Preconditioning Large Indefinite Linear Systems

Giovanni Fasano[1,2] and Massimo Roma[3]

[1] Department of Management
University Ca' Foscari Venezia
S.Giobbe, Cannaregio, 873
30121 Venice, ITALY
E-mail: fasano@unive.it

[2] The Italian Ship Model Basin - INSEAN, CNR
via di Vallerano, 139
00128 Rome, ITALY

[3] Dipartimento di Informatica e Sistemistica "A. Ruberti"
SAPIENZA, Università di Roma
via Ariosto, 25
00185 Roma, ITALY
E-mail: roma@dis.uniroma1.it

Abstract

After briefly recalling some relevant approaches for preconditioning large symmetric linear systems, we describe a novel class of preconditioners. Our proposal is tailored for large indefinite linear systems, which arise very frequently in many different contexts of numerical analysis and nonlinear optimization. Our preconditioners are built as by–product of the Krylov subspace method used to solve the system. We describe theoretical properties of the class of the preconditioners we propose, namely their capability of both shifting some eigenvalues of the systems matrix to controlled values, and reducing the modulus of the other ones. The results of a numerical experimentation give evidence of the performance of our proposal.

Keywords: Preconditioners, large indefinite linear systems, large scale optimization, Krylov subspace methods, quasi-Newton updates.

1 Introduction

The efficient solution of large linear systems (or a sequence of slowly varying linear systems) is of fundamental importance in many contexts of numerical analysis and nonlinear optimization. In this paper we first recall a few relevant approaches for preconditioning large indefinite linear systems. Observe that in many contexts of numerical analysis and nonlinear optimization, the iterative efficient solution of linear systems and sequences of linear systems is sought (see e.g. http://math.nist.gov/MatrixMarket). Truncated Newton methods in unconstrained

optimization, KKT systems arising in constrained optimization, interior point methods, and PDE constrained optimization are just some examples from optimization.

We first show that using information from quasi-Newton updates may often provide effective preconditioners. The latter are sometimes endowed with theoretical properties related to the spectrum and the condition number of the preconditioned matrix. Then, we describe a novel class of preconditioners for the solution of large indefinite linear systems, without assuming any sparsity pattern of the system matrix.

In particular, the class of preconditioners we propose uses information collected from Krylov subspace methods, in order to assess the structural properties of the system matrix. We iteratively construct our preconditioners either indirectly using (but not performing) a factorization of the system matrix (see, e.g. Fasano and Roma, 2007; Golub and Van Loan, 1996; Stoer, 1983), obtained as by product of Krylov subspace methods, or performing a Jordan canonical form on a *very small size* matrix. We address our preconditioners using a general Krylov subspace method; then, we prove theoretical properties for such preconditioners, and describe results which indicate how to possibly select the parameters which characterize the definition of the preconditioners.

The basic idea of our approach is to apply a Krylov subspace method to generate a positive definite approximation of the inverse of the system matrix. The latter is then used to build our preconditioners, needing to store just a few vectors, without requiring any product of matrices. We assume that the entries of the system matrix are not known and the information necessary to build the preconditioner is gained by using a routine, which computes the product of the system matrix times a vector.

In the paper Fasano and Roma, 2011b, we experiment with our preconditioners both numerical analysis and nonconvex optimization frameworks. In particular, we test our proposal on significant linear systems from the literature. Then, we focus on the so called Newton–Krylov methods, also known as Truncated Newton methods (see Nash, 1985, for a survey). In these contexts, both positive definite and indefinite linear systems are considered.

We recall that in case the optimization problem in hand is nonconvex, i.e. the Hessian matrix of the objective function is possibly indefinite and at least one eigenvalue is negative, the solution of Newton's equations within Truncated Newton schemes may require some care. Indeed, the Krylov subspace method used to solve Newton's equation, should be suitably applied considering that optimization frameworks require the computation of descent directions, which have to satisfy additional properties (Dennis and Schnabel, 1983; Nocedal and Wright, 2000). In this regard our proposal provides a tool, in order to preserve the latter properties.

The paper is organized as follows. In Section 2 we briefly recall relevant approaches from the literature. Then, in Section 3 we describe our class of preconditioners for large indefinite linear systems, by using a general Krylov subspace method. We detail some theoretical properties of our proposal, along with some hints on its calculation. Finally, a section of conclusions and future work completes the paper.

As regards the notations, for a $n \times n$ real matrix M we denote with $\Lambda[M]$ the spectrum of M; I_k is the identity matrix of order k and we use the capital letter T to indicate a *tridiagonal* matrix. Finally, with $C \succ 0$ we indicate that the matrix C is positive definite, and $\|\cdot\|$ denotes the Euclidean norm.

2 Some approaches for preconditioning large symmetric systems

Let us consider the following linear system

$$Ax = b, \qquad A \in \mathbb{R}^{n\times n}, \quad b \in \mathbb{R}^n, \tag{2.1}$$

where A is *symmetric* and *nonsingular*, and n is *large*. We assume that the structure of the matrix A is unknown as well as its sparsity pattern. We recall that in case of special structures of matrix A, suitable preconditioners may be built for solving (2.1) (see Higham, 2002).
As is well-known, the main rationale behind the idea of using a preconditioner to solve the linear system (2.1), consists in introducing the nonsingular matrix M, such that solving

$$MAx = Mb \tag{2.2}$$

is *possibly simpler* in some sense than solving (2.1). Of course, to the latter purpose the extreme choices for M are $M = I$ and $M = A^{-1}$ (the latter being the ideal choice). In most of the cases (always when n is large) there is no chance to compute A^{-1} (or computing A^{-1} is no cheaper than solving (2.1)). Notwithstanding this difficulty, the preconditioner M can be chosen according with the following alternative guidelines:

- the linear system (2.2) should be of easy solution thanks to the structure of matrix M (e.g. preconditioners for linear systems from PDEs discretization have often a suitable block-structure which is suggested by the problem in hand);
- the condition number $\kappa(MA)$ should be relatively small. The latter fact may be helpful when attempting to solve the preconditioned system (2.2), with a technique sensitive to $\kappa(MA)$ (e.g. the Krylov subspace methods);
- the eigenvalues in the spectrum $\Lambda[MA]$ should be as clustered as possible (see e.g. Nocedal and Wright, 2000). The latter fact may again be helpful whenever Krylov subspace methods are adopted to solve (2.2).

Since we want to deal with the large scale case, without any assumption on the sparsity pattern of A, the main approaches in the literature for building and applying a preconditioner to (2.1) often gain information on the system matrix by computing the matrix-vector product $A \times v$, with $v \in \mathbb{R}^n$, or rely on numerical differentiation. In particular, among the approaches proposed we have the following:

- Approximate Inverse Preconditioners based on using the BFGS update method (see also Nash, 1985; Benzi et *al.*, 2000, and therein references). Here, the main idea adopted is that a BFGS update may be suitably used to compute the approximate inverse $A^\sharp$ of matrix A. Then, the matrix $A^\sharp$ is applied as a preconditioner.
- Preconditioners based on the L-BFGS method (see also Morales and Nocedal, 2000; Morales and Nocedal, 2001, and the class of preconditioners Limited Memory Preconditioners (LMPs) by Giraud and Gratton, 2006, which pursue an idea similar to Approximate Inverse Preconditioners in the previous item.
- Approximate Inverse Preconditioners based on the use of Krylov subspace methods (see also Simoncini and Szyld, 2007; Fasano and Roma, 2009), where a Krylov subspace method is used to determine the solution of (2.1) and to provide information in order to build a preconditioner for (2.1).

- Band Preconditioners based on matrix scaling/balancing (see also Roma, 2005).
- Preconditioners based on numerical differentiation (see e.g. Higham, 2002).
- Band Preconditioners based on the BFGS method (see e.g. Luksan et *al.*, 2010, where a BFGS update is partially modified, so that suitable band preconditioners are defined for linear systems. This approach was mainly tested within truncated-Newton frameworks.

For each of the preconditioning strategies mentioned above we have both a theoretical analysis and a numerical experience, for validation. In this paper we want to follow and generalize the approach proposed in Fasano and Roma, 2009, where at any step, an iterative Krylov subspace method is used to compute the approximate inverse $A^\sharp$, over an increasing dimensional subspace. Note that as detailed in Section 3, our approach also encompasses diagonal and block-diagonal preconditioners.

3 Our class of preconditioners

In this section we first introduce some preliminaries, then we propose our class of preconditioners. Consider the *indefinite* linear system

$$Ax = b, \tag{3.1}$$

where $A \in \mathbb{R}^{n\times n}$ is *symmetric*, n is *large* and $b \in \mathbb{R}^n$. Suppose any Krylov subspace method is used for the solution of (3.1), e.g. the Lanczos process or the CG method (Golub and Van Loan, 1996), but MINRES (Saad, 2000) or Planar-CG methods (Hestenes, 1980; Fasano, 2005) may be also alternative choices. They are equivalent as long as $A \succ 0$, whereas the CG, though cheaper, in principle may not cope with the indefinite case. In the next Assumption 3.1 we suppose that a finite number of steps, say $h \ll n$, of the Krylov subspace method adopted are performed.

