

## On Variable-Metric Methods for Sparse Hessians\*

By D. F. Shanno

**Abstract.** The relationship between variable-metric methods derived by norm minimization and those derived by symmetrization of rank-one updates for sparse systems is studied, and an analogue of Dennis's nonsparse symmetrization formula derived. A new method of using norm minimization to produce a sparse analogue of any nonsparse variable-metric method is proposed. The sparse BFGS generated by this method is tested against the sparse PSB and variable-memory conjugate gradient methods, with computational experience uniformly favoring the sparse BFGS.

**I. Introduction.** Extensive analysis, both computational and theoretical, has verified that variable-metric algorithms are highly successful methods for minimization of nonlinear functions, both unconstrained and subject to both linear and nonlinear constraints. As expertise in solving these problems has grown, problem size, determined by the number of variables, has also grown, leading to storage problems on digital computers. This arises from the fact that variable-metric algorithms generate a sequence of approximations  $x_k$  to the minimizer  $\hat{x}$  of a nonlinear function  $f(x)$ ,  $x$  an  $n$  vector, by

$$(1) \quad x_{k+1} = x_k - c_k H_k g_k, \quad H_{k+1} = H_k + D_k,$$

where  $H_k$  is an  $n \times n$  approximation to the inverse Hessian matrix,  $g_k = \nabla F(x_k)$ , the gradient of  $f$  at  $x_k$ ,  $c_k$  is an appropriately chosen scalar, and  $D_k$  is chosen to assure that  $H_{k+1}$  satisfies the quasi-Newton equation

$$(2) \quad H_{k+1} y_k = p_k, \quad y_k = g_{k+1} - g_k, \quad p_k = x_{k+1} - x_k.$$

Historically, the first known choice of  $D_k$  was the DFP (Davidon [4], Fletcher and Powell [8]). Dropping the subscript  $k$ , and replacing the subscript  $k + 1$  with the superscript \*, the DFP update is defined by

$$(3) \quad H^* = H + \frac{pp'}{p'y} - \frac{Hy y'H}{y'Hy}.$$

In recent years, the DFP has been supplanted by the BFGS (Broyden [2], Fletcher [9], Goldfarb [10], and Shanno [20]) update, defined by

$$(4) \quad H^* = H - \frac{Hy p' + p y'H}{p'y} + \left(1 + \frac{y'H y}{p'y}\right) \frac{pp'}{p'y}.$$

---

Received September 11, 1978; revised November 20, 1978 and May 11, 1979.

AMS (MOS) subject classifications (1970). Primary 65F30; Secondary 15A24.

\*This research was supported by the National Science Foundation under research grant MCS 77-07327.

While recent numerical experience suggests that for nonsparse problems the BFGS is superior to all the known variable-metric methods (Shanno-Phua [22]), for sparse problems to other updates, the PSB (Powell [17], Broyden [1]) defined by

$$(5) \quad H^* = H - \{ \eta \eta' (p' H p) - [\eta p' H + H p \eta'] p' H y + H p p' H (\eta' y) \} / \{ (\eta' y) (p' H p) - (p' H y)^2 \},$$

$$\eta = H y - p,$$

and the SR1 (Fiacco-McCormick [7]) defined by

$$(6) \quad H^* = H + \frac{\eta \eta'}{\eta' y},$$

attain increased importance.

The difficulty on large problems is that all four updates (3)–(6) require  $n^2/2$  memory locations, which often becomes impractical as  $n$  increases. If the true Hessian of  $f$  is nonsparse, the only alternative is to replace a variable-metric algorithm with a conjugate gradient algorithm; see, for example, Shanno [21]. If, however, the true Hessian matrix of  $f$  has a known sparsity pattern, and an approximation to the Hessian rather than the inverse Hessian is stored with only the known nonzero elements approximated, the problem may become computationally tractable. Denoting  $H^{-1} = B$ , the analogues of (3)–(6) can easily be determined as

$$(7) \quad B_{\text{DFP}}^* = B - \frac{B p y' + y p' B}{p' y} + \left( 1 + \frac{p' B p}{p' y} \right) \frac{y y'}{p' y},$$

$$(8) \quad B_{\text{BFGS}}^* = B + \frac{y y'}{p' y} - \frac{B p p' B}{p' B p},$$

$$(9) \quad B_{\text{PSB}}^* = B + \frac{\gamma p' + p \gamma'}{p' p} - \frac{\gamma' p}{p' p} \frac{p p'}{p' p}, \quad \gamma = y - B p,$$

and

$$(10) \quad B_{\text{SR1}}^* = B + \frac{\gamma \gamma'}{\gamma' p}.$$

Note that the analogue of (2), satisfied by (7)–(10) is

$$(11) \quad B^* p = y.$$

Clearly, for sparse problems, an approximation to the Hessian rather than inverse Hessian must be computed, as sparsity is not necessarily preserved in the inverse of a sparse matrix.

