Data assimilation with the Lorenz equations

Amos S. Lawless, University of Reading

Contents

1 The Lorenz equations

We consider various data assimilation schemes applied to the Lorenz equations, a simple dynamical model with chaotic behaviour. The Lorenz equations are given by the nonlinear system

$$\frac{dx}{dt} = -\sigma(x-y), \tag{1}$$
$$\frac{dy}{dt} = \rho x - y - xz, \tag{2}$$
$$\frac{dz}{dt} = xy - \beta z, \tag{3}$$

where $x = x(t)$, $y = y(t)$, $z = z(t)$ and σ, ρ, β are parameters, which in these experiments are chosen to have the values 10, 28 and 8/3 respectively.

The system is discretized using a second order Runge-Kutta method, which gives the following discrete equations:

$$\begin{aligned} x^{k+1} &= x^k + \sigma\Delta t/2[2(y^k - x^k) + \Delta t(\rho x^k - y^k - x^k z^k) \\ &- \sigma\Delta t(y^k - x^k)], \end{aligned} \tag{4}$$
$$\begin{aligned} y^{k+1} &= y^k + \Delta t/2[\rho x^k - y^k - x^k z^k + \rho(x^k + \sigma\Delta t(y^k - x^k)) - y^k \\ &- \Delta t(\rho x^k - y^k - x^k z^k) \\ &- (x^k + \sigma\Delta t(y^k - x^k))(z^k + \Delta t(x^k y^k - \beta z^k))], \end{aligned} \tag{5}$$
$$\begin{aligned} z^{k+1} &= z^k + \Delta t/2[x^k y^k - \beta z^k \\ &+ (x^k + \Delta t\sigma(y^k - x^k))(y^k + \Delta t(\rho x^k - y^k - x^k z^k)) \\ &- \beta z^k - \Delta t(x^k y^k - \beta z^k)], \end{aligned} \tag{6}$$

where Δt is the model time step and k is the time step index.

2 Four-dimensional variational data assimilation (4D-Var)

2.1 Introduction

The 4D-Var schemes in these programs minimize a function of the form

$$\mathcal{J} = \frac{1}{2}(\mathbf{x}_0 - \mathbf{x}_0^b)^T \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}_0^b) + \frac{1}{2}\sum_{i=0}^{n}(\mathbf{y}(i) - H_i(\mathbf{x}_i))^T \mathbf{R}^{-1}(\mathbf{y}(i) - H_i(\mathbf{x}_i), \tag{7}$$

where we assume that $\mathbf{B} = \sigma_b^2\mathbf{I}$ and $\mathbf{R} = \sigma_o^2\mathbf{I}$. A full 4D-Var scheme minimizes this cost function by use of the nonlinear model and its adjoint, whereas an incremental 4D-Var scheme minimizes a series of simplified cost functions in different 'outer loops'. You are provided with routines for both types of schemes. The routines used are as follows:

lorenz4d.m	Top level routine for full 4D-Var
lorenz4d_inc.m	Top level routine for incremental 4D-Var
calcfg.m	Calculate cost function and its gradient for full 4D-Var
calcfg_inc.m	Calculate cost function and its gradient for incremental 4D-Var
modeuler.m	Nonlinear model for Lorenz system
modeuler_tl.m	Tangent linear model
modeuler_adj.m	Adjoint model
test_tl.m	Test tangent linear model
test_adj.m	Test adjoint model
test_grad.m	Test of *calcfg*
test_gradinc.m	Test of *calcfg_inc*
menu_asl	Used to provide menus

2.2 Test routines - Building a 4D-Var system

When building a 4D-Var system, there are standard ways of testing the various components before it is used for assimilation. You can experiment with these tests.

2.2.1 Test of tangent linear model

Suppose that M is a nonlinear model and $\mathbf{M}$ is the tangent linear model. Then for small perturbations $\gamma\delta\mathbf{x}$ we have

$$M(\mathbf{x} + \gamma\delta\mathbf{x}) - M(\mathbf{x}) \approx \mathbf{M}(\mathbf{x})\gamma\delta\mathbf{x}. \tag{8}$$

Hence if we plot the *relative error*

$$E_R = \frac{M(\mathbf{x} + \gamma\delta\mathbf{x}) - M(\mathbf{x})}{\mathbf{M}(\mathbf{x})\gamma\delta\mathbf{x}} \tag{9}$$

as $\gamma \to 0$ we should find that $E_R \to 0$.

Exercise: Use the routine *test_tl* to plot the relative error. Try introducing an error into the tangent linear code *modeuler_tl* and see what effect it has on the test.

