# Factorized Variable Metric Methods
# for Unconstrained Optimization*

### By Donald Goldfarb

Abstract. Several efficient methods are given for updating the Cholesky factors of a symmetric positive definite matrix when it is modified by a rank-two correction which maintains symmetry and positive definiteness. These ideas are applied to variable metric (quasi-Newton) methods to produce numerically stable algorithms.

1. Introduction. This paper is concerned with variable metric (VM) methods, (sometimes called quasi-Newton methods), for finding a local minimum of a nonlinear function, $f(x)$, of a vector $x = (x_1, \ldots, x_n)'$ of $n$ variables, where the prime "'" denotes transposition. The $k$th iteration of a VM method, is usually expressed as

$$(1.1) \qquad x^{(k+1)} = x^{(k)} - t_k H^{(k)} g^{(k)}, \qquad H^{(k+1)} = H^{(k)} + E^{(k)},$$

where $g^{(k)} = $ the gradient, $\nabla f(x)$, evaluated at $x^{(k)}$, and $t_k$ is a scalar, usually determined so that either $f(x^{(k+1)}) = \min_t f(x^{(k)} - tH^{(k)}g^{(k)})$ or $f(x^{(k+1)}) < f(x^{(k)})$. $H^{(k)}$ is a symmetric $n \times n$ matrix approximation to the inverse of the Hessian matrix, $G = \{\partial^2 f/\partial x_i \partial x_j\}$, of $f(x)$ at $x = x^{(k)}$, and $E^{(k)}$ is a matrix, typically of rank two, which is formed from $H^{(k)}$ and the vectors

$$(1.2) \qquad y^{(k)} = g^{(k+1)} - g^{(k)}$$

and

$$(1.3) \qquad s^{(k)} = x^{(k+1)} - x^{(k)},$$

subject to the condition that $H^{(k+1)}y^{(k)} = \rho s^{(k)}$. ($\rho$ is almost always required to be 1.)

The corrections $E^{(k)}$ used in the most widely known and used VM methods (e.g., the Davidon-Fletcher-Powell (DFP), [7], [12], the complementary DFP (comp-DFP), [6], [10], [18], [25], and the rank-one [5], [8], [23], [26] methods), all belong to the one-parameter family of correction formulas

$$(1.4) \qquad H^+ = H + \frac{ss'}{s'y} - \frac{Hyy'H}{y'Hy} + \beta rr',$$

where $r = Hy/y'Hy - s/s'y$ [5], [18], [25]. To simplify notation, we have suppressed the superscript $(k)$ and replaced $(k + 1)$ by a "plus". If a line search is performed at each step, these methods can all be shown to be superlinearly convergent for $f(x)$ strictly convex by combining Powell's elegant proof of this for the DFP method [24], with

Dixon's results for this one-parameter family of methods [9].

If in both the DFP ($\beta = 0$) and comp-DFP ($\beta = y'Hy$) methods the step length parameter $t$ is chosen so that $y's > 0$, and if exact arithmetic is used, then $H^+$ will be positive definite if $H$ is. This guarantees that at each step the direction

$$(1.5) \qquad\qquad\qquad p = - Hg$$

is downhill as long as the initial $H$ is chosen positive definite. Unfortunately, if one is computing on a finite precision machine, then $H$ can become indefinite or even singular because of rounding errors, leading to failure of the algorithm or premature termination. Even if the algorithm is designed to handle such eventualities, the nonpositive definiteness of $H$ will only be discovered when $g'Hg \leqslant 0$.

Strategies to deal with these difficulties that have been suggested include periodically resetting $H$ to the identity $I$, [1], [22] and rescaling variables [1]. In this paper the strategy we propose is to use a factorized positive definite approximation, $B = L_1 DL_1'$ to the Hessian matrix $G$ rather than an approximation $H$ to $G^{-1}$. $L_1$ and $D$ are, respectively, unit lower triangular and diagonal matrices. Formula (1) is then replaced by formulas for updating $L_1$ and $D$.

Gill and Murray [16] first suggested this approach. They give two methods for updating the Cholesky factors $L_1$ and $D$ when a symmetric rank-one matrix is added to $L_1 DL_1'$. Since the correction terms in (1) can be written as the sum of two symmetric rank-one terms, their procedures need be perfomed twice when applied to VM methods. Gill, Golub, Murray and Saunders [15] give further results along these lines for the factorization $LL'$. Modifying $LL'$ by analogs of the methods given below are described in [20]. Other methods for modifying the Cholesky factors of a positive definite matrix, when a matrix of rank one is added to it, have also been given by Bennett [2], Gentleman [14], and Fletcher and Powell [13].

In this paper recurrence formulas are derived for effecting the rank-two modification of $L_1 DL_1'$ by expressing it in product form. Three methods based upon this idea were first described by the author in [19] and [20]. In the next section, two efficient methods for orthogonally triangularizing $I + zw'$ are described. In Section 3, these methods are applied to VM updating formulas, expressed in product form to give methods for updating the Cholesky factors of a VM matrix. The recurrences of Section 2, although applied here to a particular problem, are general in nature and may be useful in other contexts. A rank-two analog of Gill and Murray's Method B [16], which is a natural outgrowth of the results of Sections 2 and 3 is presented in Section 4, while Section 5 outlines a way of coping with negative diagonal elements in this method, should they arise. Finally, Section 6 offers some comments and observations on the methods presented.

2. **Orthogonal Triangularization of** $I + zw'$. Let us consider the problem of finding an orthogonal matrix $Q$ that will reduce $I + zw'$ to a lower triangular matrix $\widetilde{L}$, i.e. $(I + zw')Q = \widetilde{L}$. Henceforth, the symbols $L$, $L_1$, $D$, and $Q$ will always denote lower triangular, unit lower triangular, diagonal and orthogonal matrices, respectively.

When $z = w$, Methods 1 and 2 given below reduce to Methods C5 and C2 (called

Method A in [16]) of Gill et al. [15], respectively. The derivations that follow parallel those given in [16] and [15] for the symmetric case. Gill and Murray [16] also give a method for determining $Q$ so that $(L + zw')Q$ is lower triangular. When $L = I$ their method becomes analytically equivalent to Method 1. However, it does not take advantage, as does Method 1, of the special nature of the matrices involved.

