

# On the Global Convergence of the PERRY-SHANNO Method for Nonconvex Unconstrained Optimization Problems\*

Linghua Huang<sup>1</sup>, Qingjun Wu<sup>2</sup>, Gonglin Yuan<sup>3</sup>

<sup>1</sup>Department of Mathematics and Statistics, Guangxi University of Finance and Economics, Nanning, China

<sup>2</sup>Department of Mathematics and Computer Science, Yulin Normal University, Yulin, China

<sup>3</sup>Department of Mathematics and Information Science, Guangxi University, Nanning, China

E-mail: [linghuahuang@163.com](mailto:linghuahuang@163.com), [wqj600@yahoo.com.cn](mailto:wqj600@yahoo.com.cn), [glyuan@gxu.edu.cn](mailto:glyuan@gxu.edu.cn)

Received November 15, 2010; revised January 12, 2011; accepted January 16, 2011

## Abstract

In this paper, we prove the global convergence of the Perry-Shanno's memoryless quasi-Newton (PSMQN) method with a new inexact line search when applied to nonconvex unconstrained minimization problems. Preliminary numerical results show that the PSMQN with the particularly line search conditions are very promising.

**Keywords:** Unconstrained Optimization, Nonconvex Optimization, Global Convergence

## 1. Introduction

We consider the unconstrained optimization problem:

$$\min \{ f(x) | x \in R^n \}, \quad (1.1)$$

where  $f: R^n \rightarrow R$  is continuously differentiable. Perry and Shanno's memoryless quasi-Newton method is often used to solve the problem (1.1) when  $n$  is large. The PSMQN method was originated from the works of Perry (1977 [1]) and Shanno (1978 [2]), and subsequently developed and analyzed by many authors. Perry and Shanno's memoryless method is an iterative algorithm of the form

$$x_{k+1} = x_k + \alpha_k d_k, \quad (1.2)$$

where  $\alpha_k$  is a steplength, and  $d_k$  is a search direction which given by the following formula:

$$d_1 = -g_1, \quad (1.3)$$

$$d_{k+1} = -\frac{y_k^T s_k}{\|y_k\|^2} g_{k+1} + \left( \frac{y_k^T g_{k+1}}{\|y_k\|^2} - 2 \frac{s_k^T g_{k+1}}{y_k^T s_k} \right) s_k + \frac{s_k^T g_{k+1}}{\|y_k\|^2} y_k, \quad k \geq 1 \quad (1.4)$$

where  $s_k = x_{k+1} - x_k$ ,  $y_k = g_{k+1} - g_k$  and  $g_j$  denotes the

gradient of  $f$  at  $x_j$ . If we denote

$$B_{k+1} = \frac{\|y_k\|^2}{y_k^T s_k} I + \frac{y_k y_k^T}{y_k^T s_k} - \frac{\|y_k\|^2}{s_k^T y_k \|s_k\|^2} s_k s_k^T \quad (1.5)$$

and

$$H_{k+1} = B_{k+1}^{-1},$$

then

$$H_{k+1} = \frac{y_k^T s_k}{\|y_k\|^2} I + 2 \frac{s_k s_k^T}{y_k^T s_k} - \frac{1}{\|y_k\|^2} (s_k y_k^T + y_k s_k^T). \quad (1.6)$$

By (1.4) and (1.5), we can rewrite  $d_{k+1}$  as

$$d_{k+1} = -H_{k+1} g_{k+1}.$$

In practical testing, it is shown that the memoryless method is much more superior to the conjugate gradient methods, and in theoretic analysis, Perry and Shanno had proved that this method will be convergent for uniform convex function with Armijor or Wolfe line search. Shanno pointed out that this method will be convergent for convex function if the Wolfe line search is used. Despite of many efforts has been put to its convergence behavior, the global convergence of the PSMQN method is still open for the case of  $f$  is not a convex function. Recently, Han, Liu and Yin [3] proved the global convergence of the PSMQN method for nonconvex function under the following condition

$$\lim_{k \rightarrow \infty} \|s_k\| = 0.$$

\***Foundation Item:** This work is supported by Program for Excellent Talents in Guangxi Higher Education Institutions, China NSF grants 10761001, Guangxi Education research project grants 201012MS013, and Guangxi SF grants 0991028.

