SIAM J. NUMER. ANAL.
Vol. 16, No. 5, October 1979

0036-1429/79/1605-0006/$01.00/0

A RELATIONSHIP BETWEEN THE BFGS AND CONJUGATE GRADIENT ALGORITHMS AND ITS IMPLICATIONS FOR NEW ALGORITHMS*

LARRY NAZARETH†

Abstract. Based upon analysis and numerical experience, the BFGS (Broyden–Fletcher–Goldfarb–Shanno) algorithm is currently considered to be one of the most effective algorithms for finding a minimum of an unconstrained function, $f(x)$, $x \in \mathbb{R}^n$. However, when computer storage is at a premium, the usual alternative is to use a conjugate gradient (CG) method. In this paper we show that the two algorithms are related to one another in a particularly close way. Based upon these observations a new family of algorithms is proposed.

1. Introduction.

We are concerned here with the problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}}\ f(x)$$

where $f(x)$ is a nonlinear function. We first give a concise statement of the algorithms under consideration and summarize briefly some of their well known properties. We then show, in § 2, an exact correspondence between the search vectors developed by the BFGS and CG algorithms, when applied to quadratic functions. For arbitrary differentiable functions we give an interpretation of the BFGS algorithm as a CG algorithm with variable metric, chosen at each step from the Broyden β-class. These observations then lead us to a family of algorithms termed variable storage generalized conjugate gradient methods, introduced in § 4. A particular implementation and numerical results are given in § 5. We conclude with some remarks on the implications and future directions of this research.

The *conjugate gradient method* was originally proposed by Hestenes and Stiefel [1] for solving linear systems and extended to nonlinear optimization by Fletcher and Reeves [2]. Various formulations of the algorithm (see [2], [3], [7]) are equivalent when applied to quadratic functions, but differ for arbitrary functions. We shall consider here the CG method in a fixed metric defined by the positive definite symmetric matrix H and started from a given point x_1, which develops successive search directions d_j^{CG}, iterates x_j and gradients $g_j = \nabla f(x_j)$ as follows:

$$d_1^{\mathrm{CG}} = -Hg_1,$$

$$d_j^{\mathrm{CG}} = -Hg_j + \left[\frac{y_{j-1}^T Hg_j}{y_{j-1}^T d_{j-1}^{\mathrm{CG}}}\right] d_{j-1}^{\mathrm{CG}}, \qquad j > 1, \tag{1}$$

$$x_{j+1} = x_j + \lambda_j d_j^{\mathrm{CG}}, \quad \text{where } \lambda_j = \arg\min_{\lambda} f(x_j + \lambda d_j^{\mathrm{CG}}) \text{ and } y_{j-1} \triangleq (g_j - g_{j-1}).$$

In the basic form of the CG algorithm, H is set to the identity matrix, and it is well known that using an arbitrary positive definite symmetric matrix H corresponds to applying the change of variables $y = H^{-1/2}x$ to the basic algorithm.

Variable metric methods were originally introduced by Davidon [4], and subsequently clarified by Fletcher and Powell [11]. Broyden [5] developed a 1-parameter family of variable metric updates which has become known as Broyden's

* Received by the editors July 13, 1977, and in revised form February 2, 1979. This work was performed under the auspices of the U.S. Department of Energy. This paper constitutes a revised version of ANL-AMD Tech Memo 282.

† Applied Mathematics Division, Argonne National Laboratory, Argonne, Illinois 60439.

β-class. Given a positive definite symmetric matrix H and initial point x_1, these develop successive positive definite and symmetric approximations H_j^β to the inverse Hessian, successive search direction d_j^β, and iterates x_j as follows:

$$H_1^\beta = H,$$

$$H_j^\beta = H_{j-1}^\beta - \frac{H_{j-1}^\beta y_{j-1} y_{j-1}^T H_{j-1}}{y_{j-1}^T H_{j-1}^\beta y_{j-1}} + \frac{s_{j-1} s_{j-1}^T}{s_{j-1}^T y_{j-1}} + \beta_{j-1}(H_{j-1}^\beta y_{j-1} - \theta_{j-1}^\beta s_{j-1})(H_{j-1}^\beta y_{j-1} - \theta_{j-1}^\beta s_{j-1})^T, \qquad j > 1, \tag{2}$$

where

$$\beta_{j-1} \geqq 0,$$
$$\theta_{j-1}^\beta \triangleq y_{j-1}^T H_{j-1}^\beta y_{j-1} / (s_{j-1}^T y_{j-1}),$$
$$s_{j-1} \triangleq (x_j - x_{j-1}),$$
$$d_j^\beta = -H_j^\beta g_j$$

