A data assimilation tutorial based on the Lorenz-95 system

Martin Leutbecher, ECMWF, 17th April 2009

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

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

where $i = 1, 2 \ldots N$, and cyclic boundary conditions are used $x_0 = x_N,\ x_{-1} = x_{N-1},\ x_{N+1} = x_1$. 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 and $N = 40$, 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 $i \to i + 1$. Variables fluctuate about the mean in a non-periodic manner with a climatological standard deviation of $\sigma_{\text{clim}} \equiv$ `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), Lorenz and Emanuel (1998) and Lorenz (2005) for a more detailed discussion of the system.

b. How to start the programs

All programs used in this tutorial have been coded in `scilab`. This is free software available from `www.scilab.org` and is quite similar to the commercial software package matlab in many respects. Operations like matrix transponse and multiplications can be coded easily and elegantly. The main Kalman filter code is given by the following 7 lines of code:

```
[xf,q1,q2,q3]=c_traj(xa0,dt,ndt_assim,'rk4')  // background and trajectory
M   = tl_propagator(dt,xf,'rk4', q1, q2, q3)  // propagator
Pf1 = M * Pa0 * M' + Q                         // propagate covariances
K   = Pf1*H'*inv( R + H * Pf1 * H')            // Kalman gain
d   = y - H*xf1;                               // obs minus background departure
xa1 = xf1 + K*d                                // analysis
Pa1 = (eye(ndim,ndim) - K*H) * Pf1             // covariance update
```

An appendix describes how to install the scilab programs elsewhere.

c. Illustration of Lorenz-95 dynamics

The chaotic dynamics of the Lorenz-95 system can be explored by running (`l95demo.sci`). It displays an unperturbed solution and an ensemble in which the initial conditions have been perturbed (either locally at one site or globally at all sites).

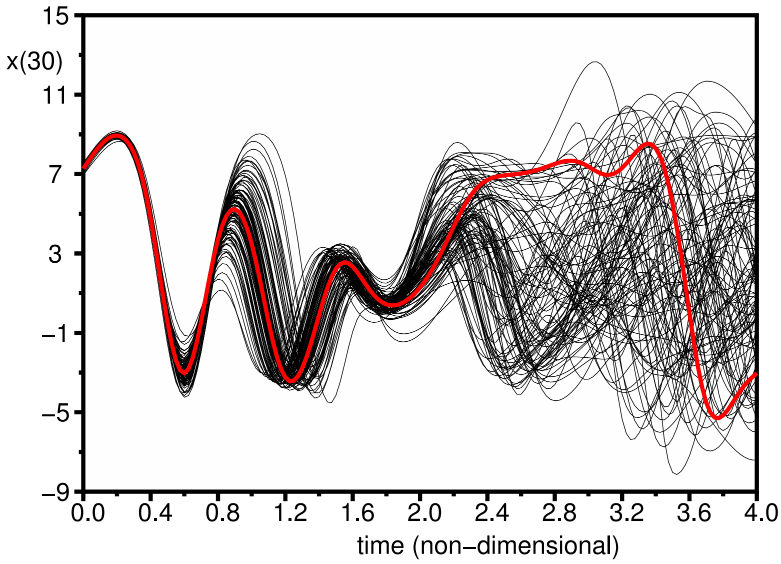

Figure 1: Evolution of the 40-variable Lorenz-95 system at site 30. Red solid line: unperturbed forecast; Black lines: ensemble of 100 forecasts which start from slightly perturbed initial conditions. The unperturbed initial conditions are an arbitralily selected state from a long integration of Eq. (1), i.e. a state on or close to the system attractor.

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).

a. Single observation experiments with the Kalman filter

Programme `kf1.sci` is designed to study single observation experiments using an extended Kalman filter. It works well for 8 and 12 dimensional systems. For the 16-dimensional system, the Kalman filter diverges.

b. Observing networks for the 40-dimensional system

Different assimilation systems can be compared for the 40-dimensional system.

Three 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.
- **Observation network 3** has gaps of 4 unobserved sites every 5 sites. There are observations at sites 1, 6, 11, ... 36; this amounts to 8 observations in total.

The observation error standard deviation is $\sigma_o = 0.15\,\sigma_{\text{clim}}$ in networks 1 and 2. For network 3, the error is three times smaller $\sigma_o = 0.05\,\sigma_{\text{clim}}$.

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

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

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

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

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

$$\left(\mathbf{P}^a_j\right)^{-1} = \left(\mathbf{P}^f_j\right)^{-1} + \mathbf{H}_j^{\mathrm{T}}\mathbf{R}_j^{-1}\mathbf{H}_j, \tag{3}$$

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

$$\mathbf{P}^f_{j+1} = \mathbf{M}_j\,\mathbf{P}^a_j\,\mathbf{M}_j^{\mathrm{T}} + \mathbf{Q}, \tag{4}$$

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 slides: Kalman filter exercises). The model error covariance is represented by σ_q^2 times the identity matrix.

3. Outlook

Did you enjoy 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`), see also Fisher et al. (2005)
- comparison of different techniques for background error covariance estimation
- ensemble data assimilation algorithms
- ensemble prediction; see tutorial in the predictability, diagnostics and seasonal forecasting training course and Leutbecher et al. (2007).

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.

References

Fisher, M., M. Leutbecher, and G. A. Kelly, 2005: On the equivalence between Kalman smoothing and weak-constraint four-dimensional variational data assimilation. *Quart. J. Roy. Meteor. Soc.*, **131**, 3235–3246.

Leutbecher, M., R. Buizza, and L. Isaksen, 2007: Ensemble forecasting and flow-dependent estimates of initial uncertainty. In *Flow-dependent aspects of data assimilation*, Workshop Proceedings, ECMWF, Shinfield Park, Reading, UK, 185–201. Available at www.ecmwf.int/publications/library/ecpublications/_pdf/workshop/2007/Data_assimilation.

Lorenz, E. N., 1995: Predictability: A problem partly solved. In *Seminar on Predictability*, volume Vol. I, ECMWF, Reading, UK, 1–18. Available at http://www.ecmwf.int/publications/library/ecpublications/_pdf/seminar/1995.

Lorenz, E. N., 2005: Designing chaotic models. *J. Atmos. Sci.*, **62**, 1574–1587.

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.

A How to install the tutorial

You are welcome to use this tutorial for educational purposes as long as you acknowledge its origin. The installation of the tutorial should be straightforward.

1. Install scilab (version 4.1 or higher is recommended). Scilab is free software available from `www.scilab.org`. Scilab is quite similar to the commercial software package matlab.

2. Obtain a copy of the tutorial software (distributed as gzipped tar-file `DA-TC.tar.gz`). You can e-mail me (M. Leutbecher "at" ecmwf.int) to receive a copy of the programs in case they are not included in the training course CD-ROM.

3. Unpack the archive `DA-TC.tar.gz` in a suitable directory

4. Create the data used by the tutorial.

   - On UNIX architectures:

     ```
     cd DA-TC/sci
     make
     ```

   - On Windows: start scilab and execute the scilab programs called in `DA-TC/sci/makefile`

5. start the tutorial:

   - On UNIX, execute `DA-TC/bin/tutorial`;
   - On Windows, start scilab, change directory to `DA-TC/sci`, execute `init.sci`