BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 68, No. 2, 2020
DOI: 10.24425/bpasts.2020.133113

SPECIAL SECTION

Computer methods for calculating tuple solutions of polynomial matrix equations

J. DOROŻYŃSKI* and M. NEDASHKOVSKYY

Kazimierz Wielki University, Institute of Computer Science, M. Kopernika 1, 85-074 Bydgoszcz

Abstract. Schemes are presented for calculating tuples of solutions of matrix polynomial equations using continued fractions. Despite the fact that the simplest matrix equations were solved in the second half of the 19th century, and the problem of multiplier decomposition was then deeply analysed, many tasks in this area have not yet been solved. Therefore, the construction of computer schemes for calculating the sequences of solutions is proposed in this work. The second-order matrix equations can be solved by a matrix chain function or iterative method. The results of the numerical experiment using the MatLab package for a given number of iterations are presented. A similar calculation is done for a symmetric square matrix equation of the 2nd order. Also, for the discrete (time) Riccati equation, as its analytical solution cannot be performed yet, we propose constructing its own special scheme of development of the solution in the matrix continued fraction. Next, matrix equations of the n-th order, matrix polynomial equations of the order of non-canonical form, and finally, the conditions for the termination of the iterative process in solving matrix equations by branched continued fractions and the criteria of convergence of matrix branching chain fractions to solutions are discussed.

Key words: matrix polynomial equations, discrete (time) Riccati equation, tuples of solutions, MatLab.

1. Introduction

The issues raised in the article have appeared in generalised Leontiev-Ford simulations for the macroeconomic balance [1, 2] in solving Fredholm integral equations with non-linear nuclei. Contemporary scientific literature lacks publications related to the problem. In the case of determining the interval for the elements of algebraic equations of real numbers, this can be done with the Sturm sequence, but this method cannot be generalised for the case of matrix equations. This leads to the conclusion that this problem has not been solved yet. A similar case of simple 1×1 matrices for which there is a solution does not apply to Leontiev-Ford simulations, i.e. solving integral equations. The aim of the publication is to propose a scheme of approach to receiving sequences of solutions.

So the matrix polynomial equations of general form are presented as

$$A_n X^n + A_{n-1} X^{n-1} + \ldots + A_1 X + A_0 = 0, \qquad (1)$$

where $A_i \in R^{m\times m}\left(\overline{0, n}\right)$ – square nonzero matrices of order m with real elements, and X – an unknown square matrix of order m.

The simplest matrix equations were solved in the second half of the 19th century [3]. The problem of decomposition on a multiplier is deeply analysed in [4]. Most known computational schemes are described in [3] and [5], however, many tasks in this area have not yet been solved [6, 7]. Attention will be focused on one of them in this work – the construction of computer schemes for calculating the tuples of solutions.

2. Second-order polynomial matrix equations

Consider the equation

$$AX^2 + BX + C = 0 \qquad (2)$$

where A, B, C and $X \in R^{m\times m}$.

After regrouping the members, we obtain

$$X = -(B + AX)^{-1} C. \qquad (3)$$

On the basis of (3), it is possible to carry out the following iterative procedure

$$X_i = -(B + AX_{i-1})^{-1} C. \qquad (4)$$

On the other hand, composition (3) gives the following development of solution (2) in the matrix chain fraction

$$X = -\Big(B - A\big(B - A(B - A(B - \ldots)^{-1}C)^{-1}C\big)^{-1}C\Big)^{-1}C. \qquad (5)$$

There is another scheme for constructing an iterative method of the computation X-1. Assuming that there are invertible matrices A-1, X-1 for (2), we can write a recursive formula

$$X = -A^{-1}\cdot B - A^{-1}\cdot C\cdot X^{-1}. \qquad (6)$$

*e-mail: janusz.dorozynski@ukw.edu.pl

Manuscript submitted 2019-07-05, revised 2019-12-06, initially accepted for publication 2020-01-07, published in April 2020

Equality (6) can be used to construct an iterative process

$$X_i = -A^{-1} \cdot B - A^{-1} \cdot C \cdot X_{i-1}^{-1}.$$

And composition (6) gives another development of X in a continued matrix fraction

$$X = -\left(A^{-1}B - A^{-1}C\left(-A^{-1}B - A^{-1}C\left(-A^{-1}B - \ldots\right)^{-1}\right)^{-1}\right)^{-1}, \quad (7)$$

which may converge to another solution; it follows that matrix multiplication is not, in general, commutative.

Formulas of development (5) and (7) for X in equation (2) are separated and coincide with different solutions.

Consider as an example, the matrix equation

$$AX^2 + BX + C = 0$$

where

$$A = \begin{pmatrix} 7 & -3 & -5 \\ 0.22 & 5.1 & 2.5 \\ 0.22 & -0.234 & -3.2 \end{pmatrix};$$

$$B = \begin{pmatrix} 1 & 6 & -5 \\ 0.5 & 1.22 & -2.51 \\ 0.234 & -0.13 & 2.2 \end{pmatrix};$$

$$C = \begin{pmatrix} -49.0707 & 56.0938 & 88.7682 \\ 7.6545 & -57.6309 & -115.6766 \\ -1.2741 & 13.4398 & 35.2964 \end{pmatrix};$$

For matrix A, B, C one of the elements is known, but the given formula allows to determine two others.

