I present the derivation of the Preconditioned Optimizing Utility for Large-dimensional analyses (POpULar), which is developed for adopting a non-diagonal background error covariance matrix in nonlinear variational analyses (i.e., analyses employing a non-quadratic cost function). POpULar is based on the idea of a linear preconditioned conjugate gradient method widely adopted in ocean data assimilation systems. POpULar uses the background error covariance matrix as a preconditioner without any decomposition of the matrix. This preconditioning accelerates the convergence. Moreover, the inverse of the matrix is not required. POpULar therefore allows us easily to handle the correlations among deviations of control variables (i.e., the variables which will be analyzed) from their background in nonlinear problems.

In order to demonstrate the usefulness of POpULar, we illustrate two effects which are often neglected in studies of ocean data assimilation before. One is the effect of correlations among the deviations of control variables in an adjoint analysis. The other is the nonlinear effect of sea surface dynamic height calculation required when sea surface height observation is employed in a three-dimensional ocean analysis. As the results, these effects are not so small to neglect.

1. Introduction

Variational methods are widely adopted in objective data analysis of the atmosphere and ocean for assimilation systems, observational studies, and so on. The general consensus holds that it is essential to include the term of constraint on the difference between background (or first-guess) and analyzed values in a cost function in variational analyses (see, Lorenc, 1986; Courtier, 1997). If this constraint is adopted, the cost function necessarily includes the inverse of \( B \) as follows:

\[
J = \frac{1}{2} x^T B^{-1} x + J_{nl},
\]

where the column vector \( x \) is the increment of the state (or control) vector, i.e., \( x - x_f \) where \( x \) is the analysis state vector which will be optimized and \( x_f \) is their first-guess (or background). Matrix \( B \) is the background error covariance matrix. The second term \( J_{nl} = \frac{1}{2} (\mathcal{H}(x) - x_o)^T \mathbf{R}^{-1} (\mathcal{H}(x) - x_o) \) in (57) for an example) is a function of \( x \) which includes constraints imposed by observations or physical conditions. Inversion of \( B \) is thus inevitable if a conventional descent scheme (e.g., conjugate gradient and quasi-Newton methods) is adopted without any transformation of control variables \( x \). However, \( B \) always has too large a dimension to calculate the inverse in meteorology and oceanography. Because of this difficulty, the background constraint is often omitted from the cost function in oceanography (e.g., Wenzel et al., 2001). A diagonal matrix has also often been adopted as \( B \), even though control variables correlate with each other (e.g., Morrow and DeMey, 1995). The correlations, however, have important effects on analyses. For example, Derber and Bouttier (1999) improved their Numerical Weather Prediction (NWP) system by sophisticating the representation of the background error covariance matrix.

A solution to overcome the difficulty is diagonalizing \( B \). Fourier transformation is often adopted for this purpose for NWP systems (e.g., Heckley et al., 1992). The amplitudes of waves which have different wavelengths do not correlate with each other in linear theories. It is therefore reasonable to regard the background error covariance matrix in the spectral field as diagonal in order to make it easy to invert. This method, however, is
difficult when one seeks to introduce the inhomogeneity of the correlation fields. For this reason, homogeneous and isotropic assumptions are made about the background error correlations in NWP systems of most operational centers around the world. It is also difficult to use Fourier transformation for ocean analysis because the ocean has irregular lateral boundaries and the correlation field is extremely different in different areas (see, e.g., Kuragano and Kamachi, 2000). Empirical Orthogonal Function (EOF) modes are also used to diagonalize \( B \) in ocean analyses (e.g., Maes et al., 2000; Kaplan et al., 2000). The calculation and storage of the EOF modes, however, are hard tasks in the case where the matrix \( B \) has a large dimension. Moreover, there are usually not enough data to calculate valid EOF modes.

Some other methods for avoiding the inversion of \( B \) have been suggested, as summarized in Huang (2000). Physical-space Statistical Analysis Scheme (PSAS) was introduced by Cohn et al. (1998) in order to handle inhomogeneity of a first-guess error correlation field. They employed an exact Hessian matrix to seek a minimum for a cost function having a linear gradient, although certain approximation is made for a cost function with a nonlinear gradient. This scheme finds a minimum by only one iteration when the function has a linear gradient or by a few iterations even when the function has a nonlinear gradient. It does, however, require an inversion of \((BH^T + R)\) in each iteration, where \( H \) is a linearized observation operator and \( R \) is an observation error covariance matrix. This is a heavy computational burden, although the dimension of this matrix is usually much smaller than \( B \). Moreover, rapid convergence is not expected in the inversion of the matrix without special preconditioning.

Lorenc (1988) suggested adopting a transformed state vector \( \tilde{x} \) instead of \( x \) through \( x = B \tilde{x} \). By this transformation, the first term of the cost function (1) becomes \( \tilde{x}^T B \tilde{x} / 2 \). The inverse of \( B \) thus vanishes. Further preconditioning, however, is required to accelerate convergence.

Preconditioning is a technique to accelerate convergence by transforming control variables (see, Golub and Van Loan, 1996). Many descent schemes use the gradient information to decide the direction in which to search for the minimum state. If a Hessian of a linear cost function becomes identity matrix \( I \), the optimal value of the increment \( \tilde{x} \) becomes \(-\tilde{g}(0)\), where \( \tilde{g}(\tilde{x}) \) is the gradient of the function along \( \tilde{x} \). Therefore rapid convergence is expected if we can transform a function so that the Hessian is approximated by \( I \).

The matrix \( U^{-2} \) is called preconditioner because \( \tilde{g}(\tilde{x}) = U^{-2}g(x) \), where \( g(x) \) is the gradient of the cost function along \( x \).

Lorenc (1988) suggested performing a preconditioning by the transformation \( \tilde{x} = (\sqrt{B})^{-1}x \). In this case, the first term of the cost function (1) becomes \( \tilde{x}^T \tilde{x} / 2 \). If the first term is dominant in (1), the Hessian is approximated by \( I \), which offers us a rapid convergence. This is often true in large-dimensional data assimilation. See Courtier (1997). Moreover, the inversion of \( B \) is not required. It does, however, require \( \sqrt{B} \) and the calculation of \( \sqrt{B} \) is usually a hard task requiring a large memory.

Derber and Rosati (1989) (Hereafter DR89) adopted this transformation implicitly. Their scheme is based on the theory of a preconditioned conjugate gradient method. The scheme adopted iterative calculations. In each iteration, a search direction is defined by the conjugacy theory and a new \( x \) is set to a minimum point in the direction from the current \( x \). The scheme does not require \( B^{-1} \) nor \( \sqrt{B} \). Rapid convergence is also expected because of preconditioning. This scheme, however, cannot be adopted for a cost function having a nonlinear gradient. The variational method of DR89 has been adopted in ocean assimilation systems by some institutes, such as the National Centers for Environmental Prediction (NCEP) (see, Ji et al., 1995), Center for Ocean-Land-Atmosphere Studies (COLA) (see, Huang et al., 1999), and Japan Meteorological Agency (JMA) (Climate Prediction Division, 2002). In these systems, the nonlinear relation between temperature/salinity and Sea surface Dynamic Height (SDH) is substituted for a linear relation with the regression coefficient for assimilating the Sea Surface Height (SSH) data (see, Ji et al., 2000). Another descent scheme will be required in order to adopt the nonlinear calculation of SDH without any approximation.

