DOI: 10.13656/j.cnki.gxkx.2003.04.004
广西科学 Guangxi Sciences 2003, 10 (4): 253~257, 261

A Modified BFGS Method with Global Convergence for Unconstrained Optimization Problems
无约束最优化问题中具有全局收敛性的修改的 BFGS方法

Xiao Yunhai　Ye Hun
肖运海　叶魂

(Dept. of Math. & Info. Sci., Guangxi Univ., 100 Daxuelu, Nanning, Guangxi, 530004, China)
(广西大学数学与信息科学系　南宁市大学路 100号　530004)

Abstract　A new BFGS-type formula and a new BFGS-type method with weak Wolfe-Powell (WWP) step size rule are presented. The numerical results are better than that by using the other method mentioned in relevant references.

Key words　unconstrained optimization, Broyden-Fletcher-Glodfard-shanna, quasi-Newton method, global convergence

摘要　给出新的 BFGS型公式,并利用弱的 Wolfe-Powell步长准则给出新的 BFGS型方法.该方法的数值结果比相关文献的方法好.

关键词　无约束优化　BFGS　拟牛顿方法　全局收敛
中图法分类号　O221.2

1 Introduction

Consider the unconstrained optimization problem

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

where $f(x)$ is continuously differentiable, whose gradient at x_k will be denoted by g_k, i.e., $\nabla f(x_k) = g_k$. Quasi-Newton methods for solving (1.1) often need to update the iterate matrix B_k. Traditionally, $\{B_k\}$ satisfies the following Quasi-Newton equation

$$B_{k+1} S_k = y_k, \quad (1.2)$$

where $y_k = g_{k+1} - g_k$ and $s_k = x_{k+1} - x_k$. The very famous update B_k is the BFGS formula

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{s_k^T y_k}. \quad (1.3)$$

It has been shown that BFGS is the most effective in Quasi-Newton methods. But the global convergence for a general function f is still open even, if it is convergent (globally and superlinearly) for convex minimization [1~7]. Our pioneers have made great efforts to find out a Quasi-Newton method which is not only possessing global convergence but also superior than BFGS [8~14]. In Reference [12], Wei, Li and Qi proposed a new quasi-Newton equation as follows. If we use the Taylor formula to the objective function $f(x)$, we have

$$f(x) \simeq f(x_{k+1}) + \nabla f(x_{k+1})^T (x - x_{k+1}) + \frac{1}{2}(x - x_{k+1})^T \nabla^2 f(x_{k+1})(x - x_{k+1}).$$

Hence

$$f(x_k) \simeq f(x_{k+1}) - \nabla f(x_{k+1})^T s_k + \frac{1}{2} s_k^T \nabla^2 f(x_{k+1}) s_k.$$

Therefore

$$s_k^T \nabla^2 f(x_{k+1}) s_k \simeq 2[f(x_k) - f(x_{k+1})] + 2\nabla f(x_{k+1})^T s_k = 2[f(x_k) - f(x_{k+1})] + (g_{k+1} + g_k)^T s_k + s_k^T y_k.$$

The above equality gives us a new idea that, if we set

$$A_k = \frac{2[f(x_k) - f(x_{k+1})] + (g_{k+1} + g_x)^T s_k}{\|s_k\|^2} I \quad (1.4)$$

and

$$y_k^* = y_k + A_k s_k,$$

which replace y_k in (1.2), then we can get

$$B_{k+1} s_k = y_k^* = y_k +$$

2003-03-27收稿, 2003-06-02修回。

$$\frac{2[f(x_k)-f(x_{k+1})]+(g_{k+1}+g_k)^T s_k}{\|s_k\|^2} s_k. \quad (1.5)$$

In Reference [12], Wei, Li and Qi replace all the y_k in (1.3) and in the following modified BFGS method

$$B_{k+1}=B_k-\frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k}+\frac{y_k^*(y_k^*)^T}{s_k^T y_k^*}. \quad (1.6)$$

