Extended and Constrained Cimmino-type Algorithms with Applications in Tomographic Image Reconstruction

Stefania Petra[1], Constantin Popa,[2] Christoph Schnörr[3]

Abstract. In the first part of this paper we propose an extension of Cimmino's reflections algorithm to inconsistent least squares problems. For proving convergence of the extended algorithm we first make a convergence analysis of the classical method that reveals other interesting aspects which don't appear in the convergence analysis made in the original paper by G. Cimmino. In the second part of the paper we introduce in both original and extended Cimmino algorithms constraining procedures and prove convergence of mixed algorithms in both cases to constrained classical or least squares solutions, respectively. In the third part of the paper we present numerical experiments on some algebraic image reconstruction models, with particular attention to tomographic particle image reconstruction, which illustrate the performance of both extended and constrained algorithms.

Keywords: Cimmino and Cimmino Extended algorithms, constraining strategies, algebraic image reconstruction, tomographic particle image reconstruction

2000 MS Classification: 65F10, 65F22

1 Introduction

One of the basic problems in scientific computation consists in the solution of systems of linear algebraic equations. Almost all problems of computational mathematics boil down, in the end, to the solution of such systems, often

[1] petra@math.uni-heidelberg.de, University of Heidelberg, Dept. of Mathematics and Computer Science, Germany

[2] cpopa@univ-ovidius.ro, Ovidius University Constanta, Dept. of Mathematics and Computer Science, Romania

[3] schnoerr@math.uni-heidelberg.de, University of Heidelberg, Dept. of Mathematics and Computer Science, Germany

of very large dimensions. A classical example is image reconstruction from limited-view projections. Upon discretization these problems are reduced to linear algebraic systems of the form

$$Ax = b \ , \tag{1}$$

with rank-deficient system matrix A and b a measurement vector. It is well known that the minimal ℓ_2-norm of the least squares problem

$$\min \|Ax - b\| \ , \tag{2}$$

is the best (minimum variance) unbiased estimator for the "true" solution x of (1), due to inevitable measurement errors in b. Moreover, the errors in b often make the problem (1) inconsistent, i.e. $b \notin \mathcal{R}(A^T)$. Thus, the pursuit of the least squares solution of (2) instead of (1) turns out to be a basic necessity. However, the minimum norm solution of (2) or (1) (in the consistent case) may considerably differ from the true solution, and hence a priori knowledge is needed to improve the reconstruction. Usually we have a priori information about the range within the values of the components of acceptable image vectors must lie, i.e.

$$x \in \mathcal{B} \ , \tag{3}$$

with $\mathcal{B}$ some compact set in $\mathbb{R}^n$. This assumption can substantially regularize the ill-posed image reconstruction problems as we will see in Section 6 and should be exploited by the iterative method considered for solving (2). Iterative methods, preferred over direct methods due to the huge dimensionality of A in real-world applications, produce a sequence of successive approximations x^k which, under appropriate conditions, converge to a solution of (2) as $k \to \infty$. Hence, we are interested in techniques able to steer the approximations x^k in the set $\mathcal{B}$. Such techniques traditionally termed as *constraining strategies* were investigated in [9] applied to iterative methods for consistent systems (1) in particular for Kaczmarz-like methods. For the inconsistent case such constraining strategies are investigated in [16] for Kaczmarz-type algorithms, likewise. In Kaczmarz's method, the current approximation x^k is orthogonally sequentially projected onto the hyperplanes defined by the rows of A. The projection onto the last hyperplane is taken as the new approximation x^k, and the process is repeated. The method was rediscovered in the field of computerized tomography, where the method of Kaczmarz was rediscovered around 1970 under the name of *Algebraic Reconstruction Technique (ART)*; see [3, 5]. Convergence results of ART are well established also for the inconsistent case, see e.g. [17]. A different approach was adopted by the authors in [14], where a Kaczmarz-type algorithm, called *Kaczmarz Extended with Relaxation Parameters (KERP)*, employs a modified right-hand side vector b to deal with the inconsistent case. Convergence of KERP towards a least-squares solution was proved.

In this work we follow a similar approach as the authors in [9] and [14, 16] and will investigate the compatibility of Cimmino‘s method and its extension in the above mentioned sense to constraining strategies.

Cimmino's method, the method of simultaneous orthogonal reflections with respect to the every hyperplane described by every equation in (1) simply writes as

$$x^{k+1} = x^k + \frac{2}{\omega} A^T D^T D(b - Ax^k) \ ,$$

with some $\omega > 0$ and D some diagonal matrix, see also next section. Cimmino shows that the method is always convergent, even in the inconsistent case: no restriction is imposed on the system matrix A except the extremely mild one of having rank at least 2. We will refine this result by giving an expression for the limit of the above sequence $\{x^k\}$ in Section 2. This result will be used in Section 3 to prove the convergence of an extended version of Cimmino's method to a least squares solution even in the inconsistent case. We stress that the above approximations $\{x^k\}$ in Cimmino's original method converge in the inconsistent case to a weighted least squares solution, i.e.

$$\min \|D(Ax - b)\| \ , \tag{4}$$

which might substantially differ from the least squares solution (2). The proposed extended Cimmino algorithm employs a corrected right-hand side b^k which approaches the projection of the given measurement vector b onto the range of A^T, thus the inconsistent system $Ax \approx b$ will approach its consistent counterpart $Ax = P_{\mathcal{R}(A^T)}(b)$. A constrained version for the proposed extended Cimmino algorithm is investigated in Section 5, where the iterates of the extended Cimmino algorithm are steered in the range $\mathcal{I}(C)$ of a constraining function C. Convergence of the new algorithm towards a point in the intersection $LSS(A, b) \cap \mathcal{I}(C)$ is proved for weaker assumptions on the function C than in the case of the constrained original Cimmino algorithm which we investigate in Section 4. Here C has to be strictly nonexpansive to obtain convergence towards a point within the intersection of $\mathcal{I}(C)$ and the solution set of the weighted least squares problem (4). Similar results are obtained for the Kaczmarz-type methods by the author in [16].

In this respect the methods of Cimmino and Kaczmarz are closely related. However, Cimmino's algorithm has been found to be better suited for parallel computers, whereas Kaczmarz's method tends to converge somewhat faster. This can be best understood by observing that Cimmino's method is equivalent to a damped Jacobi iteration applied to the normal equations $A^TAx = A^Tb$. Likewise, Kaczmarz method is equivalent to the classical Gauss-Seidel method applied to the system $AA^Ty = b$, with $A^Ty = x$, see e.g. [18]. A further attractive feature of both methods is their extremely low storage demand. This is especially evident for Kaczmarz's method, which only requires one row of the coefficient matrix A to perform one iteration: in other words, Kaczmarz's method is a *row action* method, see [3]. Cimmino's method and our proposed extensions can also be implemented in this way. As for parallel computing, as we already comment, Cimmino's method and our proposed extensions is ideally suited for parallel implementation, and stress that our interest in old algorithms like Cimmino's is primarily due to this feature, having in mind the huge dimensionality of the real-world application we have to tackle in the field of image reconstruction.

Our notation is standard; in particular we will denote by $\| \cdot \|$, the Euclidean ℓ_2-norm and by $\| \cdot \|_1$ the ℓ_1-norm in the n-dimensional real vector space $\mathbb{R}^n$. For a $m \times n$ matrix A, A_i is the i-th row of A, A^j is the j-th row of A and a_{ij} denotes the i, j-th element of A. In what follows we shall suppose that the matrix A has nonzero rows A_i and columns A^j, i.e.

$$A_i \neq 0, i = 1, \ldots, m, \;\; A^j \neq 0, j = 1, \ldots, n. \tag{5}$$

The (nonempty) set of all least-squares solution of (2) will be denoted by $LSS(A, b)$ and the minimal ℓ_2-norm solution in $LSS(A, b)$ by x_{LS}. The null space of an matrix M will be denoted by $\mathcal{N}(M)$, whereas by $\mathcal{R}(M)$ we will denote the range of M. The projection onto linear subspaces $\mathcal{S}$ will be denoted by $P_{\mathcal{S}}$. The same symbol $P_{\mathcal{S}}$ is often used to denote the matrix and the linear mapping it represents. The image of a (nonlinear) mapping C will be denoted by $\mathcal{I}(C)$.

2 Original Cimmino's algorithm revisited

In [4], Cimmino considers a system of linear algebraic equations $Ax = b$ where A is a real matrix (initially assumed square and nonsingular). A solution point will lie in the intersection of the m hyperplanes described by

$$H_i := \{x \mid A_i^T x = b_i\}, \quad i = 1, \ldots, m. \tag{6}$$

Given a current approximation x^k, Cimmino takes, for each $i = 1, \ldots, m$ the reflection $y^{k,i}$ of x^k with respect to the hyperplane (6)

$$y^{k,i} = x^k + 2\frac{b_i - A_i^T x^k}{\|A_i\|^2} A_i \ . \tag{7}$$

Given m arbitrarily chosen positive quantities

$$\omega_i > 0, \quad \sum_{i=1}^{m} \omega_i =: \omega \ , \tag{8}$$

the next iterate x^{k+1} is the center of gravity of the system formed by placing the m masses ω_i at the points $y^{k,i}$ given by (7). Cimmino notes that the point x^k and its reflections with respect to the hyperplanes (6) all lie on a hypersphere the center of which is precisely a point x common to all hyperplanes. Because the center of gravity of the system of masses ω_i must necessarily fall inside this hypersphere, it follows that the new iterate x^{k+1} is a better approximation to the solution x^k,

$$\|x^{k+1} - x\| < \|x^k - x\| \ .$$

The above considerations can be summarized as taking

$$x^{k+1} = \sum_{i=1}^{m} \frac{\omega_i}{\omega} y^{k,i} = x^k + 2\sum_{i=1}^{m} \frac{\omega_i}{\omega} \frac{b_i - A_i^T x^k}{\|A_i\|^2} A_i \ ,$$

thus, Cimmino's method can be written as follows
Algorithm Cimmino (C). Let $x^0 \in \mathbb{R}^n$; for $k = 0, 1, \ldots$ do

$$x^{k+1} = x^k + \frac{2}{\omega} A^T D^T D(b - Ax^k) \ , \tag{9}$$

where

$$D := \begin{pmatrix} \frac{\sqrt{\omega_1}}{\|A_1\|} & & & \\ & \frac{\sqrt{\omega_2}}{\|A_2\|} & & \\ & & \ddots & \\ & & & \frac{\sqrt{\omega_m}}{\|A_m\|} \end{pmatrix} . \tag{10}$$

Cimmino showed the following result:

Theorem 1 *[4] Provided that the rows and columns of A satisfy (5) and*

$$rank(A) \geq 2 , \tag{11}$$

then for any initial approximation $x^0 \in \mathbb{R}^n$ the sequence $\{x^k\}$ generated by (9) *converges*

(i) to one solution of $Ax = b$, if the problem (1) *is consistent;*

(ii) to a solution of the perturbed least squares problem

$$\min_x \|D(Ax - b)\| . \tag{12}$$

As we can see from the previous theorem, Cimmino's algorithm (9) gives approximations of the solutions of (1) only in the consistent case. In the inconsistent case, because the positive weights ω_i are arbitrary the solutions of the weighted least squares problem (12) can be far enough from the least squares solutions of the original one (12). For overcoming this difficulty we shall present in the next section an extended version of Cimmino's original method (9).

Remark 1 *From* (10) *it results that, for the particular choice $\omega_i = \|A_i\|^2$, $\forall i = 1, \ldots, m$ we get $D = I$ thus problem* (12) *is exactly the initial one* (2). *So, for the above particular choice for ω_i (and only for this !) the original Cimmino's algorithm* (9) *converges also in the inconsistent case to a solution of* (2) *(see for details [15]).*

As will be presented in Section 3, our extension of Cimmino's algorithm (9) is based on an extension of Kaczmarz's method, proposed by one of the authors in [12] (see also [14]), for inconsistent and rank-deficient least squares problems (2). From this view point we need a convergence analysis for it similar with that made for Kaczmarz's method in [17] (see also [13]). In particular we need to gain more knowledge about the limit of the sequence in (9) in dependence of the starting point x^0. This will be done in the rest of this section.

