# Variational Data Assimilation with a Variable Resolution Finite-Element Shallow-Water Equations Model

KEYUN ZHU*

*Supercomputer Computations Research Institute, The Florida State University, Tallahassee, Florida*

I. MICHAEL NAVON

*Department of Mathematics and Supercomputer Computations Research Institute, The Florida State University, Tallahassee, Florida*

XIAOLEI ZOU

*Supercomputer Computations Research Institute, The Florida State University, Tallahassee, Florida*

(Manuscript received 2 July 1993, in final form 20 October 1993)

ABSTRACT

The adjoint model of a finite-element shallow-water equations model was obtained with a view to calculate the gradient of a cost functional in the framework of using this model to carry out variational data assimilation (VDA) experiments using optimal control of partial differential equations.

The finite-element model employs a triangular finite-element Galerkin scheme and serves as a prototype of 2D shallow-water equation models with a view of tackling problems related to VDA with finite-element numerical weather prediction models. The derivation of the adjoint of this finite-element model involves overcoming specific computational problems related to obtaining the adjoint of iterative procedures for solving systems of nonsymmetric linear equations arising from the finite-element discretization and dealing with irregularly ordered discrete variables at each time step.

The correctness of the adjoint model was verified at the subroutine level and was followed by a gradient check conducted once the full adjoint model was assembled. VDA experiments were performed using model-generated observations. In our experiments, assimilation was carried out assuming that observations consisting of a full-model-state vector are available at every time step in the window of assimilation. Successful retrieval was obtained using the initial conditions as control variables, involving the minimization of a cost function consisting of the weighted sum of difference between model solution and model-generated observations.

An additional set of experiments was carried out aiming at evaluating the impact of carrying out VDA involving variable mesh resolution in the finite-element model over the entire assimilation period. Several conclusions are drawn related to the efficiency of VDA with variable horizontal mesh resolution finite-element discretization and the transfer of information between coarse and fine meshes.

## 1. Introduction

In recent years 4D variational data assimilation (VDA) techniques have progressively gained in popularity, being applied both in meteorology and oceanography. Recent research papers in this area include, for instance, Derber (1987, quasigeostrophic finite-difference model), Le Dimet and Talagrand (1986), Courtier and Talagrand (1987, 2D finite-difference model), Ghil (1989), Talagrand and Courtier (1987, finite-difference model), Derber (1989, quasigeostrophic finite-difference model), Navon et al. (1992, 3D spectral model), Zou et al. (1992), and Thépaut et al. (1993, 3D spectral model), to cite but a few. The adjoint model of a meteorological or oceanographic model is the most important component in the implementation of the 4D VDA process, being used to efficiently calculate the gradient of the cost function with respect to the initial conditions and/or other control variables. The cost function is minimized by efficient large-scale limited-memory quasi-Newton unconstrained minimization techniques (Liu and Nocedal 1989; Nocedal 1980; Gilbert and Lemarechal 1989) that require both function and gradient values.

The main practical problem related to the operational implementation of VDA is reducing CPU time required to carry out minimization of VDA at operational resolution. In current operational practice at the Euro-

[*] Current affiliation: Department of Meteorology, Chengdu Meteorological Institute, Chengdu, China.

*Corresponding author address:* Dr. I. Michael Navon, SCRI-4052, 415 Science Library, The Florida State University, Tallahassee, FL 32306-4052.

pean Centre for Medium-Range Weather Forecasts (ECMWF), for example, the CPU time required for a 24-h 4D VDA is equivalent to the CPU time of 100 days of integration of the model (Courtier et al. 1993). To achieve this amount of computation within operational deadlines requires both a significantly faster computer and substantial algorithmic improvements. The massively parallel computers are promising faster performance, and the finite-element method is a promising candidate for yielding efficient algorithms aimed at porting numerical weather prediction models to high-performance parallel computers.

Until now most of the VDA techniques in meteorology were applied to either finite-difference (such as Talagrand and Courtier 1987; Courtier and Talagrand 1987; Zou et al. 1992) or spectral discretized models (Thépaut et al. 1993; Navon et al. 1992, among others), and thus a rich body of experience in deriving the adjoint models of finite-difference and 2D and 3D spectral numerical weather prediction models has been gained recently in meteorology. Our motivation is to explore the feasibility of carrying out VDA using a finite-element model. In this paper, therefore, we present for the first time the derivation of the adjoint of a finite-element shallow-water equations model in meteorology (Navon 1979) to perform VDA, which will serve as a test-bed for carrying out further research along this line using a primitive equations 3D global model.

In general, the solution of a continuum problem approximated by the finite-element method always follows an orderly step-by-step process. With reference to static structural problems, the step-by-step procedure always includes the following stages: assembly of element equations to obtain the global equilibrium equations and solution for the unknown nodal displacements, that is, solving for the unknown variables in the global matrix using iterative methods (Rao 1989). This process of solution is different from the finite-difference and spectral discretization methods. Therefore, the adjoint model development of a finite-element model has its specific numerical difficulties—for example, the treatment of the iterative process required for solving the systems of linear algebraic equations resulting from the finite-element discretization.

A finite-element Galerkin discretization method with piecewise linear triangular elements is employed in the model, and the variable value at a node is related to variable values at surrounding six nodes (for inner nodes) or four nodes (for boundary nodes) in different directions. [See standard textbooks on finite-element methods such as Strang and Fix (1973) and Huebner and Thornton (1982).] The systems of algebraic linear equations resulting from the finite-element discretization of the shallow-water equations model were solved by a Gauss–Seidel iterative method. To save computer memory, a compact storage scheme for the banded and sparse global matrices (see Hinsman 1975; Hinsman and Archer 1976; Woodward 1981) was used. It reduces the $N \times N$ global coefficient matrix to a compact system of an $N \times 7$ algebraic equation and introduces irregularly distributed storage locations for the subscript of a variable in the process of solving the systems of algebraic linear equations. This compact storage procedure induces certain difficulties in building the adjoint of such a model.

In the present research we describe techniques employed for building the adjoint of the finite-element model along with some simple retrieval experiments, while in a follow-up paper we will develop an adjoint VDS system for Staniforth's 3D global finite-element numerical weather prediction model.

While the present finite-element model is not tied directly to Staniforth's model (see Staniforth and Temperton 1986; Côté et al. 1990a, 1990b; Tanguay et al. 1989), the two models can be easily generalized to either a vorticity divergence or a primitive formulation so as to have a large resemblance between them (Staniforth 1993, personal communication; Neta 1992; Williams 1981; Cullen 1979). The difference between using finite-element discretization with basis functions on either triangles or rectangles is not of essence and both this model and Staniforth's use the same Galerkin techniques. As such, once the computational issues related to developing the adjoint of one finite-element model are addressed, deriving the adjoint of another finite-element model of the shallow-water equations would constitute a much simpler task.

The main focus of this paper centers on describing the development of the tangent linear and the adjoint models of the finite-element shallow-water equations model as well as illustrating its use on a retrieval case when the initial conditions serve as control variables. The plan of this paper is as follows. The finite-element Galerkin method for the shallow-water equations model and the derivation of its tangent linear model are briefly described in section 2. The adjoint model of the nonlinear finite-element shallow-water equations model and the verification of the gradient of the cost function with respect to the control variables are described in section 3. Basic assimilation experiments using a random perturbation of the initial conditions as observations and their results are presented in section 4. An experiment aimed at assessing the impact of different horizontal mesh resolutions of the finite-element model on variational data assimilation and the transfer of information between coarse and fine horizontal mesh resolutions is provided in section 5, followed in section 6 by a summary and conclusions. A description of finite-element assembled matrices used in compact storage schemes is included in appendix A, while the description of the derivation of the adjoint of the Gauss–Seidel iterative procedure is provided in appendix B.

## 2. The shallow-water equations model

### *a. The shallow-water equations and its tangent linear model*

The barotropic nonlinear shallow-water equations on a limited-area domain of a rotating earth (using the $\beta$-plane approximation) assume the following form in Cartesian coordinates:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + \frac{\partial \phi}{\partial x} - fv = 0 \qquad (2.1)$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + \frac{\partial \phi}{\partial y} + fu = 0 \qquad (2.2)$$

$$\frac{\partial \phi}{\partial t} + \frac{\partial(\phi u)}{\partial x} + \frac{\partial(\phi v)}{\partial y} = 0 \qquad (2.3)$$

$$0 \leqslant x \leqslant L, \quad 0 \leqslant y \leqslant D,$$

where $u$ and $v$ are the velocity components in the $x$ and $y$ directions, respectively; $f$ is the Coriolis parameter defined by the $\beta$-plane approximation:

$$f = \hat{f} + \beta\left(y - \frac{D}{2}\right), \qquad (2.4)$$

where $\hat{f}$ and $\beta$ are constants, $L$ and $D$ are the dimensions of the rectangular domain of integration, $\phi = gh$ is the geopotential height, $h$ is the depth of the fluid, and $g$ is the acceleration of gravity. The Coriolis parameter $\hat{f} = 2\Omega \sin\theta$ is defined at a mean latitude $\theta_0$, where $\Omega$ is the angular velocity of the earth's rotation and $\theta$ is latitude.

Periodic boundary conditions are assumed in the east–west direction, while a solid wall condition on the north–south boundaries is imposed:

$$v(x, 0, t) = v(x, D, t) = 0. \qquad (2.5)$$

The tangent linear model of the shallow-water equations (2.1)–(2.3) can be written as

$$\frac{\partial u'}{\partial t} + u'\frac{\partial u}{\partial x} + v'\frac{\partial u}{\partial y} + \frac{\partial \phi'}{\partial x} + u\frac{\partial u'}{\partial x} + v\frac{\partial u'}{\partial y} - fv' = 0 \qquad (2.6)$$

$$\frac{\partial v'}{\partial t} + u'\frac{\partial v}{\partial x} + v'\frac{\partial v}{\partial y} + \frac{\partial \phi'}{\partial y} + u\frac{\partial v'}{\partial x} + v\frac{\partial v'}{\partial y} + fu' = 0 \qquad (2.7)$$

$$\frac{\partial \phi'}{\partial t} + \frac{\partial(\phi' u)}{\partial x} + \frac{\partial(\phi' v)}{\partial y} + \frac{\partial(\phi u')}{\partial x} + \frac{\partial(\phi v')}{\partial y} = 0, \qquad (2.8)$$

where the prime denotes a perturbation around the basic-state variables.

In operator form, the discretized (2.6)–(2.8) can be written as (see Navon et al. 1992)

$$\mathbf{x}'(t) = \mathbf{P}(t)\mathbf{x}'(t_0), \qquad (2.9)$$

where $\mathbf{P}(t)$ represents the result of applying all the operator tor matrices in the linear model to obtain $\mathbf{x}'(t)$ from $\mathbf{x}'(t_0)$:

$$\mathbf{x}'(t) = (u', v', \phi')^{\mathrm{T}}, \qquad (2.10)$$

where the superscript T notation stands for the transposition of a matrix; $\mathbf{x}'(t)$ is an $N$-component vector of forecast variables that contains the values of two components of the velocity field and the geopotential field at all nodes at a given time $t$.

### *b. The Galerkin finite-element algorithm of the shallow-water equations*

Linear piecewise polynomials on triangular elements are used in the finite-element Galerkin discretization. The representation of a variable $\xi$ can be written as

$$\xi_{\mathrm{el}} = \sum_{j=1}^{3} \xi_j(t)V_j(x, y), \qquad (2.11)$$

where $\xi_j$ represents the scalar node value of variable $\xi$ at the nodes of the triangular element, and $V_j$ represents a basis function (interpolation function) defined by the coordinates of the nodes. Here a Galerkin formulation with the Einsteinian notation is used; for example, a repeated index implies summation with respect to that index. The notation used is

$$\langle f(x, y), V_i \rangle = \sum_{\text{elements}}^{m} \iint f(x, y)V_i\,dxdy = \iint_{\text{global}} f(x, y)V_i\,dxdy, \qquad (2.12)$$

which defines the inner product when a function is multiplied by a trial function.

A time-extrapolated Crank–Nicolson time-differencing scheme was applied for integrating in time the system of ordinary differential equations resulting from the application of the Galerkin finite-element method (see Navon 1979).

Using linear piecewise polynomials on triangular elements, the linearized continuity equation is solved first in the tangent-linear model given by (2.6)–(2.8) (see Navon 1979a, 1979b, 1987):

$$\left(\mathbf{M} - \frac{\Delta t}{2}\tilde{\mathbf{K}}_1\right)(\phi_j^{n+1} - \phi_j^n) - \Delta t(\Phi_{1j} + \Phi_{2j}) = 0, \qquad (2.13)$$

where the tilde represents the basic-state variable, $n$ is the time level ($t_n = n\Delta t$), and $\Phi_{1j}$ and $\Phi_{2j}$ are given by

$$\Phi_{1j} = \tilde{\mathbf{K}}_1 \phi_j^n + \mathbf{K}_1 \tilde{\phi}_j^n \quad (2.14)$$

$$\Phi_{2j} = \frac{1}{2} \mathbf{K}_1 (\tilde{\phi}_j^{n+1} - \tilde{\phi}_j^n); \quad (2.15)$$

**M** is the mass matrix given by

$$\mathbf{M} = \iint_{\omega_{\rm el}} V_i V_j d\omega, \quad (2.16)$$

where $\omega_{\rm el}$ is the area of triangular element, the matrix $\mathbf{K}_1$ gives rise also to a 3 × 3 element matrix, and, only after the assembly process, the global $N \times N$ matrices are obtained, whose expression is given by

$$\mathbf{K}_1 = \iint_{\omega_{\rm el}} V_j V_k u_k^* \frac{\partial V_i}{\partial x} d\omega + \iint_{\omega_{\rm el}} V_j V_k v_k^* \frac{\partial V_i}{\partial y} d\omega \quad (2.17)$$

$$\tilde{\mathbf{K}}_1 = \iint_{\omega_{\rm el}} V_j V_k \tilde{u}_k^* \frac{\partial V_i}{\partial x} d\omega + \iint_{\omega_{\rm el}} V_j V_k \tilde{v}_k^* \frac{\partial V_i}{\partial y} d\omega, \quad (2.18)$$

where $u_k^*$ and $v_k^*$ are given by

$$u_k^* = \frac{3}{2} u_k^n - \frac{1}{2} u_k^{n-1} + O(\Delta t^2) \quad (2.19)$$

$$v_k^* = \frac{3}{2} v_k^n - \frac{1}{2} v_k^{n-1} + O(\Delta t^2); \quad (2.20)$$

$u_k^*$ and $v_k^*$ result from a time-extrapolated Crank–Nicolson time-differencing method (see Wang et al. 1972; Douglas and Dupont 1970), which is used to quasi-linearize the nonlinear advective terms.

