RELAZIONE SECONDA ESERCITAZIONE

1) E' POSSIBILE UTILIZZARE IL METODO DI BISEZIONE PER APPROSSIMARE GLI ZERI DI

?

$$f(x) = x^2 - 2x + 1$$

ANALISI:

Il metodo di bisezione può essere applicato ad una funzione solo se in [a,b] è continua e f(a)*f(b) < 0.
Dato che la funzione di sopra è una parabola che ha la radice con molteplicità 2 in x = 1, non assume mai un valore negativo, quindi il metodo di bisezione non può essere applicato su questa funzione.

2) PER OGNI FUNZIONE APPLICARE IL METODO DI BISEZIONE PER DETERMINARE UNO ZERO DELLA FUNZIONE CON UN ERRORE ASSOLUTO MINORE O UGUALE A

10^{-6}.

(1) $f(x) = x^3 - 2x$ [-1,1]

2) $f(x) = x^3 \cos(x)$ [-1,1]

(3) $f(x) = x^2 - x + 2$ [-2,2]

(4) $f(x) = x^2 - 2 \cdot x - \log(x)$ [1,4]

(5) $f(x) = \log(2/(3-x))$ [-1,2]

DETERMINARE TUTTE LE RADICI DELLA FUNZIONE

$$f(x) = x^3 - 1.9x^2 - 1.2x + 2.5$$

NELL'INTERVALLO [-2,3], FACENDO USO DEL GRAFICO PER INDIVIDUARE I DIVERSI INTERVALLI DI STUDIO

CODICE MATLAB , RISULTATI E ANALISI:

Questo è il codice delle 5 funzioni:

```
function y = f1(x)
y = x.^3 - 2.*x;

function y = f2(x)
y = x.^3 .* cos(x);

function y = f3(x)
y = x.^2 -x + 2;

function y = f4(x)
y = x.^2 - 2.*x - log(x);

function y = f5(x)
y = log(2/(3-x));
```

Questo è il codice della bisezione:

```
function zero = bisezione(a,b,f, emax)
k = 1;
errore = 2*emax;
while(errore > emax)
    c = a + (b - a) / 2; %vedi nelle dispense meglio calcolarlo
così
    c1 = feval(f,c);
    if(c1 == 0)
        zero = c; %fine
        return
    end
   %controllo segno
   a1 = feval(f,a);

   b1 = feval(f,b);

   if((a1 * c1) < 0)
        b = c;
   else %((b1 * c1) < 0)
       a = c;
   end
   %se f(a) e f(b) sono concordi (stesso segno) errore
   if(a1 > 0 && b1 > 0)

       disp('errore: il metodo di bisezione non si può applicare
')
       return;
   end
   %aggiorno errore
   errore = abs(b - a)/(2^(k - 1));
   k = k + 1;
   zero = c;
end
```

RISULTATI:

```
bisezione(-1,1,@f1,1e-6)

ans =     0

bisezione(-1,1,@f2,1e-6)

ans =     0

bisezione(-2,2,@f3,1e-6)

errore: il metodo di bisezione non si può applicare

bisezione(1,4,@f4,1e-6)

ans =    2.3645

bisezione(-1,2,@f5,1e-6)

ans =    1.0002
```

Questo è il codice della sesta funzione:

```
function y = f6(x)
y = x.^3 - 1.9 * x.^2 - 1.2 * x + 2.5;
```

Codice del grafico:

