The spectral conjugate gradient method in variational adjoint assimilation for model terrain correction II: Numerical test

Sulin Tao¹²⁺, Yuhong Li³, Isaac Mugume⁴ and Shuanghe Shen¹

¹Key Laboratory of Meteorological Disaster, Ministry of Education (KLME), Joint International Research Laboratory of Climate and Environment Change (ILCEC), Collaborative Innovation Centre on Forecast and Evaluation of Meteorological Disasters (CIC-FEMD), School of Applied Meteorology, Nanjing University of Information Science & Technology, Nanjing 210044, P.R.China
²Technology Development Centre, Nanjing Daqiao Machine CO.,Ltd, Nanjing 211101, P.R.China
³Liaoning Institute of Meteorological Sciences, Shenyang 110166, P.R.China
⁴Department of Geography, Geo-Informatics & Climatic Sciences, Makerere University, P.O. Box 7062, Kampala, Uganda

* Corresponding author e-mail: sulin.tao@outlook.com

Abstract. The performance of the spectral conjugate gradient (SCG) method proposed in part I of this paper is evaluated by comparison with that of the limited-memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) method and Hager and Zhang’s CG DESCENT method. The step length is determined by solving the tangent linear model and a line search algorithm with cubic interpolation is introduced for comparison. Numerical tests indicate that the SCG method is effective at correcting the bottom terrain. It is robust, providing small root mean square (RMS) errors and smooth profiles for the corrected bottom terrain for all tests conducted. It has much higher optimizing efficiency than Hager and Zhang’s CG DESCENT and Liu’s LBFGS methods. The SCG method reduces the norm of the gradient sufficiently and the value of the cost function very quickly. It always provides linear convergence rates, especially when including a regularization term. This study suggests the combined utilization of the SCG method and line search strategy to ensure high efficiency when solving the large-scale unconstrained problems.

1. Introduction

The bottom terrain figures prominently in controlling atmospheric circulation in numerical models. The direct measurement of bottom terrain in the grids of the prediction models may be difficult, especially when the bottom terrain slowly changes over time, such as the underlying surface of moving sand dunes, lake and ocean. Numerical models require optimal topographic data, as mismatches between bottom topography and meteorological fields will lead to model errors [1, 2].

In part I of this paper, we provide a spectral conjugate gradient (SCG) method within the mathematical framework of a variational adjoint assimilation system to correct the bottom terrain of a shallow-water equations model. We describe the formulation of this method from the mathematical point of view with determination of the descent direction by using Andrei’s limited-memory form and the step length by solving the tangent linear model.
The objective of part II of this paper is to provide the SCG method with Andrei’s formulation [3] and evaluate its performance within the variational adjoint assimilation system by addressing (i) the influence of sparse and noisy observations and (ii) the influence of different types of the initial guess. We evaluate the performance of the SCG method by comparison with that of the LBFGS method [4] and the CG_DESCENT method [5]. The difference between LBFGS, CG_DESCENT and SCG, as considered in our study, mainly involves the updating type of the descent direction. Tests might help us to understand the SCG method in detail, especially the positive effect of the special properties that inherits Andrei’s limited-memory formulation and adopts an optimal step length. The potential of this method for inverse problems could also be confirmed in such a way.

2. Description of the descent algorithms tested

We evaluate the performance of the SCG method and introduce the LBFGS method with \( m = 3, 5 \) and 7 updates [4] and the CG_DESCENT method [5]. They have the different updating type for the descent direction compared with the SCG algorithm. Note that 3 \( \leq m \leq 7 \) is sufficient and \( m > 7 \) usually does not improve the performance of the LBFGS method [6]. More detailed information on LBFGS can be found in a previous publication [7]. The parameter in relation to the truncation operation in CG_DESCENT is defined as \( 1.0 \times 10^5 \) for infrequent truncation [5]. We also introduce a line search algorithm with cubic interpolation [8] for comparison. Thus we construct the codes with the step length

1. by solving the tangent linear model, or
2. by using the line search algorithm.

