Filomat 38:26 (2024), 9143–9150
https://doi.org/10.2298/FIL2426143G

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

A matrix reduction based algorithm to solve k-almost normal systems

Mojtaba Ghasemi Kamalvand[a,*], Mojtaba Moradipour[a], Kobra Niaziasil[a]

[a] *Department of Mathematics, Lorestan University, Khorramabad, Iran*

Abstract. In this paper, we develop an efficient algorithm to solve linear system $Ax = b$ where the coefficient matrix A is k-almost normal. We propose an algorithm based on the orthogonalization of the Kyrylov subspace and reduction of the k-almost normal matrix A to a block tridiagonal form. A comparison with the popular GMRES method shows that the proposed algorithm is efficient and in many particular cases generates more accurate results.

1. introduction

Solving linear systems using reduction of the coefficient matrices to block tridiagonal forms, is a popular technique in the literature of numerical linear algebra. Many researchers have studied the application of such techniques for various types of linear algebraic systems. For example, perturbed normal and rank structural matrices [2], the complex symmetric systems [3], the normal systems [4] and conjugate normal systems [6,7] have been investigated using similar techniques. In [8], the authors obtained a dense block tridiagonal form for k-almost normal matrices. In [1], the authors performed an extended study on almost normal matrices and their condensed form. The "Minres-N" method (the minimal residual method for normal systems) proposed by Danna, Zykov and Ikramov [4], has been extended by Ghasemi-Kamalvand and Ikramov [7] for conjugate normal systems and they named it the "Minres C-N" method. The aim of this work is to find a similar technique for k-almost normal systems. For this purpose, we examine the condensed form obtained by [8] with a slight modification and then we optimize it and achieve an efficient algorithm to solve k-almost normal systems. Finally, we present a comparison between the proposed algorithm and GMRES method to demonstrate the efficiency of our proposed algorithm.

2. Reducing the coefficient matrices of linear systems to the block tridiagonal forms

Definition 2.1. *An $n \times n$ complex matrix A is said to be k-almost normal, if there exists a matrix $C \in M_n(\mathbb{C})$ of rank k, such that $A(A^* - C) = (A^* - C)A$.*

2020 *Mathematics Subject Classification*. Primary 65F10; Secondary 65F25.
Keywords. k-almost normal matrix, systems of linear equations, block tridiagonal matrix, generalized Krylov subspace.
Received: 12 March 2023; Accepted: 14 June 2024
Communicated by Yimin Wei
* Corresponding author: Mojtaba Ghasemi Kamalvand
Email addresses: ghasemi.m@lu.ac.ir (Mojtaba Ghasemi Kamalvand), moradipour.mo@lu.ac.ir (Mojtaba Moradipour), Kniaziasil@yahoo.com (Kobra Niaziasil)

The main idea behind the Minres method is to seek the solution of k-almost normal system $Ax = b$ in the subspace spanned by the following vectors

$$x, Ax, (A^* - C)x, A^2x, A(A^* - C)x, (A^* - C)Ax, (A^* - C)^2x, A^3x, \ldots, \tag{1}$$

since $rank(C) = k$ we have,

$$\exists x_i \ , \exists y_i \quad , C = \sum_{i=1}^{k} x_i y_i^*. \tag{2}$$

Substituting (2) in (1) and performing some simplifications, the sequence (1) becomes (see more details in [8])

$$x, x_1, ..., x_k, Ax, Ax_1, ..., Ax_k, A^*x, A^*x_1, ..., A^*x_k, A^2x, A^2x_1, ..., A^2x_k, AA^*x, AA^*x_1, ..., AA^*x_k, \ldots. \tag{3}$$

The above discussion states that both sequences (1) and (3) generate the same subspace. The i-th layer of (3) is defined by the collection of following vectors,

$$A^\alpha A^{*\beta}x, \quad A^\alpha A^{*\beta}x_t, \quad \alpha + \beta = i, \quad 1 \le t \le k.$$

For example the 0-th layer of sequence (3) comprises

$$x, x_1, x_2, \ldots, x_k, \tag{4}$$

and the first layer of sequence (3) contains

$$Ax, Ax_1, \ldots, Ax_k, A^*x, A^*x_1, \ldots, A^*x_k, \tag{5}$$

and so on. The subspace spanned by the first i layers of sequence (3) is called the i-th generalized Krylov subspace denoted by K_i. In other words

$$K_i = span\{v : v = A^\alpha A^{*\beta}x \quad or \quad v = A^\alpha A^{*\beta}x_t, \quad \alpha + \beta \le i, \quad 1 \le t \le k\},$$

Now similar to [8], we orthogonalize the sequence (3) and achieve the following algorithm for reducing the matrix A to the block tridiagonal matrix H.

