EXTRA MULTISTEP BFGS UPDATES IN QUASI-NEWTON METHODS

ISSAM A. R. MOGHRABI

Received 5 December 2005; Revised 7 March 2006; Accepted 25 April 2006

This note focuses on developing quasi-Newton methods that combine $m+1$ multistep and single-step updates on a single iteration for the sake of constructing the new approximation to the Hessian matrix to be used on the next iteration in computing the search direction. The approach considered here exploits the merits of the multistep methods and those of El-Baali (1999) to create a hybrid technique. Our numerical results are encouraging and reveal that our proposed approach is promising. The new methods compete well with El-Baali's extra update algorithms (1999).

1. Introduction

This note considers methods that efficiently minimize unconstrained functions of the form

$$\text{minimize } f(x), \quad \text{where } f : \mathbb{R}^n \longrightarrow \mathbb{R},\ x \in \mathbb{R}^n. \tag{1.1}$$

Quasi-Newton methods require only the function and its first partial derivatives (gradient) to be available. However, an approximating matrix to the Hessian is used and updated throughout the iterations to reflect the changes in the function and its gradient.

Let g and G denote the gradient and the Hessian of f, respectively. Given B_i, the current approximation to the Hessian, we need to find a new approximating matrix B_{i+1} to the Hessian. The new Hessian approximation, B_{i+1}, satisfies in standard quasi-Newton methods the so-called *secant equation*:

$$B_{i+1}s_i = y_i, \tag{1.2}$$

where

$$\begin{aligned} s_i &= x_{i+1} - x_i, \\ y_i &= g_{i+1} - g_i. \end{aligned} \tag{1.3}$$

Hindawi Publishing Corporation
International Journal of Mathematics and Mathematical Sciences
Volume 2006, Article ID 12583, Pages 1–8
DOI 10.1155/IJMMS/2006/12583

In general, updating formulas are of the form $B_{i+1} = B_i + C_i$, where C_i is a correction matrix.

The most successful rank-two formula, developed independently by Broyden, Fletcher, Goldfarb, and Shanno, is known as the BFGS formula. This formula is given by

$$B_{i+1}^{\mathrm{BFGS}} = B_i + \frac{y_i y_i^T}{y_i^T s_i} - \frac{B_i s_i s_i^T B_i}{s_i^T B_i s_i}. \tag{1.4}$$

Numerical results indicate that the BFGS method is superior to other updating formulas, especially when inaccurate line searches are used [8, 10, 11].

The note starts with a brief account of some of the successful multistep algorithms that will be used in the numerical comparisons in the last section of this note. Then the new implicit updated algorithms are derived. We finally present the numerical comparisons.

2. Multistep quasi-Newton methods

Let $\{x(\tau)\}$ or X denote a differentiable path in $\mathbb{R}^n$, where $\tau \in \mathbb{R}$, or simply the path X. Then we apply the chain rule to the gradient vector $g(x(\tau))$ in order to find the derivative of the gradient g with respect to τ to obtain

$$\frac{dg}{d\tau} = G(x(\tau))\frac{dx}{d\tau}. \tag{2.1}$$

Therefore, at any point on the path X the Hessian G must satisfy (2.1) for any value of τ, specifically for $\tau = \tau_c$, where τ_c is a constant scalar and $\tau_c \in \mathbb{R}$. This will result in the following equation, called the "Newton equation" [4, 8]:

$$\left.\frac{dg}{d\tau}\right|_{\tau=\tau_c} = G(x(\tau))\left.\frac{dx}{d\tau}\right|_{\tau=\tau_c}. \tag{2.2}$$

