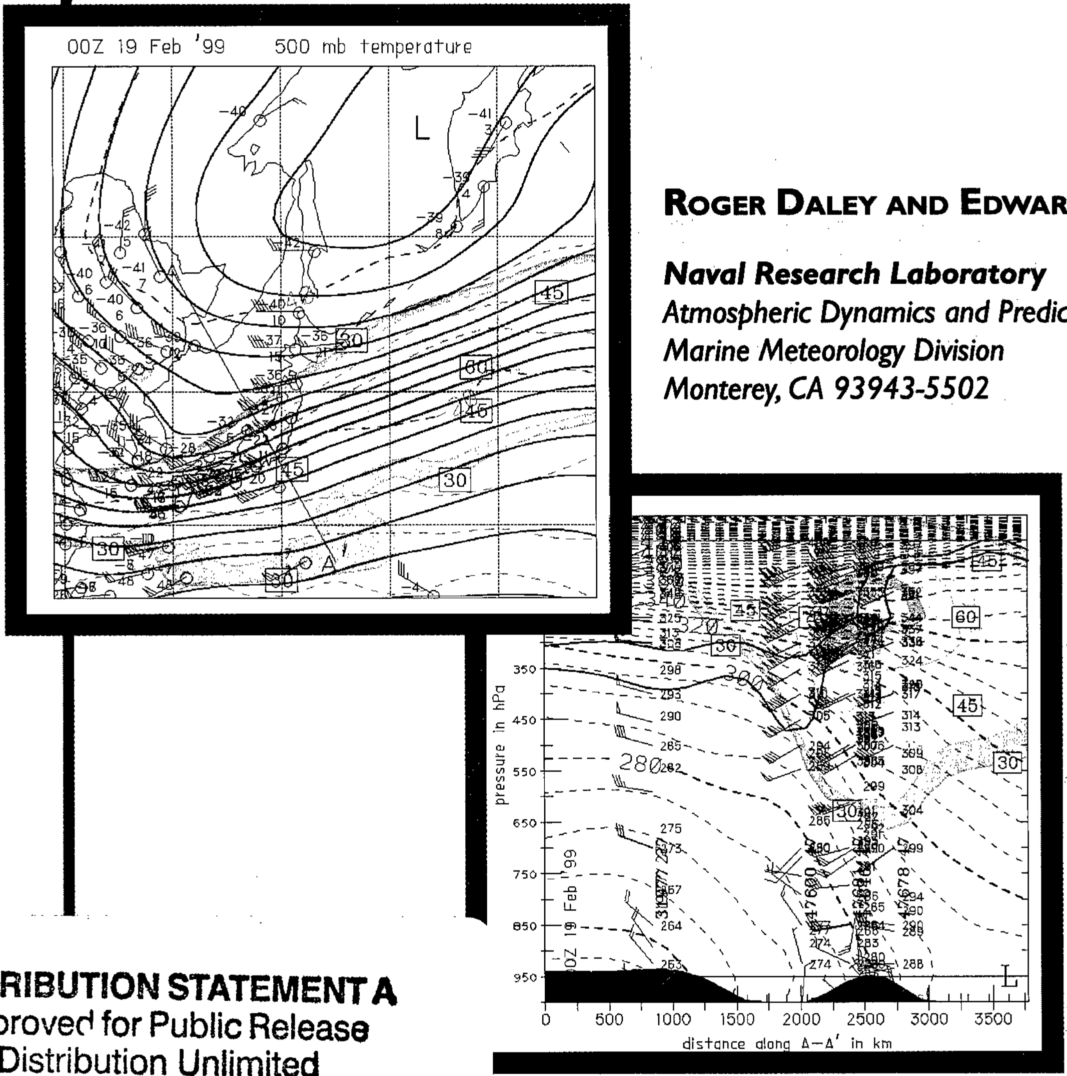
NAVDAS Source Book 2000

NRL Atmospheric Variational Data Assimilation System



Roger Daley and Edward Barker

Naval Research Laboratory
Atmospheric Dynamics and Prediction Branch
Marine Meteorology Division
Monterey, CA 93943-5502

Cover: NAVDAS analysis over Japan and Siberia for February 19, 1999 00Z. Top panel: 500 hPa temperature (solid contours) and windspeed (colored). Lower panel: pressure cross-section along line AA′ marked in top panel. Orography marked in solid black. Isentropes – dashed lines, windspeed – colored. Observations of temperature, windspeed, and direction are shown in both panels.

NRL Publication
NRL/PU/7530--00-418

Contents

1. Introduction

The development of the NRL Atmospheric Variational Data Assimilation System (NAVDAS) began in 1996. At that time, both the regional and global atmospheric data assimilation requirements of Fleet Numerical Meteorological and Oceanographic Center (FNMOC) were met with versions of a multivariate optimal interpolation algorithm (MVOI), originally developed in the mid-1980s (Barker, 1992; Goerss and Phoebus, 1992). The NRL MVOI algorithm was based on the most powerful formulation of the problem available at that time, that of Andrew Lorenc (1981) at theEuropean Centre for Medium Range Forecasting (ECMWF). In particular, the NRL MVOI used a box or volume formulation that permitted a few hundred observations located in the same region to be processed simultaneously, thus minimizing (but not eliminating) data selection. Moreover, the forecast error covariance was (in conception, but not in execution) reasonably general, permitting the geostrophic and nondivergence constraints to be imposed weakly, if desired.

However, in the years since the implementation of the NRL MVOI system, great strides had been made in atmospheric data assimilation-both in the academic world and in other operational centers. Firstly, the OI algorithm had been generalized to the three- dimensional variational (3DVAR) algorithm. Like the OI algorithm, this was a static three-dimensional algorithm in which all the observations over a particular time window were processed simultaneously (as if they were all valid at exactly the same time) and the time evolution was entirely handled by the evolution of the forecast (or background) field. Compared to the OI algorithm, the 3DVAR algorithm had several advantages:

(1) A global solution was obtained—there was no data selection

(2) Many observation types that are difficult to handle properly with the OI algorithm could be handled properly in 3DVAR. An example was the direct assimilation of radiances from polar-orbiting sounders.

(3) More powerful and realistic formulations of the error covariances (required in generating the analysis weights) were possible.

3DVAR algorithms were deployed operationally (for the global problem) at the National Centers for Environmental Prediction (NCEP) in 1992 (in a rudimentary form) and in more mature form at ECMWF in 1995. After 1995, they became operational at the Data Assimilation Office (DAO) at the NASA/Goddard Space Flight Center, at the Canadian Meteorological Center (CMC) in Montreal, and at Meteo France. The general properties of 3DVAR algorithms are covered in Section 2.

Beyond the 3DVAR algorithms, there were several classes of four-dimensional algorithms, in which the analysis weights (as well as the background fields) could evolve (either implicitly, as in the case of the four-dimensional variational (4DVAR) algorithm, or explicitly, as in the case of the Extended Kalman Filter (EKF)). These formulations permitted flow-dependent analysis weights, with more complicated relationships between variables than the simple linear relationships of geostrophy or nondivergence. The observations could be inserted at the correct times, without binning over a time interval. However, these algorithms were much more complex and computer-intensive than the three-dimensional algorithms.

An assessment of the NRL/FNMOC data assimilation situation led to the following strategy:

(1) The first priority was the development of a competitive, state-of-the art 3DVAR system. Considering that the leading centers were some years ahead of NRL in this respect, existent NRL expertise and software would have to be exploited heavily. The resulting 3DVAR system would have to be implementable for the global, regional, and shipboard problem with a common code. The 3DVAR code was not to be thought of as a mere way station on the way to a four-dimensional algorithm-it had to stand on its own merits. The implementation of features that permitted limited four-dimensional capability was not precluded and has, in fact, been pursued.

(2) Four-dimensional data assimilation was relegated to a slower track. There were two reasons for this decision. First, it was felt that the work on four-dimensional algorithms at other institutions was too immature to warrant substantial investment in a particular four-dimensional algorithm. Second, it was felt that there were neither the human resources, nor sufficient commitment of computer resources to tackle the problem at NRL on a compressed timetable. However, it was felt that a relatively modest research program (of 6.1 type) would be a useful long-term strategy. To this end, a four-dimensional algorithm called the cycling representer algorithm was developed and applied to simple data assimilation problems. The present text does not discuss any of this work. However, we note that this algorithm is a logical extension of the NAVDAS algorithm described here (see Xu and Daley, 2000).

The first step in formulating a 3DVAR strategy was to examine work in the field and visit existing 3VAR programs at other institutions. Thus, in September 1995, Roger Daley, Ed Barker, and Nancy Baker visited NCEP to examine the spectral statistical interpolation (SSI) code of David Parrish and John Derber (1992) and the Data Assimilation Office (DAO) at NASA/Goddard to examine the Physical Space Assimilation System (PSAS) of Steve Cohn et al. (1998).

Work then began using one- (x) and two-dimensional horizontal (x,y) univariate assimilation systems to select the most promising 3DVAR algorithm (Section 2) and find the most appropriate descent method for the selected 3DVAR algorithm (Section 3). This emphasis on simpler models was found to be very valuable in that it allowed a considerable amount of experience to be gained quickly. These simple model techniques were later extended to the generation of multivariate horizontal (x,y) and cross-section (x,z) 3DVAR codes. This philosophical approach to the solution of practical data assimilation problems was similar to that of Daley (1985) and Daley, Wergen and Cats (1986) at ECMWF. Sections 2 and 3, which cover most of this material, serve as a useful pedagogical introduction to the 3DVAR algorithm.

This experimentation led to the decision in March, 1996 to construct an observation space 3DVAR system using pre-conditioned conjugate gradient descent. This was to be a completely new system. In particular, the data handling and model interfaces were to be rewritten in a much more general way, so that the potential power of the 3DVAR algorithm was not lost because of unnecessary compromises or approximations that had been made in the older NRL MVOI system. This work was eventually to lead to NAVDAS, although the acronym was not devised until 1999.

The bulk of the coding was done by the two authors, with a very clear division of labor. The work was divided into three parts.

(1) Data ingest, quality control, processing the background fields, generation of the observation error statistics, production of the linearized forward operators (Section 5) and formation of the innovation vectors (Section 7) was the responsibility of Edward Barker.

(2) Sorting of the innovations into prisms (Section 3), specification of the background error covariance (Section 4), the descent algorithm (Section 3), the postmultiplier (Section 3), use of the linearized forward operators (Section 5), production of the correction vectors, the buddy check algorithm (Section 9) and parallel implementation (Appendix C) using MPI (Message Passing Interface) was the responsibility of Roger Daley.

(3) Interpolation of the correction field to the model surfaces, mean sea level pressure, and other subterrain problems; production of statistics for both the Naval Operational Global Atmospheric Prediction System (NOGAPS) and the Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS) models (Section 8); and controlling the forecast/analysis cycle was the responsibility of Edward Barker.

Other NRL scientists made invaluable contributions to the project. Quality control software for AMDAR and cloud-drift and water vapor winds observations was written and tested by Patricia Pauley. The one-dimensional variational retrieval of TOVS radiances, together with bias correction, quality control, and production of the linearized forward operators was done by Nancy Baker. Steve Swadley produced a Web-based graphical data monitoring system for NAVDAS. An Australian visitor (Peter Steinle) constructed the isentropic version of NAVDAS.

A (non-nested) COAMPS cycle was successfully tested in the spring of 1999 for the CALJET experiment (California coast). Responsibility for the COAMPS cycle, particularly the development of the nested version, was assumed by Keith Sashegyi. The NOGAPS cycle was first tested successfully in December 1999 and responsibility for its implementation was assumed by Jim Goerss.

2. General Theory

This section covers the derivation of the 3DVAR algorithm, the analysis grid space, observation space forms of the algorithm, and various properties of the algorithm. Experiments are performed with simplified versions of three alternate solution schemes, ending with the choice of a particular algorithm.

2.1 Derivation of the 3DVAR algorithm

This derivation follows Daley (1991, section 13.1) and uses the increasingly common notation displayed in Daley (1997). The main text in this area is Tarantola (1987) and other useful sources are Lorenc (1986), Heckley et al. (1992), Parrish and Derber (1992), Courtier et al. (1997), and Cohn et al. (1998). At some time (t) define $\mathbf{x}_b$ and $\mathbf{x}_a$ as column vectors of forecast (background or prior) and analyzed values on some regular analysis grid $\mathbf{r}_i$, where $1 \leq I \leq I$. Define $\mathbf{x}$ to be the vector of length I of true values of the variables at the same gridpoints. (Note that spectral coefficients could be used instead of gridpoint values). Then, define the background error vector as $\mathbf{e}_b = \mathbf{x}_b - \mathbf{x}$. We assume that the background error is unbiased but may be correlated. Thus, $<\mathbf{e}_b> = 0$ and $<\mathbf{e}_b(\mathbf{e}_b)^T> = \mathbf{P}_b$ is the IxI square, symmetric positive-definite background error covariance matrix. (Angle brackets indicated expected value, superscript "T" indicates matrix transpose, and positive-definiteness implies that all the eigenvalues of $\mathbf{P}_b$ are real and positive.)

Define $\mathbf{y}$ as a column vector of length L of observations. L is generally different than I, and the variables of $\mathbf{x}$ may be different than the variables of $\mathbf{y}$. Thus, $\mathbf{x}$ might be winds and temperatures, while $\mathbf{y}$ might be radiances in different channels from some remote sensing device and/or radar reflectivity. Define H as a forward operator from the analysis/background grid variables $\mathbf{x}_a$, $\mathbf{x}_b$ to the observed variables $\mathbf{y}$. For example, if $\mathbf{x}$ is temperature and $\mathbf{y}$ is radiance, then H would be the radiative transfer equation (perhaps nonlinear). Then, $\mathbf{y} = H(\mathbf{x}) + \mathbf{e}_r$, where $\mathbf{e}_r$ is the error in the observation. This error has two sources, the instrument error and the error in the forward model. In cases where $\mathbf{x}$ and $\mathbf{y}$ refer to the same variables, but the spatial location of $\mathbf{y}$ is not a gridpoint, then the operator H is simply an interpolation operator and the error in the forward model is simply the error of representativeness. We assume that $\mathbf{e}_r$ is unbiased $<\mathbf{e}_r> = 0$ but may be spatially correlated with the L×L square, symmetric positive-definite observation error covariance matrix $\mathbf{R} = <\mathbf{e}_r(\mathbf{e}_r)^T>$.

Assuming that the background and observation errors are distributed normally, then it can be shown (see Daley 1991, section 2.2) that the most probable (i.e., maximum likelihood estimate) analysis state vector $\mathbf{x}_a$ is obtained by minimizing the scalar cost function J with respect to $\mathbf{x}_a$, where

$$J = 0.5[\mathbf{y} - H(\mathbf{x}_a)]^T\mathbf{R}^{-1}[\mathbf{y} - H(\mathbf{x}_a)] + 0.5[\mathbf{x}_b - \mathbf{x}_a]^T\mathbf{P}_b^{-1}[\mathbf{x}_b - \mathbf{x}_a], \qquad (2.1)$$

(Note that we have also assumed that the observation and background errors are not mutually correlated).

Differentiation of the scalar J with respect to the vector $\mathbf{x}_a$ produces a vector ∇J, which is known as the gradient of J with respect to $\mathbf{x}_a$,

$$\nabla J = \mathbf{H}^T\mathbf{R}^{-1}[H(\mathbf{x}_a) - \mathbf{y}] + \mathbf{P}_b^{-1}[\mathbf{x}_a - \mathbf{x}_b]. \qquad (2.2)$$

The matrix **H** is the Jacobian matrix corresponding to the (possibly) nonlinear forward operator H. (If H is linear, then **H** and H are the same). If H is nonlinear, as when H is the radiative transfer equation, then **H** is defined as $\mathbf{H} = \partial H(\mathbf{x}) / \partial \mathbf{x}$ evaluated at $\mathbf{x} = \mathbf{x}_a$. Thus, the Jacobian matrix is a matrix whose elements consist of partial derivatives of H with respect to $\mathbf{x}$. Of course, we do not know $\mathbf{x}_a$ when we attempt to evaluate **H**, so we are forced to use the background $\mathbf{x}_b$ to evaluate it. Thus, if we assume that $\mathbf{x}_b$ is not too far from $\mathbf{x}_a$, then we can expand $H(\mathbf{x}_a)$ in the first two terms of a Taylor series around $\mathbf{x} = \mathbf{x}_b$,

$$H(\mathbf{x}_a) = H(\mathbf{x}_b + [\mathbf{x}_a - \mathbf{x}_b]) = H(\mathbf{x}_b) + \mathbf{H}[\mathbf{x}_a - \mathbf{x}_b], \tag{2.3}$$

where this time, $\mathbf{H}$ is evaluated at $\mathbf{x} = \mathbf{x}_b$. Inserting this expression into Eq. (1) and differentiating with respect to $\mathbf{x}_a$ yields Eq.(2.2).

For some purposes, it is useful to take the second derivative of J with respect to $\mathbf{x}_a$. The second derivative of a scalar with respect to a vector yields an I×I matrix known as the Hessian matrix and corresponds to a measure of the curvature. Thus, differentiating the gradient in Eq. (2.2) with respect to $\mathbf{x}_a$ (and ignoring small terms due to the nonlinearity of H) yields the Hessian matrix $\nabla^2 J = \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{P}_b^{-1}$. We note that if the extremum of J is to be a minimum, then the Hessian should be positive-definite. This should be the case if $\mathbf{R}$ and $\mathbf{P}_b$ are positive-definite.

At the minimum of J,$\nabla J = 0$ (and the Hessian is positive-definite). Setting $\nabla J = 0$ in Eq. (2.2) yields (after adding and subtracting $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}[\mathbf{x}_a - \mathbf{x}_b]$),

$$[\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{P}_b^{-1}][\mathbf{x}_a - \mathbf{x}_b] = \mathbf{H}^T\mathbf{R}^{-1}[\mathbf{y} - H(\mathbf{x}_a) + \mathbf{H}[\mathbf{x}_a - \mathbf{x}_b]]. \tag{2.4}$$

Application of the approximation (2.3) gives,

$$\mathbf{x}_a - \mathbf{x}_b = [\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{P}_b^{-1}]^{-1}\mathbf{H}^T\mathbf{R}^{-1}[\mathbf{y} - H(\mathbf{x}_b)]. \tag{2.5}$$

We refer to $\mathbf{y} - H(\mathbf{x}_b)$ as the innovation vector (in observation space). $\mathbf{x}_a - \mathbf{x}_b$ is referred to as the correction vector, and $\mathbf{y} - H(\mathbf{x}_a)$ is the residual vector.

We refer to Eq. (2.5) as the information or analysis space form of the solution. If $\mathbf{A}$ is an error covariance matrix and $\|\mathbf{A}\|$ is some norm of $\mathbf{A}$, then as $\|\mathbf{A}\|$ decreases the error decreases but $\|\mathbf{A}^{-1}\|$ increases. When the error is small, the information content is large, thus we refer to $\mathbf{A}^{-1}$ as an information matrix. (Note there are other definitions of information, but this is the nomenclature used here). Thus, as the error covariances $\mathbf{R}$ and $\mathbf{P}_b$ appear in Eq. (2.5) in inverse form, we refer to Eq. (2.5) as the information form.

Another useful form of Eq. (2.5), we refer to as the error or observation space form. This form can be obtained from (2.5) by application of the Sherman-Morrison-Woodbury formula,

$$\mathbf{x}_a - \mathbf{x}_b = \mathbf{P}_b\mathbf{H}^T[\mathbf{H}\mathbf{P}_b\mathbf{H}^T + \mathbf{R}]^{-1}[\mathbf{y} - H(\mathbf{x}_b)]. \tag{2.6}$$

The error or observation space form is closely related to the optimal interpolation (OI) algorithm. OI does not explicitly include the forward model. However, suppose the observed variables $\mathbf{y}$ and the grid variables $\mathbf{x}$ are the same variables and the forward model is simply spatial interpolation. Then, H is linear ($\mathbf{H} = H$), and we can approximate $\mathbf{H}\mathbf{P}_b\mathbf{H}^T$ by the forecast error covariance directly between the observation locations, and $\mathbf{P}_b\mathbf{H}^T$ directly by the forecast error covariance between observation locations and grid locations. $H(\mathbf{x}_b)$ is simply the background interpolated to the observation locations. In practice, of course, additional assumptions are made in deriving operational OI algorithms-such as allowing only a limited number of observations to influence each gridpoint. The advantages of (2.5) or (2.6) over operational OI algorithms are:

(1) all observations influence the analysis at every gridpoint; and

(2) more general forward models can be used, and thus observations that are not related directly to analyzed variables can be more easily assimilated.

Before proceeding, we note that either (2.5) or (2.6) can be written in the form,

$$\mathbf{x}_a - \mathbf{x}_b = \mathbf{K}[\mathbf{y} - \mathrm{H}(\mathbf{x}_b)], \tag{2.7}$$

where $\mathbf{K}$ is an IxL rectangular weight or gain matrix. It is straightforward to show that if the analysis error vector is defined as $\mathbf{e}_a = \mathbf{x}_a - \mathbf{x}$ and the analysis error covariance as $\mathbf{P}_a = \langle \mathbf{e}_a(\mathbf{e}_a)^T \rangle$, then the analysis error covariance for any choice of weights $\mathbf{K}$ is

$$\mathbf{P}_a = [\mathbf{I} - \mathbf{KH}]\mathbf{P}_b[\mathbf{I} - \mathbf{KH}]^T + \mathbf{KRK}^T, \tag{2.8}$$

where $\mathbf{I}$ is the identity matrix. If the weights are optimal (i.e., minimize the cost function J), then they will be given by Eq. (2.5) or (2.6), that is $\mathbf{K} = [\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{P}_b]^{-1}\mathbf{H}^T\mathbf{R}^{-1} = \mathbf{P}_b\mathbf{H}^T[\mathbf{HP}_b\mathbf{H}^T + \mathbf{R}]^{-1}$. Inserting these values into Eq. (2.7) yields the analysis error covariance under optimal conditions

$$\mathbf{P}_a = [\mathbf{I} - \mathbf{KH}]\mathbf{P}_b, \tag{2.9}$$

that can also be written

$$\mathbf{P}_a^{-1} = \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{P}_b^{-1}. \tag{2.10}$$

Note the resemblance between the inverse of the analysis error covariance and the Hessian matrix. It is simple to see from Eq. (2.10) that since $\mathbf{R}$ and $\mathbf{P}_b$ (and their inverses) are positive-definite, the information content of the analysis must be greater than the information content in either the background or the observations.

2.2 Three Forms of the 3DVAR Algorithm

We now discuss three forms of the 3DVAR algorithm. Most operational 3DVAR algorithms can be described by one of these forms. Two of these are information or analysis space algorithms (Methods A and B) and the third (Method C) is an error or observation space algorithm.

2.2.1 Method A (Analysis Grid/Physical Space)

Let us write Eq. (2.5) in the form

$$\mathbf{x}_a - \mathbf{x}_b = \mathbf{Q}_A^{-1}\mathbf{H}^{T-1}[\mathbf{y} - \mathrm{H}(\mathbf{x}_b)], \tag{2.11}$$

where $\mathbf{Q}_A = [\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{P}_b^{-1}]$ is the Hessian matrix of J and the optimal analysis error covariance. In essence, the minimization of J is equivalent to the inversion of $\mathbf{Q}_A$ (actually the solution of the linear problem involving the matrix $\mathbf{Q}_A$). This form has advantages and disadvantages. The principal work done in implementing this algorithm is the solution of this large linear problem. This linear problem involves an IxI matrix and the solution is performed in the physical space of the analysis gridpoints. For a global model, I might be very large O(10^6–10^7). However, the size of the problem does not depend on the number of observations. Obviously, this method is advantageous when there are many observations and few gridpoints. Another, related advantage occurs because information is additive, while error is not. This is illustrated using a scalar version of Eq. (2.11). Suppose we have a background x_b with error variance ε_b^2 and two observations-x_1 with error variance ε_1^2 and x_2 with error variance ε_2^2. Assume that the observations are not correlated either with each other or the background, and that the forward model is simply equal to one. Then, if we use observation x_1 only, the appropriate form of Eq. (2.11) is

$$x_a = x_b + \{\varepsilon_1^{-2}(x_1 - x_b)\}/\{\varepsilon_1^{-2} + \varepsilon_b^{-2}\}, \tag{2.12a}$$

whereas, if both x_1 and x_2 are used, Eq. (2.11) becomes

$$x_a = x_b + \{\varepsilon_1^{-2}(x_1 - x_b) + \varepsilon_2^{-2}(x_2 - x_b)\} / \{\varepsilon_1^{-2} + \varepsilon_2^{-2} + \varepsilon_b^{-2}\}. \qquad (2.12b)$$

Adding more observations (as in Eq. 2.12b) means that in the denominator, we are simply adding information (inverse of error). This property is particularly attractive for adding new (nongeophysical) observation types.

This additive property of Method A has another advantage. Suppose we wished to impose a constraint on the cost function J (Eq. (2.1)). That is, we require that the analysis satisfy some constraint $g(\mathbf{x}_a) = 0$, which could represent geostrophy, the linear balance equation, no generation of fast gravity modes, etc. If the constraint were to be applied exactly, it would be a strong constraint, and if applied approximately, it would be a weak constraint. It is relatively easy to add such a constraint to (2.1). If the constraint is weak, then we add a term $\beta^{-1}g(\mathbf{x}_a)$ to (2.1). As β decreases, the constraint is applied more strongly. For strong constraints, we use the method of Lagrange multipliers and introduce a new cost function $J_1 = J + \lambda g(\mathbf{x}_a)$, where λ is the undetermined Lagrange multiplier.

Now if the constraints are strong and linear, then there are philosophical objections to imposing them in this way. That is, if the analysis is to reflect such a constraint, then presumably it should be reflected in the background error statistics in $\mathbf{P}_b$. If the background error statistics already reflect this constraint, then nothing is to be gained by adding it. On the other hand, if the background error statistics do not reflect this constraint, then the external imposition of the constraint is inconsistent with the background error statistics.

The four-dimensional extension of this algorithm is the strong constraint 4DVAR algorithm.

This method was used by Derber and Rosati (1989) and extended by Passi et al. (1993) for ocean data assimilation. Solution is obtained by a preconditioned conjugate gradient method (see Section 3). Although we discuss preconditioning in more detail in Section 3, we note that if we wish to solve $\mathbf{Ap} = \mathbf{r}$ for $\mathbf{p}$ and there is a matrix $\mathbf{A}^*$ that is reasonably similar to $\mathbf{A}$ and for which we easily solve $\mathbf{A}^*\mathbf{p} = \mathbf{r}$, then $\mathbf{A}^*$ can be used as a preconditioner for $\mathbf{A}$. The availability of simple and reasonably accurate preconditioners makes the solution of equations like (2.11) much simpler. In the case of Eq. (2.11), a good choice of preconditioner is the matrix $\mathbf{P}_b^{-1}$, which is an approximation to $\mathbf{Q}_A$, and its inverse $\mathbf{P}_b$ is known.

More recently a variant of this method has been applied to the atmospheric mesoscale assimilation problem (Derber et al., 1996). Gaussian correlations are horizontally separable and multiplication by a univariate Gaussian background error covariance matrix on a separable grid can be simulated exactly by the application of recursive digital filters in each of the two horizontal directions. This is, of course, very efficient. Extension to the multivariate case is more difficult, and the operational solution has been to add weak geostrophic or other constraints while keeping the background error covariances univariate. However, it is clearly inconsistent to specify that the background error have no multivariate correlations while permitting the resulting correction fields to be correlated.

2.2.2 Method B (Analysis Grid/Semi-modal Space)

Method B is a variant of Method A, which is sometimes referred to as the spectral or semi-modal method. The analysis is still performed in the analysis grid space, but it uses a spectral (modal) decomposition and spectral transforms. Define $\mathbf{E}_b$ as the I×I matrix of eigenvectors of the background error covariance $\mathbf{P}_b$. Define Λ_b as the diagonal I×I matrix of the corresponding (positive, real) eigenvalues. Then, because $\mathbf{P}_b$ is symmetric, positive-definite, we have

$$\mathbf{P}_b = \mathbf{E}_b\Lambda_b\mathbf{E}_b^T \text{ and } \mathbf{E}_b\mathbf{E}_b^T = \mathbf{I}. \qquad (2.13)$$

Substitution of (2.13) into (2.11) yields

$$\mathbf{x}_a - \mathbf{x}_b = \mathbf{E}_b\Lambda_b{}^{1/2}\mathbf{Q}_b{}^{-1}\Lambda_b{}^{1/2}\mathbf{E}_b{}^{T}\mathbf{H}^{T}\mathbf{R}^{-1}[\mathbf{y} - \mathrm{H}(\mathbf{x}_b)], \tag{2.14}$$

where $\mathbf{Q}_b = \mathbf{I} + \mathrm{L}_b{}^{1/2}\mathbf{E}_b{}^{T}\mathbf{H}^{T}\mathbf{R}^{-1}\mathbf{H}\mathbf{E}_b\mathrm{L}_b{}^{1/2}$.

Note that $\mathbf{Q}_b$ is symmetric. Method B has most of the advantages of Method A-observations are easily added, constraints can be imposed, etc. The difficulty with the information form (2.5) is that it involves manipulation of the (generally) huge background error covariance matrix $\mathbf{P}_b$. Method B attempts to deal with this problem by manipulating the diagonal matrix Λ_b instead of the full matrix $\mathbf{P}_b$. Method B cannot be used for an arbitrary background error covariance matrix, because determining the full eigenstructure of an arbitrary $\mathbf{P}_b$ matrix would be extremely costly. What is actually done is to use a very simple background error covariance, in which a representation like (2.13) can be defined using known spectral functions such as spherical harmonics or Hough functions. For example, for the univariate case on the sphere, if the background error covariance is assumed to be isotropic and homogeneous, then a form like (2.13) is possible for a triangularly truncated spherical harmonic expansion. The multivariate case can be handled by using a Hough function expansion (Heckley et al., 1992), or by application of the linear balance equation (Parrish and Derber, 1992, or Daley, 1996). In this case, the matrices $\mathbf{E}_b$ and $\mathbf{E}_b{}^T$ are simply spectral transform matrices, which are widely used in global spectral models. Note that while the horizontal is handled spectrally, the vertical is still handled in physical space, hence the term "semi-modal."

Method B is used operationally (with spherical harmonics or Hough functions) at NCEP, ECMWF, CMC, and Meteo France. It has the primary disadvantage of Method A in that, for the most part, the specification of the background error covariance (which is absolutely crucial) is in some ways less sophisticated than in OI.

2.2.3 Method C (Observation/Physical Space)

Method C uses the error form (2.6) and can be written

$$\mathbf{x}_a - \mathbf{x}_b = \mathbf{P}_b\mathbf{H}^{T}\mathbf{Q}_C{}^{-1}[\mathrm{y} - \mathrm{H}(\mathbf{x}_b)], \text{ with } \mathbf{Q}_C = \mathbf{H}\mathbf{P}_b\mathbf{H}^{T} + \mathbf{R}. \tag{2.15}$$

We can also obtain this algorithm by minimum variance estimation. It bears some similarity to the OI algorithm.

In this algorithm, the solution to the linear problem involving $\mathbf{Q}_c$ is performed in the space of the observations. This LxL space may be smaller than the IxI space required in Method A. The technique has been used at DAO (NASA/Goddard) where it is referred to as PSAS (Physical Space Assimilation System) (see Cohn et al.,1998). While the acronym is snappy, it is a bit of a misnomer; Method C is really an observation space assimilation system, although performed entirely in physical space. It is fairly obvious that the cost of the algorithm increases as the number of observations increase. For situations in which there are few observations (which is quite likely to occur in many Navy applications), this aspect of the algorithm might be quite attractive.

The four-dimensional extension of this algorithm is the Kalman filter or the representer algorithm.

Method C does not have the nice additive properties of Methods A and B. This makes it more difficult to add constraints, particularly strong constraints. However, weak constraints can be added simply by treating the constraints as extra observations and augmenting the observation vector and observation error covariance matrices.

There is also a semi-modal variant of Method C, which is discussed in Section 4.4. In this variant, the horizontal operations are handled in physical space as before while the vertical operations are handled in a decomposed eigenvector space. This variant is very important for the formulation of NAVDAS, but we do not discuss it further until Chapter 4.

It should be noted that all three 3DVAR algorithms are equivalent in principle (see Courtier, 1997). Where they will differ is in the approximations required to make them tractable, and these differing approximations will lead to differing analyses. All of these algorithms involve solving large linear problems, by iterative or descent methods. Thus, we next examine the condition numbers of the matrices $\mathbf{Q}_A$, $\mathbf{Q}_B$, and $\mathbf{Q}_C$.

2.3 Condition Numbers for the Three Methods

Solving these large linear problems by iteration is expensive, and it is clearly desirable that they converge in as few iterations as possible. The convergence rate will depend largely on the condition numbers of these matrices. The larger the condition number, the slower the rate of convergence. There are several ways to define condition number, but we define it as the ratio of the maximum and minimum eigenvalues (assuming they are all non-negative). Thus, if the eigenvalues are all similar, convergence will be quick, but if some of the eigenvalues are very small, there will be problems. We first examine a case where the condition numbers can be obtained analytically. Assume that (1) forecast and observed variables are the same, (2) the observation network coincides with the analysis grid, and (3) $\mathbf{R}$ and $\mathbf{P}_b$ commute, that is $\mathbf{RP}_b = \mathbf{P}_b\mathbf{R}$. Under these conditions, $\mathbf{H} = \mathbf{I}$ and the eigenvectors (but not necessarily the eigenvalues) of $\mathbf{R}$ and $\mathbf{P}_b$ are the same. We can then write, $\mathbf{R} = \mathbf{E}\Lambda_r\mathbf{E}^T$ and $\mathbf{P}_b = \mathbf{E}\Lambda_b\mathbf{E}^T$, where $\mathbf{E}$ is the common eigenvector matrix and Λ_r and Λ_b are the diagonal matrices of eigenvalues of $\mathbf{R}$ and $\mathbf{P}_b$ respectively. Substitution into the definition of the $\mathbf{Q}$ matrices (Eqs. (2.11), (2.14), and (2.15)) gives

$$\mathbf{Q}_A = \mathbf{E}[\Lambda_b^{-1} + \Lambda_r^{-1}]\mathbf{E}^T,\ \mathbf{Q}_B = \mathbf{E}[\mathbf{I} + \Lambda_b\Lambda_r^{-1}]\mathbf{E}^T,\ \mathbf{Q}_C = \mathbf{E}[\Lambda_b + \Lambda_r]\mathbf{E}^T. \tag{2.16}$$

Denote individual eigenvalues of $\mathbf{P}_b$ and $\mathbf{R}$ as λ_b and λ_r, respectively, and denote the maximum and minimum eigenvalues of $\mathbf{P}_b$ as λ_b^{max} and λ_b^{min}. Denote the condition number of $\mathbf{Q}_A$ as c_A and define it as the largest eigenvalue of $\mathbf{Q}_A$ divided by the smallest. Similarly define c_B and c_C. We consider two scenarios–uncorrelated observation error and correlated observation error.

Scenario 1 – uncorrelated observation error (white noise) λ_r = constant

$$c_A = \lambda_b^{max}[\lambda_b^{min} + \lambda_r]/\lambda_b^{min}[\lambda_b^{max} + \lambda_r],\ c_B = c_C = [\lambda_b^{max} + \lambda_r]/[\lambda_b^{min} + \lambda_r]. \tag{2.17}$$

To explore these condition numbers, let us consider four limiting cases:

(1) uncorrelated background error ($\lambda_b^{max} = \lambda_b^{min}$) $c_A = c_B = c_C = 1$.

(2) red background error ($\lambda_b^{max} >> \lambda_r >> \lambda_b^{min}$) $c_A = \lambda_r / \lambda_b^{min} >> 1$, $c_B = c_C = \lambda_b^{max} / \lambda_r >> 1$.

(3) accurate background ($\lambda_r >> \lambda_b^{max}$) $c_A = \lambda_b^{max} / \lambda_b^{min}$, $c_B = c_C = 1$.

(4) accurate observations ($\lambda_b^{min} >> \lambda_r$) $c_A = 1$, $c_B = c_C = \lambda_b^{max} / \lambda_b^{min}$.

Thus, for uncorrelated observation error, we may conclude that for all three methods, the condition number increases as the background error becomes redder (increasing correlation length). For Method A, the condition number increases as the background becomes more accurate, with the converse being true for Methods B and C.

Scenario 2 – correlated observation error (same spectrum as forecast error) $\lambda_r = \lambda_b$.

$$c_A = c_C = \lambda_b^{max} / \lambda_b^{min},\quad c_B = 1. \tag{2.18}$$

Clearly, the effect of correlated observation error is to increase the condition number for Methods A and C, but to decrease it for Method B.

The assumptions used in deriving (2.16-18) are very restricted, so we will now discuss a more realistic scenario. Consider the univariate two-dimensional case with an observation network in which the observations are randomly located. We will assume that both the background error and observation error correlations take the form of a second order autoregressive function (SOAR), but that the error variances and the correlation lengths are different.

$$\text{Background error covariance: } \varepsilon_b^2[1 + s/L_b]\exp(-s/L_b), \tag{2.19}$$

$$\text{Observation error covariance: } \varepsilon_r^2[1 + s/L_r]\exp(-s/L_r), \tag{2.20}$$

Uncorrelated observation error is the limit as $L_r \rightarrow 0$. s is the absolute distance between any two observation locations or any two gridpoint. The forward operator H is bilinear interpolation. The grid was 9×9 equally spaced, and there are 81 (randomly spaced) observations. The domain is doubly periodic $-\pi \leq x,y \leq \pi$. As before, we define the condition number as the ratio of the maximum and minimum eigenvalues of the **Q** matrices, determined numerically using standard eigenvector decomposition software. We compare the condition numbers obtained with this random network, with the condition numbers determined analytically (from Eqs. (2.16)-(2.18)) for a coincident network/grid under the same experimental conditions.

We consider four cases.

Case 1: $\varepsilon_r = \varepsilon_b = 1, L_r = 0, L_b = 1/3$ *weakly correlated background error uncorrelated observation error*

$\lambda_r = 1, \lambda_b^{max} = 3.9, \lambda_b^{min} = 0.27$

Analytic Random Observation Network

$c_A = 3.7$ $c_B = 3.9$ $c_C = 3.9$ $c_A = 5.5$ $c_B = 6.1$ $c_C = 6.1$

Case 2: $\varepsilon_r = \varepsilon_b = 1, L_r = 0, L_b = 1$ *strongly correlated background error uncorrelated observation error*

$\lambda_r = 1, \lambda_b^{max} = 22.3, \lambda_b^{min} = 0.016$

Analytic Random Observation Network

$c_A = 62$ $c_B = 23$ $c_C = 23$ $c_A = 87$ $c_B = 25$ $c_C = 25$

Case 3: $\varepsilon_r = 0.1, \varepsilon_b, L_r, L_b$ as in Case 1 *accurate, uncorrelated observations weakly correlated background error*

$\lambda_r = 0.01, \lambda_b^{max} = 3.96, \lambda_b^{min} = 0.27$

Analytic Random Observation Network

$c_A = 1.03$ $c_B = 14.2$ $c_C = 14.2$ $c_A = 212$ $c_B = 511$ $c_C = 511$

Case 4: same as Case 1, except with varying number of (random) observations

10 observations	81 observations	300 observations
$c_A = 12$ $c_B = 2.2$ $c_C = 1.6$	$c_A = 5.5$, $c_B = 6.1$ $c_C = 6.1$	$c_A = 5.5$ $c_B = 18.8$ $c_C = 58$

Note that for Case 4 the domain remained constant, so it was actually the observation density that increased. Another experiment (not shown here) considered the case when the number of observations was held constant, but the number of gridpoints was allowed to vary. In this case, c_A increased strongly as the number of gridpoints was increased; c_C was essentially invariant.

From these analytical and experimental results, it is possible to draw several conclusions.

The experimental results with a random network are qualitatively consistent with the analytic results. In all cases, however, the condition numbers from the more realistic networks were higher. All methods have increasing condition number as the background error correlation length increases-this effect appears to be most marked for Method A.

For Methods B and C, condition number becomes large for accurate observations, while for Method A, it becomes large for an accurate background. Correlated observation error increases the condition number for Methods A and C and decreases it for Method B.

For Method C, condition number is sensitive to the observation density but not to the grid resolution, while for Method A, the reverse is true.

2.4 The Way Forward (the NAVDAS Algorithm)

Methods A and B were the earliest (first generation) 3DVAR algorithms and have been used successfully by NCEP, ECMWF, and others. Method C is a later algorithm; although a variant has already been implemented at NASA/Goddard by Cohn et al. (1998).

Our investigations have shown that all methods have advantages and disadvantages. However, a choice had to be made. The choice was Method C, and the choice was made for the following reasons:

(1) The primary drawback of Method A, and to a lesser extent Method B, is that cruder approximations must be made to the background error statistics than in OI. For Method C, the background error statistics could, in principle, be more sophisticated than in OI.

(2) Naval data assimilation problems are more likely to suffer from a scarcity of observations than from an overabundance. Since Method C is an observation-based algorithm, it is more suitable for this environment.

(3) Of all the methods, Method C is most like OI. As we see in the next section, an implementation of Method C that uses a preconditioned conjugate-gradient descent algorithm, in which the preconditioner requires the observations to be sorted into observation volumes, has some similarities to the volume technique already used in the NRL MVOI algorithm. In fact, some of the NRL MVOI software can be adapted to NAVDAS purposes.

(4) Method B has been successfully implemented for the global problem at NCEP, but NCEP uses a completely different algorithm for the regional problem. On the other hand, an algorithm based on

Method C has little dependence on the forecast model or the analysis grid and could use essentially the same code for both shipboard and global problems. This would imply a much more easily maintainable code, which is a major advantage for a small group.

3. Descent Algorithms

This section discusses the descent algorithm used in the NAVDAS algorithm. We start with three sections of a pedagogical nature, which explore some of the general properties of descent algorithms and also explain the reasons for choosing a preconditioned conjugate gradient descent procedure. The remainder of the section discusses various aspects of the implemented descent algorithm. Appendix E provides additional relevant material.

3.1 General Concepts

There is a large literature on descent algorithms, but the standard text is Gill et al. (1982). The treatment in the present paper is very elementary.

Consider the very simple cost function

$$J = u^2/a^2 + v^2/b^2, \tag{3.1}$$

where "u" and "v" are variables and "a" and "b" are (known) constants. In more general cases, u and v would be functions of the spatial and/or temporal coordinates of the problem. In this instance, however, u and v are simply scalars. The lines of constant J are ellipses in the (u,v) plane, with major and minor axes a and b. Define unit vectors **i** in the u direction and **j** in the v direction. Then, the gradient operator in the (u,v) plane can be written

$$\nabla J = (\partial J/\partial u)\mathbf{i} + (\partial J/\partial v)\mathbf{j}. \tag{3.2}$$

∇J, in this case, is a vector of length 2, with components $\partial J/\partial u = 2u/a^2$ and $\partial J/\partial v = 2v/b^2$. The change in J caused by varying u and v is $\delta J = \nabla J \cdot \delta U$, where $U = u\mathbf{i} + v\mathbf{j}$ and $\delta U = \delta u\mathbf{i} + \delta v\mathbf{j}$.

Consider the point $U_0 = (u_0,v_0)$. If we make a small change in u_0 and v_0 $(\delta u_0,\delta v_0)$, then the change in J is given by

$$\delta J_0 = \nabla_0 J \cdot \delta U_0 = 2u/a^2\, \delta u_0 + 2v/b^2\, \delta v_0. \tag{3.3}$$

The goal is to minimize J. In fact, it is trivial in this case; the minimum value of J occurs at u = v = 0 and is equal to zero. ∇J points to larger values of J, and $-\nabla J$ points to smaller values of J. Clearly, if we start from (u_0, v_0) and proceed in the direction of $-\nabla J$, we will be reducing the value of the cost function J.

Now, suppose we wish to find the minimum of J by iteration, starting at (u_0, v_0). We seek a next iterate (u_1, v_1) that has a smaller value of J than does (u_1, v_1). If we proceed in the direction of $-\nabla J$, the difficulty lies in determining how far to proceed in that direction. Obviously, if we go too far in the direction $-\nabla J$, J will start to increase again, and if we don't go far enough, J will not be reduced very much. The idea is to find the optimal distance (referred to as the *step-length*) to proceed in the direction $-\nabla J$. There are several ways to do this, but the general idea is to proceed along $-\nabla J$ until J reaches a minimum (in that direction).

We now have the ingredients to describe the simplest descent method, the method of steepest descent. In this procedure, we start at some starting point (0), calculate the local gradient at that point, proceed in the direction of the negative of that gradient until a minimum is reached in that direction. At this point (1), the value of the cost function should be less than it was at point (0). At point (1), we calculate the local gradient and proceed as before. We keep on repeating the procedure until the gradient becomes vanishingly small or the cost function does not diminish any further. The procedure is illustrated in Fig. 3.1 (after Walsh, 1975). The steepest descent procedure always proceeds down the local gradient. It takes many iterations to converge to the minimum, because the local gradient is generally **not** pointing toward the actual minimum of the cost function.

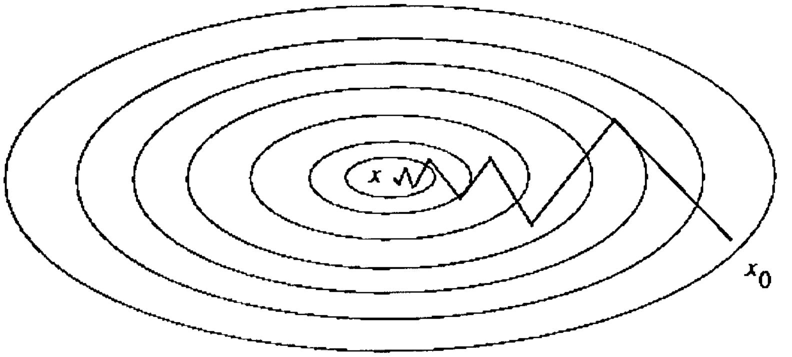


Figure 3.1
Schematic showing the iteration path for the method of steepest descent

Before proceeding, we make a few general statements about cost functions. Cost functions that are quadratic (such as the simple example above) have only one minima and lead to linear problems. Consider a slightly more complicated example, $J = u^2 + v^2 + (au + bv - c)^2$, a, b, and c known positive constants. Then, $\partial J/\partial u = 2[(1 + a^2) u + abv - ac]$ and $\partial J/\partial v = 2[(1 + b^2)v + abu - bc]$. The minimum is found by setting the two components of the gradient vector equal to zero. This is equivalent to solving the 2×2 matrix equation,

$$\begin{bmatrix} 1+a^2 & ab \\ ab & 1+b^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} ac \\ bc \end{bmatrix}$$

For quadratic cost functions, there is always an equivalent matrix form.

Examination of Fig. 3.1 suggests that, for an elliptic cost function, the local gradient rarely points at the minimum, and this means the method of steepest descent usually takes many iterations to converge. In fact, in Fig. 3.1, if we draw a line between x_0 and the minimum, the steepest descent path keeps crossing back and forth across this line. Ironically, proceeding down the fastest local direction is usually a very slow way to get to the minimum. Suppose the cost function in Fig. 3.1 were circular instead of elliptic, then the negative of the gradient would always point at the minimum. Obviously, we could reach convergence very quickly in this case. Is it possible to modify the cost function so that it becomes circular? Consider again the cost function of (3.1). Suppose we introduce new variables $x = u/a$ and $y = v/b$, then $J = x^2 + y^2$. On the (x,y) plane, J is circular, and $-\nabla J$ points directly at the minimum. The minimum is still at $u = v = 0$ or $x = y = 0$, but now $-\nabla J$ points directly at the minimum. Let us examine this transformation from the point of view of the condition number. The equivalent matrix form for cost function (3.1) is

$$\begin{bmatrix} a^{-2} & 0 \\ 0 & b^{-2} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

The eigenvalues of this matrix are a^{-2} and b^{-2}. Let us suppose that the cost function (3.1) is highly elliptic, say, $b^2 >> a^2$. Now make the substitution $x = u/a$, $y = v/b$ as above. This time, both eigenvalues are equal to 1. The condition number (ratio of the eigenvalues) for the transformed problem is now 1, while the condition number of the original problem (3.2) was $b^2/a^2 >> 1$. Thus, a transformation that makes the cost function more circular also improves the condition number.

This type of transformation is referred to as preconditioning. In essence, one multiplies the matrix to be inverted by its approximate inverse to create a new matrix, which is as close to the identity matrix as possible. (This was trivial for cost function (3.1)). A simple preconditioning matrix might consist of the inverse of the diagonal elements of the matrix to be inverted. However, one can determine much better (but more complex) preconditioners than that. If $\mathbf{Ax} = \mathbf{b}$, and $\mathbf{S} \approx \mathbf{A}^{-1}$, then $\mathbf{SAx} = \mathbf{Sb}$ and $\mathbf{SA}$ has a lower condition number than $\mathbf{A}$.

There is another way to achieve the same result. Suppose, instead of proceeding down the direction of steepest descent ($-\nabla J$), one were to define a new descent direction $-\theta\nabla J$. θ is a matrix that multiplies the vector ∇J to produce a new vector (and a different direction). If θ were the identity matrix; then the descent direction would be the steepest descent. Now define θ in the following way: $\theta^{-1} = \nabla^2 J$, which is the Hessian matrix defined after (2.3). The descent direction $-[\nabla^2 J]^{-1}\nabla J$ has a remarkable property for quadratic cost functions–it converges to the minimum in a single iteration. This type of descent is known as a Newton descent because it bears some resemblance to Newton's method in root finding. This method works very well for small problems, where the Hessian can be found. For large problems, the Hessian may be difficult to obtain or too large to store (it may be a huge matrix). However, there are quasi-Newton methods that attempt to circumvent these problems.

3.2 Formulation of Four Descent Algorithms for Observation Space Algorithms

We now consider four descent algorithms for application to Method C. That is, we wish to solve the problem

$$\mathbf{x}_a - \mathbf{x}_b = \mathbf{P}_b\mathbf{H}^T\ [\mathbf{HP}_b\mathbf{H}^T + \mathbf{R}]^{-1}[\mathbf{y} - H(\mathbf{x}_b)]. \tag{3.4}$$

We can break this problem into two steps,

3.2.1 Solve the system

$$[\mathbf{HP}_b\mathbf{H}^T + \mathbf{R}]\mathbf{z} = \mathbf{d}, \tag{3.5}$$

where the vector $\mathbf{d} = \mathbf{y} - H(\mathbf{x}_b)$ and the vector $\mathbf{z}$ is to be determined.

3.2.2 Post-multiplication step

$$\mathbf{x}_a - \mathbf{x}_b = \mathbf{P}_b\mathbf{H}^T\mathbf{z}. \tag{3.6}$$

Note that in (3.5), we merely wish to solve the linear system, **not** invert the matrix. It is easy to show that the vector **z** that solves (3.5) also minimizes

$$I(\mathbf{z}) = 0.5\mathbf{z}^T[\mathbf{HP}_b\mathbf{H}^T + \mathbf{R}]\mathbf{z} - \mathbf{z}^T\mathbf{d}. \tag{3.7}$$

The gradient of (3.7) is $\nabla I = [\mathbf{HP}_b\mathbf{H}^T + \mathbf{R}]\mathbf{z} - \mathbf{d}$, which at zero yields (3.5).

The major problem is (3.5), solving the linear system $\mathbf{Az} = \mathbf{d}$, where $\mathbf{A} = [\mathbf{HP}_b\mathbf{H}^T + \mathbf{R}]$. Let us define a general descent algorithm or iterative procedure as

$$\mathbf{z}_{k+1} = \mathbf{z}_k + \alpha_k\mathbf{p}_k, \tag{3.8}$$

where k is the iteration number, $\mathbf{z}_k$ is the kth iterate, α_k (scalar) is the step-length at the kth iterate, and $\mathbf{p}_k$ is a vector defining some search direction. We can write $\mathbf{p}_k = \theta_k\nabla_k I$, where$\nabla_k I$ is the gradient of the cost function (3.7) at the kth iterate, and $\mathbf{q}_k$ is a matrix that multiplies the gradient vector, so that the actual search direction $\mathbf{p}_k$

may be different than the steepest descent direction $\nabla_k I$. Let us define the kth residual

$$\mathbf{r}_k = \mathbf{d} - \mathbf{A}\mathbf{z}_k = \nabla_k I. \tag{3.9}$$

Then, it can be shown that to minimize I along the direction $\mathbf{p}_k$ with respect to α_k, the step-length should be chosen as

$$\alpha_k = \mathbf{p}_k{}^T\mathbf{r}_{k-1}/\mathbf{p}_k{}^T\,\mathbf{A}\mathbf{p}_k. \tag{3.10}$$

3.2.3 The method of steepest descent

In this algorithm, the search occurs along the gradient direction $\nabla_k I$. Our first estimate is $\mathbf{z}_0 = 0$, and thus $\mathbf{r}_0 = \mathbf{d}$. Then, for k > 0, $\mathbf{p}_k = \mathbf{r}_{k-1}$ followed by

$$\begin{aligned}
&\mathbf{q}_k = \mathbf{A}\mathbf{p}_k,\\
&\alpha_k = \mathbf{p}_k{}^T\mathbf{r}_{k-1} / \mathbf{p}_k{}^T\mathbf{q}_k = \mathbf{r}_{k-1}{}^T\mathbf{r}_{k-1} / \mathbf{r}_{k-1}{}^T\mathbf{q}_k,\\
&\mathbf{z}_k = \mathbf{z}_{k-1} + \alpha_k\mathbf{p}_k = \mathbf{z}_{k-1} + \alpha_k\mathbf{r}_{k-1.}\\
&\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k\mathbf{q}_k\,.
\end{aligned} \tag{3.11}$$

This process is continued until we are satisfied that the residual $\mathbf{r}_k = \nabla_k I$, for some k, has become sufficiently small.

3.2.4 The quasi-Newton (BFGS) algorithm

This algorithm attempts to construct increasingly more accurate approximations to the Hessian. The algorithm used here is the Broyden, Fletcher, Goldfarb, Shanno (BFGS) algorithm, which is discussed in Tarantola (1987) and is not described further here.

3.2.5 The standard conjugate gradient algorithm

As noted earlier, the steepest descent algorithm does not converge very quickly. One way of accelerating convergence is to make sure that we always travel in a direction perpendicular to the direction already traveled. Following Golub and van Loan (1990), the conjugate gradient algorithm has this property. As before, at k = 0, we set $\mathbf{z}_0 = 0$, and thus $\mathbf{r}_0 = \mathbf{d}$. Then, for k = 1, $\mathbf{p}_1 = \mathbf{r}_0$.

For k >1, define the scalar $\beta_k = \mathbf{r}_{k-1}{}^T\mathbf{r}_{k-1} / \mathbf{r}_{k-2}{}^T\mathbf{r}_{k-2}$ and $\mathbf{p}_k = \mathbf{r}_{k-1} + \beta_k\mathbf{p}_{k-1}$.

For all k > 0, we then proceed as follows:

$$\begin{aligned}
&\mathbf{q}_k = \mathbf{A}\mathbf{p}_k,\\
&\alpha_k = \mathbf{p}_k{}^T\mathbf{r}_{k-1}/\mathbf{p}_k{}^T\mathbf{q}_k = \mathbf{r}_{k-1}{}^T\mathbf{r}_{k-1} / \mathbf{p}_k{}^T\mathbf{q}_k, \text{ as } \mathbf{p}_k{}^T\mathbf{p}_{k-1} = 0,\\
&\mathbf{z}_k = \mathbf{z}_{k-1} + \alpha_k\mathbf{p}_k,\\
&\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k\mathbf{q}_k
\end{aligned} \tag{3.12}$$

3.2.6 The preconditioned conjugate gradient algorithm

As noted earlier, the purpose of preconditioning is to lower the condition number of the matrix by using an approximate (but easily calculated) inverse. Suppose **A*** is an approximation to **A**, and we can solve **A*s** = **f** for the unknown vector **s** given the vector **f** fairly easily. Then the preconditioned conjugate gradient algorithm works as follows. As before, at k = 0, set $\mathbf{z}_0 = 0$ and $\mathbf{r}_0 = \mathbf{d}$. Then for all $k \geq 0$, solve

$\mathbf{A}^*\mathbf{s}_k = \mathbf{r}_k$ for $\mathbf{s}_k$.
If $k = 1$, $\mathbf{p}_1 = \mathbf{s}_0$,
If $k > 1$ $\beta_k = \mathbf{r}_{k-1}{}^T\mathbf{s}_{k-1}/\mathbf{r}_{k-2}{}^T\mathbf{s}_{k-2}$ and $\mathbf{p}_k = \mathbf{s}_{k-1} + \beta_k\mathbf{p}_{k-1}$.

Then, the remainder of the algorithm is similar to standard conjugate gradient,

$$\begin{aligned} \mathbf{q}_k &= \mathbf{A}\mathbf{p}_k, \\ \alpha_k &= \mathbf{p}_k{}^T\mathbf{r}_{k-1}/\mathbf{p}_k{}^T\mathbf{q}_k = \mathbf{s}_{k-1}{}^T\mathbf{r}_{k-1}/\mathbf{p}_k{}^T\mathbf{q}_k, \\ \mathbf{z}_k &= \mathbf{z}_{k-1} + \alpha_k\mathbf{p}_k, \\ \mathbf{r}_k &= \mathbf{r}_{k-1} - \alpha_k\mathbf{q}_k \end{aligned} \tag{3.13}$$

The important question here is to define a suitable preconditioning matrix **A***. Cohn et al (1998) have considered the solution of problems such as (3.5) using the preconditioned conjugate gradient descent (3.13). The preconditioners **A*** that they examined were block diagonal approximations to **A**, obtained by dividing the observations up into subgroups and considering only the interactions between members of the subgroup. This idea is really ideal for an implementation of Method C, because the NRL MVOI scheme already divides the observations into volumes and does a separate inversion for each volume.

The procedure would work as follows. The horizontal (latitude/longitude) domain is divided into triangles on the sphere (or any subdomain), which in three dimensions are actually prisms. These triangles (prisms) are large where the observation density is low; and small where the observation density is high. The observations are sorted so that there are approximately the same number of observations (perhaps a few hundred) in each observation prism. This is illustrated in Fig. 3.2. All observations in a radiosonde ascent or a vertical sounding would be placed in the same volume.

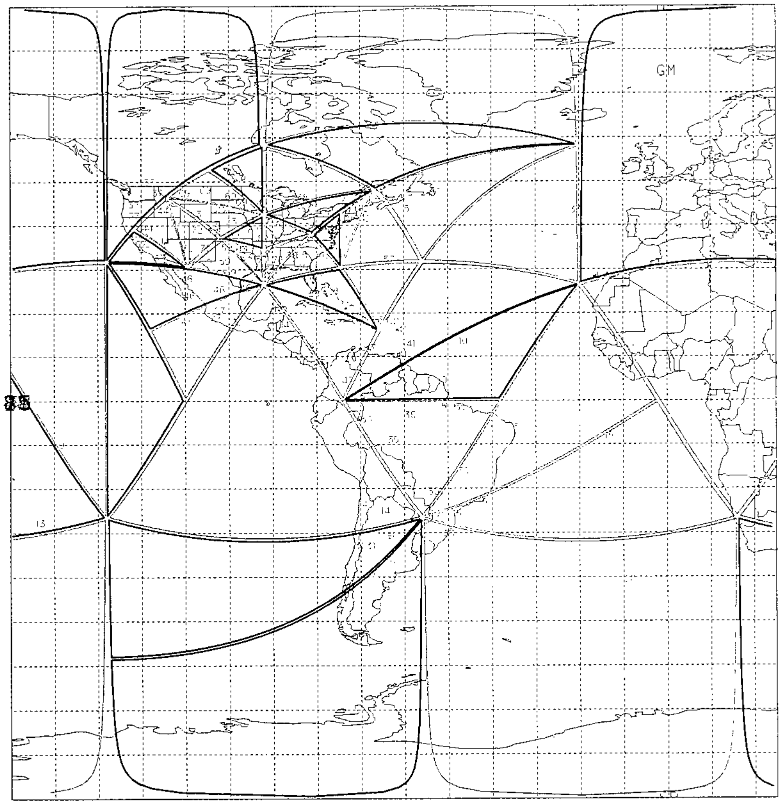
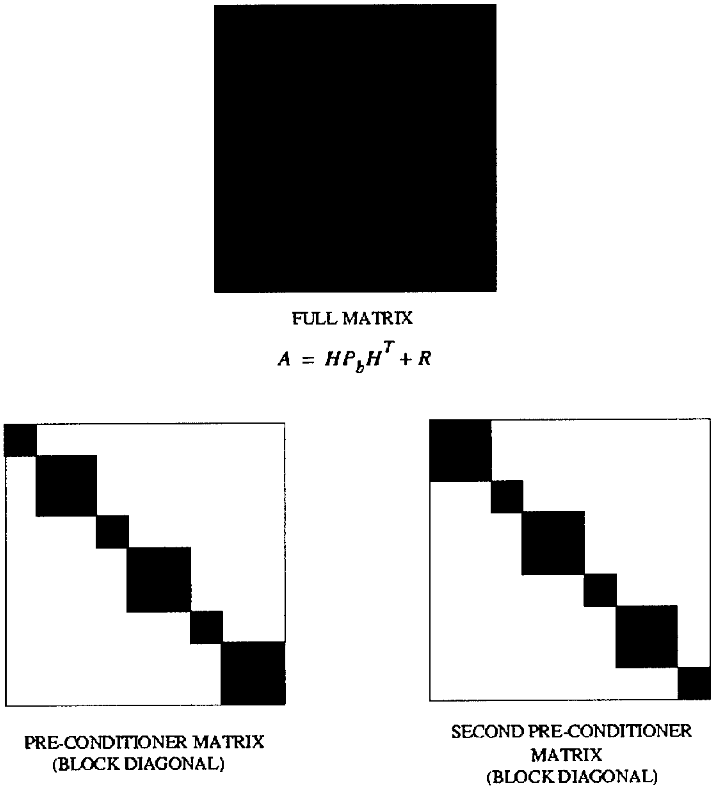


Figure 3.2
Division of the globe into triangular observation prisms

We then consider an approximation to $\mathbf{A} = \mathbf{HP_bH^T} + \mathbf{R}$, which ignores all correlations between observation locations that are not in the same observation prism. That is, all interprism correlations are ignored. This approximate matrix $\mathbf{A}^*$ will then be block-diagonal, as opposed to the original (essentially full) matrix $\mathbf{A}$. This is illustrated in Fig. 3.3, which shows the full matrix $\mathbf{A}$ on the top and two different block-diagonal preconditioner matrices $\mathbf{A}^*$ on the bottom. At this time, we consider only the use of a single preconditioner matrix, leaving the discussion of the second preconditioning matrix until section 3.5. Consider finding $\mathbf{s}$, satisfying $\mathbf{A}^*\mathbf{s} = \mathbf{f}$ with $\mathbf{A}^*$ containing N diagonal blocks $\mathbf{A}_1^* \ldots \mathbf{A}_N^*$. Break up $\mathbf{s}$ into subvectors $\mathbf{s}_1 \ldots \mathbf{s}_N$ and $\mathbf{f}$ into subvectors $\mathbf{f}_1 \ldots\ldots \mathbf{f}_N$. Then,

$$\text{solving } \mathbf{A}^*\mathbf{s} = \mathbf{f} \text{ is equivalent to solving } \mathbf{A}_n^*\mathbf{s}_n = \mathbf{f}_n,\ 1 \leq n \leq N. \tag{3.14}$$

The solution of these smaller problems can be found by direct methods such as Cholesky decomposition or perhaps by iteration using a standard conjugate gradient method. Finally, the vector $\mathbf{s}$ is constructed by piecing together the vectors $\mathbf{s}_1 \ldots \mathbf{s}_N$. In the limit where N = L (the number of observations), the diagonal blocks of $\mathbf{A}^*$ would be of dimension one, and the preconditioning operation would consist of dividing by the diagonal elements of $\mathbf{A}$. At the other limit N = 1 and $\mathbf{A}^* = \mathbf{A}$, which does not get us anywhere. A reasonable choice might be $N = L^{1/2}$, which would give $L^{1/2}$ volumes, each containing $L^{1/2}$ observations. In any case, the observations should be divided up so that there are no volumes that contain substantially more observations than any other volume. If such a volume exists, it should be divided into two smaller volumes.


