Modifications of the limited-memory BNS method for better satisfaction of previous quasi-Newton conditions

Vlček, Jan
2011

Institute of Computer Science
Academy of Sciences of the Czech Republic

Modifications of the limited-memory BNS method for better satisfaction of previous quasi-Newton conditions.

Jan Vlček[a], Ladislav Lukšan[a,b]

[a]Institute of Computer Science, Academy of Sciences of the Czech Republic,
Pod vodárenskou věží 2, 182 07 Prague 8, Czech Republic and
[b]Technical University of Liberec, Hálkova 6, 461 17 Liberec, Czech Republic

Technical report No. V 1127

December 2011

Pod Vodárenskou věží 2, 182 07 Prague 8 phone: +420 266 052 083, fax: +420 286 585 789,
e-mail:luksan@cs.cas.cz, vlcek@cs.cas.cz

Institute of Computer Science
Academy of Sciences of the Czech Republic

Modifications of the limited-memory BNS method for better satisfaction of previous quasi-Newton conditions.[1]

Jan Vlček[a], Ladislav Lukšan[a,b]

[a]Institute of Computer Science, Academy of Sciences of the Czech Republic,
Pod vodárenskou věží 2, 182 07 Prague 8, Czech Republic and

[b]Technical University of Liberec, Hálkova 6, 461 17 Liberec, Czech Republic

Technical report No. V 1127

December 2011

Abstract:

Several modifications of the limited-memory variable metric BNS method for large scale unconstrained optimization are proposed, which consist in corrections (derived from the idea of conjugate directions) of the used difference vectors to improve satisfaction of previous quasi-Newton conditions, utilizing information from previous or subsequent iterations. In case of quadratic objective functions, conjugacy of all stored difference vectors and satisfaction of quasi-Newton conditions with these vectors is established. There are many possibilities how to realize this approach and although only two methods were implemented and tested, preliminary numerical results are promising.

Keywords:
Unconstrained minimization, variable metric methods, limited-memory methods, the BFGS update, conjugate directions, preliminary numerical results

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

1 Introduction

In this report we propose some modifications of the BNS method (see [2]) for large scale unconstrained optimization

$$\min f(x): \ x \in \mathcal{R}^N,$$

where it is assumed that the problem function $f: \mathcal{R}^N \to \mathcal{R}$ is differentiable.

Similarly as in the multi-step quasi-Newton methods (see e.g. [11]), we utilize information from previous (or also subsequent) iterations to correct the used difference vectors and change quasi-Newton conditions (see below) correspondingly. However, while the multi-step methods derive the corrections of the difference vectors from various interpolation methods, our approach is based on the idea of conjugate directions (see e.g. [4], [13]). Note that some of these thoughts are presented in our report [14].

The BNS method belongs to the variable metric (VM) or quasi-Newton line search iterative methods, see [4], [9]. They 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, usually chosen in such a way that

$$f_{k+1} - f_k \le \varepsilon_1 t_k g_k^T d_k, \qquad g_{k+1}^T d_k \ge \varepsilon_2 g_k^T d_k, \tag{1.1}$$

$k \ge 0$, 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 VM update to satisfy quasi-Newton condition (QNC)

$$H_{k+1} y_k = s_k \tag{1.2}$$

(see [4], [9]), where $y_k = g_{k+1} - g_k$, $k \ge 0$. For $k \ge 0$ we denote

$$b_k = s_k^T y_k$$

(note that $b_k > 0$ for $g_k \ne 0$ by (1.1)). To simplify the notation we frequently omit index k and replace index $k+1$ by symbol $+$ and index $k-1$ by symbol $-$.

Among VM methods, the BFGS method, see [4], [9], [13], belongs to the most efficient; the BNS and L-BFGS (see [6], [12]) methods represent its well-known limited-memory adaptations. In every iteration, we recurrently update matrix $\zeta_k I$, $\zeta_k > 0$, by the BFGS method, using $\tilde{m}+1$ couples of vectors $(s_{k-\tilde{m}}, y_{k-\tilde{m}}), \dots, (s_k, y_k)$, where

$$\tilde{m} = \min(k, m-1) \tag{1.3}$$

and $m \ge 1$ is a given parameter. Matrix H_+ can be explicitly expressed either in the form, see [2],

$$H_+ = \zeta I + \begin{bmatrix} S, \zeta Y \end{bmatrix} \begin{bmatrix} U^{-T}(D + \zeta Y^T Y) U^{-1} & -U^{-T} \\ -U^{-1} & 0 \end{bmatrix} \begin{bmatrix} S^T \\ \zeta Y^T \end{bmatrix},$$

or in the form, which can be also found in [2],

$$H_+ = S U^{-T} D U^{-1} S^T + \zeta (I - S U^{-T} Y^T)(I - Y U^{-1} S^T), \tag{1.4}$$

where $S_k = [s_{k-\tilde{m}}, \ldots, s_k]$, $Y_k = [y_{k-\tilde{m}}, \ldots, y_k]$, $D_k = \mathrm{diag}[b_{k-\tilde{m}}, \ldots, b_k]$ and $(U_k)_{i,j} = (S_k^T Y_k)_{i,j}$ for $i \le j$, $(U_k)_{i,j} = 0$ otherwise (an upper triangular matrix), $k \ge 0$. In both of the methods, direction vectors can be efficiently calculated, without computing of matrix H_+; e.g. from (1.4) we get

$$H_+ g_+ = \zeta g_+ + S\Big[U^{-T}\Big((D + \zeta Y^T Y)U^{-1}S^T g_+ - \zeta Y^T g_+\Big)\Big] - Y\Big[\zeta U^{-1}S^T g_+\Big], \tag{1.5}$$

where in brackets we have low-order matrices.

In this report we will investigate the more general form of update formula

$$H_+ = SMS^T + \zeta(I - SK^T Y^T)(I - YKS^T), \tag{1.6}$$

here K, M are $(\tilde{m}+1) \times (\tilde{m}+1)$ matrices. From (1.6) we obtain

$$H_+ Y = S(MS^T Y) + \zeta(Y - SK^T Y^T Y)(I - KS^T Y). \tag{1.7}$$

We can see that if the last column of matrices $KS^TY - I$, $MS^TY - I$ is null, then the last QNC (1.2) is satisfied. For $K = M = (S^TY)^{-1}$ we get update formula

$$H_+ = S(S^TY)^{-1}S^T + \zeta\left(I - S(S^TY)^{-T}Y^T\right)\left(I - Y(S^TY)^{-1}S^T\right), \tag{1.8}$$

which satisfies $H_+ Y = S$ by (1.7), i.e. QNC with all stored difference vectors are satisfied. Although this simple choice does not guarantee that the corresponding direction vector is descent, our new methods are based on this idea. Note that a similar approach is also used in the method, described in [5], which however needs a higher in order number of arithmetic operations than the BNS method.

From many possible variants, only two methods were implemented and tested. In Section 2 we derive the first method, which attempts to approximate matrix S^TY by RL with suitable matrices R, L, where R is upper triangular and L lower triangular. Matrices R^{-T}, L^{-1} can be also understood as matrices of transformations $S \to \bar{S} = SR^{-T}$, $Y \to \bar{Y} = YL^{-1}$. These transformations use subsequent difference vectors, which means that each column of $\bar{S}$, $\bar{Y}$ is expressed by means of columns of S, Y with greater or equal indices. If $S^TY = RL$, we obviously obtain $\bar{S}^T\bar{Y} = I$.

The second method presented in Section 3 attempts to approximate matrix S^TY by $L^{-1}\tilde{D}R^{-1}$, where matrix R is upper triangular, L lower triangular and $\tilde{D}$ diagonal (matrices L, R have here another meaning than at the first method), therefore corresponding transformations $S \to \tilde{S} = SL^T$, $Y \to \tilde{Y} = YR$ use previous difference vectors, which means that each column of $\tilde{S}$, $\tilde{Y}$ is expressed by means of columns of S, Y with smaller or equal indices. If $S^TY = L^{-1}\tilde{D}R^{-1}$, we obtain $\tilde{S}^T\tilde{Y} = \tilde{D}$. This method generalizes the method in [14] in some sense and has also similar theoretical properties.

