6

Nonlinear variational inverse problems

This chapter considers highly nonlinear variational inverse problems and their properties. More general inverse formulations for nonlinear dynamical models will be treated extensively in the following chapters, but an introduction is in place here. The focus will be on some highly nonlinear problems which cannot easily be solved using the representer method. Examples are given were instead, so-called direct minimization methods are used.

6.1 Extension to nonlinear dynamics

It was pointed out in the previous chapter that, rather than solving one nonlinear inverse problem, one may define a convergent sequence of linear iterates for the nonlinear model equation, and then solve a linear inverse problem for each iterate using the representer method.

On the other hand, it is also possible to define a variational inverse problem for a nonlinear model. As an example, when starting from the system (5.21–5.23) but with the right-hand-side of (5.21) replaced by a nonlinear function, $G(\psi)$, we obtain Euler–Lagrange equations on the form

$$\frac{d\psi}{dt} - G(\psi) = \int_0^T C_{qq}(t,t_1)\lambda(t_1)\,dt_1, \tag{6.1}$$

$$\psi(0) = \Psi_0 + C_{aa}\lambda(0), \tag{6.2}$$

$$\frac{d\lambda}{dt} + G^*(\psi)\lambda = -\boldsymbol{\mathcal{M}}_{(2)}^{\mathrm{T}}\big[\delta(t-t_2)\big]\boldsymbol{W}_{\epsilon\epsilon}\Big(\boldsymbol{d} - \boldsymbol{\mathcal{M}}\big[\psi\big]\Big), \tag{6.3}$$

$$\lambda(T) = 0, \tag{6.4}$$

where $G^*(\psi)$ is the transpose of the tangent linear operator of $G(\psi)$ evaluated at ψ. Thus, like in the EKF we need to use linearized model operators, but this time for the backward or adjoint equation. We can expect that this may lead to similar problems as was found using the EKF.

Note that, for nonlinear dynamics the adjoint operator (or adjoint equation) does not exist, since the penalty function no longer defines an inner product for a Hilbert space. This is resolved by instead using the adjoint of the tangent linear operator.

In the following we will consider a variational inverse problem for the highly nonlinear and chaotic Lorenz equations and use this to illustrate typical problems that may show up when working with nonlinear dynamics.

6.1.1 Generalized inverse for the Lorenz equations

Several publications have examined assimilation methods with chaotic and unstable dynamics. In particular, the Lorenz model (*Lorenz*, 1963) has been examined with many different assimilation methods. Results have been used to suggest properties and possibilities of the methods for applications with oceanic and atmospheric models which may also be strongly nonlinear and chaotic.

The Lorenz model is a system of three first order coupled and nonlinear differential equations for the variables x, y and z,

$$\frac{dx}{dt} = \sigma(y - x) + q_x, \tag{6.5}$$

$$\frac{dy}{dt} = \rho x - y - xz + q_y, \tag{6.6}$$

$$\frac{dz}{dt} = xy - \beta z + q_z, \tag{6.7}$$

with initial conditions

$$x(0) = x_0 + a_x, \tag{6.8}$$

$$y(0) = y_0 + a_y, \tag{6.9}$$

$$z(0) = z_0 + a_z. \tag{6.10}$$

Here $x(t)$, $y(t)$ and $z(t)$ are the dependent variables and we have chosen the following commonly used values for the parameters in the equations: $\sigma = 10$, $\rho = 28$ and $\beta = 8/3$. We have also defined the error terms $\boldsymbol{q}(t)^{\mathrm{T}} = (q_x(t), q_y(t), q_z(t))$ and $\boldsymbol{a}^{\mathrm{T}} = (a_x, a_y, a_z)$ which have error statistics described by the 3×3 error covariance matrices $\boldsymbol{C}_{qq}(t_1, t_2)$ and $\boldsymbol{C}_{aa}$. The system leads to chaotic solutions where small perturbations of initial conditions lead to a completely different solution after a certain time integration.

Measurements of the solution are represented through the measurement equation

$$\mathcal{M}[\boldsymbol{x}] = \boldsymbol{d} + \boldsymbol{\epsilon}. \tag{6.11}$$

Further, by allowing the dynamical model equations (6.5–6.7) to contain errors, we obtain the standard weak constraint variational formulation,

$$\mathcal{J}[x,y,z] = \iint_0^T \boldsymbol{q}(t_1)^{\mathrm{T}}\, \boldsymbol{W}_{qq}(t_1,t_2)\, \boldsymbol{q}(t_2)\, dt_1\, dt_2 + \boldsymbol{a}^{\mathrm{T}}\, \boldsymbol{W}_{aa}\, \boldsymbol{a} + \boldsymbol{\epsilon}^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon} \boldsymbol{\epsilon}. \tag{6.12}$$

The weight matrix, $\boldsymbol{W}_{qq}(t_1,t_2) \in \Re^{3\times 3}$, is defined as the inverse of the model error covariance matrix, $\boldsymbol{C}_{qq}(t_2,t_3) \in \Re^{3\times 3}$, from

$$\int_0^T \boldsymbol{W}_{qq}(t_1,t_2) \boldsymbol{C}_{qq}(t_2,t_3)\, dt_2 = \delta(t_1 - t_3) \boldsymbol{I}, \tag{6.13}$$

and we have the weight matrices, $\boldsymbol{W}_{aa} = \boldsymbol{C}_{aa}^{-1} \in \Re^{3\times 3}$ and $\boldsymbol{W}_{\epsilon\epsilon} = \boldsymbol{C}_{\epsilon\epsilon}^{-1} \in \Re^{M\times M}$.

6.1.2 Strong constraint assumption

The strong constraint assumption leads to the adjoint method which has proven to be efficient for linear dynamics, given that the strong constraint assumption is valid.

The strong constraint assumption, solved by the adjoint method, has been extensively used in the atmosphere and ocean communities. Particular effort has been invested in developing the adjoint method for use in weather forecasting systems, where it is named 4DVAR (4–dimensional variational method). 4DVAR implementations are today in operational or preoperational use at atmospheric weather forecasting centers, but common for these is that they still only works well for rather short assimilation time intervals of one day or less. The causes for this may be connected to the tangent linear approximation but also to the chaotic nature of the dynamical model.

The strong constraint inverse problem for the Lorenz equations is defined by assuming that the model is perfect, $\boldsymbol{q}(t) \equiv 0$, and only the initial conditions contain errors. A number of papers have examined the adjoint method with the Lorenz model, see e.g. *Gauthier* (1992), *Stensrud and Bao* (1992), *Miller et al.* (1994), *Pires et al.* (1996). In these works it was found that there is a strong sensitivity of the penalty function with respect to the initial conditions. In particular there is a problem when the assimilation time interval exceeds a few times the predictability time of the model.

Miller et al. (1994) found that the penalty function changed from a nearly quadratic shape around the global minimum, for short assimilation time intervals, to a shape similar to a white noise process when the assimilation time interval was extended.

