

Image regularization by nonnegatively constrained conjugate gradient

P. Favati ¹ G. Lotti ² O. Menchi ³ F. Romani³

¹IIT - CNR Via G. Moruzzi 1, 56124 Pisa, Italy

²Dip. di Matematica, University of Parma, Parco Area delle
Scienze 53/A, 43124 Parma, Italy

³Dip. di Informatica, University of Pisa, Largo Pontecorvo 3,
56127 Pisa, Italy

8 January 2018

Abstract

In the image reconstruction context the nonnegativity of the computed solution is often required. Conjugate Gradient (CG), used as a reliable regularization tool, may give solutions with negative entries, particularly when large nearly zero plateaus are present. The active constraints set, detected by projection onto the nonnegative orthant, turns out to be largely incomplete leading to poor effects on the accuracy of the reconstructed image. In this paper an inner-outer method based on CG is proposed to compute nonnegative reconstructed images with a strategy which enlarges subsequently the active constraints set. This method appears to be especially suitable for the reconstruction of images having large nearly zero backgrounds. The numerical experimentation validates the effectiveness of the proposed method when compared to other strategies for nonnegative reconstruction.

Key words: Image Reconstruction, Conjugate Gradient, Nonnegativity Constraints.

1 Introduction

A Fredholm integral equation of the first kind

$$g(s) = \int \widehat{K}(s, t) f(t) dt \quad (1)$$

is often used for modeling the image formation process, where $f(t)$ and $g(s)$ represent a real object and its image, respectively. The kernel $\widehat{K}(s, t)$, called the *point spread function* (PSF) and assumed to be square integrable, represents

the imaging system and is responsible for the blurring of the image. In practical applications the *blurred image* $g(s)$ is not available, being replaced by a finite set \mathbf{g} of measured quantities, and is degraded by the noise which affects the process of image recording. Hence the problem of restoring $f(t)$ from \mathbf{g} is an ill-posed problem. The linear system obtained by the discretization of (1) inherits this feature, in the sense that the resulting matrix is severely ill-conditioned, and regularization methods must be used to solve it [2, 14, 24]. This kind of problem arises for example in the reconstruction of astronomical images taken by a telescope and of medical and microscopy images.

One of the main features of the problem is the nonnegativity of the functions involved in (1). When discretized, the equation leads to a linear problem whose solution is constrained to be nonnegative. Iterative methods applied as regularization techniques may give solutions with negative entries. A projection onto the nonnegative orthant may have poor effects on the accuracy of the reconstructed image. In this paper an inner-outer method based on CG is proposed to compute nonnegative reconstructed images with a strategy which enlarges subsequently the active constraints set. This method appears to be especially suitable for images having large zero backgrounds. For this type of problems, one naturally wonders if the zeros of the original image are correctly reconstructed. As a matter of fact, an algorithm can fail by putting to zero nonzero values of the original image (false positive) or by giving nonzero values in correspondence to zero values of the original image (false negative). This behavior will be analyzed using the well-known measures of Information Retrieval, namely the F_1 score [22], which takes into account both types of errors.

The outline of the paper is the following: first, in Section 2 the problem under consideration is introduced and in Section 3 some strategies for nonnegative regularization present in the literature are recalled. In Section 4 our proposed inner-outer algorithm is motivated and described. In Section 5 the results of a numerical experimentation which compares the performance of our algorithm with those of four chosen methods are presented and discussed.

Notation: Throughout the paper, $\|\mathbf{v}\|$ denotes the Euclidean norm of the vector \mathbf{v} , i.e. $\|\mathbf{v}\|^2 = \mathbf{v}^T \mathbf{v}$. The elementwise multiplication and division between two vectors are denoted by \odot and \oslash .

2 The problem

Let $\hat{\mathbf{b}} = A\hat{\mathbf{x}}$ be the discretized version of equation (1). In image reconstruction problems the N -vector $\hat{\mathbf{x}}$ stores columnwise the pixels of an $n \times n$ original object, with $N = n^2$, and $\hat{\mathbf{b}}$ analogously stores the blurred image. The imaging system is represented by a large not necessarily square matrix A , often severely ill-conditioned with singular values decaying to zero without significant gap to indicate numerical rank. The matrix A might not be explicitly available, as long as the products $A\mathbf{x}$ and $A^T \mathbf{x}$ are computable for any vector \mathbf{x} . A common special case is the one which occurs when the PSF is bandlimited space invariant,

i.e. invariant with respect to translations and with a bounded support, and A is a square 2-level Toeplitz matrix with a limited bandwidth. Moreover, it may happen that the image has sufficiently large zero background along the boundary, so that periodic boundary conditions can be safely imposed. In this case A becomes a 2-level circulant matrix and the matrix-by-vector product can be computed by low cost Fourier transforms. This is the structure we assume for our numerical examples, but the proposed algorithm can be applied equally well to general matrices A .

In practical problems vector $\widehat{\mathbf{b}}$ is not exactly known, because it is contaminated by measurement inaccuracies or discretization. Hence only a noisy image $\mathbf{b} = \widehat{\mathbf{b}} + \boldsymbol{\eta}$ is available, where the noise level is measured by

$$\eta = \frac{\|\mathbf{b} - \widehat{\mathbf{b}}\|}{\|\widehat{\mathbf{b}}\|}, \quad (2)$$

and in some cases can be roughly estimated. The system to be solved is thus

$$A\mathbf{x} = \mathbf{b}. \quad (3)$$

In this paper we consider the case where the entries of the noise vector $\boldsymbol{\eta}$ are normally distributed with zero mean and normalized in such a way that η ranges in a given interval.

The i th component of the vectors $\widehat{\mathbf{x}}$, $\widehat{\mathbf{b}}$ and \mathbf{b} represents respectively the light intensity or the radiation emitted by the i th pixel of the object, arriving at the i th pixel of the blurred image and recorded in the i th pixel of the noisy image. The component a_{ij} of matrix A measures the fraction of the light or of the rays emitted by the i th pixel of the object which arrives at the j th pixel of the image. Because of the ill-conditioning of A and of the presence of the noise, the solution $A^\dagger \mathbf{b}$ of (3), where A^\dagger is the Moore-Penrose generalized inverse, may be quite different from the original image $\widehat{\mathbf{x}}$.

All the quantities involved in the problem, i.e. A , $\widehat{\mathbf{x}}$, $\widehat{\mathbf{b}}$ and \mathbf{b} , are assumed componentwise nonnegative. Actually, when simulated test problems are considered for the experimentation, negative entries of \mathbf{b} could arise corresponding to very small nonnegative entries of $\widehat{\mathbf{b}}$. In this case the vector \mathbf{b} is further projected onto the nonnegative orthant. Anyway, it is reasonable to expect the approximation of $\widehat{\mathbf{x}}$ obtained by solving (3) to be nonnegative. The constrained *least squares* approximation to the solution $\widehat{\mathbf{x}}$ is given by

$$\mathbf{x}_{ls} = \underset{\mathbf{x} \geq \mathbf{0}}{\operatorname{argmin}} \phi(\mathbf{x}), \quad \text{where} \quad \phi(\mathbf{x}) = \frac{1}{2} \|\mathbf{b} - A\mathbf{x}\|^2. \quad (4)$$

The gradient of $\phi(\mathbf{x})$ is $\operatorname{grad}_x \phi(\mathbf{x}) = A^T A\mathbf{x} - A^T \mathbf{b} = -A^T \mathbf{r}$, where $\mathbf{r} = \mathbf{b} - A\mathbf{x}$ is the residual vector. The function $\phi(\mathbf{x})$ is convex and its minimum points are found by solving the system $\operatorname{grad}_x \phi(\mathbf{x}) = 0$, i.e. the so-called *normal equations*

$$A^T A\mathbf{x} = A^T \mathbf{b}. \quad (5)$$

Due to the large dimension of system (5) and to the presence of the noise $\boldsymbol{\eta}$, a regularization method must be employed, coupled with suitable strategies for enforcing nonnegativity. Iterative methods enjoying the semiconvergence property are often used. According to this property, an integer K exists for which the error attains a minimum. After the K th iteration, the computed vectors \boldsymbol{x}_k are progressively contaminated by the noise and move away from $\widehat{\boldsymbol{x}}$ toward $A^\dagger \boldsymbol{b}$ which can be largely different from $A^\dagger \widehat{\boldsymbol{b}}$. A good terminating procedure is hence needed to detect the correct index K where to stop the iteration.

