Second-Order Information in Data Assimilation*

FRANCOIS-XAVIER LE DIMET

Laboratoire de Modélisation et Calcul, IDOPT Project, Université Joseph Fourier, Grenoble, France

I. M. NAVON

Department of Mathematics and School of Computational Science and Information Technology, The Florida State University, Tallahassee, Florida

DACIAN N. DAESCU

Department of Mathematics, The University of Iowa, Iowa City, Iowa

(Manuscript received 29 December 2000, in final form 24 August 2001)

ABSTRACT

In variational data assimilation (VDA) for meteorological and/or oceanic models, the assimilated fields are deduced by combining the model and the gradient of a cost functional measuring discrepancy between model solution and observation, via a first-order optimality system. However, existence and uniqueness of the VDA problem along with convergence of the algorithms for its implementation depend on the convexity of the cost function. Properties of local convexity can be deduced by studying the Hessian of the cost function in the vicinity of the optimum. This shows the necessity of second-order information to ensure a unique solution to the VDA problem.

In this paper a comprehensive review of issues related to second-order analysis of the problem of VDA is presented along with many important issues closely connected to it. In particular issues of existence, uniqueness, and regularization through second-order properties are examined. The focus then shifts to second-order information related to statistical properties and to issues related to preconditioning and optimization methods and second-order VDA analysis. Predictability and its relation to the structure of the Hessian of the cost functional is then discussed along with issues of sensitivity analysis in the presence of data being assimilated. Computational complexity issues are also addressed and discussed.

Automatic differentiation issues related to second-order information are also discussed along with the computational complexity of deriving the second-order adjoint.

Finally an application aimed at illustrating the use of automatic differentiation for deriving the second-order adjoint as well as the Hessian/vector product applied to minimizing a cost functional of a meteorological problem using the truncated-Newton method is presented. Results verifying numerically the computational cost of deriving the second-order adjoint as well as results related to the spectrum of the Hessian of the cost functional are displayed and discussed.

1. Introduction

Data assimilation can be described as the ensemble of techniques for retrieving geophysical fields from different sources such as observations, governing equations, statistics, etc.

Being heterogeneous in nature, quality, and density, these data sources have to be put together to optimally retrieve (the meaning of ‘‘optimal’’ has to be precisely defined) the geophysical fields. Due to its inherent operational tasks, meteorology has played an important role in the development of data assimilation techniques. An ever-increasing amount of data and models are considered as an ensemble from which the optimal information should be extracted. Behind most of the classical methods used in meteorology such as optimal interpolation, variational methods, statistical estimation, etc., there is a variational principle, that is, the retrieved fields are obtained through the minimization of some functional depending on the various sources of information. The retrieved fields are obtained through some optimality condition, which can be an Euler or Euler–Lagrange condition if the regularity conditions are satisfied. Since these conditions are first-order conditions, it follows that they involve the first-order derivatives of the functional that is minimized. In this sense, data as-

* This work is dedicated to the late Professor Jaques Louis Lions in deep homage.

Corresponding author address: Dr. I. M. Navon, School of Computational Science and Information Technology, The Florida State University, Tallahassee, FL 32306-4120.
E-mail: navon@csit.fsu.edu

similation techniques are first-order methods. But first-order methods are only necessary conditions for optimality but not sufficient ones. To obtain sufficient conditions we need to proceed one step further and to introduce second-order information. By the same token, from the mathematical point of view sensitivity studies with respect to some parameter can be obtained through Gâteaux derivatives with respect to this parameter. Therefore if we seek the sensitivity of fields that have already been defined through some first-order conditions, we will have to proceed one order of derivation further and in this sense our sensitivity studies require second-order information.

The purpose of this review paper is to show how to obtain and how to use in an efficient way second-order information in data assimilation. In a first part we will show how the second-order derivative can be computed, primarily in a very general framework, then illustrate it with some examples. Then we will show how this second-order information can be linked to the issues of uniqueness of a solution to the problem of data assimilation. This will be shown to be not only a mathematical consideration but also rather a practical issue whereby information can be extracted by studying second-order information.

In a second part of the paper we will proceed to show how to derive sensitivity analyses from models and data. The analysis of the impact of uncertainties in the model and in the data provides essential links between purely deterministic methods (such as variational data assimilation) and stochastic methods (Kalman filter–type). We will then proceed to demonstrate how the link between these methods can be clearly understood through use of second-order information.

Researchers in other disciplines have carried out pioneering work using second-order information. Work in seismology using second-order information and applying it to obtain accurate Hessian/vector products for truncated-Newton minimization was carried out by Santosa and Symes (1988, 1989) and by Symes (1990, 1991, 1993). Reuther (1996) and Arian and Ta'asan (1999) illustrated the importance of second-order adjoint analysis for optimal control and shape optimization for inviscid aerodynamics. Hou and Sheen (1993) used second-order sensitivity analysis for heat conduction problems.

Second-order information was tackled in automatic differentiation (AD) by Abate et al. (1997), Giering and Kaminski (1998a,b), Gay (1996), Hovland (1995), Bischof (1995), Burger et al. (1992), Griewank and Corliss (1991), and Griewank (1993, 2000, 2001), to cite but a few. Several AD packages such as the tangent linear and adjoint model compiler (TAMC) of Giering and Kaminski (1998a) allow calculation of the Hessian of the cost functional.

Early work on second-order information in meteorology includes Thacker (1989) followed by work of Wang et al. (1992, 1993) and Wang (1993). Wang et al. (1995, 1998) considered use of second-order information for optimization purposes namely to obtain truncated-Newton and adjoint Newton algorithms using exact Hessian/vector products. Application of these ideas was presented in Wang et al. (1997).

Kalnay et al. (2000) introduced an elegant and novel pseudo-inverse approach and showed its connection to the adjoint Newton algorithm of Wang et al. (1997) (see Kalnay et al. 2000; Pu and Kalnay 1999; Pu et al. 1997).

Ngodock (1996) applied second-order information in conjunction with sensitivity analysis in the presence of observations and applied it to the ocean circulation. Le Dimet et al. (1997) presented the basic theory for second-order adjoint analysis related to sensitivity analysis. A condensed summary of the theory is presented in Le Dimet and Charpentier (1998).

The structure of the paper is as follows. Section 2 deals with the theory of the second-order adjoint method, both for time-independent and time-dependent models. The methodology is briefly illustrated using the shallow-water equations model. Section 3 deals with the connection between sensitivity analysis and second-order information. Section 4 briefly presents the Kalnay et al. (2000) quasi-inverse method and its connection with second-order information. Issues related to second-order Hessian information in optimization theory are addressed in section 5. Both unconstrained and constrained minimization issues are briefly discussed. Finally the use of accurate Hessian/vector products to improve the performance of the truncated Newton method are presented along with the adjoint truncated-Newton method. A method for approximating the Hessian of the cost function with respect to the control variables proposed by Courtier et al. (1994), based on rank-*p* approximation and bearing similarity to approximation of the Hessian in quasi-Newton methods (see Davidon 1959, 1991), is presented in section 5d.

Section 6 is dedicated to methods of obtaining the second-order adjoint via AD technology, while issues of computational complexity of AD for the second-order adjoint are presented in the appendix. Use of the Hessian of the cost functional to estimate error covariance matrices is briefly discussed in section 7. The use of Hessian singular vectors used for development of a simplified Kalman filter is addressed briefly in section 8.

Finally as a numerical illustration we present in section 9 the application of the second-order adjoint of a limited area model of the shallow-water equations to obtain an accurate Hessian/vector product compared to an approximate Hessian vector product obtained by finite differences. Automatic differentiation is implemented using the adjoint model compiler TAMC. The Hessian/vector information is used in a truncated-Newton minimization algorithm of the cost functional with respect to the initial conditions taken as the control variables and its impact versus the Hessian/vector product obtained via finite differences is assessed. The numerical results obtained verify the theoretically derived computational cost of obtaining the second-order adjoint via automatic differentiation. The Arnoldi package (ARPACK) was then used in conjunction with the second-order adjoint to gain information about the spectrum of the Hessian of the cost function. The unified notation of Ide et al. (1997) for data assimilation will be used.

Summary and conclusions are finally presented in section 10.

2. Computing the second-order information

In this section we will deal with deterministic models while the case of stochastic modeling will be discussed later in this manuscript.

In general we will assume that the model has the general form

$$F(\mathbf{X}, \mathbf{U}) = 0, \tag{1}$$

where **X**, the state variable, describes the state of the environment and **U** is the input of the model, that is, an initial condition that has to be provided to the model to obtain from Eq. (1) a unique solution **X**(**U**). We will assume that **X** and **U** belong to a space equipped with an inner product.

The closure of the model is obtained through a variational principle that can be considered as the minimization of some functional:

$$J(\mathbf{X}, \mathbf{U}). \tag{2}$$

For instance, in the case of variational data assimilation, J may be viewed as representing the cost function measuring the discrepancy between the observation and the solution associated with the value **U** of the input parameter. Therefore the optimal input for the model will minimize J.

a. First-order necessary conditions

If the optimal **U** minimizes J, then it satisfies the Euler equations given by

$$\nabla J(\mathbf{U}) = \mathbf{0}, \tag{3}$$

where ∇J is the gradient of J with respect to control variables.

The gradient of J is obtained in the following way:

We compute the Gâteaux (directional) derivative of the model and of F in some direction **u**. We may write

$$\frac{\partial F}{\partial \mathbf{X}} \times \hat{\mathbf{X}} + \frac{\partial F}{\partial \mathbf{U}} \times \mathbf{u} = \mathbf{0}, \tag{4}$$

where (ˆ) stands for the Gâteaux derivative. Let Z be an application from $\mathbb{R}^n$ into $\mathbb{R}^n$ with variable **U**. We define the Gâteaux derivative of Z in the direction **u** when this limit exists. For a generic function Z it is given by

$$\hat{Z}(\mathbf{U}, \mathbf{u}) = \lim_{\alpha \to 0} \frac{Z(\mathbf{U} + \alpha \mathbf{u}) - Z(\mathbf{U})}{\alpha}. \tag{5}$$

If $\hat{Z}(\mathbf{U}, \mathbf{u})$ is linear in **u**, we can write

$$\hat{Z}(\mathbf{U}, \mathbf{u}) = \langle \nabla Z(\mathbf{U}), \mathbf{u} \rangle,$$

where ∇Z is the gradient of Z with respect to **U**, and $\langle \cdot , \cdot \rangle$ stands for the inner product. The Gâteaux derivative is also called a directional derivative.

Here, $\partial F/\partial \mathbf{X}$ (or $\partial F/\partial \mathbf{U}$) is the Jacobian of F with respect to **X** (or **U**) and

$$\hat{J}(\mathbf{X}, \mathbf{U}, \mathbf{u}) = \left\langle \frac{\partial J}{\partial \mathbf{X}}, \hat{\mathbf{X}} \right\rangle + \left\langle \frac{\partial J}{\partial \mathbf{U}}, \mathbf{u} \right\rangle. \tag{6}$$

The gradient of J is obtained by exhibiting the linear dependence of $\hat{J}$ with respect to **u**. This is done by introducing the adjoint variable **P** (to be defined later according to convenience).

Taking the inner product between (4) and **P** yields

$$\left\langle \frac{\partial F}{\partial \mathbf{X}} \times \hat{\mathbf{X}}, \mathbf{P} \right\rangle + \left\langle \frac{\partial F}{\partial \mathbf{U}} \times \mathbf{u}, \mathbf{P} \right\rangle = 0 \tag{7}$$

$$\left\langle \left(\frac{\partial F}{\partial \mathbf{X}}\right)^{\mathrm{T}} \times \mathbf{P}, \hat{\mathbf{X}} \right\rangle + \left\langle \left(\frac{\partial F}{\partial \mathbf{U}}\right)^{\mathrm{T}} \times \mathbf{P}, \mathbf{u} \right\rangle = 0. \tag{8}$$

Therefore using (6), if **P** is defined as the solution of the adjoint model,

$$\left(\frac{\partial F}{\partial \mathbf{X}}\right)^{\mathrm{T}} \times \mathbf{P} = \frac{\partial J}{\partial \mathbf{X}}, \tag{9}$$

then we obtain

$$\nabla J(\mathbf{U}) = -\left(\frac{\partial F}{\partial \mathbf{U}}\right)^{\mathrm{T}} \times \mathbf{P} + \frac{\partial J}{\partial \mathbf{U}}. \tag{10}$$

Therefore the gradient is computed by solving Eq. (9) to obtain **P**, then by applying Eq. (10).

b. Second-order adjoint

To obtain second-order information, we seek the product of the Hessian matrix **G**(**U**) of J with some vector **u**. As before we apply a perturbation **u** to Eqs. (1) and (9), and from Eqs. (9) and (10) we obtain

$$\left(\frac{\partial^2 F}{\partial \mathbf{X}^2} \times \hat{\mathbf{X}} + \frac{\partial^2 F}{\partial \mathbf{X} \partial \mathbf{U}} \times \mathbf{u}\right)^{\mathrm{T}} \times \mathbf{P} + \left(\frac{\partial F}{\partial \mathbf{X}}\right)^{\mathrm{T}} \times \hat{\mathbf{P}}$$
$$= \frac{\partial^2 J}{\partial \mathbf{X}^2} \times \hat{\mathbf{X}} + \frac{\partial^2 J}{\partial \mathbf{X} \partial \mathbf{U}} \times \mathbf{u} \quad \text{and} \tag{11}$$

$$\begin{aligned} \widehat{\nabla J(\mathbf{U})} &= \mathbf{G}(\mathbf{U}) \times \mathbf{u} \\ &= -\left(\frac{\partial^2 F}{\partial \mathbf{U}^2} \times \mathbf{u} + \frac{\partial^2 F}{\partial \mathbf{U} \partial \mathbf{X}} \times \hat{\mathbf{X}}\right)^{\mathrm{T}} \times \mathbf{P} \\ &\quad - \left(\frac{\partial F}{\partial \mathbf{U}}\right)^{\mathrm{T}} \times \hat{\mathbf{P}} + \frac{\partial^2 J}{\partial \mathbf{U}^2} \times \mathbf{u} \\ &\quad + \frac{\partial^2 J}{\partial \mathbf{U} \partial \mathbf{X}} \times \hat{\mathbf{X}}. \end{aligned} \tag{12}$$

We introduce here **Q** and **R**, two additional variables. To eliminate $\hat{\mathbf{X}}$ and **P**, we will take the inner product of Eqs. (4) and (11) with **Q** and **R**, respectively; then we add the results. We then obtain

$$\left\langle \hat{\mathbf{X}}, \left(\frac{\partial F}{\partial \mathbf{X}}\right)^{\mathrm{T}} \times \mathbf{Q} \right\rangle + \left\langle \mathbf{u}, \left(\frac{\partial F}{\partial \mathbf{U}}\right)^{\mathrm{T}} \times \mathbf{Q} \right\rangle + \left\langle \mathbf{P}, \frac{\partial^2 F}{\partial \mathbf{X}^2} \times \hat{\mathbf{X}} \times \mathbf{R} \right\rangle + \left\langle \mathbf{P}, \left(\frac{\partial^2 F}{\partial \mathbf{X}\partial \mathbf{U}}\right) \times \mathbf{u} \times \mathbf{R} \right\rangle + \left\langle \hat{\mathbf{P}}, \left(\frac{\partial F}{\partial \mathbf{X}}\right) \times \mathbf{R} \right\rangle = \left\langle \hat{\mathbf{X}}, \left(\frac{\partial^2 J}{\partial \mathbf{X}^2}\right)^{\mathrm{T}} \times \mathbf{R} \right\rangle + \left\langle \mathbf{u}, \left(\frac{\partial^2 J}{\partial \mathbf{X}\partial \mathbf{U}}\right)^{\mathrm{T}} \times \mathbf{R} \right\rangle. \quad (13)$$

