

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/319486999>

Preconditioned Nonlinear Conjugate Gradient methods based on a modified secant equation

Article in *Applied Mathematics and Computation* · September 2017

DOI: 10.1016/j.amc.2017.08.029

CITATIONS

13

READS

189

3 authors:



Andrea Caliciotti

Sapienza University of Rome

10 PUBLICATIONS 60 CITATIONS

[SEE PROFILE](#)



Giovanni Fasano

Università Ca' Foscari Venezia

69 PUBLICATIONS 822 CITATIONS

[SEE PROFILE](#)



Massimo Roma

Sapienza University of Rome

49 PUBLICATIONS 760 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Numerical infinities and infinitesimals in applied and pure mathematics, computer science, and philosophy of science [View project](#)



Conjugate Gradient methods and Approximate Inverse Preconditioners [View project](#)

Preconditioned Nonlinear Conjugate Gradient methods based on a modified secant equation

ANDREA CALICIOTTI¹, GIOVANNI FASANO², MASSIMO ROMA¹

¹ Dipartimento di Ingegneria Informatica, Automatica e Gestionale “A. Ruberti”
SAPIENZA, Università di Roma
via Ariosto, 25 – 00185 Roma, Italy
E-mail: caliciotti,roma@dis.uniroma1.it

² Department of Management
University Ca’ Foscari of Venice
S. Giobbe, Cannaregio 873 – 30121 Venice, Italy
E-mail: fasano@unive.it

Abstract. This paper includes a twofold result for the Nonlinear Conjugate Gradient (NCG) method, in large scale unconstrained optimization. First we consider a theoretical analysis, where preconditioning is embedded in a strong convergence framework of an NCG method from the literature. Mild conditions to be satisfied by the preconditioners are defined, in order to preserve NCG convergence. As a second task, we also detail the use of novel matrix-free preconditioners for NCG. Our proposals are based on quasi-Newton updates, and either satisfy the secant equation or a *secant-like* condition at some of the previous iterates. We show that, in some sense, the preconditioners we propose also approximate the inverse of the Hessian matrix. In particular, the structures of our preconditioners depend on low-rank updates used, along with different choices of specific parameters. The low-rank updates are obtained as by-product of NCG iterations. The results of an extended numerical experience using large scale CUTEst problems is reported, showing that our preconditioners can considerably improve the performance of NCG methods.

Keywords: Nonlinear Conjugate Gradient method, Large Scale optimization, Secant equation, Low-rank updates.

1 Introduction

Several iterative methods have been proposed in the literature, for the solution of the large scale unconstrained problem $\min_{x \in \mathbb{R}^n} f(x)$, where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ (see e.g. [11, 21, 28]). Among them, the NCG along with quasi-Newton methods are undoubtedly the most commonly used, since they both proved to be really effective in practice and show mature convergence properties (see e.g. [28, 29]).

In the case we consider the minimization of a convex quadratic function, then the NCG and the BFGS quasi-Newton method show a well studied correspondence of the search directions they respectively generate [26]. This spots some light on the relation between the latter two classes of methods, and in this paper we want to partially exploit benefits from possibly coupling their respective underlying ideas.

On this purpose let us first consider a Preconditioned Nonlinear Conjugate Gradient (PNCG) method, where essentially three choices at current step k strongly determine both the effectiveness and the efficiency of the overall method. In particular, the first choice refers to the *linesearch procedure*, which selects the steplength $\alpha_k > 0$ used to compute the next iterate x_{k+1} , being

$$x_{k+1} = x_k + \alpha_k p_k.$$

Then, the second choice refers to the selection of the parameter β_{k+1} , which determines the next search direction as

$$p_{k+1} = -g_{k+1} + \beta_{k+1} p_k.$$

Finally, a proper choice $M_{k+1} \in \mathbb{R}^{n \times n}$ for a preconditioner may also be part of the computation of p_{k+1} , as in

$$p_{k+1} = -M_{k+1} g_{k+1} + \beta_{k+1} p_k.$$

The latter three choices are not independent, inasmuch as for instance an improper preconditioner risks to possibly destroy both convergence properties and numerical performance of the PNCG. This observation imposes some care before adopting a preconditioner, in order to first verify that it complies with the requirements claimed by the convergence analysis.

In the first part of this paper we show that, given a theoretically strong convergence framework of a well-known NCG method, under mild assumptions on a preconditioning strategy, we can easily obtain a PNCG scheme which preserves convergence properties. Then, we report rather efficient proposals of our preconditioners, which actually ensure convergence for the PNCG.

Observe that addressing good preconditioners for NCG methods still remains an intriguing research issue, in particular when the solution of large scale problems is sought and no structure of the problem is known in advance [6, 9, 12, 21, 29, 30]. Similarly, matrix-free preconditioning for linear systems or sequences of linear systems is currently an appealing research topic, too (see e.g. [4, 5, 13, 14]).

On this guideline, this work is devoted to investigate ideas from quasi-Newton updates, in order to build possible preconditioners for NCG. In particular, we are interested both to obtain a numerically efficient preconditioner, and to analyze its theoretical properties. The preconditioners we propose are iteratively constructed and based on satisfying a modified secant equation, and partially recover the structure of quasi-Newton updates. To sum up, we definitely remark that our preconditioners are designed for the PNCG and

- do not rely on the structure of the minimization problem in hand;
- are matrix-free, hence they are naturally conceived for large scale problems;
- are built drawing inspiration from quasi-Newton schemes;
- convey information from previous iterations of the PNCG.

Finally, for the sake of completeness we urge to recall that the idea of using a quasi-Newton update as a possible preconditioner, within NCG algorithms, is not new; examples of such an approach can be found for instance in [3, 8, 24].

The paper is organized as follows: in Section 2 some preliminaries on PNCG are reported. In Section 3 we give theoretical results on the global convergence of an effective PNCG algorithm. Section 4 reports a description of a recent preconditioner from the literature. In Section 5 we describe a novel preconditioner which is based on a *modified* secant equation. Section 6 details a numerical experience which refers to our preconditioner described in Section 5. Finally, Section 7 reports some concluding remarks. An Appendix completes the paper.

As regards the notation, $\|\cdot\|$ denotes the Euclidean norm. Moreover, we indicate with $A \succ 0$ the positive definiteness of matrix $A \in \mathbb{R}^{n \times n}$, and we use $tr(A)$ for its *trace*, while $\lambda_m(A)$ [$\lambda_M(A)$] is used for the smallest [largest] eigenvalue of A . Finally, given the function $f(x_k)$ and its gradient $g(x_k)$ at the iterate x_k , we use respectively f_k and g_k to denote them.

2 General remarks on PNCG

Here we first recall a general scheme of a Preconditioned Nonlinear Conjugate Gradient (PNCG) algorithm (see also [29] for a general survey). Then, we detail the description of the rationale behind our proposal, namely the idea of using low-rank updates to satisfy a secant-like equation. In the following PNCG scheme the matrix $M_k \in \mathbb{R}^{n \times n}$ denotes a possible positive definite preconditioner at the iteration k .

Preconditioned Nonlinear Conjugate Gradient (PNCG) scheme

Step 1: Data $x_1 \in \mathbb{R}^n$ and $M_1 \succ 0$. Set $p_1 = -M_1 g_1$ and $k = 1$.

Step 2: Use a linesearch procedure to compute the steplength α_k , which satisfies *Wolfe conditions*, and set the next iterate as

$$x_{k+1} = x_k + \alpha_k p_k.$$

Step 3: If a *stopping criterion* is satisfied then STOP, else compute the coefficient β_{k+1} , along with the preconditioner $M_{k+1} \succ 0$. Compute the search direction

$$p_{k+1} = -M_{k+1} g_{k+1} + \beta_{k+1} p_k, \quad (2.1)$$

set $k = k + 1$ and go to *Step 2*.

Evidently, in case $M_k = I$ for any k , the PNCG scheme reduces to the NCG method. As an alternative, in order to improve, at least partially, the efficiency of the NCG by introducing preconditioning strategies (see also Section 3), the *Step 3* of PNCG might be replaced by the next one.

Step 3: If a *stopping criterion* is satisfied then STOP, else compute the coefficient β_{k+1} , along with the preconditioner M_{k+1} . If $M_{k+1} \not\succeq 0$ or $M_{k+1} g_{k+1} = 0$ then set $M_{k+1} = I$. Compute the search direction

$$p_{k+1} = -M_{k+1} g_{k+1} + \beta_{k+1} p_k,$$

set $k = k + 1$ and go to *Step 2*.

The steplength α_{k+1} and the parameter β_{k+1} can be chosen in a variety of ways, in order to ensure convergence properties or to improve the overall efficiency (see e.g. [19, 21]). Here we neither intend to propose a novel choice of β_{k+1} , nor we want to consider any specific linesearch procedure to compute α_{k+1} for PNCG algorithm. To this regards, Wolfe conditions are well-suited for our purposes, inasmuch as they easily guarantee that the curvature condition

$$s_k^T y_k > 0$$

is fulfilled for all k , where

$$s_k = x_{k+1} - x_k = \alpha_k p_k, \quad y_k = g_{k+1} - g_k. \quad (2.2)$$

On the contrary, in this paper on one hand we justify in Section 3 the importance of positive definiteness for preconditioners, also in order to prove global convergence results. This will indirectly provide

additional general requirements for the construction of potentially novel preconditioners. Then, we show that our proposals comply with the latter requirements.

As already said, preconditioning is usually applied for increasing the efficiency of the NCG method. In this regard, in this work we exploit some matrix updates and their capability to possibly mimic quasi-Newton schemes, in order to generate a ‘reasonable’ approximation of the inverse Hessian matrix and use it as a preconditioner within a PNCG framework. As well known, quasi-Newton methods are iterative algorithms which generate at current step k the search direction

$$p_k = -H_k g_k,$$

being H_k an approximation of the inverse Hessian matrix $[\nabla^2 f(x_k)]^{-1}$. Then, the next iterate x_{k+1} is generated as $x_{k+1} = x_k + \alpha_k p_k$, where α_k is a steplength. The matrix H_{k+1} is obtained by a low-rank modification of matrix H_k , in order to limit the computational effort.