By using the MatLab package, this equation was validated using the "left" recurrence formula $X_i = -(B + AX_{i-1})^{-1}C$ and "right" — $X_i = -A^{-1} \cdot B - A^{-1} \cdot C \cdot X_{i-1}^{-1}$, respectively. The text of the M-file for the MatLab package is shown below:

```
function [] =AXsquaredPLusBXplusC_PlnmEquation
( Iter_Count )

   % Calculations solution sequences of the
   % matrix polynomial equations
   %       A*X^2+B*X+C=0
   % by matrix continued fractions
   clear all;
disp(' ');
disp('---------------------------------------------------------') ;
   A=[7 –3 –5;0.22 5.1 2.5;0.22 –0.234 –3.2 ];
   B=[1 6 –5; 0.5 1.22 –2.51; 0.234 –0.13 2.2];
   C=[–49.0707 56.0938 88.7682;7.6545 –57.6309
      –115.6766;–1.2741 13.4398 35.2964];
   k=1;
   while(k<9)
   % First approximations of solutions
   X=[1 0 0;0 1 0; 0 0 1] ;
   Y=[ 2.5 –1 0.2; –0.236 –1.33 –7.232; 0.256 –1.1 –1.267];
   Z=Y;
   i = 1;
     while (norm(X–Z)/norm(X)>10^(–k) );
       i = i+1;
       Z=X;
       X = -A^(–1)*B-A^(–1)*C*X^(–1);
     end;
disp(' ') ;
fprintf('%s %e', '       epsilon= ' ,10^(–k));
disp(' ');
fprintf('%s %i' ,       'Count iteration of 1st scheme', i);
fprintf(' %s %e ','Error for X1 ', norm(A*X^2+B*X+C));
  Z=Y;
  % Second approximations of solutions
   Y=[1 0 0;0 1 0; 0 0 1];
   i=1;
   while (norm(Y–Z)/norm(Y)>10^(–k) );
       i = i+1;
       Z=Y;
       Y= –(A*Y+B)^(–1)*C;
     end;
      X1=X;
      X2=Y;
disp(' ');
fprintf('%s %i', 'Count iteration of 2nd scheme', i)
fprintf(' %s %e ', 'Error for X2 ', norm(A*Y^2+B*Y+C));
   k=k+1;
    end
    X1=X
    X2=Y
   end
```

where X and Y are arbitrary, which does not affect the result.

As a result of calculations at the initial approximation

$$X_0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

the solution of the "left" formula is calculated

$$X_l = \begin{pmatrix} 1.7429 & -2.0338 & 3.2240 \\ -11.2515 & -16.3708 & 36.7645 \\ -4.4950 & -7.5871 & 17.7088 \end{pmatrix}.$$

At the same time, another solution is calculated for the "right" scheme

$$X_r = \begin{pmatrix} -7.7367 & 31.4896 & 65.9651 \\ -4.1912 & 24.4310 & 53.9026 \\ 1.2948 & -8.5957 & -19.2974 \end{pmatrix}.$$

The results of the numerical experiment are given in the following Table 1

Table 1

$\|X_k - X_{k-1}\|/\|X_k\|$	Left scheme	Right scheme
.000000e-001	37 iterations	42 iterations
.000000e-002	59 iterations	65 iterations
.000000e-003	81 iterations	87 iterations
1.000000e-004	103 iterations	110 iterations
1.000000e-005	126 iterations	132 iterations
1.000000e-006	148 iterations	154 iterations
1.000000e-007	170 iterations	177 iterations
1.000000e-008	193 iterations	199 iterations

3. Symmetric polynomial square matrix equation of the 2nd order

Let it be:

$$AX + BX + XFX + C = 0 \tag{8}$$

where A, B, C, D and X are matrices of size y $m \times m$.

After regrouping the members, we obtain

$$X = -F^{-1}B + (A + XF)^{-1} \cdot (AF^{-1}B - C). \tag{9}$$

From (9), we obtain an iterative formula of calculation X

$$X_i = -F^{-1}B + (A + X_{i-1}F)^{-1} \cdot (AF^{-1}B - C).$$

On the basis of (9), it is also possible to give a solution in the form of a matrix continued fraction

$$\begin{aligned} X = -F^{-1}B + \left(A + \left(-F^{-1}B + \ldots\right) \cdot F\right)^{-1} \times \\ \times \left(AF^{-1}B - C\right)\right) \cdot F\right)^{-1} \cdot \left(AF^{-1}B - C\right). \end{aligned} \tag{10}$$

