Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique

B-Preconditioned Minimization Algorithms for Variational Data Assimilation with the Dual Formulation

S. Gürol, A. T. Weaver, A. M. Moore, A. Piacentini, H. G. Arango and S. Gratton

Technical Report TR/PA/12/118

Publications of the Parallel Algorithms Team

http://www.cerfacs.fr/algor/publications/

Abstract

Variational data assimilation problems arising in meteorology and oceanography require the solution of a regularized nonlinear least-squares problem. Practical solution algorithms are based on the incremental (Truncated Gauss-Newton) approach, which involves the iterative solution of a sequence of linear least-squares (quadratic minimization) sub-problems. Each sub-problem can be solved using a primal approach, where the minimization is performed in a space spanned by vectors of the size of the model control vector, or a dual approach, where the minimization is performed in a space spanned by vectors of the size of the observation vector. The dual formulation can be advantageous for two reasons. First, the dimension of the minimization problem with the dual formulation does not increase when additional control variables, such as those accounting for model error in a weak-constraint formulation, are considered. Second, whenever the dimension of observation space is significantly smaller than that of the model control space, the dual formulation can reduce both memory usage and computational cost.

In this paper, a new dual-based algorithm called Restricted **B**-preconditioned Lanczos (RBLanczos) is introduced, where **B** denotes the background-error covariance matrix. RBLanczos is the Lanczos formulation of the Restricted **B**-preconditioned Conjugate Gradient (RBCG) method. RBLanczos generates mathematically equivalent iterates to those of RBCG and the corresponding **B**-preconditioned Conjugate Gradient and Lanczos algorithms used in the primal approach. All these algorithms can be implemented without the need for a square-root factorization of **B**. RBCG and RBLanczos, as well as the corresponding primal algorithms, are implemented in two operational ocean data assimilation systems and numerical results are presented. Practical diagnostic formulae for monitoring the convergence properties of the minimization are also presented.

Keywords: 3D-Var, 4D-Var, PSAS, conjugate gradient method, Lanczos method, dual approach, ocean data assimilation

B-Preconditioned Minimization Algorithms for Variational Data Assimilation with the Dual Formulation

S. Gürol[*], A. T. Weaver[†], A. M. Moore[‡], A. Piacentini[§], H. G. Arango[¶] and S. Gratton[||]

December 18, 2012

1 Introduction

Variational assimilation seeks to solve a regularized nonlinear least-squares problem to determine a model state that optimally fits both observational information and *a priori* information in the form of a model background state. The fit is quantified by a cost function that measures the sum of the weighted squared differences between the available information (observations and background state) and the corresponding model-predicted fields. The weights are defined by matrix operators that define the error statistics (inverse error covariance) of the information.

The basic problem involves the optimization of a set of n control variables given a set of $(m+n)$ pieces of information where m and n denote the number of physical observations and background estimates of the control variables, respectively. Iterative techniques must be used to identify an approximate minimum of the cost function when n is large. In meteorological applications, variational assimilation is implemented using an iterative technique based on the incremental approach (Courtier *et al.*, 1994), which in optimization theory is known as a Truncated Gauss-Newton (TGN) method (Lawless *et al.*, 2005; Gratton *et al.*, 2007). This approach is also widely used in oceanographic applications (Weaver *et al.*, 2003; Moore *et al.*, 2011a).

The incremental approach solves a sequence of linear least-squares (quadratic minimization) problems where each member of the sequence is a local quadratic approximation of the original nonlinear least-squares problem. Conjugate Gradient (CG) or Lanczos methods, which belong to the general class of Krylov subspace methods, are effective for

[*] CERFACS, Toulouse, France

[†] CERFACS/SUC URA 1875, Toulouse, France

[‡] Department of Ocean Sciences, University of California at Santa Cruz, Santa Cruz, CA, USA

[§] CERFACS, Toulouse, France

[¶] Institute of Marine and Coastal Sciences, The Rutgers University of New Jersey, NJ, USA

[||] CERFACS, Toulouse, France / ENSEEIHT, Toulouse, France

solving quadratic minimization problems when n is large and when the system (Hessian) matrix is symmetric and positive definite, and only available in operator form (i.e., as a matrix-vector product). When the quadratic minimization is performed directly in $\mathbb{R}^n$, the method is referred to as the *primal approach.* Alternatively, the solution can be found using the so-called *dual approach* (Egbert *et al.*, 1994; Courtier, 1997; Daley and Barker, 2001; Bennett, 2002) which performs the quadratic minimization in $\mathbb{R}^m$. The solution in $\mathbb{R}^m$ is then mapped to $\mathbb{R}^n$ through the application of an $n \times m$ matrix operator that defines the background-error covariances of the model-equivalent of the observations with the control variables.

The dual approach can provide a significant reduction in the computational cost and storage when $m \ll n$ since all the recurrence formulae in the minimization algorithm involve m-dimensional vectors instead of n-dimensional vectors as in the primal approach. For weak-constraint variational assimilation problems (Courtier, 1997; Bennett, 2002; Trémolet, 2006), n can be very large since the control vector includes a time-sequence of corrective terms or model states in order to account for model error. For those problems, the primal approach may become intractable.

The dual approach of Courtier (1997) consists of solving the quadratic minimization problem in $\mathbb{R}^m$ with a first-level preconditioner given by the inverse of the observation-error covariance matrix ($\mathbf{R}^{-1}$). The preconditioner was applied using a factorized form involving $\mathbf{R}^{-1/2}$. The method is commonly referred to as PSAS which refers to the Physical-space Statistical Analysis System (Cohn *et al.*, 1998) where it was applied. However, it has been shown by El Akkraoui *et al.* (2008), Gratton and Tshimanga (2009) (hereafter referred to as GT09) and El Akkraoui and Gauthier (2010) that this version of the dual approach does not generate corresponding n-dimensional iterates that reduce monotonically the quadratic cost function in $\mathbb{R}^n$. A prohibitively large number of iterations may then be required to obtain an acceptable solution, which should be measured by the reduction of the cost function in primal space (El Akkraoui and Gauthier, 2010). If the minimization is terminated after a limited number of iterations, it can even yield a result that is inferior to the initial guess. El Akkraoui and Gauthier (2010) demonstrate experimentally that this problem can be alleviated by minimizing the ($\mathbf{R}^{-1}$–preconditioned) cost function in $\mathbb{R}^m$ using the MINimum RESidual (MINRES) method, although no mathematical proof was provided to support this result.

Another dual algorithm known as the Restricted Preconditioned Conjugate Gradient (RPCG) method has been proposed by GT09. This method again performs the minimization in $\mathbb{R}^m$, but contrary to PSAS, it generates mathematically equivalent iterates to those of the primal approach in which the cost function is minimized using a CG method. This allows the dual approach to benefit from the computational savings when $m < n$, while preserving the desired convergence properties of the primal approach.

In this paper, we derive a new dual algorithm called the Restricted **B**-preconditioned Lanczos method (RBLanczos) where **B** is the background-error covariance matrix. RBLanczos generates mathematically identical iterates to the **B**-preconditioned Lanczos algorithm often used in primal approaches. This algorithm can also be interpreted as the Lanczos version of a special case of RPCG in which the corresponding primal first-level preconditioner is **B**. We call this specific algorithm Restricted **B**-preconditioned Conjugate Gradient (RBCG). The Lanczos vectors, which are directly computed by RBLanczos, are important for preconditioning (Tshimanga *et al.*, 2008; Fisher *et al.*, 2009; Desroziers and Berre, 2012) and for quantifying the performance of the data assimilation system (Cardinali *et al.*, 2004; Gelaro and Zhu, 2009; Moore *et al.*, 2011c).

A practical demonstration of the benefits of RBCG and RBLanczos is provided using two operationally-based variational data assimilation systems for the ocean. RBCG is implemented in a three-dimensional variational assimilation (3D-Var) system for a global configuration of the NEMO (Nucleus for European Modelling of the Ocean) model. This system, called NEMOVAR, is used for operational monthly/seasonal forecasting and ocean reanalysis at the European Centre for Medium Range Weather Forecasts (ECMWF) (Mogensen *et al.*, 2012; Balmaseda *et al.*, 2012). RBLanczos is implemented in a four-dimensional variational assimilation (4D-Var) system for a California Current configuration of the ROMS (Regional Ocean Modeling Systems) model. This system, which includes a weak-constraint formulation of 4D-Var, is used for regional ocean forecasting and reanalysis applications (Moore *et al.*, 2011a,b,c).

The outline of the paper is as follows. In section 2 the variational assimilation problem is formulated and the Lanczos algorithm for the primal approach is introduced. The RBLanczos algorithm is given in the same section. Practical formulae for diagnosing quantities needed for monitoring the convergence of the algorithm are also derived. Numerical experiments with NEMOVAR and ROMS are presented in section 3. Conclusions and future directions are given in section 4. Appendix A derives the relations between the vectors defined in the primal and dual formulations of the Lanczos algorithm. Appendix B provides the **B**-preconditioned Conjugate Gradient (BCG) algorithm, while Appendix C provides the RBCG algorithm. These algorithms have been extended to include terms that can be used for monitoring convergence as in the Lanczos algorithms. Practical formulae for computing these diagnostics with RBCG and BCG are given in Appendix D.

2 Lanczos method with the primal and dual approaches

2.1 Problem Formulation

In its standard formulation, 4D-Var is designed to estimate the initial state of a dynamical system by combining observations over a given time window with an *a priori* estimate of the initial conditions called the background state. This approach has its origin in maximum likelihood estimation under a Gaussian assumption (Tarantola, 2005, pp. 24-32). It leads to the minimization of the cost function

$$\mathcal{J}[\mathbf{u}] = \frac{1}{2}\left(\mathbf{u}-\mathbf{u}_\mathrm{b}\right)^\mathrm{T}\mathbf{B}^{-1}\left(\mathbf{u}-\mathbf{u}_\mathrm{b}\right) + \frac{1}{2}\sum_{j=0}^{N_t}\left(H_j(\mathbf{x}_j)-\mathbf{y}_j\right)^\mathrm{T}\mathbf{R}_j^{-1}\left(H_j(\mathbf{x}_j)-\mathbf{y}_j\right) \tag{1}$$

with respect to a control vector $\mathbf{u} = \mathbf{x}(t_0)$, chosen to be the initial state of the dynamical model at time t_0. In this formulation, observations are given by an m_j-dimensional vector $\mathbf{y}_j$ at time t_j, and the background state is given by an n-dimensional vector $\mathbf{u}_\mathrm{b} = \mathbf{x}_\mathrm{b}(t_0)$. The $n \times n$ matrix $\mathbf{B}$ is an estimate of the background-error covariance matrix, and the $m_j \times m_j$ matrix $\mathbf{R}_j$ is an estimate of the observation-error covariance matrix at time t_j, with the observation errors assumed here to be uncorrelated in time. The inverse of these matrices defines the weighting matrix of the quadratic terms in Eq. (1). In order to calculate the model counterpart of the observation vector at t_j, first the state $\mathbf{x}_j = \mathbf{x}(t_j) = M_{0,j}(\mathbf{x}(t_0))$ is estimated by propagating the initial state to time t_j using the dynamical model operator $M_{0,j} \equiv M(t_0, t_j)$, and then the state is mapped to observation space using the observation operator H_j. Note that this formulation of 4D-Var assumes that the model operator $M_{0,j}$ is perfect. Adopting the terminology of Sasaki (1970), it is referred to as strong-constraint 4D-Var.

The incremental approach (Courtier *et al.*, 1994) is a practical algorithm for solving the 4D-Var problem when the system is weakly nonlinear and of large dimension n. Incremental 4D-Var consists of solving a sequence of linear least-squares approximations of the nonlinear least-squares problem (1). On each iteration (k), a quadratic cost function

$$\begin{aligned} J[\delta\mathbf{u}^{(k)}] &= \frac{1}{2}\left(\mathbf{u}^{(k-1)}-\mathbf{u}_\mathrm{b}+\delta\mathbf{u}^{(k)}\right)^\mathrm{T}\mathbf{B}^{-1}\left(\mathbf{u}^{(k-1)}-\mathbf{u}_\mathrm{b}+\delta\mathbf{u}^{(k)}\right) \\ &+ \frac{1}{2}\sum_{j=0}^{N_t}\left(\mathbf{H}_j^{(k-1)}\mathbf{M}_{0,j}^{(k-1)}\delta\mathbf{u}^{(k)}-\mathbf{d}_j^{(k-1)}\right)^\mathrm{T}\mathbf{R}_j^{-1}\left(\mathbf{H}_j^{(k-1)}\mathbf{M}_{0,j}^{(k-1)}\delta\mathbf{u}^{(k)}-\mathbf{d}_j^{(k-1)}\right) \end{aligned} \tag{2}$$

is minimized to determine a correction (increment) $\delta\mathbf{u}^{(k)} = \delta\mathbf{x}^{(k)}(t_0)$ to the initial state estimate $\mathbf{u}^{(k-1)} = \mathbf{x}^{(k-1)}(t_0)$ such that the updated estimate is

$$\mathbf{u}^{(k)} = \mathbf{u}^{(k-1)} + \delta\mathbf{u}^{(k)}.$$

The initial estimate is usually taken to be the background state, $\mathbf{u}^{(0)} = \mathbf{u}_\mathrm{b}$. In the quadratic formulation (2), $\mathbf{M}_{0,j}^{(k-1)} = \mathbf{M}^{(k-1)}(t_0, t_j)$ is the tangent-linear (TL) of the nonlinear model $M_{0,j}$ defined with respect to the time sequence of reference states $\left\{\mathbf{x}_i^{(k-1)}\right\}$, $i = 1, \ldots, j$, and $\mathbf{H}_j^{(k-1)}$ is the TL of the observation operator $H_j^{(k)}$ defined with respect to $\mathbf{x}_j^{(k-1)}$. In practice, the TL operators are often approximated. The vector $\mathbf{d}_j^{(k-1)} = \mathbf{y}_j - H_j(\mathbf{x}_j^{(k-1)})$ is the difference between the observation vector and the corresponding model-predicted values at time t_j. The main loop of the incremental 4D-Var algorithm that generates the time sequence of states $\left\{\mathbf{x}_j^{(k-1)}\right\}$ and difference vectors $\left\{\mathbf{d}_j^{(k-1)}\right\}$, for $j = 1, \ldots, N_t$, is called the *outer loop*. For large problems, an iterative method is used to solve the quadratic minimization problem (2). The iterative loop of the quadratic minimization problem is called the *inner loop* since it is nested within the outer loop.

This paper focuses on minimization algorithms for the inner loop. For clarity and without loss of generality, we consider a single outer iteration ($k = 1$) starting from the background state $\mathbf{u}_\mathrm{b}$. For this special case, the quadratic cost function of the inner-loop problem can be written as

$$J[\delta\mathbf{u}] = \frac{1}{2}\delta\mathbf{u}^\mathrm{T}\mathbf{B}^{-1}\delta\mathbf{u} + \frac{1}{2}(\mathbf{G}\,\delta\mathbf{u} - \mathbf{d})^\mathrm{T}\mathbf{R}^{-1}(\mathbf{G}\,\delta\mathbf{u} - \mathbf{d}), \tag{3}$$

which is a compact form of the linearized problem (2) with $k = 1$ and $\mathbf{u}^{(0)} = \mathbf{u}_\mathrm{b}$. The outer iteration counter (k) has been dropped for clarity of notation. In Eq. (3), the generalized observation operator $\mathbf{G}$ is a $m \times n$ matrix of concatenated operators $\mathbf{H}_j\mathbf{M}_{0,j}$ over time where $m = \sum_{j=0}^{N_t} m_j$, $\mathbf{R}$ is a $m \times m$ block-diagonal matrix whose jth block is $\mathbf{R}_j$, and $\mathbf{d}$ is the vector of concatenated difference vectors $\mathbf{d}_j$.

