Applied Mathematics E-Notes, 17(2017), 142-156 © ISSN 1607-2510
Available free at mirror sites of http://www.math.nthu.edu.tw/∼amen/

The Block Kaczmarz Algorithm Based On Solving Linear Systems With Arrowhead Matrices*

Andrey Aleksandrovich Ivanov†, Aleksandr Ivanovich Zhdanov‡

Received 3 July 2016

Abstract

This paper proposes a new implementation of the block Kaczmarz algorithm for solving systems of linear equations by the least squares method. The implementation of the block algorithm is related to solving underdetermined linear systems. Each iteration of the proposed algorithm is considered as the solution of a sub-system defined by a specific arrowhead matrix. This sub-system is solved in an effective way using the direct projection method. In this work we demonstrate the existence of an optimal partition into equally-sized blocks for the test problem.

1 Introduction

The projection iterative algorithm was developed by Kaczmarz in 1937 [12], later various modifications of this method were used in many applications of signal and image processing. Each equation of the linear system can be interpreted as a hyperplane, and the solution of the consistent system can be interpreted as the point of intersection of these hyperplanes. The search for an approximate solution by the Kaczmarz algorithm is carried out in the directions perpendicular to these hyperplanes. Active discussion of this algorithm was stimulated by [16], which proposes a randomized version of this algorithm and estimates its rate of convergence. The main purpose of the randomization of projection method is to provide the speed of convergence regardless of the number of equations in the system [17]. The block Kaczmarz modification was developed with the study of the convergence of the random projection algorithm [7, 18]. The idea of randomization for a block algorithm is discussed in [2, 15, 20]. From the geometric point of view, the projection is not made onto the hyperplane, but on intersection of several hyperplanes.

The block algorithm implementation is related with the least squares solution of an underdetermined system of linear algebraic equations in each iteration. This problem can be solved using the Moore–Penrose pseudo-inverse. This is a very computationally complex problem. In this paper, we show that each iteration of the block algorithm is

*Mathematics Subject Classifications: 65F10, 65F05, 65F20.

†Department of Applied Mathematics and Informatics, Samara State Technical University, Samara 443120, Russia

‡Department of Applied Mathematics and Informatics, Samara State Technical University, Samara 443120, Russia

equivalent to solving a system of equations with a generalized arrowhead matrix. To solve this system, we propose a special modification of the direct projection method [4, 22, 23].

2 Related Results

There are two large groups of numerical methods for solving systems of linear algebraic equations (SLAEs)

$$Au = f, \quad A \in \mathbb{R}^{m\times n}, \quad u \in \mathbb{R}^n, \quad f \in \mathbb{R}^m.$$

There are the direct and iterative methods in computational mathematics. Direct methods for solving linear systems can be classified as follows [6]:

- Orthogonal direct methods (for the case of $m = n$), characterized by the fact that the main stages of the method use only orthogonal transformations which do not change the conditionality of the computational problem, and thus are computationally stable.
- Methods for solving the system of normal equations, the initial stage of which is to transform the original system to an equivalent one, in the sense of the least squares method, using Gauss' left transformation. And the condition number of the equivalent system increases greatly. The Cholesky decomposition is mainly used for the solution of these systems.
- Methods of the augmented matrix, the basic idea of which is to transform an SLAEs into an equivalent augmented consistent system.

As an augmented SLAEs, in this class, the following system is often used [5, 9]

$$\begin{pmatrix} I_{m\times m} & A \\ A^T & O_{n\times n} \end{pmatrix} \begin{pmatrix} y \\ u \end{pmatrix} = \begin{pmatrix} f \\ O_n \end{pmatrix} \Leftrightarrow \hat{A}\hat{u} = \hat{f}. \tag{1}$$

The main drawback of this SLAEs is that the spectral condition number of its matrix is estimated as $k_2\left(\hat{A}\right) = O\left(k_2^2\left(A\right)\right)$, for example [14]. This fact may have a catastrophic effect on the computational stability in floating-point arithmetic for most of the computational methods if the minimal singular value of A is small. Another example of an equivalent system (1) can be regarded as an augmented system from [1]

$$\begin{pmatrix} I_{m\times m} & -A \\ A^T & O_{n\times n} \end{pmatrix} \begin{pmatrix} y \\ u \end{pmatrix} = \begin{pmatrix} O_m \\ A^T f \end{pmatrix}. \tag{2}$$

This system has no advantages over (1), but a class of direct methods has been designed to solve it, such as special modifications of the algorithms of the so-called ABS class[1]. It should be noted that the system (2) can be considered as one of the classes of so-called equilibrium systems, the solution of which is important for a variety of applications,

[1] It's the acronym contains the initials of Jozsef Abaffy, Charles G. Broyden and Emilio Spedicato.

such as [19]. It also points to the often insurmountable difficulties with computational stability when implementing the known direct algorithms for solving such systems. The expediency of iterative SOR (successive over-relaxation) methods is demonstrated in [8] for solving systems of the type (1) and (2).