Schubert [19] initially proposed a method for updating approximate Jacobian matrices of nonlinear systems of equations which was a sparse analogue of Broyden's variable-metric method for nonlinear systems. Briefly, Broyden [1] proposed the update formula

$$(12) \quad B^* = B + \frac{(y - B p) p'}{p' p},$$

and Schubert noted that this could be decomposed into row-by-row updating by

$$(13) \quad B^* = B + \sum_{i=1}^n e_i e_i' \frac{(y - Bp)p_i'}{p_i' p},$$

where  $e_i$  is the unit vector with 1 as the  $i$ th component. He then noted that if we denote by  $p_i$  the vector

$$(14) \quad p_i = D_i p,$$

where  $D_i$  is the diagonal matrix with the  $j$ th diagonal component 1 if  $B_{ij}$  is nonzero, 0 if  $B_{ij}$  is zero, then the update

$$(15) \quad B^* = B + \sum_{i=1}^n e_i e_i' \frac{(y - Bp)p_i'}{p_i' p}$$

satisfies (11) and has the desired sparseness. The method has proved quite successful for nonlinear equations, but has the drawback for optimization problems that the matrix  $B^*$  is nonsymmetric.

In recent papers, Toint [24] and Marwil [14] addressed the problem of generating sparse update formulas with symmetric Hessians. Toint used variational means, and Marwil the sequence of symmetrizations of Powell to derive the update formula

$$(16) \quad B^* = B + \sum_{i=1}^n \lambda_i (e_i p_i' + p_i e_i'),$$

where the  $\lambda_i$ 's satisfy the system of linear equations

$$(17) \quad Q\lambda = y - Bp,$$

where

$$(18) \quad Q = \sum_{i=1}^n p_i' p_i e_i e_i' + e_i' p_i p_i e_i'.$$

Inspection verifies that  $Q$  has the same sparsity pattern as  $B$  if the diagonal elements of  $B$  are updated at each step, and Toint shows that  $Q$  is positive definite. We note here that the restriction that  $B$  always updates diagonal elements, imposed by Toint, is required only to ensure that  $Q$  and  $B$  have identical structures. While in any minimization problem it is assumed  $B$  will have strictly positive diagonal elements, these may be known constants. Thus, if  $B_0$  has these correctly inserted along the diagonal, one will never wish to update them; and thus, for future updates they will be treated identically to zero elements. The formulas (16)–(18) can accommodate this quite easily, with the above noted exception that  $Q$  must have more elements than the correction matrix, but not more than  $B$ .

The above argument, somewhat modified, can be extended to known constant off-diagonal elements as well, as these may also be fixed and considered as zeros for future updating. In this case, these will not affect the size of  $Q$ ; and  $Q$  may actually contain less nonzero elements than  $B$ .

As noted, Toint derived his update formula variationally in a manner introduced by Greenstadt [12], while Marwil derived his via symmetrization. Section II will show

how to generate symmetric sparse updates by a sparse analogue of a symmetrization method introduced by Dennis [5] for nonsparse matrices. A numerical instability arising in general sparse symmetrization will be discussed.

Section III explores an alternative means of using norm minimization to derive a sparse analogue of any known nonsparse variable-metric method by computing a normal, nonsparse update and then finding the closest sparse matrix in the Frobenius norm, and also shows why positive definiteness can not necessarily be maintained.

Section IV discusses variable-memory conjugate gradient methods as stable alternatives to sparse variable-metric methods. Finally, Section V contains numerical tests which show great computational promise for the sparse BFGS derived in Section III.

As a final note on this section, Goldfarb [11] recently proposed an entirely different approach to sparse variable-metric updating based on updating the Cholesky decomposition of the matrix  $B$ . Unfortunately, the Cholesky decomposition of a sparse matrix is not necessarily sparse, as is seen by the following example which demonstrates the well-known difficulty of fill-in which may occur when applying Gaussian elimination to sparse matrices.

Let

$$B = \begin{bmatrix} 4 & 1 & 1 & 1 \\ 1 & 4 & 0 & 0 \\ 1 & 0 & 4 & 0 \\ 1 & 0 & 0 & 4 \end{bmatrix}.$$

Then if

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/4 & 1 & 0 & 0 \\ 1/4 & -1/15 & 1 & 0 \\ 1/4 & -1/15 & -1/14 & 1 \end{bmatrix},$$

and

$$D = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 15/4 & 0 & 0 \\ 0 & 0 & 56/15 & 0 \\ 0 & 0 & 0 & 26/7 \end{bmatrix},$$

then computation verifies

$$B = LDL',$$

and  $L$  preserves none of the sparsity of  $B$ . In view of this, Goldfarb's method will not be further considered here, for this paper is concerned solely with storage minimization by exploiting known sparsity. Clearly, if the  $LDL'$  factorization of a sparse matrix is unacceptably nonsparse, Gaussian elimination cannot be used to solve linear equations with

either  $Q$  or  $B$ . Rather, either a conjugate gradient method, an iterative method, or some alternative sparsity-preserving solution technique will have to be used.

**II. Symmetrization for Sparse Systems.** Dennis [5] considered the problem of symmetrizing rank-one updates of the form

$$(19) \quad D^* = \frac{(p - Hy)z'}{z'y}$$

by

$$(20) \quad D = D^* + D^{*'} - \theta zz',$$

where  $\theta$  is chosen to ensure (2) is satisfied.

While (20) is a correction term for the inverse Hessian, analogous updates for the Hessian can be determined as

$$(21) \quad \hat{B} = B + \frac{(y - Bp)z'}{z'p}$$

and

$$(22) \quad B^* = B + \frac{(y - Bp)z' + z(y - Bp)'}{z'p} - \frac{(y - Bp)'pzz'}{z'p} = \frac{1}{2}(\hat{B} + \hat{B}') - \theta zz'.$$

Corresponding to (7)–(10), Schnabel [18] has shown that the choice  $z = y\sqrt{p'Bp/p'y} + Bp$  in (22) yields the BFGS, while various authors have shown that  $z = y$ ,  $z = y - Bp$ , and  $z = p$  yield, respectively, the DFP, SR1, and PSB updates.

