Use of Differentiable and Nondifferentiable Optimization Algorithms for Variational Data Assimilation with Discontinuous Cost Functions

S. ZHANG, X. ZOU, J. AHLQUIST, AND I. M. NAVON

The Florida State University, Tallahassee, Florida

J. G. SELA

National Centers for Environmental Prediction, Camp Springs, Maryland

(Manuscript received 4 January 2000, in final form 24 April 2000)

ABSTRACT

Cost functions formulated in four-dimensional variational data assimilation (4DVAR) are nonsmooth in the presence of discontinuous physical processes (i.e., the presence of ''on–off'' switches in NWP models). The adjoint model integration produces values of subgradients, instead of gradients, of these cost functions with respect to the model's control variables at discontinuous points. Minimization of these cost functions using conventional differentiable optimization algorithms may encounter difficulties. In this paper an idealized discontinuous model and an actual shallow convection parameterization are used, both including on–off switches, to illustrate the performances of differentiable and nondifferentiable optimization algorithms. It was found that (i) the differentiable optimization, such as the limited memory quasi-Newton (L-BFGS) algorithm, may still work well for minimizing a nondifferentiable cost function, especially when the changes made in the forecast model at switching points to the model state are not too large; (ii) for a differentiable optimization algorithm to find the true minimum of a nonsmooth cost function, introducing a local smoothing that removes discontinuities may lead to more problems than solutions due to the insertion of artificial stationary points; and (iii) a nondifferentiable optimization algorithm is found to be able to find the true minima in cases where the differentiable minimization failed. For the case of strong smoothing, differentiable minimization performance is much improved, as compared to the weak smoothing cases.

1. Introduction

Since optimal control theory was introduced into atmospheric data assimilation by Le Dimet and Talagrand (1986), four-dimensional variational data assimilation (4DVAR) has been used for atmospheric data assimilation, initially in a research mode, and currently moving toward an operational implementation (Thépaut and Courtier 1991; Navon et al. 1992; Zupanski and Mesinger 1995; Kuo et al. 1996; Zou 1997; Rabier et al. 1998; Zou et al. 1999, manuscript submitted to *Quart. J. Roy. Meteor. Soc.*). The 4DVAR approach adjusts the control variables of a numerical weather prediction (NWP) model to their optimal values by minimizing a specified error measurement called the cost function. Information in observations is extracted, based not only on background and observational error covariances, but also on model dynamical and physical constraints. Due to its sound mathematical basis, 4DVAR is receiving more and more attention in the applications of NWP.

The optimization procedure in 4DVAR is completed by employing a large-scale unconstrained minimization algorithm. Among several choices, the limited memory quasi-Newton algorithm (Broyden 1970; Fletcher 1970; Goldfarb 1970; Shanno 1970; Liu and Nocedal 1989, collectively referred to hereafter as L-BFGS) was found to be one of the most efficient when the dimensions of the control variable are large, that is, the dimensions of the control variables are greater than 10^5. The most attractive advantage of the L-BFGS algorithm is its modest storage requirement and good convergence rate (Zou et al. 1993). The algorithm was originally designed for minimizing differentiable cost functions. Early 4DVAR experiments used either simple models or adiabatic primitive equation models without parameterized physics. The cost functions defined by these models are differentiable since solutions of the assimilation model are differentiable and the governing equations are uniformly valid over a global domain. Therefore, the L-BFGS algorithm, which is based on the assumption that the cost function is differentiable, has performed very well (Navon et al. 1992; Zou et al. 1993).

Corresponding author address: X. Zou, Department of Meteorology, The Florida State University, 404 Love Bldg., Tallahassee, FL 32306-4520.
E-mail: zou@met.fsu.edu

Nonsmooth cost functions, that is, cost functions with discontinuities and discontinuous gradients, arise in 4DVAR when assimilation models include discontinuous physical processes. A more complex diabatic model including parameterized physics simulates the evolution of the true atmospheric state better than an adiabatic one. Various physical processes are included in the conservation equations of physical variables such as momentum, mass, and energy. The physical processes are controlled by local conditions that form a set of ''IF'' statements depending on model variables and prescribed threshold values. These IF statements are often called ''on–off'' switches since they determine whether a certain physical process should be turned on or off. The presence of on–off switches renders the diabatic model solution nondifferentiable, which in turn makes the cost function defined by a diabatic model nondifferentiable even if the cost function itself is a continuous function (usually a quadratic form) of the model solution.

Computational methods of smooth and nonsmooth optimization algorithms were developed that do not assume a specific structure of the minimized function, but require only the evaluation of the function and its gradients (or their analogs in the nondifferentiable case, which are referred to as generalized gradients or subgradients) at any given point. For a class of almost everywhere differentiable functions (such as piecewise differentiable functions), the subgradients are points of nondifferentiability and can be defined by taking limits, or any linear combinations of them with the sum of the coefficients being unity.

Attempts have been made to carry out 4DVAR using a diabatic model to assimilate rainfall observations (Zupanski and Mesenger 1995; Zou and Kuo 1996; Tsuyuki 1997). There are at least two methods that have dealt with the on–off switches in the physical parameterization schemes used in 4DVAR. One consists of introducing a transitional function to make the nonlinear model solution smooth (local smoothing), that is, carrying out a local smoothing to remove the discontinuities caused by on–off switches (Zupanski 1993; Tsuyuki 1997). The other keeps the on–off switches in the tangent linear and the adjoint models the same as in the nonlinear model (Zou et al. 1993). An integration of the adjoint model in which the on–off switches are kept the same as in the nonlinear model provides the gradient of the cost function at a differentiable point and a subgradient (the limit of gradients close to the discontinuous point) at a nondifferentiable point. The the following questions need to be answered: (i) Does local smoothing eliminate problems related to on–off switches, (ii) does it cause other problems, and (iii) what are the potential problems when a differentiable optimization algorithm is employed for minimizing a nondifferentiable cost function?

It is worth mentioning that in practice there is no sharp distinction between nonsmooth and smooth functions. From the point of view of applied mathematics and computational practice, a function with a rapidly changing gradient is similar in its properties to a nonsmooth cost function. Therefore, one may expect that a differentiable optimization algorithm may work well for minimizing a nondifferentiable cost function.

In this paper, a simple model containing a typical form of the discontinuity in an NWP model is used to illustrate the performance of the L-BFGS algorithm with and without removing discontinuities by smooth functions. A nondifferentiable optimization (Lemaréchal 1978, 1989; Lemaréchal and Sagastizabal 1997) is employed to find the minimum of a cost function for cases when the L-BFGS algorithm fails to find the true minimum. Then a physical parameterization scheme [the shallow convection scheme in the National Centers for Environmental Prediction (NCEP) global spectral model] is used to examine the performance of both the differentiable and the nondifferentiable minimization algorithms (section 3). Discussion and conclusions are presented in section 4.

2. A discontinuous cost function using an idealized simple model with ''on–off'' switches

a. A nonsmooth cost function defined by a discontinuous model

We use an idealized model with a single variable and a typical form of discontinuous physics to examine the behavior of a differentiable minimization to solve nonsmooth problems. The numerical forecast model takes the form of

$$\frac{\partial x}{\partial t} = \begin{cases} f_1(x) & \text{if } x < x_c \\ f_2(x) & \text{if } x \geq x_c, \end{cases} \tag{1}$$

where $f_1(x)$ and $f_2(x)$ represent two parameterized physical processes identified by a threshold value x_c.

We can consider a cost function defined as the quadratic form of the model-predicted state at time t_R:

$$J_1(x_0) = x^2(t_R) \quad \text{and} \tag{2}$$

$$\frac{\partial x}{\partial t} = \begin{cases} 2x - 2 & x < 1 \\ x - 4 & x \geq 1, \end{cases} \tag{3}$$

where $x_0 = x(t_0)$ and $t_R > t_0$.

The discretization of the forecast model is carried out by using a time step $\Delta t = 0.1$. The length of the assimilation time window is $t_R = t_0 + 4\Delta t$.

b. Performance of the L-BFGS algorithm

The performance of the limited-memory quasi-Newton (L-BFGS) method of Liu and Nocedal (1989) is examined with different initial guess values for x_0.

