Research Article

Diagonal Hessian Approximation for Limited Memory Quasi-Newton via Variational Principle

Siti Mahani Marjugi¹ and Wah June Leong²

¹ Institute for Mathematical Research, University Putra Malaysia, 43400 Serdang, Selangor, Malaysia

² Department of Mathematics, University Putra Malaysia, 43400 Serdang, Selangor, Malaysia

Correspondence should be addressed to Siti Mahani Marjugi; smahani@upm.edu.my

Received 23 July 2013; Revised 21 October 2013; Accepted 19 November 2013

Academic Editor: Martin Weiser

Copyright © 2013 S. M. Marjugi and W. J. Leong. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes some diagonal matrices that approximate the (inverse) Hessian by parts using the variational principle that is analogous to the one employed in constructing quasi-Newton updates. The way we derive our approximations is inspired by the least change secant updating approach, in which we let the diagonal approximation be the sum of two diagonal matrices where the first diagonal matrix carries information of the local Hessian, while the second diagonal matrix is chosen so as to induce positive definiteness of the diagonal approximation at a whole. Some numerical results are also presented to illustrate the effectiveness of our approximating matrices when incorporated within the L-BFGS algorithm.

1. Introduction

Our investigation begins by seeking effective ways to diagonally scale an identity matrix, which is often used to initialize the L-BFGS method. For this purpose, it is useful to state the weak quasi-Newton equation:

$$s_k^T M_{k+1} s_k = s_k^T y_k, \quad (1)$$

where the n -dimensional vector $s_k = x_{k+1} - x_k$ denotes the step corresponding to two different points x_k and x_{k+1} , $y_k = g_{k+1} - g_k$ denotes the gradient change corresponding to the gradients g_k and g_{k+1} at the two points, M_{k+1} is a full $n \times n$ matrix that approximates the Hessian of f . Both s and y are used explicitly and $O(n^2)$ storage is required for the matrix M_{k+1} . If M is further chosen to be a diagonal matrix, say D_{k+1} , then we can establish the so-called quasi-Cauchy relation denwol 1993, [1] by which

$$s_k^T D_{k+1} s_k = s_k^T y_k. \quad (2)$$

Different from the standard quasi-Newton which requires $O(n^2)$ of storage, here only $O(n)$ storage is required to store the diagonal update that satisfies the quasi-Cauchy relation.

In addition, we suppose the matrix D_{k+1} to be positive definite by which it is able to define a metric.

The first example of diagonal updating that satisfies the quasi-Cauchy relation is the well-known Oren-Luenberger OREN1974 scaling matrix, given by

$$D_{k+1} = \frac{s_k^T y_k}{s_k^T s_k} I, \quad (3)$$

where I is the identity matrix. Expression (3) would be obtained from the quasi-Cauchy relation with the further restriction where the diagonal matrix is a scalar multiple of the identity matrix. Therefore, scaling matrices that are derived from the quasi-Cauchy relation are a natural generalization of Oren-Luenberger scaling.

In general, a procedure to obtain diagonal updating formulae via quasi-Cauchy relation can be summarized as follow. Suppose that $D_k > 0$ is a positive definite diagonal matrix and D_{k+1} is the updated version of D , which is also diagonal. Then the updated D_{k+1} is required to satisfy the quasi-Cauchy relation. It is also essential to require the deviation between D_k and D_{k+1} being minimized under some variational principle, which in return will encourage numerical stability. Very often the Frobenius matrix norm

is used to measure this deviation. As noted earlier, a diagonal matrix uses the same computer storage as a vector, resulting in potential use in limited memory algorithms.

This paper is structured as follows. In Section 2, we consider a briefly review on the diagonal updating via quasi-Cauchy relation and the preconditioning strategy. New diagonal initial approximation for L-BFGS method is introduced in this section as well. It follows that in Section 3, the convergence properties of the proposed L-BFGS methods are investigated. Numerical results are presented in Section 4, on a large set of unconstrained minimization problems, mainly from the collections of Moré et al. [2], CUTE [3], and Toint [4]. In Section 5, through the numerical results, we do the discussion regarding the new diagonal initial approximating which matrix is more effective at improving the performance of the L-BFGS update compared to the standard preconditioner in the literature. Finally, Section 6 ends the paper by presenting a summary and conclusion.

2. Diagonal Updating via Quasi-Cauchy Relation

The performance of the L-BFGS method is depending on a good approximations of the actual Hessian. In the basic implementation of the L-BFGS method, the correction pairs are as follows:

$$(s_i, y_i) \quad \text{for } k - \widehat{m} + 1 \leq i \leq k, \quad (4)$$

to correct H_k^0 . The choices of H_k^0 often influences the behaviour of the method. It is worth the investigation on the choices of H_k^0 .

