INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 10, Number 3, Pages 745–755

AN ALGORITHM FOR FINDING NONNEGATIVE MINIMAL NORM SOLUTIONS OF LINEAR SYSTEMS

SAÏD BAHI AND AARON ROSS

(Communicated by Edward Allen)

Abstract. A system of linear equations $Ax = b$, in n unknowns and m equations which has a nonnegative solution is considered. Among all its solutions, the one which has the least norm is sought when $\mathbb{R}^n$ is equipped with a strictly convex norm. We present a globally convergent, iterative algorithm for computing this solution. This algorithm takes into account the special structure of the problem. Each iteration cycle of the algorithm involves the solution of a similar quadratic problem with a modified objective function. Duality conditions for optimality are studied. Feasibility and global convergence of the algorithm are proved. As a special case we implemented and tested the algorithm for the ℓ^p-norm, where $1 < p < \infty$. Numerical results are included.

Key words. Linear equations, Least norms, Optimality, Duality conditions.

1. Introduction

We will be considering a system of real linear equations

$$Ax = b, \tag{1.1}$$

where A is an $m \times n$-real matrix, b in m-real vector and x an n-real vector. Under the assumption that the system (1.1) has a non-negative solution, we study the following problem: out of all non-negative solutions of (1.1) compute the solution that has the least norm when the norm considered on $\mathbb{R}^n$ is strictly convex. This naturally, includes the ℓ^p-norm, $1 < p < \infty$. The algorithm proposed in this work solves the minimal norm problem

$$\min \{\|x\| \mid Ax = b,\ x \in \mathbb{R}^n,\ x \geq 0\}. \tag{P}$$

We assume that $b \neq 0$, because otherwise the problem is trivial. All the steps of the algorithm for computing the solution of (P) will be shown to be feasible. Its global convergence will then be proved.

To solve the given problem, a dual problem, denoted (P'), will be associated with (P). An outline of the correspondence between (P) and (P') will be given. The main application of this work is the ℓ^p-norm case. Namely, find $x \in \mathbb{R}^n$ that minimizes the ℓ^p- problem

$$minimize\{ \|x\|_p \mid Ax = b, x \in \mathbb{R}^n, x \geq 0\}. \tag{1.2}$$

It should be noted here that the objective function in (1.2) need not be twice differentiable. The case $1 < p < 2$ has been more troublesome since methods requiring second derivatives will not be defined for certain non-zero points. While (1.2) is a smooth convex programming problem and thus susceptible to general programming procedures, it seems natural to take into account in our algorithm the special structure of the problem. For its convergence the proposed algorithm does not need extra differentiability of the norm.

Received by the editors July 5, 2012 and, in revised form, October 9, 2012.
2000 *Mathematics Subject Classification.* 90C25, 90C30.

For $p = 2$, problem (1.2) becomes a special case of what Lawson and Hanson referred to in [7] as the least distance programming (LDP) and for which they gave a finite algorithm. This algorithm (or any other similar purpose one) is used in this work as follows: at each iteration step of our algorithm, the LDP problem

$$Ax = b, x \geq 0, \quad \|x - a_k\|_2(min),$$

where (a_k) is a sequence defined by the algorithm, is solved using the LDP algorithm.

The main contribution of this paper is to propose a method of solution of probelm (P) that is not limited to a single norm such as the ℓ^2-norm. Different applications suggest different norms to use. Ideally, we seek a solution that optimize general norms. In many applications, a system of linear equations may have many solutions (e.g. when solving linear operator equations) and it may be needed, following a discretization, to select one solution under a given criteria. This criteria could be to find a solution that has the least norm or a solution that is the closest to a (target) point a in which case one needs to minimize $\|x - a\|$ among all solutions of a linear system. The classical ℓ^2-norm may not be always the best choice. For instence, in sparse solution construction and compressed sensing, similar ℓ^p minimization problems arise for $0 < p < 1$. Other applications arise when solving some variational problems.

2. Main notation and duality

Let the norm $\|\cdot\|$ on $\mathbb{R}^n, n \geq 1$, be arbitrary. The norm is said to be *smooth* if and only if through each point of unit norm there passes a unique hyperplane supporting the closed unit ball $B = \{x \in \mathbb{R}^n \mid \|x\| \leq 1\}$. The norm is said to be *strictly convex* if and only if the unit sphere $S = \{x \in \mathbb{R}^n \mid \|x\| = 1\}$ has no line segment on it.

To introduce the dual problem (P'), we define the dual norm $\|\cdot\|'$ on $\mathbb{R}^n$ by

$$\|y\|' = \max\{\langle x, y\rangle \mid \|x\| = 1, \ x \in \mathbb{R}^n\}.$$

For any vector $v \in \mathbb{R}^n, v \neq 0$, a $\|\cdot\|$-*dual vector*, v' is defined by

$$\|v'\| = 1, \ \langle v', v\rangle = \|v\|'. \tag{2.1}$$

Similarly, for the dual norm, a $\|\cdot\|'$-dual vector v^* is defined by

$$\|v^*\|' = 1, \ \langle v^*, v\rangle = \|v\|.$$

The map $y \mapsto y'$ (resp. $y \mapsto y^*$) is odd, continuous and positively homogeneous of degree zero on $\mathbb{R}^n\backslash\{0\}$, if the norm is strictly convex (resp. smooth). For $v \neq 0$, we have the relations $v^{'*} = v/\|v\|'$ (resp. $v^{*'} = v/\|v\|$) when the norm $\|\cdot\|$ is smooth (resp. strictly convex.)

