PANM 15

Jan Vlček; Ladislav Lukšan

Limited-memory variable metric methods that use quantities from the preceding iteration

In: Jan Chleboun and Petr Přikryl and Karel Segeth and Jakub Šístek (eds.): Programs and Algorithms of Numerical Mathematics, Proceedings of Seminar. Dolní Maxov, June 6-11, 2010. Institute of Mathematics AS CR, Prague, 2010. pp. 227--232.

Persistent URL: http://dml.cz/dmlcz/702762

LIMITED-MEMORY VARIABLE METRIC METHODS THAT USE QUANTITIES FROM THE PRECEDING ITERATION*

Jan Vlček, Ladislav Lukšan

1. Introduction

In this contribution, a new family of globally convergent limited-memory (LM) variable metric (VM) line search methods for unconstrained minimization is presented. Numerical results indicate that the new methods can save computational time substantially for certain problems in comparison with the well-known L-BFGS method, see [3], [8].

VM or quasi-Newton line search methods, see [2], [4], start with an initial point $x_0 \in \mathcal{R}^N$ and generate iterations $x_{k+1} \in \mathcal{R}^N$ by the process $x_{k+1} = x_k + s_k$, $s_k = t_k d_k$, $k \ge 0$, where d_k is the direction vector and $t_k > 0$ is a stepsize.

It is assumed that the problem function $f : \mathcal{R}^N \to \mathcal{R}$ is differentiable and stepsize t_k is chosen in such a way that

$$f_{k+1} - f_k \le \varepsilon_1 t_k g_k^T d_k, \qquad g_{k+1}^T d_k \ge \varepsilon_2 g_k^T d_k, \tag{1}$$

$k \ge 0$, where $0 < \varepsilon_1 < 1/2$, $\varepsilon_1 < \varepsilon_2 < 1$, $f_k = f(x_k)$, $g_k = \nabla f(x_k)$ and $d_k = -H_k g_k$ with a symmetric positive definite matrix H_k; usually H_0 is a multiple of I and H_{k+1} is obtained from H_k by a rank-two VM update to satisfy the quasi-Newton condition $H_{k+1} y_k = s_k$ (see [2], [4]), where $y_k = g_{k+1} - g_k$, $k \ge 0$. For $i \ge 0$ we denote

$$b_i = s_i^T y_i, \qquad V_i = I - (1/b_i) s_i y_i^T$$

(note that $s_i^T y_i > 0$ for $g_i \ne 0$ by (1)). To simplify the notation we frequently omit index k and replace index $k+1$ by symbol $+$ and index $k-1$ by symbol $-$.

The L-BFGS method (LM variant of the well-known BFGS method, see [3], [8]) is based on the following quasi-product form of the BFGS update

$$H_+ = (1/b) s s^T + V H V^T. \tag{2}$$

The advantage of this form consists in the fact that only the last $\tilde{m}+1 = \min(k+1, m)$ couples $\{s_i, y_i\}_{i=k-\tilde{m}}^k$, where $m \ge 1$ is a given parameter, are stored to compute the direction vector $d_{k+1} = -H_{k+1} g_{k+1}$ by the Strang recurrences, see [8]. Matrices H_{k+1} are not computed, only defined by $H_{k+1} = H_{\tilde{m}+1}^{k+1}$, $k \ge 0$, where

$$H_0^{k+1} = (b_k/|y_k|^2) I, \tag{3}$$

$$H_{i+1}^{k+1} = (1/b_j) s_j\, s_j^T + V_j H_i^{k+1} V_j^T, \quad j = k - \tilde{m} + i, \quad 0 \le i \le \tilde{m}\,. \tag{4}$$

*This work was supported by the Grant Agency of the Czech Republic, project No. 201/09/1957, and the Institutional research plan No. AV0Z10300504.

Note that matrix H_k, which satisfies $d_k = -H_k g_k$, is different from matrix $H_{\tilde{m}}^{k+1}$ in the last update (4) in general; among others since matrix H_k is created by updating of matrix $H_0^k = (b_{k-1}/|y_{k-1}|^2)I$, not $H_0^{k+1} = (b_k/|y_k|^2)I$. Thus $H_{\tilde{m}}^{k+1} g_k \neq -d_k$ generally.