We now examine how a sparse analogue of (25) can be developed. Marwill [14] derives the sparse PSB as the limit of an infinite sequence of matrices which are alternately symmetric or which satisfy (11), all of which possess the desired sparsity. An alternative method for generating a general sparse symmetric update formula, analogous to (22), can be derived by a reexamination of (22) decomposed in the manner of Schubert.

Recalling (13), we first note that any variable metric-update can be decomposed into the sum of its rows. For example, the BFGS update can be written as

$$(23) \quad B^* = B + \sum_{i=1}^n e_i e_i' \frac{yy'}{p'y} - \sum_{i=1}^n e_i e_i' \frac{Bpp'B}{p'Bp} = B + \sum_{i=1}^n e_i \left( \frac{e_i'y}{p'y} y' - \frac{e_i'Bp}{p'Bp} p'B \right).$$

A general form of (23) can be defined to be

$$(24) \quad B^* = B + \sum_{i=1}^n e_i w_i'$$

Note that in (23),  $B^*$  is symmetric due to the choice of  $w_i$ 's, but for general  $w_i$ 's, (24) may not be. However, an interesting variant of Dennis's symmetrization may be

applied to (24). Let

$$(25) \quad B^* = B + \sum_{i=1}^n \beta_i(e_i w'_i + w_i e'_i),$$

where the  $\beta_i$ 's are chosen to assure  $B^*p = y$ . Clearly, this update is the most general possible, as defining

$$(26) \quad w_i = \begin{pmatrix} 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \\ \cdot \\ w_{i,j+1} \\ \cdot \\ w_{in} \end{pmatrix}$$

allows for any symmetric matrix to be represented by the correction term of (25).

An interesting relationship between (25) and Dennis's derivation of (22) exists.

Again, let

$$(27) \quad \hat{B} = B + \frac{(y - Bp)z'}{z'p}$$

and

$$(28) \quad \hat{B}_{SYM} = B + \frac{(y - Bp)z' + z(y - Bp)'}{z'p}.$$

Clearly (28) is symmetric, but fails to satisfy (11). Rewriting (29) as (13), and symmetrizing as in (28), we can write

$$(29) \quad \begin{aligned} \hat{B}_{SYM} &= B + \sum_{i=1}^n \left( e_i e'_i \frac{(y - Bp)z'}{z'p} + \frac{z(y - Bp)'}{z'p} e_i e'_i \right) \\ &= B + \sum_{i=1}^n \zeta_i (e_i z' + z e'_i), \quad \zeta_i = \frac{(y - Bp)' e_i}{z'p}. \end{aligned}$$

Now as  $\hat{B}_{SYM}$  fails to satisfy (11), we modify the  $\zeta_i$ 's to ensure that (11) is satisfied as follows:

Let

$$(30) \quad B^* = B + \sum_{i=1}^n (\zeta_i + \delta_i)(e_i z' + z e'_i),$$

when the  $\delta$ 's are to be chosen to satisfy (11). Then

$$(31) \quad \begin{aligned} B^*p &= Bp + (y - Bp) + (z'p)\delta + (\zeta'p + \delta'p)z \\ &= y + (z'p)\delta + (\zeta'p + \delta'p)z, \end{aligned}$$

where

$$\delta = \begin{pmatrix} \delta_1 \\ \cdot \\ \delta_n \end{pmatrix}, \quad \zeta = \begin{pmatrix} \zeta_1 \\ \cdot \\ \zeta_n \end{pmatrix}.$$

Now since (11) must be satisfied, (31) yields

$$(32) \quad \delta = -\frac{(\zeta'p + \delta'p)z}{(z'p)},$$

or  $\delta = rz$  for some constant  $r$ . Substituting in (32) and solving for  $r$ , we obtain

$$(33) \quad r = -\frac{1}{2} \frac{\zeta'p}{z'p},$$

and by (29) and (30)

$$(34) \quad B^* = \hat{B}_{SYM} - \frac{\zeta'p}{z'p} zz' = \hat{B}_{SYM} - \frac{(y - Bp)'p}{z'p} \frac{zz'}{z'p}.$$

But (34) is exactly (22), and hence Dennis's symmetrization and the method (30) are identical for nonsparse systems.

We also note that (30) is completely equivalent to

$$(35) \quad B^* = B + \sum_{i=1}^n \alpha_i (e_i z' + z e_i'),$$

where  $\alpha$  is chosen so that  $B^*p = y$ . This follows trivially from the fact that if  $\delta$  is the solution to (11) for (30) and  $\alpha$  is the solution for (35),  $\alpha = \delta + \zeta$  by the linearity of this system. Also, we note that  $B^*$  is independent of the length of  $z$ , as is  $B^*$  defined by (22), again due to the linearity of the system.

Thus, from the above analysis, we can easily extend Dennis's symmetrization to sparse Hessians using Schubert's method of guaranteeing sparseness.

In the manner of (15), we note that the nonsymmetric update  $\hat{B}$  can be written

$$(36) \quad \hat{B} = B + \sum_{i=1}^n \alpha_i e_i z'_i, \quad \alpha_i = \frac{e_i(y - Bp)}{p'_i p}, \quad z_i = D_i z;$$

and we thus symmetrize exactly as above by

$$(37) \quad B^* = B + \sum_{i=1}^n \alpha_i (e_i z'_i + z_i e'_i),$$

where again the  $\alpha_i$ 's are chosen to satisfy (11). By the symmetry of  $B^*$ ,  $B^*$  clearly satisfies the sparseness requirements whenever each  $z_i$  satisfies the sparseness requirements for the  $i$ th row (or column).