2.2.2 Test of adjoint model

For a linear model $\mathbf{M}$ and its adjoint $\mathbf{M}^*$ we have the identity

$$< \mathbf{M}\delta\mathbf{x}, \mathbf{M}\delta\mathbf{x} > = < \delta\mathbf{x}, \mathbf{M}^*\mathbf{M}\delta\mathbf{x} > \tag{10}$$

for any inner product $<,>$ and perturbation $\delta\mathbf{x}$. This can be used to test that the adjoint is coded correctly.

Exercise: Use the routine *test_adj* to test the adjoint code. Try introducing an error into the adjoint code *modeuler_adj* and see what effect it has.

2.2.3 Gradient test

Let $\mathcal{J}$ be a cost function and $\nabla\mathcal{J}$ be its gradient. Then we can check that the exact gradient of the cost function has been coded by using the identity

$$\Phi(\alpha) = \frac{\mathcal{J}(\mathbf{x}+\alpha\mathbf{h}) - \mathcal{J}(\mathbf{x})}{\alpha\mathbf{h}^T\nabla\mathcal{J}(x)} = 1 + O(\alpha), \tag{11}$$

where $\mathbf{h}$ is a vector of unit length, which we can take to be $\nabla\mathcal{J}(x)||\nabla\mathcal{J}(x)||^{-1}$. For small values of α not too close to machine zero we expect $\Phi(\alpha)$ to be close to 1.

Exercise: Use the program *test_grad* or *test_gradinc* to test either of the two cost functions. The output of these routines is a plot of $\Phi(\alpha)$ and a plot of $|\Phi(\alpha) - 1|$. Try introducing an error into the gradient calculation to see how this affects the test results.

2.3 Assimilation program

The routines used to run assimilation experiments are *lorenz4d* for the full 4D-Var and *lorenz4d_inc* for incremental 4D-Var. The menu options you must specify, with some suggested values, are

Initial values of x, y, z	0.0–5.0
Assimilation period (in seconds)	0–10
Forecast period (in seconds)	Any
Time step (in seconds)	0.0–0.05
Frequency of observations (in time steps)	Any
Noise on background	Variance = 0–4 (excluding zero)
Noise on observations	Variance = 0–4 (excluding zero)
Convergence criteria	Default values given
Number of outer loops	2 (*Incremental version only*)

Note that the time step must be a divisor of your total time, so values such as 0.02, 0.025, 0.05 work well. The output of the program is the fields of x and z, the errors in x and z and the convergence of the cost function and its gradient. The final norm of the gradient is also output in the Matlab command window.

The noise on the background and observations is produced randomly each time the program is run. In order to compare the effect of different settings you can choose to use the same realisation of random noise as in your previous experiment by answering 'Yes' to the question 'Read in noise from file?'. Note that in order for this to work the number of observations must remain the same.

2.4 Suggested exercises

Start with the conditions
truth=(1.0,1.0,1.0)
Assimilation period = 2
Forecast period = 3
Time step = 0.05
Frequency of observations = 2

1. Run the 4DVar with the different relative errors on the background and observations. How does the behaviour of the scheme change?

2. Is it better to have very few accurate observations or more observations which are less accurate? Consider both the accuracy of the analysis and the rate of convergence.

3. Is it better to have a long assimilation window with few observations or a short assimilation window with more observations? Does this depend on how much error there is on the observations? Consider both the accuracy of the analysis and the rate of convergence.

4. How does the rate of convergence change as the background error changes?

5. Compare full 4D-Var and incremental 4D-Var for the same total number of iterations. Are there conditions for which one scheme is better than the other?

6. For the incremental 4D-Var investigate the effect of different outer loops with the same total number of iterations. Compare against the full 4D-Var solution.

2.5 Advanced exercises

The following exercises require you to understand and change the code.

1. Investigate the effect of correlated observation errors in 4D-Var.
   Introduce correlations in your observation errors by changing the program so that the same random noise is used to create the observation error for x, y and z (or just two of these). How does this affect the assimilation results? Consider what happens when the observation error covariance matrix is assumed to be diagonal and not.

2. Investigate the effect of biased observations in 4D-Var.
   Try replacing the random observation error with a constant bias for one or more of the variables. What is the effect on the analysis?

3. Investigate the performance of 4D-Var when the model state is only partially observed.
   Change the code so that only two components of the state vector are observed. How well is the other component retrieved by the assimilation? Compare the effect of using a diagonal and non-diagonal background error covariance matrix.

4. Investigate the effect of model error in 4D-Var. You may consider
   (a) random stochastic error;
   (b) an error in the parameters;
   (c) a bias error.
   Usually the numerical model we use to assimilate is not an exact representation of the true system, but will contain model errors. We can investigate the effect of this in a simple assimilation experiments by using one version of the model to produce the 'truth' trajectory and the observations and using a different version of the model to assimilate. To add error to the assimilation model you may
   (a) add a random forcing to one of the model equations;
   (b) change one of the model parameters σ, β or ρ to be slightly different in the assimilation model;
   (c) add a constant forcing to one of the model equations.
   How does this affect the analysis?

