ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 9, No. 4, 2014, pp. 267-276

A smoothing Newton method for the minimum norm solution of linear program

Lina Zhang , Zhensheng Yu, Yanyan Zhu

[1]University of Shanghai for Science and Technology, Mobile: +86 18817849166;
E-mail address: zhanglina19880806@126.com.

(Received March 4, 2014, accepted October 1, 2014)

Abstract. In this paper, we propose a smoothing Newton method to find the minimum norm solution of linear program problems. By using the smoothing technique, we reformulate the problem as an unconstrained minimization problem with a twice continuous differentiable objective function. The minimization of this objective function can be carried out by the classical Newton-type method which is shown to be globally convergence.

Keywords: Linear program, Minimum norm solution, Unconstrained minimization reformulation, Smoothing function, Newton-type method.

1. Introduction

Consider the linear program in primal form

$$\min c^T x \quad s.t. \quad Ax = b, x \geq 0 \tag{1}$$

Together with its dual

$$\max b^T \lambda \quad s.t. \quad A^T \lambda \leq c \tag{2}$$

Where $A \in R^{m \times n}, c \in R^n, and\ b \in R^m$ are the given data, and A is assumed to have full row rank. Let us denote the optimal value of the primal problem (1) by

$$\inf(\mathrm{P}) := \{\mathrm{c}^T \mathrm{x} \,|\, Ax = b, x \geq 0\}$$

Throughout this manuscript, we assume

$$\inf(P) \in R \tag{3}$$

This is equivalent to saying that the primal (and hence also the dual) linear program has a nonempty solution set.

The aim of this paper is to find the minimum norm solution of the primal program (1), i.e., we want to find the solution x^* of the program

$$\min \frac{1}{2}\|x\|^2 \quad s.t. \quad Ax = b, c^T x = \inf(\mathrm{P}), \mathrm{x} \geq 0 \tag{4}$$

Note that this problem has a unique solution under the assumption (3).Since the minimum norm solution could be a vertex as well as a point belonging to the relative interior of the solution set, neither the simplex method [1] nor the class of interior-point methods [2] will be assured to find the minimum norm solution of (1).

The standard method for finding a minimum norm solution of a convex program is based on the Tikhonov regularization[3]. Specialized to our linear program (1), the Tikhonov regularization generates a sequence of iterates $\{x^k\}$ with x^k being the unique solution of the regularized program

$$\min c^T x + \frac{\varepsilon_k}{2}\|x\|^2 \quad s.t. \quad Ax = b, x \geq 0 \tag{5}$$

where $\varepsilon_k > 0$ is a positive parameter and the sequence $\{\varepsilon_k\}$ tends to zero. However, the Tikhonov regularization is, in general, quite costly since it has to solve a sequence of quadratic programs. On the other hand, due to special properties of linear programs, it is known that a solution of a single quadratic program (5)

with a sufficiently small but positive parameter already gives a solution of (4). This follows from Mangasarian and Meyer[[4],Corollary2].

To overcome this drawback, Kanzow, Qi and Qi [5] describes a new technique to solve problem (4), their main idea is to reformulate the problem as an unconstrained minimization problem with a convex and smooth objective function, a Newton-type method with global convergence was proposed. We note that although the objective function in [5] is smooth and convex, it does not twice continuous differentiable, and therefore the classical Newton method can't be used. Based on the recently development of smoothing function for complementarity problems, see, for examples [8,15,16,19], in this paper, we propose a smoothing unconstrained optimization reformulation and a Newton method to find the solution of problem (2). This reformulation provides a twice continuous differentiable objective function of the reformulation in [5], we obtain the positive definite of the Hessian matrix of the objective function under certain conditions and hence the classical Newton method can be used to solve the problem directly.

The organization of this paper is as follows. In Section 2, we give the smoothing unconstrained optimization reformulation and some properties of the reformulation including the Newton method. The numerical results are reported in Section 3.

A few words about our notation: We denote the n-dimensional real space by R^n. For a vector $x \in R^n$, we write $\|x\|$ for its Euclidean norm, and x_+ or $[x]_+$ for the vector $\max\{0, x\}$, where the maximum is taken componentwise, i.e., x_+ is the projection of x onto the nonnegative orthant. The rows of a matrix B will be denoted by B_i, where as we write b_{ij} for the (i, j) th element of the matrix B.