When $\|\cdot\| = \|\cdot\|_p, \ 1 < p < \infty$, is the usual ℓ^p-norm, then $\|\cdot\|' = \|\cdot\|_q$, where $p + q = pq$. In terms of components, the dual vectors are given by

$$v_i' = (|v_i|/\|v\|_q)^{q-1} sgn \ (v_i), \quad v_i^* = (|v_i|/\|v\|_p)^{p-1} sgn \ (v_i), \quad i = 1, \ldots, n.$$

Let $K = \{x \in \mathbb{R}^n \mid x \geq 0, \ Ax = b\}$. Given problem (P), we associate a dual problem ([1], [8])

$$(P') \qquad \max\{\langle b, y\rangle \mid \ \xi \in \mathbb{R}^n, \xi \geq 0, \ y \in \mathbb{R}^m, \ \|\xi + A^T y\|' \leq 1\},$$

where A^T is the transpose of the matrix A. The relation between (P) and (P') is studied in the next two results.

Lemma 2.1 *If the norm $\|\cdot\|$ on $\mathbb{R}^n$ is arbitrary and K is non-empty, then value of $(P') \leq$ value of (P).*

Proof. Let $x \in K$, and let ξ and y be as defined in problem (P'). Then

$$\langle b, y\rangle = \langle x, A^T y\rangle \leq \langle x, A^T y + \xi\rangle \leq \|x\|.\|A^T y + \xi\|' \leq \|x\|. \quad \square$$

In their paper [8], Nikolopoulos and Sreedharan investigated in details the duality between (P) and (P') and stated that for an arbitrary norm on $\mathbb{R}^n$ the two problems have the same value. A useful characterization of the solution of (P) was then proposed by the authors. We discuss this characterization below. If K is non-empty and the norm $\|\cdot\|$ on $\mathbb{R}^n$ is strictly convex, then $\bar{x}$ is a non-negative minimal norm solution of problem (P) if and only if $A\bar{x} = b$, $\bar{x} \geq 0$ and there exist $\xi \in \mathbb{R}^n, \xi \geq 0$ and $y \in \mathbb{R}^m$ such that

$$\langle b, y\rangle > 0, \ \|A^T y + \xi\|' = 1 \tag{2.2}$$

and

$$\bar{x} = \langle b, y\rangle(A^T y + \xi)', \ \langle \xi, (A^T y + \xi)'\rangle = 0. \tag{2.3}$$

If in addition the norm $\|\cdot\|$ is smooth, then $\bar{x}^* = \xi + A^T y$ and $\langle \xi, \bar{x}\rangle = 0$. The couple (y, ξ) solves (P') and finally $\langle b, y\rangle = \langle \bar{x}^*, \bar{x}\rangle$. A characterization of the solution of problem (P) in the ℓ^2-norm case follows immediately from the discussion above. This is an important ingredient in the subsequent development of this work. We state it as

Corollary 2.2 *If $\|\cdot\| = \|\cdot\|_2$, then $\bar{x}$ is the solution of the problem (P) if and only if $\bar{x} \in K$, and*

$$\bar{x} = A^T y + \xi \text{ and } \langle \xi, \bar{x}\rangle = 0 \tag{2.4}$$

for some $\xi \in \mathbb{R}^n, \xi \geq 0$ and $y \in \mathbb{R}^m$.

It should be noted that when $\|\cdot\| = \|\cdot\|_2$, then $\bar{x}^* = \bar{x}/\|\bar{x}\|_2$, so that the corollary follows easily from the general case discussed earlier.

3. Algorithm

We assume that the system of linear equations $Ax = b$, $x \geq 0$, is feasible and that the norm $\|\cdot\|$ is strictly convex. We present an algorithm for computing the solution of (P). The feasibility and all other assertions made will be proven subsequently.

Algorithm 3.1

Step 0. Find the solution x_0 of $Ax = b$, $x \geq 0$, $\|x\|_2$ (min). Let

$$g_0 = x_0/\| x_0\|', \ \beta_0 = \langle g_0, x_0\rangle \text{ and } k = 0.$$

Step 1. Set $a_k = \beta_k g_k'$. Find x_{k+1} solution of

$$Ax = b, x \geq 0, \|x - a_k\|_2 \ (min).$$

Let $u_k = x_{k+1} - a_k$

Step 2. If $u_k = 0$, stop. x_{k+1} is the solution of (P); else continue.

Step 3. Set $\gamma_k = \langle u_k, x_{k+1}\rangle$.

Step 4. If $\gamma_k \geq \beta_k\|u_k\|' + \frac{1}{4}\|u_k\|_2^2$, let $g_{k+1} = u_k/\|u_k\|'$ *and* $\beta_{k+1} = \gamma_k/\|u_k\|'$ and GO TO step 8; else continue.

Step 5. Let $\mu_k = (\gamma_k - \frac{1}{2}\|u_k\|_2^2)/\beta_k$.

Step 6. Find $\alpha_k > 0$ such that $\|g_k + \alpha_k u_k\|' = 1 + \alpha_k\mu_k$.