The Strang recurrences cannot be used directly for other updates from the Broyden class (see [2], [4]) than for the BFGS update (but another efficient approach is possible, see [6]). Some generalizations of the L-BFGS method are investigated in [9]. Here we focus on the approach which uses quantities from the preceding iteration.

Note that our methods do not belong to the Broyden class and has some common features with the multi-step quasi-Newton methods (see e.g. [7]).

We describe the new class of VM updates in Section 2 and the corresponding algorithm in Section 3; global convergence is treated in Section 4 and numerical results are reported in Section 5. Details and proofs of assertions can be found in [9].

2. The new class of methods

The Broyden class updates except for the BFGS update need calculate vector Hy in every iteration. This drawback can be eliminated by utilization of the quasi-Newton condition $Hy_- = s_-$. Although it is not satisfied in LM case, in this way we can construct efficient methods that use the same number of stored vectors and matrix by vector multiplications as the L-BFGS method, see Section 3.

Theorem 2.1. *Let matrix H be symmetric positive definite, $Hy_- = s_-$, $\sigma \in (-1, 1)$, $\bar{s} = s - \sigma\sqrt{b/b_-}\, s_-$, $\bar{y} = y - \sigma\sqrt{b/b_-}\, y_-$, $\bar{b} = \bar{s}^T y \neq 0$ and $\bar{\varrho} = (1 - \sigma^2)\, b/\bar{b}$. Then update H_+^{NB} with parameter σ given by*

$$H_+^{NB} = (\bar{\varrho}/\bar{b})\, \bar{s}\bar{s}^T + \bar{V} H \bar{V}^T, \qquad \bar{V} = I - (1/\bar{b})\, \bar{s}\bar{y}^T, \tag{5}$$

is positive definite and satisfies the quasi-Newton condition $H_+^{NB} y = s$ (for $\sigma = 0$ we obtain the BFGS update and assumption $Hy_- = s_-$ can be omitted). If $\sigma = s^T y_- / \sqrt{bb_-}$ then $\bar{s}^T y_- = 0$, $\bar{b} = \bar{s}^T \bar{y}$ and if also $\sigma \in (-1, 1)$ and $\bar{b} > 0$, then (5) represents the generalized BFGS update with nonquadratic correction parameter $\bar{\varrho}$ (see [4]), with vectors s and y replaced by $\bar{s}$, $\bar{y}$. If $\sigma = s_-^T y / \sqrt{bb_-}$ then $s_-^T \bar{y} = 0$ and $\bar{\varrho} = 1$.

Our numerical experiments indicate that convergence is significantly deteriorated when $|\sigma| \to 1$ and that all values σ satisfying $|\sigma| \leq 1/2$ with a suitable sign (Theorem 2.1 and Lemma 2.1 motivate us to use the sign of $s^T y_-$) give very good results.

Lemma 2.1. *Let $Hy_- = s_-$ and f be quadratic function $f(x) = \frac{1}{2}(x - x^*)^T G (x - x^*)$, $x^* \in \mathcal{R}^N$, with a symmetric positive definite matrix G. If vectors s, s_- are linearly independent and update H_+^{NB} of matrix H is given by (5) then choice $\sigma = s^T y_- / \sqrt{bb_-}$ (or equivalently $\sigma = s_-^T y / \sqrt{bb_-}$) satisfies $\bar{b} > 0$, $\sigma \in (-1, 1)$, $\bar{\varrho} = 1$ and $H_+^{NB} y_- = s_-$.*

Note that we need not calculate value $s^T y_-$. We use only the sign of $s^T y_-$, therefore in view of the following lemma we can utilize the value $s_-^T g$, computed during the line search procedure, in spite of the fact that assumption $d = -Hg$ is not appropriate to LM updates, see Section 1. In Section 3 we describe a choice of the sign of σ in details.

Lemma 2.2. *Let H be nonsingular matrix, $Hy_- = s_-$. If $d = -Hg$ then $s^T y_- = -t s_-^T g$.*

Taking into account Theorem 2.1 and Lemma 2.1, we will choose such parameter $\sigma \in (-1, 1)$ that corresponding $\bar{b}$ is positive and not too small in comparison with b in a sense that $\bar{b} \equiv b(1 - \sigma\, s_-^T y/\sqrt{bb_-}\,) \geq b(1 - \lambda)$, $\lambda \in (0, 1)$, which is equivalent to $\sigma\, s_-^T y \leq \lambda\sqrt{bb_-}$. The following lemma shows that in case that $\bar{b} < b(1 - \lambda)$ for some $\sigma \in (-1, 1)$, we can replace this σ by a more appropriate value.

