FRIEDRICH-ALEXANDER-UNIVERSITÄT ERLANGEN-NÜRNBERG
INSTITUT FÜR INFORMATIK (MATHEMATISCHE MASCHINEN UND DATENVERARBEITUNG)

Lehrstuhl für Informatik 10 (Systemsimulation)

Projection algorithms with correction

Aurelian Nicola, Constantin Popa, Ulrich Rüde

Lehrstuhlbericht 09 - 15

Projection algorithms with correction

Aurelian Nicola[*], Constantin Popa[†], Ulrich Rüde[‡]

Abstract

We present in this paper two versions of a general correction procedure applied to a classical iterative method, which gives us the possibility, under certain assumptions, to obtain an extension of it to inconsistent linear least-squares problems. We prove that some well known extended projection type algorithms from image reconstruction in computerized tomography fit into one or the other of these general versions and are derived as particular cases of them. We also present some numerical experiments on two phantoms widely used in image reconstruction literature. The experiments show the importance of this correction procedure, reflected in the quality of reconstructed images.

2000 MS Classification: 65F10, 65F20

Key words and phrases: projection algorithms, correction procedure, inconsistent least-squares problems, Kaczmarz Extended, Cimmino Extended, KECG.

1 Introduction

Many classes of "real world problems" give rise, after appropriate discretizations to large, sparse and ill-conditioned linear systems of equations of the form $Ax = b$, where the $m \times n$ matrix A contains information concerning the problem, whereas $b \in I\!R^m$ represents measured "effects" produced by the unknown "cause" $x \in I\!R^n$. But, due to inevitable measurements errors, the "effect" b may go out of the "range of action" of the problem information matrix A, such that the above system becomes inconsistent and must be reformulated in the least squares sense: find $x \in I\!R^n$ such that

$$\| Ax - b \| = \min\{\| Az - b \|, z \in I\!R^n\}, \tag{1}$$

where $\| \cdot \|$ denotes the Euclidean norm ($\langle \cdot, \cdot \rangle$ will be the Euclidean scalar product)

Remark 1 *Concerning the matrix involved in (1) we shall suppose for the whole paper that it has nonzero rows A_i and columns A^j, i.e.*

$$A_i \neq 0, i = 1, \dots, m, \; A^j \neq 0, j = 1, \dots, n. \tag{2}$$

These assumptions are not essential restrictions of the generality of the problem (1) because, if A has null rows and/or columns, it can be easily proved that they can be eliminated without affecting its set of classical ($S(A; b)$, in the consistent case) or least squares ($LSS(A; b)$, in the inconsistent case) solutions.

We shall first introduce some notations. The spectrum and spectral radius of a square matrix will be denoted by $\sigma(B)$ and $\rho(B)$, respectively. By A^T , $\mathcal{N}(A), \mathcal{R}(A)$ we shall denote the transpose, null space and range of A. $P_S(x)$ will be the orthogonal (Euclidean) projection onto a vector subspace S of some $I\!R^q$. $S(A; b), LSS(A; b), x_{LS}$ will stand for the set of classical and least squares solution

[*] "Ovidius" University of Constanta, Faculty of Mathematics and Computer Science, Romania; e-mail: anicola@univ-ovidius.ro.

[†] "Ovidius" University of Constanta, Faculty of Mathematics and Computer Science, Romania; e-mail: cpoopa@univ-ovidius.ro.; for this author the paper was supported by a DAAD Grant in the period 01.07 - 31.08.2009 at Institute for Computer Science 10 (Systemsimulation), Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

[‡] Institute for Computer Science 10 (Systemsimulation), Friedrich-Alexander-Universität Erlangen-Nürnberg, Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany; e-mail: ulrich.ruede@informatik.uni-erlangen.de

of (1), respectively, and the (unique) minimal norm one (in both cases). In the consistent case for (1) we have $S(A;b) = LSS(A;b)$. In the general case the following properties are known (see e.g. [1])

$$x_{LS} \perp \mathcal{N}(A),\ b = b_A + b_A^*,\ b_A = P_{\mathcal{R}(A)}(b),\ b_A^* = P_{\mathcal{N}(A^T)}(b), \tag{3}$$

$$LSS(A;b) = x_{LS} + \mathcal{N}(A) \text{ and } x \in LSS(A;b) \ \Leftrightarrow\ Ax = P_{\mathcal{R}(A)}(b), \tag{4}$$

$$S(A;b) = x_{LS} + \mathcal{N}(A) \text{ and } x \in S(A;b) \ \Leftrightarrow\ Ax = b. \tag{5}$$

The spectral norm of A will be defined by

$$\| A \| = \sup_{x \neq 0} \frac{\| Ax \|}{\| x \|} = \sup_{\|x\|=1} \| Ax \|. \tag{6}$$

Remark 2 *The minimal norm solution x_{LS} is the unique element of $LSS(A;b)$ (or $S(A;b)$) which belongs to $\mathcal{R}(A^T)$.*

Algorithm General Projections (GP). *Initialization:* $x^0 \in I\!R^n$
Iterative step:

$$x^{k+1} = Qx^k + Rb, \tag{7}$$

where $Q : n \times n$ and $R : n \times m$ are real matrices. Now, we introduce the following basic assumptions on the above considered matrices Q and R.

$$I - Q = RA. \tag{8}$$

$$\text{if } \tilde{Q} = QP_{\mathcal{R}(A^T)}, \text{ then } \ \| \tilde{Q} \| < \ 1. \tag{9}$$

As examples of such kind of algorithms we can give Kaczmarz, Cimmino (see also section 3) and many other "row-action" methods from image reconstruction from projections (see e.g. [2, 10]). Unfortunately, it is well know that the sequence $(x^k)_{k \geq 0}$ generated with these algorithms converges to an element of $S(A;b)$ only in the consistent case of (1), whereas in the inconsistent one, although it still converges its limit is a solution of a weighted formulation of (1) ([5, 3]) or lies at a certain distance from $LSS(A;b)$ ([13]). In order to overcome this difficulty, in the next section of the paper we propose two version with correction of the above General Projections algorithm (7). Under additional assumptions on the correction vector and the matrices Q, R from (7) we prove that the new algorithms generate sequences of approximations that always converge to an arbitrary element of $LSS(A;b)$ or x_{LS}. In section 3 we present well known projection algorithms that fit into this general construction: Kaczmarz Extended, KECG, Cimmino Extended and a general extension procedure proposed in [6]. Section 4 is devoted to some numerical experiments on two phantoms widely used in the image reconstruction literature.

2 The GP algorithm with correction

The version with correction of the algorithm (7) is the following.
Algorithm General Projections with Correction. *Initialization:* $x^0 \in I\!R^n$
Iterative step:

$$x^{k+1} = Qx^k + Rb + v^k. \tag{10}$$

In the two following subsections of the paper, we shall introduce two sets of additional assumptions on the elements Q, R and v^k of the iteration (10) such that it will generate a sequence $(x^k)_{k \geq 0}$ convergent to a least squares solution of (1).

