Tests with the CONGRAD minimization algorithm within the ALADIN/HU 3DVAR system

January 25, 2007

Zoltán Juhász[1] and Gergely Bölöni[2]

Introduction

This is a summary about a comparison of the two existing minimization algorithms in ALADIN 3DVAR. Under minimization algorithm we understand the mathematical method for computing the minimum of the cost function in the variational assimilation. These two algorithms are coded in ARPEGE/ALADIN under the M1QN3 and the CONGRAD subroutines. In ALADIN so far the M1QN3 method has been used for both test and operational versions. The purpose of the presented comparison was on one hand to understand the algorithms more in depth and on the other hand to compare their efficiency in practice. The M1QN3 method belongs to the family of quasi-Newton methods while the CONGRAD method is a conjugate gradient method combined with the Lanczos algorithm. Both methods aim to minimize the

$$J(\boldsymbol{x})=(\boldsymbol{x}-\boldsymbol{x_b})^T B^{-1}(\boldsymbol{x}-\boldsymbol{x_b})+(\boldsymbol{y}-H[\boldsymbol{x}])^T R^{-1}(\boldsymbol{y}-H[\boldsymbol{x}])$$

variational cost function with respect to the $\boldsymbol{x}$ control vector through an iterative algorithm. In the formula of the costfunction $\boldsymbol{x_b}$ and $\boldsymbol{y}$ denotes respectively the background and the observations, B and R stand for their error covariance matrices and H is the observation operator. The iterative minimization process can be written as follows:

$$\boldsymbol{x}_{\boldsymbol{k+1}}=\boldsymbol{x}_{\boldsymbol{k}}+a_k\boldsymbol{d}_{\boldsymbol{k}} \qquad k=0,1,\ldots,n$$

where n is number of iterations, $\boldsymbol{d_k}$ is the vector pointing to a descent direction and a_k is the factor determining the length of the step to be taken in this direction. The two methods differ in generating of the $\boldsymbol{d_k}$ sequence. While M1QN3 uses the first derivative of the cost function and the limited storage required approximation of its second derivative, CONGRAD operates with the conjugate directions and the $\boldsymbol{d_1},\ldots,\boldsymbol{d_{k-1}}$ vectors are taken from the so called Lanczos method using favorable eigenvalue properties to find the best directions in the first few iterations. For the reason of simplicity this paper will not include more detailed comparison and analysis of the minimization algorithms but will concentrate on some tests done with ALADIN 3DVAR at HMS. For those interested more in depth the following papers are proposed for reading: *Bertsekas (1999), Golub and Van Loan (1989), Eijkhout (1995), Meurant and Strakos (2006)*

---

1 University of Eötvös Loránd
2 Hungariam Meteorological Service (HMS)

Tests and results

The tests consisted of running 3DVAR analyses with both methods described above within the ALADIN/HU system. All the tests were done on one single date using the same background and observations (SYNOP and TEMP). The model geometry used is the presently operational Central European domain with ~8 km resolution (360 x 320 points) and 49 vertical levels. All the results were obtained with 4 processor runs (1.3 GHz each) on the IBM p690 machine of HMS. The efficiency of the two methods can be measured by comparing their total CPU costs until reaching a certain J_{min} value. We choose the required J_{min} to be the value corresponding to our operational setting (M1QN3 with 70 iterations).

	No. of iterations	***Total CPU***	***Average CPU / iteration***	***Costfunction (J)***
M1QN3	70	1506 sec	21.51 sec	'0.2752845E+04'
CONGRAD	45	975 sec	21.66 sec	'0.2751173E+04'

Table 1. Comparison of the M1QN3 and CONGRAD methods

Looking at the summary of the comparisons (Table 1.) one can see that the CONGRAD method uses less CPU than M1QN3 in order to reach the same costfunction value. The difference is around 10 minutes within the 4 processor runs. In percentage CONGRAD uses 65% of the CPU used by M1QN3. Taking into account this percentage, in an operational environment where the minimization last for 10 minutes (using 24 CPUs) the expected gain with the CONGRAD method is about 3.5 minutes. Following Table 1. it turns out as well that one single iteration step is more efficient in case of the CONGRAD method as it reaches the required J_{min} within less iterations (45) than M1QN3 (70). One iteration step costs approximately the same in both methods in terms of CPU. Some of our statements above can be seen also on Fig1. Note that at the beginning of the minimization there are about 10 iterations with an almost constant costfunction value. With CONGRAD even a large oscillation in the costfunction value is present before the decrease starts. Unfortunately we did not find an explanation or any reference yet to this behavior in the literature. We mention also that these first 10 iterations are not counted in the minimization. For instance if one maximizes the number of iterations as 60 through the namelist of conf. 131 in the reality 70 iterations will be performed including the extra 10 before the decrease of the costfunction value starts.