Among the quasi-Newton schemes, L-BFGS method is usually considered much efficient for large scale problems, provided that ill-conditioning of H_k is kept under control. This method is based on exploiting curvature information obtained from the most recent iterations. The update H_{k+1} at the k -th iteration of BFGS is given by the formula

$$H_{k+1} = V_k^T H_k V_k + \rho_k s_k s_k^T, \quad (2.3)$$

being

$$V_k = I - \rho_k y_k s_k^T \quad \text{and} \quad \rho_k = \frac{1}{y_k^T s_k}.$$

Among well known reasons for the success of the L-BFGS method we find that H_{k+1} is the positive definite matrix ‘as close as possible’ (in the sense of Frobenius norm) to the matrix H_k , and satisfying the *secant equation*

$$H_{k+1} y_k = s_k. \quad (2.4)$$

Now, we explicitly focus on the case where $f(x)$ is quadratic, i.e. $f(x) = 1/2 x^T A x - b^T x$, with $A \in \mathbb{R}^{n \times n}$ positive definite and $b \in \mathbb{R}^n$. The latter case is particularly appealing to our purposes, since it allows to exploit the strong relation between BFGS and the conjugacy of search directions with respect to matrix A [26]. Indeed, the BFGS update (2.3) is explicitly given by

$$H_k = \left(I - \frac{y_{k-1} s_{k-1}^T}{y_{k-1}^T s_{k-1}} \right)^T H_{k-1} \left(I - \frac{y_{k-1} s_{k-1}^T}{y_{k-1}^T s_{k-1}} \right) + \frac{s_{k-1} s_{k-1}^T}{y_{k-1}^T s_{k-1}}, \quad (2.5)$$

and recalling the expression of $f(x)$, along with relation $y_k = A s_k$, we can set

$$V_k = I - \frac{A s_k s_k^T}{s_k^T A s_k}, \quad (2.6)$$

where the vectors $\{p_1, \dots, p_k\}$ are mutually conjugate with respect to A . Then, using BFGS recursion (2.5), we can write

$$\begin{aligned} H_k &= V_{k-1}^T H_{k-1} V_{k-1} + \frac{s_{k-1} s_{k-1}^T}{y_{k-1}^T s_{k-1}} \\ &= V_{k-1}^T (V_{k-2}^T H_{k-2} V_{k-2}) V_{k-1} + V_{k-1}^T \frac{s_{k-2} s_{k-2}^T}{y_{k-2}^T s_{k-2}} V_{k-1} + \frac{s_{k-1} s_{k-1}^T}{y_{k-1}^T s_{k-1}}. \end{aligned} \quad (2.7)$$

Finally, the conjugacy among vectors $\{p_1, \dots, p_k\}$ with respect to A implies that for any k

$$V_k^T s_{k-1} = \left(I - \frac{A s_k s_k^T}{s_k^T A s_k} \right)^T s_{k-1} = s_{k-1},$$

and again recalling that $y_k = As_k$, relation (2.7) yields

$$H_k = V_{k-1}^T V_{k-2}^T \cdots V_1^T H_0 V_1 \cdots V_{k-2} V_{k-1} + \sum_{i=1}^{k-1} \frac{s_i s_i^T}{s_i^T A s_i}. \quad (2.8)$$

Remark 2.1 To our purposes the importance of formula (2.8) relies on the fact that, as proved in Proposition 1 of [10] (see also [17]), when $f(x)$ is quadratic the rightmost term in (2.8) may be treated as an approximation of the inverse Hessian A^{-1} . This may represent an appealing tool to build possible preconditioners, as revealed in Sections 4 and 5.

3 Global convergence for an effective PNCG

Several preconditioned NCG schemes were proposed in the literature, with a strong focus on efficiency (see e.g. [29]). The latter schemes also include algorithms where the Nonlinear Conjugate Gradient method is often integrated with an *ad hoc* preconditioner, and is coupled with a linesearch procedure to compute the steplength α_k . An example of recent methods where this approach was used are given by CG-DESCENT and L-CG-DESCENT algorithms [19, 20, 22], which include a proper linesearch procedure, along with a formula for the parameter β_k specifically defined, to ensure both global convergence and efficiency of the overall algorithm.

In this section we aim at using a globally convergent NCG scheme from the literature, endowed with strong convergence properties, and studying how embedding a positive definite preconditioner in order to preserve the global convergence. This approach on one hand provides an overall preconditioned NCG scheme with strong convergence properties; on the other hand, it gives general clear guidelines to build fruitful preconditioners.

On this purpose, we selected the NCG scheme in [18], since the authors prove rather strong and appealing convergence results for it (further results can also be found in [1] and [2]). We remark that the proposal in [18] (here addressed as PR-NCG since Polak–Ribière method is considered) also adopts a simple linesearch procedure (which devises results from the literature of derivative-free optimization methods), whose implementation is, to some extent, simpler than the use of standard Wolfe conditions. Then, in this section we show how introducing a very general preconditioner in PR-NCG, still maintaining its global convergence.

Assumption 3.1 (see also [18])

- a) Given the vector $x_1 \in \mathbb{R}^n$ and the function $f \in C^1(\mathbb{R}^n)$, the level set $\mathcal{L}_1 = \{x \in \mathbb{R}^n : f(x) \leq f_1\}$ is compact.
- b) There exists an open ball $\mathcal{B}_r := \{x \in \mathbb{R}^n : \|x\| < r\}$ containing \mathcal{L}_1 where $f(x)$ is continuously differentiable and its gradient $g(x)$ is Lipschitz continuous. In particular, there exists $L > 0$ such that

$$\|g(x) - g(y)\| \leq L\|x - y\| \quad \text{for all } x, y \in \mathcal{B}_r. \quad (3.1)$$

- c) There exist $\delta_1 > 0$, $\delta_2 > 0$ such that the preconditioner $M(x)$, for any $x \in \mathcal{B}_r$, is positive definite with eigenvalues satisfying $0 < \delta_1 < \lambda_m[M(x)] \leq \lambda_M[M(x)] < \delta_2$.

Note that by Assumption 3.1 there exists a value Ω (say $\Omega \geq 1$) such that

$$\|g(x)\| \leq \Omega, \quad \text{for all } x \in \mathcal{L}_1. \quad (3.2)$$

Moreover, due to technical reasons we assume that the radius r of \mathcal{B}_r is large enough to satisfy relation

$$r > \sup_{x \in \mathcal{L}_1} \|x\| + \frac{\rho_2}{\sigma} \Omega, \quad (3.3)$$

where $\sigma \in (0, 1)$ and $\rho_2 > 0$ (being the latter parameters used in the next Algorithm PR-NCG_M).

Now we report the algorithm PR-NCG_M, which represents our preconditioned version of the algorithm PR-NCG in [18]. Then, we are going to prove that, under Assumption 3.1, PR-NCG_M maintains the same global convergence properties of PR-NCG. For the sake of simplicity we indicate $M(x_k)$ with M_k , being $x_k \in \mathcal{B}_r$.

Algorithm PR-NCG_M

Data: Choose $\rho_2 > \rho_1 > 0$, $\gamma > 0$, $\sigma \in (0, 1)$. Set $k = 1$, $x_1 \in \mathbb{R}^n$. For any $k \geq 1$ choose M_k such that $0 < \delta_1 < \lambda_m(M_k) \leq \lambda_M(M_k) < \delta_2$.

Step 0: Set $p_1 = -M_1 g_1$.

Step 1: If $g_k = 0$ STOP.

Step 2: Set $\tau_k = \frac{|g_k^T p_k|}{\|p_k\|^2}$ and choose $\Delta_k \in [\rho_1 \tau_k, \rho_2 \tau_k]$.

Step 3: Compute $\alpha_k = \max \{\sigma^j \Delta_k, j = 0, 1, \dots\}$ such that the vectors $x_{k+1} = x_k + \alpha_k p_k$ and

$$p_{k+1} = -M_{k+1} g_{k+1} + \beta_{k+1} p_k, \text{ with } \beta_{k+1} = \frac{(g_{k+1} - g_k)^T M_{k+1} g_{k+1}}{g_k^T M_k g_k}, \text{ satisfy the conditions:}$$

$$(C1) \quad f_{k+1} \leq f_k - \gamma \alpha_k^2 \|p_k\|^2$$

$$(C2) \quad -\delta_2 \|g_{k+1}\|^2 \leq g_{k+1}^T p_{k+1} \leq -\delta_1 \|g_{k+1}\|^2.$$

Set $k = k + 1$ and go to *Step 1*.

Remark 3.1 Observe that here the parameters δ_1 and δ_2 do not have to satisfy the condition $\delta_1 < 1 < \delta_2$ as in [18]. This additional generality of our proposal relies on the freedom to choose the preconditioner. In this regard, since δ_1 and δ_2 are no more related to the unit value, no significant bound on the condition number of the preconditioner is imposed (see *c*) of Assumption 3.1.

Now, we prove first that conditions (C1) and (C2) and the entire *Step 3* of PR-NCG_M are well defined.

Proposition 3.2 *Suppose that $g_k \neq 0$ and $\|p_k\| < +\infty$ for all k . Let Assumption 3.1 hold. Then, for every $k \geq 1$ there exists a finite index j_k such that $\alpha_k = \sigma^{j_k} \Delta_k$ satisfies conditions (C1) and (C2) at Step 3.*

Proof: First observe that as $g_1^T p_1 < 0$, using (C2) we have by induction that $g_k^T p_k < 0$ for all k . Now, by assuming $g_k^T p_k < 0$, we prove that the number $\alpha_k = \Delta_k \sigma^j$ satisfies conditions (C1) and (C2) for all sufficiently large j .

By contradiction, assume first that there exists an infinite set \mathcal{J} of the index j such that condition (C1) is violated, i.e. for every $j \in \mathcal{J}$ we have:

$$\frac{f(y^{(j)}) - f_k}{\sigma^j \Delta_k} > -\gamma \sigma^j \Delta_k \|p_k\|^2,$$

where $y^{(j)} := x_k + \sigma^j \Delta_k p_k$. Then, taking the limit for $j \in \mathcal{J}, j \rightarrow +\infty$, we have $g_k^T p_k \geq 0$ which contradicts the assumption $g_k^T p_k < 0$.

Suppose now that by contradiction there exists an infinite set, say it again \mathcal{J} , such that for $j \in \mathcal{J}$ condition **(C2)** is violated. This implies that by the boundedness of $\|p_k\|$, for all $j \in \mathcal{J}$ at least one of the following conditions holds (the subscript in M_j denotes the fact that the preconditioner possibly depends on the point $y^{(j)}$):

$$g(y^{(j)})^T \left(-M_j g(y^{(j)}) + \frac{(g(y^{(j)}) - g_k)^T M_j g(y^{(j)})}{g_k^T M_k g_k} p_k \right) > -\delta_1 \|g(y^{(j)})\|^2,$$

$$g(y^{(j)})^T \left(-M_j g(y^{(j)}) + \frac{(g(y^{(j)}) - g_k)^T M_j g(y^{(j)})}{g_k^T M_k g_k} p_k \right) < -\delta_2 \|g(y^{(j)})\|^2.$$

Then, taking limits for $j \in \mathcal{J}, j \rightarrow +\infty$, we obtain that at least one of the two inequalities $-g_k^T M_k g_k \geq -\delta_1 \|g_k\|^2$ and $-g_k^T M_k g_k \leq -\delta_2 \|g_k\|^2$ must be valid. But in both cases we get a contradiction to the assumptions $g_k \neq 0$ and $0 < \delta_1 < \lambda_m(M_k) \leq \lambda_M(M_k) < \delta_2$.

Therefore, under the assumption $g_k^T p_k < 0$, we can conclude that *Step 3* is well defined, by taking j_k as the largest index for which both conditions **(C1)** and **(C2)** are satisfied, and setting the parameters as $\alpha_k = \sigma^{j_k} \Delta_k$. \square

The main properties of the sequence of iterates produced by Algorithm PR-NCG_M, which are at the basis of our convergence result, are stated in the next proposition.

Proposition 3.3 *Let Assumption 3.1 hold. Suppose that $g_k \neq 0$ and $\|p_k\| < \infty$ for all $k \geq 1$. Then we have:*

(i) $x_k \in \mathcal{L}_1$ for all $k \geq 1$;

(ii) the sequence $\{f_k\}$ has a limit;

(iii) $\lim_{k \rightarrow +\infty} \alpha_k \|p_k\| = 0$;

(iv) $\alpha_k \|p_k\|^2 \leq \rho_2 \delta_2 \Omega^2$, for all $k \geq 1$;

(v) for every k there exists a positive number θ such that $\alpha_k \geq \theta \frac{|g_k^T p_k|}{\|p_k\|^2}$.

