

A data assimilation tutorial based on the Lorenz-95 system

Martin Leutbecher, ECMWF, May 5, 2004

1. Introduction

a. *The Lorenz-95 system*

We will explore different data assimilation schemes using a low-dimensional dynamical system introduced by Edward Lorenz in 1995. The system is given by

$$\frac{dx_i}{dt} = -x_{i-2}x_{i-1} + x_{i-1}x_{i+1} - x_i + F, \quad (1)$$

where $i = 1, 2 \dots 40$, and cyclic boundary conditions are used $x_0 = x_{40}$, $x_{-1} = x_{39}$, $x_{41} = x_1$. The magnitude of the forcing is set to $F = 8$. For this forcing the system is chaotic, i.e. it has positive Lyapunov exponents. Lorenz (1995) concluded that similar error growth characteristics to operational NWP systems are obtained if a time unit in the L95-system is associated with 5 days. This scaling will be used here, too. Solutions of the system are obtained by numerical integration with a fourth-order Runge-Kutta scheme using a 3-hour time-step ($\Delta t = 0.025$). For the chosen forcing, the system has 13 positive Lyapunov exponents, the largest corresponds to a doubling time of 2.1 days. The dynamics is the same for each variable as eqn. (1) is invariant under the transformation $i \rightarrow i + 1$. Variables fluctuate about the mean in a non-periodic manner with a climatological standard deviation of $\sigma_{\text{clim}} \equiv \text{sigma_clim} = 3.6$. A perturbation of the initial condition will grow with time and its leading edge propagates “eastward” (to higher indices) at a speed of about 25 degrees/day — this corresponds to a shift of 14 indices in a (non-dimensional) time unit. See Lorenz (1995) and Lorenz and Emanuel (1998) for a more detailed discussion of the system.

b. *How to start the programmes*

All programmes used in this tutorial have been coded in `scilab`. This is free software available from <http://scilabsoft.inria.fr> and is quite similar to the commercial software package `matlab` in many respects. Operations like matrix transpose and multiplications can be coded easily and elegantly.

In order to run a `scilab` programme you have to

1. create a directory for temporary data: `mkdir $SCRATCH/L95data`
2. start `scilab`: `cd Tutorial; scilab`
3. integrate the system by typing `exec('nonlin.sci',-1)` in the `scilab` window.
4. now, any of the other programmes can be run (`195demo.sci`, `di.sci`, `oi.sci`, `kf.sci`)

c. Illustration of Lorenz-95 dynamics

Explore the chaotic dynamics of the Lorenz-95 system (`195demo.sci`).

Issues to look at:

- “westward” (towards lower indices) phase velocity of “highs” and “lows” (Hovmueller plot of solution)
- “eastward” (that is to higher indices) propagation of perturbations (Hovmueller plot of ensemble spread and plumes at different locations). Use localized perturbations.
- growth of errors with forecast range (error growth curve and plumes).

2. Data assimilation experiments

The model has been run for many time steps (`nonlin.sci`) to obtain a simulated truth (`SCRATCH/L95data/traj040.truth.025rk4.r8`). Observations are constructed by taking the values from the truth run and adding noise (normal distribution with zero mean and specified standard deviation).

Two different observation networks are considered:

- **Observation network 1** has an observation at every site 1–40.
- **Observation network 2** has gaps of two unobserved sites every 5 sites. Thus, there are observations at sites 3, 4, 5, 8, 9, 10, ... 38, 39, 40; this amounts to $3 \times 40/5 = 24$ observations in total.

The observation error standard deviation is $\sigma_o = 0.15\sigma_{\text{clim}}$ in both networks.

Three data assimilation schemes can be run which are all of the form

$$\mathbf{x}^a = \mathbf{x}^b + \mathbf{K}(\mathbf{H}\mathbf{x}^b - \mathbf{y}), \quad (2)$$

where the gain matrix \mathbf{K} is given by

scheme	gain matrix
direct insertion	\mathbf{H}^T
optimum interpolation	$\mathbf{B}\mathbf{H}^T(\mathbf{R} + \mathbf{H}\mathbf{B}\mathbf{H}^T)^{-1}$
Kalman filter	$\mathbf{P}^f\mathbf{H}^T(\mathbf{R} + \mathbf{H}\mathbf{P}^f\mathbf{H}^T)^{-1}$

The covariance matrix \mathbf{B} for the optimum interpolation scheme is obtained from forecast minus truth differences and exploits the symmetry of the dynamics and observational network (Files `B6h1.r8` and `B6h2.r8` for networks 1 and 2, respectively). The covariance matrix is “tuned” for a 6-hourly observation frequency and observation errors with $\sigma_o = 0.15\sigma_{\text{clim}}$.

