A recursive formulation of limited memory variable metric methods

Lukšan, Ladislav
2010

Lukšan, Ladislav ; Vlček, Jan. A recursive formulation of limited memory variable metric methods. In Seminar on Numerical Analysis. Modelling and Simulation of Challenging Engineering Problems. Prague : Institute of Computer Science AS CR, 2010, S. 107-109. ISBN 978-80-87136-07-2. [SNA '10. Seminar on Numerical Analysis, Nové Hrady, 18.01.2010-22.01.2010, CZ].

A recursive formulation of limited memory variable metric methods

L. Lukšan, J. Vlček

Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2, 182 07 Praha 8, and
Technical University of Liberec, Hálkova 6, 461 17 Liberec

Variable metric methods with limited memory can be efficiently used for large-scale unconstrained optimization in case the sparsity pattern of the Hessian matrix is not known. These methods are usually realized in the line-search framework so that they generate a sequence of points $x_i \in \mathcal{R}^n$, $i \in \mathcal{N}$, by the simple process

$$x_{i+1} = x_i + \alpha_i d_i, \tag{1}$$

where $d_i = -H_i g_i$ is a direction vector, H_i is a positive definite approximation of the inverse Hessian matrix and $\alpha_i > 0$ is a scalar step-size chosen in such a way that

$$F_{i+1} - F_i \le \varepsilon_1 \, \alpha_i \, d_i^T g_i, \quad d_i^T g_{i+1} \ge \varepsilon_2 \, d_i^T g_i \tag{2}$$

(the weak Wolfe conditions), where $F_i = F(x_i)$, $g_i = \nabla F(x_i)$ and $0 < \varepsilon_1 < 1/2$, $\varepsilon_1 < \varepsilon_2 < 1$. Matrices H_i, $i \in N$, are computed either by using a limited number ($m \ll n$) of variable metric updates applied to the scaled unit matrix or by updating low dimension matrices. The first approach, used in [9], is based on the computation of the direction vector d_i using the Strang recurrences [8]. The second approach, used in [1], is based on the matrix expression described below. To simplifying notation, we omit index i and replace index $i+1$ by $+$.

Variable metric method from the Broyden class use the update

$$\begin{aligned} H_+ &= H + UMU^T = H + [d, Hy] \begin{bmatrix} m_1, & m_2 \\ m_2, & m_3 \end{bmatrix} \begin{bmatrix} d \\ Hy \end{bmatrix} \\ &= H + \frac{1}{b} dd^T - \frac{1}{a} Hy(Hy)^T + \frac{\eta}{a} \left(\frac{a}{b} d - Hy \right) \left(\frac{a}{b} d - Hy \right)^T, \end{aligned} \tag{3}$$

where $d = x_+ - x$, $y = g_+ - g$, $a = y^T Hy$, $b = y^T d$ and η is a free parameter. We need to express m consecutive steps of (3) (with the initial matrix γI) in the form $H_+ = \gamma I + \bar{U}\bar{M}\bar{U}^T$, where $\bar{U} \in R^{n \times 2m}$ and $\bar{M} \in R^{2m \times 2m}$. In [1], the authors propose explicit expressions of the matrix $\bar{M}$ for three classic variable metric updates: DFP ($\eta = 0$), BFGS ($\eta = 1$) and the rank one ($\eta = b/(b-a)$). For other values of the parameter η, such explicit expressions are not known. In this contribution we describe another way, based on recursive construction of the matrix $\bar{M}$, which allows us to realize any member of the Broyden class of the variable metric updates. The following theorem is proved in [7].

Theorem 1 *Let H_+ be a matrix defined by (3) and $H = \gamma I + \bar{U}\bar{M}\bar{U}^T$. Then*

$$H_+ = H_1 + \bar{U}_+ \bar{M}_+ \bar{U}_+^T,$$

where $\bar{U}_+ = [\bar{U},\, d,\, H_1 y]$ and

$$\bar{M}_+ = \begin{bmatrix} \bar{M} + m_3 \, zz^T, & m_2 \, z, & m_3 \, z \\ m_2 \, z^T, & m_1, & m_2 \\ m_3 \, z^T, & m_2, & m_3 \end{bmatrix}. \tag{4}$$

Here $m_1 = (1/b)(\eta a/b + 1)$, $m_2 = -\eta/b$, $m_3 = (\eta - 1)/a$ are elements of matrix M, $z = \bar{M}\bar{r}$ and $\bar{r} = \bar{U}^T y$.