Step 7. Let $g_{k+1} = (g_k + \frac{1}{2}\alpha_k u_k)/\|g_k + \frac{1}{2}\alpha_k u_k\|'$, and $\beta_{k+1} = (\beta_k + \frac{1}{2}\alpha_k \gamma_k)/\|g_k + \frac{1}{2}\alpha_k u_k\|'$.

Step 8. Increase k by 1 and return to step1.

Later the stopping rule of step 2 will be used as follows. We will show that the constructed sequence $(u_k)_{k\geq 0}$ converges to zero. Then because of this convergence, it will be proven that the algorithm converges to the solution of problem (P). However, for implementation purpose, one would use a more realistic stopping rule such as the duality gap criterion. In other words, we replace the condition $u_k = 0$ of step 2 by the condition

$$(\|x_k\| - \beta_k)/\|x_k\| \leq \eta,$$

where $\eta > 0$ is a stopping rule parameter.

The main interest in the remaining discussion will be focused on proving that the various steps of the algorithm are valid and that the important step 6, defining the step length $\alpha_k > 0$, is answered affirmatively and, finally, proving that the algorithm leads effectively to the solution of problem (P).

Because of corollary 2.2 we have

$$g_0 = \xi_0 + A^T y_0, \|g_0\|' = 1 \textit{ and } \beta_0 = \langle b, y_0 \rangle > 0, \tag{3.1}$$

for some $\xi_0 \geq 0$ in $\mathbb{R}^n$ and y_0 in $\mathbb{R}^m$. To see this, we recall that a necessary and sufficient condition for x_0 to be l^2-solution, as in step 0, is that $x_0 = A^T z + \xi$, and $\langle \xi, x_0 \rangle = 0$. If we set $g_0 = x_0/\|x_0\|'$ we get $g_0 = \xi_0 + A^T y_0$ with the appropriate ξ_0 and y_0. Finally, we note that $\beta_0 = \langle g_0, x_0 \rangle = \|x_0\|_2^2/\|x_0\|' > 0$ and $\langle g_0, x_0 \rangle = \langle \xi_0 + A^T y_0, x_0 \rangle = \langle y_0, Ax_0 \rangle = \langle y_0, b \rangle$.

In the following results of this paper, it will be shown that algorithm 3.1 is feasible and converges to a solution of problem (P). The lemmas presented in the remaining of this section show the feasibility of the algorithm. It depends mainly on the feasibility of steps 5 to 7. Lemma 3.2 proves the feasibility of step 7 of the algorithm and that the sequence of parameters β_k in fact approximates the optimal value of the dual problem. Lemma 3.4 shows that this sequence is positive strictly increasing and hence step 5 of the algorithm is well defined. The convergence of this sequence is also shown. The central result of the existence of a solution $\alpha_k > 0$ to the equation in step 6 and hence the feasibility of this step is shown in lemma 3.3.

Lemma 3.2. *(a) Let $u_k \neq 0$, be as defined in the algorithm. Then*

$$\forall \alpha > 0, \ g_k + \alpha u_k \neq 0. \tag{3.2}$$

(b) Let $\alpha_k > 0$ be as defined by step 6 of the algorithm, then there exist $\xi_{k+1} \in \mathbb{R}^n$, $\xi_{k+1} \geq 0$, $y_{k+1} \in \mathbb{R}^m$ such that

$$g_{k+1} = \xi_{k+1} + A^T y_{k+1}, \ \|g_{k+1}\|' = 1 \textit{ and } \beta_{k+1} = \langle b, y_{k+1} \rangle. \tag{3.3}$$

Proof. (a) is proved by induction on k. For $k = 0$, suppose that (a) does not hold,

i.e. $\exists \alpha > 0, \ g_0 = -\alpha u_0$. Using step 1 of the algorithm for $k = 0$, $u_0 = x_1 - a_0$, *thus*

$$\begin{aligned}
\|u_0\|_2^2 &= \langle u_0, x_1 - a_0 \rangle = -\alpha^{-1}(\langle g_0, x_1 \rangle - \langle g_0, a_0 \rangle) \\
&= -\alpha^{-1}(\langle x_1, \xi_0 + A^T y_0 \rangle - \beta_0) = -\alpha^{-1}(\langle Ax_1, y_0 \rangle + \langle x_1, \xi_0 \rangle - \beta_0) \\
&= -\alpha^{-1}(\beta_0 + \langle x_1, \xi_0 \rangle - \beta_0) = -\alpha^{-1}(\langle x_1, \xi_0 \rangle),
\end{aligned}$$

where the second equality is due to (3.1) and the fact that $\beta_0 = \langle g_0, x_0 \rangle$. This yields a contradiction since $x_1 \geq 0$, $\xi_0 \geq 0$, and $u_0 \neq 0$.

