Variable-storage Quasi-Newton Operators as Inverse Forecast/Analysis Error Covariance Matrices in Variational Data Assimilation

Fabrice Veersé

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Variable-storage quasi-Newton operators as inverse forecast/analysis error covariance matrices in variational data assimilation

Fabrice VEERSÉ

No 3685

April 26, 1999

THÈME 4

Rapport de recherche

ISSN 0249-6399

Variable-storage quasi-Newton operators as inverse forecast/analysis error covariance matrices in variational data assimilation

Fabrice VEERSÉ*

Thème 4 — Simulation et optimisation
de systèmes complexes
Projet IDOPT

Rapport de recherche n ° 3685 — April 26, 1999 — 28 pages

Abstract: Two approximations of the Hessian matrix as limited-memory operators are built from the limited-memory BFGS inverse Hessian approximation provided by the minimization code, in view of the specification of the inverse analysis/forecast error covariance matrix in variational data assimilation. Some numerical experiments and theoretical considerations lead to reject the limited-memory DFP Hessian approximation and to retain the BFGS one for the applications foreseen. Conditioning issues are explored and a preconditioning strategy via a change of control variable is proposed, based on a suitable Cholesky factorization of the limited-memory inverse Hessian matrix. This factorization is implemented as the composition of linear operators. The memory requirements and the number of floating-point operations required by the method are given and confirmed by numerical experiments. The method is found to have a strong potential for variational data assimilation systems using high resolution ocean or atmosphere general circulation models.

Key-words: limited-memory Hessian, variational data assimilation, analysis error covariances, forecast error covariances

(Résumé : tsvp)

* Fabrice.Veerse@imag.fr

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN (France)
Téléphone : 04 76 61 52 00 - International: +33 4 76 61 52 00
Télécopie : 04 76 61 52 52 - International: +33 4 76 61 52 52

Opérateurs de quasi-Newton à mémoire limitée comme matrices inverses de covariances d'erreur d'analyse/prévision pour l'assimilation variationnelle de données

Résumé : Deux approximations de la matrice hessienne sous la forme d'opérateurs à mémoire limitée sont construites à partir de l'approximation à mémoire limitée de la matrice hessienne inverse fournie par le code de minimisation, en vue de servir pour la spécification de la matrice inverse de covariances d'erreur d'analyse/prévision pour l'assimilation variationnelle de données. Des expériences numériques et des considérations théoriques amènent à rejeter l'approximation à mémoire limitée DFP de la hessienne et à retenir l'approximation BFGS pour les applications envisagées. Les problèmes de conditionnement sont abordés et une stratégie de préconditionnement par changement de variable de contrôle, basée sur une factorisation Cholesky de l'approximation à mémoire limitée de la hessienne inverse, est proposée. Cette factorisation est mise en oeuvre sous la forme d'une composition d'opérateurs linéaires. L'espace mémoire et le nombre d'opérations en virgule flottante de la méthode sont donnés et confirmés par des expériences numériques. La méthode proposée dispose d'un fort potentiel pour les systèmes d'assimilation variationnelle de données utilisant des modèles de circulation générale atmosphériques ou océaniques à haute résolution.

Mots-clé : hessienne à mémoire limitée, assimilation variationnelle de données, covariances d'erreur d'analyse, covariances d'erreur de prévision

Contents

1 Introduction

A key issue in variational data assimilation, and also in Kalman filtering, is the specification of the forecast error covariance matrices and the computation of the analysis error covariances, to take into account the so-called 'errors of the day'. The main difficulty lies in the dimension of the corresponding matrices which have typically between 10^{10} and 10^{12} scalar components.

A variable-storage quasi-Newton minimization algorithm is commonly used to compute the analysis in variational data assimilation, which approximates the inverse Hessian matrix as a linear operator using a limited and controlable amount of memory. The approximation and specification of the analysis error covariance matrix under the form of limited-memory quasi-Newton operators is still largely unexplored. Fisher and Courtier (1995) studied the approximation of the analysis error covariance matrix (that is the inverse Hessian matrix approximation) provided by the variable-storage BFGS minimization code. They introduce a dynamic scaling which proves useful for estimating the small scales fluctuations of the analysis error standard deviations as well as approximating the long range analysis error correlations. They also proposed as a Lanczos method that provided the best approximation and a randomization method that can be used for preconditioning purposes (Yang *et al.*, 1996). To the author's knowledge, there has been no other study of the use of quasi-Newton operators for approximating analysis error covariances.

The concern of the present paper is rather the approximation as limited-memory quasi-Newton operators of the *inverse* analysis error covariance matrix by the Hessian of the cost function at the minimum. It can thus be viewed as a complement to Fisher and Courtier's approach (1995). Our aim is to use in the future this inverse analysis error covariance operator for specifying the inverse background error covariance matrix of the next assimilation period. As pointed out by De Mey (1997) this approach is legitimate whenever the assimilation period is small compared to the local and advective time scales of the flow, since the 'errors of the day' are then well approximated by the 'errors of the previous assimilation period' and their evolution along the assimilation period may be omitted.

Taking this fact into account as well as the relation between the analysis error covariance matrix and the inverse Hessian at the minimum, two methods are proposed for the specification of the inverse error covariance matrices requiring a controlable and limited amount of storage space. The first one uses the exact inverse of the BFGS approximation of the inverse Hessian, and requires the inversion of a matrix of reduced dimension. The second one computes an approximate Hessian matrix using a limited-memory version of the DFP Hessian update formula.

The relation between the inverse Hessian matrix and the analysis error covariance matrix is recalled in the following section. The two methods for the limited-memory approximation of the inverse analysis error covariance matrix are detailed in section 3. The following section concerns the approximation of the inverse forecast error covariance matrix. The issue of preconditioning of the minimization is adressed in section 5. The computer requirements both in terms of memory and number of floating-point operations are given in section 6. Some numerical expriments are presented in section 7. A discussion of the approach proposed for the specification of inverse analysis/forecast error covariance matrices as well as the underlying hypothesis is finally given.

2 Inverse Hessian and analysis error covariance matrices

The relation between the inverse Hessian matrix and the analysis error covariance matrix of 3D variational data assimilation (3D-Var) or 4D variational data assimilation (4D-Var) at the initial time of the assimilation period is well known (Thacker, 1989; Rabier and Courtier, 1992; Wunsch, 1996; Gunson and Malanotte-Rizzoli, 1996; Cohn, 1997). However it is believed useful to recall it and give a few comments here. The cases of standard and incremental variational formulation are distinguished in the next subsections.

2.1 Standard variational formulation

In the standard formulation of 3D-Var and 4D-Var (Le Dimet and Talagrand, 1986) one determines the control variable $\mathbf{x}_0$ by minimizing the cost function

$$\begin{aligned} J(\mathbf{x}_0) &= J_b(\mathbf{x}_0) + J_o(\mathbf{x}_0) \\ &= \frac{1}{2}\left(\mathbf{x}_0 - \mathbf{x}^b\right)^T \mathbf{B}^{-1}\left(\mathbf{x}_0 - \mathbf{x}^b\right) + \frac{1}{2}\left(H(\mathbf{x}_0) - \mathbf{y}^o\right)^T \mathbf{R}^{-1}\left(H(\mathbf{x}_0) - \mathbf{y}^o\right) \end{aligned}$$

where

$\mathbf{x}_0$ is usually taken as the initial state of the atmospheric or oceanic model, but may also include some boundary conditions or model error terms;

J_b is the background penalty cost function;

J_o is the observational cost function;

$\mathbf{x}^b$ is the so-called background field, that is an a priori estimate of the optimal control;

$\mathbf{B}$ is the covariance matrix of background (forecast) error;

$\mathbf{y}^o$ is the observation vector which collects all the observations at the analysis time in the 3D-Var case and over the assimilation period in the 4D-Var one;

$\mathbf{R}$ is the observation error covariance matrix, which accounts for both measurement and representativeness errors (see *e.g.* Lorenc, 1986);

H is the (nonlinear) observation operator which from the control variable computes quantities equivalent to observations. In the 4D-Var case where the model initial condition is used as control variable, it includes temporal integrations of the model from the initial time of the assimilation period to the various individual observation instants.

In the above formulation, ^{T} and $^{-1}$ denote matrix transposition and inversion respectively.