Numerical experiments are reported in Section 4.

2 Method that use subsequent difference vectors

In this section we propose a recursive process to construct transformation matrices L, R with suitable properties, where R is upper triangular and L lower triangular, which approximate solution to problem $\bar{S}^T\bar{Y} = R^{-1}(S^TY)L^{-1} = I$.

In Section 2.1, a basic recursive step of the process is derived and some its properties are given. In Section 2.2 we show how it can be used to obtain new VM updates. In Section 2.3 we present a simple modification of the BNS method, which can be used for transformed matrices $\bar{S}$, $\bar{Y}$ to another efficiency improvement in case that matrix $\bar{S}^T\bar{Y}$ is near to I. In Section 2.4, the situation when the recursive step cannot be performed or has unsuitable properties is discussed. Finally, in Section 2.5 we describe a selected implementation of the first testing method in detail.

2.1 Nonsymmetric decomposition

First we show how to replace problem to solve equation $R^{-1}(S^TY)L^{-1} = I$ by a similar problem with a smaller dimension. Let $S = [\underline{S}, s]$, $Y = [\underline{Y}, y]$, $L = \left[\begin{smallmatrix} \underline{L} & 0 \\ l^T & \sqrt{b} \end{smallmatrix}\right]$, $R = \left[\begin{smallmatrix} \underline{R} & r \\ 0^T & \sqrt{b} \end{smallmatrix}\right]$. Then we can write

$$\begin{aligned} R^{-1}(S^TY)L^{-1} &= \begin{bmatrix} \underline{R}^{-1} & -\underline{R}^{-1}r/\sqrt{b} \\ 0^T & 1/\sqrt{b} \end{bmatrix} \begin{bmatrix} \underline{S}^T\underline{Y} & \underline{S}^Ty \\ s^T\underline{Y} & b \end{bmatrix} \begin{bmatrix} \underline{L}^{-1} & 0 \\ -l^T\underline{L}^{-1}/\sqrt{b} & 1/\sqrt{b} \end{bmatrix} \\ &= \begin{bmatrix} \underline{R}^{-1}\Big(\underline{S}^T\underline{Y} - \underline{S}^Tyl^T/\sqrt{b} - rs^T\underline{Y}/\sqrt{b} + rl^T\Big)\underline{L}^{-1} & \underline{R}^{-1}\Big(\underline{S}^Ty/\sqrt{b} - r\Big) \\ \Big(s^T\underline{Y}/\sqrt{b} - l^T\Big)\underline{L}^{-1} & 1 \end{bmatrix}. \end{aligned}$$

From this we can derive the recursive step of the decomposition. If we set

$$\Delta_l = \underline{Y}^Ts/\sqrt{b} - l, \qquad \Delta_r = \underline{S}^Ty/\sqrt{b} - r, \tag{2.1}$$

we obtain

$$R^{-1}(S^TY)L^{-1} = \begin{bmatrix} \underline{R}^{-1}\Big(\underline{S}^T\underline{Y} - \underline{S}^Tys^T\underline{Y}/b + \Delta_r\Delta_l^T\Big)\underline{L}^{-1} & \underline{R}^{-1}\Delta_r \\ \Delta_l^T\underline{L}^{-1} & 1 \end{bmatrix}. \tag{2.2}$$

For the choice

$$l = \underline{Y}^Ts/\sqrt{b}, \qquad r = \underline{S}^Ty/\sqrt{b} \tag{2.3}$$

we get

$$R^{-1}(S^TY)L^{-1} = \begin{bmatrix} \underline{R}^{-1}(\underline{S}^T\underline{Y} - \underline{S}^Tys^T\underline{Y}/b)\underline{L}^{-1} & 0 \\ 0^T & 1 \end{bmatrix}. \tag{2.4}$$

Therefore, we converted the problem to find nonsingular matrices R, L satisfying $R^{-1}(S^TY)L^{-1} = I$ to the problem to find nonsingular matrices $\underline{R}$, $\underline{L}$ satisfying $\underline{R}^{-1}(\underline{S}^T\underline{Y} - \underline{S}^Tys^T\underline{Y}/b)\underline{L}^{-1} = I$. If matrix $\underline{S}^T\underline{Y} - \underline{S}^Tys^T\underline{Y}/b$ has the last diagonal element positive, we can repeat this recursive step, etc.

We show that in case of symmetric positive definite matrix S^TY, e.g. if objective function f is strictly convex quadratic (see Lemma 2.2), this recursive process is well defined.

Lemma 2.1. *Let matrix V be symmetric positive definite, $V = \left[\begin{smallmatrix} \underline{V} & v \\ v^T & \nu \end{smallmatrix}\right]$. Then also matrix $\underline{V} - vv^T/\nu$ is symmetric positive definite.*

Proof. Since $w^TVw > 0$ for any nonzero w of corresponding dimension, setting $w = [u^T, \mu]^T$, $\mu = -u^Tv/\nu$, we obtain

$$w^TVw = u^T\underline{V}\,u + 2\mu\, u^Tv + \mu^2\nu = u^T\underline{V}\,u - (u^Tv)^2/\nu = u^T(\underline{V} - vv^T/\nu)u > 0$$

for any nonzero u, thus matrix $\underline{V} - vv^T/\nu$ is symmetric positive definite. □

Lemma 2.2. *Let 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 and suppose that columns of matrix S are linearly independent. Then matrix $S^T Y$ is symmetric positive definite, the decomposition process described above is well defined, $S^T Y = RL$ and $L^T = R$.*

Proof. This immediately follows from $S^T Y = S^T G S$ and Lemma 2.1. □

2.2 VM updates derived from the complete decomposition

We suppose here that the complete decomposition of $S^T Y$ was performed and that we have matrices R, L satisfying $\bar{S}^T \bar{Y} = R^{-1}(S^T Y)L^{-1} = I$, or $S^T Y = RL$. From (1.8) we get nonsymmetric update formula

$$H_+^{(1)} = SL^{-1}R^{-1}S^T + \zeta(I - SR^{-T}L^{-T}Y^T)(I - YL^{-1}R^{-1}S^T)\,, \tag{2.5}$$

which satisfies $H_+^{(1)} Y = S$. Relation $R^{-1}(S^T Y)L^{-1} = I$ can also be rewritten as $\bar{S}^T \bar{Y} = I$, where $\bar{S} = SR^{-T}$, $\bar{Y} = YL^{-1}$. Considering (1.8) with transformed matrices $\bar{S}$, $\bar{Y}$ instead of S, Y and supposing that $\bar{S}^T \bar{Y} = I$, we obtain symmetric update formula

$$\bar{H}_+ = \bar{S}\bar{S}^T + \zeta(I - \bar{S}\bar{Y}^T)(I - \bar{Y}\bar{S}^T)\,, \tag{2.6}$$

which satisfies $\bar{H}_+ \bar{Y} = \bar{S}$ by $\bar{S}^T \bar{Y} = I$; it together means that columns of $\bar{S}$ are conjugate with respect to $\bar{H}_+^{-1}$. Setting $\bar{S} = SR^{-T}$, $\bar{Y} = YL^{-1}$ into (2.6), we have

$$H_+^{(2)} = SR^{-T}R^{-1}S^T + \zeta(I - SR^{-T}L^{-T}Y^T)(I - YL^{-1}R^{-1}S^T)\,. \tag{2.7}$$

In comparison to $H_+^{(1)}$, matrix $H_+^{(2)}$ is always symmetric positive definite.

