

## **Paper B**

# **Solving a TRS which has linear inequality constraints**

This is a draft of our paper which has not yet been published.



where

$$\begin{aligned} f_\mu(x) &= \frac{1}{2}x^T Hx + g^T x + \frac{1}{2}b^T b - \mu c(x) \\ \nabla f_\mu(x) &= Hx + g - \mu \nabla c(x) \\ \nabla^2 f_\mu(x) &= H - \mu \nabla^2 c(x) \end{aligned}$$

and

$$\begin{aligned} c(x) &= \sum_{i=1}^m \log(d_i - \sum_{j=1}^n C_{ij}x_j) \\ \nabla c(x) &= -C^T \text{diag}(d - Cx)^{-1} e \\ \nabla^2 c(x) &= -C^T \text{diag}(d - Cx)^{-2} C \end{aligned}$$

yielding a quadratic objective function in the unknown  $h$ , for a given  $x$ . A solution to problem (3) can then by equation (4) be approximated by solving

$$\begin{aligned} \min_h q_\mu(x+h) \\ \text{s.t. } \|x+h\| \leq \Delta \end{aligned} \quad (5)$$

when  $h$  is small. Expanding the terms from equation (4) we get

$$\begin{aligned} q_\mu(x+h) &= f_\mu(x) + \nabla f_\mu(x)^T h + \frac{1}{2}h^T \nabla^2 f_\mu(x) h \\ &= \frac{1}{2}x^T Hx + g^T x + \frac{1}{2}b^T b - \mu c(x) \\ &\quad + (Hx + g - \mu \nabla c(x))^T h + \frac{1}{2}h^T (H - \mu \nabla^2 c(x)) h \end{aligned}$$

Now consider the function

$$p_\mu(x+h) = \frac{1}{2}(x+h)^T (H - \mu \nabla^2 c(x))(x+h) + (\mu \nabla^2 c(x)x - \mu \nabla c(x) + g)^T (x+h)$$

and observe that

$$p_\mu(x+h) = q_\mu(x+h) + p_\mu(x) - f_\mu(x)$$

Since we keep  $x$  fixed and minimize in terms of  $h$ , the  $p_\mu(x) - f_\mu(x)$  term is a constant which can be ignored. We can therefore use  $p_\mu$  instead of  $q_\mu$  when solving problem (5). We now set  $z = x+h$  and this gives a final TRS problem which approximates a solution to problem (1)

$$\begin{aligned} \min_z \frac{1}{2}z^T (H - \mu \nabla^2 c(x))z + (\mu \nabla^2 c(x)x - \mu \nabla c(x) + g)^T z \\ \text{s.t. } \|z\| \leq \Delta \end{aligned} \quad (6)$$

Finally, it can be shown that the following first order KKT conditions is necessary and sufficient for a feasible point  $z$  and scalar Lagrange multiplier  $\lambda$  to be a solution to problem (6)

$$\begin{aligned} (H - \mu \nabla^2 c(x) + \lambda I)z &= -\mu \nabla^2 c(x)x + \mu \nabla c(x) - g \\ H - \mu \nabla^2 c(x) + \lambda I &\succeq 0 \\ \lambda(\|z\|^2 - \Delta^2) &= 0 \\ \lambda &\geq 0 \end{aligned} \quad (7)$$



# Solving a TRS which has linear inequality constraints

Ørjan Bergmann\* and Trond Steihaug\*

October 21, 2007

## Abstract

In this work we present a new algorithm for solving the augmented *trust-region subproblem* with a set of additional linear inequality constraints. The method can be considered a generalization of previous published methods [10]. We discuss the additional types of problems our formulation can solve and reproduce regularized large-scale inverse-problem results from image processing in our general framework.