On the other hand, we have for equation (8)

$$X = -AF^{-1} + (AF^{-1}B - C) \cdot (FX + B)^{-1}. \tag{11}$$

From (11), we can also write a recursive formula for the computation of X

$$X_i = -AF^{-1} + (AF^{-1}B - C) \cdot (FX_{i-1} + B)^{-1}.$$

On the basis of (11), another solution of the equation in the form of a matrix continued fraction can also be applied

$$\begin{aligned} X = -AF^{-1} + (AF^{-1}B - C) \cdot \big(F \cdot \big(-AF^{-1} + \\ + (AF^{-1}B - C) \cdot \big(F \cdot (-AF^{-1} + \ldots) + B\big)^{-1}\big) + B\big)^{-1}. \end{aligned}$$

Let us consider as an illustration, a concrete matrix

$$XFX + AX + XB + C = 0,$$

where

$$A = \begin{pmatrix} 12 & -3 & -5 \\ 0.22 & 0.251 & 0.25 \\ 0.22 & -0.234 & -0.13 \end{pmatrix};$$

$$B = \begin{pmatrix} 1 & 6 & -5 \\ 0.25 & 0.22 & 0.251 \\ 0.234 & -0.13 & 0.22 \end{pmatrix};$$

$$C = \begin{pmatrix} -17.8735 & 4.3189 & 7.3513 \\ 10.1108 & -11.6472 & -2.6332 \\ 3.4000 & -3.5873 & 2.1216 \end{pmatrix}.$$

Also, for matrix A, B, C one of the elements is known, but the given formula allows to determine two others.

Using the MatLab package, this matrix equation was validated using the "left" recursive formula

$$X_i = -F^{-1}B + (A + X_{i-1}F)^{-1}(AF^{-1} - B - C)$$

and the "right" formula

$$X_i = -AF^{-1} + (AF^{-1}B - C)(FX_{i-1} + B)^{-1},$$

respectively.

As a result of calculations at the initial approximation

$$X_0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

the solution is calculated of the left recursive formula

$$X_l = \begin{pmatrix} -2.9984 & 6.5088 & -0.0999 \\ -4.7394 & 7.0966 & 1.5527 \\ -11.1806 & 18.7830 & 0.2514 \end{pmatrix}.$$

At the same time, another solution is calculated for the right

$$X_r = \begin{pmatrix} -7.3635 & -3.7180 & 16.8727 \\ 3.4571 & -1.1975 & -8.1790 \\ 1.7465 & -0.6429 & -3.0706 \end{pmatrix}.$$

The results of the experiments are given in Table 2

Table 2

$\|X_k - X_{k-1}\|/\|X_k\|$	Left scheme	Right scheme
1.000000e-001	17 iterations	16 iterations
1.000000e-002	22 iterations	22 iterations
1.000000e-003	25 iterations	27 iterations
1.000000e-004	31 iterations	30 iterations
1.000000e-005	35 iterations	iterations 35
1.000000e-006	42 iterations	iterations 41
1.000000e-007	45 iterations	44 iterations
1.000000e-008	51 iterations	50 iterations

4. Discrete Riccati equation

Consider the case of the Riccati equation, a type of nonlinear equation that arises in the context of problems in either continuous or discrete time. For continuous time, we have a state-dependent Riccati equation (SDRE) [8]. In general, the analytical solution of the SDRE cannot be performed yet. One of the ways of solving the SDRE are software packages, such as Matlab [9]. The text of proper M-file for the MatLab package is shown below:

```
function [] = RiccatiEquation
   %     A'*Y*A-Y-A'*Y*B*(R-B'*Y-B)^(-1)*B'*Y*A -Q=0
   % by matrix continued fractions
   clear all;
   disp(' ');
   disp('_________________________________________');
   A=[7 -3 -5;0.22 5.1 2.5;0.22 -0.234 -3.2 ];
   B=[1 6 -5; 0.5 1.22 -2.51; 0.234 -0.13 2.2];
   R=[3 -6 5; 5 22 -2.51; 2.34 -13 1.22];
   Y=[ 15 -1 0.2; -0.236 -1.33 -1.232; 0.256 -1.1 -0.267];
   Q=-A'*Y*A+Y+A'*Y*B*(R+B'*Y+B)^(-1)*B'*Y*A k=1;
while(k<9);
   % First apromations of solutions
   X=[1 0 0;0 1 0; 0 0 1] ;
  Z=Y;
   i = 1;
   while (norm(X-Z)/norm(X)>10^(-k) );
     i = i+1;
     Z=X;
     X =Q+A'*(A^(-1)*B*R^(-1)*B'+A^(-1)*X^(-1))^(-1);
  end;
  X1=X;
  disp(' ');
 fprintf('%s %i' , 'Count itaration of 1th sheme' ,i )
 fprintf(' %s %e ','Error for X1 ', norm( A'*X*A-X-A'
 *X*B*(R+B'*X*B)^(-1)*B'*X*A+Q));
  X=[1 0 0;0 1 0; 0 0 1] ;
  Z=Y;
  i=1;
  while (norm(X-Z)/norm(X)>10^(-k) );
     i = i+1;
     Z=X;
     X=Q+(B'*(A')^(-1)+R*B^(-1)*X^(-1)*(A')^(-1))^(-1)
     *R*B^(-1)*A ;
  end;
  disp(' ');
 fprintf('%s %i' , 'Count itaration of 2th sheme' ,i )
 fprintf(' %s %e ','Error for X2 ', norm( A'*X*A-X-A'
 *X*B*(R+B'*X*B)^(-1)*B'*X*A+Q) );
 k=k+1;
  end
X1=X
X2=Z
 end
```

