NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. 2005; **12**:755–768
Published online 15 June 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nla.449

Low-complexity minimization algorithms

Carmine Di Fiore[*,†], Stefano Fanelli[‡] and Paolo Zellini[§]

*Department of Mathematics, University of Roma 'Tor Vergata', Via della Ricerca Scientifica,
00133 Roma, Italy*

SUMMARY

Structured matrix algebras $\mathscr{L}$ and a generalized BFGS-type iterative scheme have been recently investigated to introduce low-complexity quasi-Newton methods, named $\mathscr{L}$QN, for solving general (non-structured) minimization problems. In this paper we introduce the $\mathscr{L}^k$QN methods, which exploit *ad hoc* algebras at 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 © 2005 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 (QN) algorithms for the minimization of a function $f:\mathbb{R}^n \rightarrow \mathbb{R}$, which are a generalization of some previous methods introduced in Reference [1]. The innovative algorithms, named $\mathscr{L}^k$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

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

where the $n \times n$ matrix A_k is picked up in a structured matrix algebra $\mathscr{L}^k$, shares some significant property with B_k and is p.d. In (1), $\varphi(\cdot, \mathbf{s}_k, \mathbf{y}_k)$ denotes a function updating p.d. matrices into p.d. matrices, whenever $\mathbf{s}_k^{\mathrm{T}} \mathbf{y}_k > 0$, i.e.

$$\left.\begin{array}{r} A \text{ p.d.} \\ \mathbf{s}_k^{\mathrm{T}} \mathbf{y}_k > 0 \end{array}\right\} \Rightarrow \varphi(A, \mathbf{s}_k, \mathbf{y}_k) \text{ p.d.} \tag{2}$$

[*] Correspondence to: Carmine Di Fiore, Department of Mathematics, University of Roma 'Tor Vergata', Via della Ricerca Scientifica, 00133 Roma, Italy.
[†] E-mail: difiore@mat.uniroma2.it
[‡] E-mail: fanelli@mat.uniroma2.it
[§] E-mail: zellini@mat.uniroma2.it

Received 5 December 2003
Revised 26 March 2004

Condition (2) is in particular satisfied by the BFGS updating formula (5) in the next section, whereas a suitable choice of the step length λ_k assures the inequality $\mathbf{s}_k^{\mathrm{T}}\mathbf{y}_k > 0$ by using classical Armijo–Goldstein conditions (see (7) or Reference [2]). We underline that other possible updating formulas could be utilized (see e.g. Reference [3]).

Hessian approximations of type (1) were studied in the case $\mathscr{L}^k = \mathscr{L}$ for any k, where $\mathscr{L}$ is a fixed space, and the matrix A_k is the best approximation in the Frobenius norm of B_k in $\mathscr{L}$ [1]. Such matrix A_k, denoted by $\mathscr{L}_{B_k}$, inherites positive definiteness from B_k. So, by property (2), $B_{k+1} = \varphi(\mathscr{L}_{B_k}, \mathbf{s}_k, \mathbf{y}_k)$ is a p.d. matrix (provided that $\mathbf{s}_k^{\mathrm{T}}\mathbf{y}_k > 0$). As a consequence, the $\mathscr{L}$QN methods of Reference [1] yield a descent direction $\mathbf{d}_{k+1}$.

If $\mathscr{L}$ is defined as the set $\mathrm{sd}\, U$ of all matrices simultaneously diagonalized by a fast discrete transform U (see (8)), then the time and space complexity of $\mathscr{L}$QN is $O(n \log n)$ and $O(n)$, respectively [1, 4]. The latter result makes $\mathscr{L}$QN methods suitable for minimizing functions f where n is large. In fact, numerical experiences show the competitivity of $\mathscr{L}$QN with limited-memory BFGS (L-BFGS), which is an efficient method for solving large-scale problems [4]. Moreover, a global linear convergence result for the class of $\mathscr{N}\mathscr{S}$ $\mathscr{L}$QN methods is obtained in References [1, 5], by extending the analogous BFGS convergence result of Powell [6] with a proper use of some crucial properties of the matrix $\mathscr{L}_{B_k}$.

