Data Assimilation in Chaotic Systems

Mark S. Roulston

1 Introduction

Data assimilation is a crucial component of numerical weather prediction (NWP) schemes. It is the method by which the initial conditions for the model are determined. NWP models are obviously not perfect models of the atmosphere but if they were then the accuracy of the initial conditions would be the sole factor determining the forecast skill of the model.

Data assimilation involves using time series of observations over the recent past to estimate the state of the system at the present moment. Simply using the observations taken at the present moment is not enough because firstly these observations will contain errors and secondly these observations are incomplete. That is, there is not an observation of every relevant variable at every model gridpoint.

Data assimilation methods can be placed into two categories. Sequential methods such as *nudging* integrate the model forward in time with extra non-physical forcing terms which push the model towards the observations. If the model is linear (which NWP models are not) then it can be shown that the optimal nudging scheme takes the form of a *Kalman filter*. Variational methods seek to minimize a cost function which is the measure of the misfit between the model and observations over the assimilation period. Variational methods are regarded as the "next generation" of assimilation schemes and they are now used operationally.

An important distinction between sequential schemes and the variational method is that the model is used as a strong constraint and thus the trajectory that results from assimilation (called the analysis) must be a valid trajectory of the model. This is not so in sequential methods in which observations are essentially blended with the model. This means that if the aim of assimilation is to validate the model the variational method is to be preferred. In this paper only variational based assimilation schemes will be considered.

In section 2 the adjoint method will be described. The Lorenz model will then be used to illustrate the problem of multiple minima when assimilating into strongly nonlinear models. The problem of data assimilation in strongly nonlinear systems has been addressed by several researchers [Stensrud and Bao, 1992, Miller et al., 1994, Pires et al., 1996, Evensen, 1997]. As with this previous work the Lorenz model will be used as a simple example. However, the previous work has assumed that the system and the model are identical. In this work the case when the model is not an exact copy of the system will also be considered. The Quasi Static Variational Algorithm (QSVA) [Pires et al., 1996] will be explained and simple but effective improvement to this method will be outlined. The modified and standard QSVA algorithms will be compared in section 5.

2 Variational data assimilation and the adjoint method

As the name suggests variational data assimilation uses a variational method to find the model state at time $t = 0$ which leads to the best fit between the model and observations over the period $0 < t < T$. "Best fit" is usually means the minimum mean square error but other cost functions (or *objective* functions) can be used. The computational efficiency of the variational method is greatly aided by the existence of a means to calculate the gradient of the cost function in the space of initial conditions which is only slightly more computationally expensive than calculating the cost function itself. This is done by integrating the model equations forward in time and then integrating the *adjoint* version of the model backwards in time.

Consider a time-discrete dynamical model, $\mathbf{M}$, which can be written as

$$\mathbf{x}_{t+1} = \mathbf{M}(\mathbf{p}, \mathbf{x}_t) \tag{1}$$

where $\mathbf{p}$ is a vector of model parameters and $\mathbf{x}_t$ is the state of the model at time t. An *objective* function, J, which is a measure of the misfit between the model and the actual observations is now introduced. The goal is to minimize the objective function, J, under the constraint given by Eq. 1. This can be done using the method of undetermined multipliers. Thus the function that must be minimized is

$$L = J + \sum_{t=0}^{T} \lambda_t'(\mathbf{x}_{t+1} - \mathbf{M}(\mathbf{p}, \mathbf{x}_t)) \tag{2}$$

where λ_t is the vector of undetermined multipiers and a prime denotes the transpose. Differentiation of Eq. 2 with respect to λ_t recovers the original model equation. Differentiation of Eq. 2 with respect to $\mathbf{x}_t$ gives

$$\frac{\partial J}{\partial \mathbf{x}_t} - \left(\frac{\partial \mathbf{M}(\mathbf{p}, \mathbf{x}_t)}{\partial \mathbf{x}_t}\right)' \lambda_t + \lambda_{t-1} \tag{3}$$