In [24] there is offered an augmented system of the type

$$\begin{pmatrix} \omega I_{m\times m} & A \\ A^T & -\omega I_{n\times n} \end{pmatrix} \begin{pmatrix} y \\ u \end{pmatrix} = \begin{pmatrix} f \\ O_n \end{pmatrix} \Leftrightarrow \hat{A}\hat{u} = \hat{f}, \tag{3}$$

where, if both $\omega^2 < \varepsilon_{mach}$ and $\omega > \varepsilon_{mach}$, it is proved that $k_2\left(\hat{A}\right) = k_2\left(A\right)$ in floating-point arithmetic. The disadvantages of this system include the fact that the matrix of the system is semi-definite. In [10] there is an effective modification of the projection iterative algorithm of Kaczmarz for solving this system.

It should be noted that iterative methods, which converge in a finite number of iterations, can be considered part of the group of direct methods for solving linear systems, so the difference between the direct and iterative methods for some mathematical results may not be hard and fast. For example, in [21], it is stated that the classical iterative method of Kaczmarz for a system with a row-orthogonal matrix converges in a finite number of iterations, and each u_k is a solution of the first k equations SLAEs, where $k = 1, 2, \ldots, n$. The authors of that work offer a recursive modification of Kaczmarz's method; its iterative process includes the stages of the reduction of the matrix of the original system to row-orthogonal form by implicit decomposition into a product of a lower triangular and an orthogonal matrix (the LQ decomposition). This idea is also implemented in one of the ABS methods–Huang's method [1]; the same method can be successfully applied to the solution of the system (2).

According to the classification of Abaffi and Spedikatto [1], there are other three classes of direct methods for solving SLAEs:

- The first class consists of methods whose structure is such that during solution, the initial system is transformed into a system with smaller rank, for example, in Gaussian's elimination, every step eliminates one equation of the system.

- The second class includes methods that convert the matrix of the original SLAEs to another matrix. Solving SLAEs with the other matrix is reached in a simple way, for example, by reduction of the matrix to triangular or diagonal form.

- A third class of methods is such that the matrix of the system does not change, but some supporting matrix is modified. Examples of this class include ABS algorithms.

The so-called direct projection method [4] relates to the third class. It is based on the Sherman–Morrison formula and it is equal to Gaussian elimination has been proved [22]. It is important to note that a computationally stable implementation of the direct projection method requires the choice of the leading element (or pivot), because of this equivalence.

3 Direct Projection Method

Consider a system of linear algebraic equations

$$Au = f,\ A \in \mathbb{R}^{n\times n},\ u \in \mathbb{R}^n,\ f \in \mathbb{R}^n,\ \det(A) \neq 0. \tag{4}$$

Obviously, the first k equations of the system (4) can be written as

$$\begin{pmatrix} \bar{A}_{k,k} & \bar{A}_{k,n-k} \end{pmatrix} \begin{pmatrix} \bar{u}_k \\ \bar{u}_{n-k} \end{pmatrix} = \bar{f}_k \ \Rightarrow\ \bar{A}_{k,k}\bar{u}_k + \bar{A}_{k,n-k}\bar{u}_{n-k} = \bar{f}_k,$$

where $\bar{A}_{k,k} \in \mathbb{R}^{k\times k}$ are the minors of the matrix A of order k, $\bar{u}_k = (u_1, u_2, \ldots, u_k)$, $\bar{f}_k = (f_1, f_2, \ldots, f_k)$ and $k = 1, 2, \ldots, n$. Then, the direct projection method is determined by the following recurrent procedure [22, 23]:

$$\begin{aligned} u^0 = \theta_n,\ P_0 = I_{n\times n},\ \delta_{k+1} = a_{k+1}^T P_k e_{k+1}, \\ u_{k+1} = u_k + P_k e_{k+1}\left(f_{k+1} - a_{k+1}^T u_k\right)\delta_{k+1}^{-1}, \\ P_{k+1} = P_k - \left(P_k e_{k+1} a_{k+1}^T P_k\right)\delta_{k+1}^{-1}, \end{aligned} \tag{5}$$

where θ_n is the n-dimensional zero vector, $I_{n\times n}$ is the $n\times n$ identity matrix, $[\cdot]^T$ signifies the transpose, a_{k+1}^T is a row of the matrix A, and $k = 0, 1, \ldots, n-1$. To successfully complete this method, as with Gauss' method, it is necessary that all the principal minors of the matrix A be nonsingular, i.e.,