Figure 1 shows that the minimization using the L-BFGS method converged with the initial guess of $x_0^{(0)} = 2.00$ (Fig. 1a), $x_0^{(0)} = 2.29$ (Fig. 1b), $x_0^{(0)} = 2.80$ (Fig. 1c), or $x_0^{(0)} = 2.90$ (Fig. 1d). However, the mini-

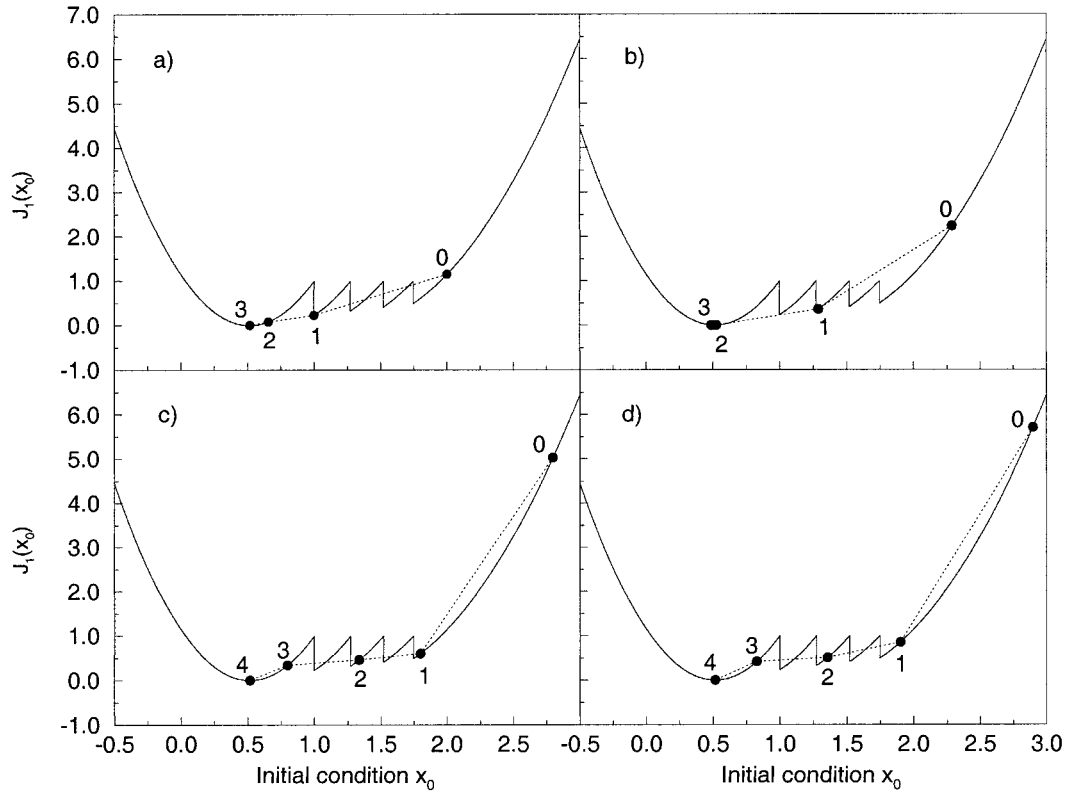

FIG. 1. The performances of the L-BFGS algorithm for minimizing the cost function J_1 defined in (2) and (3), a piecewise differentiable cost function, starting from different initial guess points: (a) $x_0 = 2$, (b) $x_0 = 2.29$, (c) $x_0 = 2.80$, and (c) $x_0 = 2.90$. The time window includes four-step time steps, i.e., $t_R = t_0 + 4\Delta t$, $\Delta t = 0.1$. The cost function (solid curve) is obtained by evaluating the function at the different initial conditions with an interval of 0.01. The numbers in the figures are the iteration numbers and the black dots represent the value of J obtained at each iteration.

mization failed to find the true minimum $x_0^* = 0.5$ if it starts with an initial guess of $x_0^{(0)} = 2.55$ (Fig. 2). After about five iterations, the minimization was trapped in a local minimum generated by the on–off switches, causing the iterative procedure to switch back and forth between $x_0^{(0)} = 1.521$ ($J = 0.41$) and $x_0^{(0)} = 1.518$ ($J = 0.99$).

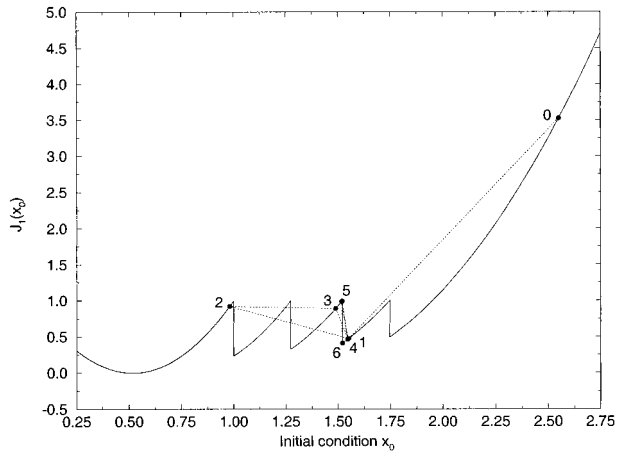

FIG. 2. Same as Fig. 1 except for the minimization starting from the initial guess point $x_0 = 2.55$.

In order to gain some insight into the general performance of the L-BFGS method, we selected 300 points in the interval [0, 3] for $x_0^{(0)}$, incremented by 0.01 for the successive initial guess points. Numerical results are presented in the second row of Table 1. The L-BFGS minimization failed to find the true minimum in 18 cases. In the other 282 cases, it converged to the true solution.

We found that the total number of minimization failures depends strongly on the size of the jump in the forecast model and, thus, the size of the jump in the cost function (see Fig. 3). When $f_2(x) = x - 4$ is replaced with $f_2(x) = x - 3.5$ (i.e., the size of the jump is reduced, see Fig. 3), the total number of failing cases was reduced from 18 to 13. If the size of the jump is further reduced, to say $f_2(x) = x - 3$, the L-BFGS algorithm successfully finds the solution ($x_0 = 0.5$) for the entire 300 cases tested while increasing the jump size generally leads to more failing cases.

This example suggests that the differentiable mini-

TABLE 1. The performance of the L-BFGS method.

$f_2(x)$	Jump size in the model	Ratio between the failed and successful cases		
		No smoothing	Weak smoothing	Strong smoothing
x-4.5	3.5	37/300	54/300	0/300
x-4.4	3.4	24/300	54/300	0/300
x-4.3	3.3	28/300	54/300	0/300
x-4.2	3.2	19/300	44/300	0/300
x-4.1	3.1	15/300	52/300	0/300
x-4.0	3.0	18/300	37/300	0/300
x-3.5	2.5	13/300	20/300	0/300
x-3.0	2.0	0/300	18/300	0/300
x-2.0	1.0	0/300	18/300	0/300
x-1.5	0.5	0/300	19/300	0/300

mization may still work well for solving nondifferentiable 4DVAR problems, especially when the sizes of the jumps caused by discontinuous physics are small. However, it is possible that the differentiable minimization may remain stuck with a local minimum and fail to find the true minimum. The possibility of encountering a minimization failure in the presence of a discontinuity is greater when the forecast model contains larger jumps.

c. *Introducing a local smooth function to remove discontinuities*

Given the fact that the presence of discontinuities associated with on–off switches in the 4DVAR assimilation model may cause difficulties for the minimization to find the true minimum, a natural step to remedy this situation is to introduce a smooth transitional function to remove the discontinuities (Zupanski and Mesinger 1995; Tsuyuki 1997). For example, a function

$$f_{\text{smooth}} = 0.5\{1 + \tanh[\alpha(x - x_c)]\} \tag{4}$$

is introduced into the assimilation model at threshold points, where α is a scalar that controls the accuracy of the smoother. The transition is implemented by calculating $f_{\text{transition}}$ as

$$f_{\text{transition}}(x) = \begin{cases} \dfrac{f_1(0.9) + f_2(1.1)}{2} + \left[\dfrac{f_1(0.9) + f_2(1.1)}{2} - f_1(x)\right](2f_{\text{smooth}} - 1) & \text{if } x_c - h \le x < x_c \\ \dfrac{f_1(0.9) + f_2(1.1)}{2} - \left[\dfrac{f_1(0.9) + f_2(1.1)}{2} - f_2(x)\right](2f_{\text{smooth}} - 1) & \text{if } 1 \le x < x_c + h \end{cases} \tag{5}$$

and defining the forecast model after smoothing as

$$\frac{\partial x}{\partial t} = f(x)$$

$$f(x) = \begin{cases} f_1(x) & \text{if } x < x_c - h \\ f_{\text{transition}}(x) & \text{if } x_c - h \le x < x_c + h \\ f_2(x) & \text{if } x \ge x_c + h, \end{cases} \tag{6}$$

where h is a small positive scalar. Choosing $h = 0.1$ and $\alpha = 100$ (a weak smoothing), the introduction of the transitional smooth function produces a smooth distribution of the source term $f(x)$ (thick dotted line in Fig. 4), and a continuous distribution of the cost function in the control variable space (dotted line in Fig. 5a). However, the introduction of the smooth function into the forecast model changes the distribution of the gradient of J with respect to the control parameter (Fig. 5b), introducing additional stationary points. The cost function without smoothing has a unique stationary point (zero gradient) that corresponds to the true solution ($x_0 = 0.5$) (solid line in Fig. 5b), while the one with smoothing has two extra stationary points for each jump (dotted line in Fig. 5b). Therefore, although the gradient may be discontinuous when the on–off switches

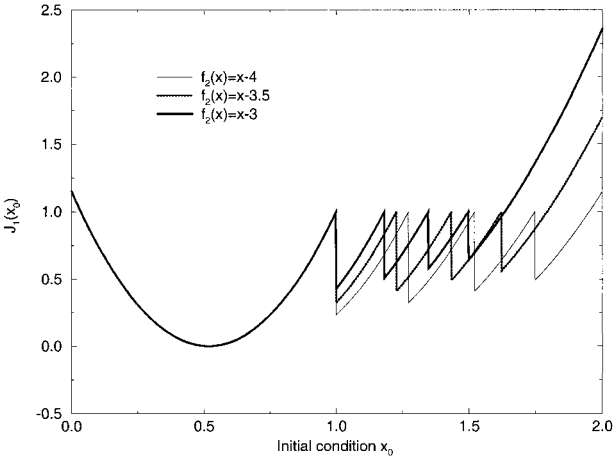

FIG. 3. The distributions of the cost function J_1 defined in (2) through (1) with $f_1 = 2x - 2$ and $f_2(x) = x - 4$ (thin solid line), $f_2(x) = x - 3.5$ (thick dotted line), and $f_2(x) = x - 3$ (thick solid line).

