

Calcolo Numerico

Tutoraggio, lezione 11

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 implementi il metodo iterativo di **Newton-Fourier** che genera due sequenze

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (1)$$

$$z_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)} \quad (2)$$

che permette di determinare la zero di f con dati iniziali x_1, z_1 . Si faccia attenzione della frazione di (2). I valori iniziali devono essere scelti cosicchè $x_1 < z_1$ e nell'intervallo $[x_1, z_1]$ ci sia uno zero di f .

A tal proposito,

- Si crei una function di nome `newton_fourier` che implementi l'algoritmo relativo a tale metodo, avente come input:
 - (a) le funzioni `f`, `f1`,
 - (b) i due valori iniziali `x1` e `z1`,
 - (c) la tolleranza `toll` per il test di arresto (basato sul valore assoluto della differenza tra x_n e z_n),
 - (d) il numero massimo di iterazioni consentite `nmax`;
- la stessa function deve dare in output
 - (a) il vettore `xv` che contiene le iterate x_1, \dots, x_n ,
 - (b) il vettore `zv` che contiene le iterate z_1, \dots, z_n ;
- la function dovrà avere la seguente intestazione:

```
function [xv,zv] = newton_fourier(f,f1,x1,z1,toll,nmax)
% Uso:
% [xv,zv] = newton_fourier(f,f1,x1,z1,toll,nmax)
% Dati di ingresso:
% f:      funzione
% f1:     derivata di f
% x1:     prima iterata in x
% z1:     prima iterata in z (con x1 < z1)
% toll:   tolleranza richiesta per il valore assoluto
%         tra le iterate di xv e zv
% nmax:   massimo numero di iterate permesse
% Dati di uscita:
% xv:     vettore contenente le iterate in x
% zv:     vettore contenente le iterate in z
```

- lo script inizialmente, ponga `xv(1)=x1`, `zv(1)=z1`; se `xv(1)` è maggiore o uguale a `zv(1)`, esca mediante `return` dopo aver fornito il messaggio di errore **dati iniziali non accettabili**;
- lo script mediante un ciclo-for con `n=1:nmax`
 - valuti $f'(x_n)$ e la assegni a `f1xv(n)` e se è nullo tale valore esca mediante `return` dopo aver fornito il messaggio di errore **la derivata si annulla nel punto in considerazione**;
 - valuti $f(x_n)$ e la assegni a `fxv(n)`;
 - calcoli il valore di x_{n+1} e lo assegni a `xv(n+1)`;
 - calcoli il valore di z_{n+1} e lo assegni a `zv(n+1)`;
 - esca dalla routine mediante `return` qualora

$$\text{abs}(xv(n+1)-zv(n+1)) \leq \text{toll}; \quad (3)$$

altrimenti prosegua con le iterazioni.

- se si esce dal ciclo-for per un numero di iterazioni superiori a `nmax`, scriva su monitor il messaggio di errore **sono state effettuate troppe iterazioni**;

(continua →)

2. Si scriva una function `demo_I` che mediante un comando del tipo `f=@(x) ...` definisca la funzione $f(x) = x^4 - 10$, ad `f1` la sua derivata, e ponga `x1=1`, `z1=2`, `toll` pari a 10^{-6} , `nmax=1000`. Di seguito risolva mediante `newton_fourier` l'equazione $f(x) = 0$ in oggetto, calcolando lo zero positivo. Si scrivano sotto l'ultima componente di `xv` e `zv` con 1 cifra prima della virgola e 15 dopo la virgola, in formato decimale, come pure la loro differenza in modulo `err` con 1 cifra prima della virgola e 2 dopo la virgola, in formato esponenziale.

<code>xv(end)</code>	<code>fxv(end)</code>	<code>err</code>

3. Si scriva una function `demo_II` che

- definisca la funzione $f(x) = \log(x) - 100$ mediante il comando `@` (attenzione, il logaritmo è in base e) e la assegni a `f`;
- definisca la funzione f' mediante il comando `@` e la assegni a `f1`;
- si pongano `x1=3`, `z1=8`, `toll` pari a 10^{-12} , `nmax=1000`;
- si calcolino con `newton_fourier` le sequenze di $\{x_n\}_{n=1,\dots,N}$, $\{z_n\}_{n=1,\dots,N}$
- si esegua il grafico, in scala semilogaritmica, delle coppie $(k, |x_k - z_k|)$, con $k = 1, \dots, M$ (qui M è la lunghezza del vettore `xv`) e lo si salvi sul file `errore.jpg`;
- scriva su monitor
 - (a) l'ultima componente di `xv` con 1 cifra prima della virgola e 15 dopo la virgola, in notazione esponenziale,
 - (b) l'ultima componente di `zv` con 1 cifra prima della virgola e 15 dopo la virgola, in notazione esponenziale;
 - (c) il valore assoluto `err` dell'ultima componente di `xv-zv` con 1 cifra prima della virgola e 3 dopo la virgola, in notazione esponenziale;
- si scrivano tali risultati nella tabella

<code>xv(end)</code>	<code>fxv(end)</code>	<code>err</code>