



HAL
open science

Sparsity Constrained Image Restoration: An Approach Using the Newton Projection Method

Germana Landi

► **To cite this version:**

Germana Landi. Sparsity Constrained Image Restoration: An Approach Using the Newton Projection Method. 27th IFIP Conference on System Modeling and Optimization (CSMO), Jun 2015, Sophia Antipolis, France. pp.341-350, 10.1007/978-3-319-55795-3_32 . hal-01626912

HAL Id: hal-01626912

<https://inria.hal.science/hal-01626912v1>

Submitted on 31 Oct 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Sparsity constrained image restoration: an approach using the Newton Projection method

Germana Landi

Department of Mathematics, University of Bologna
germana.landi@unibo.it

Abstract. Image restoration under sparsity constraints has received increased attention in recent years. This problem can be formulated as a nondifferentiable convex optimization problem whose solution is challenging. In this work, the non-differentiability of the objective is addressed by reformulating the image restoration problem as a nonnegatively constrained quadratic program which is then solved by a specialized Newton projection method where the search direction computation only requires matrix-vector operations. A comparative study with state-of-the-art methods is performed in order to illustrate the efficiency and effectiveness of the proposed approach.

Keywords: ℓ_1 -norm based regularization, Newton projection method, image restoration, inverse problems

1 Introduction

This work is concerned with the general problem of image restoration under sparsity constraints formulated as

$$\min_{\mathbf{x}} \phi(\mathbf{x}) = \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{x}\|_1 \quad (1)$$

where \mathbf{A} is a $m \times n$ real matrix (usually $m \leq n$), $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$ and λ is a positive parameter. (Throughout the paper, $\|\cdot\|$ will denote the Euclidean norm). In some image restoration applications, $\mathbf{A} = \mathbf{KW}$ where $\mathbf{K} \in \mathbb{R}^{m \times n}$ is a discretized linear operator and $\mathbf{W} \in \mathbb{R}^{n \times n}$ is a transformation matrix from a domain where the image is *a priori* known to have a sparse representation. The variable \mathbf{x} contains the coefficients of the unknown image and the data \mathbf{b} is the measurements vector which is assumed to be affected by Gaussian white noise intrinsic to the detection process. The formulation (1) is usually referred to as *synthesis formulation* since it is based on the synthesis equation of the unknown image from its coefficients \mathbf{x} .

The penalization of the ℓ_1 -norm of the coefficients vector \mathbf{x} in (1) simultaneously favors sparsity and avoids overfitting. For this reason, sparsity constrained image restoration has received considerable attention in the recent literature and has been successfully used in various areas. The efficient solution of problem (1)

is a critical issue since the nondifferentiability of the ℓ_1 -norm makes standard unconstrained optimization methods unusable. Among the current state-of-the-art methods there are gradient descent-type methods as TwIST [5], SparSA [14], FISTA [2] and NESTA [3]. GPSR [6] is a gradient-projection algorithm for the equivalent convex quadratic program obtained by splitting the variable \mathbf{x} in its positive and negative parts. Fixed-point continuation methods [9], as well as methods based on Bregman iterations [7] and variable splitting, as SALSA [1], have also been recently proposed. In [12], the classic Newton projection method is used to solve the bound-constrained quadratic program formulation of (1) obtained by splitting \mathbf{x} . A Modified Newton projection (MNP) method has been recently proposed in [11] for the *analysis formulation* of the ℓ_1 -regularized least squares problem where \mathbf{W} is the identity matrix and \mathbf{x} represents the image itself. The MNP method uses a fair regularized approximation to the Hessian matrix so that products of its inverse and vectors can be computed at low computational cost. As a result, the only operations required for the search direction computation are matrix-vector products.

The main contribution of this work is to extend the MNP method of [11], developed for the case $\mathbf{W} = \mathbf{I}_n$, to the synthesis formulation of problem (1) where $\mathbf{W} \neq \mathbf{I}_n$. In the proposed approach, problem (1) is firstly formulated as a nonnegatively constrained quadratic programming problem by splitting the variable \mathbf{x} into the positive and negative parts. Then, the quadratic program is solved by a special purpose MNP method where a fair regularized approximation to the Hessian matrix is proposed so that products of its inverse and vectors can be computed at low computational cost. As a result, the search direction can be efficiently obtained. The convergence of the proposed MNP method is analyzed. Even if the size of the problem is doubled, the low computational cost per iteration and less iterative steps make MNP quite efficient. The performance of MNP is evaluated on several image restoration problems and is compared with that of some state-of-the-art methods. The results of the comparative study show that MNP is competitive and in some cases is outperforming some state-of-the-art methods in terms of computational complexity and achieved accuracy.

The rest of the paper can be outlined as follows. In section 2, the quadratic program formulation of (1) is derived. The MNP method is presented and its convergence is analyzed in section 3. In this section, the efficient computation of the search direction is also discussed. In section 4, the numerical results are presented. Conclusions are given in section 5.

