

Low complexity minimization algorithms[†]

Carmino Di Fiore^{1,*}, Stefano Fanelli¹ and Paolo Zellini¹

¹ Department of Mathematics, University of Rome “Tor Vergata”, Via della Ricerca Scientifica, 00133 Roma, Italy

SUMMARY

Structured matrix algebras \mathcal{L} and a generalized *BFGS*-type iterative scheme have been recently investigated to introduce low complexity quasi-Newton methods, named $\mathcal{L}QN$, for solving general (nonstructured) minimization problems. In this paper we introduce the $\mathcal{L}^{(k)}QN$ methods, which exploit ad hoc algebras for each step. Since the structure of the updated matrices can be modified at each iteration, the new methods can better fit the Hessian matrix, thereby improving the rate of convergence of the algorithm. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: Unconstrained minimization; quasi-Newton methods; matrix algebras;

1. INTRODUCTION

In this paper we study a new class of quasi-Newton algorithms for the minimization of a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, which are a generalization of some previous methods introduced in [10]. The innovative algorithms, named $\mathcal{L}^{(k)}QN$, exploit, in the quasi-Newton iterative scheme $\mathbf{x}_{k+1} = \mathbf{x}_k - \lambda_k B_k^{-1} \nabla f(\mathbf{x}_k)$, positive definite (p.d.) Hessian approximations of the type

$$B_{k+1} = \varphi(A_k, \mathbf{s}_k, \mathbf{y}_k), \quad A_k \in \mathcal{L}^{(k)}, \quad \mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k, \quad \mathbf{y}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k) \quad (1.1)$$

where the $n \times n$ matrix A_k is picked up in a structured matrix algebra $\mathcal{L}^{(k)}$, shares some significant property with B_k and is p.d.. In (1.1), $\varphi(\cdot, \mathbf{s}_k, \mathbf{y}_k)$ denotes a function updating p.d. matrices into p.d. matrices, whenever $\mathbf{s}_k^T \mathbf{y}_k > 0$, i.e.

$$\left. \begin{array}{l} A \text{ p.d.} \\ \mathbf{s}_k^T \mathbf{y}_k > 0 \end{array} \right\} \Rightarrow \varphi(A, \mathbf{s}_k, \mathbf{y}_k) \text{ p.d.} \quad (1.2)$$

The condition (1.2) is in particular satisfied by the *BFGS* updating formula (2.3) in the next section, whereas a suitable choice of the step length λ_k assures the inequality $\mathbf{s}_k^T \mathbf{y}_k > 0$ (see [8]).

*Correspondence to: Department of Mathematics, University of Rome “Tor Vergata”, Via della Ricerca Scientifica, 00133 Roma, Italy

†

Contract/grant sponsor: 00000000; contract/grant number: 0–0

Hessian approximations of type (1.1) were studied in the case $\mathcal{L}^{(k)} = \mathcal{L} \forall k$, where \mathcal{L} is a fixed space, and the matrix A_k is the best approximation in the Frobenius norm of B_k in \mathcal{L} [10]. Such matrix A_k , denoted by \mathcal{L}_{B_k} , inherits positive definiteness from B_k . So, by the property (1.2), $B_{k+1} = \varphi(\mathcal{L}_{B_k}, \mathbf{s}_k, \mathbf{y}_k)$ is a p.d. matrix (provided that $\mathbf{s}_k^T \mathbf{y}_k > 0$). As a consequence, the $\mathcal{L}QN$ methods of [10] yield a descent direction \mathbf{d}_{k+1} .

If \mathcal{L} is defined as the set of all matrices simultaneously diagonalized by a fast discrete transform U ($\mathcal{L} = sdU$), then the time and space complexity of $\mathcal{L}QN$ is $O(n \log n)$ and $O(n)$, respectively [10], [5]. The latter result makes $\mathcal{L}QN$ methods suitable for minimizing functions f where n is large. In fact, numerical experiences show the competitiveness of $\mathcal{L}QN$ with Limited-memory $BFGS$, which is an efficient method for solving large scale problems [5]. Moreover, a global linear convergence result for the class of $\mathcal{NS} \mathcal{L}QN$ methods is obtained in [10], [11], by extending the analogous $BFGS$ convergence result of Powell [15] with a proper use of some crucial properties of the matrix \mathcal{L}_{B_k} .

The local convergence properties of $\mathcal{L}QN$ were studied in [9]. It is proved, in particular, that $\mathcal{L}QN$ converges to a minimum point \mathbf{x}_* of f with a superlinear rate of convergence whenever $\nabla^2 f(\mathbf{x}_*) \in \mathcal{L}$. The latter result suggests that, in order to improve $\mathcal{L}QN$ efficiency, one might modify the algebra \mathcal{L} in each iteration k , i.e. introduce the $\mathcal{L}^{(k)}QN$ methods. This requires a concept of “closeness” of a space $\mathcal{L}^{(k)}$ with respect to a matrix B_k and the construction, at each iteration, of a space $\mathcal{L}^{(k)}$ as “close” as possible to B_k . Two important properties of B_k are that B_k is p.d. and $B_k \mathbf{s}_{k-1} = \mathbf{y}_{k-1}$. So, we can say that a structured matrix algebra $\mathcal{L}^{(k)}$ is “close” to B_k if $\mathcal{L}^{(k)}$ includes matrices satisfying the latter properties, i.e. if

- the set $\{X \in \mathcal{L}^{(k)} : X \text{ is p.d. and } X \mathbf{s}_{k-1} = \mathbf{y}_{k-1}\}$ is not empty.

Once such space $\mathcal{L}^{(k)}$ is introduced, we can conceive at least two $\mathcal{L}^{(k)}QN$ algorithms, based on the updating formula (1.1):

Algorithm 1: (1.1) with $A_k = \mathcal{L}_{\mathbf{s}\mathbf{y}}^{(k)}$, where $\mathcal{L}_{\mathbf{s}\mathbf{y}}^{(k)} \in \mathcal{L}^{(k)}$ is p.d. and solves the previous secant equation $X \mathbf{s}_{k-1} = \mathbf{y}_{k-1}$.