Since we wish to derive a relation satisfied by the Hessian at x_{i+1}, we choose a value for τ, τ_m, that corresponds to the most recent iterate in the Newton equation as follows:

$$g'(\tau_m) = B_{i+1}x'(\tau_m) \tag{2.3}$$

or equivalently,

$$w_i = B_{i+1}r_i, \tag{2.4}$$

where vectors r_i and w_i are given, with respect to the m most recent step vectors $\{s_k\}_{k=i-m+1}^{i}$ and the m most recent gradient difference vectors $\{y_k\}_{k=i-m+1}^{i}$, respectively, in the following forms:

$$\begin{aligned} r_i &= \sum_{j=0}^{m-1} s_{i-j}\left\{\sum_{k=m-j}^{m} L_k'(\tau_m)\right\}, \\ w_i &= \sum_{j=0}^{m-1} y_{i-j}\left\{\sum_{k=m-j}^{m} L_k'(\tau_m)\right\}, \end{aligned} \tag{2.5}$$

where $s_i \overset{\text{def}}{=} x_{i+1} - x_i$, and $y_i \overset{\text{def}}{=} g_{i+1} - g_i$,

$$L'_k(\tau_m) = (\tau_k - \tau_m)^{-1}\left[\frac{\tau_m - \tau_j}{\tau_k - \tau_j}\right], \quad k < m,$$
$$L'_m(\tau_m) = \sum_{j=0}^{m-1}(\tau_m - \tau_j)^{-1}, \tag{2.6}$$

and $\{L_k\}_{k=0}^m$ are the standard Lagrange polynomials.

The choice of the parameters, τ_k, for $k = 0, 1, 2, \ldots, m$, are chosen such that they depend on some metric of the following general form:

$$\phi_M(z_1, z_2) = \left[(z_1 - z_2)^T M(z_1 - z_2)\right]^{1/2}, \tag{2.7}$$

where **M** is a symmetric positive-definite matrix.

This metric is used to define the values $\{\tau_k\}_{k=0}^m$ used in computing the vectors r_i and w_i. The choices based on this metric are numerically better than the unit-spaced method since they take into account the spacing between the iterates using some norm of measure (see [5]).

Several choices were considered for the metric matrix **M**. For instance, if $\mathbf{M} = \mathbf{I}$, we obtain the following (for $m = 2$):

(a) accumulative algorithm A1: $\tau_2 = \|s_i\|_2$, $\tau_1 = 0$, $\tau_0 = -\|s_{i-1}\|_2$;

(b) fixed-point algorithm F1: $\tau_2 = 0$, $\tau_1 = -\|s_i\|$, $\tau_0 = -\|s_i + s_{i-1}\|_2$.

The new B-version BFGS formula is given by

$$B_{i+1}^{\text{Multistep}} = B_i + \frac{w_i w_i^T}{w_i^T r_i} - \frac{B_i r_i r_i^T B_i}{r_i^T B_i r_i}. \tag{2.8}$$

3. Extra BFGS updates

The BFGS formula corrects the eigenvalues of the Hessian approximation, B_i, although this correction is found inadequate in practice when the eigenvalues are large (see Liu and Nocedal [9]). It is, therefore, as El-Baali [3] states, desirable to further correct those values. It should, however, be stated that such values are not readily available and they can only be estimated rather than do any expensive computations for exactly obtaining them. A well-known formula can be employed to detect the presence of large eigenvalues as will be specified in the next section. It is thus envisaged that extra updates applied to B_i will introduce the desired corrections to the large eigenvalues.

We now present the ingredients of the extra update methods. Byrd et al. [2] propose for some $m \le n$ (n is the problem dimension) the following extra update:

$$G_{i+1}u^{(t)} = v^{(t)}, \quad 1 \le t \le m, \tag{3.1}$$

where $\{u^{(t)}\}_{t=1}^m$ and $\{v^{(t)}\}_{t=1}^m$ are any sequence of independent vectors. The authors were able to prove, under certain assumptions, global and superlinear convergence on convex functions. They use some finite differences to approximate the left-hand side of this last relation since the Hessian itself is not explicitly available. The convergence results do not necessarily hold since the above relation is approximated.

One specific choice made by Liu and Nocedal [9] for limited memory BFGS and one that is also adopted by El-Baali [3] for the vectors $u^{(t)}$ and $v^{(t)}$ is

$$u^{(t)} = s^{i-m+t}, \quad u^{(t)} = y^{i-m+t}, \quad 1 \le t \le m, \tag{3.2}$$

which are retained from the latest $m-1$ iterations. El-Baali conjectured that the curvature information obtained from (3.1) may be employed m times ($m > 1$) to improve the Hessian approximations.