The purpose of this paper is to study the global convergence behavior of PSMQN method by introducing a new line search conditions. The line search strategy used in this paper is as follows: find  $t_k$  satisfying

$$f(x_k + t_k d_k) - f(x_k) \leq \delta t_k g_k^T d_k - \min\{\varepsilon_1, \|g_k\|^\mu\} t_k^2 \|d_k\|^4 \tag{1.7}$$

and

$$g(x_k + t_k d_k)^T d_k \geq \sigma g_k^T d_k, \tag{1.8}$$

where  $\varepsilon_1 \in (0,1)$  is a small scalar and  $\mu \in (1, +\infty)$  is a large scalar. It is clear that (1.7) and (1.8) are a modification of the weak Wolfe-Powell (MWWP) line search conditions.

This paper is organized as follows. In Section 2, we present the PSMQN with the new line search MWWP. In Section 3, we establish the global convergence of the proposed method. The preliminary numerical results are contained in Section 4.

### 2. Algorithm

By combining the PSMQN and the MWWP, we can obtain a modified Perry-Shanno's memoryless quasi-Newton method as follows:

**Algorithm 1** (A Modified PSMQN method: MPS-MQN)

Step 0: Given  $x_1 \in R^n$ , set  $d_1 = -g_1, k = 1$ . If  $g_1 = 0$ , then stop.

Step 1: Find a  $t_k > 0$  satisfying MWWP.

Step 2: Let  $x_{k+1} = x_k + t_k d_k$  and  $g_{k+1} = g(x_{k+1})$ . If  $g_{k+1} = 0$ , then stop.

Step 3: Generate  $d_{k+1}$  by the PSMQN formula (1.4).

Step 4: Set  $k := k + 1$ , go to Step 1.

### 3. Global Convergence

In order to prove the global convergence of Algorithm 1, we will impose the following two assumptions, which have been used often in the literature to analyze the global convergence of conjugate gradient and quasi-Newton methods with inexact line searches.

**Assumption A.** The level set

$$\Omega = \{x \in R^n \mid f(x) \leq f(x_1)\}$$

is bounded.

**Assumption B.** There exists a constant  $L$  such that for any  $x, y \in \Omega$ ,

$$\|g(x) - g(y)\| \leq L \|x - y\|. \tag{3.1}$$

Since  $\{f(x_k)\}$  is a decreasing sequence, it is clear that the sequence  $\{x_k\}$  generated by Algorithm 1 is contained

in, and there exists a constant  $f^*$ , such that

$$\lim_{k \rightarrow \infty} f(x_k) = f^*. \tag{3.2}$$

**Lemma 3.1** Suppose that Assumption A holds and there exists a positive constant  $\varepsilon$  such that

$$\|g_k\| \geq \varepsilon. \tag{3.3}$$

then,

$$\lim_{k \rightarrow \infty} t_k \|d_k\|^2 = 0 \tag{3.4}$$

and

$$\lim_{k \rightarrow \infty} t_k \|d_k\|^2 = 0. \tag{3.5}$$

**Proof.** From (3.2), we have

$$\begin{aligned} \sum_{k=1}^{\infty} (f(x_k) - f(x_{k+1})) &= \lim_{N \rightarrow \infty} \sum_{k=1}^N (f(x_k) - f(x_{k+1})) \\ &= \lim_{N \rightarrow \infty} (f(x_1) - f(x_{k+1})) \\ &= f(x_1) - f^*. \end{aligned}$$

thus

$$\sum_{k=1}^{\infty} (f(x_k) - f(x_{k+1})) < +\infty,$$

which combining with

$$f(x_k + t_k d_k) - f(x_k) \leq \delta t_k g_k^T d_k - \min\{\varepsilon_1, \|g_k\|^\mu\} t_k^2 \|d_k\|^4$$

and (3.3), yields

$$\sum_{k=1}^{\infty} t_k^2 \|d_k\|^4 < +\infty, \tag{3.6}$$

and

$$\sum_{k=1}^{\infty} -t_k g_k^T d_k < +\infty. \tag{3.7}$$

therefore, (3.4) and (3.5) hold.

The property (3.4) is very important for proving the global convergence of Algorithm 1, and it is not known yet for us whether (3.4) holds for other types line search (for example, the weak Wolfe-Powell conditions or the strong Wolfe-Powell conditions).

