A Rapid Optimization Algorithm for GPS Data Assimilation

KUANG Zheng*[1,2] (匡 正), WANG Bin[2] (王 斌), and YANG Hualin[1,2] (杨华林)

[1] *Harbin Institute of Technology, Harbin* 150001

[2] *LASG, Institute of Atmospheric Physics, Chinese Academy of Sciences, Beijing* 100029

(Received January 20, 2002; revised January 24, 2003)

ABSTRACT

Global Positioning System (GPS) meteorology data variational assimilation can be reduced to the problem of a large-scale unconstrained optimization. Because the dimension of this problem is too large, most optimal algorithms cannot be performed. In order to make GPS/MET data assimilation able to satisfy the demand of numerical weather prediction, finding an algorithm with a great convergence rate of iteration will be the most important thing. A new method is presented that dynamically combines the limited memory BFGS (L-BFGS) method with the Hessian-free Newton(HFN) method, and it has a good rate of convergence in iteration. The numerical tests indicate that the computational efficiency of the method is better than the L-BFGS and HFN methods.

Key words: GPS data assimilation, L-BFGS method, HFN method, large-scale optimization

1. Introduction

With the improvement of numerical weather prediction models, their dynamic frame will not have more change than now. To improve numerical weather prediction further, we may do so by way of data variational assimilation. Now, with the aid of some new surveying means such as satellite and radar, we are almost able to probe the atmosphere and ocean environment immediately and successively. This greatly complements ordinary observation systems, and improves the quality of the numerical inital fields. The variational data assimilation analysis,therefore, has been paid more and more attention to extract useful meteorology information from remote sensing data.

Since the first Low Earth Orbit satellite equipped with a GPS receiver was launched by the USA in 1995, the research and application of the GPS occultation technique have greatly progressed (Zou et al., 2000). Because the refraction angle of a GPS ray originates from the refraction effects of the atmosphere, we can obtain information on atmospheric state variables from GPS/MET (Meteorology) Data by the assimilation method. The GPS occultation technique presents good vertical resolution, while the horizontal resolution only depends on the number of satellites. So GPS/MET data assimilation is especially suitable to the fields of mountain areas or the ocean.

GPS/MET data assimilation can be reduced to a minimum problem of a cost functional. In consideration of the applicable situation, the demension of the solution of the problem is at least 10^6. Because of the huge computation cost of this optimal problem, most of optimal algorithms are not suitable, and even a two-dimensional array would not be able to appear in the algorithm. Therefore finding an algorithm with little storage requirements and CPU time becomes the most important thing.

In this paper, we present a new optimal algorithm which is constructed by combining the L-BFGS method with the HFN method. The new algorithm can be performed perfectly on an SGI Origin 2000 for GPS/MET data assimilation. Numerical tests in which the dimensions reach 10^6 indicate that the computational efficiency of the given algorithm is better than that of L-BFGS algorithm.

2. Cost functional

The objective of GPS/MET data variational assimilation is to find a model solution which will best fit a series of observation data distributed over some space and time intervals. One possible measure of the fit between model and observation, the cost functional J, consists of a weighted least square fit of the model forecast to the observations based on optimal unbiased estimate theory and Kalman filtering (Zou et al.,

*E-mail: kuangzh@hope.hit.edu.cn

1999). The cost functional J can be written as

$$J(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x})^{\mathrm{T}}(\boldsymbol{O} + \boldsymbol{F})^{-1}(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_{\mathrm{b}})\boldsymbol{B}^{-1}(\boldsymbol{x} - \boldsymbol{x}_{\mathrm{b}}), \quad (1)$$

where $\boldsymbol{x}$ is the vector of atmospheric state variables; $\boldsymbol{x}_{\mathrm{b}}$ is the atmosphere background estimate; $\boldsymbol{B}$ is the covariance matrix of background error; $\boldsymbol{y}$ is the observations of GPS/MET; $\boldsymbol{O}$ is the covariance matrix of observation error and $\boldsymbol{F}$ is the covariance matrix of the GPS observation operator error; and $\boldsymbol{H}$ is the GPS observation operator which is determined by