Assumption 3.1 *Let us consider any Krylov subspace method to solve the symmetric linear system (3.1). Suppose at 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, such that*

$$AR_h = R_hT_h + \rho_{h+1}u_{h+1}e_h^T, \qquad \rho_{h+1} \in \mathbb{R}, \tag{3.2}$$

$$T_h = \begin{cases} V_hB_hV_h^T, & \text{if } T_h \text{ is indefinite} \\ L_hD_hL_h^T, & \text{if } T_h \text{ is positive definite} \end{cases} \tag{3.3}$$

where

$R_h = (u_1 \cdots u_h)$, $u_i^Tu_j = 0$, $\|u_i\| = 1$, $1 \leq i \neq j \leq h$,

$u_{h+1}^Tu_i = 0$, $\|u_{h+1}\| = 1$, $1 \leq i \leq h$,

T_h is irreducible and nonsingular, with eigenvalues $\mu_1, \ldots, \mu_h$ not all coincident,

$B_h = diag_{1\leq i\leq h}\{\mu_i\}$, $V_h = (v_1 \cdots v_h) \in \mathbb{R}^{h\times h}$ *orthogonal, (μ_i, v_i) is an eigenpair of T_h,*

$D_h \succ 0$ is diagonal, L_h is unit lower bidiagonal.

Remark 3.1 Note that most of the common Krylov subspace methods for the solution of symmetric linear systems at iteration h may easily satisfy Assumption 3.1 (Saad, 2000; Stoer, 1983). In particular, also observe that from (3.2) we have $T_h = R_h^T A R_h$, so that whenever $A \succ 0$ then $T_h \succ 0$. Since the Jordan form of T_h in (3.3) is required only when T_h is indefinite, it is important to check whenever $T_h \succ 0$, without computing the eigenpairs of T_h if unnecessary. For this purpose, note that the Krylov subspace method adopted always provides relation $T_h = L_h D_h L_h^T$, with L_h nonsingular and D_h block diagonal (blocks can be 1×1 or 2×2 at most), even when T_h is indefinite (Saad, 2000; Stoer, 1983; Fasano and Roma, 2007). Thus, checking the entries of D_h will suggest if the Jordan form $T_h = V_h B_h V_h^T$ is really needed for T_h, i.e. if T_h is indefinite.

Furthermore, the matrix T_h captures much information on the eigenvalues of A, corresponding to eigenvectors in $span\{u_1, \ldots, u_h\}$. Indeed, let $v \in \mathbb{R}^n$, then $R_h v$ belongs to the Krylov subspace $span\{u_1, \ldots, u_h\}$. Now, considering that $R_h^T A R_h = T_h$ from (3.2), and recalling that $R_h^T R_h = I_h$, relation

$$\bar{m}\|v\|^2 \leq v^T T_h v \leq \bar{M}\|v\|^2$$

implies

$$\bar{m}\|R_h v\|^2 = \bar{m}\|v\|^2 \leq (R_h v)^T A (R_h v) \leq \bar{M}\|v\|^2 = \bar{M}\|R_h v\|^2.$$

Thus, $\bar{m}$ and $\bar{M}$ are respectively a lower bound and an upper bound of eigenvalues of A, corresponding to eigenvectors in $span\{u_1, \ldots, u_h\}$.
Observe also that from Assumption 3.1 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 performed by matrix A (thus, in this paper we consider a more general case with respect to Baglama *et al.*, 1998).

Remark 3.2 The Krylov subspace method adopted may in general perform $m \geq h$ iterations, generating the orthonormal vectors $u_1, \ldots, u_m$. Then, we can set $R_h = (u_{\ell_1}, \ldots, u_{\ell_h})$, with $\{\ell_1, \ldots, \ell_h\} \subseteq \{1, \ldots, m\}$, and change relations (3.2)-(3.3) accordingly; i.e. Assumption 3.1 may hold selecting any h out of the m vectors (among $u_1, \ldots, u_m$) computed by the Krylov subspace method.

Remark 3.3 For relatively small values of the parameter h in Assumption 3.1 (say $h \leq 20$, as often suffices in the applications), the computation of the eigenpairs (μ_i, v_i), $i = 1, \ldots, h$, of T_h when T_h is indefinite may be extremely fast, with standard codes. E.g. if the CG is the Krylov subspace method used in Assumption 3.1 to solve (3.1), then the `Matlab` (MathWorks, 2011) general function `eigs()` requires as low as $\approx 10^{-4}$ seconds to fully compute all the eigenpairs of T_h, for $h = 20$, on a commercial laptop. In the case where the CG is the Krylov-subspace method of choice, the matrix T_h is tridiagonal. Nonetheless, in the separate paper Fasano and Roma, 2009, we consider a special case where the request (3.3) on T_h may be considerably weakened under mild assumptions. Moreover, in the paper Fasano and Roma, 2011b, we also prove that for a special choice of the parameter 'a' used in our class of preconditioners (see below), strong theoretical properties may be stated.

On the basis of the latter assumption, we can now define our preconditioners and show their properties. To this aim, considering for the matrix T_h the expression (3.3), we define (see also Gill et *al.*,1992)

$$|T_h| \stackrel{\text{def}}{=} \begin{cases} V_h |B_h| V_h^T, \quad |B_h| = diag_{1 \leq i \leq h}\{|\mu_i|\}, & \text{if } T_h \text{ is indefinite,} \\ \\ T_h, & \text{if } T_h \text{ is positive definite.} \end{cases}$$

As a consequence, when T_h is indefinite we have $T_h|T_h|^{-1} = |T_h|^{-1}T_h = V_h\hat{I}_hV_h^T$, where the h nonzero diagonal entries of the matrix $\hat{I}_h$ are in the set $\{-1,+1\}$. Furthermore, it is easily seen that $|T_h|$ is positive definite, for any h, and the matrix $|T_h|^{-1}T_h^2|T_h|^{-1} = I_h$ is the identity matrix.

Now let us introduce the following $n \times n$ matrix, which *depends on the real parameter* 'a':

$$\begin{aligned} M_h &\stackrel{\text{def}}{=} (I - R_hR_h^T) + R_h|T_h|R_h^T + a\left(u_{h+1}u_h^T + u_hu_{h+1}^T\right), \qquad h \le n-1, \\ &= [R_h \mid u_{h+1} \mid R_{n,h+1}] \left[\begin{array}{c|c} \left(\begin{array}{c|c} |T_h| & ae_h \\ \hline ae_h^T & 1 \end{array}\right) & 0 \\ \hline 0 & I_{n-(h+1)} \end{array}\right] \left[\begin{array}{c} R_h^T \\ \hline u_{h+1}^T \\ \hline R_{n,h+1}^T \end{array}\right] \end{aligned} \tag{3.4}$$

$$M_n \stackrel{\text{def}}{=} (I - R_nR_n^T) + R_n|T_n|R_n^T = R_n|T_n|R_n^T, \tag{3.5}$$

where R_h and T_h satisfy relations (3.2)-(3.3), $a \in \mathbb{R}$, the matrix $R_{n,h+1} \in \mathbb{R}^{n\times[n-(h+1)]}$ is such that $R_{n,h+1}^TR_{n,h+1} = I_{n-(h+1)}$ and $[R_h \mid u_{h+1} \mid R_{n,h+1}]$ is orthogonal. Observe that of course the matrix $R_{n,h+1}$ in (3.4) always exists, with

$$R_{n,h+1}R_{n,h+1}^T = I_n - (R_h|u_{h+1})(R_h|u_{h+1})^T.$$

Using the parameter dependent matrix M_h in (3.4)-(3.5) we are now ready to introduce our class of preconditioners

$$\begin{aligned} M_h^\sharp(a,\delta,D) &= D\left[I_n - (R_h \mid u_{h+1})(R_h \mid u_{h+1})^T\right]D^T \\ &\quad + (R_h \mid Du_{h+1}) \left(\begin{array}{c|c} \delta^2|T_h| & ae_h \\ \hline ae_h^T & 1 \end{array}\right)^{-1} (R_h \mid Du_{h+1})^T \end{aligned} \tag{3.6}$$

$$M_n^\sharp(a,\delta,D) = R_n|T_n|^{-1}R_n^T. \tag{3.7}$$

Theorem 3.1 *Consider any Krylov subspace method to solve the symmetric linear system (3.1). Suppose that Assumption 3.1 holds and the Krylov method performs $h \le n$ iterations. Let $a \in \mathbb{R}$, $\delta \neq 0$, and let the matrix $D \in \mathbb{R}^{n\times n}$ be such that $[R_h \mid Du_{h+1} \mid DR_{n,h+1}]$ is nonsingular, where $R_{n,h+1}R_{n,h+1}^T = I_n - (R_h \mid u_{h+1})(R_h \mid u_{h+1})^T$. Then, we have the following properties:*

a) the matrix $M_h^\sharp(a,\delta,D)$ is symmetric. Furthermore

- *when $h \le n-1$, for any $a \in \mathbb{R} - \{\pm\delta(e_h^T|T_h|^{-1}e_h)^{-1/2}\}$, $M_h^\sharp(a,\delta,D)$ is nonsingular;*
- *when $h = n$ the matrix $M_h^\sharp(a,\delta,D)$ is nonsingular;*

b) the matrix $M_h^\sharp(a,\delta,D)$ coincides with M_h^{-1} as long as either $D = I_n$ and $\delta = 1$, or $h = n$;

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

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

d) when $h \le n-1$:

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

e) *when $h=n$, then $M_n^\sharp(a,\delta,D)=M_n^{-1}$, $\Lambda[M_n]=\Lambda[|T_n|]$ and $\Lambda[M_n^{-1}A]=\Lambda[AM_n^{-1}]\subseteq\{-1,+1\}$, i.e. the n eigenvalues of the preconditioned matrix $M_h^\sharp(a,\delta,D)A$ are either $+1$ or -1.*

Proof: See Fasano and Roma, 2011a, for the proof. □

The Corollary which follows considers the important particular case obtained by setting $a=0$, $\delta=1$ and $D=I_n$, in the definition of the preconditioner $M_h^\sharp(a,\delta,D)$.