The local convergence properties of $\mathscr{L}$QN were studied in Reference [7]. It is proved, in particular, that $\mathscr{L}$QN converges to a minimum point $\mathbf{x}_*$ of f with a superlinear rate of convergence whenever $\nabla^2 f(\mathbf{x}_*) \in \mathscr{L}$. The latter result is rather restrictive but suggests that, in order to improve $\mathscr{L}$QN efficiency, one might modify the algebra $\mathscr{L}$ in each iteration k, i.e. introduce the $\mathscr{L}^k$QN methods. This requires a concept of 'closeness' of a *space* $\mathscr{L}^k$ with respect to a *matrix* B_k and the construction, at each iteration, of a space $\mathscr{L}^k$ 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 $\mathscr{L}^k$ is 'close' to B_k if $\mathscr{L}^k$ includes matrices satisfying the latter properties, i.e. if

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

Once such space $\mathscr{L}^k$ is introduced, we can conceive at least two $\mathscr{L}^k$QN algorithms, based on the updating formula (1):

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

Algorithm 2: (1) with $A_k = \mathscr{L}^k_{B_k}$, where $\mathscr{L}^k_{B_k}$ is the best least squares fit to B_k in $\mathscr{L}^k$.

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) [2, 8]. In Section 3 we describe the basic properties of the $\mathscr{L}$QN methods, recently introduced in Reference [1]. The latter methods turn out to be more efficient than BFGS and extremely competitive with L-BFGS for solving large-scale minimization problems [4]. In order to improve the $\mathscr{L}$QN efficiency, in Section 4 we introduce the innovative $\mathscr{L}^k$QN algorithms. Assuming that $\mathscr{L}^k$ 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 (16)); then, we define in detail the $\mathscr{L}^k$QN Algorithm 1. In Section 5 we prove that matrix algebras $\mathscr{L}^k$ satisfying • exist without special conditions on the algorithm. Moreover, we illustrate how to compute such $\mathscr{L}^k$, by expressing the corresponding matrix U_k as a product of two Householder matrices. In Section 6 we introduce the $\mathscr{L}^k$QN Algorithm 2, and we prove that the $\mathscr{N}\mathscr{S}$ version of such algorithm is convergent. Moreover, we discuss some possible improvements of Algorithms 1 and 2 (see also Section 7). In particular, we define a method $\mathscr{L}^k$QN which turns out to be superior to $\mathscr{L}$QN in solving large-scale minimization problems.

2. QUASI-NEWTON METHODS FOR THE UNCONSTRAINED MINIMIZATION

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

$$f(\mathbf{x}_*) = \min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \quad \text{find } \mathbf{x}_* \tag{3}$$

Let us apply a QN method to the gradient vector function ∇f. Given $\mathbf{x}_0 \in \mathbb{R}^n$, $B_0 = n \times n$ 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:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k, \quad \mathbf{d}_k = -B_k^{-1}\nabla f(\mathbf{x}_k) \quad (\lambda_k \in \mathbb{R}^+) \tag{4}$$

The matrix B_k is chosen p.d. so that the search direction $\mathbf{d}_k$ is always a descent direction $(\nabla f(\mathbf{x}_k)^{\mathrm{T}}\mathbf{d}_k < 0)$. 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 $\mathbf{y}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)$:

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

Note 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) [2, 8, 9], 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

$$\varphi(B, \mathbf{s}, \mathbf{y}) = B + \frac{1}{\mathbf{y}^{\mathrm{T}}\mathbf{s}}\mathbf{y}\mathbf{y}^{\mathrm{T}} - \frac{1}{\mathbf{s}^{\mathrm{T}}B\mathbf{s}}B\mathbf{s}\mathbf{s}^{\mathrm{T}}B \tag{5}$$

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:

$$\left.\begin{array}{r} B_k \text{ p.d.} \\ \mathbf{s}_k^{\mathrm{T}}\mathbf{y}_k > 0 \end{array}\right\} \Rightarrow B_{k+1} = \varphi(B_k, \mathbf{s}_k, \mathbf{y}_k) \text{ p.d.} \tag{6}$$

The condition $\mathbf{s}_k^{\mathrm{T}}\mathbf{y}_k>0$ can be assured by a suitable choice of the step length λ_k [2]. In particular, it is satisfied if λ_k is chosen in the Armijo–Goldstein set

$$\begin{aligned} \mathrm{AG}_k=\{\lambda\in\mathbb{R}^+ : f(\mathbf{x}_k+\lambda\mathbf{d}_k)\leqslant f(\mathbf{x}_k)+c_1\lambda\mathbf{d}_k^{\mathrm{T}}\nabla f(\mathbf{x}_k) \quad &\text{and} \\ \mathbf{d}_k^{\mathrm{T}}\nabla f(\mathbf{x}_k+\lambda\mathbf{d}_k)\geqslant c_2\mathbf{d}_k^{\mathrm{T}}\nabla f(\mathbf{x}_k)\}& \\ 0<c_1<c_2<1 \qquad & \end{aligned} \tag{7}$$

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 (3) the number n of variables is large. Such large scale problems arise, for example, in the learning process of neural networks [4, 10].