As a result of calculations at the initial approximation

$$X_0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

the solution of the “left” formula is calculated

$$X_l = \begin{pmatrix} 2.1313 & -0.8828 & -1.4406 \\ -0.9733 & 0.4443 & 0.6673 \\ -1.5319 & 0.6337 & 1.0369 \end{pmatrix}.$$

At the same time, another solution is calculated for the “right” scheme

$$X_r = \begin{pmatrix} 2.1313 & -0.8828 & -1.4406 \\ -0.9733 & 0.4443 & 0.6673 \\ -1.5319 & 0.6337 & 1.0369 \end{pmatrix}.$$

For the second case, i.e. a discrete (time) Riccati equation, let us now consider what is known in the applications of the equation:

$$\begin{aligned} &A^TXA - X - A^TXB\big(R + \\ &+ B^TXB\big)^{-1}B^TXA + Q = 0, \end{aligned} \tag{12}$$

where A, B, C, R, Q and X are matrices of size $m \times m$.

As a result of transformations, equation (12) can be written as

$$X = Q + A^T\big(A^{-1}BR^{-1}B^T + A^{-1}X^{-1}\big)^{-1}. \tag{13}$$

This expression can also be carried out for iterations to calculate X:

$$X_i = Q + A^T\big(A^{-1}BR^{-1}B^T + A^{-1}X_{i-1}^{-1}\big)^{-1}. \tag{14}$$

On the basis of (14), we obtain a solution in the form of a two-period matrix continued fraction

$$\begin{aligned} X = {} & Q + A^T\big(A^{-1}BR^{-1}B^T + A^{-1}\big(Q + \\ & + A^T\big(A^{-1}BR^{-1}B^T + A^{-1}\big(Q + \\ & + A^T\big(A^{-1}BR^{-1}B^T + \ldots\big)^{-1}\big)^{-1}\big)^{-1}\big)^{-1}\big)^{-1}. \end{aligned} \tag{15}$$

On the other hand, equation (12) can be obtained from:

$$X = Q + \left[B^T\big(A^T\big)^{-1} + B^{-1}X^{-1}\big(A^T\big)^{-1}\right]^{-1}RB^{-1}A. \tag{16}$$

On the basis of (16), we have an iterative formula for calculating the solution

$$X_i = Q + \left[B^T\big(A^T\big)^{-1} + RB^{-1}X_{i-1}^{-1}\big(A^T\big)^{-1}\right]^{-1}RB^{-1}A.$$

From (16), the development of a solution in a two-period matrix continued fraction is obtained

$$X = Q + \big(B^T(A^T)^{-1} + RB^{-1}(Q + \\ + \big(B^T(A^T)^{-1} + RB^{-1}(Q + \ldots)^{-1} \times \\ \times \big(A^T\big)^{-1}\big)^{-1}RB^{-1}A\big)^{-1}\big(A^T\big)^{-1}\big)^{-1}RB^{-1}A. \tag{17}$$

Consequently, for each of the considered equations, (2), (8), and (12), its own special scheme of development of the solution in the matrix continued fraction can be constructed, converging to another solution.

5. Polynomial matrix equations of the n-th order

It turns out that, in this case, you can build a general scheme. Let us have

$$X^n + A_{n-1}X^{n-1} + A_{n-2}X^{n-2} + \\ + \ldots + A_1X + A_0 = 0, \tag{18}$$

where $A_i \in R^{m\times m}\,(i = \overline{0, n-1})$, $X \in R^{m\times m}$ is the matrix, and $n \geq 2$ is an integer number.

Then the solution of equation (18) can be in the form of a one-branched branched matrix continued fraction with branches of $n-1$ branching.

$$X = P_0 + \sum_{k=1}^{n-1} P_k\left(P_0 + \sum_{k=1}^{n-1} P_k\left(P_0 + \right.\right. \\ \left.\left. + \sum_{k=1}^{n-1} P_k\big(\ldots - Q_k\big)^{-1}X - Q_k\right)^{-1} - Q_k\right)^{-1} \tag{19}$$

which is obtained by the composition of fractional linear expressions of the species

$$X = P_0 + \sum_{k=1}^{n-1} P_k(X - Q_k)^{-1} \tag{20}$$

where $P_k \in R^{m\times m}$ and $Q_k \in R^{m\times m}(k = \overline{0, n-1})$ are square matrices whose elements $p_{i,j,k}(i, j = \overline{1, m})$ and $q_{i,j,k}(i, j = \overline{1, m};$ $k = \overline{1, n-1})$ are determined from the system of equations obtained by the method of indefinite coefficients