But we found that the numerical behavior is not good enough. Now, we replace two y_k in (1.3) only and get another modified BFGS formula (MBFGS)

$$B_{k+1}=B_k-\frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k}+\frac{(y_k+A_k s_k)(y_k+A_k s_k)^T}{s_k^T y_k} \quad (1.7)$$

$$=B_k-\frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k}+\frac{y_k y_k^T}{s_k^T y_k}+\frac{y_k(A_k s_k)^T+A_k s_k y_k^T+A_k s_k(A_k s_k)^T}{s_k^T y_k}=BFGS+\frac{y_k(A_k s_k)^T+A_k s_k y_k^T+A_k s_k(A_k s_k)^T}{s_k^T y_k}. \quad (1.8)$$

Using (1.8), (1.4) and the following weak Wolfe-Powell (WWP) step-size rule

$$f(x_{k+1})\leqslant f(x_k)+\delta\alpha_k g_k^T d_k \quad (1.9)$$

and

$$g_{k+1}^T d_k\geqslant \sigma g_k^T d_k, \quad (1.10)$$

where $\delta\in(0,1/2)$ and $\sigma\in(\delta,1)$, we proposed the following algorithms.

Algorithm MBFGS

Step 0 Choose an initial point $x_1\in R^n$ and a symmetric positive definite matrix $B_0\geqslant 0$. Let $\varepsilon>0$ and set $k=1$.

Step 1 If $\|g_k\|\leqslant\varepsilon$, stop.

Step 2 Solve $B_k d_k+g_k=0$ to obtain a search direction d_k.

Step 3 Find α_k by WWP.

Step 4 Set $x_{k+1}=x_k+\alpha_k d_k$. Calculate updated matrix B_{k+1} by formula (1.8).

Step 5 Set $k=k+1$ and go to step 1.

If calculating the updated matrix by Formula (1.6), we will get Algorithm WLQBFGS.

This paper is organized as follows. The global convergence properties of the MBFGS are represented in the next section. The preliminary numerical results for the Algorithm MBFGS are given in section 3, and the results would be compared with that by using WLQBFGS method and the original BFGS method.

2 Global convergence analysis

In order to obtain the global convergence, we need the following assumptions.

Assumption 2.1 The level set

$$K=\{x\mid f(x)\leqslant f(x_0)\}$$

is contained in a bounded convex set D.

Assumption 2.2 The function f is continuously differentiable on D and there exists a constant $L>0$ such that

$$\|g(x)-g(y)\|\leqslant L\|x-y\|, \text{ for all } x,y\in D.$$

Assumption 2.3 The function f is uniformly convex, i.e., there are positive constants λ_1 and λ_2 such that

$$\lambda_1\|z\|^2\leqslant z^T G(x) z\leqslant\lambda_2\|z\|^2$$

for all $x,z\in R^n$, where G denotes the Hessian matrix of f.

Since $\{f(x_k)\}$ is a decreasing sequence, it is clear that the sequence $\{x_k\}$ generated by Algorithm MBFGS is contained in K, and there exists a constant f^* such that

$$\lim_{k\to\infty} f(x_k)=f^*. \quad (2.1)$$

Moreover, from the fact that $\{x_k\}$ is bounded, by using Assumption 2.2, we can deduce that there exists $M\geqslant 0$ such that for all k

$$\|g_k\|\leqslant M. \quad (2.2)$$

To establish the global convergence of Algorithm MBFGS, we give some useful lemmas.

Lemma 2.1 Let $\{x_k\}$ be generated by Algorithm MBFGS, then we have

$$m_1\|s_k\|^2\leqslant s_k^T y_k\leqslant m_2\|s_k\|^2, \quad (2.3)$$

$$\lambda_1\|s_k\|^2\leqslant s_k^T y_k^*\leqslant\lambda_2\|s_k\|^2, \quad (2.4)$$

$$\sum_{k=0}^{+\infty}(-g_k^T s_k)<+\infty \quad (2.5)$$

and

$$\|y_k^*\|\leqslant(2L+\lambda_2)s_k. \quad (2.6)$$

Proof From (2.1) we have

$$\sum_{k=1}^{\infty}(f(x_k)-f(x_{k+1}))=\lim_{N\to+\infty}\sum_{k=1}^{N}(f(x_k)-f(x_{k+1}))=\lim_{N\to+\infty}(f(x_1)-f(x_{k+1}))=f(x_1)-f^*.$$