Algorithm 2: (1.1) with $A_k = \mathcal{L}_{B_k}^{(k)}$, where $\mathcal{L}_{B_k}^{(k)}$ is the best least squares fit to B_k in $\mathcal{L}^{(k)}$.

The present paper is organized as follows. In Section 2 we recall some basic notions on quasi-Newton methods in unconstrained minimization and, in particular, the $BFGS$ algorithm (Broyden et al. '70) [8], [14]. In Section 3 we describe the basic properties of the $\mathcal{L}QN$ methods, recently introduced in [10]. The latter methods turn out to be more efficient than $BFGS$ and extremely competitive with L - $BFGS$ for solving large scale minimization problems [5]. In order to improve the $\mathcal{L}QN$ efficiency, in Section 4 we introduce the innovative $\mathcal{L}^{(k)}QN$ algorithms. Assuming that $\mathcal{L}^{(k)}$ is the set of all matrices diagonalized by a unitary matrix U_k , we translate the previous requirement • to a condition on U_k (see (4.5)); then, we define in detail the $\mathcal{L}^{(k)}QN$ Algorithm 1. In Section 5 we prove that matrix algebras $\mathcal{L}^{(k)}$ satisfying • exist without special conditions on the algorithm. Moreover, we illustrate how to compute such $\mathcal{L}^{(k)}$, by expressing the corresponding matrix U_k as a product of two Householder matrices. In Section 6 we introduce the $\mathcal{L}^{(k)}QN$ Algorithm 2, and we prove that the \mathcal{NS} version of such algorithm is convergent. Moreover, we discuss some possible improvements of Algorithms 1 and 2. Finally, some preliminary numerical experiences illustrated in Section 7 show that $\mathcal{L}^{(k)}QN$ performances are particularly encouraging.

2. QUASI-NEWTON METHODS FOR THE UNCONSTRAINED MINIMIZATION

We have to minimize a function f , i.e. solve the problem:

$$f(\mathbf{x}_*) = \min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}), \quad \text{find } \mathbf{x}_*. \tag{2.1}$$

Let us apply a quasi-Newton (QN) method to the gradient vector function ∇f . Given $\mathbf{x}_0 \in \mathbb{R}^n$, $B_0 = n \times n$ positive definite (p.d.), a QN method generates a sequence $\{\mathbf{x}_k\}_{k=0}^\infty$ convergent to a zero of ∇f , by exploiting a QN iterative scheme, i.e. a Newton scheme where the Hessian $\nabla^2 f(\mathbf{x}_k)$ is replaced by a suitable approximation B_k :

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k, \quad \mathbf{d}_k = -B_k^{-1} \nabla f(\mathbf{x}_k) \quad (\lambda_k \in \mathbb{R}^+). \tag{2.2}$$

The matrix B_k is chosen positive definite (p.d.) so that the search direction \mathbf{d}_k is always a descent direction ($\nabla f(\mathbf{x}_k)^T \mathbf{d}_k < 0$). How to choose the next Hessian approximation B_{k+1} ? In *Secant methods* B_{k+1} satisfies the secant equation, i.e. maps the vector \mathbf{s}_k into the vector \mathbf{y}_k , where $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ and $\mathbf{y}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)$:

$$B_{k+1} \mathbf{s}_k = \mathbf{y}_k \tag{Secant equation}$$

Notice that for $n = 1$ the matrix B_{k+1} is a scalar and is uniquely defined as the difference quotient of the derivative function $f'(x)$, i.e. we retrieve the ordinary secant method applied to f' . In the general case ($n > 1$), the secant equation has many possible solutions and, as a consequence, several secant algorithms can be defined.

In *BFGS* (Broyden et al. '70) [8], [7], [14], the matrix B_{k+1} is defined as a rank-2 perturbation of the previous Hessian approximation B_k :

$$B_{k+1} = \varphi(B_k, \mathbf{s}_k, \mathbf{y}_k) \tag{BFGS}$$

where

$$\varphi(B, \mathbf{s}, \mathbf{y}) = B + \frac{1}{\mathbf{y}^T \mathbf{s}} \mathbf{y} \mathbf{y}^T - \frac{1}{\mathbf{s}^T B \mathbf{s}} B \mathbf{s} \mathbf{s}^T B. \tag{2.3}$$

By the structure of φ , 1) *BFGS* is a secant method 2) *BFGS* has the following property: if B_k is p.d. then B_{k+1} is p.d. provided that the inner product between \mathbf{y}_k and \mathbf{s}_k is positive. Thus:

$$\left. \begin{array}{l} B_k \text{ p.d.} \\ \mathbf{s}_k^T \mathbf{y}_k > 0 \end{array} \right\} \Rightarrow B_{k+1} = \varphi(B_k, \mathbf{s}_k, \mathbf{y}_k) \text{ p.d.} \tag{2.4}$$

The condition $\mathbf{s}_k^T \mathbf{y}_k > 0$ can be assured by a suitable choice of the step length λ_k [8]. In particular, it is satisfied if λ_k is chosen in the Armijo-Goldstein set

$$AG_k = \{ \lambda \in \mathbb{R}^+ : f(\mathbf{x}_k + \lambda \mathbf{d}_k) \leq f(\mathbf{x}_k) + c_1 \lambda \mathbf{d}_k^T \nabla f(\mathbf{x}_k) \ \& \\ \mathbf{d}_k^T \nabla f(\mathbf{x}_k + \lambda \mathbf{d}_k) \geq c_2 \mathbf{d}_k^T \nabla f(\mathbf{x}_k) \}, \\ 0 < c_1 < c_2 < 1.$$

The *BFGS* method has a local superlinear rate of convergence and an $O(n^2)$ time and space complexity. As a consequence, in unconstrained minimization *BFGS* is often more efficient than the modified Newton algorithm. However, the implementation of *BFGS* becomes prohibitive when in problem (2.1) the number n of variables is large. Such large scale problems arise, for example, in the learning process of neural networks [5], [16].