This is illustrated in Fig. 6.1 which plots values of the cost function with respect to variation in $x(0)$ while $y(0) = y_0$ and $z(0) = z_0$ are kept constant at their prior estimates. It is further assumed that all components of the solution $\boldsymbol{x}(t)$ are observed at regular time intervals $t_j = j\,\Delta t_{\mathrm{obs}}$, for $j = 1,\ldots,m$, with $\Delta t_{\mathrm{obs}} = 1$. We can then define the measurement equation for each measurement time t_j, as

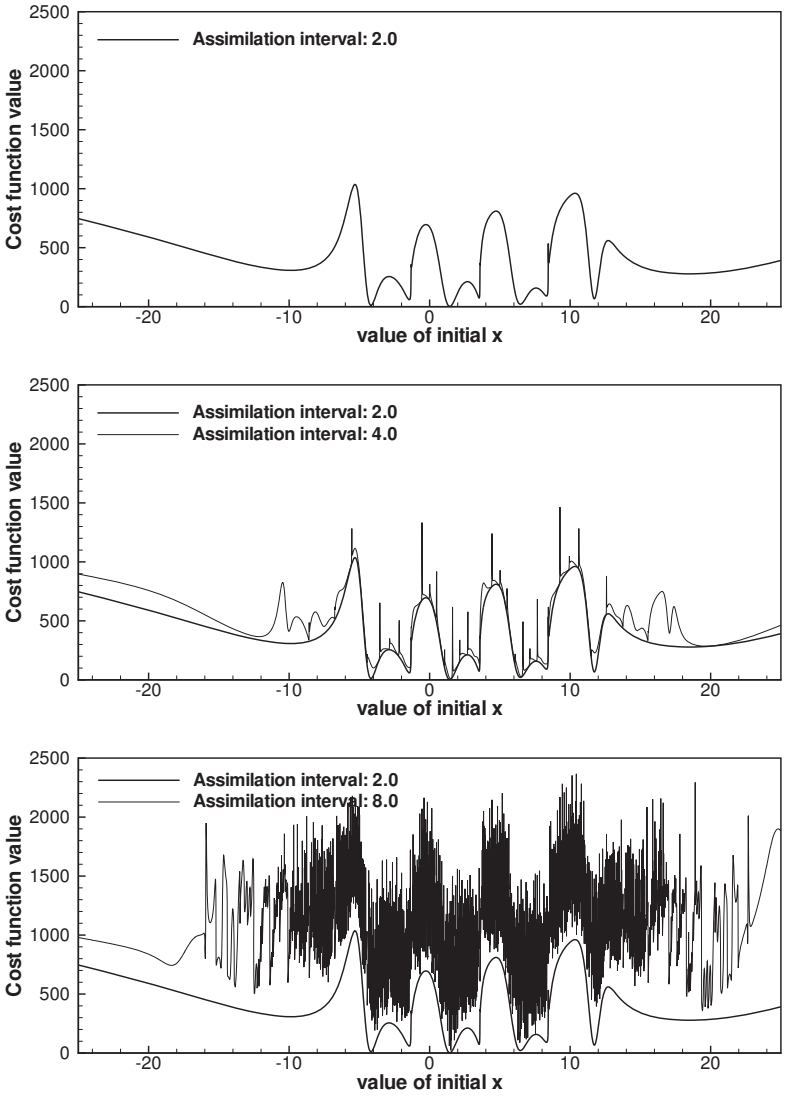

Fig. 6.1. Strong constraint penalty function for the Lorenz model as a function of the initial x-value, keeping y and z constant, when using data in the intervals $t \in [0, 2]$ *(upper plot)*, $t \in [0, 4]$ *(middle plot)*, and $t \in [0, 8]$ *(lower plot)*

$$\mathcal{M}_j[\boldsymbol{x}] = \boldsymbol{d}_j + \boldsymbol{\epsilon}_j, \tag{6.14}$$

where $\boldsymbol{\epsilon}_j$ represents the random errors in the measurements.

The value of the penalty function can be evaluated from

$$\begin{aligned}\mathcal{J}_J[\boldsymbol{x}(0)] = &\big(\boldsymbol{x}(0) - \boldsymbol{x}_0\big)^{\mathrm{T}} \boldsymbol{W}_{aa} \big(\boldsymbol{x}(0) - \boldsymbol{x}_0\big) \\ &+ \sum_{j=1}^{J} \Big(\boldsymbol{d}_j - \mathcal{M}_j\big[\boldsymbol{x}\big]\Big)^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon}(j) \Big(\boldsymbol{d}_j - \mathcal{M}_j\big[\boldsymbol{x}\big]\Big),\end{aligned} \tag{6.15}$$

where the subscript J, defines the length of the assimilation time interval and indicates that measurements up to the J'th measurement time are included. The weights $\boldsymbol{W}_{aa}$ and $\boldsymbol{W}_{\epsilon\epsilon}(j)$ are three by three matrices and have the same interpretation as in the previous sections.

The upper plot of Fig. 6.1 is for a very short assimilation time interval of $t \in [0, 2]$, i.e. only twice the characteristic time scale of the model dynamics. It is clear that even for this short time-interval there are local minima in the cost function and a very good prior estimate of the initial state is needed for a gradient based method to converge to the global minimum near $x(0) = 1.5$. In the middle plot the assimilation interval is extended to $t \in [0, 4]$ and we see that even though the basic shape is the same there now appear some additional spikes and local minima in the cost function. When the assimilation time interval is extended to $t \in [0, 8]$ in the lower plot, the shape of the cost function appears nearly as a white noise process. It is obvious that these cost functions cannot be minimized using traditional gradient based methods, and obviously, the strong constraint problem for the Lorenz equations becomes practically impossible to solve for long assimilation time intervals, independent of the method used.

It should at this time be noted that this is mainly a result of the formulation of the problem, i.e. the assumption that the model is an exact representation of unstable and chaotic dynamics. It is not unlikely that similar problems can occur in models of the ocean and atmosphere which resolves the chaotic mesoscale circulation, and this may be one of the reasons why 4DVAR appears to be limited to short assimilation time intervals in these applications.

The approach for resolving this problem in the atmospheric community has been to solve a sequence of strong constraint inverse problems, of the form (6.15), defined for separate subintervals in time. To illustrate this, assume that we have divided the assimilation time interval into one-day sub-intervals, and we define a strong constraint inverse problem for each one-day time interval on the form (6.15). Thus:

1. We start by solving the first sub-problem for day one which results in an estimate for the initial conditions at day one.
2. Integration of the model from this initial condition provides the strong constraint inverse solution for day one.
3. We then use the inverse solution from the end of day one to specify the prior estimate of the initial conditions for day two.

4. The problem now is that, for day two, one cannot easily compute an estimate of a new updated prior error statistics $\boldsymbol{W}_{aa}$, for the initial conditions, that accounts for the new information introduced in the previous inverse calculation. Thus, the original prior $\boldsymbol{W}_{aa}$ is used repeatedly for each sub-interval.

Using this procedure, there is no proper time evolution of the error covariances, thus a different problem than the originally posed strong constraint problem is solved. Estimation of the proper error covariance matrix would require the computation of the inverse of the Hessian of the penalty function, which equals the error covariance matrix for the estimated initial conditions, followed by the evolution of this error covariance matrix through the assimilation interval using an approximate error covariance equation like in the EKF.

6.1.3 Solution of the weak constraint problem

We already saw that if the dynamical model is not too nonlinear, a convergent sequence of linear iterates may be defined, and each iterate can be optimally solved using the representer method. For dynamical models with stronger nonlinearities the sequence of linear iterates may not converge and alternative methods need to be used.

Another class of methods for minimizing (6.12) is named substitution methods. These are methods that guess candidates for the minimizing solution and then evaluate the value of the penalty function. Dependent of the algorithm used the new candidate may be accepted with a specified probability if it results in a lower value for the penalty function.

A discrete version of the penalty function is now needed and we represent the model variables $x(t)$, $y(t)$, and $z(t)$ on a numerical grid in time. The variables are stored in the state vectors $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$, all belonging to $\Re^n$, i.e. we have the vector $\boldsymbol{x}^{\mathrm{T}} = (x_1, x_2, \ldots, x_n)$, and similarly for $\boldsymbol{y}$ and $\boldsymbol{z}$, where n is the number of grid points in time. The discrete analog to (6.12) then becomes

$$\mathcal{J}[\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}] = \Delta t^2 \sum_{i=1}^{n} \sum_{j=1}^{n} \boldsymbol{q}(i)^{\mathrm{T}} \boldsymbol{W}_{qq}(i,j)\, \boldsymbol{q}(j) + \boldsymbol{a}^{\mathrm{T}} \boldsymbol{W}_{aa}\, \boldsymbol{a} + \boldsymbol{\epsilon}^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon} \boldsymbol{\epsilon}, \quad (6.16)$$

where $\boldsymbol{q}(i)^{\mathrm{T}} = (q_x(t_i), q_y(t_i), q_z(t_i))$. Furthermore, there will be no integration of the model equations required using the substitution methods and simple numerical discretizations based on second order centered differences for the time derivatives can be used, i.e.

$$\begin{aligned} \frac{x_{i+1} - x_{i-1}}{2\Delta t} &= \sigma(y_i - x_i) + q_x(t_i), \\ \frac{y_{i+1} - y_{i-1}}{2\Delta t} &= \rho x_i - y_i - x_i z_i + q_y(t_i), \\ \frac{z_{i+1} - z_{i-1}}{2\Delta t} &= x_i y_i - \beta z_i + q_z(t_i), \end{aligned} \quad (6.17)$$

where $i = 2, \ldots, n-1$ is the time-step index, with n the total number of time steps.

