***AEnKF_for_SACUH***

## Description

The *AEnKF_for_SACUH* package implements adaptive conditional bias-penalized ensemble Kalman filter (AEnKF, Shen et al. 2022a,b) and, for comparison, ensemble Kalman filter (EnKF, Lorentzen and Naevdal 2011) for streamflow prediction using the Sacramento (SAC, Burnash et al. 1973) and unit hydrograph (UH, Chow et al. 1988) models. The two hydrologic models are referred to herein as SACUH.

This document describes the content of the package, how to create an executable and use it, how to run the test for 4 forecast points and how the AEnKF-specific subroutines may be adapted for possible implementation of AEnKF for other applications. The *AEnKF_for_SACUH* package is self-contained and includes the source code, the filter parameter files, the hydrologic model parameter files, the hydrologic model input files, a script to run the test, and the R scripts to plot the summary results. Table 1 lists all files in the package.

The all-inclusive source code, *AEnKF_for_SACUH_for_CandG.f90*, uses AEnKF and EnKF to assimilate retrospectively in a real-time mode with clairvoyant forcings the observations of streamflow, mean areal precipitation (MAP) and mean areal potential evapotranspiration (MAPE) into SACUH to improve streamflow prediction (Shen et al. 2022a). Most of the subroutines in *AEnKF_for_SACUH_for_CandG.f90* support the hydrologic models. Only the 3 subroutines, *eval_aenkf_gain, eval_opt_alpha1* and *eval_opt_alpha2*, and the lower-level subroutine *eval_aenkf_gain* are specific to the AEnKF algorithm as described in Seo et al. (2022).

SACUH have been used extensively for operational hydrologic forecasting in the US National Weather Service (NWS) for over a generation. For this reason, *AEnKF_for_SACUH_for_CandG.f90* largely retains the Fortran 77 style with a few exceptions, including dynamic memory allocation and matrix algebra.

To aid understanding of the code, *AEnKF_for_SACUH_for_CandG.f90* is commented throughout. In particular, the AEnKF-specific subroutines are commented extensively for those interested in implementing the AEnKF algorithm for other applications. It is expected that the user who is familiar with Fortran will be able to adapt the code with a modest level of effort (see the Adaptation for Other Applications Section below).

## Content

Table 1 lists the files 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* | All-inclusive source code | See the Description Section above for details. |
| *params_COLI2.in* | Parameters for AEnKF and EnKF for SACUH | See the Usage Section below for details. |
| *params_DLTC1.in* | | |
| *params_GTBM3.in* | | |
| *params_MONN7.in* | | |
| *SACSMA_COLI2_COLI2_UpdateStates.xml* | Parameters for SAC | See Seo et al. (2022) for details. |
| *SACSMA_DLTC1H_DLTC1HLF_UpdateStates.xml* | | |
| *SACSMA_GTBM3SNE_GTBM3SNE_UpdateStates.xml* | | |
| *SACSMA_MONN7_MONN7_UpdateStates.xml* | | |
| *UNITHG_COLI2_COLI2_UpdateStates.xml* | Parameters for UH | |
| *UNITHG_DLTC1H_DLTC1_UpdateStates.xml* | | |
| *UNITHG_GTBM3SNE_GTBM3SNE_UpdateStates.xml* | | |
| *UNITHG_MONN7_MONN7_UpdateStates.xml* | | |
| *new_map06_COLI2* | MAP data | |
| *new_map06_DLTC1* | | |
| *new_map06_GTBM3* | | |
| *new_map06_MONN7* | | |
| *COLI2.qin* | Streamflow data | |
| *DLTC1.qin* | | |
| *GTBM3.qin* | | |
| *MONN7.qin* | | |
| *run_AEnKF_for_SACUH* | Script for running the executable for testing | See the Testing Section below for details. |
| *plot_mean_crps.R* | R script for plotting the mean CRPS results | |
| *plot_rmse.R* | R script for plotting the RMSE results | |

**Requirements**

- Linux environment
- GNU or Intel® Fortran compiler
- R

**Installation**

R (if not already installed)

**Usage**

Step 1) Compile the source code. The simplest way is to generate an executable is to type the following compilation command which will produce the executable named *a.out* by default:

*gfortran AEnKF_for_SACUH_for_CandG.f90*

If an Intel® compiler is available for faster execution, type *ifort* instead of *gfortran* in the above.

