STUDIA MATHEMATICA 216 (3) (2013)

Kaczmarz algorithm with relaxation in Hilbert space

by

RYSZARD SZWARC and GRZEGORZ ŚWIDERSKI (Wrocław)

Abstract. We study the relaxed Kaczmarz algorithm in Hilbert space. The connection with the non-relaxed algorithm is examined. In particular we give sufficient conditions when relaxation leads to the convergence of the algorithm independently of the relaxation coefficients.

1. Introduction. Let $\{e_n\}_{n=0}^{\infty}$ be a linearly dense sequence of unit vectors in a Hilbert space $\mathcal{H}$. Define

$$x_0 = \langle x, e_0 \rangle e_0,$$
$$x_n = x_{n-1} + \langle x - x_{n-1}, e_n \rangle e_n.$$

The formula is called the *Kaczmarz algorithm* ([4]).

In this work we fix a sequence $\lambda = \{\lambda_n\}_{n=0}^{\infty}$ of relaxation coefficients so that $0 < \lambda_n < 2$ for any n. Then we define

$$\begin{aligned} x_0 &= \lambda_0 \langle x, e_0 \rangle e_0, \\ x_n &= x_{n-1} + \lambda_n \langle x - x_{n-1}, e_n \rangle e_n. \end{aligned} \tag{1.1}$$

Let Q_n denote the orthogonal projection onto the line $\mathbb{C}e_n$ and let $P_n = I - Q_n$. Then (1.1) takes the form

$$x_n = x_{n-1} + \lambda_n Q_n (x - x_{n-1}). \tag{1.2}$$

The last formula can be transformed into

$$x - x_n = (I - \lambda_n Q_n)(x - x_{n-1}) = [(1 - \lambda_n) Q_n + P_n](x - x_{n-1}). \tag{1.3}$$

Define

$$R_n = (1 - \lambda_n) Q_n + P_n. \tag{1.4}$$

Clearly R_n is a contraction. Iterating (1.3) gives

$$x - x_n = R_n R_{n-1} \ldots R_0 x.$$

2010 *Mathematics Subject Classification*: Primary 41A65.

Key words and phrases: Kaczmarz algorithm, Hilbert space, Gram matrix, relaxation, tight frame.

DOI: 10.4064/sm216-3-3

We are interested in determining when the algorithm converges, i.e. $x_n \to x$ for any x in the space.

This is always satisfied in a finite-dimensional space for a periodic choice of vectors and relaxation coefficients. Indeed, let $\dim \mathcal{H} < \infty$ and $\{e_n\}_{n=0}^{\infty}$, $\{\lambda_n\}_{n=0}^{\infty}$ be N-periodic. For $A = R_{N-1} \ldots R_1 R_0$ it suffices to show that A^n tends to zero. We claim that $\|A\| < 1$. If not, there is a vector x such that $\|Ax\| = \|x\| = 1$. Then $\|R_0 x\| \geq \|Ax\| = \|x\|$, hence $R_0 x = x$, which implies $P_0 x = x$. In the same way $P_1 x = x, \ldots, P_{N-1} x = x$, which implies that $x \perp e_0, e_1, \ldots, e_{N-1}$. As the vectors $\{e_n\}_{n=0}^{N-1}$ are linearly dense we get $x = 0$. The speed of convergence in the finite-dimensional case has been studied in [2].

In the infinite-dimensional case this work is a natural continuation of [6] where the non-relaxed algorithm was studied in detail. In particular the convergence was characterized in terms of the Gram matrix of the vectors e_n.

2. Main formulas. Define vectors g_n recursively by

$$g_n = \lambda_n e_n - \lambda_n \sum_{k=0}^{n-1} \langle e_n, e_k \rangle g_k \tag{2.1}$$

(see [5]). Then by straightforward induction it can be verified that

$$x_n = \sum_{k=0}^{n} \langle x, g_k \rangle e_k. \tag{2.2}$$

As the images of the projections P_n and Q_n are mutually orthogonal, in view of (1.3) we get

$$\begin{aligned} \|x - x_n\|^2 &= (1 - \lambda_n)^2 \|Q_n(x - x_{n-1})\|^2 + \|P_n(x - x_{n-1})\|^2, \\ \|x - x_{n-1}\|^2 &= \|Q_n(x - x_{n-1})\|^2 + \|P_n(x - x_{n-1})\|^2. \end{aligned}$$

Subtracting gives

$$\|x - x_{n-1}\|^2 - \|x - x_n\|^2 = \lambda_n(2 - \lambda_n)\, \|Q_n(x - x_{n-1})\|^2.$$