Fujii and Kamachi (2003b) introduced the Preconditioned Optimizing Utility for Large-dimensional analyses (POpULar). In this paper, I present examples of applying POpULar to adjoint and oceanic 3DVAR analyses with the detailed derivation of the schemes in POpULar. POpULar is based on the idea of DR89, and three improvements have been made in order to adopt this idea for functions having both linear and nonlinear gradients.

First, the cost function (1) is divided into two parts, the first term bearing \( B^{-1} \) and the other part \( J_{nl} \). The procedure of DR89 can be adopted for the first term because its gradient is linear, that is, the gradient \( B^{-1}x (=e_i \) in Section 2 in this paper) can be calculated by a recursive equation without the inversion of \( B \). On the other hand, the gradient of \( J_{nl} \) is evaluated directly from \( x \).

Second, the method of a line search (i.e., the procedure to search a minimum point in a search direction from the current \( x \)) is changed for a nonlinear problem. The minimum point is exactly calculated using a linear property of the gradient in the method of DR89. However, it is difficult to find a minimum point exactly when the gradient is not linear. In POpULar, the point satisfying a certain condition is substituted for the minimum point, and found by evaluating the cost function at a few points. The
first term of the cost function \( x^T B^{-1} x / 2 (= K_\ell \) in Section 2: this value is not calculated in the original method of DR89) is also calculated for evaluating the function by a recursive equation without inversion of \( B \).

Third, the procedure defining a new search direction is improved. The procedure of DR89 for defining a search direction is only suitable for a linear problem with the accurate line search. The procedure is improved for a nonlinear problem under the concepts of nonlinear conjugate gradient and quasi-Newton methods.

Thus, POpULar can be adopted for a cost function having a nonlinear gradient. Lorenc’s (1988) transformation for preconditioning, that is, \( \tilde{x} = (\sqrt{B})^{-1} x \), is also adopted implicitly, and the schemes require neither \( B^{-1} \) nor \( \sqrt{B} \). Moreover, the schemes require only one operation of \( B \) in each iteration step.

This paper is organized as follows. The derivations of the schemes are described in Section 2. Next, I introduce two examples in order to demonstrate the usefulness of the schemes in Section 3. One is an adjoint analysis of a Burgers’ equation model. I examine effects of the correlations among background errors, i.e., off-diagonal elements of \( B \). The other is a three-dimensional variational (3DVAR) ocean analysis in the sea east of Japan, employing gridded analysis of Sea Surface Temperature (SST), TOPEX/POSEIDON (T/P) altimetry, and profiling float data. Here I examine the effect of the nonlinearity in the observation operator when the operator includes the conversion from temperature and salinity to SDH. It should be noted that the effects of the correlations and the nonlinearity are difficult to consider with a conventional descent scheme. I also compare efficiency among different options of POpULar. The results are summarized in Section 4.

2. Theoretical Background

2.1 Conjugate gradient method

POpULar is a utility for finding a minimum point of the cost function \( J \) in (1). Our interesting case in this paper is the case where \( J_{nl} \) is not a quadratic function, that is, the gradient of \( J \) along \( x \) is not linear. POpULar has two options: either of the conjugate gradient or quasi-Newton methods can be chosen. Derived are the conjugate gradient method in this subsection and the quasi-Newton method in the next subsection.

First, we introduce a positive definite symmetric matrix \( U \) which satisfies \( U^{-2} = B \). Next, \( \tilde{x} \) is defined as \( \tilde{x} = U x \) and we consider \( J \) as a function of \( \tilde{x} \). It is noted that preconditioning is performed here by adopting \( B \) as a preconditioner. The cost function is rewritten as

\[
J = \frac{1}{2} \tilde{x}^T \tilde{x} + J_{nl},
\]

and if the first term of this function is dominant, the Hessian matrix is approximated by \( I \). This transformation therefore brings rapid convergence, as discussed before.

A conjugate gradient method is adopted for finding \( \tilde{x} \) which minimizes \( J \) (see Navon and Legler, 1987, for a good review). The initial values are set as follows:

\[
\tilde{x}_0 = 0,
\]

\[
\tilde{g}_0 = g(\tilde{x}_0),
\]

\[
\tilde{d}_0 = -\tilde{g}_0.
\]

The sequence of calculations performed in each iteration is then as follows:

\[
\tilde{x}_k = \tilde{x}_{k-1} + \alpha_k \tilde{d}_{k-1},
\]

\[
\tilde{g}_k = g(\tilde{x}_k),
\]

\[
\beta_k = \frac{\tilde{g}^T \tilde{y}_k}{\tilde{d}_k^T \tilde{d}_k},
\]

\[
\tilde{d}_k = -\tilde{g}_k + \beta_k \tilde{d}_{k-1},
\]

where \( k \) is the iteration counter, initially set to 0, \( \tilde{g}(\tilde{x}) \) is the gradient of \( J \) along \( \tilde{x} \), \( \tilde{y}_k = \tilde{g}_k - \tilde{g}_{k-1} \), and \( \tilde{d}_k \) is known as a search direction. The coefficient \( \alpha_k \) is the positive value determined to minimize \( J(\tilde{x}_{k-1} + \alpha \tilde{d}_{k-1}) \). The task of searching \( \alpha_k \) is called line search. Equations (8) and (9) are the formulae which decide a search direction \( \tilde{d}_k \) according to the theory of conjugacy. It should be noted that these equations are originally derived on the assumption that \( \tilde{g}(\tilde{x}) \) is linear and that \( \alpha_k \) exactly minimizes \( J(\tilde{x}_{k-1} + \alpha \tilde{d}_{k-1}) \). The equations give an effective search direction if this assumption is satisfied. For a problem in which \( \tilde{g}(\tilde{x}) \) is nonlinear, the equations are still effective because any function can be locally approximated by a linear function. The efficiency, however, is rapidly degraded when the line search is not accurate. These properties are confirmed in Subsection 3.3.

