A Square-Root Ensemble Kalman Filter Demonstration with the Lorenz model

February 21, 2018

The HTML version of this documentation is located here

www.met.reading.ac.uk/∼darc/training/lorenz_ensrkf

and a PDF version is located here

www.met.reading.ac.uk/∼darc/training/lorenz_ensrkf/lorenz_ensrkf.pdf

This document contains the following:

Section	Description
1	A brief introduction to data assimilation.
2	A description of a data assimilation method called the ensemble square-root Kalman Filter.
3	A description of the model that this assimilation method is applied to: the (non-linear and chaotic) Lorenz 63 model.
4	Instructions for the demonstration web package.
5	Instructions for downloading the source code.
6	A worksheet of suggested experiments.
7	Appendix deriving the ensemble square-root Kalman Filter

Revision: 21/02/18, Eq. (9) corrected.

1 Data assimilation

Forecasts from models provide useful information to help predict the future state of a system. Inevitably though predictions will diverge from reality as time progresses. This is an inescapable property of dynamical systems. Data assimilation (DA) is a formal approach used to help correct forecasts by introducing information observed from the environment. Many DA methods are based on the 'Kalman Filter' (KF) equations[1]. The KF equations describe how information from forecasts and from observations should be combined in an optimal way which extracts maximum information from each source of information. The KF equations are usually solved in a cyclic fashion: a forecast starts from initial conditions at time t_0 and is run to time t_1. This forecast accumulates errors over this period which are reduced by assimilating observations at t_1. In the language of DA, the assimilated forecast/observation product is called the 'analysis state'. The analysis state at t_1 is then used as the initial conditions for another forecast to time t_2 which is then combined with the next batch of observations at that time[1].

The KF equations rely heavily on quantitative information about the respective uncertainties of the forecast and observational data which influence how the data are combined (for instance more accurate data are considered with greater importance than less accurate data). The KF equations describe not only the analysis state, but also its uncertainty (for instance the KF equation reflect the fact that analysis uncertainty

[1] The "filter" terminology is adopted from signal processing. It refers to the timing of the observations, which are made at the same time as the forecast is valid (or before). The equations perform a 'filter' operation in the sense that they reduce uncertainty in a signal (in effect 'filtering-away' errors). Kalman [1] is the author of the paper who introduced the equations that are still at the heart of DA today.

will be less than the forecast uncertainty due to the information provided by the observations). They also describe how this uncertainty evolves along with the forecast to the time of the next batch of observations.

The KF equations provide a means of quantifying all of this information in an explicit way, but the use of the KF can become prohibitively expensive for systems requiring a large number of pieces of information to describe. Suppose a system requires n variables to describe its state (i.e. for a forecast or analysis). The KF then needs $\sim \mathcal{O}(n^2)$ pieces of information to describe the uncertainty of the state (uncertainties are described by error covariance matrices which represent Gaussian-shaped probability density functions). In numerical weather prediction, the number of variables is $n \sim \mathcal{O}(10^7)$ (i.e. the number of pieces of information needed to describe meteorological fields on a grid covering all relevant parts of the Earth's atmosphere). Although present-day technology can handle states with this number of variables, it cannot handle the error covariance matrices, which require $n^2 \sim \mathcal{O}(10^{14})$ pieces of information. This motivates the need for approximate (but efficient) ways of dealing with large n.

2 The Ensemble Square-Root Kalman Filter

The impracticality of the KF equations in large systems has lead to the development of approximate forms. One important development lies with ensemble methods. Such methods solve a reduced form of the KF equations which deal with not one forecast/analysis state, but with an ensemble of N forecast/analysis states. Ensemble methods describe a state's uncertainty by the spread in the ensemble at a given time instead of using error covariance matrices. These methods have proved to be practical when $N \ll n$ (e.g. $N \sim \mathcal{O}(10^2)$ for $n \sim \mathcal{O}(10^7)$), although they do introduce a range of new problems associated with the approximations.

There are many flavours of ensemble KFs. The ensemble square-root Kalman Filter (EnSRKF) is one popular variant. This method of DA has the following characteristics:

- The EnSRKF uses an ensemble of forecasts (output from a numerical model) to represent uncertainty in the model's ability to capture reality. The quantification of forecast uncertainty is essential for DA as it informs how well information from the forecast should be trusted when confronting observational information. The filter works by constructing an ensemble of analyses which is designed to have a spread equivalent to that predicted by the ordinary KF.

- The EnSRKF is a sequential method. This means that it chops up time into a number of intervals. At times where no observations are available, the method propagates the ensemble forward in time using the model equations (model error may be added during the propagation if model error is to be represented). At times where there are observations, the method combines the forecast ensemble with the observations to create a new ensemble which is consistent with all pieces of information.

- The EnSRKF combines the forecast ensemble with the observation(s) using an update equation. This update equation is derived from the ordinary KF equations.

