

Development of NAVDAS-AR: non-linear formulation and outer loop tests

By THOMAS ROSMOND and LIANG XU*, *Marine Meteorology Division, Naval Research Laboratory, Monterey, CA 93943, USA*

(Manuscript received 15 October 2004; in final form 31 March 2005)

ABSTRACT

NAVDAS-AR is the 4DVAR extension of the United States Navy's operational 3DVAR data assimilation system, NAVDAS (Naval Research Laboratory Atmospheric Variational Data Assimilation System), where AR stands for accelerated representer. Like NAVDAS, NAVDAS-AR is cast in observation space, and is an observation space formulation of the 4DVAR algorithm, in contrast to the European Centre for Medium-Range Weather Forecasts 4DVAR system, which is an analysis space algorithm. In this paper we show how an inner loop linear solution can be augmented by an outer loop iteration procedure that introduces non-linear effects in the form of a first-order Taylor series expansion. The non-linear problem is then solved iteratively as a sequence of linear problems.

We show results of the outer loop strategy, where the first loop, the 'linear' solution, is modified in the second outer loop by both a Taylor series term in the calculation of the observation innovations, and a linearized form of the background model that is linearized about an analysis 'trajectory' from the solution of the first loop. We also show results of how the outer loop strategy allows better assimilation of observations of surface wind speed and precipitable water, compared to the operational NAVDAS.

1. Introduction

Atmospheric variational data assimilation is the object of considerable research and development effort in both the operational numerical weather prediction (NWP) community and meteorological research organizations around the world. In variational assimilation, generalized cost functions allow optimal combination of observations, prior model forecast information, and the errors associated with each. From these cost functions Euler–Lagrange equations can be derived and solved if appropriate simplifications and linearizations are introduced. A major advantage of variational assimilation is that generalized cost functions can contain so-called forward models which allow conversion of non-conventional observations (e.g. satellite radiances) into forecast model state variables such as temperature and specific humidity. In fact, much of the current research effort in variational assimilation is focused on these forward models.

Variational data assimilation is normally formulated as a three-dimensional problem (3DVAR) or a four-dimensional problem (4DVAR). In 3DVAR, the cost function has no explicit representation of the time dimension, so differences in time of observation among the observations must be treated in an ad hoc way

independent of the cost function. In 4DVAR, the time dimension is explicitly contained in the cost function through a linearized form of an NWP forecast model and its adjoint, each of which must be integrated repeatedly as part of the solution process. This gives a dynamically consistent time evolution of analysis increments, but at a significantly greater computational cost than a 3DVAR scheme which does not have any model integration requirements. The extra cost can be as much as a factor of 10, so only recently have computer resources become adequate to make 4DVAR operationally feasible at major NWP centers around the world.

Besides the separation of variational data assimilation into 3DVAR and 4DVAR categories, the formulation can be done in (a) information or analysis space form or (b) error or observation space form (see Daley and Barker, 2001a for discussion and details). Examples of analysis space variants of 3DVAR have been developed and run operationally in the United States at the National Centers for Environmental Prediction (NCEP; Derber et al., 1996) and the European Centre for Medium-Range Weather Forecasts (ECMWF; Courtier et al., 1998), among others. An observation space 3DVAR scheme, the Physical Space Assimilation System (PSAS) has been developed at the Data Assimilation Office (DAO) of NASA/Goddard (see Cohn et al., 1998), and another observation space 3DVAR system, the Naval Research Laboratory (NRL) Atmospheric Variational Data

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

Assimilation System (NAVDAS; see Daley and Barker, 2001a,b), has been developed by the Marine Meteorology Division of the NRL in Monterey, California. It is run operationally by the United States Navy for their global NWP requirements. NAVDAS is the predecessor to the four-dimensional observation space system to be described in this paper.

An analysis space 4DVAR atmospheric global data assimilation system has been developed jointly by ECMWF and Meteo France as an evolution of the ECMWF 3DVAR system. It is described by Courtier et al. (1994) and has been run operationally at these institutions since the late 1990s. It uses a perfect model assumption that allows a strong constraint variational formulation. A similar global 4DVAR system recently became operational at the United Kingdom (UK) Met Office. In the United States, an observation space 4DVAR system, NAVDAS-AR, where AR stands for accelerated representer, is under development at NRL. Courtier (1997) has shown that when the perfect model assumption is made, the observation space 4DVAR algorithm is the dual of the analysis space 4DVAR algorithm, and from a linear algebra perspective, the two methods are equivalent.

The direct representer algorithm was introduced by Bennett and McIntosh (1982) and Bennett and Thorburn (1992). They applied it to oceanographic data assimilation problems that minimize a four-dimensional weak variational constraint cost function measuring the fit of an analysis to observations, an *a priori* background forecast, and time and space boundary conditions. In their application the minimization time interval is quite long (e.g. weeks or months), and each observation requires an adjoint model run and a tangent-linear model (TLM) run over this interval. The first meteorological application of the direct representer method was by Xu and Daley (2000), who applied it to short time intervals (e.g. 6 h) as is normal practice in NWP data assimilation. The cycling representer algorithm performs a series of these short interval minimizations as the system steps forward in time. Like the oceanographic application, each observation requires an adjoint and TLM integration. An important benefit of the direct representer algorithm is that it predicts the evolution of background error covariances. Unfortunately, however, the computational cost of essentially two model integration for each observation makes it totally impractical for anything except ‘toy’ problems, and certainly has no operational potential for the foreseeable future.

In light of the impracticality of the direct representer algorithm, Xu and Daley (2002) developed an accelerated form of the cycling representer algorithm. In the accelerated representer algorithm, the error covariances at the beginning of each cycle are specified, as they are in other 3DVAR and 4DVAR algorithms. The cost function is then minimized by solving the Euler–Lagrange equations as a large descent algorithm problem using methods such as conjugate gradient. Each descent iteration requires an adjoint and TLM model integration over the update cycle period, but a converged solution usually takes fewer than 100 iterations. This is in contrast to the $O(10^6)$ in-

tegrations that the direct representer algorithm would take for typical observation volumes encountered in operational global NWP. In fact, there is little difference between the cost function minimization methodologies in analysis space 4DVAR and the accelerated representer algorithm; both solve with some kind of descent algorithm. The only difference is the control variable upon which the minimization is based: in the strong constraint analysis space 4DVAR system the control variable is normally the size of the initial condition grid point or spectral field of an NWP forecast model, while in the observation space accelerated representer it is a variable the size of the observation vector. Courtier (1997) points out that if the perfect model assumption in analysis space 4DVAR is relaxed and a weak constraint cost function is defined, the control variable is defined over the much larger analysis space of the entire time period of the minimization. On the other hand, an imperfect model assumption has no impact on the size of the accelerated representer control variable, because in an observation space algorithm it is always the size of the observation vector. This has significant implications for future applications of the two forms when a model error term is included in the cost function (i.e. an imperfect model). We discuss these implications in our concluding remarks.

Xu and Daley (2000, 2002) introduced the representer algorithm concept and its application to meteorological data assimilation. These two papers describe the first steps toward an operational four-dimensional data assimilation system for the United States Navy using representer algorithms. In a recent paper (Xu et al., 2005, hereafter referred to as XRD), we describe the first application of the accelerated representer algorithm to a fully configured global atmospheric data assimilation system. The system is based on the Navy Global Atmospheric Prediction System (NOGAPS) forecast model (Hogan and Rosmond, 1991) and many of the components of NAVDAS, the Navy’s 3DVAR analysis system. XRD present the formulation of NAVDAS-AR in terms of a generalized non-linear cost function, and describe the simplifications and approximations necessary to reduce the problem to a set of linear Euler–Lagrange equations that are solvable. In this paper we take the next step in this evolution: the solution of the non-linear problem as a series of linearized Euler–Lagrange equations, the outer loop strategy. A similar procedure is performed in the ECMWF 4DVAR system and is described by Courtier et al. (1994), Rabier et al. (1998) and Rabier et al. (2000).

In Section 2, we describe the formulation of NAVDAS-AR as a generalized inverse problem in terms of a non-linear cost function. We summarize the development of the Euler–Lagrange equations described in XRD, and reintroduce the assumptions and approximations necessary to make the equations tractable. In Section 3, we give a brief overview of the fully linear NAVDAS-AR solution described in XRD. In Section 4, we formulate the outer loop strategy necessary to solve the non-linear problem as a series of linearized Euler–Lagrange equations. We present the details of the solution process for these equations, in particular

how the non-linear operators that appear in the general problem are linearized. In Section 5, we present the results of some preliminary data assimilation experiments, showing the impact of the outer loops on analysis increment fields. However, we are not yet in a position to demonstrate the impact of the outer loop strategy on actual forecast skill. Instead we will examine convergence properties of the outer loop strategy and its numerical robustness. In Section 6 we describe our plans for the next step in the evolution of NAVDAS-AR. In Section 7 we give a summary and some conclusions.

2. Formulation of NAVDAS-AR: the Euler–Lagrange equations

In XRD, we defined a generalized inverse problem formulated as a non-linear least-squares cost function, from which we derived the Euler–Lagrange equations that were linearized to yield a tractable problem. We then derived the observation space algorithm upon which NAVDAS-AR is based. These details can also be found in Xu and Rosmond (2004). In this section we will summarize the linear form, and then formulate a more general linearization strategy based on Taylor series expansion of the non-linear operators about an analysis trajectory. This leads to an iterative outer loop procedure similar in philosophy to that described by Courtier et al. (1994), Rabier et al. (1998), Veerse and Thepaut (1998) and Rabier et al. (2000).