Next, we rewrite (3)–(9) by \( x_k \), \( d_k = U^{-1} \tilde{d}_k \) and \( g_k = g(x_k) \), where \( g(x) \) is the gradient of \( J \) along \( x \), or \( \nabla_x J \), adopting the relations

\[
\begin{align*}
\tilde{x}_k &= U x_k, \\
\tilde{d}_k &= U d_k, \\
\tilde{g}_k &= U^{-1} g_k, \\
U^{-2} &= B,
\end{align*}
\]
At the same time, we introduce \( h_k =Bg_k \), \( K_k = x_k^T B^{-1} x_k / 2 \), \( c_k = B^{-1} x_k \) and \( e_k = B^{-1} d_k \). Finally, the initial values are set as follows:

\[
x_0 = 0, \quad K_0 = 0, \quad c_0 = 0, \quad g_0 = g_m(x_0), \quad h_0 = Bg_0, \quad d_0 = -h_0, \quad e_0 = -g_0,
\]

where \( g_m(x) \) is the gradient of \( J_m \) along \( x \). The sequence of calculations performed in each iteration is written as follows:

\[
x_k = x_{k-1} + \alpha_k d_{k-1}, \quad K_k = K_{k-1} + \alpha_k d_{k-1}^T \left( c_{k-1} + \alpha_k e_{k-1} \right), \quad c_k = c_{k-1} + \alpha_k e_{k-1}, \quad g_k = c_k + g_m(x_k), \quad h_k = Bg_k, \quad d_k = -h_k + \beta_k d_{k-1}, \quad e_k = -g_k + \beta_k e_{k-1},
\]

where \( y_k = g_k - g_{k-1} \). The coefficient \( \alpha_k \) minimizes \( J(x_{k-1} + \alpha d_{k-1}) \) where \( J \) is considered as a function of \( x \) again. For searching \( \alpha_k \) which minimizes \( J(x_{k-1} + \alpha d_{k-1}) \) by a line search, we can adopt the equations

\[
\begin{align*}
J(x_{k-1} + \alpha d_{k-1}) &= K_{k-1} + \alpha d_{k-1}^T \left( c_{k-1} + \frac{\alpha}{2} e_{k-1} \right) + J_m(x_{k-1} + \alpha d_{k-1}), \\
g(x_{k-1} + \alpha d_{k-1}) &= c_{k-1} + \alpha e_{k-1} + g_m(x_{k-1} + \alpha d_{k-1})
\end{align*}
\]

as the cost function and its gradient. The condition terminating the iteration is

\[
|g_k|^2 / |g_0|^2 < \varepsilon,
\]

where \( \varepsilon = 10^{-5} \) is adopted in this study. It is noted that satisfying this condition does not guarantee that the solution approximates to the optimum one, especially in the case with strong nonlinearity. Actually, there is no condition which guarantees that the solution is optimum. This condition is therefore often used for approximating the optimum solution together with adopting the background values as the first-guess (the start point) of a descent method (ex., Vialard et al., 2003) as in POpULar.

In the iterative calculations, \( x_k^T B^{-1} x_k = \langle K_k \rangle \) and \( B^{-1} x_k = \langle c_k \rangle \), which are the first term of \( J \) and its gradient, respectively, are calculated by the recursive equations (19) and (20) without the inversion of \( B \). Only one calculation of \( h_k = Bg_k \) in each iteration step is required. Moreover, preconditioning is performed by the implicit transformation of \( x \) to \( \hat{x} \): \( B \) is adopted as a preconditioner and rapid convergence is expected. This iterative optimizing scheme can apply for a nonlinear cost function because no assumption is made on its linearity.

In the conjugate gradient scheme, I also adopted a method of scaling which was introduced by Shanno and Phua (1978). Their scaling factor is \( \gamma_k = y_k^T p_k / y_k^T y_k \), where \( p_k = \hat{x}_k - \hat{x}_{k-1} = \alpha_d d_{k-1} \). In order to adopt this factor in our scheme, the factor is rewritten using the relations (10) as follows:

\[
\gamma_k = y_k^T p_k / y_k^T z_k,
\]

where \( p_k = x_k - x_{k-1} = \alpha_d d_{k-1} \) and \( z_k = h_k - h_{k-1} \). Finally, I used the scaled search direction vector \( \tilde{d}_k = \gamma_k d_k \) instead of \( d_k \) itself with setting the first-guess of \( \alpha_d \) in the line search equal to unity. This scaling makes it possible to implement an accurate line search in a few calculations of \( J \) and \( g(x) \). The description of the line search will be seen again in Subsection 2.3.

Because the conjugacy theory assumes that the gradient of the cost function \( g(x) \) is linear and that the line search is performed exactly, Eqs. (23) and (24) generate an inefficient search direction after a few iterations if the assumptions are not satisfied. In this case, we have to restart the iteration by setting \( d_k = -h_k \) (the steepest descent direction under preconditioning). Powell (1977) has suggested a condition for restarting in conjugate gradient algorithms. It is \( |g_{k-1}^T g_k| > 0.2 |g_{k-1}^T g_{k-1}| \). In the system, this is rewritten by using the relations (10) again as follows:

\[
|h_{k-1}^T g_k| > 0.2 |h_{k-1}^T g_{k-1}|
\]

170 Y. Fujii
This condition was adopted for the conjugate gradient scheme in POpULar.

Beale (1972) suggested a more efficient restart method than setting \( d_k = -h_k \). It can be incorporated in POpULar, but it did not improve the efficiency of the scheme when I checked it. The restart procedure will not be discussed further in this paper.

To finish this subsection, we consider the case where \( g_n(x) \) is linear and the Hessian of \( J_n \) is known. In this case, the preconditioned conjugate gradient method is reduced to the method introduced by DR89. The coefficient \( \alpha_k \) can be written as

\[
\alpha_k = -\frac{g_k^T d_k}{d_k^T f_k},
\]

where

\[
f_k = e_k + A d_k
\]

and \( A = H^T R^{-1} H \) in DR89 is the Hessian matrix of \( J_n \), which is independent of \( x \). We can also redefine \( \beta_k \):

\[
\beta_k = -\frac{g_k^T g_k}{g_{k-1}^T g_{k-1}} = \frac{g_k^T h_k}{g_{k-1}^T h_{k-1}}
\]

(see, Navon and Legler, 1987), where

\[
g_k = g_{k-1} + \alpha_k f_{k-1}
\]

is substituted for (21). It is not required to calculate \( K_k \) and \( e_k \) in each iteration. Finally, we can adopt Eqs. (11) and (14)–(17) as the initial values and Eqs. (32), (31), (18), (34), (22), (33), (24) and (25) as the calculations of each iteration, when \( g_n(x) \) is linear and \( A \) is known.

### 2.2 Quasi-Newton method

The quasi-Newton scheme of POpULar is derived in this subsection. Although conjugate gradient methods are sensitively degraded by nonlinearity and inaccurate line search, as discussed before, quasi-Newton methods are more insensitive to them. They are therefore more suitable for a large scale problem which has a strong nonlinearity (Liu and Nocedal, 1989).