Let us take the inner product of Eq. (12) with **u**; then we may write

$$\langle \mathbf{G}(\mathbf{U}) \times \mathbf{u}, \mathbf{u} \rangle = \left\langle -\left(\frac{\partial^2 F}{\partial \mathbf{U}^2} \times \mathbf{u} + \frac{\partial^2 F}{\partial \mathbf{U}\partial \mathbf{X}} \times \hat{\mathbf{X}}\right)^{\mathrm{T}} \times \mathbf{P}, \mathbf{u} \right\rangle + \left\langle \hat{\mathbf{P}}, \left(-\frac{\partial F}{\partial \mathbf{U}}\right) \times \mathbf{u} \right\rangle + \left\langle \frac{\partial^2 J}{\partial \mathbf{U}^2} \times \mathbf{u}, \mathbf{u} \right\rangle + \left\langle \hat{\mathbf{X}}, \left(\frac{\partial^2 J}{\partial \mathbf{U}\partial \mathbf{X}}\right)^{\mathrm{T}} \times \mathbf{u} \right\rangle. \quad (14)$$

From (13) we get

$$\left\langle \hat{\mathbf{X}}, \left(\frac{\partial F}{\partial \mathbf{X}}\right)^{\mathrm{T}} \times \mathbf{Q} + \left(\frac{\partial^2 F}{\partial \mathbf{X}^2} \times \mathbf{P}\right) \times \mathbf{R} - \frac{\partial^2 J}{\partial \mathbf{X}^2} \times \mathbf{R} \right\rangle + \left\langle \hat{\mathbf{P}}, \frac{\partial F}{\partial \mathbf{X}} \times \mathbf{R} \right\rangle = \left\langle \mathbf{u}, -\left(\frac{\partial F}{\partial \mathbf{U}}\right)^{\mathrm{T}} \times \mathbf{Q} - \left(\frac{\partial^2 F}{\partial \mathbf{X}\partial \mathbf{U}} \times \mathbf{P}\right)^{\mathrm{T}} \times \mathbf{R} + \frac{\partial^2 J}{\partial \mathbf{X}\partial \mathbf{U}} \times \mathbf{R} \right\rangle. \quad (15)$$

Therefore if **Q** and **R** are defined as being the solution of

$$\left(\frac{\partial F}{\partial \mathbf{X}}\right)^{\mathrm{T}} \times \mathbf{Q} + \frac{\partial^2 F}{\partial \mathbf{X}^2} \times \mathbf{P} \times \mathbf{R} - \left(\frac{\partial^2 J}{\partial \mathbf{X}^2}\right)^{\mathrm{T}} \times \mathbf{R} = \left(\frac{\partial^2 J}{\partial \mathbf{X}\partial \mathbf{U}}\right)^{\mathrm{T}} \times \mathbf{u} - \left(\frac{\partial^2 F}{\partial \mathbf{U}\partial \mathbf{X}} \cdot \mathbf{u}\right) \times \mathbf{P} \quad (16)$$

$$\left(\frac{\partial F}{\partial \mathbf{X}}\right) \times \mathbf{R} = -\frac{\partial F}{\partial \mathbf{U}} \times \mathbf{u}, \quad (17)$$

then we obtain

$$\mathbf{G}(\mathbf{U}) \times \mathbf{u} = -\left(\frac{\partial^2 F}{\partial \mathbf{U}^2} \times \mathbf{u}\right) \times \mathbf{P} + \frac{\partial^2 J}{\partial \mathbf{U}^2} \times \mathbf{u} - \left(\frac{\partial F}{\partial \mathbf{U}}\right)^{\mathrm{T}} \times \mathbf{Q} - \left(\frac{\partial^2 F}{\partial \mathbf{X}\partial \mathbf{U}} \times \mathbf{P}\right) \times \mathbf{R} + \frac{\partial^2 J}{\partial \mathbf{X}\partial \mathbf{U}} \times \mathbf{R}. \quad (18)$$

For Eqs. (14)–(15) we took into account the symmetry of the second derivative, for example,

$$\frac{\partial^2 J}{\partial \mathbf{X}^2} = \left(\frac{\partial^2 J}{\partial \mathbf{X}^2}\right)^{\mathrm{T}}$$

leading to some simplifications.

The system (16)–(17) will be called the second-order adjoint. Therefore we can obtain the product of the Hessian by a vector **u** by (i) solving the system (16)–(17) and (ii) applying formula (18).

c. Remarks

The system (16)–(17), which has to be solved to obtain the Hessian/vector product, can be derived from the Gâteaux derivative (4), which is the same as (17). In the literature, the system (16)–(17) is often called the tangent linear model, this denomination being rather inappropriate because it implies the issue of linearization and the subsequent notion of range of validity, which is not relevant in the case of a derivative.

In the case of an *N*-finite dimensional space the Hessian can be fully computed after *N* integrations of vector components e_i of the canonical base.

Equation (16) differs from the adjoint model by the forcing terms, which will depend on **u** and **R**.

The system (16)–(18) will yield the exact value of the Hessian/vector product. An approximation could be obtained by standard finite differences, that is,

$$\mathbf{G}(\mathbf{U}) \times \mathbf{u} \simeq \frac{1}{\alpha}[\nabla J(\mathbf{U} + \alpha \mathbf{u}) - \nabla J(\mathbf{U})], \quad (19)$$

where α is the finite-difference interval that has to be judicially chosen. In the incremental three- and four-dimensional variational data assimilation (3D-/4DVAR) approach the Hessian can readily be obtained by differencing two gradients.

However several integrations of the model and of its adjoint model will be necessary in this case to determine the range of validity of the finite-difference approximation (Wang 1995 and references therein).

d. Time-dependent model

In the case of variational data assimilation the model *F* is a differential system on the time interval [0, *T*]. The evolution of **X** between 0 and *T* is governed by the differential system:

$$\frac{\partial \mathbf{X}}{\partial t} = F(\mathbf{X}) + \mathsf{B} \times \mathbf{V}. \tag{20}$$

Here $\mathbf{X}$ belongs to the Hilbert space $\mathbb{H}$, which is a subspace of the n-dimensional Cartesian product space $[\mathbb{C}(0, T)]^n$.

The input variable is often the initial condition:

$$\mathbf{X}(0) = \mathbf{U} \in \mathbb{R}^n. \tag{21}$$

In this system F is a nonlinear operator that describes the dynamics of the model; $\mathbf{V} \in V \subset [\mathbb{C}(0, T)]^m$ is a term used to represent the uncertainties of the model, which we assume to be linearly coupled through the (n, m)-dimensional matrix B; $\mathbf{U}$ is the initial condition; and the criteria J is the discrepancy between the solution of (20)–(21) and observations:

$$J(\mathbf{U}, \mathbf{V}) = \frac{1}{2}\int_0^T \|\mathsf{H}\mathbf{X} - \mathbf{X}_{\text{obs}}\|^2 \, dt, \tag{22}$$

where H is the observation matrix, that is, a linear operator mapping $\mathbf{X}$ into $\mathbb{O}_{\text{obs}}$. The problem consists in determining $\mathbf{U}$ and $\mathbf{V}$ that minimize J.

A perturbation $\mathbf{v}$ on $\mathbf{V}$ and $\mathbf{u}$ on $\mathbf{U}$ gives $\hat{\mathbf{X}}$ and $\hat{J}$ the Gâteaux derivatives of $\mathbf{X}$ and J as a solution of

$$\frac{d\hat{\mathbf{X}}}{dt} = \frac{\partial F}{\partial \mathbf{X}} \times \hat{\mathbf{X}} + \mathsf{B} \times \mathbf{v} \tag{23}$$

$$\hat{\mathbf{X}}(0) = \mathbf{u} \tag{24}$$

$$\hat{J}(\mathbf{U}, \mathbf{V}, \mathbf{u}, \mathbf{v}) = \int_0^T \langle \mathsf{H}\mathbf{X} - \mathbf{X}_{\text{obs}}, \mathsf{H}\hat{\mathbf{X}} \rangle \, dt. \tag{25}$$

Let us introduce $\mathbf{P}$, the adjoint variable; we take the product of (23) with $\mathbf{P}$ after a summation on the interval $[0, T]$ and an integration by parts followed by identification of linearities with respect to $\mathbf{u}$ and $\mathbf{v}$ in (25); next we conclude that if $\mathbf{P}$ is defined as the solution of the adjoint model

$$\frac{d\mathbf{P}}{dt} + \left(\frac{\partial F}{\partial \mathbf{X}}\right)^{\mathrm{T}} \times \mathbf{P} = \mathsf{H}^{\mathrm{T}}(\mathsf{H}\mathbf{X} - \mathbf{X}_{\text{obs}}) \tag{26}$$

$$\mathbf{P}(T) = \mathbf{0}, \tag{27}$$

then the components of the gradient ∇J with respect to $\mathbf{U}$ and $\mathbf{V}$ are

$$\nabla J_{\mathbf{U}} = -\mathbf{P}(0) \tag{28}$$

$$\nabla J_{\mathbf{V}} = -\mathsf{B}^{\mathrm{T}}\mathbf{P}. \tag{29}$$

Because $\mathbf{V}$ is time dependent, its associated adjoint variable $\mathbf{Q}$ will be also time dependent. Let us remark that the gradient of J with respect to $\mathbf{V}$ will depend on time, which is not surprising since J also depends on time. From a computational point of view the discretization of $\mathbf{V}$ will have to be carried out in such a way that the discretized variable remains in a space of ‘‘reasonable’’ dimension.

The second derivative will be derived after a perturbation h on the control variables $\mathbf{U}$ and $\mathbf{V}$:

$$\mathsf{h} = \begin{pmatrix} \mathbf{h}_{\mathbf{U}} \\ \mathbf{h}_{\mathbf{V}} \end{pmatrix}. \tag{30}$$

The Gâteaux derivatives $\hat{\mathbf{X}}$, $\mathbf{P}$ of $\mathbf{X}$, and $\mathbf{P}$ in the direction of h are obtained as the solution of the coupled system:

$$\frac{d\hat{\mathbf{X}}}{dt} = \frac{\partial F}{\partial \mathbf{X}}\hat{\mathbf{X}} + \mathsf{B}\mathbf{h}_{\mathbf{V}} \tag{31}$$

$$\hat{\mathbf{X}}(0) = \mathbf{h}_{\mathbf{U}} \tag{32}$$

$$\frac{d\hat{\mathbf{P}}}{dt} + \left(\frac{\partial^2 F}{\partial \mathbf{X}^2} \times \hat{\mathbf{X}}\right)^{\mathrm{T}} \times \mathbf{P} + \left(\frac{\partial F}{\partial \mathbf{X}}\right)^{\mathrm{T}} \times \mathbf{P} = \mathbf{H}^{\mathrm{T}}\mathbf{H}\hat{\mathbf{X}} \tag{33}$$

$$\nabla J_{\mathbf{U}} = -\hat{\mathbf{P}}(0) \tag{34}$$

$$\nabla J_{\mathbf{V}} = -\mathsf{B}^{\mathrm{T}}\hat{\mathbf{P}}. \tag{35}$$

We then introduce $\mathbf{Q}$ and $\mathbf{R}$, second-order adjoint variables. They will be defined later for ease in use of presentations.

Taking the inner product of (31) with $\mathbf{Q}$ and of (33) with $\mathbf{R}$, integrating from 0 to T, then adding the resulting equations, we may write

$$\int_0^T \left[\left\langle \frac{d\hat{\mathbf{X}}}{dt}, \mathbf{Q}\right\rangle - \left\langle \frac{\partial F}{\partial \mathbf{X}} \times \hat{\mathbf{X}}, \mathbf{Q}\right\rangle - \langle \mathsf{B}\mathbf{h}_{\mathbf{V}}, \mathbf{Q}\rangle + \left\langle \frac{d\hat{\mathbf{P}}}{dt}, \mathbf{R}\right\rangle + \left\langle \left(\frac{\partial^2 F}{\partial \mathbf{X}^2} \times \hat{\mathbf{X}}\right)^{\mathrm{T}} \times \mathbf{P}, \mathbf{R}\right\rangle + \left\langle \left(\frac{\partial F}{\partial \mathbf{X}}\right)^{\mathrm{T}} \times \hat{\mathbf{P}}, \mathbf{R}\right\rangle - \langle \mathsf{H}^{\mathrm{T}}\mathsf{H}\hat{\mathbf{X}}, \mathbf{R}\rangle\right] dt = 0, \tag{36}$$

The terms in $\hat{\mathbf{P}}$ and $\hat{\mathbf{X}}$ are collected and after integration by parts and some additional transformations we obtain

$$\int_0^T \left\langle \hat{\mathbf{X}}, -\frac{d\mathbf{Q}}{dt} - \left(\frac{\partial F}{\partial \mathbf{X}}\right)^{\mathrm{T}} \times \mathbf{Q} + \left[\frac{\partial^2 F}{\partial \mathbf{X}^2} \times \mathbf{P}\right]^{\mathrm{T}} \times \mathbf{R} - \mathsf{H}^{\mathrm{T}}\mathsf{H}\mathbf{R}\right\rangle dt + \int_0^T \left\langle \hat{\mathbf{P}}, -\frac{d\mathbf{R}}{dt} + \left(\frac{\partial F}{\partial \mathbf{X}}\right) \times \mathbf{R}\right\rangle dt$$

$$- \int_0^T \langle \mathbf{h}_{\mathbf{V}}, \mathsf{B}^{\mathrm{T}} \times \mathbf{Q}\rangle \, dt + \langle \hat{\mathbf{X}}(T), \mathbf{Q}(T)\rangle - \langle \hat{\mathbf{X}}(0), \mathbf{Q}(0)\rangle + \langle \hat{\mathbf{P}}(T), \mathbf{R}(T)\rangle - \langle \hat{\mathbf{P}}(0), \mathbf{R}(0)\rangle = 0. \tag{37}$$

Let G be the Hessian matrix of the cost J. We then have

$$\mathsf{G} = \begin{pmatrix} \mathsf{G}_{UU} & \mathsf{G}_{UV} \\ \mathsf{G}_{VU} & \mathsf{G}_{VV} \end{pmatrix}. \tag{38}$$

Therefore if we define the second-order adjoint as being the solution of

$$\frac{d\mathbf{Q}}{dt} + \left(\frac{\partial F}{\partial \mathbf{X}}\right)^{\mathrm{T}} \times \mathbf{Q} = \left[\frac{\partial^2 F}{\partial \mathbf{X}^2} \times \mathbf{P}\right]^{\mathrm{T}} \times \mathbf{R} - \mathsf{H}^{\mathrm{T}}\mathsf{H}\mathbf{R} \tag{39}$$

$$\frac{d\mathbf{R}}{dt} = \left[\frac{\partial F}{\partial \mathbf{X}}\right] \times \mathbf{R} \quad \text{and} \tag{40}$$

$$\mathbf{Q}(T) = \mathbf{0} \tag{41}$$

$$\mathbf{R}(0) = \mathbf{h}_{\mathbf{U}}, \tag{42}$$

then it remains

$$-\langle \mathbf{h}_{\mathbf{U}}, \mathbf{Q}(0)\rangle = \langle \hat{\mathbf{P}}(0), \mathbf{R}(0)\rangle \tag{43}$$

$$\hat{\mathbf{P}}(0) = -\mathbf{Q}(0). \tag{44}$$

We would like to point out that Eq. (44) follows directly from Eq. (43) by using Eq. (42). The product of the Hessian by a vector is obtained exactly by a direct integration of (40) and (42) followed by a backward integration in time of (39) and (41).