2.1. Generalized non-linear cost function

The generalized inverse problem can be posed as the minimization of the following generalized cost function

$$J = J_0^b + J^q + J^r \quad (1)$$

$$J_0^b = \frac{1}{2} [\mathbf{x}_0^b - \mathbf{x}_0]^T [\mathbf{P}_0^b]^{-1} [\mathbf{x}_0^b - \mathbf{x}_0], \quad (2)$$

$$J^q = \frac{1}{2} \sum_{n=1}^N \sum_{n'=1}^N [\mathbf{x}_n - \mathcal{M}(\mathbf{x}_{n-1})]^T \mathbf{Q}_{nn'}^{-1} [\mathbf{x}_{n'} - \mathcal{M}(\mathbf{x}_{n'-1})], \quad (3)$$

$$J^r = \frac{1}{2} \sum_{n=0}^N \sum_{n'=0}^N [\mathbf{y}_n - \mathcal{H}(\mathbf{x}_n)]^T \mathbf{R}_{nn'}^{-1} [\mathbf{y}_{n'} - \mathcal{H}(\mathbf{x}_{n'})], \quad (4)$$

for the time period $t_0 \leq t_n \leq t_N$. J^r and J^q are the scalar cost functions for the observation and model error, respectively, and J_0^b is the scalar cost function for the background error at the beginning of the period ($t = t_0$). Superscripts ‘r’, ‘q’ and ‘b’ refer to the observations, model and background, respectively. Superscript ‘T’ refers to a matrix transpose. Subscripts, ‘0’, ‘n’ and ‘n – 1’ refer to the time at $t = t_0$, $t = t_n$ and $t = t_{n-1}$, respectively. $\mathcal{M}$ is the non-linear forecast model and has a state vector of length I . $\mathbf{x}_n^b$ is the *a priori* background forecast and $\mathbf{x}_n$ is the state vector, each describing atmospheric estimates of length I at time $t = t_n$. $\mathbf{Q}_{nn'}$ is a rank I model error covariance matrix between $t = t_n$ and $t = t_{n'}$. $\mathbf{P}_0^b$ is a rank I background

error covariance matrix at the initial time $t = t_0$. $\mathbf{y}_n$ is a vector of observations of length k_n at time t_n , where $\sum_{n=0}^N k_n = K$. K is therefore the length of the observation vector. $\mathcal{H}$ is non-linear observation operator that produces k_n pseudo-observations from a given state vector $\mathbf{x}_n$ at time t_n . $\mathbf{R}_{nn'}$ is a rank k_n observation error covariance matrix between $t = t_n$ and $t = t_{n'}$.

Taking the first variation of eq. (1) yields

$$\begin{aligned} \delta J = & -[\delta \mathbf{x}_0^a]^T [\mathbf{P}_0^b]^{-1} [\mathbf{x}_0^b - \mathbf{x}_0^a] + \sum_{n=1}^N \sum_{n'=1}^N [\delta \mathbf{x}_n^a - \mathbf{M}_{n-1} \delta \mathbf{x}_{n-1}^a]^T \\ & \times \mathbf{Q}_{nn'}^{-1} [\mathbf{x}_{n'}^a - \mathcal{M}(\mathbf{x}_{n'-1}^a)] \\ & - \sum_{n=0}^N \sum_{n'=0}^N [\mathbf{H}_n \delta \mathbf{x}_n^a]^T \mathbf{R}_{nn'}^{-1} [\mathbf{y}_{n'} - \mathcal{H}(\mathbf{x}_{n'}^a)], \end{aligned} \quad (5)$$

where $\mathbf{x}^a$ is the analysis that is the value of $\mathbf{x}$ which minimizes J . The matrix $\mathbf{M}_{n-1} = \partial \mathcal{M}(\mathbf{x}) / \partial \mathbf{x}$ evaluated at $\mathbf{x}_{n-1}^a$ is an $I \times I$ Jacobian matrix (also commonly known as the TLM in atmospheric sciences) corresponding to the (possibly) non-linear model $\mathcal{M}$. When $\mathcal{M}$ is linear $\mathcal{M}(\mathbf{x}_{n-1}^a) = \mathbf{M}_{n-1} \mathbf{x}_{n-1}^a$. The matrix $\mathbf{H}_n = \partial \mathcal{H}(\mathbf{x}) / \partial \mathbf{x}$ is evaluated at $\mathbf{x}_n^a$ and is an $I \times k_n$ Jacobian matrix corresponding to the (possibly) non-linear forward operator $\mathcal{H}$. When $\mathcal{H}$ is linear we have $\mathcal{H}(\mathbf{x}_n^a) = \mathbf{H}_n \mathbf{x}_n^a$.

In XRD we showed how eq. (5) leads to the following coupled non-linear Euler–Lagrange equations:

$$\lambda_n^a - \mathbf{M}_n^T \lambda_{n+1}^a = \mathbf{H}_n^T \sum_{n'=0}^N \mathbf{R}_{nn'}^{-1} [\mathbf{y}_{n'} - \mathcal{H}(\mathbf{x}_{n'}^a)],$$

for $1 \leq n \leq N - 1$, and subject to

$$\lambda_N^a = \mathbf{H}_N^T \sum_{n'=0}^N \mathbf{R}_{Nn'}^{-1} [\mathbf{y}_{n'} - \mathcal{H}(\mathbf{x}_{n'}^a)] \quad (6)$$

and

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

for $1 \leq n \leq N$, and subject to

$$[\mathbf{x}_0^a - \mathbf{x}_0^b] = \mathbf{P}_0^b \left\{ \mathbf{M}_0^T \lambda_1^a + \mathbf{H}_0^T \sum_{n'=0}^N \mathbf{R}_{0n'}^{-1} [\mathbf{y}_{n'} - \mathcal{H}(\mathbf{x}_{n'}^a)] \right\}. \quad (7)$$

Equation (6) is an equation in the adjoint variable λ_n^a that integrates backward in time, from $n = N$ to $n = 1$, assimilating observations as a right-hand side forcing term (i.e. the innovation vector). Equation (7) integrates the analysis variable $\mathbf{x}_n^a$ forward in time, with an initial condition coming from the convolution of the background error covariance $\mathbf{P}_0^b$ with initial time solution λ_1^a of eq. (6), plus an $n = 0$ innovation vector term that can be discarded, as explained below. For the imperfect model case, a model error term also appears on the right-hand side of eq. (7) as a time-dependent forcing term formed by the convolution of the model error covariance matrix with adjoint variable λ_n^a .

At this point we have made essentially no simplifying assumptions in the derivation of eqs. (6) and (7). They form a coupled

non-linear Euler–Lagrange system that is very difficult to solve. When the forecast model $\mathcal{M}$ and the observation operator $\mathcal{H}$ are linear, however, various representer methods can be used to decouple and solve the linear Euler–Lagrange system. (see Xu and Daley, 2000, 2002; Chua and Bennett, 2001; Bennett, 2002). Beside the linearization that is necessary to achieve a tractable solution, it is also convenient to make some additional assumptions. If we assume model errors and observation errors are not correlated in time, we have $\mathbf{Q}_{nn'} = \mathbf{0}$ and $\mathbf{R}_{nn'} = \mathbf{0}$ for $n \neq n'$ and $\mathbf{Q}_n = \mathbf{Q}_{nn'}$ and $\mathbf{R}_n = \mathbf{R}_{nn'}$ for $n = n'$. If we also assume that observation errors are uncorrelated in space, then $\mathbf{R}_n$ is a diagonal matrix of observation error variance at each $t = t_n$. $\mathbf{Q}_n$ is a model error covariance matrix at each $t = t_n$. A perfect model assumption gives $\mathbf{Q}_n = \mathbf{0}$, and because of our lack of understanding of the form of $\mathbf{Q}_n$, in the subsequent development we will make this assumption. However, we will briefly discuss the imperfect model case in the conclusions.

For cycling operational NWP applications, the observations at beginning of a given data assimilation period, $t = t_0$, are usually already used in the previous data assimilation cycle, i.e. at $t = t_N$ of the previous cycle. Because observations should not be assimilated twice, we must exclude observations in the $t = t_0$ time bin, which requires that the innovation vector term in the initial condition ($n = 0$) of eq. (7) be dropped.

In the next section we summarize the case of linear $\mathcal{M}$ and $\mathcal{H}$ described in XRD, and then in the following section we show how a more general linearization leads to a sequence of linear problems that solves the non-linear problem with an outer loop strategy.

3. Solution of the fully linear case

The approximations and simplifications introduced at the end of the previous section, plus the assumption that $\mathcal{M}(\mathbf{x}_n) = \mathbf{M}_n \mathbf{x}_n$ and $\mathcal{H}(\mathbf{x}_n) = \mathbf{H}_n \mathbf{x}_n$ yield the following linear Euler–Lagrange equations:

$$\begin{aligned} \lambda_n^a - \mathbf{M}_n^T \lambda_{n+1}^a &= \mathbf{H}_n^T \mathbf{R}_n^{-1} [\mathbf{y}_n - \mathbf{H}(\mathbf{x}_n^a)], \\ \text{for } 1 \leq n \leq N-1, \\ \text{subject to } \lambda_N^a &= \mathbf{H}_N^T \mathbf{R}_N^{-1} [\mathbf{y}_N - \mathbf{H}(\mathbf{x}_N^a)] \end{aligned} \quad (8)$$

and

$$\begin{aligned} \mathbf{x}_n^a - \mathbf{M}_{n-1} \mathbf{x}_{n-1}^a &= 0, \text{ for } 1 \leq n \leq N, \\ \text{subject to } [\mathbf{x}_0^a - \mathbf{x}_0^b] &= \mathbf{P}_0^b \mathbf{M}_0^T \lambda_1^a. \end{aligned} \quad (9)$$

The prior background state estimate $\mathbf{x}^b$ is from a non-linear model forecast:

$$\begin{aligned} \mathbf{x}_n^b &= \mathcal{M}(\mathbf{x}_{n-1}^b) \text{ for } 1 \leq n \leq N, \\ \text{subject to the initial condition } \mathbf{x}_0^b. \end{aligned} \quad (10)$$

In XRD the representer method is used to decouple eqs. (8)–(9) and yield the observation space NAVDAS-AR algorithm:

$$\mathbf{x}^a = \mathbf{x}^b + \mathbf{P}^b \mathbf{H}^T [\mathbf{H} \mathbf{P}^b \mathbf{H}^T + \mathbf{R}]^{-1} [\mathbf{y} - \mathbf{H} \mathbf{x}^b] \quad (11)$$

and

$$\mathbf{P}^b = \mathbf{M} \mathbf{M}_0 \mathbf{P}_0^b \mathbf{M}_0^T \mathbf{M}^T. \quad (12)$$

Here, $\mathbf{P}_0^b$ is the background error covariance at initial time ($n = 0$). NAVDAS-AR uses the same $\mathbf{P}_0^b$ as is used in NAVDAS (see Daley and Barker, 2001a). The details of the derivation of eq. (11) can be found in XRD and Xu and Rosmond (2004). The solution procedure is also described in great detail in XRD and will not be repeated here.