2 Nonnegatively constrained quadratic program formulation

The proposed approach firstly needs to reformulate (1) as a nonnegatively constrained quadratic program (NCQP). The NCQP formulation is obtained by splitting the variable \mathbf{x} into its positive and negative parts [6], i.e

$$\mathbf{x} = \mathbf{u} - \mathbf{v}, \quad \mathbf{u} = \max(\mathbf{x}, 0), \quad \mathbf{v} = \max(-\mathbf{x}, 0).$$

Problem (1) can be written as the following NCQP:

$$\begin{aligned} \min_{(\mathbf{u}, \mathbf{v})} \mathcal{F}(\mathbf{u}, \mathbf{v}) &= \frac{1}{2} \|\mathbf{A}(\mathbf{u} - \mathbf{v}) - \mathbf{b}\|^2 + \lambda \mathbf{1}^H \mathbf{u} + \lambda \mathbf{1}^H \mathbf{v} \\ \text{s.t. } \mathbf{u} &\geq 0, \mathbf{v} \geq 0 \end{aligned} \quad (2)$$

where $\mathbf{1}$ denotes the n -dimensional column vector of ones. The gradient \mathbf{g} and Hessian \mathbf{H} of $\mathcal{F}(\mathbf{u}, \mathbf{v})$ are respectively defined by

$$\mathbf{g} = \begin{bmatrix} \mathbf{A}^H \mathbf{A}(\mathbf{u} - \mathbf{v}) - \mathbf{A}^H \mathbf{b} + \lambda \mathbf{1} \\ -\mathbf{A}^H \mathbf{A}(\mathbf{u} - \mathbf{v}) + \mathbf{A}^H \mathbf{b} + \lambda \mathbf{1} \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \mathbf{A}^H \mathbf{A} & -\mathbf{A}^H \mathbf{A} \\ -\mathbf{A}^H \mathbf{A} & \mathbf{A}^H \mathbf{A} \end{bmatrix}. \quad (3)$$

We remark that the computation of the objective function and its gradient values requires only one multiplication by \mathbf{A} and one by \mathbf{A}^H , nevertheless the double of the problem size. Since \mathbf{H} is positive semidefinite, we propose to approximate it with the positive definite matrix \mathbf{H}_τ :

$$\mathbf{H}_\tau = \begin{bmatrix} \mathbf{A}^H \mathbf{A} + \tau \mathbf{I}_n & -\mathbf{A}^H \mathbf{A} \\ -\mathbf{A}^H \mathbf{A} & \mathbf{A}^H \mathbf{A} + \tau \mathbf{I}_n \end{bmatrix} \quad (4)$$

where τ is a positive parameter and \mathbf{I} is the identity matrix of size n .

Proposition 21 *Let $\sigma_1, \sigma_2, \dots, \sigma_n$ be the nonnegative eigenvalues of \mathbf{A} in non-increasing order:*

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_m \geq \sigma_{m+1} = \dots = \sigma_n = 0. \quad (5)$$

Then, \mathbf{H}_τ is a positive definite matrix whose eigenvalues are

$$2\sigma_1 + \tau, 2\sigma_2 + \tau, \dots, 2\sigma_n + \tau, \tau, \dots, \tau. \quad (6)$$

The proof is immediate since the spectrum of \mathbf{H}_τ is the union of the spectra of $\mathbf{A}^H \mathbf{A} + \tau \mathbf{I} + \mathbf{A}^H \mathbf{A}$ and $\mathbf{A}^H \mathbf{A} + \tau \mathbf{I} - \mathbf{A}^H \mathbf{A}$.

The following proposition shows that an explicit formula for the inverse of \mathbf{H}_τ can be derived.

Proposition 22 *The inverse of the matrix \mathbf{H}_τ is the matrix \mathbf{M}_τ defined as*

$$\mathbf{M}_\tau = \frac{1}{\tau} \mathbf{M}_1 \mathbf{M}_2 \quad (7)$$

where

$$\mathbf{M}_1 = \begin{bmatrix} \mathbf{A}^H \mathbf{A} + \tau \mathbf{I} & \mathbf{A}^H \mathbf{A} \\ \mathbf{A}^H \mathbf{A} & \mathbf{A}^H \mathbf{A} + \tau \mathbf{I} \end{bmatrix}, \quad \mathbf{M}_2 = \begin{bmatrix} (2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} & \mathbf{0} \\ \mathbf{0} & (2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} \end{bmatrix}.$$

Proof. We have

$$\begin{aligned} \frac{1}{\tau} \mathbf{H}_\tau \mathbf{M}_1 &= \frac{1}{\tau} \begin{bmatrix} (\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^2 - (\mathbf{A}^H \mathbf{A})^2 & \mathbf{0} \\ \mathbf{0} & (\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^2 - (\mathbf{A}^H \mathbf{A})^2 \end{bmatrix} \\ &= \frac{1}{\tau} \begin{bmatrix} 2\tau \mathbf{A}^H \mathbf{A} + \tau^2 \mathbf{I} & \mathbf{0} \\ \mathbf{0} & 2\tau \mathbf{A}^H \mathbf{A} + \tau^2 \mathbf{I} \end{bmatrix} = \mathbf{M}_2^{-1}. \end{aligned} \quad (8)$$