In the following we assume that both $A\boldsymbol{x} \neq \mathbf{0}$ and $A^T \boldsymbol{x} \neq \mathbf{0}$ for any $\boldsymbol{x} \geq \mathbf{0}$ with $\boldsymbol{x} \neq \mathbf{0}$, and that $A\boldsymbol{e} > \mathbf{0}$ and $A^T \boldsymbol{e} > \mathbf{0}$, where \boldsymbol{e} is the vector of all ones (i.e. the sums by rows and columns of A are all nonzero).

3 Nonnegativity strategies

We recall here how classic and more recent nonnegativity techniques can be applied to iterative regularization methods.

Actually, the two classic methods “Expectation Maximization” (EM) [17, 21, 23] and “Iterative Space Reconstruction Algorithm” (ISRA) [5] enjoy the nonnegativity feature in a natural way when $\boldsymbol{x}_0 > \mathbf{0}$, in the sense that the zero components of \boldsymbol{x}_k produced at the k th iteration are kept at all the subsequent iterations. Unfortunately, they suffer from slow convergence rate, as shown in [9].

Descent methods have the form

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k, \quad (6)$$

where the direction \boldsymbol{p}_k satisfies $\boldsymbol{p}_k^T \boldsymbol{g}_k < 0$, with $\boldsymbol{g}_k = \text{grad}_{\boldsymbol{x}} \phi(\boldsymbol{x}_k)$, and the step size α_k is chosen in such a way that $\phi(\boldsymbol{x}_{k+1}) < \phi(\boldsymbol{x}_k)$. The iteration is carried out for $k = 0, \dots, K-1$ steps, until \boldsymbol{x}_K satisfies the stopping condition imposed by the regularization request. The choice $\alpha_k = -\boldsymbol{g}_k^T \boldsymbol{p}_k / \|\boldsymbol{A}\boldsymbol{p}_k\|^2$ satisfies the minimum problem

$$\phi(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k) = \min_{\alpha} \phi(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k),$$

but the vectors \boldsymbol{x}_k so computed may have negative components even if $\widehat{\boldsymbol{x}}$ is nonnegative.

A simple technique for imposing nonnegativity consists in choosing a reduced step size α_k to guarantee that $\boldsymbol{x}_{k+1} \geq \mathbf{0}$ if $\boldsymbol{x}_k \geq \mathbf{0}$. This technique, for example, is implemented in MRNSD, which is a modified version of the “Residual Norm Steepest Descent” method (RNSD) [1, 19]. The direction is $\boldsymbol{p}_k = X_k \boldsymbol{g}_k$, where $X_k = \text{diag}(\boldsymbol{x}_k)$ and the step size is

$$\bar{\alpha}_k = \alpha_k \text{ if } \boldsymbol{p}_k \geq \mathbf{0}, \quad \text{otherwise } \bar{\alpha}_k = \min \left\{ \alpha_k, \min_{j \in K_-} -(\boldsymbol{x}_k)_j / (\boldsymbol{p}_k)_j \right\}, \quad (7)$$

where K_- is the set of indices j such that $(\boldsymbol{p}_k)_j < 0$. The drawback of this technique is that small $\bar{\alpha}_k$ may occur, inducing slow convergence rate.

Another technique for imposing nonnegativity consists in performing at each iteration the projection of \mathbf{x}_k onto the nonnegative orthant by setting to zero its negative components.

The Conjugate Gradient method (CG) is known for its good convergence rate and its regularizing properties (see [13, 14, 20]). CG has the form (6) with directions \mathbf{p}_k that are $A^T A$ -conjugate. Unfortunately, if nearly zero plateaus are present in the original image $\hat{\mathbf{x}}$, CG may produce many negative components in the neighborhood of these areas, so a projection step should be performed. Theoretically, the projection should be applied at each iteration, but in the case of CG the directions \mathbf{p}_k would stop being $A^T A$ -conjugate, i.e. the method would lose its most important feature, which is the basis of its success. For this reason, when CG is used, the projection is only applied to the final iteration \mathbf{x}_K which satisfies the given stopping condition, but its projected vector $\tilde{\mathbf{x}}_K$ might not satisfy the stopping condition. In this event $\tilde{\mathbf{x}}_K$ could not be accepted as a regularized solution.

Thanks to its fast convergence rate, CG is used as a basis for regularizing nonnegative schemes. As an inner-outer scheme, [4] suggests to apply an iterative improvement to $\tilde{\mathbf{x}}_K$. The residual vector $\mathbf{r}_K = \mathbf{b} - A\tilde{\mathbf{x}}_K$ is computed and the system $A\mathbf{y} = -\mathbf{r}_K$ is solved using again CG. The regularized solution is used to update $\tilde{\mathbf{x}}_K$. If the vector so obtained still has negative components, it is projected and the improvement step is repeated. This first scheme, called in [4] “Projected Restarted Iteration” (here denoted PRI) is tested in [18] against a more refined scheme which restricts the update to the components corresponding to the nonactive constraints having positive Lagrangian multipliers. This second scheme is called “Active Set-type Method” (here denoted ASM).

An inner-outer scheme is also implemented in a new method described in [10] under the name NN-FCGLS. In the inner iterations, the step size $\bar{\alpha}_k$ is bounded as in (7) but the direction, scaled through the premultiplication by X_k , is computed by a Krylov method instead of a RNSD method. In particular, fixed an integer \hat{k} the chosen Krylov method is a CGLS-like method where the direction \mathbf{p}_k is obtained by a linear combination of at most \hat{k} previously computed \mathbf{p}_j with j varying in $\{\max\{0, k - \hat{k}\}, \dots, k - 1\}$. If a maximum number of iterations k_{\max} is assigned for the inner cycle, the choice $\hat{k} = k_{\max}$ corresponds to a full recursion, while a lower \hat{k} corresponds to a truncated recursion and with $\hat{k} = 1$, only the last computed vector \mathbf{p}_{k-1} is used. The outer cycle relies on suitable restarts, in order to avoid stagnation. In the following this scheme is denoted NNFCG.

