An Interior-Point Trust-Region-Based Method for Large-Scale Nonnegative Regularization

Marielba Rojas [*] Trond Steihaug [†]

July 6, 2001
(Revised December 19, 2001)

CERFACS Technical Report TR/PA/01/11

Abstract

We present a new method for large-scale nonnegative regularization, based on a quadratically and nonnegatively constrained quadratic problem. Such problems arise for example in the regularization of ill-posed problems in image restoration where, in addition, some of the matrices involved are very ill-conditioned. The method is an interior-point iteration that requires the solution of a large-scale and possibly ill-conditioned parameterized trust-region subproblems at each step. The method uses recently developed techniques for the large-scale trust-region subproblem. We describe the method and present preliminary numerical results on test problems and image restoration problems.

AMS classification: Primary: 86A22. Secondary: 65K10, 90C06

Key words and phrases: Regularization, constrained quadratic optimization, trust region, Lanczos method, ill-posed problems, inverse problems, image restoration.

1 Introduction

We consider the problem

$$\begin{array}{ll} \min & \frac{1}{2}\|Ax-b\| \\ s.t. & \|x\| \leq \Delta \\ & x \geq 0 \end{array} \quad (1)$$

[*]CERFACS, 42 avenue Gaspard Coriolis, 31057 Toulouse CEDEX 1, France (mrojas@cerfacs.fr). This author was supported in part by the Research Council of Norway.

[†]Department of Informatics, University of Bergen, Høyteknologisenteret, N-5020 Bergen, Norway (trond@ii.uib.no). This author was supported by the Research Council of Norway.

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, and $\Delta > 0$. Throughout the paper, $\|\cdot\|$ denotes the ℓ_2-norm.

We are assuming that m and n are large, and that the matrix A might not be explicitly available, but that we know how to compute the action of A and A^T on vectors of appropriate dimensions.

Problem (1) is an important problem that arises for example in the regularization of ill-posed problems from image restoration (cf. [1]), where we want to recover an image from a blurred and noisy version of it. In these problems, the matrix A is a discretized version of a blurring operator, and b is a vector representation of a degraded image. In image restoration problems, the norm constraint is a so-called regularization term that controls the growth in the size of the least squares solution observed in most ill-posed problems with noisy data, and the nonnegativity constraints reflect the fact that each component of the vector x represents either the color value, or the light-intensity value of a pixel in the digital representation of the image, and therefore must be nonnegative.

Most techniques for image restoration do not take into account the nonnegativity constraints. Instead, they solve the regularization problem, for example, the least squares problem with the norm constraint only, and set to zero the negative components in the solution. This strategy clearly introduces some errors, but produces satisfactory results in certain cases, such as when the images are normal photographs. However, this is not the case in other applications. As pointed out in [9], [15], in astronomical imaging applications most of the pixel values in the desired solution are actually zero or nearly zero, and therefore setting negative values to zero might introduce considerable errors, and might yield a restored image with missing details or with artifacts.

Some of the methods for regularization with nonnegativity constraints are [4], [5], [9], [15]. We also have methods that follow the approach of projections on convex sets or POCS methods such as [17], [25]. Finally, it is worth mentioning the collection of software in [23] which includes routines for least squares problems with nonnegativity constraints.

The methods in [4] are based on Truncated Singular Value Decomposition (TSVD) regularization [11]. The approach can be used on large-scale problems by computing only a few singular values, although in general it is difficult to determine how many values should be computed. The authors propose methods based on a linearly constrained quadratic problem and its dual. The primal approach, at least in the current version, can be used only on small problems. The dual approach is suitable for large-scale problems, although it might require additional regularization and the method might fail to converge in certain cases. The authors report results for small problems only. The solutions computed by these methods have very attractive theoretical properties. The work in [5] proposes an active-set quadratic-programming method for a problem closely related to (1). That method has been successfully applied to problems of moderate size ($n \approx 2000$), but the current version is prohibitively expensive for larger problems, such as those in image restoration, where typically $n = 65536$. The methods in [9] are based on a quasi-Newton approach, and appear as promising strategies. The authors point out that good preconditioners are needed to improve efficiency, and more experiments are also needed to assess the effectiveness of the methods. The methods in [15] are iterative methods for linear systems that impose a nonnegativity constraint at each step. The methods achieve regularization by early termination of the iteration, which is based on the heuristic that initial iterates will not be contaminated by the noise in the data, and which requires a procedure for determining when to stop the iteration. In practice, most regularization approaches based on iterative methods use visual inspection for stopping the iteration. Preconditioners are used in [15] to obtain computationally competitive methods.

In this work, we present a new method for large-scale nonnegative regularization. Our method is not based on a heuristic and it does not depend on the availability of a preconditioner. The method is matrix-free in the sense that only matrix-vector products with A and A^T are required.

The organization of the paper is the following. We derive our method for problem (1) in Section 2. In Section 3 we show some preliminary numerical results on ill-posed problems from inverse problems, including image restoration problems, and discuss some of the properties of the method. Concluding remarks are presented in Section 4.

We will also use the following notation throughout the paper. We denote vectors by lower case letters, vector components by the corresponding Greek letter with a subscript, and matrices by capital letters. Given a vector $x = (\xi_1, \xi_2 \ldots, \xi_n)^T$, we write $diag(x)$ or $diag(\xi_1, \xi_2 \ldots, \xi_n)$ to denote a diagonal matrix with element ξ_i in the (i, i) position. Finally, we denote the vector of all ones by $e = (1, \ldots, 1)^T$.