$$\det\left(\bar{A}_{k,k}\right) \neq 0,\ \forall k = 1, 2, \ldots, n. \tag{6}$$

If special structure of the matrix P_k is ignored, the number of arithmetic operations of the direct projection method is estimated to be $O\left(2n^3\right)$, but [22] proposed that P_k for $k = 1, 2, \ldots, n$

$$P_k = \begin{pmatrix} O_{k\times k} & -\bar{A}_{k,k}^{-1}\bar{A}_{k,n-k} \\ O_{(n-k)\times k} & I_{(n-k)\times(n-k)} \end{pmatrix}, \tag{7}$$

where $P_k \in \mathbb{R}^{n\times n}$, $O_{k\times k}$ is a $k \times k$ zero matrix. We further give more details about the direct projection method referred at [22].

THEOREM 1 ([22]). Let us consider a matrix

$$A_k = \begin{pmatrix} \bar{A}_{k,k} & \bar{A}_{k,n-k} \\ O_{(n-k)\times k} & O_{(n-k)\times(n-k)} \end{pmatrix},$$

where $\bar{A}_{k,k} \in \mathbb{R}^{k\times k}$, $k < n$ and $rank\left(\bar{A}_{k,k}\right) = k$.

Then the limit

$$P_{A_k} = \lim_{\alpha\to 0}\left(A_k + \alpha I_{n\times n}\right)^{-1}\alpha$$

for all $|\alpha| < \sigma_{min}\left(\bar{A}_{k,k}\right)$ exist and $P_{A_k} = P_k$.

THEOREM 2 ([22]). Let's assume that all principal minors of matrix A are nonsingular, i.e. $\det\left(\bar{A}_{i,i}\right) \neq 0,\ \forall i = 1, 2, \ldots, n$, and let us consider the recurrence equations

$$u_{k+1} = u_k + g_{k+1}^k \left(f_{k+1} - a_{k+1}^T u_k\right) \delta_{k+1}^{-1}, u^0 = \theta_n,$$

$$P_{k+1} = P_k - \left(g_{k+1}^k a_{k+1}^T P_k\right) \delta_{k+1}^{-1},\ P_0 = I_{n \times n},$$

where

$$P_k = \begin{pmatrix} g_1^k & \ldots & g_n^k \end{pmatrix} \in \mathbb{R}^{n \times n},\ \ \delta_{k+1} = a_{k+1}^T g_{k+1}^k \ \text{ for } k = 0, 1, \ldots, n-1.$$

Then the (6) is a necessary and sufficient condition for $\delta_{k+1} \neq 0,\ \forall k = 0, 1, \ldots, n-1$ and wherein the $u_n = A^{-1} f$ is a solution of system (4). The proof of this theorem is demonstrated in [22] and uses a famous Sherman–Morrison formula and Theorem 1.

The first k of columns of the matrix P_k are zero, so their use in arithmetic calculations is redundant, and finally, the number of multiplication needed for the direct projection method is estimated by $O\left(\frac{2}{3} n^3\right)$, which is similar to the complexity of Gaussian elimination, for example [9]. Assuming that A is dense, for P_k, on each iteration, it is enough to store an array of $(n-k)(k+1)$ elements. It is important to note that solving problems with the same matrix A and various right parts, it may be appropriate to calculate and store all the P_k in memory before the main computational process. In this case, the complexity of the solution of the problem can be reduced to $O\left(\frac{2}{3} n^2\right)$.

See Table 2 in the Conclusion, for a definition of this algorithm in MATLAB language and more see in [11].

4 The Block Kaczmarz Algorithm

Consider a consistent SLAEs

$$Au = f,\ A \in \mathbb{R}^{m \times n},\ u \in \mathbb{R}^n,\ f \in \mathbb{R}^m,\ m \geq n$$

and its block form

$$A = \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_p \end{pmatrix},\quad f = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_p \end{pmatrix},\quad B_i \in \mathbb{R}^{l \times n},\ d_i \in \mathbb{R}^l, \tag{8}$$

where p is the number of blocks, l is the number of rows in the block B_i, $m = l \cdot p$, and

$$B_i = \begin{pmatrix} a_{(i-1)\cdot l+1}^T \\ a_{(i-1)\cdot l+2}^T \\ \vdots \\ a_{(i-1)\cdot l+l}^T \end{pmatrix} = \left(b_{i,1}, b_{i,2}, \ldots, b_{i,n}\right),\ d_i = \begin{pmatrix} f_{(i-1)\cdot l+1} \\ f_{(i-1)\cdot l+2} \\ \vdots \\ f_{(i-1)\cdot l+l} \end{pmatrix},\ i = 1, 2, \ldots, p,$$

where $b_{i,j} \in \mathbb{R}^l$ is the column of the matrix B_i and $j = 1, 2, \ldots, n$.