3. $\mathscr{L}$QN METHODS

The aim of $\mathscr{L}$QN methods [1] 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 e.g. References [11–13]). The main idea in Reference [1] is to replace B_k with a simpler matrix chosen in an algebra $\mathscr{L}$. Let U be a $n\times n$ unitary matrix and define $\mathscr{L}$ as the set of all matrices diagonalized by U ($\mathscr{L}=\mathrm{sd}\,U$):

$$\mathscr{L}=\mathrm{sd}\,U:=\{Ud(\mathbf{z})U^* : \mathbf{z}\in\mathbb{C}^n\},\quad d(\mathbf{z})=\begin{bmatrix} z_1 & & O \\ & \ddots & \\ O & & z_n \end{bmatrix} \tag{8}$$

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

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

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

$$\mathscr{L}_{B_k}=Ud(\mathbf{z}_k)U^*,\quad (\mathbf{z}_k)_i=[U\mathbf{e}_i]^*B_k[U\mathbf{e}_i] \tag{9}$$

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

$$\left.\begin{aligned} B_k\ \text{p.d.}\ \Rightarrow\ \mathscr{L}_{B_k}\ \text{p.d.} \\ \mathbf{s}_k^{\mathrm{T}}\mathbf{y}_k>0 \end{aligned}\right\}\ \Rightarrow\ B_{k+1}\ \text{p.d.}\ \Rightarrow\ \mathscr{L}_{B_{k+1}}\ \text{p.d.} \tag{10}$$

Thus we have two possible descent directions.

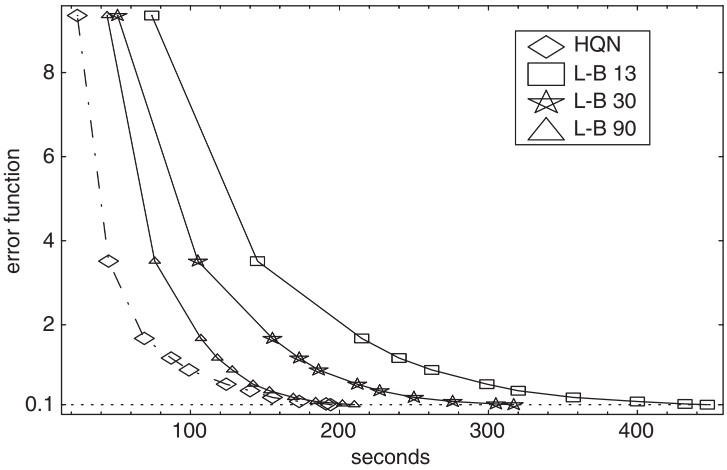

Figure 1. $\mathscr{H}$QN and L-BFGS applied to a function of 1408 variables.

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

$$\mathbf{d}_{k+1} = -B_{k+1}^{-1}\nabla f(\mathbf{x}_{k+1}) \qquad (\mathscr{S}\,\mathscr{L}\text{QN})$$

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

2. The second one in terms of $\mathscr{L}_{B_{k+1}}$, leading to a non-secant method:

$$\mathbf{d}_{k+1} = -\mathscr{L}_{B_{k+1}}^{-1}\nabla f(\mathbf{x}_{k+1}) \qquad (\mathscr{N}\mathscr{S}\,\mathscr{L}\text{QN})$$

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

In References [1, 5] it is proved that $\mathscr{N}\mathscr{S}\,\mathscr{L}$QN has a linear rate of convergence, whereas numerical experiences in Reference [4] show that S $\mathscr{L}$QN has a faster convergence rate.