To prove (b), from step 1 of the algorithm and (2.4), there exist $\xi \in \mathbb{R}^n$, $\xi \geq 0$, $z \in R^m$ such that

$$u_0 = x_1 - a_0 = \xi + A^T z, \tag{3.4}$$

and $\langle z, b\rangle = \langle u_0, x_1\rangle = \gamma_0$. Suppose α_0 is determined by step 6 of the algorithm. Then $g_0 + (\alpha_0/2)u_0$ is nonzero. Let

$$y_1 = (y_0 + (\alpha_0/2)z)/\|g_0 + (\alpha_0/2)u_0\|'$$

and

$$\xi_1 = (\xi_0 + (\alpha_0/2)\xi)/\|g_0 + (\alpha_0/2)u_0\|'.$$

From (3.1) and (3.4), we obtain

$$\begin{aligned} A^T y_1 + \xi_1 &= (\xi_0 + A^T y_0 + (\alpha_0/2)(\xi + A^T z))/\|g_0 + (\alpha_0/2)u_0\|' \\ &= (g_0 + (\alpha_0/2)u_0)/\|g_0 + (\alpha_0/2)u_0\|' = g_1, \end{aligned}$$

and

$$\begin{aligned} \langle y_1, b\rangle &= (\langle y_0, b\rangle + (\alpha_0/2)\langle z, b\rangle)/\|g_0 + (\alpha_0/2)u_0\|' \\ &= (\beta_0 + (\alpha_0/2)\gamma_0)/\|g_0 + (\alpha_0/2)u_0\|' = \beta_1. \end{aligned}$$

The same argument applies for any integer k if we assume the proposition to be true for $k-1$. $\square$

It will be shown later that the sequence (β_k) generated by the algorithm is strictly increasing. This observation combined with (3.1) implies that $\beta_k > 0$, for all k. For this reason, step 5 is properly formulated. The crucial step of finding $\alpha_k > 0$ is step 6 of the algorithm will now be proved to be feasible.

Lemma 3.3. *Suppose the algorithm is at the stage of executing step 5. Let μ_k be defined by $\mu_k = (\gamma_k - \|u_k\|_2^2/2)/\beta_k$. Then, there exists $\alpha_k > 0$ such that*

$$\|g_k + \alpha_k u_k\|' = 1 + \alpha_k \mu_k. \tag{3.5}$$

Proof. We only give a sketch of the proof which actually follows the same lines as in theorem 4.1 of [1]. We define the real valued functions $f(\lambda) = \|g_k + \lambda u_k\|'$ and $l(\lambda) = 1 + \lambda\mu_k$, where λ is a real number. Then f is a strictly convex function, $f(0) = \|g_k\|' = 1 = l(0)$, by the definition of g_k in step 4 of the algorithm, and $f'(0) = \langle g_k', u_k\rangle$. It is easily seen that if the algorithm is at the stage of executing step 5, then $u_k \neq 0$, $g_k' = \frac{1}{\beta_k} a_k$ and

$$\begin{aligned} f'(0) - l'(0) &= \langle g_k', u_k\rangle - \mu_k \\ &= \frac{1}{\beta_k}\langle a_k, u_k\rangle - \mu_k \\ &= \frac{1}{\beta_k}\langle x_{k+1} - u_k, u_k\rangle - \mu_k \\ &= \frac{1}{\beta_k}\langle x_{k+1}, u_k\rangle - \frac{1}{\beta_k}\langle u_k, u_k\rangle - \frac{1}{\beta_k}(\gamma_k - \|u_k\|_2^2/2) \\ &= \frac{1}{\beta_k}\gamma_k - \frac{1}{\beta_k}\|u_k\|_2^2 - \frac{1}{\beta_k}\gamma_k + \frac{1}{2\beta_k}\|u_k\|_2^2 \\ &= -\|u_k\|_2^2/(2\beta_k) < 0. \end{aligned}$$

Hence, there must exist $\lambda > 0$ such that $f(\lambda) - l(\lambda) < 0$. Now, $\alpha_k > 0$ in step 5 is sought only if step 4 in the algorithm is answered negatively, i.e. if $\gamma_k < \beta_k\|u_k\|' + \|u_k\|_2^2/4$, in which case

$$f(\lambda) - l(\lambda) \geq \lambda(\beta_k\|u_k\|' - \gamma_k + \|u_k\|_2^2/2)/\beta_k \ - 2 \longrightarrow \infty,$$

as $\lambda \longrightarrow \infty$. Thus, because of continuity, (3.5) holds. $\square$

Lemma 3.4. *Let the norm $\|.\|$ be smooth. Then: (a) the sequence (β_k) generated by the algorithm is strictly increasing, and (b) either the sequence (β_k) is finite or it is a convergent infinite sequence.*

Proof. β_k is computed either in step 4 or step 7 of the algorithm. If the first is executed then $\gamma_k \geq \beta_k\|u_k\|' + \|u_k\|_2^2/4$, so that by step 4 defining β_{k+1}

$$\beta_{k+1} = \gamma_k/\|u_k\|' \geq \beta_k + \|u_k\|_2^2/4\|u_k\|' > \beta_k$$

(since $u_k \neq 0$). Now on the contrary, if we assume that step 4 of the algorithm is answered negatively, then β_{k+1} is defined in step 7. Because the norm is smooth and so the dual norm $\|.\|'$ is strictly convex, it follows that for $\alpha_k > 0$ as defined in step 6 and by equation (3.5)

$$\begin{aligned}
\|g_k + (\alpha_k u_k)/2\|' &< \|g_k + \alpha_k u_k\|'/2 + \|g_k\|'/2 = \frac{1}{2}(1 + \alpha_k\mu_k) + \frac{1}{2} \\
&= 1 + \frac{1}{2}\alpha_k\mu_k \\
&= 1 + \alpha_k(\gamma_k - \|u_k\|_2^2/2)/(2\beta_k),
\end{aligned}$$

where the last equality is due to the definition of μ_k in step 5 of the algorithm. Multiplying both sides by β_k, we have

$$\beta_k\|g_k + (\alpha_k u_k)/2\|' < \beta_k + \alpha_k(\gamma_k - \|u_k\|_2^2/2)/2 < \beta_k + (\alpha_k\gamma_k)/2.$$