$$(-1)^{n-1}Q_1Q_2\ldots Q_{n-1} + P_0 = A_1;$$

$$(-1)^{n-2}\sum_{k=1}^{n-1}\prod_{l=1}^{k-1}Q_l\prod_{l=k+1}^{n-1}Q_l - \sum_{k=1}^{n-1}P_k + \\ + (-1)^{n-1}P_0Q_1Q_2\ldots Q_{n-1} = A_2;$$

$$(-1)^{n-2}\sum_{k=2}^{n-1}\sum_{l=k+1}^{n-2}(1-\delta_{kl})\prod_{r=1}^{k-1}Q_k\prod_{r=k+1}^{l-1}Q_r\prod_{r=l+1}^{n-2}Q_r + \\ + \sum_{k=1}^{n-1}P_k\prod_{r=1}^{k-1}Q_r\prod_{r=k+1}^{n-1}Q_r + (-1)^{n-1}\sum_{k=1}^{n-1}\prod_{r=1}^{k-1}P_0Q_r = A_3;$$

$$\sum_{k=1}^{n-1}Q_k + \sum_{k=2}^{n-1}\sum_{l=k+1}^{n-1}P_1Q_kQ_l + \ldots + \\ + \sum_{k=1}^{n-1}\sum_{l=k+1}^{n-1}(1-\delta_{kr})P_rQ_kQ_l + \ldots + \\ + (-1)^{n-1}\sum_{k=1}^{n-1}\prod_{r=1}^{k-1}P_{n-1}Q_kQ_l = A_{n-1};$$

$$\sum_{k=1}^{n-1}P_1Q_k + \ldots + \sum_{k=1}^{n-1}(1-\delta_{kr})P_rQ_r + \ldots + \\ + \sum_{k=1}^{n-2}P_{n-1}Q_k + \sum_{k=1}^{n-1}P_0Q_k = A_0.$$

If you put $Q_k = q_k \cdot E\;(k = \overline{1, n-1})$ and provide for all scalar q_k the pairs of different, distinct numerical values, then the last system of matrix equations will become linear, relatively unknown P_i for all $(i = \overline{0, n-1})$ and will have a single solution. To calculate the numerical value of a solution for X using a computer, the recursive formula (20) is rewritten in the form

$$X_i = P_0 + \sum_{k=1}^{n-1} P_k\big(X_{i-1} - Q_k\big)^{-1}. \tag{21}$$

6. Matrix polynomial equation of the n-th order of non-canonical form

Let us now introduce into consideration

$$X^n + X^{n-1}A_{n-1} + X^{n-2}A_{n-2} + \\ + \ldots + XA_1 + A_0 = 0. \tag{22}$$

where $A_i \in R^{m\times m}\,(i = \overline{0, n-1})$ $X \in R^{m\times m}$ is the matrix, and $n \geq 2$ is an integer number.

For equation (22), the solution can also be written in the form of a one-branched matrix continued fraction with $n-1$ branches:

$$X = P_0 + \sum_{k=1}^{n-1}\big(P_0 - Q_k + \ldots + \\ + \sum_{k_s=1}^{n-1}\big(P_0 - Q_{k_s} + \ldots P_{k_s}\big)^{-1}P_{k_1}\big)^{-1}P_k,$$

which is obtained by the composition of fractional linear expressions of the species

$$X = P_0 + \sum_{k=1}^{n-1}(X - Q_k)^{-1}P_k, \tag{23}$$

where $P_k \in R^{m\times m}$ and $Q_k \in R^{m\times m}(k = \overline{0, n-1})$ are square matrices whose elements $p_{i,j,k}(i, j = \overline{1, m})$ and $q_{i,j,k}(i, j = \overline{1, m};$ $k = \overline{1, n-1})$ are determined from the system of equations obtained by the method of indefinite coefficients

$$(-1)^{n-1}Q_1Q_2\ldots Q_{n-1}+P_0=A_1;$$

$$(-1)^{n-2}\sum_{k=1}^{n-1}\prod_{l=1}^{k-1}Q_l\prod_{l=k+1}^{n-1}Q_l-\sum_{k=1}^{n-1}P_k+$$
$$+(-1)^{n-1}Q_1Q_2\ldots Q_{n-1}P_0=A_2;$$

$$(-1)^{n-2}\sum_{k=2}^{n-1}\sum_{l=k+1}^{n-2}(1-\delta_{kl})\prod_{r=1}^{k-1}Q_k\prod_{r=k+1}^{l-1}Q_r\prod_{r=l+1}^{n-2}Q_r+$$
$$+\sum_{k=1}^{n-1}\prod_{r=1}^{k-1}Q_r\prod_{r=k+1}^{n-1}Q_rP_k+(-1)^{n-1}\sum_{k=1}^{n-1}\prod_{r=1}^{k-1}Q_rP_0=A_3;$$

$$\sum_{k=1}^{n-1}Q_k+\sum_{k=2}^{n-1}\sum_{l=k+1}^{n-1}Q_kQ_lP+\ldots+$$
$$+\sum_{k=1}^{n-1}\sum_{l=k+1}^{n-1}(1-\delta_{kr})Q_kQ_lP_r+\ldots+$$
$$+(-1)^{n-1}\sum_{k=1}^{n-1}\prod_{r=1}^{k-1}Q_kQ_lP_{n-1}=A_{n-1};$$

$$\sum_{k=1}^{n-1}Q_kP_1+\ldots+\sum_{k=1}^{n-1}(1-\delta_{kr})Q_rP_r+\ldots+$$
$$+\sum_{k=1}^{n-2}Q_kP_{n-1}+\sum_{k=1}^{n-1}Q_kP_0=A_0.$$