2.1 First set of additional assumptions

Beside (8) and (9) we shall request the following additional assumptions:

$$\text{if } x \in \mathcal{N}(A) \text{ then } Qx = x \in \mathcal{N}(A), \tag{11}$$

$$\forall y \in I\!R^m, \;\; Ry \in \mathcal{R}(A^T), \tag{12}$$

$$v^k \in \mathcal{R}(A^T), \tag{13}$$

and for ε^k defined by

$$\varepsilon^k = v^k + Rb_A^*, \tag{14}$$

we suppose that there exist constants $c > 0$ and $\delta \in [0, 1)$ such that

$$\| \varepsilon^k \| \leq c\delta^k, \;\; \forall k \geq 0. \tag{15}$$

Lemma 1 *In the above hypothesis, if $(x^k)_{k\geq 0}$ is the sequence generated by the algorithm (10) then*

$$P_{\mathcal{N}(A)}(x^k) = P_{\mathcal{N}(A)}(x^0), \forall k \geq 0. \tag{16}$$

Proof. We use mathematical induction. For $k = 0$ the property is true and let k be an arbitrary fixed nonnegative integer such that (16) holds for it. Let's first remark that from (11) and (9) we get

$$Q = Q(P_{\mathcal{N}(A)} \oplus P_{\mathcal{R}(A^T)}) = QP_{\mathcal{N}(A)} + \tilde{Q} = P_{\mathcal{N}(A)} + \tilde{Q}. \tag{17}$$

The, form (10), (17) and the induction hypothesis we successively obtain

$$x^{k+1} = \tilde{Q}x^k + Rb + v^k + P_{\mathcal{N}(A)}(x^0), \tag{18}$$

in which the first three terms belong to $\mathcal{R}(A^T)$ because of (9), (12) and (13), respectively. From (18) we get $P_{\mathcal{N}(A)}(x^{k+1}) = P_{\mathcal{N}(A)}(x^k) = P_{\mathcal{N}(A)}(x^0)$ and the proof is complete. ♢

According to (16) we can now define the error vector of the iteration (10) as

$$e^k = x^k - (P_{\mathcal{N}(A)}(x^0) + x_{LS}). \tag{19}$$

Theorem 1 *In the above hypothesis, the sequence $(x^k)_{k\geq 0}$ generated by the algorithm (10) converges and*

$$\lim_{k\to\infty} x^k = P_{\mathcal{N}(A)}(x^0) + x_{LS} \in LSS(A; b). \tag{20}$$

Proof. from (9) it results that $(I - \tilde{Q})$ is invertible and (see e.g. [1])

$$(I - \tilde{Q})^{-1} = \sum_{ju\geq 0} \tilde{Q}^j. \tag{21}$$

By using the property of x_{LS} from Remark 2 and (9) we get

$$Qx_{LS} = \tilde{Q}x_{LS}, \tag{22}$$

which together with (8) and (3) gives us

$$(I - \tilde{Q})x_{LS} = x_{LS} - QP_{\mathcal{R}(A^T)}(x_{LS}) = x_{LS} - Qx_{LS} = (I - Q)x_{LS} = RAx_{LS} = Rb_A,$$

thus

$$x_{LS} = (I - \tilde{Q})^{-1}Rb_A \text{ and } (I - Q)x_{LS} = Rb_A. \tag{23}$$

Then, from (19), (10), (17), (16), (23), (22), $\tilde{Q}P_{\mathcal{N}(A)} = 0$ and (14) we successively obtain

$$e^{k+1} = Qx^k + Rb + v^k - (P_{\mathcal{N}(A)}(x^0) + x_{LS}) = \tilde{Q}x^k + Rb + v^k - x_{LS} =$$

$$\tilde{Q}x^k + Rb + v^k - (\tilde{Q}x_{LS} + Rb_A) = \tilde{Q}x^k + Rb_A^* - \tilde{Q}P_{\mathcal{N}(A)}(x^0) - \tilde{Q}x_{LS} + v^k = \tilde{Q}e^k + \varepsilon^k, \forall k \geq 0. \quad (24)$$

A recursive application of formula (24) gives

$$e^k = \tilde{Q}^k e^0 + \sum_{j=0}^{k-1} \tilde{Q}^j \varepsilon^{k-1-j}, \ \forall k \geq 1. \quad (25)$$

But, according to (15) and (9) we obtain $\forall k \geq 1$

$$\| \sum_{j=0}^{k-1} \tilde{Q}^j \varepsilon^{k-1-j} \| \leq \sum_{j=0}^{k-1} \| \tilde{Q} \|^j \| \varepsilon^{k-1-j} \| \leq c \left[\sum_{j=0}^{k-1} \| \tilde{Q} \|^j \delta^{k-1-j} \right] \leq \sum_{j=0}^{k-1} \gamma^j \gamma^{k-1-j} = k\gamma^{k-1}, \quad (26)$$

with

$$\gamma = \max\{\| \tilde{Q} \|, \delta\} \in [0, 1), \quad (27)$$

hence

$$\lim_{k\to\infty} \sum_{j=0}^{k-1} \tilde{Q}^j \varepsilon^{k-1-j} = 0. \quad (28)$$

From (25) and (28) we then get

$$\lim_{k\to\infty} e^k = \lim_{k\to\infty} \left(\tilde{Q}^k e^0 + \sum_{j=0}^{k-1} \tilde{Q}^j \varepsilon^{k-1-j} \right) = 0,$$

which together with the definition of e^k in (19) gives us (20) and completes the proof. ♢

2.2 Second set of additional assumptions

Beside (8) and (9) we shall request the following additional assumptions:

$$x^k \in \mathcal{R}(A^T), \ \forall k \geq 0 \quad (29)$$

and for ε^k defined by (14), the property (15).

Theorem 2 *In the above hypothesis, the sequence $(x^k)_{k\geq 0}$ generated by the algorithm (10) converges and*

$$\lim_{k\to\infty} x^k = x_{LS}. \quad (30)$$

Proof. We first remark that the properties (22) and (23) still hold in this new situation. According to the assumption (29), we shall define in this case the error vector for the algorithm (10) by

$$e^k = x^k - x_{LS}. \quad (31)$$

Then, by using (31), (10), (29), (22), (23), (3) and (14) we obtain

$$e^{k+1} = x^{k+1} - x_{LS} = \tilde{Q}x^k + Rb + v^k - x_{LS} = \tilde{Q}x^k + Rb + v^k - (\tilde{Q}x_{LS} + Rb_A) = \tilde{Q}(x^k - x_{LS})v^k + Rb_A^* = \tilde{Q}e^k + \varepsilon^k. \quad (32)$$

We now follow exactly the same steps as in the proof Theorem 1 and get (30). ♢

Remark 3 *The number of assumptions in the second set is smaller that in the first set. This is explained by the property (29) which, on a side contains behind it many important properties of the matrices Q, R and the correction v^k, and on the other side because of the particular result obtained in (30) (i.e. we approximate only the minimal norm solution, not every least squares solution as in (20)).*

3 Some particular cases

3.1 Applications for the first set of additional assumptions

Kaczmarz-like algorithms Extended

As a first example we consider Kaczmarz's successive projection method from [8] (see also [15]).
Algorithm Kaczmarz (K). *Initialization:* $x^0 \in I\!R^n$
Iterative step:

$$x^{k+1} = P_{H_1} \circ \ldots \circ P_{H_m}(x^k), \tag{33}$$

where

$$P_{H_i}(x) = P_i(x) + \frac{b_i}{\parallel A_i \parallel^2} A_i, \;\; P_i(x) = x - \frac{\langle x, A_i \rangle}{\parallel A_i \parallel^2} A_i. \tag{34}$$

Algorithm K can be written in the form (7) with Q given by

$$Q = P_1 \circ \ldots \circ P_m \tag{35}$$

and R the $n \times m$ matrix of which i-th column is

$$(R)^i = \frac{1}{\parallel A_i \parallel^2} P_1 \circ \ldots \circ P_{i-1}(A_i), \;\; i = 1, \ldots, m \tag{36}$$

where by P_0 we denoted the unit matrix (for $i = 1$). The properties (8)-(9) and (11)-(12) are proved in [15]. If we define the applications

$$\Phi = \varphi_1 \circ \ldots \circ \varphi_n, \quad \varphi_j(y) = y - \frac{\langle y, A^j \rangle}{\parallel A^j \parallel^2} A^j, \tag{37}$$

we can define the Kaczmarz Extended algorithm, proposed in [12]. .
Algorithm Kaczmarz Extended (KE).
Initialization: $x^0 \in I\!R^n, y^0 = b$
Iterative step:

$$y^{k+1} = \Phi y^k, \tag{38}$$
$$b^{k+1} = b - y^{k+1}, \tag{39}$$
$$x^{k+1} = Q x^k + R b^{k+1}. \tag{40}$$

Remark 4 *In view of (37), the step (38) is a Kaczmarz like iteration (33)-(34) applied to the consistent system*

$$A^T y = 0. \tag{41}$$

Moreover (see e.g. [15, 12]), for $y^0 = b$ we find

$$\lim_{k \to \infty} y^k = b^*_A. \tag{42}$$

By analogy with the properties in (9) and (17) we obtain

$$\text{if } \tilde{\Phi} = \Phi P_{\mathcal{R}(A)} \text{ then } \Phi = P_{\mathcal{N}(A^T)} \oplus \tilde{\Phi}, \tilde{\Phi} P_{\mathcal{N}(A^T)} = P_{\mathcal{N}(A^T)} \tilde{\Phi} = 0 \text{ and } \parallel \tilde{\Phi} \parallel < 1. \tag{43}$$

Proposition 1 *The KE algorithm can be written in the form of (10) with the correction v^k given by*

$$v^k = -R \Phi^{k+1} b, \tag{44}$$

such that the assumptions (13) and (15) hold.