(a) the GPS ray trajectory equation in the calculation technique;

(b) the algebraic relation of ray refractivity and the atmosphere state variables—the temperature, pressure and water vapor of the atmosphere; and

(c) the geometric relation of the GPS ray and refraction angles (Li et al., 2001)

To find the minimum of the cost functional (1), we need the gradient information of the cost functional with regard to control variables. Of the methods for generating the gradient, the adjoint method is a good one. Especially in the high dimensional cases, it can greatly reduce computational cost. The adjoint method completes a backward integration of the adjoint operator to obtain the gradient. The complexity of its computation of the gradient is the same as a forward integration of the model. Before introducing our method, we review two related methods, the limited memory BFGS (L-BFGS) and the Hessian-free Newton (HFN) method.

3. L-BFGS method

The L-BFGS method originates from the standard BFGS method (Nocedal, 1980; Liu and Nocedal, 1989). For the minimization of a nonlinear function

$$\min f(\boldsymbol{x}), \quad f: R^n \longrightarrow R,$$

the iteration of L-BFGS method is

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$$

where $\boldsymbol{d}_k$ is the iteration direction, and α_k is the iteration steplength which satisfies the Wolfe condition (Gilbert and Lemarechal, 1989; Gill et al., 1981):

$$f(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k) \le f(\boldsymbol{x}_k) + c_1 \alpha_k \boldsymbol{g}(\boldsymbol{x}_k)^{\mathrm{T}} \boldsymbol{d}_k$$
$$\left| \boldsymbol{g}(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k)^{\mathrm{T}} \boldsymbol{d}_k \right| \le c_2 \left| \boldsymbol{g}(\boldsymbol{x}_k)^{\mathrm{T}} \boldsymbol{d}_k \right| \quad (2)$$

$0 < c_1 < c_2 < 1$, $\boldsymbol{g}(\boldsymbol{x}_k) = \nabla f(\boldsymbol{x}_k)$. $\boldsymbol{d}_k$ can be obtained by computing

$$\boldsymbol{d}_k = -\boldsymbol{H}_k \boldsymbol{g}(\boldsymbol{x}_k),$$

where

$$\boldsymbol{H}_{k+1} = \boldsymbol{v}_k^{\mathrm{T}} \boldsymbol{H}_k \boldsymbol{v}_k + \rho_k \boldsymbol{s}_k \boldsymbol{s}_k^{\mathrm{T}}, \quad (3)$$
$$\rho_k = \frac{1}{\boldsymbol{y}_k^{\mathrm{T}} \boldsymbol{s}_k}, \qquad \boldsymbol{v}_k = \boldsymbol{I} - \rho_k \boldsymbol{y}_k \boldsymbol{s}_k^{\mathrm{T}},$$
$$\boldsymbol{s}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k, \qquad \boldsymbol{y}_k = \boldsymbol{g}(\boldsymbol{x}_{k+1}) - \boldsymbol{g}(\boldsymbol{x}_k). \quad (4)$$

Now the $n \times n$ BFGS matrices $\boldsymbol{H}_k$ are dense matrices. When n is very large it will not be possible to retain and operate on the matrices. In the L-BFGS method, only m vector pairs $\{\boldsymbol{s}_k, \boldsymbol{y}_k\}$ stored, and the updating matrices are generated by formula (3).