Lemma 2.3. *Matrix $H_+^{(2)}$ (if well defined) is always symmetric positive definite.*

Proof. Let $q \in \mathcal{R}^N$, $q \neq 0$. If $\bar{S}^T q \neq 0$, then $q^T H_+^{(2)} q \geq |\bar{S}^T q|^2 > 0$ by (2.6), otherwise $q^T H_+^{(2)} q = \zeta q^T q > 0$. □

Theorem 2.1. *Let the assumptions of Lemma 2.2 be satisfied, $\bar{S} = SR^{-T}$ and $\bar{Y} = YL^{-1}$. Then $\bar{S}^T G \bar{S} = I$ (conjugacy of columns of $\bar{S}$), matrices $H_+^{(i)}$ are well defined and symmetric positive definite and $H_+^{(i)} Y = S$, $H_+^{(i)} \bar{Y} = \bar{S}$ hold, $i = 1, 2$.*

Proof. From Lemma 2.2 we have $S^T Y = RL$, which yields $I = \bar{S}^T \bar{Y} = \bar{S}^T G \bar{S}$, and $L^T = R$, thus $H_+^{(1)} = H_+^{(2)}$. Since $H_+^{(1)} Y = S$, $H_+^{(2)} \bar{Y} = \bar{S}$, it suffices to use Lemma 2.3. □

Note that none of these results requires exact line searches and that formulas (2.5) and (2.7) can be also used in situations when the decomposion mentioned above can be performed only approximately, see Section 2.4.

2.3 A simple modification of the BNS method

Setting $\hat{R} = UD^{-1/2}$, $\hat{L} = D^{1/2}$, we can express the BNS update formula (1.4) in the form similar to (2.7)

$$H_+^{BNS} = S\hat{R}^{-T}\hat{R}^{-1}S^T + \zeta(I - S\hat{R}^{-T}\hat{L}^{-T}Y^T)(I - Y\hat{L}^{-1}\hat{R}^{-1}S^T)\,. \tag{2.8}$$

Since we can write $D\!=\operatorname{diag}[\underline{D},b]$, $U\!=\!\left[\begin{smallmatrix}\underline{U} & \underline{S}^Ty\\ 0^T & b\end{smallmatrix}\right]$, we have

$$\hat{R} = \left[\begin{array}{cc}\underline{U} & \underline{S}^Ty\\ 0^T & b\end{array}\right]\left[\begin{array}{cc}\underline{D}^{-1/2} & 0\\ 0^T & 1/\sqrt{b}\end{array}\right] = \left[\begin{array}{cc}\underline{U}\,\underline{D}^{-1/2} & \underline{S}^Ty/\sqrt{b}\\ 0^T & \sqrt{b}\end{array}\right]. \tag{2.9}$$

Writing $\hat{L} = \left[\begin{smallmatrix}\underline{\hat{L}} & 0\\ \hat{l}^T & \sqrt{b}\end{smallmatrix}\right]$, $\hat{R} = \left[\begin{smallmatrix}\underline{\hat{R}} & \hat{r}\\ 0^T & \sqrt{b}\end{smallmatrix}\right]$, we have $\hat{l}\!=0$ and $\hat{r} = \underline{S}^Ty/\sqrt{b}$ by (2.9). Using (2.2) with $\hat{R}$, $\hat{L}$ instead of R, L, in view of $\hat{l}\!=\Delta_{\hat{r}}\!=0$ we obtain

$$\hat{R}^{-1}(S^TY)\hat{L}^{-1}\!=\!\left[\begin{array}{cc}\underline{D}^{1/2}\underline{U}^{-1}(\underline{S}^T\underline{Y}-\underline{S}^Tys^T\underline{Y}/b)\underline{D}^{-1/2} & 0\\ s^T\underline{Y}\,\underline{D}^{-1/2}/\sqrt{b} & 1\end{array}\right]. \tag{2.10}$$

If we replace matrix $\hat{L}$ by $\check{L} = D^{-1/2}V = \left[\begin{smallmatrix}\underline{\check{L}} & 0\\ \check{l}^T & \sqrt{b}\end{smallmatrix}\right]$, where $V = S^TY-(U-D)\!=\!\left[\begin{smallmatrix}\underline{V} & 0\\ s^T\underline{Y} & b\end{smallmatrix}\right]$, we get

$$\check{L} = \left[\begin{array}{cc}\underline{D}^{-1/2} & 0\\ 0^T & 1/\sqrt{b}\end{array}\right]\left[\begin{array}{cc}\underline{V} & 0\\ s^T\underline{Y} & b\end{array}\right] = \left[\begin{array}{cc}\underline{D}^{-1/2}\underline{V} & 0\\ s^T\underline{Y}/\sqrt{b} & \sqrt{b}\end{array}\right], \tag{2.11}$$

i.e. $\check{l} = \underline{Y}^Ts/\sqrt{b}$, which together with $\hat{r} = \underline{S}^Ty/\sqrt{b}$ implies

$$\hat{R}^{-1}(S^TY)\check{L}^{-1}\!=\!\left[\begin{array}{cc}\underline{D}^{1/2}\underline{U}^{-1}(\underline{S}^T\underline{Y}-\underline{S}^Tys^T\underline{Y}/b)\underline{V}^{-1/2}\underline{D}^{1/2} & 0\\ 0^T & 1\end{array}\right] \tag{2.12}$$

by (2.2) with $\hat{R}$, $\check{L}$ instead of R, L (i.e. with $\Delta_{\hat{r}} = \Delta_{\check{l}} = 0$). We see that the replacement of $\hat{L}$ by $\check{L}$ causes that not only the last column, but also the last row of $\hat{R}^{-1}(S^TY)\check{L}^{-1}-I$ are null. These considerations lead us to formula

$$\begin{aligned} H_+^{MBNS} &= S\hat{R}^{-T}\hat{R}^{-1}S^T+\zeta(I-S\hat{R}^{-T}\check{L}^{-T}Y^T)(I-Y\check{L}^{-1}\hat{R}^{-1}S^T)\\ &= SU^{-T}DU^{-1}S^T+\zeta(I-SU^{-T}DV^{-T}Y^T)(I-YV^{-1}DU^{-1}S^T), \end{aligned} \tag{2.13}$$

i.e. (2.7) with $L = \check{L}$, $R = \hat{R}$. Note that matrix H_+^{MBNS} is always symmetric positive definite by Lemma 2.3, that the choice $\check{l} = \underline{Y}^Ts/\sqrt{b}$ and $\hat{r} = \underline{S}^Ty/\sqrt{b}$ is in fact the choice (2.3) in the first step of decomposition (see Section 2.1) and that formula (2.13) can be more efficient then the BNS method in case that matrix S^TY is near to I; this situation can occur for transformed matrices when we get a good approximation of the complete decomposition mentioned in Section 2.2.

2.4 Approximate decomposition

If the last diagonal element of matrix $J = \underline{S}^T\underline{Y}-\underline{S}^Tys^T\underline{Y}/b$ in (2.4) is not positive and the dimension of J is greater than one, the recursive step should not be performed, since we could not divide in (2.3) by $\sqrt{J_{\tilde{m},\tilde{m}}}$ to calculate vectors l, r in the next step. This drawback can be eliminated by modifications of transformation matrices R, L or by suitable corrections before or after construction of matrix J. Note that we consider here only the first recursive step, but it can be easily generalized to other steps.

Besides, $H_+\bar{Y} = \bar{S}$ implies $H_+Y = S(R^{-T}L)$ by $\bar{S} = SR^{-T}$ and $\bar{Y} = YL^{-1}$. Therefore QNC with transformed matrices $\bar{S}$, $\bar{Y}$ can be a good substitute for QNC with S, Y only in case that $R^{-T}L \approx I$, i.e. $L \approx R^T$. Thus we should not modify matrices R,

L or suitably correct matrix J, whenever the corresponding elements of vectors $\underline{S}^T y$, $\underline{Y}^T s$ are too different.

Modifications of transformation matrices R, L can be realized in the following way - we first set $l = \underline{Y}^T s/\sqrt{b}$, $r = \underline{S}^T y/\sqrt{b}$ and then replace some elements of l, r by zero if corresponding elements of vectors $\underline{S}^T y$, $\underline{Y}^T s$ are too different or if relevant diagonal element of matrix J would be negative or too small. Subsequently, we define vectors Δ_r, Δ_l by (2.1) with these new vectors l, r. To obtain matrix $\underline{S}^T\underline{Y} - \underline{S}^T y s^T \underline{Y}/b + \Delta_r \Delta_l^T = J + \Delta_r \Delta_l^T$ in (2.2), we then modify matrix J in such a way that we simply replace elements of $J + \Delta_r \Delta_l^T$ corresponding to zero elements of both l and r by relevant elements (with the same indices) of matrix $\underline{S}^T\underline{Y}$.