One can obtain G by n integrations of the differential system:

$$\frac{d\mathbf{Q}}{dt} + \left(\frac{\partial F}{\partial \mathbf{X}}\right)^{\mathrm{T}} \times \mathbf{Q} = \left(\frac{\partial^2 F}{\partial \mathbf{X}^2} \times \mathbf{P}\right)^{\mathrm{T}} \times \mathbf{R} - \mathsf{H}^{\mathrm{T}}\mathsf{H}\mathbf{R} \tag{45}$$

$$\frac{d\mathbf{R}}{dt} = \left(\frac{\partial F}{\partial \mathbf{X}}\right) \times \mathbf{R}, \tag{46}$$

with the conditions

$$\mathbf{Q}(T) = \mathbf{0} \tag{47}$$

$$\mathbf{R}(0) = \mathbf{e}_i \tag{48}$$

where $\mathbf{e}_i$ are the n vectors of the canonical base of $\mathbb{R}^n$ thus obtaining

$$\mathsf{G}_{UU}\mathbf{e}_i = \mathbf{Q}(0) \tag{49}$$

$$\mathsf{G}_{UV}\mathbf{e}_i = \mathsf{B}^{\mathrm{T}} \times \mathbf{Q}. \tag{50}$$

One then integrates m times the differential system:

$$\frac{d\mathbf{Q}}{dt} + \left(\frac{\partial F}{\partial \mathbf{X}}\right)^{\mathrm{T}} \times \mathbf{Q} = \left(\frac{\partial^2 F}{\partial \mathbf{X}^2} \times \mathbf{P}\right)^{\mathrm{T}} \times \mathbf{R} - \mathsf{H}^{\mathrm{T}}\mathsf{H}\mathbf{R} \tag{51}$$

$$\frac{d\mathbf{R}}{dt} - \left(\frac{\partial F}{\partial \mathbf{X}}\right) \times \mathbf{R} = \mathbf{f}_j, \tag{52}$$

with terminal and, respectively, initial conditions

$$\mathbf{Q}(T) = \mathbf{0} \tag{53}$$

$$\mathbf{R}(0) = \mathbf{0}, \tag{54}$$

where $\mathbf{f}_j$ are the m canonical base vectors of $\mathbb{R}^m$ obtaining

$$\mathsf{G}_{VV} \times \mathbf{f}_j = \mathsf{B}^{\mathrm{T}} \times \mathbf{Q}. \tag{55}$$

The system defined by these equations is the second-order adjoint model. The Hessian matrix is obtained via $n + m$ integrations of the second-order adjoint. The second-order adjoint is easily obtained from the first-order adjoint; differing from it only by some forcing terms, in particular the second-order term. The second equation is that of the linearized model (the tangent linear model).

One can also obtain the product of a vector of the control space, times the Hessian at the cost of a single integration of the second-order adjoint.

e. Example: The shallow-water equations

The shallow-water equations (SWEs) represent the flow of an incompressible fluid whose depth is small with respect to the horizontal dimension.

The SWEs can be written in a Cartesian system:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} - fv + \frac{\partial \phi}{\partial x} = 0 \tag{56}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + fu + \frac{\partial \phi}{\partial y} = 0 \tag{57}$$

$$\frac{\partial \phi}{\partial t} + \frac{\partial u\phi}{\partial x} + \frac{\partial v\phi}{\partial y} = 0. \tag{58}$$

In this system of equations $\mathbf{X} = (u, v, \phi)^{\mathrm{T}}$ is the state variable, u and v are the components of the horizontal velocity, ϕ is the geopotential, and f the Coriolis parameter. We aim to present this example in order to provide a didactic setup, thus we will make the strongest simplifications.

a) We neglect the model error, which following the previous notation, implies $\mathsf{B} \equiv 0$. We only control the initial conditions.
b) We impose periodic boundary conditions.
c) The observations are assumed continuous in both space and time, which is tantamount to assuming $\mathsf{H} \equiv \mathsf{I}$, where I is the identity operator. Let $\mathbf{U}_0 = (u_0, v_0, \phi_0)^{\mathrm{T}}$, that is, the initial condition, then the cost function assumes the form

$$J(\mathbf{U}_0) = \frac{1}{2}\int_0^T [(u - u_{\mathrm{obs}})^2 + (v - v_{\mathrm{obs}})^2 + \gamma(\phi - \phi_{\mathrm{obs}})^2]\, dt, \tag{59}$$

where γ is a nonunit weighting term.

We derive directly the tangent linear model (TLM). The barred variables $\overline{\mathbf{X}} = (\overline{u}, \overline{v}, \overline{\phi})^{\mathrm{T}}$ are the directional derivatives in the direction of the perturbation $\mathbf{h} = (\mathbf{h}_u, \mathbf{h}_\upsilon, \mathbf{h}_\phi)^{\mathrm{T}}$ applied to the initial condition and we obtain

$$\frac{\partial \overline{u}}{\partial t} + u\frac{\partial \overline{u}}{\partial x} + \overline{u}\frac{\partial u}{\partial x} + \upsilon\frac{\partial \overline{u}}{\partial y} + \overline{\upsilon}\frac{\partial u}{\partial y} - f\,\overline{\upsilon} + \frac{\partial \overline{\phi}}{\partial x} = 0 \qquad (60)$$

$$\frac{\partial \overline{\upsilon}}{\partial t} + u\frac{\partial \overline{\upsilon}}{\partial x} + \overline{u}\frac{\partial \upsilon}{\partial x} + \upsilon\frac{\partial \overline{u}}{\partial y} + \overline{\upsilon}\frac{\partial \upsilon}{\partial y} + f\,\overline{u} + \frac{\partial \overline{\phi}}{\partial y} = 0 \qquad (61)$$

$$\frac{\partial \overline{\phi}}{\partial t} + \frac{\partial \overline{u}\phi}{\partial x} + \frac{\partial u\overline{\phi}}{\partial x} + \frac{\partial \overline{\upsilon}\phi}{\partial y} + \frac{\partial \upsilon\overline{\phi}}{\partial y} = 0. \qquad (62)$$

By transposing the TLM we obtain the adjoint model. Let $\mathbf{P} = (\tilde{u}, \tilde{\upsilon}, \tilde{\phi})^{\mathrm{T}}$ be the adjoint variable, then the adjoint model satisfies

$$\frac{\partial \tilde{u}}{\partial t} + u\frac{\partial \tilde{u}}{\partial x} + \upsilon\frac{\partial \tilde{\upsilon}}{\partial y} + \tilde{u}\frac{\partial \upsilon}{\partial y} - \tilde{\upsilon}\frac{\partial \upsilon}{\partial y} - f\tilde{\upsilon} + \phi\frac{\partial \tilde{\phi}}{\partial x}$$

$$= u_{\mathrm{obs}} - u \qquad (63)$$

$$\frac{\partial \tilde{\upsilon}}{\partial t} + \tilde{u}\frac{\partial u}{\partial y} - u\frac{\partial \tilde{\upsilon}}{\partial x} + \tilde{\upsilon}\frac{\partial u}{\partial x} + \upsilon\frac{\partial \tilde{\upsilon}}{\partial y} + f\tilde{u} + \phi\frac{\partial \tilde{\phi}}{\partial y}$$

$$= \upsilon_{\mathrm{obs}} - \upsilon \qquad (64)$$

$$\frac{\partial \tilde{\phi}}{\partial t} + \frac{\partial \tilde{u}}{\partial x} + \frac{\partial \tilde{\upsilon}}{\partial y} + u\frac{\partial \tilde{\phi}}{\partial x} + \upsilon\frac{\partial \tilde{\phi}}{\partial y} = \gamma(\phi_{\mathrm{obs}} - \phi). \qquad (65)$$

To obtain the second-order model we linearize the couple direct model and adjoint model; we then transpose and obtain the second-order adjoint variable $\mathbf{Q} = (\hat{u}, \hat{\upsilon}, \hat{\phi})^{\mathrm{T}}$ and the variable $\mathbf{R} = (\overline{u}, \overline{\upsilon}, \overline{\phi})^{\mathrm{T}}$ defined by the TLM:

$$\frac{\partial \hat{u}}{\partial t} + u\frac{\partial \hat{u}}{\partial x} + \upsilon\frac{\partial \hat{\upsilon}}{\partial y} + \hat{u}\frac{\partial \upsilon}{\partial y} - \hat{\upsilon}\frac{\partial \upsilon}{\partial y} - f\hat{\upsilon} + \phi\frac{\partial \hat{\phi}}{\partial x}$$

$$= \tilde{\upsilon}\frac{\partial \overline{\upsilon}}{\partial x} - \overline{u}\frac{\partial \tilde{u}}{\partial x} - \overline{\upsilon}\frac{\partial \overline{u}}{\partial y} + \tilde{u}\frac{\partial \overline{\upsilon}}{\partial y} - \overline{\phi}\frac{\partial \tilde{\phi}}{\partial x} - \overline{u} \qquad (66)$$

$$\frac{\partial \hat{\upsilon}}{\partial t} + \hat{u}\frac{\partial u}{\partial y} - u\frac{\partial \hat{\upsilon}}{\partial x} + \hat{\upsilon}\frac{\partial u}{\partial x} + \upsilon\frac{\partial \hat{\upsilon}}{\partial y} + f\hat{u} + \phi\frac{\partial \hat{\phi}}{\partial y}$$

$$= \tilde{u}\frac{\partial \overline{u}}{\partial x} - \overline{u}\frac{\partial \tilde{\upsilon}}{\partial x} - \tilde{\upsilon}\frac{\partial \overline{u}}{\partial y} + \overline{u}\frac{\partial \tilde{\upsilon}}{\partial y} - \overline{\phi}\frac{\partial \tilde{\phi}}{\partial y} - \overline{\upsilon} \qquad (67)$$

$$\frac{\partial \hat{\phi}}{\partial t} + \frac{\partial \hat{u}}{\partial x} + \frac{\partial \hat{\upsilon}}{\partial y} + u\frac{\partial \hat{\phi}}{\partial x} + \upsilon\frac{\partial \hat{\phi}}{\partial y}$$

$$= -\overline{u}\frac{\partial \tilde{\phi}}{\partial x} - \overline{\upsilon}\frac{\partial \tilde{\phi}}{\partial x} - \gamma\overline{\phi}. \qquad (68)$$

We see that formally the first- and second-order adjoint models differ only by second-order terms, which contain the adjoint variables.

The calculation of second-order derivatives requires the storage of the model trajectory, the tangent linear model, and the adjoint model.

3. Sensitivity analysis and second-order information

a. General sensitivity analysis

In general a model has the following kinds of variables:

(i) *State variable*: Z in a space $\mathbb{L}$, which describes the physical properties of the medium (velocity, pressure, temperature, . . .); Z depends on time and space.

(ii) *Input variable*: I in a space $\mathbb{L}$, which has to be provided to the model (e.g., initial or boundary conditions), most of the time these variables are not directly measured but they can be estimated through data assimilation.

(iii) *Parameters*: K represents empirical parameters (e.g., diffusivity) that most models contain, which have to be tuned to adjust the model to the observations.

From the mathematical point of view a model is written as

$$F'(\mathsf{Z}, \mathsf{I}, \mathsf{K}) = 0, \qquad (69)$$

where F' is some PDE operator (depending on time or being steady state) or its discrete counterpart. We assume that if I and K are given, then the model has a unique solution $\mathsf{Z}(\mathsf{I}, \mathsf{K})$.

In many applications sensitivity analysis is carried out; for instance if we consider some scalar quantity linked to a solution of the model, what will be its variation if there is some perturbation on the inputs of the model?

From the formal point of view a *sensitivity analysis* is defined by a so-called response function $\mathsf{G}\colon \mathbb{L} \to \mathbb{R}$, depending on the state variable $\mathbf{X}$ (and therefore indirectly depending on I and K). By definition the sensitivity of G with respect to K (respectively I) is the gradient of G with respect to K (respectively I):

$$\mathsf{S} = \frac{\partial \mathsf{G}}{\partial \mathsf{K}}.$$

There are two ways to estimate the sensitivity.

b. Sensitivity analysis via finite differences

Assume we are looking for the sensitivity with respect to K in a finite space of dimension N. Let k_i, $1 \leq i \leq N$ be the components of K, then

$$\left(\frac{\partial \mathsf{G}}{\partial \mathsf{K}}\right)_i = \frac{\partial \mathsf{G}}{\partial k_i}, \qquad 1 \leq i \leq N,$$

and estimation of $\partial\mathsf{G}/\partial k_i$ can be carried out by computing

$$\frac{\partial \mathsf{G}}{\partial k_i} \simeq \frac{\mathsf{G}(\mathsf{K} + \alpha k_i) - \mathsf{G}(\mathsf{K})}{\alpha}, \qquad (70)$$

where k_i are the canonical base vectors. This procedure for estimating a sensitivity is simple. Nevertheless it can be costly if N is large since it will require as many integrations of the model as N. Furthermore, (70) gives only one approximation of the gradient. The scalar α has to be chosen such that the model has a linear behavior with respect to α. The determination of α may require several integrations of the model for each value of k_i. The main advantage of this method is that it does not require important software development.

c. *Sensitivity via the adjoint*

Because the sensitivity is a gradient, the adjoint variable may be used to derive it. Let $\mathbf{i}$ and $\mathbf{k}$ be perturbations on $\mathbf{I}$ and $\mathbf{K}$. The derivation of (69) leads to

$$\frac{\partial F'}{\partial \mathbf{Z}} \times \hat{\mathbf{Z}} + \frac{\partial F'}{\partial \mathbf{I}} \times \mathbf{i} + \frac{\partial F'}{\partial \mathbf{K}} \times \mathbf{k} = \mathbf{0}, \qquad (71)$$

and for the response function,

$$\hat{\mathbf{G}}(\mathbf{Z}, \mathbf{I}, \mathbf{K}, \mathbf{i}, \mathbf{k}) = \left\langle \frac{\partial \mathbf{G}}{\partial \mathbf{Z}}, \hat{\mathbf{Z}} \right\rangle. \qquad (72)$$

The gradient will be obtained by exhibiting the linear dependence of $\hat{\mathbf{G}}$ with respect to $\mathbf{i}$ and $\mathbf{k}$. Let us introduce $\mathbf{\Pi}$, an adjoint variable of the same dimension as $\mathbf{Z}$.