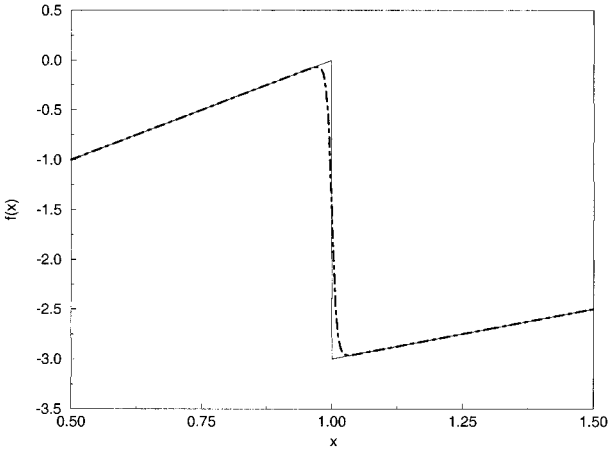

FIG. 4. Distributions of the model forcing $f(x)$ [see Eq. (6)] with (dotted line) and without (solid line) smoothing when $h = 0.1$ and $\alpha = 100$.

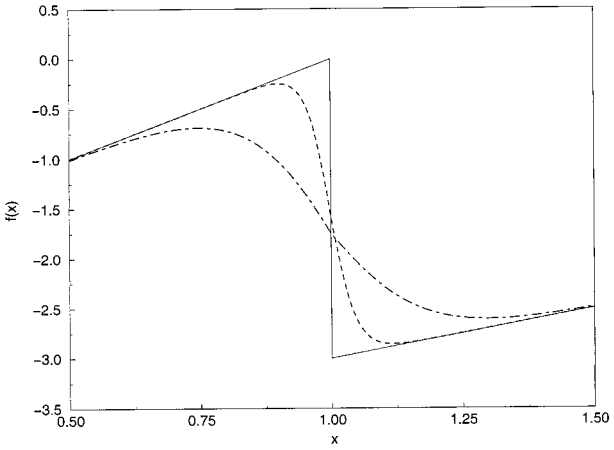

FIG. 6. Same as Fig. 4 except for $h = 0.2$, $\alpha = 20$ (thick dashed line) and $h = 0.5$, $\alpha = 5$ (thick dotted–dashed line).

in the adjoint model are kept the same as in the nonlinear model, such a discontinuity in the gradient does not change the general convexity feature, and the adjoint model integration provides useful subgradient information. However, a smooth function that seems to remove the discontinuity may introduce false stationary points, which may render the minimization to converge to a wrong solution (Zou et al. 1993).

We indicate, however, that when a very strong smoothing is applied (e.g., the value of α is reduced and that of h is increased), the zigzag behavior of the cost function can be eliminated and the artificial stationary points will not exist. Figure 6 shows the distributions of the smoothed $f(x)$ if $h = 0.2$, $\alpha = 20$ (thick dashed line), and $h = 0.5$, $\alpha = 5$ (thick dotted–dashed line). The resulting cost function and gradient distributions are displayed in Figs. 7a and 7b (thick dashed line for $h = 0.2$, $\alpha = 20$ and thick dotted–dashed line for $h = 0.5$, $\alpha = 5$), respectively. It is found that with a strong smoothing, the zigzag behavior of the cost function is eliminated and the minimization of the smoothed cost function using a differentiable optimization algorithm performs well. But the impact of such a strong smoothing on model solutions needs to be investigated before it is applied in 4DVAR.

In order to test the performance of the L-BFGS method for the cost function in which the discontinuity is removed by a transitional smooth function, we repeated the experiments carried out in section 2b without the local smoothing function introduced at the switch point.

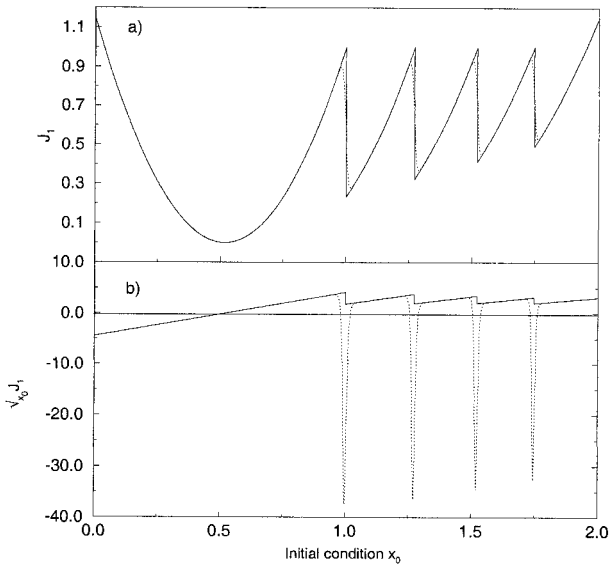

FIG. 5. Distributions of (a) the cost function and (b) the gradient with (dotted lines for $h = 0.1$ and $\alpha = 100$) and without (dotted lines) smoothing.

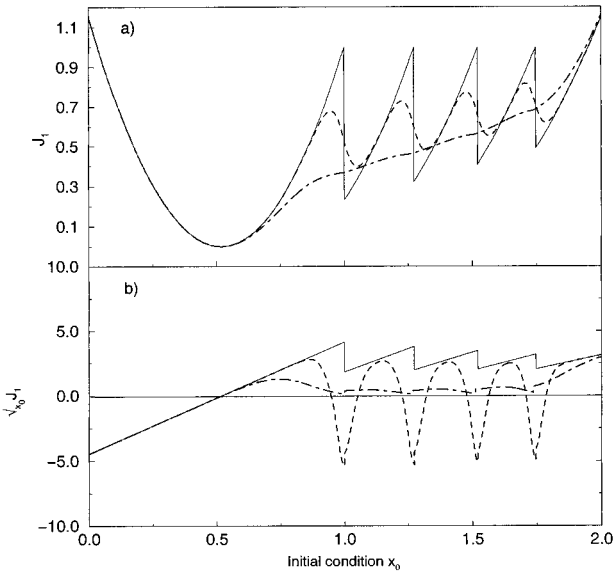

FIG. 7. Same as Fig. 5 except for $h = 0.2$, $\alpha = 20$ (thick dashed line) and $h = 0.5$, $\alpha = 5$ (thick dotted–dashed line).