The Hessian of this cost function is readily found to be

$$\begin{aligned}\nabla^2\mathcal{J}(\mathbf{x}_0) &= \mathbf{B}^{-1} + \mathbf{H}'(\mathbf{x}_0)^T\mathbf{R}^{-1}\mathbf{H}'(\mathbf{x}_0) \\ &+ \frac{1}{2}\left[\mathbf{H}''(\mathbf{x}_0)\mathbf{R}^{-1}\left(H(\mathbf{x}_0)-\mathbf{y}^o\right) + \left(H(\mathbf{x}_0)-\mathbf{y}^o\right)\mathbf{R}^{-1}\mathbf{H}''(\mathbf{x}_0)\right]\end{aligned}$$

where $'$ denotes the differentiation. Therefore, when the nonlinearities are negligible the analysis error covariance matrix $\mathbf{B}^{-1} + \mathbf{H}'(\mathbf{x}_0)^T\mathbf{R}^{-1}\mathbf{H}'(\mathbf{x}_0)$ of the BLUE (Best Linear Unbiased Estimator) is well approximated by this inverse Hessian at the minimum (Thacker, 1989). As shown by the derivation of Gunson and Malanotte-Rizzoli (1996), this relation holds under the assumption of validity of the linearization of the observation operator (the so-called tangent-linear hypothesis).

Following the work of Le Dimet *et al.* (1997), it can be stressed that the Hessian matrix of the cost function at the minimum contains the information on the sentivities of the analysis. Therefore whenever the nonlinearities cannot be neglected anymore the inverse Hessian matrix at the minimum can still be expected to inform on the analysis errors, which is not the case for the analysis error covariance matrix of the BLUE because the underlying hypothesis does not hold anymore.

2.2 Incremental variational formulation

In the incremental formulation of Courtier *et al.* (1994) the observation operator is linearized around a first guess $\mathbf{x}_0$, initially taken equal to the background field $\mathbf{x}^b$. A correction $\delta\mathbf{x}_0$ to the first guess is computed by minimizing the quadratic cost function

$$\begin{aligned}\mathcal{J}(\delta\mathbf{x}_0) &= \tfrac{1}{2}\left(\mathbf{x}_0+\delta\mathbf{x}_0-\mathbf{x}^b\right)^T\mathbf{B}^{-1}\left(\mathbf{x}_0+\delta\mathbf{x}_0-\mathbf{x}^b\right) \\ &+ \tfrac{1}{2}\left(H(\mathbf{x}_0)+\mathbf{H}'(\mathbf{x}_0).\delta\mathbf{x}_0-\mathbf{y}^o\right)^T\mathbf{R}^{-1}\left(H(\mathbf{x}_0)+\mathbf{H}'(\mathbf{x}_0).\delta\mathbf{x}_0-\mathbf{y}^o\right).\end{aligned}$$

The first-guess $\mathbf{x}_0$ is then updated by adding the computed correction, a new quadratic function is defined by linearizing the observation operator around this updated first-guess. It is minimized to compute a new correction, and the whole process of updating and minimization of the quadratic cost function is repeated a number of times to account for the nonlinearities of the observation operator. The incremental formulation also includes a simplication operator which consists in a change of resolution and/or physics (Courtier *et al.*, 1994; Veersé and Thépaut, 1998; Laroche and Gauthier, 1998). The present discussion restricts to the above formulation of the quadratic cost function for clarity purposes, but can easily be generalized to the case involving this simplification operator.

The Hessian of the incremental quadratic cost function is

$$\nabla^2\mathcal{J}(\mathbf{x}_0) = \mathbf{B}^{-1} + \mathbf{H}'(\mathbf{x}_0)^T\mathbf{R}^{-1}\mathbf{H}'(\mathbf{x}_0)$$

which would be equal to the analysis error covariance matrix of the BLUE if $\mathbf{x}_0$ would be the analysis. However in practice $\mathbf{x}_0$ is the first guess of the last minimization performed and is different from the analysis. Therefore the Hessian matrix of the last quadratic cost function minimized can be expected to approximate the analysis error covariance matrix only if the corresponding first guess is not too far from the analysis and the corresponding computed correction is small. In pratice this condition is likely to be satisfied after a number of updates, since the jump in the observational term of the cost function due to the first-guess corrections goes diminishing (see for example the composite cost functions in Veersé and Thépaut (1998)).

The same restriction holds for the standard variational formulation since in practice the minimization is stopped after a number of iterations and the gradient is not exactely zero.

3 Limited-memory inverse analysis error covariance operators

It has been seen in the previous section that the analysis error covariances can be approximated to some degree by the inverse Hessian at the end of the minimization of the variational data assimilation. An approximation of this inverse Hessian is built by the limited-memory quasi-Newton minimization code usually used. The validity of this approximation has been shown by Fisher and Courtier (1995) who also studied two other methods, namely a Lanczos one and a randomization method. All these methods are useful for obtaining some information about the quality of the computed analysis.

However it is the inverse covariance matrices that appear in the variational formulations of data assimilation. Hence it is interesting to build also computationally efficient approximations of the Hessian matrix to approximate the inverse analysis error covariance matrix. This approximation could be used for cycling purposes whenever the assimilation period is short enough for the evolution of the errors to be legitimately omitted.

After the variable-storage inverse BFGS algorithm for the minimization is recalled (§ 3.1), two methods are proposed for the approximation of the Hessian matrix by limited-memory quasi-Newton operators in § refqnops.

3.1 Limited-memory inverse BFGS minimization algorithm

In the present study the limited-memory quasi-Newton minimization code M1QN3 from INRIA is used (Gilbert and Lemaréchal, 1989). It is based on the Broyden-Fletcher-Goldbard-Shanno update formula for the inverse Hessian (Broyden, 1969; Dennis and Moré, 1977) :

$$\mathbf{H}^+ = U(\mathbf{H},\mathbf{y},\mathbf{s}) = \left(\mathbf{I} - \frac{\mathbf{s}\otimes\mathbf{y}}{\langle\mathbf{y},\mathbf{s}\rangle}\right)\mathbf{H}\left(\mathbf{I} - \frac{\mathbf{s}\otimes\mathbf{y}}{\langle\mathbf{y},\mathbf{s}\rangle}\right)\frac{\mathbf{s}\otimes\mathbf{s}}{\langle\mathbf{y},\mathbf{s}\rangle}$$

where $\mathbf{H}^+$ is the updated version of the inverse Hessian $\mathbf{H}$, $\mathbf{s} = \mathbf{x}^+ - \mathbf{x}$ is the difference between the new iterate and the previous one, and $\mathbf{y} = \mathbf{g}^+ - \mathbf{g}$ is the corresponding gradient increment. Here $\langle\ ,\ \rangle$ is the scalar product with respect to which the gradient is defined and the minimization is to be performed; and $\mathbf{u} \otimes \mathbf{v}$ is the linear operator that to a vector $\mathbf{d}$ associate the vector $\langle \mathbf{v}, \mathbf{d} \rangle \, \mathbf{u}$.

In the limited-memory version (Nocedal, 1980; Gilbert and Lemaréchal, 1989; Liu and Nocedal, 1989) one can afford to store, say, m couples of vectors $(\mathbf{s}, \mathbf{y})$. The above update formula is used for the first m iterations. For the subsequent ones, the following algorithm is used :

$$\begin{aligned} \mathbf{H}_k^o &= \mathbf{D}_k, \\ \mathbf{H}_k^{i+1} &= U(\mathbf{H}_k^i, \mathbf{y}_{k-m+i}, \mathbf{s}_{k-m+i}), \quad \text{for } 0 \le i \le m-1, \\ \mathbf{H}_k &= \mathbf{H}_k^m. \end{aligned}$$

The starting matrix $\mathbf{D}_k$ is diagonal and several formulations have been proposed and tested by Gilbert and Lemaréchal (1989) and Liu and Nocedal (1989). The M1QN3 minimization code from INRIA either specifies $\mathbf{D}_k$ as the identity matrix multiplied by the Oren-Spedicato factor $\langle \mathbf{y}_k, \mathbf{s}_k \rangle / \langle \mathbf{y}_k, \mathbf{y}_k \rangle$ (scalar initial scaling mode), or updates $\mathbf{D}_k$ using a diagonalized BFGS formula involving $\mathbf{s}_k$ and $\mathbf{y}_k$ (diagonal initial scaling mode).

