Tema lab04

Table of Contents

Metoda Newton

```
function [xaprox, N] = Newton(F, J, x0, eps)
    k=1;
    x(1,1) = x0(1);
    x(2,1) = x0(2);

    while true
        k = k+1;
        %z = J(x(1,k-1), x(2,k-1))\(-F(x(1,k-1), x(2,k-1)));
        %z = inv(J(x(1,k-1), x(2,k-1)))*(-F(x(1,k-1), x(2,k-1)));
        z = GaussPivTot(J(x(1,k-1), x(2,k-1)), (-F(x(1,k-1),
 x(2,k-1))));
        x(:,k) = z+x(:,k-1);
        if norm(z, 2) < eps
            break
        end
    end
    xaprox(1,1)=x(1,k);
    xaprox(2,1)=x(2,k);
    N=k;
end
```

Metoda directa

```
function [a] = MetDirecta(x, y)
  n = length(x);
  for i=1:n
      A(i,1)=1;
  end
  for i=1:n
      for j=2:n
          A(i,j) = x(i)^(j-1);
```

```
        end
    end
    a = GaussPivTot(A, y);
end
```

Metoda directa sintaxa 2

```
function [y] = MetDirecta2(X, Y, x)
    syms Q;
    n = length(X);
    for i=1:n
        A(i,1)=1;
    end
    for i=1:n
        for j=2:n
            A(i,j) = X(i)^(j-1);
        end
    end
    a = GaussPivTot(A, Y);
    Pn = 0;
    for i=1:length(a)
      Pn = Pn + a(i)*Q^(i-1);
    end
    Pn = matlabFunction(Pn, 'vars', {Q});
    vectorize(Pn);
    y = Pn(x);
end
```

Metoda Lagrange

```
function [y] = MetLagrange(X, Y, x)
    syms Q;
    n = length(X);
    Pn = 0;
    for k=1:n
        Lnk = 1;
        for i=1:n
            if i==k
                continue
            end
            Lnk = Lnk * (Q-X(i)) / (X(k)-X(i));
        end
        Pn = Pn + Lnk*Y(k);
    end
    Pn = matlabFunction(Pn, 'vars', {Q});
    vectorize(Pn);
    y = Pn(x);
end
```

Metoda Newton interpolare

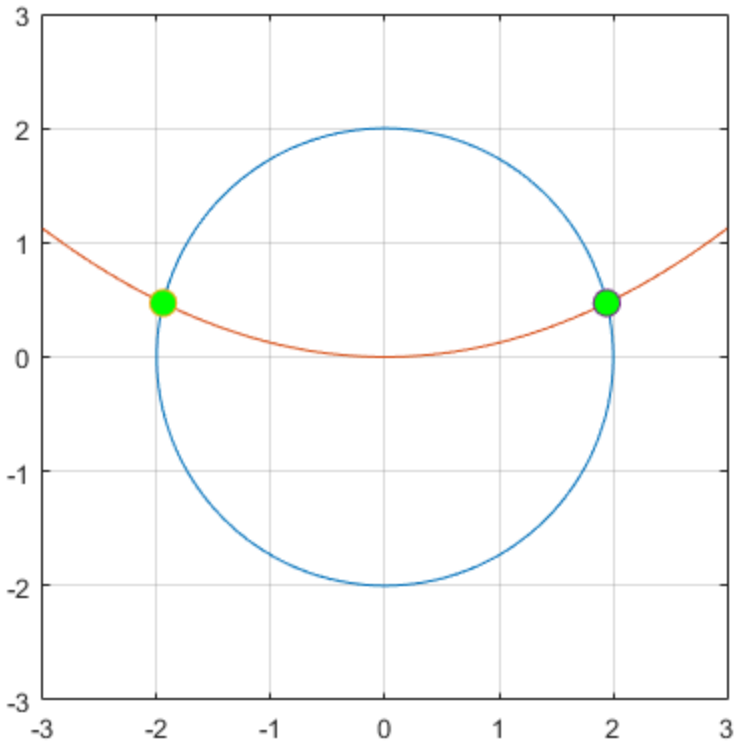
```
function [y] = MetN(X, Y, x)
  syms Q;
  n = length(X);
  Pn = 0;
  for i=1:n
      for j=1:n
          if j==1
              A(i,j)=1;
          elseif j>i
              A(i,j)=0;
          else
              prod = 1;
              for k=1:j-1
                  prod = prod * (X(i)-X(k));
              end
              A(i,j) = prod;
          end
      end
  end
  c = SubsAsc(A, Y');
  for i=1:n
      coef = c(i);
      for k=1:i-1
          coef = coef * (Q - X(k));
      end
      Pn = Pn + coef;
  end
  Pn = matlabFunction(Pn, 'vars', {Q});
  vectorize(Pn);
  y = Pn(x);
end
```