Throughout this section, when we mention a direct initial matrix, we mean a matrix that is a rough approximation to the Hessian; otherwise an initial matrix is an approximation to the inverse of Hessian.

Our approach is inspired by [5, 6] which employed a variational technique that is analogue to the one used to derive the Powell Symmetric Broyden (PSB) quasi-Newton update (see, e.g, Dennis and Schnabel [7]). The resulting update for approximating the Hessian matrix diagonally is derived as follows:

$$D_k = D_{k-1} + \frac{(s_{k-1}^T y_{k-1} - s_{k-1}^T D_{k-1} s_{k-1})}{\text{tr}(E_{k-1}^2)} E_{k-1}, \quad (5)$$

where $E_{k-1} = \text{diag}(s_{k-1,1}^2, s_{k-1,2}^2, \dots, s_{k-1,n}^2)$, $s_{k-1,i}$ denotes the i th component of the vector s_{k-1} , and tr is the trace operator.

Note that when $s_{k-1}^T y_{k-1} < s_{k-1}^T D_{k-1} s_{k-1}$, the resulting D_{k+1} is not necessarily positive definite. Thus, like their counterpart of PSB update in the quasi-Newton setting, the foregoing update may suffer from the loss of positive definiteness and it is not appropriate for use within a quasi-Newton-based algorithm.

In this study, our approach in finding an efficient diagonal Hessian approximation is done through letting the diagonal approximating matrix be a combination of two diagonal matrices. This gives us a freedom to incorporate curvature

information into one of these diagonal matrices, while the property of hereditary positive definiteness is carried over to the second matrix.

To begin, suppose that the Hessian matrix of an objective function has positive diagonal elements. Let us divide the Hessian matrix into two parts:

$$\nabla^2 f(x) = \nabla^2 f_A(x) + \nabla^2 f_B(x), \quad (6)$$

where $\nabla^2 f_A(x)$ is a diagonal matrix consisting the diagonal entries of the Hessian and $\nabla^2 f_B(x)$ would resemble the actual Hessian except that its diagonal entries are all zero. Thus, we intend to form two diagonal approximating matrices to approximate each part of the Hessian, respectively; that is,

$$\nabla^2 f(x) \approx D^0 = \Psi_1 + \Psi_2. \quad (7)$$

Since it is assume that the entries of the actual Hessian are all positive, an excellent choice would be to let a positive definite diagonal matrix, says Ψ_1 , to approximate $\nabla^2 f_A(x)$. Meanwhile, $\nabla^2 f_B(x)$ is expected to be dense and expensive to compute and would be approximated by Ψ_2 . To preserve positive-definiteness, we introduce additional term in the form of θI , that is, a Levenberg-Marquat-like step to maintain positive definite of Ψ_2 in a way that Ψ_2 is expressed as $\Psi_2 = \theta I + \Psi_3$. Thus, the technique for calculating D_k^0 subjected to the weak-quasi-Newton relation is as follows.

Theorem 1. Suppose that $s_{k-1} \neq 0$; then the optimal solution of the following minimization problem:

$$\begin{aligned} \min \quad & \frac{1}{2} \|\Psi_3\|_F^2, \\ \text{s.t.} \quad & s_{k-1}^T (\Psi_1 + (\theta I + \Psi_3)) s_{k-1} = s_{k-1}^T y_{k-1}, \\ & \Psi_3 \text{ is diagonal,} \end{aligned} \quad (8)$$

is given by

$$\Psi_3 = \frac{(s_{k-1}^T y_{k-1} - s_{k-1}^T \Psi_1 s_{k-1} - \theta s_{k-1}^T s_{k-1})}{\text{tr}(E_{k-1}^2)} E_{k-1}. \quad (9)$$

Proof. Since the objective function in (8) is convex and the feasible set is also convex, then (8) has a unique solution. Its Lagrangian function is given by

$$\begin{aligned} L(\Psi_3, \mu) = & \frac{1}{2} \|\Psi_3\|_F^2 + \mu (s_{k-1}^T \Psi_3 s_{k-1} + \theta s_{k-1}^T s_{k-1} \\ & + s_{k-1}^T \Psi_1 s_{k-1} - s_{k-1}^T y_{k-1}), \end{aligned} \quad (10)$$

where μ is the Lagrange multiplier associated with the constraint. Differentiating (10) with respect to each of elements of Ψ_3 and setting the results to zero yields,

$$\Psi_3 = -\mu E_{k-1}. \quad (11)$$

Pre- and postmultiplying (11) by s_{k-1} and invoking the constraint,

$$\begin{aligned} s_{k-1}^T \Psi_3 s_{k-1} &= s_{k-1}^T y_{k-1} - s_{k-1}^T \Psi_1 s_{k-1} - \theta s_{k-1}^T s_{k-1} \\ &= -\mu s_{k-1}^T E_{k-1} s_{k-1}. \end{aligned} \quad (12)$$