   *Method* 1: *Using Givens' Plane Rotations.* In this method $Q$ will be formed as a product of two sequences of Givens' plane rotations $Q_{n-1}, \ldots, Q_1$ and $\overline{Q}_1, \ldots, \overline{Q}_{n-1}$. The first sequence of rotations are chosen so that the last $n - 1$ elements of $w$ are successively transformed to zeros. Specially, if we define

(2.1)
$$Q_j = \begin{bmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & \ddots & & & & & \\ & & & 1 & & & & \\ & & & & c_j & s_j & & \\ & & & & -s_j & c_j & & \\ & & & & & & 1 & \\ & & & & & & & \ddots \\ & & & & & & & & \ddots \\ & & & & & & & & & 1 \end{bmatrix} \begin{matrix} \\ \\ \\ \\ j \\ j+1 \\ \\ \\ \\ \end{matrix},$$

$$\hat{w}^{(1)} = w \quad \text{and} \quad \hat{w}^{(n-j+1)} = Q_j \hat{w}^{(n-j)}, \quad j = n - 1, \ldots, 1,$$

where

$$c_j = \hat{w}_j^{(n-j)}/r_j, \qquad s_j = \hat{w}_j^{(n-j)}/r_j$$

and

$$r_j^2 = \hat{w}_{j+1}^{(n-j)^2} + \hat{w}_j^{(n-j)^2},$$

then

(2.2)
$$\hat{w}^{(n-j+1)} = (w_1, \ldots, w_{j-1}, \hat{w}_j, 0, \ldots, 0)'.$$

Except for the first element $\hat{w}_1$, all elements of $\hat{w}^{(n)}$ are zero. Moreover, as pointed out by Gill et al. [15], the matrices

$$Q^{(j)} = Q_j Q_{j+1} \cdots Q_{n-1}, \quad j = 1, \ldots, n - 1,$$

are special upper Hessenberg matrices of the form

(2.3)
$$Q^{(j)} = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & \overline{\beta}_j q_j & \overline{\beta}_j q_{j+1} & \cdots & \overline{\beta}_j q_{n-1} & \overline{\beta}_j q_n \\ & & -s_j & \overline{\beta}_{j+1} q_{j+1} & \cdots & \overline{\beta}_{j+1} q_{n-1} & \overline{\beta}_{j+1} q_n \\ & & & -s_{j+1} & \ddots & \vdots & \vdots \\ & & & & \ddots & \overline{\beta}_{n-1} q_{n-1} & \overline{\beta}_{n-1} q_n \\ & & & & & -s_{n-1} & \overline{\beta}_n q_n \end{bmatrix}.$$

Since $\hat{w}^{(n-j+1)} = Q^{(j)}w$ and $Q^{(j)\prime}Q^{(j)} = I$,

$$w = Q^{(j)\prime}\hat{w}^{(n-j+1)} = (w_1, \ldots, w_{j-1}, \bar{\beta}_j\hat{w}_j q_j, \ldots, \bar{\beta}_j\hat{w}_j q_n),$$

where the last equality follows from (2.2) and (2.3). Therefore, $\bar{\beta}_j\hat{w}_j = c$ and $q_k = w_k/c$, $k = j, \ldots, n$, for some arbitrary nonzero constant $c$. If $w_n \neq 0$, and we choose $c = 1$, then the following recurrence with $d_j = 1$, for all $j$, can be used to determine $Q^{(1)}$. A simple proof of this is given in [20].

*Recurrence* 1.   1. Set $\bar{\beta}_n = 1/w_n$.
    2. For $j = n - 1, \ldots, 1$, set

$$r_j = \bar{\beta}_{j+1}w_j, \qquad s_j = (r_j^2 + d_j/d_{j+1})^{-\frac{1}{2}}, \qquad c_j = r_j s_j,$$

$$\bar{\beta}_j = s_j\bar{\beta}_{j+1}, \qquad \bar{\beta}_{j+1} = c_j\bar{\beta}_{j+1}.$$

The quantities $d_j$ that appear in the recurrences given in this section have been included so that these recurrences can be used to generate the $\beta_j$, $\gamma_j$, and $\lambda_j$ that appear in formulas (3.7) and (3.13) of Section 3 for updating the Cholesky factors of a VM matrix. For the purposes of this section these $d_j$ should all be considered to have the value 1.

If $w_i = 0$ for $k + 1 \leqslant i \leqslant n$ and $w_k \neq 0$, then all recurrences given in this section apply with $n$ replaced by $k$, and for the purposes of the next section $\beta_i = \gamma_i = 0$ and $\lambda_i = 1$ for $k + 1 \leqslant i \leqslant n$. In a computer code any $w_i$ such that $|w_i| < \epsilon\|w\|$, should be treated as zero, where $\epsilon$ is the machine precision. Also, one should then not have problems of overflow since it can be shown that

$$r_j^2 \leqslant \frac{d_n}{d_{j+1}} \frac{w_j^2}{w_n^2}.$$

It should be clear from the above discussion that

$$\bar{H} = (I + zw')Q^{(1)\prime} = Q^{(1)\prime} + \bar{\gamma}_1 ze_1',$$

where $\bar{\gamma}_1 = 1/\bar{\beta}_1$, $e_1' = (1, 0, \ldots, 0)$, and $\bar{H}$ is the lower Hessenberg matrix

$$\bar{H} = \begin{bmatrix} \bar{\beta}_1 w_1 + \bar{\gamma}_1 z_1 & -s_1 & & & \\ \bar{\beta}_1 w_2 + \bar{\gamma}_1 z_2 & \bar{\beta}_2 w_2 & \ddots & & \\ \vdots & \vdots & \ddots & -s_{n-2} & \\ \bar{\beta}_1 w_{n-1} + \bar{\gamma}_1 z_{n-1} & \bar{\beta}_2 w_{n-1} & \cdots & \bar{\beta}_{n-1} w_{n-1} & -s_{n-1} \\ \bar{\beta}_1 w_n + \bar{\gamma}_1 z_n & \bar{\beta}_2 w_n & \cdots & \bar{\beta}_{n-1} w_n & \bar{\beta}_n w_n \end{bmatrix}.$$

By properly choosing a sequence of rotations $\bar{Q}_1, \ldots, \bar{Q}_{n-1}$ to successively reduce the superdiagonal elements of $\bar{H}$ to zero, we can transform $\bar{H}$ into the lower triangular matrix

$$(2.4) \quad \widetilde{L}(\beta, w, \gamma, z, \lambda) = \begin{bmatrix} \lambda_1 & & & & \\ \beta_1 w_2 + \gamma_1 z_2 & \lambda_2 & & & \\ \beta_1 w_3 + \gamma_1 z_3 & \beta_2 w_3 + \gamma_2 z_3 & \cdot & & \\ \vdots & \vdots & & \ddots & \\ & & & & \lambda_{n-1} \\ \beta_1 w_n + \gamma_1 z_n & \beta_2 w_n + \gamma_2 z_n & \cdots & \beta_{n-1} w_n + \gamma_{n-1} z_n & \lambda_n \end{bmatrix},$$

where the $\lambda_j$, $\beta_j$, and $\gamma_j$ can be computed from

*Recurrence* 2. 1. Set $\overline{\gamma}_1 = 1/(d_1 \overline{\beta}_1)$, $\overline{\lambda}_1 = \overline{\beta}_1 w_1 + \overline{\gamma}_1 z_1$.

2. For $j = 1, \ldots, n - 1$ set

$$\lambda_j^2 = \overline{\lambda}_j^2 + s_j^2 \frac{d_j}{d_{j+1}}, \qquad \overline{c}_j = \overline{\lambda}_j / \lambda_j, \qquad \overline{s}_j = s_j / \lambda_j,$$

$$\beta_j = \overline{c}_j \overline{\beta}_j - \overline{s}_j \overline{\beta}_{j+1}, \qquad \gamma_j = \overline{c}_j \overline{\gamma}_j,$$

$$\overline{\gamma}_{j+1} = \frac{d_j}{d_{j+1}} \overline{s}_j \overline{\gamma}_j, \qquad \overline{\beta}_{j+1} = \frac{d_j}{d_{j+1}} \overline{s}_j \overline{\beta}_j + \overline{c}_j \overline{\beta}_{j+1},$$

$$\overline{\lambda}_{j+1} = \overline{\beta}_{j+1} w_{j+1} + \overline{\gamma}_{j+1} z_{j+1}.$$

3. Set $\lambda_n = \overline{\lambda}_n$.

$\overline{Q}_j$ is given by (2.1) with $c_j$ and $s_j$ replaced by $\overline{c}_j$ and $\overline{s}_j$ as defined in the above recurrence.