Finally, we recall the “Scaled Gradient Projection” (SGP) methods [3]. The methods of this class use the projection for imposing the nonnegativity, but do not adopt an inner-outer scheme. They have the form (6) with directions \mathbf{p}_k and step size α_k which depend on a diagonal scaling matrix. Their performances have been analyzed in [6] and have shown good reconstruction accuracy and convergence rate. One of the most effective methods of this class generalizes ISRA. The direction is computed in the following way. Denoting $\mathbf{w}_k = \mathbf{x}_k \oslash$

$(A^T A \mathbf{x}_k)$, $\mathbf{s}_k = \mathbf{x}_k - \mathbf{x}_{k-1}$ and $\mathbf{z}_k = \mathbf{w}_k \odot (\mathbf{g}_k - \mathbf{g}_{k-1})$, the vector

$$\mathbf{u}_k = \mathbf{x}_k - \gamma \mathbf{w}_k \odot \mathbf{g}_k, \quad \text{where } \gamma = \mathbf{s}_k^T \mathbf{z}_k / \|\mathbf{z}_k\|^2,$$

is computed. Let $\bar{\mathbf{u}}_k$ be its projection. The direction and step size at the k th step in (6) are

$$\mathbf{p}_k = \bar{\mathbf{u}}_k - \mathbf{x}_k \quad \text{and} \quad \alpha_k = \max \left\{ \theta, \min \left\{ 1, -\mathbf{g}_k^T \mathbf{p}_k / \|A \mathbf{p}_k\|^2 \right\} \right\},$$

where $\theta > 0$ is a bound for α_k from below (typically $\theta = 10^{-3}$). The bounds on α_k guarantee that $\mathbf{x}_{k+1} \geq \mathbf{0}$ if $\mathbf{x}_k \geq \mathbf{0}$. Note that this method applies the projection to the direction \mathbf{u}_k and not to the approximated solution \mathbf{x}_k . In [6] this method is identified by the name **SGP-GcB**, but for simplicity we denote it here just **SGP**.

On the basis of the convergence rate, the four methods **PRI**, **ASM**, **NNFCG** and **SGP** will be considered in order to test the performance of our algorithm.

4 The algorithm we propose

In this section we describe an algorithm, called "Inner-Outer CG" (**IOCG**), based on restarted **CG**, coupled with a projection technique which exploits both the regularizing properties and the good convergence rate of **CG**.

IOCG and **ASM** share the same inner-outer scheme, with **CG** used for the inner process and an active set based technique for dealing with the constraints. The main difference is that **ASM** lets the constraints enter or leave the active set at each outer step, while with **IOCG** the constraints that belong to the active set are not allowed to leave it in the subsequent outer steps and the active set can only expand. The progressive enlargement of the active set ensures convergence. In a certain sense, this behavior reminds the **EM** and **ISRA** behaviors.

IOCG detects a sequence of progressively enlarged sets $\mathcal{A}^{(h)}$, $h = 0, 1, \dots, H$ of active constraints for the solution. On each set $\mathcal{A}^{(h)}$ the algorithm computes a regularized solution $\mathbf{y}^{(h)}$ with the components belonging to $\mathcal{A}^{(h)}$ fixed to zero, stopping when no further enlargement is required. Note that, unlike the standard iterative methods, the outer iteration has a natural termination.

Note: the projection $\tilde{\mathbf{y}}$ of a vector \mathbf{y} onto the nonnegative orthant can be formally implemented by means of a projection matrix P , i.e. a diagonal matrix of zeros and ones. In the pseudo-code of **IOCG** we will refer to a function **project** which constructs $\tilde{\mathbf{y}}$ and the diagonal \mathbf{d} of P .

4.1 Outline

We recall here some basic facts on which **IOCG** relies. In the formulas a permutation matrix Π appears, having only the descriptive role of shifting the zeros upward, in order to simplify the notation. Since its use is irrelevant, no reference to Π will appear in the pseudo-code.

Let \mathbf{x}_k , $k = 1, 2, \dots$, be the sequence of vectors obtained by applying CG to the normal equations (5), starting with a vector \mathbf{x}_0 . Because of the semiconvergence property, an *optimal* index K exists such that

$$\|\hat{\mathbf{x}} - \mathbf{x}_K\| \leq \|\hat{\mathbf{x}} - \mathbf{x}_k\|, \quad \text{for any } k.$$

If $\mathbf{x}_K \geq 0$, then it is accepted as the regularized solution of (4). Otherwise, let P be its projection matrix and $\tilde{\mathbf{x}}_K = P\mathbf{x}_K$. Since $\hat{\mathbf{x}} \geq \mathbf{0}$, it is

$$\|\hat{\mathbf{x}} - P\mathbf{x}_K\| < \|\hat{\mathbf{x}} - \mathbf{x}_K\|.$$

Let Π be a permutation matrix such that

$$\Pi P\mathbf{x}_K = \begin{bmatrix} \mathbf{0} \\ \underline{\mathbf{x}}_K \end{bmatrix}, \quad \text{with } \underline{\mathbf{x}}_K > 0.$$

Applying the same projection and permutation to $\hat{\mathbf{x}}$ we have

$$\Pi P\hat{\mathbf{x}} = \begin{bmatrix} \mathbf{0} \\ \underline{\mathbf{x}} \end{bmatrix} \quad \text{where } \Pi\hat{\mathbf{x}} = \begin{bmatrix} \bar{\mathbf{x}} \\ \underline{\mathbf{x}} \end{bmatrix},$$

then

$$\|\hat{\mathbf{x}} - P\mathbf{x}_K\|^2 = \|\Pi\hat{\mathbf{x}} - \Pi P\mathbf{x}_K\|^2 = \|\bar{\mathbf{x}}\|^2 + \|\underline{\mathbf{x}} - \underline{\mathbf{x}}_K\|^2. \quad (8)$$

The basic assumption on which IOCG relies is that the negative components of \mathbf{x}_K , which have been set to zero in $\tilde{\mathbf{x}}_K$, really correspond to nearly zero components of $\hat{\mathbf{x}}$, so that the zero components of $P\mathbf{x}_K$ are correctly placed in the active constraints set \mathcal{A} . The experimentation will show that this assumption is reasonable for images having a large zero background.

The matrix $AP\Pi^T = [O \mid \underline{A}]$ is obtained by zeroing the columns with the indices in \mathcal{A} and shifting them to the left. Then

$$\Pi P A^T A P \Pi^T = \left[\begin{array}{c|c} O & O \\ \hline O & \underline{A}^T \underline{A} \end{array} \right],$$

and system (5) becomes $\Pi P A^T A P \Pi^T \mathbf{y} = \Pi P A^T \mathbf{b}$, where $\mathbf{x} = P\Pi^T \mathbf{y}$. Setting

$$\mathbf{y} = \begin{bmatrix} \bar{\mathbf{y}} \\ \underline{\mathbf{y}} \end{bmatrix} \quad \text{and} \quad \Pi P A^T \mathbf{b} = \begin{bmatrix} \mathbf{0} \\ \underline{\mathbf{c}} \end{bmatrix},$$

we get

$$\underline{A}^T \underline{A} \underline{\mathbf{y}} = \underline{\mathbf{c}}. \quad (9)$$

The smallest $\|\bar{\mathbf{x}}\|$ in (8), the most effective the approximation of $\underline{\mathbf{x}}$ obtained by solving (9) in a regularization sense. Then we apply CG to system (9) starting with $\underline{\mathbf{y}}_0 = \underline{\mathbf{x}}_K$. The semiconvergence will push the iterates $\underline{\mathbf{y}}_k$ toward $\underline{\mathbf{x}}$, i.e. after K' iterations a vector $\underline{\mathbf{y}}_{K'}$ is obtained such that $\|\underline{\mathbf{x}} - \underline{\mathbf{y}}_{K'}\| \leq \|\underline{\mathbf{x}} - \underline{\mathbf{y}}_k\|$, for any k . The vector

$$\mathbf{y}_{K'} = \begin{bmatrix} \mathbf{0} \\ \underline{\mathbf{y}}_{K'} \end{bmatrix}$$

verifies

$$\|\Pi\hat{\mathbf{x}} - \mathbf{y}_{K'}\| \leq \|\Pi\hat{\mathbf{x}} - \Pi P\mathbf{x}_K\| = \|\hat{\mathbf{x}} - P\mathbf{x}_K\|.$$

If also $\underline{\mathbf{y}}_{K'}$ has negative components, the previous arguments are repeated to further improve the approximation to $\hat{\mathbf{x}}$ starting from $\mathbf{y}_{K'}$, and so on.