$\boldsymbol{H}_0$ is usually chosen as

$$\boldsymbol{H}_k^0 = \gamma_{k-1} \boldsymbol{I}, \qquad \gamma_k = \frac{\boldsymbol{s}_{k-1}^{\mathrm{T}} \boldsymbol{y}_{k-1}}{\boldsymbol{y}_{k-1}^{\mathrm{T}} \boldsymbol{y}_{k-1}},$$

where $\boldsymbol{I}$ is identity matrix. We call these matrices $\boldsymbol{H}_k$ limited memory matrices. When a new iteration step starts, the newest vector pair $\{\boldsymbol{s}_k, \boldsymbol{y}_k\}$ is used to replace the old pair. That is, the L-BFGS method always uses the newest m vector pairs $\{\boldsymbol{s}_k, \boldsymbol{y}_k\}$ to define the limited memory matrices $\boldsymbol{H}_k$. Although the L-BFGS method settles the storage problem in large-scale computation and performs inexpensive iterations, the quality of the curvature information it gathers can be poor, and as a result it can be very inefficient on ill-conditioned problems. Unfortunately, ill-conditioned situations will almost be sure to emerge in a large-scale problem. This is a weakness of the L-BFGS method.

4. HFN method

For Newton-type methods, we need to solve the equation

$$\nabla^2 f(\boldsymbol{x}_k)\, \boldsymbol{d}_k = -\boldsymbol{g}(\boldsymbol{x}_k), \quad (5)$$

so as to obtain iteration direction. An inner iteration may be performed to solve equation (5). Since we only need an approximate solution of equation (5), the inner iteration may be terminated as soon as the iterated solution satisfies the desired accuracy. This idea forms the Hessian-free Newton method (Byrd et al., 1996, Nocedal and Wright, 1999, Evans, 1967). In general, as the residual error

$$\gamma_k = \nabla^2 f(\boldsymbol{x}_k)\, \boldsymbol{d}_k + \boldsymbol{g}(\boldsymbol{x}_k) \quad (6)$$

is sufficiently small or a direction of negative curvature is detected, the inner iteration is terminated. The negative curvature terminating measure is to guarantee that the $\boldsymbol{d}_k$ is in a descending direction.

In the HFN method, it is assumed that the elements of the Hessian matrices $\nabla^2 f$ are not available.

One must therefore compute the matrix-vector products $\nabla^2 f(\boldsymbol{x}_k)\, \boldsymbol{d}$. They may be approximated by finite-differences

$$\nabla^2 f(\boldsymbol{x}_k)\, \boldsymbol{d} = \frac{\boldsymbol{g}(\boldsymbol{x}_k + \varepsilon \boldsymbol{d}) - \boldsymbol{g}(\boldsymbol{x}_k)}{\varepsilon}, \tag{7}$$

where

$$\varepsilon = \frac{2\sqrt{\Delta}\,(1 + \|\boldsymbol{x}_k\|)}{\|\boldsymbol{d}\|}, \tag{8}$$

and Δ denotes unit roundoff.

The HFN method normally requires much fewer iterations to approach the solution, but the effort invested in one iteration can be very high and the curvature information gathered in the process is lost after the iteration is completed. Therefore, it is not suitable to apply the HFN method in the whole iteration process.

5. The new method

In the HFN method, the large computation is spent on generating gradient and function information. However the information is not involved in the subsequent iteration, hence it is not economical to compute HFN steps at every iteration. If we use the function and gradient information generated by the inner iterations of the HFN method to improve the quality of the L-BFGS iterations, then it can be advantageous to bypass the weakness of only using gradient information generated in the L-BFGS method. We note that the strengths and weaknesses of the HFN and L-BFGS methods are complementary. In the following, we present a new method which combines the best features of both methods in a dynamic manner.

In the new method, l steps of the L-BFGS method are alternated with t steps of the HFN method. We illustrate this as

$$l^*\,(L - \text{BFGS}) \xrightarrow{H(m)} t^*\,(\text{HFN}) \xrightarrow{H(m)} \text{repeat}.$$

During the cycle of L-BFGS iterations, a limited memory matrix $H(m)$ is updated, where m denotes the number of correction pairs stored. The matrix obtained at the end of this cycle is used to precondition the first of the t HFN iterations. During each of the remaining t-1 HFN iterations, the limited memory matrix $H(m)$ is updated using information generated by the inner preconditioned conjugate gradient (PCG) iteration, and is used to precondition the next HFN iteration. Once the t HFN steps have been executed, the most current matrix $H(m)$ is used as the initial limited memory matrix in the new cycle of L-BFGS steps. The process continues in this manner, alternating cycles of L-BFGS and HFN iterations, and transmitting curvature information from one cycle to the next.