Ten types of codes are constructed correspondingly (Table 1). For the LBFGS1 codes, the step size is initiated always as one when using line search algorithm, while for the CG_DESCENT1 and SCG1 codes, \( 1 / \| \nabla \mathbf{w}^i \| \) is used as the initial step length and at every iteration \( i \geq 1 \) the starting guess for the step length is defined as \( l^{i+1} = \| \mathbf{d}^i \| / \| \mathbf{d} \| \). In addition, the stopping criterions for these codes are the same as in step 9 of algorithm SCG.

| Algorithm          | Step length                      | Algorithm          | Step length                      |
|--------------------|----------------------------------|--------------------|----------------------------------|
| SCG                | Solving the tangent linear model  | SCG1               | Using the line search            |
| CG_DESCENT m=3,5,7 | LBFGS m=3,5,7                    | CG_DESCENT1        | LBFGS1 m=3,5,7                   |

The tables below show the experiments description (Exp), the number of iterations (Iter), the number of function and gradient evaluations (Nfg), the CPU time used to satisfy the prescribed convergence criteria (CPU), and the root mean square (RMS) error. The results in the corresponding tables are presented in the form

\[
\text{SCG/ CG_DESCENT/LBFGS (m=3)/ LBFGS (m=5)/ LBFGS (m=7) / SCG1/ CG_DESCENT1/ LBFGS1 (m=3)/ LBFGS1 (m=5)/ LBFGS1 (m=7).}
\]

3. Definition of the true bottom terrain tested

The bottom terrain considered in part I of this paper is defined as a ridge located in the middle of the horizontal domain. The shapes of terrain in the real world are much more complex. Thus, in the rest experiments, we introduce three other complex forms of the true bottom terrain:

1. a true bottom terrain with a valley shape,
2. a true bottom terrain with a basin shape, and
3. a true bottom terrain with a plateau shape,
respectively defined by
4. Accuracy evaluation of the adjoint model

The model-generated observations are prepared by running the forward model from $t_0$ to $t_N$ with the true bottom terrain and the initial values of the states. The full set of model-generated observations contains all of the observational fields available at all grid points and time steps:

$$D = \{ \tilde{u}_j, \tilde{v}_j, \tilde{\phi}_j | j = 1, 2, \ldots, J; k = 0, 1, \ldots, N \}.$$  

(2)

The dimension of the control variable is 100 for the treatment of model errors. The programs are run on a Core i3-2100 CPU, 3.10GHz clock and 2GB RAM memory.

It is critical for the adjoint model to prepare the gradient of the cost function for the minimization process. The accuracy of the discrete scheme of the adjoint model is verified via gradient test. Let

$$T J = J(\chi, H) + \xi \nabla J = J(\chi, H) + \xi (\nabla J)^\top \nabla J + O(\xi^2),$$

(3)

be a Taylor expansion of the cost function. By using the Taylor series of one order, we obtain the relation

$$\rho(\xi) = \left[ J(\chi, H + \xi \nabla J) - J(\chi, H) \right] / \left[ \xi (\nabla J)^\top \nabla J \right],$$

(4)

where $\rho$ is defined in terms of a small scalar $\xi$, the cost function and its gradient. If the values of $\rho(\xi)$ linearly approach one when we decrease $\xi$ within a certain range covering several orders of magnitude, we can conclude that the values of the cost function and its gradient are correctly computed. We then set the varying range of $\xi$ between $10^{-13}$ and $10^{-3}$ m$^2$ (Figure 1). This finding verifies the correctness of the gradient calculation of our schemes.
5. Determination of the regularization parameter

Theoretically, the optimal weight for the regularization term could be chosen in such a way that all terms in cost function are of the same order of magnitude, which means that $\tau$ should scale the regularization term to become the same order of magnitude of $\eta$ in Eq. (9) in part I of this paper. Here we intend to verify the effectiveness of this method by solving a sequence of unconstrained problems with increasing values of the regularization parameter $\tau$.

We define the relative error indicator $\text{Re}(\tau)$ between the corrected $H$ and known average terrain elevation $H_0$ for the determination of the regularization parameters in terms of $\tau$ and set a varying range for $\tau$ of $[10^{-6}, 10^6]$ m$^2$. Thus, for each value of the regularization parameter $\tau$, there is a corresponding relative error $\text{Re}(\tau)$ at final time of the correction procedure, and a curve can be obtained by plotting the relative errors $\text{Re}(\tau)$ against the log of the regularization parameter values $\tau$. The parameter value in relation to the point with minimum relative error of the corrected bottom terrain is chosen to be the optimal regularization parameter. According to this method, the regularization parameters are obtained for the following tests (Table 2).