During the minimization only the multiplication of this approximate inverse Hessian matrix by a given vector is needed and is performed efficiently using a two-loop recursion proposed by Nocedal (1980), and the corresponding matrices are never formed. It is precisely this aspect that make the limited-memory inverse BFGS algorithm suitable for variational data assimilation in meteorology and oceanography, as the size of the corresponding matrices is important (at least $10^5 \times 10^5$).

3.2 Limited-memory quasi-Newton covariance operators

Using the approximation of the inverse Hessian built by the variable-storage inverse BFGS method just described, that is the diagonal matrix $\mathbf{D}_k$ and the m couples $(\mathbf{s}, \mathbf{y})$, two approximations of the Hessian matrix are built.

Limited-memory DFP covariance operator

It is well known that if in the inverse BFGS update formula

$$\mathbf{H}^+ = U(\mathbf{H}, \mathbf{y}, \mathbf{s})$$

the following interchanges are made:

$$\begin{aligned} \mathbf{H} &\longrightarrow \mathbf{H}^{-1} \\ \mathbf{H}^+ &\longrightarrow (\mathbf{H}^+)^{-1} \\ \mathbf{s} &\longrightarrow \mathbf{y} \\ \mathbf{y} &\longrightarrow \mathbf{s} \end{aligned}$$

then the DFP (Davidon-Fletcher-Powell) update formula for the Hessian is recovered (Dennis and Moré, 1977; Navon and Legler, 1987; Bonnans *et al.*, 1997).

Therefore a limited-memory DFP approximation of the Hessian matrix is obtained by inverting the diagonal matrix $\mathbf{D}_k$ and interchanging the $\mathbf{s}$'s and the $\mathbf{y}$'s. This is easily implemented by using the inverse Hessian matrix-vector product built in the minimization code and based on Nocedal's (1980) algorithm.

Limited-memory BFGS covariance operator

Another possibility is to build the exact inverse of the inverse Hessian approximation made by the minimization code. After minimization, the inverse Hessian approximation has the following form of a $2m$-rank correction to a diagonal matrix:

$$\mathbf{H} = \mathbf{D}_k + \mathbf{V}\Lambda\mathbf{V}^T \tag{1}$$

where $\mathbf{D}_k$ is the final value of the diagonal matrix updated by the limited-memory inverse BFGS algorithm, and Λ and $\mathbf{V}$ contain respectively the $2m$ eigenvalues and eigenvectors of the rank-deficient update matrix. The product of $\mathbf{H}$ by any vector is built in the minimization code using Nocedal's (1980) two-loop recursion and is thus available. The product of $\mathbf{D}_k$ with any vector is straightforwardly obtained. Therefore, the eigen-decomposition of the rank-deficient update matrix can be obtained using a Lanczos or Arnoldi-type algorithm.

Applying the Woodbury formula to the above expression of the inverse Hessian, one obtains the limited-memory BFGS approximation of the Hessian matrix:

$$\mathbf{H}^{-1} = \mathbf{D}_k^{-1} - \mathbf{D}_k{}^{-1}\mathbf{V}\left[\Lambda^{-1} + \mathbf{V}^T\mathbf{D}_k^{-1}\mathbf{V}\right]^{-1}\mathbf{V}^T\mathbf{D}_k^{-1}.$$

The matrix between the brackets is of small size $2m \times 2m$ and can be easily factored under the classical form LDL^T (note that it is not positive definite and therefore it cannot be Cholesky-factored). This way a limited-memory BFGS Hessian-vector product is obtained.

A refinement of this approach has been proposed by Byrd *et al.* (1994) and requires only the Cholesky factorization of an $m \times m$ matrix, a $2m$ forward substitution and a $2m$ backward one. However their method is developed for a quasi-Newton method based on the Euclidean scalar product and their diagonal matrix $\mathbf{D}_k$ is a multiple of the identity matrix.

Here the cruder but more easily implemented above approach will be used. But for operational implementation a generalized version of the method of Byrd *et al.* (1994) ought to be considered.

At this point, it is worth noting that both Hessian-vector product operators have been built directly from the inverse Hessian approximation provided by the limited-memory inverse BFGS algorithm at the end of the minimization. Therefore whenever the assimilation problem is preconditioned by a change of control variable (usually scaling them by the corresponding standard error deviations), what is obtained is an approximation of the Hessian matrix of the preconditioned problem. An approximation of the Hessian matrix can be easily recovered as explained in Fisher and Courtier (1995).

4 Limited-memory inverse forecast error covariance operators

For the next assimilation, one should use as forecast (background) error covariances the analysis error covariances evolved from the previous analysis time (3D-Var case) or from the beginning of the previous assimilation period (4D-Var case) according to the model dynamics. The extended Kalman filter transports these error covariances using the model linearized around the current trajectory (the so-called tangent linear model) between analysis times (3D-Var) or along the assimilation period (4D-Var), and the evolved analysis error covariance matrix $\mathbf{A}^+$ is:

$$\mathbf{A}^+ = \mathbf{M}\mathbf{A}\mathbf{M}^T + \mathbf{Q},$$

where $\mathbf{M}$ and $\mathbf{Q}$ are the tangent linear model resolvant and the model error covariance matrix over the corresponding time period. In the strong formulation of variational data assimilation studied here, no model error is assumed and $\mathbf{A}^+$ is given by

$$\mathbf{A}^+ = \mathbf{M}\mathbf{A}\mathbf{M}^T.$$

At this point, a remark can be made. Approximating the analysis error covariance matrix $\mathbf{A}$ by the expression of the inverse Hessian approximation given in (1), one obtains

$$\mathbf{A}^+ \approx \mathbf{M}\mathbf{D}_k\mathbf{M}^T + \mathbf{M}\mathbf{V}\Lambda\mathbf{V}^T\mathbf{M}^T. \tag{2}$$

Now, if the scalar initial mode of the minimization code is used $\mathbf{D}_k$ is a multiple of the identity matrix; and it has been found with some spectral atmospheric models that when the diagonal scaling mode is used $\mathbf{D}_k$ is close to a multiple of the identity (Veersé, 1997). The first term in the above sum can be written in terms of suitably scaled tangent-linear-model modes over the corresponding time period, whereas the second term is just a low-rank correction. The qualitative link between unstables modes and dynamical structure functions diagnosed by Thépaut *et al.* (1996) is thus not suprising if the low-rank correction does not impact much on the most unstable modes.

In the present study, the focus is on flows for which the local and advective time scales are much bigger that the length of the assimilation period and the transport of the analysis error covariances can be legitimately omitted. However, from equation (2) a proposal can be made to approximate this evolution when these hypotheses are not satisfied. The idea is to approximate the tangent-linear model in the first term of the sum by a cheap adaptively tuned operator, such as a solid rotation around the Earth poles. This operator should be easily invertible to enable the use of suitably modified versions of the methods proposed in the previous section.

5 Preconditioning issues

To precondition the variational data assimilation problem, a change of variable is often used. The inverse background error covariance matrix is Cholesky-factored as $\mathbf{B}^{-1} = \mathbf{U}^T\mathbf{U}$ and the cost function is reformulated in terms of $\chi = \mathbf{U}(\mathbf{x}_0 - \mathbf{x}^b)$ for the standard variational problem or $\chi = \mathbf{U}(\mathbf{x}_0 + \delta\mathbf{x}_0 - \mathbf{x}^b)$ for the incremental formulation. For example, the cost function which is effectively minimized for the standard variational problem is:

$$\tilde{\mathcal{J}}(\chi) = \frac{1}{2}\,\chi^T\chi + \frac{1}{2}\,\left(H\left(\mathbf{U}^{-1}\chi + \mathbf{x}^b\right) - \mathbf{y}^o\right)^T \mathbf{R}^{-1}\left(H\left(\mathbf{U}^{-1}\chi + \mathbf{x}^b\right) - \mathbf{y}^o\right). \qquad (3)$$

In the absence of the observational (second) term in the sum, this cost function would be minimized in one iteration provided a careful choice of the initialization parameters of the limited-memory inverse BFGS minimization code. This preconditioning strategy is found to be effective in practice and is used by most meteorological centers performing variational data assimilations.