3. $\mathcal{L}QN$ METHODS

The aim of $\mathcal{L}QN$ methods [10] is to reduce the complexity of $BFGS$ by maintaining as more as possible a quasi-Newton behaviour. Several attempts were performed towards this direction (see f.i. [2], [12], [13]). The main idea in [10] is to replace B_k with a simpler matrix chosen in an algebra \mathcal{L} . Let U be a $n \times n$ unitary matrix and define \mathcal{L} as the set of all matrices diagonalized by U ($\mathcal{L} = sdU$):

$$\mathcal{L} = sdU := \{Ud(\mathbf{z})U^* : \mathbf{z} \in \mathbb{C}^n\}, \quad d(\mathbf{z}) = \begin{bmatrix} z_1 & & O \\ & \ddots & \\ O & & z_n \end{bmatrix}. \quad (3.1)$$

Pick up in \mathcal{L} the best approximation of B_k in the Frobenius norm. Call this matrix the *best least squares fit* to B_k in \mathcal{L} and denote it by \mathcal{L}_{B_k} . Then apply the updating function φ to \mathcal{L}_{B_k} :

$$B_{k+1} = \varphi(\mathcal{L}_{B_k}, \mathbf{s}_k, \mathbf{y}_k). \quad (\mathcal{L}QN)$$

If B_k is p.d. then \mathcal{L}_{B_k} is p.d.. This fact is a simple consequence of the following expression of the eigenvalues of \mathcal{L}_{B_k} :

$$\mathcal{L}_{B_k} = Ud(\mathbf{z}_k)U^*, \quad (\mathbf{z}_k)_i = [U\mathbf{e}_i]^* B_k [U\mathbf{e}_i]. \quad (3.2)$$

Thanks to this property, we have also for $\mathcal{L}QN$ methods that B_{k+1} inherits p.d. from B_k whenever $\mathbf{s}_k^T \mathbf{y}_k > 0$; moreover, under the same condition $\mathbf{s}_k^T \mathbf{y}_k > 0$, $\mathcal{L}_{B_{k+1}}$ inherits p.d. from \mathcal{L}_{B_k} :

$$B_k \text{ p.d.} \Rightarrow \left. \begin{array}{l} \mathcal{L}_{B_k} \text{ p.d.} \\ \mathbf{s}_k^T \mathbf{y}_k > 0 \end{array} \right\} \Rightarrow B_{k+1} \text{ p.d.} \Rightarrow \mathcal{L}_{B_{k+1}} \text{ p.d.} \quad (3.3)$$

Thus we have two possible descent directions.

1. The first one in terms of B_{k+1} , leading to a Secant method:

$$\mathbf{d}_{k+1} = -B_{k+1}^{-1} \nabla f(\mathbf{x}_{k+1}) \quad (S \mathcal{L}QN)$$

($B_{k+1} \mathbf{s}_k = \mathbf{y}_k$ since $\varphi(A, \mathbf{s}_k, \mathbf{y}_k) \mathbf{s}_k = \mathbf{y}_k, \forall A$).

2. The second one in terms of $\mathcal{L}_{B_{k+1}}$, leading to a Non Secant method:

$$\mathbf{d}_{k+1} = -\mathcal{L}_{B_{k+1}}^{-1} \nabla f(\mathbf{x}_{k+1}) \quad (\mathcal{NS} \mathcal{L}QN)$$

($\mathcal{L}_{B_{k+1}}$ does not map \mathbf{s}_k into \mathbf{y}_k , in general).

In [10], [11] it is proved that $\mathcal{NS} \mathcal{L}QN$ has a linear rate of convergence, whereas numerical experiences in [5] show that $S \mathcal{L}QN$ has a faster convergence rate.

Moreover, each step of any $\mathcal{L}QN$ method can be implemented so that the most expensive operations are two U transforms and some vector inner products. This fact can be easily proved by examining the identity

$$\mathbf{z}_{k+1} = \mathbf{z}_k + \frac{1}{\mathbf{s}_k^T \mathbf{y}_k} |U^* \mathbf{y}_k|^2 - \frac{1}{\mathbf{z}_k^T |U^* \mathbf{s}_k|^2} d(\mathbf{z}_k)^2 |U^* \mathbf{s}_k|^2 \quad (3.4)$$

and the Sherman-Morrison-Woodbury inversion formula (see [10], [5] for details). Thus, if U defines a fast discrete transform (\mathcal{L} structured), then $\mathcal{L}QN$ can be implemented with

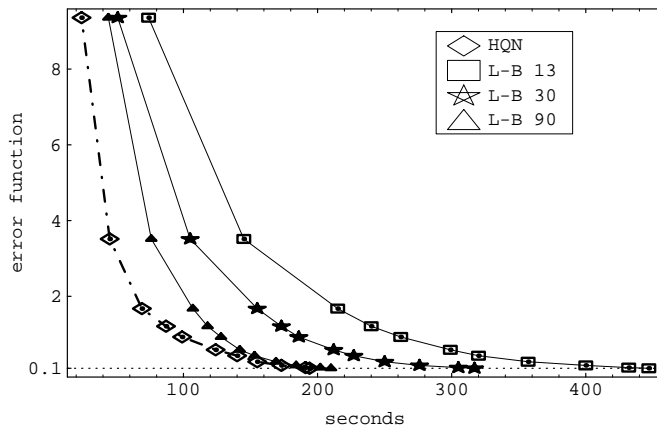


Figure 1. \mathcal{HQN} and L - $BFGS$ applied to a function of 1408 variables

SPACE COMPLEXITY: $O(n)$ = memory allocations for U , for vectors involved in iteration (3.4) and in computing \mathbf{d}_{k+1} ,

TIME COMPLEXITY (per step): $O(n \log n)$ = cost of $U \cdot \mathbf{z}$.