Proof. From (38) we obtain $y^{k+1} = \Phi^{k+1} y^0 = \Phi^{k+1} b$, which yields, by also using (39)

$$x^{k+1} = Qx^k + Rb^{k+1} = Qx^k + Rb + (-R\Phi^{k+1} b),$$

i.e. (10)) with v^k from (44). Assumption (13) then results directly from (44) and (12). According to (15), from (44) and (43) we obtain

$$v^k = -R\Phi^{k+1} b = -R\left(\tilde{\Phi}^{k+1} b + b_A^*\right) = -R\tilde{\Phi}^{k+1} b - Rb_A^*. \quad (45)$$

Then from (45) and again (43) (last inequality) we get

$$\| \varepsilon^k \| \leq \| v^k + Rb_A^* \| = \| R\tilde{\Phi}^{k+1} b \| \leq \| R \| \| b \| \| \tilde{\Phi} \|^{k+1}, \quad (46)$$

i.e. (15) with $c = \| R \| \| b \| \| \tilde{\Phi} \|$ and $\delta = \| \tilde{\Phi} \| \in [0, 1)$. ♢

We now consider Kaczmarz algorithm with relaxation parameter (ω-Kaczmarz, for short).
Algorithm ω-Kaczmarz. *Initialization:* $\omega > 0, x^0 \in I\!R^n$
Iterative step:

$$x^{k+1} = P_{H_1}^{\omega} \circ \ldots \circ P_{H_m}^{\omega}(x^k), \quad (47)$$

where

$$P_{H_i}^{\omega}(x) = (1 - \omega)x + \omega P_{H_i}(x) \quad (48)$$

and $P_{H_i}(b; x)$ from (34). It can be written in the form (7) as

$$x^{k+1} = Q^{\omega} x^k + R^{\omega} b \quad (49)$$

with

$$Q^{\omega} = P_1^{\omega} P_2^{\omega} \ldots P_m^{\omega}, \ R^{\omega} = \omega col\left(\frac{1}{\| A_1 \|^2} Q_0^{\omega} A_1, \ldots, \frac{1}{\| A_m \|^2} Q_{m-1}^{\omega} A_m\right), \quad (50)$$

where

$$P_i^{\omega} = (1 - \omega)I + \omega P_i, \ Q_0^{\omega} = I, \ Q_i^{\omega} = P_1^{\omega} \ldots P_i^{\omega}, \ i = 1, \ldots, m, \quad (51)$$

with P_i from (34). In [10] it is proved that, if

$$\omega \in (0, 2) \quad (52)$$

then the matrices Q^{ω}, R^{ω} from (49) satisfy (8)-(9) and (11)-(12). Its extended version, proposed in [12] is Kaczmarz Extended with Relaxation Parameters (KERP, for short).
Algorithm KERP.
Initialization: $\omega, \alpha \in (0, 2)$, $x^0 \in I\!R^n, y^0 = b$
Iterative step:

$$\begin{aligned} y^{k+1} &= \Phi^{\alpha} y^k, && (53) \\ b^{k+1} &= b - y^{k+1}, && (54) \\ x^{k+1} &= Qx^k + Rb^{k+1}, && (55) \end{aligned}$$

where

$$\Phi^{\alpha} = \varphi_1^{\alpha} \circ \ldots \circ \varphi_n^{\alpha}, \quad \varphi_j^{\alpha}(y) = (1 - \alpha)y + \alpha\varphi_j(y), \quad (56)$$

with $\varphi_j(y)$ from (37). The algorithm KERP can be written in the form (10) as

$$x^{k+1} = Q^{\omega} x^k + R^{\omega} b + v^k, \quad (57)$$

with

$$v^k = -R^{\omega}(\Phi^{\alpha})^{k+1} b. \quad (58)$$

A similar result as in Proposition 1 can then be obtained with the constants c and δ defined by

$$c = \| R^{\omega} \| \| b \| \| \Phi^{\alpha} \|, \quad \delta = \| \Phi^{\alpha} \| \in [0, 1). \quad (59)$$

Algorithm KECG

The classical CG algorithm applied to the normal equation of (1), $A^TAx = A^Tb$ gives us the CG for Least Squares method (CGLS) from below (see [1, 9]).

Algorithm CGLS. *Initialization:* $x^0 \in I\!R^n, r^0 = b - Ax^0, p^0 = A^T r^0$

Iterative step:

$$\begin{aligned} \alpha_k &= \| A^T r^k \|^2 \,/\, \| Ap^k \|^2, \\ x^{k+1} &= x^k + \alpha_k p^k, \\ r^{k+1} &= b - Ax^{k+1} = r^k - \alpha_k A p^k, \\ \beta_k &= \| A^T r_{k+1} \|^2 \,/\, \| A^T r^k \|^2, \\ p^{k+1} &= A^T r^{k+1} + \beta_k p^k. \end{aligned} \tag{60}$$

Theorem 3 ([9]) *Let $x^0 \in I\!R^n$ be an arbitrary vector, $r = \mathrm{rank}(A)$, $U^TAV = \mathrm{diag}(\sigma_1, \dots, \sigma_r, 0, \dots, 0)$ a singular value decomposition of A, with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$ and the numbers δ and c defined by*

$$\mu = \frac{\sigma_1 - \sigma_r}{\sigma_1 + \sigma_r} \in [0, 1), \quad \bar{c} = \frac{\| Ax^0 - P_{R(A)}(b) \|}{\sigma_r}. \tag{61}$$

Then, the sequence $(x^k)_{k\geq 0}$ generated with the CGLS algorithm converges to $P_{\mathcal{N}(A)}(x^0) + x_{LS}$ and we have the estimate

$$\| x^k - (P_{\mathcal{N}(A)}(x^0) + x_{LS}) \| \leq \ \bar{c} \cdot \delta^k, \ \forall k \geq 1. \tag{62}$$

According to the above Theorem 3, if we apply the CGLS algorithm to the (consistent) system (41) with $y^0 = b$, we obtain a sequence $(y^k)_{k\geq 0}$ convergent to b^*_A. Moreover, from (62) we get

$$\| y^k - b^*_A \| \leq \ \alpha \cdot \mu^k, \ \forall k \geq 1, \tag{63}$$

with μ from (61) and α given by

$$\gamma = \frac{\| A^T b \|}{\sigma_r}. \tag{64}$$

We shall denote the k-th iteration of this algorithm by $y^k = CGLS(A^T; y^{k-1})$.
In [14] we proposed the following mixed Kaczmarz - Conjugate Gradient algorithm.

Algorithm KECG. *Initialization:* $x^0 \in I\!R^n, y^0 = b$

Iterative step:

$$y^{k+1} = CGLS(A^T; y^k), \tag{65}$$

$$b^{k+1} = b - y^{k+1}, \tag{66}$$

$$x^{k+1} = Qx^k + Rb^{k+1}, \tag{67}$$

with Q and R from (35)-(36). Let us denote, in a matrix form by

$$y^{k+1} = \Pi(y^k) \tag{68}$$

the step (65) of the KECG algorithm.