By using (3.4) and the result given in [3], we can deduce that Algorithm 1 is convergent globally. In the following, we give a direct proof for the global convergence of Algorithm 1.

**Lemma 3.2** Assume that  $B_k$  is a positive definite matrix. Then

$$\frac{\|g_k\|^2}{g_k^T H_k g_k} \geq T_r(B_k). \tag{3.8}$$

**Proof.** Omitted.

**Lemma 3.3** Supposed that Assumption A and B hold, and  $x_k$  is generated by Algorithm 1. Then there exists a positive constant  $c$  such that

$$\sum_{i=1}^k t_k \geq ck. \tag{3.9}$$

**Proof.** By (1.5) and (1.6), we have

$$T_r(B_{k+1}) = n \frac{\|y_k\|^2}{y_k^T s_k}, \tag{3.10}$$

and

$$T_r(H_{k+1}) = (n-2) \frac{\|y_k\|^T s_k}{\|y_k\|^2} + 2 \frac{\|s_k\|^2}{y_k^T s_k}. \tag{3.11}$$

From  $(y_k^T s_k)^2 \leq \|s_k\|^2 \|y_k\|^2$ , we obtain

$$T_r(H_{k+1}) = (n-2) \frac{\|y_k\|^T s_k}{\|y_k\|^2} + 2 \frac{\|s_k\|^2}{y_k^T s_k}. \tag{3.12}$$

Using (1.8) and (3.12), we have

$$T_r(H_{k+1}) \leq \frac{n}{1-\beta} \frac{t_k \|H_k g_k\|^2}{g_k^T H_k g_k}. \tag{3.13}$$

By using the positive definiteness of  $H_k$ , we have

$$\frac{\|H_k g_k\|^2}{g_k^T H_k g_k} \leq T_r(H_k), \tag{3.14}$$

from which we can deduce that

$$T_r(H_{k+1}) \leq \left(\frac{n}{1-\beta}\right)^k T_r(H_1) \prod_{i=1}^k t_k. \tag{3.15}$$

By the Assumption B, (3.10) and (1.8), we get

$$\begin{aligned} T_r(B_{k+1}) &= n \frac{\|y_k\|^2}{y_k^T s_k} \\ &\leq nL^2 \frac{\|s_k\|^2}{-(1-\beta) g_k^T s_k} \\ &= t_k \frac{nL^2 \|H_k g_k\|^2}{1-\beta g_k^T H_k g_k}, \\ &= c_1 t_k \frac{\|H_k g_k\|^2}{g_k^T H_k g_k} \\ &\leq c_1 t_k T_r(H_k), \end{aligned}$$

where  $c_1 = \frac{nL^2}{1-\beta}$ . Combining with (3.15) yields

$$T_r(B_{k+1}) \leq c_1 t_k \left(\frac{n}{1-\beta}\right)^{k-1} T_r(H_1) \prod_{i=1}^k t_k.$$

Adding above inequality to (3.15), we obtain

$$T_r(B_{k+1}) + T_r(H_{k+1}) \leq \frac{n}{(1-\beta)^k} T_r(H_1) \left[1 + \frac{c_1(1-\beta)}{n}\right] \prod_{i=1}^k t_k.$$

Now, if we denote the eigenvalues of  $B_k$  by  $h_{k1}, h_{k2}, \dots, h_{kn}$ , then the eigenvalues of  $H_k$  are  $\frac{1}{h_{k1}}, \frac{1}{h_{k2}}, \dots, \frac{1}{h_{kn}}$ , so we have

$$T_r(B_{k+1}) + T_r(H_{k+1}) = \sum_{i=1}^n \left(h_{ki} + \frac{1}{h_{ki}}\right) \geq 2n.$$

Thus we have

$$\prod_{i=1}^k t_k \geq \frac{2n}{\left(\frac{n}{1-\beta}\right)^k T_r(H_1) \left[1 + \frac{c_1(1-\beta)}{n}\right]}.$$