2 The Method

Before deriving the method, and for clarity of presentation, we define $H = A^T A$ and $g = -A^T b$, and formulate the following problem equivalent to (1)

$$\begin{aligned} \min \quad & \tfrac{1}{2} x^T H x + g^T x. \\ s.t. \quad & \|x\| \leq \Delta \\ & x \geq 0 \end{aligned} \tag{2}$$

We observe that problem (2) always has a solution. Let $\lambda \in \mathbb{R}$ and $y \in \mathbb{R}^n$, then the Lagrangian functional associated with the problem is

$$\mathcal{L}(x, \lambda, y) = \frac{1}{2} x^T H x + g^T x - \frac{\lambda}{2}(\|x\|^2 - \Delta^2) + y^T x.$$

The Karush-Kuhn-Tucker (KKT) first-order necessary conditions for a feasible point x and Lagrange multipliers λ and y to be a solution of problem (2) are

$$\begin{array}{rl} (i) & (H-\lambda I)x = -g - y \\ (ii) & \lambda(\|x\|^2 - \Delta^2) = 0 \\ (iii) & y^T x = 0 \\ (iv) & \lambda \le 0, y \le 0. \end{array} \tag{3}$$

The vector of Lagrange multipliers (or dual variables) y and the *duality gap* $y^T x$ will play a key role in the algorithm.

In order to develop our method for problem (2), we firt eliminate the nonnegativity constraints by restricting our attention to $x > 0$ and introducing a modified objective function. Several choices are possible for this function, for example, an entropy function is used in [15]. Although such choices can also be included in our approach, here we shall derive the method for the *logarithmic barrier function*, defined as

$$f_\mu(x) = \tfrac{1}{2}x^T Hx + g^T x - \mu \sum_{i=1}^{n} \log \xi_i$$

where $x = (\xi_1, \xi_2, \ldots, \xi_n)^T$, and $\mu > 0$ is the so-called *barrier* or *penalty parameter*. The use of the modified function yields a family of problems depending on μ, where each problem is given by

$$\begin{array}{rl} \min & f_\mu(x). \\ s.t. & \|x\| \le \Delta \end{array} \tag{4}$$

The Lagrangian functional associated with this problem is

$$\mathcal{L}(x, \lambda) = f_\mu(x) - \frac{\lambda}{2}(\|x\|^2 - \Delta^2),$$

where $\lambda \in \mathbb{R}$. Let $X = diag(x)$, then the following are the KKT necessary conditions for a feasible point x and Lagrange multiplier λ to be a solution of problem (4)

$$\begin{array}{rl} (i) & (H + \mu X^{-2} - \lambda I)x = -g \\ (ii) & \lambda(\|x\|^2 - \Delta^2) = 0 \\ (iii) & \lambda \le 0. \end{array} \tag{5}$$

The idea of the method is then to solve a sequence of problems of type (4), while decreasing the parameter μ towards zero. Notice that by using problem (4) we have restricted the solution to have positive components only. This follows an interior-point approach (cf. [3], [6], [24]), in which the iterates are feasible and positive.

We shall now introduce a further simplification by substituting the nonlinear barrier problems (4) by quadratically constrained quadratic problems, or trust-region subproblems, where the objective function will be a quadratic approximation to the logarithmic barrier function, and where the trust-region radius Δ will remain fixed. The subproblems are constructed as follows.

Consider the second-order Taylor expansion of f_μ around a point x,

$$q_\mu(x+h) = f_\mu(x) + \nabla f_\mu(x)^T h + \tfrac{1}{2} h^T \nabla^2 f_\mu(x) h,$$

where $\nabla^2 f_\mu(x) = H + \mu X^{-2}$ and $\nabla f_\mu(x) = Hx + g - \mu X^{-1} e$, with $X = diag(x)$. Let us now formulate the following trust-region subproblem

$$\begin{array}{cl} \min\limits_{h} & q_\mu(x+h), \\ s.t. & \|x+h\| \leq \Delta \end{array} \tag{6}$$

and notice that setting $z = x+h$ we obtain a new trust-region subproblem equivalent to (6), and given by

$$\begin{array}{cl} \min\limits_{z} & \tfrac{1}{2} z^T (H + \mu X^{-2}) z + (g - 2\mu X^{-1} e)^T z. \\ s.t. & \|z\| \leq \Delta \end{array} \tag{7}$$

As established in [7] and [21], necessary and sufficient conditions for a feasible point z and Lagrange multiplier $\lambda \in \mathbb{R}$ to be a solution of problem (7) are

$$\begin{array}{rl} (i) & (H + \mu X^{-2} - \lambda I) z = 2\mu X^{-1} e - g \\ (ii) & H + \mu X^{-2} - \lambda I \ \text{ positive semidefinite} \\ (iii) & \lambda(\|z\|^2 - \Delta^2) = 0 \\ (iv) & \lambda \leq 0. \end{array} \tag{8}$$

Our method consists of solving a sequence of problems of type (7) for z, for different values of μ and x, while driving the barrier parameter μ towards zero, and preserving positive iterates. The latter is accomplished by means of a linesearch to be described in Section 2.3. Denoting problem (7) by $P(x, \mu)$, we can now write our method as Algorithm 2.1 in Figure 1.

We will next describe each component of the algorithm in detail, namely, the update of the barrier parameter, the choice of initial values, the linesearch, and the stopping criteria.

Algorithm 2.1 *TRUST*$_\mu$

Input: $H \in \mathbb{R}^{n\times n}$, *symmetric, or procedure for computing* H *times a vector;* $g \in \mathbb{R}^n$; $\Delta > 0$; $\varepsilon_y, \varepsilon_f, \varepsilon_x, \sigma \in (0,1)$.
Output: $x_* > 0$ *satisfying (8) for* μ *close to zero.*

1. `Choose` $x_0 > 0,\ \mu_0 > 0$, `set` $k = 0$
2. **while** `not convergence` **do**
  2.1 `Solve` $P(x_k, \mu_k)$ `for` z_k
  2.2 `Set` $h_k = z_k - x_k$
  2.3 `Compute` β_k `such that` $x_k + \beta_k h_k > 0$
  2.3 `Set` $x_{k+1} = x_k + \beta_k h_k$
  2.4 `Compute` μ_{k+1} `such that` $\{\mu_k\} \to 0$
  2.5 `Set` $k = k + 1$
 end while

Figure 1: Method for Trust-Region Subproblems with Nonnegativity Constraints.