Taking the inner product of (71) with $\mathbf{\Pi}$ gives

$$\left\langle \frac{\partial F'}{\partial \mathbf{Z}} \times \hat{\mathbf{Z}}, \mathbf{\Pi} \right\rangle + \left\langle \frac{\partial F'}{\partial \mathbf{I}} \times \mathbf{i}, \mathbf{\Pi} \right\rangle + \left\langle \frac{\partial F'}{\partial \mathbf{K}} \times \mathbf{k}, \mathbf{\Pi} \right\rangle = 0$$

$$\left\langle \hat{\mathbf{Z}}, \left(\frac{\partial F'}{\partial \mathbf{Z}}\right)^{\mathrm{T}} \times \mathbf{\Pi} \right\rangle = -\left\langle \mathbf{i}, \left(\frac{\partial F'}{\partial \mathbf{I}}\right)^{\mathrm{T}} \times \mathbf{\Pi} \right\rangle - \left\langle \mathbf{k}, \left(\frac{\partial F'}{\partial \mathbf{K}}\right)^{\mathrm{T}} \times \mathbf{\Pi} \right\rangle$$

By identification to (72), if the adjoint model is defined as the solution of

$$\left(\frac{\partial F'}{\partial \mathbf{Z}}\right)^{\mathrm{T}} \times \mathbf{\Pi} = \frac{\partial \mathbf{G}}{\partial \mathbf{Z}},$$

then the sensitivities $\mathbf{S_I}$ with respect to $\mathbf{I}$ (respectively $\mathbf{S_K}$, with respect to $\mathbf{K}$) are given by

$$\mathbf{S_I} = -\left(\frac{\partial F'}{\partial \mathbf{I}}\right)^{\mathrm{T}} \times \mathbf{\Pi}, \qquad \mathbf{S_K} = -\left(\frac{\partial F'}{\partial \mathbf{K}}\right)^{\mathrm{T}} \times \mathbf{\Pi}.$$

It is worth noting that the sensitivity is obtained only after one run of the adjoint model and the result is exact. The cost to be paid is in software development since an adjoint model has to be developed.

d. *Sensitivity analysis and data assimilation*

Previously we have assumed that the input parameters of the model are known. In fact they are indirectly derived from observations through a process of data assimilation. If a variational data assimilation procedure is carried out, and if $\mathbf{X}$ is a state variable, $\mathbf{I}$ the input, F the model, and $\mathbf{P}$ the adjoint variable, then $\mathbf{X}$ and $\mathbf{P}$ are solutions of the following optimality system (OS), $J(\mathbf{X}, \mathbf{I})$ being the cost function:

$$F(\mathbf{X}, \mathbf{I}) = \mathbf{0}$$

$$\left(\frac{\partial F}{\partial \mathbf{X}}\right)^{\mathrm{T}} \times \mathbf{P} - \frac{\partial J}{\partial \mathbf{X}} = \mathbf{0}$$

$$\left(\frac{\partial F}{\partial \mathbf{I}}\right)^{\mathrm{T}} \times \mathbf{P} - \frac{\partial J}{\partial \mathbf{I}} = \mathbf{0} \qquad (73)$$

For sensitivity studies in the presence of observations, with a given response function we have to consider the OS as a generalized model F' with a state variable $\mathbf{Z} = (\mathbf{X}, \mathbf{P})^{\mathrm{T}}$, and a general sensitivity analysis has to be applied to this general model. Therefore the adjoint of the optimally system has to be derived.

After a perturbation $\mathbf{i}$ on $\mathbf{I}$, we may write

$$\frac{\partial F}{\partial \mathbf{X}} \times \hat{\mathbf{X}} + \frac{\partial F}{\partial \mathbf{I}} \times \mathbf{i} = 0 \qquad (74)$$

$$\left(\frac{\partial^2 F}{\partial \mathbf{X}^2} \times \hat{\mathbf{X}} + \frac{\partial^2 F}{\partial \mathbf{X} \partial \mathbf{I}} \times \mathbf{i}\right)^{\mathrm{T}} \times \mathbf{P} + \left(\frac{\partial F}{\partial \mathbf{X}}\right)^{\mathrm{T}} \times \hat{\mathbf{P}} - \frac{\partial^2 J}{\partial \mathbf{X}^2} \times \hat{\mathbf{X}} - \frac{\partial^2 J}{\partial \mathbf{X} \partial \mathbf{I}} \times \mathbf{i} = 0 \qquad (75)$$

$$\left(\frac{\partial^2 F}{\partial \mathbf{X} \partial \mathbf{I}} \times \hat{\mathbf{X}} + \frac{\partial^2 F}{\partial \mathbf{I}^2} \times \mathbf{i}\right)^{\mathrm{T}} \times \mathbf{P} + \left(\frac{\partial F}{\partial \mathbf{I}}\right)^{\mathrm{T}} \times \hat{\mathbf{P}} - \frac{\partial^2 J}{\partial \mathbf{I}^2} \times \mathbf{i} - \frac{\partial^2 J}{\partial \mathbf{I} \partial \mathbf{X}} \times \hat{\mathbf{X}} = 0, \qquad (76)$$

and for the response function $\mathbf{G}$ defined earlier we get

$$\hat{\mathbf{G}}(\mathbf{X}, \mathbf{I}, \mathbf{i}) = \frac{\partial \mathbf{G}}{\partial \mathbf{X}} \times \hat{\mathbf{X}} + \frac{\partial \mathbf{G}}{\partial \mathbf{I}} \times \mathbf{i}. \qquad (77)$$

Here, $\hat{\mathbf{X}}$ and $\hat{\mathbf{P}}$ are the Gâteaux derivatives of $\mathbf{X}$ and $\mathbf{P}$ in the direction $\mathbf{i}$. After second-order adjoint variables $\mathbf{Q}$ and $\mathbf{R}$ are introduced, we take the inner product of (74) and (75) by $\mathbf{Q}$ and (76) by $\mathbf{R}$; then we add the three equations and we may write

$$\left\langle \hat{\mathbf{X}}, \left(\frac{\partial F}{\partial \mathbf{X}}\right)^{\mathrm{T}} \times \mathbf{Q} + \left(\frac{\partial^2 F}{\partial \mathbf{X}^2} \times \mathbf{P}\right)^{\mathrm{T}} \times \mathbf{Q} - \frac{\partial^2 J}{\partial \mathbf{X}^2} \times \mathbf{Q} - \frac{\partial^2 J}{\partial \mathbf{I} \partial \mathbf{X}} \times \mathbf{R} + \left(\frac{\partial^2 F}{\partial \mathbf{X} \partial \mathbf{I}} \times \mathbf{P}\right)^{\mathrm{T}} \times \mathbf{R} \right\rangle$$

$$+ \left\langle \hat{\mathbf{P}}, \left(\frac{\partial F}{\partial \mathbf{X}}\right) \times \mathbf{Q} + \left(\frac{\partial F}{\partial \mathbf{X}}\right) \times \mathbf{R} \right\rangle$$

$$+ \left\langle \mathbf{i}, \left(\frac{\partial F}{\partial \mathbf{I}}\right)^{\mathrm{T}} \times \mathbf{Q} + \left(\frac{\partial^2 F}{\partial \mathbf{X} \partial \mathbf{I}} \times \mathbf{P}\right)^{\mathrm{T}} \times \mathbf{Q} \right.$$

$$- \frac{\partial^2 J}{\partial \mathbf{X}\partial \mathbf{l}} \times \mathbf{Q} - \frac{\partial^2 J}{\partial \mathbf{l}^2} \times \mathbf{R}$$

$$+ \left(\frac{\partial^2 F}{\partial \mathbf{l}^2} \times \mathbf{P}\right)^{\mathrm{T}} \times \mathbf{R}\biggr\rangle = 0.$$

By identification in (77) it follows that if **Q** and **R** are defined as solution of

$$\left(\frac{\partial F}{\partial \mathbf{X}}\right)^{\mathrm{T}} \times \mathbf{Q} + \left(\frac{\partial^2 F}{\partial \mathbf{X}^2} \times \mathbf{P}\right)^{\mathrm{T}} \times \mathbf{Q} - \frac{\partial^2 J}{\partial \mathbf{X}^2} \times \mathbf{Q}$$

$$- \left(\frac{\partial^2 J}{\partial \mathbf{l}\partial \mathbf{X}}\right)^{\mathrm{T}} \times \mathbf{R} + \left(\frac{\partial^2 F}{\partial \mathbf{X}^2} \times \mathbf{P}\right)^{\mathrm{T}} \times \mathbf{R} = -\frac{\partial \mathbf{G}}{\partial \mathbf{X}}$$

$$\left(\frac{\partial F}{\partial \mathbf{X}}\right) \times \mathbf{Q} - \left(\frac{\partial F}{\partial \mathbf{l}}\right) \times \mathbf{R} = 0, \tag{78}$$

then we get the gradient of G with respect to **l**, that is the sensitivity

$$\mathbf{S} = \frac{d\mathbf{G}}{d\mathbf{l}} = -\left(\frac{\partial F}{\partial \mathbf{l}}\right)^{\mathrm{T}} \times \mathbf{Q} - \left(\frac{\partial^2 F}{\partial \mathbf{X}\partial \mathbf{l}} \times \mathbf{Q}\right)^{\mathrm{T}} \times \mathbf{P}$$

$$+ \left(\frac{\partial^2 J}{\partial \mathbf{X}\partial \mathbf{l}}\right) \times \mathbf{Q} - \frac{\partial^2 J}{\partial \mathbf{l}^2} \times \mathbf{R}$$

$$- \left(\frac{\partial^2 F}{\partial \mathbf{l}^2} \times \mathbf{R}\right)^{\mathrm{T}} \times \mathbf{P} + \frac{\partial \mathbf{G}}{\partial \mathbf{l}}. \tag{79}$$

Therefore the algorithm to get the sensitivity is as follows:

(i) solve the optimality system to obtain **X** and **P**,
(ii) solve the coupled system (78) to obtain **Q** and **R**, and
(iii) compute the sensitivity by (79).

The sensitivity in the presence of observations requires taking into account the second-order information. A very simple example given by Le Dimet et al. (1997) clearly shows the necessity of the introduction of this term.

4. Kalnay et al. (2000) quasi-inverse method and second-order information

The inverse 3DVAR method proposed by Kalnay et al. (2000) is introduced by considering a cost functional:

$$J = \frac{1}{2}(\mathbf{L}\delta\mathbf{x})^{\mathrm{T}}\mathbf{B}^{-1}(\mathbf{L}\delta\mathbf{x})$$

$$+ \frac{1}{2}[\mathbf{HL}\delta\mathbf{x} - \delta\mathbf{y}]^{\mathrm{T}}\mathbf{R}^{-1}[\mathbf{HL}\delta\mathbf{x} - \delta\mathbf{y}], \tag{80}$$

where $\delta\mathbf{x}$ is the difference between the analysis and the background at the beginning of the assimilation window, **L** and $\mathbf{L}^{\mathrm{T}}$ are the TLM and its adjoint, and **H** is the tangent linear version of the forward observation operator H. In addition, **B** is the forecast error covariance and **R** is the observational error covariance.

Taking the gradient of J with respect to the initial change $\delta\mathbf{x} = \mathbf{x}^a - \mathbf{x}^b$, where $\mathbf{x}^a$ and $\mathbf{x}^b$ are the analysis and first guess, respectively, we obtain

$$\nabla J = \mathbf{L}^{\mathrm{T}}[\mathbf{B}^{-1}\mathbf{L}\delta\mathbf{x} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{HL}\delta\mathbf{x} - \delta\mathbf{y})]. \tag{81}$$

In an adjoint 4DVAR an iterative minimization algorithm such as the quasi-Newton or conjugate gradient is employed to obtain the optimal perturbation:

$$\delta\mathbf{x}^i = \alpha_i \nabla J^{i-1}; \tag{82}$$

here i is the minimization counter, where α_i is the step size in the minimization algorithm.

One stops after a number of minimization iterations when $\|\nabla J\|$ is small enough to satisfy a convergence criterion.

In order to determine the optimal value of the step size, the minimization algorithm, say quasi-Newton, requires additional computations of the gradient ∇J^{i-1}, so that the number of direct and adjoint integrations required by adjoint 4DVAR can be larger than the number of minimization iterations (see Kalnay et al. (2000).

The inverse 3DVAR approach of Kalnay seeks to obtain directly the ‘‘perfect solution,’’ that is, the special $\delta\mathbf{x}$ that makes $\nabla J = 0$, provided $\delta\mathbf{x}$ is small.

Eliminating in (81) the adjoint operator one gets

$$\mathbf{L}\delta\mathbf{x} = (\mathbf{B}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\delta\mathbf{y}. \tag{83}$$

Since we have the quasi-inverse model obtained by integrating TLM backward, that is, a good approximation of $\mathbf{L}^{-1}$, we obtain

$$\delta\mathbf{x} = \mathbf{L}^{-1}(\mathbf{B}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\delta\mathbf{y}. \tag{84}$$

As shown by Kalnay et al. (2000) this is equivalent to the adjoint Newton algorithm used by Wang et al. (1997) except that it does not require a line minimization.

Wang et al. (1998) proposed an adjoint Newton algorithm that also required the backward integration of the tangent linear model and proposed a reformulation of the adjoint Newton when the TLM is not invertible. They did not explore this idea in depth. Physical processes are generally not parameterized in a reversible form in atmospheric models—a problem that can be only overcome to some extent by using simplified reversible physics. Also truly dissipative processes in atmospheric models are not reversible and as such will constitute a problem for the inverse 3DVAR. To show the link of inverse 3DVAR to second-order information we follow Kalnay et al. (2000) to show that inverse 3DVAR is equivalent to using a perfect Newton iterative method to solve the minimization problem at a given time level.

If we look for the minimum of the cost functional at

$\mathbf{x} + \delta\mathbf{x}$ given that our present estimate of the solution is $\mathbf{x}$, then expanding $\boldsymbol{\nabla} J(\mathbf{x} + \delta\mathbf{x})$ in a Taylor series to second term yields

$$\boldsymbol{\nabla} J(\mathbf{x} + \delta\mathbf{x}) = \boldsymbol{\nabla} J(\mathbf{x}) + \nabla^2 J(\mathbf{x})\delta\mathbf{x} = \mathbf{0}, \tag{85}$$

where $\nabla^2 J(\mathbf{x})$ is the Hessian matrix.

The Newton iteration is

$$\delta\mathbf{x} = -[\nabla^2 J(\mathbf{x})]^{-1}\boldsymbol{\nabla} J(\mathbf{x}). \tag{86}$$

For the cost function (80) the Hessian is given by

$$\boldsymbol{\nabla}^2 J(\mathbf{x}) = \mathbf{L}^{\mathrm{T}}(\mathbf{B}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H})\mathbf{L}. \tag{87}$$

A first iteration with the Newton minimization algorithm yields

$$\delta\mathbf{x}_1 = [\mathbf{L}^{\mathrm{T}}(\mathbf{B}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H})\mathbf{L}]^{-1}\mathbf{L}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\delta\mathbf{y}, \tag{88}$$

which is identical with the inverse 3DVAR solution.

Since cost functions used in 4DVAR are close to quadratic functions, one may view 3DVAR as a perfect preconditioner of a simplified 4DVAR problem.

In general, availability of second-order information allows powerful minimization algorithms to be used (Wang et al. 1995, 1997) even when the inverse 3DVAR is difficult to obtain as is the case with full physics models.

5. Hessian information in optimization theory

Hessian information is crucial in many aspects of both constrained and unconstrained minimization. All minimization methods start by assuming a quadratic model in the vicinity of the minimum of a multivariate minimization problem.

For the problem

$$\min_{\mathbf{X}\in\mathbb{R}^n} F(\mathbf{X}), \tag{89}$$

the necessary condition for $\mathbf{X}^*$ to be a stationary point is

$$\boldsymbol{\nabla} F(\mathbf{X}^*) = 0 \tag{90}$$

and in order to obtain sufficient conditions for the existence of the minimum of the multivariate unconstrained minimization problem, we must require that the Hessian at $\mathbf{X}^*$ is positive definite.

a. Spectrum of the Hessian and rate of convergence of unconstrained minimization

The eigenvalues of the Hessian matrix predict the behavior and convergence rate for unconstrained minimization. To show this, let us consider again the multivariate nonlinear function $F(\mathbf{X})$ of (89) and let $\mathbf{X}^*$ denote a local minimizer point that satisfies the condition

$$F(\mathbf{X}^*) \le F(\mathbf{X}) \tag{91}$$

for all $\mathbf{X}$ such that

$$\|\mathbf{X} - \mathbf{X}^*\| < \epsilon, \tag{92}$$

where ϵ is typically a small positive number whose value may depend on the value of $\mathbf{X}^*$. We define $F(\mathbf{X}^*)$ as an acceptable solution of (89).

