CONVERGENCE OF TRUNCATED HALF-QUADRATIC AND NEWTON ALGORITHMS, WITH APPLICATION TO IMAGE RESTORATION.

CHRISTIAN LABAT* AND JÉRÔME IDIER†

June 22, 2007

Abstract. We address the minimization of penalized least squares (PLS) criteria customarily used for edge-preserving restoration and reconstruction of signals and images. The minimization of PLS criteria can be addressed using a half-quadratic (HQ) scheme, according either to Geman & Reynolds (1992) or to Geman & Yang (1995) constructions. In the case of large-scale problems, the cost of the HQ approach is usually too high. In practice, it is rather proposed to implement an inexact HQ algorithm using a truncated conjugate gradient (TCG) method. This principle echoes that of truncated-Newton algorithms. Our contribution is to establish the convergence of the resulting truncated algorithms (HQ or Newton), under the same conditions required for the exact HQ scheme. Indeed, convergence is granted whatever the number of performed iterations of TCG. According to our experimental study on a deconvolution problem, the fastest versions correspond to severe truncation. This reinforces the interest for the truncated schemes, as fully valid algorithms in the field of image restoration.

Key words. Convergence, half-quadratic algorithms, truncated-Newton methods, conjugate gradient, inverse problems, optimization, preconditioning, signal and image restoration.

AMS subject classifications. 68W40, 68W25, 49N45, 90C06, 65Y20, 68U10

1. Introduction. This work addresses a wide class of problems where a sought vector $\hat{\boldsymbol{x}} \in \mathbb{R}^N$ (*e.g.,* an image or a signal) is estimated based on degraded data $\boldsymbol{y} \in \mathbb{R}^Q$. A usual approach is to minimize the following penalized least squares (PLS) criterion $\mathcal{J} : \mathbb{R}^N \mapsto \mathbb{R}$ which combines a quadratic data-fidelity term and a regularization term Φ, weighted by a parameter $\lambda > 0$:

$$\mathcal{J}(\boldsymbol{x}) = \left\|\mathbf{H}\boldsymbol{x} - \boldsymbol{y}\right\|^2 + \lambda\Phi(\boldsymbol{x}). \tag{1.1}$$

Here, we restrict ourselves to the case where $\mathbf{H}$ is a linear operator, described as a $Q \times N$ matrix, and we focus on regularization terms of the form

$$\Phi(\boldsymbol{x}) = \sum_{c=1}^{C} \phi([\mathbf{V}\boldsymbol{x}]_c), \tag{1.2}$$

where $\mathbf{V} \in \mathbb{R}^{C\times N}$ is typically the first or second-order difference matrix. The case where $\mathbf{V}$ is the identity matrix is also of interest.

Such a minimization problem is most commonly met in image restoration and reconstruction, which is our main concern [7, 8, 4, 15, 1]. It is also relevant in much wider contexts including inverse problems, robust statistical estimation, matching pursuit decomposition, and machine learning issues. Here we focus on coercive, edge-preserving potential functions $\phi : \mathbb{R} \mapsto \mathbb{R}$ in (1.2), because they give rise to image and signal estimates of high quality, involving edges and homogeneous regions.

The local minimization (*i.e.,* the search of a local minimum) of criterion $\mathcal{J}$ (1.1) can be addressed using a half-quadratic (HQ) approach that exploits the PLS structure

[*] Post Doctoral Associate, Image Analysis and Communications Lab, Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, USA (`Christian.Labat@jhu.edu`).

[†] CNRS Researcher, Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN, UMR CNRS 6597), Nantes, France (`Jerome.Idier@irccyn.ec-nantes.fr`).

exhibited by $\mathcal{J}$, either according to Geman and Reynolds (GR) or to Geman and Yang (GY) constructions [7, 8]. In both cases, the resulting algorithm is shown to fall within the class of *constant stepsize*, *gradient-related* methods [1]. Both GR and GY algorithms are of the following form

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \theta \boldsymbol{d}_k \tag{1.3}$$

$$\boldsymbol{d}_k = -\, \mathbf{A}_k^{-1} \nabla \mathcal{J}(\boldsymbol{x}_k), \tag{1.4}$$

where the normal matrix $\mathbf{A}_k$ is symmetric, positive definite (SPD). In the GR case, $\mathbf{A}_k = \mathbf{B}_{\mathrm{GR}}(\boldsymbol{x}_k)$ depends on the current iteration according to:

$$\mathbf{B}_{\mathrm{GR}}(\boldsymbol{x}) = 2\mathbf{H}^{\mathrm{t}}\mathbf{H} + \lambda \mathbf{V}^{\mathrm{t}}\mathbf{L}(\boldsymbol{x})\mathbf{V}, \tag{1.5}$$

$$\mathbf{L}(\boldsymbol{x}) = \mathrm{Diag}\{\phi'([\mathbf{V}\boldsymbol{x}]_c)/[\mathbf{V}\boldsymbol{x}]_c\}, \tag{1.6}$$

while it is a constant matrix $\mathbf{A}_k = \mathbf{B}^a_{\mathrm{GY}}$ in the GY case, defined by:

$$\mathbf{B}^a_{\mathrm{GY}} = 2\mathbf{H}^{\mathrm{t}}\mathbf{H} + \lambda \mathbf{V}^{\mathrm{t}}\mathbf{V}/a. \tag{1.7}$$

Some convergence results for both HQ algorithms can be found in [4, 15, 1].

In general, the nontrivial inversion of a linear system (1.4) is needed within the HQ approach to compute the search direction $\boldsymbol{d}_k$, at each iteration. In the case of large-scale problems, the resulting numerical cost is usually too high. In practice, it is rather proposed to compute an inexact search direction using a truncated preconditioned conjugate gradient (TPCG) method [4, 14, 15]. *Truncated* means that the preconditioned conjugate gradient (PCG) method is stopped before exact convergence, since the latter would require a prohibitive number of iterations, even as large as the size of the sought vector. In short, let us refer to the resulting family of truncated HQ algorithms as HQ+TPCG.

The principle of HQ+TPCG methods echoes that of truncated-Newton algorithms for solving large nonlinear optimization problems by approximately solving the Newton equation [5, 6, 12]

$$\boldsymbol{d}_k = -\, \nabla^2 \mathcal{J}(\boldsymbol{x}_k)^{-1} \nabla \mathcal{J}(\boldsymbol{x}_k), \tag{1.8}$$

assumed defined, using an inner iterative algorithm. Here the Hessian of $\mathcal{J}$ reads

$$\nabla^2 \mathcal{J}(\boldsymbol{x}) = 2\mathbf{H}^{\mathrm{t}}\mathbf{H} + \lambda \mathbf{V}^{\mathrm{t}} \mathrm{Diag}\{\phi''([\mathbf{V}\boldsymbol{x}]_c)\}\mathbf{V}. \tag{1.9}$$

Some variant of the linear conjugate gradient (CG) method is almost always used to perform the inner iterative algorithm for solving (1.8) [12, p. 46].

The HQ+TPCG family and the truncated-Newton method can be cast into a unique formulation:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k \tag{1.10}$$

$$\boldsymbol{d}_k = \boldsymbol{u}_{I_k}(\boldsymbol{x}_k), \tag{1.11}$$

where $\boldsymbol{u}_{I_k}(\boldsymbol{x}_k)$ is the vector obtained after $I_k \in \{1, \ldots, N\}$ iterations of the PCG algorithm described in Section 2 with $\boldsymbol{u}_0(\boldsymbol{x}_k) = \mathbf{0}$ as initial guess, applied to the following linear system

$$\mathbf{A}_k \boldsymbol{u} = -\nabla \mathcal{J}(\boldsymbol{x}_k), \tag{1.12}$$

where $\mathbf{A}_k$ is either the GR matrix (1.5), the GY matrix (1.7), or the Hessian (1.9). Note that $\boldsymbol{u}_{I_k}(\boldsymbol{x}_k)$ turns out to be a descent direction in $\boldsymbol{x}_k$ and the sequence $\{I_k\}$ is not necessarily constant.

The new estimate of the solution is obtained as $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$, where α_k is the stepsize. In inexact Newton methods, the stepsize is determined either by a constant stepsize (unitary) [5] [17, Algorithm 1] with local convergence or by a line search strategy (backtracking) [6] [3] [17, Algorithm 2] with global convergence.

In the present paper, we propose the following stepsize formula as a simple line search strategy:

$$\alpha_k = -\theta \frac{\boldsymbol{d}_k^{\mathrm{t}} \nabla \mathcal{J}(\boldsymbol{x}_k)}{\boldsymbol{d}_k^{\mathrm{t}} \mathbf{B}_k \boldsymbol{d}_k} \tag{1.13}$$

where $\mathbf{B}_k$ is either the GR matrix (1.5) or the GY matrix (1.7). In the exact HQ case (*i.e.,* when $\mathbf{A}_k = \mathbf{B}_k$ and $I_k = N$), it is obvious that $\alpha_k = \theta$. Actually, it will be shown later that $\alpha_k = \theta$ remains true for the HQ+TPCG family, for all $I_k \in \{1, \ldots, N\}$.