In a quasi-Newton method using BFGS formula with limited memory (see Nocedal, 1980; Liu and Nocedal, 1989), equations updating a search direction are written as follows:

\[
\mathbf{H}_{k-m} = \gamma_k \mathbf{I},
\]

(35)

\[
\mathbf{H}_{k,l} = \mathbf{V}_{k,l} + \mathbf{H}_{k,l-1} \mathbf{V}_{k,l}^T + \mathbf{p}_{k+l} \mathbf{p}_{k+l}^T,
\]

(36)

\[
\mathbf{d}_k = -\mathbf{H}_k \mathbf{g}_k,
\]

(37)

where \( \mathbf{p}_k = 1/\gamma_k \mathbf{p}_k \), \( \mathbf{V}_k = \mathbf{I} - \mathbf{p}_k \mathbf{p}_k^T \), \( l \) is the counter for the iteration to calculate an approximate inverse of the Hessian matrix \( \mathbf{H}_{k,l} \), and initially set to \( -m \), in which the number \( m \) is related to the number of vectors required to store; this scheme uses the information obtained before \( m \) iterations. It is noted that Shanno and Phua’s (1978) scaling is adopted in the initial condition (35). When these equations are rewritten by using the relations (10), they do not change the forms (only tildes are removed) but the initial condition (35) is changed to

\[
\mathbf{H}_{k,-m} = \gamma_k \mathbf{B},
\]

(38)

where \( \mathbf{H}_{k,l} = \mathbf{U}^{-1} \mathbf{H}_{k,l} \mathbf{U}^{-1} \). It should be noted that \( \mathbf{H}_{k,l} \) is different from a linearized observation operator \( \mathbf{H} \). If the reader notes the relations \( \mathbf{B} \mathbf{g}_k = \mathbf{h}_k \) and \( \mathbf{B} \mathbf{y}_k = \mathbf{z}_k \), the equations defining a search direction (37) with (36) and (38) can be transformed as follows:

\[
\mathbf{d}_k = -\gamma_k \mathbf{h}_k + \sum_{l=-m+1}^{k} (a_{z,k} \mathbf{y}_l + a_{p,k} \mathbf{p}_l),
\]

(39)

where \( a_{z,k} \) and \( a_{p,k} \) are scalar coefficients which can be defined from \( \mathbf{y}_{k-m+1}, \mathbf{y}_{k-m+2}, \ldots, \mathbf{y}_k, \mathbf{z}_{k-m+1}, \mathbf{z}_{k-m+2}, \ldots, \mathbf{z}_k, \mathbf{p}_{k-m+1}, \mathbf{p}_{k-m+2}, \ldots, \mathbf{p}_k, \mathbf{g}_k \), and \( \mathbf{h}_k \) without operating \( \mathbf{B} \) or its inverse (see Appendix). If we define \( \mathbf{q}_l = \mathbf{B}^{-1}(\mathbf{x}_l - \mathbf{x}_{k-l}) = \alpha_l \mathbf{e}_{l-1} \), the updating equation of \( \mathbf{e}_k = \mathbf{B}^{-1} \mathbf{d}_k \) is written as

\[
\mathbf{e}_k = -\gamma_k \mathbf{g}_k + \sum_{l=-m+1}^{k} (a_{z,k} \mathbf{y}_l + a_{p,k} \mathbf{q}_l).
\]

(40)

If (39) and (40) are substituted for (24) and (25) in the iterative calculations (18)–(25), we obtain a new quasi-Newton scheme without the inversion of \( \mathbf{B} \).

Practically, \( \mathbf{d}_k \) and \( \mathbf{e}_k \) are calculated iteratively as follows:

\[
\mathbf{s}_0 = -\mathbf{h}_k,
\]

(41)

\[
\mathbf{t}_0 = -\mathbf{g}_k,
\]

(42)

\[
\mathbf{s}_{l+1} = \mathbf{s}_l - \rho_{k+l} \mathbf{t}_l \mathbf{p}_{k+l}, \quad (l = 0, -1, \ldots, -m+1),
\]

(43)

\[
\mathbf{t}_{l+1} = \mathbf{t}_l - \rho_{k+l} \mathbf{t}_l \mathbf{y}_{k+l}, \quad (l = 0, -1, \ldots, -m+1),
\]

(44)
\[ s^\prime_m = \gamma s_m, \]  
\[ t^\prime_m = \gamma t_m, \]  
\[ s_l^\prime = s_{l-1}^\prime + \rho_{k+l} (t_{l}^T p_{k+l} - s_{l-1}^T y_{k+l}) p_{k+l} \]  
\[(l = -m+1, -m+2, \ldots, 0), \]  
\[ t_l^\prime = t_{l-1}^\prime + \rho_{k+l} (t_{l}^T p_{k+l} - s_{l-1}^T y_{k+l}) q_{k+l} \]  
\[(l = -m+1, -m+2, \ldots, 0), \]  
\[ d_k = s^\prime_0, \]  
\[ e_k = t^\prime_0. \]

A supplementary explanation of these equations is presented in Appendix. Then, \(d_k\) and \(e_k\) calculated in this way are adopted in the iterative calculations (18)–(25) instead of \(d_k\) and \(e_k\) in (24) and (25).

Rapid convergence is expected with the quasi-Newton method because of preconditioning by adopting \(B\) as a preconditioner, and the inversion of \(B\) is not required. Moreover, an operation of \(B\) is required only once in each iteration step. This scheme also works more effectively than the conjugate gradient method for a strong nonlinear function with an inaccurate line search. Thus this scheme has many desirable aspects for large-dimensional variational analyses such as in the atmosphere and ocean.

Furthermore, Shanno (1978) suggested a scheme which combines ideas of a quasi-Newton method and a restart. Although I tried this idea, the efficiency is no better than the quasi-Newton method in the experiments described in Section 3. I do not discuss this further.

2.3 Line search

It is difficult to find \(\alpha_k\) which perfectly minimizes \(J(x_{k-1} + \alpha d_{k-1})\) when the gradient of \(J\) is nonlinear, although it can be calculated exactly by (31) when the gradient of \(J\) is linear. Especially in the case where it takes a long time to calculate the cost function and its gradient, an accurate line search is hardly practical because it requires calculating the cost function and its gradient many times. Therefore, instead of the accurate value of \(\alpha_k\), we often adopt a value that satisfies some conditions. POpUlAr has two options for the line search procedure.

