INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2007; **53**:985–1004
Published online 23 August 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1316

Efficiency of a POD-based reduced second-order adjoint model in 4D-Var data assimilation

D. N. Daescu[1,*,†] and I. M. Navon[2,‡]

[1]*Department of Mathematics and Statistics, Portland State University, Portland, OR 97207, U.S.A.*
[2]*Department of Mathematics and School of Computational Science, Florida State University, Tallahassee, FL 32306, U.S.A.*

SUMMARY

Order reduction strategies aim to alleviate the computational burden of the four-dimensional variational data assimilation by performing the optimization in a low-order control space. The proper orthogonal decomposition (POD) approach to model reduction is used to identify a reduced-order control space for a two-dimensional global shallow water model. A reduced second-order adjoint (SOA) model is developed and used to facilitate the implementation of a Hessian-free truncated-Newton (HFTN) minimization algorithm in the POD-based space. The efficiency of the SOA/HFTN implementation is analysed by comparison with the quasi-Newton BFGS and a nonlinear conjugate gradient algorithm. Several data assimilation experiments that differ only in the optimization algorithm employed are performed in the reduced control space. Numerical results indicate that first-order derivative methods are effective during the initial stages of the assimilation; in the later stages, the use of second-order derivative information is of benefit and HFTN provided significant CPU time savings when compared to the BFGS and CG algorithms. A comparison with data assimilation experiments in the full model space shows that with an appropriate selection of the basis functions the optimization in the POD space is able to provide accurate results at a reduced computational cost. The HFTN algorithm benefited most from the order reduction since computational savings were achieved both in the outer and inner iterations of the method. Further experiments are required to validate the approach for comprehensive global circulation models. Copyright © 2006 John Wiley & Sons, Ltd.

Received 17 January 2006; Revised 22 June 2006; Accepted 24 June 2006

KEY WORDS: data assimilation; model reduction; adjoint model; optimization

*Correspondence to: D. N. Daescu, Department of Mathematics and Statistics, Portland State University, P.O. Box 751, Portland, OR 97207, U.S.A.
†E-mail: daescu@pdx.edu
‡E-mail: navon@csit.fsu.edu

Contract/grant sponsor: National Science Foundation Collaboration in Mathematical Geosciences Program; contract/grant number: ATM-0327818

Copyright © 2006 John Wiley & Sons, Ltd.

1. INTRODUCTION

The development of efficient tools to assimilate observational data into models is crucial to the weather/climate analysis and forecast activities. The theoretical foundations of modern data assimilation techniques used in atmospheric sciences and oceanography, including implementation and operational issues, are discussed in the work of Daley [1], Bennett [2], and Kalnay [3]. The four-dimensional variational data assimilation (4D-Var) method [4] combines in a consistent manner a dynamical model, a background estimate, and observational data to provide optimal initial conditions of a dynamical system. The optimality criteria is formulated as the minimization of a cost functional that incorporates the least-squares distance to a prior (background) estimate of the initial state and to time distributed observations in an analysis interval $[t_0, t_N]$. Statistical information on the errors in the background and data is used to define appropriate weights.

A major difficulty in the operational use of 4D-Var for oceanographic and atmospheric global circulation models is the large dimension m of the control space which is the size of the discrete model initial conditions $\mathbf{x}_0 \in R^m$, typically in the range 10^6–10^8. A feasible implementation may be achieved by using the incremental 4D-Var [5] which is currently the method adopted at operational centres implementing 4D-Var [6]. In this approach a succession of quadratic problems is solved in an inner loop using a simplified/coarse resolution model and its adjoint then corrected by full model runs in an outer loop. The method is characterized by the fact that the dimension of the control space remains very large and in practice only a few outer loop iterations are affordable. Veersé and Thépaut [7] provide convergence conditions and analyse the quality of the incremental solution. Issues related to the linearization, accuracy and preconditioning of the inner loop minimization and the overall (outer loop) convergence of the method are discussed by Trémolet [8, 9].

Significant research efforts to reduce the computational burden of the data assimilation by reducing the dimension of the system have centred on Kalman and extended Kalman filter methods (see, for example, References [10–13]). Model reduction techniques [14, 15] may be used in the 4D-Var data assimilation context to define a reduced-order control space. To drastically decrease the dimension of the control space without significantly compromising the quality of the solution, the reduced space must capture most of the energy and the main directions of variability of the model. If such a space exists, one would then attempt to control the vector of initial conditions by projecting it into the low-dimensional space.

The proper orthogonal decomposition (POD) method (also known as the Karhunen–Loève decomposition, principal component analysis) is a singular value decomposition (SVD)-based method that provides an optimal representation of a data set into a subspace of lower dimension. Theoretical foundations and recent applications of the methodology may be found in References [14–18]. A reduced-order strategy in oceanic 4D-Var data assimilation was considered by Blayo *et al*. [19], Durbiano [20], and Hoteit *et al*. [21]. They used a low-dimensional space based on first few empirical orthogonal functions (EOFs) appropriately chosen from a sampling of the model trajectory (EOFs are basically equivalent to the POD basis). Recently, Robert *et al*. [22] implemented an EOFs-based reduction method for the incremental 4D-Var.

The experience gained from the application of the EOFs/POD procedure to 4D-Var data assimilation problems indicates that the computational savings result mainly from speeding up the optimization by controlling the initial conditions in a subspace of much lower dimension. The efficiency of the optimization algorithm is therefore crucial for the practical implementation and performance of the reduced-order control strategy. Typically, the number of iterations may

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

be reduced by on order of magnitude as compared to controlling the full model initial state. So far, quasi-Newton and conjugate gradient algorithms have been the methods of choice since they require only first-order derivatives of the cost functional. The gradient in the reduced control space may be computed with a full model forward/adjoint integration to obtain the 'full' gradient which is then projected onto the reduced space [20, 21]. Truncated-Newton (TN) minimization methods [23, 24] can be highly effective for solving the optimization problem since they take full advantage of the first- and second-order derivative information. The Hessian-free implementation (HFTN) requires only Hessian/vector products that may be obtained from a second-order adjoint model as explained in Reference [25].

In this work a POD-based reduced second-order adjoint (SOA) model is developed for a two-dimensional global shallow water model and used to implement the HFTN optimization algorithm in the POD space. The implementation and use of a second-order adjoint model to the reduced control optimization problem is *per se* a novel element and was not previously considered in data assimilation. Several numerical experiments that differ only in the optimization algorithm employed are first set up in the POD control space. In each case the efficiency of the SOA/HFTN implementation is analysed by comparison with the quasi-Newton BFGS and the nonlinear conjugate gradient algorithm of Shanno and Phua [26]. Data assimilation experiments in the full model space are then performed and are used to assess the potential computational savings of the reduced-order control procedure.

The paper is organized as follows: the 4D-Var data assimilation problem is set up in a twin experiments framework in Section 2 and the construction of the POD-based reduced control space is presented in Section 3. Implementation of the discrete second-order adjoint model, evaluation of the derivatives in the reduced space and verification are discussed in Section 4. The minimization algorithms are briefly revised in Section 5 and the use of the reduced SOA model in the HFTN algorithm is explained. Numerical results, analysis, and concluding remarks are presented in Section 6.

2. THE 4D-VAR DATA ASSIMILATION SET-UP

We use a two-dimensional global shallow water model in spherical coordinates. The state variables are the zonal and meridional wind components u and v, respectively, and the height of the homogeneous atmosphere, h. A full model description may be found in Reference [27]. We consider a spatial discretization on a 72×36 grid ($5^\circ \times 5^\circ$) such that the dimension of the discrete state vector $\mathbf{x} = (u, v, h)$ is $m = 7776$. The software developed by Giraldo and Neta [28] was used to implement the numerical integration with an explicit Turkel-Zwas scheme and a time step $\Delta t = 200\,\mathrm{s}$. An analytic geopotential height field as in References [27, 29] is specified at the initial time

$$h_0(\lambda, \theta) = \frac{1}{g}(\overline{\Phi} + 2\Omega a \vartheta \sin^3\theta \cos\theta \sin\lambda) \tag{1}$$

and the initial velocities u_0, v_0 are derived from the geostrophic relations; λ and θ denote the longitudinal and latitudinal coordinates, respectively, and the values of the constants were specified $g = 9.8\,\mathrm{m\,s^{-2}}$, $\overline{\Phi} = 5.768 \times 10^4\,\mathrm{m^2\,s^{-2}}$, $\vartheta = 20\,\mathrm{m\,s^{-1}}$, $a = 6.37 \times 10^6\,\mathrm{m}$, $\Omega = 7.292 \times 10^{-5}\,\mathrm{rad\,s^{-1}}$. The initial state of the model is displayed in Figure 1. We will refer to this configuration as the reference state at the initial time and denote it by $\mathbf{x}_0^{\mathrm{ref}}$.

