Applied Analysis and Optimization
Volume 2, Number 2, 2018, 219–236

Yokohama Publishers
ISSN 2189-1664 Online Journal

ROW AND COLUMN BASED ITERATIONS

TOMMY ELFVING

Abstract. We survey old and more recent results on row- and column-action iterative methods for solving ill-conditioned linear systems. Our main application is in X-ray tomography problems with missing and/or noisy data. We consider the stationary case with cyclic control. A unified framework is presented the use of which allows deriving both necessary and sufficient convergence conditions for many of the methods presented.

1. Introduction

Among tomographic reconstruction algorithms the class of algebraic iterative methods is useful in situations with few (to avoid excessive use of radiation) and/or noisy data. Censor [14] coined the expression row-action methods for a specific class of algebraic iterative methods. This class includes Kaczmarz's algorithm [48] which was independently suggested, under the name ART, in [41] where it was used for the first time in the open literature to solve tomographic reconstruction problems. Some recent applications include electromagnetic tomography [58] and proton imaging [55].

Although it has been known [9, 32] how to base the reconstruction algorithm on columns rather than on rows this possibility has not been explored much. An exception is Watt [66] who derives a column-based reconstruction method and compares it with ART (also using nonnegativity constraints). In [5] a two-parameter algorithm (including, e.g. the SOR-method) based on column partitionings is studied. In [36] also algorithms based on column partitionings are considered. In particular two techniques for saving computational work by not performing small updates (typical for solution elements that have converged) are proposed and investigated.

Here we will survey block row- and column-iterations. We will also present, for the stationary case, some unified convergence results. The stationary case means keeping a fixed relaxation parameter for each block of rows (or columns). Necessary (to our knowledge most of this is new) and sufficient bounds on these parameters to guarantee convergence will be given. Apart from theoretical interest this knowledge can be helpful when implementing the 'optimal fixed parameter' strategy. Then one searches for that constant parameter value which, within a fixed number of iterations, gives the smallest relative error. Here one may use some appropriate training samples see [43, 63]. For some other choices of relaxation parameters see [43].

2010 *Mathematics Subject Classification.* 65F10, 65R32.
Key words and phrases. Algebraic iterative reconstruction, Block-iteration, ART, Kaczmarz, Cimmino, Convergence.

Row and column methods seek to solve the minimum norm problem, and the least squares problem respectively. For inconsistent data their asymptotic behavior is therefore different. The row-action methods exhibit cyclic convergence but not in general to a least squares solution. The column methods on the other hand converge to a least squares solution but not in general to the minimum norm solution. In tomographic reconstructions it is not uncommon that there are errors in the implementation of the forward projector, and/or the backprojector [67]. Hence the matrices that represent these operators are not each other's transpose. The influence of such errors were recently studied [33], both on the two underlying minimization problems, and on the behavior of row- and column iterations used to solve these problems. The connections between row-action methods and multiplicative Schwarz methods are considered in [54]. Here also convergence rates are studied both for random and cyclic control.

We will here only consider cyclic control, i.e. the blocks of rows/columns respectively are picked up in a cyclic order. For other controls see [1, 24, 19, 23], and for randomized Kaczmarz's methods see [64, 30, 52, 54].

Notation. Let Q be a matrix, then $Q^\dagger$ is its pseudoinverse, and $R(Q), N(Q)$ denote the rangespace and nullspace of Q respectively. Further P_Ω is the orthogonal projector onto a closed convex set Ω. For square matrices we let $\rho(Q)$ be the spectral radius of Q. The inner product of two vectors is denoted (x, y) with the corresponding norm $\|x\| = \|x\|_2 = \sqrt{(x,x)}$. Further spd stands for symmetric and positive definite, $Q^{1/2}$ for the square root of a spd matrix Q, and $\|x\|_Q^2 = x^T Q x$.

2. Block-Row Iteration

Our starting point will be a large linear system of equations (not necessarily consistent),

$$Ax = b,\ A \in \mathbb{R}^{m\times n},\ b \in \mathbb{R}^m. \tag{2.1}$$

The system is assumed to arise from discretization of an ill-posed problem. We assume that the matrix A does not contain rows/columns identically equal zero. Let A be partitioned into p disjoint block rows and let b be partitioned accordingly,

$$A = \begin{pmatrix} R_1 \\ \vdots \\ R_p \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_p \end{pmatrix}, \quad R_i \in \mathbb{R}^{m_i\times n},\ b_i \in \mathbb{R}^{m_i},\ i = 1, 2, \ldots, p.$$

Also, let $\{\omega_i\}_{i=1}^p$ be a set of positive relaxation parameters and let $M_i \in \mathbb{R}^{m_i\times m_i}$, $i = 1, 2, \ldots, p$ be a set of given spd matrices.

The following generic algorithm, which uses the blocks R_i in a *sequential* fashion, covers several important special cases.

Algorithm BRI: Block-Row Iteration

****

Initialization: $x^0 \in \mathbb{R}^n$ is arbitrary.

For $k = 0, 1, 2, \ldots$ (cycles or outer iterations)

$v^0 = x^k$

For $i = 1, 2, \ldots, p$ (inner loop)

$$v^i = v^{i-1} + \omega_i \, R_i^T M_i \big(b_i - R_i \, v^{i-1}\big)$$

End

$x^{k+1} = v^p$

End

****

With $p = 1$ there is just one block so $M_i = M \in \mathbf{R}^{m \times m}$, and the method becomes fully simultaneous. On the other end when $p = m$ each block consists of a single row so $M_i \in \mathbf{R}, \; i = 1, 2, \ldots, m$, and the iteration becomes fully sequential. For a study of implementation/performance issues of block-iterative iterations on multicore architectures see [62].

Let a *cycle* denote one pass through all blocks, i.e., one outer iteration. Since block iteration uses a single block in each inner iteration it takes p iterations to complete a cycle.

Let $slt(Q)$ denote the strictly lower block-triangular part of a matrix Q. Define

$$L_r = slt(AA^T) = \begin{pmatrix} O & & & O \\ R_2 R_1^T & \ddots & & \\ \vdots & \ddots & \ddots & \\ R_p R_1^T & \ldots & R_p R_{p-1}^T & O \end{pmatrix}, \tag{2.2}$$

and

$$D_r = \begin{pmatrix} \omega_1^{-1} M_1^{-1} & & & O \\ & \ddots & & \\ & & \ddots & \\ O & & & \omega_p^{-1} M_p^{-1} \end{pmatrix} := diag(\omega_i^{-1} M_i^{-1}). \tag{2.3}$$