| Terrain shape | SCG  | CG DESCENT | LBFGS $m=$ | SCG1 | CG DESCENT1 | LBFGSI $m=$ |
|---------------|------|-------------|-------------|------|-------------|-------------|
| Ridge         | 0.7  | 0.9         | 1.0         | 1.0  | 0.9         | 1.0         |
| Valley        | 1.0  | 0.7         | 1.0         | 1.0  | 1.0         | 1.0         |
| Basin         | 0.9  | 0.9         | 1.0         | 1.0  | 0.7         | 0.9         |
| Plateau       | 0.7  | 0.9         | 1.0         | 1.0  | 0.9         | 1.0         |

6. Numerical results

6.1. The influence of sparse and noisy observations

A low density of observation sites or the uncertainty of the measuring equipment may decrease the density of observations in the space and time dimensions, and the quality of the observations (e.g., noisy observations). Two important questions related to sparse observations concern the influence of the density of observations in space on the uniqueness of the numerical solutions and the influence of the density of observations in time on the convergence rate of the minimization procedure [9].

We consider two common phenomena related to meteorological observations:
(1) sparse observations in the space and time dimensions, and
(2) sparse observations with random noise.
We conduct the following experiments using the observational fields \( \mathbf{u}, \mathbf{v} \) and contaminating the sparse observations available at all grid points and every five time steps by adding random Gaussian noise with a mean of 0 m s\(^{-1}\) and a standard deviation (std) of \( 5.0 \times 10^{-3}, 1.0 \times 10^{-2}, 5.0 \times 10^{-2} \) or \( 1.0 \times 10^{-1} \) m s\(^{-1}\). The initial guess with the elevation at each grid is \( x_j (j = 1, \ldots, J - 1, J) \) equal to zero metres is considered. The RMS error between the initial guess and the true terrain is \( 1.633 \times 10^{-1} \) m.

The SCG method provides perfect bottom terrain corrections when the random noise was composed to the sparse observations (Table 3). It performs better than the CG_DESCENT method somewhat. The superior performance of the SCG method over the LBFGS codes is much more obvious from the requirements of the number of iterations, the number of function and gradient evaluations, the CPU time and the RMS error of the bottom terrain corrections. Moreover, it’s much more economical when the SCG and CG_DESCENT methods load the line search algorithm over the optimal step length.

The SCG and CG_DESCENT methods reach much higher conformity with the true bottom terrain than the LBFGS codes both when the line search algorithm and the optimal step length are used, though the LBFGS code with \( m = 7 \) updates can provide competitive corrections (Figure 2). The bottom terrain corrections provided by the SCG, SCG1, CG_DESCENT and CG_DESCENT1 methods fit well with the true one when the standard deviation of the random noise is not over \( 1.0 \times 10^{-2} \) m s\(^{-1}\). As the intensity of the random noise becomes large, their performance becomes worse, but their inversions can still provide the brief shapes of the true bottom terrain.

**Table 3.** Performance of the SCG, CG_DESCENT, LBFGS (\( m = 3/5/7 \)), SCG1, CG_DESCENT1 and LBFGS1 (\( m = 3/5/7 \)) methods for bottom terrain inversion on the basis of sparse observations with random noise.