Four-dimensional variational data assimilation (4D-Var)

Chiara Piccolo
UK Met Office

ESA Summer School 2012

Historical Background:
What has been important for getting the best NWP forecast?

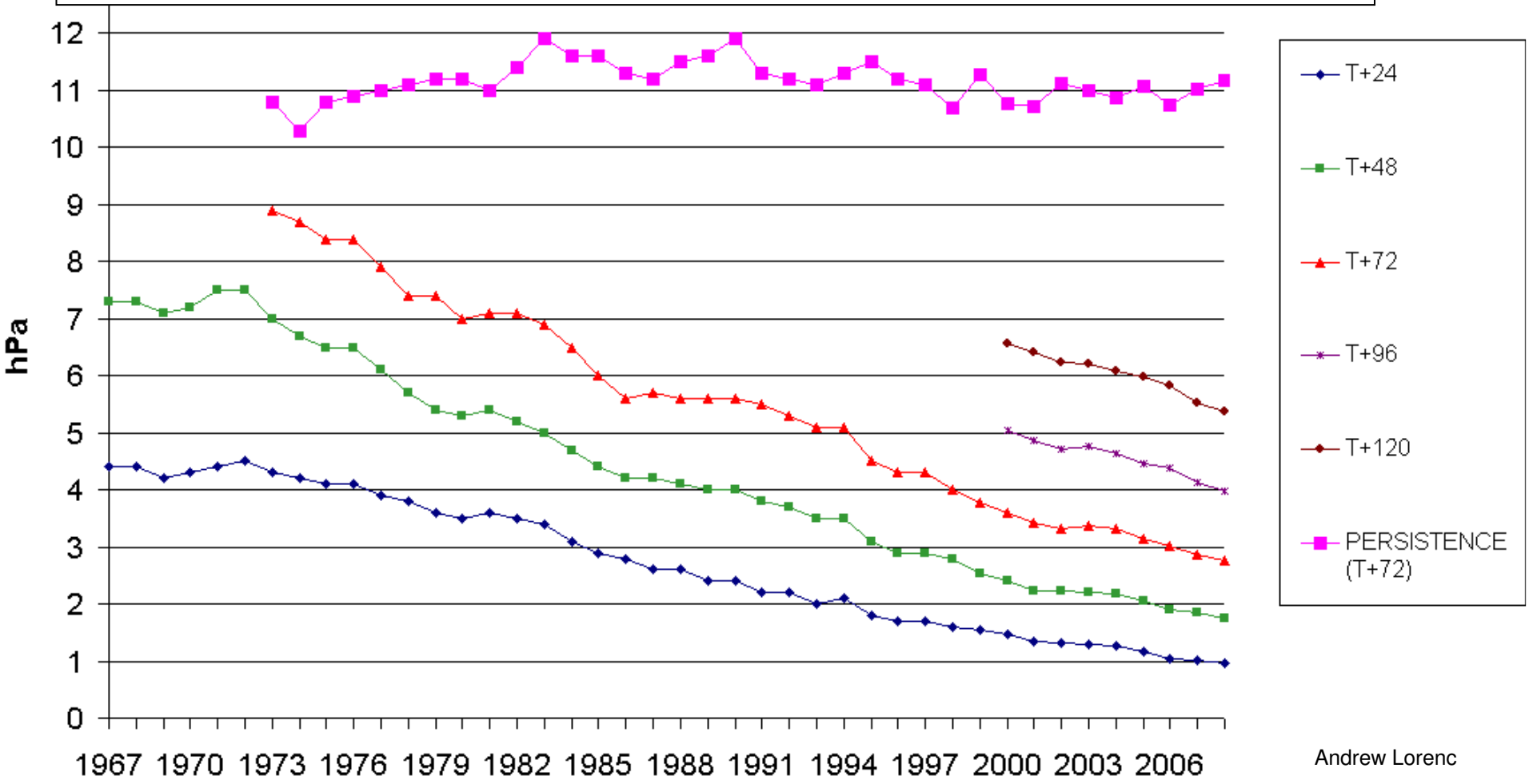
NWP systems are improving by 1 day of predictive skill per decade. This has been due to:

1. *Model improvements, especially resolution.*
2. *Careful use of forecast & observations, allowing for their information content and errors. Achieved by variational assimilation e.g. of satellite radiances.*
3. *4D-Var.*
4. *Better observations.*

Performance Improvements

"Improved by about a day per decade"

