

Towards a true 4-dimensional data assimilation algorithm: application of a cycling representer algorithm to a simple transport problem

By LIANG XU^{1*} and ROGER DALEY², *Naval Research Laboratory, 7 Grace Hopper Ave., Monterey, CA 93943-5502, USA*; ¹*Science Applications International Corporation*, ²*University Corporation for Atmospheric Research*

(Manuscript received 3 December 1998; in final form 1 July 1999)

ABSTRACT

3-dimensional variational algorithms are widely used for atmospheric data assimilation at the present time, particularly on the synoptic and global scales. However, mesoscale and convective scale phenomena are considerably more chaotic and intermittent and it is clear that true 4-dimensional data assimilation algorithms will be required to properly analyze these phenomena. In its most general form, the data assimilation problem can be posed as the minimization of a 4-dimensional cost function with the forecast model as a weak constraint. This is a much more difficult problem than the widely discussed 4DVAR algorithm where the model is a strong constraint. Bennett and collaborators have considered a method of solution to the weak constraint problem, based on representer theory. However, their method is not suitable for the numerical weather prediction problem, because it does not cycle in time. In this paper, the representer method is modified to permit cycling in time, in a manner which is entirely internally consistent. The method was applied to a simple 1-dimensional constituent transport problem where the signal was sampled (perfectly and imperfectly) with various sparse observation network configurations. The cycling representer algorithm discussed here successfully extracted the signal from the noisy, sparse observations.

1. Introduction

The optimal estimation of the initial state is a continuing and fundamental concern in the production of accurate atmospheric forecasts in the troposphere to mesosphere and for all spatial scales from global to convective. Initial state estimates have improved in recent years, in part, due to the new remote sensing technology and its contribution to a greatly increased observational database. However, these new observational technologies have also made the state estimation (or data assimilation problem) much more difficult

because these systems generally sample non-standard variables remotely in a spatially and temporally heterogeneous manner. Fortunately, modern estimation provides a sufficiently powerful formulation that all these instruments can be accommodated in theory, at least, even if practice is much more difficult.

Most data assimilation algorithms are used operationally in a data assimilation cycle, or more accurately, an intermittent data assimilation cycle. In such a cycle, the observations are divided into time intervals (6 h is commonly used) and the assimilation algorithm works on these observations alone. In a real time operation, the assimilation algorithm works only on the latest observation interval. In order to commence each

* Corresponding author.
e-mail: xu@nrlmry.navy.mil

cycle, it is usually necessary to have available a state estimate from the previous cycle and some estimate of the forecast error statistics from the previous cycle. The data assimilation cycle is not an iteration procedure in which the same observations are repeatedly introduced, but a progressive process in which new observations are introduced in every new cycle.

The transformation between such observed variables as channel radiance, bending angles, Doppler shifts and the meteorological variables such as temperature, winds, moisture, trace constituents etc can be accommodated by introducing a forward or instrument model and corresponding tangent and adjoint models. The 3-dimensional variational (3DVAR) algorithm minimizes a three (spatial) dimensional cost function and is sufficiently general to accommodate such operators. This algorithm requires the model forecast, the observations, the observation error statistics, but also the forecast (background) error statistics (at least the first and second moments). The 3DVAR (Parrish and Derber, 1992; Courtier et al., 1998) algorithm does not provide an internal method of estimating the forecast error statistics, and they must be specified in an ad hoc manner. This algorithm is widely used in intermittent cycling data assimilation algorithms for the analysis of global and synoptic scale. It can handle relatively slowly evolving flows and observations systems which sample heterogeneously in space. However, highly intermittent flows which are not governed by simple balance relationships and observation systems which sample irregularly in time will not be well-accommodated within an intermittent 3-dimensional framework.

Instruments and meteorological phenomena of this kind can be accommodated, in principle, within a 4-dimensional variational framework. Such algorithms attempt to minimize a 4-dimensional cost function (over some time interval) subject strongly or weakly to a constraint provided by the prediction model. The strong constraint formulation of this problem has received a great deal of attention over the past few years because it is the least demanding in computer resources of all the 4-dimensional variational algorithms. This strong constraint algorithm, known popularly as the 4DVAR algorithm, has been successfully applied in a simplified manner to the problem of estimating the initial

state for global numerical prediction (Rabier et al., 1998).

The strong constraint formulation (perfect model assumption) of the standard 4DVAR algorithm makes the method relatively tractable, because the minimization only needs to be carried out with respect to the model variables at a single timestep (usually the first), as all other timesteps are strongly linked through the model (Navon et al., 1992; Zupanski, 1993; Courtier et al., 1994). In other words, the control vector for the minimization is only of length the number of gridpoints, rather than the number of gridpoints times the number of timesteps as it is for the generalized inverse problem. However, the strong constraint method has 2 serious weaknesses. Firstly, the perfect model assumption is frequently inappropriate (Daley, 1992b). It is true that there have been several attempts to account for model error in this algorithm (Derber, 1989; Zupanski, 1997). However, these schemes are not based on the more general weak constraint formulation, but are essentially extensions of the strong constraint algorithm. Secondly, the strong constraint algorithm, like the 3DVAR algorithm, does not provide an internal estimate of the forecast error statistics at the beginning of the interval. Thus, this algorithm is not inherently cyclable. In practice the strong constraint algorithm can be cycled (if not necessarily soundly from a theoretical point of view), by either specifying the appropriate initial forecast error statistics (as is done at present at ECMWF) or by introducing a parallel cycle employing some sort of approximation to a Kalman filter to temporally evolve the required forecast error statistics (Fisher and Courtier, 1995).

There are, however, 4-dimensional algorithms, which do not suffer, in principle, from either of these defects of the strong constraint 4DVAR algorithm. They are, unfortunately, and it is important to stress this, very much more expensive computationally, than the strong constraint 4DVAR algorithm. For this reason, they are not likely to become operational soon, but it is not too early to begin to examine their properties and potential.

One such algorithm is the minimization of a weak constraint 4-dimensional cost function with respect to the whole model state instead of only to one single time step by the method of representers. This algorithm has been discussed in a

generally oceanographic context in a series of papers by Andrew Bennett and co-workers (Bennett, 1992; Bennett et al., 1996; Bennett et al., 1997). In this algorithm, there is no perfect model assumption, in fact the model error covariance can be both non-stationary and non-homogeneous. However, all the applications of Bennett (above) have been non-cycling applications, useful for determining state and parameter estimates for the ocean, but not for operational weather prediction. Fortunately, the general representer algorithm, unlike the 4DVAR algorithm, actually provides an internally consistent means of generating at the end of each cycle, not only a state estimate, but an estimate of the necessary forecast error covariances. These are sufficient to commence the next cycle.

In this paper, we will develop the cycling representer algorithm and examine its properties with respect to a simple 1-dimensional constituent transport problem. This is the first step in a long sequence of steps to be taken towards the goal of developing a tractable weak constraint 4-dimensional variational algorithm for atmospheric data assimilation for the meso- and convective scales.

In Section 2, we introduce the generalized 4-dimensional cost function with the forecast model as a weak constraint, and its minimization using an expansion of representer functions. In Sections 3 and 4, we apply this single cycle representer algorithm to a simple 1-dimensional constituent transport problem. In Section 5, we introduce the cycling representer algorithm and apply it to the same transport problem. The experimental results are summarized in Section 6.

2. The non-cycling representer algorithm

The derivation of the (non-cycling) representer algorithm follows Bennett (1992), with one significant difference. In oceanography it is customary to derive data assimilation algorithms in a form which is continuous in space and time and makes use of multiple integrals, convolutions and Dirac delta functions. In meteorology, it is customary to first discretize the relevant equations in space and time, resulting in linear algebraic equations (Amodei, 1995; Courtier, 1998). We will follow the meteorological convention, and thus begin with

discretized equations. The generalized inverse problem can be posed as the minimization of the cost (objective) function J :

$$J = J_0^b + J^q + J^r, \quad (2.1)$$

where

$$\begin{cases} J_0^b = \frac{1}{2} [\mathbf{x}_0^b - \mathbf{x}_0]^T [\mathbf{B}_0^b]^{-1} [\mathbf{x}_0^b - \mathbf{x}_0], \\ J^q = \frac{1}{2} \sum_{n=1}^N \sum_{n'=1}^N [\mathbf{x}_n - \mathcal{M}(\mathbf{x}_{n-1})]^T \mathbf{Q}_{nn'}^{-1} \\ \quad \times [\mathbf{x}_{n'} - \mathcal{M}(\mathbf{x}_{n'-1})], \\ J^r = \frac{1}{2} [\mathbf{y} - \mathcal{H}(\mathbf{x})]^T \mathbf{R}^{-1} [\mathbf{y} - \mathcal{H}(\mathbf{x})]. \end{cases} \quad (2.2)$$

J_0^b , J^q and J^r are the cost functions of the initial, model and observation errors, respectively. For a limited area model an additional cost function is needed when the lateral boundary conditions are imperfect. These cost functions are all quadratic, consistent with the assumption of normal error probability distributions. The superscripts “b”, “q” and “r” refer to the background, model and observations respectively. The subscripts “0”, “n” and “n’” refer to the initial, n th and n' th model time-steps. The model $\mathcal{M}$ is assumed to have I gridpoints (or more precisely the number of gridpoints times the number of prediction variables) and N timesteps with $1 \leq n, n' \leq N$. $\mathbf{x}$ is the true state vector in model space (of length $N \cdot I$). The background $\mathbf{x}^b$ is the model state that would be obtained by integrating the model $\mathcal{M}$ from the initial state estimate $\mathbf{x}_0^b$ without any observation. $\mathbf{B}_0^b = \langle (\mathbf{x}_0^b - \mathbf{x}_0)(\mathbf{x}_0^b - \mathbf{x}_0)^T \rangle$ is the symmetric $I \times I$ covariance matrix of initial errors and must be specified. Here $\langle \rangle$ indicates expectation operator. $\mathbf{y}$ is the vector of observations of length K which include all observations that occur between $n = 1$ and $n = N$. $\mathbf{R}$ is the $K \times K$ symmetric matrix of observation error covariances which must also be specified. $\mathcal{H}$ is the non-linear forward operator, which connects the model space to the observation space. The (non-linear) model $\mathcal{M}$ is assumed to be imperfect (weak constraint formulation), thus requiring the model misfit term J^q . $\mathbf{Q}_{nn'}$ is the $N \cdot I \times N \cdot I$ symmetric model error covariance matrix. If the model error is not serially correlated then all but the $I \times I$ blocks where $n = n'$ will be zero. We note, that by definition, $\mathbf{x}_n^b - \mathcal{M}(\mathbf{x}_{n-1}^b) = 0$. Superscript “-1” indicates

matrix inverse and superscript “T” indicates matrix transpose.

