

Calcolo Numerico Tutoraggio, lezione 7

AUTORE: ALVISE SOMMARIVA

Tempo previsto: 60 minuti. **Difficoltà:** ●●●○○

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

Si supponga che $f : [a, b] \rightarrow \mathbb{R}$ sia una funzione continua invertibile, con $f(a) \cdot f(b) < 0$ e si supponga di voler calcolare $x^* \in (a, b)$ tale che $f(x^*) = 0$.

Se è nota una tabella di coppie (x_i, y_i) , $i = 0, \dots, n$, con $x_i \neq x_j$ qualora $i \neq j$ e $y_i = f(x_i)$, si può calcolare il polinomio p_n tale che $p_n(y_i) = x_i$ (attenzione alle variabili!) e di seguito valutare $\bar{x} = p_n(0)$. Tale metodo, noto come *interpolazione inversa*, permette di fornire un'approssimazione \bar{x} dello zero x^* .

1. A tale scopo si definisca una funzione Matlab `interpolazione_inversa`

- abbia la seguente intestazione:

```
function xbar=interpolazione_inversa(x,y)

% Oggetto:
% Routine che implementa la interpolazione inversa di una funzione
% invertibile e continua "f" in un intorno del suo zero x*.
%
% Input:
% x: vettore di ascisse a due a due distinte;
% y: valori assunti da "f" nelle componenti di "x";
%
% Output:
% xbar: approssimazione dello zero di "f", pari al vettore nullo se "flag=1".
% flag: se vale 0 il processo e' andato a buon fine, 1 altrimenti.
```

- ponga `flag=0`;
- calcoli il minimo `a` del vettore `x` e determini in quale componente `ia` di `x` si trovi (aiutarsi con l'help di Matlab relativamente a `min`),
- calcoli il massimo `b` del vettore `x` e determini in quale componente `ib` di `x` si trovi (aiutarsi con l'help di Matlab relativamente a `max`),
- se `y(ia)*y(ib) > 0` allora
 - ponga `xbar=[]` e `flag=1`;
- se `y(ia)*y(ib) <= 0` allora

- mediante il comando `polyfit` calcoli il polinomio p_n , avente opportuni coefficienti `p`, che interpola le $n + 1$ coppie le cui componenti sono immagazzinate in `y` e `x` (attenzione all'ordine!);
- valuti `xbar = p_n(0)` mediante il comando `polyval` (in che punto si deve valutare p_n ?);
- si ragioni se serva porre o meno `flag=0`.

2. Si definisca una function `demo_interpolazione_inversa` che

- definisca un vettore `x`, le cui componenti sono [2.1 2.2 ... 2.8 2.9];
- definisca un vettore `y` ottenuto valutando la funzione di Bessel J_0 nei punti che sono componenti di `x` (aiutarsi con l'help di Matlab della funzione `besselj` per `NU=0`);
- mediante la routine `interpolazione_inversa` approssimare con `xbar` lo zero x^* della funzione di Bessel J_0 ;
- stampare il valore di `xbar` con 1 cifra prima della virgola e 15 dopo la virgola, in formato esponenziale;
- sapendo che la soluzione è $x^* \approx 2.404825557695780$, calcolare l'errore assoluto `err` e stamparlo con una cifra prima della virgola e 3 dopo la virgola in formato esponenziale;

3. si scrivano tali valori nella seguente tabella:

<code>xbar</code>	<code>err</code>