FULL MATRIX

$A = HP_bH^T + R$

PRE-CONDITIONER MATRIX (BLOCK DIAGONAL)

SECOND PRE-CONDITIONER MATRIX (BLOCK DIAGONAL)

Figure 3.3
Schematic showing the full matrix **A** and two pre-conditioners

It might be noted that there are two major operations in the preconditioned conjugate gradient algorithm (3.14)-the matrix multiplication $\mathbf{q}_k = \mathbf{Ap}_k$ and solving $\mathbf{A}^*\mathbf{s}_k = \mathbf{r}_k$. The matrix multiplication is $O(L^2)$ operations and is expensive. The solve using the block diagonal preconditioner should be somewhat cheaper because it (crudely) involves solving $L^{1/2}$ linear problems of dimension $L^{1/2}$. The remaining operations are either inner products of vectors or multiplication of vectors by scalars, which are very inexpensive.

3.3 One- and Two-Dimensional Univariate Experiments with the Four Descent Algorithms

All of the above descent algorithms were candidates for application to Eq. (3.5). It was decided that existing commercially available descent codes would not be used, but that the descent algorithm would be coded by the developers. This meant that there had to be some rational basis for making a choice. This led to the following series of one and two-dimensional experiments in which all four of the above algorithms were applied.

3.3.1 One-dimensional experiments-choosing the most rapid descent

In this case, 25 gridpoints and 25 observation stations were collocated with the gridpoints on a periodic domain $-\pi \leq x \leq \pi$. Thus, the forward operator **H** was the identity matrix. There was one variable, the geopotential field. It was assumed that the observation errors were not correlated. The background error correlation model was second-order autoregressive (SOAR) (see Eq.(2.20)).

The descent algorithms are designed to solve the linear problems (3.5); they are not designed to produce the analysis weight or gain matrix (2.7). However, for small problems like this one, it is possible to use any descent algorithm to produce the gain matrix using a simple trick. That is, for each observation station in turn, set that observation value equal to 1 and all the other observation values equal to 0. One run of the descent algorithm will then produce one column of the gain matrix **K**. The procedure is then repeated for each of the L observation locations. Obviously, this is very expensive if L is large. However, possession of the gain matrix allows us to calculate the analysis error covariance, using (2.8). We can do this at each iteration step until convergence, when the analysis error covariance will be given by (2.9). At all steps of the iterative procedure before convergence, trace($\mathbf{P}_a$) will always be larger than the converged value.

In the first experiment, the background error correlation length was $L_b = \pi/12$, the background and observation errors were $\varepsilon_b = 1.0$ and $\varepsilon_r = 0.1$. This case corresponds to very accurate observations. The condition number of the matrix $\mathbf{HP}_b\mathbf{H}^T + \mathbf{R}$ was 38. Figure 3.4 shows the rms analysis error on the grid (square root of trace($\mathbf{P}_a$)) as a function of iteration number (abscissa) for three descent algorithms-steepest descent (dash-dot), standard conjugate gradient (solid), and BFGS (dashed). Consistent with Fig. 3.1, the steepest descent converges slowly and seems to oscillate around the solution.

Figure 3.5 shows the analysis error spectrum (obtained from $\mathbf{P}_a$ by pre- and post-multiplication by Fourier matrices) at iteration 3, for the steepest descent (dash-dot) and standard conjugate gradient algorithms (solid) for the case illustrated in Fig. 3.4. The abscissa is spatial wavenumber, and the ordinate is the square root of the spectral analysis error variance. The steepest descent seems to converge very slowly in the long waves, perhaps a consequence of using the local gradient to define the descent direction.

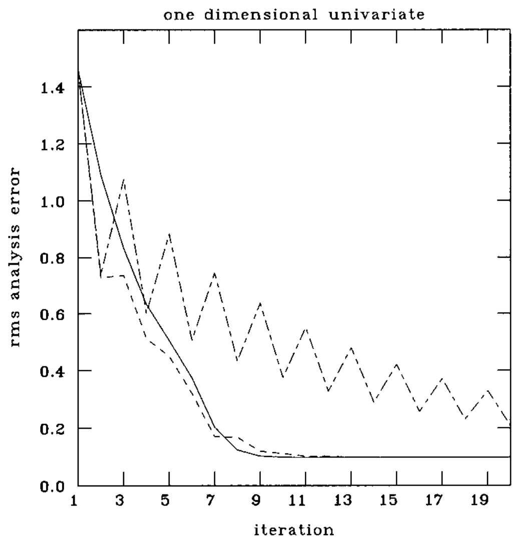


Figure 3.4
Rms analysis grid error (trace **Pa**) for 3 descent algorithms

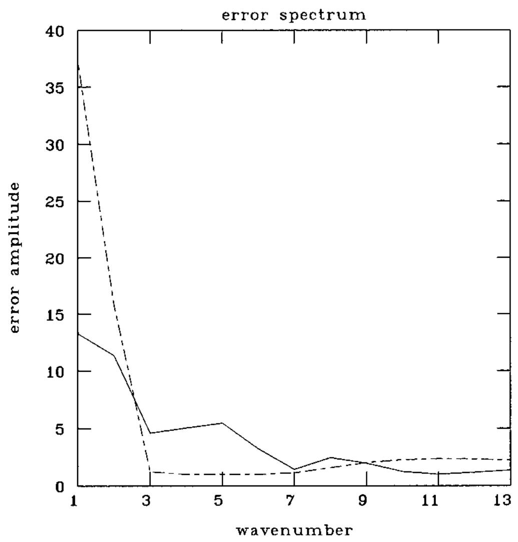


Figure 3.5
Analysis error spectra for steepest descent and standard conjugate gradient

Figure 3.6 is in the same format as Fig. 3.4 and compares the standard conjugate gradient (solid) vs the preconditioned conjugate gradient (dash-dot). The preconditioner was defined by dividing the domain into five equal intervals, each with five observations. The preconditioned algorithm converges significantly more quickly. Note the much lower analysis error on the first iteration because of the use of the approximate inverse.

Figure 3.7 shows the rms analysis error as a function of x between –π and π at iteration 2 for the standard conjugate gradient (solid) and preconditioned conjugate gradient (dash-dot). It can be seen that, although the preconditioned conjugate gradient is converging more rapidly, it is not converging evenly. In fact, the minimums of the error (most rapidly converging locations) are located in the center of each observation interval and the maximums occur on the interval boundaries.

Finally, Fig. 3.8 shows the rms analysis error as a function of iteration number in the same format as Fig. 3.4 for the BFGS (dash-dot) and preconditioned conjugate gradient (solid). In this case, e_r and e_b are as in Fig. 3.4 but $L_b = \pi/1.2$, a background error correlation length that is 10 times as large. In this case, the condition number is

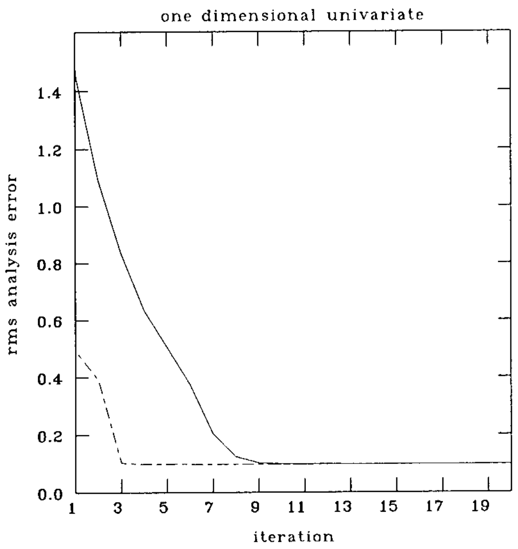


Figure 3.6
Rms analysis grid error - the effect of preconditioning

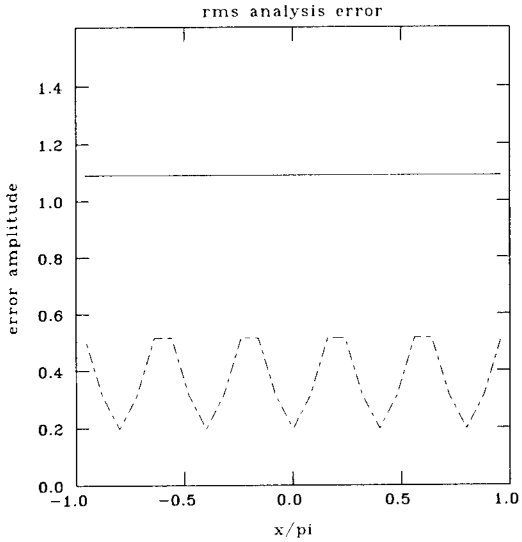


Figure 3.7
Spatial analysis error distribution at iteration 2

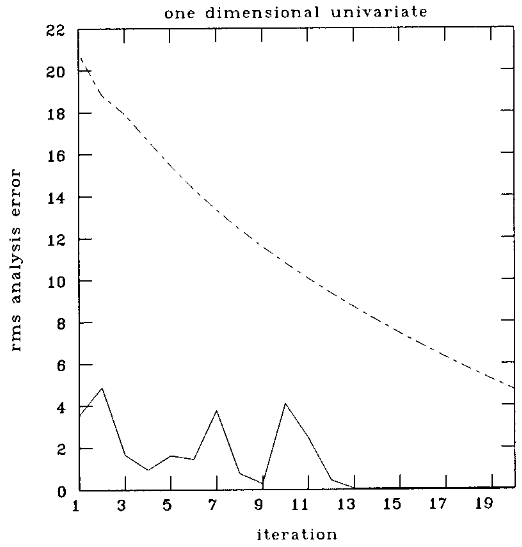


Figure 3.8
Comparison of BFGS and pre-conditioned conjugate gradient descents

much larger (2215), which might be expected from the discussion in section 2.3. Comparisons with Figs. 3.4 and 3.6 show that both descent algorithms converge more slowly for this higher condition number problem, but the preconditioned conjugate gradient is considerably faster. Note that the asymptotic analysis error level in this case is very small, but positive.

3.3.2 Two-dimensional experiments-the question of convergence

We will now consider a slightly more realistic two-dimensional, randomly located network. The problem remains univariate, and we again consider spatially uncorrelated observation error with error ε_r. The background error correlation is modelled with a SOAR model, as before, with error ε_b and correlation length L_b. The grid is 9×9 periodic $-\pi \leq x,y \leq \pi$, and there are 81 random observations. The forward model **H** is a Fourier interpolator.

In these two-dimensional experiments, we are going to examine the question of convergence-when can we consider the descent to have converged? We define three diagnostic techniques; two of them can only be used in simple situations and one with more general application.

(1) Examine trace ($\mathbf{P}_a$) as in the one dimensional experiments. In general, this is difficult because we do not calculate the weight matrix. However, see Section 10.1.

(2) Compare the analysis with the solution obtained by direct inversion of the $\mathbf{HP}_b\mathbf{H}^T + \mathbf{R}$ matrix. This is obviously impractical for large L.

(3) Calculate $\left\| \nabla J \right\|$, the norm of the gradient vector as a function of iteration number.

Since the gradient at any iteration is actually the residual (3.9), which is available at every iteration of the four descent algorithms considered above, the norm of the gradient can be calculated easily for any problem. These experiments were performed for all four descent algorithms, but the only results shown here are for the preconditioned conjugate gradient algorithm (3.13).

We consider the case $\varepsilon_b = 1.0$, $\varepsilon_r = 0.1$, and $L_b = \pi/2.4$. The condition number is 2506. We first attempt preconditioned conjugate gradient descent with nine observation volumes. These observation volumes happen to be rectangles rather than the triangles shown in Fig. 3.1, but this is not particularly significant for these simple experiments. Figure 3.9 shows the square root of trace ($\mathbf{P}_a$)–solid curve as in Fig. 3.4 (see (1) above), for this two-dimensional random network case as a function of iteration number (abscissa). The straight dash-dot line at 0.2 is the asymptotic value of the rms analysis error obtained by direct solution. (The problem is small enough for the matrix inverse to be obtained by direct methods).

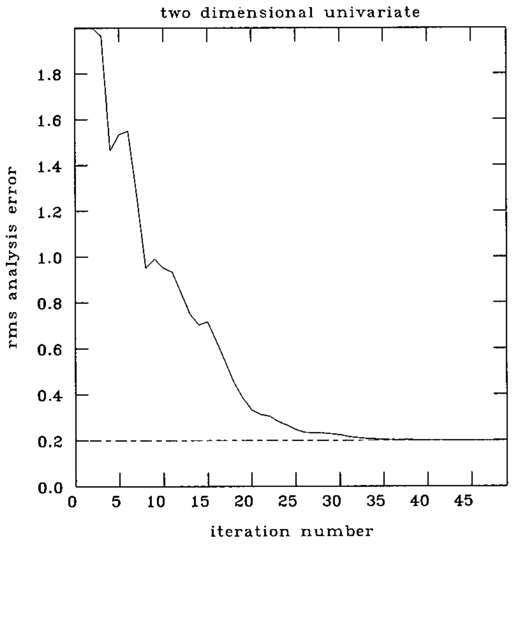


Figure 3.9
Rms analysis error (trace **Pa**) - conjugate gradient descent for 2D case

We also compared iterative solutions with the solution obtained by direct methods (see (2) above). Thus, we generated 10 innovation vectors **d** (3.5) using a random number generator, and determined the corresponding correction vector $\mathbf{x}_a - \mathbf{x}_b$ from these vectors using a direct solution as control. We then calculated estimated correction vectors at each step of the iterative process (for each of the descent algorithms) and calculated the rms difference on the grid between the control and descent correction vectors. Figure 3.10 is example of the preconditioned conjugate gradient algorithm. What is plotted is the logarithm (to the base 10) of the rms difference (ordinate) as a function of iteration number, for each of the 10 correction vectors. A decrease of 1 in the log(difference) implies that the difference field has decreased by a factor of 10. It can be seen that the scatter between the different realizations at any given iteration is close to a factor of 10. Comparing Figs. 3.9 and 3.10 indicates that convergence has been reached at about 30 iterations, or when the difference field has decreased by about three orders of magnitude.

Criteria for determining convergence cannot be practically based on the diagnostics of Figs. 3.9 and 3.10. A practical measure, however, can be based on estimating the norm of the gradient (see (3) above). Thus, the gradient of the cost function (actually the residual) is estimated at each iteration step. The gradient is a vector of length L, and the norm is simply defined as the square root of the sum of the elements of the gradient. Figure 3.11 shows the logarithm (base 10) of the norm of the gradient, for the same 10 innovation vectors as in Fig. 3.10, as a function of iteration number for the preconditioned conjugate gradient algorithm. Again, the spread between realizations is about an order of magnitude. Convergence is reached (about iteration 30), after the norm of the gradient has decreased by two orders of magnitude.

Other experiments (not shown) with steepest descent and the BFGS algorithm show much less rapid convergence for this example.

We might conclude that reduction of the norm of the gradient by two orders of magnitude is sufficient convergence. In practice, two orders of magnitude reduction in the norm of the gradient is more than sufficient for practical problems. In fact, one order of magnitude reduction in the norm of the gradient is usually sufficient for the whole algorithm ((3.5)-(3.6)). This is because the higher iterations are required primarily for convergence of the smaller spatial scales. However, the post-multiplication by $\mathbf{P}_b\mathbf{H}^T$ is a spatially smoothing operation when the background error covariance is derived from a red spectrum (as it usually is). Thus, the extra iterations in the solver required to resolve the smaller spatial scales usually do not have much effect on the final correction vector $\mathbf{x}_a - \mathbf{x}_b$ because of the smoothing effect of the post-multiplier.

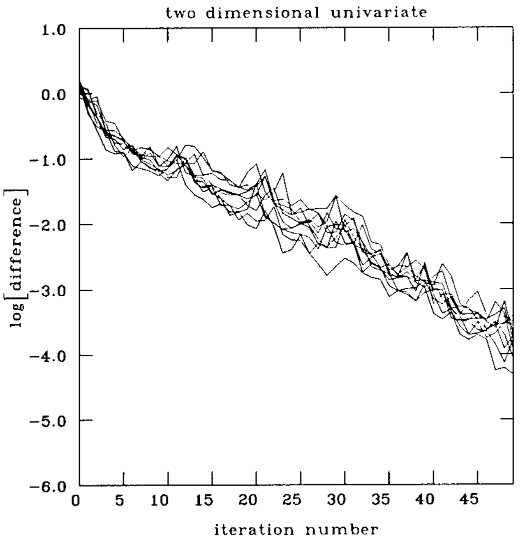


Figure 3.10
Comparison between direct solution and conjugate gradient descent

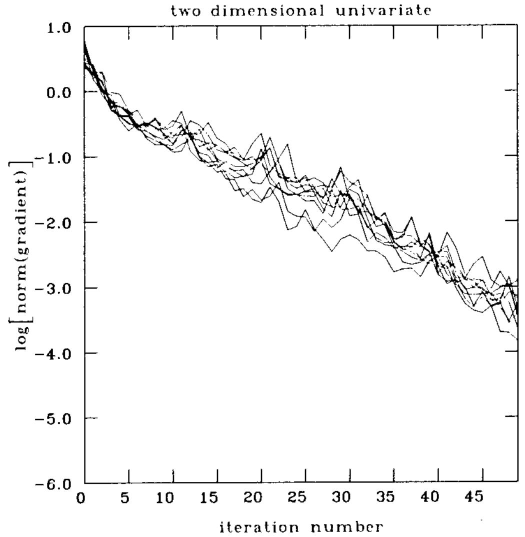


Figure 3.11
log$||\nabla J||$ for 10 realizations

3.4 Solving the Block-Diagonal Problems for the Conjugate Gradient Descent

The preconditioned conjugate gradient descent equation (3.13) involves solving a number of smaller matrix problems of the form (3.14). That is, we have to solve N problems of the form $\mathbf{A}_n*\mathbf{s} = \mathbf{f}$, where $\mathbf{A}_n*$ is always symmetric and positive-definite. As noted earlier, we solve these problems in one of two ways-a standard conjugate gradient descent (3.12) or by Cholesky decomposition. The standard conjugate gradient approach has already been explained, but it is worthwhile dicussing an efficient implementation of the Cholesky decomposition algorithm.

Cholesky decomposition works by rewriting the matrix $\mathbf{A}_n* = \mathbf{L}_n\mathbf{L}_n^T$, where $\mathbf{L}_n$ is a unique lower triangular matrix (Golub and Van Loan, 1996). Suppose $\mathbf{A}_n*$ is of order K_n. Then, it can be shown that obtaining the Cholesky matrix $\mathbf{L}_n$ is an order K_n^3 operation. However, once the Cholesky matrix has been obtained, then the problem (3.14) can be rewritten as $\mathbf{L}_n\mathbf{r}_n = \mathbf{f}_n$ and $\mathbf{L}_n^T\mathbf{s}_n = \mathbf{r}_n$. Thus, if $\mathbf{f}_n$ is known, we can obtain $\mathbf{s}_n$ by solving two triangular problems, each of which takes order K_n^2 operations. Thus, we can calculate and store the Cholesky matrix $\mathbf{L}_n$ for each of the N diagonal blocks immediately after calculation of the elements of $\mathbf{A}_n*$. There are only $K_n(K_n+1)/2$ nonzero elements of $\mathbf{L}_n$, as opposed to K_n^2 locations for $\mathbf{A}_n*$ itself. Moreover, during the iterative loop in (3.13), we only have to perform the K_n^2 operation to solve for $\mathbf{s}$ at each iteration and do not have to perform the K_n^3 operation generating the Cholesky matrices $\mathbf{L}_n$.

3.5 A Convergence Accelerator Based on a Re-sorting of the Observations (the Dual Choleski Algorithm)

The preconditioning strategy for the conjugate gradient algorithm discussed hitherto is very similar to that previously developed by Cohn et al (1998). However, we have developed two algorithms for accelerating the convergence of the conjugate gradient descent by modifying the preconditioning strategy. The first algorithm involves a re-sorting of the observations, while the second algorithm is much more complex and is discussed in Appendix E. The re-sorting algorithm is discussed in this section and has been implemented in both serial and parallel versions.

Figure 3.2 shows schematically how the observations are sorted into (volumes) triangular prisms; in Fig. 3.3 (lower left panel) shows the block-diagonal preconditioner matrix. Consider one of the points in Fig. 3.2 where several triangles intersect. In the vicinity of such a point, many observations may lie whose background error is highly correlated because they are close to each other. These high correlations would be reflected by elements of the $\mathbf{A}$ matrix of (3.13). However, in the preconditioner matrix $\mathbf{A}*$, all correlations are equal to zero except for correlations between observations in the same volume (triangular prism). Clearly, in the vicinity of one of these points, there would be many large correlations that would be ignored by the preconditioner $\mathbf{A}*$, and this would mean that the preconditioner may not be as efficient as we would like.

Suppose however, we re-sorted the observations into a different set of triangular prisms, and this time the central points of the triangles were located at the intersection points of Fig. 3.2. This may or may not always be geometrically possible, we just choose the second set of volumes so that (as much as possible) they contain different sets of observations than the original set of volumes. We then calculate only the block-diagonal elements of A corresponding to this new sort (we do not have to calculate the corresponding off-diagonal block of A, because we do not need them). We refer to this new preconditioner as $\boldsymbol{\alpha}*$, with diagonal blocks $\boldsymbol{\alpha}_n*$. The diagonal blocks will, of course, be of different size than in $\mathbf{A}*$. The second preconditioner is shown schematically in the right-hand lower panel of Fig. 3.3.

Now we solve two preconditioning problems, $\mathbf{A}*\mathbf{s} = \mathbf{f}$ for $\mathbf{s}$ as before, and a second problems $\boldsymbol{\alpha}*\boldsymbol{\sigma} = \boldsymbol{\phi}$ for $\boldsymbol{\sigma}$. Here $\boldsymbol{\phi}$ is simply an appropriately reshuffled version of $\mathbf{f}$. We then simply take the simple average of the two vectors $\mathbf{s}$ and $\boldsymbol{\sigma}$. This combined preconditioner includes many interactions between observations that were excluded from the simple preconditioner. Figure 3.2 (lower panel) suggests that many additional matrix elements of **A** would now be included in the preconditioner. In principle, one could have 3, 4, or more preconditioners of this sort, but these additional complications may be counterproductive.

The algorithm was first tested on one-dimensional univariate problems with randomly scattered observations. For these problems, the observations were sorted into equal intervals in the first preconditioner, and these intervals were shifted by one-half interval for the second preconditioner. We can define a simple sorting efficiency in this case. For the nth observation, define the number of observations in the same interval in the first preconditioner as $I_1(n)$. Now define $I_2(n)$ as the number of observations that were in the same interval as observation n in the first preconditioner, but that are also in the same interval as observation n in the second preconditioner. Define the ratio $I_2(n)/I_1(n)$ for all n, and its average over n as the sorting efficiency. For the one-dimensional case with equal intervals and the second preconditioner shifted by half an interval, the sorting efficiency would be 0.5. In this simple example, the use of two preconditioners sped up the descent by a factor of 2.

The algorithm was also tested on two-dimensional univariate examples with randomly scattered observations. If the observations were sorted into squares of equal spatial size, then the second preconditioner can be obtained by shifting the squares one-half interval in each direction and re-sorting into the new squares. This gives a sorting efficiency of 0.25. If equilateral triangles are used instead of squares, and the second preconditioner shifts the triangles appropriately, the optimum sorting efficiency is also 0.25. In this case also, the use of the second preconditioner sped up the descent by a factor of 2.

This algorithm was also tested on a full three-dimensional set of observations, including direct assimilation of radiances (Section 5.3). This problem is considerably more difficult because the observation density is variable; there are many types of observations, and we have to keep all observations in the same profile or sounding together. Although the problem is three dimensional, the sorting is two dimensional, so we would like to achieve a sorting efficiency for the second preconditioner that is not too much larger than 0.25. However, the second preconditioner also has to be as good a preconditioner as the first preconditioner; there is no point combining a bad preconditioner with a good one.

Figure 3.12 is an example of iterating with two preconditioners. The observation set consisted of 5210 observations (radiosonde, SSM/I windspeeds, SSM/I total precipitable waters, and TOVS radiances in 20 channels). The second preconditioner was created by rerunning the sorter algorithm with slightly different parameters. The sorting efficiency, by the measure above, turned out to be 0.49, which is perhaps a little larger than desired. Figure 3.12 is in the same format as Fig. 3.11. The solid curve shows $\log_{10}(\|\nabla \mathbf{J}\|)$ for the original preconditioner and the dash-dot curve for the dual preconditioners as a function of iteration number. The descent rate is similar at first, but then the dual preconditioner descends much more rapidly. Note the change in curvature in the solid curve after about eight iterations. The same thing can also be seen in Fig. 3.11, indicating that with a single preconditioner, neglected large correlations in the single preconditioner cause the descent rate to slow down after $\|\nabla \mathbf{J}\|$ has decreased by only about one order of magnitude. There is some overhead connected with the algorithm, in particular, the calculation and (more importantly), the storage of the Choleski decomposition triangular matrices for the second preconditioner. The use of the extra Choleski matrices during the descent has only a tiny negative effect on computational efficiency.

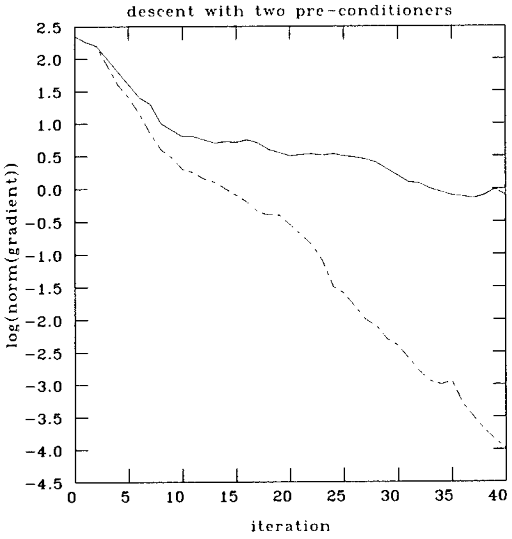


Figure 3.12
The effect of dual Choleskis

3.6 Comparison between NAVDAS and a Simulated Volume OI Algorithm

The NRL MVOI and NAVDAS algorithms are both observation space algorithms. There are many differences between the way these two algorithms are applied, but we concentrate here on the difference between the global solution of NAVDAS and the local volume solution of MVOI. As discussed earlier, the preconditioner used in OI bears some resemblance to the volume procedure of OI. Thus, it is possible to modify the NAVDAS algorithm so that it simulates the solution obtained by volume OI.

The following procedure produces a simulated OI correction field. We use only the preconditioner (Section 3.4) to solve the system (Eq. (3.5)). Then, in the post-multiplication step (Eq. (3.6)), we only allow each block of analysis gridpoints (Section 3.7) to be affected by observations that are less than that 2000 km away. This means that there may be discontinuities in the correction field at the junctions between the analysis blocks. Note that this simulation does not simulate the overlap between analysis volumes that occur in the actual NRL MVOI implementation. This means that the simulation probably is not as good as an actual OI algorithm.

Figure 3.13 a shows a comparison between NAVDAS and volume OI simulated in this manner. The experiment was run over the globe, using all the radiosondes (temperatures, winds, moisture on all mandatory and significant levels). There were about 62,000 observations. We show the resulting 250 hPa geopotential correction field (m) on a global 1-degree grid. In the NAVDAS run (panel a), there were eight iterations using two preconditioners. The simulated volume OI run (panel b) was constructed as above.

Looking first at the southern hemisphere, we see that the two solutions are similar, because what little data there is, is widely separated. The solution obtained by the preconditioner is not affected by other data that are too far away to have any influence. In the northern hemisphere, particularly over land where data density is high, the situation is very different. The two solutions differ substantially; in panel (b), the analysis volume boundaries are easily visible. Generally speaking, the wind correction fields of NAVDAS and simulated OI are much closer to each other; the wind observations converge more quickly in the 3DVAR solver because the effective horizontal correlation length for winds is much shorter than for the geopotential.

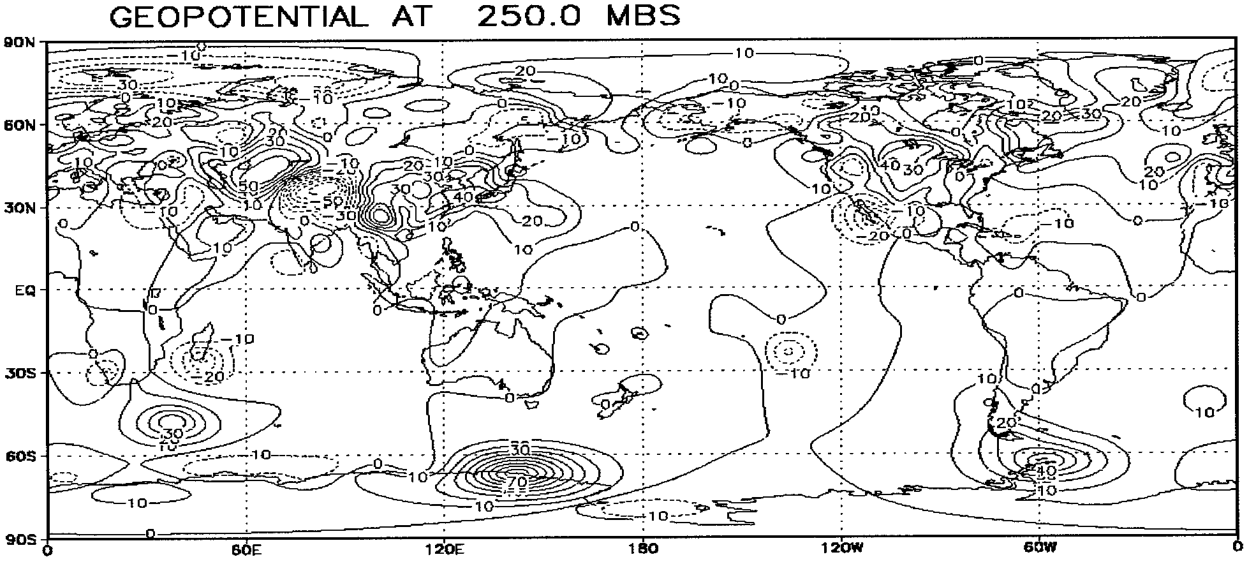

GEOPOTENTIAL AT 250.0 MBS
90N
60N
30N
EQ
30S
60S
90S
0
60E
120E
180
120W
60W
0
10
20
30
40
50
70
-10
-20
-30
ANALYSIS ERROR FOR 1998011400 : test

Figure 3.13a
3 DVAR correction field

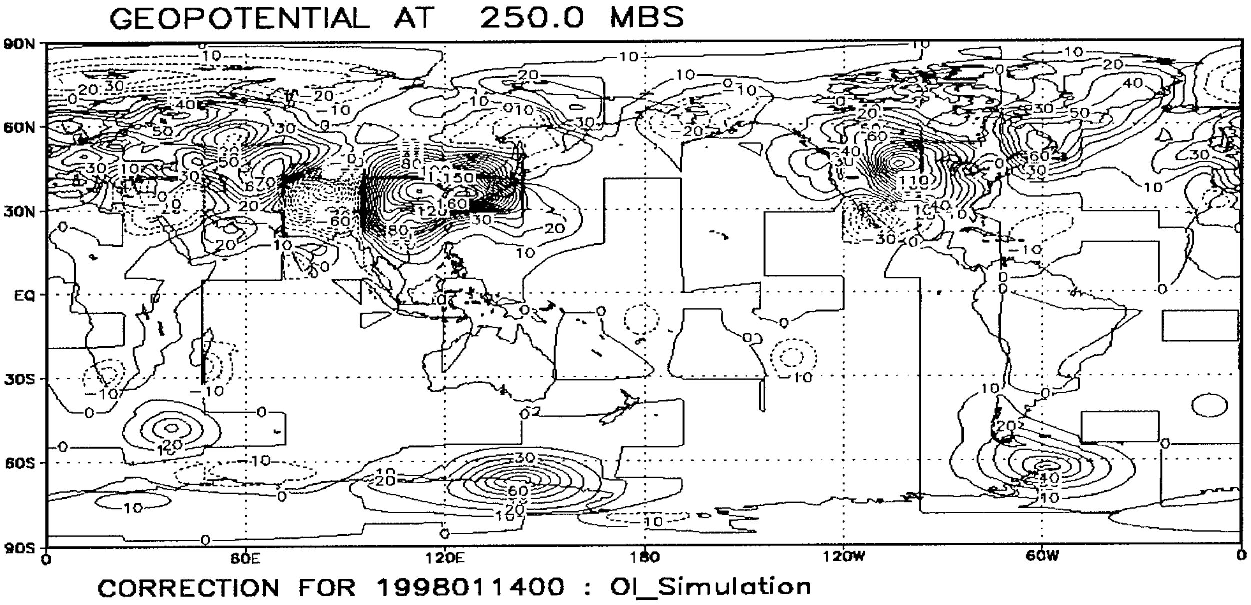

GEOPOTENTIAL AT 250.0 MBS
90N
60N
30N
EQ
30S
60S
90S
0
60E
120E
180
120W
60W
0
CORRECTION FOR 1998011400 : OI_Simulation

Figure 3.13b
Simulated OI correction field

3.7 Defining the Analysis Grid Volumes

As described in Sections 3.2.6 and 3.4, the preconditioner is defined by sorting the observations into prisms. A similar procedure is applied to the analysis gridpoints for the calculation of the correction fields by the post-multiplier (Eq. (3.6)). Thus the analysis gridpoints are broken into rectangular blocks. All correction field variables and vertical levels would be in this block. Then the interactions are calculated between each of the analysis gridpoints in the block and the observations in each of the observation prisms. (We may exclude interactions between observation prisms and analysis volumes that are too widely separated.)

While this break-up of the analysis grid into grid volumes is not as important to the algorithm as the break-up of the observations into prisms, it is very useful in parallel implementations of the post-multiplier (see Appendix C).

3.8 Scaling

For the multivariate problem where there is a mix of variables with different units, it is desirable that the equations be scaled. This helps improve the convergence of the descent algorithms. Scaling for Method C has some similarity to the normalization process used in the MVOI algorithm. Before describing the scaling, we slightly modify Eq. (3.4) for Method C. Suppose that the forward interpolation operator H is the product of a horizontal spatial interpolation operator H_* from the grid to the observation location and a forward operator H that transforms from the analyzed/forecast variable to the observed variable (the radiative transfer equation, for example). That is, we redefine H so that $\mathbf{P}_b\mathbf{H}^T$ is now written $\mathbf{P}_b\mathbf{H}_*^T\mathbf{H}^T$, and $\mathbf{H}\mathbf{P}_b\mathbf{H}^T$ is now written as $\mathbf{H}\mathbf{H}_*\mathbf{P}_b\mathbf{H}_*^T\mathbf{H}^T$. We now make the same approximation that is used in MVOI, that is, we write

$$\mathbf{P}_b\mathbf{H}_*^T = \mathbf{P}_b^{gr/ob} \text{ and } \mathbf{H}_*\mathbf{P}_b\mathbf{H}_*^T = \mathbf{P}_b^{ob/ob}, \qquad (3.15)$$

where $\mathbf{P}_b^{gr/ob}$ is the background error covariance between grid and observation locations and $\mathbf{P}_b^{ob/ob}$ is the background error covariance between observation locations. This approximation is very good as long as the grid resolution is adequate, which it generally is. Now, let us write

$$\begin{aligned} &\mathbf{P}_b^{gr/ob}\mathbf{H}^T = [\mathbf{S}_b]^{1/2}\mathbf{C}_b^{gr/ob}[\mathbf{S}_b^{ob}]^{1/2}\,\mathbf{H}^T, \\ &\mathbf{H}\mathbf{P}_b^{ob/ob}\mathbf{H}^T = \mathbf{H}\,[\mathbf{S}_b^{ob}]^{1/2}\mathbf{C}_b^{ob/ob}[\mathbf{S}_b^{ob}]^{1/2}\,\mathbf{H}^T, \\ &\mathbf{S}_h = \mathrm{diag}(\mathbf{H}\mathbf{P}_b^{ob/ob}\mathbf{H}^T)\ , \end{aligned} \tag{3.16}$$

where $\mathbf{S}_b = \mathrm{diag}(\mathbf{P}_b)$, $\mathbf{S}_b^{ob} = \mathrm{diag}(\mathbf{P}_b^{ob/ob})$, and $\mathbf{C}_b^{gr/ob}$ and $\mathbf{C}_b^{ob/ob}$ are the correlations that correspond to the covariances $\mathbf{P}_b^{gr/ob}$ and $\mathbf{P}_b^{ob/ob}$ respectively. Then, we rewrite Eq. (3.4), using Eqs. (3.15)-(3.16) as

$$\mathbf{x}_a - \mathbf{x}_b = \mathbf{S}_b^{1/2}\mathbf{C}_b^{gr/ob}\,[\mathbf{S}_b^{ob}\,]^{1/2}\mathbf{H}^T\mathbf{S}_h^{-1/2}\,[\mathbf{C}_h^{ob/ob} + \mathbf{S}_h^{-1/2}\mathbf{R}\mathbf{S}_h^{-1/2}]^{-1}\,\mathbf{S}_h^{-1/2}[\mathbf{y} - \mathrm{H}(\mathbf{x}_b^{ob})], \tag{3.17}$$

where $\mathbf{C}_h^{ob/ob} = \mathbf{S}_h^{-1/2}\mathbf{H}[\mathbf{S}_b^{ob}]^{1/2}\mathbf{C}_b^{ob/ob}[\mathbf{S}_b^{ob}]^{1/2}\mathbf{H}^T\mathbf{S}_h^{-1/2}$, $\mathbf{x}_b^{ob}$ is the background spatially interpolated to the observation location, and the operator **H** transforms from grid variable to observation variable but does not perform spatial interpolation. Equation (3.17) amounts to the scaling of the innovations by the rms background error (in **H** space) and post-multiplication of the correction vector by the rms background error at the grid locations. Equation (3.17) is now essentially nondimensional. We can think of $\mathbf{S}_h^{-1/2}\mathbf{H}[\mathbf{S}_b^{ob}]^{1/2}$ as a scaled forward operator. In the special case where observation and grid variables are the same, then $\mathbf{H} = \mathbf{I}$ (the identity matrix), $\mathbf{C}_h^{ob/ob} = \mathbf{C}_b^{ob/ob}$, $\mathbf{S}_h = \mathbf{S}_b^{ob}$, and Eq. (3.17) becomes

$$\mathbf{x}_a - \mathbf{x}_b = \mathbf{S}_b^{1/2}\mathbf{C}_b^{gr/ob}[\mathbf{C}_b^{ob/ob} + [\mathbf{S}_b^{ob}]^{-1/2}\mathbf{R}[\mathbf{S}_b^{ob}]^{-1/2}]^{-1}\,[\mathbf{S}_b^{ob}]^{-1/2}[\mathbf{y}\quad \mathbf{x}_b^{ob}]. \tag{3.18}$$

3.9 High-density Single-level Observations – Clustering within Prisms

NAVDAS is an observation space algorithm. Thus, it tends to be computationally efficient when the observation density is lower than the grid density and less efficient when the observation density exceeds the grid density. High observation density in the vertical (such as in a radiosonde) can be handled very efficiently using the background vertical eigenvector decomposition to be discussed in Section 4.3.3. What we wish to discuss here is the case of high horizontal density observations, particularly single-level observations such as conventional surface observations, aircraft observations, cloud drift winds, and SSM/I windspeed and total precipitable water observations. The vertical eigenvector decomposition (of Section 4.3) is not efficient for these observations. We can and do use super-obbing or thinning techniques on these types of observations. We also have the following clustering algorithm.

Re-write Eq. (3.4) in unscaled form using the forward operators **H** and $\mathbf{H}_*$ of Section 3.8 as

$$\mathbf{x}_a - \mathbf{x}_b \ = \ \mathbf{P}_b\mathbf{H}_*^T\mathbf{H}^T[\mathbf{H}\mathbf{H}_*\mathbf{P}_b\mathbf{H}_*^T\mathbf{H}^T + \mathbf{R}]^{-1}[\mathbf{y} - \mathrm{H}(\mathbf{H}_*\mathbf{x}_b)]. \tag{3.19}$$

Here, $\mathbf{H}_*$ is the horizontal interpolation operator and **H** denotes any other forward operations. As described in Section 3.8, we generally make the approximation (3.15). This is the standard MVOI approximation in which the covariances are calculated directly at the observation locations rather than first using the horizontal interpolation operator $\mathbf{H}_*^T$ to interpolate to the analysis grid. This is generally a good and efficient approximation. However, when the local horizontal observation density is greater than the horizontal grid density, this approximation becomes inefficient. Thus, for a given observation type (or group of related observation types), for a given variable type (temperature, wind, etc.) and for a given vertical interval; when there is more than one observation within a horizontal grid square, the MVOI approximation above is no longer efficient. Thus, within an observation prism, when the local observation density becomes sufficiently high, it may be more efficient to explicitly include the interpolation operator $\mathbf{H}_*$ in Eq. (3.19). In this special case, we have the rather desirable situation where NAVDAS might become more efficient as well as being truer to the algorithm (3.19).

In practice, we define each set of observations within a prism that share the same observation type (or group of observation types), variable type and vertical level (or interval) as an observation cluster. If this observation cluster has fewer observation locations than gridpoints that it would project onto (using the $\mathbf{H}_*$ operator), then we treat all the observations in it conventionally (as in Section 3.8). However, if there are more observation locations in the cluster than gridpoints to be projected onto, then we actually use the operators $\mathbf{H}_*$ and $\mathbf{H}_*^T$ as in Eq. (3.19). This means that NAVDAS becomes increasingly a grid-based algorithm as the horizontal observation density increases.

In principle, one should use the analysis grid and the horizontal interpolation operator that is actually used in interpolating the background field to the observation locations. In the case of the NOGAPS and COAMPS data assimilation systems, this would imply two different grids and at present non local cubic spline interpolators. This approach would have several difficulties. First, it would make NAVDAS considerably more analysis-grid-dependent than before, reducing flexibility and portability. Secondly, the cubic spline interpolators are non local, which means that each observation is projected over the entire domain, clearly undesirable in a spatially local algorithm like NAVDAS.

Consequently, we adopted the following modified approach. Firstly, we define a prism grid that has the same local resolution as the analysis grid, and the locations of whose gridpoints are simple to determine. Secondly, the forward interpolation operator $\mathbf{H}_*$ used in the above clustering procedure was chosen to be linear and local—a four-point interpolator between the observation location and the four surrounding gridpoints. The four weights were calculated from the Great Circle distances between the observation locations and the four gridpoints.

When the local observation density of the cluster exceeds the local grid density, then the clustering algorithm above comes into play. The local grid density is fixed, which means that the cost of the NAVDAS algorithm would remain essentially constant as the local observation density increased. Furthermore, if we are prepared to accept a possible small decline in the quality of NAVDAS analyses, then by modestly decreasing the grid resolution, NAVDAS will run considerably more efficiently when there are large numbers of single level observations of the types described above.

4. Background Error Covariances

Specification of the background error covariances is very important in NAVDAS as it is in the NRL MVOI algorithm. The formulation of the background error covariances in MVOI was reasonably powerful, being based on Lorenc (1981), Lonnberg and Hollingsworth (1986), and Daley (1991). However, not all the power was exploited in MVOI. In NAVDAS, new features have been added to produce a very general formulation of the background error covariance.

From Eq. (3.16), we separate the background error covariances into the background error variance $\mathbf{S}_b$ and correlation matrix $\mathbf{C}_b$. We discuss separately the specification of the background error correlations and variances, starting with the variances.

4.1 Background Error Variances

Specification of the background error variances $\mathbf{S}_b$ is very straightforward. In Section 3.8, the variances $\mathbf{S}_b$ are actually diagonal matrices and we treat them that way in the derivations to follow. However, in practice, we carry only the diagonal components. These variances are permitted to vary by latitude, longitude, vertically, and (of course) by variable. The geopotential and temperature variances are specified to be in exact hydrostatic balance. The wind and geopotential background error variances are approximately geostrophically related in the extratropics, but are independent in the tropics. The background error variances are not required to be invariant within observation or grid volumes.

If Φ is the geopotential and T is the temperature, the $<\Phi T>$ and $<TT>$ covariances must be related to the $<\Phi\Phi>$ covariances hydrostatically. Since this is a temperature-based system, not a geopotential-based analysis system (see Section 5.2), we generally specify the $<TT>$ variance and demand that the $<\Phi\Phi>$ variance be related hydrostatically by integrating the hydrostatic relation up from the surface. This is necessary to ensure that the covariances (not the correlations) are related hydrostatically. In the vertical, this rule has been carefully adhered to.

There is a similar problem in the horizontal with respect to the geostrophic relation of wind/wind and wind/geopotential covariances to the $<\Phi\Phi>$ covariances. This requires that the $<\Phi\Phi>$ covariance be differentiated analytically or discretely once or twice. In MVOI (and most implementations of OI), it has been considered adequate to differentiate (analytically) the $<\Phi\Phi>$ correlation. This procedure is correct if the geopotential error variances are horizontally invariate, but ignores an extra term if they are not. That is, there is a term involving the horizontal derivatives of the background error variances that is nonzero when the background error geopotential error is horizontally variable.

However, this extra term will be small as long as the horizontal variation of the background error variance is on horizontal scales that are large compared with the specified horizontal scales of the background error correlation. Thus, we only permit the background error variances to vary horizontally on scales that are large compared to the characteristic horizontal scales of the background error correlation.

In general, the background error variances are assumed to be the product of a dimensional constant, a nondimensional O(1) horizontal function, and a nondimensional O(1) vertical function. This vertical function indicates the variation of the background error variance with respect to the vertical coordinate. For future reference, we denote the vertical background error variance for streamfunction, wind, geopotential, and temperature as $\mathbf{S}_\psi$, $\mathbf{S}_v$, $\mathbf{S}_\Phi$, and $\mathbf{S}_T$, respectively.

4.2 Background Error Correlations

Specification of the background error correlations is much more complex, and its importance cannot be overestimated. All correlations are written as the product of a horizontal and vertical correlation. Thus, if we define horizontal coordinates λ (longitude) and θ (latitude) and vertical coordinate z (which could stand for pressure, potential temperature, or other vertical variable), then the background error correlation between points 1 and 2 can be written

$$c_b(\lambda_1,\theta_1,z_1,\lambda_2,\theta_2,z_2) = c_b^v(z_1,z_2,L_b^v(z_1,z_2))\, c_b^h(s_{12},\alpha_{12},L_b^h(\lambda_1,\theta_1,\lambda_2,\theta_2)), \qquad (4.1)$$

where c_b^v is the vertical correlation and c_b^h is the horizontal correlation between points 1 and 2. s_{12} is the great-circle distance, and α_{12} is the angle between the two points.

L_b^v is a vertical correlation scale (which may be a vertically variable), and L_b^h is a horizontal correlation scale (which could be horizontally variable). There could be other parameters in addition to the horizontal and vertical correlation lengths. The form (4.1) is horizontally separable. A nonseparable generalization can be created by making L_b^v a function of the horizontal variables, or L_b^h a function of the vertical variables (see Section 4.7). If c_b^h depends on s_{12} but not α_{12}, and L_b^h is spatially invariant, then c_b^h is said to be isotropic. Anisotropy can be introduced by allowing L_b^h to vary horizontally, or by permitting dependence on α_{12}.

The horizontal background error correlations are considered in Section 4.6. We first consider the background vertical error correlations in detail. They constitute the most important and original components of the background error formulation.

4.3 Vertical Background Error Correlations — Separable Formulation

We pay considerable attention to formulation of the vertical background error correlation because of some evident shortcomings in the NRL MVOI formulation.

4.3.1 Problems with MVOI

Examination of the background error vertical structure functions of the MVOI revealed two fairly basic problems. Denote Φ,T as geopotential and temperature, respectively, and the vertical coordinate z will be log(pressure).

(1) The hydrostatic relationship used to calculate <ΦT> , <TΦ> and <TT> correlations from <ΦΦ> correlations produced anomalies at the lowest levels.

(2) The background error vertical structure used for the <ΦΦ> correlation used a correlation function, which was not twice differentiable. This meant that the <TT> correlation had some unfortunate properties.

The <ΦΦ> correlation function used in the NRL MVOI (see Lonnberg and Hollingsworth, 1986) was

$$c_b^v(z_n,z_m) = \exp[-(\left|z_n - z_m\right| / L_b^v)^{1.6}],$$

where $z_n = \log_e(P_n)$ and L_b^v is the background error vertical correlation scale. This function has a continuous first derivative, but not a continuous second one. This was not a serious problem for the <TT> correlation at the relatively low vertical resolution used in the NRL MVOI, but it would cause problems in the much higher vertical resolution used in NAVDAS.

4.3.2 The NAVDAS streamfunction/streamfunction correlation models

In the NAVDAS vertical background error correlation formulation, the starting point is the streamfunction/ streamfunction $<\psi\psi>$ correlation. We define a vertical index $1 \leq n \leq N_v$, where N_v is the number of vertical levels. Generally speaking, N_v will be a reasonably large number (30-100) so that the vertical resolution is adequate. Thus, when using any vertical correlation, a vertical location will always be denoted in terms of the integer index n. All we require in defining the vertical correlation between any two observations or grid points is the location of these points in whatever vertical coordinate we are using. Thus, the use of a vertical index to define vertical locations makes it easy to use different vertical coordinate systems such as pressure or isentropic.

At this point, we assume that the vertical coordinate system is pressure. For pressure coordinates, we assume that the vertical levels for the correlation functions are arbitrarily (but sometimes equally) spaced in log pressure from some bottom level (P_{bot}) to some top level (P_{top}). We use a simple algorithm that will take the pressure location of any observation or grid point and convert it into the vertical index.

The vertical structure of the $<\psi\psi>$ correlation is specified with a simple model such as the SOAR or Gaussian. The model must be twice differentiable. We assume that the background error vertical correlation scale is vertically dependent. That is, we define L(P) to be the local vertical correlation length scale as a function of pressure P. (We have dropped the understood subscript b and superscript v). We also define $z = \log_e(p)$ and L(z). Now define

$$y = -\int_{P_{bot}}^{P} \left[L(P')P'\right]^{-1} dP' = \int_{0}^{z} \left[L(z')\right]^{-1} dz' . \tag{4.2}$$

Then, for two levels n and m, we can define the $<\psi\psi>$ correlation as follows:

$$\begin{aligned} &(1)\ \text{SOAR} && c_{nm} = (1 + |\Delta y|)\exp(-|\Delta y|), \\ &(2)\ \text{Gaussian} && c_{nm} = \exp[-(\Delta y)^2], \end{aligned} \tag{4.3}$$

where $\Delta y = \int_{z_m}^{z_n} L^{-1}(z)dz$. Note that if L is independent of P, then $\Delta y = [z_n - z_m] / L$. In practice, the integrals are quadratures. This formulation, using integrals, tends to yield positive definite correlations and covariances, even when L varies rapidly with pressure.

Finally define $\mathbf{C}_{\psi\psi}$ as the $<\psi\psi>$ correlation, whose elements are given by Eq.(4.3).

4.3.3 The vertical eigenvector decomposition

We first define a diagonal $N_v \times N_v$ matrix $\tilde{\mathbf{D}}$, all of whose elements are positive. We then define the eigenvector decomposition

$$\tilde{\mathbf{D}}\mathbf{C}_{\psi\psi}\tilde{\mathbf{D}} = \tilde{\mathbf{E}}\mathbf{D}_{\psi\psi}\tilde{\mathbf{E}}^{T}, \tag{4.4}$$

where $\tilde{\mathbf{E}}$ is the $N_v \times N_v$ eigenvector matrix and $\mathbf{D}_{\psi\psi}$ is the diagonal matrix of (positive) eigenvalues. Let us define $\mathbf{E} = \tilde{\mathbf{D}}^{-1}\tilde{\mathbf{E}}$, which are not strictly eigenvectors unless $\tilde{\mathbf{D}}$ is the identity matrix. Then, $\mathbf{C}_{\psi\psi} = \mathbf{E}\mathbf{D}_{\psi\psi}\mathbf{E}^{T}$.

We first consider the separable formulation. In the separable formulation, $\tilde{\mathbf{D}}$ is the identity matrix. In this case, $\mathbf{E}$ really is the eigenvector matrix of $\mathbf{C}_{\psi\psi}$. Figure 4.1 illustrates the first three (gravest) eigenmodes of (4.4) for the separable case as a function of 50 pressure levels (equally spaced in log(P) between 1070 and 50 hPa) using the SOAR model. The vertical scale $L = L_h^v$ is pressure-invariant in this case. The gravest mode (largest eigenvalue in $\mathbf{D}_{\psi\psi}$) is the solid curve, the second mode is the dash-dot, and the third mode is the dashed curve. Note that the first mode has a minimum at the upper and lower boundaries. The structures of these modes would change if the vertical scale L varied with pressure and/or the pressure levels were unequally spaced in pressure. However, the general pattern of the largest vertical scales being associated with the largest eigenvalues would remain true.

For the nonseparable case to be considered in section (4.7.2), we may wish to choose a more general form of $\tilde{\mathbf{D}}$. For example, if the (diagonal) elements of $\tilde{\mathbf{D}}$ increase monotonically with decreasing pressure, then the leading (most grave) eigenvectors will have their maximum amplitude near the top of the atmosphere. This is illustrated by Fig. 4.2 (in the same format as Fig. 4.1). Everything is the same in Fig. 4.2 as in Fig. 4.1 except that the elements of $\tilde{\mathbf{D}}$ increase linearly in log(pressure), from a value of 1.0 at 1000 hPa to 4.0 at 50 hPa. Not shown are the eigenvalues, which are also different from those corresponding to Fig. 4.1. Comparison of Figs. 4.1 and 4.2 shows that for the gravest modes the maximum amplitudes tend to be shifted to lower pressure, as expected. This will turn out to be a useful property when we consider nonseparable correlations in Section (4.7).

Note that even for the nonseparable case, $\mathbf{C}_{\psi\psi}$ will remain independent of $\tilde{\mathbf{D}}$ unless only a subset of the eigenvectors are used (Section 5.3). Other covariances derived from $\mathbf{C}_{yy}$ may also depend on $\tilde{\mathbf{D}}$ if horizontal parameters such as the geostrophic coupling parameter (Section 4.3.7) or the horizontal correlation length (Section 4.7.3) are allowed to vary with the vertical eigenmodes.

The streamfunction/streamfunction error covariance would be $\mathbf{S}_\psi^{1/2}\mathbf{C}_{\psi\psi}\mathbf{S}_\psi^{1/2}$.

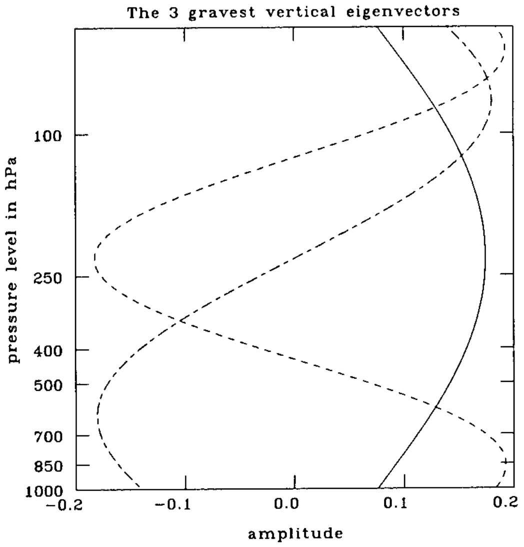


Figure 4.1
The three gravest vertical eigenvectors of $C\varphi\varphi$

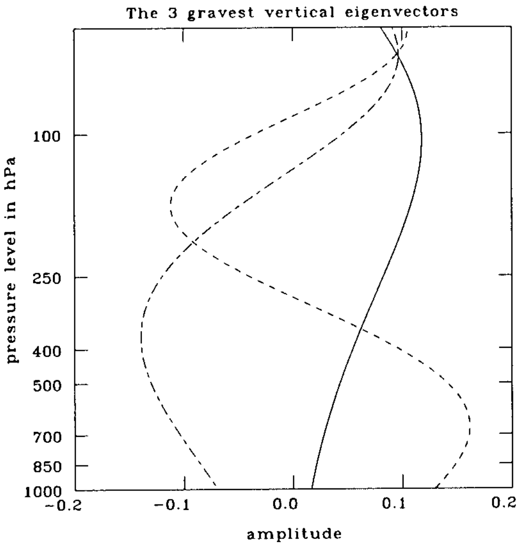


Figure 4.2
Same as (4.1) except for general $\tilde{\mathbf{D}}$

4.3.4 Wind/wind correlations

We denote the velocity potential/velocity potential correlation as $\mathbf{C}_{\chi\chi}$ and the velocity potential/streamfunction correlations as $\mathbf{C}_{\psi\chi} = \mathbf{C}_{\chi\psi}$. We define these correlations as in Eq. (4.4), with the same eigenvectors, but we retain the generality of using different eigenvalues. Thus, we write $\mathbf{C}_{\chi\chi} = \mathbf{E}\mathbf{D}_{\chi\chi}\mathbf{E}^T$ and $\mathbf{C}_{\psi\chi} = \mathbf{C}_{\chi\psi} = \mathbf{E}\mathbf{D}_{\psi\chi}\mathbf{E}^T$ using the matrices $\mathbf{E}$ and $\mathbf{E}^T$ defined in (4.4). Here $\mathbf{D}_{\chi\chi}$ and $\mathbf{D}_{\psi\chi}$ are diagonal matrices, all of whose elements are positive.

In the separable formulation $\mathbf{D}_{\chi\chi} = \mathbf{D}_{\psi\psi}$ and $\mathbf{D}_{\psi\chi} = 0$, but in the nonseparable formulation (Section 4.7), they may differ. Since $\mathbf{D}_{\chi\chi} = \mathbf{D}_{\psi\psi}$ in the separable formulation, the rotational wind/rotational wind and divergent wind/

divergent wind correlations have the same vertical structure, and we can define the vertical wind/wind correlations as

$$\mathbf{C}_{vv} = \mathbf{E}\mathbf{D}_{\psi\psi}\mathbf{E}^T. \tag{4.5}$$

The vertical background wind and streamfunction error variances are the same $\mathbf{S}_v = \mathbf{S}_\psi$, but this formulation must be modified for the nonseparable formulation (Section 4.7).

4.3.5 Geopotential/geopotential correlations

We define $\mathbf{C}_{\Phi\Phi}$ (the geopotential/geopotential correlation) in a way that is slightly more general than necessary for the separable formulation. We define a diagonal $N_v \times N_v$ matrix $\mathbf{D}_{\Phi\Phi}$ with nth (diagonal) element $d_{\Phi\Phi}{}^n$ equal to

$$d_{\Phi\Phi}{}^n = \delta(\phi)\, d_{\psi\psi}{}^n + (1 - \delta(\phi))\, b_{\Phi\Phi}{}^n, \tag{4.6}$$

where $b_{\Phi\Phi}{}^n$ is the spectrum of the vertical background geopotential error at the equator. It may be different than $d_{\psi\psi}{}^n$. ϕ is latitude and $\delta(\phi)$ is a positive function, which in the general case is 1 at the poles and drops to zero at the equator. In the special case where either $\delta(\phi) = 1$ everywhere or $b_{\Phi\Phi}{}^n = d_{\psi\psi}{}^n$ for all n, then $d_{\Phi\Phi}{}^n = d_{\psi\psi}{}^n$ for all n. We define $\mathbf{B}_{\Phi\Phi}$ as the diagonal matrix whose elements are $b_{\Phi\Phi}{}^n$. Then the (O(1)) nondimensional) $N_v \times N_v$ diagonal geopotential background error variance matrix $\mathbf{S}_\Phi$ is given by

$$\mathbf{S}_\Phi = \delta(\Phi)\mathbf{S}_{\Phi 1} + (1 - \delta(\Phi))\mathbf{S}_\Phi, \tag{4.7}$$

with $\mathbf{S}_{\Phi 1} = \mathrm{diag}(\mathbf{S}_\psi{}^{1/2}\mathbf{E}\mathbf{D}_{\Phi\Phi}\mathbf{E}^T\,\mathbf{S}_\psi{}^{1/2}) = \mathbf{S}_\psi$, $\mathbf{S}_{\Phi 2} = \mathrm{diag}[\mathbf{S}_\psi{}^{1/2}\mathbf{E}\mathbf{B}_{\Phi\Phi}\mathbf{E}^T\mathbf{S}_\psi{}^{1/2}]$. The <ΦΦ> vertical correlation is given by $\mathbf{C}_{\Phi\Phi} = \mathbf{S}_\Phi{}^{-1/2}\mathbf{S}_\psi{}^{1/2}\mathbf{E}\,\mathbf{D}_{\Phi\Phi}\,\mathbf{E}^T\,\mathbf{S}_\psi{}^{1/2}\mathbf{S}_\Phi{}^{-1/2}$, where $\mathbf{D}_{\Phi\Phi}$ is the diagonal matrix whose elements are the $d_{\Phi\Phi}{}^n$ defined in (4.6).