The $u$ and $v$ momentum equations are obtained as

$$\left(\mathbf{M} + \frac{\Delta t}{2} \tilde{\mathbf{K}}_2\right)(u_j^{n+1} - u_j^n) + \Delta t(\tilde{\mathbf{K}}_2 u_j^n + \mathbf{K}_2 \tilde{u}_j^n) + \frac{\Delta t}{2} \mathbf{K}_2 (\tilde{u}_j^{n+1} - \tilde{u}_j^n) + \frac{\Delta t}{2} (\mathbf{K}_{21}^{n+1} + \mathbf{K}_{21}^n) + \Delta t \mathbf{P}_2 = 0, \quad (2.21)$$

where **M** is the same mass matrix as (2.16), while the matrices $\mathbf{K}_2$, $\tilde{\mathbf{K}}_2$, $\mathbf{K}_{21}$, and $\mathbf{P}_2$ are given, respectively, by

$$\mathbf{K}_2 = \iint_{\omega_{\rm el}} u_k^* V_i V_k \frac{\partial V_j}{\partial x} d\omega + \iint_{\omega_{\rm el}} v_k^* V_i V_k \frac{\partial V_j}{\partial y} d\omega \quad (2.22)$$

$$\tilde{\mathbf{K}}_2 = \iint_{\omega_{\rm el}} \tilde{u}_k^* V_i V_k \frac{\partial V_j}{\partial x} d\omega + \iint_{\omega_{\rm el}} \tilde{v}_k^* V_i V_k \frac{\partial V_j}{\partial y} d\omega \quad (2.23)$$

$$\mathbf{K}_{21}^{n+1} = \iint_{\omega_{\rm el}} \phi_k^{n+1} V_i \frac{\partial V_k}{\partial x} d\omega \quad (2.24)$$

TABLE 1. Transposition of matrix LOCAT(6, 180), all the locations of nonzero elements in Eq. (2.32) are stored using the compact storage scheme for nodes $i$ ($i = 1, 2, \cdots, N$).

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 13 | 2 | 169 | 170 | 0 | 0 |
| 2 | 1 | 13 | 14 | 3 | 170 | 171 |
| 3 | 2 | 14 | 15 | 4 | 171 | 172 |
| ··· | ··· | ··· | ··· | ··· | ··· | ··· |
| 11 | 10 | 22 | 23 | 12 | 179 | 180 |
| 12 | 11 | 23 | 24 | 180 | 0 | 0 |
| 13 | 1 | 2 | 14 | 25 | 0 | 0 |
| 14 | 2 | 13 | 3 | 15 | 25 | 26 |
| 15 | 3 | 14 | 4 | 16 | 26 | 27 |
| ··· | ··· | ··· | ··· | ··· | ··· | ··· |
| 90 | 88 | 99 | 89 | 101 | 111 | 112 |
| 91 | 89 | 100 | 90 | 102 | 112 | 113 |
| 92 | 90 | 101 | 91 | 103 | 132 | 114 |
| ··· | ··· | ··· | ··· | ··· | ··· | ··· |
| 169 | 157 | 158 | 170 | 1 | 0 | 0 |
| 170 | 158 | 169 | 159 | 171 | 1 | 2 |
| 171 | 159 | 170 | 160 | 172 | 2 | 3 |
| ··· | ··· | ··· | ··· | ··· | ··· | ··· |
| 178 | 166 | 177 | 167 | 179 | 9 | 10 |
| 179 | 167 | 178 | 168 | 180 | 10 | 11 |
| 180 | 168 | 179 | 11 | 12 | 0 | 0 |

$$\mathbf{P}_2 = -\iint_{\omega_{\rm el}} f v_k^* V_k V_i d\omega \quad (2.25)$$

$$\left(\mathbf{M} + \frac{\Delta t}{2} \tilde{\mathbf{K}}_3\right)(v_j^{n+1} - v_j^n) + \Delta t(\tilde{\mathbf{K}}_2 v_j^n + \mathbf{K}_2 \tilde{v}_j^n) + \frac{\Delta t}{2} \mathbf{K}_2 (\tilde{v}_j^{n+1} - \tilde{v}_j^n) + \frac{\Delta t}{2} (\mathbf{K}_{31}^{n+1} + \mathbf{K}_{31}^n) + \Delta t \mathbf{P}_3 = 0, \quad (2.26)$$

where $\mathbf{K}_3$, $\tilde{\mathbf{K}}_3$, $\mathbf{K}_{31}$, and $\mathbf{P}_3$ are given, respectively, by

$$\mathbf{K}_3 = \iint_{\omega_{\rm el}} u_k^{n+1} V_i V_k \frac{\partial V_j}{\partial x} d\omega + \iint_{\omega_{\rm el}} v_k^* V_i V_k \frac{\partial V_j}{\partial y} d\omega \quad (2.27)$$

$$\tilde{\mathbf{K}}_3 = \iint_{\omega_{\rm el}} \tilde{u}_k^{n+1} V_i V_k \frac{\partial V_j}{\partial x} d\omega + \iint_{\omega_{\rm el}} \tilde{v}_k^* V_i V_k \frac{\partial V_j}{\partial y} d\omega \quad (2.28)$$

$$\mathbf{K}_{31}^{n+1} = \iint_{\omega_{\rm el}} \phi_k^{n+1} V_i \frac{\partial V_k}{\partial y} d\omega \quad (2.29)$$

$$\mathbf{P}_3 = \iint_{\omega_{\rm el}} f u_k^{n+1} V_k V_i d\omega. \quad (2.30)$$

The explicit form of matrices computed along with the compact storage scheme is detailed in appendix A. For recent applications of the finite-element model to the shallow-water equations, see Navon (1983), Steppeler et al. (1990), Priestley (1992), and Neta (1992). For operational applications, see Staniforth and Mitchell (1977, 1978), Staniforth and Daley (1979), Beland et al. (1983), and Silvester (1969). For comprehensive reviews of the application of the finite-element model to the shallow-water equations, see an early review by Navon (1977), Staniforth (1984, 1987), Navon (1988), and Neta (1992).

The boundary treatment in the tangent linear model is the same as that presented by Navon (1979).

### *c. The Gauss–Seidel iterative method for the linear system of the finite-element equations model*

The solution of the resulting nonsymmetric systems of linear algebraic equations is obtained by a Gauss–Seidel iterative method where the system of algebraic equations is given in matrix form as

$$\mathbf{A}\boldsymbol{\xi} = \mathbf{b}; \qquad (2.31)$$

that is,

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1j} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2j} & \cdots & a_{2N} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{i1} & a_{i2} & \cdots & a_{ij} & \cdots & a_{iN} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{N1} & a_{N2} & \cdots & a_{Nj} & \cdots & a_{NN} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \\ \cdots \\ \xi_i \\ \cdots \\ \xi_N \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \cdots \\ b_i \\ \cdots \\ b_N \end{pmatrix}, \qquad (2.32)$$

where $\xi_i$ represents either $u_i$, or $v_i$, or $\phi_i$, respectively. The Gauss–Seidel iteration assumes the form

$$\xi_i^{(k)} = \sum_{j=1}^{i-1} c'_{ij}\xi_j^{(k)} + \sum_{j=i+1}^{N} c'_{ij}\xi_j^{(k-1)} + d_i \qquad (i = 1, 2, \cdots, N), \qquad (2.33)$$

where $k$ stands for the iteration count, while $c_{ij}$ and $d_i$ are

$$c'_{ij} = -\frac{a_{ij}}{a_{ii}}, \quad d_i = \frac{b_i}{a_{ii}}, \quad (i, j = 1, 2, \cdots, N). \qquad (2.34)$$

We see from (2.33) that the $k$th values of the variables are used as soon as they are available.

In the finite-element equations model, a compact storage scheme was adopted that is based on the fact that the maximum number of triangles supporting any node is six. Each row $i$ in the $N \times N$ global matrix (here corresponding to the matrix $\mathbf{A}$) represents the equations written for the point $i$ and will have at most seven entries. In other words, there are at most seven nonzero elements at each row of the matrix. Thus, to reduce the $N \times N$ global matrix into an $N \times 7$ condensed matrix, a correlation address matrix is used whose size is $N \times 7$. The location of nonzero elements was recorded in a matrix of dimension $(N, 7)$, which will be denoted LOCAT$(j, i)$ (see Table 1). Now, (2.33) becomes

$$\xi_i^{(k)} = \sum_{j=1}^{i-1} c_{ij}\xi_{L_{ji}}^{(k)} + \sum_{j=i+1}^{6} c_{ij}\xi_{L_{ji}}^{(k-1)} + d_i \qquad (i = 1, 2, \cdots, N), \qquad (2.35)$$

where $c_{ij}$ represents the element of the global matrix that had been stored by the compact scheme, where $\xi$ represents either of the $u$, $v$, or $\phi$ variables, respectively, and

$$L_{ji} = \text{LOCAT}(j, i) \quad (j = 1, 2, \cdots 6; i = 1, 2, \cdots N).$$

We see from (2.35) that the subscript of the variable $\xi$ results in a nonsequential storage location due to the adoption of the compact storage scheme.

## 3. The adjoint of the finite-element model

### *a. The cost function and its gradient*

We define a cost function whose gradient with respect to the control variable is found by integrating the adjoint model backward in time and forcing its right-hand side with the difference between the direct model forecast and the observations when an observation distributed in space and time is encountered. The cost function is defined as the weighted squared difference between the model solution and observations over the interval (time window) of assimilation:

$$J[\mathbf{X}(t_0)] = \frac{1}{2}\sum_{r=0}^{R} \langle \mathbf{W}(t_r)[\mathbf{X}(t_r) - \mathbf{X}_{\text{obs}}(t_r)], \mathbf{X}(t_r) - \mathbf{X}_{\text{obs}}(t_r)\rangle, \qquad (3.1)$$

where $J[\mathbf{X}(t_0)]$ is a cost function depending on the initial conditions $\mathbf{X}(t_0)$, $\mathbf{X}_{\text{obs}}(t_r)$ are the observed value of the model variables at time $t_r$, $R$ is the number of time levels where observations are available in the assimilation interval, $\langle\, , \rangle$ denotes the inner product of two vectors in a given norm (usually the Euclidean norm), and $\mathbf{W}(t_r)$ is a weighting factor that is taken to be the inverse of the statistical root-mean-square observational errors on geopotential and wind components, respectively. In our test problem, the diagonal entries of $\mathbf{W}$ are taken to be constant values: namely, $W_\phi = 10^{-4}\ \text{m}^{-2}\ \text{s}^2$ for geopotential component, and $W_v = 10^{-2}\ \text{m}^{-2}\ \text{s}^2$ for the velocity field components.

The VDA problem consists in minimizing the cost function (3.1) by using large-scale unconstrained minimization algorithms such as the quasi-Newton algorithm and the limited-memory Q-N algorithm L-BFGS (limited-memory Broyden–Fletcher–Goldfarb–Shanna) of Liu and Nocedal (1989) that require the gradient of the

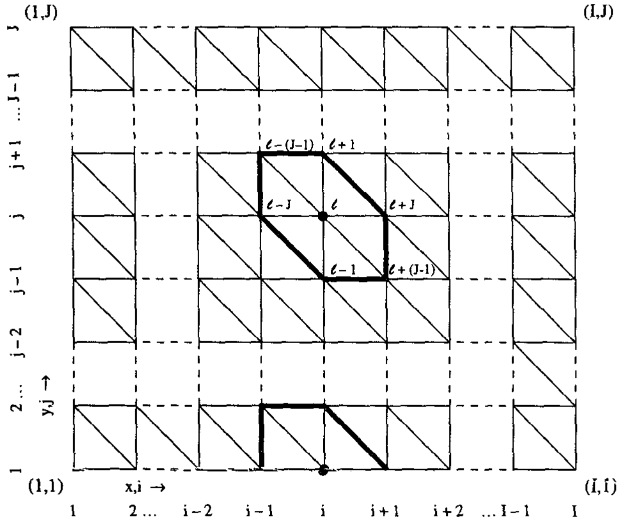

FIG. 1. The domain is subdivided by triangular right-angled elements, the coarse solid lines denoting a variable value at a node $l$, which is related to variable values at the other six neighboring nodes (inner nodes) or four nodes (boundary nodes) in different directions: $l = j + (i - 1)J; j = 1, 2, \cdots, J; i = 1, 2, \cdots, I; N = IJ$.

cost function with respect to control variables. The gradient of the cost function with respect to the initial condition can be found using the adjoint of the nonlinear finite-element model (see Navon et al. 1992) as follows:

$$\nabla J[\mathbf{X}(t_0)] = \sum_{r=0}^{R} \hat{\mathbf{X}}^r(t_0), \qquad (3.2)$$

where $\hat{\mathbf{X}}^r(t_0)$ is the solution of the following adjoint model:

$$\hat{\mathbf{X}}^r(t_0) = \mathbf{P}_r^{\mathrm{T}}\mathbf{W}(t_r)[\mathbf{X}(t_r) - \mathbf{X}_{\mathrm{obs}}(t_r)], \qquad (3.3)$$

where $\mathbf{P}_r^{\mathrm{T}}(t)$ represents the adjoint of $\mathbf{P}_r(t)$ [see (2.9)]. For instance, if we assume that the linear model is the result of the multiplication of a number of operator matrices,

$$\mathbf{P}_r = \mathbf{Q}_1^r\mathbf{Q}_2^r \cdots \mathbf{Q}_K^r\mathbf{P}_{r-1}, \qquad (3.4)$$

where each matrix $\mathbf{Q}_i^r$ ($i = 1, 2, \cdots K$) stands for either a subroutine or a DO loop, then the adjoint model will be the product of the following adjoint subproblems:

$$\mathbf{P}_r^{\mathrm{T}} = \mathbf{P}_{r-1}^{\mathrm{T}}\mathbf{Q}_K^{r\mathrm{T}}\mathbf{Q}_{K-1}^{r\mathrm{T}} \cdots \mathbf{Q}_1^{r\mathrm{T}}. \qquad (3.5)$$

Since both (3.2) and (3.3) are linear, the gradient of the cost function with respect to the initial condition can be obtained by integrating the adjoint model backward in time from $t_R$ to $t_0$ with zero initial conditions at time $t_R$ while a forcing term

$$\mathbf{W}[\mathbf{X}(t_r) - \mathbf{X}_{\mathrm{obs}}(t_r)] \qquad (3.6)$$

is added to the right side of the adjoint model equations whenever observations are encountered at time $t_r$, $r = R, R - 1, \cdots, 1, 0$.