- The "square-root" part of the method's name applies to the particular flavour of ensemble DA that is used here, which uses the forecast/analysis ensemble perturbations (from their mean) as a 'square-root' of the forecast/analysis error covariance matrices. The square-root formulation (unlike non-square-root formulations) does not require the observations to be perturbed during the assimilation procedure.

The equations for the EnSRKF are derived in the appendix for interested readers, but are summarized here (shown for a general non-specific system). At time t, let the forecast state of the system be represented by an n-element state vector, $\mathbf{x}_k^{\mathrm{f}}(t)$, and suppose that an ensemble of N such model states exist ($1 \leq k \leq N$), and let these state vectors comprise the columns of the matrix $\mathbf{A}^{\mathrm{f}}$ (an $n \times N$ matrix) as follows:

$$\mathbf{A}^{\mathrm{f}}(t) = \left(\mathbf{x}_1^{\mathrm{f}}(t), \mathbf{x}_2^{\mathrm{f}}(t), \cdots, \mathbf{x}_N^{\mathrm{f}}(t)\right). \tag{1}$$

As long as the ensemble is distributed correctly, this forecast ensemble will represent possible realizations of the real system. The spread (disagreement between) members represents the uncertainty of the forecasts.

2.1 Time steps where observations are available

Suppose that observations are available at time t, and let these be assembled in the p-element vector $\mathbf{y}(t)$ (for p observations). The EnSRKF may be used to update the members in such a way as to reduce the spread of the ensemble consistent with the uncertainties of the forecasts and the observations (like models, even observations are not perfect representations of reality).

Now let $\overline{\mathbf{x}^{\mathrm{f}}}(t)$ be the n-element vector which is the mean of the ensemble members contained in $\mathbf{A}^{\mathrm{f}}$. The first stage of the EnSRKF is to find a new vector, $\overline{\mathbf{x}^{\mathrm{a}}}(t)$, which represents a mean analysis state which is closer to the observations than the forecast state is (the superscript "a" stands for "analysis"). The formula for $\overline{\mathbf{x}^{\mathrm{a}}}(t)$ is:

$$\overline{\mathbf{x}^{\mathrm{a}}} = \overline{\mathbf{x}^{\mathrm{f}}} + \mathbf{A}^{\mathrm{f}'}\mathbf{S}^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{y} - \mathbf{H}\overline{\mathbf{x}^{\mathrm{f}}}), \tag{2}$$

where the time labels have been removed for convenience. The new symbols that appear in (2) are as follows. $\mathbf{A}^{\mathrm{f}'}$ is the $n \times N$ matrix of forecast perturbations from the forecast mean state, ie

$$\mathbf{A}^{\mathrm{f}'} = \left(\mathbf{x}_1^{\mathrm{f}} - \overline{\mathbf{x}^{\mathrm{f}}}, \mathbf{x}_2^{\mathrm{f}} - \overline{\mathbf{x}^{\mathrm{f}}}, \cdots, \mathbf{x}_N^{\mathrm{f}} - \overline{\mathbf{x}^{\mathrm{f}}}\right), \tag{3}$$

$\mathbf{H}$ is the $p \times n$ observation operator matrix ($\mathbf{H}$ acts on a model state and outputs the model's version of those observations that are consistent with the input model state), $\mathbf{S}$ is the $p \times N$ matrix product

$$\mathbf{S} = \mathbf{H}\mathbf{A}^{\mathrm{f}'}, \tag{4}$$

and $\mathbf{C}$ is the $p \times p$ matrix

$$\begin{aligned} \mathbf{C} &= \mathbf{S}\mathbf{S}^{\mathrm{T}} + (N-1)\mathbf{R}, & (5)\\ &= \mathbf{Z}\mathbf{\Lambda}\mathbf{Z}^{\mathrm{T}}, & (6) \end{aligned}$$

In (5) $\mathbf{R}$ is the $p \times p$ observation error covariance matrix, which describes the uncertainty of the observational data. In (6), the matrix $\mathbf{C}$ has been decomposed into its eigenvectors, $\mathbf{Z}$, and eigenvalues, $\mathbf{\Lambda}$, which will be used below.

