New Non-Secant Implicit Update Method for Unconstrained Optimization

Issam Moghrabi

***Abstract*—** Multi-step methods are secant-like techniques of the quasi-Newton type that, unlike the classical methods, construct nonlinear alternatives to the quantities $\boldsymbol{s_i = x_{i+1} - x_i}$ and $\boldsymbol{y_i = g(x_{i+1}) - g(x_i)}$ in the so-called Secant equation $\boldsymbol{B_{i+1}s_i = y_i}$, where $\boldsymbol{x_i}$ denotes the current estimate for the minimum of the function being minimized ($\boldsymbol{(f(x)}$) and where $\boldsymbol{g(x)}$ denote the gradient of *f* and the *n* x *n* matrix $\boldsymbol{B_{i+1}}$ is the new approximation to the Hessian of *f*. Multi-step methods instead utilize data available from the *m* most recent iterations and thus create an alternative to the Secant equation with the intention of creating better Hessian approximation that induce faster convergence to the minimizer of *f*. Such methods now satisfy a relationship of the like $\boldsymbol{B_{i+1}r_i = w_i}$, for $\boldsymbol{r_i = s_i - \mu_i s_{i-1}}$ and $\boldsymbol{w_i = y_i - \mu_i y_{i-1}}$, for some carefully chosen $\boldsymbol{\mu_i}$. The methods, based on reported numerical results published in several research papers related to the subject, have introduced substantial savings in both iteration and function evaluation counts. Implicit updates were developed within the context of multi-step methods and have been implemented as a means to improve the quality of the interpolation curves in such methods. The idea that prior to updating the Hessian approximation at the current iteration, an extra update is carried out implicitly and in a cheap manner to use that update in the construction of the nonlinear polynomials that interpolate the recent *m* available step and gradient vectors used in the upcoming approximated Hessian computation so that the numerical performance of the multi-step methods improves further. In this paper, an improved implicit technique is introduced that is based on a variant of the Secant equation that is intended to include further already computed data (such as function values), that would, otherwise, be discarded and left unexploited, in the construction of a, hopefully, better Hessian approximation that yields, eventually, faster convergence. The implicit methods used here satisfy a relationship of the type $\boldsymbol{B_{i+1}r_i = r_i + \beta_i w_i}$. The numerical experimentations on the new methods are promising and open venue for further investigation of such techniques.

***Keywords*—** Implicit update methods, quasi-Newton Methods, multi-step methods, unconstrained optimization.

I. Introduction

THIS work addresses problems of the form:

$$minmize\ f(x), x \in R^n, where\ f: R^n \rightarrow R.$$

Conjugate gradient methods are a class of methods for solving unconstrained optimization problems where they gain significance for large-scale problems. Their storage requirements are modest compared to other methods as they do not require the storage of any matrices. While such methods converge in at most *n* iterations on quadratic functions for exact line searches, they are also used to minimize non-quadratic functions under relaxed line search conditions. For carrying out minimization on non-quadratic functions, the methods need to be restarted periodically or when certain criteria is met (see [22]). Such methods have been extensively studied where variations to the original method of Fletcher and Reeves [9] have been introduced (see, for example, [2,5,13,14]).

To minimize *f*, the sequence of iterates generated is given by

$$x_{i+1} = x_i + \alpha_i d_i, \qquad (1)$$

where α_i is a positive scalar and d_i is a CG search direction. The search direction is computed using the following recurrence

$$d_i = \begin{cases} -g_i, & for\ i = 0, \\ -g_i + \beta_i d_{i-1}, & for\ i \geq 1, \end{cases} \qquad (2)$$

for some scalar β_i and where g_i denotes the gradient of the function f evaluated at the point x_i. The search direction d_i is usually required to satisfy

$$d_i^T g_i < 0,$$

to ensure it is a descent one of the function *f*(*x*) at x_i. In order to guarantee global convergence, d_i may be required to satisfy the sufficient descent condition

$$d_i^T g_i \leq -\text{ʒ}\|g_i\|^2,$$

for some constant ʒ.

