International Journal of Pure and Applied Mathematics
Volume 119 No. 1 2018, 131-143
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
doi: 10.12732/ijpam.v119i1.11

ijpam.eu

CURVATURE-BASED QUASI-NEWTON METHODS FOR OPTIMIZATION

Issam A.R. Moghrabi

Department of M.I.S.
College of Business Administration
Gulf University for Science and Technology (GUST)
P.O. Box 7207, Hawally 32093, KUWAIT

Abstract: This paper presents a framework model for building minimum curvature Multi-step methods. The Multi-step methods were derived in [6,7] and have consistently outperformed the traditional quasi-Newton methods that satisy the classical linear Secant equation. The methods derived here aim at improving further the Multi-step methods by ensuring that the interpolating curve used in updating the Hessian approximation has minimum a curvature. The model used in the derivation of such methods utilizes a free parameter that is employed as a tuning variable. The idea of minimizing the curvature of the interpolant was introduced in [2]. The encouraging results justify the investigation of these methods further. The algorithms derived are benchmarked against some of the most successful quasi-Newton methods such as the standard BFGS and the methods derived in [2,5,6,7]. The results of the numerical experiments indicate that the improvements obtained are substantially good and that the methods are indeed promising.

AMS Subject Classification: 65K10
Key Words: unconstrained optimization, conjugate gradient methods,variable metric methods

1. Introduction

This work directs attention to problems of the form:

$$\text{minimize } f(x) \text{ (where } f : R^n \to R).$$

Received: April 25, 2018
Revised: June 11, 2018
Published: June 13, 2018

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

$$B_{i+1}s_i = y_i$$

where

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

and

$$y_i = g_{i+1} - g_i \tag{2}$$

The BFGS formula is numerically the most successful rank-two formula [12,13,14,15] that approximates the true Hessian matrix. It is given by

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

and for the inverse $H_{i+1}(= B_{i+1}^{-1})$

$$H_{i+1}^{BFGS} = H_i + \left[1 + \frac{y_i^T H_i y_i}{s_i^T y_i}\right] \frac{s_i s_i^T}{s_i^T y_i} - \frac{s_i y_i^T H_i + H_i y_i s_i^T}{s_i^T y_i}.$$

The paper starts with an account of the multi-step methods that form the base of the methods derived in this paper. Then the general rational model is introduced and then used in the derivation of the new minimum curvature algorithms. We finally present the numerical comparisons.

2. Multi-Step Methods

Unlike the standard quasi-Newton methods in a straight line L is used to find a new iterate x_{i+1}, given the current iterate x_i, multi-step quasi-Newton methods exploit higher order polynomials in such computations. Let $\{x(\tau)\}$ or X denote a differentiable path in R^n, where $\tau \in R$. Applying the Chain rule to the gradient vector $g(x(\tau))$ in order to find the derivative of the gradient g with respect to τ gives

$$\frac{dg}{d\tau} = G(x(\tau))\frac{dx}{d\tau}. \tag{3}$$

Therefore, at any point on the path X the Hessian G must satisfy (3) for any value of τ, specifically for $\tau = \tau_c$, where τ_c is a constant scalar and $\tau_c \in R$. This will result in the so-called "Newton Equation" ([6],[12]):

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

To derive a relation satisfied by the Hessian at x_{i+1}, a value for τ is chosen, namely τ_m, that corresponds to the most recent iterate in the Newton equation as follows (where $\widehat{g}'(\tau_m) \approx g'(\tau_m)$)

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

or equivalently,

$$w_i = B_{i+1}r_i, \tag{4}$$

where the 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

$$r_i = \sum_{j=0}^{m-1} s_{i-j}\{\sum_{k=m-j}^{m} L'_k(\tau_m)\}\ , \tag{5}$$

and

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

where s_i and y_i are as in (1) and (2), respectively, and

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

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

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

Ford and Moghrabi [7] introduced choices for the parameters $\{\tau_k\}_{k=0}^{m}$, that influence the shape of the interpolating curve. The natural choice for the parameter τ is $\tau_k = k - m + 1$, or any similar scaling of this, for $k = 0, 1, 2, ..., m$. In [7], it was proposed that the parameters τ_k for be chosen such that they depend on some metric used to take into account the spacing among the iteration points. The metric takes the form

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

where M is a symmetric positive-definite matrix.

