Data Assimilation Practicals: Four dimensional variational data assimilation (4D-Var)

Polly Smith

Data Assimilation Research Centre

p.j.smith@reading.ac.uk

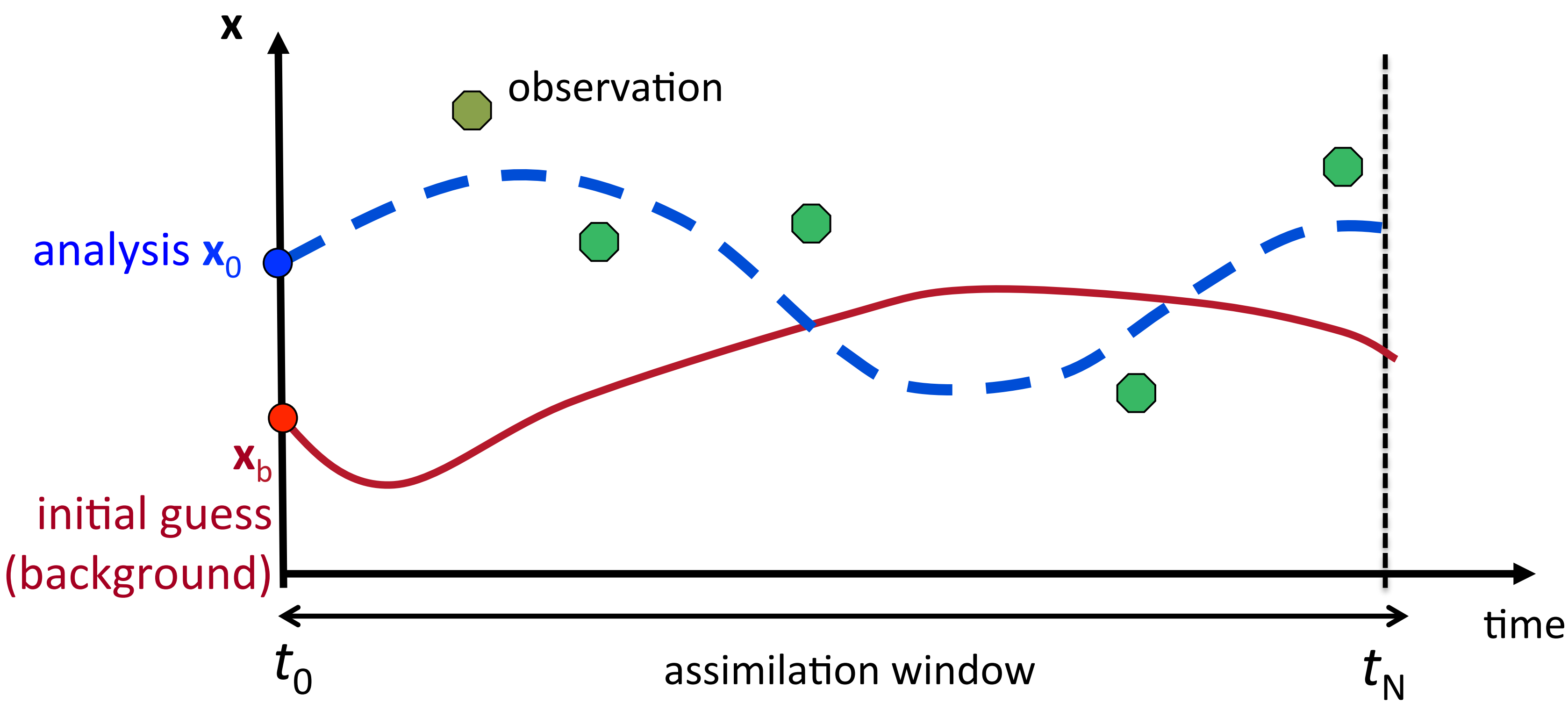
Introduction

In this practical we will be exploring two different variational data assimilation algorithms

- Full 4D-Var
- Incremental 4D-Var

4D-Var data assimilation

Aim: find the best estimate of the true state of the system (*analysis*) consistent with both observations distributed in time and the system dynamics.



4D-Var cost function

Minimize

$$\mathcal{J}(\mathbf{x}_0) = \frac{1}{2}(\mathbf{x}_0 - \mathbf{x}^b)^{\mathrm{T}}\mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}^b) + \frac{1}{2}\sum_{i=0}^{N}(\mathcal{H}_i(\mathbf{x}_i) - \mathbf{y}_i)^{\mathrm{T}}\mathbf{R}_i^{-1}(\mathcal{H}_i(\mathbf{x}_i) - \mathbf{y}_i)$$