The exact solution $\delta\mathbf{u}^*$ of Eq. (3) is obtained by setting the gradient of the cost function to zero (Tarantola, 2005; Nocedal and Wright, 2006), which yields

$$\delta\mathbf{u}^* = (\mathbf{B}^{-1} + \mathbf{G}^\mathrm{T}\mathbf{R}^{-1}\mathbf{G})^{-1}\mathbf{G}^\mathrm{T}\mathbf{R}^{-1}\mathbf{d}. \tag{4}$$

Since the matrices appearing in Eq. (4) are large and typically only available in operator form (i.e., as a matrix-vector product), an approximate solution is usually found by solving the $n \times n$ linear system

$$(\mathbf{B}^{-1} + \mathbf{G}^\mathrm{T}\mathbf{R}^{-1}\mathbf{G})\,\delta\mathbf{u} = \mathbf{G}^\mathrm{T}\mathbf{R}^{-1}\mathbf{d}, \tag{5}$$

using a Krylov subspace iterative method (Saad, 1996, Chapter 6). Equation (5), which involves directly optimizing in control space to determine $\delta\mathbf{u}$, is referred to as the *primal* problem.

Alternatively, using the Sherman-Morrison-Woodbury formula (Nocedal and Wright, 2006, pp. 612-613) or from duality theory (Gratton *et al.*, 2012), the solution (4) can be expressed as

$$\delta\mathbf{u}^* = \mathbf{B}\mathbf{G}^\mathrm{T}(\mathbf{G}\mathbf{B}\mathbf{G}^\mathrm{T} + \mathbf{R})^{-1}\mathbf{d}. \tag{6}$$

An approximate solution of Eq. (6) can be obtained by applying a Krylov subspace method to the $m \times m$ linear system

$$(\mathbf{G}\mathbf{B}\mathbf{G}^\mathrm{T} + \mathbf{R})\boldsymbol{\lambda} = \mathbf{d}, \tag{7}$$

and then transforming the solution as

$$\delta\mathbf{u} = \mathbf{B}\mathbf{G}^\mathrm{T}\boldsymbol{\lambda}. \tag{8}$$

In ocean data assimilation this approach has been referred to as the indirect representer method (Egbert *et al.*, 1994) whereas in meteorological data assimilation has been called PSAS (Courtier, 1997). Equations (7) and (8), which involve optimizing the m-dimensional vector $\boldsymbol{\lambda}$, constitute the *dual* problem. The dual problem can be preferable to the primal problem for computational reasons when the dimension (m) of observation space is much smaller than the dimension (n) of control space.

For some problems it may be desirable to include additional variables in the control vector $\mathbf{u}$. For example, extra control variables can be included to account for errors in the dynamical model $M_{0,j}$, leading to the so-called weak-constraint formulation of 4D-Var (Sasaki, 1970), or to account for errors in boundary conditions. The dimension of the primal problem is determined by the size of the control vector. In contrast, the dimension of the dual problem is determined by the size of the observation vector and thus does not change by including additional control variables. For weak-constraint 4D-Var, the number of extra variables can be so large that the primal problem becomes impractical. In such cases, the dual problem is particularly appealing.

2.2 Solving the linearized problem: primal approaches

Equation (5) is a problem of the standard form

$$\mathbf{A}\,\delta\mathbf{u} = \mathbf{b}$$

where

$$\mathbf{A} = \nabla^2 J = \mathbf{B}^{-1} + \mathbf{G}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{G} \tag{9}$$

is the (symmetric and positive-definite) Hessian or system matrix, and

$$\mathbf{b} = -\nabla J[\mathbf{0}] = \mathbf{G}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{d} \tag{10}$$

is the negative of the cost function gradient evaluated at $\delta\mathbf{u} = \mathbf{0}$. Krylov subspace methods search for an approximate solution $\delta\mathbf{u}_l$ in a subspace $\delta\mathbf{u}_0 + \mathcal{K}^l(\mathbf{A}, \mathbf{r}_0)$ where $\delta\mathbf{u}_0$ is the initial guess,

$$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\,\delta\mathbf{u}_0 = -\nabla J[\delta\mathbf{u}_0] \tag{11}$$

is the initial residual (the negative of the gradient evaluated at $\delta\mathbf{u} = \delta\mathbf{u}_0$), and

$$\mathcal{K}^l(\mathbf{A}, \mathbf{r}_0) = \mathrm{span}\Big\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \ldots, \mathbf{A}^{l-1}\mathbf{r}_0\Big\}$$

is the Krylov subspace of dimension l.

To define the lth iterate uniquely, several Krylov subspace methods impose the Petrov-Galerkin condition (Saad, 1996, p.144)

$$\mathbf{r}_l \perp \mathcal{L}^l(\mathbf{A}, \mathbf{r}_0), \tag{12}$$

where $\mathbf{r}_l$ is the residual of the lth iterate, and $\mathcal{L}^l(\mathbf{A}, \mathbf{r}_0)$ is an l-dimensional subspace. The choice of the subspace $\mathcal{L}^l(\mathbf{A}, \mathbf{r}_0)$ and properties of the matrix $\mathbf{A}$ yield different Krylov subspace methods. For instance, when $\mathcal{L}^l(\mathbf{A}, \mathbf{r}_0) = \mathcal{K}^l(\mathbf{A}, \mathbf{r}_0)$, the condition (12) is called a Galerkin condition. It leads to the Lanczos method when $\mathbf{A}$ is symmetric, which finds an approximate solution $\delta\mathbf{u}_l$ by minimizing the $\mathbf{A}^{-1}$-norm of the residual; $\|\mathbf{r}_l\|_{\mathbf{A}^{-1}} = \sqrt{\mathbf{r}_l^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{r}_l}$. On the other hand, choosing $\mathcal{L}^l(\mathbf{A}, \mathbf{r}_0) = \mathbf{A}\,\mathcal{K}^l(\mathbf{A}, \mathbf{r}_0)$ with $\mathbf{A}$ symmetric gives the Minimum RESidual method (MINRES). It finds the approximate solution $\delta\mathbf{u}_l$ by minimizing the Euclidean norm of the residual; $\|\mathbf{r}_l\|_2 = \sqrt{\mathbf{r}_l^{\mathrm{T}}\mathbf{r}_l}$. Furthermore, if $\mathbf{A}$ is symmetric and positive definite then the Lanczos method is mathematically equivalent to the CG method (Saad, 1996, p.176).

This paper focuses on the Lanczos method which is widely used in meteorological and ocean variational data assimilation (Tshimanga *et al.*, 2008; Fisher *et al.*, 2009; Moore *et al.*, 2011a). As mentioned above, the Lanczos method imposes the Galerkin condition which implies

$$\mathbf{V}_l^{\mathrm{T}}(\mathbf{b} - \mathbf{A}\delta\mathbf{u}_l) = 0$$

where $\mathbf{V}_l = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_l]$ is an $n \times l$ matrix whose column vectors $\mathbf{v}_i$, $i = 1, \ldots, l$, form an orthonormal basis for $\mathcal{K}^l(\mathbf{A}, \mathbf{r}_0)$. The basis vectors are known as Lanczos vectors and are constructed by using a simplified version of Arnoldi's algorithm when $\mathbf{A}$ is symmetric (Saad, 1996, p. 174-175). They can be shown to be related to the normalized residual (gradient) vectors generated by CG on the same iterations (Paige and Saunders, 1975). Using this orthonormal basis, the solution on the lth iteration becomes

$$\delta\mathbf{u}_l = \delta\mathbf{u}_0 + \mathbf{V}_l\mathbf{s}_l, \tag{13}$$

where $\mathbf{s}_l$ is the solution of the linear system

$$\mathbf{T}_l\mathbf{s}_l \;=\; \mathbf{V}_l^{\mathrm{T}}\mathbf{r}_0 \tag{14}$$

involving the $l \times l$ tridiagonal matrix

$$\mathbf{T}_l \;=\; \mathbf{V}_l^{\mathrm{T}}\mathbf{A}\mathbf{V}_l. \tag{15}$$

The tridiagonal system (14) which is of size l, l being typically small, can be easily solved using standard factorization methods (Golub and Van Loan, 1996, p. 138-139). The eigenvalues of $\mathbf{T}_l$ are approximate eigenvalues (also called Ritz values) of the matrix $\mathbf{A}$, while $\mathbf{V}_l$ times the eigenvectors of $\mathbf{T}_l$ are the corresponding approximate eigenvectors (Ritz vectors) of $\mathbf{A}$ (Saad, 1996, p. 152-153). The right-hand side of Eq. (14) can be simplified by recalling that the vectors $\mathbf{v}_i$, $i = 1, \ldots, l$, form an orthonormal basis for $\mathcal{K}^l(\mathbf{A}, \mathbf{r}_0)$ and that $\mathbf{v}_0 = \mathbf{r}_0/\|\mathbf{r}_0\|_2$, so that

$$\mathbf{V}_l^{\mathrm{T}}\mathbf{r}_0 = \|\mathbf{r}_0\|_2\,\mathbf{e}_1 \tag{16}$$

where $\|\mathbf{r}_0\|_2 = \sqrt{\mathbf{r}_0^{\mathrm{T}}\mathbf{r}_0}$ and $\mathbf{e}_1 = [1, 0, \ldots, 0]^{\mathrm{T}}$.

When applying the Lanczos or CG method, a *preconditioner* is desirable to accelerate the convergence. A preconditioner typically transforms the linear system into one with "better" spectral properties; e.g., more clustered eigenvalues. At the very least, a *first-level* preconditioner is needed to non-dimensionalize the control vector when it involves more than one physical variable. In this way, results obtained using the standard implementations of Lanczos or CG (with the canonical inner product) will be independent of the choice of physical units. This can be easily done by separating out the units using the factorization $\mathbf{B} = \mathbf{D}^{1/2}\mathbf{C}\mathbf{D}^{1/2}$ where $\mathbf{D}^{1/2}$ is a diagonal matrix of (dimensional) standard deviations, and $\mathbf{C}$ is a (dimensionless) correlation matrix, with 1s on the diagonal. Equation (5) can then transformed as

$$(\mathbf{C}^{-1} + \mathbf{D}^{1/2}\mathbf{G}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{G}\,\mathbf{D}^{1/2})\,\delta\widetilde{\mathbf{u}} = \mathbf{D}^{1/2}\mathbf{G}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{d} \tag{17}$$

and solved for $\delta\widetilde{\mathbf{u}} = \mathbf{D}^{-1/2}\delta\mathbf{u}$ using Lanczos or CG equipped with the canonical inner product. The solution in control space is then obtained from $\delta\mathbf{u} = \mathbf{D}^{1/2}\delta\widetilde{\mathbf{u}}$.

Equation (17) is poorly conditioned in general since typical correlation matrices $\mathbf{C}$ used in variational data assimilation contain a wide range of eigenvalues (Lorenc, 1997). It also requires specification of the inverse correlation operator $\mathbf{C}^{-1}$ which is difficult in practice. To alleviate these problems, the full background-error covariance matrix $\mathbf{B}$ is used as a first-level preconditioner. Assuming $\mathbf{B}$ can be factored as $\mathbf{B} = \mathbf{U}\mathbf{U}^{\mathrm{T}}$, where $\mathbf{U}$ is a $n \times n'$ left square-root matrix of $\mathbf{B}$, with n' possibly different from n, then the preconditioned system becomes

$$(\mathbf{I}_{n'} + \mathbf{U}^{\mathrm{T}}\mathbf{G}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{G}\mathbf{U})\,\delta\widetilde{\mathbf{u}} \;=\; \mathbf{U}^{\mathrm{T}}\mathbf{G}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{d} \tag{18}$$

where $\mathbf{I}_{n'}$ is the $n' \times n'$ identity matrix. Equation (18) is solved for $\delta\widetilde{\mathbf{u}}$ using Lanczos or CG formulated with the canonical inner product, and the solution in control space is obtained from $\delta\mathbf{u} = \mathbf{U}\,\delta\widetilde{\mathbf{u}}$. The eigenvalue spectrum of the system matrix in Eq. (18) is bounded below by 1, and has a cluster at 1 of size at least $\max(0, n' - m)$. A *second-level* preconditioner that approximates the inverse of the linear system matrix can also be used to further accelerate the convergence. Tshimanga *et al.* (2008) and Gratton *et al.* (2011b) discuss second-level preconditioning techniques within the context of multi-outer-loop iterations of incremental 4D-Var. In this paper, however, only first-level preconditioning with $\mathbf{B}$ will be considered. Furthermore, the initial guess $\delta\mathbf{u}_0$ will be assumed to be zero, as is typically the case in applications of 3D-Var and 4D-Var that involve a single outer-loop iteration.

Notice that the dimension of minimization space for solving Eq. (18) is determined by n', the dimension of $\delta\widetilde{\mathbf{u}}$. The case $n' < n$ arises with reduced-rank formulations of $\mathbf{B}$ where, for example, $\mathbf{U}$ is defined by a limited number of ensemble perturbations or some other appropriately chosen basis vectors (Evensen, 2009; Gratton *et al.*, 2011a). Square-root preconditioning is clearly advantageous when $n' \ll n$. The case $n' > n$ arises when $\mathbf{B}$ is formulated as a weighted sum of two or more matrices. This situation can arise with general covariance models constructed from spectral or grid-point filters (Fisher, 2003;

Purser *et al.*, 2003; Weaver and Mirouze, 2012) and with *hybrid* ensemble-variational **B** formulations (Buehner, 2005; Wang *et al.*, 2007, 2008). Square-root preconditioning can be less convenient in these latter cases.

The preconditioned linear system (18) can be written in the standard form

$$\widetilde{\mathbf{A}}\,\delta\widetilde{\mathbf{u}} = \widetilde{\mathbf{b}} \tag{19}$$

where

$$\begin{aligned} \widetilde{\mathbf{A}} &= \mathbf{U}^{\mathrm{T}}\mathbf{A}\mathbf{U} \\ \text{and} \quad \widetilde{\mathbf{b}} &= \mathbf{U}^{\mathrm{T}}\mathbf{b}, \end{aligned}$$

with **A** and **b** given by Eqs (9) and (10). The Lanczos algorithm applied to Eq. (19) searches for a solution in the Krylov subspace $\mathcal{K}^l(\widetilde{\mathbf{A}}, \widetilde{\mathbf{r}}_0)$ where $\widetilde{\mathbf{r}}_0 = \mathbf{U}^{\mathrm{T}}\mathbf{r}_0$. Applying the standard version of the Lanczos algorithm described by Eqs (13)–(15) to the system (19), with the assumption $\delta\mathbf{u}_0 = \mathbf{0}$, gives an approximate solution on the lth iteration as

$$\delta\widetilde{\mathbf{u}}_l = \widetilde{\mathbf{V}}_l\mathbf{s}_l, \tag{20}$$

where

$$\left.\begin{aligned} \mathbf{T}_l\mathbf{s}_l &= \widetilde{\mathbf{V}}_l^{\mathrm{T}}\,\widetilde{\mathbf{r}}_0, \\ \text{and} \quad \mathbf{T}_l &= \widetilde{\mathbf{V}}_l^{\mathrm{T}}\widetilde{\mathbf{A}}\widetilde{\mathbf{V}}_l. \end{aligned}\right\} \tag{21}$$

The Lanczos algorithm applied to the preconditioned linear system (19) is based on the availability of a factored form for **B** (Fisher *et al.*, 2009). Hereafter we use the term Lanczos to refer to this particular form of the Lanczos algorithm. Alternatively, **B** preconditioning of the linear system (5) can be achieved by employing **B** as a *right* symmetric preconditioner (Nour-Omid *et al.*, 1988; Axelsson, 1996; Chan *et al.*, 1999). This leads to

$$(\mathbf{I}_n + \mathbf{G}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{G}\,\mathbf{B})\,\delta\overline{\mathbf{u}} \;=\; \mathbf{G}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{d}. \tag{22}$$

The system matrix in Eq. (22) is symmetric (self-adjoint) with respect to the **B**-inner product. The solution in control space itself is obtained from $\delta\mathbf{u} = \mathbf{B}\,\delta\overline{\mathbf{u}}$. This approach generates mathematically equivalent iterates to those of the Lanczos approach using **U** and $\mathbf{U}^{\mathrm{T}}$ but is more general since it does not require **B** to be factored, this factorization being particularly inconvenient when $n' \gg n$ as mentioned above.

The Lanczos algorithm that solves the linear system (22) with the **B**-inner product is given in Algorithm 2.1. For future reference, we refer to it as BLanczos.