| Exp | Noise Level | Iter | Nfg | CPU (s) | RMS Error (m) |
|-----|-------------|------|-----|---------|---------------|
| (1) | mean=0 m s\(^{-1}\), std=5.0 \times 10^{-3} m s\(^{-1}\) | 91/106/30 | 184/214/60 | 71.59/83.67/237.7 | 1.946E-3/1.958E-3/3.0E-3 |
|     | 0/300/300  | 2/602/602/236/223/25 | 8/240.03/237.68/6.36/35.28/48.49 | 136.26/95.68 | 2.7/3.84E-2/1.924E-3 |
|     | /51/50/91/300/210 | 4/602/423 | 1.984E-1/2.435E-2/5.339E-3 | 7.44/263.36/248.52/27.08/50.07/45.17 | 6.36/35.28/48.49/8/240.03/237.7 | 1.984E-1/2.435E-2/5.339E-3 |
| (2) | mean=0 m s\(^{-1}\), std=1.0 \times 10^{-2} m s\(^{-1}\) | 131/103/3 | 130/208/60 | 48.97/72.56/234.5 | 5.775E-3/5.532E-3/3.16E-2 |
|     | 0/300/300  | 2/602/602/236/223/25 | 1/207.18/214.02/126.67/85.67/25.0 10 | 3.16E-2 | 2.564E-2/9.639E-3/1.950E-2 |
|     | /54/43/9/236/217 | 2/602/411 | 6.36/35.28/56.15/1.950 10 | 5.339E-2 | 7.687E-2/5.025E-2/2.539E-2 |
| (3) | mean=0 m s\(^{-1}\), std=5.0 \times 10^{-2} m s\(^{-1}\) | 74/143/30 | 150/288/60 | 58.44/117.43/248.9 | 2.440E-2/2.429E-2/2.373E-2 |
|     | 0/300/300  | 2/602/602/236/223/25 | 44/263.36/248.52/27.08/50.07/45.17 | 1.984E-1/2.435E-2/5.339E-3 | 7.44/263.36/248.52/27.08/50.07/45.17 | 6.36/35.28/48.49/8/240.03/237.7 | 1.984E-1/2.435E-2/5.339E-3 |
|     | /38/64/9/157/150 | 0/347/323 | 8/240.03/237.7 | 3.16E-2 | 2.564E-2/9.639E-3/1.950E-2 |
| (4) | mean=0 m s\(^{-1}\), std=1.0 \times 10^{-1} m s\(^{-1}\) | 71/141/30 | 144/284/60 | 62.60/114.30/255.1 | 5.243E-2/5.144E-2/2.516E-2 |
|     | 0/300/300  | 2/602/602/236/223/25 | 76/293.77/254.86/26.24/33.71/61.95 | 7.86E-2/75.64 | 7.687E-2/5.025E-2/2.539E-2 |
|     | /32/41/12 | 140/193/28 | 2.516E-2/5.144E-2/2.539E-2 | 7.86E-2/75.64 | 7.687E-2/5.025E-2/2.539E-2 |
|     | 0/159/130 | 3/351/323 | 1.984E-1/2.435E-2/5.339E-3 | 7.44/263.36/248.52/27.08/50.07/45.17 | 6.36/35.28/48.49/8/240.03/237.7 | 1.984E-1/2.435E-2/5.339E-3 | 7.44/263.36/248.52/27.08/50.07/45.17 | 6.36/35.28/48.49/8/240.03/237.7 | 1.984E-1/2.435E-2/5.339E-3 |

The advantage of the SCG method is also clear from the variations of the values of the RMS error with the number of iterations (Figure 3). It reduces the values of the RMS error by approximately 2 orders of magnitude when the random noise level is small. This capability also can be found with the CG_DESCENT method and the LBFGS code with \( m = 7 \) updates coupled with the optimal step length. The reduction factors become small when the intensity of the random noise grows. However, the SCG and CG_DESCENT method can always reach the optimal solution within few iterations compared to the SCG and CG_DESCENT methods with the line search algorithm over the optimal step length.
Figure 2. The inversed bottom terrain with observations imposed by random noise. (a–d) Exp (1–4).

Figure 3. The trend of the RMS error between inversed bottom terrain and true terrain with the number of iterations and observations imposed by random noise. (a–d) Exp (1–4).
with the LBFGS codes. The superior performance of the SCG method is much clear in the tests when the tangent linear model provides the step length.

6.2. The influence of different types of initial guess

Usually the required bottom terrain data cannot be obtained directly. As mentioned above, these data are generated by interpolation from coarse-resolution data and then smoothed before use in the models. These data are generally not matched with the models, but we can treat these unmatched data as different initial guess types for our correction process.

Two definitions for our initial guess related to those conventional treatments are considered:

(1) that generated from true terrain using a moving average filter with a span of 51 grids and then multiplying by 0.5 (Definition 1, blue dashed line in Figure 4), which might generate an initial guess far from the true bottom terrain; and

(2) that generated from true terrain using a moving average filter with a span of 11 grids and then multiplying by 0.5 (Definition 2, orange dashed line in Figure 4).

We use the same observational fields $u$, $v$ and contaminate the sparse observations available at all grid points and every five time steps by adding random Gaussian noise with a mean of 0 m s$^{-1}$ and a standard deviation of $1.0 \times 10^{-2}$ m s$^{-1}$. The RMS errors between the initial guess and true bottom terrain can be found in Table 4.

![Figure 4](image-url)

**Figure 4.** The true bottom terrain with (a) a ridge shape, (b) a valley shape, (c) a basin shape and (d) a plateau shape and the different initial guesses.

