Linear Algebra and its Applications 471 (2015) 544–568

Contents lists available at ScienceDirect

Linear Algebra and its Applications

www.elsevier.com/locate/laa

Adaptive matrix algebras in unconstrained minimization

S. Cipolla, C. Di Fiore [*], F. Tudisco, P. Zellini

University of Rome "Tor Vergata", Department of Mathematics, Via della Ricerca Scientifica, 1 – 00133 Rome, Italy

ARTICLE INFO

Article history:
Received 8 August 2014
Accepted 9 January 2015
Available online 30 January 2015
Submitted by R. Brualdi

MSC:
65K10
90C53
47J25

Keywords:
Unconstrained minimization
Quasi-Newton methods
Matrix algebras
Iterative procedures

ABSTRACT

In this paper we study adaptive $\mathcal{L}^{(k)}$QN methods, involving special matrix algebras of low complexity, to solve general (non-structured) unconstrained minimization problems. These methods, which generalize the classical BFGS method, are based on an iterative formula which exploits, at each step, an *ad hoc* chosen matrix algebra $\mathcal{L}^{(k)}$. A global convergence result is obtained under suitable assumptions on f.

1. Introduction

Quasi-Newton methods for the unconstrained minimization of a function $f : \mathbb{R}^n \to \mathbb{R}$ are based on iterative schemes of the form $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$, where $\mathbf{d}_k$ is a descent direction in $\mathbf{x}_k$, i.e. $\nabla f(\mathbf{x}_k)^T \mathbf{d}_k < 0$, and λ_k is the steplength.

[*] Corresponding author.
E-mail addresses: cipolla@mat.uniroma2.it (S. Cipolla), difiore@mat.uniroma2.it (C. Di Fiore), tudisco@mat.uniroma2.it (F. Tudisco), zellini@mat.uniroma2.it (P. Zellini).

http://dx.doi.org/10.1016/j.laa.2015.01.010
0024-3795/© 2015 Elsevier Inc. All rights reserved.

Let us recall that *any* descent direction $\mathbf{d}_k$ for f in the current guess $\mathbf{x}_k$ solves the equation $A_k\mathbf{d}_k = -\mathbf{g}_k$ for some real symmetric positive definite (pd) matrix A_k approximating the Hessian of f in $\mathbf{x}_k$, where $\mathbf{g}_k$ is the first derivative vector $\nabla f(\mathbf{x}_k)$ (see [9]).

A *good* property that quasi-Newton methods should have, seems to be that A_{k+1} satisfies the equation $A_{k+1}\mathbf{s}_k = \mathbf{y}_k$ (Secant equation), where $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ and $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$. Quasi-Newton methods with such property will be referred to as Secant. Apparently, the secant equation is far to be a mere optional condition. In [12, p. 24] it is observed that the equality $A_{k+1}\mathbf{s}_k = \mathbf{y}_k$ mimics the fundamental property of the Hessian $\nabla^2 f(\mathbf{x}_{k+1})\mathbf{s}_k \approx \mathbf{y}_k$, whereas in [4, p. 54] the same equality "is central for the development of quasi-Newton methods, and therefore it has often been called the *quasi-Newton equation*". Also in [1, p. 223] the secant equation appears as a fundamental ingredient in the *definition* of quasi-Newton methods.

In [7,10,8,9,6] it was introduced a new class of algorithms, named $\mathcal{L}$QN, which includes methods of Secant type, in particular the well known BFGS method, and, at the same time, some methods which are not Secant but have relevant good properties (f.i. global convergence). The main purpose consisted in saving the second order information of the matrix B_k, produced by the BFGS method to approximate a full (not sparse) Hessian of f, in a form that allows to reduce the high ($O(n^2)$) computational cost per step of BFGS. More in detail, a substantial generalization of the BFGS scheme has been therein proposed by an updating Hessian approximation formula of the form

$$B_{k+1} = \Phi(\tilde{B}_k, \mathbf{s}_k, \mathbf{y}_k) \tag{1}$$

where $\tilde{B}_k$ is a suitable approximation of B_k and Φ is the BFGS-type rank-two correction of $\tilde{B}$:

$$\Phi(\tilde{B}, \mathbf{s}, \mathbf{y}) := \tilde{B} - \frac{1}{\mathbf{s}^T\tilde{B}\mathbf{s}}\tilde{B}\mathbf{s}\mathbf{s}^T\tilde{B} + \frac{1}{\mathbf{y}^T\mathbf{s}}\mathbf{y}\mathbf{y}^T.$$

The BFGS method is retrieved if $\tilde{B}_k = B_k$ for all k. Moreover, a suitable choice of $\tilde{B}_k$ yields the important class of $\mathcal{L}$QN methods, where the quasi-Newton matrix approximating the Hessian is defined also in terms of a matrix algebra $\mathcal{L}$. The matrices of this algebra $\mathcal{L}$ are simultaneously reduced to diagonal form by a unitary matrix U, i.e. $\mathcal{L} = \operatorname{sd} U = \{L = Ud(\mathbf{z})U^H\}$ where $d(\mathbf{z})$ denotes the diagonal matrix of the eigenvalues z_i of L. In fact, if $\tilde{B}_k$ is the best approximation $\mathcal{L}_{B_k}$ in $\mathcal{L}$ of B_k in Frobenius norm, then from (1) we obtain a simple single-array iteration to compute the eigenvalues of $\mathcal{L}_{B_{k+1}}$ from the eigenvalues of $\mathcal{L}_{B_k}$ [7]. At least two choices are possible for the *new* descent direction $\mathbf{d}_{k+1}$:

$$\mathbf{d}_{k+1} = -B_{k+1}^{-1}\mathbf{g}_{k+1} \quad \text{or} \quad \mathbf{d}_{k+1} = -\tilde{B}_{k+1}^{-1}\mathbf{g}_{k+1}.$$

The first choice yields a Secant (S) algorithm, because $B_{k+1}\mathbf{s}_k = \mathbf{y}_k$, whereas the second choice yields a Non-Secant (NS) procedure, as $\tilde{B}_{k+1}\mathbf{s}_k$ is in general different from $\mathbf{y}_k$.

If U is defined by a fast transform (Fourier, Hartley or others), then in both cases the essential computation can be reduced to exactly two fast transforms at each step k, with a total cost of $O(n \log n)$ FLOPS per step and $O(n)$ memory allocations.

The gain of efficiency with respect to BFGS and its variants is due, essentially, to the simple fact that the matrix $\mathcal{L}_B$ inherits from a matrix B its main spectral properties. In particular we have that

$$B \text{ pd} \quad \Rightarrow \quad \mathcal{L}_B \text{ pd}$$

(pd = real symmetric positive definite; $\mathcal{L}$ spanned by real matrices), and, if $\nu_j(X)$, $j = 1, \dots, n$, denote the eigenvalues of X in non-decreasing order, then [14]

$$\sum_{j=1}^{i} \nu_j(B) \leq \sum_{j=1}^{i} \nu_j(\mathcal{L}_B), \qquad \sum_{j=i}^{n} \nu_j(\mathcal{L}_B) \leq \sum_{j=i}^{n} \nu_j(B).$$

Thus in $\mathcal{L}$QN algorithms the information given by the single array $\mathbf{z}_{B_k}$ of the eigenvalues of $\mathcal{L}_{B_k}$ captures the essential of the second order information that B_k inherits from $\nabla^2 f(\mathbf{x}_k)$. Moreover, a global convergent result has been obtained for NS$\mathcal{L}$QN algorithms [7]. However, one may expect that their secant version S$\mathcal{L}$QN, with more cumbersome formulas but roughly with the same cost per step, are more efficient than NS$\mathcal{L}$QN, even if no proof of convergence of S$\mathcal{L}$QN has been found. In fact, numerical experiments have shown a better efficiency of Secant with respect to Non-Secant $\mathcal{L}$QN procedures, and especially for large scale minimization problems, as in the case of neural networks learning [2] or impulse noise removal from images [3], S$\mathcal{L}$QN methods can be extremely competitive, even with L-BFGS (limited memory BFGS [12]).

In this paper we improve the idea, preliminarily investigated in [6] and [8], to change at each step the *structure* of the pd matrix $\tilde{B}_k$ involved in the Hessian approximation updating formula (1), or, equivalently, the matrix algebra $\mathcal{L} = \mathrm{sd}\, U$ where to choose $\tilde{B}_k$. We do this in a way that appears nearly optimal.

In particular, we are interested in $\mathcal{L}^{(k)}$QN algorithms where, at each step, as soon as the (efficient) secant search direction $\mathbf{d}_{k+1} = -B_{k+1}^{-1}\mathbf{g}_{k+1}$ is computed, a matrix algebra $\mathcal{L}^{(k+1)} = \mathrm{sd}\, U_{k+1}$ and a pd matrix $\tilde{B}_{k+1} \in \mathcal{L}^{(k+1)}$ are chosen so that the Non-Secant search direction $-\tilde{B}_{k+1}^{-1}\mathbf{g}_{k+1}$ achieves the same aim and the same effect of $\mathbf{d}_{k+1}$. So, our $\mathcal{L}^{(k)}$QN algorithms, should satisfy, in principle, the following fundamental conditions:

- At each step of $\mathcal{L}^{(k)}$QN, even if $\tilde{B}_{k+1}$ does not satisfy the secant equation, its inverse $\tilde{B}_{k+1}^{-1}$ produces the same effect of the updated matrix $B_{k+1}^{-1} = \Phi(\tilde{B}_k, \mathbf{s}_k, \mathbf{y}_k)^{-1}$ on the vector $\mathbf{g}_{k+1}$. Precisely, given the secant search direction $\mathbf{d}_{k+1} = -B_{k+1}^{-1}\mathbf{g}_{k+1}$, the matrix $\tilde{B}_{k+1}$ satisfies the equality

$$-\tilde{B}_{k+1}^{-1}\mathbf{g}_{k+1} = \sigma_{k+1}\mathbf{d}_{k+1} \tag{2}$$

for a real $\sigma_{k+1} > 0$.

- $\mathcal{L}^{(k)}$QN methods have a linear computational cost per step, in terms of FLOPS and memory allocations.
- $\mathcal{L}^{(k)}$QN methods are globally convergent.

We will prove that the equality (2) can be obtained if and only if a special condition on the minimal and maximal eigenvalues of $\tilde{B}_{k+1}$ is verified. In particular, if such condition holds, then (2) is obtained by choosing $\tilde{B}_{k+1}$ in a suitable algebra $\mathcal{L}^{(k+1)} = \mathrm{sd}\, U_{k+1}$ where U_{k+1} is defined as the product of two Householder unitary matrices, and the effect of this choice is to reduce to $O(n)$ both computational cost per step and memory allocations. Moreover, the $\mathcal{L}^{(k)}$QN methods obtained by forcing at each step equality (2) turn out to be globally convergent, and this is not surprising since the sequence of approximations $\{\mathbf{x}_k\}_{k\in\mathbb{N}}$ they yield can be seen as produced by the corresponding Non-Secant $\mathcal{L}^{(k)}$QN methods defined in terms of the search directions $-\tilde{B}_{k+1}^{-1}\mathbf{g}_{k+1}$ (it is known that Non-Secant $\mathcal{L}^{(k)}$QN methods are globally convergent [7,8]).