Met Office RMS surface pressure error over the N. Atlantic & W. Europe



Importance of forecast model

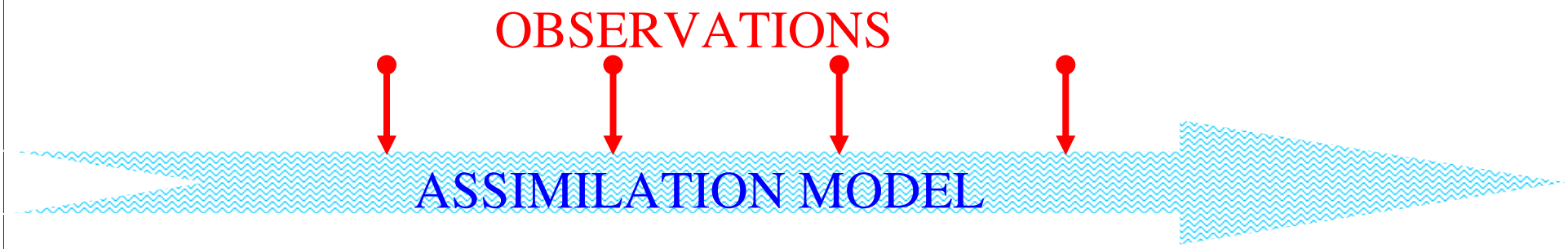
- A large part of the increase in ***assimilation*** accuracy comes from improvements to the model
- A large part of the increase in model accuracy comes from improvements in resolution
- The resolution has been limited by computer power.
- Still true today – a larger part of this year's increases in computer power at the Met Office will be spent on increased resolution than on improved algorithms.

NWP is an extreme example here.
Other applications of DA place less emphasis on the model and more on use of data.

Data Assimilation is the process of absorbing and incorporating observed information into a prognostic model.

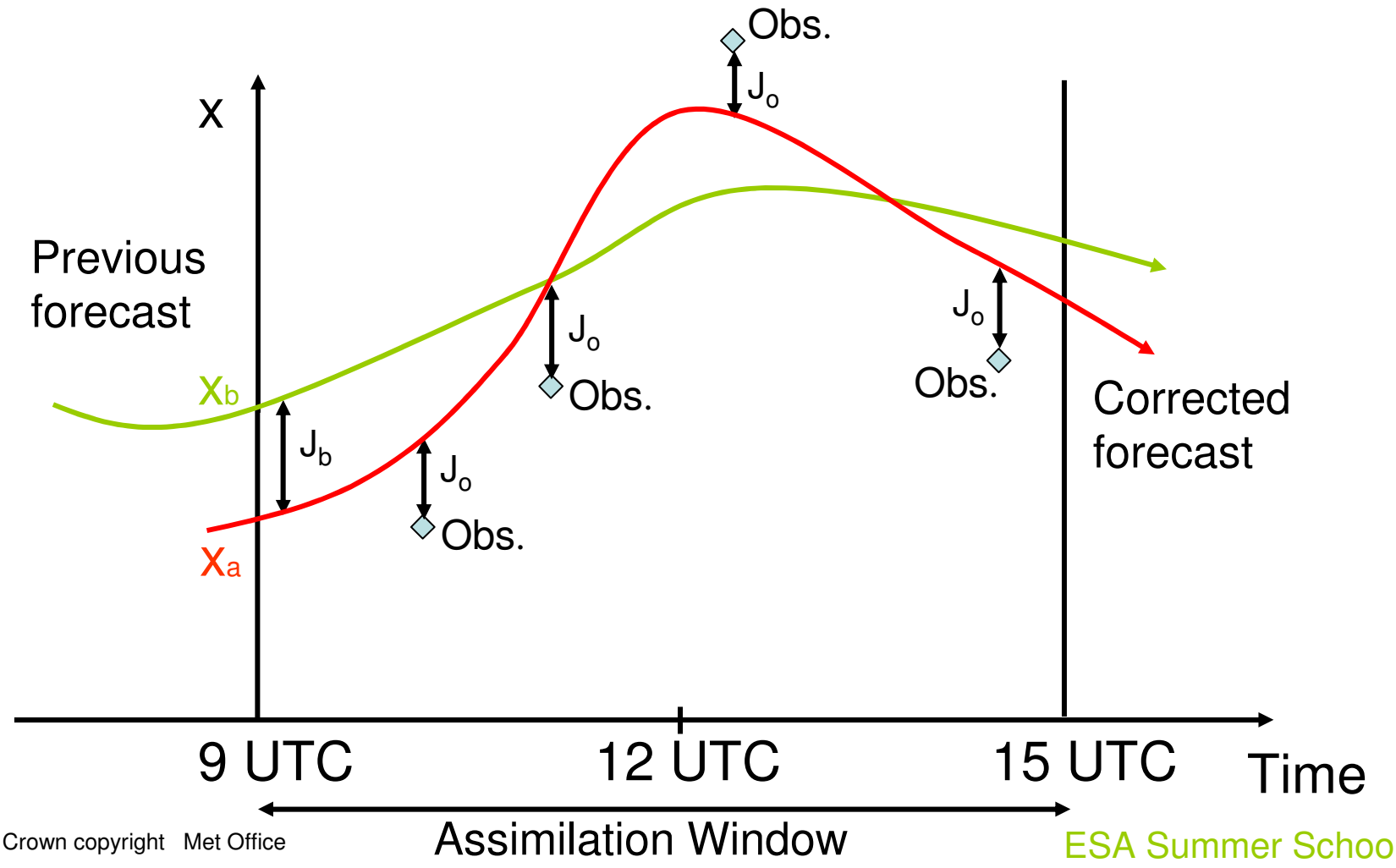
This is normally done by integrating the model forward in time, adding observations.