$\mathbf{S}_\Phi{}^{-1/2}\mathbf{S}_\psi{}^{1/2}$ is the identity matrix in the separable formulation, but in the more general nonseparable case, its elements may differ from 1 and vary as a function of latitude. **E** is defined in Eq. (4.4). We note that the scaled form of Eqs. (3.17) and (3.18) can be derived using either $\mathbf{S}_\psi$ or $\mathbf{S}_\Phi$ without changing the result, because the scaling (like the preconditioning) is simply a means of improving the conditioning of the problem and does not affect the final result. We have found it more straightforward to scale with $\mathbf{S}_\psi = \mathbf{S}_{\Phi 1}$ rather than $\mathbf{S}_\Phi$ because the latter is latitudinally dependent.

4.3.6 Temperature/temperature and geopotential/temperature correlations

The correlations involving the temperature (strictly speaking, the virtual temperature) are constructed by first building a hydrostatic matrix relating the temperature and geopotential. This is a simple finite-difference formulation of

$$\partial\Phi/\partial\log(P) = -RT/g \text{ , where R,g are the gas and gravitational constants.} \tag{4.8}$$

The hydrostatic relation is applied as a strong constraint. We refer to the $N_v \times N_v$ hydrostatic matrix as $\mathbf{H}_s$. Then if $\mathbf{C}_{\Phi\Phi}$ is the <ΦΦ> correlation and $\mathbf{S}_\Phi$ is the vertical background error geopotential variance defined in Eq. (4.7), then the <ΦT>, <TΦ>, and <TT> correlations are given by

$$\mathbf{C}_{\Phi T} = \mathbf{C}_{\Phi\Phi}\mathbf{S}_\Phi{}^{1/2}\mathbf{H}_s{}^T\,\mathbf{S}_T{}^{-1/2},\ \mathbf{C}_{T\Phi} = \mathbf{C}_{\Phi T}{}^T,\ \mathbf{C}_{TT} = \mathbf{S}_T{}^{-1/2}\mathbf{H}_s\mathbf{S}_\Phi{}^{1/2}\mathbf{C}_{\Phi\Phi}\mathbf{S}_\Phi{}^{1/2}\mathbf{H}_s{}^T\mathbf{S}_T{}^{-1/2}, \tag{4.9}$$

where $\mathbf{S}_T = \delta(\Phi)\mathbf{S}_{T1} + (1-\delta(\Phi))\mathbf{S}_{T2}$ is the diagonal matrix of background temperature error variances, $\mathbf{S}_{T1} = \mathrm{diag}[\mathbf{H}_s\mathbf{S}_\psi{}^{1/2}\,\mathbf{E}\mathbf{D}_{\Phi\Phi}\mathbf{E}^T\,\mathbf{S}_\psi{}^{1/2}\ \mathbf{H}_s{}^T]$, $\mathbf{S}_{T2} = \mathrm{diag}[\mathbf{H}_s\mathbf{S}_\psi{}^{1/2}\,\mathbf{E}\mathbf{B}_{\Phi\Phi}\mathbf{E}^T\mathbf{S}_\psi{}^{1/2}\,\mathbf{H}_s{}^T]$, which are used to ensure that the diagonal elements of $\mathbf{C}_{TT}$ are equal to 1.

The anomaly that occurred at the Earth's surface in the MVOI vertical correlations (Section 4.3.1) has been avoided in this formulation. The temperatures are calculated hydrostatically for the layers above each geopotential level and are assumed to be at intermediate levels, which are determined (for the pressure case), by averaging the Exner function (below 200 hPa) and the logarithm (above 200 hPa) of the two adjacent geopotential levels. (This means we still use an integer vertical temperature index; it just corresponds to different (pressure) values than for the geopotential or winds). A fictitious temperature layer is added above the top pressure level so that $\mathbf{H}_s$ is square N_vx$N_{v.}$ This procedure produces more reasonable <TT>, <ΦT>, and <TΦ> correlations than occurred in MVOI.

Substituting from (4.7) into (4.9) gives

$$\mathbf{C}_{TT} = \mathbf{S}_T^{-1/2}\,\mathbf{H}_s\mathbf{S}_\psi^{1/2}\mathbf{E}\mathbf{D}_{\Phi\Phi}\mathbf{E}^T\mathbf{S}_\psi^{1/2}\mathbf{H}_s^T\,\mathbf{S}_T^{-1/2}$$

$$\mathbf{C}_{\Phi T} = \mathbf{S}_\Phi^{-1/2}\mathbf{S}_\psi^{1/2}\,\mathbf{E}\mathbf{D}_{\Phi\Phi}\mathbf{E}^T\mathbf{S}_\psi^{1/2}\mathbf{H}_s^T\mathbf{S}_T^{-1/2}, \text{ and } \mathbf{C}_{T\Phi} = \mathbf{C}_{\Phi T}^T. \tag{4.10}$$

Figure 4.3 shows the <TT> correlation for a case (separable formulation) in which the vertical scale $L = L_b^v$ varies in the vertical. This figure contains 32 vertical levels unequally spaced in log(P) between 1070 and 10 hPa. The contour intervals are 0.1 and values between –0.1 and 0.1 are "white." The maximum value along the main diagonal is 1. Figure 4.3, shows that the vertical correlation length is a minimum at the surface, increasing to a maximum about 600 hPa and a second minimum at about 200 hPa, and then increases up to 10 hPa.

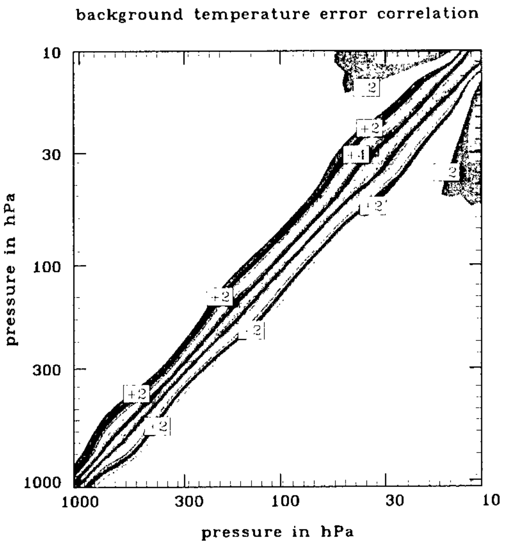


Figure 4.3
<TT> correlation – variable vertical scale

As we did for the <ΦΦ> correlation, we scale with $\mathbf{S}_{TI}$ rather than $\mathbf{S}_T$, in (3.17 or 3.18) because the latter is latitudinally dependent.

4.3.7 The wind/geopotential and wind/temperature correlations

Under separable conditions, we assume that the geopotential and velocity potential are not correlated, that is, $\mathbf{C}_{\Phi\chi} = \mathbf{C}_{\chi\Phi} = 0$. We then define the geopotential/wind vertical background error correlations as

$$\mathbf{C}_{\Phi v} = \mathbf{S}_\Phi^{-1/2}\mathbf{S}_\psi^{1/2}\mathbf{E}\mathbf{D}_{\Phi v}\mathbf{E}^T \text{ and } \mathbf{C}_{v\Phi} = \mathbf{C}_{\Phi v}^T, \tag{4.11}$$

where $\mathbf{D}_{\Phi v}$ is an N_vxN_v diagonal matrix whose nth element is $d_{\Phi v}^n$. We define $d_{\Phi v}^n$ under separable conditions as

$$d_{\Phi v}^n = \mu_v^n d_{\psi\psi}^n, \tag{4.12}$$

where $\mu_v^n = 1$ if completely geostrophically coupled and zero if univariate. Note that μ_v^n is always non-negative. The $\mathbf{C}_{v\Phi}$ and $\mathbf{C}_{v\Phi}$ correlations all involve coupling between wind and mass fields. Lorenc (1981) introduced a geostrophic coupling parameter μ, that varied with latitude, varying between 1 at high latitudes and 0 (no coupling) in the tropics. We specify that μ_v^n can vary with the vertical eigenmodes of background error correlation. In particular, following the ideas of normal mode initialization (Daley, 1991, chapters 9-10), we might specify that the gravest vertical modes (deep modes) are strongly geostrophically coupled and the shallow modes (small eigenvalues, many zero crossings) are weakly geostrophically coupled. We do not consider here the horizontal variation of the geostrophic coupling parameter; this is discussed in Section 4.6.

The wind/temperature correlations are defined by operating on (4.11) with the hydrostatic operator

$$\mathbf{C}_{Tv} = \mathbf{S}_T^{-1/2}\mathbf{H}_s\mathbf{S}_\Phi^{1/2}\mathbf{C}_{\Phi v} \text{ and } \mathbf{C}_{vT} = \mathbf{C}_{Tv}^T. \tag{4.13}$$

Note that although **E** are the eigenvectors of $\mathbf{C}_{\psi\psi}$ and $\mathbf{C}_{vv}$ in the separable case, that $\mathbf{S}_T^{-1/2}\,\mathbf{H}_s\mathbf{S}_\psi^{1/2}\mathbf{E}$ are not usually the eigenvectors of $\mathbf{C}_{TT}$. (They may not be orthogonal.) However, we refer to $\mathbf{S}_T^{-1/2}\mathbf{H}_s\mathbf{S}_\psi^{1/2}\,\mathbf{E}$ and $\mathbf{S}_\Phi^{1/2}\mathbf{S}_\psi^{1/2}\mathbf{E}$ as modified eigenvectors appropriate for the temperature and geopotential, respectively.

Figure 4.4 is plotted in the same format as Fig. 4.3, the <Φv> correlation for a case in which $\mu_v^n = 1$ for the gravest vertical mode and then falls off gradually to zero for the shallowest modes. The correlation model is SOAR, and the vertical correlation length and pressure levels are exactly as in Fig. 4.3. The maximum correlation is less than 1, as noted, and the correlation is broader than for the corresponding <ΦΦ> correlation (not shown). The effective vertical scale for the geostrophic coupling scale has increased (because the shallower vertical modes are only weakly coupled).

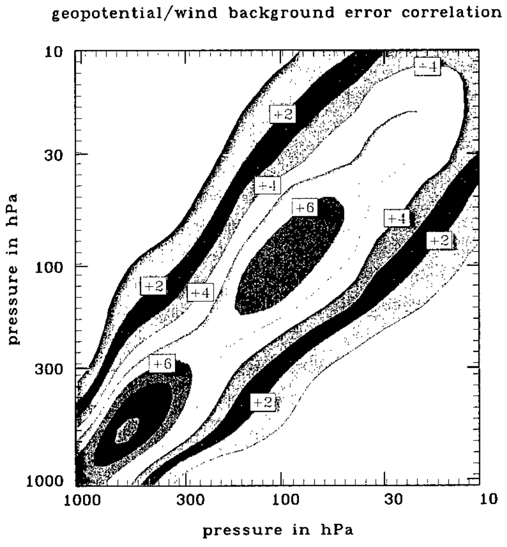


Figure 4.4
<Φv> correlation – vertical geostrophic coupling

Under nonseparable conditions, Eqs. (4.11)-(4.13) may contain extra terms. If it were assumed that the divergent and rotational winds were correlated, then this would also imply that the divergent wind was correlated with the geopotential and temperature.

4.3.8 Univariate correlations (moisture and ozone)

We also have provision for other vertical background error correlations that are not multivariately correlated. At this time, moisture and ozone are the only such variables that have been considered.

For moisture, we assume that the basic moisture variable is $s = \log_e(q)$, where q is the specific humidity. Thus, if the errors in s are normally distributed, then the errors in q obey a lognormal distribution, which is reasonable for atmospheric constituents such as moisture. Additive errors in s are multiplicative errors in q. Then, if we define background values $q_b = \exp(s_b)$, and differences Δ q $= q - q_b$ and $\Delta s = s - s_b$, then to first order $\Delta q = q_b \Delta s$. Thus, we assume that the background error covariance for moisture is defined in terms of Δ s.

For moisture, vertical correlation $\mathbf{C}_{ss}$ adopts the same basic functional form (SOAR or Gaussian) as the <ψψ> correlation. However, the pressure levels for this correlation are the intermediate pressure levels used in the <TT> correlation, rather than the geopotential/wind pressure levels. The vertical correlation lengths L_n, may well be different than those used for geopotential, temperature, and wind, giving rise to both different eigenvectors and different eigenvalues. The vertical background error variance $\mathbf{S}_s$ must be specified.

Although there is provision for ozone in the code, the vertical correlation has not yet been modified beyond using the same formulation as for the streamfunction correlations.

4.4 Projection onto the Vertical Eigenvectors of the Background Error Correlation

The formulation (4.4)-(4.13) offers distinct computational advantages and also facilitates the inclusion of a forward operator **H** in the direct assimilation of radiances. We now discuss a procedure that is useful in two aspects of the calculation. Equations (3.5)-(3.6) break the algorithm into two steps, a solver (3.5) and a post-multiplication (3.6).

4.4.1 Matrix/vector operations in the solver and post-multiplier

All of the solvers (steepest descent (3.11), standard conjugate gradient (3.12), and preconditioned conjugate gradient (3.13)) contain an equation of the form $\mathbf{q}_k = \mathbf{A}\mathbf{p}_k$ that is a matrix/vector multiplication involving the covariances or correlations between every observation. When the observed and analyzed/forecast variables are the same, the form of the matrix **A** in this algorithm is given by Eq. (3.18), viz. $\mathbf{A} = \mathbf{C}_b^{ob/ob} + [\mathbf{S}_b^{ob}]^{-1/2}\mathbf{R}[\mathbf{S}_b^{ob}]^{-1/2}$. We can then write $\mathbf{q}_k = \mathbf{C}_b^{ob/ob}\mathbf{p}_k + [\mathbf{S}_b^{ob}]^{-1/2}\mathbf{R}[\mathbf{S}_b^{ob}]^{-1/2}\mathbf{p}_k$ In the solvers, it is this operation (which is the order of the square of the number of observations) that must be performed every iteration and that takes up the largest portion of the solver computation time. We concern ourselves here with the background error covariance and concentrate on the operation $\mathbf{C}_b^{ob/ob}\mathbf{p}_k$.

As noted above, a similar matrix/vector operation occurs in the post-multiplication step (3.6). In Eq. (3.18), this would involve multiplication by the matrix $\mathbf{C}_b^{gr/ob}$, which is a background error correlation matrix of order the number of observations by the number of grid points. Since the number of grid points may well exceed the number of observations, this matrix/vector operation is also very expensive, even if it is done only once.

4.4.2 Vertical eigenvector decomposition for profiles (separable examples)

Both of these operations are matrix/vector operations

$$\mathbf{q} = \mathbf{C}\mathbf{r}, \tag{4.14}$$

where **C** is a background error correlation matrix and **q**,**r** are vectors. The following method is used to make these expensive operations much less computer-intensive.

We first begin with a simple separable, univariate example. Suppose that **r** is a vertical profile of length M with horizontal location (λ_1,θ_1) and vertical locations z_1^m, $1 \le m \le M$, and **q** is a second vertical profile of length K with horizontal location (λ_2,θ_2) and vertical locations z_2^k, $1 \le k \le K$. Then, the background error correlation between any two locations (m,k) in these profiles is (following the notation in (4.1)),

$$c_b(\lambda_1,\theta_1,z_1^m,\lambda_2,\theta_2,z_2^k) = c_b^h(s_{12},\alpha_{12})\, c_b^v(z_1^m,z_2^k), \tag{4.15}$$

where s_{12} and α_{12} are the great-circle distance and angle between the two profiles. Now define **C** to be the M×K forecast error correlation matrix with elements given by (4.15), and consider the matrix/vector operation (4.14). Following (4.4), suppose there are N_v eigenvalues and eigenvectors of the forecast error correlation and define $\mathbf{E}_1$ as the $M \times N_v$ eigenvector matrix corresponding to profile (1) and $\mathbf{E}_2$ as the $K \times N_v$ eigenvector matrix for profile (2). Define $\mathbf{D}_v$ as the diagonal $N_v \times N_v$ matrix of the vertical eigenvalues of the forecast error correlation. Then, we can write (4.14) as $\mathbf{q} = c_b(s_{12},\alpha_{12})\,\mathbf{E}_2\mathbf{D}_v\mathbf{E}_1^T\,\mathbf{r}$ since $c_b(s_{12},\alpha_{12})$ is independent of the vertical coordinate. Now define $\mathbf{D}_{12} = c_b^h(s_{12},\alpha_{12})\,\mathbf{D}_v$ as a diagonal $N_v \times N_v$ matrix so that each (diagonal) element is a function of the horizontal locations of the two profiles and the vertical mode number. Then, we re-write (4.14) as

$$\mathbf{q} = \mathbf{E}_2\mathbf{D}_{12}\mathbf{E}_1^T\,\mathbf{r}. \tag{4.16}$$

Note that while we refer to the matrices $\mathbf{E}_1$ and $\mathbf{E}_2$ as eigenvector matrices, we use this term rather loosely. In Eqs. (4.4)-(4.13) they are not always eigenvectors, especially for nonseparable formulations. However, this does not really matter because we do not intend to take any advantage of the orthogonality properties of the eigenvector matrices. The only property that is of interest to us is that the background error correlations can be written as the product of a diagonal matrix, with premultiplication by a rectangular matrix and post-multiplication by its transpose.

Now, the representation of (4.16) in itself does not get us very far. But now consider a slightly bigger problem. Suppose that the vector **r** = $[\mathbf{r}_1\ \mathbf{r}_2]^T$ is a vector of length M_1+M_2 consisting of the observation profile vector $\mathbf{r}_1$ of length M_1 at horizontal location (λ_1^1,θ_1^1) and observation profile vector $\mathbf{r}_2$ of length M_2 at horizontal location (λ_2^1,θ_2^1). Similarly, define **q** = $[\mathbf{q}_1, \mathbf{q}_2]^T$ as another vector of length K_1+K_2 consisting of the observation profile vector $\mathbf{q}_1$ of length K_1 at horizontal location $(1_1^2,q_1^2)$, and observation profile vector $\mathbf{q}_2$ of length K_2 at horizontal location (λ_2^2,θ_2^2). Then, consider the matrix/vector multiply problem in this case. Matrix **C** in (4.14) in this case is now an $(M_1+M_2)\times(K_1+K_2)$ forecast error correlation matrix, and following (4.16) we can write (4.14) as

$$\begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} E_1 & 0 \\ 0 & E_2 \end{bmatrix} \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} E_1^T & 0 \\ 0 & E_2^T \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \tag{4.17}$$

Now the D matrix consists of four diagonal blocks, while the two E matrices are block-diagonal. All horizontal correlations are contained in the diagonal D matrix. We can represent (4.17) as a sequence of three operations.

(1) Multiply the vector $\mathbf{r}_1$ by the matrix $\mathbf{E}_1^T$ and the vector $\mathbf{r}_2$ by the matrix $\mathbf{E}_2^T$. This operation does not involve interactions between the two profiles and will produce a new vector of length $2\times N_v$. This operation takes the real space vertical profiles and projects them into vertical eigenvector space.

(2) Multiply the vector resulting from operation (1) by the diagonal matrices $\mathbf{D}_{11}$,$\mathbf{D}_{12}$,$\mathbf{D}_{21}$,$\mathbf{D}_{22}$ to produce another vector of length $2\times N_v$. This operation is where all the interactions between the two profiles occur, but the matrix/vector multiplications are all performed with diagonal matrices.

(3) Multiply the two vectors of length N_v resulting from operation (2) by $\mathbf{E}_1$ and $\mathbf{E}_2$ respectively to produce the vectors $\mathbf{q}_1$ of length M and $\mathbf{q}_2$ of length K. This operation also does not involve interactions between the two profiles and is essentially the back transformation from eigenvector to real space.

4.4.3 Operation count for real space and eigenvector methods

Now we consider an operation count for the full problem. Suppose the number of eigenvectors and the number of elements in each profile is the same and is equal to N. Suppose instead of two profiles there are J profiles. Typically J might be several hundred or thousand and N might of order 50 to 100. Then, if the problem (4.14) is done totally in real space, the operation count is $(N\times J)^2$. For the eigenvector decomposition (4.17), the first step requires $J\times N^2$ operations, the second $N\times J^2$,and the third step $J\times N^2$, for a total of $2J\times N^2 + N\times J^2$ operations. Under realistic conditions N << J, and the cost of the second step dominates the first and third step, and total operation cost for the eigenvector decomposition method is approximately $N\times J^2$ operations, or a factor of 1/N fewer operations than the real space method. This factor of N, which is order 30 to 100, is not negligible in either storage or execution time and is definitely worth pursuing.

4.4.4 The effect of single level observations

Now all observations are not profiles-what about single-level observations? Suppose **r** and **q** are vectors of single-level observations of length I. If we consider the interactions between I single-level observations and J profiles (of length N), then the traditional real space approach takes I×N×J operations, while the eigenvector approach takes $J\times N^2$ + I×N×J + I×N operations. For N << J and N<< I, the real space and eigenvector approaches have essentially the same operation count for single-level/profile interactions.

Now consider the interactions between I single-level observations only. Using the traditional real space approach, the operation count is I^2, while for the eigenvector approach it can easily be shown that the operation count is 2IxN + NxI^2. For N << I, the eigenvector approach is N times as expensive as the real space approach when considering single-level/single level interactions. This obviously is not good.

4.4.5 A solution for the single-level observation problem for solver and post-multiplier

From the above operation count, it is clearly not desirable to project each single-level observation onto N_v vertical eigenvectors. Consequently, any code must be modified to handle single-level operations separately. We separately discuss the matrix/ vector operation that occurs in the solver and the post-multiplication. Consider first the solver problem. Here, we have to perform an operation (4.14) in which the both the vectors **r** and **q** consist of mixtures of profiles and single-level observations. Thus there will be four types of interactions:

(1) **r** and **q** are both profiles - this is the case considered in (4.17)

(2) **r** and **q** are both single-level - this is done entirely in real space

(3) **r** is a profile, **q** is single-level - in this case **r** is projected into eigenvector space using a multiplication by $\mathbf{E}^T$ (first step of (4.17)). The second step (multiplication by **D** matrices in (4.17)) is the same except that the elements of **D** are modified. Thus, the n_vth element of **D** has been multiplied by $e(n_{v,},z)$, which is the element of **E** corresponding to the n_vth eigenvalue at the location of the single-level observation in **q**. The final step (multiplication by **E** in (4.17)) is then not necessary.

(4) **r** is a single-level, **q** is a profile - this is like (3), except in reverse.

If the observations are properly sorted into profiles and single-level observations, these four cases can all be reduced to matrix/vector operations in an orderly fashion.

For the post-multiplication, the procedure is slightly simpler. In this case, the vector **r** in (4.14) would consist of a mixture of profiles and single-level observations, but the vector **q** would consist only of profiles. This is because the output from an analysis scheme is always a vertical column of corrections at each horizontal location of the analysis grid. In this case, only operations of type (1) and (4) are required.

4.4.6 Operator form

In Eqs. (4.16)-(4.17), the transformation from vertical eigenvector space to real space **E** and the transform from real space to eigenvector space $\mathbf{E}^T$ (and the necessary modifications to take care of the single-level case) are written as matrices. In preparation for the implementation of the forward operator **H**, these operations are actually coded as operators. Thus, if we have an operation such as (4.16) to perform, viz, $\mathbf{q} = \mathbf{EDE}^T\,\mathbf{r}$, where **r** and **q** are vectors, **E** and $\mathbf{E}^T$ are block-diagonal eigenvector transformation matrices as in (4.17), and **D** consists of diagonal blocks as in (4.17), the actual operation would take the form

$$\mathbf{q} = \mathrm{E}(\mathbf{f}),\ \mathbf{f} = \mathbf{Dg},\ \mathbf{g} = \mathrm{E}^T(\mathbf{r}), \tag{4.18}$$

where E and E^T are operators and **g**, **f** are intermediate vectors. In these operators, the single-level case is explicitly accounted for.

4.4.7 Application of vertical eigenvector decomposition to the pre-conditioner

Vertical eigenvector decomposition can also be applied to the preconditioner of (3.13), that is,

$$\mathbf{A}^*\mathbf{s}_k = \mathbf{r}_k,\ \text{where } \mathbf{A}^* = \mathbf{C}_b^{ob/ob} + [\mathbf{S}_b^{ob}]^{-1}\,\mathbf{R}[\mathbf{S}_b^{ob}]^{-1} \text{ is the preconditioner matrix.} \tag{4.19}$$

In this case, we must solve for $\mathbf{s}_k$ given $\mathbf{r}_k$ separately for each observation volume. This problem may be solved (1) directly, by Cholesky decomposition, or (2) iteratively, by using the standard conjugate gradient algorithm (3.12). If we use the eigenvector decomposition, we can, in principle, apply either method.

The method that we have actually implemented is method (1), the Cholesky decomposition. To implement the eigenvector transform in this algorithm is straightforward. The calculation of $[\mathbf{S}_b]^{-1}\mathbf{R}[\mathbf{S}_b]^{-1}$ is unchanged; it is only the calculation of $\mathbf{C}_b^{ob/ob}$ in (4.19) that has to be changed. In this case, we must use the E and E^T operators of (4.18) to generate a matrix. This can be done by defining a vector **r** in (4.18), which is zero for all elements except the jth element, which is set equal to 1. Then, application of the E^T operator, multiplication by the **D** matrix, and application of the E operator (as in (4.18)), will produce a vector that is actually the jth column of the $\mathbf{C}_b^{ob/ob}$ matrix. We can exploit the fact that the matrix $\mathbf{C}_b^{ob/ob}$ (for the preconditioner) is always symmetric.

4.4.8 Inclusion of the forward operator

We defer this discussion until the subject of direct assimilation of radiances is introduced in Section 5.

4.4.9 Modification for nonseparable background error correlations

Section 4.4.2 considered separable background error covariances, as in Eq. (4.15). However, certain types of nonseparability can be accommodated within this formulation. We define d_v^n, d_{12}^n as the nth elements of the N_vxN_v diagonal matrices $\mathbf{D}_v$ and $\mathbf{D}_{12}$ respectively (4.16). For the separable formulation,

$$d_{12}^n = c_b^h(s_{12},\alpha_{12})d_v^n. \tag{4.20}$$

For a nonseparable formulation that can be accommodated in this framework, we define a horizontal correlation $c_b^h(s_{12},\alpha_{12},n)$ that is vertically mode-dependent. Then, $d_{12}^n = c_b^h(s_{12},\alpha_{12},n)d_v^n$. In this case, all the features of the eigenvector decomposition discussed in this section are still relevant. Of course, it is more expensive when the horizontal correlation varies by vertical mode number (as it would be if the horizontal correlations varied by pressure) because the horizontal correlation operator has to be called more frequently. We discuss nonseparable background error covariances in greater detail in Section 4.7.

4.5 Changing the Vertical Metric (Isentropic Coordinates)

Equation (4.3) gives several models for the vertical structure of the background error correlation at two levels, m and n, with local vertical correlation lengths L_m and L_n. These models are given in terms of $|\Delta \mathbf{z}|$, where z = P is the pressure.

It is not necessary that z = P; in fact, another attractive possibility is z = θ, where θ is a potential temperature. That is, we apply the models (4.3) based on the differences in potential temperature rather than the differences in pressure between the two levels. (We could also alter the values of L_b^v at the levels). Now $\theta = T(P/P_0)^{R/C_P}$, where T is the temperature, P the pressure, P_0 the pressure at 1000 mb , R the gas constant, and C_P the specific heat at constant pressure. Everything except the temperature will always be known. Because it is the background error correlation that we wish to model, it is perfectly legitimate to determine the temperature (and therefore the potential temperature) at any desired location from the background field itself. That is, $z = \theta_b$, the background potential temperature. It is important to note that this procedure does not require a transformation to isentropic coordinates and there are none of the well-known problems caused by the intersection of the isentropes with the Earth's surface.

If we assume an isotropic horizontal background error correlation model and assume $z = \theta_b$, then we are implicitly assuming that the background error correlations are isotropic along isentropic surfaces. Whether this is a better or worse assumption than assuming the correlations are isotropic on pressure surfaces is not known. However, the assumption of isotropy on an isentropic surface does make the forecast error correlations considerably more flow-dependent. This can be seen in Figs. 7 and 8 of Benjamin (1989), which developed an optimal interpolation scheme in isentropic coordinates for initializing an isentropic forecast model.

Figures 4.5 and 4.6 illustrate the effect of setting $z = \theta_b$. Each panel is a two-dimensional map (abscissa is horizontal distance; ordinate is equally spaced in pressure). The isentropic fields θ_b are plotted on each panel as line contours (every 5° K). The tropopause and stratosphere are visible at the top of each panel, an area of neutral stability at the lower right, and a frontal structure running diagonally from the lower left-hand corner. On each of the nine panels of the two figures, shaded contours mark the height/height <ΦΦ>, temperature/temperature <TT>, wind/wind <vv> correlations and all the cross correlations for a point located in the front, just below the tropopause. Figure 4.5 defines the vertical coordinate in Eq. (4.1) as z = P and Fig. 4.6 is for $z = \theta_b$. From the <TT> correlations, it is clear that along-front points are much more highly correlated than across-front points, and statically neutral points are much more highly correlated than statically stable points. Not shown is the jet core (it sits slightly to the left of center of the <ΦΦ> correlation), but examination of the correlations indicates quite different behavior on one side of the jet than the other. Essentially, correlations drop off rapidly when they have to cross-isentropic contours. At the lower left of each panel is a marine boundary layer; at the lower right is an almost unstable continental air mass. Because isentropic contours lie between, it is likely that observations in the marine boundary layer would not influence the analysis over the continents and vice versa. The $z = \theta_b$ system is also able to "see" the tropopause and the different stratification in the stratosphere.

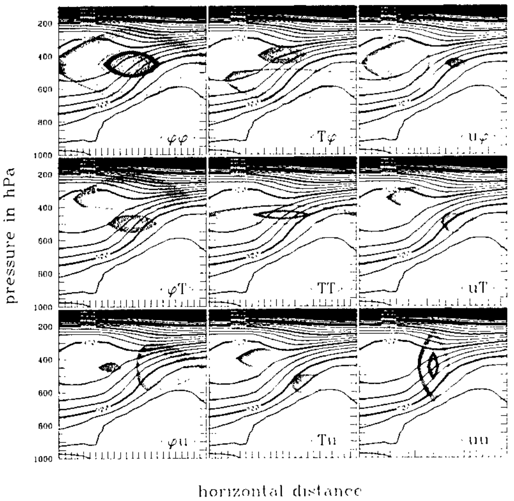


Figure 4.5
x/P cross-section of 9 correlations – pressure as vertical coordinate

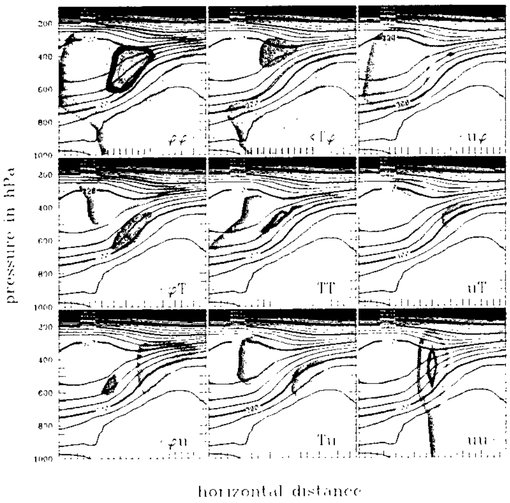


Figure 4.6
Same as Figure 4.5, except for θ as vertical coordinate

There are four further notes concerning this transformation.

(1) $z = \theta_b$ requires monotonicity; if the background is statically unstable, it is corrected to be neutral. This was, in fact, done in Fig. 4.6;

(2) the value of the transformation $z = \theta_b$, depends on the accuracy of θ_b. If the background field is not very accurate, then z = P is likely to be a more useful form;

(3) while it may be useful to define the correlations along θ surfaces, it would seem desirable that the variances be defined on pressure surfaces; and

(4) this transformation is more likely to be useful for regional problems, because, in the global problem, certain isentropes that are present in the tropics may be missing in the polar regions, and vice versa.

This transformation was coded for NAVDAS and tested over North America in the COAMPS system by Peter Steinle of the Bureau of Meteorology Research Center, Melbourne, Australia.

4.6 Horizontal Background Error Correlations

By and large, the horizontal background error correlations are much less radically changed from the NRL MVOI system than are the vertical background error correlations. The NRL MVOI formulation is essentially based on Lonnberg and Hollingsworth (1986) and Daley (1991), as is the present NAVDAS formulation. However, some generalizations are introduced here: horizontal scales, which vary with horizontal position; geostrophic coupling parameters, which are horizontally scale-dependent; and background error correlations, in which the horizontal and vertical representations are nonseparable. We begin the discussion with the univariate case.

4.6.1 The univariate case

The horizontal univariate correlation between a location with longitude and latitude (λ_n,θ_n) and a second location with longitude and latitude (λ_m,θ_m) is of the form

$$c_b^h(\lambda_n,\theta_n,\lambda_m,\theta_m) = c_b^h(s_{nm},L^h_{nm}), \tag{4.21}$$

where s_{nm} is the great-circle distance between the two points and $L^h_{nm} = (L_n^h L_m^h)^{1/2}$. Here, L_n^h and L_m^h are the horizontal correlation lengths at the two horizontal locations. If $L_n^h = L_m^h$, then a correlation of the form (4.21) will be isotropic. If the correlation lengths vary horizontally, then the correlation (4.21) will be anisotropic.

It is important to note that allowing L^h to vary horizontally has implications for the multivariate case somewhat similar to the horizontal variation of the background error variance discussed earlier. That is, if we want to calculate the wind/wind and wind/geopotential covariances from the geopotential/geopotential covariances using the geostrophic relation, we will create an extra term involving the horizontal derivatives of the variation of L^h. These extra terms will remain small as long as the horizontal variation of L^h is on scales that are large compared to L^h itself.

4.6.2 Horizontal correlation models

At present, we can use the following horizontal correlation models:

(1) SOAR $\quad c_b^h(s_{nm},L^h_{nm}) = (1 + s_{nm}/L^h_{nm}) \exp(-s_{nm}/L^h_{nm})$

(2) Gaussian $\quad c_b^h(s_{nm},L^h_{nm}) = \exp[-(s_{nm}/L^h_{nm})^2]. \qquad (4.22)$

(3) Compact spline (Gaspari and Cohn, 1999). This is a function of compact support that goes identically to zero (with its first two derivatives) at some finite distance (2c). It is positive definite and twice differentiable. Define $c = (10/3)^{1/2}$ and $r = s_{nm}/(cL_{nm})$. Then,

$$\begin{aligned} c_b^h(s_{nm},L_{nm}) = c_b^h(r) &= -r^5/4 + r^4/2 + 5r^3/8 - 5r^2/3 + 1, && 0 \le r \le 1, \\ &= r^5/12 - r^4/2 + 5r^3/8 + 5r^2/3 - 5r + 4 - 2/3r, && 1 < r \le 2, \\ &= 0. && r > 2. \end{aligned}$$

These models are not guaranteed to yield positive-definite matrices when L^h is horizontally variable. In practice, however, if L_h varies on scales larger than L_h itself, there do not seem to be problems. The great-circle distance is calculated by a procedure similar to but more efficient than that used in NRL MVOI. This procedure is discussed in more detail in Appendix A. The radial correlations for the SOAR and compact spline formulations are displayed in Figs. A1 and A2.

The Schur product

The Schur or Hadamard product of two matrices **A** and **B** having the same dimensions is the matrix **C** of the same dimensions with $c_{ij} = a_{ij} b_{ij}$. It can be shown that the Schur product of two covariances is also a covariance. In particular, if **A** and **B** are both positive definite, then the Schur product of the two matrices is also positive definite. As described in Gaspari and Cohn (1999), this gives the possibility of taking the Schur product of any covariance with another covariance constructed from a function of compact support. Thus, we could Schur multiply a SOAR covariance matrix with a compact spline covariance to yield a new covariance that was SOAR-like, and yet went to zero smoothly at some finite distance. This Schur product covariance is useful when we wish to ignore correlations between observations that are widely separated. (This is preferable to ignoring small but nonzero correlations).

4.6.3 An example of anisotropy

There is good evidence that the horizontal length scale L^h is larger in the tropics than in the extratropics, and this type of variation can be easily introduced.

We now discuss another example of anisotropy. The NRL mesoscale model COAMPS is often run on a triply nested grid, with maximum resolution at the innermost grid. One way to provide analyses for this system is to provide three separate analyses-one for each grid. Since the resolvable scales are finer in the innermost grid, it seems reasonable to suppose that the horizontal correlation length L^h might be smaller for this grid than for the coarser outer grids. This sort of variation can be easily accommodated with three separate analyses. However, with NAVDAS and horizontally variable L^h, the same result can be achieved with a single analysis in which the horizontal correlation length varies smoothly across the interfaces, even though the analysis grid length jumps discontinuously. (Strictly speaking, there would be a single solve (Eq. 3.5), but a separate post-multiply (Eq. 3.6) for each of the nests.)

Figures 4.7 and 4.8 demonstrate this capability in a univariate (geopotential) analysis of geopotential observations on a two-dimensional grid. The grid is 40×40 (equally-spaced, not varying discontinuously, as would be the case with a COAMPS analysis). There are 1600 observations randomly scattered within the domain, and each observation is generated with a random number generator. L^h is constant around the boundaries of the domain and decreases by a factor of 4 smoothly toward the center. The correlation function is of the SOAR form (4.22) and is plotted for a point near the upper left-hand corner of the domain in Fig. 4.7. The contour interval is 0.1, and all values less than 0.1 are "white." It can be seen that the correlation is stretched toward the outside of the domain and is shrunk toward the inside, as one would expect using the prescribed variation of the correlation length L^h. (It might be noted that here L^h varies on scales that are not large compared to L^h itself, in violation of the warning in Section (4.6.1), but this is only intended to be a demonstration).

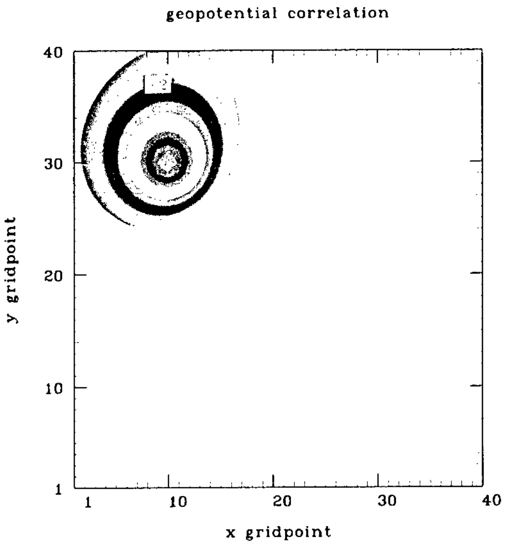


Figure 4.7
<ΦΦ> correlation for variable horizontal scale

Figure 4.8 shows two resulting geopotential correction fields. In panel (a), the horizontal scale is invariant; in panel (b), it decreases toward the center of the domain as noted above. The contour interval is 2.0, and all values between –2 and +2 are "white." At the domain boundaries, the horizontal scale is the same for the two plots. Consequently, near the boundaries, the two panels are very similar, resolving only large-scale features. Toward the center of the domain, panel (b) shows much more small-scale detail, as would be expected. In fact, the correction field in the center of panel (b) is very similar to that obtained in the case where a constant horizontal scale L^h is set equal to the minimum value used in constructing Fig. 4.8(b).

The univariate formulation (4.22) is used for all variables except geopotentials, winds, and temperatures, which are related multivariately.

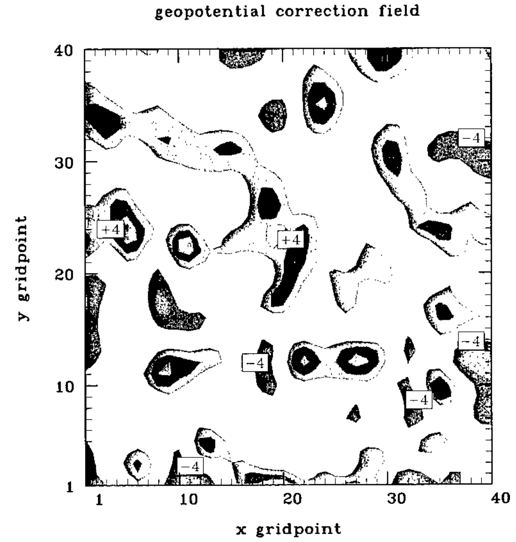


Figure 4.8a
Φ correlation for invariant horizontal scale

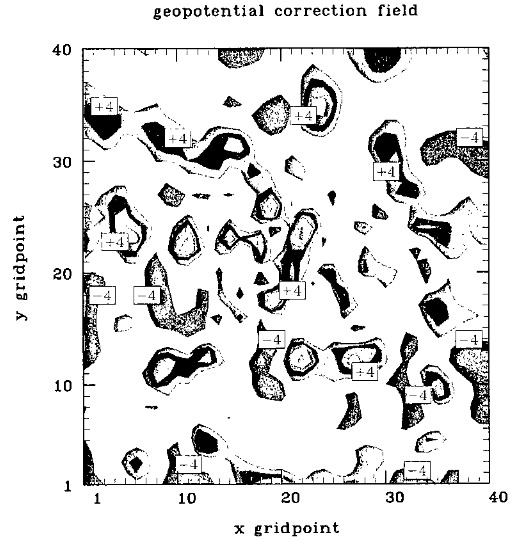


Figure 4.8b
Effect of variable horizontal scale

4.6.4 The multivariate case

The formulation of the multivariate horizontal background error correlation basically follows Daley (1991, Chapter 5), as does the NRL MVOI. Consequently, we discuss this aspect (important as it is) relatively briefly. Because the horizontal correlation length L^h depends on location, it is convenient to introduce a nondimensional scaled distance. Thus,

$$c_b^h(s_{nm},L^h_{nm}) = c_b^h(r_{nm}), \text{ where } r_{nm} = s_{nm}/L^h_{nm}. \qquad (4.23)$$

The multivariate correlations must be formulated on the sphere. This introduces some complication over formulation in Cartesian geometry and is discussed at some length in Appendix B. Figure B2 shows (for the case of no correlations with the divergent wind) the nine correlations involving the geopotential and the two wind components. We now discuss the wind/wind correlations.

4.6.5 The wind/wind correlations

We discuss first the separable case in which the divergent wind is not correlated with the rotational wind. This case is discussed in detail in Daley (1991, Section 5.2). The representation requires the calculation of the derivatives $1/r\ dc_b^h/dr$ and $d^2c_b^h/dr^2$, the angles between the two locations α_{nm} and α_{mn}, and a parameter ν that is a measure of the divergence permitted in the wind correlations. The parameter $\nu = 0$ is strictly nondivergent, and normally ν is set to a value such as 0.05 or 0.10, which produces correction vectors that may be weakly divergent. ν is assumed to be set to a global value. For the wind/wind correlations, there are actually two horizontal correlations-a streamfunction/ streamfunction correlation ($c_{\psi\psi}{}^{h}$) and a velocity potential/velocity potential correlation ($c_{\chi\chi}{}^{h}$). We assume that these two correlations have the same horizontal correlation length, although this is not strictly necessary. The horizontal wind/wind correlations are derived from these two univariate correlations using Eqs. (5.2.23)-(5.2.32) of Daley, 1991).

The calculation of the angles α_{nm} and α_{mn} is considerably more complex in NAVDAS than in NRL MVOI. In NRL MVOI, the analysis is always done in a local grid because correlations between locations that are widely separated are ignored. In NAVDAS, correlations between locations up to 6000 km apart are considered. This means that wind/wind correlations must be done properly in spherical geometry, and the rather delicate situations around the poles must be properly accounted for. This subject is discussed in considerable detail in Appendix B.

The nonseparable case contains extra terms because the divergent wind may be correlated with the rotational wind. This case is discussed in Daley (1985) and the appropriate spherical form of the equations are also given in Appendix B.

4.6.6 Geopotential/wind correlations

We first discuss the separable case where the divergent wind is not correlated with the mass field. The formulation of geopotential/wind correlations follows Daley (1991, Section 5.3). This representation requires calculation of the derivative $dc_h{}^h/dr$, the angles α_{nm} and α_{mn} as above, and a parameter μ that specifies the strength of the geostrophic coupling. Calculation of the derivatives and angles is as above.

It has already been indicated (Section 4.3.7) that the geostrophic coupling may be vertical mode dependent. It is much more important that the geostrophic coupling be latitudinally dependent. Ignoring vertical modal variations, $\mu = 1$ is completely geostrophically coupled in the Northern Hemisphere, and $\mu = -1$ is completely geostrophically coupled in the Southern Hemisphere. Generally, μ is set to a number between 0.9 and 1.0 in the northern extratropics, falling to zero at the equator, and then decreasing to a value between –0.9 and –1.0 in the southern extratropics. In the NAVDAS code, it is technically possible to have correlations between points in different hemispheres, for example points at 20°N and 20°S may have nonzero correlations and nonzero values of m. This situation is handled as follows: Suppose we wish to find the correlation for a wind at point n, with a geopotential at some other point m. Define μ_n and μ_m as the values of the coupling parameter at the two points. If μ_m and μ_n are both positive, the coupling parameter would be $(\mu_m\mu_n)^{1/2}$; if both are negative, it would be -$(\mu_m\mu_n)^{1/2}$; otherwise, it would be zero.

In an earlier section, we showed how μ can also vary in the vertical. That is μ is close to 1 (or -1 depending on the hemisphere) for grave vertical modes, but μ becomes closer and closer to zero for the shallow vertical modes. In other words, it is only the deep vertical modes that are highly geostrophically coupled. It is possible to make the same argument in the horizontal, that is that the geostrophic coupling should be a maximum at large horizontal scales and the smaller horizontal scales (below meso α, say) should be increasingly uncoupled, because geostrophy is not relevant on those scales. This geostrophic decoupling at smaller spatial scales would be particularly relevant for the inner mesh of COAMPS. A simple procedure for geostrophic decoupling at smaller horizontal scales is discussed in Appendix F.

In the nonseparable case, there may be correlations between the mass field and the divergent wind field. This case is discussed in Daley (1985), and the appropriate spherical form of the equations is given in Appendix B. This completes our discussion of the horizontal background error correlations; we now consider the combined horizontal/vertical correlations in a nonseparable formulation.

4.7 Nonseparable Background Error Correlations

Equation (4.1) describes a vertically/horizontally separable background error correlation in which the characteristic vertical scale is, at most, vertically dependent and the horizontal scales are, at most, horizontally dependent. A nonseparable generalization of (4.1) is written as

$$c_b(\lambda_1,\theta_1,z_1,\lambda_2,\theta_2,z_2) = c_b^v(z_1,z_2,L_b^v)\ c_b^h(s_{12},\alpha_{12},L_b^h), \qquad (4.24)$$

where $L_b^v = L_b^v(z_1,z_2,\lambda_1,\lambda_2,\theta_1,\theta_2)$ and $L_b^h = L_b^h(\lambda_1,\lambda_2,\theta_1,\theta_2,z_1,z_2)$. We now discuss briefly three possibilities for nonseparability.

4.7.1 Horizontal variation of the vertical correlations

Bouttier et al. (1997) provides some evidence that the background error correlations have a smaller vertical scale (whiter spectrum) in the tropics than in the extratropics. This sort of latitudinal variation of the vertical correlations can be accommodated within the present formulation. While latitudinal variation of the vertical eigenvectors **E** (4.4) is not out of the question, a more straightforward idea is to keep the **E** invariant and allow the eigenvalues to vary latitudinally. This idea can be illustrated most simply by considering the <ΦΦ> correlation since the more general notation has already been introduced in Eq. (4.6). Thus, we suppose that the tropical vertical geopotential background error correlation is given by $b_{\Phi\Phi}^n$, which differs from $d_{\psi\psi}^n$. There is one drawback to this assumption. That is, given the eigenvectors **E** defined in (4.4), the only diagonal matrix $\mathbf{D}_{\Phi\Phi}$ that will give a correlation $\mathbf{C}_{\Phi\Phi}$, whose main diagonal elements are all equal to 1, is any linear combination of $\mathbf{D}_{\psi\psi}$ and the identity matrix, whose trace is equal to N_v. Such a formulation is very limiting (allowing only relative adjustment between the diagonal and all the off-diagonal elements simultaneously). To allow more general formulations, we have adopted the form (4.6), but it requires the multiplication of $\mathbf{S}_\Phi^{-1/2}\mathbf{S}_v^{1/2}$ in (4.7) to produce a proper correlation (all main diagonal elements equal to 1). This multiplication can potentially produce very noisy correlations if $\mathbf{S}_\Phi^{-1/2}\mathbf{S}_v^{1/2}$ is very different from the identity matrix. Consequently, the spectra $b_{\Phi\Phi}^n$ and $d_{\Phi\Phi}^n$ in (4.6) should not differ markedly. (As noted in Sections 4.3.5 and 4.3.6, the actual scaling of equations (3.17) and (3.18) is performed by $\mathbf{S}_{\Phi 1}$ for the geopotential and by $\mathbf{S}_{T1}$ for the temperature.)

Figure 4.9 is an example of a vertical <TT> correlation constructed from Eqs. (4.6), (4.7), and (4.9) using the above ideas. The abscissa is latitude from north pole to south pole and the ordinate is pressure. There are 32 vertical levels unequally-spaced in log(P) between 1070 hPa and 50 hPa. δ(φ) in (4.6) is 1 at the poles and falls symmetrically to 0 at the equator. We have chosen a SOAR formulation with the tropical spectrum $b_{\Phi\Phi}$ having a whiter spectrum than the extratropical spectrum $b_{\psi\psi}$. The contour interval is 0.1, and values between –0.1 and +0.1 are white. The correlations at each latitude are with respect to the 300 hPa level. This figure can be loosely compared with Fig. 26 of Bouttier et al. (1997), bearing in mind that the pressure levels, correlation models, and vertical presentation are different. In Fig. 4.9, the tropical vertical correlations are tighter than at high latitudes, as evident in Bouttier et al. (1997).

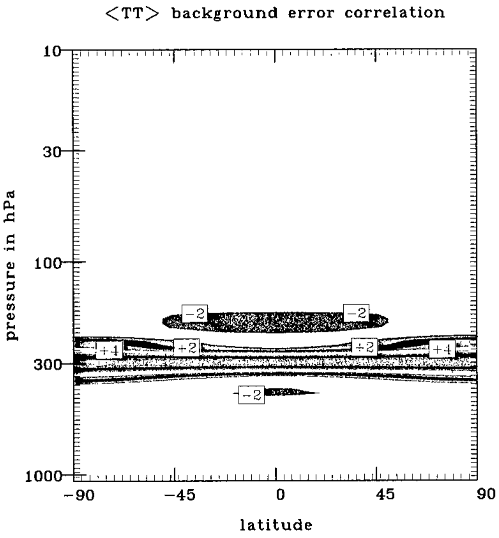


Figure 4.9
<TT> correlation with 300 mb as a function of latitude

4.7.2 Vertical variation of horizontal length scales

There are distinct advantages to a nonseparable formulation. For example, the ECMWF 3DVAR background error correlations are formulated in wave-space in the horizontal and discretely in the vertical. This permits a different vertical length scale for each horizontal wave. In particular, the vertical length scale is permitted to be shorter for the smaller scale horizontal waves-sort of a three-dimensional isotropy. Consider the wind and geopotential in this formulation and assume the geostrophic constraint is strictly applied. Now, the effective horizontal scale of the winds is always shorter than for the geopotential because of the horizontal derivatives implied by the geostrophic relation. Now through a nonseparable formulation, the shorter horizontal scales of the wind field are associated with the shorter vertical scales, whereas for the geopotential, the horizontal scales are longer and therefore the vertical scales are also longer. This results in a wind correlation (which already has a shorter horizontal correlation length than the geopotential) also having a shorter vertical scale. This is a distinct advantage because the vertical decorrelation length for winds really is shorter than for the geopotential. The same effect tends to produce shorter horizontal correlation lengths for the temperatures than for the geopotentials, which is also desirable. A desirable side effect is that individual observations at one level affect only the large horizontal scales of the analysis at distant levels. The well-known increase with height of the horizontal correlation scales (see Lonnberg and Hollingsworth, 1986) can also be accommodated within this formulation. The observation space algorithm used by NRL and NASA Goddard can also accommodate some nonseparability of this type, although it is less straightforward. At NASA Goddard, the vertical correlations are formulated in real space, which certainly permits the vertical variation of the horizontal correlation length-a very important feature. However, it is not easy to see how the other nonseparable features of the ECMWF formulation could be achieved with their formulation.

Section (4.4.9) discussed how the horizontal correlations could be made vertically mode dependent. In particular, we can permit the horizontal correlation length to vary as a function of the vertical mode number. In (4.12), we have described how the geostrophic coupling parameter is a maximum at large vertical scales and decreases for smaller vertical scales. This is already a form of nonseparability for the multivariate problem. In the same way, we can vary the horizontal length scale L_b^h as a function of vertical mode number.

This requires a few modifications to the theory of Section 4.3. Thus define $d_{\psi\psi}^n$ as the nth element of the $N_v \times N_v$ diagonal matrix $\mathbf{D}_{\psi\psi}$ defined in Eq. (4.4). Now define the $N_v \times N_v$ diagonal matrix $\mathbf{D}_{vv}$ with elements d_{vv}^n. Define L_h^0 as the nominal background error horizontal length scale and L_h^n as the background error horizontal length scale for the nth vertical mode. Then, we define the elements of the wind/wind error correlation in vertical eigenspace as

$$d_{vv}^n = d_{\psi\psi}^n (L_h^0/L_h^n)^2. \tag{4.25}$$

We define the vertical variation of the background wind error variance as

$$\mathbf{S}_v = \mathrm{diag}[\mathbf{S}_\psi^{1/2}\mathbf{E}\mathbf{D}_{vv}\mathbf{E}^T\mathbf{S}_\psi^{1/2}]. \tag{4.26}$$

Then, the wind/wind correlation is given by

$$\mathbf{C}_{vv} = \mathbf{S}_v^{-1/2}\mathbf{S}_\psi^{1/2}\mathbf{E}\mathbf{D}_{vv}\mathbf{E}^T\mathbf{S}_\psi^{1/2}\mathbf{S}_v^{-1/2}. \tag{4.27}$$

Under separable conditions, $L_h^n = L_h^0$, $1 \leq n \leq N_v$, and consequently $\mathbf{D}_{vv} = \mathbf{D}_{\psi\psi}$ and $\mathbf{S}_v = \mathbf{S}_\psi$ and (4.27) collapses to (4.5).

One other modification is required, that is, Eq.(4.12) is replaced by

$$d_{\Phi v}^n = \mu_v^n d_{\psi\psi}^n L_h^0/L_h^n. \tag{4.28}$$

Figure 4.10 (in the same format as Figs. 4.5 and 4.6) plots an example of nonseparability achieved by varying L_b^h as a function of vertical mode number. Thus, L_b^h is a maximum for the gravest vertical mode and monotonically decreases for the higher vertical modes. Each panel shows a two-dimensional (pressure/horizontal distance) plot of a various correlations with a given point. Figure 4.10(a) is for a separable correlation and Fig. 4.10(b) is for nonseparable correlations. All of the correlations are strictly hydrostatically and geostrophically coupled. Note that the horizontal scale of the <TT> correlation and the vertical scale of the <vv> correlation are shortened in the nonseparable formulation, e.g., these correlations are more three-dimensionally isotropic. A desirable side effect of this nonseparable form is that observations tend to affect only the larger horizontal scales of the analysis at distant levels (because the deep vertical modes have large horizontal scales).

A disadvantage of the nonseparable correlations of Fig. 4.10(b) is that the effective horizontal correlation length is the same at all levels. Thus, the rather important advantage of increasing the horizontal correlation scale for decreasing pressure has not been achieved with this nonseparable formulation. However, all is not lost; there is still one other set of free parameters to vary. Figure 4.10(b) was produced by using the <ψψ> vertical correlation of (4.4), with $\tilde{\mathbf{D}}$ equal to the identity matrix. Suppose instead, that we permit the elements of the diagonal matrix $\tilde{\mathbf{D}}$ to increase with height (decrease with pressure). Figure 4.11 illustrates this effect on the horizontal correlation length L_b^{h}. In this example, there were 40 vertical levels, unequally spaced in log(pressure) from 1070 to 1 hPa. The elements of $\tilde{\mathbf{D}}$ increased from 1.0 at 1070 hPa to 3.0 at 100 hPa and then remained constant above 100 hPa. The horizontal correlation length decreased for the higher vertical modes. Although the horizontal length scale is specified as a function of vertical mode number, an effective horizontal length scale can be plotted as a function of pressure. This is what is plotted in Fig. 4.11. The solid curve is the effective horizontal scale for the geopotential and winds, and the dash-dot curve is the corresponding scale for the temperature. As discussed above, the horizontal temperature scale is expected to be shorter in the nonseparable formulation. In this formulation, the horizontal length scale for the geopotential and winds increases slowly with decreasing pressure in the troposphere and then increases much more rapidly in the lower stratosphere.

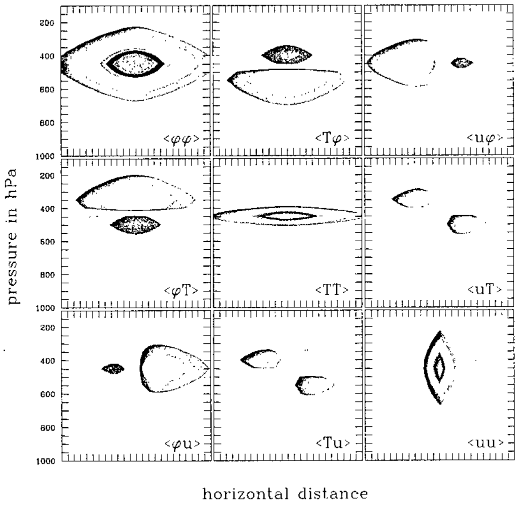


Figure 4.10a
x/P cross-sections of 9 correlations – separable formulation

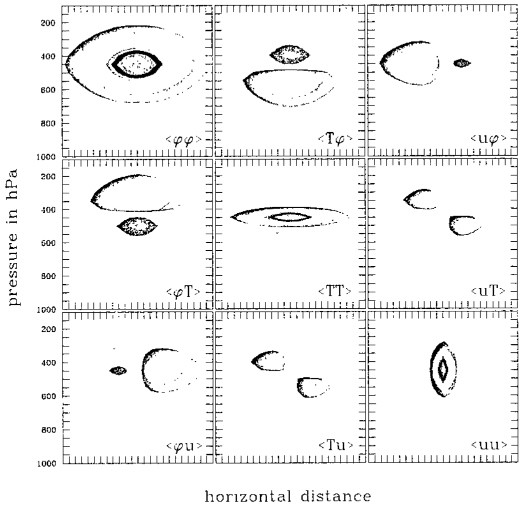


Figure 4.10b
Same as Figure 4.10a, except for non-separable formulation

As noted earlier, this nonseparable formulation means the S_v and S_ψ may not be the same. An example of this is shown in Fig. 4.12, corresponding to the same case as Fig. 4.11. We then calculate S_Φ , S_T, and S_v and normalize them by their value at the bottom(at 1070 hPa). The solid line is S_Φ, the dashed-dot line is S_T, and the dashed line is S_v. In the separable case, the normalized geopotential and wind error variances would be the same (solid and dashed lines would overlay). This shows that because of the increase in horizontal length scale with decreasing pressure (Fig. 4.11), the wind error variance increases less rapidly than the geopotential error variance. Note the wind error maximum near the tropopause (300 hPa).

The principal disadvantage of all nonseparable formulations is that they are inherently more costly in the calculation of the correlations (although not necessarily in their subsequent use). Thus, in this nonseparable formula-

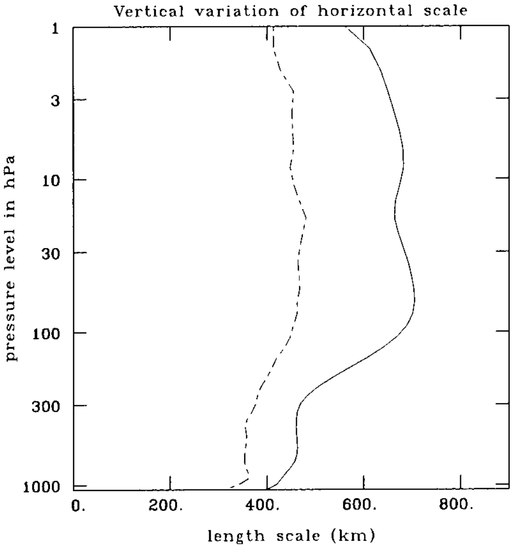


Figure 4.11
Background error horizontal correlation length

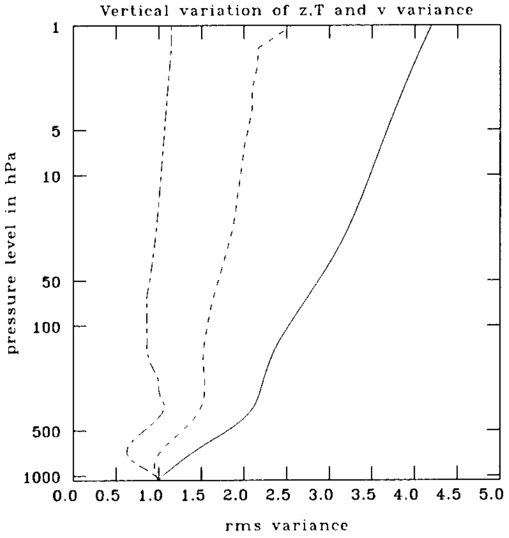


Figure 4.12
Background error horizontal correlation length

tion, it would be necessary to calculate all horizontal correlations for each vertical mode, rather than calculating them once for all vertical modes as in the separable formulation. However, even though the horizontal correlations are different from each vertical mode, the great-circle distance and angles (see Appendix A) remain invariant. In practice, there are ways to make the computational penalty for a nonseparable formulation quite light-less than a factor of 2 in cost.