Further put

$$M_{rw} = (D_r + L_r)^{-1}. \tag{2.4}$$

Proposition 2.1. *One cycle of Algorithm BRI can be written*

$$x^{k+1} = x^k + A^T M_{rw} (b - Ax^k). \tag{2.5}$$

Proof. [34, Proposition 4]. □

We stress that (2.5) holds for the particular subsequence $\{x^k\}$ generated in Algorithm BRI (i.e. using the blockrow-ordering $(1, 2, \ldots, p)$). For other subsequences the matrix M_{rw} will change. We next study how it is changed. First let $P_{ij} A$ interchange block rows i and j in A. The entries of P_{ij} are 0 or 1. In each row and column of P_{ij} there is exactly one 1. It's easily seen that $P_{ij}^T = P_{ij}^{-1}$, i.e. P_{ij} is orthogonal. The matrix P_{ij} is called an elementary permutation matrix see [45] for the case of block rows of dimension 1. We provide an example. Let $p = 2$ and

$R_1, 3 \times n,\ R_2, 2 \times n$. Then

$$P_{12} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}, \text{ so that } P_{12}A = \begin{pmatrix} R_2 \\ R_1 \end{pmatrix}.$$

Let $(i_1, i_2, \ldots, i_p)$ denote an arbitray subsequence. Obviously such a sequence can be generated by Algorithm BRI by considering a permuted system $PAx = Pb$ with P a product of elementary permutation matrices. Let $\hat{A} = PA$, and $\hat{L}_r = slt(\hat{A}\hat{A}^T), \hat{D}_r = PD_rP^T$. It follows

$$\hat{M}_{rw} = (slt(PAA^TP^T) + PD_rP^T)^{-1}.$$

Continuing the example above we get

$$\hat{M}_{rw} = \begin{pmatrix} \omega_2^{-1}M_2^{-1} & 0 \\ R_1R_2^T & \omega_1^{-1}M_1^{-1} \end{pmatrix}, \text{whereas } M_{rw} = \begin{pmatrix} \omega_1^{-1}M_1^{-1} & 0 \\ R_2R_1^T & \omega_2^{-1}M_2^{-1} \end{pmatrix}.$$

Remark 2.2. The iterates are not sensitive to the ordering of the unknowns, unless such a transformation also changes M_i (to say $\bar{M}_i$) as can be seen as follows. Let P be any permutation matrix, and consider now the transformation (from the right) AP so that the system becomes $APP^Tx = b$. With $\bar{A} = AP,\ y = P^Tx$ we then get

$$\bar{A}y = b. \tag{2.6}$$

Note that $\bar{A}$ has blockrows equal $\{R_iP\}$. We now apply (2.5) to (2.6) and get

$$y^{k+1} = y^k + \bar{A}^T\bar{M}_{rw}(b - \bar{A}y^k).$$

After multiplication with P from the left, and putting $Py^k = x^k$ we retrieve (2.5), provided $M_{rw} = \bar{M}_{rw}$. Now $L_r = slt(AA^T) = slt(\bar{A}\bar{A}^T) = \bar{L}_r$. Hence $M_{rw} = (D_r + L_r)^{-1} = (\bar{D}_r + \bar{L}_r)^{-1} = \bar{M}_{rw}$ provided $M_i = \bar{M}_i,\ i = 1, 2, \ldots, p$. It follows then that the use of P only affects the order of the components in the iterates but not their numerical values.

We shall later see how M_i transforms for different examples. The iterates do however, even in the consistent case, depend on the order of the equations. In fact the bigger the angle between successive hyperplanes ($p = m$) the faster convergence.We will return to this in connection with the examples.

Theorem 2.3. *Assume that $b \in R(A)$. The iterates of Algorithm BRI converge towards a solution $\hat{x}$ of $Ax = b$ if*

$$\omega_i \in \Omega_i = (\epsilon, (2-\epsilon)/\rho(R_i^TM_iR_i)),\ i = 1, 2, \ldots, p, \quad 0 < \epsilon < 2. \tag{2.7}$$

If further

$$(i)\ N(A^T) = \emptyset \quad and/or\ (ii)\ R(M_{rw}) \subseteq R(A), \tag{2.8}$$

then (2.7) is also necessary for convergence. If, in addition $x^0 \in R(A^T)$, then $\hat{x}$ is unique and of minimal 2-norm.

The proof is given in Section 4. Sufficient convergence conditions are also given in [47, Theorem II.I] and [12, Theorem 3.1].

We next consider the inconsistent case.

Theorem 2.4. *If conditions (2.7) hold then the iterates in (2.5) converge towards a solution of*

$$A^T M_{rw} A x = A^T M_{rw} b. \tag{2.9}$$

Proof. The convergence of subsequences in the inconsistent case (also called cyclic convergence) was shown in [65] for Kaczmarz's method ($\omega_i = 1,\ p = m$). For general weight matrices see [29, Theorem 1.3] (and cf. [34, Proposition 6]) or [47, Theorem II.I]. The fact that the limit satisfies (2.9) follows from Proposition 2.1. □

We now consider the fully simultaneous case $p = 1$. Put $\omega_1 = \omega,\ M_1 = M$. Then Algorithm BRI becomes

$$x^{k+1} = x^k + \omega A^T M (b - A x^k). \tag{2.10}$$

Theorem 2.5. *The iterates of (2.10) converge towards a minimizer $\hat{x}$ of $\|b - Ax\|_M$ if and only if*

$$\omega \in (\epsilon, (2-\epsilon)/\rho(A^T M A)),\ 0 < \epsilon < 2. \tag{2.11}$$

If, in addition $x^0 \in R(A^T)$, then $\hat{x}$ is unique and of minimal 2-norm.

The proof is given in Section 4. The fully simultaneous iteration (2.10) (often referred to as Landweber iteration) has been analyzed and used frequently. In particular replacing $\omega A^T M$ in (2.10) by $\omega_k U^T A^T M$ with U, M both diagonal with positive diagonal elements, and $\omega_k \in (0, 1]$ [12, Theorem 4.1] provides sufficient conditions on the diagonal elements to insure convergence. For the case U, M spd and allowing the relaxation parameter ω to depend on k (nonstationary iteration) [59, Theorem IV.3] gives a necessary and sufficient condition for convergence. It is easily verified that this condition, using a constant relaxation parameter ω becomes identical to (2.11).

2.1. **Symmetric Block-Row Iteration.** In the symmetric version one first performs one cycle of Algorithm BRI followed by another cycle but now taking the row-blocks in reverse order.

Algorithm SBRI: Symmetric Block-Row Iteration

```
*********************************************
Initialization: x^0 ∈ R^n is arbitrary.
For k = 0, 1, 2, ... (cycles or outer iterations)
    v^0 = x^k
    For i = 1, 2, ..., 2p − 1, (inner loop)
        If i ≤ p then j(i) = i else j(i) = 2p − i
        v^i = v^{i−1} + ω_{j(i)} R^T_{j(i)} M_{j(i)} (b_{j(i)} − R_{j(i)} v^{i−1}),
    End
    x^{k+1} = v^{2p−1}
End
*********************************************
```

For the fully sequential case $p = m$ this method was considered in [9].

Let

$$M_{sr} = M_{rw}^T \tilde{D}_r M_{rw}, \quad \tilde{D}_r = 2D_r - \text{blockdiag}(AA^T). \tag{2.12}$$

Proposition 2.6. *One cycle of Algorithm SBRI can be written*

$$x^{k+1} = x^k + A^T M_{sr}(b - Ax^k). \tag{2.13}$$

If the conditions (2.7) hold then M_{sr} is positive definite.