This metric is then used to define the $\{\tau_k\}_{k=0}^{m}$ values used in the computation of the vectors r_i and w_i.

One option for calculating $\{\tau_k\}_{k=0}^{m}$ is to choose one of the iterates, say x_j, as a "base-point". This point corresponds to some value of τ, which is assumed to be the origin and therefore this value τ_j is set to 0 . Then, any value τ_k corresponds to the point $x_{i-m+k+1}$ for $k \neq j$. This approach locates the other iterates $\{x_{i-m+k+1}\}_{k=0}^{m}$, except for $k = j$, by accumulating the distance (measured by the chosen metric ϕ_M) between each two consecutive pair of points in the sequence from $x_{i-m+j+1}$ to $x_{i-m+k+1}$. Moreover, the values of τ_k are negative for $k < j$ and positive otherwise. Therefore, in this approach we can find any value τ_k, for $k = 0, 1, ..., m$, using

$$\tau_k = \left\{ \begin{array}{lr} -\sum_{p=k+1}^{j}[\phi_M(x_{i-m+p+1}, x_{i-m+p})], k < j, \\ 0, & k = j, \\ \sum_{p=j+1}^{k}[\phi_M(x_{i-m+p+1}, x_{i-m+p})], k > j. \end{array} \right\} \tag{6}$$

This approach will yield values $\{\tau_k\}_{k=0}^{m}$, satisfying the following property

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

for distinct non overlapping $\tau-$values.

The particular choice of the matrix M results in different algorithms. Moreover, the numerical results reported in [7] seem to favour the choice $m = 2$ over larger values of m. This may be largely attributed to the non-smoothness of the interpolant. The methods derived here choose $m = 2$. Thus, the update done at each iteration for these methods satisfies

$$H_{i+1}(y_i - \frac{\delta^2}{2\delta+1} y_{i-1}) = s_i - \frac{\delta^2}{2\delta+1} s_{i-1} \tag{7}$$

where

$$\delta = \frac{\tau_2 - \tau_1}{\tau_1 - \tau_0}.$$

For $m = 2$ and $M = I$, for example, the parameter values are given by $\tau_2 = \|s_i\|_2, \tau_1 = 0, \tau_0 = -\|s_{i-1}\|_2$.

The new B-version **BFGS** formula is given as

$$B_{i+1}^{Multi-Step} = B_i + \frac{w_i w_i^T}{w_i^T r_i} - \frac{B_i r_i r_i^T B_i}{r_i^T B_i r_i} \ , \tag{8}$$

with similar structure for the H-version, obtained by replacing the vectors s_i and y_i with r_i and w_i,respectively.

2.1. Minimum Curvature Algorithms

The idea of determining the parameters defining the curve $\{\tau_k\}_{k=0}^{m}$, such that a minimum curvature is yielded, was first proposed in [2]. Here, we carry on with the idea further and propose a rational form that hosts a free parameter at our disposal that is chosen so that minimum curvature is yielded at some chosen $\tau-$point for the interpolating curve in the space of the variables.

The model is given as follows:

$$u(\tau,\theta) = \frac{q(\tau)}{(1+\theta\tau)}, \tag{9}$$

where the vector $u(\tau,\theta)$ represents either a model for $x(\tau,\theta)$ or $\widehat{g}^{'}(\tau_m)(\approx g'(\tau_m))$, the scalar θ serves as a tuning parameter to control the curvature, and $q(\tau)$ is a quadratic polynomial expressed as

$$q(\tau) = \sum_{k=0}^{2} L_k(1+\theta\tau_k)x_{i-m+k+1}\ , \tag{10}$$

where $L_j(\tau)$ is the Lagrange polynomial of degree two associated with the abscissae $\{\tau_k\}_{k=0}^{2}$.