Then the block Kaczmarz algorithm can be written in the form [18, 7]

$$u^{k+1} = u^k - B^+_{j(k)} \left(B_{j(k)} u^k - d_{j(k)} \right), \tag{9}$$

$$j\,(k) = k \mod p + 1,$$

where $B^+_{j(k)}$ is the Moore-Penrose pseudo-inverse of $B_{j(k)}$ and $k = 0, 1, 2 \ldots, j\,(k) = 1,2, \ldots, p, 1, 2, \ldots, p, \ldots$ for the circle control scheme.

We'll show that one iteration of this type is equivalent to the solution of SLAEs with an arrowhead matrix

$$\begin{pmatrix} I_{n\times n} & B^T_{j(k)} \\ B_{j(k)} & O_{l\times l} \end{pmatrix} \begin{pmatrix} u^{k+1} \\ y^{k+1} \end{pmatrix} = \begin{pmatrix} u^k \\ d_{j(k)} \end{pmatrix},\ j\,(k) = k \bmod p + 1. \tag{10}$$

THEOREM 3. The iterations of the form (9) are equivalent to iterations of the form (10).

PROOF. We write (10) as a system of two equations

$$\begin{cases} u^{k+1} + B^T_{j(k)} y^{k+1} = u^k, \\ B_{j(k)} u^{k+1} = d_{j(k)}. \end{cases} \tag{11}$$

Eliminating $u^{k+1} = u^k - B^T_{j(k)} y^{k+1}$ in (11), we get

$$-B_{j(k)} B^T_{j(k)} y^{k+1} = d_{j(k)} - B_{j(k)} u^k. \tag{12}$$

We multiply the left and right parts from the left (12) by the matrix $B^+_{j(k)}$, and get

$$-B^+_{j(k)} B_{j(k)} B^T_{j(k)} y^{k+1} = B^+_{j(k)} \left(d_{j(k)} - B_{j(k)} u^k \right). \tag{13}$$

From the Moore-Penrose conditions

$$B_{j(k)} B^+_{j(k)} B_{j(k)} = B_{j(k)} \text{ and } B^+_{j(k)} B_{j(k)} = \left(B^+_{j(k)} B_{j(k)} \right)^T,$$

from which it follows directly that

$$B^T_{j(k)} = \left(B_{j(k)} B^+_{j(k)} B_{j(k)} \right)^T = \left(B^+_{j(k)} B_{j(k)} \right)^T B^T_{j(k)} = B^+_{j(k)} B_{j(k)} B^T_{j(k)}.$$

So, (13) can be written in the form

$$-B^T_{j(k)} y^{k+1} = B^+_{j(k)} \left(d_{j(k)} - B_{j(k)} u^k \right). \tag{14}$$

The first equation of the system (11) is $B^T_{j(k)} y^{k+1} = u^k - u^{k+1}$, and in virtue of (14), we get the final recurrent equation

$$u^{k+1} = u^k - B^+_{j(k)} \left(B_{j(k)} u^k - d_{j(k)} \right).$$

Our proof is obvious enough. Note that the representation (10) was given in [13] for the case when the dimension of the block $l = 1$.

Consider the matrix of the system (10)

$$\hat{B}_k = \begin{pmatrix} I_{n\times n} & B_k^T \\ B_k & O_{l\times l} \end{pmatrix},$$

for which, for example [14], the spectral condition number is

$$k_2\left(\hat{B}_k\right) = \frac{\frac{1}{2} + \sqrt{\frac{1}{4} + \sigma^2_{max}\left(B_k\right)}}{-\frac{1}{2} + \sqrt{\frac{1}{4} + \sigma^2_{min}\left(B_k\right)}}.$$

It should be noted that for small values of the condition number $k_2\left(\hat{B}_k\right)$, it may be advantageous to use iterative methods, for example [8], to solve the system (10).

5 Proposed Algorithm

For solving SLAEs (10) at each iteration it is proposed to use the direct projection method. Consider the first n of iterations of the direct projection method for solving (10):

$$u_{i+1}^{k+1} = u_i^{k+1} + P_i^{j(k)} e_{i+1}\left(u^k\left(i+1\right) - u_i^{k+1}\left(i+1\right) - b^T_{j(k),i+1} y_i^{k+1}\right)\left(\delta_{i+1}^{j(k)}\right)^{-1}, \quad (15)$$

where $i = 0, 1, \ldots, n-1$, $b^T_{j(k),i+1}$ is the transposed $i+1$th column of $B_{j(k)}$, and $u^k\left(i+1\right)$ is the $i+1$th component of the vector u^k. Please note that in this section, we use the i-index as a counter for the direct projection algorithm iterations and k-index as a counter for Kaczmarz steps.