4.2 The inner-outer structure

The previous considerations suggest an algorithm of inner-outer type for finding a regularized nonnegative solution of problem (4), which appears to be especially suitable for the reconstruction of images having large nearly zero backgrounds. The algorithm consists of two loops:

- (a) the outer loop generates a sequence of projection matrices $P^{(h)}$, $h = 0, 1, \dots$, starting with $P^{(0)} = I$ and for any h calls the inner loop;
- (b) starting with $\mathbf{x}_0^{(h)} \geq \mathbf{0}$, the inner loop computes the sequence $\mathbf{x}_k^{(h)}$, $k = 1, \dots, K_h$ by applying CG to the system

$$A^{(h)T} A^{(h)} \mathbf{x} = A^{(h)T} \mathbf{b}, \quad \text{where } A^{(h)} = AP^{(h)}. \quad (10)$$

The first inner loop is stopped at index K_0 according to a suitable inner stopping rule. The vector $\mathbf{y}^{(0)} = \mathbf{x}_{K_0}^{(0)}$ is assumed as the regularized solution of (10) for $h = 0$. If $\mathbf{y}^{(0)}$ has negative components, a new inner loop starts: the vector $\mathbf{y}^{(0)}$ is projected and the corresponding projection matrix $P^{(1)}$ is constructed. The starting vector for the new inner loop is $\mathbf{x}_0^{(1)} = P^{(1)}\mathbf{y}^{(0)}$ and a new regularized solution $\mathbf{y}^{(1)} = \mathbf{x}_{K_1}^{(1)}$ of (10) is computed for $h = 1$, and so on. As the outer loop goes on, the h th initial vector $\mathbf{x}_0^{(h)}$ has more zeros than the previous initial vector $\mathbf{x}_0^{(h-1)}$. The stopping condition for the outer loop is satisfied by the first h such that $\mathbf{y}^{(h)}$ has all nonnegative components. When this happens, we say that the algorithm has reached its natural termination and the last computed vector is assumed as the regularized solution \mathbf{x}_{reg} of problem (3).

Thanks to the semiconvergence property, an excessive precision in the computation of the last $\mathbf{x}_{K_h}^{(h)}$ is not required at the h th outer phase. For this reason we will impose an upper bound to the number of iterations allowed during an inner loop through a parameter k_{max} .

The k th iteration of CG applied to (10) is

$$\begin{aligned} \mathbf{z}_k^{(h)} &= A^{(h)} \mathbf{p}_k^{(h)}, \quad \alpha_k^{(h)} = \|\mathbf{q}_k^{(h)}\|^2 / \|\mathbf{z}_k^{(h)}\|^2, \quad \mathbf{x}_{k+1}^{(h)} = \mathbf{x}_k^{(h)} + \alpha_k^{(h)} \mathbf{p}_k^{(h)}, \\ \mathbf{r}_{k+1}^{(h)} &= \mathbf{r}_k^{(h)} - \alpha_k^{(h)} \mathbf{z}_k^{(h)}, \quad \mathbf{q}_{k+1}^{(h)} = A^{(h)T} \mathbf{r}_{k+1}^{(h)}, \quad \beta_k^{(h)} = \|\mathbf{q}_{k+1}^{(h)}\|^2 / \|\mathbf{q}_k^{(h)}\|^2, \\ \mathbf{p}_{k+1}^{(h)} &= \mathbf{q}_{k+1}^{(h)} + \beta_k^{(h)} \mathbf{p}_k^{(h)}, \end{aligned} \quad (11)$$

where $\mathbf{x}_0^{(h)}$ is the projection of $\mathbf{x}_{K_{h-1}}^{(h-1)}$, $\mathbf{r}_0^{(h)} = \mathbf{b} - A^{(h)} \mathbf{x}_0^{(h)}$ and $\mathbf{p}_0^{(h)} = \mathbf{q}_0^{(h)} = A^{(h)T} \mathbf{r}_0^{(h)}$.

Due to the presence of the zeros introduced by the matrix $P^{(h)}$, system (10) has an effective size smaller than the size N of system (5). This fact could be exploited to reduce the computational cost of CG at the h th inner loop. However, when A has a structure that can be exploited to reduce the cost of the matrix-vector products (which accounts for the greatest part of the computational cost of the method) and matrix $A^{(h)}$ loses this structure (as it happens with Toeplitz matrices), it is more convenient to keep the matrix A unchanged and to transfer

the effect of $P^{(h)}$ to the vectors, by setting

$$\mathbf{q}_{k+1}^{(h)} = \mathbf{d}^{(h)} \odot A^T \mathbf{r}_{k+1}^{(h)}, \quad (12)$$

where $\mathbf{d}^{(h)}$ is the diagonal of $P^{(h)}$. By effect of the recursion, also the vectors $\mathbf{x}_k^{(h)}$ and $\mathbf{p}_k^{(h)}$ have zero in the positions indicated by $\mathbf{d}^{(h)}$ for any k if this holds for $k = 0$.

4.3 Stopping conditions

The execution flow of the algorithm is ruled by the stopping conditions for the outer and the inner loop. From a theoretical point of view the two stopping conditions could be independent, but in practice it is better to relate them. In fact, nearly stall situations where only few components are eliminated by each inner step for the current outer step should be avoided. This suggests to repeat the outer iterations until the Boolean

$$outer_stop_cond = (\min(\mathbf{y}^{(h)}) < \tau) \ \& \ (k_{in} > k_{min}) \ \& \ (h \leq h_{max})$$

becomes false. The constant τ is a projection threshold, k_{min} is a small fixed integer, k_{in} is the number of iterations performed in the current inner step, h_{max} bounds the number of allowed outer iterations (in the experiments we have set $\tau = -10^{-15}$, $k_{min} = 4$ and $h_{max} = 512$). In any case, it is not possible to continue the outer iteration after its natural termination.

For the inner condition, we note that it is not worthwhile to solve (10) too accurately, for example by determining with a good precision small positive components which would be probably eliminated in a further step. Moreover, the choice of the index k at which the h th inner loop should be stopped may be critical. We assume a stopping condition of the form

$$inner_stop_cond = (\delta_k^{(h)} < \delta_{k-1}^{(h)}) \ \& \ (k \leq k_{max}), \quad (13)$$

where k_{max} bounds the number of allowed inner iterations and $\delta_k^{(h)}$ is a quantity that must be specified in such a way that its minimum is reached when the noise starts contaminating the computation. Ideally, the inner h th loop should be stopped when the relative error $\epsilon_k^{(h)} = \|\hat{\mathbf{x}} - \mathbf{x}_k^{(h)}\| / \|\hat{\mathbf{x}}\|$ starts increasing, but of course such an ideal condition cannot be exploited in a realistic context, where a more practical condition must be implemented.