If you put $Q_k=q_k\cdot E\left(k=\overline{1,n-1}\right)$ and provide for all scalar q_k the pairs of different distinct numerical values, then the last system of n matrix equations will become linear, relatively unknown P_i to all $\left(i=\overline{0,n-1}\right)$ and will have a single solution. To calculate the solution X on a computer, recurrence formula (23) should be rewritten in the form:

$$X_i=P_0+\sum_{k=1}^{n-1}\left(X_{i-1}-Q_k\right)^{-1}P_k. \tag{24}$$

In order to informally solve and study all of the considered levels, it is necessary to also conduct a study of convergence of the solution and stability of the corresponding branched matrix continued fractions.

We now test the resulting schemes for the matrix equation of the third order

$$X^3+A_2X^2+A_1X+A_0=0$$

where

$$A_2=\begin{pmatrix}2 & -3 & -5\\ 0.22 & 0.251 & 0.25\\ 0.22 & -0.234 & -0.13\end{pmatrix};$$

$$A_1=\begin{pmatrix}1 & 6 & -5\\ 0.25 & 0.22 & 0.251\\ 0.234 & -0.13 & 0.22\end{pmatrix};$$

$$A_0=\begin{pmatrix}136.0000 & 139.0000 & 134.0000\\ -274.0240 & -269.0270 & -282.0490\\ -350.2980 & -358.7900 & -336.5740\end{pmatrix}.$$

Using the MatLab package, this matrix equation was validated using a recursive formula

$$X_k=q_0+p_1\cdot\left(q_1\cdot E+X_{k-1}\right)^{-1}+p_2\cdot\left(q_2\cdot E+X_{k-1}\right)^{-1}.$$

The parameter q_1, q_2 value are selected as any different $q_1=.92$, $q_2=1.92$, and the parameter q_0 is calculated by the formula $q_0=q_1\cdot E+q_2\cdot E-A_2$; p_1 and p_2 are calculated from the system of equations

$$A\cdot P=B$$

where

$$A=\begin{pmatrix}E & E\\ q_2\cdot E & q_1\cdot E\end{pmatrix};$$

$$B=\begin{pmatrix}-A_1+q_1\cdot q_2\cdot E-q_0\cdot q_1\cdot E-q_0\cdot q_2\cdot E\\ -A_0-q_0\cdot q_1\cdot q_2\cdot E\end{pmatrix};$$

$$P=\begin{pmatrix}p_1\\ p_2\end{pmatrix}=A^{-1}\cdot B.$$

As a result of the calculations, the initial approximation $X_0=E$ for the values of parameters $q_1=0.96$ and $q_2=1.92$ for the iteration of 41 provides the calculated approximations of the solution.

$$X_k=\begin{pmatrix}12.3600 & 147.9411 & -107.2121\\ -28.9221 & -290.3746 & 224.4685\\ -36.2185 & -363.6585 & 282.0369\end{pmatrix}.$$

The results of the experiment are presented in the Table 3

Table 3

$\lVert X_k-X_{k-1}\rVert/\lVert X_k\rVert$	Number of iterations k
1.000000e-001	10
1.000000e-002	17
1.000000e-003	19
1.000000e-004	30
1.000000e-005	32
1.000000e-006	41
1.000000e-007	48
1.000000e-008	52
1.000000e-009	54
1.000000e-010	63
1.000000e-011	71

The same initial approximation X_0 for parameter values $q_1 = 10.96$ and $q_2 = 1.92$ for 210 iterations allows for calculating an approximate solution

$$X_k = \begin{pmatrix} 1.2733 & 0.3902 & 2.3444 \\ 6.1634 & 4.4915 & 1.1393 \\ 1.3978 & 4.2253 & 5.4958 \end{pmatrix}.$$

With the same initial approximation X_0 for the values of parameters $q_1 = 9.59076$ and $q_2 = 6.0192$ for 982 iterations, an approximate solution is calculated.

$$X_k = \begin{pmatrix} 21.3944 & 67.9436 & 52.9237 \\ -9.0895 & -37.6830 & 36.5578 \\ -3.3096 & -21.2760 & 25.3785 \end{pmatrix}.$$

The results of the experiment are presented in the Table 4

Table 4

$\|X_k - X_{k-1}\|/\|X_k\|$	Count of iterations k
1.000000e-001	91
1.000000e-002	118
1.000000e-003	137
1.000000e-004	164
1.000000e-005	183
1.000000e-006	210
1.000000e-007	229
1.000000e-008	256
1.000000e-009	275
1.000000e-010	302
1.000000e-011	321

7. Conditions for the termination of the iterative process in solving matrix equations by branched continued fractions

Now consider the signs of the convergence of the iterative process to the solution and substantiation of the criteria for completing the calculation of solutions of matrix equations with the use of the apparatus of branched matrix continued fractions.

