Laboratorio Calcolo Numerico

Esercizio 1 (comando Matlab `num2str`, vettori con componenti stringa, uso del carattere : per forzare che un vettore (riga o già colonna) sia sempre vettore colonna, dato che il ciclo `while` crea vettori riga, visualizzazione sotto forma di tabella di numeri)

Si vuole determinare la soluzione α_2 appartenente all'intervallo $\mathcal{I} = [0.6, 0.8]$ dell'equazione non lineare $f(x) = x^2 - 1 + e^{-x} = 0$ (la stessa funzione utilizzata per il metodo di Bisezione del precedente laboratorio che ha una soluzione esatta in $\alpha_1 = 0$), 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 (**da calcolare a mano**) non si annulla mai nell'intervallo $\mathcal{I}$.
Si desidera scrivere un algoritmo per uno script che permetta di implementare tale metodo, con le stesse modalità utilizzate per quello della Bisezione. Tale algoritmo deve poi essere tradotto in uno script di nome `newtonscript.m`. L'algoritmo e lo script si serviranno di un algoritmo (che diventerà una function Matlab di nome `newtonfun.m`) che implementa il metodo di Newton.

- Si inizia scrivendo l'algoritmo per lo script. L'algoritmo avrà bisogno di leggere la funzione f e la sua derivata f', la tolleranza, il numero massimo di iterate ammesse $n_{\max}$ ed il valore iniziale x_1 (per rendere più facile la traduzione in Matlab è meglio indicare la prima iterata con indice 1, anzichè 0).
- I risultati che devono essere visualizzati sono, oltre ai dati precedentemente letti ed assegnati, l'ultima soluzione approssimata determinata dal metodo, il relativo residuo, ed il numero di iterazioni effettuate. Si deve prevedere anche di visualizzare la successione delle iterate e dei corrispondenti residui (possibilmente mettendoli in due colonne di una tabella, utilizzando il `format long e`), ed una eventuale figura.
- L'algoritmo deve dare un opportuno messaggio se vengono calcolate $n = n_{\max}$ iterate o se durante il procedimento di calcolo delle iterate si verifica la condizione $f'(x_n) = 0$ (gestita da una variabile `flag` $\neq 0$ trasmessa dalla funzione tra i parametri di uscita).
- Alla fine, si preveda anche un grafico che rappresenti (scala logaritmica sull'asse y) il valore assoluto del vettore che contiene i residui (`fxv`), e nel titolo il valore iniziale x_1.
  Un possibile algoritmo quindi è

  > **leggi e assegna** f funzione
  > **leggi e assegna** f' funzione derivata
  > **leggi e assegna** `toll` tolleranza
  > **leggi e assegna** `nmax` numero massimo di iterate
  > **leggi e assegna** x_1 valore iniziale
  > **esegui** `[xv, fxv, n, flag] = newtonfun (f, f1, x1, toll, nmax)`
  > **if** `n` = `nmax` **then**
  >   **print** raggiunto il numero massimo di iterate
  > **else if** `flag` $\neq$ `0` **then**
  >   **print** la derivata prima si e' annullata
  >   **print** `n` numero di iterate calcolate
  > **else**
  >   **print** `xv(n)` ultima approssimazione calcolata
  >   **print** `fxv(n)` ultimo residuo
  >   **print** `n` numero di iterate
  >   **print** `xv, fxv`
  >   **definisci** `x(i), i= 1:n` (ascisse che servono SOLO per il disegno)

disegna i residui per punti (x(i),abs(fxv(i))), i= 1:n
end if

- I parametri di ingresso della function Matlab `newtonfun.m` devono essere: **sia** la funzione $f \rightarrow$ `f`, **sia** la *derivata* $f' \rightarrow$ `f1`, entrambe di tipo anonymous functions, il *valore iniziale* $x_1 \rightarrow$ `x1`, la *tolleranza* $\varepsilon \rightarrow$ `toll` ed il numero massimo di iterate consentite $n_{max} \rightarrow$ `nmax`;
- I parametri in uscita devono essere: il vettore `xv` che contiene tutte le iterate calcolate, approssimazioni successive della soluzione (incluso il valore iniziale x_1), il vettore `fxv` dei residui calcolati nelle corrispondenti iterate, il numero `n` di iterate calcolate e una variabile di controllo `flag` che indichi un'eventuale divisione per zero;
- la function dovrà avere la seguente intestazione:

```
function [xv, fxv, n, flag] = newtonfun (f, f1, x1, toll, nmax)
%NEWTONFUN Metodo di Newton
% Uso:
%   [xv, fxv, n, flag] = newtonfun (f, f1, x1, toll, nmax)
%
% Dati di ingresso:
%   f:      funzione (anonymous function)
%   f1:     derivata prima (anonymous function)
%   x1:     valore iniziale
%   toll:   tolleranza richiesta per il valore assoluto
%           della differenza di due iterate successive
%   nmax:   massimo numero di iterate permesse
%
% Dati di uscita:
%   xv:     vettore contenente le iterate
%   fxv:    vettore contenente i corrispondenti residui
%   n:      numero di iterate
%   flag:   Se flag = 1 la derivata prima si e' annullata
```

ed un possibile algoritmo del Metodo di Newton è

$$[\texttt{xv}, \texttt{fxv}, \texttt{n}, \texttt{flag}] = \textbf{Newton}\ (f, f', x_1, \texttt{toll}, n_{\max})$$

