

# Nonlinear Conjugate Gradient Methods and Their Implementations by TAO on Dawning 2000-II<sup>+</sup>

Jian Wang<sup>\*</sup>, Xuebin Chi<sup>\*</sup>, Tongxiang Gu<sup>†</sup>

## Abstract

Nonlinear conjugate gradient (CG) method is a typical unconstrained optimization method. TAO has three implementations of CG method: CG\_FR, CG\_PR and CG\_PRP. In this paper, we describe their implementations in TAO and give a test result.

**Key words:** nonlinear conjugate gradient methods, global convergence, TAO, PETSc.

## 1. Introduction

Nonlinear conjugate gradient method is one of the most useful and the earliest techniques for solving large-scale nonlinear optimization problems. Many variants of this original scheme have been proposed, and some are widely used in practice. CG method only uses the first-order-derivative information of the objective function and need not update the Hessian matrix each iterative.

The algorithms in TAO place strong emphasis on the reuse of external tools where appropriate. TAO allows the reuse of toolkits that provide lower-level support (parallel sparse matrix data structures, preconditioners, linear solvers) provided in toolkits such as PETSc[13], which relies on MPI [6] for all interprocessor communication. In terms of efficiency and development time, the advantages are significant [4].

## 2. TAO (Toolkit for Advanced Optimization)

TAO focuses on the development of algorithms and software for the solution of large-scale optimization problems on high-performance architectures. Areas of interest include nonlinear least squares, unconstrained and bound-constrained optimization, and general nonlinear optimization. The aim is to use object-oriented techniques to produce high-quality optimization software for a range of computing environments.

TAO owes much to the PETSc and has benefited from its experience, tools, and software. In many ways, TAO is a natural outcome of the PETSc development. TAO has also benefited from the work of various researchers who have provided solvers, test problems, and interfaces. In the implementations, TAO makes no assumptions about the representation of these objects by passing pointers to data-structure-neutral objects for the execution of these numerical operations [4].

## 3. Nonlinear conjugate gradient methods

Nonlinear CG methods are widely used for unconstrained optimization, especially for large-scale problems. The key property of CG method is that sometimes approaches the solution very quickly and requires little storage. The definition of the conjugate is seen [11,pp.102].

$$Q(x) = \frac{1}{2} x^T A x + b^T x + c \quad (1)$$

where  $A$  is a positive symmetric definite matrix,  $x$ ,  $b$  and  $c$  are vectors in  $R^n$ . We restate some theorems about the conjugate gradient method without proof.

**Theorem 1**<sup>[12]</sup> For (1), CG method with an exact line search terminates at the solution at most  $n$  iterative steps, where  $n$  is the dimension of variables.

**Theorem 2**<sup>[11]</sup> If  $A$  in (1) has only  $r$  distinct eigenvalues, then CG method will terminate at the solution at most  $r$  iterative steps.

CG method can compute a new vector by only using the previous vector. The new is automatically conjugate to the previous vectors. It's obvious that CG method requires little storage and computation. So it is suited to solve large-scale problems. For the objective function is quadratic the CG-preliminary version is seen in [11,pp.108].The basic properties of CG method is

---

<sup>+</sup> This work was partially supported by State Hi-Tech Research and Development Program of China(863)(2001AA111043) and the Special Funds for Major State Basic Research Projects of China (G1999032805)

<sup>\*</sup> Supercomputing Center, Computer Network Information Center, Chinese Academy of Sciences, Beijing 100080, P.R. China, [jwang@jupiter.cnc.ac.cn](mailto:jwang@jupiter.cnc.ac.cn), [chi@jupiter.cnc.ac.cn](mailto:chi@jupiter.cnc.ac.cn).