This recurrence with $d_j = 1$ for all $j$ follows directly from the identity

$$\begin{bmatrix} \overline{\lambda}_j & -s_j \\ \overline{\beta}_j w_{j+1} + \overline{\gamma}_j z_{j+1} & \overline{\beta}_{j+1} w_{j+1} \end{bmatrix} \begin{bmatrix} \overline{c}_j & \overline{s}_j \\ -\overline{s}_j & \overline{c}_j \end{bmatrix}$$

$$= \begin{bmatrix} \overline{c}_j \overline{\lambda}_j + \overline{s}_j s_j & \overline{s}_j \overline{\lambda}_j - \overline{c}_j s_j \\ (\overline{c}_j \overline{\beta}_j - \overline{s}_j \overline{\beta}_{j+1}) w_{j+1} + \overline{c}_j \overline{\gamma}_j z_{j+1} & (\overline{s}_j \overline{\beta}_j + \overline{c}_j \overline{\beta}_{j+1}) w_{j+1} + \overline{s}_j \overline{\gamma}_j z_{j+1} \end{bmatrix}$$

and the requirement that $\overline{c}_j / \overline{s}_j = \overline{\lambda}_j / s_j$.

Consequently, if $\overline{Q} = \overline{Q}_1, \ldots, \overline{Q}_{n-1}$, then $(I + zw')Q^{(1)'} \overline{Q} = \overline{H} \overline{Q} = \widetilde{L}$.

*Method* 2: *Using Householder Transformations.* In this method $I + zw'$ is reduced to lower triangular form by a sequence of $n - 1$ Householder transformations, $P_1, \ldots, P_{n-1}$. The $j$th step of this method is based upon the use of a Householder matrix $P$ to transform the first row of a matrix of the form $I + (vz - \eta w)w'$ to a multiple of $e_1$. In particular, $P_j$ has the form

$$\begin{bmatrix} I & \vdots & 0 \\ \hline 0 & \vdots & P \end{bmatrix} \begin{matrix} \} j - 1 \\ \} n - j + 1 \end{matrix}$$

If we partition $w$ and $z$ as

$$w' = [w_1 \mathbin{\vdots} \overline{w}'] \quad \text{and} \quad z' = [z_1 \mathbin{\vdots} \overline{z}']$$

and choose $P = I - \sigma uu'$, where

$$a = \nu z_1 - \eta w_1, \quad u = [\mu \mathbin{\vdots} a\overline{w}'], \quad \lambda = \pm (\theta^2 + a^2 \overline{w}'\overline{w})^{1/2},$$

(2.5)

$$\theta = 1 + aw_1, \quad \mu = \theta - \lambda \quad \text{and} \quad \sigma = - 1/\mu\lambda,$$

then $(I + (\nu z - \eta w)w')P$ can be written as

$$\left[\begin{array}{c:c} \theta & a\overline{w}' \\ \hdashline w_1(\nu\overline{z} - \eta\overline{w}) & I + (\nu\overline{z} - \eta\overline{w})\overline{w}' \end{array}\right] \left[\begin{array}{c:c} 1 - \sigma\mu^2 & -\sigma a\mu\overline{w}' \\ \hdashline -\sigma a\mu\overline{w} & I - \sigma a^2 \overline{w}\overline{w}' \end{array}\right]$$

$$= \left[\begin{array}{c:c} \theta - \sigma\mu(\theta\mu + a^2\overline{w}'\overline{w}) & [1 - \sigma(\theta\mu + a^2\overline{w}'\overline{w})]\,a\overline{w}' \\ \hdashline \gamma\overline{z} + \beta\overline{w} & I + (\overline{\nu}\overline{z} - \overline{\eta}\,\overline{w})\overline{w}' \end{array}\right],$$

where

$$\gamma = [w_1 - \sigma\mu(\mu w_1 + a\overline{w}'\overline{w})]\,\nu, \quad \beta = - \{[w_1 - \sigma\mu(\mu w_1 + a\overline{w}'\overline{w})]\,\eta + \sigma\mu a\},$$

(2.6)

$$\overline{\nu} = [1 - \sigma a(\mu w_1 + a\overline{w}'\overline{w})]\,\nu \quad \text{and} \quad \overline{\eta} = [1 - \sigma a(\mu w_1 + a\overline{w}'\overline{w})]\,\eta + \sigma a^2.$$

After some algebraic manipulation it can be shown that the top row of the last partitioned matrix equals $[\lambda \mathbin{\vdots} 0]$,

$$\gamma = (w_1\theta + a\overline{w}'\overline{w})\nu/\lambda, \quad \beta = - (w_1\theta + a\overline{w}'\overline{w})\eta/\lambda + a/\lambda,$$

$$\overline{\nu} = - \nu/\lambda \quad \text{and} \quad \overline{\eta} = - (\eta + a^2/\mu)/\lambda.$$

Since the submatrix $I + (\overline{\nu}\overline{z} - \overline{\eta}\overline{w})\overline{w}'$ has the same form as the original, the entire orthogonal triangularization of $I + zw'$ may be specified recursively starting with $\nu = 1$ and $\eta = 0$. The resulting lower triangular matrix is given by

$$(I + zw')P_1 \cdots P_{n-1} = \widetilde{L}(\beta, w, \gamma, z, \lambda),$$

where $\widetilde{L}$ is defined by (2.4) and the scalars $\beta_j$, $\gamma_j$, and $\lambda_j$ are computed from

*Recurrence* 3. 1. Set $\nu_0 = 1, \eta_0 = 0$.

2. For $j = 1, \ldots, n - 1$ set

$$a_j = \nu_{j-1} z_j - \eta_{j-1} w_j, \quad \theta_j = 1 + a_j w_j/d_j,$$

$$\lambda_j = \pm \left(\theta_j^2 + \frac{a_j^2}{d_j} \sum_{k=j+1}^{n} \frac{w_k^2}{d_k}\right)^{1/2}, \quad \mu_j = \theta_j - \lambda_j,$$

$$b_j = \theta_j w_j + a_j \sum_{k=j+1}^{n} \frac{w_k^2}{d_k},$$

$$\gamma_j/\lambda_j = b_j \nu_{j-1}/\lambda_j^2 d_j, \quad \beta_j/\lambda_j = (a_j - b_j\eta_{j-1})/\lambda_j^2 d_j,$$

$$\nu_j = - \nu_{j-1}/\lambda_j, \quad \eta_j = - (\eta_{j-1} + a_j^2/\mu_j d_j)/\lambda_j.$$

3. Set $a_n = \nu_{n-1} z_n - \eta_{n-1} w_n, \lambda_n = 1 + a_n w_n/d_n$

with $d_j = 1$ for all $j$. Cancellation in the computation of $\mu_j$ is avoided if the sign of $\lambda_j$ is chosen to be opposite to that of $\theta_j$.