Indeed, the stepsize formula (1.13) identifies with a single iteration of a one-dimensional (1D) HQ algorithm applied to the minimization of $f(\alpha) = \mathcal{J}(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k)$. This is readily seen by applying the scheme (1.3)-(1.4) with a zero initialization to the minimization of f, which reads

$$\alpha_k = 0 - \theta \frac{f'(0)}{\boldsymbol{d}_k^{\mathrm{t}} \mathbf{B}_k \boldsymbol{d}_k} = -\theta \frac{\boldsymbol{d}_k^{\mathrm{t}} \nabla \mathcal{J}(\boldsymbol{x}_k)}{\boldsymbol{d}_k^{\mathrm{t}} \mathbf{B}_k \boldsymbol{d}_k}$$

where either $\mathbf{B}_k = \mathbf{B}_{\mathrm{GR}}(\boldsymbol{x}_k)$ or $\mathbf{B}_k = \mathbf{B}_{\mathrm{GY}}^a$. Note that the stepsize formula (1.13) is the same as in [11] and is also reminiscent of the one in [16], both within nonlinear CG methods. Up to our knowledge, the stepsize formula (1.13) is new for Newton type methods. In what follows, the truncated-Newton methods obtained by (1.10)-(1.13) when $\mathbf{A}_k = \nabla^2 \mathcal{J}(\boldsymbol{x})$ will be referred to as N+TPCG, and the whole family defined by (1.10)-(1.13) and gathering HQ+TPCG and N+TPCG methods will be designated as HQ/N+TPCG.

The contribution of this paper can be examined at different levels of generality. The first level of results deals with the convergence of truncated HQ schemes to minimize the PLS criterion (1.1). A higher level of generality is obtained by introducing two distinct matrix sequences $\{\mathbf{A}_k\}$ and $\{\mathbf{B}_k\}$ to govern the search direction and the stepsize scheme, respectively. Whereas the two sequences are structurally identical within the HQ approach, our paper shows that a broad family of converging, truncated methods can be obtained by relaxing this constraint. In particular, the case where $\{\mathbf{A}_k\}$ is the Hessian of the criterion is of specific interest.

Moreover, our convergence results are available for any level of truncation $I_k \in \{1, \ldots, N\}$, which means that even very rough inversion of (1.12) leads to convergence. In practice, this property has important consequences: according to our experience, the fastest versions of HQ/N+TPCG are obtained when (1.12) is only roughly inverted, *i.e.,* when the HQ scheme or Newton is severely truncated. Of course, a more severely truncated algorithm may require a larger number of global iterations to meet a given stopping criterion, but each of these global iterations takes less time. Using a severely truncated scheme, the computation time is actually reduced by a factor two to ten compared to the HQ algorithms with a lighter truncation, as implemented in [14].

Finally, let us underline that our core results address the minimization of criteria with a more general structure than the PLS expression (1.1). Basically, they deal with all differentiable criteria, lower bounded, and admitting an upper convex quadratic approximation in a neighborhood of the level set $\mathcal{L}(\boldsymbol{x}_0) = \{\boldsymbol{x} \in \mathbb{R}^N | \mathcal{J}(\boldsymbol{x}) \leq \mathcal{J}(\boldsymbol{x}_0)\}$.

The paper is structured as follows. Useful properties concerning the PCG directions (1.11) are gathered in Section 2. The convergence of the HQ/N+TPCG algorithms defined by (1.10)-(1.13) is studied in Section 3. Finally, Section 4 is devoted to numerical results.

2. Background material on linear preconditioned conjugate gradient.

The present section gathers technical results concerning the directions generated by (1.11), which will be useful afterwards to derive global convergence properties. Some of them could probably be derived as particular instances of more general properties of linear CG. For the sake of a self-contained paper, they are rather stated and proved as standalone results.

The method [10] and the preconditioned extension are among the most prominent iterative methods for solving linear systems. Let us consider the linear system

$$\mathbf{A}\boldsymbol{u} = \boldsymbol{b},$$

where $\boldsymbol{b}$, $\boldsymbol{u} \in \mathbb{R}^N$, and $\mathbf{A} \in \mathbb{R}^{N\times N}$ is a SPD matrix.

Let us consider a SPD preconditioning matrix $\mathbf{M} \in \mathbb{R}^{N\times N}$. Let $\boldsymbol{u}_0 \in \mathbb{R}^N$ be the initial guess, $\boldsymbol{r}_0 = \boldsymbol{b} - \mathbf{A}\boldsymbol{u}_0$ and $\boldsymbol{p}_0 = \mathbf{M}^{-1}\boldsymbol{r}_0$. The PCG algorithm can be defined as follows [9, Algorithm 10.3.1]:

$$\alpha_i = \frac{\|\boldsymbol{r}_i\|^2_{\mathbf{M}^{-1}}}{\|\boldsymbol{p}_i\|^2_{\mathbf{A}}} \qquad \text{(Optimal stepsize)}, \tag{2.1}$$

$$\boldsymbol{u}_{i+1} = \boldsymbol{u}_i + \alpha_i \boldsymbol{p}_i \qquad \text{(Update iterate)}, \tag{2.2}$$

$$\boldsymbol{r}_{i+1} = \boldsymbol{r}_i - \alpha_i \mathbf{A}\boldsymbol{p}_i \qquad \text{(Update residual)}, \tag{2.3}$$

$$\beta_i = \frac{\|\boldsymbol{r}_{i+1}\|^2_{\mathbf{M}^{-1}}}{\|\boldsymbol{r}_i\|^2_{\mathbf{M}^{-1}}} \qquad \text{(Conjugacy formula)}, \tag{2.4}$$

$$\boldsymbol{p}_{i+1} = \mathbf{M}^{-1}\boldsymbol{r}_{i+1} + \beta_i \boldsymbol{p}_i \qquad \text{(Update descent direction)}, \tag{2.5}$$

where the notation $\|\boldsymbol{y}\|_{\mathbf{Q}} = (\boldsymbol{y}^{\mathrm{t}}\mathbf{Q}\,\boldsymbol{y})^{1/2}$ denotes the $\mathbf{Q}$-norm of vector $\boldsymbol{y}$. In exact arithmetic, convergence is obtained in at most N steps. In (2.1)-(2.5), index i belongs to $\{1,\ldots,N\}$, which will remain implicit in the rest of the paper.

The basic CG method involves no preconditioning, *i.e.,* $\mathbf{M}$ identifies with the identity matrix. The use of a well-chosen, context-dependent preconditioning matrix can significantly improve the efficiency of CG method [9].

Let us first state the following orthogonal property of the PCG algorithm.

LEMMA 2.1. *For any initial guess* $\boldsymbol{u}_0 \in \mathbb{R}^N$*, the PCG algorithm* (2.1)-(2.5) *ensures that the residual* $\boldsymbol{r}_i$ *is orthogonal to the previous descent directions* $\boldsymbol{p}_0, \ldots, \boldsymbol{p}_{i-1}$

$$\boldsymbol{r}_i^{\mathrm{t}}\boldsymbol{p}_j = 0, \quad \forall i, \forall j < i. \tag{2.6}$$

Proof. ¿From [9, p. 534] we have $\boldsymbol{r}_i^{\mathrm{t}}\mathbf{M}^{-1}\boldsymbol{r}_j = 0$ for all i, $j < i$, and given (2.5) we deduce (2.6) by immediate recursion. □

In the sequel, a zero initial guess $\boldsymbol{u}_0 = \boldsymbol{0}$ is assumed.