BLanczos searches for a solution in the Krylov subspace (Chan *et al.*, 1999)

$$\mathcal{K}^l(\mathbf{BA}, \mathbf{Br}_0) = \mathrm{span}\Big\{\mathbf{Br}_0, (\mathbf{BA})\mathbf{Br}_0, \ldots, (\mathbf{BA})^{l-1}\mathbf{Br}_0\Big\}\,. \tag{23}$$

It can be easily shown that the basis $\mathbf{V}_l$ constructed by BLanczos is related to the basis $\widetilde{\mathbf{V}}_l$ via the relation $\widetilde{\mathbf{V}}_l = \mathbf{U}^{\mathrm{T}}\mathbf{V}_l$. From this relation, Eqs (20) and (21), and the definitions of $\widetilde{\mathbf{A}}$ and $\widetilde{\mathbf{r}}_0$, it is straightforward to show that the solution on the lth iteration of BLanczos can be written as

$$\delta\mathbf{u}_l = \mathbf{Z}_l\mathbf{s}_l, \tag{24}$$

where

$$\left.\begin{array}{rrcl} & \mathbf{Z}_l & = & \mathbf{B}\mathbf{V}_l, \\ & \mathbf{T}_l\,\mathbf{s}_l & = & \mathbf{Z}_l^{\mathrm{T}}\mathbf{r}_0, \\ \text{and} & \mathbf{T}_l & = & \mathbf{Z}_l^{\mathrm{T}}\mathbf{A}\mathbf{Z}_l. \end{array}\right\} \tag{25}$$

From the **B**-orthonormality of the vectors $\mathbf{z}_i$, $i = 1, \ldots, l$, the right-hand side of the second expression in Eq. (25) can be simplified as (cf. Eq. (16))

$$\mathbf{Z}_l^{\mathrm{T}}\mathbf{r}_0 = \beta_0\,\mathbf{e}_1$$

where $\beta_0 = \|\mathbf{r}_0\|_{\mathbf{B}} = \sqrt{\mathbf{r}_0^{\mathrm{T}}\mathbf{B}\mathbf{r}_0}$.

Each iteration of BLanczos requires only one matrix-vector multiplication with each of the matrices $\mathbf{G}$, $\mathbf{G}^{\mathrm{T}}$, $\mathbf{R}^{-1}$ and $\mathbf{B}$. The $n \times l$ matrix $\mathbf{Z}_l$ and the $l \times l$ matrix $\mathbf{T}_l$ need to be stored for computing the solution with Eq. (24). As mentioned earlier, these matrices contain approximate information on the eigenspectrum of the Hessian matrix which is valuable for diagnostic studies and for building second-level preconditioners. The matrix $\mathbf{Z}_l$ is also required for re-orthogonalization with the matrix $\mathbf{V}_l$ as explained in section 2.5.

When re-orthogonalization is not necessary, the solution can also be found using recurrence relationships that do not require the matrices $\mathbf{Z}_l$ and $\mathbf{T}_l$ (Papadrakakis and Smerou, 1990; Saad, 1996; Chien and Chang, 2003). Therefore, the memory requirements can be reduced by using this version of the Lanczos algorithm (known also as the direct version of the Lanczos algorithm (Saad, 1996, p.177)) which is derived from the LU factorization of $\mathbf{T}_l$. This direct version of BLanczos is given in Algorithm 2.2.

As mentioned earlier, another well-known method for solving the linear system (5) is CG. CG is mathematically equivalent to the Lanczos method in exact arithmetic when the system matrix **A** is symmetric and positive definite. A **B**–preconditioned CG algorithm for solving the linear system (5), which we call BCG, is given in Appendix B. Like BLanczos, BCG only requires **B** for preconditioning (not its factorization or inverse) and only one application per iteration of the matrix operators $\mathbf{G}$, $\mathbf{G}^{\mathrm{T}}$, $\mathbf{R}^{-1}$ and $\mathbf{B}$ (Derber and Rosati, 1989).

Another way to avoid specifying a factorization of **B** is to treat Eq. (22) as a nonsymmetric system (with respect to the canonical inner product) and to solve it with the Full Orthogonalization Method (FOM), which is a generalization of the Lanczos method that does not require **A** to be symmetric but is more computationally expensive. El Akkraoui *et al.* (2012) used a BiConjugate Gradient (BiCG) algorithm[1] (Saad, 1996, p. 211-212) to solve the nonsymmetric system (22) with second-level preconditioning. BiCG is less robust numerically than PCG (Golub and Van Loan, 1996, p. 551), although this problem can be handled by using the BiCG Stabilized method (Saad, 1996, p. 217). When using only first-level (**B**) preconditioning, El Akkraoui *et al.* (2012) show that BiCG is mathematically equivalent to BCG or "double CG" as referred to in their paper.

2.3 Solving the linearized problem: dual approaches

Alternatively, the solution of the linearized problem (3) can be found by solving the dual problem using a CG or Lanczos method. By analogy with the use of **B** as a preconditioner for the primal problem, one possibility is to use $\mathbf{R}^{-1}$ as a preconditioner for the dual problem. This is the basic approach taken in PSAS. For example, employing $\mathbf{R}^{-1}$ *left* symmetric preconditioning on Eq. (7) leads to solving the system

$$(\mathbf{R}^{-1}\mathbf{G}\mathbf{B}\mathbf{G}^{\mathrm{T}} + \mathbf{I}_m)\,\boldsymbol{\lambda} \;=\; \mathbf{R}^{-1}\mathbf{d} \tag{26}$$

using a CG or Lanczos algorithm formulated with the **R**-inner product. In many data assimilation systems, **R** is assumed to be diagonal in which case applying **R** and $\mathbf{R}^{-1}$ is trivial. Conventional implementations of PSAS (Courtier, 1997; El Akkraoui *et al.*, 2008; El Akkraoui and Gauthier, 2010) employ $\mathbf{R}^{-1}$ preconditioning via a square-root matrix $\mathbf{R}^{-1/2}$, where $\mathbf{R}^{-1} = \mathbf{R}^{-1/2}\mathbf{R}^{-1/2}$ (assuming **R** is diagonal), and solve

$$(\mathbf{R}^{-1/2}\mathbf{G}\mathbf{B}\mathbf{G}^{\mathrm{T}}\mathbf{R}^{-1/2} + \mathbf{I}_m)\,\widetilde{\boldsymbol{\lambda}} \;=\; \mathbf{R}^{-1/2}\mathbf{d}, \tag{27}$$

where $\boldsymbol{\lambda} = \mathbf{R}^{-1/2}\widetilde{\boldsymbol{\lambda}}$, using CG or Lanczos furnished with the canonical inner product.

As illustrated by GT09 and El Akkraoui and Gauthier (2010), PSAS has a nonmonotonic convergence behaviour when viewed in terms of the reduction of the quadratic cost function (3). This can yield a quadratic cost value that is larger than the initial value when the minimization is terminated after a few iterations. As a remedy, El Akkraoui and Gauthier (2010) suggested the use of MINRES to solve the linear system (27). Their experimental results showed that MINRES could produce a monotonically decreasing quadratic cost function (3), although no mathematical theory was provided to guarantee this behaviour.

A better alternative is the Restricted Preconditioned Conjugate Gradient (RPCG) method (GT09) which also solves Eq. (26) using a CG method but equipped with the (possibly semi-definite) $\mathbf{G}\mathbf{B}\mathbf{G}^{\mathrm{T}}$-inner product instead of the **R**-inner product. As discussed by GT09, RPCG generates, in exact arithmetic, the same iterates as those generated by PCG. This can also be achieved by solving Eq. (27) with the $\mathbf{R}^{-1/2}\mathbf{G}\mathbf{B}\mathbf{G}^{\mathrm{T}}\mathbf{R}^{-1/2}$-inner product instead of the canonical inner product as in PSAS.

[1] BiCG is referred to as BCG in Saad (1996) and should not be confused with the BCG algorithm described in this paper.

Since only first-level preconditioning with $\mathbf{B}$ is considered in this study, we restrict our attention to RBCG which is the dual equivalent of BCG and a special case of RPCG. For reference, the RBCG algorithm, which is based on Algorithm 5 of GT09, is provided in Appendix C. (See GT09 and Gratton *et al.* (2012) for a more general presentation of the algorithm that allows for second-level preconditioning).

In the same way that RBCG is the dual equivalent of BCG, there exists a Lanczos algorithm that is the dual equivalent of BLanczos. We call this algorithm Restricted BLanczos (RBLanczos). It produces identical iterates, in exact arithmetic, to RBCG, BCG and BLanczos. RBLanczos simply solves the linear system (26) using a Lanczos algorithm equipped with the (possibly semi-definite) $\mathbf{GBG}^{\mathrm{T}}$-inner product. The solution in control space is then recovered from Eq. (8). RBLanczos is a variant of the Restricted FOM (RSFOM) algorithm (Gratton *et al.*, 2009) adapted to symmetric and positive-definite matrices. The derivation of RBLanczos is sketched in the next section, with more technical details provided in Appendix A.

2.4 RBLanczos algorithm

Following GT09 and Gratton *et al.* (2009), we assume that the initial residual $\mathbf{r}_0 \in \mathbb{R}^n$ is in the range of the operator $\mathbf{G}^{\mathrm{T}}$. This assumption is valid when the initial guess $\delta\mathbf{u}_0$ is taken to be zero, which is the case here. When the initial guess is different from zero, the assumption still holds but a generalized algorithm with augmented vectors and matrices is required (see GT09).

The residual can be written as

$$\mathbf{r}_0 = \mathbf{G}^{\mathrm{T}}\widehat{\mathbf{r}}_0 \tag{28}$$

where $\widehat{\mathbf{r}}_0 = \mathbf{R}^{-1}\mathbf{d} \in \mathbb{R}^m$. From the basic assumption (28), it is shown in Appendix A that there exist m-dimensional vectors $\widehat{\mathbf{v}}_i$, $\widehat{\mathbf{z}}_i$, $\widehat{\mathbf{q}}_i$, $\widehat{\mathbf{w}}_i$ and $\widehat{\mathbf{t}}_i$ that can be related to the corresponding n-dimensional vectors $\mathbf{v}_i, \mathbf{z}_i, \mathbf{q}_i$, $\mathbf{w}_i$ and $\mathbf{t}_i$ of BLanczos Algorithm 2.1 according to

$$\left.\begin{array}{rcl} \mathbf{G}\mathbf{t}_i &=& \widehat{\mathbf{t}}_i, \\ \mathbf{v}_i &=& \mathbf{G}^{\mathrm{T}}\widehat{\mathbf{v}}_i \end{array}\right\} \tag{29}$$

for $i \geq 0$, and

$$\left.\begin{array}{rcl} \mathbf{G}\mathbf{z}_i &=& \widehat{\mathbf{z}}_i, \\ \mathbf{q}_i &=& \mathbf{G}^{\mathrm{T}}\widehat{\mathbf{q}}_i, \\ \mathbf{w}_i &=& \mathbf{G}^{\mathrm{T}}\widehat{\mathbf{w}}_i \end{array}\right\} \tag{30}$$

where $i \geq 1$.

Equations (28)-(30) allow all the recurrence relations in BLanczos Algorithm 2.1 involving vectors of dimension n to be transformed directly into corresponding recurrence relations involving vectors of dimension m. This yields the RBLanczos algorithm given by Algorithm 2.3. As with BLanczos, one loop of RBLanczos requires a single matrix-vector product with each of the operators $\mathbf{G}$, $\mathbf{G}^{\mathrm{T}}$, $\mathbf{R}^{-1}$ and $\mathbf{B}$, but with the important difference that all vectors are defined in $\mathbb{R}^m$ instead of $\mathbb{R}^n$. The two algorithms are mathematically equivalent.

As discussed in section 2.2, BLanczos searches for the solution $\delta\mathbf{u}_l$ in the Krylov subspace (23). It is instructive to determine the corresponding Krylov subspace for RBLanczos. Replacing $\mathbf{r}_0$ in Eq. (23) with Eq. (28) yields

$$\mathcal{K}^l(\mathbf{BA},\mathbf{BG}^{\mathrm{T}}\widehat{\mathbf{r}}_0) = \ \mathrm{span}\Big\{\mathbf{BG}^{\mathrm{T}}\widehat{\mathbf{r}}_0, \ldots, (\mathbf{BA})^{l-1}\mathbf{BG}^{\mathrm{T}}\widehat{\mathbf{r}}_0\Big\}. \tag{31}$$

It can be easily shown that

$$\mathbf{ABG}^{\mathrm{T}} = \mathbf{G}^{\mathrm{T}}\widehat{\mathbf{A}},$$

where $\widehat{\mathbf{A}} = \mathbf{I}_m + \mathbf{R}^{-1}\mathbf{GBG}^{\mathrm{T}}$. Equation (31) can then be written as

$$\mathcal{K}^l(\mathbf{BA},\mathbf{BG}^{\mathrm{T}}\widehat{\mathbf{r}}_0) = \mathrm{span}\Big\{\mathbf{BG}^{\mathrm{T}}\widehat{\mathbf{r}}_0, \ldots, \mathbf{BG}^{\mathrm{T}}\widehat{\mathbf{A}}^{l-1}\widehat{\mathbf{r}}_0\Big\} = \mathbf{BG}^{\mathrm{T}}\mathcal{K}^l(\widehat{\mathbf{A}}, \widehat{\mathbf{r}}_0)$$

where

$$\mathcal{K}^l(\widehat{\mathbf{A}}, \widehat{\mathbf{r}}_0) = \mathrm{span}\Big\{\widehat{\mathbf{r}}_0, \widehat{\mathbf{A}}\widehat{\mathbf{r}}_0, \ldots, \widehat{\mathbf{A}}^{l-1}\widehat{\mathbf{r}}_0\Big\}$$

is the Krylov subspace generated by RBLanczos.

When re-orthogonalization is not required, as with BLanczos, RBLanczos can be transformed into an equivalent algorithm that does not require storing the matrices $\mathbf{T}_l$ and $\widehat{\mathbf{V}}_l$ to find the solution. The algorithm can be derived following the basic approach used to derive RBLanczos Algorithm 2.3 from BLanczos Algorithm 2.1. This is done by using the relations between the vectors defined in $\mathbb{R}^n$ and $\mathbb{R}^m$ given by Eqs (28)-(30) and the additional equations

$$\mathbf{p}_i = \mathbf{BG}^{\mathrm{T}}\widehat{\mathbf{p}}_i, \tag{32}$$

$$\delta\mathbf{u}_i = \mathbf{BG}^{\mathrm{T}}\boldsymbol{\lambda}_i, \tag{33}$$

that relate m-dimensional vectors $\widehat{\mathbf{p}}_i$ and $\boldsymbol{\lambda}_i$ with the vectors $\mathbf{p}_i$ and $\delta\mathbf{u}_i$ in Algorithm 2.2, for $i \geq 0$. It can be shown that relations (32) and (33) are satisfied by choosing

$$\widehat{\mathbf{p}}_i = (\widehat{\mathbf{v}}_i - \beta_i\widehat{\mathbf{p}}_{i-1})/\eta_i,$$
$$\boldsymbol{\lambda}_i = \boldsymbol{\lambda}_{i-1} + \zeta_i\widehat{\mathbf{p}}_i$$

where $\widehat{\mathbf{p}}_0 = \boldsymbol{\lambda}_0 = \mathbf{0}$. Direct substitution of these recurrence relations in BLanczos Algorithm 2.2 yields the memory efficient RBLanczos algorithm without re-orthogonalization given by Algorithm 2.4.

2.5 Re-orthogonalization

The Lanczos vectors in BLanczos (or the residuals in BCG) are, in exact arithmetic, mutually orthogonal with respect to the **B**-inner product. Round-off errors can result in a loss of **B**-orthogonality and can significantly hinder the convergence of these methods as a consequence. It is possible to alleviate this problem by re-orthogonalizing the Lanczos vectors (or the CG residuals) on each iteration using a modified Gram-Schmidt (MGS) procedure (Saad, 1996, p. 11-12).

With reference to Algorithm 2.1 for BLanczos, the re-orthogonalization procedure acts on the vectors $\mathbf{w}_i$. Making use of the orthogonality relationship $\mathbf{v}_i^{\mathrm{T}}\mathbf{B}\mathbf{v}_j = 0$, for $i \neq j$, MGS re-orthogonalization can be described in compact notation by

$$\mathbf{w}_i \leftarrow \left[\prod_{j=i-1}^{1}(\mathbf{I}_n - \mathbf{v}_j\mathbf{v}_j^{\mathrm{T}}\mathbf{B})\right]\mathbf{w}_i, \tag{34}$$

$$= \left[\prod_{j=i-1}^{1}(\mathbf{I}_n - \mathbf{v}_j\mathbf{z}_j^{\mathrm{T}})\right]\mathbf{w}_i. \tag{35}$$

Equations (34) and (35) require the storage of the Lanczos vectors $\mathbf{v}_j$ from all previous iterations $j = 1, \ldots, i-1$. Equation (34) is clearly less practical than Eq. (35) since it requires $i-1$ additional, and generally expensive, applications of **B** to compute the **B**-inner product of $\mathbf{v}_j$ and $\mathbf{w}_i$. This expensive matrix-vector product can be avoided by storing the vectors $\mathbf{z}_j$ and using them in Eq. (35).

