```
%%%%%%%%%%%%%%%%%%%%%%
% Esercizio 1
%%%%%%%%%%%%%%%%%%%%%%

% punto 1
f1=@(x,alpha)(alpha.*sin(x).*cos(x)-x.^3);
f1der=@(x,alpha)( alpha.*(cos(x).^2-sin(x).^2)-3*x.^2);
a=-1; b=1;
xstar=0;
figure(1)

for alpha=[1, -1e-7, 1e7]

    fplot( @(x)(f1(x,alpha)),[a,b]);
hold on
    derivata=f1der(xstar,alpha)

end
axis([-1,1,-1,1])
hold off

% punto 2
tolla=0; tollr=0; tollf=1e-8;
kmax=100; stampe=1; grafico=0;
a=-0.2; b=0.1;

for alpha=[1, -1e-7, 1e7]
fprintf('alpha= %d\n',alpha)
   [ind,x,y,k,af,bf]=bisezione(@(x)
(f1(x,alpha)),a,b,tolla,tollr,tollf,kmax,stampe,grafico);
end

% punto 3
figure(2)
alpha=1;
a=-0.2; b=0.1;
tolla=0; tollr=0; tollf=1e-9;
kmax=100; stampe=1; grafico=1;
[ind,x,y,k,af,bf]=bisezione(@(x)(f1(x,alpha)),a,b,tolla,tollr,tollf,kmax,stampe,grafico);
hold on
%   |e_k| < = (b-a)/2^(k+1) < 1e-9   k+1 > log( (b-a)/tol)/log(2)
tol=1e-9;
k1=ceil(log( (b-a)/tol)/log(2));
semilogy(k1,tol,'*r')


%%%%%%%%%%%%%%%%%%%%
% Esercizio 2
%%%%%%%%%%%%%%%%%%%%

fprintf('\n')
fprintf('\n')
fprintf(' Esercizio 2\n')
fprintf('\n')
f=@(x)(atan(x));
df=@(x)(1./(1+x.^2));
a=-1; b=1;
xstar=0;
figure(10)
fplot( f,[a,b]);
title('funzione arctan')

% punto 1
tolla=1e-8; tollr=0; tollf=1e-8;
kmax=20; stampe=1; grafico=1;
```

```
figure(100)
x0=1.3;

fprintf('\n')
fprintf(' Newton, dato iniziale x0 = %d \n', x0)
fprintf('\n')
[ind,x,fx,k,res]=newton1d(x0,f,df,tolla,tollr,tollf,kmax,stampe,grafico);
semilogy(res)
title('Convergenza Newton')
hold on
x0=1.4;
fprintf('\n')
fprintf(' Newton, dato iniziale x0 = %d \n', x0)
fprintf('\n')
[ind,x,fx,k,res]=newton1d(x0,f,df,tolla,tollr,tollf,kmax,stampe,grafico);
semilogy(res)

a=-0.5;  b=1.35;
fprintf('\n')
fprintf(' Bisezione,  \n', x0)
fprintf('\n')
[ind,x,y,k,af,bf]=bisezione(f,a,b,tolla,tollr,tollf,kmax,stampe,grafico);
legend('Newton 1','Newton 2','bisezione')
hold off



%%%%%%%%%%%%%%%%%%%%
% Esercizio 3
%%%%%%%%%%%%%%%%%%%%

f=@(x)(x.^3-2*x-5);
tollr=1e-12; tolla=1e-12; tollf=1e-12; kmax=100; stampe=1;
%bisezione
a=2; b=3;
[ind,x,y,k,afinal,bfinal]=bisezione(f,a,b,tolla,tollr,tollf,kmax,stampe);
fprintf('Bisezione, x= %d\n',x)
hold on

xtrue=2.0945514815423280;
%newton
fder=@(x)(3*x.^2-2);
x0=3;
[ind,x,fx,k]=newton1(f,fder,x0,tolla,tollr,tollf,kmax,stampe,xtrue);
fprintf('Newton, x= %d\n',x)

%secanti
x0=2;x0m1=3;
[ind,x,fx,k]=secanti(f,x0,x0m1,tolla,tollr,tollf,kmax,stampe,xtrue);
fprintf('Secanti, x= %d\n',x)

%%%%%%%%%%%%%%%%%%%%
% Esercizio 4
%%%%%%%%%%%%%%%%%%%%

% Newton per polinomi
x0=3; coeff=[1,0,-2,-5];
[ind,x,fx,k]=newton1_pol(coeff,x0,tolla,tollr,tollf,kmax,stampe,xtrue);
fprintf('Newton pol, x= %d\n',x)

legend('bisezione','newton','secanti','Newton pol')


end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [ind,x,y,k,afinal,bfinal]=bisezione(f,a,b,tolla,tollr,tollf,kmax,stampe)



fa = f(a);
ind=-1;
% alternativa:   h=(b-a)/2;
for k=1:kmax,
    x = a + (b-a)/2;
    % alternativa:   x = a + h;  h = h/2;
    y = f(x);
    res(k)=abs(y);
    % criterio d'arresto su f
    if abs(y) <= tollf, ind=2; break,end
    % criterio di arresto su [a,b]
 %   disp([k,x,f(x), abs(b-a)/abs(a)])
    if abs(b-a) <= tolla + tollr*abs(a), ind=1;break,end
    if sign(fa)*y < 0,
        b=x;
    else
        a=x; fa=y;
    end
end
disp([k,x,f(x)])
afinal=a; bfinal=b;



if (stampe==1), semilogy(res,'o-'),end

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [ind,x,fx,k]=newton1(f,fder,x,tolla,tollr,tollf,kmax,stampe,xtrue)

fx=f(x);
err0=abs(xtrue-x);
ind=-1;