If F is twice continuously differentiable, and $\mathbf{X}^*$ is an absolute minimum, then

$$\boldsymbol{\nabla} F(\mathbf{X}^*) = 0, \tag{93}$$

and the Hessian $\mathbf{G}(\mathbf{X}^*)$ of F at $\mathbf{X}^*$ is positive definite, that is,

$$\mathbf{p}^{\mathrm{T}}\mathbf{G}(\mathbf{X}^*)\mathbf{p} > 0, \quad \forall\ \mathbf{p} \in \mathbb{R}^n. \tag{94}$$

Let us expand F in a Taylor series about $\mathbf{X}^*$:

$$\begin{aligned} F(\mathbf{X}) &= F(\mathbf{X}^* + h\mathbf{p}) \\ &= F(\mathbf{X}^*) + \frac{1}{2}h^2\mathbf{p}^{\mathrm{T}}\mathbf{G}(\mathbf{X}^*)\mathbf{p} + O(h^3) \\ &\qquad [\text{since } \boldsymbol{\nabla} F(\mathbf{X}^*) = 0], \end{aligned} \tag{95}$$

where

$$\|\mathbf{p}\| = 1 \quad \text{and} \quad h = \|\mathbf{X} - \mathbf{X}^*\|. \tag{96}$$

For any acceptable solution we obtain

$$h^2 = \|\mathbf{X} - \mathbf{X}^*\|^2 \approx \frac{2\epsilon}{\mathbf{p}^{\mathrm{T}}\mathbf{G}(\mathbf{X}^*)\mathbf{p}} \tag{97}$$

such that the condition number of the Hessian $\mathbf{G}(\mathbf{X}^*)$ substantially affects the size of $\|\mathbf{X} - \mathbf{X}^*\|$, that is, the rate of convergence of the unconstrained minimization (Gill et al. 1981).

If $\mathbf{G}(\mathbf{X}^*)$ is ill-conditioned, the error in $\mathbf{X}$ will vary with the direction of the perturbation $\mathbf{p}$.

If $\mathbf{p}$ is a linear combination of eigenvectors of $\mathbf{G}(\mathbf{X}^*)$ corresponding to the largest eigenvalues, the size of $\|\mathbf{X} - \mathbf{X}^*\|$ will be relatively small, while if, on the other hand, $\mathbf{p}$ is a linear combination of eigenvectors of $\mathbf{G}(\mathbf{X}^*)$ corresponding to the smallest eigenvalues, the size of $\|\mathbf{X} - \mathbf{X}^*\|$ will be relatively large; that is, there will be slow convergence.

b. Role of the Hessian in constrained minimization

The Hessian information plays a very important role in constrained optimization as well. We shall deal here with optimality conditions where again Taylor series approximations are used to analyze the behavior of the objective function F and constraints h_i about a local constrained minimizer $\mathbf{X}^*$.

We shall consider first optimal conditions for linear equality constraints.

The problem is cast as

$$\min_{\mathbf{X}\in\mathbb{R}^n} F(\mathbf{X}), \quad \text{subject to } \mathbf{A}\mathbf{X} = \mathbf{b}, \tag{98}$$

where $\mathbf{A}$ is an $m \times n$ matrix, $m \le n$. We assume that F is twice continuously differentiable and that rows of $\mathbf{A}$ are independent, that is, $\mathbf{A}$ has full row rank. The feasible region consists of the set of points satisfying all the constrains.

Any problem with linear constraints $\mathbf{AX} = \mathbf{b}$ can be recast as an equivalent unconstrained problem. Assume we have a feasible point $\overline{\mathbf{X}}$; that is,

$$\mathbf{A}\overline{\mathbf{X}} = \mathbf{b}.$$

Then any other feasible point can be expressed as

$$\mathbf{X} = \overline{\mathbf{X}} + \mathbf{p},$$

where $\mathbf{p}$ is a feasible direction.

Any feasible direction must lie in the null space of $\mathbf{A}$, the set of vectors $\mathbf{p}$ satisfying

$$\mathbf{Ap} = \mathbf{0}.$$

Denoting the null space of $\mathbf{A}$ by $\mathbb{N}(\mathbf{A})$, the feasible region is given by

$$\{\mathbf{X}\colon \mathbf{X} = \overline{\mathbf{X}} + \mathbf{p}, \mathbf{p} \in \mathbb{N}(\mathbf{A})\}.$$

Let $\mathbf{Z}$ be the null space matrix for $\mathbf{A}$ of dimension $n \times r$ with $r \geq n - m$. Then the feasible region is given by

$$\{\mathbf{X}\colon \mathbf{X} = \overline{\mathbf{X}} + \mathbf{Zv}, \text{ where } \mathbf{v} \in \mathbb{R}^{\mathbf{r}}\} \quad \text{and} \quad \mathbf{AZ} = \mathbf{0}. \tag{99}$$

Then the constrained minimization problem in $\mathbf{X}$ is equivalent to the unconstrained problem:

$$\min_{\mathbf{v}\in\mathbb{R}^r} \Phi(\mathbf{v}) = F(\mathbf{X} + \mathbf{Zv}), \tag{100}$$

where $\mathbf{X}$ is a feasible point (Gill et al. 1981; Nash and Sofer 1996). The function Φ is the restriction of F onto the feasible region, called the reduced function. If $\mathbf{Z}$ is a basis matrix for the null space of $\mathbf{A}$, then Φ is a function of $n - m$ variables. The constrained problem has been transformed into an unconstrained problem with a reduced number of variables.

Optimality conditions involve derivatives of the reduced function. If $\mathbf{X} = \overline{\mathbf{X}} + \mathbf{Zv}$,

$$\nabla\phi(\mathbf{v}) = \mathbf{Z}^{\mathrm{T}}\nabla F(\overline{\mathbf{X}} + \mathbf{Zv}) = \mathbf{Z}^{\mathrm{T}}\nabla F(\mathbf{X})$$

$$\nabla^2\Phi(\mathbf{v}) = \mathbf{Z}^{\mathrm{T}}\nabla^2 F(\overline{\mathbf{X}} + \mathbf{Zv})\mathbf{Z} = \mathbf{Z}^{\mathrm{T}}\nabla^2 F(\mathbf{X})\mathbf{Z}. \tag{101}$$

The vector

$$\nabla\Phi(\mathbf{v}) = \mathbf{Z}^{\mathrm{T}}\nabla F(\mathbf{X}) \tag{102}$$

is called the reduced gradient of F at $\mathbf{X}$. Similarly the matrix

$$\nabla^2\Phi(\mathbf{v}) = \mathbf{Z}^{\mathrm{T}}\nabla^2 F(\mathbf{X})\mathbf{Z} \tag{103}$$

is called the reduced or projected Hessian matrix.

The reduced gradient and Hessian matrix are the gradient and Hessian of the restriction of F onto the feasible region evaluated at $\mathbf{X}$. If $\mathbf{X}^*$ is a local solution of the constrained problem, then

$$\mathbf{X}^* = \overline{\mathbf{X}} + \mathbf{Zv}^* \quad \text{for some } \mathbf{v}^* \tag{104}$$

and $\mathbf{v}^*$ is the local minimizer of Φ. Hence we can write

$$\nabla\Phi(\mathbf{v}^*) = 0 \tag{105}$$

and $\nabla^2\Phi(\mathbf{v}^*)$ is positive semidefinite.

Here we briefly present in the framework of optimality conditions for linear equality constraints the necessary conditions for a local minimizer. If $\mathbf{X}^*$ is a local minimizer of F and $\mathbf{Z}$ is the null-space matrix for $\mathbf{A}$, then

$$\mathbf{Z}^{\mathrm{T}}\nabla F(\mathbf{X}^*) = 0 \tag{106}$$

and $\mathbf{Z}^{\mathrm{T}}\nabla^2 F(\mathbf{X}^*)\mathbf{Z}$ is positive semidefinite. That is, the reduced gradient is zero and the reduced Hessian matrix is positive semidefinite. (The second-order derivative information is used to distinguish local minimizers from other stationary points.) The second-order condition is equivalent to the condition

$$\mathbf{v}^{\mathrm{T}}\mathbf{Z}^{\mathrm{T}}\nabla^2 F(\mathbf{X}^*)\mathbf{Zv} \geq 0 \quad \text{for all } \mathbf{v}. \tag{107}$$

Noting that $\mathbf{p} = \mathbf{Zv}$ is a null-space vector, we can rewrite (107) as

$$\mathbf{p}^{\mathrm{T}}\nabla^2 F(\mathbf{X}^*)\mathbf{p} \geq 0 \quad \text{for all } \mathbf{p} \in \mathbb{N}(\mathbf{A}); \tag{108}$$

that is, the Hessian matrix at $\mathbf{X}^*$ must be positive semidefinite on the null space of $\mathbf{A}$.

c. *Application of second-order-adjoint technique to obtain exact Hessian/vector product*

We will exemplify this application by considering a truncated-Newton algorithm for large-scale minimization.

DESCRIPTION OF TRUNCATED-NEWTON METHODS

Truncated-Newton methods are used to solve the problem

$$\min f(\mathbf{X}), \qquad \mathbf{X} = (x_1, x_2, \ldots, x_n)^{\mathrm{T}}. \tag{109}$$

They are a compromise on the Newton method [see also Gill and Murray (1979) and O'Leary (1983)], whereby they compute a search direction by finding an approximate solution to the Newton's equations

$$\nabla^2 f(\mathbf{X}_k)\mathbf{p} \approx -\nabla f(\mathbf{X}_k) \tag{110}$$

using a conjugate-gradient (CG) iterative method; we note here that Newton equations are a linear system of the form

$$\mathbf{AX} = \mathbf{b}, \tag{111}$$

where

$$\mathbf{A} = \nabla^2 f(\mathbf{X}_k), \qquad \mathbf{b} = -\nabla f(\mathbf{X}_k). \tag{112}$$

The conjugate gradient method is ''truncated'' before the exact solution to the Newton equations has been found. The CG method computes the search direction and requires storage of a few vectors.

The only obstacle for using minimization is the requirement that it computes Hessian matrix/vector products of the type

$$\mathbf{Av} = \nabla^2 f(\mathbf{X}_k)\mathbf{v} \tag{113}$$

for arbitrary vectors $\mathbf{v}$. One way to bypass the storage difficulty is to approximate the Hessian matrix/vector products using values of the gradient in such a way that the Hessian matrix need not be computed or stored. Using Taylor series

$$\nabla f(\mathbf{X}_k + h\mathbf{v}) = \nabla f(\mathbf{X}_k) + h\nabla^2 f(\mathbf{X}_k)\mathbf{v} + O(h^2), \qquad (114)$$

we obtain

$$\nabla^2 f(\mathbf{X}_k)\mathbf{v} = \lim_{h\to 0} \frac{\nabla f(\mathbf{X}_k + h\mathbf{v}) - \nabla f(\mathbf{X}_k)}{h}; \qquad (115)$$

that is, we approximate the matrix/vector product

$$\nabla^2 f(\mathbf{X}_k)\mathbf{v} \approx \frac{\nabla f(\mathbf{X}_k + h\mathbf{v}) - \nabla f(\mathbf{X}_k)}{h} \qquad (116)$$

for some small values of h.

The task of choosing an adequate h is an arduous one (see Nash and Sofer 1996, chapter 11.4.1 and references therein). For in-depth descriptions of the truncated-Newton (also referred to as the Hessian-free) method, see Nash (1984a–d, 1985) and Nash and Sofer (1989a,b), as well as Schlick and Fogelson (1992a,b), and early work by Dembo et al. (1982) and Dembo and Steihaug (1983). A comparison of limited memory quasi-Newton (see Liu and Nocedal 1989) and truncated-Newton methods is provided by Nash and Nocedal (1991), while a comprehensive well-written survey of truncated-Newton methods is presented in Nash (2000). A comparison between limited memory quasi-Newton and truncated-Newton methods applied to a meteorological problem is described in depth by Zou et al. (1990, 1993).

d. A method for estimating the Hessian matrix

The cost function measuring the misfit between the forecast model solution and available observations distributed in space and time may be expressed as

$$J[\mathbf{X}(t_0)] = \frac{1}{2}\sum_{r=0}^{R} \{\mathbf{B}[\mathbf{X}(t_r)] - \mathbf{X}^{\mathrm{obs}}(t_r)\}^{\mathrm{T}} \times \mathbf{W}(t_r)\{\mathbf{B}[\mathbf{X}(t_r)] - \mathbf{X}^{\mathrm{obs}}(t_r)\}. \qquad (117)$$

For the sake of simplicity we choose $\mathbf{R} = 1$, which yields

$$\begin{aligned} J[\mathbf{X}(t_0)] = {} & \frac{1}{2}\{\mathbf{B}[\mathbf{X}(t_0)] - \mathbf{X}^{\mathrm{obs}}(t_0)\}^{\mathrm{T}} \\ & \times \mathbf{W}(t_0)\{\mathbf{B}[\mathbf{X}(t_0)] - \mathbf{X}^{\mathrm{obs}}(t_0)\} \\ & + \frac{1}{2}(\mathbf{B}\{F\,[\mathbf{X}(t_0)]\} - \mathbf{X}^{\mathrm{obs}}(t_N))^{\mathrm{T}} \\ & \times \mathbf{W}(t_N)(\mathbf{B}(F\,[\mathbf{X}(t_0)] - \mathbf{X}^{\mathrm{obs}}(t_N)), \qquad (118) \end{aligned}$$

where $\mathbf{B}$ is an observation operator, $\mathbf{X}(t_r)$ the vector of model control variables, $\mathbf{X}^{\mathrm{obs}}(t_r)$ the vector of observational data at time $t = t_r$, $t = t_N$ is the final time of model integration, and $\mathbf{W}(t_r)$ is the inverse of the observation covariance matrix.

The operator of model integration from time $t = t_0$ to $t = t_N$ is

$$F = \prod_{n=1}^{N} F_n. \qquad (119)$$

At the minimum $\mathbf{X}_{\mathrm{min}}$, the gradient of the cost function ∇J vanishes.

If we introduce random variables $\boldsymbol{\eta}(t_0)$ and $\boldsymbol{\eta}(t_N)$, with zero expectations and whose covariances are the diagonal elements of $\mathbf{W}^{-1}(t_0)$ and $\mathbf{W}^{-1}(t_N)$, respectively, to the observations

$$\mathbf{X}_1^{\mathrm{obs}}(t_0) = \mathbf{X}^{\mathrm{obs}}(t_0) + \boldsymbol{\eta}(t_0) \qquad (120)$$

$$\mathbf{X}_1^{\mathrm{obs}}(t_N) = \mathbf{X}^{\mathrm{obs}}(t_N) + \boldsymbol{\eta}(t_N), \qquad (121)$$

then ∇J(at $\mathbf{X}^{*}_{\mathrm{min}}$) is a random variable and we obtain

$$E[\nabla J(\nabla J)^{\mathrm{T}}] = \mathbf{J}'', \qquad (122)$$

where $E[\cdot]$ is the mathematical expectation and $\mathbf{J}''$ is the Hessian matrix. We can see that we obtain an outer vector product expression, which is a rank-one matrix.