Let $\mathbf{x}^a$ be the analysis (or the estimated state), which is the value of $\mathbf{x}$ which minimizes J and for linear $\mathcal{M}$ and $\mathcal{H}$ and normal observation and background error probability distributions is also the maximum likelihood estimate. Taking the first variation of (2.1) yields,

$$\begin{aligned} \delta J = & -[\delta \mathbf{x}_0^a]^T [\mathbf{B}_0^b]^{-1} [\mathbf{x}_0^b - \mathbf{x}_0^a] + \sum_{n=1}^N \sum_{n'=1}^N \\ & \times [\delta \mathbf{x}_n^a - \mathbf{M}_{n-1} \delta \mathbf{x}_{n-1}^a]^T \mathbf{Q}_{nn'}^{-1} [\mathbf{x}_{n'}^a - \mathcal{M}(\mathbf{x}_{n'-1}^a)] \\ & - [\mathbf{H} \delta \mathbf{x}^a]^T \mathbf{R}^{-1} [\mathbf{y} - \mathcal{H}(\mathbf{x}^a)], \end{aligned} \quad (2.3)$$

where the matrix $\mathbf{M}_{n-1} = \partial \mathcal{M}(\mathbf{x}_{n-1}) / \partial \mathbf{x}_{n-1}$ evaluated at $\mathbf{x} = \mathbf{x}^a$ is the $I \times I$ Jacobian matrix corresponding to the (possibly) non-linear model operator $\mathcal{M}(\mathbf{x}_{n-1}^a)$. If $\mathcal{M}(\mathbf{x}_{n-1}^a)$ is linear, then $\mathbf{M}_{n-1} \mathbf{x}_{n-1}^a$ and $\mathcal{M}(\mathbf{x}_{n-1}^a)$ are the same. The $N \cdot I \times K$ matrix $\mathbf{H} = \partial \mathcal{H}(\mathbf{x}) / \partial \mathbf{x}$, evaluated at $\mathbf{x} = \mathbf{x}^a$, is the Jacobian matrix corresponding to the (possibly) non-linear forward operator $\mathcal{H}$. If $\mathcal{H}$ is linear, then $\mathbf{H}\mathbf{x}$ and $\mathcal{H}(\mathbf{x})$ are the same.

We next introduce the adjoint field. Denote the adjoint field $\vec{\lambda} = (\vec{\lambda}_1 \cdots \vec{\lambda}_N \cdots \vec{\lambda}_{N+1})^T$ as a vector of length $(N+1) \cdot I$. The sub-vector $\vec{\lambda}_n$ (of length I) of $\vec{\lambda}$ for the n th timestep, is defined as follows:

$$\vec{\lambda}_n \equiv \sum_{n'=1}^N \mathbf{Q}_{nn'}^{-1} [\mathbf{x}_{n'}^a - \mathcal{M}(\mathbf{x}_{n'-1}^a)], \quad (2.4)$$

where, $1 \leq n \leq N$. For later convenience, we define $\vec{\lambda}_{N+1} = 0$.

Substituting eqs. (2.4) into (2.3) and rearranging terms, gives,

$$\begin{aligned} \delta J = & -[\delta \mathbf{x}_0^a]^T [\mathbf{B}_0^b]^{-1} [\mathbf{x}_0^b - \mathbf{x}_0^a] \\ & + \sum_{n=1}^N [\delta \mathbf{x}_n^a]^T \vec{\lambda}_n - \sum_{n=0}^{N-1} [\delta \mathbf{x}_n^a]^T \mathbf{M}_n^T \vec{\lambda}_{n+1} \\ & - [\delta \mathbf{x}^a]^T \mathbf{H}^T \mathbf{R}^{-1} [\mathbf{y} - \mathcal{H}(\mathbf{x}^a)], \end{aligned} \quad (2.5)$$

where $\mathbf{M}_n^T$ is the transpose (or the adjoint) of $\mathbf{M}_n$ and $\mathbf{H}^T$ is the transpose of $\mathbf{H}$. It is important to note that $\mathbf{H}^T$ connects the observation space to the model space.

Setting the partial derivatives $\partial J / \partial \mathbf{x}_a^n = 0$, $1 \leq n \leq N$, or equivalently, zeroing the coefficients of $\delta \mathbf{x}_n^a$ in (2.5) gives

$$\vec{\lambda}_n - \mathbf{M}_n^T \vec{\lambda}_{n+1} = \{\mathbf{H}^T \mathbf{R}^{-1} (\mathbf{y} - \mathcal{H}(\mathbf{x}^a))\}_n, \quad (2.6)$$

where $1 \leq n \leq N$ and $\vec{\lambda}_{N+1} = 0$.

$$\{\mathbf{H}^T \mathbf{R}^{-1} (\mathbf{y} - \mathcal{H}(\mathbf{x}^a))\}_n$$

is a vector of length I , valid at timestep n .

Inverting eq. (2.4) yields,

$$\mathbf{x}_n^a - \mathcal{M}(\mathbf{x}_{n-1}^a) = \sum_{n'=1}^N \mathbf{Q}_{nn'} \vec{\lambda}_{n'}, \quad (2.7)$$

where $1 \leq n \leq N$. The initial condition for eq. (2.7) may be obtained as:

$$\mathbf{x}_0^a = \mathbf{x}_0^b + \mathbf{B}_0^b \mathbf{M}_0^T \vec{\lambda}_1, \quad (2.8)$$

by setting the partial derivatives $\partial J / \partial \mathbf{x}_a^0 = 0$, or equivalently, zeroing the coefficients of $\delta \mathbf{x}_0^a$ in eq. (2.5).

The system (2.6)–(2.8) forms the discretized Euler–Lagrange equations. This is a coupled system and cannot be solved with a simple direct integration. However, several algorithms have been proposed to decouple the Euler–Lagrange equations (see Bennett, 1992 and 1997). Among them is the method of representers. In the following derivation, we assume both the model and the forward operator are linear. We will address non-linear observation and prediction models at a later date, but suffice it to say that Bennett (1997) has begun to address the non-linear problem. With the linear assumption, we may replace $\mathcal{M}(\mathbf{x}_{n'-1}^a)$ with $\mathbf{M}_{n'-1} \mathbf{x}_{n'-1}^a$ in eq. (2.4) and replace $\mathcal{H}(\mathbf{x}^a)$ with $\mathbf{H}\mathbf{x}^a$.

Before going any further, we introduce the following notation,

$$\left\{ \begin{array}{l} \vec{\theta}_k \text{ is a vector of length } (N+1) \cdot I, \\ \{\vec{\theta}_k\}_n \text{ is a vector of length } I \text{ valid} \\ \quad \text{at model timestep } n, \\ \{\vec{\theta}_k\}_{i,n} \text{ is a scalar, a single element of } \vec{\theta}_k, \\ \quad \text{at the model grid point } (i, n), \end{array} \right. \quad (2.9)$$

where subscript “ k ” is an index for the k th observation.

With both $\mathcal{H}$ and $\mathcal{M}$ being linear operators, we next introduce the representer functions in vector notation, $\vec{\gamma}_k = (\{\vec{\gamma}_k\}_0 \cdots \{\vec{\gamma}_k\}_n \cdots \{\vec{\gamma}_k\}_N)^T$, which is a vector of length $(N+1) \cdot I$, where $1 \leq k \leq K$ is the index of the k th observation. Each representer function has an adjoint $\vec{\alpha}_k = (\{\vec{\alpha}_k\}_1 \cdots \{\vec{\alpha}_k\}_n \cdots \{\vec{\alpha}_k\}_{N+1})^T$, which can also be expressed as a vector of length $(N+1) \cdot I$ and

satisfies,

$$\{\tilde{\alpha}_k\}_n - \mathbf{M}_n^T \{\tilde{\alpha}_k\}_{n+1} = \{\mathbf{H}_k^T\}_n, \quad (2.10)$$

where $1 \leq n \leq N$ and $\{\tilde{\alpha}_k\}_{N+1} = 0$. $\mathbf{H}_k^T$ is the k th column (of length $N \cdot I$) of $\mathbf{H}^T$ and projects the k th observation from the observation space to the gridpoints in the model space. When the k th observation is collocated with a model gridpoint, then the element of $\mathbf{H}_k^T$ at the i th gridpoint and n th timestep takes a very special form,

$$\{\mathbf{H}_k^T\}_{i,n} = \delta_{i,i_k} \delta_{n,n_k}, \quad (2.11)$$

where the point (i_k, n_k) is the location of the k th observation in the model space and δ_{i,i_k} and δ_{n,n_k} are the Kronecker deltas, respectively.

We can now integrate eq. (2.10) backward from $n = N$ to $n = 1$ to obtain the adjoint field, $\tilde{\alpha}_k$, for the k th observation. Once $\tilde{\alpha}_k$ is known, we may then solve for $\tilde{\gamma}_k$ by,

$$\{\tilde{\gamma}_k\}_n - \mathbf{M}_{n-1} \{\tilde{\gamma}_k\}_{n-1} = \sum_{n'=1}^N \mathbf{Q}_{nn'} \{\tilde{\alpha}_k\}_{n'}, \quad (2.12)$$

where $1 \leq n \leq N$. The initial condition of eq. (2.12) is,

$$\{\tilde{\gamma}_k\}_0 = \mathbf{B}_0^T \mathbf{M}_0^T \{\tilde{\alpha}_k\}_1. \quad (2.13)$$

The optimally estimated state (the analysis) $\mathbf{x}^a$ may now be expressed by the background plus a correction, at each gridpoint and at each timestep, which is a linear combination of the representer functions,

$$\mathbf{x}^a = \mathbf{x}^b + \sum_{k=1}^K \tilde{\gamma}_k \beta_k = \mathbf{x}^b + \Gamma^T \tilde{\beta}, \quad (2.14)$$

where $\tilde{\beta} = (\beta_1 \cdots \beta_k \cdots \beta_K)^T$ is a vector of length K of unknown (scalar) coefficients. β_k is the representer coefficient for the k th representer function $\tilde{\gamma}_k$. We define the $N \cdot I \times K$ matrix $\Gamma^T = (\tilde{\gamma}_1 \cdots \tilde{\gamma}_k \cdots \tilde{\gamma}_K)$. Note that Γ is actually the a priori error covariance between the K observations (in observation space) and the $N \times I$ gridpoints (in model space). Once the representer coefficient vector $\tilde{\beta}$ is known, we can then calculate the optimal solution $\mathbf{x}^a$ using eq. (2.14). In Section 8, it is shown that,

$$\tilde{\beta} = [\mathbf{R}_e + \mathbf{R}]^{-1} [\mathbf{y} - \mathbf{H}\mathbf{x}^b], \quad (2.15)$$

where $\mathbf{R}_e = \mathbf{H}\Gamma^T$ is the $K \times K$ representer matrix and the k th column of the matrix is the same as $\mathbf{H}\tilde{\gamma}_k$. $\mathbf{y} - \mathbf{H}\mathbf{x}^b$ is the innovation vector (sometimes referred to as the observation increment). Note that the representer matrix, $\mathbf{R}_e$, is only determined

by the locations of the observations and by the initial and model error covariances. Combining eqs. (2.14) and (2.15), we may now express the optimally estimated state as,

$$\mathbf{x}^a = \mathbf{x}^b + \Gamma^T [\mathbf{R}_e + \mathbf{R}]^{-1} [\mathbf{y} - \mathbf{H}\mathbf{x}^b]. \quad (2.16)$$

Eq. (2.16) is formally identical to the observation space form of the 3DVAR algorithm (except that the observations are distributed in 4 dimensions). This completes our derivation of the (single cycle) representer algorithm and we will now apply it to a simple inverse problem.