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

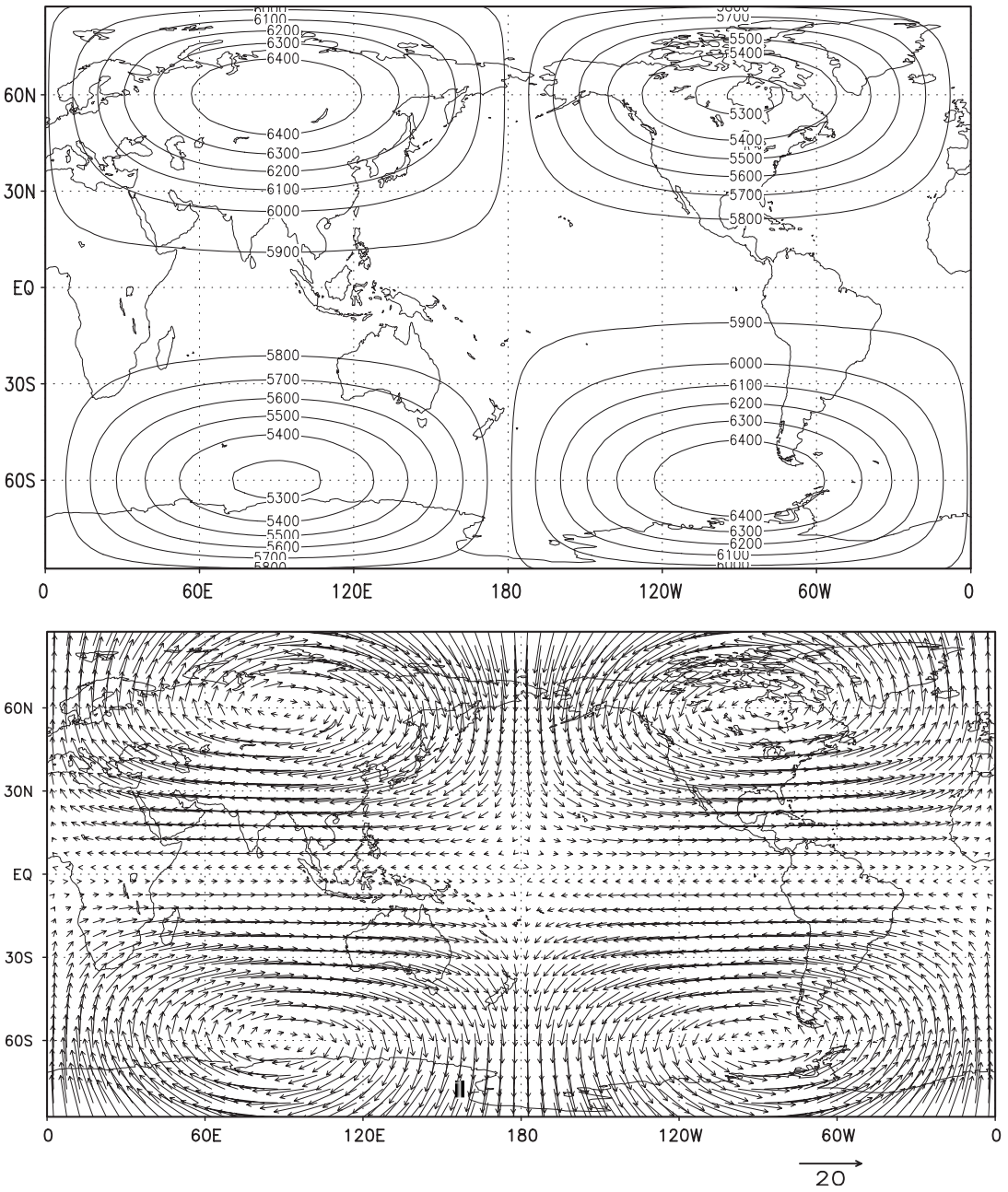

Figure 1. Configuration of the reference initial state of the 2D global shallow water model. Top figure: isopleths of the geopotential height field h in meters. Bottom figure: the velocity vector field (u, v).

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

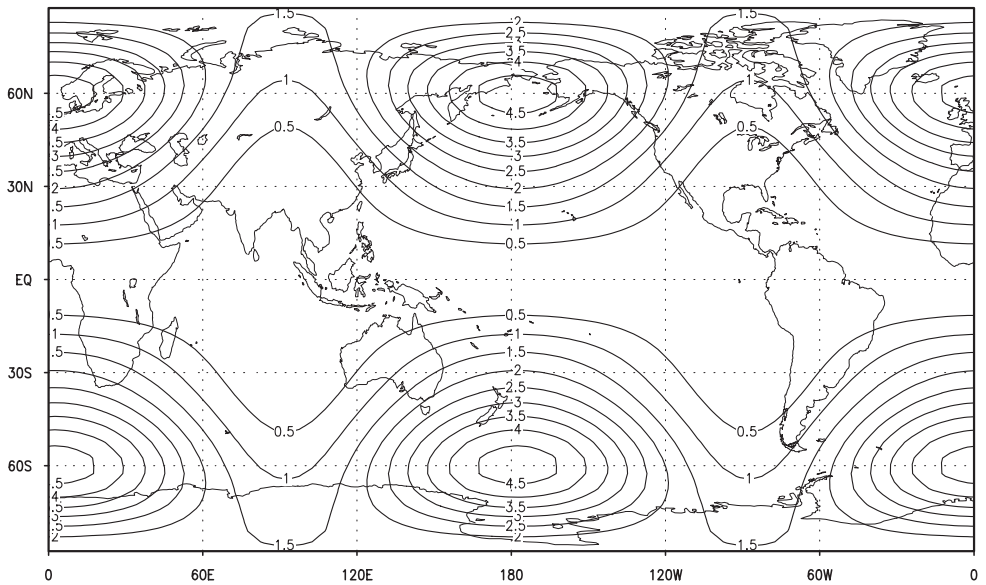

Figure 2. Isopleths of the errors in the first guess (background) estimate to initial conditions. At each grid point the magnitude of the error is evaluated in the total energy norm.

The cost functional in the 4D-Var data assimilation is defined as

$$\mathscr{J} = \mathscr{J}^{\mathrm{b}} + \mathscr{J}^{\mathrm{o}} = \frac{1}{2}(\mathbf{x}_0 - \mathbf{x}^{\mathrm{b}})^{\mathrm{T}}\mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}^{\mathrm{b}}) + \frac{1}{2}\sum_{k=0}^{N}(\mathbf{H}_k\mathbf{x}_k - \mathbf{x}_k^{\mathrm{o}})^{\mathrm{T}}\mathbf{R}_k^{-1}(\mathbf{H}_k\mathbf{x}_k - \mathbf{x}_k^{\mathrm{o}}) \tag{2}$$

where $\mathbf{x}_k$ is the discrete model state at time t_k, $\mathbf{x}^{\mathrm{b}}$ is the prior estimate for the initial condition (background), $\mathbf{B}$ is the covariance matrix of the background errors, $\mathbf{R}_k$ is the covariance matrix of the observational errors, and $\mathbf{H}_k$ is the observational operator that maps the state space into observations $\mathbf{x}_k^{\mathrm{o}}$ taken at time t_k. Since in practice it is difficult to provide statistical information on the errors, simplifying assumptions need to be made [30]. Without statistical interpretation, $\mathbf{B}$ and $\mathbf{R}_k$ specify appropriate weights in the least-squares fitting of model to data. The model equations $\mathbf{x}_k = \mathscr{M}(\mathbf{x}_0)$ are used to express the state evolution in terms of the initial conditions, and the optimal initial state (analysis) $\mathbf{x}_0^{\mathrm{a}}$ is obtained by solving the unconstrained optimization problem

$$\min_{\mathbf{x}_0} \mathscr{J}(\mathbf{x}_0); \quad \mathbf{x}_0^{\mathrm{a}} = \mathrm{Arg}\min \mathscr{J} \tag{3}$$

The twin experiments framework for the 4D-Var data assimilation is set in a time interval of length six hours. A model integration initiated from $\mathbf{x}_0^{\mathrm{ref}}$ is first performed to obtain a reference state trajectory. 'Observations' $\mathbf{x}_k^{\mathrm{o}}$ are selected from the reference run at a time increment of 10 min (every third time step) at all grid points. A first guess approximation $\mathbf{x}^{\mathrm{b}}$ to the reference initial conditions is constructed by shifting $\mathbf{x}_0^{\mathrm{ref}}$ one grid point in the longitudinal direction $\mathbf{x}^{\mathrm{b}}(\lambda, \theta) = \mathbf{x}_0^{\mathrm{ref}}(\lambda - \Delta\lambda, \theta)$. The errors in the background estimate, $\delta\mathbf{x}_0 = \mathbf{x}_0^{\mathrm{ref}} - \mathbf{x}^{\mathrm{b}}$, are shown in Figure 2 in the total energy norm $\|\delta\mathbf{x}\|^2 = \frac{1}{2}((\delta u)^2 + (\delta v)^2) + \frac{g}{\overline{h}}(\delta h)^2$, where $g\overline{h} = \overline{\Phi}$. In the cost functional the distance to

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

background and observations is also measured in the total energy norm such that the weights are diagonal matrices. By solving the minimization problem (3) with the initial guess $\mathbf{x}_0 = \mathbf{x}^{\mathrm{b}}$ we attempt to recover $\mathbf{x}_0^{\mathrm{a}} \approx \mathbf{x}_0^{\mathrm{ref}}$.