2.1 Update of the barrier parameter μ

We can derive a formula for updating the barrier parameter by first computing an approximation to the dual variables y in (3) in the following way. Recall from (3)(i), that y satisfies

$$y = -(H - \lambda I)x - g,$$

and notice that a solution z, λ of the trust-region subproblem (7) will satisfy (8)(i), which we can rewrite as

$$-(H - \lambda I)z - g = \mu X^{-2} z - 2\mu X^{-1} e. \tag{9}$$

We now define

$$\widetilde{y} = -(H - \lambda I)z - g,$$

and use $\widetilde{y}$ as an approximation to y. Observe that we can compute $\widetilde{y}$ from (9) as

$$\widetilde{y} = \mu(X^{-2} z - 2X^{-1} e), \tag{10}$$

and when $x = z$, we have the following approximation to the duality gap in (3)(iii)

$$\widetilde{y}^T x = -\mu n, \tag{11}$$

which leads to the following formula for μ

$$\mu = \frac{1}{n}\, \widetilde{y}^T x.$$

In practice, the update for μ will be

$$\mu_{k+1} = \frac{\sigma}{n} \, |\tilde{y}^T x|, \tag{12}$$

with $\sigma \in (0,1)$, and $x = x_k$ or $x = x_{k+1}$, and with $\tilde{y}$ as in (10) for $\mu = \mu_k$, $x = x_k$ or $x = x_{k+1}$, and $z = z_k$.

2.2 Choice of initial values x_0, μ_0

To compute initial values for x and μ, we first solve the trust-region subproblem *without* the nonnegativity constraints, i.e.

$$\begin{array}{ll} \min & \frac{1}{2}\|Ax - b\| \ , \\ s.t. & \|x\| \leq \Delta \end{array} \tag{13}$$

and we denote the solution to this problem and the corresponding Lagrange multiplier by x_{TRS} and λ_{TRS}, respectively. We use x_{TRS} and λ_{TRS} to compute an initial value for μ in the following way.

We first compute an approximate initial value for the dual variables y as

$$\tilde{y}_0 = -g - (H - \lambda_{TRS} I) x_{TRS},$$

and then compute μ_0 as

$$\mu_0 = \frac{\sigma}{n} \, |\tilde{y}_0^T x_{TRS}|.$$

We then choose x_0 as either $x_0 = |x_{TRS}|$ with zero components replaced by a small positive value, or $x_0 = x_{TRS}$ with negative and zero components replaced by a small positive value, so $x_0 > 0$. We use x_0 to test for convergence as described in Section 2.4.

2.3 Linesearch

A linesearch is necessary to ensure that the iterates x_k remain positive, since there is no guarantee that z_k computed in Step 2.1 will have only positive components. The *k+1*-st iterate is computed as $x_{k+1} = x_k + \beta_k h_k$, where $h_k = z_k - x_k$ and

$$\beta_k < \min_{i \text{ s.t. } 1 \leq i \leq n \text{ and } \zeta_i \leq 0} \frac{\xi_i}{|\eta_i|}$$

where $x_k = (\xi_1, \xi_2, \ldots, \xi_n)^T$, $z_k = (\zeta_1, \zeta_2, \ldots, \zeta_n)^T$, and $h_k = (\eta_1, \eta_2, \ldots, \eta_n)^T$.

In practice, we use the following safeguarded formula to update the iterates

$$x_{k+1} = x_k + \ \min\{1, 0.9995\beta_k\} h_k.$$

2.4 Stopping Criteria

The stopping criteria relies on the change in value of the objective function, the proximity of the iterates, and the size of the duality gap. For the latter, we compute $\widetilde{y}_k$ according to (10), with $\mu = \mu_{k-1}$ or $\mu = \mu_k$, $x = x_{k-1}$ or $x = x_k$, and $z = z_{k-1}$ computed in Step 2.1 of Algorithm 2.1.

Let $f(x) = \frac{1}{2}x^T Hx + g^T x$, and $\varepsilon_f, \varepsilon_x, \varepsilon_y \in (0,1)$, then for $k \geq 1$, Algorithm 2.1 proceeds until

$$|f(x_k) - f(x_{k-1})| \leq \varepsilon_f |f(x_k)| \quad \text{or} \quad \|x_k - x_{k-1}\| \leq \varepsilon_x \|x_k\| \quad \text{or} \quad |\widetilde{y}_k^T x_k| \leq \varepsilon_y \|x_k\|.$$

For $k = 0$ we only check the last condition for the initial values of x and $\widetilde{y}$, i.e. $|\widetilde{y}_0^T x_0| \leq \varepsilon_y \|x_0\|$, with $\widetilde{y}_0$ and x_0 as in Section 2.2.

3 Numerical Results

In this section we present numerical results that illustrate some properties of the method TRUST$_\mu$ and the performance of the method on image restoration problems.

The results in this section were obtained with a `MATLAB 5.3` implementation of Algorithm 2.1 applied to constrained least squares problems of type (1). We ran our experiments on a SUN Ultra-250 with a 400 MHZ processor and 2048 Megabytes of RAM running Solaris 5.8. The floating point arithmetic was IEEE standard double precision with machine precision $2^{-52} \approx 2.2204 \cdot 10^{-16}$.

In Section 3.1 we discuss the solution of the large-scale trust-region subproblems in Step 2.1 of Algorithm 2.1. In Sections 3.2 and 3.3 we illustrate two aspects of the method. The first one is the convergence behavior with respect to the update of the barrier parameter μ. The second one is the accuracy of the solutions with respect to the desired solution. For this purpose we use test problems from the Regularization Tools package [10]. All the problems in this package are discretized versions of Fredholm integral equations of the first kind. In Section 3.4 we report results on astronomical imaging problems from [12].