We have

$$\mu = \frac{s_{k-1}^T \Psi_1 s_{k-1} + \theta s_{k-1}^T s_{k-1} - s_{k-1}^T y_{k-1}}{s_{k-1}^T E_{k-1} s_{k-1}}. \quad (13)$$

Finally, by substituting (13) into (11) and using the fact that $s_{k-1}^T E_{k-1} s_{k-1} = \text{tr}(E_{k-1}^2)$ yields (9),

$$\begin{aligned} \Psi_3 &= -\mu E_{k-1} \\ &= -\frac{(s_{k-1}^T \Psi_1 s_{k-1} + \theta s_{k-1}^T s_{k-1} - s_{k-1}^T y_{k-1})}{\text{tr}(E_{k-1}^2)} E_{k-1}. \end{aligned} \quad (14)$$

□

A direct result of Theorem 1 leads to the following diagonal preconditioning formulation:

$$D_k^0 = \Psi_1 + \left(\theta I + \frac{s_{k-1}^T y_{k-1} - s_{k-1}^T \Psi_1 s_{k-1} - \theta s_{k-1}^T s_{k-1}}{\text{tr}(E_{k-1}^2)} E_{k-1} \right). \quad (15)$$

An analogue of the above approach can be used to derive the initial inverse Hessian approximation H^0 , which is more useful for algorithmic purpose. Once again by letting the inverse Hessian of an objective be separated by parts;

$$(\nabla^2 f(x))^{-1} = H^{(1)} + H^{(2)}, \quad (16)$$

then the initial approximating matrix H^0 can be expressed as

$$H^0 = \Delta_1 + (\hat{\theta} I + \Delta_3). \quad (17)$$

Since it is our intention to derive an initial approximation for the L-BFGS method, then an excellent choice of $\Delta_{k+1,i}$ for approximating the diagonal entries of the inverse Hessian would be the diagonalized inverse BFGS formula of Gilbert and Lemaréchal [8] updated from a multiple of identity matrix; that is,

$$\Delta_{k+1,i} = \Delta_{k,i} + \left(\frac{1}{y_k^T s_k} + \frac{y_k^T \Delta_{k,i} y_k}{(y_k^T s_k)^2} \right) (s_{k,i})^2 - \frac{2\Delta_{k,i} s_{k,i} y_{k,i}}{y_k^T s_k}, \quad (18)$$

where we choose $\Delta_{k,i} = y_k^T s_k / y_k^T y_k$ as the Oren-Luenberger scalar. Thus, diagonal matrix Δ_1 can be obtained in (18). Note that, to safeguard very small or very large $\|s_{k-1}\|$, we impose the additional following condition: if $\varepsilon_2 \leq \|s_{k-1}\| \leq \varepsilon_1$, for some small and large positive ε_1 and ε_2 , we set $B_0 = I$. Here, we used $\varepsilon_1 = 10^8$ and $\varepsilon_2 = 10^{-8}$.

By interchanging the role of s and y in Theorem 1, one can obtain the formula for H^0 at step k as follows:

$$H_k^0 = \Delta_1 + \left(\theta I + \frac{y_{k-1}^T s_{k-1} - y_{k-1}^T \Delta_1 y_{k-1} - \theta y_{k-1}^T y_{k-1}}{\text{tr}(Y_{k-1}^2)} Y_{k-1} \right), \quad (19)$$

where Δ_1 is given by (18) and $Y_{k-1} = \text{diag}(y_{k-1,1}^2, \dots, y_{k-1,n}^2)$ with $y_{k-1,i}$ is the i th component of the vector y_{k-1} .

For purposes of numerical illustrations, the latter diagonal formula (19) is used to initialize the L-BFGS method, although the potential use of (15) should not be neglected. Furthermore, one can observe that Δ_3 involved in the solution of the variational problem is isolated from the solution and its value does not affect the quality of the solution. This allows us to choose the value of θ freely to ensure that H_k^0 are positive definite while satisfying the weak-secant equation.

Note that maintaining positive definiteness for H_k^0 is crucial for L-BFGS method to generate descent direction. For this purpose, the following lemma suggests a possible choice on θ .