Proof: Condition **(C1)** at *Step 3* implies

$$f_k - f_{k+1} \geq \gamma \alpha_k^2 \|p_k\|^2. \quad (3.4)$$

From (3.4) we have $x_k \in \mathcal{L}_1$ for all k , which establishes (i); then (ii) follows from (3.4) and the compactness of \mathcal{L}_1 . Recalling the expression of α_k and taking limits in (3.4) for $j \rightarrow +\infty$, then (ii) yields (iii). Now, the instructions at *Step 2* and *Step 3* imply

$$\alpha_k \|p_k\|^2 \leq \Delta_k \|p_k\|^2 \leq \rho_2 |g_k^T p_k| \leq \rho_2 \delta_2 \|g_k\|^2 \leq \rho_2 \delta_2 \Omega^2,$$

so that we get (iv).

In order to establish (v) we distinguish the two cases: $\alpha_k = \Delta_k$ and $\alpha_k < \Delta_k$, where Δ_k is the scalar defined at *Step 2*. In the first case, we have obviously

$$\alpha_k \geq \rho_1 \frac{|g_k^T p_k|}{\|p_k\|^2}. \quad (3.5)$$

Now suppose that $\alpha_k < \Delta_k$, so that $\frac{\alpha_k}{\sigma}$ violates at least one of the conditions at *Step 3*. Recalling (3.3) we have that the point

$$w_k := x_k + \frac{\alpha_k}{\sigma} p_k$$

belongs to the ball \mathcal{B}_r introduced in Assumption 3.1, being

$$\left\| \frac{\alpha_k}{\sigma} p_k \right\| \leq \rho_2 \frac{|g_k^T p_k| \|p_k\|}{\|p_k\|^2} \leq \frac{\rho_2}{\sigma} \Omega.$$

If **(C1)** is violated we can write, using the Mean Value Theorem:

$$f_k + \frac{\alpha_k}{\sigma} g_k^T p_k + \frac{\alpha_k}{\sigma} \left(g(z_k)^T p_k - g_k^T p_k \right) > f_k - \gamma \left(\frac{\alpha_k}{\sigma} \right)^2 \|p_k\|^2, \quad (3.6)$$

where $z_k := x_k + \eta_k \frac{\alpha_k}{\sigma} p_k \in \mathcal{B}_r$ with $\eta_k \in (0, 1)$. Recalling (3.1) and by the Cauchy–Schwarz inequality, we get

$$g(z_k)^T p_k - g_k^T p_k \leq |g(z_k)^T p_k - g_k^T p_k| \leq \|g(z_k) - g_k\| \|p_k\| \leq L \|z_k - x_k\| \|p_k\| \leq L \frac{\alpha_k}{\sigma} \|p_k\|^2. \quad (3.7)$$

Using (3.7) in (3.6) we get

$$\frac{\alpha_k}{\sigma} g_k^T p_k + \left(\frac{\alpha_k}{\sigma} \right)^2 L \|p_k\|^2 \geq -\gamma \left(\frac{\alpha_k}{\sigma} \right)^2 \|p_k\|^2,$$

whence we obtain

$$\alpha_k \geq \frac{\sigma}{L + \gamma} \frac{|g_k^T p_k|}{\|p_k\|^2}, \quad (3.8)$$

which proves (v). Now assume that $\frac{\alpha_k}{\sigma}$ violates **(C2)**. Suppose first that the rightmost inequality does not hold, i.e.

$$g(w_k)^T \left(-M_j g(w_k) + \frac{(g(w_k) - g_k)^T M_j g(w_k)}{g_k^T M_k g_k} p_k \right) > -\delta_1 \|g(w_k)\|^2. \quad (3.9)$$

Recalling (3.1) and by the Cauchy–Schwarz inequality we have

$$\begin{aligned} (g(w_k) - g_k)^T M_j g(w_k) &\leq |(g(w_k) - g_k)^T M_j g(w_k)| \leq \|g(w_k) - g_k\| \|M_j g(w_k)\| \\ &\leq L \|w_k - x_k\| \|M_j g(w_k)\| \leq L \frac{\alpha_k}{\sigma} \lambda_M(M_j) \|g(w_k)\| \|p_k\|. \end{aligned} \quad (3.10)$$

Furthermore we get

$$g(w_k)^T M_j g(w_k) \geq \lambda_m(M_j) \|g(w_k)\|^2; \quad (3.11)$$

thus, using (3.10) and (3.11) in (3.9) we obtain:

$$L \frac{\alpha_k}{\sigma} \lambda_M(M_j) \|g(w_k)\| \|p_k\| g(w_k)^T p_k > (\lambda_m(M_j) - \delta_1) \|g(w_k)\|^2 g_k^T M_k g_k. \quad (3.12)$$

Again, by the Cauchy–Schwarz inequality we have

$$\|g(w_k)\| \|p_k\| g(w_k)^T p_k \leq \|g(w_k)\|^2 \|p_k\|^2 \quad (3.13)$$

along with

$$g_k^T M_k g_k \geq \lambda_m(M_k) \|g_k\|^2. \quad (3.14)$$

Finally, using (3.13) and (3.14) in (3.12) we obtain

$$L \frac{\alpha_k}{\sigma} \lambda_M(M_j) \|g(w_k)\|^2 \|p_k\|^2 > (\lambda_m(M_j) - \delta_1) \|g(w_k)\|^2 \lambda_m(M_k) \|g_k\|^2. \quad (3.15)$$

Taking the limits for $j \rightarrow +\infty$ and taking into account that the rightmost inequality **(C2)** holds at Step k , we get

$$\begin{aligned}\alpha_k &\geq \frac{\sigma(\lambda_m(M_k) - \delta_1)}{L\lambda_M(M_k)} \lambda_m(M_k) \frac{\|g_k\|^2}{\|p_k\|^2} \geq \frac{\sigma(\lambda_m(M_k) - \delta_1)}{L\delta_2} \lambda_m(M_k) \frac{\|g_k\|^2}{\|p_k\|^2} \\ &\geq \frac{\sigma(\lambda_m(M_k) - \delta_1)}{\delta_2^2 L} \lambda_m(M_k) \frac{|g_k^T p_k|}{\|p_k\|^2},\end{aligned}\quad (3.16)$$

implying (v). Now suppose that **(C2)** is violated because the leftmost inequality is not fulfilled, i.e.

$$g(w_k)^T \left(-M_j g(w_k) + \frac{(g(w_k) - g_k)^T M_j g(w_k)}{g_k^T M_k g_k} p_k \right) < -\delta_2 \|g(w_k)\|^2. \quad (3.17)$$

Using a similar reasoning we obtain

$$\begin{aligned}\alpha_k &\geq \frac{\sigma(\delta_2 - \lambda_M(M_k))}{L\lambda_M(M_k)} \lambda_m(M_k) \frac{\|g_k\|^2}{\|p_k\|^2} \geq \frac{\sigma(\delta_2 - \lambda_M(M_k))}{L\delta_2} \lambda_m(M_k) \frac{\|g_k\|^2}{\|p_k\|^2} \\ &\geq \frac{\sigma(\delta_2 - \lambda_M(M_k))}{\delta_2^2 L} \lambda_m(M_k) \frac{|g_k^T p_k|}{\|p_k\|^2}.\end{aligned}\quad (3.18)$$

Thus, from (3.5), (3.8), (3.16) and (3.18) we obtain (v) by taking

$$\theta = \min \left\{ \rho_1, \frac{\sigma}{L + \gamma}, \frac{\sigma(\lambda_m(M_k) - \delta_1)}{\delta_2^2 L} \lambda_m(M_k), \frac{\sigma(\delta_2 - \lambda_M(M_k))}{\delta_2^2 L} \lambda_m(M_k) \right\}.$$

□

Finally, in order to establish the main global convergence properties of Algorithm PR-NCG_M we can state the following theorem.

Theorem 3.4 *Let $\{x_k\}$ be the sequence of points generated by Algorithm PR-NCG_M. Then, either there exists an index $\nu \geq 1$ such that $g(x_\nu) = 0$ and the algorithm terminates, or $\{x_k\}$ is an infinite sequence such that:*

(i) $\lim_{k \rightarrow +\infty} \|g(x_k)\| = 0;$

(ii) every limit point of $\{x_k\}$ is a stationary point of f .

Proof: Suppose that Algorithm PR-NCG_M does not terminate in a finite number of iterations and that (i) is false. Then, there exists a subsequence $\{x_k\}_{k \in K} \subseteq \{x_k\}$ and $\varepsilon > 0$ such that

$$\|g_{k-1}\| \geq \varepsilon, \quad \text{for all } k \in K, \quad (3.19)$$

and by (iii) of Proposition 3.3

$$\lim_{k \rightarrow +\infty, k \in K} \|x_k - x_{k-1}\| = 0.$$

Now, by the instructions of Algorithm PR-NCG_M, using (3.1), (3.2), (3.19) and (iv) of Proposition 3.3 we can write for $k \in K$:

$$\begin{aligned}
\|p_k\| &\leq \|M_k g_k\| + \left(\frac{\|g_k - g_{k-1}\| \|M_k g_k\|}{\|g_{k-1}\| \|M_{k-1} g_{k-1}\|} \|p_{k-1}\| \right) \\
&\leq \lambda_M(M_k) \Omega + \left(\frac{L \|x_k - x_{k-1}\| \lambda_M(M_k) \Omega}{\varepsilon^2 \lambda_m(M_{k-1})} \|p_{k-1}\| \right) \\
&= \lambda_M(M_k) \Omega + \left(\frac{L \|\alpha_{k-1} p_{k-1}\| \lambda_M(M_k) \Omega}{\varepsilon^2 \lambda_m(M_{k-1})} \|p_{k-1}\| \right) \\
&= \lambda_M(M_k) \Omega + \left(\frac{\lambda_M(M_k) \Omega L \alpha_{k-1} \|p_{k-1}\|^2}{\varepsilon^2 \lambda_m(M_{k-1})} \right) \\
&\leq \lambda_M(M_k) \Omega + \left(\frac{\lambda_M(M_k)}{\lambda_m(M_{k-1})} \right) \left(\frac{\Omega^3 L \rho_2 \delta_2}{\varepsilon^2} \right). \tag{3.20}
\end{aligned}$$

Therefore, using (iii) of Proposition 3.3 we get

$$\lim_{k \rightarrow +\infty, k \in K} \alpha_k \|p_k\|^2 = 0,$$

and hence by (v) of Proposition 3.3 it follows

$$\lim_{k \rightarrow +\infty, k \in K} |g_k^T p_k| = 0.$$

The latter condition implies, by (C2) of Step 3

$$\lim_{k \rightarrow +\infty, k \in K} \|g_k\| = 0,$$

so that (iii) of Proposition 3.3 and the Lipschitz continuity of g contradict (3.19). Thus, (i) holds and (ii) follows from the continuity of $f(x)$. \square

This theorem shows that for the preconditioned version PR-NCG_M the same global convergence properties of PR-NCG still hold.