THEOREM 4. Suppose that the components of the vector of the initial approximation satisfy the condition of consistency

$$u_0^{k+1} = u^k - B^T_{j(k)} y_0^{k+1}, \quad (16)$$

then

$$u_{i+1}^{k+1} \equiv u_i^{k+1},\ i = 0, 1, \ldots, n-1.$$

PROOF. The proof of this theorem is omitted in this article: it may be easily carried out, for example, by the method of mathematical induction.

It should be noted that a similar trick was also used in the context of the total least square problem, for more, see in [23]. As a result, when the matching conditions (16) are fulfilled, the first n iterations of the direct projection method for solving (10) can be omitted. From (7) and the fact that the main minor of the matrix of the system (10) of order n is the identity matrix, one has

$$P_n = \begin{pmatrix} O_{n\times n} & -B_{j(k)}^T \\ O_{l\times n} & I_{l\times l} \end{pmatrix}.$$

Consider the ith iteration of the projection method ($i = n, n+1, \dots, n+l-1$). Then

$$u_{i+1}^{k+1} = u_i^{k+1} + P_i^{j(k)} e_{i+1} \left(d_{j(k)} (i+1-n) - a_{(j(k)-1)\cdot l+i-n+1}^T u_i^{k+1} \right) \left(\delta_{i+1}^{j(k)} \right)^{-1}. \quad (17)$$

Noting that the first n iterations are redundant, we introduce a new iteration numbering of the direct projection method:

$$\omega = i - n = 0, 1, \dots, l-1,\ i = n, n+1, \dots, n+l-1.$$

Then the direct projection method for the solution of (10) is

$$u_0^{k+1} = u^k,\ y_0^{k+1} = \theta_l,\ P_0 = \begin{pmatrix} O_{n\times n} & -B_{j(k)}^T \\ O_{l\times n} & I_{l\times l} \end{pmatrix},$$

$$\delta_{\omega+1}^{j(k)} = \begin{pmatrix} a_{(j(k)-1)\cdot l+\omega+1}^T & \theta_l^T \end{pmatrix} P_\omega e_{n+\omega+1},$$

$$u_{\omega+1}^{k+1} = u_\omega^{k+1} + P_\omega^{j(k)} e_{n+\omega+1} d_{j(k)} (\omega+1) - a_{(j(k)-1)\cdot l+\omega+1}^T u_\omega^{k+1} \left(\delta_{\omega+1}^{j(k)} \right)^{-1},$$

$$P_{\omega+1} = P_\omega - P_\omega e_{n+\omega+1} \begin{pmatrix} a_{(j(k)-1)\cdot l+\omega+1}^T & \theta_l^T \end{pmatrix} P_\omega \left(\delta_{\omega+1}^{j(k)} \right)^{-1}, \quad (18)$$

$$\omega = 0, 1, \dots, l-1.$$

Upon completion of the iteration of the direct projection method, one needs to put $u_l^{k+1} = u^{k+1}$.

The complexity of the solution of (10), taking into account the special structure of the matrix P_i, can be estimated as $O\left(\frac{1}{6} l^3 + n^2 l\right)$, where l is the number of rows in the block $B_i \in \mathbb{R}^{l\times n}$, $i = 1, 2, \dots, p$.

6 Limitations of the Proposed Algorithm

The algorithm, in theory, was proposed for only consistent and overdetermined systems. But in practice, this algorithm could apply for solving inconsistent systems also, especially in the randomized form. If we want to solve underdetermined systems we can apply the pre-conversion to the regularized augmented system (3) with the small value of the regularization parameter, for more, see in [24].

7 Numerical Example

Consider the matrix $B_k = \{a_{i,j}\} \in \mathbb{R}^{l\times n}$, where $a_{i,j} \sim U\left(-\sqrt{\frac{3}{n}}, \sqrt{\frac{3}{n}}\right)$ has a continuous uniform distribution on the segment $\left[-\sqrt{\frac{3}{n}}, \sqrt{\frac{3}{n}}\right]$, and $E\left(a_{i,j}\right) = 0$, $E\left(a_{i,j}^2\right) = \frac{1}{n}$, $l \leq n$, $rank\left(B_k\right) = l$, $E\left\|a^T_{(k-1)\cdot l+1}\right\| = 1$,

$$k = 1, 2, \dots p, \quad i = (k-1)\,l + 1, (k-1)\,l + 2, \dots, (k-1)\,l + l \text{ and } j = 1, 2, \dots n.$$