Similarly, we can prove that

$$\frac{1}{\tau} \mathbf{M}_1 \mathbf{H}_\tau = \mathbf{M}_2^{-1}. \quad (9)$$

We now show that $\mathbf{M}_1 \mathbf{M}_2 = \mathbf{M}_2 \mathbf{M}_1$. We have

$$\begin{aligned} \mathbf{M}_1 \mathbf{M}_2 &= \begin{bmatrix} (\mathbf{A}^H \mathbf{A})(2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} & (\mathbf{A}^H \mathbf{A})(2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} \\ (\mathbf{A}^H \mathbf{A})(2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} & (\mathbf{A}^H \mathbf{A})(2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} \end{bmatrix} \\ &\quad + \begin{bmatrix} \tau(2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} & \mathbf{0} \\ \mathbf{0} & \tau(2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} \end{bmatrix} \\ \mathbf{M}_2 \mathbf{M}_1 &= \begin{bmatrix} (2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} (\mathbf{A}^H \mathbf{A}) & (2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} (\mathbf{A}^H \mathbf{A}) \\ (2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} (\mathbf{A}^H \mathbf{A}) & (2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} (\mathbf{A}^H \mathbf{A}) \end{bmatrix} \\ &\quad + \begin{bmatrix} \tau(2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} & \mathbf{0} \\ \mathbf{0} & \tau(2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} \end{bmatrix}. \end{aligned}$$

After some simple algebra, it can be proved that

$$(\mathbf{A}^H \mathbf{A})(2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} = (2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} (\mathbf{A}^H \mathbf{A}).$$

and thus

$$\mathbf{M}_1 \mathbf{M}_2 = \mathbf{M}_2 \mathbf{M}_1. \quad (10)$$

From (8), (9) and (10), it follows

$$\begin{aligned} \mathbf{H}_\tau \mathbf{M}_\tau &= \frac{1}{\tau} \mathbf{H}_\tau \mathbf{M}_1 \mathbf{M}_2 = \mathbf{M}_2^{-1} \mathbf{M}_2 = \mathbf{I}_{2n} \\ \mathbf{M}_\tau \mathbf{H}_\tau &= \frac{1}{\tau} \mathbf{M}_1 \mathbf{M}_2 \mathbf{H}_\tau = \frac{1}{\tau} \mathbf{M}_2 \mathbf{M}_1 \mathbf{H}_\tau = \mathbf{M}_2^{-1} \mathbf{M}_2^{-1} = \mathbf{I}_{2n}. \end{aligned}$$

3 The Modified Newton projection method

The algorithm. The Newton projection method [4] for problem (2) can be written as

$$\begin{bmatrix} \mathbf{u}^{(k+1)} \\ \mathbf{v}^{(k+1)} \end{bmatrix} = \left[\begin{bmatrix} \mathbf{u}^{(k)} \\ \mathbf{v}^{(k)} \end{bmatrix} - \alpha^{(k)} \mathbf{p}^{(k)} \right]^+, \quad \mathbf{p}^{(k)} = \mathbf{S}^{(k)} \mathbf{g}^{(k)}, \quad \mathbf{g}^{(k)} = \begin{bmatrix} \mathbf{g}_\mathbf{u}^{(k)} \\ \mathbf{g}_\mathbf{v}^{(k)} \end{bmatrix} \quad (11)$$

where $[\cdot]^+$ denotes the projection on the positive orthant, $\mathbf{g}_\mathbf{u}^{(k)}$ and $\mathbf{g}_\mathbf{v}^{(k)}$ respectively indicate the partial derivatives of \mathcal{F} with respect to \mathbf{u} and \mathbf{v} at the current iterate. The scaling matrix $\mathbf{S}^{(k)}$ is a partially diagonal matrix with respect to the index set $\mathcal{A}^{(k)}$ defined as

$$\begin{aligned} \mathcal{A}^{(k)} &= \left\{ i \mid 0 \leq y_i^{(k)} \leq \varepsilon^{(k)} \text{ and } g_i^{(k)} > 0 \right\} \\ \mathbf{y}^{(k)} &= \begin{bmatrix} \mathbf{u}^{(k)} \\ \mathbf{v}^{(k)} \end{bmatrix}, \quad \varepsilon^{(k)} = \min\{\varepsilon, w^{(k)}\}, \quad w^{(k)} = \|\mathbf{y}^{(k)} - [\mathbf{y}^{(k)} - \mathbf{g}^{(k)}]^+\| \end{aligned}$$

and ε is a small positive parameter.