- The model state summarises in an organised way the information from earlier observations.
- It is modified to incorporate new observations, by combining new & old information in a statistically optimal way.



- *At any time, the model state usually contains more information than the current observations.*
- *Only parameters well represented by the model can be assimilated in this way.*

Data Assimilation is the process of absorbing and incorporating observed information into a prognostic model.



Four-dimensional analysis problem

$$J(\mathbf{x}) = (\mathbf{x}-\mathbf{x}_\mathrm{b})^\mathrm{T}\mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}_\mathrm{b}) + \sum_{i=0}^{n}(\mathbf{y}_i - H_i[\mathbf{x}_i])^\mathrm{T}\mathbf{R}_i^{-1}(\mathbf{y}_i - H_i[\mathbf{x}_i])$$

$$\forall i,\quad \mathbf{x}_i = M_{0\rightarrow i}(\mathbf{x})$$

Strong constrain: the sequence of model states must be a solution of the model equations

$$\mathbf{x}_\mathrm{a} = \mathbf{x}_\mathrm{b} + \mathbf{K}(\mathbf{y} - H[\mathbf{x}_\mathrm{b}])$$

$$\mathbf{K} = \mathbf{B}\mathbf{H}^\mathrm{T}(\mathbf{H}\mathbf{B}\mathbf{H}^\mathrm{T} + \mathbf{R})^{-1}$$

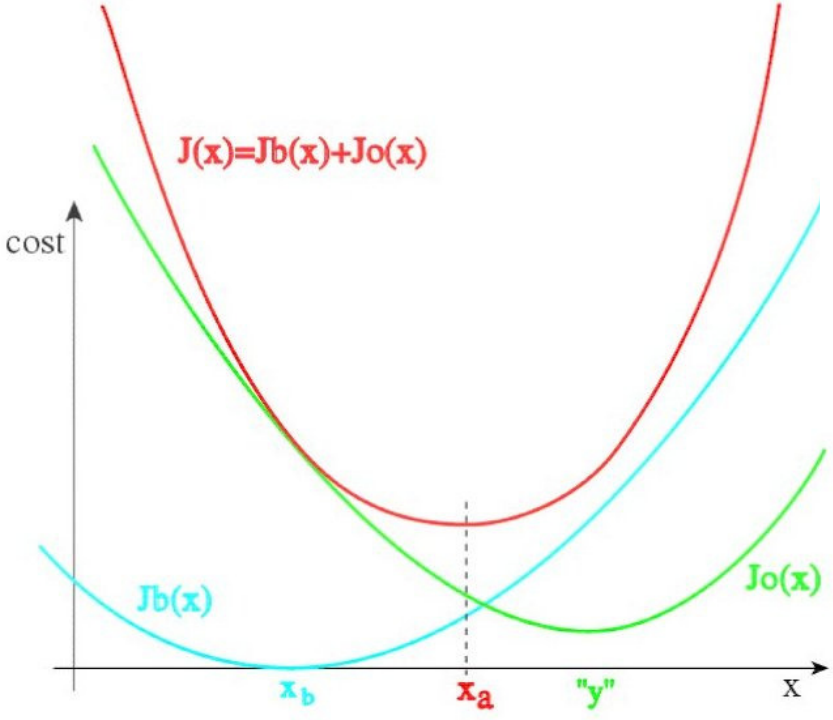
$$\mathbf{A} = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{B}(\mathbf{I} - \mathbf{K}\mathbf{H})^\mathrm{T} + \mathbf{K}\mathbf{R}\mathbf{K}^\mathrm{T}$$