The specific choice made for β_i leads to different algorithms. Some well-known choices are

$$\beta_i^{FR} = \frac{\|g_i\|^2}{\|g_{i-1}\|^2}, \quad \beta_i^{PRP} = \frac{g_i^T(g_i - g_{i-1})}{\|g_{i-1}\|^2},$$

$$\beta_i^{HS} = \frac{g_i^T(g_i - g_{i-1})}{d_{i-1}^T(g_i - g_{i-1})}, \quad \beta_i^{LS} = \frac{g_i^T(g_i - g_{i-1})}{d_{i-1}^T g_{i-1}},$$

$$\beta_i^{DY} = \frac{g_i^T g_i}{d_{i-1}^T(g_i - g_{i-1})}$$

in addition to many other suggestions (see, for example, [13,16,20,22,23]). The above methods are, respectively, due to Fletcher-Reeves [9], Polak–Ribiére–Polyak [21,22], Hestenes-Stiefel [15], Liu-Storey [16] and Dai–Yuan [5].

This paper derives a new CG algorithm that uses a weighted multi-step update matrix in the computation of the search direction without having to retain the matrix in storage. The new method is inspired by the works of Anderi [2] and Ford et al. [13]. However, in this work, the derivation utilizes a multi-step preconditioning matrix while attempting to keep the computational cost to a minimum. Anderi [2] applies updates the identity matrix to build the conditioning matrix.

The authors is with the Gulf University for Science and Technology, Kuwait, +965-94453637, moughrabi.i@gust.edu.kw

Ford et al. [13] develop multi-step CG methods that do not involve any weighting matrix. Our derivation follows rather a different approach which seems to be numerically more impactful. The next section summarizes the multi-step methods idea. Section 3 presents the derivation of the new method. Finally, the numerical results are summarized and conclusions presented.

II. Multi-step quasi-newton methods

Quasi-Newton methods retain an approximation to the Hessian matrix that is updated at each iteration to reflect most recent changes in the data [3]. Given B_i, the current approximation to the Hessian, the new Hessian approximation, B_{i+1} is updated to satisfy the standard secant equation:

$$B_{i+1}s_i = y_i, \qquad (3)$$

where

$$s_i = x_{i+1} - x_i,$$

and

$$y_i = g_{i+1} - g_i.$$

The BFGS formula [3,10,11] is the mostly used update formula that satisfies the Secant equation especially that, in most reported research, it seems to work well with inexact line search algorithms [3,13,14]. This rank-two update that approximates the actual Hessian, in its standard form, is given by

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

In the standard secant equation, a straight line L is used to find a new iterate x_{i+1}, given the previous iterate x_i, while in the multi-step methods higher order polynomials are used.

Let $\{x(\tau)\}$ or X denote a differentiable path in R^n, where $\tau \in R$. The vector polynomial $x(\tau)$ satisfies

$$x(\tau_j^{(i-1)}) = x_{i-m+j}, \text{ for } j = 0, 1, \ldots, m,$$

for some distinct values $\left\{\tau_j^{(i-1)}\right\}_{j=0}^{m}$. The corresponding gradient points are interpolated by a similar polynomial $z(\tau)$ satisfying

$$z(\tau_j^{(i-1)}) = g_{i-m+j}, \text{ for j } = 0, 1, \ldots, m.$$

Then upon applying the Chain rule to the gradient vector $z(x(\tau)) \approx g(x(\tau))$ in order to find the derivative of the gradient g with respect to τ, we get

$$\frac{dg}{d\tau} = G(x(\tau))\frac{dx}{d\tau} \quad . \qquad (4)$$

Thus, at any point on the path X, the Hessian G must satisfy (4) for any value of τ. More specifically for $\tau = \tau_c$, where $\tau_c \in \mathcal{R}$. This will result in the following relation

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

By analogy with the secant equation, the aim is to derive a relation satisfied by the Hessian at the new iterate $x_{i+1.}$ We choose a value for the parameter τ, namely $\tau_{m,}$, that corresponds to the most recent iterate as follows

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

or

$$w_i = B_{i+1}r_i, \qquad (5)$$

where the vectors r_i and w_i are defined in terms of 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, as follows

$$r_i = \sum_{j=0}^{m-1} s_{i-j}\left\{\sum_{k=m-j}^{m} L'_k(\tau_m)\right\}$$

and

$$w_i = \sum_{j=0}^{m-1} y_{i-j}\left\{\sum_{k=m-j}^{m} L'_k(\tau_m)\right\}$$

for

$$L'_k(\tau_m) = (\tau_k - \tau_m)^{-1}\left[\frac{\tau_m - \tau_j}{\tau_k - \tau_j}\right], k < m$$

and

$$L'_m(\tau_m) = \sum_{j=0}^{m-1}(\tau_m - \tau_j)^{-1}.$$

are the standard Lagrange polynomials.