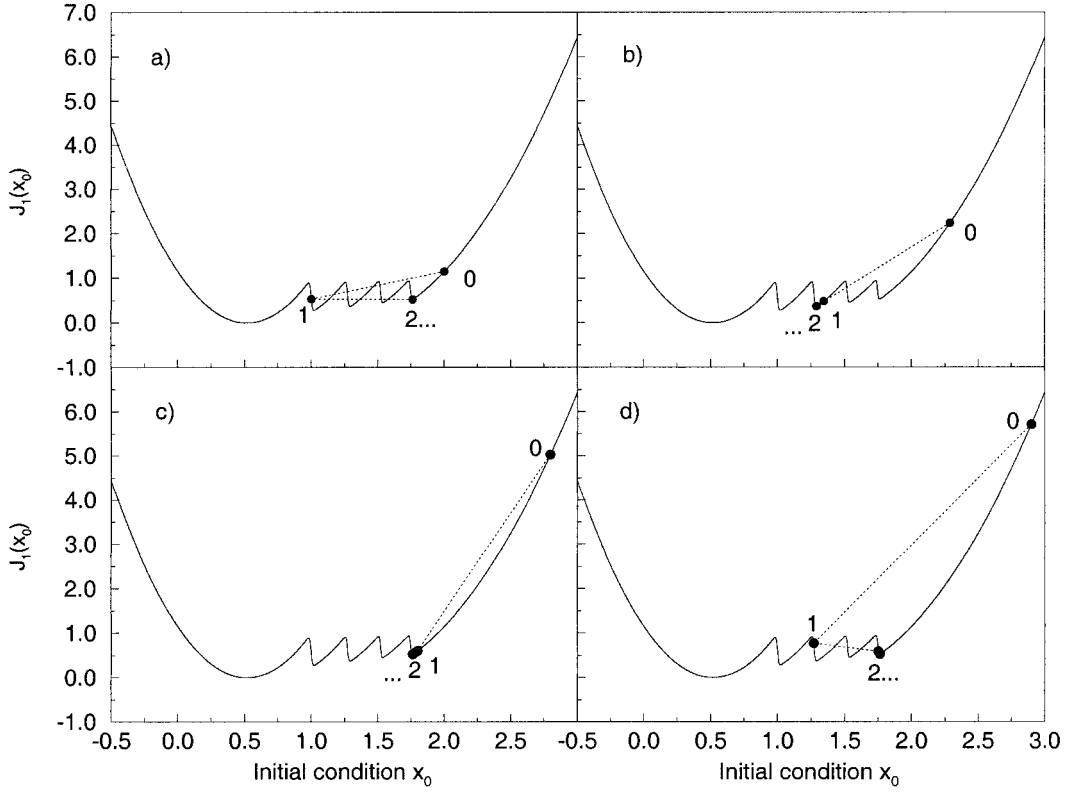

FIG. 8. Same as Fig. 1 except that the local smoothing [see Eq. (6)] is introduced at the switching point. A "2. . ." indicates the minimization process is stuck around the same point after two iterations.

Figure 8 displays the performance of the L-BFGS algorithm on the smoothed cost function with the same length of time integration and initial guess points as in Fig. 1. For all four cases examined, the minimization algorithm stuck at a local stationary point, instead of the true minimum.

When all 300 cases are repeated with a weak local smoothing, we find that there are cases for which the minimization with smoothing succeeds and the one without smoothing fails. The introduction of a weak smoothing increases the total number of failed cases, which is not observed in the case of strong smoothing. Table 1 summarizes the performances of the L-BFGS algorithm without smoothing, with a weak smoothing ($h = 0.1$ and $\alpha = 100$), and with a strong smoothing ($h = 0.5$ and $\alpha = 5$). Compared with the results without smoothing, the introduction of a weak smooth function increased the total number of failed cases by a factor of 2. For $f_2(x) = x - 3.0$, or $x - 2$, or $x - 1.5$, the introduction of a weak smooth function produces 18 failed cases for the L-BFGS method while the L-BFGS minimization had no problem finding the true minimum without smoothing. These results indicate that the local smoothing for discontinuous physics may sometimes do more damage than good in 4DVAR with discontinuous physics.

d. Performance of the bundle method for nonsmooth cost functions

Most differentiable minimization algorithms depend on two basic assumptions: (i) the negative gradient at a given point is the steepest direction and is used to approximate the search direction, and (ii) the cost function achieves a monotone and significant decrease along the search direction at each iteration. For a convex nonsmooth function, the direction negative to that of a subgradient is not always a direction of descent. The differentiable unconstrained minimization algorithms, such as L-BFGS, can fail in minimizing a nonsmooth cost function, as was shown in section 2 for the L-BFGS method. Nondifferential minimization algorithms, therefore, consider only the basic facts of convex analysis for convex functions, including nondifferential ones. There are various types of nonsmooth optimization algorithms. The basic mathematical considerations related to implementation of nonsmooth optimization are briefly described below.

1) BUNDLE METHODS IN NONSMOOTH OPTIMIZATION

A nondifferentiable minimization algorithm uses subgradients, called generalized gradients instead of gra-

dients, to attempt to force the function to decrease along the subgradient direction. A vector $\partial J(\mathbf{x}) \in \Re^n$ is called the subgradient of $J(\mathbf{x})$ at the point $\mathbf{x}$, if it satisfies

$$J(\mathbf{x} + \mathbf{y}) \geq J(\mathbf{x}) + \langle \boldsymbol{\partial} J(\mathbf{x}), \mathbf{y} \rangle, \qquad \forall \mathbf{y} \in \Re^n, \tag{7}$$

where $J(\mathbf{x})$ is a convex function in $\Re^n$, which is minimized. The subgradient method generalizes the gradient method and the quasi-Newton method for differentiable functions (Shor 1985; Kiwiel 1985).

At the nondifferentiable point, the cost function can be expressed as a selection among ''pieces'' [see (1) as an example]. Then a popular way to construct the subgradient is a linear combination of gradients corresponding to every piece $[\nabla_i J, i \in I(\mathbf{x})]$, as

$$\boldsymbol{\partial} J(\mathbf{x}) = \sum_{i \leq I(\mathbf{x})} \alpha_i \nabla_i J(\mathbf{x}),$$

$$\alpha_i \geq 0 \quad \text{and} \quad \sum_{i \leq I(\mathbf{x})} \alpha_i = 1. \tag{8}$$

The subgradient minimization for $J(\mathbf{x})$ on $\Re^n$ is an iterative process of the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \gamma_k \partial J(\mathbf{x}_k)/\|\partial J(\mathbf{x}_k)\|, \tag{9}$$

where $\gamma_k \geq 0$ is a step size.

For a single iterative point $\mathbf{x}_k$, $-\partial J(\mathbf{x}_k)$ still may not be a descent direction. A few further improvements finally form the bundle method. First, the search direction in the basic subgradient method given by

$$\mathbf{d}^{(k)} = -\boldsymbol{\zeta}_J^{(k)}/\|\boldsymbol{\zeta}_J^{(k)}\| \quad \text{where } \boldsymbol{\zeta}_J^{(k)} \equiv \partial J(\mathbf{x}_k) \tag{10}$$

is modified by introducing a space-dilation method (Kiwiel 1985; Shor 1985) as

$$\mathbf{d}^{(k)} = \frac{-\mathbf{B}^{(k)} \boldsymbol{\zeta}_J^{(k)}}{[\boldsymbol{\zeta}_J^{(k)\mathrm{T}} \mathbf{B}^{(k)} \boldsymbol{\zeta}_J^{(k)}]^{1/2}}, \tag{11}$$

where $\mathbf{B}^{(k)}$ is a space-dilation matrix that implements the space transformation on the subgradient to carry out a more efficient descent. Second, the stopping criterion in the line search has to be modified, since an arbitrary subgradient does not yield information about the optimality condition. The bundle methods (Lemaréchal 1977, 1978; Hiriatt-Urraty and Lemaréchal 1993) exploit the previous subgradient iterations by gathering subgradient information into a bundle to approximate the whole subdifferential $\partial J(\mathbf{x}^{(k)})$.

It is assumed that, in addition to current iteration point $\mathbf{x}^{(k)}$, there exist auxiliary points $\mathbf{y}^{(j)} \in \Re^n$ from past iterations and subgradients:

$$\boldsymbol{\zeta}^{(j)} \in \partial J(\mathbf{y}^{(j)}) \qquad j \in I_J^{(k)}, \tag{12}$$

where the index set $I_J^{(k)}$ is a nonempty subset of $\{1, 2, \ldots, k\}$. The search direction in the bundle algorithm is obtained as a solution of the quadratic optimization problem

$$\min_{\mathbf{d}, v} \left[\frac{1}{2} u^{(k)} \mathbf{d}^{\mathrm{T}} \mathbf{d} + v \right] \tag{13}$$

subject to $-\alpha_j[\mathbf{x}^{(k)}, \mathbf{y}^{(j)}] + [\boldsymbol{\zeta}_J^{(j)}]^{\mathrm{T}} \mathbf{d} \leq v \; [j \in I_J^{(k)}]$, where $u^{(k)} > 0$ is some weighting parameter and v is a constant term (scalar) in the quadratic form, and

$$\alpha_j[\mathbf{x}^{(\mathbf{x})}, \mathbf{y}^{(j)}] = J[\mathbf{x}^{(k)}] - J[\mathbf{y}^{(j)}] - [\boldsymbol{\zeta}_J^{(j)}]^{\mathrm{T}}[\mathbf{x}^{(k)} - \mathbf{y}^{(j)}]$$

denotes the linearization error at $\mathbf{x}^{(k)}$.

The next bundle iteration point is obtained via the following line search strategy:

Let $\mathbf{y}^{(k+1)} = \mathbf{x}^{(k)} + \lambda^{(k)} \mathbf{d}^{(k)}$ for some $\lambda^{(k)} > 0$.
Let $\boldsymbol{\zeta}_J^{(k+1)} \in \partial J[\mathbf{y}^{(k+1)}]$.
Then if $J[\mathbf{y}^{(k+1)}] \leq J[\mathbf{x}^{(k)}] - \delta^{(k)}$ for some $\delta^{(k)} > 0$,
(i) make a serious step $\mathbf{x}^{(k+1)} = \mathbf{y}^{(k+1)}$,
(ii) otherwise, make a null-step $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$.
In both cases add $\boldsymbol{\zeta}_J^{(k+1)}$ to the existing bundle.