Thus

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

which combines with

$$f(x_{k+1})\leqslant f(x_k)+\delta\alpha_k g_k^T d_k,$$

yields

$$\sum_{k=1}^{\infty}(-\alpha_k g_k^T d_k)<+\infty.$$

Therefore, (2.5) holds. From the definition of y_k^*, we have

$\|y_k^*\| = \|y_k + (2[f(x_k) - f(x_{k+1})] + (g(x_{k+1}) + g(x_k))^T s_k) s_k / \|s_k\|^2\| \leqslant \|y_k\| + |2[f(x_k) - f(x_{k+1})] + (g(x_{k+1}) + g(x_k))^T s_k| / \|s_k\| \leqslant 2\|y_k\| + |s_k^T G(x_k + \theta(x_{k+1} - x_k)) s_k| / \|s_k\| \leqslant 2L\|s_k\| + \lambda_2 \|s_k\| = (2L + \lambda_2) s_k$.

Therefore, (2.6) holds. Using (1.4) and Taylor expansion, we have

$s_k^T y_k^* = s_k^T (y_k + (2 s_k [f(x_k) - f(x_{k+1})] + (g_{k+1} - g_k)^T s_k) / \|s_k\|^2) = s_k^T y_k + 2[f(x_k) - f(x_{k+1})] + (g_{k+1} - g_k)^T s_k = 2[f(x_k) - f(x_{k+1})] + 2 g_{k+1}^T s_k = 2[-g_{k+1}^T s_k + \frac{1}{2} s_k^T G(x_k + \theta(x_{k+1} - x_k)) s_k] + 2 g_{k+1}^T s_k = s_k^T G(x_k + \theta(x_{k+1} - x_k)) s_k$.

Hence (2.4) holds by Assumption 2.3. (2.3) can be got from (2.2) and Assumption 2.3, which is omitted here.

The above lemma indicates that $y_k s_k > 0$, which combines with (1.7) yields $B_{k+1} > 0$, so that $\{B_k\}$ is a positive definite sequence.

Lemma 2.2 Let $\{x_k\}$ be generated by Algorithm MBFGS. Suppose that (2.3) holds, then there must be a positive constant M_1 such that

$$\mathrm{Tr}(B_{k+1}) \leqslant M_1 (k+1) \tag{2.7}$$

and

$$\sum_{i=0}^{k} \frac{\|B_i s_i\|^2}{s_i^T B_i s_i} \leqslant M_2 (k+1). \tag{2.8}$$

Proof From Lemma 2.1, by taking the trace operation in both sides of (1.8), we have

$$\mathrm{Tr}(B_{k+1}) = \mathrm{Tr}(B_k) - \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} + \frac{\|y_k^*\|^2}{y_k^T s_k} \leqslant \mathrm{Tr}(B_k) - \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} + \frac{(2L + m_2)^2}{m_1} \leqslant \cdots \leqslant \mathrm{Tr}(B_0) - \sum_{i=0}^{k} \frac{\|B_i s_i\|^2}{s_i^T B_i s_i} + \frac{(2L + m_2)^2}{m_1}(k+1).$$

Using that B_{k+1} is positive definite, we have $\mathrm{Tr}(B_{k+1}) > 0$. Therefore, the last inequality implies (2.7) and (2.8).

We can also see the similar proof of the above two lemmas in Reference [14].