Ford and Moghrabi [10-12] examined several choices for the parameters $\{\tau_k\}_{k=0}^{m}$ where such choices influence the structure of the interpolating curve. Of the approaches considered in [12, 17, 18], we elect here to use the most numerically successful choice. The choice is based on, what the authors termed as, the Accumulative Approach.

The choices made for the parameters$\{\tau_k\}_{k=0}^{m}$ rely on some metric of the following form

$$\emptyset_M(z_1, z_2) = [(z_1 - z_2)^T M (z_1 - z_2)]^{1/2},$$

where M is a symmetric positive-definite matrix.

The Accumulative approach chooses one of the iterates, say x_j, as a base-point and sets the parameter τ_j corresponding to it to 0. Then, any value τ_k, corresponding to the point $x_{i-m+k+1}$ for any k except for $k=j$, is computed by distance accumulation (measured by the chosen metric Φ_M) between each two consecutive pair of points in the sequence from $x_{i-m+j+1}$ to $x_{i-m+k+1}$. Therefore, any value τ_k, for $k=0,1,\ldots,m$, is obtainable using

$$\tau_k = -\sum_{p=k+1}^{j}\emptyset_M(x_{i-m+p+1}, x_{i-m+p}), k < j,$$
$$= 0, k = j,$$
$$= -\sum_{p=j+1}^{k}\emptyset_M(x_{i-m+p+1}, x_{i-m+p}), k > j. \qquad (6)$$

This approach will yield values of τ that satisfy

$$\tau_k < \tau_{k+1}, for\ k = 0, 1, \ldots, m-1.$$

under the assumption that no consecutive points overlap.

Those values of the parameters $\{\tau_k\}$ are the ones used in computing the vectors $x'(\tau_m)$ and $g'(\tau_m)$ in (5) (or vectors r_i and w_i, respectively). The two vectors r_i and w_i are then used to compute the new Hessian approximation B_{i+1} satisfying (5).

It should be noted that different choices of the metric matrix M in Φ_M will result in different methods. Ford and Moghrabi [11,12] indicate that values of $m > 2$ do not seem to result in substantial numerical gains in performance due to the non-smoothness of the interpolant. Thus, $m = 2$ is chosen here and such methods are termed two-step methods as they utilize data from the two most recent iterations to update the Hessian approximation.

Choices investigated for the matrix M (see [12, 17,18]), include $M = I$, $M = B_i$, and $M = B_{i+1}$. The inverse Hessian approximation update generally satisfies:

$$H_{i+1}(y_i - \mu_{i-1}y_{i-1}) = s_i - \mu_{i-1}s_{i-1} \qquad (7)$$

or

$$w_i = B_{i+1} r_i$$

where

$$\mu_{i-1} = \frac{\delta_{i-1}^2}{2\delta_{i-1} + 1}.$$

and

$$\delta_{i-1} = \frac{\tau_2^{(i-1)} - \tau_1^{(i-1)}}{\tau_1^{(i-1)} - \tau_0^{(i-1)}}.$$

For our numerical tests, the particular choices used for the τ values are (corresponding to choosing $M = I$ in (6))

$$\tau_0 = -(\|s_i\|_2 + \|s_{i-1}\|_2), \tau_2 = 0, \text{and } \tau_1 = -\|s_i\|_2 .$$

This, hence, gives

$$\delta = \frac{\|s_i\|}{\|s_{i-1}\|}. \tag{8}$$

Equation (8) may be generalized by introducing a scaling factor, $\gamma \geq 0$ (see [13]) that provides more control in this context since by setting the scalar to zero convenient switching to the standard secant equation one-step update method is easy. Therefore,

$$\delta = \gamma \frac{\|s_i\|}{\|s_{i-1}\|}. \tag{9}$$

The multi-step B-version BFGS formula is given by

$$B_{i+1}^{MS} = B_i + \frac{w_i w_i^T}{w_i^T r_i} - \frac{B_i r_i r_i^T B_i}{r_i^T B_i r_i}. \tag{10}$$

III. A NEW MULTI-STEP PRECONDITIONED CG METHOD (MSPCG)

The search direction considered in this paper takes the form

$$d_i = -\sigma_i g_i + \beta_i s_{i-1}, \tag{11}$$

(see [2]) where σ_i can be chosen to be a scalar or some positive definite matrix. For example, if $\sigma_i = 1$, then (11) is equivalent to (2). If, however, σ_i is chosen to be some approximation to the inverse of the Hessian matrix, then d_i becomes a combination of the quasi-Newton and the conjugate gradient directions. In this work, we consider the latter case.