Notice that the condition $l \leq n$ is used only for this particular task, and in general it is not a limitation of the proposed algorithm. Then consider the matrix $A \in \mathbb{R}^{m\times n}$, whose block rows are $B_k \in \mathbb{R}^{l\times n}$, $k = 1, 2, \dots, p$, $m = l \cdot p$, as in (8), and which satisfy the above mentioned conditions. In [3] (see the limit conditions of its Theorem 2), estimates are made for the minimum and maximum eigenvalues of $S_k = B_k B_k^T$

$$\lim \lambda_{min}\left(S_k\right) = \left(1 - \sqrt{\frac{l}{n}}\right)^2, \ \lim \lambda_{max}\left(S_k\right) = \left(1 + \sqrt{\frac{l}{n}}\right)^2. \tag{19}$$

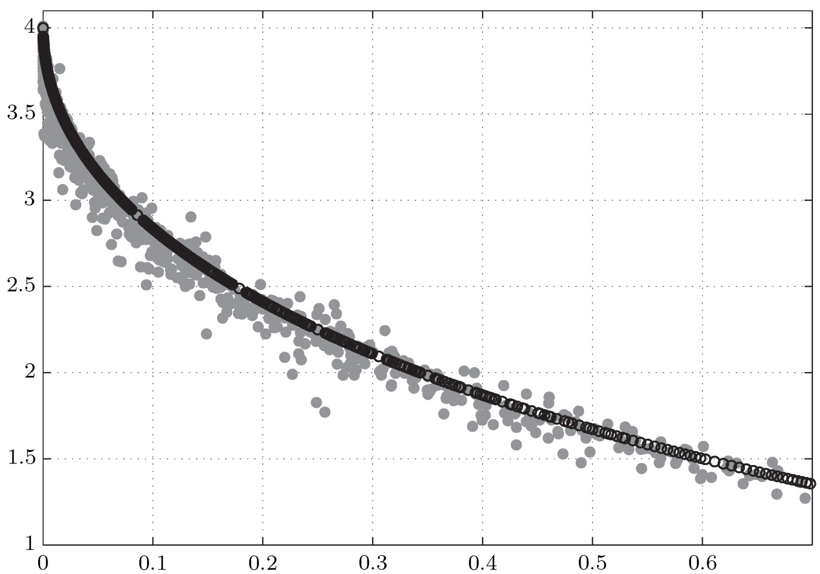

Figure 1: Black points: on the abscissa axis is $\left(1 - \sqrt{\frac{l}{n}}\right)^2$, on the ordinate axis is $\left(1 + \sqrt{\frac{l}{n}}\right)^2$. Gray points: on the abscissa axis is $\lambda_{min}\left(S_k\right)$, on the ordinate axis is $\lambda_{max}\left(S_k\right)$. The figure has been plotted for all values $n = 10, 11, \dots, 400$ and $l = 10, 11, \dots n$.

We give Fig. 1 as an illustration of these limit equations.Considering (19), an approximate estimate is $\|A\|^2 \approx 1 + \sqrt{\frac{m}{n}}$ and

$$\sigma_{min}^2(B_k) \approx \left(1 - \sqrt{\frac{l}{n}}\right)^2, \sigma_{max}^2(B_k) \approx \left(1 + \sqrt{\frac{l}{n}}\right)^2. \tag{20}$$

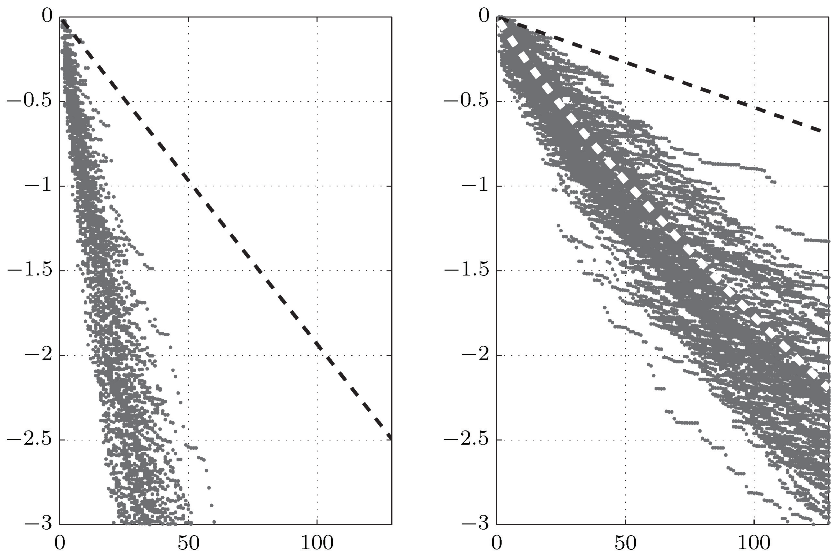

Figure 2: Dependence of $\log_{10}\left\|u^* - u^k\right\|^2$ from the iteration number k (on the abscissa axis) for 100 realizations. The dotted black line shows the error estimate based on (21). The dotted white line shows the exact error based on [2].