Corollary 3.2 *Consider any Krylov subspace method to solve the symmetric linear system (3.1). Suppose that Assumption 3.1 holds and the Krylov subspace method performs $h\leq n$ iterations. Then, the preconditioner*

$$\begin{aligned} M_h^\sharp(0,1,I_n) &= \left[I_n-(R_h \mid u_{h+1})\,(R_h \mid u_{h+1})^T\right] \\ &\quad + (R_h \mid u_{h+1})\left(\begin{array}{c|c} |T_h| & 0 \\ \hline 0 & 1 \end{array}\right)^{-1}(R_h \mid u_{h+1})^T \qquad (3.8) \\ M_n^\sharp(0,1,I_n) &= R_n|T_n|^{-1}R_n^T, \qquad (3.9) \end{aligned}$$

is such that

a) *the matrix $M_h^\sharp(0,1,I_n)$ is symmetric and nonsingular for any $h\leq n$;*

b) *the matrix $M_h^\sharp(0,1,I_n)$ coincides with M_h^{-1}, for any $h\leq n$;*

c) *the matrix $M_h^\sharp(0,1,I_n)$ is positive definite. Moreover, its spectrum $\Lambda[M_h^\sharp(0,1,I_n)]$ is given by*

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

d) *when $h\leq n-1$, then the matrix $M_h^\sharp(0,1,I_n)A$ has at least $(h-2)$ singular values equal to $+1$;*

e) *when $h=n$, then $\Lambda[M_n]=\Lambda[|T_n|]$ and $\Lambda[M_n^\sharp(0,1,I_n)A]=\Lambda[M_n^{-1}A]=\Lambda[AM_n^{-1}]\subseteq\{-1,+1\}$, i.e. the n eigenvalues of $M_h^\sharp(0,1,I_n)A$ are either $+1$ or -1.*

Proof: The result is directly obtained from (3.4)-(3.5) and Theorem 3.1, with $a=0$, $\delta=1$ and $D=I_n$. □

Remark 3.4 As stated in the comments to relation (3.4), the matrix $R_{n,h+1}$ in the statement of Theorem 3.1 always exists, such that $[R_h \mid u_{h+1} \mid R_{n,h+1}]$ is orthogonal. However, $R_{n,h+1}$ is neither built nor used in (3.6)-(3.7), and it is introduced only for theoretical purposes. Furthermore, it is easy to see that since $[R_h \mid u_{h+1} \mid R_{n,h+1}]$ is orthogonal, any nonsingular diagonal matrix D may be used in order to satisfy the hypotheses of Theorem 3.1.

Remark 3.5 Observe that the case $h \approx n$ in Theorem 3.1 is of scarce interest for large scale problems. Indeed, in the literature of preconditioners the values of 'h' typically do not exceed $10 - 20$ (Morales and Nocedal, 2000; Gratton et *al.*, 2009). Moreover, for small values of h in (3.6) the computation of the inverse matrix

$$\left(\begin{array}{c|c} \delta^2 |T_h| & a e_h \\ \hline a e_h^T & 1 \end{array} \right)^{-1} \tag{3.10}$$

in order to provide $M_h^\sharp(a, \delta, I_n)$ or $M_h^\sharp(a, \delta, D)$, may be cheaply performed when T_h is either indefinite or positive definite. Indeed, after a brief computation we have

$$\left[\begin{array}{c|c} \delta^2 |T_h| & a e_h \\ \hline a e_h^T & 1 \end{array} \right]^{-1} = \left(\begin{array}{c|c} \frac{1}{\delta^2} |T_h|^{-1} - \frac{a}{\delta^4} \omega |T_h|^{-1} e_h e_h^T |T_h|^{-1} & \frac{\omega}{\delta^2} |T_h|^{-1} e_h \\ \hline \frac{\omega}{\delta^2} e_h^T |T_h|^{-1} & -\frac{\omega}{a} \end{array} \right), \tag{3.11}$$

with

$$\omega = -\frac{a}{1 - \frac{a^2}{\delta^2} e_h^T |T_h|^{-1} e_h}. \tag{3.12}$$

Thus, when T_h is indefinite, Remark 3.3 and relation (3.11) will provide the result. On the other hand, in case $T_h \succ 0$, it suffices to use (3.11). Finally, the proper setting of the parameter 'a' allows to easily control the condition number of matrix (3.10).

4 Preliminary numerical results

In order to preliminarily test our proposal on a general framework, without any assumption on the sparsity pattern of the matrix A, we used our parameter dependent class of preconditioners $M_h^\sharp(a, \delta, D)$, setting $\delta = 1$ and $D = I_n$.

We anticipate that in our numerical experience we obtain very interesting results as concerns the correspondence between theoretical and numerical results. Indeed, all the results stated in Theorem 3.1 for the *singular values* of the (possibly) unsymmetric matrix $M_h^\sharp(a, \delta, D)A$, seem to hold in practice also for the *eigenvalues* of $M_h^\sharp(a, \delta, D)A$ (it is worth to recall that since $M_h^\sharp(a, \delta, D) \succ 0$ then $\Lambda[M_h^\sharp(a, \delta, D)A] \equiv \Lambda[M_h^\sharp(a, \delta, D)^{1/2} A M_h^\sharp(a, \delta, D)^{1/2}]$), so that $M_h^\sharp(a, \delta, D)A$ has only real eigenvalues. As regards the numerical investigation, we used 3 different sets of test problems.

First, we considered a set of symmetric linear systems as in (3.1), where the number of unknowns n is set as $n = 1000$, and the matrix A has also a moderate condition number. We simply wanted to experience how our class of preconditioners affects the condition number of A. In particular (see also Geman, 1980), a possible choice for the latter class of matrices is given by

$$A = \{a_{i,j}\}, \qquad a_{ij} \in U[-10, 10], \quad i, j = 1, \ldots, n, \tag{4.1}$$

where $a_{i,j} = a_{j,i}$ are random entries in the uniform distribution $U[-10, 10]$, between -10 and $+10$. Then, also the vector b in (3.1) is computed randomly with entries in the set $U[-10, 10]$. We computed the preconditioners (3.6)-(3.7) by using the *Conjugate Gradient* (CG) method (Saad, 2000), which is one of the most popular Krylov subspace iterative methods to solve (3.1) (Golub and Van Loan, 1996). We remark that the CG is often used also in case the matrix A is indefinite, though it can prematurely stop. As an alternative choice, in order to satisfy Assumption 3.1 with A indefinite, we can use the Lanczos process (Lanczos, 1950), MINRES

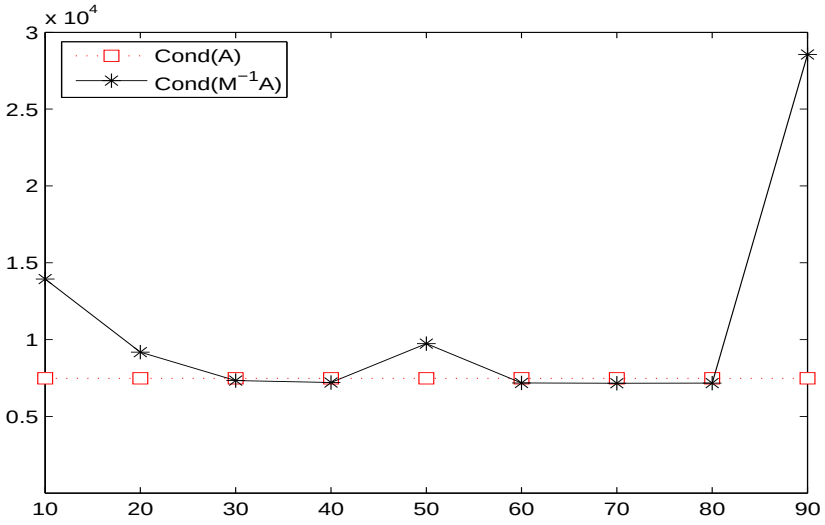

Figure 4.1: The condition number of matrix A (i.e. $Cond(A)$) along with the condition number of matrix $M_h^\sharp(0,1,I_n)A$ (i.e. $Cond(M^{-1}A)$), when $h \in \{10,20,30,40,50,60,70,80,90\}$, and A is randomly chosen with entries in the uniform distribution $U[-10,10]$.

