User manual LOTOS-EUROS Local Ensemble Kalman Smoother-4D

The Local Ensemble Kalman Smoother-4D (LEKS-4D) is based on the Local Ensemble Kalman Filter (LEKF) module provided by TNO and integrated into the LOTOS-EUROS (LE) model. This module is not part of the open-source version of LE, and TNO must approve its use. The LEKS-4D uses the creation and propagation of ensembles, the generation of the noises used to estimate parameters, and the processor of instrument observations included in the LEKF. The LEKS-4D analysis step is executed externally to the LEKF. Being closely related to the LEKF, the LEKS-4D folders are integrated into the LEKF directory. From now on, **$my_LEKF_path** will denote the path where the folder containing the LEKF is located.

El LEKS-4D consta de dos carpetas:

1. **$my_LEKF_path**/run_smoother_v2: Contains the LEKS-4D code. In the LE_communication folder are the scripts that allow LEKS-4D to communicate with LE, managing the compilation and execution of the model and the reading and writing of inputs and outputs. The LETKF_4D folder contains the scripts that allow you to perform the filter update step. Simulation configuration files are saved and run from the root.
2. **$my_LEKF_path**/proj/smoother_V2/002: This folder contains the LEKS-4D Project in the same format as a LEKF Project. LEKS-4D overwrites some configuration files to keep LEKF simulations compatible with LEKS-4D analysis. For more information on how the folder is structured, see the LEKF documentation.

Configuring a LEKS-4D run

A .py file located in the $my_LEKF_path/run_smoother_v2 folder is used to configure and run a simulation of the LEKS-4D. In this folder, a file serves as a template Template_smoother_2_V1.py. It is recommended to follow the following order (the template file has the necessary comments to follow the steps):

1. Define the scratch path (**$my_LE_scratch**) where the LE simulation will be executed. This path is configured in the LE machine.rc file. For more information, see the LE documentation.
2. Define the window of assimilation in days. Precisely the Background step (Tb) and assimilation step (Ts).
3. Define the name of the simulation, the start date, and the number of assimilation cycles (number of Tb).
4. Define the number of ensembles to use and the location radius in degrees.
5. Define the observations to be used in the assimilation. The template file contains the names corresponding to the operators developed. In case of an error when one of these operators is not found in the model outputs, check the lotos-euros-output.rc file in the LE Project folder. The operators to be included in the model simulation are defined in the CSO section.

6. Define if the noise in the emissions and other parameters will be created using spatial correlation.
7. Define if a short initial run of 2 hours will be used to generate model output samples and automatically derive the dimensions and domains used.

After setting up the simulation, the python file must be executed to run the simulation. The python file is responsible for performing the necessary LE compilations and LE executions.

LEKS-4D outputs

The LEKS-4D generates outputs that contain the correction factors of the parameters to be estimated. The parameters to be evaluated are defined in $my_LEKF_path/proj/smoother_V2/002/rc/lekf.rc following the same process as in a LEKF simulation. The files with the correction factors, called DC files, are saved in the folder $my_LE_scratch/proj/Smoother_V2/$Project_name/dc_smoother/, there they are read by LE and incorporated into the simulation. Additionally, LE generates a version of these files along with the other model outputs in the $my_LE_scratch/proj/Smoother_V2/$Project_name/outputs folder.
In the model outputs folder, files are generated for each group of outputs specified in the LE simulation settings (see LE documentation) and output files for each satellite operator. The operator output files contain a list with the values in each model pixel and the instrument. It is possible to generate gridded files where these files are resampled to the model domain. For this, the CSO tool supplied by TNO is used. For more information, consult the CSO documentation.