For an acceptable approximation of the optimal index we use the *Generalized Cross Validation* method (GCV) described in [11, 25]. The performance of GCV has been tested in [7, 8] and it resulted that GCV is more effective than other stopping rules when information about the entity of the noise is not available.

Given a sequence \mathbf{x}_k generated by an iterative method applied to solve system (3), the GCV functional is defined as

$$V_k = \frac{N \|\mathbf{b} - A\mathbf{x}_k\|^2}{(N - \text{trace}(A_k))^2}, \quad (14)$$

where the *influence* matrix A_k is such that $A_k \mathbf{b} = A \mathbf{x}_k$. The minimizer of V_k can be taken as an estimate for the optimal index. If the applied iterative method is **CG**, it is not easy to say how A_k depends on \mathbf{b} and an approximated estimate of the trace of A_k must be provided. Different reliable techniques for estimating the trace are proposed in [8].

Another stopping condition which could be applied is based on the *discrepancy rule*, according which the iteration should stop when the residual vector $\mathbf{r}_k = \mathbf{b} - A \mathbf{x}_k$ verifies $\|\mathbf{r}_k\| \leq \vartheta \eta$, with $\vartheta \geq 1$ a constant and η as given in (2).

To validate the use of the GCV functional, we have performed a preliminary ad-hoc experimentation on the problems considered in the next section, for which the exact solution $\hat{\mathbf{x}}$ is known. By comparing the minima of the sequences $\epsilon_k^{(h)}$ and $V_k^{(h)}$ obtained for fixed h , we have found that the two indications are comparable, even if in the majority of cases the index indicated by GCV is slightly smaller than the one indicated by the error, leading to an anticipated stop of the inner loop. This outcome is considered not damaging in a regularizing context. An analogous experimentation has shown that the use of the discrepancy rule resulted in too anticipated stops of the inner loop.

4.4 Implementation

A pseudo-code for **IOCG**, specifically tailored for circulant matrices, is given at the end of the section. It calls the function **inner** which applies **CG** with the GCV stopping condition implemented in function **isc**. A function **project** is also used, but not coded.

For our experiments we have chosen Toeplitz matrices having the circulant structure which arises when periodic boundary conditions are set. For these matrices the product of A and A^T by a vector can be performed using a low cost algorithm based on the fast Fourier transform (FFT), which is briefly recalled below [12].

A circulant matrix A of size $N = n^2$ is diagonalized by the Fourier matrix \mathcal{F} , whose elements are

$$f_{r,s} = \frac{1}{n} \omega^{rs}, \quad r, s = 0, \dots, N-1, \quad \text{with} \quad \omega = \exp(2\pi i/N).$$

Denoting by \mathbf{a}^T the first row of A , it holds $A = n \mathcal{F} \text{diag}(\mathcal{F} \mathbf{a}) \mathcal{F}^*$, where \mathcal{F}^* is the inverse (i.e. transpose conjugate) of \mathcal{F} . Hence the product $\mathbf{z} = A \mathbf{v}$, where \mathbf{v} and \mathbf{z} are N -vectors, can be so computed

$$\tilde{\mathbf{a}} = \mathcal{F} \mathbf{a}, \quad \tilde{\mathbf{v}} = \mathcal{F}^* \mathbf{v}, \quad \mathbf{z} = n \mathcal{F} (\tilde{\mathbf{a}} \odot \tilde{\mathbf{v}}). \quad (15)$$

To obtain the product $\mathbf{z} = A^T \mathbf{v}$ it is sufficient to take the first column of A as \mathbf{a} or to replace $\tilde{\mathbf{a}}$ by its conjugate $\tilde{\mathbf{a}}^*$. The multiplications by \mathcal{F} and \mathcal{F}^* can be efficiently computed by calling two FFT routines, with a computational cost of order $O(N \log n)$. Using (15), the single step (11) of **CG** requires 4 FFT computations and 7 vector or scalar by vector products, giving a cost of order $O(N \log n)$ per iteration of the algorithm.

A further reduction of the cost when A is circulant is obtained by shifting the computation from the signal domain where A , \mathbf{x} and \mathbf{b} live to the frequency domain, i.e. by replacing matrices and vectors with their transformations in the Fourier basis. All the instructions of the k -th iteration of CG in (11) can be performed in the frequency domain, except instruction (12) which must be performed in the signal domain, as shown in the pseudo-code of function `inner`.

Since A has a 2-level circulant structure, we look for a 2-level circulant approximation C_k of the matrix A_k required by GCV, i.e. $A\mathbf{x}_k \approx C_k \mathbf{b}$. Denoting by \mathbf{c}_k^T the first row of C_k , we have $C_k \mathbf{b} = \mathcal{F} \text{diag}(\mathcal{F}\mathbf{c}_k) \mathcal{F}^* \mathbf{b}$, hence

$$\mathcal{F}\mathbf{c}_k \approx \mathcal{F}^*(A\mathbf{x}_k) \oslash \mathcal{F}^* \mathbf{b} = n \tilde{\mathbf{a}} \oslash \tilde{\mathbf{b}} \odot \tilde{\mathbf{x}}_k, \quad \text{where } \tilde{\mathbf{b}} = \mathcal{F}^* \mathbf{b}, \tilde{\mathbf{x}}_k = \mathcal{F}^* \mathbf{x}_k,$$

and $\text{trace}(A_k) \approx \sum_i (\mathcal{F}\mathbf{c}_k)_i$. Thus in the h th outer loop we use (13) with

$$\delta_k^{(h)} = V_k^{(h)} = N \|\mathbf{r}_k^{(h)}\|^2 / (N - \text{trace}(A_k^{(h)}))^2. \quad (16)$$

