Experiments with variational (ensemble-variational) methods.

Joint NCEO/ECMWF Intensive Course on Data Assimilation

University of Reading, 5th-8th March 2019

Preliminaries

In this exercise you will experiment with 2 types of hybrid methods:

- First, those that use a hybrid background error covariance $\mathbf{B}_h \in \mathcal{R}^{N_x \times N_x}$ in a traditional variational minimisation framework. Here you will use 4DVar-LETKF.
- Second, those that avoid computing tangent linear and adjoint models and instead use 4-dimensional (cross-time) ensemble covariances to communicate the impact of observations to the initital time. Here you will use SC-4DEnsemble Var (SC denotes strong constraint).

You will experiment with the Lorenz 1996 model with 20 variables. This size allows for relatively easy visualisation while requiring localisation of the ensemble covariances.

You have been given a folder containing python code. This folder is labeled *hybrids*. The only file you have to modify and run is *ControlHybrids.py*. This file has been divided in cells. You can highlight the instructions with the cursor and then press F9. Cells allow to run separate parts of the code without having to run the whole document.

To begin, run the cell at the top of the document to import some necessary packages.

1 Nature Run

The first part of the exercise is to generate our nature run (which we consider to be the truth). The model is perfect, i.e. no model error is considered.

Run cell 1 to create the nature run. Let us denote our state variable as $\mathbf{u} \in \mathcal{R}^{N_x}$. Recall that this model is described by the following differential equation:

$$\frac{du_n}{dt} = (u_{n+1} - u_{n-2})u_{n-1} - u_n + F \tag{1}$$

for $n = 1 \cdots N_x$, and where the indices are modular, i.e. $u_j = u_{mod(j,N_x)}$. The first term in the right hand side of this equation represents non-linear advection, the second term represents diffusion, and the third term represents forcing, with $F = 8$.

You can modify the following variables:
- N_x: number of variables in the model.
- t_{max}: maximum time. The time step of the model is $dt = 0.025$.

The code starts the model from 'the cold', spins up, gets rid of the transient, and plots the trajectory for the time span specified.

2 Observations

The second cell of the code deals with the generation of observations, which are simply:

$$\mathbf{y}^t = \mathbf{H}\mathbf{u}^{true} + \boldsymbol{\eta}^t \tag{2}$$

where $\mathbf{y}^t \in \mathcal{R}^{N_y}$, $\mathbf{H} \in \mathcal{R}^{N_y \times N_x}$ and $\boldsymbol{\eta}^t \in \mathcal{R}^{N_y}$. You can modify the following 3 parameters:
- *periodobs*: the observational period, in model time steps. Observations every 2 time steps (i.t. $\Delta_t = 0.05$ can be considered frequent (about 6 hours in a synoptic scale atmospheric process).
- *gridobs*: the observational grid. The options are: 1 (all gridpoints observed), 2 (observations in every other gridpoint), 3 (observations in half of the domain, also known as land/sea configuration), 4 (observations every 4 gridpoint).
- *stdobs*: standard deviation of the observational error. In the code the observations are considered uncorrelated.

3 Variational DA

Before using the hybrid DA methods, it would be useful to look at the performance of the variational and ensemble methods as a benchmark for comparison. Cell 3 creates a climatological background error covariance matrix $\mathbf{B}_c$ anc plots it. Is the decorrelation lengthscale of the model narrow or broad?

Cell 3.1. computes a 3DVar and a SC4DVar (both in preconditioned, incremental mode). For SC-4DVar you can modify the following variable:
- *obsperwin*: the number of observations per window. The total lenght of the assimilation window will be: $obsperwin \times freqobs$ model time steps.

The code plots the trajectories (both background and analysis) obtained by the different assimilation methods. The analysis RSME (root mean squared error) with respect to the truth is also plotted; this is done separately for observed and unobserved variables. Cell 3.2 compares the performance of 3DVar and SC4DVar.

4 Ensemble DA

Now we move into ensemble data assimilation. Cell 4 computes and plots the localisation matrix in both the state space ($\mathbf{L}_{xx}$) and in the mixed state/observation space ($\mathbf{L}_{xy}$). In this case you can modify the variables:
- *loctype*: 0 for a step function, 1 for Gasphari-Cohn function.
- *lam*: the localisation halfwidth. If you do not want any localisation, you should choose a large value.

The code has adaptive inflation implemented, so you do not have to worry about this parameter at all.

Cell 4.1. performs the actual assimilation using LETKF, and plots the resulting trajectories. In this cell you can modify the variable:
- M: the ensemble size.

Cell 4.2 compares the results obtained by the LETKF with those obtained by variational methods.

5 Exploring 3D covariances

One of the purposes of hybrid DA is to combine covariance information from a static yet full-rank source (the climatological $\mathbf{B}_c$) used in the VAR methods, with the flow-dependent yet low-rank information

coming from a sample of trajectories (the $\mathbf{P}^b(t)$ obtained by ensemble methods).

Cell 5 compares the climatological $\mathbf{B}_c$) with that obtained by the LETKF (computed from the background ensemble) at different times. The raw and localised versions are ploted for different times instants. In this case you can modify the variables:
- *nsample*: the number of instants in which you want to display the $\mathbf{P}_b(t)$.

6 Hybrid DA part 1

It is now time to start doing real hybrid DA. The first method we will try is 4DVar-LETKF. Recall that this method uses:

$$P_h^b = \beta_1 B_c + \beta_2 P^b \tag{3}$$

within a regular SC-4DVar cost function. Cell 6 contains the general settings to do this, it imports the necessary routines and it creates the localisation matrices needed. You can play with the following variables:
- *loch*: 0 if you do not want localisation in the ensemble part of the covariance, 1 if you do.

Cell 6.1 runs the assimilation, displays the trajectories obtained by this assimilation. You can modify the following variables:
- M: ensemble size.
- *obsperwin*: observations in the assimilation window.
- *beta*: the coefficients for the static (first number) and the dynamic (second number) part of the covariance matrix.

7 Exploring 4D covariances

Now we move into more complicated hybrid DA methods. We will use SC-4DEnVar. Remember that this method avoids using the tangent linear and adjoint models by computing 4-dimensional ensemble covariances. Let us start by comparing the error evolution coming from 2 sources: (a) evolving $\mathbf{B}_c$ using the tangent linear $\mathbf{M}^{0\to t}$ and adjoint $(\mathbf{M}^{0\to t})^{\mathbf{T}}$ models, and by evolving an ensemble run with different initial conditions (sampled from a normal distribution centered on the truth with covariance $\mathbf{B}_c$). This is done in cell 7.1. You can vary the parameters:
- M: ensemble size.
- lags: number of time steps for which you want to compute the covariance.

This cell will plot three rows of covariances. Can you tell what is being plotted in each row?

8 Hybrid DA part 2

The final section, found in cell 7.2, runs SC-4DEnVar and computes the analysis RMSE of this method with respect to the truth. In this case you have to generate the localisation matrix (with the same options as before), and you can vary the next variables:
- *lam*: the localisation half width.
- M: the number of ensemble members.
- *locenvar*: whether you want localisation or not.

We use a fixed (in time) localisation. Remember this can be problematic when localising cross-time covariances in long assimilation windows. Can you think of a way to test this?