The step-length $\alpha^{(k)}$ is computed with the Armijo rule along the projection arc [4]. Let $\mathbf{E}^{(k)}$ and $\mathbf{F}^{(k)}$ be the diagonal matrices [13] such that

$$\{\mathbf{E}^{(k)}\}_{ii} = \begin{cases} 1, & i \notin \mathcal{A}^{(k)}; \\ 0, & i \in \mathcal{A}^{(k)}; \end{cases}, \quad \mathbf{F}^{(k)} = \mathbf{I}_{2n} - \mathbf{E}^{(k)}.$$

In MNP, we propose to define the scaling matrix $\mathbf{S}^{(k)}$ as

$$\mathbf{S}^{(k)} = \mathbf{E}^{(k)}\mathbf{M}_\tau\mathbf{E}^{(k)} + \mathbf{F}^{(k)}. \quad (12)$$

Therefore, the complexity of computation of the search direction $\mathbf{p}^{(k)} = \mathbf{S}^{(k)}\mathbf{g}^{(k)}$ is mainly due to one multiplication of the inverse Hessian approximation \mathbf{M}_τ with a vector since matrix-vector products involving $\mathbf{E}^{(k)}$ and $\mathbf{F}^{(k)}$ only extracts some components of the vector and do not need to be explicitly performed. We remark that, in the Newton projection method proposed in [4], the scaling matrix is $\mathbf{S}^{(k)} = (\mathbf{E}^{(k)}\mathbf{H}_\tau\mathbf{E}^{(k)} + \mathbf{F}^{(k)})^{-1}$ which requires to extract a submatrix of \mathbf{H}_τ and then to invert it.

Convergence analysis. As proved in [4], the convergence of Newton-type projection methods can be proved under the general following assumptions which basically require the scaling matrices $\mathbf{S}^{(k)}$ to be positive definite matrices with uniformly bounded eigenvalues.

A1 The gradient \mathbf{g} is Lipschitz continuous on each bounded set of \mathbb{R}^{2n} .

A2 There exist positive scalars c_1 and c_2 such that

$$c_1\|\mathbf{y}\|^2 \leq \mathbf{y}^H\mathbf{S}^{(k)}\mathbf{y} \leq c_2\|\mathbf{y}\|^2, \quad \forall \mathbf{y} \in \mathbb{R}^{2n}, \quad k = 0, 1, \dots$$

The key convergence result is provided in Proposition 2 of [4] which is restated here for the sake of completeness.

Proposition 31 [4, Proposition 2] Let $\{\{\mathbf{u}^{(k)}, \mathbf{v}^{(k)}\}\}$ be a sequence generated by iteration (11) where $\mathbf{S}^{(k)}$ is a positive definite symmetric matrix which is diagonal with respect to $\mathcal{A}^{(k)}$ and α^k is computed by the Armijo rule along the projection arc. Under assumptions A1 and A2 above, every limit point of a sequence $\{\{\mathbf{u}^{(k)}, \mathbf{v}^{(k)}\}\}$ is a critical point with respect to problem (2).

Since the objective \mathcal{F} of (2) is twice continuously differentiable, it satisfies assumption A1. From propositions 21 and 22, it follows that \mathbf{M}_τ is a symmetric positive definite matrix and hence, the scaling matrix $\mathbf{S}^{(k)}$ defined by (12) is a positive definite symmetric matrix which is diagonal with respect to $\mathcal{A}^{(k)}$. The global convergence of the MNP method is therefore guaranteed provided $\mathbf{S}^{(k)}$ verifies assumption A2.

Proposition 32 Let $\mathbf{S}^{(k)}$ be the scaling matrix defined as $\mathbf{S}^{(k)} = \mathbf{E}^{(k)}\mathbf{H}_\tau\mathbf{E}^{(k)} + \mathbf{F}^{(k)}$. Then, there exist two positive scalars c_1 and c_2 such that

$$c_1\|\mathbf{y}\|^2 \leq \mathbf{y}^H\mathbf{S}^{(k)}\mathbf{y} \leq c_2\|\mathbf{y}\|^2, \quad \forall \mathbf{y} \in \mathbb{R}^{2n}, \quad k = 0, 1, \dots$$

Proof. Proposition 21 implies that the largest and smallest eigenvalue of \mathbf{M}_τ are respectively $1/\tau$ and $1/(2\sigma_1 + \tau)$, therefore

$$\frac{1}{2\sigma_1 + \tau} \|\mathbf{y}\|^2 \leq \mathbf{y}^H \mathbf{M}_\tau \mathbf{y} \leq \frac{1}{\tau} \|\mathbf{y}\|^2, \quad \forall \mathbf{y} \in \mathbb{R}^{2n}. \quad (13)$$

We have $\mathbf{y}^H \mathbf{S}^{(k)} \mathbf{y} = (\mathbf{E}^{(k)} \mathbf{y})^H \mathbf{M}_\tau (\mathbf{E}^{(k)} \mathbf{y}) + \mathbf{y}^H \mathbf{F}^{(k)} \mathbf{y}$. From (13) it follows that

$$\frac{\|\mathbf{E}^{(k)} \mathbf{y}\|^2}{2\sigma_1 + \tau} + \mathbf{y}^H \mathbf{F}^{(k)} \mathbf{y} \leq (\mathbf{E}^{(k)} \mathbf{y})^H \mathbf{M}_\tau (\mathbf{E}^{(k)} \mathbf{y}) + \mathbf{y}^H \mathbf{F}^{(k)} \mathbf{y} \leq \frac{\|\mathbf{E}^{(k)} \mathbf{y}\|^2}{\tau} + \mathbf{y}^H \mathbf{F}^{(k)} \mathbf{y}.$$