Lemma 2.3 Let $\{x_k\}$ be generated by Algorithm MBFGS and G is continuous at x^*. Then we have

$$\lim_{k \to \infty} \|A_k\| = 0. \tag{2.9}$$

Proof By using Taylor's formula, we have

$y_k^T s_k = (g_{k+1} - g_k)^T s_k = s_k^T G(\xi_{1k}) s_k$

and

$$f(x_k) - f(x_{k+1}) = -g_{k+1}^T s_k + \frac{1}{2} s_k^T G(\xi_{2k}) s_k.$$

Where $\xi_{1k} = \theta_{1k}(x_{k+1} - x_k)$, $\xi_{2k} = \theta_{2k}(x_{k+1} - x_k)$, and $\theta_{1k}, \theta_{2k} \in (0, 1)$. From the definition of A_k and the following equality

$$f(x_k) = f(x_{k+1}) + g_{k+1}^T (x_k - x_{k+1}) + \frac{1}{2}(x_k - x_{k+1}) B_{k+1} (x_k - x_{k+1}),$$

we get

$$A_k = \frac{s_k^T B_{k+1} s_k - s_k^T G(\xi_{1k}) s_k}{\|s_k\|^2} I$$

and

$$s_k^T B_{k+1} s_k = s^T G(\xi_{2k}) s_k.$$

Hence

$$\|A_k\| \leqslant \|G(\xi_{2k}) - G(\xi_{1k})\|.$$

Therefore, (2.9) holds.

Lemma 2.4 Let $\{x_k\}$ be generated by Algorithm MBFGS, then there must be a positive constant c_1 such that

$$\prod_{i=0}^{k} \alpha_i \geqslant c_1^k. \tag{2.10}$$

Proof Using $s_k = -\alpha_k B_k^{-1} g_k$ and (1.10), we have

$$(1 - \sigma) s_k^T B_k s_k = -(1 - \sigma) \alpha_k s_k^T g_k \leqslant \alpha_k s_k^T y_k = \alpha_k s_k^T [\int_0^1 G(x_k + t s_k) \, dt] s_k.$$

Therefore,

$$\alpha_k \geqslant (1 - \sigma) \frac{s_k^T B_k s_k}{s_k^T \hat{G}_k s_k}.$$

Combining with (2.4) and Assumpton 2.3, we obtain

$$\frac{s_k^T y_k^*}{s_k^T B_k s_k} \geqslant \frac{1}{\alpha_k} C,$$

where $C = \frac{\lambda_1 (1 - \sigma)}{\lambda_2}$ and $\hat{G}_k = \int_0^1 G(x_k + t s_k) \, dt$.

From Lemma 2.1, by taking the determinant in both sides of (1.8), we have

$$\mathrm{Det}(B_{k+1}) \geqslant \mathrm{Det}(B_k) \frac{y_k^* s_k}{y_k s_k} \frac{(y_k^*)^T s_k}{s_k^T B_k s_k} \geqslant \mathrm{Det}(B_k) \frac{\lambda_2}{m_1} \frac{C}{\alpha_k} = \frac{D}{\alpha_k} \mathrm{Det}(B_k) \geqslant \cdots \geqslant D^{k+1} \mathrm{Det}(B_0) \prod_{i=0}^{k} \frac{1}{\alpha_i},$$

where $D = \frac{\lambda_2 C}{m_1}$. Using the following inequality

$$\mathrm{Det}(B_{k+1}) \leqslant [\frac{1}{n} \mathrm{Tr}(B_{k+1})]^n$$

and (2.7), we obtain that

$$\prod_{i=0}^{k} \alpha_i \geqslant \frac{D^{k+1} \mathrm{Det}(B_0)}{\mathrm{Det}(B_{k+1})} \geqslant \frac{D^{k+1} \mathrm{Det}(B_0)}{[\frac{Tr(B_{k+1})}{n}]^n} \geqslant$$

$$\frac{D^{k+1}\mathrm{Det}(B_0)}{[\frac{M_1(k+1)}{n}]^n}.$$

Therefore (2.10) holds for all large k.

The following theorem is taken from Theorem 5.1 of Reference[14].