2. Unconstrained Minimization Reformulation

The following result is given in [6], which is essentially based on some related results given by Mangasarian [7] and Mangasarian and Meyer[4].

Theorem 2.1 A vector $x^* \in R^n$ is the minimum norm solution of the primal linear program (1) if and only if there exists a positive number R>0 such that, for each $r \geq R$, we have

$$x^* = \left[A^T \lambda_r^* - \mathrm{rc} \right]_+$$

Where λ_r^* denotes a solution of the nonlinear system

$$A\left[A^T \lambda_r - rc \right]_+ = b$$

Motivated by the characterization stated in Theorem 2.1, Kanzow, Qi and Qi [5] gave an unconstrained minimization reformulation of problem (4) as follows:

$$\min \ f(\lambda) = \frac{1}{2} \left\| \left[A^T \lambda_r - \mathrm{rc} \right]_+ \right\|^2 - b^T \lambda_r, \lambda_r \in R^m \tag{6}$$

The objective function in (6) has the following property, the proof can be found in [5].

Lemma 2.1 The function f from (6) is convex and continuously differentiable with gradient:

$$\nabla f(\lambda_r) = A\left[A^T \lambda_r - rc \right]_+ - b \tag{7}$$

The lemma shows that x^* is the minimum norm solution of (1) if and only if λ_r^* is the stationary point of problem (6). The objective function f is once but not twice continuously differentiable, [5] employed a generalized Newton method to find the solution of problem (6).

In what follows, we consider reformulating the problem into a smoothing unconstrained optimization and then employ the classical Newton to solve it. Firstly, we introduce the smoothing function used in this paper. Define the step function

$$\sigma(x) = \begin{cases} 1 & if \ x > 0 \\ 0 & if \ x \leq 0 \end{cases}$$

In the extensive neural network literature, the step function is very effectively approximated by the sigmoid function

$$s(x, \alpha) = \frac{1}{1 + e^{-\alpha x}}, \alpha > 0$$

In this work we utilize the integral of the sigmoid function as an approximation to the plus function $(x)_+$ as follows:

$$(x)_+ \approx p(x,\alpha) = \int_{-\infty}^{x} s(y,\alpha)d_y = x + \frac{1}{\alpha}\log(1+e^{-\alpha x}) \quad (8)$$

For even moderate value of α, the function $p(x,\alpha)$ is a good approximation to the plus function. As α approaches infinity, $p(x,\alpha)$ approaches $(x)_+$ from above and remains continuously differentiable as many times as we wish. Hence first order and second order gradient methods can be used to solve the reformulated problem involving the p function. We treat α as a parameter in the function $p(.,x)$. Hence when we say p' or p^{-1}, we mean the derivative or inverse of p with respect to the first variable with the parameter α fixed.

The basic properties of p(x,α) can be found in [8].

Lemma 2.2 The function of $p(x,\alpha), \alpha > 0$ has the following properties:

1. $p(x,\alpha)$ is k-times continuously differentiable for any positive integer k, with $p'(x,\alpha) = \frac{1}{1+e^{-\alpha x}}$ and $p''(x,\alpha) = \frac{\alpha e^{-\alpha x}}{(1+e^{-\alpha x})^2}$.
2. $p(x,\alpha)$ is strictly convex and strictly increasing on R.
3. $p(x,\alpha) > x_+$, for all $x \in R$.
4. $\max_{x\in R}\{p(x,\alpha) - x_+\} = p(0,\alpha) = \frac{\log 2}{\alpha}$
5. $\lim_{|x|\to\infty} p(x,\alpha) - x_+ = 0$, for all $\alpha > 0$.
6. $\lim_{\alpha\to\infty} p(x,\alpha) = x_+$, for all $x \in R$.
7. $p(x,\alpha) \in (o,\infty)$ for all $x \in R, \alpha > 0$. The inverse function p^{-1} is well defined for $x \in (0,\infty)$.
8. $p(x,\alpha) > p(x,\beta)$, for $\alpha < \beta, x \in R$.