$n = 1$
$\texttt{flag} = 0$
$\texttt{diff} = \texttt{toll} + 1$
$\texttt{xv}(1) = x_1$
$\texttt{fxv}(1) = f(\texttt{xv}(1))$
while $|\texttt{diff}| \geq \texttt{toll}$ and $n < n_{\max}$ and $\texttt{flag} = 0$ **do**
  $\texttt{f1x} = f'(\texttt{xv}(\texttt{n}))$
  if $\texttt{f1x} = 0$ **then**
    $\texttt{flag} = 1$
  else
    $\texttt{diff} = -\texttt{fxv}(\texttt{n})/\texttt{f1x}$
    $\texttt{xv}(\texttt{n}+1) = \texttt{xv}(\texttt{n}) + \texttt{diff}$
    $\texttt{fxv}(\texttt{n}+1) = f(\texttt{xv}(\texttt{n}+1))$
    $n = n + 1$
  end if
end while

Esercizio 2

- Si utilizzi lo script tre volte, definendo tre diversi valori iniziali: l'estremo sinistro dell'intervallo $\mathcal{I} = [0.6, 0.8]$, l'estremo destro dello stesso intervallo, un valore a scelta all'interno dell'intervallo stesso, mantenendo la stessa tolleranza `toll = 1e-8` ed `nmax = 20`). Si scelgano, ad esempio, i tre valori $x_1 = 0.6$, $x_1 = 0.69$ e $x_1 = 0.8$). Si salvino le tre figure corrispondenti alla rappresentazione dei residui in valore assoluto, in formato pdf (nel titolo si scriva a quale valore iniziale x_1 corrisponde la figura).
- Si applichi il metodo della bisezione nell'intervallo $\mathcal{I}$ con la stessa tolleranza e `nmax = 30` e si salvi la figura corrispondente, sempre in pdf.
- Che cosa si può notare paragonando i vari risultati di Newton ottenuti variando il valore iniziale (successione delle iterate e residui corrispondenti) ed anche quelli ottenuti con la Bisezione?
- Si utilizzi poi nuovamente lo script di Newton, definendo come valore iniziale $x_1 = -10$, stessa tolleranza e stesso `nmax = 20`. Che cosa si può dedurre dai risultati ottenuti?

Esercizio 3

Si scriva un'altra function Matlab di nome `newtonfunI`, che utilizzi per la creazione dei vettori `xv` e `fxv` la modalità Matlab di *aggiunta di componenti ad un vettore esistente*. Si modifichi nello script **solamente** la chiamata di esecuzione della funzione che implementa il metodo di Newton scrivendo

```
[xv, fxv, n, flag] = newtonfunI (f, f1, x1, toll, nmax);
```

e si testi che i risultati siano gli stessi di quelli ottenuti con la function `newtonfun`.

Esercizio 4

Si costruisca un algoritmo e poi una function di nome `stimap` che permetta di costruire, a partire da una successione data, le stime dell'ordine di convergenza (formula di pagina 108; non si consideri l'algoritmo di pag. 109, ma si usi solo la formula in un ciclo fisso `for`), e si richiami tale funzione al fine di calcolare le stime per la successione precedentemente ottenuta con un metodo iterativo qualsiasi (ad esempio la Bisezione o Newton).
Si inseriscano gli opportuni controlli tramite strutture IF (numero di termini della successione da considerare minore o uguale alle componenti del vettore; controllo sui denominatori che non possono essere nulli). La funzione deve avere per intestazione:

```
function p = stimap (xv, nt)
%STIMAP  Stima dell'ordine di convergenza di una successione
%
% Uso:
%   p = stimap (xv, nt)
%
% Dati di ingresso:
%   xv:     vettore colonna contenente le iterate della successione.
%   nt:     numero dei termini della successione da considerare per la stima.
%
% Dati di uscita:
%   p:      vettore contenente le stime dell'ordine calcolate.
%           Le prime tre componenti del vettore sono sempre nulle.
```

Si chieda l'esecuzione di questa function all'interno degli script predisposti per il Metodo di Bisezione e per il Metodo di Newton, utilizzando il vettore `xv` che viene restituito e visualizzando il vettore delle stime. Se possibile, si faccia anche un grafico di tali stime, considerandole in valore assoluto.
Lo si applichi, ad esempio, al caso del Metodo di Newton con valore iniziale $x_1 = -10$.
Domanda: dalle stime ottenute si può ipotizzare quale sia l'ordine di convergenza del Metodo scelto?