Линейни и квадратни матрични уравнения

Георги Георгиев, Иван Иванов, Милена Михайлова, Цветелина Динкова

Linear and Quadratic Matrix Equations: *The paper is devoted to a direct approach for studying of matrix equations. Using the computer algebra system* MATLAB *we give short algorithms for solving general linear, Sylvester and Lyapunov matrix equations. Some cases of quadratic matrix equations and algebraic Riccati equations are also considered. This way for a presentation of matrix equations is suitable for the students in Economics and Engineering .*

Key words: *Sylvester Matrix Equations, Lyapunov Matrix Equations, algebraic Riccati equation.*

ВЪВЕДЕНИЕ

Развитието на компютърните технологии и все по-широкото им приложение от една страна, а от друга - намаляването на аудиторните часове по Линейна алгебра и аналитична геометрия за нематическите специалности в университетите, води до необходимостта от провеждането на лабораторни упражнения по дисциплината.

Целта на настоящата работа е да се представи един начин, по който студентите от икономически и технически специалности могат да се запознаят с различни видове матрични уравнения и да придобият умения за решаването им със системата MATLAB. Представени са алгоритъм за решаване на общото линейно матрично уравнение и са дадени примери за решаване на частни случаи на квадратно матрично уравнение. Специално внимание е отделено на матричните уравнения на Силвестър, Ляпунов и Рикати, които се използват широко в практиката.

1. Операции с матрици. Елементарните действия с матрици, познати в линейната алгебра, се извършват в MATLAB с помощта на обикновените аритметични оператори +, - , * , ^ .

C=A+B – събиране на две матрици; $c_{ij} = a_{ij} + b_{ij}$

C=A-B – изваждане на две матрици; $c_{ij} = a_{ij} - b_{ij}$

C=A*λ – умножение на матрица с число; $c_{ij} = a_{ij} * \lambda$

C=A*B – умножение на две матрици;

C=A^n – степенуване ($n \in \mathbf{N}$);

C=A' – пресмятане на комплексно спрегната матрица.

Събирането и изваждането на матрици е възможно, ако те са от една и съща размерност. Под произведение на матрицата $A = (a_{ij})$ от тип $m \times s$ с матрицата $B = (b_{ij})$ от тип $s \times n$, се разбира матрицата $C = (c_{ij})$ от тип $m \times n$ с елементи $c_{ij} = \sum_{k=1}^{s} a_{ik} b_{kj}$ (i=1,…,m; j=1,…,n).

Пример 1. Ако $A = \begin{pmatrix} -1 & 4 & 2 \\ 3 & -1 & -2 \end{pmatrix}$ и $B = \begin{pmatrix} -2 & 1 & -1 \\ 1 & 0 & 5 \\ 3 & -4 & -2 \end{pmatrix}$ ще пресметнем произведенията AB и BA.

```
>> A=[-1 4 2; 3 -1 -2]; B=[-2 1 -1; 1 0 5; 3 -4 -2];
>> C=A*B
C =
    12   -9   17
   -13   11   -4

>> C1=B*A
```

??? Error using ==> *
Inner matrix dimensions must agree.

Произведението на две матрици е възможно само, когато броя на стълбовете на първия множител е равен на броя на редовете на втория множител. В частност произведение на квадратни матрици от един и същи ред винаги съществува и е матрица от същия ред. Произведението на квадратни матрици дава възможност да определим A^k, $k \in \mathbf{N}$.

В MATLAB обратна на квадратна неособена матрица се намира чрез вградената функция *inv(A).*

Освен описаните вече действия, в MATLAB са дефинирани и действията деление на матрици отляво A\B и деление на матрици отдясно B/A, които се използват съответно при решаването на матричните уравнения AX=B и XA=B (с изключение на случая, когато A е особена).

Пример 2. Да решим матричните уравнения

а) $\begin{pmatrix} 2 & -1 & 3 \\ -1 & 0 & 4 \\ 5 & 1 & -2 \end{pmatrix} X = \begin{pmatrix} 10 & 4 & 16 \\ 16 & 3 & 21 \\ -6 & 4 & -18 \end{pmatrix}$ **б)** $X \begin{pmatrix} -1 & 2 & 0 \\ 4 & -3 & 2 \\ 1 & -1 & 3 \end{pmatrix} = \begin{pmatrix} -9 & 8 & 9 \\ 2 & -1 & -4 \\ -6 & 11 & 10 \end{pmatrix}$

а) >> A=[2 -1 3; -1 0 4; 5 1 -2]; B=[10 4 16; 16 3 21; -6 4 -18];

```
>> X=A\B
X =
    0.0000    1.0000   -1.0000
    2.0000    1.0000   -3.0000
    4.0000    1.0000    5.0000
```