Consequently,

$$\beta_k < (\beta_k + (\alpha_k\gamma_k)/2)/\|g_k + (\alpha_k u_k)/2\|' = \beta_{k+1},$$

by definition of β_{k+1} in step 7 of the algorithm. This shows that, in all cases, the sequence β_k is strictly increasing. To prove (b), suppose that the sequence (β_k) is infinite. Then, we see immediately from the weak duality that $\beta_k = \langle y_k, b\rangle \leq \|x\|$, for any fixed feasible solution $x \in \mathbb{R}^n$. This shows that (β_k) is bounded and thus, convergent. $\square$

4. Convergence

After the preliminary results in the previous section concerning the feasibility of the algorithm, we begin the study of its convergence. The next two Lemmas will be needed to establish the convergence to the solution of problem (P).

Lemma 4.1. Let $\alpha_k > 0$ be as defined in step 6 of the algorithm. Then

$$\beta_k\langle(g_k + \alpha_k u_k)', u_k\rangle + \|u_k\|_2^2/2 \geq \gamma_k. \tag{4.1}$$

Proof. From the definition of the dual vectors and the construction of g_k, we know that $\|(g_k + \alpha_k u_k)'\| = 1$ and $\|g_k\|' = 1$, and this implies

$$\langle(g_k + \alpha_k u_k)', g_k\rangle \leq \|(g_k + \alpha_k u_k)'\|.\|g_k\|' = 1.$$

Due to the definition of the dual norm, we also have the defining equation

$$\|g_k+\alpha_k u_k\|' = \langle(g_k+\alpha_k u_k)', (g_k+\alpha_k u_k)\rangle = \langle(g_k+\alpha_k u_k)', g_k\rangle+\alpha_k\langle(g_k+\alpha_k u_k)', u_k\rangle.$$

Therefore

$$\|g_k + \alpha_k u_k\|' \leq 1 + \alpha_k\langle(g_k + \alpha_k u_k)', u_k\rangle.$$

This, combined with the equation (3.5) defining α_k and step 5 defining μ_k, yields

$$1 + \alpha_k(\gamma_k - \|u_k\|_2^2/2)/\beta_k = 1 + \alpha_k\mu_k = \|g_k + \alpha_k u_k\|' \leq 1 + \alpha_k\langle(g_k + \alpha_k u_k)', u_k\rangle,$$

so that

$$\alpha_k(\gamma_k - \|u_k\|_2^2/2)/\beta_k \leq \alpha_k \langle (g_k + \alpha_k u_k)', u_k \rangle,$$

which implies (4.1) since $\alpha_k > 0$. □

Lemma 4.2. let α_k, x_k and u_k be as defined in the algorithm. Then the sequences (a_k), (x_k) and (u_k) are bounded.

Proof. If the algorithm terminates in a finite number of iterations, the lemma is trivial. Consider the case when the sequences are infinite. Let $d > 0$ be the value of the minimization problem (P). Then, as mentioned above, $\beta_k = \langle b, y_k \rangle \leq d$, for all k, by Lemma (2.1). From step 1 of the algorithm, we see that $\|a_k\|_2 = \beta_k \|g_k'\|_2 \leq M\beta_k \|g_k'\| \leq Md$, where $M > 0$ is such that $\|v\|_2 \leq M\|v\|$ for all $v \in \mathbb{R}^n$. Hence (a_k) is bounded.

To see that the sequence (x_k) is bounded, let $\tilde{x}$ be any fixed feasible solution of problem (P). Because x_{k+1} is the minimizer of the problem

$$Ax = b, x \geq 0, \|x - a_k\|_2 \ (min),$$

(this minimization is done in step 1 of the algorithm at each iteration cycle) we get

$$\|x_{k+1}\| \leq \|x_{k+1} - a_k\|_2 + \|a_k\|_2 \leq \|\tilde{x} - a_k\|_2 + \|a_k\|_2 \leq \|\tilde{x}\|_2 + 2\|a_k\|_2 \leq \|\tilde{x}\|_2 + 2Md.$$

Thus (x_k) is bounded. From this it clearly follows that the sequence (u_k) where $u_k = x_{k+1} - a_k$ is also bounded. □

We now put everything together to prove that the algorithm converges to the solution of problem (P). The first thing is to observe that step 2 is true, i.e. for the sequence (u_k) generated by the algorithm, if $u_k = 0$ for some k, then x_{k+1} is the solution of problem (P). If $u_k = 0$ as in step 2, then because of step 1, $x_{k+1} = a_k = \beta_k g_k'$. By virtue of lemma 3.2, $\beta_k = \langle b, y_k \rangle$ and $g_k = \xi_k + A^T y_k$, from which it follows that

$$x_{k+1} = \langle b, y_k \rangle (\xi_k + A^T y_k)'.$$

Moreover, $\|\xi_k + A^T y_k\|' = 1$ and $\langle b, y_k \rangle > 0$. Using (2.2) and (2.3), it is obvious now that x_{k+1} solves (P). The previous discussion is summarized in the first part of the following convergence theorem.