To this end, we rewrite the iteration in (9) as

$$x^{k+1} = Tx^k + Rb \ , \tag{13}$$

where

$$T := I - \frac{2}{\omega} A^T D^T D A \tag{14}$$

and

$$R := \frac{2}{\omega} A^T D^T D \ . \tag{15}$$

First, we observe that T can be written as

$$T := \sum_{i=1}^{m} S_i \ , \tag{16}$$

where S_i is the Householder linear transformation

$$S_i := I - \frac{2}{\omega} \frac{A_i A_i^T}{\|A_i\|^2} \ , \tag{17}$$

i.e. the orthogonal reflector with respect to the reflection hyperplane orthogonal to A_i. Next we will study several important properties of the linear operator T and note that $\mathcal{N}(A)$ and $\mathcal{R}(A^T)$ are invariant subspaces of T.

Lemma 1 *(i) If $x \in \mathcal{N}(A)$ then $Tx = x \in \mathcal{N}(A)$.*

(ii) If $x \in \mathcal{R}(A^T)$ then $Tx \in \mathcal{R}(A^T)$.

(iii) For any $y \in \mathbb{R}^m$, $Ry \in \mathcal{R}(A^T)$.

Proof. The statements in (i) - (iii) follow directly from (14) and (15). □

Now we will see that T from (14) is contractive on $\mathcal{R}(A^T)$.

Lemma 2 *(i) For any $x \in \mathbb{R}^n$, $\|S_i x\| = \|x\|$; in particular, $\|S_i\| = 1$ for all $i = 1, \ldots, m$.*

(ii) The matrix T satisfies

$$\|T\| = 1 \ . \tag{18}$$

(iii) If $\operatorname{rank}(A) \geq 2$*, we have*

$$\|T|_{\mathcal{R}(A^T)}\| \ < \ 1 \ , \tag{19}$$

where by $T|_{\mathcal{R}(A^T)}$ we denoted the restriction of T to the corresponding linear subspace $\mathcal{R}(A^T)$.

(iv) $\|Tx\| = \|x\| \Longleftrightarrow x \in \mathcal{N}(A)$.

Proof. Since S_i is orthogonal, statement (i) follows directly. Indeed,

$$S_i^T S_i = \left(I - 2\frac{A_i A_i^T}{\|A_i\|^2}\right)^2 = I - 4\frac{A_i A_i^T}{\|A_i\|^2} + 4\frac{A_i A_i^T A_i A_i^T}{\|A_i\|^4} = I.$$

Hence, $\|S_i x\|^2 = \|x\|^2$ for all $i = 1, \ldots, m$ and thus (i) holds.

(ii) For an arbitrary $x \in \mathbb{R}^n$ we get

$$\|Tx\| = \|\sum_{i=1}^{m} \frac{\omega_i}{\omega} S_i x\| \leq \sum_{i=1}^{m} \frac{\omega_i}{\omega} \|S_i x\| \overset{(8),(i)}{=} \|x\| \tag{20}$$

By Lemma 1 (i) $\|T\| = 1$ now follows.

(iii) Let $x \in \mathcal{R}(A^T) \setminus 0$. Hence $x \notin \mathcal{N}(A)$ as $\mathbb{R}^n = \mathcal{R}(A^T) \oplus \mathcal{N}(A)$. By the nonsingularity of S_i we have $S_i x \neq 0$ for all $i = 1, \ldots, m$. Since the ℓ_2-norm is strictly convex, $\omega_i > 0$ and $\sum_{i=1}^{m} \frac{\omega_i}{\omega} = 1$ the equality in (20) only holds if $S_1 x = \cdots = S_m x$. Let us suppose that $S_1 x = S_i x$ for all $i = 2, \ldots, m$. This is equivalent to

$$\frac{A_1^T x}{\|A_1\|^2} A_1 - \frac{A_i^T x}{\|A_i\|^2} A_i = 0 \quad \text{for all } i = 2, \ldots, m.$$

Since $\operatorname{rank}(A) \geq 2$ the equalities above imply that $A_i^T x = 0$ for all i=1,. . . ,m. But this contradicts $x \notin \mathcal{N}(A)$. Hence we showed that the reflectors of x with respect to every hyperplane orthogonal to A_i cannot be all equal. Thus, the strict inequality in (20) holds for all $x \in \mathcal{R}(A^T) \setminus 0$.

The implication ”$\Leftarrow$” in (iv) follows directly from Lemma 1 (i) whereas the reverse implication ”$\Rightarrow$” follows from (iii). □

Next we are interested in the recursive application of T.

Lemma 3 *We have*

$$T^k = P_{\mathcal{N}(A)} + \widetilde{T}^k \ , \tag{21}$$

where $\widetilde{T} := TP_{\mathcal{R}(A^T)}$ *and* $T^k = TT^{k-1}$ *with* $T^0 = I$*.*

Proof. We will use an induction argument. First, we decompose the operator T from as

$$T = TI = T\left(P_{\mathcal{N}(A)} + P_{\mathcal{R}(A^T)}\right) \overset{Lem.1(i)}{=} P_{\mathcal{N}(A)} + \widetilde{T} \ .$$

Thus, the statement in (21) holds for $k = 1$. Let us assume it for an arbitrary k. Then

$$\begin{aligned} T^{k+1} = TT^k &= \left(P_{\mathcal{N}(A)} + \widetilde{T}\right)\left(P_{\mathcal{N}(A)} + \widetilde{T}^k\right) \\ &= P^2_{\mathcal{N}(A)} + P_{\mathcal{N}(A)}\widetilde{T}^k + \widetilde{T}P_{\mathcal{N}(A)} + \widetilde{T}^{k+1} \\ &= P_{\mathcal{N}(A)} + \widetilde{T}^{k+1} \end{aligned}$$

holds, since

$$P_{\mathcal{N}(A)}\widetilde{T}x = P_{\mathcal{N}(A)} \underbrace{TP_{\mathcal{R}(A^T)}x}_{\in\mathcal{R}(A^T)} = 0 \quad \text{for all } x \in \mathbb{R}^n \tag{22}$$

and

$$\widetilde{T}P_{\mathcal{N}(A)} = T \underbrace{P_{\mathcal{R}(A^T)}P_{\mathcal{N}(A)}}_{=0} = 0 \ . \tag{23}$$

□

We can now prove a convergence result, similar with Cimmino's original one from Theorem 1, in which we give more information about the expression for the limit of the sequence of approximations.

Theorem 2 *Let us suppose that the matrix A satisfies* (5) *and* (11)*. Then the following hold.*

(i) For any initial approximation $x^0 \in \mathbb{R}^n$, the sequence $\{x^k\}$ generated by Cimmino's algorithm (9) *converges and its limit is given by*

$$\lim_{k\to\infty} x^k = P_{\mathcal{N}(A)}(x^0) \ + \ \left(I - \widetilde{T}\right)^{-1} Rb \ . \tag{24}$$

(ii) If the problem (1) *is consistent, i.e.* $b \in R(A)$ *then*

$$\left(I - \widetilde{T}\right)^{-1} Rb = x_{LS} \tag{25}$$

and the limit point in (24) *is one of its solutions.*

Proof. (i) Using (13) and a recursive argument, we obtain

$$x^k = Tx^{k-1} + Rb = T\left(Tx^{k-2} + Rb\right) + Rb = \cdots = T^k x^0 + \sum_{j=0}^{k-1} T^j Rb \ ,$$

as for Kaczmarz method in [17]. Now using Lemma 3 we obtain

$$\begin{aligned} x^k &= \widetilde{T}^k x^0 + P_{\mathcal{N}(A)}(x^0) + \sum_{j=0}^{k-1} T^j Rb \\ &\overset{(27)}{=} \widetilde{T}^k x^0 + P_{\mathcal{N}(A)}(x^0) + \sum_{j=0}^{k-1} \widetilde{T}^j Rb \ , \end{aligned} \tag{26}$$

since

$$T^j R = \widetilde{T}^j R, \ \forall j \in \mathbb{N} \ , \tag{27}$$

holds by Lemma 1 (iii) and the definition of $\widetilde{T}$.

Since $\|\widetilde{T}\| < 1$ the Neumann series $\sum_{j=0}^{\infty} \widetilde{T}^j$ converges and we obtain

$$\lim_{k\to\infty} \widetilde{T}^k x^0 = 0 \text{ and } \lim_{k\to\infty} \sum_{j=0}^{k-1} \widetilde{T}^j Rb = \left(I - \widetilde{T}\right)^{-1} Rb \ ,$$

which gives us in view of (26) exactly the statement in (24).
(ii) It is well known (see e.g. [2]) that the consistency assumption, $b \in R(A)$ is equivalent with the equality

$$AGb = b \ ,$$

where G is a matrix (the generalized inverse of A) that satisfies

$$AGA = A. \tag{28}$$

Moreover, in this case the vector Gb is the minimal norm solution of the system $Ax = b$. According to the above considerations, (25) will hold if we prove that the matrix G given by

$$G = \left(I - \widetilde{T}\right)^{-1} R \tag{29}$$

satisfies (28). To this end, we observe that T and R from (14) - (15) satisfy

$$I - T = RA\ . \tag{30}$$

Indeed, $I - RA = I - \frac{2}{\omega} A^T D^T D A = T$.

Finally we obtain

$$\begin{aligned} AGA &= A\left(I - \widetilde{T}\right)^{-1} RA \overset{(30)}{=} A\left(I - \widetilde{T}\right)^{-1} (I - T) \\ &\overset{(21)}{=} A\left(I - \widetilde{T}\right)^{-1}\left((I - \widetilde{T}) - P_{\mathcal{N}(A)}\right) \\ &= A - A\left(I - \widetilde{T}\right)^{-1} P_{\mathcal{N}(A)} = A - A\sum_{j=0}^{\infty} \underbrace{\widetilde{T}^j P_{\mathcal{N}(A)}}_{=0} \\ &= A\ , \end{aligned}$$

which completes the proof.

Remark 2 *From the above theorem, it results that, in the consistent case any solution of* (1) *can be obtained as a limit point in (24), for appropriate values of* x^0. *Although in practice we usually approximate the minimal* ℓ_2 *norm solution, recent trend has been to replace the* ℓ_2 *norm with an* ℓ_1 *norm. This* ℓ_1 *regularization has many of the beneficial properties of* ℓ_2 *regularization, but yields sparse solutions of* (1) *and* (2) *that are better suited for a large set of applications. This will be illustrated also in Section 6.*

Remark 3 *It turns out that Cimmino's algorithm* (9) *is a damped gradient method applied to*

$$f_D := \frac{1}{2}\|D(Ax - b)\|^2\ , \tag{31}$$

thus

$$x^{k+1} = x^k - t_k \nabla f_D(x^k) \ ,$$

with constant stepsize $t_k = \frac{2}{\omega}$. *Note that convergence of the overall method to a minima of* f_D *holds for this particular stepsize even in the case of a rank deficient matrix* A.

3 Cimmino Extended Algorithm

The extension that we propose for Cimmino's algorithm (9) is based on an extension of Kaczmarz's algorithm, previously proposed by one of the authors in [12] (see also [13]). It will be briefly presented in the first part of this section. To this end, let us suppose that the rows and columns of A satisfy (5). For an arbitrary starting point x^0 Kaczmarz's method can be written as

$$x^{k+1} := P_{H_m}(\dots(P_{H_2}(P_{H_1}(x^k)))\dots) \ , \tag{32}$$

where H_i denote the hyperplanes in (6) and $k = 0, 1, \dots$. We must mention that the order of the sequential projections in (32) can be chosen arbitrarily.

Likewise, the *Kaczmarz Extended (KE)* algorithm, can be written as follows.

Algorithm Kaczmarz Extended (KE). Let $x^0 \in \mathbb{R}^n, y^0 = b$;
for $k = 0, 1, \dots$ do

$$\begin{aligned} y^{k+1} &= P_{\hat{H}_n}(\dots(P_{\hat{H}_2}(P_{\hat{H}_1}(y^k)))\dots) =: \Phi(y^k) \ , && (33)\\ b^{k+1} &= b - y^{k+1} \ , && (34)\\ x^{k+1} &= P_{\widetilde{H}_m}(\dots(P_{\widetilde{H}_2}(P_{\widetilde{H}_1}(x^k)))\dots) =: Qx^k + Ub^{k+1} \ , && (35)\end{aligned}$$

where $\hat{H}_j := \{y \mid (A^j)^T y = 0\}$ and $\widetilde{H}_i := \{x \mid A_i^T x = b_i^{k+1}\}$.