A function `isc` which constructs the sequence $\delta_k^{(h)}$ according to (16) is included in the pseudo-code.

```
function  $\mathbf{x}_{reg} = \text{IOCG}(A, \mathbf{b})$ 
```

```
the code is specific for circulant matrices
```

```
 $\tilde{\mathbf{a}} = \mathcal{F}\mathbf{a}; \tilde{\mathbf{b}} = \mathcal{F}^*\mathbf{b}; \tilde{\mathbf{s}} = \tilde{\mathbf{a}} \oslash \tilde{\mathbf{b}};$ 
```

```
 $\mathbf{x}_0^{(0)} = A^T \mathbf{b}; \mathbf{d}^{(0)} = \text{ones}(N, 1);$ 
```

```
 $\text{outer\_stop\_cond} = \text{true};$ 
```

```
 $h = 0;$ 
```

```
while  $\text{outer\_stop\_cond}$ 
```

```
 $\mathbf{y} = \text{inner}(\mathbf{x}_0^{(h)}, \mathbf{d}^{(h)});$ 
```

```
 $[\mathbf{x}_0^{(h+1)}, \mathbf{d}^{(h+1)}] = \text{project}(\mathbf{y});$ 
```

```
 $h = h + 1;$ 
```

```
 $\text{outer\_stop\_cond} = (\min(\mathbf{y}) < \tau) \ \& \ (k_{in} > k_{min}) \ \& \ (h \leq h_{max});$ 
```

```
end
```

```
 $\mathbf{x}_{reg} = \mathbf{x}_0^{(h+1)};$ 
```

```
end
```

```
function  $[\mathbf{y}, k_{in}] = \text{inner}(\mathbf{x}_0, \mathbf{d})$ 
```

```
 $k = 0;$ 
```

```
 $\tilde{\mathbf{x}}_0 = \mathcal{F}^* \mathbf{x}_0; \tilde{\mathbf{r}}_0 = \tilde{\mathbf{b}} - n \tilde{\mathbf{a}} \odot \tilde{\mathbf{x}}_0;$ 
```

```
 $\tilde{\mathbf{p}}_0 = \tilde{\mathbf{q}}_0 = \mathcal{F}^*(\mathbf{d} \odot \mathcal{F}(n \tilde{\mathbf{a}}^* \odot \tilde{\mathbf{r}}_0));$ 
```

```
 $\text{inner\_stop\_cond} = \text{true};$ 
```

```
while  $\text{inner\_stop\_cond}$ 
```

```
 $\tilde{\mathbf{z}}_k = n \tilde{\mathbf{a}} \odot \tilde{\mathbf{p}}_k;$ 
```

```
 $\alpha_k = \|\tilde{\mathbf{q}}_k\|^2 / \|\tilde{\mathbf{z}}_k\|^2;$ 
```

```
 $\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{x}}_k + \alpha_k \tilde{\mathbf{p}}_k;$ 
```

```
 $\tilde{\mathbf{r}}_{k+1} = \tilde{\mathbf{r}}_k - \alpha_k \tilde{\mathbf{z}}_k;$ 
```

```


$$\tilde{\mathbf{q}}_{k+1} = \mathcal{F}^*(\mathbf{d} \odot \mathcal{F}(n \tilde{\mathbf{a}}^* \odot \tilde{\mathbf{r}}_{k+1}));$$


$$\beta_k = \|\tilde{\mathbf{q}}_{k+1}\|^2 / \|\tilde{\mathbf{q}}_k\|^2;$$


$$\tilde{\mathbf{p}}_{k+1} = \tilde{\mathbf{q}}_{k+1} + \beta_k \tilde{\mathbf{p}}_k;$$


$$k = k + 1;$$


$$inner\_stop\_cond = \text{isc}(\tilde{\mathbf{x}}_k, \tilde{\mathbf{r}}_k);$$

end

$$\mathbf{y} = \mathcal{F}\tilde{\mathbf{x}}_{k-1};$$


$$k_{in} = k - 1;$$

end

function  $\chi = \text{isc}(\tilde{\mathbf{x}}_k, \tilde{\mathbf{r}}_k)$ 

$$\mathbf{t} = n \tilde{\mathbf{s}} \odot \tilde{\mathbf{x}}_k; \delta_k = N \|\tilde{\mathbf{r}}_k\|^2 / (N - \sum_i t_i)^2;$$


$$\chi = (\delta_k < \delta_{k-1}) \ \& \ (k \leq k_{max});$$

end

```

5 Numerical experiments

The numerical experimentation has been conducted with double precision arithmetic. We consider three reference objects, namely a satellite image [16], a star cluster¹ and the Hoffman phantom [15], widely used in the literature for testing image reconstruction algorithms. The corresponding images (see Figure 1) are of size $N = 256^2$, normalized between 0 and 1. The percentage of zeros

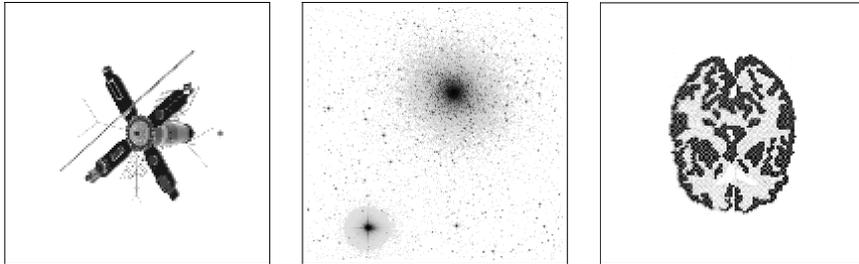


Figure 1: the satellite image (left), the star cluster (middle) and the Hoffman phantom (right).

is 90% for the satellite and 75% for the others. In the images the zeros are associated to a white background. We have adopted this version instead of the standard version which associates zeros to black points, to improve readability. Note that the satellite and the phantom have a large solid interconnected zero background, while the background of the cluster is punctuated by many stars

¹star cluster Messier 5 at http://www.skycrumbles.net/img_info/m5/

and no large zero areas exist. This difference, as we will see, has a certain effect on the reconstructions.

The matrix A which performs the blur is a 2-level Toeplitz matrix generated by a positive space invariant bandlimited PSF with a bandwidth $\nu = 8$, normalized in such a way that the sum of the elements is equal to 1. For the astronomical images (satellite and star cluster) we consider a motion-type PSF, which simulates the one taken by a ground-based telescope, represented by the following mask:

$$m_{i,j} = \exp(-\alpha(i+j)^2 - \beta(i-j)^2), \quad -\nu \leq i, j \leq \nu, \quad \alpha = 0.04, \quad \beta = 0.02.$$

For the medical image (the phantom), we consider a Gaussian PSF represented by the following mask:

$$m_{i,j} = \exp(-\alpha i^2 - \beta j^2), \quad -\nu \leq i, j \leq \nu, \quad \alpha = \beta = 0.1.$$

Since the images have sufficiently large zero background along the boundary, the coefficient matrix is approximated by a 2-level circulant matrix.

For each image, 4 test problems are generated by adding to $\hat{\mathbf{b}} = A\hat{\mathbf{x}}$ Gaussian noises with zero mean and different variances. The test problems so obtained have an average relative noise level η ranging from 1.5% to 5.5% and are solved by IOCG algorithm with a maximum number $k_{max} = 10$ of allowable inner iterations. Experiments with larger values of k_{max} have also been carried out, without significant differences from the point of view of the error, but with a waste of iterations.

We give for comparison purposes also the results of the four methods PRI, ASM, NNFCG and SGP recalled in Section 3 applied with the discrepancy rule as a stopping condition, as suggested in the papers [4, 18, 10, 6]. Since SGP cannot start with the null vector, the vector $A^T \mathbf{b}$ is chosen as initial starting point for all the methods.

For what concerns NNFCG, in [10] it is shown that the quality of the computed results depends on the number \hat{k} only for the early iterations. We report in the tables the results obtained by setting $\hat{k} = 10$; other values of \hat{k} have given very similar results, of course with different storage requirements.

It is easy to see that the cost per iteration of IOCG is dominated by two matrix-by-vector products. This holds also for the other considered methods. Hence only the number of iterations is given in the tables.

Besides the standard measures used to evaluate the effectiveness of a numerical iterative algorithm, i.e. the iteration number and the obtained final error, we borrow from Information Retrieval [22] a tool to measure the ability of the algorithm to correctly detect the zeros of the original image. Let \mathbf{x}_{reg} be the computed solution. Denote ‘‘true zeros’’ the zeros of $\hat{\mathbf{x}}$ and define

- **tp** as the number of zeros in \mathbf{x}_{reg} which are also true zeros (true positives),
- **fp** as the number of zeros in \mathbf{x}_{reg} which are not true zeros (false positives),
- **fn** as the number of nonzeros in \mathbf{x}_{reg} which are true zeros (false negatives),
- **tn** as the number of nonzeros in \mathbf{x}_{reg} which are true nonzeros (true negatives),

- by $p = \text{tp}/(\text{tp} + \text{fp})$ the *precision*,
- by $r = \text{tp}/(\text{tp} + \text{fn})$ the *recall*.