If we define $s_j = \Sigma_{k=j+1}^n w_k^2/d_k$, $j = 1, \ldots, n-1$ and $s_n = 0$, then

$$s_{j-1} = s_j - w_j^2/d_j, \quad j = n, n-1, \ldots, 2.$$

The $s_j$, $j = n-1, \ldots, 1$, should be computed and stored prior to using Recurrence 3. If they are not stored, they can be regenerated in the reverse order in the recurrence itself, as long as some precaution is taken to prevent any $s_j$ from becoming negative because of rounding errors.

Computing the $\beta_j$, $\gamma_j$, and $\lambda_j$ either by Method 1 (Recurrences 1 and 2) or by Method 2 (Recurrence 3) requires just $O(n)$ operations. In addition $2(n-1)$ and $n-1$ square roots are required, respectively, by these methods.

3. **Updating the Cholesky Factors of a VM matrix.** By replacing $H$ by $B$, $B$ by $H$, $y$ by $s$, $s$ by $y$, and $\beta$ by $\bar{\beta}$ in formula (1.4), we obtain a one-parameter family of updating formulas for $B$ that are "dual" or "complementary", in the sense of Fletcher [10], to the family (1.4) for $H$:

$$(3.1) \qquad B^+ = B + \frac{yy'}{y's} - \frac{Bss'B}{s'Bs} + \bar{\beta}rr',$$

where

$$r = Bs/s'Bs - y/y's.$$

Following Brodlie et al. [3], we can write the above in product form as

$$(3.2) \qquad B^+ = (I + vu')B(I + uv'),$$

where

$$(3.3) \qquad u = s + \alpha Hy,$$

$$(3.4) \qquad v = \theta_1 Bs + \theta_2 y,$$

$$\theta_1 = -\left[\frac{(\alpha^2 d + \alpha b)a + 1}{\alpha^2 d + 2\alpha b + c}\right], \qquad \theta_2 = a + \frac{\alpha a(c + \alpha b) - \alpha}{\alpha^2 d + 2\alpha b + c},$$

and

$$a = [(\alpha + 1)^2 b^2 + (c - b)(b - \alpha^2 d)]^{-\frac{1}{2}},$$

$b = y's$, $c = s'Bs$, $d = y'Hy$ and $H = B^{-1}$. As in [3], it can be shown that the parameters $\alpha$ and $\bar{\beta}$ are related by the equation

$$\bar{\beta} = \frac{bc[\alpha^2(b + d) + 2\alpha b]}{(\alpha + 1)^2 b^2 + (c - b)(b - \alpha^2 d)};$$

and that the conditions $(\alpha + 1)^2 b^2 + (c - b)(b - \alpha^2 d) > 0$, $\alpha \neq b/d$, and $y's > 0$ guarantee that $B^+$ will be positive definite if $B$ is, assuming exact arithmetic.

Setting $\alpha = 0$ or $-2b/(b + d)$ yields the product form of the DFP updating

formula for $B$. As is well known, this corresponds to updating $H = B^{-1}$ by the comp-DFP formula. If $u$ and $v$ in (3.3) and (3.4) are multiplied by an appropriate scalar $\rho$ and $1/\rho$, respectively, they can be expressed in this case, simply as

$$u = p = -Hg \quad \text{and} \quad v = \frac{y}{((y's)(-p'g))^{1/2}} - \frac{g}{p'g}.$$

Similarly, if $\alpha = \pm\sqrt{b/d}$, we obtain the comp-DFP updating formula for $B$ and can write $u$ and $v$ as

$$u = \frac{Hy}{((y'Hy)(y's))^{1/2}} - \frac{s}{y's} \quad \text{and} \quad v = y.$$

Let us now consider updating the factorization

(3.5) $$B = L_1 D L_1'.$$

If we let $L = L_1 D^{1/2}$, then from (3.2) we have that

$$B^+ = (I + vu')LL'(I + uv') = L(I + \hat{z}\hat{w}')(I + \hat{w}z')L',$$

where $L\hat{z} = v$ and $L\hat{w} = LL'u = Bu$.

In the previous section, it was shown how to obtain the factorization $I + \hat{z}\hat{w}' = \widetilde{L}Q'$, where $\widetilde{L} = \widetilde{L}(\hat{\beta}, \hat{w}, \hat{\gamma}, \hat{z}, \lambda)$ is a special lower triangular matrix of the form (2.4). Since $Q'Q = I$, we have that

$$B^+ = L\widetilde{L}Q'Q\widetilde{L}'L' = L^+L^{+'},$$

where

$$L^+ = L\widetilde{L} = L_1 D^{1/2}\widetilde{L}$$

is lower triangular. It should be apparent that

$$D^{1/2}\widetilde{L}(\hat{\beta}, \hat{w}, \hat{\gamma}, \hat{z}, \lambda) = \widetilde{L}(\hat{\beta}, w, \hat{\gamma}, z, \hat{\lambda}) = \widetilde{L}(\widetilde{\beta}, w, \widetilde{\gamma}, z, e)\hat{D}^{1/2},$$

where

$$w = D^{1/2}\hat{w}, \quad z = D^{1/2}\hat{z}, \quad \hat{\lambda} = D^{1/2}\lambda, \quad e = (1, \ldots, 1)',$$

$$D^{1/2} = \text{diag}(\hat{\lambda}_1, \ldots, \hat{\lambda}_n), \quad \widetilde{\beta} = \hat{D}^{-1/2}\hat{\beta} \quad \text{and} \quad \widetilde{\gamma} = \hat{D}^{-1/2}\hat{\gamma}.$$

Clearly then, $B^+ = L_1^+ D^+ L_1^{+'}$, where

(3.6) $$L_1^+ = L_1\widetilde{L}(\widetilde{\beta}, w, \widetilde{\gamma}, z, e),$$

(3.7) $$D^+ = \hat{D} = \text{diag}(d_1\lambda_1^2, \ldots, d_n\lambda_n^2),$$

(3.8) $$L_1 w = Bu$$

and

(3.9) $$L_1 z = v.$$

If we define $\widetilde{w}$ and $\widetilde{z}$ by

(3.10) $$L_1\widetilde{w} = -g$$

and

$$(3.11) \qquad\qquad\qquad L_1 \widetilde{z} = y,$$

then by (1.1)–(1.3), (1.5), (3.7) and (3.8)

$$w = t\widetilde{w} + \alpha\widetilde{z} \quad \text{and} \quad z = \theta_1\widetilde{w} + \theta_2\widetilde{z}.$$