Hence there exists a positive constant  $c$  such that

$$\prod_{i=1}^k t_k \geq c^k.$$

Therefore

$$\sum_{i=1}^k t_k \geq ck.$$

**Theorem 3.1** Supposed that Assumption A and B hold, and  $x_k$  is generated by Algorithm 1. Then

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0. \tag{3.16}$$

**Proof.** Suppose that the conclusion doesn't hold, i.e., there exists a positive constant  $\varepsilon$  such that

$$\|g_k\| \geq \varepsilon.$$

From (3.3), (3.10) and Lemma 3.3, we have

$$\begin{aligned} T_r(B_{k+1}) &= n \frac{\|y_k\|^2}{y_k^T s_k} \\ &\leq n \frac{\|y_k\|^2}{t_k (1-\beta) g_k^T H_k g_k} \\ &= n \frac{\|y_k\|^2 \|g_k\|^2}{t_k \|g_k\|^2 (1-\beta) g_k^T H_k g_k} \\ &\leq \frac{nL^2 \|s_k\|^2 \|g_k\|^2}{(1-\beta) \varepsilon^2 t_k g_k^T H_k g_k} \\ &\leq \frac{nL^2 \|s_k\|^2}{(1-\beta) \varepsilon^2 t_k} T_r(B_k) \\ &\leq \dots \\ &\leq \left(\frac{nL^2}{(1-\beta) \varepsilon^2}\right)^k T_r(B_1) \left[\frac{\left(\prod_{i=1}^k \|s_i\|^2\right)}{\left(\prod_{i=1}^k t_i\right)}\right] \\ &= c_2^k T_r(B_1) \prod_{i=1}^k t_i \|d_i\|^2, \end{aligned}$$

where  $c_2 = \frac{nL^2}{(1-\beta) \varepsilon^2}$ . From (3.4) we may obtain that

there exists a positive constant  $k_0$  such that

$$t_i \|d_i\|^2 \leq \frac{1}{c_2} \text{ for all } i \geq k_0.$$

Therefore

$$\begin{aligned} T_r(B_{k+1}) &\leq (c_2)^{k_0} (c_2)^{k-k_0} T_r(B_1) \prod_{i=1}^{k_0} t_i \|d_i\|^2 \prod_{i=k_0+1}^k t_i \|d_i\|^2 \\ &\leq (c_2)^{k_0} T_r(B_1) \prod_{i=1}^{k_0} t_i \|d_i\|^2 (\underline{\Delta}c_3) \\ &\leq c_3 (k+1) \\ &\leq \frac{c_3}{c} \sum_{i=1}^{k+1} t_i. \end{aligned}$$

Hence from the above inequality and Lemma 3.2 and Lemma 3.3, we have

$$\begin{aligned} \sum_{i=1}^{\infty} \left( \frac{t_k}{\sum_{i=1}^k t_k} \right) &\leq \frac{c_3}{c} \sum_{i=1}^{\infty} \frac{t_k}{T_r(B_k)} \\ &\leq \frac{c_3}{c} \sum_{i=1}^{\infty} \frac{t_k g_k^T H_k g_k}{\|g_k\|^2} \\ &\leq \frac{c_3}{c\mathcal{E}^2} \sum_{i=1}^{\infty} t_k g_k^T H_k g_k < \infty. \end{aligned}$$

Therefore

$$\sum_{i=1}^{\infty} t_k < \infty,$$

which contradicts to Lemma 3.3. Thus (3.16) holds.

**Remark:** In this remark, we give a cautious update of Perry-Shanno’s Memoryless quasi-Newton method (CP-SMQN), we do not discuss the global convergence here. We let it to be a topic of further research.

**Algorithm 2:** (CPSMQN method)

Step 0: Given  $x_1 \in R^n$ , set  $d_1 = -g_1, k = 1$ . If  $g_1 = 0$ , then stop.

Step 1: Find a  $t_k > 0$  satisfying WWP.

Step 2: Let  $x_{k+1} = x_k + t_k d_k$  and  $g_{k+1} = g(x_{k+1})$ . If  $g_{k+1} = 0$ , then stop.

Step 3: Choose  $B_{k+1}$  by the following equation and Generate  $d_{k+1}$ ,

$$B_{k+1} = \begin{cases} \frac{\|y_k\|^2}{y_k^T s_k} I + \frac{y_k y_k^T}{y_k^T s_k} - \frac{\|y_k\|^2}{s_k^T y_k \|s_k\|^2} s_k s_k^T & \text{if } \frac{-g_k^T s_k}{\|s_k\|^2} \geq m, \\ B_k & \text{else} \end{cases} \quad (3.17)$$

where  $m$  is a positive constant.