Clearly, the L-BFGS and HFN methods are particular cases of the new method, since they are obtained by setting t=0 and l=0, respectively. In our implementation of the new method, the lengths of the cycles, l and t, are chosen dynamically as the optimization process takes place.

The following is a broad outline of the new algorithm.

First choose a starting point x which is usually taken as x_0, the memory parameter m, and an initial choice of the length l of the L-BFGS cycle; set method='L-BFGS'; the first l iterations are L-BFGS cycles. After l steps of L-BFGS, $q = \max(l, m)$ pairs $\{\boldsymbol{s}_k, \boldsymbol{y}_k\}$ are stored. Then the t HFN steps are executed. The preconditioned matrix which is constructed by (3) is used in the inner PCG iterations

$$\boldsymbol{d}_{l+1}^{i+1} = \boldsymbol{d}_{l+1}^{i} + \lambda_i \boldsymbol{v}_i,$$

where λ_i is the steplength and $\{\boldsymbol{v}_i\}$ is a conjugate vector sequence. The products $\nabla^2 f(\boldsymbol{x}_k)\, d$ in the inner iterations are computed by (7). At the same time,

$$\boldsymbol{s}_{l+1}^{i} = \varepsilon \boldsymbol{v}_i, \quad \boldsymbol{y}_{l+1}^{i} = \boldsymbol{g}(\boldsymbol{x}_{l+1} + \varepsilon \boldsymbol{v}_i) - \boldsymbol{g}(\boldsymbol{x}_{l+1})$$
$$i = 1, \dots, q \tag{9}$$

are stored. In the subsequent t-1 HFN iterations, the preconditioned matrices are generated by the stored information of the preceding PCG step. When all the t HFN steps are completed, L-BFGS cycles start again. In such a way of alternating, the process continues until the results satisfy the requirements.

A subroutine ADJUST is designed to set the values of l and t. We now list the situations in which ADJUST modifies the lengths of these cycles:

1) If the PCG iteration generates a direction of negative curvature, we judge that we are in a region where L-BFGS steps are to be preferred over HFN steps. We therefore reset t=1, l=l+1, and set method='L-BFGS'.

2) If λ <0.8 in a HFN iteration, the iterates do not appear to have reached the region where a Newton-type iteration is rapidly convergent. In this case we set t=max{2, t-1}, and define method='L-BFGS'.

3) If the algorithm has reached the region where Newton's method is rapidly convergent, it is advisable to take as many HFN steps as is economically possible. Therefore in this case we increase t by one.

4) If at least 2 successful Newton iterations are performed in the cycle, we use the variable force2 to ensure that at least two HFN iterations are computed in succession. This variable is introduced because the full benefit of limited memory preconditioning is obtained only if more than one HFN iterations are performed in succession.

6. Numerical experiments and conclusions

To test the effectiveness of the new method, we perform a simulated computation for the model of GPS/MET data assimilation with dimension 2.58048×10^6 . All tests are performed on an SGI Origine 2000. We set the initial values m=5 and l=6. We use the background field of the atmosphere as the iterative initial vector and try the L-BFGS method and the new method respectively. Under the same terminating rule, the L-BFGS method performs 40 iterations involving 43 evaluations of function and gradient, and costs CPU time of 1267 seconds. In contrast, the new method performs 17 iterations involving 40 evaluations of function and gradient, and costs CPU time of 987 seconds. The related results are shown in Table 1 and Table 2. The results indicate that the new method is significantly more efficient than the L-BFGS method.