The evolution of the covariance matrix in the Kalman filter is described by an alternating sequence of forecast steps and analysis updates (see also Mike Fisher’s lecture on Kalman filters). The covariance update can be written as

$$(\mathbf{P}_j^a)^{-1} = (\mathbf{P}_j^f)^{-1} + \mathbf{H}_j^T \mathbf{R}_j^{-1} \mathbf{H}_j, \quad (3)$$

where the index j denotes operators at time t_j . The covariance forecast from time t_j to the next assimilation cycle at t_{j+1} is given by

$$\mathbf{P}_{j+1}^f = \mathbf{M}_j \mathbf{P}_j^a \mathbf{M}_j^T + \mathbf{Q}, \quad (4)$$

where \mathbf{M}_j denotes the propagator of the tangent-linear model from time t_j to time $t_{j+1} = t_j + \Delta t$. The matrix \mathbf{Q} represents a model error covariance term that increases the covariances (see issues under Kalman filter below).

a. Direct insertion — issues

- How sensitive is the forecast to the actual realization of the observational error? Start with a sample of 10 forecasts and try different seeds for the random number generator. (You can use the cursor-up-key ↑ to scroll through the history of commands in scilab).
- How many forecasts are required to estimate the rms error with an accuracy of 10% or 1%?
- Investigate how errors depend on the temporal frequency of observations.

b. Optimum interpolation — issues

- Start with the identity matrix as estimate for \mathbf{B} by setting `'file from which B is read'` to `none`. Compare errors with direct insertion.
- “Tune” the background error variances using a scaled identity matrix for \mathbf{B} . This can be done by keeping `'file from which B is read'` to `none` and setting `'sigma_b(DA-estimate)/ sigma_b'` to the desired value of σ_b .
- Now, rerun using a good estimate of \mathbf{B} by setting `'file from which B is read'` to `B6h1.r8` or `B6h2.r8` for observing network 1 and 2, respectively.
- Investigate the importance of having a good estimate of the correlations by
 - using the correct \mathbf{B} but removing all (or some) of the off-diagonal elements. Set `'localize bg-err covariances'` to `y` and set `'distance for cut-off'` to the distance at which the correlation is truncated to zero (1 corresponds to a diagonal \mathbf{B}).

– using a wrong correlation model and the correct variances by using the files `C6h1.r8` for observation network 1 and `C6h2.r8` for observation network 2 under `'file from which B is read'`.

- Investigate how the variance estimate affects the quality of the analysis and forecasts. Use again the correct \mathbf{B} and scale the matrix using a factor between 0.5 and 2 (under `'sigma_b(DA-estimate)/ sigma_b'`).
- Investigate the sensitivity to the estimate of observation error. Scale the DA-estimate of σ_o between 0.5 and 2 (under `'sigma_o(DA-estimate)/ sigma_o'`).
- Change the time interval between observations / assimilation cycles to 3 hours or 12 hours but keep the \mathbf{B} tuned for 6-hourly data (`B6h1.r8` or `B6h2.r8`). Can you scale \mathbf{B} to improve the scheme? Do you need to increase or decrease the variances for a longer/shorter time interval?

c. Extended Kalman Filter — issues

- Run the KF with default settings. How many cycles are required to spin up the system and obtain a good estimate of \mathbf{P}^f ?
- Examine the sensitivity to the specification of the model error term σ_q ? Can you cause the filter to diverge? Try to find an optimal σ_q .

Why is the performance best for nonzero model error term despite the fact that a perfect forecast model is used?

- Try multiplicative variance inflation by setting `'sigma_b(DA-estimate)/ sigma_b'` to a value larger than 1 and $\sigma_q = 0$. Can you find an optimal inflation factor?
- Explore the relevance of background error correlations in the KF by truncating \mathbf{P}^f . Set `'localize bg-err covariances'` to `y` and set `'distance for cut-off'` to the distance at which the correlation is truncated to zero (1 corresponds to a diagonal \mathbf{P}^f).

d. Intercomparison — issues

- Compare analysis errors and forecast errors for optimum interpolation with diagonal and full \mathbf{B} and the KF with diagonal and full \mathbf{P}^f .
- Do this for both observing networks. Does the relative performance (using the standard optimum interpolation with full \mathbf{B} as reference) of the different schemes depend on the observing network?

3. Outlook

Are you already addicted to playing with Lorenz-95? There are more aspects that did not fit in a short tutorial. You may want to modify the code yourself. Potentially interesting issues are

- model error
- more complicated observing networks, e.g. temporally varying observation coverage (see lecture on observation targeting)
- biased observations and bias estimation
- observational quality control
- variational data assimilation (tangent-linear and adjoint have already been coded, see `kf.sci`)
- comparison of different techniques for covariance estimation
- ensemble data assimilation algorithms

The installation of scilab should be straightforward on most hardware platforms. I welcome feedback on the tutorial and the scilab-code in general and bugs in particular — I buy you a pint of beer for each bug that you report back to me.

References

- Lorenz, E. N., 1995: Predictability: A problem partly solved. In *Seminar on Predictability*, volume Vol. I, ECMWF, Reading, UK, 1–18.
- Lorenz, E. N. and K. A. Emanuel, 1998: Optimal sites for supplementary weather observations: Simulation with a small model. *J. Atmos. Sci.*, **55**, 399–414.