### b. *The adjoint of the iterative procedure*

The development of the adjoint model of the finite-element shallow-water equations model involves overcoming issues related to building the adjoint of the Gauss–Seidel iterative procedure as well as the treatment of the irregular subscripts of variables required to solve the nonsymmetric systems of linear equations (see section 2c). In the finite-element model the variables are discretized on an area of compact support [a hexagon with implicit gridpoint index (see Fig. 1)] rather than on a line with explicit gridpoint index, and some of the variables to be solved at the current iteration level are reused, while some are not reused in the Gauss–Seidel iterative procedure. For example, the Gauss–Seidel iteration formula for the inner nodes $l$ ($j \neq 1, 2, J$; $i \neq 1, I$, where $I$ is the total number of the grid points in the $x$ direction and $J$ is the total number of the grid points in the $y$ direction and $N = IJ$) at the $k$th iteration (excluding the last iteration) is

$$\xi_l^{(k)} = c_{l,1}\xi_{L_{1,l}}^{(k)} + c_{l,2}\xi_{L_{2,l}}^{(k)} + c_{l,3}\xi_{L_{3,l}}^{(k)} + c_{l,4}\xi_{L_{4,l}}^{(k-1)} + c_{l,5}\xi_{L_{5,l}}^{(k-1)} + c_{l,6}\xi_{L_{6,l}}^{(k-1)} + d_l, \qquad (3.7)$$

where

$$l = j + (i - 1)J, \quad j = 3, 4, \cdots, J - 1, \quad i = 2, 3, \cdots, I - 1. \qquad (3.8)$$

Following Table 1 and Fig. 1, formula (3.7) can be written as

$$\xi_l^{(k)} = c_{l,1}\xi_{l-J}^{(k)} + c_{l,2}\xi_{l-1}^{(k)} + c_{l,3}\xi_{l-(J-1)}^{(k)} + c_{l,4}\xi_{l+1}^{(k-1)} + c_{l,5}\xi_{l+(J-1)}^{(k-1)} + c_{l,6}\xi_{l+J}^{(k-1)} + d_l. \qquad (3.9)$$

We see from Fig. 1 that the variable $\xi_{l-J}^{(k)}$ will be reused at the next iteration when $\xi_{l-(2J)}^{(k+1)}$ is iteratively solved; $\xi_{l-1}^{(k)}$ and $\xi_{l+(J-1)}^{(k-1)}$ will be reused when $\xi_{l+(J-2)}^{(k)}$ is iteratively solved; $\xi_{l-(J-1)}^{(k)}$ will be reused at following step $l' = l + 1$, that is, when $\xi_{l+1}^{(k)}$ is iteratively solved; $\xi_{l+1}^{(k-1)}$ is not reused; $\xi_{l+J}^{(k-1)}$ will be reused when $\xi_{l+(J-1)}^{(k)}$ is iteratively solved.

In short, only the $\xi_{l+1}^{(k-1)}$ is not reused while all the rest of the variables in formula (3.9) are reused. For the reused variable values, only $\xi_{l-J}^{(k)}$ will be reused at the next iteration while all the others will be reused at same iteration. [For the difference in writing the adjoint code when a variable is reused or not, see Navon et al. (1992, appendix A).] Therefore, the corresponding adjoint model counterparts of the Gauss–Seidel iteration given by (3.9) assume the following form:

$$\begin{aligned}
\hat{\xi}_{l-J}^{(k)} &= c_{l,1}\hat{\xi}_l^{(k)} + \hat{\xi}_{l-J}^{(k)} \\
\hat{\xi}_{l-1}^{(k)} &= c_{l,2}\hat{\xi}_l^{(k)} + \hat{\xi}_{l-1}^{(k)} \\
\hat{\xi}_{l-(J-1)}^{(k)} &= c_{l,3}\hat{\xi}_l^{(k)} + \hat{\xi}_{l-(J-1)}^{(k)} \\
\hat{\xi}_{l+1}^{(k-1)} &= c_{l,4}\hat{\xi}_l^{(k)} \\
\hat{\xi}_{l+(J-1)}^{(k-1)} &= c_{l,5}\hat{\xi}_l^{(k)} + \hat{\xi}_{l+(J-1)}^{(k-1)} \\
\hat{\xi}_{l+J}^{(k-1)} &= c_{l,6}\hat{\xi}_l^{(k)} + \hat{\xi}_{l+J}^{(k-1)},
\end{aligned} \qquad (3.10)$$

where the caret represents the adjoint variable.

The key issues related to developing the adjoint of the Gauss–Seidel iterative procedure are the following.

First, the maximum number of iterations $K$ must be recorded when the forward model is integrated, and second, in order to obtain the adjoint of the Gauss–Seidel iterative procedure the relationship between different nodes must be analyzed. (For details of the analysis and the corresponding adjoint model formulas for the Gauss–Seidel iterations on other nodes, see appendix B.)

In summary, the maximum number of triangles supporting any node is six, while the minimum number is four (boundary triangle), when the piecewise linear triangular Galerkin finite-element scheme for the shallow-water equations is used. The variable value at any given inner node is related to the values at six neighboring nodes surrounding it, and the variables in the Gauss–Seidel iteration play both input and output roles. Thus, when a variable value is iteratively solved, some inputs in the iterative formulas are reused while others are not, depending both on the position of the variable as well as on the level of iterations.

### c. *Verification of the correctness of the adjoint model*

The correctness of the adjoint model can be checked by requiring the following equality to be satisfied:

$$(\mathbf{PX})^{\mathrm{T}}(\mathbf{PX}) = \mathbf{X}^{\mathrm{T}}[\mathbf{P}^{\mathrm{T}}(\mathbf{PX})], \qquad (3.11)$$

where $\mathbf{P}$ represents the code of the tangent linear finite-element model while $\mathbf{X}$ represents the initial conditions of the tangent linear model, and $\mathbf{P}^{\mathrm{T}}(\mathbf{PX})$ denotes the backward integration of the adjoint model. We have verified equality (3.11) for each subroutine up to machine accuracy.

We observe that the adjoint check given by (3.11) depends only on the tangent linear model. To ensure the correctness of the adjoint model with respect to the original nonlinear model, we also need to check the correctness of the tangent linear model. We used two methods to check the correctness of the tangent linear model version of the finite-element model. First, for a state vector $\mathbf{X}$ and a perturbation $\mathbf{X}'$, the following formula derived from a Taylor's series expansion should be verified (see Thépaut and Courtier 1991):

$$\Psi(\alpha) = \frac{\mathbf{G}(\mathbf{X} + \alpha\mathbf{X}')_i - \mathbf{G}(\mathbf{X})_i}{\alpha\mathbf{L}(\mathbf{X}')_i} = 1 + O(\alpha),$$
$$i = 1, 2, \cdots, N, \qquad (3.12)$$

where $\mathbf{G}$ denotes the nonlinear finite-element model operator, $\mathbf{L}$ represents the tangent linear model operator resulting from the linearization of the nonlinear finite-element model, and the subscript $i$ denotes the $i$th component of a vector.

Both the nonlinear as well as the linear models were integrated for a 10-h period with various $\alpha$ values, and a good approximation of the evolution of perturbations for the tangent linear model were obtained.

We also tested the accuracy of the gradient of the cost function. The check formula is

$$F(\alpha) = \frac{J(\mathbf{X} + \alpha\nabla J) - J(\mathbf{X})}{\langle\alpha\nabla J, \nabla J\rangle} = 1 + O(\alpha), \qquad (3.13)$$

where $\langle\ ,\ \rangle$ denotes the inner product. For a small $\alpha$ that is not too close to the machine accuracy, the value of $F(\alpha)$ must be close to 1 if the gradient $\nabla J$ is correct (see Courtier 1987). A value of $F(\alpha)$ is obtained that is close to unity with increasing accuracy as the values of $\alpha$ vary between $10^{-2}$ and $10^{-12}$ (see Fig. 2).

## 4. Numerical experiments

### a. *The test problem*

To test the ability of carrying out VDA with the finite-element model, we test a problem related to the limited-area shallow-water equations model, using piecewise linear triangular elements to discretize by a Galerkin finite-element method the equations over a

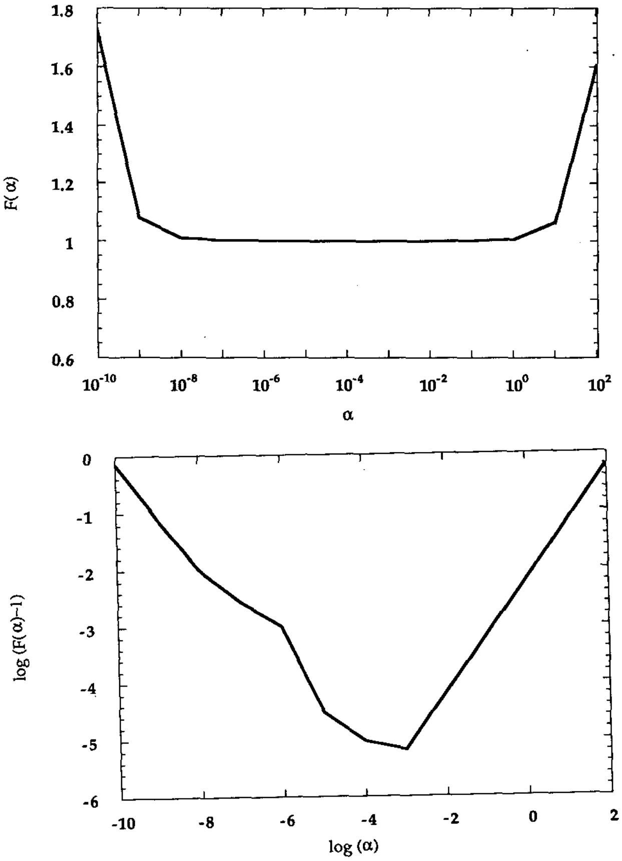

FIG. 2. Verification of the gradient calculation: (a) variation of $F(\alpha)$ with respect to $\alpha$ and (b) variation of $\log(F - 1)$ with respect to $\log(\alpha)$.

limited-area domain that consists of a cylindrical channel on a latitude belt with north–south walls. Cyclic continuity is assumed in the $x$ direction. An initial height field condition of Grammeltvedt (1969) was used. This initial condition is given by

$$h(x, y) = H_0 + H_1 \tanh \frac{9[(D/2) - y]}{2D} + H_2 \operatorname{sech}^2 \frac{9[(D/2) - y]}{D} \sin \frac{2\pi x}{L}. \quad (4.1)$$

The initial velocity field components were derived from the initial height field by means of the geostrophic relationship; that is,

$$u = -\frac{g}{f}\frac{\partial h}{\partial y}$$

$$v = \frac{g}{f}\frac{\partial h}{\partial x}. \quad (4.2)$$

The constants used are given as

$$L = 6000 \text{ km}, \quad g = 10 \text{ m s}^{-2}$$
$$D = 4400 \text{ km}, \quad \hat{f} = 10^{-4} \text{ s}^{-1}$$
$$\beta = 1.5 \times 10^{-11} \text{ s}^{-1} \text{ m}^{-1}$$
$$H_0 = 2000 \text{ m} \quad H_1 = 220 \text{ m}$$
$$H_2 = 133 \text{ m}. \quad (4.3)$$

The time and space increments are

$$\Delta x = \Delta y = 400 \text{ km}, \quad \Delta t = 1800 \text{ s}. \quad (4.4)$$

The boundary conditions used in the numerical integrations are cyclic in the $x$ direction and solid wall for the normal component of velocity in the $y$ direction, shown as follows:

$$v|_{y=0} = v|_{y=D} = 0,$$
$$u|_{x=0} = u|_{x=L},$$
$$v|_{x=0} = v|_{x=L},$$
$$\phi|_{x=0} = \phi|_{x=L}. \quad (4.5)$$

The model was run with a resolution of $(I \times J = 15 \times 12)$ nodal mesh points, using a regular grid domain.

In the following experiments, observational data consists of model-integrated values of wind and geopotential at each time step starting from the Grammeltvedt initial condition defined by (4.1) and (4.2) (see Fig. 3). Random perturbations of these fields, obtained by using a standard library randomizer RANF on the CRAY-YMP machine, serve as the initial guess (Fig. 4) for the solution. The length of the assimilation window is 20 time steps (10 h).

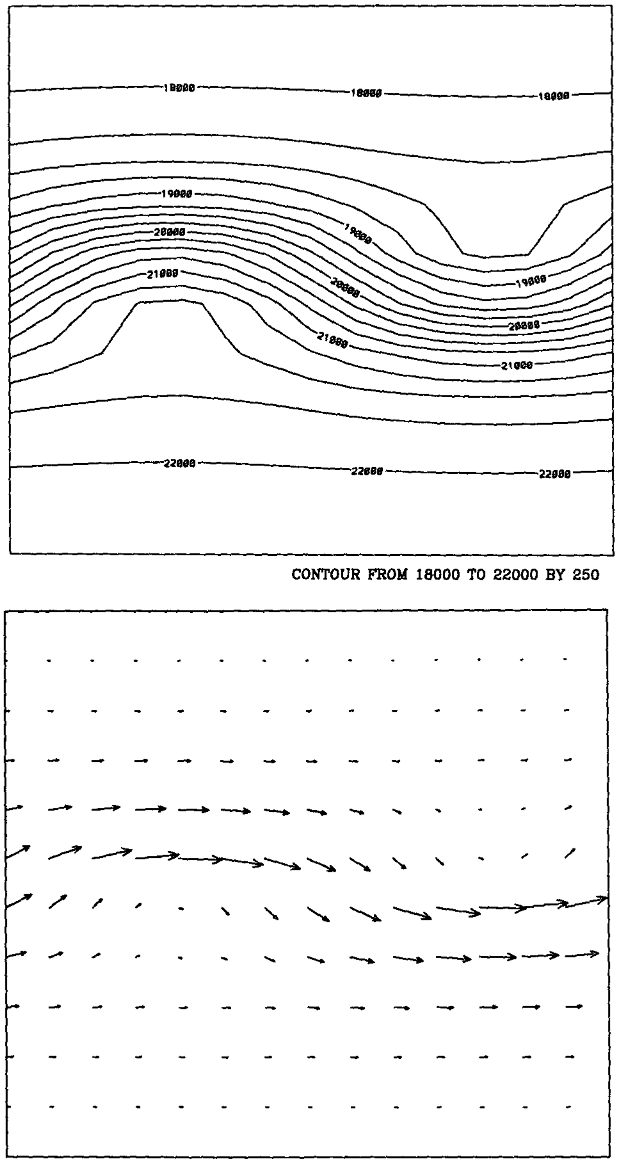

FIG. 3. (a) Geopotential field for the Grammeltvedt initial condition. (b) Wind field calculated from the geopotential field by the geostrophic approximation.

