

# On the Incomplete Oblique Projections Method for Solving Box Constrained Least Squares Problems

H. Scolnik<sup>a</sup>, N. Echebest<sup>b</sup>, M. T. Guardarucci<sup>c</sup>

<sup>a</sup>*Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Argentina(hugo@dc.uba.ar)*

<sup>b</sup>*Departamento de Matemática, Facultad de Ciencias Exactas, Universidad Nacional de La Plata, CP 152, 50 y 115, La Plata (1900), Argentina (opti@mate.unlp.edu.ar).*

<sup>c</sup>*Departamento de Ciencias Básicas, Facultad de Ingeniería, Universidad Nacional de La Plata, Argentina(marite.guardarucci@ing.unlp.edu.ar)*

---

## Abstract

The aim of this paper is to extend the applicability of the incomplete oblique projections method (IOP) previously introduced by the authors for solving inconsistent linear systems to the box constrained case. The new algorithm employs incomplete projections onto the set of solutions of the augmented system  $Ax-r=b$ , together with the box constraints, based on a scheme similar to the one of IOP, adding the conditions for accepting an approximate solution in the box. The theoretical properties of the new algorithm are analyzed, and numerical experiences are presented comparing its performance with some well-known methods.

*Keywords:* Inconsistent systems, box constrained, incomplete projections.

*2000 MSC:* 65F10

---

## 1. Introduction

Large and sparse systems of linear equations arise in many important applications [5, 24], as image reconstruction from projections, radiation therapy treatments planning, computational mechanics and optimization problems. In practice, those systems are often inconsistent, and one usually seeks a point  $x^* \in \mathfrak{R}^n$ ,  $l_i \leq x_i^* \leq u_i$ ,  $i = 1, \dots, n$ , that minimizes a certain proximity function. In this paper, for such possibly inconsistent systems we will consider the standard problem

$$\min_{x \in \mathcal{B}} \|Ax - b\|_{D_m}^2, \quad (1)$$

where  $\mathcal{B} = \{x : x \in \mathfrak{R}^n, l_i \leq x_i \leq u_i, i = 1, \dots, n\}$ ,  $A \in \mathfrak{R}^{m \times n}$ ,  $b \in \mathfrak{R}^m$ ,  $m \geq n$ .  $\|\cdot\|_{D_m}$  denotes the norm induced by the positive definite diagonal matrix  $D_m \in \mathfrak{R}^{m \times m}$ .

In [19] we have introduced, for inconsistent problems  $Ax = b$ , the IOP algorithm that converges to a weighted least squares solution of that system. That algorithm uses an incomplete oblique projections scheme onto the solution set of the augmented system  $Ax - r = b$ , in order to solve  $\min_{x \in \mathfrak{R}^n} \|Ax - b\|_{D_m}^2$ . We proved that this problem is equivalent to the problem [19]:

$$\min\{\|p - q\|_D^2 : \text{for all } p \in \mathcal{P} \text{ and } q \in \mathcal{Q}\}, \quad (2)$$

being  $\mathcal{P} = \{p : p = [x; r] \in \mathfrak{R}^{n+m}, x \in \mathfrak{R}^n, r \in \mathfrak{R}^m, Ax - r = b\}$ , and  $\mathcal{Q} = \{q : q = [x; 0] \in \mathfrak{R}^{n+m}, x \in \mathfrak{R}^n, 0 \in \mathfrak{R}^m\}$ .  $D$  is a diagonal matrix of order  $n + m$ , whose  $n$  first elements are 1's, and the last  $m$  coincide with those of  $D_m$ . That result led us to develop the IOP method for solving (2), applying an alternate projections scheme between the sets  $\mathcal{P}$  and  $\mathcal{Q}$ , similar to the one of Csiszár and Tusnády [10], but replacing the computation of the exact projections onto  $\mathcal{P}$  by suitable incomplete or approximate projections. In [21] the results of [19] were generalized for solving rank deficient problems.

In this paper we add to the IOP alternating algorithm the condition  $x \in \mathcal{B}$ ; for that purpose we define the sets:

$$\mathcal{P}^b = \mathcal{P} \cap \{(x; r) : x \in \mathcal{B}\} \quad \text{and} \quad \mathcal{Q}^b = \mathcal{Q} \cap \{(x; 0) : x \in \mathcal{B}\}, \quad (3)$$

adopting the distance  $d(p, q) = \|p - q\|_D$ , for all  $p \in \mathcal{P}^b, q \in \mathcal{Q}^b$ .

We consider the following basic scheme:

**Algorithm 1.** (*Basic Alternating Scheme*)

Given feasible vectors  $p^k = [x^k; r^k] \in \mathcal{P}^b, q^k = [x^k; 0] \in \mathcal{Q}^b$ .

Let  $p = [x; r], \hat{p} = [\hat{x}; \hat{r}]$ .

Define

$$\begin{aligned} \hat{p} &= \operatorname{argmin}_{p \in \mathcal{P}^b} g_k(p), \quad \text{where} \\ g_k(p) &= \|x - x^k\|^2 + \|r\|_{D_m}^2, \\ \mathcal{P}^b &= \{p : x \in \mathcal{B}, r = Ax - b\}. \end{aligned} \quad (4)$$

Define  $p^{k+1} = [x^{k+1}; r^{k+1}] = [\hat{x}; \hat{r}]$ , and  $q^{k+1} = [x^{k+1}; 0]$

In each step of Algorithm 1 a minimum norm problem with linear constraints (4) needs to be solved. For that purpose we use the HLWB algorithm [9]. This inner iteration needs of course to be terminated at some step. Therefore in practice we must use instead of  $\hat{p}$  some approximation of it.

In the following sections we present the extended *BIOP* algorithm based on the basic scheme of Algorithm 1, adding the conditions for accepting an approximate solution of (4). The theoretical properties of the new algorithm are analyzed in Section 3, and numerical experiments are presented in Section 4 comparing its performance with some known methods.

**Notations.**

- $\|\cdot\|$  denotes the Euclidean norm, and  $\|\cdot\|_D$  the norm induced by a positive definite matrix  $D$ .
- $I_n$  denotes the identity matrix in  $\Re^{n \times n}$ , and  $\mathbf{e}_i$ , the  $i$ -th column of  $I_n$ .
- The upper index T stands for the transpose of a matrix.
- $D = \text{diag}(d)$ , denotes a diagonal matrix of order  $n$ , whose diagonal is  $d = (d_1, \dots, d_n)$ .
- $P(q^k)$  will denote the exact solution  $\hat{p} = [\hat{x}, \hat{r}]$  of problem (4), where  $q^k = [x^k; 0]$ .