4. Solution of the non-linear case: the outer loop strategy

The coupled Euler–Lagrange eqs. (6)–(7) are complicated by the non-linearity of the forecast model $\mathcal{M}$ and the observation operator $\mathcal{H}$. However, it is possible to recast the equations as a sequence of coupled problems, each iteration step linearized about an analyzed basic state (trajectory) from the previous iteration. The non-linearity is therefore approximated by Taylor series expansions. If the non-linearity of the system is not too strong, we expect the procedure to converge to a good approximation of the non-linear solution. In this section we develop this iterative system in detail.

First, let $\mathbf{x}^j$ and $\mathbf{x}^{j-1}$ be the estimated state for the j th and $(j-1)$ th iterations, respectively. Let $\mathbf{x}_n^j = \mathbf{x}_n^{j-1} + \delta \mathbf{x}_n^j$. For simplicity we have dropped the ‘a’ superscripts on the x and λ variables. The non-linear model $\mathcal{M}$ and the non-linear observation operator $\mathcal{H}$ at the j th iteration are linearized around the previous analysis ($j-1$) using Taylor expansion, such that

$$\begin{aligned} \lambda_n^j - [\mathbf{M}_n^{j-1}]^T \lambda_{n+1}^j &= [\mathbf{H}_n^{j-1}]^T \sum_{n'=0}^N \mathbf{R}_{nn'}^{-1} \\ &\times [\mathbf{y}_{n'} - \mathcal{H}(\mathbf{x}_{n'}^{j-1}) - \mathbf{H}_{n'}^{j-1} \delta \mathbf{x}_{n'}^j], \\ \text{for } 1 \leq n \leq N-1, \text{ and subject to} \end{aligned} \quad (13)$$

$$\begin{aligned} \lambda_N^j &= [\mathbf{H}_N^{j-1}]^T \sum_{n'=0}^N \mathbf{R}_{Nn'}^{-1} \\ &\times [\mathbf{y}_{n'} - \mathcal{H}(\mathbf{x}_{n'}^{j-1}) - \mathbf{H}_{n'}^{j-1} \delta \mathbf{x}_{n'}^j] \end{aligned}$$

and

$$\mathbf{x}_n^j - \mathcal{M}(\mathbf{x}_{n-1}^{j-1}) - \mathbf{M}_{n-1}^{j-1} \delta \mathbf{x}_{n-1}^j = 0,$$

for $1 \leq n \leq N$, and subject to

$$\begin{aligned} [\mathbf{x}_0^j - \mathbf{x}_0^b] &= \mathbf{P}_0^b \{ [\mathbf{M}_0^{j-1}]^T \lambda_1^j + [\mathbf{H}_0^{j-1}]^T \\ &\times \sum_{n'=0}^N \mathbf{R}_{0n'}^{-1} [\mathbf{y}_{n'} - \mathcal{H}(\mathbf{x}_{n'}^{j-1}) - \mathbf{H}_{n'}^{j-1} \delta \mathbf{x}_{n'}^j] \}. \end{aligned} \quad (14)$$

Note the similarity of eqs. (13) and (14) to eqs. (6) and (7). The matrix $\mathbf{M}_{n-1}^{j-1} = \partial \mathcal{M}(\mathbf{x}) / \partial \mathbf{x}$, evaluated at $\mathbf{x}_{n-1}^{j-1}$ is an $I \times I$

Jacobian matrix or TLM corresponding to the non-linear model $\mathcal{M}$. The matrix $\mathbf{H}_n^{j-1} = \partial H(\mathbf{x})/\partial \mathbf{x}$, evaluated at $\mathbf{x}_n^{j-1}$, is an $I \times k_n$ Jacobian matrix corresponding to the non-linear observation operator $\mathcal{H}$.

Equations (13) and (14) form a coupled linearized Euler–Lagrange system. The system can now be decoupled using similar procedures to those used in XRD.

4.1. Solution of the coupled linearized Euler–Lagrange system

Although there are many similarities between the coupled linear system defined by eqs. (13) and (14) and that defined by eqs. (8) and (9), there are differences in the details that require careful examination. With similar assumptions to those used in Section 3, we simplify eqs. (13) and (14) to yield

$$\begin{aligned} \lambda_n^j - [\mathbf{M}_n^{j-1}]^T \lambda_{n+1}^j &= [\mathbf{H}_n^{j-1}]^T \mathbf{R}_n^{-1} [\mathbf{y}_n - \mathcal{H}(\mathbf{x}_n^{j-1}) - \mathbf{H}_n^{j-1} \delta \mathbf{x}_n^j], \\ \text{for } 1 \leq n \leq N-1, \text{ and subject to} \\ \lambda_N^j &= [\mathbf{H}_N^{j-1}]^T \mathbf{R}_N^{-1} \\ &\times [\mathbf{y}_N - \mathcal{H}(\mathbf{x}_N^{j-1}) - \mathbf{H}_N^{j-1} \delta \mathbf{x}_N^j] \end{aligned} \quad (15)$$

and

$$\begin{aligned} \mathbf{x}_n^j - \mathcal{M}(\mathbf{x}_{n-1}^{j-1}) - \mathbf{M}_{n-1}^{j-1}(\mathbf{x}_{n-1}^j - \mathbf{x}_{n-1}^{j-1}) &= 0, \\ \text{for } 1 \leq n \leq N, \text{ and subject to} \\ \mathbf{x}_0^j &= \mathbf{x}_0^b + \mathbf{P}_0^b [\mathbf{M}_0^{j-1}]^T \lambda_1^j. \end{aligned} \quad (16)$$

Let us introduce a prior linear state $(\mathbf{x}^p)^j$ for the j th iteration, which satisfies the following governing equation linearized around the previous analysis $\mathbf{x}^{j-1}$:

$$\begin{aligned} (\mathbf{x}^p)_n^j - \mathcal{M}(\mathbf{x}_{n-1}^{j-1}) - \mathbf{M}_{n-1}^{j-1}[(\mathbf{x}^p)_{n-1}^j - \mathbf{x}_{n-1}^{j-1}] &= 0, \\ \text{for } 1 \leq n \leq N, \text{ and subject to} \\ (\mathbf{x}^p)_0^j &= \mathbf{x}_0^b. \end{aligned} \quad (17)$$

We also introduce the definitions

$$\delta \mathbf{x}_n^j = \mathbf{x}_n^j - \mathbf{x}_n^{j-1} \quad (18)$$

and

$$(\delta \mathbf{x}^p)_n^j = (\mathbf{x}^p)_n^j - \mathbf{x}_n^{j-1}. \quad (19)$$

$(\mathbf{x}^p)_n^j$ is used here as the background state for the minimization of the linear problem (the inner loop) in the representer method. The analysis $\mathbf{x}^j$ for the j th iteration can now be written as

$$\mathbf{x}^j - (\mathbf{x}^p)^j = (\mathbf{P}^b)^{j-1} [\mathbf{H}^{j-1}]^T \beta^j \quad (20)$$

or

$$\delta \mathbf{x}^j - \delta (\mathbf{x}^p)^j = (\mathbf{P}^b)^{j-1} [\mathbf{H}^{j-1}]^T \beta^j, \quad (21)$$

where β^j is the coupling vector or the analysis residual normalized by the observation error covariance for the j th iteration and satisfies the following:

$$\begin{aligned} \beta^j &= [\mathbf{H}^{j-1} (\mathbf{P}^b)^{j-1} [\mathbf{H}^{j-1}]^T + \mathbf{R}]^{-1} \\ &\times [\mathbf{y} - \mathcal{H}(\mathbf{x}^{j-1}) - \mathbf{H}^{j-1} \delta (\mathbf{x}^p)^j]. \end{aligned} \quad (22)$$

Detailed derivation of eqs. (21) and (22) can be found in Appendices A and B, respectively. Also notice that eqs. (21) and (22) are, respectively, the post-multiplier and inner loop solver steps described in XRD.

4.2. Calculation of $(\mathbf{x}^p)^j$, the background trajectory

Equation (17) as written cannot be solved because of the presence of the $\mathcal{M}(\mathbf{x}_{n-1}^{j-1})$ term. However, because our Taylor series linearization is about a base state that is the previous analysis, the obvious choice is to assume that $\mathbf{x}_n^{j-1} = \mathcal{M}(\mathbf{x}_{n-1}^{j-1})$. This assumption, along with eq. (19), yields

$$\begin{aligned} (\delta \mathbf{x}^p)_n^j - \mathbf{M}_{n-1}^{j-1} (\delta \mathbf{x}^p)_{n-1}^j &= 0, \\ \text{for } 1 \leq n \leq N, \text{ and subject to} \\ (\delta \mathbf{x}^p)_0^j &= \mathbf{x}_0^b - \mathbf{x}_0^{j-1}. \end{aligned} \quad (23)$$

Equation (23) is a tangent-linear integration of an initial condition correction to the previous analysis, yielding an adjusted background trajectory that is used to correct the innovation term on the right-hand side of eq. (22). Notice, however, that we still have $(\mathbf{x}^p)_0^j = \mathbf{x}_0^b$, which is consistent with our original formulation.