Eq. 3 can be rearranged to give

$$\lambda_{t-1} = \left(\frac{\partial \mathbf{M}(\mathbf{p}, \mathbf{x}_t)}{\partial \mathbf{x}_t}\right)' \lambda_t + \mathbf{G}_t \tag{4}$$

where

$$\mathbf{G}_t = -\frac{\partial J}{\partial \mathbf{x}_t} \tag{5}$$

Eq. 4 is sometimes called the *adjoint* equation. Notice that it can be solved by integrating it backwards in time forced by the forcing function $\mathbf{G}_t$e The initial condition for λ is $\lambda_N = \mathbf{0}$. It will be shown below that this initial condition will result in the vector λ_0 being equal to the negative of the gradient of the objective function with respect to the initial condition, $\mathbf{x}_0$. First J is differentiated with respect to the initial condition, $\mathbf{x}_0$, to obtain

$$\frac{\partial J}{\partial \mathbf{x}_0} = -\sum_{t=0}^{T} \left(\frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_0}\right)' \mathbf{G}_t \tag{6}$$

The matrix of partial derivatives in Eq. 6 must now be evaluated. This can be done by induction as follows.

$$\frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_0} = \frac{\partial \mathbf{M}(\mathbf{p}, \mathbf{x}_{t-1})}{\partial \mathbf{x}_0} = \frac{\partial \mathbf{M}(\mathbf{p}, \mathbf{x}_{t-1})}{\partial \mathbf{x}_{t-1}} \frac{\partial \mathbf{x}_{t-1}}{\partial \mathbf{x}_0} \tag{7}$$

Taking the transpose of Eq. 7 gives

$$\left(\frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_0}\right)' = \left(\frac{\partial \mathbf{x}_{t-1}}{\partial \mathbf{x}_0}\right)' \left(\frac{\partial \mathbf{M}(\mathbf{p}, \mathbf{x}_{t-1})}{\partial \mathbf{x}_{t-1}}\right)' = \left(\frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_0}\right)' \mathcal{M}'_{t-1} \tag{8}$$

In the last step the matrix $\mathcal{M}_t$ has been introduced for convenience. This matrix is defined as

$$\mathcal{M}'_t = \left(\frac{\partial \mathbf{M}(\mathbf{p}, \mathbf{x}_t)}{\partial \mathbf{x}_t}\right)' \tag{9}$$

Substitution of Eq. 8 into Eq. 6 gives

$$\frac{\partial J}{\partial \mathbf{x}_0} = -\sum_{t=0}^{T} \mathcal{M}'_1 \mathcal{M}'_2 ... \mathcal{M}'_{t-1} \mathbf{G}_t \tag{10}$$

It will now be shown that λ_0 is equal to the negative of the expression given in Eq. 10. First Eq. 4 is rewritten as

$$\lambda_{t-1} = \mathcal{M}'_t \lambda_t + \mathbf{G}_t \tag{11}$$

Thus λ_{t-2} can be expressed as

$$\lambda_{t-2} = \mathcal{M}'_{t-1} \lambda_{t-1} + \mathbf{G}_t \tag{12}$$

Substituting for λ_{t-1} in Eq. 12 using Eq. 11 gives

$$\lambda_{t-2} = \mathcal{M}'_{t-1}(\mathcal{M}'_t \lambda_t + \mathbf{G}_t) + \mathbf{G}_{t-1} \tag{13}$$

This process can be continued to obtain an expression for λ_0.

$$\lambda_0 = \sum_{t=0}^{T} \mathcal{M}'_1 \mathcal{M}'_2 ... \mathcal{M}'_{t-1} \mathbf{G}_t = -\frac{\partial J}{\partial \mathbf{x}_0} \tag{14}$$

where the last step can be made by comparison with Eq. 10. Once the undetermined multipliers have been determined the gradient of the cost function with respect to the parameters, $\mathbf{p}$, can also be calculated.

$$\frac{\partial L}{\partial \mathbf{p}} = -\sum_{t=0}^{T} \lambda'_t \frac{\partial \mathbf{M}(\mathbf{p}, \mathbf{x}_t)}{\partial \mathbf{p}} \tag{15}$$

The gradients with respect to the initial conditions and the parameters can be combined to give $\nabla J = [-\lambda_{\mathbf{0}}, \partial L / \partial \mathbf{p}]$ enabling the best combination of parameters and initial condition to be found. For this work the optimization was performed with an iterative Newton method from the IDL package. However, this requires the storage of the Hessian matrix. Real NWP models have around 10^4 variables and thus the Hessian matrix is unfeasibly large. Conjugate-gradient methods are usually employed for such large models [Navon and Legler, 1987].