б) >> A=[-1 2 0; 4 -3 2; 1 -1 3]; B=[-9 8 9; 2 -1 -4; -6 11 10];

```
>> X=B/A
X =
   2   -3    5
   0    1   -2
   6   -1    4
```

2. Общо линейно матрично уравнение. Най-общото линейно матрично алгебрично уравнение може да се запише във вида $\sum_{i=1}^{n} A_i X B_i = C$, където $A_i \in \mathbf{C}^{p\times m}$, $B_i \in \mathbf{C}^{n\times q}$, $C \in \mathbf{C}^{p\times q}$ са дадени матрични коефициенти, а $X \in \mathbf{C}^{m\times n}$ е неизвестна матрица. Това уравнение може да се сведе до матричното линейно уравнение Mx=c, където $M = \sum_{i=1}^{n} B_i^T \otimes A_i \in \mathbf{C}^{pq\times mn}$, c е вектор-стълб с размерност pq, а $A \otimes B$ е кронекерово произведение на матрици. За подробности виж [1] глава 8.

Предлагаме файл-функцията LME с променлив брой входящи аргументи, която ако се запише като M-файл със същото име, могат да се решават многократно общи линейни матрични уравнения с произволен брой събираеми.

```
function X=LME(C,varargin)
n=length(varargin);
v=[size(varargin{1});size(varargin{2})];
M=zeros(v(1)*v(4), v(2)*v(3));
for i=1:2:n
   A=varargin{i};
   B=varargin{i+1};
   KR=kron(B.',A);
   M=M+KR;
end
```

```
c=C(:); Mc=[M c];
if rank(M)~=rank(Mc)
   disp('Уравнението няма решение.');
elseif v(1)*v(4)~=v(3)*v(2)|rank(M)==min(size(M))
   X=reshape(M\c,v(3),v(2));
else x=pinv(M)*c;
X=reshape(x,v(3),v(2));
end
```

На мястото на varargin трябва да се изброят със запетая, всички дадени матрични коефициенти от последователните събираеми в лявата част на общото матрично линейно уравнение. Ако някой от коефициентите липсва, той се записва като единичната матрица с подходяща размерност.

Пример 3. Да намерим матрицата X от уравненията:

а) $$\begin{pmatrix} 1 & -2 \\ -1 & 3 \\ -3 & 1 \end{pmatrix} X \begin{pmatrix} 1 & -1 & 2 & -3 \\ -2 & 1 & 5 & 1 \\ 4 & -2 & 1 & -1 \end{pmatrix} + \begin{pmatrix} 2 & -1 \\ -4 & 2 \\ 1 & 3 \end{pmatrix} X \begin{pmatrix} -2 & 1 & 3 & -4 \\ 1 & 6 & -2 & -3 \\ -4 & 2 & -1 & 3 \end{pmatrix} = \begin{pmatrix} -28 & -14 & -1 & -8 \\ 69 & 20 & -6 & 3 \\ -98 & 85 & -24 & 63 \end{pmatrix}$$

```
>> A1=[1 -2; -1 3; -3 1]; B1=[1 -1 2 -3; -2 1 5 1; 4 -2 1 -1];
>> A2=[2 –1; -4 2; 1 3]; B2=[-2 1 3 -4; 1 6 -2 -3; -4 2 -1 3];
>> C=[-28 -14 -1 -8; 69 20 –6 3; -98 85 -24 63];
>> X=LME(C,A1,B1,A2,B2)
X =
   2.0000  -1.0000   4.0000
  -1.0000   2.0000   3.0000
```

б) $$\begin{pmatrix} -1 & 2 \\ 1 & 4 \\ -2 & 3 \end{pmatrix} X \begin{pmatrix} 1 & 5 & 4 \\ 0 & -1 & 1 \\ -2 & 2 & -3 \\ 4 & -3 & -1 \end{pmatrix} + \begin{pmatrix} 4 & 1 \\ -2 & 3 \\ 1 & -2 \end{pmatrix} X \begin{pmatrix} 2 & -1 & -2 \\ 1 & 2 & 3 \\ -1 & 0 & -1 \\ 3 & -5 & 1 \end{pmatrix} + X \begin{pmatrix} 0 & 2 & 5 \\ 1 & -1 & 1 \\ -1 & -2 & -1 \\ 3 & 4 & 0 \end{pmatrix} + \begin{pmatrix} 3 & -1 \\ -2 & 4 \\ 1 & -2 \end{pmatrix} X = \begin{pmatrix} 75 & 18 & 72 \\ 213 & -104 & 81 \\ 34 & 66 & 67 \end{pmatrix}$$

```
>> A1=[-1 2; 1 4; -2 3]; B1=[1 5 4; 0 -1 1; -2 2 -3; 4 -3 -1];
>> A2=[4 1; -2 3; 1 -2]; B2=[2 -1 -2; 1 2 3; -1 0 -1; 3 -5 1];
>> A3=eye(size(A1)); B3=[0 2 5; 1 -1 1; -1 -2 -1; 3 4 0];
>> A4=[3 -1; -2 4; 1 -2]; B4= eye(size(B1));
>> C=[75 18 72; 213 -104 81; 34 66 67];
>> X=LME(C,A1,B1,A2,B2,A3,B3,A4,B4)
X =
  -1.0000   2.0000  -3.0000  -1.0000
   3.0000  -1.0000  -5.0000   4.0000
```

