

**Corso di Laurea Magistrale in  
Ingegneria Biomedica  
e Ingegneria elettronica e telecomunicazioni per l'energia  
Laboratorio di Analisi Numerica  
A.A. 2019/2020 – I Ciclo**

## Esercitazione 3

Creare una cartella <cognome> in C: dove verranno salvati i file creati nella sessione di lavoro.  
Appena entrati in MATLAB posizionarsi in <cognome>.  
Risolvere in ambiente MATLAB i seguenti esercizi.

### INTERPOLAZIONE

1. Realizzare lo script **interpola\_dati.m** che calcoli il polinomio interpolante in forma di Newton (utilizzare **InterpN.m**, **HornerN.m**), e spline (built-in function **spline()**) di un set di dati **dataset1.dat** in [a,b]. Visualizzare in due grafici diversi in un'unica finestra: 1) l'interpolante polinomiale e i punti di interpolazione 2) l'interpolante spline e i punti di interpolazione.
2. Realizzare lo script **interpola\_funz.m** che calcoli il polinomio interpolante in [a,b] di grado n di un insieme di punti  $P_i=(x_i, y_i)$  con  $x_i$  a scelta dell'utente:

- punti  $x_i$  equidistanti (utilizzare  $x=\text{linspace}(a,b,n+1)$  );
- punti  $x_i$  definiti dagli zeri dei polinomi di Chebychev:

$$x_i = \frac{a+b}{2} - \frac{b-a}{2} \cos\left(\frac{2(i-1)+1}{2(n+1)} * \pi\right), \quad i = 1, \dots, n+1$$

e  $y_i$  ottenuti dalla campionamento (valutazione) nei punti  $x_i$  della funzione  $y=\sin(x)-2\sin(2x)$ .

Si utilizzi sia il metodo di Newton (utilizzare **InterpN.m**, **HornerN.m**), sia l'interpolazione a tratti con spline cubiche (built-in function **spline()**). Lo script visualizza in uno stesso grafico la funzione test da interpolare, i punti di interpolazione e i due polinomi interpolanti.

Modificare lo script affinché consideri la funzione test da interpolare  $y=1/(1+x^2)$ ,  $x \in [-5,5]$  (funzione di Runge). Verificare cosa succede al variare del grado n.

Calcolare e stampare il max. dell'errore di interpolazione in valore assoluto nei due casi:

$$\max_{x \in [a,b]} |f(x) - p(x)|$$

3. La temperatura T in prossimità del suolo varia al variare della concentrazione k dell'acido carbonico e della latitudine L. Per k=1.5 la temperatura al suolo subisce una variazione dipendente dalla temperatura secondo la seguente tabella

L	-55	-45	-35	-25	-15	-5	5	15	25	35	45	55	65
T	3.7	3.7	3.52	3.27	3.2	3.15	3.15	3.25	3.47	3.52	3.65	3.67	3.52

Si vuole costruire un MODELLO che descriva la legge  $T=T(L)$  anche per latitudini non misurate.

Ad esempio si vuole valutare la variazione di temperatura a Roma ( $L=42^\circ$ ).

Sperimentare nello script *test2.m* le seguenti tecniche:

- Interpolazione con un polinomio di grado 12;
- Interpolazione con spline cubiche.

4. Vogliamo determinare una curva che segua il contorno di un dettaglio contenuto in un'immagine. Consideriamo per esempio la figura contenuta nell'immagine MRI **tumor.jpg**. Vogliamo riprodurre il contorno del tumore cerebrale in essa contenuto utilizzando o due polinomi di interpolazione (uno per la parte superiore ed uno per quella inferiore), o un polinomio lineare a tratti o un polinomio cubico a tratti (spline cubiche).

1. Selezionare un insieme di punti (x,y) sul contorno della struttura.

2. Utilizzare questi punti per produrre tre interpolanti:

- a) un polinomio interpolante della parte sopra e uno per la parte sotto,
- b) un polinomio lineare a tratti (semplice uso di *plot*), e
- c) una curva spline cubica con parametrizzazione uniforme (uso di *spline()* per l'interpolazione delle x e delle y).

NOTA: il polinomio interpolante è una funzione: per ogni valore distinto di x c'è solo un valore di p(x). Il contorno da estrarre non è una funzione, così sarà necessario mettere insieme più di un polinomio per ricostruire l'intera silhouette o considerare un interpolante per curve. Si può per esempio considerare un polinomio per la parte superiore ed uno per quella inferiore dell'elefante.

d) Valuta la bontà dell'interpolazione eseguita.

e) Rappresentare in una finestra i tre interpolanti trovati e i dati da interpolare.

=====

Inizializzare il file con i seguenti comandi:

```
%Load the picture
[I] = imread('tumor.jpg');
imagesc(I)
colormap(gray)

% Pick the interpolation points for the upper profile until the
%   return key is pressed..
disp('Pick interpolation points for the upper profile')
[xup,yup] = ginput
% Note that the x-coordinates must be in ascending order
nup = length(xup);
yup = - yup;
%Because of the way imagesc displays pictures, the y coordinates need to be
negated to make our interpolated right-side up.

% Pick the interpolation points for the lower profile, but reuse the
% end points from the upper.
disp('Pick interpolation points for the lower profile.')
disp('The two endpoints from the upper will be reused.')
[xlo,ylo] = ginput;
xlo = [xup(1);xlo;xup(nup)];
ylo = [yup(1);-ylo;yup(nup)];
nlo = length(xlo);

disp('Interpolation points for the upper profile:')
```

```
[xup,yup]
```

```
disp('Interpolation points for the lower profile:')  
[xlo,ylo]
```

**5. Simulazione del percorso di un braccio meccanico di un robot controllato mediante curve spline.** Realizzare uno script **robot.m** che, dati in input una sequenza di punti di coordinate (x,y) che rappresentano punti lungo una traiettoria sulla quale la mano del robot deve muoversi, calcoli la spline cubica di interpolazione delle coppie dei corrispondenti angoli (spline con derivate agli estremi nulle, poichè si suppone che il robot parta e arrivi con velocità zero). Il programma salvi in output il file dati **motori.dat** contenente la sequenza degli opportuni valori degli angoli (da fornire ai due motori che pilotano i joint).

Il programma deve permettere l'inserimento dei punti iniziale, intermedi e finale e quindi visualizzare lo spostamento lungo la traiettoria calcolata.

Supponendo le lunghezze dei bracci  $L_1=L_2=0.5$ , sfruttare le seguenti relazioni tra posizioni (x,y) ed angoli:

$$R^2 = x^2 + y^2; \quad \cos(\theta_2) = \frac{R^2 - L_1^2 - L_2^2}{2L_1L_2}; \quad \cos(\beta) = \frac{R^2 + L_1^2 - L_2^2}{2L_1R}; \quad \alpha = \arctan \frac{y}{x}$$

$$\theta_1 = \begin{cases} \alpha + \beta & \text{se } \theta_2 < 0 \\ \alpha - \beta & \text{se } \theta_2 \geq 0 \end{cases}$$