*Keywords:* constrained optimization; regularization; image processing

## 1 Introduction

The trust-region subproblem (TRS), find a point that minimizes a quadratic function subject to remaining inside an ellipsoidal region, is an important problem in optimization and linear algebra. Frequently an application motivated TRS will require that the solution satisfy an additional collection of linear constraints. In this case the problem can be written as

$$\begin{aligned} \min_x \quad & \frac{1}{2} \|Ax - b\|^2 \\ \text{s.t.} \quad & Cx \leq d \\ & \|Sx\| \leq \Delta \end{aligned} \tag{1}$$

where  $A \in \mathbb{R}^{\hat{m} \times n}$ ,  $C \in \mathbb{R}^{m \times n}$ ,  $S \in \mathbb{R}^{n \times n}$ ,  $x \in \mathbb{R}^n$ ,  $d \in \mathbb{R}^m$ ,  $b \in \mathbb{R}^{\hat{m}}$  and  $\Delta$  is a positive scalar. We let  $\|\cdot\|$  denote the Euclidean norm. The matrix  $S$  is usually selected to scale the problem appropriately and therefore is frequently a diagonal matrix. For simplicity we will assume that a linear transformation has been employed to the variables  $x$  so that the matrix  $S$  is the identity matrix without loss of generality (i.e.  $S = I$ ).

Problems on the form of equation (1) arise in many scientific and engineering fields, such as regularization of certain ill-posed problems. Among such applications, the regularization of large-scale ill-posed inverse problems in image processing will be the topic of interest in this paper. Linear discrete ill-posed problems are numerically under-determined because of the clustering of the singular values of  $A$  at the origin. Therefore numerical methods that impose known properties of the true solution on the computed approximate solution typically produce better approximations than those that

---

\*Institute for Informatics, University of Bergen, Pb 7800, N-5020 Bergen, NORWAY

do not. In image processing such known properties are often the component-wise non-negativity of the solution. Large linear discrete ill-posed problems with a non-negative solution arise for instance in image restoration where the entries of  $x$  corresponds to pixel values, which are non-negative.

In literature different techniques for solving the TRS with non-negativity constraints has been proposed. The methods proposed in [2] are based on truncated singular-value decomposition regularization. The approach can be used on large-scale regularization problems by computing only a few singular values, although determining exactly how many is in general difficult. The authors report results for small problems only. The methods in [7] are iterative methods for linear systems that impose a non-negativity constraint at each step. Regularization is achieved by early termination of the iteration. In practice it is difficult to determine good termination criteria for stopping the iteration and the method requires preconditioners to obtain competitive results. Finally, [3] propose enforcing the non-negativity constraint by rewriting the variables of the regularization problem as  $x = e^z$ , and then solve the resulting problem in terms of the new variable  $z$ . The resulting problem is then solved with different regularization formulations using a Newton-based iterative algorithm. The results are promising but the authors point out that effective preconditioners would further improve the results. Additionally it is not clear how the method can be generalized to incorporate other types of constraints.

In terms of equation (1) a non-negativity constraint can be modeled by letting  $C = -I$  and  $d = 0$ . Indeed, this is precisely the formulation that Rojas and Steihaug solves [10]. By approximating the Lagrangian relaxation with a second order Taylor polynomial, the non-negatively constrained TRS problem can be approximated by a different TRS on standard form [10, 4, 6, 1]. Solving this large-scale ill-posed TRS can then be attempted using trust-region based techniques.

Motivated by this recent and successful work on the solution of the TRS with non-negativity constraint, we will in this work solve the more general problem (1) where  $C$  and  $d$  can be an arbitrary matrix and vector respectively, using similar techniques. In image processing this allows us to impose even more *a priori* knowledge about the properties of the true solution on the regularized solution, either in the whole image or in parts of it:

- We can impose both *upper* and *lower* bounds on the solution.
- We can impose constraints on the *mean* of the whole or in certain smaller regions of the image, possibly compared to other regions in the same image or a scalar quantity estimated from other sources.
- We can impose limitations on certain measures of *deviations* from the mean in regions.
- We can impose constraints on the change in intensity (i.e. the *gradient*) from one pixel or region in the image to the next in parts or in the whole solution image.

By enforcing such constraints we hope that we will be able to compute a solution which is closer to the true undegraded image.