In this extension, the auxiliary sequence $\{y^k\}$ is generated in (33) by applying Kaczmarz's algorithm (32) to the consistent system

$$A^T y = 0 \ , \tag{36}$$

with the initial approximation $y^0 = b$. According to the convergence results concerning Kaczmarz's method (32) (see e.g. [13], [17]) we know that

$$\lim_{k\to\infty} y^k = P_{\mathcal{N}(A^T)}(b), \quad \text{thus} \quad \lim_{k\to\infty} b^k = P_{\mathcal{R}(A)}(b).$$

In this way, in each iteration (34), the "new" right hand side b^k approaches the component $P_{R(A)}(b)$, i.e. the inconsistent problem $Ax \approx b = P_{\mathcal{N}(A^T)}(b) + P_{\mathcal{R}(A)}(b)$ approaches its consistent equivalent formulation $Ax = P_{\mathcal{R}(A)}(b)$.

By analogy, the "additional" part in the further proposed *Cimmino Extended (CE)* algorithm, will be the Cimmino's algorithm (9) applied to the consistent problem (36). In this end, we will consider other positive weights

$$\alpha_j > 0, \quad \alpha = \sum_{j=1}^{n} \alpha_j \tag{37}$$

and the "column" versions of the above operators T and D, denoted by $\mathfrak{T}$ and $\mathfrak{D}$ respectively, and defined as follows (see (14) and (10))

$$\mathfrak{D} := \begin{pmatrix} \frac{\sqrt{\alpha_1}}{\|A^1\|} & & & \\ & \frac{\sqrt{\alpha_2}}{\|A^2\|} & & \\ & & \ddots & \\ & & & \frac{\sqrt{\alpha_n}}{\|A^n\|} \end{pmatrix} \tag{38}$$

and

$$\mathfrak{T} := I - \frac{2}{\omega} A \mathfrak{D}^T \mathfrak{D} A^T \ . \tag{39}$$

Then, by analogy with the KE method (33) – (35), the Cimmino Extended (CE) algorithm can be written as:
Algorithm Cimmino Extended (CE). Let $x^0 \in \mathbb{R}^n, y^0 = b$;
for $k = 0, 1, \dots$ do

$$\begin{aligned} y^{k+1} &= \mathfrak{T} y^k \ , & (40) \\ b^{k+1} &= b - y^{k+1} \ , & (41) \\ x^{k+1} &= T x^k + R b^{k+1} \ . & (42) \end{aligned}$$

The following results for the operator $\mathfrak{T}$ and the sequence y^k can be obtained by simply replacing A^T with A in the corresponding analogous statements for T and x^k in Lemma 1, Lemma 2 and Theorem 2.

Lemma 4 *(i) $\mathcal{N}(A^T)$ and $\mathcal{R}(A)$ are invariant subspaces for the application $\mathfrak{T}$ and we have*

$$\mathfrak{T}^k = P_{\mathcal{N}(A^T)} + \widetilde{\mathfrak{T}}^k, \ \forall k \in \mathbb{N} \ , \tag{43}$$

where $\widetilde{\mathfrak{T}}$ *is defined by*

$$\widetilde{\mathfrak{T}} = \mathfrak{T}\ P_{\mathcal{R}(A)}\ . \tag{44}$$

(ii) The matrix $\widetilde{\mathfrak{T}}$ *has the property*

$$\parallel \widetilde{\mathfrak{T}} \parallel = \sup_{y \in \mathcal{R}(A)} \frac{\|\mathfrak{T} y\|}{\|y\|} < 1\ . \tag{45}$$

(iii) If $\{y^k\}$ *is the sequences defined in* (40) *where* $y^0 = b \in \mathbb{R}^m$ *then*

$$\lim_{k \to \infty} y^k = P_{N(A^T)}(b) = b - P_{R(A)}(b)\ . \tag{46}$$

Moreover, the projection of every iterate generated by CE algorithm on the null space of A equals the projection of the starting point x^0 on $\mathcal{N}(A)$.

Lemma 5 *Let* $\{x^k\}$ *be the sequence generated by CE algorithm* (40) – (42) *for an arbitrary initial approximation* $x^0 \in \mathbb{R}^n$*. Then*

$$P_{\mathcal{N}(A)}(x^k) = P_{\mathcal{N}(A)}(x^0),\ \forall k \in \mathbb{N}\ . \tag{47}$$

Proof. Using mathematical induction the result can be proved using Lemma 1 (i) and (iii) and $P_{\mathcal{N}(A)} P_{\mathcal{R}(A^T)} = 0$. □

We are now able to prove the main convergence result for the CE algorithm.

Theorem 3 *Let us suppose that the matrix A satisfies* (5) *and* (11)*. Then, for any initial approximation* $x^0 \in \mathbb{R}^n$*, the sequence* $\{x^k\}$ *generated by Cimmino Extended algorithm* (40) – (42) *converges and its limit is given by*

$$\lim_{k \to \infty} x^k = P_{\mathcal{N}(A)}(x^0)\ +\ x_{LS} \in LSS(A, b). \tag{48}$$

Proof. It is well known that $LSS(A, b) = \{x \in \mathbb{R}^n \mid Ax = P_{\mathcal{R}(A^T)} b\}$. Let G denote the generalized inverse of A. Then the minimal norm least squares solution satisfies $x_{LS} = G P_{\mathcal{R}(A^T)} b$. In the following we will try to evaluate the error vector $e_k := x^k - \big(P_{\mathcal{N}(A)}(x^0) - x_{LS}\big)$ using $G = (I - \widetilde{T})^{-1} R$ according to the proof of Theorem 2.

Following the proof from [13] we successively obtain

$$\begin{aligned} e_k &= x^k - (P_{\mathcal{N}(A)}(x^0) + x_{LS}) \\ &= Tx^{k-1} + Rb^k - (P_{\mathcal{N}(A)}(x^0) + x_{LS}) \\ &\overset{(43)}{=} P_{\mathcal{N}(A)}x^{k-1} + \widetilde{T}x^{k-1} + Rb^k - (P_{\mathcal{N}(A)}(x^0) + x_{LS}) \\ &\overset{(47)}{=} \widetilde{T}x^{k-1} + Rb^k - (I - \widetilde{T} + \widetilde{T})(I - \widetilde{T})^{-1}RP_{\mathcal{R}(A^T)}(b) \\ &= \widetilde{T}(x^{k-1} - x_{LS}) + R(b^k - P_{\mathcal{R}(A^T)}(b)) \\ &= \widetilde{T}(x^{k-1} - P_{\mathcal{N}(A)}(x^0) - x_{LS}) + R(b - y^k - P_{\mathcal{R}(A^T)}(b)) \\ &= \widetilde{T}e^{k-1} + R(P_{\mathcal{N}(A)}(b) - y^k) \ . \end{aligned} \tag{49}$$

On the other hand

$$P_{\mathcal{N}(A)}(b) - y^k \overset{(40)}{=} P_{\mathcal{N}(A)}(b) - \mathfrak{T}^k b \overset{(43)}{=} P_{\mathcal{N}(A)}(b) - (P_{\mathcal{N}(A)} + \widetilde{\mathfrak{T}}^k)b = -\widetilde{\mathfrak{T}}^k b \ . \tag{50}$$

Then, combining (49) and (50) we obtain

$$e^k = \widetilde{T}e^{k-1} - R\widetilde{\mathfrak{T}}^k b \tag{51}$$

and a recursive application of (51) gives us

$$e^k = \widetilde{T}^k e^0 - \sum_{j=1}^{k-1} \widetilde{T}^{k-j} R \widetilde{\mathfrak{T}}^j b - R\widetilde{\mathfrak{T}}^k b, \ k \geq 2. \tag{52}$$

Thus, by taking norms, we get

$$\|e^k\| \leq \|\widetilde{T}\|^k \|e^0\| + \|R\|\|b\| \left(c_k + \|\widetilde{\mathfrak{T}}\|^k \right), \tag{53}$$

where by $\{c_k\}_{k\geq 2}$ we denoted the sequence of positive numbers defined by

$$c_k = \sum_{j=1}^{k-1} \| \ \widetilde{T} \ \|^{k-j} \| \ \widetilde{\mathfrak{T}} \ \|^j, \ k \geq 2 \ . \tag{54}$$

Let now $\delta \in (0, 1)$ be defined by

$$\delta = \max\{\| \ \widetilde{T} \ \|, \| \ \widetilde{\mathfrak{T}} \ \|\} \ . \tag{55}$$

Then, from (53) and (54) we obtain

$$\| \ e^k \ \| \leq \delta^k \left(\| \ e^0 \ \| + k \cdot \|R\|\|b\| \right) \ \rightarrow \ 0, \quad \text{as } k \rightarrow \infty \ , \tag{56}$$

which completes the proof. □

Remark 4 *From the above theorem it results that, in the general case for* (2) *any of its least squares solutions can be obtained as a limit point in* (48)*, for an appropriate value of* x^0.

4 Constrained Cimmino algorithm

As we already mentioned, additional information about the real-world problem can further restrict the solution set of both (1) and (2). A common idea in image reconstruction is to look for solutions satisfying $x \geq 0$ or $x \in [l, u] \subset \mathbb{R}^n_+$. Therefore the iterates generated by ART are usually projected onto the range within the values of the components of an acceptable image vector must lie, see [3]. These interval constraining techniques were generalized by Koltracht and Lancaster in [9], were the authors made a complete theoretical analysis of a more general class of constraining strategies applied to (classical) Kaczmarz projection method, for consistent systems of equations. In paper [9] the authors consider a constraining function $C : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ with a closed image $\mathcal{I}(C) \subset \mathbb{R}^n$ and the properties

$$\|C(x) - C(y)\| \leq \|x - y\| \ , \tag{57}$$

$$\text{if } \ \|C(x) - C(y)\| = \|x - y\| \ \text{ then } \ C(x) - C(y) = x - y \ , \tag{58}$$

$$\text{if } \ y \in \mathcal{I}(C) \ \text{ then } \ y = C(y) \ , \tag{59}$$

and proposed the constrained version of Kaczmarz's algorithm:
Algorithm Constrained Kaczmarz (CK). Let $x^0 \in \mathcal{I}(C)$;
for $k = 0, 1, \ldots$ do

$$x^{k+1} = C(Qx^k + Ub) \ . \tag{60}$$

Here we stress that Kaczmarz's algorithm can be written in the form

$$x^{k+1} = Qx^k + Ub \ , \tag{61}$$

for some matrices $Q \in \mathbb{R}^{n \times n}$ and $U \in \mathbb{R}^{n \times m}$, compare also [17], similar to Cimmino's algorithm, see (13).

According to (60) the constrained version of Cimmino's algorithm (9) is the following.
Algorithm Constrained Cimmino (CC). Let $x^0 \in \mathcal{I}(C)$;
for $k = 0, 1, \ldots$ do

$$x^{k+1} = C(Tx^k + Rb). \tag{62}$$

Now, by following exactly the same way from [9, Th. 3], we can show the following convergence result for the CC algorithm (62).

Theorem 4 *Let us suppose that the matrix A satisfies* (5) *and* (11)*, the constraining function C satisfies* (57) – (59) *and the set $\mathcal{V}$ defined by*

$$\mathcal{V} = \{y \in \mathcal{I}(C),\ y - \Delta \in LSS(A,b)\} \tag{63}$$

is nonempty, where Δ is defined by

$$\Delta = (I - \widetilde{T})^{-1} R P_{\mathcal{N}(A^T)}(b), \tag{64}$$

with $\widetilde{T}, T, R$ from (21)*,* (14)*,* (15)*, respectively. Then, for any $x^0 \in \mathcal{I}(C)$ the sequence $\{x^k\}$ generated by* (62) *converges and its limit belongs to the set $\mathcal{V}$. If the problem* (1) *is consistent, then the above limit is one of its constrained solutions.*

Remark 5 *First we have to observe that all the assumptions* (57) – (59) *are essential in the proof of the above theorem, see the comments of the authors in [9, p. 562]. Second, if the problem* (1) *is consistent then*

$$\Delta = 0 \ \text{ and } \ \mathcal{V} = LSS(A,b) \cap \mathcal{I}(C), \tag{65}$$

i.e. in this case, the algorithm CC generates a "constrained" solution of the problem (1)*.*

The following two results are essential for the proof of Theorem 4 from above, compare [9].