Note that the evaluation of the double sum in (6.16) is costly. Here, an alternative method like the one used for the convolutions in the representer method could be used.

An even more efficient approach was used by *Evensen and Fario* (1997). It is assumed that the model weight can be written as

$$\boldsymbol{W}_{qq}(t_1, t_2) = \boldsymbol{W}_{qq}\delta(t_1 - t_2), \tag{6.18}$$

where $\boldsymbol{W}_{qq}$ is a constant 3×3 matrix. This eliminates one of the summations in the model term in (6.16) and allows for more efficient computational algorithms. However, the correlation in time of the model errors has a time regularizing effect on the inverse estimate which has now been lost.

To ensure a smooth solution in time the regularization is instead accounted for by a smoothing term

$$\mathcal{J}_S[\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}] = \Delta t \sum_{i=1}^{n} \boldsymbol{\eta}_i^{\mathrm{T}} \boldsymbol{W}_{\eta\eta} \boldsymbol{\eta}_i, \tag{6.19}$$

where $\boldsymbol{\eta}_i^{\mathrm{T}} = \big(\eta_x(t_i), \eta_y(t_i), \eta_z(t_i)\big)$, with

$$\eta_x(t_i) = \frac{x_{i+1} - 2x_i + x_{i-1}}{\Delta t^2}, \tag{6.20}$$

and $\boldsymbol{W}_{\eta\eta}$ is a weight matrix determining the relative impact of the smoothing term.

It would have been more consistent to actually smooth the model errors instead of the inverse estimate, since it can be shown that such a smoothing constraint, used together with the penalty term for the model errors, would define a norm. Moreover, there is a unique correspondence between such a smoothing norm and a covariance matrix, as shown by *McIntosh* (1990). On the other hand, the smoothing term as included here, will improve the conditioning of the method since only smooth functions are searched for.

The penalty function now becomes

$$\begin{aligned} \mathcal{J}[\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}] = \Delta t \sum_{i=1}^{n} \boldsymbol{q}_i^{\mathrm{T}} \boldsymbol{W}_{qq} \boldsymbol{q}_i + \boldsymbol{a}^{\mathrm{T}} \boldsymbol{W}_{aa} \boldsymbol{a} + \boldsymbol{\epsilon}^{\mathrm{T}} \boldsymbol{w} \boldsymbol{\epsilon} \\ + \Delta t \sum_{i=1}^{n} \boldsymbol{\eta}_i^{\mathrm{T}} \boldsymbol{W}_{\eta\eta} \boldsymbol{\eta}_i. \end{aligned} \tag{6.21}$$

For $\boldsymbol{q}_1$, $\boldsymbol{q}_n$, $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_n$ we use second order one-sided difference formulas.

6.1.4 Minimization by the gradient descent method

A very simple approach for minimizing the penalty function (6.21) is to use a gradient descent algorithm as was done by *Evensen* (1997), *Evensen and Fario*

(1997). The gradient $\nabla_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})}\mathcal{J}[\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}]$, with respect to the full state vector in time $(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$, is easily derived. When the gradient is known it can be used in a descent algorithm to search for the minimizing solution. Thus, for the Lorenz model we solve the iteration

$$\begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{y}\\ \boldsymbol{z}\end{pmatrix}^{i+1} = \begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{y}\\ \boldsymbol{z}\end{pmatrix}^{i} - \gamma \begin{pmatrix}\nabla_{\boldsymbol{x}}\mathcal{J}[\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}]\\ \nabla_{\boldsymbol{y}}\mathcal{J}[\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}]\\ \nabla_{\boldsymbol{z}}\mathcal{J}[\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}]\end{pmatrix}^{i}. \tag{6.22}$$

with γ being a step length. Given a first guess estimate, the gradient of the cost function is evaluated and a new state estimate can be searched for in the direction of the gradient.

The required storage for the gradient descent method is of order the size of the state vector in space and time, which is the same as for the adjoint and representer methods.

Note that, using a gradient descent method there is no need for any model integrations. This is contrary to the representer and adjoint methods which integrate both the forward model and the adjoint model, and to the Kalman filter where the forward model is needed.

As long as the penalty function does not contain any local minima, the gradient method will eventually converge to the minimizing solution. However, the obvious drawback is that the dimension of the problem becomes huge for high dimensional problems, i.e. the number of dependent variables times the grid points in time and space. For the Lorenz model this becomes $3n$. This is normally much larger than the number of measurements which defines the dimension of the problem as solved by the representer method. Thus, a proper conditioning may be needed to ensure that high dimensional problems converge in an acceptable number of iterations.

6.1.5 Minimization by genetic algorithms

With nonlinear dynamics the penalty function is clearly not convex in general due to the first term in (6.21) containing the model residuals. However, both the measurement penalty term and the smoothing norm will give a quadratic contribution to the penalty function and if the weights, $\boldsymbol{W}_{\epsilon\epsilon}$ and $\boldsymbol{W}_{\eta\eta}$, are large enough compared to the dynamical weight $\boldsymbol{W}_{qq}$, one can expect a nearly quadratic penalty function. On the contrary, if the model residuals are the dominating terms in the penalty function, clearly a pure descent algorithm may get trapped in local minima and the solution found may depend on the first guess in the iteration.

A special class of substitution methods contains the so-called genetic algorithms. These are typically statistical methods which guess new candidates for the minimizing solution at random or using some wise candidate selection algorithm. Then an acceptance algorithm is used to decide whether the new candidate is accepted or not. The acceptance algorithm is dependent on the

value of the penalty function but also has a random component which allows it to escape local minima.

Statistical versions of the genetic methods exploit the fact that the minimizing solution can be interpreted as the maximum likelihood estimate of a probability density function,

$$f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \propto \exp\Big(-\mathcal{J}\big[\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}\big]\Big). \tag{6.23}$$

Moments of $f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ could be estimated using standard numerical integration based on Monte Carlo methods using points selected at random from some distribution. However, this would be extremely inefficient due to the huge state space associated with many high dimensional models, such as models of the ocean and atmosphere.

Metropolis algorithm

Instead a method by *Metropolis et al.* (1953) is useful, and we now illustrate it for the variable $\boldsymbol{\psi}^{\mathrm{T}} = (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. The algorithm samples a pdf by performing a random walk through the space of interest. At each sample position $\boldsymbol{\psi}$, a perturbation is added to generate a new candidate $\boldsymbol{\psi}_1$, and this candidate is accepted according to a probability

$$p = \min\left(1, \frac{f(\boldsymbol{\psi}_1)}{f(\boldsymbol{\psi})}\right). \tag{6.24}$$

The mechanism for accepting the candidate with probability p, is implemented by drawing a random number ξ, from the uniform distribution on the interval $[0, 1]$ and then accepting $\boldsymbol{\psi}_1$ if $\xi \leq p$. The conditional uphill climb, based on the value of p and ξ, is due to *Metropolis et al.* (1953) and is named the Metropolis algorithm. They also gave a proof that the method was ergodic, i.e. any state can be reached from any other, and that the trials would sample the probability distribution $f(\boldsymbol{\psi})$. Clearly, in a high dimensional space with strongly nonlinear dynamics, the random trials may be too random and most of the time lead to candidates $\boldsymbol{\psi}_1$, with very low probabilities, which are only occasionally accepted. Thus, the algorithm becomes very inefficient.