In all the experiments, x_{TRS} denotes the solution of the trust-region subproblem (13), x_μ denotes the solution of the nonnegatively constrained trust-region problem (1) computed with TRUST$_\mu$, and x_{IP} denotes the discretized version of the exact solution of the inverse problem which was available for all tests. We have chosen $\Delta = \|x_{IP}\|$ in all the experiments. We computed formulas (10) and (12) using the previous iterate. The relative errors in x_{TRS} and x_μ are computed as $\frac{\|x_{TRS} - x_{IP}\|}{\|x_{IP}\|}$ and $\frac{\|x_\mu - x_{IP}\|}{\|x_{IP}\|}$, respectively.

3.1 Solution of the Trust-Region Subproblems

The trust-region subproblems in Step 2.1 of Algorithm 2.1, were solved with a `MATLAB 5.3` implementation of the method LSTRS from [19]. The method is based on matrix-vector products with A and A^T, and can handle the singularities associated with ill-posed problems. The method has been successfully used to solve regularization problems through the trust-region approach (13) in seismic inversion and related problems [18], [20]. LSTRS was also used to compute an initial iterate x_0 as discussed in Section 2.2.

LSTRS is an iterative method that requires the solution of a large-scale eigenvalue problem at each step. Unless otherwise indicated, the eigenvalue problems were solved by means of the Implicitly Restarted Lanczos Method (IRLM) [22] as implemented in `ARPACK` [13]. The IRLM is particularly suitable for large-scale problems since it has low and fixed storage requirement and relies upon matrix-vector products only. In the current implementation of LSTRS, a Mexfile interface was used to access `ARPACK`. Notice that the capabilities of `ARPACK` are now available through the routine `eigs` in `MATLAB 6`.

The eigenvalue problems in LSTRS have the form

$$\begin{pmatrix} \alpha & g^T \\ g & H \end{pmatrix} y = \lambda y \tag{14}$$

where α is a scalar parameter updated at each iteration, and where we are interested in the smallest eigenvalue. The goal of LSTRS is to find an optimal value α_* for the parameter α. A solution to the trust-region subproblem can then be recovered from the solution of (14) for $\alpha = \alpha_*$. We refer the reader to [19] for more details.

One strategy that we have implemented in most of our experiments is to use the optimal value of α for the previous subproblem as initial α to solve the current subproblem. Intuitively, this would reduce the number of LSTRS iterations since we do not expect x and μ to change significantly as TRUST$_\mu$ converges, and therefore the trust-region subproblems in Step 2.1 of Algorithm 2.1 should not differ significantly. In practice, we have observed that using this strategy indeed reduces the number of LSTRS iterations and consequently the number of matrix-vector products required.

In all the experiments the parameters for the IRLM were eleven Lanczos basis vectors with nine shifts (nine matrix-vector products) on each implicit restart, with a maximum of thirteen implicit restarts allowed. The eigenvalues were computed to a relative accuracy of 0.5.

LSTRS also needs some tolerances as input. The tolerances are ε_Δ, the relative accuracy in the norm of the trust-region solution with respect to the prescribed trust-region radius Δ; ε_{Int}, to decide when a solution is interior; ε_{HC} to compute a nearly optimal solution in a special case usually called the hard case; ε_α, to decide when the parameter α can no longer be updated; and ε_ν, to determine when an eigenvector component is too small (close to zero). In all the experiments, $\varepsilon_{Int} = 0$, $\varepsilon_\alpha = 10^{-8}$, and $\varepsilon_\nu = 10^{-2}$. We will indicate the values for the rest of the tolerances of LSTRS when we describe each particular experiment.

3.2 Convergence rate with respect to μ

In these experiments we have chosen problem **foxgood** from [10], and set the dimensions to $m = n = 300$. As described in Section 3.1, the method LSTRS was used to compute x_{TRS}, a solution to (13). LSTRS computed a positive solution for problem **foxgood**, thus we could use this solution to test the convergence of the TRUST$_\mu$ iterates.

The eigenvalue problems in LSTRS were solved with the Matlab routine `eig`, and the initial value of the parameter α was set to zero in each call to LSTRS. The tolerances for LSTRS were $\varepsilon_\Delta = 10^{-3}$ and $\varepsilon_{HC} = 10^{-10}$. In TRUST$_\mu$ we did not used the stopping criteria in Section 2.4, and we let the method run for a large number of iterations to be able to observe the convergence behavior.

The convergence behavior of TRUST$_\mu$ for linear and quadratic updates of μ is reported in Tables 1 and 2, respectively, where x_k denotes the TRUST$_\mu$ iterates. These results seem to indicate that the method TRUST$_\mu$ has the very desirable property that the convergence rate of the sequence of iterates is determined by the update of the barrier parameter.

μ_k	$\frac{\|x_k - x_{TRS}\|}{\|x_{TRS}\|}$
1.0000e+00	2.8622e-01
1.0000e-01	4.0446e-01
1.0000e-02	4.6372e-01
1.0000e-03	2.0026e-01
1.0000e-04	1.4872e-01
1.0000e-05	8.1839e-02
1.0000e-06	3.0494e-02
1.0000e-07	7.7923e-03
1.0000e-08	1.3599e-03
1.0000e-09	1.6415e-04
1.0000e-10	1.6770e-05
1.0000e-11	1.6798e-06
1.0000e-12	1.6800e-07

Table 1: Convergence rate of TRUST$_\mu$ iterates for linear update $\mu_{k+1} = 0.1\mu_k$.

μ_k	$\frac{\|x_k - x_{TRS}\|}{\|x_{TRS}\|}$
1.0000e-01	2.8723e-01
1.0000e-02	4.0645e-01
1.0000e-04	1.4009e-01
1.0000e-08	5.4001e-04
1.0000e-16	1.2630e-10
1.0000e-32	0

Table 2: Convergence rate of TRUST_μ iterates for quadratic update $\mu_{k+1} = \mu_k^2$.