**2. Some Basic Results**

The convex feasibility problem (CFP) is to find a point (any point) in the nonempty intersection  $C = \bigcap_{i=1}^{m'} C_i \neq \emptyset$ , of a family of closed convex subsets  $C_i \subset \Re^{n'}$ ,  $i = 1, 2, \dots, m'$  of the  $n'$ -dimensional Euclidean space.

Projection algorithms employ projections onto the individual convex sets in order to reach the required point in the intersection. They employ projections onto the individual sets in various ways [5, 6, 7, 12, 18]. In the sequel we denote by  $P_i$  the orthogonal projection onto  $C_i$ , and  $P_i^D$  the oblique projection onto  $C_i$ , where  $D$  is a positive definite matrix of  $\Re^{n' \times n'}$ .

The best approximation problem (BAP) is to find the projection of a given point  $y \in \Re^{n'}$  onto the nonempty intersection  $C$ . Thus, it seeks a point in the intersection of the convex sets which is closest to the point  $y$ .

Aiming at clarifying the applicability of HLWB within the new approach, it is convenient to point out that, it solves the BAP problem [8, 9]. It computes the simultaneous projections onto each set, takes a convex combination of the intermediate points, and defines the next iterate.

Next, the simultaneous HLWB algorithm [8] is presented for the sake of completeness.

**Algorithm 2.** *Initialization:* Let  $y_0 = y$  be the given point  $y$  whose projection  $P_C(y)$  onto  $C = \bigcap_{i=1}^{m'} C_i \neq \emptyset$  is sought after by the BAP. Let  $\{\sigma_j\}_{j=0}^{\infty}$  be a user chosen steering sequence and let  $\{w_i\}_{i=1}^{m'}$ , be real positive numbers (weights) such that  $\sum_{i=1}^{m'} w_i = 1$ .

*Iterative step:* Take the steering parameter  $\sigma_j$  from the chosen steering sequence.

Given the current iterate  $y_j$ , calculate  $T(y_j) = \sum_{i=1}^{m'} w_i P_i^D(y_j)$  and the next iterate:

$$y_{j+1} = \sigma_j y^0 + (1 - \sigma_j) T(y_j).$$

The sequence  $\{\sigma_j\}$  must satisfy the following conditions:

- (i)  $0 \leq \sigma_j < 1$ ; (ii)  $\lim_{j \rightarrow \infty} \sigma_j = 0$ ; (iii)  $\sum_{j=0}^{\infty} \sigma_j = \infty$  and
- (iv)  $\sum_{j=0}^{\infty} |\sigma_j - \sigma_{j+q}| < \infty$ ,  $q > 0$ .

In the new algorithm, HLWB will be used for obtaining an approximation to the solution of problem (4), where the former convex set  $C$  will be the intersection of  $Az - r = b$ , with  $l_i \leq z_i \leq u_i$ ,  $i = 1, \dots, n$ , and the initial point at the  $k$ -th iteration of Algorithm 1, will be  $y_0 = [x^k; 0]$ .

For each constraint of  $Az - r = b$ , we will denote by  $C_i = \{y = [z; r] \in \mathfrak{R}^{n+m} : \mathbf{a}_i^T z - \mathbf{e}_i^T r = b_i\}$ ,  $i = 1, \dots, m$ .

Given an arbitrary  $y = [z; r] \in \mathfrak{R}^{n+m}$ , we denote for each  $i = 1, \dots, m$ ,  $r_i(y) = \mathbf{a}_i^T z - \mathbf{e}_i^T r - b_i$ ,  $\bar{\mathbf{a}}_i^T = [\mathbf{a}_i; -\mathbf{e}_i]^T$ ,  $\mathbf{e}_i \in \mathfrak{R}^m$ , and the oblique projection of  $y$  onto  $C_i$ , by

$$P_i^D(y) = y - \frac{r_i(y)}{\bar{\mathbf{a}}_i^T D^{-1} \bar{\mathbf{a}}_i} D^{-1} \bar{\mathbf{a}}_i. \quad (5)$$

We define  $C_{m+i} = \{y = [z; r] \in \mathfrak{R}^{n+m} : l_i \leq z_i \leq u_i\}$ , for  $i = 1, 2, \dots, n$ .

Given an arbitrary  $y = [z; r] \in \mathfrak{R}^{n+m}$ , for each  $i = 1, \dots, n$  we denote  $r_{m+i}(y) = \max(0, \max(z_i - u_i, l_i - z_i))$ , and  $P_{m+i}^D(y)$  is calculated by

$$P_{m+i}^D(y) = y - \text{sgn}(z_i - l_i) \frac{r_{m+i}(y)}{\|\mathbf{e}_i\|_{D^{-1}}^2} D^{-1} \mathbf{e}_i, \text{ where } \mathbf{e}_i \in \mathfrak{R}^{n+m}. \quad (6)$$

In particular, in this paper we will use a diagonal weighting matrix  $D = \begin{pmatrix} I_n & 0 \\ 0 & D_m \end{pmatrix}$ , therefore the last previous projections will be computed by  $P_{m+i}^D(y) = y - \text{sgn}(z_i - l_i)r_{m+i}(y)\mathbf{e}_i$ ,  $i = 1, \dots, n$ , where  $\mathbf{e}_i \in \mathfrak{R}^{n+m}$ .

Analogously, the projections (5) onto  $C_i$ ,  $i = 1, \dots, m$ , will be computed by

$$P_i^D(y) = y - \frac{r_i(y)}{\|\mathbf{a}_i\|^2 + \mathbf{e}_i^T D_m^{-1} \mathbf{e}_i} \begin{pmatrix} \mathbf{a}_i \\ D_m^{-1}(-\mathbf{e}_i) \end{pmatrix}, \text{ where } \mathbf{e}_i \in \mathfrak{R}^m. \quad (7)$$

### 3. Incomplete Oblique Projections Algorithm

As said in the Introduction, for possibly inconsistent systems  $Ax = b$ ,  $x \in \mathcal{B}$ ,  $A \in \mathfrak{R}^{m \times n}$ ,  $b \in \mathfrak{R}^m$ ,  $m \geq n$ , we will consider the standard problem (1). We will follow the alternating scheme of Algorithm 1 for solving the equivalent problem that minimizes the distance between the sets described in (3). For that purpose we will use a diagonal weighting matrix  $D \in \mathfrak{R}^{(n+m) \times (n+m)}$ , as defined in the previous Section.

Given  $p^k = [x^k; r^k] \in \mathcal{P}^b$ , and its projection  $q^k = [x^k; 0]$  onto  $\mathcal{Q}^b$ , we will denote by  $P(q^k)$  the projection of  $q^k$  onto  $\mathcal{P}^b$ , which is the solution of problem (4).