and

$$x_{j+1} = x_j + \lambda_j d_j^\beta$$

with

$$\lambda_j = \arg\min_\lambda f(x_j + \lambda d_j^\beta).$$

Particular cases are given by $\beta_{j-1} = 0$ (Davidon–Fletcher–Powell (DFP)), and $\beta_{j-1} = 1/(y_{j-1}^T H_{j-1}^\beta y_{j-1})$ (BFGS). In the latter case (2) simplifies to

$$H_j^{\mathrm{BFGS}} = H_{j-1}^{\mathrm{BFGS}} + \frac{1}{s_{j-1}^T y_{j-1}}\left[1 + \frac{y_{j-1}^T H_{j-1}^{\mathrm{BFGS}} y_{j-1}}{s_{j-1}^T y_{j-1}}\right] s_{j-1} s_{j-1}^T - \frac{1}{s_{j-1}^T y_{j-1}}(s_{j-1} y_{j-1}^T H_{j-1}^{\mathrm{BFGS}} + H_{j-1}^{\mathrm{BFGS}} y_{j-1} s_{j-1}^T), \tag{3}$$

$$d_j^{\mathrm{BFGS}} = -H_j^{\mathrm{BFGS}} g_j.$$

Analysis and numerical experience indicate that this is the most effective of the variable metric updates. The following properties of the conjugate gradient and variable metric algorithms are well known. See, e.g., [7].

When applied to the minimization of a quadratic function $\psi(x) = a + b^T x + \frac{1}{2} x^T A x$, with A positive definite and symmetric ($A > 0$), and using the same initial metric defined by H we have (i) termination in at most n steps, (ii) search vectors are conjugate, (iii) $g_i^T H g_j = 0$, $i \neq j$, (iv) the jth direction lies in the subspace spanned by $Hg_1, \cdots, Hg_j$, (v) since there is no flexibility in choice of directions given the above conditions, d_j^{CG} and d_j^β must be linearly dependent, (vi) $H_j^\beta A(d_1^\beta, \cdots, d_{j-1}^\beta) = (d_1^\beta, \cdots, d_{j-1}^\beta)$, (vii) provided premature termination does not occur $H_{n+1}^\beta = A^{-1}$.

Furthermore, *Dixon's theorem* [12] demonstrates that for general continuously differentiable objective functions, the jth search directions developed by any two members of Broyden's β-class, say $d_j^{\beta'}$ and $d_j^{\beta''}$, are linearly dependent and successive iterates are identical. This result requires that the same starting point x_1 and initial approximation H are used and that line searches are exact and unambiguously defined.

2. A result for quadratics. We now strengthen property (v) of § 1 to show that for one member of the β-class (the BFGS update), the search vectors d_j^{CG} and d_j^{BFGS} are the same in norm as well as direction. This correspondence is, we feel, indicative of underlying structure, and is developed further in § 3, for arbitrary functions.

LEMMA. *When the CG and BFGS algorithms are applied to a quadratic function* $\psi(x) = a + b^T x + \frac{1}{2}x^T Ax$, $A > 0$, *using the same starting point* x_1 *and positive definite symmetric* H, *then*

$$d_j^{CG} = d_j^{BFGS}, \qquad j = 1, 2, \cdots, n.$$

Proof. Fact 1. $g_{j+1}^T s_j = 0$. See, e.g., [7]. Further a well known result is that

$$g_k^T s_j = 0, \; k > j.$$

Fact 2. $H_j^{BFGS} g_k = Hg_k$, $j < k \leqq n+1$, $1 \leqq j \leqq n$.

Proof of Fact 2. This may be shown by induction on j. Assume true for H_{j-1}^{BFGS}, i.e.,

$$H_{j-1}^{BFGS} g_k = Hg_k, \qquad j-1 < k \leqq n.$$

Now combining (3), Fact 1 above, property (iii) of § 1, and the induction hypothesis we have

$$H_j^{BFGS} g_k = H_{j-1}^{BFGS} g_k = Hg_k, \qquad j < k \leqq n.$$

Since $H_1 g_k = Hg_k$ for $1 \leqq k \leqq n$, the result follows by induction. □