The representer algorithm is more costly than present day 3DVAR and strong constraint 4DVAR algorithms. The solution of eq. (2.16) once Γ has been obtained is a purely observation space calculation and is thus no worse than the equivalent calculation performed in the 3DVAR algorithm. The expense comes in determining the representers themselves which require K forward and K back integrations. This would be expensive but is embarrassingly parallel. Furthermore, Bennett (1997) has discussed ways to perform this calculation more efficiently. However, it should always be kept in mind that this algorithm is not intended to compete with present algorithms, it is aimed at the future, when the potential of present algorithms has been exhausted.

3. Experimental configuration

We will apply the algorithm of Section 2 to a simple 1-dimensional constituent transport model with cyclic boundary conditions. Despite the simplicity of the model, this is a very rich system and there are many aspects of the problem that we could choose to examine. We could examine the effect of the model error, the specification of the model error covariance, $\mathbf{Q}$ and the effect of misspecification of the model error covariance (i.e. inconsistency between the actual model error and our perception of the model error characteristics). In the same way, we could examine the effect of different spatial and or temporal observation networks, observation errors, specification of observation error covariances $\mathbf{R}$ and the misspecification of the observation error covariances. Finally, we could examine the initial error, the specification of the initial background error

covariance $\mathbf{B}_0^b$ and the effect of mis-specifying the initial error covariance.

This is too much for one article and a number of these aspects have been previously examined. Thus, the effect of model error covariances have been examined by Daley (1992a,b), Mitchell and Daley (1997). Since the main purpose of this article is to examine a cycling representer algorithm, we will concentrate our attention on the initial field, the specification of the initial background error covariance $\mathbf{B}_0^b$ and the effect of mis-specification of this covariance. The experiments will be conducted using 2 completely different observational networks, with both perfect and imperfect observations. However, the observation error statistics will always be appropriately specified, and we will not consider the effect of mis-specification of the observation error statistics.

We will describe first the truth, the model and model error covariance, then the observation network and errors and finally the initial conditions and initial error covariance. It might be noted that certain choices of observation networks with the constituent transport eq.(3.1) can lead to pathological results (Callies and Eppel, 1995; Mitchell and Daley, 1997). All experimental parameters have been chosen to avoid such non-physical cases.

3.1. The truth, the model and model error covariance specification

The constituent $c(x, t)$ is a function of space (x) and time (t). In this section “ x ” is the independent spatial variable and should not be confused with the bold $\mathbf{x}$ used in Section 2. The true time dependent state are determined by integrating the following eq.(3.1) from a specified initial state $c(x, 0)$. The spatial domain was periodic with $0 \leq x \leq 2\pi$, and the true state always satisfied this boundary condition,

$$\frac{\partial c(x, t)}{\partial t} + U \frac{\partial c(x, t)}{\partial x} = 0. \quad (3.1)$$

The true initial state was assumed to be of the form

$$c(x, 0) = \sin(px), \quad (3.2)$$

where p is an integer. In these experiments, a comparison with the true state is always possible.

The model is in finite difference form with the

second order approximation (in space and time) to the continuous eq. (3.1). There were $I = 21$ model gridpoints in the domain, giving a grid-length $\Delta x = 2\pi/I$. The timestep Δt was chosen to give a Courant number $U\Delta t/\Delta x = 0.5$, which was well within the stable regime of the leapfrog algorithm. This implied that there would be $N = 43$ timesteps for any spatial structure to pass through the domain. The integration was started off with a forward timestep.

Now, model error depends both on the model and the physical phenomena to be simulated. The leapfrog algorithm is reasonably accurate for scales which are much larger than the gridlength Δx , but has serious phase and amplitude errors for small scale phenomena. Since model error was not the focus of this investigation, we decided to make the model error very small, by defining the truth to be a large-scale structure. Thus, the true initial state was specified by setting $p = 1$ in (3.2). The true state after 43 timesteps (one period) is shown by the solid line in Fig. 1, as a function of x .

Also shown in Fig. 1, by the dotted line is the model forecast at timestep 43, starting from eq. (3.2). It can be seen that the error is very small and thus we can regard this model as almost perfect for the simulation of this wavenumber 1 phenomenon.

The model error, while small, is non-zero and therefore we must specify the model error covariance $\mathbf{Q}$. In reality, the model error covariance can be both spatially and serially correlated (Daley, 1992a,b) and can also be correlated with the signal (i.e., truth) (Mitchell and Daley, 1997). In this

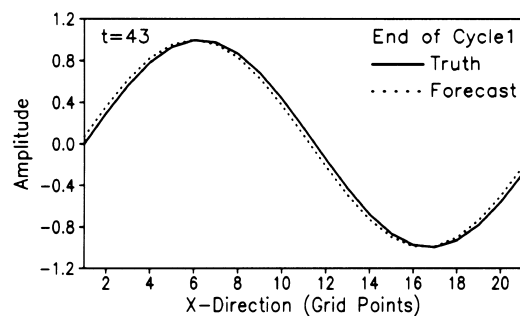


Fig. 1. True state (solid) and forecast state (dotted) after integration from the true initial conditions as a function of x .

case, we specified the model error covariances as:

$$[\mathbf{Q}]_{i,n,i',n'} = q \exp\left(-\frac{|i-i'|^2}{L_q^2} - \frac{|n-n'|^2}{\tau_q^2}\right), \quad (3.3)$$

where q is the amplitude which represents the constant variance of the model errors, i and i' are the spatial gridpoint indices, n and n' are the temporal indices in terms of numbers of time step L_q and τ_q are the spatial and temporal decorrelation length scales of the model error covariances respectively. All the length scales are in terms of model gridpoints.

3.2. Assimilation cycles, observation networks, and observation error specification

For this study, a single assimilation cycle will be defined to consist of a single period. Thus, the true state for each timestep of a given cycle will be the same as the true state at the same timestep of any other cycle. All observations are at gridpoints and we define 2 observation networks.

NETWORK A

There are no observations except at timestep 22 of every cycle, when there is an observation at every gridpoint. This network gives maximum observation density at one time per cycle. This network is illustrated (as a function of space) in Fig. 6a. The value of the k th column of the transpose of the forward operator, eq. (2.11), at gridpoint (i, n) is given by $\{\mathbf{H}_k^T\}_{i,n} = \delta_{i,i_k} \delta_{n,n_k}$, where, $n_k = 22$, $i_k = k$, for $1 \leq k \leq K$; $1 \leq i \leq I$ and $1 \leq n \leq N$.

NETWORK B

There are no observations anywhere, except at gridpoint 10, at every other timestep. This network has very low spatial density, but high temporal density. This network is illustrated (as a function of time) in Fig. 6b. The value of the k th column of the transpose of the forward operator, eq. (2.11), at gridpoint (i, n) is now given by $\{\mathbf{H}_k^T\}_{i,n} = \delta_{i,i_k} \delta_{n,n_k}$, where, $i_k = 10$, $n_k = 2k$, for $1 \leq k \leq K$; $1 \leq i \leq I$ and $1 \leq n \leq N$.

Note that both networks have $K = 21$ observations per cycle. Both observation networks are observable in the sense defined by Cohn and Dee (1988). See also Menard (1994) and Daley (1997). There were 2 choices of observation error.

Perfect observations — in this case the observed

values were the same as the true values, the observation error covariance $\mathbf{R}$ was assumed to be zero and the analysis were fitted exactly to the observations.

Imperfect observations — the observations were assumed to have spatially and serially uncorrelated errors chosen from a normal distribution with a standard deviation equal to 0.1. The observation error covariance $\mathbf{R} = \varepsilon_r^2 \mathbf{I}$, where $\mathbf{I}$ is the identity matrix and $\varepsilon_r = 0.1$.

3.3. Initial conditions and specification of initial error covariance

The true state at $t = 0$ (eq. (3.2)) can be seen by the solid curve in Fig. 2a. For all experiments, the initial state is chosen to be a very bad estimate, namely $c(x, 0) = 0$. The initial state is shown by the dotted curve in Fig. 2a.

The specification of the initial error covariance was the heart of the present experiments, and a number of different prescriptions were tested. However, the initial error covariance always had the simple form,

$$[\mathbf{B}_0^b]_{i,i'} = b_0 \exp\left(-\frac{|i-i'|^2}{L_{b_0}^2}\right), \quad (3.4)$$

where, b_0 is the amplitude which represent the constant variances of initial and model errors, i and i' are the spatial indices in terms of number of gridpoint, L_{b_0} is the decorrelation length scale for the initial error covariance and is in terms of model gridpoints.

4. Single cycle experimental results

There were 8 experiments performed over all and these experiments are tabulated in Table 1. Experiments 1–4 can be described as single cycle experiments using the representer theory of Section 2. The observations were always perfect for experiments 1–4.