In conclusion, with the $\mathcal{L}^{(k)}$QN methods introduced in this paper, it is solved implicitly the degree of freedom, and thus of uncertainty, in choosing the space $\mathcal{L}$ of the $\mathcal{L}$QN algorithms, and, simultaneously, it is nullified the difference between the classes of Secant and Non-Secant algorithms since, eventually, the search directions produced by the new $\mathcal{L}^{(k)}$QN methods are simultaneously of Secant and of Non-Secant type.

2. Notation and preliminaries

We will freely use familiar properties of symmetric positive definite matrices and fundamental results concerning algebras of matrices simultaneously diagonalized by a given unitary transform.

We use the shorthand pd to denote a real symmetric positive definite matrix. Given a vector $\mathbf{z} \in \mathbb{R}^n$ we write $\mathbf{z} > 0$ to denote entrywise positivity. We let $d(\mathbf{z})$ be the diagonal matrix whose diagonal entries are the components of $\mathbf{z}$, analogously the symbol $\mathrm{diag}(x_i,\, i = 1, \dots, n)$ denotes the diagonal matrix whose diagonal entries are the x_i, writing briefly $\mathrm{diag}(x_i)$ when no ambiguity may occur. Thus for instance $d(\mathbf{z}) = \mathrm{diag}(z_i)$.

Let $M_n(\mathbb{C})$ be the set of all $n \times n$ matrices with complex entries. Given a unitary matrix $U \in M_n(\mathbb{C})$ (i.e. $U^H = U^{-1}$), set

$$\mathcal{L} := \mathrm{sd}\, U = \left\{ U d(\mathbf{z}) U^H \; : \; \mathbf{z} \in \mathbb{C}^n \right\}$$

The space $\mathcal{L}$ is a closed subspace of $M_n(\mathbb{C})$ which is a Hilbert space with respect to the inner product $(X, Y) = \sum_{i,j=1}^n \bar{x}_{ij} y_{ij}$. Note that the norm induced by $(\cdot,\cdot)$ is the Frobenius norm $\|X\|_F = (\sum_{i,j=1}^n |x_{ij}|^2)^{\frac{1}{2}}$. Thus, by the Hilbert projection theorem, given a matrix $B \in M_n(\mathbb{C})$ there exists a unique element $\mathcal{L}_B \in \mathcal{L}$ such that

$$\|\mathcal{L}_B - B\|_F \le \|X - B\|_F, \quad \forall X \in \mathcal{L}, \tag{3}$$

or, equivalently, such that

$$(X, B - \mathcal{L}_B) = 0, \quad \forall X \in \mathcal{L}.$$

For the sake of completeness we recall hereafter few important results on pd matrices and on the projection $\mathcal{L}_B$. For further details see [15,11,7].

Lemma 1 *(Kantorovich inequality). Let A be a hermitian positive definite matrix and $\mathbf{z} \in \mathbb{C}^n$. If $\lambda_{\max}$ and $\lambda_{\min}$ denote the maximum and the minimum eigenvalue of A, respectively, then*

$$1 \leq \frac{(\mathbf{z}^H A\mathbf{z})(\mathbf{z}^H A^{-1}\mathbf{z})}{(\mathbf{z}^H \mathbf{z})^2} \leq \frac{(\lambda_{\max} + \lambda_{\min})^2}{4\lambda_{\max}\lambda_{\min}} = \frac{(1+\mu(A))^2}{4\mu(A)} \tag{4}$$

where $\mu(A) = \lambda_{\max}/\lambda_{\min}$ is the condition number of A in the spectral norm.

Lemma 2.

1. *$\mathcal{L}_B = Ud(\mathbf{z}_B)U^H$ where $[\mathbf{z}_B]_i = [U^H BU]_{ii}$, $i = 1, \ldots, n$; in particular $\mathbf{z}_{\mathbf{x}\mathbf{y}^T} = d(U^H\mathbf{x})U^T\mathbf{y}$, where $\mathbf{x}, \mathbf{y} \in \mathbb{C}^n$.*
2. *If $B = B^H$, then $\mathcal{L}_B = \mathcal{L}_B^H$ and $\min \nu(B) \leq \nu(\mathcal{L}_B) \leq \max \nu(B)$ where $\nu(X)$ denotes the generic eigenvalue of X. Therefore $\mathcal{L}_B$ is hermitian positive definite whenever B is hermitian positive definite.*
3. *If $B \in \mathbb{R}^{n\times n}$ then $\mathcal{L}_B \in \mathbb{R}^{n\times n}$ provided that $\mathcal{L}$ is spanned by real matrices or more generally whenever $\overline{\mathcal{L}} \subset \mathcal{L}$ (i.e. $\mathcal{L}$ is closed under conjugation).*

Lemma 3. *Let $\mathcal{L} = \mathrm{sd}\, U$ and let $B \in M_n(\mathbb{C})$. Then*

1. $\mathrm{tr}(\mathcal{L}_B) = \mathrm{tr}(B)$
2. $\det(B) \leq \det(\mathcal{L}_B)$.

Proof. Use Lemma 2 and Hadamard's inequality for the determinant of a matrix. □

For a more exhaustive treatment of the contents of Lemma 2 and Lemma 3, and their relevance for $\mathcal{L}$QN minimizations algorithms and optimal preconditioning, one can see [11] and [7].

3. The Secant scheme

Let $f : \mathbb{R}^n \to \mathbb{R}$ and consider the minimum problem

$$\text{find } \mathbf{x}_* \text{ such that } f(\mathbf{x}_*) = \min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}). \tag{5}$$

In [7] it is proposed a Secant BFGS-type algorithm for the solution of (5), whose main instructions are summarized in Algorithm 3.1 here below:

Algorithm 3.1: Secant BFGS-type.

```
Data: $\mathbf{x}_0 \in \mathbb{R}^n$, $\mathbf{g}_0 = \nabla f(\mathbf{x}_0)$, $\tilde{B}_0$ pd, $\mathbf{d}_0 \in \mathbb{R}^n$ s.t. $\mathbf{d}_0^T \mathbf{g}_0 < 0$;
while $\mathbf{g}_k \neq 0$ do
    $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$;                    /* $\lambda_k$ verifies (36) */
    $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$, $\mathbf{g}_{k+1} = \nabla f(\mathbf{x}_{k+1})$;
    $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$;
    $B_{k+1} = \Phi(\tilde{B}_k, \mathbf{s}_k, \mathbf{y}_k)$;
    $\mathbf{d}_{k+1} = -B_{k+1}^{-1} \mathbf{g}_{k+1}$;
    Construct $\tilde{B}_{k+1}$ pd;
```

where

- $\tilde{B}_k$ is chosen for all k as a suitable, step-by-step approximation of B_k, such that, if B_k is pd, then $\tilde{B}_k$ is pd;
- $\Phi(\tilde{B}, \mathbf{s}, \mathbf{y}) = \tilde{B} + \frac{1}{\mathbf{y}^T\mathbf{s}}\mathbf{y}\mathbf{y}^T - \frac{1}{\mathbf{s}^T\tilde{B}\mathbf{s}}\tilde{B}\mathbf{s}\mathbf{s}^T\tilde{B}$ is a Hessian approximation BFGS-type updating formula.

Observe that the matrix

$$B_{k+1} = \Phi(\tilde{B}_k, \mathbf{s}_k, \mathbf{y}_k)$$

verifies the *secant equation* $B_{k+1}\mathbf{s}_k = \mathbf{y}_k$. For this reason we refer to Algorithm 3.1 as *Secant* BFGS-type. For $\tilde{B}_k = B_k$ one has the BFGS method, which is a well known secant quasi-Newton minimization algorithm [5,12].

Moreover, observe that, by the Sherman–Morrison–Woodbury formula, we have

$$B_{k+1}^{-1} = \Psi\left(\tilde{B}_k^{-1}, \mathbf{s}_k, \mathbf{y}_k\right) \tag{6}$$

where

$$\Psi(\tilde{H}, \mathbf{s}, \mathbf{y}) = \left(I - \frac{1}{\mathbf{y}^T\mathbf{s}}\mathbf{y}\mathbf{s}^T\right)^T \tilde{H} \left(I - \frac{1}{\mathbf{y}^T\mathbf{s}}\mathbf{y}\mathbf{s}^T\right) + \frac{1}{\mathbf{y}^T\mathbf{s}}\mathbf{s}\mathbf{s}^T. \tag{7}$$

Identities (6), (7) assure that the new search direction

$$\mathbf{d}_{k+1} = -B_{k+1}^{-1}\mathbf{g}_{k+1}$$

can be computed with at most $O(n^2)$ FLOPS (the cost of computing a general matrix-vector product $\tilde{B}_k^{-1}\mathbf{z}$, $\mathbf{z} \in \mathbb{R}^n$), which turns out to be an estimation of the total computational cost per step (as it will be clear afterwards, we assume that less than $O(n^2)$ FLOPS are sufficient to compute the matrix $\tilde{B}_k^{-1}$ from B_k).

An important property of the Hessian approximation updating formula Φ, used in the definition of the Secant BFGS-type Algorithm 3.1, is that, assuming $\tilde{B}$ pd, the matrix $\Phi(\tilde{B}, \mathbf{s}, \mathbf{y})$ is a well defined pd matrix iff $\mathbf{y}^T\mathbf{s} > 0$. If f is continuously differentiable and lower bounded, whenever the positive parameter λ_k is chosen, at each step, so that the two Armijo–Goldstein conditions are satisfied – briefly $\lambda_k \in AG$ – we have that the value of $f(\mathbf{x}_{k+1})$ is less than $f(\mathbf{x}_k) - \eta_k$, with $\eta_k > 0$, and $\mathbf{y}_k^T\mathbf{s}_k > 0$ (see Section 5 for a more precise definition of η_k). Thus B_{k+1} is pd and Secant BFGS-type yields a well defined, strictly decreasing sequence $\{f(\mathbf{x}_k)\}_{k\in\mathbb{N}}$.

