Variational assimilation – Building a 4D-Var system

Amos S. Lawless

Data Assimilation Research Centre
University of Reading

a.s.lawless@reading.ac.uk

@amoslawless

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

University of Reading

4D-Var problem

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

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

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

University of Reading

Minimization using iterative methods

The minimization of the cost function in variational data assimilation usually requires an iterative *gradient method* such as conjugate gradient or quasi-Newton.

These methods need to be able to calculate the cost function and its first derivative with respect to the initial state on each iteration.

In general the user must supply a routine which calculates $J(\mathbf{x_0})$ and $\nabla J\,(\mathbf{x_0})$ for any value of $\mathbf{x_0}$.

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

University of Reading

Calculating the gradient

The gradient is given by solving the adjoint equation backwards in time:

$$\boldsymbol{\lambda}_i = \mathbf{M}_i^T \boldsymbol{\lambda}_{i+1} - \mathbf{H}_i^T \mathbf{R}_i^{-1}(\mathcal{H}_i(\mathbf{x}_i) - \mathbf{y}_i)$$

Then

$$\nabla \mathcal{J}(\mathbf{x}_0) = -\boldsymbol{\lambda}_0 + \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}^b).$$

Notes

- The words *adjoint* and *transpose* are often used interchangeably. In fact the transpose is an adjoint for a particular inner product.
- The use of other inner products is only important if we want to give a physical meaning to adjoint variables.

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

University of Reading

Discrete method

Let us suppose we have the line of code

$$\boxed{z = x*y + y*y}$$

We can linearize these lines of code by putting

$$x = X + \delta x\ , \quad y = Y + \delta y, \quad z = Z + \delta z$$

X, Y, Z are called linearization states.

Then we obtain

$$\boxed{\delta z = X*\delta y + \delta x*Y + 2*Y*\delta y}$$

TLM

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

University of Reading

To obtain the adjoint model we consider the TLM statement as a matrix system in which we also consider δx and δy to be unchanged inputs to the system.

We can write the TLM line of code as a matrix system as follows:

$$\delta z = X*\delta y + \delta x*Y + 2*Y*\delta y$$

TLM

$$\begin{pmatrix} \delta x \\ \delta y \\ \delta z \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ Y & X + 2Y \end{pmatrix} \begin{pmatrix} \delta x \\ \delta y \end{pmatrix}$$

The adjoint model can then be found by transposing this system of equations

ADJ

$$\begin{pmatrix} \delta x'' \\ \delta y'' \end{pmatrix} = \begin{pmatrix} 1 & 0 & Y \\ 0 & 1 & X + 2Y \end{pmatrix} \begin{pmatrix} \delta x'' \\ \delta y'' \\ \delta z'' \end{pmatrix}$$

which implies the adjoint code

δx // = δx// + Y * δz//

δy// = δy// + (X + 2*Y) * δz//

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

University of Reading

We note that this adjoint code can be derived directly from the TLM code, without writing out the matrices

TLM

$$\boxed{\delta z = X*\delta y + \delta x*Y + 2*Y*\delta y}$$

ADJ

$$\delta x^{//} = \delta x^{//} + Y * \delta z^{//}$$

$$\delta y^{//} = \delta y^{//} + (X + 2*Y) * \delta z^{//}$$

We also need to set $\delta z^{//} = 0$

Hence the adjoint code can be developed directly from the TLM code, following some simple rules.

- Set initial values of adjoint variables to zero.
- Work backwards through the TLM code, taking the transpose of each line of code and setting LHS variables to zero.
- Reverse also the order of any loops which depend on the loop order.
- For each line of adjoint code increment the adjoint variables.

Automatic adjoint compilers

The procedure shown in the previous slides is so automatic that it is possible to for 'adjoint compilers' to do it automatically. Such packages will produce a TLM and adjoint model from a nonlinear model source code, *e.g.*

- TAF
- ODYSSEE
- ADIFOR
- Python modules

Testing of a TLM - Correctness

Is the TLM coded correctly?