Further,  $\alpha$  satisfies

$$(38) \quad S\alpha = y - Bp,$$

where

$$(39) \quad S = \sum_{i=1}^n z'_i p e_i e'_i + e'_i p z_i e'_i$$

which reduces precisely to the system (17)–(18) if  $z_i = p_i$  is substituted. Thus, the sparse PSB can be derived directly by symmetrization, as well as by Marwil's symmetrization and by norm minimization, as can a larger class of updates.

While symmetrization in this form appears to provide an attractive means of generating a symmetric analogue to any known variable-metric method, a serious difficulty arises from the fact that while it is not obvious in the form of Eq. (39), there is an implied denominator in the update matrices which is implicitly defined in the  $\alpha_i$ 's. For example, for the nonsparse BFGS, before symmetrizing (21) we have in the denominator  $p'(\gamma\sqrt{p'Bp/p'y} + Bp)$ . While linear search criteria for nonsparse problems guarantee that this is significantly different from zero, when sparsity is introduced, this is no longer necessarily so. The PSB, however, has an implied denominator of  $p'_i p$ , which is positive, and in fact maximizes the denominator over all possible  $z$  given the sparsity conditions. In practice, the sparse BFGS derived directly from (37) proved totally unsatisfactory due entirely to this numerical instability, while the sparse PSB performed reasonably. This feature casts serious doubt as to whether symmetrization techniques can prove suitable for the generation of sparse variable-metric methods, and the problem of finding suitable weighting matrices make it doubtful if norm minimization can be used to derive in a reasonable fashion sparse analogues to the BFGS of DFP.

In view of this, the next section examines an alternative way to use norm minimization to generate sparse analogues of any nonsparse variable-metric method.

**III. The New Method.** In this section we show a new way to use norm minimization to generate a sparse analogue to any variable-metric update, and the computational results of Section V will demonstrate that the sparse BFGS generated in this fashion has superior performance to the sparse PSB update generated by minimizing the norm of the correction matrix.

We now consider the problem of generating a sparse analogue to the BFGS update, and note that the technique developed for this will apply to any variable-metric update in a straightforward manner.

We first note that even if  $B$  has the desired sparsity pattern,  $B^*$  defined by (8) will not. In order to modify  $B^*$  so that it has the desired sparsity, we first denote by  $K$  the set of ordered pairs of integers  $(i, j)$  such that if  $(i, j) \in K$ ,  $B_{ij} = 0$ . We then define

$$(40) \quad \hat{B}^* = B^* + E,$$

and consider the problem

$$(41) \quad \text{minimize } \|E\| = \frac{1}{2} \text{Tr}(E'E)$$

subject to

$$(42a) \quad Ep = 0,$$

$$(42b) \quad E_{ij} = -B_{ij}^*, \quad (i, j) \in K,$$

$$(42c) \quad E = E'.$$

Clearly, any solution to (41) possesses the desired sparsity pattern, and is the



closest sparse matrix to the BFGS in the Frobenius norm. In order to solve (41) we define the Lagrangian  $\phi$  by

$$\begin{aligned}
 \phi(E, \mu, \Lambda, \lambda) &= \frac{1}{2}\text{Tr}(E'E) - \text{Tr}(Epu') - \text{Tr}(\Lambda(E - E')) \\
 &\quad - \sum_{(i,j) \in K} \lambda_{ij} \text{Tr}(E + B^*)e_j e_i' \\
 (43) \qquad \qquad \qquad &= \frac{1}{2}\text{Tr}(E'E) - \text{Tr}(Epu') - \text{Tr}(\Lambda(E - E')) \\
 &\quad - \text{Tr}(\Delta'(E + B^*)),
 \end{aligned}$$

where  $\Delta$  is a matrix which is 0 if  $(i, j) \notin K$ , and  $\lambda_{ij}$  if  $(i, j) \in K$ .

Differentiating (43), we obtain

$$(44) \qquad \qquad \qquad \frac{\partial \phi}{\partial E} = E - \mu p' + \Delta + \Lambda - \Lambda' = 0,$$

or

$$(45) \qquad \qquad \qquad E = \mu p' + \Delta + \Lambda - \Lambda',$$

and

$$(46) \qquad \qquad \qquad E' = p\mu' + \Delta' + \Lambda' - \Lambda.$$

Utilizing (42c), we obtain

$$(47) \qquad \qquad \qquad E - E' = \mu p' - p\mu' + \Delta - \Delta' + 2(\Lambda - \Lambda') = 0$$

or

$$(48) \qquad \qquad \qquad (\Lambda - \Lambda') = \frac{1}{2}(-\mu p' + p\mu' + \Delta' - \Delta),$$

and

$$(49) \qquad \qquad \qquad E = \frac{1}{2}(\mu p' + p\mu' + \Delta' + \Delta).$$

Now we use (42b) to obtain

$$(50) \qquad \qquad \qquad e_i' E e_j = \frac{1}{2}(e_i' \mu p' e_j + e_i' p \mu' e_j + \lambda_{ij} + \lambda_{ji}) = -B_{ij}^*,$$

and

$$(51) \qquad \qquad \qquad \lambda_{ij} + \lambda_{ji} = -2B_{ij}^* - e_i' \mu p' e_j - e_i' p \mu' e_j, \quad (i, j) \in K.$$

Equation (51) can be written in matrix form as

$$(52) \qquad \qquad \qquad \Delta + \Delta' = -2B_K^* - \sum_{i=1}^n e_i e_i' (\mu \hat{p}_i' + p \hat{\mu}_i'),$$

where  $B_K^* = B^*$ ,  $(i, j) \in K$ ,  $B_K^* = 0$  otherwise, and  $\hat{p}_i = \bar{D}_i p$ ,  $\hat{\mu}_i = \bar{D}_i \mu$ , where  $\bar{D}_i$  is the diagonal matrix which is the complement of  $D_i$ , i.e.  $\bar{D}_i$  has diagonal 1's where  $D_i$  has 0's and vice versa.