4.1. Experiments 1 and 2 — single cycle with well-specified initial error covariances

Experiments 1 and 2 were essentially tuning experiments, to determine appropriate values of the parameters q , L_q and τ_q for the model error covariance formulation (3.3) and the parameters

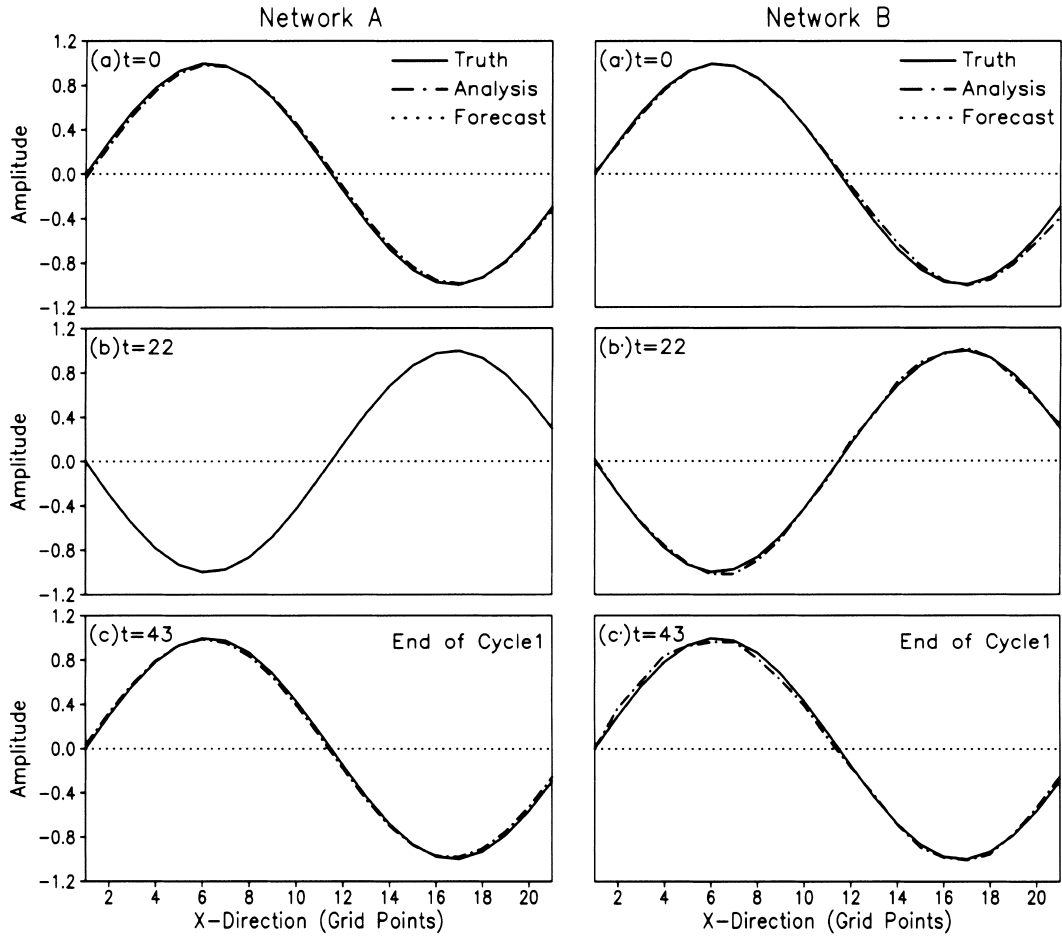


Fig. 2. True state (solid), forecast state (dotted) and analyzed state (dash-dot) at $t = 0$ (panels a, a'), $t = 22\Delta t$ (panels b, b') and $t = 43\Delta t$ (panels c, c'), for perfect observations and optimal initial state covariance estimate. Network A is left panels and Network B, right panels.

b_0 and L_{b_0} for the initial error covariance formulation (3.4). The 2 experiments were identical, except that experiment 1 used Network A and experiment 2 used Network B. In both cases, the observations were perfect as above.

The 5 parameters q , L_q , τ_q , b_0 and L_{b_0} were first specified arbitrarily (a null hypothesis). The representer algorithm of Section 2 was then run for one cycle and the innovation $y - Hx^b$ collected at each observation point. We then calculated the minimum value of the cost function (see eq. (2.1)), $J_{\min} = [y - Hx^b]^T (R_e + R)^{-1} [y - Hx^b]$. The χ^2 test was then used to check and modify our hypotheses (see more detail in Bennett, 1997).

Note that we used the perfect observation case, $R = 0$, to tune the parameters of the model and initial background error covariance. This experiment was performed for both networks and a single choice of the 4 parameters was made for both networks.

For the model error parameters, the following values were chosen, $q = 0.0001$, $L_q = 3$ and $\tau_q = 0$. These parameter values were used for all experiments with both networks. Note, that this tuning method only depends on the innovations and the inverse of the representer matrix which do not require knowledge of the state.

The specified initial error covariance, of the

form eq. (3.4), with parameters $b_0 = 1$ and $L_{b_0} = 6$, was hopefully optimally consistent with the actual zero initial conditions specified in Section 3. This specified initial error covariance was then used in experiments 1 and 2, whose results are shown in Fig. 2, in the same format as Fig. 1. In panels (a–c), we show the results for Network A (all gridpoints sampled perfectly at timestep 22) and in panels (a'–c') for Network B (gridpoint 10 sampled perfectly at every other timestep). Panels (a, a') are $t = 0$, panels (b, b') are $t = 22$ timesteps and panels (c, c') are the end of the cycle ($t = 43$ timesteps). The true state is indicated by the solid curve. The dotted curve (marked forecast) is the model forecast from the initial conditions ($c(x, 0) = 0$), in the absence of any observations. The dot-dash curve (marked “analysis”) is the state estimate after the completion of the assimilation.

Examining first panels (a–c), it can be seen from panel (b), that the analysis and truth are identical, because we know the true state everywhere at timestep 22 with Network A. However, examination of panels 2a and 2c (and any other time) shows that the algorithm is essentially able to recover the true state everywhere and at every timestep after a single cycle of assimilation.

Panels (a'–c') describe a very different network, and also show that the true state has essentially been recovered by the algorithm. In this case, the estimate at gridpoint 10 is perfect at every even timestep. The errors are a little larger than in

panels (a–c), but this is not surprising, considering the true state is only sampled at a single observation location. It might be noted, that modifying Network B, so that there were observations at gridpoint 10, every timestep, instead of every other timestep, did not improve the assimilation, indicating that 21 observations were sufficient (as might be expected from observability theory).

4.2. Experiments 3 and 4 — single cycle with poorly specified initial error covariances

Experiments 1 and 2 were intended to demonstrate how the representer algorithm can recover the true solution when the observation, model and initial error covariance are optimally specified. Unfortunately, this would be difficult to do under realistic conditions and therefore it is important to see what happens when the covariances are mis-specified. As discussed earlier, we shall concentrate on mis-specification of the initial error covariance and not alter the model and observation error covariances. The observations were perfect, as before.

In experiment 3 (Table 1), we used Network A, but assumed that the initial error covariance was zero everywhere. In other words, we assumed that the initial state $c(x, 0) = 0$ was perfect, when in fact it was completely different from the true initial state (3.2). The results of this experiment are shown in Figs. 3a–c, in the same format as Fig. 1. It can be seen that at $t = 0$, the specification that

Table 1. *Experimental configuration*

Experiment	Network	Observations	Initial error covariance	Cycling	Figs.
experiment 1	A	perfect	consistent	no	Figs. 2a–c
experiment 2	B	perfect	consistent	no	Figs. 2a'–c'
experiment 3	A	perfect	inconsistent (zero)	no	Figs. 3a–c
experiment 4	B	perfect	inconsistent (diagonal)	no	Figs. 3a'–c'
experiment 5	A	perfect	inconsistent (zero)	yes	Figs. 3d–e
experiment 6	B	perfect	inconsistent (diagonal)	yes	Figs. 3d'–e'
experiment 7	A	imperfect	inconsistent (zero)	yes	Figs. 5a–d
experiment 8	B	imperfect	inconsistent (diagonal)	yes	Figs. 7a–e
					Figs. 7a'–e'

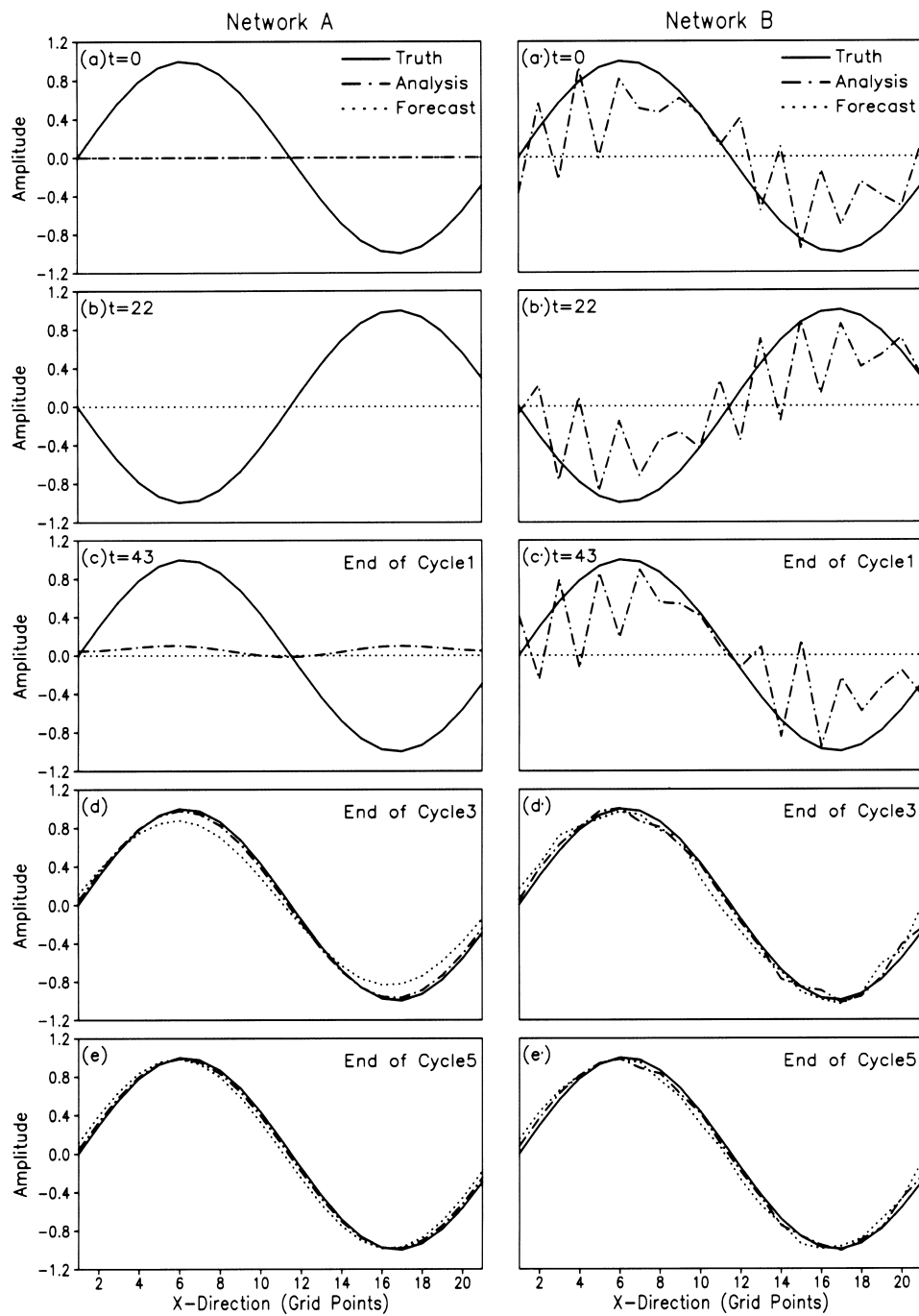


Fig. 3. True, forecast and analyzed state for Networks A and B, with imperfectly specified initial error covariances and perfect observations for cycle 1 (first 3 panels) and at the end of cycles 3 and 5 (last 2 panels).

the initial state was perfect, caused the algorithm to draw towards the complete erroneous initial state $c(x, 0) = 0$. At timestep 22, where there is a (perfect) observation at every gridpoint, the algorithm fits the observations (and therefore, the truth) very well. However, by the end of the cycle (timestep 43), the analysis is again very poor, although slightly better than the forecast. This result clearly shows that a poor estimate of the initial error covariance, when coupled with a poor initial state estimate, yields very poor results, even though the model is good, the model error has been well tuned, the observations are perfect and the system is observable.