As mentioned before, instead of defining  $p^{k+1}$  as the exact solution of (4), we define it by means of the incomplete resolution of that problem.

In order to define an inexact projection, we consider the following:

**Definition 1.** *Given an approximation  $[z; v]$ ,  $z \in \mathfrak{R}^n$ ,  $v \in \mathfrak{R}^m$ , of  $P(q^k)$ , we will denote by  $P^r([z; v]) = [z^B; r(z^B)]$ , where  $z^B$  is the projection of  $z$  onto half spaces  $x_i \geq l_i$ , or  $x_i \leq u_i$ ,  $i = 1, \dots, n$ , and  $r(z^B) = Az^B - b$ .*

Aiming at obtaining properties of the sequence  $\{p^k\}$  generated by the new algorithm that guarantees convergence to the solution of (1) we establish an “acceptance condition” that an approximation to  $P(q^k)$  must satisfy, using the iterative HLWB algorithm for solving (4).

**Definition 2.** *Acceptance Condition. Given  $0 < \beta_k < 1$ ,  $0 < \gamma < 1$  an approximation  $p_j = [z_j; v_j]$  of the solution of (4) is acceptable, considering the inexact projection  $P^r(p_j) = [z_j^B; r(z_j^B)]$ , if it satisfies:*

$$(i) \|p_j - T(p_{j-1})\|_D \leq \beta_k \|q^k - T(p_{j-1})\|_D. \quad (8)$$

$$(ii) \|P^r(p_j) - q^k\|_D^2 < \|p^k - q^k\|_D^2, \text{ and} \\ \|P^r(p_j) - p_j\|_D^2 \leq \gamma \|P^r(p_j) - p^k\|_D^2. \quad (9)$$

**Lemma 1.** *Applying the HLWB algorithm for solving (4) it is possible to find an iterate  $p_j = [z_j, v_j]$  satisfying conditions (8) and (9).*

*Proof:* Since the sequence obtained by HLWB converges to the solution of (4), the sequence  $\{p_j\} = \{[z_j; v_j]\}$  tends to  $P(q^k) = [\hat{z}; \hat{v}]$ , then  $\|P^r(p_j) - P(q^k)\|_D^2$  goes to zero when  $\sigma_j$  tends to zero.

By definition  $p_j$  satisfies  $\|p_j - T(p_{j-1})\|_D = \sigma_j \|q^k - T(p_{j-1})\|_D$ . Then, given  $0 < \beta_k < 1$ , as  $\sigma_j$  tends to zero, a  $\sigma_j$  exists such that  $\sigma_j < \beta_k$ . Hence, it is possible to satisfy condition (8).

If  $P(q^k) \neq p^k$ ,  $\|P(q^k) - q^k\|_D^2 < \|p^k - q^k\|_D^2$ , and considering that  $\|p_j - P(q^k)\|_D^2$  tends to zero, there exists an  $j$  such that  $\|p_j - q^k\|_D^2 < \|p^k - q^k\|_D^2$ . Furthermore, considering that the sequences  $\|P^r(p_j) - P(q^k)\|_D^2$  and  $\|P^r(p_j) - p_j\|_D^2$  go to zero, it is possible to satisfy condition (9).  $\square$

Our proposal is to replace in the new algorithm the exact solution  $P(q^k)$  of (4) by  $P^r(p_j)$  if it satisfies the above conditions. Thus, we will define

$$p^{k+1} = P^r(p_j), \text{ if } p_j = [z_j; v_j] \text{ satisfies conditions (8) and (9).} \quad (10)$$

**Notation.** In the following, the index  $j$  corresponding to the accepted approximation  $p_j = [z_j, v_j]$  will be denoted by  $j_k$  to indicate that it corresponds to the  $k$ -th iteration of the new algorithm.

**Algorithm 3.** *Bounded Incomplete Oblique Projections (BIOP)*

**Initialization:** *Given  $\gamma$ ,  $0 < \gamma < 1$ , a positive definite diagonal matrix  $D_m$  of order  $m$  and  $\{\beta_k\}_0^\infty$ , such that  $\beta_k > 0$  and  $\beta_k$  tends to zero, if  $k$  tends to  $\infty$ .*

*Given  $p^0 = [x^0; r^0]$ ,  $x^0 \in \mathcal{B}$ ,  $r^0 = Ax^0 - b$  and  $q^0 = [x^0; 0] \in \mathcal{Q}^b$ .*

*Set  $k \leftarrow 0$ .*

**Iterative Step:** *Given  $p^k = [x^k; r^k]$ , set  $q^k = [x^k; 0]$ ,  $x^k \in \mathcal{B}$ ,  $\beta_k > 0$ .*

- Calculate an approximation to the solution of (4) satisfying the conditions of definition 2, applying HLWB from the initial point  $y_0 = [z_0; v_0] = q^k$ , and using the sequence  $\{\sigma_j\}$ :

- Compute for  $j = 1, 2, \dots$ ,  $y_j = \sigma_j y_0 + (1 - \sigma_j)T(y_{j-1})$ , until finding an index  $j_k$  such that  $y_{j_k} = [z_{j_k}, v_{j_k}]$  and  $Pr(y_{j_k})$  satisfy conditions (8) and (9).

- Define  $p^{k+1} = [x^{k+1}; r^{k+1}] = Pr([z_{j_k}, v_{j_k}])$ .
- If  $[x^{k+1}; r^{k+1}]$  satisfies the stopping condition,  $\| \|r^{k+1}\| - \|r^k\| \| < \epsilon \max(\|r^0\|, 1)$ , define  $x^* = x^{k+1}$  and  $r^* = r^{k+1}$ . STOP
- Define  $q^{k+1} = [x^{k+1}; 0] \in \mathcal{Q}^b$ .
- $k \leftarrow k + 1$ .

### 3.1. Convergence of the BIOP algorithm

We will consider the set of solutions to the problem (1):

$$L_D^{sq} = \{x^* \in \mathcal{B} : \text{for which } r^* = Ax^* - b \text{ satisfies } A^T D_m r^* = \mu^*,\} \quad (11)$$

such that  $\mu_i^* \geq 0$  if  $x_i^* = l_i$ , and  $\mu_i^* \leq 0$ , if  $u_i = x_i^*$ , otherwise  $\mu_i^* = 0$ , when  $l_i < x_i^* < u_i$ .

The corresponding set in  $\mathfrak{R}^{m+n}$ :

$$\mathcal{S}_D = \{p^* : p^* = [x^*; r^*] \in \mathcal{P}^b \text{ such that } x^* \in L_D^{sq}\}. \quad (12)$$

In the following we analyze the convergence properties of the sequence given by Algorithm 3 studying the relationship between two consecutive iterates  $p^k$  and  $p^{k+1}$ .