By (1.2) we thus get

$$\|x - x_{n-1}\|^2 - \|x - x_n\|^2 = \frac{2 - \lambda_n}{\lambda_n} \|x_n - x_{n-1}\|^2. \tag{2.3}$$

Now taking (2.2) into account results in

$$\|x - x_{n-1}\|^2 - \|x - x_n\|^2 = \frac{2 - \lambda_n}{\lambda_n} |\langle x, g_n \rangle|^2.$$

By summing up the last formula we obtain

$$\|x\|^2 - \lim_n \|x - x_n\|^2 = \sum_{n=0}^{\infty} \frac{2 - \lambda_n}{\lambda_n} |\langle x, g_n \rangle|^2.$$

Therefore the algorithm converges if and only if

$$\|x\|^2 = \sum_{n=0}^{\infty} \frac{2-\lambda_n}{\lambda_n} |\langle x, g_n\rangle|^2, \qquad x \in \mathcal{H}. \tag{2.4}$$

Define

$$h_n = \sqrt{\frac{2-\lambda_n}{\lambda_n}}\, g_n, \qquad f_n = \sqrt{\frac{2-\lambda_n}{\lambda_n}}\, e_n.$$

Then (2.1) takes the form

$$h_n = f_n - \sum_{k=0}^{n-1} \frac{1}{2-\lambda_k} \langle f_n, f_k\rangle h_k. \tag{2.5}$$

In view of (2.4) the algorithm converges if and only if

$$\|x\|^2 = \sum_{n=0}^{\infty} |\langle x, h_n\rangle|^2, \qquad x \in \mathcal{H}. \tag{2.6}$$

The last condition states that $\{h_n\}_{n=0}^{\infty}$ is a so-called *tight frame* (see [1]; cf. [6]). Equivalently the sequence h_n is linearly dense and the Gram matrix of the vectors h_n is a projection.

We are now going to describe the Gram matrix of the vectors h_n in more detail.

Define the lower triangular matrix M_λ by the formula

$$(M_\lambda)_{nk} = \frac{1}{2-\lambda_k} \langle f_n, f_k\rangle, \qquad n > k. \tag{2.7}$$

Thus (2.5) can be rewritten as

$$f_n = h_n + \sum_{k=0}^{n-1} (M_\lambda)_{nk} h_k. \tag{2.8}$$

Let U_λ be the lower triangular matrix defined by

$$(I + U_\lambda)(I + M_\lambda) = I. \tag{2.9}$$

Denote

$$(U_\lambda)_{nk} = c_{nk}, \qquad n > k.$$

Then (2.7)–(2.9) imply

$$h_n = f_n + \sum_{k=0}^{n-1} c_{nk} f_k.$$

Moreover setting $c_{nn} = 1$ gives

$$\langle h_i, h_j\rangle = \sum_{k=0}^{i} c_{ik} \sum_{l=0}^{j} \overline{c_{jl}} \langle f_k, f_l\rangle = \langle (I + U_\lambda) F_\lambda (I + U_\lambda^*) \delta_j, \delta_i\rangle, \tag{2.10}$$

where F_λ denotes the Gram matrix of the vectors f_n, i.e.

$$(F_\lambda)_{nk} = \langle f_n, f_k\rangle, \tag{2.11}$$

and δ_i is the standard basis in $\ell^2(\mathbb{N})$. We will denote by D_{a_n} the diagonal matrix with the numbers a_n on the main diagonal. By definition of the vectors f_n and by (2.7) we have

$$F_\lambda = D_{(2-\lambda_n)\lambda_n} + M_\lambda D_{2-\lambda_n} + D_{2-\lambda_n} M_\lambda^*. \tag{2.12}$$

We have

Lemma 2.1.

$$(I+U_\lambda)F_\lambda(I+U_\lambda^*) = I - (D_{1-\lambda_n} + U_\lambda D_{2-\lambda_n})(D_{1-\lambda_n} + D_{2-\lambda_n}U_\lambda^*). \tag{2.13}$$

Proof. The formula follows readily by using the relation

$$M_\lambda U_\lambda = U_\lambda M_\lambda = -M_\lambda - U_\lambda,$$

which comes from (2.9). ∎

Now we are ready to state one of the main results.