Returning to the proof of Lemma 1, we have

$$d_j^{BFGS} = -H_j^{BFGS} g_j.$$

Using (3) and the fact that line searches are exact, we have

$$d_j^{BFGS} = -H_{j-1}^{BFGS} g_j + \left[\frac{y_{j-1}^T H_{j-1}^{BFGS} g_j}{y_{j-1}^T d_{j-1}^{BFGS}}\right] d_{j-1}^{BFGS}.$$

Now from Fact 2 above, this gives

$$\begin{aligned} d_j^{BFGS} &= -Hg_j + \left[\frac{y_{j-1}^T Hg_j}{y_{j-1}^T d_{j-1}^{BFGS}}\right] d_{j-1}^{BFGS} \\ &= -Hg_j + \frac{y_{j-1}^T Hg_j}{y_{j-1}^T d_{j-1}^{CG}} d_{j-1}^{CG} \quad \text{using property (v) of § 1.} \\ &= d_j^{CG} \qquad\qquad\qquad\qquad \text{using (1),} \end{aligned}$$

and this is the desired result.

3. Interpretation of the BFGS algorithm for arbitrary differentiable functions. We employ the following theorem due to Powell. This is paraphrased below, and for the proof we refer the reader to [6].

THEOREM. *Let the variable metric method of* § 1 *be applied to a differentiable function* $f(x)$, *and assume that all line searches are exact and that the* λ_j *are chosen unambiguously. Let* $x_1, \cdots, x_j$ *be the sequence of iterates and* $H_1^\beta, \cdots, H_{j-1}^\beta$ *the sequence of matrices developed prior to the* j*-th iteration, and assume that no search vector* d_j^β *vanishes. Then, if the choice of* β *corresponding to the BFGS update is used at iteration* j, *the matrix* H_j^{BFGS} *obtained is independent of the parameter values* β *used during previous iterations.*

Invoking this theorem, setting $\beta_{j-1} = 1/(y_{j-1}^t H_{j-1}^\beta y_{j-1})$ in (2), and using the fact that line searches are exact, we can state the BFGS algorithm as follows:

$$
\begin{aligned}
d_1^{\mathrm{BFGS}} &= -H_1 g_1, \\
d_j^{\mathrm{BFGS}} &= -H_{j-1}^\beta g_j + \left[\frac{y_{j-1}^T H_{j-1}^\beta g_j}{y_{j-1}^T d_{j-1}^{\mathrm{BFGS}}}\right] d_{j-1}^{\mathrm{BFGS}}
\end{aligned} \tag{4}
$$

and

$$
x_{j+1} = x_j + \lambda_j d_j^{\mathrm{BFGS}}
$$

where $\lambda_j = \arg\min_\lambda f(x_j + \lambda d_j^{\mathrm{BFGS}})$. H_j^β is developed from H_{j-1}^β using (2) and x_1 and H_1 are specified.

By comparing (4) and (1) we see that the BFGS algorithm may be interpreted as a CG algorithm for which the metric, instead of being fixed as in (1), is updated at each step to be any member of the Broyden β-class. This interpretation is of value because it motivates techniques for using limited storage to improve the conjugate gradient method, discussed in the next section.

4. Variable storage generalized conjugate gradient algorithms. Conjugate gradient algorithms require the storage of only a few vectors, typically four. Variable metric methods on the other hand require $O(n^2)$ storage. As Fletcher states [7, p. 82] "practical experience with the Fletcher–Reeves conjugate gradient method is that more iterations have usually been required for convergence as against variable metric algorithms—a factor of two is typical. This has been ascribed to the fact that less information is stored in the Fletcher–Reeves method about the behavior of the function." Therefore, by using more information about the function one might hope to accelerate the convergence of the conjugate gradient method. For example, in a problem with 10^3 variables, a user may not be able to provide $10^6/2$ words of working storage, thus ruling out variable metric codes implemented in the standard way. However, it may be quite feasible for him to provide $2 * 10^5$ words, well above the $4 * 10^3$ words required by conjugate gradient methods.

The observations made in earlier sections lead us to suggest the following family of algorithms which can exploit additional storage and form a continuum between the BFGS and CG methods.

The following algorithm describes the family in general terms. We also explain the possible options and discuss them. Numerical results for a particular implementation are given in § 5.

On Input.

- n dimension of problem.
- x_1 starting point.
- δ vector giving diagonal elements of initial diagonal approximation to inverse Hessian H_0. Note in particular that the symbol H_j represents the $n \times n$ Hessian inverse approximation at step j. This is *not* stored. Instead it is defined implicitly by storing vectors and scalars defining the rank-1 or rank-2 updates at Step 5B below.