**Lemma 2.** *Let  $\{p^k\} = \{[x^k; r^k]\}$  be the sequence generated by the Algorithm 3, then*

(i)  $p^k = [x^k; r^k]$  and  $p^{k+1} = [x^{k+1}; r^{k+1}]$  satisfy

$$\|r^{k+1}\|_{D_m}^2 \leq \|r^k\|_{D_m}^2 - \|x^{k+1} - x^k\|^2.$$

(ii) The sequence  $\{\|r^k\|_{D_m}\}$  is decreasing and bounded, therefore it converges.

(iii) The following three sequences tend to zero:  $\{\|p^{k+1} - p^k\|_D^2\}$ ,  $\{\|p^{k+1} - P(q_k)\|_D^2\}$  and  $\{\|p^k - P(q_k)\|_D^2\}$ , where  $P(q_k)$  is the exact solution of (4).

(iv) The sequence  $\{A^T D_m r^k - \mu^k\}$  goes to zero,  $\mu^k$  being the vector of the Karush-Kuhn Tucker (KKT) multipliers associated to the bound constraints at the solution of the problem (4).

(v) The sequence  $\{\mu_i^k * \min(x_i^k - l_i, u_i - x_i^k)\}$  goes to zero.

*Proof* We get, as a consequence of the definition of  $p^{k+1} = [x^{k+1}; r^{k+1}]$ , that  $p^{k+1} = Pr([z_{j_k}; v_{j_k}])$  satisfies the condition (9),  $\|p^{k+1} - q^k\|_D^2 < \|p^k - q^k\|_D^2$ , then it follows that  $\|r^{k+1}\|_{D_m}^2$  satisfies  $\|r^{k+1}\|_{D_m}^2 + \|x^{k+1} - x^k\|^2 < \|r^k\|_{D_m}^2$  from which it follows that  $\|r^{k+1}\|_{D_m}^2 < \|r^k\|_{D_m}^2 - \|x^{k+1} - x^k\|^2$ . Thus, the sequence  $\{\|r^k\|_{D_m}\}$  is decreasing and bounded, and therefore it converges. Hence,  $\|x^{k+1} - x^k\|^2$  tends to zero. As a consequence of that, we also obtain that  $\{\|p^{k+1} - p^k\|_D^2\}$  goes to zero.

By the conditions required in Definition 2 to accept  $p_{j_k}$  and  $Pr(p_{j_k}) = [z_{j_k}^B; v_{j_k}^B]$  we know that  $\|Pr(p_{j_k}) - p_{j_k}\|_D \leq \gamma \|p^k - p^{k+1}\|_D^2$ , then  $\|Pr(p_{j_k}) - p_{j_k}\|_D$  tends to zero.

Also, from the conditions in Definition 2 to accept  $p_{j_k} = [z_{j_k}; v_{j_k}]$ , we obtain that  $\|p_{j_k} - T(p_{j_k-1})\|_D = \sigma_{j_k} \|q^k - T(p_{j_k-1})\|_D \leq \beta_k \|q^k - T(p_{j_k-1})\|_D$ . Thus, the sequence of parameters  $\{\sigma_{j_k}\}$  converges to zero because  $\{\beta_k\}$  tends to zero. Therefore, from the properties of the HLWB algorithm [9],  $\|p_{j_k} - P(q^k)\|_D$  tends to zero, when  $k$  tends to  $\infty$ . Hence, we also obtain that  $\|p^{k+1} - P(q^k)\|_D^2$  tends to zero. Then, also  $\|p^k - P(q^k)\|_D^2$  tends to zero.

In order to prove (iv), the components of the exact solution  $P(q^k)$  will be denoted by  $[\hat{x}^k; \hat{r}^k]$ , to exhibit their dependence of the k-th iterative step.

The solution of a convex problem with linear equalities and inequalities constraints satisfies KKT conditions [5]. Therefore, the exact solution  $P(q^k) = [\hat{x}^k; \hat{r}^k]$  of problem (4), satisfies

$$\hat{x}^k - x^k = -A^T D_m \hat{r}^k + \mu^k, \quad (13)$$

where  $\mu^k$  is the vector of multipliers associated to the bound constraints at  $P(q^k)$ . It is known that, if  $l_i < \hat{x}_i^k < u_i$  then  $\mu_i^k = 0$ . Otherwise, if  $\hat{x}_i^k = l_i$ ,  $\mu_i^k \geq 0$ , and if  $\hat{x}_i^k = u_i$  then  $\mu_i^k \leq 0$ . Then, since  $\|p^k - P(q^k)\|_D^2 = \|x^k - \hat{x}^k\|^2 + \|r^k - \hat{r}^k\|_{D_m}^2$  tends to zero, considering the results in (iii), it follows that both  $\|A^T D_m \hat{r}^k - \mu^k\|$  and  $\|r^k - \hat{r}^k\|_{D_m}$  tend to zero. Therefore, since  $\|A^T D_m r^k - \mu^k\| = \|A^T D_m (r^k - \hat{r}^k) + A^T D_m \hat{r}^k - \mu^k\|$ , we get that  $\|A^T D_m r^k - \mu^k\|$  goes to zero. Thus, (iv) follows.

Furthermore, as a consequence of the previous result it turns out that  $\mu^k$  is bounded, and considering that  $\|x^k - \hat{x}^k\|$  tends to zero, we get that  $(\mu^k)^T (x^k - \hat{x}^k)$  also tends to zero.

In order to prove (v), it is known that for each  $i \in \{1, 2, \dots, n\}$  such that  $l_i < \hat{x}_i^k < u_i$ , the product  $\mu_i^k \min(x_i^k - l_i, u_i - x_i^k)$  is zero, because  $\mu_i^k = 0$ . Also, if  $\hat{x}_i^k = l_i$ , the product  $\mu_i^k \min(x_i^k - l_i, u_i - x_i^k)$  coincides with  $\mu_i^k \min(x_i^k - \hat{x}_i^k, u_i - \hat{x}_i^k + \hat{x}_i^k - x_i^k)$ , then since  $x_i^k - \hat{x}_i^k$  goes to zero, an index  $k_l$  exists such that for  $k > k_l$ ,  $\mu_i^k \min(x_i^k - l_i, u_i - x_i^k) = \mu_i^k (x_i^k - \hat{x}_i^k)$  that we know that goes to zero. Likewise, if  $\hat{x}_i^k = u_i$ , the product  $\mu_i^k \min(x_i^k - l_i, u_i - x_i^k)$  coincides with  $\mu_i^k \min(x_i^k - \hat{x}_i^k + \hat{x}_i^k - l_i, \hat{x}_i^k - x_i^k)$ . Hence, since  $x_i^k - \hat{x}_i^k$  converges to zero, an index  $k_u$  exists such that for  $k > k_u$  the product is equal to  $\mu_i^k (\hat{x}_i^k - x_i^k)$ . Hence, we get (v).  $\square$