Hybrid Monte Carlo algorithm

In *Bennett and Chua* (1994) an alternative to a random walk, which provided a significantly faster convergence, was used when solving for the inverse of a nonlinear open ocean shallow water model. The algorithm which is due to *Duane et al.* (1987) ensures that candidates with acceptable probabilities are constructed. It is based on constructing the Hamiltonian

$$\mathcal{H}[\boldsymbol{\psi}, \boldsymbol{\pi}] = \mathcal{J}[\boldsymbol{\psi}] + \frac{1}{2}\boldsymbol{\pi}^{\mathrm{T}}\boldsymbol{\pi}, \tag{6.25}$$

and then deriving the canonical equations of motion in $(\boldsymbol{\psi}, \boldsymbol{\pi})$ phase space, with respect to a pseudo time variable τ,

$$\frac{\partial \psi_i}{\partial \tau} = \frac{\partial \mathcal{H}}{\partial \pi_i} = \pi_i, \tag{6.26}$$

$$\frac{\partial \pi_i}{\partial \tau} = -\frac{\partial \mathcal{H}}{\partial \psi_i} = -\frac{\partial \mathcal{J}}{\partial \psi_i}. \tag{6.27}$$

This system is integrated for a pseudo time interval, $\tau \in [0, \tau_1]$, using the previously accepted value of $\boldsymbol{\psi}$ and a random guess for $\boldsymbol{\pi}(0)$ as initial conditions. The Metropolis algorithm can then be used for the new guess $\boldsymbol{\psi}(\tau_1)$. In *Duane et al.* (1987), it was proved that this algorithm also preserved detailed balance, i.e.

$$f(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2) = f(\boldsymbol{\psi}_2|\boldsymbol{\psi}_1) f(\boldsymbol{\psi}_1) = f(\boldsymbol{\psi}_1|\boldsymbol{\psi}_2) f(\boldsymbol{\psi}_2), \tag{6.28}$$

which is needed for showing that a long sequence of random trials will converge towards the distribution (6.23).

The interpretation of the method is clear. In the Hamiltonian (6.25), the penalty function defines a potential energy while a kinetic energy is represented by the last term. The canonical equations describe motion along lines of constant total energy. Thus, with a finite and random initial momentum, the integration of the canonical equation over a pseudo time interval will result in a new candidate with a different distribution of potential and kinetic energy. Unless the initial momentum is very large this will always result in a candidate which has a reasonable probability. If the initial momentum is zero, it will result in a candidate with less potential energy and higher probability. If the initial candidate is a local minimum, the random initial momentum may provide enough energy to escape the local minimum.

Note that, after a minimum of the variational problem has been found, the posterior error statistics can be estimated by collecting samples of nearby states. Thus, by using the hybrid Monte Carlo method to generate a Markov chain that samples the probability function, a statistical variance estimate can be generated. This method may be used to generate error estimates independently of the minimization technique used to solve the weak constraint problem. Hence, it could also be used in combination with the representer method which does not easily provide error estimates.

Simulated annealing

When working with a penalty function which has many local minima, the so-called simulated annealing technique may be used to improve the convergence to the stationary distribution, based on the method's capability of escaping local minima.

The simulated annealing method (see *Kirkpatrick et al.*, 1983, *Azencott*, 1992) is extremely simple in its basic formulation and can be illustrated using an example where a penalty function $\mathcal{J}[\boldsymbol{\psi}]$, which may be nonlinear and discontinuous, is to be minimized with respect to the variable $\boldsymbol{\psi}$:

$\boldsymbol{\psi}$ first guess
for $i = 1 : \ldots$
  $\boldsymbol{\psi}_1 = \boldsymbol{\psi} + \Delta\boldsymbol{\psi}$
  if $(\mathcal{J}[\boldsymbol{\psi}_1] < \mathcal{J}[\boldsymbol{\psi}])$ **then**
    $\boldsymbol{\psi} = \boldsymbol{\psi}_1$
  else
    $\xi \in [0, 1]$ random number
    $p = \exp\Big(\big(\mathcal{J}[\boldsymbol{\psi}] - \mathcal{J}[\boldsymbol{\psi}_1]\big)/\theta\Big) \in [0, 1]$
    if $p > \xi$ **then** $\boldsymbol{\psi} = \boldsymbol{\psi}_1$
  end
  $\theta = f(\theta, i, \mathcal{J}_{\min})$
end

Here $\Delta\boldsymbol{\psi}$ might be a normal distributed random vector, but it is more efficient to simulate it using the hybrid Monte Carlo technique just described.

The temperature scheme $\theta = \theta(\theta, i, \mathcal{J}_{\min})$ is used to cool or relax the system and is normally a decreasing function of iteration counter i.

The trials will then converge towards a distribution

$$f(\boldsymbol{\psi}) \propto \exp\big(-\mathcal{J}[\boldsymbol{\psi}]/\theta\big), \tag{6.29}$$

By slowly decreasing the value of θ the distribution will approach the delta function at the minimizing value of $\boldsymbol{\psi}$. The clue is then to choose a temperature scheme where one avoids getting trapped in local minima for too many iterations, or where too many uphill climbs are accepted. In *Bohachevsky et al.* (1986), it was suggested that the temperature should be chosen so that $p \in [0.5, 0.9]$. Here also a generalized algorithm was proposed where p was calculated according to $p = \exp\big(\beta(\mathcal{J}[\boldsymbol{\psi}] - \mathcal{J}[\boldsymbol{\psi}_1])/(\mathcal{J}[\boldsymbol{\psi}] - \mathcal{J}_{\min})\big)$, where β is approximately 3.5 and $\mathcal{J}_{\min}$ is an estimate of the normally unknown minimum value of the penalty function. Then the probability of accepting a detrimental step tend to zero as the random walk approaches the global minimum. If a value of the cost function is found which is less than $\mathcal{J}_{\min}$ this value will replace $\mathcal{J}_{\min}$.

Simulated annealing was previously used by *Barth and Wunsch* (1990) to optimize an oceanographic data collection scheme. The use of the hybrid Monte Carlo method in combination with simulated annealing has been extensively discussed by *Neal* (1992, 1993) in the context of Bayesian training of back-propagation networks. The method was also used to invert an inverse for a primitive equation model on a domain with ill-posed open boundaries by *Bennett and Chua* (1994). An application with the Lorenz equations was discussed by *Evensen and Fario* (1997) and will be illustrated below.

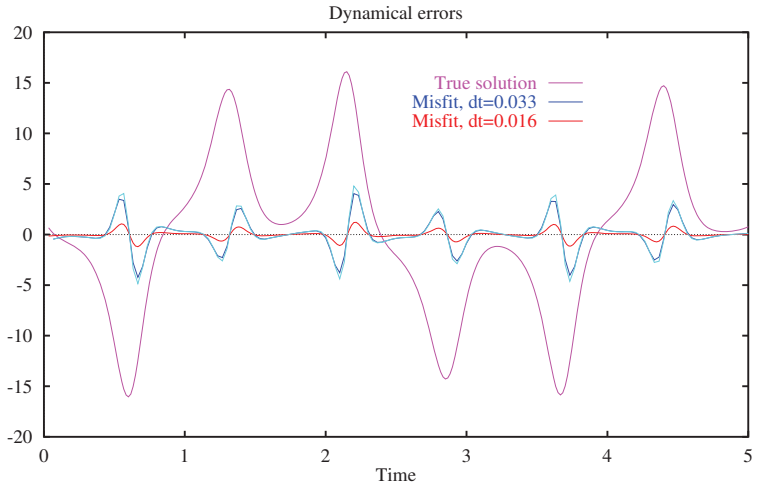

Fig. 6.2. Errors in the difference approximation used for the time derivative, plotted together with the reference solution used in the calculation of the errors. The two similar curves for $\Delta t = 0.033$ are comparing the actual calculated misfits and the lowest-order error term in the discrete time derivative. Reproduced from *Evensen and Fario* (1997)

6.2 Example with the Lorenz equations

We will now present an example where the gradient descent and the simulated annealing algorithm are used with the Lorenz equations. This example is similar to the one discussed by *Evensen and Fario* (1997).

