

```

function ProgramI1
f = inline('sqrt(x)-cos(x)','x');
A=0; B=1; eps = 10^(-5);
[xaprox,N]=MetBisectie(f,A,B,eps);
fprintf('Metoda: Bisectie\n')
fprintf('Ecuatia: sqrt(x)-cos(x) = 0\n ')
fprintf('Intervalul: [%5.2f,%5.2f]\n', A,B)
fprintf('Eroarea: %5.2e\n',eps)
fprintf('Solutia aproximativa xaprox= %4.2f\n',xaprox)
fprintf('Numarul de iteratii: N = %3i\n',N)
X=linspace(A,B,100); %Discretizarea intervalului [A,B]
Y=f(X);
close all %Inchide figurile deschise de la rularile precedente
plot(X,Y,'Linewidth',3) %Constructia graficului functiei f(x)
xarrow = [0.38 0.46];
yarrow = [0.7 0.53];
annotation('textarrow',xarrow,yarrow,'String','f(x) = sqrt(x)-cos(x)')
grid on
hold on
%Construirea axelor de coordonate
xL = xlim; %xlim furnizeaza limitele axei Ox
yL = ylim; %ylim furnizeaza limitele axei Oy
line(xL, [0 0],'color','k','linewidth',3) %axa Ox
line([0 0], yL,'color','k','linewidth',3) %axa Oy
plot(xaprox,f(xaprox),'o','MarkerFaceColor','g','MarkerSize',10) %Reprezentarea solutiei pe grafic
text(0.55,0.1,'x_{aprox}','EdgeColor','y','LineWidth',2)
xlabel('x')
ylabel('y')
title('Metoda bisectiei')
function [xaprox,N]=MetBisectie(f,A,B,eps)
a(1)=A; b(1)=B; x(1)=1/2*(a(1)+b(1));
N = floor(log2((B-A)/eps));
for k=2:N+1
    if (f(x(k-1))==0)
        x(k) = x(k-1);
        break;
    elseif (f(a(k-1))*f(x(k-1))<0)
        a(k)=a(k-1); b(k)=x(k-1); x(k)=1/2*(a(k-1)+b(k-1));
    elseif (f(a(k-1))*f(x(k-1))>0)
        a(k)=x(k-1); b(k)=b(k-1); x(k)=1/2*(a(k-1)+b(k-1));
    end
end
end
xaprox=x(k);

```

```
end  
end
```

Metoda: Bisectie
Ecuatia: $\sqrt{x} - \cos(x) = 0$
Intervalul: [0.00, 1.00]
Eroarea: 1.00e-05
Solutia aproximativa xapprox= 0.64
Numarul de iteratii: N = 16