Numerical experiences on large scale problems [5] have shown that $\mathcal{S} \mathcal{LQN}$, $\mathcal{L} = \mathcal{H} \equiv$ Hartley algebra = $sd U$, $U_{ij} = \frac{1}{\sqrt{n}} (\cos \frac{2\pi ij}{n} + \sin \frac{2\pi ij}{n})$ [6], [3], [4] has a good rate of convergence and is competitive with the well known Limited-memory $BFGS$ method (for L - $BFGS$ see [13], [14], [1]). For example, in Figure 1 is reported the time required by $\mathcal{S} \mathcal{LQN}$, and by L - $BFGS$, $m = 13, 30, 90$, to minimize the error function associated with a 34-38-2 neural network for learning the ionosphere data (see [16]). Here the number of variables n is 1408 (for more details see [5]). We recall that the L - $BFGS$ procedure is defined in terms of the m pairs $(\mathbf{s}_j, \mathbf{y}_j)$, $j = k, \dots, k - m + 1$. Thus Figure 1 shows that strong storage requirements are needed ($m = 90$) in order to make L - $BFGS$ competitive with \mathcal{LQN} .

4. $\mathcal{L}^{(k)}$ QN METHODS

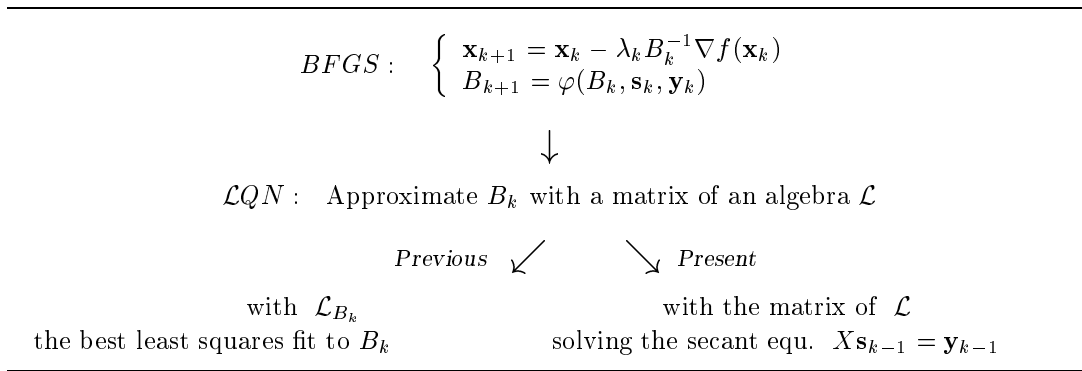
The idea in \mathcal{LQN} methods is to replace B_k in $B_{k+1} = \varphi(B_k, \mathbf{s}_k, \mathbf{y}_k)$ with a suitable matrix A_k of a structured algebra \mathcal{L} . In [10] this matrix A_k is the best approximation in the Frobenius norm of $B_k = \varphi(A_{k-1}, \mathbf{s}_{k-1}, \mathbf{y}_{k-1})$. In the present paper, we try to satisfy the secant equation

$$X \mathbf{s}_{k-1} = \mathbf{y}_{k-1} \tag{4.1}$$

by means of a suitable approximation A_k of B_k where A_k belongs to an algebra \mathcal{L} . It is clear that in order to implement this new idea, the space \mathcal{L} and therefore the structure of \mathcal{L} must change at each iteration k . The innovative $\mathcal{L}^{(k)}$ QN methods obtained in this way can better fit the Hessian structure, thereby improving the rate of convergence of the algorithm. As a matter of fact, some theoretical and experimental results reported in [9] had already suggested

that an adaptive choice of \mathcal{L} during the minimization process is perhaps the best way to obtain more efficient \mathcal{LQN} algorithms.

Both the previous and the innovative procedures are shown in the following scheme:



Let us introduce a basic criterion for choosing $\mathcal{L}^{(k)}$, by assuming

$$\mathcal{L}^{(k)} = sdU_k := \{U_k d(\mathbf{z}) U_k^* : \mathbf{z} \in \mathbb{C}^n\}, \quad U_k \text{ } n \times n \text{ unitary,} \quad (4.2)$$

at the generic step k .

Let A_k denote the matrix of $\mathcal{L}^{(k)}$ that we have to update. So

$$B_{k+1} = \varphi(A_k, \mathbf{s}_k, \mathbf{y}_k). \quad (4.3)$$

We require

- (i) A_k is p.d.,
- (ii) A_k solves the secant equation in the previous iteration, i.e. $A_k \mathbf{s}_{k-1} = \mathbf{y}_{k-1}$.

Note that the latter conditions may yield a matrix A_k which is not the best approximation in Frobenius norm of B_k in $\mathcal{L}^{(k)}$ (i.e. $A_k \neq \mathcal{L}_{B_k}^{(k)}$, in general).

Since A_k must be an element of the matrix algebra $\mathcal{L}^{(k)}$, it will have the form $A_k = U_k d(\mathbf{w}_k) U_k^*$, for some vector \mathbf{w}_k . Then, the secant condition (ii) can be rewritten in order to determine \mathbf{w}_k via U_k^* , i.e.

$$(\mathbf{w}_k)_i = \frac{(U_k^* \mathbf{y}_{k-1})_i}{(U_k^* \mathbf{s}_{k-1})_i}, \quad (U_k^* \mathbf{s}_{k-1})_i \neq 0, \quad \forall i. \quad (4.4)$$

Finally, the positive definiteness condition (i) is verified if $(\mathbf{w}_k)_i > 0$. So, the basic criterion for choosing $\mathcal{L}^{(k)}$ is the following one:

- Choose U_k such that

$$(\mathbf{w}_k)_i = \frac{(U_k^* \mathbf{y}_{k-1})_i}{(U_k^* \mathbf{s}_{k-1})_i} > 0, \quad \forall i \quad (4.5)$$

and define $\mathcal{L}^{(k)}$ as in (4.2).

Notice that (4.5) is equivalent to say that $\exists U_k$ unitary and $(\mathbf{w}_k)_i > 0$ such that the secant equation $U_k d(\mathbf{w}_k) U_k^* \mathbf{s}_{k-1} = \mathbf{y}_{k-1}$ is verified. By multiplying on the left by \mathbf{s}_{k-1}^T the latter equation, we find that $\mathbf{s}_{k-1}^T \mathbf{y}_{k-1} > 0$ is a necessary condition for the existence of U_k satisfying (4.5).