Proof. [34, Proposition 10]. □

We again stress that M_{sr} depends on the row-ordering.

Theorem 2.7. *The iterates of Algorithm SBRI converge towards a solution of* $\min \|Ax - b\|_{M_{sr}}$ *if and only if the conditions (2.7) hold.*

The proof will be given in Section 4. In the consistent case it follows that the limit point satisfies $Ax = b$. In the inconsistent case we will have cyclic convergence since the limit point depends on M_{sr}, i.e. on the row-ordering.

We summarize the convergence results in a table as follows (where N means necessary condition and S sufficient condition, and $\hat{x}$ is the limit point).

Algorithm	Restriction	cond (2.7)	source	limit
BRI	$b \in R(A)$	S.	Theorem 2.3	$A\hat{x} = b$
BRI	$b \in R(A)$, and (2.8)	N. and S.	Theorem 2.3	$A\hat{x} = b$
BRI	$b \notin R(A)$	S.	Theorem 2.4	cyclic convergence
BRI	$p = 1$ (simultaneous)	N. and S.	Theorem 2.5	$\arg\min \|b - A\hat{x}\|_M$
SBRI	$b \in R(A)$	N. and S.	Theorem 2.7	$A\hat{x} = b$
SBRI	$b \notin R(A)$	N. and S.	Theorem 2.7	cyclic convergence

2.2. **Examples: Row-Iteration.** Several well known iterative methods appear as special cases of Algorithm BRI.

Example 1 (Block-Kaczmarz)**.** Then $M_i = (R_i R_i^T)^{-1}$, so that $\bar{M}_i = M_i$, cf. Remark 2.2. Here it is tacitly assumed that R_i has full row-rank. The block version of Kaczmarz's method was studied in [31] (later published in [32]) without this restriction. There the method is derived by applying the classical SOR-method on the equations $AA^T y = b,\ x = A^T y$. We next simplify the convergence conditions by first observing that $R_i^T M_i R_i = P_{R(R_i^T)}$. It follows that $\rho(R_i^T M_i R_i) = 1$, and hence condition (2.7) in this case is equivalent with $\omega_i \in (0, 2),\ i = 1, 2, \ldots, p$. The classical row version appears for $p = m$. We also note that the inner loop can be written

$$v^i = \omega_i P_{\{R_i z = b_i\}} v^{i-1} + (1 - \omega_i) v^{i-1},$$

and hence can be seen as an instance of the projection onto convex sets (POCS) algorithm [24]. Kaczmarz's method [48] has a long and rich history see, e.g. [44, 51, 24, 65] ($\omega_i = 1,\ p = m$), [32, Theorem 2] (block-case), [26, 46] among others. In [50] the convergence, taking all relaxation parameters equal one, was studied in an infinite dimensional Hilbert space setting. For results on more general classes of projection algorithms see [6, 24, 38, 11, 13].

As remarked above the rate of convergence depends on the angles between hyperplanes. In [44, p. 209] a strategy is suggested (for matrices appearing in image reconstruction) for sorting the rows in an efficient way. A similar strategy is suggested in [40]. A different approach for picking the order of the equations is based on the fact that the vectors $v^i - v^{i-1}$ and $v^i - x$ are perpendicular when $\omega_i = 1$, and x is a solution of $Ax = b$. Thus

$$\|v^i - x\|^2 = \|v^{i-1} - x\|^2 - \|v^i - v^{i-1}\|^2.$$

Hence picking i such that $\|v^i - v^{i-1}\| = \|R_i^T M_i\big(b_i - R_i v^{i-1}\big)\|$, is maximal gives the largest possible decrease of the error in step i. This would however require evaluating $b_j - R_j v^{i-1}$ over all j. A more efficient implementation (specially when p is large), requiring only evaluation over a subset is considered in [30]. Note however that this approach neither leads to a stationary iteration nor to cyclic control.

Example 2 (Block-Cimmino [25])**.** Let $\{\theta_j^i\}$ be given positive weights such that $\Sigma_{j=1}^{m_i} \theta_j^i = 1$. Further r_j^i denotes the jth row in R_i. Then

$$M_i = M_i^{Cim} = \mathrm{diag}(\theta_j^i / \|r_j^i\|^2), \ \ i = 1, 2, \dots, p.$$

Since A by assumption has no zero rows M_i^{Cim} is well defined. Taking equal weights it holds $M_i^{Cim} = 1/m_i(\,\mathrm{diag}(R_i^T R_i)\,)^{-1}$, so Cimmino can be considered as using a diagonal approximation of the corresponding matrix in Kaczmarz's method. It also follows that $\bar{M}_i = M_i$. It's easily seen that $\rho(R_i^T M_i R_i) \leq 1$ (see, e.g., [18, (7.27)] so that by Theorems 2.3 and 2.4 convergence occurs for $\omega_i \in (0, 2)$. However this is now only a sufficient condition even if conditions (2.8) are satisfied. In fact, as has been verified experimentally in, e.g. [20, Figures 4.7-9] the upper bound 2 is quite restrictive especially for large and sparse matrices so that taking too small value of ω_i results in poor rate of initial convergence.

Example 3 (Component averaging (CAV))**.** This method was introduced by Censor, Gordon and Gordon [21] to overcome the slow initial rate of Cimmino. Let s_ν^i be the number of nonzero elements in column ν in R_i. Then

$$M_i = diag\left(\frac{1}{\|r_j^i\|_{S_i}^2}\right), \ S_i = diag\left(s_\nu^i\right), \ i = 1, 2, \dots, p. \tag{2.14}$$

Since A has no zero-row it easily follows that $\|r_j^i\|_{S_i}^2 \neq 0$. In [18, Corollary 7.1] it is shown that $\rho(R_i^T M_i R_i) \leq 1$. Hence we can again deduce convergence when $\omega_i \in (0, 2)$. Although this is again only a sufficent condition it seems more tight (for sparse matrices) than the bound for Cimmino [20, Figures 4.7-9]. Further using that $\bar{r}_j^i = r_j^i P$ and $\bar{S}_i = P S_i P$ (note that a permutation matrix is always symmetric) it holds $\|r_j^i\|_{S_i}^2 = \|\bar{r}_j^i\|_{\bar{S}_i}^2$. Hence $M_i = \bar{M}_i$. Sufficient convergence conditions for CAV (both for strictly block and for fully simultaneous versions) are also given in [12].