When applied to the minimization of a quadratic function, the search directions generated by the linear CG methods satisfy the conjugacy condition

$$d_i^T A d_j = 0, \forall i \neq j, \tag{12}$$

where A is the positive definite matrix Hessian matrix of the function. As for non-quadratic functions, relation (12) may be replaced by (see [13])

$$d_i^T y_{i-1} = 0. \tag{13}$$

Now, using the mean value theorem, there exists some $\omega \in (0,1]$ such that

$$d_i^T y_{i-1} = \alpha_{i-1} d_{i-1} \nabla^2 f(x_{i-1} + \omega \alpha_{i-1} d_{i-1}) d_{i-1}.$$

Perry [20] studied acceleration options to the CG methods which exploit the advantages of the quasi-Newton methods. His approach uses the secant equation (3) and given that the quasi-Newton search direction is given by $d_i = -H_i g_i$, then Perry replaced (13) with

$$d_i^T y_{i-1} = -g_i^T s_{i-1}. \tag{14}$$

Using (14), we obtain

$$d_i^T y_{i-1} = -g_i^T (H_i y_{i-1}),$$

or (from (7))

$$d_i^T y_{i-1} = -g_i^T r_{i-1} - \mu_{i-1} g_i^T H_i y_{i-2}.$$

This yields

$$d_i^T w_{i-1} = -\epsilon g_i^T r_{i-1}, \tag{15}$$

for some $\varepsilon \geq 0$ that serves as a scaling factor to impose conjugacy.

Upon substituting (compare to (11))

$$d_i = -\sigma_i g_i + \beta_i s_{i-1} \tag{16}$$

in (15), we obtain

$$-\sigma_i w_{i-1}^T g_i + \beta_i w_{i-1}^T s_{i-1} = -\, g_i^T r_{i-1},$$

thus yielding an expression for β_i as follows

$$\beta_i = \frac{g_i^T [\sigma_i w_{i-1} - \varepsilon r_{i-1}]}{s_{i-1}^T w_{i-1}}. \tag{17}$$

If $\varepsilon = 0$, then (17) reduces to the choice of β_i obtained in [13].

We proceed with our derivation first with the choice $\sigma_i = H_i$ (for $\varepsilon = 1$). To complete the implementation details of the algorithm, the quantity (see (11))

$$d_{i+1} = -H_{i+1} g_{i+1},$$

need to be computed efficiently. In specific,

$$z_{i+1} = H_{i+1} g_{i+1} \tag{18}$$

need to be computed without having to store the matrix H_{i+1} or having to carry out any matrix-vector multiplication, thus maintaining the spirit of the CG methods.

From the H-multi-step version of the BFGS formula, given by

$$H_{i+1} = H_i - \frac{r_i w_i^T H_i + H_i w_i r_i^T}{w_i^T r_i} + \left(1 + \frac{w_i^T H_i w_i}{w_i^T r_i}\right) \frac{r_i r_i^T}{w_i^T r_i}, \tag{19}$$

it follows that

$$H_{i+1} g_{i+1} = H_i g_{i+1} - \frac{r_i^T g_{i+1}}{w_i^T r_i} v_i + \left[\left(1 + \frac{w_i^T v_i}{w_i^T r_i}\right) \frac{(r_i^T g_{i+1})}{w_i^T r_i} - \frac{(v_i^T g_{i+1})}{w_i^T r_i}\right] r_i, \tag{20}$$

where

$$v_i \cong H_i w_i.$$

Using (19), we obtain an expression for v_i in (20) as follows

$$v_i = w_i - \frac{r_{i-1}(w_{i-1}^T w_i) + w_{i-1}(r_{i-1}^T w_i)}{w_{i-1}^T r_{i-1}} + \left(1 + \frac{w_{i-1}^T w_{i-1}}{w_{i-1}^T r_{i-1}}\right) \frac{r_{i-1}(r_{i-1}^T w_i)}{w_{i-1}^T r_{i-1}}, \tag{21}$$

and

$$H_i g_{i+1} = v_i + \beta_i s_{i-1} - d_i + \mu_{i-1} H_i y_{i-1},$$

where μ_{i-1} is as in (7).We need now to be able to compute $H_i y_{i-1}$.

