Analysis of Interior Point methods for edge–preserving removal of Poisson noise

Silvia Bonettini, Valeria Ruggiero [1]

Contents

[1] This research is supported by the PRIN2008 project of the Italian Ministry of University and Research *Optimization Methods and Software for Inverse Problems*, grant 2008T5KA4L.

Introduction

Interior Point (IP) methods are the core of the more robust packages for solving large–scale, smooth nonlinear programming (NLP) problems (see, for example, LOQO [26], BARNLP [4], IPOPT [27], MOSEK [1], KNITRO [6]). The IP approach is used in a variety of algorithms with different options and heuristics; each variant offers benefits and troubles, often in dependence on the class of the applications. Furthermore, for the same problem, the performance of an IP algorithm can depend on the choices of the values of several parameters.

The aim of this paper is to analyze the behavior of the IP approach for an image processing application that requires to solve a large–scale nonlinear programming problem, such as the denoising of an image corrupted by Poisson noise. In the last years, first–order methods, as gradient and gradient–projection type methods, have been largely exploited for solving the image denoising problem, especially in case of Gaussian data. The interest for these methods is motivated by their simplicity, the low memory requirement and the efficiency in managing nonnegativity constraints. In order to improve the low convergence rate of the gradient methods, a variety of accelerating strategies, especially tailored for the image processing problems have been proposed in the recent literature ([29]). On the other hand, the IP methods have fast asymptotic convergence, but any iteration is, in general, quite expensive to compute and the overall practical behavior depends on several parameters.

Then, we are interested to evaluate the IP approach within the context of image reconstruction, with particular attention to the choice of the parameters for this problem. We devise two different IP algorithms following the two well known globalization strategies, line search and trust region, with the aim to obtain an acceptable compromise between convergence rate and computational cost per iteration. We show that the obtained algorithms can be useful for computing high accuracy solutions.

The paper is organized as follows. In the first section we formulate the image denoising problem in the case of Poisson noise as an NLP problem. In section 2 we give a brief outline of the IP methods, taking into account of the specific structure of the considered problem. In sections 3 and 4 we describe two IP

algorithms: the first one, named IP–LS algorithm, is based on the Newton's method with a backtracking procedure to guarantee feasibility and global convergence; the second one, denoted as IP–TR, follows a trust region strategy to determine a feasible descent direction. In section 5 we report the results of a numerical experimentation on a set of denoising problems, showing the behavior of the two IP algorithms. Our main conclusions are drawn at the end of section 5.
In the following, we denote a diagonal matrix with diagonal entries equal to the elements of the vector a by the capital letter A.

1. The image denoising problem for Poisson data

When the main source of noise is the photon counting, as in emission tomography, microscopy and astronomy, the detected image $y \in \mathbb{R}^N$ can be assumed to be the realization of a Poisson multi–valued random variable [15, 24] whose mean is the true, unknown object $x \in \mathbb{R}^N$. A denoising problem consists in finding an approximation of the true object given the noisy data y and, in the Bayesian framework, the following variational model based on the Total Variation (TV) regularization has been proposed [8, 21, 30]

$$\begin{aligned} &\min f(x) \equiv f_0(x) + \beta f_1(x) \\ &\text{subject to } x \geq \theta \end{aligned} \tag{1.1}$$

where $f_0(x)$ is the Kullback–Leibler divergence of the data y from x:

$$f_0(x) = \sum_{k=1}^{N} \left\{ y_k \ln \frac{y_k}{x_k} + x_k - y_k \right\} \tag{1.2}$$

with $y_k \ln y_k = 0$ if $y_k = 0$ and

$$f_1(x) = \sum_{k=1}^{N} \sqrt{D_1(x_k)^2 + D_2(x_k)^2 + \delta^2} \tag{1.3}$$

where $D_1(x_k)$ and $D_2(x_k)$ represent the forward finite difference operators in the horizontal and vertical directions at the pixel x_k and δ is a nonzero parameter. The regularization function $f_1(x)$ formally describes the HS potential

proposed in [9] and, for small values of δ, it can be considered as a discrete approximation of the TV functional.
The feasible region $\{x \in \mathbb{R}^N, x \geq \theta\}$ is a closed convex subset of the non-negative orthant of $\mathbb{R}^N$ defined by the vector $\theta \in \mathbb{R}^N$, whose components are such that $\theta_k > 0$ if $y_k > 0$ and $\theta_k = 0$ otherwise. Furthermore β is a positive regularization parameter balancing the relative weight of $f_0(x)$ and $f_1(x)$.
We mention that a similar model is proposed in [30] also for other choices of the regularization functional in a more general context of edge–preserving removal of Poisson noise (hypersurface (HS) regularization [9], Markov Random Field (MRF) regularization [16]).
The objective function $f(x)$ is strictly convex on the feasible region [5]. Other features of the NPL problem (1.1) are the large number of variables, the very simple constraints and a structured, sparse and ill-conditioned Hessian matrix. Indeed the Hessian matrix of $f_0(x)$ is a diagonal matrix with entries y_i/x_i^2 while $\nabla^2 f_1(x) = A^T D(x) A$, where $D(x)$ is a nonsingular block diagonal matrix with nonsingular diagonal blocks and A is the matrix of the finite difference approximations to the horizontal and vertical first order partial derivatives, including the chosen boundary conditions (see [5] for details).