6.2.1 Estimating the model error covariance

In an identical twin experiment it is possible to generate accurate estimates of the model error covariance. First the reference or true solution is computed using a highly accurate ordinary differential equation solver. Then the only significant contribution to the dynamical error term $\boldsymbol{q}_n$, is the error introduced in the approximate time discretization (6.17). These misfits can be evaluated and used to determine the weight matrices $\boldsymbol{W}_{qq}$ and $\boldsymbol{W}_{\eta\eta}$, which are needed in the inverse calculation.

An alternative is to evaluate the first order error term in the centered first derivative approximation used in the discrete model equations (6.17), i.e. we write for the time derivative of $x(t)$,

$$\frac{\partial x}{\partial t} = \frac{x(t+\Delta t) - x(t-\Delta t)}{2\Delta t} + \frac{1}{6}\frac{\partial^3 x}{\partial t^3}\Delta t^2 + \ldots, \tag{6.30}$$

and evaluate the error term given the true solution.

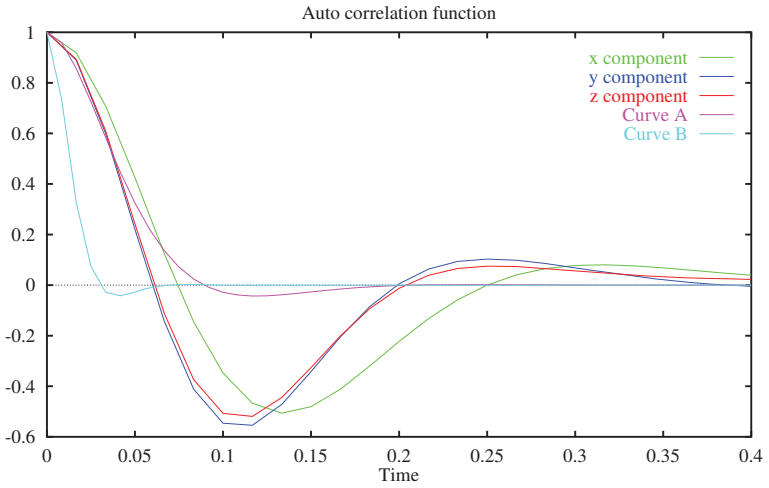

Fig. 6.3. Auto-correlation functions calculated for the computed dynamical misfits for the x, y, and z component of the solution, and two auto-correlation functions corresponding to the smoothing norm with $\gamma = 0.0008$ *(curve A)* and $\gamma = 0.00001$ *(curve B)*. Reproduced from *Evensen and Fario* (1997)

In Fig. 6.2 the dynamical misfits are plotted using two different time steps. Clearly, the errors increase with the length of the time step and the maximum errors are located at the peaks of the reference solution. The two almost identical curves for $\Delta t = 0.033$ are generated using the two different approaches just described.

The error covariance matrix $\boldsymbol{C}_{qq}$ can be estimated from a long time series of these errors, and is of course dependent on the time step used. In the experiments presented here we use a time step of $\Delta t = 0.01667$ and the corresponding error covariance matrix then becomes

$$\boldsymbol{C}_{qq} = \begin{bmatrix} 0.1491 & 0.1505 & 0.0007 \\ 0.1505 & 0.9048 & 0.0014 \\ 0.0007 & 0.0014 & 0.9180 \end{bmatrix}, \tag{6.31}$$

where the integration has been performed for a long time interval $t \in [0, 1667]$, i.e. 100 000 time steps. The inverse of this matrix is used for $\boldsymbol{W}_{qq}$ in the penalty function (6.21).

6.2.2 Time correlation of the model error covariance

The errors are also clearly correlated in time. In Fig. 6.3 the auto-correlation functions for the x, y, and z components of the dynamical errors are plotted. Since it is inconvenient to use a full space and time covariance matrix, we

introduce the smoothing term (6.19), which act as a regularization term on the minimizing solution.

It can be shown that a smoothing norm of the type

$$||\psi|| = \int_0^T \psi^2 + \gamma \psi_{tt}^2 \, dt \tag{6.32}$$

has a Fourier transform equal to

$$\hat{\psi} = \left(1 + \gamma \omega^4\right)^{-1}. \tag{6.33}$$

The limiting behaviour for increasing frequency ω is then proportional to $(\gamma\omega^4)^{-1}$; thus high frequencies are penalized most strongly in the smoothing norm. The ψ^2 term is added here, as a first guess penalty, for illustrational purposes. Without this term, the limiting behaviour for $\omega \to 0$ would be singular and the corresponding auto-correlation function would become very flat. In the actual inverse formulation, the dynamical and initial residual will provide the first guess penalty, ensuring a well-behaved limiting behaviour when $f \to 0$.

An inverse Fourier transform of the spectrum (6.33) gives an auto-correlation function which is shown in Fig. 6.3 for two values of γ, i.e. $\gamma = 0.0008$ for *curve A* and $\gamma = 0.00001$ for *curve B*. For $\gamma = 0.0008$ the auto-correlation function has a similar half width to the auto-correlation functions of the dynamical errors. However, it turned out that for this value of γ the inverse estimate became too smooth, i.e. the peaks in the solutions were to low compared to the reference solution. We decided to use $\gamma = 0.00001$ which gave an inverse estimate more in agreement with the reference solution. Based on the time series of dynamical misfits in Fig. 6.2, it is also clear that the errors are rather smooth for most of the time while they have sudden changes close to the peaks of the reference solution. The computed auto-correlation function will describe an "average" smoothness of the dynamical misfits which is too smooth near the peaks in the reference solution. This can then justify the use of the smaller smoothing weight $\gamma = 0.00001$.

The error covariance matrix $\boldsymbol{C}_{aa}$ for the errors in the initial conditions, and the measurement error covariance matrix $\boldsymbol{C}_{\epsilon\epsilon}$, are both assumed to be diagonal and with the same error variance equal to 0.5. The model error covariance matrix is given by (6.31) and the smoothing weight matrix is chosen to be diagonal and given by $\boldsymbol{W}_{\eta\eta} = \gamma \boldsymbol{I}$ with $\gamma = 0.00001$.

6.2.3 Inversion experiments

For all the cases to be discussed the initial condition for the reference case is given by $(x_0, y_0, z_0) = (1.508870, -1.531271, 25.46091)$ and the observations and first guess initial conditions are simulated by adding normal distributed noise, with zero mean and variance equal to 0.5, to the reference solution.