Lemma 2. Assume that $y_k \neq 0$ for all k . Then H_k^0 is a positive definite, if

$$\theta = \min \left(1, \frac{y_{k-1}^T s_{k-1} - y_{k-1}^T \Delta_1 y_{k-1}}{y_{k-1}^T y_{k-1}} \right). \quad (20)$$

Proof. Note that to keep positive definiteness of H_k^0 , we should choose θ such that

$$y_{k-1}^T s_{k-1} - y_{k-1}^T \Delta_1 y_{k-1} - \theta y_{k-1}^T y_{k-1} \geq 0, \quad (21)$$

which also implies

$$\theta \leq \frac{y_{k-1}^T s_{k-1} - y_{k-1}^T \Delta_1 y_{k-1}}{y_{k-1}^T y_{k-1}}. \quad (22)$$

Therefore by our choice on θ as (20), we have the following.

Case 1. If $(y_{k-1}^T s_{k-1} - y_{k-1}^T \Delta_1 y_{k-1}) / y_{k-1}^T y_{k-1} \geq 1$, we will let $\theta = 1$, and thus,

$$H_k^0 = \Delta_1 + I + \frac{y_{k-1}^T s_{k-1} - y_{k-1}^T \Delta_1 y_{k-1} - y_{k-1}^T y_{k-1}}{\text{tr}(Y_{k-1}^2)} Y_{k-1}. \quad (23)$$

Since $y_{k-1}^T s_{k-1} - y_{k-1}^T \Delta_1 y_{k-1} \geq y_{k-1}^T y_{k-1}$, then H_k^0 would be positive definite.

Case 2. If $(y_{k-1}^T s_{k-1} - y_{k-1}^T \Delta_1 y_{k-1}) / y_{k-1}^T y_{k-1} < 1$, $\theta_1 = (y_{k-1}^T s_{k-1} - y_{k-1}^T \Delta_1 y_{k-1}) / y_{k-1}^T y_{k-1}$ is set, which leads to

$$H_k^0 = \Delta_1 + \frac{y_{k-1}^T s_{k-1} - y_{k-1}^T \Delta_1 y_{k-1}}{y_{k-1}^T y_{k-1}} I. \quad (24)$$

It is clear that for both cases, H_k^0 would maintain positive definiteness. □

Hence, H_k^0 can be expressed in the following form:

$$H_k^0 = \begin{cases} \Delta_1 + \frac{y_{k-1}^T s_{k-1} - y_{k-1}^T \Delta_1 y_{k-1}}{y_{k-1}^T y_{k-1}} I, \\ \text{if } \frac{y_{k-1}^T s_{k-1} - y_{k-1}^T \Delta_1 y_{k-1}}{y_{k-1}^T y_{k-1}} < 1, \\ \Delta_1 + I + \frac{y_{k-1}^T s_{k-1} - y_{k-1}^T \Delta_1 y_{k-1} - y_{k-1}^T y_{k-1}}{\text{tr}(Y_{k-1}^2)} Y_{k-1}, \\ \text{if } \frac{y_{k-1}^T s_{k-1} - y_{k-1}^T \Delta_1 y_{k-1}}{y_{k-1}^T y_{k-1}} \geq 1. \end{cases} \quad (25)$$

Now, we can set up the basic algorithm of our L-BFGS methods using H_k^0 as (25).

2.1. LBFGS-USD Algorithm

Step 1. Choose an initial point $x_0 \in \mathbb{R}^n$ and a positive definite matrix $H_k^0 = I$. Let $k = 0$.

Step 2. Compute $d_k = -H_k g_k$; then $x_{k+1} = x_k + \alpha_k d_k$ where α_k satisfies the Wolfe conditions given by

$$\begin{aligned} f(x_k + \alpha_k d_k) &\leq f(x_k) + \delta_1 \alpha_k g_k^T d_k \\ g(x_k + \alpha_k d_k)^T d_k &\geq \delta_2 g_k^T d_k \end{aligned} \quad (26)$$

(we always try the step length $\alpha_k = 1$).

Step 3. Let $\widehat{m} = \min\{k, m-1\}$. Update $H_k^0 \widehat{m} + 1$ times using the pairs $\{y_j, s_j\}_{j=k-\widehat{m}}^k$; that is, let

$$\begin{aligned} H_{k+1} &= (V_k^T \cdots V_{k-\widehat{m}}^T) H_k^0 (V_{k-\widehat{m}} \cdots V_k) \\ &+ \rho_{k-\widehat{m}} (V_k^T \cdots V_{k-\widehat{m}+1}^T) s_{k-\widehat{m}} s_{k-\widehat{m}}^T (V_{k-\widehat{m}+1} \cdots V_k) \\ &+ \rho_{k-\widehat{m}+1} (V_k^T \cdots V_{k-\widehat{m}+2}^T) \\ &\times s_{k-\widehat{m}+1} s_{k-\widehat{m}+1}^T (V_{k-\widehat{m}+2} \cdots V_k) \\ &\vdots \\ &+ \rho_k s_k s_k^T. \end{aligned} \quad (27)$$

Step 4. Set $k := k + 1$ and return to Step 2.