Theorem 5 *Suppose that the assumptions of Theorem 2 hold. Let $\{x^k\}$ be the sequence generated by Cimmino's algorithm* (9)*, then*

$$\lim_{k\to\infty} x^k = P_{\mathcal{N}(A)}(x^0) + x_{LS} + \Delta\ , \tag{66}$$

with Δ given in (64)*.*

Proof. According to the decomposition $b = P_{\mathcal{R}(A)}(b) + P_{\mathcal{N}(A^T)}(b)$ we obtain in view of (24) exactly

$$\lim_{k\to\infty} x^k = P_{\mathcal{N}(A)}(x^0) + \underbrace{\left(I - \widetilde{T}\right)^{-1} R P_{\mathcal{R}(A)}(b)}_{=x_{LS}} + \underbrace{\left(I - \widetilde{T}\right)^{-1} R P_{N(A^T)}(b)}_{=\Delta}. \tag{67}$$

□

In the rest of this section we shall suppose that the set $\mathcal{V}$ is nonempty. The following counterpart of Lemma 1 from [9] shows that the image of a vector in $\mathcal{I}(C)$ through the map $\cdot \mapsto C(T(\,\cdot\,) + Rb)$, with C satisfying (57) – (59), can only be closer to $\mathcal{V}$ than this vector itself.

Lemma 6 *If the constraining function C satisfies all the assumptions* (57) – (59)*, $h \in \mathcal{I}(C)$ and g is given by*

$$g = C(Th + Rb) \ , \tag{68}$$

with T, R from (14), (15) *then, for any $y \in \mathcal{V}$ we have*

$$\|g - y\| \leq \|h - y\| \ . \tag{69}$$

Moreover, either

$$\|g - y\| < \|h - y\| \tag{70}$$

or

$$g = h \in \mathcal{V} \ . \tag{71}$$

Proof. Here we follow the proof in [9, Lem. 1]. Let $y \in \mathcal{V}$ and $h \in \mathcal{I}(C)$. In order to prove the inequality (69), we first prove

$$Rb = (I - T)y. \tag{72}$$

Since $y \in \mathcal{V}$ it exists a $\zeta \in LSS(A, b)$ such that

$$y = \zeta + \Delta \ . \tag{73}$$

Further, we denote by ξ the limit of the sequence from (66), for the initial approximation $x^0 = 0$, thus by Theorem 5 we also have

$$\xi = x_{LS} + \Delta. \tag{74}$$

Combining (73) and (74) we obtain

$$y - \xi = \zeta - x_{LS} \in \mathcal{N}(A) \tag{75}$$

and by Lemma 1 (i)

$$(I - T)(y - \xi) = 0 \ . \tag{76}$$

On the other hand, by taking limits in (13) we get

$$(I-T)\xi = Rb \; . \tag{77}$$

Now equation (72) follows from (77) and (76).

For any $h \in \mathcal{I}(C)$ we successively obtain

$$\begin{aligned} g - y &\overset{y\in\mathcal{I}(C)}{=} C(Th+Rb) - C(y) \\ &\overset{(72)}{=} C(Th+(I-T)y) - C(y) \\ &= C(T(h-y)+y) - C(y) \end{aligned} \tag{78}$$

and by taking norms

$$\begin{aligned} \|g-y\| &= \|C(T(h-y)+y) - C(y)\| \\ &\overset{(57)}{\leq} \|T(h-y)+y) - y\| = \|T(h-y)\| \\ &\overset{\|T\|\leq 1}{\leq} \|h-y\| \; , \end{aligned} \tag{79}$$

thus (69) holds.

Let us suppose now that (70) doesn't hold, thus, according to (69) we get

$$\|g-y\| = \|h-y\| \; . \tag{80}$$

Then, in (79) we have a sequence of equalities and thus, according to (58)

$$C(T(h-y)+y) - C(y) = T(h-y)$$

or, using (78)

$$g - y = T(h-y) \; . \tag{81}$$

From (80) and (81) we then obtain

$$\|T(h-y)\| = \|h-y\| \; . \tag{82}$$

By Lemma 2 (iv) $h-y \in N(A)$ and using Lemma 1 we get $T(h-y) = h-y$, which implies

$$g = h \tag{83}$$

in view of (81). Thus, for completely proving (71) we must show that $h \in \mathcal{V}$ or equivalently (compare (63))

$$h - \Delta \in LSS(A,b) \; . \tag{84}$$

Indeed,

$$h - \Delta = \underbrace{y - \Delta}_{\in LSS(A,b)} + \underbrace{h - y}_{\in \mathcal{N}(A)} \in LSS(A, b) \tag{85}$$

which completes the proof. □

Remark 6 *For the particular choice*

$$C(x) = \Pi_{\mathcal{B}}(x) := \text{argmin}\{\|x - y\| \mid y \in \mathcal{B}\} \ ,$$

with $\mathcal{B}$ a closed convex set Cimmino Constrained algorithm is a projected gradient method defined by $x^{k+1} = \Pi_{\mathcal{B}}(x^k - t_k \nabla f_D(x^k))$, with constant stepsize $t_k = \frac{2}{\omega}$ and f_D defined in (31)*.*

5 Constrained Cimmino Extended algorithm

In this section we will investigate the constrained version of Cimmino Extended algorithm (40) – (42), which is constructed analogously to the constrained version of Kaczmarz Extended algorithm (CKE) (33) – (35), proposed by one of the authors in [16]. This will be briefly described in what follows.

Algorithm Constrained Kaczmarz Extended (CKE). Let $x^0 \in \mathcal{I}(C), y^0 = b$; for $k = 0, 1, \dots,$ do

$$\begin{aligned} y^{k+1} &= \Phi(y^k) \ , && (86)\\ b^{k+1} &= b - y^{k+1} \ , && (87)\\ x^{k+1} &= C(Qx^k + Ub^{k+1}) \ . && (88) \end{aligned}$$

Let Γ be the set defined by

$$\Gamma = LSS(A, b) \cap \mathcal{I}(C). \tag{89}$$

The following convergence result was proved in [16] for the algorithm CKE.

Theorem 6 *If A satisfies* (5)*, the set Γ from* (89) *is nonempty and the constraining function C satisfies the assumptions* (57) *and* (59)*, then for any $x^0 \in \mathbb{R}^n$ the sequence $\{x^k\}$ generated with the algorithm CKE* (86)- (88) *converges to an element of the set Γ.*

Remark 7 *As we have already observed in in the Introduction, the assumption* (58) *on the constraining function C doesn't appear in the hypothesis of the above Theorem 6. This, together with* (57) *is called "strict nonexpansivity" property for C and is essentially used in the proof of Theorem 4 and [9, Th. 3].*

By analogy with CKE, we shall consider in what follows the *Constrained Cimmino Extended (CCE)* algorithm.
Algorithm Constrained Cimmino Extended (CCE). Let $x^0 \in \mathcal{I}(C)$, $y^0 = b$; for $k = 0, 1, \ldots,$ do

$$y^{k+1} = \mathfrak{T}(y^k) \ , \tag{90}$$

$$b^{k+1} = b - y^{k+1} \ , \tag{91}$$

$$x^{k+1} = C(Tx^k + Rb^{k+1}) \ . \tag{92}$$

The proof of Theorem 6 is based on the Lemmas 2 – 5 from [16]. These lemmas follow from the properties of the matrices Q, U and Φ, which were proved also for the corresponding operators T, R and $\mathfrak{T}$, respectively. Thus, we claim that similar results analogous to the above mentioned lemmas hold also for the CCE algorithm. Then, the following convergence result (similar with Theorem 6) will be true for the CCE algorithm.

Theorem 7 *If the matrix A satisfies* (5) *and* (11)*, the set Γ from* (89) *is nonempty and the constraining function C satisfies the assumptions* (57) *and* (59)*, then for any $x^0 \in \mathbb{R}^n$ the sequence $\{x^k\}$ generated by the algorithm CCE* (90) – (92) *converges to an element of the set Γ.*

Remark 8 *The assumption that the set Γ from* (89) *is nonempty is directly connected with the perturbation of b, which makes the system inconsistent. If we do not have anymore least squares solution in the set $\mathcal{I}(C)$, we conjecture that the CCE algorithm still converges, but to a vector at a certain distance from $LSS(A, b)$, see also next section.*

6 Numerical experiments

In this section we illustrate the numerical performance of the proposed methods in the context of image reconstruction from limited-data. In particular we focus on two examples of tomographic inversion problems inspired by real-world applications.

6.1 Test Data

6.1.1 Transmission tomography.

First we describe a discretized model for an image-reconstruction problem; it is the simplest *algebraic image reconstruction* model [5] for X-ray transmission tomography. It assumes that the cross section of an object represented by an image consists of an array of unknowns, and then sets up algebraic equations for the unknowns in terms of the measured projection data. To this end, a Cartesian grid of square picture elements (pixels) is introduced into the region of interest so that it covers the whole image that has to be reconstructed. The pixels are numbered in some agreed manner, say from 1 to n (see Fig. 1, left). The image, here the X-ray attenuation function, is assumed to take a constant value x_i throughout the j-th pixel for $j = 1, 2, \ldots, n$. Source S_i and detector D_i are assumed to be points and the rays between them lines. Further, assume that the length of intersection of the i-th ray with the j-th pixel, denoted by a_{ij} for all $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$ represents the weight of the contribution of the j-th pixel to the total attenuation along the i-th ray. The physical measurement of the total attenuation of the i-th ray, denoted by b_i, represents approximatively (due to measurement errors) the line integral of the unknown attenuation function along the path of the ray. Therefore, in this discretized model, the line integral turns out to be a finite sum and the whole model is described by a system of linear equations

$$\sum_{j=1}^{n} a_{ij} x_j \approx b_i \quad i = 1, 2, \ldots, m \ .$$

Our first set of numerical experiments are concerned with the two images $x^{ex,p} \in \mathbb{R}^{16}$ (4×4 pixels), $p \in \{1, 2\}$, from Fig. 1, middle and right. The equally spaced sources S_i and detectors D_i are arranged according to a parallel beam geometry from angles $0^o, 45^o, 90^o$, as depicted in Fig. 1, left, resulting in $m = 15$ measurements. The corresponding right hand side $b \in \mathbb{R}^{15}$ was computed for according to

$$b = Ax^{ex,p} + e \ , \qquad (93)$$

for some error vector $e \in \mathbb{R}^m$, which we will define in Section 6.2 and $p \in \{1, 2\}$. It must be pointed out that in clinical *Computerized Tomography*

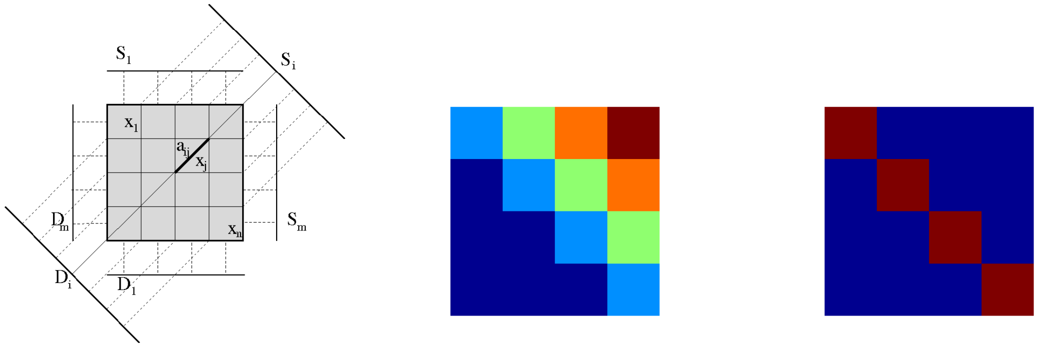


Figure 1: **Left to:** Scanning procedure in transmission tomography according to a parallel beam geometry. **Middle:** First test image $x^{ex,1}$. **Right:** Second test image $x^{ex,2}$.

(CT) transmission profiles of a patient are acquired under a large range of angles, i.e. the image to be reconstructed is highly oversampled. Still, there are situations when substantially fewer projections are sufficient to obtain a good enough reconstruction of the object under consideration, due to a simpler structure of this image. Typical examples are angiography or geophysical imaging. A further example is presented in what follows.