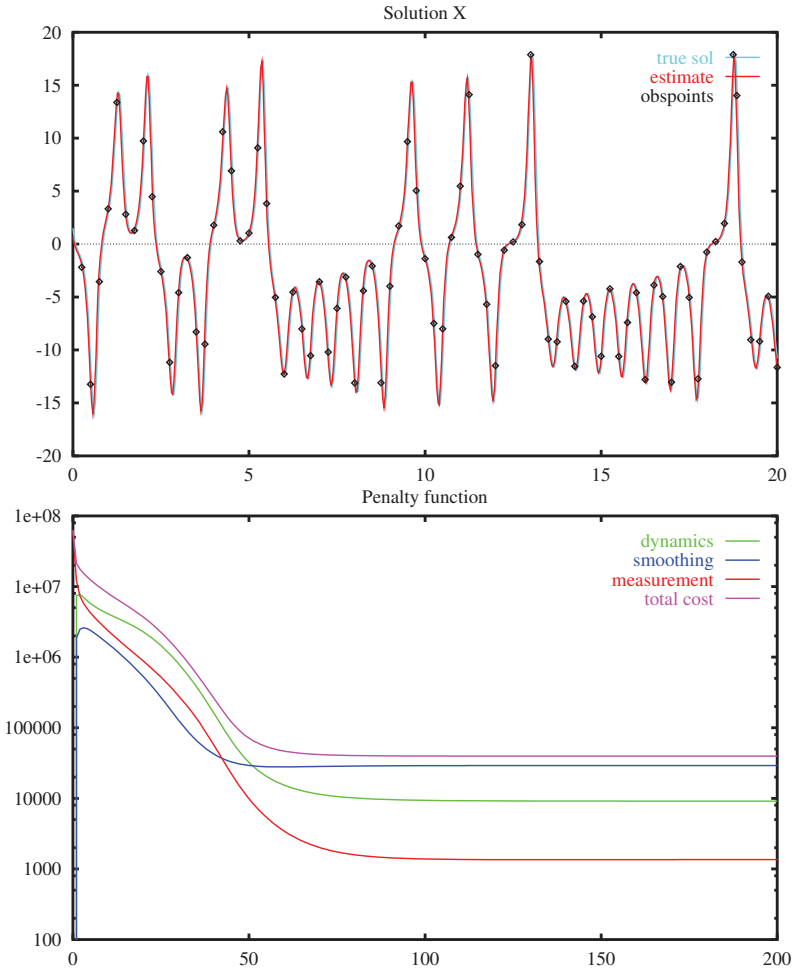

Fig. 6.4. Case A: The inverse estimate for x *(top)* and the terms in the penalty function *(bottom)*. The estimated solution is given by the solid line. The dashed line is the true reference solution, and the diamonds show the simulated observations. The same line types will be used also in the following figures. Reproduced from *Evensen and Fario* (1997)

These are lower values than the variances equal to 2.0, used in *Miller et al.* (1994) and *Evensen and Fario* (1997).

The first guess used in the gradient descent method was initially chosen as the mean of the reference solution, i.e. about $(0, 0, 23)$. However, there seems to be a possibility for a local minima close to the zero solution where both the dynamical penalty term and the smoothing penalty vanish. It is therefore

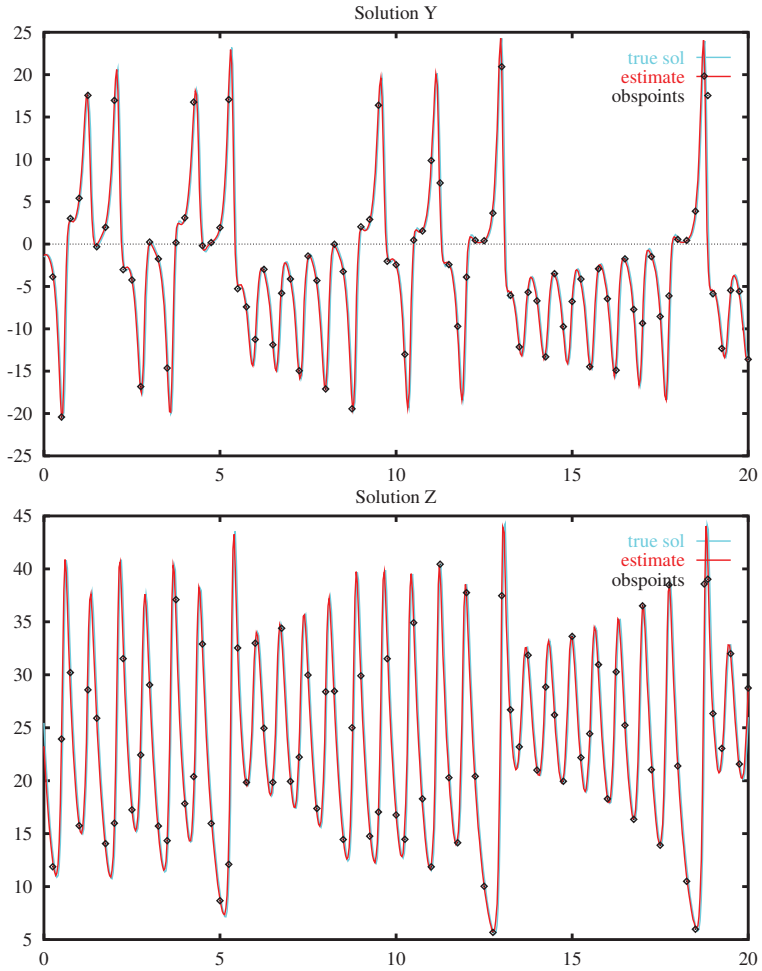

Fig. 6.5. Case A: The inverse estimate for y (top) and z (bottom). Reproduced from *Evensen and Fario* (1997)

not wise to use an estimate close to the zero solution as the first guess in the descent algorithm. To reduce the probability of getting trapped in eventual local minima, an objective analysis estimate, consistent with the measurements, was used as a first guess in the descent algorithm. It was calculated using a smoothing spline minimization algorithm which is equivalent to objective analysis (*McIntosh*, 1990). This could easily be done by replacing the dynamical misfit term with a penalty of a first-guess estimate in the inverse formulation (6.21). Some examples will now be discussed.

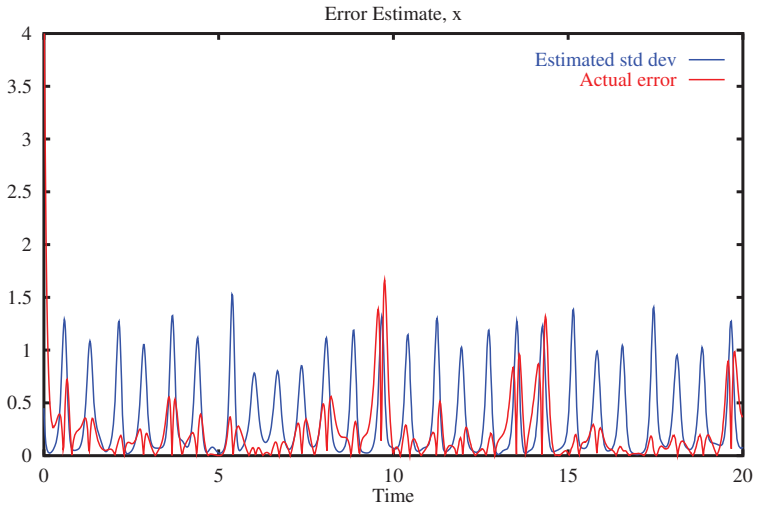

Fig. 6.6. Case A: Statistical error estimates (standard deviations) for x together with the absolute value of the actual errors. Reproduced from *Evensen and Fario* (1997)

Case A

This case can be considered as a base case and is, except for the lower measurement errors, similar to the case discussed by *Miller et al.* (1994); i.e. the time interval is $t \in [0, 20]$ and the distance between the measurements is $\Delta t_{\text{obs}} = 0.25$. The gradient descent method was in this case capable of finding the global minimum when starting from the objective analysis estimate. The minimizing solution for the three variables is given in Figs. 6.4 and 6.5 together with the terms in the penalty function as a function of iteration. We find it amazing how close the inverse estimate is to the reference solution. The quality of this inverse estimate is clearly superior to previous inverse calculations using the extended Kalman filter or a strong constraint formulation.

From the terms in the penalty function given in Fig. 6.4, it is seen that the first guess is close to the measurements and rather smooth, while the dynamical residuals are large and contribute with more than 99 % of the total value of the cost function. During the iterations, the dynamical misfit is reduced while there is an initial increase in the smoothing and measurement terms, which indicates that the final inverse solution is further from the measurements and less smooth than the first guess.

The hybrid Monte Carlo method was used to estimate the standard deviations of the errors in the minimizing solution. These are plotted together with the true differences between the estimate and the reference solution in Fig. 6.6 for the x-component. The largest errors appear around the peaks of the solution and the statistical and true errors are similar.