In the above derivation a general cost function was used. A specific cost function, namely the mean square error, will now be considered. Let $\mathbf{H}$ be the *observation matrix* which relates the state of the system, $\mathbf{x}$, to the observations that would be made if the system were in the state $\mathbf{x}$. That is, if the system is in state $\mathbf{x}$ the vector of observations is given by $\mathbf{Hx}$. If $\mathbf{y}_t$ is the vector of actual observations at time t then the mean square error cost function is given by

$$J = \frac{1}{2}\sum_{i=0}^{T}(\mathbf{y}_i - \mathbf{H}\mathbf{x}_i)'(\mathbf{y}_i - \mathbf{H}\mathbf{x}_i) \tag{16}$$

Thus $\mathbf{G}_t$ is given by

$$\mathbf{G}_t = -\frac{\partial J}{\partial \mathbf{x}_t} = \mathbf{H}'(\mathbf{y}_t - \mathbf{H}\mathbf{x}_t) \tag{17}$$

To keep computational expense down when the adjoint method is used with large NWP models the forward model described by Eq. 1 is linearized about the basic trajectory. If the basic trajectory is written as $\bar{\mathbf{x}}_t$ and the deviation from this trajectory is written as $\mathbf{x}'_t$ then the linearization of Eq. 1 can be written as

$$\mathbf{x}'_{t+1} = \mathbf{L}(\bar{\mathbf{x}}_t)\mathbf{x}'_t \tag{18}$$

The linear model $\mathbf{L}$ is referred to as the *Tangent Linear Model* or TLM. The adjoint model given by Eq. 4 is also linearized about the trajectory $\bar{\mathbf{x}}_t$. The linearized adjoint model will be referred to as the ADJM.

3 The Lorenz model

As with previous work in this area the Lorenz model will be used as toy nonlinear system with which to test ideas.

The equations of the Lorenz model can be written in matrix form as

$$\frac{dX}{dt} = -\sigma(X - Y) \tag{19}$$

$$\frac{dY}{dt} = rX - Y - XZ \tag{20}$$

$$\frac{dZ}{dt} = XY - bZ \tag{21}$$

where σ, r and b are parameters of the model. The Lorenz model can be written in discrete form as

$$\begin{bmatrix} X_{t+1} \\ Y_{t+1} \\ Z_{t+1} \end{bmatrix} = \begin{bmatrix} 1-\sigma\tau & +\sigma\tau & 0 \\ r\tau & 1-\tau & -X_t\tau \\ Y_t\tau & 0 & 1-b\tau \end{bmatrix} \begin{bmatrix} X_t \\ Y_t \\ Z_t \end{bmatrix} \tag{22}$$