Right-hand side vector of the test system of equations can be built by any method ensuring the consistency of the system, for example, if

$$f = Au^*, \text{ where } u^* = (u_1, u_2, \dots, u_n)$$

such as $\|u^*\|^2 = 1$. As a stopping criterion for the iterative algorithm, we will use the rule $\left\|u^* - u^{K_{stop}}\right\|^2 \leq 10^{-q}$, where K_{stop} is a stopping number. Note that the estimate of the convergence rate that is used for a system of this type is referred at [15], and take into consideration that the authors treat a consistent system of linear algebraic equations, and the initial approximation $u^0 = \theta$

$$E\left\|u^* - u^k\right\|^2 \leq \left(1 - \frac{\sigma_{min}^2(A)}{\beta p}\right)^k \|u^*\|^2, \tag{21}$$

where $\max\left\{\sigma_{max}^2(B_j)\right\}_{j=1}^p \leq \beta$, and the selection of the block in each iteration is determined randomly: $P\left(j\left(k\right)=i\right)=\frac{1}{p}$, $i=1,2,\ldots,p$. It follows that

$$\tilde{K}_{stop}=\frac{-q-\log_{10}\left\|u^*\right\|^2}{\log_{10}\left(1-\frac{\sigma_{min}^2(A)}{\beta p}\right)}. \tag{22}$$

A software implementation of these numerical simulations is presented in [11]. The logarithm of the error versus the iteration number is plotted for $p=182$ in Fig. 2, from which the pessimistic estimate (21) is obvious for this problem.

Conclusion

Analyzing the results of the numerical experiment in Table 1, it should be noted that the optimal number of blocks for the research task, in terms of the total time of the algorithm, is $p=182$. As noted above, the complexity of each iteration of the proposed

l	1	2	**4**	7	8	13	$\cdots$	364
K_{stop}	109032	57020	**30247**	16244	13980	8750	$\cdots$	118
β	1.09	1.13	**1.18**	1.24	1.26	1.34	$\cdots$	3.4
$\tilde{K}_{stop}$	368952	190996	**100204**	60298	98015	34977	$\cdots$	3149
p	728	364	**182**	104	91	56	$\cdots$	2
$Time$	11.0	8.95	**8.88**	9.46	9.31	11.23	$\cdots$	54.8

Table 1: Results averaged over 100 realizations of the numerical example for $m=728$ and $n=512$, where $Time$ is the average time of the algorithm in seconds until reaching the prescribed accuracy when $q=8$. The equation (20) was used to estimate the parameter β.

algorithm is estimated as $O\left(\frac{1}{6}l^3+n^2l\right)$, from which it can be concluded that it is appropriate to choose $p=\frac{m}{l}$ such that $l<n$. In this way, we maintain a quadratic growth of the complexity of the calculations per iteration. It is important to note that the system of linear equations with the same matrix $B_{j(k)}$ and various right-hand parts is solved at each iteration, therefore, it is necessary to carry out a direct computation of P_ω only once for each matrix $B_{j(k)}$, and they can then be saved in memory for reuse. This trick will significantly reduce the number of arithmetic operations in each iteration, but the requirements for RAM space will increase substantially.

In the present paper, one iteration of Kaczmarz's block algorithm has been presented for the task of solving a system of linear equations with a special arrowhead matrix. But not only a direct projection method can be employed to solve such a system, so can, for example, Huang's algorithm [1]. The particular interest for computational mathematics is the use of a direct projection method for solving linear systems with an arrowhead matrix.

Remarks. According to the numerical experiment, the obtained estimate of the number of iterations before stopping, using the expression (22), is too pessimistic; in [2], the authors note this fact and offer an accurate estimate for $\mathrm{E}\left\|u^*-u^k\right\|^2$. It would be extremely useful to generalize the results of [2] to the block randomized algorithm: in this case, it is analytically possible to obtain the optimum value for p using the estimate of the algorithmic complexity.

The developed algorithm in its randomized modification, as in [16], taking into account the results of [15, 20], can be applied also to the inconsistent case.