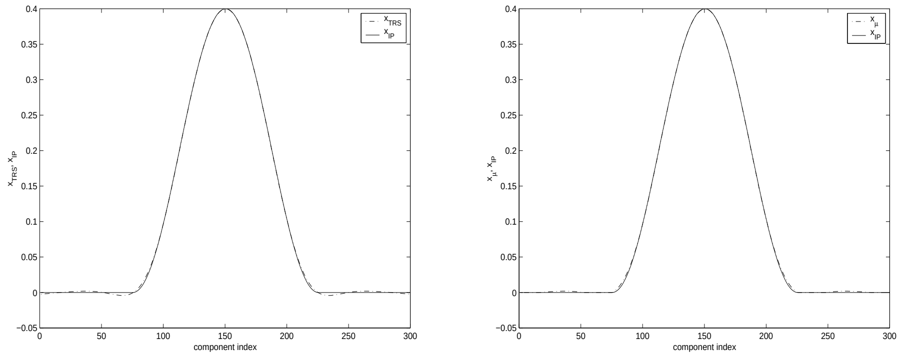
3.3 Accuracy of the regularized solution

In this section we compare the accuracy of regularized nonnegative solutions computed with the method TRUST_μ with respect to regularized solutions that do not take into account the nonnegative constraints. For this purpose, we chose problem **phillips** from [10], with dimensions $m = n = 300$. We again computed x_{TRS}, a solution to the trust-region subproblem (13) by means of LSTRS, and we used this solution to compute $x_0 = x_{TRS}$ as well as to compare the two regularization approaches: with and without the nonnegativity constraints. The eigenvalue problems were solved with `ARPACK`. The tolerances for LSTRS were $\varepsilon_\Delta = 10^{-3}$ and $\varepsilon_{HC} = 10^{-10}$. The tolerances for TRUST_μ were $\varepsilon_y = \varepsilon_f = \varepsilon_x = 10^{-12}$. The update of μ according to (12) was computed with $\sigma = 0.01$.

LSTRS required 5 iterations and 525 matrix-vector products to compute x_{TRS}. This solution has a relative error of order 10^{-2} with respect to x_{IP}. Both x_{IP} and x_{TRS} are shown in Figure 2 (a) (solid and dashed-dotted curves, respectively). As we can observe in Figure 2 (a), and more clearly in a closer look in Figure 3 (a), x_{TRS} has some negative components. Figure 3 (a) also illustrates the well-known Gibbs phenomenon, which is usually seeing with band-pass filters (cf. [16]).

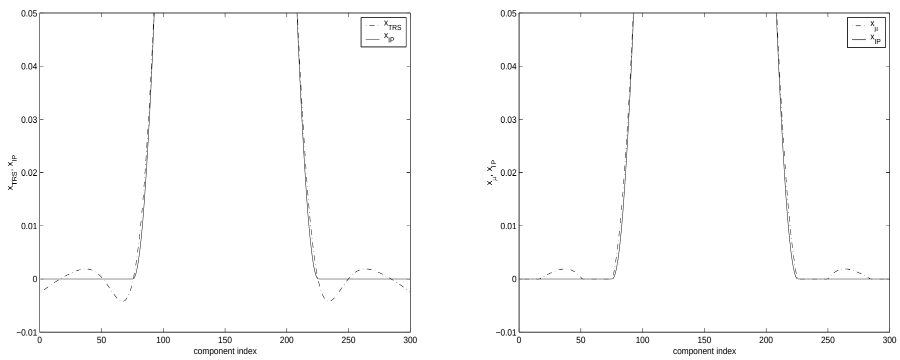
We then applied the method TRUST_μ to solve problem (1) for the same test problem. We used the same LSTRS parameters as before, and also used the initial value for α described in Section 3.1. We obtained the solution x_μ shown in Figure 2 (b) (dashed-dotted curve), which has a relative error of order 10^{-3} with respect to x_{IP}, and has only positive components. Figure 3 (b) shows that the components in x_μ remain positive, and that the ripples are considerably smaller compared to those in Figure 3 (a).

TRUST_μ required 1 iteration of the main loop in Algorithm 2.1 (i.e. 2 calls to LSTRS), 5 LSTRS iterations, and 631 matrix-vector products. The value of μ was of order 10^{-16}. As we can observe the cost is only slightly higher than the one needed to compute the first LSTRS solution, and the result is considerably



(a) (b)

Figure 2: Problem **phillips**, $n = 300$. (a) LSTRS and (b) TRUST$_\mu$ solutions. Relative error in LSTRS solution: 1.0065×10^{-2}. Relative error in TRUST$_\mu$ solution: 6.9218×10^{-3}.



(a) (b)

Figure 3: Problem **phillips**, $n = 300$. Close up of (a) TRS and (b) TRUST$_\mu$ solutions.

more accurate. These results seem to indicate that although solving the problem with nonnegative constraints involves an additional computational cost, such cost is not too high and might be justified by the possibility of obtaining a more accurate regularized solution.

3.4 Image Restoration Problems

In this section we present results on image restoration problems from [12]. In imaging applications such as image restoration, image formation is modeled by an integral equation of the first kind

$$\gamma(s) = \int K(s,t)\phi(t)dt \qquad (15)$$

where the kernel $K(s,t)$ is called the *point spread function* (PSF) that determines the image of a single point source under the imaging system, $\phi(t)$ is the true image, and $\gamma(s)$ is called a blurred version of $\phi(t)$. Imaging systems are called *space invariant* when the PSF acts uniformly across source and image spaces, i.e. a translation in the source space corresponds to a proportional translation on the image space. Most imaging systems can be modeled by a space invariant PSF. If the system is not space invariant it is called *space variant*. Such systems arise for example when recording images of objects that move with different velocities with respect to the recording device.