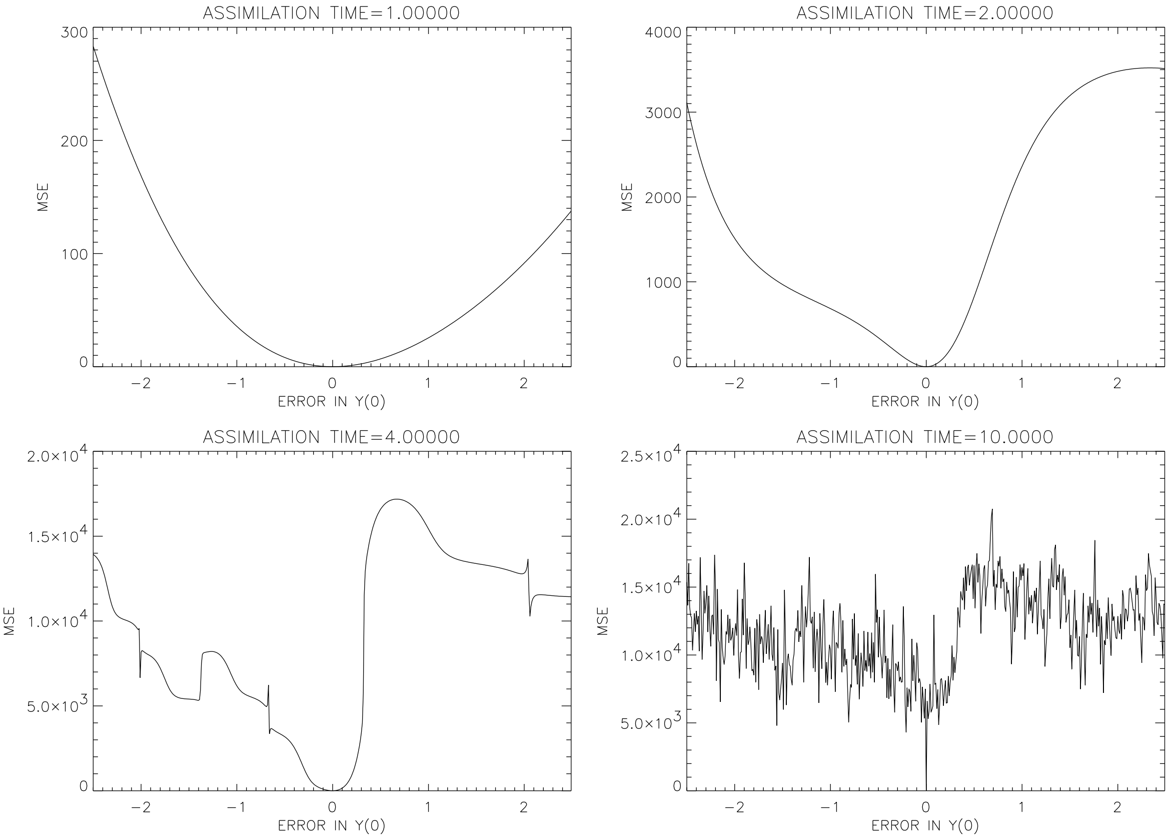

Figure 1: The MSE cost function in the Lorenz model as a function of error in the initial value of the Y coordinate. The function becomes increasingly pathological as the assimilation period is increased.

where τ is the length of the time step used in integrating the model. The TLM of Eq. 22 is given by

$$\begin{bmatrix} X'_{t+1} \\ Y'_{t+1} \\ Z'_{t+1} \end{bmatrix} = \begin{bmatrix} 1-\sigma\tau & +\sigma\tau & 0 \\ (r-\bar{Z}_t)\tau & 1-\tau & 1-\bar{X}_t\tau \\ \bar{Y}_t\tau & \bar{X}_t\tau & 1-b\tau \end{bmatrix} \begin{bmatrix} X'_t \\ Y'_t \\ Z'_t \end{bmatrix} \tag{23}$$

Where an overbar denotes the trajectory about which the model is linearized and a prime denotes the deviation from this trajectory.

Using Eq. 9 the adjoint form of Eq. 22 is

$$\mathcal{M}'_t = \begin{bmatrix} 1-\sigma\tau & (r-Z_t)\tau & Y_t\tau \\ \sigma\tau & 1-\tau & X_t\tau \\ 0 & -X_t\tau & 1-b\tau \end{bmatrix} \tag{24}$$

To illustrate a major problem in variational assimilation in chaotic systems a set of "observations" was generated using the Lorenz model by running it with the standard parameters ($\sigma = 10$,$r = 28$,$b = 8/3$) and taking the X component only as the observed variable, this corresponds to $\mathbf{H} = [1, 0, 0]$. The MSE cost function defined in Eq. 16 was then calculated as a function of error in the initial value of the Y coordinate. The results when the cost function was evaluated over periods of 1, 2, 4 and 10 dimensionless time units are shown in Fig. 1. For short periods the cost function is smooth and has a single, global minimum in the vicinity of the true initial condition. However, as the time period is increased local minima appear and the cost function becomes pathological for longer assimilation periods. Clearly finding the global minimum of the cost function when the assimilation time is 10 units is a daunting task. There are methods, such as simulated annealing and genetic algorithms, which could be tried but exploiting the option of controlling the cost function would seem to be the best line of attack.