Although eq. (23) is a natural choice for a linearization scheme to predict the $(\mathbf{x}^p)_0^j$, because it is consistent with the representer derivation, there is no reason to restrict ourselves to this linearization. In fact, Bennett (2002) suggests experimenting with linearization schemes to assess the impact on iteration strategies such as the NAVDAS-AR outer loop, particularly if numerical instabilities are a potential problem in the scheme. We have experimented with an alternative scheme to eq. (23), specifically a Picard iteration in which the non-linear model $\mathcal{M}$ is linearized about the previous analysis in the form of a relaxation term on $(\mathbf{x}^p)_n^j$ to that analysis. A simple one-dimensional example of a Picard iteration is shown in Appendix C. This approach allows us to retain the non-linear model diabatic physics as a forcing term to the linear model, an appealing property. In fact, in the results of the next section, we use this Picard iteration with a linearized form of NOGAPS model. Preliminary tests show better

convergence of the outer loops with the Picard iteration model than with eq. (23), and so far no instabilities are apparent, so we intend to continue to test the system with the Picard iteration model.

4.3. Summary of the non-linear case

Step 1. For the first iteration ($j = 1$) use eq. (10) to find x_n^b . For $j > 1$ find $(\delta x^p)^j$ from the solution of eq. (23). Notice that $(x^p)^j = x^{j-1} + (\delta x^p)^j$. For the first iteration $(\delta x^p)^j = \mathbf{0}$, i.e. $(x^p)^1 = x^0 = x^b$. With this assumption and using eq. (10) the first outer loop iteration is identical to the inner loop procedure described in Section 3.

Step 2. Calculate $[y - \mathcal{H}(x^{j-1}) - \mathbf{H}^{j-1}(\delta x^p)^j]$.

Step 3. Find the analysis residual for the j th iteration, β^j , which is the solution of the solver (22).

Step 4. Find $\delta x^j - (\delta x^p)^j = x^j - (x^p)^j = \delta^* x^j$, which is the solution of the post-multiplier (20).

Step 5. The analysis for the j th iteration is $x^j = (x^p)^j + \delta^* x^j$. If $\delta^* x^j$ satisfies the convergence criteria, stop the outer loop, otherwise go back to step 1.

5. Results

At this point in NAVDAS-AR development we are not in a position to demonstrate the impact of the new system on NOGAPS forecast performance, i.e. we have not yet run the system in a full update cycle mode. Instead, we are concentrating on testing individual system components in isolation to ensure numerical stability and robustness. In XRD, we demonstrated the ability of NAVDAS-AR to produce comparable analysis increments to the operational 3DVAR NAVDAS, but only for the linear case, i.e. no outer loops.

In this paper, we are extending the tests to the non-linear case with the introduction of an outer loop strategy as described in Section 4. There are two principal components to be tested in the outer loop process:

(a) the linearization of the NOGAPS forecast model about an analysis trajectory, yielding more accurate TLM and adjoint models ($\mathbf{M}$ and $\mathbf{M}^T$) in eq. (A7);

(b) the linearization of non-linear observation operators $\mathcal{H}$ in eq. (22), yielding $\mathbf{H}$, which allows adjustments to the innovations during the outer loop iterations.

For the test results in this section, the outer loop spectral resolution is T239L30, with a Gaussian grid of 720×360 . The calculation of observation innovations is carried out on this grid. The inner loop spectral resolution is T79L30, with a Gaussian grid of 240×120 . The analysis increment fields are computed at this resolution. Combining background fields with the increments is all done in spectral space, so complications due to the mismatch in grid resolutions are avoided.

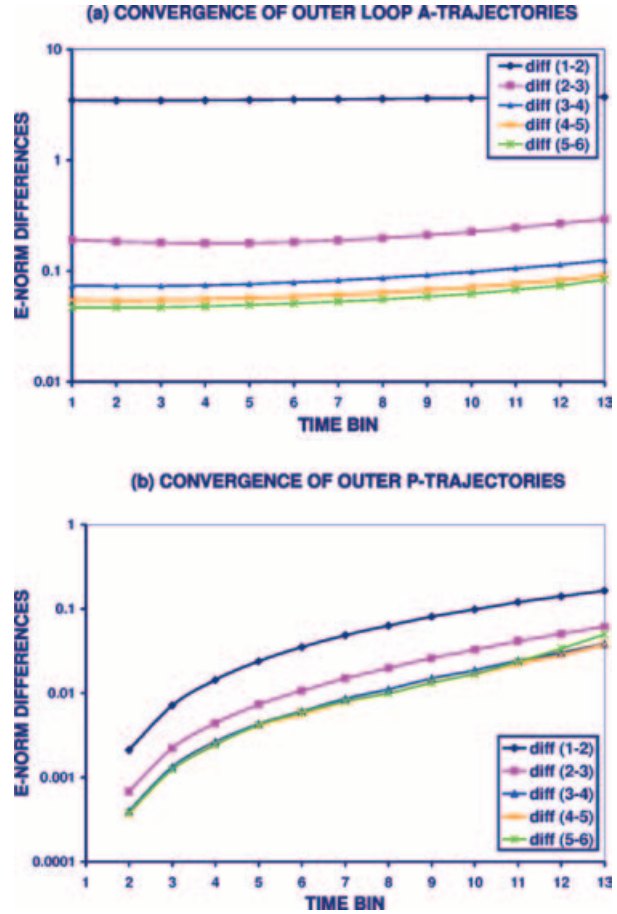


Fig. 1. (a) Outer loop convergence of analysis trajectories. (b) Outer loop convergence of forecast trajectories.

5.1. Inner and outer loop convergence properties

Before showing results for (a) and (b), however, it is worthwhile to examine the numerical behavior of both the outer loop and inner loop iteration processes. An important detail in the outer loop strategy is the distinction between the analysis trajectories x_n^j and the forecast model trajectories $(x^p)^j_n$. x_n^j is constrained by the observations during the 6-h update cycle window, while the $(x^p)^j_n$ is constrained by the fixed initial condition x_0^b and the analysis trajectory from the previous outer loop. Figure 1 shows the convergence behavior of x_n^j and $(x^p)^j_n$, respectively, for six outer loop iterations. Each line is a perturbation energy norm change in $\delta(x)^j$ and $\delta(x^p)^j$ for the analysis trajectories and forecast trajectories, respectively, for $1 \leq j \leq 6$. It is quite apparent that the outer loops converge very rapidly, with over 90% of the convergence taking place after only two loops (note that the ordinate axis is a log scale). The analysis trajectories x_n^j converge quite uniformly over the 6-h window, which is not surprising, because they are directly constrained by the observations over the entire time window. The forecast trajectories $(x^p)^j_n$ behave

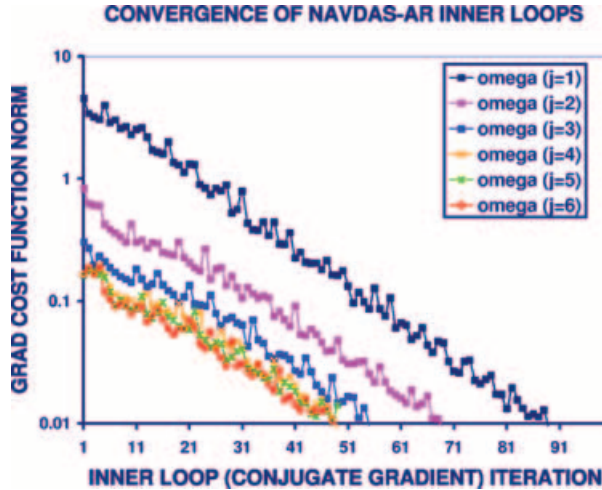


Fig. 2. Inner loop convergence (cost function gradient) as a function of outer loop index.

differently, however. Every forecast trajectory starts from the same $\mathbf{x}_0^b$, so the initial differences are zero, which we cannot plot on the logarithmic ordinate axis. Notice also that the forecast trajectory norm changes are about an order of magnitude less than analysis norm changes. This is clearly a consequence of the fixed $\mathbf{x}_0^b$ for each iteration, which is a stronger constraint than the varying analysis trajectories about which the Picard iteration model is linearized. It also seems likely that the fixed initial condition makes model error more important for the forecast trajectories.

The conjugate gradient solutions of the inner loop as a function of the outer loop index j are shown in Fig. 2. Here we are finding the solution of $\mathbf{r} = \mathbf{d}^j - \nabla \mathbf{A}^j \mathbf{z}$, the solver equation described in XRD. $\mathbf{A}^j$ and $\mathbf{d}^j$ are functions of the analysis trajectories, $\mathbf{z}$ is the analysis residual we are solving for, and $\mathbf{r}$ is the cost function gradient we are iterating toward zero.

For the first outer loop ($j = 1$), we start the conjugate gradient iteration with ($\mathbf{z} = \mathbf{0}$). The large value of the initial gradient norm, ≈ 4 (see Fig. 2), shows that this is not a very good starting point, so the first outer loop, i.e. the linear solution, takes many more iterations to converge¹ than subsequent outer loops. We exploit the fact that because $\mathbf{A}^j$ and $\mathbf{d}^j$ are not changing substantially as the outer loop iterations proceed, the converged solution $\mathbf{z}$ of outer loop ($j - 1$) should be an excellent starting guess for outer loop J . The much smaller starting gradients in the solver equation for subsequent outer loops confirm this. It is important to point out that in these experiments we are reducing our cost function gradient to values much smaller than necessary, because we are interested in testing the robustness of our conjugate gradient algorithm. For $j = 1$ the starting gradient/ending

gradient ratio is about 400. For subsequent iterations the ratio is smaller, but this is primarily because the starting gradient is smaller; the converged outer loop solutions are quite similar after the first two iterations, as shown in Fig. 1. Daley and Barker (2001a) suggest that a ratio of about 100 is adequate for practical applications, and in fact that for very small gradient values the data are being ‘fitted’ with small-scale analysis features that may be undesirable, in addition to the extra computational costs of the additional conjugate gradient iterations. In future NAVDAS-AR applications, we will certainly stop our iterations at more reasonable gradient values.