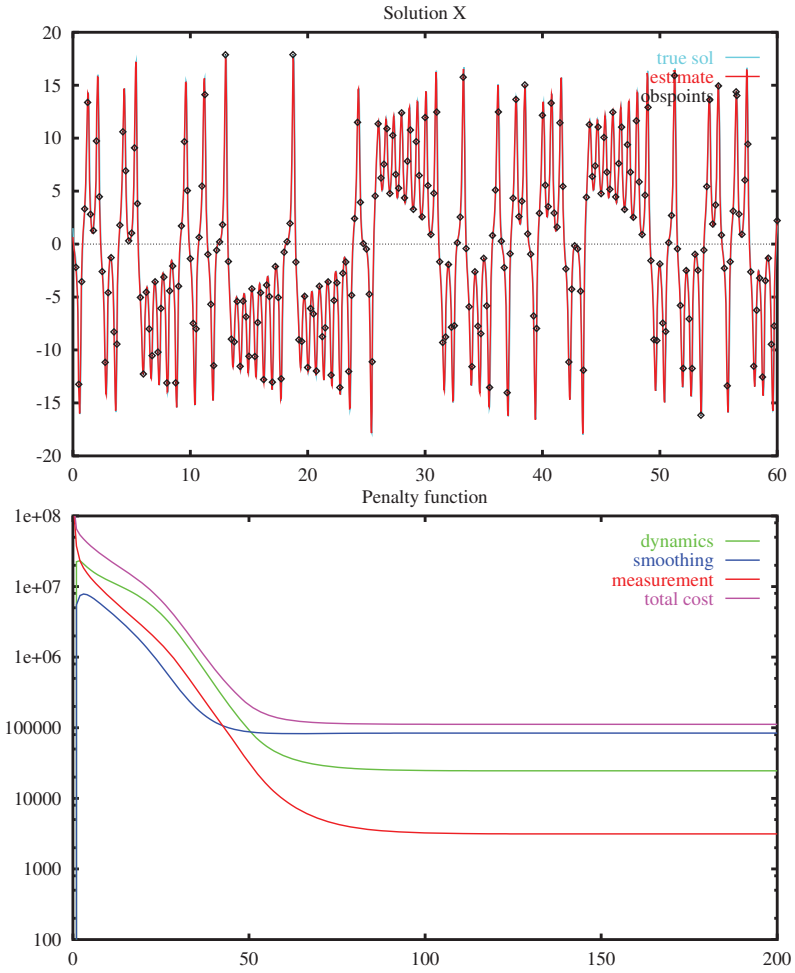

Fig. 6.7. Case B: The inverse estimate for x *(top)* and the penalty function *(bottom)*. Reproduced from *Evensen and Fario* (1997)

Case B

Here, we extended the time interval to $T = 60$, to test the sensitivity of the inverse estimate with respect to a longer time interval. The number of measurements is increased by a factor of 3 to give the same data density as in Case A. Note that the value of the cost function is also increased by about a factor of 3. This case behaves similarly to Case A, with convergence to the global minimum at a similar rate as in Case A. In Fig. 6.7 the x-component of the solution is given together with the terms in the penalty function.

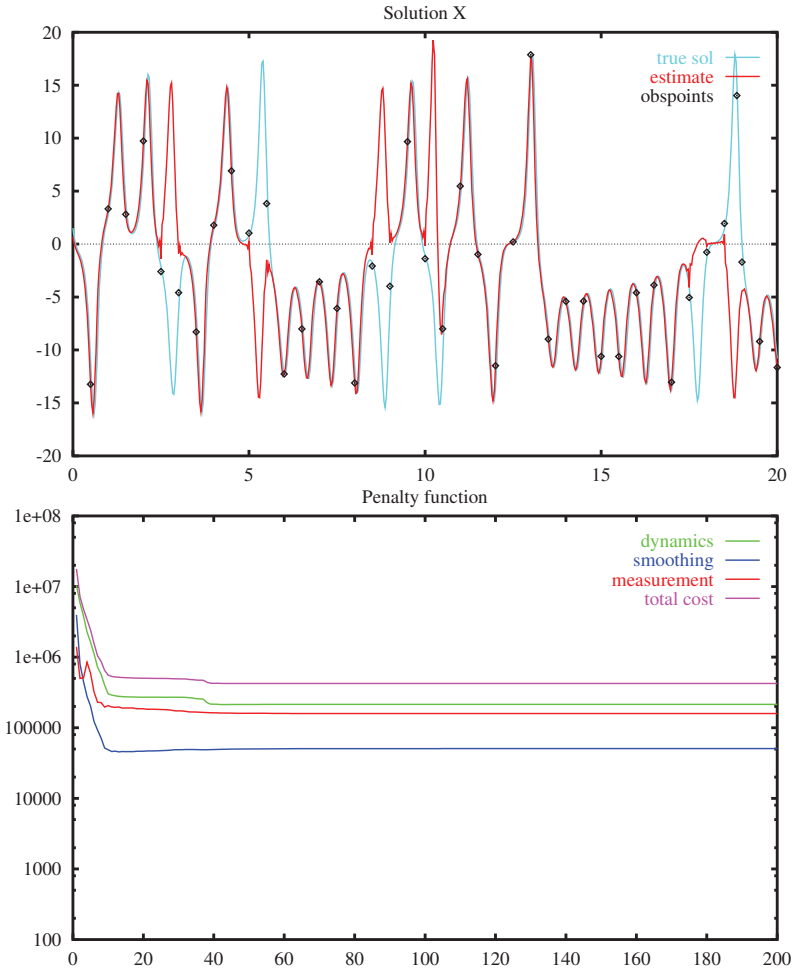

Fig. 6.8. Case C: The inverse estimate for x *(top)* and the penalty function *(bottom)*. Reproduced from *Evensen and Fario* (1997)

An important conclusion from this example is that by using a weak constraint variational formulation for the inverse, the strong sensitivity with respect to perturbations in initial conditions which is observed for strong constraint variational formulations, is completely removed. The weak constraint formulation allows the dynamical model to "forget" very past and future information. The convergence of the inverse calculation therefore has a "local" behaviour where the current estimate at two distant locations have vanishing influence on each other.

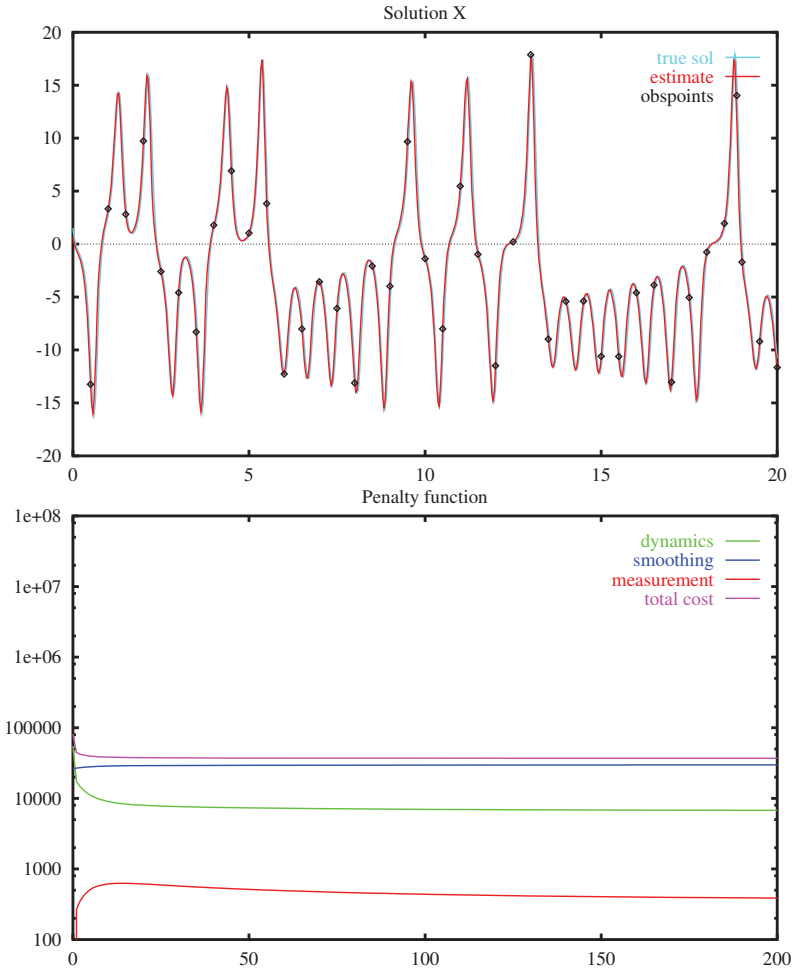

Fig. 6.9. Case C: The inverse estimate for x *(top)* and the penalty function *(bottom)* when the reference solution is used as the first guess in the gradient descent algorithm. Reproduced from *Evensen and Fario* (1997)