methods (Paige and Saunders, 1975) or Planar-CG methods (Fasano, 2005). In (3.6) we set the parameter h in the range

$$h \in \{\ 20\ ,\ 30\ ,\ 40\ ,\ 50\ ,\ 60\ ,\ 70\ ,\ 80\ ,\ 90\ \},$$

and we preliminarily chose $a = 0$ (though other choices of the parameter 'a' yield similar results), which satisfied items a) and c) of Theorem 3.1. We have generated several systems like (4.1), obtaining very similar results. In particular, given one instance of A as in (4.1), we plotted in Figure 4.1 the condition number $\kappa(A)$ of A ($Cond(A)$), along with the condition number $\kappa(M_h^\sharp(0,1,I_n)A)$ of $M_h^\sharp(0,1,I_n)A$ ($Cond(M^{-1}A)$): in both cases the condition number κ is calculated by preliminarily computing the eigenvalues $\lambda_1,\ldots,\lambda_n$ (using `Matlab` MathWorks, 2011, routine `eigs()`) of A and $M_h^\sharp(0,1,I_n)A$ respectively, then obtaining the ratio

$$\kappa = \frac{\max_i\ |\lambda_i|}{\min_i\ |\lambda_i|}.$$

Evidently, numerical results confirm that the *order* of the condition number of A is pretty similar to that of the condition number of $M_h^\sharp(0,1,I_n)A$. This indicates that if the preconditioners (3.6) are used as a tool to solve (3.1), then most preconditioned iterative methods which are sensitive to the condition number (e.g. the Krylov subspace methods), on average are not expected to perform worse with respect to the unpreconditioned case. However, it is important to remark that the spectrum $\Lambda[M_h^\sharp(0,1,I_n)A]$ tends to be shifted with respect to $\Lambda[A]$, inasmuch as the eigenvalues in $\Lambda[A]$ whose absolute value is larger than $+1$ tend to be scaled in $\Lambda[M_h^\sharp(0,1,I_n)A]$ (see Figure 4.2). The latter property is an appealing result as described in Section 1, since the eigenvalues of $M_h^\sharp(0,1,I_n)A$ will be 'more clustered'. The latter phenomenon was better investigated by introducing other sets of test problems, described hereafter.

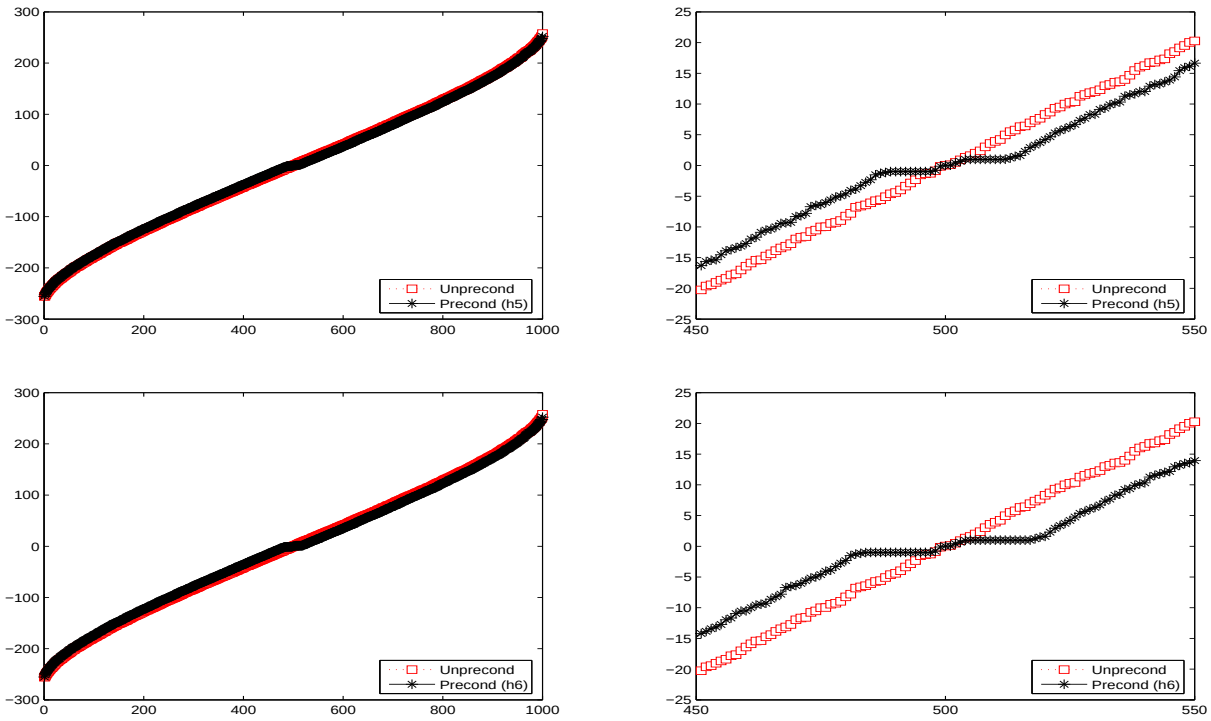

Figure 4.2: Comparison between the *full/detailed* spectra (left/right figures) $\Lambda[A]$ (*Unprecond*) and $\Lambda[M_h^\sharp(0,1,I_n)A]$ (*Precond*), with A randomly chosen (eigenvalues are sorted for simplicity); without loss of generality we show the results for the values $h = h5 = 20$ and $h = h6 = 30$. The intermediate eigenvalues in the spectrum $\Lambda[M_h^\sharp(0,1,I_n)A]$, whose absolute value is larger than 1, are in general smaller than the corresponding eigenvalues in $\Lambda[A]$. The eigenvalues in $\Lambda[M_h^\sharp(0,1,I_n)A]$ are more clustered near $+1$ or -1 than those in $\Lambda[A]$.

In a second experiment we generated the set of matrices A such that

$$A = H\mathcal{D}H, \tag{4.2}$$

where $H \in \mathbb{R}^{n\times n}$, $n = 500$, is an Householder transformation given by $H = I - 2vv^T$, 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 non-zero entries are also eigenvalues of A, while each column of H is also an eigenvector of A). 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 $\texttt{perc} = 0.3$). Finally, again we computed the preconditioners (3.6)-(3.7) by using the CG, 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, in order to build our preconditioners. In the latter case the CG method is indeed expected to perform exactly n iterations before stopping (see also Nocedal and Wright, 2000; Saad, 2000), so that the matrices (4.2) may be significant to test the effectiveness of our preconditioners, in case of *small values* of h (broadly speaking, h small implies that the preconditioner contains correspondingly a little information on the inverse matrix A^{-1}). We compared the spectra $\Lambda[A]$ and $\Lambda[M_h^\sharp(0, 1, I_n)A]$, in order to verify again how the preconditioners (3.6) are able to *cluster the eigenvalues of* A. Following the guidelines in Morales and Nocedal, 2000, in order to test our proposal also on a different range of values for the parameter h, we set

$$h \in \{\ 4\ ,\ 8\ ,\ 12\ ,\ 16\ ,\ 20\ \}.$$

The results are given in Figure 4.3 (*full comparisons*) which includes all the 500 eigenvalues, and Figure 4.4 (*details*) which includes only the eigenvalues from the 410-th to the 450-th. Observe that our preconditioners are able to shift the largest absolute eigenvalues of A towards -1 or $+1$, so that the clustering of the eigenvalues is enhanced when the parameter h increases. For each value of h the matrix A is (randomly) recomputed from scratch, according to relation (4.2). This explains why in the five plots of Figures 4.3-4.4 the spectrum of A changes. Again, a behavior very similar to Figures 4.3-4.4 is obtained also using different values for the parameter 'a'.