2) NUMERICAL RESULTS

We choose to use the bundle method of Lemaréchal (1977, 1978) and compare its performance with that of the L-BFGS method. The subgradients (or subdifferentials), instead of gradients, are used in this method at nondifferentiable points. From (8), we know that this method using the subgradient takes smoothing on the gradient rather than the function itself in order to seek a decrease direction. The advantage of smoothing the gradient is that the original problem is not changed. But, we will see that more local gradients need to be evaluated in order to implement the smoothing. Therefore, the bundle method is much more expensive than the L-BFGS.

We arbitrarily choose four initial guess points from which the L-BFGS, both with and without local smoothing, failed to converge to the true solution. Figure 9 shows the performance of the bundle method for these cases chosen. It is encouraging to find that the bundle method converged to the true solution in all four cases. We then proceeded with more experiments. We repeated all the cases (a total of 49) for which either the L-BFGS without smoothing or the L-BFGS with smoothing failed. Numerical results are presented in Fig. 10. We observe that there are 18 cases where the L-BFGS method failed to find the true minimum, and 37 cases where the L-BFGS method with smoothing failed. There are six cases where both the L-BFGS without smoothing and the L-BFGS with smoothing failed. The bundle method succeeded in all 49 cases for which the L-BFGS with or without smoothing failed.

We mention, however, that there are 12 cases (out of 300) where the bundle method failed to find the true minimum and where the minimization using the L-BFGS method with and without smoothing was successful. This indicates that the performance of the smooth and nonsmooth optimization algorithms is case dependent. However, we have not found a case where both the L-BFGS and the bundle method failed.

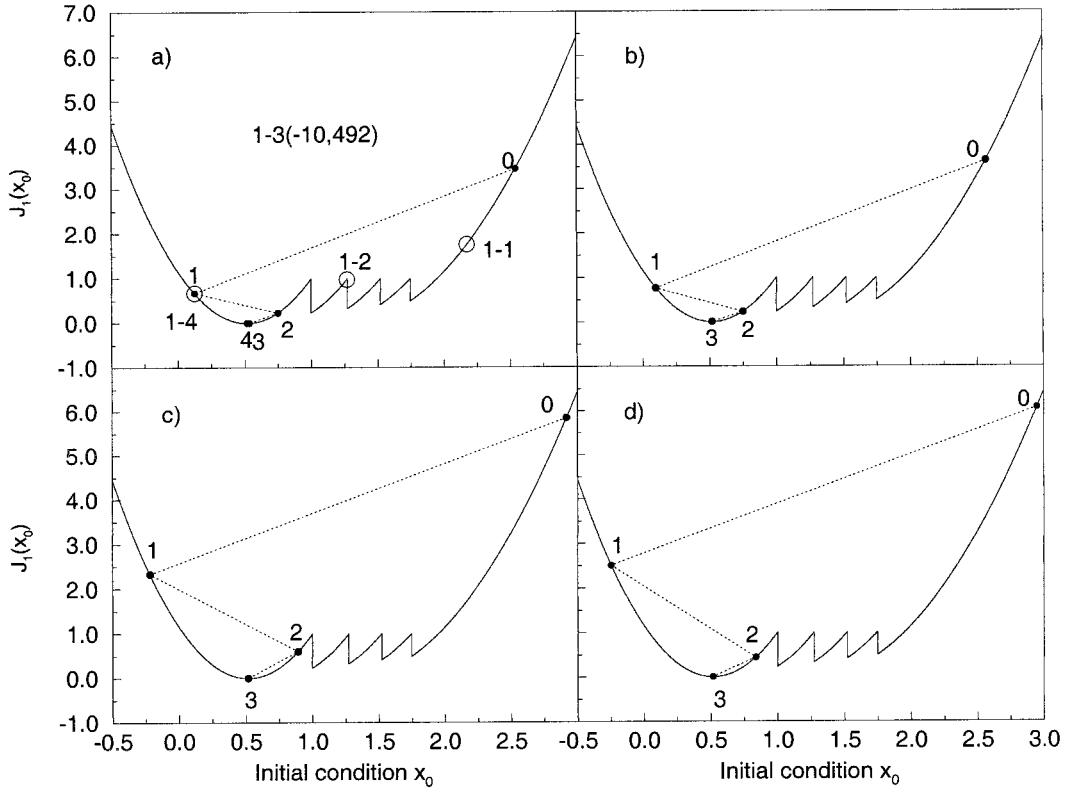

FIG. 9. Same as Fig. 1 except that the nondifferentiable bundle algorithm is used and the initial guess values of (a) $x_0 = 2.54$, (b) $x_0 = 2.57$, (c) $x_0 = 2.92$, and (d) $x_0 = 2.95$ are used. The L-BFGS with and without smoothing failed in all four of these cases.

3. A discontinuous cost function using a shallow-convection observation operator

a. Shallow convection

Shallow convection is sometimes called dry convection. In the 1960s, it was mainly used to allow convective adjustment in response to radiative heating so that a thermal equilibrium profile closer to the observation could be obtained (Manabe and Strickler 1964). In the 1980s, it began to be widely used in numerical modeling as a necessary compensation when deep convection does not occur (Betts 1986).

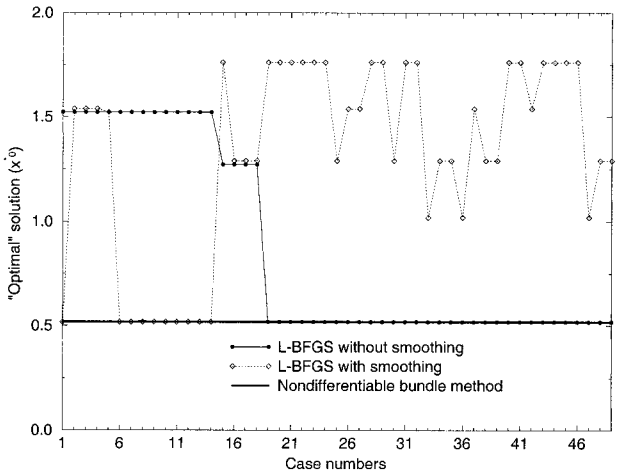

FIG. 10. The solutions obtained by the L-BFGS method with (diamonds connected by dotted line) and without (dots connected by thin solid line) and the bundle method (thick solid line) for the 49 cases for which the L-BFGS method with or without smoothing failed.

Shallow convection is a parameterization scheme that produces a vertical thermodynamical adjustment in the atmosphere. Unlike deep convection, which has a larger vertical scale due to water vapor convergence and condensational heating, shallow convection only deals with the vertical diffusion of unstable energy. It does not involve precipitation and condensation, and its vertical scale is rather small.

The diffusion equations used in the shallow convection of the NCEP spectral model are given by

$$\frac{\partial T}{\partial t} = \frac{1}{\rho}\frac{\partial}{\partial z}\left[K_{QT}\left(\frac{\partial T}{\partial z} + \Gamma\right)\right] \quad \text{and} \tag{14}$$

$$\frac{\partial q}{\partial t} = \frac{1}{\rho}\frac{\partial}{\partial z}\left(K_{QT}\frac{\partial q}{\partial z}\right), \tag{15}$$

where T is temperature, q is specific humidity, K_{QT} is the diffusion coefficient for the temperature and humidity, and Γ is the dry-adiabatic lapse rate. The discretized version of these diffusion equations forms a

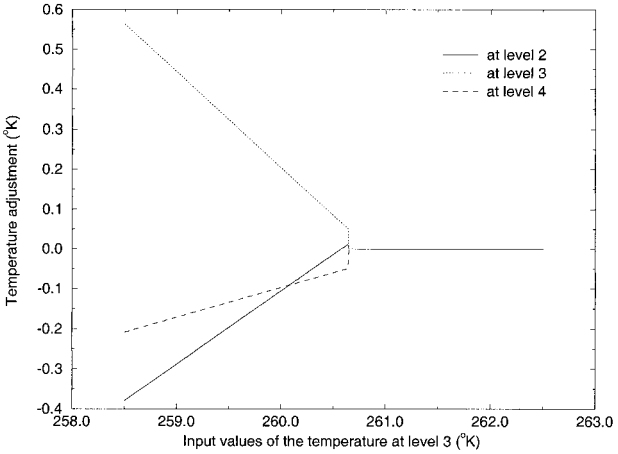

FIG. 11. Distributions of the temperature adjustments at model levels 2 (solid line), 3 (dotted line), and 4 (dashed line) due to shallow convection for various values of the input temperature at level 3. The temperature interval for the input temperature at level 3, at which these calculations are made is 0.01°C.