Case C

When the distance between the measurements is increased to $\Delta t_{\text{obs}} = 0.50$, a solution is found which misses several of the transitions, as seen in the solution for the x-component given in Fig. 6.8 together with the terms in the penalty function. This is an indication that the gradient algorithm converged to a local minimum. We can verify that this is in fact the case by running another minimization where the true reference solution is used as the first guess for the gradient method. The result is given in Fig. 6.9 where, after

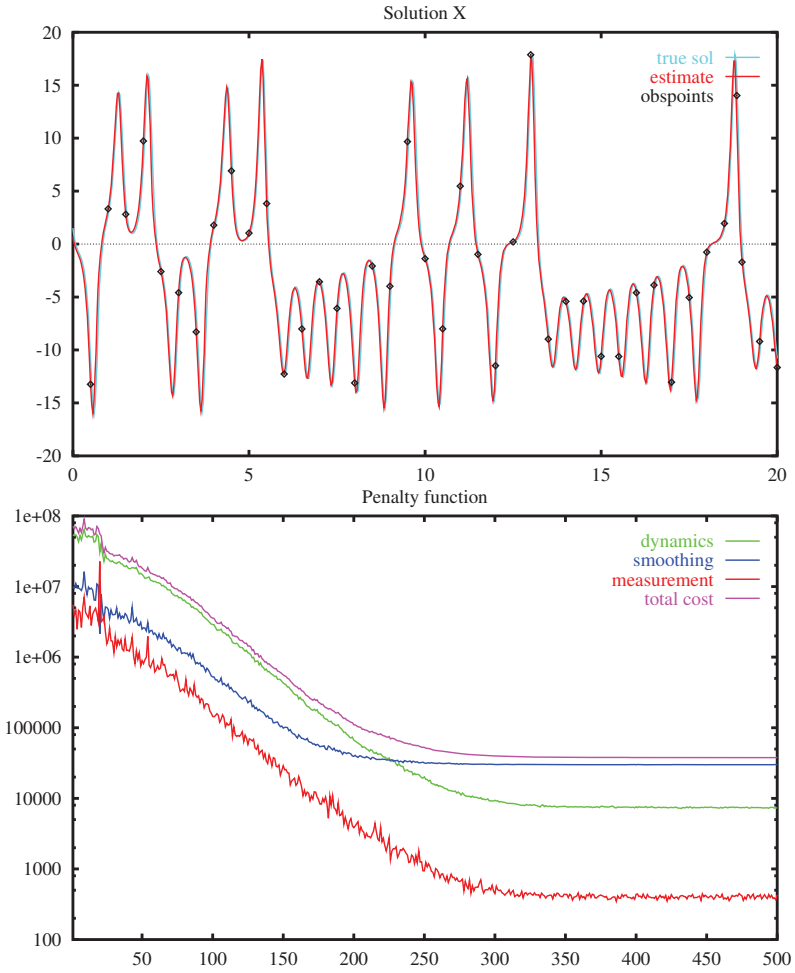

Fig. 6.10. Case C1: The inverse estimate for x *(top)* and the penalty function *(bottom)* where a genetic algorithm based on simulated annealing is used. Reproduced from *Evensen and Fario* (1997)

a minor initial adjustment, the algorithm converges to the global minimum which has a significantly lower value of the cost function and which captures all the transitions. Thus, we can conclude that when the measurement density is lowered the measurement term will give a smaller quadratic contribution to the cost function and at some stage local minima start to appear.

Case C1

This case is similar to Case C, but now the hybrid Monte Carlo method is used in combination with simulated annealing for minimizing the penalty function. The minimizing solution is in this case given in Fig. 6.10. Note that the number of iterations required for convergence is higher in this case than in the previous ones. This is due to perturbations caused by the annealing process that allows uphill moves to migrate out of local minima. The method used here is actually not proper annealing but should be denoted quenching, since the system is cooled too fast to guarantee that the global minimum will be found. In fact, in a similar case in *Evensen and Fario* (1997) a local minimum was found.

6.2.4 Discussion

A weak constraint variational formulation for the Lorenz model has been minimized using a gradient descent method.

It has been illustrated that by imposing the dynamical model as a weak constraint, by allowing the dynamics to contain errors, this leads to a better posed problem than the strong constraint formulation. The weak constraint formulation eliminates the sensitivity with respect to the initial conditions since, by allowing for model errors, the estimate can deviate from an exact model trajectory and thereby forget very past and future information. Further, there are no limitations on the length of the assimilation interval.

The inverse was calculated using the full state in "space" and time as control variables. The huge state space associated with such a formulation is the main objection against using a gradient descent method for a weak constraint inverse calculation. It could be compared to the mathematically very appealing representer method (Bennett, 1992), where the solution is searched for in a space with dimension equal to the number of measurements. On the other hand, with a gradient descent approach there is no need to integrate any dynamical equations, since a new candidate for the solution in space and time is substituted in every iteration. This gives rise to the notation substitution methods, where the important issue is the method used for proposing the solution candidates.

A gradient descent method will always provide a solution. However, it may be a local minimum if the penalty function is not convex. Statistical methods based on simulated annealing in combination with a hybrid Monte Carlo method for generating the candidates are much more expensive than a gradient descent approach but has a higher probability of finding the global minimum. The genetic methods will, for practical problems, only lead to a marginal improvement since they can only solve a slightly more difficult problem to a much larger cost. Thus, one should rather try to define a better posed problem, e.g. by introducing additional measurements.

It should be noted that with reasonable good measurement coverage the penalty function is essentially convex, but when either the number of measurements is decreased or with poorer quality of the measurements, the quadratic contribution to the penalty function from the measurement term has less influence and nonlinearities in the dynamics may give raise to local minima. Thus, the success of the substitution methods is strongly dependent on the measurement density. With sufficient number of measurements the algorithms converged to the global minimum of the weak constraint problem. When the number of measurements decreased, this resulted in a penalty function with multiple local minima and the gradient descent method was unable to converge to the global minimum.

It should also be pointed out that the gradient descent method does not directly provide error estimates for the minimizing solution. However, if the gradient descent method is first used to find the solution then the hybrid Monte Carlo method can be used to sample from the posterior distribution function and error variance estimates can be calculated.

An example of this method was used by *Natvik et al.* (2001) with a simple but nonlinear three component marine ecosystem model. In this case the dimension of the problem was equal to three variables times the number of grid nodes in time. Results similar to those found by *Evensen* (1997) were obtained, and the global minimum was found in the cases with sufficient measurement density. With a small number of measurements the gradient method converged to a local minimum.

The substitution methods solve for a state vector which consists of the model state vector in space and time. Clearly, this can be very large for realistic models and it does not appear to be a smart approach since we noted that the real dimension of the linear inverse problem equals the number of measurements. If the number of grid nodes is large, slow convergence is expected, and this was indeed a result from these studies.

In a final case, similar to case A, but only using measurements of the x-component of the solution, the global minimum was still found using the gradient descent method. In this case the estimates for y and z were entirely determined by the choice of model error covariance matrix and interactions through the dynamical equations. However, this case converged significantly slower. This is a result of poor conditioning and can be expected since the quadratic contribution from the measurement term is lower when only the x-component of the solution is measured. It also indicates that if the method is used with high dimensional problems, or with to sparse measurements, convergence problems may become crucial.