**Lemma 3.** *Let  $\{p^k\} = \{[x^k; r^k]\}$  be the sequence generated by the Algorithm 3, then*

(i) *There exists a subsequence of  $\{p^k\}$ , denoted  $\{p^{k_s}\}_{s=1}^\infty = \{[x^{k_s}; r^{k_s}]\}$ , satisfying  $r^{k_s} = Ax^{k_s} - b$ , convergent to  $[\bar{x}; \bar{r}] \in \mathcal{S}_D$ .*

*Furthermore,*

(ii) *Every limit point of  $\{p^k\}$  is in  $\mathcal{S}_D$ .*

(iii) *If  $\mathcal{S}_D$  is a singleton set the sequence  $\{[x^k; r^k]\}$  converges to  $p^*$ ,  $p^* \in \mathcal{S}_D$ .*

*Proof* Considering that  $x^k \in \mathcal{B}$ , we know a subsequence  $\{x^{k_s}\}$  exists convergent to  $\bar{x}$  in  $\mathcal{B}$ . Hence, the subsequence  $\{p^{k_s}\} = \{[x^{k_s}; r^{k_s}]\}$ , satisfying  $r^{k_s} = Ax^{k_s} - b$ , is convergent to  $[\bar{x}; \bar{r}] \in \mathcal{P}^b$ . By (iv) of the Lemma 2, we know that  $A^T D_m r^{k_s} - \mu^{k_s}$  tends to zero. Furthermore, by (iii) of the Lemma 2  $\|x^{k_s} - \hat{x}^{k_s}\|$  and  $\|r^{k_s} - \hat{r}^{k_s}\|_{D_m}$  tend to zero, then  $\|\hat{x}^{k_s} - \bar{x}\|$  tends to zero, as well as  $\|\hat{r}^{k_s} - \bar{r}\|_{D_m}$ .

Therefore,  $\mu^{k_s} = x^{k_s} - \hat{x}^{k_s} + A^T D_m (\hat{r}^{k_s})$  converges to  $\bar{\mu} = A^T D_m (\bar{r})$ , satisfying  $\bar{r} = A\bar{x} - b$ . Moreover, by the KKT conditions of problem (4),  $\mu_i^{k_s} \min(\hat{x}_i^{k_s} - l_i, u_i - \hat{x}_i^{k_s}) = 0$ , with  $\mu_i^{k_s} \geq 0$  if  $\hat{x}_i^{k_s} = l_i$ , and  $\mu_i^{k_s} \leq 0$  if  $\hat{x}_i^{k_s} = u_i$ . Then, it follows that if  $\bar{x}_i$  satisfies  $l_i < \bar{x}_i < u_i$ , there exists  $k_i$ , such that for  $k_s > k_i$ ,  $|\hat{x}_i^{k_s} - \bar{x}_i| < 0.5 * \min(\bar{x}_i - l_i, u_i - \bar{x}_i)$ . Hence, for  $k_s > k_i$ ,  $\mu_i^{k_s} = 0$ , so we get that  $\bar{\mu}_i = 0$ . If  $\bar{x}_i$ , satisfies  $\bar{x}_i = l_i$ , since  $|\hat{x}_i^{k_s} - \bar{x}_i|$  tends to zero, the same happens with  $|\hat{x}_i^{k_s} - l_i|$ , and therefore there exists  $k_i^L$ , such that for  $k_s > k_i^L$ ,  $|\hat{x}_i^{k_s} - l_i| < 0.5 * |u_i - l_i|$ , then  $\mu_i^{k_s} \geq 0$ , and it follows that  $\bar{\mu}_i \geq 0$ . Likewise we deduce that if  $\bar{x}_i = u_i$ , then  $\bar{\mu}_i \leq 0$ .

Hence,  $\bar{\mu}_i \min(\bar{x}_i - l_i, u_i - \bar{x}_i) = 0$ , with  $\bar{\mu}_i \geq 0$  if  $\bar{x}_i - l_i = 0$ , and  $\bar{\mu}_i = 0$  if  $l_i < \bar{x}_i < u_i$ , and  $\bar{\mu}_i \leq 0$  when  $\bar{x}_i = u_i$ , taking into account that  $\hat{x}^{k_s}$  tends to  $\bar{x}$ . As a consequence of the last result,  $[\bar{x}; \bar{r}] \in \mathcal{P}^b$ , satisfies  $[\bar{x}; \bar{r}] \in \mathcal{S}_D$

Therefore,  $[\bar{x}; \bar{r}]$  satisfies the optimality conditions of the problem (1). Then, (i) follows.

From the proof of (i), we get (ii). Every convergent subsequence has its limit point in  $\mathcal{S}_D$ .

In order to prove (iii), under the hypothesis that the set  $\mathcal{S}_D$  has only one element, we consider that the domain is bounded, and since the sequence  $\|x^{k+1} - x^k\|$  goes to zero, applying Theorem 14.1.3 in J.M. Ortega and W. C. Rheinboldt [15], we obtain that the sequence  $\{[x^k; r^k]\}$  converges to  $p^* \in \mathcal{S}_D$ .  $\square$

## 4. Numerical Experiments

The objectives of the following experiments are two-fold. First we compare our algorithm with other methods, in relation to the rate of decrease of the norm of the residual, reporting the number of iterations needed for satisfying the convergence conditions and the corresponding CPU time. In the second part we analyze the *BIOP* behavior when solving image reconstruction problems comparing it with IOP (Scolnik et al. (2008a)). All visualizations have been obtained using MATLAB 7.0

For the first purpose, we have implemented our algorithm in Fortran 77 and MATLAB 7. We made some comparisons of the *BIOP* algorithm with BVLS, implemented in Fortran 77, which solves linear least-squares problems with upper and lower bounds on the variables, using an active set strategy. It is documented by Stark and Parker (1995) [23]. We have also compared our algorithm with one method of the SLS system, a software in MATLAB for sparse least squares problems. That method is SBLS2, for solving sparse box constrained least squares problems, in the version of Mikael Lundquist, Linkoping University, Sweden, <http://www.mai.liu.se/milun/sls>, which computes the solution by a version of the block principal pivoting algorithm described by Portugal et al. in [17].