### b. *Numerical experiments and results for the control of initial conditions*

One of the difficulties related to successfully implementing large-scale nonlinear unconstrained optimization problems resides in choosing adequate scaling factors. It is well known that a badly scaled nonlinear programming problem results in slow convergence due to the distortion of the ellipsoid generated (Navon and de Villiers 1983; Courtier and Talagrand 1990). The general form of a scaling procedure is

$$\mathbf{X}^s = \mathbf{SX} \quad (4.6)$$

$$\mathbf{g}^s = \mathbf{Sg}, \quad (4.7)$$

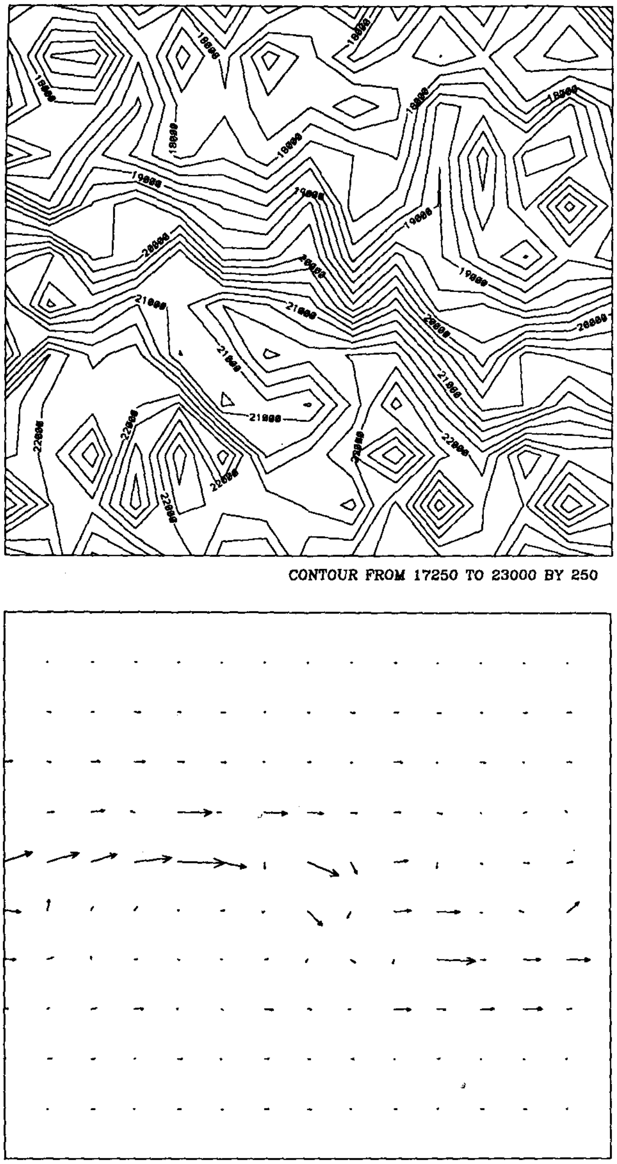

FIG. 4. Random perturbation of the (a) geopotential field and (b) wind field.

where **S** is a diagonal scaling matrix and **X** and **g** are the state variable and the gradient, respectively. In the first experiment, the velocity and geopotential fields are scaled by the factors $10^4$ and $10^0$, respectively.

Computations were carried out on the CRAY-YMP supercomputer at the Supercomputer Computations Research Institute at The Florida State University. Random perturbations of (4.1) and (4.2) were used as the initial guess of the integration of the finite-element model. In all following experiments, the limited-memory quasi-Newton method of Liu and Nocedal (1989) was used to minimize the cost function consisting of the weighted sum of the difference between model solution and observations (3.1). The adjustment was performed on the 10-h interval $[t_0, t_R]$ (window of assimilation) preceding $t_R$. The adjoint model is integrated backward in time, with a forcing term (3.6) being added to the adjoint variable at every time step when an observation is encountered. When a prescribed convergence criterion is attained, the minimization process was terminated. The convergence criterion applied for the numerical experiments was

$$\|\mathbf{g}_k\| \leq \epsilon \max\{1, \|\mathbf{X}_k\|\}, \qquad (4.8)$$

where $\mathbf{g}_k$ is the gradient of the cost function at the $k$th iteration and $\epsilon = 10^{-11}$ was chosen to obtain an accurate solution.

In the first experiment, a VDA was performed in which a scaled cost function was used. The minimization iteration process converged without detecting any errors after 64 iterations and 83 function calls. In Fig. 5 we present the variation of the value of the normalized cost function (solid line) and the normalized gradient of the cost function (dotted line) with the number of iterations in the unconstrained minimization process. After 60 iterations, the cost function was reduced by about three orders of magnitude. The norm of the gradient was reduced by two orders of magnitude while the prescribed convergence criterion was met.

After 64 iterations, a satisfactory solution with an almost perfect retrieval was obtained. It is well known, that in the case of random perturbations one should expect to retrieve the unperturbed initial conditions. The differences of geopotential fields between the retrieved and the unperturbed fields are shown in Fig. 6. The error is reduced by more than two orders of magnitude. This clearly shows the capability of the variational data assimilation method using the adjoint model to adjust the model solution to a set of "observations" distributed in both time and space.

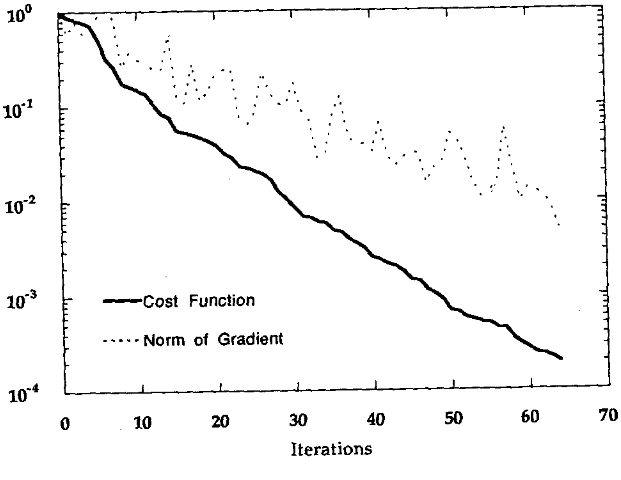

FIG. 5. Variation of value of the normalized scaled cost function $J/J_0$ (solid line) and the norm of gradient $\|\nabla J_k\|/\|\nabla J_0\|$ (dotted line) with the number of iterations.

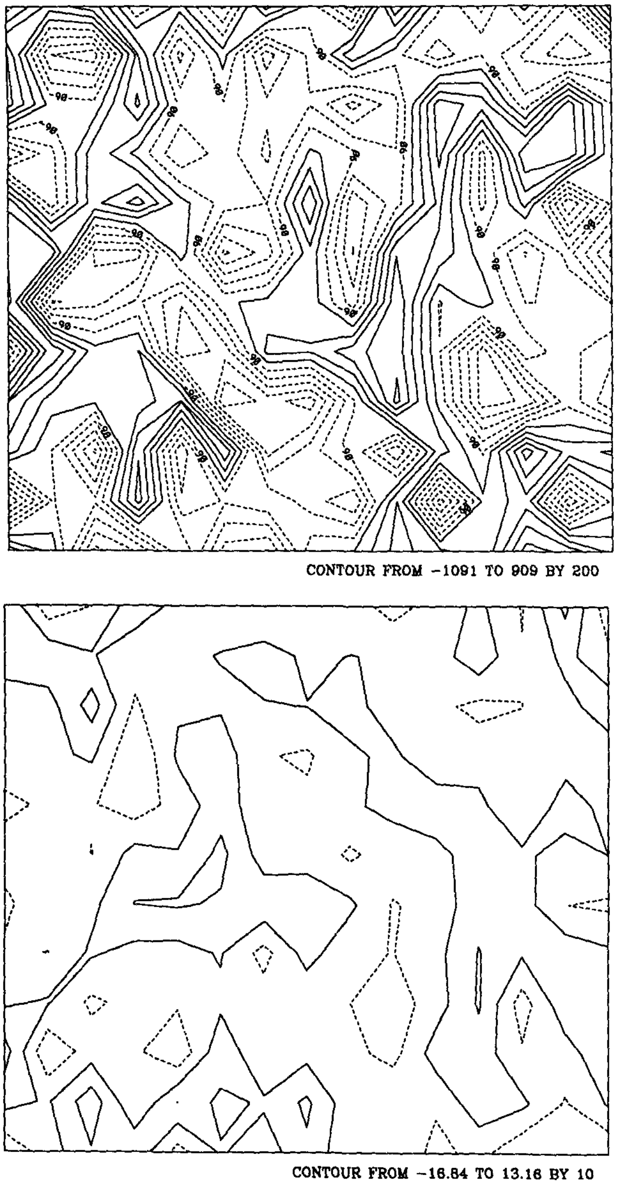

FIG. 6. Retrieved initial (a) geopotential field (Fig. 6a) and (b) wind field using proper scaling.

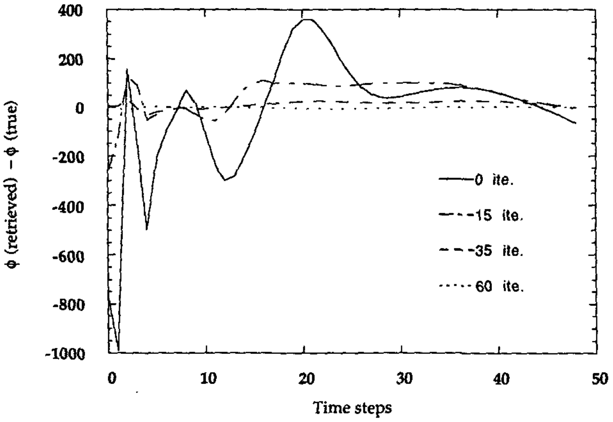

FIG. 7. The differences of time variations of the geopotential field at a grid point between the initial guess and true solution (solid), the retrieved initial conditions after 15 (dash–dot), 35 (dashed), and 60 iterations (dotted), and the true solution.

To assess the accuracy of the retrieved initial conditions, we display the maximum absolute difference between the retrieved initial conditions and observational initial conditions and the difference between the initial guess and observational initial conditions in Table 2. We see that the maximum differences of both the geopotential and wind fields were reduced by about two orders of magnitude during the VDA retrieval.

To illustrate the issue that the VDA minimization procedure can be stopped prior to the cost function achieving its asymptotic rate of decrease, we present in Fig. 7 the differences of the time integrations of the geopotential field at a grid point started from the initial guess and true solution (solid line); the VDA retrieved initial conditions after 15 (dash–dot line), 35 (dashed line), and 60 (dotted line) iterations; and the true solution, respectively. We note that even after 15 iterations, when the cost function and the norm of the gradient have decreased only one order of magnitude, the difference between time integrations started from the true solution and the retrieved initial conditions becomes much smaller than that between the true solution and the guess field. After 30 iterations, the difference is very small. After 60 iterations, there is practically no difference between the time integrations started from the true solution and the retrieved initial conditions. Therefore, we conclude that the VDA achieves most of the large-scale retrieval for the shallow-water finite-el-

TABLE 2. Maximum absolute differences between the perturbed and the unperturbed, the retrieval and the unperturbed initial wind and geopotential fields.

| | $\max(u'^2 + v'^2)^{1/2}$ | $\max\lvert\phi'\rvert$ |
|---|---|---|
| $X_{\rm per} - X_{\rm obs}$ | 0.1606E+2 | 0.1012E+4 |
| $X_{\rm ret} - X_{\rm obs}$ | 0.7428E0 | 0.1762E+2 |

TABLE 3. CPU time used by the finite-element shallow-water equations, its tangent linear model, and adjoint model for a 10-h time integration, the cost function and its gradient with respect to the initial condition. The time step is 1800 s.

| | CPU time (s) |
|---|---|
| Nonlinear model | 0.5120 |
| Linear model | 0.7429 |
| Adjoint model | 1.4835 |
| Cost function $J$ | 0.5123 |
| Gradient $\nabla J$ | 1.5997 |

ement model during the first 15–30 minimization iterations, while in the latter part of the minimization process only small-scale features are being assimilated.

An estimate of the computational cost required for calculating the gradient of the cost function using the adjoint technique was obtained. Table 3 displays the CPU time used by the finite-element model, its tangent-linear model, its adjoint model, and the gradient of the cost function using the adjoint technique for a 10-h time integration that spans the window of assimilation. The time step used is 1800 s. The values of the state variables used in the tangent and adjoint models were stored in memory during the nonlinear model integration to avoid recomputation in the adjoint. We observe that the integration of the nonlinear finite-element model requires only 0.512 s—the CPU time required by the tangent linear model integration being longer than that of the nonlinear model integration—and that the CPU time required for the integration of the adjoint model is the longest among the three aforementioned models, almost tripling the CPU time required for the integration of the nonlinear model. The high CPU computational cost of the adjoint model, compared to the tangent linear model, is due to the high cost incurred mainly in the Gauss–Seidel iterative algorithm where we need to calculate location of lower indices of the reused variables.

In the second experiment, we used the same model and method except that no scaling factors were used in the cost function and again we tested the retrieval of the initial conditions using random perturbations as the initial guess of the solution. The aim of this experiment is to assess the importance of using a good scaling procedure to ensure the success of the large-scale unconstrained minimization problem measured by a faster rate of convergence that translates into fewer numbers of iterations. It is known that if adequate scaling factors are not used to scale the cost function and its gradient, the minimization algorithms yield poor results characterized by a longer, slowly convergent minimization process, thus implying very costly CPU time requirements.