Proposition 2 *The KECG algorithm can be written as (10) with the correction v^k given by*

$$v^k = Ry^{k+1} = -R\Pi^{k+1}b, \tag{69}$$

such that the assumptions (13) and (15) hold.

Proof. From (67) and (68) we obtain

$$x^{k+1} = Qx^k + Rb^{k+1} = Qx^k + Rb + (-Ry^{k+1}),$$

i.e. (10) with v^k from (69). Assumption (13) then results directly from (69) and (12). Then, from (19) and (69)) we get

$$\| \varepsilon^k \| \leq \| v^k + Rb^*_A \| = \| -Ry^{k+1} + Rb^*_A \| \leq \| R \| \| y^{k+1} - b^*_A \| \leq \| R \| \, \alpha \mu^{k+1}, \tag{70}$$

i.e. (15) with $c = \parallel R \parallel \alpha\mu$, with $\delta = \mu$ (μ from (61)). $\diamondsuit$

Algorithm Cimmino Extended

Our example in this paragraph refers to the simultaneous reflections algorithm proposed by its author in [5]. Cimmino considers a consistent linear least squares problem of the form (1). A (classical) solution of it will lie in the intersection of the m hyperplanes defined by the equations of the system

$$H_i := \{x, \langle x, A_i \rangle = b_i\}, \quad i = 1, \dots, m. \tag{71}$$

Given a current approximation x^k, the next one x^{k+1} is constructed via

$$x^{k+1} = \sum_{i=1}^{m} \frac{\omega_i}{\omega} S_{H_i}(x) \tag{72}$$

where $S_{H_i}(x)$ are the reflections of x^k with respect to the hyperplane (71), defined by

$$S_{H_i}(x^k) = x^k + 2\frac{b_i - \langle x, A_i \rangle}{\|A_i\|^2} A_i \text{ and } \omega_i > 0, \quad \omega = \sum_{i=1}^{m} \omega_i. \tag{73}$$

From (72) and (73) we derive for Q, R in (7) the following expressions

$$Q = \sum_{i=1}^{m} \frac{\omega_i}{\omega} S_i, \quad S_i := I - 2\frac{A_i A_i^T}{\|A_i\|^2}, \quad R = \sum_{i=1}^{m} \frac{\omega_i}{\omega} \frac{b_i}{\|A_i\|^2} A_i. \tag{74}$$

Then, Cimmino's algorithm (72) can be written as follows.
Algorithm Cimmino (Cmm). *Initialization:* $\omega_i > 0, i = 1, \dots, m; x^0 \in I\!R^n$
Iterative step:

$$x^{k+1} = Qx^k + Rb. \tag{75}$$

The properties (8)-(9) and (11)-(12) were proved in [11]. Now, if we define the matrix T by

$$T = \sum_{i=1}^{n} \frac{\alpha_j}{\alpha} F_j, \text{ with } F_j = I - 2\frac{A^j {A^j}^T}{\|A^j\|^2}, \text{ and } \alpha = \sum_{j=1}^{n} \alpha_j, \tag{76}$$

where $\alpha_j > 0$ are arbitrary weights we can consider the Extended Cimmino algorithm.
Algorithm Cimmino Extended (CmmE).
Initialization: $\omega_i > 0, i = 1, \dots, m;\ \alpha_j > 0, j = 1, \dots, n,\ x^0 \in I\!R^n, y^0 = b$
Iterative step:

$$y^{k+1} = Ty^k, \tag{77}$$

$$b^{k+1} = b - y^{k+1}, \tag{78}$$

$$x^{k+1} = Qx^k + Rb^{k+1}. \tag{79}$$

Moreover, as for Kaczmarz Extended we get for $\tilde{T} = TP_{\mathcal{R}(A)}$

$$\parallel \tilde{T} \parallel < 1. \tag{80}$$

The algorithm (77)-(79) can be written in the form (10), with the correction v^k given by

$$v^k = -RT^{k+1}b. \tag{81}$$

Then, the theoretical considerations from Section 3.1 directly apply and we get a result similar to that in Proposition 1 with c and δ given by

$$c = \parallel R \parallel \parallel b \parallel \parallel \tilde{T} \parallel, \quad \delta = \parallel \tilde{T} \parallel \in [0, 1). \tag{82}$$

3.2 Applications for the second set of additional assumptions

In the paper [6] the author considers the following two steps algorithm.
Algorithm Elfving (ELF).
Initialization:

$$x^0 \in \mathcal{R}(A^T), \quad y^0 = b - Az^0, \text{ for some } z^0 \in I\!R^n. \tag{83}$$

Iterative step:

$$y^{k+1} = (I - A\Gamma)y^k, \tag{84}$$

$$x^{k+1} = Qx^k + R(b - y^{k+1}), \tag{85}$$

where the matrices $Q : n \times n$, $R : n \times m$, $\Gamma : m \times n$ satisfy

$$Q + RA = I, \tag{86}$$

and

$$\text{for } w \in \mathcal{R}(A), z = Qz + Rw \text{ if and only if } Az = w, \tag{87}$$

$$\alpha = \| (I - A\Gamma)P_{\mathcal{R}(A)} \| < 1, \tag{88}$$

$$\beta = \| QP_{\mathcal{R}(A^T)}P_{\mathcal{R}(A)} \| < 1, \tag{89}$$

$$z \in \mathcal{N}(A^T) \implies \Gamma z = 0, \tag{90}$$

$$u \in \mathcal{R}(A) \implies Ru \in \mathcal{R}(A^T). \tag{91}$$

Remark 5 *In the original paper [6] the assumptions (88) and (89) are weaker, by using the spectral radii of the corresponding matrices instead of the spectral norms. We have chose this version because it fits into the considerations of section 2.2.*

From (84) and (85) we get for x^{k+1} the expression in (10), with the correction v^k given by

$$v^k = -Ry^{k+1}, \tag{92}$$

with R from (85) and y^{k+1} from (84).

Proposition 3 *In the above hypothesis, the matrices Q, R the approximations x^k and the corrections ε^k (defined as in (14), with v^k from (92), satisfy the assumptions (8), (9), (29) and (15), from section 2.2.*

Proof. **Assumption (8)**. This is used in the same form in [6] - see (86).
Assumption (9). Is exactly (89).
Assumption (29). We shall use the mathematical induction. The property holds for x^0 from (83). Let $k \geq 0$ be an arbitrary fixed number such that (29) holds for x^k. We shall prove it for x^{k+1}. As in [6], let $y^0_A = P_{\mathcal{R}(A)}(y^0)$, $y^{0,*}_A = P_{\mathcal{N}(A^T)}(y^0)$. From (83) and (3) we get $y^0 = (b_A - Az^0) + b^*_A$, and from all the above equalities

$$y^{0,*}_A = b^*_A. \tag{93}$$

Then, (84), (93), (90) give us

$$y^{k+1} = \left[(I - A\Gamma)P_{\mathcal{R}(A)}\right]^{k+1}(y^0_A) + b^*_A. \tag{94}$$

From (10), (92), (85) and (94) it results

$$x^{k+1} = Qx^k + R(b - y^{k+1}) = (x^k - RAx^k) + Rb_A - R\left[(I - A\Gamma)P_{\mathcal{R}(A)}\right]^{k+1}(y^0_A). \tag{95}$$

But, from (91) and the induction hypothesis we get

$$x^k - RAx^k \in \mathcal{R}(A^T), \ Rb_A \in \mathcal{R}(A^T), \tag{96}$$

i.e. by using again (91)

$$R\left[(I - A\Gamma)P_{\mathcal{R}(A)}\right]^{k+1}(y^0_A) \in \mathcal{R}(A^T). \tag{97}$$

Then, from (95) - (97) we get $x^{k+1} \in \mathcal{R}(A^T)$, i.e by induction process the assumption (29).
Assumption (15). From (14), (92), (94) we get

$$\| \varepsilon^k \| = \| v^k + Rb_A^* \| = \| -R \left[(I - A\Gamma) P_{\mathcal{R}(A)} \right]^{k+1} (y_A^0) \| \leq$$

$$\| R \| \| (I - A\Gamma) P_{\mathcal{R}(A)} \| \| (y_A^0) \| \cdot \| (I - A\Gamma) P_{\mathcal{R}(A)} \|^k,$$

i.e. (15) with (see (88))

$$c = \| R \| \| (I - A\Gamma) P_{\mathcal{R}(A)} \| \| (y_A^0) \|, \quad \delta = \| (I - A\Gamma) P_{\mathcal{R}(A)} \| \in [0, 1). \tag{98}$$

$\diamondsuit$

It then results that the extension procedure from [6], with the modification in (88)-(89) is a particular case of the general extension procedure in section 2.2.

