Laboratorio Calcolo Numerico

Esercizio 1

Si vogliono calcolare le soluzioni approssimate (una è anche determinabile in modo analitico esatto!) dell'equazione non lineare

$$f(x) = x^2 - 1 + e^{-x} = 0.$$

1. Si determinino graficamente, utilizzando le capacità grafiche di Matlab, degli intervalli sufficientemente piccoli (**di ampiezza non maggiore di 0.2**) che contengono **una e una sola soluzione**, dell'equazione $f(x) = 0$.

Esercizio 2

Si crei una function di nome `bisezione.m` che, prendendo spunto dall'algoritmo relativo al metodo di bisezione (pag. 70 del libro di Calcolo Numerico) permetta la determinazione di una radice reale contenuta nell'intervallo $[a, b]$.

1. Tale function deve avere come parametri **in ingresso** la funzione (definita come *inline function*[1]), gli estremi dell'intervallo, la tolleranza e il numero massimo di iterazioni.

2. Come parametri **in uscita** ci dovranno essere il vettore delle iterate `xv` che collezioni tutti i punti medi degli intervalli generati, il vettore delle semilunghezze dei successivi intervalli `slv`, il vettore dei residui corrispondenti `fxv` ed il numero `n` corrispondente all'ultimo valore x_n della successione (numero di iterazioni).

La function avrà quindi la seguente intestazione:

```
function [xv, slv, fxv, n] = bisezione (f, a, b, toll, nmax)
% BISEZIONE Metodo di Bisezione
%
% Uso: [xv, slv, fxv, n] = bisezione (f, a, b, toll, nmax)
%
% Dati di ingresso:
%   f:      funzione (inline function)
%   a:      estremo sinistro
%   b:      estremo destro
%   toll:   tolleranza richiesta per il test di arresto (semiampiezza dell'intervallo/residuo)
%   nmax:   massimo indice dell'iterata permesso
%
% Dati di uscita:
%   xv:     vettore contenente le iterate
%   slv:    vettore contenente le semilunghezze degli intervalli
%   fxv:    vettore contenente i corrispondenti residui
%   n:      indice dell'iterata finale calcolata
```

Esercizio 3

Si scriva uno script per utilizzare la funzione creata. Tale script deve:

1. Richiedere a video (comando `input`) i dati, ovvero la funzione, gli estremi dell'intervallo iniziale, la tolleranza e il numero massimo di iterazioni richieste.

2. Ottenuti gli argomenti di uscita della funzione, visualizzare (comando `disp`) l'ultima soluzione approssimata determinata dalla function, il relativo residuo, e l'indice dell'ultima iterata calcolata.

3. Realizzare un grafico che rappresenti (scala logaritmica sull'asse y) il valore assoluto del vettore che contiene i residui (`fxv`).

[1] La funzione può essere definita anche come *anonymous function* ma in tale caso non sarà possibile usare il comando `formula` nell'esercizio successivo.

Considerata la soluzione **positiva** della funzione indicata nell'Esercizio 1 ed utilizzando l'intervallo precedentemente determinato, si applichi il metodo di bisezione eseguendo lo script. Si utilizzi come test di arresto il valore $\varepsilon \longrightarrow \texttt{toll} = 1e-8$ e $\eta_{\max} = 100$.
Si provi a ripetere l'esecuzione inserendo $\eta_{\max} = 20$ e la stessa tolleranza e si analizzino i risultati ottenuti paragonandoli ai precedenti. In particolare, **quali considerazioni possono essere effettuate relativamente al valore x_n ottenuto**?

Esercizio 4
Per una visualizzazione dei risultati più accurata, si crei la seguente funzione, e se ne chieda l'esecuzione alla fine dello script che risolve ogni equazione nonlineare con il metodo di bisezione:

```
function [] = risultati_bis(a,b,f,xv,slv,fxv)
%RISULTATI_BIS function per visualizzare risultati provenienti dal metodo
% di bisezione per la ricerca degli zeri di equazioni non lineari
% Uso:
% risultati_bis(a,b,f,xv, slv,fxv)
%
% Dati di ingresso:
% a: estremo sinistro dell'intervallo
% b: estremo destro dell'intervallo
% f: funzione di cui cercare lo zero (inline function)
% xv: vettore contenente le iterate
% slv: vettore contenente le semilunghezze
% fxv: vettore contenente i corrispondenti residui
%
xv=xv(:);
slv=slv(:);
fxv=fxv(:);
n=length(xv);
fprintf('\nf: %s \tIntervallo: a=%g b=%g Bisezione \n\n', ...
formula(f),a,b);
fprintf('n \t x_n \t\t\t\t f(x_n) \t\t b_n-a_n\n');
fprintf('%d\t %20.15f \t %10.2e \t %10.4e \n', ...
[(1:n);xv';fxv';slv']);
```

Esercizio 5
Si modifichi oppportunamente lo script in modo che

- stampi a video quante iterazioni sono necessarie al metodo per ottenere una radice approssimata con un'accuratezza di $\tau = \varepsilon/2$. Suggerimento: si veda la formula a pagina 72 del libro di Calcolo Numerico e la funzione Matlab `ceil` (`ceil`(nu) $\longrightarrow \lceil \nu \rceil$).
- stampi a video una scritta di avviso qualora sia stato raggiunto l'indice massimo per le iterazioni, senza aver raggiunto la tolleranza desiderata sull'ampiezza dell'intervallo.

