Tema lab03

Table of Contents

Metoda substitutiei descendente

```
function [x] = SubsDesc(A, b)
   n = length(b);
   x(n) = b(n)/A(n,n);
   for k=n-1:-1:1
      suma = 0;
      for j=k+1:n
          suma = suma + A(k,j)*x(j);
      end
      x(k) = (b(k) - suma)/A(k,k);
   end
   x = x';
end
```

Metoda substitutiei ascendente

```
function [x] = SubsAsc(A, b)
   n = length(b);
   x(1) = b(1)/A(1,1);
   for k=2:n
      suma = 0;
      for j=1:k-1
          suma = suma + A(k,j)*x(j);
      end
```

```
        x(k) = (b(k) - suma)/A(k,k);
     end
     x = x';
end
```

Gauss fara pivotare

```
function [x] = GaussFaraPiv(A,b)
  n = length(b);
  A = [A b];
  for k=1:n-1
      for p=k:n
          if A(p,k)~=0
              break
          end
      end
      if p~=k
          A([p k],:) = A([k p],:);
      end
      for l=k+1:n
          mlk = A(l,k)/A(k,k);
          A(l,:) = A(l,:) - mlk*A(k,:);
      end
  end
  if A(n,n)==0
      disp 'Sistem incompatibil sau sistem compatibil nedeterminat';
      return;
  end
  x = SubsDesc(A(:,1:n),A(:,n+1));
end
```

Gauss pivotare partiala

```
function [x] = GaussPivPart(A,b)
  n = length(b);
  A = [A b];
  for k=1:n-1
      apk = 0;
      for j=k:n
          if abs(A(j,k))>apk
              apk = abs(A(j,k));
              p = j;
          end
      end
      if apk==0
        disp 'Sistem incompatibil sau sistem compatibil nedeterminat';
        return;
      end
      if p~=k
      A([p k],:) = A([k p],:);
    end
    for l=k+1:n
      mlk = A(l,k)/A(k,k);
      A(l,:) = A(l,:) - mlk*A(k,:);
    end
  end
  if A(n,n)==0
    disp 'Sistem incompatibil sau sistem compatibil nedeterminat';
    return;
  end
  x = SubsDesc(A(:,1:n),A(:,n+1));
end
```

Gauss pivotare totala

```
function [x] = GaussPivTot(A,b)
  n = length(b);
  A = [A b];
  index = 1:n;
  for k=1:n-1
    apm = 0;
    for i=k:n
      for j=k:n
        if abs(A(i,j))>apm
          apm = abs(A(i,j));
          p = i;
          m = j;
        end
      end
    end
    if apm==0
     disp 'Sistem incompatibil sau sistem compatibil nedeterminat';
     return;
    end
    if p~=k
      A([p k],:) = A([k p],:);
    end
    if m~=k
      A(:,[m k]) = A(:,[k m]);
      index([m k]) = index([k m]);
    end
    for l=k+1:n
      mlk = A(l,k)/A(k,k);
      A(l,:) = A(l,:) - mlk*A(k,:);
    end
  end
  if A(n,n)==0
    disp 'Sistem incompatibil sau sistem compatibil nedeterminat';
    return;
  end
  x_tmp = SubsDesc(A(:,1:n),A(:,n+1));
  for i=1:n
      x(index(i)) = x_tmp(i);
  end
  x = x';
end
```

InvGPP

```
function [invA, detA] = InvGPP(A)
  n = size(A, 1);
  A = [A eye(n)];
  for k=1:n-1
      apk = 0;
      for j=k:n
          if abs(A(j,k))>apk
              apk = abs(A(j,k));
              p = j;
          end
      end
      if apk==0
        disp 'Sistem incompatibil sau sistem compatibil nedeterminat';
        return;
      end
      if p~=k
          A([p k],:) = A([k p],:);
      end
      for l=k+1:n
          mlk = A(l,k)/A(k,k);
          A(l,:) = A(l,:) - mlk*A(k,:);
      end
  end
  if A(n,n)==0
      disp 'Sistem incompatibil sau sistem compatibil nedeterminat';
      return;
  end
  invA = [];
  detA = 1;
  for i=1:n
      invA = [invA SubsDesc(A(:,1:n),A(:,n+i))];
      detA = detA*A(i,i);
  end
end
```

Factorizarea LU

```
function [L, U, w] = FactLU(A)
  n = size(A, 1);
  L = eye(n);
  for k=1:n-1
     p=k;
     max = abs(A(p,k));
     for i=k:n
         if abs(A(i,k))>max
             max = abs(A(i,k));
             p=i;
         end
     end
     if A(p,k)==0
         disp 'Sistem incompatibil';
         return;
     end
     w(k) = p;
     if p~=k
         A([p k],:) = A([k p],:);
     end
     for l=k+1:n
         L(l,k) = A(l,k)/A(k,k);
         A(l,:) = A(l,:) - L(l,k)*A(k,:);
     end
     if k>1
         L([p k], 1:k-1) = L([k p], 1:k-1);
     end
  end
  if A(n,n)==0
      disp 'Sistem incompatibil';
      return;
  end
  U = A;
end
```