Lukšan, Ladislav ; Vlček, Jan. A recursive formulation of limited memory variable metric methods. In Seminar on Numerical Analysis. Modelling and Simulation of Challenging Engineering Problems. Prague : Institute of Computer Science AS CR, 2010, S. 107-109. ISBN 978-80-87136-07-2. [SNA '10. Seminar on Numerical Analysis, Nové Hrady, 18.01.2010-22.01.2010, CZ].

If $i \le m$, the construction of matrix H_{i+1} follows straightforwardly from Theorem 1. Thus we describe the construction of matrix H_{i+1} in case $i > m$. We will assume that $H_{i+1-m} = \gamma_i I$. At the beginning of the i-th iteration, we have available the rectangular matrix $\bar{U}_{i-1} = [d_{i-m}, y_{i-m}, \dots, d_{i-1}, y_{i-1}]$ and the block upper triangular matrix

$$\bar{R}_{i-1} = \begin{bmatrix} d_{i-m}^T y_{i-m}, & \dots & d_{i-m}^T y_{i-1} \\ y_{i-m}^T y_{i-m}, & \dots & y_{i-m}^T y_{i-1} \\ \dots\dots\dots & \dots & \dots\dots\dots \\ 0, & \dots & d_{i-1}^T y_{i-1} \\ 0, & \dots & y_{i-1}^T y_{i-1} \end{bmatrix},$$

whose every block contains two rows and one column. First we determine matrix $\bar{U}_i = [d_{i-m+1}, y_{i-m+1}, \dots, d_i, y_i]$ from matrix $\bar{U}_{i-1}$ by deleting the first two columns and adding the last two columns. Similarly easily we obtain matrix $\bar{R}_i$ from matrix $\bar{R}_{i-1}$. Only the last column $\bar{U}_i^T y_i$ of this matrix has to be computed. Furthermore, we compute recursively matrix $\bar{M}_i = \bar{M}_i^i$ in such a way that we set

$$\bar{M}_{i-m+1}^i = \begin{bmatrix} m_{i-m+1}^1, & m_{i-m+1}^2 \\ m_{i-m+1}^2, & m_{i-m+1}^3 \end{bmatrix}$$

(indices 1, 2, 3 are now placed up) and for $i-m+1 \le j \le i-1$, compute vector $z_j = \bar{M}_j^i \bar{r}_j$, where $\bar{r}_j$ is $j-i+m$-th column of matrix $\bar{R}_i$, whose every even element is multiplied by number γ_i (since $H_{i+1-m} = \gamma_i I$), and set

$$\bar{M}_{j+1}^i = \begin{bmatrix} \bar{M}_j^i + m_{j+1}^3 z_j z_j^T, & m_{j+1}^2 z_j, & m_{j+1}^3 z_j \\ m_{j+1}^2 z_j^T, & m_{j+1}^1, & m_{j+1}^2 \\ m_{j+1}^3 z_j^T, & m_{j+1}^2, & m_{j+1}^3 \end{bmatrix}.$$

Vector $H_{i+1} g_{i+1}$ is computed by the formula

$$\begin{aligned} H_{i+1} g_{i+1} = \gamma_i g_{i+1} \quad + \quad & [d_{i-m+1}, \gamma_i y_{i-m+1}, \dots, d_i, \gamma_i y_i] \, \bar{M}_i \\ & [d_{i-m+1}, \gamma_i y_{i-m+1}, \dots, d_i, \gamma_i y_i]^T g_{i+1} \end{aligned}$$