Of course, we have an analogous result if, in the above Algorithm 3.1, the matrix $\tilde{B}_{k+1}$ is constructed immediately after the definition of B_{k+1}, and the secant search direction $\mathbf{d}_{k+1} = -B_{k+1}^{-1}\mathbf{g}_{k+1}$ is replaced by the alternative

$$\mathbf{d}_{k+1} = -\tilde{B}_{k+1}^{-1}\mathbf{g}_{k+1}, \tag{8}$$

which may be called *Non-Secant search direction.*

Remark 1. Note that any pd matrix $\tilde{B}_k$ we use in BFGS-type algorithms has the structure

$$\tilde{B}_k = U_k d(\mathbf{z}_k) U_k^H, \quad U_k \text{ unitary}, \ [\mathbf{z}_k]_i > 0.$$

If for each step the eigenvalues of $\tilde{B}_k$ – i.e. the $[\mathbf{z}_k]_i$ of Remark 1 – are such that

$$\det(B_k) \le \det(\tilde{B}_k) = \prod_i [\mathbf{z}_k]_i, \qquad \operatorname{tr}(B_k) \ge \operatorname{tr}(\tilde{B}_k) = \sum_i [\mathbf{z}_k]_i, \tag{9}$$

then the NS BFGS-type algorithms (where $\mathbf{d}_{k+1}$ is defined as in (8)) are convergent [7] without any assumption on the matrix which diagonalizes $\tilde{B}_k$ – i.e. on the U_k of Remark 1 –. In particular, it is easy to check that such conditions (9) are satisfied when $[\mathbf{z}_k]_i = (U_k^H B_k U_k)_{ii}$, $i = 1, \dots, n$ (Hadamard inequality is used for the first of (9)). But these $[\mathbf{z}_k]_i$ are nothing else than the eigenvalues of the best approximation in Frobenius norm of B_k in the space

$$\mathcal{L}^{(k)} := \operatorname{sd} U_k = \left\{ U_k d(\mathbf{z}) U_k^H \ : \ \mathbf{z} \in \mathbb{C}^n \right\}.$$

In [7,8] this approximation of B_k is denoted by $\mathcal{L}_{B_k}^{(k)}$, so the choice $[\mathbf{z}_k]_i = (U_k^H B_k U_k)_{ii}$ corresponds to the choice $\tilde{B}_k = \mathcal{L}_{B_k}^{(k)}$.

On the other hand numerical experiments, performed mainly with $\mathcal{L}^{(k)}$ equal to a fixed $\mathcal{L}$ for all k, show that the NS$\mathcal{L}^{(k)}$QN algorithms, defined by the search direction $\mathbf{d}_{k+1} = -(\mathcal{L}_{B_{k+1}}^{(k+1)})^{-1}\mathbf{g}_{k+1}$, have a slow convergence rate, whereas the S$\mathcal{L}^{(k)}$QN algorithms, with $\mathbf{d}_{k+1} = \Phi(\mathcal{L}_{B_k}^{(k)}, \mathbf{s}_k, \mathbf{y}_k)^{-1}\mathbf{g}_{k+1}$, appear competitive, even with methods like L-BFGS [2].

Now the crucial question is the following: *is it possible to design a BFGS-type algorithm with minimum cost per step which combines the convergence of Non-Secant with the efficiency of the Secant* $\mathcal{L}^{(k)}$QN *methods?* We will see that this approach, where the adaptive character of the $\mathcal{L}^{(k)}$QN algorithm is exploited, will yield a Non-Secant algorithm with the same efficiency of a Secant one in solving minimization problems or, equivalently, to a Secant algorithm which is convergent as a Non-Secant one.

In particular, one could impose directly that the NS algorithm yields the *same* search direction of the S one. In this case $\tilde{B}_{k+1}$ should be chosen such that

$$\tilde{B}_{k+1}^{-1}\mathbf{g}_{k+1} = \sigma_{k+1}B_{k+1}^{-1}\mathbf{g}_{k+1}, \qquad \tilde{B}_{k+1} = U_{k+1}d(\mathbf{z}_{k+1})U_{k+1}^H, \quad \sigma_{k+1} > 0. \tag{10}$$

Notice that (10) is equivalent to say that

$$B_{k+1}\mathbf{s}_{k+1} = \sigma_{k+1}\tilde{B}_{k+1}\mathbf{s}_{k+1}, \tag{11}$$

that is B_{k+1} and $\tilde{B}_{k+1}$ act as the same operator on the vector $\mathbf{s}_{k+1}$.

Remark 2. Every Secant BFGS-type algorithm satisfying (9) and (10) (or (11)), turns out to be at least convergent (see Section 5). Recall that any convergent quasi-Newton method $\mathbf{x}_{k+2} = \mathbf{x}_{k+1} - \lambda_{k+1}A_{k+1}^{-1}\mathbf{g}_{k+1}$, with A_{k+1} pd and satisfying the Dennis–Moré condition

$$\lim_{k\to+\infty}\frac{\|(A_{k+1} - \nabla^2 f(\mathbf{x}_*))\mathbf{s}_{k+1}\|}{\|\mathbf{s}_{k+1}\|} = 0, \tag{12}$$

has a superlinear rate of convergence. Thus, in our context, assuming that U_{k+1} and $\mathbf{z}_{k+1}$ solving (9) and (10) (or (11)) depend on a set of free parameters, one may try, at least in principle, to use these parameters to impose (12) with $\sigma_{k+1}\tilde{B}_{k+1}$ replacing $A_{k+1} = B_{k+1}$, in order to obtain a secant BFGS-type *superlinearly* convergent method.

In the following we investigate in two different cases if (10) or (11) can be effectively verified (note that in both cases the conditions (9) automatically hold):

- $\mathcal{L}^{(k)}$QN
  The matrix $\tilde{B}_{k+1}$ is the best approximation of B_{k+1} in $\mathcal{L}^{(k+1)}$, i.e.

$$\tilde{B}_{k+1} = \mathcal{L}_{B_{k+1}}^{(k+1)} = U_{k+1}\,\mathrm{diag}\big((U_{k+1}^H B_{k+1}U_{k+1})_{ii}\big)U_{k+1}^H. \tag{13}$$

- Hybrid $\mathcal{L}^{(k)}$QN
  The matrix $\tilde{B}_{k+1}$ is chosen in $\mathcal{L}^{(k+1)}$ as follows:

$$\tilde{B}_{k+1} = U_{k+1}\,\mathrm{diag}\big((V_{k+1}^H B_{k+1}V_{k+1})_{ii}\big)U_{k+1}^H, \tag{14}$$

  where V_{k+1} is an arbitrary unitary matrix; in other words $[\mathbf{z}_{k+1}]_i = (V_{k+1}^H B_{k+1}V_{k+1})_{ii}$ are free and are not forced to be the eigenvalues of $\mathcal{L}_{B_{k+1}}^{(k+1)}$.

Concerning the first choice of $\tilde{B}_{k+1}$, one could formulate the problem of calculating U_{k+1} as follows:

Problem 1 *(Totally Non-Linear Problem (TNLP)).*

$$\begin{aligned}&\text{Given } \mathbf{g}_{k+1} \in \mathbb{R}^n,\ B_{k+1} = \Phi(\tilde{B}_k, \mathbf{s}_k, \mathbf{y}_k) \text{ pd and } \mathbf{d}_{k+1} = -B_{k+1}^{-1}\mathbf{g}_{k+1}\\ &\text{find a unitary } U_{k+1} \in \mathbb{R}^{n\times n},\ \text{such that, if } \mathcal{L}^{(k+1)} = \operatorname{sd} U_{k+1},\\ &\text{then } -\big[\mathcal{L}^{(k+1)}_{B_{k+1}}\big]^{-1}\mathbf{g}_{k+1} = \sigma_{k+1}\mathbf{d}_{k+1} \text{ for some } \sigma_{k+1} > 0.\end{aligned}$$

For the sake of clarity, let us remember, once more, that $\mathcal{L}^{(k+1)}_{B_{k+1}} = U_{k+1} d(\mathbf{z}^{(k+1)}_{B_{k+1}}) U^H_{k+1}$ where

$$\big[\mathbf{z}^{(k+1)}_{B_{k+1}}\big]_i = \big(U^H_{k+1} B_{k+1} U_{k+1}\big)_{ii}, \quad i = 1, \ldots, n. \tag{15}$$

If for each step Problem 1 has a solution, then the following $\mathcal{L}^{(k)}$QN method (Algorithm 3.1 with $\tilde{B}_k = \mathcal{L}^{(k)}_{B_k}$ for all k) is well defined and turns out to be globally convergent (see Theorem 3 and Remark 5 in Section 5).

Algorithm 3.2: $\mathcal{L}^{(k)}$QN.

```
Data: $\mathbf{x}_0 \in \mathbb{R}^n$, $\mathbf{g}_0 = \nabla f(\mathbf{x}_0)$, $B_0$ pd, $\mathbf{d}_0 \in \mathbb{R}^n$ s.t. $\mathbf{d}_0^T \mathbf{g}_0 < 0$, define $\mathcal{L}^{(0)}$;
while $\mathbf{g}_k \neq 0$ do
    $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$;                /* $\lambda_k$ verifies (36) */
    $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$, $\mathbf{g}_{k+1} = \nabla f(\mathbf{x}_{k+1})$;
    $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$;
    $B_{k+1} = \Phi(\mathcal{L}^{(k)}_{B_k}, \mathbf{s}_k, \mathbf{y}_k)$;
    $\mathbf{d}_{k+1} = -B_{k+1}^{-1}\mathbf{g}_{k+1}$;
    Construct $\mathcal{L}^{(k+1)}$ s.t. $-[\mathcal{L}^{(k+1)}_{B_{k+1}}]^{-1}\mathbf{g}_{k+1} = \sigma_{k+1}\mathbf{d}_{k+1}$, $\sigma_{k+1} > 0$;
```

Observe, however, that the dependence of the vector $\mathbf{z}^{(k+1)}_{B_{k+1}}$ in identity (15) from the unknown operator U_{k+1}, gives rise to a four degree non-linear problem for each entry of the matrix U_{k+1} we are looking for. At the moment, no low complexity solution has been found for Problem 1 (we don't know even if such solution exists) and thus Algorithm 3.2 has a theoretical interest.[1]

So, in the next section we shall deal with the second case (hybrid $\mathcal{L}^{(k)}$QN), i.e. we will calculate suitable matrices U_{k+1} and V_{k+1} to define $\tilde{B}_{k+1}$ as in (14) and satisfying (10) (or (11)).

4. Existence of solution of PNLP

In this section, for the sake of simplicity, the index $k+1$ will be dropped. We now give necessary and sufficient conditions for the existence of a solution of the following

[1] Obviously Problem 1 is solved by the space $\mathcal{L}^{(k+1)}$ such that $\mathcal{L}^{(k+1)}_{B_{k+1}} = B_{k+1}$, but such $\mathcal{L}^{(k+1)}$ is in general not of low complexity and, of course, is not cheaply computable.

