Laboratorio Calcolo Numerico

Esercizio 1

Considerata la funzione di Runge
$$f(x) = \frac{1}{1+x^2}$$
nell'intervallo $I = [-5, 5]$, si vuole costruire il polinomio interpolatore $P_{10} \in \mathcal{P}_{10}$ utilizzando i nodi di Chebyshev del primo tipo.

Si modifichi opportunamente lo script `interpola` del laboratorio precedente in modo che si possa in alternativa scegliere una discretizzazione uniforme nell'intervallo di interpolazione oppure la scelta dei nodi di Chebyshev. Lo script modificato deve anch'esso scrivere i risultati su file e creare le due figure (polinomio + funzione e la seconda la funzione errore) che dovranno avere nel titolo anche la specifica del tipo di nodi utilizzati. La function `metodop.m` non necessita cambiamenti e neppure la funzione scelta per l'implementazione, tenendo però presente che con i nodi di Chebyshev non possono essere utilizzate le differenze finite.

Lo script, avrà, nel caso dei nodi di Chebyshev, la necessità di calcolare tali nodi nell'intervallo prescelto. Pertanto si dovrà scrivere una function `chzeros1` che implementi (utilizzando ovviamente la vettorizzazione del Matlab) il seguente algoritmo per il calcolo degli zeri di Chebyshev in un intervallo $[a, b]$ (che saranno i nuovi nodi). $n1$ rappresenta il numero di nodi desiderato, ovvero il grado del polinomio di Chebyshev.

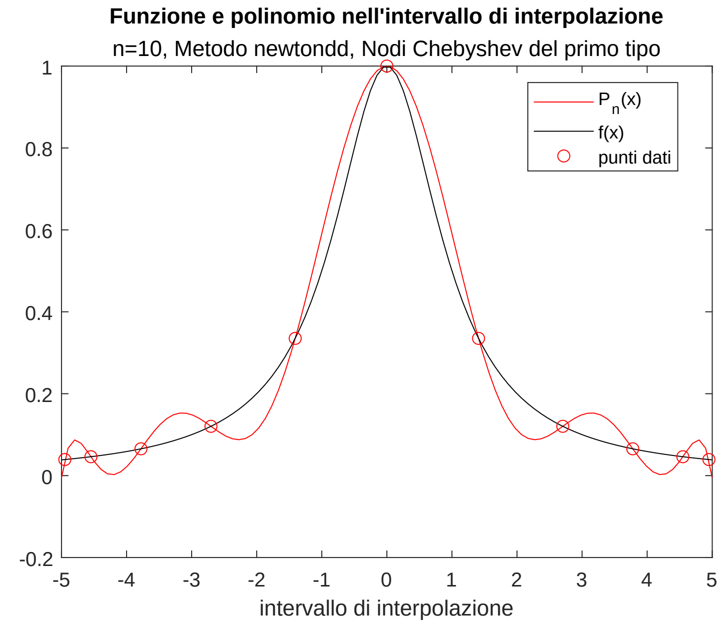
$$[\mathbf{z}] = \textbf{Zeri\_Chebyshev\_1}\ (a, b, n1)$$

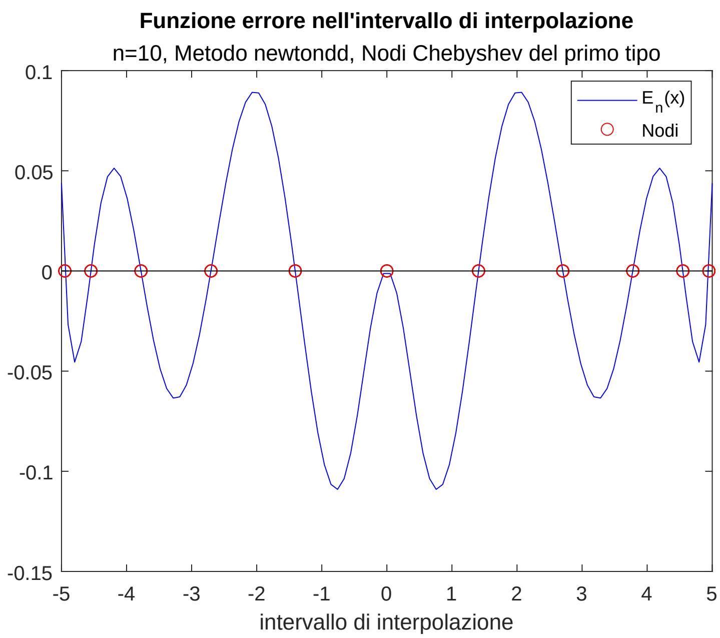
for $i = 1, \ldots, n1$
  $\beta = (2 \times i - 1)/(2 \times n1)$
  $z_i = -\cos(\beta \times \pi)$
end for i
if $n1$ dispari **then**
  $z_{(n1+1)/2} = 0$
end if
Mappa nell'intervallo $[a, b]$ gli zeri
for $i = 1, \ldots, n1$
  $z_i = (b - a)/2 \times z_i + (b + a)/2$
end for i