For each realization i of $\mathbf{X}_1^{\mathrm{obs}}(t_0)$ and $\mathbf{X}_1^{\mathrm{obs}}(t_N)$ we can calculate ∇J^i at $\mathbf{X}_{\mathrm{min}}$ and after p such realizations we obtain at most a rank p approximation of the Hessian of the cost function (Yang et al. 1996; Rabier and Courtier 1992; Courtier et al. 1994):

$$\mathbf{H} \approx \mathbf{J}'' \approx \mathbf{J}''_p = \frac{1}{p}\sum_{i=1}^{P} \nabla J^i(\nabla J^i)^{\mathrm{T}}. \qquad (123)$$

This approach is analogous to quasi-Newton methods where symmetric rank-one or rank-two updates are collected to update approximations of the Hessian or the inverse of the Hessian matrix as the minimization proceeds. As shown by Yang et al. (1996) use of the approximate $\mathbf{J}''_p$ as a preconditioner is extremely efficient. Forsythe and Strauss (1955) have already shown that using the diagonal of the Hessian is optimal among all diagonal preconditioning methods.

6. Second-order adjoint via automatic differentiation

There is an increased interest in obtaining the second-order adjoint via automatic differentiation (AD).

Research work has been carried out in the recent version of the FORTRAN TAMC AD package designed by Giering and Kaminski (1998a) allowing for both the calculation of Hessian/vector products as well as for the more computationally expensive derivation of the full Hessian with respect to the control variables. Comparable CPU times to those required for hand coding were reported (Giering and Kaminski 1998b).

The importance of the Hessian/vector products derived by AD is particularly important in minimization where there is often interest not only in the first but rather in the second derivatives of the cost functional, which convey crucial information.

Griewank (2000) estimated the computational complexity of implementing second-order adjoints in a thorough manner.

He found that for calculating Hessian/vector products an effort leading to a run-time ratio of about a factor of 13 was required.

The calculation of the ratio between the effort required to obtain Hessian/vector products and that required to calculate the gradient of the cost was found to be a factor between 2 and 3 only.

All analytic differentiation methods are based on the observation that most vector functions f are being evaluated as a sequence of assignments:

$$v_i = \phi_i(v_j)_{j<i} \quad \text{for } i = 1, \ldots, l + m.$$

Here variables v_i are real scalars and the elemental functions ϕ_i are either binary arithmetic operations or univariate intrinsics.

Consequently, only one or two of the partial derivatives,

$$c_{ij} \equiv \frac{\partial}{\partial v_j}\phi_i(v_k)_{k<i},$$

do not vanish identically.

Without loss of generality we may require that the first n variables

$$v_{j-n} = x_j, \qquad j = 1, \ldots, n$$

represent independent variables and that the last m variables

$$y_i = v_{l+i}, \qquad i = 1, \ldots, m$$

represent dependent variables.

In AD calculation of $f'(x)$, it can be represented by a sparse triangular matrix

$$\mathbf{C} = \mathbf{C}(x) = (c_{ij})\begin{cases} j = 1 - n \ldots l + n \\ i = 1 - n \ldots l + m. \end{cases}$$

This $\mathbf{C}$ can also be interpreted as a computational graph whose vertices are elemental functions ϕ_i and the edges are the nonvanishing partial c_{ij}.

Exploiting sparsity for AD calculation of the second-order adjoint Griewank (2000) shows that economy can be realized when the computational Hessian graph symmetry allows the AD-computed Hessian to assume the form

$$\nabla^2 f = \dot{\mathbf{Z}}\mathbf{S}\dot{\mathbf{Z}} \in \mathbb{R}^{n\times n}, \tag{124}$$

which leads to a dyadic representation first put forward in a paper by Jackson and McCormick (1988); see below Eqs. (125)–(126).

The derivation originates in the approach put forward by Griewank (2001) of Jacobian accumulation procedures using implicit function theorem.

Griewank (2000) derives a class of derivative accumulation procedures as edge eliminations on the linearized computational Hessian graph.

Functions defined by an evaluation procedure can be characterized by a triangular system of nonlinear equations $E(X, v) = 0$. Applying the implicit function theorem one obtains the derivative of $\partial v/\partial x$. The $(n + i)$-th row of $\partial v/\partial x$ represents exactly the gradient

$$\dot{\mathbf{V}}_i = \boldsymbol{\nabla}_x v_i$$

of the ith intermediate value v_i with respect to all independent variables:

$$x_j = v_{j-n}, \qquad j = 1, 2, \ldots, n, \qquad x \in \mathbb{R}^n.$$

Now

$$\mathbf{Z} = (\dot{\mathbf{V}}_i^{\mathrm{T}})_{i=1-n}^{l-m} \in \mathbb{R}^{(l-m+n)\times n}$$

denotes the matrix formed by the gradients

$$\dot{\mathbf{V}}_i^{\mathrm{T}} = \boldsymbol{\nabla}_x v_i$$

of all intermediates v_i, with respect to the independents $x \in \mathbb{R}^n$, which is computed during a forward sweep to calculate the gradient

$$\boldsymbol{\nabla} f = \boldsymbol{\nabla}_x f.$$

Then the Hessian takes the product form

$$\nabla^2 f = \dot{\mathbf{Z}}^{\mathrm{T}}\mathbf{S}\dot{\mathbf{Z}} \in \mathbb{R}^{n\times n},$$

where $\mathbf{S} \in \mathbb{R}^{(l-m+n)\times(l-m+n)}$ is zero except for diagonal elements of the form $\bar{v}_j\phi_j$ and diagonal blocks of the form

$$\begin{pmatrix} 0 & \bar{v}_j \\ \bar{v}_j & 0 \end{pmatrix}$$

(ϕ_j being a nonlinear elemental).

One can show that first-order derivatives form the nonzero elements of matrix $\dot{\mathbf{Z}}$.

The representation

$$\nabla^2 f = \dot{\mathbf{Z}}\mathbf{S}\dot{\mathbf{Z}}$$

results in a sum of outer products called dyadic representation, which was used extensively by Jackson and McCormick (1988), who referred to the functions defined by the evolution procedures as ‘‘factorable.’’

Here,

$$\mathbf{Z} \equiv (\dot{\mathbf{V}}_i^{\mathrm{T}})_{i=1,\ldots,n}^{l-m} \in \mathbb{R}^{(l-m+n)\times n} \quad \text{and} \tag{125}$$

$$\mathbf{S} \in \mathbb{R}^{(l-m+n)\times(l-m+n)}, \tag{126}$$

where

$$\dot{\mathbf{V}}_i^{\mathrm{T}} = \boldsymbol{\nabla}_x v_i, \quad \text{and} \tag{127}$$

v_i are intermediates with respect to independents $x \in \mathbb{R}^n$.

7. Use of Hessian of cost functional to estimate error covariance matrices

A relationship exists between the inverse Hessian matrix and the analysis error covariance matrix of either

3DVAR or 4DVAR (see Thacker 1989; Thepaut and Courtier 1991; Rabier and Courtier 1992; Yang et al. 1996; Le Dimet et al. 1997).

Following Courtier et al. (1994) we consider methods for estimating the Hessian in the weakly nonlinear problem when the tangent linear dynamics is a good approximation to nonlinear dynamics. As a consequence the cost function is near to being quadratic. If as in Gauthier (1992) we consider the observations as random variables and we look at variational analysis as attempting to solve the minimization problem

$$\min J(\mathbf{v}) = \frac{1}{2}(\mathbf{x} - \mathbf{x_b})^{\mathrm{T}}\mathbf{B}^{-1}(\mathbf{x} - \mathbf{x_b}) + \frac{1}{2}(\mathbf{Hx} - \mathbf{y})^{\mathrm{T}}\mathbf{O}^{-1}(\mathbf{Hx} - \mathbf{y}), \quad (128)$$

where $\mathbf{x_b}$ is the unbiased background field and y the set of unbiased observations, both being realizations of random variables of covariances **B** and **O**, respectively, and where the operator **H** computes the model equivalent $H\mathbf{x}$ of the observation $\mathbf{y}$, then the Hessian $\mathbf{J}''$ of the cost function J at the minimum is given by

$$\mathbf{J}'' = \mathbf{B}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{O}^{-1}\mathbf{H}, \quad (129)$$

obtained by differentiating (128) twice.

Moreover the analysis error covariance matrix is the inverse of the Hessian as shown in appendix B of Rabier and Courtier (1992). Calling $\mathbf{x_a}$ the result of the minimization (i.e., the analysis) and $\mathbf{x_t}$ the truth, one sees that the error covariance at the minimum is

$$\mathbf{E}[(\mathbf{x_a} - \mathbf{x_t})(\mathbf{x_a} - \mathbf{x_t})^{\mathrm{T}}] = (\mathbf{J}'')^{-1} = (\mathbf{B}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{O}^{-1}\mathbf{H})^{-1}. \quad (130)$$

A requirement is that the background error and the observation error are uncorrelated (Rabier and Courtier 1992; Fisher and Courtier 1995). See also the work of Thepaut and Moll (1990) who point out that the diagonal of the Hessian is optimal among all diagonal preconditioners.

8. Hessian singular vectors

Computing Hessian singular vectors (HSVs) uses the full Hessian of the cost function in the variational data assimilation, which can be viewed as an approximation of the inverse of the analysis error covariance matrix and it is used at initial time to define a norm. The total energy norm is still used at optimization time. See work by Barkmeijer et al. (1998, 1999). The HSVs are consistent with the 3DVAR estimates of the analysis error statistics. They are also defined in the context of 4DVAR. In practice one never knows the full 3DVAR Hessian in its matrix form and a generalized eigenvalue problem is solved as will be described below.

The HSVs are also used in a method first proposed by Courtier (1993) and tested by Rabier et al. (1997) for the development of a simplified Kalman filter fully described by Fisher (1998) and compared with a low-resolution explicit extended Kalman filter by Ehrendorfer and Bouttier (1998).

Let **M** be the propagator of the tangent linear model, and **P** a projection operator setting a vector to zero outside a given domain.

Consider positive-definite and symmetric operators including a norm at initial and optimization time, respectively.

Then the HSVs defined by

$$\frac{\langle \mathbf{P}\epsilon(t), \mathbf{EP}\epsilon(t)\rangle}{\langle \epsilon(t_0), \mathbf{C}\epsilon(t_0)\rangle} \quad (131)$$

under an Euclidean norm are the solution of the generalization eigenvalue problem. Here the positive-definite and symmetric operators **C** and **E** induce a norm at initial and optimization time, respectively. Usually the total energy metric is used and then **C** and **E** are identical:

$$\mathbf{M}^*\mathbf{P}^*\mathbf{EPM}x = \lambda\mathbf{C}x \quad (132)$$

The adjoint operators **M*** and **P*** are determined with respect to the Euclidean inner product.

In HSVs the operator **C** is equal to the Hessian of the 3D-/4DVAR cost function.

The operator **C** is specified to be equal to the full Hessian of the 3DVAR cost function. While **C** is not known in matrix form and determining its square root is not feasible, Barkmeijer et al. (1999) show that one can solve (132) by a generalized eigenvalue problem solver called the generalized Davidson algorithm (see Barkmeijer et al. 1998; Davidson 1975). See also Sleijpen and van der Vorst (1996). Using

$$\mathbf{C} \equiv \nabla^2 J = \mathbf{B}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H} \quad (133)$$

and carrying out a coordinate transformation,

$$x = \mathbf{L}^{-1}x, \qquad \mathbf{LL}^{-1} = \mathbf{B}. \quad (134)$$

Then we have a transformed operator

$$(\mathbf{L}^{-1})^{\mathrm{T}}\mathbf{CL}, \quad (135)$$

and the Hessian becomes equal to the sum of identity and a matrix with rank less than or equal to the dimensions of the vector of observations (see Fisher and Courtier 1995).

The reduced-rank Kalman filter requires as input pairs of vectors that satisfy

$$\mathbf{z_k} = (\mathbf{P}^f)^{-1}\mathbf{s}_k,$$

where $\mathbf{P}^f$ is a flow-dependent approximation to the covariance matrix of the background error. Such pairs of vectors can be calculated during the course of Hessian singular vector calculation (see Fisher 1998).

In this calculation, in which the inner product at the initial time is defined by the Hessian matrix of an analysis cost function, the vectors $\mathbf{s_k}$ are partially evolved singular vectors, while the vectors $\mathbf{z_k}$ are produced during the adjoint model integration.

Veerse (1999) proposes to take advantage of this form of the appropriate Hessian in order to obtain approximations of the inverse analysis error covariance matrix, using the limited memory inverse Broyden, Fletcher, Goldfarb, and Shanno (BFGS) minimization algorithm.

Let H be $(\nabla^2 J)^{-1}$ the inverse Hessian and H^+ the updated version of the inverse Hessian, and

$$\mathbf{s} = \mathbf{x}^{n+1} - \mathbf{x}^n, \qquad (136)$$

where **s** is the difference between the new iterate and the previous one in a limited memory quasi-Newton minimization procedure and

$$\mathbf{y} = \mathbf{g}^{n+1} - \mathbf{g}^n \qquad (137)$$

is the corresponding gradient increment.

One has the formula

$$\mathsf{H}^+ = \mathsf{U}(\mathsf{H}, \mathbf{y}, \mathbf{s}) = \left(\mathsf{I} - \frac{\mathbf{s} \otimes \mathbf{y}}{\langle \mathbf{y}, \mathbf{s} \rangle}\right)\frac{\mathbf{s} \otimes \mathbf{s}}{\langle \mathbf{y}, \mathbf{s} \rangle}, \qquad (138)$$

where $\langle , \rangle$ is a scalar product with respect to which the gradient is defined and $\otimes$ stands for the outer product.

Many minimization methods are implemented by using the inverse Hessian matrix/vector product that is built into the minimization code, such as Nocedal's algorithm (Nocedal 1980). These methods are useful when the second-order adjoint is not available due to either memory or CPU limitations.

9. Numerical experiments: Application of AD Hessian/vector products to the truncated Newton algorithm

For the numerical experiments we consider the truncated-Newton algorithm to minimize the cost function (59) associated with the SWE model (56)–(58). The spatial domain considered is a 6000 km × 4400 km channel with a uniform 21 × 21 spatial grid, such that the dimension of the initial condition vector $(u, v, \phi)^T$ is 1083, and the Hessian of the cost function is a 1083 × 1083 matrix.

The initial conditions are those of Grammeltvedt (1969). As for the boundary conditions, on the southern and northern boundaries the normal velocity components are set to zero, while periodic boundary conditions are assumed in the west–east direction. Integration is performed with a time increment $\Delta t = 600$s and the length of the assimilation window is 10 h. Data assimilation is implemented in a twin experiments framework such that the value of the cost function at the minimum point must be zero. As the set of control parameters, we consider the initial conditions that are perturbed with random values chosen from a uniform distribution.

The second-order adjoint model was generated using TAMC (Giering and Kaminski 1998a). The correctness of the adjoint-generated routines was checked using the small perturbations technique. Assuming that the cost function $J(\mathbf{X})$ is evaluated by the subroutine model $(J, \mathbf{X})$, computation of the Hessian/vector products $\mathsf{G}(\mathbf{X})\mathbf{u}$ via automatic differentiation is performed in two steps. First the reverse (adjoint) mode is applied to generate the adjoint model. Next, the tangent (forward) mode is applied to the adjoint model to generate the second-order adjoint (SOA) model. The performance of the minimization process using AD SOA is analyzed versus an approximate Hessian/vector product computation given by (116), with a hand-coded adjoint model implementation. The absolute and relative differences between the computed Hessian/vector product at the first iteration (initial guess state) are shown in Fig. 1 for the first 100 components. The first-order finite-difference method (FD) provides in average an accuracy of two to three significant digits. The optimization process using FD stops after 28 iterations when the line search fails to find an acceptable step size along the search direction, whereas for the SOA method a relative reduction in the cost function up to the machine precision is reached at iteration 29. The evolution of the normalized cost function and gradient norm are presented in Figs. 2 and 3, respectively.