We also got good results with various corrections of matrix J, e.g. increasing of some diagonal elements of $\underline{S}^T\underline{Y}$, but the best results were obtained with the following simple corrections - before each step of decomposition, we replace problematic elements of vector $\underline{Y}^T s$ by zero, while vector $\underline{S}^T y$ is left unchanged.

Finally, a side effect of transformations is deterioration of stability; thus sometimes, if a contribution of transformation would be too small, it is better to omit a corresponding part of the decomposition, see Section 2.5 for details.

After the whole approximate decomposition, we can use transformation matrices R, L and calculate matrix H_+, using formula (2.5) or (2.7). We can also utilize the BNS formula (1.4) or its variant (2.13) with transformed matrices $\bar{S} = SR^{-T}$, $\bar{Y} = YR^{-1}$, $\bar{D}$, $\bar{U}$, $\bar{V}$ instead of S, Y, D, U, V, where $\bar{D}_{i,j} = (\bar{S}^T\bar{Y})_{i,j}$ for $i = j$, $\bar{D}_{i,j} = 0$ otherwise (the diagonal of $\bar{S}^T\bar{Y}$), $\bar{U}_{i,j} = (\bar{S}^T\bar{Y})_{i,j}$ for $i \le j$, $\bar{U}_{i,j} = 0$ otherwise and $\bar{V} = \bar{S}^T\bar{Y} - (\bar{U} - \bar{D})$. E.g. update formula (1.4) with transformed matrices has the form

$$\begin{aligned} H_+^{(3)} &= \bar{S}\bar{U}^{-T}\bar{D}\bar{U}^{-1}\bar{S}^T + \zeta(I - \bar{S}\bar{U}^{-T}\bar{Y}^T)(I - \bar{Y}\bar{U}^{-1}\bar{S}^T) \\ &= SR^{-T}\bar{U}^{-T}\bar{D}\bar{U}^{-1}R^{-1}S^T + \zeta(I - SR^{-T}\bar{U}^{-T}L^{-T}Y^T)(I - YL^{-1}\bar{U}^{-1}R^{-1}S^T). \end{aligned} \tag{2.14}$$

We tested all these update methods and the results were comparable.

2.5 Implementation

In this section we describe only the first selected testing method. First we give a procedure for updating of basic low-order matrices S^TY, Y^TY, similar to the algorithm given in [2] for updating of matrices D, U, Y^TY used in (1.5). Note that step (*ii*) requires mn extra multiplications (to compute vector Y^Ts), compared with the corresponding algorithm in [2] and that the number of operations can be reduced, but it is not implemented in this testing version of the method).

Procedure 2.1 *(Matrix Updating)*

Given: s, y, g_+, matrices S_-, Y_-, $S_-^TY_-$, $Y_-^TY_-$ and vectors S_-^Tg, Y_-^Tg.

(i): Set $S = [S_-, s]$, $Y = [Y_-, y]$.

(ii): Compute $S^Tg_+ = [S_-^Tg_+, s^Tg_+]$, $Y^Tg_+ = [Y_-^Tg_+, y^Tg_+]$, $Y^Ts = [Y_-^Ts, y^Ts]$, y^Ty.

(iii): Compute $S_-^Ty = S_-^Tg_+ - S_-^Tg$, $Y_-^Ty = Y_-^Tg_+ - Y_-^Tg$.

(iv): Set $S^TY = \begin{bmatrix} S_-^TY_- & S_-^Ty \\ s^TY_- & s^Ty \end{bmatrix}$, $Y^TY = \begin{bmatrix} Y_-^TY_- & Y_-^Ty \\ y^TY_- & y^Ty \end{bmatrix}$ and return.

To decompose matrix S^TY with simple corrections, mentioned in Section 2.4, we use the following procedure. Conditions in steps *(ii)* and *(iii)* were found empirically. Note that condition $\sigma_{in}\sigma_{ni} > 0.99999\sigma_{ii}\sigma_{nn}$ in step *(iii)*, which guarantees that new diagonal elements of matrix $[\sigma_{ij}]_{\substack{j=1,\dots,n-1\\ i=1,\dots,n-1}}$ in step *(v)* will not be too small, can be written in the form $\sigma_{ii} - \sigma_{in}\sigma_{ni}/\sigma_{nn} < 10^{-5}\sigma_{ii}$.

Procedure 2.2 *(Decomposition)*

Given: A dimension $\bar{n}$ of matrix S^TY.

(i): Set $i_D = 1$ (decomposition indicator), $n = \bar{n}$ and $\sigma_{ij} = (S^TY)_{i,j}$, $1 \le i \le n$, $1 \le j \le n$.

(ii): Set $\Delta = \max_{i=1,\dots,n-1} |\sigma_{in} - \sigma_{ni}|/\sqrt{\sigma_{ii}\sigma_{nn}}$. If $\Delta < 10^{-7}$ set $i_D = 0$ and go to *(vi)*. Set $i = 1$.

(iii): If $\sigma_{in}\sigma_{ni} < 0$ or $\sigma_{in}\sigma_{ni} > 0.99999\sigma_{ii}\sigma_{nn}$ or $|\sigma_{in} - \sigma_{ni}| > 0.02(\sigma_{ii} + \sigma_{nn})$ or $\sigma_{ii} < 0.01 s_i^T y_i$ then set $\sigma_{ni} = 0$.

(iv): Set $i := i + 1$. If $i < n$ go to *(iii)*.

(v): Set $[\sigma_{ij}]_{\substack{j=1,\dots,n-1\\ i=1,\dots,n-1}} := [\sigma_{ij}]_{\substack{j=1,\dots,n-1\\ i=1,\dots,n-1}} - [\sigma_{n1}, \dots, \sigma_{n,n-1}][\sigma_{1n}, \dots, \sigma_{n-1,n}]^T/\sigma_{nn}$ and then $n := n - 1$. If $n > 1$ goto *(ii)*.

(vi): Set $L_{i,j} = \sigma_{ij}/\sqrt{\sigma_{nn}}$ for $1 \le j \le i \le n$, $R_{i,j} = \sigma_{ij}/\sqrt{\sigma_{nn}}$ for $1 \le i \le j \le n$, $L_{i,j} = R_{i,j} = 0$ otherwise. If $i_D = 0$ set $L_{i,j} = 0$ for $1 \le j < i \le n$. Return.

We now state the method in details. To compute the direction vector in Step 5, we use formula (2.5) or (2.7) and express H_+g_+ similarly as in (1.5):

$$H_+^{(1)}g_+ = \zeta g_+ + S\Big[(L^{-1} + \zeta R^{-T}E)R^{-1}S^Tg_+ - \zeta R^{-T}L^{-T}Y^Tg_+)\Big] - Y\Big[\zeta L^{-1}R^{-1}S^Tg_+\Big], \quad (2.15)$$

$$H_+^{(2)}g_+ = \zeta g_+ + S\Big[R^{-T}\Big((I + \zeta E)R^{-1}S^Tg_+ - \zeta L^{-T}Y^Tg_+\Big)\Big] - Y\Big[\zeta L^{-1}R^{-1}S^Tg_+\Big], \quad (2.16)$$

where $E = L^{-T}(Y^TY)L^{-1}$ and in brackets we have low-order matrices. For simplicity, we omit stopping criteria and contingent restarts when the direction vector is not descent.

Algorithm 2.3

Data: A number $m \ge 1$ of VM updates per iteration and line search parameters ε_1, ε_2, $0 < \varepsilon_1 < 1/2$, $\varepsilon_1 < \varepsilon_2 < 1$.

Step 0: Initiation. Choose starting point $x_0 \in \mathcal{R}^N$, define starting matrix $H_0 = I$ and direction vector $d_0 = -g_0$ and initiate $i_D = 1$ and iteration counter k to zero.

Step 1: Line search. Compute $x_{k+1} = x_k + t_k d_k$, where t_k satisfies (1.1), $g_{k+1} = \nabla f(x_{k+1})$, $s_k = t_k d_k$, $y_k = g_{k+1} - g_k$ and $\zeta_k = s_k^T y_k / y_k^T y_k$. If $k = 0$ set $S_k = [s_k]$, $Y_k = [y_k]$, $S_k^T Y_k = [s_k^T y_k]$, $Y_k^T Y_k = [y_k^T y_k]$, compute $S_k^T g_{k+1}$, $Y_k^T g_{k+1}$ and go to Step 4.