Equation (2) says how the mean of the ensemble should be updated given the new observations. The following formula specifies how the whole ensemble should be updated. This is found from the $n \times N$ matrix of analysis perturbations, $\mathbf{A}^{\mathrm{a}'}$:

$$\mathbf{A}^{\mathrm{a}'} = \mathbf{A}^{\mathrm{f}'}\mathbf{V}(\mathbf{I} - \mathbf{\Sigma}^{\mathrm{T}}\mathbf{\Sigma})^{1/2}\mathbf{V}^{\mathrm{T}}. \tag{7}$$

The new symbols that appear in (7) are found as follows. First define the $p \times N$ matrix $\mathbf{X}$:

$$\mathbf{X} = \mathbf{\Lambda}^{-1/2}\mathbf{Z}^{\mathrm{T}}\mathbf{S}. \tag{8}$$

The $N \times N$ matrix $\mathbf{X}^{\mathbf{T}}\mathbf{X}$ is the following, which is also given in its eigen-decomposition, with eigenvectors, $\mathbf{V}$, and eigenvalues ($\mathbf{\Sigma}^{\mathrm{T}}\mathbf{\Sigma}$):

$$\mathbf{X}^{\mathbf{T}}\mathbf{X} = \mathbf{S}^{\mathrm{T}}\mathbf{Z}\mathbf{\Lambda}^{-1}\mathbf{Z}^{\mathrm{T}}\mathbf{S} = \mathbf{V}\mathbf{\Sigma}^{\mathrm{T}}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}, \tag{9}$$

(the reason for writing the eigenvalues in a composite form $\mathbf{\Sigma}^{\mathrm{T}}\mathbf{\Sigma}$ - rather than with a single symbol - is to allow the matrix $\mathbf{\Sigma}$ to be identified as the matrix of singular values of $\mathbf{X}$; this will be important to some readers, but is not essential to the understanding of the method). This completes the definition of all of the symbols in (7). It is useful to note that (7) may be written in the following form

$$\mathbf{A}^{\mathrm{a}'} = \mathbf{A}^{\mathrm{f}'}\mathbf{T}, \tag{10}$$

where $\mathbf{T} = \mathbf{V}(\mathbf{I} - \mathbf{\Sigma}^{\mathrm{T}}\mathbf{\Sigma})^{1/2}\mathbf{V}^{\mathrm{T}}$. This is useful because $\mathbf{T}$ may be interpreted as the $N \times N$ matrix of weighting coefficients (in the sense that the kth analysis perturbation may be regarded as a linear combination of the N forecast perturbations, whose weighting coefficients are specified in the kth column of $\mathbf{T}$).

The final step is to construct the full ensemble from the perturbations and then to propagate this ensemble to the next time step (to give the forecast ensemble for the next time step):

$$\begin{aligned} \mathbf{A}^{\mathrm{a}}(t) &= (\overline{\mathbf{x}^{\mathrm{a}}}, \overline{\mathbf{x}^{\mathrm{a}}}, \cdots, \overline{\mathbf{x}^{\mathrm{a}}}) + \mathbf{A}^{\mathrm{a}'}, & (11)\\ \mathbf{A}^{\mathrm{f}}(t + \Delta t) &= \mathcal{M}(\mathbf{A}^{\mathrm{a}}(t)) + \delta\mathbf{A}(t), & (12) \end{aligned}$$

where $(\overline{\mathbf{A}^{\mathrm{a}}}, \overline{\mathbf{A}^{\mathrm{a}}}, \cdots, \overline{\mathbf{A}^{\mathrm{a}}})$ is the $n \times N$ matrix of identical mean analysis states in each column, $\mathcal{M}$ is the (potentially) non-linear forecast model (acting on each column of its matrix argument separately), and $\delta\mathbf{A}$ is an $n \times N$ matrix of stochastic error perturbations (to simulate an imperfect model). The stochastic perturbations have specified error covariance $\mathbf{Q}$.

2.2 Time steps where observations are not available

In the absence of observational information, no data assimilation can be performed and so only the forecast is performed:

$$\mathbf{A}^{\mathrm{f}}(t+\Delta t) = \mathcal{M}(\mathbf{A}^{\mathrm{f}}(t)) + \delta\mathbf{A}(t), \tag{13}$$

(note that in (13) the initial conditions are the forecast states at time t, rather than the analysis states in (12)).

3 The model that the EnSRKF will be used with

The EnSRKF will be used with observations to update a run of the Lorenz 63 model (with stochastic forcing if specified). The Lorenz 63 model propagates a state of three variables ($n = 3$, x, y and z) according to the following coupled, non-linear ordinary differential equations:

$$\frac{\partial x}{\partial t} = -\sigma(x-y), \tag{14}$$

$$\frac{\partial y}{\partial t} = x(\rho - z) - y, \tag{15}$$

$$\frac{\partial z}{\partial t} = xy - \beta z, \tag{16}$$

where $\sigma = 10$, $\rho = 28$ and $\beta = 8/3$ are fixed parameters. In the demonstration package, the Lorenz 63 equations are solved using the fourth-order Runge-Kutta algorithm e.g.[3] and stochastic model error terms are added if required.

Since n is small for this model, it is actually practical to apply the ordinary KF. We examine the EnSRKF here, not for reasons of constructing a reduced representation of this system, but instead just to demonstrate the workings of the method.