4 Quasi Static Variational Assimilation

Quasi Static Variational Assimilation (QSVA) was introduced by Pires *et al.* [Pires et al., 1996].

The QSVA algorithm can be outlined as follows:-

1. Start with an initial guess of the state of the system and let the initial assimilation time be $n = \Delta$.
2. Minimize the MSE over the assimilation period $t = 0$ to $t = n$ to produce a new estimate of the state of the system at time $t = 0$.
3. Increase n to $n + \Delta$ and repeat step 2 until $n = T$.

The aim of the QSVA algorithm is to keep the estimate of the state of the system at time $t = 0$ in the basin of the global minimum, even as the local minima appear. In this algorithm Δ is a preset parameter. How should Δ be chosen? In their experiments with the Lorenz model Pires *et al.* used a value of $\Delta = 0.2/\tau$ where τ is the model time step. The most efficient value of Δ will depend on the particular system under consideration and also the region of phase space which the system is in at the time. In some regions of their phase space nonlinear systems can be well approximated by linear models and in these regions assimilation can be performed over relatively long periods. In other regions the assimilation period must be increased more slowly to remain in the basin of the global minimum. Thus a simple modification of the QSVA algorithm is proposed. In the modified algorithm Δ can be adapted so this modified algorithm will be refered to as *Adaptive Quasi Static Variational Assimilation* or AQSVA. The criteria for choosing the assimilation period will be that the model can "shadow" the observations over the assimilation period. The shadowing time T_s will be taken to mean that $||\mathbf{H}\mathbf{x}_t - \mathbf{y}_t|| < \varepsilon$ for all $t \leq T_s$.

The AQSVA algorithm is as follows:-

1. Start with an initial guess of the state of the system. Integrate the model until either

Perfect model/noise free	QSVA	AQSVA	Standard
Computational expense	36316	16460	41300
Shadowing time	5.00	5.00	0.42
Error in $\mathbf{x}_0$	10^{-5}	3×10^{-4}	4.3

Table 1: Perfect model without observational noise.

$t = T$ in which case $T_s = T$ or until $||\mathbf{H}\mathbf{x}_t - \mathbf{y}_t|| \geq \varepsilon$ in which case T_s is the shadowing time.
2. Minimize the MSE over the period $t = 0$ to $t = T_s$ to produce a new estimate of the state of the system at time $t = 0$.
3. Unless $T_s = T$ or T_s is no longer increasing return to step 1.

The parameter ε must be chosen. It should be large enough to allow for experimental noise but it should be small enough such that the tangent linear model is valid. Since the TLM has a single quadratic minimum in the MSE cost function.

5 A comparison of QSVA and AQSVA

A series of numerical experiments was performed to compare the accuracy and efficiency of the standard QSVA algorithm and the modified AQSVA version. The Lorenz equations with parameters $\sigma = 10$, $r = 28$ and $b = 8/3$ were used as the system. The Lorenz equations were also used as the model into which the observations were assimilated. A perfect model and imperfect model were considered. Perfect model refers to the case when the parameters of the model were the same as those for the system while in the imperfect case the value of r in the model was changed from 28 to 30. In all the experiments only the X variable of the system was "observed". In some of the experiments noise was added to the observed values of X. This noise was uncorrelated and Gaussian with a standard deviation of 0.2. The full assimilation period in each experiment was 5.00 time units.

The results of the experiments are shown in Tables 1-4. Each table was constructed by averaging the results of 10 experiments with different initial conditions. All the initial conditions lay on the attractor of the system. As well as the QSVA and AQSVA algorithms standard MSE minimization was also included in the experiments. That is when an attempt was made to minimize MSE over the entire assimilation period from the initial first guess. The computational expense of each of the methods was measured in terms of integration steps. That is, forward integration steps of the TLM plus backward integration steps of the ADJM.