Here it is assumed that the transport can be omitted in (2) so that $(\mathbf{A}^+)^{-1}$ would be one approximation of the Hessian proposed in section 3 in the absence of preconditioning. When the above preconditioning is used, substituting $\mathbf{B}^{-1}$ by $(\mathbf{A}^+)^{-1}$ in the standard variational data assimilation cost function for the new assimilation period yields

$$\tilde{\mathcal{J}}(\chi) = \frac{1}{2}\,\chi^T(\mathbf{H})^{-1}\chi + \frac{1}{2}\,\left(H\left(\mathbf{U}^{-1}\chi + \mathbf{x}^b\right) - \mathbf{y}^o\right)^T \mathbf{R}^{-1}\left(H\left(\mathbf{U}^{-1}\chi + \mathbf{x}^b\right) - \mathbf{y}^o\right). \qquad (4)$$

where χ is defined as before and $(\mathbf{H})^{-1}$ is the preconditioned Hessian matrix approximated by any of the two methods of section 3 on the previous assimilation period. One realizes this problem is not well preconditioned anymore, though the minimization can be started using the approximation of the inverse Hessian $\mathbf{H}$ generated during the previous 3D-Var analysis or 4D-Var assimilation. To precondition it, a Cholesky factorization of $(\mathbf{H})^{-1}$ would be useful. Powell (1987) showed how the full-memory inverse BFGS update formula can be implemented using a Cholesky factorization $\mathbf{H}_k = \mathbf{Z}_k\mathbf{Z}_k^T$ of the inverse Hessian approximation. He obtains the following update formulae for $\mathbf{Z}_k$:

$$\mathbf{z}_i^+ = \begin{cases} \mathbf{s}_k/\sqrt{\mathbf{s}_k{}^T\mathbf{y}_k} & i = 1 \\ \mathbf{z}_i - \left(\frac{\mathbf{y}_k{}^T\mathbf{z}_i}{\mathbf{s}_k{}^T\mathbf{y}_k}\right)\mathbf{s}_k & i = 2, ..., n \end{cases}$$

where $\mathbf{z}_i$ and $\mathbf{z}_i^+$ are the i-th columns of $\mathbf{Z}_k$ and $\mathbf{Z}_{k+1}$ respectively. Unfortunately it is not clear how it could be implemented in terms of operators rather than matrices, and the inverse of $\mathbf{Z}_k$ which would be needed for the preconditioning change of control variable is not easily obtained. In the remaining part of this section, it is shown how the Hessian approximations proposed in section 3 can be Cholesky-factored in terms of cheaply and easily invertible operators.

The key point is to realize that the expressions of the Hessian approximations proposed consist in a low-rank ($2m$-rank) correction to a positive definite diagonal matrix, and the latter is trivially of the form LDL^T with L equal to the identity matrix. Therefore what is needed is an algorithm for updating this type of factorization whenever a rank-one matrix is added, which can be implemented as an easily invertible operator and which can be applied recursively. It turns out that such an algorithm exists and has been proposed by Gill *et al.*

(1974). The reader is refered to their article for the details of the algorithms, but the main idea is now exposed.
Assume a factorization $\mathbf{L}_0\mathbf{D}_0\mathbf{L}_0{}^T$ is available and one wants to modify $\mathbf{L}_0$ and $\mathbf{D}_0$ to produce $\mathbf{L}$ and $\mathbf{D}$ satisfying

$$\mathbf{LDL}^T = \mathbf{L}_0\mathbf{D}_0\mathbf{L}_0{}^T + \alpha\mathbf{v}\mathbf{v}^T$$

where α is a scalar (not necessarily positive) and $\mathbf{v}$ is a vector.
Define $\mathbf{p}$, $\mathbf{M}$ and Δ by

$$\mathbf{L}_0\mathbf{p} = \mathbf{v}, \qquad \mathbf{M}\Delta\mathbf{M}^T = \mathbf{D}_0 + \alpha\mathbf{p}\mathbf{p}^T.$$

Thus $\mathbf{p}$ is obtained by forward substitution, and $\mathbf{M}$ and Δ are the Cholesky factors of a simple matrix. Hence

$$\mathbf{LDL}^T = \mathbf{L}_0\left(\mathbf{D}_0 + \alpha\mathbf{p}\mathbf{p}^T\right)\mathbf{L}_0{}^T = \mathbf{L}_0\left(\mathbf{M}\Delta\mathbf{M}^T\right)\mathbf{L}_0{}^T = \left(\mathbf{L}_0\mathbf{M}\right)\Delta\left(\mathbf{L}_0\mathbf{M}\right)^T.$$

Therefore the updated Cholesky factors are

$$\mathbf{L} = \mathbf{L}_0\mathbf{M}, \qquad \mathbf{D} = \Delta.$$

Gill *et al.* (1974) show that the lower triangular matrix $\mathbf{M}$ is defined by $\mathbf{p}$ and another vector β as follows:

$$\mathbf{M} = \begin{bmatrix} 1 & & & & & \\ p_2\beta_1 & 1 & & & & \\ p_3\beta_1 & p_3\beta_2 & 1 & & & \\ . & & & . & & \\ . & & & & . & \\ . & & & & 1 & \\ p_n\beta_1 & p_n\beta_2 & & & p_n\beta_{n-1} & 1 \end{bmatrix}$$

and give an algorithm for the computation of β and Δ. The matrix $\mathbf{M}$ is particularly simple to invert and the corresponding operator performs a forward substitution. Note also that this type of operation is usually efficiently implemented on vector and parallel computers.

At this point, it is worth noting that the limited-memory BFGS inverse Hessian approximation is a low-rank correction to a diagonal matrix (see Eq. (1)). Therefore the algorithm of Gill *et al.* for updating Cholesky factorizations can be applied to the inverse Hessian approximation directly rather than the Hessian approximations proposed in the previous section. As a result the background error covariance matrix can be Cholesky-factorized using the composition of $2m$ forward substitution operators, and the usual background-term preconditioning of the assimilation problem can be efficiently performed. For our problem, the $4m$ vectors $\mathbf{p}$ and β, and the vector Δ corresponding to the $2m$-rank correction to the diagonal matrix can be precomputed once for all before the minimization is started.
Again, an approximate evolution of the error covariances could be performed by substituting the tangent linear model in the first term of the sum in equation (2) by a cheaply and easily

invertible operator defined in terms of quantities that can be adaptively tuned.
Nevertheless the methods of the previous section for approximating the Hessian matrix as a limited-memory operator are still useful, *e.g.* for the approximative computations of analysis sensitivy patterns whenever the second order adjoint model is not available (Le Dimet *et al*, 1997).

6 Memory requirements and complexity

Let m and N stand for the number of $(\mathbf{s}, \mathbf{y})$ pairs used for the minimization and the dimension of the control variable respectively.

6.1 Memory requirements

DFP Hessian approximation

The additional memory requirements for implementing the DFP Hessian matrix/vector product of section 3 inside the background cost function $\mathcal{J}^b$ are paramount to the working-space memory requirements of the minimization code. These are:

- an integer array of dimension 5;
- a real array of dimension $6N + m$ if the $(\mathbf{s}, \mathbf{y})$ pairs are not stored in memory;
- otherwise a real array of dimension $3N + (2N + 1)m$ or $4N + (2N + 1)m$ according to the initial scaling mode used.

BFGS Hessian approximation

The additional memory requirements for the BFGS Hessian/vector product of section 3 are:

- a real array of size $2m$ and a real array of dimension $2mN$ to store the inverse Hessian update matrix eigen-decomposition;
- a real array of dimension $m(2m + 1)$ to store the low-rank matrix that is LDL^T factorized; plus an integer array of size $2m$ to handle the corresponding pivots;
- a real array of size N for the diagonal matrix $\mathbf{D}^{-1}$ if the diagonal initial scaling (DIS) mode of the minimization code is used;
- a real array of dimension $2m$ and a real array of dimension N needed locally to perform the Hessian matrix/vector product.