Moreover we have $\mathbf{y}^H \mathbf{F}^{(k)} \mathbf{y} = \sum_{i \in \mathcal{A}^{(k)}} y_i^2$, $\|\mathbf{E}^{(k)} \mathbf{y}\|^2 = \sum_{i \notin \mathcal{A}^{(k)}} y_i^2$; hence:

$$\frac{1}{2\sigma_1 + \tau} \sum_{i \notin \mathcal{A}^{(k)}} y_i^2 + \sum_{i \in \mathcal{A}^{(k)}} y_i^2 \leq \mathbf{y}^H \mathbf{S}^{(k)} \mathbf{y} \leq \frac{1}{\tau} \sum_{i \notin \mathcal{A}^{(k)}} y_i^2 + \sum_{i \in \mathcal{A}^{(k)}} y_i^2$$

and

$$\min\left\{\frac{1}{2\sigma_1 + \tau}, 1\right\} \|\mathbf{y}\|^2 \leq \mathbf{y}^H \mathbf{S}^{(k)} \mathbf{y} \leq \max\left\{\frac{1}{\tau}, 1\right\} \|\mathbf{y}\|^2.$$

The thesis follows by setting $c_1 = \min\{\frac{1}{2\sigma_1 + \tau}, 1\}$ and $c_2 = \max\{\frac{1}{\tau}, 1\}$.

Computing the search direction. We suppose that \mathbf{K} is the matrix representation of a spatially invariant convolution operator with periodic boundary conditions so that \mathbf{K} is a block circulant with circulant blocks (BCCB) matrix and matrix-vector products can be efficiently performed via the FFT. Moreover, we assume that the columns of \mathbf{W} form an orthogonal basis for which fast sparsifying algorithms exist, such as a wavelet basis, for example. Under these assumptions, \mathbf{A} is a full and dense matrix but the computational cost of matrix-vector operations with \mathbf{A} and \mathbf{A}^H is relatively cheap. As shown by (12), the computation of the search direction $\mathbf{p}^{(k)} = \mathbf{S}^{(k)} \mathbf{g}^{(k)}$ requires the multiplication of a vector by \mathbf{M}_τ . Let $[\mathbf{z}, \mathbf{w}] \in \mathbb{R}^{2n}$ be a given vector, then it immediately follows that

$$\mathbf{M}_\tau \begin{bmatrix} \mathbf{z} \\ \mathbf{w} \end{bmatrix} = \frac{1}{\tau} \begin{bmatrix} \mathbf{A}^H \mathbf{A} (2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} (\mathbf{z} + \mathbf{w}) + \tau (2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} \mathbf{z} \\ \mathbf{A}^H \mathbf{A} (2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} (\mathbf{z} + \mathbf{w}) + \tau (2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} \mathbf{w} \end{bmatrix}. \quad (14)$$

Formula (14) needs the inversion of $2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I}$. Our experimental results indicate that the search direction can be efficiently and effectively computed as follows. Using the Sherman-Morrison-Woodbury formula, we obtain

$$(2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} = \frac{1}{\tau} \left(\mathbf{I} - \mathbf{W}^H \mathbf{K}^H (\mathbf{K} \mathbf{K}^H + \frac{\tau}{2})^{-1} \mathbf{K} \mathbf{W} \right) \quad (15)$$

$$\mathbf{A}^H \mathbf{A} (2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} = \frac{1}{\tau} \left(\mathbf{W}^H (\mathbf{K}^H \mathbf{K} - \mathbf{K}^H \mathbf{K} \mathbf{K}^H (\mathbf{K} \mathbf{K}^H + \frac{\tau}{2})^{-1} \mathbf{K}) \mathbf{W} \right). \quad (16)$$

Substituting (15) and (16) in (14), we obtain

$$\begin{aligned} \mathbf{A}^H \mathbf{A} (2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} (\mathbf{z} + \mathbf{w}) + \tau (2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} \mathbf{z} = \\ \frac{1}{\tau} \mathbf{W}^H (\mathbf{K}^H \mathbf{K} - \mathbf{K}^H \mathbf{K} \mathbf{K}^H (\mathbf{K} \mathbf{K}^H + \frac{\tau}{2})^{-1} \mathbf{K}) (\mathbf{W} \mathbf{z} + \mathbf{W} \mathbf{w}) \\ - \mathbf{W}^H \mathbf{K}^H (\mathbf{K} \mathbf{K}^H + \frac{\tau}{2})^{-1} \mathbf{K} \mathbf{W} \mathbf{z} + \mathbf{z} \end{aligned} \quad (17)$$

$$\begin{aligned} \mathbf{A}^H \mathbf{A} (2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} (\mathbf{z} + \mathbf{w}) + \tau (2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} \mathbf{w} = \\ \frac{1}{\tau} \mathbf{W}^H (\mathbf{K}^H \mathbf{K} - \mathbf{K}^H \mathbf{K} \mathbf{K}^H (\mathbf{K} \mathbf{K}^H + \frac{\tau}{2})^{-1} \mathbf{K}) (\mathbf{W} \mathbf{z} + \mathbf{W} \mathbf{w}) \\ - \mathbf{W}^H \mathbf{K}^H (\mathbf{K} \mathbf{K}^H + \frac{\tau}{2})^{-1} \mathbf{K} \mathbf{W} \mathbf{w} + \mathbf{w}. \end{aligned} \quad (18)$$