This p function with a smoothing parameter α is used here to replace the plus function of (6). Then we can rewrite the unconstrained minimization problem as (9), whose objective function is twice continuously differentiable.

$$f(\lambda_r) := \frac{1}{2}\left\| p\left(A^T\lambda_r - rc\right), \alpha\right\|^2 - b^T\lambda_r, \lambda_r \in R^m \quad (9)$$

Lemma 2.3 Combining lemma 2.1 and the property 2 of lemma 2.2, we know that the objective function (9) is also convex.

In order to describe our Newton-type method we have to give the gradient and Hessian matrix of (9). For the sake of writing conveniently in our calculating, we use x instead of λ_r, let $A \in R^{m\times n}, c \in R^n, and\ b \in R^m$, the objective function (9) is the following form:

$$f(x) = \frac{1}{2}\| A^T x - rc + \frac{1}{\alpha}\log(1+e^{-\alpha(A^T x - rc)})\|^2 - b^T x,\ x \in R^m$$

The gradient of $f(x)$ as well as its Hessian matrix is described as follows:

$$\nabla f(x) = A\left[I - \begin{pmatrix} t_1 & & & \\ & t_2 & & \\ & & \ddots & \\ & & & t_n \end{pmatrix}\right]T - b$$

$$\nabla^2 f(x) = \nabla T^T T + \sum_{i=1}^{n} T_i \nabla^2 T_i$$

Where $T(x) = A^T x - rc + \frac{1}{\alpha}\log(1+e^{-\alpha(A^T x - rc)})$ and $t_j = \frac{1}{e^{\alpha(\sum_{i=1}^{m} a_{ij}x_i - rc_i + 1)+1}}$, $j = 1,2,\cdots n$.

The calculate process is described as follows:

Let $T(x) = A^T x - rc + \frac{1}{\alpha}\log(1+e^{-\alpha(A^T x - rc)})$, $T(x)$ is a n dimensional vector

$$T(x) = \begin{pmatrix} T_1(x) \\ T_2(x) \\ \vdots \\ T_n(x) \end{pmatrix} \tag{10}$$

Thus the objective function of (9) can be wrote as

$$f(x) = \frac{1}{2}T^T(x)T(x) - b^T x \tag{11}$$

then $\nabla f(x) = \nabla T^T T - b$, we need to calculate ∇T first.

$$\nabla T = A^T + \frac{1}{\alpha}\nabla \log(1+e^{-\alpha(A^T x - rc)}) \tag{12}$$

$$\begin{aligned} &\nabla \log(1+e^{-\alpha(A^T x - rc)}) \\ &= -\alpha \begin{pmatrix} \frac{1}{e^{\alpha(\sum_{i=1}^{m} a_{i1}x_i - rc_i + 1)}} a_{11} & \frac{1}{e^{\alpha(\sum_{i=1}^{m} a_{i1}x_i - rc_i + 1)}} a_{21} & \cdots & \frac{1}{e^{\alpha(\sum_{i=1}^{m} a_{i1}x_i - rc_i + 1)}} a_{n1} \\ \vdots & \vdots & & \vdots \\ \frac{1}{e^{\alpha(\sum_{i=1}^{m} a_{in}x_i - rc_n + 1)}} a_{1n} & \frac{1}{e^{\alpha(\sum_{i=1}^{m} a_{in}x_i - rc_n + 1)}} a_{2n} & \cdots & \frac{1}{e^{\alpha(\sum_{i=1}^{m} a_{in}x_i - rc_n + 1)}} a_{mn} \end{pmatrix} \\ &= -\alpha \begin{pmatrix} t_1 & & & \\ & t_2 & & \\ & & \ddots & \\ & & & t_n \end{pmatrix} A^T \end{aligned} \tag{13}$$

with $t_j = \frac{1}{e^{\alpha(\sum_{i=1}^{m} a_{ij}x_i - rc_i + 1)+1}}$, $j = 1,2,\cdots n$. Consider both (12) and (13), we can get

$$\nabla T = A^T - \begin{pmatrix} t_1 & & & \\ & t_2 & & \\ & & \ddots & \\ & & & t_n \end{pmatrix} A^T \tag{14}$$

So the gradient of $f(x)$ is

$$\begin{aligned}\nabla f(x) &= \left[A^T - \begin{pmatrix} t_1 & & & \\ & t_2 & & \\ & & \ddots & \\ & & & t_n \end{pmatrix} A^T \right]^T T - b \\ &= A\left[I - \begin{pmatrix} t_1 & & & \\ & t_2 & & \\ & & \ddots & \\ & & & t_n \end{pmatrix} \right] T - b \end{aligned} \tag{15}$$

