# Experiments with variational DA in L63 and L96

ECMWF/NCEO data assimilation training course

March 2, 2018

## 1 Objective

This activity will allow the student to experiment using both 3DVar and 4DVar (strong-constraint) 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.

## 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 value $\mathbf{x}^a \in \mathcal{R}^{N_x}$ is the minimizer of a cost-function, i.e. $\mathbf{x}^a = \text{argmin}\mathcal{J}(\mathbf{x})$. This cost-function is different for 3DVar and 4DVar.

### 2.1 3DVar

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

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

where $\mathbf{x}^b \in \mathcal{R}^{N_x}$ is the background value of the 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 the 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.

### 2.2 4DVar

4DVar assimilates observations over a time window, also called assimilated 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 and the problem is reduced to finding *optimal* initial conditions $\mathbf{x}^{0,a} \in \mathcal{R}^{N_x}$. For observations every $\delta$ model-steps, the cost-function is:

$$\begin{aligned}\mathcal{J}\left(\mathbf{x}^0\right) = &\frac{1}{2}\left(\mathbf{x}^0-\mathbf{x}^{0,b}\right)^{\mathbf{T}}\mathbf{B}^{-1}\left(\mathbf{x}^0-\mathbf{x}^{0,b}\right) \\ &+\frac{1}{2}\sum_{l=1}^{L}\left(\mathbf{y}^l-h^l\left(m^{0\to\delta l}\left(\mathbf{x}^0\right)\right)\right)^{\mathbf{T}}\mathbf{R}^{-1}\left(\mathbf{y}^l-h^l\left(m^{0\to\delta l}\left(\mathbf{x}^0\right)\right)\right)\end{aligned} \tag{2}$$

where $\mathbf{x}^{0,b} \in \mathcal{R}^{N_x}$ is the background value for the initial conditions, and $\mathbf{B} \in \mathcal{R}^{N_x \times N_x}$ and $\mathbf{R} \in \mathcal{R}^{N_y \times N_y}$ are defined as in 3DVar. The observations term in the cost-function becomes a sum, one term for each observational time. The $l^{th}$ observation is $\mathbf{y}^l \in \mathcal{R}^{N_y}$ (we keep the size constant), $h^l : \mathcal{R}^{N_x} \to \mathcal{R}^{N_y}$ is the $l^{th}$ observation operator, and $m^{0 \to \delta l} : \mathcal{R}^{N_x} \to \mathcal{R}^{N_y}$ evolves the initial conditions to the time of the observations.

# 3 Instructions for Lorenz 63

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

- *ControlL63Var.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.
- *L63var.py*. This file contains the routines to perform 3DVar and SC4DVar. 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 *ControlL63Var.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.* This short section creates the climatological $\mathbf{B}$ matrix for this model. There is a scaling tuning parameter which can be varied depending on the observational frequency.

- *Section 3a.* This section runs the 3DVar 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.* This section runs the 4DVar assimilation. You can select the length of the assimilation window, which is expressed in number of observational times in the window. 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.

Repeat the previous steps and vary the following parameters: variables being observed, observational period, observational variance, tuning factor for the **B** matrix, initial guess for the data assimilation, length of the assimilation window (for 4DVar).

# 4 Instructions for Lorenz 96

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

- *ControlL96Var.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.

- *L96var.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 $\mathbf{R}$ 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). 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 short section creates the climatological $\mathbf{B}$ matrix for this model. There is a scaling tuning parameter which can be varied depending on the observational frequency.

- *Section 3a.* This section runs the 3DVar 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.* This section runs the 4DVar assimilation. You can select the length of the assimilation window, which is expressed in number of observational times in the window. 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.

Repeat the previous steps and vary the following parameters: variables being observed, observational period, observational variance, tuning factor for the $\mathbf{B}$ matrix, initial guess for the data assimilation, length of the assimilation window (for 4DVar).