Algorithm 2.2.

1. *Due to the equation* (2)*, find vectors* $x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k \in \mathbb{C}^n$*, such that* $C = \sum_{i=1}^{k} x_i y_i^*$.
2. *Find an appropriate initial vector x and set*

$$q_1 = \frac{1}{\|x\|_2} x. \tag{6}$$

3. *Apply the Gram-Schmidt orthogonalization process, orthonormalize the* 0*-th layer* (4) *and extract the orthonormal vectors* $\{q_1, q_2, \ldots, q_t\}$ *from it. So we have*

$$K_0 = span\{q_1, q_2, \ldots, q_t\}, \tag{7}$$

   where $t \le k + 1$.
4. *Multiply the basis of previous layer by A and* A^* *and orthonormalize the resulting set by applying the Gram-Schmidt process.*
5. *The next steps are similar to step 4. The scalars generated during these calculations are the same as the entries of the matrix H.*

The procedure described above, lead us to the following theorem.

Theorem 2.3. *Every k-almost normal matrix in $\mathbb{C}^{n\times n}$ is unitarily similar to a block tridiagonal matrix of the form*

$$H = \begin{pmatrix} H_{11} & H_{12} & & \\ H_{21} & H_{22} & H_{23} & \\ & H_{32} & H_{33} & \ddots \\ & & \ddots & \ddots \end{pmatrix}, \tag{8}$$

where the diagonal blocks have the maximum orders $k+1, 2(k+1), 3(k+1), \ldots, i(k+1), \ldots$ respectively. In fact, if by orthonormalization procedure of the sequence (3), the orthonormal basis $q_1, \ldots, q_n$ is constructed for $\mathbb{C}^n$, then we have $Q^*AQ = H$, where $Q = [q_1 \ldots q_n]$.

Proof: See [8, Theorem 2.1].

It is noteworthy that similar theorems have been proved for normal and conjugate normal matrices (see [5] and [6]).

3. solving linear systems using matrix reduction

In this section, we propose an algorithm to solve the k-almost normal system $Ax = b$ by reducing the coefficient matrix A to its block tridiagonal form.

Let Q be the unitary $n \times n$ matrix formed column-wise by vectors $q_1, ..., q_n$, then theorem 2.3 states that $Q^*AQ = H$ or equivalently $(AQ = QH)$.

Assume that the $s_t = (t+1)k + (t+1)$ steps of the procedure described in the previous section have been done, so the first s_t columns of the block tridiagonal matrix H and the orthonormal vectors $q_1, ..., q_{s_t}$ are known. The vectors $q_1, ..., q_{s_t}$ construct an orthonormal basis for generalized Krylov subspace

$$K_t = K_t(A, x, x_1, ..., x_k).$$

We choose the vector $x_{s_t} \in K_t$ as the approximate solution of the system $Ax = b$ provided that $r = b - Ax_{s_t} \perp K_t$, i.e., $Q_{s_t}^* r = Q_{s_t}^*(b - Ax_{s_t}) = 0$. We seek x_{s_t} in the form $x_{s_t} = y_1 q_1 + ... + y_{s_t} q_{s_t}$, thus

$$Q_{s_t} Y_{s_t} = x_{s_t}, \quad for \;\; Y_{s_t} = [y_1, \;\; \cdots, \;\; y_{s_t}]^T, \tag{9}$$

so we have $Q_{s_t}^*(b - AQ_{s_t}Y_{s_y}) = 0$. Note that after performing t steps of the algorithm, we achieve the following relation

$$A[q_1, ..., q_{s_t}] = [q_1, ..., q_{s_t}]H_{s_t} + R_{s_t},$$

in other words

$$AQ_{s_t} = Q_{s_t} H_{s_t} + R_{s_t}, \tag{10}$$

or

$$AQ_{s_t} = Q_{s_{t+1}} \hat{H}_{s_t}, \quad \hat{H}_{s_t} = H'_{s_t} + R'_{s_t},$$