### 4.1. Test Problems

The problems used for these numerical experiments are some from the RRA (real, rectangular, assembled) part of Harwell-Boeing Collection. Those least squares problems are from the set LSQ. In particular we used four matrices, such that the second and fourth matrix have the same pattern as the first and third respectively but are much more ill-conditioned. The matrices in this set are: WELL1033 and ILLC1033(real unsymmetric, 1033

Problem	$m \times n$
A1	$144 \times 144$
A2	$900 \times 900$
B1	$3452 \times 1025$
B4	$896 \times 625$
B6	$896 \times 625$
B7	$27376 \times 9025$

Table 1: Dimensions

by 320, 4732 entries), WELL1850 and ILLC1850 (real unsymmetric, 1850 by 712, 8758 entries). Also, aiming at testing problems with larger dimensions, we tested some randomly generated dense matrices with DQRT15(LAPACK) which, using the parameter RKSEL = 2, generates a rank deficient matrix  $A(m \times n)$  such that  $m$  is the number of rows and  $n$  is the number of columns of  $A$ , and  $\text{rank} = \frac{3}{4} \min(m, n)$ .

Other systems arise from the two image reconstruction problems used by Popa and Zdunek in [16] and other problems from the SNARK System [2]. The first simulate real objects in electro-magnetic geotomography, leading to problems whose data comes from projections made with a limited angular range. Those problems,  $A1$  and  $A2$ , lead to inconsistent systems, and the corresponding matrix has deficient rank due to the angle limitations of the projections. Also aiming at testing the algorithm with larger dimensions, we considered other problems from the SNARK System, with  $m \geq n$ . They are  $B1$ ,  $B4$ ,  $B6$  and  $B7$ . The dimensions of these matrices are showed in Table 1. For  $A1$  and  $A2$  we analyze the results with a system  $Ax = b + \delta b$ , arising from simulating noisy perturbations of the right hand side  $b = Ax^{exact}$ . The problems here analyzed exactly correspond to those mentioned as case 2 in [16]. Starting from the knowledge of  $b = Ax^{exact}$ , a perturbation  $\delta b$  is defined satisfying  $\|\delta b\|/\|b\| \approx 5.5\%$ . Since  $\delta b = \delta b_A + \delta b_{A^\perp}$ , where  $\delta b_A \in R(A)$  and  $\delta b_{A^\perp} \in R(A)^\perp$ , where  $R(A)$  denotes the subspace spanned by the columns of  $A$ , and  $R(A)^\perp$  stands for the orthogonal subspace to  $R(A)$ . Those perturbations are applied to each problem according to:

$$A1 \text{ case (2): } \|\delta b_{R(A1)}\| = \|\delta b_{R(A1)^\perp}\| = 1.76.$$

$$A2 \text{ case (2): } \|\delta b_{R(A2)}\| = \|\delta b_{R(A2)^\perp}\| = 11.3076.$$

Perturbations for both problems were computed using a standard procedure in MATLAB.

#### 4.2. Numerical Results

In these numerical experiments we have considered in the implementation of *BIOP* the matrix  $D_m = I_m$ , and the sequence  $\{\sigma_j\} = \{\frac{1}{j+1}\}$ , for  $j > 0$  and parameters  $w_i$ , equal to  $\frac{1}{n+m}$ , for  $i = 1, 2, \dots, n+m$ . *BIOP* was implemented sequentially, with parameters  $\gamma = 10^{-1}$  and  $\beta_k = \frac{1}{k+1}$ .

The stopping condition for the three algorithms was: *if*  $\| \|r^{k+1}\| - \|r^k\| \| < \epsilon \max(\|r^0\|, 1)$ , with  $\epsilon = 10^{-6}$ . The convergence criterion allows to terminate the iterations when decreasing stagnates.

In Table 2 we compare the results obtained with *BIOP* and *BVLS*, implemented in Fortran, in regard to the number of iterations(*Iter*) and the CPU time required by them for each problem  $p$ .  $R_E(p)$  denotes the final value of the norm of the residual obtained by the algorithms for each problem  $p$ , when using  $\epsilon = 10^{-6}$  for the stopping criterion.

We report for *BIOP* the number of iterations(*Iter*) and CPU time required for each problem  $p$ , for reaching the stopping condition using  $\epsilon = 10^{-6}$ . The number of iterations (*Iter*) of *BIOP* is the total of inner iterations required by the process since in each one of the main iterations another inner iterative process is needed. We report for *BVLS* the number of iterations(*Iter*) and CPU time required for each problem  $p$ , for reaching the stopping condition or a norm of the residual less or equal than the one obtained by *BIOP*.

We also give, the corresponding CPU time in seconds.

The dimensions of the test problems are shown using  $m$  for the number of rows,  $n$  for the number of columns.

It is necessary to point out that the solver *BVLS* does not deal with sparsity. Thus, as observed in the previous Table 2, its CPU times are quite significant. Taking this into account it is fair to compare our algorithm with another able to solve sparse problems under the same computational framework.

In Table 3 we compare the number of iterations and the CPU time for *BIOP* and for the *SBL2*, both implemented in MATLAB. We report the results using the same notation of the previous table.

It follows from those results that the *SBL2* is unable to solve several problems. The reason is that matrices  $A1$ ,  $A2$  are rank deficient. On the other hand, *BIOP* solved all test problems, a convenient property of the

p	CPU/Iter		$R_E(p)$	
	BIOP	BVLS	BIOP	BVLS
B1 3452 x 1025	1.31/163	563.23/275	8.44	8.43
B6 896 x 625	0.37/275	213.58/307	3.80	3.80
Well 1033x322	0.25/382	29.2 /152	4362.7	4362.1
Illc 1033x322	0.75/678	19.3 /129	5188.2	5187.6
Well 1850x712	1.95/1728	814.4 /388	4757.8	4757.6
Illc 1850x712	6.7 /2297	214.2/240	5469.0	5468.6
A1(case 2) 144 x 144	0.01 /111	5.9 /186	0.5	0.7
A2 (case 2) 900 x 900	0.39 /145	6741.4 /1120	2.9	2.9
Rand 40 x 20	0.06 /932	0.05 /9	4.76	4.76
Rand 500 x 300	0.06 /9	NC*/900	0.14	NC*

Table 2: BIOP-BVLS

NC\*: Does not converge to a solution into the box.