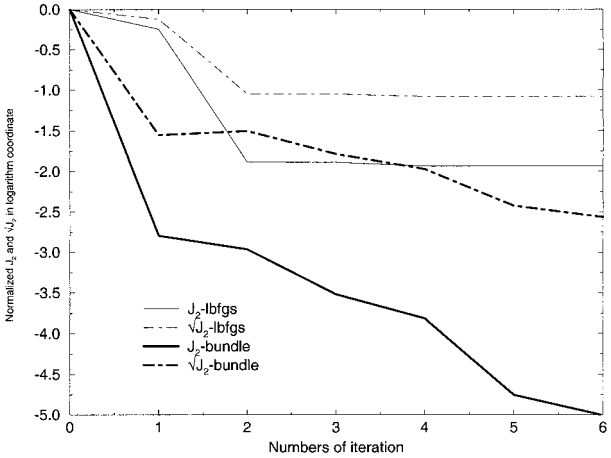

FIG. 12. The logarithmic variations of the normalized values of the cost function (solid lines) and the gradient (dashed lines) with the number of iterations using the L-BFGS (thin solid and dotted lines) and the bundle (thick solid and dotted lines) methods. The cost function is defined by (16) using shallow convection.

tridiagonal system that is solved using Gaussian elimination.

In the physical package of the NCEP global spectral model, all the unstable columns in which cumulus does not occur are picked up immediately following the Arakawa–Schubert cumulus scheme. We can summarize it in three steps.

1) Calculate the moist static energy ($gZ + C_pT + Lq$) and identify the columns with the conditional instability [$(\partial/\partial Z)(gZ + C_pT + Lq) < 0$]. Calculate the lifting condensation level as the cloud base and the highest instability level as the cloud top. Because of vertical discretization, the cloud base/top does not change continuously when the moist static energy profile is perturbed by the change of the thermodynamical condition.
2) Assign $K_{QT} = 1.5\ \text{m}^2\ \text{s}^{-1}$ at the base; $K_{QT} = 1.0\ \text{m}^2\ \text{s}^{-1}$, at the top; $K_{QT} = 3.0\ \text{m}^2\ \text{s}^{-1}$, for the next-to-top layers; and $K_{QT} = 5.0\ \text{m}^2\ \text{s}^{-1}$, for any other layers, preventing development of unrealistic kinks in the T and q profiles. The assignment of the different values of diffusion coefficient for the different cloud layer may cause discontinuities of the solution of the shallow-convection adjustment.
3) Gaussian elimination is then employed to solve the resulting tridiagonal system and the adjusted temperature and specific humidity profiles are obtained for the identified unstable columns.

The first two steps of the computational implementation of the shallow-convection parameterization may introduce discontinuities into the distribution of a cost function, defined by the shallow convection, with respect to the model temperature and specific humidity profiles.

Figure 11 shows, for example, the distribution of the temperature adjustments at the three model levels resulting from the shallow convection with various input values of the temperature at model level 3. The shallow-convection process is turned on and off by changing only the temperature at level 3 from 258.5 to 262.5 K at an interval of 0.01 K. The adjustment occurred at three levels: levels 2, 3, and 4. It is obvious that the

TABLE 2. Statistics on minimization of the L-BFGS and bundle methods for the example of shallow convection with 122d column as the initial guess.

	Function calls		Rms errors					
			T (°)		q (g kg^{-1})		J	
Iteration	L-BFGS	Bundle	L-BFGS	Bundle	L-BFGS	Bundle	L-BFGS	Bundle
0	—	—	2.456	2.456	0.415	0.415	8.613	8.613
1	1	2	1.794	0.387	0.315	0.012	4.919	0.014
2	1	5	0.711	0.211	0.091	0.009	0.112	0.009
3	3	1	0.709	0.172	0.091	0.008	0.112	0.003
4	1	2	0.677	0.083	0.088	0.006	0.100	0.001
5	2	5	0.677	0.030	0.088	0.002	0.100	1.5×10^{-4}
6	2	3	0.677	0.023	0.088	0.002	0.100	0.8×10^{-4}

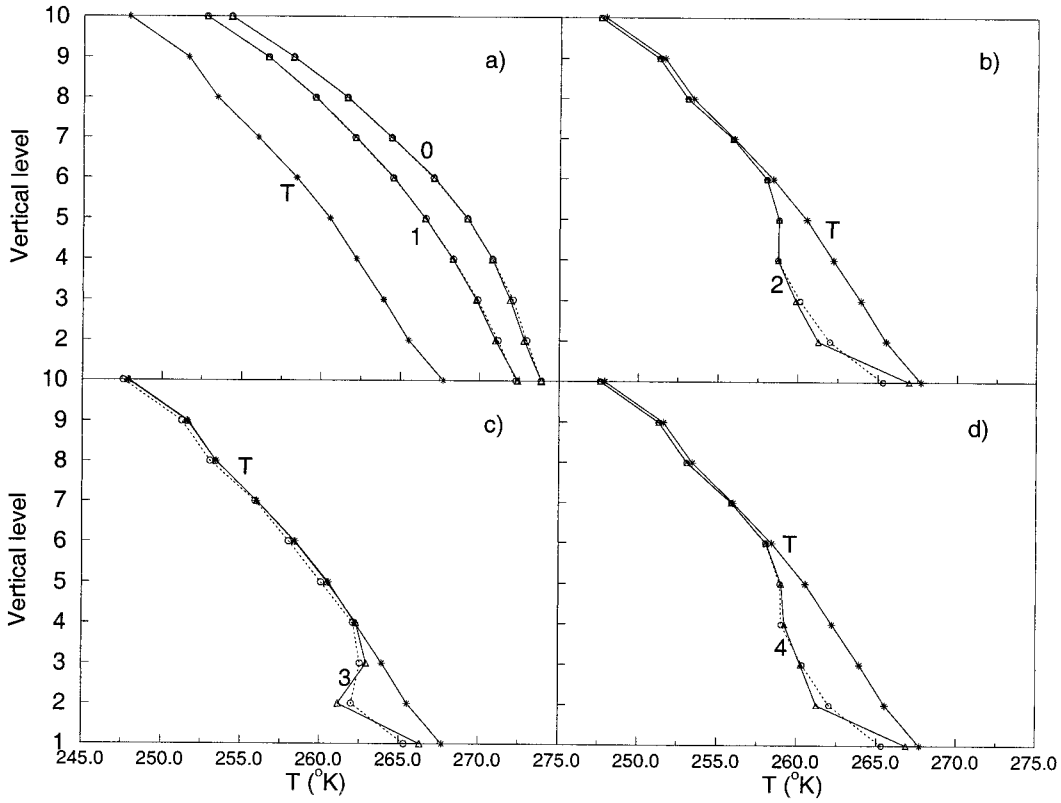

FIG. 13. The temperature profiles before and after the shallow-convection adjustments at (a) the zeroth and the first iteration, (b) the second iteration, (c) the third iteration, and (d) the fourth iteration using the L-BFGS method. The true temperature profiles, i.e., the profile that was used to generate observations using shallow convection, is presented in each panel (solid lines with stars).

solution of the shallow convection is discontinuous with respect to the input variable.

b. Experiment design and cost function

In order to test the performances of the L-BFGS and the nonsmooth optimization bundle algorithms using a realistic physical parameterization, the shallow-convection operator is used to define a twin experiment. The cost function is defined to measure the distance between the output of the shallow-convection operator (adjusted temperature and specific humidity profiles) and ''observations'' (the output temperature and specific humidity profiles for a selected input profiles of temperature and specific humidity, i.e., the true solution):

$$J_2(T, q) = [\mathbf{H}_T(T, q) - \mathbf{T}^{\text{obs}}]^{\text{T}}\mathbf{W}_T[\mathbf{H}_T(T, q) - \mathbf{T}^{\text{obs}}] + [\mathbf{H}_q(T, q) - \mathbf{q}^{\text{obs}}]^{\text{T}}\mathbf{W}_q[\mathbf{H}_q(T, q) - \mathbf{q}^{\text{obs}}], \quad (16)$$

where $\mathbf{W}_T$ and $\mathbf{W}_q$ are constant diagonal weighting matrices and their values are set to be 10^{-5} and 10^{6} em-

TABLE 3. Statistics on minimization of the L-BFGS and bundle methods for the example of shallow convection with 135th column as the initial guess.