4.7.3 Correlation with the divergent wind

It is possible to include correlations with the divergent wind in a separable formulation and indeed (as shown in Section 4.3.4) we already include $\mathbf{C}_{\chi\chi}$ correlations with the same vertical structure as the $\mathbf{C}_{\psi\psi}$ correlations. However, this is not appropriate when the divergent wind is correlated with the geopotential and rotational winds. These correlations are known to be small, but nonzero, particularly near the Earth's surface. In particular, there is convergence into low-pressure regions and divergence from high-pressure regions. As shown in Appendix B, this results in a clockwise rotation of all the correlations in the Northern Hemisphere and an anti-clockwise rotation in the Southern Hemisphere. The rotations are maximized at the Earth's surface.

If there is to be differential rotation of the correlations with decreasing pressure, this requires a nonseparable formulation of the $\mathbf{C}_{\psi\chi}$ and $\mathbf{C}_{\Phi\chi}$ correlations. We illustrate the $\mathbf{C}_{\psi\chi}$ correlation, but the $\mathbf{C}_{\Phi\chi}$ correlation is handled in a similar fashion. We define the velocity potential/streamfunction correlation as in Section 4.3.4, that is,

$$\mathbf{C}_{\psi\chi} = \mathbf{E}\mathbf{D}_{\psi\chi}\mathbf{E}^T, \tag{4.29}$$

where $\mathbf{D}_{\psi\chi}$ is diagonal with positive elements. The vertical eigenvalues of the <ψψ> correlation tend to decrease for the shallower modes. Moreover, if we use the form of the matrix $\tilde{\mathbf{D}}$ illustrated in Fig. 4.2, then the deep vertical modes of $\mathbf{C}_{\psi\psi}$ will have their maximum amplitude in the stratosphere, and most of the contributions near the Earth's surface will come from the shallow modes. Then, we define a much whiter spectrum (i.e., falls off much less rapidly for increasing mode number) for $\mathbf{D}_{\psi\chi}$ than for $\mathbf{D}_{\psi\psi}$. This results in much shallower vertical correlations for the <ψχ> correlations than for the <ψψ> correlations. But it will also have another effect. For the deep modes, the <ψχ> correlations will be completely dominated by the <ψψ> correlations. For the shallow modes, however, the <ψψ> and <ψχ> correlations will be much more similar in magnitude. Since the deep modes are maximimized in the stratosphere and the shallow modes near the Earth's surface, the effect of including the <ψχ> correlations is maximized at the Earth's surface and decreases with decreasing pressure. As illustrated in Appendix B, this results in a maximum rotation of the correlations near the ground, with virtually no rotation at high level. (The rotation is clockwise in the Northern Hemisphere and anti-clockwise in the Southern Hemisphere).

4.8 Projection of the Correction Field on Background Vertical Eigenvectors

Section (4.4) described the method for projecting onto the vertical eigenvectors of the background error correlation for use in matrix/vector operations in the solver and the post multiplication. Let us now consider the post multiplication step (3.6) more closely. The input to the post multiplier is in observation space, and the output is in analysis grid space. Note that the output consists entirely of profiles for each variable at each horizontal grid point. Following Section (4.4), the post multiplication step can be written in the form (4.16). There is a sequence of three operations:

(1) the $\mathbf{E}^T$ operation-a projection for all observations (profiles, soundings and single level observations) into vertical eigenvector space;

(2) the **D** operation-a matrix vector/multiplication entirely in vertical eigenvector space to produce vertical eigenvalue projections of the correction field for each variable at each analysis grid point; and

(3) the **E** operation-a transformation for each variable at each horizontal grid point from vertical eigenvector space to real space.

At the end of step (2), we have, essentially, a vertical spectral form of the correction field at each analysis grid point. There is some advantage to terminating the post-multiplier at the end of step (2) rather than immediately proceeding to step (3). In particular, by storing the output from the 3DVAR algorithm as vertical spectral amplitudes for each variable at each analysis grid point, we retain an enormous flexibility in transforming the correction fields to any vertical coordinate at any specified levels. Thus, we could perform the final transformation (step 3) to real space as the first operation of a model initialization (COAMPS or NOGAPS) in whatever vertical coordinate system (sigma surfaces, for example) with the field of vertically decomposed corrections. The output on constant pressure surface (for display purposes) could be produced with the same field.

Following the procedure of Section (4.4) precisely will produce N_v eigenvector amplitudes for each variable and each horizontal grid point. If N_v is relatively large, it may well be that we can produce sufficiently accurate correction fields by only calculating and storing the amplitudes of the projection on $M_v < N_v$ of the gravest vertical modes. That is, we could use this device to vertically filter the correction fields, thus making considerable savings in both storage and computation. Thus, if $M_v = N_v/2$, the post-multiplication step would be half as costly. The choice of M_v would be made experimentally.

5. Instruments — Observation Errors and Forward Models

This section discusses properties of the instruments used to obtain the observations. We are not concerned here with the reading, sorting, or quality control of the data, but rather the instrument characteristics that must be accounted for in correctly determining the observation and background error covariances. For each instrument, we will discuss its observation error characteristics, followed by a discussion of the forward model for the instrument (if there is one).

5.1 Vertical Profiles, Vertical Soundings, and Single-Level Observations

In Section 4 (Eqs. (4.14)-(4.20)), we discussed the projection of vertical profiles onto the eigenvectors of the background error correlation matrix. We also demonstrated for the solver (Eq. (3.5)) that this projection was inefficient for single-level observations (surface observations, cloud track winds, etc.), and consequently such observations were handled in real space. Now the projection into eigenvector space can, in principle, be used for any type of vertical profile information, including in situ observations (radiosondes, in particular) or remotely sensed information (satellite radiances, total precipitable water, etc.). We refer to in situ observations, in which the variable measured is the same as one of the analyzed variables, as profiles. We refer to remotely sensed information in which the variable measured is not an analysis variable as soundings. We treat soundings and profiles slightly differently.

For both sounding and profile, we project into eigenvector space. However, for profiles we assume that the observations project onto all the vertical modes; for soundings we project onto a subset of the gravest vertical modes. Thus, suppose that there are N_v vertical modes, and $N_v^s \leq N_v$ is a subset of modes with the largest eigenvalues (corresponding to the gravest vertical scales of the background error correlation, see Fig. 4.4). Then, we would project vertical soundings onto only the N_v^s gravest modes. Following Section 4.4.2, the implication of this assumption is that in calculating background error correlations between two soundings, or a sounding and a profile, the calculation would involve only the first N_v^s elements of the **D** matrices of Section 4.4.2 instead of all N_v elements. Thus we would define $\underline{\mathbf{D}}$ as the $N_v^s \times N_v^s$ diagonal matrix consisting of the N_v^s largest eigenvalues. We would similarly define the corresponding reduced $N_v \times N_v^s$ eigenvector matrix $\underline{\mathbf{E}}$. Similarly, interactions between single-level observations and soundings would involve only the first N_v^s vertical eigenfunctions. Thus, for modes $n \leq N_v^s$, we would project both soundings and profiles; for $N_v^s < n \leq N_v$, we would project only profiles.

Justification for this procedure is based on the fact that sounding and profiles are fundamentally different. A profile (radiosonde, say) has high vertical resolution, but may be incomplete (i.e., it does not always sample the whole atmosphere). A sounding, on the other hand, always samples the whole atmosphere (we would reject it if most of the channels were missing) using broad, overlapping weighting functions that have a very low vertical resolution. In effect (as we demonstrate in Appendix G) a sounder only "sees" the gravest vertical modes (i.e., has only a few degrees of freedom). Thus, for example, TOVS has about five pieces of temperature information, a hyperspectral sounder like AIRS has 10-20, and a total precipitable water measurement has only 1. Thus, it is clearly advantageous to project information of this sort only onto the gravest modes of the background error correlation.

We treat temperature or moisture profiles derived from outside agencies (such as the National Environmental Satellite, Data, and Information Service, NESDIS) as profiles. For NRL 1DVAR (one-dimensional variational) retrievals, the correction vector for the 1DVAR retrieval becomes part of the innovation vector for 3DVAR.

Since we know that the correction vector for the 1DVAR retrieval contains very few degrees of freedom, we can treat 1DVAR retrievals as soundings in which the forward operator **H** is the identity matrix. Direct assimilation of radiances, which we treat as soundings, will actually require the use of a nontrivial forward operator (see Section 5.3)

5.2 Radiosondes and Pibals

Radiosondes provide vertical profiles of temperature, height, and horizontal wind components. Pibals provide only winds. Temperatures are measured at significant levels, but both temperatures and heights are also provided at the (lower vertical resolution) mandatory levels. In the NRL MVOI code, the radiosonde mass observations were provided in the form of geopotential observations at the mandatory levels (1000, 850, 700 hPa, etc.). The geopotential observation errors were assumed to be vertically uncorrelated. The NAVDAS code is not tied in any way to the mandatory levels, and it is possible to analyze radiosonde temperature observations from the significant levels. This has two advantages:

(1) using the significant levels gives higher vertical resolution; and

(2) while the radiosonde temperature observation errors are vertically uncorrelated (or perhaps weakly correlated), the geopotential observation errors are strongly correlated.

5.2.1 Vertical correlation of radiosonde geopotential errors

This correlation of the geopotential errors is caused by the fact that the actual radiosonde mass observations are temperature and the geopotential observations are obtained by vertical integration of the hydrostatic equation. This can be demonstrated as follows. Consider vertically uncorrelated temperature observation error. Figure 5.1 (solid curve) shows the temperature error correlation <TT> for an atmosphere with 100 vertical levels spaced equally in log(pressure) between 1070 and 50 hPa with the middle level (50). The temperature error correlation is a spike, as specified.

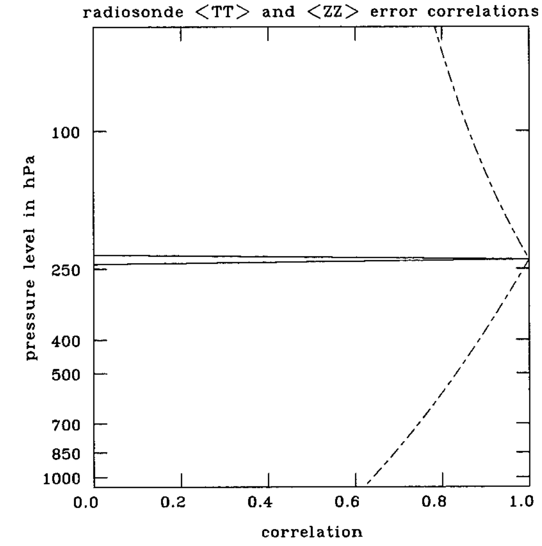


Figure 5.1
Radiosonde observation error correlations – <TT> and <ΦΦ>

The geopotential observation error correlation can be obtained from the temperature error correlation by integrating the hydrostatic equation, which in integral form is written,

$$\Phi(P) = \Phi_s - \mathrm{R} \int_{Ps}^{P} T\, d\ln P,$$

where P is pressure, R is the gas constant, P_s is the pressure at the surface, and T is the temperature. This form is straightforwardly discretized to produce an NxN triangular matrix, where N is the number of levels in the discrete representation.

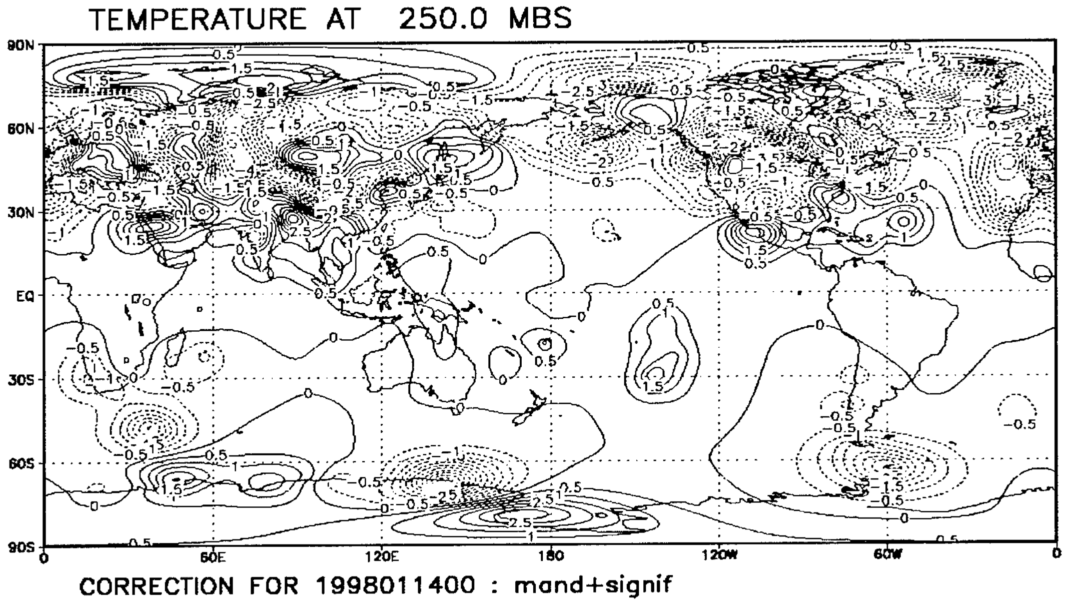
$$\Phi_n = \Phi_s + \mathrm{R} \sum_{m=1}^{n-1} T_m[\ln P_{m-1} - \ln P_m]. \tag{5.1}$$

Pre-multiplication of the diagonal temperature error covariance by the hydrostatic matrix and post-multiplication by its transpose produces the radiosonde geopotential observation error covariance matrix (dash-dot curve in Fig. 5.1). It can be seen that the geopotential error is highly vertically correlated, with a greater correlation above than below. Note the different curvature above, rather than beneath. This is the signature of a hydrostatic integration and is visible in the radiosonde observation geoptential error correlations plotted by Lonnberg and Hollingsworth (1986, Fig. 3).

Radiosonde observational geoptential error correlations can be included in the NAVDAS formulation, and will obviously make more optimal use of the geoptential observations than if this correlation is neglected. There is a problem, however. If we attempt to assimilate a vertical column of radiosonde geopotential observations, we find that there is a significant difference in the condition number of the $\mathbf{HP}_b\mathbf{H}^T + \mathbf{R}$ matrix, depending on whether or not we assume the radiosonde geopotential error is spatially correlated. Consider a radiosonde ascent with 24 geopotential observations. If the observation error correlation is ignored, then the condition number is 10.5. If the observation error correlation is included, then the condition number is 1161.3. If 24 (uncorrelated) temperature observations are used instead, then the condition number is 3.25. As noted earlier, correlated observation error increases the condition number for the present observation space implementation of the 3DVAR algorithm. Thus, radiosonde temperature observations at the significant levels are likely to provide more usable information.

Thus, for both radiosondes and pibals the observed variables are wind, temperature, and moisture ($\log_e(q)$) for all significant and mandatory levels. (The geopotential at the surface is also known and used.) It might be noted that some observations that are coded as pibals are actually the significant levels winds from a radiosonde ascent. The winds from these fictitious pibals are, of course, combined with the relevant radiosonde mandatory level winds before the quality control stage.

Figure 5.2 illustrates the differences between assimilating winds and geopotentials at mandatory levels only with assimilating winds and temperatures at all mandatory and significant levels. Both panels show the 250 hPa temperature correction field (in degrees K, contoured at 0.5 degree) for a global analysis for January 14, 1998 at 0000 GMT, produced from radiosonde and pibal observations only. Panel (a) is for 63,600 mandatory and significant level T,u,v observations; panel (b) is for 24,800 mandatory Z,u,v observations. Height observation errors were assumed to be vertically uncorrelated in panel (b). The differences are not insignificant, but it is not obvious which would give the superior analysis.


TEMPERATURE AT 250.0 MBS

CORRECTION FOR 1998011400 : mand+signif

Figure 5.2a
Correction fields due to radiosonde (mandatory plus significant levels)

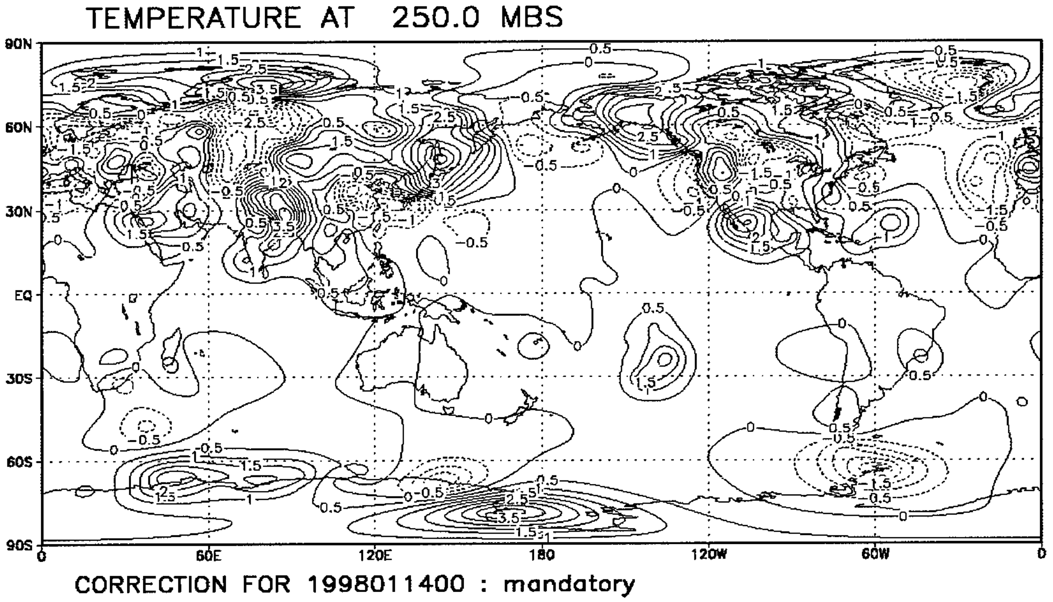


Figure 5.2b
Same as 5.2a, except for mandatory only

5.3 Nadir Temperature Sounders (TOVS) — Linearized Form

The NAVDAS algorithm has three options for assimilating TOVS temperature and moisture information:

(1) Assimilate retrieved temperatures and moistures from outside agencies such as NESDIS

(2) Assimilate retrieved temperatures and moistures using NRL 1DVAR off-line retrievals

(3) Assimilate TOVS radiances directly into the 3DVAR algorithm.

We discuss each option in turn.

5.3.1 Assimilation of NESDIS retrievals

In this case, observations are treated exactly as if they were radiosonde data, except that the specified observation errors may be different.

5.3.2 Assimilation of NRL 1DVAR retrievals

Nancy Baker (NRL Monterey) has developed a 1DVAR retrieval for TOVS that takes brightness temperatures and converts them into retrieved temperatures (and moistures) by using Eq. (6.2). The background temperature (and moisture) fields for these NRL retrievals are the same as for the NAVDAS algorithm described in Section (5.3.3).

The 1DVAR algorithm is an off-line counterpart of the 3DVAR algorithm to be described in Section (5.3.3). It can be used in its own right, with the retrieved temperatures and moistures ingested into the 3DVAR codes exactly as if they were radiosonde observations (but with different specified observation errors). The correction vector from the NRL 1DVAR retrievals will become an innovation vector for the 3DVAR procedure.

As shown in Appendix G, the TOVS instrument contains only large vertical information. This means that the correction field from the NRL retrieval will contain only large vertical scale temperature increments. Since this innovation vector contains only the largest vertical scales of the temperature (or moisture), we can represent this field with only $N_v^s \leq N_v$ vertical background error eigenvectors. (This would not necessarily be true for externally produced retrievals (see Section 5.3.1), which use different background fields). We would treat NRL 1DVAR retrievals as if they were soundings, projecting them on only the N_v^s gravest vertical modes.

The main objection to using the NRL 1DVAR retrievals is that they use the same background field as the 3DVAR system. Therefore, the 1DVAR retrieval error (which is the observation error for the 3DVAR algorithm) is, in fact, correlated with the background error. This correlation is probably quite large and difficult to account for in the 3DVAR algorithm.

Even if we decide to directly assimilate the TOVS radiances (as in Section 5.3.3), we still perform the off-line 1DVAR retrieval of temperature and winds. There are several reasons for this.

(1) It is an important step in the quality control of radiances, both to reject bad radiances and to correct the inherent biases in each channel.

(2) It is simpler to perform a nonlinear minimization in 1DVAR than in 3DVAR, so for inherently nonlinear radiative transfer models, we can use the 1DVAR retrieval to obtain a better state estimate than the background field.

(3) While we still assimilate the radiances in the 3DVAR procedure with respect to the original (forecast) background field, we can use the 1DVAR-derived state estimate to linearize the nonlinear forward model H(**x**) to produce the associated tangent linear operator **H**.

5.3.3 Direct assimilation of TOVS radiances

Section 5.1 discussed the projection of sounding information onto a subset of the gravest ($N_v^s \leq N_v$) vertical eigenmodes of the background error correlation matrix. We now illustrate how this is done with the TOVS instrument. This instrument measures radiances in about 20 microwave and IR channels. For the moment, we restrict ourselves to a single vertical column, that is the one-dimensional case. Denote the number of channels as N_c, and define some vertical column of temperatures (actually virtual temperatures) **T** at N_v discrete pressure levels. Then H(**T**) is defined as the (possibly nonlinear) radiative transfer operator, which produces radiances (or brightness temperatures) in N_c channels from the temperature column (**T**). Define the background and analyzed (retrieved) temperature vectors $\mathbf{T}_b$ and $\mathbf{T}_a$ at the same N_v pressure levels. Then, define $\mathbf{H} = \partial H(\mathbf{T})/\partial \mathbf{T}$ evaluated at $\mathbf{T} = \mathbf{T}_b$ as the $N_v \times N_c$ Jacobean matrix or tangent linear matrix, or defined as in Section 5.3.2 using the off-line NRL 1DVAR retrieval Note that this last procedure uses the same radiance observations twice, but only to linearize the radiative transfer operator. Then define **R** as the $N_c \times N_c$ radiance observation error matrix (usually, but not necessarily diagonal) and **y** as the vector of length N_c radiances. Define $\mathbf{C}_{TT}$ as the background temperature error correlation ($N_v \times N_v$) (Eq. (4.10)) and $\mathbf{S}_T$ as the diagonal background error variance matrix ($N_v \times N_v$). Then, we can define the temperature correction field using the scaled form (3.17),

$$\mathbf{T}_a - \mathbf{T}_b = \mathbf{S}_T{}^{1/2}\mathbf{C}_{TT}\mathbf{S}_T{}^{1/2}\mathbf{H}^T\,[\mathbf{H}\mathbf{S}_T{}^{1/2}\mathbf{C}_{TT}\mathbf{S}_T{}^{1/2}\mathbf{H}^T + \mathbf{R}]^{-1}[\mathbf{y} - H(\mathbf{T}_b)]. \tag{5.2}$$

We also define (as in 3.17) $\mathbf{S}_h = \mathrm{diag}[\mathbf{H}\mathbf{S}_T{}^{1/2}\mathbf{C}_{TT}\mathbf{S}_T{}^{1/2}\mathbf{H}^T]$ as a diagonal $N_c \times N_c$ background radiance error variance matrix. From Eq. (4.10), we can define $\mathbf{C}_{TT}$ as

$$\mathbf{C}_{TT} = \mathbf{S}_T{}^{-1/2}\mathbf{H}_s\mathbf{S}_\psi{}^{1/2}\mathbf{E}\mathbf{D}\mathbf{E}^T\mathbf{S}_\psi{}^{1/2}\mathbf{H}_s{}^T\mathbf{S}_T{}^{-1/2}, \tag{5.3}$$

where $\mathbf{H}_s$ is the $N_v \times N_v$ hydrostatic matrix to determine temperatures from geopotentials, $\mathbf{S}_\psi$ is the $N_v \times N_v$ matrix of vertical background error variances specified after Eq. (4.5), and **E** and **D** are the $N_v \times N_v$ matrices (**D** diagonal) of vertical background error eigenvectors and eigenvalues. Now, following the discussion in Section 5.1, suppose only $N_v^s \leq N_v$ vertical background error eigenvectors are to be used. We then define $\underline{\mathbf{D}}$ to be a $N_v^s \times N_v^s$ reduced diagonal matrix consisting of the N_v^s largest eigenvalues of **D**, and $\underline{\mathbf{E}}$ to be a $N_v \times N_v^s$ rectangular matrix consisting of the eigenvectors **E** corresponding to the N_v^s largest eigenvalues.

Now define the $N_c \times N_v^s$ matrix,

$$\underline{\mathbf{H}} = \mathbf{S}_h^{-1/2} \mathbf{H} \mathbf{H}_s \mathbf{S}_\psi^{1/2} \underline{\mathbf{E}} . \tag{5.4}$$

For TOVS, $N_c < 20$ and $N_v^s < 10$, so the matrix $\underline{\mathbf{H}}$ is relatively small. Substitution of (5.4) into (5.2) gives

$$\mathbf{T}_a - \mathbf{T}_b = \mathbf{H}_s \mathbf{S}_\psi^{1/2} \underline{\mathbf{E}}\underline{\mathbf{D}}\underline{\mathbf{H}}^T [\underline{\mathbf{H}}\underline{\mathbf{D}}\underline{\mathbf{H}}^T + \mathbf{S}_h^{-1/2} \mathbf{R} \mathbf{S}_h^{-1/2}]^{-1} \mathbf{S}_h^{-1/2} [\mathbf{y} - H(\mathbf{T}_b)]. \tag{5.5}$$

In Eq. (5.5), the nadir sounder radiance observations have been vertically projected onto a subset of the background error vertical eigenvectors. We can thus apply all of Section 4.4 to this case. Suppose we consider the case with many soundings, not just a single sounding. Then, we would be using a conjugate gradient descent method that would involve large matrix/vector multiplications ((3.13) and (4.14)-(4.20)). The difference would be that equations such as (4.14) involving two soundings (instead of two profiles) would be written as $\mathbf{q} = \underline{\mathbf{H}}_2 \underline{\mathbf{D}}_{12} \underline{\mathbf{H}}_1^T \mathbf{r}$, while interactions between a sounding and a profile would become $\mathbf{q} = \underline{\mathbf{H}}_2 \underline{\mathbf{D}}_{12} \underline{\mathbf{E}}_1^T \mathbf{r}$ or $\mathbf{q} = \underline{\mathbf{E}}_2 \underline{\mathbf{D}}_{12} \underline{\mathbf{H}}_1^T \mathbf{r}$. Note that in both cases, the diagonal matrix $\underline{\mathbf{D}}_{12}$ contains, in addition to the vertical eigenvalues, all the horizontal background and multivariate (wind/geopotential) correlation information. Thus, as in (4.17), matrix multiplications by $\underline{\mathbf{H}}$, $\underline{\mathbf{E}}$, $\underline{\mathbf{H}}^T$, or $\underline{\mathbf{E}}^T$ are all performed on individual soundings or profiles; they do not involve interactions between different soundings or profiles. The additional complication with soundings that did not exist with profiles; is that **E** or $\underline{\mathbf{E}}$ are universal matrices, whereas $\underline{\mathbf{H}}$ may be different for every sounding.

We now discuss three technical issues with this algorithm. The first issue is that it has been assumed here that the pressure levels of the **H** matrix are the same as those of the C_{TT} and $\underline{\mathbf{E}}$ matrices. This may not, in general, be true. To handle the more general situation, where these pressure levels are not the same, define the interpolation matrix **G** that interpolates any quantity from the N_v levels of the background temperature $\mathbf{T}_b$ to the levels required by the forward radiative transfer equation (H). Then, in this case, we simply replace (5.4) by

$$\underline{\mathbf{H}} = \mathbf{S}_h^{-1/2} \mathbf{H} \mathbf{G} \mathbf{H}_s \mathbf{S}_\psi^{1/2} \underline{\mathbf{E}}. \tag{5.6}$$

The second issue is that the radiative transfer equation requires temperatures at much greater altitudes than are generally available in the background field $\mathbf{T}_b$. Suppose that there are N_v temperatures in a vertical column of the background field, with a minimum (top) pressure P_{top}. Suppose also that the radiative transfer equation requires an additional N_t temperature $\mathbf{T}_{rad}$ at pressure levels from P_{top} to some much lower pressure (higher top) P_{rad}. Then, the total temperature vector required for the radiative transfer equation is $[\mathbf{T}_b^T \;\; \mathbf{T}_{rad}^T]^T$. Now, split the H operator into two parts-$H_{rad,}$ which goes from the highest background temperature level to the top of the atmosphere ($P_{rad} \leq P \leq P_{top}$), and $H_{b,}$ which is for pressure levels greater than P_{top}. Similarly, divide **H** into $\mathbf{H}_{rad}$ and $\mathbf{H}_b$. Then, simply replace **H** in (5.4) by $\mathbf{H}_b$ and the innovation $[\mathbf{y} - H(\mathbf{T}_b)]$ in (5.5) with the new innovation $[\mathbf{y} - H_{rad}(\mathbf{T}_{rad}) - H_b(\mathbf{T}_b)]$. $\mathbf{T}_{rad}$ is assumed to be specified externally from climatology or a previous off-line 1DVAR retrieval. Thus, $H_{rad}(\mathbf{T}_{rad})$ is, in effect, the simulated radiance from the top of the background field to the top of the atmosphere, and $\mathbf{y} - H_{rad}(\mathbf{T}_{rad})$ is already known before the 3DVAR algorithm begins. This formulation is equivalent to specifying that the prespecified $\mathbf{T}_{rad}$ ($P_{rad} \leq P \leq P_{top}$) is perfect.

The third issue concerns the diagonal scaling matrix $\mathbf{S}_h$ (introduced after Eq. (5.2)). As noted after Eq. (3.17), scaling is introduced in the descent algorithm to ensure that all the elements of the $\mathbf{H}\mathbf{P}_b\mathbf{H}^T + \mathbf{R}$ matrix are dimensionless and O(1), which improves the condition number. This becomes increasingly important as new types of background and observed variables are introduced. The important difference between $\mathbf{S}_h$ and other scaling matrices, such as S_v, S_T and S_Φ introduced in Section 4, is that S_h depends not only the background error specification, but also on the instrument characteristics. This makes it somewhat more awkward to calculate.

We now illustrate calculations made using these ideas for a single vertical column. We can assess the accuracy of given retrieval without using any radiance observations, simply by estimating the analysis (retrieval) error using Eq. (2.8). Thus, given an radiative transfer operator H and its Jacobean **H**, the radiance observation error covariance **R**, and the background temperature error correlation $\mathbf{C}_{TT}$and variance $\mathbf{S}_\psi$ (from which we can calculate the background error covariance $\mathbf{P}_b$ in (2.8)), we can calculate the retrieval error covariance $\mathbf{P}_a$. Consider the diagonal elements of $\mathbf{P}_a$ (i.e., the retrieval error variances) and normalize by the diagonal elements of the background error covariance $\mathbf{P}_b$-that is, calculate diag($\mathbf{P}_a$)/diag($\mathbf{P}_b$), and then take the square root of each element. Each of these elements will be between 0 and 1; 0 indicates a retrieval with no error and 1 indicates a retrieval that is no more accurate than the background temperature. Table 5.1 is an example of such a calculation. The retrieval had 18 radiance channels and 40 pressure levels from 1000 hPa to 0.1 hPa. The first column gives the normalized retrieval errors for the standard case, with the background error available from 1000 hPa to 0 mb and all 40 eigenvectors used. We regard this as the optimal case for the given specification of **H**, **R**, $\mathbf{C}_{TT}$, and $\mathbf{S}_v$ and it would also satisfy (2.9). We show only selected pressure levels.

Table 5.1 — Retrieval Error as a Function of Pressure

Pressure Level	Retrieval Error Normalized By Background Error (RMS)		
	Optimal Case	Suboptimal (7 Modes, 1 hPa Top)	
0.1 hPa	0.9723	1.0000	1.0000
1.0 hPa	0.8044	0.8538	0.8359
10.0 hPa	0.6607	0.6871	0.6871
50.0 hPa	0.7843	0.7981	0.7980
100.0 hPa	0.7797	0.7838	0.7838
200.0 hPa	0.6599	0.6700	0.6714
500.0 hPa	0.7920	0.8044	0.8043
700.0 hPa	0.7359	0.7411	0.7433
850.0 hPa	0.9125	0.9277	0.9244
1000.0 hPa	0.9875	0.9956	0.9920

It can be seen that the observed radiances have the most effect at 200 hPa and again at 10 hPa, with very little effect at the top of the atmosphere or at very low levels. This is a consequence of the structure of the H operator (not shown) and reflects the spectroscopic properties of the various channels of the instrument.

Now consider a suboptimal case (but still normalized by the diag($\mathbf{P}_b$) from the optimal case). In this case (shown in the second column), there are only the seven gravest vertical background error eigenvectors, and the top background error level is 1 hPa. It can be seen that in all cases, the retrieval error is higher (but only slightly higher) despite having a lower top and only seven (instead of 40) vertical modes. The third column is described in Appendix G.

A word of warning is appropriate here. In Table 5.1, all the 18 radiance channels have been assumed to be available. If a number of the channels are missing, then of course, the relative errors in the optimal case (first column) would be higher. More importantly, the suboptimal case (second column) may deteriorate even more quickly. That is, because the background error eigenvectors sample the whole atmosphere, and if a portion of the atmosphere is not sampled by the instrument at all, then we have a situation more like an incomplete radiosonde profile, which would require all of the vertical eigenvectors to resolve properly.

Figure 5.3 shows an example of the background temperature error covariance in the vertical (degrees Kelvin)^2 for the optimal case, with pressure decreasing logarithmically from 1000 hPa to 0.1 hPa going from left to right on the abscissa and from bottom to top on the ordinate. It can be seen (as specified) that both the background temperature error variance and vertical scale increase from 1000 hPa to 0.1 hPa. (The contour intervals indicated in the small square boxes have been multiplied by 10). Covariances whose magnitudes are less than 0.8 (degrees Kelvin)^2 are colored as "white". Figure 5.4 is in the same format as Fig. 5.3; we show the background temperature error covariance for the suboptimal case (top at 1.0 hPa, seven vertical modes). The effect of the model top is evident in the background error , and the projection on only seven modes of the 40 results in broadening of the covariance. As Table 5.1 shows, however, this drastically different background temperature correlation seems to have very little effect on retrievals with the TOVS instrument. Why this should be so is explored in Appendix G. In case of excessive concern about the radically changed nature of the background error temperature covariance in Fig. 5.4, it should be remembered that this is the effective covariance used for the TOVS sounders. For radiosondes, all the eigenvectors are included so the effective covariance would appear more like Fig. 5.3 (at least below 1 hPa).

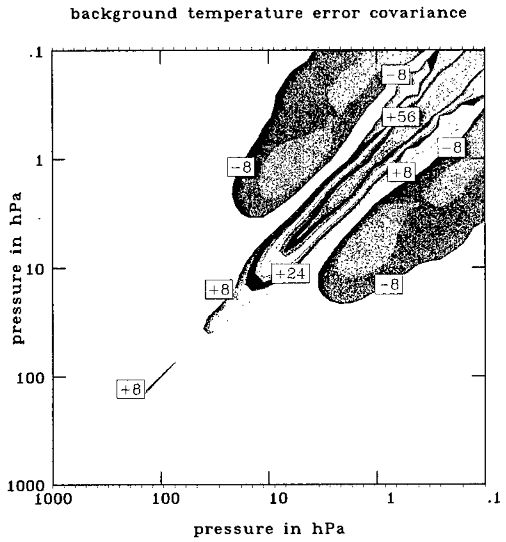


Figure 5.3
Specified <TT> background error covariance from 1000 to 0.1 mb

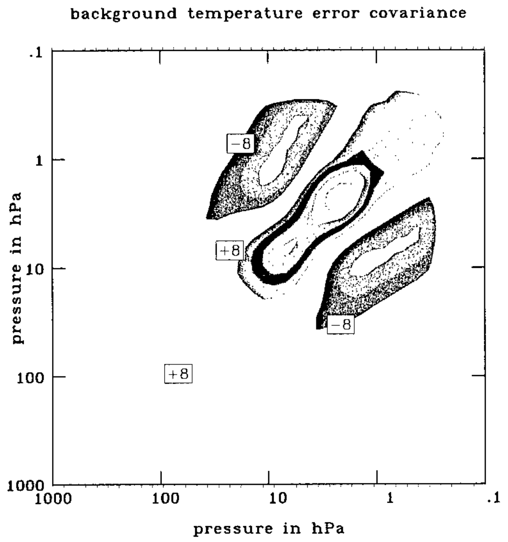


Figure 5.4
Same as Figure 5.3, except for 7 modes and top at 1.0 mb

Figure 5.5 shows an actual example of a global correction field derived from TOVS radiances alone using the above procedure. The correction field is for the 250 hPa temperature (degrees Kelvin, with contour interval 0.5 degrees) on a 1 degree global grid. There were 19 channels (approximately 1700 soundings, or approximately 32000 radiance observations). There was a separate $\underline{\mathbf{H}}$ matrix for each sounding, based on linearization about the background temperature at the location of the sounding. The background error covariance had 39 pressure levels to 1 hPa. In panel (a), the 10 gravest vertical modes of the background error correlation were used. In panel (b), all 39 vertical vertical modes were used. Consistent with the results of Table 5.1, the two panels are generally quite similar, indicating that most of the information in the correction field is in the 10 gravest modes. However, this is not always the case, as can be seen by examination of the correction fields over Europe (upper left hand corners). For such soundings, it may be necessary to use all N_v eigenvectors, rather than N_v^s eigenvectors as in most of the soundings.

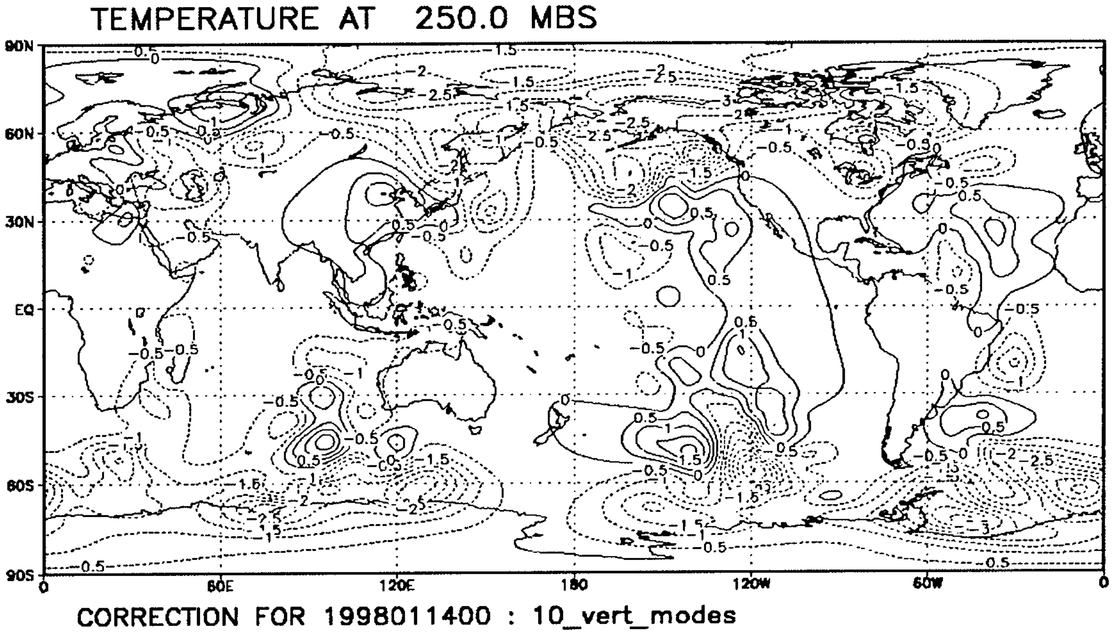


Figure 5.5a
Temperature correction from TOVS sounder (10 modes)

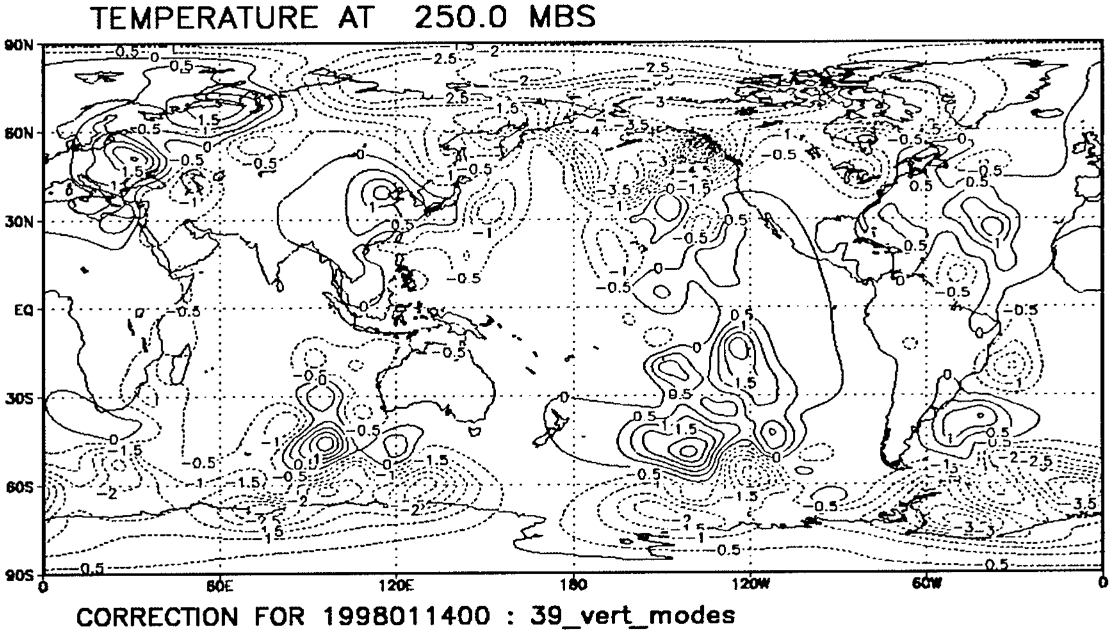


Figure 5.5b
Same as Figure 5.5a, except all 39 modes

To use this algorithm, the radiance innovations must be projected onto the vertical background error eigenvectors and then back to radiance space during every iteration of the conjugate gradient descent. These transformations are performed using the precomputed $\underline{\mathbf{H}}$ matrices (Eqs. (5.4) or (5.6)), which are different for every sounding. These $\underline{\mathbf{H}}$ matrices are N_cxN_v^s that are perhaps 100 to 200 elements, which is not prohibitive from a storage point of view for 1000-5000 soundings. The transformation themselves are also not costly because they do not involve interactions between soundings. Thus, this algorithm appears to be practical and efficient, and experiments with 2000 to 5000 soundings over the globe have confirmed this. It should be noted that this section has considered a linearized forward model; the full nonlinear operator is discussed in Section 6.

However, temperature sounders (called hyperspectral sounders) that may have more than 1000 channels are now in the development stage. Such a sounder could easily overwhelm any observation space assimilation system, even with the algorithm described above. For such hyperspectral sounders, this algorithm has to be taken a step further; this development is described in Appendix G.

5.4 Surface Observations

Conventional surface observations (land surface and shipboard) include measurements of wind, temperature, humidity, and pressure. These are all treated in a straightforward fashion using the station pressure. Pressure innovations are converted to geopotential innovations.

Mean sea level pressure (msl) is handled somewhat differently. Over land it is a fictitious variable, but it is still very important to forecasters, and the sea level pressure observations should fit the "observations" of msl pressure as accurately as possible. Thus, the sea level pressure analysis is part of the interface between the NAVDAS code and the model (NOGAPS or COAMPS) code. A background field of mean sea level pressure is obtained by extrapolation procedures from the NAVDAS analysis in the atmosphere (i.e., above terrain). Mean sea level innovations are then calculated at observation locations by differencing the mean sea level "observations" and the background field. These innovations are then analyzed using a two-dimensional univariate version of the NAVDAS code. This code performs a universal solve by pre-conditioned conjugate gradient descent, divides the innovations into prisms, and runs on distributed memory machines.

5.5 SSM/I Windspeeds (Linearized Form)

The SSM/I instrument indirectly measures surface windspeed. We assume that the windspeed retrieval algorithms produce a reasonable windspeed estimate and attempt to directly assimilate these windspeeds. This means that we do not make any explicit attempt to assign a direction to the SSM/I windspeed observation.

Windspeed is treated as a single-level quantity with a nontrivial forward operator. If w is the windspeed, then

$$w = H(u,v) = (u^2 + v^2)^{1/2}, \tag{5.7}$$

where u and v are the wind components in some coordinate system. Define **v** as the vector of length 2 with components u and v. The forward operator H is nonlinear in this case. If we linearize this operator around the background wind $\mathbf{v}_b$ with components u_b and v_b, then the appropriate tangent linear operator **H** is the 1×2 matrix

$$\mathbf{H} = [\delta H/\delta u_b \quad \delta H/\delta v_b] = [u_b/(u_b^2+v_b^2)^{1/2} \quad v_b/(u_b^2+v_b^2)^{1/2}]. \tag{5.8}$$

It is easy to show that $\mathbf{HH}^T = 1$ and $\mathbf{Hv}_b^T = H(u_b,v_b) = (u_b^2 + v_b^2)^{1/2}$. Thus, for the wind speed operator, the nonlinear operator H and the linearized (around the background) operator **H** have the same effect when applied to the background wind.

Now if we consider the background error covariance for the u and v fields at a fixed location, then we know that they are not correlated (even though u and v background errors may be correlated for two separate locations). Thus, the covariance for collocated u and v background errors is given by

$$\mathbf{C}_{vv} = \begin{bmatrix} \varepsilon_v^2 & 0 \\ 0 & \varepsilon_v^2 \end{bmatrix}, \tag{5.9}$$

where ε_v^2 is the background error variance (which is the same) for each of the u and v wind components. Then, the background windspeed error variance ε_w^2 is given by

$$\varepsilon_w^2 = \mathbf{HC}_{vv}\mathbf{H}^T = \varepsilon_v^2 . \tag{5.10}$$

Thus, in the notation of Eq. (3.17), we would define $\mathbf{C}_h^{ob/ob}$ as the 2x2 identity matrix, $[\mathbf{S}_h^{ob}]^{1/2}$ as the 2x2 diagonal matrix with diagonal elements ε_v, and $\mathbf{S}_h^{-1/2}$ as the single number ε_v^{-1}. Equations (5.7)-(5.10) contain all the elements necessary for the (linearized) assimilation of windspeeds.

Now let us examine a simple case where we have a single windspeed observation w_r and a single (collocated) background wind vector $\mathbf{v}_b$ with components u_b and v_b. Then, we apply (2.6) to this simple problem, yielding 3DVAR analyzed wind components u_a and v_a,

$$\begin{aligned} \mathbf{v}_a &= \mathbf{v}_b + (\varepsilon_\mathbf{v}^2 + \varepsilon_\mathbf{r}^2)^{-1} \begin{bmatrix} \varepsilon_v^2 & 0 \\ 0 & \varepsilon_v^2 \end{bmatrix} \mathbf{H}^T [w_r - H(u_b,v_b)] \ , \\ &= (\varepsilon_\mathbf{v}^2 + \varepsilon_\mathbf{r}^2)^{-1} (\varepsilon_\mathbf{r}^2 + \varepsilon_\mathbf{v}^2 w_r(u_b^2 + v_b^2)^{-1/2})\, \mathbf{v}_{b.} \end{aligned} \tag{5.11}$$

using (5.7)–(5.10) and $\varepsilon_\mathbf{r}^2$ is the specified SSM/I windspeed error variance. It is easy to see from (5.11) that the analyzed wind direction $\tan^{-1}(u_a/v_a)$ is equal to the background wind direction $\tan^{-1}(u_b/v_b)$. From (5.11), the analyzed windspeed is given by

$$H(u_a,v_a) = (u_a^2 + v_a^2)^{1/2} = (\varepsilon_\mathbf{v}^2 + \varepsilon_\mathbf{r}^2)^{-1} (\varepsilon_\mathbf{r}^2 (u_b^2 + v_b^2)^{1/2} + w_r \varepsilon_\mathbf{v}^2) \tag{5.12}$$

using Eq. (5.8). Thus, the analyzed windspeed is a linear combination of the SSM/I windspeed observation and the background windspeed. Equations (5.11) and (5.12) are not valid when there are many SSM/I observations, except in the special case when the background error correlation scale $L_h = 0$, in which case, the SSM/I observations do not influence each other.

Equations (5.7)-(5.10) provide the framework for the direct assimilation of SSM/I windspeeds. Three additional points must be made: First, background error wind components u_b and v_b are required at every SSM/I windspeed location that we desire to assimilate. Second, when $(u_b^2 + v_b^2)$ is very small, the tangent linear model is not defined, and we cannot use an SSM/I windspeed observation at that point. Third, Eq. (5.7) is nonlinear, and the optimal solution can only be obtained by iterating the 3DVAR algorithm. This can be done by modifying the innovation and performing an outer iteration (see Section 6.1).

5.6 SSM/I Total Precipitable Water

The retrieval of moisture from SSM/I total precipitable water measurements has been studied by Phalipou and Gerard (1996). As for the SSM/I windspeed, we assume that the total precipitable water in a vertical column has been adequately obtained by outside retrieval from the SSM/I instrument. We then attempt to directly assimilate the total precipitable water observations to analyze our moisture variable $s = \log_e q$, where q is the specific humidity (see Section 4.3). We treat the total precipitable water as a sounding, following the discussion in Section 5.3.3. In other words, since the total precipitable water is a vertical integral, we treat it as if it were a single channel sounding. Define $m = q/(1 - q) = e^s/(1 - e^s)$ as the water vapor mixing ratio and W as the total column precipitable water. Then,

$$W = g^{-1} \int_0^{P_s} m dP = g^{-1} \int_0^{P_s} e^s / (1 - e^s) dP \tag{5.13}$$

where g is the gravitational constant and P_s is the surface pressure. In discrete form, we can write (5.13) as

$$W \approx g^{-1}\sum_{n=1}^{N_v} m_n\left[P_{n+1}-P_n\right] = g^{-1}\sum_{n=1}^{N_v}\exp(s_n)\left[1-\exp(s_n)\right]^{-1}\left[P_{n+1}-P_n\right] = \mathrm{H}(\mathbf{s}), \tag{5.14}$$

where N_v is the number of vertical levels, m_n and q_n are defined at the intermediate levels as described in Section (4.3), and **s** is the vector of length N_v of log specific humidities s_n. H is the forward operator relating W to s. Equation (5.14) is nonlinear in s and q.

Now let us construct the linear tangent operator **H**. Suppose that the background-specific humidity is given by $\mathbf{q}_b$ with N_v elements $q_b{}^n$; and the corresponding background log specific humidities are denoted $\mathbf{s}_b$ with elements $s_b{}^n$. Then we can linearize (5.14) about this background-specific humidity profile by noting that increments in the mixing ratio are (to first order) $\Delta m_n = \Delta q_n/(1 - q_b{}^n)^2$ and in specific humidity $\Delta q_n = q_b{}^n\ \Delta s_n$. Then,

$$\Delta W = g^{-1}\sum_{n=1}^{N_v} q_b^n\left[1-q_b^n\right]^{-2}\left[P_{n+1}-P_n\right]\ \Delta s_n = \mathbf{H}\,\Delta\mathbf{s}, \tag{5.15}$$

where **H** is the $1\times N_v$ matrix with elements defined in (5.15), ΔW is the total precipitable water increment, and Δ**s** is a vector of length N_v of increments of the log specific humidity. In (5.15), $\mathbf{q}_b{}^n$, $\mathbf{P}_n$, etc. are all known.1 – q and $(1 - q)^2$ are very close to 1. However, to use (5.15), we are required to have available, whenever needed, complete vertical profiles of the background-specific humidity $\mathbf{q}_b{}^n$, $1 \leq n \leq N_v$ for every single SSM/I total precipitable water observation. It would seem to be an unnecessarily large storage burden to carry around a complete background profile of background-specific humidity for every SSM/I perceptible water observation. Unfortunately, this is a price we pay for using s, rather than q as our humidity variable. However, there is a way around this problem.

$\Delta W = W_r - W_b$, where W_r is the observed SSM/I precipitable water and $W_b = H(s_b)$ following (5.13) is the background precipitable water. Then, replace (5.15) with

$$\Delta W/W_b = \mathbf{H}_*\,\Delta\mathbf{s}, \tag{5.16}$$

where $\mathbf{H}_*$ is the $1\times N_v$ matrix with elements

$$q_b^n\ \left[1-q_b^n\right]^{-2}\left[P_{n+1}-P_n\right] / \sum_{n=1}^{N_v} q_b^n\ \left[1-q_b^n\right]^{-1}\left[P_{n+1}-P_n\right]. \tag{5.17}$$

The elements of $\mathbf{H}_*$ (5.16), unlike the elements of **H** (5.15), are normalized and nondimensional. The elements of **H** would vary substantially between pole and equator, while the elements of $\mathbf{H}_*$ would have a much smaller horizontal variation. We therefore assume (for the present) that the elements (5.17) are horizontally invariant.

Following Sections (3.7) and (5.3.3), we use the scaled form of the 3DVAR equation (3.17). Thus, we now define the (normalized) background precipitable water error (a scalar) $e_w{}^2 = \mathbf{H}_*\mathbf{S}_s{}^{1/2}\mathbf{C}_{ss}\mathbf{S}_s{}^{1/2}\mathbf{H}_*{}^T$, where $\mathbf{S}_s$ is the $N_v\times N_v$ diagonal matrix of background log specific humidity error variances and $\mathbf{C}_{ss}$ is the symmetric $N_v\times N_v$ background log specific humidity error correlation matrix defined in Section 4.3. In the notation of Section 3.7 and Eq. (3.17) in particular, we associate $\mathbf{C}_b{}^{ob/ob}$ with $\mathbf{C}_{ss}$, $\mathbf{S}_b{}^{ob}$ with $\mathbf{S}_s$, and $\mathbf{S}_h$ with $e_w{}^2$. Then, following Section (5.3) and Eq. (5.4) in particular, we define $\underline{\mathbf{H}}$ as

$$\underline{\mathbf{H}} = \mathbf{S}_\mathbf{h}^{-1/2}\mathbf{H}_*\left(\mathbf{S}_\mathbf{b}^{\mathbf{ob}}\right)^{1/2}\underline{\mathbf{E}} = \varepsilon_\mathbf{w}^{-1}\mathbf{H}_*\mathbf{S}_\mathbf{s}^{1/2}\underline{\mathbf{E}}, \tag{5.18}$$

where $\underline{\mathbf{E}}$ is the $N_v \times N_v^s$ matrix of eigenvectors of $\mathbf{C}_{ss}$ corresponding to the N_v^s largest eigenvalues, and $\underline{\mathbf{H}}$ is a $1 \times N_v^s$ matrix that, from the assumption following (5.17), is independent of horizontal location.

For a one-dimensional retrieval, the counterpart (for SSM/I precipitable water) to Eq. (5.5) is

$$\mathbf{s}_a - \mathbf{s}_b = \mathbf{S}_s^{1/2} \underline{\mathbf{EDH}^T}[\underline{\mathbf{HDH}^T} + \varepsilon_w^{-1}\mathbf{R}_w\varepsilon_w^{-1}]^{-1}\varepsilon_w^{-1}[W_r - W_b] / W_b, \tag{5.19}$$

where $\underline{\mathbf{D}}$ is a diagonal $N_v^s \times N_v$ matrix consisting of the N_v^s largest eigenvalues of $\mathbf{C}_{ss}$, $\mathbf{R}_w$ is the SSM/I precipitable water observation error variance, W_r is the observed precipitable water, and $W_b = H(\mathbf{s}_b)$ is the background precipitable water defined above.

Extension from one dimension to three dimensions follows exactly as in Section (5.3), but note that $\underline{\mathbf{H}}$ is a $1 \times N_v$ universal matrix for this instrument.

Table 5.2 shows one dimensional specific humidity correction fields retrieved from a total precipitable water measurement using this algorithm. The normalized precipitable water innovation (5.16) is 0.1. Following (5.19), the log specific humidity correction field is retrieved. For display purposes, this correction field is then converted into a specific humidity correction field (gm/kgm) by multiplying by the standard atmosphere specific humidity vertical profile. There are 32 pressure levels from 1070-10 hPa and thus 32 vertical modes of the background error correlation. We show the specific humidity correction using all 32 modes, seven modes (as in Section 5.3), and two modes.

As would be expected (see also Phalipou and Gerard, 1996), the specific humidity correction decreases with height. Also, not surprisingly since precipitable water is a vertical integral, it does not take very many modes of the background error correlation matrix to represent the specific humidity correction field. Seven modes is more than enough, but two modes clearly has systematic errors at high levels and near the ground.

Figure 5.6 illustrates the analysis of a single SSM/I total precipitable water observation. Temperature, background, and analyzed dew point are plotted as a function of pressure on a skew/logp diagram. In this case, the observed and background precipitable water were 46 and 30.9 mm, respectively. The vertical correlation length (Eq. (4.3)) was 0.35, and the observation and background error for the total precipitable water were the same. In this example, three vertical modes of the background error correlation were used. The

Table 5.2 — Specific Humidity Correction

Pressure Level	Optimal Case (32 Modes)	Seven Modes	Two Modes
50 hPa	0.0000	0.0000	-0.0001
200 hPa	0.0001	0.0001	0.0002
300 hPa	0.0036	0.0036	0.0057
500 hPa	0.0277	0.0278	0.0368
700 hPa	0.0916	0.0915	0.0984
850 hPa	0.1266	0.1265	0.1197
950 hPa	0.1505	0.1506	0.1408
1000 hPa	0.1808	0.1813	0.1657

analyzed total precipitable water came out to be 38.8 mm, which lies between the observed and background values, as would be expected. Figure 5.6 shows that the total precipitable water innovation increment is spread out in the vertical, with the maximum effect in the lower troposphere. The result of Fig. 5.6, with three vertical modes used, is indistinguishable from a result using all 32 vertical modes.

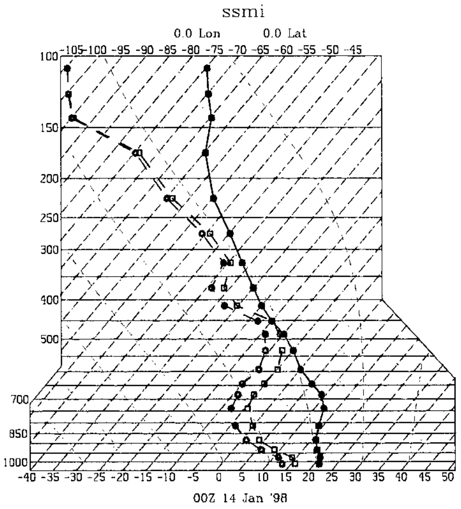


Figure 5.6
Temperature, background, and analyzed dew point from SSM/I total precipitable water

5.7 Cloud Drift and Water Vapor Winds

Cloud drift winds are can be determined from geostationary satellites using automatic tracking of cloud filaments. Distance vectors over short time periods are converted into wind vectors reasonably accurately. The chief problem is in determining the height of the clouds, and therefore assigning accurate heights (pressures) to the cloud drift winds. The pressures can be assigned incorrectly to groups of cloud drift wind vectors in the same vicinity, thus giving rise to the possibility of horizontally correlated observation errors.

Such cloud drift wind observations are treated conventionally by the NAVDAS system. The observations are thinned to minimize horizontal correlations, and the observation errors assigned are fairly large.

Water vapor winds are a more recent innovation. They are obtained by tracking features in several water vapor channels on geostationary satellites, and then converting into wind vectors in a procedure similar to those used for cloud track winds. Since the radiance in the water vapor channels comes from a deep layer rather than a point, these winds actually represent a vertically integrated quantity (with a weight function given by the vertical response of the instrument). In principle, one could determine a forward operator **H** (as in Section 5.3) appropriate for these channels and assimilate these channel radiance vectors directly.

Unfortunately, we do not have the resources to attempt the direct assimilation of these radiance vectors at this time. Consequently, we assimilate the derived wind vectors, using the provided height assignments. Thinning is also used for these observations.

5.8 Pilot Reports and AMDAR Observations

Conventional pilot reports that occur along standard aircraft tracks are used in a conventional fashion.

AMDAR (aircraft meteorological data reporting) observations taken on wide-bodied aircraft and then up-linked to satellites are an important new source of information. Where available, they are given higher weight and precedence over conventional aircraft reports. There are many sources of error in these observations, and they must be carefully quality controlled. Patricia Pauley has devoted an enormous amount of time identifying the various kinds of error and, in some cases, correcting them (Pauley and Stephens, 1998).

We divide AMDAR observations into three categories: ascents, level flight, and descents. We treat level flight information in much the same manner as aircraft reports. Descents are not used much except in the absence of other information. Ascents are treated as vertical profiles. For lower resolution analyses such as those for NOGAPS, we may ascribe a single latitude and longitude to the whole ascent and treat it much like a radiosonde-using vertical eigenvector decomposition, which has a huge computational advantage, as in Section (4.4). For higher resolution analyses such as those for COAMPS, we retain the option of treating each observation of the ascent as a single observation with its correct latitude and longitude. This is more expensive, but more accurate, and expense is not such an important consideration for regional analyses.

Over a 6-hour period, a number of AMDAR ascents may occur from the same airport. It would be foolish to include all the ascents, so we choose the best ascent based on: minimum deviation from vertical; time proximity; passes all the quality control checks; previous history of the particular aircraft; etc.

5.9 Scatterometer Winds

Scatterometers, such as ERS-1 and QUICKSCAT can estimate windspeed and direction at the ocean surface by measuring the radar return from surface capillary waves. The wind directions are ambiguous, however, with up to four possible wind directions. The retrieval algorithm itself can rank these directions in order of decreasing likelihood. However, the background 10-m wind can also be used to pick the most likely direction (Stoffeln and Anderson, 1997). It is important to use a background wind that is at the appropriate time and for the right model (NOGAPS or COAMPS).

Choosing the wind direction as the one closest to the background is a good choice most of the time, but it is not always the best choice. ECMWF has used a filtering procedure called PRESCAT, which will occasionally suggest that the direction closest to the background wind direction is less optimal than one of the other three choices.