The precision can be seen as a qualitative measure of exactness and the recall can be seen as a quantitative measure of completeness. In simple terms, a large p indicates that the algorithm returns substantially more positive results than irrelevant ones, while a large r means that the algorithm returns most of the relevant results. A measure which combines p and r is the harmonic mean of the two,

$$F_1 = \frac{2}{1/p + 1/r}$$

called *balanced F_1 score*. The largest F_1 , the best the performance of the algorithm.

The tables show the results corresponding to the four different noise levels for the three problems and all the considered methods. For each noise level five different realizations of the random noise vector $\boldsymbol{\eta}$ are considered. Column “it” lists the averages of the total number of iterations, column “err” lists the averages of the final relative errors $\|\hat{\boldsymbol{x}} - \boldsymbol{x}_{reg}\|/\|\hat{\boldsymbol{x}}\|$ and column F_1 lists the averages of the balanced F_1 score. The symbol – for this last measure, which occurs for NNFCG, means that less than 1% of the zeros of the original image were detected.

noise level $\eta = 1.5\%$									
method	satellite			star cluster			Hoffman		
	it	err	F_1	it	err	F_1	it	err	F_1
IOCG	125.6	0.221	0.91	83.4	0.432	0.56	70.6	0.292	0.88
PRI	45.2	0.251	0.61	50.6	0.440	0.37	28.8	0.312	0.62
ASM	29.2	0.249	0.60	35.0	0.440	0.38	23.2	0.311	0.60
NNFCG	29.4	0.253	–	103.0	0.429	–	22.8	0.314	–
SGP	23.0	0.248	0.33	31.2	0.438	0.02	27.0	0.313	0.45

noise level $\eta = 2.5\%$									
method	satellite			star cluster			Hoffman		
	it	err	F_1	it	err	F_1	it	err	F_1
IOCG	77.2	0.228	0.87	41.0	0.439	0.51	42.8	0.299	0.83
PRI	24.0	0.265	0.45	22.4	0.446	0.31	15.2	0.323	0.49
ASM	19.2	0.261	0.47	20.2	0.446	0.33	15.4	0.322	0.51
NNFCG	18.0	0.262	–	56.4	0.437	–	15.0	0.323	–
SGP	15.6	0.255	0.23	25.0	0.440	0.01	15.2	0.323	0.28

noise level $\eta = 4\%$									
method	satellite			star cluster			Hoffman		
	it	err	F ₁	it	err	F ₁	it	err	F ₁
IOCG	51.6	0.236	0.83	31.0	0.443	0.48	37.0	0.306	0.80
PRI	9.0	0.284	0.31	13.6	0.451	0.25	9.4	0.335	0.35
ASM	10.4	0.278	0.36	14.0	0.451	0.21	11.0	0.333	0.33
NNFCG	12.0	0.277	—	35.4	0.444	—	9.8	0.336	—
SGP	11.0	0.266	0.13	14.6	0.446	0.01	11.0	0.331	0.13

noise level $\eta = 5.5\%$									
method	satellite			star cluster			Hoffman		
	it	err	F ₁	it	err	F ₁	it	err	F ₁
IOCG	40.4	0.244	0.78	20.2	0.448	0.41	28.0	0.312	0.75
PRI	7.0	0.299	0.20	8.0	0.455	0.20	6.0	0.345	0.26
ASM	9.0	0.291	0.23	8.0	0.455	0.20	6.0	0.345	0.26
NNFCG	9.0	0.282	—	23.8	0.448	—	7.0	0.343	—
SGP	10.0	0.277	0.01	10.0	0.450	0.01	10.0	0.339	0.02

We note that from the error point of view, **IOCG** outperforms all the other methods with the satellite and the phantom images, while with the star cluster image sometimes **NNFCG** performs better. This means that **IOCG** is particularly suitable for images having a large solid zero background. **IOCG** has a larger number of iterations than the other methods, due to the fact that **IOCG** employs GCV as the inner stopping condition and the natural termination for the outer loop. The discrepancy rule is employed for both the inner and outer stopping conditions with methods **PRI**, **ASM** and **NNFCG** and as unique stopping rule with **SGP**. The better error performance of **IOCG** is not due only to the larger number of iterations. To evidence this fact we have allowed all the other methods to run until a maximum total number of iterations by ignoring the stop controlled by the discrepancy rule. Figure 2 shows typical plots of the relative error histories corresponding to noise level $\eta = 2.5\%$ and obtained by applying **IOCG** (solid black line), **ASM** (dotted line), **NNFCG** (solid gray line) and **SGP** (dashed line). The plots are limited to the first 85 iterates, except for **IOCG** which has its own natural termination. The errors of **PRI** are not shown because nearly overlapped by the errors of **ASM**. We see that if we let **ASM** continue ignoring the stop controlled by the discrepancy rule, the error stagnates while the errors of the other methods tend to decrease. In the case of the star cluster, **NNFCG** and **SGP** might obtain better results than **IOCG**. On the contrary, for the other two images they do not produce lower errors than **IOCG**.

We must note that improvements of the error of such small entities like the ones appearing in the tables produce only slight improvements in the quality of the reconstructed images. Figure 3 shows the enlarged central part of the

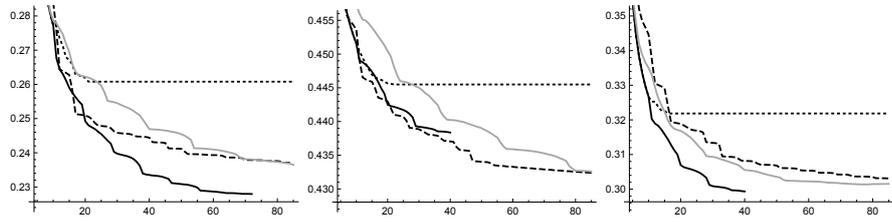


Figure 2: histories of the relative error of IOCG (solid black line), ASM (dotted line), NNFCG (solid gray line) and SGP (dashed line) for the satellite (left), the star cluster (middle) and the Hoffman phantom (right) with $\eta = 2.5\%$.

satellite image reconstructed using IOCG, ASM, NNFCG and SGP, compared to the original and to the blurred images (the image produced by PRI do not differ from the one produced by ASM).