## 2 The method

We observe that the linearly constrained TRS in equation (1) always has a solution and that this solution is unique when  $A$  has full rank. We will now derive optimality

conditions that a solution to this problem must satisfy. The Lagrangian functional associated with the problem is

$$\begin{aligned} \mathfrak{L}(x, \lambda, y) &= \frac{1}{2}x^T A^T A x - (A^T b)^T x + \frac{1}{2}b^T b \\ &\quad - \frac{\lambda}{2}(\Delta^2 - \|x\|^2) - y^T (d - Cx) \end{aligned}$$

where  $\lambda \in \mathbb{R}$  and  $y \in \mathbb{R}^m$  are multipliers. The gradient of this functional with respect to  $x$  is

$$\nabla_x \mathfrak{L}(x, \lambda, y) = A^T A x - A^T b + \lambda I x + C^T y$$

Suppose that  $x$  is a local solution to problem (1). Then the first order necessary Karush-Kuhn-Tucker (KKT) conditions [8] states that here exists multipliers  $\lambda$  and  $y$  such that

$$(A^T A + \lambda I)x = A^T b - C^T y \quad (2a)$$

$$\lambda(\Delta^2 - \|x\|^2) = 0 \quad (2b)$$

$$y^T (d - Cx) = 0 \quad (2c)$$

$$\lambda \geq 0, y \geq 0$$

Observe that when  $A^T A$  is positive definite then the objective function in problem (1) is strictly convex and the KKT conditions are both necessary *and* sufficient. We will later use the KKT expressions to find initial values for the algorithm and to update the iterates. In order to solve our problem (1) we follow Rojas and Steihaug [10] and simplify by relaxing the  $Cx \leq d$  constraint and introduce a logarithmic penalty term to the objective function which penalizes when  $d - Cx$  approaches zero. This yields the modified problem

$$\begin{aligned} \min_x \quad & \frac{1}{2}\|Ax - b\|^2 - \mu \sum_{i=1}^m \log(d_i - \sum_{j=1}^n C_{ij}x_j) \\ \text{s.t.} \quad & \|x\| \leq \Delta \end{aligned} \quad (3)$$

This relaxation transforms our original problem into an *interior point barrier problem* implying  $Cx < d$  in which the  $\mu$  parameter controls the penalty imposed by approaching the borders  $Cx = d$  of the feasible region. The *central path* of this problem is defined as the set of minimizers  $x(\mu)$  for different choices of the penalty parameter  $\mu$ . A standard result from the theory of barrier algorithms guarantees that the unique global minimizer  $x^*$  of problem (1) is the limit of the central path [8], i.e.

$$x^* = \lim_{\mu \searrow 0} x(\mu)$$

Unfortunately, exact minimization of equation (3) cannot in general be completed in finite time, but it is well known [4] that it is not necessary to solve the minimization exactly in order to guarantee convergence to  $x^*$ . We will in the following therefore only approximate a solution to the barrier problem as we for each iteration decrease  $\mu > 0$ .

So as to simplify our notation and for clarity of presentation we now define  $H = A^T A$  and  $g = -A^T b$ . As in [10, 4, 1, 6] we then approximate the objective function of the barrier problem with a second-order Taylor-polynomial

$$q_\mu(x + h) = f_\mu(x) + \nabla f_\mu(x)^T h + \frac{1}{2}h^T \nabla^2 f_\mu(x) h \quad (4)$$

Our method consists of solving a sequence of problems on the form of equation (6) for different values of  $x$  and  $\mu$ , while driving the barrier parameter  $\mu$  towards zero and ensuring that the linear constraint  $Cx \leq d$  are satisfied.

### 3 The algorithm

Based on the previous section, we propose algorithm `lincontrs` for solving problem (1) with  $H = A^T A$  and  $g = -A^T b$ . This top-level algorithm is similar to the one

---

**Algorithm 1** `lincontrs`( $H, g, \Delta, C, d, x_0$ )

---

**Require:**  $H$  symmetric positive semidefinite,  $0 \leq \|x_0\|_2 < \Delta$  and  $Cx_0 < d$

- 1: Choose initial  $\mu_0 \in \mathbb{R}^+$  and set  $k = 0$
  - 2: **while** not convergence **do**
  - 3:   Solve equation (6) with respect to the unknown  $z_k$
  - 4:   Set  $h_k \leftarrow x_k - z_k$
  - 5:   Calculate  $\beta_k^*$  such that  $C(x_k + \beta_k^* h_k) < d$
  - 6:   Set  $x_{k+1} \leftarrow x_k + \beta_k^* h_k$
  - 7:   Calculate  $\mu_{k+1} < \mu_k$
  - 8:   Set  $k \leftarrow k + 1$
  - 9: **end while**
- 

presented by Rojas and Steihaug [10] and later Kearsley [4], Morigi *et al* [6], and Calvetti *et al* [1] although the details of how we implement some of the steps differ significantly. We will point out these differences as we describe the algorithm below.