Cholesky-factorized BFGS Hessian approximation

The memory requirements for the BFGS Hessian matrix/vector product built from a Cholesky factorization of the inverse Hessian matrix, following the procedure of section 5, are:

- a real array of size $2m$ and a real array of dimension $2mN$ to store the inverse Hessian update matrix eigen-decomposition;
- a real array of dimension N to store the diagonal of the Cholesky factorization;
- a real array of dimension $2mN$ to store the β vectors of the Cholesky factorization (the $\mathbf{p}$ vectors overwrite the eigenvectors of the inverse Hessian update matrix);
- an array of dimension N is used locally for the multiplication by the lower-triangular M matrices.

6.2 Complexity

The order of floating points operations needed for the various approximations are summarized in Table 1. Is is recalled that typical values for m are in the range $[3, 20]$ and N is of order $O(10^6)$ to $O(10^9)$.

The number of floating points operations (flops) needed for the BFGS inverse Hessian matrix/vector product is given in Nocedal (1980). The DFP Hessian matrix/vector product requires just the inversion of the diagonal matrix once for all and swapping the $\mathbf{s}$'s and the $\mathbf{y}$'s in the two-loop recursion, therefore its complexity order is the same as that of the BFGS inverse Hessian matrix/vector product, namely $O(mN)$ flops.

The BFGS Hessian approximation of section 3 and the Cholesky factorization of the inverse Hessian of section 5 both require the eigen-decomposition of the inverse Hessian update matrix. Using a Lanczos algorithm, the cost is dominated by the update matrix/vector product at each iteration. The update matrix/vector product requires the BFGS inverse Hessian matrix/vector product plus the substraction of the product of the vector by the diagonal matrix, essentially $O(mN)$ flops. The number of Lanczos iterations needed is found to be of order $O(m)$. Therefore the inverse Hessian update matrix eigen-decomposition is roughly $O(m^2N)$. This operation has to be done only once per assimilation period.

The BFGS low-rank matrix LDL^T factorization requires the computation and packing of the low-rank symmetric matrix, $O(m^2N)$ flops, and its factorization, $O(m^3)$. The complexity is therefore roughly $O(m^2N)$ flops. This factorization needs to be done once per assimilation period.
The BFGS Hessian matrix/vector product is found to require $O(mN)$ flops. This product is needed at each iteration of the minimization of the cost-function (4).

The Cholesky factorization of the BFGS inverse Hessian matrix requires m Cholesky updates using the method C2 of Gill *et al.* (1974) and $(m-1)^2/2$ forward substitutions. Each Cholesky updates and forward substituion needs $O(N)$ flops. The complete factorization therefore require $O(m^2N)$ flops. This needs to be done only once per assimilation period.

Table 1: Order of complexity for the various approximations in floating-point operations

Minimization code:	
BFGS inverse Hessian matrix/vector product	$O(mN)$
Limited-memory DFP covariance operator:	
DFP Hessian matrix/vector product	$O(mN)$
Limited-memory BFGS covariance operator:	
Inverse Hessian update matrix eigen-decomposition	$O(m^2N)$
BFGS low-rank matrix LDL^T factorization	$O(m^2N)$
BFGS Hessian matrix/vector product	$O(mN)$
Preconditioned limited-memory covariance operator:	
Inverse Hessian update matrix eigen-decomposition	$O(m^2N)$
Inverse Hessian Cholesky factorization	$O(m^2N)$
Inverse Cholesky factor/vector product	$O(mN)$
Cholesky factor/vector product	$O(mN)$

This factorization is used to implement a preconditioned version of the cost function (4), via a change of control variable.
The inverse Cholesky factor/vector product is used in the change of control variable. It requires $2m$ forward substitutions and the division by the positive square-root of the diagonal matrix, that is $O(mN)$ flops.
The Cholesky factor/vector product is needed in the inverse change of control variable. It requires the multiplication by the positive square-root diagonal matrix and the multiplication by the $2m$ lower-triangular M matrices. Each of these multiplications needs $O(N)$ flops. Therefore the Cholesky factor/vector product requires $O(mN)$ flops. This product is needed at each minimization iteration during the assimilation process.

7 Numerical experiments

To perform some numerical tests, the 2D atmospheric spectral vorticity equation model and related variational data assimilation system of Veersé and Thépaut (1998) is used. The model is written in terms of the stream function. 500 hPa extra-tropical geopotential height observations from radiosondes are assimilated using a linear balance equation as observation operator. The observation error covariance matrix $\mathbf{R}$ is a multiple of the identity. The background error covariance matrix is originally specified using a constant error variance for the stream function and the parametric diagonal correlation spectrum of Courtier *et al.* (1993).

The first experiment compares the eigen-spectrum of the Hessian approximations proposed in section 3. The second one assesses the efficiency of the preconditioning via the Cholesky factorization of the BFGS inverse Hessian operator, described in section 5. The cycling strategy for the variational data assimilation cannot be assessed with this model, as it violates the assumptions made on the local and advection time scales of the flow. Finally some timings are given for the various Hessian approximations.

7.1 Hessian approximations spectra

For this experiment, the standard 4D-Var assimilation of Veersé and Thépaut is repeated but at a cruder T42 spectral truncation, using the M1QN3 minimization code from INRIA (Gilbert and Lemaréchal). The eigen-spectrum of the inverse Hessian approximated after the required 80 iterations of the minimizations is computed via a Lanczos algorithm, and the corresponding eigen-values are inversed to get the eigen-spectrum of the BFGS Hessian approximation.
The eigen-spectrum of the DFP Hessian approximation proposed in section 3 is also computed by means of a Lanczos algorithm. For this purpose, the Implicitly Restarted Arnoldi package ARPACK of Lehoucq *et al.* (1997) is used, which enables to compute all the eigenvalues and eigenvectors of the Hessian approximations but one. The cruder spatial resolution enables to compute these almost complete eigen-decomposition with a manageable memory and CPU-time requirements. Since all but one eigen-pairs are computed, there are 1848

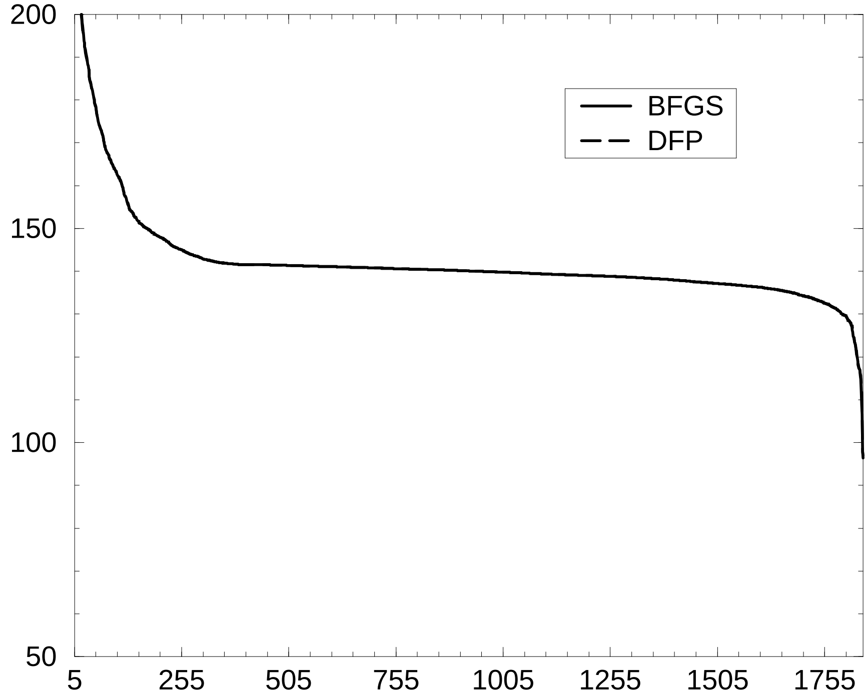

Figure 1: Eigen-spectra of the DFP and BFGS Hessian approximations

eigenvalues. Note however that the Hessian approximations themselves require at most the eigen-decomposition of a small $2m \times 2m$ update matrix, and this is affordable even when a high spatial resolution is used.

Figure 1 show that the limited-memory BFGS and DFP Hessian eigen-spectra are very close (they are indistinguishable on the figure). The theory of limited-memory quasi-Newton minimization methods does not allow to expect such a result, both because of the impact of the limited-memory aspects on the Hessian approximations and because the number of iterations used to build the BFGS inverse Hessian approximation was small compared to the dimension of the problem. In addition, such a result may be due to the special form of the Hessian of the cost function (3), and is indeed a nice surprise.