We also note that we can simultaneously compute the solution $\widetilde{w}$ of (3.10) and the vectors $\widetilde{w}^{(2)}, \ldots, \widetilde{w}^{(n)}$ needed in (3.13) below, from

$$(3.12) \qquad\qquad\qquad \widetilde{w}^{(j+1)} + \widetilde{w}_j l^{(j)} = \widetilde{w}^{(j)}$$

starting with $\widetilde{w}^{(1)} = -g$, where $l^{(j)}$ denotes the $j$th column of $L_1$. The solution $\widetilde{z}$ of (3.11) and the vectors $\widetilde{z}^{(2)}, \ldots, \widetilde{z}^{(n)}$ can also be computed in this way starting with $\widetilde{z}^{(1)} = y$. Finally, we obtain that (3.6) is equivalent to the following updating formulas for the columns of $L_1$:

$$(3.13) \qquad
\begin{aligned}
l^{(j)+} &= l^{(j)} + t\left(\frac{\beta_j}{\lambda_j} + \theta_1\frac{\gamma_j}{\lambda_j}\right)\widetilde{w}^{(j+1)} \\
&\quad + \left(\alpha\frac{\beta_j}{\lambda_j} + \theta_2\frac{\gamma_j}{\lambda_j}\right)\widetilde{z}^{(j+1)}, \qquad j = 1, \ldots, n-1,
\end{aligned}$$

where $\beta_j, \gamma_j, \lambda_j, j = 1, \ldots, n-1$, and $\lambda_n$ can be computed recursively in terms of $w_j, z_j$ and $d_j, j = 1, \ldots, n$, by Methods 1 or 2 of Section 2.

$\beta_j$ and $\gamma_j$ determined by these recurrences and $\hat{\gamma}_j, \widetilde{\gamma}_j, \hat{\beta}_j$ and $\widetilde{\beta}_j$ defined above satisfy the relations

$$(3.14) \qquad
\begin{aligned}
\gamma_j &= \hat{\gamma}_j/d_j^{1/2} = \widetilde{\gamma}_j\lambda_j, \\
\beta_j &= \hat{\beta}_j/d_j^{1/2} = \widetilde{\beta}_j\lambda_j.
\end{aligned}$$

The scalars $\theta_1$ and $\theta_2$ can be computed with $O(n)$ operations and one square root since the quantities $b, c$ and $d$, which appear in their definition can be expressed as $b = ty'p = t\widetilde{z}'D^{-1}\widetilde{w}, c = -t^2p'g = t\widetilde{w}'D^{-1}\widetilde{w}$ and $d = \widetilde{z}'D^{-1}\widetilde{z}$. Consequently, one full step of a VM algorithm which updates the factorization (3.5) using (3.7), (3.13) and either Recurrences 1 and 2 or Recurrence 3 can be done with just $5/2n^2 + O(n)$ multiplications and divisions. This includes the costs of computing the search direction $p$ from $L_1'p = D^{-1}\widetilde{w}$ and all square roots, since each of the latter require $O(1)$ operations.

From (3.6), (3.14) and analogs of (3.12) corresponding to the forward eliminations (3.8) and (3.9) we have that

$$(3.15) \qquad
\left.
\begin{aligned}
l^{(j)+} &= l^{(j)} + \frac{\beta_j}{\lambda_j} w^{(j+1)} + \frac{\gamma_j}{\lambda_j} z^{(j+1)} \\
&= \left(1 - \frac{\beta_j}{\lambda_j} w_j - \frac{\gamma_j}{\lambda_j} z_j\right)l^{(j)} + \frac{\beta_j}{\lambda_j} w^{(j)} + \frac{\gamma_j}{\lambda_j} z^{(j)}
\end{aligned}
\right\}, \qquad j = 1, \ldots, n-1.$$

As it is possible to show, either from Recurrences 1 and 2 or from Recurrence 3, that the term in brackets in (3.15) is equal to $1/\lambda_j^2$,

$$(3.16) \qquad l^{(j)+} = (1/\lambda_j^2)l^{(j)} + \frac{\beta_j}{\lambda_j} w^{(j)} + \frac{\gamma_j}{\lambda_j} z^{(j)}, \qquad j = 1, \ldots, n-1.$$

Fletcher and Powell analyze the growth of rounding errors for formulas analogous to (3.13) and (3.16) that arise when the Cholesky factors of $B$ are updated to account for the addition of a symmetric rank-one matrix and show that factors of $\lambda_j$ and $1/\lambda_j$, respectively, appear in their error bounds [13]. Since use of (3.16) for $j = 1, \ldots,$ $n-1$ entails an additional $1/2n^2$ operations, Fletcher and Powell follow Gentleman's idea [14] of only using (3.16) when the term involving $\lambda_j$ (i.e., $\sqrt{d_j^+/d_j}$) in the error bound for (3.13) becomes dominant. In their case this occurs for $\lambda_j > 2$. It is suggested that a similar strategy be used with the methods given here.

Fletcher and Powell [13] also report that on the basis of a large number of applications of their "composite-$t$" method that $\lambda_j > 2$ on fewer than $1/n$ occasions on the average. In our methods, the use of (3.16) should be required even less frequently, since the values of $\lambda_j$ that are computed in our method cannot be greater than those computed by the composite-$t$ method. Consequently, the operation counts (based solely on (3.13)), already given still hold on the average.

It is evident from (3.7) that $d_j^+ \geqslant 0, j = 1, \ldots, n$. From Recurrences 1 and 2 one can show that in Method 1

$$\lambda_j^2 \geqslant \frac{d_j}{d_{j+1}} s_j^2 \geqslant \frac{d_j}{d_j + d_n(w_j^2/w_n^2)} > 0, \qquad j = 1, \ldots, n-1.$$

If Method 2 is used, then from Recurrence 3 and the assumption that $w_n \neq 0$,

$$\lambda_j^2 \geqslant \max\left(\theta_j^2, \frac{a_j^2}{d_j} \sum_{k=j+1}^{n} \frac{w_k^2}{d_k}\right) > 0, \qquad j = 1, \ldots, n-1,$$

since if $\theta_j = 0$ then $a_j \neq 0$.

Because of rounding errors there is the possibility (with essentially zero probability), that $\lambda_n = 0$. Therefore, one can state, with probability equal to one, that Methods 1 and 2 preserve the positive definiteness of $B$.

**4. An Alternate Method.** The VM updating formula (3.1) can always be written in the form

$$B^+ = B + \sigma uu' + \tau vv',$$

where

$$\tau = \bar{\beta} + \bar{u}'s, \qquad \sigma = \frac{\bar{\beta}(\bar{u} - u)'s - (\bar{u}'s)(u's)}{\tau(u's)^2}, \qquad v = \frac{\bar{u}}{\bar{u}'s} - \frac{\bar{\beta}}{\tau} \frac{u}{u's}$$

and either (i) $u = Bs$ and $\bar{u} = y$, or (ii) $u = -y$ and $\bar{u} = -Bs$. If $\bar{\beta} = s'Bs$ or $\bar{\beta} = -y's$, then $\bar{u}$ and $u$ must be defined by (i) and (ii), respectively. These two special

cases correspond to the variationally derived updates considered by Goldfarb [18]. $\bar{\beta} = (\bar{u}'s)\,(u's)/(\bar{u}-u)'s$, i.e., $\sigma = 0$, gives the rank-one updating formula.