Now we compute the Hessian matrix of $f(x)$.

$$\text{Since } \nabla T(x) = \begin{pmatrix} \frac{\partial T_1(x)}{\partial x_1} & \frac{\partial T_1(x)}{\partial x_2} & \cdots & \frac{\partial T_1(x)}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial T_n(x)}{\partial x_1} & \frac{\partial T_n(x)}{\partial x_2} & \cdots & \frac{\partial T_n(x)}{\partial x_n} \end{pmatrix} \tag{16}$$

so its transposition can be noted as

$$\nabla T^T(x) = \begin{pmatrix} \frac{\partial T_1(x)}{\partial x_1} & \frac{\partial T_2(x)}{\partial x_1} & \cdots & \frac{\partial T_n(x)}{\partial x_1} \\ \vdots & \vdots & & \vdots \\ \frac{\partial T_1(x)}{\partial x_n} & \frac{\partial T_2(x)}{\partial x_n} & \cdots & \frac{\partial T_n(x)}{\partial x_n} \end{pmatrix} \tag{17}$$

Using (16) and (17) we kwon

$$\nabla(\nabla T^T T) = \begin{pmatrix} \sum_{i=1}^{n}(\frac{\partial T_i}{\partial x_1}\frac{\partial T_i}{\partial x_1} + T_i\frac{\partial^2 T_i}{\partial x_1^2}) & \sum_{i=1}^{n}(\frac{\partial T_i}{\partial x_2}\frac{\partial T_i}{\partial x_1} + T_i\frac{\partial^2 T_i}{\partial x_1 x_2}) & \cdots & \sum_{i=1}^{n}(\frac{\partial T_i}{\partial x_n}\frac{\partial T_i}{\partial x_1} + T_i\frac{\partial^2 T_i}{\partial x_1 x_n}) \\ \vdots & \vdots & & \vdots \\ \sum_{i=1}^{n}(\frac{\partial T_i}{\partial x_1}\frac{\partial T_i}{\partial x_n} + T_i\frac{\partial^2 T_i}{\partial x_1 x_n}) & \sum_{i=1}^{n}(\frac{\partial T_i}{\partial x_2}\frac{\partial T_i}{\partial x_n} + T_i\frac{\partial^2 T_i}{\partial x_1 x_n}) & \cdots & \sum_{i=1}^{n}(\frac{\partial T_i}{\partial x_n}\frac{\partial T_i}{\partial x_n} + T_i\frac{\partial^2 T_i}{\partial x_n^2}) \end{pmatrix} \quad \text{where}$$

$$= \nabla T^T T + \sum_{i=1}^{n} T_i \nabla^2 T_i$$

$$T_i = \sum_{j=1}^{m} a_{ji}x_j - rc_i + \frac{1}{\alpha}\log\left[1 + e^{-\alpha\left(\sum_{j=1}^{m} a_{ji}x_j - rc\right)} \right].$$

As we know $\nabla f(x) = \nabla T^T T - b$, so the Hessian matrix of $f(x)$ is

$$\nabla^2 f(x) = \nabla T^T T + \sum_{i=1}^{n} T_i \nabla^2 T_i .$$

Since we get the Hessian matrix of the objective function (9), our next theorem is supposed to prove that it's positive definite.

Theorem 2.2 Let assume the matrix A has row full rank, then the Hessian matrix of $f(\mathrm{x})$ is positive definite.

Proof The Hessian matrix of $f(x)$ is the form of

$$\nabla^2 f(x) = \nabla T^T T + \sum_{i=1}^{n} T_i \nabla^2 T_i$$

where $T_i = \sum_{j=1}^{m} a_{ji} x_j - rc_i + \frac{1}{\alpha} \log\left[1 + e^{-\alpha\left(\sum_{j=1}^{m} a_{ji} x_j - rc\right)}\right]$.

It is obviously that $\nabla T^T T$ is positive definite, thus we only need to prove $\sum_{i=1}^{n} T_i \nabla^2 T_i$ is at least positive semidefinite.