with respect to x_0, subject to

$$\mathbf{x}_{i+1} = \mathcal{M}_i(\mathbf{x}_i),$$

$\mathbf{x}^b$ - *a priori* (background) state

$\mathbf{y}_i$ - Observations

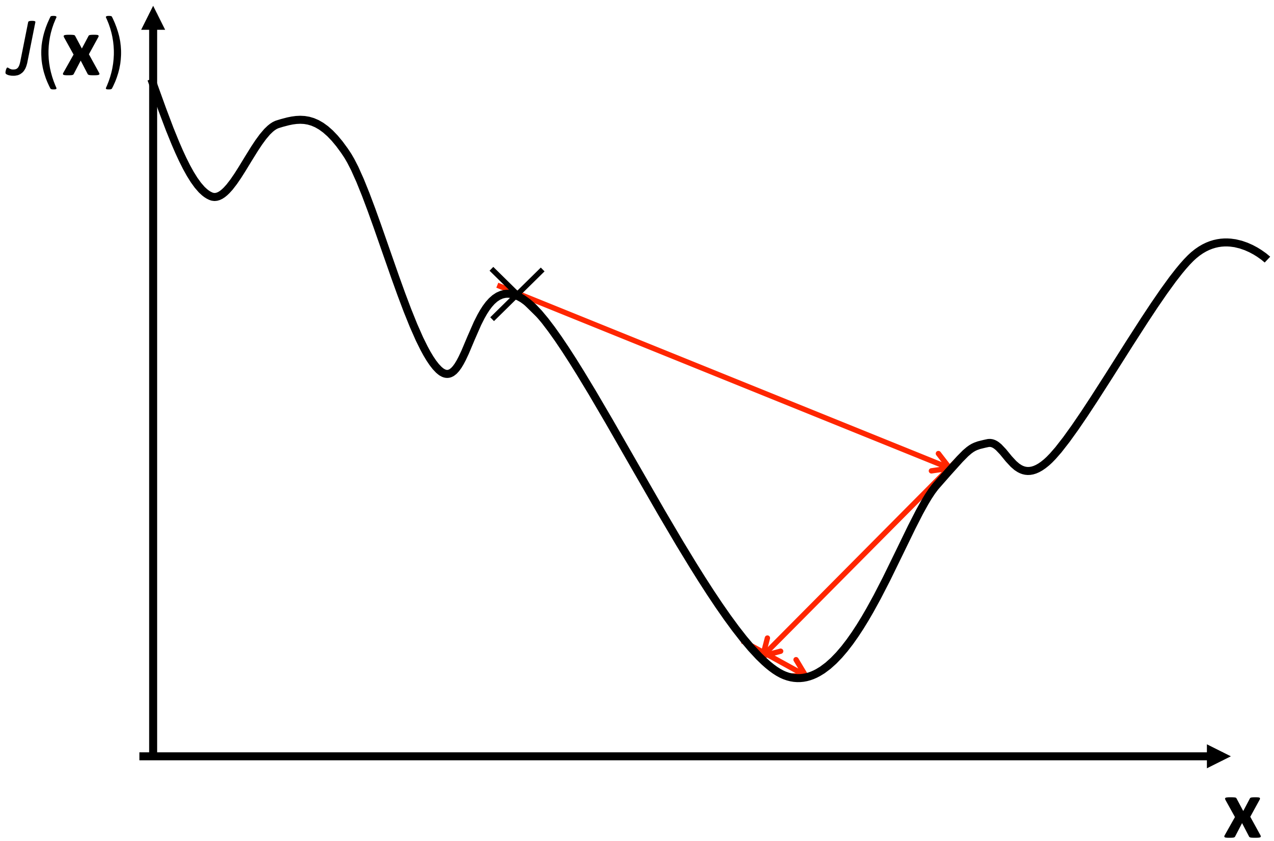
H_i - Observation operator

$\mathbf{B}$ - Background error covariance matrix

$\mathbf{R}_i$ - Observation error covariance matrix

Minimization

use iterative gradient descent method to find minimum; requires information about the gradient of the cost function, $\nabla J(\mathbf{x})$



Minimization

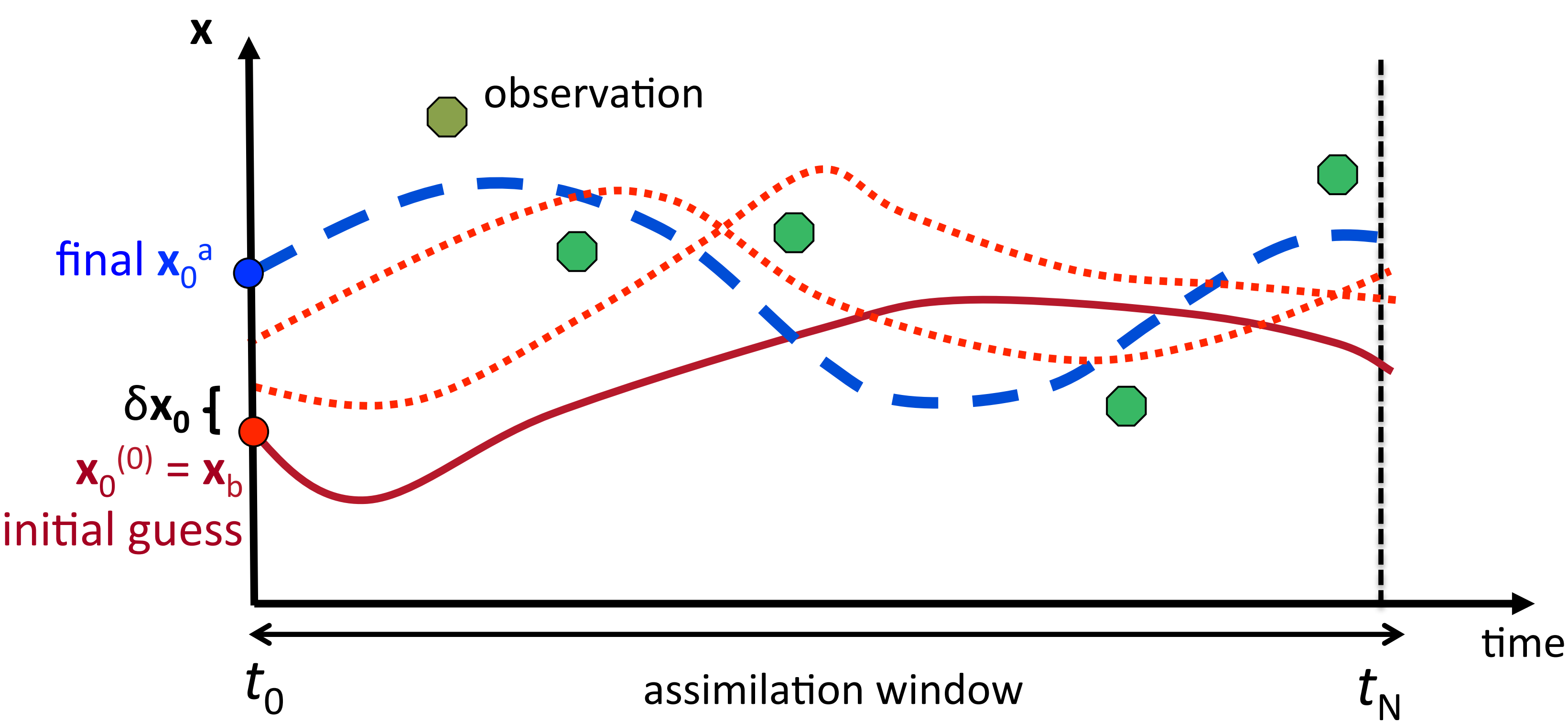
On each iteration we have to calculate $J(\mathbf{x})$ and its gradient