We now note from (49) and (52) that

$$\begin{aligned}
 (53) \quad E &= \frac{1}{2} \left( \sum_{i=1}^n e_i e'_i (\mu p' + p \mu') - 2\hat{B}_K^* - \sum_{i=1}^n e_i e'_i (\mu \hat{p}'_i + p \hat{\mu}'_i) \right) \\
 &= \frac{1}{2} \left( \sum_{i=1}^n e_i e'_i (\mu p'_i + p \mu'_i) - 2\hat{B}_K^* \right),
 \end{aligned}$$

where  $p_i = D_i p$ ,  $\mu_i = D_i \mu$ .

Then by (42a)

$$(54) \quad E p = \frac{1}{2} \left( \sum_{i=1}^n e_i e'_i (\mu p'_i + p \mu'_i) - 2\hat{B}_K^* \right) p = 0,$$

and

$$(55) \quad \sum e_i e'_i (\mu p'_i p + p \mu'_i p) = 2\hat{B}_K^* p.$$

But (55) can be rewritten as

$$(56) \quad \sum_{i=1}^n \mu_i (e_i p'_i + p_i e'_i) p = 2\hat{B}_K^* p,$$

which is a system of linear equations in the Lagrange multipliers  $\mu_i$ . Further, the coefficient matrix is exactly Toint's coefficient matrix [24], which has the same sparsity pattern as  $B$  and is positive definite. Thus, (56) can be solved for the  $\mu_i$ 's exactly as in Toint's method.

We now note that the full matrix  $B^*$  need never be stored. Clearly, the nonzero elements of  $B^*$  must be computed and stored in  $\hat{B}^*$  so that (40) may be used for updating. The elements which correspond to zeros of  $\hat{B}^*$  must be computed, but need not be stored, as (56) demonstrates that all that need be stored is the vector  $r$  defined by

$$(57) \quad r = 2\hat{B}_K^* p.$$

Thus, as each element of  $B_K^*$  is computed,  $r$  is updated, and the element immediately discarded, and the method uses no more storage than the sparse PSB. Also, note that once (56) has been solved for the  $\mu_i$ 's,  $\hat{B}^*$  is defined by

$$(58) \quad \hat{B}^* = \bar{B}^* + \sum_{i=1}^n \mu_i (e_i p'_i + p_i e'_i),$$

where  $\bar{B}_{ij}^* = B_{ij}^*$ ,  $(i, j) \notin K$ ,  $\bar{B}_{ij}^* = 0$  otherwise. Thus, at no time is any nonsparse matrix required to be stored.

In a private communication, R. Schnabel has pointed out that the BFGS so derived can be derived using the least norm projection technique of Dennis and Schnabel [6], and thus updates of the form proposed here fall into the general category of least norm updates.

The update formula (58) was derived using an unweighted Frobenius norm. If a positive definite weighting matrix  $W$  is used instead, so that we minimize  $\text{Tr}(WE'WE)$ ,

analogous to (49), we obtain

$$(59) \quad E = \frac{1}{2}(z(p'M) + (Mp)z' + M(\Delta + \Delta')M),$$

where  $M = W^{-1}$  and  $z = Mu$ . This yields the solution

$$(60) \quad \hat{B}^* = \bar{B}^* + \sum_{i=1}^n z_i(e_i(Mp)'_i + (Mp)_i e'_i), \quad (Mp)_i = \bar{D}_i(Mp),$$

$z_i$  solving the system

$$\sum_{i=1}^n z_i(e_i(Mp)'_i + (Mp)_i e'_i)p = 2B_K^*p,$$

if and only if  $M(\Delta + \Delta')M$  has the identical sparsity pattern to  $\Delta + \Delta'$ , as will happen when, for instance,  $M$  is diagonal. Otherwise, no straightforward solution for  $\Delta$  and  $z$  is known.

Experimental results in this work have used only  $W = I$ . It is possible that different choices of  $W$  may produce better updates, but it is not clear how an appropriate  $W$  might be chosen. This remains for further study.

In all events, choosing  $W = I$  here still yields a matrix which reduces to the BFGS matrix when  $B$  is nonsparse, whereas using  $W = I$  in Toint's method reduces to the PSB update when  $B$  is nonsparse. In view of the known superiority of the BFGS update, this appears to be a more reasonable means of using norm minimization to obtain sparse updates, and the computational results of Section V bear this out.

Analytically, this method is also appealing, as both  $B^*$  and  $\hat{B}^*$  behave identically when multiplied by the vector  $p$ , as  $Ep = 0$  is a condition imposed on  $E$  to ensure that the quasi-Newton equation  $\hat{B}^*p = y$  is satisfied. As  $p$  is the direction in which we have obtained information about the function, this means that  $\hat{B}^*$  and  $B^*$  both retain the same information along this direction.