We calculate ∇T_i which is the form as we show next.

$$\nabla T_i = \begin{pmatrix} a_{1i} \\ a_{2i} \\ \vdots \\ a_{mi} \end{pmatrix} + \frac{1}{\alpha} \frac{1}{1 + e^{-\alpha(\sum_{j=1}^{m} a_{ji} x_j - rc_i)}} (-\alpha) e^{-\alpha(\sum_{j=1}^{m} a_{ji} x_j - rc_i)} \begin{pmatrix} a_{1i} \\ a_{2i} \\ \vdots \\ a_{mi} \end{pmatrix}$$

$$= \left[1 - \frac{e^{-\alpha(\sum_{j=1}^{m} a_{ji} x_j - rc_i)}}{1 + e^{-\alpha(\sum_{j=1}^{m} a_{ji} x_j - rc_i)}}\right] \begin{pmatrix} a_{1i} \\ a_{2i} \\ \vdots \\ a_{mi} \end{pmatrix}$$

$$= \frac{1}{1 + e^{-\alpha(\sum_{j=1}^{m} a_{ji} x_j - rc_i)}} \begin{pmatrix} a_{1i} \\ a_{2i} \\ \vdots \\ a_{mi} \end{pmatrix}$$

Now we need to compute $\nabla^2 T_i$, which is necessary for our proof.

$$\nabla^2 T_i = \nabla^2 \begin{pmatrix} \frac{a_{1i}}{1 + e^{-\alpha(\sum_{j=1}^{m} a_{ji} x_j - rc_i)}} \\ \frac{a_{2i}}{1 + e^{-\alpha(\sum_{j=1}^{m} a_{ji} x_j - rc_i)}} \\ \vdots \\ \frac{a_{mi}}{1 + e^{-\alpha(\sum_{j=1}^{m} a_{ji} x_j - rc_i)}} \end{pmatrix}$$

$$= \frac{\alpha e^{-\alpha(\sum_{j=1}^{m} a_{ji} x_j - rc_i)}}{\left[1 + e^{-\alpha(\sum_{j=1}^{m} a_{ji} x_j - rc_i)}\right]^2} \begin{pmatrix} a_{1i}a_{1i} & a_{1i}a_{2i} & a_{1i}a_{mi} \\ a_{2i}a_{1i} & a_{2i}a_{2i} & a_{2i}a_{mi} \\ & & \\ a_{mi}a_{1i} & a_{mi}a_{2i} & a_{mi}a_{mi} \end{pmatrix}$$

$$= \frac{\alpha e^{-\alpha(\sum_{j=1}^{m} a_{ji}x_j - rc_i)}}{\left[1+e^{-\alpha(\sum_{j=1}^{m} a_{ji}x_j - rc_i)}\right]^2} \begin{pmatrix} a_{1i} \\ a_{2i} \\ \vdots \\ a_{mi} \end{pmatrix} (a_{1i} \quad a_{2i} \quad \cdots \quad a_{mi})$$

We can easily find $\nabla^2 T_i \geq 0$, consider the description of lemma 2.2 for all $x \in R$, $p(x,\alpha) > x_+$, so $p(x,\alpha) > 0$. Now we can conclude $\sum_{i=1}^{n} T_i \nabla^2 T_i \geq 0$, i.e. $\sum_{i=1}^{n} T_i \nabla^2 T_i$ is positive semidefinite. Then the Hessian matrix of $f(x)$ is positive definite.

Since the Hessian matrix of $f(x)$ is positive definite, we can use a Newton-type algorithm with an Armijo stepsize. We next describe our Newton-type algorithm for the objective function f(λ_r), which is refer to the algorithm in [9].

Algorithm 2.1(Newton-type method)

(S.0) choose parameters $0 \leq \varepsilon \square\ 1, \delta \in (0,1), \sigma \in (0,0.5), \alpha \geq 1, \mathrm{r} \geq 1$, and a starting point $\lambda_r^0 \in R^n$. Set $k:0$.

(S.1) calculate $g_k = \nabla f(\lambda_r^k)$, if $\square g_k \square \leq \varepsilon$, stop, set $\lambda_r^* \approx \lambda_r^k$.