The need to store and manipulate the sequence of n-dimensional vectors $\mathbf{v}_j$ and $\mathbf{z}_j$ can lead to significant computational overhead, as will be illustrated in the experiments in section 3. In this respect, the dual algorithms, which involve sequences of m-dimensional vectors, are clearly an attractive alternative when $m \ll n$. From the relationship $\mathbf{v}_i = \mathbf{G}^{\mathrm{T}}\widehat{\mathbf{v}}_i$, the orthogonality condition for the Lanczos vectors can be written as

$$\mathbf{v}_i^{\mathrm{T}}\mathbf{B}\mathbf{v}_j = \widehat{\mathbf{v}}_i^{\mathrm{T}}\mathbf{G}\mathbf{B}\mathbf{G}^{\mathrm{T}}\widehat{\mathbf{v}}_j = 0 \tag{36}$$

for $i \neq j$. Combining Eq. (36) with Eqs (34) and (35) leads to the MGS re-orthogonalization scheme for RBLanczos in Algorithm 2.3:

$$\widehat{\mathbf{w}}_i \leftarrow \left[\prod_{j=i-1}^{1}(\mathbf{I}_m - \widehat{\mathbf{v}}_j\widehat{\mathbf{v}}_j^{\mathrm{T}}\mathbf{G}\mathbf{B}\mathbf{G}^{\mathrm{T}})\right]\widehat{\mathbf{w}}_i,$$

$$= \left[\prod_{j=i-1}^{1}(\mathbf{I}_m - \widehat{\mathbf{v}}_j\widehat{\mathbf{z}}_j^{\mathrm{T}})\right]\widehat{\mathbf{w}}_i.$$

2.6 Monitoring convergence

The values of the quadratic cost function and the norm of the cost function gradient are important for monitoring the convergence of the minimization. In addition to the total value of the cost function (J), the relative contributions to J from the background term ($J_{\rm b}$) and observation term ($J_{\rm o}$) provide additional useful diagnostic information. Inexpensive formulae for computing all these quantities on each iteration from the recurrence relations of the BLanczos and RBLanczos algorithms with re-orthogonalization (Algorithms 2.1 and 2.3) are derived in this section. Similar formulae can also be derived for the corresponding algorithms without re-orthogonalization (Algorithms 2.2 and 2.4), and for BCG and RBCG as described in Appendix D.

2.6.1 Quadratic cost function

Using a Taylor series expansion about the initial guess $\delta\mathbf{u}_0 = \mathbf{0}$, the quadratic cost function $J[\delta\mathbf{u}_i]$ in Eq. (2) can be expressed as

$$J[\delta\mathbf{u}_i] = J[\mathbf{0}] + \delta\mathbf{u}_i^{\rm T}\nabla J[\mathbf{0}] + \frac{1}{2}\delta\mathbf{u}_i^{\rm T}(\nabla^2 J)\,\delta\mathbf{u}_i.$$

Replacing $\nabla^2 J$ by $\mathbf{A}$ (Eq. (9)) and $\nabla J[\mathbf{0}]$ by $-\mathbf{r}_0$ (Eq. (11)), and using the relations (24) and (25), the quadratic cost function can be expressed in terms of quantities from BLanczos as

$$\begin{aligned} J[\delta\mathbf{u}_i] &= J[\mathbf{0}] - \mathbf{s}_i^{\rm T}\mathbf{Z}_i^{\rm T}\mathbf{r}_0 + \frac{1}{2}\mathbf{s}_i^{\rm T}\mathbf{Z}_i^{\rm T}\mathbf{A}\mathbf{Z}_i\mathbf{s}_i, \\ &= J[\mathbf{0}] - \mathbf{s}_i^{\rm T}\mathbf{Z}_i^{\rm T}\mathbf{r}_0 + \frac{1}{2}\mathbf{s}_i^{\rm T}\mathbf{T}_i\mathbf{s}_i, \\ &= J[\mathbf{0}] - \mathbf{s}_i^{\rm T}\mathbf{Z}_i^{\rm T}\mathbf{r}_0 + \frac{1}{2}\mathbf{s}_i^{\rm T}\mathbf{Z}_i^{\rm T}\mathbf{r}_0, \\ &= J[\mathbf{0}] - \frac{1}{2}\mathbf{s}_i^{\rm T}\mathbf{Z}_i^{\rm T}\mathbf{r}_0, \end{aligned} \tag{37}$$

where

$$J[\mathbf{0}] = \frac{1}{2}\mathbf{d}^{\rm T}\mathbf{R}^{-1}\mathbf{d}. \tag{38}$$

Using again Eqs (24) and (25), the $J_{\rm b}$ term can be evaluated as

$$\begin{aligned} J_{\rm b}[\delta\mathbf{u}_i] &= \frac{1}{2}\delta\mathbf{u}_i^{\rm T}\mathbf{B}^{-1}\delta\mathbf{u}_i, \\ &= \frac{1}{2}\mathbf{s}_i^{\rm T}\mathbf{Z}_i^{\rm T}\mathbf{B}^{-1}\mathbf{Z}_i\mathbf{s}_i, \\ &= \frac{1}{2}\mathbf{s}_i^{\rm T}\mathbf{V}_i^{\rm T}\mathbf{Z}_i\mathbf{s}_i. \end{aligned} \tag{39}$$

The J_o term can be computed simply from the difference between J and J_b:

$$J_\mathrm{o}[\delta\mathbf{u}_i] = J[\delta\mathbf{u}_i] - J_\mathrm{b}[\delta\mathbf{u}_i]. \tag{40}$$

The expressions for J and J_b also depend on the vectors $\mathbf{s}_i$. These vectors can be computed on each iteration by solving the (inexpensive for small i) tridiagonal system $\mathbf{T}_i\mathbf{s}_i = \beta_0\mathbf{e}_1$.

The values of J and J_b can be evaluated in terms of quantities from RBLanczos by using the relation (28) and the relations for $\mathbf{v}_i$ and $\mathbf{z}_i$ in Eqs (29) and (30) in the expressions (37) and (39). This yields

$$\begin{aligned} J[\delta\mathbf{u}_i] &= J[\mathbf{0}] - \frac{1}{2}\mathbf{s}_i^\mathrm{T}\mathbf{Z}_i^\mathrm{T}\mathbf{G}^\mathrm{T}\widehat{\mathbf{r}}_0, \\ &= J[\mathbf{0}] - \frac{1}{2}\mathbf{s}_i^\mathrm{T}\widehat{\mathbf{Z}}_i^\mathrm{T}\widehat{\mathbf{r}}_0, \end{aligned}$$

and

$$\begin{aligned} J_\mathrm{b}[\delta\mathbf{u}_i] &= \frac{1}{2}\mathbf{s}_i^\mathrm{T}\widehat{\mathbf{V}}_i^\mathrm{T}\mathbf{G}\mathbf{Z}_i\mathbf{s}_i, \\ &= \frac{1}{2}\mathbf{s}_i^\mathrm{T}\widehat{\mathbf{V}}_i^\mathrm{T}\widehat{\mathbf{Z}}_i\mathbf{s}_i, \end{aligned}$$

where $J[\mathbf{0}]$ is again given by Eq. (38). The J_o term can then be evaluated from Eq. (40). The matrices $\widehat{\mathbf{V}}_i$ and $\widehat{\mathbf{Z}}_i$ contain the $j = 1, \ldots, i$ column vectors $\widehat{\mathbf{v}}_j$ and $\widehat{\mathbf{z}}_j$. These matrices are also needed for re-orthogonalization.

2.6.2 Norm of the quadratic cost function gradient

The residual vector of the approximate solution $\delta\mathbf{u}_i$ computed by BLanczos satisfies (Saad, 1996, p.153)

$$\mathbf{r}_i = \mathbf{b} - \mathbf{A}\delta\mathbf{u}_i = -\beta_{i+1}\mathbf{e}_i^\mathrm{T}\mathbf{s}_i\mathbf{v}_{i+1},$$

where $\mathbf{e}_i$ is the i-th column of the $i \times i$ identity matrix, and β_{i+1}, $\mathbf{s}_i$ and $\mathbf{v}_i$ are as defined in Algorithm 2.1. Since $\mathbf{r}_i = -\nabla J[\delta\mathbf{u}_i]$, the $\mathbf{B}$-norm of the cost function gradient can be readily computed as

$$\|\nabla J[\delta\mathbf{u}_i]\|_\mathbf{B} = \|\mathbf{r}_i\|_\mathbf{B} = \beta_{i+1}|\mathbf{e}_i^\mathrm{T}\mathbf{s}_i|.$$

The same formula can be used to compute the gradient norm with RBLanczos.

3 Numerical Experiments

3.1 Experimental framework

This section provides a brief description of the NEMOVAR and ROMS data assimilation systems and the framework in which the experiments are performed.

3.1.1 NEMOVAR

NEMOVAR is a variational data assimilation system (Mogensen *et al.*, 2009) for the NEMO (Nucleus for European Modelling of the Ocean) ocean model (Madec, 2008). It is designed as an incremental 4D-Var algorithm. Three-dimensional variational assimilation (3D-Var) is also supported, using the First-Guess at Appropriate Time (FGAT) approach. 3D-Var FGAT follows the basic formulation of incremental 4D-Var but replaces the linearized model propagator with the identity operator ($\mathbf{M}(t_j, t_0) \equiv \mathbf{I}_n$). Dynamical information is still incorporated into the 3D-Var analysis through a balance operator, which relates the control vector to the initial state vector (see later).

A close variant of the operational version of NEMOVAR used at ECMWF for seasonal forecasting and ocean reanalysis (Mogensen *et al.*, 2012) is used for the experiments presented here. The system is based on 3D-Var FGAT. The configuration is global, with 42 vertical levels and approximately 1° resolution in the extratropics. In the tropics, the meridional resolution is refined, reaching a minimum value of 0.3° directly at the equator.

Both BCG and RBCG have been implemented in NEMOVAR, following very similar technical specifications. NEMOVAR also includes a version of Lanczos (the Lanczos method on the linear system (19) with the canonical inner product). Here the experiments focus on comparing BCG and RBCG, and will be presented in section 3.2. These experiments are conducted on a single 3D-Var analysis cycle where the time period covered by the cycle is 10 days. The background initial conditions are taken from the ECMWF ORAS4 ocean reanalysis (Balmaseda *et al.*, 2012) on 01 January, 2006. The model state variables comprise potential temperature (T), salinity (S), sea-surface height (η) and the horizontal components (u, v) of velocity. The background-state trajectory on the 10-day window is generated by integrating forward the NEMO model using the same daily surface forcing fluxes and relaxation strategies to sea-surface temperature and climatology as used in ORAS4 for the same period. The observations that are assimilated on this cycle are the same as those used in ORAS4 on the same cycle. They consist of temperature and salinity (T/S) profiles from the quality-controlled EN3_v2a data-set[2] and along-track sea-level anomaly (SLA) data from altimeter database at AVISO[3]. The SLA data are referenced to a mean dynamic topography defined as the time-mean sea-level from an ocean reanalysis similar to ORAS4 but assimilating only T/S profiles. The total number of individual observations assimilated on the 10-day 3D-Var cycle from 01 January – 11 January, 2006 is approximately 5.0×10^5, with 2.1×10^5 for T, 1.6×10^5 for S, and 1.3×10^5 for SLA.

The variables in the control vector $\delta\mathbf{u}$ consist of increments δT, δS_{U} and $\delta\eta_{\mathrm{U}}$ where the subscript U corresponds to an "unbalanced" component of that variable (Weaver *et al.*, 2005). The control variables are assumed to be mutually uncorrelated. A linearized balance operator $\mathbf{K}$ relates $\delta\mathbf{u}$ to the increments δT, δS, $\delta\eta$, δu and δv comprising the initial state $\delta\mathbf{x}(t_0)$. The balance operator acts as a multivariate constraint in the linearized

[2] http://www.metoffice.ogv.uk/hadobs/en3.

[3] http://www.aviso.oceanobs.com/en/data/products/sea-surface-height-products/global/sla/index.html.

generalized observation operator $\mathbf{G}$ in the observation term of the cost function (3). In this experiment, the horizontal velocity increments δu and δv are determined entirely by geostrophic constraints in $\mathbf{K}$ (Mogensen *et al.*, 2012). The total number of *active* control variables is approximately 4.6×10^6.

For computational convenience, a land-ocean mask with values of 0 (land) and 1 (ocean) is used in NEMO, as in many ocean models, to account for the complex geometry of land-ocean boundaries. As a consequence, (passive) land points are included along with (active) ocean points in the model-variable arrays in the computer code. By including the land points, the size of the control vector is roughly 9.2×10^6. Half of the memory allocation in a control vector is thus associated with land points. A control vector in this experiment requires 19 times more memory than an observation vector.

3.1.2 ROMS 4D-Var

ROMS is a hydrostatic, primitive equation, Boussinesq ocean general circulation model designed primarily for coastal applications. Terrain-following vertical coordinates are employed allowing for greater vertical resolution in shallow water and regions with complex bathymetry (Marchesiello *et al.*, 2001; Shchepetkin and McWilliams, 2003, 2005; Haidvogel *et al.*, 2008). ROMS supports both a primal and dual formulation of incremental 4D-Var. The primal form follows very closely that of Weaver *et al.* (2003) and solves Eq. (19) using the standard Lanczos algorithm which is equivalent to solving Eq. (5) using the BLanczos algorithm.

Two forms of dual 4D-Var are available in ROMS, which differ in the choice of linearization strategy on the outer loop (Moore *et al.*, 2011a,b). One follows the TGN approach, while the other is based on the indirect representer approach of Egbert *et al.* (1994). For the current study, there is no distinction between these approaches since the experiments are conducted with a single outer loop where the linearization trajectory is the background.

As noted earlier, one of the primary advantages of the dual approach to 4D-Var is in the weak-constraint formulation. Since Eq. (27) is solved in a space spanned by vectors whose size is determined by the number of observations, the dimension of the problem is not changed by the addition of control variables to account for model error. Conversely, in the primal form embodied by Eq. (5), the dimension of the control vector swells very quickly as additional control variables are added.

Four options are available in ROMS for solving the quadratic problem (3), namely Lanczos (the Lanczos method on the linear system (19) with the canonical inner product), dual-Lanczos (the Lanczos method on the linear system (27) with the canonical inner product), dual-MINRES (MINRES on the linear system (27) with the canonical inner product), and RBLanczos. A comparison of the four approaches under the assumption of both strong and weak constraint will be presented in section 3.2.

The configuration of ROMS used for the experiments presented here is for the California Current System (CCS), an eastern boundary current characterized by a pronounced seasonal cycle of upwelling and by energetic mesoscale circulations (Hickey, 1998; Checkley and Barth, 2009), providing a challenge for linear-based data assimilation methods such as incremental 4D-Var. The ROMS CCS domain and circulation is described by Veneziani *et al.* (2009a,b), Broquet *et al.* (2009a) and Broquet *et al.* (2009b). It spans the region 134°W to 116°W and 31°N to 48°N, with 10 km resolution in the horizontal and 42 σ-levels. The minimization experiments described here are conducted for a single 4D-Var analysis cycle spanning the 7-day period 29 March – 4 April, 2003. The background initial conditions are taken from a 4D-Var sequence described in Moore *et al.* (2011b), and the model control variables are the same as those of NEMOVAR, with the addition of unbalanced variables $\delta\mathbf{u}_\mathrm{U}$ and $\delta\mathbf{v}_\mathrm{U}$ for horizontal velocity. The background trajectory is generated using forcing derived from daily averaged output of atmospheric boundary layer fields from the Naval Research Laboratory's Coupled Ocean-Atmosphere Mesoscale Prediction System (COAMPS) (Doyle *et al.*, 2009). The ocean surface fluxes were derived using the bulk formulations of Liu *et al.* (1979), Fairall *et al.* (1996a) and Fairall *et al.* (1996b), and represent the background surface forcing for 4D-Var. The model domain has open boundaries at the northern, southern, and western edges, and at these boundaries the tracer and velocity fields were prescribed, while the free surface and vertically integrated flow are subject to Chapman (1985) and Flather (1976) boundary conditions respectively. The prescribed open boundary solution was taken from the Estimating the Circulation and Climate of the Ocean (ECCO) global data assimilation product (Wunsch and Heimbach, 2007), and represents the background open boundary conditions for 4D-Var.