If we combine expressions (16), (20) and (23), it is easy to note that all matrix continued fractions formed by them are a partial case of the next law of the composition

$$X_i = P_0 + \sum_{k=1}^{n-1} P_k (X_{i-1} - Q_k)^{-1} P_k. \qquad (25)$$

In fact, the same signs of convergence for matrix branching continued fractions have already been submitted [10], but it is equally important to obtain reliable criteria for the termination of iterations, and the convergence of the process precisely to the solution of a particular equation.

8. Criteria of convergence of matrix branching chain fractions to solutions

It is assumed that the solution of (18) equation on a certain interval is sought after the iterative procedure of the form

$$X_{i+1} = P_0 + \sum_{k=1}^{n-1} P_k (-Q_k + X_i)^{-1} R_k. \qquad (26)$$

Matrix elements P_0, P_k, Q_k, R_k $\left(k = \overline{1, n-1}\right)$ are determined from the system of equations composed of the coefficients of this equation. Then the solution $X = \lim_{i\to\infty} X_i$ can be represented as an infinite one-period matrix continued fraction

$$X = P_0 + \sum_{k=1}^{n-1} P_k \Big(P_0 - Q_k + \sum_{k=1}^{n-1} P_k \big(P_0 - \\ - Q_k + \ldots R_k \big)^{-1} R_k \Big)^{-1} R_k. \qquad (27)$$

Of course, this fraction will be convergent, and even more so, converging to a solution only under certain conditions. To study the convergence problem for the solution of such developments, let us consider a non canonical matrix branching continued fraction

$$D = b_0 + \mathop{D}_{s=1}^{\infty} \sum_{k_{(s)}=1}^{n} a_{k_{(s)}} b_{k_{(s)}}^{-1} c_{k_{(s)}} \qquad (28)$$

and we formulate the criteria for deciding the end of iterations and the convergence of the process precisely to the solution of the specific equations.

The following theorems refer to equation (18).

Theorem 1. If there is only one solution of a polynomial matrix equation in the interval $[-n, n]$, then its development (27) in the iterative procedure (26) in the matrix branch continued fraction with elements satisfying the conditions (4)

$$\left\| b_{k_{(s)}}^{-1} \right\| \le \frac{1}{\left\| a_{k_{(s)}} \right\| \left\| c_{k_{(s)}} \right\| + n} \, (1 \le k_s \le n;\, s = 1, 2, 3, \ldots) \qquad (29)$$

coincides with this solution.
where n means order of matrix.

Proof. The fact of the convergence of the fraction (28) and the region of convergence $D \in [-n, n]$ in the conditions of the theorem is discussed in detail in [4]. To bring the iteration procedure closer to the solution of the equation, a BCD with real elements is introduced:

$$\hat{D} = \hat{b}_0 + \mathop{D}_{s=1}^{\infty} \sum_{k_s=1}^{n} \frac{\hat{a}_{k_{(s)}}}{\hat{b}_{k_{(s)}}}$$

where

$$\hat{a}_{k_{(s)}} = -\left\|a_{k_{(s)}}\right\|, \; \hat{b}_{k_{(s)}} = \left\|a_{k_{(s)}}\right\| + n$$

to all $s = 1, 2, \ldots p$; $k_s = 1, 2, \ldots n$. Next, the inequalities for the suitable fractions are proved

$$\hat{D}_0 > \hat{D}_1 > \hat{D}_2 > \hat{D}_3 > \cdots > \hat{D}_k > \cdots$$

or

$$\left\|D_{m+1} - D_m\right\| < \alpha\left|\hat{D}_1 - \hat{D}_0\right|$$

where $0 < \alpha < 1$.
That is, the principle of compression mappings is fulfilled, which proves the theorem. ☐

Strict proofing is too long and will be provided in a separate publication.

Theorem 2. If there is only one solution of the polynomial matrix equation in the interval

$$\left[-\sum_{k_1=1}^{n}\left\|a_{k_1}\right\|\left\|c_{k_1}\right\|, \;\; \sum_{k_1=1}^{n}\left\|a_{k_1}\right\|\left\|c_{k_1}\right\|\right],$$

then its development (27) in the iterative procedure (26) in the matrix branch continued fraction with elements satisfying the conditions (4)

$$\left\|b_{k_{(s)}}^{-1}\right\| \le \frac{1}{1 + \sum\limits_{k_{s+1}=1}^{n}\left\|a_{k_{(s+1)}}\right\|\left\|c_{k_{(s+1)}}\right\|} \quad (s = 1, 2, 3, \ldots). \tag{30}$$

coincides with this solution.