Once U_k is found, the corresponding space $\mathcal{L}^{(k)}$ includes the desired matrix satisfying (i) and (ii). Denote this matrix by $\mathcal{L}_{\mathbf{s}\mathbf{y}}^{(k)}$ and define the new Hessian approximation B_{k+1} by applying φ to $\mathcal{L}_{\mathbf{s}\mathbf{y}}^{(k)}$:

$$B_{k+1} = \varphi(\mathcal{L}_{\mathbf{s}\mathbf{y}}^{(k)}, \mathbf{s}_k, \mathbf{y}_k), \quad \mathcal{L}_{\mathbf{s}\mathbf{y}}^{(k)} \mathbf{s}_{k-1} = \mathbf{y}_{k-1}. \quad (\mathcal{L}^{(k)} QN)$$

The two possible descent directions are:

$$\mathbf{d}_{k+1} = \begin{cases} -B_{k+1}^{-1} \nabla f(\mathbf{x}_{k+1}), & \text{(I)} \\ -\mathcal{L}_{\mathbf{s}\mathbf{y}}^{(k+1)-1} \nabla f(\mathbf{x}_{k+1}). & \text{(II)} \end{cases}$$

Notice that both directions (I) and (II) are defined in terms of matrices satisfying the secant equation.

The following questions arise: There exists a unitary matrix U_k satisfying the condition (4.5)? Can this matrix be easily obtained?

5. PRACTICAL $\mathcal{L}^{(k)}$ QN METHODS

We have observed that $\mathbf{s}_{k-1}^T \mathbf{y}_{k-1} > 0$ is a necessary condition for the existence of a unitary matrix U_k satisfying (4.5). Actually, the following result holds

Theorem 5.1. *The existence of a matrix U_k^* satisfying (4.5) is guaranteed iff*

$$\mathbf{y}_{k-1}^T \mathbf{s}_{k-1} > 0 \quad (5.1)$$

Observe that (5.1) is the condition required to obtain a p.d. matrix $B_k = \varphi(A_{k-1}, \mathbf{s}_{k-1}, \mathbf{y}_{k-1})$, i.e. (5.1) is already satisfied (remember that the inequality $\mathbf{s}_{k-1}^T \mathbf{y}_{k-1} > 0$ can be obtained by choosing the step-length λ_{k-1} in the AG_{k-1} set).

Let $H(\mathbf{z})$ denote the Householder matrix corresponding to the vector \mathbf{z} , i.e.

$$H(\mathbf{z}) = I - \frac{2}{\|\mathbf{z}\|^2} \mathbf{z}\mathbf{z}^*, \quad \mathbf{z} \in \mathbb{C}^n \quad (5.2)$$

($H(\mathbf{0}) = I$). In order to prove Theorem 5.1, we need a preliminary result which turn out to be useful for the explicit computation of U_k^* .

Lemma 5.2. *Given two vectors $\mathbf{s}, \mathbf{y} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$, let $\mathbf{r}, \mathbf{x} \in \mathbb{R}^n$ be such that $\|\mathbf{r}\| \|\mathbf{x}\| \neq 0$, $r_i \neq 0 \forall i$ and the cosine of the angle between \mathbf{r} and \mathbf{x} is equal to the cosine of the angle between \mathbf{s} and \mathbf{y} , i.e.*

$$\frac{\mathbf{r}^T \mathbf{x}}{\|\mathbf{r}\| \|\mathbf{x}\|} = \frac{\mathbf{s}^T \mathbf{y}}{\|\mathbf{s}\| \|\mathbf{y}\|}. \quad (5.3)$$

Set $\mathbf{u} = \mathbf{s} - \mathbf{y} - \left(\frac{\|\mathbf{s}\|}{\|\mathbf{r}\|} \mathbf{r} - \frac{\|\mathbf{y}\|}{\|\mathbf{x}\|} \mathbf{x} \right)$, $\mathbf{p} = H(\mathbf{u})\mathbf{s} - \frac{\|\mathbf{s}\|}{\|\mathbf{r}\|} \mathbf{r} = H(\mathbf{u})\mathbf{y} - \frac{\|\mathbf{y}\|}{\|\mathbf{x}\|} \mathbf{x}$ and

$$U^* = H(\mathbf{p})H(\mathbf{u}). \quad (5.4)$$

Then $U^* \mathbf{s} = \frac{\|\mathbf{s}\|}{\|\mathbf{r}\|} \mathbf{r}$, $U^* \mathbf{y} = \frac{\|\mathbf{y}\|}{\|\mathbf{x}\|} \mathbf{x}$ and

$$w_i = \frac{(U^* \mathbf{y})_i}{(U^* \mathbf{s})_i} = \frac{\|\mathbf{y}\| \|\mathbf{r}\| x_i}{\|\mathbf{s}\| \|\mathbf{x}\| r_i}.$$

Proof. First observe that if $\mathbf{p} = \mathbf{v} - \frac{\|\mathbf{v}\|}{\|\mathbf{z}\|} \mathbf{z}$, $\mathbf{v} \in \mathbb{R}^n$, $\mathbf{z} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$, then

$$H(\mathbf{p}) \mathbf{v} = \frac{\|\mathbf{v}\|}{\|\mathbf{z}\|} \mathbf{z}$$

(in fact, $\frac{\mathbf{p}^* \mathbf{v}}{\|\mathbf{p}\|^2} = \frac{1}{2}$). As a consequence, for any unitary matrix Q we have

$$\begin{aligned} \mathbf{p}_1 &= Q \mathbf{s} - \frac{\|\mathbf{s}\|}{\|\mathbf{r}\|} \mathbf{r}, \quad \mathbf{r} \neq \mathbf{0} \Rightarrow H(\mathbf{p}_1) Q \mathbf{s} = \frac{\|\mathbf{s}\|}{\|\mathbf{r}\|} \mathbf{r}, \\ \mathbf{p}_2 &= Q \mathbf{y} - \frac{\|\mathbf{y}\|}{\|\mathbf{x}\|} \mathbf{x}, \quad \mathbf{x} \neq \mathbf{0} \Rightarrow H(\mathbf{p}_2) Q \mathbf{y} = \frac{\|\mathbf{y}\|}{\|\mathbf{x}\|} \mathbf{x}. \end{aligned}$$

