Laboratorio Calcolo Numerico

Esercizio 1
Si scriva una function Matlab di nome `newtonfun.m` che implementi il metodo di Newton.

- La function `newtonfun.m` deve avere come dati di ingresso la funzione $f \to$ `f` e la derivata $f' \to$ `df` che deve essere preventivamente calcolata a mano. Devono anche essere forniti in ingresso il valore iniziale $x_0 \to$ `x0`, la tolleranza $\varepsilon \to$ `toll` ed il numero massimo di iterazioni consentite $n_{\max} \to$ `nmax`.
- La stessa function deve restituire il vettore `xv` che contiene le iterate, approssimazioni successive della soluzione (incluso il valore iniziale x_0), il vettore `scarti` contenente le differenze $|x_n - x_{n-1}|$, il numero `n` di iterazioni effettuate e una variabile di controllo `flag` che indichi un'eventuale divisione per zero;

La function dovrà avere la seguente intestazione:

```
%-------------------------------------------------------------
% function [xv,scarti,n,flag]=newtonfun(f,df,x0,toll,nmax)
% NEWTONFUN Metodo di Newton
% Uso: [xv,scarti,n,flag]=newtonfun(f,df,x0,toll,nmax)
%--------------------------------------------------------------
% Dati in ingresso:
%       f:      funzione della quale si cerca una radice in [a,b]
%       df:     derivata prima di f
%       x0:     approssimazione iniziale
%       toll:   tolleranza
%       nmax:   numero massimo di iterazioni
% Dati in uscita:
%       xv:     vettore contenente le iterate
%       scarti: vettore degli scarti successivi
%       n:      numero di iterazioni effettuate
%       flag:   se flag = 1 la derivata prima si e' annullata
%--------------------------------------------------------------
```

Esercizio 2
Si consideri l'equazione non lineare $f(x) = x^2 - 1 + e^{-x} = 0$. Il grafico della funzione f e l'individuazione degli intervalli contenenti le tre radici della stessa era giá stato fatto per esercizio nel laboratorio precedente. L'equazione ha una soluzione esatta $\alpha_1 = 0$. Si vuole ora determinare la soluzione α_2 appartenente all'intervallo $\mathcal{I} = [0.6, 0.8]$ utilizzando il metodo di Newton.

La funzione $f(x)$ soddisfa le ipotesi del Teorema di convergenza locale del Metodo di Newton per la radice α_2 (Teorema 3.12, libro) nell'intervallo $\mathcal{I}$ e la sua derivata non si annulla mai nell'intervallo $\mathcal{I}$.

- Si scriva uno script `newtonscript.m` che definisca la funzione, la sua derivata, il valore iniziale, la tolleranza, il numero $n_{\max}$.
- Lo script `newtonscript.m` deve chiedere all'utente la stringa `exprf` contenente la funzione, la stringa `exprdf` contenente la derivata prima della funzione, la tolleranza, il valore iniziale x_0 e il numero n_{max}, tramite il comando `input`;
- Ottenuti gli argomenti di uscita dalla funzione (risultati), lo script deve visualizzare a video (comando `fprintf`) l'ultima soluzione approssimata determinata dalla function, l'ultimo scarto, l'ultimo residuo e l'indice dell'ultima iterata calcolata;
- Lo script deve dare un opportuno messaggio d'errore se vengono effettuate $n = n_{\max}$ iterazioni o se `flag` $\neq 0$.
- Lo script deve generare un grafico che rappresenti (scala logaritmica sull'asse y) il valore assoluto del vettore che contiene gli scarti (`scarti`).

- Infine lo script deve scrivere a video i risultati come colonne di una tabella (indice di iterazione, valore approssimato e scarto corrispondente). A questo fine si scriva la function `risultati_newton` e se ne chieda l'esecuzione all'interno dello script. La function `risultati_newton` potrebbe essere la seguente:

```
function [] = risultati_newton(a,b,f,xv,scarti,x0)
%RISULTATI_NEWTON function per visualizzare risultati provenienti dal metodo
% di Newton per la ricerca degli zeri di equazioni non lineari
% Uso:
% risultati_newton(a,b,f,xv,scarti,x0)
%
% Dati di ingresso:
% a: estremo sinistro dell'intervallo
% b: estremo destro dell'intervallo che contiene lo zero cercato
% f: funzione di cui cercare lo zero (inline function)
% xv: vettore contenente le iterate (incluso x0)
% scarti: vettore contenente gli scarti
% x0: approssimazione iniziale
%
xv=xv(:);
scarti=scarti(:);
n=length(scarti);
fprintf('\nf: %s \tIntervallo: a=%g b=%g Newton x0=%g\n\n', ...
formula(f),a,b,x0);
fprintf('n \t\t x_n \t\t x_n -x_n-1 \n\n');
fprintf('%d\t %20.15f \t %10.2e  \n', ...
[(1:n);xv(2:end)';scarti']);
```

$n = n_{\max}$ iterazioni o se `flag` $\neq 0$.