LEMMA 2.2. *Let* $\boldsymbol{u}_0 = \boldsymbol{0}$*. The PCG algorithm* (2.1)-(2.5) *ensures that the residual* $\boldsymbol{r}_i$ *is orthogonal to the iterate* $\boldsymbol{u}_i$

$$\boldsymbol{r}_i^{\mathrm{t}}\boldsymbol{u}_i = \boldsymbol{0}, \quad \forall i, \tag{2.7}$$

and

$$\boldsymbol{b}^{\mathrm{t}}\boldsymbol{u}_i = \|\boldsymbol{u}_i\|_{\mathbf{A}}^2\,, \quad \forall i. \tag{2.8}$$

Proof.
According to (2.2), we have

$$\boldsymbol{u}_j = \boldsymbol{u}_0 + \sum_{\ell=0}^{j-1} \alpha_\ell \boldsymbol{p}_\ell, \quad \forall j.$$

Using (2.6), we deduce that $\boldsymbol{r}_i^{\mathrm{t}}\boldsymbol{u}_j = \boldsymbol{r}_i^{\mathrm{t}}\boldsymbol{u}_0$, for all i, $j \leq i$. Hence, (2.7) is readily obtained given $\boldsymbol{u}_0 = \mathbf{0}$. On the other hand, (2.2) and (2.3) imply $\boldsymbol{r}_{i+1} = \boldsymbol{r}_i - \mathbf{A}(\boldsymbol{u}_{i+1} - \boldsymbol{u}_i)$. ¿From $\boldsymbol{u}_0 = \mathbf{0}$, we deduce

$$\boldsymbol{r}_i = \boldsymbol{r}_0 - \mathbf{A}(\boldsymbol{u}_i - \boldsymbol{u}_0) = \boldsymbol{b} - \mathbf{A}\boldsymbol{u}_i, \quad \forall i \tag{2.9}$$

by immediate recursion. The latter identity yields $\boldsymbol{r}_i^{\mathrm{t}}\boldsymbol{u}_i = \boldsymbol{b}^{\mathrm{t}}\boldsymbol{u}_i - \boldsymbol{u}_i^{\mathrm{t}}\mathbf{A}\boldsymbol{u}_i$, for all i, from which we deduce (2.8), given (2.7). □

LEMMA 2.3. *Let $\boldsymbol{u}_0 = \mathbf{0}$. The PCG algorithm (2.1)-(2.5) ensures that*

$$\|\boldsymbol{u}_{i+1}\|_{\mathbf{M}}^2 \geq \|\boldsymbol{u}_i\|_{\mathbf{M}}^2\,, \quad \forall i. \tag{2.10}$$

Proof. According to (2.2) we have

$$\|\boldsymbol{u}_{i+1}\|_{\mathbf{M}}^2 = \|\boldsymbol{u}_i\|_{\mathbf{M}}^2 + \alpha_i^2\,\|\boldsymbol{p}_i\|_{\mathbf{M}}^2 + 2\alpha_i\boldsymbol{u}_i^{\mathrm{t}}\mathbf{M}\boldsymbol{p}_i \geq \|\boldsymbol{u}_i\|_{\mathbf{M}}^2 + 2\alpha_i\boldsymbol{u}_i^{\mathrm{t}}\mathbf{M}\boldsymbol{p}_i.$$

Since $\alpha_i \geq 0$, we deduce that (2.10) holds if

$$\boldsymbol{u}_i^{\mathrm{t}}\mathbf{M}\boldsymbol{p}_i \geq 0 \tag{2.11}$$

is true. Let us show the latter inequality by recursion on i. Since $\boldsymbol{u}_0 = 0$, we have $\boldsymbol{u}_0^{\mathrm{t}}\mathbf{M}\boldsymbol{p}_0 = 0$.

Let us assume now that (2.11) holds, and let us show that $\boldsymbol{u}_{i+1}^{\mathrm{t}}\mathbf{M}\boldsymbol{p}_{i+1} \geq 0$. According to (2.5), we have

$$\boldsymbol{u}_{i+1}^{\mathrm{t}}\mathbf{M}\boldsymbol{p}_{i+1} = \boldsymbol{u}_{i+1}^{\mathrm{t}}\boldsymbol{r}_{i+1} + \beta_i\boldsymbol{u}_{i+1}^{\mathrm{t}}\mathbf{M}\boldsymbol{p}_i.$$

According to (2.7) we deduce

$$\boldsymbol{u}_{i+1}^{\mathrm{t}}\mathbf{M}\boldsymbol{p}_{i+1} = \beta_i\boldsymbol{u}_{i+1}^{\mathrm{t}}\mathbf{M}\boldsymbol{p}_i.$$

Given (2.2) we get

$$\boldsymbol{u}_{i+1}^{\mathrm{t}}\mathbf{M}\boldsymbol{p}_{i+1} = \beta_i(\boldsymbol{u}_i^{\mathrm{t}}\mathbf{M}\boldsymbol{p}_i + \alpha_i\,\|\boldsymbol{p}_i\|_{\mathbf{M}}^2),$$

which is nonnegative since β_i, $\alpha_i \geq 0$. □

Consider a SPD matrix $\mathbf{Q} \in \mathbb{R}^{N\times N}$. Let $\nu_1(\mathbf{Q}) > 0$ and $\nu_2(\mathbf{Q}) > 0$ denote the smallest and largest eigenvalues of $\mathbf{Q}$, respectively, so that we have

$$\nu_1(\mathbf{Q})\,\|\boldsymbol{v}\|^2 \leq \|\boldsymbol{v}\|_{\mathbf{Q}}^2 \leq \nu_2(\mathbf{Q})\,\|\boldsymbol{v}\|^2\,, \quad \forall \boldsymbol{v} \in \mathbb{R}^N. \tag{2.12}$$

LEMMA 2.4. *Let $\boldsymbol{u}_0 = \mathbf{0}$. The PCG algorithm (2.1)-(2.5) ensures that*

$$\boldsymbol{b}^{\mathrm{t}}\boldsymbol{u}_i \geq \frac{\tau^2}{\nu_1(\mathbf{A})} \|\boldsymbol{b}\|^2, \quad \forall i, \tag{2.13}$$

where

$$\tau = \frac{\nu_1(\mathbf{M})\nu_1(\mathbf{A})}{\nu_2(\mathbf{M})\nu_2(\mathbf{A})} \in (0,1). \tag{2.14}$$

Proof. Given (2.12) we have

$$\mu_1 \|\boldsymbol{v}\|_{\mathbf{A}}^2 \leq \|\boldsymbol{v}\|_{\mathbf{M}}^2 \leq \mu_2 \|\boldsymbol{v}\|_{\mathbf{A}}^2, \quad \forall \boldsymbol{v} \in \mathbb{R}^N, \tag{2.15}$$

where $\mu_1 = \nu_1(\mathbf{M})/\nu_2(\mathbf{A}) > 0$ and $\mu_2 = \nu_2(\mathbf{M})/\nu_1(\mathbf{A}) > 0$.

¿From (2.10), by immediate recursion we get $\|\boldsymbol{u}_1\|_{\mathbf{M}}^2 \leq \|\boldsymbol{u}_i\|_{\mathbf{M}}^2$. According to (2.15), we deduce

$$\mu_1 \|\boldsymbol{u}_1\|_{\mathbf{A}}^2 \leq \|\boldsymbol{u}_1\|_{\mathbf{M}}^2 \leq \|\boldsymbol{u}_i\|_{\mathbf{M}}^2 \leq \mu_2 \|\boldsymbol{u}_i\|_{\mathbf{A}}^2 \tag{2.16}$$

On the other hand, (2.9) yields $\|\boldsymbol{u}_i\|_{\mathbf{A}}^2 = \boldsymbol{u}_i^{\mathrm{t}}\mathbf{A}\boldsymbol{u}_i = \boldsymbol{b}^{\mathrm{t}}\boldsymbol{u}_i - \boldsymbol{r}_i^{\mathrm{t}}\boldsymbol{u}_i$. According to (2.7), we deduce

$$\|\boldsymbol{u}_i\|_{\mathbf{A}}^2 = \boldsymbol{b}^{\mathrm{t}}\boldsymbol{u}_i. \tag{2.17}$$

In particular,

$$\|\boldsymbol{u}_1\|_{\mathbf{A}}^2 = \boldsymbol{b}^{\mathrm{t}}\boldsymbol{u}_1. \tag{2.18}$$

Given $\boldsymbol{u}_1 = \boldsymbol{u}_0 + \alpha_0\boldsymbol{p}_0 = \alpha_0\boldsymbol{p}_0 = \alpha_0\mathbf{M}^{-1}\boldsymbol{b}$ and

$$\alpha_0 = \frac{\|\boldsymbol{r}_0\|_{\mathbf{M}^{-1}}^2}{\|\boldsymbol{p}_0\|_{\mathbf{A}}^2} = \frac{\|\boldsymbol{b}\|_{\mathbf{M}^{-1}}^2}{\|\mathbf{M}^{-1}\boldsymbol{b}\|_{\mathbf{A}}^2},$$

we have

$$\boldsymbol{b}^{\mathrm{t}}\boldsymbol{u}_1 = \frac{\|\boldsymbol{b}\|_{\mathbf{M}^{-1}}^4}{\|\mathbf{M}^{-1}\boldsymbol{b}\|_{\mathbf{A}}^2} = \frac{\left\|\mathbf{M}^{-1}\boldsymbol{b}\right\|_{\mathbf{M}}^4}{\|\mathbf{M}^{-1}\boldsymbol{b}\|_{\mathbf{A}}^2}.$$