Problem 2 *(Partially Non-Linear Problem (PNLP)).*

$$\begin{aligned}&\text{Given } \mathbf{g} \in \mathbb{R}^n, \mathbf{d} = -B^{-1}\mathbf{g}, \text{ where } B \text{ pd } \left(\Rightarrow \mathbf{g}^T\mathbf{d} < 0\right),\\ &\text{and } \mathbf{z} \in \mathbb{R}^n,\ z_i > 0\ i = 1, \ldots, n, \text{ find a unitary } U \in \mathbb{R}^{n\times n}, \text{ such that}\\ &-\left[Ud(\mathbf{z})U^H\right]^{-1}\mathbf{g} = \sigma\mathbf{d},\ \sigma \in \mathbb{R}.\end{aligned} \tag{16}$$

Given $\mathbf{z} > 0$ let us write:

$$z_m = \min_{i=1,\ldots,n} z_i, \qquad z_M = \max_{i=1,\ldots,n} z_i.$$

Lemma 4. *If there exists a unitary matrix U solution of Problem 2 then:*

$$\sigma = \frac{\|\mathbf{g}\|}{\|d(\mathbf{z})U^H\mathbf{d}\|} = \frac{\mathbf{d}^T(-\mathbf{g})}{\mathbf{d}^TUd(\mathbf{z})U^H\mathbf{d}}, \tag{17}$$

$$\frac{(\mathbf{d}^T(-\mathbf{g}))^2}{\|\mathbf{d}\|^2\|\mathbf{g}\|^2} \in \left[\frac{4z_mz_M}{(z_m+z_M)^2}, 1\right]. \tag{18}$$

Proof. To prove (17) observe that from Problem 2 we have

$$\mathbf{g} = -\sigma Ud(\mathbf{z})U^H\mathbf{d} \quad \Rightarrow \quad \mathbf{d}^T\mathbf{g} = -\sigma\mathbf{d}^TUd(\mathbf{z})U^H\mathbf{d}. \tag{19}$$

As $\mathbf{d}^T\mathbf{g} < 0$ and $\mathbf{d}^TUd(\mathbf{z})U^H\mathbf{d} > 0$, we obtain $\sigma > 0$. Moreover, being U unitary, we obtain

$$\|\mathbf{g}\| = \sigma\left\|Ud(\mathbf{z})U^H\mathbf{d}\right\| = \sigma\left\|d(\mathbf{z})U^H\mathbf{d}\right\|,$$

thus (17). Regarding (18), observe that from (17) and (19) we have:

$$\frac{(\mathbf{d}^T(-\mathbf{g}))^2}{\|\mathbf{d}\|^2\|\mathbf{g}\|^2} = \frac{(\mathbf{d}^TUd(\mathbf{z})U^H\mathbf{d})^2}{(\mathbf{d}^TUd(\mathbf{z})^2U^H\mathbf{d})(\mathbf{d}^T\mathbf{d})}. \tag{20}$$

Setting

$$\mathbf{y} = d(\mathbf{z})^{\frac{1}{2}}U^H\mathbf{d} \quad \left(\Rightarrow \mathbf{d} = Ud(\mathbf{z})^{-\frac{1}{2}}\mathbf{y}\right)$$

we obtain

$$\frac{(\mathbf{d}^T(-\mathbf{g}))^2}{\|\mathbf{d}\|^2\|\mathbf{g}\|^2} = \frac{(\mathbf{y}^T\mathbf{y})^2}{(\mathbf{y}^Td(\mathbf{z})\mathbf{y})(\mathbf{y}^Td(\mathbf{z})^{-1}\mathbf{y})} \in \left[\frac{4z_mz_M}{(z_m+z_M)^2}, 1\right] \tag{21}$$

where the inclusion statement follows from the Kantorovich inequality. □

Lemma 5. *Given* $\mathbf{z} > 0$ *and a real number* c *such that* $0 < c^2 \leq 1$*, if there exists a pair of indexes* (h, k) *such that*

$$\frac{4z_h z_k}{(z_h + z_k)^2} \leq c^2$$

then there exists a vector $\mathbf{y} \in \mathbb{R}^n$ *such that*

$$\frac{(\mathbf{y}^T\mathbf{y})^2}{(\mathbf{y}^T d(\mathbf{z})\mathbf{y})(\mathbf{y}^T d(\mathbf{z})^{-1}\mathbf{y})} = c^2. \tag{22}$$

Proof. Set $\mathbf{y} = \alpha\mathbf{e}_h + \beta\mathbf{e}_k$. We will show that there exists $\alpha, \beta \in \mathbb{R}$ for which identity (22) holds. Substituting the expression for $\mathbf{y}$, and requiring that $\|\mathbf{y}\| = 1$ we obtain:

$$\begin{cases} \alpha^2 + \beta^2 = 1 \\ \alpha^4 + \beta^4 + \alpha^2\beta^2 m_{hk} = c^{-2} \end{cases} \tag{23}$$

where

$$m_{hk} = \frac{z_h}{z_k} + \frac{z_k}{z_h} = \frac{z_h^2 + z_k^2}{z_h z_k}.$$

From the first row of (23) we get $\beta^2 = 1 - \alpha^2$ and substituting in the second one:

$$\alpha^4(2 - m_{hk}) - \alpha^2(2 - m_{hk}) + 1 - c^{-2} = 0.$$

Setting $t = \alpha^2$ and solving we obtain:

$$t_\pm = \frac{1}{2} \pm \sqrt{\Delta}, \qquad \Delta = \frac{1}{4} - \frac{c^{-2} - 1}{m_{hk} - 2}. \tag{24}$$

Thus a real $t_\pm$ exists only if $\frac{1}{4} \geq \frac{c^{-2}-1}{m_{hk}-2}$, i.e.

$$\frac{(z_h + z_k)^2}{4z_h z_k} \geq c^{-2}$$

which is true by hypothesis. So $\Delta \geq 0$ and, moreover, $\sqrt{\Delta} \leq \frac{1}{2}$. We conclude that $t_\pm \in [0, 1]$ ($t_\pm \in (0, 1)$ if $c^2 < 1$) and the solutions of the system (23) are:

$$\alpha^2 = t_+, \quad \beta^2 = 1 - t_+ \quad \text{or} \quad \alpha^2 = 1 - t_+, \quad \beta^2 = t_+. \qquad \square$$

Theorem 1. *If the* z_i *are such that*

$$\frac{4z_m z_M}{(z_m + z_M)^2} \leq \frac{(\mathbf{d}^T(-\mathbf{g}))^2}{\|\mathbf{d}\|^2\|\mathbf{g}\|^2}, \tag{25}$$

then there exists a unitary matrix U *solution of Problem 2. In particular,* U *can be effectively constructed in* $O(n)$ *FLOPS as the product of two Householder matrices.*

Proof. Solution will be built explicitly. To this end set

$$U = H(\mathbf{w})H(\mathbf{v}),$$

where $H(\mathbf{x}) = (I - \frac{2}{\|\mathbf{x}\|^2}\mathbf{x}\mathbf{x}^H) = (I - \bar{\mathbf{x}}\bar{\mathbf{x}}^H)$ with $\|\bar{\mathbf{x}}\| = \sqrt{2}$. Substituting the expression of U in Problem 2 we obtain:

$$\left[\left(\mathbf{v}^H H(\mathbf{w})\mathbf{g}\right)I + \sigma\left(\mathbf{v}^H H(\mathbf{w})\mathbf{d}\right)d(\mathbf{z})\right]\mathbf{v} = H(\mathbf{w})\mathbf{g} + \sigma d(\mathbf{z})H(\mathbf{w})\mathbf{d} \tag{26}$$

where the unknowns are the vectors $\mathbf{w}$, $\mathbf{v}$. Suppose to choose $\mathbf{w}$ such that

$$\mathbf{v}^H H(\mathbf{w})\mathbf{d} = 0, \quad \|\mathbf{w}\| = \sqrt{2} \tag{27}$$

(we will show later that this choice is possible). The identity (26) becomes:

$$\left(\mathbf{v}^H H(\mathbf{w})\mathbf{g}\right)\mathbf{v} = H(\mathbf{w})\mathbf{g} + \sigma d(\mathbf{z})H(\mathbf{w})\mathbf{d}. \tag{28}$$

Let us search a solution of the form:

$$\mathbf{v} = \alpha\left(H(\mathbf{w})\mathbf{g} + \sigma d(\mathbf{z})H(\mathbf{w})\mathbf{d}\right), \quad \|\mathbf{v}\| = \sqrt{2}. \tag{29}$$

Using the identity

$$\mathbf{v}^H H(\mathbf{w})\mathbf{g} = \alpha\left(\mathbf{g}^H\mathbf{g} + \sigma\mathbf{d}^H H(\mathbf{w})d(\mathbf{z})H(\mathbf{w})\mathbf{g}\right),$$

by (28) we obtain

$$\alpha^2 = \frac{1}{\|\mathbf{g}\|^2 + \sigma\mathbf{d}^H H(\mathbf{w})d(\mathbf{z})H(\mathbf{w})\mathbf{g}} \tag{30}$$

and, forcing $\|\mathbf{v}\| = \sqrt{2}$,

$$\sigma^2 = \frac{\|\mathbf{g}\|^2}{\mathbf{d}^H H(\mathbf{w})d(\mathbf{z})^2 H(\mathbf{w})\mathbf{d}}. \tag{31}$$

The identity (27) becomes, substituting in $\mathbf{v}$ the expressions (31) found for σ:

$$\mathbf{d}^H\mathbf{g} + \frac{\|\mathbf{g}\|\mathbf{d}^H H(\mathbf{w})d(\mathbf{z})H(\mathbf{w})\mathbf{d}}{\sqrt{\mathbf{d}^H H(\mathbf{w})d(\mathbf{z})^2 H(\mathbf{w})\mathbf{d}}} = 0, \tag{32}$$

from which

$$\left(\frac{\mathbf{d}^H(-\mathbf{g})}{\|\mathbf{g}\|\|\mathbf{d}\|}\right)^2 = \frac{(\mathbf{d}^H H(\mathbf{w})d(\mathbf{z})H(\mathbf{w})\mathbf{d})^2}{(\mathbf{d}^H H(\mathbf{w})d(\mathbf{z})^2 H(\mathbf{w})\mathbf{d})(\mathbf{d}^H\mathbf{d})}. \tag{33}$$

By setting

$$\mathbf{y} = d(\mathbf{z})^{\frac{1}{2}} H(\mathbf{w})\mathbf{d} \quad \left(\Rightarrow \mathbf{d} = H(\mathbf{w}) d(\mathbf{z})^{-\frac{1}{2}}\mathbf{y}\right) \tag{34}$$

in the right hand side of the above identity (33), we obtain

$$\left(\frac{\mathbf{d}^H(-\mathbf{g})}{\|\mathbf{g}\|\|\mathbf{d}\|}\right)^2 = \frac{(\mathbf{y}^H\mathbf{y})^2}{(\mathbf{y}^H d(\mathbf{z})\mathbf{y})(\mathbf{y}^H d(\mathbf{z})^{-1}\mathbf{y})}. \tag{35}$$