In the strong-constraint 4D-Var experiments, the control vector comprises increments to the model initial conditions, the ocean surface forcing (namely the surface wind stress, and the surface fluxes of heat and freshwater), and open boundary conditions. The number of control variables is 7.3×10^6. In the weak-constraint 4D-Var experiments, the control vector is augmented to include corrections for model error as well, which take the form of space-time corrections to the model equations of motion. The size of the control vector in this case is 5.2×10^7. The implementation of the ROMS weak-constraint 4D-Var used here is the same as that described in Moore *et al.* (2011b).

The observations assimilated are the same as those described in Moore *et al.* (2011b) and include: gridded sea surface height analyses in the form of dynamic topography from AVISO at approximately 1/3° resolution every 7 days; a blended SST product with 10 km resolution, available daily from NOAA CoastWatch, and consisting of 5-day means derived from the GEOS, AVHRR and MODIS satellite instruments; and *in situ* hydrographic observations extracted from the quality controlled EN3_v1d data set. The total number of observations is of the order 10^5 (1.0×10^5 for T, 5.0×10^2 for S, and 2.4×10^3 for SSH).

3.2 Results

This section describes the results from numerical experiments with NEMOVAR and ROMS that are designed to compare the various primal and dual algorithms discussed in this paper. The distinguishing characteristics of these algorithms are summarized in Table 1. The different experiments are summarized in Table 2.

3.2.1 NEMOVAR

The experiments with NEMOVAR are based on 3D-Var FGAT with a single outer-loop iteration and 40 inner-loop iterations. The properties of the minimization are summarized in Figure 1. The upper panel of this figure shows the iterative evolution of the quadratic cost function (3) using the primal formulation (BCG) and dual formulation (RBCG) of the CG algorithm. The curves are indistinguishable as expected from the theory. This provides further evidence of the practical equivalence of these algorithms, in this case for a minimization problem involving hundreds of thousands of variables and observations. Numerical results with Lanczos (not shown) also produce indistinguishable results from those of BCG and RBCG.

The results also illustrate that, for this 3D-Var experiment, there is no visible improvement in the cost function minimization when using re-orthogonalization. However, a positive impact from re-orthogonalization is seen after iteration 22 in the evolution of the gradient norm (lower panel). With re-orthogonalization, the gradient norms in the BCG_O and RBCG_O experiments are indistinguishable (when the norm is defined with respect to the appropriate metric: $\mathbf{B}$ in BCG, and $\mathbf{GBG}^{\mathrm{T}}$ in RBCG).

While Figure 1 shows that the algorithms for the primal and dual formulations have near identical convergence properties, Figure 2 illustrates that the memory requirements of the algorithms can be very different, particularly with re-orthogonalization. This figure shows the memory usage as a function of execution time, where the initial point has been chosen just before entering the minimization loop. The jump in the curves near the start corresponds to static memory allocation within the CG algorithms (∼2.5 Gb for BCG; ∼2 Gb for RBCG). Thereafter, the total memory usage remains constant if re-orthogonalization is not activated, resulting in a saving of ∼0.5 Gb with RBCG. Re-orthogonalization requires storing an extra pair of vectors on each iteration as discussed in section 2.5. The required memory is allocated dynamically in NEMOVAR, which explains the steady increase of the curves BCG_O and RBCG_O. Re-orthogonalization produces only small memory overhead with RBCG, with the total memory being comparable to that required by BCG *without* re-orthogonalization. On the other hand, re-orthogonalization with BCG requires a significantly larger amount of extra memory (∼6 Gb). With higher resolution model configurations or with applications, such as 4D-Var, that involve additional control variables, the memory requirements could easily become prohibitive. This figure also illustrates that the cost of algebraic computations, such as scalar products, is noticeably cheaper with the dual approach than the primal approach. The total execution time with RBCG is roughly equivalent with or without re-orthogonalization, and is noticeably shorter than with BCG. This is especially noticeable when re-orthogonalization is activated where the computational saving with RBCG is about 22%.

3.2.2 ROMS 4D-Var

In all of the experiments presented here, ROMS 4D-Var was run, as was with NEMO 3D-Var, with a single outer-loop iteration and 40 inner-loop iterations. Figures 3 and 4 summarize the results of various experiments employing either strong- or weak-constraint 4D-Var. These figures show the behaviour of the quadratic cost function (3) as a function of the number of inner-loop iterations.

The results for the strong-constraint 4D-Var are shown in the upper panel of Figure 3. The curves correspond to experiments with the primal solver BLanczos and the dual solvers RBLanczos, dual-MINRES and dual-Lanczos. As anticipated, RBLanczos and BLanczos yield the same sequence of cost function values. Clearly the performance of RBLanczos is superior to both dual-MINRES and dual-Lanczos. The upper panel of Figure 3 illustrates the poor convergence properties of dual-Lanczos, in agreement with results from GT09 and El Akkraoui and Gauthier (2010) who employed CG to the linear system (27) with a canonical inner product (which is mathematically equivalent to applying dual-Lanczos to system (27)). In fact, approximately 100 inner-loop iterations are generally required to achieve an acceptable level of convergence with dual-Lanczos, as shown by Moore *et al.* (2011b). The results of three corresponding weak-constraint dual 4D-Var calculations are shown in the lower panel of Figure 3, revealing again the superior convergence properties of RBLanczos. The convergence of weak-constraint 4D-Var is typically slower than that of strong-constraint 4D-Var, and in this case, the dual-Lanczos algorithm requires more than 200 inner-loop iterations to yield an acceptable level of convergence (Moore *et al.*, 2011b).

The importance of re-orthogonalization with the dual approach is illustrated in Figure 4 which shows, for the case of weak-constraint 4D-Var, the results from applying dual-MINRES and RBLanczos without re-orthogonalization (cf. Figure 3). After an initial period of convergence, the cost function diverges. This is in stark contrast to the NEMO 3D-Var experiment where re-orthogonalization had a relatively minor impact on convergence. The need for re-orthogonalization in weak-constraint 4D-Var places an enormous memory burden on the primal approach. Indeed, in the ROMS weak-constraint 4D-Var experiments, only the dual approach was practically feasible. Unlike NEMO 3D-Var, the computational cost of ROMS 4D-Var, with strong- or weak-constraints, is comparable for both the primal and dual formulations for the same number of inner-loop iterations since the majority of the computational burden comes from the integration of the TL and adjoint models.

4 Summary and conclusions

Variational data assimilation involves the solution a nonlinear least-squares problem to estimate an n-dimensional control vector given an m-dimensional observation vector and an n-dimensional background estimate of the control vector. A nonlinear generalized observation operator relates the model control vector to the observation vector. Variational data assimilation problems in meteorology and oceanography are commonly solved using an incremental or Truncated Gauss-Newton approach. This approach requires minimizing a sequence of quadratic cost functions, where each member of the sequence is derived by linearizing the generalized observation operator about a recent estimate.

The quadratic minimization problems can be solved using a primal or dual approach. Primal approaches search for a solution directly in an n-dimensional space associated with the model control variables. Dual approaches exploit the fact that the solution of the quadratic minimization problem is in the range of the adjoint of the generalized observation operator to search for the solution in an m-dimensional space associated with the observations. The dual approach becomes especially attractive from a computational viewpoint when $m \ll n$ which is a typical situation in ocean data assimilation and with weak-constraint formulations of 4D-Var in general.

CG and Lanczos methods, which belong to the general class of Krylov subspace iterative methods, are well suited for solving quadratic minimization problems when n and m are large. Conventional implementations of incremental 4D-Var employ a primal approach in conjunction with a CG or Lanczos algorithm. The system is preconditioned by the background-error covariance matrix ($\mathbf{B}$) which in practice is achieved through a variable transformation that requires a square-root factorization of $\mathbf{B}$. $\mathbf{B}$-preconditioning of the CG and Lanczos algorithms can also be achieved without the need to factor $\mathbf{B}$. This property, which is convenient with general $\mathbf{B}$ formulations, is shared by the BCG and BLanczos primal algorithms that were presented in this paper. Both algorithms are equivalent in exact arithmetic.

The dual equivalent of BCG is the RPCG algorithm of GT09 with no second-level preconditioning. We called this specific algorithm RBCG to distinguish it from RPCG which allows for general preconditioners. Numerical experiments comparing the performance of RBCG and BCG in a global-ocean 3D-Var assimilation system for the NEMO model were presented. In the experiments, the control vector was roughly 19 times larger than the observation vector. As expected from theory, and as demonstrated in an idealized framework by GT09, RBCG and BCG produce identical iterates to within machine precision. The memory requirements and computational costs of the algorithms were very different, however, particularly when a re-orthogonalization scheme was used to compensate for the effects of round-off error. While RBCG was relatively insensitive to re-orthogonalization in terms of memory overhead, BCG with re-orthogonalization required nearly twice as much total memory as BCG without re-orthogonalization. RBCG was also nearly 22% faster than BCG.

The dual equivalent of BLanczos is RBLanczos which is a new algorithm that was derived in this paper. The derivation of RBLanczos follows directly from BLanczos by exploiting the fundamental property that, under the assumption of a zero initial guess, the initial value of the cost function gradient is in the range of the adjoint of the generalized observation operator. As a result, all recurrence relations involving n-dimensional vectors in BLanczos can be transformed into recurrence relations involving m-dimensional vectors.

The resulting algorithm requires the same number of matrix-vector products as BLanczos.

Numerical experiments comparing the performance of RBLanczos and Lanczos (the equivalent form of BLanczos that employs the factorized form of **B** as a preconditioner) in a regional-ocean 4D-Var assimilation system for the ROMS model were presented. As expected from theory, RBLanczos and Lanczos produce identical iterates to within machine precision. In experiments that employed both strong- and weak-constraint formulations of 4D-Var, the convergence properties of RBLanczos were also clearly superior to those of two other dual algorithms (dual-Lanczos and dual-MINRES) that have been proposed in the literature. In all algorithms, re-orthogonalization was found to be crucial for convergence and produced much smaller memory overhead with the dual algorithms than with the primal algorithms. It is also important to remark that only the dual approach was practically feasible in the weak-constraint 4D-Var experiment due to the very large size of the control vector.

The two ocean models and data assimilation systems used here differ significantly. On the one hand, NEMO is configured as a coarse resolution model of the global ocean and captures the dominant large-scale features of the global circulation. ROMS, on the other hand, is configured as an eddy-resolving model of the west coast of North America, and captures the energetic meso-scale circulation environment associated with the CCS. In addition, in this study, NEMO employs 3D-Var while ROMS employs 4D-Var with the model applied as either a strong or weak constraint. It is therefore noteworthy that RBCG and RBLanczos are robust in two very different circulation environments, and across a range of data assimilation approaches within the variational framework.

This work has illustrated the benefits of RBCG and RBLanczos from different perspectives. In this study, we considered only a single outer loop of the incremental algorithm and only first-level preconditioning using **B**. Extensions of this work to account for multiple outer-loop iterations and second-level preconditioners based on Limited-Memory Preconditioners such as Quasi-Newton and Ritz (Gratton *et al.*, 2011b) will be described in a future publication. In view of the importance of re-orthogonalization in the 4D-Var experiments, it would also be of interest to investigate the use of FOM, which is more stable with respect to loss of orthogonality since it includes naturally an explicit full re-orthogonalization of the Krylov subspace basis.

Acknowledgements

This work is a contribution to the ADTAO and FILAOS projects which are financed by the RTRA STAE foundation. Additional support from the French National Research Agency (ANR) COSINUS programme (VODA project, n° ANR-08-COSI-016) and the French LEFE programme is also acknowledged. The authors also gratefully acknowledge the support of the U.S. Office of Naval Research (N00014-10-1-0476, N00014-10-1-0322) and the National Science Foundation (OCE-1061434). Any opinions, findings, and conclusions or recommendations expressed here are those of the authors and do not necessarily reflect

the views of the National Science Foundation. Kristian Mogensen is thanked for his help in implementing the minimization algorithms in NEMOVAR.

Appendix A: Relationships between vectors in RBLanczos and BLanczos

Following GT09 and Gratton *et al.* (2009) in their derivation of RPCG, we suppose that

$$\mathbf{r}_0 \in \text{range}(\mathbf{G}^{\mathrm{T}})$$

which is satisfied under the assumption that the initial guess $\delta\mathbf{u}_0 = \mathbf{0}$. In particular,

$$\mathbf{r}_0 = \mathbf{G}^{\mathrm{T}}\widehat{\mathbf{r}}_0 \tag{41}$$

where $\widehat{\mathbf{r}}_0 = \mathbf{R}^{-1}\mathbf{d} \in \mathbb{R}^m$.

Let us define the vectors in $\mathbb{R}^m$ such that

$$\begin{aligned}
\widehat{\mathbf{v}}_0 &= \mathbf{0}, \\
\widehat{\mathbf{t}}_0 &= \mathbf{G}\mathbf{B}\mathbf{G}^{\mathrm{T}}\widehat{\mathbf{r}}_0, && (42)\\
\widehat{\mathbf{v}}_1 &= \widehat{\mathbf{r}}_0/\beta_0, && (43)\\
\widehat{\mathbf{z}}_1 &= \widehat{\mathbf{t}}_0/\beta_0, && (44)
\end{aligned}$$

where $\beta_0 = \sqrt{\widehat{\mathbf{r}}_0^T\,\widehat{\mathbf{t}}_0}$. From the definitions (41)-(44), it is possible to derive the following relations from BLanczos:

$$\begin{aligned}
\mathbf{G}\mathbf{t}_0 &= \widehat{\mathbf{t}}_0, \\
\mathbf{v}_1 &= \mathbf{G}^{\mathrm{T}}\widehat{\mathbf{v}}_1, && (45)\\
\mathbf{G}\mathbf{z}_1 &= \widehat{\mathbf{z}}_1, && (46)\\
\mathbf{q}_1 &= \mathbf{v}_1 + \mathbf{G}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{G}\mathbf{z}_1 \\
&= \mathbf{G}^{\mathrm{T}}\left(\widehat{\mathbf{v}}_1 + \mathbf{R}^{-1}\widehat{\mathbf{z}}_1\right) && (47)\\
\mathbf{w}_1 &= \mathbf{q}_1 - \alpha_1\mathbf{v}_1 \\
&= \mathbf{G}^{\mathrm{T}}\left[(\widehat{\mathbf{v}}_1 + \mathbf{R}^{-1}\widehat{\mathbf{z}}_1) - \alpha_1\widehat{\mathbf{v}}_1\right] && (48)\\
\mathbf{t}_1 &= \mathbf{B}\mathbf{w}_1, \\
&= \mathbf{B}\mathbf{G}^{\mathrm{T}}\left[(\widehat{\mathbf{v}}_1 + \mathbf{R}^{-1}\widehat{\mathbf{z}}_1) - \alpha_1\widehat{\mathbf{v}}_1\right]. && (49)
\end{aligned}$$

Equations (47)-(49) can be written as

$$\begin{aligned}
\mathbf{q}_1 &= \mathbf{G}^{\mathrm{T}}\widehat{\mathbf{q}}_1, && (50)\\
\mathbf{w}_1 &= \mathbf{G}^{\mathrm{T}}\widehat{\mathbf{w}}_1, && (51)\\
\mathbf{G}\mathbf{t}_1 &= \widehat{\mathbf{t}}_1, && (52)
\end{aligned}$$

where $\widehat{\mathbf{q}}_1 = \widehat{\mathbf{v}}_1 + \mathbf{R}^{-1}\widehat{\mathbf{z}}_1$, $\widehat{\mathbf{w}}_1 = \widehat{\mathbf{q}}_1 - \alpha_1\widehat{\mathbf{v}}_1$, $\alpha_1 = \widehat{\mathbf{q}}_1^{\mathrm{T}}\widehat{\mathbf{z}}_1$ and $\widehat{\mathbf{t}}_1 = \mathbf{G}\mathbf{B}\mathbf{G}^{\mathrm{T}}\widehat{\mathbf{w}}_1$.