4 Numerical experiments

In our experiments we used the head and mitochondrion phantoms from the paper [3] (63×63 pixels resolution for each of them, but with the scanning matrices with 1376, respectively 1378 rays - i.e. the number of rows in A). For each phantom we had a consistent and an inconsistent right hand side b for our reconstruction problem. The main idea of the experiments that follow was to use a method from each principal classes of ART algorithms: successive (Kaczmarz) and simultaneous (Cimmino). We have used in our experiments the following measures for the approximation errors (see also [7]).

- x^{ex} = head or mitochondrion phantom ; $N = 63^2 = 3969$
- $x^{ex} = (x_1^{ex}, \dots, x_N^{ex})^T; x^k = (x_1^k, \dots, x_N^k)^T$; $\bar{x}^{ex} = \frac{\sum_{i=1}^N x_i^{ex}}{N}$; $\bar{x}^k = \frac{\sum_{i=1}^N x_i^k}{N}$
- **Distance** = $\sqrt{\frac{\sum_{i=1}^N (x_i^{ex} - x_i^k)^2}{\sum_{i=1}^N (x_i^{ex} - \bar{x}^{ex})^2}}$
- **Relative error** = $\frac{\sum_{i=1}^N |x_i^{ex} - x_i^k|}{\sum_{i=1}^N x_i^{ex}}$
- **Standard deviation** = $\frac{1}{\sqrt{N}} \sqrt{\sum_{i=1}^N (x_i^k - \bar{x}^k)^2}$
- **Residual error** = $\| Ax^k - b \|$ (consistent case); $\| A^T(Ax^k - b) \|$ (inconsistent case)

In all tests we used $x^0 = 0$, 60 iterations of the corresponding algorithms (as in [4]).
Test 1: Consistent case, classical algorithms
We applied for the consistent problems associated to both phantoms, the algorithms Kaczmarz (33), Cimmino (75) with $\omega_i = 1, \forall i = 1, \dots, m$, Kaczmarz Extended (38)-(40) and Cimmino Extended (77)-(79) with $\omega_i = 1, \forall i = 1, \dots, m$ and $\alpha_j = 1, \forall j = 1, \dots, n$. The results are presented in figures 1-4 and indicate a better behavior for Kaczmarz algorithm (an well known property; see e.g. [2].
Test 2: Inconsistent case, classical algorithms
We performed similar tests as in the above **Test 1**, but for the inconsistent problems associated to both phantoms. The influence of the perturbed right hand side in (1) can be seen in figures 5-8, by comparing them with figures 1-4, respectively. Also according to the corresponding theoretical results, the classical versions of the Kaczmarz and Cimmino algorithms do not handle very well the inconsistency.
Test 3: Inconsistent case, extended algorithms
We applied for the inconsistent problems associated to both phantoms, the algorithms Kaczmarz Extended and Cimmino Extended. The results presented in figures 9 - 12, by comparing them with figures 5 - 8 indicate better results than in **Test 2** where the classical versions of the algorithms have been used.
Acknowledgements. All these data have been provided to us by Prof. Dr. Tommy Elfving and Dr. Touraj Nikazad from Linköping University Sweden. We express them our gratitude.

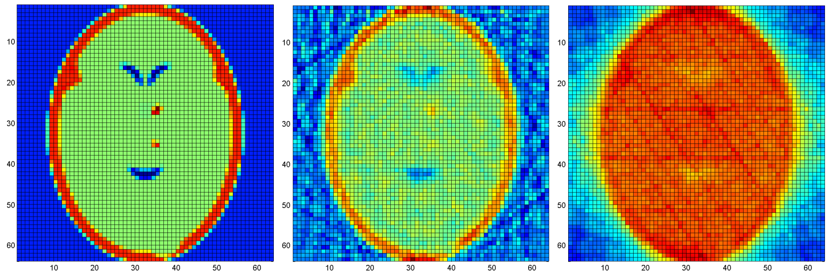

Figure 1: HEAD consistent; left: exact, middle: Kaczmarz, right: Cimmino.

References

[1] Bjork A., *Numerical methods for least squares problems*, SIAM Philadelphia, 1996.

[2] Censor Y., Stavros A. Z. *Parallel optimization: theory, algorithms and applications*, "Numer. Math. and Sci. Comp." Series, Oxford Univ. Press, New York, 1997.

[3] Censor Y., Elfving T., Herman G. T., Nikazad T. *On diagonally relaxed orthogonal projection methods*, SIAM J. Sci. Comp., **30**(2007/08), 473 - 504.

[4] Censor Y., Gordon D., Gordon R. *Component averaging: An efficient iterative parallel algorithm for large and sparse unstructured problems*, Parallel Computing **27**(2001), 777 - 808.

[5] Cimmino G. *Calcolo approssiomatto per le soluzioni dei sistemi di equazioni lineari*, Ric. Sci. progr. tecn. econom. naz. **1** (1938), pp. 326-333.

[6] Elfving T., *A stationary iterative pseudoinverse algorithm*, *B I T*, **38(2)** (1998), 275-282.

[7] Herman, G. T., *Image reconstruction from projections. The fundamentals of computerized tomography*, Academic Press, New York, 1980.

[8] Kaczmarz S. *Angenaherte Auflosung von Systemen linearer Gleichungen*, Bull. Acad. Polonaise Sci. et Lettres **A** (1937), pp.355-357.

[9] Kammerer W. J., Nashed M. Z., *On the convergence of the conjugate gradient method for singular linear operator equations*, SIAM J. Numer. Anal., **9**(1972), 165–181.

[10] Natterer F., *The Mathematics of Computerized Tomography*, John Wiley and Sons, New York, 1986.

[11] Petra, S., Popa C. and Schnörr C., *Extended and Constrained Cimmino-type Algorithms with Applications in Image Reconstruction*, IWR Preprint No. 8798 (2008), University of Heidelberg.

[12] Popa C., *Extensions of block-projection methods with relaxation parameters to inconsistent and rank-defficient least-squares problems*, *B I T*, **38(1)**(1998), 151–176.

[13] Popa C. and Zdunek R., *Kaczmarz extended algorithm for tomographic image reconstruction from limited-data*, Math. and Computers in Simulation, **65**(2004), 579–598.

[14] Popa C., *A hybrid Kaczmarz - Conjugate Gradient algorithm*, FAU Erlangen-N ürnberg, Lehrstuhl für Informatik 10 (Systemsimulation), Technical Report 07-2(2007).

[15] Tanabe K., *Projection Method for Solving a Singular System of Linear Equations and its Applications*, Numer. Math., **17**(1971), 203–214.