Table 1. The results of L-BFGS method iteration

ITER	NFN	FUNC	GNORM	STEPLENGTH
1	3	1.349D+02	6.524D+02	2.252D-04
4	6	6.236D+01	2.142D+02	1.000D+00
8	10	2.923D+01	9.437D+01	1.000D+00
12	15	2.691D+01	2.060D+01	1.000D+00
16	19	2.539D+01	3.050D+01	1.000D+00
20	23	2.491D+01	2.193D+01	1.000D+00
24	27	2.417D+01	1.192D+01	1.000D+00
28	31	2.377D+01	1.386D+01	1.000D+00
32	35	2.241D+01	1.573D+01	1.000D+00
36	39	2.164D+01	1.233D+01	1.000D+00
40	43	2.118D+01	1.278D+01	1.000D+00

ITER: the iteration cycle number; NFN: the number of evaluations of gradient and function; FUNC: the values of the cost function; GNORM: the norm of gradient; STEPLENGTH: steplength of iteration

Table 2. The results of the new method iteration

ITER	NFN	FUNC	GNORM	STEPLENGTH
1	3	1.349D+02	6.524D+02	2.252D-04
4	6	6.236D+01	2.142D+02	1.000D+00
8	14	2.655D+01	7.047D+01	1.000D+00
12	22	2.416D+01	1.042D+01	1.000D+00
16	38	2.123D+01	1.500D+01	1.000D+00
17	40	2.113D+01	1.607D+01	3.102D-01

The note is the same as in Table 1.

We also perform the HFN iteration (l is constantly zero). The HFN method costs CPU time of 1392 seconds.

In the whole GPS data assimilation computation, the cost of a gradient evaluation is too large. If we can use gradient evaluation method which costs less in terms of computation, the new method will be more advantageous.

In conclusion, the numerical results suggest that the new method should be considered as a serious competitor to the L-BFGS and HFN methods.

Acknowledgments. This research was supported by the National Excellent Youth Fund (Grant No. 49825109), the CAS Key Innovation Direction Project (Grant No. KZCX2-208), and LASG Project.

REFERENCES

Byrd, R. H., J. Nocedal, and C. Zhu, 1996: Towards a discrete Newton method with memory for large scale optimization. *Nonlinear Optimizations*, G. Di Pillo and F. Giannessi, Eds., Plenum, 1–12.

Evans, K. J. D., 1967: The use of pre-conditioning in iterative methods for solving linear equations with symmetric positive definite matrices. *J. Inst. Math. Application*, **4**, 295–314.

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

Gill, P. E., W. Murray, and M. H. Wright, 1981: *Practical Optimization*, London, Academic Press, 420 pp.

Li S. Y., B. Wang, and Zhang X., 2001: The parallel computing of GPS ray-shooting model. *Advances in Atmospheric Sciences*, **18**, 1185–1191.

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

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

Nocedal J., and S. J. Wright, 1999: *Numerical Optimization*. Springer Series in Operations Research, New York, 388 pp.

Zou X. L., B. Wang, H. Liu, R. A. Anthes, T. Matsumura, and Y. J. Zhu, 2000: Use of GPS/MET refraction angles in 3D variational analysis. *Quart. J. Roy. Meteor. Soc.*, **126**, 3013–3040.

Zou X. L., F. Vandenberghe, B. Wang, M. E. Gorbunov, S. Sokolovskiy, J. C. Chang, J. G. Sela, and R. A. Anthes, 1999: A ray-tracing operator and its adjoint for the use of GPS/MET refraction angle measurements. *J. Geophy. Res.*, **104**(D18), 22301–22318.

GPS资料同化中一种快速优化算法

匡正 王斌 杨华林

P4 A

摘 要

GPS资料同化最终归结为一个大规模的无约束优化问题。由于问题的维数很高，使得寻找到一种快速且节约存储的优化算法成为GPS资料同化能否满足在业务数值天气预报上的要求的关键。提出了一种新的优化算法。该方法把L-BFGS和HFN两种方法动态的结合起来，同时利用有限差分技术和截去技术，使算法在存储和计算量适合当前设备的前提下，较大的提高了算法的收敛速度。数值试验表明，该方法的计算效率与L-BFGS方法相比有十分明显的改善。

关键词: GPS资料同化, L-BFGS 方法, HFN 方法, 大型优化