One is a moderate line search procedure with quadratic interpolation. The procedure is suitable for the quasi-Newton method because the method does not require an accurate line search. The conditions for \(\alpha_k\) are

\[ \alpha_k > 0, \]  
\[ J(x_{k-1} + \alpha_k d_{k-1}) < J(x_{k-1}) + \alpha_k d_{k-1}^T g_{k-1} b_1, \]  
\[ b_1 = 10^{-4}. \]

where \(b_1 = 10^{-4}\). The value satisfying the condition is searched in POpUlAr as follows. If the \(j\)-th guess \(\alpha_{k,j}\) does not satisfy the condition, the next guess \(\alpha_{k,j+1}\) is calculated with quadratic interpolation:

\[ \alpha_{k,j+1} = \frac{\sqrt{2} \alpha_{k,j} d_{k-1}^T d_{k-1} + J(x_{k-1}) - J(x_{k-1} + \alpha_{k,j} d_{k-1})}{\alpha_{k,j} d_{k-1}^T d_{k-1}}, \]

and the condition is examined again. It should be noted that the gradient for the guess of \(\alpha_k\) is not required for examining the condition and calculating the next guess. Calculation of the gradient requires a great deal of time in adjoint analyses because of the inclusion of the adjoint model integration. The Wolfe condition (see Navon and Legler, 1987) with cubic interpolation (Davidon, 1959) is often adopted for a moderate line search. It requires the calculation of the gradient for every guess of \(\alpha_k\). The method introduced here can save computation time by avoiding the heavy calculation of the gradient for the guess of \(\alpha_k\).

The other is a precise line search procedure with cubic interpolation. This procedure is suitable for a conjugate gradient scheme. The conditions for \(\alpha_k\) are (51), (52) and

\[ \left| \frac{d_{k-1}^T g(x_{k-1} + \alpha d_{k-1})}{d_{k-1}^T g_{k-1}} \right| < b_2, \]

where \(b_2 = 10^{-2}\).

Except for the first step, the first-guess \(\alpha_{k,0}\) is set to unity because the search direction is scaled by \(\gamma_k\). The search direction for the first step is not scaled, and it is desirable to adopt (31). The Hessian matrix of \(J_{nl}\) (i.e., \(A\)), however, is not constant and it takes a great deal of time to calculate the Hessian exactly if the gradient of \(J_{nl}\) is not linear. I adopted the approximation of the Hessian calculation suggested by Cohn (1997) in order to adopt (31) to the first step.

3. Examples

3.1 Theoretical experiments on adjoint method using a nonlinear model

In this subsection I illustrate the importance of correlations among each background error, carrying out a simple theoretical experiment adopting a nonlinear model of Burgers’ equation. This can easily be done by adopting POpUlAr because they do not require to calculation of the inverse of the first-guess error covariance matrix \(B\).
Burgers’ equation is
\[
\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}.
\]  
(55)

It describes a one-dimensional advection-diffusion process over a spatial domain \((-\infty, \infty)\). The numerical model is defined as a finite difference leapfrog scheme, with a forward step every 15 time steps. The numerical model which was adopted in this study is the same model as used in former studies (Uboldi and Kamachi, 2000; Zhu and Kamachi, 2000). The computational space domain is \([-1000, 1000]\) km, but only solutions within \([-100, 150]\) km are considered. The large domain was adopted to reduce the boundary effects on interior solution. I set \(\Delta t = 60\) s, \(\Delta x = 5\) km, \(\nu = 1.0 \times 10^4\) m\(^2\)s\(^{-1}\).

The shape of initial state is defined as follows:
\[
u(x, 0) = \begin{cases} 
0 & \text{if } x < -L; x > L \\
u_0(1 + \frac{2x}{L})(1 - \frac{L}{x})^2 & 0 \leq x < L \\
u_0(1 - \frac{2x}{L})(1 + \frac{L}{x})^2 & -L < x \leq 0 
\end{cases}
\]  
(56)

where \(L = 50\) km. I first created a pseudo true state by integrating Burgers’ equation using (56) as the initial state in which \(\nu_0\) was set to 5 m/s. Pseudo observation data was created by picking values from this true state and adding random observation errors which had a Gaussian distribution with a variance of \((1\) m/s\(^2\)). It is assumed that the observations are available every 25 km between \([-100, 150]\) km and at hour 1, 4, 7 and 10. Finally I estimated an optimal initial state by assimilating the observation data into Burgers’ equation model. Here the initial value of \(\nu\) at grid points is considered as control variables and (56) with \(\nu_0 = 2.5\) m/s was used as the first-guess.

The cost function for the assimilation is
\[
J = \frac{1}{2} x^T B^{-1} x + \frac{1}{2} (\mathcal{H}(x) - x_o)^T R^{-1} (\mathcal{H}(x) - x_o),
\]  
(57)

where \(\mathcal{H}(x)\) is the difference between the true state and the first-guess at \(x\) at the initial. The calculated covariance function is shown in Fig. 1. I illustrate the difference between the two estimations; one is using the complete covariance matrix \(B\), and the other is using only the diagonal elements of \(B\) and setting off-diagonal elements equal to 0.

The results of the two estimations are shown in Fig. 2. It is notable that the errors of estimations are much smaller than the observation errors. The average error is reduced from 0.98 m/s to 0.76 m/s and 0.54 m/s on the analyses with the diagonal and non-diagonal (complete) covariance matrices, respectively. This means that the adjoint method is an efficient method. The peaks of both estimations move forward. This is because the advection is not estimated adequately since the estimated speeds are smaller than the true state.

Comparing the two estimations, it is apparent that the estimation with the non-diagonal covariance matrix is better than that with the diagonal covariance matrix. For example, the peak of the estimation with the non-diagonal matrix is higher and the variation is sharper. They are closer to the true state than the estimation with the diagonal matrix. While the adjoint equation is integrated, or \(\mathbf{H}^T\) is calculated, the information of the difference between analysis and observation spreads spatially. If the
diagonal covariance matrix is adopted, the error information in each grid point that has been spread by the adjoint operator is regarded as different individual signals and it does not have such large effects on the increment. However, if the non-diagonal covariance matrix is adopted, the information in each grid point spread by the adjoint operator is regarded as some shaped coherent signal and it effectively gives a great improvement to the analysis field. Moreover, the error information is redistributed according to the covariance function if the adequate first-guess covariance matrix $B$ is adopted.

It seems that the analysis field with the non-diagonal covariance matrix is adopted, the error information in each grid point that has been spread by the adjoint operator is regarded as different individual signals and it does not have such large effects on the increment. However, if the non-diagonal covariance matrix is adopted, the information in each grid point spread by the adjoint operator is regarded as some shaped coherent signal and it effectively gives a great improvement to the analysis field. Moreover, the error information is redistributed according to the covariance function if the adequate first-guess covariance matrix $B$ is adopted.

Fig. 2. The result of the Burgers’ equation experiment. (a) 0 h, (b) 1 h, (c) 4 h, (d) 7 h, (e) 10 h. Blank circle: true state, filled circle: first-guess, blank (filled) square: the estimation using diagonal (non-diagonal) first-guess error covariance matrix, cross: observation.
nal matrix has larger pseudo wavy signals at grids without any wave in the true state than the one with the diagonal covariance matrix. Adopting the diagonal matrix has the danger of regarding a pseudo signal due to an observation error as a coherent signal. It should be noted, however, that the pseudo signal is much smaller than observation errors. Finally, off-diagonal elements of $B$ are important to pick up the observed information effectively and to redistribute it adequately.