To complete our preliminary experience we tested our class of preconditioners in optimization frameworks. In particular, we considered a standard linesearch-based truncated Newton method in Table 5, where for any $k \geq 0$ the solution of the symmetric linear system (Newton's equation) $\nabla^2 f(x_k)d = -\nabla f(x_k)$ is required. We considered several unconstrained optimization problems from `CUTEr` (Gould et *al.*, 2003) collection, and for each problem we applied the truncated Newton method in Table 5. At the outer iteration k we computed the preconditioner $M_h^\sharp(0, 1, I_n)$, with $h \in \{4, 8, 12, 16, 20\}$, by using the CG to solve the equation $\nabla^2 f(x_k)d = -\nabla f(x_k)$. Then, we adopted $M_h^\sharp(0, 1, I_n)$ as a preconditioner for the solution of Newton's equation at the subsequent iteration

$$\nabla^2 f(x_{k+1})d = -\nabla f(x_{k+1}).$$

The iteration index 'k' was the first index such that both relations

$$\left\| \frac{x_{k+1} - x_k}{\alpha_k} \right\| \leq 10^{-3}\|x_k\| \qquad \text{and} \qquad \alpha_k \geq 0.95 \tag{4.3}$$

hold (the first relation implies that $x_{k+1} \approx x_k$, while the second holds when the search direction $(x_{k+1} - x_k)/\alpha_k$ approaches Newton's step). Thus, the index k was chosen in order to have $\|x_{k+1} - x_k\|$ *small*, i.e. the entries of the Hessian matrices $\nabla^2 f(x_k)$ and $\nabla^2 f(x_{k+1})$ are not expected to differ significantly. For simplicity we just report the results on two test problems,

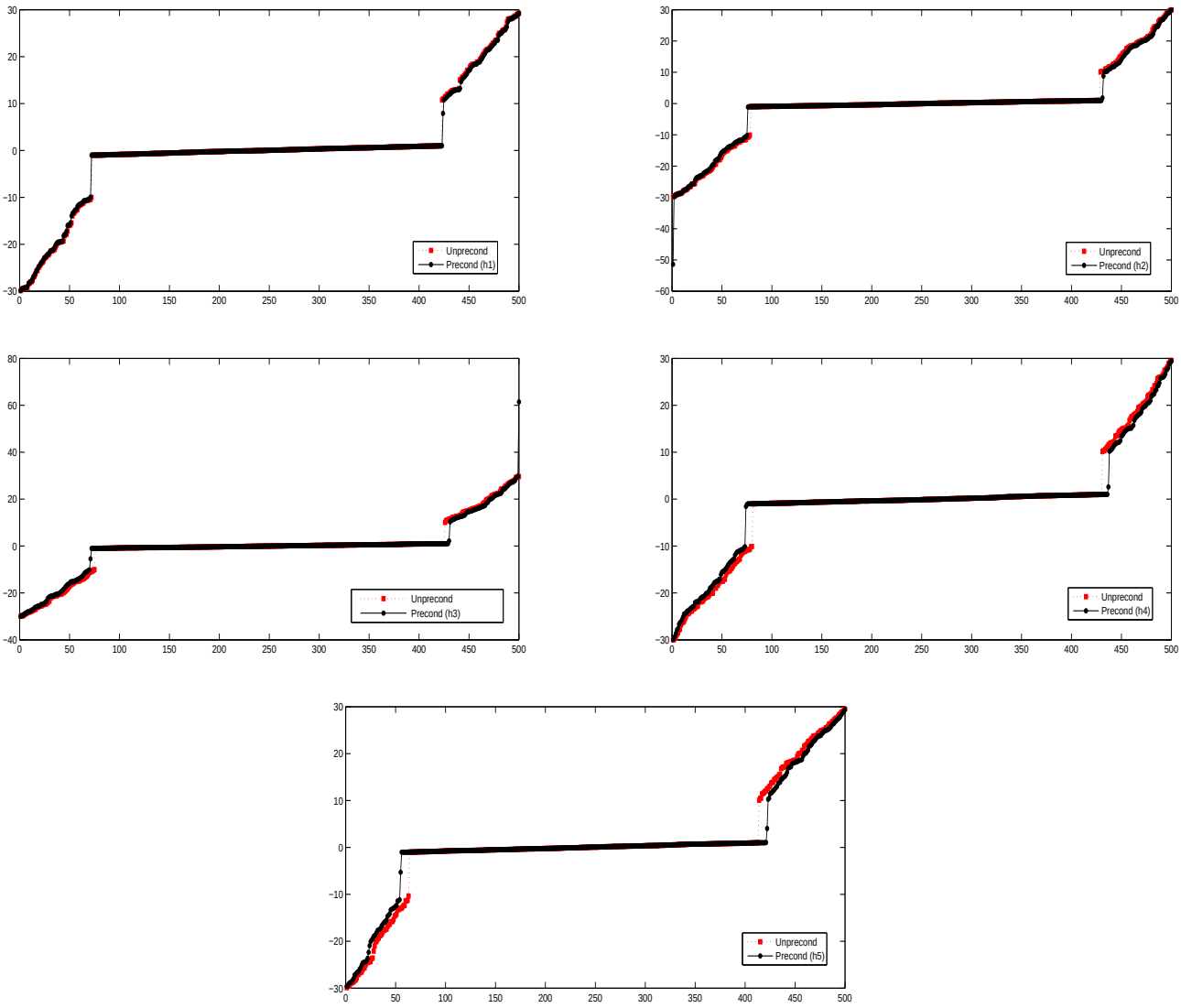

Figure 4.3: Comparison between the *full spectra* $\Lambda[A]$ (*Unprecond*) and $\Lambda[M_h^\sharp(0,1,I_n)A]$ (*Precond*), with A nonsingular and given by (4.2) (eigenvalues are sorted for simplicity); we used different values of h ($h_1 = 4$, $h_2 = 8$, $h_3 = 12$, $h_4 = 16$, $h_5 = 20$), setting $n = 500$. The large eigenvalues in the spectrum $\Lambda[M_h^\sharp(0,1,I_n)A]$ are in general smaller (in modulus) than the corresponding large eigenvalues in $\Lambda[A]$. A 'flatter' piecewise-line of the eigenvalues in $\Lambda[M_h^\sharp(0,1,I_n)A]$ indicates that the eigenvalues tend to cluster around -1 and $+1$, according with the theory.

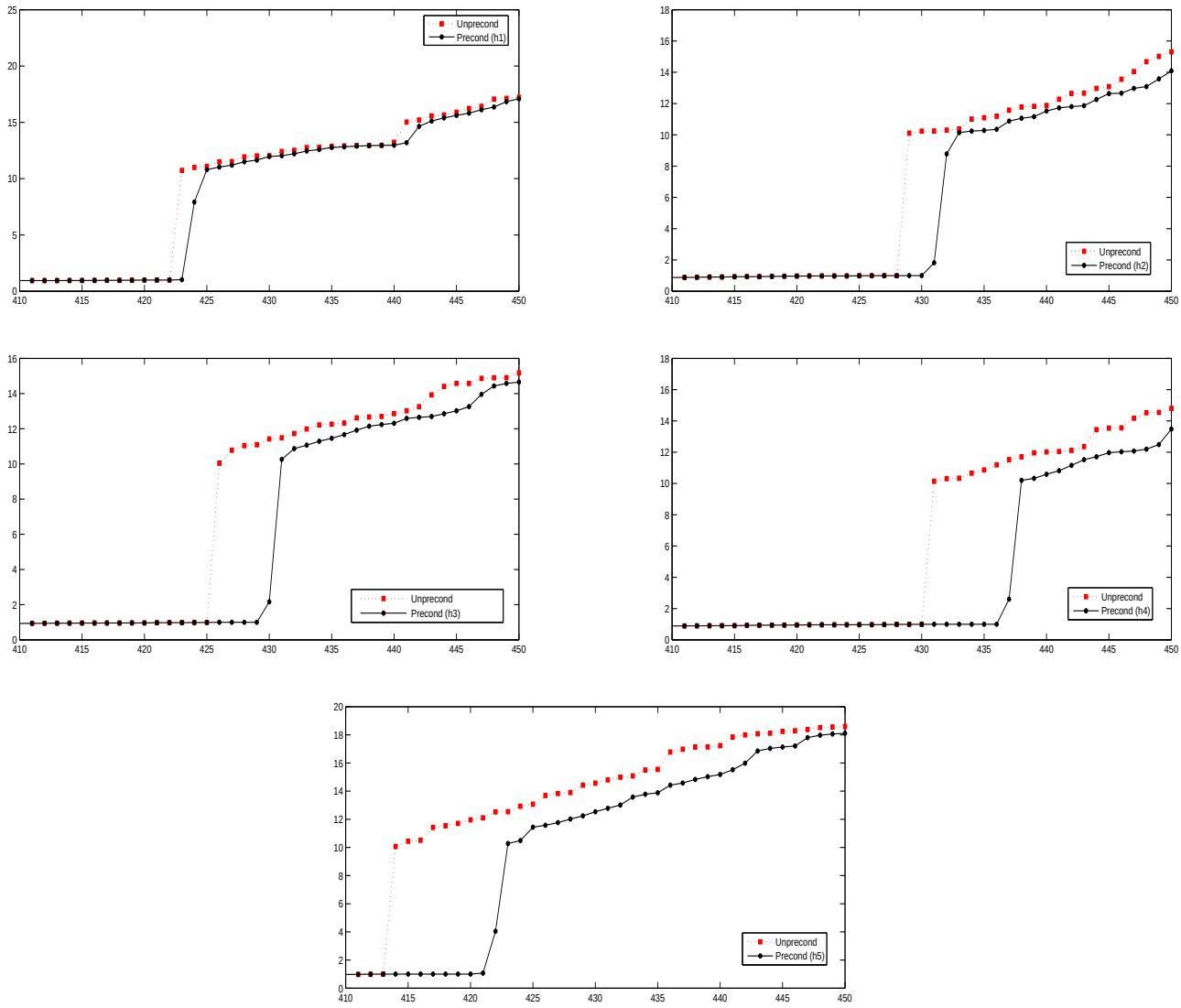

Figure 4.4: Comparison between a *detail of the spectra* $\Lambda[A]$ (*Unprecond*) and $\Lambda[M_h^\sharp(0,1,I_n)A]$ (*Precond*), with A nonsingular and given by (4.2) (eigenvalues are sorted for simplicity); we used different values of h ($h_1 = 4$, $h_2 = 8$, $h_3 = 12$, $h_4 = 16$, $h_5 = 20$), setting $n = 500$. The large eigenvalues in the spectrum $\Lambda[M_h^\sharp(0,1,I_n)A]$ are in general smaller (in modulus) than the corresponding large eigenvalues in $\Lambda[A]$. A 'flatter' piecewise-line of the eigenvalues in $\Lambda[M_h^\sharp(0,1,I_n)A]$ indicates that the eigenvalues tend to cluster around -1 and $+1$, according with the theory.