6.1.2 Tomographic Particle Image Reconstruction

Our second model problem has its origin in tomographic particle image velocimetry (TomoPIV) [6], a measurement technique which provides the basis for estimating turbulent flows and related flow patterns through image processing directly in 3D. The essential step of this technique is the 3D reconstruction of an particle distribution within the fluid from few projections (2D images). It is assumed that the image I to be reconstructed can be approximated by a linear combination of *basis functions* $\mathcal{B}_j$,

$$I(z) \approx \sum_{j=1}^{n} x_j \mathcal{B}_j(z), \quad \forall z \in \Omega \subset \mathbb{R}^3\,, \tag{94}$$

where Ω denotes the volume of interest. The main task is to estimate the weights x_j from the recorded 2D images, corresponding to basis functions located at a Cartesian equidistant 3D grid $p_j\,,\ j = 1, \dots, n$. We consider Gaussian-type basis functions ("blobs"), an alternative to the classical voxels, of the form

$$\mathcal{B}_j(z) = e^{-\frac{\|z-p_j\|_2^2}{2\sigma^2}}, \quad \text{for } z \in \mathbb{R}^3 : \|z - p_j\|_2 \leq r \,, \tag{95}$$

or value 0, if $\|z - p_j\|_2 > r$. The recorded pixel intensity b_i is the object intensity integrated along the corresponding line of sight L_i, obtained from a calibration procedure. Similar to the previous example the i-th measurement obeys

$$b_i :\approx \int_{L_i} I(z)dz \approx \int_{L_i} \hat{I}(z)dz = \sum_{j=1}^{n} x_j \underbrace{\int_{L_i} \mathcal{B}_j(z)dz}_{:=a_{ij}} \;. \tag{96}$$

In order to enable visualization we further present a 2D model and stress that 3D models are direct extensions of the present one. We consider 3 and 20 particles in a 2D volume $V = [-\frac{1}{2}, \frac{1}{2}] \times [-\frac{1}{2}, \frac{1}{2}]$, see Fig. 6.1.2, middle and right. The grid refinement was chosen $d = 0.0154$, resulting in 4356 gridpoints. At these gridpoints we center a Gaussian-type basis function, where $\sigma = d$. Particle positions were chosen randomly but at grid positions, to avoid discretization errors. Thus, $x^{ex,p}$, $p = \{3, 4\}$, is a binary vector in $\mathbb{R}^{4356}$ having 3 or 20 nonzero components. Four 50−pixel cameras are measuring the 2D volume from angles $45^o, 15^o, -15^o, -45^o$, according to a fan beam geometry, see Fig. 2, left. The screen and focal length of each camera is 0.5. The pixel intensities in the measurement vector b are computed according to (96), integrating the particle image exactly along each line of sight and perturbing the result according to

$$b = Ax^{ex,p} + e \ ,$$

where $p \in \{3, 4\}$ and e from (97) in Section 6.2.

6.2 General Considerations

We applied the algorithms Cimmino (C) (9) from Section 2, Constrained Cimmino (CC) (62) from Section 4, Cimmino Extended (CE) (40) – (42) from Section 3 and Constrained Cimmino Extended (CCE) (90) – (92) from Section 5 to the perturbed system

$$b = Ax^{ex,p} + e \ ,$$

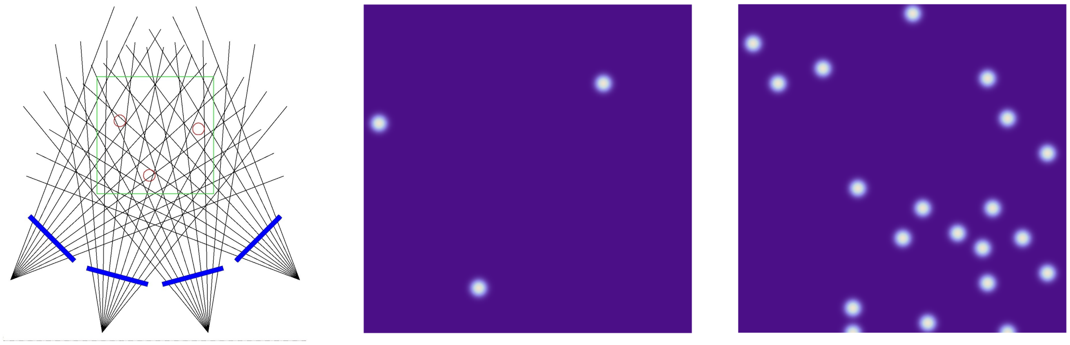


Figure 2: **Left:** Four cameras measuring the 2D volume from angles $45^o, 15^o, -15^o, -45^o$. **Middle:** The original image $x^{ex,3}$ contains 3 particles. **Right:** The original image $x^{ex,4}$ contains 20 particles.

where $p \in 1, 2, 3, 4$ and A and $x^{ex,p}$ obtained as detailed in Section 6.1. The error vector $e = e(\varepsilon) \in \mathbb{R}^m$ is defined by

$$e(\varepsilon) := \varepsilon \frac{v}{\|v\|} \|b\| \ , \tag{97}$$

where the components of v are chosen at random drawn from a uniform distribution on the unit interval. We have chosen four different values for ε, i.e. $\varepsilon \in \{0, 0.05, 0.1, 0.15\}$. The bigger is ε, the bigger will be

$$\|\Delta\| = \|GP_{\mathcal{N}(A^T)}(b)\|,$$

see (64) and we obtain an inconsistent least squares problem $Ax = b$.

The constraining function used in all computations was the orthogonal projection onto the box $[0, 1]^n$, i.e. $C : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ defined by

$$[C(x)]_i := \begin{cases} x_i, & x_i \in [0, 1] \\ 0, & x_i < 0 \\ 1, & x_i > 1 \ . \end{cases} \tag{98}$$

All original images x^{ex} lie in $\mathcal{I}(C) = [0, 1]^n$. Moreover, the original images $x^{ex,2}$, $x^{ex,3}$ and $x^{ex,4}$ are the unique solutions of $Ax = b$, for $e = 0$, which also lie in the set $[0, 1]^n$. Thus, $\mathcal{V}$ from (63) and Γ from (89) consists of only one point for $\varepsilon = 0$ and Cimmino Constrained and Cimmino Constrained

Extended algorithm will converge according to Theorem 4 and Theorem 7 to $x^{ex,p}$ in the noiseless (and consistent) case.

Our uniqueness claim is based on the following observations. If $b_i = 0$, then we can remove all columns of A, whose i-th entry is positive, as well as the i-th row. This procedure will lead to a "equivalent" feasible set of reduced dimensionality, see [10, Prop. 2.1] $\mathcal{F}_r := \{x \in \mathbb{R}^{n_r} : A_r x = b_r, x \geq 0\}$, where $A_r \in \mathbb{R}^{m_r \times n_r}$ is the reduced projection matrix and $b_r > 0$ the new data vector. All x_j variables corresponding to removed columns in A can be set to zero.

As a preprocessing step we reduce system $Ax = Ax^{ex,p}$ according to the above methodology for the two TomoPIV examples. The reduced dimensionalities for all considered examples are summarized in Tab. 1. Indeed, $x^{ex,2}$, and $x^{ex,3}$ respectively, is the unique positive solution of $Ax = Ax^{ex}$ since A_r is a full rank and overdetermined matrix. The uniqueness claim for $x^{ex,4}$ is due to the sparsity of $x^{ex,4}$, see e.g. [10].

In all computations we used $x^0 = 0$ as an initial approximation and terminating if the relative error at the current iterate x^k or the difference between two subsequent iterations is small enough, i.e.

$$\frac{\|x^k - x^{ex}\|}{\|x^{ex}\|} < 10^{-3} \quad \text{or} \quad \|x^k - x^{k-1}\| < 10^{-8} n$$

or if the maximum iteration number is reached, i.e. $k \geq k_{max}$, where $k_{max} = 10^6$. Moreover the implementation of Cimmino's algorithm uses an additional termination criterion which verifies if the relative residual of the weighted normal equations at the current iterate is small enough,

$$\frac{\|A^T D^2 (Ax^k - b)\|}{\|A^T D^2 b\|} < 10^{-6} , \tag{99}$$

whereas CE algorithm employs

$$\frac{\|A^T (Ax^k - b)\|}{\|A^T b\|} < 10^{-6} . \tag{100}$$

Different termination criteria we used for the constrained algorithms Cimmino Constrained and CCE. Additionally to the above mentioned criteria we test if

$$K(x^k) = \|\min(x^k, \max(x^k - 1, -\nabla f(x^k))\|_\infty < 10^{-6} \tag{101}$$

for $f = f_D$ from (31) for CC, and $f(x) = \frac{1}{2}\|Ax - b\|^2$ for CCE algorithm. Note that $K(x) = 0$ iff x is a solution of

$$\min f(x) \quad \text{s.t. } x \in [0,1]^n$$

since f and $[0,1]^n$ are convex.

The weights ω_i in (8) and α_i in (37) equal 1 in all computations.

Table 1: Dimensions of the reduced matrix A_r

Example	m	n	m_r	n_r	rank(A_r)	x^{ex} unique?
1	15	16	15	16	12	no
2	15	16	12	8	8	yes
3	200	4356	41	7	7	yes
4	200	4356	128	1901	128	yes

6.3 Results

Here we summarize the results obtained by the proposed constrained and extended versions of Cimmino's algorithm, for both consistent noiseless and inconsistent noisy case. Table 2 shows the results for the first considered example $x^{ex,1}$, whereas the reconstructed images are presented in Fig. 3 when no constraints are taken into account and in Fig. 4 the constrained reconstructions. The corresponding results for the second example $x^{ex,2}$ are presented in Table 2, Fig. 3 and Fig. 4 respectively. Since $x^{ex,3}$ corresponding to the 3 particles example is the unique solution of $A_r x = b_r$ in the noise-free case, we applied to this test example Cimmino and CE. No constraints are necessary. We stress that 3 particles correspond to a particle density of 0.06 pp (particles/pixel) that is currently in use in TomoPIV. To obtain reconstructions of denser particle distributions constraining strategies should be used. Therefore we applied CC and CCE to the 20 particles example. Results are summarized in Table 4 and reconstructed images are presented in Fig. 7 and 8. Although pictures speak for themselves several remarks are in order.

Cimmino and CE algorithm outperforms CC and CCE in terms of speed (i.e. # iterations). Constraining leads, as expected, to an improved reconstruction. This becomes evident especially when the original image $x^{ex,p}$ to be reconstructed is the unique solution of $Ax = Ax^{ex,p}$ in $[0,1]^n$, thus for

$p = \{2, 3, 4\}$. However, we note that in case of error measurements the systems $Ax = b$ (or $DAx = Db$) as well as $A^TAx = A^Tb$ (or $A^TD^2Ax = A^TD^2b$) do not have positive solutions. This findings we verified by using Farkas's lemma. For instance to verify that $Ax = b, x \geq 0$ has no solution we solved $A^Ty >= 0, b^Ty < 0$.

This situation is reflected also by the high value of $K(x^k)$ at the final iterate.

In Fig. 8 we present some preliminary results obtained by the *Bregman Iterative Algorithm* in [19], especially designed to find the minimum ℓ_1-norm solution of a linear system. In theory (at least for $e = 0$) this method will converge to the same solution as CC and CCE, namely to $x^{ex,4}$. The reconstructions differ significantly due to the slow convergence of CC and CCE. The iteration process was stopped since the difference between two subsequent iterates become small enough and the algorithms started to stagnate. Since constraining seem to decelerate C and CE in the underdetermined case, an interesting question would be how to choose the starting point x^0 to obtain the minimal ℓ_1-norm (least squares) solution of $Ax = b$ directly.