In NAVDAS at this time, the wind direction is chosen purely on the basis of being closest to the background wind direction. However, Appendix H describes an iterative preprocessing procedure based on minimizing the J_{min} diagnostic of Section (9.1). This procedure usually picks the direction closest to the background wind direction as most likely, but consistency between the innovations and specified background and observation error statistics occasionally picks one of the other three directions.

6. Nonlinear Instrument (Forward) Operators

Section 5 introduced several forward operators that were actually nonlinear, even though only their linearized forms were discussed at that point. Such operators are the radiative transfer operators used in TOVS retrievals (Sections 5.3 and Appendix G) and the wind speed operator (Section 5.5). We now consider how nonlinear forward operators can be accounted for.

Using the notation of Section 2, consider a sequence of state estimate vectors $\mathbf{x}_0$, $\mathbf{x}_1$, $\mathbf{x}_2$,... $\mathbf{x}_i$ and a nonlinear forward operator H($\mathbf{x}$). Now define a linearized operator (linearized about the state estimate $\mathbf{x}_i$) given by

$$\mathbf{H}_i = \delta H(\mathbf{x})/\delta\mathbf{x} \text{ at } \mathbf{x} = \mathbf{x}_i. \tag{6.1}$$

The observation and background vectors are $\mathbf{y}$ and $\mathbf{x}_b$, the observation and background error covariances are $\mathbf{R}$ and $\mathbf{P}_b$. Then, the nonlinear extension to Eq. (2.6) can be written (following Tarantola (1987, p. 244)) as

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{P}_b\mathbf{H}_i^T [\mathbf{H}_i\mathbf{P}_b\mathbf{H}_i^T + \mathbf{R}]^{-1} [\mathbf{y} - H(\mathbf{x}_i) + \mathbf{H}_i\mathbf{x}_i - \mathbf{H}_i\mathbf{x}_b]. \tag{6.2}$$

Equation (6.2) defines an iterative procedure that begins with an initial guess $\mathbf{x}_{0;}$ this is usually chosen to be $\mathbf{x}_b$ but does not have to be. The iterative process then continues until $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ are very similar, in which case the procedure is said to converge. Convergence cannot be guaranteed. Moreover, this is a nonlinear, not a linear, procedure, and there is no guarantee that a convergent solution is necessarily unique. In other words, the cost function may have many minima, and the minimum that is found may depend on the choice $\mathbf{x}_0$. Behavior of this sort can be expected as the operator H($\mathbf{x}$) becomes increasingly nonlinear.

What has been done in Eq. (6.2) is to put the whole 3DVAR procedure described in Sections 2 and 3 inside a loop. We refer to the iteration (6.2) as the outer iteration. This name is to distinguish it from the inner iteration, which is the iteration performed by descent algorithms such as conjugate gradient (described in Section 3.2) to perform the matrix solve $\mathbf{H}_i\mathbf{P}_b\mathbf{H}_i^T + \mathbf{R}$ at each of the outer iterative steps.

It is very straightforward mechanically to take the linearized NAVDAS algorithm described in Section 3 and put it in an outer loop to attempt to obtain a nonlinear solution. This procedure will inevitably be more costly than a linearized solution, and it may not be any better. If only a small proportion of the observations are nonlinear, the outer iteration process can be substantially sped up. This is discussed in the next section.

6.1 Practical Implementation in the NAVDAS Algorithm

We can avoid a complete outer iteration around both equations ((3.5) and (3.6)). After we have obtained $\mathbf{z}$ from Eq. (3.5), we operate on it in such a way as to obtain a vector of corrections at the observation locations that are connected with any nonlinear operators, except in the analyzed variables (u,v,T, etc.). To see this, consider Eq. (3.17) and re-write as

$$\mathbf{x}_a - \mathbf{x}_b = \mathbf{S}_b^{1/2}\mathbf{C}_b^{gr/ob}[\mathbf{S}_b^{ob}]^{1/2}\mathbf{H}^T\mathbf{z}, \tag{6.3}$$

where $\mathbf{z} = \mathbf{S}_h^{-1/2}[\mathbf{C}_h^{ob/ob} + \mathbf{S}_h^{-1/2}\mathbf{R}\mathbf{S}_h^{-1/2}]^{-1}\mathbf{S}_h^{-1/2}[\mathbf{y} - \mathbf{H}(\mathbf{x}_b^{ob})]$.

These operations produce a correction field $\mathbf{x}_a - \mathbf{x}_b$, which is for the analyzed variables (u,v,T etc) but at the analysis grid points. What we want are the corrections in the analyzed variables, but at the observation locations.

At the end of a linear solve, we have a vector **z** that is in observation space. Now define the background error correlation $\mathbf{C}_b^{ob/ob}$, which is defined at the observation locations, but for the analyzed variables. Similarly, define $\mathbf{S}_b^{ob}$ as the corresponding error variances of the analyzed variables at the observation locations. Then we can define a correction vector at the observation locations as

$$[\mathbf{x}_a - \mathbf{x}_b]^{ob} = [\mathbf{S}_b^{ob}]^{-1/2}\mathbf{C}_b^{ob/ob}[\mathbf{S}_b^{ob}]^{-1/2}\mathbf{H}^T\mathbf{z}. \qquad (6.4)$$

Then, since we already have available the background value defined at the observation location, it is straightforward to get the new analyzed value at the observation location. This can then be used to recalculate the tangent linear model, and we can proceed to the next nonlinear iterate.

Since the observation space is much smaller than the observation space, and moreover, the number of observations that have a nonlinear forward operator is even smaller, it is clearly much more efficient to use (6.4) than to use (6.3) followed by an interpolation from grid point to observation locations. In fact application of (6.4) is about as costly as one multiplication by the matrix **A** in the conjugate gradient operator (3.12).

We now start with a very simple (but nonetheless relevant) nonlinear forward operator-the wind speed operator.

6.2 SSM/I Wind speed (Nonlinear)

Let us begin the discussion of this operator by reconsidering the simple problem introduced in Section 5.5, namely, producing analyzed wind components u_a and v_b from a single observed wind speed w_r and background wind components u_b and v_b. As in Section 5.5, we define the observed wind speed error variance as ε_r^2 and the background wind component error as ε_v^2. Then, the cost function appropriate for this situation is given by

$$J = 0.5\{\varepsilon_r^{-2}(w_r - H(u_a,v_a))^2 + \varepsilon_v^{-2}(u_b - u_a)^2 + \varepsilon_v^{-2}(v_b - v_a)^2\}, \qquad (6.5)$$

where $H(u,v) = (u^2 + v^2)^{1/2}$. Taking the gradient dJ/du_a yields

$$\delta J/\delta u_a = -\varepsilon_v^{-2}(u_b - u_a) - \varepsilon_r^{-2}(w_r(u_a^2 + v_a^2)^{-1/2} - 1)\,u_a, \qquad (6.6)$$

with a similar equation for $\delta J/\delta v_a$. Setting $\delta J/\delta u_a = \delta J/\delta v_a = 0$ (and some manipulation) yields the estimates u_a and v_a that minimize the nonlinear functional (6.5), viz.,

$$u_a = \gamma u_b,\ v_a = \gamma v_b, \text{ where } \gamma = (\varepsilon_v^2 + \varepsilon_r^2)^{-1}(\varepsilon_r^2 + \varepsilon_v^2 w_r(u_b^2 + v_b^2)^{-1/2}). \qquad (6.7)$$

Equation (6.7), which was obtained for a nonlinear minimization, is exactly the same as Eq. (5.11), which arose from the linearized case. From this result, we can see that the analysis estimate for this simple one observation case is the same whether obtained linearly or nonlinearly. In other words, for this case the nonlinear iteration procedure (6.2) would converge in a single (outer) iteration. The resulting analyzed wind speed would be a linear combination of the observed and background wind fields (5.12) and the analyzed wind direction would be taken from the background. In the special case where the wind speed observation was perfect, the analyzed and observed wind speeds would be the same.

Now, in practice, we have many SSM/I wind speed and also other forms of wind data such as surface and upper air wind component observations and surface scatterometer data. It seems reasonable to suppose, that in the presence of other observations, we might be able to obtain better estimates of the wind direction than that given by the background field. We pose this question in a very simple context and then perform an experiment to answer it. Consider the two-dimensional horizontal assimilation problem and suppose we have available only SSM/I wind speed observations. We know from (6.5)-(6.7) that if we consider each wind speed observation separately, we cannot analyze the wind direction any more accurately than the background wind. The question is, if we analyze all the wind speed observations together, can we produce a wind direction analysis that is more accurate than that obtained from the background field alone. In other words, can we extract wind direction information from a large number of wind speed observations?

To examine this question, we performed the following experiment. We created an (x,y) grid with 15 grid points in each direction. We assumed that at each analysis grid point we had a background u and v wind component. At each grid point, we also had a wind speed w estimate. Thus, the observation location and analysis grid points coincided. We assumed that the wind speed observations were perfect, $\varepsilon_r^2 = 0$ at each observation point (except in Table 6.3). This meant that we did not have to worry whether or not the wind speed observational errors were horizontally correlated. We would therefore expect that the wind speed analyses would also be perfect (and this was verified).

We first created a "true" wind field - u and v wind components at every grid point. These fields were created with a gaussian random number generator and contained both rotational and divergent wind components.

The background error covariance and the background field were constructed as follows. The background wind error was assumed to be nondivergent and to have a characteristic horizontal scale L_h. The background error covariance $\mathbf{P}_b$ was consistent with this assumption. The background field itself was constructed by adding a perturbation that was consistent with $\mathbf{P}_b$ to the "true" wind field. In other words, the background error was nondivergent, random red noise (with characteristic scale L_h). The background field itself was not nondivergent, because the "truth" was not nondivergent.

Define $\overline{\mathbf{v}_t}$ as the true wind speed averaged over the domain and $\overline{\varepsilon_v}$ as the rms background wind error. Then, define $\alpha = \overline{\varepsilon_v} / \overline{\mathbf{v}_t}$ as the ratio of the rms background vector wind error to the true wind speed. As α approaches 0, the problem becomes increasingly linear, and when $\alpha = 1$, the problem is very nonlinear. We now examine the wind direction error as a function of the two parameters α (measuring the nonlinearity) and L_h (measuring the horizontal scale of the background error).

We choose two values of α, $\alpha = 0.1$ and $\alpha = 1.0$. We first consider the case $L_h = 0$, which should yield the same results as Equations (6.5)-(6.7). For each case, the results from an ensemble of 10 members were averaged (each ensemble member had different random numbers). The nonlinear analysis used (6.2) with 10 iterations, and the linear case used a single iteration. The wind speed observations were assumed to be perfect.

To put these numbers in perspective, it has been estimated (Phalipou, 1996) that the ECMWF background wind direction error is about 20 degrees. A wind direction error of 90 degrees indicates no information. In calculating the wind direction error, we ignored all grid points where the true wind speed was very small and the wind direction indeterminate.

We can see from Table 6.1 that even for the large background error case ($\alpha = 1.0$), there is some (not much) information in the background field. However, neither linear nor nonlinear iteration are able to improve on the background estimate. When $L_h = 0$, the background error is spatially uncorrelated, the nondivergence constraint has no meaning, and the results are exactly as predicted from equations (6.5)-(6.7).

As L_h is increased, the nondivergence constraint in the background error comes into play, and the observations start to interact with each other. Table 6.2 shows the results with $L_h = 3\Delta x$, where Δx is the grid length (same in both directions). As in Table 6.1, the wind speed observations are perfect.

It can be seen that in this case, both the linear and nonlinear algorithms manage to produce smaller wind direction errors than the background. The differences between linear and nonlinear results are small when the background error is small, but the nonlinear result is relatively much more accurate when the background error is large.

Table 6.2 has been derived assuming perfect wind speed observations. We would expect some deterioration when the observed wind speeds are in error. Assume that the observed wind speed error is equal to $0.5\,\overline{\varepsilon_v}$, that is, the observed wind speed error is smaller than the background wind error, but it is not negligible. However, we assume that the observation wind speed errors are not spatially correlated. Table 6.3 shows the results for this case when $L_h = 3\Delta x$, as in Table 6.2.

Table 6.1 — Wind Direction Error (degrees) - L_h=0

	Background	Analysis (linear)	Analysis (nonlinear)
$\alpha = 0.1$	8.4	8.4	8.4
$\alpha = 1.0$	57.6	57.6	57.6

Table 6.2 — Wind Direction Error (degrees) - L_h=3Δx

	Background	Analysis (linear)	Analysis (nonlinear)
$\alpha = 0.1$	8.7	4.9	4.5
$\alpha = 1.0$	60.8	54.3	43.7

Table 6.3 — Wind Direction Error (degrees) - L_h=3Δx

	Background	Analysis (linear)	Analysis (nonlinear)
$\alpha = 0.1$	8.7	5.9	5.8
$\alpha = 1.0$	60.8	54.9	52.6

Our conclusion is that some wind direction information can be extracted from SSM/I wind speed observations (even if imperfect) when the background error is nondivergent and red and the background error statistics are correct.

To complete this section, we compare the results of Table 6.2 with an ad hoc, traditional method of assimilating SSM/I wind speed observations. In this procedure, the wind speed observations are turned into wind component pseudo-observations by taking wind directions from the background field. This is very straightforward. The most obvious difficulty is in determining a suitable observation error covariance matrix **R**. Because the wind direction of these pseudo-observations comes from the background field, the observation error is both spatially correlated and correlated with the background error. Traditionally, both these correlations are ignored and **R** is assumed to be diagonal. Note that this is a linear, not a nonlinear assimilation.

Since this is an ad hoc method, we also make an ad hoc assumption about **R**, namely, that $\mathbf{R} = \beta \overline{\varepsilon_v^2}\, \mathbf{I}$, where **I** is the identity matrix, $\overline{\varepsilon_v^2}$ is the domain-averaged rms background wind error defined above, and β is a constant. Normal values for β would lie in the 0.1 to 1.0. range. Table 6.4 shows the results of an experiment with $\beta = 0.5$ for the same case shown in Table 6.2, namely, perfect wind speed observations and $L_h = 3\Delta x$.

Table 6.4 shows that the analysis wind direction error (even with perfect wind speed observations) shows no improvement over the background wind direction error. In fact, when the background wind direction is reasonably accurate ($\alpha = 0.1$), the analyzed wind direction may even be worse. The same general conclusion is true for all settings of β.

Table 6.4 — Wind Direction Error (degrees) – L_h=3Δx

	Background	Analysis (linear)
$\alpha = 0.1$	8.7	11.1
$\alpha = 0.1$	60.8	54.9

Comparison of Tables 6.2 and 6.4 shows that direct assimilation of wind speeds may be able to improve the background estimate of wind direction, but the assimilation of wind component pseudo-observations (even when the observed wind speeds are perfect) gives no new wind direction information (at best).

This algorithm was implemented in the NAVDAS code using Eq. (6.4). It was tested in the global problem using real SSM/I observations and in the presence and absence of other observations.

7. Input Processing — Background and Observation Ingest and Quality Control

The data flow for NAVDAS has been designed to work in any of the expected environments required by the Navy. The backbone of the system is the innovation vector file, which carries all of the information needed for analysis and display of the observations. The routines that read the observations and background fields from the database are isolated from the rest of the code so that adapting NAVDAS to any computer environment requires only applying the appropriate interface software. The multiple interfaces are indicated by several versions of the observation and field reading routines in the data-flow diagram, Fig. 7.1.

Once the fields and observations are read, the innovation, which is observation minus (forward interpolated) background (Section 2.5), is computed using a cubic spline interpolator in the horizontal, a linear interpolator as a function of pressure in the vertical, and the Lagrange interpolator in time using three time levels of background fields. The backgrounds used are defined on pressure coordinates. The function of pressure used in the vertical interpolation can be one of several choices to be selected by the user, including ln p, p^{κ}, and p^{a}, where $\kappa = 2/5$, and $a = 0.2051$. These choices are accurate for isothermal, isentropic, or standard atmosphere conditions, respectively.

The data quality control routines are rule-based and derived from experience with the various types of observations. One can learn the errors that typically occur in various data sets by studying the observations and understanding how the measurement is taken. For example, data entry, transmission, and instrument calibration problems cause recurring errors in radiosonde data that can be detected with software. Position reporting and stuck instruments cause aircraft and ship errors that are detectable, and data distribution centers cause errors in duplication and time labeling.

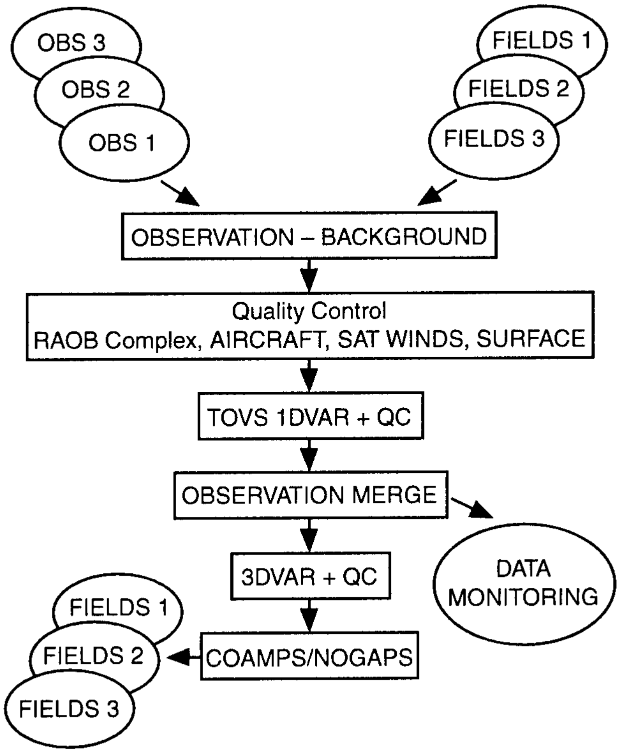


Figure 7.1
NAVDAS data flow diagram

The rawinsonde checks are made using the complex method of Collins and Gandin (1992), which does a gross, hydrostatic, and baseline check along with vertical and horizontal checks using the optimal method described by Lorenc (1981). Duplicates are removed prior to performing these checks. The baseline check ensures that the base of the sounding is consistent with the station's elevation and reported surface pressure. This code also attempts to unscramble entries made in data entry by transposing digits and then rerunning the checks to determine whether the quality scores improve. The final step in this process removes the biases caused by radiation contamination of the temperature sensor.

The aircraft data are exposed to a wide variety of position consistency checks, along with checks for stuck clocks and instruments, see Pauley and Stephens (1998).

The satellite radiance data are subjected to a wide variety of tests to ensure that the forward model is capable of handling the conditions of observation. Examples of data rejection include clouds in the path of view, hot desert surfaces, snow and ice on the ground, and fields of view that cannot be classified as either land or sea, i.e., the instrument is seeing a combination of both land and water, see Baker (1999).

Checks are also planned for the satellite-derived winds to determine errors around mountains, zones of high vertical wind shear, and along coasts where the sea breezes may cause unrepresentative errors.

To take advantage of multiprocessors available on the workstations, separate programs are run for each data type. After the preprocessing is complete, a program is called to merge the innovation vector files for the various data types into a single file, and then to edit out the rejected data from the assimilation run. This routine can also withhold observations from the run for testing purposes.

Finally, the innovation vector is compressed and stored for post processing operations such as computing background error statistics, locating malfunctioning platforms and instruments, and measuring the success of the data assimilation algorithm.

Data preparation software described in this section runs on specialized data handling workstations and the resulting innovation vectors are ported to the massively parallel computers that run the analysis and models. On workstations, the processing runs sufficiently fast to be rerun after each data cut.

Likewise, when running NAVDAS at regional centers or at remote sites, central-site support will be required in the form of boundary conditions for the regional model. In addition, the global innovation vector file can be supplied to ensure that the remote analysis contains all of the information available at central site, plus any data collected locally. Besides ensuring consistency in the data used at both sites, this will enable the remote site to use observations outside the regional model's grid since the global model's background information will already be contained in the innovation vector file.

In summary, the front-end processor prepares the innovation vector along with all the other information needed by the analysis and display software, including quality control checks, observation location, and various other associated information needed for processing. Table 7.1 lists the makeup of the innovation vector. The header for this file defines all of the parameters required for specification of grid parameters and transformations using the utility routines, plus it gives the source for the background fields. The file is formatted in simple ASCII so that any editor can be used to look at the observations directly. A data-monitoring module is being built to display the innovation vector and associated information using a web browser. This will enable customers to view the data that has gone into the data assimilation, and to quickly detect problems related to library update errors, instrument malfunction, and data transmission. This approach will improve productivity of the data users and provide more information to a wider audience, thereby improving the quality of the final product.

Table 7.1 — Makeup of the Innovation Vector

Observation	Measured value in mks units
Background	Forecast background in mks units
Backgroundtemperature	Background temperature at observation location, needed to compute isentropic surface
Observation error	Observation error in mks units
Latitude	Observation location
Longitude	Observation location
Pressure	Observation location
Variable type	e.g., wind, temperature, pressure height, logarithm of specific humidity, or brightness temperature
Instrument type	e.g., rawinsonde, MDCRS, AIREP, SATDAT, TOVS
Number of values in profile	e.g., for rawinsonde, 145 means the next 144 entries belong to the same instrument with the same latitude and longitude
Quality control check	An unique integer for each check made
Time from analysis time	e.g., 3600 means observation taken 3600 s prior to analysis time
Platform identifier	e.g., 72468 –9 –9aRTD02 for rawinsonde gives station number, instrument type, and level type
Database identifier	T_RAOB3455 would mean data was pulled from tfile at line 3455. This designation will be modified to suit DBMS in use and is needed to set quality control designators in the DBMS.
Et cetera	This varies; in some cases it is the precorrected value, and in the case of SSMI winds, it is the v-component of the background wind, while the u-component of the background field is stored in the background location.

8. Output Processing — Interface with COAMPS and NOGAPS

As described in Section 4.8, the analysis corrections are stored while they are still projected onto the vertical eigenvectors of the background error correlation. The interface to the model involves transforming these analyses onto the pressures defined on the model grid. Handling the corrections this way saves storage space of the analysis while retaining its accuracy. Transforming corrections onto the model's grid is nonlinear because updating temperature and specific humidity also changes the pressures at the grid points; therefore, an iterative method is used to transform the corrections onto the updated pressure values.

8.1 COAMPS Interface

To explain the steps required to update COAMPS, we introduce its hydrostatic equation (see Hodur, 1997). The vertical coordinate uses a sigma-z system defined as

$$\sigma = z_{top}\left(\frac{z - z_{sfc}}{z_{top} - z_{sfc}}\right), \tag{8.1}$$

to specify heights. The mass variables are carried in potential temperature θ, specific humidity q_v, and the Exner function of pressure $\pi = \left(\frac{p}{p_{00}}\right)^{R/c_p}$. Each variable is partitioned into a mean state, $\overline{(\ )}$, and departure from this mean, $(\ )'$. Presently, the mean state potential temperature is derived from the standard atmosphere, and the mean Exner function is computed using the hydrostatic relationship

$$G_z\frac{\partial\bar{\pi}}{\partial\sigma} = -\frac{g}{c_p\bar{\theta}_v}, \tag{8.2}$$

where c_p is the specific heat at constant pressure for the atmosphere, θ_v is the virtual potential temperature = $\theta(1 + 0.608q_v)$, g is the acceleration due to gravity, and

$$G_z = \frac{\partial\sigma}{\partial z} = \frac{z_{top}}{z_{top} - z_{sfc}}.$$

Simplifying the equation for vertical motion (Eq. (4) of Hodur, 1997) by assuming a hydrostatic balance gives

$$c_p\theta_v G_z\frac{\partial\pi'}{\partial\sigma} = g\left(\frac{\theta'}{\bar{\theta}} + 0.608q_v\right), \tag{8.3}$$

which is the equation needed to compute π' from θ' an q_v.

In the COAMPS version of NRL MVOI, the output was pressure height corrections that were interpolated to the model's coordinates and used to compute temperature. In NAVDAS, temperature is analyzed, with geopotential being computed from the temperatures using the hydrostatic equation (Section 4.3.6).

The update procedure is to first transform the temperature, specific humidity, and geopotential onto the background defined by the pressure at each grid point. Then these values are used to compute surface pressure corrections using the following form of the hydrostatic equation:

$$\Delta p_s = \tilde{p}_s\{\exp(\frac{\Delta\Phi_s}{RT_v}) - 1\}, \qquad (8.4)$$

where the tilde designates background values, and $\Delta\Phi_s$ is the geopotential height analysis correction defined at the background surface pressure $\tilde{p}_s$. The surface temperature T_v is estimated using θ' at the model's lowest mass level, which is one half-grid interval from the surface and $\tilde{\theta}$ defined at the surface. Equation (8.4) does not require a precise definition $\tilde{T}_v$ because errors in surface temperature of around 10°C produce errors in pressure correction that are less than 0.1hPa.

After surface pressure is updated, it is used as the reference level in the integration of Eq. (8.3) to compute updated values of π' from θ' and q_v. This new value π' is then used to compute pressure on the model coordinates so that the entire update procedure can be repeated in an iterative fachion. The updates reach their asymptotic value after only two iterations.

What has just been described is an update system that ties the model to the analysis of temperature and surface pressure. This means that the largest errors in the mass variables will occur at the model's top. At issue is the model's vertical coordinate, which is in height, but it is more susceptible to truncation error than the pressure coordinate system used in NAVDAS.

In a series of experiments using varying vertical resolution and a hypothetical 1°C temperature correction at all levels, Eq. (8.3) was integrated downward to compute hydrostatically consistent corrections to surface pressure. Figure 8.1 shows the surface pressure corrections for each grid, along with the asymptotic value determined using 4,000 levels. In these tests, all grids were evenly spaced, with the model top set at 17,000 m, which is about 80 hPa on the standard atmosphere. The errors are about 10% for a 10-level model and 5% for 30 levels. The vertical location of this truncation error depends on where the reference level is set. In these experiments, the reference level was set at the model's top so that the truncation error would be at the bottom. In our updating procedure, the reference level is set at the surface. A similar set of experiments using the hydrostatic equation of Section 4.3.6 to compute geopotential at the model's top reached asymptotic values after only 20 levels.

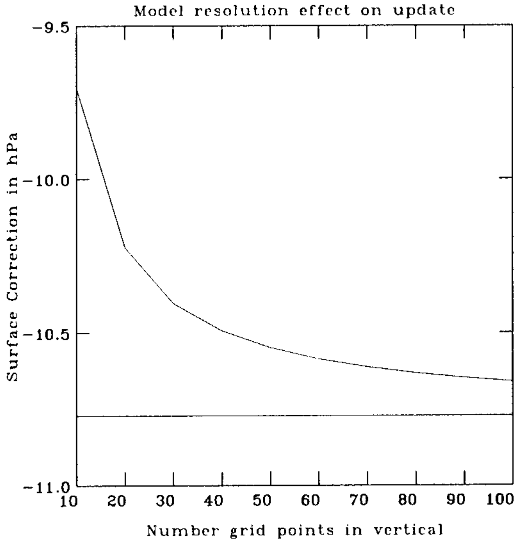


Figure 8.1
Accuracy of COAMPS hydrostatic equation as a function of number of levels compared to a 4,000 level model, which represents the asymptotic solution

8.2 NOGAPS Interface

The interface for NOGAPS is simpler than the COAMPS interface because NOGAPS uses the sigma coordinate system (see Hogan and Rosmond, 1991) so that the vertical coordinate can be easily determined in terms of pressure. For interpolation, the pressure on the model's sigma surfaces is determined from the updated surface pressure using (8.4). Prior to addition of the analysis corrections, however, the background values are interpolated to the updated coordinate system using a cubic spline fit of the variables. This is done in place of the iteration step used in the COAMPS interface.

For NOGAPS, the current version of NAVDAS generates the analysis corrections projected onto the vertical eigenvectors of the background error correlation on a 1-degree spherical grid. NAVDAS will eventually analyze on the gaussian model grid directly, but for now the 1-degree fields are interpolated horizontally prior to transforming them onto the updated pressure coordinates.

Prior to the forecast integration, NOGAPS initializes the corrections with a normal mode method described in Hogan and Rosmond (1991).

9. Internal Diagnostics

This section has two purposes. The first is to describe some internal diagnostics that are calculated while we are determining the corrections from the innovations. These diagnostics can be useful in tuning the background and observation error statistics and in examining the properties of the assimilation algorithm. The second purpose is to describe the "buddy check" or consistency test, which is used to determine whether any of the observations are inconsistent with their neighbors. The buddy check involves interactions between observations and is performed in the middle of the iteration procedure of Section 3.2.6.

9.1 The J_{min} Diagnostic

Following Section 2, define L as the number of observations, $\mathbf{P}_b$ as the background error covariance, $\mathbf{R}$ as the observation error covariance, $\mathbf{H}$ as the linearized forward operator, and $[\mathbf{y} - H(\mathbf{x}_b)]$ as the innovation vector. Then, define J_{min} as the scalar

$$J_{min} = [\mathbf{y} - H(\mathbf{x}_b)]^T[\mathbf{HP}_b\mathbf{H}^T + \mathbf{R}]^{-1}[\mathbf{y} - H(\mathbf{x}_b)]. \tag{9.1}$$

Then, if the observation and background errors are normally distributed, it can be shown (Bryson and Ho, 1975; Menard et al., 1999), that the conditional mean of J_{min} is equal to the number of observations L. Thus, if we have specified $\mathbf{P}_b$ and $\mathbf{R}$ in a manner consistent with the actual innovations (in a statistical sense), then J_{min}/L should be equal to 1. If $J_{min}/L < 1$, then perhaps the specified observation or background error variances are too large. Alternatively, if $J_{min}/L > 1$, then the specified observation or background error variances are too small, or alternatively, the innovation vector may contain observations which are erroneous. Equation (9.1) can also be written

$$J_{min} = \mathbf{d}^T\mathbf{z}\ , \tag{9.2}$$

where $\mathbf{d}$ and $\mathbf{z}$ are given in Eq. (3.5). Both $\mathbf{z}$ and $\mathbf{d}$ are available and can be used in a straight scalar product to calculate J_{min}.

J_{min} is the cost function

$$\mathbf{J} = [\mathbf{y} - \mathbf{hx}_a]^T\mathbf{R}^{-1}[\mathbf{y} - \mathbf{Hx}_a] + [\mathbf{x}_a - \mathbf{x}_b]^T\mathbf{P}_b^{-1}[\mathbf{x}_a - \mathbf{x}_b]$$

(see Eq. (2.1)) at the minimum. That is, when

$$\mathbf{x}_a = \mathbf{x}_b + \mathbf{K}[\mathbf{y} - \mathbf{Hx}_b] \text{ with } \mathbf{K} = \mathbf{P}_b\mathbf{H}^T[\mathbf{HP}_b\mathbf{H}^T + \mathbf{R}]^{-1}.$$

This can be seen by noting that

$$\mathbf{x}_a - \mathbf{x}_b = \mathbf{K}[\mathbf{y} - \mathbf{Hx}_b] \text{ and } \mathbf{y} - \mathbf{Hx}_a = [\mathbf{I} - \mathbf{HK}][\mathbf{y} - \mathbf{Hx}_b],$$

and substituting in, giving

$$\mathbf{J} = [\mathbf{y} - \mathbf{Hx}_b]^T[(\mathbf{I} - \mathbf{HK})^T\mathbf{R}^{-1}(\mathbf{I} - \mathbf{HK}) + \mathbf{K}^T\mathbf{P}_b^{-1}\mathbf{K}][\mathbf{y} - \mathbf{Hx}_b].$$

But,

$\mathbf{I - HK = R[HP_bH^T + R]^{-1}}$, and $\mathbf{K^TP_b^{-1}K = [HP_bH^T + R]^{-1}HP_bH^T[HP_bH^T + R]^{-1}}$,

that gives

$\mathbf{J = [y - Hx_b]^T[HP_bH^T + R]^{-1}[y - Hx_b]} = J_{min}$

from Eq. (9.1).

The J_{min} diagnostic, which costs nothing to compute, allows us to estimate the value of the cost function (2.1) while remaining in observation space. It also serves as the basis for our innovation and buddy checks in Section 9.3. Appendix I illustrates practical use of this diagnostic.

9.2 The Synthetic Residual Vector

In the estimation literature, $[\mathbf{y} - H(\mathbf{x}_b)]$ is known as the innovation. But $[\mathbf{y} - H(\mathbf{x}_a)]$, where $\mathbf{x}_a$ is the analysis vector, is known as the residual vector. This can be calculated at the end of the algorithm described in Section 3.2.2, that is, following the post-multiplication. However, it is also possible to obtain a very good estimate of the residual immediately after the completion of the iteration process of Section 3.2.1 from the vector **z** of Eq. (3.5). This synthetic or approximate residual vector is then very useful in assessing the fit of the analysis to specific observations or observation systems. The synthetic residual can be derived as follows. Define $\mathbf{x}_a,\mathbf{x}_b$ as the vectors of length I of the analyzed and background values on the analysis grid. Define **y** as the vector of observation of length L and **H**, $\mathbf{P}_b$, and **R** as the linearized forward model and the background and observation error covariances, respectively. Then, from Eqs. (3.5)-(3.6), we have

$$\mathbf{x}_a = \mathbf{x}_b + \Delta\,\mathbf{x}, \text{ where } \Delta\,\mathbf{x} = \mathbf{P}_b\mathbf{H}^T[\mathbf{HP}_b\mathbf{H}^T + \mathbf{R}]^{-1}[\mathbf{y} - H(\mathbf{x}_b)]. \quad (9.3)$$

Operate on (9.3) using the forward operator H. $H(\mathbf{x}_a) = H(\mathbf{x}_b + \Delta\,\mathbf{x}) = H(\mathbf{x}_b) + \mathbf{H}\,\Delta\,\mathbf{x}$ to first order using Eq. (2.3). Thus,

$$\mathbf{y}_a \approx H(\mathbf{x}_b) + \mathbf{HP}_b\mathbf{H}^T[\mathbf{HP}_b\mathbf{H}^T + \mathbf{R}]^{-1}[\mathbf{y} - H(\mathbf{x}_b)], \quad (9.4)$$

where $\mathbf{y}_a = H(\mathbf{x}_a)$ is the vector of length L of analyzed values at the observation locations and of the same variable as the observation. Thus if **y** were a vector of radiances, then so would be the vector $\mathbf{y}_a$. Then, subtract both sides of (9.4) from the observation vector **y** and reorganize, noting that $\mathbf{HP}_b\mathbf{H}^T[\mathbf{HP}_b\mathbf{H}^T + \mathbf{R}]^{-1} = \mathbf{I} - \mathbf{R}[\mathbf{HP}_b\mathbf{H}^T + \mathbf{R}]^{-1}$, where **I** is the identity matrix. The result is

$$\mathbf{y} - \mathbf{y}_a \approx \mathbf{R}[\mathbf{HP}_b\mathbf{H}^T + \mathbf{R}]^{-1}[\mathbf{y} - H(\mathbf{x}_b)] \approx \mathbf{Rz}, \quad (9.5)$$

where **z** is defined in Eq. (3.5).

Now the true residual is obtained by horizontally interpolating the analysis vector $\mathbf{x}_a$ to the observation locations and then using the nonlinear forward operator H to obtain $H(\mathbf{x}_a)$ in observation space and then subtracting from the observations. Equation (9.5) is an approximation to the true residual because it does not explicitly invoke the horizontal interpolation operator ($\mathbf{H}_*$ in Eq. (3.15)). Experiments indicate that (9.5) is a good approximation to the true residual, but its magnitude is usually an underestimate.

If we subtract (9.5) from the observation vector **y**, we obtain $\mathbf{y}_a$, which is an estimate of the analysis in observation space.

Figures 9.1 and 9.2 illustrate the three observation vectors **y** (the actual observations), $\mathbf{y}_b = H(\mathbf{x}_b)$ (the background field projected into observation space), and $\mathbf{y}_a$ (the analysis projected into observation space and obtained from Eq. (9.5)). This is done for an analysis of eastern North America performed with the NAVDAS system and shows a particular radiosonde (Cape Kennedy, Florida). Plotted are both the temperatures (solid) and dewpoints (dashed). Figure 9.1 uses only mandatory level data, and Fig. 9.2 uses the significant levels as well. The effect of using the significant level data is particularly noticeable in the dewpoint. The actual observations **y** are shown with open blue circles, the background values (in observation space) are black solid triangles, and the analyzed values (in observation space) are open red circles.

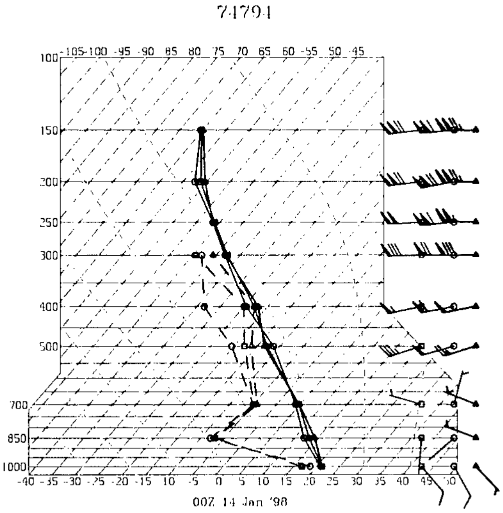


Figure 9.1
Illustration of synthetic residuals (mandatory levels only)

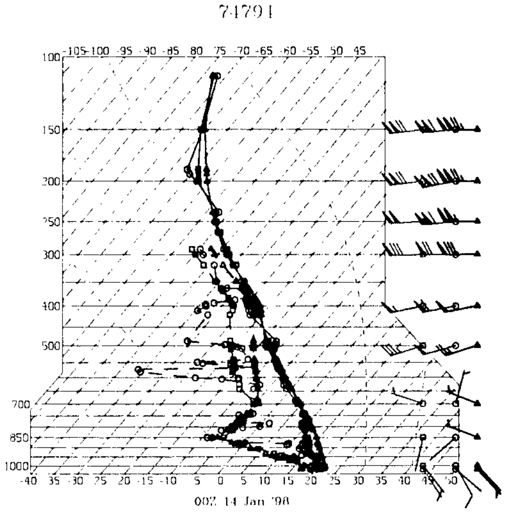


Figure 9.2
Same as Figure 9.1, except includes significant level data

9.3 The Buddy Check Algorithm

The buddy check or consistency test is a form of quality control that must be done in the heart of the assimilation algorithm. This is because the buddy check should check every observation against every other observation to see if the observation is consistent with the other observations. To do this requires a detailed knowledge of the specified background and observation error covariances, which is available only in the middle of the algorithm.

In the buddy check, we devise a metric that involves other observations and the background and observation error statistics. We then test each observation against this metric and decide whether or not to reject the observation at this point. Another way of looking at the buddy check is that it is a procedure to determine whether the innovations are likely or unlikely with respect to the specified innovation error statistics. Once the decision is made, it necessary to intervene in the assimilation process to ensure that the rejected observation has no effect on the analysis.

Before beginning our discussion of the buddy check, we first discuss a preliminary step-the innovation check.

9.3.1 The Innovation Check

We can divide the observations into three categories, acceptable, marginal, and clearly unacceptable. Clearly unacceptable observations will have been largely eliminated during the quality control process. However, in case any clearly unacceptable observations remain, we first perform a check on the magnitude of the innovations. We do this for the following reasons.

The buddy check is the last defense of the assimilation algorithm against bad observations. Its real purpose is to decide whether marginal observations are acceptable or unacceptable. However, if clearly unacceptable observations are introduced into the buddy check, then the buddy check decisions may be corrupted by these clearly unacceptable observations. Thus, as we protect the assimilation algorithm by performing a buddy check, we protect the buddy check by first removing any remaining clearly unacceptable observations by performing the innovation check. This is done in the following straightforward way, which is essentially cost-free.

If L is the number of observations, define the L×L symmetric positive definite matrix

$$\mathbf{A} = \mathbf{H}\mathbf{P}_b\mathbf{H}^T + \mathbf{R}, \tag{9.6}$$

where $\mathbf{H}$, $\mathbf{P}_b$ and $\mathbf{R}$ are defined as in Section 9.1. Define the innovation vector $\mathbf{d} = \mathbf{y} - H(\mathbf{x}_b)$ of length L, where $\mathbf{y}$ is the observation vector, $\mathbf{x}_b$ is the background vector, and H is the forward operator. Denote $\hat{\mathbf{A}} = \text{diag}(\mathbf{A})$ and define $\hat{\mathbf{d}} = \hat{\mathbf{A}}^{-1/2}\mathbf{d}$ as the normalized innovation. Thus, the elements of $\hat{\mathbf{d}}$ will consist of the elements of $\mathbf{d}$, each individually normalized by the square root of the main diagonal elements of the matrix $\mathbf{H}\mathbf{P}_b\mathbf{H}^T + \mathbf{R}$. (In practice, we use the scaled forms of Eq. (3.17), rather than the unscaled form (3.5), but this is a straightforward modification, and we do not dwell on it.)

The elements of the normalized innovation $\hat{\mathbf{d}}$ (over many realizations) should be distributed in a normal distribution with a standard deviation equal to 1 if the background and observation error covariances $\mathbf{P}_b$ and $\mathbf{R}$ have been properly specified. We assume this to be the case. Then, for large L, we would expect the elements $\hat{\mathbf{d}}_\ell$, $1 \le \ell \le$ L of the vector $\hat{\mathbf{d}}$ to be distributed more or less normally, with 31% of the values $|\hat{\mathbf{d}}_\ell|$ being greater than 1, 4.5% being greater than 2, 0.26% being greater than 3, etc. Thus, if any element $|\hat{\mathbf{d}}_\ell|$ is larger than some given tolerance, 4 say, we would conclude that it was so unlikely that we could safely discard that observation. Of course, we know that $\mathbf{P}_b$ and $\mathbf{R}$ are never perfectly specified, so that it would be better to work with a higher tolerance than a lower one.

This procedure is used to remove any clearly unacceptable observations, and we are now ready to make decisions on the acceptability of the marginal observations using the buddy check.

9.3.2 The Metric

Most optimal interpolation schemes (including the NRL MVOI algorithm) used a buddy check procedure developed by Lorenc (1981). This procedure detects suspect observations and then performs mini-analyses with the suspect observation missing, and differences the suspect observation with the analysis (at the observation location) using all other observations, except that observation. If this distance is too large, then the suspect observation is assumed to be inconsistent with other observations and is rejected.

The procedure developed here is quite different. In particular, the metric adopted will judge each observation against the entire observation set, not just nearby observations. Observations will not be removed and tested; this is obviously impossible when all observations are tested at the same time. The metric of choice in this case is defined as follows. For, the L×L matrix $\mathbf{A}$ defined in Eq. (9.6) and the innovation vector $\mathbf{d}$ defined in Section 9.3.1, define a new vector of length L,

$$\mathbf{d}^* = \mathbf{A}^{-1/2}\mathbf{d}. \tag{9.7}$$

We see that $\mathbf{d}^*$ is something like the vector $\hat{\mathbf{d}}$ defined in the innovation check, except that we use the whole matrix $\mathbf{A}$ instead of its diagonal elements, and each element of $\mathbf{d}^*$ is related to all the elements of $\mathbf{d}$. Note that in the special case where the background and observation error covariances are completely uncorrelated, then $\mathbf{d}^* = \hat{\mathbf{d}}$. $\mathbf{A}^{-1/2}$ is defined because $\mathbf{A}$ is a symmetric, positive definite real matrix. Define $\mathbf{E}$ as the L×L matrix of eigenvectors of **A**, and **D** as the LxL diagonal matrix of corresponding eigenvalues. Since **A** is symmetric positive definite, we have $\mathbf{EE}^T = \mathbf{I}$, and all elements of **D** are positive. Then, we can write

$$\mathbf{A} = \mathbf{EDE}^T,\ \mathbf{A}^{1/2} = \mathbf{ED}^{1/2}\mathbf{E}^T \text{ and } \mathbf{A}^{-1/2} = \mathbf{ED}^{-1/2}\mathbf{E}^T. \tag{9.8}$$

We note that $[\mathbf{d}^*]^T\mathbf{d}^* = J_{\min}$ from Eq. (9.2) and should therefore equal L.

At this point, let us leave aside questions about the practicality of calculating **d***. Let us instead, concentrate on the question of whether or not **d*** can be used for making buddy check decisions. Suppose that we have calculated (the absolute value) of an element of the normalized innovation (see Section 9.3.1), that is, $|\hat{d}_\ell|$, and found it to be equal to 3. This observation is marginal when examined in isolation. Now suppose that we calculate the same element of the vector **d***, that is, $|d_\ell^*|$. This calculation involves all of the observations. Then, we postulate that if $|d_\ell^*| \le |\hat{d}_\ell|$, then the buddy check indicates that the marginal observation is more likely than is indicated by the innovation check. That is, the marginal observation is supported by surrounding observations; and we would be more likely to accept this observation than we might have been if we had only done the innovation check. Conversely, if $|d_\ell^*| \ge |\hat{d}_\ell|$; then we would be less likely to accept the observation. At this point, we have not yet shown whether the vector **d*** can be used successfully in this fashion. To do so, we consider a very simple example.

This example is a simple three-observation problem, similar to the two-observation problem considered by Daley (1991, p.128). The problem is in two dimensions, and the three observations are placed at the corners of an equilateral triangle. We consider a scaled form of the equations and consider the formation of the **A** matrix for this problem. We assume that the observation errors are uncorrelated, and the background error correlation between each of the three points is given by ρ. We denote the observation error variances divided by the background error variance as e_0^2 for each observation. Then, the **A** matrix for this problem is

$$\mathrm{A} = \begin{vmatrix} 1+\varepsilon_o^2 & \rho & \rho \\ \rho & 1+\varepsilon_o^2 & \rho \\ \rho & \rho & 1+\varepsilon_o^2 \end{vmatrix}, \tag{9.9}$$

and $\hat{A} = \lim_{\rho\to 0} A$. Then, define a vector of length 3, of innovations **d**, and the corresponding normalized vectors $\hat{\mathbf{d}}$ and **d***. In the limit as ρ approaches zero, $\mathbf{d}^* = \hat{\mathbf{d}}$. There are three elements of the $\hat{\mathbf{d}}$ vector, namely, $\hat{d}_1, \hat{d}_2, \hat{d}_3$ and three elements of the **d*** vector d_1^*, d_2^*, d_3^*. We consider three cases; in each of the three cases, we vary ρ and e_0^2.

In the first case, we set $\hat{d}_1 = \hat{d}_2 = \hat{d}_3 = 3.0$. Thus we consider all three observations to be marginal. Then, if the correlation ρ, were close to 1, i.e., highly correlated, then the innovations, although large, are self-consistent and a good buddy check should accept them. Let us see what actually happens in this case. Table 9.1 plots the values of d_1^* for this case for ρ = 0.8 and –0.4 and $e_0^2 = 2.0$ and 0.1. We note that when ρ = 0, then $d_1^* = \hat{d}_1 = 3.0$.

Consider first the situation when $e_0^2 = 0.1$ (observations specified to be accurate). Then, when ρ = 0.8 (highly correlated background error), $d_1^* = 1.9$, which is less than $\hat{d}_1 = 3.0$ and therefore more likely, as we would hope. Note that in this situation, there is a three-way symmetry between the three observations, so that $d_1^* = d_2^* = d_3^*$, Thus, in this case, the correlations are large; the normalized innovations are similar; and the buddy check indicates that these observations, although marginal, are much more likely than does the innovation check. Clearly, the buddy check is doing the right thing in this situation.

Now, consider the situation when $\varepsilon_0^2 = 0.1$ and ρ = –0.4. Then, $d_1^* = d_2^* = d_3^* = 5.8$, which is much more unlikely than indicated by the innovation check. This also is what we would expect. For example, if two wind innovations

Table 9.1 — Case 1

$\hat{d}_1 = \hat{d}_2 = \hat{d}_3 = 3.0$			
		$\rho = -0.4$	$\rho = 0.8$
$\lvert d_1^* \rvert$	$\varepsilon_o^2 = 2.0$	3.5	2.4
	$\varepsilon_o^2 = 0.1$	5.8	1.9

were large and in the same direction but the correlation was negative (implying the two wind innovation should be in opposite directions), then clearly one or both of the winds would fail the buddy check. This is what happens in this situation.

Finally, setting $\varepsilon_o^2 = 2.0$ (observations specified to be inaccurate) gives the same direction of change as the accurate observation case, but shows much less sensitivity to the correlation ρ, which is what would be expected for inaccurate observations.

In the next case, we set $\hat{d}_1 = 3.0, \hat{d}_2 = \hat{d}_3 = 1.0$. That is, observations 2 and 3 are acceptable and observation 1 is marginal, as indicated by the innovation check. If the background error correlation ρ were large and positive, then we would suspect that observation 3 is inconsistent with observations 2 and 3 and that $\left|d_1^*\right|$would be larger than $\left|\hat{d}_1\right|$. The results are plotted in Table 9.2.

Table 9.2 — Case 2

$\hat{d}_1 = 3.0, \hat{d}_2 = \hat{d}_3 = 1.0$			
		$\rho = -0.4$	$\rho = 0.8$
$\lvert d_1^* \rvert$	$\varepsilon_o^2 = 2.0$	3.2	2.9
	$\varepsilon_o^2 = 0.1$	4.3	3.6

In this case, note that there is no three-way symmetry, so Table 9.2 is valid only for observation 1. From Table 9.2, it is clear that $d_1^* > \hat{d}_1$ for accurate observations and highly correlated background error, as suspected above, and therefore the buddy check would more likely reject observation 1 under these conditions. Moreover, from Table 9.2, the rejection of observation 1 by the buddy check is also more likely when ρ = –0.4, which is also not surprising. As in case 1, the effects are weaker when the specified observation error is larger.

In case 3, we specify $\hat{d}_1 = 3.0, \hat{d}_2 = \hat{d}_3 = -1.0$, that is, observations 2 and 3 are self-consistent, but completely inconsistent with observation 1. We might expect that observation 1 would be considered less likely for ρ = 0.8 and more likely for ρ= –0.4. This turns out to be the case, as can be seen from Table 9.3.

These three cases suggest that the metric (9.7) does indeed increase/decrease with respect to the innovation metric for a suspect observation when that observation is inconsistent/consistent with other observations, as determined from the specified error statistics.

Now before proceeding to test the metric (9.7) under more realistic conditions, we first develop a straightforward approximation to this metric. Consider the innovation vector **d** and the vector $\mathbf{z} = \mathbf{A}^{-1}\mathbf{d}$. We note, from

Table 9.3 — Case 3

$\hat{d}_1 = 3.0, \hat{d}_2 = \hat{d}_3 = 1.0$			
		$\rho = -0.4$	$\rho = 0.8$
$\lvert d_1{}^* \rvert$	$\varepsilon_o{}^2 = 2.0$	2.9	3.4
	$\varepsilon_o{}^2 = 0.1$	2.9	5.3

Eq. (9.1), that $J_{min} = \mathbf{d}^T\mathbf{z}$. Note that each term of the scalar product $[\mathbf{d}^*]^T\mathbf{d}^*$ is positive because of its quadratic nature. However, while $\mathbf{d}^T\mathbf{z}$ is positive, each individual term of the scalar product is not necessarily positive.

$$\text{We approximate } \left|d_\ell^*\right| \text{ by } \left|d_\ell^a\right| = \sqrt{|z_\ell|\,|d_\ell|}\,. \tag{9.10}$$

We note that the approximation (9.10) is exact under two conditions:

(1) when the background error is spatially uncorrelated, and

(2) when the innovation vector **z** is proportional to a single eigenvector of **A**.

The approximate metric (9.10) was tested using the simple 3-observation test of Eq. (9.9). For Case 1 (Table 9.1), the approximation is perfect, because the innovation vector is proportional to one of the eigenvectors of **A**. Table 9.4 shows Case 2, except here using the approximation (9.10). The results can be compared directly with Table 9.2.

Table 9.4 — Case 2 (approximate metric)

$\hat{d}_1 = 3.0, \hat{d}_2 = \hat{d}_3 = 1.0$			
		$\rho = -0.4$	$\rho = 0.8$
$\lvert d_1{}^* \rvert$	$\varepsilon_o{}^2 = 2.0$	3.2	2.9
	$\varepsilon_o{}^2 = 0.1$	4.6	4.1

Clearly, the approximation (9.10) is not perfect and becomes less accurate as the specified observation error decreases. However, comparing Tables 9.2 and 9.4, one would probably make the same buddy check decisions using either the exact (9.7) or the approximate (9.10) metric. Remember, approximation (9.10) would only be used in the buddy check, not in the analysis itself.

A number of experiments (not shown) were performed with AMDAR profiles to see if this buddy check procedure could detect innovations that were inconsistent with the specified background and observation statistics. The results of these experiments were consistent with Tables 9.1 - 9.4. That is, when individual innovations were inconsistent with nearby innovations, the buddy check algorithm was able to detect such inconsistencies.

We now proceed to a more realistic situation. At the moment, we are not aware of any technique for easily calculating the exact metric (9.7) for matrices as large as will be encountered in global data assimilation. This leaves essentially two choices.

1. Apply the exact metric (9.7) to the block pre-conditioner matrix (3.13). This is a block diagonal matrix, and it is feasible (albeit more expensive than one would like) to perform the eigenvector decomposition of (9.8) on each diagonal block in turn. However, only the observations in the same (nonoverlapping) block would participate in the buddy check. This might work well for observations in the centers of the observation prism, but would not work so well for observations on the edge.

2. Apply the approximate metric (9.10) after a few iterations of the descent algorithm (3.13). Although the metric is approximate, all the observations take part in the buddy check for any given observation.

These two strategies were tested in the following way. We considered a set of observations over eastern North America and the western Atlantic. This consisted of 5509 radiosonde and SSM/I wind speeds and total precipitable water. There were 50 observation prisms. First, we calculated the exact (9.7) and approximate (9.10) metrics for each of the diagonal blocks, corresponding to the 50 observation prisms.

We calculated $|\hat{d}_\ell|$, $|d_\ell^*|$, and $|d_\ell^a|$ for each observation and then two measures,

$$\mathrm{M}_1 = L^{-1}\sum_\ell [|d_\ell^*| - |\hat{d}_\ell|]^2 \text{ and } \mathrm{M}_2 = \sqrt{L^{-1}\sum_\ell [|d_\ell^*| - |d_\ell^a|]^2}\,. \qquad (9.11)$$

In the calculation of M_1 and M_2, only observations in which either $|\hat{d}_\ell|$ or $|d_\ell^*|$ exceed 3 were used, that is, we were only interested in marginal observations. M_1 is a measure of the distance between the innovation check and the buddy check using the exact metric (9.7). It should be a number of O(1). M_2 is a measure of the distance between the buddy check using the exact metric (9.7) and the approximate metric (9.10). If the approximate metric (9.10) is to be of any value, then M_2 should be smaller than M_1. The results of the above experiment over eastern North America were

$\mathrm{M}_1 = 1.334$, $\mathrm{M}_2 = 0.350$.

This result suggests that the exact metric has the potential to change the likelihood of any marginal observation by more than one standard deviation (in either direction). Second, the approximate metric (9.10) gives a result that is surprisingly close to the exact metric (9.7).

We then tested the approximate metric (9.10) given by the preconditioner against the same metric calculated at different iterations of the descent algorithm. (Calculating the exact metric (9.7) under the same experimental conditions was out of the question). The purpose of this test was to determine how the inclusion of all the observations in the buddy check (which becomes increasingly the case as the descent algorithm proceeds) compares with including only the observations in the same observation prism (as when using the preconditioner only). Consequently, we first calculated $|d_\ell^a|$ for each observation during the preconditioner as before, and calculated the same approximate metric $|d_\ell^a(j)|$, for each iteration number j. We then constructed a third measure,

$\mathrm{M}_3^j = \sqrt{L^{-1}\sum_\ell [|d_\ell^a(j)| - |d_\ell^a|]^2}$, and evaluated it at each iteration.

$\mathrm{M}_3^3 = 0.389$, $\mathrm{M}_3^6 = 0.644$, and $\mathrm{M}_3^{20} = 0.646$. This result indicates that the approximate metric (and almost certainly the exact metric, although it could not be verified) change substantially as all the observations are included.

The conclusion was that it would be better to apply the approximate metric (9.10) after a few iterations, rather than the exact metric (9.7) during the preconditioner. This is, in fact, how the buddy check metric is calculated.

9.3.3 The Decision

Having decided how to measure the consistency of any observation with respect to other observations, the next step is to decide what to do about that particular observation-reject or retain? Let us review the information we have available to help us make our decision for the *l*th observation,

(1) $|\hat{d}_\ell|$ the normalized innovation magnitude,

(2) $|d_\ell^a|$ the buddy check magnitude using the approximate metric,

(3) from a normal distribution, the expected number of observations that would exceed a given tolerance (1,2,3,4,...),

(4) the actual numbers of observations that exceed that tolerance for both the innovation and buddy checks,

(5) previous quality control flags raised in the off-line quality control.

At this time, the decision process is very simple. We simply reject any observations, whose buddy check magnitude $|d_\ell^a|$ exceeds a certain tolerance (4, say).

Now the metric and decisions in the buddy check are predicated on the fact that the observation and background error statistics are good. What if they are not? Suppose the J_{min} /L value (Eq. (9.1)) is greater than 1. Then, the calculated values of $|d_\ell^*|$ would be larger than they should be, and we might spuriously reject good observations.

The buddy check is more susceptible to bad error statistics than the analysis algorithm itself. Suppose the J_{min} /L value came out to be 2. Then, it would be a simple matter to multiply all the background and observation error variances by 2 to make the J_{min} /L values come out equal to 1. From Eqs. (3.4) and (3.5), this change in background and observation error statistics would have no effect on the analysis (although from Eq. (2.10) it would increase the expected analysis error variance by a factor of 2). However, this change would completely change the buddy check decisions, with the tolerance (in real space) being effectively multiplied by $\sqrt{2}$.

Thus, it is important to tune the error statistics first and make sure the J_{min} /L for each area and each observation system is close to 1. (Note however, that the innovation and buddy check are not independent of this process, because it will have to be decided which observations take place in the J_{min} /L calculation.)

9.3.4 Modifying the Decision Process for Extreme Events

Another problem with the buddy check decision (and also the innovation check decision) is that the error statistics are produced by averaging time series of innovations and reflect average, rather than extreme, conditions in each part of the domain. It could happen that where large, rapid changes are taking place in the atmosphere (such as a frontal passage), that the background error is considerably larger than normal. In this case, the possibility exists that the buddy or innovation checks would spuriously reject good (and very important) observations.

We have adopted two procedures to deal with this problem.

(1) We first calculate the normalized buddy check or innovation check magnitude for every observation in an observation prism. We then calculate the average value for the prism. We use the mode rather than the mean because it is less sensitive to outliers. If this value is much larger than 1, then we can assume that for this prism, there is a serious discrepancy between the observations and backgrounds. This is a critical situation, and we do not wish to spuriously reject a number of observations in this prism. Consequently, we increase the tolerances for this prism.

(2) The second method is based on Onogi (1998). In Onogi's procedure, one uses diagnostic information (time tendencies or spatial gradients) of the background field to signal that there is an extreme event in the background field and therefore the normal background error statistics are probably invalid. Then, one would change the tolerances appropriately. This method has one drawback compared to (1). That is, if the background field is extremely inaccurate, then the decisions to increase the tolerances could be completely wrong.

We use a combination of (1) and (2).

9.3.5 The Action

In principle, the action is very straightforward. If we have the elements of the main diagonal blocks of $\mathbf{HP}_b\mathbf{H}^T + \mathbf{R}$ (i.e. the preconditioner), we simply have to add a large constant - c - to each of the main diagonal elements of this block that have been rejected by the buddy check or innovation check. This then modifies the matrix $\mathbf{HP}_b\mathbf{H}^T + \mathbf{R}$ in such a way as to effectively prevent the rejected observation from affecting the analysis.

However, a problem arises with the buddy check (because it is performed after the descent has begun) if the block diagonal matrices of the preconditioner have already been Choleski decomposed, so that only the lower triangular matrices are available, and the original block diagonal matrices have been lost (for storage reasons). Following Section 3.4, consider the nth diagonal block, containing K observations. Then, the matrix $\mathbf{A}_n = [\mathbf{HP}_b\mathbf{H}^T + \mathbf{R}]_n = \mathbf{LL}^T$, where the subscript n on the Choleski matrices is to be understood. As noted, K observations are in this prism, and we suppose the jth observation has been rejected by the buddy check. Then define the K×K matrix $\mathbf{D}_j$, which is zero everywhere, except the jth main diagonal element $d_{jj} = c$. (If more than one rejected observation is in the prism, then other main diagonal elements will be augmented by the constant c in this way).

The most straightforward procedure

The nth diagonal block matrix, after buddy check rejection of the jth element, will be $\mathbf{LL}^T + \mathbf{D}_j$. If we only have available $\mathbf{L}$ and we require a new lower triangular matrix $\mathbf{L}^*$ that reflects the rejection of the jth observation, then we must first calculate $\mathbf{LL}^T$, which is an $O(K^3)$ operation and add the element $d_{jj} = c$, which is trivial. We must then take this new K×K symmetric matrix and Choleski decompose it again to produce the new lower triangular matrix $\mathbf{L}^*$. This last operation is also $O(K^3)$. This whole procedure is rather more expensive than one would like, although certainly not a "show stopper."

A much more efficient procedure

There is however, a much more efficient, if slightly convoluted way of achieving exactly the same result. We first consider the case with a single rejected observation, the jth. Define a vector $\mathbf{d}_j$ of length K, with elements d_k, $1 \leq k \leq K$, in which all elements are equal to zero except $d_j = c$. Also define a vector $\mathbf{a}_j$, of length K, with elements a_k, $1 \leq k \leq K$, which is obtained from $\mathbf{d}_j$ by

$$\mathbf{a}_j = [\mathbf{L}^T]^{-1}\,\mathbf{L}^{-1}\,\mathbf{d}_j. \qquad (9.12)$$

This operation is O(K^2). Then, define the K×K matrix

$$S_j = \begin{bmatrix} 1 & a_1 & & 0 \\ & 1+a_j & & \\ & a_{j+1} & 1 & \\ 0 & a_K & & 1 \end{bmatrix}. \qquad (9.13)$$

Then, it can easily be shown that

$$\mathbf{LL}^T + \mathbf{D}_j = \mathbf{LL}^T\mathbf{S}_j. \qquad (9.14)$$

Note that $\mathbf{L}^T\mathbf{S}_j$ or $\mathbf{S}_j\mathbf{L}$ are not triangular matrices, but that if **x** is some arbitrary vector of length K, the following operations are equivalent:

$$[\mathbf{LL}^T + \mathbf{D}_j]\mathbf{x} = \mathbf{LL}^T\,\mathbf{S}_j\,\mathbf{x}, \text{ and } [\mathbf{LL}^T + \mathbf{D}_j]^{-1}\,\mathbf{x} = \mathbf{S}_j^{-1}[\mathbf{L}^T]^{-1}\mathbf{L}^{-1}\mathbf{x}. \qquad 9.15)$$

Note that the operations $\mathbf{S}_j\,\mathbf{x}$ and $\mathbf{S}_j^{-1}\,\mathbf{x}$ are each trivial and O(K).