Then for a specified set of distinct τ values $\{\tau_k\}_{k=0}^{2}$ in the two-step method, the curve $x(\tau,\theta)$ should satisfy:

$$x(\tau_k,\theta) = x_{i+k-1} \quad , \ for \quad k = 0, 1, 2. \tag{11}$$

The update matrix in this case satisfies

$$\widehat{g}^{'}(\tau_2,\theta) = B_{i+1}x'(\tau_2,\theta). \tag{12}$$

We now proceed in the derivation of expressions for the parameter θ. From (7) and (9) we obtain

$$x'(\tau,\theta) \equiv \frac{(1+\theta\tau)q' - \theta q}{(1+\theta\tau)^2} \tag{13}$$

and

$$x'(\tau_2,\theta) = \frac{s_i[(\tau_1-\tau_0)(-\tau_1-\tau_0+2\tau_2) + \theta(\tau_2^2-\tau_1\tau_0)] - s_{i-1}[(\tau_2-\tau_1)^2(1+\theta\tau_0)]}{(1+\theta\tau_2)(\tau_1-\tau_0)(\tau_2-\tau_0)(\tau_2-\tau_1)}\ . \tag{14}$$

If we define $\tau_{i,,j} \overset{def}{=} \tau_i - \tau_j$, then

$$x''(\tau_2,\theta) = \frac{\theta^2(s_i\alpha + s_{i-1}\beta) + \theta(s_i\gamma + s_{i-1}\lambda) + (2s_i\tau_{1,0} + 2s_{i-1}\tau_{2,1})}{(1+\theta\tau_2)^2\tau_{1,0}\tau_{2,1}\tau_{2,0}}\ , \quad (15)$$

for

$\alpha = 2\tau_{1,0}\tau_0\tau_1, \beta = 2\tau_0\tau_1\tau_{2,1}, \gamma = 2(\tau_1^2 - \tau_0^2)$ and $\lambda = 2(\tau_1^2 + \tau_0\tau_1 - \tau_0\tau_2 - \tau_1\tau_2)$.

Then, upon defining

$$\sigma_j \overset{def}{=} \|s_j\|_M^2 \geq 0, \text{ and } \mu_j \overset{def}{=} s_{j-1}^T M s_j, \quad (16)$$

the minimum condition for the curvature function $\phi(\tau_2,\theta) = \|x\text{"}(\ \tau_2,\theta)\|_M^2$ can be expressed as

$$\begin{aligned}&\phi(\tau_2,\theta)\\ &= \varsigma^{-1}[\theta^3(4\upsilon - \varkappa\tau_2) + \theta^2(3\varkappa - 2\varrho\tau_2) + \theta(2\varrho - 3\xi\tau_2) + (\xi - 4\iota\tau_2)] = 0. \quad (17)\end{aligned}$$

which is a cubic equation in θ and where

$$\begin{aligned}
\varsigma =&(1+\theta\tau_2)^5(\tau_{1,0}\tau_{2,1}\tau_{2,0})^2,\\
\nu =&\alpha^2\sigma_i + \beta^2\sigma_{i-1} + 2\alpha\beta\mu_i,\\
\varkappa =&2\alpha\gamma\sigma_i + 2\beta\lambda\sigma_{i-1} + 2(\alpha\lambda + \beta\gamma)\mu_i,\\
\varrho =&(\gamma^2 + 4\tau_{1,0}\alpha)\sigma_i + (\lambda^2 + 2\tau_{2,1}\beta)\sigma_{i-1} + 2(\gamma\lambda + 2\alpha + 2\tau_{1,0}\beta)\mu_i,\\
\xi =&4\tau_{1,0}\gamma\sigma_i + 4\tau_{2,1}\lambda\sigma_{i-1} + 4(\tau_{2,1}\gamma + \tau_{1,0}\lambda)\mu_i,\\
\iota =&(2\tau_{1,0})^2\sigma_i + (2\tau_{2,1})^2\sigma_{i-1} + 8\tau_{1,0}\tau_{2,1}\mu_{i.}
\end{aligned}$$