			Rms Errors					
	Function calls		T (°)		q (g kg^{-1})		J	
Iteration	L-BFGS	Bundle	L-BFGS	Bundle	L-BFGS	Bundle	L-BFGS	Bundle
0	—	—	2.245	2.245	0.366	0.366	6.787	6.787
1	1	2	1.636	0.015	0.266	0.011	3.570	0.003
2	1	5	0.365	0.033	0.063	0.005	0.046	0.8×10^{-3}
3	4	11	0.365	0.022	0.063	0.004	0.046	0.2×10^{-3}
4	1	2	0.365	0.013	0.063	0.003	0.046	0.6×10^{-4}
5	1	2	0.365	0.009	0.063	0.002	0.046	0.2×10^{-4}
6	1	2	0.365	0.005	0.063	0.6×10^{-3}	0.046	0.4×10^{-5}

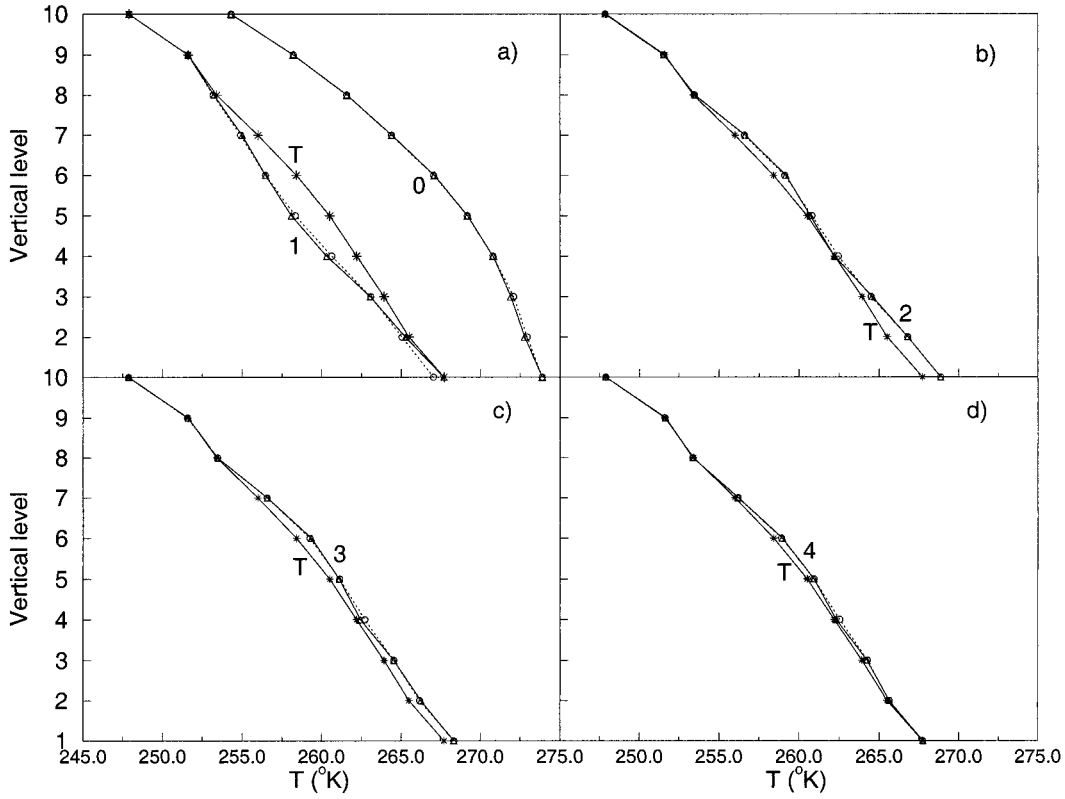

FIG. 14. Same as Fig. 11 except using the nondifferentiable bundle method.

pirically, corresponding to typical orders of simulated temperature and specific humidity errors of 1°C and 1 g kg^{-1}. The variables T and q in (16) are vectors of dimension K_0, which is equal to 10, the highest model level that may be affected by the shallow-convection process.

The gridded data of temperature (T), specific humidity (q), and the surface pressure (p_s) at 0000 UTC on 1 July 1995 from the NCEP reanalysis data are used for testing the performances of the L-BFGS method and the nonsmooth bundle method for minimizing J defined in (16). For a resolution of T62L28, there are a total of 47 $\times$ 384 columns over the entire global domain. We choose 384 columns corresponding to all the Gaussian grids around 12°N for the test.

c. Performances of the L-BFGS and bundle algorithms

Among the 384 columns, there are 51 columns in which the shallow-convection process is turned on. Within these 51 columns, we selected column 111 (near 150°W) as the truth, forming the observations in (16). Starting from the analysis profiles at the other 383 columns, we applied both the L-BFGS and bundle algorithms to approximate the ''true'' atmospheric profiles of temperature and specific humidity. The L-BFGS algorithm succeeded in all 380 cases except for three columns: 122 (at 132°W), 135 (at 107°W), and 145 (at 88°W). The bundle algorithm converged to the true solution for all 383 cases.

Tables 2–4 show the results of the minimization starting from the temperature and specific humidity profiles at the 122th, 135th, and 145th columns, respectively. Figure 12 shows the variations of the normalized cost function and the norm of the gradient (or subgradient) for the minimization starting from the temperature and specific humidity profiles at the 135th column. With six iterations the L-BFGS algorithm decreased the cost functions by about two orders of magnitude. The bundle algorithm decreased the cost functions by three to five orders of magnitude. The temperature and specific humidity profiles retrieved by the bundle method is much more accurate (more than an order of magnitude) than those retrieved by the L-BFGS method. The bundle method uses more function calls at each iteration than the L-BFGS method. The computational expenses of the bundle method is about twice that of the L-BFGS method.

Figures 13–16 present the temperature and specific humidity profiles obtained at each iteration using the L-BFGS method (Figs. 13 and 15) and the bundle method (Figs. 14 and 16). We find that the adjustment made by shallow convection to the input profiles of T and q (solid line with stars) for simulated observations occurred at model levels 3, 4, and 5, and were rather small

TABLE 4. Statistics on minimization of the L-BFGS and bundle methods for the example of shallow convection with 145th column as the initial guess.

	Function calls		Rms errors					
			T (°)		q (g kg^{-1})		J	
Iteration	L-BFGS	Bundle	L-BFGS	Bundle	L-BFGS	Bundle	L-BFGS	Bundle
0	—	—	2.162	2.162	0.315	0.315	4.950	4.950
1	1	2	1.402	0.317	0.215	0.010	2.268	0.012
2	1	5	0.434	0.186	0.058	0.010	0.031	0.004
3	2	3	0.413	0.139	0.057	0.007	0.029	0.001
4	2	2	0.407	0.103	0.056	0.006	0.028	0.9×10^{-3}
5	1	2	0.301	0.088	0.051	0.006	0.021	0.8×10^{-3}
6	3	4	0.292	0.077	0.050	0.005	0.019	0.6×10^{-3}

(not shown). After the second iteration, the L-BFGS minimization was stuck in a local minimum with switching turned on and off on several model levels, and had difficulty getting free and approaching the true minimum. Such a phenomenon was not seen with the bundle algorithm. Within four iterations, the minimization results approximated the true solution closely. By six iterations, differences between the minimization of the retrieved and the observed profiles of temperature and specific humidity were negligible. This is due to the bundle algorithm property: in the first few iterations the algorithm collects information about the cost function by bundling the subgradients. After four to five iterations, the subgradient bundle approximates the whole generalized gradient well and the ''optimal'' descent direction is generated.

In order to ensure a sufficient decrease in the value of the cost function, the bundle method evaluates many subgradients at each iteration. For example, 10 and 19 function calls, respectively, are required by L-BFGS and the bundle method to complete the six iterations. Therefore, this algorithm is much more expensive than the L-BFGS algorithm.

4. Summary and conclusions

The cost function in 4DVAR using a diabatic assimilation model with parameterized physics is only piece-

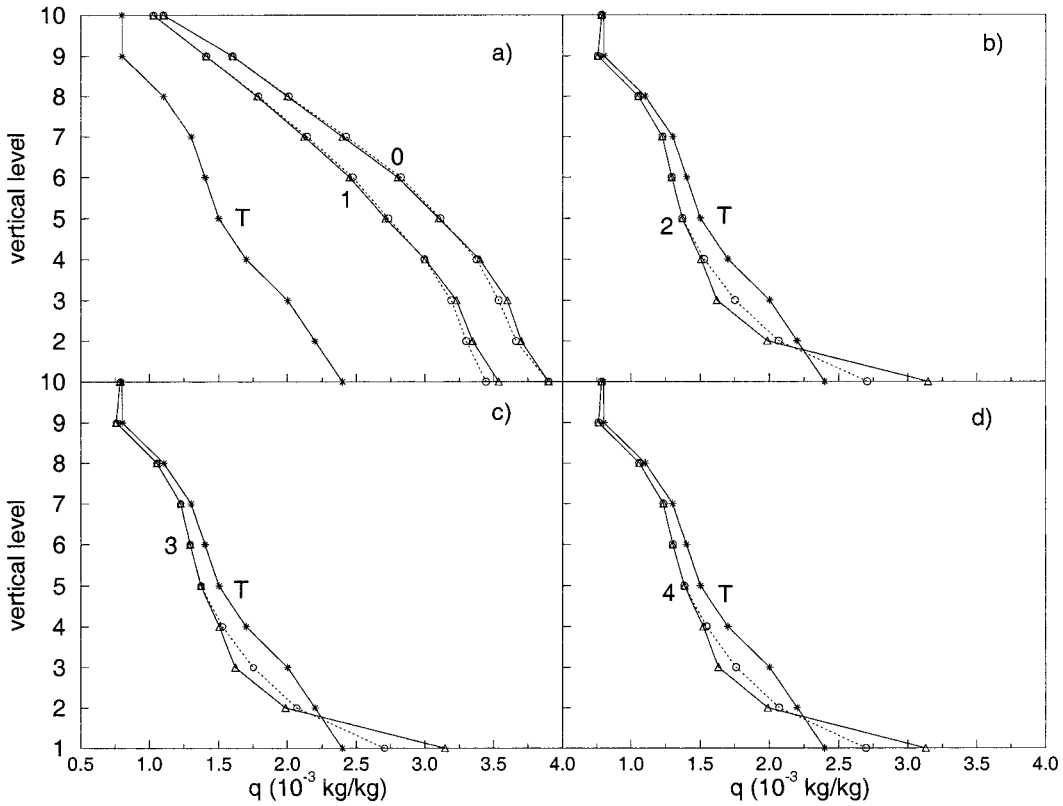

FIG. 15. Same as Fig. 11 except for the specific humidity variable.