Hence, given the Cholesky factorization (3.5) for $B$, we can write

$$(4.1) \qquad\qquad B^+ = L_1(D + \sigma ww' + \tau zz')L_1',$$

where

$$L_1 w = u \quad \text{and} \quad L_1 z = v.$$

If $\bar{\beta}$ is chosen so that $B^+$ is positive definite, i.e. $\bar{\beta} > cb^2/(b^2 - dc)$, then

$$(4.2) \qquad\qquad D + \sigma ww' + \tau zz' = \widetilde{L}_1\widetilde{D}\widetilde{L}_1'$$

must also be positive definite and have a Cholesky factorization $\widetilde{L}_1\widetilde{D}\widetilde{L}_1'$. From the previous section we know that $\widetilde{L}_1 = \widetilde{L}(\beta, w, \gamma, z, e)$ is of the special type (2.4). This could also be proved by induction; (see [15] for the case $z = w$). Therefore, the equations that are usually used for obtaining the factorization (4.2) become

$$\left.\begin{array}{l}
\displaystyle\sum_{k=1}^{j-1} \widetilde{d}_k(\beta_k w_j + \gamma_k z_j)^2 + \widetilde{d}_j = d_j + \sigma w_j^2 + \tau z_j^2 \quad \text{and} \\[2em]
\displaystyle\sum_{k=1}^{j-1} \widetilde{d}_k(\beta_k w_j + \gamma_k z_j)(\beta_k w^{(j+1)} + \gamma_k z^{(j+1)}) + \widetilde{d}_j(\beta_j w^{(j+1)} + \gamma_j z^{(j+1)}) \\[2em]
\qquad\qquad\qquad = \sigma w_j w^{(j+1)} + \tau z_j z^{(j+1)}
\end{array}\right\},\; j = 1, \ldots, n,$$

where $w_j$ and $z_j$ are the $j$th elements of $w$ and $z$ and $w^{(j+1)}$ and $z^{(j+1)}$ are vectors of the last $n - j$ elements of $w$ and $z$. When $j = 1$, $\Sigma_{k=1}^{j-1}\, a_j$ is defined to be zero. Equating coefficients of $w^{(j+1)}$ and $z^{(j+1)}$ yields

$$\beta_j = \frac{1}{\widetilde{d}_j}\left[\sigma w_j - \sum_{k=1}^{j-1}\widetilde{d}_k(\beta_k w_j + \gamma_k z_j)\beta_k\right], \quad j = 1, \ldots, n-1,$$

and

$$\gamma_j = \frac{1}{\widetilde{d}_j}\left[\tau z_j - \sum_{k=1}^{j-1}\widetilde{d}_k(\beta_k w_j + \gamma_k z_j)\gamma_k\right], \quad j = 1, \ldots, n-1.$$

If we define

$$\left.\begin{array}{l}
\eta_j = \gamma_j\widetilde{d}_j, \qquad \alpha_j = \beta_j\widetilde{d}_j, \\[1.5em]
\sigma_j = \sigma - \displaystyle\sum_{k=1}^{j-1}\alpha_k\beta_k \\[2em]
\tau_j = \tau - \displaystyle\sum_{k=1}^{j-1}\eta_k\gamma_k
\end{array}\right\}, \qquad j = 1, \ldots, n-1,$$

and

$$\mu_j = - \sum_{k=1}^{j-1} \alpha_k \gamma_k,$$

then recurrence relations for $\beta_j$, $\gamma_j$, $\widetilde{d}_j$ and these quantities are given by

Method 3: *Using Cholesky Factorization.*

Recurrence 4.

1. $\sigma_1 = \sigma, \qquad \tau_1 = \tau, \qquad \mu_1 = 0$

2. $\alpha_j = \sigma_j w_j + \mu_j z_j, \qquad \eta_j = \tau_j z_j + \mu_j w_j$

$$\left. \begin{aligned} \widetilde{d}_j &= d_j + \alpha_j w_j + \eta_j z_j \\ \beta_j &= \alpha_j / \widetilde{d}_j, \ \gamma_j = \eta_j / \widetilde{d}_j \\ \sigma_{j+1} &= \sigma_j - \alpha_j \beta_j, \qquad \tau_{j+1} = \tau_j - \eta_j \gamma_j \\ \mu_{j+1} &= \mu_j - \alpha_j \gamma_j, \end{aligned} \right\}, \quad j = 1, \ldots, n-1,$$

3. $\alpha_n = \sigma_n w_n + \mu_n z_n, \qquad \eta_n = \tau_n z_n + \mu_n w_n,$

$$\widetilde{d}_n = d_n + \alpha_n w_n + \eta_n z_n.$$

From (4.1) and (4.2) we have that

$$L_1^+ = L_1 + L_1 \widetilde{L}(\beta, w, \gamma, z, 0) \quad \text{and} \quad D^+ = \widetilde{D}.$$

Since we can derive an analog of (3.13), the operation count given in the last section applies to this method as well. However, no square roots are required.

When $z = w$ this method becomes equivalent to Method B of Gill and Murray [16] (called Method C1 in [15]).

**5. Special Iterations.** The recurrence of Section 4, although simpler than those of Section 3, can result in the elements of $D$ becoming zero or negative because of rounding errors. Therefore, it should not be used without some strategy to ensure the positivity of the $d_j$. If $d_j$ becomes $< 0$, one might use the strategy proposed by Fletcher [11], that $d_j$ be replaced by the smallest $d_j > 0$ in any previous matrix. A more conservative strategy would be to replace $d_j$ whenever it dropped below some specified $\epsilon > 0$, by the smallest $d_j \geqslant \epsilon$ in any previous matrix $D$. This could be used with the methods of Section 3 as well as with the one of Section 4. Bounding the spectral condition number of $D$ has been suggested by Gill, Murray and Pitfield [17].

Strategies of this kind are quite different from those that periodically reset $H$, and consequently $B$, to the identity matrix. These latter approaches can effect drastic changes in $B$, since all of the information incorporated in $H$ (and $B$) is discarded.

In Recurrence 4 if some $\widetilde{d}_i$, say $\widetilde{d}_j$, becomes nonpositive, the special iteration given below can be used to modify the factors $L_1$ and $D$ so that $L_1 D_1 L_1'$ is positive definite and, if desired, obtain a new approximation to the optimal solution. To simplify matters we will illustrate this special iteration only for the DFP updating formula for $B$ ($\bar{\beta} = 0$ in (3.1)).

*Special Iteration.* Solve

(5.1)                                    $$L_1' p = e_j$$

for $p$, and let $s = tp$, $\bar{x} = x + s$ and $y = \bar{g} - g$. The sign and value of $t$ are chosen so that $s'g < 0$ and $\bar{y}'s > 0$ if $p'g \neq 0$ and $\bar{y}'s > 0$ if $p'g = 0$.

Update $L_1$ and $D$ according to the factorized version of the DFP formula for $B$, and set $x^+ = x$.

Optional: if $f(\bar{x}) < f(x)$, set $x^+ = \bar{x}$.

When $\bar{\beta} = 0$, the matrix (4.2) becomes

(5.2)                                    $$D + \sigma w w' - d_j e_j e_j',$$

where $L_1 w = y$ and $\sigma = 1/\bar{y}'s > 0$.