In our two following examples the inner loop in Algorithm BRI is replaced by

$$v^i = v^{i-1} + \omega_i U R_i^T M_i (b_i - R_i v^{i-1}),$$

where $U \in \mathbf{R}^{n\times n}$ is a given spd matrix. By defining

$$\tilde{R}_i = R_i U^{1/2},\ u^i = U^{-1/2} v^i$$

the inner loop takes the form

$$u^i = u^{i-1} + \omega_i \tilde{R}_i^T \bar{M}_i (b_i - \tilde{R}_i u^{i-1}). \tag{2.15}$$

Hence using the matrix U corresponds to performing Algorithm BRI on the system $\tilde{A}z = b$ where $\tilde{A} = AU^{1/2},\ z = U^{-1/2}x$. Let $U.alg$ denote algorithm BRI on the form (2.15). If column permutations are done (which we denote $P \circ U.alg$) Remark 2.2 applies. If however the order is reversed ($U.alg \circ P$), i.e. the matrix U is chosen based on the permuted system the iterates will change (unless U remains the same for the permuted and the unpermuted system).

Example 4 (Diagonally-Relaxed orthogonal projection Methods (DROP))**.** Here we consider the version DROP1 [20, (3.10)]. Then

$$M_i = M_i^{Cim} = \bar{M}_i,\ i = 1, 2, \ldots, p,\ U = \mathrm{diag}(1/\tau_\nu), \tag{2.16}$$

where

$$\tau_\nu \geq \max_i \{s_\nu^i, 1 \leq i \leq p\},\ 1 \leq \nu \leq n.$$

Here it is not assumed that $\Sigma_{j=1}^{m_i} \theta_j^i = 1$. It holds [20, Theorem 2.10]

$$\rho(U^{1/2} R_i^T M_i R_i U^{1/2}) \leq \max_j \theta_j^i.$$

Hence picking $\theta_j^i = 1$ for all i, j we can again deduce convergence when $\omega_i \in (0, 2)$.

Both CAV and DROP were constructed to improve slow rate of convergence of the Cimmino method by explicitly allowing the iteration parameters to depend on sparsity. For a fully dense matrix the three methods coincide.

Example 5 (Simultaneous Algebraic Reconstruction Technique (SART))**.** This method was introduced by Andersen and Kak [2]. Here we consider the block-version BSSART proposed in [18, (7.15)]. Then (note that here the 1-norm is used)

$$U = diag(1/\|a_c^\ell\|_1),\ M_i = diag\left(\frac{1}{\|r_j^i\|_1}\right),\ i = 1, 2, \ldots, p. \tag{2.17}$$

Here a_c^ℓ is the ℓth column of A. Then following result is from [18, (7.18)]

$$\begin{aligned}\rho(U^{1/2} R_i^T M_i R_i U^{1/2}) &= \rho(R_i^T M_i R_i U) \\ &\leq \|R_i^T M_i\|_1 * \|R_i U\|_1 = 1.\end{aligned}$$

Hence we can again deduce convergence when $\omega_i \in (0, 2)$. Since $\|x\|_1 = \|xP\|_1$ it follows that $M_i = \bar{M}_i$. Note that U is well defined by the assumption that A does not contain zero columns.

Example 6 (Mixture)**.** Note that we can mix the first three examples, i.e., within each cycle pick different types of M_i. However the choice should be independent of the outer index k so that the resulting iteration still is stationary.

All six examples given above can also be used in Algorithm SBRI. Row-iteration methods can also be used for solving linear inequalities [27, 22, 18, 28, 20, 13, 15].

We finish this section by a few notes on inconsistency. As mentioned, see Theorem 2.4, the BRI-algorithm exhibits in general cyclic convergence when applied to an inconsistent system. In [17] however it was shown for Kaczmarz's method with $\omega_i = \omega$ that as $\omega \to 0$ the iterates converge towards a scaled least squares solution (a similar result is [34, Proposition 12] for algorithm SBRI). An interesting approach for handling inconsistency is due to Popa and coo-workers, e.g. [57, 58, 56]. A starting point here is to apply Kaczmarz's method on the augmented system [8, section 2.5.3]

$$\begin{pmatrix} I & A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} r \\ x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}.$$

This system is equivalent with the normal equations and therefore always consistent. Note that the method requires access to both rows and columns of A during a cycle.

3. Block-Column iteration

Let A be partitioned into q disjoint column-blocks $\{A_i\}_1^q$, where

$$A_i \in \mathbb{R}^{m \times n_i}, \ \sum_{i=1}^{q} n_i = n,$$

and let the vector x be partitioned similarly, i.e.

$$x = (x_1, x_2, \dots, x_q), \ x_i \in \mathbb{R}^{n_i}.$$

Let $\{\omega_i\}_{i=1}^q$ be a set of positive relaxation parameters, and $\{N_i \in \mathbb{R}^{n_i \times n_i}\}_{i=1}^q$ a set of given spd matrices.

Algorithm BCI: Block-Column Iteration
****

Initialization: $x^0 \in \mathbb{R}^n$ is arbitrary. $r^{0,1} = b - A\,x^0$.
For $k = 0, 1, 2, \dots$ (cycles or outer iterations)
  For $i = 1, 2, \dots, q$ (inner loop)
    $x_i^{k+1} = x_i^k + \omega_i N_i A_i^T r^{k,i}$
    $r^{k,i+1} = r^{k,i} - A_i(x_i^{k+1} - x_i^k)$
  End
  $r^{k+1,1} = r^{k,q+1}$
End

****

Hence for each cycle the method requires q applications of A_i, A_i^T respectively. It is easily seen that the update generating $r^{k,i+1}$ in the inner loop is an efficient way to compute the residual given by

$$r^{k,i+1} = b - \sum_{j=1}^{i} A_j x_j^{k+1} - \sum_{j=i+1}^{q} A_j x_j^k. \tag{3.1}$$

Let

$$(3.2) \qquad L_c = slt(A^T A) = \begin{pmatrix} O & & & O \\ A_2^T A_1 & \ddots & & \\ \vdots & \ddots & \ddots & \\ A_q^T A_1 & \dots & A_q^T A_{q-1} & O \end{pmatrix}, \; D_c = diag(\omega_i^{-1} N_i^{-1}).$$

Proposition 3.1. *Let* $M_{cl} = (D_c + L_c)^{-1}$. *One cycle of Algorithm CRI can be written*

$$(3.3) \qquad x^{k+1} = x^k + M_{cl} A^T (b - Ax^k).$$

Proof. [36, Proposition 4]. □

Theorem 3.2. *The iterates of Algorithm BCI converge toward a minimizer of* $\|b - Ax\|$ *if and only if*

$$(3.4) \qquad \omega_i \in \bar{\Omega}_i = \big(\epsilon, (2-\epsilon)/\rho(A_i N_i A_i^T)\big), \quad i = 1, 2, \dots, q, \quad 0 < \epsilon < 2.$$

Theorem 3.2 will be proved in section 4.

Remark 3.3. The iterates in Algorithm BCI are not sensitive to the ordering of the equations unless N_i also changes. To see this let again P be a permutation matrix, and consider a permutation of (2.1)

$$\bar{A}x = \bar{b}, \; \bar{A} = PA, \; \bar{b} = Pb.$$

By applying (3.3) on this system we get

$$x^{k+1} = x^k + \bar{M}_{cl} \bar{A}^T (\bar{b} - \bar{A}x^k) = x^k + \bar{M}_{cl} A^T (b - Ax^k).$$

Since $L_c = \bar{L}_c$ the iterates based on the permuted system will be the same as the iterates based on the original system provided $N_i = \bar{N}_i, \; i = 1, 2, \dots, q$. The rate of convergence however depends on the ordering of the unknowns.

