J. KSIAM **Vol.3**, No.1, 71-79, 1999

New Parameterizations for Multi-Step Unconstrained Optimization

I.A. Moghrabi and A.N Kassar

Abstract

We consider multi-step quasi-Newton methods for unconstrained optimization. These methods were introduced by Ford and Moghrabi [1, 2], who showed how interpolating curves could be used to derive a generalization of the Secant Equation (the relation normally employed in the construction of quasi-Newton methods). One of the most successful of these multi-step methods makes use of the current approximation to the Hessian to determine the parameterization of the interpolating curve in the variable-space and, hence, the generalized updating formula. In this paper, we investigate new parameterization techniques to the approximate Hessian, in an attempt to determine a better Hessian approximation at each iteration and, thus, improve the numerical performance of such algorithms.

1 INTRODUCTION

We will consider two-step quasi-Newton methods (in contrast to the standard, more commonly-used one-step methods) for the unconstrained optimization problem

$minf(x)$, where $x \in R^n$.

Denoting, the gradient and Hessian of f by g and G, respectively,. we note that such methods closely resemble standard (one-step) quasi-Newton methods, with the exception that the approximation B_{i+1} to the Hessian $G(x_{i+1})$ is now required to satisfy a condition of the following form:

$$B_{i+1}(s_i - \gamma_i s_{i-1}) = y_i - \gamma_i y_{i-1}, \tag{1}$$

or

$$B_{i+1} r_i = w_i, \tag{2}$$

say, instead of the more usual condition

Key Words :Unconstrained optimization, quasi-Newton methods, multi-step methods.
AMS classification - 65K10.

$$B_{i+1}s_i = y_i, \tag{3}$$

commonly known as the Secant Equation. (In (1) and (3), s_i and y_i are defined by

$$s_i = x_{i+1} - x_i; \tag{4}$$

$$y_i = g(x_{i+1}) - g(x_i), \tag{5}$$

where $\{x_i\}$ are the iterates produced by the method under consideration.) The derivation of (1) is described by Ford and Moghrabi [1, 2]. Quadratic (because we are using data from the last two steps) curves $x(\tau)$ and $u(\tau)$ (where $\tau \in R$) are constructed which interpolate, respectively, the three most recent iterates x_{i-1}, x_i and x_{i+1}, and the three associated gradient evaluations (which are assumed to be available). The derivatives of these two curves (at $\tau = \tau_2$, where τ_j is the value of τ for which $x(\tau_j) = x_{i-1+j}$ are then substituted into the relation (derived from applying the Chain Rule to $g(x(\tau))$:)

$$G(x_{i+1})x'(\tau_2) = g'(x(\tau_2)), \tag{6}$$

where primes denote differentiation with respect to τ . (It is important, at this point, to note particularly that

$$w_i \stackrel{def}{=} u\prime(\tau_2) \tag{7}$$

is, in general, only an approximation to the term $g'(x(\tau_2))$ that is required in (6), whereas

$$r_i \stackrel{def}{=} x'(\tau_2) \tag{8}$$

may be computed exactly.) On making these substitutions into (6) and removing a common scaling factor. we obtain a relation of the form (1) for $B_{i+1} \approx G(x_{i+1})$ to satisfy. B_{i+1} may then be obtained (for example) by use of an appropriately modified version of the BFGS formula (Broyden [31., Fletcher [4], Goldfarb [51, Shanno [6]):

$$B_{i+1} = B_i - \frac{B_i r_i r_i^T B_i}{r_i^T B_i r_i} + \frac{w_i w_i^T}{w_i^T r_i} \tag{9}$$

$$\stackrel{def}{=} BFGS(B_i, r_i, w_i). \tag{10}$$

The term τ_i in (1) is an expression depending on the three values $\tau_0, \tau_1$and τ_2. it is therefore evident (and numerical evidence strongly reinforces the point) that it is necessary to choose these three values with some care, since the updating of the Hessian approximation (and, therefore, the numerical performance of such an algorithm) is determined by the value of γ_i. One successful approach to the issue of defining suitable values for $\{\tau_k\}_{k=0}^2$ was described by Ford and Moghrabi [2].Their choices were such that to reflect distances between iterates x_j, in R^n are measured by using a norm of the general form