#### 3.1 Choosing initial values of the multipliers

An initial estimate of the solution vector  $x_0 \in \mathbb{R}^n$  that lies strictly on the interior of the feasible region (i.e. that satisfies  $Cx_0 < d$  and  $\|x_0\| < \Delta$ ) must be supplied to the algorithm by the user. From equation (2b) we then see that  $\lambda = 0$ , so equation (2a) gives

$$-C^T y = Hx_0 + g$$

which can be solved by minimizing the linear least-squares problem

$$\min_y \|C^T y + (Hx_0 + g)\|$$

For a given  $y$ , the scalar  $\mu_0$  required in line 1 of the algorithm can be found as will be discussed in a later section. This enables us to calculate the gradient and the Hessian needed to solve the trust-region subproblem in line 3.

In non-negative regularization [10, 1, 6] the problem (1) is often first solved as a standard TRS, disregarding the linear non-negativity constraint. This allows an estimate of the dual variable  $\lambda_{\text{TRS}}$  to be calculated. From  $\lambda_{\text{TRS}}$  an estimate of  $y$  is then computed using the same set of equations as here. The solution  $x_{\text{TRS}}$  to the TRS problem is then projected into the feasible region before the algorithm proceeds. Since the linear constraint only impose non-negativity such a projection is easy and is implemented by replacing negative intensities with small positive values. In our case of general linear inequality constraints this projection may be non-trivial. Furthermore, if the TRS solution  $x_{\text{TRS}}$  lies far from the linearly constrained feasible region then  $\lambda_{\text{TRS}}$  may be a poor estimate of the dual of the projected TRS solution.



### 3.2 Solve the TRS

There exists efficient algorithms for solving the large-scale trust-region subproblem in line 3 of the algorithm

$$\begin{aligned} \min_x \quad & \frac{1}{2}x^T Qx + r^T x \\ \text{s.t.} \quad & \|x\| \leq \Delta \end{aligned}$$

where  $Q$  is a symmetric positive semidefinite  $n \times n$  matrix and  $r$  is a  $n$ -element vector. One such algorithm presented by Rojas *et al* in [9] is implemented in the Matlab package `lstrs`. This package solves the TRS as a parameterized eigenvalue problem of the form

$$\begin{bmatrix} Q & r \\ r^T & \theta \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix} = -\lambda \begin{bmatrix} x \\ 1 \end{bmatrix} \quad (8)$$

where  $\begin{bmatrix} x & 1 \end{bmatrix}^T$  is the eigenvector corresponding to the eigenvalue  $-\lambda$ . This package was used for solving the TRS problem in both [10] and [4].

The basic idea of the eigenvalue based approach is to solve this eigenvalue-problem for various choices of  $\theta$  until a solution is obtained which can be normalized to  $\begin{bmatrix} x & 1 \end{bmatrix}^T$  such that  $x$  satisfies the KKT conditions for optimality of the TRS. This procedure does not rely on matrix factorizations and are thus “matrix free”, which makes it highly suitable for large-scale problems.

Another advantage with this type of method is the fact that they provide a simple way to detect and handle the hard case which usually occur in regularization; in this framework the hard case manifests itself as an eigenvector which has zero in the last component. Obviously this vector cannot be normalized to be on the form needed, but in such cases it can be shown that there exists an eigenvalue slightly larger than  $\lambda_1$  which has an eigenvector with non-zero last component. This can then be used to construct a *quasi-optimal* solution  $\hat{x}$  instead [9].

Other approaches for solving the TRS are of course also possible. Landi [5] use a *truncated conjugate gradient* based method proposed by Steihaug [8]. Morigi [6] use a variant of the LSQR algorithm proposed by Björck [11] which is an implementation of the conjugate gradient method applied to the normal equations. Finally Calvetti [1] develop their own algorithm that solves the TRS with equality constraints by searching for the roots of the function  $\|x(\tau)\| - \Delta = 0$ , where  $x(\tau)$  is an approximation to the Thikonov regularized solution and  $\tau$  is a regularization parameter. It should be noted however that all of these methods may have problems associated with the hard case which usually occur in regularization problems.