- to calculate $J(\mathbf{x})$ we need to run the non-linear model
- the gradient of $J(\mathbf{x})$ is obtained by backward integration of the adjoint model (from t_N to t_0)

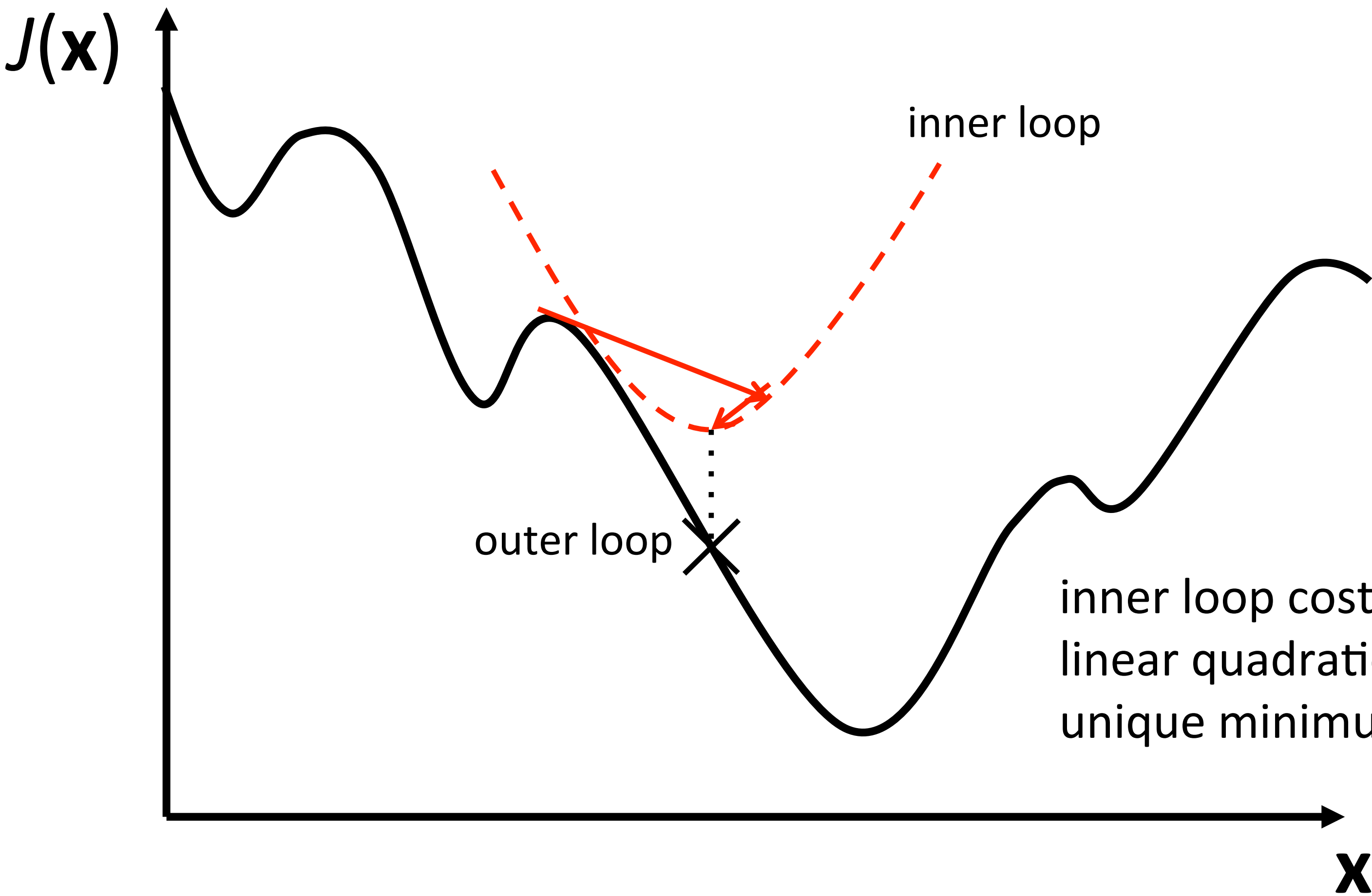
BUT using the full non-linear model equations can be very computationally expensive.

Incremental 4D-Var

Full non-linear 4D-Var problem is replaced by a sequence of (easier) linear least squares problems