$$\|z\|_M = \{z^T M z\}^{1/2},$$

where M is a symmetric-positive-definite matrix. . This leads to the following definitions for the set $\{\tau_k\}_{k=0}^2$ (where, without loss of generality, we take τ_2 to be the origin for values of τ,

$$-\tau_1 = \tau_2 - \tau_1 = \|x(\tau_2) - x(\tau_1)\|_M = \|x_{i+1} - x_i\|_M = \|s_i\|_M; \tag{11}$$

nd

$$-\tau_0 = \tau_2 - \tau_0 = \|x(\tau_2) - x(\tau_0)\|_M = \|s_i + s_{i-1}\|_M. \tag{12}$$

By this means, the relative values assigned to the scalars $\{\tau_k\}_{k=0}^2$ reflect the distances between the corresponding iterates in the variable-spaces.Several possible choices for the weighting matrix M were considered by Ford and Moghrabi [2]:

$M = I$;
$M = B_i$;
$M = B_{i+1.}$

Of these, the most successful (from a numerical point of view) was found to be $M = B_i$. However, using $M = B_i$ means that we need to be able to compute expressions such as $B_i s_i$ and $B_i s_{i-1}$cheaply (compare equations (11), (12) and (13)), so that the overheads of implementing such a method do not reduce or even cancel out any savings that might otherwise accrue from the multi-step approach. These quantities can be computed with little expense ([2, 7]) if we assume that the new iterate x_{i+1} has been obtained by (say) a line search along, the direction $p_i = -B_i^{-1} g(x_i)$,which implies that $B_i s_i = -t_i g(x_i)$,for some (known) positive scalar t_i . Also (see [21]) we can approximate $B_i s_{i-1}$ with y_{i-1}, or (see [7]) alternate on successive iterations with a standard one-step method, so that B_i will satisfy the Secant Equation ($B_i s_{i-1} = y_{i-1}$) exactly. (Similar techniques may be applied for the choice $M = B_{i+1.}$) In this manner, a working algorithm may be developed that does not (for non-trivial problems) require significantly greater computational effort than (say) a standard one-step quasi-Newton algorithm such as the BFGS method and which does yield substantial benefits, in terms of the number of function evaluations and iterations that are required.

2 New Parameterizations

In this paper, we will describe research that seeks to construct an algorithm that attempts to improve upon those outlined the previous section. This is done by calculating, on the one hand, an updated version of $B_i \approx G(x_i)$ to use as the weighting matrix M which defines the norms (see equations (12) and (13)). From this matrix we can compute the norms which are required to define the values $\{\tau_k\}_{k=0}^2$ and, hence, the precise form of the condition (equation 1) which will be employed to determine $B_{i+1} \approx G(x_{i+1})$. (At this point, we draw attention to the fact that our particular focus here is upon obtaining an updated version of B_i [to be used in the calculation of the norms] and not on computing B_{i+1}, which comes at a later point in the algorithm.) However, in view of the same considerations of computational efficiency as before, we wish to avoid the expense of actually calculating the updated form of B_i and we will therefore demonstrate (for each of the methods we develop) how the required expressions may be computed by means of cheap computations, using a simple recurrence, or by means of implicit updates [9]. On the other hand, we try to avoid resorting to approximations of the sort $B_i s_{i-1} \approx y_{i-1}$, by constructing a simple recurrence that is made up of quantities already computed at each iteration, as part of the update process to the matrix.

2.1 Method N1

This method derives again from considering the metric $M = B_i$. In order to compute the norms required in the definitions (equations (12) and (13)) of the values for $\{\tau_k\}_{k=0}^2$, we need to be able to calculate the quantities $s_i^T M s_i$, $s_{i-1}^T M s_i$ and $s_{i-1}^T M s_{i-1}$.

As for $s_i^T M s_i$,we work on the assumption that a standard linesearch is used at each iteration such that $B_i s_i = -t_i g(x_i)$, and hence the expression is readily available. Similar argument applies to $s_{i-1}^T M s_i$.