However to produce this figure, it has been necessary to shift the BFGS Hessian spectrum to the right by 4 units. This means that DFP produces 4 additional larger eigenvalues,

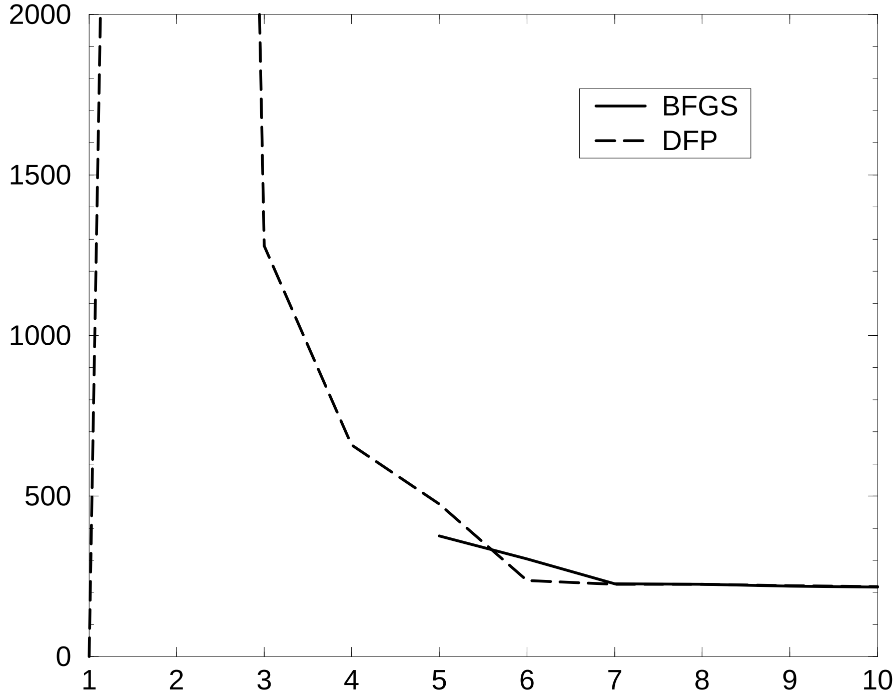

Figure 2: Largest eigenvalues of the Hessian approximations

whereas BFGS produces 4 additional smaller ones. These are shown in figures 2 and 3 respectively. The machine epsilon has been used as criterion on the relative accuracy of the computated eigenvalues (with respect to the true unknown values). Note however that the computed first DFP eigenvalue is negative, about -0.032, and therefore must be rejected. The overshoot of the second and third ones looks also suspect. This observation is consistent with the theory stating that DFP can be expected to develop excessively large Hessian eigenvalues or, at the least, to have difficulties in reducing large eigenvalues from the initial Hessian approximation (see *e.g.* the section 5 of Nocedal, 1992). As for BFGS, the last computed value is also suspect because ARPACK returns the eigenvalues in monotonic order and this last value is misplaced.

In the overall a good agreement is found between the spectra of the DFP and BFGS Hessian approximations. However the BFGS approximation is to be considered more reliable

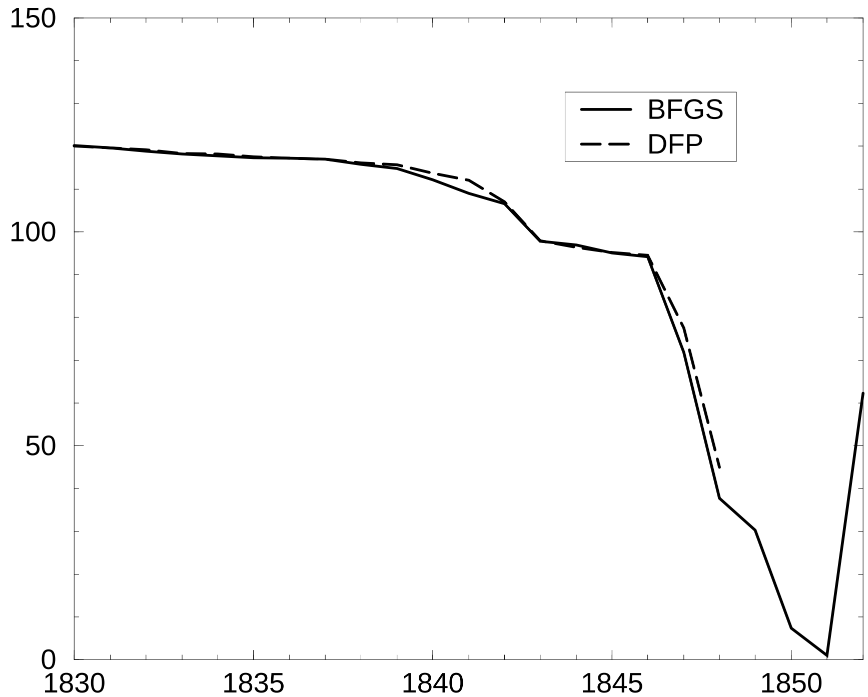

Figure 3: Smallest eigenvalues of the Hessian approximations

as this quasi-Newton update formula has self-correction properties of both large and small eigenvalues and no overshoot of the large eigenvalues values, as evidenced for DFP, is to be expected (Nocedal, 1992). Moreover BFGS is to be prefered for approximating the analysis error covariances because of the additional small Hessian eigenvalues that correspond to the worst estimated directions of the control space during the data assimilation process. In that sense BFGS is more conservative that DFP, the latter tending to underestimate analysis errors. Therefore, although the BFGS approximation of section 3 is more expensive to compute than the DFP one, it is recommended for the applications forseen.
Another reason why to prefer the BFGS arises from the consideration of starting the minimization for the new assimilation period using the inverse Hessian approximation generated by the minimization code at the previous one, *i.e.* using the warm-start option of M1QN3. It is certainly advisable then that the Hessian approximation appearing in the background term of the cost function be the exact inverse of the inverse Hessian approximation from which the minimization is started. In the numerical tests performed, this property was satisfied to machine epsilon accuracy by the BFGS approximation of section 3 provided the $2m$ update matrix eigenvalues are also computed to machine epsilon accuracy.

7.2 Test of the preconditioning

In the previous experiment, a standard 4D-Var assimilation over a 24h period has been performed and an approximation of the inverse Hessian has been built by the minimization code. In this second experiment, different standard 4D-var assimilations are performed over the next 24h period starting from this inverse Hessian approximation (*i.e.* using the warm start option of the minimization code). The triangular spectral truncation is thus kept to T42 and $m = 10$ $(\mathbf{s}, \mathbf{y})$ pairs are used for the limited-memory minimization code. Figure 4 shows the corresponding cost functions and gradients decrease versus the minimization iterations. The top row corresponds to the original preconditioned cost function (3). After 80 iterations, the gradient norm is reduced to 0.289602×10^{-3} its original value. The middle row shows the results for the modified cost function (4). After 8 iterations, the minimization fails to find a suitable step length during the line-search. The poor decrease of the observational cost function $\mathcal{J}^o$, as compared to the original cost function one in the top row, may reflect the poor quality of the background error covariance matrix associated with the violation of the hypothesis on the time scales with this model over a 24h period. The problem is also badly conditioned as the gradient norm is only reduced to 0.216167×10^{-1} its initial value. The bottom row of the figure corresponds to the preconditioned version of this modified cost function, based on a Cholesky factorization of the inverse Hessian approximation. The minimization fails during the line-search after 30 iterations, but the gradient is reduced to 0.223043×10^{-9} its initial value, reflecting the efficiency of the preconditioning. However the final cost function values are close to those of the unpreconditioned case (see Table 2). But this is consistent with the knowledge that in 4D-Var assimilations the minimization acts preferentially on the large scales of the control first and that finer and finer scales are adjusted as the minimization proceeds (Tanguay *et al.*, 1995). The impact of these