Exercitiul 1

```
C1 = @(x,y)x.^2+y.^2-4;
C2 = @(x,y)(x.^2)/8-y;

figure(1);
fimplicit(C1, [-2, 2, -2 ,2]);
grid on;
axis equal;
hold on;
fimplicit(C2, [-3,3,-3,3]);

F = @(x,y)[C1(x,y);C2(x,y)];
syms x y
f1 = C1(x,y);
f2 = C2(x,y);
```

```
J = [diff(f1,x) diff(f1,y)
     diff(f2,x) diff(f2,y)];

disp 'Jacobianul este'
J

J = matlabFunction(J, 'vars', {x, y});
eps = 10^(-5);

x0 = [-2;0];
[xaprox,N] = Newton(F, J, x0, eps);
plot(xaprox(1,1),xaprox(2,1), 'o', 'MarkerFaceColor', 'g', 'MarkerSize',
 10);

x0 = [2;0];
[xaprox,N] = Newton(F, J, x0, eps);
plot(xaprox(1,1),xaprox(2,1), 'o', 'MarkerFaceColor', 'g', 'MarkerSize',
 10);

hold off;
```

```
Jacobianul este

J =

[ 2*x, 2*y]
[ x/4,  -1]
```

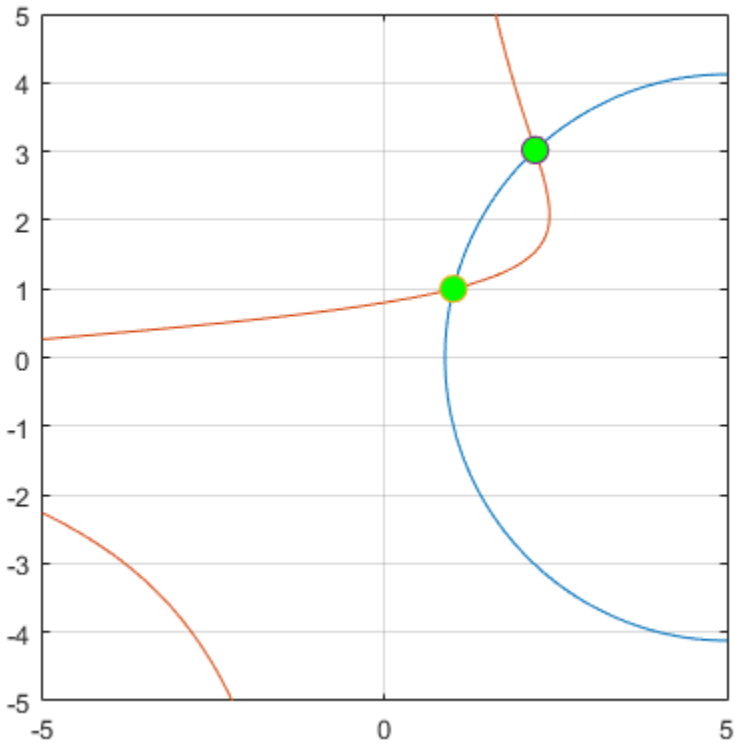


Exercitiul 2

```
C1 = @(x,y)x.^2-10.*x+y.^2+8;
C2 = @(x,y)x.*(y.^2)+x-10.*y+8;

figure(2);
fimplicit(C1, [-5, 5, -5 ,5]);
grid on;
axis equal;
hold on;
fimplicit(C2, [-5, 5, -5, 5]);

F = @(x,y)[C1(x,y);C2(x,y)];
syms x y
f1 = C1(x,y);
f2 = C2(x,y);

J = [diff(f1,x) diff(f1,y)
     diff(f2,x) diff(f2,y)];

disp 'Jacobianul este'
J

J = matlabFunction(J, 'vars', {x, y});
eps = 10^(-5);
```

```
x0 = [1;1];
[xaprox,N] = Newton(F, J, x0, eps);
plot(xaprox(1,1),xaprox(2,1), 'o', 'MarkerFaceColor', 'g', 'MarkerSize',
 10);

x0 = [3;3];
[xaprox,N] = Newton(F, J, x0, eps);
plot(xaprox(1,1),xaprox(2,1), 'o', 'MarkerFaceColor', 'g', 'MarkerSize',
 10);

hold off;
```

```