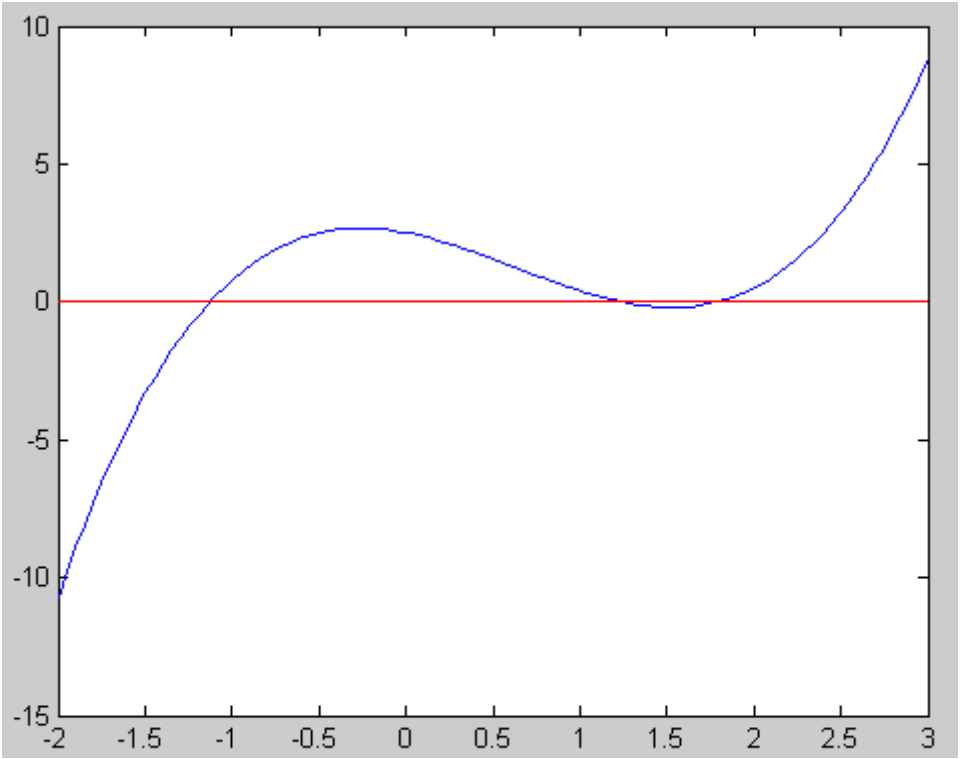
```
x=linspace(-2,3,100);
f=zeros(1,100);

%funzione da studiare
f=x.^3 - 1.9*x.^2 - 1.2 .* x + 2.5;

%crea la retta y = 0 che va da -2 a 3
x1 = zeros(1,2);
x1(1,1) = -2;
x1(1,2) = 3;
x2 = zeros(1,2);

figure
plot(x,f);
hold on
%stampa retta y = 0
plot(x1,x2,'r');
```

Grafico della funzione:



Come si vede dal grafico, la funzione ha 3 radici, quindi se si vuole usare il metodo di bisezione per trovarle tutte bisogna separare i 3 intervalli di studio, esempio:

```
bisezione(-2,0,@f6,1e-6)

ans =   -1.1279              trova la prima radice

bisezione(0,1.5,@f6,1e-6)

ans =    1.2385              trova la seconda radice

bisezione(1.5,3,@f6,1e-6)

ans =    1.7908              trova la terza radice
```

3) SI IMPLEMENTI IL METODO DI NEWTON PER LA RICERCA DEGLI ZERI DELLE FUNZIONI DELL'ESERCIZIO 2. SI CONFRONTINO LE VELOCITA' DI CONVERGENZA DEL METODO DI NEWTON, AL VARIARE DEL PUNTO INIZIALE, E DEL METODO DI BISEZIONE, PER DETERMINARE UNO ZERO DELLA FUNZIONE CON UN ERRORE ASSOLUTO MINORE O UGUALE A

10^{-6}.

CODICE MATLAB, RISULTATI E ANALISI:

Questo è il codice delle 5 derivate:

```
function y = d1(x)
y = 3.*x.^2 - 2;

function y = d2(x)
y = x.^2 .* cos(x) - sin(x).*x.^3;

function y = d3(x)
y = x - 1;

function y = d4(x)
y = 2.*x - 2 - 1./x;

function y = d5(x)
y = (3 - x)./ 2 - 0.5;
```

Questo è il codice della funzione di newton:

```
%[a,b] è l'intervallo
%x è il punto iniziale
%f è la funzione, d è la derivata
%emax è l'errore assoluto
function zero = newton(a,b,x,f,d,emax)
%parte il timer
tic
%calcolo f(x)
y = feval(f,x);
%calcolo f'(x)
m = feval(d,x);
%calcolo xk + 1
k = x - y / m;
%controllo se sono finito fuori dall'intervallo di studio
if(k < a || k > b)
    disp('errore: la x è fuori dall intervallo di studio');
    toc %fine timer
    return
end
%ciclo, finisce quando la differenza fra xk + 1 e xk in modulo è <
%dell'errore massimo
while(abs(x - k) > emax)
    x = k; % xk + 1 = xk
    %calcolo f(x)
    y = feval(f,x);
    %calcolo f'(x)
    m = feval(d,x);
    %calcolo xk + 1
    k = x - y / m;
    %controllo se sono finito fuori dall'intervallo di studio
    if(k < a || k > b)
        disp('errore: la x è fuori dall intervallo di studio');
        toc %fine timer
        return
    end
end
zero = k;
toc %fine timer
```