Remark 3. The LBFGS-USD algorithm is exactly the L-BFGS algorithm of Liu and Nocedal [9], except that H_k^0 is computed by (25).

3. Convergence Analysis

We shall also establish the convergence of the LBFGS-USD algorithm. The following standard assumptions are made on the objective function.

- (a) The objective function f is twice continuously differentiable.
- (b) The level set $\mathcal{E} = \{x \in \mathbb{R}^n \mid f(x) < f(x_0)\}$ is convex, and there exist positive constants m_1 and m_2 such that,

$$m_1 \|z\|^2 \leq z^T G(x) z \leq m_2 \|z\|^2, \quad (28)$$

for all $z \in \mathbb{R}^n$ and $x \in \mathcal{E}$.

Theorem 4. Let x_0 be a starting point for which f satisfies assumptions above, and assume that the matrices B_k^0 are chosen so that $\{\|B_k^0\|\}(\|(H_k^0)^{-1}\|)$ are bounded. Subsequently, for any positive definite B^0 (most often, $B^0 = I$ is chosen), LBFGS-USD algorithm generates a sequence $\{x_k\}$ which converges to x^* . Moreover, there is a constant $0 \leq r \leq 1$ such that

$$f(x_k) - f(x^*) \leq r^k [f(x_0) - f(x^*)], \quad (29)$$

which implies that $\{x_k\}$ converges $\mathbb{R}$ -linearly.

Proof. See [9]. $\square$

Theorem 4 suggests that as long as H_k^0 is chosen such that $\|H_k^0\|$ is bounded for all k , then the corresponding L-BFGS algorithm restarts by H_k^0 would generate $\{x_k\}$ that converges globally and $\mathbb{R}$ -linearly. For this purpose, we give the following result which ensures that $\|H_k^0\|$ where H_k^0 is given by (25) is upper and lower bounded by some constants.

Lemma 5. Let x_0 be a starting point for which f satisfies assumptions (a)-(b). Consider the sequence $\{x_k\}$ generated by L-BFGS algorithms subject to the diagonal initial approximation, H_k^0 given by (25). If $y_k \neq 0$ for all k , then $\|H_k^0\|$ is upper and lower bounded for all k .

Proof. Let $\overline{G}_{k-1} = \int_0^1 G(x_{k-1} + ts_{k-1}) dt$, then $y_{k-1} = \overline{G}_{k-1} s_{k-1}$, and assumption (a) implies that

$$m_1 \leq \frac{s_{k-1}^T \overline{G}_{k-1} s_{k-1}}{\|s_{k-1}\|^2} \leq m_2. \quad (30)$$

Subsequently, (28) also gives

$$\frac{s_{k-1}^T y_{k-1}}{\|y_{k-1}\|^2} = \frac{s_{k-1}^T \overline{G}_{k-1} s_{k-1}}{s_{k-1}^T \overline{G}_{k-1}^2 s_{k-1}} \leq \frac{m_2 \|s_{k-1}\|^2}{m_1^2 \|s_{k-1}\|^2}, \quad (31)$$

and therefore,

$$s_{k-1}^T y_{k-1} \leq \frac{m_2}{m_1^2} \|y_{k-1}\|^2. \quad (32)$$

First, we begin by showing that every component of Δ_1 in (18) is bounded, so that $\|\Delta_1\|$ is bounded. It is more convenient to show that each part of Δ_1 , namely,

- (i) $\sigma_{k-1} = y_{k-1}^T s_{k-1} / y_{k-1}^T y_{k-1}$,
- (ii) $(1/y_{k-1}^T s_{k-1} + y_{k-1}^T \sigma_{k-1} y_{k-1} / (y_{k-1}^T s_{k-1})^2) (s_{k-1}^T e_i)^2$, and
- (iii) $-(2\sigma_{k-1} (y_{k-1}^T e_i) (s_{k-1}^T e_i)) / y_{k-1}^T s_{k-1}$,

is bounded, such that Δ_1 is bounded at whole.

