Mathematical Analysis & Convex Optimization

Vol. 5 (2024), No. 2, 101-108
https:\\ maco.lu.ac.ir
DOI: 10.22034/maco.5.2.10

Research Paper

ALGEBRAIC VERSIONS OF TWO GEOMETRIC METHODS FOR SOLVING SYSTEMS OF LINEAR EQUATIONS

N. NAJAFI AND M. GHASEMI KAMALVAND*

Abstract. In this article, we obtain algebraic versions of algorithms NGHK and KL (two geometric methods for solving systems of linear equations), that is, we show that in these two algorithms, the vector sequences that converge to the solution of the system of linear equations are located in the row space of the coefficients matrix of system. In the following, we will compare these algorithms with the Gauss-Seidel method by providing a few examples.

MSC(2010): 65F10.
Keywords: systems of linear equations, row space of matrix, geometric methods for solving systems of linear equations.

1. Introduction and Background

Solving systems of linear equations is one of the most important topics in mathematics. There are two direct and iterative methods for solving systems (see [2], [3], [5], [6], [7], [8], [9] and [10]). In [1], a method for solving these systems is presented, and in [4], the authors introduced another method inspired by that approach and examined its advantages using geometric and numerical techniques. For a system of linear equations $Ax = b$, when the matrix A is nonsingular, we know that the solution lies in the row space of A. Therefore, the solution can be written as a linear combination of the rows of A. Considering this, it follows that by constructing a sequence of linear combinations of the rows of A and choosing the coefficients appropriately, one can generate a sequence that converges to the solution of the system $Ax = b$. This is achieved using the method described in [4].

In the next section, we briefly outline the methods presented in [4]. In the final sections, we introduce our proposed methods and provide the corresponding algorithms, concluding with several numerical examples.

2. NGHK and KL method

In [4], we presented a geometric method for solving systems of linear equations, which we call it method NGHK in this article, in this section, we briefly recall this method. To find the solution of systems of linear equations $Ax = b$, where $A = (a_{ij})_{n\times n}$ is a non-singular matrix and $b = (b_1, b_2, b_3, \ldots, b_n)^T$ is an n vector, we imagine every linear equation as a plane and we find the closest point on the plane 1 from an arbitrary point p_1 and call it p_2, now we find the closest point on the plane 2 from point p_2 and call it p_3, by continuing this process,

Date: September 17, 2024, Accepted: December 13, 2024.
*Corresponding author.

we reach the intersection of all planes, which is the solution of the system of linear equation $Ax = b$. Below we describe the algorithm of this method which is mentioned in [4].

Therefore, we will call the closest point in plane $a_{i1}x_1 + a_{i2}x_2 + a_{i3}x_3 + \cdots + a_{in}x_n = b_i$ from point $p_i = (p_{i1}, p_{i2}, p_{i3}, \ldots, p_{in})^T$, as

$$p_{i+1} = (p_{i+1,1}, p_{i+1,2}, p_{i+1,3}, \ldots, p_{i+1,n})^T,$$

we will have

$$p_{i+1,j} = a_{ij}\frac{b_i - \sum_{j=1}^{n} a_{ij}p_{ij}}{\sum_{j=1}^{n} a_{ij}^2} + p_{ij},\ 1 \le j \le n. \tag{2.1}$$

From these relations, the following algorithm can be concluded to find the solution of the linear equations system.

Algorithm 1: Algorithm NGHK Method
Input: A, b, $p = (p_1, p_2, p_3, \cdots, p_n)$, e, n.
$e_1 = 1$, $k = 0$
whille $e_1 < e$
$j = \overline{k} + 1$ ($\overline{k}$ is the remainder of dividing k by n.)
$s_1 = \sum_{u=1}^{n} a_{ju}p_u$
$s_1 = b_j - s_1$
$s_2 = \sum_{w=1}^{n} a_{jw}^2$
for $i = 1$ *to* n
$q_i = a_{ji}(s_1/s_2) + p_i$
end
$e_1 = ||q - p||/||p||$, $(q = (q_1, q_2, q_3, \cdots, q_n))$
$p = q$
$k = k + 1$
end
Output: p, e_1, k.
$||.||$ means Euclidean norm.