Now choose Q such that $\mathbf{p}_1 = \mathbf{p}_2$ or, equivalently,

$$Q(\mathbf{s} - \mathbf{y}) = \frac{\|\mathbf{s}\|}{\|\mathbf{r}\|} \mathbf{r} - \frac{\|\mathbf{y}\|}{\|\mathbf{x}\|} \mathbf{x}. \quad (5.5)$$

Such choice is possible provided that $\|\mathbf{s} - \mathbf{y}\| = \|\frac{\|\mathbf{s}\|}{\|\mathbf{r}\|} \mathbf{r} - \frac{\|\mathbf{y}\|}{\|\mathbf{x}\|} \mathbf{x}\|$ from which we deduce the condition (5.3) on \mathbf{r} and \mathbf{x} . A matrix Q satisfying (5.5) is $H(\mathbf{u})$, $\mathbf{u} = \mathbf{s} - \mathbf{y} - \left(\frac{\|\mathbf{s}\|}{\|\mathbf{r}\|} \mathbf{r} - \frac{\|\mathbf{y}\|}{\|\mathbf{x}\|} \mathbf{x} \right)$.

□

Now, in order to construct U_k^* satisfying (4.5), we need to prove the effective existence of $\mathbf{r}, \mathbf{s} \in \mathbb{R}^n$ such that (5.3) holds with $r_i x_i > 0$.

Proof of Theorem 5.1. Given the two vectors \mathbf{s}_{k-1} and \mathbf{y}_{k-1} , an example of vector pair (\mathbf{x}, \mathbf{r}) such that

$$\frac{\mathbf{r}^T \mathbf{x}}{\|\mathbf{r}\| \|\mathbf{x}\|} = \frac{\mathbf{s}_{k-1}^T \mathbf{y}_{k-1}}{\|\mathbf{s}_{k-1}\| \|\mathbf{y}_{k-1}\|} \equiv \sqrt{\beta_{k-1}}, \quad r_i x_i > 0 \quad \forall i. \quad (5.6)$$

is the following:

$$\mathbf{x} = [1 \ \epsilon \ \cdots \ \epsilon]^T, \quad \mathbf{r} = [\epsilon \ \cdots \ \epsilon 1]^T, \quad \epsilon = \epsilon(\sqrt{\beta_{k-1}}),$$

where

$$\epsilon(\sqrt{\beta}) = \frac{\sqrt{\beta}}{1 + \sqrt{1 - \beta(n-1)} + \sqrt{\beta(n-2)}}.$$

In fact, $\epsilon(\sqrt{\beta}) > 0$ if $0 < \sqrt{\beta} \leq 1$. So, under the condition (5.1), we have that \mathbf{x} and \mathbf{r} have positive entries. Once \mathbf{x} and \mathbf{r} have been introduced, define the two vectors \mathbf{u} and \mathbf{p} as in Lemma 5.2, in terms of \mathbf{s}_{k-1} , \mathbf{y}_{k-1} , \mathbf{r} , \mathbf{x} , and consider the corresponding Householder matrices $H(\mathbf{u})$ and $H(\mathbf{p})$. Then the matrix U_k^* is $H(\mathbf{p})H(\mathbf{u})$. In fact, $H(\mathbf{p})H(\mathbf{u})$ maps \mathbf{s}_{k-1} and \mathbf{y}_{k-1} into two vectors whose directions are the same of \mathbf{r} and \mathbf{x} , respectively. So, by the condition $r_i x_i > 0$, the ratio of the two transformed vectors has positive entries:

$$(\mathbf{w}_k)_i = \frac{(U_k^* \mathbf{y}_{k-1})_i}{(U_k^* \mathbf{s}_{k-1})_i} = \frac{\|\mathbf{y}_{k-1}\| \|\mathbf{r}\| x_i}{\|\mathbf{s}_{k-1}\| \|\mathbf{x}\| r_i} > 0 \quad \forall i. \quad (5.7)$$

□

Notice that if we define U_k^* as the product of the two Householder matrices $H(\mathbf{p})$ and $H(\mathbf{u})$ (as above suggested), then the corresponding $\mathcal{L}^{(k)}QN$ method can be implemented with only $O(n)$ arithmetic operations per step and $O(n)$ memory allocations. This result is easily obtained from the identity (method (I))

$$\mathbf{d}_{k+1} = -\varphi(U_k d(\mathbf{w}_k)U_k^*, \mathbf{s}_k, \mathbf{y}_k)^{-1} \nabla f(\mathbf{x}_{k+1})$$

by applying the Shermann-Morrison-Woodbury formula. Obviously, the method (II) can be implemented with the same complexity.

6. ALTERNATIVE $\mathcal{L}^{(k)}QN$ METHODS

In this section an alternative $\mathcal{L}^{(k)}QN$ method is obtained in order to regain the best least squares fit condition.