2.2. Algorithm C1

For this algorithm, the focus is on determining the minimum curvature at τ_2, corresponding to the most recent iterate, for some defined τ_0 and τ_1 values. We choose the origin $\tau_2 = 0$. From (15) it follows that

$$\begin{aligned}&\phi(\tau_2,\theta) =\\ &\eta\left[\theta^3\left(16\tau_0^2\tau_1^2\right) + \theta^2\left(24\tau_0\tau_1\left(\tau_1+\tau_0\right)\right) + \theta(8\left(\tau_1+\tau_0\right)^2 + 16\tau_0\tau_1) + 8\left(\tau_1+\tau_0\right)\right].\end{aligned}$$

for

$$\eta = \frac{\tau_{1,0}^2\sigma_i + \tau_1^2\sigma_{i-1} + 2\tau_1\tau_{1,0}\mu_i}{\tau_0\tau_1\tau_{1,0}}.$$

This yields three real roots as follows

$$\theta_1 = (\|s_{i-1}\|_2 + \|s_i\|_2)^{-1},\ \theta_2 = (\|s_i\|_2)^{-1}$$

and

$$\theta_3 = -1/2\left((\|s_{i-1}\|_2 + \|s_i\|_2)^{-1} + (\|s_i\|_2)^{-1}\right).$$

The first two values are points of singularity. As for the third root, the curvature expression is given by

$$\phi\,(\tau_2, \theta_3) = -4[7\,(\tau_1 + \tau_0)^2 - \frac{4}{\tau_0\tau_1}].$$

Whether θ_3 corresponds to a minimum or not depends on the values chosen for τ_0 and τ_1 and this needs to be numerically tested before it is accepted on each iteration. The value of θ_3 corresponds to a minimum if values of the form $\tau_1 = -\|s_i\|_2$ and $\tau_0 = -\|s_i\|_2 - \|s_{i-1}\|_2$ are used and when the magnitude of the step vectors is less than one. The update matrix for this algorithm satisfies (12). If no such minimum can be computed, a plain 2-step method iteration is carried out to satisfy (4).

2.3. Algorithm C2

In **C1**, θ is derived such that minimium curvature for the interpolant is obtained. In this method, a specific member of the model in (9) is the focus where by one of the τ-values is determined independently of the free parameter θ such that a minimum curvature is yielded. The choices made here are $\tau_0 = 0, \tau_2 = 1$ and $\theta = 0$ and τ_1 is what we seek to determine. Consequently, (14) reduces to:

$$x'(\tau_1, 0) = (s_i\tau_1(1-\tau_1)^{-1} + s_{i-1}\tau_1^{-1}(1-\tau_1)).$$

Given the chosen values for the $\tau-$parameters, (15) consequently is given as

$$x''(\tau_1, 0) = \frac{2(s_i\tau_1 + s_{i-1}(\tau_1 - 1))}{\tau_1(1-\tau_1)}.$$

The corresponding curvature function is given as

$$\phi\left(\tau_1,\theta\right)=x''(\tau_1,0)^T M x''(\tau_1,0)=\frac{4\left(\tau_1^2\sigma_i+\left(\tau_1-1\right)^2\sigma_{i-1}+2\tau_i\left(\tau_1-1\right)\mu_i\right)}{\tau_1^2\left(1-\tau_1\right)^2}.$$

It follows that

$$\phi' \overset{def}{=} \frac{d\phi(\tau_1)}{d\tau_1}=\frac{8[\tau_1^3\sigma_i+\sigma_i-(1-\tau_1)^3\sigma_{i-1}+\tau_1(1-\tau_1)(1-2\tau_1)\mu_i]}{\tau_1^3(1-\tau_1)^3}\ . \qquad (18)$$

If σ,μ and δ are defined as $\sigma=\frac{\sigma_i}{\sigma_{i-1}}$, $\mu=\frac{\mu_i}{\delta_{i-1}}$ and $\delta=\frac{\tau_{2,1}}{\tau_{1.0}}$, respectively, then (18) reduces to a cubic equation of the form (for σ_i as in (16))

$$\phi' \overset{def}{=} \delta^3-\mu\delta^2+\mu\delta-\sigma=0\ . \qquad (19)$$

that can be solved for δ.