Since

$$\frac{(\mathbf{y}^H\mathbf{y})^2}{(\mathbf{y}^H d(\mathbf{z})\mathbf{y})(\mathbf{y}^H d(\mathbf{z})^{-1}\mathbf{y})} \in \left[\frac{4z_m z_M}{(z_m + z_M)^2}, 1\right],$$

it is possible to find $\mathbf{y}$ such that equality (35) holds only if (25) holds. But (25) is also sufficient for the existence of $\mathbf{y}$ solving (35). In fact due to Lemma 5 we know that there exists, and it is easy to compute, a vector $\mathbf{y}$ such that

$$\left(\frac{\mathbf{d}^H(-\mathbf{g})}{\|\mathbf{g}\|\|\mathbf{d}\|}\right)^2 = \frac{(\mathbf{y}^H\mathbf{y})^2}{(\mathbf{y}^H d(\mathbf{z})\mathbf{y})(\mathbf{y}^H d(\mathbf{z})^{-1}\mathbf{y})}.$$

Observe moreover that if $\mathbf{y}$ satisfies the above identity, then $k\mathbf{y}$ satisfies the above identity for all $k \in \mathbb{R}$, so it is possible to choose $\mathbf{y}$ such that

$$\left\|d(\mathbf{z})^{-\frac{1}{2}}\mathbf{y}\right\| = \|\mathbf{d}\|.$$

This assures that (34) has a solution $\mathbf{w}$, precisely given by

$$\mathbf{w} = \frac{\sqrt{2}}{\|\mathbf{d} - d(\mathbf{z})^{-\frac{1}{2}}\mathbf{y}\|}\left[\mathbf{d} - d(\mathbf{z})^{-\frac{1}{2}}\mathbf{y}\right]$$

(see [8] where it is displayed the Householder transform mapping a vector into another one of the same norm). We have proved the existence of a matrix $H(\mathbf{w})$ which satisfies condition (27), and thus the existence of a matrix $U = H(\mathbf{w})H(\mathbf{v})$ solution of Problem 2. □

Using the above results the following theorem can be stated

Theorem 2. *Given $\mathbf{z} > 0$ and $\mathbf{d}, \mathbf{g}, \in \mathbb{R}^n$ such that $\mathbf{d}^T\mathbf{g} < 0$, the inequality (25) is a necessary and sufficient condition for the existence of a solution of Problem 2.*

Remark 3 *(On Totally Non-Linear Problem).* Given the pd matrix B ($B = B_{k+1} = \Phi(\tilde{B}_k, \mathbf{s}_k, \mathbf{y}_k)$) and $\mathbf{d} = -B^{-1}\mathbf{g}$, suppose there exists a unitary matrix U such that, if $\mathcal{L} = \operatorname{sd} U$, then

$$-\mathcal{L}_B^{-1}\mathbf{g} = \sigma\mathbf{d}, \quad \sigma > 0,$$

i.e. U is a solution of Problem 1. Reasoning exactly in the same way of Lemma 4, and setting

$$[\mathbf{z}_B]_i = \left(U^H BU\right)_{ii}, \quad i \in \{1, \dots, n\},$$

we obtain:

$$\sigma = \frac{\|\mathbf{g}\|}{\|d(\mathbf{z}_B)U^H\mathbf{d}\|},$$
$$\frac{(\mathbf{d}^T(-\mathbf{g}))^2}{\|\mathbf{d}\|^2\|\mathbf{g}\|^2} \in \left[\frac{4[\mathbf{z}_B]_m[\mathbf{z}_B]_M}{([\mathbf{z}_B]_m + [\mathbf{z}_B]_M)^2}, 1\right].$$

It follows that, in some sense, the PNLP (Problem 2) mimics quite closely the TNLP (Problem 1).

Remark 4. Note that we have written a solution of Problem 2 as a product of two Householder matrices. As a matter of fact, it is not yet clear if one Householder matrix is sufficient to solve Problem 2. It is easy to prove that Problem 2 is solvable by $U = H(\mathbf{w})$ (i.e. with one Householder matrix) at least for $n = 2$.

5. Convergence analysis

The known result of Powell [13] on the global convergence of the BFGS method and the known result [7] of global convergence of NS BFGS-type algorithms are now extended to the Secant BFGS-type Algorithms 3.1, by adding few simple hypotheses on $\tilde{B}_k$. We first recall the following

Proposition 1. *(See [5].) Let $\tilde{B}_k$ be a pd $n \times n$ matrix and let $\mathbf{s}_k, \mathbf{y}_k \in \mathbb{R}^n$. Then the matrix $\Phi(\tilde{B}_k, \mathbf{s}_k, \mathbf{y}_k)$ is a well defined pd matrix iff $\mathbf{y}_k^T\mathbf{s}_k > 0$.*

By Proposition 1 it is possible to state that, if the positive parameters λ_k are properly chosen, then Algorithm 3.1 yields a well defined and strictly decreasing sequence $\{f(\mathbf{x}_k)\}_{k\in\mathbb{N}}$. In particular, for a continuously differentiable and lower bounded function f, such a sequence is obtained if the step length λ_k satisfies the Armijo–Goldstein (AG) prescriptions (see [5]), that is, λ_k belongs to the set Λ_k defined here below.

Definition 1. Fix two constants c_1, c_2, $0 < c_1 < c_2 < 1$, and set $\chi_k(\lambda) = f(\mathbf{x}_k + \lambda\mathbf{d}_k)$. Then the AG set Λ_k is the set of all $\lambda \in \mathbb{R}^+$ such that

$$\begin{cases} \chi_k(\lambda) \le \chi_k(0) + \lambda c_1 \chi'_k(0), \\ \chi'_k(\lambda) \ge c_2 \chi'_k(0). \end{cases} \tag{36}$$

In fact, since $\mathbf{d}_k$ is a descent direction in $\mathbf{x}_k$ ($\chi'_k(0) = \mathbf{g}_k^T\mathbf{d}_k < 0$), the set Λ_k is nonempty, and the choice $\lambda_k \in \Lambda_k$ yields the inequalities $f(\mathbf{x}_{k+1}) \le f(\mathbf{x}_k) - \eta_k < f(\mathbf{x}_k)$,

$\eta_k = -\lambda_k c_1 \chi_k'(0) > 0$, and $\mathbf{s}_k^T \mathbf{y}_k = \lambda_k(\chi_k'(\lambda) - \chi_k'(0)) \geq \lambda_k(c_2 - 1)\chi_k'(0) > 0$. So, by Proposition 1, B_{k+1} in Algorithm 3.1 is a well defined pd matrix and

$$\chi_{k+1}'(0) = \mathbf{g}_{k+1}^T \mathbf{d}_{k+1} = -\mathbf{g}_{k+1}^T B_{k+1}^{-1} \mathbf{g}_{k+1} < 0$$

(unless $\mathbf{g}_{k+1} = \mathbf{0}$), i.e. $\mathbf{d}_{k+1}$ is a well defined descent direction in $\mathbf{x}_{k+1}$.

Denote by $\mathcal{I}_0$ the level set $\{\mathbf{x} : f(\mathbf{x}) \leq f(\mathbf{x}_0)\}$. As a consequence of Proposition 1 and the subsequent considerations, we have

Proposition 2. *Assume that the step-lengths λ_k satisfy the AG conditions in (36). Then Algorithm 3.1 yields a sequence of points $\mathbf{x}_{k+1}$, $k = 0, 1, \ldots,$ such that*

$$f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k) \quad \textit{and} \quad \mathbf{y}_k^T \mathbf{s}_k > 0.$$

Therefore, $\mathbf{x}_{k+1}$ belongs to the set $\mathcal{I}_0$ and the matrix $B_{k+1} = \Phi(\tilde{B}_k, \mathbf{s}_k, \mathbf{y}_k)$ is well defined and pd, until $\mathbf{g}_k = \mathbf{0}$.

Now assume that $\mathbf{g}_k \neq \mathbf{0}$, $\forall k$ (otherwise the algorithm terminates in a finite number of steps at a stationary point for f). Since $\{f(\mathbf{x}_k)\}_{k\in\mathbb{N}}$ is a lower bounded strictly decreasing sequence, obviously $\lim_{k\to\infty} f(\mathbf{x}_k) \geq \inf f(\mathbf{x})$.

In the following fundamental theorem we prove that under special prescriptions on $\tilde{B}_k$ and suitable analytical properties of f, a subsequence of $\{\mathbf{g}_k\}_{k\in\mathbb{N}}$ converges to the null vector. For the sake of completeness we recall all the steps of the proof of global convergence of NS BFGS-type algorithms in [7]. The present proof is different only for the last part, where it is shown the role of the third further condition in (37) in proving the convergence of S BFGS-type methods.

Theorem 3. *Let $\tilde{B}_k$ in Algorithm 3.1 satisfy the conditions:*

$$\operatorname{tr} B_k \geq \operatorname{tr} \tilde{B}_k, \tag{37a}$$

$$\det B_k \leq \det \tilde{B}_k, \tag{37b}$$

$$\frac{\|B_k \mathbf{s}_k\|^2}{(\mathbf{s}_k^T B_k \mathbf{s}_k)^2} \leq \frac{\|\tilde{B}_k \mathbf{s}_k\|^2}{(\mathbf{s}_k^T \tilde{B}_k \mathbf{s}_k)^2}. \tag{37c}$$

If $\exists M > 0$ such that

$$\frac{\|\mathbf{y}_k\|^2}{\mathbf{y}_k^T \mathbf{s}_k} \leq M \tag{38}$$

then

$$\liminf \|\mathbf{g}_k\| = 0. \tag{39}$$

Proof. The points $\mathbf{x}_{k+1}$ are in the level set $\mathcal{I}_0$ and satisfy conditions (36):

$$AG_1: \ f(\mathbf{x}_{k+1}) \le f(\mathbf{x}_k) + \lambda_k c_1 \mathbf{g}_k^T \mathbf{d}_k$$
$$AG_2: \ \mathbf{g}_{k+1}^T \mathbf{d}_k \ge c_2 \mathbf{g}_k^T \mathbf{d}_k$$

Applying AG_1 for $k = 0, \dots, j$ we have:

$$c_1 \sum_{k=0}^{j} \lambda_k \left(-\mathbf{g}_k^T \mathbf{d}_k\right) \le f(\mathbf{x}_0) - f(\mathbf{x}_{j+1}) \le f(\mathbf{x}_0) - \inf f(\mathbf{x}) < \infty.$$

From the convergence of $\sum_{k=0}^{+\infty} \lambda_k(-\mathbf{g}_k^T \mathbf{d}_k)$ we have:

$$\lambda_k \left(-\mathbf{g}_k^T \mathbf{d}_k\right) = -\mathbf{g}_k^T \mathbf{s}_k = \|\mathbf{g}_k\| \|\mathbf{s}_k\| \cos\theta_k \to 0 \tag{40}$$

where $\theta_k = \widehat{(-\mathbf{g}_k)\mathbf{d}_k}$. On the other hand, by (37) for $k = 0, \dots, j$:

$$\begin{aligned} \operatorname{tr} B_{k+1} &= \operatorname{tr} \tilde{B}_k + \frac{1}{\mathbf{y}_k^T \mathbf{s}_k} \mathbf{y}_k^T \mathbf{y}_k - \frac{1}{\mathbf{s}_k^T \tilde{B}_k \mathbf{s}_k} (\tilde{B}_k \mathbf{s}_k)^T (\tilde{B}_k \mathbf{s}_k) \\ &\le \operatorname{tr} B_k + \frac{1}{\mathbf{y}_k^T \mathbf{s}_k} \|\mathbf{y}_k\|^2 - \frac{1}{\mathbf{s}_k^T \tilde{B}_k \mathbf{s}_k} \|\tilde{B}_k \mathbf{s}_k\|^2. \end{aligned} \tag{41}$$