Table 2: Results of Cimmino, CE, CC and CCE applied to $x^{ex,1}$

	Cimmino	CE	CC	CCE
ε	0	0	0	0
#Iter.	180	327	675	771
$\frac{\|x^k-x^{ex}\|}{\|x^{ex}\|}$	0.100000	0.100000	0.022530	0.008987
$\frac{\|Ax^k-b\|}{\|b\|}$	0.000002	0.000003	0.000044	0.000011
$\frac{\|A^T(Ax^k-b)\|}{\|A^Tb\|}$	0.000001	0.000001	0.000028	0.000007
$\frac{\|A^TD^2(Ax^k-b)\|}{\|A^TD^2b\|}$	0.000001	–	0.000025	–
$\frac{\|Ax^k-b^k)\|}{\|b^k\|}$	–	0.000002	–	0.000011
$K(x^k)$	–	–	0.000100	0.000100
$\|x^k - x^{k-1}\|$	0.000001	0.000000	0.000004	0.000001
ε	0.05	0.05	0.05	0.05
#Iter.	174	313	745	860
$\frac{\|x^k-x^{ex}\|}{\|x^{ex}\|}$	0.122644	0.118048	0.076295	0.066922
$\frac{\|Ax^k-b\|}{\|b\|}$	0.011153	0.010267	0.011150	0.010267
$\frac{\|A^T(Ax^k-b)\|}{\|A^Tb\|}$	0.002623	0.000001	0.002632	0.000007

Table 2: Results of Cimmino, CE, CC and CCE applied to $x^{ex,1}$ (continued)

	Cimmino	CE	CC	CCE
$\frac{\Vert A^T D^2(Ax^k-b)\Vert}{\Vert A^T D^2 b\Vert}$	0.000001	–	0.000024	–
$\frac{\Vert Ax^k-b^k)\Vert}{\Vert b^k\Vert}$	–	0.000002	–	0.000010
$K(x^k)$	–	–	0.000099	0.000100
$\Vert x^k - x^{k-1}\Vert$	0.000001	0.000000	0.000004	0.000001
ε	0.10	0.10	0.10	0.10
#Iter.	156	214	777	898
$\frac{\Vert x^k-x^{ex}\Vert}{\Vert x^{ex}\Vert}$	0.239630	0.216823	0.234172	0.205273
$\frac{\Vert Ax^k-b\Vert}{\Vert b\Vert}$	0.031943	0.029404	0.031940	0.029404
$\frac{\Vert A^T(Ax^k-b)\Vert}{\Vert A^T b\Vert}$	0.007431	0.000001	0.007440	0.000006
$\frac{\Vert A^T D^2(Ax^k-b)\Vert}{\Vert A^T D^2 b\Vert}$	0.000001	–	0.000022	–
$\frac{\Vert Ax^k-b^k)\Vert}{\Vert b^k\Vert}$	–	0.000002	–	0.000010
$K(x^k)$	–	–	0.000100	0.000100
$\Vert x^k - x^{k-1}\Vert$	0.000001	0.000001	0.000004	0.000001
ε	0.15	0.15	0.15	0. 15
#Iter.	162	324	802	644
$\frac{\Vert x^k-x^{ex}\Vert}{\Vert x^{ex}\Vert}$	0.464404	0.412956	0.487774	0.423684
$\frac{\Vert Ax^k-b\Vert}{\Vert b\Vert}$	0.059689	0.054945	0.059686	0.054945
$\frac{\Vert A^T(Ax^k-b)\Vert}{\Vert A^T b\Vert}$	0.013755	0.000001	0.013762	0.000005
$\frac{\Vert A^T D^2(Ax^k-b)\Vert}{\Vert A^T D^2 b\Vert}$	0.000001	–	0.000019	–
$\frac{\Vert Ax^k-b^k)\Vert}{\Vert b^k\Vert}$	–	0.000002	–	0.000009
$K(x^k)$	–	–	0.000099	0.000100
$\Vert x^k - x^{k-1}\Vert$	0.000001	0.000001	0.000004	0.000001

Table 3: Results of Cimmino, CE, CC and CCE applied to $x^{ex,2}$

	Cimmino	CE	CC	CCE
ε	0	0	0	0
#Iter.	201	397	531	582
$\frac{\Vert x^k-x^{ex}\Vert}{\Vert x^{ex}\Vert}$	0.500000	0.500000	0.000997	0.000995

Table 3: Results of Cimmino, CE, CC and CCE applied to $x^{ex,2}$ (continued)

	Cimmino	CE	CC	CCE
$\frac{\|Ax^k-b\|}{\|b\|}$	0.000002	0.000003	0.000315	0.000314
$\frac{\|A^T(Ax^k-b)\|}{\|A^Tb\|}$	0.000001	0.000001	0.000211	0.000211
$\frac{\|A^TD^2(Ax^k-b)\|}{\|A^TD^2b\|}$	0.000001	–	0.000154	–
$\frac{\|Ax^k-b^k)\|}{\|b^k\|}$	–	0.000002	–	0.000314
$K(x^k)$	–	–	0.000504	0.002029
$\|x^k-x^{k-1}\|$	0.000000	0.000002	0.000024	0.000024
ε	0.05	0.05	0.05	0.05
#Iter.	202	396	435	491
$\frac{\|x^k-x^{ex}\|}{\|x^{ex}\|}$	0.502039	0.501592	0.036469	0.035431
$\frac{\|Ax^k-b\|}{\|b\|}$	0.011041	0.010164	0.021668	0.021738
$\frac{\|A^T(Ax^k-b)\|}{\|A^Tb\|}$	0.002678	0.000001	0.007075	0.007568
$\frac{\|A^TD^2(Ax^k-b)\|}{\|A^TD^2b\|}$	0.000001	–	0.021090	–
$\frac{\|Ax^k-b^k)\|}{\|b^k\|}$	–	0.000002	– 0.019217	
$K(x^k)$	–	–	0.072239	0.080728
$\|x^k-x^{k-1}\|$	0.000000	0.000000	0.000000	0.000000
ε	0.10	0.10	0.10	0.10
#Iter.	202	396	467	460
$\frac{\|x^k-x^{ex}\|}{\|x^{ex}\|}$	0.519121	0.514985	0.112168	0.109400
$\frac{\|Ax^k-b\|}{\|b\|}$	0.031295	0.028808	0.061413	0.061613
$\frac{\|A^T(Ax^k-b)\|}{\|A^Tb\|}$	0.007569	0.000001	0.020003 0.021395	
$\frac{\|A^TD^2(Ax^k-b)\|}{\|A^TD^2b\|}$	0.000001	–	0.059326	–
$\frac{\|Ax^k-b^k)\|}{\|b^k\|}$	–	0.000002	–	0.054486
$K(x^k)$	–	–	0.223072	0.249289
$\|x^k-x^{k-1}\|$	0.000001	0.000000	0.000000	0.000000
ε	0.15	0.15	0.15	0.15
#Iter.	202	395	439	414
$\frac{\|x^k-x^{ex}\|}{\|x^{ex}\|}$	0.579955	0.563380	0.236114	0.230287
$\frac{\|Ax^k-b\|}{\|b\|}$	0.057917	0.053314	0.113657	0.114027
$\frac{\|A^T(Ax^k-b)\|}{\|A^Tb\|}$	0.013987	0.000001	0.036966	0.039539

Table 3: Results of Cimmino, CE, CC and CCE applied to $x^{ex,2}$ (continued)

	Cimmino	CE	CC	CCE
$\frac{\|A^T D^2(Ax^k-b)\|}{\|A^T D^2 b\|}$	0.000001	–	0.108889	–
$\frac{\|Ax^k-b^k)\|}{\|b^k\|}$	–	0.000002	–	0.100938
$K(x^k)$	–	–	0.469567	0.524754
$\|x^k - x^{k-1}\|$	0.000001	0.000000	0.000000	0.000000

Table 4: Results of Cimmino, CE, CC and CCE applied to $x^{ex,3}$ and $x^{ex,4}$

	3 Particles		20 Particles	
	Cimmino	CE	CC	CCE
ε	0	0	0	0
#Iter.	162	655	29684	30919
$\frac{\|x^k-x^{ex}\|}{\|x^{ex}\|}$	0.000980	0.000997	0.765976	0.836548
$\frac{\|Ax^k-b\|}{\|b\|}$	0.000200	0.000176	0.022421	0.035621
$\frac{\|A^T(Ax^k-b)\|}{\|A^T b\|}$	0.000122	0.000038	0.012632	0.019959
$\frac{\|A^T D^2(Ax^k-b)\|}{\|A^T D^2 b\|}$	0.000086	–	0.009629	–
$\frac{\|Ax^k-b^k)\|}{\|b^k\|}$	–	0.000051	–	0.034496
$K(x^k)$	–	–	0.003712	0.184839
$\|x^k - x^{k-1}\|$	0.000065	0.000017	0.000019	0.000019
ε	0.05	0.05	0.05	0.05
#Iter.	253	1011	40328	37978
$\frac{\|x^k-x^{ex}\|}{\|x^{ex}\|}$	1.349164	0.055816	1.043464	0.844548
$\frac{\|Ax^k-b\|}{\|b\|}$	0.794428	0.033868	0.600412	0.032052
$\frac{\|A^T(Ax^k-b)\|}{\|A^T b\|}$	0.830303	0.000001	0.291489	0.015632
$\frac{\|A^T D^2(Ax^k-b)\|}{\|A^T D^2 b\|}$	0.000001	–	0.391864	–
$\frac{\|Ax^k-b^k)\|}{\|b^k\|}$	–	0.000001	–	0.027487
$K(x^k)$	–	–	1.000000	0.140784
$\|x^k - x^{k-1}\|$	0.000001	0.000000	0.000019	0.000019
ε	0.10	0.10	0.10	0.10
#Iter.	285	976	32469	117714

Table 4: Results of Cimmino, CE, CC and CCE applied to $x^{ex,3}$ and $x^{ex,4}$ (continued)

	3 Particles		20 Particles	
	Cimmino	CE	CC	CCE
$\frac{\|x^k-x^{ex}\|}{\|x^{ex}\|}$	4.134589	0.171002	1.108025	0.909299
$\frac{\|Ax^k-b\|}{\|b\|}$	2.278256	0.097125	0.568035	0.091394
$\frac{\|A^T(Ax^k-b)\|}{\|A^Tb\|}$	2.371626	0.000001	0.280415	0.039024
$\frac{\|A^TD^2(Ax^k-b)\|}{\|A^TD^2b\|}$	0.000001	–	0.721760	
$\frac{\|Ax^k-b^k)\|}{\|b^k\|}$	–	0.000001	–	0.084547
$K(x^k)$	–	–	1.000000	0.677917
$\|x^k-x^{k-1}\|$	0.000002	0.000000	0.000019	0.000019
ε	0.15	0.15	0.15	0.15
#Iter.	292	891	39376	150386
$\frac{\|x^k-x^{ex}\|}{\|x^{ex}\|}$	8.599377	0.355635	1.262440	1.082098
$\frac{\|Ax^k-b\|}{\|b\|}$	4.262568	0.181720	0.572720	0.217518
$\frac{\|A^T(Ax^k-b)\|}{\|A^Tb\|}$	4.446508	0.000001	0.279046	0.098684
$\frac{\|A^TD^2(Ax^k-b)\|}{\|A^TD^2b\|}$	0.000001	–	0.856854	–
$\frac{\|Ax^k-b^k)\|}{\|b^k\|}$	–	0.000001	–	0.208584
$K(x^k)$	–	–	1.000000	1.000000
$\|x^k-x^{k-1}\|$	0.000004	0.000001	0.000019	0.000019

7 Conclusions

We presented two extensions and a corresponding convergence analysis of the classical Cimmino algorithm for iteratively computing constrained solutions to inconsistent least-squares solutions. Taking into account previous related work of one author on Kaczmarz iterations, this completes the picture of row-action iterations with respect to parallel version of the overall method.