A VDA retrieval was performed in which the cost function was not scaled. Table 4 shows that for an identical prescribed convergence criterion (here, the maximum absolute differences were almost identical) to be met, the number of minimization iterations for the same problem differs dramatically from the case when the cost function was adequately scaled. More than 300 iterations were required for the nonscaled case to attain the same prescribed accuracy previously attained in 64 iterations when adequate scaling was used. When the number of minimization iterations reached the one for which the first scaled experiment converged (Nfun, the number of function calls, is then 87) the solution obtained at this iteration for the unscaled case was unsatisfactory. The solution of the geopotential field without scaling the cost function after 78 iterations is the same as Fig. 4a, that the geopotential field after 78 iterations resembles the perturbed field; that is, no successful retrieval was achieved.

TABLE 4. Maximum absolute differences between the retrieval and the unperturbed initial wind and geopotential fields by scaling and not scaling the gradient of the cost function.

| | Iterations | Nfun | $\max(u'^2 + v'^2)^{1/2}$ | $\max\|\phi'\|$ |
|---|---|---|---|---|
| Scaling | 64 | 87 | 0.7428E0 | 0.1762E+2 |
| Not scaling | 64 | 71 | 0.9371E+1 | 0.1010E+4 |
| Not scaling | 347 | 365 | 0.7493E0 | 0.1784E+2 |

## 5. Experiments with different horizontal mesh resolutions

In recent years, higher-resolution numerical prediction models have been used due to the availability of powerful computers with larger memories and faster CPU time. The results of high-resolution models may describe in more detail the realistic state of the atmosphere and result in reduced truncation and phase errors. However, both the availability of data as well as the required CPU time may still constitute a problem for high-resolution models.

If the results obtained from a coarse-mesh model can retrieve most of the information of a fine-mesh model by using VDA methods, the results may turn out to be useful in meteorology or oceanography.

In this section we first carry out experiments in which we use the VDA to control initial conditions in finite-element models with various horizontal resolutions. The purpose is to observe and compare the efficiency of VDA for variable horizontal resolution in finite-element shallow-water equation models and the transfer of information among different scales of motion. Then, we try to test and assess the impact of fitting observations obtained with different horizontal mesh resolutions using the VDA optimal control method.

### a. *The case of coarse-mesh versus fine-mesh horizontal resolution*

In our experiments we used a coarse-mesh model with 8 × 9 gridpoint resolution, while for the fine-mesh case we used a resolution of 20 × 21 grid points on the same domain as described in section 4a. In both cases the initial conditions as given by (4.1) and (4.2) were used while the boundary conditions were identical with those given by (4.5). In the coarse-mesh (8 × 9) model the time and space increments are

$$\Delta x = 750 \text{ km}$$

$$\Delta y = 550 \text{ km}$$

$$\Delta t = 1800 \text{ s}, \qquad (5.1)$$

respectively. In the fine-mesh (20 × 21) model the time and space increments are

$$\Delta x = 300 \text{ km}$$
$$\Delta y = 220 \text{ km}$$
$$\Delta t = 600 \text{ s}, \qquad (5.2)$$

respectively. The adjoint model is unchanged except for the variable number of mesh points. The length of the assimilation window is 10 h.

In the first experiments, by testing the VDA with various mesh resolution (8 × 9 and 20 × 21) models, we found out that by using the same scaling as the one (with 15 × 12 gridpoint resolution) used in section 4b, the minimization failed to converge for both the fine- and the coarse-mesh resolution models. For the fine-mesh model, the minimization stopped due to rounding errors after 11 iterations, and no reasonable retrieval was obtained. For the coarse-mesh model, after 27 iterations, the minimization terminated with an error message. However, when suitable scaling factors varying with the horizontal resolution were chosen (namely, the velocity and geopotential fields were scaled by factors $5 \times 10^4$ and $10^0$ for the coarse-mesh model, while for the fine-mesh model the scaling factors were taken to be $5.5 \times 10^4$ and $10^0$, respectively), the minimization was successful. The results of the minimization with observations available at every time step in the assimilation window are similar to results presented in Fig. 5. The evolution of the value of the normalized cost functions for three variable horizontal resolution meshes—namely, fine mesh 1 (20 × 21) (solid line), coarse mesh 1 (15 × 12) (dotted line), and coarse mesh 2 (8 × 9) (dashed line)—versus the number of iterations is displayed in Fig. 8. Figure 9 presents the variation of the value of the normalized gradient of the cost functions in three meshes—fine mesh 1 (20 × 21) (solid), coarse mesh 1 (15 × 12) (dotted), and coarse mesh 2 (8 × 9) (dashed)—versus the number of iterations. We observe from these figures that the rate of convergence of the cost function associated with the coarse mesh is faster than the rate of convergence corresponding to the fine-resolution models, the cost function associated with coarse-mesh resolution being reduced by more than four orders of magnitude, while the value of the norm of the gradient of the cost function decreased three orders of magnitude after 60 iterations. However, the value of the cost function associated with the fine-mesh resolution was reduced by only two or three orders of magnitude, while the norm of the gradient of the cost function decreased by only two orders of magnitude for an identical number of iterations.

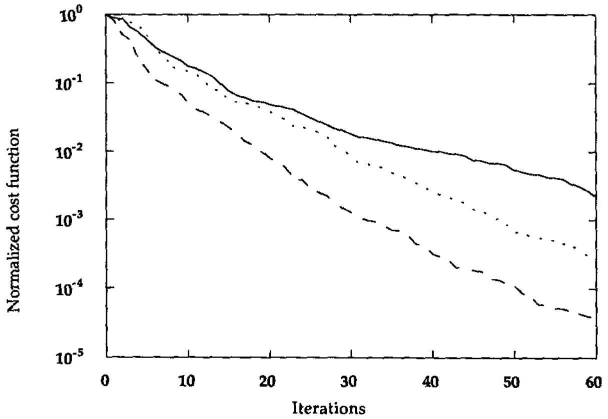

FIG. 8. The variation of the value of the normalized cost functions on three different horizontal mesh resolutions: that is, the fine-mesh 1 (20 × 21) (solid), the coarse-mesh 1 (15 × 12) (dotted), and the coarse-mesh 2 (8 × 9) (dashed) with the number of iterations.

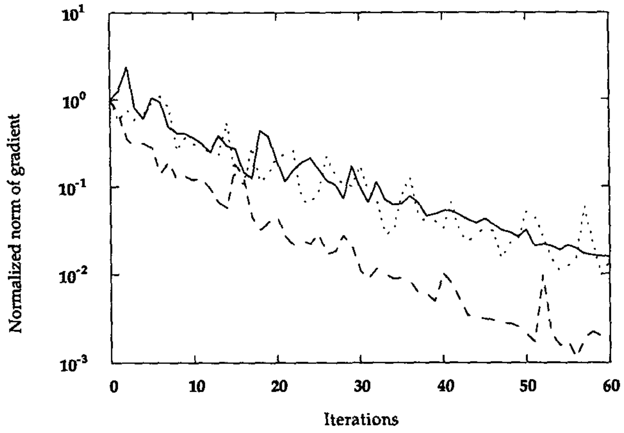

FIG. 9. The variation of the value of the normalized gradient of the cost functions on three different horizontal mesh resolutions: that is, the fine-mesh 1 (20 × 21) (solid), the coarse-mesh 1 (15 × 12) (dotted), and the coarse-mesh 2 (8 × 9) (dashed) with the number of iterations.

This can be explained by noting that the results from the fine mesh contain more small-scale features than the corresponding ones from the coarse mesh, and the dimension of the control variable also impacts upon the convergence rate so that the retrieval with the fine-mesh model data becomes more difficult. This results in a slowdown in the convergence rate of the minimization procedure since the presence of small-scale features results in an increase in the condition number of the Hessian of the cost functional of the fine-mesh resolution model due to the introduction of small eigenvalues in the spectrum of the Hessian (see Axellson and Barker 1984).

### b. *Fitting model results with various horizontal mesh resolutions*

One of the applications of the optimal control method for distributed parameters is to minimize the difference between two different states, so as to obtain

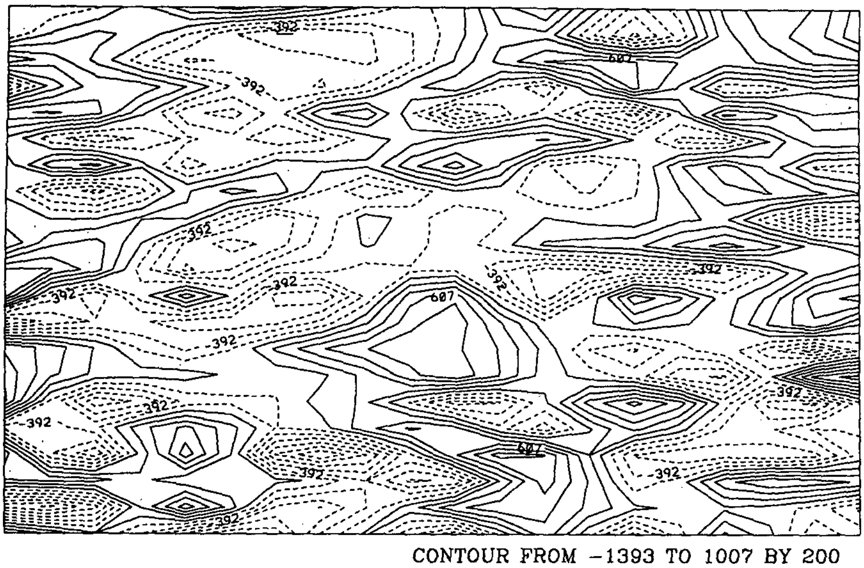

FIG. 10. The difference between perturbed initial geopotential height field of the coarse-mesh model and initial geopotential height field of the fine-mesh model.

the best fit between these states. Here we carry out some experiments to study transfer of information among different horizontal mesh resolutions using the same finite-element model.

To perform our experiment, we define a cost function that differs from the one given by (3.1); that is, the cost function is now defined as

$$J[\mathbf{X}_{\rm coa}(t_0)] = \frac{1}{2}\sum_{r=0}^{R}\langle \mathbf{W}(t_r)[\mathbf{H}\mathbf{X}_{\rm coa}(t_r) - \mathbf{X}_{\rm fin}(t_r)], [\mathbf{H}\mathbf{X}_{\rm coa}(t_r) - \mathbf{X}_{\rm fin}(t_r)]\rangle, \qquad (5.3)$$

where $\mathbf{X}(t_r)$ is an $m$-dimensional vector of the model variables at time $t_r$ defined on the coarse-mesh space $\Re_m$; $\mathbf{X}_{\rm fin}$ is an $n$-dimensional vector ($n \geqslant m$) defined on the fine-mesh space $\Re_n$; and $\mathbf{H}$ is a projection operator (interpolation or extrapolation operator) from $\Re_m$ to $\Re_n$, which is the cubic-spline interpolation in following experiments; $R$ is also the number of time levels where observations are available in the assimilation interval; $\langle\, ,\, \rangle$ denotes the inner product of two vectors in a given norm (usually the Euclidean norm); and $\mathbf{W}(t_r)$ is an $N \times N$ diagonal weighting matrix. The proof of the uniqueness of the solution of this optimal control problem is equivalent to that obtained by Zou et al. (1992, 1993).

The gradient of the cost function with respect to the initial conditions is obtained in the same way as that in section 4, but the forcing term added to the adjoint variable during the integration of the adjoint model is replaced by the following term:

$$\mathbf{H}^{\rm T}\mathbf{W}(\mathbf{H}\mathbf{X}_{\rm coa} - \mathbf{X}_{\rm fin}), \qquad (5.4)$$

where $\mathbf{H}^{\rm T}$ is the adjoint version of the projection operator $\mathbf{H}$. Here, we may view data originating from the fine-mesh resolution model $\mathbf{X}_{\rm fin}$ as observations and $\mathbf{X}_{\rm coa}$ as the model solution.

In all the following experiments, we use the L-BFGS (Liu and Nocedal 1989) unconstrained large-scale minimization algorithm to minimize the cost function, and a mesh with a resolution of 20 × 21 grid points is used as a fine-mesh model. The time step for the fine-mesh model is 600 s, and the initial conditions for the fine-mesh model are given by (4.1) and (4.2).

In the first experiments a 10 × 21 coarse-mesh resolution model was used for $\mathbf{X}_{\rm coa}$. The time step for the coarse-mesh model is also 600 s, and the initial conditions on the coarse-mesh model were perturbed by random perturbations of (4.1) and (4.2), obtained by using a standard library randomizer RANF on the CRAY-YMP machine for the solution. Figure 10 shows the difference between the perturbed initial height field of the coarse-mesh model and initial height field of the fine-mesh model. The difference formula is as follows:

$$\delta\boldsymbol{\phi} = \mathbf{H}\boldsymbol{\phi}_{\rm coa} - \boldsymbol{\phi}_{\rm fin}. \qquad (5.5)$$

The length of the assimilation window is 3 h and the forcing terms are inserted on the right-hand side of the adjoint model at every time step during the adjoint integration in this experiment. Figure 11 shows the variation of the normalized cost functions and norm of the gradient of the cost function with the number of iterations. The cost function and norm of its gradient display a decrease with the number of iterations. The difference between the VDA retrieved initial geopotential height field of the coarse-mesh model and the initial geopotential height field of the fine-mesh model is displayed

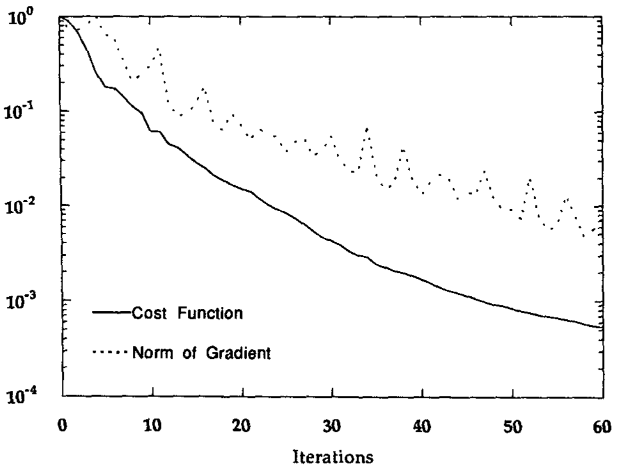

FIG. 11. The variation of the normalized cost functions (solid) and norm of its gradient (dotted) of the cost function with the number of iterations.

in Fig. 12. It represents a sizable decrease in the difference between the corresponding geopotential fields prior to VDA as portrayed in Fig. 10. This result illustrates also the fact that the VDA solution of the model using the coarse-mesh horizontal resolution is closer to the true solution.