### Performance profile – CPU time

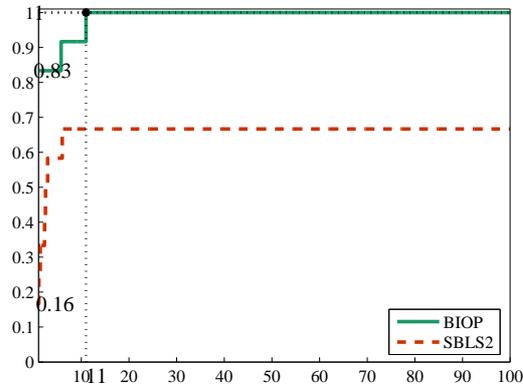


Figure 1: CPU-time: BIOP-SBLS2

projection methods. Moreover, when both algorithms solve a given problem BIOP is faster than SBLS2.

In Figure 1, we analyze the performance data using the performance profiles of Dolan and Moré, as described in [11]. We compare the performance of BIOP and SBLS2 methods in relation to the CPU time required for each problem when the convergence criterion is met. The percentage of the test problems for which a given method is best, is given on the vertical axis of the plot.

It follows from those results that the BIOP method is also able to reduce the residual more rapidly.

#### 4.3. Image reconstruction Problems

In our previous papers we have compared IOP with other algorithms for solving image reconstruction problems. For instance, in [19] we compared IOP with ART (underrelaxed)[5], which is one of the most important class of methods in this field, and CAV [6] using test problems from SNARK [2], showing that is very efficient. Those results showed that IOP requires more CPU time than ART in the sequential implementations, but it obtains better images in some of the problems. In regard to CAV our algorithms are superior both from the points of view of the CPU time and the images' qualities as shown in[19]. In [21] we have also compared a version of IOP called EIOP for dealing with rank deficient problems with KERP (Popa and Zdunek

p	CPU/Iter		$R_E(p)$	
	BIOP	SBLS2	BIOP	SBLS2
B1 3452 x 1025	1.77/163	1.95/6	8.44	8.45
B4 896 x 625	0.34/135	0.78/9	3.82	3.91
B6 896 x 625	0.55/216	0.77/9	3.80	3.91
B7 27376 x 9025	34.1/153	NC*/100	74.3	NC*
Well 1033 x 322	0.44/336	NC* /100	4346.9	NC*
Ilc 1033x322	0.52/406	0.09 /10	5182.0	5155.0
Well 1850 x 712	0.55/225	0.05 /9	4679.6	4749.9
Ilc 1850x712	0.61 /261	NC*/100	5845.2	NC*
A1 (case 2) 144 x 144	0.02 /95	NC* /100	0.70	NC*
A2 (case 2) 900 x 900	0.75 /200	NC* /200	2.90	NC*
Rand 2000 x 750	0.16 /43	0.40 /12	2.28e09	2.28e09
Rand 1200 x 760	0.05 /24	0.30 /13	3.58e07	3.58e07
Rand 1000 x 765	0.03 /20	NC* /100	6.25e07	NC*

Table 3: BIOP-SBLS2

NC\*: Does not converge to a solution into the box.

[16]) and LANDW [14]. The numerical results given there demonstrate that EIOP outperforms KERP and LANDW, both in the rate of convergence and the required CPU time. In the case of image reconstruction problems, EIOP obtains the closest image to the original one in much less CPU time than KERP. In that paper we used problems  $A1$ ,  $A2$  with the perturbations proposed in [16], and it turned out that EIOP gave better results than KERP both in terms of CPU time and distance to the true image.

In all those papers we have considered unconstrained linear least squares problems. Now, in this paper we use BIOP with bounds on the variables, for testing if it is possible to improve in image reconstruction problems the final distance to the true solution when using  $x \geq 0$ , and suitable upper bounds. Hence, in this part we analyze the *BIOP* behavior when solving image reconstruction problems compared with IOP. From the theoretical results we observe that IOP requires  $\gamma \leq 0.5$  for convergence in the case of rank deficient problems [21]. We have used  $\gamma = 0.01$  in the initial iteration, then  $\gamma = 10^{-1}$ .

We show curves representing the distance of the obtained density with regard to the one of the true image, considering the next Definition [6]:

- *Distance*:  $\frac{\|x^k - x^{exact}\|}{\sigma_{exact}\sqrt{N}}$  where

$$\sigma_{exact} = \frac{\|x^{exact} - \rho_{exact}\|}{\sqrt{N}}, \quad \text{and} \quad \rho_{exact} = \frac{\sum_{j=1}^N |x_j^{exact}|}{N},$$

$N$  = number of pixels of each image, and  $x^{exact}$  the density of true image.

In Figure 2 we compare the performances of BIOP and IOP by means of the distance of the reconstructed images for problem  $A1(case2)$  and  $A2(case2)$ , from [16].

In Figure 3 we compare the performances of BIOP and IOP by means of the distance of the reconstructed images for problem  $B1$ ,  $B4$ ,  $B6$  and  $B7$ , from SNARK.

From the point of view of the number of iterations BIOP reaches faster than IOP the image closest to the original for all test problems. BIOP also obtains the minimum distance, and keeps it below the one corresponding to IOP. It reaches distances similar to those obtained with regularization techniques [20, 22].

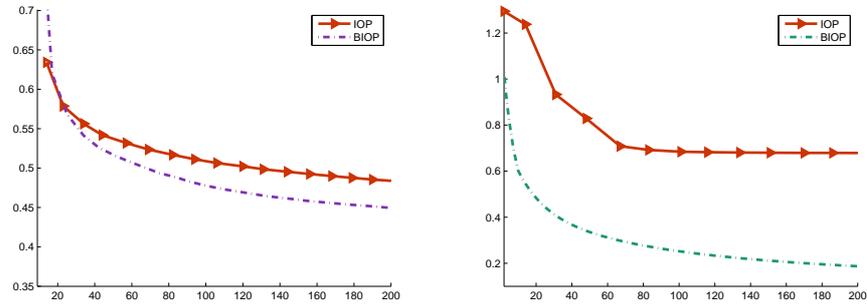


Figure 2: Distance BIOP-IOP:  $A1(\text{case } 2)$ (left)-  $A2(\text{case } 2)$  (right)