Step 2) Choose a forecast point. The package includes files for 4 forecast points: COLI2, DLTC1, GTBM3 and MONN7 (see Table 2 for summary information and Seo et al. 2022 for further details).

Table 2. Forecast points for testing and their attributes.

| NWS ID | RFC | Geographic name | USGS ID | Starting time (yyyymmddhh) | Ending time (yyyymmddhh) | Area (km^2) | UH duration (hrs) | Cutoff (cms) | N |
|---|---|---|---|---|---|---|---|---|---|
| GTBM3 | NE | Housatonic River at Great Barrington, NY | 1197500 | 1990100106 | 2017123106 | 730 | 5 | 60 | 3248 |
| COLI2 | NC | Mazon River near Coal City, IL | 5542000 | 1986090106 | 2013123124 | 1178 | 2.5 | 50 | 5416 |
| DLTC1 | CN | Sacramento River at Delta, CA | 1.1E+07 | 1988100106 | 2019093024 | 1100 | 3.5 | 175 | 5113 |
| MONN7 | SE | Deep River near Moncure, NC | 2102000 | 1985100106 | 2010123118 | 2840 | 7.5 | 300 | 3954 |

Step 3) Verify the settings of the filter parameters in *params_*****.in* where ***** denotes the 5-character identifier for the forecast point chosen. The *params_*****.in* files in the package contain the default settings for the following 11 parameters appearing in *AEnKF_for_SACUH_for_CandG.f90*:

- *varp* – Observational error variance in $mm^2$ for 6-hr MAP. Not used for the heteroscedastic option (see Step 4 below).
- *vare* – Same as *varp* but for MAPE. Not used for the heteroscedastic option.
- *varw* – Same as *varp* but for 6-hr runoff depth, i.e., Total Channel Inflow (TCI) in SAC. Not used for the heteroscedastic option.
- *varq* – Observational error variance in $(m^3/s)^2$ for instantaneous streamflow in $m^3/s$. Not used for the heteroscedastic option.
- *ns* – Ensemble size, i.e., the number of ensemble traces to be used in AEnKF and EnKF.
- *nf* – Number of streamflow observations within the assimilation window to be used in DA. The default is 1, i.e., use the single observation valid at the end of the assimilation window.
- *ifrq* – Sampling frequency of streamflow observations within the assimilation window. This parameter is relevant only if *nf* > 1. The default is 1.
- *jseed* – Seed for random number generation in AEnKF and EnKF.

- *iscale* – Length of the subwindow in $hrs$ which equally divides the assimilation window to build the control vector for MAP, MAPE and TCI. If *iscale*=0, the assimilation window is not subdivided and MAP, MAPE and TCI are adjusted uniformly across the entire assimilation window. If *iscale* is greater than 0, the value of *iscale* specifies the length in $hrs$ of the subwindow. The smallest subwindow allowed is 1 $hr$.
- *fra_cold* – Variance of the perturbation for the model dynamical error expressed as a fraction of the state variable for cold-state data assimilation (DA) runs. The default is 0.01, i.e., 1% of the current state.
- *fra_warm* – Same as *fra_cold* but for warm-state DA runs. The default is 0.01.

For testing (see the Testing Section below), it is recommended that the default settings be used.

Step 4) Run the executable. If, e.g., GTBM3 is chosen, type:

./a.out GTBM3 1 0 > capture_GTBM3

The first command-line argument specifies the 5-character identifier of the forecast point of choice.

The second argument of '1' specifies that the heteroscedastic option (see Shen et al. 2022a for details) is to be used for uncertainty modeling. An argument of '0' would use instead the homoscedastic option, in which case the observational error variance values prescribed in *params_GTBM3.in* are used.

The third argument of '0' specifies that the hydrologic models is to be weakly-constrained (see Shen et al. 2022a for details). An argument of '1' would use the strongly-constrained formulation under the generally unrealistic assumption that the hydrologic models have no structural or parametric uncertainties.

The above command would hence run weakly-constrained AEnKF for SACUH for GTBM3 and save all output written to the screen to the file named *capture_GTBM3*.