Esercizio 6 (Da svolgere e consegnare prima della successiva lezione di laboratorio)

Si scriva uno script Matlab chiamato `scriptbis.m` che invocando la function `bisezione.m` che implementa il suddetto metodo iterativo risolva l'equazione:

$$5x = \exp(-x).$$

Si usi una tolleranza $\epsilon = 10^{-8}$ e $nmax = 100$.
Lo script, prima dell'istruzione che invoca la function `bisezione` perchè venga eseguita, deve contenere le istruzioni necessarie per rappresentare graficamente la funzione e l'asse x nella stessa finestra grafica. Dopo aver individuato la posizione della radice dal grafico, si scelga un opportuno intervallo iniziale per il metodo di bisezione.

Si confronti il numero di iterazioni effettivamente eseguite dal metodo con il numero stimato mediante

$$n = \left\lceil \log_2 \left(\frac{b-a}{\epsilon} \right) - 1 \right\rceil.$$

Si aggiungano allo script le istruzioni necessarie per creare, in una finestra grafica diversa da quella che contiene il grafico della funzione (comando `figure`), un grafico in scala logaritmica sull'asse y con il profilo di convergenza del metodo. Si salvi la figura ottenuta in formato pdf con il nome `convergenza.pdf`.
Lo script deve scrivere i risultati forniti dal programa `bisezione.m` come righe di una tabella. Per facilitare la verifica del lavoro svolto si allegano nella pagina seguente i risultati che si devono ottenere se si sceglie come intervallo iniziale $[-1, 1]$.

```
f: 5*x - exp(-x)  Intervallo: a=-1 b=1 Bisezione

n     x_n        f(x_n)    b_n-a_n
1      0.000000000000000     1.00e+00   1.0000e+00
2      0.500000000000000     1.89e+00   5.0000e-01
3      0.250000000000000     4.71e-01   2.5000e-01
4      0.125000000000000     2.57e-01   1.2500e-01
5      0.187500000000000     1.08e-01   6.2500e-02
6      0.156250000000000     7.41e-02   3.1250e-02
7      0.171875000000000     1.73e-02   1.5625e-02
8      0.164062500000000     2.84e-02   7.8125e-03
9      0.167968750000000     5.54e-03   3.9062e-03
10      0.169921875000000     5.88e-03   1.9531e-03
11      0.168945312500000     1.71e-04   9.7656e-04
12      0.168457031250000     2.68e-03   4.8828e-04
13      0.168701171875000     1.26e-03   2.4414e-04
14      0.168823242187500     5.42e-04   1.2207e-04
15      0.168884277343750     1.85e-04   6.1035e-05
16      0.168914794921875     6.89e-06   3.0518e-05
17      0.168930053710938     8.23e-05   1.5259e-05
18      0.168922424316406     3.77e-05   7.6294e-06
19      0.168918609619141     1.54e-05   3.8147e-06
20      0.168916702270508     4.26e-06   1.9073e-06
21      0.168915748596191     1.31e-06   9.5367e-07
22      0.168916225433350     1.47e-06   4.7684e-07
23      0.168915987014771     7.90e-08   2.3842e-07
24      0.168915867805481     6.18e-07   1.1921e-07
25      0.168915927410126     2.69e-07   5.9605e-08
26      0.168915957212448     9.52e-08   2.9802e-08
27      0.168915972113609     8.10e-09   1.4901e-08
```

Esercizio 7
Si approssimino con il metodo di bisezione le radici delle seguenti funzioni:

$$\begin{aligned}
f(x) &= 1 - x - e^{-2x} \\
f(x) &= 2xe^{x} - 1 \\
f(x) &= e^{x^2} - 1 \\
f(x) &= x^3 - x - 2 \\
f(x) &= x^2 - 2 - \sin x \\
f(x) &= 3x - \cos x \\
f(x) &= \sin x - x^2/2 \\
f(x) &= 2 + \log(1 + x^2) - x \\
f(x) &= \log(3x^2 - x + 1) - 4 \\
f(x) &= e^{x} - 4x^2 \\
f(x) &= \sqrt{x + 2} - e^{-x} \\
f(x) &= (x - 2)(x - 1)\log(x)
\end{aligned}$$