5.2. Linearization of NOGAPS about the analysis trajectories

As discussed above, we want to look at the impact of linearizing both the forecast model $\mathcal{M}$ and the observation operators $\mathcal{H}$ on outer loop performance. In a first experiment, we limited the data types assimilated into NAVDAS-AR to ‘conventional’ observations, i.e. observations in the actual analysis variables, such as temperature, wind vectors and relative humidity. In this case, $\mathcal{H}$ reduce to simple interpolations, and $\mathcal{H} = \mathbf{H}$. The innovation term on the right-hand side of eq. (22) reduces to $\mathbf{y} - \mathbf{H}(\mathbf{x}^p)^j$, which is only changing as the forecast trajectory $(\mathbf{x}^p)^j$ changes for each outer loop iteration, i.e. there is no dependence on the analysis trajectory $\mathbf{x}^{j-1}$. The linearized model operator $\mathbf{M}^j$ is still a function of $\mathbf{x}^{j-1}$, however, and this is what we want to examine.

The top panel of Fig. 3 shows the 00Z 12 March 2004 NAVDAS-AR increment field for 250-hPa temperature after the first outer loop, i.e. the ($j = 1$) ‘linear’ solution. The bottom panel is the change in this increment field due to the second outer loop. The only significant changes are in the central United States, where the difference field is generally of the same sign as the increment field. The sign of this difference field is such that it should be subtracted from the ($j = 1$) solution to yield the ($j = 2$) solution. The net result of the updated analysis and forecast trajectories in the second outer loop is to slightly reduce the amplitude of the increments. We have seen this pattern frequently, i.e. apparently the linear solution tends to overestimate the increments, and the second outer loop reduce them. Which solution is actually the best will have to wait until we can run forecast experiments from each candidate solution.

An even more dramatic example of the difference between outer loop 1 and outer loop 2 is shown in Fig. 4. The top panel is the loop 1 linear solution, showing relatively large increments of terrain pressure over land masses, particularly Australia and North America. The bottom panel shows almost none of these landlocked features. Figure 5 clearly shows why this happens; the differences almost exactly cancel the loop 1 result. We believe the explanation for this somewhat troubling behavior lies in the way that terrain pressure innovations are derived in NAVDAS-AR. Innovations of terrain pressure are computed at observation

¹We have not yet implemented a preconditioner for the matrix $\mathbf{A}$ in the solver equation, so our conjugate gradient algorithm is relatively slow to converge. This is one of our highest priorities for developing a viable operational NAVDAS-AR.

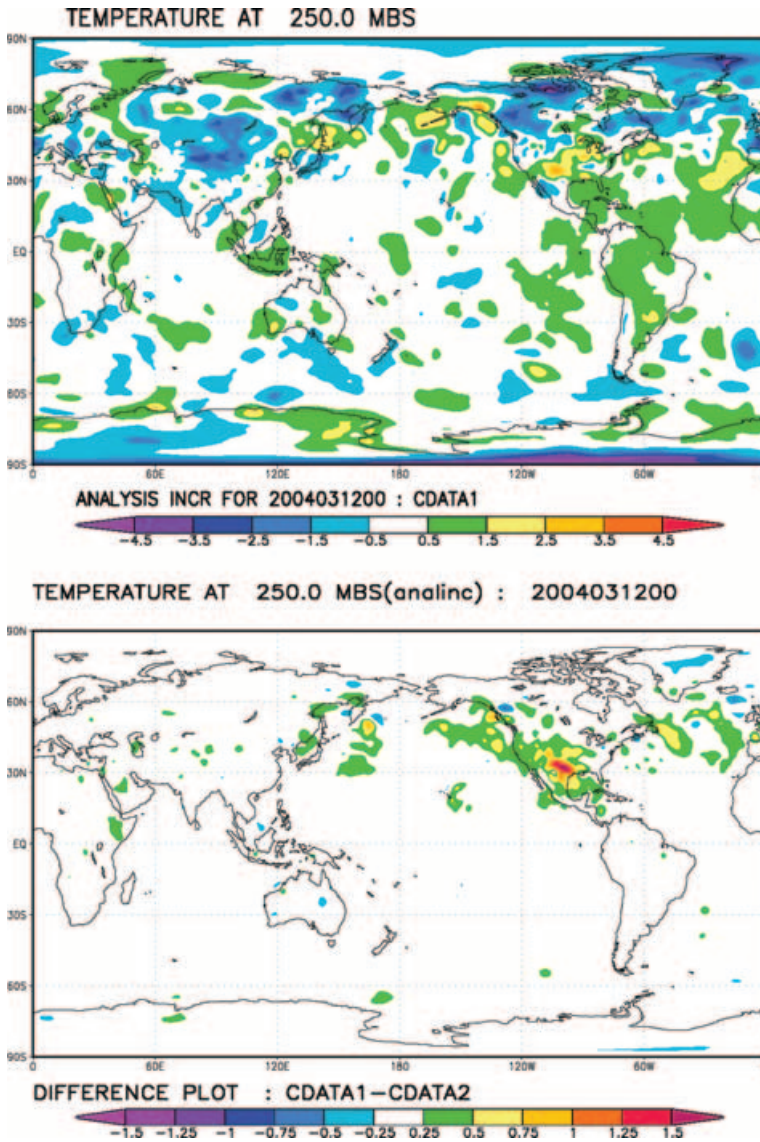


Fig. 3. Top: 250-hPa temperature increments, outer loop 1. Bottom: 250-hPa temperature difference, outer loop 1 – outer loop 2. Units = °C.

locations, but at the terrain pressure of the true terrain height at these locations. The NOGAPS terrain pressure is defined at the terrain heights of the spectrally truncated T239 model terrain, which often departs substantially from the true terrain. We only use terrain pressure innovations at locations where there is a reasonable close match between model and true terrain height (within ≈ 250 m), but this still requires a hydrostatic extrapolation of pressure innovations between the two levels that is subject to error. The second outer loop adjustments are clearly correcting for these errors, because we are sure that the bottom panel result of Fig. 4 is a better solution than the top panel result.

In both Figs. 1 and 2 we see convincing evidence that two outer loops is enough for practical convergence of NAVDAS-AR. Figure 6 is another indication of this convergence condition.

It is equivalent to the bottom panel of Fig. 3, except that it is the difference between outer loop 1 and outer loop 6. The similarities between the two figures are striking, showing that the extra four outer loop iterations added nothing to the NAVDAS-AR solution. These results, and the cost of another outer loop iteration, make it unlikely that we will go beyond two outer loops for any practical NAVDAS-AR applications.

5.3. Linearization of observation operators

At this point in NAVDAS-AR development we only have the ability to assimilate two ‘non-conventional’ observation types: SSM/I (Special Sensor Microwave Imager) wind speeds and SSM/I total column precipitable water. The SSM/I wind speeds require a non-linear observation operator $\mathcal{H}$, so they will have

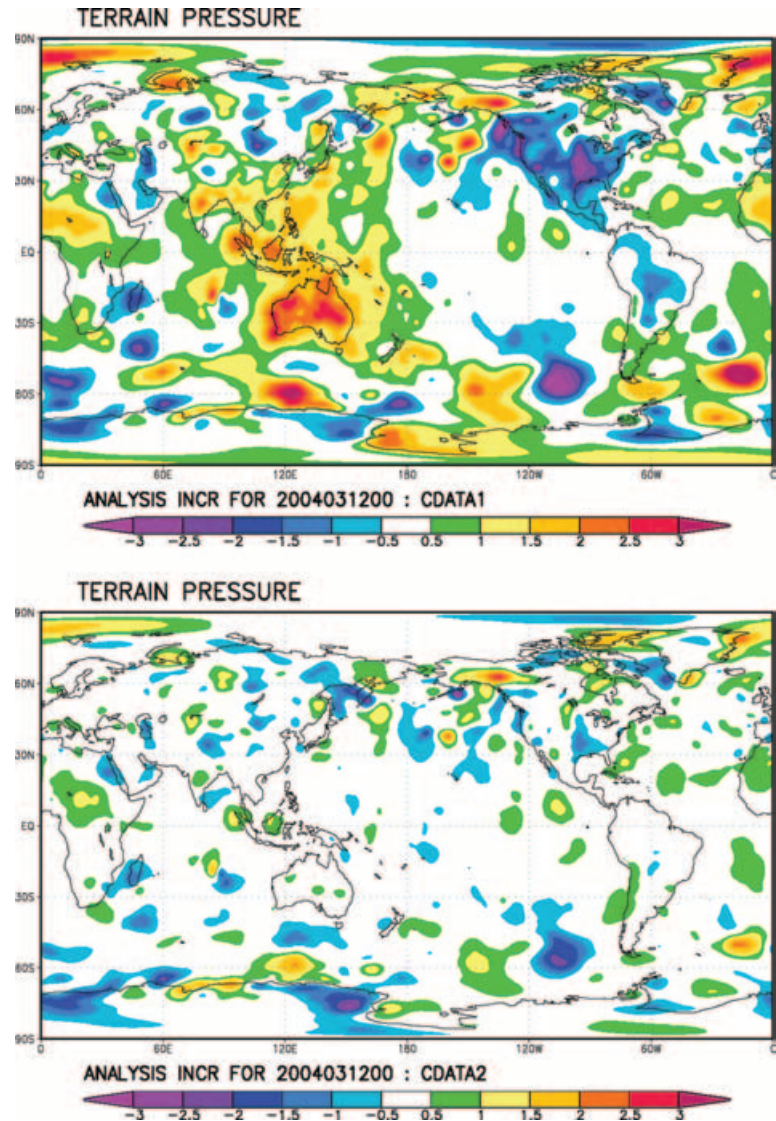


Fig. 4. Top: terrain pres increments, outer loop 1. Bottom: terrain pres increments, outer loop 2. Units = hPa.

an impact on the right-hand side of eq. (22) during outer loop iterations. The observation operator for the SSM/I precipitable water is not non-linear, but it must transform these observations from column integral values, expressed in units of kg m^{-2} to profiles of relative humidity, the native moisture variable in NAVDAS-AR. So the operator does much more than the simple interpolations that **H** performed in our ‘conventional’ data results previously shown. The details of this operator and the non-linear wind speed operator can be found in Daley and Barker (2001a).