A second similar experiment 3a (not shown here) was also performed with Network A. In this case, we used the tuned initial error covariance with $b_0 = 1$, except we made the diagonal approximation, which is set $L_{b_0} = 0$. The results in this case, were almost as good as those in experiment 1 (Figs. 2a–c). Thus, for Network A, which sampled perfectly and completely at a single time in the cycle, the spatial correlation of the initial error covariance had little effect on the assimilation.

This result implies that the precise specification of the initial error covariance at the beginning of each cycle might not be very important in weak or strong constraint algorithms, if the domain is completely sampled at some time during the assimilation interval. Unfortunately, in reality, the atmosphere is never sampled very completely.

In experiment 4, we used Network B, but assumed that the initial error covariance was specified using eq. (3.4), with $b_0 = 1$ and $L_{b_0} = 0$, that is the diagonal approximation. The results are shown in Figs. 3a'–c'. In this case, the analysis is perfect at gridpoint 10 (at even timesteps), as would be expected, but is really very poor everywhere else, for the whole cycle. Thus, the diagonal approximation for the initial error covariance, which worked well for Network A, failed when used with Network B.

5. Multi-cycle algorithm — theory and experimental results

The representer algorithm developed in Section 2 is not suitable for atmospheric data assimilation in a numerical weather prediction

(NWP) context. As noted in Section 1, in NWP there is a continuous cycle of observation, assimilation and forecast (Daley, 1997, Fig. 3). Thus, before commencing a new cycle, there is always available an initial state estimate obtained from the previous cycle. However, it is also necessary to have available an estimate of the error covariance of the initial state estimate. This error covariance estimate was not derived in Section 2, but the representer algorithm is sufficiently powerful to provide such a covariance in an internally consistent manner. Then, the initial state estimate and the corresponding error covariance estimate can be used to initiate the next cycle. This situation is different from the strong constraint 4DVAR algorithm, which provides a state estimate at the beginning of each cycle, but does not provide an estimate of the initial error covariance. (Of course, there are operational ways of estimating such error covariance estimates, as discussed in Section 1.)

5.1 Theory

There are approximate methods to estimate the initial error covariance for the representer algorithm, which we will discuss later in this section. For the present, however, we will introduce the internally-consistent method of calculating these covariances. The cycling representer theory of Section 2 requires the following additions in order to generate an internally consistent initial error covariance to commence each new cycle. Following Bennett (1992, 1997), we can express the estimation of the a posteriori (analysis) error covariance as the expectation value of the outer product,

$$\mathbf{B}^a = \langle (x - x^a)(x - x^a)^T \rangle = \mathbf{B}^b - \Gamma^T (\mathbf{R}_e + \mathbf{R})^{-1} \Gamma, \quad (5.1)$$

where $\mathbf{B}^a$ is the posterior analysis error covariance, and,

$$\mathbf{B}^b = \langle (x - x^b)(x - x^b)^T \rangle, \quad (5.2)$$

is the background error. Both $\mathbf{B}^a$ and $\mathbf{B}^b$ are defined in gridpoint space. Γ^T , $\mathbf{R}_e$, and $\mathbf{R}$ are the same as defined in Section 2 and are available from the regular representer calculation. Note that $\mathbf{R}_e$ is the background error covariance in observation space. While eq. (5.1) might appear unfamiliar, it is in fact a well-known equation used in

Kalman filter theory and is a form of eq. (17) of Daley (1997). Note, however, that both the observations and error covariances apply to the whole cycle, not just to a particular time.

We will use eq. (5.1) to estimate the a posteriori (analysis) error covariance at the end of the cycle and it will be then available as the a priori (initial) error covariance at the beginning of the next cycle. Both $\mathbf{B}^a$ and $\mathbf{B}^b$ involve correlations between every gridpoint and every time during the cycle. However, we do not need to know $\mathbf{B}^a$ or $\mathbf{B}^b$ for all times during the cycle or the cross-correlations between gridpoints at different times, we only need to know the correlations between gridpoints at the final time. Thus, to make the calculation of $\mathbf{B}^b$ at final time, everything is already available except an estimate of $\mathbf{B}^b$ (i.e., the a priori correlation between gridpoints at the end of the cycle).

The special form of (5.1), appropriate for this calculation we will write as,

$$\mathbf{B}_{NN}^a = \mathbf{B}_{NN}^b - \Gamma_N(\mathbf{R}_e + \mathbf{R})^{-1}\Gamma_N^T, \quad (5.3)$$

where $\mathbf{B}_{NN}^a$ and $\mathbf{B}_{NN}^b$ are $I \times I$ error covariances at the N th timestep (only involving interactions between gridpoints at the N th timestep). The $K \times I$ matrix $\Gamma_N = (\{\gamma_1\}_N \cdots \{\gamma_k\}_N \cdots \{\gamma_K\}_N)^T$ is the standard representer field (of Section 2) at the N th timestep. Thus, Γ_N involves interactions between all the observations during the interval and all the gridpoints at $n = N$. Note that although, Γ_N , $\mathbf{R}_e$ and $\mathbf{R}$ refer to observations which occur at any time during the interval, $\mathbf{B}_{NN}^a$ and $\mathbf{B}_{NN}^b$ refer only to the last timestep. As noted above, everything is available except $\mathbf{B}_{NN}^b$.

To calculate $\mathbf{B}_{NN}^b$, we assume that there is a pseudo-observation at each of the I gridpoints at time $n = N$. These pseudo-observations have only 2 properties which are of interest. (1) Their locations (at gridpoints). (2) The “observed” variables are the analyzed variables. This means that the forward operator for each of these pseudo-observations is the identity matrix (see eq. (2.11)). It is important to note that we do not need to specify either the observations or the observation error characteristics of these pseudo-observations. Thus, in a certain sense, these are not really observations at all, but probes for each of the gridpoints and each of the analyzed variables.

We then use the representer mechanics of Section 2 to calculate a different set of representer. Thus, using these I pseudo-observations we integrate

eq. (2.10), backwards (with $\mathbf{H}$ defined by eq. (2.11)) to obtain a new set of $\tilde{\alpha}$. We then integrate eq. (2.12) forward from initial conditions (2.13), to calculate a new representer matrix Γ_N^* . (We do not make use of eqs. (2.14)–(2.16) in this calculation.) This matrix is an $I \times I$ matrix involving interactions between the I gridpoints at $n = N$ and the I pseudo-observations (at $n = N$) which occur at the same gridpoints. We use the (*) notation to indicate that this is a different matrix than the $K \times I$ matrix Γ_N . Note that Γ_N^* is only determined by the gridpoint locations (at $n = N$), and the initial and model error covariances.

In Section 2, we noted that Γ was the a priori error covariance between the gridpoints and the observations. When the pseudo-observations are at gridpoints and the observation variables are the analysis variables, it is easy to see that Γ_N^* is actually the square $I \times I$ a priori error covariance between the I gridpoints at time $n = N$. Thus, analogous to the definition of $\mathbf{R}_e$ in Section 2 and noting that $\mathbf{H}$ is the identity matrix in this case, we have,

$$\mathbf{B}_{NN}^b = \Gamma_N^*, \quad (5.4)$$

which is substituted into (5.3) to complete the calculation.

To begin the next cycle we set $\mathbf{x}_0^b$ equal to the state estimate at the end of the previous cycle and $\mathbf{B}_0^b$ equal to the covariance error estimate (5.4) obtained at the end of the previous cycle.

5.2. Computational considerations and possible approximations

It is important to emphasize that eqs. (5.3)–(5.4) are entirely internally consistent with the representer algorithm of Section 2. Thus, they represent a standard of comparison against which other procedures might be measured. However, these calculations are clearly expensive and it is important to consider how they might be simplified and approximated. In this sub-section we will only discuss computational issues and possible approximations with respect to the update of the error covariances discussed in eqs. (5.1)–(5.4). The computational cost and possible approximations with respect to the state estimate itself were discussed briefly at the end of Section 2.

There are 2 parts to the covariance update — the calculation of the a priori error covariance of

eq. (5.4) and the a posteriori error covariance of eq. (5.3). It is instructive to compare these 2 equations with the most widely used alternative method of obtaining the same information — the Kalman filter or approximations thereof. The Kalman filter would obtain an estimate of the a priori error covariance by using eq. (20) of Daley (1997). We re-write this equation using the notation of Sections 2 and 5, for a timestep n , in the absence of observations, as,

$$\mathbf{B}_{n+1,n+1}^b = \mathbf{M}_n \mathbf{B}_{nn}^a \mathbf{M}_n^T + \mathbf{Q}_{nn}. \quad (5.5)$$

This equation would be applied N times to march from $n=0$ to $n=N$. The Kalman filter would obtain the a posteriori error covariance using basically the same equation as (5.3), which as noted earlier, is a form of eq. (17) of Daley (1997). Let us first note that a standard Kalman filter update of the error covariances would be more restrictive than that obtained from eqs. (5.3)–(5.4), because it is not valid when the model or observation error is serially correlated. (A much more complicated version of the Kalman filter can be applied in these cases, Daley, 1992a). However, if the model and forward observation models are linear and the model and observation errors are not serially correlated, the standard Kalman filter update and eqs. (5.3)–(5.4) should produce the same a priori and a posteriori error covariances estimates at the end of the time interval.