Step 2: Matrix updating. If $k > m$ delete the first column of S_{k-1}, Y_{k-1} and the first row and column of $S_{k-1}^T Y_{k-1}$, $Y_{k-1}^T Y_{k-1}$. Using Procedure 2.1, form matrices S_k, Y_k, $S_k^T Y_k$, $Y_k^T Y_k$.

Step 3: Decomposition. Using Procedure 2.2, find matrices L_k, R_k and value of i_D.

Step 4: Direction vector. Compute $d_{k+1} = -H_{k+1}g_{k+1}$ by (2.15) for $i_D = 0$ or by (2.16) for $i_D = 1$, set $k := k + 1$ and go to Step 1.

3 Method that use previous difference vectors

In this section we propose a recursive process to construct diagonal matrix $\tilde{D}$ with positive elements and transformation matrices L, R with suitable properties, where R is upper triangular and L lower triangular, which approximate solution to problem $L(S^TY)R = \tilde{D}$. Then we give some properties of this process and show how it can be used to derive a new VM update, which is the BNS method with transformed matrices $\tilde{S} = SL^T$, $\tilde{Y} = YR$ instead of S, Y. We also discuss situation when some modifications of the decomposition process are needful or suitable. Finally, we describe a selected implementation of the second testing method in detail.

The use of previous difference vectors together with the BNS method with transformed matrices has the additional advantage – we recurrently update matrix $\zeta_k I$, $\zeta_k > 0$, by the BFGS method, using $\tilde{m}+1$ couples of vectors, here columns of transformed matrices $\tilde{S}$, $\tilde{Y}$. Therefore elements of transformation matrices L^T, R in case of quadratic objective functions can be derived not only from the idea of conjugate directions, but also variationally, see Section 3.2.

3.1 Nonsymmetric decomposition

Similarly as in Section 2.1, we first derive the recursive step of the decomposition. Let $S = [\underline{S}, s]$, $Y = [\underline{Y}, y]$, $L = \left[\begin{smallmatrix} \underline{L} & 0 \\ l^T & 1 \end{smallmatrix}\right]$, $R = \left[\begin{smallmatrix} \underline{R} & r \\ 0^T & 1 \end{smallmatrix}\right]$. Then we can write

$$\tilde{S} = SL^T = [\underline{S}, s]\begin{bmatrix} \underline{L}^T & l \\ 0^T & 1 \end{bmatrix} = [\underline{S}\,\underline{L}^T, \underline{S}\,l + s] \;\triangleq\; [\underline{\tilde{S}}, \tilde{s}], \tag{3.1}$$

$$\tilde{Y} = YR = [\underline{Y}, y]\begin{bmatrix} \underline{R} & r \\ 0^T & 1 \end{bmatrix} = [\underline{Y}\,\underline{R}, \underline{Y}\,r + y] \;\triangleq\; [\underline{\tilde{Y}}, \tilde{y}], \tag{3.2}$$

which implies

$$L(S^TY)R = \tilde{S}^T\tilde{Y} = \begin{bmatrix} \underline{\tilde{S}}^T\underline{\tilde{Y}} & \underline{\tilde{S}}^T\tilde{y} \\ \tilde{s}^T\underline{\tilde{Y}} & \tilde{s}^T\tilde{y} \end{bmatrix} = \begin{bmatrix} \underline{L}(\underline{S}^T\underline{Y})\underline{R} & \underline{L}(\underline{S}^T\tilde{y}) \\ (\tilde{s}^T\underline{Y})\underline{R} & \tilde{s}^T\tilde{y} \end{bmatrix}. \tag{3.3}$$

If we find vectors l, r in such a way that $\tilde{b} \triangleq \tilde{s}^T\tilde{y} > 0$ (then the corresponding BFGS update with transformed matrices preserves positive definiteness of the VM matrix) and $\underline{S}^T\tilde{y} = \underline{Y}^T\tilde{s} = 0$, we obtain

$$L(S^TY)R = \tilde{S}^T\tilde{Y} = \begin{bmatrix} \underline{L}(\underline{S}^T\underline{Y})\underline{R} & 0 \\ 0^T & \tilde{b} \end{bmatrix} = \begin{bmatrix} \underline{\tilde{S}}^T\underline{\tilde{Y}} & 0 \\ 0^T & \tilde{b} \end{bmatrix}. \tag{3.4}$$

Therefore, we converted the problem to find matrix $\tilde{D}$ with positive elements and nonsingular matrices L, R satisfying $\tilde{S}^T\tilde{Y} = L(S^TY)R = \tilde{D}$ to the problem to find matrix $\underline{\tilde{D}}$ with positive elements and nonsingular matrices $\underline{L}$, $\underline{R}$ satisfying $\underline{\tilde{S}}^T\underline{\tilde{Y}} = \underline{L}(\underline{S}^T\underline{Y})\underline{R} = \underline{\tilde{D}}$ and we can repeat this recursive step, etc.

Lemma 3.1. *Let matrix $\underline{S}^T\underline{Y}$ be nonsingular and let*

$$l^* = -(\underline{Y}^T\underline{S})^{-1}\underline{Y}^Ts, \qquad r^* = -(\underline{S}^T\underline{Y})^{-1}\underline{S}^Ty. \tag{3.5}$$

Then the unique solution (l,r) *to* $\underline{S}^T\tilde{y}=\underline{Y}^T\tilde{s}=0$ *is* (l^*,r^*). *Moreover, for any* l,r *it holds*

$$\tilde{b} = (l-l^*)^T \underline{S}^T \underline{Y}(r-r^*) + b^*, \qquad b^* = b - s^T \underline{Y}(\underline{S}^T \underline{Y})^{-1} \underline{S}^T y. \tag{3.6}$$

Proof. From $\underline{S}^T\underline{Y}r^* = -\underline{S}^T y$ and $\underline{Y}^T\underline{S}\,l^* = -\underline{Y}^T s$, we obtain firstly $\underline{S}^T\underline{Y}(r-r^*) = \underline{S}^T(\underline{Y}r+y) = \underline{S}^T\tilde{y}$ and similarly $\underline{Y}^T\underline{S}(l-l^*) = \underline{Y}^T\tilde{s}$, which gives the first part, and secondly

$$(l-l^*)^T \underline{S}^T \underline{Y}(r-r^*) = \left(l^T \underline{S}^T \underline{Y} r + l^T \underline{S}^T y + s^T \underline{Y} r\right) - s^T \underline{Y} r^*,$$

which by (3.5) yields

$$\begin{aligned} \tilde{b} &= (\underline{S}l+s)^T(\underline{Y}r+y) = \left(l^T \underline{S}^T \underline{Y} r + l^T \underline{S}^T y + s^T \underline{Y} r\right) + b \\ &= (l-l^*)^T \underline{S}^T \underline{Y}(r-r^*) + b - s^T \underline{Y}(\underline{S}^T \underline{Y})^{-1} \underline{S}^T y \end{aligned}$$

and concludes the proof. □

We show that in case of symmetric positive definite matrix S^TY, e.g. if objective function f is strictly convex quadratic (see Lemma 2.2), this recursive step is well defined.

Lemma 3.2. *Let matrix* S^TY *be symmetric positive definite. Then the recursive step described above is well defined,* $l^* = r^*$ *and* $b^* > 0$. *Moreover,* $\tilde{b} \ge b^*$ *for any* l, r *satisfying* $l=r$, *i.e.* $\tilde{b} > 0$ *holds and this value is minimized by the choice* $l = l^*$, $r = r^*$.

Proof. The recursive step is well defined by Lemma 3.1. Relations $l^* = r^*$ and $\tilde{b} \ge b^*$ for $l=r$ immediately follow from (3.5), (3.6) and symmetry and positive definiteness of S^TY. Let $u = \underline{S}^T y = \underline{Y}^T s$, $v = -(\underline{S}^T\underline{Y})^{-1}u = l^* = r^*$, $w = [v^T, 1]^T$. Then

$$0 < w^T(S^TY)w = [v^T,1]\begin{bmatrix} \underline{S}^T\underline{Y} & u \\ u^T & b \end{bmatrix}\begin{bmatrix} v \\ 1 \end{bmatrix} = v^T(\underline{S}^T\underline{Y})v + 2\,u^T v + b = b + u^T v = b^*$$

by (3.6) and positive definiteness of S^TY. □

Corollary 3.1. *If matrix* S^TY *is symmetric positive definite then the decomposition process described above is well defined and* $L^T = R$.