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

Then by a Taylor series expansion we have

$$M(\mathbf{x}_0 + \gamma\delta\mathbf{x}) = M(\mathbf{x}_0) + \mathbf{M}(\mathbf{x}_0)\,\gamma\delta\mathbf{x} + h.o.t.$$

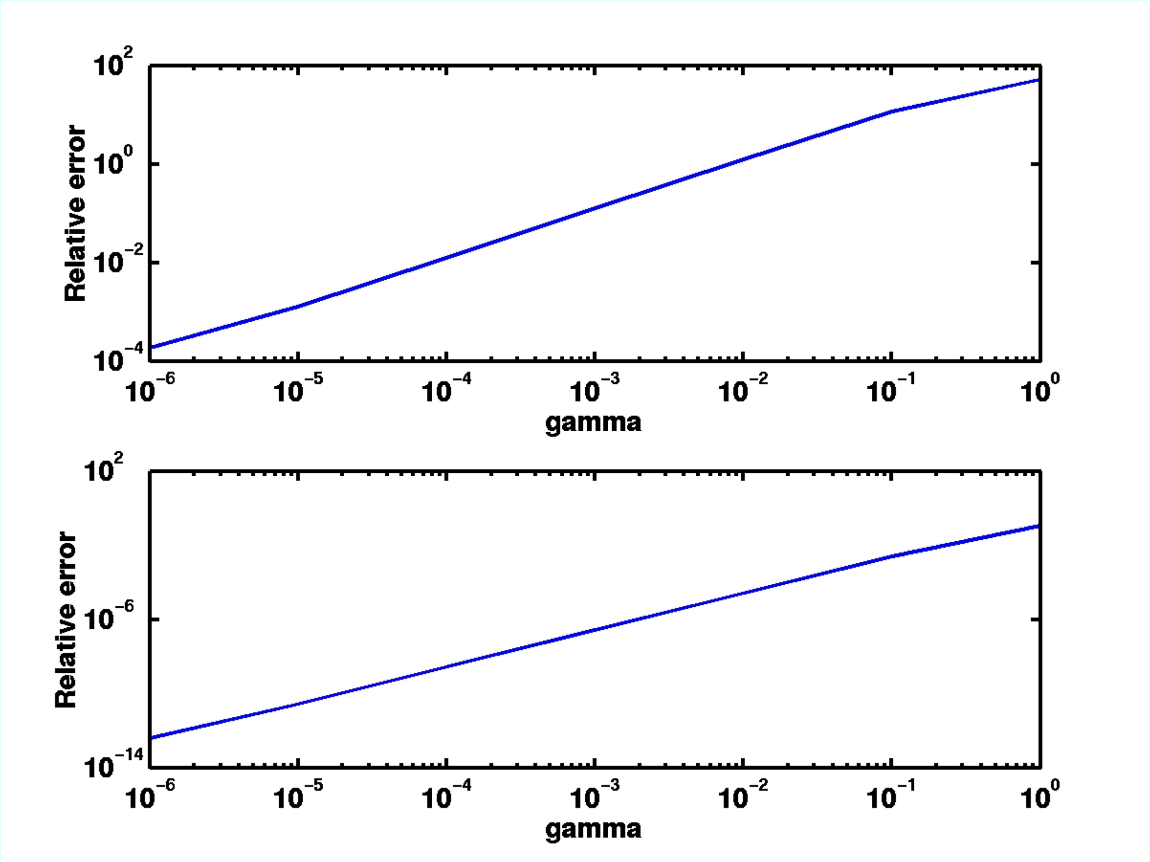
Hence

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

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

University of Reading

Testing of a TLM - Correctness



Testing of a TLM - Validity

Does the linear model provide a good approximation?