Step 5) Once the run is complete, verify that the output files are successfully generated in subdirectory */output0*. The suffix '0' in the subdirectory name signifies that the output is from a weakly-constrained run. A strongly-constrained run would write output to */output1* instead.

In */output0*, subdirectory */GTBM3* should hold two subdirectories, *run_GTBM3_enkf_10* and *run_GTBM3_aenkf_10* which hold the EnKF and AEnKF output, respectively. In the above, the suffix '10' signifies that the output is from a heteroscedastic ('1') weakly-constrained ('0') run. The subdirectories mentioned in this step are all created automatically under the directory where the executable is run.

The output files specific to EnKF or common to both EnKF and AEnKF are written into */run_GTBM3_enkf_10*:

- *GTBM3.fcst_base* – DA-less (i.e., base) ensemble streamflow predictions for lead times of 6 to 120 hours
- *GTBM3.fcst_enkf* – EnKF-aided ensemble predictions
- *GTBM3.fcst_obs* - Verifying observations matched with predictions
- *GTBM3.fcst_sngl* – DA-less single-valued streamflow predictions

The output files specific to AEnKF are written into */run_GTBM3_aenkf_10*:

- *GTBM3.fcst_suppl* – Supplemental information including the degrees of freedom for signal
- *GTBM3.fcst_aenkf* – AEnKF-aided ensemble predictions

If desired, one may repeat Step 4 for the other 3 forecast points for Step 6 below or follow the testing procedure described in the Testing Section below.

Step 6) Plot the root mean square error (RMSE) and mean continuous ranked probability score (CRPS) vs. lead time for all 4 forecast points by running the R scripts, *plot_rmse.R* and *plot_mean_crps.R*, respectively. To run them in a batch mode, type:

*R CMD BATCH plot_mean_crps.R Rout_mean_crps*

R CMD BATCH *plot_rmse.R Rout_rmse*

Each run will display and produce 4 jpeg files, *mean_crps_*****.jpg* and *rmse_*****.jpg*, corresponding to the 4 forecast points.

**Testing**

The testing procedure steps through the usage described above by running each of the following 4 commands successively, each time verifying that the execution is successful:

*gfortran AEnKF_for_SACUH_for_CandG.f90*

*./run_AEnKF_for_SACUH > capture*

*R CMD BATCH plot_mean_crps.R Rout_mean_crps*

R CMD BATCH *plot_rmse.R Rout_rmse*

In the above, *run_AEnKF_for_SACUH* runs Step 4 in the Usage Section for the 4 forecast points. Once completed, verify that the 4 jpeg files each from *plot_mean_crps.R* and *plot_rmse.R* match the 4 panels in Figs 1 and 2 shown below, respectively.

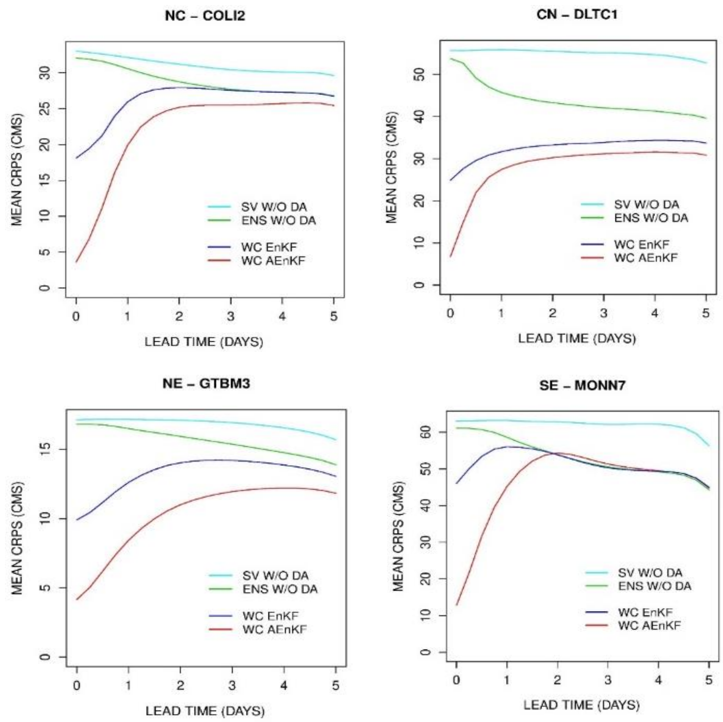


Fig 1. Mean CRPS results.