We now prove by induction that the relations given by Eqs (45)-(46) and Eqs (50)-(52) also hold for $i > 1$. First, assume that the relations are satisfied for a given i:

$$\begin{aligned}
\mathbf{v}_i &= \mathbf{G}^{\mathrm{T}}\widehat{\mathbf{v}}_i, && (53)\\
\mathbf{G}\mathbf{z}_i &= \widehat{\mathbf{z}}_i, &&\\
\mathbf{q}_i &= \mathbf{G}^{\mathrm{T}}\widehat{\mathbf{q}}_i, &&\\
\mathbf{w}_i &= \mathbf{G}^{\mathrm{T}}\widehat{\mathbf{w}}_i, && (54)\\
\mathbf{G}\mathbf{t}_i &= \widehat{\mathbf{t}}_i. && (55)
\end{aligned}$$

We now show that the relations hold for $i+1$. From lines 18 and 19 of Algorithm 2.1 for BLanczos, we have, using Eqs (54) and (55),

$$\begin{aligned}
\mathbf{v}_{i+1} &= \mathbf{w}_i/\beta_{i+1} = \mathbf{G}^{\mathrm{T}}\widehat{\mathbf{w}}_i/\beta_{i+1}, && (56)\\
\mathbf{G}\mathbf{z}_{i+1} &= \mathbf{G}\mathbf{t}_i/\beta_{i+1} = \widehat{\mathbf{t}}_i/\beta_{i+1}. && (57)
\end{aligned}$$

Defining

$$\begin{aligned}
\widehat{\mathbf{v}}_{i+1} &= \widehat{\mathbf{w}}_i/\beta_{i+1},\\
\widehat{\mathbf{z}}_{i+1} &= \widehat{\mathbf{t}}_i/\beta_{i+1},
\end{aligned}$$

we obtain from Eqs (56) and (57) that

$$\begin{aligned}
\mathbf{v}_{i+1} &= \mathbf{G}^{\mathrm{T}}\widehat{\mathbf{v}}_{i+1}, && (58)\\
\mathbf{G}\mathbf{z}_{i+1} &= \widehat{\mathbf{z}}_{i+1}. && (59)
\end{aligned}$$

From line 11 of Algorithm 2.1 and using the relations (53), (58) and (59), we have

$$\begin{aligned}
\mathbf{q}_{i+1} &= (\mathbf{v}_{i+1} + \mathbf{G}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{G}\mathbf{z}_{i+1}) - \beta_{i+1}\mathbf{v}_i\\
&= (\mathbf{G}^{\mathrm{T}}\widehat{\mathbf{v}}_{i+1} + \mathbf{G}^{\mathrm{T}}\mathbf{R}^{-1}\widehat{\mathbf{z}}_{i+1}) - \beta_{i+1}\mathbf{G}^{\mathrm{T}}\widehat{\mathbf{v}}_i\\
&= \mathbf{G}^{\mathrm{T}}\left[(\widehat{\mathbf{v}}_{i+1} + \mathbf{R}^{-1}\widehat{\mathbf{z}}_{i+1}) - \beta_{i+1}\widehat{\mathbf{v}}_i\right]\\
&= \mathbf{G}^{\mathrm{T}}\widehat{\mathbf{q}}_{i+1}, && (60)
\end{aligned}$$

where

$$\widehat{\mathbf{q}}_{i+1} = (\widehat{\mathbf{v}}_{i+1} + \mathbf{R}^{-1}\widehat{\mathbf{z}}_{i+1}) - \beta_{i+1}\widehat{\mathbf{v}}_i.$$

Similarly, from line 13 of Algorithm 2.1 and using the relations (58) and (60), we have

$$\begin{aligned}
\mathbf{w}_{i+1} &= \mathbf{q}_{i+1} - \alpha_{i+1}\mathbf{v}_{i+1}\\
&= \mathbf{G}^{\mathrm{T}}\widehat{\mathbf{q}}_{i+1} - \alpha_{i+1}\mathbf{G}^{\mathrm{T}}\widehat{\mathbf{v}}_{i+1}\\
&= \mathbf{G}^{\mathrm{T}}(\widehat{\mathbf{q}}_{i+1} - \alpha_{i+1}\widehat{\mathbf{v}}_{i+1})\\
&= \mathbf{G}^{\mathrm{T}}\widehat{\mathbf{w}}_{i+1},
\end{aligned}$$

where

$$\widehat{\mathbf{w}}_{i+1} = \widehat{\mathbf{q}}_{i+1} - \alpha_{i+1}\widehat{\mathbf{v}}_{i+1},$$

which completes the proof.

Appendix B: BCG

The BCG algorithm is given in Algorithm 4.1. All vectors are in $\mathbb{R}^n$. Each loop of the algorithm requires one matrix-vector multiplication with $\mathbf{G}$, $\mathbf{G}^{\mathrm{T}}$, $\mathbf{R}^{-1}$ and $\mathbf{B}$.

From the theoretical equivalence of BLanczos and BCG, it is possible to construct the tridiagonal matrix $\mathbf{T}_l$ from the coefficients α_i and β_i available in BCG (Saad, 1996, p. 181-182), and hence to obtain approximate eigenvalues of the Hessian matrix from the eigenvalues $\mathbf{T}_l$. In particular, the diagonal and off-diagonal entries of $\mathbf{T}_l$ can be obtained from the relations

$$(\mathbf{T}_l)_{i,i} = \begin{cases} \dfrac{1}{\alpha_{i-1}} & \text{if } i = 1, \\ \dfrac{1}{\alpha_{i-1}} + \dfrac{\beta_{i-2}}{\alpha_{i-2}} & \text{if } i > 1, \end{cases} \tag{61}$$

and

$$(\mathbf{T}_l)_{i+1,i} = (\mathbf{T}_l)_{i,i+1} = \frac{\sqrt{\beta_{i-1}}}{\alpha_{i-1}}. \tag{62}$$

Appendix C: RBCG

The RBCG algorithm is given in Algorithm 4.2. Apart from the solution $\delta\mathbf{u}_l$, all vectors are in $\mathbb{R}^m$. As with BCG, each loop of the RBCG algorithm requires one matrix-vector multiplication with $\mathbf{G}$, $\mathbf{G}^{\mathrm{T}}$, $\mathbf{R}^{-1}$ and $\mathbf{B}$. The tridiagonal matrix $\mathbf{T}_l$ can be generated using Eqs (61) and (62) since the coefficients α_i and β_i are equivalent in BCG and RBCG.

Appendix D: Monitoring convergence with BCG and RBCG

Quadratic cost function

For BCG, the quadratic cost function can be calculated from Eq. (37) using the relation (24):

$$J[\delta\mathbf{u}_i] = J[\mathbf{0}] - \frac{1}{2}\delta\mathbf{u}_i^{\mathrm{T}}\mathbf{r}_0, \tag{63}$$

where $J[\mathbf{0}]$ is given by Eq. (38).

The J_{b} term can be calculated as

$$\begin{aligned} J_{\mathrm{b}}[\delta\mathbf{u}_i] &= \frac{1}{2}\delta\mathbf{u}_i^{\mathrm{T}}\mathbf{B}^{-1}\delta\mathbf{u}_i \\ &= \frac{1}{2}\delta\mathbf{u}_i^{\mathrm{T}}\mathbf{f}_i \end{aligned}$$

where $\mathbf{f}_i = \mathbf{B}^{-1}\delta\mathbf{u}_i$ can be computed without the need to apply $\mathbf{B}^{-1}$ by including an additional recurrence relation in Algorithm 4.1. The J_{o} term can then be computed from Eq. (40).

For RBCG, the quadratic cost function is calculated from Eq. (63) using the relations (8) and (28):

$$\begin{aligned} J[\delta\mathbf{u}_i] &= J[\mathbf{0}] - \frac{1}{2}\boldsymbol{\lambda}_i^{\mathrm{T}}\mathbf{G}\mathbf{B}\mathbf{G}^{\mathrm{T}}\widehat{\mathbf{r}}_0 \\ &= J[\mathbf{0}] - \frac{1}{2}\boldsymbol{\lambda}_i^{\mathrm{T}}\mathbf{w}_0, \end{aligned}$$

where $\mathbf{w}_0 = \mathbf{G}\mathbf{B}\mathbf{G}^{\mathrm{T}}\widehat{\mathbf{r}}_0$ is available from Algorithm 4.2. Using Eq. (8), the J_{b} term can be written as

$$\begin{aligned} J_{\mathrm{b}}[\delta\mathbf{u}_i] &= \boldsymbol{\lambda}_i^{\mathrm{T}}\mathbf{G}\mathbf{B}\mathbf{G}^{\mathrm{T}}\boldsymbol{\lambda}_i \\ &= \boldsymbol{\lambda}_i^{\mathrm{T}}\mathbf{c}_i, \end{aligned}$$

where $\mathbf{c}_i = \mathbf{G}\mathbf{B}\mathbf{G}^{\mathrm{T}}\boldsymbol{\lambda}_i$ can be diagnosed at little extra cost by including an additional recurrence relation in Algorithm 4.2. The J_{o} term can then be computed from Eq. (40).

Norm of the quadratic cost function gradient

For BCG, the gradient norm is given by

$$\|\nabla J[\delta\mathbf{u}_i]\|_{\mathbf{B}} = \|\mathbf{r}_i\|_{\mathbf{B}} = \sqrt{\mathbf{r}_i^{\mathrm{T}}\mathbf{z}_i}$$

where the vectors $\mathbf{z}_i$ and $\mathbf{r}_i$ are defined in Algorithm 4.1. For RBCG, the norm is obtained from

$$\|\nabla J[\delta\mathbf{u}_i]\|_{\mathbf{B}} = \|\mathbf{G}^{\mathrm{T}}\widehat{\mathbf{r}}_i\|_{\mathbf{B}} = \|\widehat{\mathbf{r}}_i\|_{\mathbf{G}\mathbf{B}\mathbf{G}^{\mathrm{T}}} = \sqrt{\widehat{\mathbf{r}}_i^{\mathrm{T}}\mathbf{w}_i}$$

where the vectors $\widehat{\mathbf{r}}_i$ and $\mathbf{w}_i$ are defined in Algorithm 4.2.

References

Axelsson O. 1996. *Iterative Solution Methods.* Cambridge University Press: Cambridge, England.

Balmaseda MA, Mogensen K, Weaver AT. 2012. Evaluation of the ECMWF ocean reanalysis ORAS4. *Q. J. R. Meteorol. Soc.* In press.

Bennett AF. 2002. *Inverse Modelling of the Ocean and Atmosphere.* Cambridge University Press: Cambridge.

Broquet G, Edwards CA, Moore AM, Powell BS, Veneziani M, Doyle JD. 2009a. Application of 4D-variational data assimilation to the California Current System. *Dyn. Atmos. Oceans.* **48**: 69–91.

Broquet G, Moore AM, Arango HG, Edwards CA, Powell BS. 2009b. Ocean state and surface forcing correction using the ROMS-IS4DVAR data assimilation system. *Mercator Ocean Quarterly Newsletter.* **34**: 5–13.

Buehner M. 2005. Ensemble-derived stationary and flow-dependent background-error covariances: evaluation in a quasi-operational NWP setting. *Q. J. R. Meteorol. Soc.* **131**: 1013–1043.

Cardinali C, Pezzulli S, Andersson E. 2004. Influence-matrix diagnostic of a data assimilation system. *Q. J. R. Meteorol. Soc.* **130**: 2767–2786.

Chan TF, Chow E, Saad Y, Yeung MC. 1999. Preserving symmetry in preconditioned Krylov subspace methods. *SIAM J. Sci. Comput.* **20**: 568–581.

Chapman DC. 1985. Numerical treatment of cross-shelf open boundaries in a barotropic coastal ocean model. *J. Phys. Oceanogr.* **15**(8): 1060–1075.

Checkley DM, Barth JA. 2009. Patterns and process in the California Current system. *Prog. Oceanogr.* **83**: 49–64.

Chien CS, Chang SL. 2003. Application of the Lanczos algorithm for solving the linear systems that occur in continuation problems. *Numer. Linear Algebra with Appl.* **10**(4): 335–355, doi:10.1002/nla.306, URL `http://dx.doi.org/10.1002/nla.306`.

Cohn S, Da Silva A, Guo J, Sienkiewicz M, Lamich D. 1998. Assessing the effects of data selection with the DAO physical-space statistical analysis system. *Mon. Weather. Rev.* **126**: 2913–2926.

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

Courtier P, Thépaut JN, Hollingsworth A. 1994. A strategy for operational implementation of 4D-Var using an incremental approach. *Q. J. R. Meteorol. Soc.* **120**: 1367–1388.

Daley R, Barker E. 2001. NAVDAS: Formulation and diagnostics. *Mon. Weather Rev.* **120**: 869–883.

Derber J, Rosati A. 1989. A global oceanic data assimilation system. *J. Phys. Oceanogr.* **19**: 1333–1347.

Desroziers G, Berre L. 2012. Accelerating and parallelizing minimizations in ensemble and deterministic variational assimilations. *Q. J. R. Meteorol. Soc.* DOI:10.1002/qj.1886.

Doyle JD, Jiang Q, Chao Y, Farrara J. 2009. High-resolution atmospheric modeling over the Monterey Bay during AOSN II. *Deep Sea Res. II* **56**: 87–99.

Egbert GD, Bennett AF, Foreman MCG. 1994. TOPEX/POSEIDON tides estimated using a global inverse method. *J. Geophys. Res.* **99**(24): 821–24,852.

El Akkraoui A, Gauthier P. 2010. Convergence properties of the primal and dual forms of variational data assimilation. *Q. J. R. Meteorol. Soc.* **136**: 107–115.

El Akkraoui A, Gauthier P, Pellerin S, Buis S. 2008. Intercomparison of the primal and dual formulations of variational data assimilation. *Q. J. R. Meteorol. Soc.* **134**: 1015–1025.

El Akkraoui A, Trémolet Y, Todling R. 2012. Preconditioning of variational data assimilation and the use of a bi-conjugate gradient method. *Q. J. R. Meteorol. Soc.* DOI:10.1002/qj.1997.

Evensen G. 2009. *Data Assimilation: The Ensemble Kalman Filter*. Springer Verlag: Berlin.

Fairall CW, Bradley EF, Godfrey JS, Wick GA, Ebson JB, Young GS. 1996a. Cool-skin and warm layer effects on the sea surface temperature. *J. Geophys. Res.* **101**: 1295–1308.

Fairall CW, Bradley EF, Rogers DP, Ebson JB, Young GS. 1996b. Bulk parameterization of air-sea fluxes for tropical ocean global atmosphere coupled-ocean atmosphere response experiment. *J. Geophys. Res.* **101**: 3747–3764.

Fisher M. 2003. Background error covariance modelling. In: *Proceedings of the ECMWF Seminar on Recent Developments in Data Assimilation for Atmosphere and Ocean.* ECMWF, Reading, UK, pp. 87–101.

Fisher M, Nocedal J, Trémolet Y, Wright SJ. 2009. Data assimilation in weather forecasting: A case study in PDE-constrained optimization. *Optim. Eng.* **10**: 409–426.

Flather RA. 1976. A tidal model of the northwest European continental shelf. *Mem. Soc. R. Sci. Liege* **10**(6): 141–164.

Gelaro R, Zhu Y. 2009. Examination of observation impacts derived from observing system experiments (OSEs) and adjoint models. *Tellus.* **61A**: 179–193.

Golub GH, Van Loan CF. 1996. *Matrix Computations.* Johns Hopkins University Press: Baltimore, third edn.

Gratton S, Gürol S, Toint PL. 2012. Preconditioning and globalizing conjugate gradients in dual space for quadratically penalized nonlinear least squares problems. *Comput. Optim. Appl.* doi:10.1007/s10589-012-9478-7. In press.

Gratton S, Laloyaux P, Sartenaer A, Tshimanga J. 2011a. A reduced- and limited-memory preconditioned approach for the 4D-Var data-assimilation problem. *Q. J. R. Meteorol. Soc.* **137**: 452–466.