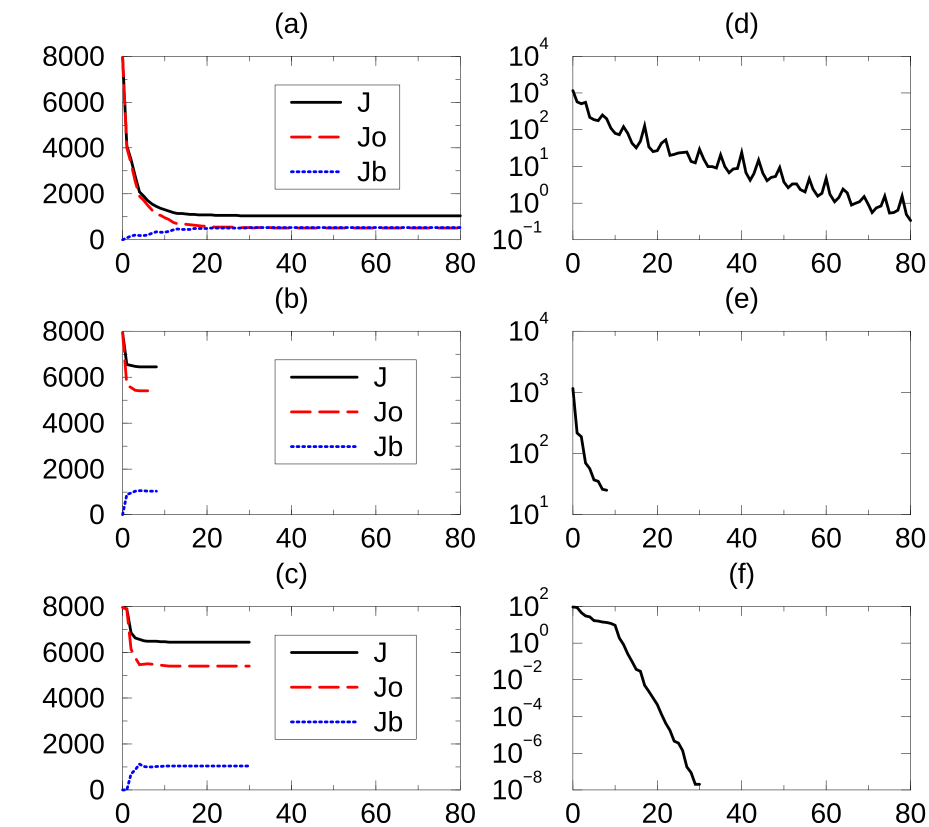

Figure 4: Cost functions (left column) and gradients (rigth column) as a function of the number of iterations. The original background error covariance matrix B is used in (a) and (d). The background covariance matrix B is deduced from the limited-BFGS operator of section 3 in (b) and (e). A preconditioned version of the latter B matrix is used in (c) and (f), built from a Cholesky factorization of the inverse Hessian approximation.

Table 2: Modified cost function final values

	unpreconditioned 8th iteration	preconditioned 30th iteration
J	0.6462E+04	0.6440E+04
Jo	0.5422E+04	0.5396E+04
Jb	0.1040E+04	0.1044E+04

finer scales is small on the cost function value although they contribute significantly to the gradient norm.

One difficulty with the modified cost function (either preconditioned or not) is to specify a convergence criterion or a maximum number of iterations that avoids the failure during the line-search, before which a large number of unwanted cost function and gradient evaluations may be performed. For the preconditioned case, a hint may be given by noting that at the 30th iteration the directional derivative, that is the scalar product of the gradient and the descent direction, is very close to zero (-0.32005×10^{-15}). Therefore the failure during the line-search is likely to be associated with (rounding) errors in the gradient computation (see the M1QN3 documentation). For this preconditioned modified problem the Hessian matrix is equal to the sum of the identity matrix and a positive symmetric matrix. Therefore

$$|\langle \mathbf{Hg}, \mathbf{g} \rangle| \quad \leq \quad ||\mathbf{Hg}||\,||\mathbf{g}|| \quad \leq \quad ||\mathbf{H}||\,||\mathbf{g}||^2 \quad < \quad ||\mathbf{g}||^2$$

and the relative precision on the gradient norm ε_g satisfies

$$\varepsilon_g > \frac{\sqrt{|\langle \mathbf{Hg}, \mathbf{g} \rangle|}}{||\mathbf{g}_0||},$$

where $\mathbf{g}_0$ is the initial gradient. Here $||\mathbf{g}_0|| = 92.851989$, taking as a lower bound $|\langle \mathbf{Hg}, \mathbf{g} \rangle| = \varepsilon_m \approx 2.22 \times 10^{-16}$ yields $\varepsilon_g > 0.106 \times 10^{-9}$. Since the majoration of $||\mathbf{H}||$ by 1.0 may be crude, the value achieved (0.223×10^{-9}) can be considered optimal. This suggests to set safely the relative criterion on the gradient norm as $10\sqrt{\varepsilon_m}/||\mathbf{g}_0||$.

7.3 Timings

Some timings have been performed to confirm the complexity order estimates of section 6. The standard 4D-Var assimilation of the first experiment has been redone using the higher spectral triangular truncation T95. Therefore the number of $(\mathbf{s}, \mathbf{y})$ pairs used is still $m = 10$, but the dimension of the control variable is now $N = 9216$. The tests have been performed on a Sun Ultra-1 workstation (Sparc processor). The average CPU times over 100 realizations are reported in Table 3.

In the previous subsection the potential of the preconditioned Cholesky-factorized Hessian matrix/vector product for specifying the inverse background error covariance matrix has

Table 3: Average CPU times in seconds. The operations preceded by (*) need to be performed only once per assimilation period. The others are likely to be done at each iteration of the minimization during the assimilation process.

(*) Reading of the inverse Hessian approximation	0.020
(*) Inverse Hessian update matrix eigen-decompostion	5.411
(*) Inverse Hessian Cholesky factorization	1.466
Additional change of variable	0.127
Additional inverse change of variable	0.148
BFGS inverse Hessian matrix/vector product	0.062
DFP Hessian matrix/vector product	0.063
(*) BFGS low-rank matrix LDL^T factorization	0.565
BFGS Hessian matrix/vector product	0.052

been evidenced. The execution times reported in Table 3 are consistent with the complexity order estimates given in section 6. All the operations needed only once per assimilation period take at most $O(m^2N)$ flops and those that need to be performed at each iteration of the minimization require $O(mN)$ flops. This makes the method pratical for variational assimilations using high resolution models.

8 Discussion

This study was motivated by a main difficulty of variational data assimilation methods in meteorology and oceanography, the specification of the inverse forecast (background) error covariance matrix. The problem comes from the dimensions of the covariance matrix (about $10^9 \times 10^9$ for the future MERCATOR operational ocean prediction system) which prevents from handling it explicitly with current computers, and the lack of knowledge and information for specifying all the corresponding scalar components. Consequently the forecast covariance matrix is usually modeled from physical principals as the composition of invertible linear operators and represents a climatologic information on the forecast error covariances rather than the true background error covariances. This has been identified as a major deficiency of current variational data assimilation systems (Tanguay *et al.*, 1997).

However the extended Kalman filter theory states that the forecast error covariance matrix may be taken as the previous analysis error covariance matrix evolved in time according to the linearized model dynamics. In section 2 it is recalled that the analysis error covariance matrix can be approximated by the inverse Hessian approximation at the minimum, an approximation of which is provided by the limited-memory minimization code. The quality of this approximation has been demonstrated by Fisher and Courtier (1995) for the INRIA M1QN3 code. Nevertheless a better approximation of the inverse Hessian may be obtained by using a different formula for updating the diagonal matrix $\mathbf{D}_k$, such as the quasi-Cauchy proposal of Nazareth *et al.* (1999).

Restricting the attention to the case where the assimilation period is much smaller than the local and advective time scales of the flow, the time evolution of the analysis error covariances may be neglected and the inverse forecast error covariance matrix can be specified as the Hessian matrix of the previous assimilation.

In section 3 it has been show how two limited-memory Hessian approximations can be built from the limited-memory BFGS inverse Hessian approximation provided by the minimization code. The comparison of their eigen-spectra in section 7 as well as theoretical considerations lead to reject the limited-memory DFP Hessian approximation and to prefer the BFGS one for approximating the inverse analysis error covariance matrix.

