Solving systems of nonlinear equations by means of an accelerated successive orthogonal projections method

José Mário MARTINEZ
Applied Mathematics Laboratory, IMECC-UNICAMP, 13.100, Campinas, SP, Brazil

Received 7 Augustus 1985

Abstract: In this paper we introduce an acceleration procedure for a block version of the generalization of Kaczmarz's method for nonlinear systems of equations. We prove a local linear convergence theorem. Some numerical experiments are presented, which show that the new method improves the nonlinear Kaczmarz's method without acceleration.

1. Introduction

Let \( D \) be a subset of \( \mathbb{R}^n \) and let \( F: D \to \mathbb{R}^n \) be a nonlinear function. We wish to consider computing a solution \( x^* \) of the system of nonlinear equations
\[
F(x) = 0, \quad F = (f_1, \ldots, f_n)^T,
\]
where the Jacobian matrix
\[
J(x) = \left( \frac{\partial f_i(x)}{\partial x_j} \right)
\]
may be very large.

Tompkins [27], McCormick [16] and others [13,14,15,16,17] proposed generalizations of Kaczmarz's method [9] for nonlinear systems of equations. Kaczmarz's method and its generalizations make no changes in the original system, perform no operation on the system as a whole, and require access to only one component, or small group of components, at a time. These are the reasons why storage requirements for these methods are very low, in comparison to traditional methods, including those which were specially introduced for solving large problems [4,11,12,23].

The generalizations of Kaczmarz's method apply to many different problems (see [2] and references therein) and are called by Censor [2] row-action methods. The nonlinear Kaczmarz's methods belong to this class.

On the other hand, some authors incorporated acceleration procedures in order to improve the speed of convergence of different row-action methods (see, for example [28]). In this paper, we accelerate the nonlinear Kaczmarz method using a generalization of the scheme introduced by De Pierro [5] for linear systems. The idea is simple: Given two consecutive iterations \( x^k \) and \( x^{k+1} \), the accelerated iteration \( y \) is an approximation to the point on the line \( [x^k, x^{k+1}] \) which is
closest to the solution. De Pierro [5] proved the convergence of this algorithm for the linear case, both in the singular and nonsingular situations.

The nonlinear Kaczmarz's type algorithms may be classified as Generalized Linear Methods, in the sense of [21, pp. 214–229]. Most of these methods are potentially useful for solving huge systems of equations (see [25, 26]), but to our knowledge, only the generalizations of Kaczmarz's method converge without special conditions on the Jacobian matrix of the system at the solution.

In Section 2 of this paper we present the new method and in Section 3 we prove a local convergence theorem related to it. The convergence of the new method is obtained for a relaxed version of it and under a cyclic control assumption (see [2]). However, proving the convergence for a quasi-cyclic control in the sense of [2], using the arguments of Section 3, is more a matter of language than anything else.

In Section 4, we present some numerical experiments, which show that the new method is really an improvement over the nonlinear Kaczmarz's method without acceleration. Finally, in Section 5 we state some conclusions and suggest the lines for future research.

Notation

We adopt the following notation:
- $\| \cdot \|$ the 2-norm of a vector or matrix in $\mathbb{R}^n$;
- $\| \cdot \|_E$ the norm on the normed linear space $E$;
- $B(x, \epsilon)$ the open ball with center $x$ and radius $\epsilon$;
- $N(A)$ the null-space of the matrix $A$;
- $J(x)(J_i(x))$ the Jacobian matrix of $F(x)(F_i(x))$.

2. The proposed method

General hypotheses

Let $F: D \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$, $F \in C'(D)$, $D$ an open and convex set. Let $F(x) = 0$ and $J(x^*)$ be a nonsingular $n \times n$ matrix. Moreover, assume that for all $x \in D$,

$$\|J(x) - J(x^*)\| \leq K \|x - x^*\|^p, \quad K, \ p > 0.$$ 

This implies (see [1]) that for all $x \in D$

$$\|F(x) - J(x^*)(x - x^*)\| \leq K \|x - x^*\|^{p+1}.$$ 

Grouping some components of $F$ and with possible repetitions, the system $F(x) = 0$ is equivalent to

$$F_1(x) = 0,$$

$$\vdots$$

$$F_m(x) = 0,$$

where $F_i: D \rightarrow \mathbb{R}^n$, $i = 1, \ldots, m$.