Recently, a number of very efficient gradient methods have been developed and analyzed for image denoising, especially for Gaussian noise. These methods require only gradient computation and matrix-vector products (see, for example, [3, 17, 29, 30] and references therein). In literature, a lot of second order methods with superlinear or quadratic convergence have been proposed for box constrained nonlinear programming problems. In the framework of the IP methods, the affine–scaling algorithms developed by Coleman and Li [10, 11] have been intensively investigated (see for example [2, 13, 19, 18, 31]). Their main drawback is that they require the solution of a linear system of equations at each iteration, as well as the evaluation of the Hessian matrix. Nevertheless, in order to preserve computational performance on very large scale problems such as (1.1), it is convenient to devise algorithms whose computational core is given only by matrix-vector products. In the following we consider the IP approach from this point of view, adapting the method to the special features of the image denoising problem (1.1).

2. Interior Point framework

The IP approach consists in solving the problem (1.1) by finding (approximate) solutions of the following barrier problems for a sequence of positive barrier parameter $\{\mu_k\}$ converging to zero:

$$\min \phi_\mu(x) \equiv f(x) - \mu \sum_{i=1}^{N} \ln(x_i - \theta_i) \tag{2.1}$$

where $\mu > 0$ and the domain of $\phi_\mu(x)$ is given by $x > \theta$. The problem (2.1) can be treated as an unconstrained problem and its first–order optimality conditions are

$$\nabla \phi_\mu(x) = \nabla f(x) - \mu(X - \Theta)^{-1} e = 0 \tag{2.2}$$

where $e = (1, ..., 1)^T \in R^N$. By introducing the variable $z = \mu(X - \Theta)^{-1} e$, we can reformulate (2.2) as the following system

$$\begin{aligned} \nabla f(x) - z &= 0 \\ Z(x - \theta) &= \mu e \end{aligned} \tag{2.3}$$

with $x - \theta > 0$ and $z > 0$. In order to understand the role of the variable z, we observe that the nonlinear system (2.3) can be considered a perturbation of the *primal–dual system* representing the Karush Khun Tucker (KKT) conditions of the original problem (1.1). Indeed, introducing the vector z of the Lagrange multipliers of the constraint $x \geq \theta$, the KKT conditions of problem (1.1) are given by the following system

$$\begin{aligned} \nabla f(x) - z &= 0 \\ Z(x - \theta) &= 0 \\ x \geq \theta \qquad & \qquad z \geq 0 \end{aligned} \tag{2.4}$$

As observed in [14], if the system (2.4) is solved by an iterative method, we can avoid stagnation when an iterate reaches the boundary of the feasible region, by perturbing this *primal–dual system* in the last N equations as in (2.3). Then, within the IP approach, starting from a strictly feasible vector $x^{(0)}$ and

a positive vector $z^{(0)}$, we have to determine sequences of strictly feasible iterates $x^{(k)}$ and positive $z^{(k)}$ that, under suitable hypotheses, from the interior of the feasible region converge to a solution x^* of (1.1) and to the related Lagrange multiplier z^*.

In literature, there exist two main strategies to ensure the global convergence of the sequences $\{x^{(k)}\}$ and $\{z^{(k)}\}$, the line search (IP-LS) and trust region (IP-TR) approaches (for a list of references see [7, 28]).

In the line search approach, we apply Newton's method to the system (2.3), backtracking if necessary so that the iterates $z^{(k)}$ and $x^{(k)}$ remain strictly feasible and a convenient merit function is sufficiently reduced.

In the trust region approach, we associate a quadratic program with (2.1) and we define the algorithm steps as approximate solutions of this quadratic subproblem.

In the following two sections, we give further details about the implementation of the two approaches for solving the image denoising problem (1.1)-(1.3).

With regard to the sequence of barrier parameters $\{\mu_k\}$, there exist many procedures to choose the parameter μ_k. We mention the *Fiacco–McCormick monotone* approach, where μ_k is held fixed for a series of iterations until the KKT conditions are satisfied to some accuracy or the *adaptive strategy*, in which the parameter is updated at any iteration with different rules (see [22]). At the present, it is not known which one is the most effective in practice. In the numerical experiments reported in the last section, we use the following rule [26]:

$$\mu_k = \min\left(0.1\frac{(x^{(k)}-\theta)^T z^{(k)}}{N}, \mu_{k-1}\right) \tag{2.5}$$

that assures $\mu_k \leq \mu_{k-1}$.

3. IP-Line Search approach

3.1. Algorithm description

The scheme of IP–LS is reported in Figure 1. The main task consists in

Choose $\mu_0 > 0$, $x^{(0)} > \theta$, $z^{(0)} > 0$.
For $k = 0, 1, ...$

- Compute d_x, d_z;
- Update the primal and dual variables

$$\begin{aligned} x^{(k+1)} &= x^{(k)} + \alpha_x d_x \\ z^{(k+1)} &= z^{(k)} + \alpha_z d_z \end{aligned}$$

- where α_x and α_z are sufficiently small such that
  - $x^{(k+1)} > \theta$, $z^{(k+1)} > 0$
  - an Armijo rule holds for $\phi_{\mu_k}(x^{(k+1)})$
- Update the barrier parameter $\mu_{k+1} \leq \mu_k$