In [1] and [4], another method called KL was introduced, the difference with NGHK was that we found the closest points to p_1 in all the plans of the system and called the average of these points p_2, continuing this process to the following algorithm we were arriving.

Algorithm 2: Algorithm KL Method
Input: A, b, $p_1 = (p_{11}, p_{12}, p_{13}, \ldots, p_{1n})$, e.
$e_1 = 1$, $k = 1$
whille $e_1 < e$
for $j = 1$ *to* n
$s_1 = \sum_{u=1}^{n} a_{ju}p_{ku}$
$s_1 = b_j - s_1$
$s_2 = \sum_{w=1}^{n} a_{jw}^2$
for $i = 1$ *to* n
$q_{ji} = a_{ji}(s_1/s_2) + p_{ki}$
end

end
for $j = 1$ *to* n
$s_3 = \sum_{i=1}^{n} q_{ij}$
$p_{k+1,j} = s_3/n$
end
$p_{k+1} = (p_{k+1,1}, p_{k+1,2}, p_{k+1,3}, \ldots, p_{k+1,n})$
$p = p_{k+1}$
$e_1 = ||p - p_k||/||p_k||$
$k = k + 1$
end
Output: p, e_1, k.

3. **Algebraic versions of algorithms**

In this section, we find an algorithm to find the solution of the system of linear equations $Ax = b$, so that this solution is given as a linear combination of the rows of the matrix A. We put

$$m_i = \frac{b_i - a_i^T p_i}{a_i^T a_i},\ 1 \leq i \leq n,$$

in which a_i^T is the $i-$th row of the matrix A, then using relation (2.1), we have

$$p_{i+1,j} = m_i a_{i,j} + p_{i,j},\ 1 \leq j \leq n,$$

or

$$p_{i+1} = m_i a_i + p_i,\ 1 \leq i \leq n,$$

in other words

$$p_{i+1} = A^T \begin{bmatrix} 0 \\ \vdots \\ 0 \\ m_i \\ 0 \\ \vdots \\ 0 \end{bmatrix} + p_i,\ 1 \leq i \leq n. \tag{3.1}$$

After n steps, we put $p_1 := p_{n+1}$ and repeat this process.

If we put $i - 1$ instead of i in (3.1), we have

$$p_i = A^T \begin{bmatrix} 0 \\ \vdots \\ 0 \\ m_{i-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + p_{i-1}. \tag{3.2}$$

By substituting (3.2) into (3.1) we will have

$$p_{i+1} = A^T \begin{bmatrix} 0 \\ \vdots \\ 0 \\ m_{i-1} \\ m_i \\ 0 \\ \vdots \\ 0 \end{bmatrix} + p_{i-1}. \tag{3.3}$$

Continuing this process, we will have the following equality

$$p_{i+1} = A^T \begin{bmatrix} m_1 \\ \vdots \\ m_i \\ 0 \\ \vdots \\ 0 \end{bmatrix} + p_1,\ 1 \leq i. \tag{3.4}$$

Relation (3.4) can be written as follows

$$p_{i+1} = m_1 a_1 + m_2 a_2 + \cdots + m_i a_{\bar{i}} + p_1,\ 1 \leq i,$$

this means that

$$p_{i+1} \in span\{a_1, a_2, a_3, \cdots, a_{\bar{i}}\} + p_1,\ 1 \leq i,$$

where $\bar{i}$ is the remainder of dividing i by n. The following algorithm can be concluded from the above discussion.

Algorithm 3: Algebraic version for NGHK method
Input: $A = (a_1^T, a_2^T, a_3^T, \cdots, a_n^T)_{n\times n}^T,\ b = (b_1, b_2, b_3, \cdots, b_n)^T,\ p_1,\ e,\ n,\ m$
$for\ k = 1 : m$
    $for\ i = 1 : n$
        $m_i = (b_i - (a_i^T p_i))/(a_i^T a_i)$
        $p_{i+1} = \sum_{j=1}^{i} m_j a_j + p_1$
    end
    $e_1 = ||p_{n+1} - p_n||/||p_n||$
    $if\ e_1 < e$
        $stop$
    $else$
        $p_1 = p_{n+1}$
    end
end
Output: p_{n+1}

In fact, the sequence of linear combinations of the rows of the matrix A is convergent to p_{n+1}.