Synthetical numerical experiments simulating a challenging real-world application revealed a significantly different impact of the two extensions, however. While removal of inconsistency converges quite slowly - but comparably to the state of the art adopted in the respective application area, see [10] -

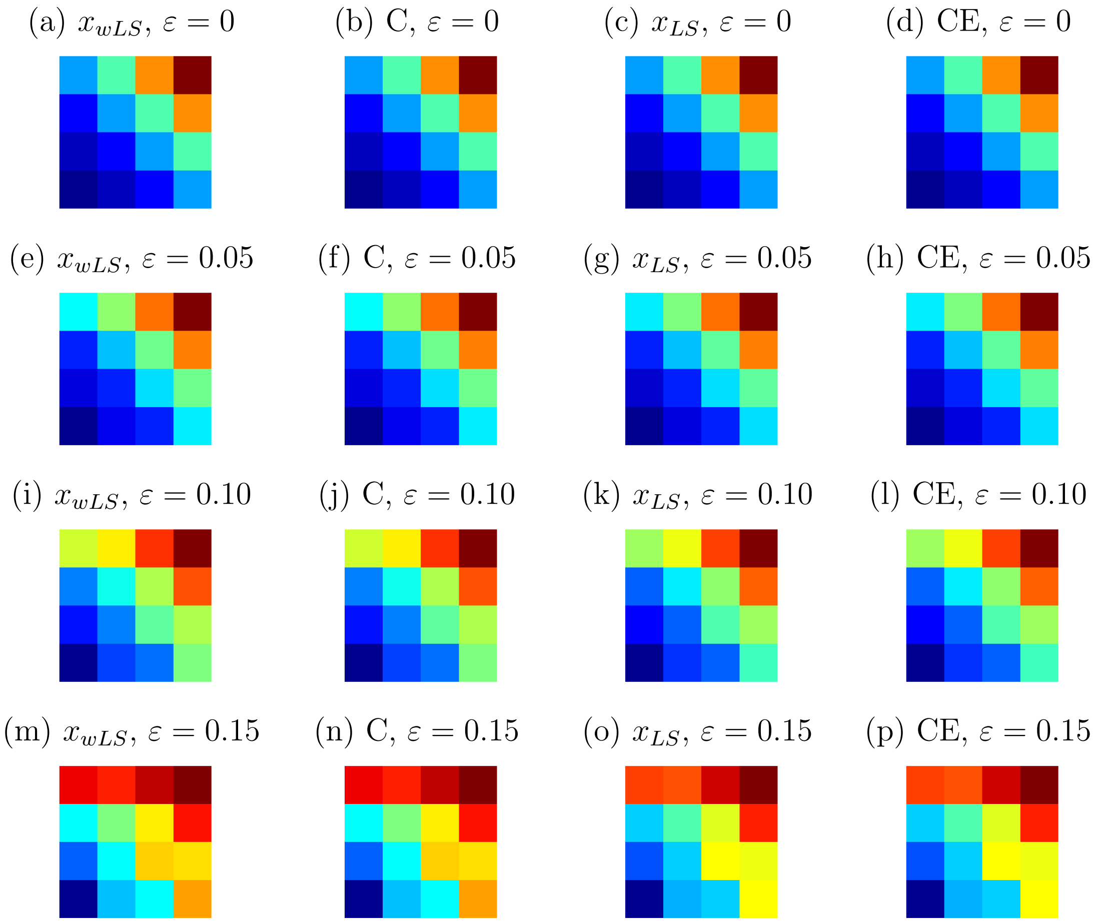


Figure 3: Unconstrained reconstruction results for image $x^{ex,1}$: (a),(e),(i),(m): Minimal norm solution x_{wLS} of the weighted least squares problem (4) obtained via the Moore-Penrose pseudoinverse for the four level of perturbation, i.e. $\varepsilon \in \{0, 0.05, 0.10, 0.15\}$. (b),(f),(j),(n): Reconstruction using Cimmino's (C) algorithm for different perturbation levels. (c),(g),(k),(o): Minimal norm solution x_{LS} of the least squares problem (2) obtained via the Moore-Penrose pseudoinverse for the four level of perturbation. (d),(h),(l),(p): Reconstruction using Cimmino Extended (CE) algorithm.

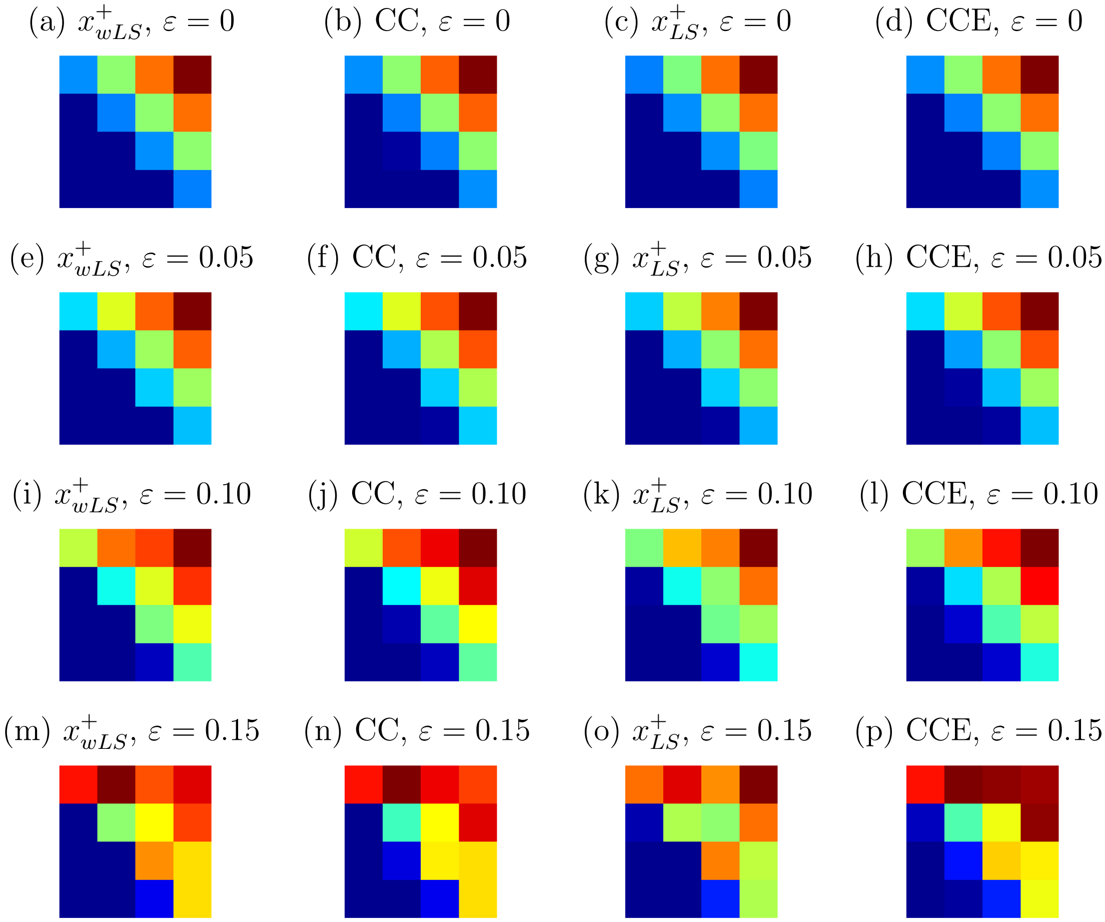
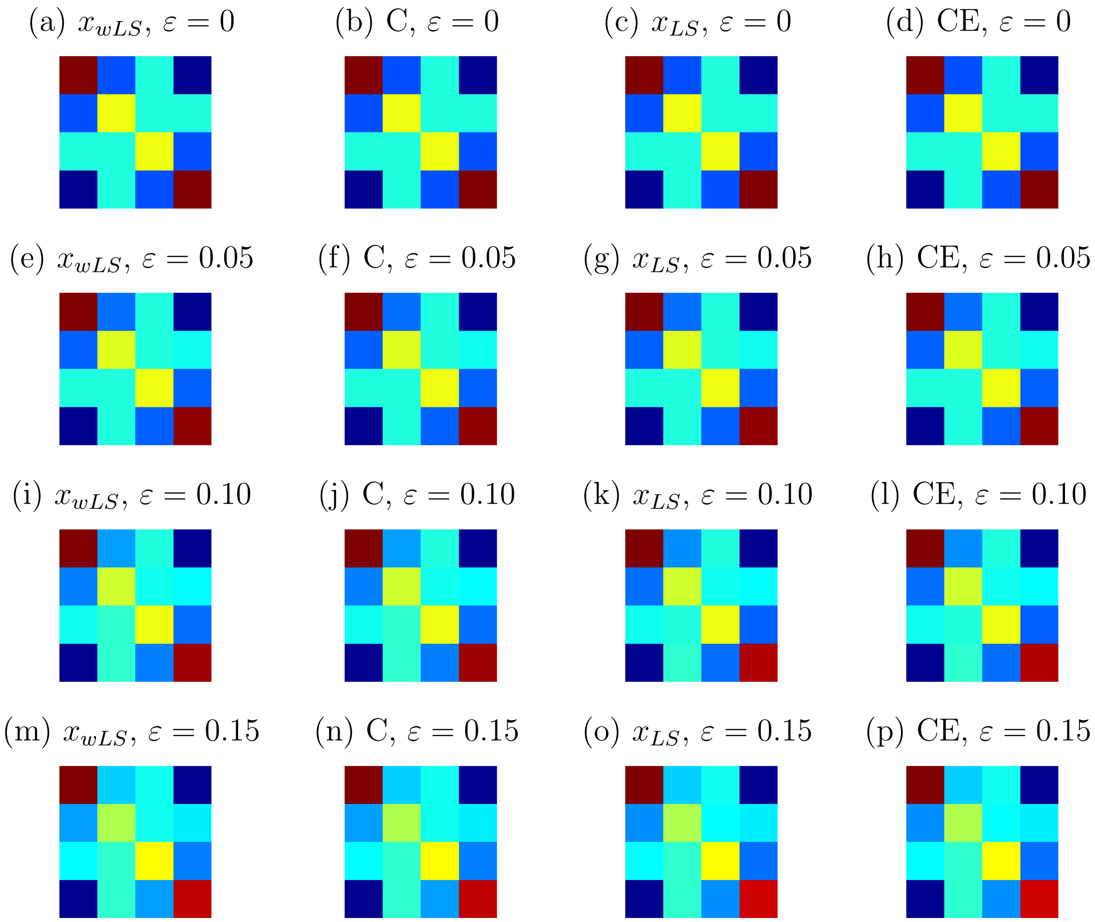


Figure 4: Constrained reconstruction results for image $x^{ex,1}$: (a),(e),(i),(m): The nonnegative minimal norm solution x^+_{wLS} of the weighted least squares problem (4) obtained via the MATLAB solver `lsqnonneg(D*A,D*b)` for the four level of perturbation, i.e. $\varepsilon \in \{0, 0.05, 0.10, 0.15\}$. (b),(f),(j),(n): Reconstruction using Cimmino Constrained (CC) algorithm for different perturbation levels. (c),(g),(k),(o): The nonnegative minimal norm solution x^+_{LS} of the least squares problem (2) obtained via the MATLAB solver `lsqnonneg(A,b)` for the four level of perturbation. (d),(h),(l),(p): Reconstruction using Cimmino Constrained Extended (CCE) algorithm.



(a) x_{wLS}, $\varepsilon = 0$ (b) C, $\varepsilon = 0$ (c) x_{LS}, $\varepsilon = 0$ (d) CE, $\varepsilon = 0$

(e) x_{wLS}, $\varepsilon = 0.05$ (f) C, $\varepsilon = 0.05$ (g) x_{LS}, $\varepsilon = 0.05$ (h) CE, $\varepsilon = 0.05$

(i) x_{wLS}, $\varepsilon = 0.10$ (j) C, $\varepsilon = 0.10$ (k) x_{LS}, $\varepsilon = 0.10$ (l) CE, $\varepsilon = 0.10$

(m) x_{wLS}, $\varepsilon = 0.15$ (n) C, $\varepsilon = 0.15$ (o) x_{LS}, $\varepsilon = 0.15$ (p) CE, $\varepsilon = 0.15$

Figure 5: Unconstrained reconstruction results for image $x^{ex,2}$: (a),(e),(i),(m): Minimal norm solution x_{wLS} of the weighted least squares problem (4) obtained via the Moore-Penrose pseudoinverse for the four level of perturbation, i.e. $\varepsilon \in \{0, 0.05, 0.10, 0.15\}$. (b),(f),(j),(n): Reconstruction using Cimmino's (C) algorithm for different perturbation levels. (c),(g),(k),(o): Minimal norm solution x_{LS} of the least squares problem (2) obtained via the Moore-Penrose pseudoinverse for the four level of perturbation. (d),(h),(l),(p): Reconstruction using Cimmino Extended (CE) algorithm.