3.2 VM update derived from the complete decomposition

We suppose here that the complete decomposition of nonsingular $\underline{S}^T\underline{Y}$ was performed and that we have matrix $\underline{\tilde{D}}$ with positive elements and nonsingular matrices $\underline{R}$, $\underline{L}$ satisfying

$$\underline{\tilde{S}}^T \underline{\tilde{Y}} = \underline{L}(\underline{S}^T\underline{Y})\underline{R} = \underline{\tilde{D}}. \tag{3.7}$$

Using the BNS formula (1.4) with transformed matrices $\underline{\tilde{S}} = \underline{S}\,\underline{L}^T$, $\underline{\tilde{Y}} = \underline{Y}\,\underline{R}$, $\underline{\tilde{D}}$, $\underline{\tilde{U}} = \underline{\tilde{D}}$ instead of S, Y, D, U, we get

$$\begin{aligned} \tilde{H} &= \underline{\tilde{S}}\,\underline{\tilde{U}}^{-T}\underline{\tilde{D}}\,\underline{\tilde{U}}^{-1}\underline{\tilde{S}}^T + \zeta(I - \underline{\tilde{S}}\,\underline{\tilde{U}}^{-T}\underline{\tilde{Y}}^T)(I - \underline{\tilde{Y}}\underline{\tilde{U}}^{-1}\underline{\tilde{S}}^T) \\ &= \underline{\tilde{S}}\,\underline{\tilde{D}}^{-1}\underline{\tilde{S}}^T + \zeta(I - \underline{\tilde{S}}\,\underline{\tilde{D}}^{-1}\underline{\tilde{Y}}^T)(I - \underline{\tilde{Y}}\,\underline{\tilde{D}}^{-1}\underline{\tilde{S}}^T)\,, \end{aligned} \tag{3.8}$$

which yields

$$\tilde{H}\underline{\tilde{Y}} = \underline{\tilde{S}} \tag{3.9}$$

by (3.7). For $\tilde{b} > 0$ we will consider matrix

$$\tilde{H}_+ = \frac{\tilde{s}\tilde{s}^T}{\tilde{b}} + \Big(I - \frac{\tilde{s}\tilde{y}^T}{\tilde{b}}\Big)\tilde{H}\Big(I - \frac{\tilde{y}\tilde{s}^T}{\tilde{b}}\Big), \tag{3.10}$$

satisfying $\tilde{H}_+\tilde{y} = \tilde{s}$, and investigate its properties for various vectors l, r (recall that $\tilde{s}$, $\tilde{y}$ are given by (3.1), (3.2)). Since (3.10) represents the BFGS update of $\tilde{H}$ with vectors $\tilde{s}$, $\tilde{y}$, from theory given in [2] we can deduce that matrix $\tilde{H}_+$ can be obtained by the BNS formula (1.4) with transformed matrices $\tilde{S}$, $\tilde{Y}$, i.e. by analogy with (3.8):

$$\begin{aligned} \tilde{H}_+ &= \tilde{S}\,\tilde{D}^{-1}\tilde{S}^T + \zeta(I - \tilde{S}\,\tilde{D}^{-1}\tilde{Y}^T)(I - \tilde{Y}\tilde{D}^{-1}\tilde{S}^T) \\ &= SL^T\tilde{D}^{-1}LS^T + \zeta(I - SL^T\tilde{D}^{-1}R^TY^T)(I - YR\tilde{D}^{-1}LS^T)\,, \end{aligned} \tag{3.11}$$

where $\tilde{D} = \left[\begin{smallmatrix} \underline{\tilde{D}} & 0 \\ 0^T & \tilde{b} \end{smallmatrix}\right]$. As we can expect, matrix $\tilde{H}_+$ is always symmetric positive definite.

Lemma 3.3. *Matrix $\tilde{H}_+$ (if well defined) is always symmetric positive definite.*

Proof. Let $q \in \mathcal{R}^N$, $q \neq 0$. If $\tilde{S}^Tq \neq 0$, then also $\tilde{D}^{-1/2}\tilde{S}^Tq \neq 0$ and $q^T\tilde{H}_+q \geq |\tilde{D}^{-1/2}\tilde{S}^Tq|^2 > 0$ by (3.11), otherwise $q^T\tilde{H}_+q = \zeta\, q^Tq > 0$. □

For the choice $l = l^*$, $r = r^*$, QNC $\tilde{H}_+\tilde{Y} = \tilde{S}$ is satisfied and columns of $\tilde{S}\tilde{D}^{-1/2}$ are conjugate with respect to $\tilde{H}_+^{-1}$.

Lemma 3.4. *Let matrix $\tilde{H}_+$ be given by (3.11) and $l = l^*$, $r = r^*$ with $\tilde{b} = b^* > 0$. Then $\tilde{H}_+\tilde{Y} = \tilde{S}$ and $\tilde{S}^T\tilde{H}_+^{-1}\tilde{S} = \tilde{D}$, i.e. columns of $\tilde{S}\tilde{D}^{-1/2}$ are conjugate with respect to $\tilde{H}_+^{-1}$. If, in addition, matrix S^TY is symmetric positive definite, then also $\tilde{H}_+Y = S$ and $\tilde{H}\underline{Y} = \underline{S}$ hold.*

Proof. From (3.10) we get $\tilde{H}_+\tilde{y} = \tilde{s}$ and also $\tilde{H}_+\underline{\tilde{Y}} = \underline{\tilde{S}}$ by (3.9) and $\underline{Y}^T\tilde{s} = \underline{S}^T\tilde{y} = 0$ in view of $l = l^*$, $r = r^*$ by Lemma 3.1. Altogether, we have $\tilde{H}_+\tilde{Y} = \tilde{S}$ and from (3.7), (3.1) and (3.2) we obtain

$$\tilde{S}^T\tilde{H}_+^{-1}\tilde{S} = \tilde{S}^T\tilde{Y} = \begin{bmatrix} \underline{\tilde{S}}^T\underline{\tilde{Y}} & \underline{\tilde{S}}^T\tilde{y} \\ \tilde{s}^T\underline{\tilde{Y}} & \tilde{s}^T\tilde{y} \end{bmatrix} = \begin{bmatrix} \underline{\tilde{D}} & \underline{L}\,\underline{S}^T\tilde{y} \\ \tilde{s}^T\underline{Y}\,\underline{R} & \tilde{b} \end{bmatrix} = \begin{bmatrix} \underline{\tilde{D}} & 0 \\ 0^T & \tilde{b} \end{bmatrix} = \tilde{D}.$$

If matrix S^TY is symmetric positive definite, we have $L^T = R$ by Corollary 3.1, thus $\tilde{H}_+Y = SL^TR^{-1} = S$ and similarly $\tilde{H}\underline{Y} = \underline{S}$ by (3.9). □

The following theorem shows that for a quadratic objective function and the choice $l = l^*$, $r = r^*$, improvement of convergence is the best in some sense for linearly independent direction vectors ($\|.\|_F$ denotes the Frobenius matrix norm).