Since \mathbf{K} is BCCB, it is diagonalized by the Discrete Fourier Transform (DFT), i.e. $\mathbf{K} = \mathbf{U}^H \mathbf{D} \mathbf{U}$ where \mathbf{U} denotes the unitary matrix representing the DFT and \mathbf{D} is a diagonal matrix. Thus, we have:

$$\mathbf{K}^H (\mathbf{K} \mathbf{K}^H + \frac{\tau}{2})^{-1} \mathbf{K} = \mathbf{U}^H \left(\frac{|\mathbf{D}|^2}{|\mathbf{D}|^2 + \frac{\tau}{2}} \right) \mathbf{U} \quad (19)$$

$$\mathbf{K}^H \mathbf{K} - \mathbf{K}^H \mathbf{K} \mathbf{K}^H (\mathbf{K} \mathbf{K}^H + \frac{\tau}{2})^{-1} \mathbf{K} = \mathbf{U}^H \left(|\mathbf{D}|^2 - \frac{|\mathbf{D}|^4}{|\mathbf{D}|^2 + \frac{\tau}{2}} \right) \mathbf{U}. \quad (20)$$

Substituting (19) and (20) in (17) and (18), we obtain

$$\begin{aligned} \mathbf{A}^H \mathbf{A} (2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} (\mathbf{z} + \mathbf{w}) + \tau (2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} \mathbf{z} = \\ \frac{1}{\tau} \mathbf{W}^H \mathbf{U}^H \left((|\mathbf{D}|^2 - \frac{|\mathbf{D}|^4}{|\mathbf{D}|^2 + \frac{\tau}{2}}) (\mathbf{U} \mathbf{W} \mathbf{z} + \mathbf{U} \mathbf{W} \mathbf{w}) - \left(\frac{|\mathbf{D}|^2}{|\mathbf{D}|^2 + \frac{\tau}{2}} \right) \mathbf{U} \mathbf{W} \mathbf{z} \right) + \mathbf{z} \end{aligned} \quad (21)$$

$$\begin{aligned} \mathbf{A}^H \mathbf{A} (2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} (\mathbf{z} + \mathbf{w}) + \tau (2\mathbf{A}^H \mathbf{A} + \tau \mathbf{I})^{-1} \mathbf{w} = \\ \frac{1}{\tau} \mathbf{W}^H \mathbf{F}^H \left((|\mathbf{D}|^2 - \frac{|\mathbf{D}|^4}{|\mathbf{D}|^2 + \frac{\tau}{2}}) (\mathbf{U} \mathbf{W} \mathbf{z} + \mathbf{U} \mathbf{W} \mathbf{w}) - \left(\frac{|\mathbf{D}|^2}{|\mathbf{D}|^2 + \frac{\tau}{2}} \right) \mathbf{U} \mathbf{W} \mathbf{w} \right) + \mathbf{w}. \end{aligned} \quad (22)$$

Equations (14), (21) and (22) show that, at each iteration, the computation of the search direction $\mathbf{p}^{(k)}$ requires two products by \mathbf{W} , two products by \mathbf{W}^H , two products by \mathbf{U} and two products by \mathbf{U}^H . The last products can be performed efficiently by using the FFT algorithm.



Fig. 1: Top line: exact image (left), Gaussian blurred image (middle) and MNP reconstruction (right). Bottom line: out-of-focus blurred image (left) and MNP reconstruction (right). The noise level is $NL=7.5 \cdot 10^{-3}$.

4 Numerical results

In this section, we present the numerical results of some image restoration test problems. The numerical experiments aim at illustrating the performance of MNP compared with some state-of-the-art methods as SALSA [1], CGIST [8], the nonmonotonic version of GPSR [6], and the Split Bregman method [7]. Even if SALSA has been shown to outperform GPSR [1], we consider GPSR in our comparative study since it solves, as MNP, the quadratic program (2). The Matlab source code of the considered methods, made publicly available by the authors, has been used in the numerical experiments. The numerical experiments have been executed in Matlab R2012a on a personal computer with an Intel Core i7-2600, 3.40 GHz processor.