<sup>†</sup>Laboratory of Computational Physics, Institute of Applied Physics and Computational Mathematics, Beijing, 100088, P.R. China, [txgu@iapcm.ac.cn](mailto:txgu@iapcm.ac.cn).

seen in [1,pp.189]. For general nonlinear functions, Fletcher and Reeves showed the algorithm for the nonlinear unconstrained optimization is as CG\_FR[11,pp.120]. In aforementioned algorithm, the search direction  $p_k$  is a descent direction and so (2) should be negative.

$$\nabla f_k^T p_k = -\|\nabla f_k\|^2 + \beta_k^{FR} \nabla f_k^T p_{k-1} \quad (2)$$

$$\beta_{k+1}^{PR} = \frac{\nabla f_{k+1}^T \nabla f_{k+1}}{\nabla f_k^T \nabla f_k} \quad (3)$$

The line search with the strong Wolfe conditions is applied:

$$f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k \nabla f_k^T p_k \quad (4a)$$

$$|\nabla f(x_k + \alpha_k p_k)^T p_k| \leq c_2 |\nabla f_k^T p_k| \quad (4b)$$

where  $0 < c_1 < c_2 < 1/2$ . It can be shown that (4b) implies that (2) is negative and the conclusion is that any line search procedure that yields a  $\alpha_k$  satisfying (4) will ensure that all directions  $p_k$  are descent direction for the function  $f$ , see [11].

In practice, there are many variants of the Fletcher-Reeves method that differs from each other mainly in the choice of the parameter  $\beta_k$ . TAO has another two CG methods as following.

$$\beta_{k+1}^{PR} = \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{\nabla f_k^T \nabla f_k}, \quad (5)$$

$$\beta_{k+1}^{PRP} = \max(\beta_{k+1}^{PR}, 0) \quad (6)$$

CG algorithms using (5) and (6) are called as Algorithm CG\_PR and CG\_PRP, respectively.

For general nonlinear functions, because of the approximation of the algorithm, some properties of CG method are violated. Thus it is impossible to obtain convergence in limited iterative steps. In practice, if we get the precision required when the iterative  $k$  is less than  $n$ ,  $x_k$  is the approximation solution. Otherwise, we restart the iterative, i.e. set  $\beta_k = 0$ . Because the nonlinear CG method is recommended only for large problems, i.e.  $n$  very large, in such problems, restart may never occur. The most popular strategy is whenever two consecutive gradients are far from orthogonal, as measure by the test

$$\frac{|\nabla f_k^T \nabla f_{k-1}|}{\|\nabla f_k\|^2} \geq \nu \quad (7)$$

where a typical value for the parameter  $\nu$  is in  $[0,1]$ . Another strategy of restart, often used, is if (2) is nonnegative. But it will significantly reduce the efficiency of nonlinear CG method if the restarts are used frequently, which makes it more like the steepest descent method.

The global convergence properties of the CG without regular restarts have been studied by many authors, including Dai and Yuan [5], [7], [8], Gilbert and Nocedal[9], Yuhong Dai [10] etc.

## 4. Implementations in TAO and computational results

### 4.1 Implementation of CG

To implement the nonlinear CG method, TAO chooses to compute the function value and gradient in parallelism. We give the implementation of CG\_FR as below.

```
{...
/*Get the initial solution xx, and then evaluate the function value f and gradient gg in
parallelism.*/
dx:=0; /* the descent direction */
gnorm2:=( gg, gg ); /* Euclidean norm */
gnorm2Prev=gnorm2; /* Previous gradient */
beta:=0; restarts:=0; /* the parameter restarts records the number of restart. */
while(1)
{TaoMonitor(...); /* Check if the convergence conditions are satisfied ,then record the
information about iterative and solution, and stop; otherwise continue. */
beta:=gnorm2/gnorm2Prev; dx:=-gg+beta*dx; gdx:=( dx, gg );
```

```

if(gdx>=0) /* If dx is not a descent direction, use gradient and restart.*/
{
    dx := - gg ; restarts:=restarts+1; beta := 0 ;
}
gnorm2Prev=gnorm2; step=1.0; /* the initial line search step */
TaoLineSearchApply(...); /* The line search routine searches the new xx along the
search direction dx and returns new function value, new gradient
gg and optimal step */
gnorm2:=( gg, gg );
}...}

```

The norm of the gradient is a standard measure used by the solvers to define convergence conditions. This quantity is always nonnegative and equals zero at the solution. The solver will pass this quantity, the current function value, the current iteration number, and a measure of infeasibility to TAO with the routine TaoMonitor(...). Nonlinear conjugate gradient algorithms require a line search. TAO provides several line searches and support for using them. The routine TaoLineSearchApply(...) passes the current solution, gradient, and objective value to the solver and returns the corresponding new ones. More details on line searches can be found in [4].