In the case of perfect models the accuracy of the AQSVA algorithm is comparable with the QSVA algorithm. Although the accuracy of the final estimate for the initial condition is slightly worse when using AQSVA this inaccuracy is much less than the inaccuracy in this estimate caused by adding noise to the observations. Both algorithms can find model trajectories that shadow the observations to the end of the assimilation period with the exception of one of the ten AQSVA cases without noise. Standard MSE minimization always performs very poorly. This isn't suprising given the fact that for an assimilation period of 5 time units the cost

Perfect model/noise s.d. = 0.2	QSVA	AQSVA	Standard
Computational expense	54912	17674	42700
Shadowing time	5.00	4.94	1.17
Error in $\mathbf{x}_0$	0.12	0.12	5.8

Table 2: Perfect model with observational noise.

Imperfect model/no noise	QSVA	AQSVA	Standard
Computational expense	109940	25402	46000
Shadowing time	0.11	1.58	0.25
Error in $\mathbf{x}_0$	9.28	2.60	3.90

Table 3: Imperfect model without observational noise.

function has many local minima in which the optimization algorithm can become trapped. The most important difference between QSVA and AQSVA is the computational expense. AQSVA is typically about half as expensive as QSVA.

For the imperfect model making a direct comparison between the QSVA and AQSVA algorithms is complicated by the fact that they are essentially attempting to do different things. While the QSVA algorithm is trying to minimize the MSE over the entire assimilation period the AQSVA algorithm stops as soon as shadowing breaks down. This difference is reflected in the shadowing times which the algorithms achieve. QSVA shadowing times are never very long because in its efforts to fit the model to observations at later times the algorithm sacrifices a good fit for earlier times. The AQSVA algorithm does not make this sacrifice and it thus achieves shadowing times an order of magnitude longer than QSVA. AQSVA also obtains better estimates for the state of the system at the beginning of the assimilation period. Again AQSVA is several times cheaper computationally than QSVA. In the imperfect model case the computational saving of AQSVA is even higher than for the perfect model because AQSVA never assimilates to the end of the assimilation period.

The reason for the cheapness of AQSVA compared to QSVA can be explained as follows. In some parts of phase space the model is close to linear and thus MSE can be minimized over relatively large time periods without secondary minima becoming a problem. However, in QSVA the value of Δ is determined by the regions of phase space where the model is most

Imperfect model/noise s.d. =0.2	QSVA	AQSVA	Standard
Computational expense	125028	15683	48400
Shadowing time	0.11	1.39	0.30
Error in $\mathbf{x}_0$	27.6	2.66	4.34

Table 4: Imperfect model with observational noise.

nonlinear and where the assimilation period must be increased slowly. If this limiting value of Δ is used in all regions of phase space the algorithm is less efficient than AQSVA where the value of Δ can be varied, taking a large value when the model is well approximated by the TLM and a small value when it is not.

When data assimilation is being used to obtain an estimate of the state vector at time $t = T$ to initialize a forecast it is of no use if the AQSVA algorithm stops short of $t = T$. In this case the AQSVA can be run from the point when shadowing breaks down, using the estimate of the state vector at $t = T_s$ as the initial estimate for the next run of the AQSVA algorithm. This procedure can be repeated until a sequence of discontinuous trajectories has been constructed which shadow the observations for the complete assimilation period.

6 AQSVA in a system with dynamical noise

In the previous section imperfect model was taken to mean that the parameters of the model were not the same of those of the system. Another way in which a model can be imperfect is by lacking extrinsic forcing terms which are present in the system. To investigate this effect stochastic forcing terms were added to the Lorenz equations of the system. Thus the system became

$$\frac{dX}{dt} = -\sigma(X - Y) + W_X(t) \tag{25}$$

$$\frac{dY}{dt} = rX - Y - XZ + W_Y(t) \tag{26}$$

$$\frac{dZ}{dt} = XY - bZ + W_Z(t) \tag{27}$$

where $W_{X,Y,Z}$ represent uncorrelated gaussian noise with zero mean and variance $w^2 = 25.0$. Again the X coordinate of the system was the observed variable. The AQSVA algorithm was then used to find a shadowing trajectory of the model which had the same parameters as the system but lacked the stochastic forcing terms. A typical result is shown in Fig. 2.