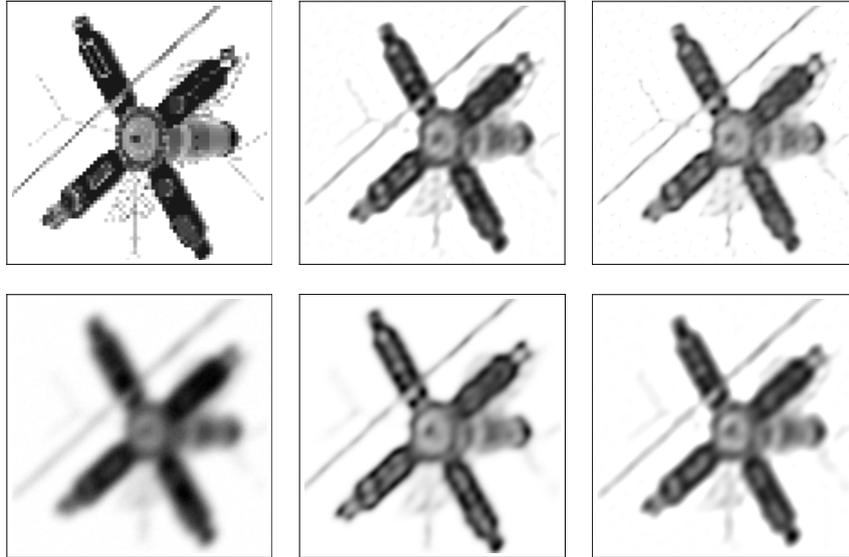


Figure 3: upper row: original image (left), reconstructed images by IOCG (middle), by ASM (right); lower row: blurred image (left), reconstructed images by NNFCG (middle), by SGP (right) for the satellite with $\eta = 2.5\%$.

The parameter F_1 , which decreases with the increasing of the noise, shows that when IOCG is applied, larger active sets imposed on the reconstructed solution effectively correspond to zero components of the original solution. For example, the score $F_1 = 0.83$ of IOCG for the Hoffman phantom with $\eta = 2.5\%$

corresponds to

$$\text{tp} = 35520, \quad \text{fp} = 568, \quad \text{fn} = 14472, \quad \text{tn} = 14976,$$

while the score $F_1 = 0.51$ of ASM for the same problem corresponds to

$$\text{tp} = 17263, \quad \text{fp} = 223, \quad \text{fn} = 32729, \quad \text{tn} = 15321,$$

showing that ASM produces less true results and more false results than IOCG.

As seen in the tables, the F_1 scores of SGP in the case of larger noise levels and of NNFCG for every noise levels are very small. For SGP this depends on the fact that SGP does not apply the projection to the computed solutions but to the directions, and this produces solutions with a reduced number of zeros, especially when there is not a solid background in the original image. For NNFCG this depends on the fact that only one or few zeros are produced at each inner iteration, due to the reduced step length $\bar{\alpha}_k$ computed according to (7). Anyway, the smallness of F_1 does not prevent overall acceptable approximations, as shown by the relative errors.

6 Conclusions

The inner-outer method IOCG has been proposed for the reconstruction of images having large nearly zero backgrounds. The method, tested on ill-posed problems with this feature, appears to be more reliable than the other considered nonnegatively constrained regularization methods from the point of view of the relative error and the number of the reconstructed zeros, even at the cost of a larger number of iterations.

References

- [1] J. M. Bardsley and J. G. Nagy, *Covariance-preconditioned iterative methods for nonnegatively constrained astronomical imaging*, Siam J. Matrix Anal. Appl. **27** (2006), pp. 1184–1197.
- [2] M. Bertero and P. Boccacci, *Introduction to Inverse Problems in Imaging*, Institute of Physics Publishing, Bristol, 1998.
- [3] S. Bonettini, R. Zanella and L. Zanni, *A scaled gradient projection method for constrained image deblurring*, Inverse Problems **25** (2009), 015002.
- [4] D. Calvetti, G. Landi, L. Reichel and F. Sgallari, *Nonnegativity and iterative methods for ill-posed problems*, Inverse Problems **20** (2004), pp. 1747–1758.
- [5] M. E. Daube-Witherspoon and G. Muehllehner, *An iterative image space reconstruction algorithm suitable for volume ECT*, IEEE Trans. Med. Imngng **5** (1986), pp. 61–66.

- [6] P. Favati, G. Lotti, O. Menchi and F. Romani, *Performance analysis of maximum likelihood methods for regularization problems with nonnegativity constraints*, *Inverse Problem* **26** (2010), 085013.
- [7] P. Favati, G. Lotti, O. Menchi and F. Romani, *Stopping rules for iterative methods in nonnegatively constrained deconvolution*, *Applied Numerical Mathematics* **75** (2014), pp. 154–166.
<http://dx.doi.org/10.1016/j.apnum.2013.07.006>
- [8] P. Favati, G. Lotti, O. Menchi and F. Romani, *Generalized Cross-Validation applied to Conjugate Gradient for discrete ill-posed problems*, *Applied Mathematics and Computation*, **243** (2014), pp. 258–268.
- [9] P. Favati, G. Lotti, O. Menchi and F. Romani, “Regularization by conjugate gradient of nonnegatively constrained least squares”, Tech. Report IIT-CNR TR-08/2014.
- [10] S. Gazzola and Y. Wiaux, *Fast nonnegative least squares through flexible Krylov subspaces*, *SIAM J. Sci. Comput.* **39**, pp.A655–A679.
- [11] G. H. Golub, M. Heath and G. Wahba, *Generalized cross-validation as a method for choosing a good ridge parameter*, *Technometrics*, **21** (1979), pp. 215–223.
- [12] G. H. Golub and C. F. Van Loan, *Matrix Computation*, second edition, John Hopkins University Press, Baltimore, MD, 1989.
- [13] M. Hanke, *Conjugate gradient type methods for ill-posed problems*, Pitman Research Notes in Mathematics, Longman, Harlow, 1995.
- [14] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems*, SIAM Monographs on Mathematical Modeling and Computation, Philadelphia, 1998.
- [15] E. J. Hoffman, P. D. Cutler, W. M. Digby and J. C. Mazziotta, *3-D phantom to simulate cerebral blood flow and metabolic images for PET*, *IEEE Trans. Nucl. Sci.* **37** (1990), pp. 616–620.
- [16] K. P. Lee, J. G. Nagy and L. Perrone, *Iterative methods for image restoration: a Matlab object oriented approach*, 2002. Available from <http://www.mathcs.emory.edu/~nagy/RestoreTools>.
- [17] L. B. Lucy, *An Iteration Technique for the Rectification of Observed Distributions*, *Astron. Journal* **79** (1990), pp. 745–754.
- [18] S. Morigi, L. Reichel, F. Sgallari and F. Zama, *An iterative method for linear discrete ill-posed problems with box constraints*, *J. Comput. Appl. Math.* **198** (2007), pp. 505–520.

- [19] J. Nagy and Z. Strakos, *Enforcing nonnegativity in image reconstruction algorithms*, Math. Modeling, Estimation, and Imaging, D. C. Wilson et al., Eds., 4121 (2000), pp. 182–190.
- [20] R. Plato, *Optimal algorithms for linear ill-posed problems yield regularization methods*, Num. Funct. Anal. and Optimiz., **11** (1990), pp. 111–118.
- [21] W. Richardson, *Bayesian-based iterative method of image restoration*, Journal of the Optical Society of America **62** (1972), pp. 55–59.
- [22] C. J. van Rijsbergen, *Information Retrieval*, Butterworth, London, GB, 1979.
- [23] L. A. Shepp and Y. Vardi, *Maximum likelihood reconstruction for emission tomography*, IEEE Trans. Med. Imag., MI-1 (1982), pp. 113–122.
- [24] C. Vogel, *Computational Methods for Inverse Problems*, SIAM Frontier in Applied Mathematics, Philadelphia, 2002.
- [25] G. Wahba, *Practical approximate solutions to linear operator equations when the data are noisy*, SIAM J. Numer. Anal., **14** (1977), pp. 651–667.