### 4. Numerical Experiments

In this section, we report the numerical results for PSM-

QN, MPSMQN and CPSMQN method. The problems that we tested are from [4]. For each test problem, the termination condition is

$$\|g(x_k)\| \leq 10^{-5}.$$

We will test the following three methods:

• **PSMQN:** the Perry-Shanno’s Memoryless quasi-Newton method with the WWP, where  $\delta = 0.1$  and  $\sigma = 0.9$ ;

• **MPSMQN:** Algorithm 1 with  $\delta = 0.1, \sigma = 0.9, \epsilon_1 = 10^{-16}$  and  $\mu = 10$ ;

• **CPSMQN:** Algorithm 2 with  $\delta = 0.1, \sigma = 0.9$  and  $m = 10^{-18}$ .

In order to rank the iterative numerical methods, one can compute the total number of function and gradient evaluations by the formula

$$N_{total} = NF + m * NG, \quad (4.1)$$

where  $NF, NG$  denote the number of function evaluations and gradient evaluations, respectively, and  $m$  is some integer. According to the results on automatic differentiation ([5] and [6]), the value of  $m$  can be set to  $m = 5$ . That is to say, one gradient evaluation is equivalent to  $m$  number of function evaluations if automatic differentiation is used.

**Tables 1** shows the computation results, where the columns have the following meanings:

- Problem: the name of the test problem in MATLAB;
- Dim: the dimension of the problem;
- NI: the number of iterations;
- NF: the number of function evaluations;
- NG: the number of gradient evaluations;
- SD: the number of iterations for which the steepest descent direction used;

In this part, we compare the PSMQN, MPSMQN and CPSMQN method as follows: for each testing example  $i$ , compute the total numbers of function evaluations and gradient evaluations required by the evaluated method  $j(EM(j))$  and the PSMQN method by the formula (4.1), and denote them by  $N_{total,i(EM(j))}$  and  $N_{total,i(PSMQN)}$ ; then calculate the ratio

$$r_i(EM(j)) = \frac{N_{total,i(EM(j))}}{N_{total,i(PSMQN)}}. \quad (4.2)$$

If  $EM(j_0)$  does not work for example  $i_0$ , we replace the  $N_{total,i_0(EM(j_0))}$  by a positive constant  $\tau$  which define as follows