Theorem 2.2. *The relaxed Kaczmarz algorithm defined by* (1.1) *is convergent if and only if the matrix* $V_\lambda := D_{1-\lambda_n} + U_\lambda D_{2-\lambda_n}$ *is a partial isometry.*

Proof. By Lemma 2.1 the operator V_λ is a contraction. Again by Lemma 2.1 and (2.10) we get

$$\langle h_i, h_j\rangle = \langle (I - V_\lambda V_\lambda^*)\delta_j,\, \delta_i\rangle.$$

From the discussion after formula (2.6) we know that the algorithm converges if and only if the Gram matrix of the vectors h_i is a projection. But the latter is equivalent to V_λ being a partial isometry. ∎

3. Relaxed versus non-relaxed algorithm. For a constant sequence $\lambda \equiv 1$ let $M = M_1$ and $U = U_1$. From the definition of M_λ we get

$$M_\lambda = D_{\sqrt{\lambda_n(2-\lambda_n)}} M D_{\sqrt{\lambda_n/(2-\lambda_n)}}. \tag{3.1}$$

We would like to have a similar relation for V_λ (see Thm. 2.2). Clearly for $\lambda \equiv 1$ we have $V_1 = U$.

Lemma 3.1. *Let D_1 and D_2 be diagonal matrices with non-zero elements on the main diagonal. Let M, $\widetilde{M}$, U and $\widetilde{U}$ be lower triangular matrices so that $\widetilde{M} = D_1 M D_2$ and*

$$(I+M)(I+U) = I, \quad (I+\widetilde{M})(I+\widetilde{U}) = I.$$

Then

$$\widetilde{U} = D_1 U[I + (I - D_1 D_2)U]^{-1} D_2.$$

Proof. We have

$$M = -U(I+U)^{-1}, \qquad \widetilde{U} = -\widetilde{M}(I+\widetilde{M})^{-1}.$$

Thus

$$\begin{aligned}\widetilde{U} &= -D_1MD_2(I+D_1MD_2)^{-1} = -D_1M(I+D_1D_2M)^{-1}D_2\\ &= D_1U(I+U)^{-1}[I-D_1D_2U(I+U)^{-1}]^{-1}D_2\\ &= D_1U[(I+U)-D_1D_2U]^{-1}D_2 = D_1U[I+(I-D_1D_2)U]^{-1}D_2. \ \blacksquare\end{aligned}$$

PROPOSITION 3.2. *We have*

(3.2) $$V_\lambda := D_{1-\lambda_n} + U_\lambda D_{2-\lambda_n} = (A_\lambda + B_\lambda U)(B_\lambda + A_\lambda U)^{-1},$$

where

(3.3) $$A_\lambda = D_{(1-\lambda_n)/\sqrt{\lambda_n(2-\lambda_n)}}, \qquad B_\lambda = D_{1/\sqrt{\lambda_n(2-\lambda_n)}}.$$

Proof. Let

$$D_1 = D_{\sqrt{\lambda_n(2-\lambda_n)}}, \qquad D_2 = D_{\sqrt{\lambda_n/(2-\lambda_n)}}.$$

By (3.1) we have $M_\lambda = D_1MD_2$. We can apply Lemma 3.1 to get

$$U_\lambda = D_1U[I+(I-D_1D_2)U]^{-1}D_2.$$

Observe that $D_1D_2 = D_{\lambda_n}$ and $D_2D_{2-\lambda_n} = D_1$. Thus

$$\begin{aligned}V_\lambda &= I - D_1D_2 + D_1U[I+(I-D_1D_2)U]^{-1}D_1\\ &= \{D_1^{-1}(I-D_1D_2)[I+(I-D_1D_2)U] + D_1U\}[I+(I-D_1D_2)U]^{-1}D_1\\ &= \{(D_1^{-1}-D_2) + [D_1^{-1}(I-D_1D_2)^2 + D_1]U\}[D_1^{-1} + (D_1^{-1}-D_2)U]^{-1}.\end{aligned}$$

The proof will be finished once we notice that

$$D_1^{-1} - D_2 = A_\lambda, \quad D_1^{-1} = B_\lambda, \quad (I-D_1D_2)^2 + D_1^2 = I. \ \blacksquare$$

Basing on Proposition 3.2 we can derive a simple formula for $V_\lambda^*V_\lambda$ in terms of U and U^*.

MAIN THEOREM 3.3. *Assume the sequence λ_n satisfies $\varepsilon \le \lambda_n \le 2-\varepsilon$ for any $n \ge 0$. Then*

$$I - V_\lambda^*V_\lambda = (B_\lambda + U^*A_\lambda)^{-1}(I-U^*U)(B_\lambda + A_\lambda U)^{-1},$$

where A_λ and B_λ are defined in (3.3). *In particular the relaxed algorithm is convergent for any sequence λ_n with $\varepsilon \le \lambda_n \le 2-\varepsilon$ if $U^*U = I$.*

Proof. Both operators A_λ and B_λ are bounded as soon as the coefficients λ_n stay away from 0 and 2. Moreover the operator $B_\lambda + A_\lambda U$ is invertible as

$$B_\lambda + A_\lambda U = B_\lambda(I + D_{1-\lambda_n}U), \qquad \|D_{1-\lambda_n}\| \le 1-\varepsilon < 1.$$