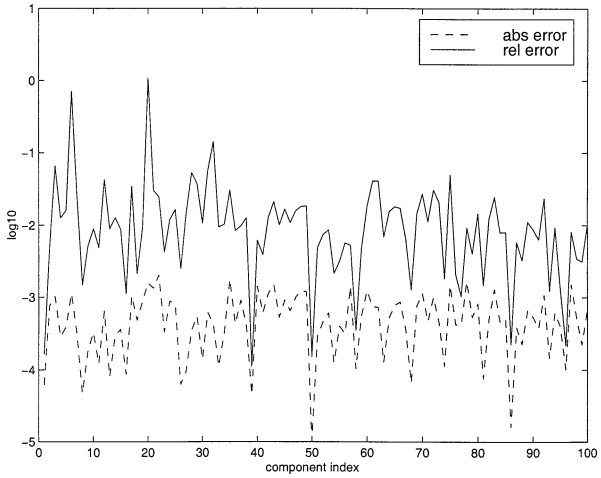

FIG. 1. The absolute (dashed line) and relative (solid line) differences between the Hessian/vector product computed with the SOA method and with the finite-difference method at the first iteration (initial guess state). The first 100 components are considered.

The computational cost is of same order of magnitude for both the finite-difference approach and the exact second-order adjoint approach. The second-order adjoint approach requires integrating the original nonlinear model and its TLM forward in time and integrating the first-order adjoint model and second-order adjoint model backward in time once. The average ratio of the CPU time required to compute the gradient of the cost function to the CPU time used in evaluating the cost function was CPU(∇J)/CPU(J) ≈ 3.7. If we assume that the value of the gradient $\nabla J(\mathbf{X})$ in (116) is already available (previously computed in the minimization algorithm),

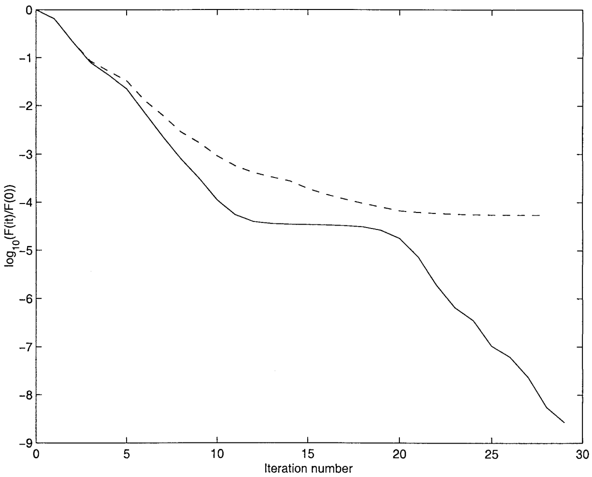

FIG. 2. The evolution of the normalized cost function during the minimization using the SOA method (solid line) and the finite-difference method (dashed line) to compute the Hessian/vector product.

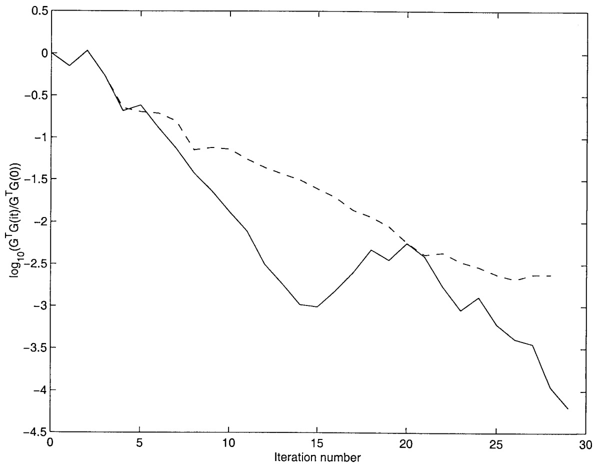

FIG. 3. The evolution of the normalized gradient norm during the minimization using the SOA method (solid line) and the finite-difference method (dashed line) to compute the Hessian/vector product.

only one additional gradient evaluation $\nabla J(\mathbf{X} + h\mathbf{u})$ is needed in (116) to evaluate the Hessian/vector product using the FD method. In this case, we then have an average ratio to compute the Hessian/vector product $\mathrm{CPU}(\mathbf{Gu})_{\mathrm{FD}}/\mathrm{CPU}(J) \approx 3.7$. Using the SOA method to compute the exact Hessian/vector product, we obtained an average $\mathrm{CPU}(\mathbf{Gu})_{\mathrm{SOA}}/\mathrm{CPU}(J) \approx 9.4$, in agreement with the estimate (A4) in the appendix. We notice that in addition to the Hessian/vector product the AD SOA implementation also provides the value of the gradient of the cost function. The average ratio $\mathrm{CPU}(\mathbf{Gu})_{\mathrm{SOA}}/\mathrm{CPU}(\nabla J) \approx 2.5$ we obtained is also in agreement with the CPU estimate (A2) in the appendix.

Numerical calculation of Hessian eigenvalues

Iterative methods and the SOA model may be combined to obtain information about the spectrum of the Hessian matrix of the cost function. In this application we used the ARPACK package (Lehoucq et al. 1998) to compute five of the largest and smallest eigenvalues of the Hessian matrix. The method used is the implicitly restarted Arnoldi method (IRAM), which reduces to the implicitly restarted Lanczos method (IRLM) since **G** is symmetric. For our application, only the action of the Hessian matrix on a vector is needed and we provide this routine using the SOA model. The condition number is evaluated as

$$k(\mathbf{G}) = \frac{\lambda_{\max}}{\lambda_{\min}}. \tag{139}$$

The computed Ritz values and the relative residuals are included in Table 1 for the Hessian evaluated at the initial guess point, and in Table 2 for the Hessian evaluated at the optimal point $\mathbf{X}^*$. For our test example the eigenvalues of the Hessian are positive, such that the Hessian is positive definite and the existence of a minimum point is assured. The condition number of the Hessian is of order $k(\mathbf{G}) \sim 10^4$, which explains the slow convergence of the minimization process.

Use of the Hessian of cost function eigenvalue information in regularization of ill-posed problems was illustrated by Alekseev and Navon (2001, 2002). The application consisted of a wavelet regularization approach for dealing with an ill-posed problem of adjoint parameter estimation applied to estimating inflow parameters from downflow data in an inverse convection case applied to the two-dimensional parabolized Navier–Stokes equations. The wavelet method provided a decomposition into two subspaces, by identifying both a well-posed as well as an ill-posed subspace, the scale of which was determined by finding the minimal eigenvalues of the Hessian of a cost functional measuring the lack of fit between model prediction and observed parameters. The control space is transformed into a wavelet space. The Hessian of the cost was obtained

TABLE 1. First five largest and smallest computed Ritz values of the Hessian matrix and the corresponding relative residuals. The Hessian is evaluated at the initial guess point.

Largest values	Relative residuals	Smallest values	Relative residuals
$5.294\,32 \times 10^{2}$	$1.743\,29 \times 10^{-6}$	$3.190\,71 \times 10^{-2}$	$3.040\,94 \times 10^{-3}$
$4.871\,11 \times 10^{2}$	$2.186\,54 \times 10^{-6}$	$6.023\,01 \times 10^{-2}$	$2.856\,39 \times 10^{-3}$
$4.356\,18 \times 10^{2}$	$1.795\,99 \times 10^{-6}$	$7.779\,66 \times 10^{-2}$	$1.443\,37 \times 10^{-3}$
$3.868\,87 \times 10^{2}$	$2.036\,00 \times 10^{-6}$	$7.830\,50 \times 10^{-2}$	$1.994\,69 \times 10^{-3}$
$3.815\,11 \times 10^{2}$	$1.808\,12 \times 10^{-6}$	$9.164\,25 \times 10^{-2}$	$1.756\,24 \times 10^{-3}$

TABLE 2. First five largest and smallest computed Ritz values of the Hessian matrix and the corresponding relative residuals. The Hessian is evaluated at the computed optimal point.

Largest values	Relative residuals	Smallest values	Relative residuals
$5.129\ 37 \times 10^{2}$	$1.635\ 03 \times 10^{-6}$	$1.467\ 26 \times 10^{-2}$	$5.878\ 33 \times 10^{-3}$
$4.739\ 81 \times 10^{2}$	$1.540\ 48 \times 10^{-6}$	$1.714\ 99 \times 10^{-2}$	$1.605\ 47 \times 10^{-2}$
$4.196\ 11 \times 10^{2}$	$1.731\ 89 \times 10^{-6}$	$4.569\ 08 \times 10^{-2}$	$3.072\ 14 \times 10^{-3}$
$3.888\ 57 \times 10^{2}$	$1.704\ 23 \times 10^{-6}$	$7.229\ 96 \times 10^{-2}$	$1.895\ 07 \times 10^{-3}$
$3.785\ 70 \times 10^{2}$	$1.892\ 42 \times 10^{-6}$	$8.282\ 72 \times 10^{-2}$	$3.006\ 16 \times 10^{-3}$

either by a discrete differentiation of the gradients of the cost derived from the first-order adjoint or by using the full second-order adjoint. The minimum eigenvalues of the Hessian are obtained either by employing a shifted iteration method following Zou et al. (1992) or by using the Rayleigh quotient. The numerical results obtained illustrated the usefulness and applicability of this algorithm if the Hessian minimal eigenvalue is greater or equal to the square of the data error dispersion, in which case the problem can be considered to be well-posed (i.e., regularized). If the regularization fails, that is, the minimal Hessian eigenvalue is less than the square of the data error dispersion of the problem, the following wavelet scale should be neglected, followed by another algorithm iteration.

10. Summary and conclusions

The recent development of variational methods in operational meteorological centers (European Centre for Medium-Range Weather Forecasts, Météo-France) has demonstrated the strong potential of these methods.

Variational techniques require the development of powerful tools such as the adjoint model, which are useful for the adjustment of the inputs of the model (initial and/or boundary conditions). From the mathematical point of view the first-order adjoint will provide only necessary conditions for an optimal solution. The second-order analysis goes one step further and provides information that is essential for many applications:

(i) Sensitivity analysis should be derived from a second-order analysis, that is, from the derivation of the optimality system. This is made crystal clear when sensitivity with respect to observations is required. In the analysis, observations appear only as a forcing term in the adjoint model; therefore, in order to estimate the impact of observations this is the system that should be derived.

(ii) Second-order information will improve the convergence of the optimization methods, which are the basic algorithmic components of variational analysis.

(iii) The second-order system permits estimating the covariances of the fields. This information is essential for the estimation of the impact of errors on the prediction.

The computational cost to be paid in order to obtain the second-order adjoint system is twofold: 1) We have to consider the computational cost for the derivation of the SOA. It has been seen that we can get it directly from the linear tangent model and from the adjoint model. Only the right-hand sides should be modified. 2) Computing the second-order information. Basically the first-order information has the same dimension as the input of the model. Let *n* be this dimension. The second-order information will be represented by an $n \times n$ matrix. For operational models the computation of the full Hessian matrix is prohibitive; nevertheless, it is possible to extract the most useful information (eigenvalues and eigenvectors, spectrum, condition number, etc.) at a reasonable computational cost. The numerical results obtained illustrate the ease with which present-day automatic differentiation packages allow one to obtain second-order adjoint models as well as Hessian/vector products. They also confirm numerically the CPU estimates for computational complexity as derived in section 7 (see also Griewank 2000).

Numerical calculation of the leading eigenvalues of the Hessian along with its smallest eigenvalues yields results similar to those obtained by Wang et al. (1998) and allows for valuable insight into the Hessian spectrum, thus allowing us to deduct the important information related to condition number of the Hessian and, hence, to the expected rate of convergence of minimization algorithms.

With the advent of ever more powerful computers, the use of second-order information in data assimilation will be within realistic reach for 3D models and is expected to become more prevalent.

The purpose of this paper has been to demonstrate the importance of new developments in second-order analysis: many directions of research remain open in this domain.

Acknowledgments. The authors wish to thank two anonymous reviewers whose constructive comments led to a marked improvement in the presentation of the present paper.

The second author would like to acknowledge support from the NSF through Grant ATM-9731472, managed by Dr. Pamela Stephens, whom we would like to thank for her support.

IDOPT is a joint project CNRS–INRIH–Université Joseph Fourier INPG.

APPENDIX

Computational Complexity of AD Calculation of the Second-Order Adjoint

Griewank (2000) starts by working out a representation of the complexity measure as a task consisting of moves, adds, multiplications, and nonlinear operations, thus obtaining a representation of work (task) as:

$$\text{work(task)} = \begin{pmatrix} \text{moves} \\ \text{adds} \\ \text{mults} \\ \text{nlops} \end{pmatrix} = \begin{pmatrix} \text{no. of fetches and stores} \\ \text{no. of adds and subtracts} \\ \text{no. of multiplications} \\ \text{no. of nonlinear operations} \end{pmatrix}.$$

Then run time can be written as

$$\text{TIME}[\text{task}(F)] = \mathbf{w}^{\text{T}}\text{work}[\text{task}(F)].$$

Here **w** is a vector of positive weights that depend on the computing system and represent the number of clock cycles needed for fetching and/or storing data items, multiplication, addition, and finally for taking into account nonlinear operations.

Usually the vector $\mathbf{w}^{\text{T}}$ assumes the form

$$\mathbf{w}^{\text{T}} = (\mu, 1, \pi, \nu) \tag{A1}$$

and for most computing platforms $\mu \geq \max(1, \pi/2)$, $\pi \leq 1$ and $\nu \leq 2\pi$. For example, this assumption implies that a memory access (μ) is at least as slow as an addition or half a multiplication (π). Griewank (2000) derives the computational complexity of the tangent model (directional derivative) w_{tang}, gradient (first order adjoint) w_{grad}, and second-order adjoint w_{SOA} normalized by the complexity of the model evaluation as

$$w_{\text{tang}} = \max\left\{\frac{2\mu}{\mu}, \frac{6\mu + 2}{3\mu + 1}, \frac{6\mu + 1 + 3\pi}{3\mu + \pi}, \frac{4\mu + \pi + 2\nu}{2\mu + \nu}\right\} \in \left[2, \frac{5}{2}\right] \tag{A2}$$

$$w_{\text{grad}} = \max\left\{\frac{2\mu}{\mu}, \frac{9\mu + 3}{3\mu + 1}, \frac{11\mu + 2 + 3\pi}{3\mu + \pi}, \frac{7\mu + 1 + \pi + 2\nu}{2\mu + \nu}\right\} \in [3, 4] \tag{A3}$$

$$w_{\text{SOA}} = \max\left\{\frac{4 + \mu}{\mu}, \frac{18\mu + 6}{3\mu + 1}, \frac{22\mu + 7 + 9\pi}{3\mu + \pi}, \frac{\mu + 3 + 5\pi + 3\nu}{2\mu + \nu}\right\} \in [7, 10]. \tag{A4}$$

As mentioned by Nocedal and Wright (1999) automatic differentiation has increasingly been using more sophisticated techniques that allow, when used in reverse mode, for calculation either of full Hessians or Hessian/vector products. However the automatic differentiation technique should not be regarded as a substitute for the user to think that this is a fail-safe product and each derivative calculation obtained with AD should be carefully assessed.

Gay (1996) has shown how to use partial separability of the Hessian in AD while Powell and Toint (1979) and Coleman and More (1984), along with Coleman and Cai (1986), have shown how to estimate a sparse Hessian using either graph-coloring techniques or other highly effective schemes.

Software for the estimation of sparse Hessians is available in the work of Coleman et al. (1985a,b). See also the work of Dixon (1991) and the general presentation of Gilbert (1992).