According to (2.15), we deduce

$$\boldsymbol{b}^{\mathrm{t}}\boldsymbol{u}_1 \geq \mu_1 \left\|\mathbf{M}^{-1}\boldsymbol{b}\right\|_{\mathbf{M}}^2 = \mu_1 \|\boldsymbol{b}\|_{\mathbf{M}^{-1}}^2.$$

According to (2.15) and (2.12), we obtain

$$\boldsymbol{b}^{\mathrm{t}}\boldsymbol{u}_1 \geq \mu_1\nu_1(\mathbf{M}^{-1}) \|\boldsymbol{b}\|^2 = \frac{\mu_1}{\nu_2(\mathbf{M})} \|\boldsymbol{b}\|^2 = \frac{\mu_1}{\mu_2\nu_1(\mathbf{A})} \|\boldsymbol{b}\|^2. \tag{2.19}$$

Finally, according to (2.16), (2.17), (2.18), (2.19), and $\tau = \mu_1/\mu_2$, we deduce

$$\boldsymbol{b}^{\mathrm{t}}\boldsymbol{u}_i \geq \tau\boldsymbol{b}^{\mathrm{t}}\boldsymbol{u}_1 \geq \frac{\tau^2}{\nu_1(\mathbf{A})} \|\boldsymbol{b}\|^2.$$

□

LEMMA 2.5. *Let $\boldsymbol{u}_0 = \mathbf{0}$. The PCG algorithm (2.1)-(2.5) ensures that*

$$\|\boldsymbol{u}_i\| \leq \frac{1}{\tau^{1/2}\nu_1(\mathbf{A})}\|\boldsymbol{b}\|, \quad \forall i, \tag{2.20}$$

where τ is defined by (2.14).

Proof. ¿From (2.10), we get $\|\boldsymbol{u}_i\|_{\mathbf{M}}^2 \leq \|\boldsymbol{u}_N\|_{\mathbf{M}}^2$ by immediate recursion. According to (2.15), we obtain

$$\|\boldsymbol{u}_i\|_{\mathbf{A}}^2 \leq \frac{1}{\tau}\|\boldsymbol{u}_N\|_{\mathbf{A}}^2. \tag{2.21}$$

According to (2.12), we have

$$\|\boldsymbol{u}_i\|^2 \leq \frac{1}{\nu_1(\mathbf{A})}\|\boldsymbol{u}_i\|_{\mathbf{A}}^2. \tag{2.22}$$

The PCG algorithm (2.1)-(2.5) ensures that $\boldsymbol{u}_N = \mathbf{A}^{-1}\boldsymbol{b}$ after N iterations. Hence,

$$\|\boldsymbol{u}_N\|_{\mathbf{A}}^2 = \left\|\mathbf{A}^{-1}\boldsymbol{b}\right\|_{\mathbf{A}}^2 = \|\boldsymbol{b}\|_{\mathbf{A}^{-1}}^2 \leq \nu_2(\mathbf{A}^{-1})\|\boldsymbol{b}\|^2 = \frac{1}{\nu_1(\mathbf{A})}\|\boldsymbol{b}\|^2, \tag{2.23}$$

according to (2.12). Finally, (2.20) is easily deduced from (2.22), (2.21) and (2.23). □

3. Global convergence.

3.1. Results for general criterion. For the sake of generality, the convergence of algorithm (1.10)-(1.13) is established for a generic SPD matrix A_k. We also consider a generic differentiable, lower bounded criterion $\mathcal{J}$ in a neighborhood $\mathcal{N}$ of the level set $\mathcal{L}(\boldsymbol{x}_0) = \{\boldsymbol{x} \in \mathbb{R}^N | \mathcal{J}(\boldsymbol{x}) \leq \mathcal{J}(\boldsymbol{x}_0)\}$.

DEFINITION 3.1. *A sequence of SPD matrices $\mathcal{Q} = \{\mathbf{Q}_k\} \in \mathbb{R}^{N\times N}$ has a uniformly bounded spectrum with a strictly positive lower bound if there exist $\nu_1(\mathcal{Q})$, $\nu_2(\mathcal{Q}) \in \mathbb{R}$ such that*

$$\nu_2(\mathcal{Q}) \geq \nu_2(\mathbf{Q}_k) \geq \nu_1(\mathbf{Q}_k) \geq \nu_1(\mathcal{Q}) > 0, \quad \forall k.$$

For the sake of brevity, $\mathcal{Q}$ will then be said uniformly bounded.

The following assumptions are considered.

ASSUMPTION 3.2. *The matrix sequence $\mathcal{A} = \{\mathbf{A}_k\}$ used in (1.12) to determine the search direction (1.11) and the preconditioning matrix sequence $\mathcal{M} = \{\mathbf{M}_k\}$ in the PCG algorithm are uniformly bounded.*

ASSUMPTION 3.3. *The matrix sequence $\mathcal{B} = \{\mathbf{B}_k\}$ in the stepsize formula (1.13) is uniformly bounded and such that $\mathcal{J}$ is upper bounded by $\widehat{\mathcal{J}}_k(\cdot, \boldsymbol{x})$, i.e.,*

$$\widehat{\mathcal{J}}_k(\boldsymbol{x}^+, \boldsymbol{x}) \geq \mathcal{J}(\boldsymbol{x}^+), \quad \forall \boldsymbol{x}, \boldsymbol{x}^+ \in \mathcal{N}, \forall k, \tag{3.1}$$

where

$$\widehat{\mathcal{J}}_k(\boldsymbol{x}^+, \boldsymbol{x}) = \mathcal{J}(\boldsymbol{x}) + (\boldsymbol{x}^+ - \boldsymbol{x})^{\mathrm{t}}\nabla\mathcal{J}(\boldsymbol{x}) + (\boldsymbol{x}^+ - \boldsymbol{x})^{\mathrm{t}}\,\mathbf{B}_k\,(\boldsymbol{x}^+ - \boldsymbol{x})/2. \tag{3.2}$$

LEMMA 3.4. *Let $\boldsymbol{x}_k$ be defined by (1.10)-(1.13) with $\theta \in (0, 2)$, and let Assumptions 3.2 and 3.3 hold. Then (1.11) defines a descent direction in $\boldsymbol{x}_k$, i.e.,*

$$\boldsymbol{d}_k^{\mathrm{t}}\nabla\mathcal{J}(\boldsymbol{x}_k) < 0, \quad \forall k,$$

and the stepsize sequence $\{\alpha_k\}$ *is positive. Moreover, the sequence* $\{\mathcal{J}(\boldsymbol{x}_k)\}$ *is nonincreasing:*

$$\mathcal{J}(\boldsymbol{x}_k) \geq \mathcal{J}(\boldsymbol{x}_{k+1}), \quad \forall k. \tag{3.3}$$

Thus, the whole sequence $\{\boldsymbol{x}_k\}$ *stays inside* $\mathcal{N}$*.*

Proof. Let $\boldsymbol{b} = -\nabla\mathcal{J}(\boldsymbol{x}_k)$, $\mathbf{A} = \mathbf{A}_k$ and $\boldsymbol{d}_k = \boldsymbol{u}_{I_k}(\boldsymbol{x}_k)$. According to (2.8) we have

$$-\boldsymbol{d}_k^{\mathrm{t}}\nabla\mathcal{J}(\boldsymbol{x}_k) = \boldsymbol{d}_k^{\mathrm{t}}\mathbf{A}_k\boldsymbol{d}_k \tag{3.4}$$

which is strictly positive for $\boldsymbol{d}_k \neq \mathbf{0}$ since $\mathbf{A}_k$ is SPD. From (2.10) we have

$$\|\boldsymbol{d}_k\|_{\mathbf{M}}^2 = \|\boldsymbol{u}_{I_k}\|_{\mathbf{M}}^2 \geq \|\boldsymbol{u}_1\|_{\mathbf{M}}^2 = \alpha_0^2\,\|\boldsymbol{b}\|_{\mathbf{M}^{-1}}^2 = \alpha_0^2\,\|\nabla\mathcal{J}(\boldsymbol{x}_k)\|_{\mathbf{M}^{-1}}^2 .$$

Thus, $\boldsymbol{d}_k$ is a descent direction and from (1.13) the stepsize α_k is positive. The nonincreasing property of the sequence $\{\mathcal{J}(\boldsymbol{x}_k)\}$ is a corollary of [11, Lemma 3.1], which is true for a generic differentiable criterion in $\mathcal{N}$ and a generic search direction $\boldsymbol{d}_k$[1]. □