The methods we develop here (as those of El-Baali [3]) generate, at each iteration, matrices built using only B_i, and the sequences $\{u^{(t)}\}_{t=1}^{m}$ and $\{v^{(t)}\}_{t=1}^{m}$ are readily available. We proceed at each iteration i by doing a single standard quasi-Newton single-step update as follows:

$$\overline{B_{i+1}^{(1)}} = \text{BFGS}\left(B_i, s_i, y_i\right). \tag{3.3}$$

The newly obtained Hessian approximation is used in the subsequent $m-1$ subiterations to do multistep updates as follows:

$$\overline{B_{i+1}^{(t+1)}} = \text{BFGS}\left(\overline{B_{i+1}^{(t)}}, u^{(t)}, v^{(t)}\right), \quad t = 1,2,\dots,m-1, \tag{3.4}$$

where m is a prescribed constant and

$$u^{(t)} = r_{i-m+t}, \qquad v^{(t)} = w_{i-m+t}, \tag{3.5}$$

and the vectors r_i and w_i are as in (2.5).

The $(m+1)$th update is done as

$$\overline{B_{i+1}^{(m+1)}} = \text{BFGS}\left(\overline{B_{i+1}^{(m)}}, s_i, y_i\right), \tag{3.6}$$

where the matrix obtained from (3.2)–(3.4) updates is the one used in computing the search direction and hence the new iterate x_{i+1}. When $i \le m$, then $m = i$. The secant equation is satisfied since the last update uses s_i and y_i while trying to exploit the numerical advantages of the multistep methods that use linear combinations of the most recent step and gradient difference vectors (r_i and w_i) in the update process. Thus, in addition to the secant equation, the following are satisfied:

$$\overline{B_{i+1}^{(t+1)}}\, r^{(i-m+t)} = w^{(i-m+t)}, \quad t = 1,2,\dots,m-1. \tag{3.7}$$

For $m = 1$, the proposed approach is equivalent to the standard 1-step BFGS method.

We now state a theorem that addresses the convergence properties of the derived technique.

Theorem 3.1. *Let x_0 be a starting point for a twice continuously differentiable objective function f. The sequence $\{x_i\}$ generated by (3.2)–(3.4), assuming B_0 is positive definite,* *with a line search for which the following two criteria are satisfied:*

$$f_{i+1} \leq f_i + \sigma_1 \alpha_i s_i^T g_i, \tag{3.8}$$

$$s_i^T g_{i+1} \geq \sigma_2 s_i^T g_i \tag{3.9}$$

(for $\sigma_1 \in (0,1/2)$ and $\sigma_2 \in (\sigma_1, 1)$) converges to the minimum x^*. There is also a constant $\lambda \in [0,1)$ such that

$$\begin{gathered} f_{i+1} - f^* \leq \lambda^k (f^1 - f^*) \quad \forall i, \\ \sum_{i=1}^{\infty} ||x_i - x^*|| < \infty. \end{gathered} \tag{3.10}$$

The proof is similar to that of El-Baali [3] and Byrd and Nocedal [1], and the reader is referred to those papers for details.

The following theorem shows that the sequence (3.2)–(3.4) possesses the superlinear convergence property.

Theorem 3.2. *Let x_0 be a starting point for a twice continuously differentiable objective function f. Assume that B_0 in (3.2)–(3.4) is positive definite and that the actual Hessian G satisfies a Lipschitz condition*

$$||G(x) - G(x^*)|| \leq \gamma ||x - x^*||, \tag{3.11}$$

where γ is a positive constant, for all x in the neighborhood of x^*. Assume also that a line search satisfying (3.7)-(3.8) is carried out such that if $\|x_i - x^*\|$ and $\|B_i s_i - G(x^*) s_i\| / \|s_i\|$ are sufficiently small, then the step length $\alpha_i = 1$ is used. The sequence $\{x_i\}$ thus generated converges superlinearly to x^*.

Again, the proof is a simple modification of the proof of El-Baali [3, Theorem 2].

