

**Calcolo Numerico (Ingegneria Energia/Meccanica, Canale A)**  
**Prova di Laboratorio II, del 10 Luglio 2019, fila II**

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. Sia  $n$  un numero naturale positivo. L'insieme di  $n + 1$  nodi di Chebyshev, scalati in  $[a, b]$ , sono definiti da

$$x_k = \frac{b+a}{2} + \frac{b-a}{2} \cdot \cos\left(\frac{(2k+1)\pi}{2n+2}\right), \text{ ove } k = 0, 1, \dots, n.$$

Si definisca una function `punti_chebyshev` che fissato dall'utente  $n$  definisca l'insieme dei punti  $\{x_k\}_{k=0,\dots,n}$ . La funzione dovrà avere la seguente intestazione:

```
function x=punti_chebyshev(n,a,b)

% punti_chebyshev: Punti di Chebyshev (scalati in [a,b]).
%
% Dati di ingresso:
% n: numero naturale.
% a,b: estremi dell'intervallo [a,b]
%
% Dati di uscita:
% x: vettore di lunghezza (n+1), contenente i punti scalati di Chebyshev,
% unisolventi a grado "n".
```

In particolare

- (a) dovrà controllare che  $a < b$  altrimenti ponga "x=[];" ed esca mediante `return`;
  - (b) dovrà controllare che  $n > 0$  altrimenti ponga "x=[];" ed esca mediante `return`;
2. Si definisca un file `demo_I` che determini i punti di Chebyshev unisolventi a grado  $n = 20$ , nell'intervallo  $[-10, 10]$ . Si scrivano, nella tabella che segue, la prima e l'ultima componente di  $x$  con 1 cifra prima della virgola e 6 dopo la virgola, in formato esponenziale.

x(1)	x(end)

3. Si definisca una function `esperimento_sub` avente la seguente intestazione

```
function i_err=esperimento_sub(m,n,f,a,b)

% Oggetto:
% errore approssimante polinomiale ai minimi quadrati di grado "m" di f
% in "n" punti di Chebyshev (scalati in [a,b]).
%
% Dati di ingresso:
% m: grado dell'approssimante polinomiale ai minimi quadrati
% n: numero di nodi di Chebyshev.
% f: funzione da approssimare in [a,b].
% a,b: estremi dell'intervallo [a,b]
%
% Dati di uscita:
% i_err: numero reale non negativo che rappresenta una stima dell'errore di approssimazione
```

(continua →)

che effettui quanto segue:

- (a) controlli che  $m < n$  altrimenti ponga "i\_err=[];" ed esca mediante `return`;
  - (b) generi il vettore  $\mathbf{x}$  di  $n$  punti di Chebyshev scalati in  $[a, b]$  (fare attenzione al numero di punti generati da `punti_chebyshev`);
  - (c) valuti la funzione  $f$  in ogni componente del vettore  $\mathbf{x}$  definendo così il vettore di valutazioni  $\mathbf{y}$ ;
  - (d) utilizzando un opportuno comando Matlab determini i coefficienti del polinomio  $p_m$ , approssimante polinomiale ai minimi quadrati di grado  $m$  (relativamente alle coppie  $(x_k, y_k)$  con  $k = 1, \dots, n$ );
  - (e) definisca il vettore  $\mathbf{v}$  di 20000 punti equispaziati nell'intervallo  $[a, b]$  (deve contenere i punti agli estremi);
  - (f) utilizzando un opportuno comando Matlab, valuti il polinomio  $p_m$  in ogni componente del vettore  $\mathbf{v}$ , definendo così il vettore di valutazioni  $\mathbf{pv}$ ;
  - (g) valuti la funzione  $f$  in ogni componente del vettore  $\mathbf{v}$ , definendo così il vettore di valutazioni  $\mathbf{fv}$  (senza utilizzare cicli-for);
  - (h) calcoli la stima del massimo errore di approssimazione quantità  $\max_{k=1, \dots, 20000} |f(v_k) - p_m(v_k)|$  mediante un opportuno comando Matlab (senza cicli-for).
4. Si definisca uno script `esperimento` che effettui quanto segue:
- (a) tramite il comando `warning off` non mostri messaggi *warning* di Matlab;
  - (b) definisca la funzione  $f(x) = \exp(-x^3) + \cos(x)$  in  $[0, \pi/2]$ , ovvero `f=@(x) ...` e assegni `a=0` e `b=pi/2`;
  - (c) mediante un opportuno ciclo-for, al variare di  $k$  da 1 a 8, ponga nella  $k$ -sima componente di `i_err` il valore ottenuto dalla routine `esperimento_sub` per  $m_k = 2^k$  e  $n = 15000$ ;
  - (d) registri (in modo leggibile!) sul file `errori.txt` le coppie  $(m(k), i\_err(k))$  per  $k = 1, \dots, 8$ , ove siano rappresentati
    - `m(k)` in formato decimale con 4 cifre prima della virgola e nessuna dopo la virgola,
    - `i_err(k)` in formato esponenziale con 1 cifra prima della virgola e 15 dopo la virgola,(si usi la specifica `'w'` nel comando `'fopen'`);
  - (e) si stampino su monitor, la prima e l'ultima componente di `i_err` con 1 cifra prima della virgola e 6 dopo la virgola, in formato esponenziale;
  - (f) si scrivano, nella tabella che segue, la prima e l'ultima componente di `i_err` con 1 cifra prima della virgola e 6 dopo la virgola, in formato esponenziale.

<code>i_err(1)</code>	<code>i_err(end)</code>