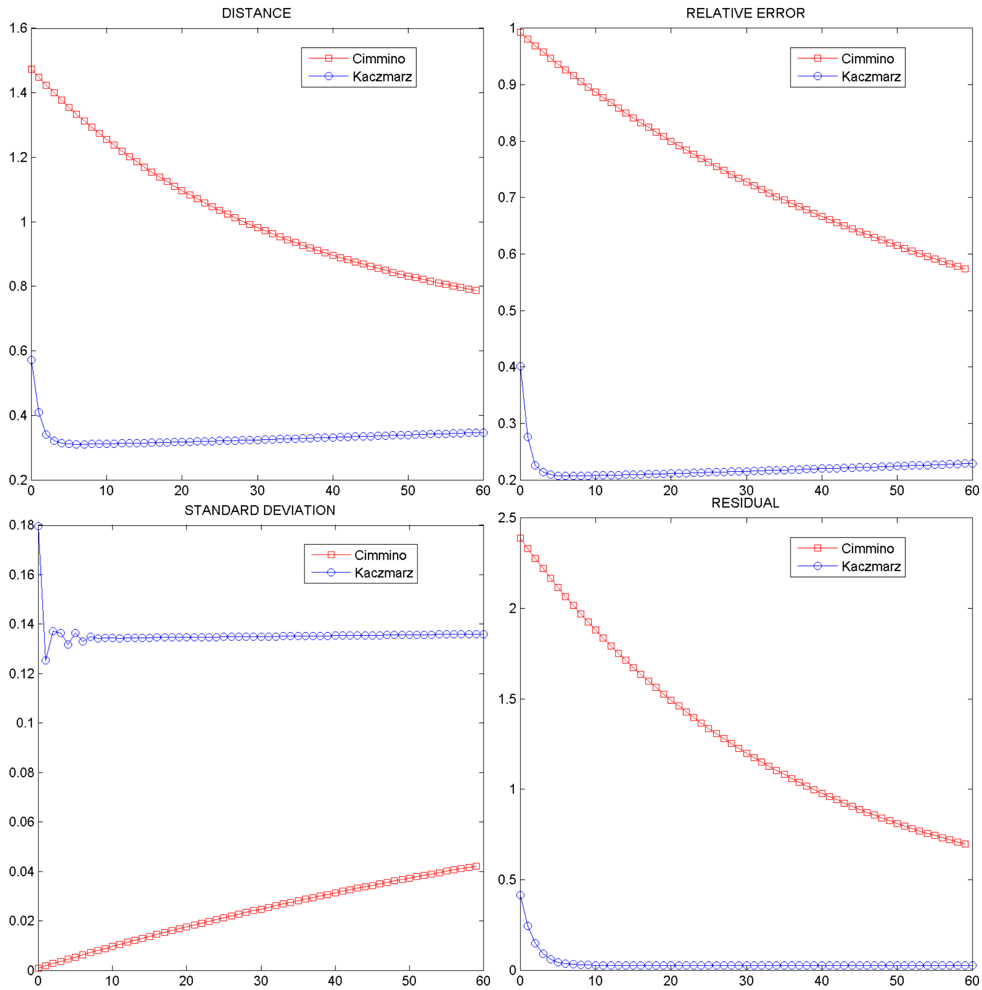

Figure 2: HEAD consistent; errors

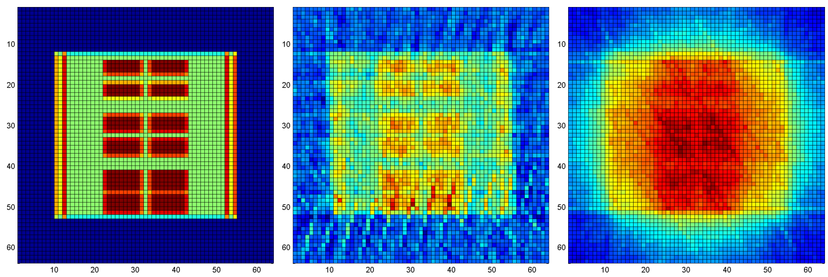

Figure 3: MIT consistent; left: exact, middle: Kaczmarz, right: Cimmino.

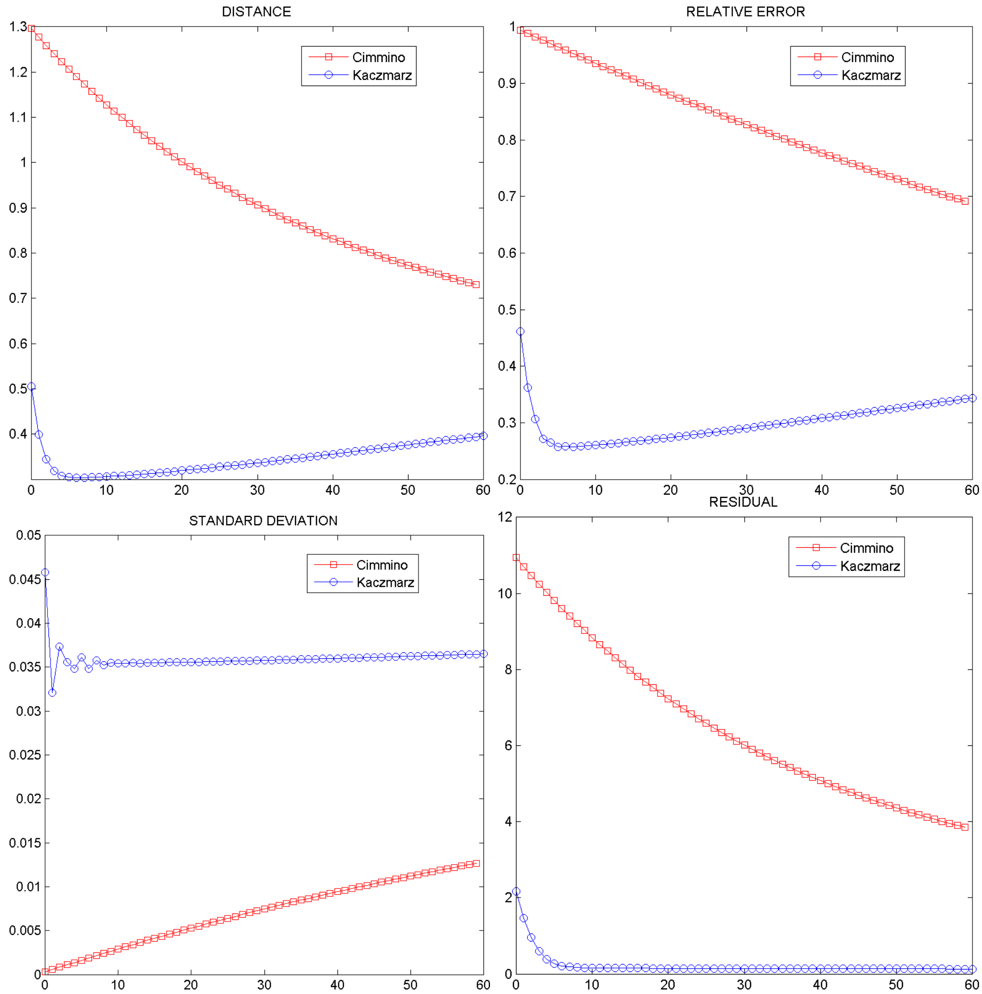

Figure 4: MIT consistent; errors

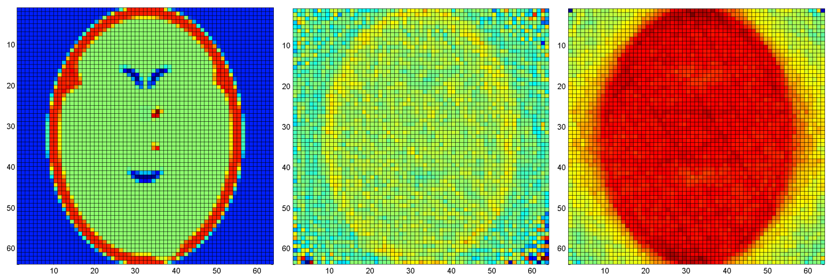

Figure 5: HEAD inconsistent; left: exact, middle: Kaczmarz, right: Cimmino.