In Algorithm 2, it is easy to see that

$$p_2^T = 1/n \sum_{j=1}^{n} m_j a_j^T + p_1^T,$$

where

$$m_j = \frac{b_j - a_j^T p_1}{a_j^T a_j},$$

and a_j^T is the $j-$th row of the coefficients matrix A. Therefore

$$p_2 = 1/n(m_1 a_1 + m_2 a_2 + \cdots + m_n a_n) + p_1.$$

This results that each p_j^T is a member of the row space of the coefficients matrix A.

Now, the algebraic version of Algorithm 2 can be expressed as follows.

Algorithm 4: Algebraic version for KL method

Input: $A = (a_1^T, a_2^T, a_3^T, \cdots, a_n^T)_{n\times n}^T,\ b = (b_1, b_2, b_3, \cdots, b_n)^T,\ p_1,\ e,\ n,\ m$

for $k = 1 : m$

  for $i = 1 : n$

    $m_i = (b_i - (a_i^T p_i))/(a_i^T a_i)$

  end

  $p_{k+1} = 1/n \sum_{j=1}^{n} m_j a_j + p_1$

  $e_1 = ||p_{k+1} - p_k||/||p_k||$

  if $e_1 < e$

    stop

  else

    $p_1 = p_{k+1}$

  end

end

Output: p_{k+1}

4. **Numerical experiments**

We have already examined the performance of methods NGHK and KL in Section [4], but in this section we want to test the performance of the algebraic versions of these two methods.

We compare three methods algebraic version for NGHK method (algorithm 3), algebraic version for KL method (algorithm 4) and as well as the iterative method for solving system of linear equations Gauss Seidel (GS).

After coding the algorithms with MATLAB software, the calculations were performed on a 2 Duo E630 OEM 1.86 GHz PC with core memory of 1024 Mb. In all examples, k is the number of iterations of the algorithm. When $e_1 < e$, the obtained approximate solution are correct to at least four decimal places.

Now we will write the above algorithm with MATLAB language and provide two examples.

Example 4.1. Assume that

$$A = \begin{bmatrix} 3 & 1 & 9 \\ 1 & 2 & 1 \\ 9 & 1 & 1 \end{bmatrix}, b = \begin{bmatrix} 5 \\ 4 \\ 7 \end{bmatrix}, P_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},\ e = 10^{-7}.$$

Table 1 compares the methods NGHK, KL and GS to solve the linear system $Ax = b$.

method	k	cpu time	e_1
NGHK (algorithm 3)	14	0.0036	9.1231×10^{-7}
KL (algorithm 4)	87	0.0085	9.1488×10^{-7}
GS	10000	0.0491	NaN

TABLE 1. Comparison of methods for Example 4.1.

The graphs of the residual norms (e_1) are shown in the following figure.

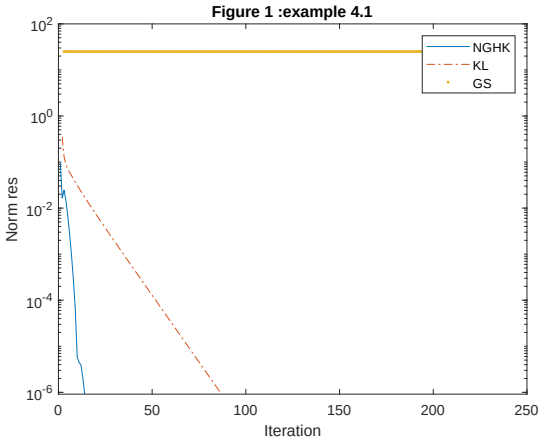

FIGURE 1. Residual norms for Example 4.1.