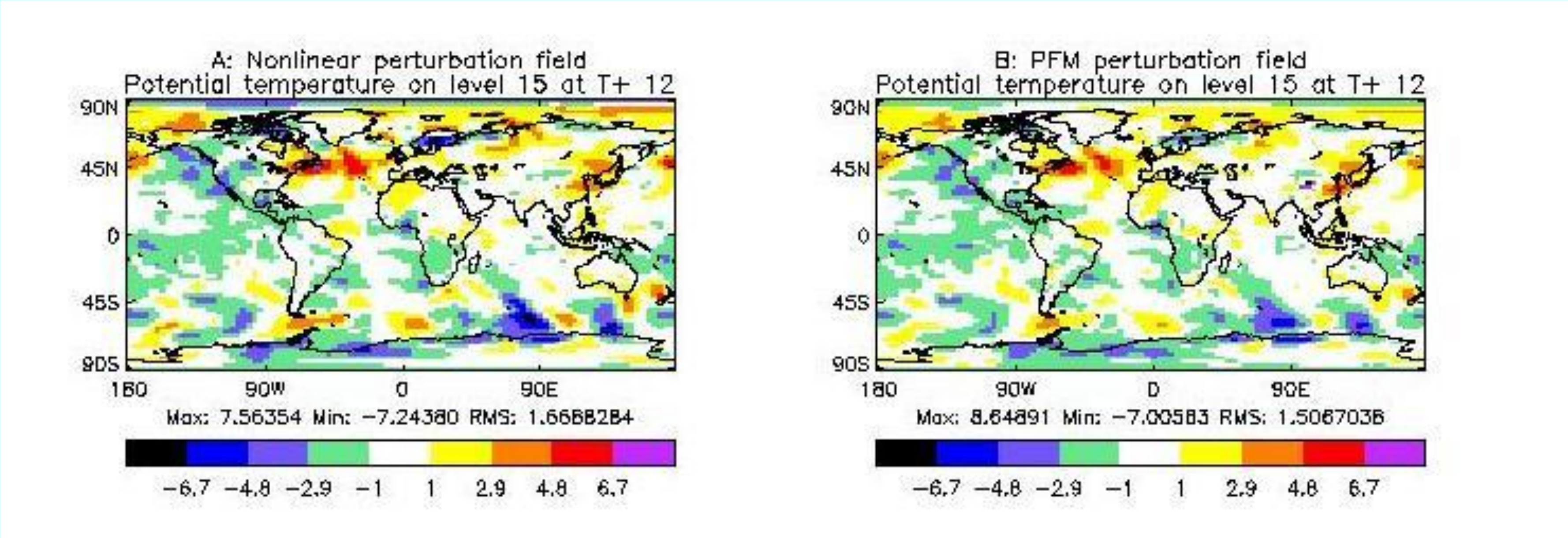
For a *realistic* perturbation $\delta\mathbf{x}$, compare the nonlinear evolution

$$M(\mathbf{x}_0 + \delta\mathbf{x}) - M(\mathbf{x}_0)$$

with the linear evolution $\mathbf{M}(\mathbf{x}_0)\,\delta\mathbf{x}$

Realistic implies a size of the order of analysis error and not dominated by gravity waves.

Testing of a TLM - Validity



National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

University of Reading

Note that validity will depend on

- Size of perturbation
- Time of evolution
- Linearization state
- Application

Test of adjoint model

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

$$(\mathbf{M}\,\delta\mathbf{x}\,,\,\mathbf{M}\,\delta\mathbf{x}) = (\delta\mathbf{x}\,,\,\mathbf{M}^T\mathbf{M}\,\delta\mathbf{x})$$

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 3, to obtain $\mathbf{M}^T\mathbf{M}\,\delta\mathbf{x}$
4. Verify that the above identity is satisfied to machine precision

Summary so far

So far we have been able to

- Code the cost function using the nonlinear model.
- Calculate the tangent linear model from the nonlinear model & test the TLM.
- Calculate the adjoint model from the tangent linear model & test the adjoint.
- Code the gradient of the cost function using the adjoint model.

As a final step we want to test the gradient.