RISULTATI (gli zeri sono stati presi dal metodo di bisezione):

Lo zero della prima funzione è x = 0, col metodo di newton viene calcolato lo zero al variare del punto iniziale:

punto iniziale = 0.6

newton(-1,1,0.6,@f1,@d1,1e-6)

ans = -1.7867e-022

Elapsed time is 0.029812 seconds.

punto iniziale = -0.5

newton(-1,1,-0.5,@f1,@d1,1e-6)

ans = -2.3431e-019

Elapsed time is 0.000267 seconds.

punto iniziale = -0.7

newton(-1,1,-0.7,@f1,@d1,1e-6)

errore: la x è fuori dall intervallo di studio

Elapsed time is 0.000282 seconds.

punto iniziale = 0.1

newton(-1,1,0.1,@f1,@d1,1e-6)

ans =   0

Elapsed time is 0.000390 seconds.

Col metodo di bisezione il tempo impiegato per trovare uno zero è:

bisezione(-1,1,@f1,1e-6)

ans =   0

Elapsed time is 0.005519 seconds.

Dal tempo impiegato si vede che il metodo di bisezione è più lento del metodo di newton.

Lo zero della seconda funzione è x = 0, col metodo di newton viene calcolato lo zero al variare del punto iniziale:

Punto iniziale = 0.6

newton(-1,1,0.6,@f2,@d2,1e-6)

ans = 1.0340e-025

Elapsed time is 0.087081 seconds.

Punto iniziale = 0.8

newton(-1,1,0.8,@f2,@d2,1e-6)

errore: la x è fuori dall intervallo di studio

Elapsed time is 0.000239 seconds.

Punto iniziale = 0.2

newton(-1,1,0.2,@f2,@d2,1e-6)

ans = -2.2002e-019

Elapsed time is 0.000370 seconds.

Punto iniziale = -0.1

newton(-1,1,-0.1,@f2,@d2,1e-6)

ans =   0

Elapsed time is 0.000798 seconds.

Col metodo di bisezione il tempo impiegato per trovare uno zero è:

bisezione(-1,1,@f2,1e-6)

ans =   0

Elapsed time is 0.000174 seconds.

In questo caso il metodo di bisezione è stato più veloce del metodo di newton.

La terza funzione è una parabola sempre positiva che non ha zeri, quindi col metodo di newton viene dato errore qualunque sia il punto iniziale, esempio:

Punto iniziale = 1

```
newton(-2,2,1,@f3,@d3,1e-6)

errore: la x è fuori dall intervallo di studio

Elapsed time is 0.019397 seconds.
```

Lo zero della quarta funzione è x = 2.3645, col metodo di newton viene calcolato lo zero al variare del punto iniziale:

Punto iniziale = 2

```
newton(1,4,2,@f4,@d4,1e-6)

ans = 2.3639

Elapsed time is 0.007833 seconds.
```

Punto iniziale = 3

```
newton(1,4,3,@f4,@d4,1e-6)

ans =    2.3639

Elapsed time is 0.000434 seconds.
```

Punto iniziale = 3.9

```
newton(1,4,3.9,@f4,@d4,1e-6)

ans =    2.3639

Elapsed time is 0.000462 seconds.
```

Col metodo di bisezione il tempo impiegato per trovare uno zero è:

```
bisezione(1,4,@f4,1e-6)

ans =    2.3645

Elapsed time is 0.003429 seconds.
```

In questo caso il metodo di bisezione è stato più lento del metodo di newton.