3.1. **Symmetric Block-Column Iteration.** We finally present, for completeness, the symmetric version (which was, for $q = n$ also described in [9]).

Algorithm SBCI: Symmetric Block-Column Iteration

```
*********************************************
  Initialization: x^0 ∈ R^n is arbitrary. r^{0,1} = b − A x^0.
  For k = 0, 1, 2, ... (cycles or outer iterations)
     For i = 1, 2, ..., 2q − 1 (inner loop)
        if i ≤ q then
          z_i = x_i^k + ω_i N_i A_i^T r^{k,i}
          r^{k,i+1} = r^{k,i} − A_i (z_i − x_i^k)
        else j(i) = 2q − 1
          x_{j(i)}^{k+1} = z_{j(i)} + ω_{j(i)} N_{j(i)} A_{j(i)}^T r^{k,i}
          r^{k,i+1} = r^{k,i} − A_{j(i)} (x_{j(i)}^{k+1}) − z_{j(i)})
     End
     r^{k+1,1} = r^{k,2q}
  End
********************************************
```

3.2. **Examples: Column-Iteration.** Here we shortly describe the column-versions of the row-iterations presented in 2.2.

Example 1, SOR. Then

$$N_i = (A_i^T A_i)^{-1}, \tag{3.5}$$

so that $A_i N_i A_i^T = A_i A_i^\dagger = P_{R(A_i)}$. It follows that $\rho(A_i N_i A_i^T) = 1$, and hence by Theorem 3.2 the method converges if and only if $\omega_i \in (0,2)$. Here we assume that A_i has full column rank, otherwise see [32]. This method is matematically equivalent with applying the classical SOR method on the normal equations.

Example 2, Column-Cimmino. Let $\{\theta_j^i\}$ be given positive weights such that $\Sigma_{j=1}^{n_i} \theta_j^i = 1$. Further a_i^j denote the jth column of block A_i. Then

$$N_i = \mathrm{diag}(\theta_j^i / \|a_i^j\|^2),\ i = 1,2,\ldots,q.$$

Taking equal weights it holds $N_i = 1/n_i(\,\mathrm{diag}(A_t^T A_t)\,)^{-1}$, so Cimmino can be considered as using a diagonal approximation of the matrix from SOR. This can be useful when $A_i^T A_i$ is a full matrix (which occurs in certain models in computed tomography). We next investigate condition (3.4) (assuming for notational convenience equal weights). It holds

$$\begin{aligned} \rho(A_i N_i A_i^T) &= \|A_i N_i A_i^T\| = \frac{1}{n_i} \left\| \sum_{j=1}^{n_i} \frac{1}{\|a_i^j\|} a_i^j (a_i^j)^T \right\| \\ &= \frac{1}{n_i} \left\| \sum_{j=1}^{n_i} P_{\mathcal{R}(a_i^j)} \right\| \le 1 \quad \Rightarrow \quad \omega_i \in (0,2). \end{aligned} \tag{3.6}$$

Note however that now the upper bound 2 is only a sufficient condition (similarly as for row-iteration) and may lead to slow rate of convergence.

Example 3, Column-CAV. Let s_i^ν be the number of nonzero elements in row ν of A_i. Then

$$N_i = diag\left(\frac{1}{\|a_i^j\|_{S_i}^2}\right),\ S_i = diag\,(s_i^\nu)\,,\ i = 1,2,\ldots,q. \tag{3.7}$$

This defines the column version of the row-action method CAV [21] mentioned in the previous section. Similarly as for the row version [18, Corollary 7.1] one finds that $\rho(A_i N_i A_i^T) \le 1$ so convergence holds for $\omega_i \in (0,2)$.

In a similar way we can define column versions of DROP and SART, and also combine, within a cycle, Examples 1,2,3.

4. Convergence analysis

Consider the possibly singular but consistent linear system

$$Qx = d,\ d \in R(Q), \tag{4.1}$$

where $Q \in \mathbb{R}^{n\times n},\ d \in \mathbb{R}^n$. Let V be a given nonsingular matrix. We will study the following stationary iteration

$$x^{k+1} = Tx^k + c,\ T = I - V^{-1}Q,\ c = V^{-1}d, \tag{4.2}$$

for finding a solution of (4.1). The following result is [61, Corollary 2.2].

Proposition 4.1. *The iterates $\{x^k\}$ in method (4.2) converge to a solution of (4.1) if and only if $\rho(PT) < 1$, where $P = P_{R(Q^T)} = Q^\dagger Q$.*

For the symmetric case the following classical result by Keller holds [49, Theorem 2].

Proposition 4.2. *Assume that Q is symmetric, and let*

$$S = V + V^T - Q \text{ be positive definite.} \tag{4.3}$$

Then the iteration (4.2) is convergent if and only if Q is positive semidefinite.

However we wish to find both necessary and sufficient convergence conditions on the iteration parameters (which are hidden in the matrix S) so the following result is more appropiate [37, Theorem 3.1].

Theorem 4.3. *Assume that Q is symmetric and positive semidefinite. Then the iteration (4.2) is convergent if and only if S is positive definite.*

The proof in [37] is different in character from the one in [49]. Here we give a 'Keller'-like proof of Theorem 4.3. The first part (sufficiency) is taken from Keller (which we repeat for the convenience of the reader). Also by using Proposition 4.1 the proof can be much shortened.

Proof. Let (λ, u) be any eigenpair of T, i.e. $Tu = \lambda u$. Hence

$$Qu = (1 - \lambda) V u. \tag{4.4}$$

It follows that $\lambda = 1$ if and only if $Qu = 0$. Now let $u \notin N(Q)$ be an eigenvector of T with $\lambda = \alpha + i\beta$ the corresponding eigenvalue. Then $Qu \neq 0$ so that $\lambda \neq 1$ and $(Qu, u) > 0$. Take the inner product of (4.4) with u to get

$$\frac{1}{1 - \lambda} = \frac{(Vu, u)}{(Qu, u)}. \tag{4.5}$$

By adding (4.5) to its complex conjugate, and using that Q is symmetric and that $(V\bar{u}, u) = (u, Vu) = (V^T u, u)$ it follows

$$2Re \frac{1}{1 - \lambda} = \frac{((V + V^T)u, u)}{(Qu, u)} = \frac{((S + Q)u, u)}{(Qu, u)} = 1 + \frac{(Su, u)}{(Qu, u)}. \tag{4.6}$$

Now $Re(1 - \lambda)^{-1} = (1 - \alpha)/((1 - \alpha)^2 + \beta^2)$. Let

$$\varphi(\alpha, \beta) = \frac{2(1 - \alpha)}{(1 - \alpha)^2 + \beta^2} \tag{4.7}$$

Then (4.6) becomes

$$\varphi(\alpha, \beta) = 1 + \frac{(Su, u)}{(Qu, u)}. \tag{4.8}$$

Sufficiency, (following [49]). Assume that S is positive definite. Then by (4.8) $\varphi(\alpha, \beta) > 1$ which yields

$$2(1 - \alpha) > (1 - \alpha)^2 + \beta^2, \text{ or } \alpha^2 + \beta^2 < 1.$$

Since $\rho(PT) = \rho(TP)$ convergence follows by Proposition 4.1.