The thin solid line is the X coordinate of the system with stochastic forcing. The dashed line is the X coordinate of the model when the actual initial condition is used. Note that in this case the trajectoroy of the model deviates from the observations after about 3 time units. The thicker solid line is the trajectory found using the AQSVA algorithm. This trajectory shadows the stochastically forced system for about 6 time units even though the forcing terms were missing. The reason for this is that in the Lorenz system the relatively small forcing terms are only important when the system is close to a decision point in phase space. At such a point a small perturbation can determine which of the two unstable fixed points the system will orbit. A slight change in the state of the system at $t = 0$ can “fake” the effect of this forcing.

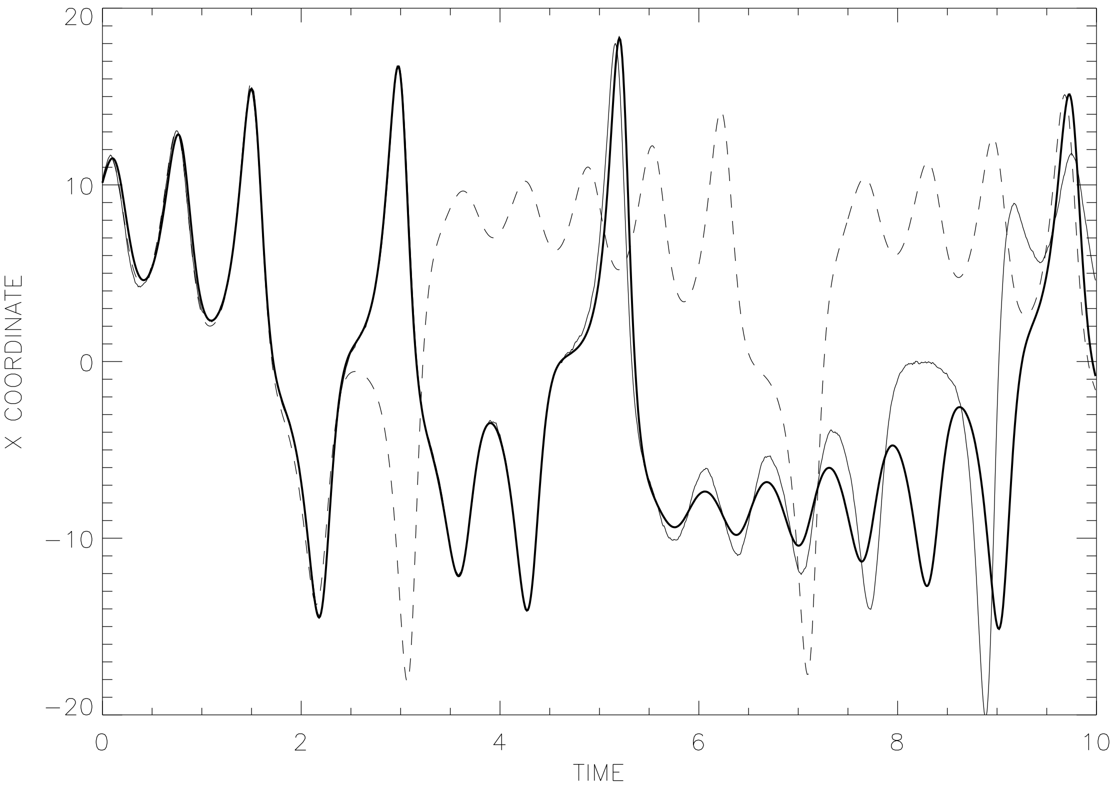

Figure 2: Assimilation in the presence of dynamical noise. The thin solid line is the X coordinate of the system with dynamical noise. The dashed line is the X coordinate of the model with identical parameters to the system and initialized with the same initial state but without the stochastic forcing terms. The thick solid line (which overlies the thin solid line until about 6 time units) is the X coordinate of the model without the dynamical forcing terms after the thin solid line was assimilated into it.

7 An alternative cost function

As described above one of the main problems with the MSE cost function in strongly nonlinear systems is the existence of many local minima in addition to the global minimum. Perhaps an alternative approach to the problem is to use a different cost function. The length of time for which the model can shadow the observations would seem to be a reasonable measure of the validity of the model. The shadowing time, T_s, can be written as follows.

$$T_s = \sum_{i=0}^{T} \prod_{k=0}^{i} \mathcal{H}(\varepsilon^2 - r_k^2) \tag{28}$$

where $\mathcal{H}$ is the Heaviside step function and $r_k = ||\mathbf{y}_k - \mathbf{H}\mathbf{x}_k||$.