Now,

$$H_i y_{i-1} = H_i g_i - H_i g_{i-1} = z_i - H_i g_{i-1},$$

for z_i as in (18). From (19), we have

$$H_i g_{i-1} = z_{i-1} - \frac{r_{i-1}(w_{i-1}^T z_{i-1}) + v_{i-1}(r_{i-1}^T g_{i-1})}{w_{i-1}^T r_{i-1}} + \left(1 + \frac{w_{i-1}^T v_{i-1}}{w_{i-1}^T r_{i-1}}\right) \frac{r_{i-1}(r_{i-1}^T g_{i-1})}{w_{i-1}^T r_{i-1}}. \tag{22}$$

This completes the details of the computation.

As for the case $\sigma_i = 1$ in (11), the search direction is expressed simply as

$$d_{i+1} = -g_{i+1} + \beta_{i+1} s_i,$$

with β_{i+1} given by (17).

Al-Baali [1] proved the global convergence of the Fletcher-Reeves method on general functions with inexact line search. Dai and Yuan [5] developed a CG method that is based on the secant condition and proved global convergence of their method. In order to guarantee the convergence of his algorithm, requires that the step size α_i in (1) is accepted if it satisfies the Wolfe conditions [28] (see [1,2,5,13, 14,25,26,27]):

$$f(x_i + \alpha_i d_i) - f(x_i) \leq \rho_1 \alpha_i d_i^T g_i, \quad (23)$$
$$g(x_i + \alpha_i d_i)^T d_i \geq \rho_2 d_i^T g_i, \quad (24)$$

where $0 < \rho_1 \leq \rho_2 < 1$.

We now present the following theorem that highlights the conditions that ensure the search direction is downhill.

Theorem 1. Suppose that α_i in (1) satisfies the Wolfe conditions (23) and (24); If $w_i^T r_i > 0$, then $-H_{i+1} g_{i+1}$ given by (20) is a descent direction.

Proof. Given that $d_0 = -g_0$, it follows that $g_0^T d_0 = -\|g_0\|^2 \leq 0$.

$$H_{i+1} g_{i+1} = H_i g_{i+1} - \frac{r_i^T g_{i+1}}{w_i^T r_i} v_i + \left[\left(1 + \frac{w_i^T v_i}{w_i^T r_i}\right) \frac{(r_i^T g_{i+1})}{w_i^T r_i} - \frac{(v_i^T g_{i+1})}{w_i^T r_i}\right] r_i.$$

As for subsequent iterations, pre-multiplying (20) by $-g_{i+1}^T$ gives

$$-g_{i+1}^T H_{i+1} g_{i+1} = \frac{1}{(w_i^T r_i)^2} \begin{bmatrix} -g_{i+1}^T H_i g_{i+1} (w_i^T r_i)^2 + 2(r_i^T g_{i+1})(w_i^T r_i)(g_{i+1}^T v_i) \\ -\left(g_{i+1}^T r_i\right)^2 (w_i^T r_i) - \left(r_i^T g_{i+1}\right)^2 (w_i^T v_i) \end{bmatrix}.$$

If the inequality $u^T q \leq \frac{1}{2}(\|u\|^2 + \|q\|^2)$ is applied to the second term above with $u = (w_i^T r_i) g_{i+1}$ and $q = (r_i^T g_{i+1})\, v_i$, we obtain

$$g_{i+1}^T H_{i+1} g_{i+1} \geq -\frac{(r_i^T g_{i+1})^2}{w_i^T r_i}.$$

If $w_i^T r_i > 0$, it follows that $-g_{i+1}^T H_{i+1} g_{i+1}$ is negative.