4 A framework for Symmetric Rank-2 updates

The previous section reveals the importance of mild assumptions on the preconditioners, in order to prove global convergence properties of PR-NCG_M. On this guideline, now we briefly report a positive definite preconditioner for NCG, based on quasi-Newton updating formula, recently proposed in [10]. It exploits the properties of a parameter dependent symmetric rank-2 (SR2) update of the inverse Hessian matrix, and will represent one of our benchmarks for the numerical experience in Section 6. In particular (see [10]), after k iterations of the NCG let the sequence of iterates $\{x_1, \dots, x_k\}$ be available. Then, the idea is that of building a structured preconditioner M_{k+1} which satisfies the secant equation at least at the current iteration, i.e.

$$M_{k+1} y_k = s_k. \tag{4.1}$$

In a more general framework, similarly to the Broyden class, the authors in [10] address a family of preconditioners of the form

$$M_{k+1}(\gamma_{k+1}) = M_k(\gamma_k) + \Delta_k, \quad \Delta_k \in \mathbb{R}^{n \times n}, \text{ symmetric}, \tag{4.2}$$

where the sequence $\{M_k(\gamma_k)\}$ depends on the parameters $\gamma_k \in \mathbb{R}^p$, $p \geq 1$, and provides an approximation of $[\nabla^2 f(x)]^{-1}$ according with Remark 2.1. The new update $M_{k+1}(\gamma_{k+1})$ is claimed to satisfy the following appealing conditions (quoting from [10]):

- (1) $M_{k+1}(\gamma_{k+1})$ is well-defined, positive definite and can be iteratively updated;
- (2) $M_{k+1}(\gamma_{k+1})$ collects information from the iterations $k-m, k-m+1, \dots, k$, of an NCG method, being $m \geq 0$ integer;
- (3) $M_{k+1}(\gamma_{k+1})$ satisfies the secant equation at a subset K of iterations, with $K \subseteq \{1, 2, \dots, k\}$;
- (4) $M_{k+1}(\gamma_{k+1})$ “tends to preserve” in some sense the inertia of the inverse Hessian $[\nabla^2 f(x_{k+1})]^{-1}$, in case $f(x)$ is a quadratic function.

In particular, the proposal in [10] sets in (4.2)

$$\begin{aligned} M_k(\gamma_k) &= \gamma_k^{(1)} C_k, & \gamma_k^{(1)} &> 0, \quad C_k \succ 0, \\ \Delta_k &= \gamma_k^{(2)} v_k v_k^T + \gamma_k^{(3)} \sum_{j=k-m}^k \alpha_j \frac{p_j p_j^T}{y_j^T p_j}, & \gamma_k^{(2)}, \gamma_k^{(3)} &> 0, \quad v_k \in \mathbb{R}^n, \end{aligned} \quad (4.3)$$

with p_k generated at the k -th iteration of the NCG method, and α_j computed by the linesearch procedure. The resulting overall update is given by

$$M_{k+1}(\gamma_{k+1}) = \gamma_k^{(1)} C_k + \gamma_k^{(2)} v_k v_k^T + \gamma_k^{(3)} \sum_{j=k-m}^k \alpha_j \frac{p_j p_j^T}{y_j^T p_j}, \quad (4.4)$$

with

$$v_k = \sigma_k \left(s_k - \gamma_k^{(1)} C_k y_k - \gamma_k^{(3)} \sum_{j=k-m}^k \frac{p_j^T y_k}{y_j^T p_j} p_j \right), \quad \sigma_k \in \{-1, +1\}. \quad (4.5)$$

Finally, in order to satisfy the secant equation $M_{k+1}(\gamma_{k+1}) y_k = s_k$ at step k , the following relation among the parameters $\gamma_k^{(1)}$, $\gamma_k^{(2)}$, $\gamma_k^{(3)}$ and σ_k must hold

$$\gamma_k^{(2)} \sigma_k^2 = \frac{1}{s_k^T y_k - \gamma_k^{(1)} y_k^T C_k y_k - \gamma_k^{(3)} \sum_{j=k-m}^k \alpha_j \frac{(p_j^T y_k)^2}{y_j^T p_j}}. \quad (4.6)$$

The next result (see (iii) of Proposition 2 in [10]) summarizes a relevant issue of the proposal in [10], in order to satisfy the secant equation or secant-like conditions. The efficiency of the preconditioner in [10] gives a numerical evidence of the latter fact.

Proposition 4.1 ([10] - Proposition 2 (iii)) *Let $f(x) = 1/2x^T A x - b^T x$, with $A \succ 0$, and suppose $k \geq 2$ iterations of the NCG algorithm are performed, using an exact linesearch. Then, there exist values of $\gamma_k^{(1)}$, $\gamma_k^{(2)}$, $\gamma_k^{(3)}$, and a positive semidefinite matrix C_k , such that $M_{k+1} \succ 0$ and the following modified secant conditions*

$$M_{k+1} y_j = \gamma_k^{(3)} s_j, \quad j = k-m, k-m+1, \dots, k-1,$$

are satisfied.

We complete this section observing that by the choice of C_k in [10] the preconditioner in (4.4) satisfies item c) of Assumption 3.1.

5 A Symmetric Rank-2 update based on modified secant equations

In this section we propose a novel quasi-Newton updating formula, by considering the properties of a parameter dependent symmetric rank-2 (SR2) update of the inverse Hessian matrix, used as a possible preconditioner for the NCG (see [10]). We claim that our quasi-Newton update M_{k+1} , which aims at approximating $[\nabla^2 f(x)]^{-1}$ in some sense, satisfies the following *modified* secant equation along all previous directions; namely it results

$$M_{k+1}y_j = \delta_j s_j \quad \text{with} \quad \begin{cases} \delta_j > 0, & \text{for } j < k, \\ \delta_j = 1, & \text{for } j = k. \end{cases} \quad (5.1)$$

The satisfaction of (5.1) is a distinguishing property of our proposal in this paper, and though (5.1) imposes weaker conditions with respect to the satisfaction of the secant equation at any step $j < k$, numerical performance seems to confirm its effectiveness and efficiency.

On this guideline, in order to build an approximate inverse of the Hessian matrix and explicitly indicating the parameters it depends on, in place of (4.2) we consider here the update

$$M_{k+1} = \delta_k M_k + \Delta_k, \quad \Delta_k \in \mathbb{R}^{n \times n} \text{ symmetric}, \quad (5.2)$$

where the sequence $\{M_k\}$ explicitly depends on the real parameters $\delta_k, \gamma_k, \omega_k$. Considering the relation (5.2) we set now more explicitly

$$\Delta_k = \gamma_k v_k v_k^T + \omega_k \frac{p_k p_k^T}{y_k^T p_k}, \quad \gamma_k, \omega_k \in \mathbb{R} \setminus \{0\}, \quad v_k \in \mathbb{R}^n,$$

where p_k is generated at the k -th iteration of the NCG method. Thus, we have the novel update

$$M_{k+1} = \delta_k M_k + \gamma_k v_k v_k^T + \omega_k \frac{p_k p_k^T}{y_k^T p_k}, \quad \gamma_k, \omega_k \in \mathbb{R} \setminus \{0\}, \quad v_k \in \mathbb{R}^n. \quad (5.3)$$

We immediately remark that the main difference between (5.3) and the proposal in [10] relies on the following fact. In (4.4) the rightmost contribution aims at possibly computing an approximate inverse Hessian matrix. Then, the coefficients $\gamma_k^{(1)}, \gamma_k^{(2)}$ and $\gamma_k^{(3)}$ are set so that M_{k+1} fulfills the secant equation. On the contrary, in (5.3) the rightmost term of M_{k+1} plays a role similar to that of the term $\rho_k s_k s_k^T$ in (2.3), and thus it does not contain the contribution from any previous step.

In order to comply with (5.1), M_{k+1} must satisfy the secant equation $M_{k+1}y_k = s_k$, which implies

$$\delta_k M_k y_k + \gamma_k (v_k^T y_k) v_k + \omega_k \frac{p_k p_k^T}{y_k^T p_k} y_k = s_k,$$

that is

$$\gamma_k (v_k^T y_k) v_k = s_k - \delta_k M_k y_k - \omega_k p_k. \quad (5.4)$$

Therefore it results explicitly

$$v_k = \sigma_k (s_k - \delta_k M_k y_k - \omega_k p_k)$$

for some scalar $\sigma_k \in \mathbb{R}$. By replacing this expression of v_k in (5.4) we have

$$\gamma_k \sigma_k^2 [y_k^T (s_k - \delta_k M_k y_k - \omega_k p_k)] (s_k - \delta_k M_k y_k - \omega_k p_k) = s_k - \delta_k M_k y_k - \omega_k p_k.$$

Thus, the following relation among the parameters γ_k, σ_k and ω_k must hold

$$\gamma_k \sigma_k^2 = \frac{1}{s_k^T y_k - \delta_k y_k^T M_k y_k - \omega_k p_k^T y_k}.$$

By the arbitrariness of γ_k , without loss of generality, we can set $\sigma_k \in \{-1, 1\}$, i.e. $\sigma_k^2 = 1$. Therefore, in the sequel we refer to (5.3) with

$$v_k = (s_k - \delta_k M_k y_k - \omega_k p_k) \quad (5.5)$$

and the parameters γ_j and ω_j are chosen, for any $j = 1, \dots, k$, as

$$\gamma_j = \frac{1}{s_j^T y_j - \delta_j y_j^T M_j y_j - \omega_j p_j^T y_j}, \quad \omega_j \neq \frac{s_j^T y_j - \delta_j y_j^T M_j y_j}{p_j^T y_j}, \quad j = 1, \dots, k. \quad (5.6)$$

Now, in the next proposition we first consider the case of quadratic functions, and prove that under mild assumptions the update (5.3) satisfies the modified secant equation (5.1), along all previous directions.

Proposition 5.1 *Let $f(x) = \frac{1}{2}x^T A x - b^T x$, where $A \in \mathbb{R}^{n \times n}$ is symmetric and $b \in \mathbb{R}^n$. Suppose that k steps of the CG are performed, in order to detect the stationary point (if any) of the function f , and that the vectors p_1, \dots, p_k are generated. Then, the matrix M_{k+1} in (5.3), with v_k given by (5.5), satisfies the modified secant equations (5.1), provided that the nonzero coefficients $\gamma_j, \omega_j, j = 1, \dots, k$ are chosen such that (5.6) hold.*

Proof: The proof proceeds by induction: we prove it for $k = 1$ (and $j = 1$ in (5.1)), we assume it holds for $k - 1$ (and $j \leq k - 1$ in (5.1)) and prove it for k . Equations (5.1) immediately hold for $k = 1$, that is $M_2 y_1 = s_1$, as a direct consequence of (5.5) and (5.6).

Now, suppose that the relations (5.1) hold for the index $k - 1$. To complete the induction we need to prove that the relations (5.1) hold for the index k (and $j \leq k$ in (5.1)).