Let U_k be a unitary matrix satisfying the condition (4.5), so that the corresponding matrix algebra $\mathcal{L}^{(k)}$ includes a p.d. matrix $\mathcal{L}_{\mathbf{y}}^{(k)}$ solving the previous secant equation $X\mathbf{s}_{k-1} = \mathbf{y}_{k-1}$. Pick up in $\mathcal{L}^{(k)}$ the best approximation $\mathcal{L}_{B_k}^{(k)}$ in the Frobenius norm of B_k and apply φ to $\mathcal{L}_{B_k}^{(k)}$. So:

$$B_{k+1} = \varphi(\mathcal{L}_{B_k}^{(k)}, \mathbf{s}_k, \mathbf{y}_k). \tag{Alternative $\mathcal{L}^{(k)}QN$ }$$

Since

$$B_k \text{ p.d. } \Rightarrow \left. \begin{array}{l} \mathcal{L}_{B_k}^{(k)} \text{ p.d.} \\ \mathbf{s}_k^T \mathbf{y}_k > 0 \end{array} \right\} \Rightarrow B_{k+1} \text{ p.d. } \Rightarrow \mathcal{L}_{B_{k+1}}^{(k+1)} \text{ p.d.},$$

the latter definition of B_{k+1} leads to two possible descent directions which are expressed in terms of B_{k+1} and in terms of $\mathcal{L}_{B_{k+1}}^{(k+1)}$, respectively. The former leads to a Secant method, the latter to a Non Secant one:

$$\mathbf{d}_{k+1} = \begin{cases} -B_{k+1}^{-1} \nabla f(\mathbf{x}_{k+1}) & \mathcal{S} \mathcal{L}^{(k)}QN \\ -\mathcal{L}_{B_{k+1}}^{(k+1)-1} \nabla f(\mathbf{x}_{k+1}) & \mathcal{NS} \mathcal{L}^{(k)}QN \end{cases}$$

Since $\mathcal{NS} \mathcal{L}^{(k)}QN$ is a *BFGS*-type algorithm where $\tilde{B}_k = \mathcal{L}_{B_k}^{(k)}$ (see [10]) and, by (3.2), $\mathcal{L}_{B_k}^{(k)}$ satisfies the conditions $\det B_k \leq \det \mathcal{L}_{B_k}^{(k)}$ and $\text{tr } B_k \geq \text{tr } \mathcal{L}_{B_k}^{(k)}$, we can apply Theorem 3.2 of [10], thereby stating the following convergence results.

Theorem 6.1. *If the $\mathcal{NS} \mathcal{L}^{(k)}QN$ iterates $\{\mathbf{x}_k\}$, defined with $\lambda_k \in AG$, satisfy the condition*

$$\frac{\|\mathbf{y}_k\|^2}{\mathbf{y}_k^T \mathbf{s}_k} \leq M, \tag{6.1}$$

for some constant M , then a subsequence of the gradients converges to the null vector. If, moreover, the level set $\mathcal{I}_0 = \{\mathbf{x} : f(\mathbf{x}) \leq f(\mathbf{x}_0)\}$ is bounded, then a subsequence of $\{\mathbf{x}_k\}$ converges to a stationary point \mathbf{x}_ of f and $f(\mathbf{x}_k) \rightarrow f(\mathbf{x}_*)$.*

Corollary 6.2. *Let f be a twice continuously differentiable convex function in the level set \mathcal{I}_0 . Assume \mathcal{I}_0 convex and bounded. Then all the assertions of Theorem 6.1 hold, moreover $f(\mathbf{x}_*) = \min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x})$.*

In the next section, experimental results (where the number of variables is $n \leq 32$) will clearly show that the novel $\mathcal{NS} \mathcal{L}^{(k)}QN$ outperforms the previous $\mathcal{NS} \mathcal{L}QN$ [10] in which the space \mathcal{L} is maintained unchanged in the optimization procedure.

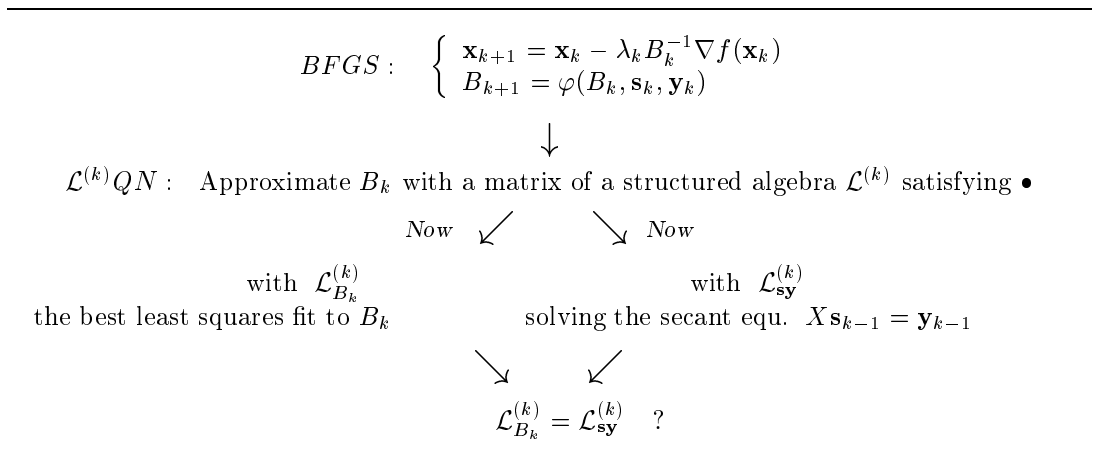
So we have introduced two different $\mathcal{L}^{(k)}QN$ methods. In the first one, in the definition $B_{k+1} = \varphi(B_k, \mathbf{s}_k, \mathbf{y}_k)$, the matrix B_k is replaced with $\mathcal{L}_{\mathbf{sy}}^{(k)}$, i.e. the matrix of $\mathcal{L}^{(k)}$ solving the secant equation. In the second one, B_k is replaced with $\mathcal{L}_{B_k}^{(k)}$, i.e. the best l.s. fit in $\mathcal{L}^{(k)}$ of B_k . Both methods exploit the same criterion \bullet for choosing $\mathcal{L}^{(k)}$.

Open problem: Is it possible to compute an approximation A_k (in $\mathcal{L}^{(k)}$) whose eigenvalues are linked to those of B_k and whose structure is close to a solution of the secant equation? If one is able to find a *structured* (or *low complexity*) matrix algebra $\mathcal{L}^{(k)}$ such that

$$\mathcal{L}_{\mathbf{sy}}^{(k)} = \mathcal{L}_{B_k}^{(k)},$$

the corresponding $\mathcal{L}^{(k)}QN$ algorithm would be the optimal one.