If more than one rejected observation is in an observation prism, the procedure is only slightly more complicated. We just illustrate the case with two rejected observations j_1 and j_2, and cases with higher number of rejections can be inferred. Thus, we define, the vectors $\mathbf{d}_{j_1}$ and $\mathbf{d}_{j_2}$ as in (9.12) corresponding to these two rejected observations. Then define the corresponding vectors α_{j_1} and α_{j_2} by

$$\alpha_{j_1} = [\mathbf{L}^T]^{-1}\mathbf{L}^{-1}\,\mathbf{d}_{j_1} \text{ and } \alpha_{j_2} = \mathbf{S}_{j_1}^{-1}\,[\mathbf{L}^T]^{-1}\mathbf{L}^{-1}\,\mathbf{d}_{j_2}, \qquad (9.16)$$

where $\mathbf{S}_{j_1}$ is obtained from α_{j_1} following (9.12), and we can similarly obtain $\mathbf{S}_{j_2}$ from α_{j_2}.

Then, the multiplicative and inversion operations shown in (9.15) for a single rejected observation become the two operations,

$$\mathbf{LL}^T\mathbf{S}_{j_1}\mathbf{S}_{j_2}\mathbf{x} \text{ and } \mathbf{S}_{j_2}^{-1}\mathbf{S}_{j_1}^{-1}\,[\mathbf{L}^T]^{-1}\mathbf{L}^{-1}\,\mathbf{x}. \qquad (9.17)$$

This procedure can be extended to any number of rejected observations.

Suppose J rejected observations are in the prism. Then operations such as those in (9.12) and (9.16) are one-time operations, done only at the time of rejection. These operations are O(JK^2). The remaining operations, such as those of (9.15) or (9.17), are performed during every subsequent descent iteration. These operations are O(JK). Now we have assumed all along that there will not be many rejections at the buddy check stage and therefore J << K. This means that these operations are all relatively negligible compared to the O(K^3) operations of the straightforward method above. A modest amount of extra storage is required for the α vectors.

Proceeding with the descent

One other issue still must be faced. We have modified the matrix $\mathbf{A} = \mathbf{HP}_h\mathbf{H}^T + \mathbf{R}$ using either the (explicit) straightforward method or the faster (implicit) method of equations (9.12)-(9.17). We refer to this buddy check-modified matrix as **A** + **D**; **D** is the matrix that is zero everywhere except for any observation j, which has been tagged by the buddy check, where the element d_{jj} = c, as above. Referring to the preconditioned conjugate gradient equation (3.13), this (explicitly or implicitly) modified matrix is now no longer consistent with the vectors $\mathbf{z}_k$, $\mathbf{p}_k$, $\mathbf{r}_k$, etc., which have been evolving during the descent. One has choices here too. One of them is to restart the descent (k = 0) using the new matrix **A** + **D**. Then, the total cost of the descent would be the number of steps used before application of the buddy check plus whatever number of iterations it takes to reach the specified convergence after restarting the descent algorithm. Now, if we wish to involve as many observations as possible in each buddy check decision, we should not do the buddy check too early in the decision process. (We probably should not wait until the descent has converged, either.)

There is a second possibility, however. That is, after performing the buddy check and (explicitly or implicitly) modifying the **A** matrix, we can continue the descent after suitably modifying the vectors $\mathbf{z}_k$, $\mathbf{p}_k$, $\mathbf{r}_k$, etc. of Eq. (3.13). The simplest way to do this is as follows. Do not change $\mathbf{z}_k$, but calculate a new residual vector $\underline{\mathbf{r}_k}$ as

$$\underline{\mathbf{r}_k} = \mathbf{d} - [\mathbf{A} + \mathbf{D}]\mathbf{z}_k = \mathbf{r}_k - \mathbf{D}\mathbf{z}_k . \tag{9.18}$$

Then, set $\beta_k = 0$ and continue with the descent.

9.4 The NAVDAS Adjoint Operator

One of the most effective techniques in determining where to take observation are the adjoint and singular vector targeting techniques developed at NRL by Ron Gelaro, Rolf Langland, Greg Rohaly, and Caroline Reynolds. In such applications, the adjoint of the model is used together with some measure J of the forecast error to determine the sensitivity of the forecast to the analysis. This sensitivity is expressed in the form of a gradient $\partial \mathbf{J} / \partial \mathbf{x}_a$, where $\mathbf{x}_a$ is the analysis vector. As shown by Baker and Daley (2000), this procedure can be extended to calculate the sensitivity of the forecast to the observations ($\partial \mathbf{J} / \partial \mathrm{y}$, where **y** is the observation vector) and the background field ($\partial \mathbf{J} / \partial \mathbf{x}_b$, where $\mathbf{x}_b$ is the background). From Baker and Daley (2000), we have

$$\partial \mathbf{J} / \partial \mathrm{y} = \mathbf{K}^T \, \partial \mathbf{J} / \partial \mathbf{x}_a, \text{ where } \mathbf{K} = \mathbf{P}_b\mathbf{H}^T[\mathbf{H}\mathbf{P}_b\mathbf{H}^T + \mathbf{R}]^{-1}, \tag{9.19}$$

and $\mathbf{P}_b$, **R**, and **H** are as defined in Section 3.2. **K** is the Kalman gain or weight matrix, and its transpose or adjoint $\mathbf{K}^T$ is given by

$$\mathbf{K}^T = [\mathbf{H}\mathbf{P}_b\mathbf{H}^T + \mathbf{R}]^{-1}\mathbf{H}\mathbf{P}_b. \tag{9.20}$$

In other words, given an analysis sensitivity vector, we first operate with the transpose or adjoint of the post-multiplier (Eq. (3.6)) and then apply the solver (Eq. (3.5)). The solver is symmetric or self-adjoint and therefore operates the same way in the forward and adjoint directions. The only difference between the forward and adjoint codes is in the post-multiplier. A small complication is the vertical eigenvector decomposition used in the post-multiplier. Consider a single vertical column of the analysis sensitivity vector **r** (denoted 1) and a single observation profile **q** (denoted 2). Then, following Eq. (4.16), we can write the adjoint post-multiplication operation as follows,

$$\mathbf{q} = \mathbf{H}\mathbf{P}_b\mathbf{r} = \mathbf{E}_2\mathbf{D}_{12}\mathbf{E}_1{}^T\,\mathbf{r}, \tag{9.21}$$

where $\mathbf{E}_1$, $\mathbf{E}_2$ are eigenvector matrices, and $\mathbf{D}_{12}$ is diagonal and a function of the horizontal (background error) correlations between the two vectors and the vertical mode number. We note that (9.21) is the transpose of the forward post-multiplier operation (viz, $\mathbf{E}_1\mathbf{D}_{12}\mathbf{E}_2{}^T$). $\mathbf{D}_{12}$ is symmetric and therefore the same in both forward and adjoint directions. The other operators ($\mathbf{E}_1$, $\mathbf{E}_2$, and their transposes) already exist in the forward code, so it is just a question of applying them in a different order in the adjoint code.

9. Internal Diagnostics

It is relatively simple to code the adjoint NAVDAS code from the forward NAVDAS code. The adjoint code would have to be updated as new forward operators or other modifications are added to the forward code. The adjoint code has the same properties with respect to multiprocessing as the forward code.

10. Estimating the Analysis Error Covariance

A valuable output from a data assimilation system is an estimate of the analysis error covariance matrix; it can be used for many purposes. In some cases, it may be sufficient to obtain the diagonal component of this matrix, that is, the analysis error variance. In principle, both are straightforward to compute, using Eq. (2.10), which can also be re-written in a form more suitable for observation space systems as

$$\mathbf{P}_a = \mathbf{P}_b - \mathbf{P}_b\mathbf{H}^T[\mathbf{H}\mathbf{P}_b\mathbf{H}^T + \mathbf{R}]^{-1}\mathbf{H}\mathbf{P}_b, \tag{10.1}$$

where $\mathbf{P}_a,\mathbf{P}_b$ are the I×I analysis and background error covariance matrices, **R** is the K×K observation error matrix, and **H** is the K×I linearized forward operator. Here, K is the number of observations, and I is the number of analysis grid points (as in Section 2.)

Equations (2.10) or (10.1) estimate the analysis error covariance exactly if the background and observation error covariances are known perfectly. Otherwise, (2.10) and (10.1) will, in general, underestimate the true analysis error variance. This is, unfortunately, the usual situation.

Even though Eq. (10.1) produces only an estimate of the true analysis error covariance, it is, in general, too expensive to use. In fact, to calculate the second right-hand side term of (2.10) takes $2K^2I$ operations, which is enormously more costly than computing the analysis estimate itself. All operations in (2.10) are matrix/matrix operations, rather than the matrix/vector operations of the analysis estimate itself.

Thus, we seek some way to approximate (10.1), in particular the second right-hand side term of the equation, which is both accurate and computationally feasible. Because Eq.10) itself produces an underestimate of the analysis error variance, we select conservative methods, that is, methods whose analysis error variances exceed the values that would be produced by (10.1).

We consider two such methods, a local estimation method based on the preconditioners developed in Section 3 and a global method based on finding the largest eigenvectors of the background error covariance. We then apply both approximations to a simple one-dimensional univariate problem, and finally apply one of the methods to the full NAVDAS algorithm.

10.1 A Local Estimate of the Analysis Error Covariance

This method of estimating the analysis error covariance is based on the block diagonal preconditioner described in Section 3.2.5 and the Choleski decomposition of the diagonal blocks described in Section 3.4 We assume there are N diagonal blocks, and that there are K_n innovations for the nth diagonal block. We denote $[\mathbf{H}\mathbf{P}_b\mathbf{H}^T + \mathbf{R}]_n$ as the $K_n \times K_n$ symmetric diagonal block matrix used in the pre-conditioner. We can use Choleski decomposition to write this matrix as

$$[\mathbf{H}\mathbf{P}_b\mathbf{H}^T + \mathbf{R}]_n = \mathbf{L}_n\mathbf{L}_n^T, \tag{10.2}$$

where $\mathbf{L}_n$ is a $K_n \times K_n$ lower triangular matrix.

Now obtain the centroid (θ_n,λ_n) of the nth observation prism (containing K_n observations and associated with the nth diagonal block). This is done using the methods of Appendix A. We examine the latitudes and longitudes of each of the I analysis grid points, and for each of these grid points find the closest observation prism centroid (as measured by the great-circle distance). Then for the nth observation prism, there will be I_n such grid points and $\sum_{n=1}^{N} I_n = I$. Now define the $I_n \times K_n$ matrix $[\mathbf{P_b H^T}]_n$, whose elements are the elements of the $\mathbf{P_b H^T}$ matrix that involve interactions between the K_n observations and the I_n grid points. Now, define

$$\mathbf{G}_n = [\mathbf{P}_b\mathbf{H}^T]_n \mathbf{L}_n^{-1}, \tag{10.3}$$

which is an $I_n \times K_n$ matrix.

Approximate the matrix, $\mathbf{P_b H^T [H P_b H^T + R]^{-1} H P_b}$ by

$$\mathbf{G}_n\mathbf{G}_n^T = [\mathbf{P}_b\mathbf{H}^T]_n \mathbf{L}_n^{-1} [\mathbf{L}_n^T]^{-1} [\mathbf{HP}_b]_n \text{, for } 1 \leq n \leq N. \tag{10.4}$$

We have used the fact that $[\mathbf{P}_b\mathbf{H}^T]_n^T = [\mathbf{HP}_b]_n$. $\mathbf{G}_n\mathbf{G}_n^T$ is an $I_n \times I_n$ symmetric matrix and there are N such blocks. Thus, we have produced a block diagonal approximation to the second right-hand side term of Eq. (2.1). The last step is to produce the corresponding N $I_n \times I_n$ diagonal blocks of $\mathbf{P}_b$. Following (10.1), subtraction produces an estimate of the N $I_n \times I_n$ diagonal blocks of $\mathbf{P}_a$. Approximation (10.4) is inherently conservative, because for any analysis grid point, only a subset of the observations are used in determining the analysis error variance.

It is desirable to calculate (10.4) using, as much as possible, the pre-existing code of the analysis algorithm itself. To do this, we simply define $\mathbf{G}_n$ as a sequence of operators. Introduce $1 \leq k \leq K_n$ vectors $\mathbf{e}_k$ of length K_n, each of which consists of all zero elements except the kth element, which is set equal to 1. Then calculate

$$\mathbf{L}_n\mathbf{e}_k = \mathbf{f}_k,\ \ \mathbf{g}_k = [\mathbf{P}_b\mathbf{H}^T]_n \mathbf{f}_k,\ 1 \leq k \leq K_n, \tag{10.5}$$

where $\mathbf{f}_k$ is an intermediate vector of length K_n and $\mathbf{g}_k$ is a vector of length I_n. Then, it is easy to see that

$$\mathbf{G}_n\mathbf{G}_n^T = \sum_{k=1}^{K_n} \mathbf{g}_k\mathbf{g}_k^T, \tag{10.6}$$

which can easily be calculated using a running sum. The formulation (10.5-6) essentially turns the matrix/matrix operation of Eq. (10.4) into a sequence of matrix/vector operations, for which all the necessary operators already exist in the 3dvar code itself.

Warning: it is absolutely critical to ensure that each analysis grid point is associated with one and only one observation prism. If this rule is violated there is the possibility that at a given horizontal location, it may be assumed that the background error is zero in calculating $\mathbf{L}_n$, but not in calculating $[\mathbf{P}_b\mathbf{H}^T]_n$, which is inconsistent. In Section 10.3, we show an example of what can happen if this rule is violated.

A variant on the above scheme is to replace (10.2) with an eigenvector decomposition following Eq. (9.8). That is, write

$$[\mathbf{HP}_b\mathbf{H}^T + \mathbf{R}]_n = \mathbf{E}_n\mathbf{D}_n\mathbf{E}_n^T, \tag{10.7}$$

where $\mathbf{E}_n$ is a $K_n \times K_n$ matrix of eigenvectors and $\mathbf{D}_n$ is a diagonal $K_n \times K_n$ matrix of (positive) eigenvalues. Following Eq. (9.8), we can replace Eq. (10.3) by

$$\mathbf{G}_n = [\mathbf{P}_b\mathbf{H}^T]_n \mathbf{E}_n\mathbf{D}_n^{-1/2}, \tag{10.8}$$

noting that $\mathbf{E}_n\mathbf{E}_n{}^T$ is the identity matrix. Now, it is more costly to obtain the complete eigenstructure in this way. Equation (10.8) is useful only if many of the eigenvalues and eigenvectors of (10.7) make no contribution to $\mathbf{G}_n$. For example, if only the K_n / j gravest eigenvectors of (10.7) were important in calculating (10.8), then the sum in (10.6) would be only over K_n / j, rather than K_n, clearly a computational advantage as j increases. This idea was suggested and tested by Riishojgaard (2000). We found the errors caused by this approximation to be acceptable for $K_n / 2$ but unacceptable for $K_n / 4$. This approximation is conservative, which is attractive.

Finally, let us consider some of the computational implications of the local approximation algorithm. The elements of the matrices $\mathbf{L}_n$ and $[\mathbf{P}_b\mathbf{H}^T]_n$ have to calculated only once for each n; they do not have to be recalculated during each pass through the k loop of Eqs. (10.5)-(10.6). Since I_n is usually much greater than K_n, the bulk of the calculation is done in the operation $\mathbf{g}_k = [\mathbf{P}_b\mathbf{H}^T]_n\, \mathbf{f}_k$ of (10.5). Note also that this algorithm is embarrassingly parallel; there is no communication between processors except at the very end of the algorithm when all the analysis error variances have to be collected. The only impediment to complete scalability would be improperly balanced loads on each processor. For optimality, the processor loads may have to be balanced differently than they are in the NAVDAS algorithm itself. (We found that a self-scheduling algorithm works very well for this problem.) The calculation is accelerated by the use of a reduced number of vertical eigenmodes (see Section 4.8) and also by the use of the Riishojgaard (2000) procedure, discussed above.

We now consider a quite different algorithm for estimating the analysis error covariance.

10.2 A Global Estimate of the Analysis Error Covariance

Fisher and Courtier (1995) discuss various methods for obtaining analysis error covariance estimates based on low rank approximations to (10.1). Thus, if there are I grid points, define the $I_v \times I_v$ symmetric matrix **F** and the $I_v \times I$ matrix **E**. Then, we would approximate (10.1) by

$$\mathbf{P}_a \approx \mathbf{P}_b - \mathbf{E}\mathbf{F}\mathbf{E}^T. \qquad (10.9)$$

Equation (10.9) is said to be a low rank approximation to (10.1) if $I_v << I$. For analysis space algorithms, such low rank approximations can be obtained as a by-product of the descent algorithm. This is because there is a connection between the Hessian matrix and the analysis error covariance (see equation 2.10). Moreover, there is also a connection between each iteration of a conjugate gradient descent and the leading eigenvectors found by a Lancos algorithm. This makes low rank approximations particularly attractive for analysis space methods, because the estimates of the analysis error covariance are essentially free. Even though I could be greater than 10^7, I_v would normally be less than 100. Such an approximation is global (whole domain) rather than local as are the approximations of Section (10.1). Low rank approximations of this type are inherently conservative.

We now introduce a global approximation to (10.1) in the spirit of the estimates of Fisher and Courtier (1995) but from a different perspective. It seems more difficult to introduce global approximations into an observation space system, but the following procedure is possible, in principle at least. In an analysis procedure, the greatest effect of the observations occurs in those modes of the background error covariance that have the largest error and that also project strongly onto the observation network. The idea then, is to find the reduction of error due to the observations in the I_v gravest modes of the background error covariance $\mathbf{P}_b$ and ignore the reduction of error in the remaining modes. To do this, we return to Eq. (2.10), which we rewrite here as

$$\mathbf{P}_a{}^{-1} = \mathbf{P}_b{}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}. \qquad (10.10)$$

Assume that $\mathbf{P}_b$ is expanded in its eigenvalues and eigenvectors,

$$\mathbf{P}_b = \mathbf{E}_b\mathbf{D}_b\mathbf{E}_b{}^T, \qquad (10.11)$$

where $\mathbf{D}_b$ is an I×I diagonal matrix of (positive) eigenvalues and $\mathbf{E}_b$ is an I×I matrix of the corresponding eigenvectors. Also define the I×I symmetric matrix **G** as

$$\mathbf{G}^{-1} = \mathbf{E}_b^{T}\mathbf{H}^{T}\mathbf{R}^{-1}\mathbf{H}\mathbf{E}_b. \tag{10.12}$$

Then, we can re-write (10.10) as

$$\mathbf{P}_a^{-1} = \mathbf{P}_b^{-1} - \mathbf{E}_b\mathbf{G}^{-1}\mathbf{E}_b^{T}. \tag{10.13}$$

By analogy with the pair of equations (10.1) and (10.10), it is easy to see that equation (10.13), can be rewritten as

$$\mathbf{P}_a = \mathbf{P}_b - \mathbf{P}_b\mathbf{E}_b[\mathbf{E}_b^{T}\mathbf{P}_b\mathbf{E}_b + \mathbf{G}]^{-1}\mathbf{E}_b^{T}\mathbf{P}_b = \mathbf{P}_b - \mathbf{E}_b\mathbf{D}_b[\mathbf{D}_b + \mathbf{G}]^{-1}\mathbf{D}_b\mathbf{E}_b^{T}, \tag{10.14}$$

using (10.11) and remembering that $\mathbf{E}_b\mathbf{E}_b^{T}$ is the identity matrix.

Now let us consider only the I_v gravest modes of $\mathbf{P}_b$ and define the I_v×I_v diagonal matrix $\underline{\mathbf{D}_b}$ and the corresponding I×I_v eigenvector matrix $\underline{\mathbf{E}_b}$. Then, define the I_v×I_v matrix $\underline{\mathbf{G}}$ following Eq. (10.12), that is,

$$\underline{\mathbf{G}^{-1}} = \underline{\mathbf{E}_b^{T}}\mathbf{H}^{T}\mathbf{R}^{-1}\mathbf{H}\underline{\mathbf{E}_b}. \tag{10.15}$$

We can then approximate Eq. (10.1) by

$$\mathbf{P}_a \approx \mathbf{P}_b - \underline{\mathbf{E}_b\mathbf{F}_b\mathbf{E}_b^{T}}, \tag{10.16}$$

where $\underline{\mathbf{F}_b} = \underline{\mathbf{D}_b}[\underline{\mathbf{D}_b} + \underline{\mathbf{G}}]^{-1}\underline{\mathbf{D}_b}$ is a symmetric I_v×I_v matrix. Equation (10.16) is now in the same form as (10.10), provided I_v << I is a low rank approximation to (10.1). In any case, if I_v is O(100), then the manipulation, inversion, etc., of matrices $\underline{\mathbf{D}_b}, \underline{\mathbf{G}}, \underline{\mathbf{F}_b}$ is not costly. If this scheme were to be used in practice ,one would first have to obtain the first I_v gravest eigenmodes of $\mathbf{P}_b$, perhaps by a Lancos procedure, and then apply (10.15 and 10.16).

The question surrounding all low rank approximations such as (10.9) is, how many modes do you need to get a good approximation to the analysis error covariance $\mathbf{P}_a$? If I = 10^7, how much of the reduction of the error by the observation system can you characterize with only 100 of the gravest modes? That, of course, depends on the actual rank of $\mathbf{P}_b\mathbf{H}^{T}[\mathbf{H}\mathbf{P}_b\mathbf{H}^{T} + \mathbf{R}]^{-1}\mathbf{H}\mathbf{P}_{b.}$ which may well be considerably less than I, but may still be a good bit greater than I_v. Some feeling for this can be gained by examination of Fig. 5 from Fisher and Courtier (1995). Their results for the ECMWF global system with 24 and 52 vectors indicate that the global approximation for estimating the analysis error covariance does not "see" isolated observations, nor is it "aware" of much of the transient observation system (satellite observations in particular). This is unfortunate because, of course, an analysis error estimate for the fixed network is inherently time invariant and not all that valuable.

We now show some simple experimental results with the two analysis error estimation procedures.

10.3 Experiments with a One-Dimensional Univariate System

Fisher and Courtier (1995) tested a number of their global analysis error approximation algorithms on a simple one-dimensional univariate problem. We also performed experiments with the algorithms of Sections 10.1 and 10.2 on a very similar problem. The problem was to define the analysis error variance for wind observations on a one-dimensional periodic domain. Thus, we defined a periodic domain $-p \le x \le p$, with I = 151 grid points. The background error correlation is of Guassian form appropriate for nondivergent wind/wind correlations and is shown in Fig. 10.1. Scattered irregularly on the grid (in the same sort of distribution as shown in Fig. 1 of Fisher and Courtier, 1995) were K = 91 observations. These observations were intended to mimic wind observations, and the observation error was assumed to be spatially uncorrelated.

The background error variance varied spatially in the same manner as in Fisher and Courtier (1995). The spatial distribution of its square root (in m/s) is indicated as the solid line in Fig. 10.2(a). Considerable spatial variation of the background error variance can be seen. The corresponding spectral distribution of the background error variance is obtained by using the background error variance of Fig. 10.2.(a), the correlation of Fig. 10.1, and pre- and post-multiplication by the Fourier matrices as in Section 3.3.1. The result is shown as the solid line in Fig.10.2(b) as a function of Fourier wavenumber. The maximum background error variance is at a nonzero wavenumber, as would be expected for a wind/wind correlation (see Daley, 1991, Chapter 5). The observation error variance was specified to be constant for every observation and was defined to be the domain average of the background error variance shown by the solid curve of Fig. 10.2(a).

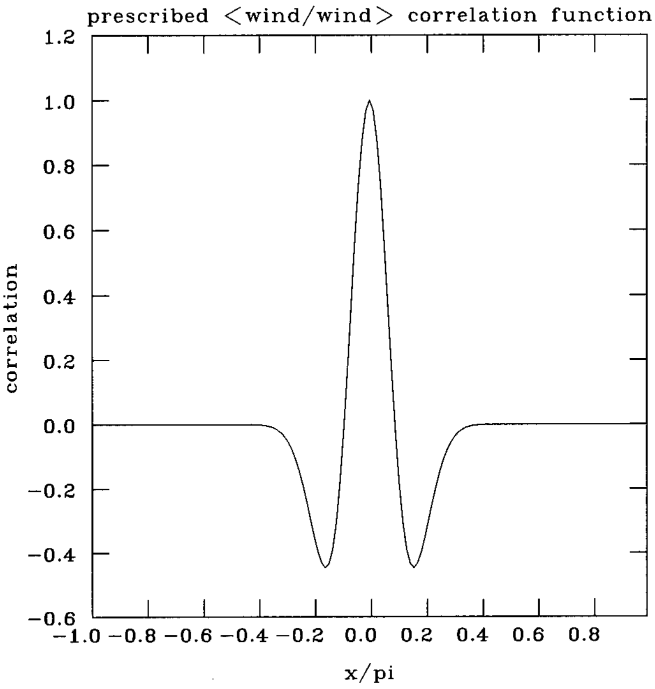


Figure 10.1
One dimensional <wind / wind> correlation

The true analysis error covariance was calculated using Eq. (10.1). The spatial distribution of the square root of the analysis error variance is shown by the dash-dot line of Fig 10.2(a); it has more spatial variation than the background error variance. It is evident from Fig. 10.2(a) that there are several areas where the analysis error variance is only slightly less than the background error variance, indicating very low local observation density. The dash-dot line of Fig. 10.2(b)

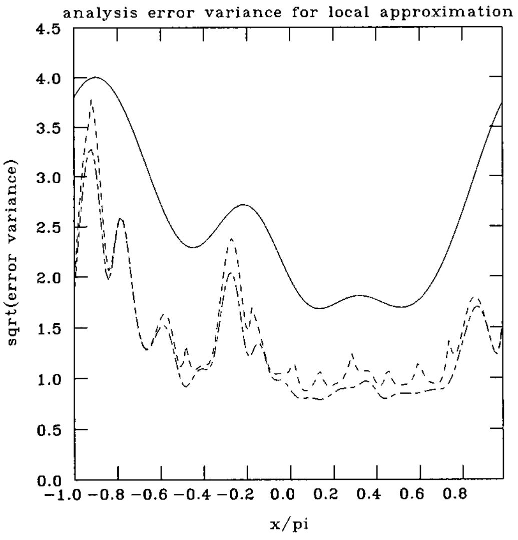


Figure 10.2a
Background and analysis error variance for local approximation

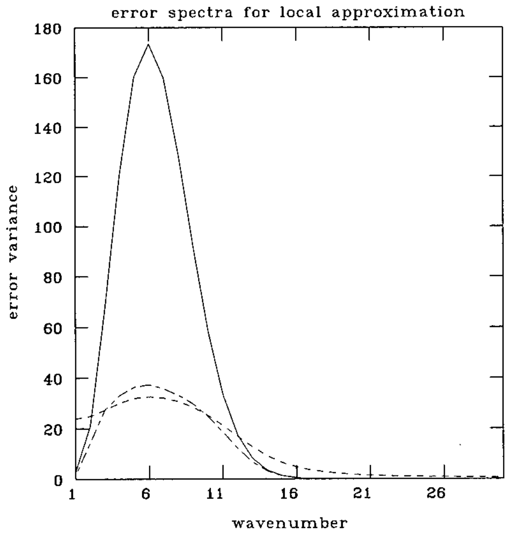


Figure 10.2b
Spectra corresponding to (a)

shows the corresponding spectrum of the analysis error variance. It is always less than the background error variance, and the greatest reduction in error due to the assimilation of the observations is about wavenumber 6, where the background error variance is the greatest. This is consistent with the remarks at the beginning of Section 10.2.

We now examine the local estimate of the analysis error variance obtained by applying Eqs. (10.5)-(10.6). The 91 observations were divided into nine observation boxes, each of which contained 9 or 10 observations. The number of observations was approximately equal in each observation box, although the proportion of the domain covered by each box differed because of the variable observation density. The number of observation boxes was approximately equal to the square root of the number of observations, consistent with the full NAVDAS algorithm itself. The resulting analysis error variance due to the local approximation is shown in Fig. 10.2(a) by the dashed line. The approximation is conservative in that the approximate analysis error variance always exceeds the result from the full equation (10.1). It can also be seen that the approximation ((10.5)-(10.6)) is rougher, which is a manifestation of the observation box boundaries. The corresponding spectrum is indicated by the dashed line of Fig. 10.2(b). The local approximation overestimates the analysis error at very large scales (because it does not account for correlations at large distances) and at small scales (because of discontinuities at observation box boundaries). On the whole, however, the local approximation gives quite encouraging results for this problem.

Figure 10.3 (in the same format as Fig 10.2) shows the results with the global approximation of Section 10.2. In this case, the eigenvectors of the background error covariance were the discrete Fourier modes. Thus, panel (a) shows the spatial distribution of the background error variance (solid curve), true analysis error variance (dash-dot curve), and approximate analysis error variance (dashed curve). The same convention is used in the spectra shown in panel (b). For the global approximation, we assumed that $I_v = 6$, which is much less than the number of grid points, I = 151. Figure 10.3(a) shows that this procedure provides a smooth, large-scale, conservative, but not very good approximation to the true analysis error variance. The spectra of Fig. 10.3b) show that the global method provides a very good approximation for the scales near wavenumber 6, where the background error variance is a maximum. The approximation is very poor at other scales, where the approximate analysis error variance reverts to the background error variance.

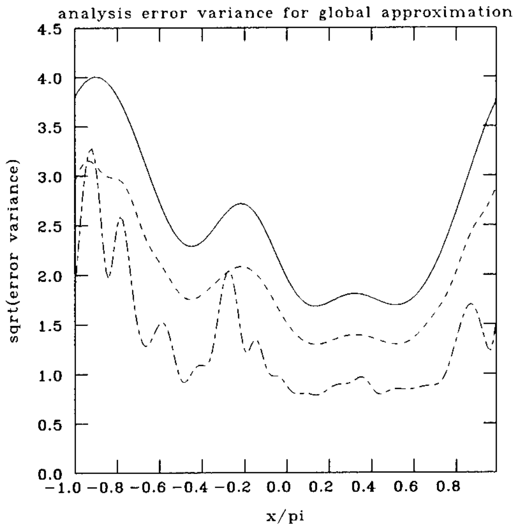


Figure 10.3a
Background and analysis error variance for global approximation

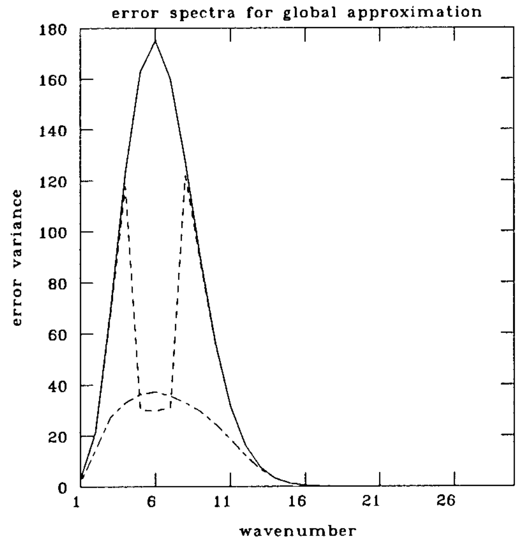


Figure 10.3b
Spectra corresponding to (a)

The results shown in Fig. 10.3 are consistent with the set of algorithms tested by Fisher and Courtier (1995). However, Fisher and Courtier (1995) tested larger values of $I_v = 10$ and $I_v = 35$, which, of course, produced much better results than our use of $I_v = 6$. However, consider the real situation. There one might expect $O(10^7)$ grid points and perhaps $I_v < 200$. Thus, in the one-dimensional experiments of Fisher and Courtier with $I < 300$, $K < 200$, one suspects that the use of $I_v = 35$, is likely to lead to wildly optimistic results.

Experiments were performed with both local and global algorithms in which the background error correlation length and the observation error variance were varied. The problem of assimilating geopotential was also examined. The conclusions from these experiments (all in accord with intuition) are as follows.

1. As the observation accuracy is increased, the local approximation improves and the global approximation gets worse.
2. As the background error horizontal correlation length increases, the global approximation improves and the local approximation gets worse.
3. The local approximation does a better job with winds than geopotentials, and the reverse is true for the global approximation.

In Fig. 10.4 (in the same format as Fig. 10.2(a)), we show the analysis error variance (dashed curve) for a misapplication of the local approximation. Refer to the warning after Eq. (10.6). In this experiment, we used the local approximation (10.2) for the inversion of the matrix $\mathbf{HP_bH^T + R}$ matrix, but used the full $\mathbf{P_bH^T}$ and $\mathbf{HP_b}$ matrices in (10.1) for the right and left post-multiplication. This would seem to be a more accurate approximation than using the local approximation for the post-multiplication, as in equations (10.5-6). However, this approximation is inconsistent, resulting in the catastrophic results of Fig. 10.4. Thus, it is very unwise to ignore the warning of Section 10.1.

Based on the results of this section and the relative difficulties of implementing either of the algorithms of Sections10.1 and 10.2, we decided to implement the local approximation of Section 10.1 in the NAVDAS code.

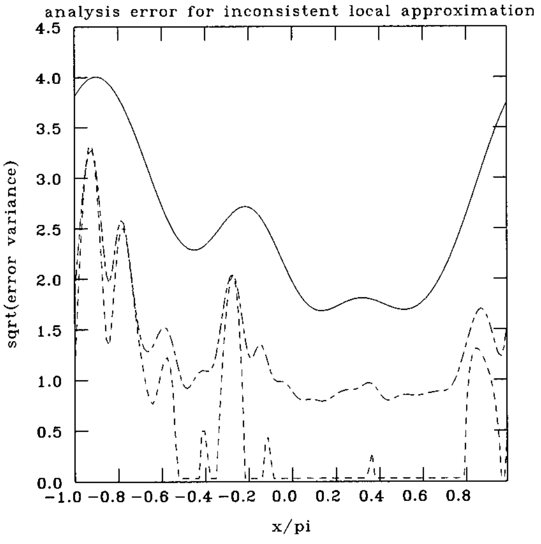


Figure 10.4
Effect of inconsistency in local approximation

10.4 Practical Implementation of the Local Approximation

It turned out to be very straightforward to implement the local approximation because it is firmly based on the observation prism structure that already exists. All the covariances, observation characteristics, vertical eigenvector decomposition, etc., were available. What was required was code to associate each grid point with a particular observation prism. Minor modification was required in the post-multiplication, so that the elements of the background error covariance did not have to be recalculated. The loop over observation prisms that is used in the NAVDAS code to calculate the Choleski matrices $\mathbf{L}_n$ for $1 \leq n \leq N$ became the central loop of the code and was extended right to the end of the program. This meant that there were no communications between processors until the very end of the programs, resulting in almost complete parallelism.

The NAVDAS analysis has no seams between observations prisms because it uses a global solve. However, an analysis error estimate based on a local approximation does have "seams," and they will almost certainly be visible. That is the price of a tractable algorithm.

Figure 10.5 shows the results of a global calculation of analysis error variance using the local approximation. We show the estimate of the error reduction due to the observation network, that is, the estimate of the term $P_bH^T[HP_bH^T + R]^{-1}HP_b$ of Eq. (2.1) derived using the approximation ((10.5)-(10.6). (It is easier to see the effect of observations in this term than by plotting the analysis error covariance itself.) We show only the square root of the variance (i.e., the diagonal of the matrix) for the 250 hPa temperature field (in degrees Kelvin, with a

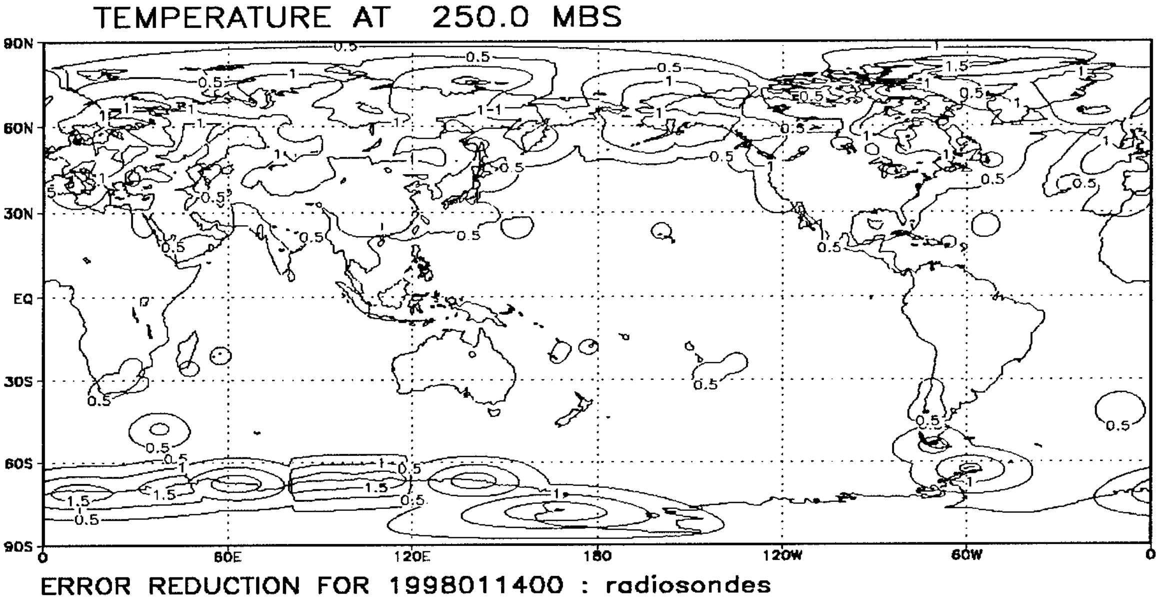


Figure 10.5a
Error reduction due to radiosondes

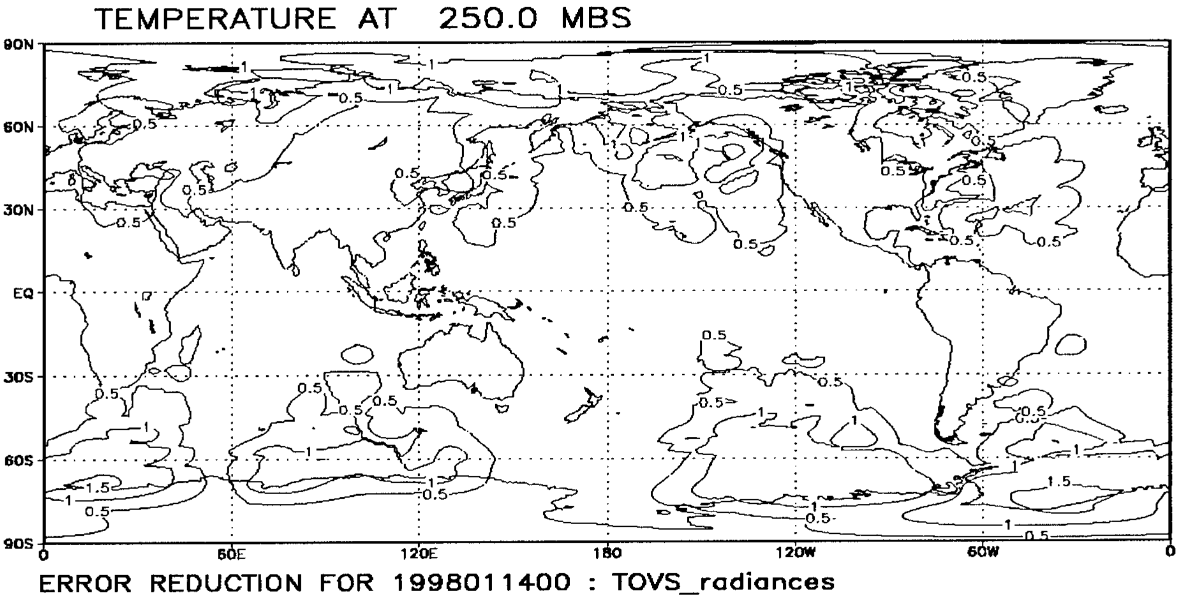


Figure 10.5b
Error reduction due to TOVS radiances

contour interval of 0.5 degrees). The results are plotted on a one-degree grid. This figure can be compared (roughly) with Fig. 5 of Fisher and Courtier (1995), which shows the same error reduction (in the geopotential) using their global approximation with 52 vectors. In Figure 10.5a, the observation set consists of 62,000 mandatory and significant level radiosonde observations of u,v,T, and in panel (b) of 32,000 TOVS radiance observations in 20 channels. Panel (b) shows four satellite passes; in panel (a), the preponderance of radiosondes over Northern Hemisphere land is obvious. Error reduction in the tropics is small because the background error is assumed to be relatively small there anyway. Both panels show some evidence of the "seams" between the observation prisms, but it is not too severe.

This algorithm has been implemented in the NAVDAS code and is run routinely.

ACKNOWLEDGMENTS

We express our appreciation to the Office of Naval Research and the Naval Research Laboratory for their support of this research under Program Element 0602435N. We also say thanks to our many colleagues in the United States and abroad: Stephen Cohn, Philippe Courtier, John Derber, Tony Hollingsworth, Joanna Joiner, Peter Lyster, Richard Menard, Arlindo da Silva, and others for many helpful discussions. We give thanks and appreciation to our co-workers for help with many facets of this work, particularly, Nancy Baker for her knowledge of TOVS assimilation and her help in interfacing with her 1DVAR, Pat Pauley for her knowledge and help in data quality control, and Tom Rosmond for advice in the MPI implementation. Alan Caughey gave very important system support for our workstations and the NRL DEC machine. We particularly thank Soheila Judy for her meticulous preparation of the document and for the enthusiasm that she brought to this project. Maureen Long and Jan Morrow of NRL's Technical Information Branch provided much-appreciated editing and design for the publication.

References

1. Baker, N., and R. Daley, 2000: Observation and background adjoint sensitivity in the adapative observation problem. *Quart. J. Roy. Meteor. Soc.* **126**, 1431-1454

2. Barker, E., 1992: Design of the Navy's multivariate optimum interpolation analysis system. *Weather and Forecasting* **7**, 220-231.

3. Benjamin, S., 1989: An isentropic meso α analysis system and its sensitivity to aircraft and surface observations. *Mon. Wea. Rev.* **117**, 1586-1603.

4. Bryson, A., and Y. Ho., 1975: *Applied optimal control.* Hemisphere Press, New York.

5. Bouttier, F., J. Derber and M. Fisher, 1997: The 1997 revision of the J_b term in 3D/4DVAR. ECMWF Research Department. Technical Memorandum.

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

7. Collins, W. and L. Gandin, 1992: Comprehensive quality control at the National Meteorolgical Center. *Mon. Wea. Rev.* **120**, 2752-2760.

8. Courtier, P., 1997: Dual formulation of four-dimensional variational assimilation. *Quart. J. Roy. Meteor. Soc.* **124**, 2449-2461.

9. Courtier, P., E. Andersson, W. Heckley, J. Pailleux, D. Vasiljevic, M. Hamrud, A. Hollingsworth, F. Rabier, and M. Fisher, 1997: The ECMWF implementation of three dimensional assimilation (3DVAR). Part I: Formulation. *Quart. J. Roy. Meteor. Soc.* **124**, 1783-1807.

10. Daley, R., 1985: The analysis of synoptic scale divergence by a statistical interpolation procedure. *Mon. Wea. Rev.* **113**, 1066-1079.

11. Daley, R., W. Wergen, and G. Cats, 1986: The objective analysis of planetary scale flow. *Mon. Wea. Rev.* **114**, 1892-1908.

12. Daley, R., 1991: *Atmospheric Data Analysis.* Cambridge University Press, Cambridge, 457 pp.

13. Daley, R., 1996: Generation of global multivariate error covariances by singular value decomposition of the linear balance equation. *Mon. Wea. Rev.* **124**, 2574-2587.

14. Daley, R., 1997: Atmospheric data assimilation. *J. Meteor. Soc. Japan.* Special volume "Data Assimilation in Meteorology and Oceanography: Theory and Practice," Vol. **75**, No. **1B**, 319-329.

15. Derber, J., and A. Rosati, 1989: A global data assimilation system. *J. Phys. Ocean* **19**, 1333-1347.

16. Derber, J.,W. Wu, M. Zupanski, D. Zupanski, D. Parrish, J. Purser, E. Rogers, Y. Lin, and G. DiMego, 1996: Variational assimilation at NCEP. Proceedings of the workshop on the non-linear aspects of data assimilation, 9-11 Sept., 1996. ECMWF, Reading, England.

17. Fisher, M., and P. Courtier, 1995: Estimating the covariance matrices of analysis and forecast error in variational data assimilation. ECMWF Technical Memorandum No. 220.

18. Goerss, J. and P. Phoebus, 1992: The Navy's operational atmospheric analysis. *Weather and Forecasting* **7**, 232-249.

19. Gaspari, G. and S. Cohn, 1999: Construction of correlation functions in two and three dimensions. *Quart. J. Roy. Meteor. Soc.* **125**, 723-757.

20. Gill, P., W. Murray, and M. Wright, 1982: *Practical optimization.* Academic Press, London, 293 pp.

21. Golub, G. and H. van Loan, 1996: *Matrix computations, third edition.* The John Hopkins University Press, 694 pp.

22. Heckley, W., P. Courtier, J. Pailleux, and E. Andersson, 1992: The ECMWF variational analysis: General formulation and use of background information. In Workshop Proceedings of "Variational assimilation, with special emphasis on three-dimensional aspects," ECMWF, Reading, UK, November 9-12, 1992.

23. Hodur, R., 1997: The Naval Research Laboratory's Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS). *Mon. Wea. Rev.* **125**, 1414-1430.

24. Hogan, T. and T. Rosmond, 1991: The description of the Navy Operational Global Atmospheric Prediction Systems spectral forecast model. *Mon. Wea. Rev.* **119**, 1786-1815

25. Joiner, J. and A. da Silva, 1998: Efficient methods to assimilate satellite retrievals based on information content. *Quart. J. Roy. Meteor. Soc.* **124**, 1669-1694.

26. Lonnberg, P. and A. Hollingsworth, 1986: The statistical structure of short-range forecast errors as determined from radiosonde data. Part II: The covariance of height and wind errors. *Tellus* **38A**, 137-161.

27. Lorenc, A., 1981: A global three-dimensional multivariate statistical analysis system. *Mon. Wea. Rev.* **109**, 701-721.

28. Lorenc, A., 1986: Analysis methods for numerical weather prediction. *Quart. J. Roy. Meteor. Soc.* **112**, 205-240.

29. Lyster, P., J. Larson, C. Ding, J. Guo, W. Sawyer, A. da Silva, and I. Stajner, 1997: Requirements and preliminary design of the Goddard Earth Observing System (GEOS) parallel physical space statistical analysis system (PSAS). DAO Office Note 97-05. Data Assimilation Office, Goddard Space Flight Center, Greenbelt, Maryland 20771.

30. Menard, R., S. Cohn, P. Lyster, and L. Chang, 1999: Stratospheric assimilations of chemical tracer observations using a Kalman filter. Part II: χ^2 validated results and analysis of variance and correlation dynamics. *Mon. Wea. Rev.* (in press)

31. Onogi, K., 1998: A data quality control method using forecasted horizontal gradient and tendency in a NWP system: dynamic QC. *J. Meteor. Soc. Japan* **76**, 497-515.

32. Parrish, D. and J. Derber, 1992: The National Meteorological Center's spectral statistical analysis system. *Mon. Wea. Rev.***120**, 1747-1763.

33. Pauley, P. and E. Stephens, 1998: A comparison of upper front strength analyzed by NORAPS and as observed by ACARS-equipped aircraft. Reprint 16th AMS Conference on Weather Forecasting and Analysis, 11-16 January, Phoenix, Arizona.

34. Passi, R., K. Goodrich, J. Derber, and M. Limber, 1993: An efficient data assimilation algorithm with a Gaussian covariance structure. COAM Center for Ocean and Atmospheric Modelling. The University of Southern Mississippi TR-2/94

35. Phalipou, L., 1996: Variational retrieval of humidity profile, wind speed and cloud liquid water path with the SSM/I; Potential for numerical weather prediction. *Quart. J. Roy. Meteor. Soc.* **122**, 327-355.

36. Phalipou, L. and E. Gerard, 1996: Use of precise microwave imagery in numerical weather prediction. ESA Study Report. ECMWF, Reading, UK.

37. Polavarapu, S., 1995: Divergent wind analysis in the oceanic boundary layer. *Tellus* **47A**, 221-239

38. Riishojgaard, L., 1998: A direct way of specifying flow dependent background error for meteorological analysis systems. *Tellus* **50A**, 42-57.

39. Riishojgaard, L., 2000: A method for estimating the analysis error variance in a physical space data assimilation system. *Quart. J. Roy. Meteor. Soc.* **126**, 1367-1386.

40. Rodgers, C., 1998: Information content and optimisation of high spectral resolution measurements. *Adv. Space. Res.* **21**, 361-367.

41. Shannon, C. and W. Weaver, 1949: *The Mathematical Theory of Communication.* University of Illinois Press.

42. Soffelen, A. and D. Anderson, 1997: Ambiguity removal and assimilation of scatterometer data. *Quart. J. Roy. Meteor. Soc.* **123**, 491-518.

43. Tarantola, A., 1987: *Inverse problem theory.* Elsevier, Amsterdam, 613 pp.

44. Walsh, G., 1975: *Methods of optimization.* Wiley, New York, 304 pp.

45. Xu, L., and R. Daley, 2000: Towards a true 4-dimensional data assimilation algorithm: Application of a cycling representer algorithm to a simple transport problem. *Tellus* **52A,** 109-128.

Appendix A

Calculating Great Circle Distance and Radial Derivatives

The Great Circle distance between a point with latitude and longitude (θ_1, λ_1) and a second point (θ_2, λ_2) can be written

$$s = a \cos^{-1}[\sin \theta_1 \sin \theta_2 + \cos \theta_1 \cos \theta_2 \cos(\lambda_2 - \lambda_1)], \qquad \text{(A1)}$$

where "a" is the radius of the Earth. This is a very slow algorithm; it requires five trigonometric evaluations plus an evaluation of the arcosine (which is very slow).

This can be made more efficient, as follows. Write the expression as

$$s = a \cos^{-1}[x]\,,\ \text{with}\ x = a_1 a_2 + b_1 b_2 + c_1 c_2\,,\ \text{and}\ a_1 = \sin(\theta_1)\,,\ a_2 = \sin(\theta_2), \qquad \text{(A2)}$$

$$b_1 = \cos(\theta_1)\cos(\lambda_1)\,,\ b_2 = \cos(\theta_2)\cos(\lambda_2)\,,\ c_1 = \cos(\theta_1)\sin(\lambda_1)\,,\ c_2 = \cos(v\theta_2)\sin(\lambda_2)\,.$$

Note that a_1 , b_1, and c_1 are only functions of the first location; similarly, a_2 , b_2 , c_2 are only functions of the second location—these constants can thus be precalculated for each location without reference to the other location. The constants can then be combined using three multiplies into x.

Calculating Great Circle Distance with a Look-up Table

x varies between zero and one, with x = 1 corresponding to zero Great Circle distance and x = 0 corresponding to the maximum Great Circle distance. Examination of the functional form of the arcosine function (see Abramovitz and Stegun (1960, p. 80) indicates that its derivative is singular at x =1, i.e., there is great sensitivity for small Great Circle distances. However, the transformation to the arcsine function transfers the sensitivity to large values of Great Circle distance, that is, $\sin^{-1}[(1 - x^2)^{1/2}]$ is well-behaved for x near 1. Since correlation functions are large at small Great Circle distances and small at large Great Circle distances, it is better to have the sensitivity at large s (small x). A table look-up can then be safely applied to the arcsine function.

Calculating Radial Derivatives

Three radial derivatives are required in the calculation of the multivariate correlations.

If f(s) is the correlation function, then we require df/ds in the calculation of the <Zu>, <Zv> correlations and s^{-1} df/ds and d^2f/ds^2 in the calculation of the <uu>, <vv>, and <vu> correlations. All three radial derivatives are well-behaved at s = 0 (for normal correlation functions).

In the NRL MVOI code, df/ds and d^2f/ds^2 are calculated first, and s^{-1} df/ds is later calculated when required. Special code had to be written to take care of the problem of small s. In the NAVDAS code, this problem is circumvented by first calculating s^{-1} df/ds and d^2f/ds^2, which are both well-behaved at s = 0. Then, when df/ds is required, it can simply be obtained by multiplying s^{-1} df/ds by s, which is straightforward.

Another change from the NRL MVOI code is that s is first normalized by the horizontal correlation length L_h, before calculating f(s) and the radial derivatives.

Two horizontal correlation functions are illustrated in Figures A1 and A2 (see Section 4.6.2). Figure A1 shows the SOAR function, and Fig. A2, shows the compact spline function. In each panel, the abscissa shows normalized Great Circle distance(s) and the ordinate-correlation. The solid curve is the correlation function f(s), the dash-dot curve is df/ds, and the dashed curve is d^2f/ds^2. The compact spline bears some resemblance to the Gaussian correlation function.

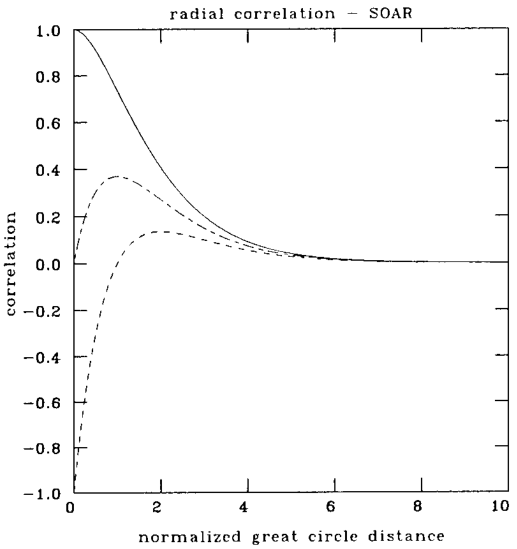


Figure A1
SOAR function

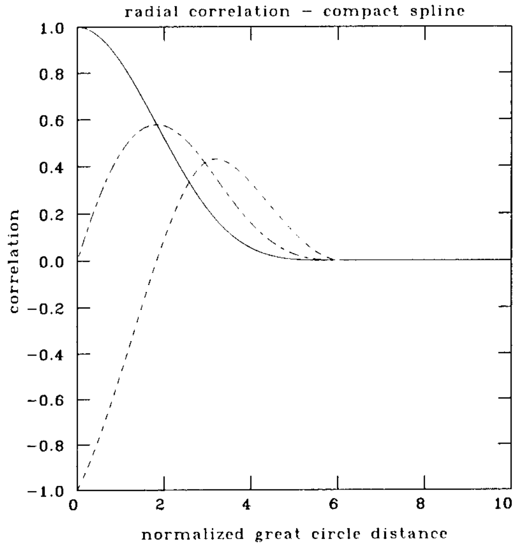


Figure A2
Compact spline function

Finding the Centroid of an Observation Prism or Analysis Gridbox on the Sphere

For the global case, observation prisms or analysis gridboxes that are more than a certain maximum distance apart (many thousands of kilometers) are not allowed to interact, because the background correlations are vanishingly small at these distances. In order to perform this calculation, we must be able to find approximately, the centroid of each analysis grid box and observation prism. While this is simple to do on a limited (x,y) domain, we wish to be able to perform this calculation both for a limited domain and on a sphere, all with one code. This obliges us to perform this calculation using latitude θ and longitude λ. The following is a simple method for performing this calculation, which takes advantage of Eqs. (A1) and (A2).

Suppose there are K points (θ_k,λ_k) in some observation prism or analysis grid box. We wish to find the centroid (θ_c,λ_c). Denote $s(\theta_c,\lambda_c,\theta_k,\lambda_k)$ as the Great Circle distance between points "k" and "c" and define it following Eqs. (A1) and (A2),

$$\cos[a^{-1}s(\theta_c,\lambda_c,\theta_k,\lambda_k)] = x_k = a_k\sin\theta_c + b_k\cos\theta_c\cos\lambda_c + c_k\cos\theta_c\sin\lambda_c, \qquad \text{(A3)}$$

where $a_k = \sin\theta_k$, $b_k = \cos\theta_k\cos\lambda_k$, and $c_k = \cos\theta_k\sin\lambda_k$.

The centroid of the analysis grid box or observation prism can be obtained by minimizing $K^{-1}\sum_{k=1}^{K} s(\theta_c,\lambda_c,\theta_k,\lambda_k)$ with respect to θ_c and λ_c. However, note that as the Great Circle distance gets smaller, x_k

in Eq. (A3) gets larger. Thus, an alternate and more straightforward procedure for deriving an approximate centroid can be derived by maximizing,

$$K^{-1}\sum_{k=1}^{K} x_k = A\sin\theta_c + B\cos\theta_c\cos\lambda_c + C\cos\theta_c\sin\lambda_c, \qquad \text{(A4)}$$

with respect to θ_c and λ_c. Here, $A = K^{-1}\sum_{k=1}^{K}\sin\theta_k$, $B = K^{-1}\sum_{k=1}^{K}\cos\theta_k\cos\lambda_k$, and $C = K^{-1}\sum_{k=1}^{K}\cos\theta_k\sin\lambda_k$. This will yield a good estimate of the centroid as long as $|\theta_k - \theta_c|$ and $|\lambda_k - \lambda_c|$ are small for all k. That means the observation prisms or analysis grid boxes are not too large. Differentiating (A4) with respect to θ_c and λ_c respectively, and setting the results to zero, yields

$$A\cos\theta_c = \sin\theta_c[B\cos\lambda_c + C\sin\lambda_c] \quad \text{and} \quad B\cos\theta_c\sin\lambda_c = C\cos\theta_c\cos\lambda_c. \qquad \text{(A5)}$$

From (A5), we can determine four extrema,

(1) $\lambda_c = \tan^{-1}[C/B]$, $\theta_c = \tan^{-1}[A/(B\cos\lambda_c + C\sin\lambda_c)]$, (A6)
(2) $\lambda_c + 180^0$, $-\theta_c$ (antipodal point),
(3) $\theta_c = 90^0$ (North Pole),
(4) $\theta_c = -90^0$ (South Pole).

In general, extremum (1) is the absolute maximum, although it might coincide with extremum (3) or (4) for points scattered around the poles.

Experiments using (A6) seem to give a good centroid, provided the points are not too widely scattered.

The Most Accurate Great Circle Formula

A more accurate expression than Eq. (A1) is the so-called haversine formula,

$$s = 2\ a\ \sin^{-1}[\sin^2((\theta_2 - \theta_1)/2) + \cos(\theta_2)\cos(\theta_1)\sin^2((\lambda_2 - \lambda_1)/2). \qquad \text{(A7)}$$

Equation (A7) is particularly accurate for small s. From the point of view of efficiency, (A7) is about as efficient as (A1). It is possible to create a form of (A7) that is similar to (A2) and almost as efficient. Unfortunately, it has about the same accuracy as (A1) and (A2) and is not as accurate as (A7). Consequently, we use (A7) only if accuracy is very important.

Independence of the NAVDAS System from the Analysis Grid

The horizontal locations of both the observations and the analysis gridpoints are defined entirely in terms of their latitudes and longitude, not with respect to the analysis grid. They can be processed in any order, and all sorting into observation prisms or analysis volumes requires only latitude/longitude information. This generality is possible because of the operations described in this Appendix.

This generality permits an enormous flexibility in the form of the analysis grid. For example, it is possible to perform the analysis on three (or more) nested grids simultaneously. One could provide a simultaneous analysis for three (or more) unconnected COAMPS areas at the same time (and have each of them multiply nested). Whether or not this would be a good idea is not clear, but the capability is there.

Appendix B

Calculating Wind/Wind and Wind/Geopotential Correlations in Spherical Coordinates (Including Correlations with the Divergent Wind)

In the NRL MVOI analysis, the correlation models use a polar stereographic projection for winds in the volumes over the poles, and a spherical north and east projection elsewhere. This does not work in a 3DVAR algorithm because the correlations are needed between all observations, not just within volumes, so a single correlation model is needed. Two different approaches were found that work over the globe in physical space, one developed by NASA Goddard (Cohn et al, 1998), and the other by ECMWF (1994).

Case 1 - No Mass/Divergent Wind and Rotational Wind/Divergent Wind Correlations

The NASA approach relates the wind correlations to the streamfunction correlation model using latitude θ and longitude λ as the independent variables. The correlation model is a function of the Great Circle distance between two locations, which in turn is a function of θ and λ. The winds are described in terms of the gradient of the streamfunction, and derivatives are determined using the chain rule on the Great Circle distance formula. The final formulas are functions of the Great Circle distance, resulting in many complicated trigonometric functions. Computationally, these functions are very expensive.

The ECMWF method uses vectors in a Cartesian coordinate system to relate the vector connecting two locations to local north and east coordinates. Their method is similar to the one described in Daley (1991), except the local coordinate is not the same at the two locations. In Daley (1991), the correlations on a tangential plane in polar coordinates are derived from expressions of correlation on a sphere, whereas the ECMWF derivation starts in polar coordinates to derive the relationships in spherical coordinates.