### 3.3 The line search

Notice that the approximate solution  $z$  to the TRS problem computed in line 3 of the algorithm may not be feasible given the linear constraints, i.e.  $Cz_k > d$ . One approach used by Rojas and Steihaug [10] is then to estimate an update  $h_k = z_k - x_k$  and then take a step of length  $\beta$  away from the feasible point  $x_k$  in direction  $h_k$ . The scalar step length  $\beta$  must then be chosen so that the next iterate  $x_{k+1}$  in line 5 of the algorithm is feasible given the linear constraints. We then get

$$\begin{aligned} x_{k+1} &= \beta_k z_k + (1 - \beta_k)x_k \\ &= x_k + \beta_k h_k \end{aligned}$$

where  $\beta_k \geq 0$  is chosen so that  $Cx_{k+1} < d$ . Some recent publications [4, 5] has suggested using an Armijo-type line search in order to estimate the step-length. We will follow [10, 6] and choose  $\beta_k^*$  as large as possible given the constraints i.e. we define

$$\begin{aligned} \beta_k^* &= \max \beta_k \\ \text{s.t. } \beta_k Ch_k &\leq d - Cx_k \end{aligned}$$

If any  $(d - Cx_k)_j > (Ch_k)_j$  then the solution to this problem is

$$\beta_k^* = \min \left\{ \frac{(d - Cx_k)_i}{|(Ch)_i|} \right\} \quad (9)$$

where  $i \in \{j : (d - Cx_k)_j > (Ch_k)_j\}$ , otherwise we define  $\beta_k^* = 1$ . We then define the next iterate as

$$x_{k+1} = x_k + \min\{1, 0.995\beta_k^*\}h_k \quad (10)$$

This ensures that  $x_{k+1}$  is close to the TRS solution  $z_k$  while simultaneously ensuring that  $x_{k+1}$  is on the interior of the linearly constrained feasible region.

### 3.4 Updating $\mu$

As pointed out in section 2, in order for the algorithm to converge to a solution to problem (1) the penalty parameter  $\mu$  must be iteratively decreased towards zero. We will now consider formulas for updating  $\mu$  needed in line 7 of the algorithm. Kearsley [4] and Morigi [6] use the simple update used in classical barrier algorithms of letting

$$\mu_{k+1} = \frac{\mu_k}{M}$$

where  $M > 1$  is a user specified constant. We develop a more elaborate scheme similar to the one proposed by Rojas and Steihaug [10] in order to improve the rate of convergence of the algorithm. First note from equation (2a) that

$$C^T y = -(H + \lambda I)x - g$$

and from equation (7)

$$-(H + \lambda I)z - g = \mu \nabla^2 c(x)(x - z) - \mu \nabla c(x)$$

When  $(H + \lambda I)x \approx (H + \lambda I)z$  then we have

$$\begin{aligned} C^T y &\approx \mu \nabla^2 c(x)(x - z) - \mu \nabla c(x) \\ &= \mu C^T \text{diag}(d - Cx)^{-2} C(z - x) + \mu C^T \text{diag}(d - Cx)^{-1} e \end{aligned}$$

From this we define  $\hat{y}$  as an approximation to the Lagrangian multiplier  $y$  so that

$$\hat{y} = \mu \text{diag}(d - Cx)^{-2} C(z - x) + \mu \text{diag}(d - Cx)^{-1} e \quad (11)$$

and note that when  $z = x$  we have

$$\hat{y} = \mu \text{diag}(d - Cx)^{-1} e \quad (12)$$

From the duality gap in equation (2c) and using our last expression for  $\hat{y}$  we have

$$\begin{aligned}\hat{y}^T(d - Cx) &= (d - Cx)^T \mu \text{diag}(d - Cx)^{-1} e \\ &= \mu e^T e \\ &= \mu m\end{aligned}$$

Therefore we will estimate  $\mu$  as

$$\mu = \frac{\sigma}{m} \hat{y}^T(d - Cx) \quad (13)$$

where  $\sigma$  is a constant chosen so that  $\mu_k > \mu_{k+1}$  and  $\hat{y}$  is calculated from equation (11).

It is worth noting here that the approximation  $\hat{y}$  used by Rojas and Steihaug [10], and later by Calvetti [1], assumes  $z = x$  and therefore equation (12) is used for approximating the dual variables instead of equation (11).