In the entire implementation of CG, no assumptions are made about the representations of data in the vectors and matrices. This approach eliminates some of the barriers in using independently developed software components by accepting data that is independent of representation and interfacing to numerical routines with appropriate data formats.

## 4.2 Computational Result

The elastic plastic torsion problem tested arises from the determination of the stress field on an infinitely long cylindrical bar, which is equivalent to the solution of the following problem [3]:

$$\min_v q(v)$$

where  $q: K \rightarrow R$ ,  $q(v) = \int_D \left\{ \frac{1}{2} \|\nabla v(x)\|^2 - c \cdot v(x) \right\} dx$  and  $c$  is the torsion angle per unit

length, the convex set  $K$  is defined by

$$K = \{v \in H_0^1(D) : |v(x)| \leq \text{dist}(x, \partial D), x \in D\}.$$

The  $\text{dist}(\cdot, \partial D)$  is the distance function to the boundary of  $D$ , and  $H_0^1(D)$  is the Hilbert space of all functions with compact support in  $D$  such that  $v$  and  $\|\nabla v\|^2$  belong to  $L^2(D)$ . In this experiment,  $c$  is selected to be 5.0.  $D = (l_1, u_1) \times (l_2, u_2)$  is a rectangle in  $R^2$ . Vertices  $z_{i,j} \in D$  for a triangulation of  $D$  are obtained by choosing grid spacing  $h_x$  and  $h_y$ , and defining grid points

$$z_{i,j} = (l_1 + ih_x, l_2 + jh_y) \quad 0 \leq i \leq n_x + 1, 0 \leq j \leq n_y + 1$$

such that  $z_{n_x+1, n_y+1} = (u_1, u_2)$ , where  $h_x = \frac{u_1 - l_1}{n_x + 1}$ ,  $h_y = \frac{u_2 - l_2}{n_y + 1}$ .

$$q(v) = \frac{1}{2} \sum_{i,j} (q_{i,j}^L(v) + q_{i,j}^U(v)) - h_x h_y c \sum_{i,j} v_{i,j},$$

where  $q_{i,j}^L(v) = \frac{h_x h_y}{2} \left[ \left( \frac{v_{i+1,j} - v_{i,j}}{h_x} \right)^2 + \left( \frac{v_{i,j+1} - v_{i,j}}{h_y} \right)^2 \right]$  and  $q_{i,j}^U(v) = \frac{h_x h_y}{2} \left[ \left( \frac{v_{i-1,j} - v_{i,j}}{h_x} \right)^2 + \left( \frac{v_{i,j-1} - v_{i,j}}{h_y} \right)^2 \right]$ .

Note that in this formulation the quadratic  $q_{i,j}^L$  is defined only when  $0 \leq i \leq n_x$  and  $0 \leq j \leq n_y$ , while  $q_{i,j}^U$  is defined  $0 \leq i \leq n_x + 1$  and  $0 \leq j \leq n_y + 1$ .

The results of the experiments appear in Table 1. TAO implementations are tested on an 82-node Dawning 2000-II with the AIX4.2.1 operating system. CG\_PR and CG\_PRP in TAO have some differences. In CG\_FR, the only restart condition is (2) is nonnegative. In CG\_PR, (6) is added to judge restart. In CG\_PRP, (6) and (7) are both added to judge restart.

It's fair to compare the three CG implementations to set the same parameters required. Specifically, we use grid with 1000 points in each direction of two, leading to a problem with 1000,000 variables. The initial point, parameter and tolerance are the same for each solver.

