Springer

Dear Author,

Here are the proofs of your article.

- You can submit your corrections **online**, via **e-mail** or by **fax**.
- For **online** submission please insert your corrections in the online correction form. Always indicate the line number to which the correction refers.
- You can also insert your corrections in the proof PDF and **email** the annotated PDF.
- For fax submission, please ensure that your corrections are clearly legible. Use a fine black pen and write the correction in the margin, not too close to the edge of the page.
- Remember to note the **journal title**, **article number**, and **your name** when sending your response via e-mail or fax.
- **Check** the metadata sheet to make sure that the header information, especially author names and the corresponding affiliations are correctly shown.
- **Check** the questions that may have arisen during copy editing and insert your answers/ corrections.
- **Check** that the text is complete and that all figures, tables and their legends are included. Also check the accuracy of special characters, equations, and electronic supplementary material if applicable. If necessary refer to the *Edited manuscript*.
- The publication of inaccurate data such as dosages and units can have serious consequences. Please take particular care that all such details are correct.
- Please **do not** make changes that involve only matters of style. We have generally introduced forms that follow the journal's style.
  Substantial changes in content, e.g., new results, corrected values, title and authorship are not allowed without the approval of the responsible editor. In such a case, please contact the Editorial Office and return his/her consent together with the proof.
- If we do not receive your corrections **within 48 hours**, we will send you a reminder.
- Your article will be published **Online First** approximately one week after receipt of your corrected proofs. This is the **official first publication** citable with the DOI. **Further changes are, therefore, not possible.**
- The **printed version** will follow in a forthcoming issue.

Please note

After online publication, subscribers (personal/institutional) to this journal will have access to the complete article via the DOI using the URL: http://dx.doi.org/[DOI].
If you would like to know when your article has been published online, take advantage of our free alert service. For registration and further information go to: http://www.link.springer.com.

Due to the electronic nature of the procedure, the manuscript and the original figures will only be returned to you on special request. When you return your corrections, please inform us if you would like to have these documents returned.

Metadata of the article that will be visualized in OnlineFirst

Please note: Images will appear in color online but will be printed in black and white.

ArticleTitle	Novel preconditioners based on quasi–Newton updates for nonlinear conjugate gradient methods	
Article Sub-Title		
Article CopyRight	Springer-Verlag Berlin Heidelberg (This will be the copyright line in the final PDF)	
Journal Name	Optimization Letters	

Corresponding Author	Family Name	**Fasano**
	Particle	
	Given Name	**Giovanni**
	Suffix	
	Division	Department of Management
	Organization	University Ca' Foscari of Venice
	Address	S. Giobbe, Cannaregio 873, 30121, Venice, Italy
	Email	fasano@unive.it
	ORCID	http://orcid.org/0000-0003-4721-8114
Author	Family Name	**Caliciotti**
	Particle	
	Given Name	**Andrea**
	Suffix	
	Division	Dipartimento di Ingegneria Informatica, Automatica e Gestionale "A. Ruberti", SAPIENZA
	Organization	Università di Roma
	Address	via Ariosto, 25, 00185, Rome, Italy
	Email	caliciotti@dis.uniroma1.it
	ORCID	
Author	Family Name	**Roma**
	Particle	
	Given Name	**Massimo**
	Suffix	
	Division	Dipartimento di Ingegneria Informatica, Automatica e Gestionale "A. Ruberti", SAPIENZA
	Organization	Università di Roma
	Address	via Ariosto, 25, 00185, Rome, Italy
	Email	roma@dis.uniroma1.it
	ORCID	

	Received	27 October 2015
Schedule	Revised	
	Accepted	28 June 2016

Abstract

In this paper we study new preconditioners to be used within the nonlinear conjugate gradient (NCG) method, for large scale unconstrained optimization. The rationale behind our proposal draws inspiration from quasi-Newton updates, and its aim is to possibly approximate in some sense the inverse of the Hessian matrix. In particular, at the current iteration of the NCG we consider some preconditioners based on new low-rank quasi-Newton symmetric updating formulae, obtained as by-product of the NCG method at the previous steps. The results of an extensive numerical experience are also reported, showing the effectiveness, the efficiency and the robustness of this approach, which suggests promising guidelines for further studies.

Keywords (separated by '-')	Approximate inverse preconditioners - Preconditioned nonlinear conjugate gradient - Large scale nonconvex optimization - Quasi-Newton updates
Footnote Information	

Optim Lett
DOI 10.1007/s11590-016-1060-2

ORIGINAL PAPER

Novel preconditioners based on quasi–Newton updates for nonlinear conjugate gradient methods

Andrea Caliciotti[1] · **Giovanni Fasano**[2] · **Massimo Roma**[1]

Received: 27 October 2015 / Accepted: 28 June 2016

Abstract In this paper we study new preconditioners to be used within the nonlinear conjugate gradient (NCG) method, for large scale unconstrained optimization. The rationale behind our proposal draws inspiration from quasi-Newton updates, and its aim is to possibly approximate in some sense the inverse of the Hessian matrix. In particular, at the current iteration of the NCG we consider some preconditioners based on new low-rank quasi-Newton symmetric updating formulae, obtained as by-product of the NCG method at the previous steps. The results of an extensive numerical experience are also reported, showing the effectiveness, the efficiency and the robustness of this approach, which suggests promising guidelines for further studies. 1

Keywords Approximate inverse preconditioners · Preconditioned nonlinear conjugate gradient · Large scale nonconvex optimization · Quasi-Newton updates

✉ Giovanni Fasano
fasano@unive.it

Andrea Caliciotti
caliciotti@dis.uniroma1.it

Massimo Roma
roma@dis.uniroma1.it

1 Dipartimento di Ingegneria Informatica, Automatica e Gestionale "A. Ruberti", SAPIENZA, Università di Roma, via Ariosto, 25, 00185 Rome, Italy

2 Department of Management, University Ca' Foscari of Venice, S. Giobbe, Cannaregio 873, 30121 Venice, Italy

Journal: 11590 Article No.: 1060 ☐ TYPESET ☐ DISK ☐ LE ☐ CP Disp.:2016/6/29 Pages: 19 Layout: Small-X

1 Introduction

We deal with the large scale unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x), \tag{1.1}$$

where $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ is a twice continuously differentiable function and n is large. We assume that for a given $x_1 \in \mathbb{R}^n$ the level set

$$\Omega_1 = \left\{ x \in \mathbb{R}^n \;\mid\; f(x) \leq f(x_1) \right\}$$

is compact, but no convexity assumption is considered for the function $f(x)$. A considerably large number of real world applications can be modeled (or reformulated as) an optimization problem of the form (1.1), strongly motivating the interest for the solution of such problems in several contexts.

Among the iterative methods for large scale unconstrained optimization, when the Hessian matrix is possibly dense, the NCG method and Limited Memory quasi-Newton method (e.g. L-BFGS) are often the methods of choice. In their iterations they do not include explicitly second order information; nevertheless, they both exploit the local structure and curvatures of $f(x)$ through the gradient at different iterates.

In this paper we focus on the NCG method and, in particular, on effective techniques to improve it. We highlight that the main aim of the paper is not to define a challenging algorithm for large scale unconstrained optimization, but rather introducing a preconditioning strategy and showing its effectiveness.

As well known (see any textbook, e.g. [23]) the NCG method is a natural extension, to general nonconvex functions, of the linear conjugate gradient (CG) method for quadratic functions. In particular, the NCG method generates the sequence of iterates $x_{k+1} = x_k + \alpha_k p_k$, where p_k is the search direction

$$p_k = -\nabla f(x_k) + \beta_k p_{k-1},$$

with β_k a suitable scalar. The positive steplength α_k is obtained by an appropriate linesearch. Different values of β_k give rise to different algorithms (see [15] for a survey), endowed with different convergence properties. Among them, the most common and historically settled schemes are

- Fletcher and Reeves (FR) [9],
- Polak and Ribière (PR) [24],
- Hestenes and Stiefel (HS) [17].