Firstly, note that $M_{k+1} y_k = s_k$ as a direct consequence of (5.5) and (5.6), and (5.1) hold for $j = k$. Now, we have to prove that (5.1) hold for any $j < k$. Indeed, for $j < k$ the definition of M_{k+1} yields

$$M_{k+1} y_j = \delta_k M_k y_j + \gamma_k (s_k - \delta_k M_k y_k - \omega_k p_k) (s_k - \delta_k M_k y_k - \omega_k p_k)^T y_j + \omega_k \frac{p_k^T y_j}{y_k^T p_k} p_k,$$

where $M_k y_j = \delta_j s_j, j = 1, \dots, k - 2$, and $M_k y_{k-1} = s_{k-1}$ by the inductive hypothesis. Moreover,

$$(s_k - \delta_k M_k y_k)^T y_j = s_k^T y_j - \delta_k y_k^T M_k y_j = \begin{cases} s_k^T y_j - \delta_k \delta_j y_k^T s_j = s_k^T A s_j - \delta_k \delta_j s_k^T A s_j = 0, & j < k - 1, \\ s_k^T y_{k-1} - \delta_k y_k^T s_{k-1} = s_k^T A s_{k-1} - \delta_k (A s_k)^T s_{k-1} = 0, & j = k - 1, \end{cases}$$

where the third equality holds since $y_j = A s_j$, for any $j \in \{1, \dots, k - 1\}$. Moreover, the fourth equality holds since $s_j = \alpha_j p_j, j = 1, \dots, k - 1$, and p_j are conjugate directions. Finally,

$$\omega_k p_k^T y_j = \omega_k p_k^T A s_j = \omega_k \alpha_j p_k^T A p_j = 0,$$

which again follows from the conjugacy of the directions $\{p_1, \dots, p_k\}$. Thus, (5.1) hold for index k (and $j \leq k$) and the induction is complete. \square

As an immediate consequence of the previous proposition, we give now a finite termination property for quadratic functions. More specifically, we prove that after at most n steps, the recursion (5.3) provides the matrix M_{n+1} , which is, in some sense, a *modified inverse* Hessian.

Corollary 5.2 *Let $f(x) = \frac{1}{2}x^T A x - b^T x$, where $A \in \mathbb{R}^{n \times n}$ is symmetric nonsingular and $b \in \mathbb{R}^n$. Suppose that n steps of the CG are performed, in order to detect the stationary point of the function f , and that the vectors p_1, \dots, p_n are generated.*

- If (5.1) and (5.6) hold, we have

$$M_{n+1}A = (s_1 \cdots s_n)D(s_1 \cdots s_n)^{-1} \quad \text{with} \quad D = \text{diag}\{\delta_1, \dots, \delta_{n-1}, 1\}.$$

- It results

$$\lambda_m(M_{n+1}A) = \lambda_m(D), \quad \lambda_M(M_{n+1}A) = \lambda_M(D). \quad (5.7)$$

Proof: The proof immediately follows by Proposition 5.1. \square

We highlight that, whenever $k = n$, Corollary 5.2 justifies item (4) on page 11. Moreover, later on in the paper we show that for $k < n$, the update in (5.3) can be suitably used to provide a preconditioner. The next corollary details further properties of M_{k+1} in (5.3), again when the function $f(x)$ is quadratic, in the light of possibly approximating the inverse Hessian A^{-1} .

Corollary 5.3 *Let $f(x) = \frac{1}{2}x^T Ax - b^T x$, where $A \in \mathbb{R}^{n \times n}$ is symmetric nonsingular and $b \in \mathbb{R}^n$. Suppose that n steps of the CG are performed, in order to detect the stationary point of the function f , and that the vectors p_1, \dots, p_n are generated. If relations (5.1) and (5.6) hold for any $k = 1, \dots, n$, then we have*

$$1) \sum_{j=1}^k \delta_j s_j = (M_{k+1}A)(x_{k+1} - x_1),$$

$$2) AM_{k+1} \left[\sum_{j=1}^k \frac{g_{j+1} - g_j}{\delta_j} \right] = g_{k+1} - g_1,$$

3) in the case $k = n$ then

$$\sum_{j=1}^n [(s_1 \cdots s_n)D(s_1 \cdots s_n)^{-1} - \delta_j I] s_j = 0, \quad \text{where} \quad D = \text{diag}\{\delta_1, \dots, \delta_{n-1}, 1\}.$$

Proof: By (5.1) and adding with respect to index j we obtain the two relations

$$M_{k+1} [g_{k+1} - g_1] = \sum_{j=1}^k \delta_j s_j \quad (5.8)$$

$$M_{k+1} \left[\sum_{j=1}^k \frac{g_{j+1} - g_j}{\delta_j} \right] = x_{k+1} - x_1. \quad (5.9)$$

By (5.8) and recalling that $f(x)$ is quadratic, we have

$$M_{k+1}A(x_{k+1} - x_1) = \sum_{j=1}^k \delta_j s_j, \quad (5.10)$$

which yields 1). By (5.9) we immediately infer 2). Finally, by Corollary 5.2

$$M_{n+1}A = (s_1 \cdots s_n)D(s_1 \cdots s_n)^{-1}$$

and using (5.10) with $k = n$ we have

$$(s_1 \cdots s_n)D(s_1 \cdots s_n)^{-1} \sum_{j=1}^n s_j = \sum_{j=1}^n \delta_j s_j,$$

i.e.

$$\sum_{j=1}^n [(s_1 \cdots s_n) D(s_1 \cdots s_n)^{-1} - \delta_j I] s_j = 0,$$

which is relation 3). □

After analyzing the case of $f(x)$ quadratic, we turn now to the general case of a nonlinear continuously differentiable function. In particular, since we are interested in using the matrix M_{k+1} in (5.3) as a preconditioner, recalling Section 3 we need to investigate if there exists a suitable setting of the parameters δ_k , γ_k and ω_k such that M_{k+1} is positive definite, provided that (5.1) and (5.6) are satisfied.

In order to prove the next result, we highlight that by replacing (5.5) in (5.3), we obtain

$$\begin{aligned} M_{k+1} &= \delta_k M_k + \gamma_k \left[(\alpha_k - \omega_k)^2 p_k p_k^T - \delta_k (\alpha_k - \omega_k) \left((M_k y_k) p_k^T + p_k (M_k y_k)^T \right) \right. \\ &\quad \left. + \delta_k^2 (M_k y_k) (M_k y_k)^T \right] + \omega_k \frac{p_k p_k^T}{y_k^T p_k}. \end{aligned}$$

Hence, M_{k+1} can be rewritten in the form

$$M_{k+1} = \delta_k M_k + \Delta_k, \tag{5.11}$$

with

$$\Delta_k = \begin{pmatrix} p_k & M_k y_k \end{pmatrix} \begin{pmatrix} \gamma_k (\alpha_k - \omega_k)^2 + \frac{\omega_k}{y_k^T p_k} & -\delta_k \gamma_k (\alpha_k - \omega_k) \\ -\delta_k \gamma_k (\alpha_k - \omega_k) & \delta_k^2 \gamma_k \end{pmatrix} \begin{pmatrix} p_k^T \\ (M_k y_k)^T \end{pmatrix}.$$

Proposition 5.4 *Let f be a continuously differentiable function. Suppose that the NCG method is used to minimize the function f . Suppose that (5.1) and (5.6) are satisfied and let us set the arbitrary sequence of positive values $\{\hat{\lambda}_k\}$. If $M_1 \succ 0$ with*

$$\lambda_m(M_1) \geq \hat{\lambda}_1 > 0,$$

for any k the vectors p_k and $M_k y_k$ are not parallel, and the sequence $\{\delta_k\}$ in (5.11) satisfies

$$\begin{cases} \delta_k \geq \frac{\hat{\lambda}_{k+1}}{\lambda_m(M_k)} & \text{if } \lambda_m(\Delta_k) \geq 0, \\ \delta_k \geq \frac{\hat{\lambda}_{k+1} - \lambda_m(\Delta_k)}{\lambda_m(M_k)} & \text{if } \lambda_m(\Delta_k) < 0, \end{cases} \tag{5.12}$$

then the matrix M_{k+1} in (5.3) is positive definite, for any $k \geq 1$, with

$$\lambda_m(M_{k+1}) \geq \hat{\lambda}_{k+1}.$$

Proof: Since by hypothesis $\lambda_m(M_1) \geq \hat{\lambda}_1 > 0$, then a sufficient condition to have $\lambda_m(M_2) \geq \hat{\lambda}_2 > 0$ is that $\lambda_m(\Delta_1) \geq 0$, so that relation $\lambda_m(M_2) \geq \delta_1 \lambda_m(M_1) \geq \hat{\lambda}_2$ is fulfilled choosing $\delta_1 \geq \hat{\lambda}_2 / \lambda_m(M_1)$. On the other hand, if $\lambda_m(\Delta_1) < 0$ we can always set $\hat{\lambda}_2$ such that

$$\lambda_m(M_2) = \lambda_m(\delta_1 M_1 + \Delta_1) \geq \delta_1 \lambda_m(M_1) + \lambda_m(\Delta_1) \geq \hat{\lambda}_2 > 0,$$

which is satisfied provided that

$$\delta_1 \geq \frac{\hat{\lambda}_2 - \lambda_m(\Delta_1)}{\lambda_m(M_1)}.$$

Note that by (A.1) of Lemma A.1 (see the Appendix), regardless of the choice of γ_1 and ω_1 in (5.6), the quantity $\lambda_m(\Delta_1)$ can be obtained directly as

$$\lambda_m(\Delta_1) = \min\{\lambda_{n-1}, \lambda_n, 0\}, \quad (5.13)$$

where λ_{n-1} and λ_n are defined in (A.1), setting in Lemma A.1 $v = p_1$, $u = M_1 y_1$, $a = \gamma_1(\alpha_1 - \omega_1)^2 + \omega_1/y_1^T p_1$, $b = -\delta_1 \gamma_1(\alpha_1 - \omega_1)$ and $c = \delta_1^2 \gamma_1$.

Now assuming that $\lambda_m(M_k) \geq \hat{\lambda}_k$, we want to prove the result for step $k+1$. To this purpose, recalling again that by Lemma A.1 we can compute similarly to (5.13)

$$\lambda_m(\Delta_k) = \min\{\lambda_{n-1}, \lambda_n, 0\},$$

the choice (5.12) immediately yields $\lambda_m(M_{k+1}) \geq \hat{\lambda}_{k+1}$. \square

The result in Proposition 5.4 gives a characterization of the spectrum of M_{k+1} , but possibly it does not indicate a procedure to set the parameters affecting the formula of M_{k+1} . Moreover, the hypothesis that for any k the vectors p_k and $M_k y_k$ are not parallel may be difficult to be guaranteed. Thus, in order to fill the latter gap and provide a set of parameters δ_k , ω_k and γ_k , such that conditions (5.6) are satisfied (i.e. equivalently (5.1) hold) and the preconditioner M_k is positive definite, the following proposition may represent an operative tool. In particular, observe that unlike Proposition 5.4, the next result neither requires to compute $\lambda_m(\Delta_k)$ nor it needs to introduce the sequence $\{\hat{\lambda}_k\}$.

Proposition 5.5 *Let f be a continuously differentiable function. Suppose that the NCG method is used to minimize the function f . Suppose that at current step k the linesearch procedure satisfies $s_k^T y_k > 0$. Moreover, let $M_k \succ 0$ in (5.3), and set $\varepsilon_k \in (0, 1)$, with*

$$0 < \delta_k = (1 - \varepsilon_k) \frac{s_k^T y_k}{y_k^T M_k y_k}, \quad (5.14)$$

$$0 < \omega_k < \varepsilon_k \alpha_k, \quad (5.15)$$

$$0 < \gamma_k = \frac{1}{(\varepsilon_k \alpha_k - \omega_k) p_k^T y_k}. \quad (5.16)$$

Then, conditions (5.1) and (5.6) hold and $M_{k+1} \succ 0$ in (5.3).