Gratton S, Lawless A, Nichols NK. 2007. Approximate Gauss-Newton methods for nonlinear least-squares problems. *SIAM J. Optim.* **18**: 106–132.

Gratton S, Sartenaer A, Tshimanga J. 2011b. On a class of limited memory preconditioners for large scale linear systems with multiple right-hand sides. *SIAM J. Optim.* **21**(3): 912–935.

Gratton S, Toint PL, Tshimanga J. 2009. Inexact range-space Krylov solvers for linear systems arising from inverse problems. Technical Report 09/20, Department of Mathematics, FUNDP - University of Namur, Namur, Belgium.

Gratton S, Tshimanga J. 2009. An observation-space formulation of variational assimilation using a Restricted Preconditioned Conjugate-Gradient algorithm. *Q. J. R. Meteorol. Soc.* **135**: 1573–1585.

Haidvogel DB, Arango HG, Budgell WP, Cornuelle BD, Curchitser E, Di Lorenzo E, Fennel K, Geyer WR, Hermann AJ, Lanerolle L, Levin J, McWilliams JC, Miller AJ, Moore AM, Powell TM, Shchepetkin AF, Sherwood CR, Signell RP, Warner JC, Wilkin J. 2008. Ocean forecasting in terrain-following coordinates: Formulation and skill assessment of the Regional Ocean Modeling System. *J. Comput. Phys.* **227**: 3595–3624.

Hickey BM. 1998. Coastal oceanography of western North America from the tip of Baja, California to Vancouver Island. *The Sea.* **11**: 345–393.

Lawless AS, Gratton S, Nichols NK. 2005. Approximate iterative methods for variational data assimilation. *Int. J. Numer. Meth. Fluids.* **47**: 1129–1135.

Liu WT, Katsaros KB, Businger JA. 1979. Bulk parameterization of the air-sea exchange of heat and water vapor including the molecular constraints at the interface. *J. Atmos. Sci.* **36**: 1722–1735.

Lorenc AC. 1997. Development of an operational variational assimilation scheme. *J. Meteorol. Soc. Japan.* **75**: 339–346.

Madec G. 2008. NEMO ocean engine. Technical Report 27, Note du Pôle de modélisation, Institut Pierre-Simon Laplace (IPSL), France, URL `http://www.nemo-ocean.eu`.

Marchesiello P, McWilliams JC, Shchepetkin AF. 2001. Open boundary conditions for long-term integration of regional oceanic models. *Ocean Model.* **3**: 1–20.

Mogensen K, Balmaseda MA, Weaver AT. 2012. The NEMOVAR ocean data assimilation system as implemented in the ECMWF ocean analysis for System 4. Technical Report 668, ECMWF, Reading, U.K.

Mogensen K, Balmaseda MA, Weaver AT, Martin M, Vidard A. 2009. NEMOVAR: A variational data assimilation system for the NEMO model. *ECMWF Newsletter.* **120**: 17–22.

Moore AM, Arango HG, Broquet G, Edwards C, Veneziani M, Powell B, Foley D, Costa D, Robinson P. 2011b. The Regional Ocean Modeling system (ROMS) 4-dimensional variational data assimilation systems. Part II: Performance and application to the California Current System. *Prog. Oceanogr.* **91**: 50–73.

Moore AM, Arango HG, Broquet G, Edwards C, Veneziani M, Powell B, Foley D, Doyle JD, Costa D, Robinson P. 2011c. The Regional Ocean Modeling system (ROMS) 4-dimensional variational data assimilation systems. Part III: Observation impact and observation sensitivity in the California Current System. *Prog. Oceanogr.* **91**: 74–94.

Moore AM, Arango HG, Broquet G, Powell BS, Zavala-Garay J, Weaver AT. 2011a. The Regional Ocean Modeling System (ROMS) 4-dimensional variational data assimilation systems. Part I: System overview and formulation. *Prog. Oceanogr.* **91**: 34–49.

Nocedal J, Wright SJ. 2006. *Numerical Optimization.* Series in Operations Research, Springer Verlag: Heidelberg, Berlin, New York.

Nour-Omid B, Parlett BN, Raefsky A. 1988. Comparison of Lanczos with conjugate gradient using element preconditioning. In: *First International Symposium on Domain Decomposition Methods for Partial Differential Equations.* SIAM, pp. 250–260.

Paige CC, Saunders MA. 1975. Solution of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal.* **12**(4): 617–629.

Papadrakakis M, Smerou S. 1990. A new implementation of the Lanczos method in linear problems. *Int. J. Numer. Meth. Eng.* **29**(1): 141–159, doi:10.1002/nme.1620290110, URL `http://dx.doi.org/10.1002/nme.1620290110`.

Purser RJ, Wu WW, Parrish DF, Roberts NM. 2003. Numerical aspects of the application of recursive filters to variational statistical analysis. Part I: spatially homogeneous and isotropic Gaussian covariances. *Mon. Weather Rev.* : 1524–1535.

Saad Y. 1996. *Iterative Methods for Sparse Linear Systems.* PWS Publishing Company: Boston, USA.

Sasaki YK. 1970. Some basic formalisms in numerical variational analysis. *Mon. Weather. Rev.* **98**: 875–883.

Shchepetkin AF, McWilliams JC. 2003. A method for computing horizontal pressure-gradient force in an oceanic model with a nonaligned vertical grid. *J. Geophys. Res.* **108**(C3), doi:doi:10.1029/2001/JC001047.

Shchepetkin AF, McWilliams JC. 2005. The Regional Oceanic Modeling System (ROMS): a split explicit, free-surface, topography-following-coordinate oceanic model. *Ocean Model.* **9**: 347–404.

Tarantola A. 2005. *Inverse Problem Theory. Methods for Data Fitting and Model Parameter Estimation.* SIAM: Philadelphia, USA.

Trémolet Y. 2006. Accounting for an imperfect model in 4D-Var. *Q. J. R. Meteorol. Soc.* **132**: 2483–2504.

Tshimanga J, Gratton S, Weaver AT, Sartenaer A. 2008. Limited-memory preconditioners with application to incremental four-dimensional variational data assimilation. *Q. J. R. Meteorol. Soc.* **134**: 751–769.

Veneziani M, Edwards CA, Doyle JD, Foley D. 2009a. A central California coastal ocean modeling study: 1. Forward model and the influence of realistic versus climatological forcing. *J. Geophys. Res.* **114**(C04015), doi:doi:10.1029/2008JC004774.

Veneziani M, Edwards CA, Moore AM. 2009b. A central California coastal ocean modeling study: 2. Adjoint sensitivities to local and remote forcing mechanisms. *J. Geophys. Res.* **114**(C04020), doi:doi:10.1029/2008JC004775.

Wang X, Barker DM, Synder C, Hamill TM. 2008. A hybrid ETKF-3DVAR data assimilation scheme for the WRF model. Part I: Observing system simulation experiment. *Mon. Weather Rev.* **136**: 5116–5131.

Wang X, Synder C, Hamill TM. 2007. On the theoretical equivalence of differently proposed ensemble-3DVAR hybrid analysis schemes. *Mon. Weather Rev.* **135**: 222–227.

Weaver AT, Deltel C, Machu E, Ricci S, Daget N. 2005. A multivariate balance operator for variational ocean data assimilation. *Q. J. R. Meteorol. Soc.* **131**: 3605–3625.

Weaver AT, Mirouze I. 2012. On the diffusion equation and its application to isotropic and anisotropic correlation modelling in variational assimilation. *Q. J. R. Meteorol. Soc.* DOI:10.1002/qj.1955.

Weaver AT, Vialard J, Anderson DLT. 2003. Three- and four-dimensional variational assimilation with a general circulation model of the tropical Pacific ocean. Part 1: formulation, internal diagnostics and consistency checks. *Mon. Weather Rev.* **131**: 1360–1378.

Wunsch C, Heimbach P. 2007. Practical global ocean state estimation. *Physica D.* **230**: 197–208.

Algorithm 2.1: BLanczos with re-orthogonalization

```
1  $\mathbf{v}_0 = \mathbf{0}$
2  $\mathbf{r}_0 = \mathbf{G}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{d}$
3  $\mathbf{t}_0 = \mathbf{B}\,\mathbf{r}_0$
4  $\beta_0 = \sqrt{\mathbf{t}_0^{\mathrm{T}}\mathbf{r}_0}$
5  $\mathbf{v}_1 = \mathbf{r}_0/\beta_0$
6  $\mathbf{z}_1 = \mathbf{t}_0/\beta_0$
7  $\beta_1 = 0$
8  $\mathbf{Z}_1 = [\,\mathbf{z}_1\,]$
9  $\mathbf{V}_1 = [\,\mathbf{v}_1\,]$
10 for $i = 1, 2, \ldots, l$ do
11     $\mathbf{q}_i = (\mathbf{v}_i + \mathbf{G}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{G}\,\mathbf{z}_i) - \beta_i\mathbf{v}_{i-1}$
12     $\alpha_i = \mathbf{q}_i^{\mathrm{T}}\mathbf{z}_i$
13     $\mathbf{w}_i = \mathbf{q}_i - \alpha_i\mathbf{v}_i$
14     Re-orthogonalize $\mathbf{w}_i$ using $\mathbf{V}_i$ and $\mathbf{Z}_i$
15     $\mathbf{t}_i = \mathbf{B}\,\mathbf{w}_i$
16     $\beta_{i+1} = \sqrt{\mathbf{t}_i^{\mathrm{T}}\mathbf{w}_i}$
17     If $\beta_{i+1} = 0$, set $l := i$ and go to 27
18     $\mathbf{v}_{i+1} = \mathbf{w}_i/\beta_{i+1}$
19     $\mathbf{z}_{i+1} = \mathbf{t}_i/\beta_{i+1}$
20     $\mathbf{Z}_i := [\,\mathbf{Z}_i,\,\mathbf{z}_{i+1}\,]$
21     $\mathbf{V}_i := [\,\mathbf{V}_i,\,\mathbf{v}_{i+1}\,]$
22     $(\mathbf{T}_i)_{i,i} = \alpha_i$
23     if $i > 1$ then
24         $(\mathbf{T}_i)_{i-1,i} = (\mathbf{T}_i)_{i,i-1} = \beta_i$
25     end
26 end
27 Solve  $\mathbf{T}_l\mathbf{s}_l = \beta_0\,\mathbf{e}_1$
28 $\delta\mathbf{u}_l = \mathbf{Z}_l\,\mathbf{s}_l$
```

Algorithm 2.2: BLanczos without re-orthogonalization

```
$\mathbf{v}_0 = \mathbf{0}$
$\delta\mathbf{u}_0 = \mathbf{0}$
$\mathbf{r}_0 = \mathbf{G}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{d}$
$\mathbf{t}_0 = \mathbf{B}\,\mathbf{r}_0$
$\beta_0 = \sqrt{\mathbf{t}_0^{\mathrm{T}}\mathbf{r}_0}$
$\zeta_1 = \beta_0$
$\mathbf{v}_1 = \mathbf{r}_0/\beta_0$
$\mathbf{z}_1 = \mathbf{t}_0/\beta_0$
$\beta_1 = 0$
$\gamma_1 = 0$
$\mathbf{p}_0 = 0$
for $i = 1, 2, \ldots, l$ do
    $\mathbf{q}_i = (\mathbf{v}_i + \mathbf{G}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{G}\,\mathbf{z}_i) - \beta_i\mathbf{v}_{i-1}$
    $\alpha_i = \mathbf{q}_i^{\mathrm{T}}\mathbf{z}_i$
    if $i > 1$ then
        $\gamma_i = \beta_i/\eta_{i-1}$
        $\zeta_i = -\gamma_i\zeta_{i-1}$
    end
    $\eta_i = \alpha_i - \gamma_i\beta_i$
    $\mathbf{p}_i = (\mathbf{z}_i - \beta_i\mathbf{p}_{i-1})/\eta_i$
    $\delta\mathbf{u}_i = \delta\mathbf{u}_{i-1} + \zeta_i\mathbf{p}_i$
    $\mathbf{w}_i = \mathbf{q}_i - \alpha_i\mathbf{v}_i$
    $\mathbf{t}_i = \mathbf{B}\,\mathbf{w}_i$
    $\beta_{i+1} = \sqrt{\mathbf{t}_i^{\mathrm{T}}\mathbf{w}_i}$
    $\mathbf{v}_{i+1} = \mathbf{w}_i/\beta_{i+1}$
    $\mathbf{z}_{i+1} = \mathbf{t}_i/\beta_{i+1}$
end
```

Algorithm 2.3: RBLanczos with re-orthogonalization

```
1  $\widehat{\mathbf{r}}_0 = \mathbf{R}^{-1}\mathbf{d}$
2  $\widehat{\mathbf{t}}_0 = \mathbf{G}\mathbf{B}\mathbf{G}^{\mathrm{T}}\widehat{\mathbf{r}}_0$
3  $\beta_0 = \sqrt{\widehat{\mathbf{t}}_0^{\mathrm{T}}\widehat{\mathbf{r}}_0}$
4  $\widehat{\mathbf{v}}_1 = \widehat{\mathbf{r}}_0/\beta_0$
5  $\widehat{\mathbf{z}}_1 = \widehat{\mathbf{t}}_0/\beta_0$
6  $\beta_1 = 0$
7  $\widehat{\mathbf{v}}_0 = \mathbf{0}$
8  $\widehat{\mathbf{Z}}_1 := [\,\widehat{\mathbf{z}}_1\,]$
9  $\widehat{\mathbf{V}}_1 := [\,\widehat{\mathbf{v}}_1\,]$
10 for $i = 1, 2, \ldots, l$ do
11     $\widehat{\mathbf{q}}_i = (\widehat{\mathbf{v}}_i + \mathbf{R}^{-1}\widehat{\mathbf{z}}_i) - \beta_i\widehat{\mathbf{v}}_{i-1}$
12     $\alpha_i = \widehat{\mathbf{q}}_i^{\mathrm{T}}\widehat{\mathbf{z}}_i$
13     $\widehat{\mathbf{w}}_i = \widehat{\mathbf{q}}_i - \alpha_i\widehat{\mathbf{v}}_i$
14     Re-orthogonalize $\widehat{\mathbf{w}}_i$ using $\widehat{\mathbf{V}}_i$ and $\widehat{\mathbf{Z}}_i$
15     $\widehat{\mathbf{t}}_i = \mathbf{G}\mathbf{B}\mathbf{G}^{\mathrm{T}}\widehat{\mathbf{w}}_i$
16     $\beta_{i+1} = \sqrt{\widehat{\mathbf{t}}_i^{\mathrm{T}}\widehat{\mathbf{w}}_i}$
17     If $\beta_{i+1} = 0$, set $l := i$ and go to 27
18     $\widehat{\mathbf{v}}_{i+1} = \widehat{\mathbf{w}}_i/\beta_{i+1}$
19     $\widehat{\mathbf{z}}_{i+1} := \widehat{\mathbf{t}}_i/\beta_{i+1}$
20     $\widehat{\mathbf{Z}}_i := [\,\widehat{\mathbf{Z}}_i,\, \widehat{\mathbf{z}}_{i+1}\,]$
21     $\widehat{\mathbf{V}}_i := [\,\widehat{\mathbf{V}}_i,\, \widehat{\mathbf{v}}_{i+1}\,]$
22     $(\mathbf{T}_i)_{i,i} = \alpha_i$
23     if $i > 1$ then
24         $(\mathbf{T}_i)_{i-1,i} = (\mathbf{T}_i)_{i,i-1} = \beta_i$
25     end
26 end
27 Solve $\mathbf{T}_l\mathbf{s}_l = \beta_0\mathbf{e}_1$
28 $\delta\mathbf{u}_l = \mathbf{B}\mathbf{G}^{\mathrm{T}}\widehat{\mathbf{V}}_l\mathbf{s}_l$
```