It is left to tackle the issue of what m, the number of extra updates, may be chosen to be. We have tried two options: one is to determine a "good" value for m, based on the numerical test results. The other option, supported by the results obtained, is to make the choice on the basis of an estimate of the determinant of the updated Hessian approximation, B_{i+1}, as

$$\det(B_{i+1}) = \left(\frac{1}{\beta_i}\right) \det(B_i), \tag{3.12}$$

where

$$\beta_i = \frac{s_i^T B_i s_i}{s_i^T y_i}. \tag{3.13}$$

As β_i decreases, $\det(B_{i+1})$ increases. So, if β_i is sufficiently large, the BFGS update introduces slight correction to B_i. An extra update(s) is carried out if $\beta_i > 1$. Although El-Baali [3] suggests a relationship between β_i and the number of extra updates as follows:

$$\overline{m} = \min(m, \beta_i), \tag{3.14}$$

Table 4.1. Overall results (876 problems).

Method	Evaluations	Iterations	Time (s)
M1	86401 (100.00%)	73090 (100.00%)	39171.18 (100.00%)
A1	76164 (88.15%)	61335 (83.92%)	31474.71 (80.35%)
EA1	58695 (67.89%)	45492 (62.24%)	13308.42 (77.47%)

Table 4.2. Results for dimensions from 2 to 15 (440 problems).

Method	Evaluations	Iterations	Time (s)
M1	25589 (100.00%)	21648 (100.00%)	319.835 (100.00%)
A1	24494 (95.72%)	19525 (90.19%)	267.253 (83.56%)
EA1	22151 (86.56%)	16990 (76.48%)	266.1 (83.16%)

Table 4.3. Results for dimensions from 16 to 45 (240 problems).

Method	Evaluations	Iterations	Time (s)
M1	27058 (100.00%)	23844 (100.00%)	3429.78 (100.00%)
A1	22995 (84.98%)	19578 (82.11%)	2745.05 (80.04%)
EA1	21111 (78.02%)	17414 (73.03%)	2698.68 (78.68%)

(where $\overline{m}$ denotes the extra updates at a given iteration), our numerical tests have favored the choice $m = 2$ to that made in (3.14). The improvement incurred by our choice and that in (3.14) has not exceeded 8.4%, based on our experimentation on the new methods.

4. Numerical results and conclusions

Our numerical experiments of the new approach have been done with emphasis on algorithms that satisfy (3.2)–(3.4) by choosing values of $\{\tau_j\}_{j=0}^{2}$ that are consistent with successful choices published earlier (see [5, 6]). In particular, we have chosen algorithm A1 (see [5]), corresponding to the most successful *accumulative approach* as a benchmark, along with the standard single-step BFGS, to compare with the new algorithms for the same parameter and metric choices made for A1. The new algorithm is referred to as EA1.

It should be noted here that in our implementation, we are maintaining the matrix B_i in the factored form (LL^T).

Sixty functions classified into subsets of "low" ($2 \leq n \leq 15$), "medium" ($16 \leq n \leq 45$), and "high" ($46 \leq n \leq 80$) dimensions (as in [7]) were tested with four different starting points each. Some of the problems tested have variable dimensions and we tested our algorithms on several dimensions, depending on the specific nature of the problem. This has resulted in a total of 876 problems. The overall numerical results are given in Table 4.1. Tables 4.1–4.5 show the total results for each dimension. The tabulated results indicating the total number of function/gradient evaluations, the total iterations, and the

Table 4.4. Results for dimensions from 46 to 80 (134 problems).

Method	Evaluations	Iterations	Time (s)
M1	21146 (100.00%)	17431 (100.00%)	13426.87 (100.00%)
A1	18009 (85.17%)	14122 (81.02%)	10835.33 (80.70%)
EA1	15433 (72.98%)	11088 (63.61%)	10344.00 (77.04%)

Table 4.5. Results for dimensions from 81 to 100 (62 problems).