Since M is symmetric-positive-definite, it has nonsingular Cholesky factors ($M = LL^T$, say), so that

$$\mu_i^2=(s_{i-1}^T LL^T s_i)^2=(v_{i-1}^T v_i)^2,$$

say, where

$$v_k \overset{def}{=} L^T s_k.$$

Hence, by the Cauchy-Schwartz inequality,

$$\mu_i^2\le\|v_{i-1}\|_2^2\,\|v_i\|_2^2=\sigma_{i-1}\sigma_i.$$

Thus,

$$\mu^2=\left(\frac{\mu_i}{\sigma_{i-1}}\right)^2\le\sigma.$$

Provided no null steps are taken (that is, $s_j = 0$), σ is finite and positive.

It is observed here that if $\sigma < 1$, $\mu < 0$ and $\delta_* \le \mu + 2[\mu^2 - 3\mu]^{1/2}/3$, then ϕ' has one positive real zero (δ_*) and two negative real zeroes (δ_1 and δ_2 , say) which satisfy the following inequalities (see [2]):

$$-\delta_*\le\delta_1,\delta_2<0.$$

Although $\delta = -1$ gives a global minimum of the function ϕ , it is not admissible as a solution to the “minimum curvature” problem as it corresponds to τ_1 being infinite. It is important to mention at this point that this last equation is independent of the choices made for the τ parameters. The modified Secant Equation we seek to satisfy in every update of the Hessian can be expressed as:

$$[y_i\tau_1(2-\tau_1) - y_{i-1}(1-\tau_1)^2] = B_{i+1}[s_i\tau_1(2-\tau_1) - s_{i-1}(1-\tau_1)^2].$$

2.4. Algorithm C3

This algorithm mimics **C2** with the curvature minimized at $\tau = \tau_2$ this time. The remaining $\tau-$parameters are chosen as $\tau_1 = 1/2$ and $\tau_0 = 0$. Thus, (14) takes the form

$$x'(\tau_2, 0) = (s_i\tau_2^{-1}(4\tau_2 - 1)(2\tau_2 - 1)^{-1} - s_{i-1}\tau_2^{-1}(2\tau_2 - 1)).$$

From (15) it can be shown that choosing $\theta = 0$ yields

$$x''(\tau) \equiv \frac{4}{\tau_2(2\tau_2 - 1)}(s_i - s_{i-1}(2\tau_2 - 1)).$$

Consequently, the curvature expression is given as

$$\begin{aligned}\phi(\tau_2, 0) &= x''(\tau_2, 0)^T M x''(\tau_2, 0) \\ &= \frac{16}{\tau_2^2(2\tau_2 - 1)^2}\left(\sigma_i + (2\tau_2 - 1)^2\sigma_{i-1} - 2(2\tau_2 - 1)\mu_i\right),\end{aligned}$$

where σ_i, σ_{i-1} and μ_i are as in (18).

It follows that

$$\phi'(\tau) = \frac{32[-2\tau_2\sigma_i + \sigma_i - (\tau_2 - 1)^3\sigma_{i-1} + (\tau_2 - 1)(3\tau_2 - 2)\mu_i]}{\tau_2^3(2\tau_2 - 1)^3}. \quad (20)$$

If σ, μ and δ were defined as: $\sigma = \frac{\sigma_i}{\sigma_{i-1}}$, $\mu = \frac{\mu_i}{\delta_{i-1}}$and $\delta = \frac{\tau_{2,1}}{\tau_{1,0}}$, then $\delta = 2\tau_2 - 1$. Thus, (20) becomes

$$\delta^3 - 3\mu\delta^2 + \delta(2\sigma - \mu) + \sigma = 0.$$

A similar analysis of the roots followed for **C2** applies here as well. Accordingly, the modified multi-step Secant equation takes the following form

$$(y_i(\tau_2 - \frac{1}{4}) - y_{i-1}(\tau_2 - \frac{1}{2})^2) = B_{i+1}(s_i(\tau_2 - \frac{1}{4}) - s_{i-1}(\tau_2 - \frac{1}{2})^2).$$