Necessity, (new). Next assume that the iteration (4.2) is convergent, i.e., (by Proposition 4.1), $\alpha^2+\beta^2<1$. Then

$$\varphi(\alpha,\beta) > \frac{2(1-\alpha)}{(1-\alpha)^2+sup(\beta^2)} = \frac{2(1-\alpha)}{(1-\alpha)^2+1-\alpha^2} = 1.$$

We conclude by (4.8) that

$$\frac{(Su,u)}{(Qu,u)} > 0,$$

whence S is positive definite. □

We will now prove the convergence theorems, and start with Algorithm BCI.

Proof. **of Theorem 3.2** (Algorithm BCI). By Proposition 3.1, $T = I - M_{cl}A^TA$, $Q = A^TA$, $d = A^Tb$. Hence any limit-point will be a least squares solution (independent of M_{cl}). Further $V^{-1} = M_{cl} = (D_c+L_c)^{-1}$. It follows

$$S = V + V^T - Q = 2D_c + L_c + L_c^T - A^TA = 2D_\mathrm{c} - \mathrm{diag}(A_i^TA_i).$$

We will show that $\omega_i \in \bar{\Omega}_i, i = 1,2,\ldots,q \Leftrightarrow S$ is positive definite. Hence the result will follow by Theorem 4.3. Assume first that S is positive definite. Since both D_c and $\mathrm{diag}(A_i^TA_i)$ are block-diagonal with the same size of the corresponding blocks the following inequalities hold

$$\frac{2}{\omega_i}v_i^TN_i^{-1}v_i - v_i^TA_i^TA_i\,v_i > 0, \quad i = 1,2,\ldots,q$$

or equivalently (using that $v_i^TA_i^TA_iv_i = \|A_iv_i\|_2^2$)

$$0<\omega_i<2/c_i, \quad c_i = \frac{\|A_iv_i\|_2^2}{v_i^TN_i^{-1}v_i}, \quad \forall v_i \in \mathbb{R}^{n_i}, v_i \neq 0. \tag{4.9}$$

Put $v_i = N_i^{1/2}\xi_i$. Then

$$c_i = \left\|A_iN_i^{1/2}\xi_i\right\|_2^2/\|\xi_i\|_2^2 \leq \left\|A_iN_i^{1/2}\right\|_2^2.$$

For any matrix X it holds $\|X\|_2^2 = \|XX^T\|_2 = \rho(XX^T)$. Thus $\left\|A_iN_i^{1/2}\right\|_2^2 = \rho\big(A_iN_iA_i^T\big)$. Hence the relaxation parameters $\{\omega_i\}$ fulfill (3.4).

Next assume (3.4), i.e. $\omega_i < 2/\rho(A_iN_iA_i^T)$. Now (as shown above) $c_i \leq \rho(A_iN_iA_i^T)$. It follows (for all $i = 1,2,\ldots,q$) $\omega_i < 2/\rho(A_iN_iA_i^T) \leq 2/c_i$. So by (4.9) S is positive definite, which completes the proof. □

The 'if-part' was already proved in [36, Proposition 9].

Proof. **of Theorem 2.5** (simultaneous iteration). Here $T = I - \omega A^TMA$. Let $Q = A^TMA, d = A^TMb$. Since M is spd the limit-point satisfies $\arg\min \|M^{1/2}(b-Ax\|_2$. Put $V^{-1} = \omega I$. We use Theorem 4.3 where now

$$S = V + V^T - Q = \frac{2}{\omega}I - A^TMA.$$

Assume first that S is spd. We get (similarly as (4.9))

$$0<\omega<2/c, \quad c = \frac{\|M^{1/2}Av\|_2^2}{\|v\|_2^2}, \quad \forall v \in \mathbb{R}^n, v \neq 0. \tag{4.10}$$

The result now follows as in the proof of Theorem 3.2. □

Proof. **of Theorem 2.7** (Algorithm SBRI). By comparing the two iterations (2.13) and (4.2) we find that $Q = A^T M_{sr} A,\ V = I$. It is shown in [34, Proposition 10] that (i): M_{sr} (and hence Q) is spd if and only if the conditions (2.7) hold, and (ii): the matrix $I - Q$ is spd. Since $S = 2I - Q$ the convergence result follows from Theorem 4.3. Also by the fact that M_{sr} is spd the limit point is a weighted least squares solution. □

Proof. **of Theorem 2.3** (Algorithm BRI, consistent case). We first observe that Theorem 4.3 cannot be used here since $Q = A^T M_{rw} A$ does not fulfill the conditions needed. Instead we apply Algorithm BCI (iterating in y_i^k) on the system $A^T y = z$ with $b = Az$. Further we take $q = p, m_i = n_i, N_i = M_i, A_i = R_i^T,\ i = 1, 2, \ldots, p$. By Proposition 3.3 one cycle can be written

$$y^{k+1} = y^k + M_{cl} A(z - A^T y^k). \tag{4.11}$$

By the above choices it follows that $L_c = L_r, D_c = D_r$ so that $M_{cl} = M_{rw}$. Also $\rho(A_i N_i A_i^T) = \rho(R_i^T M_i R_i)$. Hence with $T = I - M_{rw} AA^T, Q = AA^T, d = Az$ we can use Theorem 3.2 to conclude that $\lim y^k \to \hat{y}$ such that $AA^T \hat{y} = Az = b$ if and only if conditions (2.7) hold. Multiplying (4.11) by A^T, and putting $x^k = A^T y^k$ it follows that

$$x^{k+1} = x^k + A^T M_{rw}(b - Ax^k)$$

which by Proposition 2.7 is identical to one cycle of Algorithm BRI. It follows that when $y^k \to \hat{y}$ then $x^k \to \hat{x}$, such that $A\hat{x} = b$. Hence (2.7) is sufficient for convergence. If on the other hand $x^k = A^T y^k \to \hat{x}$, and $N(A^T) = \emptyset$ then y^k must also converge so that (2.7) becomes necessary for convergence. Similarly if $R(M_{rw}) \subseteq R(A)$ then $y^k \in R(A)$ so y^k must converge. Hence (2.7) is necessary also in this case. □

5. Miscelaneous

Here we shortly mention some topics not covered above. The first is constraints. To impose wanted properties on the solution of an ill-posed problem can be very beneficial. In, e.g. X-ray tomography the attenuation x is known to be nonnegative. To include this property into the solution process using Kaczmarz's method one simply, usually after each cycle, project the current iterate onto the nonnegative orthant. This fits well with the overall structure of the method. For more on constraining see [24, 44, 15, 11, 57]. In particular the connection between incremental proximal methods [7], and row-action methods is utilized in [3] to extend these to include both convex constraints and regularization terms.