(i) Since

$$\frac{1}{m_2} \leq \frac{y_{k-1}^T (\bar{G}_{k-1})^{-1} y_{k-1}}{y_{k-1}^T y_{k-1}} \leq \frac{1}{m_1}, \quad (33)$$

then, we obtain

$$\frac{1}{m_2} \leq \sigma_{k-1} \leq \frac{1}{m_1}. \quad (34)$$

(ii) Note that

$$(s_{k-1}^T e_i)^2 = s_{k-1,i}^2, \quad (35)$$

where $s_{k-1,i}$ is the i th component of s_{k-1} . Then,

$$\begin{aligned} & \left(\frac{1}{y_{k-1}^T s_{k-1}} + \frac{y_{k-1}^T \sigma_{k-1} y_{k-1}}{(y_{k-1}^T s_{k-1})^2} \right) (s_{k-1}^T e_i)^2 \\ & \geq \frac{\sigma_{k-1} y_{k-1}^T y_{k-1}}{(y_{k-1}^T s_{k-1})^2} (s_{k-1}^T e_i)^2 \\ & \geq \frac{\sigma_{k-1} y_{k-1}^T y_{k-1}}{\|y_{k-1}\|^2 \|s_{k-1}\|^2} (s_{k-1}^T e_i)^2 \\ & \geq \frac{\sigma_{k-1} \varepsilon_2^2}{\|s_{k-1}\|^2} \\ & \geq \frac{\sigma_{k-1} \varepsilon_2^2}{\varepsilon_1^2} \\ & \geq \frac{1}{m_2} \frac{\varepsilon_2^2}{\varepsilon_1^2}. \end{aligned} \quad (36)$$

On the other hand,

$$\begin{aligned} & \left(\frac{1}{y_{k-1}^T s_{k-1}} + \frac{y_{k-1}^T \sigma_{k-1} y_{k-1}}{(y_{k-1}^T s_{k-1})^2} \right) (s_{k-1}^T e_i)^2 \\ & \leq \frac{\sum_{i=1}^n (s_{k-1,i})^2}{y_{k-1}^T s_{k-1}} + \frac{\|y_{k-1}\|^2 \|s_{k-1}\|^2}{m_1 (y_{k-1}^T (\bar{G}_{k-1})^{-1} y_{k-1})^2} \\ & \leq \frac{\|s_{k-1}\|^2}{s_{k-1}^T \bar{G}_{k-1} s_{k-1}} + \frac{1}{m_1} \frac{m_2^2 \|s_{k-1}\|^2}{\|y_{k-1}\|^2} \end{aligned}$$

$$\begin{aligned} & \leq \frac{1}{m_1} + \frac{m_2^2 y_{k-1}^T \bar{G}_{k-1}^{-2} y_{k-1}}{m_1 y_{k-1}^T y_{k-1}} \\ & \leq \frac{1}{m_1} + \frac{m_2^2}{m_1 m_1^2} = \frac{1}{m_1} + \frac{m_2^2}{m_1^3}. \end{aligned}$$

(37)

Therefore, it gives, in overall,

$$\begin{aligned} \frac{1}{m_2} \frac{\varepsilon_2^2}{\varepsilon_1^2} & \leq \left(\frac{1}{y_{k-1}^T s_{k-1}} + \frac{y_{k-1}^T \sigma_{k-1} y_{k-1}}{(y_{k-1}^T s_{k-1})^2} \right) (s_{k-1}^T e_i)^2 \\ & \leq \frac{1}{m_1} + \frac{m_2^2}{m_1^3}. \end{aligned} \quad (38)$$

(iii) In a similar way, we can establish

$$\begin{aligned} -\frac{2\sigma_{k-1} (y_{k-1}^T e_i) (s_{k-1}^T e_i)}{y_{k-1}^T s_{k-1}} & \geq -\frac{2}{m_1} \frac{y_{k-1}^T s_{k-1}}{y_{k-1}^T s_{k-1}} \\ & \geq -\frac{2}{m_1}. \end{aligned} \quad (39)$$

and subsequently,

$$\begin{aligned} -\frac{2\sigma_{k-1} (y_{k-1}^T e_i) (s_{k-1}^T e_i)}{y_{k-1}^T s_{k-1}} & \leq \frac{2\sigma_{k-1} (y_{k-1}^T e_i) (s_{k-1}^T e_i)}{y_{k-1}^T s_{k-1}} \\ & \leq \frac{2(1/m_1) \sum_{i=1}^n y_{k-1,i} s_{k-1,i}}{y_{k-1}^T s_{k-1}} = \frac{2}{m_1}. \end{aligned} \quad (40)$$

Then, we have

$$-\frac{2}{m_1} \leq -\frac{2\sigma_{k-1} (y_{k-1}^T e_i) (s_{k-1}^T e_i)}{y_{k-1}^T s_{k-1}} \leq \frac{2}{m_1}. \quad (41)$$