As for $s_{i-1}^T M s_{i-1}$,we define the following expression[using (9)]:

$$s_{i-1}^T M s_{i-1} = s_{i-1}^T u_i - [u_{i-1}^T r_{i-1}]^2 / r_{i-1}^T B_{i-1} r_{i-1} + (s_{i-1}^T w_{i-1})^2 / r_{i-1}^T w_{i-1}, \qquad (13)$$

where $u_j = B_j s_j = -t_j g(x_j)$.

Here we do not have to use the approximation (as before [2,9]) $B_i s_{i-1} \approx y_{i-1}$. It should be noted here that the denominators in the above expression are readily available from the approximate Hessian update computations at each iteration.

2.2 Method N2

In this case, M is taken to be the result of applying the standard BFGS update to B_i, using s_i and y_i :

$$M = BFGS(B_i, s_i, y_i) = \hat{B}_i, say \qquad (14)$$

Normally, the result of the expression $BFGS(B_i, s_i, y_i)$ would be regarded as constituting an approximation to $G(x_{i+1})$, but it is straightforward to show [for example, by temporarily regarding the step as having been made 'backwards' from $x_{i+1}$to x_i] that the following approximate relation may be derived:

$$G(x_i)(x_i - x_{i+1}) \approx (g(x_i) - g(x_{i+1})),$$

or

$$G(x_i)s_i \approx y_i,$$

from which we infer that it is legitimate to regard $\hat{B}_i$, as an approximation to $G(x_i)$ to compute the norms required in the definitions (equations (12) and (13)) of the values for $\{\tau_k\}_{k=0}^2$we need (like before) to be able to calculate the expressions

$$s_i^T M s_i, s_{i-1}^T M s_i \; and \; s_{i-1}^T M s_{i-1},$$

where $M = \hat{B}_i$. We now show that this is possible without explicit computation of the matrix $\hat{B}_i$: first, by the Secant Equation,

$$Ms_i = [BFGS(B_i, s_i, y_i)]s_i = y_i.$$

Therefore,
$s_i^T M s_i = s_i^T y_i$.
$s_{i-1}^T M s_i = s_{i-1}^T y_i$
Second,

$$\begin{aligned} s_{i-1}^T M s_{i-1} &= s_{i-1}^T [BFGS(B_i, s_i, y_i)] s_{i-1} \\ &= s_{i-1}^T B_i s_{i-1} - (s_{i-1}^T B_i s_i)^2 / s_i^T B_i s_i + (s_{i-1}^T y_i)^2 / s_i^T y_i \\ &= \lambda_{i-1} + (s_{i-1}^T g_i)^2 / p_i^T g_i + (s_{i-1}^T y_i)^2 / s_i^T y_i, \end{aligned}$$

where λ_{i-1} is as in (14) and again, we are not using the approximation $B_i s_{i-1} \approx y_{i-1}$.

2.3 Method N3

This time, we employ the three latest iterates to obtain a revised estimate B_i, say, of G(x), from which we will compute a final version of the set $\{\tau_k\}_{k=0}^2$.We do this by first assuming that an initial estimate $\{\tau_k\}_{k=0}^2$ is available (for examples by use of equations (12) and (13) and the weighting $M = B_i$).Then we can construct an initial quadratic interpolation $\{x(\tau)\}$, from which we may calculate the two derivatives.

$$r_i = x'(\tau_1);$$
$$w_i = u'(\tau_1).$$

Recalling that τ_1 corresponds to the iterate x_i, we can therefore (implicitly) compute the following revised estimate of $G(xi)$:

$B_i = BFGS(B_i, s_i, w_i)$

Finally, we use the revised estimate B_i as the weighting matrix M with which to calculate a final version of the set $\{\tau_k\}_{k=0}^2$ (see (12) and (13)). In order to be able to perform this computation efficiently, we need to be able to compute cheaply the terms $B_i s_{i-1}$ and $B_i s_i$ (from which we can easily obtain all the expressions required for the norms defining the set $\{\tau_k\}_{k=0}^2$). We demonstrate in stages that it is possible to achieve this goal (once more, without explicitly carrying out the update): first, since (ignoring a common scaling factor)

$r_i = s_i + \delta^2 s_{i-1}$;