Figure 1 – IP–LS algorithm

computing a line search step (d_x, d_z), applying the Newton's method to the system (2.3):

$$\begin{pmatrix} \nabla^2 f(x^{(k)}) & -I \\ -I & -{Z^{(k)}}^{-1}(X^{(k)} - \Theta) \end{pmatrix} \quad \begin{pmatrix} d_x \\ d_z \end{pmatrix} = \\ = -\begin{pmatrix} \nabla f(x^{(k)}) - z^{(k)} \\ -(x^{(k)} - \theta) + \mu_k {Z^{(k)}}^{-1} e \end{pmatrix} \tag{3.1}$$

As proved in [5], $\nabla^2 f(x)$ is positive definite and the coefficient matrix of (3.1) is nonsingular and quasi definite (N positive eigenvalues and N negative eigenvalues). We can determine an approximate solution of (3.1) by computing d_x first, approximately solving the following *reduced* system

$$(\nabla^2 f(x^{(k)}) + (X^{(k)} - \Theta)^{-1} Z^{(k)}) d_x = -\nabla f(x^{(k)}) + \mu_k (X^{(k)} - \Theta)^{-1} e \tag{3.2}$$

and then we compute d_z:

$$d_z = -z^{(k)} + (X^{(k)} - \Theta)^{-1}(\mu_k e - Z^{(k)} d_x) \tag{3.3}$$

Since the coefficient matrix of the reduced system is a sparse structured symmetric positive definite matrix, we can inexactly solve (3.2) by a preconditioned conjugate gradient (PCG) algorithm. In particular, we choose the diagonal preconditioner $\mathrm{diag}(\nabla^2 f(x^{(k)}) + (X^{(k)} - \Theta)^{-1} Z^{(k)})$ in order to save in computational complexity. Thus every PCG iteration requires only a matrix–vector product.
After computing (d_x, d_z) we update the iterates

$$\begin{aligned} x^{(k+1)} &= x^{(k)} + \alpha_s d_x \\ z^{(k+1)} &= z^{(k)} + \alpha_z d_z \end{aligned} \tag{3.4}$$

where α_x and α_z are chosen to assure strict feasibility and a sufficient decrease of the merit function $\phi_{\mu_k}(x)$. In particular, in order have the strict feasibility of the new iterates, we compute

$$\begin{aligned} \alpha_x^{max} &= \max\{\alpha \in (0,1] : x^{(k)} - \theta + \alpha d_x \geq (1-\tau)(x^{(k)} - \theta)\} \\ \alpha_z^{max} &= \max\{\alpha \in (0,1] : z^{(k)} + \alpha d_z \geq (1-\tau) z^{(k)}\} \end{aligned} \tag{3.5}$$

with $\tau = 0.9995$; then we perform a backtracking procedure that computes the smallest nonnegative integer n such that $\bar{\alpha} = \delta^n$, $\delta \in (0,1)$ and the following condition is satisfied:
(3.6)
$$\phi_{\mu_k}(x^{(k)} + \bar{\alpha}\alpha_x^{\max} d_x) \leq \phi_{\mu_k}(x^{(k)}) + \gamma\bar{\alpha}\alpha_x^{\max} (\nabla f(x^{(k)}) - \mu_k (X^{(k)} - \Theta)^{-1} e) d_x$$

with $\gamma \in (0,1)$ (typically $\gamma = 10^{-4}$). Then the steplengths in (3.4) are given by

$$\begin{aligned} \alpha_x &= \bar{\alpha}\alpha_x^{\max} \\ \alpha_z &= \bar{\alpha}\alpha_z^{\max} \end{aligned}$$

3.2. Starting points

A crucial choice of IP-LS algorithm is the set of starting values. Given an initial feasible vector $x^{(0)}$ and an initial value for $\mu_0 = 1$, the initial multiplier $z^{(0)}$ is computed as

$$z^{(0)} = \min(\bar{z}, \mu_0 (X - \theta)^{-1} e) \tag{3.7}$$

where the minimum is computed componentwise and $\bar{z}$ is the least–squares solution of the system (2.3):

$$\min_{z \geq 0} \|z - \nabla f(x^{(0)})\|^2 + \|(X^{(0)} - \Theta)z - \mu_0 e\|^2$$

Consequently, the initial settings depend only on the choice of $x^{(0)}$ and μ_0.

4. IP-Trust Region approach

4.1. Algorithm description

The IP-TR approach differs from the IP-LS presented in the previous section mainly in the way to compute the step d_x in (3.4). Here the direction d_x is computed by minimizing a quadratic model of the barrier function within a suitable trust region, ensuring in this way global convergence and feasibility. In practice, we can apply the sequential quadratic programming method, tailored for the barrier problem (2.1) with the constraint $x - \theta > 0$. Then, given a barrier parameter μ_k and an estimate of the Lagrange multiplier estimate $z^{(k)}$, we compute a step d_x that (approximately) solves the following subproblem:

$$\begin{aligned} &\min \psi_k(d_x) \equiv= \frac{1}{2} {d_x}^T \nabla^2 \phi_{\mu_k}(x^{(k)}) d_x + \nabla \phi_{\mu_k}(x^{(k)})^T d_x \\ &\|(X^{(k)} - \Theta)^{-1} d_x\|_2 \leq \Delta_k \\ &d_x \geq -\tau(x^{(k)} - \theta) \end{aligned} \tag{4.1}$$

where

$$\begin{aligned} \nabla^2 \phi_{\mu_k}(x^{(k)}) &= \nabla^2 f(x^{(k)}) + Z^{(k)}(X^{(k)} - \Theta)^{-1} \\ \nabla \phi_{\mu_k}(x^{(k)}) &= \nabla f(x^{(k)}) - \mu_k (X^{(k)} - \Theta)^{-1} e \end{aligned}$$