**Table 4.** The RMS errors between the initial guess and true bottom terrain (m)

| Definition for initial guess | True bottom terrain shape |
|-----------------------------|---------------------------|
| Ridge                       | Valley                    | Basin                     | Plateau                   |
| Definition 1                | $1.409 \times 10^{-1}$    | $1.660 \times 10^{-1}$   | $1.779 \times 10^{-1}$   | $2.030 \times 10^{-1}$   |
| Definition 2                | $9.043 \times 10^{-2}$    | $1.237 \times 10^{-1}$   | $1.420 \times 10^{-1}$   | $1.631 \times 10^{-1}$   |
In general all of the codes we test can retrieve the bottom terrain on the basis of the initial guess with smoothing and scaling treatments (Table 5). The SCG method is not sensitive to the initial guess. It always performs the best with high efficiency and stability and has much lower computational cost when the numerical tests consider the line search algorithm. It is competitive with the CG_DESCENT method, which also has perfect performance with satisfied bottom terrain corrections. The performance of the LBFGS codes is not as good as that of the SCG and CG_DESCENT methods.

Table 5. Performance of the SCG, CG_DESCENT, LBFGS ($m=3/5/7$), SCG1, CG_DESCENT1 and LBFGS1 ($m=3/5/7$) methods for bottom terrain inversion with different initial guesses.

| Exp | Initial guess | Terrain shape | Iter  | Nfg | CPU (s) | RMS Error (m) |
|-----|---------------|---------------|-------|-----|---------|----------------|
|     |               |               | 83/98/3 | 168/198/6 | 64.91/76.97/22 | 5.51E-3/5.484E- |
|     |               | Ridge         | 00/300/ | 02/602/60 | 7.24/229.17/22 | 3/1.1009E-2/5.053E- |
| (5) |               |               | 300/39/ | 2/177/222 | 8.19/27.94/35.2 | 2/5.353E-2/5.262E- |
|     |               |               | 50/300/ | /602/602/ | 0/133.28/132.5 | 3/5.994E-3/3.833E- |
|     |               |               | 300/159 | 325 | 9/71.36 | 2/1.657E-2/6.416E-3 |
|     |               | Valley        | 79/136/ | 160/274/6 | 65.09/112.69/2 | 5.701E-3/5.911E- |
|     |               |               | 300/300 | 02/602/60 | 46.35/253.72/2 | 3/1.053E-2/6.778E- |
|     |               |               | /300/42/ | 2/192/175 | 51.90/31.81/28 | 2/7.284E-2/6.202E- |
|     |               |               | 40/81/1 | /200/381/ | 38/43.53/69.49/ | 3/6.605E-3/8.649E- |
|     |               |               | 14/157 | 340 | 79.30 | 2/6.255E-2/6.354E-3 |

Definition 1

| Exp | Initial guess | Terrain shape | Iter  | Nfg | CPU (s) | RMS Error (m) |
|-----|---------------|---------------|-------|-----|---------|----------------|
|     |               | Basin         | 82/125/ | 166/252/6 | 65.58/98.74/23 | 5.753E-3/5.955E- |
|     |               |               | 300/300 | 02/602/60 | 3.63/227.12/25 | 3/1.404E-2/8.872E- |
|     |               |               | /300/44/ | 2/197/294 | 2.15/31.67/43.1 | 2/1.017E-1/5.837E- |
|     |               |               | 53/13/3 | /99/107/5 | 4/13.83/19.89/9 | 3/6.011E-3/1.436E- |
|     |               |               | 2/151 | 47 | 5.19 | 1/1.135E-1/3.055E-2 |
|     |               | Plateau       | 91/107/ | 184/216/6 | 77.31/91.06/26 | 5.689E-3/6.684E- |
|     |               |               | 300/300 | 02/602/60 | 0.24/241.57/23 | 3/1.663E-2/1.061E- |
|     |               |               | /300/41/ | 2/175/294 | 3.27/31.28/49.5 | 1/1.154E-1/5.176E- |
|     |               |               | 61/14/7 | /62/336/5 | 6/12.55/50.86/1 | 3/5.527E-3/1.693E- |
|     |               |               | 3/192 | 57 | 0.36/66 | 1/1.250E-1/5.071E-3 |