for k=1:kmax,
    dx = fx/fder(x);
    x = x- dx;
    err=abs(xtrue-x);
    p=log(err)/log(err0);
    err0=err;
    fx=f(x);
    res(k)=abs(fx);
    disp([k,x,fx, abs(dx),p])
    % criterio d'arresto su f
    if abs(fx) <= tollf, ind=2; break,end
    % criterio di arresto su |x_k - x_{k-1}|
    if abs(dx) <= tolla + tollr*abs(x), ind=1;break,end
end
if (stampe==1), semilogy(res,'x-r'),end
 function I2m=Cavalieri_Simpson(a,b,m,f)



H=(b-a)/m;
x=linspace(a,b,2*m+1);
I2m=H/6*( f(x(1))+ 2*sum(f(x(3:2:2*m))) + 4*sum(f(x(2:2:2*m))) + f(x(2*m+1)) );

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [ind,x,fx,k]=newton1_pol(coeff,x,tolla,tollr,tollf,kmax,stampe,xtrue)

[fx,fder]=horner(coeff,x);
err0=abs(xtrue-x);
ind=-1;
for k=1:kmax,
    dx = fx/fder;
    x = x- dx;
    err=abs(xtrue-x);
    p=log(err)/log(err0);
    err0=err;
    %fx=f(x);
    [fx,fder]=horner(coeff,x);
    res(k)=abs(fx);
    disp([k,x,fx, abs(dx),p])
    % criterio d'arresto su f
    if abs(fx) <= tollf, ind=2; break,end
    % criterio di arresto su |x_k - x_{k-1}|
    if abs(dx) <= tolla + tollr*abs(x), ind=1;break,end
end
if (stampe==1), semilogy(res,'d-c'),end

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 function [ind,x,fx,k]=secanti(f,x,xm1,tolla,tollr,tollf,kmax,stampe,xtrue)


fx=f(x);
fxm1=f(xm1);
err0=abs(xtrue-x);
ind=-1;
for k=1:kmax,
    fder= ( fx - fxm1)/(x-xm1);
   % dx = fx/fder(x);
    dx = fx/fder;
    xm1=x; fxm1=fx;
    x = x- dx;
    err=abs(xtrue-x);
    p=log(err)/log(err0);
    err0=err;
    fx=f(x);   % <-------- 1 valutaz f
    res(k)=abs(fx);
    disp([k,x,fx, abs(dx),p])
    % criterio d'arresto su f
    if abs(fx) <= tollf, ind=2; break,end
    % criterio di arresto su |x_k - x_{k-1}|

    if abs(dx) <= tolla + tollr*abs(x), ind=1;break,end
end
if (stampe==1), semilogy(res,'d-g'),end
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [fx,fder]=horner(coeff,x);
%
% a = coeff,   con a_j ordinati da a_n a a_0
%
n=length(coeff)-1;
b=coeff(1);
c=b;
```

```
for j=2:n
    b = b*x + coeff(j);
    c = c*x + b;
end
b = b*x + coeff(n+1);
fx = b;
fder = c;
end




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Esercizio 5
 % script es. facoltativo
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;close all
a=0; b=5;
a1=0;
N=100; m=20;
g=@(t)(exp(-t).*(t-1));
x=linspace(a,b,N);
for k=1:N, f(k)=Cavalieri_Simpson(a1,x(k),m,g);end
figure(1)
plot(x,f,'r')
title('grafico di f(x)=\int_0^x g(t) dt')

% Risolvo numericamente il problema  f'(x)=0
% f'(x) = g(x)  =>    trova x tale che g(x) = 0
% Chiamo la funzione newton modificata, invece di chiamarla "minf"
x0=.5;
tollr=1e-12; tolla=1e-12; tollf=1e-12; kmax=100; stampe=1;
figure(2)
[ind,xmin,fx,k]=newton1_DF(g,x0,tolla,tollr,tollf,kmax,stampe);
figure(1)
hold on
fmin=Cavalieri_Simpson(a1,xmin,m,g);
plot(xmin,fmin,'k*');
hold off
fprintf(' punto di minimo:  x= %d, f(x) = %d\n',xmin,fmin)
text(xmin,fmin,'punto di minimo')




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [ind,x,fx,k]=newton1_DF(f,x,tolla,tollr,tollf,kmax,stampe)

fx=f(x);
%err0=abs(xtrue-x);
ind=-1;
h=1e-4;
for k=1:kmax,
    fder = (f(x+h)-f(x-h))/(2*h);
    dx = fx/fder;
    x = x- dx;
    fx=f(x);
    res(k)=abs(fx);
    disp([k,x,fx, abs(dx)])
    % criterio d'arresto su f
    if abs(fx) <= tollf, ind=2; break,end
    % criterio di arresto su |x_k - x_{k-1}|
    if abs(dx) <= tolla + tollr*abs(x), ind=1;break,end
end
```

```
if (stampe==1),semilogy(res,'x-r'),end
end
```