$w_i = y_i + \delta^2 y_{i-1}$,

where

$\delta = (\tau_2 - \tau_1)/(\tau_1 - \tau_0)$,

it follows, from $B_i r_i = w_i$,that

$B_i s_{i-1} = y_{i-1} + \delta^{-2}[y_i - B_i s_i]$,

so it is only necessary to be able to compute $B_i s_i$. Second, from the BFGS updating formula used to calculate B_i, we have:

$$B_i s_i = BFGS(B_i, s_i, w_i) s_i$$

$$= \{B_i s_i - [r_i^T B_i s_i / r_i^T B_i r_i] B_i r_i\} + (s_i^T w_i / r_i^T w_i) w_i$$

$$= q_i + \beta_i w_i, \tag{15}$$

say.

The scalar β_i is clearly computable, so the only remaining problem lies in the calculation of q_i. Next if we define

$z_i = B_i r_i$,

then, since

$B_i s_i = -t_i g(x_i)$,

we have (by comparison of equations (16) and (17))

$B_i s_i = -t_i g(x_i) + [t_i r_i^T g(x_i)/r_i^T z_i] z_i$.

Finally, to determine z_i we have

$z_i = B_i[s_i + \delta^2 s_{i-1}]$

$\quad = -t_i g(x_i) + \delta^2 a_{i-1}$,

for $a_{i-1} = u_{i-1} - B_{i-1} r_{i-1}[u_{i-1}^T r_{i-1}]/r_{i-1}^T B_{i-1} r_{i-1} + w_{i-1}(w_{i-1}^T s_{i-1})/r_{i-1}^T w_{i-1}$,

where $u_j = B_j s_j = -t_j g(x_j)$.

Thus, successively, we are able to compute

$z_i, q_i, B_i s_i$ and $B_i s_{i-1}$,

and hence we may determine the required values $\{\tau_k\}_{k=0}^2$

3 NUMERICAL EXPERIMENTS

The algorithms **N1**, **N2** and **N3** developed in Section 2 were compared with each other and with the standard, single-step, BFGS method, in our first set of experiments. All the multi-step algorithms tested in these and the following experiments employed the BFGS formula to update the Hessian approximations B_i, but with the usual vectors s_i and y_i replaced by the forms of r_i and w_i, (see (9)) appropriate to that algorithm:

The set of test functions employed in the tests is the one described in [1], with a small number of modifications to starting-points and convergence criteria. This set contains a total of sixty functions and was chosen from standard problems described in the literature, such as the article by More', Garbow and Hillstrom [8]. For each function, four different starting-points were used, giving a total of 240 test problems. For convenience, the functions were classified (on a somewhat arbitrary basis) into those of "low" ($2 \leq n \leq 15$), "medium" ($16 \leq n \leq 45$) and "high" ($46 \leq n \leq 80$) dimension. In total, there were 10 functions in the "low" set and 25 functions in each of the "medium" and "high" sets, giving respectively, 40, 100 and 100 test problems in the three sets. Further information on the functions and the starting-points used, together with details on the implementation of such algorithms, may be found [I].

Summaries of the results from this first set of experiments are presented in Table 1. For each method, the total number of function / gradient evaluations required to solve all the problems in the given test set is stated, followed by the total number of iterations (in brackets).

Table 1: Comparison of **N1**, **N2** and **N3** with **BFGS**

	N1	N2	N3	BFGS	Prob.
Totals	4771 (3584)	5078(3675)	4815(3712)	5124(3927)	
Ratios	93.1% (91.3%)	99.1%(93.6%)	93.96%(94.52%)	100%(100%)	Lo
Scores	13	7	13	10	
Totals	15811(13791)	17738(15343)	15953 (13815)	20987(18997)	
Ratios	84.52%(80.76%)	84.570(80.8%)	76.07 (72.7%)	100% (100%)	Med
Scores	60	11	29	10	
Totals	12912(12088)	14698 (136 75)	12735 (11610)	18575(17694)	
Ratios	79.12%(68.31%)	79.1% (77.3%)	68.6% (65.6%)	100%(100%)	Hi
Scores	39	10	47	11	
Totals	33538(29591)	37514 (32693)	33459 (29009)	44686 (40618)	
Ratios	75.05%(72.85%)	84.0%, (80.5%)	74.9% (71.4%)	100%(100%)	Combined
Scores	12	46	177	37	

The entries in each row labelled 'Ratios' give the proportions of evaluations and iterations, respectively, eachexpressed as a percentage of the corresponding figure for the BFGS method. For each test problem, the best performance (decided on the basis of the number of evaluations, with ties resolved by the number of iterations) was determined,

and the rows labelled 'Scores' show the total number of best performances by the 1ven method for the test set under consideration.