Definition 2

| Exp | Initial guess | Terrain shape | Iter  | Nfg | CPU (s) | RMS Error (m) |
|-----|---------------|---------------|-------|-----|---------|----------------|
|     |               | Ridge         | 85/87/3 | 172/176/6 | 64.51/67.07/22 | 5.471E-3/5.282E- |
|     |               |               | 00/300/ | 02/602/60 | 7.60/229.33/22 | 3/8.233E-3/3.786E- |
|     |               |               | 300/36/ | 2/166/206 | 9.52/26.06/31 | 2/4.045E-2/5.229E- |
|     |               |               | 41/11/4 | /259/602/ | 2/56.66/134.57/ | 3/6.075E-3/4.912E- |
|     |               |               | 300/158 | 339 | 72.97 | 2/1.300E-2/6.009E-3 |
|     |               | Valley        | 85/120/ | 172/246/6 | 56.24/78.92/19 | 5.589E-3/3.086E- |
|     |               |               | 300/300 | 02/602/60 | 2.43/201.10/19 | 3/9.696E-3/5.687E- |
|     |               |               | /300/30/ | 2/146/251 | 7.56/20.67/32.9 | 2/6.014E-2/5.285E- |
|     |               |               | 54/43/3 | /120/828/ | 6/20.38/135.33/ | 3/6.056E-3/3.282E- |
|     |               |               | 0/178 | 389 | 70.41 | 2/1.864E-2/5.634E-3 |

| Exp | Initial guess | Terrain shape | Iter  | Nfg | CPU (s) | RMS Error (m) |
|-----|---------------|---------------|-------|-----|---------|----------------|
|     |               | Basin         | 78/91/3 | 158/184/6 | 51.22/59.60/20 | 5.988E-3/3.691E- |
|     |               |               | 00/300/ | 02/602/60 | 0.15/194.44/19 | 3/1.248E-2/2.245E- |
|     |               |               | 300/33/ | 2/155/157 | 4.04/20.21/21 | 2/8.170E-2/6.394E- |
|     |               |               | 36/13/1 | /95/634/1 | 1/11.02/77.04/2 | 3/6.561E-3/1.203E- |
|     |               |               | 21/39 | 59 | 1.51 | 1/7.177E-2/1.010E-1 |
|     |               | Plateau       | 118/155 | 238/312/6 | 87.60/115.00/2 | 5.783E-3/5.671E- |
|     |               |               | /300/30 | 02/602/60 | 43.07/229.12/2 | 3/1.416E-2/8.586E- |
|     |               |               | 0/300/4 | 2/178/176 | 18.21/47.67/27 | 2/9.101E-2/6.270E- |
|     |               |               | 1/36/14/ | /53/696/6 | 20.9.55/101.15/ | 3/6.573E-3/1.408E- |
|     |               |               | 128/118 | 30 | 91.31 | 1/8.127E-2/8.030E-2 |
However, as $m$ increases, the performance of the LBFGS method improves. The shapes of the corrected bottom terrain indicate also indicate the superior performance of the SCG method over the CG_DESCENT method and LBFGS codes (Figure 5).

![Figure 5. The inversed bottom terrain with different initial guesses. (a–h) Exp (5–12).](image)

The superiority of the performance of the SCG method over that of the CG_DESCENT method and LBFGS codes is clear from the variations of the value of the cost function and norm of the gradient with the number of iterations (Figure 6 and 7). We find that the value of the cost function and the norm of the gradient are reduced by at least 1 and 3 orders of magnitude, respectively, after a few iterations and function and gradient evaluations when using the SCG method. It has faster convergence rates than the CG_DESCENT method especially when the tests consider the optimal step length. The LBFGS codes reduce the value of the cost function and norm of the gradient with smaller orders of magnitude over the SCG and CG_DESCENT methods and require many more iterations and function and gradient evaluations.

7. Discussion

Different types of sparse and noisy observations will lead to quite different optimization results. Using fewer observations can considerably slow the convergence of the minimization procedure.
When the sparse observations are contaminated by random Gaussian noise, perfect bottom terrain corrections can be still obtained and the SCG method for model corrections is effective and competitive compared to the CG_DESCENT method and LBFGS codes with \(m = 5\) and 7 updates, likely as a result of a self-regularization property of the CG methods. However, its performance becomes worse when the random noise level is too high. Special attention should be paid to the random noise level when using this method.