Let us assume that inside each block $F_i$, no components of $F$ are repeated. This implies that the rows of $J_i(x^*)$ are linearly independent. By the continuity of $J_i(x)$ we may assume, without loss of generality, that the rows of $J_i(x)$ are linearly independent for all $x \in D$, $i = 1, \ldots, m$ (see [14]).
Basic definitions

Consider \( x \in D, 0 < \delta \leq 1 \), and \( w_i \in [\delta, 2 - \delta] \) for \( i = 1, \ldots, m \). We define:

\[
x_0 = x,
\]

\[
u_{i+1} = -w_{i+1} \left[ J_{i+1}(x_i) J_{i+1}(x_i)^T \right]^{-1} F_{i+1}(x_i), \quad i = 0, 1, \ldots, m - 1.
\]

Of course, \( x_i \) and \( u_i \) are functions of \( x, w_1, \ldots, w_m \), but we don’t make this dependence explicit in order to simplify the notation. If \( x^0 \) is an arbitrary initial point, and \( x^{k+1} \) is obtained from \( x^k \) by setting \( x^k = x, x^{k+1} = x_m \), then (1) describes an iteration of the Nonlinear Successive Orthogonal Projections Method, as presented in [13].

The acceleration step proceeds as follows: Define

\[
\lambda = - \sum_{i=1}^{m} \langle F_i(x), u_i \rangle / \| x_m - x \|^2, \quad \lambda = 0 \quad \text{if} \quad x = x^*,
\]

and set

\[
A(x, w) = x + \lambda (x_m - x).
\]

In the linear case [5], \( A(x, w) \) represents the point on the line \([x, x_m]\) which is closest to \( x^* \).

The following example in two variables may illustrate the behaviour of the acceleration procedure. Consider the system

\[
f_1(x) = 10(x_2 - x_1^2) = 0, \\
f_2(x) = 1 - x_1 = 0, \quad x^* = (1,1).
\]

Set \( q = 1, w_1 = w_2 = 1, \alpha(x) = x_m, \beta(x) = A(x, w) \).

The functions \( \alpha(x) \) and \( \beta(x) \) then satisfy:

\[
r = \lim_{\epsilon \to 0} \max \{ \| \alpha(x) - x^* \| / \| x - x^* \| = \epsilon \} \\
= \lim_{\epsilon \to 0} \max \{ \| \beta(x) - x^* \| / \| x - x^* \| = \epsilon \} \approx 0.895.
\]

So that \( r \) reflects the asymptotic ‘worst possible behaviour’ of the two algorithms.

On the other hand, the ‘average behaviour’ of the sequences defined by \( \alpha \) and \( \beta \) is better represented by

\[
r_1 = \lim_{\epsilon \to 0} \left( \frac{1}{2\pi\epsilon} \int_{\| x - x^* \| = \epsilon} \| \alpha(x) - x^* \| \, ds \right),
\]

\[
r_2 = \lim_{\epsilon \to 0} \left( \frac{1}{2\pi\epsilon} \int_{\| x - x^* \| = \epsilon} \| \beta(x) - x^* \| \, ds \right),
\]

in which simple numerical computations lead to

\[
r_1 \approx 0.569, \quad r_2 \approx 0.473.
\]

Hence, on the average, the accelerated SOP Method improves the classical algorithm, at least in this example.
3. A local convergence theorem

In this section we present first some simple lemmas which, in turn, will be useful in proving local convergence theorem for the accelerated nonlinear Kaczmarz method. The detailed proofs are available with the author.

Lemma 1. Let \( E \) be a normed linear space, \( W \) an arbitrary set, \( \Omega \) an open set \( \Omega \subset E \), \( x^* \in \Omega \). Let \( T: \Omega \times W \to E \), \( \Phi: \Omega \times W \to E \) such that for all \( x \in \Omega \), \( w \in W 
\)

\[
\| T(x, w) - x^* \| \leq \alpha \| x - x^* \|
\]

\[
\| T(x, w) - \Phi(x, w) \| \leq \beta(x) \| x - x^* \|
\]

with \( \beta: \Omega \to \mathbb{R}, \lim_{x \to x^*} \beta(x) = 0 \).