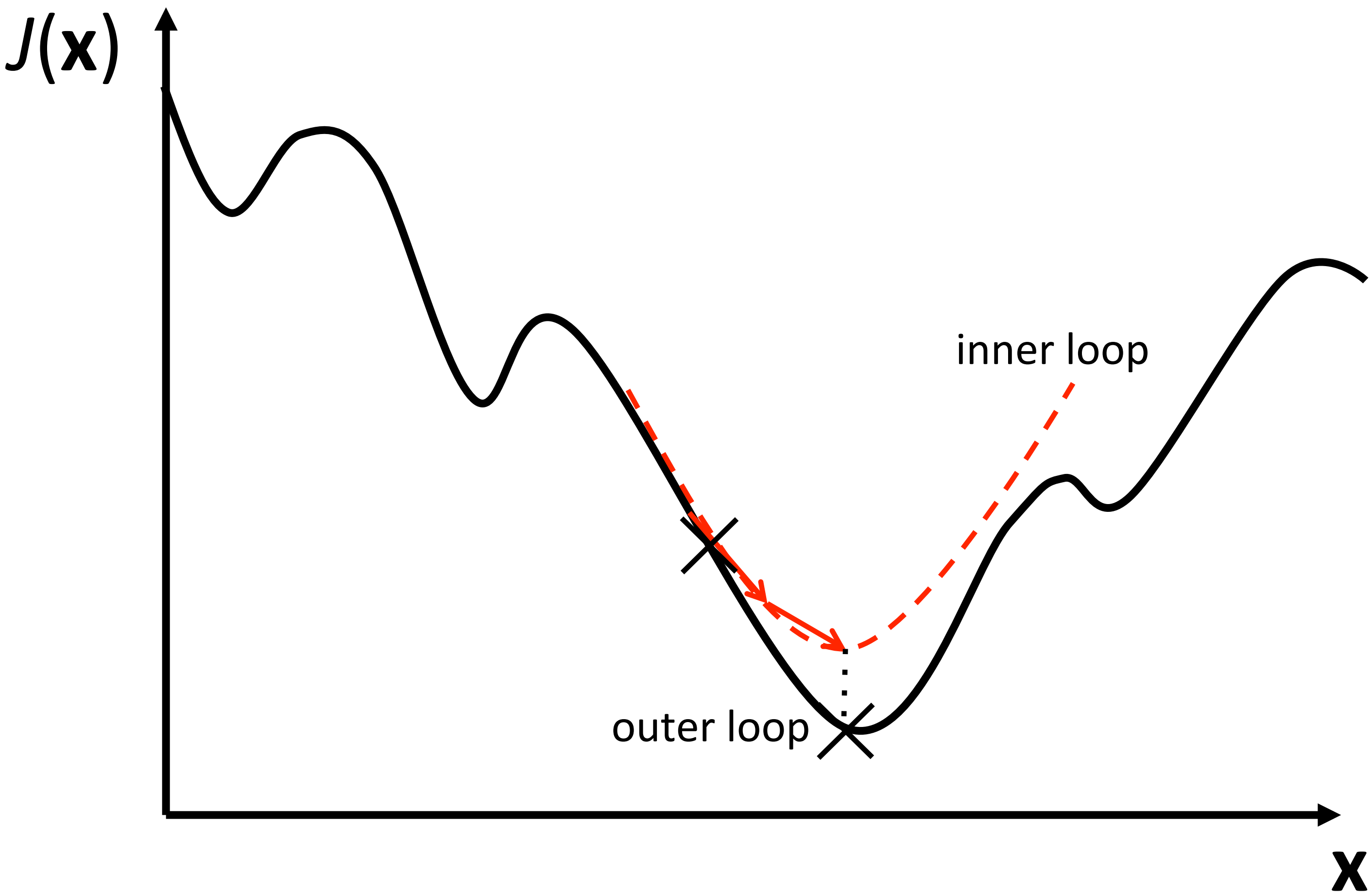


Incremental 4D-Var



inner loop cost function is linear quadratic, so has a unique minimum

Incremental 4D-Var



Incremental 4D-Var

Solve iteratively

set $\mathbf{x}_0^{(0)} = \mathbf{x}_b$

outer loop: for k = 0, … , Nouter

compute $\mathbf{d}_i^{(k)} = \mathbf{y}_i - \mathcal{H}_i(\mathbf{x}_i^{(k)})$, where $\mathbf{x}_i^{(k)} = \mathcal{M}(t_i, t_0, \mathbf{x}_0^{(k)})$

inner loop: minimise

$$\mathrm{J}^{(k)}\left(\delta\mathbf{x}_0^{(k)}\right) = \frac{1}{2}\left(\delta\mathbf{x}_0^{(k)} - \left(\mathbf{x}_b - \mathbf{x}_0^{(k)}\right)\right)^T \mathbf{B}^{-1}\left(\delta\mathbf{x}_0^{(k)} - (\mathbf{x}_b - \mathbf{x}_0^{(k)})\right) + \frac{1}{2}\sum_{i=0}^{n}\left(\mathbf{H}_i\delta\mathbf{x}_i^{(k)} - \mathbf{d}_i^{(k)}\right)^T \mathbf{R}_i^{-1}\left(\mathbf{H}_i\delta\mathbf{x}_i^{(k)} - \mathbf{d}_i^{(k)}\right)$$

$$\delta\mathbf{x}_i^{(k)} = \mathbf{M}(t_i, t_0, \mathbf{x}^{(k)})\delta\mathbf{x}_0^{(k)}$$

$\mathbf{M}$ is the tangent linear of the non-linear model

update $\mathbf{x}_0^{(k+1)} = \mathbf{x}_0^{(k)} + \delta\mathbf{x}_0^{(k)}$

The Lorenz 63 model

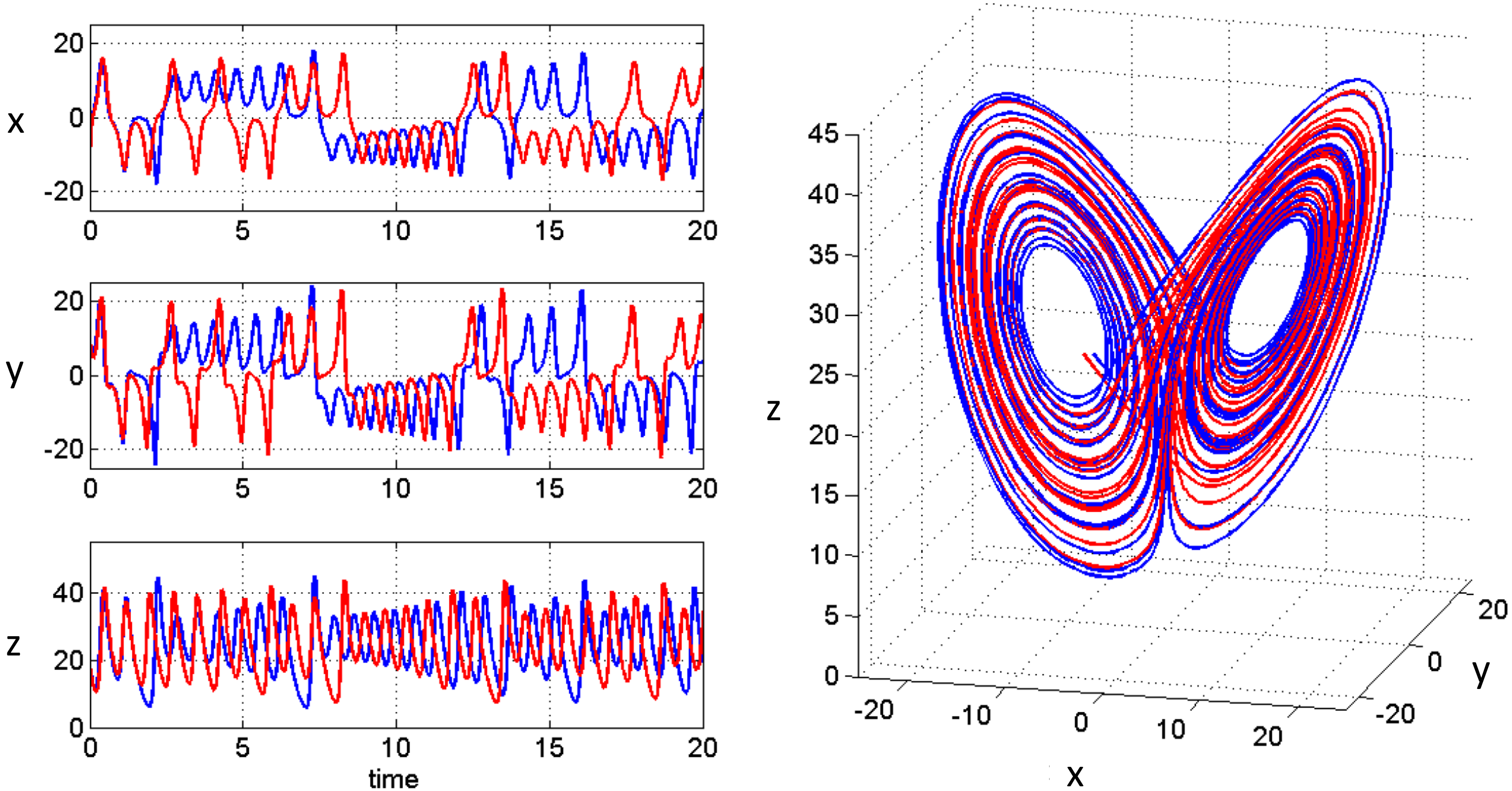
- We will use a 'simple' model to test our data assimilation techniques.
- Three variable nonlinear dynamical system that exhibits chaotic behaviour for certain parameter settings.