The discretization of equation (15) yields a linear system of equations

$$Ax = b \tag{16}$$

where $A \in \mathbb{R}^{n\times n}$, and $b \in \mathbb{R}^n$. The matrix A is a discretized version of the *blurring* operator constructed from $K(s,t)$. The vector $b = \bar{b} + \eta$ is a vector representation of a degraded version of the true image by blur and noise, with $\bar{b}$ a discretized version of the blurred image $\gamma(s)$, and η representing noise. Problem (15) is ill-posed, and the discrete problem (16) usually inherits this feature in the sense that the matrix A is highly ill conditioned, with a singular spectrum that decays to zero gradually, and high-frequency components of the singular vectors corresponding to small singular values. In most problems of interest, n is of the order of the tenths of thousands.

3.4.1 Star Cluster

In this section we consider the problem of restoring the image of a star cluster, consisting of simulated data used by astronomers to test image restoration methods for the Hubble Space Telescope (HST). Such methods are needed to restore images recorded with the HST before the mirrors in the camera were corrected. See [14] and the references therein for more details.

The problem is space variant and this should be taken into account when modeling the imaging system as suggested in [14]. Therefore we used a combination of four PSFs described in [14] where the source space is decomposed in four subdomains with a different PSF acting on each of them. The matrix A represents the blurring operator, constructed from that PSF. The discrete problem is of dimension $n = 65536$. The true image, the image recorded by the HST or data image, and the restorations obtained with LSTRS and TRUST$_\mu$ are shown in Figure 4.

The tolerances for LSTRS were $\varepsilon_\Delta = 10^{-2}$ and $\varepsilon_{HC} = 10^{-1}$. The tolerances for TRUST$_\mu$ were $\varepsilon_y = 10^{-2}$, and $\varepsilon_f = \varepsilon_x = 10^{-8}$. The update of μ according to (12) was computed with $\sigma = 0.01$. The initial value was computed as $x_0 = x_{TRS}$, with negative and zero components replaced by 10^{-5}.

LSTRS required 7 iterations and 872 matrix-vector products to compute x_{TRS} in Figure 4 (bottom left), which has a relative error of 1.6743×10^{-1} with respect

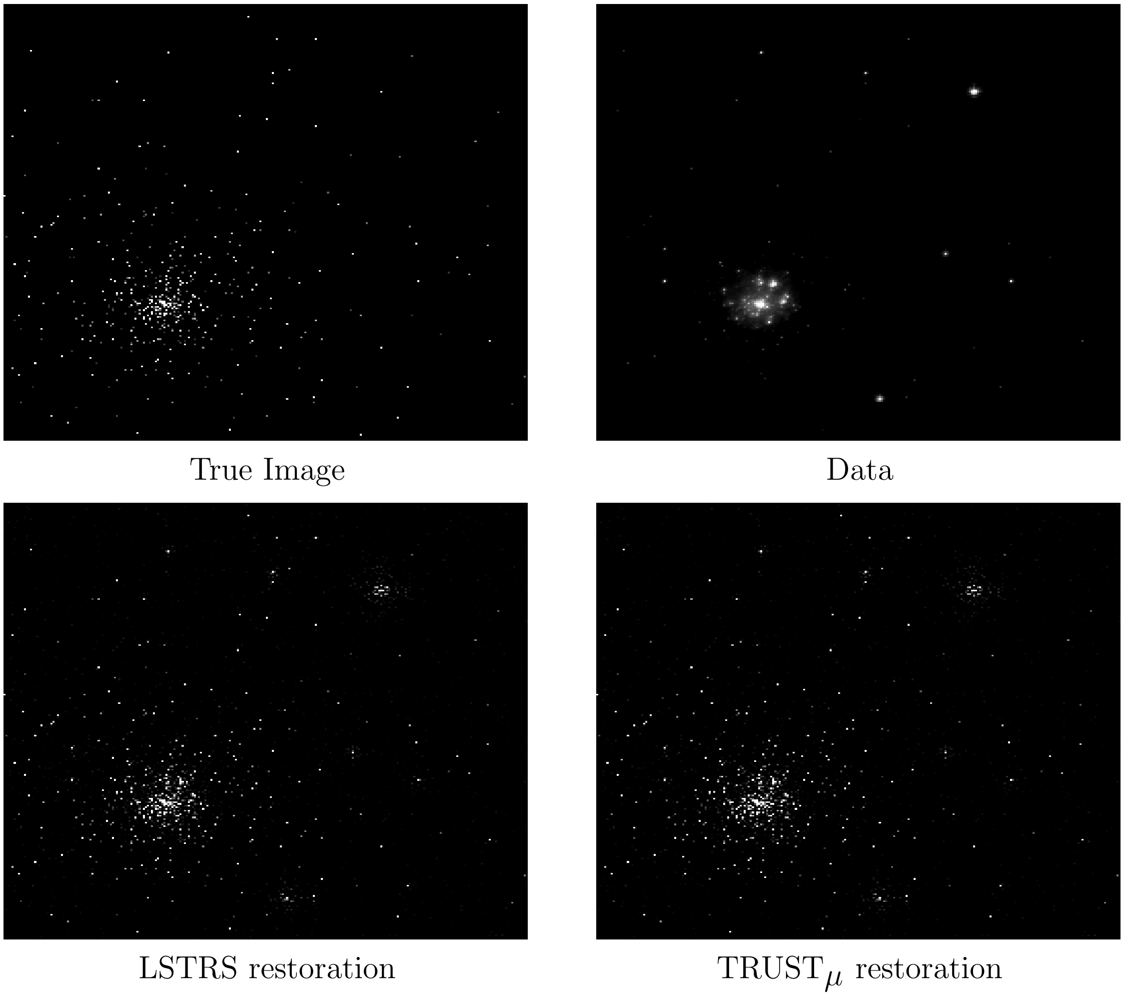


Figure 4: Problem: star cluster, $n = 65336$. Relative error in LSTRS solution: 1.6743×10^{-1}, TRUST$_\mu$ solution: 1.2358×10^{-1}.

to x_{IP}. TRUST$_\mu$ required 1 iteration of the main loop in Algorithm 2.1, 9 LSTRS iterations and 978 matrix-vector products to compute the solution shown in Figure 4 (bottom right), which has a relative error of 1.2358×10^{-1} with respect to x_{IP}, and only positive components. The value of μ was of order 10^{-12}. The storage requirement was 11 vectors of dimension 65536. In this example we observed the same behavior as in Section 3.3, namely, with a moderate additional cost over the trus-region solution, we computed a positive solution with improved accuracy.