3. THE REDUCED-ORDER CONTROL SPACE

The POD approach to model reduction [14] is used to formulate the optimization problem in a low-order control space. To implement the method of snapshots, an ensemble data set $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}\}$, $\mathbf{x}^{(i)} \in R^m$ is collected from the state evolution at various instants in the analysis time interval. The mean of data is computed $\overline{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}^{(i)}$ and a modified $m \times n$-dimensional matrix of snapshots is constructed

$$\mathbf{X} = [\mathbf{x}^{(1)} - \overline{\mathbf{x}}, \mathbf{x}^{(2)} - \overline{\mathbf{x}}, \ldots, \mathbf{x}^{(n)} - \overline{\mathbf{x}}] \tag{4}$$

In practice, $m \gg n$. The 'thin' singular value decomposition $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{T}}$ is performed [31] where $\mathbf{U} \in R^{m \times n}$, $\mathbf{V} \in R^{n \times n}$ have orthogonal columns, $\mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{I}_n$ and $\boldsymbol{\Sigma} \in R^{n \times n}$ is a diagonal matrix $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n)$ whose entries $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_n \geqslant 0$ are the singular values of $\mathbf{X}$. The columns of $\mathbf{U}$, $\mathbf{u}_i \in R^m$, are the left singular vectors and the columns of $\mathbf{V}$, $\mathbf{v}_i \in R^n$, are the right singular vectors. For $k < n$, the truncated SVD approximation

$$\mathbf{X} \approx \mathbf{X}_k = \mathbf{U}_k \boldsymbol{\Sigma}_k \mathbf{V}_k^{\mathrm{T}} \tag{5}$$

$\mathbf{U}_k = [\mathbf{u}_1, \ldots, \mathbf{u}_k]$, $\boldsymbol{\Sigma}_k = \mathrm{diag}(\sigma_1, \ldots \sigma_k)$, $\mathbf{V}_k = [\mathbf{v}_1, \ldots, \mathbf{v}_k]$ is an optimal low-rank representation of the snapshot matrix in the sense that

$$\|\mathbf{X} - \mathbf{X}_k\|_{\mathrm{F}}^2 = \min_{\mathrm{rank}\,\mathbf{Y} \leqslant k} \|\mathbf{X} - \mathbf{Y}\|_{\mathrm{F}}^2 = \sum_{i=k+1}^{n} \sigma_i^2 \tag{6}$$

where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius matrix norm. The percentage of total information captured by the reduced space is $I(k) = (\sum_{i=1}^{k} \sigma_i^2)/(\sum_{i=1}^{n} \sigma_i^2)$ and in practice, given a tolerance $0 < \gamma \leqslant 1$ in the vicinity of the unity, k must be selected such that $I(k) \geqslant \gamma$.

The reduced-order control is obtained by projecting $\mathbf{x}_0 - \overline{\mathbf{x}}$ onto the POD space spanned by the first k left singular vectors:

$$\mathscr{P}_{\mathrm{POD}}(\mathbf{x}_0 - \overline{\mathbf{x}}) = \mathbf{U}_k \boldsymbol{\eta} = \sum_{i=1}^{k} \eta_i \mathbf{u}_i \tag{7}$$

The coordinates vector of the projection in the POD space is

$$\boldsymbol{\eta} = \mathbf{U}_k^{\mathrm{T}}(\mathbf{x}_0 - \overline{\mathbf{x}}) \in R^k \tag{8}$$

and the minimization problem is thus reduced to finding the optimal coefficients $\boldsymbol{\eta}$

$$\hat{\mathscr{J}}(\boldsymbol{\eta}) := \mathscr{J}(\overline{\mathbf{x}} + \mathbf{U}_k \boldsymbol{\eta}); \quad \min_{\boldsymbol{\eta} \in R^k} \hat{\mathscr{J}}(\boldsymbol{\eta}) \tag{9}$$

If $\boldsymbol{\eta}^{\mathrm{a}}$ denotes the solution to (9), an approximation to the analysis (3) is obtained as

$$\mathbf{x}_0^{\mathrm{a}} \approx \overline{\mathbf{x}} + \mathbf{U}_k \boldsymbol{\eta}^{\mathrm{a}} \tag{10}$$

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

Remark
Equivalently [15], the POD space may be determined by finding the eigenpairs $(v_i, \boldsymbol{\psi}_i)$, $\boldsymbol{\psi}_i^{\mathrm{T}}\boldsymbol{\psi}_i = 1$ of the $n \times n$ correlation matrix

$$\mathbf{C} = \frac{1}{n}\mathbf{X}^{\mathrm{T}}\mathbf{X}, \quad \mathbf{C}\boldsymbol{\psi}_i = v_i\boldsymbol{\psi}_i$$

The POD basis vectors are defined as $\boldsymbol{\Phi}_i = (1/\sqrt{nv_i})\mathbf{X}\boldsymbol{\psi}_i$, and the equivalence follows from the relations $\sigma_i^2 = nv_i$, $\mathbf{v}_i = \boldsymbol{\psi}_i$, $\mathbf{X}\mathbf{v}_i = \sigma_i\mathbf{u}_i$.

4. THE DISCRETE SECOND-ORDER ADJOINT MODEL

Given the time evolution of the nonlinear discrete model

$$\mathbf{x}_{k+1} = \mathscr{M}_k(\mathbf{x}_k), \quad k = 0, 1, \ldots, N-1 \tag{11}$$

the gradient of the cost functional is evaluated through the backward integration of the adjoint equations

$$\boldsymbol{\lambda}_{N+1} = 0; \quad \boldsymbol{\lambda}_k = \mathbf{M}_k^{\mathrm{T}}\boldsymbol{\lambda}_{k+1} + \mathbf{H}_k^{\mathrm{T}}\mathbf{R}_k^{-1}(\mathbf{H}_k\mathbf{x}_k - \mathbf{x}_k^{\mathrm{o}}), \quad k = N, N-1, \ldots, 0 \tag{12}$$

where $\mathbf{M}_k = (\mathscr{M}_k)_{\mathbf{x}_k}(\mathbf{x}_k)$ is the *state-dependent* Jacobian of the model, to obtain

$$\nabla_{\mathbf{x}_0}\mathscr{J} = \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}^{\mathrm{b}}) + \boldsymbol{\lambda}_0 \tag{13}$$

The product $\nabla^2_{\mathbf{x}_0}\mathscr{J}\mathbf{w}$ of the Hessian matrix of the cost with a user-defined vector $\mathbf{w}$ is the $\mathbf{w}$-directional derivative of the application $\mathbf{x}_0 \to \nabla_{\mathbf{x}_0}\mathscr{J}$. The computation, known as the forward over reverse procedure, requires the linearization of the forward–backward equations (11)–(12) with respect to the state and adjoint variables:

$$\boldsymbol{\mu}_0 = \mathbf{w}; \quad \boldsymbol{\mu}_{k+1} = \mathbf{M}_k\boldsymbol{\mu}_k, \quad k = 0, 1, \ldots, N-1 \tag{14}$$

$$\boldsymbol{\zeta}_{N+1} = 0; \quad \boldsymbol{\zeta}_k = \mathbf{M}_k^{\mathrm{T}}\boldsymbol{\zeta}_{k+1} + (\mathbf{M}_k^{\mathrm{T}}\overline{\boldsymbol{\lambda}}_{k+1})_{\mathbf{x}_k}\boldsymbol{\mu}_k + \mathbf{H}_k^{\mathrm{T}}\mathbf{R}_k^{-1}\mathbf{H}_k\boldsymbol{\mu}_k, \quad k = N, \ldots, 0 \tag{15}$$

In the equation above, $(\mathbf{M}_k^{\mathrm{T}}\overline{\boldsymbol{\lambda}}_{k+1})_{\mathbf{x}_k}$ is a symmetric matrix of second-order derivatives of the model, and the notation $\overline{\boldsymbol{\lambda}}_{k+1}$ indicates that the adjoint variables are treated as constants during the differentiation. The integration of the second-order adjoint (SOA) model (14)–(15) provides the Hessian/vector product as

$$\nabla^2_{\mathbf{x}_0}\mathscr{J}\mathbf{w} = \mathbf{B}^{-1}\mathbf{w} + \boldsymbol{\zeta}_0 \tag{16}$$

Automatic differentiation (AD) tools [32–34] may be used to facilitate the development of the second-order adjoint code. User intervention is often required to optimize the AD generated subroutines and a careful validation of the SOA model needs to be performed. The computational cost is demanding both in terms of the memory storage requirements and the number of floating point operations. The relative CPU ratio $r = \mathrm{CPU}(\nabla^2\mathscr{J}\mathbf{w})/\mathrm{CPU}(\mathscr{J})$ is typically in

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

the range $7 \leqslant r \leqslant 10$ and may increase due to additional checkpointing and data manipulation costs [33, 35].

4.1. Evaluating derivatives in the reduced-order space

The gradient of the cost (9) may be expressed as

$$\nabla_{\boldsymbol{\eta}} \hat{\mathscr{J}}(\boldsymbol{\eta}) = \mathbf{U}_k^{\mathrm{T}} (\nabla_{\mathbf{x}_0} \mathscr{J})_{|\mathbf{x}_0 = \overline{\mathbf{x}} + \mathbf{U}_k \boldsymbol{\eta}} \tag{17}$$

and its evaluation requires a forward integration and trajectory storage of the full model (11) initialized with $\overline{\mathbf{x}} + \mathbf{U}_k \boldsymbol{\eta}$, followed by a full adjoint model (12) integration. The cost function is also computed during the forward run. Given a vector $\boldsymbol{\omega} \in R^k$, the Hessian/vector product in the reduced control space is

$$\nabla^2 \hat{\mathscr{J}}(\boldsymbol{\eta}) \boldsymbol{\omega} = \mathbf{U}_k^{\mathrm{T}} (\nabla^2_{\mathbf{x}_0} \mathscr{J})_{|\mathbf{x}_0 = \overline{\mathbf{x}} + \mathbf{U}_k \boldsymbol{\eta}} \mathbf{U}_k \boldsymbol{\omega} \tag{18}$$

The computation involves a full second-order adjoint model (14)–(15) integration in the direction $\mathbf{w} = \mathbf{U}_k \boldsymbol{\omega}$. A few additional matrix/vector products $\mathbf{U}_k(\cdot), \mathbf{U}_k^{\mathrm{T}}(\cdot)$ are required in (17) and (18) but their computational cost is much lower as compared to the cost of the forward/adjoint model integrations. Evaluating first- and second-order derivatives in the reduced space is thus roughly as expensive as in the full model space. Significant savings may be achieved only through a fast, highly efficient optimization algorithm in the reduced space.