3. Numerical Results

The methods were numerically tested on sixty problems classified into “low” ($2 \leq n \leq 15$), “medium” ($16 \leq n \leq 45$) and “high” ($46 \leq n \leq 80$) dimensions as in [7]. Each function was tested using four different starting-points. For problems with variable dimension, the tests were also done on different dimensions, depending on the degree of freedom permitted by the specific problem. The total number of test problems is an overall of 876 problems. The overall scores for function/gradient evaluations, iteration count and total execution times are reported in Table 1. Tables 2 to 5 show the collective scores for each dimension group. The tabulated results serve as a comparison mechanism for the performance of each method on each of the subsets as well as on the complete set. For each problem, the method yielding the least number of function/gradient evaluations as related to the total function/gradient evaluations is awarded one point, accumulated under the total score count under “Scores” in each table. In our tests, we have employed a cubic interpolation line search method where a new estimate to the minimum is accepted if the following two conditions are satisfied (see [[16]])

$$f(x_{i+1}) \leq f(x_i) + 10^{-4} s_i^T g(x_i)$$

and

$$s_i^T g(x_{i+1}) > 0.9 s_i^T g(x_i).$$

Algorithm **M1** correpsonds to the standard one-step BFGS and **A1** is the best performing multi-step method reported in [7].

A general curvature algorithm has the following outline:
1- Start with any estimate x_0, of the minimum.
2- Start with a symmetric positive-definite matrix H_0 (usually $H_0 = I$).
3. $i = 0$.
4. Find $g_0 = g(x_0)$;
5. Repeat
Step 1. Let $p_i = -H_i g_i$.
Step 2.

$$\text{Minimize}_{t \in R} \ f(x_i + tp_i), \text{ to find } t_i > 0.$$

using some line search technique (e.g., Cubic Interpolation [8]).

Step 3. $x_{i+1} = x_i + t_i p_i$.

compute $s_i = x_{i+1} - x_i$, and $y_i = g_{i+1} - g_i$.

if $i > 1$ then {solve a curvature minimization problem and compute the $\tau-$parameters, namely, τ_0, τ_1 and τ_2; compute r_i and w_i in (4);}

else { $r_i = s_i$; $w_i = y_i$; }

Step 4.

update $H_{i+1}^{BFGS} = H_i + \left[1 + \frac{w_i^T H_i w_i}{r_i^T w_i}\right] \frac{r_i r_i^T}{r_i^T w_i} - \frac{r_i w_i^T H_i + H_i w_i r_i^T}{r_i^T w_i}$

Step 5. $i = i + 1$

until $\|g_i\|_2 < \epsilon$, where $\epsilon \in R$ $(\epsilon > 0)$ is a convergence tolerance.

The results obtained from the tests show clearly that **C3** exhibits a superior numerical performance, by comparison with the other algorithms derived in this work and those published in earlier works [2,5,6,7]. In general, the curvature methods are noted to also present numerical improvement on the low dimension, unlike earlier algorithms for which such improvement was rather marginal on such dimensions.

Table 1 : Overall Results (876 problems)

Method	**Evaluations**	**Iterations**	**Time (sec.)**	**Score**
M1	86401 (100.00%)	73090 (100.00%)	39171.18 (100.00%)	101
A1	76164 (88.15%)	61335 (83.92%)	31474.71 (80.35%)	139
C1	75975(87.93%)	60490 (82.76%)	31247.42 (79.77%)	115
C2	77003 (89.12%)	61404 (84.01%)	31843.15 (81.29%)	98
C3	72020 (83.13%)	56440 (77.21%)	30316.36 (77.39%)	136

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

Method	**Evaluations**	**Iterations**	**Time (sec.)**	**Scores**
M1	25589 (100.00%)	21648 (100.00%)	319.83 (100.00%)	46
A1	24494 (95.72%)	19525 (90.19%)	267.25 (83.56%)	77
C1	24343 (95.13%)	18991 (87.72%)	265.31 (82.95%)	91
C2	25097 (98.08%)	19605 (90.56%)	273.84 (85.62%)	107
C3	22314 (87.21%)	17346 (80.12%)	258.13 (80.7%)	119

Table 3 : Results for dimensions from 16 to 45 (240 problems)