Definition 3.5. *The stepsize sequence* $\{\alpha_k\}$ *satisfies the Armijo condition with* $\Omega \in (0,1)$ *if*

$$\mathcal{J}(\boldsymbol{x}_k) - \mathcal{J}(\boldsymbol{x}_{k+1}) + \Omega\alpha_k\boldsymbol{d}_k^{\mathrm{t}}\nabla\mathcal{J}(\boldsymbol{x}_k) \geq 0, \quad \forall k. \tag{3.5}$$

Lemma 3.6. *Let* $\boldsymbol{x}_k$ *be defined by* (1.10)-(1.13) *with* $\theta \in (0,2)$*, and let Assumptions 3.2 and 3.3 hold. Then the stepsize sequence* $\{\alpha_k\}$ *defined by* (1.13) *satisfies the Armijo condition with* $\Omega = 1 - \theta/2 \in (0,1)$*.*

Proof. According to (3.1) we have

$$\widehat{\mathcal{J}}_k(\boldsymbol{x}_{k+1}, \boldsymbol{x}_k) \geq \mathcal{J}(\boldsymbol{x}_{k+1}).$$

Given (1.10) and (3.2), we get

$$\mathcal{J}(\boldsymbol{x}_k) - \mathcal{J}(\boldsymbol{x}_{k+1}) + \alpha_k\,\boldsymbol{d}_k^{\mathrm{t}}\nabla\mathcal{J}(\boldsymbol{x}_k) + \alpha_k^2\,\boldsymbol{d}_k^{\mathrm{t}}\mathbf{B}_k\boldsymbol{d}_k/2 \geq 0,$$

which also reads

$$\mathcal{J}(\boldsymbol{x}_k) - \mathcal{J}(\boldsymbol{x}_{k+1}) + \left(1 + \alpha_k\frac{\boldsymbol{d}_k^{\mathrm{t}}\mathbf{B}_k\boldsymbol{d}_k}{2\boldsymbol{d}_k^{\mathrm{t}}\nabla\mathcal{J}(\boldsymbol{x}_k)}\right)\alpha_k\boldsymbol{d}_k^{\mathrm{t}}\nabla\mathcal{J}(\boldsymbol{x}_k) \geq 0. \tag{3.6}$$

According to (1.13), we get the Armijo condition (3.5) with $\Omega = 1 - \theta/2 \in (0,1)$. □

Definition 3.7. *[2, p. 35] The direction sequence* $\{\boldsymbol{d}_k\}$ *is gradient related to* $\{\boldsymbol{x}_k\}$ *if for any subsequence* $\{\boldsymbol{x}_k\}_{k\in\mathcal{K}}$ *that converges to a nonstationary point, the corresponding subsequence* $\{\boldsymbol{d}_k\}_{k\in\mathcal{K}}$ *is bounded and satisfies*

$$\limsup_{k\to\infty, k\in\mathcal{K}} \boldsymbol{d}_k^{\mathrm{t}}\nabla\mathcal{J}(\boldsymbol{x}_k) < 0.$$

[1] In [11, Lemma 3.1], a generalized, iterated formula is considered for the stepsize. For the sake of simplicity, the present paper is restricted to the stepsize yielded by a single iteration.

Lemma 3.8. *Let $\boldsymbol{x}_k$ be defined by (1.10)-(1.13), and let Assumptions 3.2 and 3.3 hold. Then we have*

$$\frac{\mathcal{T}^2}{\nu_1(\mathcal{A})}\|\nabla\mathcal{J}(\boldsymbol{x}_k)\|^2 \leq -\boldsymbol{d}_k^{\mathrm{t}}\nabla\mathcal{J}(\boldsymbol{x}_k) \tag{3.7}$$

and

$$\|\boldsymbol{d}_k\| \leq \frac{1}{\mathcal{T}^{1/2}\nu_1(\mathcal{A})}\|\nabla\mathcal{J}(\boldsymbol{x}_k)\|, \tag{3.8}$$

where

$$\mathcal{T} = \frac{\nu_1(\mathcal{M})\nu_1(\mathcal{A})}{\nu_2(\mathcal{M})\nu_2(\mathcal{A})} \in (0,1). \tag{3.9}$$

Thus, the direction sequence $\{\boldsymbol{d}_k\}$ is gradient related to $\{\boldsymbol{x}_k\}$.

Proof. For any value of k, let $\boldsymbol{b} = -\nabla\mathcal{J}(\boldsymbol{x}_k)$, $\mathbf{A} = \mathbf{A}_k$, $\mathbf{M} = \mathbf{M}_k$ and $\boldsymbol{d}_k = \boldsymbol{u}_{I_k}(\boldsymbol{x}_k)$. Since both sequences $\mathcal{A}$ and $\mathcal{M}$ are uniformly bounded, it is easy to see that (2.14) and (3.9) imply $\mathcal{T} \leq \tau$, as well as

$$\frac{\mathcal{T}^2}{\nu_1(\mathcal{A})} \leq \frac{\tau^2}{\nu_1(\mathbf{A})}$$

and

$$\frac{1}{\tau^{1/2}\nu_1(\mathbf{A})} \leq \frac{1}{\mathcal{T}^{1/2}\nu_1(\mathcal{A})}.$$

Then (3.7) and (3.8) are obvious consequences of (2.13) and (2.20), respectively. According to [2, p. 36], inequalities (3.7) and (3.8) are sufficient conditions to ensure that $\{\boldsymbol{d}_k\}$ is gradient related to $\{\boldsymbol{x}_k\}$. □

Lemma 3.9. *The stepsize formula (1.13) identifies with the constant stepsize θ of HQ algorithms (1.3)-(1.4) when $\mathbf{A}_k$ and $\mathbf{B}_k$ are chosen accordingly, that is either $\mathbf{A}_k = \mathbf{B}_k = \mathbf{B}_{\mathrm{GY}}^a$, or $\mathbf{A}_k = \mathbf{B}_k = \mathbf{B}_{\mathrm{GR}}(\boldsymbol{x}_k)$.*

Proof. Obvious given (1.13) and (3.4). □

Theorem 3.10. *Let $\boldsymbol{x}_k$ be defined by (1.10)-(1.13) with $\theta \in (0,2)$, and let Assumptions 3.2 and 3.3 hold. Then we have convergence in the sense*

$$\lim_{k\to\infty} \nabla\mathcal{J}(\boldsymbol{x}_k) = \mathbf{0}.$$

Proof. Given (3.4) and Lemma 3.6, we have

$$\mathcal{J}(\boldsymbol{x}_k) - \mathcal{J}(\boldsymbol{x}_{k+1}) \geq -\alpha_k\Omega\boldsymbol{d}_k^{\mathrm{t}}\nabla\mathcal{J}(\boldsymbol{x}_k) = \alpha_k\Omega\boldsymbol{d}_k^{\mathrm{t}}\mathbf{A}_k\boldsymbol{d}_k.$$

Thus, according to (1.13) we get

$$\mathcal{J}(\boldsymbol{x}_k) - \mathcal{J}(\boldsymbol{x}_{k+1}) \geq -\theta\Omega\boldsymbol{d}_k^{\mathrm{t}}\nabla\mathcal{J}(\boldsymbol{x}_k)\frac{\boldsymbol{d}_k^{\mathrm{t}}\mathbf{A}_k\boldsymbol{d}_k}{\boldsymbol{d}_k^{\mathrm{t}}\mathbf{B}_k\boldsymbol{d}_k} \geq -\theta\Omega\boldsymbol{d}_k^{\mathrm{t}}\nabla\mathcal{J}(\boldsymbol{x}_k)\frac{\nu_1(\mathcal{A})}{\nu_2(\mathcal{B})}. \tag{3.10}$$

Let $\Omega_1 = \theta\Omega\mathcal{T}^2/\nu_2(\mathcal{B}) > 0$. Inequalities (3.7) and (3.10) yield

$$\mathcal{J}(\boldsymbol{x}_k) - \mathcal{J}(\boldsymbol{x}_{k+1}) \geq \Omega_1\|\nabla\mathcal{J}(\boldsymbol{x}_k)\|^2 \geq 0. \tag{3.11}$$

On the other hand, given (3.3) and the lower boundedness of $\mathcal{J}$, we have

$$\mathcal{J}(\boldsymbol{x}_\ell) \geq \inf_{\boldsymbol{x}\in\mathcal{N}} \mathcal{J}(\boldsymbol{x}) > -\infty, \quad \forall \ell.$$

Then (3.11) allows to deduce

$$\infty > \mathcal{J}(\boldsymbol{x}_0) - \inf_{\boldsymbol{x}\in\mathcal{N}} \mathcal{J}(\boldsymbol{x}) \geq \mathcal{J}(\boldsymbol{x}_0) - \mathcal{J}(\boldsymbol{x}_\ell) \geq \Omega_1 \sum_{k=0}^{\ell-1} \|\nabla \mathcal{J}(\boldsymbol{x}_k)\|^2, \quad \forall \ell. \tag{3.12}$$

Hence, $\lim_{k\to\infty} \nabla\mathcal{J}(\boldsymbol{x}_k) = \mathbf{0}$. □

3.2. Application to PLS criteria. The global convergence result of Subsection 3.1 is now applied to the PLS criterion (1.1) with either $\mathbf{A}_k = \mathbf{B}_{\mathrm{GR}}(\boldsymbol{x}_k)$, $\mathbf{A}_k = \mathbf{B}^a_{\mathrm{GY}}$ or $\mathbf{A}_k = \nabla^2\mathcal{J}(\boldsymbol{x}_k)$, which take the respective forms given by (1.5), (1.7) and (1.9). Let us consider the following assumptions on the function ϕ. The first two are associated with the HQ+TPCG schemes, while the last one is associated with the N+TPCG variant.

