Experiments with variational DA in L63 and L96

Joint NCEO/ECMWF Intensive Course on Data Assimilation

University of Reading, 5th-8th March 2019

1 Objective

This activity will allow the student to experiment using both 3D-Var and 4D-Var (strong-constraint) in some toy models. In particular, we will use the Lorenz '63 model, which has $N_x = 3$ variables, and the Lorenz '96 model with $N_x = 12$ variables.

2 Review of Theory

Variational data assimilation methods produce MAP (maximum-a-posteriori) estimators. They use optimisation techniques to minimise cost-functions, which are often quadratic forms. The cost-function can be interpreted as a sample estimator of the negative logarithm of the posterior pdf. The analysis state $\mathbf{x}^a \in \mathcal{R}^{N_x}$ is the minimizer of the cost-function, i.e. $\mathbf{x}^a = \operatorname{argmin} \mathcal{J}(\mathbf{x})$. This cost-function is different for 3D-Var and 4D-Var.

2.1 3D-Var

3D-Var assimilates observations at a given time point, i.e. there is no time component in the minimisation. The 3D-Var cost-function is:

$$\mathcal{J}(\mathbf{x}) = \frac{1}{2}\left(\mathbf{x}-\mathbf{x}^b\right)^{\mathbf{T}} \mathbf{B}^{-1}\left(\mathbf{x}-\mathbf{x}^b\right) + \frac{1}{2}\left(\mathbf{y}-h(\mathbf{x})\right)^{\mathbf{T}} \mathbf{R}^{-1}\left(\mathbf{y}-h(\mathbf{x})\right) \tag{1}$$

where $\mathbf{x}^b \in \mathcal{R}^{N_x}$ is the background estimate of the model state variables, $\mathbf{B} \in \mathcal{R}^{N_x \times N_x}$ is the background error covariance matrix, $\mathbf{y} \in \mathcal{R}^{N_y}$ is a vector of observations, $\mathbf{R} \in \mathcal{R}^{N_y \times N_y}$ is the observation error covariance matrix, and $h : \mathbf{R}^{N_x} \to \mathbf{R}^{N_y}$ is the observation operator. Here we apply the 3D-Var algorithm sequentially, which means that the model is evolved one step at a time and the observations are assimilated in order. Each time a new set of observations becomes available they are combined with the current model forecast (the background) and the cost function (1) is minimised, producing an updated analysis state. The model is then propagated forward to the time of the next observations, using the analysis as the initial state, and the assimilation process is repeated.

2.2 4D-Var

4D-Var assimilates observations distributed over a time window, also called the assimilation window. Hence, there is a time component in the minimisation. In the strong-constraint setting (which is the one we explore), the model is considered perfect (without errors) and the problem is reduced to one of finding the *optimal* initial state $\mathbf{x}_0^a \in \mathcal{R}^{N_x}$. For observations every δ model time-steps, and an assimilation window of length K observation times, the cost-function is:

$$\begin{aligned}\mathcal{J}(\mathbf{x}_0) = &\frac{1}{2}\left(\mathbf{x}_0 - \mathbf{x}_0^b\right)^{\mathbf{T}} \mathbf{B}^{-1}\left(\mathbf{x}_0 - \mathbf{x}_0^b\right) \\ &+ \frac{1}{2}\sum_{k=1}^{K}\left(\mathbf{y}_k - h_k\left(m_{0\to\delta.k}\left(\mathbf{x}_0\right)\right)\right)^{\mathbf{T}} \mathbf{R}_k^{-1}\left(\mathbf{y}_k - h_k\left(m_{0\to\delta.k}\left(\mathbf{x}_0\right)\right)\right)\end{aligned} \tag{2}$$

where $\mathbf{x}_0^b \in \mathcal{R}^{N_x}$ is the background estimate for the initial state at the start of the assimilation window, and $\mathbf{B} \in \mathcal{R}^{N_x \times N_x}$ is defined as in 3D-Var. The observation term in the cost-function becomes a sum, with one term for each observation time within the assimilation window. The k^{th} observation vector is $\mathbf{y}_k \in \mathcal{R}^{N_y}$ (we keep the size constant), $h_k : \mathcal{R}^{N_x} \to \mathcal{R}^{N_y}$ is the k^{th} observation operator, $\mathbf{R}_k \in \mathcal{R}^{N_y \times N_y}$ is the observation error covariance for the k^{th} observation vector, and the model operator $m_{0\to\delta.k} : \mathcal{R}^{N_x} \to \mathcal{R}^{N_y}$ evolves the initial state from the start of the assimilation window to the time of the k^{th} observation. In these experiments we will assume that $h_k = h$ and $\mathbf{R}_k = \mathbf{R}$ fixed.