We also carried out an experiment aimed at minimizing cost function (3.1), where the mesh resolutions of both $\mathbf{X}_{\mathrm{coa}}$ and $\mathbf{X}_{\mathrm{fin}}$ are 10 × 21. Figure 13 shows the variation of the normalized cost function and its gradient with the number of iterations. We find out that the convergence rate of minimizing the cost function given by (5.3) is slower than that required for minimizing (3.1). The difference in the convergence rates can be attributed to the condition number of the Hessian matrix that depends on the cost function given by the projection operator.

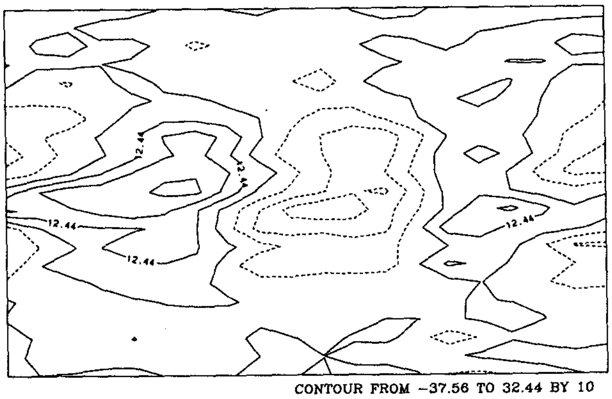

FIG. 12. The difference between retrieval initial geopotential height field of the coarse-mesh model and initial geopotential height field of the fine-mesh model.

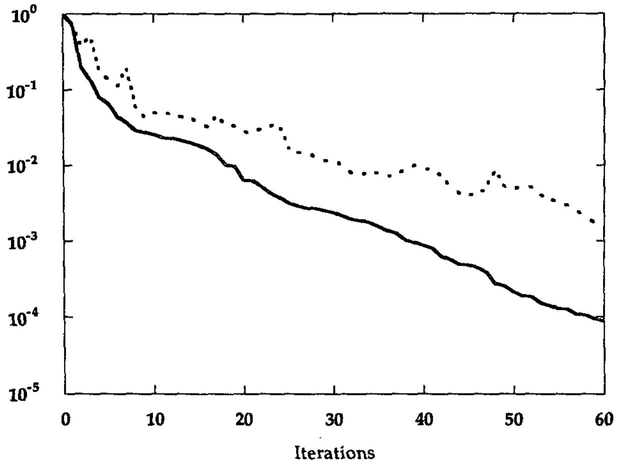

FIG. 13. Variation of the value of the normalized cost function $J/J_0$ (solid) and the norm of gradient $\|\nabla J_k\|/\|\nabla J_0\|$ (dotted) with the number of iterations.

A minimization was carried out in which a flat field served as the initial conditions of the coarse-mesh resolution model. The observations are still the same as above. A satisfactory retrieval was obtained. This experiment indicates that the retrieval difference is independent of the initial errors.

Then, we carried out a 12-h forecast of both the coarse-mesh model and the fine-mesh model (Fig. 14). The integrations started (a) from initial conditions given by (4.1) and (4.2) on the fine-mesh resolution, (b) from random perturbations of (4.1) and (4.2) on the coarse-mesh resolution, and (c) from variationally assimilated data where the cost function measured the lack of fit between the coarse- and the fine-mesh resolutions model on the coarse-mesh resolution. We ob-

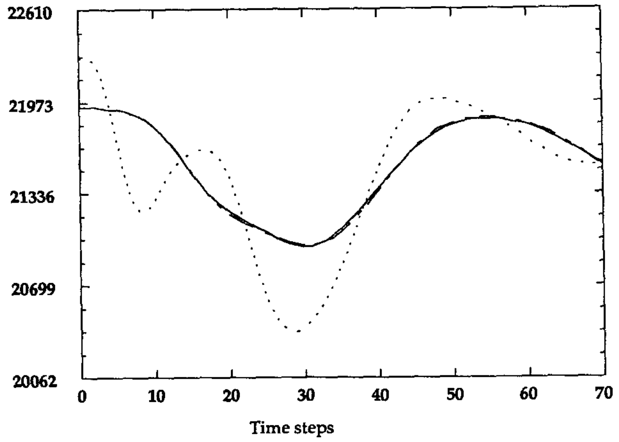

FIG. 14. A 12-h forecast of the geopotential height fields at a grid point from the initial field of the fine-mesh model (solid), the perturbed initial field of the coarse-mesh model (dotted), and the retrieval initial field of the coarse-mesh model (dash–dot).

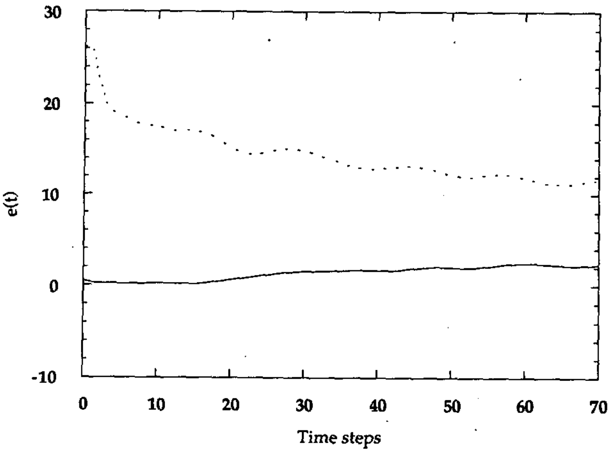

FIG. 15. Temporal variation of rmse of the geopotential fields, over a 12-h period, between the variationally assimilated integration of the coarse-mesh (10 × 21) model and integration of the fine-mesh model (solid), between the guess integration of the coarse-mesh model and the integration of the fine-mesh model (dotted).

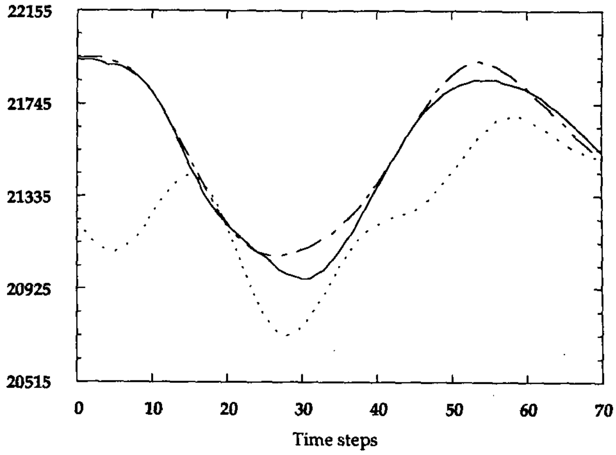

FIG. 16. A 12-h forecast of the geopotential height fields at a grid point from the initial field of the fine-mesh model (solid), the perturbed initial field of the coarse-mesh (10 × 11) model (dotted), and the variationally assimilated initial field of the coarse-mesh model (dash–dot).

serve from Fig. 14 that the forecast result integrated from the assimilated initial field of the coarse-mesh model is almost identical with the forecast result integrated from the initial field of the fine-mesh model.

To better assess the temporal variation of the height fields, we display the temporal variation of the root-mean-square error (rmse) in Fig. 15. The rmse is defined as follows:

$$e(t) = \left\{\frac{1}{N}\sum_{i=0}^{N}[\mathbf{H}\phi_{\text{coa}}(i,t) - \phi_{\text{fin}}(i;t)]^2\right\}^{1/2}, \quad (5.6)$$

where $i$ denotes the number of horizontal grid points, $N$ is the total number of grid points, and $t$ represents time. We see from Fig. 15 that the temporal variation of the rmse between the variationally assimilated result of the coarse-mesh model and the result of the fine-mesh model is much smaller than the temporal variation of the rmse between the guess result of the coarse-mesh model and the result of the fine-mesh model. These results indicate a successful VDA fitting of model data originating from the coarse- and fine-horizontal mesh resolutions after minimization of the cost function given by Eq. (5.3).

Next, another model mesh resolution (10 × 11) was used as the coarse-mesh model to carry out the VDA experiment. The initial conditions used are the same as in the first experiment and the time step for the coarse-mesh model integration is 600 s. A smaller time step is used here in order to avoid interpolations in time for the fine-mesh grid. Figure 16 shows a 12-h evolution forecast of geopotential height fields at an arbitrarily chosen fine-mesh grid point corresponding to the three initial conditions (a), (b), and (c) used in the first experiment. We observe from Fig. 16 that the result of the forecast integrated from the assimilated initial field of the coarse-mesh model is much closer to the result of the forecast integrated from the initial field of the fine-mesh model, while some difference between the two forecasts is still noticed. The temporal variation of the rmse is displayed in Fig. 17. The temporal variation of the rmse between the variationally assimilated integration of the coarse-mesh model and the integration of the fine-mesh model after 12 h is much smaller than the temporal variation of the rmse between the guess integration of the coarse-mesh model and the integration of the fine-mesh model but the magnitude of the rmse is slightly larger than that displayed in Fig. 15.

Finally, we choose a mesh resolution of 8 × 9 grid points as a very coarse mesh model to perform a sim-

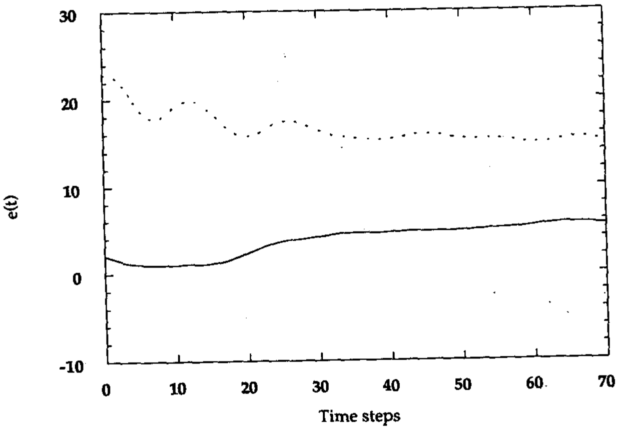

FIG. 17. Temporal variation of rmse of the geopotential fields, over a 12-h period, between the variationally assimilated integration of the coarse-mesh (10 × 11) model and the integration of the fine-mesh model (solid), between the guess integration of the coarse-mesh model and the integration of the fine-mesh model (dotted).

ilar VDA experiment. This time the minimization process failed. After 30 iterations, the minimization terminates unsuccessfully due to rounding errors prior to a prescribed convergence criterion being attained.

To better understand the underlying reasons for the failure of the minimization, we calculated the condition number of the Hessian of the cost function for the three mesh resolutions used in our VDA experiments.

Table 5 presents the maximum and minimum eigenvalues as well as the condition number of the Hessian at the initial guess of the minimization for the three mesh resolutions. Since the dimension of the control variable in variational data assimilation is very large (up to $10^6$ or more), it is computationally too expensive to calculate the eigenvalue spectrum of the full Hessian. However, it is possible to obtain an estimate of the maximum and minimum eigenvalues as well as the corresponding condition number of the Hessian by using the power method and shifted power method, which requires only the value of the Hessian multiplied by a vector (Golub and Van Loan 1989; Zou et al. 1992). We observe from Table 5 that reducing the mesh resolution (i.e., using a rather coarse mesh) results in a negative minimum eigenvalue of the Hessian in case 3, corresponding to the very coarse mesh resolution. Thus, an indefinite Hessian obtains for this case, indicating the presence of multiple minima or saddle points, that is, an ill-posed problem.

These experiments point to the fact that variational data assimilation method can be used to fit results produced with fine-mesh resolutions of the shallow-water equations model using the coarse-mesh model when the difference between the corresponding mesh resolutions is not too large and when the coarse mesh can still adequately resolve large-scale Rossby waves solution.

## 6. Summary and conclusions

In this paper, we presented a VDA scheme using a finite-element model of the shallow-water equations and its adjoint. This is to the best of our knowledge the first derivation of the adjoint of a finite-element model in meteorology. A particular hurdle in the derivation of its adjoint model was the derivation of the adjoint of the Gauss–Seidel iterative procedure that differs in a discernible way from other iterative schemes, in that some of the variables are reused while others are not, when deriving the adjoint of this iterative procedure.

The large-scale unconstrained minimization limited-memory quasi-Newton method of Liu and Nocedal (1989) was used to minimize the cost function consisting of the weighted squared difference between model solution and observations over the interval (time window) of assimilation. Using random perturbations of the Grammeltvedt (1969) initial condition fields as initial guess for the solution, we carried out VDA experiments using the adjoint model to adjust the model solution to a set of observations distributed in space and time. The minimization was able to retrieve the initial conditions with a satisfactory rate of convergence when adequate scaling was employed. However, both the convergence rate and the accuracy of the retrieval are quite different from those obtained in the case where no scaling was employed. As expected, the minimization with adequate scaling yields much better results than those obtained in the case of no scaling, converging much faster, that is, requiring a smaller number of minimization iterations.

Adequate scaling factors are a simple but crude way to do preconditioning since ultimately we need to know the Hessian matrix of the cost function to efficiently deal with small-scale features and carry out a more sophisticated preconditioning based on knowledge of the Hessian spectrum. We also illustrated in a series of numerical experiments that most of the large-scale VDA retrieval was achieved during the first 10–15 iterations of the VDA minimization process.

A number of experiments were carried out for comparison purposes, involving VDA with the same finite-element model of the shallow-water equations but with various horizontal resolutions.