If $d_i > 0$, $i \neq j$, $u'(D - d_j e_j e_j')u \geqslant 0$. It is equal to zero if, and only if, $u = \alpha e_j$. But in that case

$$u'(\sigma w w')u = \alpha^2 \sigma(w'e_j)^2 = \frac{\alpha^2}{t^2}(\bar{y}'s) > 0,$$

since $\bar{y}'s = \bar{y}'(L')^{-1}L's = tw'e_j > 0$. Therefore, the matrices (5.2) and $B^+$ are positive definite, and have Cholesky factorizations, even though $B$ may not be positive definite (i.e. $d_j \leqslant 0$).

There are several ways in which this special iteration can be used. One way is to apply it directly to the modified factors of $B$ upon first encountering a nonpositive $\tilde{d}_j$ in Recurrence 4. At this point the first $j - 1$ column of $L_1$ and $j$ columns of $D$ have already been updated using Recurrence 4 in conjunction with an analog of (3.13) or (3.16). It is interesting to note that when $\bar{\beta} = 0$ in (3.1), the special iteration only affects the first $j$ columns of $L_1$ and $D$.

An alternate approach is to compute the $\tilde{d}_i$, $\beta_i$, and $\gamma_i$, $i = 1, \ldots, n$, from Recurrence 4 prior to updating $L_1$ and $D$. If a nonpositive $\tilde{d}_j$ is obtained, a special iteration is performed, computing $p$ from (5.1) with $L_1$ replaced by $L_1\tilde{L}$, where $\tilde{L} = \tilde{L}(\beta, w, \gamma, z, e)$ is a special matrix of the form (2.4), $\beta_i = \gamma_i = 0$, $i = j, j + 1, \ldots, n$, and $\beta_i$ and $\gamma_i$, $i = 1, \ldots, j - 1$, are those quantities already computed by Recurrence 4.

One first updates $L_1$ and $D$ to give $\bar{L}_1$ and $\bar{D}$ so that $\bar{B} = \bar{L}_1\bar{D}\bar{L}_1' = L_1DL_1' + L_1\tilde{L}(\sigma w w' - \tilde{d}_j e_j e_j')\tilde{L}L_1'$ where $L_1\tilde{L}w = \bar{y}$, $\sigma = 1/\bar{y}'s > 0$ and $\tilde{d}_j < 0$. One then adds the original correction to $\bar{B}$ and updates $\bar{L}_1$ and $\bar{D}$. This corresponds to adding to $B = L_1DL_1'$ the positive definite rank-two correction term corresponding to the special iteration applied to $L_1\tilde{L}\tilde{D}\tilde{L}'L_1'$ before adding the original correction term. Consequently, the numerical difficulties caused by the closeness of the matrix $B$ + (original correction) to singularity should be avoided.

This approach can be carried out efficiently if use is made of the fact the $\tilde{L}x = y$ and $\tilde{L}'x = y$ can both be solved for $x$ in $O(n)$ operations when $\tilde{L}$ is special. However, an additional $1/2n^2 + O(n)$ operations are required since computing $z$ and updating $L_1$ can no longer be done simultaneously.

**6. Comments and Observations.** (1) All methods presented can be implemented using only $1/2n^2 + O(n)$ storage locations since the computation of the vectors $w^{(j)}$

and $z^{(j)}$, $j = 2, \ldots, n$, and the updating of $L_1$ can be incorporated into Recurrences 2, 3 and 4.

The solution $\widetilde{w}$ of (3.10) is required to determine the step direction $p$ before the recurrences are applied. Computing it using (3.12) or the usual version of forward elimination would increase the operation count by $1/2n^2$. The following observation shows that it can be computed in only $O(n)$ operations except for the first step. If the initial $L_1$ is set to $I$ computing $\widetilde{w}$ on the first step is trivial.

Let $\widetilde{w}^+$ be the solution of $L_1^+\widetilde{w}^+ = -g^+ = -g - y$. From (3.10), (3.11) and (3.6), $L_1\widetilde{L}\widetilde{w}^+ = L_1\widetilde{w} - L_1\widetilde{z}$. Therefore,

$$\widetilde{L}\widetilde{w}^+ = \widetilde{w} - \widetilde{z}.$$

Because of the "special" form of $\widetilde{L}$, this can be solved for $\widetilde{w}^+$ in $O(n)$ operations.

(2) For all members of the one-parameter family of updating formulas (3.2)–(3.4), except for case of $\alpha = 0$, the operations count of $5/2n^2 + O(n)$ aside from square roots, given for Methods 1 and 2 is valid only if one is willing to approximately double the storage requirements to $n^2 + O(n)$. If storage is limited to $1/2n^2 + O(n)$, then this count becomes $3n^2 + O(n)$ since the solution $z$ of $L_1\widetilde{z} = y$ must be computed before either Recurrence 1 or 3 is applied.

If the DFP updating formula for $B$ (i.e. $\alpha = 0$) is used, $\hat{w}$, $\theta_1$ and $\theta_2$ do not depend upon $z$, and Methods 1 and 2 in this special case can be implemented using only $1/2n^2 + O(n)$ storage locations, $5/2n^2 + O(n)$ operations per VM step and, respectively, $2n - 1$ and $n$ square roots.

Implementation of the traditional formulas (1.4) requires $1/2n^2 + O(n)$ locations of storage and $3n^2 + O(n)$ operations per VM step for all cases except the rank-one formula.

(3) The alternate method given in Section 4 can be implemented for all updating formulas (3.1) that in theory preserve positive definiteness, using only $1/2n^2 + O(n)$ storage locations and $5/2n^2 + O(n)$ operations. Unfortunately, one cannot guarantee for this method that the VM matrix $B$ will not become indefinite on account of rounding errors as one can for the methods of Section 3.

(4) Gill and Murray were the first to develop algorithms specifically for updating the Cholesky factors of a VM matrix. As presented in their paper [16], their algorithms require $4n^2 + O(n)$ operations for a full VM step. If forward substitutions are carried out as in (3.12) and use is made of the fact that $\widetilde{L}x = y$ can be solved for $x$ in $O(n)$ operations when $\widetilde{L}$ is "special", then this count can be reduced to $5/2n^2 + O(n)$ operations.

Comments (2) and (3) above apply respectively to Methods A and B of Gill and Murray [16] with two exceptions. (These methods are the rank-one counterparts of Methods 2 and 3.) Method A (two passes per step) requires twice as many square roots as does Method 2 and the updating formula requiring the least combined amount of storage and operations per VM step for Method A is the rank-one formula, rather than the DFP updating formula for $B$ as it is for Method 2. A similar statement holds for Method C5 in [15]. But the rank-one formula does not in general preserve positive definiteness as does the DFP formula. Using $1/2n^2 + O(n)$ storage locations, Method

A requires approximately $1/2n^2$ more operations and $n$ square roots per VM step for the DFP update than does Method 2. This is worth noting since past computational experience indicates that this updating formula outperforms all other members of the one-parameter family of formulas (3.1).

(5) The "composite-$t$" method of Fletcher and Powell [13] is very closely related to Gill and Murray's Method B. It too requires no square roots and can be implemented using only $1/2n^2 + O(n)$ storage locations and $5/2n^2 + O(n)$ operations for all updating formulas (3.1). It is also guaranteed (by a computationally satisfactory, although somewhat artificial device) to keep the matrix $B$ positive definite.