Jacobianul este

J =

[ 2*x - 10,        2*y]
[  y^2 + 1, 2*x*y - 10]
```



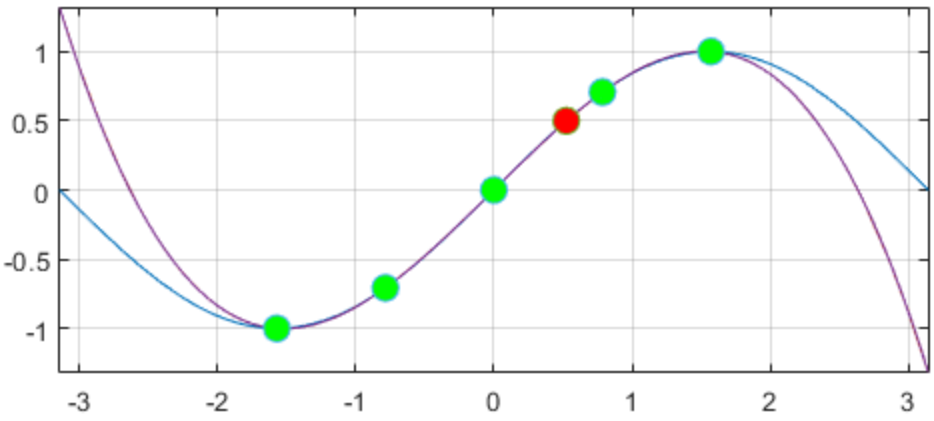
Exercitiul 3

```
f = @(x)sin(x);
x = linspace(-pi/2,pi/2,5);
y = f(x);
```

```
y = y';

figure(3);
fplot(f,[-pi,pi]);
hold on;
grid on;
axis equal;

% Prin metoda directa
disp 'Prin metoda directa'
a = MetDirecta(x,y);
a
syms X;
Pn = 0;
for i=1:length(a)
    Pn = Pn + a(i)*X^(i-1);
end
Pn
Pn = matlabFunction(Pn, 'vars', {X});

disp 'Eroarea |Pn(pi/6) - f(pi/6)|='
abs(Pn(pi/6) - f(pi/6))

fplot(Pn,[-pi,pi]);

% Prin metoda Lagrange
disp 'Prin metoda Lagrange'
disp 'Eroarea |Pn(pi/6) - f(pi/6)|='
abs(MetLagrange(x,y,[pi/6]) - f(pi/6))

plot(linspace(-pi,pi,100), MetLagrange(x,y,linspace(-pi,pi,100)));


% Prin metoda Newton
disp 'Prin metoda Newton'
disp 'Eroarea |Pn(pi/6) - f(pi/6)|='
abs(MetN(x,y,[pi/6]) - f(pi/6))

plot(linspace(-pi,pi,100), MetN(x,y,linspace(-pi,pi,100)));


plot(pi/6, f(pi/6), 'o', 'MarkerFaceColor', 'r', 'MarkerSize', 10);
plot(x, y, 'o', 'MarkerFaceColor', 'g', 'MarkerSize', 10);

hold off;
```

```
Prin metoda directa

a =

         0
    0.9882
         0
   -0.1425
```

```
         0


Pn =

(8901050889487701*X)/9007199254740992 -
 (2566984143693133*X^3)/18014398509481984

Eroarea |Pn(pi/6) - f(pi/6)|=

ans =

    0.0030

Prin metoda Lagrange
Eroarea |Pn(pi/6) - f(pi/6)|=

ans =

    0.0030

Prin metoda Newton
Eroarea |Pn(pi/6) - f(pi/6)|=

ans =

    0.0030
```