Example 4.2. Consider $A = (a_{ij})_{10\times10}$, $b = (b_i)_{10\times1}$ and $P = (p_i)_{10\times1}$ so that $100 < a_{ij} < 200$ when $i < j$, $10 < a_{ij} < 20$ when $i \geq j$, also $10 < b_i < 20$ when $1 \leq i \leq 10$, $p_i = 0$ when $1 \leq i \leq 10$, and $e = 10^{-7}$. We solve the system of linear equations $Ax = b$ with the above information by methods NGHK, KL and GS see the results in table 2.

method	k	cpu time	e_1
NGHK (algorithm 3)	249	0.0332	9.6134×10^{-8}
KL (algorithm 4)	4302	0.2260	9.9858×10^{-8}
GS	10000	0.0359	NaN

TABLE 2. Comparison of methods for Example 4.2.

The graphs of the residual norms (e_1) are shown in the following figure.

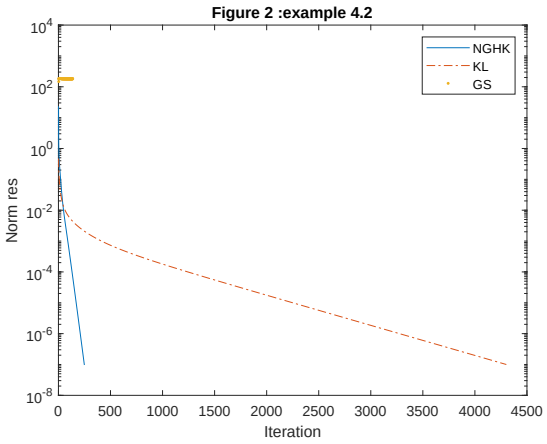

FIGURE 2. Residual norms for Example 4.2.

Conclusion

Although the NGHK and KL algorithms are obtained from a geometric point of view, it was shown in this article as an algebraic interpretation of these algorithms, that the solution of the system of linear equations obtained from a convergent sequence, in the row space of the coefficients matrix of system is obtained.

The result of the performance of the algebraic versions confirms that in the case that the systems of linear equations are such that the angle between the linear equations is appropriate, methods NGHK and KL work better than the famous Gauss Seidel method, of course, the performance of the algorithm NGHK is always is better than KL.

REFERENCES

[1] M. Kryshchuk, J. Lavendels, *Iterative method for solving a system of linear equations.* Procedia Computer Science 104, 133–137, 2017.
[2] B. N. Datta, *Numerical Linear Algebra and Applications, Second Edition.* SIAM, 2010.
[3] J. Demmel, *Applied numerical linear algebra.* SIAM, Philadelphia, PA, 1997.
[4] N. Najafi, M. Ghasemi Kamalvand, *A geometric method for solving systems of linear equations.* J. Korean Soc. Ind. Appl. Math. Vol.29, No.1, 16–25, 2025.
[5] M. Ghasemi Kamalvand, B.Farazmandnia and M.Aliyari, *A method for solving of linear system with normal coefficient matrices.* J. Korean Soc. Ind. Appl. Math. Vol.24, No.3, 305–320, 2020.
[6] M. Ghasemi Kamalvand, Kh. D. Ikramov, *A method of congruent type for linear systems with conjugate-normal coefficient matrices.* Computational mathematic and physics, Vol. 49, No. 2, 203-216, 2009.
[7] M. Ghasemi Kamalvand, Kh. D. Ikramov, *Low-rank perturbations of normal and conjugate-normal matrices and their condensed forms under unitary similarities and congruences.* Computational mathematic and physics, Vol. 33, No. 3, 109-116, 2009.
[8] M. Ghasemi Kamalvand, M. Moradipour, K. Niazi Asil, *A matrix reduction based algorithm to solve k-almost normal systems* Filomat, Volume 38(26), 9143-9150, 2024, https://doi.org/10.2298/FIL2426143G.
[9] M. Ghasemi Kamalvand, K. Niazi Asil, *Indefinite Ruhe's Variant of the Block Lanczos Method for Solving the Systems of Linear Equations* Advances in Mathematical Physics, Volume 2020, Article ID 2439801, 9 pages, https://doi.org/10.1155/2020/2439801.
[10] Y. Saad, *Iterative Methods for Sparse Linear Systems, Second Edition.* SIAM, 2003.

(N. Najafi) Department of Mathematics, Lorestan University, Khorramabad, Iran.

(M. Ghasemi kamalvand) Department of Mathematics, Lorestan University, Khorramabad, Iran.
Email address: ghasemi.m@lu.ac.ir