Lo zero della quinta funzione è x = 1.0002, col metodo di newton viene calcolato lo zero al variare del punto iniziale:

Punto iniziale = 1

```
newton(-1,2,1,@f5,@d5,1e-6)

ans =     1

Elapsed time is 0.006472 seconds.
```

Punto iniziale = 1.5

```
ans =    1.0000

Elapsed time is 0.000784 seconds.
```

Punto iniziale = -0.5

```
newton(-1,2,-0.5,@f5,@d5,1e-6)

ans =    1.0000

Elapsed time is 0.000520 seconds.
```

Punto iniziale = 2

```
newton(-1,2,2,@f5,@d5,1e-6)

errore: la x è fuori dall intervallo di studio

Elapsed time is 0.000245 seconds.
```

In questa funzione, il metodo di newton è stato più preciso del metodo di bisezione, in quanto trova lo zero teorico (ans = 1).

Col metodo di bisezione il tempo impiegato per trovare uno zero è:

```
bisezione(-1,2,@f5,1e-6)

ans =    1.0002

Elapsed time is 0.001430 seconds.
```

In questo caso il metodo di bisezione è stato più lento dwel metodo di newton.

Come ci si aspettava dalla teoria, il metodo di newton è più veloce del metodo di bisezione, infatti il metodo di bisezione trova lo zero in tempo lineare, mentre il metodo di newton trova lo zero in tempo quadratico.

4) SI IMPLEMENTI UN METODO IBRIDO CHE INIZIA FACENDO QUALCHE PASSO DEL METODO DI BISEZIONE IN MODO DA RIDURRE L'INTERVALLO E POI APPLICA IL METODO DI NEWTON PER CALCOLARE GLI ZERI DELLE FUNZIONI DELL'ESERCIZIO 2. SI CONFRONTINO I RISULTATI OTTENUTI CON UELLI FORNITI DAL METODO DI BISEZIONE E DI NEWTON (ERRORE ASSOLUTO MINORE O UGUALE A 10^{-6})

CODICE MATLAB, ANALISI E RISULTATI:

Il codice di Matlab del metodo ibrido è il seguente:

```
function zero = ibrido(a,b,f,d,emax)
%il metodo ibrido fa 3 passi del metodo di bisezione, poi trova il
punto
%iniziale da dare al metodo di newton per calcolare lo zero
k = 1;
errore = 2*emax;
cont = 0; %quando arriva a 3 si ferma la bisezione e parte newton
while(errore > emax && cont < 3)
    c = a + (b - a) / 2; %vedi nelle dispense meglio calcolarlo
così
    c1 = feval(f,c);
    if(c1 == 0)
        zero = c; %fine
        return
    end
   %controllo segno
   a1 = feval(f,a);

   b1 = feval(f,b);

   if((a1 * c1) < 0)
        b = c;
   else %((b1 * c1) < 0)
       a = c;
   end
   %se f(a) e f(b) sono concordi (stesso segno) errore
   if(a1 > 0 && b1 > 0)
       disp('errore: il metodo di bisezione non si può applicare
')
       return;
   end
   %aggiorno errore
   errore = abs(b - a)/(2^(k - 1));
   k = k + 1;
   zero = c;
   cont = cont + 1;
end
```

```
%parte il metodo di newton
%la x iniziale è lo zero della bisezione
zero = newton(a,b,zero,f,d,emax)
```

RISULTATI e ANALISI:

```
ibrido(-1,1,@f1,@d1,1e-6)

ans =     0

Risultato della bisezione =       0

Risultato del metodo di newton =  -1.7867e-022
```

Il metodo ibrido ha trovato la soluzione esatta(0).

```
ibrido(-1,1,@f2,@d2,1e-6)

ans =     0

Risultato della bisezione =       0

Risultato del metodo di newton =  1.0340e-025
```

Il metodo ibrido ha trovato la soluzione esatta (0).

```
ibrido(-2,2,@f3,@d3,1e-6)

errore: il metodo di bisezione non si può applicare
```

L'errore viene trovato quando il metodo ibrido prova a fare il metodo di bisezione.

```
ibrido(1,4,@f4,@d4,1e-6)

ans =    2.3639

Risultato della bisezione =       2.3645

Risultato del metodo di newton =  2.3639
```

Il metodo ibrido ha trovato la stessa soluzione del metodo di newton.

```
ibrido(-1,2,@f5,@d5,1e-6)

ans =    1.0000

Risultato della bisezione =       1.0002

Risultato del metodo di newton =  1.0000

Il metodo ibrido ha trovato la soluzione esatta (1).
```