(S.2) compute $G_k = \nabla^2 f(\lambda_r^k)$, and solving the following linear system of equations to get the solution d_k:

$$G_k d = -g_k$$

(S.3) let m_k is the smallest non-negative integers satisfying the under inequation.

$$f(\lambda_r^k + \delta^m d_k) \leq f(\lambda_r^k) + \sigma \delta^m g_k^T d_k$$

(S.4) let $\beta_k = \delta^{m_k}, \lambda_r^{k+1} = \lambda_r^k + \beta_k d_k, k := k+1$, go to (S.1)

Considering the twice differentiability of the objective function of problem (9) as well as an Armijo stepsize, our algorithm can be described as global quadratic convergence. The next work we'll do is to show this property of Algorithm 2.1.

Theorem 2.3 Let $\{\lambda_r^k\}$ be a sequence generated by Algorithm 2.1 and λ_r^{**} be the unique solution of problem (9).

(i) The sequence $\{\lambda_r^k\}$ converges to the unique solution λ_r^{**} from any initial point λ_r^0 in R^n.

(ii) For any initial point λ_r^0, there exists an integer $\bar{k}$ such that the stepsize β_k of Algorithm 2.1 equals 1 for $k \geq \bar{k}$ and the sequence $\{\lambda_r^k\}$ converges to λ_r^{**} quadratically.

The proof is similar to the Theorem 3.2 in [10], we omit it here.

3. Numerical results

In this section we consider some numerical results obtained with the approach described in the previous section. The Algorithm 2.1 were coded in MATLAB2013b and run on a PC with 2.50 GHz CPU processor. We implement MATLABV2013b to solve the following two examples. In the numerical experiments, we choose parameters δ=0.55, σ=0.4, x^* is the minimum norm solution of the primal linear program (1),$\square g_k \square$ denotes the Euclidean norm of the gradient.

Example1.

$$\max \quad z = 2x_1 + x_2$$
$$s.t.\begin{cases} 3x_1 + 5x_2 \le 15 \\ 6x_1 + 2x_2 \le 24 \\ x_1, x_2 \ge 0 \end{cases}$$

We choose the initial point $\lambda_r^0 = (0,0)^T$ and let $\varepsilon = 1e-10$.

Tab.1 Results for Example 1

λ_0	r	α	x^*	$\|g_k\|$
$(0,0)^T$	10	1	$(3.8160,0.6889)^T$	5.5140e-006
$(0,0)^T$	10	21	$(3.7500,0.7500)^T$	1.2946e-007
$(0,0)^T$	10	41	$(3.7500,0.7500)^T$	7.6528e-014
$(0,0)^T$	10	61	$(3.7500,0.7500)^T$	0
$(0,0)^T$	10	81	$(3.7500,0.7500)^T$	0
$(0,0)^T$	5	100	$(3.7500,0.7500)^T$	1.1916e-014
$(0,0)^T$	7	100	$(3.7500,0.7500)^T$	3.5527e-015
$(0,0)^T$	13	100	$(3.7500,0.7500)^T$	0
$(0,0)^T$	19	100	$(3.7500,0.7500)^T$	0
$(0,0)^T$	25	100	$(3.7500,0.7500)^T$	0

Example2.

$$\max \quad z = 3x_1 + 2x_2 + 2.9x_3$$
$$s.t.\begin{cases} 8x_1 + 2x_2 + 10x_3 \le 300 \\ 10x_1 + 5x_2 + 8x_3 \le 400 \\ 2x_1 + 13x_2 + 10x_3 \le 420 \\ x_1, x_2, x_3 \ge 0 \end{cases}$$

We choose the initial point $\lambda_r^0 = (0,0,0)^T$ and let $\varepsilon = 1e-20$.

Tab.2 Results for Example 2

λ_0	r	α	x^*	$\|g_k\|$
$(0,0,0)^T$	29	1	$(22.5371,23.1943,7.3356)^T$	3.0611e-013
$(0,0,0)^T$	29	38	$(22.5333,23.2000,7.3333)^T$	8.0389e-014
$(0,0,0)^T$	29	75	$(22.5333,23.2000,7.3333)^T$	1.9691e-013
$(0,0,0)^T$	29	112	$(22.5333,23.2000,7.3333)^T$	1.3924e-013
$(0,0,0)^T$	29	149	$(22.5333,23.2000,7.3333)^T$	8.0389e-014
$(0,0,0)^T$	22	100	$(22.5333,23.2000,7.3333)^T$	0
$(0,0,0)^T$	24	100	$(22.5333,23.2000,7.3333)^T$	0
$(0,0,0)^T$	26	100	$(22.5333,23.2000,7.3333)^T$	0
$(0,0,0)^T$	28	100	$(22.5333,23.2000,7.3333)^T$	1.9691e-013
$(0,0,0)^T$	29	100	$(22.5333,23.2000,7.3333)^T$	8.0389e-014