3.4.2 Satellite

In this section we present another example from astronomical imaging: restoring the image of a satellite in space. The problem was developed at the US Air Force Phillips Laboratory, Laser and Imaging Directorate, Kirtland Air Force Base New Mexico, and is available from [12]. More details about the problem can be found in [9] and the references therein. The imaging system is again modeled as an integral equation of the first kind (15), and is space invariant. Therefore only one PSF was used to construct the blurring operator. The discretized problem is of dimension $n = 65536$.

We first used LSTRS to compute the initial values as before. In this case the strategy produced a not too clear solution and required 12 iterations and 1363 matrix-vector products. We believe that the performance of LSTRS can be greatly improved by investigating the features of this problem. We have observed in other applications (cf. [18], [20]) that certain severely ill-conditioned problems benefit from changing the norm constraint in (13) to the form $\|Lx\|$, where L is a discretized form of first derivative or a scaling matrix. We did not pursue these options in this work since our purpose was a preliminary study of the method TRUST$_\mu$ and not of the particular test problems.

We then tried the following alternative to compute initial values for TRUST$_\mu$. Since in the previous example, an interior solution was detected by LSTRS, we decided to compute the initial values by first computing x_{LS}, the solution to the unconstrained least squares problem. We then set $x_0 = |x_{LS}|$ with zero components replaced by 10^{-5}. Since x_{LS} corresponds to an interior solution of the trust-region subproblem (13), we set $\lambda = 0$, and $\alpha = -g^T x_{LS}$. The tolerances for TRUST$_\mu$ and the value of σ were as in Section 3.4.1.

We used the Conjugate Gradient Method on the Normal Equations (CGLS) [2], [8] to solve the normal equations to a prescribed accuracy of the least squares residual. CGLS required 51 matrix-vector products and so did TRUST$_\mu$, because in this case the initial iterate x_0 already satisfied the stopping criteria. The value of μ was of order 10^{-16}. The storage requirement was 5 vectors of dimension 65536. The results are shown in Figure 5.

4 Conclusions

We presented the method TRUST$_\mu$ for large-scale nonnegative regularization. The method combines interior-point and trust-region strategies to solve a quadratic problem with a norm constraint and nonnegativity constraints. The method is not based on a heuristic, and it does not depend on the availability of a preconditioner. The method relies only upon matrix-vector products with the Hessian matrix, and has low and fixed storage requirements.

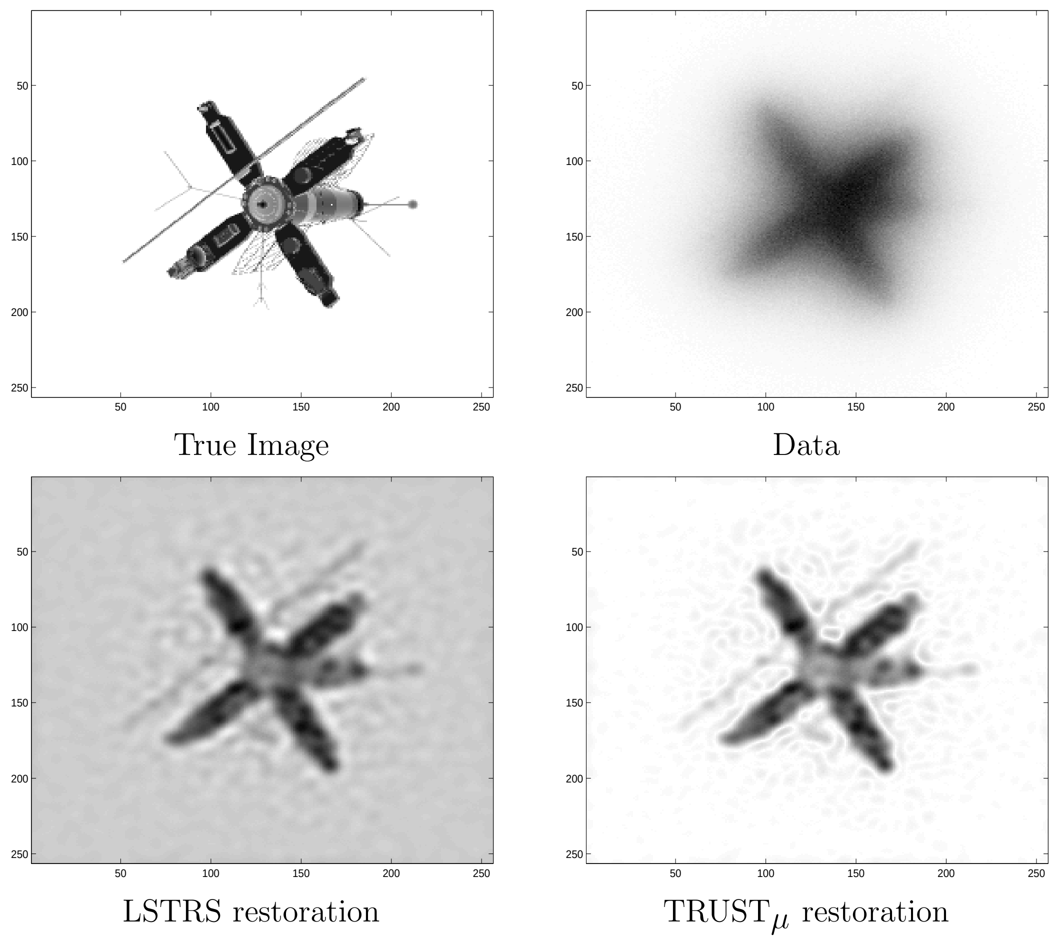


Figure 5: Problem: satellite, $n = 65536$. Relative error in LSTRS solution: 0.3599, TRUST$_\mu$ solution: 0.3585.