Set $x_0 \in \mathbb{R}^n$
Set $\eta_k \in [0,1)$ for any k, with $\{\eta_k\} \to 0$ as $k \to \infty$
OUTER ITERATIONS
for $k = 0, 1, \ldots$
  Compute $\nabla f(x_k)$; if $\|\nabla f(x_k)\|$ is small then STOP
    INNER ITERATIONS
    Compute d_k which approximately solves $\nabla^2 f(x_k)d = -\nabla f(x_k)$
    and satisfies the *truncation rule*
      $$\|\nabla^2 f(x_k)d_k + \nabla f(x_k)\| \leq \eta_k \|\nabla f(x_k)\|$$
  Compute the steplength α_k by an Armijo-type linesearch scheme
  Update $x_{k+1} = x_k + \alpha_k d_k$
endfor

Table 4.1: The linesearch-based truncated Newton method we adopted.

using $n = 1000$, in the set of all the optimization problems experienced. Very similar results were obtained for almost all the test problems. In Figures 4.5-4.6 we consider the problem `NONCVXUN`. For the sake of brevity we only show the numerical results using $h = 16$ in (3.6). Observe that since x_{k+1} is close to x_k (i.e. we are eventually converging to a local minimum) the Hessian matrix $\nabla^2 f(x_{k+1})$ is positive semidefinite. Furthermore, again the eigenvalues larger than +1 in $\Lambda[\nabla^2 f(x_{k+1})]$ are scaled in $\Lambda[M_h^\sharp(0,1,I_n)\nabla^2 f(x_{k+1})]$. Similarly we show in Figures 4.7-4.8 the results for the test function `NONDQUAR` in `CUTEr` collection. The test problems in this optimization framework, where the preconditioner $M_h^\sharp(0,1,I_n)$ is computed at the outer iteration k and used at the outer iteration $k+1$, confirm that the properties of Theorem 3.1 may hold also when $M_h^\sharp(0,1,I_n)$ is used on a sequence of linear systems $A_k x = b_k$, when A_k changes *slightly* with k.

5 Conclusions

We have given theoretical and numerical results for a class of preconditioners, which are parameter dependent. The preconditioners in our proposal can be built by using any Krylov method for the symmetric linear system (3.1), provided that it is able to satisfy the general conditions (3.2)-(3.3) in Assumption 3.1. The latter property may be appealing in several real problems, where a few iterations of the Krylov subspace method adopted may suffice to compute an effective preconditioner.
Our proposal seems tailored also for those cases where a sequence of linear systems of the form

$$A_k x = b_k, \qquad k = 1, 2, \ldots$$

requires a solution (e.g., see Morales and Nocedal, 2000, for details), where A_k slightly changes with the index k. In the latter case, the preconditioner $M_h^\sharp(a,\delta,D)$ in (3.6)-(3.7) can be computed applying the Krylov subspace method to the first linear system $A_1 x = b_1$. Then, $M_h^\sharp(a,\delta,D)$ can be used to efficiently solve $A_k x = b_k$, with $k = 2, 3, \ldots$. On this guideline, in a future work we are going to experience our proposal with other preconditioners described in Section 2. In particular, we think that a comparison with the proposals in Gratton et *al.*, 2009; Morales and Nocedal, 2000, could be noteworthy.

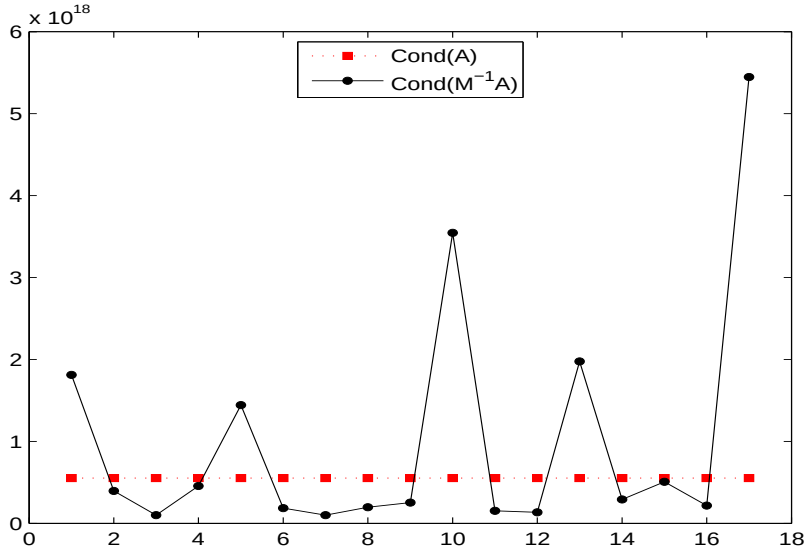

Figure 4.5: The condition number of matrix $\nabla^2 f(x_{k+1})$ $(Cond(A))$ along with the condition number of matrix $M_h^\sharp(0,1,I_n)\nabla^2 f(x_{k+1})$ $(Cond(M^{-1}A))$, for the optimization problem NONCVXUN, when $1 \leq h \leq 17$. The condition number of $\nabla^2 f(x_{k+1})$ is nearby the condition number of $M_h^\sharp(0,1,I_n)\nabla^2 f(x_{k+1})$, for any value of the parameter h. The value $k = 175$ was chosen as in (4.3) and it was $\|x_{176} - x_{175}\| \approx 0.083$.

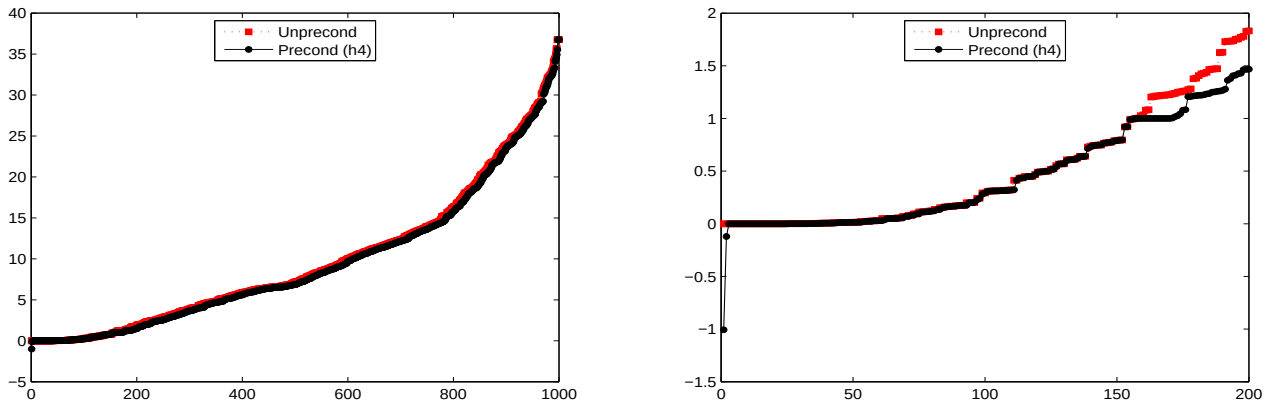

Figure 4.6: Comparison between the *full spectra/detailed spectra* (left figure/right figure) of $\nabla^2 f(x_{k+1})$ (*Unprecond*) and $M_h^\sharp(0,1,I_n)\nabla^2 f(x_{k+1})$ (*Precond*), for the optimization problem NONCVXUN, with $h = h_4 = 16$. The eigenvalues in $\Lambda[M_h^\sharp(0,1,I_n)\nabla^2 f(x_{k+1})]$ larger than $+1$ are evidently scaled, so that $\Lambda[M_h^\sharp(0,1,I_n)\nabla^2 f(x_{k+1})]$ is more clustered.