$$H'_{s_t} = \begin{bmatrix} H_{s_t} \\ 0 \end{bmatrix}, \quad 0 = 0[s_t + 1 : s_{t+1}, l : s_t], \quad Q_{s_{t+1}} R'_{s_t} = R_{s_t}$$

in which,

$$R'_{s_t} = \begin{bmatrix} 0 & 0 \\ 0 & R''_{s_t} \end{bmatrix}$$

where $R''_{s_t} = \hat{H}_{s_t}[s_t + 1 : s_{t+1}, l : s_t]$, $l = tk + t + 2$. Now, from (10) we deduce:

$$\begin{aligned} 0 &= Q_{s_t}^*(b - AQ_{s_t}Y_{s_t}) \\ &= Q_{s_t}^*(b - (Q_{s_t}H_{s_t} + R_{s_t})Y_{s_t}) \\ &= Q_{s_t}^* b - (Q_{s_t}^* Q_{s_t} H_{s_t} + Q_{s_t}^* R_{s_t})Y_{s_t} \\ &= \|b\|_2 e_1^{(s_t)} - H_{s_t}Y_{s_t} - Q_{s_t}^* R_{s_t} Y_{s_t}. \end{aligned}$$

Each column of the matrix R_{s_t} is a linear combination of several q_i vectors with indices greater than s_t, thus $Q^*_{s_t} R_{s_t} = 0$ and therefore

$$H_{s_t} Y_{s_t} = \|b\|_2 e_1^{(s_t)}. \tag{11}$$

After computing Y_{s_t} form system (11), x_{s_t} will be found by substituting Y_{s_t} in (9). Now we extend the algorithm 2.2 for k-almost normal systems.

Algorithm 3.1. *(MkAN: Matrix reduction method for k-almost normal systems)*

Inputs: Matrix A, which is k-almost normal, the positive real number ϵ and the vectors $b, x_1, x_2, x_3, \ldots, x_k$.

$q_1 = \frac{b}{\|b\|_2}$
$s = 1$
for $i = 1$ *to* k
  $z = x_i - \sum_{j=1}^{s} q_j^* x_i q_j$
  if $\|z\|_2 > 0$
    $s = s + 1$
    $q_s = z/\|z\|_2$
  end
end for
$c_1 = 0$
$c_2 = s$
$s = s + 1$
$w = 3$
while $w \leq n$
  $j = c_{w-2} + 1$
  $p = j$
  if $w < 4$
    $d = 1$
  else
    $d = c_{w-3} + 1$
  end
  for $j = c_{w-2} + 1 : c_{w-1}$
    $z = Aq_j$
    $z = z - \sum_{l=d}^{s-1} q_l^* z q_l$
    if $\|z\|_2 > 0$
      for $l = d$ *to* $s - 1$
        $h_{lp} = q_l^* A q_j$
      end for
      $q_s = z/\|z\|_2$
      $h_{sj} = \|z\|_2$
      $p = p + 1$
    end if
    if $s < n$
      $s = s + 1$
    else
      break
    end
  end for
  $a = s$
  if $w < 4$
    $f = 1$
  else

$f = b$
end if
for $j = f$ *to* c_{w-1}
$z = A^* q_j$
$z = z - \sum_{l=d}^{s-1} q_l^* z q_l$
if $\|z\|_2 > 0$
$q_s = z/\|z\|_2$
for $l = d$ *to* $s-1$
$h_{jl} = \overline{q_l^* A^* q_j}$
end for
$h_{js} = \|z\|_2$
if $s < n$
$s = s + 1$
else
break
end
end for
$Q_w = (q_1, ..., q_w)$
$y_w = H_w^{-1} \|b\|_2 e_1^{(w)}$
$x_w = Q_w y_w$
$r = b - Ax_w$
if $\|r\| < \epsilon$
breek
end if
$c_w = s - 1$
$b = a$
if $s < n + 1$
$w = w + 1$
else
break
end
end while

Outputs: Vector x_w *(approximate solution of system* $Ax = b$*) and the condensed matrix* H*.*

4. Numerical experiments

In this section, we provide some numerical examples to demonstrate the efficiency and applicability of the proposed algorithm. In all examples we implement the algorithms using MATLAB with $\epsilon = 10^{-7}$. Finally we will compare the performance of our algorithm with the famous GMRES method.

