Metadata of the chapter that will be visualized in SpringerLink

Book Title	Numerical Analysis and Optimization	
Series Title		
Chapter Title	Quasi-Newton-Based Preconditioning and Damped Quasi-Newton Schemes for Nonlinear Conjugate Gradient Methods	
Copyright Year	2018	
Copyright HolderName	Springer International Publishing AG, part of Springer Nature	
Author	Family Name	**Al-Baali**
	Particle	
	Given Name	**Mehiddin**
	Prefix	
	Suffix	
	Role	
	Division	Department of Mathematics and Statistics
	Organization	Sultan Qaboos University
	Address	Box 36, Muscat, 123, Oman
	Email	albaali@squ.edu.om
Author	Family Name	**Caliciotti**
	Particle	
	Given Name	**Andrea**
	Prefix	
	Suffix	
	Role	
	Division	Dipartimento di Ingegneria Informatica
	Organization	Automatica e Gestionale 'A. Ruberti', SAPIENZA Università di
	Address	Roma, Italy
	Email	caliciotti@dis.uniroma1.it
Corresponding Author	Family Name	**Fasano**
	Particle	
	Given Name	**Giovanni**
	Prefix	
	Suffix	
	Role	
	Division	Department of Management
	Organization	University Ca' Foscari
	Address	Venice, Italy
	Email	fasano@unive.it
Author	Family Name	**Roma**
	Particle	
	Given Name	**Massimo**
	Prefix	
	Suffix	

Role	
Division	Dipartimento di Ingegneria Informatica
Organization	Automatica e Gestionale ‘A. Ruberti’, SAPIENZA Università di
Address	Roma, Italy
Email	roma@dis.uniroma1.it

Abstract

In this paper, we deal with matrix-free preconditioners for nonlinear conjugate gradient (NCG) methods. In particular, we review proposals based on quasi-Newton updates, and either satisfying the secant equation or a secant-like equation at some of the previous iterates. Conditions are given proving that, in some sense, the proposed preconditioners also approximate the inverse of the Hessian matrix. In particular, the structure of the preconditioners depends both on low-rank updates along with some specific parameters. The low-rank updates are obtained as by-product of NCG iterations. Moreover, we consider the possibility to embed *damped techniques* within a class of preconditioners based on quasi-Newton updates. Damped methods have proved to be effective to enhance the performance of quasi-Newton updates, in those cases where the Wolfe linesearch conditions are hardly fulfilled. The purpose is to extend the idea behind damped methods also to improve NCG schemes, following a novel line of research in the literature. The results, which summarize an extended numerical experience using large-scale CUTEst problems, is reported, showing that these approaches can considerably improve the performance of NCG methods.As per style, this chapter requires Keywords. Please provide keywords.

Quasi-Newton-Based Preconditioning and Damped Quasi-Newton Schemes for Nonlinear Conjugate Gradient Methods

Mehiddin Al-Baali, Andrea Caliciotti, Giovanni Fasano and Massimo Roma

Abstract In this paper, we deal with matrix-free preconditioners for nonlinear conjugate gradient (NCG) methods. In particular, we review proposals based on quasi-Newton updates, and either satisfying the secant equation or a secant-like equation at some of the previous iterates. Conditions are given proving that, in some sense, the proposed preconditioners also approximate the inverse of the Hessian matrix. In particular, the structure of the preconditioners depends both on low-rank updates along with some specific parameters. The low-rank updates are obtained as by-product of NCG iterations. Moreover, we consider the possibility to embed *damped techniques* within a class of preconditioners based on quasi-Newton updates. Damped methods have proved to be effective to enhance the performance of quasi-Newton updates, in those cases where the Wolfe linesearch conditions are hardly fulfilled. The purpose is to extend the idea behind damped methods also to improve NCG schemes, following a novel line of research in the literature. The results, which summarize an extended numerical experience using large-scale CUTEst problems, is reported, showing that these approaches can considerably improve the performance of NCG methods.As per style, this chapter requires Keywords. Please provide keywords.

AQ1 AQ2

M. Al-Baali
Department of Mathematics and Statistics, Sultan Qaboos University,
Box 36, Muscat 123, Oman
e-mail: albaali@squ.edu.om

A. Caliciotti · M. Roma
Dipartimento di Ingegneria Informatica, Automatica e Gestionale 'A. Ruberti',
SAPIENZA Università di, Roma, Italy
e-mail: caliciotti@dis.uniroma1.it

M. Roma
e-mail: roma@dis.uniroma1.it

G. Fasano (✉)
Department of Management, University Ca' Foscari, Venice, Italy
e-mail: fasano@unive.it

M. Al-Baali et al. (eds.), *Numerical Analysis and Optimization*,
Springer Proceedings in Mathematics & Statistics 235,
https://doi.org/10.1007/978-3-319-90026-1_1

449842_1_En_1_Chapter ☑ TYPESET ☐ DISK ☐ LE ☑ CP Disp.:12/4/2018 Pages: 22 Layout: T1-Standard

1 Introduction

Several iterative methods were proposed in the literature, for the solution of the large-scale unconstrained optimization problem $\min_{x\in\mathbb{R}^n} f(x)$, where $f:\mathbb{R}^n\to\mathbb{R}$ (see, e.g., [1–6]). Among them, the nonlinear conjugate gradient (NCG) along with quasi-Newton methods is undoubtedly the most commonly used. Indeed, they both prove to be actually effective in practice and are endowed with a mature theory, including strong convergence properties.

On this purpose, let us first consider a general iterative preconditioned nonlinear conjugate gradient (PNCG) method, which generates a sequence of iterates $\{x_k\}$. Essentially, three choices at current step k strongly affect both the effectiveness and the efficiency of the overall method. The first choice refers to the adopted *linesearch procedure*, along with the selected steplength $\alpha_k > 0$ used to give the next iterate x_{k+1}, being

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

where p_k is the search direction. The second choice refers to the selection of the parameter β_k, which is responsible for the computation of the next search direction, being

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

where $p_1 = -g_1$ and g_k denotes $\nabla f(x_k)$. In the case where the function $f(x)$ is non-quadratic, different expressions for the parameter β_k in the latter formula may yield significantly different (preconditioned) NCG schemes. In particular, among the first classic choices in the literature for the parameter β, we have the proposals by Fletcher and Reeves (FR) [7], Polak and Ribière (PR) [8], Hestenes and Stiefel (HS) [9]. More modern and efficient schemes have also been studied. In particular, we urge to mention the proposals in the seminal papers [10] and [3, 4], since they raised novel ideas which have inspired several advances in the last decade. Recently, Neculai (see [11] and therein references) reported an efficient version of the NCG method, which promises to outperform the proposal in [4]. This gives room to further improvements in the latest literature (see also [5]), where some appealing properties of L-BFGS update are exploited in the context of NCG, with the purpose of improving efficiency. The latter research area has also partially inspired the results reported in the current paper.

The third proper choice for the symmetric positive definite preconditioner $M_{k+1} \in \mathbb{R}^{n\times n}$ often plays a keynote role for the computation of p_{k+1}, being

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

where β_k may depend on M_k and M_{k+1} and $p_1 = -M_1 g_1$. Of course, the latter three choices are not independent. Indeed, an inaccurate linesearch procedure turns to be harmful and may require a large number of function and gradient evaluations. Similarly, a careless choice of the preconditioner risks to possibly destroy both convergence properties and numerical performance of the PNCG. These observations impose a specific attention before selecting a preconditioner.

In the first part of this paper, we review some preconditioners for NCG, which are based on the satisfaction of a *secant-based* equation (see [12–14] for details). Our main purpose here is to show that imposing the satisfaction of the secant condition surely represents an important guideline to gain second-order information about the objective function. However, on highly nonlinear functions, when the distance among the last iterates increases, the satisfaction of the secant equation at any iterate might represent a tight request, which does not necessarily enhance the information on second-order information. On the contrary, in [12] the approximation of an average Hessian matrix is built by using an initial guess suggested by the quadratic case. Then, the initial guess is refined imposing some *secant-like* conditions, which are used to set accordingly some parameters.

We remark that the preconditioners are iteratively constructed and based on satisfying either the secant or a modified secant equation and partially recover the structure of quasi-Newton updates. On the overall, our proposals for preconditioners comply with the next specifications:

- do not rely on the structure of the minimization problem in hand;
- are matrix-free, and 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 method.

We urge to recall that the idea of using a quasi-Newton update as a possible preconditioner, within the NCG algorithms, is not new; examples of such an approach can be found for instance in [15, 16] or in the more recent proposal [17]. In particular, the efficient framework in [17] explicitly exploits the relation between the conjugate gradient method and BFGS quasi-Newton approach, in the quadratic case.

In the second part of the paper, we show how to combine damped techniques with preconditioning strategies, as introduced in [18]. Taking inspiration from [19–21], two different damping strategies are proposed. In particular, we focus on the Polak–Ribière (PR) (recently, Polak–Ribière–Polyak (PRP)) method, showing that, under reasonable assumptions, the damped and preconditioned version of this method (denoted by D-PR-PNCG), can be able to efficiently tackle also difficult problems. This is confirmed by the results of an extensive numerical testing reported (see [18] for details).

Under mild assumptions, the proposals in this paper preserve convergence properties for the PNCG method.

As regards the notations, we denote for an n-real vector x, the Euclidean norm by $\|x\|$. Moreover, for a symmetric matrix A, $A \succ 0$ indicates that A is positive definite.

1.1 Preconditioned Nonlinear Conjugate Gradient (PNCG) Method

Here, we first recall a general scheme of PNCG algorithm. In the following scheme, $M_k \in \mathbb{R}^{n\times n}$ 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 the *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 along with the preconditioner $M_{k+1} \succ 0$. Compute a search direction by

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

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

Of course, in case $M_k = I$ for all k, the PNCG scheme reduces to the NCG method. Also, observe that as an alternative, in order to possibly improve the efficiency of NCG by introducing preconditioning strategies, 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 along with the preconditioner M_{k+1}. If $M_{k+1} \not\succ 0$ or $M_{k+1} g_{k+1} = 0$ then set $M_{k+1} = I$. Compute the search direction

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

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

The steplength α_k and the parameter β_k can be chosen in a variety of ways. In particular, in order to prove global convergence properties, a Wolfe-type linesearch procedure seems mandatory, while to improve the overall efficiency, several values for β_k have appeared in the literature (see also Sect. 1). Here, we neither intend to propose a novel choice of β_k, nor we want to consider any specific linesearch procedure to compute α_k for the PNCG algorithm. In this regard, the Wolfe conditions are well-suited for our purposes, inasmuch as under mild assumptions they guarantee the fulfillment of the usual curvature condition

$$s_k^T y_k > 0,$$

being $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$. On the other hand, we strongly remark the importance of the positive definiteness for preconditioners, in order to prove global convergence results.

2 Quasi-Newton Updates for Preconditioning

In this section, we suitably exploit some quasi-Newton updates in order to build preconditioners. As well known (see, e.g., [1]), when using quasi-Newton methods in place of (1), at iteration k, we generate a search direction of the form

$$p_k = -H_k g_k,$$

where H_k represents an approximation of the inverse Hessian matrix $[\nabla^2 f(x_k)]^{-1}$. 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 as above is a steplength computed by a Wolfe-type procedure. In particular, instead of computing H_k from scratch at each iteration k, quasi-Newton methods update H_k in a simple manner by means of adding a small number of rank one matrices, 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 definitely considered one of the most efficient methods, and the amount of storage it requires can be controlled by the user throughout setting the *limited memory* parameter. This method is based on the construction of the approximation of the inverse Hessian matrix, by exploiting curvature information gained only from the most recent iterations. Specifically, H_{k-1} is updated by BFGS at the kth iteration as

$$H_k = V_{k-1}^T H_{k-1} V_{k-1} + \rho_{k-1} s_{k-1} s_{k-1}^T, \tag{2}$$

where

$$\rho_{k-1} = \frac{1}{s_{k-1}^T y_{k-1}}, \qquad V_{k-1} = I - \rho_{k-1} y_{k-1} s_{k-1}^T.$$

In case $f(x)$ is quadratic, i.e., $f(x) = \frac{1}{2} x^T A x + b^T x$, $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$, then we have explicitly $V_{k-1} = I - A s_{k-1} s_{k-1}^T / s_{k-1}^T A s_{k-1}$ and the following lemma holds.

Lemma 1 *Let us consider the quadratic function $f(x) = \frac{1}{2} x^T A x + b^T x$ with $A \succ 0$. Suppose the steplength α_k in **Step 2** of PNCG is computed using an exact linesearch procedure. Given the expression of H_k in (2), along with $H_k \succ 0$ and the positions*

$$\rho_i = \frac{1}{s_i^T y_i}, \quad s_i^T y_i \neq 0, \quad i = 1, \ldots, k,$$

$$V_i = I - \rho_i y_i s_i^T, \qquad i = 1, \ldots, k,$$

then we have

$$H_k = V_{k-1}^T V_{k-2}^T \cdots V_1^T H_k^1 V_1 \cdots V_{k-2} V_{k-1} + \sum_{i=1}^{k-1} \frac{s_i s_i^T}{s_i^T A s_i}, \tag{3}$$

where $H_k^1 \succ 0$ is given (usually, a multiple of the unit matrix).

Proof First observe that since $f(x)$ is quadratic, then $y_i = As_i$, $i = 1, \ldots, k$, and the vectors $s_1, \ldots, s_k$ are mutually conjugate, i.e., $s_i^T A s_j = 0$, for any $1 \le i \ne j \le k$. We prove (3) by complete induction.
When $k = 2$, by (2), we explicitly obtain

$$H_2 = V_1^T H_k^1 V_1 + \rho_1 s_1 s_1^T = V_1^T H_k^1 V_1 + \frac{s_1 s_1^T}{s_1^T A s_1}.$$

Now, assume (3) holds for some $k-1$, and we prove (3) for the index k as follows. Recalling the conjugacy among vectors $\{s_i\}$ yields

$$V_{k-1}^T s_i = \left(I - \frac{s_{k-1} y_{k-1}^T}{s_{k-1}^T A s_{k-1}} \right) s_i = s_i, \qquad i = 1, \ldots, k-2,$$

by (2), we immediately have after some computations

$$\begin{aligned} H_k &= V_{k-1}^T H_{k-1} V_{k-1} + \frac{s_{k-1} s_{k-1}^T}{s_{k-1}^T A s_{k-1}} \\ &= V_{k-1}^T V_{k-2}^T \cdots V_1^T H_k^1 V_1 \cdots V_{k-2} V_{k-1} + \sum_{i=1}^{k-1} \frac{s_i s_i^T}{s_i^T A s_i}. \end{aligned}$$

□

Note that Formula (3) for the quadratic case can suggest iterative updates to generate preconditioners for PNCG. Indeed, drawing inspiration from (3) and [22], in case $f(x)$ is quadratic (i.e., NCG coincides with the conjugate gradient method), we have

$$A^{-1} = \sum_{j=1}^{n} \frac{s_j s_j^T}{s_j^T A s_j}. \tag{4}$$

In view of (4), the rightmost contribution in (3) may represent an approximate inverse of the Hessian matrix A up to the kth iteration. As an extension, we can borrow the last idea also in case $f(x)$ is a general nonlinear function, in order to generate possible preconditioners which approximate the rightmost matrix in (3). In particular, in this regard, we will have to assess a couple of issues:

(a) We have to set a finite number of NCG iterations $m \le n$, which are necessary to build the approximation of the rightmost matrix in (3).

(b) We have to explicitly indicate how to approximately compute the quantities $s_i^T A s_i$, for $i \geq 1$, in (3); indeed, unlike in the quadratic case, when $f(x)$ is a general nonlinear function, the quantity $s_i^T A s_i$ is unavailable at iteration i.

3 Preconditioners Based on the BFGS Update: First Proposal

In this section, we review the preconditioners for PNCG proposed in [12], which exploits the contents of Sect. 2. We now report the general expression of this class of preconditioners.

$$M_{k+1} = \tau_k C_k + \gamma_k v_k v_k^T + \omega_k \sum_{j=k-m}^{k} \frac{s_j s_j^T}{y_j^T s_j}, \tag{5}$$

where $C_k \in \mathbb{R}^{n\times n}$, $v_k \in \mathbb{R}^n$, $\tau_k, \gamma_k, \omega_k \in \mathbb{R}$ and m is positive integer. Here, we consider

$$C_k = \frac{s_k^T y_k}{\|y_k\|^2} I, \qquad \tau_k = \omega_k, \qquad \gamma_k = \frac{2}{s_k^T y_k},$$

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

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

and $m \ll n$, $0 \leq m \leq k-1$. For further motivations along with the rationale behind this proposal, we refer to [12]. In the sequel, we report the main theoretical results and a summary of the numerical experience.

Observe that the right-hand side of (5) includes three contributions. More specifically, the rightmost matrix represents an approximate inverse Hessian, as in the guidelines of the conclusions of Sect. 2. In particular, exploiting the mean value theorem, we can write

$$y_j = g_{j+1} - g_j = \int_0^1 \nabla^2 f(x_j + t s_j)^T s_j dt, \qquad j \geq 1,$$

so that assuming $\nabla^2 f(z) = A_j$ constant for $z \in [x_j, x_{j+1}]$, we have

$$y_j^T s_j = \int_0^1 s_j^T \nabla^2 f(x_j + t s_j)^T s_j dt \approx s_j^T A_j s_j, \qquad j \geq 1,$$

showing that the issue (b), at the end of Sect. 2, can be easily treated. Moreover, the integer m in (5) represents a *memory* and guarantees that complying with (a), information from only the lattermost m iterations is collected.

A few comments need also be added, with respect to the role played by the matrix C_k and the parameter τ_k in (5). C_k is chosen similar to the matrix $H_k^1 = \lambda_k I$, where λ_k is the solution of the subproblem

$$\min_{\lambda} \|(\lambda I) y_k - s_k\|^2.$$

In other words, $\lambda_k = y_k^T s_k / \|y_k\|^2$ is a value of the parameter λ which aims at approximately solving the initial secant equation $(\lambda I) y_k = s_k$. As usual, the use of the Wolfe conditions ensures that $\lambda_k > 0$.

On the other hand, the exact role played by the parameter τ_k in (5) is a bit more technical and is in particular related to eigenvalue clustering for the preconditioner M_{k+1}, as highlighted in the next Theorem (see also Proposition 3 in [12]).

Theorem 1 *Let $f(x) = 1/2 x^T A x + b^T x$, with $A \succ 0$, and assume that*

- *$k \geq 2$ iterations of the NCG algorithm are performed.*
- *an exact linesearch procedure is adopted.*
- *M_{k+1} is defined as in (5) with $m \leq n - 2$.*

Then, at least $n - (m + 2)$ eigenvalues of M_{k+1} coincide with τ_k.

As detailed in [12], the next proposition can be proved for the update (5), showing its well-posedness and the satisfaction of some secant-like conditions.

Proposition 1 *Let f be twice continuously differentiable. Suppose that k iterations of NCG are performed, using the strong Wolfe linesearch procedure. Let M_{k+1} be defined as in (5), with $0 \leq m \leq k - 1$, $\tau_k > 0$ and $\gamma_k, \omega_k \geq 0$.*

(i) Let $C_k \in \mathbb{R}^{n \times n}$ be symmetric positive definite, then there exist values of τ_k, γ_k, ω_k such that $M_{k+1} \succ 0$ and the secant equation $M_{k+1} y_k = s_k$ is satisfied.

(ii) Let $f(x) = 1/2 x^T A x + b^T x$, with $A \succ 0$. 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 $M_{k+1} \succ 0$. Moreover, $M_{k+1} y_k = s_k$ and the modified secant conditions

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

are satisfied.

Before reporting other proposals for possible preconditioners in PNCG, we highlight the role played by the vector v_k in (5). In particular, the value of v_k is set in such a way that M_{k+1} satisfies the secant equation $M_{k+1} y_k = s_k$ (at iteration k). In this

regard, the computation of vector v_k follows a similar guideline with respect to the idea adopted by SR1 quasi-Newton update (see also [1] for details).

As a preliminary numerical experience which reveals the performance of the proposal M_{k+1} in (5), the preconditioner M_{k+1} has been embedded in PNCG, with $m = \min\{4, k-1\}$ and β_k computed as in the Polak–Ribière (PR) (recently, Polak–Ribière–Polyak (PRP)) formula

$$\beta_k = \frac{\left[g_{k+1} - g_k\right]^T M_{k+1} g_{k+1}}{g_k^T M_k g_k}.$$

In [12], the resulting PR-PNCG has been experienced over a set of 112 large-scale problems of CUTEst collection [23]. This proposal (5) (namely OUR PREC_PR) is compared with the L-BFGS update (setting the memory parameter $m = 4$) (namely PREC-LBFGS_PR), used as a preconditioner, and with the unpreconditioned NCG scheme (namely UNPREC_PR). Results are reported in Figs. 1 and 2, in terms of # iterations and # of function evaluations. Note that the steplength α_k is computed such that the strong Wolfe conditions

$$f_{k+1} \leq f_k + c_1 \alpha_k g_k^T p_k,$$

and

$$|g_{k+1}^T p_k| \leq c_2 |g_k^T p_k|,$$

where $0 < c_1 < 0.5$ and $c_1 < c_2 < 1$, hold (we used as for the code CG+, $c_1 = 0.0001$ and $c_2 = 0.9$). We also remark that in Fig. 1, the original stopping criterion of the code CG+ (see [24]), i.e., $\|g_k\|_\infty \leq 10^{-5}(1 + |f_k|)$, is adopted, while in Fig. 2, the more common criterion from the literature

$$\|g_k\| \leq 10^{-5} \max\{1, \|x_k\|\} \tag{6}$$

is used, showing the effectiveness and efficiency of our first proposal (5).

4 Preconditioners Based on the BFGS Update: Second Proposal

As second proposal for a possible preconditioning strategy, which again exploits the contents in Sect. 2, we have the following update for M_{k+1} in PNCG scheme as proposed in [14].

$$M_{k+1} = \delta_k M_k + \gamma_k v_k v_k^T + \omega_k \frac{p_k p_k^T}{y_k^T p_k}, \qquad \delta_k > 0, \tag{7}$$

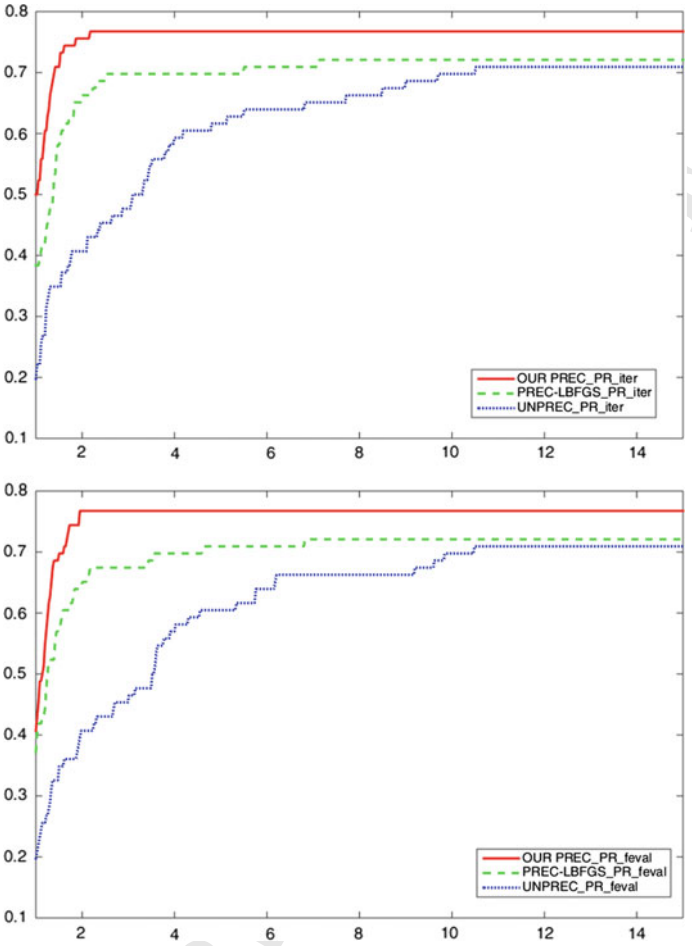

Fig. 1 Performance profiles using the *original* stopping criterion $\|g_k\|_\infty \leq 10^{-5}(1 + |f_k|)$ in the code CG+ [24], adopting PR and with respect to # $iterations$ (up) and # $function\ evaluations$ (down)

with $\gamma_k, \omega_k \in \mathbb{R} \setminus \{0\}$, and where, given M_k and the vector p_k generated by NCG, we have for v_k the expression

$$v_k = \sigma_k \left(s_k - \delta_k M_k y_k - \omega_k p_k\right), \qquad \sigma_k \in \{-1, +1\}.$$

The proposal in (7) follows a different strategy with respect to (5), inasmuch as it more closely attempts to emulate quasi-Newton updates. Indeed, similar to (5) also in (7) M_{k+1} includes three contributions, being the rightmost term $\omega_k p_k p_k^T / y_k^T p_k$ built using information collected at iteration k of the NCG method, and the leftmost term $\delta_k M_k$ being representative of the preconditioner at the previous iteration. Finally, the

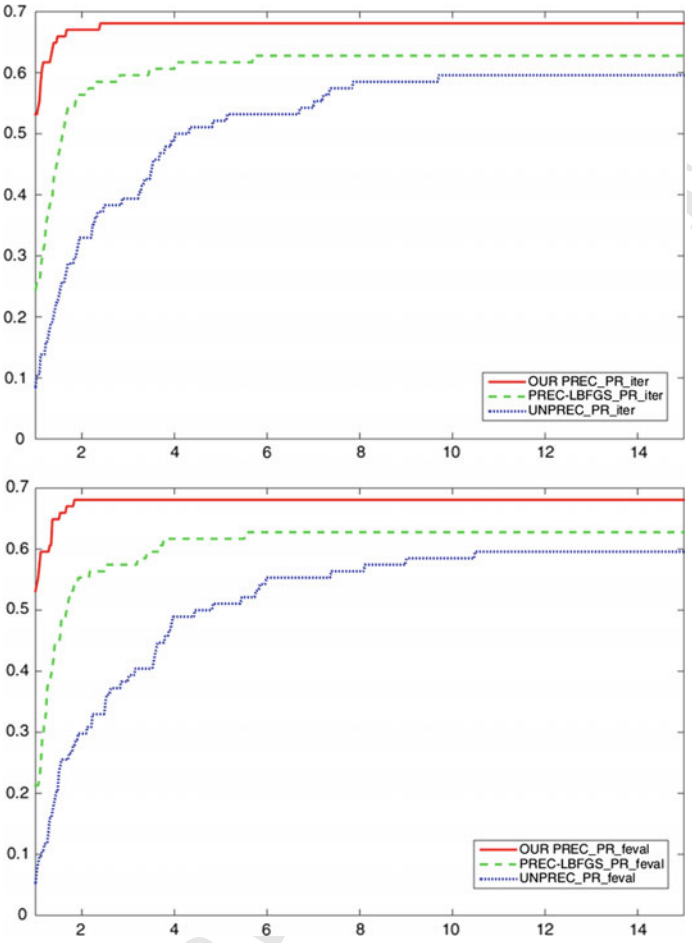

Fig. 2 Profiles using the *novel* stopping criterion (6), adopting PR and with respect to # *iterations* (up) and # *function evaluations* (down)

term $\gamma_k v_k v_k^T$ in (7) is introduced so that M_{k+1} can explicitly satisfy the secant equation $M_{k+1} y_k = s_k$. The latter considerations confirm that, similar to BFGS update, the dyad $\omega_k p_k p_k^T / y_k^T p_k$ aims at adding the most recent information from NCG to our current preconditioner.

The next couple of theoretical results can also be proved for the proposal (7), confirming to what extent (7) closely resembles quasi-Newton approaches (see [14] for details).

Proposition 2 *Let $f(x) = \frac{1}{2} x^T A x - b^T x$, where A is a symmetric matrix. Suppose k steps of the NCG method are performed, adopting an exact linesearch procedure (which imposes $\nabla f(x_{j+1})^T p_j = 0,\ j = 1, \ldots, k$), in order to detect the stationary point (if any) of the function f. Then, the matrix M_{k+1} in (7) satisfies the modified*

secant equations

$$\begin{cases} M_{k+1} y_j = \delta_j s_j, \qquad \delta_j > 0, \qquad j = 1, \ldots, k-1, \\ M_{k+1} y_k = s_k, \end{cases} \tag{8}$$

provided that the nonzero coefficients γ_j, ω_j, $j = 1, \ldots, k$ *are chosen such that*

$$\begin{cases} \gamma_j = \dfrac{1}{s_j^T y_j - \delta_j y_j^T M_j y_j - \omega_j p_j^T y_j}, \ j = 1, \ldots, k, \\ \omega_j \neq \dfrac{s_j^T y_j - \delta_j y_j^T M_j y_j}{p_j^T y_j}, \qquad j = 1, \ldots, k. \end{cases} \tag{9}$$

Proposition 2 reveals to what extent the matrix M_{k+1} substantially summarizes some second-order information on the objective function $f(x)$. In particular, by (8), the secant equation at the current iterate x_k is fulfilled, while a weaker condition holds at the previous iterates, being possibly $\delta_j \neq 1$, for $j = 1, \ldots, k-1$. Also, note that the choice of the parameters $\{\delta_j\}$, $\{\gamma_j\}$ and $\{\omega_j\}$ in Proposition 2 does not ensure in general the positive definiteness of M_{k+1}. Indeed, pre-multiplying the second relation in (8) by y_k, we obtain $y_k^T M_{k+1} y_k = y_k^T s_k$, where the right-hand side might be possibly negative, inasmuch as no Wolfe conditions were adopted in Proposition 2 when applying the NCG. On this guideline, the next result helps recover the positive definiteness of the preconditioner M_{k+1} (see [14]).

Proposition 3 *Let* f *be a continuously differentiable function. Suppose that the NCG method is used to minimize the function* f. *Suppose that* $s_k^T y_k > 0$, $M_k \succ 0$, $\varepsilon_k \in (0, 1)$ *and*

$$\begin{aligned} 0 < \delta_k &= (1 - \varepsilon_k) \frac{s_k^T y_k}{y_k^T M_k y_k}, \\ 0 < \omega_k &< \varepsilon_k \alpha_k, \\ 0 < \gamma_k &= \frac{1}{(\varepsilon_k \alpha_k - \omega_k) p_k^T y_k}. \end{aligned}$$

Then conditions (8)–(9) hold and $M_{k+1} \succ 0$ *in (7).*

By Proposition 3, a suitable interval of values for δ_k, γ_k, and ω_k always exists such that (8)–(9) hold and $M_{k+1} \succ 0$, even though an inexact linesearch procedure is adopted (but not necessary the Wolfe linesearch procedure). Moreover, the hypothesis $M_k \succ 0$ might be too restrictive to our purposes, and we can easily prove that what really matters is the weaker condition $y_k^T M_k y_k > 0$ along with the inequality $y_k^T s_k > 0$.

By Proposition 2, we have also a remarkable result in case the objective function $f(x)$ is quadratic. Indeed, after n steps, the matrix M_{n+1} retains information on the

inertia of the Hessian matrix, as in the next corollary (see [14]), where $\lambda_m(\cdot)$ and $\lambda_M(\cdot)$ represent, respectively, the smallest and the largest eigenvalue.

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

(i) If (8)–(9) hold, we have

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

with

$$D = diag\{\delta_1, \delta_2, \ldots, \delta_{n-1}, 1\}.$$

(ii) It results

$$\lambda_m(M_{n+1}A) = \lambda_m(D), \qquad \lambda_M(M_{n+1}A) = \lambda_M(D). \tag{10}$$

Several interesting conclusions arise considering the two proposals in Sects. 3 and 4 for M_{k+1}; we urge to carry out the following observations, which are also the result of a deeper investigation not reported here:

- Both the proposals for the preconditioner M_{k+1} are based on the attempt to emulate the BFGS update, in order to possibly benefit from some of its well-known features (i.e., the satisfaction of the secant equation and BFGS attitude to approximate the inverse Hessian in the quadratic case).
- while the scheme in (5) details an update based on $m+1$ pairs (s_j, y_j), $j = k-m, \ldots, k$, provided by the NCG method, the scheme in (7) simply relies on the pair (p_k, y_k) generated at step k of the NCG method.
- the proposal in (7) seems to be endowed with stronger theoretical properties with respect to (5). As also shown in the next sections, the latter fact is also reflected in an appreciable enhancement of numerical performance, over a significant large-scale test set. Indeed, comparing the proposals in Sects. 3 and 4, over the same test set specified in Sect. 3, we obtain the performance profiles in Fig. 3, using (6) for termination which is the same as that used for obtaining Fig. 2.

5 Damped Strategies for NCG Preconditioning

Damped techniques were introduced in the framework of quasi-Newton methods, and their rationale can be summarized as follows. As is well known (see, e.g., [1]), when dealing with the BFGS update, a crucial issue in order to guarantee the positive definiteness of the updated Hessian approximation is the satisfaction of the curvature condition

$$s_k^T y_k > 0. \tag{11}$$

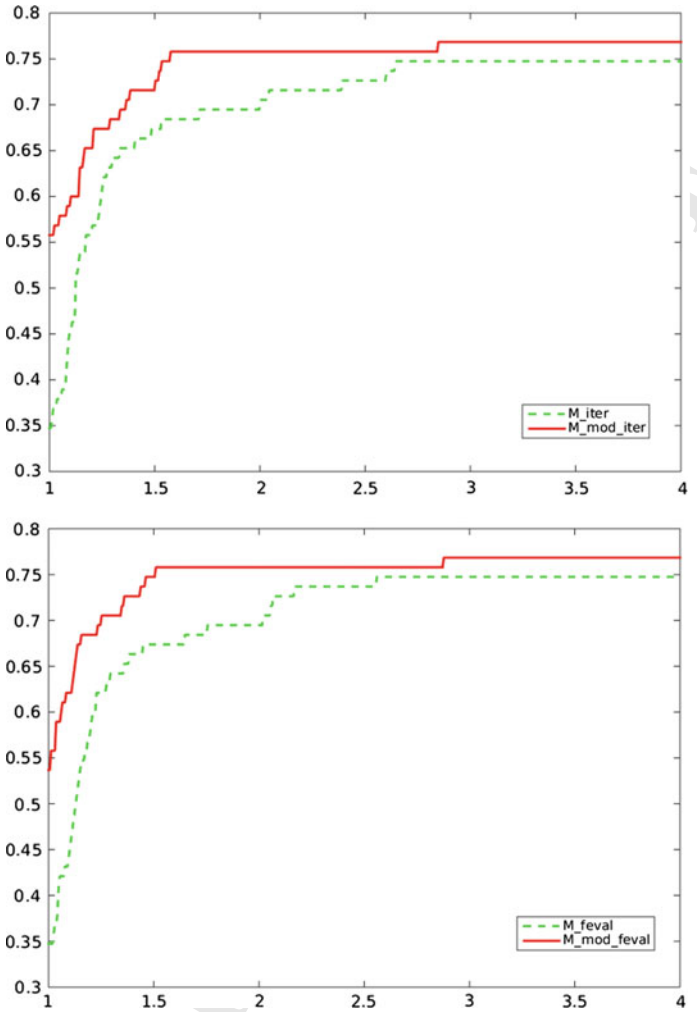

Fig. 3 Comparison between the proposal of preconditioner in (5) (namely M, *dash line*) and the proposal in (7) (namely M_mod, *solid line*), using the stopping criterion (6). Profiles with respect to # *iterations* (up) and # *function evaluations* (down)

In case f is strongly convex, then (11) holds for any pair of points x_k and x_{k+1} (see, e.g. [25]). In case of nonconvex functions, imposing the satisfaction of condition (11) requires a proper choice of the stepsize α_k, from the linesearch procedure adopted. Indeed, in principle, the satisfaction of (11) can always be obtained by a suitable linesearch procedure, provided that the objective function is bounded below. To this aim, as mentioned above, the Wolfe conditions (in practice, the strong Wolfe conditions) are usually adopted, which ensure the fulfillment of condition (11). However, for sufficiently large value of c_2, the value of $s_k^T y_k$ may not be sufficiently positive. In

addition, if only the backtracking linesearch framework is employed, the curvature condition (11) may not hold.

A possible successful strategy to cope with the last issue is to adopt the *damped technique* proposed by Powell in [19], in the context of SQP Lagrangian BFGS methods for constrained optimization and applied for the first time by Al-Baali [26] to unconstrained optimization. In [19], the author proposes to modify the difference of the gradients vector y_k in (11), before performing the BFGS update. Namely, if B_k denotes the current BFGS positive definite Hessian approximation at kth iteration, the following modified (damped) vector is used in place of y_k:

$$\widehat{y}_k = \varphi_k y_k + (1 - \varphi_k) B_k s_k, \tag{12}$$

where φ_k is chosen in $(0, 1]$ such that $s_k^T \widehat{y}_k$ is "sufficiently positive." The latter fact guarantees that the use of the damped vector $\hat{y}_k$ is in principle preferable with respect to y_k. In particular, given $\sigma \in (0, 1]$, the value of the parameter φ_k is often set according with the rule:

$$\varphi_k = \begin{cases} \dfrac{\sigma s_k^T B_k s_k}{s_k^T B_k s_k - s_k^T y_k}, & \text{if} \quad s_k^T y_k < (1 - \sigma) s_k^T B_k s_k, \\ 1, & \text{otherwise}, \end{cases} \tag{13}$$

which for $\sigma = 0.8$ yields that in Sect. 18.3 in [1]. There are several reasons which motivate (13), including the fact that by this choice we have

$$s_k^T \widehat{y}_k = (1 - \sigma) s_k^T B_k s_k, \tag{14}$$

i.e., the quantity $s_k^T \widehat{y}_k$ is sufficiently positive, inasmuch as B_k is positive definite. Al-Baali suggests using the modified damped vector (12) with (13) for unconstrained optimization and extended it to

$$\varphi_k = \begin{cases} \dfrac{\sigma_1 s_k^T B_k s_k}{s_k^T B_k s_k - s_k^T y_k}, & \text{if} \quad s_k^T y_k < (1 - \sigma_1) s_k^T B_k s_k, \\ \dfrac{\sigma_2 s_k^T B_k s_k}{s_k^T B_k s_k - s_k^T y_k}, & \text{if } s_k^T y_k > (1 + \sigma_2) s_k^T B_k s_k, \\ 1, & \text{otherwise}, \end{cases} \tag{15}$$

where $\sigma_1 \in (0, 1]$ and $\sigma_2 \geq 2$. Note that the value $\sigma_2 = \infty$ reduces choice (15) to (13).

In [18], in order to extend the definition of the damped vector $\widehat{y}_k$ in (12), a novel vector $\widehat{y}_k$ is defined as a combination of the original vector y_k and an appropriate vector z_k, namely

$$\widehat{y}_k = \varphi_k y_k + (1-\varphi_k) z_k, \tag{16}$$

(see also [27]). The vector z_k plays a noteworthy role to ensure that $s_k^T \widehat{y}_k$ is *sufficiently positive*, for suitable values of $\varphi_k \in (0, 1]$. Of course, a key point of this approach is an appropriate choice of z_k. Two choices for z_k have been proposed in [18].

The first proposal corresponds to set $z_k = \eta_k s_k$, where $\eta_k > 0$, based on approximating B_k by $\eta_k I$. This choice originates from the idea of using $z_k = A_{k+1} y_k$ in (16), where A_{k+1} is a positive definitive approximation of the inverse Hessian. In particular, $B_k \approx \eta_k I$ satisfies the *modified secant equation*

$$A_{k+1} y_k = \eta_k s_k.$$

Hence, by using the latter equation, we can set

$$\widehat{y}_k^{(a)} = \varphi_k y_k + (1-\varphi_k)\eta_k s_k. \tag{17}$$

Interesting properties of (17) are that it does not require the explicit knowledge of the approximate inverse Hessian matrix A_{k+1} and that

$$s_k^T \widehat{y}_k^{(a)} = (1-\sigma_1)\eta_k \|s_k\|^2 > 0, \tag{18}$$

for appropriate choice of the parameter in (16). This condition may be of great interest if we consider a geometric interpretation of the curvature condition (11). Indeed, since for the vector $\hat{y}_k^{(a)}$ condition (18) is satisfied, it means that $s_k^T \hat{y}_k^{(a)}$ is always *sufficiently positive*. Moreover, it can be easily proved that for proper choices of the parameters η_k and σ, we obtain (as long as (11) holds)

$$s_k^T \widehat{y}_k^{(a)} \geq s_k^T \widehat{y}_k. \tag{19}$$

Furthermore, also in case (11) does not hold, by relation (18), we immediately infer that again (19) holds.

The second proposal corresponds to set in (16) $z_k = -\alpha_k g_k$, so that the novel damped vector becomes

$$\widehat{y}_k^{(b)} = \varphi_k y_k - (1-\varphi_k)\alpha_k g_k. \tag{20}$$

This choice of z_k comes from the following observation: If $B_k \succ 0$ is an approximation of the Hessian and we consider $-B_k^{-1} g_k$ as search direction, it immediately follows that

$$s_k = x_{k+1} - x_k = -\alpha_k B_k^{-1} g_k,$$

which implies

$$B_k s_k = -\alpha_k g_k.$$

This formula allows to compute the original damped vector (12) without explicitly using the matrix B_k. Indeed, it suffices to replace $B_k s_k$ with $-\alpha_k g_k$ in (12), according with (20).
Similar to the choice $\widehat{y}_k^{(a)}$, also for $\widehat{y}_k^{(b)}$ in (20), we can guarantee that $s_k^T \widehat{y}_k^{(b)}$ is sufficiently positive. In fact, we immediately have from (14)

$$s_k^T \widehat{y}_k^{(b)} = -\alpha_k(1-\sigma_1) s_k^T g_k = -\alpha_k^2(1-\sigma_1) p_k^T g_k > 0,$$

where the last inequality holds because p_k is a descent direction. Several theoretical properties can be proved for the choices (17) and (20) (see also [18]). Some of them are summarized here below, where we assume that the coefficient β_k in PNCG is replaced by the PR–type 'damped coefficient'

$$\hat{\beta}_k^{PR} = \frac{\left(\widehat{y}_k^{(a)}\right)^T M_{k+1} g_{k+1}}{g_k^T M_k g_k}$$

(the resulting PNCG scheme, with $\widehat{y}_k^{(a)}$ in place of y_k will be addressed as D-PR-PNCG).

Assumption 1 (*see* [18])

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

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

(c) There exist $\lambda > 0$ and $\Lambda > 0$ such that the preconditioner $M(x)$, for any $x \in \mathscr{B}_r$, is positive definite with the smallest [largest] eigenvalue $\lambda_m(M(x))$ [$\lambda_M(M(x))$] satisfying

$$0 < \lambda \leq \lambda_m(M(x)) \leq \lambda_M(M(x)) < \Lambda.$$

Proposition 4 *Let $\{x_k\}$ be an infinite sequence (with $g_k \neq 0$) generated by the D-PR-PNCG method, where the steplength $\alpha_k > 0$ is determined by a linesearch procedure such that, for all k, the following conditions hold:*

(i) $x_k \in \mathscr{L}_1$ *for all* k;
(ii) $\lim_{k\to+\infty} \dfrac{|g_k^T p_k|}{\|p_k\|} = 0$;
(iii) $\lim_{k\to+\infty} \alpha_k \|p_k\| = 0$.

If Assumption 1 holds, then

$$\liminf_{k\to+\infty} \|g_k\| = 0$$

and hence there exists at least a stationary limit point of $\{x_k\}$.

Similar to the proposals in Sects. 3 and 4, we consider now a brief numerical experience on the use of the damped vectors in (17) and (20). A complete study can be found in [18]. Observe that in principle the use of damped techniques fully affects the preconditioning strategies (where y_k is replaced by $\hat{y}_k^{(a)}$ or $\hat{y}_k^{(b)}$), i.e., both the value of β_k along with the preconditioner, and not just the value of β_k. However, our

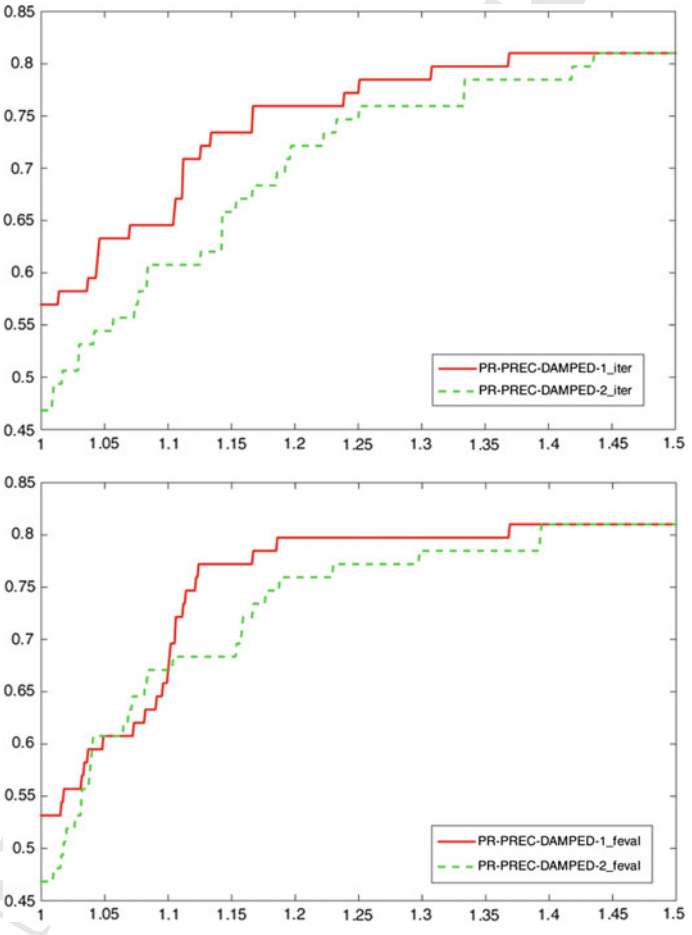

Fig. 4 Comparison between the adoption of the two damped strategies in (17) and in (20). Profiles with respect to # *iterations* (up) and # *function and gradient evaluations* (down)

preliminary aim here is to report a numerical experience with PNCG (and not D-PR-PNCG), i.e., embedding the damped techniques within the preconditioner used in a PNCG scheme, where the standard Polak–Ribière (PR) formula for β_k is used. In particular, the same settings used in Sects. 3 and 4, along with the same test set are considered. We also recall that a standard implementation of the PNCG method in CG+ code was adopted (see [24]), where the preconditioner (5) is included, and the linesearch technique is the same as that in [28]. Finally, the stopping criterion adopted is the standard one in (6). We also recall that in the linesearch procedure adopted in [28] the number of function and gradient evaluations coincide. In Fig. 4, the two damped strategies in (17) (with $\eta_k = 4$ and φ_k chosen as in (13)) and in (20)

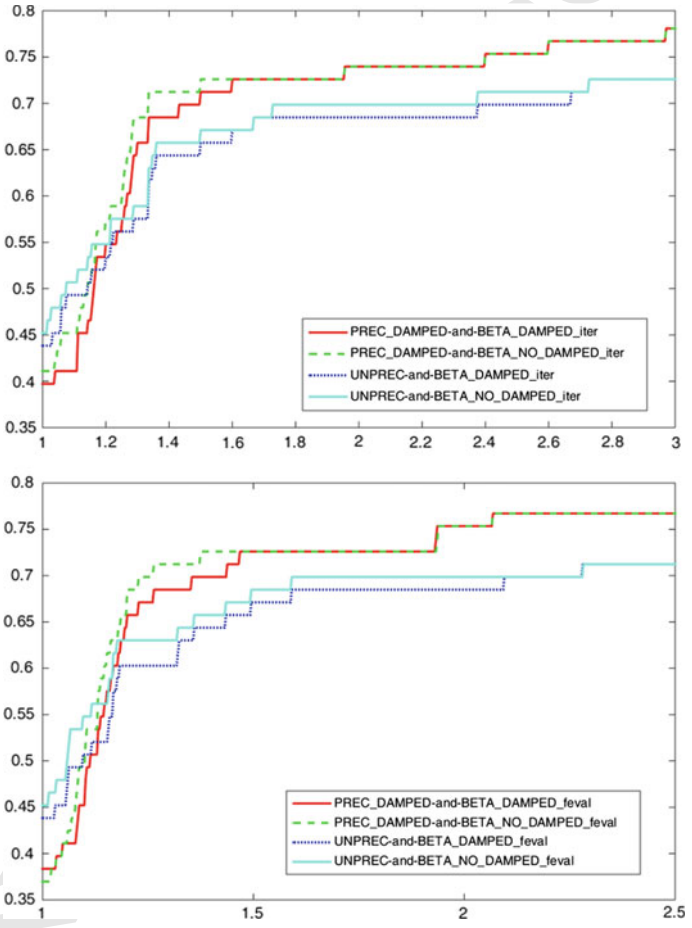

Fig. 5 Comparison between the use $\widehat{\beta}_k^{PR}$ (setting $\widehat{y}_k = \widehat{y}_k^{(a)}$) and β_k^{PR}, in both preconditioned and unpreconditioned cases. Profiles with respect to # *iterations* (up) and # *function and gradient evaluations* (down)

(with φ_k chosen as in (13)) are compared, with respect to both # iterations and # function evaluations. The strategy (17) seems to be somehow preferable to (20).

To complete our analysis, we note that a full information from damped techniques can be used, both affecting the computation of the coefficient β_k and the preconditioner M_{k+1} in PNCG (see [18]). More explicitly, the performances of PNCG vs. D-PR-PNCG (where $\hat{\beta}_k^{PR}$ is used in place of β_k^{PR}) in both the preconditioned and unpreconditioned case are compared. The corresponding results are summarized in Fig. 5 (names of the schemes are self-explanatory). As it can be observed from the profiles, the use of $\hat{\beta}_k^{PR}$ does not yield a noteworthy improvement. Nevertheless, we also observe that the D-PR-PNCG scheme, which also uses $\hat{\beta}_k^{PR}$, reveals to outperform the standard NCG method.

Acknowledgements The research is partially supported by the Italian Flagship Project RITMARE, coordinated by the Italian National Research Council and funded by the Italian Ministry of Education, University and Research.

References

1. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer, New York (2006)
2. Conn, A.R., Gould, N.I.M., Toint, PhL: Trust region methods. MOS-SIAM Series on Optimization, Philadelphia (2000)
3. Hager, W., Zhang, H.: A survey of nonlinear conjugate gradient methods. Pac. J. Optim. **2**, 35–58 (2006)
4. Hager, W., Zhang, H.: A new conjugate gradient method with guaranteed descent and efficient line search. SIAM J. Optim. **16**, 170–192 (2005)
5. Hager, W., Zhang, H.: The limited memory conjugate gradient method. SIAM J. Optim. **23**, 2150–2168 (2013)
6. Raydan, M.: The Barzilai and Borwein gradient method for large scale unconstrained minimization problems. SIAM J. Optim. **7**, 26–33 (1997)
7. Fletcher, R., Reeves, C.: Function minimization by conjugate gradients. Comput. J. **7**, 149–154 (1964)
8. Polak, E., Ribière, G.: Note sur la convergence de methodes de directions conjuguees. Rev. Fr. d'Informatique Rech. Operationnelle serie rouge **3**(1), 35–43 (1969)
9. Hestenes, M., Stiefel, E.: Methods of conjugate gradients for solving linear systems. J. Res. Natl. Bur. Stand. **49**, 409–436 (1952)
10. Dai, Y., Yuan, Y.: A nonlinear conjugate gradient method with a strong global convergence property. SIAM J. Optim. **10**, 177–182 (1999)
11. Andrei, N.: An adaptive conjugate gradient algorithm for large-scale unconstrained optimization. J. Comput. Appl. Math. **292**, 83–91 (2016)
12. Caliciotti, A., Fasano, G., Roma, M.: Novel preconditioners based on quasi-Newton updates for nonlinear conjugate gradient methods. Optim. Lett. **11**, 835–853 (2017)
13. Caliciotti, A., Fasano, G., Roma, M.: Preconditioning strategies for nonlinear conjugate gradient methods, based on quasi-Newton updates. AIP Conf. Proc. Am. Inst. Phys. [Sergeyev Y.D., Kvasov D.E., Dell'Accio F., Mukhametzhanov M.S. (eds.)] **1776**(090007):1–4 (2016)
14. Caliciotti, A., Fasano, G., Roma, M.: Preconditioned nonlinear Conjugate Gradient methods based on a modified secant equation. Appl. Math. Comput. **318**, 196–214 (2018)
15. Andrei, N.: Scaled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization. Optim. Methods Softw. **22**, 561–571 (2007)

16. Morales, J.L., Nocedal, J.: Automatic preconditioning by limited memory quasi-Newton updating. SIAM J. Optim. **10**, 1079–1096 (2000)
17. 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)
18. Al-Baali, M., Caliciotti, A., Fasano, G., Roma, M.: Exploiting damped techniques for nonlinear conjugate gradient methods, to appear on Mathematical Methods of Operations Research (2017). https://doi.org/10.1007/s00186-017-0593-1
19. Powell, M.J.D.: Algorithms for nonlinear constraints that use Lagrangian functions. Math. Program. **14**, 224–248 (1978)
20. Al-Baali, M., Grandinetti, L.: On practical modifications of the quasi-Newton BFGS methods. AMO Adv. Model. Optim. **11**, 63–76 (2009)
21. Al-Baali, M., Grandinetti, L., Pisacane, O.: Damped techniques for the limited memory BFGS method for large-scale optimization. J. Optim. Theory Appl. **161**, 688–699 (2014)
22. Fasano, G., Roma, M.: Preconditioning Newton-Krylov methods in nonconvex large scale optimization. Comput. Optim. Appl. **56**, 253–290 (2013)
23. Gould, N.I.M., Orban, D., Toint, PhL: CUTEst: a constrained and unconstrained testing environment with safe threads. Comput. Optim. Appl. **60**, 545–557 (2015)
24. Gilbert, J.C., Nocedal, J.: Global convergence properties of conjugate gradient methods for optimization. SIAM J. Optim. **2**, 21–42 (1992)
25. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, New York (2004)
26. Al-Baali, M.: Damped techniques for enforcing convergence of quasi-Newton methods. Optim. Methods Softw. **29**, 919–936 (2014)
27. Al-Baali, M.: Quasi-Newton algorithms for large-scale nonlinear least-squares. In: Di Pillo, G., Murli, A. (eds.) High Performance Algorithms and Software for Nonlinear Optimization, pp. 1–21. Kluwer Academic, Dordrecht (2003)
28. Moré, J., Thuente, D.: Line search algorithms with guaranteed sufficient decrease. ACM Trans. Math. Softw. (TOMS) **20**, 286–307 (1994)

Author Queries

Chapter 1

Query Refs.	Details Required	Author's response
AQ1	Please confirm if the corresponding author is correctly identified. Amend if necessary.	CONFIRMED
AQ2	As per style, this chapter requires Keywords. Please provide key-words.	ADDED KEYWORDS
AQ3	Please check and approve the edit made in the chapter title.	APPROVED

MARKED PROOF

Please correct and return this set

Please use the proof correction marks shown below for all alterations and corrections. If you wish to return your proof by fax you should ensure that all amendments are written clearly in dark ink and are made well within the page margins.

Instruction to printer	*Textual mark*	*Marginal mark*
Leave unchanged	··· under matter to remain	New matter followed by
Insert in text the matter indicated in the margin		or (x2)
Delete	through single character, rule or underline or through all characters to be deleted	or (x2)
Substitute character or substitute part of one or more word(s)	through letter or through characters	new character or new characters
Change to italics	under matter to be changed	
Change to capitals	under matter to be changed	
Change to small capitals	under matter to be changed	
Change to bold type	under matter to be changed	
Change to bold italic	under matter to be changed	
Change to lower case	Encircle matter to be changed	
Change italic to upright type	(As above)	
Change bold to non-bold type	(As above)	
Insert 'superior' character	through character or where required	or under character e.g. or
Insert 'inferior' character	(As above)	over character e.g.
Insert full stop	(As above)	
Insert comma	(As above)	
Insert single quotation marks	(As above)	or and/or or
Insert double quotation marks	(As above)	or and/or or
Insert hyphen	(As above)	
Start new paragraph		
No new paragraph		
Transpose		
Close up	linking characters	
Insert or substitute space between characters or words	through character or where required	
Reduce space between characters or words	between characters or words affected	