Concluding remarks

The tests shown prove that the CONGRAD algorithm is more efficient than M1QN3 considering the speed of the minimization. The gain in the CPU with CONGRAD is more pronounced in case of low number of processors (10 min. gain on 4 procs.) but it is remarkable even in an operational environment (3.5 min. gain on 24 procs.). However one should remember that the tests were done on one single case and that the qualities of the analyses were not compared or validated from meteorological point of view. In order to go to real practical conclusions test cases or score comparisons are needed as well.

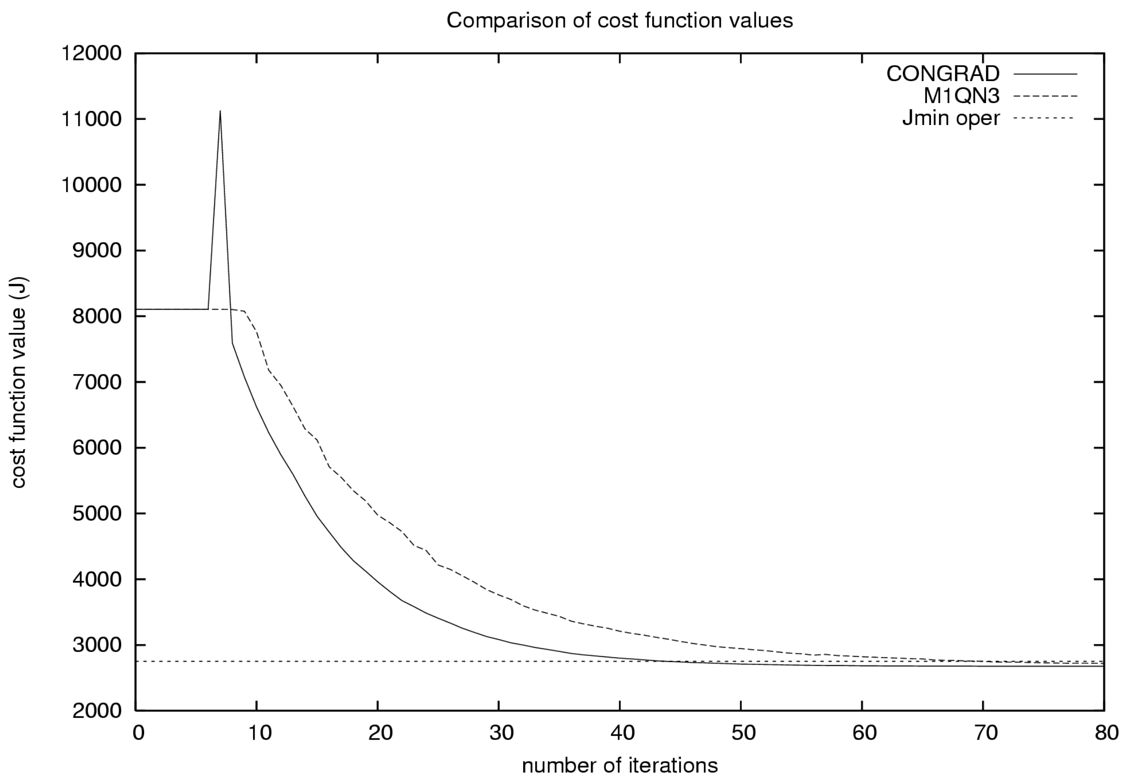

Fig 1. Evolution of the costfunction during the minimization with M1QN3 and CONGRAD. 'Jmin oper' stands for the cost function value provided by the present ALADIN/HU 3DVAR minimization settings (M1QN3 with 70 iterations).

References

Bertsekas D. P., 1999: Nonlinear Programming. *Athena Scientific, Belmont, Massachusetts*

Golub G. H. and C. F. Van Loan, 1989: Matrix Computations. *The John Hopkins University Press, Baltimore, Maryland*

Eijkhout V., 1995: LAPACK working note No. 51: Qualitative Properties of the Conjugate Gradient and Lanczos Methods in a Matrix Framework, *Univ. of Tennessee, Department of Computer Science, 33 pp*

Meurant G. and Z. Strakos, 2006: The Lanczos and conjugate gradient algorithms in finite precision arithmetic, *Acta Numerica vol. 15, pp 471-542, Cambridge University Press*