Moreover, each step of any $\mathscr{L}$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

$$\mathbf{z}_{k+1} = \mathbf{z}_k + \frac{1}{\mathbf{s}_k^{\mathrm{T}}\mathbf{y}_k}|U^*\mathbf{y}_k|^2 - \frac{1}{\mathbf{z}_k^{\mathrm{T}}|U^*\mathbf{s}_k|^2}d(\mathbf{z}_k)^2|U^*\mathbf{s}_k|^2 \tag{11}$$

and, *in the secant case*, the Shermann–Morrison–Woodbury inversion formula (see References [1, 4]). It is well known in the literature that this formula can be unstable from a numerical point of view [14]. However, the experiments performed on $\mathscr{S}\,\mathscr{L}$QN methods have not pointed out significant difficulties (see Figure 1, Section 7 and Reference [4]). Thus, if U defines a fast discrete transform ($\mathscr{L}$ structured), then $\mathscr{L}$QN can be implemented with
SPACE COMPLEXITY: $O(n) =$ memory allocations for U, for vectors involved in iteration (11) 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 [4] have shown that $\mathscr{S}\,\mathscr{L}$QN, $\mathscr{L} = \mathscr{H} \equiv$ Hartley algebra $= \mathrm{sd}\, U$, $U_{ij} = 1/\sqrt{n}(\cos(2\pi ij/n) + \sin(2\pi ij/n))$ [15–17] has a good rate of convergence and is competitive with the well-known L-BFGS method (for L-BFGS see References [8, 13, 18]). For example, in Figure 1 is reported the time required by $\mathscr{S}\mathscr{L}$QN, 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 Reference [10]). Here the number of variables n is 1408 (for more details see Reference [4]). 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 $\mathscr{L}$QN.

4. $\mathscr{L}^k$QN METHODS

The idea in $\mathscr{L}$QN 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 $\mathscr{L}$. In Reference [1] 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

$$X\mathbf{s}_{k-1}=\mathbf{y}_{k-1} \tag{12}$$

by means of a suitable approximation A_k of B_k where A_k belongs to an algebra $\mathscr{L}$. It is clear that in order to implement this new idea, the space $\mathscr{L}$ and therefore the structure of $\mathscr{L}$ must change at each iteration k. The innovative $\mathscr{L}^k$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 Reference [7] had already suggested that an adaptive choice of $\mathscr{L}$ during the minimization process is perhaps the best way to obtain more efficient $\mathscr{L}$QN algorithms.

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

$$\text{BFGS}:\quad \begin{cases}\mathbf{x}_{k+1}=\mathbf{x}_k-\lambda_k B_k^{-1}\nabla f(\mathbf{x}_k)\\ B_{k+1}=\varphi(B_k,\mathbf{s}_k,\mathbf{y}_k)\end{cases}$$

$$\downarrow$$

$\mathscr{L}$QN : Approximate B_k with a matrix of an algebra $\mathscr{L}$

Previous ↙ ↘ Present

with $\mathscr{L}_{B_k}$
the best least squares fit to B_k in $\mathscr{L}$

with the matrix of $\mathscr{L}$
solving the secant equation $X\mathbf{s}_{k-1}=\mathbf{y}_{k-1}$

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

$$\mathscr{L}^k=\text{sd}\,U_k:=\{U_k d(\mathbf{z})U_k^*:\mathbf{z}\in\mathbb{C}^n\},\quad U_k\; n\times n \text{ unitary} \tag{13}$$

at the generic step k.

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

$$B_{k+1}=\varphi(A_k,\mathbf{s}_k,\mathbf{y}_k) \tag{14}$$

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 $\mathscr{L}^k$ (i.e. $A_k\neq\mathscr{L}^k_{B_k}$, in general).

Since A_k must be an element of the matrix algebra $\mathscr{L}^k$, it will have the form $A_k=U_kd(\mathbf{w}_k)U_k^*$, 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.

$$(\mathbf{w}_k)_i=\frac{(U_k^*\mathbf{y}_{k-1})_i}{(U_k^*\mathbf{s}_{k-1})_i},\quad (U_k^*\mathbf{s}_{k-1})_i\neq 0\ \forall i \tag{15}$$

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

$$(\mathbf{w}_k)_i=\frac{(U_k^*\mathbf{y}_{k-1})_i}{(U_k^*\mathbf{s}_{k-1})_i}>0\ \forall i \tag{16}$$

and define $\mathscr{L}^k$ as in (13).
Note that (16) is equivalent to say that $\exists U_k$ unitary and $(\mathbf{w}_k)_i>0$ such that the secant equation $U_kd(\mathbf{w}_k)U_k^*\mathbf{s}_{k-1}=\mathbf{y}_{k-1}$ is verified. By multiplying on the left by $\mathbf{s}_{k-1}^{\mathrm{T}}$ the latter equation, we find that $\mathbf{s}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}>0$ is a necessary condition for the existence of U_k satisfying (16).