Lemma 2.3. *Let $\sigma\, s_-^T y > \lambda\sqrt{bb_-}$ for some $\lambda \in (0, 1)$. Then $s_-^T y \neq 0$ and value $\hat{\sigma} = \lambda\sqrt{bb_-}/|s_-^T y| > 0$ satisfies $\pm\hat{\sigma}\, s_-^T y \leq \lambda\sqrt{bb_-}$ (for both signs) and $\hat{\sigma} < |\sigma|$.*

3. Implementation

Here we give the procedure based on Section 2. We define matrices H_0^{k+1} and $H_{k+1} = H_{\tilde{m}+1}^{k+1}$, $\tilde{m} = \min(k, m-1)$, $m \geq 1$, $k \geq 0$, by relations similar to (3), (4). Instead of matrices H_k, only $\tilde{m}+1 \leq m$ couples of vectors are stored here to compute the direction vector $d_{k+1} = -H_{k+1} g_{k+1}$, using the Strang recurrences, see [8], with a little modification - using transformed nonquadratic correction parameter $\bar{\varrho}$, see [4].

We choose the sign of σ in accordance with the sign of $-t s_-^T g \approx s^T y_-$, see Lemma 2.2 and Theorem 2.1. Since $s^T y_- = s_-^T y$ for f quadratic, see Lemma 2.1, we prefer the sign of $s_-^T y$ in case that $|t s_-^T g|$ is too small in comparison with $|s_-^T y|$ (constant 20 in Step 2 was found empirically). Using Lemma 2.3, we bound $|\sigma|$ to have $\bar{b}$ not too small, compared with b. For simplicity, we omit stopping criteria.

Algorithm 3.1

Data: The number m of VM updates per iteration, upper bound $\overline{\sigma} \in (0, 1)$ for $|\sigma_k|$, safeguard parameter $\lambda \in (0, 1)$ and line search parameters ε_1 and ε_2, $0 < \varepsilon_1 < \frac{1}{2}$, $\varepsilon_1 < \varepsilon_2 < 1$.

Step 0: *Initiation.* Choose the starting point $x_0 \in \mathcal{R}^N$, define direction vector $d_0 = -g_0$ and initiate iteration counter k to zero.

Step 1: *Line search.* Compute $x_{k+1} = x_k + t_k d_k$, where t_k satisfies (1), $g_{k+1} = \nabla f(x_{k+1})$, $y_k = g_{k+1} - g_k$ and b_k.

Step 2: *Update preparation.* If $|s_-^T y| > 20t|s_-^T g|$ set $\nu_k = \mathrm{sgn}(s_-^T y)$, otherwise set $\nu_k = -\mathrm{sgn}(s_-^T g)$. Choose parameter $\check{\sigma}_k \in [0, \overline{\sigma}]$ (for $k = 0$ we choose $\check{\sigma}_k = 0$) and set $\sigma_k = \nu_k\, \check{\sigma}_k$. If $\sigma_k\, s_-^T y > \lambda\sqrt{bb_-}$ set $\sigma_k = \lambda\nu_k\sqrt{bb_-}/|s_-^T y|$. Using Theorem 2.1, compute $\bar{b}_k$, $\bar{s}_k$ and $\bar{\varrho}_k$ and define $\bar{V}_k$.

Step 3: *Update definition.* Set $\tilde{m} = \min(k, m-1)$ and define $H_0^{k+1} = (b_k/|y_k|^2)\, I$ and $H_{k+1} \equiv H_{\tilde{m}+1}^{k+1}$ by

$$H_{i+1}^{k+1} = (\bar{\varrho}_j/\bar{b}_j)\bar{s}_j\, \bar{s}_j^T + \bar{V}_j H_i^{k+1} \bar{V}_j^T, \quad j = k - \tilde{m} + i, \quad 0 \leq i \leq \tilde{m}\,. \tag{6}$$

Step 4: *Direction vector.* Set $k := k+1$ and compute $d_k = -H_k g_k$ by the modified Strang recurrences, using the definition of matrices $\{H_i^k\}_{i=0}^{\min(k,m)}$, and go to Step 1.

