

D
27
N84 22076

APPLICATION OF AUGMENTED-LAGRANGIAN METHODS IN METEOROLOGY: COMPARISON OF
DIFFERENT CONJUGATE-GRADIENT CODES FOR LARGE-SCALE MINIMIZATION

I. M. Navon

ABSTRACT

A Lagrange multiplier method using techniques developed by Bertsekas (1982) was applied to solving the problem of enforcing simultaneous conservation of the nonlinear integral invariants of the shallow water equations on a limited area domain.

This application of nonlinear constrained optimization is of the large dimensional type and the conjugate gradient method was found to be the only computationally viable method for the unconstrained minimization. In this study several conjugate-gradient codes were tested and compared for increasing accuracy requirements. Robustness and computational efficiency were our principal criteria.

1. THE AUGMENTED LAGRANGIAN ALGORITHM

We consider the problem

$$\text{minimize } f(\underline{x}) \quad (1)$$

$$\text{subject to } \underline{x} \in R^m \text{ and } h(\underline{x}) = 0$$

$$h: R^m \rightarrow R^m$$

The augmented Lagrangian function is

$$L(\underline{x}, \underline{\lambda}, c) = f(\underline{x}) + \underline{\lambda}^T h(\underline{x}) + (1/2c) |h(\underline{x})|^2 \quad (2)$$

where c is a penalty parameter and λ a multiplier vector.

The augmented Lagrangian algorithm is an iteration involving a series of unconstrained minimization problems (see Gill, Murray and Wright, 1982; Bertsekas, 1982) of an n -dimensional differentiable function, Lagrange multiplier estimates and penalty term updates - where the primary purpose of the penalty term is to make \underline{X}^* an unconstrained minimum.

2. THE MODEL ALGORITHM

We briefly describe an iteration of the Augmented Lagrangian method. The following are assumed to be available: an initial selection of the constraint functions; an initial-estimate of the Lagrange-multipliers $\underline{\lambda}_0$; a penalty parameter c , a positive integer K serving as upper bound on the number of unconstrained minimizations to be performed and an initial point \underline{x}_0 . Set $k \leftarrow 0$ and perform the following steps:

AL1. (Check termination criteria). If \underline{x}_k satisfies optimality conditions, the algorithm terminates with \underline{x}^k as the solution. If $k > K$, the algorithm, terminates with a failure.

AL2. (Minimize the augmented Lagrangian function). With \underline{x}_k as starting point execute a procedure (using a conjugate-gradient code) to solve the sub-problem:

$$\begin{aligned} \text{Minimize } L(\underline{x}, \lambda_k, c) & \quad (3) \\ \underline{x} \in R^m & \end{aligned}$$

Let \underline{x}_{k+1} denote the best approximation to the solution of (3).

AL3. (Update the multiplier estimate). If appropriate, modify specification of h . Compute λ_{k+1} , an updated estimate of the Lagrange multipliers.

AL4. (Increase the penalty parameter if necessary). Increase c if the constraint violations at \underline{x}_{k+1} have not decreased sufficiently from those at \underline{x}_k .

AL5. (Update the iteration count). Set $k \leftarrow k+1$ and go back to AL1.

3. COMPARISON OF CONJUGATE-GRADIENT (C-G) METHODS

C-G methods form a class of minimization algorithms that generate directions of minimum search without requiring the storage of a matrix. These methods are essential when the Hessian matrix is too large or too dense.

In our case we had 2000 variables, ($\underline{x} = (x_1, x_2, \dots, x_{2000})$).

a) The Fletcher-Reeves (1964) C-G Algorithm

We solve the problem

$$\begin{aligned} \text{Minimize } f(\underline{x}) & \quad \underline{x} = (x_1, \dots, x_m) & \quad (4) \\ \underline{x} \in R^m & \end{aligned}$$

$$\begin{aligned} \text{AL1. Compute } g_0 &= -\nabla f(X_0) \\ \text{Set } d_0 &= -g_0 \end{aligned} \quad (5)$$

then for $k = 1, 2, 3, \dots, m$

$$\text{Set } X_{k+1} = X_k + \alpha_k d_k \quad (6)$$

Where α_k is obtained via a line search procedure $f(X_k + \alpha_k d_k) = \min_{\alpha} f(X_k + \alpha d_k)$.