(i) Then, given \( \theta > 0 \), there exists \( \epsilon > 0 \) such that

\[
\| \Phi(x, w) - x^* \| \leq (\alpha + \theta) \| x - x^* \|
\]

for all \( x \in B(x^*, \epsilon), w \in W \).

(ii) Suppose \( \alpha < 1 \). If a sequence \( (x^k) \) is defined by

\[
x^0 \in B(x^*, \epsilon), \quad w_k \in W, \quad x^{k+1} = \Phi(x^k, w_k),
\]

for all \( k = 0, 1, 2, \ldots \), then the sequence \( (x^k) \) is well defined, \( \lim x^k = x^* \), and if \( x^k \neq x^* \) for all \( k = 0, 1, 2, \ldots \), \( \lim \sup \| x^{k+1} - x^* \| / \| x^k - x^* \| \leq \alpha \).

Some auxiliary functions

Under the same hypotheses as in Section 2, define

\[
\bar{x}_0 = x,
\]

\[
\bar{u}_{i+1} = -w_{i+1} \left[ J_{i+1}(x^*) J_{i+1}(x^*)^T \right]^{-1} J_{i+1}(x^*) (\bar{x}_i - x^*), \quad i = 0, 1, \ldots, m - 1.
\]

\[
\bar{x}_{i+1} = \bar{x}_i + J_{i+1}(x^*)^T \bar{u}_{i+1},
\]

Furthermore, introduce

\[
\bar{\lambda} = - \sum_{i=1}^{m} \langle J_i(x^*)(x - x^*), \bar{u}_i \rangle / \| \bar{x}_m - x \|^2,
\]

\[
\bar{\lambda} = - \langle x - x^*, x_m - x \rangle / \| x_m - x \|^2,
\]

and

\[
\bar{A}(x, w) = x + \bar{\lambda}(\bar{x}_m - x), \quad \tilde{A}(x, w) = x + \bar{\lambda}(x_m - x).
\]

Our technique for proving the convergence of the accelerated method under the hypotheses of Section 2, rests on the fact that the sequence defined by \( \bar{A} \) is ‘sufficiently close’ to the sequences defined by \( A \) and \( \tilde{A} \).

Lemma 2. There exists \( \epsilon > 0 \) such that, for \( x \in B(x^*, \epsilon) \) and \( w_i \in [\delta, 2 - \delta], i = 1, \ldots, m \) the functions \( x_i, \bar{x}_i, u_i, \bar{u}_i \) are well defined. Moreover, there exists \( c: B(x^*, \epsilon) \to \mathbb{R}, \lim_{x \to x^*} c(x) = 0 \)
such that for all \(i = 1, \ldots, m\)
\[\| x_i - \bar{x}_i \| \leq c(x) \| x - x^* \|,
\]
and
\[\| u_i - \bar{u}_i \| \leq c(x) \| x - x^* \|.
\]

**Lemma 3.** There exists \(\alpha \in [0, 1]\), \(r > 0\) such that for all \(x \in \mathbb{R}^n\), \(i = 1, \ldots, m\)
\[\| \bar{x}_i - x^* \| \leq \| x - x^* \|, \quad \| \bar{u}_i \| \leq r \| x - x^* \|,
\]
and
\[\| \bar{x}_m - x^* \| \leq \alpha \| x - x^* \|.
\]

**Lemma 4.** Let \(\theta > 0\), \(\alpha + \theta < 1\). There exists \(\epsilon > 0\) such that if \(x \in B(x^*, \epsilon)\),
\[\| x_i - x^* \| \leq (1 + \theta) \| x - x^* \|, \quad \| x_m - x^* \| \leq (\alpha + \theta) \| x - x^* \|,
\]
and
\[\| u_i \| \leq (r + \theta) \| x - x^* \|
\]
for all \(i = 1, \ldots, m\).

An obvious consequence of Lemma 4 is the local linear convergence of the nonlinear Kaczmarz’s method without acceleration.