Theorem 4.3. If the algorithm terminates after a finite number k of iterations, then x_{k+1} is the solution of problem (P). If the algorithm generates an infinite sequence (x_k), then it converges to the solution of (P).

Proof. Assume that the algorithm is executed for an infinite number of iterations (k). By Lemma 4.2, the sequence γ_k defined by $\gamma_k = \langle u_k, x_{k+1} \rangle$, where (x_k) and (u_k) are generated by the algorithm, is clearly a bounded sequence. Hence, by passing to a subsequence if necessary, we may assume that $\gamma_k \longrightarrow \gamma$. Since $0 < \beta_k \leq d$ and $\|g_k\|' = 1$, let us pass to further subsequence, denoted again by (k), such that $\beta_k \longrightarrow \beta$. and $g_k \longrightarrow g$. The first goal is to establish that $\lim_{k \to \infty} u_k = 0$. Suppose the claim were false. Then, once more by Lemma 4.2, there exists a subsequence, denoted again (k), such that

$$u_k \longrightarrow u \neq 0.$$

Case 1. Step 6 is executed for an infinite number of iterations. We begin by showing that the sequence of positive numbers (α_k) is bounded from above. Because $u_k \longrightarrow u \neq 0$, we can pick a subsequence, denoted once more by (k), such that

$\|u_k\|_2^2 \geq \delta$ for some $\delta > 0$. Now, using (3.5) and once more the definition of μ_k in step 5, it follows that

$$\alpha_k\|u_k\|' - 1 \leq \|g_k + \alpha_k u_k\|' = 1 + \alpha_k\mu_k = 1 + \alpha_k(\gamma_k - \|u_k\|_2^2/2)/\beta_k.$$

This shows that

$$\alpha_k(\beta_k\|u_k\|' - \gamma_k + \|u_k\|_2^2/2)/\beta_k \leq 2. \tag{4.2}$$

Recall that step 6 is executed only if

$$\gamma_k < \beta_k\|u_k\|' + \|u_k\|_2^2/4, \tag{4.3}$$

that is only if

$$\gamma_k - \beta_k\|u_k\|' < \|u_k\|_2^2/4.$$

Rewriting (4.2) as

$$\alpha_k(\beta_k\|u_k\|' - \gamma_k + \|u_k\|_2^2/2)/ \leq 2\beta_k$$

and combining with the above inequality, we have

$$\alpha_k\|u_k\|_2^2/2 \leq 2\beta_k + \alpha_k(\gamma_k - \beta_k\|u_k\|') < 2\beta_k + \alpha_k\|u_k\|_2^2/4$$

from which we get $\alpha_k\|u_k\|_2^2/4 < 2\beta_k$, so that $0 < \alpha_k < 8d/\delta$, since (β_k) is bounded from above by d as it was shown earlier and $\|u_k\|_2^2 \geq \delta$. Thus the sequence (α_k) is bounded. Passing to a further subsequence if necessary, we may assume that there exists $\alpha \geq 0$ such that $\alpha_k \longrightarrow \alpha$, as $k \longrightarrow \infty$.

If we let $k \longrightarrow \infty$ in (4.1), then by continuity of the map $z \mapsto z'$ on $\mathbb{R}^n\backslash\{0\}$ it follows that

$$\beta\langle(g + \alpha u)', u\rangle + \|u\|_2^2/2 \geq \gamma. \tag{4.4}$$

We distinguish two possibilities, $\alpha = 0$ and $\alpha > 0$ and show that these two cases both will lead as to conclude that $\lim u_k = 0$.

If $\alpha = 0$, then (4.4) becomes

$$\beta\langle g', u\rangle + \|u\|_2^2/2 \geq \gamma. \tag{4.5}$$

From the definition of u_k and a_k in step 1 of the algorithm, it follows that

$$\|u_k\|_2^2 = \langle x_{k+1} - a_k, u_k\rangle = \langle x_{k+1}, u_k\rangle - \langle a_k, u_k\rangle = \gamma_k - \beta_k\langle g_k', u_k\rangle.$$

Passing to the limit on both sides of the above relation leads to $\|u\|_2^2 = \gamma - \beta\langle g', u\rangle$. This with inequality (4.5) force u to satisfy $\|u\|_2^2/2 \leq 0$. So $u = 0$.

Suppose now that $\alpha > 0$. Using step 7 of the algorithm and allowing $k \longrightarrow \infty$, we get

$$\beta_{k+1} \longrightarrow (\beta + \frac{1}{2}\alpha\gamma)/\|g + \frac{1}{2}\alpha u)\|' = \hat{\beta}.$$

Note that $\beta_k = \langle y_k, b\rangle$ and $\beta_{k+1} = \langle y_{k+1}, b\rangle$ both have the same limit. So $\beta = \hat{\beta}$. This yields the equation

$$\beta\|g + \frac{1}{2}\alpha u\|' = \beta + \frac{1}{2}\alpha\gamma. \tag{4.6}$$

Once more allowing $k \longrightarrow \infty$ in the equation (3.5) defining α_k and using the definition of μ_k, we have

$$\beta\|g + \alpha u\|' = \beta + \alpha(\gamma - \frac{1}{2}\|u\|_2^2). \tag{4.7}$$