4 The EnSRKF/Lorenz 63 web demonstration

The web interface is located at the web address:
www.met.reading.ac.uk/∼darc/training/lorenz_ensrkf/lorenz_ensrkf.html.
This interface allows the reader to run the EnSRKF without the need to be concerned with source code or plotting software. If preferred, the source code may be downloaded - see Sec. 4 (5). The following parameters may be adjusted via the web interface:

Parameter(s)	Description	Default value(s)	Notes
δt	The time step of Lorenz model numerics.	0.01	Default is a good value to use.
N	The number of ensemble members.	6	$2 \leq N \leq 25$.
$\mathbf{x}^{\mathrm{truth}}(0)$	The initial conditions (x, y, z) of the true state.	3.0, -3.0, 12.0	Used as a reference run of the model to compare to the analyses and to generate synthetic observations.
$\mathbf{B}^{1/2}$	The initial prior standard deviations of x, y, z.	1.0, 1.0, 1.0	To set-up the starting points of the ensemble.
model error?	Switch to turn on/off model error.	off	Used to simulate an imperfect model by periodically adding stochastic noise with the error covariance $\mathbf{Q}$.

Parameter(s)	Description	Default value(s)	Notes
$\mathbf{Q}^{1/2}$	The model error standard deviations of x, y, z.	4.0, 4.0, 4.0	Relevant only if model error is switched on.
observe x, y, z?	Switches to turn on/off observations of variables.	all on	Must have at least one variable observed.
$\mathbf{R}^{1/2}$	Observation error standard deviations.	1.0, 1.0, 1.0	$\mathbf{R}$ is used in the EnSRKF algorithm. It is also used to specify the covariance of the observation noise used to generate synthetic observations.
Δt^{assim}	The number of time-steps in the assimilation period.	200	Observations are spread throughout the first Δt^{assim} time-steps.
Δt^{fore}	The number of time-steps in forecast period.	400	The last Δt^{fore}time-steps are free forecasts.
$N^{\text{obsbatches}}$	The number of batches of observations (each batch observing$x$, y, z as specified above) spread uniformly throughout the Δt^{assim} time steps.	5	The code might adjust this parameter slightly if $\Delta t^{\text{assim}}/N^{\text{obsbatches}}$ is non-integer.
seed	The random number seed.	123456	Change this to rerun with different random numbers.

Some notes on these parameters:

- The model is run over $\Delta t^{\text{assim}} + \Delta t^{\text{fore}}$ time-steps in total.
- Over the first Δt^{assim} time-steps, the ensemble members are influenced by $N^{\text{obsbatches}}$ batches of observations.
- Over the next (and last) Δt^{fore} time-steps, the ensemble members are pure forecasts.
- Observations are synthesized from the true trajectory by adding observation noise. This is the 'identical twin experiment' method.
- The initial ensemble points are synthesized from the true initial condition by adding background noise.
- Random numbers are used to simulate errors in the observations, to simulate errors in the initial conditions of the ensemble members and to simulate model errors when propagating the state with the Lorenz model.

5 Downloading the source code for the EnSRKF

For readers who prefer to work with source code, there are two versions of the EnSRKF demo for the Lorenz 63 model.

- The C++ source code may be downloaded from this location:
  www.met.reading.ac.uk/~darc/training/lorenz_ensrkf/lorenz_ensrkf.cpp.
  In order for this code to work without the need for additional mathematical libraries, this code includes an eigen-routine. Note that this routine has not the most efficient available and has not been optimized.
- A similar (but not identical) version written for the IDL language, with its own documentation, may be downloaded from this location:
  www.nceo.ac.uk/lorenz_enkf.php.

Please let us know if you download the code. We ask that the code should be used for non-profit uses only.

6 Worksheet of suggested experiments with the EnSRKF

The following tasks are suggested to help you understand the data assimilation method (and the EnSRKF in particular). The assimilation is controlled via the web page
www.met.reading.ac.uk/~darc/training/lorenz_ensrkf/lorenz_ensrkf.html
and the settings are described in Sec. 4. Helpful information is provided also on the web page just mentioned.

1. The first run is to familiarise you with the output from the web interface. Check that all settings are at their default values as specified in the table in Sec. 4 (if they are not, then click on the "`Reset to default values`" button on the web page).

   (a) Click on the "`Run Ensemble Square-root Kalman Filter`" button to start the assimilation. A page of data will appear followed by four graphs.

   (b) The left-hand column of data is a summary of the settings used and a key to the colours used in the graphs. The right-hand column shows the details of the observations that have been simulated. These observations have been derived from the truth run (shown alongside the observations) with added observation noise.

   (c) The first three graphs are plots of the x, y and z variables as a function of time. The blue line is the true trajectory from which the noisy observations and noisy background state are simulated. This is the trajectory that we would like the data assimilation method to follow. The red line is the ensemble-mean forecast/analysis trajectory and the grey bars are its one-standard-deviation error bars (found from the ensemble spread). The green dots are the observations and are shown with their one-standard-deviation error bars. What do you notice about the error growth during the forecasts, and when a forecast is confronted by an observation?