IV. Numerical Computations

Our numerical results are benchmarked against Anderi's [2] SCALCG. Our computational experiments do not include a comparison with the methods in [13] since those methods do not use a weighting matrix in the computation of the CG direction vector, contrary to what we are doing here. The results reported in Table 2 are for different choices of the parameter γ in (9) in order to determine its effect on the numerical performance of the method. The values that appear in Table 2 for γ correspond to 0, 0.5 and 1, respectively. Other values have been considered but are not reported here as they bear no significant changes. The numbers reported indicate iteration/function and gradient evaluations counts, respectively. The coding was done using C++ on a 64-bit machine with i7-3770, 3.4 GHZ CPU. Table 1 contains the problem set used in testing the derived method against SCALCG. The problems are primarily those found in [20].

TABLE I
Test problems

P	Function	Dimension
	Name	
1	Extended	*n* = 100000
2	Rosenbrock	*n* = 100000
3	Extended	*n* = 100000
4	Powell Singular	*n* = 10000
5	Trigonometric	*n* = 2
6	Function	*n* = 4
7	Oren Function	*n* = 2
8	Cube Function	*n* = 3
9	Wood Function	*n* =2
10	Beale Function	*n* = 2
11	Helical Valley	*n* = 1000
12	Penalty I Function	*n* = 1000

TABLE II
ITERATION AND FUNCTION EVALUATIONS COUNT

problem	γ	MSPCG	Anderi's	Ford et al.
1	0	22/70	21/69*	25/71
	½	24/75		
	1	28/101		
2	0	25/73	21/69*	22/69
	½	31/78		
	1	30/76		
3	0	61/151*	70/201	66/168
	½	failed		
	1	77/163		
4	0	390/1701*	390/1770	379/1709
	½	392/1707		
	1	373/1696*		
5	0	21/41	31/51	24/40*
	1/2	25/69		
	1	49/91		
6	0	71/129*	68/141	69/134
	1/2	72/147		
	1	failed		
7	0	4/11	4/12	4/10*
	1/2	4/10		
	1	4/10*		
8	0	11/201	12/193*	14/199
	1/2	12/202		
	1	14/210		
9	0	7/23	8/21	8/20
	1/2	6/19*		
	1	7/20		
10	0	8/21	7/22	7/19*
	1/2	4/11		
	1	7/19*		
11	0	401/991	368/909	361/906*
	1/2	failed		
	1	338/812		
12	0	17/77	17/78	15/71
	1/2	15/70*		
	1	16/78		
totals		951/3273	1017/3536	994/3416
scores		7	3	3

The numerical evidence, reported in Table 2, reveals that the new method MSPCG shows some improvements over Anderi's [2] on several problems. The star appearing next to a score indicates a win on that problem. The last row of Table 2 reports the total score count for each method.

Both methods use exactly the same line search implementation with choices $\rho_1 = 0.0001$ and $\rho_1 = 0.88$ in (23) and (24). The termination condition used for both methods is

$$\|g(x_i)\| \leq 10^{-5}.$$

Although execution times are not included in our reported results but, generally speaking, for most cases for which the function/gradient evaluations are less for a given algorithm, the time taken by the CPU is less for that algorithm. However, when the evaluation counts almost form a tie, SCALCG execution time turns out to be faster by about 3.61%.

While experimenting on the methods, especially for large problems, both methods were restarted periodically using Powell's [22] test to measure the degree of orthogonality

$$\|g_{i+1}^T g_k\| \geq 0.2\|g_{i+1}\|^2. \qquad (25)$$

Whenever (25) is satisfied at step i, the restart is applied. We used Anderi's [2] restart search direction for SCALCG. For our algorithm, we restarted with $\beta_i = 0$ and $H_i = \sigma_i I$ in (16), for $\sigma_i = \frac{s_i^T s_i}{s_i^T y_i}$ (see [2]). This situation has not been encountered very frequently in our numerical tests. Wolfe conditions (23) and (24) ensure that $s_i^T y_i > 0$ Anderi's SCALCG [2] search direction is downhill. In our case, if $r_i^T w_i > 0$, then the search direction is a descent one. Nevertheless, due to the approximation used in (20), numerical safeguarding remains a must.

V.Conclusion

In this paper, a new weighted Conjugate Gradient method is developed. The method generates search directions that are a combination of the multi-step quasi-Newton and CG vectors. It attempts to utilize the advantages of both methods to accelerate convergence of the CG algorithms. The method requires a few additional vectors than SCALCG [2] and other similar CG methods. This extra cost seems to incur reasonable savings in computational costs, especially on large problems.