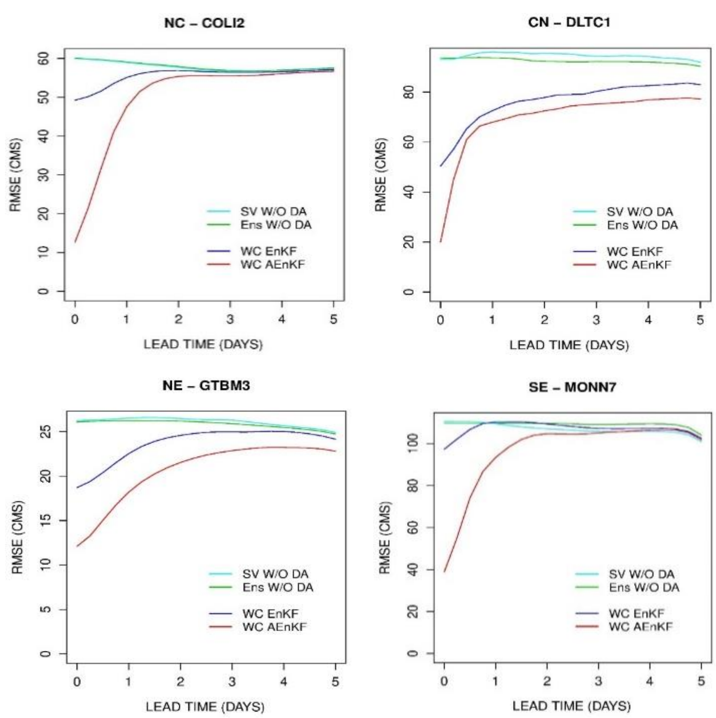


Fig 2. RMSE results.

## Adaptation for Other Applications

To use the AEnKF-specific subroutines in *AEnKF_for_SACUH_for_CandG.f90* 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, $R_k^{-1}$.

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 $\alpha_k$ for its optimization using the Newton's method. The above subroutines follow Subsection 2.2 of Seo et al. (2022) as described below.

Subroutine *eval_opt_alpha1* evaluates all terms in Subsection 2.2 of Seo et al. (2022) that are not ensemble member-specific and hence may be evaluated only once. Subroutine *eval_opt_alpha2* evaluates all terms in Subsection 2.2 that are ensemble member-specific and hence must be evaluated for each ensemble member. These subroutines are general and should not require modifications.

## References


Burnash, R.J., Ferral, R.L., McGuire, R.A., 1973. A generalized streamflow simulation system: Conceptual modeling for digital computers. U.S. Department of Commerce National Weather Service and State of California Department of Water Resources.

Chow, V.T., Maidment, D.R., Mays, L.W., 1988. Applied Hydrology. McGraw-Hill Series In Water Resources and Environmental Engineering. McGraw Hill, NY.

Lorentzen, R.J., Naevdal, G., 2011. An iterative ensemble Kalman filter. IEEE Transactions on Automatic Control 56, 1990–1995. https://ieeexplore.ieee.org/document/5766715

Seo, D.-J., H. Shen, and H. Lee, 2022. Adaptive conditional bias-penalized Kalman filter with degrees of freedom for noise minimization for superior state estimation and prediction of extremes, resubmitted to Computers and Geosciences.

Shen, H., D.-J. Seo, H. Lee, Y. Liu, and S. Noh, 2022a. Improving flood forecasting using conditional bias-aware assimilation of streamflow observations and dynamic assessment of flow-dependent information content, Journal of Hydrology, 605. https://doi.org/10.1016/j.jhydrol.2021.127247

Shen, H., Lee, H., Seo, D.-J., 2022b. Adaptive Conditional Bias-Penalized Kalman Filter for Improved Estimation of Extremes and Its Approximation for Reduced Computation. Hydrology 2022, 9, 35. https://doi.org/10.3390/hydrology9020035