However, more recently several other efficient proposals have been introduced in the literature, among them we can find for instance

- Hager and Zhang (HZ) [14],
- Dai and Yuan (DY) [3].

The NCG methods have been widely studied and are often very efficient when solving large scale problems. A keynote issue for increasing their efficiency is the use of a preconditioning strategy, especially when solving difficult ill-conditioned problems.

Journal: 11590 Article No.: 1060 TYPESET DISK LE CP Disp.:2016/6/29 Pages: 19 Layout: Small-X

Defining good preconditioners for NCG methods is currently still considered a challenging research topic. On this guideline, this work is devoted to investigate the use of quasi-Newton updates as preconditioners for NCG. In particular, we propose iteratively constructed preconditioners, exploiting information on the inverse of the Hessian matrix at the current iterate. Our proposal is based on quasi-Newton updates of the inverse of the Hessian matrix, and collects also some information from a fixed number of previous iterations. This represents an attempt to improve the efficiency of the NCG method, by conveying information from previous iterates, similarly to a limited memory quasi-Newton approach, so that a preconditioned nonlinear conjugate gradient (PNCG) method can be applied. More in detail, we study new symmetric low-rank updates for defining such preconditioners, where the information used is a by-product of NCG iterates. In this regard it is worth to recall that there exists a close connection between BFGS and NCG [21], and on the other hand, NCG algorithms can be viewed as memoryless quasi-Newton methods (see e.g., [23,25,26]).

Observe that the idea of using a quasi-Newton update, as a preconditioner within NCG algorithms, is not new. In [2], when storage is available, a preconditioner defined by m quasi-Newton updates is used within an NCG algorithm. In [1] a scaled memoryless BFGS matrix is used as preconditioner in the framework of NCG. Moreover, an automatic preconditioning strategy based on a limited memory quasi-Newton update for the linear CG is proposed in [19], within Hessian–free Newton methods, and is extended to the solution of a sequence of linear systems.

The paper is organized as follows: in Sect. 2 some preliminaries on PNCG and quasi-Newton updates are reported. In Sect. 3 we include guidelines for designing our novel preconditioners. Then, Sect. 4 contains our proposal, while Sect. 5 includes an extensive numerical experience, highlighting the benefits from adopting our preconditioners. A section of conclusions also completes the paper. As regards the notation, with $A \succ 0$ [$A \succeq 0$] we indicate that the matrix A is positive definite [semidefinite].

2 Preliminaries

In this section first we report the scheme of a general PNCG algorithm (see e.g. [25]), where $M_k \succ 0$ denotes the preconditioner at the k-th iteration.

Preconditioned nonlinear conjugate gradient (PNCG) algorithm

Step 1: Set $x_1 \in \mathbb{R}^n$ and M_1. Set $p_1 = -M_1 \nabla f(x_1)$ and $k = 1$.

Step 2: Compute the steplength α_k by using a linesearch procedure, which ensures the *strong Wolfe conditions*, and set

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

Step 3: If a *stopping criterion* is satisfied then stop,else compute β_{k+1} and

$$p_{k+1} = -M_{k+1} \nabla f(x_{k+1}) + \beta_{k+1} p_k, \tag{2.2}$$

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

Journal: 11590 Article No.: 1060 TYPESET DISK LE CP Disp.:2016/6/29 Pages: 19 Layout: Small-X

By setting $M_k = I_n$ for any k, the popular (unpreconditioned) NCG method is trivially obtained. The parameter β_{k+1} can be chosen in a variety of ways. For PNCG algorithm, among the most recurrent choices from the literature there are the following ones:

$$\beta_{k+1}^{FR} = \frac{\nabla f(x_{k+1})^T M_k \nabla f(x_{k+1})}{\nabla f(x_k)^T M_k \nabla f(x_k)}, \tag{2.3}$$

$$\beta_{k+1}^{PR} = \frac{\left[\nabla f(x_{k+1}) - \nabla f(x_k)\right]^T M_k \nabla f(x_{k+1})}{\nabla f(x_k)^T M_k \nabla f(x_k)}, \tag{2.4}$$

$$\beta_{k+1}^{HS} = \frac{\left[\nabla f(x_{k+1}) - \nabla f(x_k)\right]^T M_k \nabla f(x_{k+1})}{\left[\nabla f(x_{k+1}) - \nabla f(x_k)\right]^T p_k}, \tag{2.5}$$

which require $M_k \succ 0$. We recall that to guarantee global convergence, an accurate linesearch technique is required to determine the steplength α_k in a PNCG algorithm. The latter fact justifies the use of a linesearch procedure, ensuring the strong Wolfe conditions (see e.g. [23]). This also guarantees that the condition

$$s_k^T y_k > 0, \quad \text{for any } k \tag{2.6}$$

holds, being $s_k = x_{k+1} - x_k$ and $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$. As we will see shortly, (2.6) is a fundamental relation to our purposes.

As already said, preconditioning is applied for increasing the efficiency of the NCG method. In this regard, we remark a noticeable difference between CG and NCG. Whenever the CG is applied, the Hessian matrix does not change during the iterations of the algorithm. On the contrary, when NCG is applied to a general nonlinear function, the Hessian matrix (possibly indefinite) changes with the iterations. The latter fact implies that the mutual conjugacy of the search directions, generated by the NCG, may be hardly fulfilled. In this work our aim is to exploit possible conjugacy among vectors within a quasi-Newton approach, to generate in some sense an approximate inverse of the Hessian matrix. Namely, we want to use the latter approximation as preconditioner within a PNCG framework.

In this regard, as well known (see e.g. [23]), when using quasi-Newton methods in place of (2.2) we generate a search direction of the form

$$p_k = -H_k \nabla f(x_k),$$

where H_k is an approximation of the inverse of the Hessian matrix $\nabla^2 f(x_k)$. Then, as in ***Step 2*** of PNCG, the new iterate x_{k+1} can be obtained according to $x_{k+1} = x_k + \alpha_k p_k$, where α_k is a steplength. In particular, instead of computing H_k from scratch at each iteration k, these methods update H_k in a simple manner, in order to obtain the new approximation H_{k+1} to be used in the next iteration. Moreover, instead of storing full dense $n \times n$ approximations, they only save a few vectors of length n, which allow to represent the approximations $\{H_k\}$ implicitly.

Among the quasi-Newton schemes, the L-BFGS method is usually considered one of the most efficient [18,22]. It is well suited for large scale problems because the

Journal: 11590 Article No.: 1060 TYPESET DISK LE CP Disp.:2016/6/29 Pages: 19 Layout: Small-X

amount of storage it requires is limited and controlled by the user. This method is based on the construction of the approximation of the inverse of the Hessian matrix, by exploiting curvature information gained only from the most recent iterations. Specifically, the inverse of the Hessian matrix is updated by L-BFGS at the k-th iteration as

$$H_{k+1} = V_k^T H_k V_k + \rho_k s_k s_k^T, \tag{2.7}$$

where

$$\rho_k = \frac{1}{y_k^T s_k}, \quad V_k = I_n - \rho_k y_k s_k^T,$$

and

$$s_k = x_{k+1} - x_k = \alpha_k p_k, \quad y_k = \nabla f(x_{k+1}) - \nabla f(x_k). \tag{2.8}$$

Observe that rearranging the expression of H_k we can also iteratively obtain relation

$$\begin{aligned} H_k = & (V_{k-1}^T \cdots V_{k-m}^T) H_k^0 (V_{k-m} \cdots V_{k-1}) \\ & + \rho_{k-m} (V_{k-1}^T \cdots V_{k-m+1}^T) s_{k-m} s_{k-m}^T (V_{k-m+1} \cdots V_{k-1}) \\ & + \rho_{k-m+1} (V_{k-1}^T \cdots V_{k-m+2}^T) s_{k-m+1} s_{k-m+1}^T (V_{k-m+2} \cdots V_{k-1}) \\ & + \cdots \\ & + \rho_{k-1} s_{k-1} s_{k-1}^T, \end{aligned}$$

where m is the *memory* of the method and H_k^0 is an initial approximation of the inverse of the Hessian matrix (see [18,22,23]).