From the implementation point of view, once a first and second-order adjoint for the full model are developed, the reduced gradient and Hessian/vector evaluation require no further software modifications. Switching from the full to reduced space control space is straightforward and an adaptive optimization procedure may be developed. Hoteit *et al*. [21] used a reduced first-order adjoint to show that a hybrid approach may be of benefit for both qualitative and quantitative aspects of the assimilation. In their work the reduced adjoint method was used in the first stages of optimization then switched to full state control.

4.2. The second-order adjoint model verification

The tangent linear, adjoint, and second-order adjoint models were implemented using the TAMC software [36]. Additional user intervention was required to optimize the performance of the automatic generated subroutines. We obtained a relative CPU ratio of the forward and adjoint computation (cost function and gradient combined) to the forward integration of ~ 2.6 and a ratio of ~ 5.7 for the CPU time of the Hessian/vector computation (SOA) to the CPU time of the forward integration. To validate the implementation of the second-order adjoint model in the full and reduced control space, we consider a Taylor series test by defining

$$\xi(\varepsilon) = \frac{\mathscr{J}(\mathbf{x}_0 + \varepsilon \mathbf{w}) - \mathscr{J}(\mathbf{x}_0) - \varepsilon (\nabla \mathscr{J}(\mathbf{x}_0)) \cdot \mathbf{w}}{\frac{1}{2} \varepsilon^2 \mathbf{w} \cdot (\nabla^2 \mathscr{J}(\mathbf{x}_0)) \cdot \mathbf{w}}, \quad \mathbf{w} \in R^n \tag{19}$$

$$\hat{\xi}(\varepsilon) = \frac{\hat{\mathscr{J}}(\boldsymbol{\eta} + \varepsilon \boldsymbol{\omega}) - \hat{\mathscr{J}}(\boldsymbol{\eta}) - \varepsilon (\nabla \hat{\mathscr{J}}(\boldsymbol{\eta})) \cdot \boldsymbol{\omega}}{\frac{1}{2} \varepsilon^2 \boldsymbol{\omega} \cdot (\nabla^2 \hat{\mathscr{J}}(\boldsymbol{\eta})) \cdot \boldsymbol{\omega}}, \quad \boldsymbol{\omega} \in R^k \tag{20}$$

Theoretically, if the SOA implementation is correct, $\lim_{\varepsilon \to 0} \xi(\varepsilon) = 1$, $\lim_{\varepsilon \to 0} \hat{\xi}(\varepsilon) = 1$. Illustrative results are shown in Figure 3.

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

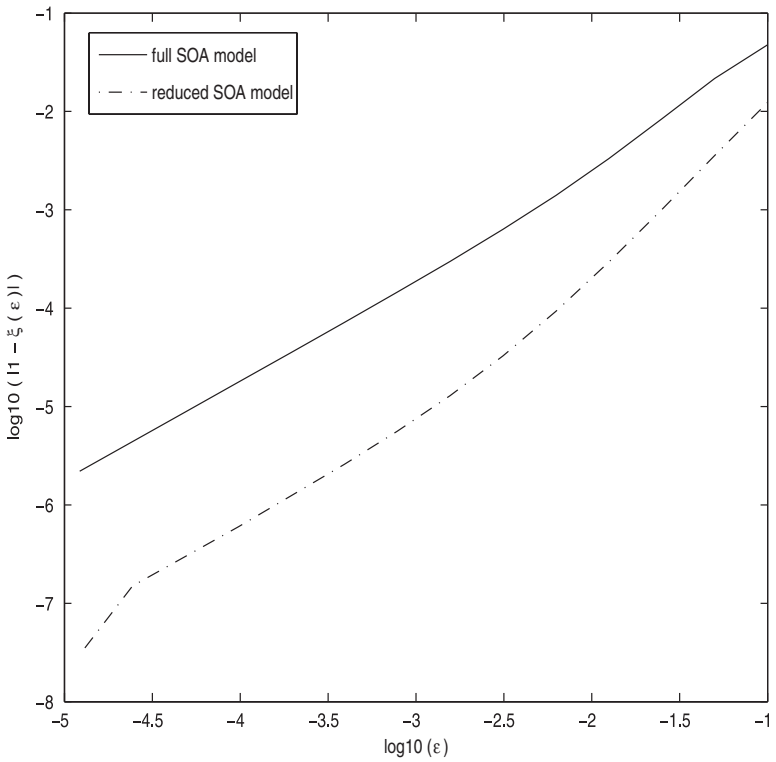

Figure 3. Illustrative results of the Taylor series tests for the verification of the second-order adjoint model in the full space and in the reduced-order space.

5. THE MINIMIZATION ALGORITHMS

The efficiency of the minimization algorithm is a key element of the 4D-Var data assimilation process. The first-order adjoint model is used to provide the gradient of the cost, whereas Hessian/vector products are obtained via SOA model integration. The limited memory L-BFGS method [37, 38] and the truncated-Newton method [24, 39] have been proved to be powerful large-scale unconstrained optimization algorithms that are more efficient than other methods. Due to the large dimensionality of the parameter space, the minimization problem is often ill-conditioned and requires a large number of iterations. Preconditioning algorithms to speed up the convergence of 4D-Var data assimilation are discussed in References [40, 41].

When the optimization is performed in the reduced-order control space, the low dimensionality entails a lower Hessian condition number. In addition, since a modest effort is required to store the reduced state/gradient pairs, full quasi-Newton methods, such as BFGS, may be implemented. Newton's method involves the solution to a low-dimensional linear system:

$$\nabla^2 \hat{\mathscr{J}}(\boldsymbol{\eta}^{(i)})\mathbf{d}^{(i)} = -\nabla \hat{\mathscr{J}}(\boldsymbol{\eta}^{(i)}) \tag{21}$$

The computational burden to obtain the exact Newton's direction $\mathbf{d}^{(i)}$ remains high, since the evaluation of the reduced Hessian matrix requires a number of SOA model integrations equal

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

to the dimension k of the reduced space, typically in the range 10–100. For practical implementation, a Hessian-free truncated-Newton (HFTN) method must be considered. In this approach, each 'outer' iteration $\boldsymbol{\eta}^{(i+1)} = \boldsymbol{\eta}^{(i)} + \alpha^{(i)}\mathbf{d}^{(i)}$ involves an 'inner' iteration where a preconditioned linear conjugate-gradient (CG) method is used to find an approximate solution to the Newton equations (21). The Hessian/vector products required by the CG algorithm are obtained by integrating the reduced SOA model. The inner CG iteration is terminated when the residual

$$\mathbf{r} = \nabla^2 \hat{\mathscr{J}}(\boldsymbol{\eta}^{(i)})\mathbf{d} + \nabla \hat{\mathscr{J}}(\boldsymbol{\eta}^{(i)})$$

satisfies the convergence criteria

$$\|\mathbf{r}\| \leqslant \varepsilon_i \|\nabla \hat{\mathscr{J}}(\boldsymbol{\eta}^{(i)})\|, \quad \varepsilon_i = \min(0.5/i, \|\nabla \hat{\mathscr{J}}(\boldsymbol{\eta}^{(i)})\|)$$

or a user prescribed maximum number of inner iterations *maxit* was achieved. A detailed description of the HFTN method and the preconditioner may be found in References [23, 24].

6. NUMERICAL RESULTS AND DISCUSSION

All the numerical results reported in this work were obtained on a Dell Precision Workstation 670 equipped with an Intel Xeon processor 2.8 GHz and 1 Gb of main memory, using double precision FORTRAN 77.

Data assimilation experiments that differ only in the optimization algorithm employed are first set up in the reduced control space. We investigate the efficiency of the reduced SOA/HFTN method by comparison with the BFGS variable metric algorithm and the Beale-restarted conjugate gradient algorithm provided in the CONMIN package [26]. Two different scenarios to build the snapshot matrix and to define the POD space are considered: in the first set of experiments (hereafter referred to as case 1) snapshots are selected from the observational (reference) data, whereas in the second set of experiments (case 2) snapshots are selected from the model trajectory initiated from the background state. Experiments in case 1 provide insight on the performance of the optimization when a reduced model that accurately approximates the 'true' state evolution is available. Experiments in case 2 correspond to the more realistic situation when the reduced model is based on the best forecast state available.

To assess the potential savings that may be achieved from the low-order control procedure, data assimilation experiments are then performed in the full model space. The large-scale optimization problem is solved using HFTN and nonlinear CG algorithms and the results are compared to the optimization in the POD-reduced-order space.

The ARPACK package [42] was used to perform the singular value decomposition and in each case we found that a dimension $k = 20$ of the POD reduced space was able to capture 99.9% of the total information of the snapshot matrix. The dimension of the control space is thus reduced from $m = 7776$ to $k = 20$. The development of the second-order adjoint model in the full and reduced-order space allowed us to estimate the condition number of the Hessian matrix of the cost functional in the data assimilation. In the full model space the Hessian condition number measured by the max/min eigenvalues ratio was of order 10^6 whereas in the POD space this ratio was reduced to order 10^4.