Method	Evaluations	Iterations	Time (s)
M1	12608 (100.00%)	10167 (100.00%)	21994.7 (100.00%)
A1	10666 (84.60%)	8110 (79.77%)	17627.08 (80.14%)
EA1	9411 (74.64%)	7083 (69.66%)	16211.81 (73.71%)

```
  Step 1: Set $B_0 = I$ and $i = 0$; evaluate $f(x_0)$ and $g(x_0)$;
Repeat
  Step 2: Compute the search-direction $p_i$ from $B_i p_i = -g_i$;
  Step 3: Compute $x_{i+1}$ by means of a line search from $x_i$ along $p_i$, using safeguarded cubic interpolation (see (3.7)-(3.8));
  Step 4: If $i = 0$, then
              {set $r_i = s_i$ and $w_i = y_i$}
          else
              {
                calculate the values $\tau_2 = \|s_i\|_2$, $\tau_1 = 0$, $\tau_0 = -\|s_{i-1}\|_2$;
                calculate the vectors $r_i$ and $w_i$ using (2.5);
                if $r_i^T w_i \leq 10^{-4} \|r_i\|_2 : \|w_i\|_2$ then
                  {$r_i$ and $w_i$ are not acceptable for updating $B_i$; set $r_i = s_i$ and $w_i = y_i$}
              };
  Step 5: If $i = 0$ and $n \geq 10$, then scale $B_0$ by the method of Shanno and Phua [11];
  Step 6: Update $B_i$ to produce $B_{i+1}$, using (3.2)–(3.4), with $m = 2$ (the number of extra updates); increment $i$.
Until $\|g(x_i)\|_2 < \varepsilon$ (where $\varepsilon$ is a problem-dependent tolerance).
```

Algorithm 4.1

total time provide a good mean to compare the efficiency on each of the subsets as well as on the complete set of the considered algorithms.

Method M1 corresponds to the standard BFGS. The results presented here show clearly that the new EA1 method exhibits a superior numerical performance, by comparison with the other algorithms derived earlier. Method EA1 is outlined in Algorithm 4.1.

The numerical evidence provided by the tests reported in Tables 4.1–4.5 demonstrates clearly that the new method EA1 shows significant improvements, when compared with the standard, single-step, BFGS method and A1. In particular it has yielded, on average, improvements in the range 26%–30%, on the problems with the highest dimensions. The results reported in Table 4.2 indicate that, while EA1 does appear to offer some improvement over the method from which it was developed (namely, A1), it is not so superior on such class of problems. We are currently investigating the issue of whether the numerical performance of similar methods can be improved further. The convergence properties of such methods need also be explored.

References

[1] R. H. Byrd and J. Nocedal, *A tool for the analysis of quasi-Newton methods with application to unconstrained minimization*, SIAM Journal on Numerical Analysis **26** (1989), no. 3, 727–739.

[2] R. H. Byrd, R. B. Schnabel, and G. A. Shultz, *Parallel quasi-Newton methods for unconstrained optimization*, Mathematical Programming **42** (1988), no. 2, 273–306.

[3] M. El-Baali, *Extra updates for the BFGS method*, Optimization Methods and Software **9** (1999), no. 2, 1–21.

[4] J. A. Ford and R.-A. Ghandhari, *On the use of function-values in unconstrained optimisation*, Journal of Computational and Applied Mathematics **28** (1989), 187–198.

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

[6] ———, *Multi-step quasi-Newton methods for optimization*, Journal of Computational and Applied Mathematics **50** (1994), no. 1–3, 305–323.

[7] J. A. Ford and A. F. Saadallah, *A rational function model for unconstrained optimization*, Numerical Methods (Miskolc, 1986), Colloquia Mathematica Societatis János Bolyai, vol. 50, North-Holland, Amsterdam, 1988, pp. 539–563.

[8] D. Goldfarb, *A family of variable-metric methods derived by variational means*, Mathematics of Computation **24** (1970), 23–26.

[9] D. C. Liu and J. Nocedal, *On the limited memory BFGS method for large scale optimization*, Mathematical Programming **45** (1989), no. 3, 503–528.

[10] D. F. Shanno, *Conditioning of quasi-Newton methods for function minimization*, Mathematics of Computation **24** (1970), 647–656.

[11] D. F. Shanno and K. H. Phua, *Matrix conditioning and nonlinear optimization*, Mathematical Programming **14** (1978), no. 2, 149–160.

Issam A. R. Moghrabi: Department of Computer Science, Faculty of Science, Beirut Arab University, Beirut 115020, Lebanon
E-mail address: imoghrabi@bau.edu.lb