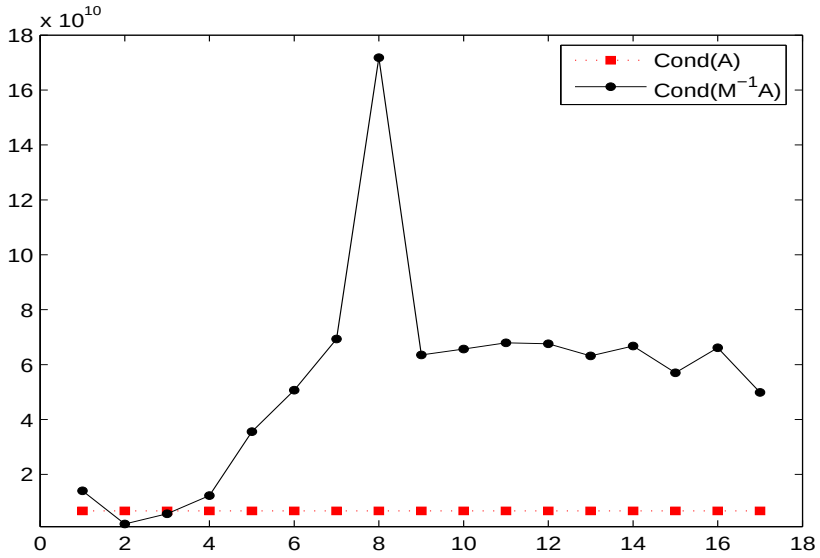

Figure 4.7: The condition number of matrix $\nabla^2 f(x_{k+1})$ $(Cond(A))$ along with the condition number of matrix $M_h^\sharp(0,1,I_n)\nabla^2 f(x_{k+1})$ $(Cond(M^{-1}A))$, for the optimization problem NONDQUAR, when $1 \leq h \leq 17$. The condition number of $\nabla^2 f(x_{k+1})$ is now slightly larger than the condition number of $M_h^\sharp(0,1,I_n)\nabla^2 f(x_{k+1})$ (though they are both $\approx 10^{10}$). The value $k = 40$ was chosen as in (4.3) and it was $\|x_{41} - x_{40}\| \approx 0.203$.

Finally, the class of preconditioners in this paper seems an interesting tool also for the solution of linear systems in financial frameworks. In particular, in future works we want to focus on symmetric linear systems arising when we impose KKT conditions in portfolio selection problems, with a large number of titles in the portfolio, along with linear equality constraints (see also Al-Jeiroudi et *al.*, 2008).

Acknowledgements The first author wishes to thank the Italian Ship Model Basin - INSEAN, CNR institute, for the support.

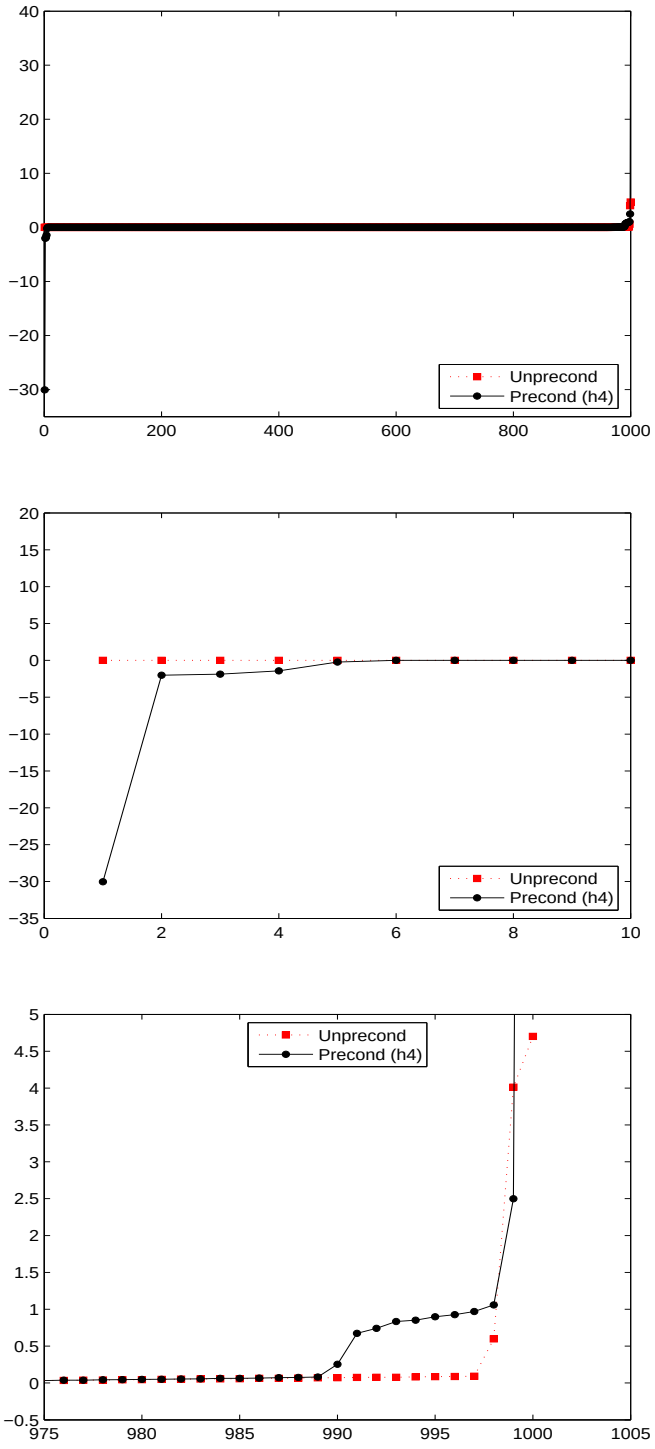

Figure 4.8: Comparison between the *full spectra/detailed spectra* (upper figure/lower figures) $\Lambda[\nabla^2 f(x_{k+1})]$ (*Unprecond*) and $\Lambda[M_h^\sharp(0,1,I_n)\nabla^2 f(x_{k+1})]$ (*Precond*), for the optimization problem NONDQUAR, with $h = h_4 = 16$. Some nearly-zero eigenvalues in the spectrum $\Lambda[\nabla^2 f(x_{k+1})]$ are shifted to non-zero values in $\Lambda[M_h^\sharp(0,1,I_n)\nabla^2 f(x_{k+1})]$. Since many eigenvalues in $\Lambda[\nabla^2 f(x_{k+1})]$ are zero or nearly-zero, the preconditioner $M_h^\sharp(0,1,I_n)$ may be of scarce effect, unless large values of the parameter h are considered.

References

MATLAB Release 2011a, The MathWorks Inc., The MathWorks Inc.

AL-JEIROUDI, G. GONDZIO, J. *and* HALL, J. 2008. Preconditioning indefinite systems in interior point methods for large scale linear optimisation, *Optimization Methods Software*, **23**, 345–363.

BAGLAMA, J. CALVETTI, D. GOLUB, G. *and* REICHEL, L. 1998. Adaptively preconditioned GMRES algorithms, SIAM Journal on Scientific Computing, **20**, 243–269.

BENZI, M. CULLUM, J. *and* TUMA, M. 2000. Robust approximate inverse preconditioner for the conjugate gradient method, *SIAM Journal on Scientific Computing*, **22**, 1318–1332.

DENNIS, J. *and* SCHNABEL, R. 1983. *Numerical Methods for Unconstrained Optimization and Nonlinear equations*, Prentice-Hall, Englewood Cliffs, 1983.

FASANO, G. 2005. Planar-conjugate gradient algorithm for large-scale unconstrained optimization, Part 1: Theory, *Journal of Optimization Theory and Applications*, **125**, 523–541.

FASANO, G. *and* ROMA, M. 2007. Iterative computation of negative curvature directions in large scale optimization, *Computational Optimization and Applications*, **38**, 81–104.

FASANO, G. *and* ROMA, M. 2009. Preconditioning Newton-Krylov methods in nonconvex large scale optimization, submitted to *Computational Optimization and Applications*.

FASANO, G. *and* ROMA, M. 2011a. A class of preconditioners for large indefinite linear systems, as by-product of Krylov-based methods: Part 1, *Technical Report n. 4*, Department of Management, University Ca'Foscari, Venice, Italy.

FASANO, G. *and* ROMA, M. 2011b. A class of preconditioners for large indefinite linear systems, as by-product of Krylov-based methods: Part 2, *Technical Report n. 5*, Department of Management, University Ca'Foscari, Venice, Italy.

GEMAN, S. 1980. A limit theorem for the norm of random matrices, *The Annals of Probability*, **8**, 252–261.

GILL, P.E. MURRAY, W. PONCELEÓN, D.B. *and* SAUNDERS, M.A. 1992. Preconditioners for indefinite systems arising in optimization, *SIAM Journal on Matrix Analysis and Applications*, **13**, 292–311.

GIRAUD, L. *and* GRATTON, S. 2006. On the sensitivity of some spectral preconditioners, *SIAM Journal on Matrix Analysis and Applications*, **27**, 1089–1105.

GOLUB, G. *and* VAN LOAN, C. 1996. *Matrix Computations*, The John Hopkins Press, Baltimore, Third edition.

GOULD, N.I.M. ORBAN, D. *AND* TOINT, P.L. 2003. CUTEr (and sifdec), a constrained and unconstrained testing environment, revised, *ACM Transaction on Mathematical Software*, **29**, 373–394.