**Lemma 5.** If \(x \in B(x^*, \epsilon)\), the operators \(A\), \(\bar{A}\), \(\hat{A}\) are well defined. Moreover, the following inequalities hold:  
\[(1 + \alpha) \| x - x^* \| \geq \| \bar{x}_m - x \| \geq (1 - \alpha) \| x - x^* \|, \]
\[(1 + \alpha + \theta) \| x - x^* \| \geq \| x_m - x \| \geq (1 - \alpha - \theta) \| x - x^* \|.
\]

**Lemma 6.** For all \(x \in \mathbb{R}^n\), \(w \in [\delta, 2 - \delta]^m\),
\[\| \bar{A}(x, w) - x^* \| \leq \alpha \| x - x^* \|.
\]

**Lemma 7.** There exists a real function \(c_1\) defined on \(B(x^*, \epsilon)\) such that \(\lim_{x \to x^*} c_1(x) = 0\) and for all \(x \in B(x^*, \epsilon)\), \(w_i \in [\delta, 2 - \delta]\), \(i = 1, \ldots, m\),
\[\| A(x, w) - \bar{A}(x, w) \| \leq c_1(x) \| x - x^* \|.
\]

**Lemma 8.** There exists a function \(c_2: B(x^*, \epsilon) \to \mathbb{R}\), \(\lim_{x \to x^*} c_2(x) = 0\), such that for all \(x \in B(x^*, \epsilon)\), \(w_i \in [\delta, 2 - \delta]\), \(i = 1, \ldots, m\),
\[\| \hat{A}(x, w) - A(x, w) \| \leq c_2(x) \| x - x^* \|.
\]

**Lemma 9.** There exists a function \(\bar{\beta}: B(x^*, \epsilon) \to \mathbb{R}\), \(\lim_{x \to x^*} \bar{\beta}(x) = 0\), such that for all \(x \in B(x^*, \epsilon)\), \(w_i \in [\delta, 2 - \delta]\), \(i = 1, \ldots, m\),
\[\| \bar{A}(x, w) - A(x, w) \| \leq \beta(x) \| x - x^* \|.
\]

We introduce now relaxation parameters on the acceleration operators as follows:
Let \(x \in B(x^*, \epsilon)\), \(w_1, \ldots, w_m \in [\delta, 2 - \delta]\), \(\xi \in [0, 2]\). Then we define
\[A_R(x, w, \xi) = x_m + \xi [A(x, w) - x_m], \quad \bar{A}_R(x, w, \xi) = \bar{x}_m + \xi [\bar{A}(x, w) - \bar{x}_m].
\]

Using Lemma 9 and the second part of Lemma 3, we may prove the following lemma.
Lemma 10. (i) For all $x \in \mathbb{R}^n$, $w \in [\delta, 2 - \delta]^m$, $\xi \in [0, 2]$,

$$\| \bar{A}_R(x, w, \xi) - x^* \| \leq \alpha \| x - x^* \|.$$ 

(ii) There exists a function $\beta : B(x^*, \epsilon) \to \mathbb{R}$, $\lim_{x \to x^*} \beta(x) = 0$, such that for all $x \in B(x^*, \epsilon)$, $w \in [\delta, 2 - \delta]^m$, $\xi \in [0, 2]$,

$$\| \bar{A}_R(x, w, \xi) - A_R(x, w, \xi) \| \leq \beta(x) \| x - x^* \|.$$ 

We are finally ready to state and prove our main result.

Theorem. Given $\theta > 0$, there exists $\epsilon > 0$ such that

$$\| A_R(x, w, \xi) - x^* \| \leq (\alpha + \theta) \| x - x^* \|$$

for all $x \in B(x^*, \epsilon)$, $w \in [\delta, 2 - \delta]^m$, $\xi \in [0, 2]$. If we define a sequence $(x^k)$ by

$$x^0 \in B(x^*, \epsilon), \quad w^k \in [\delta, 2 - \delta]^m, \quad \xi^k \in [0, 2];$$

$$x^{k+1} = A_R(x^k, w^k, \xi^k), \quad k = 0, 1, 2, \ldots,$$

then $x^k$ is well defined for all $k = 0, 1, 2, \ldots$, $\lim_{k \to \infty} x^k = x^*$, and if $x^k \neq x^*$ for all $k = 0, 1, 2, \ldots$

$$\lim \sup \| x^{k+1} - x^* \| / \| x^k - x^* \| \leq \alpha.$$ 

4. Numerical experiments

Let $F = (f_1, \ldots, f_n)^T$ be a nonlinear function on $\mathbb{R}^n$, $q$ a positive integer such that $q$ divides $n$, $\epsilon$ a small positive number. Suppose that $F$ is divided into $m = n/q$ blocks of $q$ components each. Then $F = (F_1, \ldots, F_m)$. The Successive Orthogonal Projections Method (SOP) is to be implemented as follows:

Step 0. $x \in \mathbb{R}^n$, KON = 0.
Step 1. FNOR = 0. For $i = 1, \ldots, m$ perform Steps 2–5.
Step 2. Compute $F_i(x)$.