Method	**Evaluations**	**Iterations**	**Time (sec.)**	**Scores**
M1	27058 (100.00%)	23844 (100.00%)	3429.78 (100.00%)	11
A1	22995 (84.98%)	19578 (82.11%)	2745.05 (80.04%)	36
C1	23001 (85.00%)	19450 (81.51%)	2813.11 (82.02%)	51
C2	22903 (84.64%)	19384 (81.30%)	2717.96 (79.25%)	64
C3	21393 (79.06%)	17256 (72.32%)	2533.31(73.86%)	78

Table 4 : Results for dimensions from 46 to 80 (134 problems)

Method	Evaluations	Iterations	Time (sec.)	Scores
M1	21146 (100.00%)	17431 (100.00%)	13426.87 (100.00%)	7
A1	18009 (85.17%)	14122 (81.02%)	10835.33 (80.70%)	14
C1	17930 (84.79%)	13819 (79.27%)	10315.66 (76.82%)	31
C2	18110 (85.64%)	14143 (81.14%)	10891.54 (81.12%)	39
C3	17940 (84.83%)	13920 (79.85%)	10513.71 (78.30%)	43

Table 5 : Results for dimensions from 81 to 100 (62 problems)

Method	Evaluations	Iterations	Time (sec.)	Scores
M1	12608 (100.00%)	10167 (100.00%)	21994.7 (100.00%)	3
A1	10666 (84.60%)	8110 (79.77%)	17627.08 (80.14%)	9
C1	10701 (86.40%)	8230 (81.36%)	17853.34 (81.66%)	11
C2	10893 (86.40%)	8272 (81.36%)	17959.81 (81.66%)	18
C3	10373 (82.27%)	7918 (77.87%)	17011.21 (77.34%)	21

4. Conclusions and Suggestions for Further Work

A general framework for developing parameterizations of the interpolating curves in the two-step quasi-Newton methods has been proposed. The framework relies on the minimation of the curvature of the curve that interpolates the three most recent iterates at a chosen point to ensure the 'smoothness' of the interpolant. The curvature minimaztion problem has been shown in this work to have cheaply computable roots at each iteration. Numerical experiments have proven the viability of the approach presented here where algorithms of this family have yielded substantial numerical gains over the standard BFGS method. The curvature methods have also revealed better performance compared to the original multi-step methods. There remains open the issue of whether there is an optimal choice for the parameters defining the curve, other than those corresponding to minimal curvature. The theoretical convergence properties of such methods are yet to be explored. The merits the approach presented here are yet to be assessed on higher than two multi-step methods.

References

[1] I.A.R. Moghrabi and S.Y.Obeid, *A new family of multi-step quasi-Newton methods*, 6th Int. Coll. on Differential Equations, VSP (1998), 319-326.

[2] J.A. Ford and I.A. Moghrabi, *Minimum curvature quasi-Newton methods*, Computers Math. Applic. 31 (1996),179-186.

[3] J.A. Ford and I.A. Moghrabi, *Using function-values in multi-step quasi-Newton methods*, Journal of Computational and Applied Mathematics 66 (1996), 201-212.

[4] J.A. Ford and I.A. Moghrabi, *A nonlinear model for function-value multistep methods*, Computers & Mathematics with Applications 42 (1996), no. 8-9, 1157-1164.

[5] J.A. Ford and I.A. Moghrabi, *Further investigation of multi-step quasi-Newton methods*, Scientia Iranica 1 (1995), 327-334.

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

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

[8] P. Gill, W. Murray and M. Wright, *Numerical Linear Algebra on Optimization*, volume I, Addison-Wesley, U.S.A., 1991).

[9] R.F. Fletcher, *Pratical Methods Of Optimization*, John Wiley, Great Britain, 1991.

[10] J.A. Ford and A.F. Saadallah, *Efficient utilization of function-values in unconstrained minimization*, Colloq. Math. Soc. Jan. Bolyai 50 (1986), 539-563.

[11] J.E. Dennis and R.B. Schnabel, *Minimum change variable metric update formulae*, SIAM Review 21 (1979), 443-459.

[12] C.G. Broyden, *The convergence of a class of double-rank minimization algorithms*, J. Inst. Math. Applic. 6 (1970), 76-90 and 222-231.

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

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

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

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