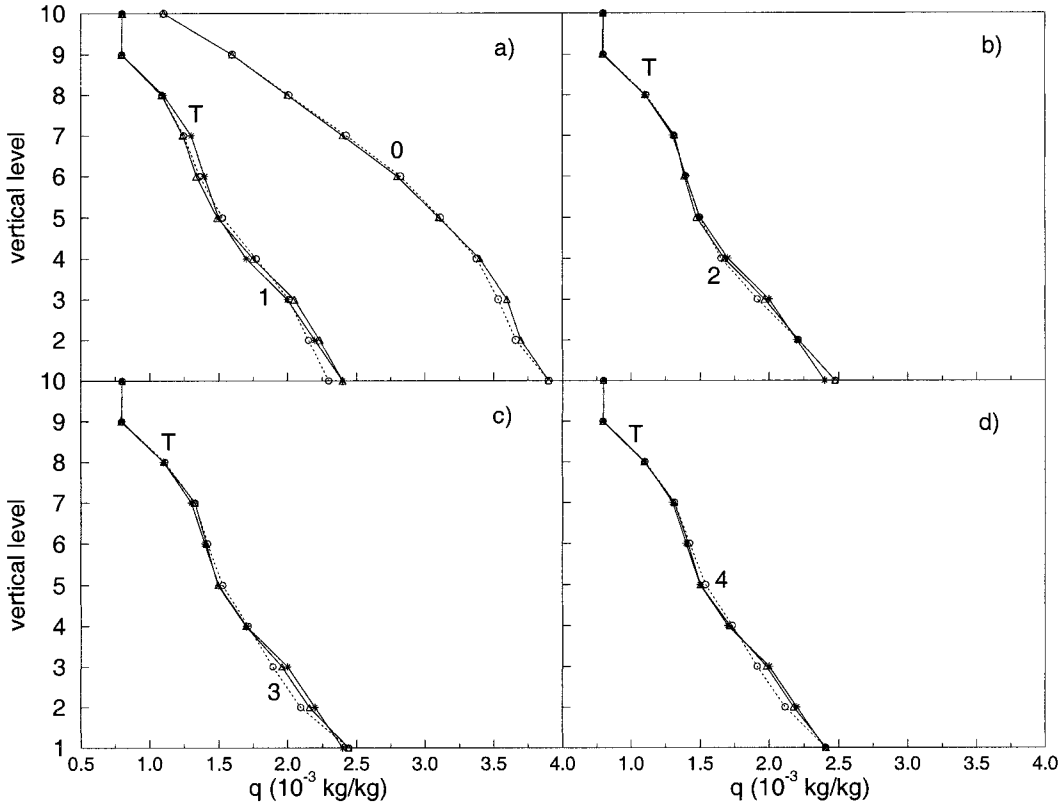

FIG. 16. Same as Fig. 12 except for the specific humidity variable.

wise differentiable. The differentiable minimization algorithms such as the limited memory quasi-Newton method that are originally designed for minimizing differentiable functions can fail. Using both a simple example and a shallow-convection scheme, we found the following. (i) The L-BFGS algorithm may still work well for minimizing nonsmooth cost functions. If the jumps in the cost function caused by discontinuous physics (controlled by on–off switches) are large, the minimization may converge to a local minimum introduced by the on–off switches. (ii) Introducing a weak smooth function to remove the discontinuities associated with on–off switches may do more damage than help to data assimilation results. The smoothing could introduce artificial stationary points, which may cause a minimization to converge to a wrong solution. (iii) The nondifferentiable bundle method performs well for minimizing nonsmooth cost functions, although it is computationally twice as expensive as the L-BFGS method. Although a strong smoothing does not cause problems for the differentiable minimization in converging and may be applied, the consequence of such a strong smoothing on changing the prediction of atmospheric state needs to be studied beforehand.

We have used simple models for examining the performance of differentiable and nondifferentiable optimization algorithms for discontinuous cost functions. We plan to test the feasibility of applying the nondifferentiable minimization algorithm, such as the bundle method (Lemaréchal 1989), to 4DVAR using the NCEP adiabatic model and comparing the accuracy of the solution with that using the differentiable minimization methods.

Acknowledgments. This research is supported by NOAA Grant NA37WA0361 and NSF Grant ATM-9812729. The authors would like to thank Dr. E. Kalnay for her persistent encouragement on this study. We also thank Dr. Claude Lemaréchal from INRIA for providing to us the software for his bundle method. The original bundle algorithm was modified by Profs. Navon and Nazareth to fit the test problems.

REFERENCES

Betts, A. K., 1986: A new convective adjustment scheme. Part I: Observational and theoretical basis. *Quart. J. Roy. Meteor. Soc.*, **112,** 677–691.

Broyden, C. G., 1970: The convergence of a class of double rank minimization algorithms. Parts I and II. *J. Inst. Maths. Appl.*, **6,** 76–90.

Fletcher, R., 1970: A new approach to variable metric algorithms. *Comput. J.*, **13,** 312–322.

Goldfarb, D., 1970: A family of variable metric methods derived by variational means. *Math. Comput.*, **24,** 23–26.

Hiriart-Urruty, J.-B., and C. Lemaréchal, 1993: *Convex Analysis and Minimization Algorithms. II: Advanced Theory and Bundle Methods.* Vol. 306, Springer-Verlag, 346 pp.

Kiwiel, K. C., 1985: *Methods of Descent for Nondifferentiable Optimization.* Lecture Notes in Mathematics, Vol. 1133, Springer-Verlag, 362 pp.

Kuo, Y.-H., X. Zou, and Y.-R. Guo, 1996: Variational assimilation of precipitable water using a nonhydrostatic mesoscale adjoint model. Part I: Moisture retrieval and sensitivity experiments. *Mon. Wea. Rev.,* **124,** 122–147.

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

Lemaréchal, C., 1977: Bundle methods in nonsmooth optimization. *Proceeding of the IIASA Series,* C. Lemaréchal and R. Mifflin, Eds., Pergamon Press, 79–103.

——, 1978: Nonsmooth optimization and descent methods. International Institute for Appl. Sys. Analysis, Laxenburg, Austria, 25 pp.

——, 1989: Nondifferentiable optimization. *Optimization,* G. L. Nemhauser et al., Eds., Handbooks in ORGMS, Vol. 1, Elsevier Science, 529–572.

——, and C. Sagastizabal 1997: Variable metric bundle methods: From conceptual to implementable forms. *Math. Programming,* **76,** 393–410.

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

Manabe, S., and R. F. Strickler, 1964: Thermal equilibrium of the atmosphere with a convection adjustment. *J. Atmos. Sci.,* **21,** 361–385.

Navon, I. M., 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.

Rabier, F., J.-N. Thepaut, and P. Courtier, 1998: Extended assimilation and forecast experiments with a four-dimensional variational assimilation system. *Quart. J. Roy. Meteor. Soc.,* **124,** 1861–1887.

Shanno, D. F., 1970: Conditioning of quasi-Newton methods for function minimization. *Math. Comput.,* **24,** 647–657.

Shor, N. Z., 1985: *Minimization Methods for Nondifferentiable Functions.* Springer-Verlag, 162 pp. (Translated from Russian by K. C. Kiwiel and A. Ruszczynski.)

Thépaut, J. N., and P. Courtier, 1991: 4-dimensional data assimilation using the adjoint of a multilevel primitive equation model. *Quart. J. Roy. Meteor. Soc.,* **117,** 1225–1254.

Tsuyuki, T., 1997: Variational data assimilation in the tropics using precipitation data. Part III: Assimilation of SSM/I precipitation rates. *Mon. Wea. Rev.,* **125,** 1447–1464.

Zou, X., 1997: Tangent linear and adjoint of ''on–off'' processes and their feasibility for use in 4-dimensional variational data assimilation. *Tellus,* **49A,** 3–31.

——, and Y.-H. Kuo, 1996: Rainfall assimilation through an optimal control of initial and boundary conditions in a limited-area mesoscale model. *Mon. Wea. Rev.,* **124,** 2859–2882.

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

Zupanski, D., 1993: The effects of discontinuities in the Betts–Miller cumulus convection scheme on four-dimensional variational data assimilation in a quasi-operational forecasting environment. *Tellus,* **45A,** 511–524.

——, and F. Mesinger, 1995: Four-dimensional variational assimilation of precipitation data. *Mon. Wea. Rev.,* **123,** 1112–1127.