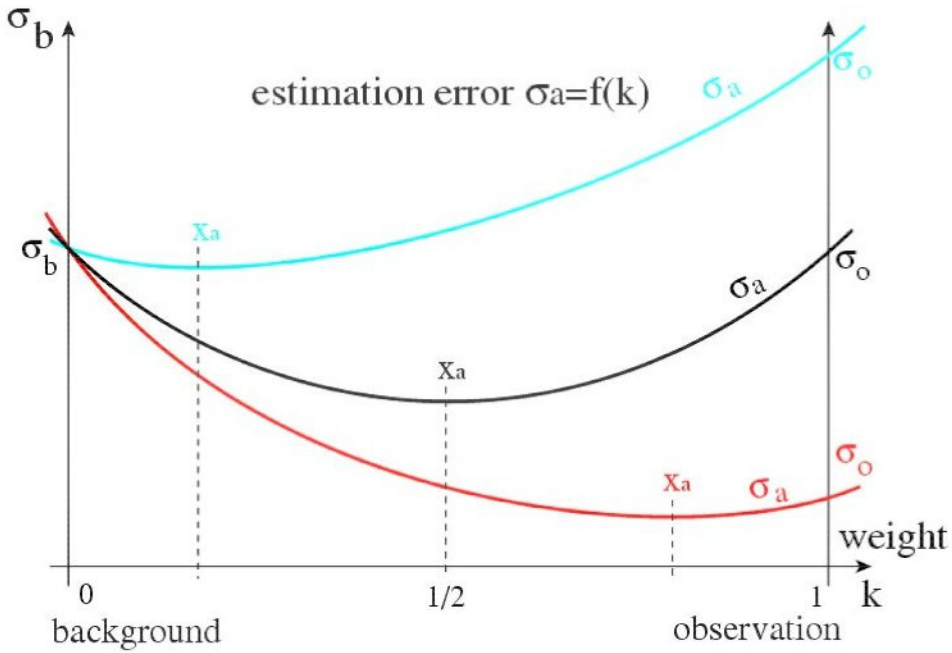
Assumptions:
Unbiased and uncorrelated Gaussian errors
Tangent linear hypothesis: the cost function can be made quadratic by assuming that the observation operator and the model can be linearised
Optimal linear analysis: we look for an analysis defined by correction to the background which is a minimum variance estimate

Example of scalar case

$$\sigma_a^2 = (1-k)^2\sigma_b^2 + k^2\sigma_o^2$$

$$k = \frac{\sigma_b^2}{\sigma_b^2 + \sigma_o^2}$$





$(\sigma_o >> \sigma_b),\ k = 0$ The analysis remains equal to the background

$(\sigma_o >> \sigma_b),\ k = 1$ The analysis is equal to the observation

$0 \le k \le 1$ The analysis is a weighted average of the background and the observation

Four-dimensional analysis problem

$$J(\mathbf{x}) = (\mathbf{x}-\mathbf{x}_\mathrm{b})^\mathrm{T}\mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}_\mathrm{b}) + \sum_{i=0}^{n}(\mathbf{y}_i - H_i[\mathbf{x}_i])^\mathrm{T}\mathbf{R}_i^{-1}(\mathbf{y}_i - H_i[\mathbf{x}_i])$$

Tangent linear hypothesis. The cost function can be made quadratic by assuming, on top of the linearization of H_i, that the M operator can be linearized, i.e.

$$\mathbf{y}_i - H_i M_{0\to i}(\mathbf{x}) \approx \mathbf{y}_i - H_i M_{0\to i}(\mathbf{x}_\mathrm{b}) - \mathbf{H}_i\mathbf{M}_{0\to i}(\mathbf{x}-\mathbf{x}_\mathrm{b})$$

where $\mathbf{M}$ is the *tangent linear (TL) model*, i.e. the differential of M.

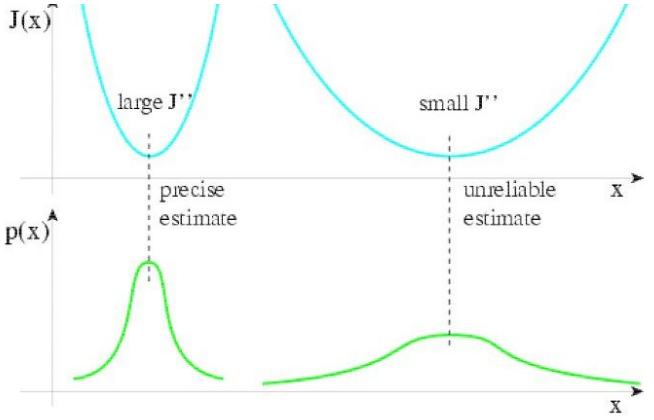
The evaluation of the 4D-Var observation cost function and its gradient, $J_\mathrm{o}(\mathbf{x})$ and $\nabla J_\mathrm{o}(\mathbf{x})$, requires one direct model integration from times 0 to n and one suitably modified *adjoint integration* made of transposes of the tangent linear model time-stepping operators $\mathbf{M}_i$.