(even columns of matrix $\bar{U}_i$ are multiplied by number γ_i). As we can see, approximately $6mn$ operations (addition and multiplication) are consumed in i-th iteration. However, approximately $2(m-1)n$ operations can be saved, if we compute and store inner products $d_j^T g_{i+1}$, $y_j^T g_{i+1}$ instead of $d_j^T y_i$, $y_j^T y_i$, $i-m+1 \le j \le i$. Then the first $m-1$ inner products $d_j^T y_i$, $y_j^T y_i$, $i-m+1 \le j \le i-1$ can be determined from the previously computed inner products by the formulas $d_j^T y_i = d_j^T g_{i+1} - d_j^T g_i$, $y_j^T y_i = y_j^T g_{i+1} - y_j^T g_i$, $i-m+1 \le j \le i-1$. Thus it is necessary to compute only two inner products $d_i^T y_i$, $y_i^T y_i$. Inner products $d_j^T g_{i+1}$, $y_j^T g_{i+1}$, $i-m+1 \le j \le i$ can be used for the computation of direction vector s_{i+1}, so we save $2mn$ operations.

The method described has been tested by using a set of 60 test problems with 1000 variables. This set (Test25) was obtained by merging the sets Test14, Test15, Test18 described in [6], which can be downloaded from `http://www.cs.cas.cz/luksan/test.html` (together with report [6]). The results of the tests are listed in Table 1, where `NIT` is the total number of iterations, `NFV` is the total number of function and gradient evaluations, `NF` is the number of failures and `TIME` is the total CPU time. We have tested the original `LBFGS` subroutine, described in [2], and our realizations of limited memory variable metric methods implemented in the UFO system [5]. In Table 1, `BFGSSTR` denotes the limited memory BFGS method with the Strang recurrences [9] (an analogy of `LBFGS`), `BFGSBNS` denotes the limited memory BFGS method with compact matrices described in [1], `BFGSNEW` denotes the limited memory BFGS method with recursive construction of matrix $\bar{M}$ described above, LMVMNEW denotes the limited memory variable metric method with recursive construction of matrix $\bar{M}$ that use parameter η proposed in [4], and CG denotes the conjugate gradient method. Note that the first four rows in Table 1 correspond to different implementations of the BFGS method and that our approach gives the best results.

Method	NIT	NFV	F	TIME
LBFGS	110406	117226	2	43.38
BFGSSTR	99125	104085	-	37.56
BFGSBNS	91650	96235	-	36.89
BFGSNEW	85430	89796	-	33.50
LMVMNEW	92877	99033	-	34.61
CG	144990	222460	1	60.77

Table 1: Test results.

Acknowledgement: This work has been supported by the Czech science foundation, project No. 201/09/1957, and the institutional research plan No. AV0Z10300504.

References

[1] R.H.Byrd, J.Nocedal, R.B.Schnabel: *Representation of quasi-Newton matrices and their use in limited memory methods.* Math. Programming 63, 129-156, 1994.

[2] D.C.Liu, J.Nocedal: *On the limited memory BFGS method for large scale optimization.* Mathematical Programming 45, 503-528, 1989.

[3] L.Lukšan: *Numerické optimalizační metody.* Nepodmíněná minimalizace. Výzkumná zpráva V-1058, Ústav informatiky AV ČR, Praha 2009.

[4] L.Lukšan, E.Spedicato: *Variable metric methods for unconstrained optimization and nonlinear least squares.* Journal of Computational and Applied Mathematics 124, 61-93, 2000.

[5] L.Lukšan, M.Tůma, J.Vlček, N.Ramešová, M.Šiška, J.Hartman, C.Matonoha: *Interactive System for Universal Functional Optimization.* Research Report V-1040, Institute of Computer Science Czech Academy of Sciences, Prague 2008.

[6] L.Lukšan, J.Vlček: *Sparse and partially separable test problems for unconstrained and equality constrained optimization.* Research Report V-767, Institute of Computer Science, Czech Academy of Sciences, Prague 1998.

[7] L.Lukšan, J.Vlček: *Limited memory variable metric methods from the Broyden class.* Research Report V-1059, Institute of Computer Science Czech Academy of Sciences, Prague 2009.

[8] H.Matthies, G.Strang: *The solution of nonlinear finite element equations.* Int. J. for Numerical Methods in Engineering 14, 1613-1623, 1979.

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