3. Compute

$$g_{k+1} = \nabla f(X_{k+1}) \quad (7)$$

(4) generate d_{k+1} by:

$$d_{k+1} = -g_{k+1} + \beta_k d_k \quad (8)$$

$$\text{Where } \beta_k = \frac{\nabla f(X_{k+1})^T \nabla f(X_{k+1})}{\nabla f(X_k)^T \nabla f(X_k)} = \frac{g_{k+1}^T g_{k+1} / g_k^T g_k}{g_{k+1}^T g_k} \quad (9)$$

b) The Polak-Ribiere (1969) Algorithm

The algorithm is totally identical in its steps to the Fletcher-Reeves algorithm except that β_k is calculated as

$$\beta_k = \frac{(g_{k+1} - g_k)^T g_{k+1}}{g_k^T g_k} = \frac{(\nabla f(X_{k+1}) - \nabla f(X_k))^T \nabla f(X_{k+1})}{\nabla f(X_k)^T \nabla f(X_k)} \quad (9^*)$$

For the Fletcher-Reeves and Polak-Ribiere algorithms we used the Numerical Algorithms Group (NAG) subroutine E04DBF with its associated service routines.

c) The Powell Conjugate Gradient Method with Restarts

This method based on the work of Beale (1972) and Powell (1977) allows us to restart after a C-G cycle with a computed direction d_t - rather than starting with the steepest gradient - g_t .

In order to ensure that the successive directions are conjugate, the following recurrence relation is used for a cycle of n directions for $k = 0, 1, \dots, n-1$

$$d_{k+1} = -g_{k+1} + \beta_k d_k + \gamma_k d_t \quad (10)$$

d_t is known as the restart direction and is the last direction of the previous C-G cycle along which a linear search was made. Powell (1977) uses the Beale restarts every n steps or whenever

$$|g_{k+1}^T g_k| \geq 0.2 \|g_{k+1}\|^2 \quad (11)$$

which checks that the direction d_k is "sufficiently downhill."

We also check that

$$-1.2 \|g_k\|^2 \leq d_k^T g_k \leq -.08 \|g_k\|^2 \quad (12)$$

If this requirement is not satisfied, a new cycle commences with d_{k-1} as the restart direction and with d_k re-computed from (8). For the Powell method we used the ZXCGR subroutine of the IMSL library with slight modifications.

d) The Shanno C-G Method Viewed as a Memoryless Quasi-Newton Method

Quasi-Newton Method

Different authors - Shanno (1978), Shanno and Phua (1980), Perry (1976), have found an analogy between the Quasi-Newton method of the form

$$x_{k+1} = x_k + d_k d_k \quad (13)$$

$$d_k = -D_k^{-1} \nabla f(x_k)$$

where D_k is a positive definite matrix approximating the Hessian, and between the C-G method if we compute D_k by updating the identity matrix with a limited number of Quasi-Newton corrections.

Although the direction of search, d_k , is equivalent to the product of a matrix and a vector, the matrix is never stored explicitly. Rather, only the vectors defining the updates are retained. Shanno (1978) used such a "memoryless" Quasi-Newton (Q-N) method combined with Beale restarts and Powell's restart criteria.

For the Shanno method we used the Shanno and Phua routine CONMIN supplied to us by Prof. Shanno which allows one either a BFGS (Broyden-Fletcher-Goldfarb-Shanno) Q-N variable metric algorithm or a Beale restarted C-G algorithm.

4. NUMERICAL RESULTS

A program (METHODS) comparing the 4 C-G algorithms for a meteorological

problem (see Navon and De-Villiers (1983)) with 2000 variables, for increasing accuracy requirements was developed (available now on YFIMN machine at NASA/GLAS) and the different methods were compared for computational efficiency, number of iterations and number of function calls.

Our accuracy criteria was

$$\|g(\underline{X})\|^2 \leq \epsilon_a \quad (14)$$

Accuracy $\epsilon_a = 10^{-3}$

Method	Iter No. of C-G cycles	IFUN No. of Function evaluations	CPU (Sec)
Fletcher-Reeves	2	25	0.11
Polak-Ribiere	2	25	0.10
Powell	2	25	0.09
Shanno	2	25	0.09
Function value at minimum = $.6047 \times 10^{-5}$			
Accuracy $\epsilon_a = 10^{-5}$			
Fletcher-Reeves	4	81	0.17
Polak-Ribiere	5	121	0.21
Powell	3	64	0.14
Shanno	4	81	0.17
Function value at minimum = $.4211 \times 10^{-5}$			
Accuracy $\epsilon_a = 10^{-7}$			
Fletcher-Reeves	11	523	0.44
Polak-Ribiere	Failed	-	-
Powell	8	379	0.34
Shanno	8	324	0.36
Function value at minimum = $.4187 \times 10^{-5}$			

Method	Iter No. of C-G cycles	IFUN No. of Function evaluations	CPU (Sec)
Accuracy $\epsilon_a = 10^{-9}$ (double precision)			
Fletcher-Reeves	Failed		
Polak-Ribiere	Failed		
Powell	16	986	0.65
Shanno	12	625	0.51
Function value at minimum = $.418549 \times 10^{-5}$			
Accuracy $\epsilon_a = 10^{-11}$ (double precision)			
Powell	19	1542	0.96
Shanno	12	676	0.58
Function value at minimum = $.48154561 \times 10^{-5}$			

5. CONCLUSIONS

For this large dimensional problem and for moderate accuracy (10^{-5}) all the 4 C-G algorithms performed well, with the Powell and Shanno C-G algorithms being slightly faster. For higher accuracy requirements both the Fletcher-Reeves and the Polak-Ribiere algorithms fail to converge for the upper limit of C-G cycles (25) and only the Powell and Shanno algorithms continue to perform with an increased advantage for the Shanno algorithm as the accuracy requirements increase to 10^{-9} or 10^{-11} .