Assumption 3.11.

(i) *ϕ is $\mathcal{C}^1$ and coercive,*
(ii) *ϕ' is L-Lipschitz.*

Assumption 3.12.

(i) *ϕ is $\mathcal{C}^1$, even and coercive,*
(ii) *$\phi(\sqrt{\cdot})$ is concave on $\mathbb{R}^+$,*
(iii) $0 < \phi'(t)/t < \infty, \quad \forall t \in \mathbb{R}$.

Assumption 3.13.

(i) *ϕ is $\mathcal{C}^2$, even and strictly convex,*
(ii) $0 < \phi''(t) < \infty, \quad \forall t \in \mathbb{R}$.

Let us introduce an additional assumption, which is a necessary condition to ensure that the penalization term (1.2) regularizes the problem of restoring $\boldsymbol{x}$ from $\boldsymbol{y}$ in a proper way.

Assumption 3.14. **H** *and* **V** *are such that*

$$\ker(\mathbf{H}^{\mathrm{t}}\mathbf{H}) \cap \ker(\mathbf{V}^{\mathrm{t}}\mathbf{V}) = \{\mathbf{0}\}. \tag{3.13}$$

The following lemma ensures the majorization character of quadratic approximations involving HQ matrices.

Lemma 3.15. *Let $\mathcal{J}$ be defined by* (1.1)*, where* **H** *and* **V** *satisfy Assumption 3.14.*

- *If Assumption 3.11 holds, then $\mathcal{B} = \{\mathbf{B}^a_{\mathrm{GY}}\}$ satisfies Assumption 3.3 for all $a \in (0, 1/L)$.*
- *If Assumption 3.12 holds, then $\mathcal{B} = \{\mathbf{B}_{\mathrm{GR}}(\boldsymbol{x}_k)\}$ satisfies Assumption 3.3.*

Proof. Let us first show that $\mathcal{B}$ is uniformly bounded. According to (3.13) the proof is immediate for $\mathcal{B} = \{\mathbf{B}^a_{\mathrm{GY}}\}$ with $0 < a$, since $\mathbf{B}^a_{\mathrm{GY}}$ is then a SPD matrix. ¿From [1, Prop. 8], $\mathcal{B} = \{\mathbf{B}_{\mathrm{GR}}(\boldsymbol{x}_k)\}$ is also uniformly bounded.

On the other hand, from [1, Prop. 1], (3.1) holds when $\mathbf{B}_k = \mathbf{B}^a_{\mathrm{GY}}$ with $0 < a < 1/L$, and also when $\mathbf{B}_k = \mathbf{B}_{\mathrm{GR}}(\boldsymbol{x}_k))$. Note that in [1, Prop. 1 and 8], the assumption that ϕ is convex can be replaced by the assumption that ϕ is coercive. □

The following theorem establishes the convergence of the HQ+TPCG schemes.

Theorem 3.16. *Let $\mathcal{J}$ be defined by* (1.1)*, where* **H** *and* **V** *satisfy Assumption 3.14. Let $\boldsymbol{x}_k$ be defined by* (1.10)-(1.13) *where $\theta \in (0, 2)$, $\mathcal{M}$ is a uniformly* *bounded sequence of SPD matrices, and* $\mathbf{A}_k = \mathbf{B}_k = \mathbf{B}_{\mathrm{GY}}^a$ *with* $0 < a < 1/L$ *(resp.,* $\mathbf{A}_k = \mathbf{B}_k = \mathbf{B}_{\mathrm{GR}}(\boldsymbol{x}_k)$*). If Assumption 3.11 (resp., Assumption 3.12) holds, then the sequence* $\{\mathcal{J}(\boldsymbol{x}_k)\}$ *is nonincreasing:*

$$\mathcal{J}(\boldsymbol{x}_k) \geq \mathcal{J}(\boldsymbol{x}_{k+1}), \quad \forall k, \tag{3.14}$$

and we have convergence in the following sense:

$$\lim_{k\to\infty} \nabla\mathcal{J}(\boldsymbol{x}_k) = \mathbf{0}. \tag{3.15}$$

Proof. According to Assumptions 3.11(i), 3.12(i) and to (3.13), the PLS criterion $\mathcal{J}$ defined by (1.1) is differentiable and lower bounded. From Lemma 3.15, Assumption 3.3 holds. Then, (3.14) and (3.15) result from the direct application of Lemma 3.4 and Theorem 3.10, respectively. □

In a similar way, the following theorem establishes the convergence of the N+TPCG variant.

THEOREM 3.17. *Let* $\mathcal{J}$ *be defined by* (1.1)*, where* $\mathbf{H}$ *and* $\mathbf{V}$ *satisfy Assumption 3.14. Let* $\boldsymbol{x}_k$ *be defined by* (1.10)-(1.13) *where* $\theta \in (0, 2)$*,* $\mathcal{M}$ *is a uniformly bounded sequence of SPD matrices,* $\mathbf{A}_k = \nabla^2\mathcal{J}(\boldsymbol{x}_k)$*, and* $\mathbf{B}_k = \mathbf{B}_{\mathrm{GY}}^a$ *with* $0 < a < 1/L$ *(resp.,* $\mathbf{B}_k = \mathbf{B}_{\mathrm{GR}}(\boldsymbol{x}_k)$*). If Assumption 3.13 holds as well as Assumption 3.11 (resp., Assumption 3.12), then the sequence* $\{\mathcal{J}(\boldsymbol{x}_k)\}$ *is nonincreasing and converging in the sense of* (3.15).

Proof. Let us first remark that ϕ is coercive, since it is even and strictly convex. Then, according to Assumption 3.13(i) and to (3.13), the PLS criterion $\mathcal{J}$ defined by (1.1) is differentiable and lower bounded. Thus, from (1.9) and Assumption 3.13(ii), the sequence $\left\{\nabla^2\mathcal{J}(\boldsymbol{x}_k)\right\}$ is uniformly bounded if $\{\boldsymbol{x}_k\} \in \mathcal{N}$. The latter is true by immediate recursion, given (3.3). Hence, Assumption 3.2 holds with $\mathbf{A}_k = \nabla^2\mathcal{J}(\boldsymbol{x}_k)$. ¿From Lemma 3.15, Assumption 3.3 also holds. Then, (3.14) and (3.15) result from the direct application of Lemma 3.4 and Theorem 3.10, respectively. □

4. Numerical results. Two image restoration problems are considered here. In the first one, $\boldsymbol{y}$ is a noisy, blurred image obtained using a Gaussian point spread function (PSF) of standard deviation 2.24 and of size 17 × 17. Moreover, the convolution product is implemented under the zero boundary condition (*i.e.*, the Dirichlet boundary condition). In the second problem, only noise is added to the original image. In both cases, the noise is white Gaussian, and the original image is the fishing boat image ($N = 512 \times 512$, 8 bits/pixel). The first-order difference matrix $\mathbf{V}$ is considered, and it is easy to show that Assumption 3.14 holds for both problems. The edge-preserving function $\phi_{\mathrm{hyp}}(u) = \sqrt{\delta^2 + |u|^2}$ is considered. It is a strictly convex function that fulfills all the conditions of Assumptions 3.11, 3.12 and 3.13, so that Theorem 3.16 and Theorem 3.17 apply, ensuring the convergence of HQ+TPCG and N+TPCG schemes, respectively. The convex PLS criterion associated with each problem depends on the parameters λ and δ. They are assessed so that the restored image is visually the closest to the original image. Fig. 4.1 respectively displays:

(a) the noisy image at a signal-to-noise ratio (SNR) of 20dB,
(b) the blurred and noisy image at a SNR of 40dB,
(c) the denoised image deduced from (a) with $\delta = 13$ and $\lambda = 10$,
(d) the restored image deduced from (b) with $\delta = 13$ and $\lambda = 0.2$.