Following that, the component of Δ_1 , namely $\Delta_{1,i}$, is bounded as follow:

$$\frac{1}{m_2} + \frac{1}{m_2} \frac{\varepsilon_2^2}{\varepsilon_1^2} - \frac{2}{m_1} \leq \Delta_{1,i} \leq \frac{4}{m_1} + \frac{m_2^2}{m_1^3}, \quad (42)$$

Note that, by the definition of H_k^0 , we have

$$\begin{aligned} H_{k,i}^0 & = \Delta_{1,i} + \frac{y_{k-1}^T s_{k-1} - y_{k-1}^T \Delta_1 y_{k-1}}{y_{k-1}^T y_{k-1}} \\ & \text{if } \frac{y_{k-1}^T s_{k-1} - y_{k-1}^T \Delta_1 y_{k-1}}{y_{k-1}^T y_{k-1}} < 1. \end{aligned} \quad (43)$$

Hence, $H_{k,i}^0$ satisfies

$$\frac{1}{m_2} + \frac{1}{m_2} \frac{\varepsilon_2^2}{\varepsilon_1^2} - \frac{2}{m_1} \leq \Delta_{1,i} \leq H_{k,i}^0 \leq \Delta_{1,i} + 1 \leq \frac{4}{m_1} + \frac{m_2^2}{m_1^3} + 1. \quad (44)$$

TABLE 1: Selected test problems.

TRIGONOMETRIC, SROSENBR, Extended White and Holst,
Perturbed Quadratic, Raydan 1, Raydan 2, Diagonal 1, Diagonal 2, Diagonal 3, Hager,
Generalized Tridiagonal 1, Extended Tridiagonal 1,
Extended Three Expo Terms, Generalized Tridiagonal 2,
Diagonal 5, HIMMELBC, Generalized PSC1, Extended PSC1,
Extended Block-Diagonal BD1, Extended Maratos, Extended Hiebert,
Quadratic QF1, Extended Quadratic Penalty QP1, Quadratic QF2,
Extended EP1, Extended Tridiagonal 2, NONDIA, DIXMAANA,
DIXMAANB, DIXMAANC, DIXMAANE, Broyden Tridiagonal,
Almost Perturbed Quadratic, EDENSCH, STAIRCASE S1,
Diagonal 6, DIXON3DQ, DIXMAANE, DIXMAANG, DIXMAANI,
DIXMAANJ, DIXMAANK, DIXMAANL, ENGVAl, COSINE, DENSCHNB,
DENSCHNF, BIGGSB1, Scaled Quadratic SQ1, Scaled Quadratic SQ2.

Whereas, when $(y_{k-1}^T s_{k-1} - y_{k-1}^T \Delta_1 y_{k-1})/y_{k-1}^T y_{k-1} \geq 1$, we have

$$\begin{aligned}
& \frac{1}{m_2} + \frac{1}{m_2} \frac{\varepsilon_2^2}{\varepsilon_1^2} - \frac{2}{m_1} \\
& \leq \Delta_{1,i} \leq H_{k,i}^0 \\
& \leq \Delta_{1,i} + 1 + \frac{y_{k-1}^T s_{k-1}}{\text{tr } Y_{k-1}^2} y_{k-1,M}^2 \\
& \leq \frac{4}{m_1} + \frac{m_2^2}{m_1^3} + 1 + \frac{m_2 \|y_{k-1}\|^2}{m_1^2 \sum_{i=1}^n (y_{k-1,i}^4) y_{k-1,M}^2} y_{k-1,M}^4 \quad (45) \\
& \leq \frac{4}{m_1} + \frac{m_2^2}{m_1^3} + 1 + \frac{nm_2 y_{k-1,M}^2}{m_1^2 y_{k-1,M}^2} \\
& = 1 + \frac{4}{m_1} + \frac{m_2^2}{m_1^3} + \frac{nm_2}{m_1^2}, \quad \forall i,
\end{aligned}$$

where $y_{k-1,M}^2$ is the largest component among all $y_{k-1,i}^2$. Therefore, we can conclude that $\|(H_{k,i}^0)\|$ is upper and lower bounded for all k . $\square$

4. Numerical Experiences

Our test used a large set of unconstrained minimization problem consisting of 50 test problems where the list of problems is given in Table 1. These test problems are selected from Moré et al. [2], CUTE [3], Toint [4], and various other test function collections such as in [10]. The subroutine of test problems is available at <http://camo.ici.ro/forum/SCALCG/evalfg.for> (accessed on Jan 2012). The method tested is as follows:

- (1) LBFGS-I: L-BFGS method with the initial matrix, $H_k^0 = I$,
- (2) LBFGS-mI: L-BFGS method with the initial matrix, $H_k^0 = \theta I$ where θ_k is the Oren-Luenberger scaling at k ,

- (3) LBFGS-USD: L-BFGS method with the initial matrix, H_k^0 is given by (19).