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

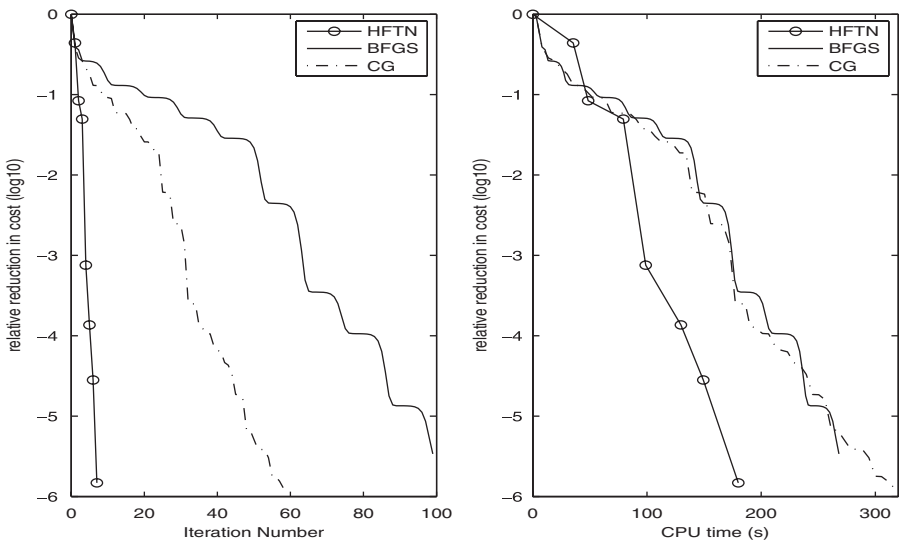

Figure 4. Case 1.1: numerical performance of the optimization algorithms in terms of number of iterations (left figure) and of CPU time (right figure) versus relative reduction in cost functional.

6.1. Experiments in reduced-order space: case 1

Data assimilation numerical experiments are used to assess the efficiency of the HFTN, BFGS and CG optimization algorithms in the reduced space. For all algorithms we use a gradient test $\|\nabla\hat{\mathscr{J}}(\boldsymbol{\eta})\|\leqslant 0.1$ as the convergence criteria and for the HFTN method a number $maxit=5$ was found to be sufficient for the convergence of the inner CG iterations. Two sets of experiments are considered in this case: in case 1.1 the cost functional is defined in terms of the distance to observations only, $\mathscr{J}=\mathscr{J}^{\mathrm{o}}$; in case 1.2 the background term is included in the cost, $\mathscr{J}=\mathscr{J}^{\mathrm{b}}+\mathscr{J}^{\mathrm{o}}$. The performance of the optimization algorithms in terms of the relative reduction in the cost functional versus the number of iterations and versus the CPU time is represented in Figure 4 (case 1.1) and in Figure 5 (case 1.2).

The BFGS and CG methods perform inexpensive iterations, as compared to the HFTN, but the lack of accurate curvature information resulted in a slow convergence rate. Each iteration of the HFTN method involves a high computational effort, but only a few iterations are required to reach a close vicinity of the minimum point.

We used the reduced second-order adjoint model in conjunction with the ARPACK package to provide information on the spectrum of the reduced Hessian matrix. In case 1.1 the extremal eigenvalues of the Hessian matrix of the cost functional evaluated at the initial guess point were $\lambda_{\max}\approx 11.66$ and $\lambda_{\min}\approx 3.76\times 10^{-4}$ which gives a condition number $\lambda_{\max}/\lambda_{\min}\approx 3.10\times 10^{4}$. In case 1.2 the inclusion of the background term in the cost functional provided a small reduction in the Hessian condition number to 2.85×10^{4}.

In both cases the BFGS and CG algorithms are effective during the first stages of the minimization and their overall performance was similar in terms of the CPU time/cost reduction. The HFTN method is much more efficient in the later stages of the minimization where the BFGS and CG methods required as much as 75% more CPU time than HFTN to achieve the same reduction in

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

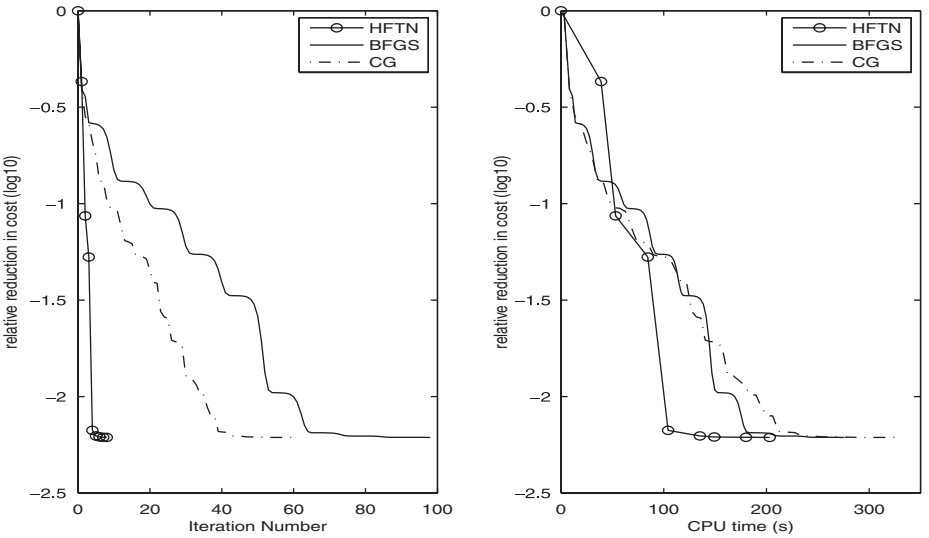

Figure 5. Same as Figure 4, for the data assimilation experiments in case 1.2.

the cost functional. For example, in case 1.1, HFTN reduced the cost to 0.07% of its initial value in four iterations and a CPU time of 98.9 s. To achieve a similar level of reduction, BFGS performed 63 iterations in 174.0 s that represent 75.9% increase in CPU time; CG performed 31 iterations in 172.1 s, an increased CPU time by 74%. These experiments show that in certain applications the computational benefits of implementing the HFTN method with the reduced SOA model may be significant as compared to algorithms that use first-order derivative information only.

From the qualitative point of view, the reduced space accurately captures the dynamics of the reference state evolution. In case 1.1 the solution $\boldsymbol{\eta}^{\mathrm{a}}$ to the reduced problem (9) provided, according to (10), a very close approximation $\mathbf{x}_0^{\mathrm{a}} \approx \mathbf{x}_0^{\mathrm{ref}}$. The analysis errors measured in the total energy norm were in this case of order 10^{-6} (results not shown) and the retrieved initial conditions were nearly identical to the reference state. In case 1.2, the reduced solution is constrained by the background term, yet the assimilation of data reduced the cost to $\sim 0.6\%$ of its initial value and provided a major improvement in the state estimation. The errors in the retrieved initial conditions are displayed in Figure 6 using the total energy norm. We notice a reduction by three orders of magnitude as compared to the backgrounds errors (see Figure 2) showing that also in this case the reduced-order data assimilation procedure was successful.

6.2. Experiments in reduced-order space: case 2

We also consider two subsets of experiments: in case 2.1 we define $\mathscr{J} = \mathscr{J}^{\mathrm{o}}$ and in case 2.2 we define $\mathscr{J} = \mathscr{J}^{\mathrm{b}} + \mathscr{J}^{\mathrm{o}}$. The performance of the optimization algorithms is represented in Figure 7 (case 2.1) and in Figure 8 (case 2.2).

From the computational point of view, in terms of the CPU time required to approach the optimal point, the numerical results agree to our previous findings. The BFGS and CG algorithms are effective in the first stages of the optimization that correspond in this case to the first HFTN iteration. The next two HFTN iterations are significantly more efficient and in both

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

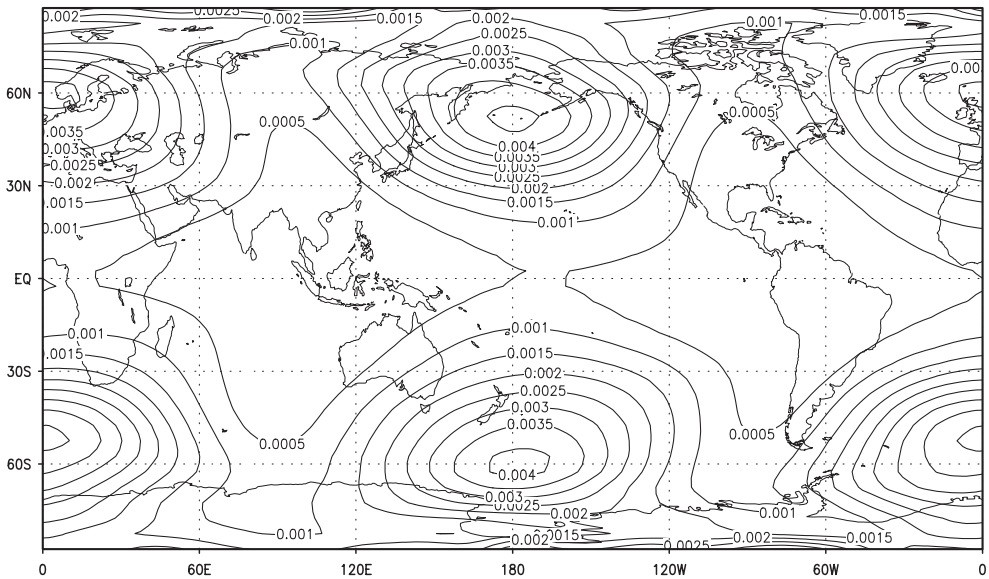

Figure 6. Isopleths of the errors in the retrieved initial conditions measured in the total energy norm. Results corresponding to the data assimilation experiments in case 1.2.