Example 4.1. *Suppose that*

$$A = \left[\begin{array}{ccc|cc} * & & & & \\ & \ddots & & 0 & \\ & & * & & \\ \hline & 0 & & 5 & 0 \\ & & & 3 & 5 \end{array}\right]_{100\times 100}$$

The entries of the main diagonal in the upper left block of matrix A are chosen uniformly random in $(0, 1)$*. By defining*

$$C = \left[\begin{array}{c|cc} 0 & 0 & \\ \hline 0 & -\frac{3}{2} & 3 \\ & \frac{3}{4} & -\frac{3}{2} \end{array}\right]_{100\times 100},$$

duo to definition 2.1 one can verify that A is 1*-almost normal (see [1]). Now, we choose the vector* $b \in \mathbf{R}^{100}$ *with uniformly random entries in* $(0, 1)$ *and focus on the linear system* $Ax = b$.

After running the algorithm MkAN (Algorithm 3.1), the matrix H is constructed with diagonal blocks having sizes of

$$2 \times 2, \;\; 4 \times 4, \;\; 6 \times 6, \;\; \cdots,$$

respectively. The matrix H has about 3,500 non-zero entries. The dimensions of the diagonal blocks as shown in Figure 1, grow increasingly, hence the matrix H has a huge amount of non-zero entries and these cause to increase the cost of computations.

$H \;\; =$

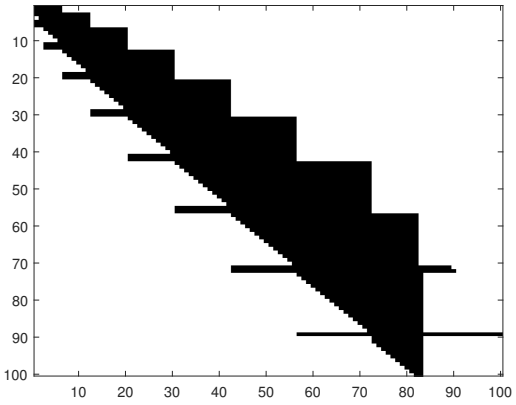


Figure 1 : The shape of the reduced matrix H in example 4.1

Let the k-almost normal matrix A be chosen such that the length of the layers in sequence (3) does not grows, then the reduced matrix H resulting from the algorithm MkAN is a block tridiagonal matrix having constant size diagonal blocks. In this situation, the performance of the algorithm improves. The following examples provides some instants of this situation.

Example 4.2. *In this example, we select the entries of the 1-almost normal matrix A in example 4.1 such that its eigenvalues are on an ellipse, in this case* $(A^*)^2x$ *and* $(A^*)^2x_1$ *are linear combinations of* A^2x *and* A^2x_1*, this causes that the length of the layers in sequence (3) does not exceed* 4*, so the size of the diagonal blocks in the matrix H is* 4×4 *and it has about 900 non-zero entries (see Figure 2). The implementation results of two algorithms MkAN and GMRES are shown in the table 1.*

method	w	*cpu time*	$r(x_s) = b - Ax_s$
MkAN	*100*	*0.023*	9.06×10^{-7}
GMRES	*64*	*0.026*	5.76×10^{-7}

table 1 : Comparison of the two methods MkAN and GMRES in Example 4.2

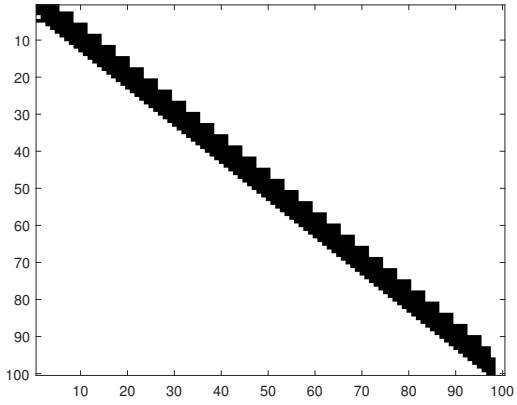
$H=$ 

Figure 2 : The shape of the reduced matrix H in example 4.2

Example 4.3. *Assume that* $x_1, x_2 \in \mathbf{R}^{100}$ *with uniformly random entries in* $(0, 1)$. *Now, consider the following relations*