Table 1: Performance of CG methods on Dawning 2000

np	time (speedup)						
	1	2	4	8	16	32	64
tao_cg_fr	2885.63 (1.00)	1563.89 (1.85)	862.16 (3.35)	483.48 (5.98)	276.80 (10.42)	176.93 (16.4)	95.52 (30.06)
tao_cg_pr	3060.88 (1.00)	1625.87 (1.88)	835.25 (3.67)	652.20 (4.69)	305.20 (10.04)	180.79 (16.91)	100.40 (30.61)
tao_cg_prp	3061.96 (1.00)	1631.86 (1.88)	1115.34 (2.75)	589.37 (5.2)	299.19 (10.24)	198.36 (15.46)	121.14 (25.31)

np	Efficiency						
	1	2	4	8	16	32	64
tao_cg_fr	100%	92.5%	83.75%	74.75%	65.13%	51.25%	46.97%
tao_cg_pr	100%	94%	91.75%	58.63%	62.75%	52.84%	47.83%
tao_cg_prp	100%	94%	68.75%	65%	64%	48.31%	39.55%

The results in Table 1 are noteworthy that due to the low memory requirements of iterative solvers, the problem can be solved with only one processor. These results show that the CG implementations have good efficiency, ranging between 40% and 100%. We have noted that as np increases, the overall efficiency decrease. For this particular problem, the efficiency of CG method is acceptable for  $np \leq 16$  processors but drops rapidly with more processors. At every loop, CG\_PR and CG\_PRP have to judge restart, so they need a little more time than CG\_FR does shown in Table 1, in which np stands for number of processors. The time unit is second.

## 5. Conclusion and Discussion

We have simply described several variants of nonlinear conjugate gradient methods. Also we have shown that TAO design leverages external parallel computing infrastructure and linear algebra toolkits to solve large-scale optimization problems on high-performance architecture. TAO extends to general optimization, but the performance issues are more subtle due to the impact of user-supplied function, gradient and Hessian (some algorithms required) code.

## Bibliography

- [1] Yaxiang Yuan, Wenyu Sun, Optimization Theory and Methods, Science Press, Beijing, 1997.
- [2] S.S. Rao, Optimization, Wiley Eastern Limited, 2<sup>nd</sup> Edition, 1984.
- [3] Brett M. Averick, Richard G. Carter, Jorge J. Moré, and Guoliang Xue, The Minipack-2 test problem Collection, Argonne National Laboratory 60439, 1992.
- [4] Steve Benson, Lois Curfman McInnes, Jorge J. Moré, Jason Sarich TAO Users Manual 1.5, Argonne National Laboratory, Jun. 2003.
- [5] Yu-Hong Dai and Ya-Xiang Yuan, A nonlinear conjugate gradient method with a strong global convergence property, *SIAM Journal on Optimization* 10 (1), pp.177-182, 1999.
- [6] Zeyao Mo, Guoxing Yuan, Message Passing Interface---Parallel Programming Environment, Science Press, 2001.
- [7] Yu-Hong Dai, etc. Convergence Properties of nonlinear conjugate gradient methods, *SIAM Journal on Optimization* 10 (2), pp.345-358. 1999.
- [8] Aaron E. Naiman, Ivo M. Babuska, and Howard C. Elman, A Note On Conjugate Gradient Convergence, Volume 76 Issue 2 pp.209-230, 1997.
- [9] J.C. Gilbert and Jorge Nocedal, Global convergence properties of conjugate gradient methods for optimization, *SIAM Journal on Optimization* Vol.2 No.1, pp.21-22, 1992.
- [10] Yu-Hong Dai, Conjugate Gradient Methods with Armijo-type Line Searches, *Acta Mathematicae Applicatae Sinica*, English Series Vol. 18, No. 1, pp.123-130, 2002.
- [11] Jorge Nocedal, Stephen J. Wright, Numerical Optimization, New York, Springer, 1999.
- [12] Shaolin Xi, Nonlinear Optimization Methods, High Education Press, 1999.
- [13] Satish Balay, Kris Buschelman, William Gropp, etc., PETSc 2.1.3 users manual, ANL-95/11-Revision 2.1.3 Argonne National Laboratory, May, 2002.