We have chosen the ECMWF system since it is an extension of the method in use in the MVOI and can be computed from correlations computed on a tangential polar plane. In this coordinate system, the correlation models for stream function and velocity potential are functions of the Great Circle distance between the two locations in question. From Daley (1991), the equations relating the longitudinal $\vec{l}$ and transverse $\vec{t}$ components of the wind correlations c_{ll} and c_{tt} are

$$c_{ll}(r) = -\frac{1}{r}\frac{d}{dr}c_{\psi\psi} - \frac{d^2}{dr^2}c_{\chi\chi}, \tag{B1}$$

$$c_{tt}(r) = -\frac{d^2}{dr^2}c_{\psi\psi} - \frac{1}{r}\frac{d}{dr}c_{\chi\chi}, \tag{B2}$$

$$c_{lt}(r) = c_{tl}(r) = 0, \tag{B3}$$

$$c_{\phi t} = -c_{t\phi} = \frac{d}{dr}c_{\phi\varphi},\ c_{\phi l} = -c_{l\phi} = \frac{d}{dr}c_{\phi\chi} = 0, \tag{B4}$$

where the correlation between ϕ and χ is assumed to be 0 for this derivation. These equations are derived in Daley (1991).

To introduce the coordinate system needed to represent the correlations on the Earth's surface in spherical coordinates, consider the two locations written with respect to a Cartesian coordinate system:

$$\vec{m} = r[\cos(\lambda_m)\cos(\theta_m)\vec{i} + \sin(\lambda_m)\cos(\theta_m)\vec{j} + \sin(\theta_m)\vec{k}]$$

$$\vec{n} = r[\cos(\lambda_n)\cos(\theta_n)\vec{i} + \sin(\lambda_n)\cos(\theta_n)\vec{j} + \sin(\theta_n)\vec{k}].$$

The angle α between $\vec{m}$ and $\vec{n}$ is computed from $\vec{m} \cdot \vec{n}$, where

$$r \cdot \cos(\alpha) = r[\cos(\lambda_m)\cos(\theta_m)\cos(\lambda_n)\cos(\theta_n) + \sin(\lambda_m)\cos(\theta_m)\sin(\lambda_n)\cos(\theta_n) + \sin(\theta_m)\sin(\theta_n)]$$

so that the Great Circle distance is

$$s_{mn} = r\alpha.$$

The vector between the points is

$$r_{mn}\vec{l} = \vec{m} - \vec{n} =$$

$$r\{[\cos(\lambda_m)\cos(\theta_m) - \cos(\lambda_n)\cos(\theta_n)]\vec{i} + [\sin(\lambda_m)\cos(\theta_m) - \sin(\lambda_n)\cos(\theta_n)]\vec{j} + [\sin(\theta_m) - \sin(\theta_n)]\vec{k}\}.$$

The local north direction for location $\vec{m}$ can be derived from this equation assuming constant λ along an infinitesimal increment in θ, or

$$\vec{N}_m = -\cos(\lambda_m)\sin(\theta_m)\vec{i} - \sin(\lambda_m)\sin(\theta_m)\vec{j} + \cos(\theta_m)\vec{k}.$$

Likewise, the local east direction is

$$\vec{E}_m = -\sin\lambda_m\vec{i} + \cos(\theta_m)\vec{j}.$$

Projecting the vector between $\vec{m}$ and $\vec{n}$ onto the local north and east directions gives

$$r_{mn}\vec{l}_m = (r_{mn}\vec{l} \cdot \vec{N}_m)\vec{N}_m + (r_{mn}\vec{l} \cdot \vec{E}_m)\vec{E}_m.$$

The angle β between $r_{mn}\vec{l}$ and $\vec{N}_m$ is

$$\sin(\beta_m) = \frac{r_{mn}\vec{l} \cdot \vec{E}}{\sqrt{(r_{mn}\vec{l} \cdot \vec{N})^2 + (r_{mn}\vec{l} \cdot \vec{E})^2}}. \tag{B5}$$

There are two conditions for which this formulation becomes singular: over the poles, and when the Great Circle distance goes to zero. To avoid these singularities, the locations are incrementally adjusted to make the computations well behaved.

The relationship between the longitudinal and transverse components of the north and east components of wind as shown in Fig. B1 are

$$v\vec{N} = v_t\vec{t} + v_l\vec{l} = v\cdot\sin(\beta) + v\cdot\cos(\beta), \qquad \text{(B6)}$$

$$u\vec{E} = u_l\vec{l} - u_t\vec{t} = u\cdot\sin(\beta)\vec{l} - u\cdot\cos(\beta)\vec{t}\,.$$

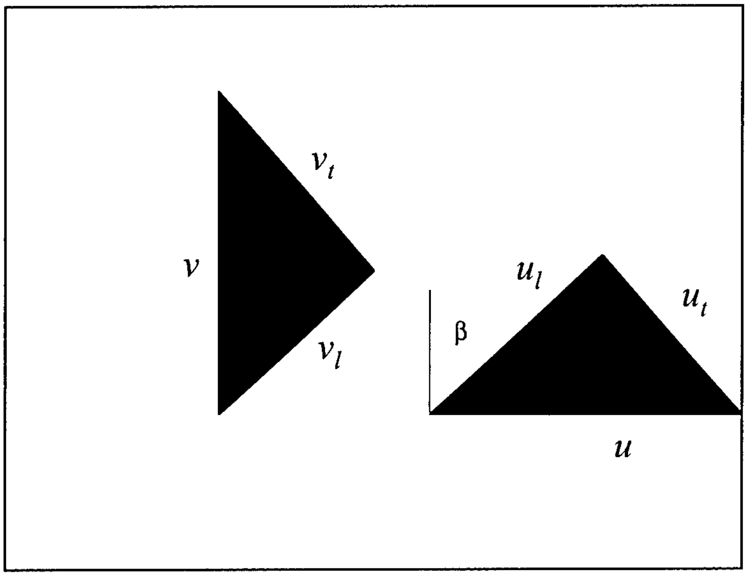


Figure B1
Relationship of the north and east components of the wind to their corresponding longitudinal and transverse components. The angle β is the angle between local north and the vector connecting two locations defining the correlation separation.

Between two points $\vec{m}$ and $\vec{n}$, the wind correlations are related as follows:

$$< u_m\vec{N}\cdot u_n\vec{N} > = < u_m\vec{l}\cdot u_n\vec{l} > \sin(\beta_m)\sin(\beta_n) + < u_m\vec{t}\cdot u_n\vec{t} > \cos(\beta_m)\cos(\beta_n).$$

Normalizing with the background error variances and using the definitions for c_{ll} and c_{tt} gives

$$< u_m u_n > / P_u^2 = c_{ll}\sin(\beta_m)\sin(\beta_n) + c_{tt}\cos(\beta_m)\cos(\beta_n).$$

Similarly,

$$\begin{aligned}
< v_m v_n > / P_v^2 &= c_{ll}\cos(\beta_m)\cos(\beta_n) + c_{tt}\sin(\beta_m)\sin(\beta_n),\\
< u_m v_n > / P_u P_v &= c_{ll}\sin(\beta_m)\cos(\beta_n) - c_{tt}\cos(\beta_m)\sin(\beta_n),\\
< v_m u_n > / P_u P_v &= c_{ll}\cos(\beta_m)\sin(\beta_n) - c_{tt}\sin(\beta_m)\cos(\beta_n), \qquad \text{(B7)}\\
< \phi_m u_n > / P_\phi P_u &= -c_{\phi t}\cos(\beta_n),\\
< u_m \phi_n > / P_u P_\phi &= -c_{t\phi}\cos(\beta_m),\\
< \phi_m v_n > / P_\phi P_v &= +c_{\phi t}\sin(\beta_n),\\
< v_m \phi_n > / P_v P_\phi &= +c_{t\phi}\sin(\beta_m).
\end{aligned}$$

These covariance models can be converted to any grid projection. First, wind conversion coefficients are determined from

$$\vec{l}\cdot\vec{E}_m = u_{1,m}\vec{i}_g + u_{2,m}\vec{j}_g,$$

$$\vec{l}\cdot\vec{N}_m = v_{1,m}\vec{i}_g + v_{2,m}\vec{j}_g,$$

for location $\vec{m}$ with similar relations for location $\vec{n}$. Using the carat to designate winds projected to the particular analysis grid we wish to convert to, the conversion equations evaluated at $\vec{m}$ are

$$\hat{u}_m = u_{1,m}u_m + u_{2,m}v_m \tag{B8}$$

$$\hat{v}_m = v_{1,m}u_m + v_{2,m}v_m$$

Using these equations to define the covariance relations gives

$$\begin{aligned}
<\hat{u}_m\hat{u}_n> &= <u_m u_n> u_{1,m}u_{1,n} + <u_m v_n> u_{1,m}u_{2,n} + <v_m u_n> u_{2,m}u_{1,n} + <v_m v_n> u_{2,m}u_{2,n},\\
<\hat{u}_m\hat{v}_n> &= <u_m u_n> u_{1,m}v_{1,n} + <u_m v_n> u_{1,m}v_{2,n} + <v_m u_n> u_{2,m}v_{1,n} + <v_m v_n> u_{2,m}v_{2,n},\\
<\hat{v}_m\hat{u}_n> &= <u_m u_n> v_{1,m}u_{1,n} + <u_m v_n> v_{1,m}u_{2,n} + <v_m u_n> v_{2,m}u_{1,n} + <v_m v_n> v_{2,m}u_{2,n},\\
<\hat{v}_m\hat{v}_n> &= <u_m u_n> v_{1,m}v_{1,n} + <u_m v_n> v_{1,m}v_{2,n} + <v_m u_n> v_{2,m}v_{1,n} + <v_m v_n> v_{2,m}v_{2,n},\\
<\hat{\phi}_m\hat{u}_n> &= <\phi_m u_n> u_{1,n} + <\phi_m v_n> u_{2,n},\\
<\hat{\phi}_m\hat{v}_n> &= <\phi_m u_n> v_{1,n} + <\phi_m v_n> v_{2,n},\\
<\hat{v}_m\hat{\phi}_n> &= <u_m\phi_n> v_{1,m} + <v_m\phi_n> v_{2,m},\\
<\hat{u}_m\hat{\phi}_n> &= <u_m\phi_n> u_{1,m} + <v_m\phi_n> u_{2,m}.
\end{aligned} \tag{B9}$$

These conversion equations were applied to the spherical covariances to produce the corresponding correlation plots shown in Fig. B2. In the actual analysis, the data are kept in spherical coordinates for all of the grids until the analysis is complete, and then are converted to the grid desired. This saves making the conversion to the covariances during the expensive iterative solution. The correlations computed using these formulas are given in Fig. B2. Note that the lines of symmetry in the spherical formulation may be curved.

Case 2 - Inclusion of Correlations with the Divergent Wind

Equations (B1)-(B9) assume that the divergent part of the wind is not correlated with either the mass field or the rotational wind field (see Eq. B4). To accommodate inflow into low-pressure regions and outflow from high-pressure regions in the planetary boundary layer, it is necessary to relax this constraint. The following derivation follows Daley (1985). We assume that the <ψχ> and <Φχ> correlations are isotropic and thus

$c_{\psi\chi} = c_{\chi\psi}$ and $c_{\Phi\chi} = c_{\chi\Phi}$. Define $\gamma_1(r) = \frac{1}{r}\frac{\partial}{\partial r}c_{\psi\chi}$, $\gamma_2(r) = \frac{\partial^2}{\partial r^2}c_{\chi\psi}$, and $\pi(r) = \frac{\partial}{\partial r}c_{\Phi\chi}$.

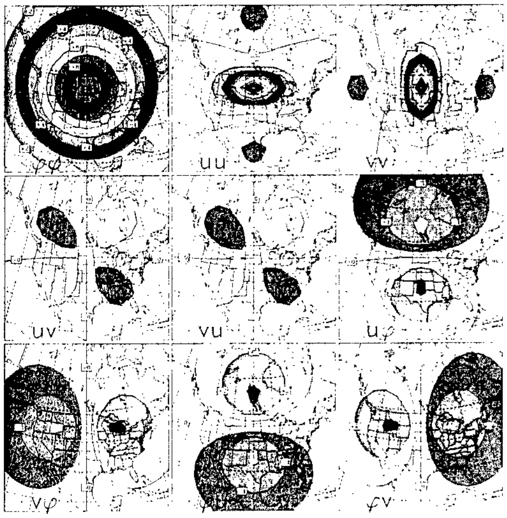


Figure B2
The 3DVAR correlation models plotted in a Lambert Conformal Projection using the SOAR model (4.22) with $L_h^{-1} = 0.96$, $\mu = 1.0$, and $\nu = 0$

Then,

$$c_{f\iota}(r) = c_{\iota f}(r) = \gamma_1(r) - \gamma_2(r), \text{ and } c_{\Phi f}(r) = -c_{f\Phi}(r) = \pi(r). \tag{B10}$$

We can write the additional contributions to Eqs. (B7) as follows:

$$\begin{aligned}
\langle u_m u_n \rangle / P_u^2 &= 2[\gamma_1 \sin(\beta_m)\cos(\beta_n) - \gamma_2 \cos(\beta_m)\sin(\beta_n)], \\
\langle v_m v_n \rangle / P_v^2 &= -2[\gamma_1 \sin(\beta_n)\cos(\beta_m) - \gamma_2 \cos(\beta_n)\sin(\beta_m)], \\
\langle u_m v_n \rangle / P_u P_v &= c_{f\iota}[\sin(\beta_m)\sin(\beta_n) - \cos(\beta_m)\cos(\beta_n)], \\
\langle v_m u_n \rangle / P_v P_u &= c_{f\iota}[\sin(\beta_m)\sin(\beta_n) - \cos(\beta_m)\cos(\beta_n)], \\
\langle \Phi_m u_n \rangle / P_\Phi P_u &= -\pi \sin(\beta_n), \\
\langle u_m \Phi_n \rangle / P_u P_\Phi &= \pi \sin(\beta_m), \\
\langle \Phi_m v_n \rangle / P_\Phi P_v &= -\pi \cos(\beta_n), \\
\langle v_m \Phi_n \rangle / P_v P_\Phi &= \pi \cos(\beta_n).
\end{aligned} \tag{B11}$$

As discussed in Daley (1985), the general effect of these extra terms is to cause a rotation of all wind/wind and wind/mass correlations. The rotation is clockwise in the Northern Hemisphere and anticlockwise in the Southern Hemisphere. There is also an increase in the amplitude of the <uv> and <vu> correlations.

Section 4.7.3 demonstrated how a nonseparable formulation of the background error correlation can produce a maximum correlation with the divergent wind near the Earth's surface. This will have the effect of producing a maximum rotation of the correlations near the ground, with the rotation diminishing with increasing altitude.

Figure B3 illustrates the effect of correlations with the divergent wind. This is the <vv> correlation (v being the northward wind component). This correlation is at 1000 hPa at 45 degrees north using the SOAR correlation. It can generally be compared with the upper right panel of Fig. B2. Note, however, that Fig. B3 is on a latitude/ longitude projection, whereas Fig. B2 is a Lambert conformal projection to a different scale. The clockwise rotation of the correlation is clearly evident in Fig. B3.

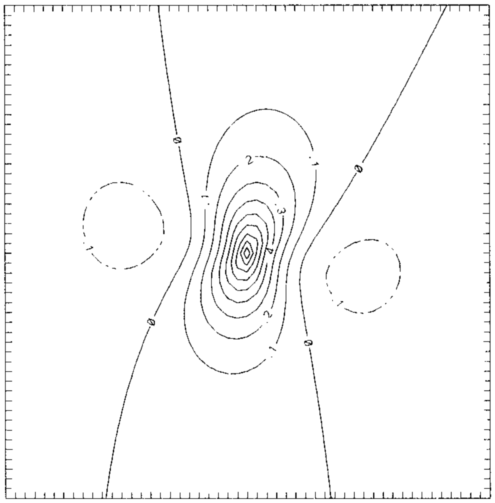


Figure B3
<vv> correlation at 1000 mb at 45°N

Appendix C

Constructing Code for Parallel Processors

An efficient method for implementing the pre-conditioned conjugate gradient solution (3.13) of the analysis algorithm (3.5)-(3.6) on parallel processors has been discussed by Lyster et al (1997). The MPI (message passing interface) instructions are used. The basic idea is quite simple. Suppose there are N observations and M analysis gridpoints. Then in Eqs. (3.5-3.6) there are basically two very expensive operations.

In the solver (Eq. (3.5)) (solved using the conjugate gradient algorithm (3.13)), the most expensive operation is the matrix vector multiplication (denoted $\mathbf{q}_k = \mathbf{A}\mathbf{p}_k$ in Eq. (3.13)). This operation requires $O(N^2)$operations for each iteration of the descent. In addition, the N×N matrix **A** must be constructed, which again requires $O(N^2)$ operations, although this operation may only have to be performed once if there is sufficient storage. In general, we only store the diagonal blocks of this matrix, but we recalculate the off-diagonal blocks at every iteration.

The second important operation is the post-multiplication (Eq. (3.6)), which is a one-time matrix/vector multiplication with MN operations. The MN matrix elements must also be calculated and used once.

All other operations (including the preconditioner) are O(N) or O(M). In general, M > N, so that the O(MN) operation of the post-multiplier will be more expensive than a single application of the $O(N^2)$ operation in the solver. However, given that the solver may take many iterations, both operations can be considered to be roughly equally expensive. (For example, if M = 10N and it takes 10 iterations of the solver.) In the limit as M,N become very large, the algorithm will be completely dominated by the O(MN) and $O(N^2)$ operations discussed above.

For a machine with J processors, the key to a scalable implementation is to divide the O(MN) and $O(N^2)$ operations equally between the processors. To keep communications between processors to a minimum, the second principle is to ensure that matrix elements are calculated, stored, and used on each processor; they are never passed between processors. At most, elements of vectors may be passed between processors.

Since matrices may have N^2 or MN elements and vectors only M or N elements, this obviously reduces the size and number of messages that must be passed between processors. Thus, a rough operation count for a parallel version of this algorithm on a J processor machine is

$$\text{Number of operations} \approx l_1N + l_2M + m_1N^2/J + m_2MN/J + c_1(J^2)N + c_2(J^2)M, \qquad \text{(C1)}$$

where l_1 and l_2 indicate O(N) or O(M) operations that are performed on all processors. m_1 and m_2 are operations involving matrix construction and matrix/vector multiplication, which are divided equally among processors. $c_1(J^2)$ and $c_2(J^2)$ indicate elements of vectors that must be communicated between processors. (Generally speaking, communication increases as the square of the number of processors). Note in the limit as M,N become much larger than J, the algorithm is dominated by the m_1 and m_2 terms and is thus asymptotically scalable. When M,N are smaller, the l_1 and l_2 terms become relatively more important. Communication costs can be expensive, costing perhaps 100 times as much to communicate one number between processors as it does to calculate the number within a processor. Thus, depending on the machine, communication could be relatively costly if M,N are too small.

For global problems, we might expect $M = 10^7$, $N = 10^5$, and $J < 10^3$, so Eq. (C1) suggests that this algorithm should be scalable, even on machines with very costly communications. For regional problems run on smaller machines, $M = 10^6$, $N = 10^4$, $J < 20$, but we might expect more efficient communication (shared memory), and scalability is also possible (and has been already demonstrated to some extent)

We now describe the present implementation in more detail.

Implementation

In this implementation, the following three considerations are very important.

(1) The serial and parallel versions should be almost identical. This requires that all MPI instructions be confined to a very small number of subroutines (< 10). Thus, most of the code is unaware of whether it is being executed serially or in parallel.

(2) The MPI instruction set should be very small.

(3) The code should be modified from a logical serial flow only if there is an enormous amount to be gained computationally by modifying the code in a nonintuitive way for parallel execution.

Thus, the present implementation is a very basic MPI installation. There are four assumptions.

(1) On every processor there is a complete copy of the innovation vector and all ancillary information, such as observation locations, observation errors, relevant background information, etc. However, see (4) below.

(2) The elements of the $\mathbf{HP_bH^T + R}$ matrix are distributed across the processors, but are calculated at every iteration step except for the (symmetric) diagonal blocks, which are calculated only once and then stored on the appropriate processor. The amount of calculation required is less than in a purely physical space implementation.

(3) The symmetry of the $\mathbf{HP_bH^T + R}$ matrix is completely exploited in both the block diagonal matrices and the off-diagonal blocks. This halves the expense required in calculating the elements of this matrix. There is also an option to not exploit the matrix symmetry in the off-diagonal blocks.

(4) Vertical eigenvector decomposition (Sections 4.4.1 and 5.3) is spread across processors. In particular, for instruments such as TOVS (Section 5.3), information required for the construction and multiplication of the $\underline{\mathbf{H}}$ and $\underline{\mathbf{H}^\mathrm{T}}$ matrices would have to be stored only on the processor that processed that particular sounding.

The Solver

The solver (3.13) is implemented as follows. The innovation vector and relevant observation information (three-dimensional locations, error information, etc.) is assumed to be available on all processors. It has already been sorted into triangular prisms (Fig. 3.2). The calculation of the elements of the $\mathbf{HP_bH^T + R}$ matrix is divided equally among the processors. This is done by first assigning a different subset of the observation prisms to each processor, and then calculating the interactions between the observations in that subset of prisms with all the observations at every iteration. (as noted above, the diagonal blocks (self-interactions) are only calculated once and stored). The idea is to have roughly the same number of interactions (i.e., matrix elements) on each processor.

Originally, we attempted to load balance the processors for the solver calculation by estimating the amount of work in the calculation and then distributing this work across the processors using a load-balancing routine. This did not work very well because the solver is entirely in observation space, and it is very difficult to estimate the load a priori. We decided to switch to a self-scheduling algorithm, which uses the master processor to control the process while the slaves do all the work. Thus, one processor is sacrificed, which does not matter when there are many processors. In this algorithm, load balancing is dynamically controlled and works much more effectively. It also adjusts automatically to changes in the background error covariance calculation and adjustments in the load.

The descent algorithm (3.13) can be run in parallel, except for the following messages that must be passed between processors once per iteration. First, we must calculate and pass the scalars α_k, β_k, etc. between each processor. More costly, once we have calculated the subvectors of the vector $\mathbf{p}_k$ on each processor, we must pass this information (after vertical eigenvector decomposition (see (4) above) to all of the other processors, so that each processor has available a complete vector of (vertically decomposed) $\mathbf{p}_k$ before beginning the computation $\mathbf{q}_k = \mathbf{A}\mathbf{p}_k$. There is one additional communication (the vertically decomposed form) of the vector $\mathbf{q}_k$ between processors.

After the solver has finished, (the vertically decomposed form) of the vector **z** in Eq. (3.5) must also be communicated between processors.

The Second Preconditioner

There are some complications for the implementation of the second preconditioner (discussed in Section 3.5) for parallel implementations. This second preconditioner is based on a re-sorting of the observations. There is an inherent conflict between the operations of re-sorting the observations and splitting the load (i.e., the observations) among processors. In fact, the operations of load splitting and re-sorting do not commute. Although there is a complete set of observations on every processor, many of the necessary fields (particularly the $\underline{\mathbf{H}}$ and $\underline{\mathbf{H}^T}$ Jacobian matrices for directly assimilated observations) are not available on every processor. There are two ways to tackle this problem in a parallel environment.

The first way is to re-sort the entire observation set and then split this re-sorted set of observations among processors. This will mean that the observations from the original sort that are to be treated on a given processor will not necessarily correspond to the observations from the second sort that are to be treated on that processor. The same goes for the elements of the $\underline{\mathbf{H}}$ and $\underline{\mathbf{H}^T}$ matrices for direct assimilation instruments (see assumption (4) above). This would imply a large amount of extra message passing. This procedure seems undesirable.

The second way is to load split the observations (i.e., load) among each processor used in the original sort. Then, the observations are re-sorted on each processor. This means the same set of observations will be treated on each processor for both the first and second preconditioners. However, a re-sorting will not be effective if all the prisms on a given processor are widely separated, because most of the correlations (apart from those internal to the original prisms) will be very small, and a re-sorting would tend to produce basically the same prism structure as in the original sort. According to Section 3.5, this re-sort will not be beneficial. What is required is that the prisms on a given processor be from the same local neighborhood (i.e., clustered together), so that a preconditioner based on the second sort will include many large correlations, which were not included in the original preconditioner. This provides an extra condition on the solver load balancer, as noted above. It should be noted that re-sorting on processors becomes increasingly ineffective as the number of processors approaches the number of prisms. It also means that when the second preconditioner is used, the results (unless the descent is iterated to machine precision) will vary slightly, depending on the number of processor specified.

Post-multiplication

This operation is very easy to run in a parallel fashion. Each processor is assumed to have a copy of the vector **z** in Eq. (3.6). The analysis volumes are then parcelled out among the processors in such a way that the number of matrix elements (interactions) in the $\mathbf{P}_b\mathbf{H}^T$ matrix is roughly the same for each processor. A load-balancing algorithm is used here to parcel out the work evenly among processors.

This part of the algorithm runs completely in parallel, with the load divided up as described above. No messages have to be passed until the post-multiplication is completed. At that time, the elements of the correction vector $\mathbf{x}_a - \mathbf{x}_b$ (actually the projection of the correction vector on the vertical eigenvectors) are scattered across the processors and must be reassembled for output and postprocessing.

Granularity

An important issue on many machines is fitting operations into the cache on each processor for greater speed. In many MPI codes, this is controlled by the granularity, which is a measure of the size of blocks, of executable code (large blocks–coarse grain; small blocks–fine grain). Granularity is controlled in the solver and in the post-multiplier by specifying the maximum number of observations in each prism (in the observation sorter, which is run before the 3DVAR code itself). Specifying this number to be small gives a greater chance of the innermost loops running in cache.

Appendix D

A Brief Guide to the NAVDAS Code

This appendix contains subroutine references to the text. Subroutine name and relevant document sections are indicated in bold or italics. The main routine is discussed first; subroutines are listed in alphabetical order for each section.

A. Background and observation ingest, formation of innovations, quality control, thinning, and sorting.

Not listed here.

B. NAVDAS - From innovation vector to vertical eigenvector projections of correction vectors

navdas_driver - sets up dimensions, work space, sets parameters.

Calls **navdas** and **navdas_aer**

---

navdas - reads in observations, sorts into observation prisms, rotates winds. Calls **n3dvar**. Gathers vertically decomposed correction fields onto a single processor, rotates wind correction, processes residual vector. Writes out vertically decomposed correction field to disk.

---

n3dvar - this routine contains the solver and post-multiplication

Input is the sorted, quality-controlled innovation vector, with one entry for each observation. This vector is accompanied by parallel vectors that contain ancillary information such as three-dimensional location of the observation, specified observation error, type of instrument, type of variable, plus some information on the background field necessary for either direct assimilation purposes for nonstandard instruments *(Section 5.1)* or for transformation to isentropic coordinates *(Section 4.5)*

Output is a vector of vertically decomposed corrections (analysis increments) for each analysis grid point and each analysis variable (geopotential, temperature, wind components, moisture). The number of retained vertical modes **nveigout** is less than or equal to the number of levels **levcor**. *(Section 4.8)*

---

anal_sort - this routine sorts the analysis gridpoints into volumes. All that is needed is the latitudes and longitudes of each of the girdpoints (in any order). Thus, it will work for any arbitrary analysis grid (global, regional nested, even nonconnected subsets). The actual sorting is done by the routine **global_sort**. (*Section 3.8)*

Input - lists of latitudes and longitudes for each gridpoint

Output - volumes containing neighbouring grid points.

Calls - global_sort, reshufflei , imov

Called by - n3dvar

---

buddy_decision - performs the buddy check decision on the observations. *(Section 9.3.3)* Called from **genince** after iteration number **iteration_buddy.** This routine calculates the buddy check metric for each observation, rejects offending observations that violate the criteria, and sets a flag **num_reject** for each rejected observation.

Input - innovation vector **xiv_ob** and correction vector **cob**

Output - the rejection flag vector **num_reject** is modified

Called from - genince

---

buddy_action - performs the buddy check action on the observations. *(Section 9.3.5)* Called from **genince** after iteration number **iteration_buddy.** It follows a call to **buddy_decision.**

Input - main diagonal blocks of $\mathbf{HP_bH^T + R}$ matrix.

Output - main diagonal blocks of the $\mathbf{HP_bH^T + R}$ matrix are not changed, but the jth column of the $\mathbf{S_j}$ matrix of *Section 9.3.5* is produced for each buddy-check-rejected observation.

Calls - triangularsort, buddymult, spotrs, fill

Called from - genince

---

buddymult - used for matrix multiplication (**idir** = +1) and solving the linear system

(**idir** = -1) using the S_j matrices of *Section 9.3.5.* This is done to avoid the expensive operation of recalculating the Choleski matrices of the main diagonal blocks after rejecting observations in the buddy check.

Input - vector of length of the number of observations in the prism.

Output - vector of the same length and stored in the same location, depends on the sign of **idir**

Called from - choleski , buddy_action , matrix_mult

---

central_point - used to determine the horizontal centroid of a grid box or an innovation/observation prism. This is required to determine if two observation prisms or an observation prism and an analysis box are too distant to interact. (i.e. with Great Circle distances greater than **obdismx** or **andismx**). Algorithm is discussed in *Appendix A.*

Input - latitude locations **rlat** and longitude locations **rlon** for the **npoint** observation or grid locations

Output - centroid location **rcenlat, rcenlon**

Called - from **n3dvar** once for each observation prism and analysis grid box

---

choleski - performs the solution of the block diagonal problems in the pre-conditioner for each diagonal block on a given processor. The lower triangular Choleski matrices have already been calculated and stored.

Input - the array **w** contains the lower triangular matrices, and **rm1** contains the input vector. **bcolumn** contains information required to correct for observations that have been rejected by the buddy check

Output - the solution vector **zm1**

Calls - work_len, triangularsort, mov, spotrs, buddymult, vchlsk, chslv

Called by - genince

---

coupscl - the horizontal length scale L_h and the geostrophic coupling parameter m of the background error correlation are allowed to vary horizontally. *(Section 4.6)* In calculating the correlations between two observation locations or an observation location and an analysis grid location for these two variables, we take the product of the square roots of the horizontal length scale or coupling parameter at each of the two locations. The case where the coupling parameter changes sign across the equator must be handled.

Input - information on horizontal length scales and coupling parameters.

Output - **elfvect** - a vector of reciprocal square roots of horizontal length scales and **xmuvect** - a vector of square roots of the coupling coefficient.

Called - by **n3dvar** for each observation/innovation prism and each analysis grid volume

---

diag_block - calculates a diagonal block for the preconditioner of background error correlations. Adds in the observation error covariance. Then, does a Choleski decomposition to produce a lower triangular matrix. *(Section 3.4)*

Input - sets of information about each observation in the prism (locations, coupling parameter, vertical information, appropriate Jacobian matrices for directly assimilated variables, etc.

Output - a square matrix of background error correlations

Calls - work_len, horiz_cor, fcovar_spe

Called by - n3dvar

---

eig_matrix_mult - does matrix vector multiplication in vertical eigenvector space of the background error correlation between all the observation prisms. Matrix symmetry is accounted for. **NOTE:** this routine calls MPI routines. The load balancing is performed by a self-scheduling algorithm.

Input - vector in eigenvector space, many fields required to calculate the correlations

Output - another vector in eigenvector space

Calls - horiz_cor, work_len, mov, fcovar_spe , matvece, trans_matvece, mpi_vector_sum, MPI_SEND , MPI_RECV

Called by - matrix_mult , nonlinear_ob

---

fcovar_spe - subroutine to calculate the matrix of background error correlations between observation locations for use in the solver. *(Section 3.2.1)* The calculation is done in vertical eigenvector space and involves seven types of interactions between profiles, soundings and single level observations for deep and shallow vertical modes. *(Sections 4.4 and 5.1)* This calculation is done for each pair of observation prisms that are separated horizontally by less than **obdismx**

Input - for each of the two observation/innovation prisms, complete information on observation locations, variables, local horizontal correlation lengths, geostrophic coupling parameters, etc. is required.

Output - seven matrices **ematl, emat2, smat , sematl, semat2 , esmatl, esmat2.**

Calls - none

Called by - diag_block and **eig_matrix_mult**

---

forevar - calculates for a prism of observation/innovations, the square root of the background error variance **rmsvar** at those observation/innovation location. This routine handles only the standard observations. *(Section 4.1)*

Input - for **nelem** observation/innovation locations the vertical index **nz_ob** and variable type **jvartype_ob.** Also vertical variations of each background error rms variances **vertforvar** for **numvar** variables (except where they are handled by **forevar_direct)**

Output - nelem values of **rmsvar** (except where they are to be handled by **forevar_direct**)

Called - for each observation/innovation prism from **n3dvar**

---

forevar_direct - similar to **forevar** but for direct assimilated nonstandard observations where we must calculate $S_h^{1/2}$ of Eq. (3.17) in S*ection 3.8.* (See also S*ections 5.3* and *Appendix G.)* Searches all **nelem** observation locations in an observation/instrument prism but only acts on directly assimilated observations, which are indicated by instrument type **insty_ob**.

Input - **nelem** values of **jvartype_ob** (variable type), **insty_ob** , plus specific information required for each type of directly assimilated observation, plus vertical variations of background error rms variances **vertforvar** for **numvar** variables.

Output - **nelem** values of **rmsvar** (except where they are handled by **forevar**)

Called - from **n3dvar** for each innovation/observation prism immediately following **rmsvar**

— —

gcirc_sp - produces Geat Circle values *(Appendix A)* and angles *(Appendix B)* from latitude and longitudes of two horizontal locations. This routine can only be run after some precalculations. Handles **nm** points

Input - various quantities that have been previously calculated for each of the two locations separately

Output - **nm** Great Circle distances **s_nm** and cosines and sines **cos_nm** and **sin_nm**

Calls - asin, sqrt, amin2

Called from -from **horiz_cor** and **post_multiply**

— —

genince - this routine performs the solve *(Sections 3.2.1* and *3.2.6)*. For multiprocessor applications, it runs across processors and contains some message passing in internal subroutines

Input - blocks - both diagonal and off-diagonal of $\mathbf{HP}_b\mathbf{H}^T + \mathbf{R}$ matrix plus innovation vector **xiv_ob**.

Output - **cob** vector (**z** in Eq. (3.5))

Calls - mov, fill, triangularsort, spotrs, vchlsk, chlsk, mpi_obvect_bcast, buddycheck, mpi_obvect_bcast, matrix_mult, scalar_vector

Called from - n3dvar

— —

geograph_var_set - sets up latitude/longitude array of factors to multiply the rms background error variance. This array is a function of season and GMT time of day and is used as input to the function **geograph_var**

Input - GMT time of day and month

Output - an array **horiz**

Called by - n3dvar

— —

global_sort - this routine takes a sequence of latitude/longitude locations corresponding to observations and sorts them into observation prisms. The observations can be located anywhere on the sphere, and the routine works independently of any grid. It is primarily use in the pre-processing, but is called in **n3dvar** if a second preconditioner is required (using the observation re-sorting method of S*ection 3.5*).

Input - vectors of latitudes (**rlat_ob**) and longitudes (**rlon_ob**) for each of **num_ob** observations. Also requires information on whether observations are in a vertical column (profiles or soundings) and how many observations are to be permitted in a prism.

Output - a vector **nbox_ob** that gives the prism number for each observation

Called by - re_sort, latlon_sequencer

---

horcor_s1, horcor_s2, horcor_pm1, horcor_pm2 - calculates horizontal background error correlations. *(Sections 4.6, 4.7, Appendix B)* **horcor_s1** is for the nondivergent calculations in the solver, **horcor_s2** is for the divergent calculations in the solver, **horcor_pm1** is for the nondivergent calculations in the post-multiplier, and **horcor_pm2** is for the divergent calculations in the post-multiplier.

Input - Great Circle distances **s_nm**, correlations and radial derivatives **f,fp,fpp,** and angle information **sin_nm, cos_nm** for **nm** locations and **numvar** variables in their interaction with another location. There is also information about local horizontal correlation scale and geostrophic coupling.

Output - **hcor(nm,numvar)** which contains the <ZZ> , <Zu> , <Zv>, <uu>, etc. horizontal correlations

horcor_s1 and **horcor_s2** are called by **horiz_cor**, while **horcor_pm1** and **horcor_pm2** are called by **horizontal**

---

horiz_cor - calculates all horizontal correlations used in solver. It is called prior to **fcovar_spe** and provides a complete two-dimensional array. It has some similarities with **horizontal**, but it includes the Great Circle calculation and produces a two-dimensional rather than a one-dimensional output array.

Input - two sets of observation locations and ancillairy information such as geostrophic coupling parameters, horizontal length scales, etc.

Output – two-dimensional array of horizontal correlations between the two sets of observations.

Calls - gcirc_sp, hormod and **horcor_sp1**

Called by - diag_block, eig_matrix_mult

---

horizontal - performs horizontal interactions in calculating background error correlation matrix elements in post-multiplier. (S*ection 4.6)* Can do either direct calculation or table look-up

Input - values of Great Circle distance, radial derivatives, angles, local horizontal correlation scales, geoostrophic coupling, etc. between a single location and **mumtot** other locations.

Output - **mumtot** values of horizontal correlations

Calls - fill , hormod , horcor_sp

Called by - post_multiply

---

hormod - computes the values of various specified horizontal covariance models and their first and second radial derivatives. *(Section 4.6.2)*

Input - nm normalized (by horizontal length scale $\mathbf{L}_h$) values of Great Circle distance **s**

Output - **nm** values of the correlation function **f**, and associated first and second radial derivatives **fp** and **fpp**.

Called by - horizontal

---

latlon_sequencer - sort the observations into observation prisms

Input - unsorted observation innovations and auxilliary observation information

Output - sorted innovations and auxiliary information, plus pointers marking the beginning and end of each observation prism.

Calls - global_sort, mov, reshuffler, reshufflei, reshufflech10, reshufflecf16

---

leftop_analob - this routine takes vertically decomposed corrections defined at observation locations and produces real space corrections in the analyzed variables (u,v,T etc.) at the observation locations. It is used in the outer iteration of the solver for observations with a nonlinear forward operator such as SSM/I windspeed. It is a modified version of the routine **leftoperator**. *(Section 6.1)*

Input - vertically decomposed corrections **ecob**

Output - real space corrections at the observation locations and in the appropriate analyzed variables

Called by - nonlinear_ob

---

leftoperator - opposite of **rightoperator**. **E** operation in Eq. (4.16) or $\underline{\mathbf{H}}$ operator in *Section 5.3* for direct assimilation of nonstandard observations.

Input - vectors **ecob1** for deep modes, **ecob2** for shallow modes, and **ecobsing** for single level observations of vertically projected observation/innovations

Output - vector **xiv_ob** in physical space

Calls - sdot

Called from - ebediag and **matrix_mult**

---

loadbalance_an - routine required to balance the load across multiprocessors for the post-multiplier *(Section 3.2.2)* when more than one processor is used. It does not contain any MPI calls. (*Appendix C)*

Input - information about the observation prisms and the analysis grid volumes

Output - a re-ordering of analysis grid volumes

Called from - **n3dvar** at the beginning of the post-multiplication step

---

matrix_mult - performs a complete matrix vector multiplication of the form $\mathbf{z} = [\mathbf{HP}_b\mathbf{H}^T\mathbf{+R}]\mathbf{x}$. The background error correlations use vertical eigenvector decomposition, and observation error correlation is calculated directly.

Input - the vector **x** (called "pv" in the code). All the information required to calculate the background and observation error correlations.

Output - the **z** vector (called "qvc" in the code)

Calls - rightoperator, leftoperator, mpi_obvect_bcast, fill, triangularsort, mov, strmv, multmv, work_len, eig_matrix_mult

Called from - genince

---

matvece - does matrix multiplication $\mathbf{Ap}_k$ in solver (Eq. (3.13)) for preconditioned conjugate gradient algorithm for off-diagonal blocks. This calculation is done in vertical eigenvector space except for single-level/single-level interactions. *(Section 4.4.2)*

Input - vectors **exiv_ob1** for deep modes, **exiv_ob2** for shallow modes, and **exiv_obsing** for single-level observations for an observation/innovation prism; also seven matrices **emat1, emat2, smat, esmat1, esmat2, semat1, semat2** for the interactions between two observation/innovation prisms.

Output - vectors **ecob1** for deep modes, **ecob2** for shallow modes, and **ecobsing** for single-level observations for the second observation prism.

Calls - multmv

Called by - ebediag, eig_matrix_mult

---

nonlinear_ob - subroutine called at the end of a nonlinear iteration to calculate new values of the analyzed variables (u,v,T etc) at the observation locations. These are to be used in recalculating the TLMs for directly assimilated variables. This routineis similar to **matrix_mult**, except that the input is in vertical eigenvector space and the output is in real space, but for the analyzed rather than the observed variables. At this time the only nonlinear operator is SSM/I windspeed. *(Section 6.2)*

Input - **ecob**, which is the vertically decomposed version of **cob**, after convergence of the linear solver.

Output - correction values at observation locations for appropriate analysis variables. For example, for SSM/I windspeed, it would be (u,v) corrections at the same location.

Calls - leftop_anal, work_len, eig_matrix_mult

Called by - n3dvar

---

obcovar - calculates observation error covariance (normalized by background error variances), that is, $\mathbf{S}_h^{-1/2}\mathbf{R}\mathbf{S}_h^{-1/2}$ in Eq. (3.17) of *Section 3.8*. This term is then added to the background error correlation $\mathbf{C}_h^{ob/ob}$ in Eq. (3.17). This is only performed for diagonal blocks.

Input - The **nelem** by **nelem** matrix **xmblok** ($\mathbf{C}_h^{ob/ob}$) and vector **obnorm** ($\mathbf{S}_h^{1/2}$) of length **nelem** plus the diagonal observation error matrix **err_ob**

Output - **xmblok** - a diagonal block of $\mathbf{C}_h^{ob/ob} + \mathbf{R}$

Called by - diag_block

---

post_multiply - performs the post multiplication. *(Section 3.2.2)* Both the assembling of the matrix elements of a single block of the $\mathbf{P}_b\mathbf{H}^T$ matrix (for a single observation prism and a single analysis grid volume) and the multiplication by the appropriate elements of the **z** vector of Eq. (3.6) are included in this routine. The actual matrix is never completely formed and the algorithm takes more of an operator approach. This operation is performed for analysis grid boxes and observation prisms that are separated by a Great Circle distance, which is less than **andismx**.

Input is three vectors **ecob1** and **ecob2**, corresponding to profile/sounding observations, and **ecobsing,** corresponding to single-level observations, that have been previously vertically projected. Also information regarding the positions and characteristics of both the analysis grid volume and the elements of the observation prism vector. This operation is done for each analysis grid box and observation prism that are not separated by a Great Circle distance of more than **andismx**

Output is a vector of corrections **outvect** for the **num_b** analysis grid points and **numvar** variables for the **nveigout** gravest vertical eigenvectors.

Calls - gcir_sp, horizontal , multmv

Called from - n3dvar

---

precipwattlm - this routine is the actual TLM for generating precipitable water from log(specific humidity). *(Section 5.7)*

Input - log(specific) humidity profile **vector**, standard atmosphere specific humidty **qzero**, vector of pressures **prescor**.

Output - precipitable water **pwlin**

Called by - ssmi_pw_eigen

---

re_sort - subroutine for re-sorting the observations for use in creating a second preconditioner for the descent algorithm. *(Section 3.5)*

Input - vectors of latitudes and longitudes for the observations

Output - for each of the observations in the re-sort, its location in the original sort

Calls - global_sort, mov, reshufflei

Called by - n3dvar

---

rightoperator - goes from physical space to vertical eigenvector space. $\mathbf{E}^T$ operator in Eq. (4.16) in *Section 4.4.2* or the $\underline{\mathbf{H}}^T$ operator in S*ection 5.3* for direct assimilation of nonstandard observations. It is never called across processors, only within a processor.

Input - vector **xiv_ob** of innovations in physical space

output - vectors **ewob1** for deep modes and **ewob2** for shallow modes in vertical eigenvector space. **ewobsing** for single-level observations remains in physical space.

Calls - fill

Called by - n3dvar, ebediag, matrix_mult

---

shuffle_choleski - this routine is called just before and just after the call to **choleski** when the second preconditioner (S*ection 3.5*) is being applied. In the forward direction (**idir** = **+1**), it re-sorts the input vector, and in the reverse direction (**idir=-1**) it re-sorts the output from the choleski operation.

Input - vector to be re-sorted

Output - the re-sorted vector

Calls - mov

Called by - genince

---

ssmi_pw_eigen - calculates the (universal) TLM for the SSM/I precipitable water for assimilating normalized precipitable water into the variable used in the assimilation - log(specific humidity) and projecting onto the background error vertical eigenvectors. *(Section 5.6)*

Input - the vertical eigenvector matrix **forevect** and eigenvalue **unieval**, background error variance **forvarcon,** and standard atmosphere specific humidty profile **qzero**

Output - **pweig** the TLM and **forvar_ssmi_pw** the background (normalized) precipitable water.

Calls - the precipitable water TLM **precipwattlm**

Called from - n3dvar

— —

ssmi_windspeed - this routine produces the TLM field for the SSM/I windspeed. *(Section 5.5)* This version can also handle the nonlinear iteration. *(Section 6.2)*. Thus, the first (linear) iteration differs from the rest.

Input - Two background wind components for each SSM/I winspeed observation.

Output - the 2×1 TLM matrix for each SSM/I windspeed observation

Called from - n3dvar

— —

storebalance_ob - this routine distributes the diagonal blocks of the $\mathbf{HP_bH^T + R}$ matrix among the processors in an equitable way. It does this while trying to maintain, as much as possible, adjacent observation prisms on the same processor (to improve the effectiveness of the second preconditioner). Note – this storage allocation is not a load balancer for the matrix/vector multiply in the solver (which uses a self-scheduling algorithm and does not require an a priori load balancing calculation). *(Appendix C)*

Input - sizes of the observation prisms

Output – array **indexproc** that allocates the prisms to each processor

Called by - n3dvar

— —

tablook - subroutine to create table look-ups of horizontal correlation models *(Section 4.6.2)* and the arcsine required in the calculation of Great Circle distance. *(Appendix A)*

Input - number of tables **numtab**, number of values in each table **lookup,** and the choice of correlation model **icormod**

Output - **lookup** by **numtab** tabulated values

Called - near the beginning of **n3dvar**

— —

trans_matvece - like **matvece** except for the transpose of the matrix. Called only if matrix symmetry is to be exploited.

Input/Output - similar to **matvece**

Calls - transp, multmv

Called by - eig_matrix_mult

---

tovs_eigen - routine that is called for every TOVS sounding to produce the $\underline{\mathbf{H}}$ matrix of *Section 5.3.3.* It also produces the background error variance in temperature brightness space ($S_h^{1/2}$ in *Section 5.3.3).*

Input - the jacobian matrix **H** that is different for every sounding vertical background eigenvector matrix **forevectt.**

Output - the matrix **tovs_eig** that is the number of channels **nchannel** by the number of eigenvalues **nveigsound**. It also produces a vector **forvar_tovs** of length **nchannel** of background brightness temperature rms error variances.

Calls – press_interp

Called by - **n3dvar**

---

verticalconst - calculates the vertical variation of various universal quantities - the pressure at temperature/ moisture levels **prestq**, the vertical variation of the background error variance for each variable **vertforvar**, the vertical variation of background error correlation vertical scales for each variable **vscale** and the standard atmosphere specific humidity **qzero**.

Input - pressure level **prescor** at **levcor** levels of the geopotential and wind correlations.

Output - **prestq**, **vertforvar, vscale**, and **qzero** at **levcor** levels

Called at the beginning of **n3dvar** (before **vforcove**)

---

vertintegral - routine to calculate vertical geopotential background error variance from specified vertical temperature background error variance using the hydrostatic relation.

Input - vectors of pressure and temperature error variances

Output - vector of geopotential error variances

Calls - hydroth

Called by - verticalconst

---

verteigset - this routine is called once for each observation/innovation prism. It is a basic set-up routine for the vertical eigenvector decomposition. *(Section 4.4)* Information such as horizontal location and variable type is required in the operators **leftoperator** and **rightoperator**, which go back and forth between physical and eigenvector space. This routine takes **nelem** standard latitude **(rlat_ob)** and longitude **(rlon_ob)** observation locations and variable types **(jvartype_ob)** and reorders them for the eigenvector decomposition into **elat_ob, elon_ob,** and **jevartype_ob**, respectively. It also calculates maps for the re-ordering **nmapprof, nmapsound,** and **nmapsing** (corresponding to profiles, sounder, or single-level observations) of the innovations themselves, although the innovations are **not** reordered in this routine.

Input - jvartype_ob, rlat_ob, rlon_ob

Output - **jevartype_ob, elat_ob, elon_ob, nmapprof, nmapsound, nmapsing**

Called from - **3dvar** once for each innovation/observation prism

---

vforcove - calculates the vertical background error correlation matrices **forvcor** for multivariate variables and **forvcorunivar** for univariate variables, the corresponding vertical eigenvector matrices **evect** and transpose **evectt**, the eigenvalue matices **foreval** for the multivarate correlations and eigenvalue vectors **univeval** for the univariate correlations. *(Section 4.3)*

Input - **levcor** pressure levels **prescor** and **prestq**, background error variances for the wind **winderr**, background error vertical scales **vscale**.

Output - **evect, evectt, foreval, univeval , forvcor, forvcorunivar** plus temperature **temperr** and geopotential **heiterr** background error variances. All vectors are length **levcor,** and matrices **evect, evectt, forvcor, and forvcorunivar** are **levcor** by **levcor**.

Calls - eigen, hydrot and utitilities **vertcor_rspace, transp, mult3, fill,** and **mov**

Called - near the beginning of **n3dvar**, but after **verticalconst**

---

vindex - calculates the vertical index of any observation/innovation based on its pressure (**prescor** for heights/ winds and **prestq** for temperature/moisture). *(Section 4.3.2)*

Input - **num** observation pressures **p_ob** , vectors of length **levcor** of correlation pressure levels **prescor** and **prestq**

Output - **num** indices **nzindex**

Called by - **n3dvar** for each observation/innovation prism.

C. Interface with COAMPS and NOGAPS.

Not discussed here

D. Utilities

add_field - adds two fields together

bctdg1,bctdg2,bctdx1 - Gaussian elimination called in **bicube**

bicinit - called in **bicube**

bicube - bicubic spline interpolator

cal_jmin - calculates the J_{min} diagnostic*(Section 9.1)*

chslv - FORTRAN version of second part of Cholesky decomposition - back substitution.

chlen - fills file with sequential integers

delta - function used to calculate the weight function of Eq. (4.6)

dotproduct - performs a dot product across processors and transmits the result to all processors.

eigen - calculates eigenvectors and eigenvalues of a symmetric real matrix

fill - fills field with a constant

gcircd - function that calculates Great Circle distance between any two points

geograph_var - function calculates the background error variance at any latitude, longitude point

grid - to calculate latitude, longitude, Coriolis force, etc., from grid locations

ghost_ob - provides ghost observations (with large observation errors) that are necessary for low observation counts

hydrot - produces a vertical temperature or thickness vector from a vector of geopotentials

hydroth - produces a geopotential vector from a temperature of thickness vector by integrating up from the surface.

ifill - integer fill

ij2ll - calculates latitude, longitude at specified points on a grid

ij_ll - same as **ij2ll**

imov - integer move

multmv - does matrix vector multiply using either BLAS or FORTRAN

mult3 - multiplies together two conformable rectangular matrices

mov - moves one array into another

outmtx - prints a (small) matrix

press_interp - creates matrix that interpolates from one set of pressure levels to another

print_structure - outputs vertical, latitudinal, or vertical/latitudinal cross-sections of various variances and scales connected with background error covariance by calling the actual routines used in the assimilation algorithm (**coupscl** and **forevar**)

reshuffler, reshufflei, reshufflech10, reshufflech16 - reshuffles a (real, integer, c10, or c16) field following a call to **global_sort** (the global sorter routine)

scalar_vector - performs operation r = a*s + b*t , where a,b are scalars and r,s,t are vectors of the same length.

scale_add - adds a constant to a field

scale_field - multiplies a field by constant

stancg - does a standard conjugate gradient solve *(Section 3.2.5)*

triangularsort - converts symmetric matrix into upper triangle or vice versa

trianglemult - multiplies two triangular matrices together (undoes a Cholesky decomposition)

transp - transposes a rectangular matrix

tune_const - sets up constants for background error covariances

vchlsk - FORTRAN version of first routine of Choleski decomposition, that is, the triangular decomposition

vertcor_rspace - calculates real space form of vertical background eigenvectors from their eigenvectors and eigenvalues

xlfil - puts transpose of lower triangle into upper triangle - called from **vchlsk**

windrotate_grid - rotates grid windfields from grid orientation to spherical orientation and vice versa

windrotate_ob - same as **windrotate_grid**, except for innovations

work_len - compares available work space with space required and sends a message and terminates execution if insufficient space is available

E. BLAS (Basic Linear Algebra System) subroutines

sdot - scalar dot product

sgemm - matrix/matrix multiply

sgemv - matrix/vector multiply

spotrf - first part of Cholesky decomposition (triangular decomposition)

spotrs - second part of Cholesky decomposition (back substitution)

strmv - multiplication of a vector by a lower or upper triangular matrix

strsv - solution of a lower or upper triangular system of equations

F. MPI routines and subroutines containing MPI calls

In addition to the following routines, the routines **eig_matrix_mult** and **anal_err** discussed above also contain MPI calls.

mpi_begend - subroutine called at the beginning and end of **n3dvar** to initiate and end MPI action

Input - processor number **myid** number of processors **numprocs.** Indicator **itype** whether beginning or end

Output - none

Calls - MPI_INIT, MPI_COMM_RANK, MPI_COMM_SIZE, MPI_FINALIZE

Called at beginning and end of driver program for **n3dvar**

---

mpi_obvect_bcast - transfers portions of a vector from each processor to all the other processors in order to assemble a complete copy of the vector on all processors. This vector may or may not have been previously vertically decomposed. **Note** - this routine is the most rudimentary way of performing this operation and could be done much more elegantly using more advanced MPI constructions.

Input - information about slave and master processors, identity of observation/innovation prisms located in each processor, and vector of values to be transferred

Output - each processor will have a complete copy of the innovation/observation vector

Calls - mov, MPI_SEND, MPI_RECV, MPI_BCAST

Called from - once during each iteration of the solver - **genince,** once just prior to the post-multiplication - **n3dvar**

---

mpi_obvect_bcast_int - exactly the same as **mpi_obvect_bcast** except for vectors of integers

Called - during the buddy check inside the solver **genince** just once

---

mpi_correction - subroutine called at the end of **n3dvar** to transfer portions of the (vertically projected) correction vector from each of the slave processors to the master processor

Input - details of analysis grid, a portion of the (vertically projected) correction field stored in **correction_eigen**

Output - complete grid (**numvar** variables, **num_b** analysis volumes, **nveigout** vertical modes) of the correction field stored in **correction_eigen**

Calls - MPI_SEND, MPI_RECV

Called - at the end of **n3dvar**

— —

mpi_timer - timer for MPI runs

Input - none

Output - time elapsed since last call

Calls - MPI_WTIME, MPI_BARRIER

Called - when desired

— —

mpi_vector_sum - adds vectors (of the same length) from each processor together and distributes the resulting sum vector to each of the processors. If the vector length is 1, this routine performs a scalar product across processors. It is called four times (per iteration) in the preconditioned conjugate gradient algorithm (with length 1) to perform scalar products. It is also called once per iteration (if the matrix symmetry is to be exploited) with a vector length, which is the order of the number of observations. *(Appendix C* and *Section 3.2.6)*

Input - vector **a** on each processor and vector length

Output - vector **a** (the summed vector) on each processor

Calls - MPI_ALLREDUCE

Called by - genince

— —

mpi_loadcalc - subroutine called in load-balancing precalculation to pass timings between processors

Input - small vectors or timings for each processor

Output - vectors of timings known on all processors

Calls - MPI_SEND, MPI_RECV, MPI_BCAST

Called by - n3dvar

— —

G. Adjoint routines

These routines are used in calculating the adjoint of the 3DVAR code. *(Section 9.4)*

adjoint_correction - this is the adjoint of the subroutine **correction**. This subroutine takes grid point representations of the analysis sensitivity vector, normalizes them using the background error variances and vertically decomposes them using the eigenvectors of the background error correlation but still remaining on the horizontal analysis grid.

Input - vectors of analysis grid point values for each of the output variables

Output - a vector of vertically decomposed variables on horizontal gridpoints

Calls - outputvindex and **forevar**

Called by - adjoint_threed

---

adjoint_n3dvar - this is the adjoint of the subroutine **n3dvar**. It is the reverse of **3dvar**. Its input is vertically decomposed analysis sensitivity vectors at analysis grid points. It then runs these through the adjoint of the post-multiplier, that is, **adjoint_post_multiply**, which produces vertically decomposed sensitivity vectors in observation space. After operation by **leftoperator**, these sensitivity vectors are in real observations space. It remains to operate on this vector with the solver **genince** (which is symmetric and self-adjoint) producing an observation sensitivity vector.

Input - a vector of vertically decomposed analysis sensitivity values on the horizontal analysis grid.

Output - the observation sensitivity vector

Calls - the same routines as **n3dvar**, except **adjoint_post_multiply** instead of **post-multiply**

Called by - adjoint_threed

---

adjoint_post_multiply - this is the adjoint of **post_multiply**. It takes a vertically decomposed analysis sensitivity vector and projects it into vertically decomposed observation space.

Input - a vertically decomposed analysis sensitivity vector (**adjoint_eigen_grid**) and many arrays required to calculate the background error correlation

Output - a vertically decomposed vector in observation space (**ecob1, ecob2, ecobsing**)

Calls - same subroutines as **post_multiply**

Called by - adjoint_n3dvar

H. Analysis error subroutines

These routines are used in estimating the analysis error variance, following *Section 10.*

As noted in *Section 10,* this code is embarrassingly parallel. However, load balancing is quite difficult because it is very hard to estimate loads a priori. The solution adopted is a "self-scheduling" MPI algorithm that dynamically distributes the loads. The price paid is that one processor is used only for accounting and storing results.

navdas_aerr - the equivalent of **navdas** for this problem. Called by **navdas_driver**. Calls **anal_err**

anal_err - this is the equivalent of **n3dvar** for this problem. The input and most of the first part of the processing (up to the construction of the Choleski matrix) is the same as **n3dvar.** After that, we remain in the same loop over the diagonal blocks until the end of the subroutine. It is at this point that the additional code is added. **Note**: this routine contains some MPI calls.

Input - vectors with all the innovation information. The innovations themselves are not used, but all the rest of the information related to the innovations is used.

Output - a vector of analysis error gridpoint values stored in **analysis_error**

Calls - many of the same routines as **n3dvar** (not **genince**, though). Also calls **correction_anal, backvar,** and **grid_assign**

– –

backvar - calculates background error variances at gridpoint locations

Input - information on grid

Output - values of background error variances for **numvar** variables at each analysis gridpoint.

Calls - outputvindex, forevar

Called by - anal_err

– –

correction_anal - transforms from vertical eigenvector space (**nveigout** modes) to vertical grid space (**lm** levels). This is essentially a matrix multiplication, but since it is used many times with the same matrix, but many right-hand-side vectors, the elements of the matrix are only calculated the first time through and stored in the matrix **vert.**

Input - analysis grid information, vertical interpolation information. Vertical eigenvector coefficients.

Output - information in vertical grid space that has not been multiplied (denormalized) by background error variances. In this way it differs from **correction_eigen.**

Calls - outputvindex

Called by - anal_err

grid_assign - routine that for each analysis gridpoint determines the closest analysis prism (as indicated by the Great Circle distance to the prism centroid). Thus, each analysis grid point is associated with a particular observation prism. It also produces arrays **ja_pos, ia_pos** for each prism that contain the true j,i locations for the points associated with that prism.

Input - locations of the prism centroids (**rcenlat, rcenlon**) and latitude and longitude of each of the analysis grid points

Output - vectors **ia_pos, ja_pos** for each prism that contain true locations

Calls - gcircd

Called by - anal_err

I. Two-dimensional univariate analysis system

This code is a version of the 3DVAR code designed for the two-dimensional (horizontal) univariate analysis of variables such as the mean sea level pressure. It is essentially a stripped-down version of the 3DVAR code, but uses the same sort of observation prism sorting, preconditioned conjugate gradient descent, and the same conventions regarding the input innovation stream. It uses a subset of the subroutines of the 3DVAR code plus the following modified versions of 3DVAR routines.

n2dvar_univ - modified version of **n3dvar**

block_2duniv - modified version of **diag_block**

genince_2duni - modified version of **genince**

matrix_mult_2duni - modified version of **matrix_mult**

J. Ensemble/singular vector hybrid form of NAVDAS

This is a version of NAVDAS that calculates the background error covariances as a linear combination of the standard statistical covariances with a second error covariance obtained by direct projection onto singular vectors or ensemble members. Most routines are in the standard NAVDAS code, but there are a few special routines.

n3dvar_ens - modified version of **n3dvar**

emat_ens - this routine has no counterpart in **n3dvar** and calculates a matrix that relates the observation space to the singular vector space. It may include the effect of the forward instrument operator (if there is one). It is used in both the solver and the post-multiplier.

genince_ens - modified version of **genince**

matrix_mult_ens - modified version of **matrix_mult**

ssmi_pw_ens - modified version of **ssmi_pw_eig**

tovs_eigen_ens - modified version of **tovs_eigen**

K. De-aliasing scatterometer wind vector observations

This code performs de-aliasing of scatterometer wind observations following the ideas of *Appendix H*. Most of the subroutines come from the 3DVAR code, but there are two special subroutines.

de_alias - contains most of the special code of Appendix H and calls **block_2dmult**

block_2dmult - version of **diag_block** appropriate for the two-dimensional bivariate (wind) case.

L. Two-dimensional multivariate analysis system

This code is a two-dimensional version of NAVDAS that is designed to handle any multivariate problem, for example, the analysis of surface wind. It uses most of the NAVDAS codes, except for the following.

n2dvar_multvar - modified version of **n3dvar**

post_multiply_2dmv - modified version of **post_multiply**

Appendix E

A Second Convergence Accelerator—The Triangular Plate Approximation

Section 3.5 introduced a convergence accelerator based on a re-sorting of the observations. We now consider another possible improvement to the basic preconditioner discussed earlier. As noted in Section 2, the condition number for Method C tends to increase as the background error correlation length or the observation density increases. In either case, the matrix $\mathbf{Q}_C$ becomes less diagonally dominant, and any descent algorithm will converge less rapidly than it would under more favorable conditions. We discuss here the development of a different preconditioner than the one described in Eq. (3.14). In developing a preconditioner, compromises must always be made. A preconditioner that is very similar to the original matrix will converge in a very few iterations, but each iteration step may be very expensive. On the other hand, a simple preconditioner (such as the diagonal elements of the matrix), is cheap, but it may take nearly as many iterations as the standard conjugate gradient. The block diagonal approach described in Eq, (3.14) is an effective compromise. The question is, though, can a reasonably inexpensive preconditioner be designed that converges even more rapidly.