3.2 Experiments on 3DVAR ocean analysis

This section examines the effects of nonlinearity of an observation function (the relation between temperature/salinity and SDH) in a 3DVAR ocean temperature and salinity analysis. The analysis method is described in Fujii and Kamachi (2003b). Amplitudes of vertical coupled Temperature-Salinity (T-S) EOF modes are employed as the control variables, and the cost function is expressed in the same form as (57). Correlations among the same modes at different grid points are considered. The correlation field is supposed to have Gaussian distribution which is isotropic and homogeneous. The decorrelation scale is set to 90 km. This value is calculated from T/P altimetry data (Kuragano and Kamachi, 2000). Correlation between different modes is neglected.

In order to adopt SSH observation in the analysis, the calculation of SDH from the grid values of temperature and salinity is included in the observation function $\mathcal{H}$. I calculated SDH as follows:

$$h = -\frac{1}{\rho_s} \int_0^{z_m} \rho'(T, S, p)dz,$$

(59)

where $h$ is the difference of SDH from a reference state, $\rho_s$ is the surface density, $z_m$ is the reference depth, $\rho'$ is the difference of the density from the reference state, $z$ denotes the vertical coordinate, and $T$, $S$ and $p$ denote temperature, salinity and pressure, respectively. The density was calculated according to the nonlinear relation between $T$, $S$, $p$, and density by the UNESCO (1981) formula. I chose the state of 0°C and 35 psu as the reference state, and the depth of 1500 m as the reference depth. It is noted that this calculation has a nonlinearity. Two analyses are performed in order to examine the effect of the nonlinearity. One adopts the exact expression of the nonlinear observation operator $\mathcal{H}$ (nonlinear analysis). The other adopts the tangent linear operator $\mathcal{H}$, which is expanded around the first-guess instead (linear analysis).

The monthly climatology of Kuragano and Kamachi (1997) is adopted as the first-guess. The horizontal grid spacing in the analysis is 0.25° with 24 vertical levels (0, 10, 20, 30, 50, 75, 100, 125, 150, 200, 250, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200, 1300, 1400, 1500 m). The statistics are calculated from temperature and salinity profile data of the World Ocean Database 1998 (WOD98, Conkright et al., 1998) in the same man-

Fig. 3. The analyzed 100 m temperature (°C) fields in June, 2001 with nonlinear observation operator. The position of profiling float data is also shown by crosses.

Fig. 4. The analyzed a) temperature (°C) and b) salinity (psu) fields of the 144°E section. Solid (dashed) line is the estimation with the nonlinear (linear) operator.
ner as Fujii and Kamachi (2003b). Profiling float data on the Global Telecommunications System (GTS) is employed with T/P altimeter data (Kuragano and Shibata, 1997) and monthly mean of SST analysis produced by JMA in the constraint of observation. The analysis was performed for every month of 2001 in the ocean east of Japan, 30–45°N and 140–160°E.

I show first the 100 m depth temperature field of the nonlinear analysis in June, 2001 with the distribution of profiling float data in the same period in Fig. 3. The Kuroshio front is sharply obtained at 37.5°N, 144°E and branches out south and north of 36°N, 147°E. The triple frontal structure (Iwao et al., 2003) is clearly expressed at the east of 150°E. Some warm and cold eddies are also obtained, including a warm one at 41°N, 147°E and a cold one at 34°N, 145°E.

The 144°E section of the analysis fields in June, 2001 are shown in Figs. 4 and 5 with the section of the observation field by Kofu-maru, the observation vessel of the Hakodate Marine Observatory in JMA, in the same period in Fig. 6. The solid (dashed) line shows the nonlinear (linear) analysis in Fig. 4.

In general, a large scale distribution of temperature and salinity in Fig. 6 is depicted satisfactorily in Fig. 4, although small scale structures are removed. The Oyashio water is expressed in the north of 40°N, where there is cold water in the subsurface layer, and sea surface salinity is less than 33.0 psu. The low salinity (less than 33.8 psu) around 300 m depth between 38–40°N is not expressed in the analyses. The Kuroshio front is analyzed at 37.5°N, although it tilts and broadens out into the south more than the observed field. The anomaly of the cold eddy seen at 34°N, 145°E in Fig. 3 is underestimated because of the shortage of T/P altimetry and profiling float data, and the warm tropical water at the south of the Kuroshio front shifts to the south. The salinity of the subtropical water is analyzed too high, the low salinity area of the North Pacific Intermediate Water (NPIW) broadens out, and the low salinity less than 34.0 psu in the cold eddy is not analyzed. The difficulty of expressing low salinity variability in the mixing area and of the NPIW is a major shortcoming of this analysis method (see also Fujii.

Fig. 5. The absolute value of the linear estimation error minus the absolute value of nonlinear estimation. a) temperature (°C), b) salinity (psu). The negative area is shaded.

Fig. 6. The observed a) temperature (°C) and b) salinity (psu) fields of the 144°E section.
It is noted that these analyses aim to express the large scale field in which the small scale variabilities are averaged rather than to describe them. The data employed in the analyses is also not sufficient to express these small scale structures. Especially the profiling float observations are too sparse and biased near the Kuroshio Front, as shown in Fig. 3, although they improved the position and structure of the Kuroshio front in the 144°E section of the analysis field (not shown).

In comparison between the nonlinear and linear analyses, the estimated temperature and salinity of the nonlinear analysis is lower than that of linear analysis in the warm subtropical water and the cold eddy. Figure 5 shows the distribution of the difference, which is the absolute value of the linear analysis error minus the absolute value of the nonlinear analysis error. It is clear that the nonlinear analysis has smaller errors than the linear analysis in these areas, although it has larger error in the Kuroshio frontal zone where the scale of variability is too small to analyze. This is because the nonlinear operator more precisely expresses SDH than the linear one. The first EOF mode represents the variability such that whole water column is warm or cold in this region. The higher the temperature of the water column, the more sensitive to SDH is the variation of temperature. This is not expressed by the linear operator (see also Fujii and Kamachi, 2003a). The first mode also represents the coupling between warm (cold) and high (low) salinity anomaly. Therefore, the linear analysis overestimates (underestimates) a warm (cold) deviation, accompanied by high salinity bias. It should be noted that the linear analysis is 1°C and 0.1 psu worse at maximum than the nonlinear analysis. That is enough to affect the dynamics of the ocean. This improvement by the nonlinear operator has already been confirmed by Fujii and Kamachi (2003b) in the case where only SST and T/P altimetry data is employed as observations. This effect still requires serious attention when profiling float data is employed with SST and T/P altimetry data.