Applying the strict convexity of the dual norm $\|.\|'$ implies

$$\begin{aligned}\beta\|g+\frac{1}{2}\alpha u\|' &< \beta\left(\frac{1}{2}\|g\|'+\frac{1}{2}\|g+\alpha u\|'\right)=\beta\left(\frac{1}{2}+\frac{1}{2}\left[1+\alpha\left[\gamma-\frac{1}{2}\|u\|_2^2\right]/\beta\right]\right)\\ &= \beta+\frac{1}{2}\alpha\left(\gamma-\frac{1}{2}\|u\|_2^2\right).\end{aligned}$$

Inserting this in (4.6) shows that

$$\beta+\frac{1}{2}\alpha\gamma<\beta+\frac{1}{2}\alpha\left(\gamma-\frac{1}{2}\|u\|_2^2\right),$$

which implies $\|u\|_2^2/4 < 0$ leading to a contradiction so that the sequence (u_k) converges to zero, as sought.

Case 2. Assume that the condition in step 4 of the algorithm is satisfied for all, but a finite number of indices (k). As in the first case we proceed by contradiction by assuming that (u_k) does not converge to zero. So assume that there exists a subsequence denoted (k) again and a nonzero vector u such that

$$\lim_{k\to\infty} u_k = u \neq 0.$$

Since step 4 is answered affirmatively for all k, we have $\gamma_k \geq \beta_k\|u_k\|' + \frac{1}{4}\|u_k\|_2^2$, for all k. Letting $k \longrightarrow \infty$ in the above inequality implies

$$\gamma \geq \beta\|u\|' + \frac{1}{4}\|u\|_2^2. \tag{4.8}$$

In step 4, $\beta_{k+1} = \gamma_k/\|u_k\|'$, thus as $k \longrightarrow \infty$, this converges to $\beta = \gamma/\|u\|'$, which combined with inequality (4.8), leads to $\frac{1}{4}\|u\|_2^2 \leq 0$, a contradiction with the assumption made. We have thus proved that in all cases, the sequence (u_k) generated by the algorithm converges to zero.

It remains now to prove that the algorithm converges to the unique solution of problem (P). Let x^* be any cluster point of the sequence (x_k) and let $(x_{k'})$ be a subsequence converging to x^*. Writing the relation in step 1 of the algorithm for all k', we have

$$x_{k'+1} = u_{k'} + \beta_{k'} g'_{k'}, \tag{4.9}$$

where $g'_{k'} = A^T y_{k'} + \xi_{k'}$, $\|g_{k'}\|' = 1$ and $\beta_{k'} = \langle y_{k'}, b\rangle$, for all k', by virtu of Lemma 3.2. Since $\|g_{k'}\|' = 1$, by passing to a further subsequence that we denote (k'), we get $g_{k'} \longrightarrow g$, $\|g\|' = 1$. It has been proven earlier that $\lim u_{k'} = 0$, so if we let $k' \longrightarrow \infty$ in (4.8), then

$$x^* = \beta g',\ \ \|g\|' = 1. \tag{4.10}$$

Clearly, x^* is feasible since $K = \{x \in \mathbb{R}^n \mid x \geq 0,\ Ax = b\}$ is closed. In other words $Ax^* = b$ and $x^* \geq 0$.

From earlier discussion in the proof, we have seen that $\beta = \lim\langle y_{k'}, b\rangle$, where $y_{k'} \in \mathbb{R}^n$, $\xi \geq 0$ and $\|A^T y_{k'} + \xi_{k'}\|' = 1$. By the weak duality lemma and (4.9)

$$\langle y_{k'}, b\rangle \leq \|x^*\| \leq \beta.$$

Letting $k' \longrightarrow \infty$ in the above relation yields equality. This shows that every cluster point of the sequence (x_k) is a solution of problem (P). Due to the uniqueness of the solution, we conclude that (x_k) converges to the unique solution of (P). This completes the proof of the theorem. □

5. Numerical Implementation

We coded algorithm 3.1 in MatLab. To compare its performance to existing algorithms, the first implementation was performed using a numerical example (example 1) from [8]. The norm $\|\cdot\|$ used was the usual l^p-norm and the algorithm was implemented for various values of p. The minimization subproblems in Steps 0 and 1 were executed using a standard minimization routine in MatLab. Note that these steps require solving a quadratic program and therefore, any of the available codes for quadratic optimization could be used. For the stopping criterion of step 2, the duality gap discussed in section 3 was used. The results are tabulated and presented in Table 1.