The numerical experiments are based on the well-known *Barbara* image (figure 1), whose size is 512×512 and whose pixels have been scaled into the range between 0 and 1. In our experiments, the matrix \mathbf{W} represents an orthogonal Haar wavelet transform with four levels. For all the considered methods, the initial iterate $\mathbf{x}^{(0)}$ has been chosen as $\mathbf{x}^{(0)} = \mathbf{W}^H \mathbf{b}$; the regularization parameter λ has been heuristically chosen. In MNP, the parameter τ of the Hessian approximation has been fixed at $\tau = 100\lambda$. This value has been fixed after a wide experimentation and has been used in all the presented numerical experiments.

The methods iteration is terminated when the relative distance between two successive objective values becomes less than a tolerance tol_ϕ . A maximum number of 100 iterations has been allowed for each method.

In the first experiment, the *Barbara* image has been convolved with a Gaussian PSF with variance equal to 2, obtained with the code `psfGauss` from [10], and then, the blurred image has been corrupted by Gaussian noise with

noise level equal to $7.5 \cdot 10^{-3}$ and $1.5 \cdot 10^{-2}$. (The noise level NL is defined as $NL := \|\boldsymbol{\eta}\|/\|\mathbf{A}\mathbf{x}_{\text{original}}\|$ where $\mathbf{x}_{\text{original}}$ is the original image and $\boldsymbol{\eta}$ is the noise vector.) In the second experiment, the *Barbara* image has been corrupted by out-of-focus blur, obtained with the code `psfDefocus` from [10] and by Gaussian noise with noise level equal to $2.5 \cdot 10^{-3}$ and $7.5 \cdot 10^{-3}$. The degraded images and the MNP restorations are shown in figure 1 for $NL = 7.5 \cdot 10^{-3}$. Table 1 reports the Mean Squared Error (MSE) values, the objective function values and the CPU times in seconds obtained by using the stopping tolerance values $tol_{\phi} = 10^{-1}, 10^{-2}, 10^{-3}$. In figure 2, the MSE behavior and the decreasing of the objective function versus time (in seconds) are illustrated.

The reported numerical results indicate that MNP is competitive with the considered state-of-the-art and, in terms of MSE reduction, MNP reaches the minimum MSE value very early.

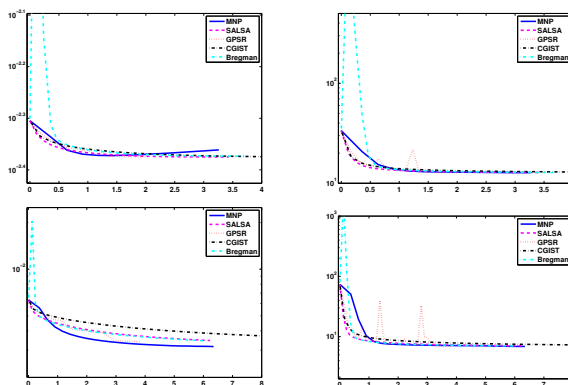


Fig. 2: MSE (left column) and objective function (right column) histories of MNP (blue solid line), SALSA (magenta dashed line), GPSR (red dotted line), CGIST (black dashdotted line) and Split Bregman (cyan dashdotted line) methods. Top line: Gaussian blur; bottom line: out-of-focus blur. The noise level is $NL=1.5 \cdot 10^{-2}$.

5 Conclusions

In this work, the MNP method has been proposed for sparsity constrained image restoration. In order to gain low computational complexity, the MNP method uses a fair approximation of the Hessian matrix so that the search direction can be computed efficiently by only using FFTs and fast sparsifying algorithms. The results of numerical experiments show that MNP may be competitive with some state-of-the-art methods both in terms of computational efficiency and accuracy.

References

1. M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo. Fast image recovery using variable splitting and constrained optimization. *IEEE Trans. Image Process.*,