3 The Lorenz 63 system

These are the python files used in this part of the activity:

- *ControlL63Var.py*. This is the control file - it is the one that you will be running and modifying.
- *L63model.py*. This file contains the instructions for running the L63 model.
- *L63misc.py*. This file generates different observation operators, creates the observations, and generates a simple background error covariance matrix.
- *L63var.py*. This file contains the routines to perform 3D-Var and strong constraint 4D-Var. This includes computing the tangent linear model and transition matrices.
- *L63plots.py*. This file contains the code for producing different output plots.

You will run different sections of the file *ControlL63Var.py*, as described in the next section. These are numbered as comments of the file (recall that in python # is used for comments). To run **only** a subsection of a file you can highlight the desired lines of code with the mouse, and then 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: The Nature Run***. This section generates the **nature** run for the experiment, i.e. what we consider to be the true system trajectory. You can change the initial conditions for the nature run, the final time (the model time step is fixed at 0.01 time units), and the initial background guess from which the assimilation will start. You can also play with the 3 parameters of the model to see how the behaviour of the system changes for different combinations of values. However, for the final experiment you should leave their values 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 of the 3 model variables.

- ***Section 2: The observations***. This section generates the observations, the observation error covariance matrix and the observation operator. You can select to observe different combinations of variables by changing the value of *'obsgrid'*: setting *obsgrid* $='xyz'$ will observe all variables, or you could choose a subset e.g. $'xz'$ or $'y'$. Different choices will create the appropriate observation operator. The **R** matrix is designed to be diagonal (common assumption), but you can choose the observation error variance. You can also choose the frequency of the observations (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 fully non-linear problem.

- ***Section 3: Data assimilation***. This short section creates the climatological **B** matrix for this model. There is a scaling tuning parameter that can be varied depending on the observation frequency.

- ***Section 3a: 3DVar***. This section runs the 3D-Var assimilation. It also computes the background and analysis RMSE (root mean squared error) with respect to the truth. The trajectories and the RMSE's are plotted.

- ***Section 3b: 4DVar***. This section runs the 4D-Var assimilation. You can select the length of the assimilation window, which is expressed in terms of the number of observation times per window. It also computes the background and analysis RMSE with respect to the truth. The trajectories and the RMSE's are plotted.

Run each of sections 1 - 3b and experiment with varying the following parameters: variables being observed, observation frequency, observation error variance, tuning factor for the **B** matrix, initial conditions for the nature run, initial background guess $\mathbf{x}_0^b$, the length of the model run and the length of the assimilation window (for 4DVar).

4 The Lorenz 96 system

These are the python files used in this part of the activity:

- *ControlL96Var.py*. This is the control file - it is the one that you will be running and modifying.

- *L96model.py*. This file contains the instructions for running the L96 model.

- *L96misc.py*. This file generates different observation operators, creates the observations, and generates a simple background error covariance matrix.

- *L96var.py*. This file contains the routines to perform 3D-Var and strong constraint 4D-Var. This includes computing the tangent linear model and transition matrices.

- *L96plots.py*. This file has instructions for producing different output plots.

You will run different sections of the file *ControlL96Var.py* as described in the net section. These are numbered 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 lines of code with the mouse, and then 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: The Nature Run***. This section generates the **nature** run of the experiment, i.e. what we consider to be the true system trajectory. You can change the initial conditions for the nature run, the final time (the model time step is fixed at 0.025 time units), and the background initial guess from which the assimilation will start. For speed of computations and to display figures in an easier manner, we have selected $N_x = 12$ variables. This model can be run from a given initial condition, 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 $N_x = 12$ panels.

- ***Section 2: The observations***. This section generates the observations, the observation error covariance matrix and the observation operator. 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 choices will create the appropriate observation error covariance matrix. The $\mathbf{R}$ matrix is designed to be diagonal (common assumption), but you can adjust the observation error variance. You can also choose the frequency of the observations (in number of model steps). In this model the time auto-correlation is quite small, so we recommend experimenting with observation frequencies no larger than 4 time steps.

- ***Section 3: Data assimilation***. This short section creates the climatological $\mathbf{B}$ matrix for this model. There is a scaling tuning parameter that can be varied depending on the observation frequency.

- ***Section 3a: 3DVar***. This section runs the 3D-Var assimilation. It also computes the background and analysis RMSE with respect to the truth. The trajectories and the RMSE's are plotted.

- ***Section 3b: 4DVar***. This section runs the 4D-Var assimilation. You can select the length of the assimilation window, which is expressed in terms of the number of observation times per window. It also computes the background and analysis RMSE with respect to the truth. The trajectories and the RMSE's are plotted.

Run each of sections 1 - 3b and experiment with varying the following parameters: variables being observed, observation frequency, observation error variance, tuning factor for the $\mathbf{B}$ matrix, initial background guess $\mathbf{x}_0^b$, the length of the model run and the length of the assimilation window (for 4D-Var).