Theorem 3.1. *Let matrix $\tilde{H}_+$ be given by (3.10) with $l = r$, matrix $\tilde{H}$ by (3.8) and f be quadratic function $f(x) = \frac{1}{2}(x - x^*)^TG(x - x^*)$, $x^* \in \mathcal{R}^N$, with a symmetric positive definite matrix G and suppose that columns of matrix S are linearly independent. Then the recursive step described in Section 3.1 is well defined, $l^* = r^*$ and the choice $l = l^*$ implies $\tilde{H}_+Y = S$ and minimizes value $\|G^{1/2}\tilde{H}_+G^{1/2} - I\|_F$ as a function of l (or r).*

Proof. Matrix S^TY is symmetric positive definite by Lemma 2.2, therefore the recursive step is well defined, $l^* = r^*$ and $\tilde{b} \geq b^* > 0$ by Lemma 3.2. Denoting $\tilde{z} = G^{1/2}\tilde{s} = G^{-1/2}\tilde{y}$, $W = G^{1/2}\tilde{H}G^{1/2}$, $W_+ = G^{1/2}\tilde{H}_+G^{1/2}$ and $T = W - I$, we can rewrite (3.10) in the form

$$W_+ = (1/|\tilde{z}|^2)\tilde{z}\tilde{z}^T + PWP = I + PTP, \qquad P = I - (1/|\tilde{z}|^2)\tilde{z}\tilde{z}^T, \tag{3.12}$$

by $|\tilde{z}|^2 = \tilde{b} > 0$ and $P^2 = P$. As a special case, denoting by $\hat{s}$, $\hat{y}$, $\hat{H}_+$ vectors $\tilde{s}$, $\tilde{y}$ and matrix $\tilde{H}_+$ for $l = r = l^*$ and subsequently, $\hat{z} = G^{1/2}\hat{s} = G^{-1/2}\hat{y}$, $\hat{W}_+ = G^{1/2}\hat{H}_+G^{1/2}$, we can rewrite update (3.10) for this choice of l, r in the form

$$\hat{W}_+ = I + \hat{P}T\hat{P}, \qquad \hat{P} = I - (1/|\hat{z}|^2)\hat{z}\hat{z}^T, \tag{3.13}$$

by $|\hat{z}|^2 = b^* > 0$ and $\hat{P}^2 = \hat{P}$.

Using Lemma 3.4, we get $\hat{H}_+Y = S$ and $\tilde{H}\underline{Y} = \underline{S}$, which implies $WZ = Z$, or $TZ = 0$, where $Z = G^{1/2}\underline{S} = G^{-1/2}\underline{Y}$. Using (3.6) with $r = r^*$ and arbitrary l, we get $\tilde{s}^T\hat{y} = b^*$, which yields $\tilde{z}^T\hat{z} = b^* = \hat{z}^T\hat{z}$. In view of $\tilde{z} - \hat{z} = Z(l - l^*)$ by (3.1), we obtain

$$P\hat{P} = \Big(I - \frac{\tilde{z}\tilde{z}^T}{\tilde{b}}\Big)\hat{P} = \hat{P} - \frac{\tilde{z}}{\tilde{b}}\Big(\tilde{z}^T - \frac{\tilde{z}^T\hat{z}}{b^*}\hat{z}^T\Big) = \hat{P} - \frac{\tilde{z}}{\tilde{b}}\big(l - l^*\big)^T Z^T, \tag{3.14}$$

which yields $P\hat{P}T = \hat{P}T$ by $TZ = 0$, thus $P\hat{P}T\hat{P}P = \hat{P}T\hat{P}$. Using this together with (3.12), (3.13) and $\hat{P}^2 = \hat{P}$, we obtain

$$\mathrm{Tr}\Big[(\hat{W}_+ - W_+)(\hat{W}_+ - I)\Big] = \mathrm{Tr}\Big[(\hat{P}T\hat{P} - PTP)\hat{P}T\hat{P}\Big] = \mathrm{Tr}(T\hat{P}T\hat{P} - TP\hat{P}T\hat{P}P) = 0$$

by the fact that the trace of a product of two square matrices is independent of the order of the multiplication. This immediately implies

$$\|W_+ - \hat{W}_+\|_F^2 + \|\hat{W}_+ - I\|_F^2 = \|W_+ - I\|_F^2.$$

Since $W_+ = \hat{W}_+$ holds for $l = r = l^*$, value $\|W_+ - I\|_F$ is minimized by $l = r = l^*$. □

3.3 Approximate decomposition

If the last diagonal element $\tilde{b}$ of matrix $\tilde{S}^T\tilde{Y}$ in (3.4) is not positive, this recursive step of decomposition is unusable, since then update (3.10) need not preserve positive definiteness of the VM matrix. This drawback can be eliminated by modifications of transformation matrices R, L. Note that we consider here only the first recursive step, but it can be easily generalized to other steps.

Transformation matrices R, L can be modified in the following way. We first set $l = l^*$, $r = r^*$. If the resultant $\tilde{b} = b^*$ is too small or the corresponding elements of vectors $\underline{S}^Ty$, $\underline{Y}^Ts$ are too different (the reason why not to decompose in this case is discussed in Section 2.4), we replace the first elements of l, r by zero and try to find vectors l, r satisfying $\tilde{b} > 0$ with matrices S, Y without the first columns in the same way. If this $\tilde{b}$ is again too small or vectors $\underline{S}^Ty$, $\underline{Y}^Ts$ are too different, we try repeat this process with matrices S, Y without the first and second columns, etc.

A side effect of transformations is deterioration of stability; thus sometimes, if a contribution of transformation would be too small, it is better to omit a corresponding part of the decomposition, see Section 3.4 for details.

After the whole decomposition, we can use the BNS formula (1.4) with transformed matrices $\tilde{S} = SL^T$, $\tilde{Y} = YR$, $\tilde{D}$, $\tilde{U}$ instead of S, Y, D, U, where $\tilde{U}_{i,j} = (\tilde{S}^T\tilde{Y})_{i,j}$ for $i \le j$, $\tilde{U}_{i,j} = 0$ otherwise. This gives update formula

$$\begin{aligned} H_+^{(4)} &= \tilde{S}\,\tilde{U}^{-T}\tilde{D}\,\tilde{U}^{-1}\tilde{S}^T + \zeta(I - \tilde{S}\,\tilde{U}^{-T}\tilde{Y}^T)(I - \tilde{Y}\tilde{U}^{-1}\tilde{S}^T) \\ &= SL^T\tilde{U}^{-T}\tilde{D}\tilde{U}^{-1}LS^T + \zeta(I - SL^T\tilde{U}^{-T}R^TY^T)(I - YR\tilde{U}^{-1}LS^T). \end{aligned} \tag{3.15}$$

3.4 Implementation

In this section we describe only the second selected testing method. We use the same procedure for updating of basic low-order matrices S^TY, Y^TY as in Section 2. To decompose matrix S^TY with modifications mentioned in Section 3.3, we use the following procedure. Conditions in steps *(iv)* and *(v)* were found empirically.

Procedure 3.1 *(Decomposition)*

Given: A dimension $\bar{n}$ of matrix S^TY and $\sigma_{ij} = (S^TY)_{i,j}$, $1 \le i \le \bar{n}$, $1 \le j \le \bar{n}$.

(i): Set $n = 1$ and for $1 \le i \le \bar{n}$, $1 \le j \le \bar{n}$ set $L_{i,i} = R_{i,i} = 1$ and $\tilde{D}_{i,i} = \sigma_{ii}$, $L_{i,j} = R_{i,j} = \tilde{D}_{i,j} = 0$ for $i \ne j$ and further, $\tilde{U}_{i,j} = \sigma_{ij}$ for $i \le j$, $\tilde{U}_{i,j} = 0$ otherwise.

(ii): Set $n := n+1$. If $n > \bar{n}$ return, otherwise set $n_0 := 0$.

(iii): Set $n_0 := n_0+1$ (index of the first transformed column of S, Y). If $n_0 = n$ go to *(ii)*, otherwise set $i = n_0$ and $\Delta = 0$.

(iv): Set $\Delta := \Delta + 4^{n-i-1}|\sigma_{in} - \sigma_{ni}|/\sqrt{\sigma_{ii}\sigma_{nn}}$. If $\sigma_{ii}\sigma_{nn} < 0$ set $\Delta := 10^4\Delta$. Set $i := i+1$. If $i < n$ go to *(iv)*. If ($\Delta < 10^{-7}$ and $n_0 < n-1$) or $\Delta > 10^2$ then go to *(iii)*.

(v): With matrix $[\sigma_{ij}]_{\substack{j=n_0,\dots,n \\ i=n_0,\dots,n}}$ instead of S^TY find vectors $l^* \triangleq \hat{l}$, $r^* \triangleq \hat{r}$ and scalar b^*, using Lemma 3.1. If $b^* < 10^{-5}$ goto *(iii)*, otherwise form vectors l, r, $l_i = \hat{l}_{i-n_0+1}$, $r_i = \hat{r}_{i-n_0+1}$ for $n_0 \le i < n$, $l_i = r_i = 0$ for $1 \le i < n_0$.