4. Global convergence

Assumption 4.1. *The objective function* $f : \mathcal{R}^N \to \mathcal{R}$ *is bounded from below and uniformly convex with bounded second-order derivatives (i.e.* $0 < \underline{G} \leq \underline{\lambda}(G(x)) \leq \overline{\lambda}(G(x)) \leq \overline{G} < \infty$, $x \in \mathcal{R}^N$, *where* $\underline{\lambda}(G(x))$ *and* $\overline{\lambda}(G(x))$ *are the lowest and the greatest eigenvalues of the Hessian matrix* $G(x)$*).*

Since our new LM methods do not belong to the Broyden class, the usual approach must be modified. The following lemma before the main theorem plays basic role.

Lemma 4.1. *Let matrix* A *be symmetric positive definite,* $\vartheta > 0$, $\tau \neq 0$, $u \in \mathcal{R}^N$ *and* $v \in \mathcal{R}^N$*. Then update* A_+ *given by* $A_+ = \tau^2\vartheta\, uu^T + \left(I - \tau\, uv^T\right) A \left(I - \tau\, vu^T\right)$ *is positive definite and satisfies*

$$\mathrm{Tr}(A_+) \leq \tau^2\vartheta|u|^2 + \mathrm{Tr}(A)\Big(1 + |\tau|(|u||v|)\Big)^2, \tag{7}$$

$$\mathrm{Tr}(A_+^{-1}) \leq \mathrm{Tr}(A^{-1}) + |v|^2/\vartheta. \tag{8}$$

Theorem 4.1. *Let objective function* f *satisfy Assumption 4.1. Then Algorithm 3.1 generates a sequence* $\{g_k\}$ *that either terminates with* $g_k{=}0$ *for some* k *or* $\lim_{k\to\infty}|g_k|{=}0$.

5. Numerical results

First we demonstrate the influence of parameter σ on the number of evaluations and computational time, using the collection of sparse and partially separable test problems from [5] (Test 14, 22 problems) with $N = 1000$, $m = 10$, $\lambda = 1/2$ and the final precision $\|g(x^\star)\|_\infty \leq 10^{-6}$.

Results are given in Table 1, where 'NFE' is the total number of function and also gradient evaluations over all problems, 'Time' the total computational time in seconds and ϕ is the arithmetic mean of values 'NFE' and 'Time' over all σ.

σ	NFE	Time	σ	NFE	Time
0.0	22522	8.36	0.3	19854	7.42
0.033	22185	8.25	0.333	19865	7.36
0.067	21121	7.80	0.367	20068	7.49
0.1	20751	7.72	0.4	21359	7.81
0.133	20940	7.82	0.433	21250	7.82
0.167	20929	7.77	0.467	20779	7.71
0.2	20144	7.55	0.5	19754	7.28
0.233	20579	7.62	0.533	20207	7.39
0.267	22064	8.08	ϕ	20845	7.72
L-BFGS:	NFE = 22092	Time = 8.91			