Note that the step function is zero if the distance between the model and observations exceeds ε. Also note that each term in the sum over time contains a product of all the step functions up to that time, thus if any term is zero all the terms that follow it are zero. Equation 28 does not have a continuous derivative which can be used to force the adjoint equation. The step function can be replaced by a smooth approximation of a step function, $\mathcal{K}$. A *pseudo-shadowing time* can now be defined as

$$T_s^* = \sum_{i=0}^{T} \prod_{k=0}^{i} \mathcal{K}(\varepsilon^2 - r_k^2) \tag{29}$$

If the cost function is defined as $J = -T_s^*$ the forcing term for the adjoint equation is given by

$$\mathbf{G}_t = -\frac{\partial J}{\partial \mathbf{x}_t} = \frac{2\mathbf{H}'(\mathbf{y}_t - \mathbf{H}\mathbf{x}_t)}{\mathcal{K}(\varepsilon^2 - r_t^2)} \left[\frac{d\mathcal{K}}{dr_t^2}\right]_{\varepsilon^2 - r_t^2} \sum_{i=t}^{T} \prod_{k=0}^{i} \mathcal{K}(\varepsilon^2 - r_k^2) \tag{30}$$

A possible form of the function $\mathcal{K}$ is

$$\mathcal{K}(z) = \left\{ \begin{array}{ll} 0 & z < -\beta \\ -\frac{1}{4\beta^3} z^3 + \frac{3}{4\beta} z + \frac{1}{2} & -\beta \leq z \leq +\beta \\ 1 & z > +\beta \end{array} \right\} \tag{31}$$

In numerical experiments using the Lorenz model the alternative cost function described by Eqs. 30 and 31 did not produce estimates of the initial state as accurately as AQSVA. This may be because the model used was always very similar to the system, even in the “imperfect” model experiments. The pseudo-shadowing cost function may be an effective way to maximize shadowing time in models which are not structurally the same as the systems they are modeling.

8 Summary

The QSVA algorithm introduced by Pires *et al.* [Pires et al., 1996] has been slightly modified to include adaptive increments in the assimilation time determined by the models ability to shadow observations. The adaptive QSVA (AQSVA) algorithm was compared with the conventional QSVA using the Lorenz equations as a simple system-model combination. The AQSVA method was found to be more computationally efficient than standard QSVA.

An alternative to the MSE cost function based on shadowing time was proposed although it was not determined what practical advantages this *pseudo-shadowing* cost function may have over AQSVA.

ACKNOWLEDGEMENTS

I would like to thank Lenny Smith who was my principal advisor for this project as well as Mike Shelley, Rupert Ford, Andy Ingersoll and Phil Morrison for helpful discussions. I would also like to thank Neil Balmforth and everyone in the GFD program for creating the environment that spawned this research. And if Neil reads this before August 31 I'll buy him a beer.

References

[Evensen, 1997] Evensen, G. (1997). Advanced data assimilation for strongly nonlinear dynamics. *Mon. Wea. Rev.*, 125:1342–1354.

[Miller et al., 1994] Miller, R. N., Ghil, M., and Gauthiez, F. (1994). Advanced data assimilation in strongly nonlinear dynamical systems. *J. Atmos. Sci.*, 51:1037–1056.

[Navon and Legler, 1987] Navon, I. and Legler, D. (1987). Conjugate-gradient methods for large-scale minimization in meteorology. *Mon. Wea. Rev.*, 115:1479–1502.

[Pires et al., 1996] Pires, C., Vautard, R., and Talagrand, O. (1996). On extending the limits of variational assimilation in nonlinear chaotic systems. *Tellus A*, 48:96–121.

[Stensrud and Bao, 1992] Stensrud, D. and Bao, J.-W. (1992). Behaviors of variational and nudging assimilation techniques with a chaotic low-order model. *Mon. Wea. Rev.*, 120:3016–3028.