Laboratorio Calcolo Numerico

Esercizio 1

Si studino graficamente 5 delle seguenti funzioni identificando per ognuna di esse tutte le soluzioni (od alcune di esse) dell'equazione $f(x) = 0$ e gli intervalli che ne contengono una ed una sola.
Per far pratica in preparazione del test finale, si salvi qualche figura in formato `.pdf`.

$$
\begin{aligned}
f(x) &= 1 - x - e^{-2x} \\
f(x) &= 2xe^{x} - 1 \\
f(x) &= e^{x^2} - 1 \\
f(x) &= x + \log x \\
f(x) &= x^3 - x - 2 \\
f(x) &= x^2 - 2 - \sin x \\
f(x) &= 3x - \cos x \\
f(x) &= \sin x - x^2/2 \\
f(x) &= e^{x} - 5 + x^2 \\
f(x) &= x^2 - 2x - e^{-x+1} \\
f(x) &= x^3 - 4x^2 + \log x \\
f(x) &= 2 + \log(1 + x^2) - x \\
f(x) &= x^2 + 3 - \frac{\sin x}{x^2} \\
f(x) &= \log(3x^2 - x + 1) - 4 \\
f(x) &= 6 - (1 + x)\frac{(1 + x)^5 - 1}{x} \\
f(x) &= e^{x} - 4x^2 \\
f(x) &= \sqrt{x + 2} - e^{-x} \\
f(x) &= x^2 + 3 - \tan(x) \\
f(x) &= (x - 2)(x - 1)\log(x) \\
f(x) &= 3 + \log(2 + x^2) - x \\
f(x) &= e^{x} - 2x^2 \\
f(x) &= \sqrt{x + 1} - e^{-x}
\end{aligned}
$$

Esercizio 2

- 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 del Matlab, dell'equazione $f(x) = 0$.
2. Si determinino degli intervalli sufficientemente piccoli (di **ampiezza non maggiore di 0.2**) che contengono **una ed una sola soluzione**.

- Si crei una function di nome `bisezfun.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 (memorizzata come variabile *inline function*), gli estremi dell'intervallo, la tolleranza ed 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 dei residui corrispondenti `fxv` ed il numero `n` corrispondente all'ultimo valore x_n della successione.

  La function avrà quindi la seguente intestazione:

```
function [xv, fxv, n] = bisezfun (f, a, b, toll, nmax)
%BISEZFUN Metodo di Bisezione
%
% Uso:
% [xv, fxv, n] = bisezfun(f, a, b, toll, nmax)
%
% Dati di ingresso:
%   f:      funzione (inline function)
%   a:      estremo sinistro
%   b:      estremo destro
%   toll:   tolleranza richiesta per l'ampiezza
%           dell'intervallo
%   nmax:   massimo indice dell'iterata permesso
%
% Dati di uscita:
%   xv:     vettore contenente le iterate
%   fxv:    vettore contenente i corrispondenti residui
%   n:      indice dell'iterata finale calcolata
```

- Si scriva uno script per utilizzare la funzione creata che

  1. Richieda 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 dalla funzione (risultati), visualizzi (comando `disp`) l'ultima soluzione approssimata determinata dalla function, il relativo residuo, e l'indice dell'ultima iterata calcolata.
  3. Alla fine preveda anche un grafico che rappresenti (scala logaritmica sull'asse y) il valore assoluto del vettore che contiene i residuo (`fxv`).

- 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 $\varepsilon \to \texttt{toll} = \texttt{1e} - 8$ e $n_{\max} = 100$.
- Si provi a ripetere l'esecuzione inserendo $n_{\max} = 20$ 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?**

Per una visualizzazione dei dati più accurata, si crei la seguente funzione, e se ne chieda l'esecuzione alla fine dello script.

```
function [] = risultati_bis(a,b,f,xv,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, fxv, metodo)
%
% 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
%   fxv:        vettore contenente i corrispondenti residui
%

xv=xv(:);
fxv=fxv(:);
n=length(xv)-1;
ampv=[(abs(b-a))./(2.^[0:n])];
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', ...
    [0:n;xv';fxv';ampv]);
```

Esercizio 3

Si modifichi opportunamente 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)` $\rightarrow \lceil \nu \rceil$).
- stampi a video una scritta di avviso qualora sia stato raggiunto l'indice massimo per le iterazioni, senza aver raggiunto la tolleranza sull'ampiezza dell'intervallo desiderata.