

# Contents

---

- [ex 1 Tema 4](#)
- [ex 2 Tema 4 Varianta 5](#)

## ex 1 Tema 4

---

```
%a
type('metCholesky');
%b
A=[1 2 3; 2 5 8; 3 8 14]
b=[-5;-14;-25]
metCholesky(A,b);
%c
fprintf("\n");
fprintf("matricea L este:");
disp(chol(A,'lower'));

function [L,x] = metCholesky(A,b)

% metCholesky calculeaza matricea L unde A=L*transpose(L)

% si vect x unde A*x=b


% Synopsis: [L,x] = metCholesky(A)

%

% Input:      A = matrice patratica

%            b = vectorul termenilor liberi

% Output:     L = matrice inferior triunghiulara

%            cu elementele L(k,k)=1

%

%

%

%            x = vectorul solutie a sistemului

%            A * x = b


% Se verifica daca matricea este patratica

[n,m]=size(A);

if n~=m

    warning('A nu e patratica')

end

L=eye(n);

% Se verifica daca matricea este simetrica

B=transpose(A);
```

```

if A~=B

    warning("A nu e simetrica");

    x=[];

    return

end

%afla reprezinta minorii principali ai matricei

alfa=A(1,1);

if alfa<=0

    warning('A nu e pozitiv definita');

    x=[];

    return

end

L(1,1)=sqrt(A(1,1));

for i=2:n

    L(i,1)=A(i,1)/L(1,1);

end

for k=2:n

    suma=0;

    for s=1:k-1

        suma=suma+L(k,s)^2;

    end

    alfa=A(k,k)-suma;

    if alfa<=0

        warning('A nu e pozitiv definita');

    end

    L(k,k)=sqrt(alfa);

    for i=k+1:n

        sum=0;

        for j=1:k-1

            sum=sum+L(i,j)*L(k,j);

        end

        L(i,k)=(A(i,k)-sum)/L(k,k);

    end

end
end

```

```

y=SubsAsc(L,b);

x=SubsDesc(transpose(L),y);

fprintf('L este:\n');

disp(L);

fprintf('solutia este:\n');

for i=1:n

    disp(x(i));

end

```

A =

1	2	3
2	5	8
3	8	14

b =

-5  
-14  
-25

L este:

1	0	0
2	1	0
3	2	1

solutia este:

1  
  
0

-2

matricea L este:

	1	0	0
2	1	0	
3	2	1	

## ex 2 Tema 4 Varianta 5

```
%a
%definirea vectorului a si a matricei A s-a facut in functia fctA
n=6;
A=fctA(n);
type('fctA');
%b,c,d
%vectorul b s-a calculat in functia fctb
b=fctb(n);
%folosirea criterului lui Sylvester, factorizarea si rezolvarea sistemului
%se fac cu functia metCholesky(A,b) unde alfa reprezinta minorii principali
%ai matricei date
metCholesky(A,b);
type('metCholesky');
fprintf("folosirea criterului lui Sylvester, factorizarea si rezolvarea sistemului se fac cu fct
metCholesky(A,b) unde alfa=minorii principali ai matricei date");
fprintf("\n functia pt construirea vectorului b: \n");
type('fctb');
```

vectorul a este:

729  
243  
81  
27  
9  
3

A este:

729	243	81	27	9	3
243	729	243	81	27	9
81	243	729	243	81	27
27	81	243	729	243	81
9	27	81	243	729	243
3	9	27	81	243	729

```

function [A] = fctA(n)

for i=1:n

    a(i)=3^(n-i+1);

end

A=eye(n);

k=0;

for i=1:n-1

    A(i,i)=a(1);

    for j=i+1:n

        A(i,j)=a(j-k);

        A(j,i)=A(i,j);

    end

    k=k+1;

end

A(n,n)=a(1);

fprintf('vectorul a este:\n');

a=transpose(a);

disp(a);

fprintf('A este:\n')

disp(A);

end

vectorul b este:

    1
    4
    9
   16
   25
   36

L este:

    27.0000         0         0         0         0         0

```

9.0000	25.4558	0	0	0	0
3.0000	8.4853	25.4558	0	0	0
1.0000	2.8284	8.4853	25.4558	0	0
0.3333	0.9428	2.8284	8.4853	25.4558	0
0.1111	0.3143	0.9428	2.8284	8.4853	25.4558

solutia este:

-5.1440e-04

0.0017

0.0051

0.0099

0.0161

0.0427

```
function [L,x] = metCholesky(A,b)

% metCholesky calculeaza matricea L unde A=L*transpose(L)

% si vect x unde A*x=b

% Synopsis: [L,x] = metCholesky(A)

%

% Input:    A = matrice patratica

%          b = vectorul termenilor liberi

% Output:   L = matrice inferior triunghiulara

%          cu elementele L(k,k)=1

%

%

%

%          x = vectorul solutie a sistemului
```

```

%           A * x = b

% Se verifica daca matricea este patratica

[n,m]=size(A);

if n~=m

    warning('A nu e patratica')

end

L=eye(n);

% Se verifica daca matricea este simetrica

B=transpose(A);

if A~=B

    warning("A nu e simetrica");

    x=[];

    return

end

%afla reprezinta minorii principali ai matricei

alfa=A(1,1);

if alfa<=0

    warning('A nu e pozitiv definita');

    x=[];

    return

end

L(1,1)=sqrt(A(1,1));

for i=2:n

    L(i,1)=A(i,1)/L(1,1);

end

for k=2:n

    suma=0;

    for s=1:k-1

        suma=suma+L(k,s)^2;

    end

    alfa=A(k,k)-suma;

    if alfa<=0

```

```

        warning('A nu e pozitiv definita');

    end

    L(k,k)=sqrt(alfa);

    for i=k+1:n

        sum=0;

        for j=1:k-1

            sum=sum+L(i,j)*L(k,j);

        end

        L(i,k)=(A(i,k)-sum)/L(k,k);

    end

end

y=SubsAsc(L,b);

x=SubsDesc(transpose(L),y);

fprintf('L este:\n');

disp(L);

fprintf('solutia este:\n');

for i=1:n

    disp(x(i));

end

```

folosirea criteriului lui Sylvester, factorizarea si rezolvarea sistemului se fac cu fct metCholesky(A,b) unde alfa=minorii principali ai matricei date

functia pt construirea vectorului b:

```

function [b] =fctb(n);

for i=1:n

    b(i)=i*i;

end

fprintf('vectorul b este:\n');

b=transpose(b);

disp(b);

end

```