The profiles of reduced ratio (i.e. the root mean square error of the analysis referred to the observed values normalized by that of the first-guess or climatology, see Fujii and Kamachi, 2003a) are shown in Fig. 7. These parameters are calculated from the temperature and salinity observation data on GTS in 2001 excluding profiling float data employed in the analyses. These profiles show that more than 70% of errors of the first-guess remain after the analysis. One reason for this is the shortage of observations. The distribution of the profiling float data in Fig. 3 is too sparse compared to the scale of variability in this area (about 90 km according to the estimation of Kuragano and Kamachi, 2000). The distance of the T/P path is about 300 km and also much larger than the scale of variability. The profiling float data improves the analysis temperature at all depths and the analysis salinity at the near surface and deeper than 600 m layers. The decreases of the temperature errors shallower than 300 m are, however, small and the salinity analysis becomes worse at the subsurface layer. This is because smaller scale variability is predominant in the upper ocean. Near surface salinity is improved because it is difficult to analyze without salinity observation (Fujii and Kamachi, 2003a).

It is clear that the nonlinear analysis has smaller temperature error than the linear analysis, although the difference is small in the upper ocean where errors caused by smaller variability are dominant. The salinity errors of the nonlinear analysis are also smaller than those of the linear analysis excluding the layer between 200–500 m where difficulty of low salinity estimation degrades the analysis fields. Although considering nonlinearity thus improves the analysis field, the errors caused by the shortage of observations or the shortcomings of the analysis method are dominant and these eventually make the difference between the nonlinear and linear analysis small in this experiment. It is, however, noted that the difference will be remarkable when profiling float and altimetry data...
data becomes more dense or the analysis method is more sophisticated.

3.3 Efficiency comparison

In order to compare the efficiencies of the schemes in POpULar, calculations with some different schemes were implemented in the experiments mentioned in Subsections 3.1 and 3.2. It should be noted that the efficiency depends on the function to be minimized. This comparison, however, offers an insight for choosing a suitable scheme in each situation. The result is shown in Table 1, where SSD denotes the preconditioned steepest slope descent scheme which is performed by substituting \( d = -\gamma_k h \) for (24), CG means the conjugate gradient scheme and QNI (QN5) means the quasi-Newton scheme which calculates the approximate inverse of the Hessian from the former (5) gradient and search direction vectors in each iteration step. DR means the scheme for a linear function, introduced by DR89. We can consider that the results almost completely converge to the same solutions because the differences between results in different calculations are at least 5 orders of magnitude smaller than the analyzed values themselves.

The table shows the numbers of iterations, calculating \( \mathcal{H} \) or \( H \), and calculating \( H^T \). A calculation of matrix \( B \) is performed at every iteration step. The calculations of \( \mathcal{H} \) and \( H^T \) are included in those of \( J_{nl} \) and \( g_{nl} \), respectively. The calculation of \( A (=H^T R^{-1} H) \) in (32) includes the calculations of \( H \) and \( H^T \). The calculations of \( \mathcal{H} \), \( H \) and \( H^T \) include the integrations of model equations, its tangent linear code, and its adjoint codes, respectively, in a general adjoint analysis like the experiment discussed in Subsection 3.1. Although model integrations are very a heavy computational burden, adjoint model integrations are usually far heavier. Therefore, the calculation of \( g_{nl} \) is usually heavier than that of \( J_{nl} \) in adjoint analyses. The time required for the tangent linear model integration is usually similar to that required for the full model integration. The calculation of \( B \) is also heavy because of its dimension. For a 3DVAR analysis like the experiment discussed in Subsection 3.2, in which \( \mathcal{H} \) does not include a model equation, a calculation of \( B \) often becomes the main part of cpu time expenditure. It should be noted that the calculation that is responsible for the main part of cpu time crucially affects the time efficiency of the descent schemes.

The number of calculations of \( \mathcal{H} \) or \( H \) is equal to the number of \( \mathcal{H} \) or \( H \) when using the precise line search because the gradient \( g_{nl} \) has to be calculated for every guess of \( \alpha_k \) together with the cost function. The number is reduced, though, when using the moderate line search because the calculation of \( g_{nl} \) is required only once

<table>
<thead>
<tr>
<th>burgers’ equation</th>
<th>3DVAR problem</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Diagonal</td>
</tr>
<tr>
<td>moderate SSD</td>
<td>13/17/15</td>
</tr>
<tr>
<td>(1.3/1.4/1.3)</td>
<td>(1.4/1.4/1.4)</td>
</tr>
<tr>
<td>CG</td>
<td>17/20/19</td>
</tr>
<tr>
<td>(1.7/1.7/1.6)</td>
<td>(1.6/1.6/1.6)</td>
</tr>
<tr>
<td>QNI</td>
<td>15/17/17</td>
</tr>
<tr>
<td>(1.5/1.4/1.7)</td>
<td>(1.1/1.1/1.1)</td>
</tr>
<tr>
<td>QN5</td>
<td>10/12/12</td>
</tr>
<tr>
<td>(1.0/1.0/1.0)</td>
<td>(1.0/1.0/1.0)</td>
</tr>
<tr>
<td>(1.9/3.3/3.3)</td>
<td>(3.1/5.3/5.6)</td>
</tr>
<tr>
<td>CG</td>
<td>10/21/21</td>
</tr>
<tr>
<td>(1.0/1.8/1.8)</td>
<td>(0.9/1.6/1.7)</td>
</tr>
<tr>
<td>QNI</td>
<td>10/21/21</td>
</tr>
<tr>
<td>(1.0/1.8/1.8)</td>
<td>(1.1/2.0/2.1)</td>
</tr>
<tr>
<td>QN5</td>
<td>9/19/19</td>
</tr>
<tr>
<td>(0.9/1.6/1.6)</td>
<td>(0.9/1.6/1.7)</td>
</tr>
<tr>
<td>DR</td>
<td>39/40/40</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
in each iteration for the decidedly value of $\alpha_k$ (see Subsection 2.3).

SSD has an unexpectedly good performance with the moderate line search. This is because the preconditioning by $B$ is successful. The convergence rate, however, becomes much worse with the precise line search. CG shows a good performance with the precise line search. The number of iterations with CG is small and almost the same as the number with DR89, although the other two numbers are about double and larger than those with DR89. The precise line search requires calculating the cost function and its gradient about twice in each step. On the other hand, DR89 evaluates the cost function and its gradient only once at the first step and performs the calculation of $A$ (including the calculation of $H$ and $H^T$) once in each step instead of calculating the cost function and its gradient. Considering the small number of iterations, CG with the precise line search is useful if only small memory is available or the calculation of $J_{nl}$ is not very heavy compared to the calculation of $B$, because it is simpler and requires smaller memory than the Quasi-Newton method. With the moderate line search the performance of CG is not good. This is because CG is the scheme derived under the assumption of the perfect line search: the conjugated direction is not an effective search direction without the precise line search (Navon and Legler, 1987), and the restart procedure is performed once in a few steps.