Theorem 2.1 Let $\{x_k\}$ be generated by Algorithm MBFGS. Then, we have

$$\liminf_{k\to\infty}\|g_k\| = 0. \quad (2.11)$$

Proof Suppose that the conclusion does not hold, then there exists a constant $X \geqslant 0$ such that for all k,

$$\|g_k\| \geqslant X.$$

Hence

$$+\infty > \sum_{k=0}^{\infty}(-g_k^T s_k) = \sum_{k=0}^{\infty}(\frac{s_k^T B_k s_k}{T_k}) =$$

$$\sum_{k=0}^{\infty}(\frac{s_k^T B_k s_k \|g_k\|}{\|B_k s_k\|}) = \sum_{k=0}^{\infty}(T_k\|g_k\|^2\frac{s_k^T B_k s_k}{\|B_k s_k\|^2}) \geqslant$$

$$X\sum_{k=0}^{\infty}(T_k\frac{s_k^T B_k s_k}{\|B_k s_k\|^2}).$$

Therefore, for any $Y > 0$ there exists constants k_0 such that for any positive integer q,

$$q\{\prod_{k=k_0+1}^{k_0+q} T_k\frac{s_k^T B_k s_k}{\|B_k s_k\|^2}\}^{\frac{1}{q}} \leqslant \sum_{k=k_0+1}^{k_0+q} T_k\frac{s_k^T B_k s_k}{\|B_k s_k\|^2} \leqslant Y,$$

where the left hand side of the inequality follows from the geometric inequality. Thus

$$(\prod_{k=k_0+1}^{k_0+q} T_k)^{\frac{1}{q}} \leqslant \frac{Y}{q}(\prod_{k=k_0+1}^{k_0+q}\frac{\|B_k s_k\|^2}{s_k^T B_k s_k})^{\frac{1}{q}} \leqslant$$

$$\frac{Y}{q^2}\sum_{k=k_0+1}^{k_0+q}\frac{\|B_k s_k\|^2}{s_k^T B_k s_k} \leqslant \frac{Y}{q^2}\sum_{k=0}^{k_0+q}\frac{\|B_k s_k\|^2}{s_k^T B_k s_k} \leqslant$$

$$\frac{Y(k_0+q+1)}{q^2}M_2.$$

Let $q \to \infty$ yield a contraction, because Lemma 2.4 ensures that the left hand side of the above inequality is greater than a positive constant, we get (2.11).

3 Numerical results

The numerical results for Algorithm MBFGS will be reported and compared with that for the original BFGS method in this section. The 34 problems that we tested come from the website ftp://ftp.mathworks.com/. The code was written in MATLAB 6.1 and in double precision arithmetic. All runs were performed on PC(CPU PentiumⅣ 1.7G). For each problem, the termination condition is

$$\|g(x_k)\| \leqslant 10^{-6}.$$

For each problem, we choose the initial matrix $B_0 = I$, i.e., the unit matrix. We will test the following quasi-Newton methods.

BFGS Methods: The BFGS method with the weak Wolfe-Powll(WWP) step-size rule and $W = 0.1$, $e = 0.9$

MBFGS Methods: The Algorithm MBFGS method with the WWP, and $W = 0.1$, $e = 0.9$.

WLQBFGS Methods: The Algorithm WLQMBFGS method with the WWP, and $W = 0.1$, $e = 0.9$.

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

$$N_{\text{total}} = NF + m*NG, \quad (3.1)$$

where NF, NG denote the number of times of function evaluations and gradient evaluation respectively, and m is an integer. According to the results of automatic differentiation[15,16], the value of m can be set to $m = 5$. It means that one gradient evaluation is equivalent to m times of function evaluation in automatic differentiation.

Table 1 shows the results of BFGS and MBFGS method, 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.

We compare BFGS and MBFGS in the following way. For each testing example i, compute the total times of function evaluations and gradient evaluations according to the evaluated methods (MBFGS) and BFGS respectively, and denote them by $N_{\text{total},i}$(MBFGS) and $N_{\text{total},i}$(BFGS); then calculate the ratio

$$r_i(\text{MBFGS}) = \frac{N_{\text{total},i}(\text{MBFGS})}{N_{\text{total},i}(\text{BFGS})}. \quad (3.2)$$

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

$$f = \max\{N_{\text{total},i}(\text{MBFGS}): (i,j) \notin S_1\},$$

where

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

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

$$r(EM(j)) = \left(\prod_{i \in S} r_i(\mathrm{MBFGS})\right)^{1/|S|}, \qquad (3.3)$$