## 4 Numerical results

In the next sections we will now show the performance of our presented `lincontrs` algorithm for two very different test cases from different applications.

### 4.1 The star cluster

We now consider an example inspired by the Hubble Space Telescope (HST) which experienced a faulty mirror which blurred the acquired images the first period of operation. It is possible to solve this problem as a TRS where  $b$  is the measured blurry data, where  $A$  represents the blurring operator, and where regularization is achieved through a constraint on the norm of the solution. However, employing this approach leads to the undesirable effects of negative intensities in the solution. Since each intensity is a measurement of the amount of light at that location, negative intensities has no physical meaning.

In literature this TRS has been solved with an additional non-negativity constraints in order to avoid undesirable solutions. This approach has produced results which are closer to the known true solution than other approaches, e.g. truncation of negative intensities to zero [10]. In terms of equation (1), the non-negativity constraint can be implemented by letting  $C = -I$  and  $d = 0$ . Indeed, this is the exact formulation that Rojas and Steihaug solve in 2002. We will now solve the same problem, using our generalization of their approach, for comparison.

As mentioned we have access to the true undegraded image  $x^*$  for reference. The minimum intensity of this undegraded image is on the order of  $10^{-11}$ , the maximum is 31651.05, with a mean of 6.36. The image consist mainly of black background with a few bright stars of high intensity. The noisy observation  $b$  available to us has been degraded by a known blurring kernel giving the poorly conditioned  $A$  matrix of the model. The norm of the true solution  $\Delta = \|x^*\|_2$  is also available to us. Finally, the noisy image has a relative error compared to the undegraded original  $x^*$  of  $\frac{\|b-x^*\|_2}{\|x^*\|_2} = 0.91073$ .

Solving this problem as a TRS without any non-negativity constraints we arrive at a solution with a relative error of 0.16135. We solve this problem using the previously described LSTRS package (version 1.2) in Matlab (version 7.4). Obtaining this solution requires 740 matrix-vector multiplication between  $A$  (or  $A^T$ ) and a vector.

Careful inspection of this solution reveals that near the stars we find some pixels which have negative intensity in the restoration; the minimum value of the restored image is  $-563.16$ .

Using our algorithm to solve the problem with a non-negativity constraint (i.e.  $C = -I$  and  $d = 0$ ) we arrive at a solution with a relative error of 0.11640. The minimum intensity in the reconstruction is on the order of  $10^{-6}$  and the maximum is 31181.39. Calculating this solution requires initialization and one iteration of the main loop, using a grand total of 934 matrix-vector products.

This result is comparable to the one published in [10] in which the published  $\text{TRUST}_\mu$  algorithm terminates after one main iteration using a grand total of 973 matrix-vector products with a relative error of 0.12358. Our estimated solution is therefore marginally more accurate, and requires slightly fewer matrix-vector products in this example. We expect that these discrepancies can be explained by different versions of both LSTRS and Matlab, used in our implementation, compared to theirs.

## 4.2 The circus tent

In this section we solve the *circus tent* example which is documented in the Optimization toolbox in Matlab, as an example of “large-scale quadratic programming”. The following introduction is given in the Matlab documentation:

“Imagine building a circus tent to cover a square lot. The tent has five poles that will be covered with an elastic material. From this structure, we want to find the natural shape of the tent. This natural shape corresponds to the minimum of a certain energy function computed from the surface position and squared norm of its gradient.”

Specifically, the energy function that we will minimize is a quadratic function of 900 variables, where each variable represents the distance from the ground level to the circus tent ceiling on a uniform  $30 \times 30$  grid. The Hessian of this problem is sparse and has a near diagonal structure. The 5 poles which hold up the elastic ceiling gives lower bounds around 5 locations.

The `quadprog` routine in the Matlab Optimization Toolbox is able to solve this problem quite efficiently. Internally it uses a subspace trust-region method based in the interior-reflective Newton method. Each iteration involves the approximate solution of a large linear using the method of preconditioned conjugate gradients (PCG). After 14 such iterations employing 158 CG iterations the Matlab routine calculates a solution to the problem which has a norm of 5.83098.