The top panel of Fig. 7 shows the impact of assimilating the SSM/I surface winds on 500-hPa vorticity after two outer loops. We are intentionally showing results that are remote from the actual observations to demonstrate the ability of NAVDAS-AR to propagate information away from the surface into the mid-

troposphere under the dynamical control of **M** and **M**^T. 3DVAR systems such as NAVDAS have difficulty accomplishing this (Ed Barker, private communication). The very strong signal in the Southern Hemisphere is a direct reflection of the inability of the operational NAVDAS to effectively assimilate SSM/I wind speeds, because the background fields we are using are still very much influenced by previous update cycles from the operational NAVDAS. When we are cycling NAVDAS-AR on its own, we assume we will see the background fields being ‘conditioned’ by the assimilation of these new data types, and the Southern Hemisphere will be more like the Northern Hemisphere in the impact of these observations.

The bottom panel of Fig. 7 shows the impact of assimilating the SSM/I precipitable water at 1000 hPa. We do not see the same strong Southern Hemisphere signal with these data,

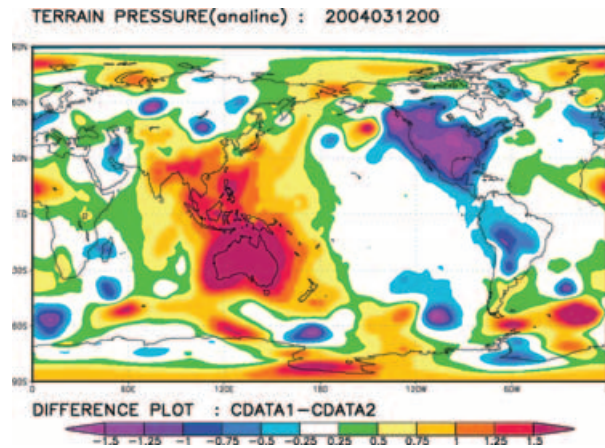


Fig. 5. Terrain pres difference, outer loop 1 – outer loop 2. Units = hPa.

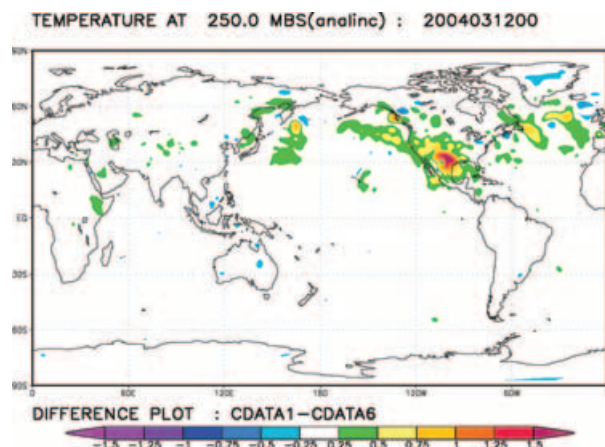


Fig. 6. 250-hPa temperature difference, outer loop 1 – outer loop 6. Units = °C.

primarily because the operational NAVDAS is assimilating the data also, although not with the same observation operator used in NAVDAS-AR. Nevertheless, we see noticeable changes due to incorporation of SSM/I precipitable water over all ocean areas, a strong indication that the data should have significant impact on NOGAPS moisture and precipitation forecasts in these areas.

6. Future plans

We are now in a position to take the next logical step in NAVDAS-AR development: running the system in full data assimilation mode to produce initial conditions for the NOGAPS forecast model that give competitive forecast skill to those coming from the operational NAVDAS. Several components must still be added to the system before this can be accomplished, however. A separate univariate 2DVAR analysis for subterranean sea level pressure is currently run as part of the operational NAVDAS;

this must also be added to the NAVDAS-AR system. A greater challenge, however, is the incorporation of satellite radiance retrievals into NAVDAS-AR. The observation operators $\mathcal{H}$, $\mathbf{H}$ and $\mathbf{H}^T$ must be adapted from the operational NAVDAS to NAVDAS-AR. Experience has shown that although direct radiance assimilation can have significant positive impact on forecast system performance, a great deal of testing and tuning is required to achieve this impact.

In its weak constraint observation space formulation, NAVDAS-AR has the unique advantage of allowing the addition of a model error term $\mathbf{Q}$ on the right-hand side of eq. (16), without a change in the size of the control variable. The form that $\mathbf{Q}$ should take is the source of considerable research in the data assimilation community, and we expect to investigate its form in future research with NAVDAS-AR.

Computational cost is an obvious issue for any operational 4DVAR assimilation system, and NAVDAS-AR is no exception to this reality. Currently, NAVDAS-AR is five to six times as expensive as the operational NAVDAS. Next-generation computer systems will certainly deliver the necessary computational power to cover this extra cost, but there is still considerable motivation to improve the system's computational efficiency, because the cost of development testing has a direct impact on the rate at which we can move the system toward operational implementation. In Fig. 2 we show the convergence properties of the conjugate gradient algorithm without preconditioning. In the operational NAVDAS, a Choleski-based preconditioner almost doubles the rate of conjugate gradient convergence, but this approach cannot be applied in NAVDAS-AR because of the time integration operators in the solver equation. A promising approach is the so-called Lanczos connection (Golub and Van Loan, 1996), which finds the leading eigenvalues and eigenvectors of the solver equation matrix, from which an approximate inverse can be constructed. This is the current direction of our research in this area.

NAVDAS-AR is a message passing interface (MPI) application, so scalability is a critical issue for overall system computational efficiency. The cost of NAVDAS-AR is dominated by the inner loop computations, divided between the integration of the NOGAPS tangent-linear $\mathbf{M}$ and adjoint $\mathbf{M}^T$ models, and the background covariance calculations with $\mathbf{P}_0^b$. The models are essentially dynamical cores² and scale very well to processor numbers of $O(100)$, but the $\mathbf{P}_0^b$ operations do not scale as well, primarily because of a complex observation space search algorithm taken from NAVDAS that is unnecessary in NAVDAS-AR. This is because although both NAVDAS and NAVDAS-AR are observation space based, the background error covariance calculations are quite different in the two systems. NAVDAS-AR performs these calculations on the native computational grid of the NOGAPS model, but we are not yet exploiting this feature.

²Adding some form of linearized diabatic processes is part of our long term goals, but we do not regard it as a top priority at this time.

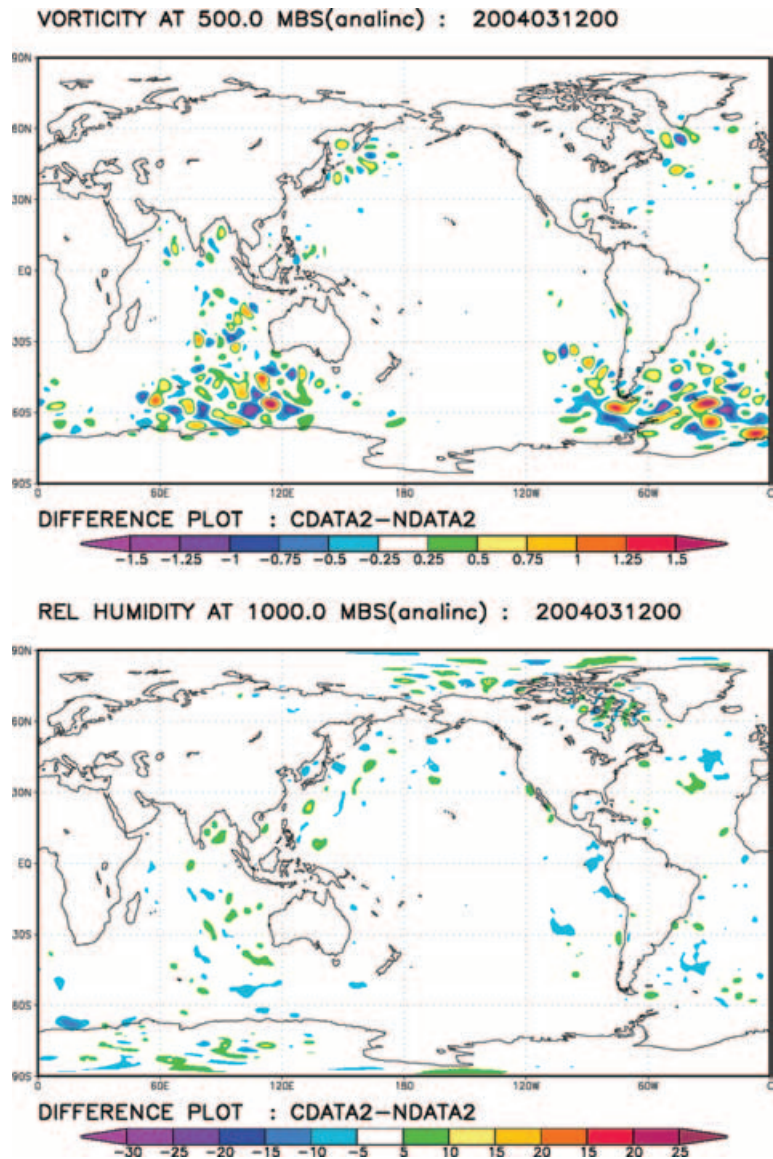


Fig. 7. Top: difference of 500-hPa vorticity due to assimilation of SSM/I surface wind speeds. Units = 10^{-6} s^{-1} . Bottom: difference of 1000-hPa relative humidity due to assimilation of SSM/I precipitable H_2O .

As an initial expedience, we adapted the NAVDAS observation space $\mathbf{P}_0^b$ algorithm by mimicking observations at the NOGAPS grid locations. This comes with a significant amount of extra overhead that can be eliminated if we redesign the system for the actual grid geometry. This is a top priority for operational implementation.