Gradient test

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

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 **h** should be of unit length, *e.g.*

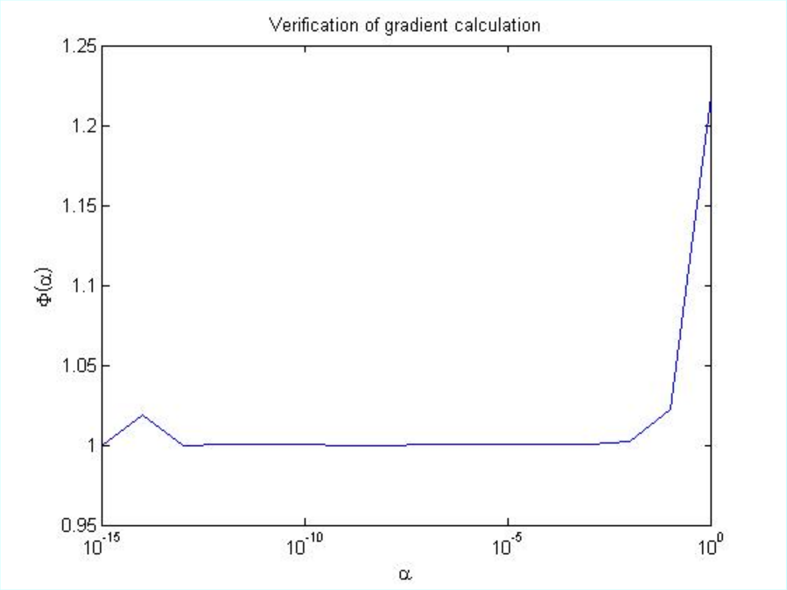
$$\mathbf{h} = \frac{\nabla J(\mathbf{x})}{\|\nabla J(\mathbf{x})\|}$$

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

University of Reading

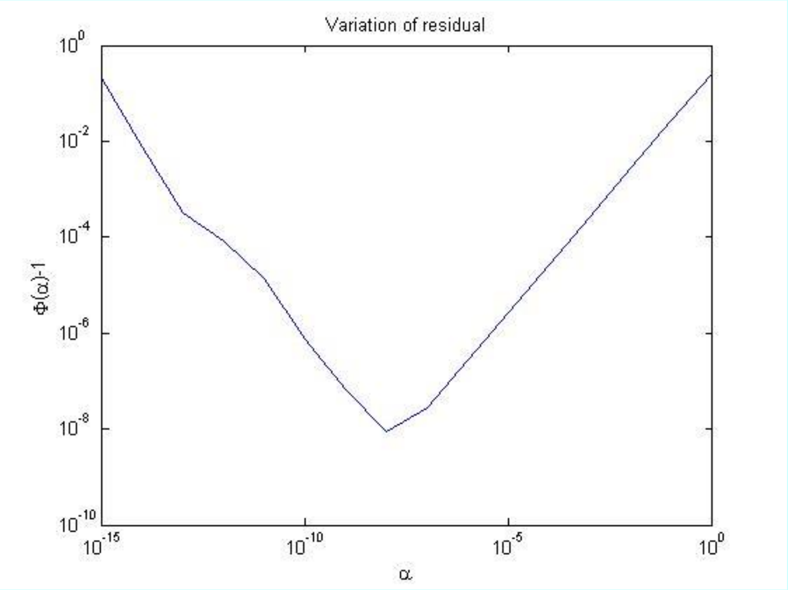
Gradient test

$\Phi(\alpha)$



α

$\Phi(\alpha)$-1



α

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

University of Reading

References

Coding a TLM and adjoint:

W.C. Chao and L-P. Chang (1992), Development of a four-dimensional variational analysis system using the adjoint method at GLA. Part I: Dynamics. *Mon. Wea. Rev.*, 120:1661-1673.

R. Giering and T. Kaminski (1998), Recipes for adjoint code construction. *ACM Trans. On Math. Software*, 24:437-474.

Testing a TLM and adjoint:

Y. Li, I.M. Navon, W. Yang, X. Zou, J.R. Bates, S. Moorthi and R.W. Higgins (1994), Four-dimensional variational data assimilation experiments with a multilevel semi-Lagrangian semi-implicit general circulation model. *Mon. Wea. Rev.*, 122:966-983.

A.S. Lawless, N.K. Nichols and S.P. Ballard (2003), A comparison of two methods for developing the linearization of a shallow-water model, *Quart. J. Roy. Met. Soc.*, 129:1237-1254.