   (d) The last plot is the truth (blue) and ensemble mean (red) trajectories in phase space.

   (e) Hint: To help compare different outputs, use the tabs feature that is available with most web browsers: e.g. for Microsoft and Linux, click with the right mouse button on "`Return to settings page`" button and choose "`Open link in new tab`".

2. Return to the settings page. Repeat this run, but this time make each observation less accurate, e.g. increase each observation error standard deviation to three times larger than the default values. How does this change affect the ensemble mean trajectory and its error bars? How long are the forecasts of x, y and z (after the last observation is assimilated) useful representations of the truth?

3. Suppose that the default settings (as in 1) are used, except this time observations of x are absent. Would you expect the x trajectory to be unaffected by observations? Would you expect forecasts of x in this system to be in complete disagreement with the true x? Run the experiment and explain what you find. Try the same by returning to the settings in 1 but now removing y and then z observations.

4. Rerun the setting that you explored in 3, but this time double the number of measurement times to 10.

5. Return to the default settings (as in 1), but now use only 2 ensemble members. Does the assimilation work, given that the observations do not sample the signal adequately? Try increasing the number of measurement times to 6, 7, 8, 9, then 10.

6. The experiments so far have used a perfect model (that is the model used in the data assimilation is the same as that used to generate the truth run, even though the initial conditions differ in each case). Return to the default settings again, but now switch on model error (the Lorenz equations have no model error added for the truth run - only for the model used to propagate the ensemble members in the DA). How does the model error affect the quality of the assimilation? Repeat with error standard deviation 16 units for each variable: does increasing the number of observation times ever make up for the fact that the assimilation model is not perfect?

7 Appendix: derivation of the EnSRKF equations

The formulas shown in Sec. 2 for the ensemble square-root Kalman Filter (EnSRKF) are derived for interested readers in this appendix.

7.1 The basic Kalman filter

The starting point is the statement of the ordinary Kalman Filter (KF) equations. These describe how a single, but imperfectly known state evolves in time. The state evolution is governed by a (linear and imperfect) model, which is partially corrected by the assimilation of (imperfect) observational information. The KF equations are:

$$\begin{aligned}
\mathbf{x}^{\mathrm{a}} &= \mathbf{x}^{\mathrm{f}} + \mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}^{\mathrm{f}}), && (17)\\
\mathbf{P}^{\mathrm{a}} &= (\mathbf{P}^{\mathrm{f}-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H})^{-1} = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}^{\mathrm{f}}, && (18)\\
\mathbf{K} &= \mathbf{P}^{\mathrm{f}}\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{P}^{\mathrm{f}}\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1}, && (19)\\
\mathbf{x}^{\mathrm{f}}(t + \Delta t) &= \mathcal{M}(\mathbf{x}^{\mathrm{a}}) + \delta\mathbf{x}, && (20)\\
\mathbf{P}^{\mathrm{f}}(t + \Delta t) &= \mathbf{M}\mathbf{P}^{\mathrm{a}}\mathbf{M}^{\mathrm{T}} + \mathbf{Q}. && (21)
\end{aligned}$$

The symbols have the following meanings (all symbols refer to time t unless explicitly specified otherwise):

- $\mathbf{x}^{\mathrm{a}}$ is the analysis state at time t. This represents the best estimate of the system's state after observations have been considered. It is sometimes called the posterior state. There are n pieces of information in $\mathbf{x}^{\mathrm{a}}$, so $\mathbf{x}^{\mathrm{a}}$ is an n-element (column) vector.
- $\mathbf{x}^{\mathrm{f}}$ is the forecast state at time t. It is a forecast from a previous analysis and is an n-element vector. It is sometimes called the prior state (or background state) because it represents the estimate of the system's state before observations have been considered.
- $\mathbf{y}$ represents the observational information at time t. There are p pieces of information in $\mathbf{y}$, so $\mathbf{y}$ is a p-element vector.
- $\mathbf{H}$ is the (linear) observation operator matrix at time t. It acts on $\mathbf{x}^{\mathrm{f}}$ and gives the model's version of the observations. $\mathbf{H}$ is a $p \times n$ matrix.
- $\mathbf{K}$ is the Kalman gain matrix at time t. It acts on the difference between the actual observations and the model's observations (this difference is called an innovation vector) and gives the analysis increment vector which is added to the forecast to give the analysis. $\mathbf{K}$ is an $n \times p$ matrix and is made up of a string of other matrices.
- $\mathbf{P}^{\mathrm{f}}$ is the forecast error covariance matrix at time t. It describes the uncertainty of the forecast state. Suppose $\delta\mathbf{x}^{\mathrm{f}}$ is one possible error in $\mathbf{x}^{\mathrm{f}}$ at time t, then $\mathbf{P}^{\mathrm{f}} = \langle \delta\mathbf{x}^{\mathrm{f}} \delta\mathbf{x}^{\mathrm{f}\mathrm{T}} \rangle$, where the angled bracket indicates average over all possible $\delta\mathbf{x}^{\mathrm{f}}$ that are consistent with information available about $\mathbf{x}^{\mathrm{f}}$. $\mathbf{P}^{\mathrm{f}}$ is an $n \times n$ matrix.
- $\mathbf{R}$ is the observation error covariance matrix at time t. It describes the uncertainty of the observations. $\mathbf{R}$ is a $p \times p$ matrix.
- $\mathcal{M}$ is the forecast model (propagating its n-element argument at time t to an n-element argument at time $t + \Delta t$. $\mathcal{M}$ can be non-linear, but the KF equations are designed to be optimal when the forecast model is linear. $\delta\mathbf{x}$ is a stochastic random noise vector to represent the fact that the forecast model is imperfect.
- $\mathbf{P}^{\mathrm{a}}$ is the analysis error covariance matrix at time t. It describes the uncertainty of the analysis state. It has been written in two forms in (18). The second form, $(\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}^{\mathrm{f}}$, indicates that the analysis error covariances are necessarily smaller than the forecast error covariances (by the presence of the minus sign).
- $\mathbf{M}$ is the Jacobian of $\mathcal{M}$. It is an $n \times n$ matrix and is found from $\mathbf{M} = \partial\mathcal{M}/\partial\mathbf{x}$ evaluated on the model trajectory from the analysis state at time t to the next analysis time at $t + \Delta t$.