Another, quite recent, methodology to incorporate constraints is superiorization. Here the original iteration, e.g. Algorithm BRI, is preserved but perturbation steps are included between iterations. The perturbation steps move the iterates according to a secondary criterion (corresponding to the constraints) [16, 23, 53].

Another subject we wish to mention is semi-convergence, which was brought up already in [51], see also [44]. Let $b = \bar{b} + \delta b$, and let $\bar{x}^k$ denote the iterate corresponding to using the (ideal) data $\bar{b}$ whereas x^k uses the given data b. With $\hat{x}$ the sought solution we have the following decomposition of the error

$$x^k - \hat{x} = (x^k - \bar{x}^k) + (\bar{x}^k - \hat{x}),$$

where the first term is called the *noise error* (or data error) and the second term is called the *iteration error*. During the first iterations of a convergent method the iteration error dominates, and hence the total error decreases - but after a while the noise error starts to grow which results in semi-convergence. Expressions for the noise error for Algorithm BRI was recently presented in [35] (this analysis also includes projections onto arbitrary closed convex sets) and independently in [50]. Both bounds are of the form $c\sqrt{k}\|\delta b\|$. However the constant c given in [35] usually grossly overestimates the real error. The analysis in [50] come with a price though. The iteration matrices governing the two errors are different. Therefore to insure the result a new sufficient condition is needed. In [33] a bound of the same form $c\sqrt{k}\|\delta b\|$ is presented for Algorithm BCI. Here the same iteration matrix governs both the noise and iteration error so no new condition is needed. Due to semi-iteration it is important to stop the iterations before the noise error starts to dominate. We refer to [43] for a recent discussion and evaluation of several stopping criteria.

As a final point we mention the possibility of accelerating the basic methods by Chebycheff or Conjugate Gradient technique [9, 42, 4, 10, 60, 39, 40].

Acknowledgement

To Yair Censor for longtime friendship and cooperation

References

[1] R. Aharoni and Y. Censor, *Block-iterative projection methods for parallel computation of solutions to convex feasibility problems*, Linear Algebra Appl. **120** (1989), 165–175.

[2] A. Andersen and A. Kak, *Simultaneous algebraic reconstruction technique (sart): A superior implementation of the art algorithm*, Ultrasonic Imaging **6** (1984), 81–94.

[3] M. S. Andersen and P. C. Hansen, *Generalized row-action methods for tomographic imaging*, Numer. Algorithms **67** (2014), 121–144.

[4] M. Arioli, I. Duff and J. Noailles, *A block projection method for sparse matrices*, SIAM J. Sci. Comput. **13** (1992), 47–70.

[5] Z-Z. Bai and C-H. Jin, *Column-decomposed relaxation methods for the overdetermined systems of linear equations*, Int. J. Appl. Math. **13** (2003), 71–82.

[6] H. Bauschke and J. M. Borwein, *On projection algorithms for solving convex feasibility problems*, SIAM Rev. **38** (1996), 367–426.

[7] D.P. Bertsekas, *Incremental proximal methods for large scale convex optimization*, Math. Program. **129** (2011), 163–195.

[8] Å. Björck, *Numerical Methods for Least Squares Problems*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996.

[9] Å. Björck and T. Elfving, *Accelerated projection methods for computing pseudoinverse solutions of systems of linear equations*, BIT **19** (1979), 145–163.

[10] R. Bramley and A. Sameh, *Row projection methods for large nonsymmetric linear systems*, SIAM J. Sci. Comput. **13** (1992), 168–193.

[11] C. Byrne, *Applied Iterative Methods*, A K Peters, Wellesley, MA, 2008.

[12] C. Byrne, *Bounds on the largest singular value of a matrix and the convergence of simultaneous and block-iterative algorithms for sparse linear systems*, Int. Trans. Oper. Res. **16** (2009), 465–479.

[13] A. Cegielski, *Iterative Methods for Fixed Point Problems in Hilbert Spaces*, vol. 2057 of Lecture Notes in Mathematics, Springer, Heidelberg, 2012.

[14] Y. Censor, *Row-action methods for huge and sparse systems and their applications*, SIAM Rev. **23** (1981), 444–466.

[15] Y. Censor, W. Chen, P. Combettes, R. Davidi and G. T. Herman, *On the effectiveness of projection methods for convex feasibility problems with linear inequality constraints*, Comput. Optim. Appl. **51** (2012), 1065–1088.

[16] Y. Censor, R. Davidi and G. T. Herman, *Perturbation resilience and superiorization of iterative algorithms*, Inverse Problems **26** (2010), 065008, 12.

[17] Y. Censor, P. Eggermont and D. Gordon, *Strong underrelaxation in Kaczmarz's method for inconsistent systems*, Numer. Math. **41** (1983), 83–92.

[18] Y. Censor and T. Elfving, *Block-iterative algorithms with diagonally scaled oblique projections for the linear feasibility problem*, SIAM J. Matrix Anal. Appl. **24** (2002), 40–58.

[19] Y. Censor, T. Elfving and G. T. Herman, *Averaging strings of sequential iterations for convex feasibility problems*, in Inherently parallel algorithms in feasibility and optimization and their applications (Haifa, 2000), vol. 8 of Stud. Comput. Math., North-Holland, Amsterdam, 2001, pp. 101–113.

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

[21] Y. Censor, D. Gordon and R. Gordon, *Bicav: An inherently parallel algorithm for sparse systems with pixel-dependent weighting*, IEEE Transactions on Medical Imaging **20** (2001), 1050–1060.

[22] Y. Censor and S. Reich, *Iterations of paracontractions and firmly nonexpansive operators with applications to feasibility and optimization*, Optimization **37** (1996), 323–339.

[23] Y. Censor and A. Zaslavski, *Convergence and perturbation resilience of dynamic string-averaging projection methods*, Comput. Optim. Appl. **54** (2013), 65–76.

[24] Y. Censor and S. Zenios, *Parallel optimization*, Theory, algorithms, and applications, With a foreword by George B. Dantzig. Numerical Mathematics and Scientific Computation, Oxford University Press, New York, 1997.

[25] G. Cimmino, *Calcolo approssimato per le soluzioni dei sistemi di equazioni lineari*, La Ricerca Scientifica, XVI (1938), 326–333.

[26] A. Dax, *The convergence of linear stationary iterative processes for solving singular unstructured systems of linear equations*, SIAM Rev. **32** (1990), 611–635.

[27] A. De Pierro and A. Iusem, *A simultaneous projections method for linear inequalities*, Linear Algebra Appl. **64** (1985), 243–253.

[28] N. Echebest, M. Guardarucci, H. Scolnik and M. Vacchino, *An accelerated iterative method with diagonally scaled oblique projections for solving linear feasibility problems*, Ann. Oper. Res. **138** (2005), 235–257.