Averbukh et al. (1994) supplemented the work of Moré et al. (1981), which provides function and gradient subroutines of 18 test functions for multivariate minimization. Their supplementary Hessian segments enable users to test optimization software that requires second derivative information.

REFERENCES

Abate, J., C. Bischof, A. Carle, and L. Roh, 1997: Algorithms and design for a second-order automatic differentiation module. *Proc. Int. Symp. on Symbolic and Algebraic Computing (ISSAC) '97,* Maui, HI, Association of Computing Machinery, 149–155.

Alekseev, K. A., and I. M. Navon, 2001: The analysis of an ill-posed problem using multiscale resolution and second order adjoint techniques. *Comput. Methods Appl. Mech. Eng.,* **190,** 1937–1953.

——, and ——, 2002: On estimation of temperature uncertainty using the second order adjoint problem. *Int. J. Comput. Fluid Dyn.*, in press.

Arian, E., and S. Ta'asan, 1999: Analysis of the Hessian for aerodynamic optimization: inviscid flow. *Comput. Fluids,* **28,** 853–877.

Averbukh, V. Z., S. Figueroa, and T. Schlick, 1994: Remark on Algorithm-566. *ACM Trans. Math Software,* **20,** 282–285.

Barkmeijer, J., M. van Gijzen, and F. Bouttier, 1998: Singular vectors and estimates of the analysis-error covariance metric. *Quart. J. Roy. Meteor. Soc.,* **124A,** 1695–1713.

——, R. Buizza, and T. N. Palmer, 1999: 3D-Var Hessian singular vectors and their potential use in the ECMWF Ensemble Prediction System. *Quart. J. Roy. Meteor. Soc.,* **125B,** 2333–2351.

Bischof, C. H., 1995: Automatic differentiation, tangent linear models, and (pseudo) adjoints. *Proceedings of the Workshop on High-Performance Computing in the Geosciences,* F.-X. Le Dimet, Ed., NATO Advanced Science Institutes Series C: Mathematical and Physical Sciences, Vol. 462, Kluwer Academic, 59–80.

Burger, J., J. L. Brizaut, and M. Pogu, 1992: Comparison of two methods for the calculation of the gradient and of the Hessian of the cost functions associated with differential systems. *Math. Comput. Simul.,* **34,** 551–562.

Coleman, T. F., and J. J. More, 1984: Estimation of sparse Hessian matrices and graph-coloring problems. *Math. Program.,* **28,** 243–270.

——, and J. Y. Cai, 1986: The cyclic coloring problem and estimation of sparse Hessian matrices. *SIAM J. Algebra Discrete Math,* **7,** 221–235.

——, ——, and ——, 1985a: Fortran subroutines for estimating sparse Hessian matrices. *ACM Trans. Math. Software,* **11** (4), 378–378.

——, B. S. Garbow, and J. J. More, 1985b: Software for estimating sparse Hessian matrices. *ACM Trans. Math. Software,* **11** (4), 363–377.

Courtier, P., 1993: Introduction to numerical weather prediction data assimilation methods. *Proc. ECMWF Seminar on Developments in the Use of Satellite Data in Numerical Weather Prediction,* Reading, United Kingdom, ECMWF, 189–209.

——, J.-N. Thepaut, and A. Hollingsworth, 1994: A strategy for operational implementation of 4D-Var, using an incremental approach. *Quart J. Roy. Meteor. Soc.,* **120,** 1367–1388.

Davidon, W. C., 1991: Variable metric method for minimization. *SIAM J. Optim.,* **1,** 1–17.

Davidson, E. R., 1975: The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of a large real symmetric matrices. *J. Comput. Phys.,* **17,** 87–94.

Dembo, R. S., and T. Steihaug, 1983: Truncated-Newton algorithms for large-scale unconstrained optimization. *Math. Program.,* **26,** 190–212.

——, S. C. Eisenstat, and T. Steihaug, 1982: Inexact Newton methods. *SIAM J. Numer. Anal.,* **19,** 400–408.

Dixon, L. C. W., 1991: Use of automatic differentiation for calculating Hessians and Newton steps. *Automatic Differentiation of Algorithms: Theory, Implementation, and Application,* A. Griewank and G. F. Corliss, Eds., SIAM, 115–125.

Ehrendorfer, M., and F. Bouttier, 1998: An explicit low-resolution extended Kalman filter: Implementation and preliminary experimentation. ECMWF Tech. Memo. 259, 27 pp.

Fisher, M., 1998: Development of a simplified Kalman filter. ECMWF Tech. Memo. 260, 16 pp.

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

Forsythe, G. E., and E. G. Strauss, 1955: On best conditioned matrices. *Proc. Amer. Math. Soc.,* **6,** 340–345.

Gauthier, P., 1992: Chaos and quadri-dimensional data assimilation: A study based on the Lorentz model. *Tellus,* **44A,** 2–17.

Gay, D. M., 1996: More AD of nonlinear AMPL models: Computing Hessian information and exploiting partial separability in computational differentiation: Techniques, applications, and tools. *Proceedings in Applied Mathematics,* M. Berz et al., Eds., Vol. 89, SIAM, 173–184.

Giering, R., and T. Kaminski, 1998a: Recipes for adjoint code construction. *ACM Trans. Math. Software,* **24** (4), 437–474.

——, and ——, 1998b: Using TAMC to generate efficient adjoint code: Comparison of automatically generated code for evaluation of first and second order derivatives to hand written code from the Minpack-2 collection. *Automatic Differentiation for Adjoint Code Generation,* C. Faure, Ed., INRIA Research Rep. 3555, 31–37.

Gilbert, J. C., 1992: Automatic differentiation and iterative processes. *Optim. Methods Software,* **1,** 13–21.

Gill, P. E., and W. Murray, 1979: Newton-type methods for unconstrained and linearly constrained optimization. *Math. Program.,* **28,** 311–350.

——, ——, and M. H. Wright, 1981: *Practical Optimization.* Academic Press, 401 pp.

Grammeltvedt, A., 1969: A survey of finite-difference schemes for the primitive equations for a barotropic fluid. *Mon. Wea. Rev.,* **97,** 387–404.

Griewank, A., 1993: Some bounds on the complexity of gradients, Jacobians, and Hessians. *Complexity in Nonlinear Optimization,* P. M. Pardalos, Ed., World Scientific, 128–161.

——, 2000: *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation.* Frontiers in Applied Mathematics, Vol. 19, SIAM, 369 pp.

——, 2001: Complexity of gradients, Jacobians and Hessians. *Encyclopaedia of Optimization,* C. A. Floudas and P. M. Pardalos, Eds., Vol. 1, Kluwer Academic, 290–300.

——, and G. F. Corliss, 1991: *Automatic Differentiation of Algorithms: Theory, Implementation, and Application.* SIAM, 353 pp.

Hou, G. J.-W., and J. Sheen, 1993: Numerical methods for second order shape sensitivity analysis with application to heat conduction problems. *Int. J. Numer. Methods Eng.,* **36,** 417–435.

Hovland, P., 1995: Using ADIFOR 1.0 to Compute Hessians. Center for Research on Parallel Computation Tech. Rep. CRPC-TR95540-S, Rice University, Houston, TX, 12 pp.

Ide, K., P. Courtier, M. Ghil, and A. Lorenc, 1997: Unified notation for data assimilation: Operational sequential and variational. *J. Meteor. Soc Japan,* **75B,** 71–79.

Jackson, R. H. F., and G. P. McCormic, 1988: Second order sensitivity analysis in factorable programming: Theory and applications. *Math. Program.,* **41,** 1–28.

Kalnay, E., S. K. Park, Z.-X. Pu, and J. Gao, 2000: Application of the quasi-inverse method to data assimilation. *Mon. Wea. Rev.,* **128,** 864–875.

Le Dimet, F. X., and I. Charpentier, 1998: Methodes de second order en assimilation de donnees. *Equations aux Dérivées Partielles et Applications (Articles Dédiées à Jacques-Louis Lions*), Gauthier-Villars, 623–640.

——, H. E. Ngodock, B. Luong, and J. Verron, 1997: Sensitivity analysis in variational data assimilation. *J. Meteor. Soc. Japan,* **75B,** 245–255.

Lehoucq, R. B., D. C. Sorensen, and C. Yang, 1998: *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods.* Software, Environments, and Tools, Vol. 6, SIAM, 160 pp.

Liu, D. C., and J. Nocedal, 1989: On the limited memory BFGS method for large scale minimization. *Math. Progam.,* **45,** 503–528.

Moré, J. J., B. S. Garbow, and K. E. Hillstrom, 1981: Testing unconstrained optimization software. *ACM Trans. Math. Software,* **7,** 17–41.

Nash, S. G., 1984a: Newton-type minimization via the Lanczos method. *SIAM J. Numer. Anal.,* **21,** 770–788.

——, 1984b: Truncated-Newton methods for large-scale function minimization. *Applications of Nonlinear Programming to Optimization and Control,* H. E. Rauch, Ed., Pergamon Press, 91–100.

——, 1984c: User's guide for TN/TNBC: Fortran routines for nonlinear optimization. Mathematical Sciences Dept. Tech. Rep. 307, The Johns Hopkins University, 17 pp.

——, 1984d: Solving nonlinear programming problems using truncated-Newton techniques. *Numerical Optimization,* P. T. Boggs, R. H. Byrd, and R. B. Schnabel, Eds., SIAM, 119–136.

——, 1985: Preconditioning of truncated-Newton methods. *SIAM J. Sci. Stat. Comput.,* **6,** 599–616.

——, 2000: A survey of truncated-Newton methods. *J. Comput. Appl. Math.,* **124,** 45–59.

——, and A. Sofer, 1989a: Block truncated-Newton methods for parallel optimization. *Math. Progam.,* **45,** 529–546.

——, and ——, 1989b: A parallel line search for Newton type methods in computer science and statistics. *Proc. 21st Symp. on the Interface: Computing Science and Statistic,* Orlando, FL, American Statistical Association, 134–137.

——, and J. Nocedal, 1991: A numerical study of the limited memory BFGS method and the truncated-Newton method for large scale optimization. *SIAM J. Optim.,* **1,** 358–372.

——, and ——, 1996: *Linear and Nonlinear Programming.* McGraw-Hill, 692 pp.

Ngodock, H. E., 1996: Data assimilation and sensitivity analysis. Ph.D. thesis, University Joseph Fourier, Grenoble, France, 213 pp.

Nocedal, J., 1980: Updating quasi-Newton matrices with limited storage. *Math. Comput.,* **35,** 773–782.

——, and S. J. Wright, 1999: *Numerical Optimization.* Springer Verlag Series in Operations Research, 656 pp.

O'Leary, D. P., 1983: A discrete Newton algorithm for minimizing a function of many variables. *Math. Progam.,* **23,** 20–23.

Powell, M. J. D., and P. L. Toint, 1979: Estimation of sparse Hessian matrices. *SIAM J. Numer. Anal.,* **16,** 1060–1074.

Pu, Z. X., and E. Kalnay, 1999: Targeting observations with the quasi-inverse linear and adjoint NCEP global models: Performance during FASTEX. *Quart. J. Roy. Meteor. Soc.,* **125,** 3329–3337.

——, and Coauthors, 1997: Sensitivity of forecast errors to initial conditions with a quasi-inverse linear method. *Mon. Wea. Rev.,* **125,** 2479–2503.

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

——, and Coauthors, 1997: Recent experimentation on 4D-Var and first results from a simplified Kalman filter. ECMWF Tech. Memo. 240, 42 pp.

Reuther, J. J., 1996: Aerodynamic shape optimization using control theory. Ph.D. dissertation, University of California, Davis, 226 pp.

Santosa, F., and W. W. Symes, 1988: Computation of the Hessian for least-squares solutions of inverse problems of reflection seismology. *Inverse Problems,* **4,** 211–233.

——, and ——, 1989: *An Analysis of Least Squares Velocity Inversion. Geophysical Monogr.*, Vol. 4, Society of Exploration Geophysicists, 168 pp.

Schlick, T., and A. Fogelson, 1992a: TNPACK—A truncated Newton minimization package for large-scale problems: I. Algorithm and usage. *ACM Trans. Math. Software,* **18,** 46–70.

——, and ——, 1992b: TNPACK—A truncated Newton minimization package for large-scale problems: II. Implementation examples. *ACM Trans. Math. Software,* **18,** 71–111.

Sleijpen, G. L. G., and H. A. van der Vorst, 1996: A Jacobi–Davidson iteration method for linear eigenvalue problems. *SIAM J. Matrix Anal.,* **17A,** 401–425.

Symes, W. W., 1990: Velocity inversion: A case study in infinite-dimensional optimization. *Math. Program.,* **48,** 71–102.

——, 1991: A differential semblance algorithm for the inverse problem of reflection seismology. *Comput. Math. Appl.,* **22** (4/5), 147–178.

——, 1993: A differential semblance algorithm for the inversion of multioffset seismic reflection data. *J. Geophys. Res.,* **98** (B2), 2061–2073.

Thacker, W. C., 1989: The role of Hessian matrix in fitting models to measurements. *J. Geophys. Res.,* **94,** 6177–6196.

Thepaut, J.-N., and P. Moll, 1990: Variational inversion of simulated TOVS radiances using the adjoint technique. *Quart. J. Roy. Meteor. Soc.,* **116,** 1425–1448.

——, and P. Courtier, 1991: Four-dimensional variational assimilation using the adjoint of a multilevel primitive equation model. *Quart. J. Roy. Meteor. Soc.,* **117,** 1225–1254.

Veerse, F., 1999: Variable-storage quasi-Newton operators as inverse forecast/analysis error covariance matrices in data assimilation. INRIA Tech. Rep. 3685, 28 pp.

Wang, Z., 1993: Variational data assimilation with 2-D shallow water equations and 3-D FSU global spectral models. Tech. Rep. FSU-SCRI-93T-149, The Florida State University, Tallahassee, FL, 235 pp.

——, I. M. Navon, F. X. Le Dimet, and X. Zou, 1992: The second order adjoint analysis: Theory and application. *Meteor. Atmos. Phys.,* **50,** 3–20.

——, ——, and X. Zou, 1993: The adjoint truncated Newton algorithm for large-scale unconstrained optimization. Tech. Rep. FSU-SCRI-92-170, The Florida State University, Tallahassee, FL, 44 pp.

——, ——, ——, and F. X. Le Dimet, 1995: A truncated-Newton optimization algorithm in meteorology applications with analytic Hessian/vector products. *Comput. Optim. Appl.,* **4,** 241–262.

——, K. K. Droegemeier, L. White, and I. M. Navon, 1997: Application of a new adjoint Newton algorithm to the 3D ARPS storm-scale model using simulated data. *Mon. Wea. Rev.,* **125,** 2460–2478.

——, ——, and ——, 1998: The adjoint Newton algorithm for large-scale unconstrained optimization in meteorology applications. *Comput. Optim. Appl.,* **10,** 283–320.

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

Zou, X., I. M. Navon, F. X. Le Dimet, A. Nouailler, and T. Schlick, 1990: A comparison of efficient large-scale minimization algorithms for optimal control applications in meteorology. Tech. Rep. FSU-SCRI-90-167, The Florida State University, Tallahassee, FL, 44 pp.

——, ——, and F. X. Le Dimet, 1992: Incomplete observations and control of gravity waves in variational data assimilation. *Tellus,* **44A,** 273–296.

——, ——, M. Berger, P. K. H. Phua, T. Schlick, and F. X. Le Dimet, 1993: Numerical experience with limited-memory quasi-Newton methods and truncated Newton methods. *SIAM J. Numer. Optim.,* **3,** 582–608.