It should be noted that the direct projection method may have an effective implementation for solutions of augmented systems of the form (1), (2), and (3). Just to show the trick for each of these systems, that is used in this paper (see the Theorem 4). The trick allow us to accept that one part of the augmented system is redundant. It can be the object of further research, particularly in the context of a direct projection method with pivoting.

Acknowledgment. Our special thanks to Prof. G. A. Kalyabin for English proofreading and editing.

Projection methods	
`dpmsolve`	Direct Projection Method (DPM) as in (5).
`pblockkaczmarz`	Block Kaczmarz Algorthm based on solving linear systems with arrowhead matrices. Cyclic Control.
`randpblockkaczmarz`	Same as above, but with randomized control schemes from [15, 16].
Demos	
`dpmsolve_demo`	Sample for DPM with pivoting.
`pblockkaczmarz_demo`	Sample contains comments for equivalence Theorem 3.
`randpblockkaczmarz_demo`	The demo for standardized matrix. Example plotted in Fig. 1.

Table 2: Overview of MATLAB Package [11].

References

[1] J. Abaffy and E. Spedicato, ABS Projection Algorithms: Mathematical Techniques for Linear and Nonlinear Equations, Prentice-Hall, Inc., 1989.

[2] A. Agaskar, C. Wang and Y. M. Lu, Randomized Kaczmarz algorithms: Exact MSE analysis and optimal sampling probabilities, In Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on, 389–393.

[3] Z. Bai and Y. Q. Yin, Limit of the smallest eigenvalue of a large dimensional sample covariance matrix, Ann. Probab., 21(1993), 1275–1294.

[4] M. Benzi and C. D. Meyer, A direct projection method for sparse linear systems, SIAM J. Sci. Comput., 16(1995), 1159–1176.

[5] A. Björck, Numerical methods for least squares problems. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996. xviii+408 pp.

[6] I. S. Duff and J. K. Reid, A comparison of some methods for the solution of sparse overdetermined systems of linear equations, IMA J. Appl. Math., 17(1976), 267–280.

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

[8] G. H. Golub, X. Wu and J. Y. Yuan, SOR-like methods for augmented systems, BIT, 41(2001), 71–85.

[9] G. H. Golub and C. F. Van Loan, Matrix Computations (Vol. 3), JHU Press, 2012.

[10] A. A. Ivanov and A.I. Zhdanov, Kaczmarz algorithm for Tikhonov regularization problem, Appl. Math. E-Notes, 13(2013), 270–276.

[11] A. A. Ivanov, Regularization Kaczmarz Tools Version 1.4 for Matlab, Matlabcentral Fileexchange, URL: http://www.mathworks.com/matlabcentral/fileexchange/43791.

[12] S. Kaczmarz, Angenäherte auflösung von systemen linearer gleichungen, Bulletin International de l'Academie Polonaise des Sciences et des Lettres, 35(1937), 355–357.

[13] A. Kushnerov, A seminumerical transient analysis of switched capacitor converters, IEEE Journal of Emerging and Selected Topics in Power Electronics, 2(2014), 814–820.

[14] A. Nicola and C. Popa, Kaczmarz extended versus augmented system solution in image reconstruction, ROMAI J, 6(2010), 145–152.

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

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

[17] K. Sabelfeld and N. Loshchina, Stochastic iterative projection methods for large linear systems, Monte Carlo Methods Appl., 16(2010), 343–359.

[18] G. P. Vasil'chenko and A. A. Svetlakov, A projection algorithm for solving systems of linear algebraic equations of high dimensionality, Comput. Math. Math. Phys., 20(1980), 1–8.

[19] S. A. Vavasis, Stable numerical algorithms for equilibrium systems, SIAM J. Matrix Anal. Appl., 15(1994), 1108–1131.

[20] C. Wang, A. Agaskar and Y. M. Lu, Randomized Kaczmarz Algorithm for Inconsistent Linear Systems: An Exact MSE Analysis, arXiv preprint arXiv:1502.00190 (2015).

[21] X. Yu, N. K. Loh and W.C. Miller, New recursive algorithm for solving linear algebraic equations, Electronics Letters, 28(1992), 2069–2071.

[22] A. I. Zhdanov, Direct sequence method for solving systems of linear algebraic equations, Proceedings of the Russian Academy of Sciences, 356(1997), 442–444.

[23] A. I. Zhdanov and P. A. Shamarov, A Direct Projection Method in the Problem of Complete Least Squares, Automation and Remote Control C/C of Avtomatika i Telemekhanika, 61(2000), 610–620.

[24] A. I. Zhdanov, The method of augmented regularized normal equations, Comput. Math. Math. Phys., 52(2012), 194–197.