[29] P. Eggermont, G. T. Herman, and A. Lent, *Iterative algorithms for large partitioned linear systems, with applications to image reconstruction*, Linear Algebra Appl. **40** (1981), 37–67.

[30] Y. Eldar and D. Needell, *Acceleration of randomized Kaczmarz method via the Johnson-Lindenstrauss lemma*, Numer. Algorithms **58** (2011), 163–177.

[31] T. Elfving, *Group-iterative methods for consistent and inconsistent linear equations*, Lith-Mat-Report-1977-11, Linköping Univ. (1977).

[32] T. Elfving, *Block-iterative methods for consistent and inconsistent linear equations*, Numer. Math. **35** (1980), 1–12.

[33] T. Elfving and P. C. Hansen, *Unmatched projector/backprojector pairs: perturbation and convergence analysis*, SIAM J. Sci. Comput. **40** (2018), A573–A591.

[34] T. Elfving and T. Nikazad, *Properties of a class of block-iterative methods*, Inverse Problems **25** (2009), 115011, 13.

[35] T. Elfving, P. C. Hansen and T. Nikazad, *Semi-convergence properties of Kaczmarz's method*, Inverse Problems **30** (2014), 055007, 16.

[36] T. Elfving, P. C. Hansen and T. Nikazad, *Convergence analysis for column-action methods in image reconstruction, (erratum: fig. 3 was incorrect, p 925)*, Numer. Algorithms **74** (2017), 905–924.

[37] A. Frommer, R. Nabben and D. Szyld, *Convergence of stationary iterative methods for Hermitian semidefinite linear systems and applications to Schwarz methods*, SIAM J. Matrix Anal. Appl. **30** (2008), 925–938.

[38] A. Galántai, *Projectors and projection methods*, Kluwer Academic Publishers, Boston, MA, 2004.

[39] D. Gordon and R. Gordon, *CARP-CG: a robust and efficient parallel solver for linear systems, applied to strongly convection dominated PDEs*, Parallel Comput. **36** (2010), 495–515.

[40] D. Gordon, *A derandomization approach to recovering bandlimited signals across a wide range of random sampling rates*, Numerical Algorithms, (2017).

[41] R. Gordon, R. Bender and G. T. Herman, *Algebraic reconstruction technique (ART) for three-dimensional electron microscopy and x-ray photography*, Journal of Theoretical Biology **29** (1970), 471–481.

[42] M. Hanke and W. Niethammer, *On the acceleration of Kaczmarz's method for inconsistent linear systems*, Linear Algebra Appl. **130** (1990), 83–98.

[43] P. C. Hansen and J. Jørgensen, *AIR Tools II: algebraic iterative reconstruction methods, improved implementation*, Numerical Algorithms, (2017).

[44] G. T. Herman, *Fundamentals of computerized tomography*, Advances in Pattern Recognition, Springer, Dordrecht, second ed., 2009.

[45] R. Horn and C. Johnson, *Matrix analysis*, Cambridge University Press, Cambridge, 1985.

[46] H. Hudson and R. Larkin, *Accelerated image reconstruction using ordered subsets projection data*, IEEE Transactions on Medical Imaging **13** (1994), 601–609.

[47] M. Jiang and G. Wang, *Convergence studies on iterative algorithms for image reconstruction*, IEEE Transactions on Medical Imaging **22** (2003), 569–579.

[48] S. Kaczmarz, *Angenäherte auflösung von systemen linearer gleichungen*, Bulletin de l'Academie Polonaise des Sciences et Lettres **A35** (1937), 355–357.

[49] H. Keller, *On the solution of singular and semidefinite linear systems by iteration*, J. Soc. Indus. Appl. Math.: ser. B **2** (1965), 281–290.

[50] S. Kindermann and A. Leitão, *Convergence rates for Kaczmarz-type regularization methods*, Inverse Probl. Imaging **8** (2014), 149–172.

[51] F. Natterer, *The mathematics of computerized tomography*, B.G. Teubner, Stuttgart, 1986.

[52] D. Needell and J. Tropp, *Paved with good intentions: analysis of a randomized block Kaczmarz method*, Linear Algebra Appl. **441** (2014), 199–221.

[53] T. Nikazad and M. Abassi, *Perturbation-resilient iterative methods with an infinite pool of mappings*, SIAM J. Numer. Anal. **53** (2015), 390–404.

[54] P. Oswald and W. Zhou, *Convergence analysis for Kaczmarz-type methods in a Hilbert space framework*, Linear Algebra Appl. **478** (2015), 131–161.

[55] S. Penfold, R. Schulte, Y. Censor, V. Bashkirov, S. McAllister, K. Schubert and A. Rosenfeld, *Block-iterative and string-averaging projection algorithms in proton computed tomography image reconstruction*, in Biomedical Mathematics: Promising directions in Imaging, Theraphy Planning and Inverse Problems, Y. Censor, M. Jiang, and G. Wang, eds., Madison, WI, Medical Physics Publishing, 2009.

[56] C. Popa, *Extended and constrained diagonal weighting algorithm with application to inverse problems in image reconstruction*, Inverse Problems **26** (2010), 065004, 17.

[57] C. Popa, *Projection Algorithms-classical results and developments. Applications to Image Reconstructions*, Lambert Academic Publishing, Saarbrucken, Germany, 2012.

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

[59] G. Qu, C. Wang and M. Jiang, *Necessary and sufficient convergence conditions for algebraic image reconstruction algorithms*, IEEE Trans. Image Process. **18** (2009), 435–440.

[60] Y. Saad, *Iterative Methods for Sparse Linear Systems, 2nd ed.*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2003.

[61] X. Shi, Y. Wei and W. Zhang, *Convergence of general nonstationary iterative methods for solving singular linear equations*, SIAM J. Matrix Anal. Appl. **32** (2011), 72–89.

[62] H. Sörensen and P. C. Hansen, *Multicore performance of block algebraic iterative reconstruction methods*, SIAM J. Sci. Comput. **36** (2014), C524–C546.

[63] C. Sorzano, R. Marabini, G. T. Herman and J.M.Carazo, *Multiobjective algorithm parameter optimization using multivariate statistics in three-dimensional electron microscopy reconstruction*, Pattern Recognition **38** (2005), 2587–2601.

[64] T. Strohmer and R. Vershynin, *A randomized Kaczmarz algorithm with exponential convergence*, J. Fourier Anal. Appl. **15** (2009), 262–278.

[65] K. Tanabe, *Projection method for solving a singular system of linear equations and its applications*, Numer. Math. **17** (1971), 203–214.

[66] D. Watt, *Column-relaxed algebraic reconstruction algorithm for tomography with noisy data*, Applied Optics **33** (1994), 4420–4427.

[67] G. L. Zeng and G. T. Gullberg, *Unmatched projection/backprojection pairs in iterative reconstruction algorithms*, IEEE Trans. on Medical imaging **19** (2000), 548–555.

Manuscript received December 26 2017
revised February 23 2018

T. Elfving
Department of Mathematics, Linköping University, SE-581 83 Linköping, Sweden
E-mail address: `tommy.elfving@liu.se`