Table 1 Test results for BFGS, MBFGS and WLQBFGS methods

No.	Problems	Dim	BFGS	MBFGS	WLQBFGS
			NI/NF/NG	*NI/NF/NG*	*NI/NF/NG*
1	ROSE	2	34/54/35	30/49/31	29/51/30
2	FROTH	2	10/22/11	8/20/9	10/22/11
3	BADSCP	2	158/233/159	146/212/149	166/244/167
4	BADSCB	2	12/56/13	12/54/13	12/55/13
5	BEALE	2	15/24/16	12/21/13	15/25/16
6	JENSAM	2	11/24/13	11/24/13	14/26/15
7	HELIX	3	28/56/30	25/52/27	28/55/29
8	BARD	3	23/34/24	21/32/22	21/34/23
9	GAUSS	3	4/7/5	4/7/5	4/7/5
10	MEYER		–	–	–
11	GULF	3	1/4/2	1/4/2	1/4/2
12	BOX	3	30/41/32	23/36/24	21/39/24
13	SING	4	29/52/30	38/63/40	23/46/24
14	WOOD	4	52/97/53	51/91/52	53/93/54
15	KOWOSB	4	28/32/29	28/33/29	28/32/29
16	BD	4	23/82/24	19/76/20	–
17	OSB1	5	–	–	–
18	BIGGS	6	36/47/39	30/38/32	35/46/36
19	OSB2	11	53/80/54	53/78/54	56/82/57
20	WATSON	20	55/91/56	57/93/58	56/93/58
21	ROSEX	8	86/152/87	74/133/75	80/140/81
		50	256/652/257	233/607/234	232/600/233
22	SINX	4	29/52/30	38/63/40	23/46/24
23	PEN1	2	179/262/182	166/233/169	178/256/185
24	PEN2	8	531/768/539	691/918/696	–
		50	293/845/298	332/907/339	331/902/336
25	VARDIM	2	6/14/7	5/13/6	5/13/6
		50	27/69/31	37/83/39	30/74/33
		100	36/83/39	73/133/74	516/8406/524
26	TRIG	3	15/23/19	13/18/15	13/18/16
		50	44/48/45	42/43/43	42/43/43
		100	48/51/49	48/49/49	49/52/50
28	BV	3	6/14/7	6/14/7	6/14/7
		10	18/39/19	18/39/19	18/39/19
29	IE	3	7/11/8	7/11/8	7/11/8
		50	12/15/13	11/15/12	12/15/13
		100	12/15/13	11/15/12	12/15/13
		200	12/15/13	11/15/12	12/15/13
30	TRID	3	12/31/14	11/27/12	12/29/13
		50	63/340/64	65/334/66	67/335/68
		100	112/637/113	109/633/110	112/644/113
		200	216/1223/217	195/1153/196	195/1155/196
31	BAND	2	11/21/12	8/20/9	10/28/11
32	LIN	2	1/3/2	1/3/2	1/3/2
		50	1/3/2	1/3/2	1/3/2
		500	1/3/2	1/3/2	1/3/2
		1000	1/3/2	1/3/2	1/3/2
33	LIN1	2	2/10/3	2/10/3	2/10/3
		10	3/22/4	3/22/4	3/22/4
34	LIN2	4	2/11/3	2/11/3	2/11/3

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 not be dominated by a few problems for which the method requires a great deal of function evaluations and gradient functions. We can also compare methods WLQBFGS and BFGS by using the same rule.

From Table 2, we found that the average performance of the MBFGS method is a little better than the other two methods.

Table 2 Relative efficiency of BFGS, MBFGS and WLQBFGS algorithms

BFGS	MBFGS	WLQBFGS
1	0.9783	1.0413

Acknowledgments

The authors express thanks to Professor Zengxin Wei for suggesting to write this paper, and his support and supervision. we also thank the referees for their comments in improving our paper.

References

1 Broyden C G, Dennis J E, Moré J J. On the local and supelinear convergence of quasi-Newton methods. J Inst Math Appl, 1973, (12): 223~246.

2 Byrd R, Nocedal J. A tool for the analysis of quasi-Newton methods withapplication to unconstrained minimization. SIAM Journal on Numerical Analysis, 1989, (26): 727~739.