Since the generation of the a posteriori error covariance estimate from eq. (5.3) is basically similar in the representer update and Kalman filter update procedures, there is no reason to compare their computational costs. The interesting question is a comparison between the Kalman filter generation of the a priori error covariance estimate of eq. (5.5) and the equivalent representer update of eq. (5.4). The costly term in the Kalman filter update is the first term of eq. (5.5), which is the predictability error growth term. To calculate this first term of eq. (5.5) for N timesteps requires $2NI^3$ operations. By comparison the representer error covariance update of eq. (5.4) requires $2NI^2$ operations. Clearly, the representer update is much cheaper as I (the number of gridpoints) is normally a very large number. If one wishes to obtain the full a priori error covariance matrix (which contains $I \times I$ elements) at the end of the interval it is difficult to see how it could be calculated much more inexpensively.

Of course, both eqs. (5.3)–(5.4), like the 2 corresponding Kalman filter equations can be approximated in some fashion. Thus, with respect to the a posteriori error covariance eq. (5.3), Fisher and Courtier (1995) have proposed using low rank approximations to the second term which exploit certain properties of standard descent algorithms. As for the a priori error covariance update of eq. (5.5), various approximations have been proposed. One idea is to use very low resolution simplified forecast models for the covariance evolution. Another idea (see Ehrendorfer and Bouttier, 1998) is to derive a low order Kalman filter based on an expansion in the leading (most unstable) singular vectors of the forecast model. Another procedure is to approximate the evolved error covariance from an ensemble of model runs (Evensen and Van Leeuwen, 1996) — the ensemble Kalman filter. There is no reason why such ideas might not be also applied to the inherently more efficient update of eq. (5.4).

5.3. Experiments 5 and 6 — cycling with perfect observations

Experiments 5 and 6 (Table 1) are the same as experiments 3 and 4, except extended to 5 cycles, using the above theory. The observations are perfect and the model error covariance is the same as in Section 4. For each network, the observation locations are the same in every cycle.

Experiment 5 used Network A and is plotted in Figs. 3a–e. Panels (a–c) correspond to the 1st cycle and panels (d and e) correspond to the end of the 3rd cycle and the end of the 5th cycle respectively. The initial error covariance ($t=0$) is $c(x, 0) = 0$, as in experiment 3. The very poor state estimate at the end of the 1st cycle (dash-dot line in panel (c)) is used as the initial condition for the 2nd cycle, and the corresponding posteriori error covariance (eq. (5.4)) is used as the initial error covariance for the 2nd cycle. The same type of operation is conducted at the end of all the subsequent cycles. It can be seen that the very poor result at the end of the 1st cycle (due to the poor specification of the initial state and initial error covariance) has been largely rectified by the end of the 3rd cycle, and even more so by the 5th. The state estimates essentially asymptote by the end of the 5th cycle.

In Fig. 4, we show the evolution of the initial

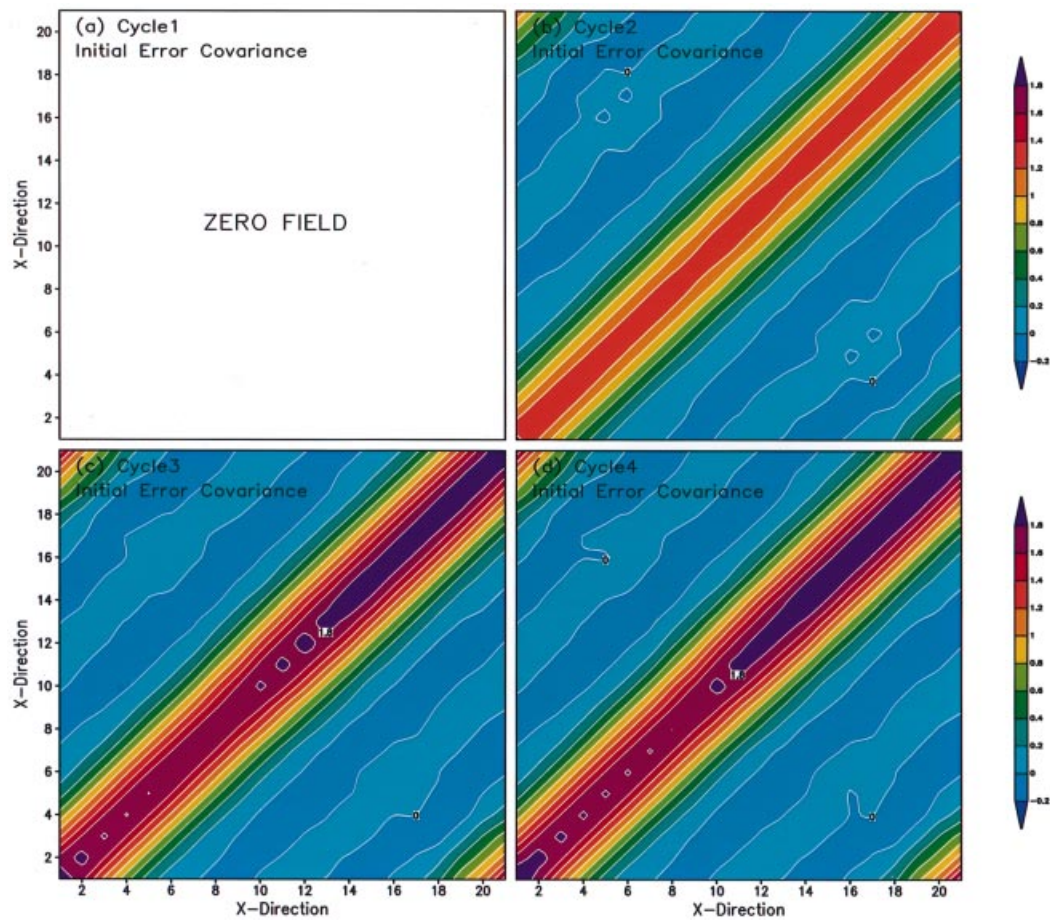


Fig. 4. The error covariance at the beginning of cycle 1 (panel a), cycle 2 (panel b), cycle 3 (panel c) and cycle 4 (panel d) as a function of x and x' . The amplitudes (multiplied by 100) are indicated by the color bars to the left.

error covariance for the 1st 4 cycles for experiment 5. Because this problem has only one spatial dimension, it is straightforward to plot any error covariance in model space as 2-dimensional (x, x') plots. Panel (a) corresponds to the beginning of cycle 1, where the initial error covariance is assumed to be zero. Panels (b–d) correspond to the beginning of cycles 2, 3, and 4, respectively, and were calculated using eq. (5.4). The positive and negative covariances are indicated by the color bars on the left. The matrices are diagonally dominant, indicating that the initial error at the beginning of each cycle is largely spatially uncorrelated. The large positive values in the upper left

and lower right corners are due to the periodic boundary conditions. These initial error covariances have essentially asymptoted by the 5th cycle, which is not surprising considering the observation locations are the same in every cycle. It might be noted that the fact that the initial error covariances asymptoted toward a diagonally dominant form, explains why experiment 3A (in which the initial error covariance was specified to be precisely diagonal), worked so well.

Experiment 6 was similar to experiment 5, except performed using Network B. The results of this experiment are plotted in Figs. 3a'–e'. The improvement in the state estimation with

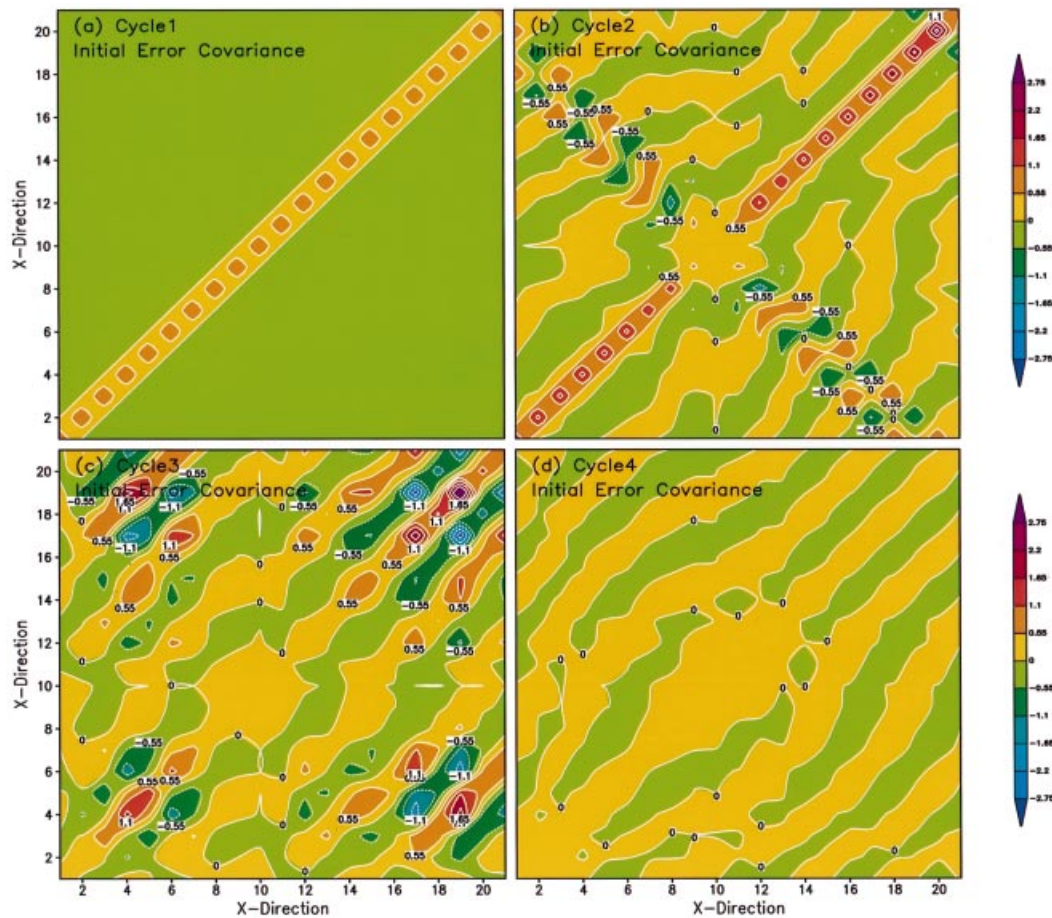


Fig. 5. Same as Fig. 4, except for Network B and no multiplication of the amplitudes.

additional cycles is quite apparent when comparing the analysis (dash-dot lines) in panels d' and e' with panel c'. The results are not quite as good as with Network A, but nonetheless, very encouraging. In Fig. 5 is shown the initial error covariance (in the same format as Fig. 4), for the 1st 4 cycles of experiment 6. The imposed diagonal structure at $t = 0$ (Section 3), is evident in panel (a). The evolving covariance structure is much more complicated than in Fig. 4, due to the single station nature of Network B. The error covariances shown in this figure are indeed, very different from the standard modeled covariances specified in 3-dimensional variational assimilation. The covariance seemed to asymptote by the end of the 4th cycle.