Tab. 1: *Influence of parameter* σ *for Test 14.*

Problem	N	NFV	Percentage increase of NFV for $\sigma =$									
		L-BFGS	.05	.10	.15	.20	.25	.30	.35	.40	.45	.50
BDQRTIC	5000	248	-29	-10	-19	-43	-33	-16	-8	-18	5	-26
BROYDN7D	2000	3029	-1	-2	-3	-3	-3	-3	-2	0	2	6
CHAINWOO	1000	515	-8	-13	-19	-14	-18	-13	-20	-17	-15	-14
CURLY10	1000	5628	4	8	8	5	-5	2	-1	3	-7	-3
CURLY20	1000	6852	-6	-7	-6	-9	-9	-7	-10	-9	-7	-10
CURLY30	1000	7222	-3	-5	-5	-7	-10	-10	-5	-9	-13	-7
DIXMAANE	3000	249	-4	-3	-4	6	-4	-4	-10	2	-11	-10
DIXMAANF	3000	189	1	2	14	14	16	14	11	-2	-4	13
DIXMAANG	3000	188	11	17	9	5	10	6	13	6	6	-7
DIXMAANH	3000	185	7	12	15	10	10	7	-6	5	-4	5
DIXMAANI	3000	881	-9	-12	-17	-14	-27	-33	-40	-64	-77	-35
DIXMAANJ	3000	317	-3	-3	-4	-5	0	-9	-6	-16	17	20
DIXMAANK	3000	270	9	-5	-11	-7	7	4	16	7	37	28
DIXMAANL	3000	263	0	-8	-10	-3	-10	-13	-9	8	8	14
FLETCBV2	1000	944	28	1	-6	26	35	35	23	54	37	-4
FMINSRF2	5625	305	5	1	2	2	2	1	2	8	6	3
FMINSURF	5625	460	0	-2	4	13	-6	-18	-4	3	-3	-13
GENHUMPS	1000	2223	8	26	14	17	41	19	27	47	52	48
GENROSE	1000	2393	-2	-2	0	0	2	3	5	8	10	13
MOREBV	5000	116	3	3	-10	-7	-1	-3	-5	-2	0	5
MSQRTALS	529	3622	-22	-9	-22	3	-7	-10	-4	-12	-27	-12
NCB20	1010	497	3	33	28	7	48	10	-5	25	4	3
NCB20B	1000	1792	-5	-23	-5	-5	-8	-9	-9	-12	-9	-6
NONCVXU2	1000	3902	-11	-17	-4	4	-2	-13	-9	0	-16	-39
NONDQUAR	5000	4244	-17	3	1	3	-1	-11	3	13	-16	-10
POWER	500	110	-5	-7	-7	-5	-12	-13	-14	-13	-11	-13
QUARTC	5000	236	0	0	0	0	0	0	0	0	0	0
SINQUAD	5000	339	5	3	3	-3	10	0	1	11	-3	7
SPARSINE	1000	10680	-10	-8	-8	-4	-12	-9	-11	-15	-26	-19
SPMSRTLS	4999	224	1	0	-1	0	-5	-2	1	-2	-2	-3
VAREIGVL	500	168	-3	-4	-3	-10	-10	-15	-5	-8	-9	-11
All problems		58291	-5.6	-3.5	-3.8	-1.2	-4.0	-5.7	-4.2	-2.6	-10.2	-8.1

Tab. 2: *CUTE - Percentage increase of NFV against L-BFGS.*

For a better comparison with the L-BFGS method, we performed additional tests with problems from the widely used CUTE collection [1] with various dimensions N, $m = 10$, $\lambda = 1/2$ and the final precision $\|g(x^\star)\|_\infty \le 10^{-6}$. The percentage increase of NFV for various values of parameter σ against NFV for the L-BFGS (negative values indicate that our results are better than for the L-BFGS) is given in Table 2, where NFV is the number of function and also gradient evaluations. In the last line, the total values over all problems and their percentage increase are given.

Our limited numerical experiments indicate that the suitable choice of parameter σ can improve efficiency of limited-memory methods, substantially for some problems.

References

[1] Bongartz, I., Conn, A.R., Gould, N., and Toint, P.L.: CUTE: constrained and unconstrained testing environment. ACM Transactions on Mathematical Software **21** (1995), 123–160.

[2] Fletcher, R.: *Practical methods of optimization.* John Wiley & Sons, Chichester, 1987.

[3] Liu, D.C. and Nocedal, J.: On the limited memory BFGS method for large scale optimization. Math. Prog. **45** (1989), 503–528.

[4] Lukšan, L. and Spedicato, E.: Variable metric methods for unconstrained optimization and nonlinear least squares. J. Comput. Appl. Math. **124** (2000), 61–95.

[5] Lukšan, L. and Vlček, J.: Sparse and partially separable test problems for unconstrained and equality constrained optimization. Report V-767, ICS AS CR, Prague, 1998.

[6] Lukšan, L. and Vlček, J.: A recursive formulation of limited memory variable metric methods from the Broyden class. Report V-1059, ICS AS CR, Prague, 2009.

[7] Moughrabi, I.A.: New implicit multistep quasi-Newton methods. Numerical Analysis and Applications **2** (2009), 154–164.

[8] Nocedal, J.: Updating quasi-Newton matrices with limited storage. Math. Comp. **35** (1980), 773–782.

[9] Vlček, J. and Lukšan, L.: Generalizations of the limited-memory BFGS method based on quasi-product form of update. Report V-1060, ICS AS CR, Prague, 2009.