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

$$B_{k+1}=\varphi(\mathscr{L}^k_{\mathbf{sy}},\mathbf{s}_k,\mathbf{y}_k),\quad \mathscr{L}^k_{\mathbf{sy}}\mathbf{s}_{k-1}=\mathbf{y}_{k-1} \tag{$\mathscr{L}^k$QN}$$

The two possible descent directions are

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

Note 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 (16)? Can this matrix be easily obtained?

5. PRACTICAL $\mathscr{L}^k$QN METHODS

We have observed that $\mathbf{s}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}>0$ is a necessary condition for the existence of a unitary matrix U_k satisfying (16). Actually, the following result holds.

Theorem 5.1
The existence of a matrix U_k^* satisfying (16) is guaranteed iff

$$\mathbf{y}_{k-1}^{\mathrm{T}}\mathbf{s}_{k-1} > 0 \tag{17}$$

Observe that (17) 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. (17) is already satisfied (remember that the inequality $\mathbf{s}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1} > 0$ 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.

$$H(\mathbf{z}) = I - \frac{2}{\|\mathbf{z}\|^2}\mathbf{z}\mathbf{z}^*, \quad \mathbf{z} \in \mathbb{R}^n \tag{18}$$

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

Lemma 5.2
Given two vectors $\mathbf{s}, \mathbf{y} \in \mathbb{R}^n\backslash\{\mathbf{0}\}$, 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.

$$\frac{\mathbf{r}^{\mathrm{T}}\mathbf{x}}{\|\mathbf{r}\|\|\mathbf{x}\|} = \frac{\mathbf{s}^{\mathrm{T}}\mathbf{y}}{\|\mathbf{s}\|\|\mathbf{y}\|} \tag{19}$$

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

$$U^* = H(\mathbf{p})H(\mathbf{u}) \tag{20}$$

Then $U^*\mathbf{s} = (\|\mathbf{s}\|/\|\mathbf{r}\|)\mathbf{r}$, $U^*\mathbf{y} = (\|\mathbf{y}\|/\|\mathbf{x}\|)\mathbf{x}$ 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 $\mathbf{p} = \mathbf{v} - (\|\mathbf{v}\|/\|\mathbf{z}\|)\mathbf{z}$, $\mathbf{v} \in \mathbb{R}^n$, $\mathbf{z} \in \mathbb{R}^n\backslash\{\mathbf{0}\}$, then

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

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

$$\mathbf{p}_1 = Q\mathbf{s} - \frac{\|\mathbf{s}\|}{\|\mathbf{r}\|}\mathbf{r},\ \mathbf{r} \neq 0 \ \Rightarrow\ H(\mathbf{p}_1)Q\mathbf{s} = \frac{\|\mathbf{s}\|}{\|\mathbf{r}\|}\mathbf{r}$$

$$\mathbf{p}_2 = Q\mathbf{y} - \frac{\|\mathbf{y}\|}{\|\mathbf{x}\|}\mathbf{x},\ \mathbf{x} \neq 0 \ \Rightarrow\ H(\mathbf{p}_2)Q\mathbf{y} = \frac{\|\mathbf{y}\|}{\|\mathbf{x}\|}\mathbf{x}$$

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

$$Q(\mathbf{s} - \mathbf{y}) = \frac{\|\mathbf{s}\|}{\|\mathbf{r}\|}\mathbf{r} - \frac{\|\mathbf{y}\|}{\|\mathbf{x}\|}\mathbf{x} \tag{21}$$

Such choice is possible provided that $\|\mathbf{s}-\mathbf{y}\| = \|(\|\mathbf{s}\|/\|\mathbf{r}\|)\mathbf{r} - (\|\mathbf{y}\|/\|\mathbf{x}\|)\mathbf{x}\|$ from which we deduce the condition (19) on $\mathbf{r}$ and $\mathbf{x}$. A matrix Q satisfying (21) is $H(\mathbf{u})$, $\mathbf{u} = \mathbf{s} - \mathbf{y} - ((\|\mathbf{s}\|/\|\mathbf{r}\|)\mathbf{r} - (\|\mathbf{y}\|/\|\mathbf{x}\|)\mathbf{x})$. □

Now, in order to construct U_k^* satisfying (16), we need to prove the effective existence of $\mathbf{r}$, $\mathbf{s} \in \mathbb{R}^n$ such that (19) 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

$$\frac{\mathbf{r}^{\mathrm{T}}\mathbf{x}}{\|\mathbf{r}\|\|\mathbf{x}\|} = \frac{\mathbf{s}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}}{\|\mathbf{s}_{k-1}\|\|\mathbf{y}_{k-1}\|} \equiv \sqrt{\beta_{k-1}}, \quad r_i x_i > 0 \ \forall i \tag{22}$$

is the following:

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

where

$$\varepsilon(\alpha) = \frac{\alpha}{1 + \sqrt{1 - \alpha^2(n-1) + \alpha(n-2)}}$$

In fact, $\varepsilon(\alpha) > 0$ if $0 < \alpha \leqslant 1$. So, under condition (17), 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:

$$(\mathbf{w}_k)_i = \frac{(U_k^*\mathbf{y}_{k-1})_i}{(U_k^*\mathbf{s}_{k-1})_i} = \frac{\|\mathbf{y}_{k-1}\|\|\mathbf{r}\|x_i}{\|\mathbf{s}_{k-1}\|\|\mathbf{x}\|r_i} > 0 \quad \forall i \tag{23}$$

□

Note 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 $\mathscr{L}^k$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, method (II) can be implemented with the same complexity.

6. ALTERNATIVE $\mathscr{L}^k$QN METHODS

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

Let U_k be a unitary matrix satisfying condition (16), so that the corresponding matrix algebra $\mathscr{L}^k$ includes a p.d. matrix $\mathscr{L}_{\mathbf{sy}}^k$ solving the previous secant equation $X\mathbf{s}_{k-1} = \mathbf{y}_{k-1}$.

Pick up in $\mathscr{L}^k$ the best approximation $\mathscr{L}^k_{B_k}$ in the Frobenius norm of B_k and apply φ to $\mathscr{L}^k_{B_k}$. So

$$B_{k+1} = \varphi(\mathscr{L}^k_{B_k}, \mathbf{s}_k, \mathbf{y}_k) \qquad \text{(Alternative } \mathscr{L}^k\text{QN)}$$

Since

$$\left.\begin{array}{r} B_k \text{ p.d. } \Rightarrow \mathscr{L}^k_{B_k} \text{ p.d.} \\ \mathbf{s}_k^{\mathrm{T}}\mathbf{y}_k > 0 \end{array}\right\} \Rightarrow B_{k+1} \text{ p.d. } \Rightarrow \mathscr{L}^{k+1}_{B_{k+1}} \text{ p.d.}$$

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 $\mathscr{L}^{k+1}_{B_{k+1}}$, respectively. The former leads to a secant method, the latter to a non-secant one:

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

Since $\mathscr{N}\mathscr{S}\,\mathscr{L}^k$QN is a BFGS-type algorithm where $\tilde{B}_k = \mathscr{L}^k_{B_k}$ (see Reference [1]) and, by (9), $\mathscr{L}^k_{B_k}$ satisfies the conditions $\det B_k \leqslant \det \mathscr{L}^k_{B_k}$ and $\operatorname{tr} B_k \geqslant \operatorname{tr} \mathscr{L}^k_{B_k}$, we can apply Theorem 3.2 of Reference [1], thereby stating the following convergence results.

Theorem 6.1
If the $\mathscr{N}\mathscr{S}\,\mathscr{L}^k$QN iterates $\{\mathbf{x}_k\}$, defined with $\lambda_k \in AG$, satisfy the condition

$$\frac{\|\mathbf{y}_k\|^2}{\mathbf{y}_k^{\mathrm{T}}\mathbf{s}_k} \leqslant M \tag{24}$$

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

Corollary 6.2
Let f be a twice continuously differentiable convex function in the level set $\mathscr{I}_0$. Assume $\mathscr{I}_0$ 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 will clearly show that the novel $\mathscr{N}\mathscr{S}\,\mathscr{L}^k$QN outperforms the previous $\mathscr{N}\mathscr{S}\,\mathscr{L}$QN [1] in which the space $\mathscr{L}$ is maintained unchanged in the optimization procedure. However, $\mathscr{N}\mathscr{S}$ and $\mathscr{S}\,\mathscr{L}^k$QN methods cannot be applied in the present form to large-scale problems since no relation exists between $\mathscr{L}^{k+1}$ and $\mathscr{L}^k$ (or U_{k+1} and U_k) which allows to compute the eigenvalues $\mathbf{z}_{k+1}$ of $\mathscr{L}^{k+1}_{B_{k+1}}$ efficiently from the eigenvalues $\mathbf{z}_k$ of $\mathscr{L}^k_{B_k}$. A simple strategy to avoid this inconvenience consists of the following three stages:

1. At the step m_j introduce a space $\mathscr{L}^{m_j} = \operatorname{sd} U_{m_j}$ where the unitary matrix U_{m_j} satisfies the condition • (written in terms of m_j instead of k).