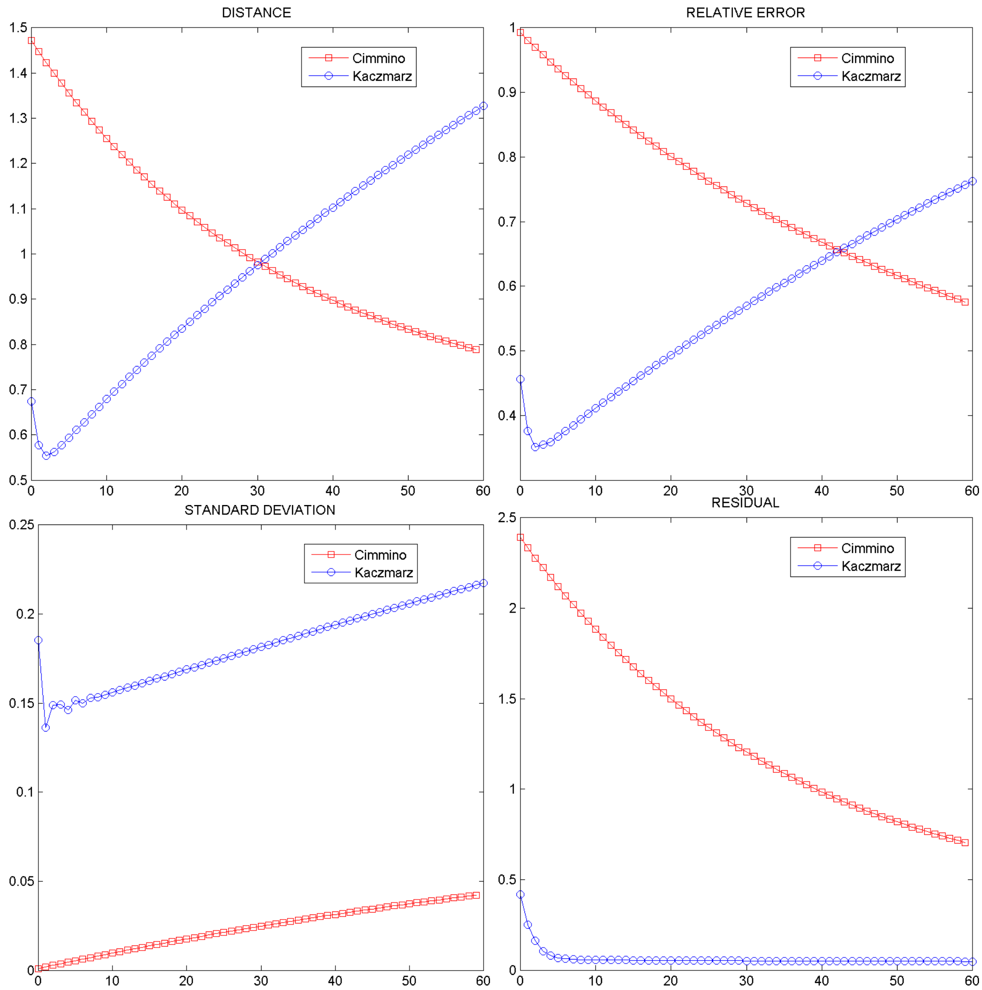

Figure 6: HEAD inconsistent; errors (classical algorithms)

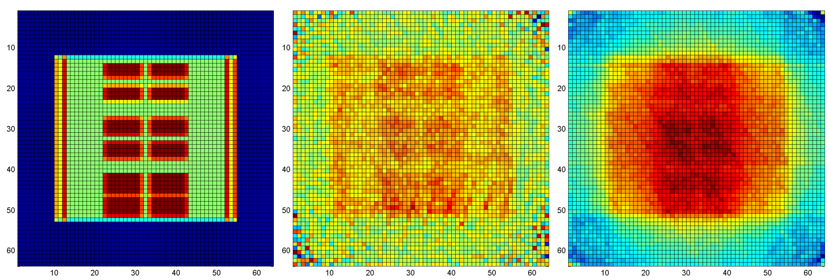

Figure 7: MIT inconsistent; left: exact, middle: Kaczmarz, right: Cimmino.

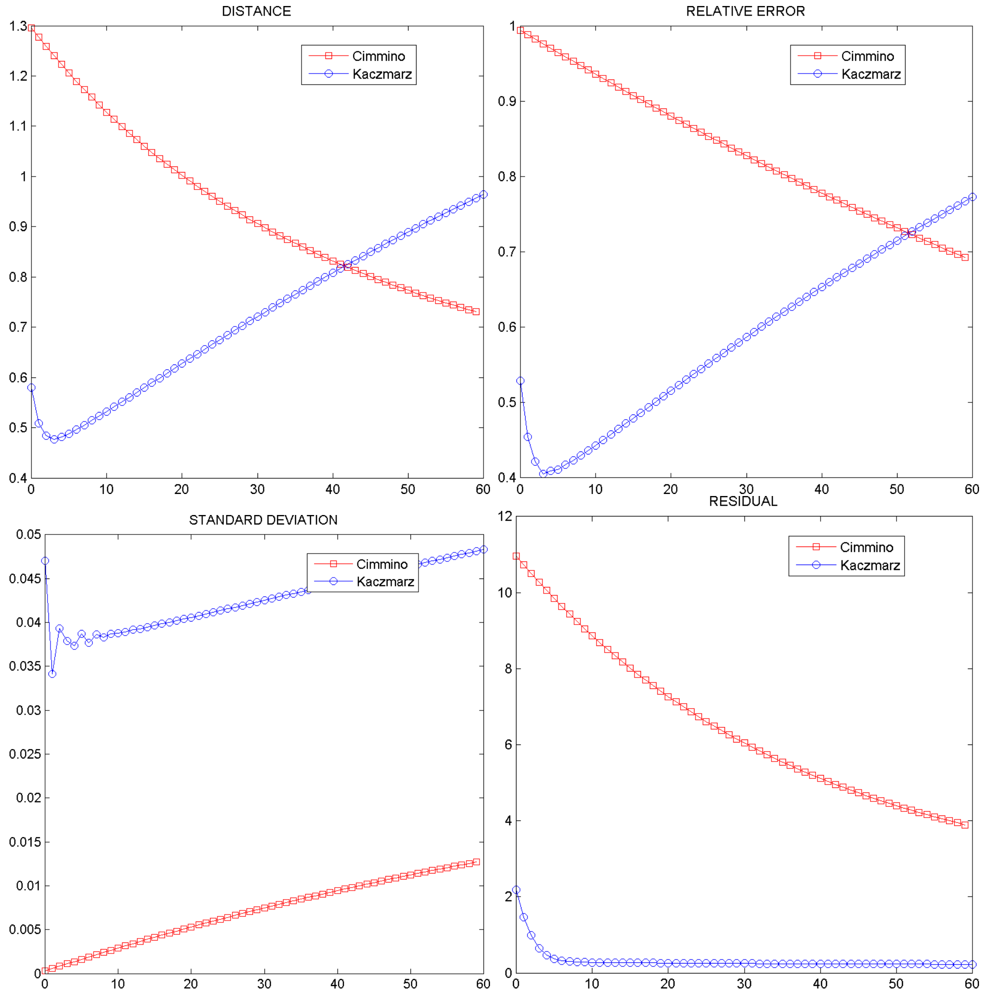

Figure 8: MIT inconsistent; errors (classical algorithms)

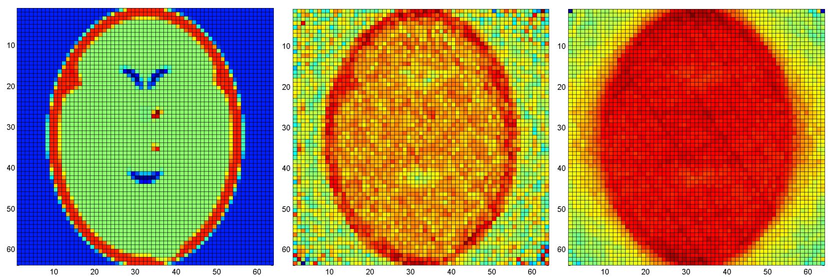

Figure 9: HEAD inconsistent; left: exact, middle: Kaczmarz Extended, right: Cimmino Extended.

