Numerical Methods

Matlab Programming

Cramers Rule (Matlab)

```
function [x,s] = CramerRule(A,b)
% Checking if the Matrix is Singular
if det(A) == 0
  s='The Matrix A is Singular';
   for i=1:max(size(A))
      x(i) = 0;
   end
   return
end
% Initializing Default Value of Variable
s = 'Solution is';
% Finding Length of A
k = max(size(A));
% Applying Cramers Rule
for i = 1:k
  matr = A;
  matr(:,i) = b;
  x(i) = det(matr)/det(A);
end
% Final Result X
x = x';
end
```

LU Method (Matlab)

```
function[L,U,x]=LU_Parker(A,B)
[m n]=size(A);
if (m ~= n )
disp ( 'LR2 error: Matrix must be square' );
return;
end;
   % Part 2 : Decomposition of matrix into L and U
 L=zeros(m,m);
 U=zeros(m,m);
 for i=1:m
 % Finding L
 for k=1:i-1
 L(i,k)=A(i,k);
 for j=1:k-1
 L(i,k)= L(i,k)-L(i,j)*U(j,k);
 end
 L(i,k) = L(i,k)/U(k,k);
 end
 % Finding U
 for k=i:m
 U(i,k) = A(i,k);
 for j=1:i-1
 U(i,k)= U(i,k)-L(i,j)*U(j,k);
 end
 end
 end
```

```
for i=1:m
L(i,i)=1;
end
% Program shows U and L
U
L
% Now use a vector y to solve 'Ly=b'
y=zeros(m,1);
y(1)=B(1)/L(1,1);
for i=2:m
y(i)=-L(i,1)*y(1);
for k=2:i-1
y(i)=y(i)-L(i,k)*y(k);
y(i)=(B(i)+y(i))/L(i,i);
end;
end;
% Now we use this y to solve Ux = y
x=zeros(m,1);
x(1)=y(1)/U(1,1);
for i=2:m
x(i)=-U(i,1)*x(1);
for k=i:m
    x(i)=x(i)-U(i,k)*x(k);
    x(i)=(y(i)+x(i))/U(i,i);
end;   end
```

Tri-Diagonal Method (Matlab)

```
function [x,s] = TriDiag( A, b )
%A= [-2 1 0 0;1 -2 1 0;0 1 -2 1;0 0 1 -2]
%b=[0.04 0.04 0.04 0.04]
% Checking if the Matrix is Singular
if det(A)==0
   s='The Matrix A is Singular';
   for i=1:max(size(A))
      x(i) = 0;
   end
   break
end
% Initializing Default Values of Variables
s='Answer is:';
bb(1)=0;
% Finding Diagonal, Upper Diagonal and Lower Diagonal Values.
for i=1:max(size(A))
   for j=1:max(size(A))
      if i==j
         a(i)= A(i,j);
      elseif i==j+1
         bb(i)= A(i,j);
      elseif j==i+1
         c(i)= A(i,j);
      end
   end
end
```

```
% Last value of C matrix is zero.

c(max(size(A))) = 0;

% Initializing default values of Variables

len = length(b);

v = zeros(len,1);

x = v;

w = a(1);

x(1) = b(1)/w;

% Applying Tri-Diagonal Method

for i=2:len

   v(i-1) = c(i-1)/w;

   w = a(i) - bb(i)*v(i-1);

   x(i) = ( b(i) - bb(i)*x(i-1) )/w;

end

for j=len-1:-1:1

  x(j) = x(j) - v(j)*x(j+1);

end
```