

ESERCIZIO N.1: Il Metodo di Bisezione e i criteri d'arresto.

Ricorda l'algoritmo di bisezione:

```
fa = f(a)
Per k = 0, 1, 2, ...
  x = (a+b)/2;
  y = f(x);
  Criterio d'arresto
  Se fa y < 0 allora b=x, altrimenti a=x, fa=y
```

Implementa la funzione `bisezione.m` con la seguente chiamata:

```
[ind,x,y,k,afinal,bfinal]=bisezione(f,a,b,tolla,tollr,tollf,kmax,stampe);
```

In output: `ind` è un indicatore del tipo di uscita (per es., `ind=1` convergenza per criterio sulla lunghezza dell'intervallo, `ind=2` convergenza per criterio sul residuo, `ind=-1` per raggiungimento di `kmax`); `x` è l'approssimazione dello zero; `y=f(x)`; `k` numero di iterazioni a convergenza; `afinal`, `bfinal` estremi dell'intervallo a convergenza.

Ad ogni iterazione k l'algoritmo deve mostrare i valori ottenuti come nella seguente tabella

k	x_k	$f(x_k)$	a	b	$b-a$

Implementa il criterio d'arresto come segue:

```
% criterio di arresto su f
if abs(y)<=tollf, ind=2; return, end
% criterio di arresto su [a,b]
if b-a<= tolla+tollr*abs(a), ind=1; return, end
```

dove `ind` indica con quale criterio d'arresto la funzione è uscita.

Considera la funzione

$$f_1(x) = \alpha \sin(x) \cos(x) - x^3, \quad x \in [-1, 1]$$

1. Localizza graficamente lo zero $x^* = 0$ usando `fplot` al variare di $\alpha \in \{1, -10^{-7}, 10^7\}$. Calcola $f'_1(x^*)$ e osserva la sua dipendenza da α .
2. Usa `bisezione.m` in $[a, b] = [-0.2, 0.1]$ per approssimare lo zero di f_1 al variare di $\alpha \in \{1, -10^{-7}, 10^7\}$ ponendo $tolla = tollr = 0$ e $tollf = 10^{-8}$. Osserva come varia l'errore al variare di α (nota che essendo $x^* = 0$, $e_k = x_k$).
3. Per $\alpha = 1$ calcola il numero k di iterazioni necessarie affinché $|e_k| < 10^{-9}$ e verificalo sperimentalmente. Riporta su un grafico la storia della convergenza, in termini del residuo $|f(x_k)|$.

ESERCIZIO N.2: Il Metodo di Newton.

Implementa il metodo di Newton `newton1d.m` con i corrispondenti criteri d'arresto. In particolare, localizza graficamente lo zero semplice $x^* = 0$ di (`help atan`)

$$f(x) = \arctan(x), \quad x \in [-1, 1].$$

1. Usa `newton1d.m` con $x_0 = 1.3$ e $x_0 = 1.4$ con $tolla = 10^{-8}$, $tollr = 0$, $tollf = 10^{-8}$.
2. Usa `bisezione.m` in $[a, b] = [-0.5, 3.1]$ con $tolla = 10^{-8}$, $tollr = 0$, $tollf = 10^{-8}$.

Osserva i diversi comportamenti del Metodo di Newton al variare di x_0 e confrontali graficamente con il Metodo di Bisezione.

ESERCIZIO N.3: Il Metodo delle Secanti.

- i) A partire dalla function `newton1d.m`, crea la function `secanti.m` che implementi il Metodo¹.
Nota: L'implementazione deve prevedere una sola valutazione di f per iterazione.
- ii) Confronta graficamente la velocità di convergenza dei metodi di Bisezione, Newton e Secanti nel calcolo di uno zero della funzione

$$f(x) = x^3 - 2x - 5$$

(dati iniziali: Metodo di Bisezione $[a, b] = [2, 3]$, Metodo di Newton $x_0 = 3$, Metodo delle Secanti $x_0 = 2, x_{-1} = 3$).

- iii) Stima l'ordine di convergenza dei metodi, sapendo che $x^* \approx 2.094551481542328$, riportando con un display tale stima per i tre metodi al crescere del numero di iterazioni.

Nota: Ricorda che l'ordine p soddisfa $p \approx \log(e_k)/\log(e_{k-1})$ per k sufficientemente grande, con $e_k = |x_k - x^*|$.

ESERCIZIO N.4: Il Metodo di Newton per zeri di polinomi.

- i) Implementa il metodo di Newton per polinomi, in cui la valutazione del polinomio $p(x^{(k)})$ e della sua derivata $p'(x^{(k)})$ sono fatte mediante la regola di Horner,

$$\begin{aligned} b_0 &= a_n, c_0 = b_0 \\ b_j &= b_{j-1}x^{(k)} + a_{n-j}, \quad j = 1, \dots, n \\ c_j &= c_{j-1}x^{(k)} + b_j, \quad j = 1, \dots, n-1 \end{aligned}$$

- ii) Applica il metodo alla funzione polinomiale dell'esercizio precedente e confronta i risultati con quelli di `newton1d.m`.

ESERCIZIO N.5: Facoltativo.

È data la funzione

$$f(x) = \int_0^x \exp(-t)(t-1)dt, \quad x \in [a, b] = [0, 5].$$

1. Fai il grafico della funzione f (Figura 1) ed individua un intervallo opportuno dove possa essere il suo punto di minimo. Stima il valore di f mediante una **tua** formula di quadratura composita, con un numero di sottointervalli superiore a 10.
2. Scrivi la funzione matlab `[valmax]=minf(f1,x0,tolr,tola,tolf,maxit);` che approssima il punto critico x^* di f tale che $x^* = \operatorname{argmin}_{x \in [a,b]} f(x)$ mediante il metodo di Newton, con parametri `tolf=tola=tolr=1e-10`, `maxit=100` (usa questi parametri per tutta la prova, determina la funzione `f1` adeguata). Fai il display descrittivo della storia della convergenza (num.iterazione, valore del residuo, valore dell'iterata x_k) e riportala su un grafico (Figura 2), inserendo etichette e titolo.

Nel caso tu abbia bisogno di fare la derivata di una funzione g , ricorda che la derivata prima di g in un punto x può essere approssimata mediante differenze finite centrate,

$$g'(x) \approx \frac{g(x+h) - g(x-h)}{2h}, \quad h > 0.$$

¹Iterazione del metodo delle secanti: $x^{(k+1)} = x^{(k)} - f(x^{(k)})/q_k$, con $q_k = (f(x^{(k)}) - f(x^{(k-1)}))/(x^{(k)} - x^{(k-1)})$.

3. Verifica che il valore ottenuto $(x_{\bar{k}}, f(x_{\bar{k}}))$ corrisponda al valore cercato aggiungendolo al grafico di f in Figura 1.