When solving the problem with our `lincontrs` method we choose  $\Delta$  large enough to not be an active constraint in the solution since the problem is not ill conditioned and does not require regularization. This makes it easy to calculate the interior solutions of the TRS using the CG algorithm. We use the linear inequality constraints to enforce that each variable is greater than the bounds specified by the circus tent poles and the ground at level zero. We use the same initial solution  $x_0 = 0.6$  as we used in `quadprog`. After 15 iterations of our algorithm we arrive at a solution with a norm of 5.83106. This solution is visualized in figure 1. The norm of the difference between our solution and the `quadprog` solution is  $5.79867 \cdot 10^{-4}$ . Based on this we conclude that for this example the performance of our algorithm is comparable to the state-of-the-art optimization algorithms provided in the Optimization Toolbox in Matlab.

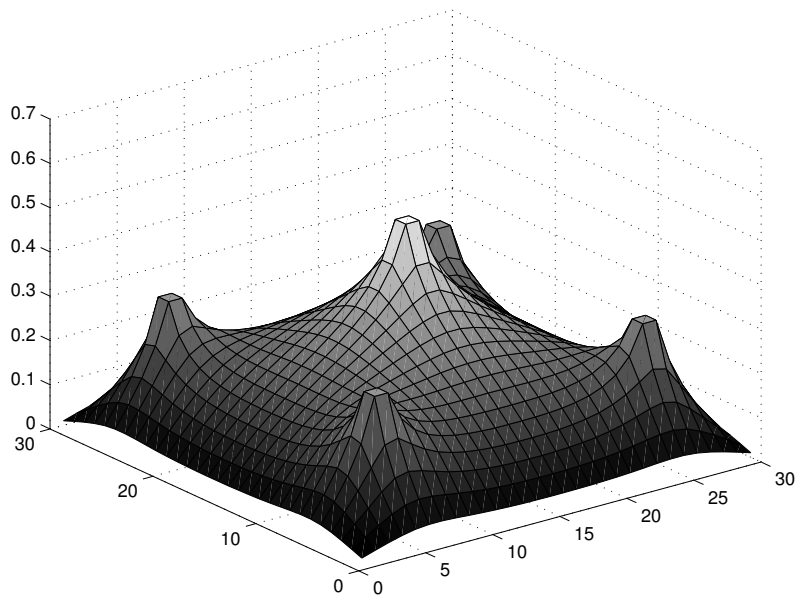


Figure 1: The circus tent example.

## 5 Conclusion

We have in this work presented a new algorithm which is a generalization of previously published works by Rojas and Steihaug [10]. We compare the results of our algorithm with those obtained using state-of-the-art algorithms from different areas and show that our algorithm can solve both these problems accurately and efficiently in our generalized formulation.

## References

- [1] D. Calvetti, B. Lewis, L. Reichel, and F. Sgallari. Tikhonov regularization with nonnegativity constraint. 2004.
- [2] G.D. de Villiers, B. McNally, and E.R. Pike. Positive solutions to linear inverse problems. *Inverse Problems*, 15:615–35, 1999.
- [3] M. Hanke, J. G. Nagy, and C. Vogel. Quasi-newton approach to nonnegative image restoration. *Linear Algebra and its applications*, 316:223–236, 2000.
- [4] Anthony J. Kearsley. A matrix free algorithm for the large-scale constrained trust-region subproblem. *Optimization Methods and Software*, 21(2):233–245, 2006.
- [5] G. Landi and E. Loli Piccolomini. A feasible direction method for optimization with one quadratic constraint and applications to image denoising. Technical report, Department of Mathematics, University of Bologna, April 2007.

- [6] S. Morigi, L. Reichel, and F. Sgallari. A barrier function method for large constrained discrete ill-posed problems. Technical report, Department of Mathematics, University of Bologna, 2007.
- [7] J.G Nagy and Z. Strakos. Enforcing nonnegativity in image reconstruction algorithms. *Mathematical Modeling, Estimation and Imaging*, 4121:182–190, 2000.
- [8] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer Verlag, August 27 1999.
- [9] Marielba Rojas, Sandra A. Santos, and Danny C. Sorensen. A new matrix-free algorithm for the large-scale trust-region subproblem. *SIAM Journal on Optimization*, 11(3):611–646, 2000.
- [10] Marielba Rojas and Trond Steihaug. An interior-point trust-region-based method for large-scale non-negative regularization. *Inverse Problems*, (18), 2002.
- [11] Åke Björck. *Numerical Methods for Least Squares Problems*. SIAM, 2000.