Proof: By (5.14) and recalling that $\varepsilon_k \in (0, 1)$, with $M_k \succ 0$, we obtain

$$0 < \delta_k < \frac{s_k^T y_k}{y_k^T M_k y_k},$$

which implies also

$$s_k^T y_k - \delta_k y_k^T M_k y_k > 0. \quad (5.17)$$

Now, observe that by the first relation (5.6) $\gamma_k > 0$ if and only if $s_k^T y_k - \delta_k y_k^T M_k y_k - \omega_k p_k^T y_k > 0$, i.e.

$$\omega_k < \frac{s_k^T y_k - \delta_k y_k^T M_k y_k}{p_k^T y_k},$$

which can be satisfied using (5.14) as

$$0 < \omega_k < \frac{s_k^T y_k - \delta_k y_k^T M_k y_k}{p_k^T y_k} = \varepsilon_k \alpha_k.$$

The latter relation is indeed the condition (5.15), and satisfies also the second relation (5.6). Finally, by (5.14) the first relation (5.6) yields (5.16) with $(\varepsilon_k \alpha_k - \omega_k) p_k^T y_k > 0$. Then, (5.14)-(5.16) yield exactly (5.1) and (5.6), along with $M_{k+1} \succ 0$. \square

5.1 Issues on ill-conditioning

The previous proposition ensures that, properly choosing the parameters δ_k , ω_k and γ_k , the preconditioner M_{k+1} is well-posed and positive definite. However, we should take into account that the search direction p_k we compute at iteration k of NCG could be not well scaled, which may introduce some ill-conditioning when applying the PNCG. Following the rationale behind the BFGS updates, a possible remedy to the latter drawback can be represented by reducing the *distance* between M_{k+1} and M_k , i.e. minimizing the Frobenious norm $\|M_{k+1} - M_k\|_F$. In this regard, as well known we have

$$\|M_{k+1} - M_k\|_F = \sqrt{\text{tr}[(M_{k+1} - M_k)^T(M_{k+1} - M_k)]} = \sqrt{\text{tr}[(M_{k+1} - M_k)^2]}. \quad (5.18)$$

By the properties of the trace of matrices (see e.g. [7]), we have

$$\sqrt{\text{tr}[(M_{k+1} - M_k)^2]} \leq \sqrt{[\text{tr}(M_{k+1} - M_k)]^2} = |\text{tr}(M_{k+1} - M_k)|. \quad (5.19)$$

Thus, a possible upper bound for $\|M_{k+1} - M_k\|_F$ is given by

$$\|M_{k+1} - M_k\|_F \leq |\text{tr}(M_{k+1} - M_k)|. \quad (5.20)$$

We recall that by Proposition 5.5, $\{\gamma_k\}$ and $\{\omega_k\}$ are positive sequences, so that using (5.3) in (5.20) we have

$$\begin{aligned} |\text{tr}(M_{k+1} - M_k)| &= \left| \delta_k \text{tr}(M_k) + \gamma_k \|v_k\|^2 + \omega_k \frac{\|p_k\|^2}{y_k^T p_k} - \text{tr}(M_k) \right| \\ &\leq |\delta_k - 1| \text{tr}(M_k) + \gamma_k \|v_k\|^2 + \omega_k \frac{\|p_k\|^2}{y_k^T p_k}. \end{aligned} \quad (5.21)$$

From (5.15) and (5.16) we observe that, after setting the arbitrary parameter ε_k , then γ_k still depends on ω_k . Thus, following the rationale behind the BFGS update, in the following proposition we investigate possible values for the bound (5.21) on $\text{tr}(M_{k+1} - M_k)$, when ω_k changes.

Proposition 5.6 *Let f be a continuously differentiable function. Suppose that the NCG method is used to minimize f . Suppose that at current step k we have $s_k^T y_k > 0$, $M_k \succ 0$, $\varepsilon_k \in (0, 1)$, and let (5.1), (5.6), (5.14)–(5.16) be satisfied. Consider the function of ω_k*

$$\phi(\omega_k) = |\delta_k - 1| \text{tr}(M_k) + \gamma_k \|v_k\|^2 + \omega_k \frac{\|p_k\|^2}{y_k^T p_k}. \quad (5.22)$$

Then $\phi(\omega_k)$ is monotone non decreasing with respect to ω_k , and $\omega_k = 0$ minimizes it.

Proof: After setting $\varepsilon_k \in (0, 1)$, by (5.16) we note that γ_k depends on ω_k . Thus, the function $\phi(\omega_k)$ in (5.22) uniquely depends on ω_k . Now we have for $\phi'(\omega_k)$

$$\phi'(\omega_k) = \frac{d(\gamma_k \|v_k\|^2)}{d\omega_k} + \frac{\|p_k\|^2}{y_k^T p_k} = \frac{d\gamma_k}{d\omega_k} \|v_k\|^2 + \gamma_k \frac{d(\|v_k\|^2)}{d\omega_k} + \frac{\|p_k\|^2}{y_k^T p_k}. \quad (5.23)$$

By using (5.5) and (5.16) in (5.23) we obtain

$$\phi'(\omega_k) = \frac{C + 2B\varepsilon_k\alpha_k + A\varepsilon_k^2\alpha_k^2}{D(\varepsilon_k\alpha_k - \omega_k)^2}, \quad (5.24)$$

where

$$A = \|p_k\|^2, \quad B = \delta_k p_k^T M_k y_k - \alpha_k \|p_k\|^2, \quad C = \|s_k - \delta_k M_k y_k\|^2, \quad D = p_k^T y_k.$$

Now, by (5.15) and $s_k^T y_k > 0$ the quantity $D(\varepsilon_k \alpha_k - \omega_k)^2$ is strictly positive, for any ω_k . Moreover, recalling that $s_k = \alpha_k p_k$, after some computation we have

$$C + 2B\varepsilon_k \alpha_k + A\varepsilon_k^2 \alpha_k^2 = (1 - \varepsilon_k)^2 \|s_k\|^2 - 2(1 - \varepsilon_k) \delta_k s_k^T M_k y_k + \delta_k^2 \|M_k y_k\|^2. \quad (5.25)$$

Now, replacing $1 - \varepsilon_k = \theta$ in (5.25), we can introduce the function

$$\zeta(\theta) = C + 2B\varepsilon_k \alpha_k + A\varepsilon_k^2 \alpha_k^2 = \theta^2 \|s_k\|^2 - 2\theta \delta_k s_k^T M_k y_k + \delta_k^2 \|M_k y_k\|^2, \quad (5.26)$$

being after a simple computation $\zeta(\theta) \geq 0$, for any $\theta \in \mathbb{R}$. Hence, $\phi'(\omega_k) \geq 0$ in (5.24) and $\phi(\omega_k)$ in (5.22) is monotone non decreasing. Finally, by (5.15) $\omega_k = 0$ minimizes $\phi(\omega_k)$. \square

The latter proposition gives some guidelines for the choice of the parameter ε_k in (5.14)-(5.16), indicating that small positive values for ε_k tend to reduce the value of ω_k and can possibly control ill-conditioning of the matrix in (5.3).

6 Numerical Experience

Here we report the results of an extensive numerical experience, in order to validate the analysis and the theoretical achievements of the previous sections. We implement our proposal (5.3) in Fortran 90 (without recurring to the alternative *Step 3* on page 3) and, analogously to [10], we embed it in a standard implementation of NCG, namely the code CG+ by Gilbert and Nocedal [15]. We select the Polak–Ribiere method and, as regards the linesearch procedure, the standard one by Moré and Thunent [25] satisfying strong Wolfe conditions, i.e. the one adopted in CG+. As concerns the parameters of the algorithm, we use all the default values of CG+. According with Proposition 5.5, for the parameters affecting our proposal in (5.3) we used the settings

$$M_1 = I, \quad \varepsilon_k = \frac{1}{2}, \quad \delta_k = (1 - \varepsilon_k) \frac{s_k^T y_k}{y_k^T M_k y_k}, \quad \omega_k = \frac{1}{2} \varepsilon_k \alpha_k, \quad \gamma_k = \frac{\alpha_k}{(\varepsilon_k \alpha_k - \omega_k) s_k^T y_k}.$$

In order to limit the computational burden, as long as the storage requirement at iteration k , we preliminarily investigated the possibility to implement the preconditioner in (5.3) neglecting the information at iterations older than iteration $(k - m)$ -th, playing m the role of a “memory” parameter. The latter choice resembles the setting of the preconditioner proposed in [10], where only the explicit contribution of the last m pairs in (2.2) was taken into account. Not surprisingly, our numerical experience highlighted that this simplification does not deteriorate the performance. Indeed, this choice matches the rationale of the limited memory BFGS method (L-BFGS) and the value of $m = 4$ seems to provide the best compromise.

As regards the stopping criterion we adopt the standard one given by (see e.g. [23], [24], [27])

$$\|g_k\| \leq 10^{-5} \max \{1, \|x_k\|\}. \quad (6.1)$$

As a test set for our numerical experience, we select all the large scale unconstrained test problems in CUTEst collection [16]. We consider those test problems whose dimension is in the range $n = 1000$ and $n = 10000$, and whenever a variable–dimension problem is used, two different sizes are included (this sums up to 112 resulting problems). The results are reported in terms of number of iterations and number of function evaluations. As usual, when comparing two algorithms we exclude all the test problems where the compared alternatives do not converge to the same stationary point.

In Figure 6.1 we compare the performance of the preconditioner in [10] and that of our proposal in (5.3). We can easily observe that our proposal definitely outperforms the one in [10] (both in terms of iterations and function evaluations). This should be due to the fact that the preconditioner in (5.3) seems to better exploit the information collected in the history of the overall algorithm, imposing a

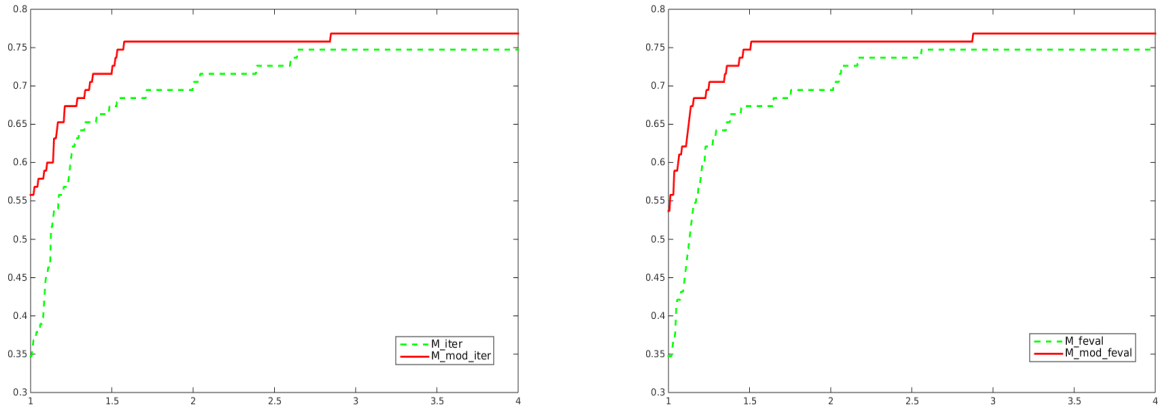


Figure 6.1: Comparison between the proposal of preconditioner in [10] (namely M , *dashed line*) and our proposal in (5.3) (namely M_{mod} , *solid line*), using the stopping criterion (6.1). Profiles with respect to number of *iterations* (left) and number of *function evaluations* (right).

modified quasi-Newton equation. The profiles reveal an appreciable improvement of the efficiency as long as the robustness.