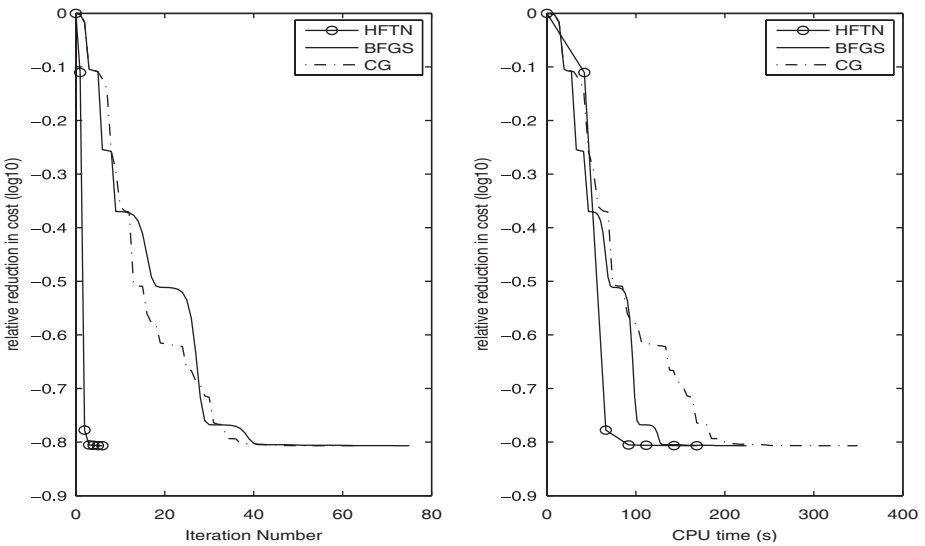

Figure 7. Same as Figure 4, for the data assimilation experiments in case 2.1.

experiments the HFTN algorithm practically reaches the optimal state after only three (outer) iterations. In case 2.1, two HFTN iterations reduced the cost to 16.6% of its initial value and required a CPU time of 66.2 s; to achieve a similar reduction in cost, BFGS performed 37

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

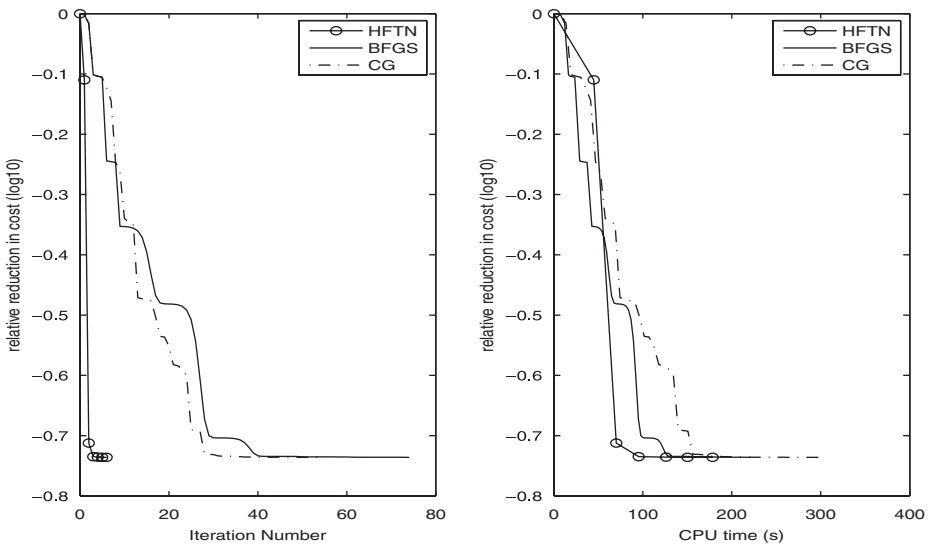

Figure 8. Same as Figure 4, for the data assimilation experiments in case 2.2.

iterations in 121.5 s that represent 83.5% increase in CPU time. In case 2.2, two HFTN iterations reduced the cost to 19.4% of its initial value and required a CPU time of 70.2 s; to achieve a similar reduction in cost, BFGS performed 37 iterations in 119.8 s, a 70.6% increase in CPU time. In this set of experiments, the CG algorithm had a poor performance near the minimum point when compared to the HFTN and BFGS methods. For all algorithms we notice a loss of efficiency near the optimal point indicating that a flexible criteria to terminate the iteration would be of benefit.

From the qualitative point of view, there are major differences as compared to the previous set of experiments. Since in this case the POD space is determined by evolution of the background state, the ability of the reduced control space to provide an accurate analysis representation is limited by the quality of the background estimate. At the optimal point the cost functional was reduced to 15.6% of its initial value in case 2.1 and to 18.3% of its initial value in case 2.2. The errors in the retrieved initial conditions were reduced accordingly and in Figure 9 we show the analysis errors corresponding to case 2.2. While this may be a satisfactory result, the experiments show that in practical applications an adaptive procedure to order reduction is required to achieve accurate estimates.

6.3. Experiments in full model space

To assess the potential computational savings of the reduced-order control procedure data assimilation experiments are performed in the full model space using the nonlinear conjugate gradient and HFTN algorithms. The moderate dimension of the full state vector allowed us to obtain information on the spectrum of the Hessian matrix using the full second-order adjoint model. At both ends of the spectrum the eigenvalues were clustered and, in particular, finding the smallest eigenvalue required an extensive use of the computational resources. For the Hessian matrix $\nabla^2 \mathcal{J}^{\mathrm{o}}$ evaluated at $\mathbf{x}_0 = \mathbf{x}^{\mathrm{b}}$ we computed the eigenvalues $\lambda_{\max} = 770.18$ and $\lambda_{\min} = 2.018 \times 10^{-4}$. The condition

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

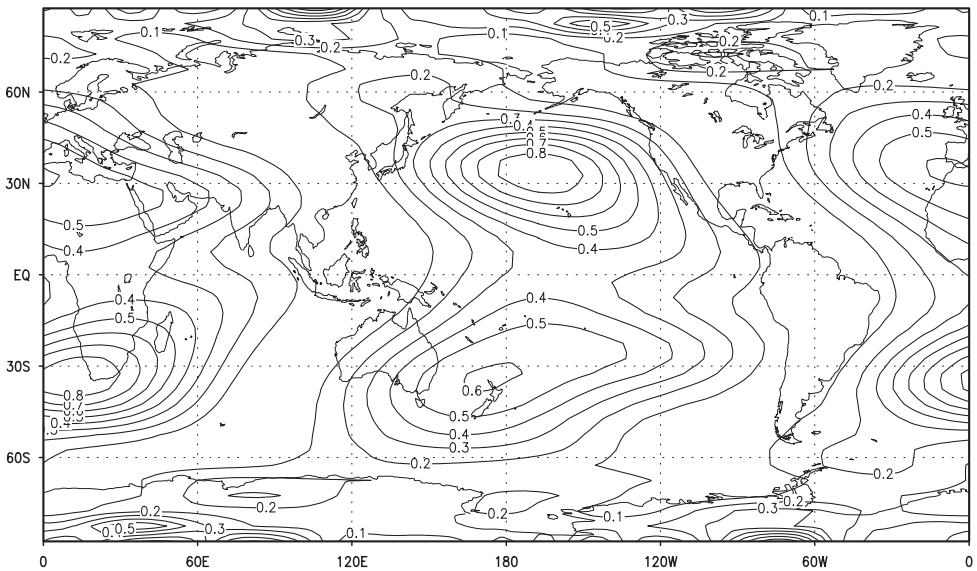

Figure 9. Isopleths of the errors in the retrieved initial conditions measured in the total energy norm. Results corresponding to the data assimilation experiments in case 2.2.

number $\lambda_{\max}/\lambda_{\min} \approx 3.8 \times 10^6$ in the full space is thus two orders of magnitude higher than in the POD space. For the Hessian matrix $\nabla^2(\mathcal{J}^{\mathrm{b}} + \mathcal{J}^{\mathrm{o}})$ the condition number was estimated to be 3.12×10^6.

The optimization is also performed in two cases, first with the cost $\mathcal{J} = \mathcal{J}^{\mathrm{o}}$ then with the distance to background taken into consideration $\mathcal{J} = \mathcal{J}^{\mathrm{b}} + \mathcal{J}^{\mathrm{o}}$. For each case, the performance of the large-scale optimization algorithms is displayed in Figures 10 and 11, respectively. The convergence process is slow and for both algorithms the large condition number entailed an increased number of iterations. It is important to notice that the initial value of the cost functional in the full model space is distinct from the initial value of the cost functional in the POD space. Starting with the initial guess given by the background estimate $\mathbf{x}_0 = \mathbf{x}^{\mathrm{b}}$, the value of the cost in the full model space is $\mathcal{J}(\mathbf{x}^{\mathrm{b}})$. The corresponding initial guess in the reduced space is obtained by projecting the background on the POD space $\boldsymbol{\eta}_0 = \mathbf{U}_k^{\mathrm{T}}(\mathbf{x}^{\mathrm{b}} - \overline{\mathbf{x}})$ thus providing an initial cost value $\hat{\mathcal{J}}(\boldsymbol{\eta}_0) = \mathcal{J}(\overline{\mathbf{x}} + \mathbf{U}_k\boldsymbol{\eta}_0)$. This situation arises also in practical applications where the background is the best prior estimate to the initial conditions and the projection/retrieval operations on/from the POD space may deteriorate the quality of the first guess estimate.