Finally, we note that unlike the BFGS, the matrix  $\hat{B}^*$  need not be positive definite, nor need it minimize a quadratic in  $n$  dimensions in at most  $n$  updates with exact line searches. Indeed, no general sparse variable-metric update which satisfies  $\hat{B}^*p = y$  can be guaranteed to be positive definite.

To see this, let  $f(x, y) = -e^{-x^2} + y^2$ , and let  $B$  be diagonal. Then if  $B_0 = I$ , and exact searches (to four decimal places) are used, with  $x_0 = 1, y_0 = 100$ ,

$$(61) \quad p = \begin{pmatrix} -.3679 \\ -100.0 \end{pmatrix} \quad \text{and} \quad y = \begin{pmatrix} .1120 \\ -200.0 \end{pmatrix}.$$

Thus, the only diagonal  $\hat{B}^*$  satisfying  $\hat{B}^*p = y$  is

$$(62) \quad \hat{B}^* = \begin{bmatrix} -.3044 & 0 \\ 0 & 2 \end{bmatrix},$$

which is clearly indefinite. Thus, if positive definite updates are desired,  $\hat{B}^*p = y$  must be abandoned.

As the results of Section V show, the indefiniteness of  $B^*$  appears to pose little problem for unconstrained problems, and maintaining the quasi-Newton condition

while modifying search strategy should prove sufficient. If a positive definite matrix is desired, however, as for example as the Hessian of a quadratic object function for a quadratic programming algorithm, some means of suitably modifying  $B^*$ , such as adding positive elements to the diagonal matrix must be devised. Acceptable methods of modification remain for further research.

**IV. Variable Memory Conjugate Gradient Algorithms.** In an attempt to conserve storage while maintaining a positive definite approximation to the Hessian while satisfying the quasi-Newton equation, a variable-storage conjugate gradient method proposed by Buckley [3], using the conjugate gradient vector proposed by Shanno [21], has been programmed and tested.

Basically, the method stores two past vectors  $p_t$  and  $y_t$ , and for  $j \neq t$  the general conjugate gradient vector is calculated by

$$(63) \quad \hat{H} = \frac{p'_t y_t}{y'_t y_t} \left( I - \frac{p_t y'_t + y_t p'_t}{p'_t y_t} + \frac{y'_t y_t}{p'_t y_t} \frac{p_t p'_t}{p'_t y_t} \right) + \frac{p_t p'_t}{p'_t y_t}$$

$$(64) \quad H^* = \hat{H} - \frac{p_j y'_j \hat{H} + \hat{H} y_j p'_j}{p'_j y_j} + \left( 1 + \frac{y'_j \hat{H} y_j}{p'_j y_j} \right) \frac{p_j p'_j}{p'_j y_j},$$

$$(65) \quad p^* = -H^* g^*.$$

In [21] it is shown how  $p^*$  can be calculated so that no matrices are stored. Further,  $p^*$  is clearly a descent direction determined as a variable-metric algorithm with information from only two past points. When  $j = t$ , only (63) is used in calculating (65).

Buckley [3] suggests keeping  $m$  past points, where  $m$  is determined by storage availability. He notes storing  $p_j$  and  $H_j y_j$  as well as the scalars  $p'_j y_j$  and  $y'_j H_j y_j$  for each point allows the recursive calculation for  $H y$  and  $-H^* g^*$  defined by

$$(66) \quad H y = \hat{H} y + \sum_{i=1}^m - \frac{p'_i y}{p'_i y_i} H_j y_i + \left( 1 + \frac{y'_i H_j y_i}{p'_i y_i} \frac{p'_i y}{p'_i y_i} - \frac{y'_i H_j y_i}{p'_i y_i} \right) p_i$$

and

$$(67) \quad -H g^* = -\hat{H} g^* + \sum_{i=1}^m \frac{p'_i g^*}{p'_i y_i} H_j y_i - \left( 1 + \frac{y'_i H_j y_i}{p'_i y_i} \frac{p'_i g^*}{p'_i y_i} - \frac{y'_i H_j g^*}{p'_i y_i} \right) p_i$$

with

$$(68) \quad -H^* g^* = -H g^* + \frac{p'_i g^*}{p'_i y} H y - \left( 1 + \frac{y'_i H y}{p'_i y} \frac{p'_i g^*}{p'_i y} - \frac{y'_i H g^*}{p'_i y} \right) p.$$

Clearly, as  $m$  increases, a better approximation to the inverse Hessian occurs, and the convergence rate should improve.

A difficulty with the scheme occurs when  $m$  past vectors have been stored. At this point, to store information at a more recent point, information at a past point must be discarded. However, doing this changes the matrix  $H_i$  for all  $i$ ,  $i = 1, \dots, m$ , and thus the remaining stored vectors  $H_i y_i$  no longer contain correct information, as the

$H_i$ 's have been altered. Correcting these requires the vectors  $y_i$ , which have not been stored. Thus once  $m$  vectors have been stored, all must be discarded and the method restarted.

It should be noted that this can be overcome by storing  $p_i$  and  $y_i$ , rather than  $H_i y_i$  at each step. In this case, however, the complexity of recalculating  $H_i y_i$  at each step becomes large, and can dominate the computation time for the algorithm. As function and gradient calculations would have to be extremely expensive in order to make this computation practicable, it was not tested here. Rather, the scheme described above, with  $m$  chosen to make the storage requirements of the sparse variable-metric methods and the conjugate gradient methods as nearly identical as possible was implemented. Buckley [3] also discards all prior information whenever the storage limit has been reached, justifying this strategy on the grounds of numerical stability.