The purpose of this section is to test the convergence speed for different members of the HQ/N+TPCG family. Both HQ matrices $\mathbf{B}_{\mathrm{GY}}^a$ and $\mathbf{B}_{\mathrm{GR}}(\mathbf{0})$ are close

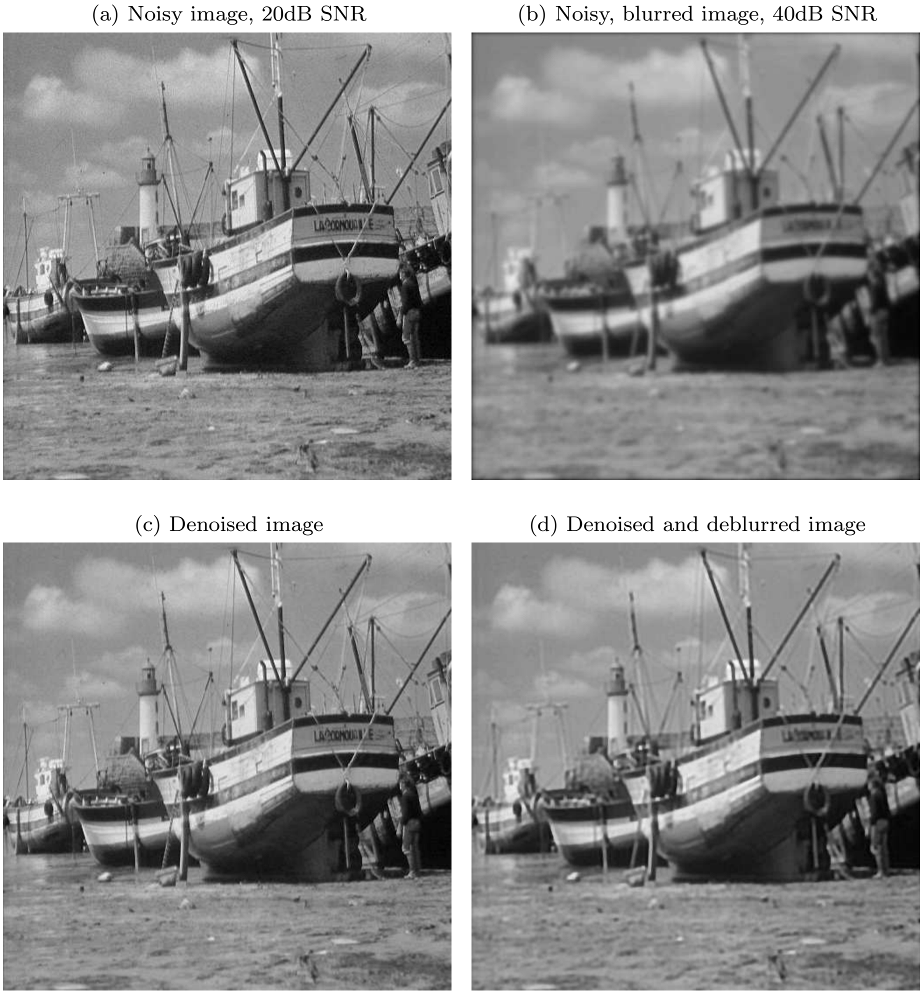
(a) Noisy image, 20dB SNR
(b) Noisy, blurred image, 40dB SNR
(c) Denoised image
(d) Denoised and deblurred image

FIG. 4.1. *(a)-(b) Degraded images. (c)-(d) Restored images by minimization of edge-preserving criteria.*

to block Toeplitz-plus-Hankel matrices. Thus, preconditioning based on 2D Cosine Transform (CT) is considered since the PSF is symmetric [13]. The global stopping rule $\|\nabla\mathcal{J}(\boldsymbol{x}_k)\|/\sqrt{N} < 10^{-4}$ is considered, while the TPCG is used with the stopping rule suggested in [14]:

$$\|\boldsymbol{r}_i\|/\|\boldsymbol{r}_0\| < \eta. \tag{4.1}$$

Note that this stopping rule corresponds to a varying sequence $\{I_k\}$. The parameter η controls the accuracy of the TPCG. The smaller it is, the more accurate the solving of (1.12). In [14], η was chosen as small as 10^{-6}. Here several values are tested within the range $[10^{-6}, 1]$, which has been empirically chosen to allow each algorithm to reach its best performance. Following [15], we choose $\theta = 1$ and $a = 1/\phi''(0) = \delta$ in the case of the GY scheme.

The experiments were undertaken with Matlab 7.1 on a PC P4 3GHz RAM 2Gb. For each tested scheme, Tab. 4.1 and 4.2 display iteration numbers under the form

	DENOISING				DEBLURRING			
	No precond.		CT precond.		No precond.		CT precond.	
	Iter.	Time	Iter.	Time	Iter.	Time	Iter.	Time
GR+TPCG(0.9)	21/1.0	28.2	19/1.0	35.9	56/4.2	508.3	53/1.3	256.8
GR+TPCG(0.8)	21/1.0	28.2	19/1.0	36.2	38/5.4	419.0	44/1.6	241.6
GR+TPCG(0.7)	21/1.0	28.1	19/1.0	35.6	33/6.2	$\underline{409.2}$	33/2.2	224.1
GR+TPCG(0.6)	14/1.1	19.4	19/1.0	36.4	30/7.1	414.7	27/2.4	199.3
GR+TPCG(0.5)	10/1.5	14.2	19/1.0	36.0	27/9.1	465.7	26/2.5	$\underline{194.7}$
GR+TPCG(0.4)	9/1.8	$\underline{12.9}$	13/1.2	26.7	26/10.4	501.8	24/2.8	197.4
GR+TPCG(0.3)	9/1.9	13.2	12/1.5	26.8	25/12.7	577.9	23/3.1	201.8
GR+TPCG(0.2)	9/1.9	13.2	10/1.9	$\underline{24.9}$	24/14.6	631.9	22/4.4	256.9
GR+TPCG(0.1)	11/2.4	16.2	10/1.9	25.2	23/19.0	775.7	21/5.6	301.9
GR+TPCG(10^{-2})	11/4.0	18.2	10/3.8	36.3	21/35.1	1273.3	21/10.5	526.3
GR+TPCG(10^{-3})	11/6.2	20.8	11/5.2	48.8	21/51.7	1852.0	21/15.5	754.2
GR+TPCG(10^{-4})	11/8.3	22.9	11/6.8	58.9	21/68.9	2448.1	21/20.4	980.6
GR+TPCG(10^{-5})	11/10.4	25.7	11/8.7	72.1	21/86.5	3079.0	21/26.1	1228.2
GR+TPCG(10^{-6})	11/12.4	27.6	11/10.6	83.7	21/103.4	3655.8	21/31.8	1472.1
GY+TPCG(0.9)	21/1.0	9.0	22/1.0	$\underline{21.8}$	124/3.2	791.5	150/1.0	489.2
GY+TPCG(0.8)	21/1.0	8.9	22/1.0	$\underline{22.2}$	98/3.7	703.8	131/1.1	445.3
GY+TPCG(0.7)	21/1.0	9.0	22/1.0	$\underline{21.9}$	84/4.1	660.1	123/1.1	440.7
GY+TPCG(0.6)	21/1.0	9.1	22/1.0	$\underline{22.1}$	79/4.3	$\underline{650.7}$	77/1.5	$\underline{335.4}$
GY+TPCG(0.5)	18/1.1	8.0	22/1.0	$\underline{21.9}$	76/4.6	669.1	77/1.6	362.2
GY+TPCG(0.4)	17/1.2	$\underline{7.8}$	22/1.0	$\underline{21.6}$	80/6.4	942.3	76/2.1	432.5
GY+TPCG(0.3)	19/1.3	8.9	22/1.0	$\underline{22.1}$	80/7.3	1057.0	75/2.3	451.4
GY+TPCG(0.2)	21/1.7	10.5	22/1.0	$\underline{21.6}$	80/7.8	1124.9	77/2.5	511.2
GY+TPCG(0.1)	23/2.0	12.3	23/2.0	35.8	79/11.3	1572.4	76/3.2	621.5
GY+TPCG(10^{-2})	23/3.6	16.1	23/2.3	41.8	78/20.1	2715.9	78/6.3	1163.5
GY+TPCG(10^{-3})	23/5.6	21.1	23/3.6	60.4	77/31.0	4065.0	77/10.2	1802.3
GY+TPCG(10^{-4})	23/7.7	26.6	23/5.0	78.8	77/41.9	5493.6	77/14.2	2484.0
GY+TPCG(10^{-5})	23/9.9	32.3	23/6.1	96.5	77/52.8	6875.7	77/18.2	3096.2
GY+TPCG(10^{-6})	23/12.2	38.2	23/7.2	113.1	77/64.0	8322.1	77/22.2	3805.7

TABLE 4.1

Comparison of HQ+TPCG algorithms for the denoising and deblurring problems. The best minimization time (s) for each algorithm is underlined.

g/s, where g is the number of global iterations, and s is the average number of TPCG subiterations per global iteration. Global minimization times in seconds are also given for each algorithm.