Potential bad-conditioning problems have been anticipated in section 5 and lead to propose an alternative implementation of the BFGS Hessian approximation based on a Cholesky factorization of the limited-memory inverse Hessian matrix. This Cholesky factorization is implemented as the composition of forward substitution operators and triangular linear operators. The bad conditioning of the problem has been evidenced in the second numerical experiment of section 7, as was the efficiency of the preconditioning based on the Cholesky factorization.

The memory requirements and the orders of the number of floating-point operations (flops) required by the various Hessian approximations have been given in section 6. It can be seen that memory requirements are controlled by the number m of $(\mathbf{s}, \mathbf{y})$ pairs used for the minimization and make the method feasible for high resolution models. All the operations to be performed once per assimilation period need at most $O(m^2N)$ flops. All the operations to be performed at each iteration of the minimization during the assimilation require $O(mN)$

flops, which is the order of flops needed by the inverse Hessian matrix/vector product built in the minimization code (Nocedal, 1980). These complexity orders have been confirmed by some CPU-time estimations in section 7.

The low memory and CPU-time requirements needed to implement the Cholesky-factorized BFGS Hessian matrix/vector product to specify the inverse background error covariance matrix, as well as the efficiency of the associated preconditioning, make the method suitable for variational data assimilation systems using high resolution models. The main restriction comes from the assumption on the local and advective times scales of the flow, that mainly restricts its applicability to tropical ocean models. However it is possible to generalize the method by performing an approximate transport of the error covariances, replacing the linearized model $\mathbf{M}$ in the first term of the sum in (2) by a cheap and easily invertible operator that could be adaptively tuned. This requires however the extension of the algorithm of Gill *et al.* (1974), for updating the Cholesky factorization, to the case of a more general initial factorization. Additionally a low-rank model error covariance matrix $\mathbf{Q}$ could easily be taken into account.

The cycling of variational assimilation using this method for the specification of the inverse background error covariance matrix remains to be assessed with models satisfying the assumption on the time scales of the flow, such as a tropical ocean circulation model.

The other approximations of the Hessian matrix proposed in section 3 could be used for other applications than the specification of the inverse forecast error covariance matrix, such as the computation of sensitivities whenever the second-order adjoint model is not available.

Acknowledgments

The author would like to thank Jean-Charles Gilbert and Claude Lemaréchal for fruitful discussions regarding some aspects of the limited-memory quasi-Newton minimization methods.

References

Bonnans, F., Gilbert, J.-C., Lemaréchal, C. and Sagastizábal, C., 1997: *Optimisation Numérique. Aspects théoriques et pratiques.* Springer Verlag, 324 pp.

Broyden, C.G., 1969: A new double-rank minimization algorithm. *Notices American Math. Soc.*, **16,** pp. 670

Byrd, R.H., Nocedal, J. and Schnabel, R.B., 1994: Representations of quasi-Newton matrices and their use in limited memory methods. *Math. Prog.*, **63,** 129-156

Cohn, S.E., 1997: An Introduction to Estimation Theory. *J. Meteor. Soc. Japan*, **75-1B,** 257-288

Courtier, P., Andersson, E., Heckley, W., Kelly, G., Pailleux, J., Rabier, F., Thépaut, J.-N., Undén, P., Vasiljevic, D., Cardinali, C., Eyre, J., Hamrud, M., Haseler, J., Hollingsworth, A., McNally, A. and Stoffelen, A., 1993: Variational assimilation at ECMWF. *ECMWF Tech. Memo.*, **194,** 84 pp.

Courtier, P., Thépaut, J.-N. and Hollingsworth, A., 1994: A strategy for operational implementation of 4D-Var, using an incremental approach. *Quart. J. Roy. Meteor. Soc.*, **120,** 1367-1387

De Mey, P., 1997: Data Assimilation at the Oceanic Mesoscale: A Review. *J. Meteorol. Soc. Japan*, **75-1B,** 415-427

Dennis, J.E. and Moré, J.J., 1977: Quasi-Newton methods, motivation and theory. *SIAM Rev.*, **19,** 46-89

Fisher, M. and Courtier, P., 1995: Estimating the covariance matrices of analysis and forecast error in variational data assimilation. *ECMWF Tech. Memo.*, **220,** 28 pp.

Gilbert, J.-C. and Lemaréchal, C., 1989: Some numerical experiments with variable storage quasi-Newton algorithms. *Mathematical Programming*, **B25,** 407-435

Gill, P.E., Golub, G.H., Murray, W. and Saunders, M.A., 1974: Methods for modifying matrix factorizations. *Math. Comp.*, **28,** 505-535

Gunson, J.R. and Malanotte-Rizzoli, P., 1996: Assimilation studies of open-ocean flows. 2. Error measures with strongly nonlinear dynamics. *J. Geophys. Res.*, **101,** 28473-28488

Laroche, S. and Gauthier, P., 1998: A validation of the incremental formulation of 4D variational data assimilation in a nonlinear barotropic flow. *Tellus*, **50A,** 557-572

Le Dimet, F.-X., Ngodock, H.-E. and Luong, B., 1997: Sensitivity Analysis in Variational Data Assimilation. *J. Meteorol. Soc. Japan*, **75-1B,** 245-255

Lehoucq, R.B., Sorensen, D.C. and Yang, C., 1997: ARPACK User's Guide: Solution of Large-Scale Eigenvalue Problems with Implicit Restarted Arnoldi Methods. *Available from http://www.caam.rice.edu/software/ARPACK/*

Liu, D.C. and Nocedal, J., 1989: On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, **49,** 503-528

Lorenc, A., 1986: Analysis methods for numerical weather prediction. *Quart. J. Roy. Meteor. Soc.*, **112,** 1177-1194

Navon, I.M. and Legler, D.M., 1987: Conjugate-Gradient Methods for Large-Scale Minimization in Meteorology. *Mon. Wea. Rev.*, **115,** 1479-1502

Nazareth, J.L., Wolkowicz, H. and Zhu, M., 1999: The Quasi-Cauchy Relation and Diagonal Updating. To appear in *SIAM J. Optimization*

Nocedal, J., 1980: Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, **35,** 773-782

Nocedal, J., 1992: Theory of Algorithms for Unconstrained Optimization. *Acta Numerica*, **1,** 199-242

Powell, M.J.D., 1987: Update conjugate directions by the BFGS formula. *Math. Prog.*, **38,** 29-46

Rabier, F. and Courtier, P., 1992: Four-dimensional assimilation in the presence of baroclinic instability. *Quart. J. Roy. Meteor. Soc.*, **118,** 649-672

Tanguay, M., Bartello, P. and Gauthier, P., 1995: Four-dimensional data assimilation with a wide range of scales. *Tellus,* **47A,** 974-997

Tanguay, M., Anderson, D.L.T., Bennett, A.F., Busalacchi, A.J., Cohn, S.E., Courtier, P., Derber, J., Lorenc, A.C., Parrish, D., Purser, J., Sato, N. and Schlatter, T, 1997: Data Assimilation in the Ocean and in the Atmosphere: What Should be Next ? *J. Meteorol. Soc. Japan*, **75-1B,** 489-496

Thacker, W.C., 1989: The Role of the Hessian Matrix in Fitting Models to Measurements. *J. Geophys. Res.*, **94,** 6177-6196

Thépaut, J.-N., Courtier, P., Belaud, G. and Lemaître, G., 1996: Dynamical structure functions in a four-dimensional variational assimilation: a case study. *Quart. J. Roy. Meteor. Soc.*, **122,** 535-561

Veersé, F., 1997: *Stratégies de minimisation pour le calcul de l'état initial en météorologie.* Ph.D. thesis in Applied Mathematics, Université de Bordeaux 1, France.

Veersé, F. and Thépaut, J.-N., 1998: Multiple-truncation incremental approach for four-dimensional variational data assimilation. *Quart. J. Roy. Meteor. Soc.*, **124,** 1889-1908

Wunsch, C., 1996: *The Ocean Circulation Inverse Problem.* Cambridge University Press, 422 pp.

Yang, W., Navon, I.M. and Courtier, P., 1996: A new Hessian preconditioning method applied to variational data assimilation experiments using the adiabatic version of NASA/GEOS-1 GCM. *Mon. Wea. Rev.*, **124,** 1000-1017

Unit´e de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unit´e de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unit´e de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unit´e de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unit´e de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399