We used our method TRUST$_\mu$ to compute regularized nonnegative solutions to inverse problems, including test problems from astronomical imaging. For the problems considered, TRUST$_\mu$ computed positive restorations with moderate computational cost and improved accuracy. Although more experiments are needed to assess the effectiveness of the method, the initial results are encouraging and present TRUST$_\mu$ as a promising method for large-scale nonnegative regularization.

We remark that although the main motivation for developing the method were constrained least squares problems of type (1), the derivation of the method was independent of the least squares problem (1) and therefore, the method can be used for solving any kind of quadratically and nonnegatively constraint quadratic problems.

Curent work includes the study of the theoretical properties of the method, in particular, the convergence rate. Future work should include a comparison with the techniques mentioned in Section 1.

Acknowledgements. We would like to thank Jim Nagy for providing the examples in Section 3.4. We also thank Per Christian Hansen for providing valuable references, and for very helpful comments on an earlier version of this manuscript.

References

[1] M. Bertero and P. Bocacci. *Introduction to Inverse Problems in Imaging*, Institute of Physics, Bristol, 1998.

[2] Å. Björck. *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.

[3] T.J. Carpenter and D.F. Shanno. An interior point method for quadratic programs based on conjugate projected gradients, *Comput. Optim. Appl.*, 2:5-28, 1993.

[4] G.D. de Villiers, B. McNally and E.R. Pike. Positive solutions to linear inverse problems, *Inverse Problems*, 15:615-635, 1999.

[5] G.C. Fehmers, L.P.J. Kamp and F.W. Sluijter. An algorithm for quadratic optimization with one quadratic constraints and bounds on the variables, *Inverse Problems*, 14:893-901, 1998.

[6] A.V. Fiacco and G.P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization*, Classics in Applied Mathematics series, SIAM, Philadelphia, 1990.

[7] D. Gay. Computing optimal locally constrained steps, *SIAM J. Sci. Stat. Comput.*, 2(2):186-197, 1981.

[8] G.H. Golub and C.F. Van Loan. *Matrix Computations*, Johns Hopkins University Press, Baltimore, third ed., 1996.

[9] M. Hanke, J.G. Nagy and C. Vogel. Quasi-Newton Approach to Nonnegative Image Restorations. *Lin. Alg. Appl.*, 316:223-236, 2000.

[10] P.C. Hansen. Regularization Tools: a MATLAB package for analysis and solution of discrete ill-posed problems, *Numer. Algo.*, 6:1-35, 1994.

[11] P.C. Hansen, Rank–Deficient and Discrete Ill–Posed Problems, Numerical Aspects of Linear Inversion, SIAM, Philadelphia, 1997.

[12] P.C. Hansen and J.G. Nagy. RESTORE TOOLS: Matlab tools for image restoration. In preparation.

[13] R.B. Lehoucq, D.C. Sorensen and C. Yang. *ARPACK User's Guide: Solution of Large Scale Eigenvalue Problems by Implicitly Restarted Arnoldi Methods*, SIAM, Philadelphia, 1998.

[14] J.G. Nagy and D.P. O'Leary. Restoring images degraded by spatially-variant blur. *SIAM J. Sci. Comput.*, 19: 1063-1082, 1998.

[15] J.G. Nagy and Z. Strakos. Enforcing nonnegativity in image reconstruction algorithms. In David C. Wilson, et al, editors, *Mathematical Modeling, Estimation, and Imaging*, 4121: 182-190, 2000.

[16] T.W. Parks and C.S. Burrus, *Digital Filter Design*, John Wiley & Sons, 1987.

[17] M. Piana and M. Bertero. Projected Landweber method and preconditioning, *Inverse Problems*, 13:441-464, 1997.

[18] M. Rojas. A Large-Scale Trust-Region Approach to the Regularization of Discrete Ill-Posed Problems. Ph.D. Thesis, Technical Report TR98-19, Department of Computational and Applied Mathematics, Rice University, Houston, May 1998.

[19] M. Rojas, S.A. Santos and D.C. Sorensen. A new matrix-free algorithm for the large-scale trust-region subproblem, *SIAM J. Optim.*, 11(3):611-646, 2000.

[20] M. Rojas and D.C. Sorensen. A trust-region approach to the regularization of large-scale discrete forms of ill-posed problems, *SIAM J. Sci. Comput.*, to appear.

[21] D.C. Sorensen. Newton's Method with a Model Trust Region Modification, *SIAM J. Numer. Anal.*, 19(2):409-426, 1982.

[22] D.C. Sorensen. Implicit application of polynomial filters in a k-step Arnoldi method, *SIAM J. Matrix Anal. Appl.*, 13(1):357-385, 1992.

[23] A.N. Tikhonov, A.V. Goncharsky, V.V. Stepanov and A.G. Yagola. *Numerical methods for the solution of ill-posed problems*, Kluwer, Dordrecht, 1995.

[24] Y. Ye. *Interior point algorithms: theory and analysis*, Wiley-Interscience, New York, 1997.

[25] D.C. Youla and H. Webb. Image restoration by the method of convex projections: part 1 - theory, *IEEE Trans. Med. Imag.*, MI-1(2): 81-94, October 1982.