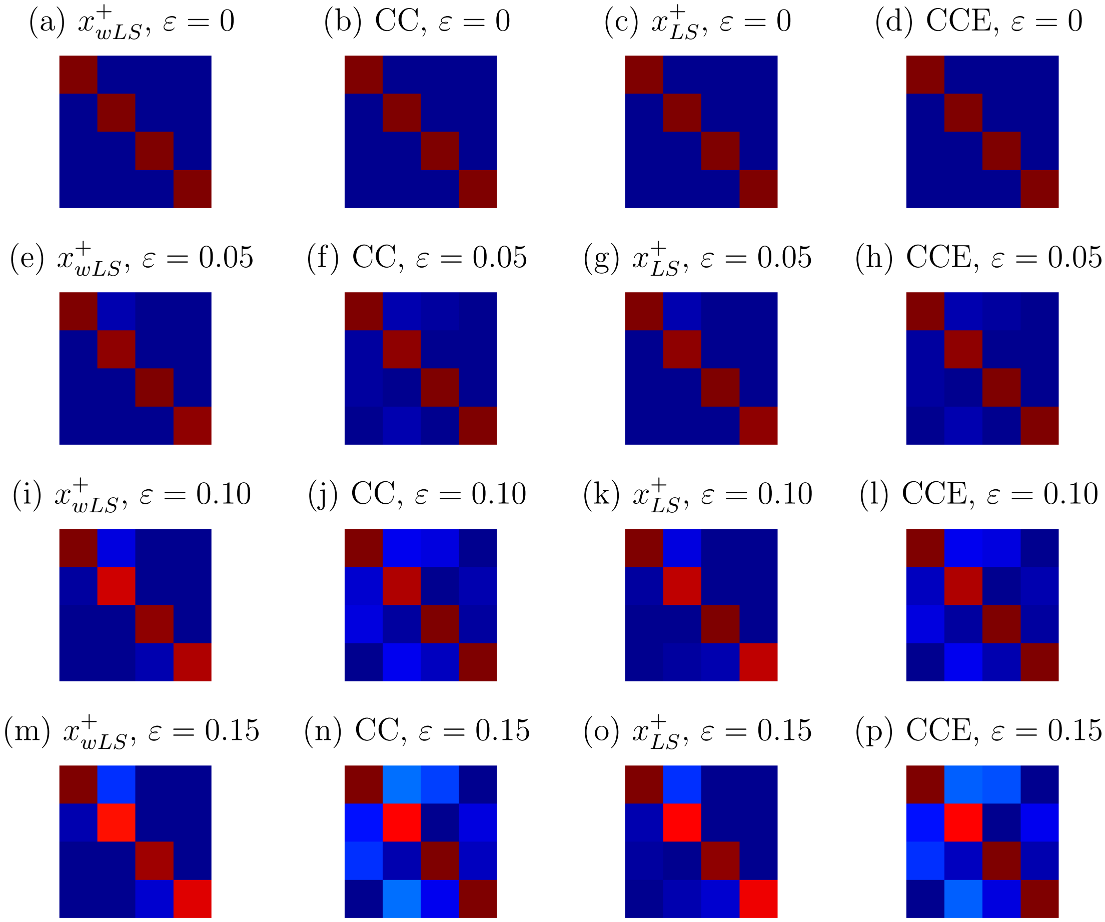


Figure 6: Constrained reconstruction results for image $x^{ex,2}$: (a),(e),(i),(m): The nonnegative minimal norm solution x^+_{wLS} of the weighted least squares problem (4) obtained via the MATLAB solver `lsqnonneg(D*A,D*b)` for the four level of perturbation, i.e. $\varepsilon \in \{0, 0.05, 0.10, 0.15\}$. (b),(f),(j),(n): Reconstruction using Cimmino Constrained (CC) algorithm for different perturbation levels. (c),(g),(k),(o): The nonnegative minimal norm solution x^+_{LS} of the least squares problem (2) obtained via the MATLAB solver `lsqnonneg(A,b)` for the four level of perturbation. (d),(h),(l),(p): Reconstruction using Cimmino Constrained Extended (CCE) algorithm.

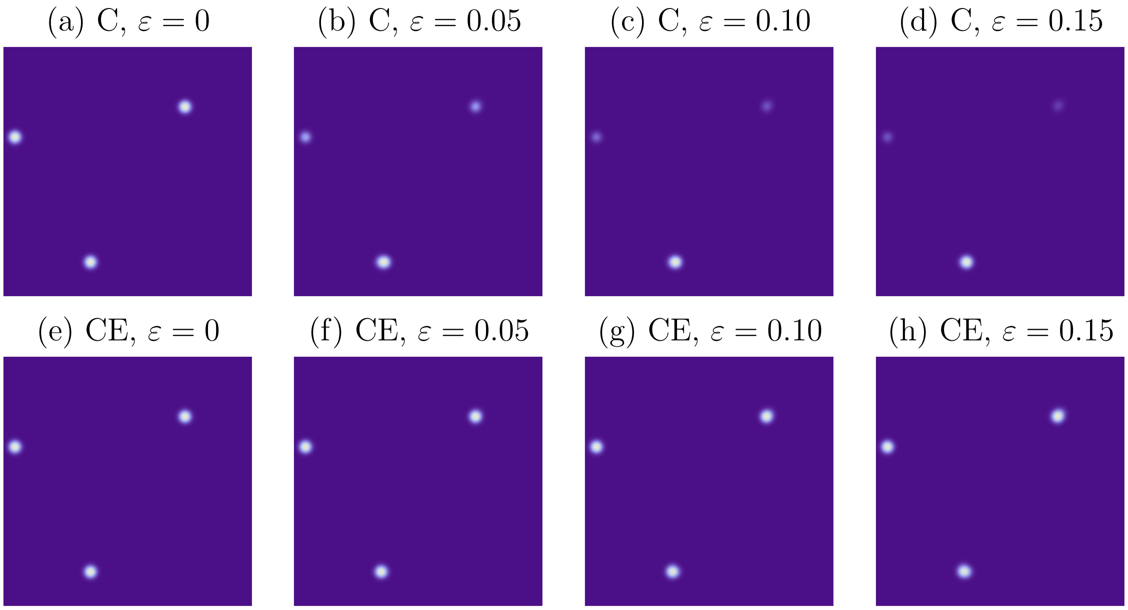


Figure 7: Unconstrained reconstruction results for the 3 particles image $x^{ex,3}$: **Top:** (a) – (d): Reconstruction using Cimmino's (C) algorithm for the four level of perturbation, i.e. $\varepsilon \in \{0, 0.05, 0.10, 0.15\}$. **Bottom:** (e) – (h): Reconstruction using Cimmino Extended (CE) algorithm for different perturbation levels.

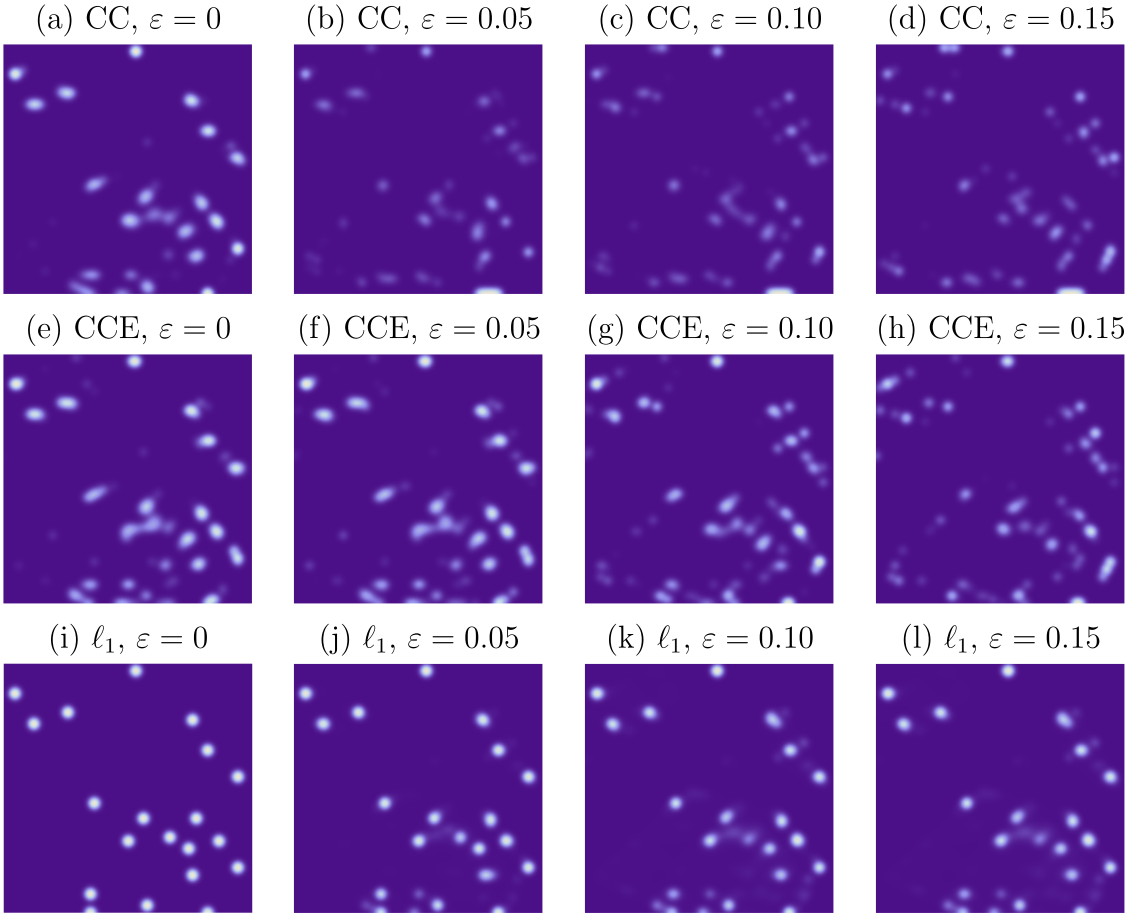


Figure 8: Constrained reconstruction results for the 20 particles image $x^{ex,4}$: **Top** (a) – (d): Reconstruction using Cimmino's (C) algorithm for the four level of perturbation, i.e. $\varepsilon \in \{0, 0.05, 0.10, 0.15\}$. **Middle** (e) – (h): Reconstruction using Cimmino Extended (CE) algorithm for different perturbation levels. **Bottom** (i) – (l) ℓ_1 reconstruction obtained via the Bregman Iterative algorithm in [19].

constraining appears to be far more effective. As a consequence, suitable constraints along with consistently modifying the objective criterion (norm) will be investigated in our future work.

Acknowledgements

The authors[1,3] gratefully acknowledge financial support from the German National Science Foundation under grant SCHN 457/10-1.

References

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

[2] Boullion T. L., Odell P. L. *Generalized inverse matrices*, Willey - Interscience, New York,1971.

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

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

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

[6] G. Elsinga, F. Scarano, B. Wieneke,B. van Oudheusden: Tomographic particle image velocimetry. Exp. Fluids, **41** (2006), pp. 933 – 947.

[7] Elsner L., Koltracht I. and Lancaster P., *Convergence properties of ART and SOR algorithms*, Numer. Math. , **59** (1991), pp. 91 – 106.

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

[9] Koltracht I. and Lancaster P., *Constraining strategies for linear iterative processes*, IMA Journal of Numerical Analysis, **10** (1990), pp. 555 – 567.

[10] Petra S., Schnörr C., Schröder A., Wieneke B., *Tomographic Image Reconstruction in Experimental Fluid Dynamics: Synopsis and Problems.* to appear in Proceedings of 6th Workshop on Math. Modeling Environ. Life & Sci. Problems (2007/2008).

[11] Petra S., Schröder A., Wieneke B., Schnörr C., *On Sparsity Maximization in Tomographic Particle Image Reconstruction.* In Pattern Recognition – 30th DAGM Symposium, volume 5096 of LNCS, 2008. Springer Verlag.

[12] Popa C., *Least-squares solution of overdetermined inconsistent linear systems using Kaczmarz's relaxation*; Intern. J. Comp. Math., **55** (1995), pp. 79 – 89.

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

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

[15] Popa C., *On Cimmino's reflection algorithm*, Proceedings of the Romanian Academy, Series A, **9**, January-April 2008.

[16] Popa C., *Constrained Kaczmarz extended algorithm for image reconstruction*, Linear Algebra and its Applications, **429** (2008), pp. 2247 – 2267.

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

[18] Saad Y. , *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2nd rev. and exp. Ed., 2003.

[19] Yin. W., Osher S., Goldfarb D., Darbon J., *Bregman Iterative Algorithms for l1-Minimization with Applications to Compressed Sensing*, SIAM J. Imaging Sciences, **1** (2008), pp. 143-168.