Hence:

$$\operatorname{tr} B_{j+1} \le \operatorname{tr} B_0 + \sum_{k=0}^{j} \frac{1}{\mathbf{y}_k^T \mathbf{s}_k} \|\mathbf{y}_k\|^2 - \sum_{k=0}^{j} \frac{1}{\mathbf{s}_k^T \tilde{B}_k \mathbf{s}_k} \|\tilde{B}_k \mathbf{s}_k\|^2. \tag{42}$$

Thus, by (38)

$$\operatorname{tr} B_{j+1} \le \operatorname{tr} B_0 + M(j+1) \le c_3(j+1).$$

Let us remember that, given n real positive numbers a_i, it holds:

$$\prod_{i=1}^{n} a_i \le \left(\frac{\sum_{i=1}^{n} a_i}{n}\right)^n \tag{43}$$

from which we obtain:

$$\det B_{j+1} = \prod_{i=1}^{n} \nu_i(B_{j+1}) \le \left(\frac{\sum_{i=1}^{n} \nu_i(B_{j+1})}{n}\right)^n \le \left(\frac{c_3(j+1)}{n}\right)^n. \tag{44}$$

Let us remember moreover that from (42), since B_{j+1} is positive definite, we have:

$$\sum_{k=0}^{j} \frac{1}{\mathbf{s}_k^T \tilde{B}_k \mathbf{s}_k} \|\tilde{B}_k \mathbf{s}_k\|^2 \le \operatorname{tr} B_0 - \operatorname{tr} B_{j+1} + \sum_{k=0}^{j} \frac{1}{\mathbf{y}_k^T \mathbf{s}_k} \|\mathbf{y}_k\|^2$$

$$\leq \operatorname{tr} B_0 + \sum_{k=0}^{j} \frac{1}{\mathbf{y}_k^T \mathbf{s}_k} \|\mathbf{y}_k\|^2 \leq c_3(j+1) \tag{45}$$

and applying once more (43) we have:

$$\prod_{k=0}^{j} \frac{1}{\mathbf{s}_k^T \tilde{B}_k \mathbf{s}_k} \|\tilde{B}_k \mathbf{s}_k\|^2 \leq c_3^{j+1}. \tag{46}$$

From (37) and from direct calculation of the determinant it holds:

$$\det B_{k+1} = \frac{\mathbf{s}_k^T \mathbf{y}_k}{\mathbf{s}_k^T \tilde{B}_k \mathbf{s}_k} \det \tilde{B}_k \geq \frac{\mathbf{s}_k^T \mathbf{y}_k}{\mathbf{s}_k^T \tilde{B}_k \mathbf{s}_k} \det B_k, \quad k = 0, \dots, j,$$

from which we obtain:

$$\prod_{k=0}^{j} \frac{\mathbf{s}_k^T \mathbf{y}_k}{\mathbf{s}_k^T \tilde{B}_k \mathbf{s}_k} \leq \frac{\det B_{j+1}}{\det B_0}. \tag{47}$$

Let us observe that AG_2 implies

$$\mathbf{y}_k^T \mathbf{s}_k \geq (c_2 - 1)\mathbf{g}_k^T \mathbf{s}_k = (1 - c_2)\left(-\mathbf{g}_k^T \mathbf{s}_k\right).$$

From Eqs. (44), (46), (47) and the third in (37), we have:

$$\begin{aligned}
(1-c_2)^{j+1} \prod_{k=0}^{j} \frac{\|\mathbf{g}_k\|^2}{\mathbf{s}_k^T(-\mathbf{g}_k)} &\leq \prod_{k=0}^{j} \frac{\|-\lambda_k \mathbf{g}_k\|^2}{\mathbf{s}_k^T(-\lambda_k \mathbf{g}_k)} \frac{\mathbf{s}_k^T \mathbf{y}_k}{\mathbf{s}_k^T(-\lambda_k \mathbf{g}_k)} \\
&= \prod_{k=0}^{j} \frac{\|B_k \mathbf{s}_k\|^2}{\mathbf{s}_k^T B_k \mathbf{s}_k} \frac{\mathbf{s}_k^T \mathbf{y}_k}{\mathbf{s}_k^T B_k \mathbf{s}_k} \\
&\leq \prod_{k=0}^{j} \frac{\|\tilde{B}_k \mathbf{s}_k\|^2}{\mathbf{s}_k^T \tilde{B}_k \mathbf{s}_k} \frac{\mathbf{s}_k^T \mathbf{y}_k}{\mathbf{s}_k^T \tilde{B}_k \mathbf{s}_k} \leq c_3^{j+1} \left(\frac{c_3(j+1)}{n}\right)^n \frac{1}{\det B_0}
\end{aligned} \tag{48}$$

and hence

$$\prod_{k=0}^{j} \frac{\|\mathbf{g}_k\|}{\|\mathbf{s}_k\| \cos \theta_k} \leq c_4^{j+1} \tag{49}$$

where c_4 is a suitable constant. Since identities (40) and (49) hold simultaneously, a sub-sequence of $\{\|\mathbf{g}_k\|\}_{k\in\mathbb{N}}$ must be convergent to zero. □

Corollary 1. *Let (37) and (38) hold and let $\mathcal{I}_0$ be bounded. Then a sub-sequence of $\{\mathbf{x}_k\}_{k\in\mathbb{N}}$ converges to a stationary point $\mathbf{x}_*$ of the function f and $f(\mathbf{x}_k) \to f(\mathbf{x}_*)$.*

Proof. See [7]. □

A sufficient condition to fulfill the inequality (38) is shown in the next Proposition 3. It consists in a suitable convexity assumption on f.

Proposition 3. *(See [13].) Assume that f is convex and has continuous and bounded second derivatives in a convex set $\mathcal{I} \subset \mathbb{R}^n$. Then, for all $\mathbf{x}, \mathbf{y} \in \mathcal{I}$,*

$$\left\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\right\|^2 \leq M\left(\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\right)^T(\mathbf{x} - \mathbf{y})$$

where $\|\nabla^2 f(\mathbf{x})\| \leq M$ in $\mathcal{I}$.

Corollary 2. *Let f be a twice continuously differentiable function in the level set $\mathcal{I}_0$. Assume $\mathcal{I}_0$ convex and bounded. Let $\tilde{B}_k$ satisfy the conditions (37). Then $\{f(\mathbf{x}_k)\}_{k\in\mathbb{N}}$ converges to the least value of f.*

Remark 5. It is important to notice that a sufficient condition for the third inequality in (37) to be verified is that

$$B_k\mathbf{s}_k = \sigma_k\tilde{B}_k\mathbf{s}_k.$$

In the previous section we have proved that this condition can be imposed or, more precisely, we have proved that the equivalent condition

$$\tilde{B}_k^{-1}\mathbf{g}_k = \sigma_k B_k^{-1}\mathbf{g}_k$$

can be forced to hold.

The above Remark 5 shows how trying to force the global convergence for the Secant method can be done by forcing that the Non-Secant method produces the same search direction of the Secant one.

6. Convergent algorithm construction

In this section we will show how the three convergence conditions (37) of Theorem 3 can be verified by $\tilde{B}_{k+1}$. For the sake of simplicity, we will drop the index $k+1$ in all symbols B, $\mathbf{g}$, λ, $\mathbf{s}$, $\mathbf{y}$, $\mathbf{d}$, $\mathbf{z}$, $\mathbf{x}$, $\mathcal{L}$, whereas "$-$" and "$+$" will denote the k-th and $(k+2)$-th indexes, respectively.

Let $\mathbf{x} \in \mathbb{R}^n$ be the current guess of a stationary point of f. Let $\mathbf{g} = \nabla f(\mathbf{x}) \in \mathbb{R}^n$ be a non-null vector, and let $B \in \mathbb{R}^{n\times n}$ be the real symmetric positive definite matrix, depending on $\mathbf{g}$,

$$B = \Phi(\tilde{B}_-, \mathbf{s}_-, \mathbf{y}_-),$$

where $\tilde{B}_- = U_- d(\mathbf{z}_-) U_-^H \in \mathcal{L}^- := \operatorname{sd} U_-$, U_- is a unitary matrix and $\mathbf{s}_- = \mathbf{x} - \mathbf{x}_-$, $\mathbf{y}_- = \mathbf{g} - \mathbf{g}_-$.

In Algorithm 3.1, at Line 7, once the computation of $\mathbf{d} = -B^{-1}\mathbf{g}$ has been performed — i.e. it is available the search direction we are going to use in the next step of the algorithm to obtain $\mathbf{x}_+ = \mathbf{x} + \lambda \mathbf{d}$ —, it is required to define the *new* unitary matrix U (the structure of $\tilde{B}$), the *new* vector $\mathbf{z} > 0$ (the information content of $\tilde{B}$), and thus the *new* matrix $\tilde{B} = U d(\mathbf{z}) U^H \in \mathcal{L} = \operatorname{sd} U$ which satisfies the conditions in (37) to ensure convergence of the Secant BFGS-type algorithm (see Remark 1). The *new* matrix $\tilde{B}$ we are going to define will be used in the next step of the algorithm in order to produce a Hessian approximation of the form

$$B_+ = \Phi(\tilde{B}, \mathbf{s}, \mathbf{y}),$$

and finally a new descent direction

$$\mathbf{d}_+ = -B_+^{-1}\mathbf{g}_+$$

in the computed guess $\mathbf{x}_+$. The matrix $\tilde{B}$ will be different from $\mathcal{L}_B$, but $\tilde{B}$ will be a matrix of $\mathcal{L}$ whose eigenvalues are the eigenvalues of $\mathcal{M}_B$ where $\mathcal{M} = \operatorname{sd} V$ and V is a suitable unitary matrix (possibly $V = U_-$).