7. Summary and Conclusions

In this paper we present results and a progress report on the development of NAVDAS-AR, an accelerated representer form of a weak constraint observation space 4DVAR algorithm. NAVDAS-AR is the basis for the United States Navy's next generation atmospheric data assimilation system. We have reached an important milestone in the development of NAVDAS-AR, the

implementation of an outer loop strategy that allows the introduction of non-linear observation operators. Preliminary tests show rapid convergence of the outer loops: two loops achieves over 90% of the non-linear adjustments. We are now ready to proceed to the next step toward operational implementation; cycling the system in a full data assimilation mode and assessing the impact of NOGAPS forecast system performance. Computational issues still remain, however, because NAVDAS-AR is significantly more expensive than the 3DVAR operational NAVDAS. Next-generation computer systems will remove most of these obstacles, but there is still room for improvement in system efficiency.

The benefits of 4DVAR data assimilation have been clearly demonstrated by the experience at ECMWF. NAVDAS-AR has

the potential to deliver the same benefits to Navy operational NWP. An important by-product of our efforts will be comparisons of the relative advantages and disadvantages of the ECMWF type analysis space algorithm versus the NAVDAS observation space algorithm. The impact of an imperfect model assumption (weak constraint formulation) on the two approaches will certainly be one of the most intriguing research questions. We know that an observational space algorithm has a theoretical advantage here because the size of the control variable remains unchanged in the weak constraint case. However, in XRD we point out that the size of the control variable only impacts the cost of the observation operators, a relatively minor part of the total cost of NAVDAS-AR compared to the cost of model integrations and covariance matrix convolutions. Adding an expensive model error covariance term will further reduce the impact of control variable size on overall computational cost. We also suggest that even with the substantially larger control variable in a weak constraint analysis space algorithm (i.e. the ECMWF system) its impact on computational cost will be minimal compared to the cost of a model error convolution. Future research in the data assimilation community will certainly answer this question.

Another significant difference between NAVDAS-AR and the ECMWF system is the use of an incremental form of the 4DVAR algorithm by ECMWF. We investigated a similar incremental approach for NAVDAS-AR and decided not to pursue it at this time. We did so because (a) we found our full variable form performed quite satisfactorily, (b) the representer method does not lend itself to an incremental method, and (c) the original theoretical background work on the accelerated representer by Xu and Daley (2002) was done in a full variable form. At this point we want to pursue NAVDAS-AR as they originally envisioned it, although in the future we may revisit an incremental form if a clear advantage is apparent.

We are now ready to bring NAVDAS-AR to a full prototype operational configuration. In the future we will be reporting on results of data assimilation experiments and the impact of NAVDAS-AR on operational NOGAPS forecast performance. In addition, we will contribute to answering the questions on the relative merits of analysis space/observation space 4DVAR algorithms, such as those posed above. The data assimilation community will clearly benefit from the answers, whatever they may be.

8. Acknowledgments

We are deeply indebted to our colleague, the late Roger Daley, who formulated and initiated the development of NAVDAS-AR. We would also like to thank Andrew Bennett and Boon Chua of Oregon State University, and Ed Barker of NRL in Monterey for their encouragement and help during the development. Support of the sponsor, the NRL, under base program elements 0601153N and 0602436N, is gratefully acknowledged. We also

thank the two anonymous reviewers who gave us numerous helpful comments and corrections, which greatly improved the manuscript.

9. Appendix A: Derivation of eq. (21)

Let $(\alpha_k)^j$ and $(\gamma_k)^j$ be the adjoint field and the representer function for the k th observation, respectively. Also let $(\alpha_k)^j$ and $(\gamma_k)^j$ satisfy the following equations

$$\begin{aligned} (\alpha_k)^j_n - [\mathbf{M}_n^{j-1}]^T (\alpha_k)^j_{n+1} &= [(\mathbf{H}_k)^{j-1}]^T, \\ \text{for } 1 \leq n \leq N-1, \text{ and subject to} \\ (\alpha_k)^j_N &= [(\mathbf{H}_k)^{j-1}]^T \end{aligned} \quad (\text{A1})$$

and

$$\begin{aligned} (\gamma_k)^j_n - \mathbf{M}_{n-1}^{j-1} (\gamma_k)^j_{n-1} &= 0, \\ \text{for } 1 \leq n \leq N, \text{ and subject to} \\ (\gamma_k)^j_0 &= \mathbf{P}_0^b [\mathbf{M}_0^{j-1}]^T (\alpha_k)^j_1. \end{aligned} \quad (\text{A2})$$

Let

$$\begin{aligned} [(\alpha_k)^j]^T &= \left[[(\alpha_k)^j_1]^T \dots [(\alpha_k)^j_n]^T \dots [(\alpha_k)^j_N]^T \right] \\ [(\gamma_k)^j]^T &= \left[[(\gamma_k)^j_1]^T \dots [(\gamma_k)^j_n]^T \dots [(\gamma_k)^j_N]^T \right] \\ [(\mathbf{H}_k)^{j-1}]^T &= \left[[(\mathbf{H}_k)^{j-1}_1]^T \dots [(\mathbf{H}_k)^{j-1}_n]^T \dots [(\mathbf{H}_k)^{j-1}_N]^T \right]. \end{aligned} \quad (\text{A3})$$

Also, let $\mathbf{M}^{j-1}$ be a matrix with $\mathbf{0}$ upper triangle, and $[\mathbf{M}^{j-1}]^T$ be its transpose:

$$\mathbf{M}^{j-1} = \begin{pmatrix} \mathbf{I} & & & \mathbf{0} \\ \prod_{n'=1}^1 \mathbf{M}_{n'}^{j-1} & & & \\ & & & \\ & & & \\ \prod_{n'=N-1}^1 \mathbf{M}_{n'}^{j-1} & & \prod_{n'=N-1}^{N-1} \mathbf{M}_{n'}^{j-1} & \mathbf{I} \end{pmatrix}. \quad (\text{A4})$$

With the definitions above, we may now rewrite eqs. (A1) and (A2) in the following compact forms:

$$(\alpha_k)^j = [\mathbf{M}^{j-1}]^T [(\mathbf{H}_k)^{j-1}]^T, \quad (\text{A5})$$

$$(\gamma_k)^j = \mathbf{M}^{j-1} \mathbf{M}_0^{j-1} \mathbf{P}_0^b [\mathbf{M}_0^{j-1}]^T (\alpha_k)^j. \quad (\text{A6})$$

We also have

$$(\mathbf{P}^b)^{j-1} = \mathbf{M}^{j-1} \left[\mathbf{M}_0^{j-1} \mathbf{P}_0^b \left[\mathbf{M}_0^{j-1} \right]^T \right] \left[\mathbf{M}^{j-1} \right]^T, \quad (\text{A7})$$

where $(\mathbf{P}^b)^{j-1}$ is the background error covariance used in the j th iteration.

According to representer theory, the analysis $\mathbf{x}^j$ for the j th iteration is simply the background $(\mathbf{x}^p)^j$ plus a correction term. The correction term is essentially a linear combination of all the representer functions, $(\gamma_k)^j$, with the coefficients being β_k^j for $k = 1, K$. The solution to the coupled linearized Euler–Lagrange system (15)–(17) can now be written as

$$\mathbf{x}^j - (\mathbf{x}^p)^j = \sum_{k=1}^K (\gamma_k)^j \beta_k^j = (\mathbf{P}^b)^{j-1} \left[\mathbf{H}^{j-1} \right]^T \beta^j. \quad (\text{A8})$$

For the special case at $j = 1$ we use eq. (10) and set $(\mathbf{x}^p)^0 = \mathbf{x}^0 = \mathbf{x}^b$ and $\delta(\mathbf{x}^p)^0 = 0$.

Recalling $\mathbf{x}^j = \mathbf{x}^{j-1} + \delta\mathbf{x}^j$, $(\mathbf{x}^p)^j = \mathbf{x}^{j-1} + \delta(\mathbf{x}^p)^j$, and $\delta\mathbf{x}^j - \delta(\mathbf{x}^p)^j = \mathbf{x}^j - (\mathbf{x}^p)^j$, we can rewrite eq. (A8) as

$$\delta\mathbf{x}^j - \delta(\mathbf{x}^p)^j = \sum_{k=1}^K (\gamma_k)^j \beta_k^j = (\mathbf{P}^b)^{j-1} \left[\mathbf{H}^{j-1} \right]^T \beta^j, \quad (\text{A9})$$

where β^j is the coupling vector or the analysis residual normalized by the observation error covariance for the j th iteration and satisfies the following:

$$\beta^j = \left[\mathbf{H}^{j-1} (\mathbf{P}^b)^{j-1} \left[\mathbf{H}^{j-1} \right]^T + \mathbf{R} \right]^{-1} \times [\mathbf{y} - \mathcal{H}(\mathbf{x}^{j-1}) - \mathbf{H}^{j-1} \delta(\mathbf{x}^p)^j]. \quad (\text{A10})$$

Notice that β_k^j is a constant related to the k th observation and is the k th component of vector $[\beta^j]^T = [\beta_1^j \dots \beta_K^j]$. The detailed derivation of eq. (A10) can be found in Appendix B.