Важни за практиката са линейните матрични уравнения на Силвестър и Ляпунов, които се получават като следствия от общото линейно матрично уравнение. Техните решения, освен с вградените функции *lyap* и *dlyap*, могат да се намерят с предложената по-горе функция LME. Матриците, които участват в уравненията на Силвестър са $A \in \mathbf{C}^{m\times m}$, $B \in \mathbf{C}^{n\times n}$ и $C, X \in \mathbf{C}^{m\times n}$.

Уравнение на Силвестър за непрекъснати системи AX+XB=C. Решението е X=LME(C,A,eye(size(B)), eye(size(A)),B)

Уравнение на Силвестър за дискретни системи AXB-X=C. Решението е X=LME(C,A, B,-eye(size(A)),eye(size(B)))

При уравненията на Ляпунов всички участващи матрици са квадратни с една и съща размерност. В реалния случай матрицата C е симетрична ($C^T = C$), $A^* = A^T$. В комплексния случай C е ермитова ($C^H = C$), $A^* = A^H$.

Уравнение на Ляпунов за непрекъснати системи $A^*X + XA = C$. Решението е X=LME(C,A',eye(size(A)), eye(size(A)),A)

Уравнение на Ляпунов за дискретни системи $A^*XA - X = C$. Решението е X=LME(C,A',A,-eye(size(A)), eye(size(A)))

Обобщено уравнение на Ляпунов за непрекъснати системи $A^*XE + E^*XA = C$. Решението е X=LME(C,A',E,E',A)

Обобщено уравнение на Ляпунов за дискретни системи $A^*XA - E^*XE = C$. Решението е X=LME(C,A',A, - E', E)

3. Квадратно матрично уравнение. Квадратно матрично уравнение се дефинира с равенството $AX^2 + BX + C = 0$, където A, B, $C \in \mathbf{C}^{n\times n}$. С помощта на вградената функция за пресмятане на корен квадратен от матрица *sqrtm* ще решим най-простия вид квадратно матрично уравнение $X^2 = C$. Наличието на последната буква m в името й, означава че тази функция оперира върху матрицата като цяло, а не поелементно.

Пример 4. $X^2 = \begin{pmatrix} -8 & -4 \\ 3 & -11 \end{pmatrix}$

```
>> C=[-8 -4; 3 -11];
>> X=sqrtm(C)
X =
    2.0000 + 0.0000i  -4.0000 + 0.0000i
    3.0000 + 0.0000i  -1.0000 - 0.0000i
```

Очевидно, че и –X е решение на уравнението. В най-общия случай броя на решенията на $X^2 = C$ е 2^n, където n е размерността на матрицата C. Ако C е особена, уравнението може и да няма решение.

Ще разгледаме квадратното матрично уравнение $AX^2 + BX + C = 0$, само в случая когато може да се реши с помощта на познатите формули за корените на квадратното скаларно уравнение. Това е точно случая, когато A=E (единичната матрица), B и C комутират и съществува корен квадратен от $D = B^2 - 4C$ (виж [2]).

Пример 5. $X^2 + \begin{pmatrix} -4 & -8 & -7 \\ 3 & 6 & 5 \\ -2 & -5 & -4 \end{pmatrix} X + \begin{pmatrix} 1 & 3 & 2 \\ 2 & 2 & -1 \\ -3 & -4 & 0 \end{pmatrix} = 0$. Реализацията за намиране на X с MATLAB е:

```
>>B=[-4 –8 –7; 3 6 5; -2 –5 –4]; C=[1 3 2; 2 2 –1; -3 –4 0];
>>D=B^2-4*C;
>>X1=(-B+sqrtm(D))/2
X1 =
   3.3266 + 0.2301i    6.3170 + 0.3485i   5.5404 + 0.0308i
  -2.4340 + 1.2400i  -4.6687 + 1.8780i  -3.9664 + 0.1660i
   1.6575  - 1.5577i   4.0707 -  2.3591i   3.3474  - 0.2085i
>>X2=(-B-sqrtm(D))/2
X2 =
   0.6734  - 0.2301i    1.6830 - 0.3485i    1.4596 - 0.0308i
  -0.5660  - 1.2400i  -1.3313 - 1.8780i   -1.0336 - 0.1660i
   0.3425 + 1.5577i   0.9293 + 2.3591i   0.6526 + 0.2085i
```

По - подробно квадратните матрични уравнения са разгледани в [3] .