2. At the next steps $k \geqslant m_j$, choose $\mathscr{L}^k = \mathscr{L}^{m_j}$ until the matrix $U_{m_j}\,\mathrm{diag}([U_{m_j}^* \mathbf{y}_{k-1}]_i/[U_{m_j}^* \mathbf{s}_{k-1}]_i)\, U_{m_j}^*$ remains p.d. or, equivalently, until the space $\mathscr{L}^{m_j}$ includes a p.d. matrix solving the secant equation $X\mathbf{s}_{k-1} = \mathbf{y}_{k-1}$.
3. Set $m_{j+1} = k$, $j := j+1$, and go to 1.

The numerical experiments on this modified alternative $\mathscr{L}^k$QN algorithm confirm for the $\mathscr{S}\,\mathscr{L}^k$QN method a satisfactory numerical behaviour of the Shermann–Morrison–Woodbury formula. Observe, in fact, that a possible instability of the latter formula could be in any case easily detected by a pathological increase of time and/or number of iterations, particularly if one requires high-precision stopping rules.

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

$$\text{BFGS:}\quad \begin{cases} \mathbf{x}_{k+1} = \mathbf{x}_k - \lambda_k B_k^{-1} \nabla f(\mathbf{x}_k) \\ B_{k+1} = \varphi(B_k, \mathbf{s}_k, \mathbf{y}_k) \end{cases}$$

$$\downarrow$$

$\mathscr{L}^k$QN : Approximate B_k with a matrix of a structured algebra $\mathscr{L}^{m_j}$, $m_j \leqslant k$, satisfying • (written in terms of m_j instead of k)

Now $\swarrow$ $\searrow$ Now

with $\mathscr{L}^k_{\mathbf{sy}}$ $(m_j = k)$ solving the secant equation $X\mathbf{s}_{k-1} = \mathbf{y}_{k-1}$ in $\mathscr{L}^k$

with $\mathscr{L}^{m_j}_{B_k}$ $(m_j \leqslant k)$ the best least squares fit to B_k in $\mathscr{L}^{m_j}$

7. PERFORMANCE OF $\mathscr{L}^k$QN METHODS

We have compared the performances of $\mathscr{L}^k$QN and $\mathscr{L}$QN methods in the minimization of some simple test functions f taken from Reference [2] (see Tables I and II) and in solving the more difficult problem cited in Section 3 where f has 1408 independent variables (see Table III and Figure 2). The $\mathscr{L}$QN method is implemented with $\mathscr{L} = \mathscr{H}$, where $\mathscr{H}$ is the Hartley algebra. The matrices U_k utilized in $\mathscr{L}^k$QN methods are the product of two

Table I. Performance of non-secant methods.

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

Table II. Performance of secant methods.

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

Table III. $\mathscr{N}\mathscr{S}\,\mathscr{L}^k$QN and $\mathscr{N}\mathscr{S}\,\mathscr{L}$QN applied to a function of 1408 variables.

	k (sec) : $f(\mathbf{x}_k)<0.1$		
Method	$\mathbf{x}_0^{(1)}$	$\mathbf{x}_0^{(2)}$	$\mathbf{x}_0^{(3)}$
$\mathscr{N}\mathscr{S}\,\mathscr{L}$QN	7639 (696)	8742 (850)	9180 (901)
$\mathscr{N}\mathscr{S}\,\mathscr{L}^k$QN	2441 (183)	3772 (321)	3919 (341)

Householder matrices, as suggested in Section 5 (see the proof of Theorem 5.1). Table III and Figure 2 refer to experiences performed on a Pentium 4, 2 GHz, whereas Tables I and II (as well as Figure 1) report results of experiments run on an Alpha Server 800 5/333.

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

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

When n is large, the best performances of $\mathscr{L}^k$QN is obtained by slightly modifying the alternative $\mathscr{L}^k$QN methods, following procedure 1–3 suggested at the end of the previous section. The unitary matrix U_{m_j}, defining the space $\mathscr{L}^{m_j}=\mathrm{sd}\,U_{m_j}$ (see (13)), is in fact kept fixed until the quotients $w_i=[U^*_{m_j}\mathbf{y}_{k-1}]_i/[U^*_{m_j}\mathbf{s}_{k-1}]_i$, $k\geqslant m_j$, remain positive. The latter procedure allows to obtain *a more significant information on the spectrum of the Hessian* $\nabla^2 f(\mathbf{x}_{k+1})$ *at a lower cost with respect to* $\mathscr{L}$QN.