As expected, the number of global iterations g generally increases with η, while the average number of TPCG subiterations s decreases. Only the number of global iterations of GY+TPCG and N+TPCG with $\mathbf{B}_k = \mathbf{B}^a_{\text{GY}}$ slightly departs from this common behavior in the denoising case, both without preconditioning. The GR schemes perform better than the GY schemes for the deblurring problem while they show similar performance for the denoising problem. As expected, the N+TPCG schemes based on the Hessian show a smaller number of global iterations than HQ+TPCG when equation (1.12) is solved accurately (η small). Nonetheless, for all tested schemes, the smallest global minimization time is achieved when a tradeoff is reached between the number of global iterations and the number of TPCG iterations. In all cases, we observed that the corresponding best value of η is comprised between 0.1 and 1, which

	DENOISING				DEBLURRING			
Stepsize with	No precond.		CT precond.		No precond.		CT precond.	
$\mathbf{B}_k = \mathbf{B}_{\mathrm{GR}}(\boldsymbol{x}_k)$:	Iter.	Time	Iter.	Time	Iter.	Time	Iter.	Time
N+TPCG(0.9)	21/1.0	37.9	19/1.0	44.7	43/6.4	582.8	39/2.1	$\underline{288.2}$
N+TPCG(0.8)	21/1.0	38.0	19/1.0	45.1	38/6.6	544.3	36/2.4	290.0
N+TPCG(0.7)	21/1.0	38.1	19/1.0	45.5	34/7.6	$\underline{534.3}$	30/3.5	312.3
N+TPCG(0.6)	11/1.3	20.7	19/1.1	45.8	33/9.0	602.2	28/3.5	296.0
N+TPCG(0.5)	9/1.6	$\underline{17.0}$	11/1.6	30.9	29/8.9	514.0	27/4.6	346.9
N+TPCG(0.4)	9/1.8	17.3	10/1.7	29.3	31/12.0	721.5	29/4.2	348.6
N+TPCG(0.3)	9/1.9	17.6	10/1.8	$\underline{29.1}$	26/20.3	966.1	27/5.6	405.1
N+TPCG(0.2)	7/2.1	13.8	9/1.9	26.5	25/20.9	960.6	25/7.6	484.9
N+TPCG(0.1)	7/2.9	14.7	7/2.7	24.7	17/28.8	884.9	26/8.6	560.9
N+TPCG(10^{-2})	8/4.8	18.3	8/4.6	36.8	11/66.4	1278.2	17/15.4	601.2
N+TPCG(10^{-3})	8/6.9	19.7	8/6.6	46.0	9/101.2	1545.5	9/29.8	615.1
N+TPCG(10^{-4})	8/8.9	21.0	8/8.5	56.0	9/160.8	2438.8	9/49.6	1006.8
N+TPCG(10^{-5})	8/10.9	22.9	8/10.4	66.1	9/208.9	3166.4	9/64.7	1282.2
N+TPCG(10^{-6})	8/12.9	24.4	8/13.1	79.6	9/252.2	3820.3	9/78.7	1553.0
$\mathbf{B}_k = \mathbf{B}^a_{\mathrm{GY}}$:								
N+TPCG(0.9)	21/1.0	38.0	22/1.0	52.9	95/5.2	$\underline{1088.0}$	74/3.0	695.0
N+TPCG(0.8)	21/1.0	37.9	22/1.0	51.5	90/8.0	1468.8	71/3.4	735.5
N+TPCG(0.7)	21/1.0	38.1	22/1.0	51.7	85/8.2	1425.1	61/3.5	$\underline{641.2}$
N+TPCG(0.6)	10/1.6	19.3	22/1.0	52.6	92/9.8	1797.7	67/5.4	984.3
N+TPCG(0.5)	10/1.6	18.7	22/1.0	51.9	96/8.6	1693.4	72/5.3	1032.2
N+TPCG(0.4)	10/1.6	19.6	22/1.0	52.2	93/9.6	1771.9	69/7.0	1252.2
N+TPCG(0.3)	9/1.9	$\underline{17.7}$	19/1.9	56.2	74/19.8	2723.2	71/6.3	1176.6
N+TPCG(0.2)	9/1.9	$\underline{17.6}$	16/1.9	46.8	75/23.7	3243.5	56/11.1	1515.5
N+TPCG(0.1)	13/2.9	26.2	11/2.8	$\underline{38.5}$	46/31.0	2509.1	82/7.0	1466.1
N+TPCG(10^{-2})	15/4.9	34.6	15/4.8	71.1	21/64.5	2372.9	18/14.5	682.3
N+TPCG(10^{-3})	15/6.9	37.8	15/6.8	89.8	12/97.6	2042.8	12/29.8	801.3
N+TPCG(10^{-4})	15/8.9	41.7	15/8.7	111.0	19/166.8	5344.5	19/51.3	2162.2
N+TPCG(10^{-5})	15/10.9	44.5	15/10.7	125.3	20/217.8	7320.4	20/67.0	2931.4
N+TPCG(10^{-6})	15/12.9	47.8	15/13.5	157.9	20/264.1	8854.3	20/81.8	3591.1

Table 4.2

Comparison of N+TPCG algorithms for the denoising and deblurring problems. The best minimization time (s) for each algorithm is underlined.

corresponds to a very approximate solving of (1.12) compared to that envisaged in [14]. The computation time is then reduced by a factor two to ten, which represents a substantial gain.

Our conclusion is that the HQ/N+TPCG family should not be viewed only as approximate versions of HQ or Newton algorithms, but as algorithms on their own, with specific convergence properties and tuning parameters. In particular, the present study indicates that they are worth to be considered to solve large-scale problems in the field of image restoration.

REFERENCES

[1] M. Allain, J. Idier, and Y. Goussard, *On global and local convergence of half-quadratic algorithms*, IEEE Transactions on Image Processing, 15 (2006), pp. 1130–1142.

[2] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, 2nd ed., 1999.

[3] E. G. Birgin, N. Krejić, and J. M. Martínez, *Globally convergent inexact quasi-Newton methods for solving nonlinear systems*, Numerical algorithms, 32 (2003), pp. 249–260.

[4] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud, *Deterministic edge-preserving regularization in computed imaging*, IEEE Transactions on Image Processing, 6 (1997), pp. 298–311.

[5] R. Dembo, S. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM Journal of Numerical Analysis, 19 (1982), pp. 400–408.

[6] S. Eisenstat and H. Walker, *Globally convergent inexact Newton methods*, SIAM Journal of Optimization, 4 (1994), pp. 393–422.

[7] D. Geman and G. Reynolds, *Constrained restoration and the recovery of discontinuities*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 14 (1992), pp. 367–383.

[8] D. Geman and C. Yang, *Nonlinear image recovery with half-quadratic regularization*, IEEE Transactions on Image Processing, 4 (1995), pp. 932–946.

[9] G. H. Golub and C. F. Van Loan, *Matrix computations*, The Johns Hopkins University Press, Baltimore, 3rd ed., 1996.

[10] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear system*, Journal of Research of the National Bureau of Standards, 49 (1952), pp. 409–436.

[11] C. Labat and J. Idier, *Convergence of conjugate gradient methods with a closed-form stepsize formula*, Journal of Optimisation Theory and Applications, 135 (2007).

[12] S. G. Nash, *A survey of truncated-Newton methods*, Journal of Computational and Applied Mathematics, 124 (2000), pp. 45–59.

[13] M. K. Ng, R. H. Chan, and W.-C. Tang, *A fast algorithm for deblurring models with Neumann boundary conditions*, SIAM Journal of Scientific Computing, 21 (1999), pp. 851–866.

[14] M. Nikolova and M. K. Ng, *Fast image reconstruction algorithms combining half-quadratic regularization and preconditioning*, in Proceedings of the International Conference on Image Processing, Thessaloniki, Greece, Oct. 2001, pp. 277–280.

[15] ———, *Analysis of half-quadratic minimization methods for signal and image recovery*, SIAM Journal of Scientific Computing, 27 (2005), pp. 937–966.

[16] J. Sun and J. Zhang, *Global convergence of conjugate gradient methods without line search*, Annals of Operations Research, 103 (2001), pp. 161–173.

[17] G. Zhou and L. Qi, *On the convergence of an inexact newton-type method*, Operations Research Letters, 34 (2006), pp. 647–652.