Experiments with ensemble DA in L63 and L96

ECMWF/NCEO data assimilation training course

March 8, 2018

1 Objective

This activity will allow the student to experiment using ETKF and LETKF in some toy models. In particular, the student will use the Lorenz '63 model with 3 variables, and the Lorenz '96 model with $N_x = 12$ variables. Parameter estimation via state augmentation is also explored in the Lorenz '63 model.

2 Review of Theory

Ensemble Kalman filters are Monte-Carlo implementations of the Kalman filter, or the Extended Kalman filter in the case of non-linear evolution models. In the Kalman filter, the analysis update formulas are:

$$\begin{aligned} \bar{\mathbf{x}}^a &= (\mathbf{I} - \mathbf{KH})\,\bar{\mathbf{x}}^b + \mathbf{Ky} \\ \mathbf{A} &= (\mathbf{I} - \mathbf{KH})\,\mathbf{B} \\ \mathbf{K} &= \mathbf{BH^T}\left(\mathbf{HBH^T} + \mathbf{R}\right)^{-1} \end{aligned} \tag{1}$$

where $\bar{\mathbf{x}}^b \mathcal{R}^{N_x}$ is the background mean, $\bar{\mathbf{x}}^a \mathcal{R}^{N_x}$ is the analysis mean $\mathbf{y} \in \mathcal{R}^{N_y}$ is the observation, $\mathbf{H} \in \mathcal{R}^{N_y \times N_x}$ is the observation operator, $\mathbf{B} \in \mathcal{R}^{N_x \times N_x}$ is the background error covariance, $\mathbf{R} \in \mathcal{R}^{N_y \times N_y}$ is the observation error covariance, and $\mathbf{A} \in \mathcal{R}^{N_x \times N_x}$ is the analysis error covariance.

The ensemble Kalman filter uses a sample estimator for B. Moreover, the equations in (1) are difficult to compute. In this practical we use two formulations:

- The Stochastic Ensemble Kalman filter of Burgers et al 1999 and its localised version.
- The Ensemble Transform Kalman filter of Wang et al. (2004) and Hunt et al (2007). This deterministic square root filter acts on ensemble space.

3 Instructions for Lorenz 63

These are the python files used in this part of the activity. Note that in this small (3-variable) model you will not require localisation. However, in this system we perform parameter estimation exercises.

- *ControlL63Enkf.py*. This is the control file, and it is the one which you will be running and modifying.
- *L63model.py*. This file contains all the instructions to run the L63 model.

- *L63misc.py.* This file generates different observation operators, creates the observations, and generates a simple background error covariance matrix.
- *L63kfs.py.* This file contains the routines to perform SenKF and ETKF. This includes computing the tangent linear model and transition matrices.
- *L63plots.py.* This file has instructions for different plots.

You will run different sections of the file *ControlL63EnKF.py.* These are enumerated as comments of the file (recall that in python # is used for comments). To run **only** a section of a file you can highlight the desired instructions with the mouse, and the press F9.

3.1 Instructions

- The first lines of the file import the different packages that the file uses: numpy, matplotlib, and the functions we have created for this activity.
- *Section 1.* This section generates the **nature** run of the experiment, i.e. what we consider to be the true system. You can change the initial conditions, the final time (consider that the model time step is 0.01 time units), and the initial guess from which the assimilation will start. You can also play with the parameters of the model to see how the behaviour of the system changes. However, for the final experiment you should leave this at $\boldsymbol{\theta} = (10, 8/3, 28)$. Running this section should also plot the 3D phase space (time is implicit in this figure), and time evolution plots for each one of the variables.
- *Section 2.* This section is related to the observations. You can select to observe different variables: e.g. all of them: $'xyz'$, or subsets: $'xz'$ or $'y'$. Different choices will create the observation matrix. The $\mathbf{R}$ matrix is designed to be diagonal (common assumption), but you can choose the observational variance. You can also choose the observational period (in number of model steps). As a rule of thumb, observations every 8 steps yield a quasi-linear problem, whereas observations every 25 steps yield a full non-linear problem.
- *Section 3.* In this section you perform the DA experiments using the Kalman filets. You can vary the number of ensemble members M, the inflation ρ, and the method: either 'SEnKF' ot 'ETKF'.
- *Section 4.* In this section you can perform parameter estimation. You can vary the initial 'guess' value of the parameters (parambad), as well as the variance of the random evolution of the parameters in the forecast (α).

Note that in this exercise a new metric is computed along the RMSE. This is the spread of the analysis ensemble. In a healthy ensemble system these two have a relationship, what is that relationship?

As usual, repeat the previous experiments with different combinations of parameters in the problem.

4 Instructions for Lorenz 96

These are the python files used in this part of the activity. You will perform localisation experiments with the SEnKF. The R-localisation needed for LETKF is very slow without parallel implementation, so we do not perform that in this exercise.

- *ControlL96EnKF.py*. This is the control file, and it is the one which you will be running and modifying.
- *L96model.py*. This file contains all the instructions to run the L96 model.
- *L96misc.py*. This file generates different observation operators, creates the observations, and generates a simple background error covariance matrix.
- *L96Kfs.py*. This file contains the routines to perform 3DVar and SC4DVar. This includes computing the tangent linear model and transition matrices.
- *L96plots.py*. This file has instructions for different plots.

You will run different sections of the file *ControlL96Var.py*. These are enumerated as comments of the file (recall that in python # is used for comments). To run **only** a section of a file you can highlight the desired instructions with the mouse, and the press F9.

4.1 Instructions

- The first lines of the file import the different packages that the file uses: numpy, matplotlib, and the functions we have created for this activity.
- *Section 1*. This section generates the **nature** run of the experiment, i.e. what we consider to be the true system. You can change the initial conditions, the final time (consider that the model time step is 0.025 time units), and the initial guess from which the assimilation will start. For speed in computations and to display figures in an easier manner, we have selected $N_x = 12$ variables. This model can be run from some given initial conditions, but the default is to spin it up from a perturbation around the unstable fixed point of the system. You will get a Hovmoller diagram (a contour plot showing the time evolution of the different variables in a circle of latitude), as well as a figure with N_x panels. This section also defines the initial guess for the data assimilation
- *Section 2*. This section is related to the observations. You can select to observe different variables with three options: $'all'$ corresponds to observing all variables, $'1010'$ corresponds to observing every other variable, and $'landsea'$ corresponds to observing only half of the domain (a challenging setting). Different options will create a corresponding observation matrix. The **R** matrix is designed to be diagonal (common assumption), but you can choose the observational variance. You can also choose the observational period (in number of model steps). In this model the time auto-correlation is quite small, so we recommend to experiment with observational periods of no larger than 4 time steps.
- *Section 3*. This section contains the DA experiments using ETKF, SEnKF and L-SEnkF. You can vary the type of function used in the localisation: $cutoff$ for a window function and GC for Gaspari-Cohn (a compact-support approximation to a Gaussian). The parameter lam is the half-width of the localisation. The system displays a tileplot of the localisation matrix.

As usual, repeat the experiments using variations of the parameters.