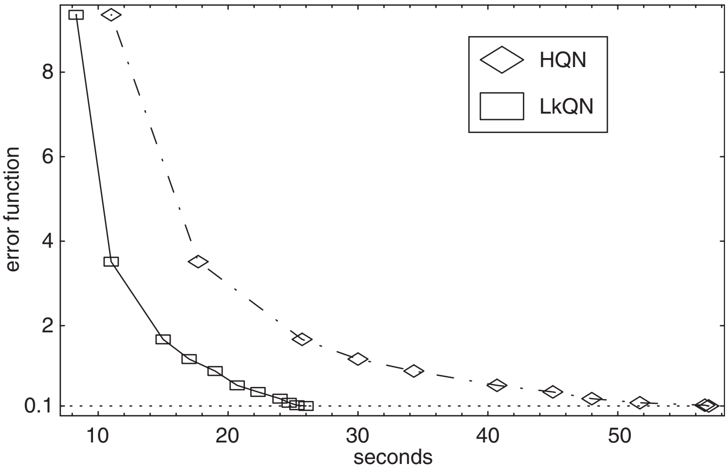

Figure 2. $\mathscr{L}^k$QN and $\mathscr{H}$QN applied to a function of 1408 variables.

As Table III and Figure 2 show, the performances of such modified alternative $\mathscr{N}\mathscr{S}$ and $\mathscr{S}\mathscr{L}^k$QN algorithms are extremely encouraging for solving the ionosphere problem [10] where $n = 1408$. In particular, this can be seen by comparing the results with those illustrated in Figure 1.

ACKNOWLEDGEMENTS

A special thanks to Dario Fasino for his remarkable suggestions on the use of Householder matrices.

REFERENCES

1. Di Fiore C, Fanelli S, Lepore F, Zellini P. Matrix algebras in quasi-Newton methods for unconstrained minimization. *Numerische Mathematik* 2003; **94**:479–500.
2. Dennis JE, Schnabel RB. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall: Englewood Cliffs, NJ, 1983.
3. Spedicato E, Xia Z. Solving secant equations via the ABS algorithm. *Optimization Methods and Software* 1992; **1**:243–252.
4. 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.
5. Di Fiore C, Lepore F, Zellini P. Hartley-type algebras in displacement and optimization strategies. *Linear Algebra and its Applications* 2003; **366**:215–232.
6. Powell MJD. Some global convergence properties of a variable metric algorithm for minimization without exact line searches. In *Nonlinear Programming*, *SIAM-AMS Proceedings*, Cottle RW, Lemke CE (eds), vol. 9. (New York, March 1975), Providence, 1976; 53–72.
7. Di Fiore C. Structured matrices in unconstrained minimization methods. *Contemporary Mathematics* 2003; **323**:205–219.
8. Nocedal J, Wright SJ. *Numerical Optimization*. Springer: New York, 1999.
9. Dennis JE, Moré JJ. Quasi-Newton methods, motivation and theory. *SIAM Review* 1977; **19**:46–89.
10. Repository of Machine Learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html (October 2002).
11. Battiti R. First- and second-order methods for learning: between steepest descent and Newton's method. *Neural Computation* 1992; **4**:141–166.
12. Fanelli S, Paparo P, Protasi M. Improving performances of Battiti-Shanno's quasi-Newtonian algorithms for learning in feed-forward neural networks. *Proceedings of the 2nd Australian and New Zealand Conference on Intelligent Information Systems*. (Brisbane, Australia, November–December 1994), 1994; 115–119.

13. Liu DC, Nocedal J. On the limited memory BFGS method for large-scale optimization. *Mathematical Programming* 1989; **45**:503–528.
14. Higham NJ. *Accuracy and Stability of Numerical Algorithms.* SIAM: Philadelphia, 1996.
15. Bracewell RN. The fast Hartley transform. *Proceedings of the IEEE* 1984; **72**:1010–1018.
16. Bini D, Favati P. On a matrix algebra related to the discrete Hartley transform. *SIAM Journal on Matrix Analysis and Applications* 1993; **14**:500–507.
17. Bortoletti A, Di Fiore C. On a set of matrix algebras related to discrete Hartley-type transforms. *Linear Algebra and its Applications* 2003; **366**:65–85.
18. Al Baali M. Improved Hessian approximations for the limited memory BFGS method. *Numerical Algorithms* 1999; **22**:99–112.