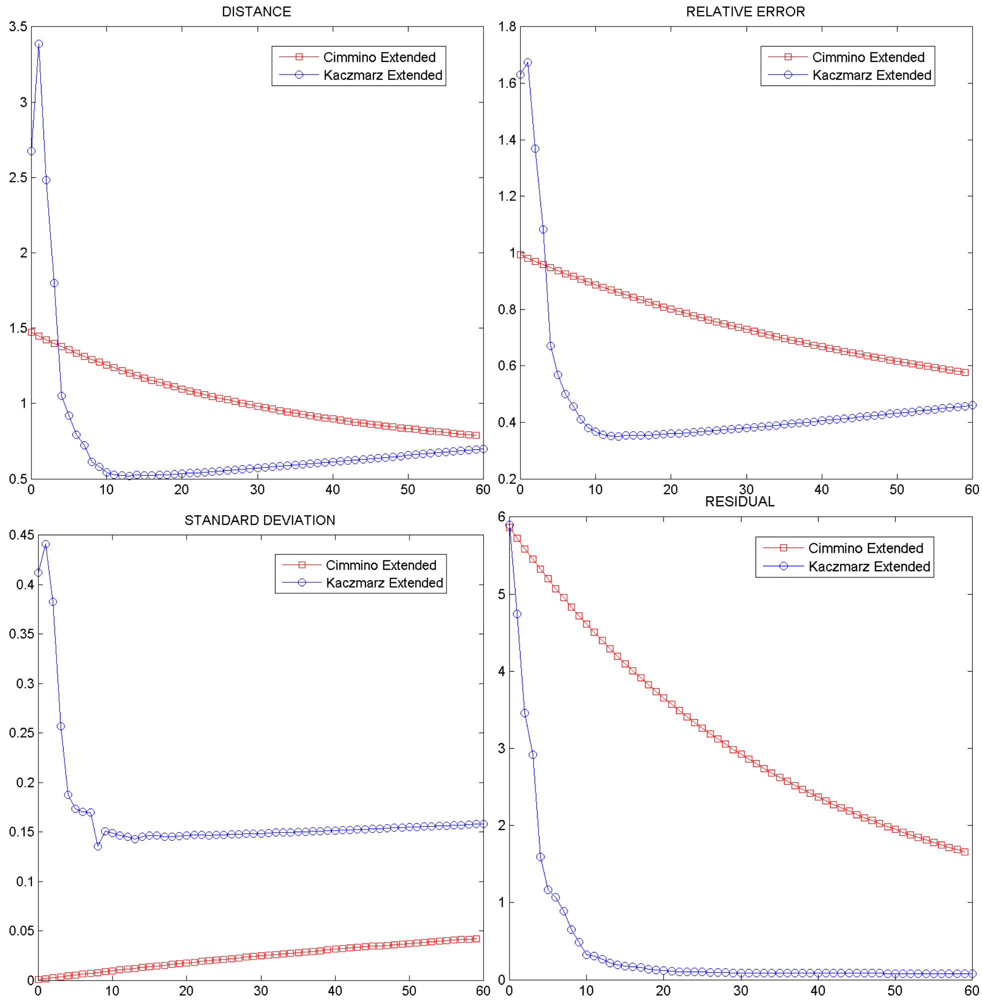

Figure 10: HEAD inconsistent; errors (extended algorithms)

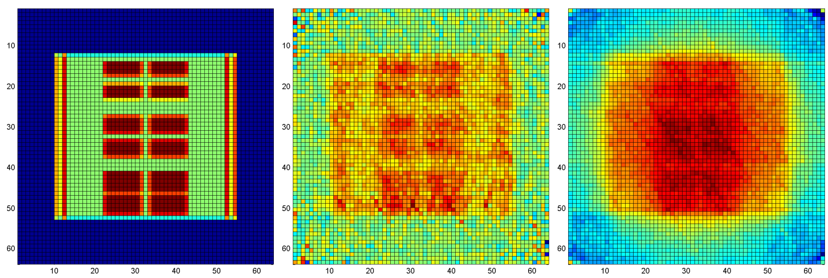

Figure 11: MIT inconsistent; left: exact, middle: Kaczmarz Extended, right: Cimmino Extended.

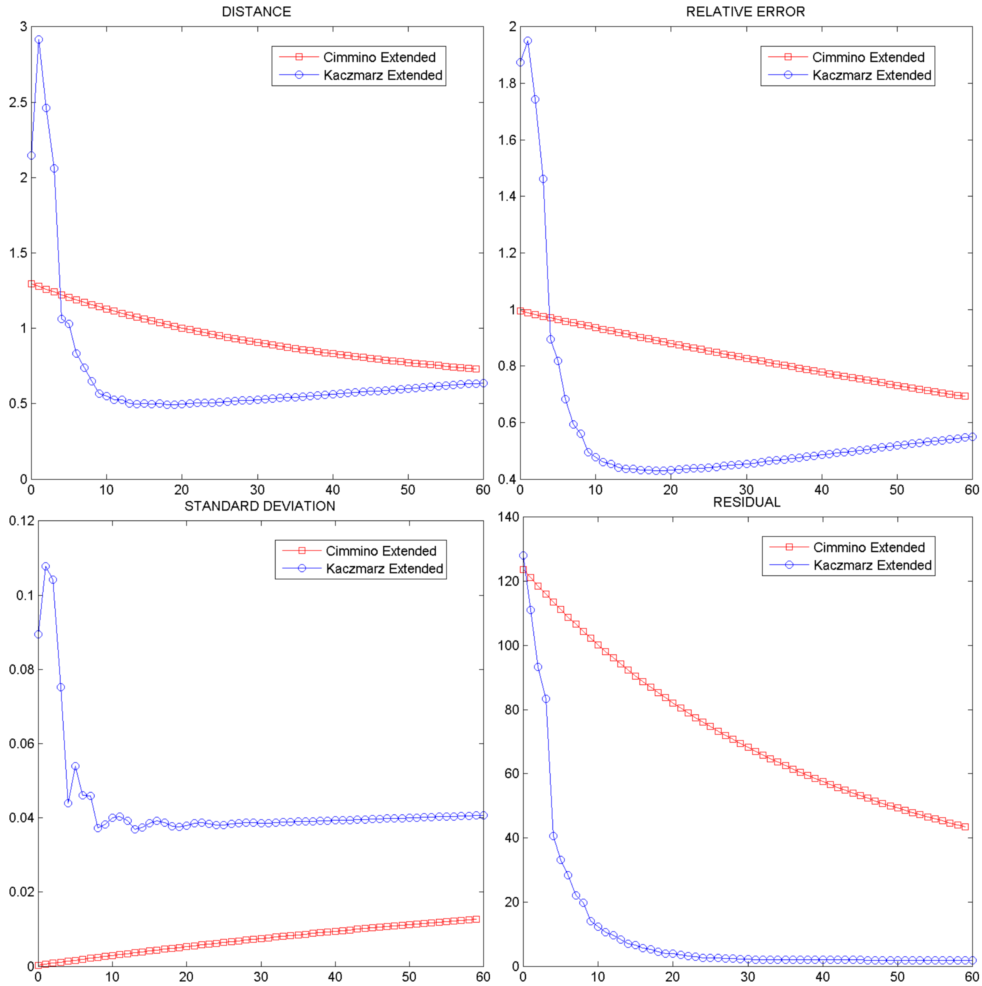

Figure 12: MIT inconsistent; errors (extended algorithms)