Step 1. *Initialize.*

$f_1 \leftarrow f(x_1)$, $g_1 \leftarrow g(x_1)$, $y_0 \leftarrow 0$, $d_0 \leftarrow 0$, H_0 and H_1 are diagonal matrices defined by δ, $j \leftarrow 0$.

Step 2. *Develop search direction.*

$$d_{j+1} \leftarrow -H_j g_{j+1} + \left[\frac{y_j^T H_j g_{j+1}}{y_j^T d_j}\right] d_j.$$

Comment. Relation (4) of § 3 is used to define search directions. When $j = 0$ the multiplier for the second term above is indefinite and is taken to be zero.

Step 3. *Search.*

$j \leftarrow j+1$ if $g_j^T d_j > 0$ *then* restart;

$x_{j+1} \leftarrow x_j + \lambda_j d_j$,

where $\lambda_j = \min_\mu f(x_j + \mu d_j)$.

Comment. For purposes of analysis, line searches are taken to be exact. In practice they will not be. In the usual CG method, a fairly accurate line search is required, but with VSGCG algorithms we can expect this requirement to be somewhat relaxed.

Step 4. *Test for convergence.*

Stop if convergence criterion is met.

Step 5. *If* available storage is exceeded.

Step 5A *then* employ a Reset Option

Comment. Possible reset options are H_j reset to the diagonal matrix defined by δ and go to Step 5B *or* H_{j+1} fixed at value of approximation when storage ran out and go to Step 6.

Step 5B. *else* update H_j to H_{j+1} using a member of the β-class.

Comment. There are a number of options here—what member of the β-class to use, whether to employ projected vectors (see [9]), and how frequently to perform the update, i.e. whether to update whenever possible or every k iterations where k is some fraction of n determined by the amount of storage available. See Remarks below. Note also that as discussed above, only the vectors and scalars defining the update are stored.

Step 6. *If* restart criterion not satisfied *then go to* Step 2

else employ suitable restart option

Comment. Possible restart criteria are to restart as suggested by Fletcher and Reeves [2] every n or $n+1$ iterations or to use techniques suggested by Powell [10]. The restart option is also linked to the choice for the reset option.

Remarks. The number of vectors of storage provided is a variable. When minimal storage is provided, so Step 5B is never executed, then the method is the standard conjugate gradient method. If n^2 words of storage are provided (assuming the symmetric rank-1 update is used as in § 6) or $2n^2$ (when a rank-2 update is used) and updating is performed at every iteration then it is the BFGS algorithm with resetting.

Also, one can easily show, provided the algorithm does not break down due to instabilities associated with the update, that it has quadratic termination. This is in contrast to a variable metric algorithm, which holds the approximation H_k fixed at some stage k, when $k < n$.

Suppose $v_1, \cdots, v_k$ are the vectors defining the metric. A particularly interesting question is how to replace v_1 by a new vector, say v_{k+1} and retain quadratic termination and properties of the metric, circumventing the need to reset H_j.

5. Numerical results. A particular choice of options were implemented as follows:

(a) *Updating option* (Step 5B). The symmetric rank-1 update, which is a member of the Broyden family, was used. This is defined by

$$H_j^{RK1} = H_{j-1}^{RK1} - \frac{(H_{j-1}^{RK1} y_{j-1} - s_{j-1})(H_{j-1}^{RK1} y_{j-1} - s_{j-1})^T}{(H_{j-1}^{RK1} y_{j-1} - s_{j-1})^T y_{j-1}}.$$

This has the advantage of requiring just one additional vector $(H_{j-1}^{RK1} y_{j-1} - s_{j-1})$ and the scalar $(H_{j-1}^{RK1} y_{j-1} - s_{j-1})^T y_{j-1}$ to define it. It has the disadvantage that the update can be unstable, though we partially control the latter by not updating H_{j-1}^{RK1} if $(H_{j-1}^{RK1} y_{j-1} - s_{j-1})^T y_{j-1}$ is too small. For purposes of experimentation this was adequate. These disadvantages can be successfully overcome by using rank-2 updates, but one then incurs the penalty of requiring 2 vectors/update to be saved.

(b) *Resetting option* (Step 5A). When the available storage is exhausted, the metric is held fixed for the rest of the cycle.

(c) *Restarting option* (Step 6). The cycle is restarted every $(n+1)$ iterations, as in the Fletcher–Reeves implementation.

(d) *Line search* (Step 3). The initial step and line search procedure follow the techniques of Fletcher [13], and are the same as those used in Nazareth [14].