First, three horizontal mesh resolutions were tested using the aforementioned random perturbations of the initial conditions as the initial guess, and all the VDA experiments were able to retrieve the initial conditions when proper scaling was employed, but the convergence rate and accuracy of the retrieval for the three cases of horizontal resolution turned out to be quite different. The minimization of the cost func-

TABLE 5. Values of the maximum and minimum eigenvalues ($\lambda_{\max}$, $\lambda_{\min}$), and the condition number (Cond num) of the Hessian of the cost function at the initial guess point, when the fine-mesh resolution is 20 × 21 and the coarse-mesh resolutions are 20 × 21 (case 1), 10 × 21 (case 2), and 8 × 9 (case 3), respectively.

| Cases | $\lambda_{\max}$ | $\lambda_{\min}$ | Cond num | Convergence criteria |
|---|---|---|---|---|
| Case 1 | $0.2444 \times 10^{-5}$ | $0.3516 \times 10^{-8}$ | 695.1 | |
| Case 2 | $0.2056 \times 10^{-5}$ | $0.1723 \times 10^{-8}$ | 1193.3 | $\lvert \lambda^{(k)} - \lambda^{(k-1)} \rvert < 10^{-9}$ |
| Case 3 | $0.2718 \times 10^{-5}$ | $-0.1936 \times 10^{-7}$ | −140.4 | ($k$ the iteration number) |

tion and the norm of its gradient for the coarse-mesh resolution model exhibited a faster convergence rate than the one corresponding to the fine-mesh model. This confirms the obvious expectation that the convergence rate of the minimization process is related to the horizontal resolution; that is, the higher the resolution, the slower the convergence rate is, due mainly to the presence of smaller scales to be balanced. It was also found that even though the variable horizontal mesh resolution is changed in conjunction with the same model it is still necessary to employ variable scaling factors corresponding to the various horizontal resolutions.

We then used a VDA method for fitting different model solutions corresponding to different mesh resolutions in order to investigate the transfer of information among different scales of motion. We found that results from coarse-mesh model integration can recover most of the information of the fine-mesh model in the framework of VDA when results of the fine-mesh model serve as observations and the L-BFGS large-scale unconstrained minimization was used for minimizing the respective cost functional measuring the lack of fit between model results originating from high- and low-resolution meshes, respectively. The smaller the difference between the mesh resolutions is, the better the VDA fitting adjustment results are for model results obtained with various horizontal mesh resolutions.

The results obtained in this research indicate that using a finite-element model to perform VDA using the initial conditions as control variables was successful. This first application in meteorology of a finite-element adjoint model development for VDA is the forerunner of a continuing effort aimed at developing 4D VDA for finite-element numerical weather prediction models with real data.

*Acknowledgments.* Support for this research was provided by SCRI at The Florida State University, which is partially funded by the Department of Energy through Contract DE-FC0583ER250000. Partial support through Grant AFOSR 4962092J0498 for the first author is also acknowledged. The authors would like to thank two anonymous reviewers for their contribution and beneficial suggestions, which lead to an improvement in the presentation of this paper.

## APPENDIX A

## Description of the Assembled Matrices Used in Section 2b

A convenient procedure for evaluating integrals for each triangle can be analytically used by the following formula for area integrals (see Silvester 1970):

$$\iint_{\omega_{\mathrm{el}}} L_1^a L_2^b L_3^c dxdy = \frac{a!b!c!}{(a+b+c+2)}, \qquad \text{(A.1)}$$

where $a$, $b$, and $c$ are integers, $L_i$ $(i = 1, 2, 3)$ are the basis functions for the triangular linear element of the natural coordinate variables:

$$L_i = \frac{1}{2\omega_{\mathrm{el}}}(a_i x + b_i y + c_i) \quad i = 1, 2, 3, \qquad \text{(A.2)}$$

where $\omega_{\mathrm{el}}$ stands for the area of the triangle and

$$\begin{aligned} a_i &= x_j - x_k, \\ b_i &= y_k - y_j, \\ c_i &= x_j y_k - x_k y_j, \end{aligned} \qquad \text{(A.3)}$$

and $i$, $j$, and $k$ are cyclically permuted $(i, j, k = 1, 2, 3)$.

The derivatives of the shape functions $L_i$ are

$$\frac{\partial L_i}{\partial x} = \frac{a_i}{2\omega_{\mathrm{el}}}, \quad \frac{\partial L_i}{\partial y} = \frac{b_i}{2\omega_{\mathrm{el}}}, \quad i = 1, 2, 3. \qquad \text{(A.4)}$$

Using these formulas we can derive four different types of element matrices (3 × 3) that are required for the assembly in the global matrices. They are

$$\mathbf{S}_1 = \iint_{\omega_{\mathrm{el}}} V_i V_j d\omega = \frac{\omega_{\mathrm{el}}}{12}\begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}, \qquad \text{(A.5)}$$

where $\omega_{\mathrm{el}}$ is the area of the triangular element—this type of element matrix is related to Eq. (2.16)—and

$$\begin{aligned} \mathbf{S}_2 &= \iint_{\omega_{\mathrm{el}}} V_k \frac{\partial V_i}{\partial x} d\omega = \iint_{\omega_{\mathrm{el}}} V_k \frac{a_i}{2\omega_{\mathrm{el}}} d\omega \\ &= \frac{1}{6}\begin{pmatrix} a_1 & a_2 & a_3 \\ a_1 & a_2 & a_3 \\ a_1 & a_2 & a_3 \end{pmatrix} \\ & \qquad a_i = x_k - x_j, \end{aligned} \qquad \text{(A.6)}$$

where $x_k$ and $x_j$ are the local Cartesian coordinates for a given triangle. This type of element matrix is identical with Eqs. (2.24) and (2.29):

$$\begin{aligned} \mathbf{S}_3 &= \iint_{\omega_{\mathrm{el}}} V_k \xi_j V_i \frac{\partial V_j}{\partial x} d\omega = \iint_{\omega_{\mathrm{el}}} V_k \xi_j V_i \frac{a_j}{2\omega_{\mathrm{el}}} d\omega \\ &= \frac{1}{2\omega_{\mathrm{el}}} \xi_j a_j \iint_{\omega_{\mathrm{el}}} V_i V_k d\omega \\ &= \frac{1}{24}\sum \xi_j a_j \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}, \end{aligned} \qquad \text{(A.7)}$$

where $\xi$ represent either $u_j$, $v_j$, or $\phi_j$—this type of element matrix is related to $\mathbf{K}_2$, $\mathbf{K}_3$ in section 2b:

$$\mathbf{S}_4 = \iint_{\omega_{\rm el}} \xi_j V_j V_k \frac{\partial V_i}{\partial x} d\omega = \iint_{\omega_{\rm el}} \xi_j V_j V_k \frac{a_i}{2\omega_{\rm el}} d\omega = \frac{a_i}{2\omega_{\rm el}} \sum_{\substack{j \neq k \\ j=1}} \left( \frac{\omega_{\rm el}}{12} \xi_j + \frac{\omega_{\rm el}}{6} \xi_k \right)$$

$$= \frac{1}{24} \begin{bmatrix} (2\xi_1 + \xi_2 + \xi_3)a_1 & (2\xi_1 + \xi_2 + \xi_3)a_2 & (2\xi_1 + \xi_2 + \xi_3)a_3 \\ (\xi_1 + 2\xi_2 + \xi_3)a_1 & (\xi_1 + 2\xi_2 + \xi_3)a_2 & (\xi_1 + 2\xi_2 + \xi_3)a_3 \\ (\xi_1 + \xi_2 + 2\xi_3)a_1 & (\xi_1 + \xi_2 + 2\xi_3)a_2 & (\xi_1 + \xi_2 + 2\xi_3)a_3 \end{bmatrix}$$

$$= \frac{1}{24} \begin{pmatrix} a_1\eta_a & a_2\eta_a & a_3\eta_a \\ a_1\eta_b & a_2\eta_b & a_3\eta_b \\ a_1\eta_c & a_2\eta_c & a_3\eta_c \end{pmatrix}; \qquad \text{(A.8)}$$

the subscripts 1, 2, and 3 standing for three points in the triangle, respectively, and $\eta_a$, $\eta_b$, and $\eta_c$ are

$$\eta_a = 2\xi_1 + \xi_2 + \xi_3$$
$$\eta_b = \xi_1 + 2\xi_2 + \xi_3$$
$$\eta_c = \xi_1 + \xi_2 + 2\xi_3. \qquad \text{(A.9)}$$

This type of element matrix is related to $\mathbf{K}_1$ in section 2b.

APPENDIX B

## The Adjoint of the Gauss–Seidel (G–S) Iterative Procedure

The adjoint of the G–S iterative procedure on some inner nodes, boundary nodes at $k$th, and the adjoint of the last iteration are as follows.

1) For the G–S $k$th iteration not including the last $K$th iteration we have the following.

(a) For the nodes [$l = 2 + (i - 1)J, i = 1, 2, \cdots, I$] the iterative formulas are given by

$$\xi_l^{(k)} = c_{l,1}\xi_{l-J}^{(k)} + c_{l,2}\xi_{l-1}^{(k)} + c_{l,3}\xi_{l-(J-1)}^{(k)} + c_{l,4}\xi_{l+1}^{(k-1)} + c_{l,5}\xi_{l+(J-1)}^{(k-1)} + c_{l,6}\xi_{l+J}^{(k-1)} + d_l$$
$$l = 2 + (i - 1) \times J, \quad i = 2, 3, \cdots, I - 1 \qquad \text{(B.1a)}$$

$$\xi_l^{(k)} = c_{l,1}\xi_{l-1}^{(k)} + c_{l,2}\xi_{l+(J-1)}^{(k-1)} + c_{l,3}\xi_{l+J}^{(k-1)} + c_{l,4}\xi_{l+1}^{(k-1)} + c_{l,5}\xi_{l+J(I-1)}^{(k-1)} + c_{l,6}\xi_{l+J(I-1)+1}^{(k-1)} + d_l$$
$$l = 2 + (i - 1)J, \quad i = 1 \qquad \text{(B.1b)}$$

$$\xi_l^{(k)} = c_{l,1}\xi_{l-J}^{(k)} + c_{l,2}\xi_{l-1}^{(k)} + c_{l,3}\xi_{l-(J-1)}^{(k)} + c_{l,4}\xi_{l+1}^{(k-1)} + c_{l,5}\xi_{l-J(I-1)-1}^{(k)} + c_{l,6}\xi_{l-J(I-1)}^{(k)} + d_l$$
$$l = 2 + (i - 1)J, \quad i = I. \qquad \text{(B.1c)}$$

One observes by using (B.1a) that the variable $\xi_{l-J}^{(k)}$ will be reused at the next iteration when $\xi_{l-(2J)}^{(k+1)}$ is iteratively solved; $\xi_{l-1}^{(k)}$ will be reused when $\xi_{l+(J-2)}^{(k)}$ is iteratively solved; $\xi_{l-(J-1)}^{(k)}$ will be reused at the following step, on node $l' = l + 1$—that is, when $\xi_{l+1}^{(k)}$ is iteratively solved; $\xi_{l+1}^{(k-1)}$ is not reused; $\xi_{l+(J-1)}^{(k-1)}$ is not reused; $\xi_{l+J}^{(k-1)}$ will be reused when $\xi_{l+(J-1)}^{(k)}$ is solved; $d_l$ is reused at the next iteration on node $l$. The corresponding adjoint formulas of (B.1a) are

$$\hat{\xi}_{l-J}^{(k)} = c_{l,1}\hat{\xi}_l^{(k)} + \hat{\xi}_{l-J}^{(k)}$$
$$\hat{\xi}_{l-1}^{(k)} = c_{l,2}\hat{\xi}_l^{(k)} + \hat{\xi}_{l-1}^{(k)}$$
$$\hat{\xi}_{l-(J-1)}^{(k)} = c_{l,3}\hat{\xi}_l^{(k)} + \hat{\xi}_{l-(J-1)}^{(k)}$$
$$\hat{\xi}_{l+1}^{(k-1)} = c_{l,4}\hat{\xi}_l^{(k)}$$
$$\hat{\xi}_{l+(J-1)}^{(k-1)} = c_{l,5}\hat{\xi}_l^{(k)}$$
$$\hat{\xi}_{l+J}^{(k-1)} = c_{l,6}\hat{\xi}_l^{(k)} + \hat{\xi}_{l+J}^{(k-1)}$$
$$\hat{d}_l = \hat{\xi}_l^{(k)} + \hat{d}_l, \qquad \text{(B.1a)}'$$

where in the following formulas the caret represents the adjoint variable.

A similar analysis as the above using the fact that the variable values in (B.1b), $\xi_{l+1}^{(k-1)}$, and $\xi_{l+(J-1)}^{(k-1)}$ are not reused. [Note: here the position of $\xi_{l+(J-1)}^{(k-1)}$ is different from (B.1a).]

In (B.1c) the variable values $\xi_{l+1}^{(k-1)}$ and $\xi_{l-J(I-1)-1}^{(k)}$ are not reused.

(b) For all the north-boundary nodes [$l = J + (i - 1)J, i = 1, 2, 3, \cdots, I$] the iterative formulas are

$$\xi_l^{(k)} = c_{l,1}\xi_{l-J}^{(k)} + c_{l,2}\xi_{l-1}^{(k)} + c_{l,3}\xi_{l+(J-1)}^{(k-1)} + c_{l,4}\xi_{l+J}^{(k-1)} + d \quad l = J + (i - 1)J, \quad i = 2, 3, \cdots, I - 1 \qquad \text{(B.2a)}$$

$$\xi_l^{(k)} = c_{l,1}\xi_{l-1}^{(k)} + c_{l,2}\xi_{l+(J-1)}^{(k-1)} + c_{l,3}\xi_{l+J}^{(k-1)} + c_{l,4}\xi_{l+J(I-1)}^{(k-1)} + d_l \quad l = J + (i - 1)J, \quad i = 1 \qquad \text{(B.2b)}$$

$$\xi_l^{(k)} = c_{l,1}\xi_{l-J}^{(k)} + c_{l,2}\xi_{l-1}^{(k)} + c_{l,3}\xi_{l-J(I-1)-1}^{(k)} + c_{l,4}\xi_{l-J(I-1)}^{(k)} + d_l \quad l = J + J(i - 1), \quad i = I. \qquad \text{(B.2c)}$$

We see from (B.2a) that the variable $\xi_{l-J}^{(k)}$ will be reused at the next iteration when $\xi_{l-(2J)}^{(k+1)}$ is iteratively solved; $\xi_{l-1}^{(k)}$ and $\xi_{l+(J-1)}^{(k-1)}$ will be reused when $\xi_{l+(J-2)}^{(k)}$ is iteratively solved; $\xi_{l-(J-1)}^{(k)}$ will be reused at the following step, that is, when $\xi_{l+1}^{(k)}$ is iteratively solved; $\xi_{l+1}^{(k-1)}$ is not reused; $\xi_{l+J}^{(k-1)}$ will be reused when $\xi_{l+(J-1)}^{(k)}$ is iteratively solved; $\xi_{l+J}^{(k-1)}$ will be reused when $\xi_{l+(J-1)}^{(k)}$ is solved; $\xi_{l+J(I-1)}^{(k-1)}$, $\xi_{l-J(I-1)-1}^{(k)}$ and $\xi_{l-J(I-1)}^{(k)}$ in (B.2b) and (B.2c) are also reused.

