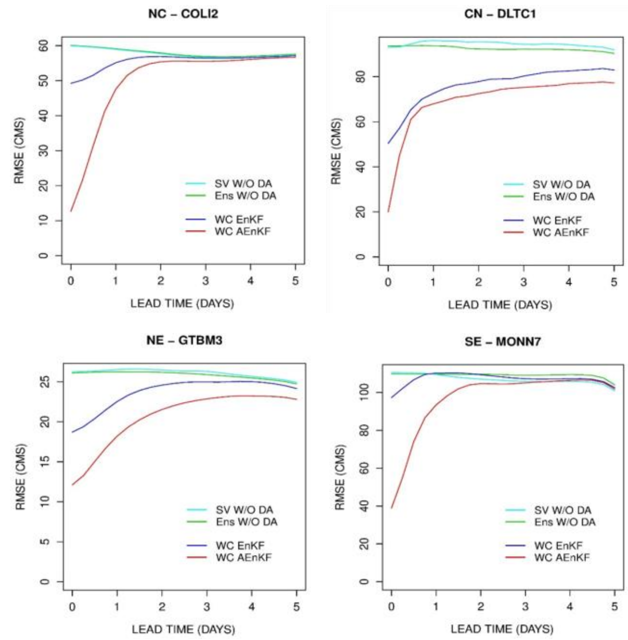

Fig 2. RMSE results from running the example.

Subroutines *eval_opt_alpha1* and *eval_opt_alpha2* evaluate the first- and second-order derivatives of the degrees of freedom for noise, $d_{n,k}$, with respect to α_k for its optimization using the Newton's method. They follow Subsection 2.2 of Seo et al. (2021). Subroutine *eval_opt_alpha1* evaluates all terms in Subsection 2.2 that are not ensemble member-specific and hence may be evaluated only once. Subroutine *eval_opt_alpha2* evaluates all terms in Subsection 2.2 of Seo et al. (2021) that are ensemble member-specific and hence evaluated for each ensemble member. These subroutines are general and no modification are necessary. To help the user provide the subroutines with correct arguments, they are extensively commented.

References

Seo, D.-J., H. Shen, and H. Lee, 2021. Adaptive conditional bias-penalized Kalman filter with degrees of freedom for noise minimization for superior state estimation and prediction of extremes, submitted to Computers and Geosciences.