These methods were tested on the problem of enforcing conservation of nonlinear integral invariants of the shallow water equations using both a finite difference model (Navon and de-Villiers, 1983) or a finite element model (Navon, 1983). We intend to test the same methods for constrained nonlinear normal mode initialization.

REFERENCES

- Bertsekas, D. K., 1982: Constrained optimization and Lagrange multiplier methods. Academic Press, New York, 412 pp.
- Gill, P. E., W. Murray, and M. H. Wright, 1981: Practical optimization. Academic Press, New York, 401 pp.

- Fletcher, R., C. M. Reeves, 1964: Function minimization by conjugate-gradients. Computer Journal, 7, 149-154.
- Polak, E., G. Ribiere, 1969: Note sur la convergence de Methodes de Directions Conjuguees. Rev. Franc. Informat. Rech. Operationnelle, 16, 35-43.
- Beale, E. M. L., 1972: A derivation of conjugate-gradients. In F. A. Lootsma, ed. Numerical Methods for Nonlinear Optimization. Academic Press, London, pp. 39-43.
- Powell, M. J. D., 1977: Restart procedures for the conjugate-gradient method. Math. Programming, 12, 241-254.
- Shanno, D. F., 1978: Conjugate-gradient methods with inexact searches. Math. Oper. Res., 3, 244-256.
- Shanno, D. F., and K. H. Phua, 1980: Remark on algorithm 500. ACM Trans. on Math. Software, 6, 618-622.
- Perry, A., 1976: A modified conjugate-gradient algorithm. Disc. Paper, 229, Northwestern Univ.
- Navon, I. M., and R. deVilliers, 1983: Combined penalty-multiplier optimization methods to enforce integral invariants conservation. Mon. Wea. Rev., 111, 1228-1243.
- Navon, I. M., 1983: A Numerov-Galerkin technique applied to a finite element shallow water equations model with enforced conservation of integral invariants and selective lumping. J. Comput. Phys., 52, 313-339.

PREDICTABILITY EXPERIMENTS USING A LOW ORDER EMPIRICALLY
CORRECTED DYNAMICAL MODEL

S. Schubert

1. INTRODUCTION

It is generally accepted that day to day weather variations possess a finite range of predictability estimated to be approximately two weeks (e.g. Lorenz, 1965). However, considerable observational evidence points to the existence of a number of low frequency flow regimes which are potentially predictable beyond this limit. These include blocking events and teleconnection patterns such as those described in Wallace and Gutzler (1981). The present study addresses the problem of the predictability of such modes by employing a highly simplified dynamical model projected onto the modes of interest. These modes are computed from an empirical orthogonal function (EOF) analysis of 10-day averaged anomalies (deviations from the mean seasonal cycle) of the 500 mb stream function for the winters of 1967-76. The first three EOF's are associated with an index cycle and some of the teleconnection patterns. The fourth and ninth are related to North Pacific and North Atlantic blocking, respectively. Details of the calculations and mode structures may be found in Schubert (1983) (hereafter S83).

2. THE EOF ANOMALY MODEL

The model is equivalent barotropic and includes crude representations of the effects of friction and orography. A long wave correction term is not included due to the filtering effect of the EOF expansion (see S83). The model is formulated as a prognostic equation for the anomalies where the mean flow acts as an inhomogeneous forcing. In its final form, the model is written as a system of prognostic equations for the EOF coefficients or principal components (PC's) (see S83 for details) as

$$\begin{aligned}
 \dot{\tilde{z}} &= D(\tilde{z})\tilde{z} + (1+\epsilon)G\tilde{z} + \gamma F\tilde{z} \\
 \text{tendency} & \quad \text{nonlinear} & \quad \text{beta} & \quad \text{orography} \\
 & \quad \text{interaction} & & \\
 + \tilde{Rz} & - \kappa \tilde{Az} + \tilde{z}^* \\
 \text{mean/anomaly} & \quad \text{friction} & \quad \text{inhomogenous} \\
 \text{interaction} & & \quad \text{forcing}
 \end{aligned} \tag{1}$$

Here $\tilde{z}(t)$ is the vector of principal components and ϵ , γ and κ are constants which are fit to the data to provide the best representation of the observed PC