$$\tau = \max \{ N_{total,i(EM(j))} : (i, j) \notin S_1 \},$$

where

$$S_1 = \{ (i, j) : \text{method } j \text{ does not work for example } i \}.$$

**Table 1. Test results for PSMQN/MPSMQN/CPSMQN.**

		PSMQN	MPSMQN	CPSMQN
Problems	Dim	NI/NF/NG/SD	NI/NF/NG/SD	NI/NF/NG/SD
ROSE	2	64/133/86/0	65/143/82/0	64/133/86/0
FROTH	2	50/137/68/0	34/56/35/0	50/137/68/0
BADSCP	2	161/558/324/3	161/558/324/3	123/408/279/1
BADSCB	2	48/255/56/4	22/35/23/0	48/255/56/4
BEALE	2	36/74/43/0	36/74/43/0	36/74/43/0
JENSAM	2	37/69/46/0	37/69/46/0	37/69/46/0
HELIX	3	53/77/57/0	53/77/57/0	53/77/57/0
BARD	3	92/117/95/0	92/117/95/0	92/117/95/0
GAUSS	3	6/14/10/0	6/14/10/0	6/14/10/0
MEYER	3	-	-	-
GULF	3	1/4/2/0	1/4/2/0	1/4/2/0
BOX	3	100/170/113/0	100/170/113/0	100/170/113/0
SING	4	147/212/159/0	175/237/182/0	147/212/159/0
WOOD	4	138/311/248/0	138/311/248/0	139/321/244/1
KOWOSB	4	142/200/148/0	142/200/148/0	142/200/148/0
BD	4	-	-	-
OSB1	5	-	-	-
BIGGS	6	518/757/554/0	518/757/554/0	518/757/554/0
OSB2	11	610/717/620/0	610/717/620/0	610/717/620/0
WATSON	20	-	-	-
ROSEX	8	55/115/73/0	54/114/66/0	55/115/73/0
	50	53/99/68/0	53/99/68/0	53/99/68/0
	100	75/149/97/0	75/149/97/0	75/149/97/0
SINGX	4	147/212/159/0	147/212/159/0	147/212/159/0
PEN1	2	7/13/9/0	7/13/9/0	7/13/9/0
PEN2	4	28/44/31/0	28/44/31/0	28/44/31/0
	50	481/845/539/0	481/845/539/0	481/845/539/0
VARDIM	2	5/13/6/0	5/13/6/0	5/13/6/0
	50	24/65/28/0	24/65/28/0	24/65/28/0
	100	30/77/33/0	30/77/33/0	30/77/33/0
	200	40/135/41/2	40/135/41/2	40/135/41/2
TRIG	3	26/31/27/0	26/31/27/0	26/31/27/0
	50	52/59/53/0	52/59/53/0	52/59/53/0
	100	54/60/55/0	54/60/55/0	54/60/55/0
BV	3	26/40/27/0	26/40/27/0	26/40/27/0
	10	177/200/178/0	177/200/178/0	177/200/178/0
IE	3	7/9/8/0	7/9/8/0	7/9/8/0
	50	8/10/9/0	8/10/9/0	8/10/9/0
	100	8/10/9/0	8/10/9/0	8/10/9/0
	200	8/10/9/0	8/10/9/0	8/10/9/0
	500	9/11/10/0	9/11/10/0	9/11/10/0
TRID	3	21/30/22/0	21/30/22/0	21/30/22/0
	50	30/39/31/0	30/39/31/0	30/39/31/0
	100	32/39/33/0	32/39/33/0	32/39/33/0
	200	63/74/64/0	63/74/64/0	63/74/64/0
BAND	2	11/21/12/0	11/21/12/0	11/21/12/0
LIN	2	1/3/2/0	1/3/2/0	1/3/2/0
	50	1/3/2/0	1/3/2/0	1/3/2/0
	500	1/3/2/0	1/3/2/0	1/3/2/0
	1000	1/3/2/0	1/3/2/0	1/3/2/0
LIN1	2	2/10/3/0	2/10/3/0	2/10/3/0
	10	2/21/3/0	2/21/3/0	2/21/3/0
LIN2	4	2/11/3/0	2/14/3/0	2/14/3/0

**Table 2. Relative efficiency of PSMQN, MPSMQN and CPSMQN.**

PSMQN	MPSMQN	CPSMQN
1	0.9752	0.9963

The geometric mean of these ratios for method  $EM(j)$  over all the test problems is defined by

$$r(EM(j)) = \left( \prod_{i \in S} r_i(EM(j)) \right)^{1/|S|}, \quad (4.3)$$

where  $S$  denotes the set of the test problems and  $|S|$  the number of elements in  $S$ . One advantage of the above rule is that, the comparison is relative and hence does not be dominated by a few problems for which the method requires a great deal of function evaluations and gradient functions.

From **Table 2**, we observe that the average performances of the Algorithm 1 are the best among the three methods, and the average performances of the Algorithm 2 are little better than the PSMQN method. Therefore, the Algorithm 1 is the best method among the three methods given in this paper from both theory and the computational point of view.

## 5. References

- [1] J. M. Perry, "A Class of Conjugate Algorithms with a Two Step Variable Metric Memory," Discussion paper 269, Northwestern University, 1977.
- [2] D. F. Shanno, "On the Convergence of a New Conjugate Gradient Algorithm," *SIAM Journal on Numerical Analysis*, Vol. 15, No. 6, 1978, pp. 1247-1257. doi:10.1137/0715085
- [3] J. Y. Han, G. H. Liu and H. X. Yin, "Convergence of Perry and Shanno's Memoryless Quasi-Newton Method of Nonconvex Optimization Problems," *On Transactions*, Vol. 1, No. 3, 1997, pp. 22-28.
- [4] J. J. More, B. S. Garbow and K. E. Hillstrome, "Testing Unconstrained Optimization Software," *ACM Transactions on Mathematical Software*, Vol. 7, No. 1, 1981, pp. 17-41. doi:10.1145/355934.355936
- [5] A. Griewank, "On Automatic Differentiation," Kluwer Academic Publishers, Norwell, 1989.
- [6] Y. Dai and Q. Ni, "Testing Different Conjugate Gradient Methods for Large-Scale Unconstrained Optimization," *Journal of Computational Mathematics*, Vol. 21, No. 3, 2003, pp. 311-320.