**V. Computational Results.** Five algorithms were coded and tested on six test problems, each problem with varying numbers of variables. The five algorithms tested were the sparse BFGS described in this paper (SBFGS), the sparse PSB (SPSB), a pure BFGS, the conjugate gradient method described in [21] (CONJ), and the conjugate gradient method with memory (CONJM) described in Section IV.

Both sparse variable-metric updates used the step length determination algorithm of MINI02 [23]. As the direction need not be a downhill direction, whenever  $p'g > 0$ ,  $-p$  was used rather than  $p$ . This is not necessarily the most satisfactory means of obtaining a downhill direction at each step. Toint [25] recommends using Hebden's [13] method for assuring that a descent direction is found. Moré and Sorensen [16] suggest another means of obtaining descent directions when an  $LDL'$  decomposition of  $B$  is available. As noted, an  $LDL'$  decomposition need not preserve sparsity, but when it does, a variant of the Moré-Sorensen scheme, either with the BFGS update described herein factored in an  $LDL'$  form or Goldfarb's sparse BFGS method [11] may prove practical.

However, as the simple scheme of simply reversing direction when an uphill direction is determined worked so well on all the problems tested it should be reasonable for a large number of problems, and the more expensive overhead required by the more sophisticated schemes would appear to be indicated only for particularly difficult problems.

The BFGS algorithm is MINI02, documented in [23]. It does not use sparsity and was run simply to determine if allowing for sparseness improved or hurt the rate of convergence.

The two conjugate gradient algorithms both do one forced cubic interpolation at each step, thus providing  $n$  step termination on quadratic problems and maintaining a superlinear rate of convergence.

For the three variable-metric algorithms, convergence was attained when each element of the gradient was less than .00001 in absolute value. For the conjugate gradient algorithms, convergence was attained when  $g'g \leq (.00001)^2 (\max(1, x'x))$ . While these are not totally equivalent, on the problems tested they are so close as to make no difference in interpreting the results.

The six test functions were a tridiagonal quadratic function (TRIDIA) defined by

$$(69) \quad f(x) = \sum_{i=1}^n i(2x_i - x_{i-1})^2$$

with starting point  $x_0 = (1, \dots, 1)$ , Toint's modified Rosenbrock (TROSEN) function [11], the extended Rosenbrock (EROSEN) function defined by

$$(70) \quad f(x) = \sum_{i=1}^n 100(x_i - x_{i-1})^2 + (1 - x_1)^2$$

with initial estimates  $x_0 = (-1, \dots, -1)$ , Toint's [25] variant on Broyden's seven diagonal function (TBROY), the boundary value function (with five diagonals) defined by Moré, Garbow, and Hillstrom [15] (BV), and a nondiagonal variant of Rosenbrock's function (NONDIA) defined by

$$(71) \quad f(x) = \sum_{i=2}^n 100(x_1 - x_i^2)^2 + (1 - x_i)^2,$$

with starting estimates  $x_0 = (-1, \dots, -1)$ .

The results are contained in Table I, where ITER is the number of iterations, IFUN the number of function and gradient evaluations,  $M$  the number of nondescent directions found, and  $K$  the number of past points stored.

TABLE I

	SBFGS			SPSB			BFGS		CONJ		CONJM		K
	ITER	IFUN	M	ITER	IFUN	M	ITER	IFUN	ITER	IFUN	ITER	IFUN	
<b>TROSEN</b>													
n = 10	20	27	1	22	31	3	33	36	31	66	33	68	1
n = 25	23	36	3	25	39	5	44	48	41	90	49	99	1
<b>EROSEN</b>													
n = 5	95	113	2	149	238	55	98	113	94	220	98	213	1
n = 10	635	744	23	725	1183	322	678	865	683	1489	896	1906	1
<b>TRIDIA</b>													
n = 10	13	16	0	14	17	0	21	24	10	21	10	21	1
n = 20	17	21	1	23	26	0	33	35	20	41	20	41	1
n = 30	26	33	3	32	40	3	42	44	31	63	31	63	1
<b>TBROY</b>													
n = 10	26	34	2	30	37	2	27	30	25	51	22	45	5
n = 20	26	31	1	27	31	0	35	38	36	74	34	70	5
n = 30	23	27	0	26	31	0	43	46	45	92	42	86	5
<b>BV</b>													
n = 10	38	44	6	26	35	3	26	29	26	74	30	82	3
n = 20	69	82	9	88	104	9	55	58	55	141	59	147	3
n = 30	30	35	1	113	137	14	61	64	240	519	130	297	3
<b>NONDIA</b>													
n = 10	31	42	2	31	38	0	34	41	25	72	22	61	1
n = 20	33	42	0	36	46	4	30	40	22	61	22	61	1
n = 30	37	46	0	74	76	0	30	40	24	56	27	66	1

Three conclusions can be drawn from the results. First, the sparse BFGS algorithm is virtually uniformly superior to the sparse PSB, with the superiority becoming more marked as the nonlinearity increases, particularly on the extended Rosenbrock function.

Second, although the sparse BFGS matrix need not be positive definite, nor does it necessarily possess quadratic termination, it generally outperforms the nonsparse BFGS on problems where sparsity is present, and appears never to be much worse even when it is not quite as efficient. This is highly encouraging.