tol_ϕ	Method	Gaussian blur						out-of-focus blur					
		NL=0.0075			NL=0.015			NL=0.0025			NL=0.0075		
		MSE	Obj	Time	MSE	Obj	Time	MSE	Obj	Time	MSE	Obj	Time
10^{-1}	MNP	$4.18 \cdot 10^{-3}$	7.33	1.15	$4.25 \cdot 10^{-3}$	13.26	1.17	$2.99 \cdot 10^{-3}$	3.04	2.68	$3.79 \cdot 10^{-3}$	7.73	1.42
	SALSA	$4.34 \cdot 10^{-3}$	7.86	0.47	$4.45 \cdot 10^{-3}$	15.05	0.36	$4.35 \cdot 10^{-3}$	3.96	0.47	$4.85 \cdot 10^{-3}$	9.78	0.47
	GPSR	$4.31 \cdot 10^{-3}$	7.49	0.89	$4.35 \cdot 10^{-3}$	13.76	0.89	$4.83 \cdot 10^{-3}$	5.41	0.66	$4.86 \cdot 10^{-3}$	9.90	0.64
	CGIST	$4.37 \cdot 10^{-3}$	8.09	0.72	$4.46 \cdot 10^{-3}$	14.92	0.56	$4.90 \cdot 10^{-3}$	5.33	0.90	$5.00 \cdot 10^{-3}$	10.36	0.72
	Breg.	$4.34 \cdot 10^{-3}$	7.50	0.72	$4.43 \cdot 10^{-3}$	14.32	0.75	$3.61 \cdot 10^{-3}$	3.03	0.48	$4.58 \cdot 10^{-3}$	8.81	0.72
10^{-2}	MNP	$4.17 \cdot 10^{-3}$	7.03	1.68	$4.25 \cdot 10^{-3}$	13.00	1.70	$2.47 \cdot 10^{-3}$	2.54	5.66	$3.45 \cdot 10^{-3}$	7.29	2.48
	SALSA	$4.21 \cdot 10^{-3}$	7.04	1.19	$4.31 \cdot 10^{-3}$	13.55	0.92	$3.25 \cdot 10^{-3}$	2.78	2.07	$4.20 \cdot 10^{-3}$	7.98	1.62
	GPSR	$4.22 \cdot 10^{-3}$	6.84	1.62	$4.34 \cdot 10^{-3}$	13.67	1.00	$3.94 \cdot 10^{-3}$	3.16	2.39	$4.03 \cdot 10^{-3}$	7.74	1.59
	CGIST	$4.24 \cdot 10^{-3}$	7.19	1.84	$4.32 \cdot 10^{-3}$	13.61	1.51	$3.87 \cdot 10^{-3}$	3.28	4.82	$4.34 \cdot 10^{-3}$	8.22	2.82
	Breg.	$4.23 \cdot 10^{-3}$	6.98	1.31	$4.33 \cdot 10^{-3}$	13.49	1.22	$2.85 \cdot 10^{-3}$	2.58	1.33	$4.15 \cdot 10^{-3}$	7.85	1.56
10^{-3}	MNP	$4.21 \cdot 10^{-3}$	6.54	3.42	$4.35 \cdot 10^{-3}$	12.77	3.26	$2.34 \cdot 10^{-3}$	2.49	8.92	$3.16 \cdot 10^{-3}$	6.88	6.38
	SALSA	$4.14 \cdot 10^{-3}$	6.60	4.09	$4.22 \cdot 10^{-3}$	12.79	3.46	$2.45 \cdot 10^{-3}$	2.46	6.74	$3.46 \cdot 10^{-3}$	7.05	6.27
	GPSR	$4.20 \cdot 10^{-3}$	6.84	1.90	$4.22 \cdot 10^{-3}$	124	2.04	$3.81 \cdot 10^{-3}$	2.57	5.19	$3.40 \cdot 10^{-3}$	6.94	4.24
	CGIST	$4.14 \cdot 10^{-3}$	6.65	7.13	$4.22 \cdot 10^{-3}$	12.82	5.71	$2.86 \cdot 10^{-3}$	2.59	18.81	$3.52 \cdot 10^{-3}$	7.11	11.65
	Breg.	$4.16 \cdot 10^{-3}$	6.57	4.38	$4.23 \cdot 10^{-3}$	12.79	3.76	$2.21 \cdot 10^{-3}$	2.41	4.15	$3.44 \cdot 10^{-3}$	7.02	6.01

Table 1: Restoring the noised and blurred *Barbara* image: numerical results.

- 19, 2345–2356 (2010)
2. A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1), 183–202 (2009)
3. S. Becker, J. Bobin, and E. J. Candès. NESTA: A fast and accurate first-order method for sparse recovery. *SIAM J. Imaging Sciences*, 4(1), 1–39 (2011)
4. D. Bertsekas. Projected Newton methods for optimization problem with simple constraints. *SIAM J. Control Optim.*, 20(2), 221–245 (1982)
5. J. Bioucas-Dias and M. Figueiredo. A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Trans. Image Process.*, 16(12), 2992173004 (2007)
6. M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE J. Sel. Top. Signal Process.*, 1, 586–597 (2007)
7. T. Goldstein and S. Osher. The split bregman method for l1 regularized problems. *SIAM J. Imag. Sci.*, 2(2), 323–343 (2009)
8. T. Goldstein and S. Setzer. High-order methods for basis pursuit. Technical Report CAM report 10-41, UCLA, (2010)
9. E. T. Hale, W. Yin, and Y. Zhang. Fixed-point continuation for l1-minimization: Methodology and convergence. *SIAM J. Optim.*, 19, 1107–1130 (2008)
10. P. C. Hansen, J. Nagy, and D. P. O’Leary. Deblurring images. *Matrices, Spectra and Filtering*. SIAM, (2006)
11. G. Landi. A Modified Newton Projection method for ℓ_1 -regularized least squares image deblurring. *J. Math. Imag. Vis.*, 51(1), 195–208 (2015)
12. M. Schmidt, D. Kim, and S. Sra. Optimization for Machine Learning, chapter Projected Newton-type Methods in Machine Learning. MIT Press (2011)
13. C. R. Vogel. Computational Methods for Inverse Problems. SIAM, Philadelphia, PA, USA (2002)
14. S. Wright, R. Nowak, and M. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Trans. Signal Process.*, 57(7), 2479172493 (2009)