(vi): Set $\tilde{U}_{n,n} = \tilde{D}_{n,n} = b^*$, $L_{n,i} = l_i$, $R_{i,n} = r_i$, $1 \le i < n$, and $[\tilde{U}_{1,n}, \dots, \tilde{U}_{n-1,n}]^T = [L_{ij}]_{\substack{j=1,\dots,n-1 \\ i=1,\dots,n-1}}\Big(\big([\sigma_{ij}]_{\substack{j=1,\dots,n-1 \\ i=1,\dots,n-1}}\big)r + [\sigma_{1n}, \dots, \sigma_{n-1,n}]^T\Big)$. Go to *(ii)*.

We now state the method in details. To compute the direction vector in Step 5, we use formula (3.15) and express H_+g_+ similarly as in (1.5):

$$\begin{aligned} H_+^{(4)}g_+ &= \zeta g_+ + \tilde{S}\Big[\tilde{U}^{-T}\Big((\tilde{D} + \zeta\tilde{Y}^T\tilde{Y})\tilde{U}^{-1}\tilde{S}^Tg_+ - \zeta\tilde{Y}^Tg_+\Big)\Big] - \tilde{Y}\Big[\zeta\tilde{U}^{-1}\tilde{S}^Tg_+\Big] \\ &= \zeta g_+ + S\Big[L^T\tilde{U}^{-T}\Big(\tilde{E}\tilde{U}^{-1}LS^Tg_+ - \zeta R^TY^Tg_+\Big)\Big] - Y\Big[\zeta R\tilde{U}^{-1}LS^Tg_+\Big] \end{aligned}, \tag{3.16}$$

where $\tilde{E} = \tilde{D} + \zeta R^T(Y^TY)R$ and in brackets we have low-order matrices. For simplicity, we omit stopping criteria and contingent restarts when the direction vector is not descent.

Algorithm 3.2

Data: A number $m \ge 1$ of VM updates per iteration and line search parameters ε_1, ε_2, $0 < \varepsilon_1 < 1/2$, $\varepsilon_1 < \varepsilon_2 < 1$.

Step 0: *Initiation*. Choose starting point $x_0 \in \mathcal{R}^N$, define starting matrix $H_0 = I$ and 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 (1.1), $g_{k+1} = \nabla f(x_{k+1})$, $s_k = t_k d_k$, $y_k = g_{k+1} - g_k$ and $\zeta_k = s_k^T y_k / y_k^T y_k$. If $k = 0$ set $S_k = [s_k]$, $Y_k = [y_k]$, $S_k^T Y_k = [s_k^T y_k]$, $Y_k^T Y_k = [y_k^T y_k]$, compute $S_k^T g_{k+1}$, $Y_k^T g_{k+1}$ and go to Step 4.

Step 2: Matrix updating. If $k > m$ delete the first column of S_{k-1}, Y_{k-1} and the first row and column of $S_{k-1}^T Y_{k-1}$, $Y_{k-1}^T Y_{k-1}$. Using Procedure 2.1, form matrices S_k, Y_k, $S_k^T Y_k$, $Y_k^T Y_k$.

Step 3: Decomposition. Using Procedure 3.1, form matrices $L_k, R_k, \tilde{U}_k$ and $\tilde{D}_k$.

Step 4: Direction vector. Compute $d_{k+1} = -H_{k+1} g_{k+1}$ by (3.16), set $k := k + 1$ and go to Step 1.

4 Numerical experiments

In this section, we demonstrate the influence of vectors corrections on the number of evaluations and computational time, using the following collections of test problems:

- [8] - Test 11 without problems 42, 48, 50, i.e. 55 problems, which are modified problems from CUTE collection [3]; used N are given in Table 1, where problems, modified in some way, are marked with '*',
- [1] - termed Test 12 here, 73 problems, $N = 1000$,
- [10] - Test 14, 22 problems, $N = 1000$,
- [7] - Test 25 without problems 40, 45, 48, 57, 58, 60, 61, 67-70, 79, i.e. 70 problems, $N = 1000$.

Problem	N	Problem	N	Problem	N	Problem	N
ARWHEAD	5000	DIXMAANI	3000	EXTROSNB	1000	NONDIA	5000
BDQRTIC	5000	DIXMAANJ	3000	FLETCBV3*	1000	NONDQUAR	5000
BROYDN7D	2000	DIXMAANK	3000	FLETCBV2	1000	PENALTY3	1000
BRYBND	5000	DIXMAANL	3000	FLETCHCR	1000	POWELLSG	5000
CHAINWOO	1000	DIXMAANM	3000	FMINSRF2	5625	SCHMVETT	5000
COSINE	5000	DIXMAANN	3000	FREUROTH	5000	SINQUAD	5000
CRAGGLVY	5000	DIXMAANO	3000	GENHUMPS	1000	SPARSINE	1000
CURLY10	1000	DIXMAANP	3000	GENROSE	1000	SPARSQUR	1000
CURLY20	1000	DQRTIC	5000	INDEF*	1000	SPMSRTLS	4999
CURLY30	1000	EDENSCH	5000	LIARWHD	5000	SROSENBR	5000
DIXMAANE	3000	EG2	1000	MOREBV*	5000	TOINTGSS	5000
DIXMAANF	3000	ENGVAL1	5000	NCB20*	1010	TQUARTIC*	5000
DIXMAANG	3000	CHNROSNB*	1000	NCB20B*	1000	WOODS	4000
DIXMAANH	3000	ERRINROS*	1000	NONCVXU2	1000		

Table 1. Dimensions for Test 11 – modified CUTE collection.

The source texts and reports can be downloaded from `camo.ici.ro/neculai/ansoft.htm` (Test 12), from `www.cs.cas.cz/~luksan/test.html` (Test 11, Test 14 and Test 25).

For comparison, Table 2 contains results for the following limited-memory methods: BNS – the BNS method, see [2], method from [14] based on conjugate directions (Algorithm 4.1) and our new Algorithm 2.3 and Algorithm 3.2. We have used $m = 5$ and the final precision $\|g(x^\star)\|_\infty \le 10^{-6}$.

Method	Test 11		Test 12		Test 14		Test 25	
	NFE	Time	NFE	Time	NFE	Time	NFE	Time
BNS	81457	44.08	26615	4.60	25763	7.92	129951	30.91
Alg. 4.1 in [14]	63107	40.36	16816	3.22	19010	5.91	118042	30.41
Algorithm 2.3	64842	31.19	24337	4.22	24316	7.23	105745	25.81
Algorithm 3.2	68557	36.64	25206	4.60	21781	6.53	113890	28.42

Table 2. Comparison of the selected methods.

References

[1] N. Andrei, *An unconstrained optimization test functions collection*, Advanced Modeling and Optimization, 10 (2008) 147-161.

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

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

[4] R. Fletcher: *Practical Methods of Optimization*, John Wiley & Sons, Chichester, 1987.

[5] Y.F. Hu, C. Storey: *Motivating Quasi-Newton Updates by Preconditioned Conjugate Gradient Methods*, Math. Report A 150, Department of Mathematical Sciences, Loughborough University of Technology, England, 1991.

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

[7] L. Lukšan, C. Matonoha, J. Vlček: *Sparse Test Problems for Unconstrained Optimization*, Report V-1064, Prague, ICS AS CR, 2010.

[8] L. Lukšan, C. Matonoha, J. Vlček: *Modified CUTE Problems for Sparse Unconstrained Optimization*, Report V-1081, Prague, ICS AS CR, 2010.

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

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

[11] I.A. Moughrabi: *New Implicit Multistep Quasi-Newton Methods*, Numerical Analysis and Applications 2 (2009) 154-164.

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

[13] J. Nocedal, S.J. Wright: *Numerical optimization*, Springer-Verlag, New York, 1999.

[14] J. Vlček, L. Lukšan: *A conjugate directions approach to improve the limited-memory BFGS method*, Report V-1120, Prague, ICS AS CR, 2011.