The corresponding adjoint model formulas for (B.2a) are

$$\hat{\xi}_{l-J}^{(k)} = c_{l,1}\hat{\xi}_{l}^{(k)} + \hat{\xi}_{l-J}^{(k)}$$

$$\hat{\xi}_{l-1}^{(k)} = c_{l,2}\hat{\xi}_{l}^{(k)} + \hat{\xi}_{l-1}^{(k)}$$

$$\hat{\xi}_{l+(J-1)}^{(k-1)} = c_{l,3}\hat{\xi}_{l}^{(k)} + \hat{\xi}_{l+(J-1)}^{(k-1)}$$

$$\hat{\xi}_{l+J}^{(k-1)} = c_{l,4}\hat{\xi}_{l}^{(k)} + \hat{\xi}_{l+J}^{(k-1)}$$

$$\hat{d}_l = \hat{\xi}_{l}^{(k)} + \hat{d}_l. \qquad \text{(B.2a)}'$$

The corresponding adjoint model formula for (B.2b) and (B.2c) are derived in a similar manner as for (B.2a)′.

(c) For the south-boundary nodes $[l = 1 + (i - 1)J, i = 1, 2, \cdots, I]$ the iterative formulas are

$$\xi_{l}^{(k)} = c_{l,1}\xi_{l-J}^{(k)} + c_{l,2}\xi_{l-(J-1)}^{(k)} + c_{l,3}\, I\, \xi_{l+1}^{(k-1)} + c_{l,4}\xi_{l+J}^{(k-1)} + d_l \quad l = 1 + (i-1)J, \quad i = 2, 3, \cdots, I-1 \qquad \text{(B.3a)}$$

$$\xi_{l}^{(k)} = c_{l,1}\xi_{l+J}^{(k-1)} + c_{l,2}\xi_{l+1}^{(k-1)} + c_{l,3}\xi_{l+J(I-1)}^{(k-1)} + c_{l,4}\xi_{l+J(I-1)+1}^{(k-1)} + d_l \quad l = 1 + (i-1)J, \quad i = 1 \qquad \text{(B.3b)}$$

$$\xi_{l}^{(k)} = c_{l,1}\xi_{l-J}^{(k)} + c_{l,2}\xi_{l-(J-1)}^{(k)} + c_{l,3}\xi_{l+1}^{(k-1)} + c_{l,4}\xi_{l-J(I-1)}^{(k)} + d_l \quad l = 1 + (i-1)J, \quad i = I; \qquad \text{(B.3c)}$$

only $\xi_{l+1}^{(k-1)}$ is not reused while all the other values with a subscript not equal to $l + 1$ are reused.

(d) For west-boundary nodes (cyclic boundary condition on west–east direction) the iterative formulas are given by

$$\xi_{l}^{(k)} = c_{l,1}\xi_{l-1}^{(k)} + c_{l,2}\xi_{l+(J-1)}^{(k-1)} + c_{l,3}\xi_{l+J}^{(k-1)} + c_{l,4}\xi_{l+1}^{(k-1)} + c_{l,5}\xi_{l+J(I-1)}^{(k-1)} + c_{l,6}\xi_{l+J(I-1)+1}^{(k-1)} + d_l$$

$$l = j, \quad j = 3, 4, \cdots, J-1; \qquad \text{(B.4)}$$

where $\xi_{l}^{k}$ is solved, only the variable $\xi_{l+1}^{(k-1)}$ is not reused while all the other variable values with subscripts different from $l + 1$ are reused. The corresponding adjoint model formulas are derived in a similar manner to that of Eq. (3.10).

(e) For east-boundary nodes

$$\xi_{l}^{(k)} = c_{l,1}\xi_{l-J}^{(k)} + c_{l,2}\xi_{l-1}^{(k)} + c_{l,3}\xi_{l-(J-1)}^{(k)} + c_{l,4}\xi_{l+1}^{(k-1)} + c_{l,5}\xi_{l-J(I-1)-1}^{(k)} + c_{l,6}\xi_{l-J(I-1)}^{(k)} + d_l$$

$$l = j + (I-1)J, \quad j = 3, 4, \cdots, J-1; \qquad \text{(B.5)}$$

when the variable $\xi_{l}^{(k)}$ is solved only $\xi_{l+1}^{(k-1)}$ is not reused while all the other variables with subscripts different from $l + 1$ are reused. The adjoint formulas are similar to that of Eq. (3.10).

2) *K*th iteration:

(a) For nodes $l = j$ and $l = j + J, j = 1, 2, \cdots, J - 1$ we obtain identical relations to those of the *k*th iteration.

(b) For nodes $l = j + J(I - 1), j = 2, 3, \cdots, J - 1$, in addition to the $\xi_{l+1}^{(K-1)}$, three other inputs $\xi_{l-J}^{(K)}$, $\xi_{l-1}^{(K)}$ and $\xi_{l-J(I-1)-1}^{(K)}$ are not reused when we solve for the variable $\xi_{l}^{(K)}$.

The corresponding adjoint model formulas are given by

$$\hat{\xi}_{l-J}^{(K)} = c_{l,1}\hat{\xi}_{l}^{(K)}$$

$$\hat{\xi}_{l-1}^{(K)} = c_{l,2}\hat{\xi}_{l}^{(K)}$$

$$\hat{\xi}_{l-(J-1)}^{(K)} = c_{l,3}\hat{\xi}_{l}^{(K)} + \hat{\xi}_{l-(J-1)}^{(K)}$$

$$\hat{\xi}_{l+1}^{(K-1)} = c_{l,4}\hat{\xi}_{l}^{(K)}$$

$$\hat{\xi}_{l-J(I-1)-1}^{(K)} = c_{l,5}\hat{\xi}_{l}^{(K)}$$

$$\hat{\xi}_{l-J(I-1)}^{(K)} = c_{l,6}\hat{\xi}_{l}^{(K)} + \hat{\xi}_{l-J(I-1)}^{(K)}$$

$$\hat{d}_l = \hat{\xi}_{l}^{(K)} + \hat{d}_l. \qquad \text{(B.6)}$$

(c) For the rest of the nodes, another variable, $\xi_{l-J}^{(K-1)}$, is not reused besides the variables discussed above for the *K*th iteration.

## REFERENCES

Axellson, O., and V. A. Barker, 1984: Finite element solution of boundary value problems: Theory and computation. *Computer Science and Applied Mathematics Series,* Academic Press, 437 pp.

Beland, M., J. Côté, and A. N. Staniforth, 1983: The accuracy of a finite element vertical discretization scheme for primitive equations models: Comparison with a finite difference scheme. *Mon. Wea. Rev.,* **111,** 1189–1207.

Côté, J., and A. Staniforth, 1990b: An accurate and efficient finite-element global model of the shallow-water equations. *Mon. Wea. Rev.,* **118,** 2707–2717.

——, S. Gravel, and A. Staniforth, 1990a: Improving variable resolution finite-element semi-Lagrangian integration schemes by pseudostaggering. *Mon. Wea. Rev.,* **118,** 2718–2731.

Courtier, P., 1987: Application du contrôle optimal á la prévision numerique en météorologie. These de doctorat de l'universite Paris 6, 252 pp.

——, and O. Talagrand, 1987: Variational assimilation of meteorological observations with the adjoint vorticity equation. Part II: Numerical results. *Quart. J. Roy. Meteor. Soc.,* **113,** 1329–1347.

——, and ——, 1990: Variational assimilation of meteorological observations with direct and adjoint shallow-water equations. *Tellus,* **42A,** 531–549.

——, J.-N. Thépaut, and A. Hollingsworth, 1993: A strategy for operational implementation of 4D-VAR. *Workshop Proc. of Variational Assimilation, with Special Emphasis on Three-Dimensional Aspects,* European Centre for Medium-Range Weather Forecasts, Shinfield Park, Reading, Berkshire, UK, 468 pp.

Cullen, M. J. P., and C. D. Hall, 1979: Forecasting and general circulation results from finite element models. *Quart. J. Roy. Meteor. Soc.,* **105,** 571–592.

Derber, C. J., 1987: Variational four-dimensional analysis using quasi-geostrophic constraints. *Mon. Wea. Rev.,* **115,** 998–1008.

——, 1989: A variational continuous assimilation technique. *Mon. Wea. Rev.,* **117,** 2437–2446.

Douglas, J., and T. Dupont, 1970: Galerkin method for parabolic problems. *S.I.A.M., J. Numer. Anal.,* **7,** 575–626.

Ghil, M., 1989: Meteorological data assimilation for oceanographers. Part I: Description and theoretical framework. *Dyn. Atmos. Oceans,* **13,** 171–218.

Gilbert, J. C., and C. Lemarechal, 1989: Some numerical experiments with variable storage quasi-Newton algorithms. *Math. Programm.,* **45,** 407–436.

Golub, H. G., and C. Van Loan, 1989: *Matrix Computation.* 2d Ed., The Johns Hopkins University Press, 351 pp.

Grammeltvedt, A., 1969: A survey of finite-difference schemes for the primitive equations for a barotropic fluid. *Mon. Wea. Rev.,* **97,** 384–404.

Hinsman, D. E., 1975: Application of a finite-element to the barotropic primitive equations. M.Sc. thesis, Department of Meteorology, Naval Postgraduate School, 107 pp.

——, and D. Archer, 1976: A finite-element application to a barotropic primitive equation model in spherical coordinates. *Proc. Sixth Conf. Weather Forecasting and Analysis,* Albany, NY, 121–124.

Huebner, K. H., and E. A. Thornton, 1982: *The Finite-Element Method for Engineers.* 2d ed. John Wiley and Sons, 623 pp.

Le-Dimet, F. X., and O. Talagrand, 1986: Variational algorithms for analysis and assimilation of meteorological observations: Theoretical aspects. *Tellus,* **38A,** 97–110.

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

Navon, I. M., 1977: A survey of finite-element methods in quasilinear flow problems. Report No. 240, National Research Institute for Mathematical Sciences, CSIR, Pretoria, South Africa, 44 pp.

——, 1979a: Finite-element simulation of the shallow-water equations model on a limited area domain. *Appl. Math. Model,* **3,** 337–348.

——, 1983: A Numerov–Galerkin technique applied to a finite-element shallow-water equations model with enforced conservation of integral invariants and selective lumping. *J. Comput. Phys.,* **52,** 313–339.

——, 1987: FEUDX: A two-stage, high-accuracy, finite-element FORTRAN program for solving shallow-water equations. *Comput. Geosci.,* **13,** 255–285.

——, 1988: A review of finite-element methods for solving the shallow-water equations. *Computer Modelling in Ocean Engineering,* B. D. Shrefler and O. C. Zienkiewicz, Eds., A. A. Balkema Publishers, 273–279.

——, and U. Muller, 1979: FESW—A finite-element FORTRAN IV program for solving the shallow-water equations. *Adv. Eng. Software,* **1,** 77–86.

——, and R. de Villiers, 1983: Combined penalty-multiplier optimization methods to enforce integral invariants conservation. *Mon. Wea. Rev.,* **111,** 1228–1243.

——, X. Zou, J. Derber, and J. Sela, 1992: Variational data assimilation with an adiabatic version of the NMC spectral model. *Mon. Wea. Rev.,* **120,** 1433–1446.

Neta, B., 1992: Analysis of finite-elements and finite differences for shallow water equations: A review. *Math. Comput. Simul.,* **34,** 141–161.

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

Priestly, A., 1992: The Taylor–Galerkin method for the shallow-water equations on the sphere. *Mon. Wea. Rev.,* **120,** 3003–3005.

Rao, S. S., 1989: *The Finite Element Method in Engineering.* Pergamon Press, 643 pp.

Silvester, P., 1969: Higher order polynomial finite elements for potential problems. *Int. J. Eng. Sci.,* **7,** 849–861.

——, 1970: Symmetric quadrature formulae for simplexes. *Math. Comput.,* **24,** 95–100.

Staniforth, A. N., and H. L. Mitchell, 1977: A semi-implicit finite-element barotropic model. *Mon. Wea. Rev.,* **105,** 154–169.

——, 1984: The application of the finite element method to meteorological simulations—A review. *Int. J. Numer. Meth. Fluids,* **4,** 1–12.

——, 1987: Review: Formulating efficient finite-element codes for flow in regular domains. *Int. J. Numer. Meth. Fluids,* **7,** 1–16.

——, and H. L. Mitchell, 1978: A variable resolution finite element technique for regional forecasting with the primitive equations. *Mon. Wea. Rev.,* **106,** 439–447.

——, and R. W. Daley, 1979: A baroclinic finite element model for regional forecasting with the primitive equations. *Mon. Wea. Rev.,* **107,** 107–121.

Steppeler, J., I. M. Navon, and H.-I. Lu, 1990: Finite-element schemes for extended integrations of atmospheric models. *J. Comput. Phys.,* **89,** 95–124.

Strang, W. G., and G. J. Fix, 1973: *An Analysis of the Finite Element Method.* Prentice-Hall, 306 pp.

Talagrand, O., and P. Courtier, 1987: Variational assimilation of meteorological observations with the adjoint vorticity equation—Part I. Theory. *Quart. J. Roy. Meteor. Soc.,* **113,** 1311–1328.

Tanguay, M., A. Simard, and A. Staniforth, 1989: A three dimensional semi-Lagrangian scheme for the Canadian regional finite-element forecast model. *Mon. Wea. Rev.,* **117,** 1861–1871.

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

——, D. Vasiljevic, and P. Courtier, 1993: Variational data assimilation of conventional meteorological observations with a multilevel primitive-equation model. *Quart. J. Roy. Meteor. Soc.,* **119,** 153–186.

Wang, H. H., P. Halpern, J. Douglas, and I. Dupont, 1972: Numerical solutions of the one-dimensional primitive equations using Galerkin approximation with localized basic functions. *Mon. Wea. Rev.,* **100,** 738–746.

Williams, R. T., 1981: On the formulation of finite element prediction models. *Mon. Wea. Rev.,* **109,** 463–466.

Woodward, E. T., 1981: Development of improved finite-element formulations for shallow-water equations. M.S. thesis, Department of Meteorology, Naval Postgraduate School, 168 pp.

Zou, X., I. M. Navon, and J. Sela, 1993: Control of gravitational oscillations in variational data assimilation. *Mon. Wea. Rev.,* **121,** 272–289.

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