$$\begin{aligned} \frac{dx}{dt} &= -\sigma(x-y), \\ \frac{dy}{dt} &= \rho x - y - xz, \\ \frac{dz}{dt} &= xy - \beta z, \end{aligned}$$

parameters $\begin{cases} \sigma = 10 \\ \beta = 8/3 \\ \rho = 28 \end{cases}$

where $x = x(t),\ y = y(t),\ z = z(t)$

The Lorenz 63 model



Twin experiments

- The model is run forward from a given initial state to produce a reference or ‘truth’ trajectory.
- This ‘truth’ is used to generate synthetic observations.
- We use the same model for the assimilation but we run from a different initial state (‘background’ or first guess).
- The assimilation system is assessed on how well the analysis approximates this reference state.

Exercises

Part I

understanding how a 4D-Var system is tested, by running tests of

- Tangent linear model *(test_tl)*
- Adjoint model *(test_adj)*
- Gradient of cost function *(test_grad.m or test_gradinc)*

Test of TLM correctness

Consider a perturbation $\gamma\delta\mathbf{x}$, where γ is a scalar.

Then by a Taylor series expansion we have

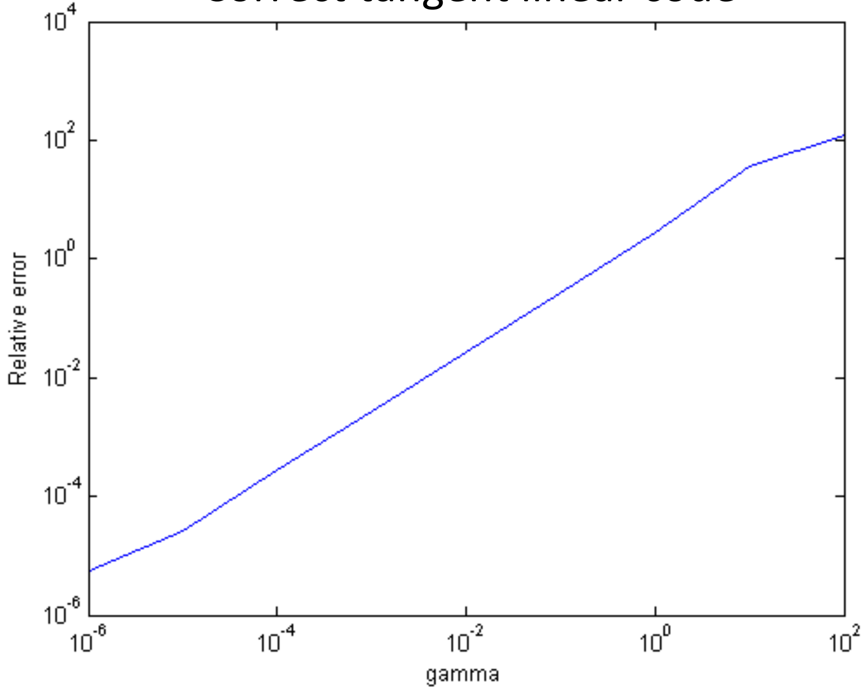
$$M(\mathbf{x}_0 + \gamma\delta\mathbf{x}) = M(\mathbf{x}_0) + \mathbf{M}(\mathbf{x}_0)\,\gamma\delta\mathbf{x} + O(\gamma^2)$$