$$A = B + x_2 x_1^T, \;\; C = x_1 x_2^T - x_2 x_1^T,$$

where B is a symmetric matrix. It can be easily seen that,

$$A^T A - A A^T = CA - AC.$$

Since rank$(C) = 2$, *the matrix A is* 2*-almost normal. Now consider the vector b as in previous example, table 2 compares the methods MkAN and GMRES to solve the linear system* $Ax = b$.

method	*w*	*cpu time*	$r(x_s) = b - Ax_s$
MkAN	*54*	*0.021*	5.96×10^{-7}
GMRES	*24*	*0.094*	4.74×10^{-7}

table 2 : Comparison of the two methods MkAN and GMRES in Example 4.3

In example (4.3), matrix A is 2-almost normal, and after running the algorithm MkAN, matrix H has diagonal blocks with sizes of 3×3, and has about 600 non-zero entries (See figure 3). The reason for the above is that, there are only three linear independent vectors in each layer of sequence (3). For example, in the first layer, the vectors A^*x, A^*x_1, and A^*x_2 are linear combinations of vectors Ax, Ax_1 and Ax_2.

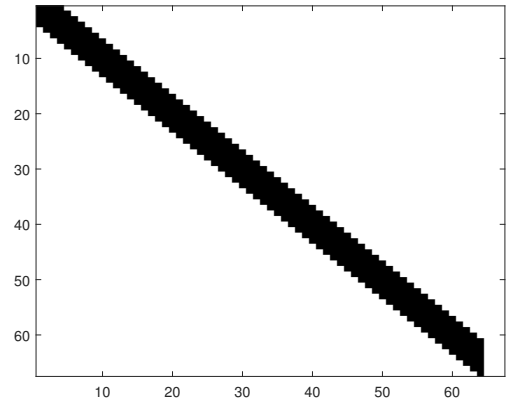
$H=$ 

Figure 3 : The shape of the reduced matrix H in example 4.3

In examples (4.2) and (4.3), although the number of iterations of the MkAN algorithm is greater than GMRES, its execution time is lesser than GMRES (Compare tables 1 and 2). The reason of this situation is that the number of arithmetic operations in each step of MkAN algorithm is lesser than GMRES algorithm.

In figures 1, 2 and 3, the black parts are non-zero and the white parts are zero entries.

Acknowledgements
The authors thanks Kh.D. Ikramov for valuable comments which greatly improved the presentation.

Conflicts of interest
The author declares that there are no conflicts of interest.

References

[1] R. Bevilacqua, G. M. Del Corso, *A condensed representation of almost normal matrices*, Linear algebra and its applications, **438** (2013), 4408–4425.
[2] R. Bevilacqua, G. M. Del Corso, L. Gemignani, *Block tridiagonal reduction of perturbed normal and rank structured matrices*, Linear algebra and its applications, **439** (2013), 3505–3517.
[3] A. Bunse-Gerstner, R. Stover, *On a conjugate-type mathod for solving complex symmetric linear systems*, Linear algebra and its applications, **287** (1999), 105–123.
[4] M. Dana, A. G. Zykov, Kh.D. Ikramov, *A minimal residual method for a special class of the linear systems with normal coefficient matrices*, Computational mathematic and physics, **25** (2005), 1854–1863.
[5] L. Elsner, Kh. D. Ikramov, *On a condensed form for normal matrices under finite sequences of elementary unitary similarities*, Linear algebra and its applications, **254** (1997), 79–98.
[6] M.Ghasemi Kamalvand, Kh. D. Ikramov, *A method of congruent type for linear systems with conjugate-normal coefficient matrices*, Computational mathematic and physics, **49**(2), (2009), 203–216.
[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, **33**(3) (2009), 109-116.
[8] K. Niazi Asil, M. Ghasemi Kamalvand, *On reduction of K-almost and K-almost conjugate normal matrices to a block tridiagonal form*, J.Korean Soc. Ind. Appl. Math., **23**(3) (2019), 267-282.