Algorithm 2.4: RBLanczos without re-orthogonalization

```
$\widehat{\mathbf{r}}_0 = \mathbf{R}^{-1}\mathbf{d}$
$\boldsymbol{\lambda}_0 = \mathbf{0}$
$\widehat{\mathbf{t}}_0 = \mathbf{G}\mathbf{B}\mathbf{G}^{\mathrm{T}}\widehat{\mathbf{r}}_0$
$\beta_0 = \sqrt{\widehat{\mathbf{t}}_0^{\mathrm{T}}\widehat{\mathbf{r}}_0}$
$\zeta_1 = \beta_0$
$\widehat{\mathbf{v}}_1 = \widehat{\mathbf{r}}_0/\beta_0$
$\widehat{\mathbf{z}}_1 = \widehat{\mathbf{t}}_0/\beta_0$
$\beta_1 = 0$
$\widehat{\mathbf{v}}_0 = \mathbf{0}$
$\gamma_1 = 0$
$\widehat{\mathbf{p}}_0 = 0$
for $i = 1, 2, \ldots, l$ do
    $\widehat{\mathbf{q}}_i = (\widehat{\mathbf{v}}_i + \mathbf{R}^{-1}\widehat{\mathbf{z}}_i) - \beta_i\widehat{\mathbf{v}}_{i-1}$
    $\alpha_i = \widehat{\mathbf{q}}_i^{\mathrm{T}}\widehat{\mathbf{z}}_i$
    if $i > 1$ then
        $\gamma_i = \beta_i/\eta_{i-1}$
        $\zeta_i = -\gamma_i\zeta_{i-1}$
    end
    $\eta_i = \alpha_i - \gamma_i\beta_i$
    $\widehat{\mathbf{p}}_i = (\widehat{\mathbf{v}}_i - \beta_i\widehat{\mathbf{p}}_{i-1})/\eta_i$
    $\boldsymbol{\lambda}_i = \boldsymbol{\lambda}_{i-1} + \zeta_i\widehat{\mathbf{p}}_i$
    $\widehat{\mathbf{w}}_i = \widehat{\mathbf{q}}_i - \alpha_i\widehat{\mathbf{v}}_i$
    $\widehat{\mathbf{t}}_i = \mathbf{G}\mathbf{B}\mathbf{G}^{\mathrm{T}}\widehat{\mathbf{w}}_i$
    $\beta_{i+1} = \sqrt{\widehat{\mathbf{t}}_i^{\mathrm{T}}\widehat{\mathbf{w}}_i}$
    $\widehat{\mathbf{v}}_{i+1} = \widehat{\mathbf{w}}_i/\beta_{i+1}$
    $\widehat{\mathbf{z}}_{i+1} := \widehat{\mathbf{t}}_i/\beta_{i+1}$
end
$\delta\mathbf{u}_l = \mathbf{B}\mathbf{G}^{\mathrm{T}}\boldsymbol{\lambda}_l$
```

Algorithm	Formulation	Description	Reference
BCG	Primal	CG on the linear system (22) with the **B**-inner product	Algorithm 4.1
Lanczos (BLanczos)	Primal (Primal)	Lanczos on the linear system (18) with the canonical inner product (equivalent to Lanczos on the linear system (22) with the **B**-inner product)	– (Algorithm 2.1)
RBCG	Dual	CG on the linear system (26) with the $\mathbf{GBG}^{\mathrm{T}}$-inner product	Algorithm 4.2
RBLanczos	Dual	Lanczos on the linear system (26) with the $\mathbf{GBG}^{\mathrm{T}}$-inner product	Algorithm 2.3
dual-Lanczos	Dual	Lanczos on the linear system (27) with the canonical inner product	–
dual-MINRES	Dual	MINRES on the linear system (27) with the canonical inner product	–

Table 1: A summary of the distinguishing characteristics of the different minimization algorithms used in the data assimilation experiments for solving either the primal problem (5) or the dual problem (7).

Algorithm 4.1: BCG

```
$\delta\mathbf{u}_0 = \mathbf{0}$
$\mathbf{f}_0 = \mathbf{0}$
$\mathbf{r}_0 = \mathbf{G}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{d}$
$\mathbf{z}_0 = \mathbf{B}\mathbf{r}_0$
$\mathbf{p}_0 = \mathbf{z}_0$
$\mathbf{h}_0 = \mathbf{r}_0$
for $i = 0, 1, ...l-1$ do
    $\mathbf{q}_i = \mathbf{h}_i + \mathbf{G}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{G}\mathbf{p}_i$
    $\alpha_i = \mathbf{r}_i^{\mathrm{T}}\mathbf{z}_i / \mathbf{q}_i^{\mathrm{T}}\mathbf{p}_i$
    $\delta\mathbf{u}_{i+1} = \delta\mathbf{u}_i + \alpha_i\mathbf{p}_i$
    $\mathbf{f}_{i+1} = \mathbf{f}_i + \alpha_i\mathbf{h}_i$      $\}$ for diagnosing $J_{\mathrm{b}}$
    $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i\mathbf{q}_i$
    Re-orthogonalize $\mathbf{r}_{i+1}$
    $\mathbf{z}_{i+1} = \mathbf{B}\mathbf{r}_{i+1}$
    $\beta_i = \mathbf{r}_{i+1}^{\mathrm{T}}\mathbf{z}_{i+1} / \mathbf{r}_i^{\mathrm{T}}\mathbf{z}_i$
    $\mathbf{p}_{i+1} = \mathbf{z}_{i+1} + \beta_i\mathbf{p}_i$
    $\mathbf{h}_{i+1} = \mathbf{r}_{i+1} + \beta_i\mathbf{h}_i$
end
```

Experiment	Description
BCG_O	NEMO, Global, 3D-Var, BCG with re-orthogonalization
BCG	NEMO, Global, 3D-Var, BCG without re-orthogonalization
RBCG_O	NEMO, Global, 3D-Var, RBCG with re-orthogonalization
RBCG	NEMO, Global, 3D-Var, RBCG without re-orthogonalization
BLanc_OS	ROMS, Regional, Strong-constraint 4D-Var, Lanczos with re-orthogonalization
DLanc_OS	ROMS, Regional, Strong-constraint 4D-Var, dual-Lanczos with re-orthogonalization
DMin_OS	ROMS, Regional, Strong-constraint 4D-Var, dual-MINRES with re-orthogonalization
RBLanc_OS	ROMS, Regional, Strong-constraint 4D-Var, RBLanczos with re-orthogonalization
DLanc_OW	ROMS, Regional, Weak-constraint 4D-Var, dual-Lanczos with re-orthogonalization
DMin_OW	ROMS, Regional, Weak-constraint 4D-Var, dual-MINRES with re-orthogonalization
RBLanc_OW	ROMS, Regional, Weak-constraint 4D-Var, RBLanczos with re-orthogonalization
DMin_W	ROMS, Regional, Weak-constraint 4D-Var, dual-MINRES without re-orthogonalization
RBLanc_W	ROMS, Regional, Weak-constraint 4D-Var, RBLanczos without re-orthogonalization

Table 2: A list of the experiments performed using data assimilation systems with the NEMO and ROMS models. The following naming convention is adopted: the first few letters are short-hand notation for the minimization algorithm used (see Table 1); the letter “O” indicates that re-orthogonalization is used (otherwise it is not used); the letter “S” indicates strong-constraint 4D-Var; and “W” indicates weak-constraint 4D-Var.

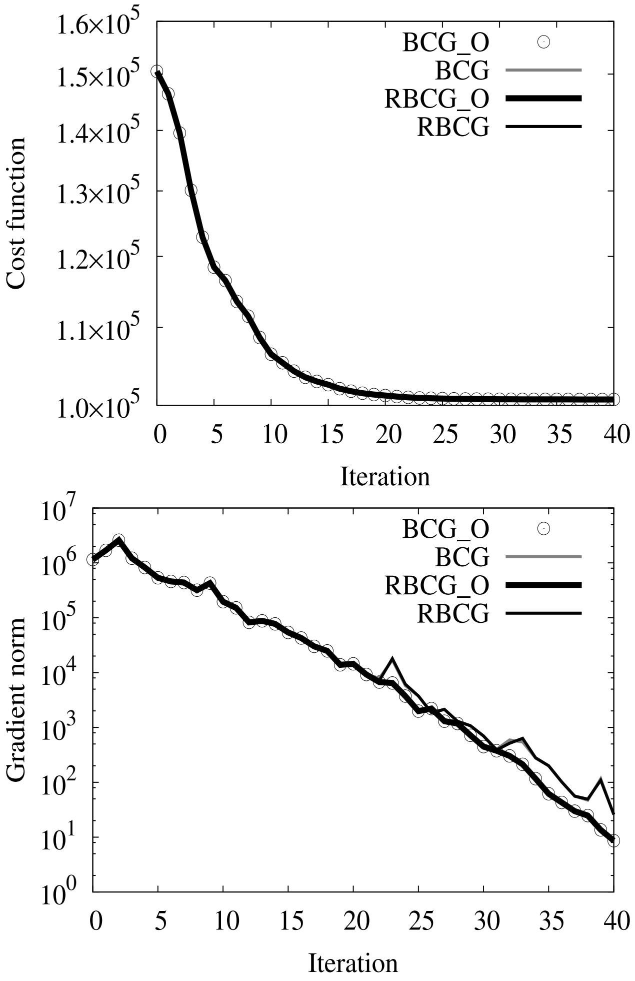

Figure 1: The value of the cost function (upper panel) and gradient norm (lower panel) as a function of the CG iteration counter. The experiments are as described in Table 2. The gradient norm is measured with respect to the **B** metric for BCG and BCG_O, and with respect to the $\mathbf{GBG}^{\mathrm{T}}$ metric for RBCG and RBCG_O. The curves in the upper panel, and the curves corresponding to BCG_O and RBCG_O in the lower panel, are indistinguishable. A logarithmic vertical scale is used in both panels.

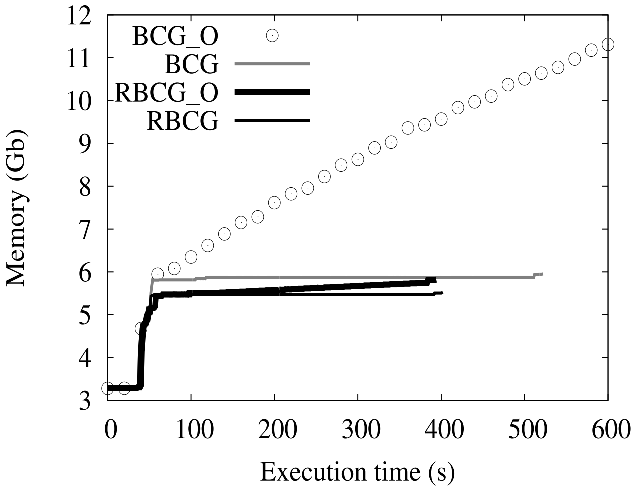

Figure 2: Memory usage (in Giga bytes) as a function of execution time in the minimization loop using CG algorithms based on the primal and dual formulations. See Table 2 for a description of the experiments. The experiments have been performed on a Dell T5500 using a single CPU processor.

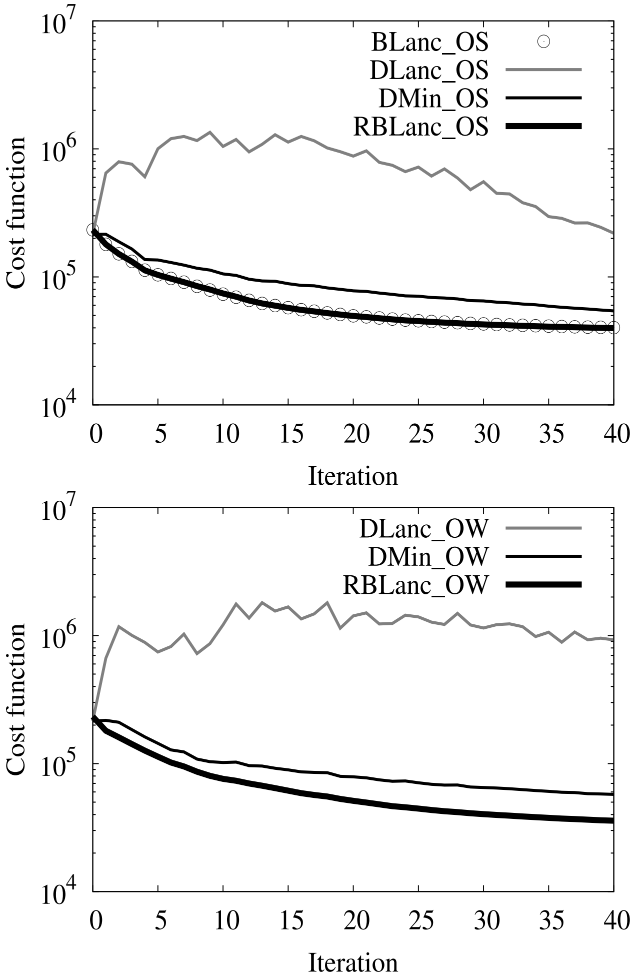

Figure 3: The value of the cost function as a function of the Lanczos iteration counter for the case of strong-constraint 4D-Var in the upper panel and weak-constraint 4D-Var in the lower panel, using the Lanczos algorithms based on the primal and dual formulations with re-orthogonalization. See Table 2 for a description of the experiments. A logarithmic vertical scale is used in both panels.

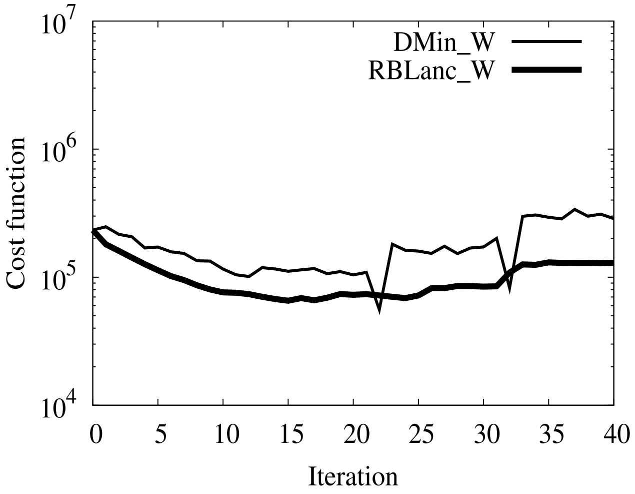

Figure 4: The value of the cost function as a function of the Lanczos iteration counter for the case of weak-constraint 4D-Var without re-orthogonalization of the Lanczos vectors. See Figure 3 caption.

Algorithm 4.2: RBCG

```
$\boldsymbol{\lambda}_0 = \mathbf{0}$
$\mathbf{c}_0 = \mathbf{0}$
$\widehat{\mathbf{r}}_0 = \mathbf{R}^{-1}\mathbf{d}$
$\widehat{\mathbf{p}}_0 = \widehat{\mathbf{r}}_0$
$\mathbf{w}_0 = \mathbf{GBG}^{\mathrm{T}}\widehat{\mathbf{r}}_0$
$\mathbf{t}_0 = \mathbf{w}_0$
for $i = 0, 1, \dots l-1$ do
    $\widehat{\mathbf{q}}_i = \mathbf{R}^{-1}\mathbf{t}_i + \widehat{\mathbf{p}}_i$
    $\alpha_i = \mathbf{w}_i^{\mathrm{T}}\widehat{\mathbf{r}}_i / \widehat{\mathbf{q}}_i^{\mathrm{T}}\mathbf{t}_i$
    $\boldsymbol{\lambda}_{i+1} = \boldsymbol{\lambda}_i + \alpha_i\widehat{\mathbf{p}}_i$
    $\mathbf{c}_{i+1} = \mathbf{c}_i + \alpha_i\mathbf{t}_i$      } for diagnosing $J_{\mathrm{b}}$
    $\widehat{\mathbf{r}}_{i+1} = \widehat{\mathbf{r}}_i - \alpha_i\widehat{\mathbf{q}}_i$
    Re-orthogonalize $\widehat{\mathbf{r}}_{i+1}$
    $\mathbf{w}_{i+1} = \mathbf{GBG}^{\mathrm{T}}\widehat{\mathbf{r}}_{i+1}$
    $\beta_i = \mathbf{w}_{i+1}^{\mathrm{T}}\widehat{\mathbf{r}}_{i+1} / \mathbf{w}_i^{\mathrm{T}}\widehat{\mathbf{r}}_i$
    $\widehat{\mathbf{p}}_{i+1} = \widehat{\mathbf{r}}_{i+1} + \beta_i\widehat{\mathbf{p}}_i$
    $\mathbf{t}_{i+1} = \mathbf{w}_{i+1} + \beta_i\mathbf{t}_i$
end
$\delta\mathbf{u}_l = \mathbf{BG}^{\mathrm{T}}\boldsymbol{\lambda}_l$
```