Hence

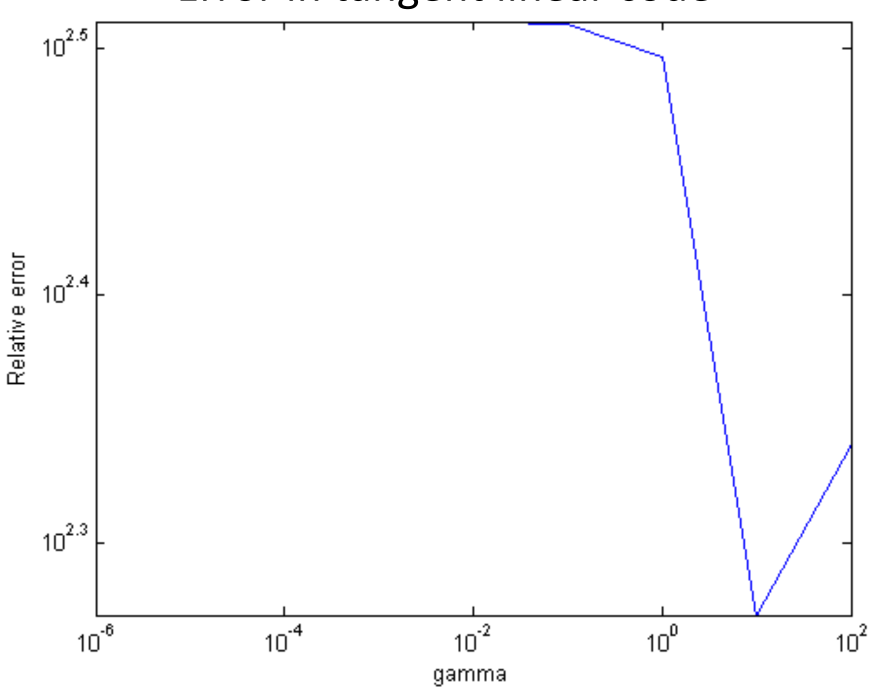
$$\lim_{\gamma \to 0} \frac{\left\| M(\mathbf{x}_0 + \gamma\delta\mathbf{x}) - M(\mathbf{x}_0) - \mathbf{M}(\mathbf{x}_0)\gamma\delta\mathbf{x} \right\|}{\left\| \mathbf{M}(\mathbf{x}_0)\gamma\delta\mathbf{x} \right\|} = 0$$

Tangent linear test

Correct tangent linear code



Error in tangent linear code



Test of adjoint model

For any operator $\mathbf{M}$ and its adjoint $\mathbf{M}^{\mathrm{T}}$ we have

$$< \mathbf{M}\delta\mathbf{x}, \mathbf{M}\delta\mathbf{x} > = < \delta\mathbf{x}, \mathbf{M}^{\mathrm{T}}\mathbf{M}\delta\mathbf{x} >$$

To test an adjoint model we

1. Start with a random perturbation $\delta\mathbf{x}$
2. Apply the TLM, which gives $\mathbf{M}\delta\mathbf{x}$
3. Apply the adjoint model to the result of 2, to obtain $\mathbf{M}^{\mathrm{T}}\mathbf{M}\delta\mathbf{x}$

If the adjoint is correct

$$< \mathbf{M}\delta\mathbf{x}, \mathbf{M}\delta\mathbf{x} > - < \delta\mathbf{x}, \mathbf{M}^{\mathrm{T}}\mathbf{M}\delta\mathbf{x} > \; = 0$$

(satisfied to machine precision)

Gradient test

By Taylor series expansion of J we have

$$J(\mathbf{x}+\alpha\mathbf{h}) = J(\mathbf{x}) + \alpha\mathbf{h}^T \nabla J(\mathbf{x}) + O(\alpha^2)$$

Define

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

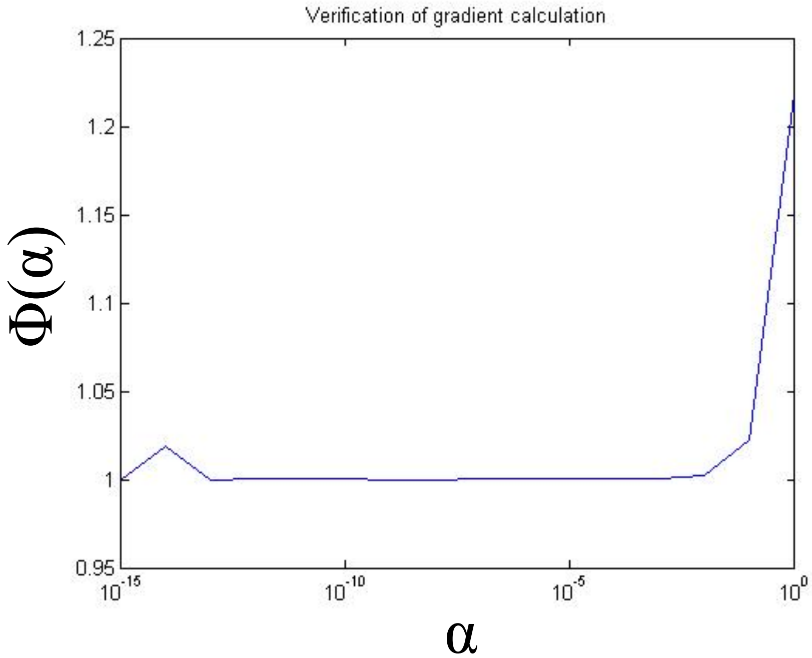
and plot $\Phi(\alpha)$ as α tends to zero.

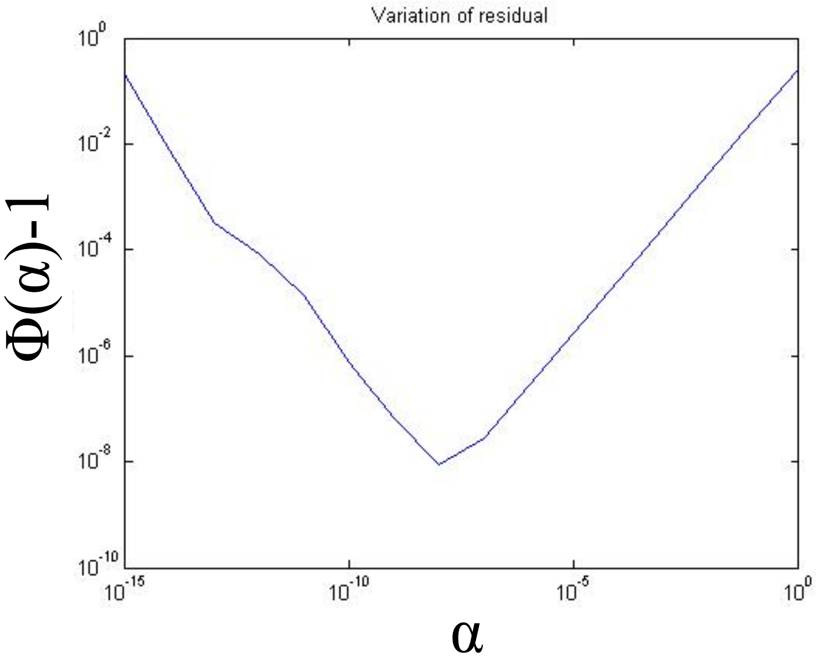
Note that $\mathbf{h}$ should be of unit length, *e.g*.

