ESERCITAZIONE MATLAB 3: Equazioni non lineari

1. Si scriva un M-file di tipo FUNCTION che implementa il metodo di Newton per il calcolo di una radice di una equazione non lineare. L'intestazione della FUNCTION deve essere la seguente

```
 function [x,it] = newton(fun,dfun,x0,tolx,itmax)

% [x,it] = newton(fun,dfun,x0,tolx,itmax)
%
% Implementa il metodo di Newton per il
% calcolo di una radice della equazione
% non lineare
%
% fun(x) = 0
%
% Input: fun  --> funzione di cui si vuole calcolare la radice
%        dfun --> derivata prima di fun
%        x0   --> punto di innesco
%       tolx  --> tolleranza per il criterio di arresto
%      itmax  --> numero massimo di iterazioni consentite
%
% Output: x   --> approssimazione della radice calcolata
%         it  --> numero di iterazioni applicate per il
%                 calcolo di x
```

   Si salvi l'M-file con il nome `newton.m`.

2. Al fine di verificare se il metodo di Newton è stato implementato correttamente lo si utilizzi per calcolare la soluzione di

$$f(x) \equiv x - 1 = 0.$$

   Per fare questo, si scriva il seguente M-file salvandolo con il nome `fun1.m`

```
function f = fun1(x)
    f = x - 1;
```

   ed il seguente M-file salvandolo con il nome `dfun1.m`

```
function df = dfun1(x)
    df = 1;
```

   Da prompt dei comandi si esegua quindi

```
>> [x,it] = newton('fun1','dfun1', x0, 1e-14, 100)
```

   sostituendo `x0` con vari valori. Se il metodo di Newton è stato implementato correttamente allora si deve ottenere `x = 1` e `it = 1` (o, al massimo, `it = 2`) per ogni `x0`.

3. Si utilizzi il metodo di Newton implementato per calcolare la soluzione della equazione

$$f(x) \equiv \mathtt{e}^{x-1} - 1 = 0.$$

Per fare questo, si scriva il seguente M-file salvandolo con il nome `fun2.m`

```
function f = fun2(x)
    f = exp(x-1)-1;
```

ed il seguente M-file salvandolo con il nome `dfun2.m`

```
function df = dfun2(x)
    df = exp(x-1);
```

In modo analogo all'esercizio precedente, da prompt dei comandi si lanci un certo numero di volte il seguente comando

```
>> [x,it] = newton('fun2','dfun2', x0, tolx, 100)
```

specificando in input vari valori di `x0` e `tolx`.

4. Operando uno studio di funzione, si può dimostrare che la seguente equazione

$$f(x) \equiv x - \cos(x) = 0 \tag{1}$$

ammette una sola soluzione $x^* \in [0, 1]$. Si applichi il metodo di Newton per determinarla e si verifichi che il metodo converge soltanto se il punto di innesco è scelto "sufficientemente" vicino a x^*.

5. Si utilizzi il metodo di Newton per calcolare la soluzione della seguente equazione

$$f(x) \equiv (x-1)^2 \mathtt{e}^x = 0 \tag{2}$$

utilizzando `x0` $= 2$ come approssimazione iniziale della radice $x^* = 1$ che ha molteplicità $m = 2$. Si scriva poi un M-file di tipo FUNCTION, da salvare con il nome `newtonmod.m`, che implementi la versione modificata del metodo di Newton per la approssimazione di radici multiple con molteplicità nota a priori. L'intestazione della nuova function deve essere la seguente

```
function [x,it] = newtonmod(fun,dfun,x0,tolx,itmax,m)
```

Si utilizzi quindi `newtmod` per risolvere (2) e si verifichi che il metodo di Newton modificato converge molto più velocemente. Si ripeta lo stesso esperimento per l'equazione

$$f(x) \equiv (x-1)^3 \mathtt{e}^x = 0. \tag{3}$$

SOLUZIONI:

1.
```
function [x,it] = newton(fun,dfun,x0,tolx,itmax)

% [x,it] = newton(fun,dfun,x0,tolx,itmax)
%
% Implementa il metodo di Newton per il calcolo di una radice
% della equazione non lineare
%
% fun(x) = 0
%
% Input: fun  --> funzione di cui si vuole la radice
%        dfun --> derivata prima di fun
%        x0   --> punto di innesco
%       tolx  --> tolleranza per il criterio di arresto
%      itmax  --> numero massimo di iterazioni consentite
%
% Output: x   --> approssimazione della radice calcolata
%         it  --> numero di iterazioni applicate per il
%                 calcolo di x


f   = feval(fun,x0);
if (f==0),
    x=x0;
    it=0;
    return,
end

df  = feval(dfun,x0);
if (df==0),
   error('Metodo di Newton non applicabile'),
end

x = x0 - f/df;

it = 1;

while ((abs(x-x0)>tolx)&(it<=itmax))

    x0 = x;

    f   = feval(fun,x0);
    if (f==0),
        return
    end

    df  = feval(dfun,x0);
    if (df==0),
       error('Metodo di Newton non applicabile'),
    end
```

```
    x = x0 - f/df;

    it = it + 1;
end

if (abs(x-x0)>tolx),
   error('Il metodo di Newton non converge'),
end
```

2.
```
>> [x,it] = newton('fun1','dfun1',2,1e-14,100)
x =

     1

it =

     1

>> [x,it] = newton('fun1','dfun1',10,1e-14,100)

x =

     1

it =

     1

>> [x,it] = newton('fun1','dfun1',-5,1e-14,100)

x =

     1

it =

     1
```

3.
```
>> format long
>> [x,it] = newton('fun2','dfun2',5,1e-3,100)

x =

   1.000000000102263

it =

     8

>> [x,it] = newton('fun2','dfun2',5,1e-10,100)
```

```
x =

     1

it =

     9

>> [x,it] = newton('fun2','dfun2',-1,1e-3,100)

x =

   1.000000028921676

it =

     9

>> [x,it] = newton('fun2','dfun2',-1,1e-10,100)

x =

     1

it =

    11
```

4. Siano `fun3.m` e `dfun3.m` le function per la valutazione della funzione (1) e della sua derivata prima. Utilizzando come approssimazioni iniziali `x0=1` e `x0=10` si ottengono i seguenti risultati

```
 >> [x,it] = newton('fun3','dfun3',1,1e-10,100)

x =

   0.739085133215161


it =

     4

>> [x,it] = newton('fun3','dfun3',10,1e-10,100)
??? Error using ==> newton at 57
Il metodo di Newton non converge
```

5. Il codice per la versione modificata del metodo di Newton è il seguente

```
function [x,it] = newtonmod(fun,dfun,x0,tolx,itmax,m)

% [x,it] = newtonmod(fun,dfun,x0,tolx,itmax,m)
%
% Implementa il metodo di Newton modificato per il calcolo di una radice
% multipla della equazione non lineare
%
% fun(x) = 0
%
% Input: fun  --> funzione di cui si vuole la radice
%        dfun --> derivata prima di fun
%        x0   --> punto di innesco
%       tolx  --> tolleranza per il criterio di arresto
%      itmax  --> numero massimo di iterazioni consentite
%          m  --> molteplicita' della radice
%
% Output: x   --> approssimazione della radice calcolata
%         it  --> numero di iterazioni applicate per il
%                 calcolo di x


f   = feval(fun,x0);
if (f==0),
    x=x0;
    it=0;
    return,
end

df  = feval(dfun,x0);
if (df==0),
   error('Metodo di Newton non applicabile'),
end

x = x0 - m*f/df;

it = 1;

while ((abs(x-x0)>tolx)&(it<=itmax))

    x0 = x;

    f   = feval(fun,x0);
    if (f==0),
        return
    end

    df  = feval(dfun,x0);
    if (df==0),
       error('Metodo di Newton non applicabile'),
    end
```

```
    x = x0 - m*f/df;

    it = it + 1;
end

if (abs(x-x0)>tolx),
   error('Il metodo di Newton non converge'),
end
```

Siano quindi `fun4.m` e `fun5.m` gli M-file per la valutazione della funzione in (2) e della sua derivata prima. I risultati che si ottengono applicando Newton e Newton modificato sono i seguenti:

```
>> [x,it]=newton('fun4','dfun4',2,1e-6,100)

x =

    1.0000


it =

    22

>> [x,it]=newtonmod('fun4','dfun4',2,1e-6,100,2)

x =

    1.0000


it =

     5
```

Il vantaggio dell'utilizzo di Newton modificato rispetto a Newton classico è ancora più evidente nel caso della equazione (3) (chiaramente specificando tre come ultimo parametro di input per `newtonmod`).