(e) *Test functions.* The implementation was run on the trigonometric functions, see, e.g., Powell [15]. They are defined as follows:

$$\min F(x) = \sum_{i=1}^{n} [f_i(x)]^2$$

where $f_i(x) = \sum_{j=1}^{n} (A_{ij} \sin x_j + B_{ij} \cos x_j) - E_i$.

TABLE 1

nv	No. of function/gradient evaluations	Final function value
0	521	0.10694E-15
1	600	0.63001E-16
2	375	0.17567E-16
3	417	0.37234E-16
4	318	0.84301E-16
5	308	0.87112E-16

TABLE 2

nv	No. of function/gradient calls	Final function value
0*	521	0.10694E-15
5*	308	0.87112E-16
10	246	0.60415E-16
15	213	0.62477E-15
20	162	0.23233E-16

* These duplicate results in Table 1.

A_{ij} and B_{ij} are uncorrelated pseudo-random numbers between -100 and $+100$ and the numbers E_i are calculated to accord with a particular solution, taken to be $x_j^* = 2.0\ \forall j$, so that $F(x) = 0$ at the solution x^*. The starting point was also selected using the random number generator. (On different runs, the random number generator uses, of course, reseeded so as to duplicate the test function.)

n was set to 20 and nv below denotes the number of vectors stored, in defining the metric. $nv = 0$ gives the usual conjugate gradient method and $nv = 20$ gives the BFGS with a restart every $n + 1$ iterations.

For further details of the implementation, see Nazareth [8].

6. Concluding remarks. The results discussed above are derived from one of many possible implementations. The central idea that underlies the new family of algorithms introduced here is that variable metric information is used within the framework of conjugate gradient methods. In particular, this enables us to develop a continuum of algorithms between the standard conjugate gradient method and the BFGS method. Many important and interesting questions remain to be answered, both within the context of quadratics (e.g. see Remarks of § 4) and when VSGCG algorithms are applied to more general functions (e.g. demonstrating n-step quadratic convergence). These are currently under investigation.

REFERENCES

[1] M. R. HESTENES AND E. STIEFEL, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Stand., 49 (1952), pp. 409–536.

[2] R. FLETCHER AND C. M. REEVES, *Function minimization by conjugate gradients*, Comput. J., 7 (1964), pp. 149–154.

[3] E. POLAK, *Computational Methods in Optimization: a Unified Approach*, Academic Press, New York, 1971.

[4] W. C. DAVIDON, *Variable metric method for minimization*, AEC Research and Development Report, ANL-5990, 1959.

[5] C. G. BROYDEN, *The convergence of a class of double-rank minimization algorithms*, J. Inst. Math. Appl., 6 (1970), pp. 76–90.

[6] M. J. D. POWELL, *Unconstrained minimization and extension for constraints*, T.P. 495, U.K.A.E.A. Research Group, Atomic Energy Research Establishment, Harwell, England, 1972.

[7] W. MURRAY, ED., *Numerical Methods for Unconstrained Optimization*, Academic Press, New York-London, 1972.

[8] L. NAZARETH, *Minkit—An Optimization System*, ANL-AMD Tech Memo 305, Applied Mathematics Div., Argonne National Laboratory, 1977. Presented at ORSA/TIMS Meeting (San Francisco, May, 1977).

[9] W. C. DAVIDON, *Optimally conditioned optimization algorithms without line searches*, Math. Programming, 9 (1975), pp. 1–30.

[10] M. J. D. POWELL, *Restart procedures for the conjugate gradient method*, C. S. S. 24, Atomic Energy Research Establishment, Harwell, England, 1975.

[11] R. FLETCHER AND M. J. D. POWELL, *A rapidly convergent descent method for minimization*, Comput. J., 6 (1963), pp. 163–168.

[12] L. C. W. DIXON, *Quasi-Newton algorithms generate identical points*, Math. Programming, 2 (1972), pp. 383–387.

[13] R. FLETCHER, *A FORTRAN subroutine for minimization by the method of conjugate gradients*, Report R-7073, Atomic Energy Research Establishment, Harwell, England, 1973.

[14] L. NAZARETH, *A Conjugate direction algorithm without line searches*, J. Optimization Theory Appl., 23 (1977), pp. 373–387.

[15] M. J. D. POWELL, *A FORTRAN subroutine for solving systems of non-linear algebraic equations*, Numerical Methods for Non-linear Algebraic Equations, P. Rabinowitz, ed., Gordon & Breach, New York, pp. 87–114.