**Figure 6.** The trend of the value of the cost function with the number of iterations. (a–h) Exp (5–12).

In the experiments accounting for the influence of different types of initial guess, the SCG method performs best, with small RMS errors, fast convergence and low computational cost, followed by the CG_DESCENT method and LBFGS codes with \(m = 3, 5\) and 7 updates. When the profile of the initial guess is far from the true terrain, the performance of the LBFGS codes becomes unacceptable. However, we find that as \(m\) increases, the performance of this algorithm improves, which agrees with the findings of Zou et al. [10]. Alekseev et al. [6] also investigated the sensitivity of descent methods to the choice of initial guess and found that the method of Hager and Zhang [11] had the top performance. Note that formulating a proper initial guess is important for conducting the assimilation procedures.
The SCG method benefits from the self-regularization property of CG methods when used for ill-posed posed problems. However, this is not enough to explain how the regularization was applied in solving the problem under consideration. Moreover, it is well known that a simple reduction of the number of eigenvectors is an ineffective approach to regularization.

One effective regularization approach is to introduce a regularization term to penalize the cost function to solve the ill-posed inverse problem under consideration. The regularization of this problem refers to solving the related regularized problem, which is a well-posed problem providing a physically meaningful solution to the given ill-posed problem [12]. The regularization function can greatly improve the precision of bottom terrain identification [13]. It has been proved that the performance of the SCG method is not limited by the choice of the initial guess if some of the known information of the true bottom terrain is considered by adding a regularization term to the cost function. The positive effect of the regularization function is also proved by the diffusion coefficient retrieval problem for a one-dimensional variable coefficient convection–diffusion equation [14].

The SCG method under consideration is highly efficient, and its convergence performance is better than that of the CG_DESCENT and LBFGS methods as a result of the combination of the properties of the classical CG method, the second-order information of the BFGS and the optimal step length. In our test experiments, the SCG method always has linear convergence, but it has additional

**Figure 7.** The trend of the norm of the gradient with the number of iterations. (a–h) Exp (5–12).
computational costs for the determination of the optimal step length, which involves solving the tangent linear model. The following two points should be noted. First, the integration of the tangent linear model will sometimes reduce the convergence rates; however, this strategy will ensure an optimal step length, which further ensures not only the self-correcting property of SCG approximation but also the descent direction. Second, the SCG method sometimes converges very fast with few iterations, but the condition number of the SCG approximation becomes large near the final stage of the correction procedure, which implies slow convergence for the SCG method in the final stage.

The LBFGS codes do not always converge linearly, especially when setting the number of updates to three. Its rates of convergence appear unstable in different test experiments, but the tests with LBFGS codes indicate that the optimal step length provides stable linear convergence. We must admit that the computational costs of the LBFGS are acceptable due to the efficient computation of the gradient and inverse Hessian approximation of the cost function. The LBFGS codes with various updates have been successfully used in many studies [4, 9, 15-17], but we suggest that it is particularly important to set the appropriate initial step size for the line search and the initial inverse Hessian of the cost function for the assimilation procedure.

Our assimilation system is competitive because of its application of a variational adjoint method that formulates the cost function to be minimized and an adjoint model to calculate the gradient of the cost function. Unlike the ensemble Kalman filter methods [18-20], our method does not need to assume any model error or bias and does not require an assumed number of members in the ensemble subspace before the start-up of the assimilation. However, the consideration of the known information about the desired bottom terrain eliminates this advantage.

The SCG method under consideration within the framework of a variational adjoint assimilation system proves to be very effective for solving optimization problems. We believe that the detailed description of our variational adjoint assimilation system will facilitate the application of variational adjoint technique and the SCG method in other inverse problems.

8. Conclusion

We introduce the SCG method within the framework of a variational adjoint assimilation system to correct the bottom terrain for a one-dimensional shallow-water equations model. This method utilizes Andrei’s formula (only latest 2 iterations information is required) approximated to the BFGS update formula and a “restart” strategy. It performs much better for well-posed test problems defined with the help of the regularization term in the cost function. It is robust when using observations with decreased number of observational fields or sparse and noisy observations. It is also proven to be much more robust than Hager and Zhang’s CG_DESCENT and Liu’s LBFGS methods, as this method always provides very small RMS errors and smooth profiles for the corrected bottom terrain at final time of correction procedure, even when the RMS errors for the initial guess and the profiles of the true bottom terrain differ greatly.