For comparison with the POD procedure, the relative reduction in cost as a metric to assess the efficiency may not be well suited. To account for both computational and qualitative aspects of the reduced/full order optimization, it is of interest to assess the CPU time required to achieve *the same value* of the cost functional, $\hat{\mathcal{J}}(\boldsymbol{\eta}) = \mathcal{J}(\mathbf{x})$. For each algorithm the POD results of case 1.1 and 1.2 are also displayed in Figures 10 and 11, respectively. To facilitate the analysis, all the results are shown in terms of the *value* of the cost functional ($\log_{10}$) versus the number of iterations and versus CPU time. For both algorithms the reduced-order optimization was able to significantly reduce the computational time while preserving the quality of the

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

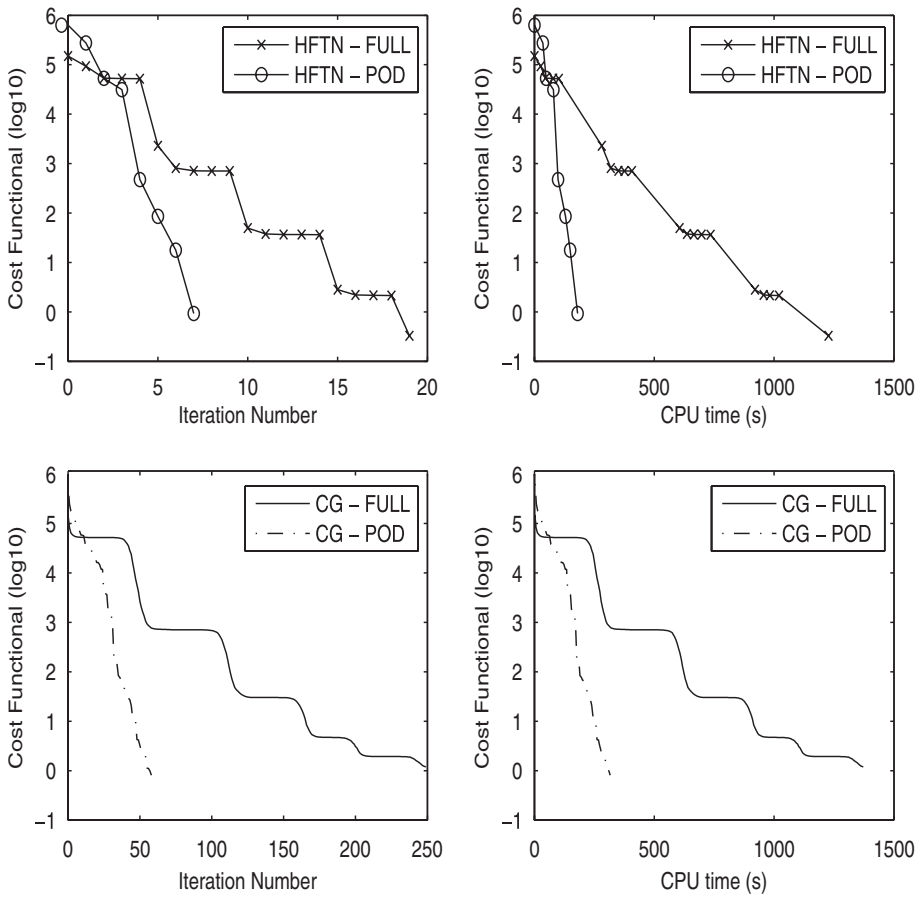

Figure 10. The optimization process in the full model space versus the reduced-order POD space for the HFTN and CG algorithms. The cost functional is defined as the distance to observations only.

solution. When the cost functional was defined as the distance to observations only (Figure 10) the CPU time near the optimal point was reduced by nearly a factor of 5 for the CG algorithm and by as much as a factor of 6 for the HFTN method. An increased efficiency is noticed as the optimization is approaching the optimal point. With the background included in the cost (Figure 11) the POD-CG optimization was able to reduce the CPU time up to a factor of 2 versus the optimization in the full space, whereas the POD-HFTN algorithm provided savings up to a factor of 3.

The increased efficiency of the POD-HFTN is due to the fact that the computational savings in the POD-HFTN versus the full model HFTN iteration are twofold: not only the number of outer iterations is reduced, but also the number of inner iterations required to satisfy the convergence criteria in the inner CG iteration is reduced. In the POD space, a number of inner iterations $maxit = 5$ was proved to be sufficient for convergence whereas in the full space some outer iterations required as many as 30 inner iterations for convergence. An attempt to force early termination (e.g. by setting

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

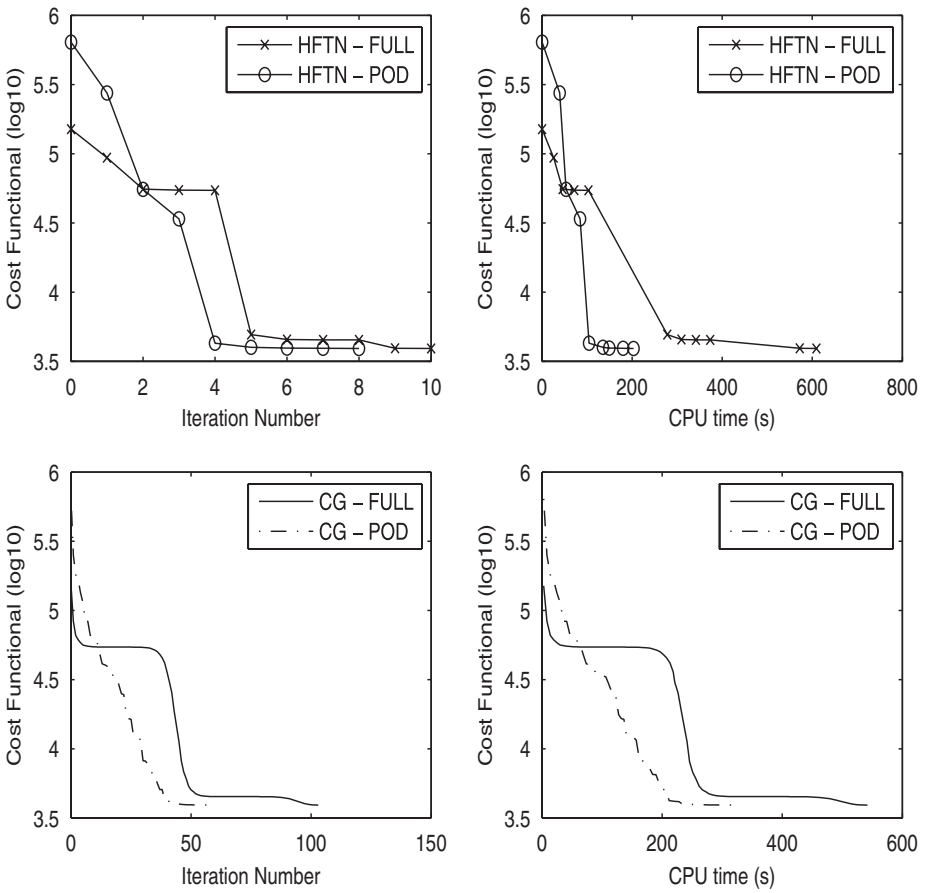

Figure 11. The optimization process in the full model space versus the reduced-order POD space for the HFTN and CG algorithms with the background term included in the cost functional.

maxit = 20) resulted in an increased number of outer iterations and overall loss of efficiency. In the low-order space the HFTN algorithm performs thus fewer and typically less expensive iterations than in the full model space.

6.4. Summary and concluding remarks

Order reduction strategies aim to reduce the computational burden of the 4D-Var data assimilation by performing the optimization in a low-order control space. A successful implementation relies on the identification of a reduced space that accurately captures the dynamics of the full system and an efficient optimization algorithm.

The POD approach to model reduction is used to illustrate the potential benefits that may be achieved by performing the optimization in the low-order control space and, in particular, from the use of a second-order adjoint model in the reduced-order minimization procedure. No claim is made here that the POD method is the optimal approach to order reduction, and in

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

practice other strategies should be also considered, as discussed in the work of Crommelin and Majda [43]. The key element is that once the basis functions are selected (prior to optimization, in an off-line process), the second-order adjoint model may be used to implement the HFTN algorithm in *any* reduced control space. The software modifications required to change the control space are minimal, as only the projection operators need to be constructed.

In this work a reduced second-order adjoint model was developed in a POD-based control space for a 2D global shallow water model. The use of a second-order adjoint model in the reduced control space is novel and was not previously considered in 4D-Var data assimilation. The second-order adjoint model was further used to provide information on the spectrum of the Hessian matrix in the full and reduced-order space. Numerical experiments show that the POD order reduction procedure entailed also a smaller condition number for the optimization problem, and thus an increased computational efficiency. In our study eigenvalues computation was feasible due to the simplicity of the shallow water model and the moderate dimension of the state vector. For realistic 4D-Var applications an attempt to obtain and use such information prior to data assimilation may not be practical since the iterative procedure to estimate the eigenvalues involves several SOA model integrations that may require a higher computational cost than the data assimilation process itself.

Twin experiments were used first in the POD space to investigate the efficiency of the SOA/HFTN implementation in a comparative analysis with the BFGS and CG algorithms. The numerical results show that first-order methods are efficient during the initial stages of the assimilation. The second-order derivative information is particularly effective during the later stages of the optimization and using the SOA/HFTN iterations significant CPU time savings were achieved. For each of CG and HFTN methods a comparison with data assimilation experiments in the full model space shows that with an appropriate selection of the basis functions the optimization in the reduced-order POD space is able to provide accurate results and requires a much lower computational cost. The HFTN algorithm benefited most from the order reduction since computational savings were achieved both in the outer and inner iterations of the method.