We now indicate a procedure which yields $\mathbf{z} > \mathbf{0}$, U unitary, $\tilde{B} = U d(\mathbf{z}) U^H$ satisfying (37). Start from an arbitrary unitary matrix V and $\mathcal{M} = \operatorname{sd} V$, and set $z_i = (V^H B V)_{ii} = \lambda_i(\mathcal{M}_B)$. Observe that

- the $z_i = (V^H B V)_{ii}$ depend on the eigenvalues of $\tilde{B}_-$ and can be computed from them by the updating formula:

$$z_i = \left[V^H \tilde{B}_- V\right]_{ii} + \frac{1}{\mathbf{y}_-^T \mathbf{s}_-}\left|\left(V^H \mathbf{y}_-\right)_i\right|^2 - \frac{1}{\mathbf{s}_-^T \tilde{B}_- \mathbf{s}_-}\left|\left(V^H \tilde{B}_- \mathbf{s}_-\right)_i\right|^2, \tag{50}$$

  for all $i \in \{1, \dots, n\}$, at a cost clearly dependent on the cost of the transforms involving U_- and V. In particular, for $V = U_-$ we have the simpler formula

$$z_i = [\mathbf{z}_-]_i + \frac{1}{\mathbf{y}_-^T \mathbf{s}_-}\left|\left(U_-^H \mathbf{y}_-\right)_i\right|^2 - \frac{1}{\mathbf{z}_-^T |U_-^H \mathbf{s}_-|^2}\left|\left(d(\mathbf{z}_-) U_-^H \mathbf{s}_-\right)_i\right|^2 \tag{51}$$

  (see [7]).
- the $z_i = (V^H B V)_{ii}$ satisfy the following inequalities (see Section 2):

$$0 < \min_j \lambda_j(B) \le z_i \le \max_j \lambda_j(B), \quad \text{for all } i \in \{1, \dots, n\},$$

$$1 \le \mu(\mathcal{M}_B) = \frac{\max_j z_j}{\min_j z_j} \le \mu(B), \tag{52}$$

$$\det(B) \le \det(\mathcal{M}_B) = \prod_{i=1}^{n} z_i, \tag{53}$$

$$\operatorname{tr}(B) \ge \operatorname{tr}(\mathcal{M}_B) = \sum_{i=1}^{n} z_i. \tag{54}$$

If, moreover, the z_i satisfy the following inequality

$$\frac{(\mathbf{d}^T(-\mathbf{g}))^2}{\|\mathbf{d}\|^2\|\mathbf{g}\|^2} = \frac{(\mathbf{g}^T B^{-1}\mathbf{g})^2}{(\mathbf{g}^T B^{-2}\mathbf{g})(\mathbf{g}^T\mathbf{g})} \ge \frac{4\mu(\mathcal{M}_B)}{(1+\mu(\mathcal{M}_B))^2}, \tag{55}$$

then, by Theorem 1, there exist a unitary U and $\sigma > 0$ such that

$$-\left(U \operatorname{diag}(z_i) U^H\right)^{-1}\mathbf{g} = -\sigma B^{-1}\mathbf{g}, \tag{56}$$

i.e. Problem 2 has a solution. Observe that, using Theorem 1, the matrix U can be constructed with $O(n)$ FLOPS. Thus, setting $\mathcal{L} := \operatorname{sd} U$ and $\tilde{B} = U \operatorname{diag}(z_i) U^H \in \mathcal{L}$, and noting that $\mu(\tilde{B}) = \mu(\mathcal{M}_B)$, $\det(\tilde{B}) = \det(\mathcal{M}_B)$, $\operatorname{tr}(\tilde{B}) = \operatorname{tr}(\mathcal{M}_B)$, from (53), (54), (55) it follows that the three convergence conditions (37) on $\tilde{B}$, in the Secant BFGS-type algorithm, are satisfied (by (56) and by Remark 5 we have that $B\mathbf{s}$ is a multiple of $\tilde{B}\mathbf{s}$ when $\tilde{B} = U \operatorname{diag}(z_i) U^H$, and thus (37c) holds). As a consequence of the two remarks here below, we shall see that, when (55) does not hold, the z_i can be corrected, without compromising (53), (54), until (55) is verified.

Remark 6. If V is the matrix which diagonalizes B (i.e. $B = \mathcal{M}_B$ with $\mathcal{M} = \operatorname{sd} V$), since B is pd, the inequality (55) is satisfied, i.e. we have

$$1 \ge \frac{(\mathbf{g}^T B^{-1}\mathbf{g})^2}{(\mathbf{g}^T B^{-2}\mathbf{g})(\mathbf{g}^T\mathbf{g})} \ge \frac{4\mu(B)}{(1+\mu(B))^2} \tag{57}$$

(the inequality on the left becomes an equality if $\mathbf{g}$ is eigenvector of B). In fact

$$\frac{(\mathbf{g}^T B^{-1}\mathbf{g})^2}{(\mathbf{g}^T B^{-2}\mathbf{g})(\mathbf{g}^T\mathbf{g})} = \frac{(\mathbf{y}^T\mathbf{y})}{(\mathbf{y}^T B^{-1}\mathbf{y})(\mathbf{y}^T B\mathbf{y})}, \quad \mathbf{y} = B^{-1/2}\mathbf{g},$$

and

$$\frac{4\mu(B)}{(1+\mu(B))^2} = \frac{4\lambda_{\min}(B)\lambda_{\max}(B)}{(\lambda_{\min}(B)+\lambda_{\max}(B))^2}.$$

Then (57) follows from the Kantorovich inequality.

Remark 7. For any V we have

$$1 \ge \frac{4\mu(\mathcal{M}_B)}{(1+\mu(\mathcal{M}_B))^2} \ge \frac{4\mu(B)}{(1+\mu(B))^2} \tag{58}$$

because the function

$$g(x) = \frac{4x}{(1+x)^2}$$

is decreasing for $x \geq 1$ and takes values in $(0, 1]$ (see inequality (52)).

From the last two remarks it follows that if (55) is not verified, i.e. if

$$\frac{4\mu(\mathcal{M}_B)}{(1+\mu(\mathcal{M}_B))^2} \geq \frac{(\mathbf{g}^T B^{-1}\mathbf{g})^2}{(\mathbf{g}^T B^{-2}\mathbf{g})(\mathbf{g}^T\mathbf{g})},$$

we can change $\mathcal{M}$ (or, directly, the z_i) until (53), (54), (55) are all fulfilled. More specifically, we can set $V := VQ$ for a suitable unitary matrix Q (and hence $\mathcal{M} = \operatorname{sd} V$), until the corrected $z_i := (V^H BV)_{ii} = \lambda_i(\mathcal{M}_B)$ satisfy (55) (observe that (53) and (54) are automatically satisfied by such z_i).

Eventually, besides the *good* z_i, we also obtain the unitary matrix V such that $z_i = (V^H BV)_{ii}$, and $\tilde{B}$ turns out to be the matrix

$$\tilde{B} = U \operatorname{diag}\big(\big(V^H BV\big)_{ii}\big)U^H.$$

So $\tilde{B}$ is not equal to $\mathcal{L}_B = U \operatorname{diag}((U^H BU)_{ii})U^H$. In fact, $\tilde{B}$ is the matrix of $\mathcal{L} = \operatorname{sd} U$ whose eigenvalues are the eigenvalues of $\mathcal{M}_B = V \operatorname{diag}((V^H BV)_{ii})V^H$, $\mathcal{M} = s\mathbf{d}\, V$ (compare with (14)).

Remark 8. In case (55) is not satisfied by the first z_i proposed, instead of changing the z_i one may try to give up the equality $B\mathbf{s} = \sigma\tilde{B}\mathbf{s}$ and impose on $\tilde{B}$ the weaker condition (37c).

Remark 9. We know, from a theorem due to Zoutendijk (see [12, p. 43]), that any iteration $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$ where $\mathbf{d}_k$ is a descent direction in $\mathbf{x}_k$, λ_k satisfies the Armijo–Goldstein conditions, f is bounded below, and $\nabla f(\mathbf{x})$ satisfies the Lipschitz condition, is globally convergent whenever

$$\cos(\widehat{-\mathbf{g}_k \mathbf{d}_k}) \geq \delta > 0, \quad \text{for infinite } k. \tag{59}$$

Thus, since the weak convexity assumption (38) implies a (discrete) Lipschitz inequality for $\mathbf{g}_k$, that is, $\|\mathbf{y}_k\| \leq M\|\mathbf{s}_k\|$, we would have convergence of Secant BFGS-type algorithms *provided that* (59) *holds, or, equivalently,* $\mu(B_k) \leq M_\delta$ *for all* k *(without requiring the three convergence conditions (37) on* $\tilde{B}_k$*)*. It is interesting to compare the general convergence condition (59) with our convergence condition (55), which regards, instead of general iterative schemes $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$, a special class of *adaptive* methods that combine the advantages of Secant and Non-Secant $\mathcal{L}^{(k)}$QN algorithms. Our algorithms automatically satisfy (37a), (37b), and their adaptive character seem to require, in order to reach convergence (via (37c)), a condition (55), different from (59), that relates the condition numbers $\mu(\tilde{B})$ and $\mu(B)$, and may require, in order to be verified, to change $\mathcal{M}$ so that $\mu(\mathcal{M}_B) = \mu(\tilde{B})$ approaches the condition numbers $\mu(B)$; note that no fixed lower bound δ for $\cos(\widehat{-\mathbf{g}_k\mathbf{d}_k})$ is required.

7. The convergent algorithm

In this section, we exhibit the convergent *hybrid* $\mathcal{L}^{(k)}$QN algorithm whose basic step has been explained in Section 6. We will write $\mathsf{K}(\mathbf{z}) = T$ or $\mathsf{K}(\mathbf{z}) = F$ to denote that a vector $\mathbf{z} > 0$ does or does not satisfy the inequality (25). Let us give the following definitions.

Definition 2. Given $\mathbf{z} > 0$ such that $\mathsf{K}(\mathbf{z}) = T$ we will use the notation

$$U = \mathbf{dHc}(\mathbf{z}) \tag{60}$$

(double Householder construction), to denote the unitary matrix constructed in Theorem 1. Let us underline, once more, that U is constructed as the product of two Householder matrices.

Definition 3. Given a vector $\mathbf{z} > 0$ such that $\sum_{i=1}^{n} z_i \leq \operatorname{tr} B$ and $\prod_{i=1}^{n} z_i \geq \det B$, if $\mathsf{K}(\mathbf{z}) = F$, we will designate with **SC** (Spectrum Correction) a correction strategy such that, for

$$\hat{\mathbf{z}} = \mathbf{SC}(\mathbf{z})$$

we have

$$\mathsf{K}(\hat{\mathbf{z}}) = T, \qquad \sum_{i=1}^{n} \hat{z}_i \leq \operatorname{tr} B, \qquad \prod_{i=1}^{n} \hat{z}_i \geq \det B.$$

Moreover, set

$$[2Ho]^{(k)} = \operatorname{sd} U_k, \quad \text{where } U_k = H(\mathbf{w}_k)H(\mathbf{v}_k). \tag{61}$$

Below we illustrate in detail the algorithm, where V_{k+1} is, at each step, initially set equal to U_k (see Algorithm 7.1).

We emphasize that, by taking into account the crucial steps of Algorithm 7.1 (formula (50) and the construction of U_{k+1}), apart from possible corrections of $[\mathbf{z}_{k+1}]_{\min}$ and $[\mathbf{z}_{k+1}]_{\max}$, the computational cost per step is $O(n)$ FLOPS and $O(n)$ memory allocations.