The well known reasons for the success of the L-BFGS method can be summarized in the following two points: firstly, even when m is small, H_{k+1} proves to be an effective approximation of the inverse of the Hessian matrix. Secondly H_{k+1} is the unique (positive definite) matrix which solves the subproblem

$$\begin{aligned} \min_H \ & \|H - H_k\|_F \\ s.t. \ & H = H^T \\ & H y_k = s_k, \end{aligned}$$

where $\|\cdot\|_F$ is the Frobenius norm. Namely, H_{k+1} is the positive definite matrix "closest" to the current approximation H_k, satisfying the *secant equation*

$$H y_k = s_k. \tag{2.9}$$

Relation (2.9) also reveals that when $f(x)$ is quadratic, then $[\nabla^2 f(x_k)]^{-1} y_k = H_k y_k$, meaning that H_k approximates the action of $[\nabla^2 f(x_k)]^{-1}$ along the direction y_k. However, as well known L-BFGS method presents some drawbacks, including the slow convergence on ill-conditioned problems, namely when the eigenvalues of the Hessian matrix are very spread.

As already noted in the Introduction, the idea of using a quasi-Newton update as a preconditioner within both PNCG algorithms and Hessian-free Newton methods is

Journal: 11590 Article No.: 1060 TYPESET DISK LE CP Disp.:2016/6/29 Pages: 19 Layout: Small-X

not new (see also [1,2,19]). Now, to introduce our proposal, let us consider the BFGS updating formula: we want to better exploit the relation with the CG in case $f(x)$ is quadratic, i.e.

$$f(x) = \frac{1}{2}x^T Ax + b^T x, \quad A \in \mathbb{R}^{n\times n}. \tag{2.10}$$

The BFGS update (2.7) can be rewritten as

$$H_k = \left(I_n - \frac{y_{k-1}s_{k-1}^T}{y_{k-1}^T s_{k-1}}\right)^T H_{k-1}\left(I_n - \frac{y_{k-1}s_{k-1}^T}{y_{k-1}^T s_{k-1}}\right) + \frac{s_{k-1}s_{k-1}^T}{y_{k-1}^T s_{k-1}}, \tag{2.11}$$

so that explicitly using the expression of $f(x)$ (see also [13]), which implies $y_k = As_k$, we can set

$$V_k = I_n - \frac{As_k s_k^T}{s_k^T As_k} \tag{2.12}$$

and write recursively

$$\begin{aligned} H_k &= V_{k-1}^T H_{k-1} V_{k-1} + \frac{s_{k-1}s_{k-1}^T}{y_{k-1}^T s_{k-1}} \\ &= V_{k-1}^T (V_{k-2}^T H_{k-2} V_{k-2}) V_{k-1} + V_{k-1}^T \frac{s_{k-2}s_{k-2}^T}{y_{k-2}^T s_{k-2}} V_{k-1} + \frac{s_{k-1}s_{k-1}^T}{y_{k-1}^T s_{k-1}}. \end{aligned} \tag{2.13}$$

Now, since $f(x)$ is quadratic, assuming the conjugacy of the vectors $\{p_1, \ldots, p_k\}$ in (2.8), we have that

$$V_k^T s_{k-1} = \left(I_n - \frac{As_k s_k^T}{s_k^T As_k}\right)^T s_{k-1} = s_{k-1} - \frac{s_k s_k^T As_{k-1}}{s_k^T As_k} = s_{k-1},$$

which implies also that (2.13) becomes

$$\begin{aligned} H_k &= V_{k-1}^T H_{k-1} V_{k-1} + \frac{s_{k-1}s_{k-1}^T}{y_{k-1}^T s_{k-1}} \\ &= V_{k-1}^T (V_{k-2}^T H_{k-2} V_{k-2}) V_{k-1} + \frac{s_{k-2}s_{k-2}^T}{y_{k-2}^T s_{k-2}} + \frac{s_{k-1}s_{k-1}^T}{y_{k-1}^T s_{k-1}} \\ &= V_{k-1}^T V_{k-2}^T \cdots V_1^T H_k^0 V_1 \cdots V_{k-2} V_{k-1} + \sum_{i=1}^{k-1} \frac{s_i s_i^T}{s_i^T As_i}. \end{aligned} \tag{2.14}$$

Formula (2.14) can be used to potentially generate preconditioners for the PNCG, by looking at the rightmost contribution

$$\sum_{i=1}^{k-1} \frac{s_i s_i^T}{s_i^T As_i}, \tag{2.15}$$

Journal: 11590 Article No.: 1060 TYPESET DISK LE CP Disp.:2016/6/29 Pages: 19 Layout: Small-X

whose range is exactly $span\{s_1, \ldots, s_{k-1}\}$. Indeed, we can draw our inspiration from (2.14) and [7], where a new preconditioner for Newton–Krylov methods is described. In particular, in [7] the set of directions generated by a Krylov subspace method is used to provide an approximate inverse preconditioner, for the solution of Newton's systems. On this guideline, observe that for $f(x)$ as in (2.10), with A positive definite, the CG method may generate n conjugate directions $\{p_j\}$ (see e.g. [11]) such that

$$A^{-1} = \sum_{j=1}^{n} \frac{p_j p_j^T}{p_j^T A p_j}. \tag{2.16}$$

This implies that the rightmost contribution in (2.14) might be viewed and used as an approximate inverse of the Hessian matrix A. In the next sections we aim at extending the latter idea, to the case where $f(x)$ is nonlinear, following similar guidelines.

3 Guidelines for a new Symmetric Rank-2 update

In this section we consider a new quasi-Newton updating formula, by considering the properties of a parameter dependent Symmetric Rank-2 (SR2) update of the inverse of the Hessian matrix. Suppose that after k iterations of NCG the sequence of iterates $\{x_1, \ldots, x_{k+1}\}$ is generated. Let us consider the quasi-Newton update H, satisfying the secant equation at the iterates $x_1, \ldots, x_k$, i.e.

$$H y_j = s_j, \quad j \le k. \tag{3.17}$$

Observe that the latter appealing property of the matrix H is satisfied by all the updates of the Broyden class, provided that the linesearch adopted is exact (see e.g. [23]). We would like to recover the motivation underlying the latter class of updates, and by using a novel rank-2 update we would like to define a preconditioner for PNCG.

On this guideline, let the matrix H in (3.17) depend on the three parameters $\{\tau_j\}$, $\{\gamma_j\}$ and $\{\omega_j\}$, and let us consider the update

$$H(\tau_{k+1}, \gamma_{k+1}, \omega_{k+1}) = H(\tau_k, \gamma_k, \omega_k) + \Delta_k, \quad \Delta_k \in \mathbb{R}^{n\times n}, \text{ symmetric}. \tag{3.18}$$

We want (3.18) to represent, to some extent, our quasi-Newton updates of $[\nabla^2 f(x)]^{-1}$, such that:

(0) $H(\tau_{k+1}, \gamma_{k+1}, \omega_{k+1})$ is well-defined and nonsingular;
(1) $H(\tau_{k+1}, \gamma_{k+1}, \omega_{k+1})$ can be iteratively updated;
(2) $H(\tau_{k+1}, \gamma_{k+1}, \omega_{k+1})$ collects the information from the iterations $k-m, k-m+1, \ldots, k$ of a NCG method, where $m < k$ is a given positive integer (memory of the preconditioner);
(3) $H(\tau_{k+1}, \gamma_{k+1}, \omega_{k+1})$ satisfies the secant equation at least at iteration k;
(4) $H(\tau_{k+1}, \gamma_{k+1}, \omega_{k+1})$ "tends to resemble" the inverse of $\nabla^2 f(x_{k+1})$, in case $f(x)$ is a general convex quadratic function and, by suitably setting the three parameters, it can be used as a preconditioner for PNCG, i.e. $M_{k+1} = H(\tau_{k+1}, \gamma_{k+1}, \omega_{k+1}) \succ 0$.

Journal: 11590 Article No.: 1060 TYPESET DISK LE CP Disp.:2016/6/29 Pages: 19 Layout: Small-X

Observe that the Symmetric Rank-1 (SR1) quasi-Newton update (see Section 6.2 in [23]) satisfies properties (1)–(4) but not the property (0), i.e. it might be possibly not well-defined for a general nonlinear function. The latter result follows from the fact that SR1 update provides only a rank-1 quasi-Newton update, unlike BFGS and DFP. On the other hand, while BFGS and DFP quasi-Newton formulae provide only positive definite updates, the SR1 formula is able to recover the inertia of the Hessian matrix, by generating possibly indefinite updates. Thus, now we want to study an SR2 quasi-Newton update, such that at iteration k

– it satisfies (0)–(4);
– at least one of the two newest dyads used for the update is provided using information from iterations $k - m, \ldots, k$ of the NCG method.

4 A preconditioner using a BFGS-like quasi-Newton update

In this section we address the final remark of Sect. 3. Indeed, we introduce a new class of preconditioners which are iteratively constructed by using information from NCG iterations, and satisfy the properties (0)–(4). On this purpose, in order to comply with properties (3) and (4), the preconditioners in our proposal satisfy two prerequisites. First they are conceived around the rightmost term (2.15) in (2.14), in order to possibly approximate the inverse Hessian matrix; then, they satisfy the secant equation at the current iterate, and not necessarily at all the previous iterates. This is a weak theoretical requirement, with respect to other quasi-Newton updates, however numerical results in Sect. 5 yet confirm its efficiency and robustness.

Now, in order to introduce a class of preconditioners for the NCG, suppose we have performed k iterations of the (unpreconditioned) NCG, so that the directions $p_1, \ldots, p_k$ are generated. Let us consider the matrix M_{k+1} defined by

$$M_{k+1} = \tau_k C_k + \gamma_k v_k v_k^T + \omega_k \sum_{j=k-m}^{k} \frac{p_j p_j^T}{p_j^T \nabla^2 f(x_j) p_j}, \tag{4.19}$$

where $0 \le m \le k-1$, $\gamma_k, \omega_k \ge 0$, $\tau_k > 0$, $C_k \in \mathbb{R}^{n \times n}$ is *symmetric positive definite* and $v_k \in \mathbb{R}^n$. In order to use M_{k+1} as a preconditioner in the PNCG, and to update its expression iteratively, we can set $\tau_k C_k = H(\tau_k, \gamma_k, \omega_k)$ (with $H(\tau_0, \gamma_0, \omega_0)$ given) and rewrite (4.19) in the form

$$H(\tau_{k+1}, \gamma_{k+1}, \omega_{k+1}) = H(\tau_k, \gamma_k, \omega_k) + \gamma_k v_k v_k^T + \omega_k \sum_{j=k-m}^{k} \frac{p_j p_j^T}{p_j^T \nabla^2 f(x_j) p_j}. \tag{4.20}$$

$H(\tau_{k+1}, \gamma_{k+1}, \omega_{k+1})$ in (4.20) may be treated as a symmetric quasi-Newton update of the form (3.18). However, for simplicity, in the sequel we prefer to use the more general form given by (4.19). Indeed, as will shortly be evident, relation (4.19) is easier to handle, in order to impose the satisfaction of the secant equation at the current iterate x_k.

Journal: 11590 Article No.: 1060 TYPESET DISK LE CP Disp.:2016/6/29 Pages: 19 Layout: Small-X

Observe that in the expression of M_{k+1} (see (4.19)), $\gamma_k v_k v_k^T$ represents a rank-1 matrix while in view of (2.14)–(2.16), the term

$$\sum_{j=k-m}^{k} \frac{p_j p_j^T}{p_j^T \nabla^2 f(x_j) p_j} \tag{4.21}$$

is aimed at building, in some sense, an approximate inverse of the Hessian matrix on a specific subspace. The next proposition better justifies the last statement.

Proposition 1 *Let $f(x) = 1/2 x^T Ax + b^T x$, with $A \succ 0$. Let $p_1, \ldots, p_n \in \mathbb{R}^n \backslash \{0\}$, with $p_i^T A p_j = 0$, $1 \le i \ne j \le n$. Then, for any $0 \le m \le \min\{n-1, k-1\}$,*

$$\left[\sum_{j=k-m}^{k} \frac{p_j p_j^T}{p_j^T \nabla^2 f(x_j) p_j}\right] Av = v, \quad \textit{for all} \quad v \in span\{p_{k-m}, \ldots, p_k\}.$$

Moreover, when $m = n-1$ then $\sum_{j=k-m}^{k} \frac{p_j p_j^T}{p_j^T \nabla^2 f(x_j) p_j} = A^{-1}$.

Proof Let $v = \sum_{i=k-m}^{k} \mu_i p_i$, $\mu_i \in \mathbb{R}$; then, since $\nabla^2 f(x) = A$, for any $x \in \mathbb{R}^n$, we have

$$\left[\sum_{j=k-m}^{k} \frac{p_j p_j^T}{p_j^T \nabla^2 f(x_j) p_j}\right] Av = \left[\sum_{j=k-m}^{k} \frac{p_j p_j^T}{p_j^T A p_j}\right] Av$$

$$= \sum_{j=k-m}^{k} \sum_{i=k-m}^{k} \mu_i \frac{p_j p_j^T}{p_j^T A p_j} A p_i = \sum_{i=k-m}^{k} \mu_i p_i = v.$$

In case $m = n-1$, since the vectors $\{p_j\}$ are also linearly independent, we directly obtain the inverse matrix A^{-1}. □

Thus, in case $f(x)$ is quadratic, then (4.21) behaves as an inverse of the Hessian matrix on the subspace spanned by the linearly independent vectors $p_{k-m}, \ldots, p_k$.

The integer m can be viewed as a "limited memory" parameter, similarly to the L-BFGS method. Moreover, we can set the matrix C_k, the vector v_k and the parameters $\tau_k, \gamma_k, \omega_k$ such that the class of preconditioners $\{M_k\}$ satisfies, for any k, the secant equation at the current iterate

$$M_{k+1} y_k = s_k, \tag{4.22}$$

along with a *modified secant equation* at some previous iterates, as described in the next proposition.

Proposition 2 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable. Suppose that k iterations of NCG are performed, using a strong Wolfe linesearch procedure. Let $M_{k+1} \in \mathbb{R}^{n \times n}$ be defined as in (4.19), with $0 \le m \le k-1$, $\tau_k > 0$, $\gamma_k, \omega_k \ge 0$.*

(i) *Let $C_k \in \mathbb{R}^{n \times n}$ be symmetric positive definite, then there exist values of $\tau_k, \gamma_k, \omega_k$ such that $M_{k+1} \succ 0$ and (4.22) holds.*

Journal: 11590 Article No.: 1060 TYPESET DISK LE CP Disp.:2016/6/29 Pages: 19 Layout: Small-X

(ii) *Let* $C_k \in \mathbb{R}^{n\times n}$ *be symmetric positive definite and* $f(x) = 1/2x^T Ax + b^T x$. *Then,* $M_{k+1} \succ 0$, (4.22) *holds and* M_{k+1} *reduces to*

$$M_{k+1} = \tau_k C_k + \gamma_k v_k v_k^T + \omega_k \sum_{j=k-m}^{k} \frac{s_j s_j^T}{y_j^T s_j}, \tag{4.23}$$

with $v_k = \sigma_k(s_k - \tau_k C_k y_k - \omega_k s_k)$, $\sigma_k \in \{-1, +1\}$.

(iii) *Let* $f(x) = 1/2x^T Ax + b^T x$, *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* τ_k, γ_k, ω_k, *and a positive semidefinite matrix* C_k, *such that* $M_{k+1} \succ 0$, (4.22) *holds and the following modified secant conditions*

$$M_{k+1} y_i = \omega_k s_i, \quad i = k-m, \ldots, k-1, \tag{4.24}$$

are satisfied.

Proof From (4.19) imposing relation (4.22) we have

$$\tau_k C_k y_k + \gamma_k (v_k^T y_k) v_k + \omega_k \sum_{j=k-m}^{k} \frac{p_j^T y_k}{p_j^T \nabla^2 f(x_j) p_j} p_j = s_k;$$

hence, assuming $\gamma_k (v_k^T y_k) \neq 0$ (which may be straightforwardly guaranteed by a suitable choice of τ_k, γ_k and ω_k),

$$v_k = \sigma_k \left[s_k - \tau_k C_k y_k - \omega_k \sum_{j=k-m}^{k} \frac{p_j^T y_k}{p_j^T \nabla^2 f(x_j) p_j} p_j \right], \tag{4.25}$$

for some $\sigma_k \in \mathbb{R}$. Replacing (4.25) in (4.22) we obtain the equation

$$\begin{aligned}
&\gamma_k \sigma_k^2 \left[s_k^T y_k - \tau_k y_k^T C_k y_k - \omega_k \sum_{j=k-m}^{k} \frac{(p_j^T y_k)^2}{p_j^T \nabla^2 f(x_j) p_j} \right] \\
&\quad \times \left[s_k - \tau_k C_k y_k - \omega_k \sum_{j=k-m}^{k} \frac{p_j^T y_k}{p_j^T \nabla^2 f(x_j) p_j} p_j \right] \\
&= s_k - \tau_k C_k y_k - \omega_k \sum_{j=k-m}^{k} \frac{p_j^T y_k}{p_j^T \nabla^2 f(x_j) p_j} p_j .
\end{aligned}$$

Thus, the following relation among the parameters γ_k, σ_k, τ_k and ω_k has to be satisfied

$$\gamma_k \sigma_k^2 = \frac{1}{s_k^T y_k - \tau_k y_k^T C_k y_k - \omega_k \sum_{j=k-m}^{k} \frac{(p_j^T y_k)^2}{p_j^T \nabla^2 f(x_j) p_j}}, \tag{4.26}$$

where, without loss of generality, we can set $\sigma_k \in \{-1, +1\}$. Then, we remark that the condition (4.26) guarantees the matrix M_{k+1} in (4.22) to satisfy the secant equation only at the k-th iteration (even for quadratic functions), and possibly not at the previous iterates. To complete the proof of item (i), observe that the Wolfe conditions used in the linesearch procedure for computing the steplength α_k ensure that (2.6) holds, i.e. $s_k^T y_k > 0$. Thus, for $\tau_k > 0$ and $\omega_k \geq 0$ sufficiently small in (4.26) we obtain that $\gamma_k > 0$, and the matrix M_{k+1} is positive definite. To prove item (ii), by the Mean Value Theorem we have

$$\int_0^1 s_j^T \nabla^2 f[x_j + \zeta(x_{j+1} - x_j)] s_j \, d\zeta = s_j^T y_j,$$

and using relation $s_j = \alpha_j p_j$ (see (2.8)), in case $f(x)$ is the quadratic function in (2.10), then we have

$$p_j^T A p_j = p_j^T \nabla^2 f(x_j) p_j = \int_0^1 p_j^T \nabla^2 f[x_j + \zeta(x_{j+1} - x_j)] p_j \, d\zeta = \frac{p_j^T y_j}{\alpha_j}, \quad (4.27)$$

which can be replaced in (4.19) to obtain (4.23). Since the Wolfe conditions are used in the linesearch procedure, then (2.6) holds, still implying that

$$\sum_{j=k-m}^{k} \frac{s_j s_j^T}{y_j^T s_j} \succeq 0.$$

In addition, since $y_k = A s_k$, now the expression of v_k in (4.25) reduces to $v_k = \sigma_k (s_k - \tau_k C_k y_k - \omega_k s_k)$.

Finally, as regards (iii), let us define

$$C_k = V_k^T V_{k-1}^T \cdots V_{k-m}^T V_{k-m} \cdots V_{k-1} V_k. \quad (4.28)$$

Even if C_k now is not positive definite, similarly to the proof of (i), we can obtain (4.25) and (4.26). Now, since $y_k = A s_k$, we have $M_{k+1} y_k = s_k$, $v_k = \sigma_k (s_k - \tau_k C_k y_k - \omega_k s_k)$ and

$$\gamma_k \sigma_k^2 = \frac{1}{(1 - \omega_k) s_k^T y_k - \tau_k y_k^T C_k y_k}. \quad (4.29)$$

We prove that the matrix M_{k+1} (which is now the sum of positive semidefinite matrices) is positive definite. Indeed, let $s_1, \ldots, s_n$ be n conjugate (hence linearly independent) directions with respect to matrix $A \succ 0$. Then, recalling that the exact linesearch along with the conjugacy among $\{s_j\}$ yield $\nabla f(x_{j+1}) = \nabla f(x_j) + A s_j$ and

$$(A s_i)^T (A s_j) = 0, \quad \text{for all} \quad |i - j| > 1, \quad (4.30)$$

Journal: 11590 Article No.: 1060 TYPESET DISK LE CP Disp.:2016/6/29 Pages: 19 Layout: Small-X

by (2.12) and for any $\tau_k \neq 0$, $\omega_k \neq 0$ it results

$$\left[\tau_k C_k + \omega_k \sum_{j=k-m}^{k} \frac{s_j s_j^T}{y_j^T s_j}\right] A s_i \neq 0, \quad i = 1, \dots, n.$$

Indeed, the latter result trivially holds for any $i \neq k-m-1$; moreover, for $i = k-m-1$ it also holds, using the relation $V_{k-m}^T(As_{k-m-1}) = As_{k-m-1} \neq 0$. This implies that the matrix $\tau_k C_k + \omega_k \sum_{j=k-m}^{k} s_j s_j^T / y_j^T s_j$ (and consequently M_{k+1}) is nonsingular. Moreover, since $f(x)$ is quadratic, by (4.23) we obtain for $i \in \{k-m, \dots, k\}$

$$M_{k+1} y_i = \left[\tau_k C_k + \gamma_k v_k v_k^T + \omega_k \sum_{j=k-m}^{k} \frac{s_j s_j^T}{y_j^T s_j}\right] y_i$$

$$= \left[\tau_k C_k + \gamma_k v_k v_k^T\right] A s_i + \omega_k s_i.$$

Now, since $v_k = \sigma_k(s_k - \tau_k C_k y_k - \omega_k s_k)$, then we obtain for $i \in \{k-m, \dots, k-1\}$ that $v_k^T A s_i = 0$. Furthermore, by a direct computation we also have for $i \in \{k-m, \dots, k-1\}$

$$C_k A s_i = V_k^T V_{k-1}^T \cdots V_{k-m}^T V_{k-m} \cdots V_{k-1} V_k A s_i = 0;$$

thus, we finally obtain

$$M_{k+1} y_i = \tau_k C_k A s_i + \omega_k s_i = \omega_k s_i, \quad i \in \{k-m, \dots, k-1\}.$$

□

In the next proposition we give some properties about the clustering of the eigenvalues of the preconditioner M_{k+1}.

Proposition 3 *Let $f(x) = 1/2x^T Ax + b^T x$, with $A \succ 0$, and suppose $k \geq 2$ iterations of the NCG algorithm are performed, using an exact linesearch. Consider the matrix C_k in (4.28) and M_{k+1} in (4.23). Then, M_{k+1} has at least $n - (m+2)$ eigenvalues equal to τ_k.*

Proof We first recall that, after some computations, we obtain the relation $V_{k-m}^T(As_{k-m-1}) = As_{k-m-1}$, and by the hypotheses (see also (4.25)), it results $v_k = \sigma_k(s_k - \tau_k C_k y_k - \omega_k s_k)$. Then, recalling (4.30), we have

$$M_{k+1} A s_i = \tau_k A s_i, \quad \text{for} \quad i \leq k-m-1 \quad \text{and} \quad k+2 \leq i \leq n,$$

so that $[k-m-1] + [n-(k+2)+1] = n-(m+2)$ eigenvalues of M_{k+1} are equal to τ_k. □

Observe that the different choices for the parameters τ_k and ω_k in (4.26) provide a different scaling of the matrices C_k and

$$\sum_{j=k-m}^{k} \frac{p_j p_j^T}{p_j^T \nabla^2 f(x_j) p_j}$$

in the preconditioners.

As regards the specific choice of ω_k, τ_k and C_k in (4.23), observe that by (4.24), the choice $\omega_k = 1$ and C_k given by (4.28) seems appealing when $f(x)$ is quadratic. However, with $\omega_k = 1$ in (4.29) γ_k might not be well defined or possibly negative. Also observe that

$$rk(C_k) = rk\left[V_k^T V_{k-1}^T \cdots V_{k-m}^T V_{k-m} \cdots V_{k-1} V_k\right] \le n-1,$$

so that C_k is consequently singular, and when $f(x)$ is non-quadratic the preconditioner M_{k+1} might be singular. To avoid the latter drawback, and possibly reduce the computational burden, while preserving a certain level of efficiency, an obvious choice could be $\omega_k \neq 1$ and

$$C_k = \varepsilon_k I_n, \quad \varepsilon_k \in \mathbb{R}.$$

The parameter ε_k may be computed as the least squares solution of the equation $(\varepsilon I_n) y_k - s_k = 0$, i.e. ε_k solves

$$\min_{\varepsilon} \|(\varepsilon I_n) y_k - s_k\|^2.$$

Hence,

$$\varepsilon_k = \frac{s_k^T y_k}{\|y_k\|^2}$$

so that since $s_k^T y_k > 0$ by the Wolfe conditions, the matrix

$$C_k = \frac{s_k^T y_k}{\|y_k\|^2} I_n \tag{4.31}$$

is positive definite. It is not difficult to verify that the choice (4.31), for C_k, also satisfies the weak secant equation $y_k^T C_k y_k = y_k^T s_k$ (see [4]), at current iterate x_k.

For the sake of clarity we report here the overall resulting expression of our class of preconditioners (4.19), including the choice (4.31) and $\sigma_k = 1$:

$$M_{k+1} = \tau_k \frac{s_k^T y_k}{\|y_k\|^2} I_n + \gamma_k v_k v_k^T + \omega_k \sum_{j=k-m}^{k} \frac{s_j s_j^T}{y_j^T s_j}, \tag{4.32}$$

Journal: 11590 Article No.: 1060 TYPESET DISK LE CP Disp.:2016/6/29 Pages: 19 Layout: Small-X

where

$$v_k = s_k - \tau_k \frac{s_k^T y_k}{\|y_k\|^2} y_k - \omega_k \sum_{j=k-m}^{k} \frac{s_j^T y_k}{y_j^T s_j} s_j, \tag{4.33}$$

$$\gamma_k = \frac{1}{(1-\tau_k) s_k^T y_k - \omega_k \sum_{j=k-m}^{k} \frac{(s_j^T y_k)^2}{y_j^T s_j}}. \tag{4.34}$$

The reader may conjecture that since M_{k+1} merely satisfies, in the convex quadratic case, the interpolation (say secant) conditions (4.22) and (4.24), then its theoretical properties with respect to BFGS are definitely poor. This seems indeed a partially correct conclusion. However, since in practice L-BFGS often performs better than BFGS, we warn the reader that on nonconvex problems the good performance of our proposal in Sect. 5 might not be so surprising. In fact, likewise L-BFGS we retain information from a limited number of previous iterates, mainly relying on the role of the rightmost term in (4.32), as detailed in Proposition 1.

We conclude this section by highlighting that, interestingly enough, similarly to (4.20) we can also construct a class of preconditioners based on DFP-like quasi-Newton updates. Indeed, we can iteratively build the matrices

$$B(\tau_{k+1}, \gamma_{k+1}, \omega_{k+1}),$$

approximating $\nabla^2 f(x)$ instead of its inverse. Then, by the Sherman–Morrison–Woodbury formula applied to $B(\tau_{k+1}, \gamma_{k+1}, \omega_{k+1})$ we can compute a class of preconditioners alternative to $H(\tau_{k+1}, \gamma_{k+1}, \omega_{k+1})$ in (4.20). However, following the current literature which privileges the use of BFGS in place of DFP [23], here we have proposed the class described in (4.32)–(4.34), which performed successfully in practice.

5 Numerical experience

In order to investigate the reliability of the class of preconditioners we have introduced, we performed a wide numerical testing using the preconditioners defined in (4.32). To this purpose, we embedded the preconditioners (4.32) within the standard CG+ code (see [10]), from the literature, available at J. Nocedal's web page. For a fair comparison we used the same stopping criterion

$$\|\nabla f(x_k)\|_\infty \leq 10^{-5}(1 + |f(x_k)|),$$

(namely *original*) and the same linesearch used by default in CG+ code. It is the Moré–Thuente linesearch [20] with a slight modification (we refer the reader to [10] for a complete description of the algorithm). Then, we also tested the robustness of

Journal: 11590 Article No.: 1060 TYPESET DISK LE CP Disp.:2016/6/29 Pages: 19 Layout: Small-X

our proposal using the following different stopping criterion

$$\|\nabla f(x_k)\|_2 \leq 10^{-5} \max\{1, \|x_k\|_2\},$$

(namely *novel*), which is also quite common in the literature.

In particular, we tested both the standard Fletcher and Reeves (FR) and Polak and Ribiere (PR) versions of the PNCG method in Sect. 2. As regards the test problems, we selected all the large scale unconstrained test problems in the CUTEst collection [12]. The dimension of the test problems is between $n = 1000$ and $n = 10{,}000$ (we considered 112 resulting problems). The parameters of the preconditioners (4.32) have been chosen as follows:

$$m = 4, \quad \omega_k = \frac{\frac{1}{2} s_k^T y_k}{y_k^T C_k y_k + \sum_{j=k-m}^{k} \frac{(s_j^T y_k)^2}{s_j^T y_j}}, \quad \tau_k = \omega_k, \quad \gamma_k = \frac{2}{s_k^T y_k},$$

where C_k is given by (4.31), for all k (this choice ensures that, by Wolfe conditions, the denominator of γ_k in (4.34) is positive). As preliminary investigation, we considered the results in terms of the number of iterations and the number of function evaluations, comparing three alternatives:

- M_{k+1} in (4.32), namely *OUR PREC*;
- $M_{k+1} = I$ (unpreconditioned case), namely *UNPREC*;
- M_{k+1} coincident with the L-BFGS update H_{k+1} in (2.7), using a memory of $m = 4$, namely *PREC-LBFGS*.

The overall comparison is reported by using performance profiles [5]. For a fair comparison, we have excluded from each profile all the test problems where the three alternatives do not converge to the same stationary point. Moreover, for $k < 4$ (i.e. in the first three PNCG iterations) we have coherently set $m = \min\{4, k\}$.

We strongly highlight that our proposal (4.32) is built using a dual standpoint with respect to *PREC-LBFGS*. Indeed, our proposal starts by first considering the third matrix in the right hand side of (4.32), in the light of approximating (in the quadratic case) the inverse of the Hessian matrix, as in (2.16). Then, the other two matrices, on the right hand side of (4.32), make our proposal M_{k+1} nonsingular and consistent with a current interpolation condition at iterate k. On the contrary, *PREC-LBFGS* update starts from imposing multiple interpolation conditions at previous iterates (i.e. the secant equations). Then, as by-product it also proves to yield in the quadratic case, after n iterations, the inverse Hessian.

The choice $m = 4$ was in our experience the best compromise over the chosen test set. This should not be surprising if compared with the results in [7,8,19], where the best choice for the memory parameter is either $m = 7$ or $m = 8$. In fact, in the latter papers the preconditioner is built using the CG (or L-BFGS for quadratics) in place of the NCG, which allows to fully exploit the mutual conjugacy among the search directions. On the contrary, in the present paper the NCG is unable to guarantee the latter property, so that the information at iterations $k - m - 1, k - m - 2, \ldots$ for large m risks to be unreliable.

Journal: 11590 Article No.: 1060 TYPESET DISK LE CP Disp.:2016/6/29 Pages: 19 Layout: Small-X

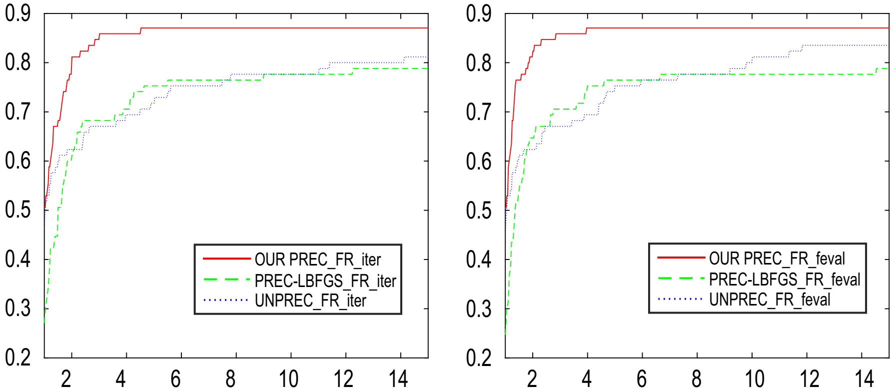

Fig. 1 Profiles using the *original* stopping criterion, adopting FR and with respect to $\#iterations$ (*left*) and $\#function\ evaluations$ (*right*)

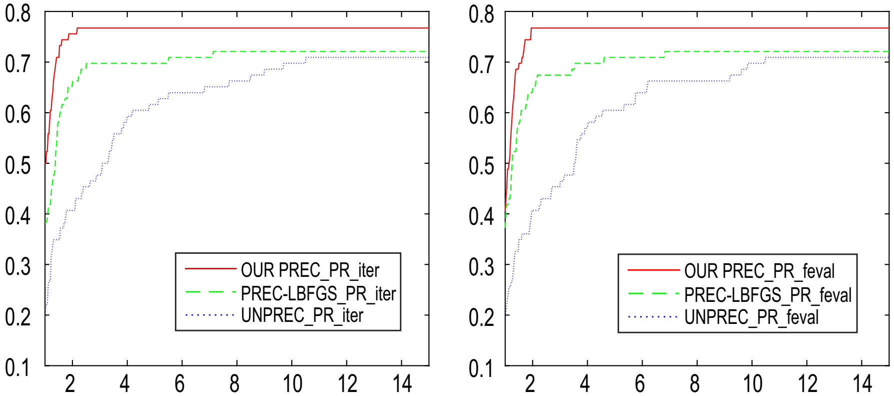

Fig. 2 Profiles using the *original* stopping criterion, adopting PR and with respect to $\#iterations$ (*left*) and $\#function\ evaluations$ (*right*)

As regards the FR version of the PNCG algorithm, in Fig. 1 we report the comparison among the three algorithms. These profiles show that using the FR algorithm and the *original* stopping criterion in CG+ code, our proposal definitely outperforms the competitors, both in terms of number of iterations and number of function evaluations. Now, we turn to the PR version of the PNCG algorithm, and in Fig. 2 we report a similar comparison, obtaining again that our proposal is definitely preferable.

On the other hand, the Figs. 3 and 4 report analogous profiles, where we used the *novel* stopping criterion in place of the *original* one in CG+. Again our preconditioner seems to be the winning strategy.

Finally, we guess that in place of (4.31), a more sophisticated choice of the matrix C_k might be conceived, which possibly summarizes more information on the function at the previous iterates.

As already claimed, the main focus of the paper is not to define a challenging algorithm for large scale unconstrained optimization, but it aims at introducing a

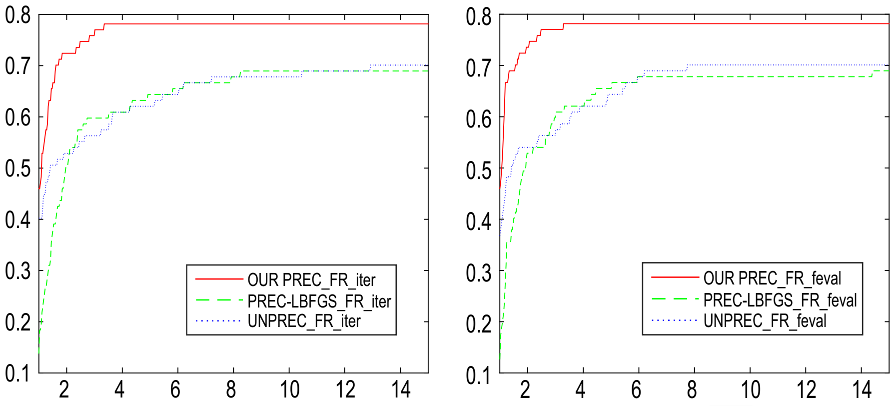

Fig. 3 Profiles using the *novel* stopping criterion, adopting FR and with respect to $\#iterations$ (*left*) and $\#function\ evaluations$ (*right*)

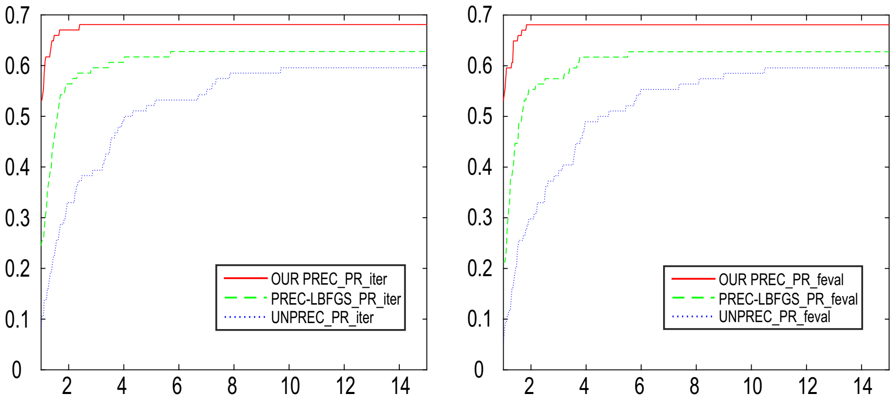

Fig. 4 Profiles using the *novel* stopping criterion, adopting PR and with respect to $\#iterations$ (*left*) and $\#function\ evaluations$ (*right*)

preconditioning strategy and showing its effectiveness. However, for the sake of completeness, in order to have an idea of the overall efficiency of our proposal, we would like to compare our results with those obtained by some benchmark algorithms. For large scale unconstrained optimization, L-BFGS [18,22] and L-CG_DESCENT [14,16] methods are currently considered the most efficient ones.

As regards L-BFGS, the original Fortran code is available in the J. Nocedal's web page. We used this code (denoted by *LBFGS*) in order to perform a comparison with the unpreconditioned (*UNPREC*) and the preconditioned (*OUR PREC*) version of PNCG algorithm. In particular, in our codes we use the FR version. Note that L-BFGS adopts the original Moré–Thuente linesearch [20], without the slight modification introduced in CG+. Therefore, for a fair comparison, we here used the original Moré–Thuente linesearch also in our codes *OUR PREC* and *UNPREC*. The profiles reporting this comparison are in Fig. 5. We can see that our proposal using FR seems enough

Journal: 11590 Article No.: 1060 TYPESET DISK LE CP Disp.:2016/6/29 Pages: 19 Layout: Small-X

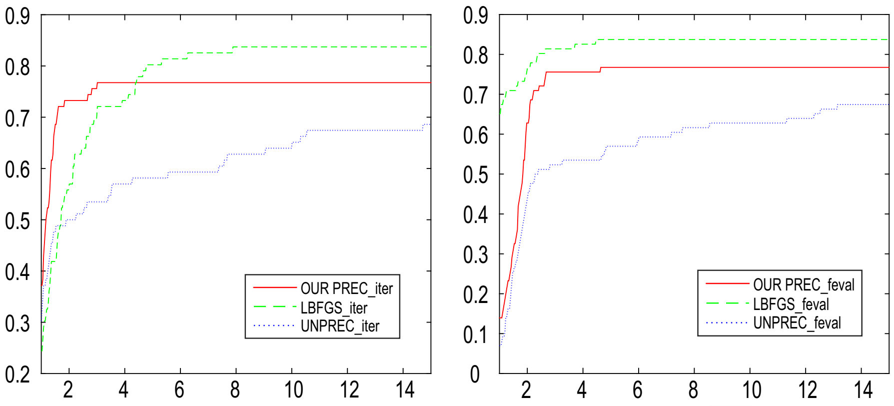

Fig. 5 Comparison between OUR PREC (FR) and L-BFGS. Profiles using the *novel* stopping criterion, with respect to $\#iterations$ (*left*) and $\#function\ evaluations$ (*right*)

competitive in terms of number of iterations. On the other hand, considering the number of function evaluations, we can observe that the search direction we compute does not seem yet well scaled. This indicates that future refinements on our preconditioners are possibly necessary.

As regards L-CG_DESCENT, the most recent version available in the W. Hager's web page is the L-CG_DESCENT 6.8 code. It is written in C, uses an hybrid version of β_k coefficient and a different linesearch expressly designed by the authors (see [14]), more efficient and accurate than the Moré–Thuente one. At present, this possibly makes unfair any comparison between our codes and L-CG_DESCENT. Anyway, embedding our preconditioner in L-CG_DESCENT 6.8 would be an interesting further numerical experiment.

6 Conclusions and future work

In this paper we have proposed a novel class of quasi-Newton updates, to be used as possible preconditioners within PNCG method. In our proposal, namely the satisfaction of the secant equation only at the current iteration is ensured, and the resulting update is guaranteed to be positive definite. Furthermore, our class of preconditioners also satisfies the theoretical properties in Sects. 3 and 4. We numerically tested the latter approach versus both the unpreconditioned case and an L-BFGS based preconditioning approach. The results obtained showed that the preconditioners we propose are definitely much efficient and robust in optimization frameworks. At this stage of the research we still urge to experience our preconditioners also on tough and significant real applications, where specific "pathologies" may be expected. Moreover, we think that our proposal may be possibly exploited also for solving difficult nonconvex problems where the fast iterative computation of negative curvatures for the function is a fruitful ingredient (see e.g. [6]).

Journal: 11590 Article No.: 1060 TYPESET DISK LE CP Disp.:2016/6/29 Pages: 19 Layout: Small-X

Acknowledgments G. Fasano thanks the National Research Council-Marine Technology Research Institute (CNR-INSEAN), for the indirect support in project RITMARE 2012–2016. The authors wish to thank the anonymous referees for the helpful comments and suggestions which led to improve the paper.

References

1. Andrei, N.: Scaled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization. Optim. Methods Softw. **22**, 561–571 (2007)
2. Buckley, B., Lenir, A.: QN-like variable storage conjugate gradients. Math. Program. **27**, 155–175 (1983)
3. Dai, Y., Yuan, Y.: A nonlinear conjugate gradient method with a strong global convergence property. SIAM J. Optim. **10**, 177–182 (1999)
4. Dennis, J., Wolkowicz, H.: Sizing and least-change secant methods. SIAM J. Numer. Anal. **30**, 1291–1314 (1993)
5. Dolan, E.D., Moré, J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–213 (2002)
6. Fasano, G., Roma, M.: Iterative computation of negative curvature directions in large scale optimization. Comput. Optim. Appl. **38**, 81–104 (2007)
7. Fasano, G., Roma, M.: Preconditioning Newton–Krylov methods in nonconvex large scale optimization. Comput. Optim. Appl. **56**, 253–290 (2013)
8. Fasano, G., Roma, M.: A novel class of approximate inverse preconditioners for large positive definite systems. Comput. Optim. Appl. Online first (2015). doi:10.1007/s10589-015-9765-1
9. Fletcher, R., Reeves, C.: Function minimization by conjugate gradients. Comput. J. **7**, 149–154 (1964)
10. Gilbert, J., Nocedal, J.: Global convergence properties of conjugate gradient methods for optimization. SIAM J. Optim. **2**, 21–42 (1992)
11. Golub, G., Van Loan, C.: Matrix Computations, 3rd edn. The John Hopkins Press, Baltimore (1996)
12. Gould, N.I.M., Orban, D., Toint, P.L.: CUTEst: a constrained and unconstrained testing environment with safe threads. Comput. Optim. Appl. **60**, 545–557 (2015)
13. Gratton, S., Sartenaer, A., Tshimanga, J.: On a class of limited memory preconditioners for large scale linear systems with multiple right-hand sides. SIAM J. Optim. **21**, 912–935 (2011)
14. Hager, W., Zhang, H.: A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM J. Optim. **16**, 170–192 (2005)
15. Hager, W., Zhang, H.: A survey of nonlinear conjugate gradient methods. Pac. J. Optim. **2**, 35–58 (2006)
16. Hager, W., Zhang, H.: The limited memory conjugate gradient method. SIAM J. Optim. **23**, 2150–2168 (2013)
17. Hestenes, M., Stiefel, E.: Methods of conjugate gradients for solving linear systems. J. Res. Natl. Bur. Stand. **49**, 409–436 (1952)
18. Liu, D., Nocedal, J.: On the limited memory BFGS method for large scale optimization. Math. Program. **45**, 503–528 (1989)
19. Morales, J., Nocedal, J.: Automatic preconditioning by limited memory quasi-Newton updating. SIAM J. Optim. **10**, 1079–1096 (2000)
20. Moré, J., Thuente, D.: Line search algorithms with guaranteed sufficient decrease. ACM Trans. Math. Softw. (TOMS) **20**, 286–307 (1994)
21. Nazareth, L.: A relationship between the BFGS and conjugate gradient algorithms and its implications for new algorithms. SIAM J. Numer. Anal. **16**, 794–800 (1979)
22. Nocedal, J.: Updating quasi-Newton matrices with limited storage. Math. Comput. **35**, 773–782 (1980)
23. Nocedal, J., Wright, S.: Numerical Optimization, 2nd edn. Springer, Berlin (2006) 2
24. Polak, E., Ribiere, G.: Note sur la convergence de methodes de directions conjuguees. Revue Francaise d'Informatique et de Recherche Operationnelle, serie rouge, tome **3**(1), 35–43 (1969)
25. Pytlak, R.: Conjugate Gradient Algorithms in Nonconvex Optimization. Springer, Berlin (2009)
26. Shanno, D.: Conjugate gradient methods with inexact searches. Math. Oper. Res. **3**, 244–256 (1978)

Journal: **11590** Article No.: **1060** ☐ TYPESET ☐ DISK ☐ LE ☐ CP Disp.:**2016/6/29** Pages: **19** Layout: **Small-X**

Journal: 11590
Article: 1060

Springer
the language of science

Author Query Form

Please ensure you fill out your response to the queries raised below and return this form along with your corrections

Dear Author

During the process of typesetting your article, the following queries have arisen. Please check your typeset proof carefully against the queries listed below and mark the necessary changes either directly on the proof/online grid or in the 'Author's response' area provided below

Query	Details required	Author's response
1.	Kindly check and approve the edit made in the article title.	
2.	Kindly check the publisher location for the reference [23].	