The more memory that is employed, the better the performance given by QN with the moderate line search. QN5 with the moderate line search requires the fewest calculations of the cost function and its gradient. The number of iterations is not so great as CG with the precise line search, too. If it takes a long time to calculate $J_{nl}$ or its gradient and much memory is available, QN5 is the best scheme among those examined here. With the precise line search, the performance of QN is almost the same as CG. This is because QN reduced to CG with the perfect line search when the cost function is quadratic (see, Shanno, 1978; Navon and Legler, 1987). The number of iterations of QN is not much reduced compared with the one using the moderate line search. There seems to be no necessity to use QN with the precise line search, as far as the problems solved in this study is concerned. It may be effective when a more complex function is concerned, though.

4. Summary

The scheme of POoULar is derived in this paper. It is based on the idea of DR89. POoULar can, however, be adopted to minimize a cost function whose gradient is not linear. Rapid convergence is expected with POoULar because POoULar adopted a background error covariance matrix $B$ as a preconditioner. POoULar does not require the inversion of $B$. Moreover, it requires only one calculation of $B$ in each iteration step.

POoULar is useful for overcoming some kinds of difficulty which confront us when we sophisticate a variational analysis. Even if we try to consider the correlation among increments of control variables, the inversion of $B$ is almost impossible because $B$ always has a very large dimension. POoULar solves this problem because it does not require the inversion of $B$.

Preconditioning is also important to accelerate the convergence, as mentioned in Section 1. Especially this is often emphasized nowadays. It is a reasonable idea to adopt $B$ as a preconditioner. POoULar adopts $B$ as a preconditioner without requiring any other matrix like $B$.

The method of DR89 has been widely used for 3DVAR global ocean assimilation systems. If POoULar substitutes for the method of DR89, SSH observations are more effectively utilized with a nonlinear observation operator which expresses the nonlinearity of the relation between temperature/salinity and SDH.

In order to demonstrate the usefulness of POoULar, two analysis experiments are demonstrated. One is an adjoint analysis using a Burgers’ equation model. I examined effects of correlations among increments of control variables, and confirmed the importance of them. The other is a 3DVAR analysis of oceanic temperature and salinity fields east of Japan using vertical coupled T-S EOF modes. SST, SSH, and profiling float observation data are employed in the analysis. I examined the difference between the estimations with the nonlinear observation operator including nonlinear SDH calculation and the linear observation operator including the linear approximation of SDH. The estimation with the nonlinear operator is better than that with the linear operator. The difference seems to be large enough to affect the ocean physics.

I also examined the efficiency among schemes in POoULar. If not much memory is available and it takes a short time to calculate $J_{nl}$ compared with a calculation of $B$, the conjugate gradient method with the precise line search is most suitable. If much memory is available or it takes a long time to calculate $J_{nl}$, as in adjoint analyses, the quasi-Newton scheme with the moderate line search is recommended as the best option.

POoULar is to be adopted in the new global and western North Pacific ocean assimilation systems of the Meteorological Research Institute in JMA. POoULar makes it easy to handle the inhomogenous horizontal model error correlations with the nonlinear observation operator required for assimilating SSH data effectively. POoULar is available from the author to any users for research.
Acknowledgements
The temperature and salinity profile data on GTS including that in 144°E section observed by Kofu-maru in the Hakodate Marine observatory and the gridded analyzed SST data were kindly provided by JMA. WOD98 was also kindly provided by Japan Oceanographic Data Center. I thank Dr. T. Kuragano for providing me the TOPEX/POSEIDON altimeter data. I also thank anonymous reviewers for helpful comments. Part of this study was supported by the Category 7 of MEXT RR2001 Project for Sustainable Coexistence of Human, Nature and the Earth and by Grant-in-Aids for Science Research 16740272 from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

Appendix: Supplementary Explanation in the Introduction of the New Quasi-Newton Scheme
I first present a supplementary explanation of Eqs. (41)–(49). We define $t_l$ as

$$ t_0 = -g_k. $$

$$ t_l = V_{k+l} \cdots V_k t_0 \qquad (l = -1, -2, \cdots, -m). $$

Then $t_l$ is calculated by (42) and (44). Also, we define $s_l = Bt_l$, then $s_l$ is calculated by (41) and (43). Moreover, we define $s'_l = H_k t_l$, then (45) is derived from (38). If (36) is used with (A1) and (A2), we obtain

$$ s'_l = V_{k+l} t'_l s'_{l-1} = \rho_{k-l} t'_l \rho_{k+l} t_{k+l}, $$

and (47) is derived. Finally,

$$ d_k = -H_{k,0} g_k = H_{k,0} t_0. $$

and (49) is derived. We also need $e_k = B^{-1} d_k$. If we define $t'_l = B^{-1} s'_l$, (46), (48) and (50) are derived after manipulations.

Next we derive coefficients $a_{c,k,j}$ and $a_{p,k,j}$ in (39) and (40). We put

$$ t_l = -g_k + \sum_{i=l+1}^{k} a_{y,k,i} y_i \qquad (l = -m, -m + 1, \cdots, -1). $$

If it is substituted into (44), we obtain

$$ a_{y,k,k} = \rho_{k} p_k^T g_k, $$

$$ a_{y,k,j} = \rho_{j} \left( p_j^T g_k - \sum_{i=j+1}^{k} p_i^T y_i a_{y,k,i} \right) \qquad (j = k - m + 1, k - m + 2, \cdots, k - 1). $$

Also, we can obtain from (A5)

$$ s_l = -h_k + \sum_{i=k+l}^{k} a_{x,k,i} z_i \qquad (l = -m, -m + 1, \cdots, -1). $$

Substituting (A8) into (45), we obtain

$$ s'_m = -\gamma_k h_k + \sum_{i=k+1}^{k} a_{z,k,i} z_i, $$

where $a_{z,k,j} = \gamma a_{z,k,j}$. If we put

$$ s'_l = -\gamma_k h_k + \sum_{i=k-m+1}^{k} a_{z,k,i} z_i + \sum_{i=k-m+1}^{k} a_{p,k,i} p_i \qquad (l = -m + 1, -m + 2, \cdots, 0), $$

we obtain

$$ a_{p,k,k-m+1} = \rho_{j} \left( \gamma_k y_{k-m+1}^T h_k - \sum_{i=k-m+1}^{k} y_i^T z_i a_{z,k,i} \right) - a_{y,k,k-m+1}, $$

$$ a_{p,k,j} = \rho_{j} \left( \gamma_k y_j^T h_k - \sum_{i=k-m+1}^{k} y_i^T z_i a_{z,k,i} - \sum_{i=k-m+1}^{k} y_i^T p_i a_{p,k,i} \right) - a_{y,k,j} \qquad (j = k - m + 2, k - m + 3, \cdots, k). $$

Because $s'_0 = d_k$, (39) is the same as (A10) when $l = 0$, and coefficients $a_{c,k,j}$ and $a_{p,k,j}$ in (39) agree with the coefficients in (A10).

References