Notice that

$$B_\lambda^2 - A_\lambda^2 = I.$$

Therefore

$$\begin{aligned} V_\lambda^* V_\lambda &= (B_\lambda + U^*A_\lambda)^{-1}(A_\lambda + U^*B_\lambda)(A_\lambda + B_\lambda U)(B_\lambda + A_\lambda U)^{-1} \\ &= (B_\lambda + U^*A_\lambda)^{-1}[B_\lambda^2 + U^*A_\lambda^2 U + U^*A_\lambda B_\lambda + A_\lambda B_\lambda U + U^*U - I](B_\lambda + A_\lambda U)^{-1} \\ &= (B_\lambda + U^*A_\lambda)^{-1}[(B_\lambda + U^*A_\lambda)(B_\lambda + A_\lambda U) + U^*U - I](B_\lambda + A_\lambda U)^{-1} \\ &= I + (B_\lambda + U^*A_\lambda)^{-1}(U^*U - I)(B_\lambda + A_\lambda U)^{-1}. \end{aligned}$$

Finally, we get

$$I - V_\lambda^* V_\lambda = (B_\lambda + U^*A_\lambda)^{-1}(I - U^*U)(B_\lambda + A_\lambda U)^{-1}. \blacksquare$$

COROLLARY 3.4. *Assume* $0 < |\lambda_n - 1| < 1 - \varepsilon$ *for any* $n \geq 0$. *The relaxed algorithm is convergent if and only if* $U^*U = I$.

Proof. By (3.2) the operator V_λ is one-to-one as $\lambda_n \neq 1$. Assume the relaxed algorithm is convergent. Then V_λ is a partial isometry. Hence $V_\lambda^* V_\lambda = I$ as V_λ is one-to-one. By Theorem 3.3 we get $U^*U = I$. The converse implication is already included in Theorem 3.3. ∎

REMARK. The assumption $U^*U = I$ is stronger than U being a partial isometry. According to [3] it ensures that the Kaczmarz algorithm is convergent even if we drop finitely many vectors from the sequence $\{e_n\}_{n=0}^\infty$.

REMARK. The assumption $\varepsilon < \lambda_n < 2 - \varepsilon$ is necessary in general for convergence of the relaxed Kaczmarz algorithm. Indeed, assume the opposite, i.e. $|\lambda_{n_k} - 1| \to 1^-$ for an increasing subsequence $\{n_k\}_{k=1}^\infty$ of natural numbers. By extracting a subsequence we may assume

$$\sum_{k=1}^\infty (1 - |\lambda_{n_k} - 1|) < 1. \tag{3.4}$$

In particular we have $\lambda_{n_k} \neq 1$. In the two-dimensional space $\mathbb{C}^2$ let

$$e_n = \begin{cases} (1,0) & \text{for } n = n_k, \\ (0,1) & \text{for } n \neq n_k. \end{cases}$$

Then for $x = (1,0)$ we have

$$x_{n_l} = \Big[1 - \prod_{k=1}^{l}(1 - \lambda_{n_k})\Big] x.$$

But the product $\prod_{k=1}^\infty (1 - \lambda_{n_k})$ does not tend to zero under assumptions (3.4).

References

[1] O. Christensen, *An Introduction to Frames and Riesz Bases*, Birkhäuser, Boston, 2003.

[2] F. Deutsch and H. Hundal, *The rate of convergence for the method of alternating projections, II*, J. Math. Anal. Appl. 205 (1997), 381–405.

[3] R. Haller and R. Szwarc, *Kaczmarz algorithm in Hilbert space*, Studia Math. 169 (2005), 123–132.

[4] S. Kaczmarz, *Angenäherte Auflösung von Systemen linearer Gleichungen*, Bull. Int. Acad. Polon. Sci. Lett. Cl. Sci. Math. Natur. Sér. A Sci. Math. 1937, 355–357; English transl.: Internat. J. Control 57 (1993), 1269–1271.

[5] S. Kwapień and J. Mycielski, *On the Kaczmarz algorithm of approximation in infinite-dimensional spaces*, Studia Math. 148 (2001), 75–86.

[6] R. Szwarc, *Kaczmarz algorithm in Hilbert space and tight frames*, Appl. Comput. Harmonic Anal. 22 (2007), 382–385.

Ryszard Szwarc, Grzegorz Świderski
Institute of Mathematics
University of Wrocław
50-384 Wrocław, Poland
E-mail: szwarc2@gmail.com
gswider@math.uni.wroc.pl

Received December 7, 2012 (7709)