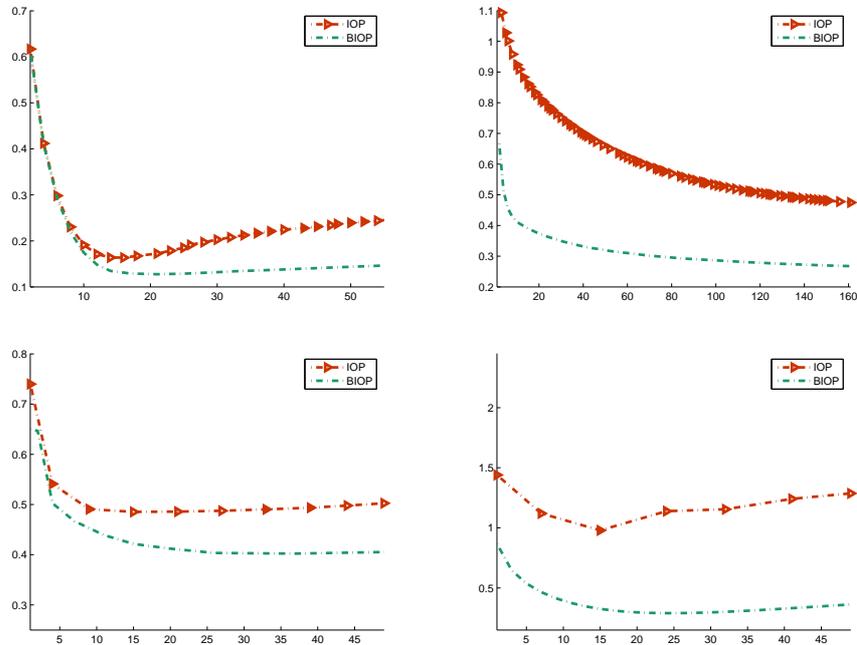


Figure 3: Distance BIOP-IOP:  $B1 - B4(\text{top})$   $B6 - B7(\text{bottom})$

## 5. Conclusions

We have presented in this paper the results obtained when including bound constraints into the IOP algorithm. The need of including these sort of

constraints appear in, for instance, image reconstruction applications where the solution is positive and the density values are bounded. Also in nonlinear optimization problems, like in the SQP method, it is necessary to solve least squares problems in a box.

As it can be seen in the graphs A1-A2 and B1-B7 when  $x \geq 0$  is used, the convergence to the true image is faster, from the point of view of the number of iterations, than in the case of using IOP. In previous papers we had to use regularization techniques in order to obtain results similar to the ones here presented, but at a higher computational cost [20, 22].

As the numerical results show, the BIOP effectiveness is remarkable in several problems as compared to SBLS2 because in particular it does not fail to converge.

## References

- [1] H.H Bauschke, *The Approximation of fixed points of compositions of non-expansive mappings in Hilbert space*, Journal of Math. Analysis and Appl., **202**(1996),150–159.
- [2] J. A. Browne, G. T. Herman, D. Odhner, *SNARK93: A Programming System for Image Reconstruction from Projections*, Department of Radiology, University of Pennsylvania, Medical Image Processing Group, Technical Report MIPG198, 1993.
- [3] C. Byrne, *Iterative oblique projection onto convex sets and the split feasibility problem*, Inverse Problems, **18** (2002), 441–453.
- [4] C. Byrne, and Y. Censor, *Proximity Function Minimization Using Multiple Bregman Projections with Applications to split feasibility and Kullback-Leibler distance minimization*, Annals of Operations Research, **105** (2001), 77–98.
- [5] Y. Censor and S. Zenios, *Parallel Optimization: Theory and Applications*, Oxford University Press, New York, 1997.
- [6] Y. Censor, D. Gordon and R. Gordon, *Component Averaging: An efficient iterative parallel algorithm for large and sparse unstructured problems*, Parallel Computing, **27** (2001), 777–808.

- [7] Y. Censor and T. Elfving, *Block-iterative algorithms with diagonally scaled oblique projections for the linear feasibility problem*, SIAM J. Matrix Anal. and Applicat., **24** (2002), 40–58.
- [8] Y. Censor, *Computational acceleration of Projections algorithms for linear best approximation problem*, Linear Algebra and its Applications, **416** (2006), 111–123.
- [9] P. L. Combettes, *Construction d'un point fixe commun a famille de contractions fermes*, C.R. Acad. Sci. Paris, Ser. I Math. 320 (1995).
- [10] I. Csiszár and G. Tusnády, *Information geometry and alternating minimization procedures*, Statistics and Decisions, Supplement, **1** (1984), 205–237.
- [11] E.D. Dolan and J.J. Moré, *Benchmarking optimization software with performance profiles*, Mathematical Programming, **91** (2002), 201–213.
- [12] U. M. García Palomares, *Parallel projected aggregation methods for solving the convex feasibility problem*, SIAM J. Optim., **3** (1993), 882–900.
- [13] M. Jiang and G. Wang, *Convergence studies on iterative Algorithms for Image Reconstruction*, IEEE Transactions on Medical Imaging, **22** (2003), 569–579.
- [14] L. Landweber, *An iteration formula for Fredholm integral equations of the first kind*, Amer. J. Math., **73** (1951), 615–624.
- [15] J.M. Ortega and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, New York and London, 1970.
- [16] C. Popa, R. Zdunek, *Kaczmarz extended algorithm for tomographic image reconstruction from limited-data*, Mathematics and Computers in Simulation, **65** (2004), 579–598.
- [17] L.I. Portugal, J. Judice and L.N. Vicente, *A comparison of block pivoting and interior-point algorithms for linear least squares problem with nonnegative variables*, Math. Comp., **63** (1994), 625–643.
- [18] H.D. Scolnik, N. Echebest, M.T. Guardarucci, M. C. Vacchino, *A class of optimized row projection methods for solving large non-symmetric linear systems*, Applied Numerical Mathematics, **41** (2002), 499–513.

- [19] H. D. Scolnik, N. Echebest, M. T. Guardarucci, M. C. Vacchino, *Incomplete Oblique Projections for Solving Large Inconsistent Linear Systems*, Mathematical Programming B, **111** (2008a), 273–300.
- [20] H.D. Scolnik, N. Echebest, M.T. Guardarucci, *Regularized Incomplete Oblique Projections Method for Solving Least-Squares Problems in Image Reconstruction*, International Transactions in Operational Research, **15** (2008b), 417–438.
- [21] H.D. Scolnik, N. Echebest, M. T. Guardarucci, *Extensions of Incomplete Oblique Projections for Rank-deficient problems*, Journal of Industrial and Management Optimization (JIMO), **5** (2009a), 175–191.
- [22] H.D. Scolnik, N. Echebest, M.T. Guardarucci, *Implicit Regularization of the Incomplete Oblique Projections Methods*, International Transactions in Operational Research, **16** (2009b), 525–546.
- [23] P.B. Stark and R.L. Parker, *Bounded Variable Least Squares: An Algorithm and Applications*, Journal Computational Statistics, **10** (1995), 129–141.
- [24] Y. Xiao, D. Michalski, Y. Censor and J.M. Galvin, *Inherent smoothness of intensity patterns for intensity radiation therapy generated by simultaneous projection algorithms*, Physics in Medicine and Biology, **49** (2004), 3227–3245.