Proof. The fact of the convergence of the fraction (28) and the region of convergence

$$D \in \left[-\sum_{k_1=1}^{n}\left\|a_{k_1}\right\|\left\|c_{k_1}\right\|, \;\; \sum_{k_1=1}^{n}\left\|a_{k_1}\right\|\left\|c_{k_1}\right\|\right]$$

in the conditions of the theorem is discussed in detail in [4]. And to bring the convergence of the iterative procedure exactly to the solution of the equation, we introduce the BCF with real elements:

$$\hat{D} = \hat{b}_0 + \mathop{D}_{s=1}^{\infty} \sum_{k_s=1}^{n} \frac{\hat{a}_{k_{(s)}}}{\hat{b}_{k_{(s)}}}$$

where

$$\hat{a}_{k_{(s)}} = -\left\|a_{k_{(s)}}\right\|, \; \hat{b}_{k_{(s)}} = \sum_{k_{s+1}=1}^{n-1}\left\|a_{k_{(s)}}\right\| + 1$$

to all $s = 1, 2, \ldots p$; $k_s = 1, 2, \ldots n$. And then again the fact of the monotonicity of the approach fractions of the introduced branching numerical fraction and the fulfillment of the criteria of the compression mapping method are proved, as in the previous theorem. ☐

The indicated signs of convergence indicate the ability to calculate X_m with the given accuracy ε based on inequality checking

$$\left\|X_m - X_{m-1}\right\| \le \varepsilon$$

if one of the following sufficient signs of convergence to the solution is fulfilled. In this case

$$\left\|X_m - X\right\| < \left\|X_m - X_{m-1}\right\|,$$

and therefore $\|X\| \subset \left(\|X\|, -\|X_{m-1}\|\right)$.

It should be noted that in practice, iterative computation processes are often convergent under significantly less stringent conditions. And the results of the numerical experiments convincingly confirm this fact.

So, in particular, for the last example, $\left\|Q_0^{-1}\right\| = 0.09671109$; at the same time $1/(1+\|P_1\|+\|P_2\|) = 0.003842229$. That is, the condition of convergence

$$\left\|Q_0^{-1}\right\| < \frac{1}{1 + \|P_1\| + \|P_2\|}$$

is not even close. But for the whole given sequence, the iteration process of the solution calculation is fast enough.

9. Conclusions

Thus, the approaches to calculating the tuples of solutions of matrix polynomial equations are given, and sufficient criteria for the completion of iterative processes and their convergence to the solution are formulated.

This approach can be applied in the aforementioned generalised Leontiev-Ford simulations.

References

[1] N.A. Nedashkovskiy and T.I. Kroshka, "Computational algorithms for linear balance models of intersectoral ecological-economic interaction", *Cybern. Syst. Anal.* 46, 14–24 (2010). https://doi.org/10.1007/s10559–010–9179–0

[2] N.A. Nedashkovskiy and T.I. Kroshka, "Solution of one class of nonlinear balance models of intersectoral ecological-economic interaction", *Cybern. Syst. Anal.* 47, 684 (2011). https://doi.org/10.1007/s10559–011–9348–9

[3] Kh.D. Ikramov, "Numerical solution of matrix equations [Chislennoye resheniye matrichnykh uravneniy]", M: Nauka, p. 192 (1984). [in Russian]

[4] P.S. Kazimirskyy, "Schedule of matrix polynomials on multipliers [Rozklad matrychnykh mnohochleniv na mnozhnyky]", Kyiv: Naukova dumka, p. 247 (1983). [in Ukrainian]

[5] V.N. Kublanovskaya, "On the spectral problem for polynomial beams of matrices [K spektral'noy zadache dlya polinomial'nykh puchkov matrits]", *Records of scientific seminars of the Leningrad Department of Mathematical Institute of the Academy of Sciences of the USSR*, 83–97 (1978). (in Russian)

[6] I. Matychyn and V. Onyshchenko, "Matrix Mittag-Leffler function in fractional systems and its computation", *Bull. Pol. Ac.: Tech*. 66 (4), 495–500 (2018), https://doi.org/10.24425/124266.

[7] W. Jakowluk, "Optimal input signal design for fractional-order system identification", *Bull. Pol. Ac.: Tech*. 67 (1), 37–44 (2019), https://doi.org/10.24425/bpas.2019.127336.

[8] S. Elloumi, I. Sansa and N. B. Braiek, "On the stability of optimal controlled systems with SDRE approach", *International Multi-Conference on Systems*, *Signals & Devices*, Chemnitz, 2012, pp. 1-5. https://doi.org/10.1109/SSD.2012.6198040.

[9] J. Bernat, J. Kołota, S. Stępień, and P. Superczyńska, "Suboptimal control of nonlinear continuous-time locally positive systems using input-state linearization and SDRE approach", *Bull. Pol. Ac.: Tech*. 66(1), 17–22 (2018), https://doi.org/10.24425/119054.

[10] M.O. Nedashkovskyy, "Signs of convergence of matrix branching chain fractions [Oznaky zbizhnosti matrychnykh hillyastykh lantsyuhovykh drobiv]", Mathematical methods and physical-mechanical fields. Lviv, 46 (4), 50–56 (2003). [in Ukrainian]