The KF works in a cycle. A forecast, $\mathbf{x}^{\mathrm{f}}$ (with error covariance $\mathbf{P}^{\mathrm{f}}$) at time t and observations, $\mathbf{y}$ (with error covariance $\mathbf{R}$) are combined in (17)to give the analysis, $\mathbf{x}^{\mathrm{a}}$. In a similar way the forecast and observation error covariances ($\mathbf{P}^{\mathrm{f}}$ and $\mathbf{R}$ respectively) are combined to give the analysis error covariance, $\mathbf{P}^{\mathrm{a}}$ in (18). The analysis is propagated to the next observation time at $t + \Delta t$ in (20) where it becomes the new forecast state,

and the analysis error covariance is propagated in (21) where it becomes the new forecast error covariance matrix. This forecast/update cycle is repeated. At no stage in this cycle is the state of the system known exactly: the forecast stage introduces inevitable forecast errors and the analysis step reduces them - but never to zero.

7.2 The ensemble square-root Kalman filter

The KF equations are simple to use for systems with small n (e.g. $n < \mathcal{O}(10^2)$). The storage required to evaluate the KF equations however goes as $\sim \mathcal{O}(n^2)$ and the scaling of the number of floating-point operations required is even higher. Ensemble methods remove the need to represent large matrices explicitly and so are often a feasible way of approximating the KF equations. Since n is only 3 for the Lorenz 63 model, it is actually practical to apply the ordinary KF, but here we examine the EnSRKF just to demonstrate the workings of the method. The starting point for derivation of the EnSRKF equations is the KF equations.

Suppose that we have N ensemble members, the KF update (17)for kth member is

$$\mathbf{x}_k^{\mathrm{a}} = \mathbf{x}_k^{\mathrm{f}} + \mathbf{P}^{\mathrm{f}}\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{P}^{\mathrm{f}}\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1}(\mathbf{y} - \mathbf{H}\mathbf{x}_k^{\mathrm{f}}) \tag{22}$$

All parts of this expression are known except for $\mathbf{P}^{\mathrm{f}}$, but $\mathbf{P}^{\mathrm{f}}$ can be approximated from the ensemble as follows

$$\mathbf{P}^{\mathrm{f}} \approx \frac{1}{N-1}\sum_{k=1}^{N}(\mathbf{x}_k^{\mathrm{f}} - \bar{\mathbf{x}}_k^{\mathrm{f}})(\mathbf{x}_k^{\mathrm{f}} - \bar{\mathbf{x}}_k^{\mathrm{f}})^{\mathrm{T}}. \tag{23}$$

This is an approximation because $\mathbf{P}^{\mathrm{f}}$ is derived from a statistical sample. It will be helpful to write this in the matrix notation used in (3), where $\mathbf{A}^{\mathrm{f}'}$ is the $n \times N$ matrix of ensemble member perturbations. Equation (23) may then be written as

$$\mathbf{P}^{\mathrm{f}} \approx \frac{1}{N-1}\mathbf{A}^{\mathrm{f}'}\mathbf{A}^{\mathrm{f}'\mathrm{T}}, \tag{24}$$

where the sum in (23) is implied in the matrix algebra. This result applies equally well to the analysis error covariance matrix:

$$\mathbf{P}^{\mathrm{a}} \approx \frac{1}{N-1}\mathbf{A}^{\mathrm{a}'}\mathbf{A}^{\mathrm{a}'\mathrm{T}}. \tag{25}$$