5. Discussion

In general, Figure 1 indicates that the new diagonal initial approximating matrix are substantially better, followed by both standard initializations of the L-BFGS method in terms of number of iterations, function/gradient calls, and CPU time, respectively. To better study the effect of our initial approximation, we include Tables 2, 3, and 4 that give ratio of function/gradient calls over iteration counts for the L-BFGS method with standard initial matrices and our diagonal initial approximating matrix. As conclusion, our diagonal initial approximation, LBFGS-USD, performs better in which the ratio is close to one and hence would likely to accept unit step length compared to the LBFGS-I (without any preconditioning) method.

Moreover for LBFGS-USD method, it requires, in general less iteration counts as much as 38% and 19% than LBFGS-I and LBFGS-mI, respectively. Meanwhile, in terms of number of function/gradient, LBFGS-MTD needs 46% and 16% less function/gradient counts, respectively. Finally, LBFGS-MTD requires 33% and 6% less CPU time in second, respectively. In conclusion, the numerical results for a broad class of the test problems show that the LBFGS-USD algorithm is efficient and vast superior in solving small to large size problems.

6. Concluding Remarks

We proposed technique that exploits the presence of the Hessian in the diagonal matrix form. Under some standard assumption on the objective function, we observe that the convergence of the diagonal initial approximation of the L-BFGS scheme is $\mathbb{R}$ -linear. Based on our numerical results, we believe that the following conclusions can be made on the diagonal approximation that is derived in this study.

FIGURE 1: Performance profile of LBFGS-USD based on iterations, function/gradient call, and CPU time for all choices of m and n .

TABLE 2: Ratio of function/gradient calls over iteration counts for $m = 3$.

LBFGS-I	LBFGS- mI	LBFGS-USD
1.822477996	1.078519988	1.183809586

TABLE 3: Ratio of function/gradient calls over iteration counts for $m = 5$.

LBFGS-I	LBFGS- mI	LBFGS-USD
1.764504202	1.057316205	1.181762512

TABLE 4: Ratio of function/gradient calls over iteration counts for $m = 7$.

LBFGS-I	LBFGS- mI	LBFGS-USD
2.000371968	1.051121217	1.194642288

- (i) Our diagonal approximating matrix is able to maintain positive definiteness in a very simple way and give $O(n)$ storage requirement.
- (ii) The numerical experiments show that our diagonal initial approximating matrix is generally effective for the L-BFGS method compared to the standard initial matrices in the literature.

Conflict of Interests

The author(s) declare(s) that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was supported by the Malaysian MOHE-FRGS Grant no. 01-11-09-722FR.

References

- [1] F. Modarres, M. A. Hassan, and W. J. Leong, "Improved Hessian approximation with modified secant equations for symmetric rank-one method," *Journal of Computational and Applied Mathematics*, vol. 235, no. 8, pp. 2423–2431, 2011.
- [2] J. J. Moré, B. S. Garbow, and K. E. Hillstom, "Testing unconstrained optimization software," *ACM Transactions on Mathematical Software*, vol. 7, no. 1, pp. 17–41, 1981.
- [3] I. Bongartz, A. R. Conn, N. Gould, and Ph. L. Toint, "CUTE: constrained and unconstrained testing environment," *ACM Transactions on Mathematical Software*, vol. 21, no. 1, pp. 123–160, 1995.
- [4] Ph. L. Toint, "Test problems for partially separable optimization and results for the routine PSPMIN," Tech. Rep. 83/4, Department of Mathematics, University of Namur, Brussels, Belgium, 1983.
- [5] M. Zhu, J. L. Nazareth, and H. Wolkowicz, "The Quasi-Cauchy relation and diagonal updating," *SIAM Journal on Optimization*, vol. 9, no. 4, pp. 1192–1204, 1997.
- [6] M. Farid and W. J. Leong, "An improved multi-step gradient-type method for large scale optimization," *Computers & Mathematics with Applications*, vol. 61, no. 11, pp. 3312–3318, 2011.
- [7] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, New York, NY, USA, 1983.
- [8] J. C. Gilbert and C. Lemaréchal, "Some numerical experiments with variable-storage quasi-Newton algorithms," *Mathematical Programming*, vol. 45, no. 1–3, pp. 407–435, 1989.
- [9] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, no. 1–3, pp. 503–528, 1989.
- [10] N. Andrei, "An unconstrained optimization test functions collection," *Advanced Modeling and Optimization*, vol. 10, no. 1, pp. 147–161, 2008.