Factorizarea Cholesky

```
function [L] = FactCholesky(A)
  a = A(1,1);
  n = size(A,2);
  if a <= 0
      disp('A nu admite factorizarea Cholesky')
      return
  end
  L(1,1) = sqrt(a);
  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)*L(k,s);
      end
      a = A(k,k) - suma;
      if a <= 0
        disp('A nu admite factorizarea Cholesky')
        return
      end
      L(k,k) = sqrt(a);
      for i=k+1:n
          suma = 0;
          for s=1:k-1
              suma = suma + L(i,s)*L(k,s);
          end
          L(i,k) = (A(i,k)-suma)/L(k,k);
      end
  end
end
```

Factorizarea Cholesky 2

```
function [L] = FactCholesky2(A)
  n = size(A,2);
  if n==1
      L = A;
  else
      a = A(1,1);
      if a<=0
        disp('A nu admite factorizarea Cholesky')
        return
      end
      L(1,1) = sqrt(a);
      L(2:n,1) = A(2:n,1)/L(1,1);
      newA = A(2:n,2:n) - L(2:n,1)*(L(2:n,1)');
      L = [L [linspace(0,0,n-1) ; FactCholesky2(newA)]];
  end
end
```

Exercitiul 2

```
A = [1 2 3
     0 2 3
     0 0 6];
b = [6; 5; 6];
SubsDesc(A,b)


ans =

     1
     1
     1
```

Exercitiul 3

```
A1 = [0 1 1
     2 1 5
     4 2 1];
b1 = [3; 5; 1];
A2 = [0 1 -2
      1 -1 1
      1 0 -1];
b2 = [4;6;2];

GaussFaraPiv(A1,b1)'
GaussFaraPiv(A2,b2)

GaussPivPart(A1,b1)'
GaussPivPart(A2,b2)

GaussPivTot(A1,b1)'
GaussPivTot(A2,b2)

eps = 10^(-20);
Aeps = [eps 1
        1 1];
beps = [1;2];
GaussFaraPiv(Aeps,beps)'
GaussPivPart(Aeps,beps)'

C = 10^20;
AC = [1 C
      1 1];
bC = [C; 2];
GaussPivPart(AC,bC)'
GaussPivTot(AC,bC)'
```

```
ans =

    -1     2     1

Sistem incompatibil sau sistem compatibil nedeterminat

ans =

    -1     2     1

Sistem incompatibil sau sistem compatibil nedeterminat

ans =

    -1     2     1

Sistem incompatibil sau sistem compatibil nedeterminat
```

```
ans =

     0     1


ans =

     1     1


ans =

     0     1


ans =

     1     1
```

Exercitiul 4

```
A = [4 2 2
    2 10 4
    2 4 6];
b = [12; 30; 10];
[invA, detA] = InvGPP(A);
x = invA*b;

invA
detA
invA*A
x
```

```
invA =

    0.3056   -0.0278   -0.0833
   -0.0278    0.1389   -0.0833
   -0.0833   -0.0833    0.2500


detA =

   144


ans =

    1.0000   -0.0000   -0.0000
    0.0000    1.0000    0.0000
         0         0    1.0000
```

```
x =

     2
     3
    -1
```

Exercitiul 6

```
A = [1 2 -1
    2 4 7
    -1 2 5];
b = [2;13;10];

[L, U, w] = FactLU(A);
bp = b;
lenw = length(w);
for i=1:lenw
    bp([i w(i)]) = bp([w(i) i]);
end

y = SubsAsc(L,bp);
x = SubsDesc(U,y);
x


x =

    -1
     2
     1
```

Exercitiul 8

```
A = [1 2 3
    2 5 8
    3 8 14];
b = [-5; -14; -25];
L = FactCholesky(A);
L
y = SubsAsc(L, b);
x = SubsDesc(L', y);

x


L =

     1     0     0
     2     1     0
     3     2     1
```

```
x =

     1
     0
    -2
```

Exercitiul 9

```
A = [1 2 3
    2 5 8
    3 8 14];
b = [-5; -14; -25];
L = FactCholesky2(A);
L
y = SubsAsc(L, b);
x = SubsDesc(L', y);

x
```

```
L =

     1     0     0
     2     1     0
     3     2     1


x =

     1
     0
    -2
```

Published with MATLAB® R2018a