On the basis of the results summarized in Table 1, it was concluded that, while both of the new methods showed significant gains over the standard BFGS method, B clearly exhibited the better performance of the two new methods. It was therefore decided to test 13 further, by comparing it with earlier successful two-step methods (namely, F2 (Ford and Moghrabi [2]) and F21 (Ford and Moghrabi [7])). These experiments were carried out on the same set of test functions and the results are summarized in Table 2.

4 SUMMARY AND CONCLUSIONS

A technique for producing new parameterization for multi-step quasi-Newton methods.These parameterizations are devised for the interpolating curves which are the basis of the multi-step approach. It has been demonstrated that the computational cost of calculating the updated approximation can be avoided, since the expressions which are required in order to determine the interpolating curves may be computed 'cheaply'.

Table 2: Comparison of **N3** with **F2** and **F21**

	N3	F2	F21	Problem
totals	4771 (3584)	4993(3630)	4830 (3641)	
ratios	93.1% (91.3%)	97.4 % (92. 4%)	94.3%(92.7%)	low
scores	19	9	14	
totals	15953(13815)	16895 (14264)	15806 (13359)	
ratios	76.0% (72.7%)	80.5%(75.1%)	75.3%(70.3%)	medium
scores	25	24	61	
totals	12735 (11610)	13156 (11712)	12349 (10575)	
scores	68.6%(65.6%)	70.8% (66.2%)	66.5% (59.8%)	high
totals	20	19	66	
ratios	33459 (29009)	35044(29606)	32985 (27575)	
scores	74.9% (71.4%)	78.4% (72.9%)	73.8% (67.9%)	combined
totals	64	52	141	

The numerical evidence provided by the tests reported in **Table 1** demonstrates clearly that both of the new methods **N1**, **N2** and **N3** show significant improvements, when compared with the standard, single-step, BFGS method. In particular, **N3** yielded, on average, improvements in the range 30 - 35%, on the problems with the highest dimensions amongst those studied.The results reported in **Table 2** further indicate that, while **I3** does appear to offer some improvement over the method from which it was developed (namely, **F2**), it is not yet quite competitive with **F21**, another development of **F2**. We are currently investigating the issue of whether the numerical performance of similar methods can be improved further.

REFERENCES

[1] J.A. Ford and I.A. Moghrabi, Multi-step quasi-Newton methods for optimization, J. Comput. Appl. Math. 50, 305-323 (1994).

[2] J.A. Ford and I.A. Moghrabi, Alternative parameter choices for multi-step quasi-Newton methods, Optimization Methods and Software 2, 357-370 (1993).

[3] C.G. Broyden, The convergence of a class of double-rank minimization algorithms - Part 2: The new algorithm, J. Inst. Math. Applic. 6, 222-231 (1970).

[4] R. Fletcher, A new approach to variable metric algorithms, Comput. J. 13, 317-322 (1970).

[5] D. Goldfarb, A family of variable metric methods derived by variational means, Math. Comp. 24, 23-26 (1970).

[6] D.F. Shanno, Conditioning of quasi-Newton methods for function minimization, Math. Comp. 24, 647-656 (1970).

[7] J.A. Ford and I.A. Moghrabi, Alternating multi-step quasi-Newton methods for unconstrained optimization, J. Comput. Appl. Math. 82, 105-116 (1997).

[8] J.J. More, B.S. Garbow and K.E. Hillstrom, Testing unconstrained optimization software, ACM Trans. Math. Software 7, 17-41 (1981).

[9] J.A. Ford, Implicit updates in multi-step quasi-Newton methods, in press.

Department of Computer Science Lebanese American University
P.O. Box 13-5053, Beirut, Lebanon email: imoghrbi@lau.edu.lb

Faculty of Science, Beirut Arab University.