The SCG method mainly benefits from (i) the iterative regularization strategy, (ii) the inverse Hessian approximation involving the second-order information from BFGS, and (iii) the optimal step length. This study suggests the combined utilization of the SCG method and line search strategy to ensure high efficiency when solving the large-scale unconstrained problems. Moreover, special attention should be paid to (i) the density of the observations in the space dimension, (ii) the level of the random noise imposed on the observations, and (iii) the extra computational cost incurred by the determination of the optimal step length by solving the tangent linear model. Thus, it is reasonable to consider model order reduction techniques for better application of the SCG method.

9. Acknowledgments

This work is supported by the National Natural Science Foundation of China (41375115) and a project funded by Norway, Improving Weather Information Management in East Africa for Effective Service Provision through the Application of Suitable ICTs (UGA-13/0018).
References

[1] A.K. Griffith and N.K. Nichols, Adjoint methods in data assimilation for estimating model error, Flow Turbul. Combust. 65 (2000) 469-488.
[2] J. Zhong, J.-F. Fei, S.-X. Huang, and H.-D. Du, Study of weak constraint 4dvar with model error forcing control variable, Acta Phys. Sin.-Ch. Ed. 61 (2012) (149203)1-8.
[3] N. Andrei, Scaled conjugate gradient algorithms for unconstrained optimization, Comput. Optim. Appl. 38 (2007) 401-416.
[4] D. Liu and J. Nocedal, On the limited memory BFGS method for large scale optimization, Math. Program. 45 (1989) 503-528.
[5] W.W. Hager and H. Zhang, Algorithm 851: CG_DESCENT, a conjugate gradient method with guaranteed descent, ACM T. Math. Software (TOMS) 32 (2006) 113-137.
[6] A.K. Alekseev, I.M. Navon, and J.L. Steward, Comparison of advanced large-scale minimization algorithms for the solution of inverse ill-posed problems, Optim. Method Softw. 24 (2009) 63-87.
[7] Y.P. Wang, S.L. Tao, and Q. Chen, Retrieving the variable coefficient for a nonlinear convection–diffusion problem with spectral conjugate gradient method, Inverse Probl. Sci. En. 23 (2015) 1342-1365.
[8] J.E. Dennis Jr and R.B. Schnabel, Numerical methods for unconstrained optimization and nonlinear equations, Philadelphia: SIAM, 1996.
[9] X. Zou, I.M. Navon, and F.X. Le Dimet, Incomplete observations and control of gravity waves in variational data assimilation, Tellus A 44 (1992) 273-296.
[10] X. Zou, I.M. Navon, M. Berger, K.H. Phua, T. Schlick, and F.X. Le Dimet, Numerical experience with limited-memory quasi-Newton and truncated Newton methods, SIAM J. Optimiz. 3 (1993) 582-608.
[11] W.W. Hager and H. Zhang, A new conjugate gradient method with guaranteed descent and an efficient line search, SIAM J. Optimiz. 16 (2005) 170-192.
[12] I.M. Navon, Practical and theoretical aspects of adjoint parameter estimation and identifiability in meteorology and oceanography, Dynam. Atmos. Oceans 27 (1997) 55-79.
[13] S.L. Tao, Y.H. Li, and I. Mugume, Model terrain correction using variational adjoint method with Tikhonov-total variation regularization, J. Phys. Conf. Ser. 1176 (2019) 022034.
[14] Y.P. Wang and S.L. Tao, Application of regularization technique to variational adjoint method: a case for nonlinear convection–diffusion problem, Appl. Math. Comput. 218 (2011) 4475-4482.
[15] J. Nocedal, Updating quasi-Newton matrices with limited storage, Math. Comput. 35 (1980) 773-782.
[16] Z. Wang, K. Droegemeier, and L. White, The adjoint Newton algorithm for large-scale unconstrained optimization in meteorology applications, Comput. Optim. Appl. 10 (1998) 283-320.
[17] I.M. Navon, D.N. Daescu, and Z. Liu. The impact of background error on incomplete observations for 4D-Var data assimilation with the FSU GSM. in Computational Science-ICCS 2005. Springer Berlin Heidelberg, 2005.
[18] G. Evensen, The ensemble Kalman filter: theoretical formulation and practical implementation, Ocean Dynam. 53 (2003) 343-367.
[19] P.L. Houtekamer and H.L. Mitchell, Data assimilation using an ensemble Kalman filter technique, Mon. Weather Rev. 126 (1998) 796-811.
[20] D. Zupanski and M. Zupanski, Model error estimation employing an ensemble data assimilation approach, Mon. Weather Rev. 134 (2006) 1337-1354.