$$\mathbf{h} = \frac{\nabla J(\mathbf{x})}{\left\| \nabla J(\mathbf{x}) \right\|}$$

Gradient test

Correct gradient code





Exercises

Part II

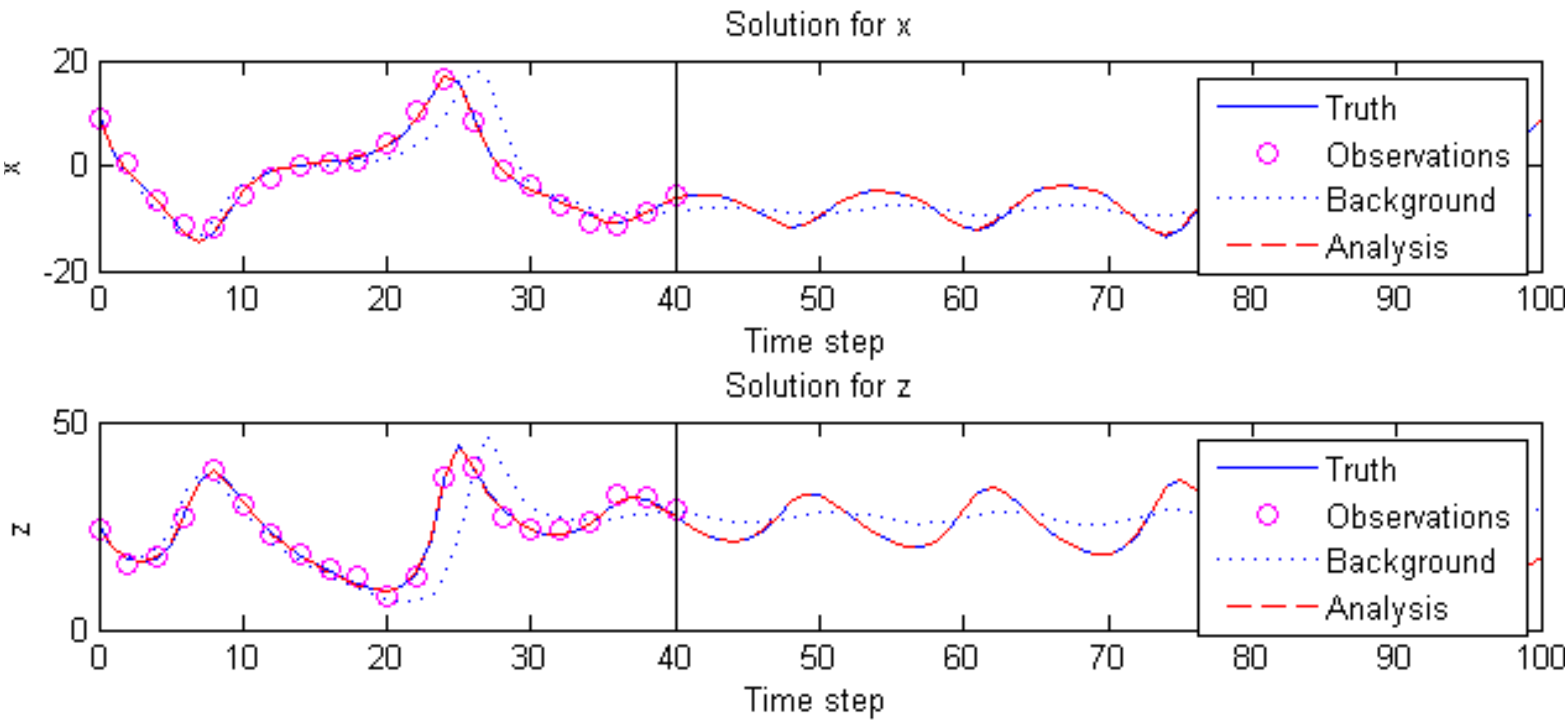
run and compare both

- full 4D-Var *(lorenz4d)*
- incremental 4D-Var *(lorenz4d_inc)*

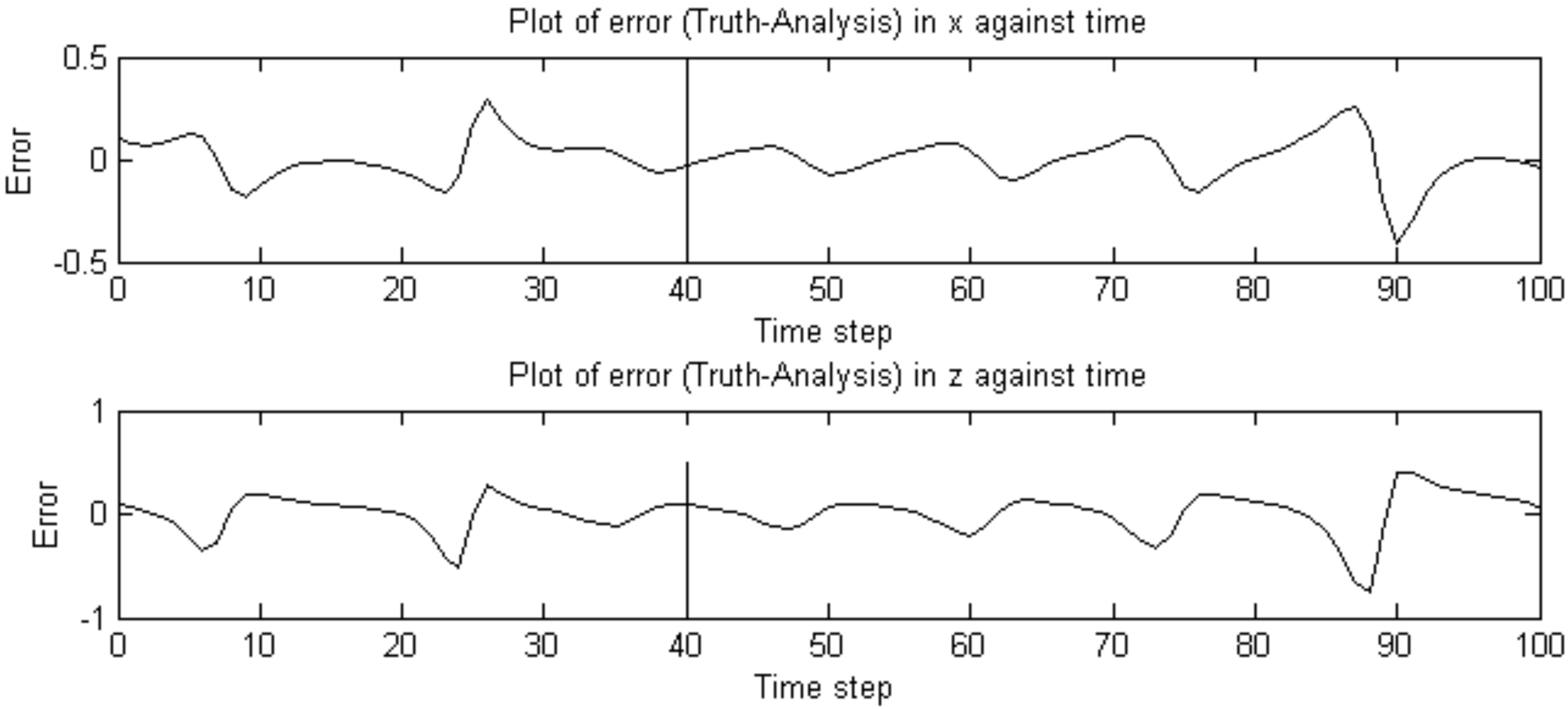