(6) In their paper [16], Gill and Murray recommend that in quasi-Newton methods for solving a system of nonlinear equations, $g(x) = 0$, an orthogonal factorization of the approximate Jacobian of $g(x)$, $B$, be used; i.e.

$$(6.1) \qquad\qquad B = LQ'.$$

The factors $L$ and $Q$ are then recurred rather than $B$ or $B^{-1}$. Gill and Murray [16] give one method for doing this. Typically the modification formula for $B$ has the form $B^+ = B + vu'$ which by (6.1) can be written as $L^+Q^{+'} = L(I + zw')Q'$.

Clearly, the methods of Section 3 can be used to compute the orthogonal factorization $I + zw' = \widetilde{L}\widetilde{Q}'$. Hence, $L^+ = L\widetilde{L}$ and $Q^+ = Q\widetilde{Q}$. Because of the special form of $\widetilde{L}$ and $\widetilde{Q} - \widetilde{Q}$ is the product of a lower and and an upper Hessenberg matrix of the special type shown in (3.3)—the modification of $L$ and $Q$ can be done efficiently. For example, if $B = LQ'$, Broyden's first method [4] can be expressed as

$$B^+ = L\left(I + \frac{\hat{z} - tw}{tw'w}\right)w'Q',$$

where $w$, $p$, and $\hat{z}$ are defined by $Lw = -g$, $p = Qw$, $L\hat{z} = y = g^+ - g$, and $x^+ = x + tp$.

If $Q$ is stored in product form and some of the ideas of Sections 2 and 3 are used, it can be shown that one full step of a factorized version of Broyden's method requires approximately $7/2n^2 + O(n)$ operations and $n$ or $2n - 1$ square roots, depending on which method of Section 2 is used. This is only slightly more computation than the standard method requires.

(7) In many applications the Hessian matrix $G(x)$ is sparse. Since $G^{-1}$ may be full even if $G$ is sparse, one must work with an approximation $B$ to $G$ rather than an approximation $H$ to $G^{-1}$ to take advantage of this sparsity. Sparsity in $B$ can often be translated into sparsity in $L$. Therefore, the methods of this paper may be of some use in algorithms specifically designed to preserve sparsity.

(8) The methods presented here for updating the factorization $L_1DL_1'$ can also be applied to VM methods for linearly constrained optimization problems. Several ways of doing this are described in [21].

of Dr. Philip Wolfe, for providing a stimulating research environment, and to the referees for several useful comments.

City College
City University of New York
New York, New York 10031

1. Y. BARD, "On a numerical instability of Davidon-like methods," *Math. Comp*, v. 22, 1968, pp. 665–666. MR 38 #858.

2. JOHN M. BENNETT, "Triangular factors of modified matrices," *Numer. Math.*, v. 7, 1965, pp. 217–221. MR 31 #1766.

3. K. W. BRODLIE, A. R. GOURLAY & J. GREENSTADT, *Product Form Corrections for Variable Metric Methods of Minimization*, IBM UKSC Report No. 10, IBM-UK Scientific Center, Peterlee, 1972.

4. C. G. BROYDEN, "A class of methods for solving nonlinear simultaneous equations," *Math. Comp.*, v. 19, 1965, pp. 577–593. MR 33 #6825.

5. C. G. BROYDEN, "Quasi-Newton methods and their application to function minimisation," *Math. Comp.*, v. 21, 1967, pp. 368–381. MR 36 #7317.

6. C. G. BROYDEN, "The convergence of a class of double-rank minimization algorithms 2, The new algorithm," *J. Inst. Math. Appl.*, v. 6, 1970, pp. 222–231.

7. W. C. DAVIDON, *Variable Metric Method for Minimization*, A.E.C. Res. and Devel. Rep. ANL-5990 (Rev.), 1959.

8. W. C. DAVIDON, "Variance algorithm for minimization," *Comput. J.*, v. 10, 1967/68, pp. 406–410. MR 36 #4790.

9. L. C. W. DIXON, "Quasi-Newton algorithms generate identical points," *Math. Programming*, v. 2, 1972, pp. 383–387. MR 46 #2852.

10. R. FLETCHER, "A new approach to variable metric algorithms," *Comput. J.*, v. 13, 1970, pp. 317–322.

11. R. FLETCHER, *Fortran Subroutines for Minimization by Quasi-Newton Methods*, Report R-7125, A.E.R.E., Harwell, England, 1972.

12. R. FLETCHER & M. J. D. POWELL, "A rapidly convergent descent method for minimization," *Comput. J.*, v. 6, 1963/64, pp. 163–168. MR 27 #2096.

13. R. FLETCHER & M. J. D. POWELL, *On the Modification of $LDL^T$ Factorizations*, Harwell Report TP. 519, 1973.

14. W. M. GENTLEMAN, *Least Squares Computations by Givens Transformations Without Square Roots*, Res. Rep. CSRR-2062, Univ. of Waterloo, Waterloo, Ont., Canada, 1973.

15. P. E. GILL, G. H. GOLUB, W. MURRAY & M. A. SAUNDERS, "Methods for modifying matrix factorizations," *Math. Comp.*, v. 28, 1974, pp. 505–535. MR 49 #8299.

16. P. E. GILL & W. MURRAY, "Quasi-Newton methods for unconstrained optimization," *J. Inst. Math. Appl.*, v. 9, 1972, pp. 91–108. MR 45 #9456.

17. P. E. GILL, W. MURRAY & R. A. PITFIELD, *The Implementation of Two Revised Quasi-Newton Algorithms for Unconstrained Optimization*, NPL Report NAC 11, 1972.

18. D. GOLDFARB, "A family of variable-metric methods derived by variational means," *Math. Comp.*, v. 24, 1970, pp. 23–26. MR 41 #2896.

19. D. GOLDFARB, *Variable Metric and Conjugate Direction Methods in Unconstrained Optimization: Recent Developments*, ACM Proceedings–1972.

20. D. GOLDFARB, *Factorized Variable Metric Methods for Unconstrined Optimization*, IBM Res. Rep. RC 4415, IBM Research Center, Yorktown Heights, New York, 1973.

21. D. GOLDFARB, "Matrix factorizations in optimization of nonlinear functions subject to linear constraints," *Math. Programming*, v. 10, 1976, pp. 1–31.

22. G. P. McCORMICK & J. D. PEARSON, "Variable metric methods and unconstrained optimization," *Optimization* (Sympos., Univ. Keele, 1968), (R. Fletcher, Editor), Academic Press, London, 1969, pp. 307–325. MR 42 #2806.

23. B. A. MURTAGH & R. W. H. SARGENT, "A constrained minimization method with quadratic convergence," *Optimization* (Sympos., Univ. Keele, 1968), (R. Fletcher, Editor), Academic Press, London, 1969, pp. 215–246. MR 44 #1442.

24. M. J. D. POWELL, "On the convergence of the variable metric algorithm," *J. Inst. Math. Appl.*, v. 7, 1971, pp. 21–36. MR 43 #5698.

25. D. SHANNO, "Conditioning of quasi-Newton methods for function minimization," *Math. Comp.*, v. 24, 1970, pp. 647–656. MR 42 #8905.

26. P. WOLFE, *Another Variable Metric Method*, IBM working paper.