FNOR = $\max\{\text{FNOR, } \| F_i(x) \|_\infty\}$.
If $\| F_i(x) \| \leq \epsilon$, go to Step 5.
Step 3. Compute $F'_i(x)$. Find $L$, a lower triangular matrix and $Q$, an orthogonal $n \times n$ matrix (which need not be stored) such that $F'_i(x) Q = L$.

If rank $L < q$, stop.
Step 4. Solve $L L^T u = -F_i(x)$.

Compute $z = F'_i(x)^T u$, $x = x + z$.
Step 5. Continue.
Step 6. If FNOR $\leq \epsilon$, declare 'convergence' and stop.
Step 7. KON $\leftarrow$ KON + 1. Go to Step 1.
The Accelerated Successive Orthogonal Projections Method (ASOP) is implemented as follows:

**Step 0.** \( x \in \mathbb{R}^n \), KON = 0.

**Step 1.** \( \hat{x} = x \), FNOR = 0.

\( P = 0 \). For \( i = 1, \ldots, m \) perform Steps 2-5.

**Step 2.** Compute \( F_i(x) \). FLAG \( \leftarrow 0 \).

FNOR = max\{FNOR, \( \| F_i(x) \|_\infty \)\}.

If \( \| F_i(x) \| \leq \epsilon \) go to Step 5.

**Step 3.** Compute \( F_i(\hat{x}) \). Find \( L \) and \( Q \) as in the Step 3 of SOP. FLAG \( \leftarrow 1 \).

**Step 4.** Solve \( LL^T u = -F_i(x) \). If FLAG = 1, put \( P \leftarrow P + \langle F_i(\hat{x}), u \rangle \).

Compute \( z = F'(x)^T u \), \( x \leftarrow x + z \).

**Step 5.** Continue.

**Step 6.** If FNOR \( \leq \epsilon \), declare 'convergence' and stop.

**Step 7.** Compute \( \lambda = -P / \| x - \hat{x} \|^2 \).

\( x \leftarrow \hat{x} + \lambda(x - \hat{x}) \),

KON \( \leftarrow \) KON + 1,

Go to Step 2.

The orthogonal factorizations in steps 3 of SOP and ASOP are computed using the algorithm of Nai-Kuan Tsao [19]. Of course, \( LL^T \) is the Choleski's factorization of \( F_i'(x)F_i'(x)^T \). However, the computation of this product leads to numerical instability, and so, it is better to compute \( L \) using Householder's transformations. The computation of \( u \) and \( z \) using this procedure is numerically stable (see [22]).

We tested ASOP against SOP for a number of classical test functions (see [18]). The experiments were performed on the PDP10 computer of the University of Campinas, in single precision, under the Fortran X compiler.

The test functions were the following:

**Function 1** (Brown).

\[
\begin{align*}
  f_i(x) &= x_{i+1} + \sum_{j=1}^{n} x_j - (n + 1), \quad i = 1, \ldots, n - 1, \\
  f_n(x) &= \prod_{j=1}^{n} x_j - 1.
\end{align*}
\]

Case 1: \( x^0 = (0.5, \ldots, 0.5) \).
Case 2: \( x^0 = (5, \ldots, 5) \).
Case 3: \( x^0 = (50, \ldots, 50) \).

**Function 2** (Deist-Sefor). \( n = 6 \).

\[
\sum_{j \neq i} \cot(\beta_j x_j)
\]
with
\[ \beta = (0.02249, 0.02166, 0.02083, 0.02, 0.01918, 0.01835). \]

Case 1: \( x^0 = (75, \ldots, 75) \).

**Function 3** (Broyden). \( n = 10 \).
\[
\begin{align*}
  f_1(x) &= (3 - 2x_1)x_1 - 2x_2 + 1, \\
  f_n(x) &= (3 - 2x_n)x_n - x_{n-1} + 1, \\
  f_i(x) &= (3 - 2x_i)x_i - x_{i-1} - 2x_{i+1} + 1. \quad i = 2, \ldots, n - 1.
\end{align*}
\]

Case 1: \( x^0 = (-1, \ldots, -1) \).
Case 2: \( x^0 = (-10, \ldots, -10) \).
Case 3: \( x^0 = (-100, \ldots, -100) \).