with $\tau = 0.9995$ as in (3.5). The solution of (2.1) is strictly related to the solution of the system (3.2). Indeed, since $z = \mu(X - \Theta)^{-1}e$, the unconstrained minimum of the quadratic function in (4.1) actually is the solution of the linear system (3.2).
In order to approximately solve the subproblem (4.1), we can use the PCG Steihaug method ([25]) combined with the preconditioner :

$$C_k = (X^{(k)} - \Theta)^{-2} \tag{4.2}$$

It is worth to stress that the previous choice of the preconditioner is related to the elliptical trust region constraint in (4.1), which is well suited for bound constrained problems since it discourages moves toward the boundary of the feasible region [2, 10, 11].
As in IP–LS, the computational complexity of the PCG algorithm is equal to a matrix-vector product per iteration. At any PCG iteration, we check if the trust region constraint is satisfied and we stop if the boundary of the region is crossed. Finally we truncate the step d_x if necessary in order to satisfy the constraint $d_x \geq -\tau(x^{(k)} - \theta)$.
Then, following the standard trust region technique [23], the computed step d_x is accepted if the actual reduction of the merit function satisfies the following classical condition:

$$\text{ared}(d_x) \geq \rho \ \text{pred}(d_x) \tag{4.3}$$

where $\text{ared}(d_x) = \phi_{\mu_k}(x^{(k)}) - \phi_{\mu_k}(x^{(k)} + d_x)$ and $\text{pred}(d_x) = \psi_{\mu_k}(0) - \psi_{\mu_k}(d_x)$; usually $\rho = 0.25$. Otherwise, the computed step d_x is discarded.
Finally the new trust region radius Δ_{k+1} is computed. Many strategies have been proposed in the literature for updating the trust region radius: in general, the existing approaches suggest to update it according to the ratio between the actual and predicted reduction, which indicates whether the quadratic model is a good approximation of the barrier function. In particular, if the ratio is sufficiently large it could be convenient to allow a larger step in the next iterate and the radius is increased, while it is reduced otherwise.
In particular, given an initial value for Δ_0 (for example $\Delta_{Max} = 2^{15}$), we use the following updating rule:

$$\begin{aligned}
&\text{if } \frac{\text{ared}(d_x)}{\text{pred}(d_x)} < 0.5 \\
&\quad \Delta_{k+1} = \frac{\Delta_k}{\gamma_0}; \\
&\text{end} \\
&\text{if } \frac{\text{ared}(d_x)}{\text{pred}(d_x)} > 0.7 \\
&\quad \Delta_{k+1} = \min\left(\gamma_1 \Delta_k, \Delta_{Max}\right) \\
&\text{end}
\end{aligned}$$

where $\gamma_0 = 16, \gamma_1 = 4$.
The initial points of IP–TR is set as described in section 3.2. Furthermore, at any iteration k, we compute a least squares approximation of the multipliers vector $z^{(k)}$ from $x^{(k)}$ and the current value of μ_k, in the same way employed to compute the starting value $z^{(0)}$ in (3.7).

4.2. A new preconditioner

We recall that for each k, starting the inner iterations from zero, the first vector computed by the PCG Steihaug method with preconditioner C_k is the Cauchy direction, that is a multiple of the scaled gradient vector of the barrier function at $x^{(k)}$, ${C_k}^{-1}\nabla\phi_{\mu_k}(x^{(k)}) = (x^{(k)} - \Theta)^2\nabla\phi_{\mu_k}(x^{(k)})$.
In other words, there is a clear connection between the preconditioner, the shape of the trust region (as mentioned above) and the scaling idea.
In the framework of the first order method for image denoising, different scaling techniques are proposed to generate efficient methods. In particular, the split–gradient method [20] is based on a decomposition of the gradient of the objective function in a positive and a non positive part

$$\nabla f(x) = v(x) - u(x) \tag{4.4}$$

where $v(x) > 0$ and $u(x) \geq 0$ for all $x \geq \theta$. Using this suggestion, in [30] the authors devise a suitable decomposition of the form (4.4) for the denoising problem (1.1) and design a scaled gradient projection algorithm based on the positive diagonal scaling matrix $(X^{(k)} - \Theta)V(x^{(k)})^{-1}$. In view of the good results obtained with this scaling, we propose as preconditioner in the PCG Steihaug method the inverse of this scaling matrix

$${C_k}^{(1)} = (X^{(k)} - \Theta)^{-1}V(x^{(k)}) \tag{4.5}$$

This choice modifies the shape of the trust region constraint in the quadratic problem (4.1), that becomes

$$\|(X^{(k)} - \Theta)^{-1/2}V(x^{(k)})^{1/2}d_x\|_2 \leq \Delta_k \tag{4.6}$$

Consequently, we obtain that the direction computed at the first inner iteration is the same of the split–gradient method applied to the function $\phi_{\mu_k}(x)$ at

$x^{(k)}$. Numerical experiments in the next section show an improvement of the performance of IP–TR algorithm with the preconditioner ${C_k}^{(1)}$ with respect to the standard scaling (4.2).