Equation (22) may also be written in matrix form

$$\mathbf{A}^{\mathrm{a}} = \mathbf{A}^{\mathrm{f}} + \mathbf{P}^{\mathrm{f}}\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{P}^{\mathrm{f}}\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1}(\mathbf{Y} - \mathbf{H}\mathbf{A}^{\mathrm{f}}), \tag{26}$$

where $\mathbf{Y}$ is the $p \times N$ matrix of identical columns comprising the observation vector $\mathbf{y}$. This matrix equation is equivalent to the vector equation (22), but where in (26) each ensemble member corresponds to a given column. The ensemble mean of this is

$$\overline{\mathbf{A}^{\mathrm{a}}} = \overline{\mathbf{A}^{\mathrm{f}}} + \mathbf{A}^{\mathrm{f}'}\mathbf{A}^{\mathrm{f}'\mathrm{T}}\mathbf{H}^{\mathrm{T}}\left\{\mathbf{H}\mathbf{A}^{\mathrm{f}'}\mathbf{A}^{\mathrm{f}'\mathrm{T}}\mathbf{H}^{\mathrm{T}} + (N-1)\mathbf{R}\right\}^{-1}(\overline{\mathbf{Y}} - \mathbf{H}\overline{\mathbf{A}^{\mathrm{f}}}). \tag{27}$$

(and noting that $\overline{\mathbf{Y}} = \mathbf{Y}$).

Equation (24) is now substituted into the second equality in (18), with the Kalman update from (19):

$$\begin{aligned}
\mathbf{P}^{\mathrm{a}} &= \mathbf{P}^{\mathrm{f}} - \mathbf{P}^{\mathrm{f}}\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{P}^{\mathrm{f}}\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1}\mathbf{H}\mathbf{P}^{\mathrm{f}}, \\
&= \frac{1}{N-1}\mathbf{A}^{\mathrm{f}'}\mathbf{A}^{\mathrm{f}'\mathrm{T}} - \frac{1}{N-1}\mathbf{A}^{\mathrm{f}'}\mathbf{A}^{\mathrm{f}'\mathrm{T}}\mathbf{H}^{\mathrm{T}}\left\{\mathbf{H}\frac{1}{N-1}\mathbf{A}^{\mathrm{f}'}\mathbf{A}^{\mathrm{f}'\mathrm{T}}\mathbf{H}^{\mathrm{T}} + \mathbf{R}\right\}^{-1}\mathbf{H}\frac{1}{N-1}\mathbf{A}^{\mathrm{f}'}\mathbf{A}^{\mathrm{f}'\mathrm{T}}.
\end{aligned}$$

In order to simplify the notation, we use matrices $\mathbf{S}$ and $\mathbf{C}$ as defined in (4) and (5) respectively. These make the above into

$$\begin{aligned}
\mathbf{P}^{\mathrm{a}} &= \frac{1}{N-1}\mathbf{A}^{\mathrm{f}'}\left[\mathbf{I} - \mathbf{S}^{\mathrm{T}}\left\{\mathbf{S}\mathbf{S}^{\mathrm{T}} + (N-1)\mathbf{R}\right\}^{-1}\mathbf{S}\right]\mathbf{A}^{\mathrm{f}'\mathrm{T}}, \\
\mathbf{A}^{\mathrm{a}'}\mathbf{A}^{\mathrm{a}'\mathrm{T}} &= \mathbf{A}^{\mathrm{f}'}\left[\mathbf{I} - \mathbf{S}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{S}\right]\mathbf{A}^{\mathrm{f}'\mathrm{T}}.
\end{aligned} \tag{28}$$

With $\mathbf{S}$ and $\mathbf{C}$, (27) is similarly

$$\overline{\mathbf{A}^{\mathrm{a}}} = \overline{\mathbf{A}^{\mathrm{f}}} + \mathbf{A}^{\mathrm{f}'}\mathbf{S}^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{Y} - \mathbf{H}\overline{\mathbf{A}^{\mathrm{f}}}). \tag{29}$$

The idea of a square-root scheme is to find an ensemble of analysis perturbations (in the matrix $\mathbf{A}^{\mathrm{a}'}$) that have the covariance given by (28) (think of the matrix $\mathbf{A}^{\mathrm{a}'}$ as the 'square-root' of the analysis error covariance matrix as in (25)). Once such an ensemble matrix is found, it is added to the mean $\overline{\mathbf{A}^{\mathrm{a}}}$ in (29) to give the full ensemble. The next step is to find $\mathbf{A}^{\mathrm{a}'}$ that has the property of (25). Firstly, since $\mathbf{C}$ is a square and symmetric matrix, it may be written in its eigen-decomposition as in (6), where in (6) $\mathbf{Z}$ is the $p \times p$ matrix of eigenvectors ($\mathbf{Z}\mathbf{Z}^{\mathrm{T}} = \mathbf{I}$) and Λ is the $p \times p$ matrix of eigenvalues. Using the additional definition of the $p \times N$ matrix $\mathbf{X}$ as in (8) makes (28) into

$$\mathbf{A}^{\mathrm{a}'}\mathbf{A}^{\mathrm{a}'\mathrm{T}} = \mathbf{A}^{\mathrm{f}'}\left[\mathbf{I} - \mathbf{X}^{\mathrm{T}}\mathbf{X}\right]\mathbf{A}^{\mathrm{f}'\mathrm{T}}.$$