3 Byrd R, Nocedal J, Yuan Y. Global convergence of a class of quasi-Newton methods onconvex problems. SIAM Journal on Numerical Analysis, 1987, (24): 1171~1189.

4 Dai Y. Convergence properties of the BFGS algorithm, In: Appear in SIAM. Journal on Optimization, 2002, 13(3): 693~701.

5 Dennis J E, Moré J J. A characterization of superlinear convergence and its application to quasi-Newton methods. Math Comp, 1974, (28): 549~560.

6 Griewank A, Toint Ph L. Local convergence analysis for partitioned quasi-Newton updates. Numer Math, 1982, (39): 429~448.

7 Yuan Y, Sun W. Theory and Methods of Beijing Optimization, Beijing Science Press of China, 1999.

8 Li D, Fukushima M. A modified BFGS method and its global convergence in nonconvex minimization. Journal of Computational and Applied Mathematics, 2001, (129): 15~35.

9 Li D, Fukushima M. A global and superlinear convergent Gauss-Newton-based BFGS method for symmetric nonlinear equations. SIAM Journal on Numerical Analysis, 1999, (37): 152~172.

(下转第261页 Continue on page 261)

$$\cos 2(2E - 3A + A^2) =$$

$$\sin 1\begin{bmatrix}1&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix} + \sin 2\begin{bmatrix}0&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix} +$$

$$\cos 2\begin{bmatrix}0&0&0&0\\0&1&-1&-1\\0&1&-1&-1\\0&0&0&0\end{bmatrix} =$$

$$\begin{bmatrix}\sin 1&0&0&0\\0&\sin 2+\cos 2&-\cos 2&-\cos 2\\0&\cos 2&\sin 2-\cos 2&-\cos 2\\0&0&0&\sin 2\end{bmatrix}.$$

参考文献

1 陈启宗著.线性系统理论与设计 [美].王纪文等译.北京: 科学出版社, 1988. 29~ 45.

2 刘升德.根子空间的几何理论与演化矩阵的实际计算.阜新矿业学院学报, 1994, (4): 111~ 114.

3 张贤科, 许甫华.高等代数学.北京: 清华大学出版社, 1998. 172~ 173, 189~ 199.

4 陈大新.矩阵理论.上海: 上海交通大学出版社, 1991. 132~ 133, 142~ 143.

5 李德光.求 n 阶 m 次方幂的一个公式.湘潭矿业学院学报, 1994, (4): 80~ 81.

(责任编辑: 黎贞崇)

(上接第 25页 Continue from page 257)

10 Perry J M. A class of conjugate algorithms with a two step variable metric memory, Discussion paper 269, Center for Mathematical Studies in Economics and Management Science. Northwestern: Northwestern University, 1977.

11 Powell M J D. A new algorithm for unconstrained optimation, In: Nonlinear Programming, J B Rosen, O L Mangasarian et al eds. New York: Academic Press, 1970.

12 Shanno D F. On the convergence of a new conjugate gradient algorithm. SIAM Journal on Numerical Analysis, 1978, (15): 1247~ 1257.

13 Wei Z, Qi L, Chen X. An SQP-typemethod and its application in stochastic programming. Journal of Optimization Theory and Applications, 2003, (116): 205~ 228.

14 Wei Z, Li G, Qi L. New Quasi-Newton methods for unconstrained optimization problems. Nanning: Department of Mathematics and Information Science, Guangxi University, Guangxi, China, 2002.

15 Wei Z, Yu G, Yuan G et al.. The superlinear convergence of a modified BFGS-type method for uconstrained optimization, to appear in Computational Optimization and Applications.

16 Dai Y, Ni Q. Testing different conjugate gradient methods for large-scale unconstrained optimization. Journal of Comput Math, 2003, 21(3): 311~ 320.

17 Griewank A. On automatic differentiation, In: Mathematical Programming: Recent Developments and Applications, M Iri, K Tanabe, eds. Kluwer: Kluwer Academic Publishers, 1989, 84~ 108.

(责任编辑: 蒋汉明 黎贞崇)