Esercizio 3

- Si utilizzi lo script tre volte, definendo come valori iniziali i due estremi dell'intervallo $\mathcal{I} = [0.6, 0.8]$ ed un valore interno all'intervallo stesso (ad esempio $x_1 = 0.69$), a scelta (con tolleranza `toll = 1e-8` e `nmax = 20`) e salvando le tre figure corrispondenti agli scarti in valore assoluto, in formato pdf (nel titolo si scriva quale valore iniziale x_0 si è scelto).
- Si applichi il metodo della bisezione nell'intervallo $\mathcal{I}$ con la stessa tolleranza e `nmax = 30` e si salvi la figura corrispondente.
- Che cosa si può notare paragonando i vari risultati di Newton ottenuti variando il valore iniziale (successione delle iterate e degli scarti corrispondenti) ed anche quelli ottenuti con il metodo di bisezione?
- Si utilizzi poi nuovamente lo script di Newton, definendo come valore iniziale $x_0 = -10$ e stessa tolleranza. Che cosa si può dedurre dai risultati ottenuti?
  Nota bene: si commentino i risultati con particolare riferimento all'ordine di convergenza esibito dai due metodi (Newton e bisezione) e alla bontà dei diversi punti iniziali scelti per approssimare α_2 con il metodo di Newton.

Esercizio 4

Si desidera calcolare la radice approssimata positiva e **non nulla** della funzione

$$f(x) = 1 - x - e^{-2x}. \tag{1}$$

usando il seguente metodo iterativo (chiamato della **Secante** o della Secante Variabile)

$$x_{n+1} = x_n - f(x_n)\,\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})},\ n \geq 1.$$

Si localizzi graficamente, la soluzione reale positiva (non nulla) dell'equazione $f(x) = 0$ e, dopo aver trovato un intervallo sufficientemente piccolo $[x_0, x_1]$ (di ampiezza non maggiore di 0.5) che contenga la soluzione, si salvi la figura (formato ed estensione `.fig` o `.pdf`).

Si scriva una function di nome `secvar.m` che implementi l'algoritmo relativo a tale metodo, ed uno script, con le seguenti indicazioni:

- La function `secvar.m` deve avere come dati in ingresso la funzione (inline function) $f \to$ `f`, precedentemente generata nello script con il comando `inline`, da una stringa inserita con il comando `input`. Devono essere anche forniti in ingresso, i due *valori iniziali* $x_0 \to$ `x0` e $x_1 \to$ `x1`, la *tolleranza* $\varepsilon \to$ `toll` per il test di arresto (basato sul valore assoluto della differenza di due iterate successive) ed il numero massimo di iterazioni consentite $n_{max} \to$ `nmax`;
- la stessa function deve dare come uscita il vettore `xv` che contiene le iterate, approssimazioni successive della soluzione (inclusi i valori iniziali x_0 e x_1), il vettore `scarti` con le differenze tra iterate successive e il numero `n` di iterazioni effettuate;
- la function dovrà avere la seguente intestazione:

```
function [xv, scarti, n] = secvar (f, x0, x1, toll, nmax)
%SECVAR Metodo della secante variabile per equazione non lineare
%
% Uso:
%   [xv, scarti, n] = secvar(f, x0, x1, toll, nmax)
%
% Dati di ingresso:
%   f:      funzione
%   x0:     prima iterata
%   x1:     seconda iterata
%   toll:   tolleranza richiesta per il valore assoluto
%           tra due iterate successive (scarto)
%   nmax:   massimo numero di iterate permesse
%
% Dati di uscita:
%   xv:     vettore contenente le iterate
%   scarti: vettore contenente i corrispondenti scarti
%   n:      numero di iterate della successione
```

- Lo script `secvarscript.m` deve chiedere all'utente la stringa `exprf` contenente la funzione, i due valori iniziali, la tolleranza, il numero n_{max};
- ottenuti gli argomenti di uscita dalla funzione (risultati), lo script deve visualizzare a video (comando `fprintf`) l'ultima soluzione approssimata determinata dalla function, l'ultimo scarto, l'ultimo residuo e l'indice dell'ultima iterata calcolata;
- lo script deve dare un opportuno messaggio d'errore se vengono effettuate $n = n_{\max}$ iterazioni;
- lo script deve generare, in una finestra grafica diversa dalla precedente (comando `figure`), un grafico che rappresenti (scala logaritmica sull'asse y) il valore assoluto del vettore che contiene gli scarti (`scarti`). Il grafico deve essere salvato in formato .pdf.
- lo script, alla fine, deve scrivere in maniera ordinata come colonne di una tabella (con opportuna intestazione) e usando il comando `fprintf`, i risultati forniti dal programma `secvar`. Si prenda spunto dalla function `risultati_newton` creata all'esercizio 2.