5. Numerical results

In order to analyze the behavior of IP–LS and IP–TR algorithms we report the results of a set of numerical experiments on some denoising test problems. The numerical experiments have been performed in Matlab environment (7.5.0) on a PC equipped with a Pentium M715 processor with 512Mb of RAM. The first test image is the LCR–phantom described in [21]. It is an array of 256×256 pixels, representing concentric circles of intensities 70, 135 and 200, enclosed by a square frame of intensity 10, all on a background of intensity 5. The second original image is a 512×512 dental radiography (DR) (see Figure 3, described in [30]. The noisy image y of each problem is obtained by adding Poisson noise to a test image x (the original object) by means of the "imnoise" function included in the Matlab Image Processing Toolbox.
The relative error difference in euclidean norm between the noisy and the noisy–free images is 0.095318 and 0.179 for LCR and DR problems respectively. The original images and the corresponding noisy version are shown in Figures 2 and 3.
For the problems LCR and DR, we use different settings of the regularization parameter β and of the value δ in the functional (1.3). When $\delta \in [10^{-2}, 1]$, $f_1(x)$ can be considered an HS regularization, while for $\delta = 10^{-8}$, it can be interpreted as a discrete approximation of the TV functional. For the choice of the values of β, we follow [30]. In both algorithms, we use the same stopping criteria for the inner PCG method and the outer iterations. In particular, for the inner PCG algorithm with preconditioner C, the inner stopping criterion is the following:

$$\sqrt{r^T C^{-1} r} = \|r\|_{C^{-1/2}} < 0.1$$

Figure 2 – Left panel: the LCR–phantom. Right panel: the corresponding noisy version.

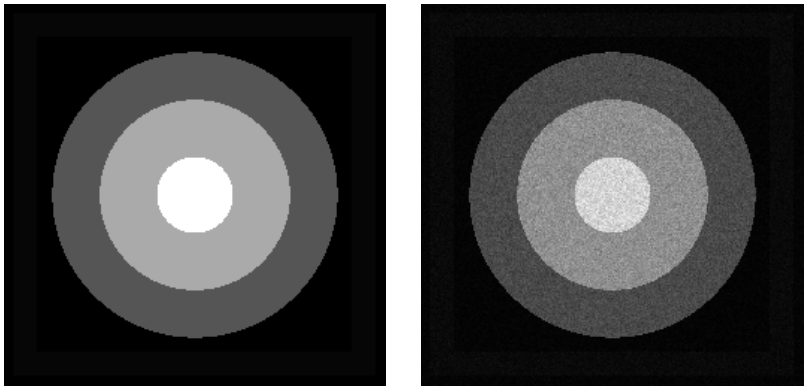

Figure 3 – Left panel: the dental radiography (DR). Right panel: the corresponding noisy version.

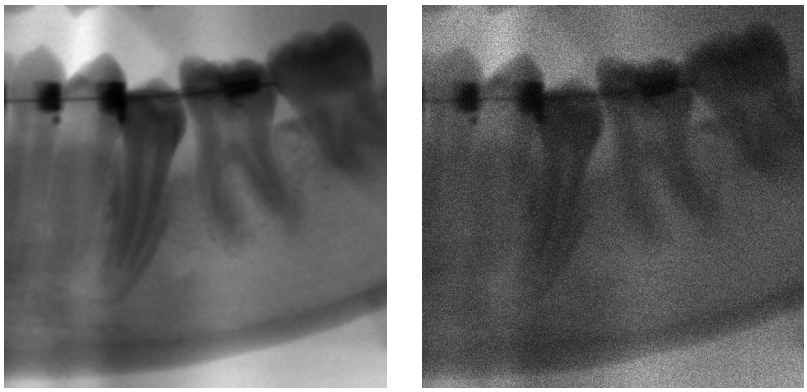

Table 1 – Behavior of IP–TR algorithm combined with different preconditioners in the PCG inner solver; $x^{(0)} = e$.

Test–problem	*Preconditioner*	*it*	*itpcg*	*fevals*
LCR - $\beta = 0.25; \delta = 0.1$	(4.2)	14	9432	15
	(4.5)	15	830	16
DR - $\beta = 0.3; \delta = 1$	(4.2)	10	856	11
	(4.5)	10	209	11

As outer stopping rule in IP–LS and IP–TR algorithms, we check the outer residual related to the KKT conditions (2.4):

$$\left\| \begin{array}{c} \nabla f(x) - z \\ Z(x-\theta) \end{array} \right\| \leq 10^{-6} \left\| \begin{array}{c} \nabla f(x^{(0)}) - z^{(0)} \\ Z^{(0)}(x^{(0)} - \theta) \end{array} \right\| \tag{5.1}$$

The method terminates also when $\mu < 10^{-14}$ or a maximum number ($MaxIter = 1500$) of outer iterations have been executed. In the following tables, we denote by *it* the number of outer iterations and by *itpcg* the total number of PCG iterations. The reconstruction error between the original image $\bar{x}$ and the computed reconstruction x^* has been evaluated by means of the relative error *err* with respect to the euclidean norm $\frac{\|\bar{x}-x^*\|}{\|\bar{x}\|}$. Furthermore, f denotes the value of the objective function at the computed solution $f(x^*)$ and *fevals* the numbers of objective function evaluations that are performed in order to obtain the computed solution x^*. The number of gradient and Hessian evaluations are equal to the number of outer iterations. Table 1 shows as the choice of the preconditioner in the PCG inner solver affects the performance of IP–TR method. While the number of outer iterations is nearly unvaried, the total number of inner iterations of the PCG solver dramatically decreases when we use the preconditioner (4.5), that is strictly related to the features of the image denoising problem. This behavior is coherent to the good performance obtained when (4.5) is used as scaling matrix for first order methods, as focused in [30]. Consequently, all the next numerical results related to IP–TR are obtained using the PCG method with the preconditioner (4.5) as inner solver.
In Tables 2 and 3, we report the results of the behavior of IP–LS and IP–TR respectively for the test problem LCR for different values of β and δ and

different starting points $x^{(0)}$. Here, max $(y, 1)$ indicates the vector obtained by computing componentwise the maximum. The symbol $*$ indicates a failure (the stopping criterion (5.1) has not been satisfied within the maximum number of outer iterations or the total numbers of inner PCG iterations exceeds 40000). From the two tables, we draw the following observations:

- the number of outer iterations is generally small for both algorithms;
- we observe a strong dependence on the starting vector $x^{(0)}$; a suitable choice for $x^{(0)}$ is the vector e or a multiple of e; for $x^{(0)} = e$, the Hessian matrix $\nabla^2 f(e)$ is given by $Y + \beta\nabla^2 f_1(e)$, where $\nabla^2 f_1(e)$ is a sparse matrix where the nonzero entries are of the order of $\frac{1}{\delta}$. For example, the diagonal entries are equal to $\frac{4}{\delta}$. Then, since the Hessian matrix is not excessively ill-conditioned, the choice of e or a multiple of e as starting point seems convenient; indeed also the initial multipliers are not too close to zero;
- the behavior of the two algorithms is very similar, although the performance of IP–TR algorithm seems more sensitive to the values δ;
- for very small values of δ, we observe a bad performance or a failure of both algorithms.

In Tables 4, 5 and 6, we report a comparison between the two IP algorithms and the scaled gradient projection (SGP) method described in [30] on the LCR and DR problems respectively. It is worth stressing that IP and SGP are very different methods, but their comparison suggests interesting remarks. In all tables, the results on the column *SGP* are obtained by stopping the method when the following condition is satisfied [30]:

$$|f(x^{(k+1)}) - f(x^{(k)})| \leq 10^{-7}|f(x^{(k+1)})|$$

The results in the column *SGP** are obtained by stopping the method when the objective function becomes less than $f(x^*)$, where x^* is the solution provided by IP–LS. In all tables, *gevals* and *Hevals* denote the number of gradient and Hessian evaluations performed by the different methods. Figure 4 show the behavior of the objective function and the relative reconstruction error for IP–LS and SGP with respect to the elapsed computation time in the case of LCR

Table 2 – Behavior of IP–LS algorithm.

Test–problem	$x^{(0)}$	f	err	it	$itpcg$	$fevals$
LCR - $\beta = 0.25; \delta = 10^{-1}$	e	55566.4	0.02588	21	815	22
	$10^{-1}e$	55566.4	0.02588	18	662	19
	$\max(y, 1)$	55566.4	0.02590	76	5427	361
LCR - $\beta = 0.2; \delta = 10^{-2}$	e	49239.8	0.02440	116	4885	437
	$10^{-2}e$	49241.3	0.02445	36	3431	121
	$\max(y, 1)$	49238.7	0.02408	273	38546	1952
LCR - $\beta = 0.25; \delta = 10^{-8}$	e	*	*	*	*	*
	$10^{-8}e$	*	*	*	*	*
	$\max(y, 1)$	*	*	*	*	*
DR - $\beta = 0.3; \delta = 1$	e	209705.5	0.02974	18	164	19
	$\max(y, 1)$	209705.5	0.02975	40	578	87

Table 3 – Behavior of IP–TR algorithm.

Test–problem	$x^{(0)}$	f	err	it	$itpcg$	$fevals$
LCR - $\beta = 0.25; \delta = 10^{-1}$	e	55566.4	0.02587	15	830	16
	$10^{-1}e$	55566.4	0.02587	15	912	16
	$\max(y, 1)$	55566.4	0.02586	553	1373	554
LCR - $\beta = 0.2; \delta = 10^{-2}$	e	49262.0*	0.02594*	1500*	5113*	1553*
	$10^{-2}e$	49445.8	0.03139	53	4387	68
	$\max(y, 1)$	49281.6*	0.02365*	1500*	2081*	1500*
LCR - $\beta = 0.25; \delta = 10^{-8}$	e	*	*	*	*	*
	$10^{-8}e$	*	*	*	*	*
	$\max(y, 1)$	*	*	*	*	*
DR - $\beta = 0.3; \delta = 1$	e	209705.5	0.02974	10	209	11
	$\max(y, 1)$	209705.5	0.02974	30	276	40

Table 4 – LCR problem with $\beta = 0.25$, $\delta = 10^{-1}$ and $x^{(0)} = 10^{-1}e$

	IP–LS	*IP–TR*	*SGP**	*SGP*
f	55566.4	55566.4	55566.4	55578.4
err	0.02588	0.02587	0.02582	0.02491
it	18	15	246	167
itpcg	662	912	/	/
fevals	19	16	258	171
gevals	19	16	246	167
Hevals	18	15	/	/
time	33.5	38.7	47.6	25.2

problem with $\beta = 0.25$, $\delta = 10^{-1}$ and $x^{(0)} = 10^{-1}e$.
For LCR problem with $\beta = 0.25$ and $\delta = 10^{-8}$, where IP-LS and IP-TR fail, SGP method produces a solution in 560.2 seconds and it=2454, f=56721.3, *err*=0.02639 and *fevals*=3964.