An important question is the relative importance at the beginning of each cycle of an accurate state estimate and an accurate error covariance estimate. Several experiments (not shown) were performed with both networks in which the initial error covariances specified at the beginning of the 1st cycle, was also specified at the beginning of each subsequent cycle. In other words, we ignored the covariance update of the eqs. (5.3)–(5.4). The results showed that if the initial error covariance was reasonably specified then the analyzed solution converged to the true solution at the 5th cycle. On the other hand, if the initial error covariance was poorly specified, then even the ingestion of perfect observations for many cycles, did not converge to the true solution. This

illustrated the value of an evolving covariance estimate such as the eqs. (5.3)–(5.4).

5.4. Experiments 7 and 8 — cycling with imperfect observations

All experiments so far, were conducted with perfect observations, which could lead to misleading conclusions. Consequently, we repeated experiments 5 and 6, with imperfect observations. The observation locations were the same, but random, spatial and serially uncorrelated observation error was added to each observation and the R matrix was altered accordingly. Note that the observation error for the same observation location was different in every cycle.

In Fig. 6 are shown the truth and (imperfect) observations for the 1st cycle. Panel (a) is for Network A and show the observations in the spatial domain at timestep 22. Panel (b) is for Network B and shows the observations at the 10th gridpoint during cycle.

Experiment 7 used Network A and the results are shown in Figs. 7a–e. Comparison with Figs. 3a–e (perfect observations), demonstrates the ability of the algorithm (after 5 cycles) to effectively filter out the observational noise.

Experiment 8 used Network B imperfect observations and can be compared with experiment 6 (Figs. 3a'–e'). This is the most severe test — imperfect single station data with both poor initial state estimate and poor initial error covariance. On the whole, the results were very satisfactory, with the system beginning to asymptote to the true solution after 5 cycles.

6. Summary and conclusions

The cycling representer algorithm is an extension of the representer algorithm introduced into the oceanographic literature by Andrew Bennett and co-workers (Bennett, 1992; Bennett et al., 1996; Bennett et al., 1997). The representer algorithm is a procedure for minimizing a 4-dimensional cost function in which the forecast model is imposed as a weak constraint. The relevant theoretical development of the representer algorithm was discussed in Sections 2 and 8. However, the representer algorithm, per se, involves minimization over a single time period and is therefore unsuitable for numerical weather prediction, which requires continuous cycling in time. Fortunately, as shown in Section 5, the representer theory can be used to construct (in an internally consistent way) the initial error covariances which are required to start each cycle of the assimilation process.

It should be noted, that the cycling representer algorithm, like any algorithm which attempts to minimize a weak constraint 4-dimensional cost function is very expensive computationally and only demonstrations with simplified atmospheric models are feasible with present day computational resources. For practical applications, approximations would have to be sought as discussed in Sections 2 and 5.

In this paper, the cycling representer algorithm was examined in a very simple context, the 1-dimensional constituent transport problem. The true state for these experiments was always available for comparison if needed. The model was a

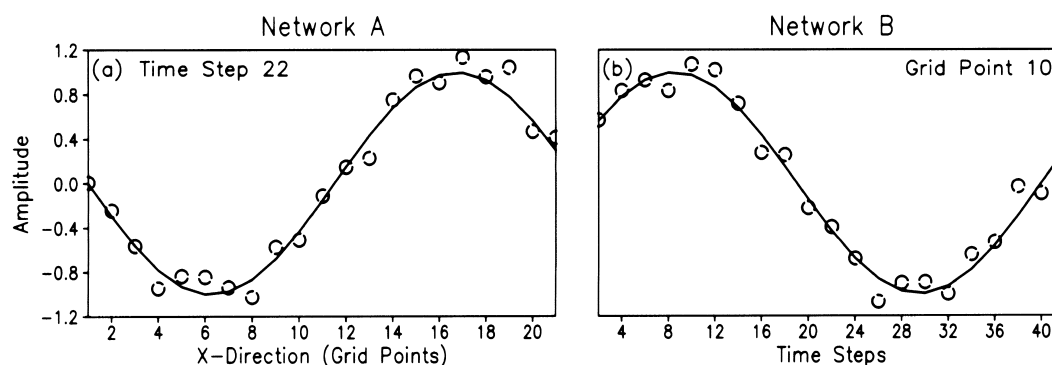


Fig. 6. True state (solid line) and (imperfect) observations (open circles) for Network A (at timestep 22) and Network B (at gridpoint 10).

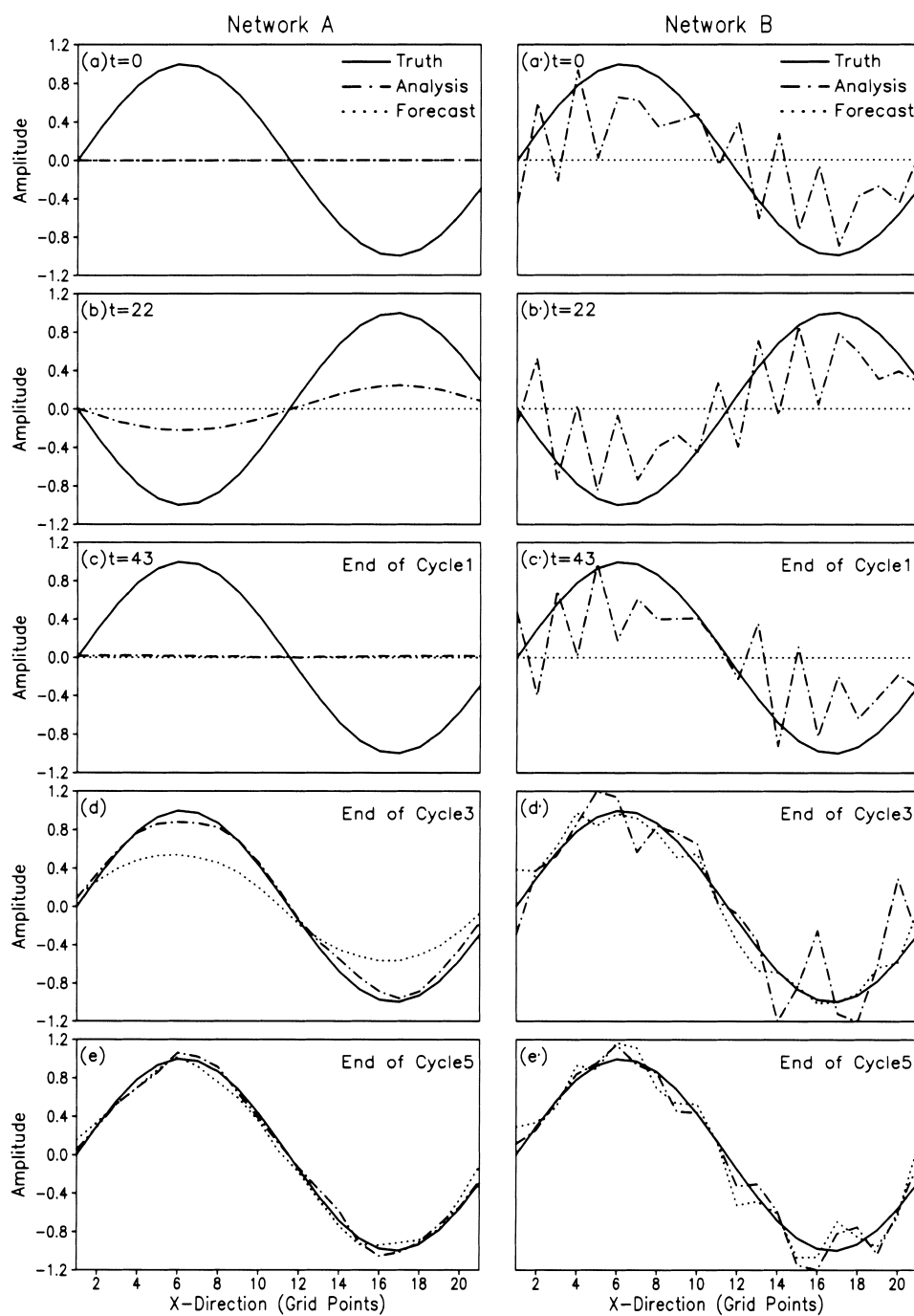


Fig. 7. Same as Fig. 3, except for imperfect observations.

centered difference (in time and space) approximation to the continuous transport equation and was therefore imperfect. The representer equations for this model were derived in Sections 2, 5 and 8. In the experiments, 2 observation networks (both observable) were used with a choice of perfect or imperfect observations. One network (A) had complete spatial coverage, once per cycle and the other network (B) had an observation at only a single gridpoint, but at every other timestep. The cycling representer algorithm requires an initial state estimate (at the beginning of the 1st cycle) and, of course, the observations. In addition, it requires 3 covariances — the initial error covariance (at the beginning of the 1st cycle), the model error covariance and the observation error covariance. For every cycle after the 1st, the cycling representer algorithm generates an initial state and an initial error covariance for the beginning of the next cycle.

Experiments with this system focused on the rôle of the initial state estimate and the initial error covariance, both at the beginning of the 1st cycle and all subsequent cycles. (The observation error and model error covariances were estimated reasonably accurately, and were NOT the subject of this investigation.) In all experiments, the initial state estimate was very inaccurate. The first experiments were performed with one cycle only and considered cases where the initial error covariance was both well-specified and badly mis-specified. It was found for all networks, that in a single cycle the true state could be recovered over the whole cycle if the initial covariance was well-specified; otherwise the true solution could not be recovered.

However, numerical weather prediction does not depend on a single cycle, one hopes the solution will improve as the algorithm is cycled. This turned out to be the case when the system was cycled for 5 cycles. In fact, for the minimal network (B), with imperfect observations, a (modestly) imperfect model, and poor initial state and initial error covariance estimates (at the beginning of the 1st cycle), the cycling representer algorithm was able to essentially recover the true solution by the end of the 5th cycle. Examination of the error covariances at the beginning of each cycle demonstrated how the algorithm generated more appropriate initial error covariances in the late cycles. It was very clear from these experiments, that a cycling data assimilation algorithm requires

good estimates of the error covariance at the beginning of each cycle in order to produce accurate state estimates.