We are currently investigating other choices for the weighting matrix to determine whether the numerical performance of similar methods can be improved further. There also remains the issue of developing automatic restart criteria that provides appropriate switching among several options similar to what was done in [1]. The global convergence properties of such methods are under consideration.

References

[1] Al-Baali, M. "New property and global convergence of the Fletcher–Reeves method with inexact line searches*", IMAJ. Numer. Anal.,* 5, pp. 122–124 (1985).

[2] Anderi, N. "A scaled BFGS preconditioned conjugate gradient algorithm for unconstrained optimization", *Applied Mathematics Letters,* 20, pp. 645-650 (2007).

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

[4] Byrd, R.H., Schnabel, R.B., and Shultz, G.A., "Parallel quasi-Newton methods for unconstrained optimization", *Math. Programing,* 42, pp. 273-306 (1988).

[5] Dai, Y.H. and Yuan Y., "A nonlinear conjugate gradient method with a strong global convergence property", *SIAM J. Optim.*, 10, pp. 177–182 (1999).

[6] Dennis, J.E. and Schnabel, R.B., "Least change secant updates for quasi-Newton methods", *SIAM Review,* 21, pp. 443-459 (1979).

[7] Fletcher, R., *Practical Methods of Optimization* (second edition), Wiley, New York, (1987).

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

[9] Fletcher, R. and Reeves, C., "Function minimization by conjugate gradients", *Computer J.,* 7, pp. 149–154 (1964).

[10] Ford, J.A. and Moghrabi I.A.R., "Using function-values in multi-step quasi-Newton methods", *J. Comput. Appl. Math*., 66, pp. 201-211 (1996).

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

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

[13] Ford, J.A., Narushima, Y. and Yabe, H., "Multi-step nonlinear conjugate gradient methods for unconstrained minimization", *Comput. Optim. Appl.*, 40, pp. 191-216 (2008).

[14] Hager W. and Zhang H.C., "A new conjugate gradient method with guaranteed descent and an efficient line search", *SIAM J.Optim.*, 16, pp.170–192 (2005).

[15] Hestenes, M.R. and Stiefel, E., "Methods of conjugate gradients for solving linear systems", *J. Res. Nat. Bur. Stan. Sec. B*, 48, pp. 409–436 (1952).

[16] Liu, Y. and Storey, C., "Efficient generalized conjugate gradient algorithms, part1:theory", *J. Optim. Theory Appl*., 69 pp. 129–137 (1991).

[17] Moghrabi, I.A.R., "Numerical experience with multiple update quasi-newton methods for unconstrained optimization", *Journal of Mathematical Modeling and Algorithms*, 6, pp. 231-238(2007).

[18] Moghrabi, I.A.R., "Implicit extra-update multi-step quasi-newton methods", Int. J. Operational Research, 28, pp. 69-81(2017).

[19] Moré, J.J., Garbow, B.S., Hillstrom, K.E., "Testing unconstrained optimization software", *ACM Trans. Math. Softw*., **7**, 17–41 (1981).

[20] Perry, A., "A modified conjugate gradient algorithm", *Oper. Res.* ,26, pp. 1073-1078 (1978).

[21] Polak, E. and Ribiére G., "Notesurla convergence de directions conjuguées", *Rev. Francaise Infomat Recherche Operatonelle*, 3e Année, 16, pp. 35–43 (1969).

[22] Polyak, B. T., "The conjugate gradient method in extreme problems", *USSR Comp. Math. Phys.*, 9, pp. 94–112 (1969).

[23] Powell, M.J.D., "Restart procedures for the conjugate gradient method", *Math. Program.* , 12, pp. 241–254 (1977).

[24] Salane, D. and Tewarson, R.P., "On Symmetric Minimum Norm Updates", *IMA Journal of Numerical Analysis*, 9, 1, pp. 235-240 (1983).

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

[26] Shanno, D.F. and Phua, K.H., "Matrix conditioning and nonlinear optimization", *Math. Programming* ,14, pp. 149-160 (1978).

[27] Shanno, D.F., "On the convergence of a new conjugate gradient algorithm", *SIAM J. Numer. Anal.,* 15, pp. 1247–1257 (1978).

[28] Wolfe, P., "Convergence conditions for ascent methods II: Some corrections", *SIAM Rev.*, 13, pp. 185-188 (1971).