In summary, the results from table1 and table2 show that the performance of our smoothing method is quite effective. The parameter r and the smoothing parameter α effects our results as they changed.

4. References

[1] D Bertsimas, J N Tsitsiklis. Introduction to Linear Optimization, Athena Scientific, Belmont, MA, 1997.
[2] L. Qi, J. Sun. A Nonsmooth Version of Newton's Method, Mathematical Programming, 58(1993), pp.353-367.
[3] A. Fischer, C. Kanzow. On Finite Termination of an Iterate Method for Linear Complementarity Problems, Mathematical Programming, ,74(1996), pp. 279-292.
[4] O. L. Mangasarian, R.R.Meyer. Nonlinear Perturbation of Linear Programs, SIAM Journal on Control and Optimization, 17(1979), pp. 745-757.
[5] C. Kanzow, H. Qi, and L. Qi. On the Minimum Norm Solution of Linear Programs, Journal of Optimization Theory and Applications, 116(2003), pp.333-345.
[6] T. Deluca, F. Facchinei, C. Kanzow. A Semismooth Equation Approach to the Solution of Nonlinear Complementarity Problems, Mathematical Programming, 75(1996), pp. 407-439 .
[7] O. L. Mangasarian. Normal Solutions of Linear Programs, Mathematical Programming, 22(1984), pp. 206-216.
[8] C. Chen, O. L. Mangasarian. A Class of Smoothing Functions for Nonlinear and Mixed Complementarity Problems. Computation Optimization with Applications. 5(1996), pp. 97-138.
[9] C. F. Ma. Optimization Method and Its Matlab Programming, Science Press, 2011, pp. 29-40.
[10] Y. J. Lee, O. L. Mangasarian. A Smooth Support Vector Machine for Classification, Computational Optimization and Applications, 20(2001), pp. 5-22.
[11] B. Chen, P. T. Harker. A Non-interior-point Continuation Method for Linear Complementarity Problems. SIAM Journal on Matrix Analysis and Applications, 4(1993), pp.1168-1190.
[12] C. Kanzow. Some Noninterior Continuation Methods for Linear Complementarity Problems. SIAM Journal on Matrix Analysis and Applications. 4(1996), pp. 851-868.
[13] J. Burke, S. Xu. The Global Linear Convergence of a Noninterior Path-following Algorithm for Linear Complementarity Problems. Mathematics of Operational Research. 23(1998), pp. 719-734.
[14] W. Y. Sun, Y. X. Yuan. Optimization Theory and Methods, Science Press, 2010, pp. 2-330.
[15] H. D. Qi, L. Z. Liao. A Smoothing Newton Method for General Nonlinear Complementarity Problems,

Computational Optimization and Applications, 17(2000), pp. 231-253.
[16] L. Qi, D. Sun, G. Zhou. A New Look at Smoothing Newton Methods for Nonlinear Complementarity Problems and Box Constrained Variational Inequalities. Math Program, A87(2000), pp. 1-35.
[17] S. H. Pan, Y. X. Jiang. Smoothing Newton Method for Minimizing the Sum of p-Norms, Journal of Optimization Theory and Applications, 137(2008), pp. 255-275.
[18] C. F. Ma. A New Smoothing and Regularization Newton method for P0- NCP, Journal of Global Optimization, 48(2010), pp. 241-261.
[19] J. G. Zhu, B. B. Hao. A New Class of Smoothing Functions and a Smoothing Newton Method for Complementarity Problems, Optimization Letters, 7(2013), pp. 481-497.
[20] L. A. Vese, T. F. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. International Journal of Computer Vision, , 50(3) (2002),pp.271-293
[21] Weiguo Lu. Fast free-form deformable registration via calculus of variations.Phys. Med. Biol., 49(2004), pp.3067–3087.