GRATTON, S. SARTENAER, A. *AND* ILUNGA, J. 2009. On a class of limited memory preconditioners for large scale linear systems with multiple right-hand sides. Submitted to *SIAM Journal on Optimization*.

HESTENES, M. 1980. *Conjugate Direction Methods in Optimization*, Springer Verlag, New York.

HIGHAM, N. 2002. *Accuracy and stability of numerical algorithms*, SIAM, Philadelphia, PA, Second edition.

LANCZOS, C. 1950. An iteration method for the solution of the eigenvalue problem of linear differential and integral, *J. Res. Nat. Bur. Standards*, **45**, 255–282.

LUKSAN, L. MATONOHA, C. *AND* VLCEK, J. 2010. Band preconditioners for the matrix free truncated Newton method, *Technical Report n. 1079*, Institute of Computer Science, Academy of Sciences of the Czech Republic.

MORALES, J.L. *AND* NOCEDAL, J. 2000. Automatic preconditioning by limited memory quasi-Newton updating, *SIAM Journal on Optimization*, **10**, 1079–1096.

MORALES, J.L. *AND* NOCEDAL, J. 2001. Algorithm PREQN: Fortran 77 subroutine for preconditioning the conjugate gradient method, *ACM Transaction on Mathematical Software*, **27**, 83–91.

NASH, S. 1985. Preconditioning of truncated-Newton methods, *SIAM J. Scientific and Statistical Computing*, **6**, 599–616.

NASH, S. 2000. A survey of truncated-Newton methods, *Journal of Computational and Applied Mathematics*, **124**, 45–59.

NOCEDAL, J. *AND* WRIGHT, S. 2000. *Numerical Optimization* (Springer Series in Operations Research and Financial Engineering) - Second edition, Springer, New York.

PAIGE, C. *AND* SAUNDERS, M. 1975. Solution of sparse indefinite systems of linear equations, *SIAM Journal on Numerical Analysis*, **12**, 617–629.

ROMA, M. 2005. A dynamic scaling based preconditioning for truncated Newton methods in large scale unconstrained optimization, *Optimization Methods and Software*, **20**, 693–713.

SAAD, Y. 2003. *Iterative Methods for Sparse Linear Systems*, Second Edition, SIAM, Philadelphia.

SIMONCINI, V. *AND* SZYLD, D.B. 2007. Recent developments in Krylov Subspace Methods for linear systems, *Numer. Linear Algebra with Appl.*, **14**, 1–59.

STOER, J. 1983. *Solution of large linear systems of equations by conjugate gradient type methods, in Mathematical Programming. The State of the Art*, A. Bachem, M.Grotschel, and B. Korte, eds., Berlin Heidelberg, Springer-Verlag, 540–565.

Figures' captions

Figure 1: The condition number of matrix A (i.e. $Cond(A)$) along with the condition number of matrix $M_h^\sharp(0,1,I_n)A$ (i.e. $Cond(M^{-1}A)$), when $h \in \{10,20,30,40,50,60,70,80,90\}$, and A is randomly chosen with entries in the uniform distribution $U[-10,10]$.

Figure 2: Comparison between the *full/detailed* spectra (left/right figures) $\Lambda[A]$ (*Unprecond*) and $\Lambda[M_h^\sharp(0,1,I_n)A]$ (*Precond*), with A randomly chosen (eigenvalues are sorted for simplicity); without loss of generality we show the results for the values $h = h5 = 20$ and $h = h6 = 30$. The intermediate eigenvalues in the spectrum $\Lambda[M_h^\sharp(0,1,I_n)A]$, whose absolute value is larger than 1, are in general smaller than the corresponding eigenvalues in $\Lambda[A]$. The eigenvalues in $\Lambda[M_h^\sharp(0,1,I_n)A]$ are more clustered near $+1$ or -1 than those in $\Lambda[A]$.

Figure 3: Comparison between the *full spectra* $\Lambda[A]$ (*Unprecond*) and $\Lambda[M_h^\sharp(0,1,I_n)A]$ (*Precond*), with A nonsingular and given by (4.2) (eigenvalues are sorted for simplicity); we used different values of h ($h_1 = 4$, $h_2 = 8$, $h_3 = 12$, $h_4 = 16$, $h_5 = 20$), setting $n = 500$. The large eigenvalues in the spectrum $\Lambda[M_h^\sharp(0,1,I_n)A]$ are in general smaller (in modulus) than the corresponding large eigenvalues in $\Lambda[A]$. A 'flatter' piecewise-line of the eigenvalues in $\Lambda[M_h^\sharp(0,1,I_n)A]$ indicates that the eigenvalues tend to cluster around -1 and $+1$, according with the theory.

Figure 4: Comparison between a *detail of the spectra* $\Lambda[A]$ (*Unprecond*) and $\Lambda[M_h^\sharp(0,1,I_n)A]$ (*Precond*), with A nonsingular and given by (4.2) (eigenvalues are sorted for simplicity); we used different values of h ($h_1 = 4$, $h_2 = 8$, $h_3 = 12$, $h_4 = 16$, $h_5 = 20$), setting $n = 500$. The large eigenvalues in the spectrum $\Lambda[M_h^\sharp(0,1,I_n)A]$ are in general smaller (in modulus) than the corresponding large eigenvalues in $\Lambda[A]$. A 'flatter' piecewise-line of the eigenvalues in $\Lambda[M_h^\sharp(0,1,I_n)A]$ indicates that the eigenvalues tend to cluster around -1 and $+1$, according with the theory.

Figure 5: The condition number of matrix $\nabla^2 f(x_{k+1})$ ($Cond(A)$) along with the condition number of matrix $M_h^\sharp(0,1,I_n)\nabla^2 f(x_{k+1})$ ($Cond(M^{-1}A)$), for the optimization problem NONCVXUN, when $1 \le h \le 17$. The condition number of $\nabla^2 f(x_{k+1})$ is nearby the condition number of $M_h^\sharp(0,1,I_n)\nabla^2 f(x_{k+1})$, for any value of the parameter h. The value $k = 175$ was chosen as in (4.3) and it was $\|x_{176} - x_{175}\| \approx 0.083$.

Figure 6: Comparison between the *full spectra/detailed spectra* (left figure/right figure) of $\nabla^2 f(x_{k+1})$ (*Unprecond*) and $M_h^\sharp(0,1,I_n)\nabla^2 f(x_{k+1})$ (*Precond*), for the optimization problem NONCVXUN, with $h = h_4 = 16$. The eigenvalues in $\Lambda[M_h^\sharp(0,1,I_n)\nabla^2 f(x_{k+1})]$ larger than $+1$ are evidently scaled, so that $\Lambda[M_h^\sharp(0,1,I_n)\nabla^2 f(x_{k+1})]$ is more clustered.

Figure 7: The condition number of matrix $\nabla^2 f(x_{k+1})$ ($Cond(A)$) along with the condition number of matrix $M_h^\sharp(0,1,I_n)\nabla^2 f(x_{k+1})$ ($Cond(M^{-1}A)$), for the optimization problem NONDQUAR, when $1 \le h \le 17$. The condition number of $\nabla^2 f(x_{k+1})$ is now slightly larger than the condition number of $M_h^\sharp(0,1,I_n)\nabla^2 f(x_{k+1})$ (though they are both $\approx 10^{10}$). The value

$k = 40$ was chosen as in (4.3) and it was $\|x_{41} - x_{40}\| \approx 0.203$.

Figure 8: Comparison between the *full spectra/detailed spectra* (upper figure/lower figures) $\Lambda[\nabla^2 f(x_{k+1})]$ (*Unprecond*) and $\Lambda[M_h^\sharp(0,1,I_n)\nabla^2 f(x_{k+1})]$ (*Precond*), for the optimization problem NONDQUAR, with $h = h_4 = 16$. Some nearly-zero eigenvalues in the spectrum $\Lambda[\nabla^2 f(x_{k+1})]$ are shifted to non-zero values in $\Lambda[M_h^\sharp(0,1,I_n)\nabla^2 f(x_{k+1})]$. Since many eigenvalues in $\Lambda[\nabla^2 f(x_{k+1})]$ are zero or nearly-zero, the preconditioner $M_h^\sharp(0,1,I_n)$ may be of scarce effect, unless large values of the parameter h are considered.

List of Tables

Set $x_0 \in \mathbb{R}^n$
Set $\eta_k \in [0,1)$ for any k, with $\{\eta_k\} \to 0$ as $k \to \infty$
OUTER ITERATIONS
for $k = 0, 1, \ldots$
    Compute $\nabla f(x_k)$; if $\|\nabla f(x_k)\|$ is small then STOP
        INNER ITERATIONS
        Compute d_k which approximately solves $\nabla^2 f(x_k)d = -\nabla f(x_k)$
        and satisfies the *truncation rule*
            $$\|\nabla^2 f(x_k)d_k + \nabla f(x_k)\| \le \eta_k \|\nabla f(x_k)\|$$
    Compute the steplength α_k by an Armijo-type linesearch scheme
    Update $x_{k+1} = x_k + \alpha_k d_k$
endfor

Table 5.1: The linesearch-based truncated Newton method we adopted.