**Function 4** (Trigonometric of Spedicato). \( n = 10 \).
\[
\begin{align*}
  f_j(x) &= n - \sum_{j=1}^{n} \cos x_j + i(1 - \cos x_j) - \sin x_j.
\end{align*}
\]

Case 1: \( x^0 = (0.1, \ldots, 0.1) \).
Case 2: \( x^0 = (1, \ldots, 1) \).
Case 3: \( x^0 = (10, \ldots, 10) \).

**Function 5** (Discrete integral equation function).
\[
\begin{align*}
  f_i(x) &= x_i + \frac{1}{2}h \left[ (1 - t_i) \sum_{j=1}^{i} t_j(x_j + t_j + 1)^3 + t_i \sum_{j=i+1}^{n} (1 - t_j)(x_j + t_j + 1)^3 \right]
\end{align*}
\]
where \( h = 1/(n + 1) \), \( t_i = ih \) and \( x_0 = x_{n+1} = 0 \).
Case 1: \( x^0 = (\xi_j) \) where \( \xi_j = t_j(t_j - 1) \).

The results are presented in Table 1. The pair \( k_1, k_2 \) means that the algorithm converged using \( k_1 \) iterations and performing \( k_2 \) projection steps (Step 3 of SOP and ASOP). The triplet \( E, k, f \) means that the execution was stopped at iteration \( k_1 \) and the best value of FNOR reached was \( f \). Finally, \( ov, k \) means that the algorithm stopped because an overflow occurred at iteration \( k \). For the case of Function 5, we reported also the CPU time of the two algorithms. This is an important example because represents a large dense nonlinear system, for which compact storage of \( F'(x) \) may not be used.

**Remark.** Additional numerical experiments were performed considering the \( 64^2 \) difference approximation to the Poisson equation (see [24])
\[
\Delta u = u^3/(1 + 5^2 + t^2), \quad 0 \leq s \leq 1, \quad 0 \leq t \leq 1,
\]
\[
u(s, t) = \begin{cases} 
  1, & s = 0, \ t \in [0,1] \ or \ t = 0, \ s \in [0,1], \\
  2 - e^t, & t = 1, \ s \in [0,1], \\
  2 - e^t, & s = 1, \ t \in [0,1].
\end{cases}
\]
<table>
<thead>
<tr>
<th>Function</th>
<th>n</th>
<th>Case</th>
<th>q</th>
<th>SOP</th>
<th>ASOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>17, 17</td>
<td>17, 17</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>426, 2017</td>
<td>63, 285</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>5</td>
<td>17, 17</td>
<td>17, 17</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>898, 4332</td>
<td>0v, 3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>5</td>
<td>15, 15</td>
<td>29, 29</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>E, 1000, 1.7</td>
<td>ov, 3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>1</td>
<td>50</td>
<td>ov, 1</td>
<td>E, 25, 0.635</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25</td>
<td></td>
<td>ov, 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td></td>
<td>E, 200, 0.03</td>
<td>ov, 65</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td></td>
<td>E, 250, 0.03</td>
<td>E, 200, 0.07</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td>E, 350, 0.06</td>
<td>E, 250, 0.03</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>1</td>
<td>6</td>
<td>5, 5</td>
<td>5, 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td>E, 1000, 0.002</td>
<td>335, 669</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td>E, 1000, 0.9 E-3</td>
<td>63, 185</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td>E, 1000, 0.5 E-3</td>
<td>89, 528</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>1</td>
<td>10</td>
<td>3, 3</td>
<td>3, 3</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>9, 18</td>
<td>7, 13</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>12, 56</td>
<td>9, 34</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>21, 200</td>
<td>15, 140</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>10</td>
<td>7, 7</td>
<td>7, 7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>13, 26</td>
<td>10, 20</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>16, 77</td>
<td>10, 50</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>26, 253</td>
<td>18, 160</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>10</td>
<td>10, 10</td>
<td>10, 10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>16, 32</td>
<td>12, 24</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>19, 93</td>
<td>16, 69</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>30, 285</td>
<td>21, 196</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>1</td>
<td>10</td>
<td>E, 100, 20.2</td>
<td>E, 100, 18.3</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>E, 100, 20.3</td>
<td>E, 100, 19.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>E, 100, 1.77</td>
<td>47, 235</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>E, 100, 1.38</td>
<td>29, 289</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>2</td>
<td>10</td>
<td>E, 100, 19.8</td>
<td>E, 100, 27.9</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>19, 38</td>
<td>E, 100, 26.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>E, 100, 1.8</td>
<td>E, 100, 0.07</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>E, 100, 1.4</td>
<td>6, 60</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>10</td>
<td>E, 100, 27.9</td>
<td>E, 100, 20.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>E, 100, 13.1</td>
<td>E, 100, 25.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>9, 41</td>
<td>97, 485</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>ov, 15</td>
<td>20, 198</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>500</td>
<td>1</td>
<td>10</td>
<td>6, 170, 192&quot;</td>
<td>5, 170, 188&quot;</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>6, 344, 142&quot;</td>
<td>5, 334, 132&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>6, 862, 117&quot;</td>
<td>5, 826, 104&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>6, 1727, 109&quot;</td>
<td>5, 1646, 106&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>1</td>
<td>2</td>
<td>6, 1728, 486&quot;</td>
<td>5, 1647, 426&quot;</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>6, 3455, 448&quot;</td>
<td>6, 3286, 436&quot;</td>
<td></td>
</tr>
</tbody>
</table>
The initial point was \( u^0 = (-1)(\| F(u^0) \|_\infty = 25015) \) and we used \( q = 1 \). None of the methods converged after 10 minutes of CPU time. SOP used 721 iterations and arrived to FNOR = 740. On the other hand ASOP used 447 iteration arriving to FNOR = 673. Finally, a version of ASOP which used acceleration steps only every 10 iterations used 686 iterations and obtained FNOR = 234. This situation is interpreted in the final section of this paper.