Now decomposing $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ into its eigen-decomposition as in (9) where $\mathbf{V}$ is the $N \times N$ matrix of eigenvectors ($\mathbf{V}\mathbf{V}^{\mathrm{T}} = \mathbf{I}$ and $\mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{I}$) and $\Sigma^{\mathrm{T}}\mathbf{\Sigma}$ is the $N \times N$ matrix of eigenvalues gives:

$$\begin{aligned}\mathbf{A}^{\mathrm{a}'}\mathbf{A}^{\mathrm{a}'\mathrm{T}} &= \mathbf{A}^{\mathrm{f}'}\left[\mathbf{I} - \mathbf{V}\mathbf{\Sigma}^{\mathrm{T}}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}\right]\mathbf{A}^{\mathrm{f}'\mathrm{T}}, \\ &= \mathbf{A}^{\mathrm{f}'}\mathbf{V}\left[\mathbf{I} - \mathbf{\Sigma}^{\mathrm{T}}\mathbf{\Sigma}\right]\mathbf{V}^{\mathrm{T}}\mathbf{A}^{\mathrm{f}'\mathrm{T}}.\end{aligned}$$

Matrix $\mathbf{I} - \mathbf{\Sigma}^{\mathrm{T}}\mathbf{\Sigma}$ is a diagonal square matrix, so finding a square root of this matrix is simple. One such square-root matrix is $[\mathbf{I} - \mathbf{\Sigma}^{\mathrm{T}}\mathbf{\Sigma}]^{1/2}\mathbf{V}^{\mathrm{T}}$ (this matrix times its transpose gives $\mathbf{I} - \mathbf{\Sigma}^{\mathrm{T}}\mathbf{\Sigma}$) and leads to

$$\mathbf{A}^{\mathrm{a}'}\mathbf{A}^{\mathrm{a}'\mathrm{T}} = \mathbf{A}^{\mathrm{f}'}\mathbf{V}\left[\mathbf{I} - \mathbf{\Sigma}^{\mathrm{T}}\mathbf{\Sigma}\right]^{1/2}\mathbf{V}^{\mathrm{T}}\mathbf{V}\left[\mathbf{I} - \mathbf{\Sigma}^{\mathrm{T}}\mathbf{\Sigma}\right]^{\mathrm{T}/2}\mathbf{V}^{\mathrm{T}}\mathbf{A}^{\mathrm{f}'\mathrm{T}},$$

which, after comparing to (29) leads to this matrix of analysis ensemble perturbations:

$$\mathbf{A}^{\mathrm{a}'} = \mathbf{A}^{\mathrm{f}'}\mathbf{V}\left[\mathbf{I} - \mathbf{\Sigma}^{\mathrm{T}}\mathbf{\Sigma}\right]^{1/2}\mathbf{V}^{\mathrm{T}},$$

(as in (7)). The important point here is that the largest matrices we need to store have dimensions $n \times N$ ($\mathbf{A}^{\mathrm{f}'}$ and $\mathbf{A}^{\mathrm{a}'}$) and we need to perform an eigen-decomposition on only a $p \times p$ matrix ($\mathbf{C}$) and on an $N \times N$ matrix ($\mathbf{X}^{\mathrm{T}}\mathbf{X}$). In real problems, usually $p \ll n$ and $N \ll n$, which makes the EnSRKF an efficient approach to the DA problem.

Acknowledgements

This documentation and the web interface and the C++ code was written by R.N. Bannister. The IDL code written by S. Migliorini. The Lorenz 63 model was invented by E.N. Lorenz [2]. An original source of the EnSRKF is given by J.S. Whitaker and T.M. Hamill (2002) [4].

References

[1] Kalman R.E. (1960). A new approach to linear filtering and prediction problems, Journal of Basic Engineering 82, pp. 35–45.

[2] Lorenz E.N. (1963). Deterministic nonperiodic flow, J. Atmos. Sci. 42, pp.433-71.

[3] Press W.H., Teukolsky S.A. ,Vetterling W.T., Flannery B.P., Numerical Recepies in Fortran, Cambridge University Press, 1992, Sec. 16.1.

[4] Whitaker J.S., Hamill T.M. (2002). Ensemble data assimilation without perturbed observations, Mon. Weather Rev. 130, pp. 1913-24.