The Hessian of the cost function of the variational analysis is equal to twice the inverse of the analysis error covariance matrix:

$$\mathbf{A} = \left(\frac{1}{2}J''\right)^{-1}$$

$$\mathbf{A} = (\mathbf{I}-\mathbf{KH})\mathbf{B}(\mathbf{I}-\mathbf{KH})^\mathrm{T} + \mathbf{KRK}^\mathrm{T}$$

$$\mathbf{K} = \mathbf{BH}^\mathrm{T}(\mathbf{HBH}^\mathrm{T} + \mathbf{R})^{-1}$$



Four-dimensional analysis problem

$$J(\mathbf{x}) = (\mathbf{x}-\mathbf{x}_\mathrm{b})^\mathrm{T}\mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}_\mathrm{b}) + \sum_{i=0}^{n}(\mathbf{y}_i - H_i[\mathbf{x}_i])^\mathrm{T}\mathbf{R}_i^{-1}(\mathbf{y}_i - H_i[\mathbf{x}_i])$$

4D-Var has the following characteristics:

- it works under the assumption that the model is perfect (strong constrain 4D-Var)
- it requires the implementation of the so-called *adjoint* model
- in a real time system it requires the assimilation to wait for the observations over the all 4D-Var window to be available before the analysis procedure can begin
- the analysis is used as initial state for a forecast so by construction the forecast will be completely consistent with the model equation and the four dimensional distribution of the observations until the end of the 4D-Var window
- 4D-Var is an optimal assimilation algorithm over its time window

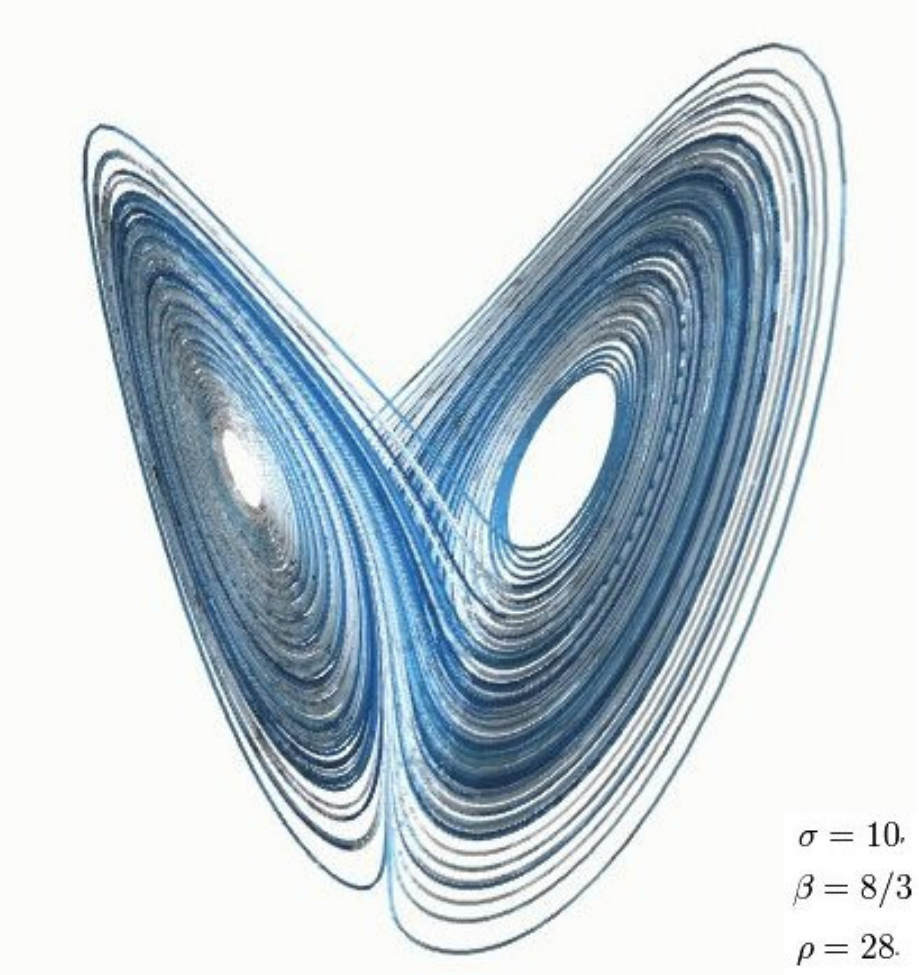
Full versus Incremental 4D-Var

- The incremental method is an empirical technique designed to reduce the cost of solving a predefined variational problem, e.g. by reducing the resolution of the increments.
- With 3D- or 4D-Var is it usually not affordable to solve the variational problem at the full model resolution. Since it is expected that most of the complexity of the analysis is in the synoptic scales, while the smaller scales are more or less forced to be realistic features by the model dynamics, the full resolution problem is solved by looking for a low-resolution correction to a high-resolution background.
- Mathematically, it can be thought as the approximation of a large problem by a sequence of smaller problems.