Finally, conjugate gradient algorithms are not nearly comparable in efficiency measured in function and gradient evaluations with sparse variable-metric algorithms. However, as the overhead of these algorithms is much lower than the sparse variable-metric algorithms, when function and gradient evaluations are relatively inexpensive these may prove competitive in terms of machine time. Also, the results would appear to indicate that keeping a few additional points is of little, if any, benefit to a conjugate gradient algorithm. Hence, Buckley's scheme would appear to be useful only when a significant number of past points can be retained.

As a final note to this section, all variable-metric algorithms used  $B_0 = I$ . Toint's results indicate that for convex object functions, performance improves if  $B_0$  is estimated using finite differences. This should equally improve the sparse BFGS.

**Acknowledgement.** The author is indebted to John Dennis and Bobby Schnabel for many valuable comments on an earlier draft of this paper, which improved the current version and eliminated an error in the previous version.

Department of Management Information Systems  
College of Business and Public Administration  
The University of Arizona  
Tucson, Arizona 85721

1. C. G. BROYDEN, "A class of methods for solving nonlinear simultaneous equations," *Math. Comp.*, v. 19, 1965, pp. 577-593.
2. 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.
3. A. BUCKLEY, *A Combined Conjugate-Gradient Quasi-Newton Minimization Algorithm*, Working Paper, Mathematics Department, Concordia University, Montreal, Quebec, Canada, 1977.
4. W. C. DAVIDON, *Variable Metric Method for Minimization*, Report ANL-5990 Rev (1959), Argonne National Laboratories, Argonne, Ill., 1959.
5. J. E. DENNIS, "On some methods based on Broyden's secant approximation to the Hessian," *Numerical Methods for Nonlinear Optimization* (F. A. Lootsma, Ed.), Academic Press, London, 1972, pp. 19-34.
6. J. E. DENNIS, JR. & R. B. SCHNABEL, *Least Change Secant Updates for Quasi-Newton Methods*, Report CS-CU-132-78, Dept. of Computer Science, University of Colorado, Boulder, July, 1978.
7. A. V. FIACCO & G. P. MCCORMICK, *Nonlinear Programming, Sequential Unconstrained Minimization Techniques*, Wiley, New York, 1968, pp. 170-172.
8. R. FLETCHER & M. J. D. POWELL, "A rapidly convergent descent method for minimization," *Comput. J.*, v. 6, 1963, pp. 163-168.
9. R. FLETCHER, "A new approach to variable metric algorithms," *Comput. J.*, v. 13, 1970, pp. 317-322.
10. D. GOLDFARB, "A family of variable-metric methods derived by variational means," *Math. Comp.*, v. 24, 1970, pp. 23-26.

11. D. GOLDFARB, "Generating conjugate directions without line searches using factorized variable-metric updating formulas," *Math. Programming*, v. 13, 1977, pp. 94–110.
12. J. GREENSTADT, "Variations on variable-metric methods," *Math. Comp.*, v. 24, 1970, pp. 1–22.
13. M. D. HEBDEN, *An Algorithm for Minimization Using Exact Second Derivatives*, Report T. P. 515, AERE, Harwell, 1973.
14. E. S. MARWIL, *Exploiting Sparsity in Newton-Like Methods*, Report TR78-335, Dept. of Computer Science, Cornell University, Ithaca, New York, 1978.
15. J. J. MORÉ, B. S. GARBOW & K. E. HILLSTROM, *Testing Unconstrained Optimization Software*, Report TM-324, Applied Mathematics Division, Argonne National Laboratory, Argonne, Ill., 1978.
16. J. J. MORÉ & D. C. SORENSEN, *On the Use of Directions of Negative Curvature in a Modified Newton Method*, Report TM-319, Applied Mathematics Division, Argonne National Laboratory, Argonne, Ill., 1977.
17. M. J. D. POWELL, "A new algorithm for unconstrained optimization," *Nonlinear Programming* (J. B. Rosen, O. L. Mangasarian & K. Ritter, Eds.), Academic Press, New York, 1970.
18. R. B. SCHNABEL, *Analyzing and Improving Quasi-Newton Methods for Unconstrained Optimization*, Report TR77-320, Dept. of Computer Science, Cornell University, Ithaca, New York, 1977.
19. L. K. SCHUBERT, "Modification of a quasi-Newton method for nonlinear equations with a sparse Jacobian," *Math. Comp.*, v. 24, 1970, pp. 27–30.
20. D. F. SHANNO, "Conditioning of quasi-Newton methods for function minimization," *Math. Comp.*, v. 24, 1970, pp. 647–656.
21. D. F. SHANNO, "Conjugate gradient methods with inexact searches," *Math. Oper. Res.*, v. 3, 1978, pp. 244–256.
22. D. F. SHANNO & K. H. PHUA, "Numerical comparison of several variable-metric algorithms," *J. Optim. Theory Appl.*, v. 25, 1978, pp. 507–518.
23. D. F. SHANNO & K. H. PHUA, "Minimization of unconstrained multivariate functions," *ACM Trans. Math. Software*, v. 2, 1976, pp. 87–94.
24. PH. L. TOINT, "On sparse and symmetric updating subject to a linear equation," *Math. Comp.*, v. 31, 1977, pp. 954–961.
25. PH. L. TOINT, "Some numerical results using a sparse matrix updating formula in unconstrained optimization," *Math. Comp.*, v. 32, 1978, pp. 839–852.