4. Решаване на алгебричното уравнение на Рикати с MATLAB. Използват се вградените функции *care* и *dare.*

Уравнението на Рикати за непрекъснати системи $A^TX + XA - XBB^TX + Q = 0$, където A, $B \in \mathbf{R}^{n\times n}$ и Q е симетрична матрица се решава с X=care(A,B,Q).

X=dare(A,B,Q,R) решава уравнението на Рикати за дискретни системи $A^TXA - X - A^TXB(B^TXB + R)^{-1}B^TXA + Q = 0$, където $A \in \mathbf{R}^{n\times n}$, $B \in \mathbf{R}^{n\times m}$, а Q и R са симетрични.

Пример 6. Ще пресметнем уравненията на Рикати

а) $$\begin{pmatrix}1 & 2 & 0\\ 1 & 0 & 1\\ 0 & -5 & 2\end{pmatrix}X + X\begin{pmatrix}1 & 1 & 0\\ 2 & 0 & -5\\ 0 & 1 & 2\end{pmatrix} - X\begin{pmatrix}2 & 0 & -1\\ -2 & 1 & 3\\ 0 & 1 & 0\end{pmatrix}\begin{pmatrix}2 & -2 & 0\\ 0 & 1 & 1\\ -1 & 3 & 0\end{pmatrix}X + \begin{pmatrix}7 & -2 & 40\\ -2 & 30 & -11\\ 40 & -11 & 92\end{pmatrix} = 0$$

```
>> A=[1 1 0; 2 0 -5; 0 1 2];
>> B=[2 0 -1; -2 1 3; 0 1 0];
>> Q=[7 -2 40; -2 30 -11; 40 -11 92];
>> X=care(A,B,Q)
X =
   3.6465   1.8429   0.4176
   1.8429   2.3979  -1.2319
   0.4176  -1.2319  12.4755
```

б) $A^T XA - X - A^T XB(B^T XB + R)^{-1}B^T XA + Q = 0$, където

$$A = \begin{pmatrix}2 & -1\\ 0 & 3\end{pmatrix},\quad B = \begin{pmatrix}3 & -1\\ 1 & 0\end{pmatrix},\quad Q = \begin{pmatrix}-2 & -21\\ -21 & -19\end{pmatrix},\quad R = \begin{pmatrix}1 & 2\\ 2 & -1\end{pmatrix}.$$

```
>> A=[2 -1; 0 3]; B=[3 -1; 1 0]; Q=[-2 -21; -21 -19]; R=[1 2; 2 -1];
>> X=dare(A,B,Q,R)
X =
  -6.2469  -14.5900
 -14.5900   11.9121
```

ЗАКЛЮЧЕНИЕ

Представения подход дава възможност, като се използват новите технологии, за кратко време студентите да се запознаят с възможностите на системата MATLAB и като се освободят от сложните аритметични пресмятания, по-бързо и целенасочено да усвоят различните видове матрични уравнения.

Благодарности. Работата е частично финансирана по проекти РД-05-474/07.05.2008 и РД-05-475/07.05.2008 на Шуменски Университет.

ЛИТЕРАТУРА

[1] Константинов М. М. Елементи на линейната алгебра: вектори и матрици, София, УАСГ, 2000

[2] Higham, N. J., H.-M. Kim. Numerical analysis of a quadratic matrix equation. IMA Journal of Numerical Analysis (2000) **20**, 499-519 .

[3] Meini B. Nonlinear matrix equations and structured linear algebra, Linear Algebra and its Applications, **413**, (2006), 440-457.

За контакти:

Доц. д-р Георги Хр. Георгиев, ас. Иван Сл. Иванов, ас. Милена Н. Михайлова, ас. Цветелина Л. Динкова, Катедра “Алгебра и геометрия”, ФМИ, Шуменски университет “Епископ Константин Преславски”, e-mails: g.georgiev@fmi.shu-bg.net; slaveicov@abv.bg; nicolova_m@abv.bg; cvetelina_d@abv.bg;

Докладът е рецензиран.