From these numerical results, we can observe that in general IP-LS algorithm exhibits fast convergence and it enables us to obtain accurate solutions of the optimization problem (1.1). Nevertheless, from the image reconstruction point of view, the goal is to obtain a satisfactory approximation of the original image with a low computational cost; with this aim in mind, the robustness of SGP is evident. Then, IP-LS can be useful for computing benchmark solutions of high accuracy.

A final consideration is concerned with the discrete approximation of TV function, i.e. when δ is very small. Although SGP is able to produce a solution also in this case, we discover a number of numerical instabilities in all methods, such as a very slow convergence or a failure (as in IP approach). In our opinion, this arises since for small δ, the problem (1.1) is very near to be a nonsmooth optimization problem and then a different class of methods should be used to successfully deal with this problem. In literature a number of contributions in this direction have been given for the reconstruction of images corrupted by Gaussian noise (see the references in [12]); a focus about the special case of Poisson noise will be the subject of future work.

Table 5 – LCR problem with $\beta = 0.2$, $\delta = 10^{-2}$ and $x^{(0)} = 10^{-2}e$

	IP–LS	*IP–TR*	*SGP**	*SGP*
f	49241.3	49445.8	49241.3	49256.9
err	0.02445	0.03139	0.02582	0.02350
it	36	53	1052	625
itpcg	3431	4387	/	/
fevals	121	68	1082	639
gevals	37	53	1052	625
Hevals	36	36	/	/
time	133.2	171.3	200.1	119.1

Table 6 – DR problem with $\beta = 0.3$, $\delta = 1$ and $x^{(0)} = e$

	IP–LS	*IP–TR*	*SGP**	*SGP*
f	209705.5	209705.5	209705.5	209705.6
err	0.02974	0.02974	0.02972	0.02972
it	18	10	44	41
itpcg	164	209	/	/
fevals	19	11	46	44
gevals	18	11	44	41
Hevals	18	11	/	/
time	90.8	85.1	43.2	41.3

References

[1] E.D. Andersen, K.D. Andersen: The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm, in H. Frenk, K. Roos, T. Terlaky, S. Zhang, editors, *High Performance Optimization*, 197–232, Dordrecht, The Netherlands, 2000. Kluwer Academic Publishers.

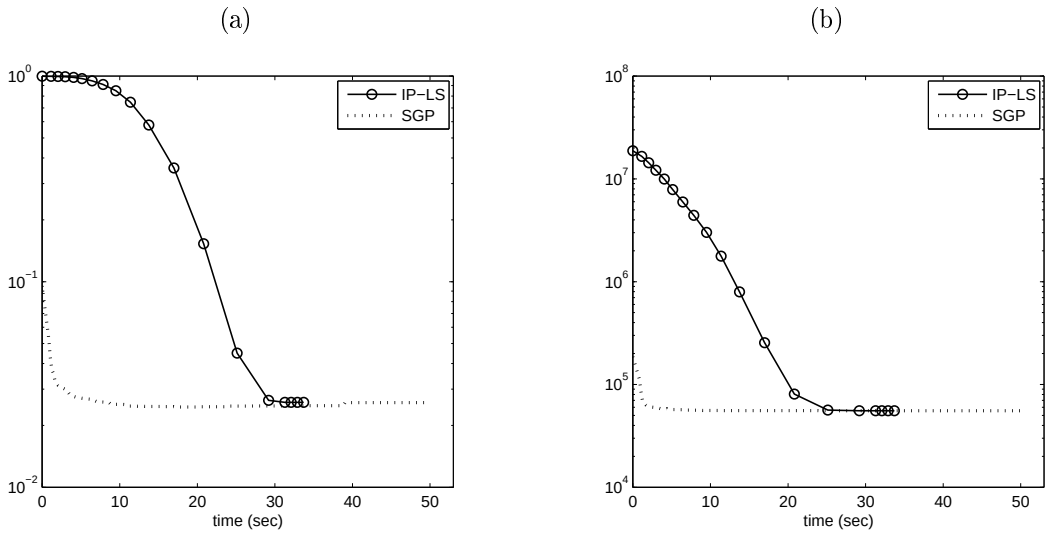

Figure 4 – LCR problem, $\beta = 0.25$, $\delta = 10^{-1}$ and $x^{(0)} = 10^{-2}e$ as starting point for IP-LS. (a) Relative reconstruction error as a function of the computational time (sec.). (b) Objective function value over the computational time. The circles indicates the outer iterations of IP-LS.

[2] S. Bellavia, M. Macconi, B. Morini: An affine scaling trast–region approach to bound–constrained nonlinear systems, in *Applied Numerical Mathematics*, **44**, 257–280, 2003.

[3] M. Bertero, H. Lantéri, L. Zanni: Iterative image reconstruction: a point of view, in Proc. of the Interdisciplinary Workshop on Mathematical Methods in Biomedical Imaging and Intensity-Modulated Radiation Therapy (IMRT), Pisa, Italy, October 2007.

[4] J. Betts, S.K. Eldersveld, P.D. Frank, J.G. Lewis: An interior–point nonlinear programming algorithm for large scale optimization, in *Technical report MCT TECH- 003, Mathematics and Computing Technology*, The Boeing Company, P.O. Box 3707, Seattle WA 98124-2207, 2000.

[5] S. Bonettini, V. Ruggiero: On the uniqueness of the solution of image reconstruction problems with Poisson data, in *AIP Conference Proc.*, **1281**, 1803–1806, 2010.