explore system behaviour when changing the input parameters

- consider both accuracy of the analysis and rate of convergence

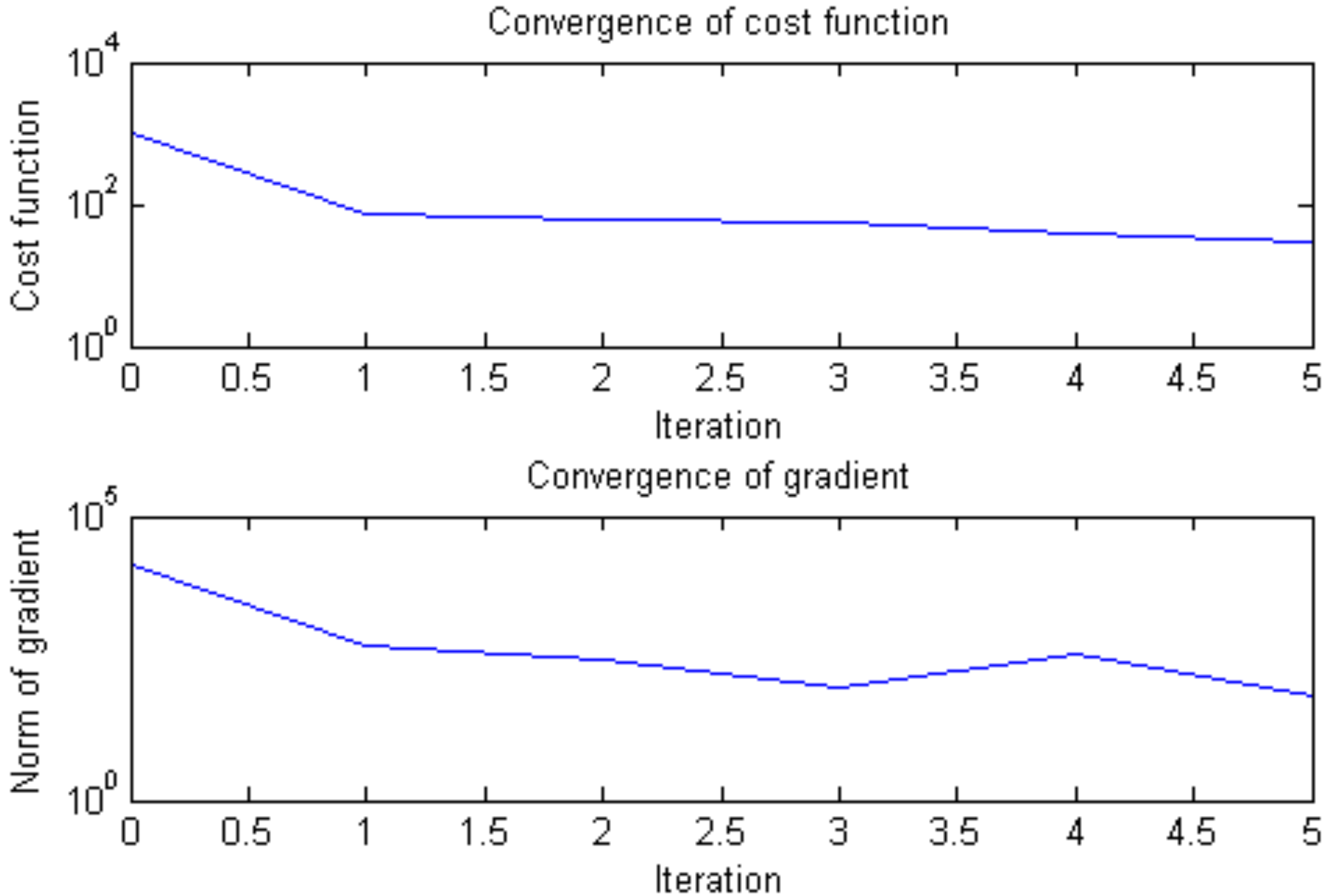
Lorenz 63 model



Lorenz 63 model



Lorenz 63 model



Matlab code

www.met.reading.ac.uk/~darc/training

Take a copy of the original files before you start to change them!