The block diagonal preconditioner of (3.14) is a local type approximation and does not work well when the spatial scale of the observation volume is small compared to the background error correlation scale. This happens, either when the background error correlation scale is very large or the observation density is high. It would be nice if a preconditioner could be designed that was global, rather than local, because, in principle, it should converge more quickly in these situations. We now describe such a global preconditioner. The new preconditioner $\mathbf{A}^*$ has to lead to an inexpensive linear solve (when compared to the matrix vector multiplication by the original matrix $\mathbf{A}$ in Eq. (3.13)).

We perform the linear solve for the new $\mathbf{A}^*$ by using a standard conjugate gradient descent (3.12). In effect, there is an outer iterative loop (3.13) for $\mathbf{A}$ and an inner iterative loop (3.12) for $\mathbf{A}^*$. Clearly, any inner iteration (3.12) must be very inexpensive compared to the cost of an outer iteration (3.13). By far the major cost in (3.12) is the matrix vector multiplication ($\mathbf{q}_k = \mathbf{A}^*\mathbf{p}_k$) for this application. So, if the new preconditioner is to be successful, this operation must be very cheap. The essence of this new preconditioner, then, is a very inexpensive multiplication of the form $\mathbf{q} = \mathbf{A}^*\mathbf{p}$, where $\mathbf{A}^*$ is an approximation to $\mathbf{A}$. If there are L observations, then a full matrix vector multiplication would take L^2 operations; the approximation we shall describe is O(L) operations.

The idea is as follows. The domain has already been divided into M triangular volumes, where M is approximately $L^{1/2}$. Within each volume are K_m, $1 \leq m \leq M$ observations. Let us consider the two-dimensional univariate case. Denote (x_i^n, y_i^n) and (x_j^m, y_j^m) as the spatial locations of the ith element of the nth volume and jth element of the mth volume respectively. The diagonal blocks (n = m) of the matrix $\mathbf{A}^*$ are the diagonal blocks of $\mathbf{A}$ (as in the preconditioner described in Eq. (3.14)). What we describe here are the elements of the off-diagonal blocks ($n \neq m$) of $\mathbf{A}^*$. For simplicity, let us assume that the observation error may be correlated within observation volumes, but not between volumes, so the off-diagonal blocks of $\mathbf{A}^*$ contain only background error covariances or correlations. Then, any element of the matrix $\mathbf{A}^*$ for which $n \neq m$ can be written $\rho_{ij}^{nm} = \rho(x_i^n, y_i^n, x_j^m, y_j^m)$, which can represent either a correlation or a covariance. Then, the ith element of the nth volume of the vector $\mathbf{q}$ can be written

$$q_i^n = \sum_{m=1}^{M} \sum_{j=1}^{Km} \rho_{ij}^{nm} \; p_j^m, \qquad \text{(E1)}$$

where p_j^m is the jth element of the mth volume. Now consider the two triangular volumes m and n as illustrated in Fig. E1. The vertices of the triangles are denoted,(x_1^n,y_1^n), (x_2^n,y_2^n), (x_3^n,y_3^n) and (x_1^m,y_1^m), (x_2^m,y_2^m), (x_3^m,y_3^m), and their midpoints are

$$x_n^* = [x_1^n + x_2^n + x_3^n]/3,\ y_n^* = [y_1^n + y_2^n + y_3^n]/3 \text{ and similarly for } x_m^* \text{ and } y_m^* . \qquad \text{(E2)}$$

Then, define $\Delta x_i^n = x_i^n - x_n^*$, $\Delta y_j^m = y_j^m - y_m^*$, etc., as the distances from the midpoints in each triangle. Then, approximate ρ_{ij}^{nm} by the 9-term approximation

$$\rho_{ij}^{nm} = a_m^n + b_m^n\Delta x_i^n + c_m^n\Delta x_j^m + d_m^n\Delta y_i^n + e_m^n\Delta y_j^m + f_m^n\Delta x_i^n\Delta x_j^m + g_m^n\Delta x_i^n\Delta y_j^m + h_m^n\Delta y_i^n\Delta x_j^m + i_m^n\Delta y_i^n\Delta y_j^m. \qquad \text{(E3)}$$

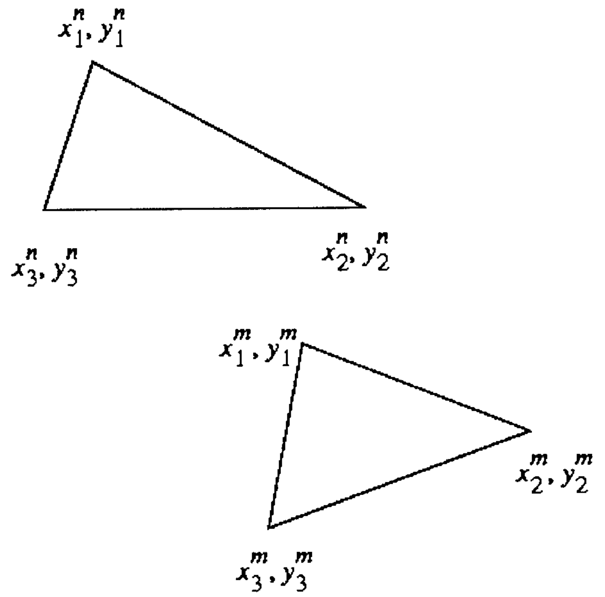


Figure E1
Labeling used for the data partition triangles (prisms)

The a_m^n, etc., are coefficients to be specified shortly. Now, define three operations:

$$(1)\ \alpha_m = \sum_{j=1}^{Km} p_j^m,\ \beta_m = \sum_{j=1}^{Km} \Delta x_j^m p_j^m \text{ and } \gamma_m = \sum_{j=1}^{Km} \Delta y_j^m p_j^m, \qquad \text{(E4)}$$

$$(2)\ r_i^n = \sum_{m=1}^{M} [\alpha_m a_m^n + \beta_m c_m^n + \gamma_m e_m^n] ,$$

$$s_i^n = \sum_{m=1}^{M} [\alpha_m b_m^n + \beta_m f_m^n + \gamma_m g_m^n], \qquad \text{(E5)}$$

$$t_i^n = \sum_{m=1}^{M} [\alpha_m d_m^n + \beta_m h_m^n + \gamma_m i_m^n].$$

$$(3)\ q_i^n = r_i^n + \Delta x_i^n s_i^n + \Delta y_i^n t_i^n. \qquad \text{(E6)}$$

For each observation volume (m), Eq. (E4) performs three sums over all the observations in the volume. This operation involves no interactions between volumes.

The second operation (E5) performs only operations between observation volumes. The final operation (E6), for each observation volume performs operations on all the observations within that volume.

Let us perform a rough operation count. Note that the a_m^n, etc., do not depend on the vector **p**. They depend only on the observation locations and the covariances (or correlations) and thus are independent of the iteration number. Consequently, we can precalculate them and store them. Now, for L observations there are L^2 operations for the complete matrix vector multiply $\mathbf{q} = \mathbf{Ap}$. There are M volumes, and let us suppose there are K observations in each volume, giving the total number of observations as L = KM. Then, the number of operations for (E4) is 3KM, for (E5) it is $9M^2$, and for (E6) it is 3KM. If $M = K = L^{1/2}$, then the total number of operations = 15L. Thus, this algorithm does an (approximate) matrix vector multiply in O(L) operations, as opposed to $O(L^2)$ operations for its full counterpart.

Before proceeding, we must show how to calculate a_m^n, etc. These are obtained by demanding that the correlation (E3) be exact at the vertices of the triangles. Thus, for example, we would specify that

$$\rho_{12}^{nm} = a_m^n + b_m^n\Delta x_1^n + c_m^n\Delta x_2^m + d_m^n\Delta y_1^n + e_m^n\Delta y_2^m + f_m^n\Delta x_1^n\Delta x_2^m + g_m^n\Delta x_1^n\Delta y_2^m + h_m^n\Delta y_1^n\Delta x_2^m + i_m^n\Delta y_1^n\Delta y_2^m, \qquad \text{(E7)}$$

be exact for all (n,m) at each of the triangle vertices. This gives nine values (ρ_{11}, ρ_{21}, ρ_{13}, ρ_{21}, ρ_{22}, ρ_{23}, ρ_{31}, ρ_{32}, and ρ_{33}) for each (n,m). The nine equations of the form (E7) can be inverted for each (n,m) to relate a_m^n, b_m^n, etc., to the ρ_{13}^{nm}, ρ_{22}^{nm}, etc. Thus,

$$a_m^n = [\rho_{11}^{nm} + \rho_{12}^{nm} + \rho_{13}^{nm} + \rho_{21}^{nm} + \rho_{22}^{nm} + \rho_{23}^{nm} + \rho_{31}^{nm} + \rho_{32}^{nm} + \rho_{33}^{nm}] / 9. \qquad \text{(E8)}$$

To derive the remaining coefficients, define

$$S_1^{nm} = \rho_{11}^{nm} + \rho_{12}^{nm} + \rho_{13}^{nm}, \quad S_2^{nm} = \rho_{21}^{nm} + \rho_{22}^{nm} + \rho_{23}^{nm}, \quad S_3^{nm} = \rho_{31}^{nm} + \rho_{32}^{nm} + \rho_{33}^{nm},$$

$$R_1^{nm} = \rho_{11}^{nm} + \rho_{21}^{nm} + \rho_{31}^{nm}, \quad R_2^{nm} = \rho_{12}^{nm} + \rho_{22}^{nm} + \rho_{32}^{nm}, \quad R_3^{nm} = \rho_{13}^{nm} + \rho_{23}^{nm} + \rho_{33}^{nm}.$$

Then define

$$\delta_n = 3[(\Delta y_3^n - \Delta y_2^n)(\Delta x_2^n - \Delta x_1^n) - (\Delta y_2^n - \Delta y_1^n)(\Delta x_3^n - \Delta x_2^n)],$$

$$\delta_m = 3[(\Delta y_3^m - \Delta y_2^m)(\Delta x_2^m - \Delta x_1^m) - (\Delta y_2^m - \Delta y_1^m)(\Delta x_3^m - \Delta x_2^m)].$$

This gives

$$b_m^n = [(\Delta y_3^n - \Delta y_2^n)(S_2^{nm} - S_1^{nm}) - (\Delta y_2^n - \Delta y_1^n)(S_3^{nm} - S_2^{nm})] / \delta_n, \qquad \text{(E9)}$$

$$c_m^n = [(\Delta y_3^m - \Delta y_2^m)(R_2^{nm} - R_1^{nm}) - (\Delta y_2^m - \Delta y_1^m)(R_3^{nm} - R_2^{nm})] / \delta_m, \qquad \text{(E10)}$$

$$d_m^n = [(\Delta x_2^n - \Delta x_1^n)(S_3^{nm} - S_2^{nm}) - (\Delta x_3^n - \Delta x_2^n)(S_2^{nm} - S_1^{nm})] / \delta_n, \qquad \text{(E11)}$$

$$e_m^n = [(\Delta x_2^m - \Delta x_1^m)(R_3^{nm} - R_2^{nm}) - (\Delta x_3^m - \Delta x_2^m)(R_2^{nm} - R_1^{nm})] / \delta_m. \qquad \text{(E12)}$$

The remaining coefficients can be determined by back substitution of Eqs. (E8)-(E12) into (E3) evaluated at the vertices and they are not given here.

The approximation (E7), which is applied for $m \neq n$, essentially replaces the smooth correlation function with a series of triangular plates. These plates are exact at the vertices and continuous to zeroth order where two plates come together. (They are not continuous where a plate with $m \neq n$, butts up against a plate where $m = n$). Clearly, the approximation is best when the spatial scale of the plates is small compared to the background error correlation scale. Thus, the approximation improves as $M/K = M^2/L$ becomes larger. However, we note from the operation count that the count for Eq. (E5), i.e., $9M^2$, also increases in this case. Thus, when the approximation becomes more accurate, it also becomes more expensive (an unfortunately all too common occurrence).

The extension to the two-dimensional multivariate case is straightforward. If there are three variables-geopotential Φ and horizontal wind components (u,v), whose background error is coupled multivariately through a geostrophic relationship, then there are nine covariances to consider-<ΦΦ>, <Φu>, <Φv>, <uΦ>, <uu>, <uv>, <vΦ>, <vu>, and <vv>. Thus, we have to calculate and store the a_m^n, b_m^n for all nine of the multivariate correlations. Operation (E4) is replaced by nine sums (over shorter vectors), and (E6) is replaced by three equations (one for each of F,u,v). However, the operation count for (E4) and (E6) does not change (assuming the total number of observations L, the number of volumes M, and number of observations within each volume do not change). The operation count for (E5) does change, however, and becomes $81M^2$. Thus, the total operation count for the two-dimensional multivariate case is

Table E1 — Reduction of the Norm of the Gradient/Iteration — Block-Diagonal Algorithm

Correlation Length (L_b) (km)	Φ Observations	Wind Observations	Mixed Observations
100	0.687	0.337	0.584
200	0.813	0.609	0.779
400	0.894	0.764	0.867
800	0.919	0.851	0.904

$$3KM + 81M^2 + 3KM. \tag{E13}$$

We note that the coefficient in front of the operation (E5) is increasing, but for $M = K = L^{1/2}$, this algorithm is still an O(L) matrix vector multiply. It might be noted that all the operations (E4)-(E6) are themselves matrix vector multiplies, so fast linear algebra operators (BLAS) can be used.

We now show some results based on two-dimensional univariate and multivariate analysis for the application of this algorithm. The algorithm is applied over a domain that covers the western half of North America. There are 1600 observations (winds, geopotential, or both), and the domain is divided into 64 triangular observation volumes. The observation error is uncorrelated, and the observation error is equal to the background error. The background error correlation function is SOAR (Eq. (2.19)) with correlation length L_h. We consider four cases-L_b = 100, 200, 400, and 800 km. We consider first the block diagonal preconditioner of (3.14) and measure the converge of the descent algorithm by calculating the reduction in the norm of the gradient at each iteration step (k),

Table E2 — Speed-Up Using the Triangular Plate Algorithm

Correlation Length (L_b) (km)	Φ Observations	Wind Observations	Mixed Observations
100	1.02	<1.00	<1.00
200	2.24	<1.00	1.94
400	3.35	1.12	2.24
800	4.22	2.43	3.06

$$c_k = \| \nabla_k J \| / \| \nabla_{k-1} J \|, \tag{E14}$$

for each value of k and then averaging over k. We plot below the results for geopotential observations, wind observations, and mixed observations.

The lower the number; the faster the convergence. As expected, convergence improves when correlation length decreases. Wind convergence is faster than geopotential convergence because it involves the second derivatives of the correlation functions and the effective correlation length is shorter. The multivariate convergence is somewhere between the geopotential and wind cases.

Now we consider the new preconditioner (E1-E12). The standard conjugate gradient (inner loop) for the solution of $\mathbf{A}^*\mathbf{p} = \mathbf{q}$, takes five iterations, where $\mathbf{A}^*$ is defined by (E3). The overhead/iteration in this case (five inner iterations each involving a matrix vector multiply with operation count given by (E13)) is negligible because the number of observations (L = 1600) is sufficiently large. If c is the average reduction in the norm of the gradient for each iteration of Eqs. (3.13)-(3.14) and c* is the same thing for the new algorithm, then we define the speed-up as $\log_e c^* / \log_e c$. Thus, if c = 0.70 and c* = 0.49, the speed-up would be 2. The speed-ups in the same format as above are

In general, one can see that where the block-diagonal preconditioner converges rapidly (small L_b), the new preconditioner does not improve the convergence. Where the block diagonal method does not converge rapidly (large L_b), the new method may substantially improve the convergence rate. This result tends to improve as the observation density increases. For example, with 3000 mixed observations on a domain of one-fourth the area (i.e., roughly eight times the observation density) and L_b = 400 km, the speed-up was 5.8.

The extension to the three-dimensional case can be done by producing sums like (E3) at each vertical level. Thus, if there were 10 vertical levels, the operation count for (E13) would increase to $3KM + 810M^2 + 3KM$. The size of the coefficient in the middle term is becoming quite large, but the overhead can always be hidden if L is sufficiently large. Thus, this algorithm is likely to become more and more useful as the number of observations become very large, a desirable property when using Method C.

Appendix F

Horizontal Scale Variation of the Geostrophic Coupling

Section (4.3.7) showed how μ can vary in the vertical. That is, μ is close to 1 (or -1, depending on the hemisphere) for grave vertical modes, but μ becomes closer and closer to zero for the shallow vertical modes. In other words, it is only the deep vertical modes that are highly geostrophically coupled. It is possible to make the same argument in the horizontal, i.e., that the geostrophic coupling should be a maximum at large horizontal scales, and the smaller horizontal scales (below meso α, say) should be increasingly uncoupled, because geostrophy is not relevent on those scales. This geostrophic decoupling at smaller spatial scales would be particularly relevent for the inner mesh of COAMPS.

A simple procedure for geostrophic decoupling at smaller horizontal scales can be developed as follows: Consider the $<\Phi u>$ correlation, given by Eq. (5.3.15) of Daley, 1991) but with the current notation

$$c_{\Phi u} = \mu\, L^h \sin(\alpha)\, dc_{\Phi\Phi}/ds, \qquad \text{(F1)}$$

where α is the angle between the two points, $c_{\Phi\Phi}$ is the $<\Phi\Phi>$ correlation, s is the Great Circle distance, and we have dropped the "n,m" notation. Let us suppose that $c_{\Phi\Phi}$ uses the SOAR model (4.22). Then, we have $dc_{\Phi\Phi}/ds = -(L^h)^{-2}\, s \exp(-s/L^h)$. The Hankel transform pair of $c_{\Phi\Phi}$ (see Daley, 1991, Section 3.3) is

$$c_{\Phi\Phi}(s) = \int_0^\infty g_{\Phi\Phi}(k)J_0(ks)kdk \quad \text{and} \quad g_{\Phi\Phi}(k) = \int_0^\infty c_{\Phi\Phi}(s)J_0(ks)sds, \qquad \text{(F2)}$$

where k is the wavenumber, J_0 is the Bessel function of the first kind of integer order zero, and $g_{\Phi\Phi}(k)$ is the Hankel transform of $c_{\Phi\Phi}$. Using (F2), we write

$$dc_{\Phi\Phi}\, /\, ds = -\int_0^\infty g_{\Phi\Phi}(k)J_1(ks)k^2dk \qquad \text{(F3)}$$

where $J_1(ks)$ is the Bessel function of the first kind of integer order 1. Let us now suppose that the coupling parameter m is a function of wavenumber k, so that we can write

$$c_{\Phi u} = -L^h \sin(\alpha)\int_0^\infty \mu(k)g_{\Phi\Phi}(k)J_1(ks)k^2dk, \qquad \text{(F4)}$$

If $\mu(k)$ is independent of k, then (F4) becomes identical to (F1). Now for the SOAR model, $g_{\Phi\Phi}(k)$ is given by Eq. (3.3.27) of Daley (1991) and is equal to

$$g_{\Phi\Phi}(k) = 3(L^h)^2\, /(1 + (L^h)^2k^2)^{5/2}. \qquad \text{(F5)}$$

Hankel transforms for other common correlation models are given in Daley (1991) and if such transforms are unknown, $g_{\Phi\Phi}(k)$ can be obtained from the second equation of (F2) using numerical quadrature.

It is now evident that a spectrally varying geostrophic coupling can be obtained using Eq. (F4) simply by specifying $\mu(k)$. For any but the most trivial choice of $\mu(k)$, this integral will have to be done by numerical quadrature. (Note that this integral can be done in advance and stored as a table look-up as a function of Great Circle distance s and horizontal scale L^h).

Scale-dependent geostrophic coupling is illustrated in Figs. F1 and F2. Figure F1 plots the spectrum (solid) of the SOAR derivative function, that is, $k^2 g_{\Phi\Phi}(k)$. This is for L^h = 100 km. The dash-dot curve plots $\mu(k)k^2 g_{\Phi\Phi}(k)$, after choosing $\mu(k)$ so that the smaller spatial scales are smoothly filtered. One can also think of the solid curve as representing the case when $\mu(k) = 1$ for all k. Figure F2 plots the function $c_{\Phi u}$ for the case when $\alpha = 90°$. The solid curve corresponds to the solid curve in Fig. F1 and is the geopotential / wind correlation when all scales are completely geostrophically coupled. The dash-dot curve of Fig. F2 corresponds to the dash-dot curve of Fig. F1 and only the larger scales are geostrophically coupled. Comparison of the two curves indicates an overall reduction in correlation except at large separations and a general shift to larger scales.

It might be noted that Hankel transforms are appropriate for infinite f-planes, but less so for the sphere, where an ordinary Legendre polynomial expansion would be more appropriate. This technique is straightforward to apply, and means that geostrophic coupling can be both (slowly) horizontally varying and horizontally scale dependent.

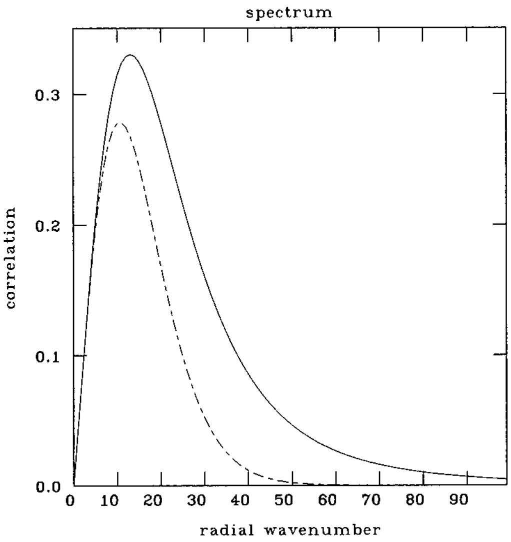


Figure F1
Radial spectrum for geostrophically-coupled SOAR model

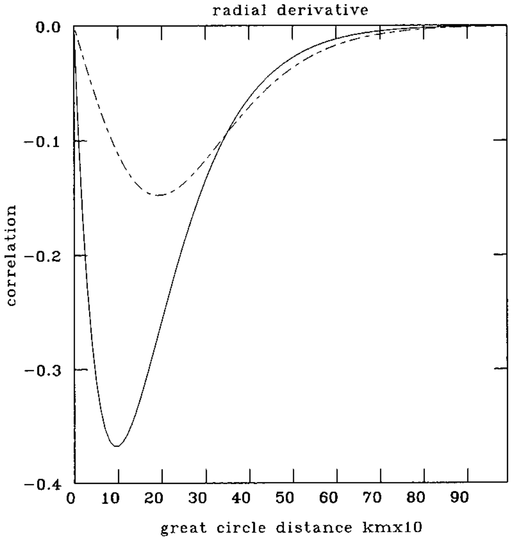


Figure F2
Correlations corresponding to the spectra of Figure F1

Appendix G

Hyperspectral Sounders (Linearized Form)

Vertical temperature sounders that have many channels (1000 or more) may soon be operational. An example of such a sounder is the NASA AIRS instrument. The high spectral resolution is intended to provide higher vertical spatial resolution in the retrievals. Now if these sounders actually can resolve features with small vertical scales and the retrieval is essentially independent of any specified background field, then we would be perfectly justified in assimilating these retrieved temperatures and moistures. However, bitter experience with previous sounders should make us skeptical of claims made for such instruments. In other words, it may still be necessary to directly assimilate radiances from these advanced sounders.

The nadir temperature-sounding algorithm developed in Section (5.3) would be very inefficient in this instance. There are two reasons for this. Firstly, it is necessary to calculate and store the $N_c x N_v^s$ rectangular matrix for each sounding. If N_c is O(1000), this is clearly not very viable. However, there is a more significant problem involving the preconditioner for the conjugate-gradient descent. As noted in Section (4.4), we must generate the $\mathbf{C}_b^{ob/ob}$ correlation in the case of the preconditioner. The linear equation solve by Choleski decomposition for the preconditioner is actually done in the space of the observations (i.e., radiance space for nadir sounders), not in vertical eigenvector space. While this operation applies only to the diagonal blocks, it could become prohibitive for sounders with a large number of channels. We have already shown that only a small number of vertical eigenmodes of the background error correlation are adequate to represent the O(20) channels of the TOVS instrument. Perhaps, for hyperspectral sounders, it might be possible to solve the whole problem in vertical eigenvector space, instead of going back and forth between radiance and eigenvector space for every iteration of the conjugate gradient algorithm. Note that the following discussion (as in Section 5.3) is limited to the linear case; the nonlinear problem is discussed in Section 6.

To explore this idea further, it is instructive to introduce a second eigenvector problem.

This problem has recently been examined by Joiner and da Silva (1998), and the following results generally confirm theirs. Following the notation of Eqs. (5.2)-(5.6), consider the $N_c x N_c$ radiance error covariance matrix $\mathbf{R}$ and the $N_v x N_c$ Jacobian matrix $\mathbf{H}$ of a given instrument (with $\mathbf{H}$ obtained by linearization about some vertical temperature profile). Form the $N_v x N_v$ symmetric matrix $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$. Now consider Eq. (2.10) and pre- and post-multiply it by $\mathbf{P}_b^{1/2}$, giving

$$\mathbf{P}_b^{-1/2}\mathbf{P}_a\mathbf{P}_b^{-1/2} = [\mathbf{I} + \mathbf{P}_b^{1/2}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{P}_b^{1/2}]^{-1}, \qquad \text{(G1)}$$

where $\mathbf{P}_a$ is the retrieval error covariance. $\mathbf{P}_b^{1/2}$ can be obtained from the background error covariance $\mathbf{P}_b$ by finding the eigenvectors and eigenvalues of $\mathbf{P}_b$; taking the square root of the eigenvalues (which is always possible for positive-definite matrices, see Eq. (9.8)); and then appropriately re-assembling the eigenvectors and square-rooted eigenvalues. We see that $\mathbf{P}_b^{-1/2}\mathbf{P}_a\mathbf{P}_b^{-1/2}$ is the retrieval error covariance normalized by the background error covariance. To simplify the discussion, let us assume that the background error covariance is of the simple diagonal form $\mathbf{P}_b = \varepsilon_b^2\mathbf{I}$, although in practice (Section 4.3) we use a vertical background temperature error covariance, which is considerably more sophisticated. For this simple background error covariance, the eigenvectors of $\mathbf{P}_b^{-1/2}\mathbf{P}_a\mathbf{P}_b^{-1/2}$ are the same as the eigenvectors of $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$. Denote the nth eigenvalues of $\mathbf{P}_b^{-1/2}\mathbf{P}_a\mathbf{P}_b^{-1/2}$ and $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ as λ_a^n and λ_r^n, respectively. Then,

$$\lambda_a^n = 1/[1 + \varepsilon_b^2\lambda_r^n]. \qquad \text{(G2)}$$

Rodgers (1998) defines the signal-to-noise ratio (SNR) of the nth retrieved mode as

$$[(1 - \lambda_a^n)/\lambda_a^n]^{1/2} = \varepsilon_b[\lambda_r^n]^{1/2}. \qquad (G3)$$

The information in the nth mode, following Rodgers (1998) and Shannon and Weaver (1949), is

$$-0.5 \log_2(\lambda_a^n) = 0.5 \log_2(1 + \varepsilon_b^2\lambda_r^n). \qquad (G4)$$

The total information is obtained by summing over all the eigenvalues. Another useful quantity is the total number of independent degrees of freedom, which is given by

$$\sum_n [1 - \lambda_a^n] = \sum_n \varepsilon_b^2\lambda_r^n/[1 + \varepsilon_b^2\lambda_r^n]. \qquad (G5)$$

We illustrate these concepts by constructing the eigenstructure of the matrix $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ for a particular vertical column with **R** and **H** appropriate for the TOVS instument. Figure G1 shows the nine eigenvectors with the largest eigenvalues for this instrument (40 eigenvectors in all). On this figure is indicated the eigenvector and corresponding eigenvalue in three panels (a, b, and c). It can be seen that the gravest vertical modes (least zero crossings) have the largest eigenvalues. Moreover, the eigenvalues decrease very rapidly for eigenvectors with many zero crossings. From Eq. (G3), we can see that the gravest vertical modes have the most favorable SNR and the "shallow" modes have an extremely unfavorable SNR. This figure illustrates the well-known fact (at least in the data assimilation community) that the TOVS instrument could not possibly "see" the tropopause, marine boundary layer, inversions, or other temperature phenomena that vary rapidly with altitude. Shallow features of this kind, which appear in externally derived TOVS retrievals, are, of course, artifacts of whatever has been specified as background information.

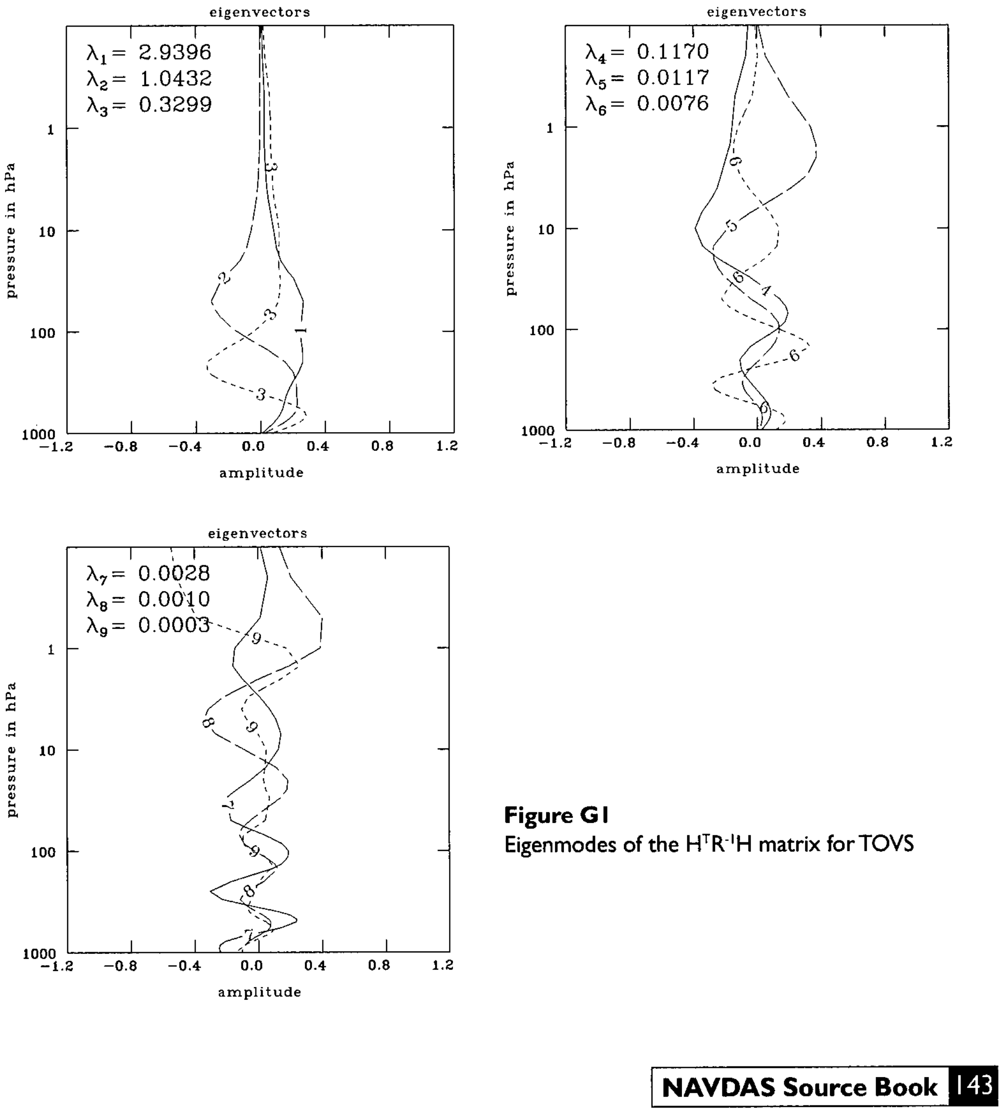


Figure G1
Eigenmodes of the $H^TR^{-1}H$ matrix for TOVS

We can estimate the real number of degrees of freedom in the TOVS instrument by using Eq. (G5) from the eigenvalues shown in Fig. G1. If the background were specified to be climatology, we might expect the background temperature error e_b to be, say, 10 degrees Kelvin, while for a background derived from a 6-hour forecast, ε_b might be 2.5 degrees Kelvin. Using (G5) and these choices of ε_b, the number of degrees of freedom is 5.2 for a climatological background and 3.0 for a forecast background. Thus, for data assimilation purposes where a reasonably good background is available, there isn't very much useful information in a TOVS retrieval.

Compare also Fig. G1 with Fig. 4.1, which shows the three gravest vertical modes of the background streamfunction/streamfunction background error correlation (which is similar to the geopotential/geopotential correlation). If these two figures are compared qualitatively, it can be seen that the gravest vertical modes (which have the largest error) of the background geopotential correlation bear some resemblance to the gravest vertical modes (which contain the most information) of the instrument information matrix. Or to put it another way, the instrument is able to observe only the large vertical scales, but fortunately, the background has maximum error in those scales. This happy coincidence explains why we are able to assimilate the TOVS radiances with so few vertical modes of the background error correlation.

Joiner and da Silva (1998) also found that there was little information except in the gravest vertical modes for nadir temperature sounders. Their suggested solution for the direct assimilation of hyperspectral radiance observations was to project the radiance innovations onto the eigenvectors of $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$. Although this approach has some advantages (to be discussed later) and was tested by us, we have not followed this approach here. In Sections 4.4 and 5.3, a methodology has been developed based on an expansion in the eigenvectors of the background error correlation, and we intend to apply this same idea to hyperspectral sounders. To see how this might be done first requires the derivation of a matrix identity.

Suppose **A** and **B** are square, symmetric positive-definite matrices that are not necessarily of the same order, and **Q** is a rectangular matrix whose two dimensions conform with those of **A** and **B**. Then, it can be shown, using the Sherman-Morrison-Woodbury formula that

$$\mathbf{A}\mathbf{Q}^T[\mathbf{Q}\mathbf{A}\mathbf{Q}^T + \mathbf{B}]^{-1} = [\mathbf{A}^{-1} + \mathbf{Q}^T\mathbf{B}^{-1}\mathbf{Q}]^{-1}\mathbf{Q}^T\mathbf{B}^{-1}$$

$$= [\mathbf{A}^{-1} + \mathbf{C}^{-1}]^{-1}\mathbf{Q}^T\mathbf{B}^{-1} = \mathbf{A}[\mathbf{A} + \mathbf{C}]^{-1}\mathbf{C}\mathbf{Q}^T\mathbf{B}^{-1}, \qquad \text{(G6)}$$

$$\text{where } \mathbf{C}^{-1} = \mathbf{Q}^T\mathbf{B}^{-1}\mathbf{Q}.$$

Now define $\underline{\mathbf{R}} = [\underline{\mathbf{H}}^T\mathbf{S}_h^{1/2}\mathbf{R}^{-1}\mathbf{S}_h^{1/2}\underline{\mathbf{H}}]^{-1}$. Application of (G6) to (5.5) yields

$$\mathbf{T}_a - \mathbf{T}_b = \mathbf{H}_s\mathbf{S}_v^{1/2}\underline{\mathbf{E}\mathbf{D}}[\underline{\mathbf{D}} + \underline{\mathbf{R}}]^{-1}\underline{\mathbf{R}\mathbf{H}}^T\mathbf{S}_h^{1/2}\mathbf{R}^{-1}[\mathbf{y} - \mathrm{H}(\mathbf{T}_b)]. \qquad \text{(G7)}$$

The matrix $\underline{\mathbf{R}}^{-1}$ is similar to the instrument temperature information matrix $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$, except that it is also involves the projection on the eigenvectors of the background error correlation. Note the form of (G7), $\mathbf{S}_h^{1/2}\mathbf{R}^{-1}[\mathbf{y} - \mathrm{H}(\mathbf{T}_b)]$ are slightly processed radiance innovations. However, after this radiance space vector has been multiplied by $\underline{\mathbf{H}}^T$; the resulting vector is in vertical background error eigenvector space, **and the calculation remains in the eigenvector space until the final multiplication by $\mathbf{H}_s\mathbf{S}_v^{1/2}\underline{\mathbf{E}}$ to grid space.** In other words, once we go to vertical eigenvector space, we need never return to the thousands of radiance channels and we never need $\underline{\mathbf{H}}^T$ again. We can stay in vertical eigenvector space right though the conjugate gradient descent; we do not have to go back and forth between radiance and eigenvector space during every iteration.

Instead of carrying around the $N_c \times N_v^s$ matrix $\underline{\mathbf{H}}$, we must carry around the $N_v^s \times N_v^s$ matrix $\underline{\mathbf{R}}$, but this is a much smaller matrix as $N_c >> N_v^s$. The big question is, can we obtain $\underline{\mathbf{R}}$ when it involves the inversion of the admittedly small, but possibly singular matrix $\underline{\mathbf{R}}^{-1} = \underline{\mathbf{H}}^T\mathbf{S}_h^{1/2}\mathbf{R}^{-1}\mathbf{S}_h^{1/2}\underline{\mathbf{H}}$. We should not be surprised if $\underline{\mathbf{R}}^{-1}$ is close to singular (contains very small eigenvalues), because we have already shown that the related matrix of $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ is essentially singular (due to the extremely poor SNR from the instrument for the the very shallow vertical structures).

$\mathbf{R}$ is generally diagonal, so there should be no problem obtaining $\mathbf{R}^{-1}$. However, as noted above, we cannot take it for granted that $\underline{\mathbf{R}}^{-1}$ can be safely inverted. However, $\underline{\mathbf{R}}^{-1}$ is of small dimension (10 say), and there is no difficulty in finding its complete eigenstructure (for every sounding). If there is a very small eigenvalue of $\underline{\mathbf{R}}^{-1}$ (large eigenvalue of $\underline{\mathbf{R}}$), we can eliminate it and obtain $\underline{\mathbf{R}}$ in the space that does not contain this troublesome eigenvector.

The third column of Table 5.1 shows the results for a case with seven vertical modes and a top at 1 mb. The results in the 2nd column were obtained by using Eq. (5.5) and those in the 3rd column by using Eq. (G7). That is, the $\underline{\mathbf{D}} + \underline{\mathbf{R}}$ matrix in (G7) was 7×7, as opposed to 18×18 in (5.5). It can be seen that the results in the 2nd and 3rd columns are very close to each other.

We complete this discussion, by comparing eigenvector projection in the eigenspace of $\mathbf{H}^{T}\mathbf{R}^{-1}\mathbf{H}$, that is, the Joiner and da Silva (1998) idea, and eigenvector projection in the vertical background error correlation space, as discussed here. Projecting onto the eigenstructure of $\mathbf{H}^{T}\mathbf{R}^{-1}\mathbf{H}$ undoubtedly requires the fewest vertical modes to represent the radiance information. Such a procedure essentially diagonalizes the observation error covariance, rather than the background error covariance that has been stressed here. Despite the disadvantage of being a less efficient method of representing the radiance information, projection on the eigenmodes of the background error correlation has two distinct advantages. Firstly, the eigenvectors are universal (even though the eigenvalues may not be), instead of being different for every sounding, as they would be if the modes of $\mathbf{H}^{T}\mathbf{R}^{-1}\mathbf{H}$ were used. Secondly, and more importantly, in the conjugate gradient descent, the observation error covariance only appears in the diagonal blocks, it does not appear in the off-diagonal blocks. From the point of view of overall operational efficiency, it is much more beneficial to vertically diagonalize the off-diagonal blocks because there are so many of them.

Appendix H

De-aliasing Scatterometer Winds Using the $J_{\min}$ Diagnostic

Section 5.9 noted that scatterometer winds can have up to four ambiguous wind directions. Choosing the right direction is very important. One has two choices, an off-line preprocessing algorithm such as the ECMWF PRESCAT routine, or allowing the assimilation algorithm to choose the best direction itself (Stofelen and Anderson, 1997). This latter goal cannot be achieved by minimizing the L_2 (i.e., quadratic) norm because the cost function has only one minimum and will generally choose a direction that lies between the possible choices. One has to have a cost function for the scatterometer that is nonquadratic and therefore has many minima, and hope that the background is sufficiently accurate and there are enough other observations to force the analysis system into the right basin of the cost function.

We have opted for the preprocessing route. This is consistent with the present NRL MVOI procedure, which chooses a "best" direction as that which is closest to the background 10-m wind direction at the appropriate time and for the appropriate model (NOGAPS or COAMPS). What we desire to do here is develop an iterative procedure in which we choose the direction closest to the background wind direction for the first iteration. We would then calculate a diagnostic that, depending on its value, may or may not imply that one of the other possible wind directions is more likely. We would then change the wind directions where indicated, recalculate the diagnostic, and continue iterating until there is no change in the diagnostic.

As noted above, there are normally four ambiguous wind directions. Two of them are relatively close together, and the other two are about 180 degrees different from the first two directions. The windspeeds are usually different for each of the choices. Sometimes, this degenerates into two or three choices.

The idea works as follows. At the first iterate, we choose the solution closest to the background wind direction. We assume we have available the specified wind observation and background error covariances appropriate for the scatterometer observations. Suppose there are L scatterometer wind vector observations, each of which has up to four choices. Denote these vectors of length 2L as $[\mathbf{u}^1, \mathbf{v}^1]$, $[\mathbf{u}^2, \mathbf{v}^2]$,$[\mathbf{u}^3, \mathbf{v}^3]$, and $[\mathbf{u}^4, \mathbf{v}^4]$, with elements $[u_\ell^1, v_\ell^1]$, $[u_\ell^2, v_\ell^2]$, $[u_\ell^3, v_\ell^3]$, $[u_\ell^4, v_\ell^4]$, $1 \leq \ell \leq L$, respectively. For convenience, we assume that the solution with direction closest to the background wind direction is $[\mathbf{u}^1, \mathbf{v}^1]$. In degenerate cases, some elements of the vectors will be the same. We define the vector length 2L of background winds as $[\mathbf{u}_b, \mathbf{v}_b]$ with elements $[u_\ell^b , v_\ell^b]$.

Now consider the $J_{\min}$ diagnostic (9.1), that is,

$$J_{\min} = [\mathbf{y} - H(\mathbf{x}_b)]^T[\mathbf{HP}_b\mathbf{H}^T + \mathbf{R}]^{-1}[\mathbf{y} - H(\mathbf{x}_b)]. \tag{H1}$$

At the first iterate, let us define $\mathbf{y}$ as the vector of length 2L that is equal to $[\mathbf{u}^1, \mathbf{v}^1]$. The forward operator is trivial in this case and we can define the 2L×2L matrix $\mathbf{HP}_b\mathbf{H}^T = \mathbf{P}_b^{ob/ob}$ as in Eq. (3.15). Then define $\mathbf{x}_b = [\mathbf{u}_b, \mathbf{v}_b]$ as the background wind field at the observation locations and $J_{\min}$ can be rewritten as

$$J_{\min} = [\mathbf{y} - \mathbf{x}_b]^T[\mathbf{P}_b^{ob/ob} + \mathbf{R}]^{-1}[\mathbf{y} - \mathbf{x}_b]. \tag{H2}$$

Following Eq. (3.5), we also define the innovation $\mathbf{d} = [\mathbf{y} - \mathbf{x}_b]$ and $\mathbf{z} = \mathbf{Ad}$, where $\mathbf{A} = [\mathbf{P}_b^{ob/ob}+\mathbf{R}]^{-1}$ is the 2L×2L inverse matrix.

The central idea of the algorithm is that we test all four solutions for the scatterometer observations to see if they can reduce the value of J_{min}, below that given in (H2). If any choice reduces the value of J_{min}, then it is considered more likely than the original choice of the closest direction to the background wind direction.

Consider the ℓth scatterometer wind observation and the innovation using the first wind direction (that is, the direction closest to the background wind direction). Then, this particular innovation is $[u_\ell^1 - u_\ell^b, v_\ell^1 - v_\ell^b]$. Now consider the second direction for the ℓth location, that is, $[u_\ell^2, v_\ell^2]$. Then define

$$J_{min}(\ell,2) = [\mathbf{d}_\ell^2]^T\mathbf{A}\mathbf{d}_\ell^2, \tag{H3}$$

$\mathbf{d}_\ell^2$ is the vector of length 2L whose elements are the same as those of $\mathbf{d}$, except for the ℓth element, where $[u_\ell^1 - u_\ell^b, v_\ell^1 - v_\ell^b]$ is replaced by $[u_\ell^2 - u_\ell^b, v_\ell^2 - v_\ell^b]$. Then, if $J_{min}(\ell,2) < J_{min}$, then the direction $[u_\ell^2, v_\ell^2]$ is more likely than $[u_\ell^1, v_\ell^1]$. Repeat with $[u_\ell^3, v_\ell^3]$ and $[u_\ell^4, v_\ell^4]$ to find the best of the four observations. Note, at this point, we are only finding the change in J_{min} caused by cycling through the four observations at the ℓth observation location while leaving the other observations unchanged.

We proceed in this way through all L observation locations, changing each in turn and finding which of the four solutions produces the minimum value of J_{min}. At the end of this process, we expect that most of the observations will still be the first observation, i.e. $[u_\ell^1, v_\ell^1]$, but for a few locations one of the other four observations will have been chosen. This is the end of the second iteration and we recalcuate J_{min} from (H2) using this new choice of observations. We would expect that J_{min} calculated at the end of the second iteration would be smaller than that calculated at the end of the first iteration, indicating that we have reduced the minimum of the cost function, and therefore this second choice of observed wind directions is closer to the true directions. We then repeat the whole process until there is no further reduction in J_{min} between iterations.

We would use Eq. (H2) to calculate J_{min} at the end of each iteration, but there is a much more efficient way to calculate $J_{min}(\ell,2)$, $J_{min}(\ell,3)$, or $J_{min}(\ell,4)$ than Eq. (H3). We illustrate for the observation $[u_\ell^2, v_\ell^2]$. Define $\mathbf{d}_\ell^2 = \mathbf{d} + \Delta\mathbf{d}_\ell^2$, where $\Delta\mathbf{d}_\ell^2 = 0$ everywhere except the ℓth element, which is equal to $[u_\ell^2 - u_\ell^1, v_\ell^2 - v_\ell^1]$. Then, Eq. (H3) can be rewritten as

$$J_{min}(\ell,2) = [\mathbf{d} + \Delta\mathbf{d}_\ell^2]^T\mathbf{A}[\mathbf{d} + \Delta\mathbf{d}_\ell^2] = \mathbf{d}^T\mathbf{A}\mathbf{d} + [\Delta\mathbf{d}_\ell^2]^T\mathbf{A}\mathbf{d} + \mathbf{d}^T\mathbf{A}\Delta\mathbf{d}_\ell^2 + [\Delta\mathbf{d}_\ell^2]^T\mathbf{A}\Delta\mathbf{d}_\ell^2. \tag{H4}$$

The first term of (H4) is equal to J_{min}, which is known, and the second two terms are equal because of the symmetry of $\mathbf{A}$. We can then write (H4) as

$$J_{min}(\ell,2) = J_{min} + 2[\Delta\mathbf{d}_\ell^2]^T\mathbf{z} + [\Delta\mathbf{d}_\ell^2]^T\mathbf{A}\Delta\mathbf{d}_\ell^2. \tag{H5}$$

The second term of (H5) includes $\mathbf{z}$, which is known, but there are only two multiplications because $\Delta\mathbf{d}_\ell^2$ is mostly zeroes. The last term in (H5) can be-written as

$$[\Delta\mathbf{d}_\ell^2]^T\mathbf{A}\Delta\mathbf{d}_\ell^2 = \begin{bmatrix} u_\ell^2 - u_\ell^1 & v_\ell^2 - v_\ell^1 \end{bmatrix}\begin{bmatrix} a_\ell^{uu} & a_\ell^{vu} \\ a_\ell^{uv} & a_\ell^{vv} \end{bmatrix}\begin{bmatrix} u_\ell^2 - u_\ell^1 \\ v_\ell^2 - v_\ell^1 \end{bmatrix}. \tag{H6}$$

a_ℓ^{uu}, a_ℓ^{vv}, etc., are the appropriate elements of $\mathbf{A}$. Equation (H6) is simply multiplication with a 2×2 matrix. In order to calculate $J_{min}(\ell,2)$ using (H4), we need the inverse $\mathbf{A} = [\mathbf{P}_b + \mathbf{R}]^{-1}$, which limits us to handling O(500) scatterometer wind vectors simultaneously.

Before proceeding to illustrate this technique, we briefly discuss why reducing J_{min} in (H1) or (H2) is likely to lead to a better choice of scatterometer wind observations than the vector $[\mathbf{u}_1\ \mathbf{v}_1]$. First, we note that since $[\mathbf{u}_1\ \mathbf{v}_1]$

is, for each element, the closest observation to the background vector [$\mathbf{u}_b$ $\mathbf{v}_b$], then the choice [$\mathbf{u}_l$ $\mathbf{v}_l$] minimizes $[\mathbf{y}-\mathbf{x}_b]^T[\mathbf{y}-\mathbf{x}_b]$. It also minimizes $[\mathbf{y}-\mathbf{x}_b]^T\mathbf{F}[\mathbf{y}-\mathbf{x}_b]$, where **F** is any diagonal matrix, all of whose elements are real and positive. In particular, [$\mathbf{u}_l$ $\mathbf{v}_l$] minimizes

$$J^*_{\min} = [\mathbf{y}-\mathbf{x}_b]^T \, \mathrm{diag}[\mathbf{P}_b^{ob/ob} + \mathbf{R}]^{-1} \, [\mathbf{y}-\mathbf{x}_b] \,. \tag{H7}$$

But, (H7) does not account for any background error correlation. Since we have assumed that the background error is spatially and multivariately correlated, minimization of (H7) is not the best that can be done, and thus [$\mathbf{u}_l$ $\mathbf{v}_l$] is not necessarily the optimal choice of scatterometer wind observation vector. If the background error covariance is correctly specified, then minimization of (H1) or (H2) should yield a better choice of scatterometer wind observations.

This can be illustrated in another way. The largest eigenvalues of the background error covariance correspond to the large-scale nondivergent modes, while the smallest eigenvalues correspond to the small-scale divergent modes. But the background error covariance appears as an inverse in (H2). This means that $J_{\min}$ in (H2) will be large if there are large-amplitude, small-scale divergent components of the innovation vector. But, divergent small-scale flows are precisely what one would expect when a few observations were assigned inconsistent directions. Thus, any procedure that reduces $J_{\min}$ is likely to suppress undesirable small-scale divergent circulations caused by incorrect direction assignments.

We now illustrate the technique using a known "true" 10-m windfield, as in Section 6.2. The "true" windfield was generated using a gaussian random number generator and contained both rotational and divergent components. The background windfield error was assumed to be nondivergent, as in Section 6.2, and to have a characteristic horizontal scale L_h. The background error covariance $\mathbf{P}_b$ was consistent with this assumption. The background windfield was obtained by adding the background wind error to the "truth." As in Section 6.2, this implied the background wind itself contained both divergent and rotational components. Following Section 6.2, we defined $\overline{\mathbf{v}^t}$ as the true domain averaged windspeed, $\overline{\varepsilon_v}$ as the rms background wind error, and a = $\overline{\varepsilon_v} / \overline{\mathbf{v}_t}$.

In order to simulate a satellite swath, we defined a rectangular domain with nine grid points in the x (cross-track) direction and 41 grid points in the y (along-track) direction. $\Delta y = \Delta x$. The observations were assumed to be the grid points, and thus there were 369 pseudo-scatterometer wind observation locations. We set $\alpha = 0.3$, which meant that the average direction error in the background windfield was 25.4 degrees. (There were, of course, subtantial errors in the background windspeed as well). $L_h = 3 \Delta x$. For each experiment, there were ten realizations and the results shown are the average for these realizations.

Four wind observations were generated at each observation location. The first was the "true" windspeed and direction. The second observation had a wind direction that was randomly perturbed from the true wind direction. Its windspeed was also perturbed randomly. The third wind direction was 180 degrees out of phase with the first, and the fourth wind direction was 180 degrees out of phase with the second. The third and fourth observation also had differently perturbed windspeeds. The specified observation error variance was consistent with these observation errors.

At the beginning of the first iteration, at each location, we chose the wind vector whose direction was closest to the background wind direction. Since we knew the "truth," we could calculate the percentage of incorrect choices. This was 13.8 %, which seems reasonable, given a background wind direction error of 25 degrees and four possible choices for each observaton location. At this point, we calculate $J_{\min}$ and then proceed on the procedure of Eqs. (H3)-(H6). Table H1 shows the results of this experiment for four iterations.

In Table H1, the $J_{\min}$ values have been normalized by the value at iteration 1. The minimum of the cost function has been reduced substantially, and we can detect most of the incorrect choices that were made at iteration 1.

We performed a second experiment experiment in which there were only two choices for each wind location—the "truth," and another wind vector that was 180 degrees out of phase with the truth and with a different windspeed. We would expect that picking the initial choice as being closest to the background wind direction would be the correct choice more often in this case. The results are shown in Table H2.

Table H1 — Four Aliased Directions

Iteration Number	Percentage Incorrect	J_{min}
1	13.8	1.00
2	5.1	0.65
3	3.8	0.63
4	3.5	0.62

Here again, the algorithm is effective in determining the incorrect initial assignments. It is also clear from Tables H1 and H2 that calculating J_{min} and watching it decrease with iteration number does indeed correspond to an increasing number of correct assignments.

Table H2 — Two Aliased Directions

Iteration Number	Percentage Incorrect	J_{min}
1	2.1	1.00
2	0.2	0.81

The algorithm apparently works, because it uses the specified background error covariance to indicate if any wind direction assignments are inconsistent with neighboring observations. Clearly, if the background error is very large and there are many initial erroneous assignments, the algorithm will be less able to pick out erroneous assignments.

We complete this section by testing the algorithm on real scatterometer observations and real NOGAPS 1000 hPa background wind fields. We generalized the background error covariances for this demonstration, including the correlations with the divergent wind discussed in Section 4.7.3. In this test, there were two ambiguous wind vectors at each observation point. At the first iteration, we chose the wind vector that was closest to the background, as before. In Table H3, we show the result for 200 wind observation pairs. This time, of course, we have no "truth" with which to compare our results. We show the cost function J_{min} and the percentage of observations that have changed after the first iteration.

While we have no way of knowing whether or not we have made a better choice for the scatterometer observations, we have clearly been able to reduce the cost function by choosing some different wind vectors.

Table H3 — Real Scatterometer Observations

Iteration Number	Percentage Changed From Initial	J_{min}
1	0	1.00
2	8.1	0.73
3	9.8	0.63

Appendix I

Principles of J_{min} Analysis

The only way that any model communicates with the atmosphere is through the innovations. It is important to monitor actual innovations in order to find out whether they correspond to what we have assumed in the generation of observation and background error statistics. Innovations are in observation space, and it is not easy, in general, to compare innovations across platforms. Ideally, one would like to normalize the innovations in some way, so that every innovation or group of innovations could be directly compared. The J_{min} diagnostic described in Section 9.1 provides a useful procedure for normalizing the innovations. When innovations in a particular region or for a given platform are very different than what has been assumed, then it should be possible to detect biased platforms and/or mis-specification of the background or observation error statistics. It is a technique that is particularly valuable for tuning the background and observation error covariances of NAVDAS, since we are trying to avoid using the ad hoc NMC Method. Following Eq. (9.1), we define

$$J_{\text{min}} = [y - H(x_b)]^T [HP_bH^T + R]^{-1} [y - H(x_b)]. \tag{I1}$$

We then divide J_{min} by the number of observations, and this number should be close to 1.0 If $J_{\text{min}} < 1.0$, either the observation or background error are specified too large, if $J_{\text{min}} > 1.0$, then the observation and/or background error are specified too small. In the remainder of this appendix, we will assume that J_{min} has been divided by the number of observations, so that it is an O(1), dimensionless number.

We intend to use this diagnostic by collecting many innovations from a long assimilation cycle, and then use these results to learn something about the correctness of our observation and background error specification or about biased platforms.

The following are the basic principles of the technique.

(1) J_{min} should be calculated over many time intervals and needs several million observations in order to reduce sampling errors for small horizontal and/or vertical regions. Do not make any inferences unless there are several hundred observations for any given stratification (see (2) below).

(2) Have the capability to stratify results by region, pressure, instrument type, variable, channel (for sounders), or other discriminators.

(3) The key assumption is that observation error is assumed to vary by instrument, channel, pressure, or variable, but not horizontally. Background error is assumed to vary horizontally, vertically, by variable, but not (of course) by instrument.

(4) The J_{min} diagnostic only indicates that either the observation or background error variances or both are specified to be too large or too small. That is why assumption (3) may be useful in separating the two. However, note that very large values of J_{min} may indicate that an instrument (or channel) has a serious problem (bias, perhaps) that should be addressed.

(5) Key on two regions – global and North America. Augment the J_{min} statistics with North American radiosonde innovation statistics (if available). Information from the observation monitoring sys-

tem or collocation statistics may also be used. The $J_{\min}$ diagnostic is most useful when it confirms other information.

(6) Make sure that all information has been quality controlled. Do not use innovations that have been rejected by the innovation or buddy check. However, note that buddy check and innovation check decisions do depend on the specified background and observation error statistics.

(7) Address observation error first. This is done by looking at global diagnostics (as functions of channel, pressure, and variable) for each instrument separately. Depending on how well the globe is sampled by that instrument, it may be possible to assume that globally the background error is about right (even if it is incorrect regionally) and thus any discrepancies that show up in this $J_{\min}$ calculation are due to problems with the observation error specification. For example, if examination of all the TOVS globally indicated that $J_{\min}$ *(global,TOVS,channel9)* = 3.00, then the observation error for channel 9 is probably set too low, or that there was a problem with that channel.

(8) Next, address horizontal variations of background error variance over land. This is done primarily with radiosondes. We do not stratify by pressure, only by variable. We would assume that the radisonde observation errors were okay in general, and then compare the $J_{\min}$ diagnostic over each of the continental areas with that from North America. (We assume that North America is basically okay, $J_{\min}$ *(NorthAmerica)* ≈ 1.00). Then, we would calculate $J_{\min}$ for temperature and winds. As an example, suppose we found $J_{\min}$ *(NorthAmerica,temperature)* > $J_{\min}$ *(Asia,temperature)* $J_{\min}$ and $J_{\min}$ *(NorthAmerica,winds)* > $J_{\min}$ *(Asia,winds)*, we might conclude that the background error variance specified for Asia might be too large (although, by how much, we do not know).

(9) Now address horizontal variations of background error variance over sea. Again, we are looking for bulk numbers and do not stratify by pressure or channel. Over the ocean we have two instruments that provide profile observations, water vapor/ cloud drift winds and TOVS temperatures. We determine $J_{\min}$ for both instruments for the global case—$J_{\min}$ *(global,TOVS)* and $J_{\min}$ *(global,SATWINDS)*. We then calculate $J_{\min}$ for each instrument for each of the oceanic areas. For example, suppose $J_{\min}$ *(IndianOcean,TOVS)* > $J_{\min}$ *(global,TOVS)* and similarly for SATWINDS, then we might suspect that the background error variance in the Indian Ocean was specified too low.

(10) Vertical structures are best obtained from North American innovation statistics. Failing those, information on the vertical structure can be obtained from $J_{\min}$ calculations. For example, consider the North America radiosondes. Suppose at 850hPa, $J_{\min}$ (*NorthAmerica,radiosondes,850 hPa, temperatures*) > 1.0 and $J_{\min}$ (*NorthAmerica,radiosondes,850hPa,winds*) > 1.0. Assuming both wind and temperature observation errors were correctly specified, we might conclude that both wind and temperature background errors were specified too low. In other instances, the wind and temperature $J_{\min}$ values might be mutually inconsistent, with one being greater than 1.0 and the other being less than 1.0. This is, of course, possible. It should be noted that for the same value of geopotential background error variance, the temperature variance will increase/decrease with decreasing/increasing vertical length scale and the wind variance will increase/decrease with decreasing/increasing horizontal length scale. Thus, we may be able to learn something about the specification of the vertical and horizontal length scales. This is clearly more delicate.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave Blank*)

2. REPORT DATE

August 2000

3. REPORT TYPE AND DATES COVERED

Final

4. TITLE AND SUBTITLE

NAVDAS Source Book 2000

5. FUNDING NUMBERS

Program Element No.
0601153N
Project No. BE-33-03-45

6. AUTHOR(S)

Roger Daley and Edward Barker

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
Marine Meteorology Division
Monterey, CA 93943-5502

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/PU/7530--00-418

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
Washington, DC 20375-5320

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

NAVDAS (NRL Atmospheric Variational Data Assimilation System) is a three-dimensional variational data assimilation suite for generating atmospheric state estimates to satisfy a variety of Navy needs. These needs range from global initial conditions for Navy global prediction models to environmental input to forward-deployed shipboard tactical decision aids. In common with many other Navy applications, the NAVDAS system has been designed to be robust, flexible, and portable. In particular, it can perform central site global assimilation on massively parallel machines as well as local data assimilation on workstations with the same code. NAVDAS is an observation space algorithm. The preconditioned conjugate gradient method is used as the descent algorithm to minimize the three-dimensional cost function. The number of iterations required to reach convergence is minimized through the use of dual block diagonal preconditioners with Choleski decomposition. Vertical eigenvector decomposition of the background error covariance matrix leads to great generality in formulating nonseparable error covariances as well as enormous efficiencies in handling vertical profile and sounding observations. Forward operators are formulated and used for the direct assimilation of TOVS radiances and SSM/I windspeeds and total precipitable water. NAVDAS also contains a complete diagnostic suite that includes complete observation trackability, Web-based observation monitoring, χ^2 monitoring of innovations, the adjoint of the assimilation system, and analysis error estimation.

14. SUBJECT TERMS

NAVDAS Source Book

15. NUMBER OF PAGES

155

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT

UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT

UNCLASSIFIED

20. LIMITATION OF ABSTRACT

SAR

Editor
Maureen Long
Naval Research Laboratory
Technical Information Division

Publication Design and Layout
Jan D. Morrow
Naval Research Laboratory
Technical Information Division

Reviewed and Approved
NRL/PU/7530--0-418
August 2000

P. Merilees

Philip E. Merilees
Superintendent,
Marine Meteorology Division