In order to verify the effectiveness of exploiting information by the modified secant equation, we perform a comparison between our proposal in formula (5.3) and the benchmark algorithm given by the L-BFGS method (see [23], [27]). To this aim, we use the L-BFGS code available at the J. Nocedal web page. Observe that, as reported in the L-BFGS code, the linesearch procedure used therein slightly differs from the original one in [25], as (quoting from the Fortran code) “*Moré’s code has been modified so that at least one new function value is computed during the line search (enforcing at least one interpolation is not easy, since the code may override an interpolation)*”. In this comparison, we also adopt this modified linesearch procedure within our PNCG code, for the sake of correctness.

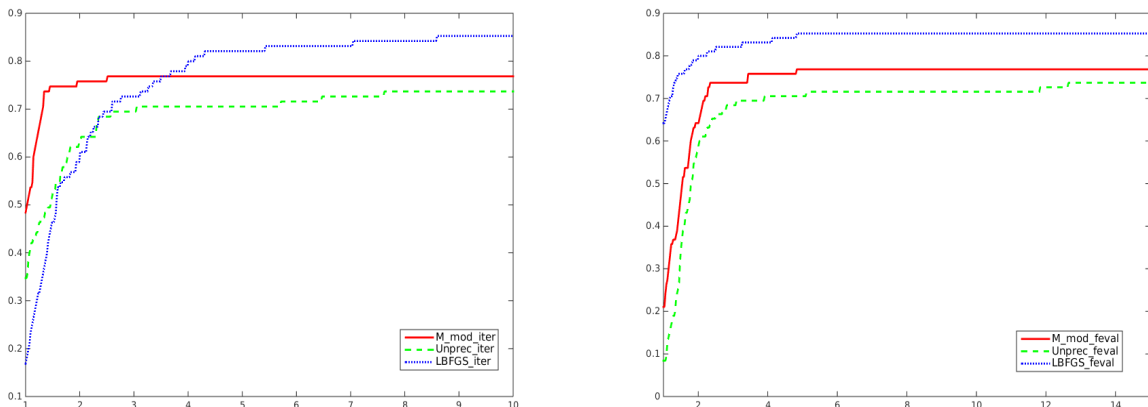


Figure 6.2: Comparison among L-BFGS (*dotted line*), our proposal in (5.3) (M_{mod} , *solid line*) and the Unpreconditioned NCG method (Unprec, *dashed line*). Profiles with respect to number of *iterations* (left) and number of *function evaluations* (right).

The results of this comparison, also including the unpreconditioned NCG case, are reported in Figure 6.2. The results w.r.t. number of iterations show that, to some extent, our proposal in (5.3) can be also competitive with L-BFGS, in terms of efficiency. On the other hand, L-BFGS seems to confirm

in any case its robustness, with respect to both a standard (unpreconditioned) NCG scheme and a preconditioned scheme including our proposal in (5.3). Conversely, L-BFGS definitely outperforms both the proposal in (5.3) and the unpreconditioned NCG, in terms of number of function evaluations. The latter fact suggests that the control of ill-conditioning of our preconditioner, as indicated by the bound (5.21), surely represents an improvement, but possibly it does not guarantee yet a fully efficient tool, so that further research in this regard seems necessary. Finally, on the overall, the results highlight that our idea in (5.3) provides a good (say efficient) search direction, but still needs a better scaling. In order to confirm this trend, we enclose in Table 6.1 the detailed results (number of iterations (it) and number of function evaluations (nf)) for those problems where our proposal compares favourably vs. L-BFGS, at least in terms of number of iterations. Table 6.1 reveals that on several test problems, our approach requires a larger amount of function evaluations w.r.t. L-BFGS, even in presence of a reduced number of iterations. This is due to a couple of facts affecting the Moré–Thuente linesearch procedure [25] used, i.e.:

- the linesearch procedure seems to be well tuned when search directions computed by quasi-Newton methods are adopted, hence the efficiency of L-BFGS (quoting from [25] “*The curvature condition [in strong Wolfe conditions] is particularly important in a quasi-Newton method, because it guarantees that a positive definite quasi-Newton update is possible*”);
- observe that, in most of the iterations, the linesearch procedure provides a unit steplength for L-BFGS, while the choice of the stepsize for PNCG is distributed on a larger interval. In Figure 6.3 a typical behaviour of the latter fact is reported (the plot refers to TRIDIA 1000 test problem). This is also motivated by a different selection of the initial stepsize in the linesearch procedure. In particular, a unit initial stepsize is used for L-BFGS, while the Shanno-Phua’s formula (default choice for CG+) is adopted as the initial trial step for our proposal in (5.3).

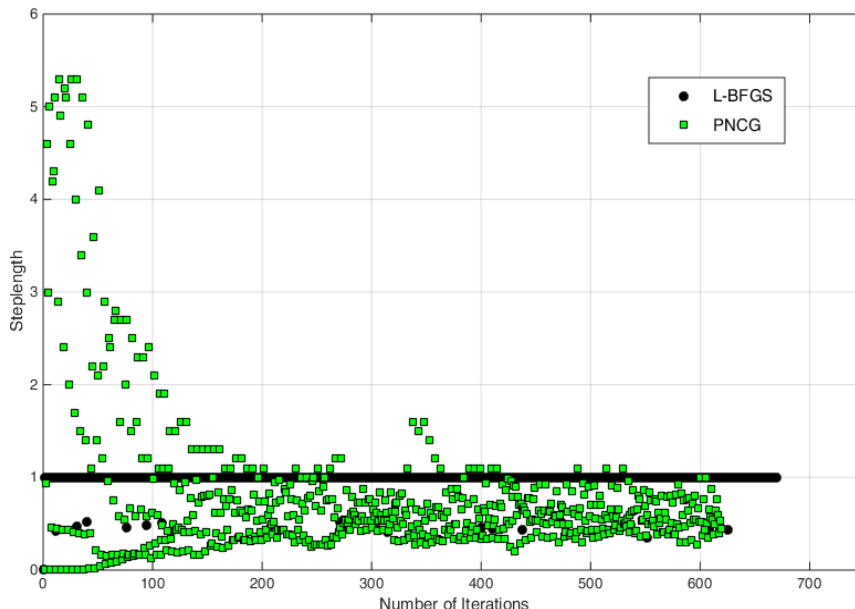


Figure 6.3: The complete sequences of steplengths generated by the linesearch procedure, when coupled to L-BFGS (*filled circles*) and to our proposal in (5.3) (*empty squares*).

On the base of the above considerations, it might be the case to investigate modifications to the linesearch procedure, to be paired with our proposal (see for instance [22]).

<i>PROBLEM</i>	<i>n</i>	<i>PNCG</i>		<i>L-BFGS</i>	
		<i>it</i>	<i>nf</i>	<i>it</i>	<i>nf</i>
ARWHEAD	1000	4	13	11	13
ARWHEAD	10000	3	11	11	14
BDQRTIC	1000	132	316	-	-
BRYBND	10000	29	70	41	50
CRAGGLVY	10000	60	136	69	79
DIXMAANA	1500	8	23	11	13
DIXMAANA	3000	9	25	12	14
DIXMAANB	1500	8	27	11	13
DIXMAANB	3000	8	27	11	13
DIXMAANC	1500	8	26	12	14
DIXMAANC	3000	8	30	2	14
DIXMAAND	1500	8	27	5	17
DIXMAAND	3000	10	33	4	16
DIXMAANE	1500	132	239	192	199
DIXMAANE	3000	224	401	252	260
DIXMAANF	1500	94	179	181	190
DIXMAANF	3000	196	343	236	241
DIXMAANG	1500	83	163	177	186
DIXMAANG	3000	153	272	226	236
DIXMAANH	1500	91	178	166	171
DIXMAANH	3000	145	270	223	229
DIXMAANJ	1500	168	301	910	941
DIXMAANJ	3000	154	277	347	359
DIXMAANK	1500	131	231	922	951
DIXMAANK	3000	139	252	326	341
DIXMAANL	1500	109	202	936	975
DIXMAANL	3000	139	252	549	564
DQDRTIC	1000	5	15	13	19
DQDRTIC	10000	6	17	13	21
EDENSCH	1000	23	66	25	29
EDENSCH	10000	20	63	25	31
FMINSURF	1024	199	353	223	224
FMINSURF	5625	428	717	614	632
FREUROTH	1000	20	55	28	38
LIARWHD	1000	14	32	20	25
MOREBV	1000	10	18	43	45
MOREBV	10000	3	5	6	8
MSQRTALS	1024	2061	3147	2143	2215
MSQRTBLS	1024	1397	2147	1811	1874
PENALTY1	10000	18	86	70	84
POWELLSG	1000	28	63	54	61
POWER	1000	134	276	139	44
POWER	10000	348	582	419	433
SCHMVETT	1000	29	63	39	45
SCHMVETT	10000	27	61	34	41
SINQUAD	1000	24	65	26	38
SPARSINE	1000	1845	3167	6029	6307
SPMSRTLS	1000	97	182	107	118
SROSENBR	1000	7	22	17	20
SROSENBR	10000	8	24	17	20
TESTQUAD	1000	1312	2322	4081	4222
TOINTGSS	1000	4	18	14	20
TOINTGSS	10000	4	20	16	23
TQUARTIC	1000	9	29	21	27
TQUARTIC	10000	12	33	22	27
TRIDIA	1000	619	960	670	695
TRIDIA	10000	1909	2851	3021	3115
VAREIGVL	1000	51	102	171	179
WOODS	1000	74	181	95	125

Table 6.1: Detailed results for those problems where our proposal in (5.3) compares favourably vs. L-BFGS.

As a final comparison, taking into account that the stopping criterion used in the original CG+ code does not match the one in (6.1), being

$$\|g_k\|_\infty \leq 10^{-5}(1 + |f_k|), \quad (6.2)$$

for the sake of completeness, we replicate the comparison between our proposal in (5.3) and that in [10], by using the default stopping criterion (6.2) in CG+. The linesearch procedure is the CG+ default one (i.e. unmodified one [25]), too. The results are reported in Figure 6.4 and confirm the conclusions inferred from Figure 6.1.

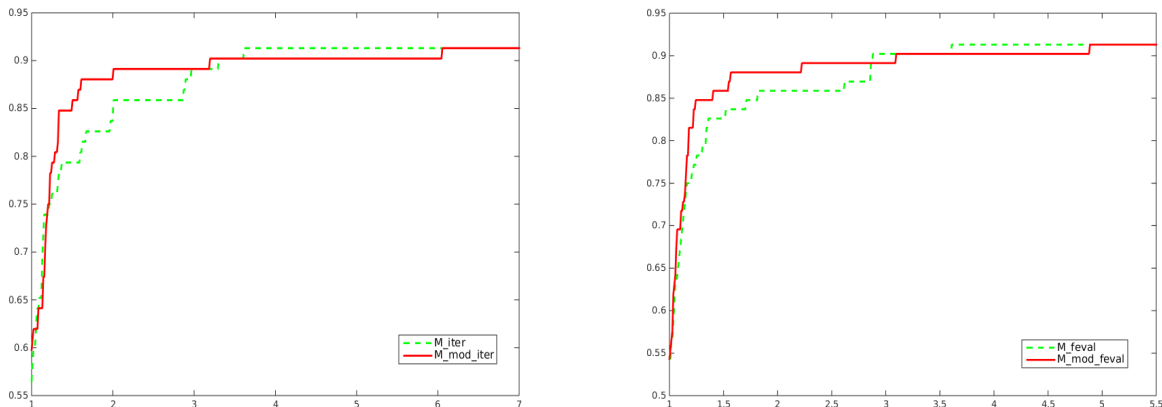


Figure 6.4: Comparison between the proposal of preconditioner in [10] (M, *dashed line*) and our proposal in (5.3) (M_mod, *solid line*), using the stopping criterion (6.2). Profiles with respect to number of *iterations* (left) and number of *function evaluations* (right).