The model used in this paper was very simple and it is not clear whether the present experimental results would hold up under more realistic conditions. The next step will be the introduction of a more realistic model. This will be the linearized shallow water (single level) version of the NRL (Naval Research Laboratory) COAMPS (Coupled Ocean/Atmosphere Mesoscale Prediction System). This system will have 2 space and one time-dimension and we will also consider the effect of lateral boundaries (COAMPS is normally driven by a global model). There will be 2 aims in this project: (1) to see if the present 1-dimensional results hold up and (2) to attempt to generate some less computer intensive approximations to the cycling representer algorithm. After that we will consider the non-linear shallow water model and more realistic observation operators.

7. Acknowledgements

The authors would like to thank Andrew Bennett and Boon Chua of Oregon State University for their encouragement and assistance during this project. Support of the sponsor, Naval Research Laboratory and the Office of Naval Research through program PE-0602435N is gratefully acknowledged. The computational resources were provided by the Naval Oceanographic Office (NAVOCEAN) of the Department of Defense (DoD) Major Shared Resources Center (MSRC). An early version of this work was presented at the Rossby-100 Symposium, 8–12 June 1998, Stockholm, Sweden.

8. Appendix

In the following we give a brief derivation of the representer coefficient $\vec{\beta}$ used in Section 2. When both the forward and model operators $\mathcal{H}$ and $\mathcal{M}$ are linear, we may replace eqs. (2.4) and (2.6) by:

$$\vec{\lambda}_n \equiv \sum_{n'=1}^N \mathbf{Q}_{nn'}^{-1} [\mathbf{x}_{n'}^a - \mathbf{M}_{n'-1} \mathbf{x}_{n'-1}^a] \quad \text{and} \quad (\text{A.1})$$

$$\vec{\lambda}_n - \mathbf{M}_n^T \vec{\lambda}_{n+1} = \{\mathbf{H}^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{H} \mathbf{x}^a)\}_n,$$

where $1 \leq n \leq N$ and $\vec{\lambda}_{N+1} = 0$.

Define a matrix A consisting of $N \times (N + 1)$ blocks, each of which is $I \times I$:

$$A = \begin{pmatrix} I & -M_1^T & & & 0 \\ & \cdot & & & \\ & I & -M_n^T & & \\ & & \cdot & & \\ 0 & & I & -M_N^T & 0 \end{pmatrix}, \quad (A.2)$$

where I is an $I \times I$ identity matrix. We may then rewrite the second eq. of (A.1) and eq. (2.10) as,

$$A\vec{\lambda} = H^T R^{-1} [y - H(x^a)] \quad \text{and} \quad A\vec{\alpha}_k = H_k^T. \quad (A.3)$$

Now define a matrix F , consisting of $(N + 1) \times (N + 1)$ blocks, each of which is $I \times I$,

$$F = \begin{pmatrix} I & & & & 0 \\ -M_0 & I & & & \\ & \cdot & & & \\ & & -M_{n-1} & I & \\ & & & \cdot & \\ 0 & & & -M_{N-1} & I \end{pmatrix}. \quad (A.4)$$

Define Q^* as a matrix consisting of $(N + 1) \times (N + 1)$ blocks, each of which is $I \times I$,

$$Q^* = \begin{pmatrix} B_0^b M_0^T & 0 \\ Q & 0 \end{pmatrix}. \quad (A.5)$$

We may then rewrite eqs. (2.7) and (2.8) and (2.12) and (2.13) as,

$$F x^a = Q^* \vec{\lambda} \quad \text{and} \quad F \vec{\gamma}_k = Q^* \vec{\alpha}_k. \quad (A.6)$$

Applying F to eq. (2.14) and using the second eq. of (A.6), we have,

$$\begin{aligned} F x^a &= F x^b + \sum_{k=1}^K \beta_k F \vec{\gamma}_k = \sum_{k=1}^K \beta_k Q^* \vec{\alpha}_k \\ &= Q^* \sum_{k=1}^K [\beta_k \vec{\alpha}_k], \end{aligned} \quad (A.7)$$

remembering that $F x^b = 0$ and β_k is a scalar. Substitution into (A.7) from the first eq. of (A.6) gives,

$$\vec{\lambda} = \sum_{k=1}^K \beta_k \vec{\alpha}_k. \quad (A.8)$$

Applying A to eq. (A.8), and using the second eq. of (A.3), we have:

$$A\vec{\lambda} = \sum_{k=1}^K \beta_k A\vec{\alpha}_k = \sum_{k=1}^K \beta_k H_k^T. \quad (A.9)$$

Comparison of the first eq. of (A.3) with (A.9) gives:

$$\begin{aligned} \sum_{k=1}^K H_k^T \beta_k &= H^T R^{-1} [y - H x^a] \\ &= \sum_{k=1}^K H_k^T [R^{-1} [y - H x^a]]_k, \end{aligned} \quad (A.10)$$

where $[R^{-1} [y - H x^a]]_k$ is a scalar. This means,

$$\beta_k = [R^{-1} [y - H x^a]]_k \quad \text{or} \quad \vec{\beta} = R^{-1} [y - H x^a]. \quad (A.11)$$

Substituting eq. (2.14) into eq. (A.11) gives:

$$\begin{aligned} \vec{\beta} &= R^{-1} [y - H x^b - H \Gamma^T \vec{\beta}] \quad \text{or} \\ \vec{\beta} &= [R_e + R]^{-1} [y - H x^b]. \end{aligned} \quad (A.12)$$

REFERENCES

- Amodei, L. 1995. Solution approchee pour un problem d'assimilation de donnees meteorologiques avec prise en compte de l'erreur de modele. *Comptes Rendus de l'Academie des Sciences* **321**, serie IIa, 1087–1094.
- Bennett, A. F. 1992. *Inverse methods in physical oceanography*. Monographs on Mechanics and Applied Mathematics. Cambridge University Press, Cambridge, 346 pp.
- Bennett, A. F. 1997. *Inverse methods and data assimilation*. 1997 Summer School Lecture Notes, Oregon State University, Corvallis, Oregon.
- Bennett, A. F., Chua, B. S. and Leslie, L. M. 1996. Generalized inversion of a global numerical weather prediction model. *Meteorol. Atmos. Phys.* **60**, 165–178.
- Bennett, A. F., Chua, B. S. and Leslie, L. M. 1997. Generalized inversion of a global numerical weather prediction model (II). Analysis and implementation. *Meteorol. Atmos. Phys.* **62**, 129–140.
- Callies, U. and Eppel, D. 1995. On the sensitivity of a least-squared fit of discretized linear hyperbolic equations to data. *Tellus* **47A**, 80–102.
- Cohn, S. and Dee, D. 1988. Observability of discretized partial differential equations. *SIAM J. Numer. Anal.* **25**, 586–617.
- Courtier, P. 1998. Dual formulation of four dimensional variational assimilation. *Q. J. Roy. Meteor. Soc.* **123**, 2449–2461.
- Courtier, P., Andersson, E., Heckley, W., Pailleu, I., Vasiljevic, D., Hamrud, M., Hollingsworth, A., Rabier, F. and Fisher, M. 1998. The ECMWF implementation

- of the three-dimensional variational assimilation (3D-Var). Part I: Formulation. *Q. J. R. Meteor. Soc.* **124**, 1783–1807.
- Courtier, P., Thepaut, J.-N. and Hollingsworth, A. 1994. A strategy for operational implementation of 4D-Var using an incremental approach. *Q. J. Roy. Meteor. Soc.* **120**, 1367–1387.
- Daley, R. 1985. The analysis of synoptic scale divergence by a statistical interpolation procedure. *Mon. Wea. Rev.* **113**, 1066–1079.
- Daley, R. 1991. *Atmospheric data analysis*. Cambridge University Press, Cambridge, 457 pp.
- Daley, R. 1992a. The effect of serially correlated observation and model error on atmospheric data assimilation. *Mon. Wea. Rev.* **120**, 164–177.
- Daley, R. 1992b. Estimating model error covariances for applications to atmospheric data assimilation. *Mon. Wea. Rev.* **120**, 1735–1746.
- Daley, R. 1997. Error propagation and observability for the constituent transport equation in steady, non-divergent, two dimensional flow. *Atmosphere-Ocean* **35**, 25–54. (Andre J. Robert Memorial Volume — NRC Research Press)
- Daley, R. 1997. Atmospheric data assimilation. *J. Meteor. Soc. Japan* **75**, 319–329.
- Derber, J. C. 1989. A variational continuous assimilation technique. *Mon. Wea. Rev.* **117**, 437–2446.
- Ehrendorfer, M. and Bouttier, F. 1998. An explicit low-resolution extended Kalman filter: implementation and preliminary experimentation. *ECMWF Tech. Memo* **259** (available from ECMWF, Reading, UK).
- Evensen, G. and van Leeuwen, P. J. 1996. Assimilation of Geosat altimeter data for the Agulhas current using the ensemble Kalman filter with a quasigeostrophic model. *Mon. Wea. Rev.* **124**, 85–96.
- Fisher, M. and Courtier, P. 1995. Estimating the covariance matrices of analysis and forecast error in variational data assimilation. *ECMWF Tech. Memo* **220** (available from ECMWF, Reading, UK).
- Menard, R. 1994. *Kalman filtering of Burgers' equation and its application to atmospheric data assimilation*. PhD Thesis. Department of Atmospheric and Ocean Sciences, McGill University, 805 Sherbrooke Street West, Montreal, Quebec H3A 2K6, Canada.
- Mitchell, H. and Daley, R. 1997. Discretization error and signal/error correlation in atmospheric data assimilation: (1) All scales resolved. *Tellus* **49A**, 32–58.
- Navon, I. M., Zou, X., Derber, J. C. and Sela, J. 1992. Variational data assimilation with an adiabatic version of the NMC spectral model. *Mon. Wea. Rev.* **120**, 1433–1446.
- Parrish, D. F. and Derber, J. C. 1992. The National Meteorological Center's spectral statistical-interpolation analysis system. *Mon. Wea. Rev.* **120**, 1747–1763.
- Rabier, F., Thepaut, J.-N. and Courtier, P. 1998. Extended assimilation and forecast experiments with a four dimensional variational assimilation system. *Q. J. Roy. Meteor. Soc.* **124**, 1861–1877.
- Zou, X., Kuo, Y.-H. and Guo, Y.-R. 1995. Assimilation of atmospheric radio refractivity using a nonhydrostatic adjoint model. *Mon. Wea. Rev.* **123**, 2229–2249.
- Zupanski, D. 1997. A general weak constraint applicable to operational 4DVAR data assimilation systems. *Mon. Wea. Rev.* **125**, 2274–2292.
- Zupanski, M. 1993. Regional four-dimensional variational data assimilation in a quasi-operational forecasting environment. *Mon. Wea. Rev.* **121**, 2396–2408.