4D-Var versus Kalman Filter

- The Kalman Filter and its extended version (EKF) are sequential data assimilation techniques, in which each background is provided by a forecast that starts from the previous analysis.
- The analysis equations of the linear Kalman Filter are exactly the ones described in 4D-Var.
- **KF/4D-Var equivalence:** Over the same time interval assuming that the model is perfect, and that both algorithms use the same observations and the same initial state and error covariance matrix, the Kalman Filter estimate is identical to that produced by 4D-Var. The Kalman Filter solves the problem sequentially whereas 4D-Var solves the 4D problem globally over the same assimilation.
- 4D-Var actually is a Kalman smoother since it uses also the observations in the future to update the initial condition of the forecast model.

Lorenz'63 Model



In 1963 Lorenz developed a simplified mathematical model for atmospheric convection.
The model is a system of three ordinary differential equations (the **Lorenz equations**).
It is notable for having a chaotic behaviour for certain parameter values and initial conditions.

The Lorenz equations are given by the nonlinear system:

$$\frac{dx}{dt} = \sigma(y - x),$$
$$\frac{dy}{dt} = x(\rho - z) - y,$$
$$\frac{dz}{dt} = xy - \beta z.$$

Exercises of today

Start by testing 4D-Var assumptions:

- Test tangent linear hypothesis
- Test of the adjoint model
- Test of cost function gradient

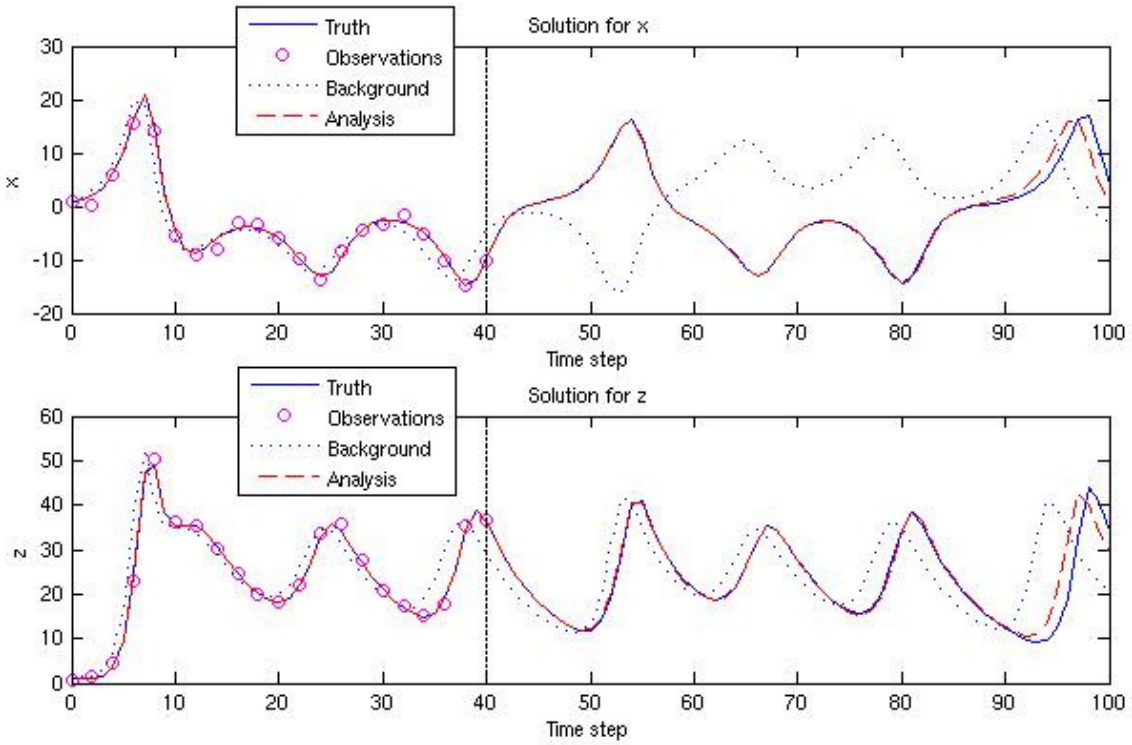
using full 4D-Var (***lorenz4d.m***) or incremental 4D-Var (***lorenz4d_inc.m***)

Explore system behaviour when changing:

- relative errors on background and observations
- number and accuracy of observations

Compare full 4D-Var and incremental 4D-Var.

Example



List of options chosen

True (x,y,z) at t=0: (1,1,1)
Length of assimilation window: 2
Length of subsequent forecast: 3
Time step: 0.05
Frequency of observations = 2
Maximum iterations: 30
Epsilon (inner loop) = 1d-5
Observations : variance of noise = 1
Background : variance of noise = 2

Background (x,y,z) at t=0:
(0.35418,0.50483,0.36102)
Analysis (x,y,z) at t=0: (0.83674,1.2238,1.0088)