10. Appendix B: Derivation of eq. (22)

Define a matrix $\mathbf{A}$ consisting of $N \times N$ blocks, each of which is an $I \times I$ matrix. For the $(j - 1)$ th iteration, we have $\mathbf{A}^{j-1}$

$$\begin{pmatrix} \mathbf{I} & -[\mathbf{M}_1^{j-1}]^T & \cdot & \cdot & \cdot & \mathbf{0} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \mathbf{I} & -[\mathbf{M}_n^{j-1}]^T & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \mathbf{0} & \cdot & \cdot & \cdot & \cdot & \mathbf{I} \end{pmatrix} \quad (\text{B1})$$

where $\mathbf{I}$ is an $I \times I$ identity matrix. We may now rewrite eq. (12) as

$$\mathbf{A}^{j-1} \lambda^j = \left[\mathbf{H}^{j-1} \right]^T \mathbf{R}^{-1} \times [\mathbf{y} - \mathcal{H}(\mathbf{x}^{j-1}) - \mathbf{H}^{j-1} \delta\mathbf{x}^j], \quad (\text{B2})$$

and eq. (A1) as

$$\mathbf{A}^{j-1} (\alpha_k)^j = \left[(\mathbf{H}_k)^{j-1} \right]^T. \quad (\text{B3})$$

Now define a matrix $\mathbf{F}$ consisting of $N \times N$ blocks, each of which is an $I \times I$ matrix, for the $(j - 1)$ th iteration. We have $\mathbf{F}^{j-1}$

$$\begin{pmatrix} \mathbf{I} & \cdot & \cdot & \cdot & \cdot & \mathbf{0} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & -\mathbf{M}_{n-1}^{j-1} & \mathbf{I} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \mathbf{0} & \cdot & \cdot & \cdot & -\mathbf{M}_{N-1}^{j-1} & \mathbf{I} \end{pmatrix} \quad (\text{B4})$$

Subtracting eq. (14) from eq. (13) and using eq. (B4), we have

$$\mathbf{F}^{j-1} \delta^* \mathbf{x}^j = \mathbf{M}_0^{j-1} \mathbf{P}_0^b \left[\mathbf{M}_0^{j-1} \right]^T \lambda^j, \quad (\text{B5})$$

where

$$[\delta^* \mathbf{x}^j]^T = \left[[\delta^* \mathbf{x}_1^j]^T \dots [\delta^* \mathbf{x}_n^j]^T \dots [\delta^* \mathbf{x}_N^j]^T \right],$$

and

$$\delta^* \mathbf{x}_n^j = \mathbf{x}_n^j - (\mathbf{x}^p)_n^j \text{ for } 1 \leq n \leq N.$$

Equation (A2) becomes

$$\mathbf{F}^{j-1} \gamma_k^j = \mathbf{M}_0^{j-1} \mathbf{P}_0^b \left[\mathbf{M}_0^{j-1} \right]^T (\alpha_k)^j. \quad (\text{B6})$$

Applying $\mathbf{F}^{j-1}$ to eq. (A8), we have

$$\begin{aligned} \mathbf{F}^{j-1} \delta^* \mathbf{x}^j &= \sum_{k=1}^K \left[\mathbf{F}^{j-1} \gamma_k^j \right] \beta_k^j \\ &= \sum_{k=1}^K \left[\mathbf{M}_0^{j-1} \mathbf{P}_0^b \left[\mathbf{M}_0^{j-1} \right]^T (\alpha_k)^j \right] \beta_k^j \\ &= \mathbf{M}_0^{j-1} \mathbf{P}_0^b \left[\mathbf{M}_0^{j-1} \right]^T \alpha^j \beta^j. \end{aligned} \quad (\text{B7})$$

Comparing eqs. (B5) and (B7), we have

$$\lambda^j = \sum_{k=1}^K (\alpha_k)^j \beta_k^j = \alpha^j \beta^j. \quad (\text{B8})$$

Applying $\mathbf{A}^{j-1}$ to eq. (B8) and using eqs. (B2) and (B3), we have

$$\begin{aligned} &\left[\mathbf{H}^{j-1} \right]^T \mathbf{R}^{-1} [\mathbf{y} - \mathcal{H}(\mathbf{x}^{j-1}) - \mathbf{H}^{j-1} \delta\mathbf{x}^j] \\ &= \mathbf{A}^{j-1} \lambda^j \sum_{k=1}^K \left[\mathbf{A}^{j-1} (\alpha_k)^j \right] (\beta_k)^j = \left[\mathbf{H}^{j-1} \right]^T \beta^j. \end{aligned} \quad (\text{B9})$$

From eq. (B9), we now have

$$\mathbf{R}^{-1} [\mathbf{y} - \mathcal{H}(\mathbf{x}^{j-1}) - \mathbf{H}^{j-1} \delta\mathbf{x}^j] = \beta^j. \quad (\text{B10})$$

Substituting for $\delta \mathbf{x}^j$ from eq. (A9) into eq. (B10), we have

$$\mathbf{y} - \mathcal{H}(\mathbf{x}^{j-1}) - \mathbf{H}^{j-1} \delta(\mathbf{x}^p)^j - \mathbf{H}^{j-1} (\mathbf{P}^b)^{j-1} [\mathbf{H}^{j-1}]^T \beta^j = \mathbf{R} \beta^j \quad (\text{B11})$$

or

$$\beta^j = \left[\mathbf{H}^{j-1} (\mathbf{P}^b)^{j-1} [\mathbf{H}^{j-1}]^T + \mathbf{R} \right]^{-1} \times [\mathbf{y} - \mathcal{H}(\mathbf{x}^{j-1}) - \mathbf{H}^{j-1} \delta(\mathbf{x}^p)^j], \quad (\text{B12})$$

which is the same as eq. (A10).

11. Appendix C: Simple Picard iteration model

A Picard model is a linearized form with a relaxation term forcing the solution toward some desired trajectory. In NAVDAS-AR we want to relax the NOGAPS forecast model toward the analysis trajectories of our outer loop iterations. To demonstrate the technique, consider a one-dimensional advection equation

$$\partial \Psi^p / \partial t = -\mathbf{U}^p \partial \Psi^p / \partial \mathbf{x}. \quad (\text{C1})$$

Let

$$\Psi^p = \Psi^a + (\Psi^p - \Psi^a), \quad (\text{C2})$$

and

$$\mathbf{U}^p = \mathbf{U}^a + (\mathbf{U}^p - \mathbf{U}^a), \quad (\text{C3})$$

where ‘p’ superscripts identify the Picard model variables and ‘a’ superscripts identify the analysis trajectory variables produced from the previous outer loop. We substitute eqs. (C2) and (C3) into eq. (C1) and expand the terms, dropping the second-order term $(\mathbf{U}^p - \mathbf{U}^a) \partial / \partial \mathbf{x} (\Psi^p - \Psi^a)$ as in any linearization. This yields

$$\partial \Psi^p / \partial t = -\mathbf{U}^a \partial \Psi^p / \partial \mathbf{x} - (\mathbf{U}^p - \mathbf{U}^a) \partial \Psi^a / \partial \mathbf{x}. \quad (\text{C4})$$

The second term on the right-hand side is the relaxation term toward the analysis trajectory.

In our application, the dynamical core equations of the NOGAPS spectral model are linearized as shown here, producing a Picard NOGAPS model relaxed toward the analysis.

References

Bennett, A. F. 2002. *Inverse Modeling of the Ocean and Atmosphere*. Cambridge University Press, Cambridge, 234 pp.

Bennett, A. F. and McIntosh, P. C. 1982. Open ocean modeling as an inverse problem: tidal theory. *J. Phys. Oceanography* **12**, 1004–1018.

Bennett, A. F., and Thorburn, M. A. 1992. The generalized inverse of a nonlinear quasigeostrophic ocean circulation model. *J. Phys. Oceanography* **22**, 213–230.

Chua, B. S. and Bennett, A. F. 2001. An inverse ocean modeling system. *Ocean Modeling* **3**, 137–165.

Cohn, S., da Silva, A., Guo, J., Sienkiewicz, M. and Lamich, D. 1998. Assessing the effects of data selection with the DAO physical-space Statistical Analysis System. *Mon. Wea. Rev.* **126**, 2913–2926.

Courtier, P. 1997. Dual formulation of four-dimensional data assimilation. *Q. J. R. Meteorol. Soc.* **123**, 2449–2461.

Courtier, P., Thepaut, J. N. and Hollingsworth, A. 1994. A strategy for operational implementation of 4D-VAR, using an incremental approach. *Q. J. R. Meteorol. Soc.* **120**, 1367–1387.

Coutier, P., Andersson, E., Heckley, W., Pailleux, J., Vasiljevic, D. and co-authors. 1998. The ECMWF implementation of three dimensional assimilation (3DVAR). Part 1: formulation. *Q. J. R. Meteorol. Soc.* **124**, 1783–1807.

Daley, R. and Barker, E. 2001a. *The NAVDAS Source Book*. NRL/PJ/7530-01-441. Naval Research Laboratory. 163 pp.

Daley, R. and Barker, E. 2001b. NAVDAS - formulation and diagnostics. *Mon. Wea. Rev.* **129**, 869–883.

Derber, J., Wu, H. W., Zupanski, M., Parrish, D., Purser, J. and co-authors. 1996. *Variational assimilation at NCEP. Proceedings of the workshop on the non-linear aspects of data assimilation*, 9–11 September 1996. ECMWF, Reading, UK.

Golub, G. H. and Van Loan, C. F. 1996. *Matrix Computations*, 3rd Edition. Johns Hopkins University Press, Baltimore, MD, 694 pp.

Hogan, T. and Rosmond, T. 1991. The description of the Navy Operational Global Atmospheric Prediction System’s spectral forecast model. *Mon. Wea. Rev.* **119**, 1786–1815.

Rabier, F., Thepaut, J. N. and Courtier, P. 1998. Extended assimilation and forecast experiments with a four-dimensional variational assimilation system. *Q. J. R. Meteorol. Soc.* **124**, 1861–1887.

Rabier, F., Javinen, H., Klinker, E., Mahfouf, J.-F. and Simmons, A. 2000. The ECMWF implementation of four-dimensional variational assimilation. I: experimental results with simplified physics. *Q. J. R. Meteorol. Soc.* **126**, 1143–1170.

Veerse, F. and Thepaut, J. N. 1998. Multiple-truncation incremental approach for four-dimensional variational data assimilation. *Q. J. R. Meteorol. Soc.* **124**, 1880–1908.

Xu, L. and Daley, R. 2000. Towards a true four-dimensional data assimilation algorithm: application of a cycling representer algorithm to a simple transport problem. *Tellus* **52A**, 109–128.

Xu, L. and Daley, R. 2002. Data assimilation with a barotropically unstable shallow water system using representer algorithms. *Tellus* **54A**, 125–137.

Xu, L. and Rosmond, T. 2004. *Formulation of the NRL Atmospheric Variational Data Assimilation System - Accelerated Representer (NAVDAS-AR)*. NRL/MR/7532-04-36. Naval Research Laboratory, 28 pp.

Xu, L., Rosmond, T. and Daley, R. 2005. Development of NAVDAS-AR. Formulation and initial tests of the linear problem. *Tellus* **57A**, 546–559.