The efficiency of the reduced second-order adjoint model remains to be further tested and validated when observational data are assimilated into atmospheric/oceanic models. For practical applications the development and an efficient implementation of the (first- and) second-order adjoint model may involve several model simplifications such as the use of a simplified physics in global circulation models. The impact of the checkpointing schemes, necessary in the adjoint coding of 3D operational models, remains to be assessed.

Generating a 'good' set of snapshots and finding optimal basis functions is crucial for the applicability of the reduced-order procedure. An ensemble of model forecasts may be used to generate snapshots taken from multiple state calculations with perturbations in the initial conditions that capture the main directions of variability of the model such as the bred vectors and singular vectors of the tangent linear model [3]. Adjoint model information may be taken into account using a dual-weighted procedure [44]. For practical applications an adaptive methodology for order reduction must be considered and this is an area where future research is much needed.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the research funding support from the National Science Foundation Collaboration in Mathematical Geosciences Program, award ATM-0327818.

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

REFERENCES

1. Daley R. *Atmospheric Data Analysis*. Cambridge University Press: Cambridge, MA, 1991.
2. Bennett AF. *Inverse Methods in Physical Oceanography*. Cambridge Monographs on Mechanics and Applied Mathematics. Cambridge University Press: Cambridge, MA, 1992.
3. Kalnay E. *Atmospheric Modeling*, *Data Assimilation and Predictability*. Cambridge University Press: Cambridge, MA, 2002.
4. Le Dimet FX, Talagrand O. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus* 1986; **38A**:97–110.
5. Courtier P, Thepaud JN, Hollingsworth A. A strategy of operational implementation of 4D-Var using an incremental approach. *Quarterly Journal of the Royal Meteorological Society* 1994; **120**:1367–1388.
6. Rabier F, Järvinen H, Klinker E, Mahfouf JF, Simmons A. The ECMWF operational implementation of four-dimensional variational data assimilation. Part I: experimental results with simplified physics. *Quarterly Journal of the Royal Meteorological Society* 2000; **126**:1143–1170.
7. Veersé F, Thépaut JN. Multiple-truncation incremental approach for four-dimensional variational data assimilation. *Quarterly Journal of the Royal Meteorological Society* 1998; **124**:1889–1908.
8. Trémolet Y. Diagnostics of linear and incremental approximations in 4D-Var. *Quarterly Journal of the Royal Meteorological Society* 2004; **130**:2233–2251.
9. Trémolet Y. Incremental 4D-Var Convergence Study. *ECMWF Technical Memorandum*, vol. 469, European Centre for Medium-range Weather Forecasts (ECMWF), Reading, U.K., 2005.
10. Dee DP. Simplification of the Kalman filter for meteorological data assimilation. *Quarterly Journal of the Royal Meteorological Society* 1990; **117**:365–384.
11. Todling R, Cohn SE. Suboptimal schemes for atmospheric data assimilation based on the Kalman filter. *Monthly Weather Review* 1994; **122**:2530–2557.
12. Cane MA, Kaplan A, Miller RN, Tang B, Hackert EC, Busalacchi AJ. Mapping tropical Pacific sea level: data assimilation via a reduced state space Kalman filter. *Journal of Geophysical Research* 1996; **101**:22599–22617.
13. Hoteit I, Pham DT. Evolution of the reduced state space and data assimilation schemes based on the Kalman filter. *Journal of the Meteorological Society of Japan* 2003; **81**(1):21–39.
14. Antoulas AC. *Approximation of Large-Scale Dynamical Systems*. Advances in Design and Control, SIAM: Philadelphia, PA, 2005.
15. Gunzburger MD. *Perspectives in Flow Control and Optimization*. Advances in Design and Control, SIAM: Philadelphia, PA, 2003.
16. Rathinam M, Petzold L. A new look at proper orthogonal decomposition. *SIAM Journal on Numerical Analysis* 2003; **41**(5):1893–1925.
17. Ravindran SS. Adaptive reduced-order controllers for a thermal flow system using proper orthogonal decomposition. *SIAM Journal on Scientific Computing* 2002; **23**(6):1924–1942.
18. Ravindran SS. Reduced-order controllers for control of flow past an airfoil. *International Journal for Numerical Methods in Fluids* 2006; **50**:531–554.
19. Blayo E, Blum J, Verron J. Assimilation variationnelle de donées en océanographie et réduction de la dimension de l'espace de contrôle. In *Equations aux Dérivées Partielles et Applications*. Gauthier-Villars: Paris, 1998; 199–219.
20. Durbiano S. Vecteurs caractéristiques de modèles océaniques pour la réduction d'ordre en assimilation de données. *Ph.D. Thesis*, Université Joseph Fourier, Grenoble, France, 2001.
21. Hoteit I, Kohl A, Stammer D. Efficiency of reduced-order time-dependent adjoint data assimilation approaches. *Estimating the Circulation and Climate of the Ocean* (*ECCO*)*—Project Report Series*, *Report No. 27*, 2004.
22. Robert C, Durbiano S, Blayo E, Verron J, Blum J, Le Dimet FX. A reduced order strategy for 4D-Var data assimilation. *Journal of Marine Systems* 2005; **57**:70–82.
23. Nash SG. Preconditioning of truncated-Newton methods. *SIAM Journal on Scientific and Statistical Computing* 1985; **6**:599–616.
24. Nash SG, Nocedal J. A numerical study of the limited memory BFGS method and the truncated-Newton method for large scale optimization. *SIAM Journal on Optimization* 1991; **1**:358–372.
25. Le Dimet FX, Navon IM, Daescu DN. Second order information in data assimilation. *Monthly Weather Review* 2002; **130**(3):629–648.
26. Shanno DF, Phua KH. Remark on 'Algorithm 500: minimization of unconstrained multivariate functions [E4]'. *ACM Transactions on Mathematical Software* 1980; **6**(4):618–622.

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

27. Neta B, Giraldo FX, Navon IM. Analysis of the Turkel-Zwas scheme for the two-dimensional shallow water equations in spherical coordinates. *Journal of Computational Physics* 1997; **133**(1):102–112.
28. Giraldo FX, Neta B. Software for the staggered and unstaggered Turkel-Zwas schemes for the shallow water equations on the sphere. *NPS-MA-95-008, Technical Report Naval Postgraduate School*, Monterey, CA, 1995.
29. McDonald A, Bates JR. Semi-Lagrangian integration of a gridpoint shallow water model on the sphere. *Monthly Weather Review* 1989; **117**(1):130–137.
30. Lorenc AC, Ballard SP, Bell RS, Ingleby NB, Andrews PLF, Barker DM, Bray JR, Clayton AM, Dalby T, Li D, Payne TJ, Saunders FW. The Met. Office global three-dimensional variational data assimilation scheme. *Quarterly Journal of the Royal Meteorological Society* 2000; **126**:2991–3012.
31. Golub GH, Van Loan CF. *Matrix Computations* (3rd edn). John Hopkins University Press: Baltimore, MA, 1996.
32. Giering R, Kaminski T. Recipes for adjoint code construction. *ACM Transactions on Mathematical Software* 1998; **24**(4):437–474.
33. Giering R, Kaminski T, Slawig T. Generating efficient derivative code with TAF: adjoint and tangent linear Euler flow around an airfoil. *Future Generation Computer Systems* 2005; **21**:1345–1355.
34. Rostaing N, Dalmas S, Galligo A. Automatic differentiation in Odyssée. *Tellus* 1993; **45A**:558–568.
35. Griewank A. *Evaluating Derivatives*: *Principles and Techniques of Algorithmic Differentiation*. Frontiers in Applied Mathematics, vol. 19. SIAM: Philadelphia, PA, 2000.
36. Giering R. *Tangent Linear and Adjoint Model Compiler. Users Manual 1.4*. http://www.autodiff.com/tamc [August 1999].
37. Gilbert JC, Lemarechal C. Some numerical experiments with variable-storage quasi-Newton algorithms. *Mathematical Programming* 1989; **45**:407–435.
38. Liu DC, Nocedal J. On the limited memory BFGS method for large scale minimization. *Mathematical Programming* 1989; **45**:503–528.
39. Schlick T, Fogelson A. TNPACK—a truncated Newton minimization package for large-scale problems: I. Algorithm and usage. *ACM Transactions on Mathematical Software* 1992; **18**:46–70.
40. Yang W, Navon IM, Courtier P. A new Hessian preconditioning method applied to variational data assimilation experiments using NASA general circulation models. *Monthly Weather Review* 1996; **124**:1000–1017.
41. Zupanski M. A preconditioning algorithm for four-dimensional variational data assimilation. *Monthly Weather Review* 1996; **124**:2562–2573.
42. Lehoucq RB, Sorensen DC, Yang C. *ARPACK User's Guide*: *Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*. Software, Environments, and Tools, vol. 6. SIAM: Philadelphia, PA, 1998.
43. Crommelin DT, Majda AJ. Strategies for model reduction: comparing different optimal bases. *Journal of Atmospheric Sciences* 2004; **1**:2206–2217.
44. Meyer M, Matthies HG. Efficient model reduction in non-linear dynamics using the Karhunen–Loève expansion and dual-weighted-residual methods. *Computational Mechanics* 2003; **31**:179–191.

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld