Comput Optim Appl (2016) 65:399–429
DOI 10.1007/s10589-015-9765-1

A novel class of approximate inverse preconditioners for large positive definite linear systems in optimization

Giovanni Fasano[1,2] · **Massimo Roma**[3]

Received: 27 June 2014 / Published online: 2 July 2015

Abstract We propose a class of preconditioners for large positive definite linear systems, arising in nonlinear optimization frameworks. These preconditioners can be computed as *by-product* of Krylov-subspace solvers. Preconditioners in our class are chosen by setting the values of some user-dependent parameters. We first provide some basic spectral properties which motivate a theoretical interest for the proposed class of preconditioners. Then, we report the results of a comparative numerical experience, among some preconditioners in our class, the unpreconditioned case and the preconditioner in Fasano and Roma (Comput Optim Appl 56:253–290, 2013). The experience was carried on first considering some relevant linear systems proposed in the literature. Then, we embedded our preconditioners within a linesearch-based Truncated Newton method, where sequences of linear systems (namely Newton's equations), are required to be solved. We performed an extensive numerical testing over the entire medium-large scale convex unconstrained optimization test set of CUTEst collection (Gould et al. Comput Optim Appl 60:545–557, 2015), confirming the efficiency of our proposal and the improvement with respect to the preconditioner in Fasano and Roma (Comput Optim Appl 56:253–290, 2013).

✉ Giovanni Fasano
fasano@unive.it

Massimo Roma
roma@dis.uniroma1.it

1 Dipartimento di Management, Università Ca'Foscari Venezia, S. Giobbe, Cannaregio 873, 30121 Venice, Italy

2 National Research Council–Marine Technology Research Institute (CNR-INSEAN), via di Vallerano, 139, 00128 Rome, Italy

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

Keywords Preconditioners · Large positive definite linear systems · Large scale convex optimization · Krylov-subspace methods

1 Introduction

We study a class of preconditioners for the solution of the symmetric positive definite linear system

$$Ax = b, \qquad A \in \mathbb{R}^{n \times n},$$

where n is *large* and we do not assume any sparsity pattern for the system matrix A. The solution of large linear systems is sought in a variety of real applications and in different contexts. Moreover, the use of preconditioning is often an essential issue to improve the efficiency of iterative solvers. Numerical Analysis and Optimization give plenty of frameworks where the solution of large linear systems (or a sequence of linear systems) is sought. Truncated Newton methods in unconstrained optimization, KKT systems, interior point methods, and PDE-constrained optimization are just some examples. Similarly, several real applications, ranging from power systems networks to economic models and queuing systems, involve the solution of large linear systems.

Typically, up to one decade ago, the specialized literature was keen on privileging the use of direct methods when n was moderately small, in view to their reasonable cost, whereas direct methods might be unaffordable for large n. However, more recently an increasing blurred use of techniques is observed, in both sparse direct methods and iterative algorithms, in order to efficiently solve linear systems (see e.g. [3,5]). Observe that for linear systems where the matrix A is block-diagonal or banded, which typically arise when solving discretized PDEs, specific solvers from the literature can be used [24], which require to include effective preconditioning strategies, too.

In this paper we focus on the use of iterative methods for solving positive definite linear systems: the iterative techniques are also used to provide sufficient information on the system matrix, in order to generate the preconditioners. We propose a parameter dependent class of preconditioners, which uses information collected by any Krylov-subspace method (or possibly using L-BFGS updates), in order to capture the structural properties of the positive definite system matrix.

Our proposal gives evidence to shift some eigenvalues of the preconditioned system matrix to a specific value. The basic idea of our approach draws its inspiration from *Approximate Inverse Preconditioners*, which have proved, to large extent, to be efficient in practice [3,4]. These methods claim that in principle, an approximate inverse of A should be computed and used as a preconditioner. However, observe that in practice it might be difficult to ensure that the approximate inverse summarizes enough information about A, and is sparse.

In this paper we apply any Krylov-subspace method to build our preconditioners, needing to store just a small tridiagonal matrix of order $k \ll n$, without requiring any product of matrices. As we collect information from Krylov-subspace methods, we assume that the entries of the system matrix are not stored and the necessary information is gained by simply using a routine, which computes the product of the system matrix times a vector. Note that, typically, the product of a matrix times a vector

allows fast parallel computing, which is another possible advantage of our approach, in large scale settings.

The preconditioners proposed in this paper depend on a couple of parameters, say δ and a, whose effect is substantially that of exalting the information on the system matrix collected by the Krylov-subspace method. Note that, for $\delta = 1$ and $a = 0$ our proposal reduces to the preconditioner [11], by the same authors.

We experience our class of preconditioners on test problems from both Numerical Analysis and Convex Optimization. In particular, we first test them on significant linear systems, from both the literature and real applications. Then, we focus on the *Newton-Krylov methods*, also known as (Hessian-free) Truncated Newton methods (see e.g. [18,20] for a survey on the importance of preconditioning in Truncated Newton methods). For suitable values of $\delta \neq 1$ we show that our novel class of preconditioners can outperform the proposal in [11].

We highlight that here, instead of following the idea early developed in [23], where a full-memory quasi-Newton formula is adopted for the preconditioner, we show that a few iterations of any Krylov-subspace method can be used, in order to provide information for building our preconditioners.

The paper is organized as follows: Sect. 2 reports some preliminaries and Sect. 3 contains the definition and the main motivations of our class of preconditioners. Section 4 studies some structural properties of our proposal, while in Sect. 5 some additional properties are included. In Sect. 6 we report the results of a relevant numerical experience and Sect. 7 adds some conclusions. Finally, in the Appendix we include the long and technical proofs of some theoretical results.

As regards the notations, for the $n \times n$ real matrix A we denote by $\Lambda[A]$ the spectrum of A. I_h is the identity matrix of order h. With $C \succ 0$ we indicate that the matrix C is positive definite, while $tr[C]$, $rk[C]$ and $\det[C]$ are the *trace*, the *rank* and the *determinant* of C, respectively. Finally, $\|\cdot\|$ denotes the Euclidean norm, e_h is the h-th unit vector and $\bigoplus$ is used to denote the direct sum of subspaces or matrices.

2 Preliminaries

Let us consider the *positive definite* linear system

$$Ax = b, \tag{2.1}$$

where $A \in \mathbb{R}^{n\times n}$ is *symmetric*, n is *large* and $b \in \mathbb{R}^n$. Some real contexts where the latter system requires efficient solvers are detailed in Sect. 1. Suppose any Krylov-subspace method is used for the solution of (2.1). Though the Conjugate Gradient (CG) method [9,10,13,17] is the most popular choice, we can use any Krylov-based method which provides a reduction of (2.1) to a tridiagonal system.

Now, suppose that $h \ll n$ steps of the Krylov-subspace method adopted have been performed when solving (2.1). At a generic step h of the Krylov-subspace method, with $h \leq n-1$, the matrices $R_h \in \mathbb{R}^{n\times h}$, $T_h \in \mathbb{R}^{h\times h}$ and the vector $u_{h+1} \in \mathbb{R}^n$ are generated [13], such that

$$AR_h = R_h T_h + \rho_{h+1} u_{h+1} e_h^T, \qquad \rho_{h+1} \in \mathbb{R}, \tag{2.2}$$

where

- $R_h = (u_1 \cdots u_h)$, $u_i^T u_j = 0$, $\|u_i\| = 1$, $1 \le i \ne j \le h+1$,
- T_h is tridiagonal, irreducible, nonsingular, with eigenvalues not all coincident.

Observe that multiplying (2.2) on the left by R_h^T we have $R_h^T A R_h = T_h$; then, since $A \succ 0$ we obtain $T_h \succ 0$, too. Also observe that no specific factorization of T_h is required to build our preconditioners, though in particular the CG provides the decomposition $T_h = L_h D_h L_h^T$, where D_h is diagonal and L_h is unit lower bidiagonal. In [11] the latter decomposition was used in order to simplify the construction of the preconditioner, which is generalized in this paper.

It is worth to highlight that also L-BFGS quasi-Newton scheme may provide information in order to satisfy (2.2). Indeed, according with [16], and using the correspondence between BFGS and CG when A is positive definite [21], in solving (2.1) a set of h conjugate directions $p_1, \ldots, p_h$ (and the vectors $Ap_1, \ldots, Ap_h$) can easily be computed after h iterations of L-BFGS. Now, following the guidelines in [25], it is not difficult to see that after a brief computation, the vectors

$$r_1 = p_1, \qquad r_{i+1} = r_i - \frac{p_i^T r_i}{p_i^T A p_i} A p_i, \qquad i = 1, \ldots, h-1$$

yield a set of orthogonal vectors, which can be used to provide T_h and relation (2.2). Thus, in practice many iterative methods commonly used for solving (2.1) may give, as by product, the information necessary to obtain the reduction (2.2).

Observe also that from (2.2) the parameter ρ_{h+1} may be possibly nonzero, i.e. the subspace $span\{u_1, \ldots, u_h\}$ is possibly not an invariant subspace under the transformation by matrix A. Thus, in this paper we consider a more general case with respect to [1].

3 Motivations for our class of preconditioners

On the basis of relation (2.2), we can now define our class of preconditioners and show its properties. The contents of the following two sections draw their inspiration from the theory in [11], where a preconditioner for indefinite linear systems was proposed. In particular, the latter preconditioner proved to be effective on several nonlinear large scale minimization test problems, within a Truncated Newton method. However, on a few test problems, both convex and nonconvex, we still experienced some severe inefficiencies of that proposal, when compared with an unpreconditioned scheme. Moreover, we observed that different pathologies arose, depending on the nature of the linear systems solved, say indefinite or positive definite. Thus, there was the necessity to further analyze the effects of a reliable preconditioner, separately for the indefinite and the positive definite case.

In order to overcome the latter drawbacks and gaps we have generalized here the proposal of [11] for positive definite systems, by adding a couple of parameters to the definition of the preconditioner, obtaining a novel class of Approximate Inverse preconditioners.

We introduce the following class of preconditioners

$$M_h^\sharp(a,\delta) \stackrel{\text{def}}{=} \left[I_n - (R_h \mid u_{h+1})\,(R_h \mid u_{h+1})^T \right] + (R_h \mid u_{h+1}) \left(\begin{array}{c|c} \delta^2 T_h & a e_h \\ \hline a e_h^T & 1 \end{array} \right)^{-1} (R_h \mid u_{h+1})^T, \quad h \le n-1, \tag{3.1}$$

$$M_n^\sharp(a,\delta) \stackrel{\text{def}}{=} \frac{1}{\delta^2} R_n T_n^{-1} R_n^T, \tag{3.2}$$

where $\delta, a \in \mathbb{R}$ are user dependent parameters. Observe that the matrix $I_n - (R_h \mid u_{h+1})\,(R_h \mid u_{h+1})^T$ in (3.1) simply represents a projector onto the subspace orthogonal to $(R_h \mid u_{h+1})$. In particular, when $h = n$ then in (2.2) $\rho_{h+1} = 0$, and the matrix R_h is orthogonal, having $T_n^{-1} = R_n^T A^{-1} R_n$. In addition, note that for $\delta = 1$ and $a = 0$ the preconditioners reduce to the proposal in [11]. The role played by the two parameters δ and a in our class of preconditioners has been investigated where A in (2.1) is positive definite, since in the indefinite case the spectral properties of the resulting preconditioned matrix require a more sophisticated analysis. That is why in this paper we preferred to detail both theoretical results and a numerical experience just focusing on the solution of positive definite linear systems. In particular, the introduction of δ and a in our preconditioners seems apparently a slight generalization of the proposal in [11]. Instead, from both a theoretical and a numerical standpoint, in the next sections we are going to show novel important conclusions. In summary, we have the following novel distinguishing features:

- the analysis developed in [11], which is referred to general indefinite linear systems, may be hardly extended to the case of our class of preconditioners. Roughly speaking, this is mainly due to the technical difficulties of explicitly calculating the symbolic inverse of the matrix

$$\left(\begin{array}{c|c} \delta^2 T_h & a e_h \\ \hline a e_h^T & 1 \end{array} \right)^{1/2}. \tag{3.3}$$

  Thus, the main conclusion in items (d1) and (d2) of Theorem 2.1 in [11] can be hardly proved, and different technicalities seem to be necessary, in order to obtain even weaker results. In particular, apart from the special case where $a = 0$ (see Theorem 4.2) we do not provide here results in terms of the eigenvalues of the preconditioned matrix (see e.g. items (c) and (d) of Theorem 4.3). Conversely, with respect to [11] in the current paper we also weaken the request on the Krylov-subspace method used in order to provide (2.2). Indeed, relation (2.3) in the Assumption 2.1 of [11] is no more necessary here;
- numerical performance of our class of preconditioners seems to be strongly affected by the choice of the parameter δ, at least in the positive definite case, on which we focused in this paper. To highlight this conclusion, unlike in [11], here we have first analyzed (in case $a = 0$) how tuning the parameter δ possibly modifies the spectral properties of the preconditioned matrix. Then, we explicitly investigated how relatively large values of δ tend to speed up the convergence of a Truncated Newton method where we embedded our preconditioners.

Finally, observe that given the matrix T_h in (2.1), in [11] it was necessary to introduce also the matrix $|T_h|$, i.e. a suitable modification of T_h in order to guarantee that the proposed preconditioner was positive definite. Since in the present paper we analyze only the case where A in (2.1) is positive definite, using the taxonomy adopted in [11] we have $|T_h| \equiv T_h$.

4 Structural properties of our preconditioners

This section summarizes some basic structural properties of our class of preconditioners. In particular, we report a couple of results concerning the structural properties of the preconditioners (3.1)–(3.2), which are strongly related to the pair of parameters a and δ. The first one is straightforwardly obtained by Theorem 2.1 in [11]. It concerns the eigenvalues of the preconditioned matrix in case $a = 0$. The second one considers the general case $a \in \mathbb{R}$, but provides results only in terms of singular values of the preconditioned matrix. Even if the analysis on singular values does not yield direct information of the convergence properties of a Krylov-subspace method, nonetheless it spots some light on the behaviour of our preconditioners thanks to some known results in literature. We refer to results connecting singular values and eigenvalues (i.e. Weyl's theorems) and, in particular, to those we report in the following proposition (see e.g., FACT 5.11.29 and FACT 5.11.28 in [6]) referred to the preconditioned matrix $M_h^\sharp(a, \delta)A$.

Proposition 4.1 *Let us denote by* $0 < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_{n-1} \le \lambda_n$ *the ordered eigenvalues and by* $\sigma_1 \le \sigma_2 \le \cdots \le \sigma_{n-1} \le \sigma_n$ *the ordered singular values of the preconditioned matrix* $M_h^\sharp(a, \delta)A$. *Then it results*

(i) $\sigma_1 \le \lambda_1 \le \cdots \le \lambda_n \le \sigma_n,$

(ii) $\displaystyle\prod_{i=k}^{n} \lambda_i \le \prod_{i=k}^{n} \sigma_i, \quad k = 1, \ldots, n,$

(iii) $\displaystyle\prod_{i=1}^{k} \sigma_i \le \prod_{i=1}^{k} \lambda_i, \quad k = 1, \ldots, n.$

On the basis of these results, we can argue that whenever some large singular values of $M_h^\sharp(a, \delta)A$ are decreased, then at least some large eigenvalues of $M_h^\sharp(a, \delta)A$ tend to decrease similarly (see (ii)). Conversely, whenever some small singular values of $M_h^\sharp(a, \delta)A$ are increased, then at least some small eigenvalues of $M_h^\sharp(a, \delta)A$ tend to increase (see (iii)). Therefore, information on singular values of the preconditioned matrix can be exploited to possibly deduce convergence properties of a preconditioned Krylov-subspace method, too. In Sect. 6 and [12] we refer to a more detailed analysis, motivating the latter statement.

In the next theorem we report the first result concerning the eigenvalues of the preconditioned matrix in case $a = 0$.

Theorem 4.2 *Consider any Krylov-subspace method to solve the symmetric positive definite linear system* (2.1). *Suppose that the Krylov-subspace method performs* $h \le n$

iterations, so that (2.2) *holds. Then, setting* $a = 0$ *and* $\delta \in \mathbb{R}$ *in* (3.1)–(3.2)*, the resulting preconditioner*

$$M_h^\sharp(0,\delta) = \left[I_n - (R_h \mid u_{h+1})\,(R_h \mid u_{h+1})^T\right] + (R_h \mid u_{h+1}) \left(\begin{array}{c|c} \delta^2 T_h & 0 \\ \hline 0 & 1 \end{array}\right)^{-1} (R_h \mid u_{h+1})^T \tag{4.1}$$

$$M_n^\sharp(0,\delta) = \frac{1}{\delta^2} R_n T_n^{-1} R_n^T, \tag{4.2}$$

is such that

(a) *the matrix* $M_h^\sharp(0,\delta)$ *is symmetric and nonsingular;*
(b) *the matrix* $M_h^\sharp(0,\delta)$ *is positive definite. Moreover, its spectrum* $\Lambda[M_h^\sharp(0,\delta)]$ *is given by*

$$\Lambda[M_h^\sharp(0,\delta)] = \frac{1}{\delta^2}\Lambda\left[T_h^{-1}\right] \cup \Lambda\left[I_{n-h}\right];$$

(c) *when* $h \le n-1$ *the matrix* $M_h^\sharp(0,\delta)A$ *has at least* $(h-1)$ *eigenvalues equal to* $+1/\delta^2$;
(d) when $h = n$ the eigenvalues of $M_h^\sharp(0,\delta)A$ are equal to $+1/\delta^2$.

Proof From (4.1), after a brief computation, we obtain

$$M_h^\sharp(0,\delta) = (I_n - R_h R_h^T) + \frac{1}{\delta^2} R_h T_h^{-1} R_h^T,$$

which coincides with the proposal in [11], in the positive definite case, as long as $\delta = 1$. Thus, the result is directly obtained following the guidelines of the proof of Theorem 2.1 in [11], considering that possibly $\delta \neq 1$. □

In the next theorem we analyze the more general case where the parameter a is possibly nonzero. The results are reported in terms of singular values of the preconditioned matrix. For the sake of readability of the paper, we moved the long proof to the Appendix.

Theorem 4.3 *Consider any Krylov-subspace method to solve the symmetric linear system* (2.1)*, where* A *is positive definite. Suppose that the Krylov-subspace method performs* $h \le n$ *iterations and provides relation* (2.2)*. Let* $a \in \mathbb{R}$ *and* $\delta \neq 0$*. Then, for the class of preconditioners* (3.1)–(3.2) *we have the following properties:*

(a) *the matrix* $M_h^\sharp(a,\delta)$ *is symmetric. Furthermore,*
   – *when* $h \le n-1$*, for any* $a \in \mathbb{R}\backslash\left\{\pm\delta(e_h^T T_h^{-1} e_h)^{-1/2}\right\}$, $M_h^\sharp(a,\delta)$ *is nonsingular. In addition*

$$\det\left(M_h^\sharp(a,\delta)\right) = \delta^{-2h}\det(T_h^{-1})\left(1 - \frac{a^2}{\delta^2} e_h^T T_h^{-1} e_h\right)^{-1};$$

– *when* $h = n$ *the matrix* $M_h^\sharp(a,\delta)$ *is nonsingular. In addition*

$$\det\left(M_n^\sharp(a,\delta)\right) = \det(T_h^{-1});$$

(b) *for* $|a| < |\delta|(e_h^T T_h^{-1} e_h)^{-1/2}$ *the matrix* $M_h^\sharp(a,\delta)$ *is positive definite. Moreover, the spectrum* $\Lambda[M_h^\sharp(a,\delta)]$ *is given by*

$$\Lambda[M_h^\sharp(a,\delta)] = \Lambda\left[\left(\begin{array}{c|c}\delta^2 T_h & ae_h\\ \hline ae_h^T & 1\end{array}\right)^{-1}\right] \cup \Lambda\left[I_{n-(h+1)}\right];$$

(c) *when* $h \le n-1$ *then*
- $M_h^\sharp(a,\delta)A$ *has at least* $(h-3)$ *singular values equal to* $+1/\delta^2$*;*
- *if* $a = 0$ *then the matrix* $M_h^\sharp(0,\delta)A$ *has at least* $(h-2)$ *singular values equal to* $+1/\delta^2$*;*

(d) *when* $h = n$*, then each of the* n *eigenvalues of the preconditioned matrix* $M_h^\sharp(a,\delta)A$ *is equal to* $+1/\delta^2$.

Proof The proof is reported in the Appendix. □

5 Some additional features

We report in this section some properties concerning both invariance and scalability of our class of preconditioners (3.1)–(3.2). Moreover, we provide possible generalizations in order to build our preconditioners.

Proposition 5.1 *Suppose* (2.2) *holds, with* $A \succ 0$. *Let* $P \in \mathbb{R}^{h\times h}$*, with* P *orthogonal. Then, the preconditioners* $M_h^\sharp(0,\delta)$ *are invariant under the transformation* $R_h = Q_h P$, $Q_h = (q_1 \cdots q_h)$, $q_i^T q_j = 0$ *and* $\|q_i\| = 1$, *for* $1 \le i \ne j \le h$, *and* $Q_h^T u_{h+1} = 0$. *Moreover, considering the scaled system* $(\varepsilon A)x = (\varepsilon b)$, $\varepsilon > 0$, *in place of* (2.1), *then the preconditioners* (3.1)–(3.2) *become*

$$\begin{aligned} M_h^\sharp(a,\delta) &= \left[I_n - (R_h \mid u_{h+1})\,(R_h \mid u_{h+1})^T\right] \\ &\quad + (R_h \mid u_{h+1})\left(\begin{array}{c|c}\delta^2\varepsilon T_h & ae_h\\ \hline ae_h^T & 1\end{array}\right)^{-1}(R_h \mid u_{h+1})^T, \qquad h \le n-1, \quad (5.1)\\ M_n^\sharp(a,\delta) &= \frac{1}{\varepsilon} R_n T_n^{-1} R_n^T, \qquad h = n. \end{aligned}$$

Proof From (2.2) and condition $R_h = Q_h P$ we have

$$T_h = P^T Q_h^T A Q_h P = P^T \tilde{T}_h P,$$

where $\tilde{T}_h \in \mathbb{R}^{h\times h}$ is possibly not tridiagonal. Moreover, we have for $h \le n-1$

$$
\begin{aligned}
M_h^\sharp(0,\delta) &= \Big[I_n - (R_h \mid u_{h+1})\,(R_h \mid u_{h+1})^T \Big] \\
&\quad + (R_h \mid u_{h+1}) \left(\begin{array}{c|c} \delta^2 P^T Q_h^T A Q_h P & 0 \cdot e_h \\ \hline 0 \cdot e_h^T & 1 \end{array} \right)^{-1} (R_h \mid u_{h+1})^T \\
&= \Big[I_n - (Q_h \mid u_{h+1})\,(Q_h \mid u_{h+1})^T \Big] \\
&\quad + \frac{1}{\delta^2} Q_h P \left(P^T Q_h^T A Q_h P \right)^{-1} P^T Q_h^T \; + \; u_{h+1} u_{h+1}^T \\
&= \Big[I_n - Q_h Q_h^T \Big] + \frac{1}{\delta^2} Q_h P P^T \left(Q_h^T A Q_h \right)^{-1} P P^T Q_h^T \\
&= \Big[I_n - (Q_h \mid u_{h+1})\,(Q_h \mid u_{h+1})^T \Big] + (Q_h \mid u_{h+1}) \left(\begin{array}{c|c} \delta^2 \tilde{T}_h & 0 \\ \hline 0 & 1 \end{array} \right)^{-1} (Q_h | u_{h+1})^T,
\end{aligned}
$$

which coincides with (3.1), setting $a = 0$, replacing R_h with Q_h and considering that T_h^{-1} and $\tilde{T}_h^{-1}$ are likely both dense, regardless of the sparsity of T_h and $\tilde{T}_h$. The previous result holds also for $h = n$, after a trivial computation.

Furthermore, observe that the matrix R_h in (3.1)–(3.2) is invariant under the scale factor ε in $(\varepsilon A)x = (\varepsilon b)$, and replacing A with εA, by (2.2) the tridiagonal matrix T_h becomes $\varepsilon R_h^T A R_h = \varepsilon T_h$. Thus, (5.1) trivially holds. □

Broadly speaking, as for other preconditioners for large positive definite linear systems in the literature (see e.g. the Limited Memory Preconditioners in [16]), the preconditioners (3.1)–(3.2) cannot be independent of the scale parameter ε. Indeed, as we can soon realize, when $h = n$ and $A \succ 0$, the matrix $M_n^\sharp(a,\delta)$ is the inverse of the system matrix εA, so that

$$
M_n^\sharp(a,\delta) \cdot (\varepsilon A) = \left[\frac{1}{\varepsilon} R_n T_n^{-1} R_n^T \right] \left[\varepsilon R_n T_n R_n^T \right] = I_n .
$$

As regards the construction of our class of preconditioners, suppose the Krylov-subspace method has performed m iterations. Then, several strategies can be adopted by selecting ℓ vectors among $\{u_1, \ldots, u_m\}$, with $\ell \le m$ (see also [19]), where $u_1, \ldots, u_m$ are the vectors generated by the Krylov-subspace method. However, the reader is warned that depending on the resulting strategy adopted, the properties in Theorem 4.3 should be suitably restated. On this guideline, now we want to analyze the strategies corresponding to choose either the *first* ℓ vectors $\{u_1, \ldots, u_\ell\}$, or the *last* $m - \ell$ vectors $\{u_{\ell+1}, \ldots, u_m\}$. To this purpose, considering (2.1) suppose a Krylov-subspace method was adopted to generate the recurrence

$$
A R_m = R_m T_m + \rho_{m+1} u_{m+1} e_m^T, \qquad m \le n. \tag{5.2}
$$

Since $R_m = (R_\ell \mid R_{m,\ell+1})$, where $R_{m,\ell+1} = (u_{\ell+1} \mid \cdots \mid u_m)$, setting for the tridiagonal matrix T_m the decomposition (where also $T_{m,\ell+1}$ is tridiagonal)

$$T_m = \left(\begin{array}{c|c} T_\ell & \begin{array}{c} 0 \\ \hline \sigma e_1^T \end{array} \\ \hline \begin{array}{c} \sigma e_\ell^T \\ \hline 0 \end{array} & T_{m,\ell+1} \end{array} \right), \qquad \text{for some } \sigma \in \mathbb{R},$$

from (5.2) we have

$$\begin{aligned} AR_m &= A(R_\ell \mid R_{m,\ell+1}) \;=\; (R_\ell \mid R_{m,\ell+1})T_m + \rho_{m+1}u_{m+1}e_m^T \\ &= \left(R_\ell T_\ell + \sigma u_{\ell+1}e_\ell^T \mid \sigma u_\ell e_1^T + R_{m,\ell+1}T_{m,\ell+1}\right) + \rho_{m+1}u_{m+1}e_m^T, \end{aligned}$$

which is equivalent to the following pair of conditions

$$AR_\ell = R_\ell T_\ell + \sigma u_{\ell+1}e_\ell^T \tag{5.3}$$

$$AR_{m,\ell+1} = R_{m,\ell+1}T_{m,\ell+1} + \sigma u_\ell e_1^T + \rho_{m+1}u_{m+1}e_m^T. \tag{5.4}$$

Thus, if only the first ℓ vectors $u_1, \ldots, u_\ell$ are used to build our preconditioners, then relation (5.3) must be adopted in place of (2.2). On the other hand, if the last $m - \ell$ vectors $u_{\ell+1}, \ldots, u_m$ are used, relation (2.2) must be replaced by (5.4). However, in the latter case the statement of Theorem 4.3 should be slightly modified, accordingly. Of course, other possible strategies to select vectors among $\{u_1, \ldots, u_m\}$ can be considered, which may require a more consistent reformulation of the statement of Theorem 4.3.

Remark 5.1 The effective choice of the parameters δ and a in (3.1) might be in our experience strongly problem dependent; nevertheless, general guidelines for their choice are given in Sect. 6. Moreover, we provide in the next section a numerical experience where several values of these parameters are selected. We recall that from a theoretical standpoint, values of δ and a may be set considering items (b) and (c) of Theorem 4.3. The latter may be used in order to impose conditions like the following, which tend to force the clustering of the eigenvalues of matrix $H_{(h+1)\times(h+1)}$ or $H_{h\times h}$ defined in (7.13)–(7.14), near $+1$ (see also the comments in Sect. 6):

$$tr\left[H_{h\times h}\right] = h,$$

$$tr\left[H_{(h+1)\times(h+1)}\right] = h + 1.$$

Observe that clustering the eigenvalues of $H_{(h+1)\times(h+1)}$ or $H_{h\times h}$ induces a clustering of some *singular values* of $M_h^\sharp(a,\delta)A$, and by Proposition 4.1 the latter fact possibly forces also a clustering of some *eigenvalues* of $M_h^\sharp(a,\delta)A$. Finally, observe that there may be real values of the parameters δ and a such that the expressions (7.14)–(7.15) are further simplified. In the next section we detail more specific motivations for the choice of δ and a, in our numerical experience.

6 Numerical experiments

In order to preliminarily test our proposal in different frameworks, where no information is known about the sparsity pattern of the matrix A, we experimented with our class of preconditioners $M_h^\sharp(a,\delta)$, setting for simplicity $a = 0$. We preferred in our numerical experience to keep one of the two parameters unchanged (say $a = 0$), and vary only the other parameter (i.e. δ), for the following three reasons:

- according with Lemma 4.4 in [12], when $a = 0$ we can suitably bound the condition number of $M_h^\sharp(a,\delta)A$ (so that a bound on the spectral condition number of $M_h^\sharp(a,\delta)A$ is also available);
- the interaction between a and δ, in order to obtain an effective final preconditioner, is itself dependent on the problem in hand, even in case (2.1) is positive definite;
- based on our experience, the overall efficiency of our preconditioners appears to be more sensible to modifications of δ than to modifications of a. This may be easily deduced by inspection of formula (3.1) and recalling the Cauchy Interlacing properties for the eigenvalues of symmetric matrices. Indeed, observe that the choice of δ performs a *scaling* of the eigenvalues of the entire matrix T_h, which means, by (7.4), that *at least* $h-1$ eigenvalues of $M_h^\sharp(a,\delta)$ are directly affected by δ. On the other hand, for a given δ, the parameter a directly affects *at most* two eigenvalues.

As concerns the Krylov-subspace method used for these numerical experiences, we choose the CG method. As expected, in our numerical experience we obtained results which match the theory in Theorem 4.3. In particular, in the following sections, in order to test the class of preconditioners (3.1)–(3.2), we used different sets of test problems. In Sects. 6.1–6.2 we first checked the results in Theorem 4.3 on positive definite linear systems suggested by the literature. Here, since the spectral properties of the matrices were known in advance, we could set δ so that the value $1/\delta^2$ (see item (c) of Theorem 4.3) is nearby the middle of the spectrum of the system matrix. In Sect. 6.3 we considered a large test set from convex optimization, which is the main focus of this paper, where we solved Newton's equation within a Truncated Newton method. Here, we had no information about the spectrum of the Hessian matrix, so that we performed a more accurate investigation using several values of δ.

We are going to prove that to a large extent our proposal is efficient and effective with respect to both the unpreconditioned case and the preconditioner in [11], showing its robustness on convex problems.

The Matlab routine `eigs()` is used to compute the eigenvalues for both A and $M_h^\sharp(a,\delta)A$. Furthermore, the values of the condition number κ reported, for both A and $M_h^\sharp(a,\delta)A$ are computed as

$$\kappa = \frac{\max_i |\lambda_i|}{\min_i |\lambda_i|}, \tag{6.1}$$

(i.e. the spectral condition number) being λ_i the respective eigenvalues of the matrices. As remarked also in Sect. 4 of [12], in the case of $M_h^\sharp(a,\delta)A$, the computation of (6.1) provides only a lower bound of the actual condition number.

6.1 Test set 1

In a very preliminary experiment we generated the positive definite matrix A in (2.1) such that

$$A = H\mathcal{D}H, \tag{6.2}$$

where $H \in \mathbb{R}^{n\times n}$, $n = 1000$, is an Householder transformation given by $H = I_n - 2vv^T$, with $v \in \mathbb{R}^n$ a unit vector, randomly chosen. The matrix $\mathcal{D} \in \mathbb{R}^{n\times n}$ is diagonal (so that its entries are also eigenvalues of A, while each column of H is also an eigenvector of A) and its entries are randomly chosen in the uniform distribution $U(0, 100]$. The matrix $\mathcal{D}$ is such that its $\texttt{perc}\cdot n$ eigenvalues are larger (about one order of magnitude) than the remaining $(1 - \texttt{perc}) \cdot n$ eigenvalues (we set without loss of generality $\texttt{perc} = 0.3$). Finally, we computed the preconditioners $M_h^\sharp(0, \delta)$ in (3.1), setting the starting point x_0 so that the initial residual $b - Ax_0$ was a linear combination (with coefficients -1 and $+1$ randomly chosen) of *all* the n eigenvectors of A. We strongly highlight that the latter choice of x_0 is expected to be not favorable when applying the CG, to build the preconditioners. In the latter case the CG method is indeed expected to perform exactly n iterations before stopping (see also [22,24]), so that the matrix (6.2) may be significant to test the effectiveness of our preconditioners, in case of *small values* of h (roughly speaking, here h *small* implies that the preconditioners contain correspondingly *little* information on the inverse matrix A^{-1}). We compared the spectra $\Lambda[A]$ and $\Lambda[M_h^\sharp(0, \delta)A]$, setting $\delta = 1/7$, in order to verify both

- how the preconditioners are possibly able to *cluster the eigenvalues of* A;
- how the preconditioners *alter the condition number* of the preconditioned matrix.

The choice of $\delta = 1/7$ was motivated by item (c) of Theorem 4.3. Indeed, since the eigenvalues of A in (6.2) are in the range $(0, 100]$, the choice $\delta = 1/7$ is expected to yield a shifting of some singular values of $M_h^\sharp(0, \delta)A$ nearby 49 (which is almost in the middle of the interval $(0, 100]$). Similarly, some eigenvalues of $M_h^\sharp(0, \delta)A$ are also expected to shift accordingly. Following the choice in [19], in order to test our proposal on a range of values for the parameter h, we set $h \in \{4, 8, 12, 16, 20, 40\}$.

The results are given in Fig. 1 (spectral condition numbers) and Figs. 2, 3, and 4 (eigenvalues distribution). In Fig. 2 we include all the 1000 eigenvalues (*left*) and a detail of the eigenvalues (*right*) from the 780th to the 850th, for the unpreconditioned

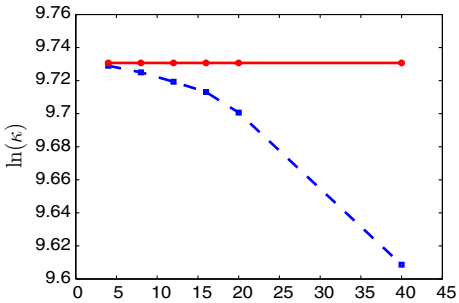

Fig. 1 The spectral condition number of matrix A (*continuous line* independent of h) along with the spectral condition number of matrix $M_h^\sharp(0, \delta)A$ (*dashed line*), when $h \in \{4, 8, 12, 16, 20, 40\}$ and $\delta = 1/7$. On the vertical axis the natural logarithm of κ

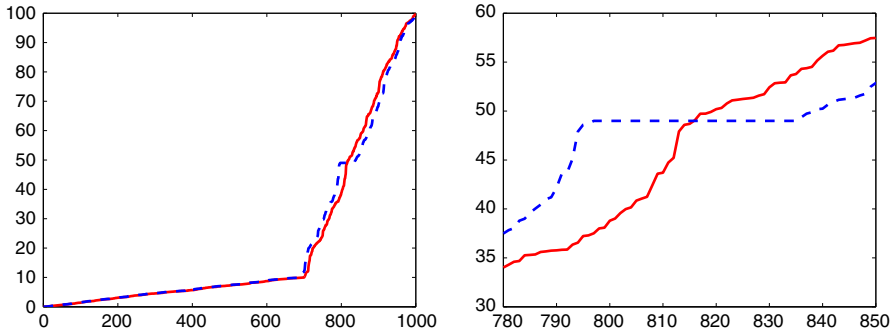

Fig. 2 Comparison between the full (*left*) and detailed (*right*) spectra $\Lambda[A]$ (*continuous line*) and $\Lambda[M_h^\sharp(0,\delta)A]$ (*dashed line*), with A given by (6.2) (eigenvalues are sorted), setting $h = 40$ and $\delta = 1/7$

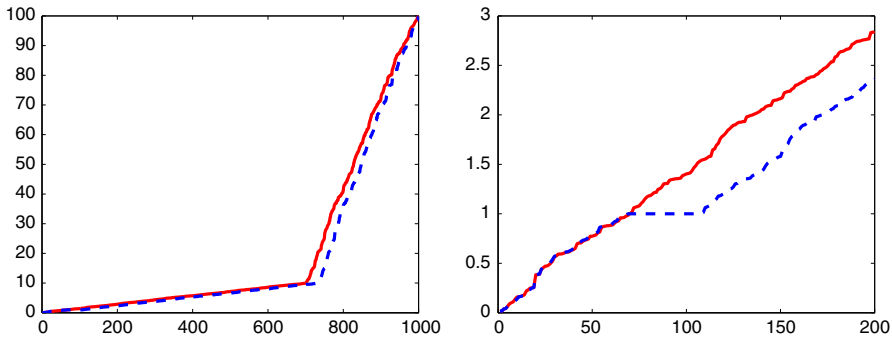

Fig. 3 Comparison between the full (*left*) and detailed (*right*) spectra $\Lambda[A]$ (*continuous line*) and $\Lambda[M_h^\sharp(0,\delta)A]$ (*dashed line*), with A given by (6.2) (eigenvalues are sorted), setting $h = 40$ and $\delta = 1$

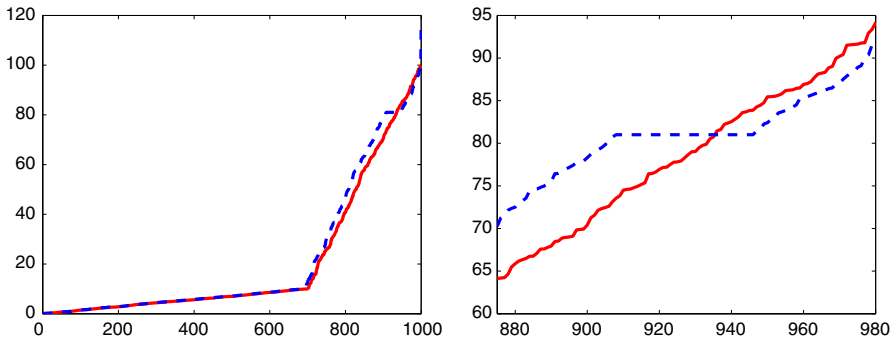

Fig. 4 Comparison between the full (*left*) and detailed (*right*) spectra $\Lambda[A]$ (*continuous line*) and $\Lambda[M_h^\sharp(0,\delta)A]$ (*dashed line*), with A given by (6.2) (eigenvalues are sorted), setting $h = 40$ and $\delta = 1/9$

matrix (continuous line) and the preconditioned matrix (dashed line). In the latter picture we used $h = 40$ in order to appreciate more evident results (though similar results are definitely obtained for any value of h). A 'flatter' piecewise-line of the

eigenvalues in $\Lambda[M_h^\sharp(0,\delta)A]$ indicates that the eigenvalues tend to cluster around $1/\delta^2$. For a more complete analysis, in Figs. 3 and 4 we also plotted the eigenvalues in $\Lambda[M_h^\sharp(0,\delta)A]$ for $\delta \in \{1, 1/9\}$, obtaining again a clustering nearby $1/\delta^2$.

Observe that the preconditioners $M_h^\sharp(0,\delta)$ are definitely able to shift some eigenvalues of A. In addition, since the eigenvalues of a matrix are a continuous function of its entries, also the remaining eigenvalues are evidently affected by the value of δ. As expected, the clustering of the eigenvalues is enhanced when the parameter h increases; moreover, the spectral condition number of the preconditioned matrix slightly improves.

6.2 Test set 2

We used another test set, obtained by considering a couple of positive definite small linear systems as (2.1), recommended in [19] and references therein, which come up from finite element problems. We addressed the latter linear systems as $A_0x = b_0$ ("from one-dimensional model, consisting of a line of two-node elements with support conditions at both ends, and a linearly varying body force") and $A_1x = b_1$ (where A_1 is the "stiffness matrix from a two-dimensional finite element model of a cantilever beam") respectively. The spectral properties of both the matrices A_0 and A_1 are extensively described in [19]. In particular $A_0 \in \mathbb{R}^{50\times 50}$ is positive definite with condition number $\kappa(A_0) = 0.2 \cdot 10^{10}$ and with a suitable pattern of the eigenvalues in the range $[10^0, 10^9]$; similarly, $A_1 \in \mathbb{R}^{170\times 170}$ is also positive definite, with condition number $\kappa(A_1) = 0.13 \times 10^9$ and a different pattern of eigenvalues in the range $[10^0, 10^{10}]$. In addition, we have $b_0^T = (0\ \ 200/49\ \ 300/49\ \cdots\ 4900/49\ \ 0)$, and

$$b_1 = 0, \quad \text{but} \quad b_1(34) = b_1(68) = b_1(102) = b_1(136) = b_1(170) = -8000.$$

The CG is again used to compute the vectors necessary to build the preconditioners $M_h^\sharp(0,\delta)$, adopting both the starting points $x_0 = 0$ and $x_0 = 100e$, where $e = (1\cdots 1)^T$, as indicated in [19]. The results of the numerical experience for the linear system $A_0x = b_0$ are summarized in Fig. 5. For simplicity the eigenvalues of the matrices are sorted, we set $h = 40$ (but other values of h yielded similar results) and the value $\delta = 10^{-4}$ is used to compute $M_h^\sharp(0,\delta)$. Again, δ was set so that the value $1/\delta^2$ was nearby the middle of the spectrum of the system matrix. Several eigenvalues of $M_h^\sharp(0,\delta)A_0$ tend to cluster around $+1/\delta^2$. Furthermore, also the remaining eigenvalues tend to be affected by the value of δ.

As regards the performance of $M_h^\sharp(0,\delta)$ on the linear system $A_1x = b_1$, we first recall that now $n = 170$. The numerical experience again considered the case $\delta = 10^{-4}$ and $h = 40$. Similar results are obtained with respect to the linear system $A_0x = b_0$ (with a just slight increase of the spectral condition number of the preconditioned matrix) and are summarized in Fig. 6.

We remark that similarly to the results plotted in Figs. 2, 3 and 4, modifying δ we can observe a clustering of some eigenvalues in $\Lambda[M_h^\sharp(0,\delta)A]$ nearby the value $1/\delta^2$ (for the sake of brevity the corresponding plots are omitted).

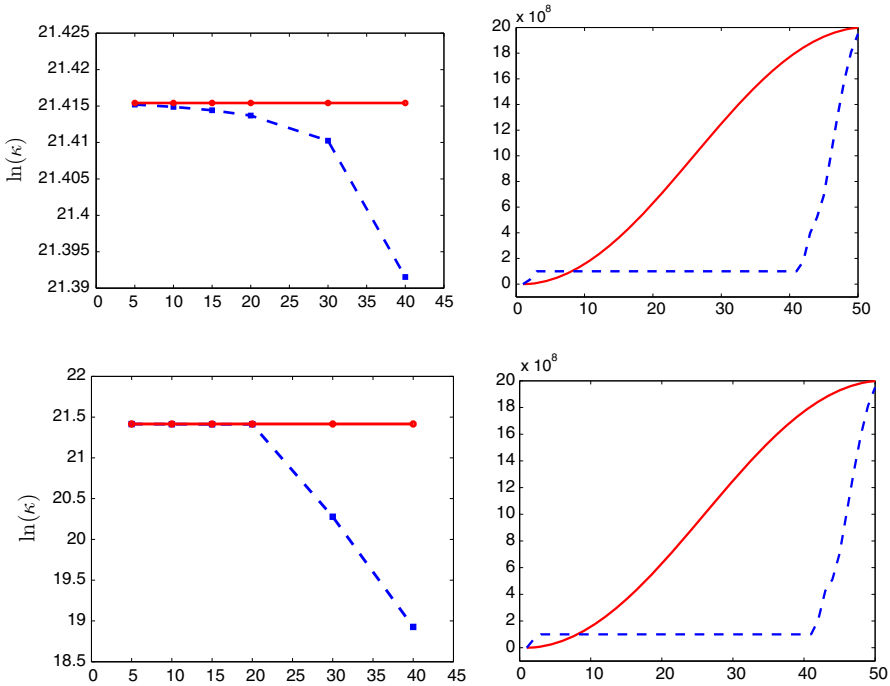

Fig. 5 Comparison between the spectral condition numbers (left) and the spectra (right) of A_0 (*continuous line*) and $M_h^\sharp(0,\delta)A_0$ (*dashed line*), setting $x_0 = 0$ (top) or $x_0 = 100e$ (bottom), and $\delta = 10^{-4}$. The pictures on the right refer to $h = 40$

6.3 Experiments on convex optimization

After the preliminary numerical tests in Sects. 6.1–6.2 we can now apply our proposal on the sequence of linear systems arising in a well known optimization framework, namely Truncated Newton methods. The fruitful use of preconditioning techniques within Truncated Newton methods is clearly pointed out in several papers (see e.g. [18,20] for a survey). We usually have no information about the Hessian matrix (e.g. no knowledge on the eigenvalues distribution) so that we tested our class of preconditioners for several values of the parameter δ, being $\delta \in \{0.1, 1, 10, 100\}$. Then, we compared the results with an unpreconditioned version of the algorithm (observe that for $\delta = 1$ we obtain on convex problems the results in [11]). In particular, as test set we considered medium-large scale unconstrained optimization problems, which were solved using the standard linesearch-based Truncated Newton method in Table 1, where the solution of the symmetric linear system (Newton's equation) $\nabla^2 f(z_k)d = -\nabla f(z_k)$ is required, at each outer iteration k.

As test problems we considered all the medium-large scale unconstrained optimization problems from CUTEst [15] collection, with $n \in [1000, 10000]$. At the outset of the outer iteration k we computed the preconditioner $M_h^\sharp(0,\delta)$, using the information collected by the CG, after $h = 7$ (inner) iterations, when solving the equation

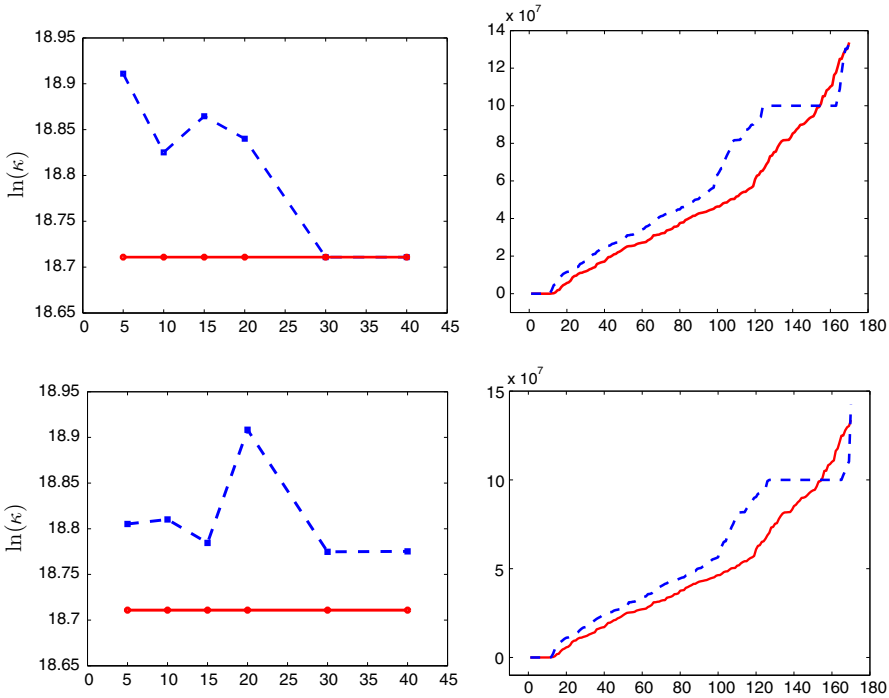

Fig. 6 Comparison between the spectral condition numbers (left) and the spectra (right) of A_1 (*continuous line*) and $M_h^\sharp(0,\delta)A_1$ (*dashed line*), setting $x_0 = 0$ (top) or $x_0 = 100e$ (bottom), and $\delta = 10^{-4}$. The pictures on the right refer to $h = 40$

Table 1 The standard linesearch-based Truncated Newton method we adopted

Set $z_0 \in \mathbb{R}^n$
OUTER ITERATIONS
for $k = 0, 1, \ldots$
Compute $\nabla f(z_k)$; if $\|\nabla f(z_k)\|$ is small then STOP

INNER ITERATIONS
Compute d_k which approximately solves $\nabla^2 f(z_k)d = -\nabla f(z_k)$ and satisfies *a truncation rule*

Compute the steplength a_k by an Armijo-type linesearch procedure
Update $z_{k+1} = z_k + a_k d_k$
endfor

$\nabla^2 f(z_k)d = -\nabla f(z_k)$ (for the choice $h = 7$ see [11,19]). Then, from the 8-th (inner) iteration we adopted $M_h^\sharp(0,\delta)$ as a preconditioner, for the solution of the linear system $\nabla^2 f(z_k)d = -\nabla f(z_k)$. All the parameters used within the preconditioning strategy, the truncated scheme and the linesearch adopted were exactly those chosen in [11].

All the test problems where the CG did not detect any negative curvature direction, for the objective function, were considered as *convex problems* and included in the comparison, so that the test set reduced to 78 *convex* problems.

We report the results in terms of number of iterations (*outer-it*), number of function evaluations (*f-eval*), number of inner CG-iterations (*CG-it*) and CPU time (*time*) in seconds. The optimal objective function value is also included (*opt-val*). We first report the complete results for the preconditioned Truncated Newton method obtained with $\delta = 1$ (see Table 2), i.e. the same results reported in [11]. Then, for sake of brevity, we do not report the complete results obtained using all the other values of δ tested (i.e. $\delta \in \{0.1, 10, 100\}$), but only those corresponding to the "most successful" value $\delta = 100$ (see Table 3). We also show in Fig. 7 (full picture) and in Fig. 8 (detail picture) the performance profiles (see [8]) where the comparison between preconditioned and unpreconditioned schemes is summarized, in terms of inner iterations (as also suggested in [19]). We highlight that profiles in terms of CPU time are possibly misleading, due to very similar times of computation on most test problems, along with the presence of other simultaneous processes.

As we can see, when $\delta \in \{0.1, 1, 10, 100\}$ the use of the preconditioner $M_h^\sharp(0, \delta)$ yields on average better results than the unpreconditioned algorithm. In particular, for $\delta \in \{10, 100\}$ our proposal definitely outperforms the results obtained by the same authors in [11] (i.e. $a = 0$ and $\delta = 1$). The latter behaviour was investigated, and some conclusions can be drawn considering also the analysis in [2,12,14].

Indeed, it is a matter of fact that regardless of the value of $\delta \neq 1$ in our class, by item (c) of Theorem 4.3 some singular values of $M_h^\sharp(a, \delta)\nabla^2 f(z_k)$ tend to be clustered. In addition, we have the following motivations to clarify why the Krylov-based method adopted to solve the (preconditioned) Newton's equation is expected to perform better:

- since (see Figs. 2, 5, 6) smaller eigenvalues are dragged upwards (towards $1/\delta^2$), while larger eigenvalues tend to be decreased (again towards $1/\delta^2$), then the overall condition number of the preconditioned matrix is beneficed;
- by (ii) of Proposition 4.1, when some large singular values of $M_h^\sharp(a, \delta)\nabla^2 f(z_k)$ are decreased, then at least some large eigenvalues of $M_h^\sharp(a, \delta)\nabla^2 f(z_k)$ tend to decrease similarly. On the other hand, reasoning as in Sect. 5 of [14] and as in [2], we can obtain that larger values of δ tend to decrease some large singular values of $M_h^\sharp(a, \delta)\nabla^2 f(z_k)$. In fact, when δ increases, the eigenvalues of $H_{h\times h}$ in (7.14) decrease, so that the trace of the matrix $M_h^\sharp(a, \delta)[\nabla^2 f(z_k)]^2 M_h^\sharp(a, \delta)$ tends to decrease. As a consequence, some large singular values of the preconditioned matrix $M_h^\sharp(a, \delta)\nabla^2 f(z_k)$ tend to decrease, too. Thus, we can loosely argue that increasing the value of δ (on average) tends to control at least some large singular values of the preconditioned matrix, thus affecting some large eigenvalues. This partially explains why the best performance for our preconditioners is obtained for $\delta = 100$;
- by (iii) of Proposition 4.1, when some small singular values of $M_h^\sharp(a, \delta)\nabla^2 f(z_k)$ are increased, then at least some small eigenvalues of $M_h^\sharp(a, \delta)\nabla^2 f(z_k)$ tend to increase;

Table 2 Results for the preconditioned Truncated Newton method with $\delta = 1$

PROBLEM	*n*	*outer-it*	*f-eval*	*CG-it*	*opt-val*	*time*
ARWHEAD	1000	34	364	37	0.000000D+00	0.02
ARWHEAD	10,000	10	102	11	1.332134D−11	0.05
BDQRTIC	1000	46	293	84	3.983818D+03	0.05
BDQRTIC	10,000	121	1217	204	4.003431D+04	0.89
BRYBND	1000	20	64	26	6.709348D−12	0.02
BRYBND	10,000	20	64	26	6.226697D−12	0.11
CRAGGLVY	1000	49	216	94	3.364231D+02	0.06
CRAGGLVY	10,000	116	776	173	3.377956D+03	0.89
CURLY10	1000	11,227	11,514	37,918	−1.003163D+05	14.35
CURLY10	10,000	59,999	61,159	203,494	−1.003163D+06	640.95
DIXMAANA	1500	8	13	9	1.000000D+00	0.01
DIXMAANA	3000	8	14	8	1.000000D+00	0.01
DIXMAANB	1500	5	10	6	1.000000D+00	0.01
DIXMAANB	3000	5	10	6	1.000000D+00	0.00
DIXMAANC	1500	5	11	6	1.000000D+00	0.01
DIXMAANC	3000	5	11	6	1.000000D+00	0.00
DIXMAAND	1500	5	8	5	1.000000D+00	0.01
DIXMAAND	3000	5	8	5	1.000000D+00	0.01
DIXMAANE	1500	76	79	168	1.000000D+00	0.09
DIXMAANE	3000	114	117	258	1.000000D+00	0.25
DIXMAANF	1500	52	57	136	1.000000D+00	0.09
DIXMAANF	3000	54	59	143	1.000000D+00	0.19
DIXMAANH	1500	54	56	134	1.000000D+00	0.11
DIXMAANH	3000	74	76	209	1.000000D+00	0.27
DIXMAANI	1500	215	218	693	1.000001D+00	0.39
DIXMAANI	3000	235	238	714	1.000003D+00	0.76
DIXMAANK	1500	60	74	173	1.000000D+00	0.15
DIXMAANK	3000	62	75	199	1.000000D+00	0.21
DIXMAANL	1500	53	55	130	1.000001D+00	0.09
DIXMAANL	3000	55	57	149	1.000000D+00	0.19
DQDRTIC	1000	33	274	34	7.461713D−26	0.03
DQDRTIC	10,000	102	868	103	2.426640D−27	0.44
DQRTIC	1000	22	81	40	2.784985D−02	0.02
DQRTIC	10,000	31	111	60	4.932478D−01	0.06
EDENSCH	1000	21	89	27	6.003285D+03	0.02
EDENSCH	10,000	18	85	23	6.000328D+04	0.09
ENGVAL1	1000	11	34	16	1.108195D+03	0.01
ENGVAL1	10,000	12	36	19	1.109926D+04	0.05
FLETCBV2	1000	1	1	0	−5.013384D−01	0.00

Table 2 continued

PROBLEM	*n*	*outer-it*	*f-eval*	*CG-it*	*opt-val*	*time*
FLETCBV2	10,000	1	1	0	−5.001341D−01	0.00
FLETCHCR	1000	52	344	87	6.453457D−07	0.04
FLETCHCR	10,000	117	1085	143	2.745120D−06	0.54
FMINSURF	1024	93	207	288	1.000000D+00	0.17
FMINSURF	5625	235	710	680	1.000000D+00	2.65
FREUROTH	1000	38	300	50	1.214697D+05	0.04
FREUROTH	10,000	107	1052	119	1.216521D+06	0.63
LIARWHD	1000	42	251	61	8.352643D−19	0.03
LIARWHD	10,000	112	1107	133	1.455368D−20	0.55
MOREBV	1000	8	8	28	2.148161D−08	0.02
MOREBV	10,000	2	2	7	2.428066D−09	0.01
NONDIA	1000	22	256	27	6.680969D−21	0.04
NONDQUAR	1000	45	111	111	1.425631D−04	0.04
NONDQUAR	10,000	46	175	98	3.744353D−04	0.18
PENALTY1	10,000	54	81	121	9.900151D−02	0.20
POWELLSG	1000	46	257	86	1.992056D−08	0.05
POWELLSG	10,000	114	783	151	7.735314D−08	0.25
POWER	1000	65	189	142	5.912729D−09	0.08
POWER	10,000	233	891	559	9.025072D−09	1.08
QUARTC	1000	22	81	40	2.784985D−02	0.02
QUARTC	10,000	31	111	60	4.932478D−01	0.07
SCHMVETT	1000	14	35	37	−2.994000D+03	0.03
SCHMVETT	10,000	19	69	38	−2.999400D+04	0.20
SINQUAD	1000	37	310	49	−2.942505D+05	0.04
SINQUAD	10,000	104	1517	111	−2.642315D+07	1.01
SPARSQUR	1000	22	66	34	6.266490D−09	0.02
SPARSQUR	10,000	22	67	39	1.069594D−08	0.18
SROSENBR	1000	35	309	40	2.842418D−22	0.02
SROSENBR	10,000	104	920	108	9.421397D−12	0.24
TESTQUAD	1000	12,401	12,950	42,766	1.636783D−05	12.76
TOINTGSS	1000	2	3	1	1.001002D+01	0.00
TOINTGSS	10,000	2	3	1	1.000100D+01	0.00
TQUARTIC	10,000	14	144	18	1.145916D−11	0.05
TRIDIA	1000	244	635	738	7.979032D−06	0.27
TRIDIA	10,000	1764	3391	5764	6.817977D−06	10.49
VARDIM	1000	37	37	72	1.058565D−20	0.03
VARDIM	10,000	54	298	99	4.475275D−18	0.17
VAREIGVL	1000	24	49	74	2.351034D−08	0.04
VAREIGVL	10,000	21	179	22	3.924839D−16	0.15

Table 3 Results for the preconditioned Truncated Newton method with $\delta = 100$

PROBLEM	*n*	*outer-it*	*f-eval*	*CG-it*	*opt-val*	*time*
ARWHEAD	1000	34	364	37	0.000000D+00	0.03
ARWHEAD	10,000	10	102	11	1.332134D−11	0.07
BDQRTIC	1000	46	293	84	3.983818D+03	0.08
BDQRTIC	10,000	121	1217	204	4.003431D+04	1.19
BRYBND	1000	20	64	26	6.709348D−12	0.02
BRYBND	10,000	20	64	26	6.226697D−12	0.15
CRAGGLVY	1000	49	216	94	3.364231D+02	0.10
CRAGGLVY	10,000	116	776	173	3.377956D+03	1.23
CURLY10	1000	759	1046	2771	−1.003163D+05	3.47
CURLY10	10,000	2246	3406	8576	−1.003163D+06	43.37
DIXMAANA	1500	8	13	9	1.000000D+00	0.00
DIXMAANA	3000	8	14	8	1.000000D+00	0.01
DIXMAANB	1500	5	10	6	1.000000D+00	0.00
DIXMAANB	3000	5	10	6	1.000000D+00	0.01
DIXMAANC	1500	5	11	6	1.000000D+00	0.00
DIXMAANC	3000	5	11	6	1.000000D+00	0.01
DIXMAAND	1500	5	8	5	1.000000D+00	0.00
DIXMAAND	3000	5	8	5	1.000000D+00	0.00
DIXMAANE	1500	72	75	161	1.000000D+00	0.16
DIXMAANE	3000	107	110	242	1.000000D+00	0.38
DIXMAANF	1500	49	54	130	1.000000D+00	0.14
DIXMAANF	3000	55	60	148	1.000000D+00	0.31
DIXMAANH	1500	50	52	124	1.000000D+00	0.14
DIXMAANH	3000	53	55	148	1.000000D+00	0.30
DIXMAANI	1500	222	225	715	1.000001D+00	0.93
DIXMAANI	3000	261	264	844	1.000002D+00	1.65
DIXMAANK	1500	56	70	169	1.000000D+00	0.24
DIXMAANK	3000	50	63	169	1.000000D+00	0.32
DIXMAANL	1500	46	48	115	1.000001D+00	0.17
DIXMAANL	3000	59	61	170	1.000000D+00	0.35
DQDRTIC	1000	33	274	34	7.461713D−26	0.02
DQDRTIC	10,000	102	868	103	2.426640D−27	0.66
DQRTIC	1000	22	81	40	2.784985D−02	0.02
DQRTIC	10,000	31	111	60	4.932478D−01	0.12
EDENSCH	1000	21	89	27	6.003285D+03	0.02
EDENSCH	10,000	18	85	23	6.000328D+04	0.12
ENGVAL1	1000	11	34	16	1.108195D+03	0.01
ENGVAL1	10,000	12	36	19	1.109926D+04	0.07
FLETCBV2	1000	1	1	0	−5.013384D−01	0.00

Table 3 continued

PROBLEM	*n*	*outer it*	*f-eval*	*CG-it*	*opt-val*	*time*
FLETCBV2	10,000	1	1	0	−5.001341D−01	0.00
FLETCHCR	1000	47	339	77	7.006731D−06	0.06
FLETCHCR	10,000	113	1080	134	1.006835D−05	0.77
FMINSURF	1024	78	185	236	1.000000D+00	0.55
FMINSURF	5625	216	615	719	1.000000D+00	6.03
FREUROTH	1000	38	300	50	1.214697D+05	0.05
FREUROTH	10,000	107	1052	119	1.216521D+06	0.88
LIARWHD	1000	42	251	61	8.352643D−19	0.05
LIARWHD	10,000	112	1107	133	1.455368D−20	0.76
MOREBV	1000	8	8	28	2.148161D−08	0.03
MOREBV	10,000	2	2	7	2.428066D−09	0.02
NONDIA	1000	22	256	27	6.680969D−21	0.02
NONDQUAR	1000	60	126	143	7.704317D−05	0.14
NONDQUAR	10,000	52	182	111	2.460212D−04	0.39
PENALTY1	10,000	54	81	121	9.900151D−02	0.30
POWELLSG	1000	46	257	86	1.992056D−08	0.04
POWELLSG	10,000	114	783	151	7.735314D−08	0.40
POWER	1000	65	189	142	5.912729D−09	0.12
POWER	10,000	196	854	455	2.103254D−08	1.36
QUARTC	1000	22	81	40	2.784985D−02	0.02
QUARTC	10,000	31	111	60	4.932478D−01	0.12
SCHMVETT	1000	14	35	37	−2.994000D+03	0.04
SCHMVETT	10,000	19	69	38	−2.999400D+04	0.39
SINQUAD	1000	37	310	49	−2.942505D+05	0.06
SINQUAD	10,000	104	1517	111	−2.642315D+07	1.27
SPARSQUR	1000	22	66	34	6.266490D−09	0.03
SPARSQUR	10,000	22	67	39	1.069594D−08	0.24
SROSENBR	1000	35	309	40	2.842418D−22	0.03
SROSENBR	10,000	104	920	108	9.421397D−12	0.40
TESTQUAD	1000	1346	1895	4853	5.716667D−06	4.34
TOINTGSS	1000	2	3	1	1.001002D+01	0.00
TOINTGSS	10,000	2	3	1	1.000100D+01	0.01
TQUARTIC	10,000	14	144	18	1.145916D−11	0.08
TRIDIA	1000	124	515	334	6.644750D−07	0.32
TRIDIA	10,000	386	2013	1179	9.438427D−07	4.21
VARDIM	1000	37	37	72	1.058565D−20	0.04
VARDIM	10,000	54	298	99	4.475275D−18	0.32
VAREIGVL	1000	24	49	74	2.351034D−08	0.09
VAREIGVL	10,000	21	179	22	3.924839D−16	0.20

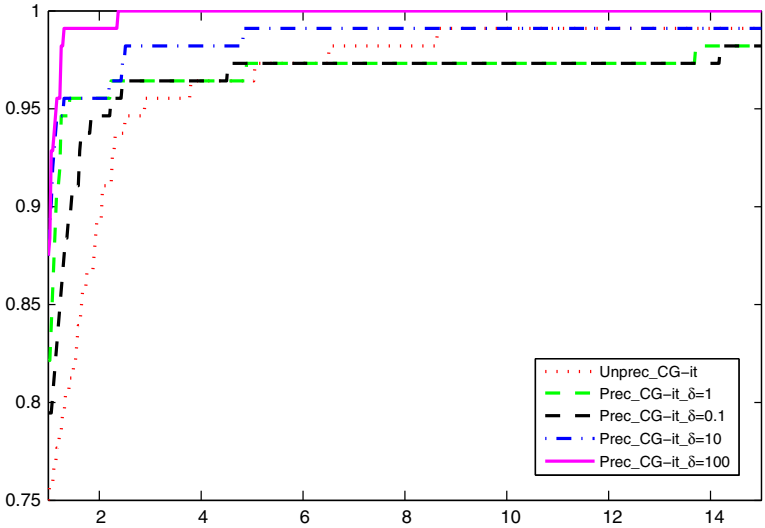

Fig. 7 Performance (full) profile for a comparison in terms of number of inner iterations, on 78 medium-large scale convex unconstrained CUTEst problems, using the Truncated Newton scheme in Table 1. Here $M_h^\sharp(0, \delta)$ with different values of $\delta \in \{0.1, 1, 10, 100\}$ and no preconditioner (Unprec) are adopted to solve Newton's equation

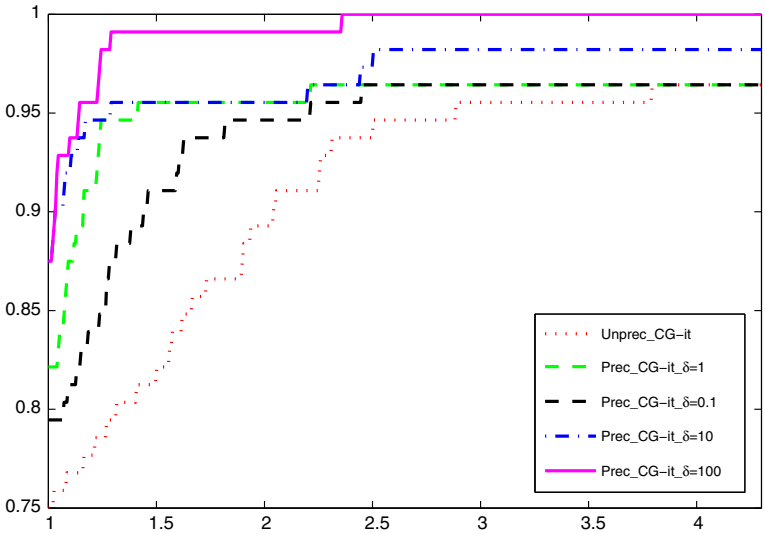

Fig. 8 Performance (detail) profile for a comparison in terms of number of inner iterations, on 78 medium-large scale convex unconstrained CUTEst problems, using the Truncated Newton scheme in Table 1. Here $M_h^\sharp(0, \delta)$ with different values of $\delta \in \{0.1, 1, 10, 100\}$ and no preconditioner (Unprec) are adopted to solve Newton's equation

- for $\delta = 0.1$, due to the clustering of the singular values of $M_h^\sharp(a, \delta)\nabla^2 f(z_k)$, in accordance with the considerations in the previous items, we first observe a similar clustering of the eigenvalues of $M_h^\sharp(a, \delta)\nabla^2 f(z_k)$. In addition, by the analysis in

Proposition 4.2 of [12], the following bound on the condition number (and consequently on the spectral condition number) of $M_h^\sharp(a,\delta)\nabla^2 f(z_k)$ is obtained

$$\kappa(M_h^\sharp(a,\delta)\nabla^2 f(z_k)) \le \xi_h \cdot \kappa\left(\nabla^2 f(z_k)\right), \qquad \xi_h > 0.$$

Then, typically for small values of δ the quantity ξ_h (see (4.7) in [12] for the expression of ξ_h) outreaches its minimum and the bound becomes tighter. However, for completeness we report that in our experience small values of δ tend to be non-competitive with larger ones, since they might also yield correspondingly small eigenvalues for $\delta^2 T_h$ in (3.1).

7 Conclusions

We have given theoretical and numerical results for a new class of preconditioners. The latter can be built by using any Krylov-subspace method for the positive definite linear system (2.1), as well as L-BFGS updates, provided that the general condition (2.2) is satisfied. We gave evidence that on several test problems and real applications, a few iterations of the Krylov-subspace method adopted may suffice to compute effective preconditioners. In particular, in many problems using a relatively small value of the index h, a significant information on the system matrix A can be captured.

On this guideline our proposal might possibly be promising also for those cases where a sequence of linear systems of the form

$$A_k x = b_k, \qquad k = 1, 2, \ldots \tag{7.1}$$

requires a solution (e.g., see also [7,19] for details), where A_k "slightly changes" with the index k. In the latter case, the preconditioners $M_h^\sharp(a,\delta)$ in (3.1)–(3.2) can be computed applying the Krylov-subspace method to the first linear system $A_1 x = b_1$. Then, the resulting preconditioners can be used to efficiently solve (7.1) for $k = 2, 3, \ldots$

A full investigation was also included, where our proposal was compared with the preconditioner in [11], showing that the new proposal outperforms that in [11]. In particular we think that a further exhaustive analysis is required to extend the class (3.1)–(3.2) to indefinite linear systems, so that a fully general proposal might be available. In the latter case, both a new theoretical approach and a completely novel numerical experience are sought, in order to show the possible robustness of our class of preconditioners.

Acknowledgments The authors wish to thank Mehiddin Al-Baali, for his valuable comments when the contents in this paper were at their early beginning. G. Fasano wishes to thank the National Research Council-Marine Technology Research Institute (CNR-INSEAN), for the indirect support in project RITMARE 2012-2016. The authors are also indebted to an anonymous referee for the helpful suggestions and comments which led to improve the analysis in the paper.

Appendix

The next lemma is used to prove the results of Theorem 4.3.

Lemma 7.1 *Given the symmetric matrices $H \in \mathbb{R}^{h\times h}$, $P \in \mathbb{R}^{(n-h)\times(n-h)}$ and the matrix $\Phi \in \mathbb{R}^{h\times(n-h)}$, suppose*

$$\Phi^T H = \begin{bmatrix} z_1^T \\ \vdots \\ z_m^T \\ 0_{[n-(h+m)],h} \end{bmatrix}, \qquad z_1, \dots, z_m \in \mathbb{R}^h, \tag{7.2}$$

with $H = \lambda[I_h + u_1 w_1^T + \cdots + u_p w_p^T]$, $p \le h$, $0 \le m \le h-p$, $\lambda \in \mathbb{R}$, $u_i, w_i \in \mathbb{R}^h$, $i = 1, \dots, p$. Then, the symmetric matrix

$$\left(\begin{array}{c|c} H & H\Phi \\ \hline \Phi^T H & P \end{array}\right) \tag{7.3}$$

has the eigenvalue λ with multiplicity at least equal to $h - rk[w_1\, w_2 \cdots w_p\, z_1\, z_2 \cdots z_m]$.

Proof Observe that H has the eigenvalue λ with a multiplicity at least $h-p$, since $Hs = \lambda s$ for any $s \perp span\{w_1, \dots, w_p\}$. Moreover, imposing the condition (with $x_1,\ x_2$ not simultaneously zero vectors)

$$\left(\begin{array}{c|c} H & H\Phi \\ \hline \Phi^T H & P \end{array}\right) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \lambda \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

is equivalent to impose the conditions

$$\begin{cases} H(x_1 + \Phi x_2) = \lambda x_1 \\ \\ \Phi^T H x_1 + P x_2 = \lambda x_2. \end{cases}$$

By (7.2), choosing $x_2 = 0$ and x_1 any h-real vector such that $x_1 \perp span\{w_1, \dots, w_p, z_1, \dots, z_m\}$, then λ is eigenvalue of (7.3) with multiplicity given by h minus the largest number of linearly independent vectors in the set $\{w_1, \dots, w_p, z_1, \dots, z_m\}$. □

Proof of Theorem 4.3 Let $N = \left[R_h \mid u_{h+1} \mid R_{n,h+1}\right]$, where $R_{n,h+1} = (u_{h+2} \mid \cdots \mid u_n) \in \mathbb{R}^{n\times(n-h-1)}$, being u_j, $j = h+2, \dots, n$, orthonormal vectors and $(R_h \mid u_{h+1})^T R_{n,h+1} = 0$. Hence, N is orthogonal. Observe that for $h \le n-1$ the preconditioners $M_h^\sharp(a, \delta)$ may be rewritten as

$$M_h^\sharp(a,\delta) = N \left[\begin{array}{c|c} \left(\begin{array}{c|c} \delta^2 T_h & a e_h \\ \hline a e_h^T & 1 \end{array}\right)^{-1} & 0 \\ \hline 0 & I_{n-(h+1)} \end{array}\right] N^T, \qquad h \le n-1. \tag{7.4}$$

The property *a)* follows from the symmetry of T_h. In addition, observe that $R_{n,h+1}^T R_{n,h+1} = I_{n-(h+1)}$. Thus, from (7.4) the matrix $M_h^\sharp(a,\delta)$ is nonsingular if and only if the matrix

$$\left(\begin{array}{c|c} \delta^2 T_h & a e_h \\ \hline a e_h^T & 1 \end{array}\right) \tag{7.5}$$

is invertible. Furthermore, by a direct computation we observe that for $h \le n-1$ the following identity holds (we recall that since $A \succ 0$ then by (2.2) $T_h \succ 0$, too)

$$\left(\begin{array}{c|c} \delta^2 T_h & a e_h \\ \hline a e_h^T & 1 \end{array}\right) = \left(\begin{array}{c|c} I_h & 0 \\ \hline \frac{a}{\delta^2} e_h^T T_h^{-1} & 1 \end{array}\right) \left(\begin{array}{c|c} \delta^2 T_h & 0 \\ \hline 0 & 1 - \frac{a^2}{\delta^2} e_h^T T_h^{-1} e_h \end{array}\right) \left(\begin{array}{c|c} I_h & \frac{a}{\delta^2} T_h^{-1} e_h \\ \hline 0 & 1 \end{array}\right). \tag{7.6}$$

Thus, since T_h is nonsingular and $\delta \neq 0$, for $h \le n-1$ the determinant of matrix (7.5) is nonzero if and only if $a \neq \pm\delta(e_h^T T_h^{-1} e_h)^{-1/2}$. Finally, for $h = n$ the matrix $M_h^\sharp(a,\delta)$ is nonsingular, since R_n and T_n are nonsingular in (3.2).

As regards (b), observe that from (7.4) the matrix $M_h^\sharp(a,\delta)$ is positive definite as long as the matrix (7.5) is positive definite. Thus, from (7.6) and relation $T_h \succ 0$ we immediately infer that $M_h^\sharp(a,\delta)$ is positive definite as long as $|a| < |\delta|(e_h^T T_h^{-1} e_h)^{-1/2}$. Moreover, we recall that N is orthogonal.

Item (c) may be proved considering the eigenvalues of the matrix

$$\left[M_h^\sharp(a,\delta)A\right]\left[M_h^\sharp(a,\delta)A\right]^T = M_h^\sharp(a,\delta)A^2 M_h^\sharp(a,\delta),$$

i.e., the singular values of $M_h^\sharp(a,\delta)A$. On this purpose, for $h \le n-1$ we have for $M_h^\sharp(a,\delta)A^2 M_h^\sharp(a,\delta)$ the expression (see (7.4))

$$\begin{aligned} &M_h^\sharp(a,\delta)A^2 M_h^\sharp(a,\delta) \\ &= N \left[\begin{array}{c|c} \left(\begin{array}{c|c} \delta^2 T_h & a e_h \\ \hline a e_h^T & 1 \end{array}\right)^{-1} & 0 \\ \hline 0 & I_{n-(h+1)} \end{array}\right] C \left[\begin{array}{c|c} \left(\begin{array}{c|c} \delta^2 T_h & a e_h \\ \hline a e_h^T & 1 \end{array}\right)^{-1} & 0 \\ \hline 0 & I_{n-(h+1)} \end{array}\right] N^T \end{aligned} \tag{7.7}$$

where $C \in \mathbb{R}^{n\times n}$, with

$$C = N^T A^2 N = \left[\begin{array}{c|c|c} R_h^T A^2 R_h & R_h^T A^2 u_{h+1} & R_h^T A^2 R_{n,h+1} \\ \hline u_{h+1}^T A^2 R_h & u_{h+1}^T A^2 u_{h+1} & u_{h+1}^T A^2 R_{n,h+1} \\ \hline R_{n,h+1}^T A^2 R_h & R_{n,h+1}^T A^2 u_{h+1} & R_{n,h+1}^T A^2 R_{n,h+1} \end{array}\right].$$

From (2.2) and the symmetry of T_h we obtain

$$\begin{aligned} R_h^T A^2 R_h &= (AR_h)^T (AR_h) \;=\; (R_h T_h + \rho_{h+1} u_{h+1} e_h^T)^T (R_h T_h + \rho_{h+1} u_{h+1} e_h^T) \\ &= T_h^2 + \rho_{h+1}^2 e_h e_h^T \end{aligned} \tag{7.8}$$

$$R_h^T A^2 u_{h+1} = (AR_h)^T A u_{h+1} \;=\; v_1 \in \mathbb{R}^h, \tag{7.9}$$

and considering relation (2.2) we obtain

$$\begin{aligned} AR_{h+1} &= A(R_h \mid u_{h+1}) \;=\; R_{h+1}T_{h+1} + \rho_{h+2}u_{h+2}e_{h+1}^T \\ &= (R_h \mid u_{h+1}) \left(\begin{array}{c|c} T_h & \rho_{h+1}e_h \\ \hline \rho_{h+1}e_h^T & t_{h+1,h+1} \end{array} \right) + \rho_{h+2}u_{h+2}e_{h+1}^T \end{aligned}$$

i.e.

$$\begin{aligned} AR_h &= R_hT_h + \rho_{h+1}u_{h+1}e_h^T \\ Au_{h+1} &= \rho_{h+1}u_h + t_{h+1,h+1}u_{h+1} + \rho_{h+2}u_{h+2}, \end{aligned} \tag{7.10}$$

so that

$$\begin{aligned} A^2R_h &= (AR_h)T_h + \rho_{h+1}Au_{h+1}e_h^T \\ &= (R_hT_h + \rho_{h+1}u_{h+1}e_h^T)T_h + \rho_{h+1}(\rho_{h+1}u_h + t_{h+1,h+1}u_{h+1} + \rho_{h+2}u_{h+2})e_h^T. \end{aligned}$$

As a consequence, from (7.10) we also have that $Au_{h+2} = span\{u_{h+1}, u_{h+2}, u_{h+3}\}$ and

$$\begin{aligned} R_h^TA^2R_{n,h+1} &= (A^2R_h)^TR_{n,h+1} = \rho_{h+1}(\rho_{h+2}u_{h+2}e_h^T)^TR_{n,h+1} \\ &= \rho_{h+1}\rho_{h+2}\begin{pmatrix} 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \end{pmatrix} = \rho_{h+1}\rho_{h+2}E_{h,1} \in \mathbb{R}^{h\times[n-(h+1)]}, \\ u_{h+1}^TA^2u_{h+1} &= c > 0 \\ u_{h+1}^TA^2R_{n,h+1} &= \left[A(\rho_{h+1}u_h + t_{h+1,h+1}u_{h+1} + \rho_{h+2}u_{h+2})\right]^T R_{n,h+1} \\ &= \left[A(t_{h+1,h+1}u_{h+1} + \rho_{h+2}u_{h+2})\right]^T R_{n,h+1} = (\alpha\ \beta\ 0 \cdots 0) \in \mathbb{R}^{n-(h+1)} \end{aligned}$$

with $\alpha, \beta \in \mathbb{R}$ and

$$R_{n,h+1}^TA^2R_{n,h+1} = V_2 \in \mathbb{R}^{[n-(h+1)]\times[n-(h+1)]},$$

where $E_{i,j}$ has all zero entries but $+1$ at position (i, j). Thus,

$$C = \left[\begin{array}{c|c|c} T_h^2 + \rho_{h+1}^2 e_h e_h^T & v_1 & \rho_{h+1}\rho_{h+2}E_{h,1} \\ \hline v_1^T & c & \alpha\ \ \beta\ \ 0\ \cdots\ 0 \\ \hline & \alpha & \\ & \beta & \\ \rho_{h+1}\rho_{h+2}E_{1,h} & 0 & V_2 \\ & \vdots & \\ & 0 & \end{array} \right].$$

Moreover, from (7.6) we can readily infer that

$$\left[\begin{array}{c|c} \delta^2 T_h & a e_h \\ \hline a e_h^T & 1 \end{array}\right]^{-1} = \left(\begin{array}{c|c} I_h & -\frac{a}{\delta^2} T_h^{-1} e_h \\ \hline 0 & 1 \end{array}\right) \left(\begin{array}{c|c} \frac{1}{\delta^2} T_h^{-1} & 0 \\ \hline 0 & \frac{1}{1-\frac{a^2}{\delta^2} e_h^T T_h^{-1} e_h} \end{array}\right) \left(\begin{array}{c|c} I_h & 0 \\ \hline -\frac{a}{\delta^2} e_h^T T_h^{-1} & 1 \end{array}\right)$$
$$= \left(\begin{array}{c|c} \frac{1}{\delta^2} T_h^{-1} - \frac{a}{\delta^4} \omega T_h^{-1} e_h e_h^T T_h^{-1} & \frac{\omega}{\delta^2} T_h^{-1} e_h \\ \hline \frac{\omega}{\delta^2} e_h^T T_h^{-1} & -\frac{\omega}{a} \end{array}\right), \tag{7.11}$$

with

$$\omega = -\frac{a}{1 - \frac{a^2}{\delta^2} e_h^T T_h^{-1} e_h}. \tag{7.12}$$

Now, recalling that $N = \left[R_h \mid u_{h+1} \mid R_{n,h+1}\right]$, for any $h \leq n-1$ we obtain from (7.7)

$$M_h^\sharp(a,\delta) A^2 M_h^\sharp(a,\delta)$$
$$= N \left[\begin{array}{c|c} \left[\begin{array}{c|c} \delta^2 T_h & a e_h \\ \hline a e_h^T & 1 \end{array}\right]^{-1} \left[\begin{array}{c|c} T_h^2 + \rho_{h+1}^2 e_h e_h^T & v_1 \\ \hline v_1^T & c \end{array}\right] \left[\begin{array}{c|c} \delta^2 T_h & a e_h \\ \hline a e_h^T & 1 \end{array}\right]^{-1} & \left(\begin{array}{c|c} \delta^2 T_h & a e_h \\ \hline a e_h^T & 1 \end{array}\right)^{-1} \left(\begin{array}{c} \rho_{h+1}\rho_{h+2} E_{h,1} \\ \hline \alpha\ \beta\ 0\ \cdots\ 0 \end{array}\right) \\ \hline \left(\begin{array}{c} \rho_{h+1}\rho_{h+2} E_{h,1} \\ \hline \alpha\ \beta\ 0\ \cdots\ 0 \end{array}\right)^T \left(\begin{array}{c|c} \delta^2 T_h & a e_h \\ \hline a e_h^T & 1 \end{array}\right)^{-1} & V_2 \end{array}\right] N^T,$$

with

$$\left(\begin{array}{c|c} \delta^2 T_h & a e_h \\ \hline a e_h^T & 1 \end{array}\right)^{-1} \left(\begin{array}{c} \rho_{h+1}\rho_{h+2} E_{h,1} \\ \hline \alpha\ \beta\ 0\ \cdots\ 0 \end{array}\right) = \left(\begin{array}{cc} * & * \\ \vdots & \vdots \\ * & * \end{array} \quad 0_{h+1,[n-(h+3)]}\right) \in \mathbb{R}^{(h+1)\times[n-(h+1)]},$$

where the '$*$' indicates entries whose computation is not relevant to our purposes.

Now, considering the second last relation, we focus on computing the submatrix $H_{h\times h}$ corresponding to the first h rows and h columns of the matrix

$$\left[\begin{array}{c|c} \delta^2 T_h & a e_h \\ \hline a e_h^T & 1 \end{array}\right]^{-1} \left[\begin{array}{c|c} T_h^2 + \rho_{h+1}^2 e_h e_h^T & v_1 \\ \hline v_1^T & c \end{array}\right] \left[\begin{array}{c|c} \delta^2 T_h & a e_h \\ \hline a e_h^T & 1 \end{array}\right]^{-1}. \tag{7.13}$$

After a brief computation, from (7.11) and (7.13) we obtain for the submatrix $H_{h\times h}$

$$\begin{aligned} H_{h\times h} = & \left[\left(\frac{1}{\delta^2} T_h^{-1} - \frac{a}{\delta^4} \omega T_h^{-1} e_h e_h^T T_h^{-1}\right)\left(T_h^2 + \rho_{h+1}^2 e_h e_h^T\right)\right. \\ & \left. + \frac{\omega}{\delta^2} T_h^{-1} e_h v_1^T\right] \cdot \left[\frac{1}{\delta^2} T_h^{-1} - \frac{a}{\delta^4} \omega T_h^{-1} e_h e_h^T T_h^{-1}\right] \\ & + \left[\left(\frac{1}{\delta^2} T_h^{-1} - \frac{a}{\delta^4} \omega T_h^{-1} e_h e_h^T T_h^{-1}\right) v_1 + \frac{\omega}{\delta^2} c T_h^{-1} e_h\right] \cdot \frac{\omega}{\delta^2} e_h^T T_h^{-1}, \end{aligned}$$

so that

$$
\begin{aligned}
H_{h\times h} = \Bigg[& \frac{1}{\delta^2} T_h + \frac{\rho_{h+1}^2}{\delta^2} T_h^{-1} e_h e_h^T - \frac{a}{\delta^4} \omega T_h^{-1} e_h e_h^T T_h \\
& - \frac{a}{\delta^4} \omega \rho_{h+1}^2 (e_h^T T_h^{-1} e_h) T_h^{-1} e_h e_h^T + \frac{\omega}{\delta^2} T_h^{-1} e_h v_1^T \Bigg] \\
& \times \left[\frac{1}{\delta^2} T_h^{-1} - \frac{a}{\delta^4} \omega T_h^{-1} e_h e_h^T T_h^{-1} \right] \\
& + \frac{\omega}{\delta^2} \left[\frac{1}{\delta^2} T_h^{-1} v_1 - \frac{a}{\delta^4} \omega (e_h^T T_h^{-1} v_1) T_h^{-1} e_h + \frac{\omega}{\delta^2} c T_h^{-1} e_h \right] e_h^T T_h^{-1}.
\end{aligned}
$$

From the last relation we finally have for $H_{h\times h}$ the expression

$$
\begin{aligned}
H_{h\times h} = \frac{1}{\delta^4} \Bigg\{ & I_h + \left[\eta T_h^{-1} e_h - \frac{a\omega}{\delta^2} e_h + \omega T_h^{-1} v_1 \right] e_h^T T_h^{-1} \\
& + \omega T_h^{-1} e_h \left[v_1^T T_h^{-1} - \frac{a}{\delta^2} e_h^T \right] \Bigg\},
\end{aligned} \tag{7.14}
$$

where

$$
\begin{aligned}
\eta = \rho_{h+1}^2 & - 2 \frac{a}{\delta^2} \omega \rho_{h+1}^2 \left(e_h^T T_h^{-1} e_h \right) + \frac{a^2 \omega^2}{\delta^4} \\
& + \frac{a^2}{\delta^4} \omega^2 \rho_{h+1}^2 \left(e_h^T T_h^{-1} e_h \right)^2 - 2 \frac{a}{\delta^2} \omega^2 \left(e_h^T T_h^{-1} v_1 \right) + \omega^2 c;
\end{aligned} \tag{7.15}
$$

moreover, since $M_h^\sharp(a,\delta) A^2 M_h^\sharp(a,\delta) \succ 0$ then also $H_{h\times h}$ is positive definite.

Let us now define the subspace (see the vectors which define the dyads in relation (7.14))

$$
\mathcal{T}_2 = span \left\{ T_h^{-1} e_h \ , \ \omega \left[T_h^{-1} v_1 - \frac{a}{\delta^2} e_h \right] \right\}. \tag{7.16}
$$

Observe that, by (7.9) and (7.10), after some computation $v_1 = \rho_{h+1} \big[T_h + t_{h+1,h+1} I_h \big] e_h$. Thus, from (7.16) the subspace $\mathcal{T}_2$ has dimension 2, unless

(i) T_h is proportional to I_h,
(ii) $a = 0$ (which, from (7.12), also implies $\omega = 0$).

We analyze separately the two cases. The condition (i) cannot hold since (2.2) would imply that the vector Au_i is proportional to u_i, $i = 1, \ldots, h-1$, i.e. the Krylov-subspace method had to stop at the very first iteration, since the Krylov-subspace generated at the first iteration did not change. As a consequence, considering any subspace $\mathcal{S}_{h-2} \subseteq \mathbb{R}^n$, such that $\mathcal{S}_{h-2} \bigoplus \mathcal{T}_2 = \mathbb{R}^h$, we can select any orthonormal basis $\{s_1, \ldots, s_{h-2}\}$ of the subspace $\mathcal{S}_{h-2}$ so that (see (7.14)) the $h-2$ vectors $\{s_1, \ldots, s_{h-2}\}$ can be thought as (the first) $h-2$ eigenvectors of the matrix $H_{h\times h}$, corresponding to the eigenvalue $+1/\delta^4$.

Now, from the formula after (7.12) the eigenvalues of $M_h^\sharp(a,\delta)A^2M_h^\sharp(a,\delta)$ coincide with the eigenvalues of (we recall that since $M_h^\sharp(a,\delta)A^2M_h^\sharp(a,\delta)\succ 0$ then $H_{h\times h}\succ 0$)

$$\left(\begin{array}{c|c} H_{h\times h} & H_{h\times h}\Phi \\ \hline \Phi^T H_{h\times h} & \begin{array}{c|c} * & *\cdots\cdots\cdots * \\ \hline * & \\ \vdots & V_2 \\ * & \end{array} \end{array}\right), \quad \Phi = H_{h\times h}^{-1}\left(z_1\ z_2\ z_3\ 0_{h\times[n-(h+3)]}\right), \quad z_1,z_2,z_3\in\mathbb{R}^h, \tag{7.17}$$

which becomes, after setting

$$P=\left(\begin{array}{c|c} * & *\cdots\cdots\cdots * \\ \hline * & \\ \vdots & V_2 \\ * & \end{array}\right),$$

of the form

$$\left[\begin{array}{c|c} H_{h\times h} & H_{h\times h}\Phi \\ \hline \Phi^T H_{h\times h} & P \end{array}\right].$$

Thus, using Lemma 7.1 with $w_1=T_h^{-1}e_h$, $w_2=\omega\left[T_h^{-1}v_1-a/\delta^2 e_h\right]$ and $m=3$, recalling that $T_h\succ 0$, and observing that we have by (7.11)

$$\begin{aligned}\left[\begin{array}{c}z_1\\ \hline *\end{array}\right] &= \left[\begin{array}{c|c}\delta^2 T_h & ae_h\\ \hline ae_h^T & 1\end{array}\right]^{-1}\left[\begin{array}{c|c}T_h^2+\rho_{h+1}^2 e_h e_h^T & v_1\\ \hline v_1^T & c\end{array}\right]\left[\begin{array}{c|c}\delta^2 T_h & ae_h\\ \hline ae_h^T & 1\end{array}\right]^{-1} e_{h+1}\\ &= \left[\begin{array}{c|c}\frac{1}{\delta^2}T_h^{-1}-\frac{a}{\delta^4}\omega T_h^{-1}e_h e_h^T T_h^{-1} & \frac{\omega}{\delta^2}T_h^{-1}e_h\\ \hline \frac{\omega}{\delta^2}e_h^T T_h^{-1} & -\frac{\omega}{a}\end{array}\right]\\ &\quad\times\left(\begin{array}{c}\frac{\omega}{\delta^2}T_h^2 T_h^{-1}e_h+\rho_{h+1}^2\frac{\omega}{\delta^2}(e_h^T T_h^{-1}e_h)e_h-\frac{\omega}{a}v_1\\ \hline \frac{\omega}{\delta^2}(v_1^T T_h^{-1}e_h)-\frac{c\omega}{a}\end{array}\right)\end{aligned}$$

so that $z_1\in span\left\{\omega e_h\ ,\ \omega T_h^{-1}e_h\ ,\ T_h^{-1}v_1\right\}$,

$$\begin{aligned}\left[\begin{array}{c}z_2\\ \hline *\end{array}\right] &= \left[\begin{array}{c|c}\delta^2 T_h & ae_h\\ \hline ae_h^T & 1\end{array}\right]^{-1}\left(\begin{array}{c}0\\ \vdots\\ 0\\ \rho_{h+1}\rho_{h+2}\\ \alpha\end{array}\right)\\ &= \left[\begin{array}{c}\frac{\rho_{h+1}\rho_{h+2}}{\delta^2}T_h^{-1}e_h-\rho_{h+1}\rho_{h+2}\frac{a\omega}{\delta^4}(e_h^T T_h^{-1}e_h)T_h^{-1}e_h+\frac{\alpha\omega}{\delta^2}T_h^{-1}e_h\\ \hline *\end{array}\right]\end{aligned}$$

so that $z_2 \in span\{T_h^{-1}e_h\}$, and

$$\left[\begin{array}{c} z_3 \\ \hline * \end{array}\right] = \left[\begin{array}{c|c} \delta^2 T_h & ae_h \\ \hline ae_h^T & 1 \end{array}\right]^{-1} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \beta \end{pmatrix} = \left[\begin{array}{c} \frac{\beta\omega}{\delta^2} T_h^{-1} e_h \\ \hline * \end{array}\right]$$

so that $z_3 \in span\{\omega T_h^{-1}e_h\}$, we conclude that considering the expression of $H_{h\times h}$, at least $h-3$ eigenvalues of $M_h^\sharp(a,\delta)A^2M_h^\sharp(a,\delta)$ coincide with $+1/\delta^4$. As a consequence, the matrix $M_h^\sharp(a,\delta)A$ has at least $h-3$ singular values equal to $+1/\delta^2$, which proves the first statement of (c).

As regards the case (ii) with $a=0$, observe that by the definition (7.12) of ω, $a=0$ implies $\omega=0$. Moreover, recalling that $T_h \succ 0$, from relations (7.14)–(7.15) we have $H_{h\times h} = 1/\delta^4[I_h + \rho_{h+1}^2 T_h^{-1}e_h e_h^T T_h^{-1}]$. Thus, the subspace $\mathcal{T}_2$ in (7.16) reduces to $\mathcal{T}_1 = span\{T_h^{-1}e_h\}$. Now, reasoning as in the case (i), we conclude that the matrix $M_h^\sharp(a,\delta)A$ has at least $(h-2)$ singular values equal to $+1/\delta^2$.

As regards item (d), observe that for $h=n$ the matrix R_n is orthogonal, so that by (2.2) and (3.2) we have

$$M_n^\sharp(a,\delta)A = \frac{1}{\delta^2} R_n T_n^{-1} R_n^T R_n T_n R_n^T = \frac{1}{\delta^2} R_n I_n R_n^T, \tag{7.18}$$

which proves that $M_n^\sharp(a,\delta)A$ has all the n eigenvalues equal to $+1/\delta^2$. □

References

1. Baglama, J., Calvetti, D., Golub, G., Reichel, L.: Adaptively preconditioned GMRES algorithms. SIAM J. Sci. Comput. **20**, 243–269 (1998)
2. Bellavia, S., Gondzio, J., Morini, B.: A matrix-free preconditioner for sparse symmetric positive definite systems and least-squares problems. SIAM J. Sci. Comput. **35**, A192–A211 (2013)
3. Benzi, M.: Preconditioning techniques for large linear systems: a survey. J. Comput. Phys. **182**, 418–477 (2002)
4. Benzi, M., Cullum, J., Tuma, M.: Robust approximate inverse preconditioner for the conjugate gradient method. SIAM J. Sci. Comput. **22**, 1318–1332 (2000)
5. Benzi, M., Tuma, M.: A comparative study of sparse approximate inverse preconditioners. Appl. Numer. Math. **30**, 305–340 (1999)
6. Bernstein, D.: Matrix Mathematics: Theory, Facts, and Formulas, 2nd edn. Princeton University Press, Princeton (2009)
7. Conn, A.R., Gould, N.I.M., Toint, P.L.: Trust-Region Methods. MPS-SIAM Series on Optimization. SIAM, Philadelphia (2000)
8. Dolan, E.D., Moré, J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–213 (2002)
9. Fasano, G.: Planar-conjugate gradient algorithm for large-scale unconstrained optimization, Part 1: Theory. J. Optim. Theory Appl. **125**, 523–541 (2005)
10. Fasano, G., Roma, M.: Iterative computation of negative curvature directions in large scale optimization. Comput. Optim. Appl. **38**, 81–104 (2007)

11. Fasano, G., Roma, M.: Preconditioning Newton-Krylov methods in non-convex large scale optimization. Comput. Optim. Appl. **56**, 253–290 (2013)
12. Fasano, G., Roma, M.: An estimation of the condition number for a class of indefinite preconditioned matrices. Technical Report n.1–2015, Dipartimento di Ingegneria Informatica, Automatica e Gestionale, SAPIENZA, Università di Roma, Italy (2015). http://www.dis.uniroma1.it/~bibdis
13. Golub, G., Van Loan, C.: Matrix Computations, 4th edn. The John Hopkins Press, Baltimore (2013)
14. Gondzio, J.: Matrix-free interior point method. Comput. Optim. Appl. **51**, 457–480 (2013)
15. Gould, N.I.M., Orban, D., Toint, P.L.: CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization. Comput. Optim. Appl. **60**, 545–557 (2015)
16. 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)
17. Hestenes, M.: Conjugate Direction Methods in Optimization. Springer, New York (1980)
18. Lukšan, L., Matonoha, C., Vlček, J.: Band preconditioners for the matrix-free truncated Newton method, Technical Report V-1079, Institute of Computer Science AS CR, Pod Vodarenskou Vezi 2, 18207 Prague 8 (2010)
19. Morales, J., Nocedal, J.: Automatic preconditioning by limited memory quasi-Newton updating. SIAM J. Optim. **10**, 1079–1096 (2000)
20. Nash, S.: A survey of truncated-Newton methods. J. Comput. Appl. Math. **124**, 45–59 (2000)
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., Wright, S.: Numerical Optimization. Springer Series in Operations Research and Financial Engineering, 2nd edn. Springer, New York (2000)
23. O'Leary, D., Yeremin, A.: The linear algebra of block quasi-Newton algorithms. Linear Algebra Appl. **212**(213), 153–168 (1994)
24. Saad, Y.: Iterative Methods for Sparse Linear Systems, 2nd edn. SIAM, Philadelphia (2003)
25. Stoer, J.: Solution of large linear systems of equations by conjugate gradient type methods. In: Bachem, A., Grötschel, M., Korte, B. (eds.) Mathematical Programming. The State of the Art, pp. 540–565. Springer, Berlin (1983)