Example 1 ([8]).

$$A = \begin{bmatrix} 3 & 1 & -1 & 0 & 0 \\ 4 & 3 & 0 & -1 & 0 \\ 1 & 2 & 0 & 0 & -1 \end{bmatrix} \qquad b = \begin{bmatrix} 3 \\ 6 \\ 2 \end{bmatrix}$$

Table 1: the values of p are shown in the first column. The number of iterations k is recorded in the second column and for comparison, the third column reports the number K of iteration in [8] in which the first 2 values were not calculated. Our objective value $\|x\|_p$ is shown in column 4 and for comparison the last column records the objective value in [8].

p	k	K	$\|x\|_p$	$\|x\|_p$ in [8]
10.0	89	-	0.919912	-
5.0	20	-	0.995628	-
4.0	18	3555	1.044519	1.044507
3.5	24	430	1.084034	1.084030
3.0	12	102	1.142350	1.142349
2.0	0	0	1.395230	1.395229
1.5	20	2	1.172643	1.726367
1.2	50	3	2.144088	2.143688
1.1	1000	3	2.357813	2.357813

These numerical results show that algorithm 3.1 performed extremely well and surpassed Nikolopoulos-Sreedharan results in [8] for values of p larger than 2. For example when $p = 4$, they found a solution after 3555 iterations while we only needed 18 iterations and for $p = 3.5$ and $p = 3$ they found a solution in 430 and 102 iterations respectively. We found it in 24 and 12 iterations respectively. However the situation is reversed for $p < 2$. For instance when $p = 1.2$, the number of iterations for a solution in [8] was 3 while we needed 50. The gap widens for $p = 1.1$, where algorithm 3.1 found a solution in 1000 iterations. The fact that we relied on MatLab existing solver to find α_k in step 6 of the algorithm may explain the relatively large number of iteration for values of $p < 2$. In fact we included step 6 as minimization of a subproblem where the equation in this step is considered as a constraint and the search is stopped once a feasible solution was found. This may not be the most efficient way to solve for α_k. A second advantage of the present algorithm is that it was able to solve problems with larger values of p than in [8] in which the highest p tested was 4.

Compared to existing literature where only relatively small problems were solved numerically, algorithm 3.1 was also tested numerically to solve relatively large size problems. Examples with hundreds of variables were solved. For each numerical experiment, a random matrix A is generated in MatLab. To guarantee the existence of at least one feasible solution, a random vector x is also generated and then the vector $b = Ax$ is computed. The algorithm converged for matrices of size 250×1000

and larger. The results were tabulated for a 250×1000 matrix and are presented in Table 2. The second column indicates the simulation time in seconds using a Pentium 4 with 1.83 GHz and 3 GB of main memory.

Table 2: the values of p are shown in the first column. The second column indicates the simulation time in seconds and the objective value is reported in the third column.

p	Time(sec)	$\|x\|_p$		p	Time(sec)	$\|x\|_p$
5.0	605.1	1.776275		1.9	410.8	13.937503
4.5	719.5	2.022385		1.8	310.2	16.867340
4.0	1232.3	2.391415		1.7	277.7	20.890271
3.5	948.1	2.987279		1.6	263.9	26.598117
3.0	1144.8	4.033336		1.5	267.4	35.008729
2.5	720.1	6.164389		1.4	265.5	47.986690
2.2	392.4	8.750690		1.1	595.1	198.626041

6. Conclusion

A common problem encountered in many applications is that of fining minimal solution of linear systems. In this paper we presented an algorithm for finding the vector of minimal solution of a system of linear equations. The feasibility and global convergence of the algorithm were proved for general strictly convex, smooth norms. This naturally, includes the ℓ^p-norm, $1 < p < \infty$. As a consequence, the algorithm in this paper covers the interesting case, where $1 < p < 2$. In this range, the ℓ^p-norm is not twice differentiable. The algorithm was implemented and tested, on various examples, when the objective function was the ℓ^p-norm, $1 < p < \infty$ and showed superior performance for values of p larger than 2 and good performance for p less than 2. The algorithm also performed well for relatively large size problem, which we have not seen covered in literature.

References

[1] Bahi, S. and Sreedharan, V.P., A least distance algorithm for a smooth strictly convex norm, Int. J. Computer Math., 75 (2000), 445-463.

[2] Bahi, S. and Sreedharan, V.P., An algorithm for a minimum norm solution of a system of linear inequalities, Int. J. Computer Math., 80 (2003), 639-647.

[3] Boyd, S. and Vandenberghe, L., Convex Optimization, Cambridge University Press, 2004.

[4] Borwein, J. M., Maximum entropy-type methods and (non-)convex programming, Atlantic Optimization Days. URL: www.cs.dal.ca/ jbrowein, 2006.

[5] Cheney, E.W., Introduction to approximation theory, 2nd ed., McGraw-Hill, New York, 1982.

[6] Donoho, D. L., For most large underdetermined systems of linear equations the minimal l_1-norm solution is also the sparsest solution, Comm. Pure Appl. Math., Vol. 6 (2006), 797-829.

[7] Lawson, C.L. and Hanson, R.J., Solving Least Squares Problems, Prentice Hall, Englewood Cliffs, NJ, 1974.

[8] Nikolopoulos, P.V. and Sreedharan, V.P., An algorithm for computing non negative minimal norm solutions, Numer. Funct. Anal. Optimiz., 15 (1994), 87-103.

[9] Nirenberg, L., Functional Analysis, Lectures given in 1960-61, notes by Lesley Sibner, New York University, 1961.

[10] Späth, H. Mathematical Algorithms for Linear Regression, Academic Press, New York, 1992.

[11] Sreedharan, V.P., An algorithm for a non negative norm minimal solutions, Numer. Funct. Anal. Optimiz., 9 (1987), 193-232.

Department of Mathematics, 351 West University Boulevard, Southern Utah University, Cedar City, UT 84720

E-mail: bahi@suu.edu and aross83@yahoo.com