

Calcolo Numerico (Ingegneria Energia/Meccanica, Canale B)
Prova di Laboratorio II, del 1 Luglio 2022

Cognome e nome _____ Matricola _____

Postazione _____

FIRMA PER CONSEGNARE _____

FIRMA PER RITIRARSI _____

1. SI RACCOMANDA AGLI STUDENTI DI **commentare adeguatamente** SCRIPT E FUNCTION MATLAB.

2. OGNI PORZIONE DI CODICE **deve avere** QUALE PRIMA RIGA UN COMMENTO MEDIANTE % CON NOME, COGNOME, NUMERO DI MATRICOLA E POSTAZIONE.

1. Si determino le iterazioni successive (relative al problema **punto fisso** $x = \phi(x)$)

$$x_{k+1} = \phi(x_k) \quad (1)$$

in cui $x_k \in [a, b]$ e $\phi : [a, b] \rightarrow [a, b]$ é tale che $\phi \in C([a, b])$.

A tal proposito,

- Si crei una function di nome `iterazionisuccessive.m` che implementi l'algoritmo relativo a tale metodo, avente come input:
 - (a) la funzione `phi`,
 - (b) un vettore `estremi` che ha per componenti, nell'ordine, gli estremi a e b ,
 - (c) il punto iniziale `x0` appartenente all'intervallo $[a, b]$
 - (d) il numero massimo di iterazioni di punto fisso `maxit`,
 - (e) la tolleranza dello step `tol`.
- la stessa function deve dare in output
 - (a) il vettore `x` che contiene le iterazioni di punto fisso compiute;
 - (b) il vettore `step` che contiene gli step delle iterazioni;
 - (c) il vettore `flag` che è uguale a 0 se il metodo di punto fisso non ha avuto successo, 1 se ha avuto successo.
- la function dovrà avere la seguente intestazione:

```
function [x,step,flag] = iterazionisuccessive (phi,estremi,x0,maxit,tol)
% Uso:
% [x,flag] = iterazionisuccessive (phi,estremi,x0,maxit,tol)
%
% Dati di ingresso:
% phi: funzione di punto fisso;
% estremi: vettore riga con componenti gli estremi "a", "b" di integrazione;
% x0: punto iniziale in [a,b]
% max: numero intero positivo (numero massimo di iterazioni).
% tol: numero reale positivo (tolleranza step).
%
% Dati di uscita:
% x: vettore delle iterazioni successive.
% flag: 0 uscita senza successo, 1: uscita con successo.
```

- lo script testi se `maxit` è strettamente positivo; se ciò non si verifica ponga `x=NaN`, `flag=0` ed esca dalla routine;
- lo script testi se `maxit` è uguale a `floor(maxit)`; se ciò non si verifica ponga `x=NaN`, `flag=0` ed esca dalla routine;
- si assegni in `a` la prima componente di `estremi` e in `b` la seconda componente di `estremi`;
- lo script testi se `a` è strettamente minore di `b`; se ciò non si verifica ponga `x=NaN`, `flag=0` ed esca immediatamente dalla routine;
- lo script testi se `a` è inferiore o uguale a `x0`; se ciò non si verifica ponga `x=NaN`, `flag=0` ed esca immediatamente dalla routine;
- lo script testi se `b` è maggiore o uguale a `x0`; se ciò non si verifica ponga `x=NaN`, `flag=0` ed esca immediatamente dalla routine;
- si assegni ad `x` il valore `x0`;
- si ponga `flag` uguale a 0;
- si ponga `step` uguale al vettore senza componenti;

(continua →)

- si definisca un ciclo-for con `k` da 1 a `maxit` in cui alla k -sima iterazione:
 - (a) si ponga `x(end+1)` uguale a `phi(x(end))` (utilizzare il comando `feval` con argomenti `phi` e `x(end)`);
 - (b) se `x(end)` è strettamente minore di `a` o strettamente maggiore di `b`, si esca immediatamente dalla routine;
 - (c) si ponga `step(end+1)` uguale al valore assoluto della differenza di `x(end)` con `x(end-1)`;
 - (d) se `step(end)` è minore o uguale a `tol` si ponga `flag` uguale a 1 e si esca immediatamente dalla routine (si usa `return` o `break`?).

2. Si scriva una function `demo_iterazioni_successive` che

- mediante un comando del tipo `phi=@(x) ...` definisca la funzione vettoriale $\phi(x) = 0.3 \cdot \sin(x^2) + 1$,
- assegni alla prima componente di `estremi`, il valore 0 e alla seconda componente di `estremi`, il valore 2;
- quale `x0` si ponga il valore 1, quale `maxit` si ponga il valore 10000, quale `tol` si ponga `1e-12`;
- si chiami la routine `iterazionisuccessive` con tali dati e si ponga l'output nelle variabili `x`, `step` e `flag`;
- si ponga in `xend` l'ultima componente di `x`;
- si visualizzi su monitor `xend` con 1 cifra prima della virgola e 15 dopo la virgola, in formato decimale;
- si ponga in `stepend` l'ultima componente di `step`;
- si visualizzi su monitor `stepend` con 1 cifra prima della virgola e 3 dopo la virgola, in formato esponenziale;
- si calcoli la lunghezza `L` del vettore `x`;
- si visualizzi su monitor `L` con 5 cifre prima della virgola e nessuna dopo la virgola, in formato decimale;

3. nella tabella che segue si scrivano i valori ottenuti relativamente a `xend`, `stepend`, `L`.

<code>xend</code>	<code>stepend</code>	<code>L</code>
1.298048177620551e+00	7.370e-13	13