We may summarize the main ideas of the present paper in the following scheme



7. PERFORMANCE OF $\mathcal{L}^{(k)}QN$ METHODS

We have compared the performances of $\mathcal{L}^{(k)}QN$ and $\mathcal{L}QN$ methods in the minimization of some simple test functions f taken from [8]. The $\mathcal{L}QN$ method is implemented with $\mathcal{L} = \mathcal{H}$, where \mathcal{H} is the Hartley algebra. The matrices U_k utilized in $\mathcal{L}^{(k)}QN$ methods are the product of two Householder matrices, as suggested in Section 5.

Table I reports the number of iterations required by $\mathcal{NS} \mathcal{L}QN$ (Section 3) and $\mathcal{NS} \mathcal{L}^{(k)}QN$ (Section 6) to obtain $f(\mathbf{x}_k) < \epsilon$, $\epsilon = 10^{-4}, 10^{-6}, 10^{-8}$. It is clear that the rate of convergence of *Non Secant* methods may be considerably improved by changing the space \mathcal{L} at each iteration k . Recall that \mathcal{NS} methods are convergent.

The same set of benchmarks is exploited to study the behaviour of the algorithms $\mathcal{S} \mathcal{L}QN$ (Section 3), $\mathcal{S} \mathcal{L}^{(k)}QN$ (Section 6) and $\mathcal{L}^{(k)}QN$ (I) (Section 4). Table II shows that the latter methods are faster than the \mathcal{NS} algorithms. Moreover, $\mathcal{S} \mathcal{L}^{(k)}QN$ and $\mathcal{L}^{(k)}QN$ (I) turn out to be superior to $\mathcal{S} \mathcal{L}QN$ in most cases.

f	upd.matr.	10^{-4}	10^{-6}	10^{-8}
Rosenbrock $n = 2$	\mathcal{L}	364	535	677
	$\mathcal{L}^{(k)}$	75	112	149
Helical $n = 3$	\mathcal{L}	447		
	$\mathcal{L}^{(k)}$	62	83	114
Powell $n = 4$	\mathcal{L}	338	>2000	
	$\mathcal{L}^{(k)}$	87	165	269
Wood $n = 4$	\mathcal{L}	277	439	623
	$\mathcal{L}^{(k)}$	121	188	223
Trigon. $n = 32$	\mathcal{L}	48		
	$\mathcal{L}^{(k)}$	29		

Table I. Performance of Non Secant methods

f	upd.matr.	10^{-4}	10^{-6}	10^{-8}
Rosenbrock $n = 2$	\mathcal{L}	11	13	16
	$\mathcal{L}^{(k)}$	14	15	15
	$\mathcal{L}_{\mathbf{sy}}^{(k)}$	19	21	22
Helical $n = 3$	\mathcal{L}	22	29	36
	$\mathcal{L}^{(k)}$	23	25	28
	$\mathcal{L}_{\mathbf{sy}}^{(k)}$	23	25	27
Powell $n = 4$	\mathcal{L}	29	47	175
	$\mathcal{L}^{(k)}$	32	56	62
	$\mathcal{L}_{\mathbf{sy}}^{(k)}$	20	21	36
Wood $n = 4$	\mathcal{L}	49	67	95
	$\mathcal{L}^{(k)}$	54	78	80
	$\mathcal{L}_{\mathbf{sy}}^{(k)}$	24	41	45
Trigon. $n = 32$	\mathcal{L}	22		
	$\mathcal{L}^{(k)}$	20		
	$\mathcal{L}_{\mathbf{sy}}^{(k)}$	27		

Table II. Performance of Secant methods

REFERENCES

1. Al Baali M. Improved Hessian approximations for the limited memory *BFGS* method. *Numerical Algorithms* 1999; **22**:99–112.
2. Battiti R. First- and second-order methods for learning: between steepest descent and Newton’s method. *Neural Computation* 1992; **4**:141–166.
3. Bini D, Favati P. On a matrix algebra related to the discrete Hartley transform. *SIAM J. Matrix Anal. Appl.* 1993; **14**:500–507.
4. Bortoletti A, Di Fiore C. On a set of matrix algebras related to discrete Hartley-type transforms. *Linear Algebra Appl.* 2003; **366**:65–85.
5. Bortoletti A, Di Fiore C, Fanelli S, Zellini P. A new class of quasi-Newtonian methods for optimal learning in *MLP*-networks. *IEEE Transactions on Neural Networks* 2003; **14**(2):263–273.
6. Bracewell RN. The fast Hartley transform. *Proc. IEEE* 1984; **72**:1010–1018.
7. Dennis JE, Moré JJ. Quasi-Newton methods, motivation and theory. *SIAM Review* 1977; **19**:46–89.

8. Dennis JE, Schnabel RB. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs: New Jersey, 1983.
9. Di Fiore C. Structured matrices in unconstrained minimization methods. *Contemporary Mathematics* 2003; **323**:205–219.
10. Di Fiore C, Fanelli S, Lepore F, Zellini P. Matrix algebras in quasi-Newton methods for unconstrained minimization. *Numerische Mathematik* 2003; **94**:479–500.
11. Di Fiore C, Lepore F, Zellini P. Hartley-type algebras in displacement and optimization strategies. *Linear Algebra Appl.* 2003; **366**:215–232.
12. Fanelli S, Paparo P, Protasi M. Improving performances of Battiti-Shanno's quasi-newtonian algorithms for learning in feed-forward neural networks. in *Proc. of the 2nd Australian and New Zealand Conference on Intelligent Information Systems*. (Brisbane, Australia, Nov.-Dec. 1994), 1994;115–119.
13. Liu DC, Nocedal J. On the limited memory *BFGS* method for large scale optimization. *Math. Programming* 1989; **45**:503–528.
14. Nocedal J, Wright SJ. *Numerical Optimization*. Springer: New York, 1999.
15. Powell MJD. Some global convergence properties of a variable metric algorithm for minimization without exact line searches. in *Nonlinear Programming, SIAM-AMS Proc*, vol. 9. (New York, March 1975), Cottle RW, Lemke CE (eds). Providence, 1976; 53–72.
16. Repository of Machine Learning databases.
<http://www.ics.uci.edu/~mlern/MLRepository.html> [October 2002]