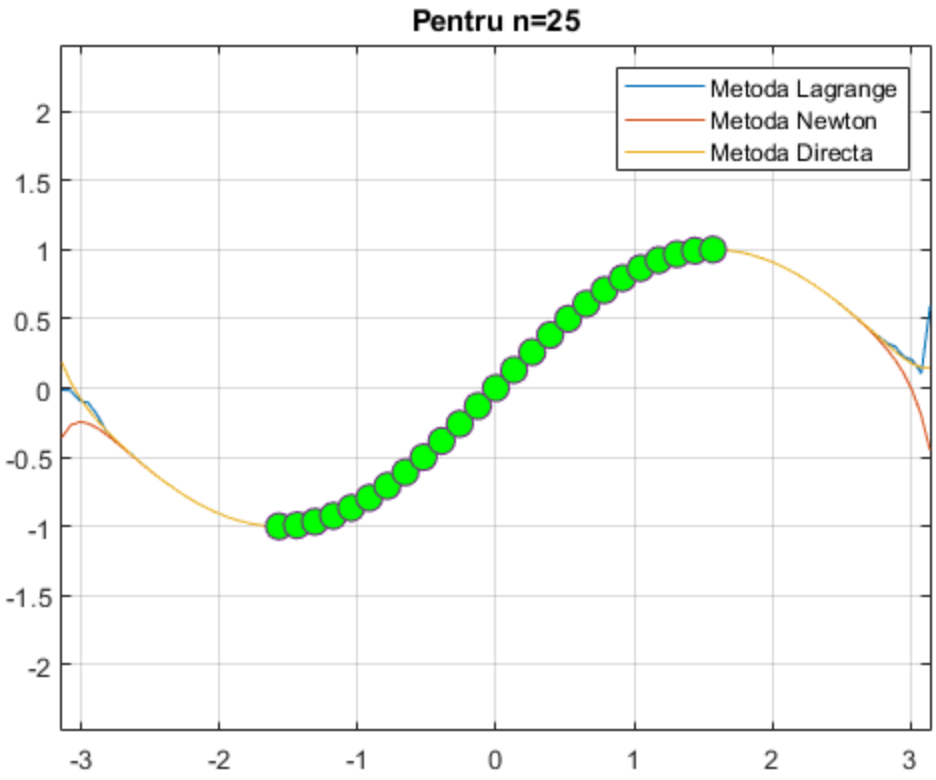


Exercitiul 4

```
f = @(x)sin(x);
n = 25;
x = linspace(-pi/2,pi/2,n);
y = f(x);
y = y';

figure(4);

disp 'Prin metoda Lagrange'
disp 'Eroarea |Pn(pi/6) - f(pi/6)|='
abs(MetLagrange(x,y,[pi/6]) - f(pi/6))

plot(linspace(-pi,pi,100), MetLagrange(x,y,linspace(-pi,pi,100)));
hold on;
grid on;
axis equal;

disp 'Prin metoda Newton'
disp 'Eroarea |Pn(pi/6) - f(pi/6)|='
abs(MetN(x,y,[pi/6]) - f(pi/6))

plot(linspace(-pi,pi,100), MetN(x,y,linspace(-pi,pi,100)));
```

```
disp 'Prin metoda directa'
disp 'Eroarea |Pn(pi/6) - f(pi/6)|='
abs(MetDirecta2(x,y,[pi/6]) - f(pi/6))

plot(linspace(-pi,pi,100), MetDirecta2(x,y,linspace(-pi,pi,100)));

plot(x, y, 'o', 'MarkerFaceColor', 'g', 'MarkerSize', 10);
title('Pentru n=25');
legend('Metoda Lagrange', 'Metoda Newton', 'Metoda Directa');
hold off;
```

```
Prin metoda Lagrange
Eroarea |Pn(pi/6) - f(pi/6)|=

ans =

   2.7756e-16

Prin metoda Newton
Eroarea |Pn(pi/6) - f(pi/6)|=

ans =

   1.6653e-16

Prin metoda directa
Eroarea |Pn(pi/6) - f(pi/6)|=

ans =

   2.7756e-16
```



Published with MATLAB® R2018a