[6] R.H. Byrd, J. Nocedal, R.A. Waltz: KNITRO: An Integrated Package for Nonlinear Optimization, in G. Pillo, M. Roma, editors, *Large Scale Nonlinear Optimization*, in *Nonconvex Optimization and Its Applications*, **83**, 35–59, Springer, 2006.

[7] R.H. Byrd, J.-CH Gilbert, J. Nocedal: A trust region method based on interior point techniques for nonlinear programming, in Math. Programm., **89**, 1, 149–185, 2000.

[8] R.H. Chan, K. Chen: Multilevel algorithms for a Poisson noise removal model with Total–Variation regularization, in *Int. J. Comput. Math.*, **84**, 1183–1198, 2007.

[9] P. Charbonnier, L. Blanc-Féraud, G. Aubert, A. Barlaud: Deterministic edge–preserving regularization in computed imaging, in *IEEE Trans. Image Processing*, **6**, 298–311, 1997.

[10] T.F. Coleman, Y. Li: On the convergence of interior–reflective Newton methods for nonlinear minimization subject to bounds, in Math. Programm., **67**, 189–224, 1994.

[11] T.F. Coleman, Y. Li: An interior point trust region approach for nonlinear minimization subject to bounds, in SIAM J. Optim., **6**, 418–445, 1996.

[12] J. Dahl, P.C. Hansen, S.H. Jensen, T.L. Jensen: Algorithms and software for total variation image reconstruction via first–order methods, in *Numer. Algor.*, **92**, 53–67, 2010.

[13] J.E. Dennis, L.N. Vicente: Trust–region interior–point algorithms for minimization problems with simple bounds, in H. Fisher, B. Riedmüller, S. Schäffler editors, *Applied Mathematics and Parallel Computing, Festschrift for Klaus Ritter*, Physica, Heidelberg, 97–107, 1996.

[14] A.S. El–Babry, R.A. Tapia, T. Tsuchiya, Y. Zang: On the formulation and theory of Newton interior–point method for nonlinear programming, in J. Optim. Theory Appl., **89**, 507–541, 1996.

[15] S. Geman, D. Geman: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, in *IEEE Trans. Pattern Anal. Intell.*, **6**, 721–741, 1984.

[16] S. Geman, K. Manbeck, D.E. McClure: A comprehensive statistical model for single photon emission tomography, in R. Chellappa, A. Jain A. editors, *Markov Random Fields: theory and Applications*, Boston, Academic Press, 93–130, 1993.

[17] W.W. Hager, B.A. Mair, H. Zhang: An affine–scaling interior–point CBB method for box–constrained optimization, in *Math. Programm.*, **119**, 1, 1–32, 2010.

[18] M. Heinkenschloss, M. Ulbrich, S. Ulbrich: Superlinear and quadratic convergence of affine–scaling interior–point Newton methods for problems with simple bounds without strict complementarity assumption, in *Math. Programm.*, **86**, 615–635, 1999.

[19] C. Kanzow, A. Klug: On affine–scaling interior–point Newton methods for nonlinear minimization with bound constraints, in Comput. Optim. Appl., **35**, 177-197, 2006.

[20] H. Lanteri, M. Roche, C. Aime: Penalized maximum likelihood image restoration with positivity constraints: multiplicative algorithms, in *Inverse problems*, **18**, 1397–1419, 2002.

[21] T. Le, R. Chartrand, T.J. Asaki: A Variational Approach to Reconstructiong Images Corrupted by Poisson Noise, in *J. Math. Imaging Vis.*, **27**, 257–263, 2007.

[22] J. Nocedal, A. Wächter, R.A. Waltz: Adaptive barrier strategies for nonlinear interior point methods, in *Technical Report RC 23563*, IBM T. J. Watson Research Center, Yorktown Heights, NY, USA, March 2005.

[23] J. Nocedal, S.J. Wright: *Numerical Optimization*, Springer Series in Operations Research and Financial Engineering, Springer, 1999.

[24] L.A. Shepp, Y. Vardi: Maximum likelihood reconstruction for emission tomography, in *Trans. Med. Imaging*, **MI-1**, 113–122, 1982.

[25] T. Steihaug: The conjugate gradient method and trust regions in large scale optimization, in *SIAM J. Numer. Anal.*, **20**, 3, 626–637, 1983.

[26] R.J. Vanderbei, D.F. Shanno: An interior point algorithm for nonconvex nonlinear programming, in *Comput. Optim. Appl.*, **13**, 231–252, 1999.

[27] A. Wächter, L.T. Biegler: On the implementation of a primal–dual interior point filter line search algorithm for large–scale nonlinear programming, in *Technical Report RC 23149*, IBM T. J. Watson Research Center, Yorktown Heights, NY, USA, March 2004.

[28] R.A. Waltz, J.L. Moralesy, J. Nocedal, D. Orban: An Interior Algorithm for Nonlinear Optimization That Combines Line Search and Trust Region Steps, in Math. Programm., **107**, 3, 391–408, 2006.

[29] C. Vogel: *Computational Methods for Inverse Problems*, SIAM, Philadelphia, 2002.

[30] R. Zanella, P. Boccacci, L. Zanni, M. Bertero: Efficient gradient projection methods for edge–preserving removal of Poisson noise, in *Inverse Problems*, **25**, 4, 2009.

[31] Y. Zhang: Interior point gradient method with diagonal–scaling for simple–bounds constrained optimization, in *Tech. Rep. TR04-06*, Departement of Computational and Applied Mathematics, Rice university, Houston, Texas, 2004.