7 Conclusions

In this paper a novel preconditioner for NCG is introduced. It is based on the satisfaction of a modified secant equation and complies with some theoretical advances for NCG global convergence. An extensive numerical testing showed that our proposal is reliable for large scale problems. Moreover, it proved to be efficient when compared with the benchmark algorithm for large scale unconstrained optimization given by L-BFGS. Further research is still expected in order to better control possible ill-conditioning of our preconditioner, since this should enhance the overall behaviour of the PNCG method.

A Appendix

Lemma A.1 *Let $u, v \in \mathbb{R}^n$ and $a, b, c \in \mathbb{R}$, with u, v linearly independent and $ac - b^2 \neq 0$. Then, the symmetric matrix $\mathcal{H} \in \mathbb{R}^{n \times n}$ given by*

$$\mathcal{H} = (v \ : \ u) \begin{pmatrix} a & b \\ b & c \end{pmatrix} (v \ : \ u)^T$$

has $n - 2$ eigenvalues equal to 0, and the two real eigenvalues

$$\lambda_{n-1} = \frac{\delta + \beta - \sqrt{(\delta - \beta)^2 + 4\alpha\gamma}}{2}, \quad \lambda_n = \frac{\delta + \beta + \sqrt{(\delta - \beta)^2 + 4\alpha\gamma}}{2}, \quad (\text{A.1})$$

where

$$\alpha = av^T u + b\|u\|^2, \quad \beta = a\|v\|^2 + bv^T u, \quad \gamma = b\|v\|^2 + cv^T u, \quad \delta = bv^T u + c\|u\|^2.$$

Proof: First note that since \mathcal{H} is symmetric then its n eigenvalues are real. Now, let $\{z_i\}$, $i = 1, \dots, n$, be an independent set of n -real vectors. Let $\{z_i\}$, $i = 1, \dots, n-2$, be orthogonal to vectors v, u . Then, for any $i \in \{1, \dots, n-2\}$ the vector z_i is trivially an eigenvector of \mathcal{H} , associated with the zero eigenvalue. Thus, the only two eigenvectors z_{n-1} and z_n of \mathcal{H} , associated with the nonzero eigenvalues λ_{n-1} and λ_n , must satisfy the relation

$$z_{n-1}, z_n \in \text{span}\{v, u\}.$$

Now, using

$$\begin{cases} z_{n-1} = \mu_{1,1}v + \mu_{1,2}u \\ z_n = \mu_{2,1}v + \mu_{2,2}u, \end{cases}$$

with $\mu_{1,1}, \mu_{1,2}, \mu_{2,1}, \mu_{2,2} \in \mathbb{R}$, and imposing the conditions $\mathcal{H}z_h = \lambda_h z_h$, for $h \in \{n-1, n\}$, we obtain the couple of relations

$$(avv^T + bvv^T + bvu^T + cuu^T)(\mu_{1,1}v + \mu_{1,2}u) = \lambda_{n-1}(\mu_{1,1}v + \mu_{1,2}u)$$

$$(avv^T + bvv^T + bvu^T + cuu^T)(\mu_{2,1}v + \mu_{2,2}u) = \lambda_n(\mu_{2,1}v + \mu_{2,2}u).$$

Note that after an easy computation, $av^T u + b\|u\|^2 \neq 0$ implies that $\mu_{1,1} \neq 0$ (indeed, $\mu_{1,1} = 0$ yields $z_{n-1} = \mu_{1,2}u$, but there is no real value for λ such that $\mathcal{H}\mu_{1,2}u = \lambda\mu_{1,2}u$). Thus, since the latter relations must hold for any choice of vectors v and u , setting $\sigma_{n-1} = \mu_{1,2}/\mu_{1,1}$ in the first of them (a similar analysis holds also for the second relation), we obtain the couple of equalities

$$[av^T u + b\|u\|^2] \sigma_{n-1} + [a\|v\|^2 + bu^T v] = \lambda_{n-1},$$

$$[b\|v\|^2 + cu^T v] + [bv^T u + c\|u\|^2] \sigma_{n-1} = \lambda_{n-1}\sigma_{n-1},$$

or equivalently

$$\begin{cases} \alpha\sigma_{n-1} + \beta = \lambda_{n-1}, \\ \gamma + \delta\sigma_{n-1} = \lambda_{n-1}\sigma_{n-1}, \end{cases}$$

which give

$$\sigma_{n-1} = \frac{(\delta - \beta) \pm \sqrt{(\delta - \beta)^2 + 4\alpha\gamma}}{2\alpha}$$

and

$$\lambda_{n-1} = \alpha\sigma_{n-1} + \beta.$$

A similar analysis holds for the computation of λ_n , which completes the proof. \square

References

- [1] M. AL-BAALI, *Descent property and global convergence of the Fletcher-Reeves method with inexact line search*, IMA Journal on Numerical Analysis, 5 (1985), pp. 121–124.
- [2] M. AL-BAALI AND R. FLETCHER, *On the order of convergence of preconditioned nonlinear conjugate gradient methods*, SIAM Journal on Scientific Computing, 17 (1996), pp. 658–665.

- [3] N. ANDREI, *Scaled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization*, Optimization Methods and Software, 22 (2007), pp. 561–571.
- [4] S. BELLAVIA, J. GONDZIO, AND B. MORINI, *A matrix-free preconditioner for sparse symmetric positive definite systems and least-squares problems*, SIAM Journal on Scientific Computing, 35 (2013), pp. A192–A211.
- [5] S. BELLAVIA, V. DE SIMONE, D. DI SERAFINO, AND B. MORINI, *A preconditioning framework for sequences of diagonally modified linear systems arising in optimization*, SIAM Journal on Numerical Analysis, 50 (2013), pp. 3280–3302.
- [6] M. BENZI, J. CULLUM, AND M. TUMA, *Robust approximate inverse preconditioner for the conjugate gradient method*, SIAM Journal on Scientific Computing, 22 (2000), pp. 1318–1332.
- [7] D. BERNSTEIN, *Matrix Mathematics*, Princeton University Press, Princeton, second ed., 2009.
- [8] B. BUCKLEY AND A. LENIR, *QN like variable storage conjugate gradients*, Mathematical Programming, 27 (1983), pp. 155–175.
- [9] A. CALICIOTTI, G. FASANO, AND M. ROMA, *Preconditioning strategies for nonlinear conjugate gradient methods, based on quasi-Newton updates*, in AIP Conference Proceedings, Y. Sergeyev, D. Kvasov, F. Dell’Accio, and M. Mukhametzhanov, eds., vol. 1776, American Institute of Physics, 2016, pp. 090007–1—090007–4.
- [10] A. CALICIOTTI, G. FASANO, AND M. ROMA, *Novel preconditioners based on quasi-Newton updates for nonlinear conjugate gradient methods*, Optimization Letters, DOI: 10.1007/s11590-016-1060-2 (First on line 09 July 2016).
- [11] A. R. CONN, N. I. M. GOULD, AND PH. L. TOINT, *Trust-region methods*, MPS–SIAM Series on Optimization, Philadelphia, PA, 2000.
- [12] I. DASSIOS, K. FOUNTOULAKIS, AND J. GONDZIO, *A preconditioner for a primal-dual Newton conjugate gradient method for compressed sensing problems*, SIAM Journal on Scientific Computing, 37 (2015), pp. A2783–A2812.
- [13] G. FASANO AND M. ROMA, *Preconditioning Newton-Krylov methods in nonconvex large scale optimization*, Computational Optimization and Applications, 56 (2013), pp. 253–290.
- [14] G. FASANO AND M. ROMA, *A novel class of approximate inverse preconditioners for large scale positive definite linear systems in optimization*, Computational Optimization and Applications, 65 (2016), pp. 399–429.
- [15] J. GILBERT AND J. NOCEDAL, *Global convergence properties of conjugate gradient methods for optimization*, SIAM Journal on Optimization, 2 (1992), pp. 21–42.
- [16] N. I. M. GOULD, D. ORBAN, AND PH. L. TOINT, *CUTEst: a constrained and unconstrained testing environment with safe threads*, Computational Optimization and Applications, 60 (2015), pp. 545–557.
- [17] S. GRATTON, A. SARTENAER, AND J. TSHIMANGA, *On a class of limited memory preconditioners for large scale linear systems with multiple right-hand sides*, SIAM Journal on Optimization, 21 (2011), pp. 912–935.
- [18] L. GRIPPO AND S. LUCIDI, *A globally convergent version of the Polak-Ribière conjugate gradient method*, Mathematical Programming, 78 (1997), pp. 375–391.

- [19] W. HAGER AND H. ZHANG, *A new conjugate gradient method with guaranteed descent and efficient line search*, SIAM Journal on Optimization, 16 (2005), pp. 170–192.
- [20] W. HAGER AND H. ZHANG, *Algorithm 851: CG_DESCENT, a conjugate gradient method with guaranteed descent*, ACM Transactions on Mathematical Software, 32 (2006), pp. 113–137.
- [21] W. HAGER AND H. ZHANG, *A survey of nonlinear conjugate gradient methods*, Pacific Journal of Optimization, 2 (2006), pp. 35–58.
- [22] W. HAGER AND H. ZHANG, *The limited memory conjugate gradient method*, SIAM Journal on Optimization, 23 (2013), pp. 2150–2168.
- [23] D. LIU AND J. NOCEDAL, *On the limited memory BFGS method for large scale optimization*, Mathematical Programming, 45 (1989), pp. 503–528.
- [24] J. MORALES AND J. NOCEDAL, *Automatic preconditioning by limited memory quasi-Newton updating*, SIAM Journal on Optimization, 10 (2000), pp. 1079–1096.
- [25] J. MORÉ AND D. THUENTE, *Line search algorithms with guaranteed sufficient decrease*, ACM Transactions on Mathematical Software (TOMS), 20 (1994), pp. 286–307.
- [26] L. NAZARETH, *A relationship between the BFGS and conjugate gradient algorithms and its implications for new algorithms*, SIAM Journal on Numerical Analysis, 16 (1979), pp. 794–800.
- [27] J. NOCEDAL, *Updating Quasi-Newton matrices with limited storage*, Mathematics of Computation, 35 (1980), pp. 773–782.
- [28] J. NOCEDAL AND S. WRIGHT, *Numerical Optimization*, Springer–Verlag, New York, 2006. Second edition.
- [29] R. PYTLAK, *Conjugate Gradient Algorithms in Nonconvex Optimization*, Springer, Berlin, 2009.
- [30] J. VLCEK AND L. LUKŠAN, *A conjugate directions approach to improve the limited-memory BFGS method*, Applied Mathematics and Computation, 219 (2012), pp. 800–809.