Algorithm 7.1: The hybrid $\mathcal{L}^{(k)}$QN convergent algorithm.

```
Data: $\mathbf{x}_0 \in \mathbb{R}^n$, $\mathbf{g}_0 = \nabla f(\mathbf{x}_0)$, $B_0$ pd, $\mathbf{d}_0 = -B_0^{-1}\mathbf{g}_0$,
define $U_0 = H(\mathbf{w}_0)H(\mathbf{v}_0)$, $\mathcal{L}^{(0)} = [2Ho]^{(0)} = \mathrm{sd}\, U_0$, $\tilde{B}_0 = U_0 d(\mathbf{z}_0) U_0^H \in \mathcal{L}^{(0)}$ pd, $k := 0$.
while $\mathbf{g}_k \neq 0$ do
    $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$;                                   /* $\lambda_k$ verifies (36) */
    $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$, $\mathbf{g}_{k+1} = \nabla f(\mathbf{x}_{k+1})$;
    $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$;
    $B_{k+1} = \Phi(\tilde{B}_k, \mathbf{s}_k, \mathbf{y}_k)$ ;
    $\mathbf{d}_{k+1} = -B_{k+1}^{-1}\mathbf{g}_{k+1}$;
    Define $\mathcal{L}^{(k)}_{B_{k+1}} = [2Ho]^{(k)}_{B_{k+1}}$;
    Compute $\mathbf{z}_{k+1} = \lambda([2Ho]^{(k)}_{B_{k+1}}) = ((U_k^H B_{k+1} U_k)_{ii})_{i=1}^n$;      /* see identity (50) with $V = U_k$ */
    if $\mathsf{K}(\mathbf{z}_{k+1}) = F$ then
        $V_{k+1} = U_k Q$,  $\mathcal{M} = \mathrm{sd}\, V_{k+1}$;
        $\hat{\mathbf{z}}_{k+1} = \mathbf{SC}(\mathbf{z}_{k+1}) = \lambda(\mathcal{M}^{(k+1)}_{B_{k+1}}) = ((V_{k+1}^H B_{k+1} V_{k+1})_{ii})_{i=1}^n$;
        $\mathbf{z}_{k+1} = \hat{\mathbf{z}}_{k+1}$;
    $U_{k+1} = \mathbf{dHc}(\mathbf{z}_{k+1}) = H(\mathbf{w}_{k+1})H(\mathbf{v}_{k+1})$;
    $\tilde{B}_{k+1} = U_{k+1} d(\mathbf{z}_{k+1}) U_{k+1}^H$;
    $\mathcal{L}^{(k+1)} = [2Ho]^{(k+1)} = \mathrm{sd}\, U_{k+1}$;
    /* Note: $-\tilde{B}_{k+1}^{-1}\mathbf{g}_{k+1} = -\sigma_{k+1} B_{k+1}^{-1}\mathbf{g}_{k+1} = \sigma_{k+1}\mathbf{d}_{k+1}$ */
    $k := k + 1$
```

8. Preliminary experimental results

In this section we show some preliminary experimental results. Further deeper numerical experiences will be collected in a paper in preparation. The experiments shown in the following are obtained by applying methods defined from *weaker* requirements compared to those used to construct Algorithm 7.1. In fact, we require that the condition

$$-\tilde{B}_{k+1}^{-1}\mathbf{g}_{k+1} = \sigma_{k+1}\mathbf{d}_{k+1} \quad \text{where } \mathbf{d}_{k+1} = -B_{k+1}^{-1}\mathbf{g}_{k+1}, \tag{62}$$

sufficient to ensure inequality (37c), is fulfilled for

$$\tilde{B}_{k+1} = U_{k+1} d\big(\mathbf{z}^{(k)}_{B_{k+1}}\big) U_{k+1}^H, \quad \sigma_{k+1} > 0, \quad \text{and} \quad U_{k+1} = H(\mathbf{u}_{k+1}) \tag{63}$$

($\mathcal{L}^{(k)}_{B_{k+1}} = U_k d(\mathbf{z}^{(k)}_{B_{k+1}}) U_k^H$), i.e. we require that the vector $\mathbf{u}_{k+1}$ satisfies

$$\big[\big(\mathbf{u}_{k+1}^H \mathbf{g}_{k+1}\big) I + \sigma_{k+1}\big(\mathbf{u}_{k+1}^H \mathbf{d}_{k+1}\big) d\big(\mathbf{z}^{(k)}_{B_{k+1}}\big)\big]\mathbf{u}_{k+1} = \mathbf{g}_{k+1} + \sigma_{k+1} d\big(\mathbf{z}^{(k)}_{B_{k+1}}\big)\mathbf{d}_{k+1}, \tag{64}$$

and, moreover, we set, in (64), $\sigma_{k+1} = 1$ and in the square brackets in (64), we replace $\mathbf{u}_{k+1}$ with an arbitrary $\mathbf{w}_{k+1}$ chosen in order to make (64) linear and uniquely solvable in $\mathbf{u}_{k+1}$. This way we have the following condition:

$$\big[\big(\mathbf{w}_{k+1}^H \mathbf{g}_{k+1}\big) I + \big(\mathbf{w}_{k+1}^H \mathbf{d}_{k+1}\big) d\big(\mathbf{z}^{(k)}_{B_{k+1}}\big)\big]\mathbf{u}_{k+1} = \mathbf{g}_{k+1} + d\big(\mathbf{z}^{(k)}_{B_{k+1}}\big)\mathbf{d}_{k+1}. \tag{65}$$

Here below we present numerical results obtained with $\mathbf{u}_{k+1}$ (in (63)) defined by (65), where

$$\mathbf{w}_{k+1} = \alpha\left(\mathbf{d}_{k+1} + \frac{\|\mathbf{d}_{k+1}\|}{\|\mathbf{g}_{k+1}\|}\mathbf{g}_{k+1}\right), \tag{66}$$

$$\mathbf{w}_{k+1} = \alpha\left(\mathbf{g}_{k+1} - \frac{\|\mathbf{g}_{k+1}\|^2}{\mathbf{d}_{k+1}^T\mathbf{g}_{k+1}}\mathbf{d}_{k+1}\right) \tag{67}$$

and α is such that $\|\mathbf{u}_{k+1}\|^2 = 2$. Observe that the complexity of the methods obtained in this way is $O(n)$ in time and space. The methods are applied to four known test functions f (see [5]). In the table are reported the numbers of iterations required by the methods corresponding to (66) and (67) to ensure that the inequality $f(\mathbf{x}_k) < 10^{-4}$ is fulfilled.

	Problem size	Rosenbrock	Powell	Wood $n = 4$	Helical $n = 3$
(66)	$n = 12 \times 10^0$	it = 15	it = 32		
	$n = 12 \times 10^1$	it = 16	it = 62	it = 48	it = 42
	$n = 12 \times 10^2$	it = 11	it = 41		
	$n = 12 \times 10^3$	it = 15	it = 60		
(67)	$n = 12 \times 10^0$	it = 12	it = 19		
	$n = 12 \times 10^1$	it = 17	it = 28	it = 53	it = 39
	$n = 12 \times 10^2$	it = 14	it = 80		
	$n = 12 \times 10^3$	it = 11	it = 291		
$\mathcal{H}$QN	$n = 12 \times 10^0$	it = 68	it = 59		
	$n = 12 \times 10^1$	it = 82	it = 142	it = 44	it = 35
	$n = 12 \times 10^2$	it = 169	it = 289		
	$n = 12 \times 10^3$	–	–		

These first experiments, enough encouraging, show that the $\mathcal{L}^{(k)}$QN methods considered in this paper, even in the *weaker* form described above, can be competitors of the $\mathcal{L}$QN methods ($\mathcal{H}$QN method in the table is $\mathcal{L}$QN where $\mathcal{L}$ is Hartley matrix algebra, see [2] and [7]), which in turn have been shown to be competitors of L-BFGS (see [2]).

Acknowledgement

We are grateful to the referee for his comments and useful suggestions.

References

[1] C.G. Broyden, J.E. Dennis Jr., J.J. Moré, On the local and superlinear convergence of quasi-Newton methods, IMA J. Appl. Math. 12 (1973) 223–245.

[2] A. Bortoletti, C. Di Fiore, S. Fanelli, P. Zellini, A new class of quasi-Newtonian methods for optimal learning in MLP-networks, IEEE Trans. Neural Net. 14 (2003) 263–273.

[3] J.F. Cai, R.H. Chan, C. Di Fiore, Minimization of a detail-preserving regularization functional for impulse noise removal, J. Math. Imaging Vision 29 (2007) 79–91.

[4] J.E. Dennis Jr., J.J. Moré, Quasi-Newton methods, motivation and theory, SIAM Rev. 19 (1977) 46–89.

[5] J.E. Dennis, R.B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice-Hall, Englewood Cliffs, New Jersey, 1983.

[6] C. Di Fiore, Structured matrices in unconstrained minimization methods, in: Proceedings of Fast Algorithms for Structured Matrices: Theory and Applications, AMS-IMS-SIAM Joint Summer Research Conference on Fast Algorithms in Mathematics, Computer Science and Engineering, August 5–9, 2001, Mount Holyoke College, South Hadley, Massachusetts, in: Contemp. Math., vol. 323, 2003.

[7] C. Di Fiore, S. Fanelli, F. Lepore, P. Zellini, Matrix algebras in quasi-Newton methods for unconstrained minimization, Numer. Math. 94 (2003) 479–500.

[8] C. Di Fiore, S. Fanelli, P. Zellini, Low-complexity minimization algorithms, Numer. Linear Algebra Appl. 12 (2005) 755–768.

[9] C. Di Fiore, S. Fanelli, P. Zellini, Low complexity secant quasi-Newton minimization algorithms for nonconvex functions, J. Comput. Appl. Math. 210 (2007) 167–174.

[10] C. Di Fiore, F. Lepore, P. Zellini, Hartley-type algebras in displacement and optimization strategies, Linear Algebra Appl. 366 (2003) 215–232.

[11] C. Di Fiore, P. Zellini, Matrix algebras in optimal preconditioning, Linear Algebra Appl. 335 (2001) 1–54.

[12] J. Nocedal, S.J. Wright, Numerical Optimization, Springer, New York, 1999.

[13] M.J.D. Powell, Some global convergence properties of a variable metric algorithm for minimization without exact line search, in: Nonlinear Programming, in: SIAM-AMS Proc., vol. 9, 1976, pp. 53–72.

[14] E.E. Tyrtyshnikov, Optimal and superoptimal circulant preconditioners, SIAM J. Matrix Anal. Appl. 13 (1992) 459–473.

[15] J.H. Wilkinson, The Algebraic Eigenvalue Problem, Clarendon Press, Oxford University Press, Walton Street, 1965.