5. Conclusions

Iterations of the form \( x^{k+1} = \Phi(x^k) \) may be written as

\[
x^{k+1} - x^* = \Phi'(x^*)(x^k - x^*).
\]

Therefore, the (unitary) error vector \( (x^{k+1} - x^*)/\| x^k - x^* \| \) tends to approximate the eigenvector of \( \Phi' \) which corresponds to the spectral radius of \( \Phi'(x^*) \). In fact, (2) represents an application of the Power Method to \( \Phi'(x^*) \) [8, pp. 187–188]. The approximate collinearity of \( x^k, x^{k+1}, x^* \) may be exploited in several ways [28]. The method which was presented in this paper may be viewed as one of the possible acceleration procedures along the above lines, but its justification does not depend of the accuracy of the collinearity as it happens to be with, e.g., Wainwright’s method [28]. Anyhow, the relation between these and other procedures is to be studied.

Although the linear rate of convergence is not better for the accelerated than for the nonaccelerated method, the numerical experiments confirm the intuitive feeling that the accelerated method is really an improvement over the nonaccelerated Kaczmarz’s algorithm. This results from the fact that the theoretical rate of convergence reflect ‘the worst case’ in the sense that the error \( \| x^k - x^* \| \) at iteration \( k \), reduces ‘at least’ to \( a \| x^k - x^* \| \) if \( x^k \) is near enough to the solution. However, the ‘average’ reduction of the error for all the possible iterates \( x \) such that \( \| x - x^* \| = \| x^k - x^* \| \) seems to be much larger for the accelerated than for the nonaccelerated method.

The block version of the method which was presented throughout (see also [6,7,13]) is interesting from several points of view: First, as in [10] we observe that for many systems of equations a substantial part of the work used to evaluate one component is common to the evaluation of other components. Therefore, much time can be saved in many problems choosing properly the blocks of components. Second, an appropriate ‘grouping’ of the \( f_j \)’s may be used in order to improve the speed of convergence, along the lines of Wainwright [28]. Finally, the partitioning of the system in rows and handling at each iterative step, a subset of equations seems to be an useful procedure in certain image reconstruction problems [3,20].

Further research is necessary in order to properly choose the relaxation parameters \( w^k, \xi^k \). In relation to the \( \xi^k \), we think that the main advantage of its introduction is to show that the determination of the acceleration step need not be very accurate. A direct consequence of this fact is the possible consideration of alternative formulae for the acceleration step.

References