L'intestazione deve essere

```
function z = chzeros1 (a,b,n)
%CHZEROS1  Funzione che calcola nell'intervallo [a,b] gli
%          zeri dei polinomi di Chebyshev del primo tipo
% Uso:
% z = chzeros1(a,b,n)
%
% Dati di ingresso:
%   a:     Estremo inferiore dell'intervallo
%   b:     Estremo superiore dell'intervallo
%   n:     Grado del polinomio di Chebyshev e numero degli zeri
%
% Dati di uscita:
%   z:     Vettore colonna contenente gli zero ordinati in modo
%          crescente
```

Per testare lo script modificato e la function `chzeros1` si può inizialmente utilizzare il metodo `'matlab'`. Le figure che devono essere ottenute sono:





Esercizio 1 bis (facoltativo)

Si inserisca nello script anche l'opzione che permette di utilizzare gli zeri dei polinomi di Chebyshev del secondo tipo (Chebyshev-Lobatto) creando una funzione di nome `chzeros2.m`.

Esercizio 2

Si vogliono approssimare con il **criterio dei minimi quadrati**, ed un *polinomio al più di primo grado*

$$P_1(x) = a_1 + a_2 x$$

alcuni dati (pesati) memorizzati in un file di nome `dati.dat`, ovvero si vuole calcolare la **retta di regressione** (si noti che, per agevolare il trattamento degli indice del Matlab, i coefficienti non partono da a_0 bensì da a_1). In questo caso il sistema delle *equazioni normali*

$$\underbrace{A^T A}_{M}\,\mathbf{a} = \underbrace{A^T \mathbf{b}}_{\mathbf{z}},$$

con M matrice simmetrica e definita positiva (dimensione 2×2), ovvero il sistema $M\mathbf{a} = \mathbf{z}$, che deve essere risolto per trovare il vettore $\mathbf{a} = (a_1, a_2)$ dei coefficienti, può essere scritto direttamente ed è uguale a

$$\begin{pmatrix} \sum_{i=1}^{m+1} w_i & \sum_{i=1}^{m+1} w_i\, x_i \\ \sum_{i=1}^{m+1} w_i\, x_i & \sum_{i=1}^{m+1} w_i\, x_i^2 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{m+1} w_i\, f(x_i) \\ \sum_{i=1}^{m+1} w_i\, x_i\, f(x_i) \end{pmatrix}.$$

Il file di nome `dati.dat` (fornito dai docenti) contiene una tabulazione a 3 colonne e 16 righe, contenente i valori x_i, $y_i = f(x_i)$ e w_i, $i = 1, \ldots, 16$:

	x_i	y_i	w_i
x_1	-3.490	27.200	0.5
x_2	-2.948	4.720	1
x_3	-2.574	-0.978	20
x_4	-2.157	4.100	1
x_5	-1.377	16.013	0.1
x_6	-1.234	19.656	16
x_7	-0.861	22.498	1
x_8	-0.116	21.650	1
x_9	0.235	16.770	1
x_{10}	0.558	12.671	1
x_{11}	1.036	4.042	16
x_{12}	1.318	-2.158	1
x_{13}	2.139	-16.901	16
x_{14}	2.566	-11.437	1
x_{15}	2.736	-13.449	1
x_{16}	3.312	31.184	20

e dovrà essere letto (istruzioni da inserire nello script), con i comandi illustrati nelle slide, memorizzando poi le colonne, separatamente, in tre vettori colonna di nome `xv, fxv` e `wv`;

- Si crei una function di nome `regre.m` che, avuti in ingresso la tabulazione dei punti (x_1 e $f(x_i)$) ed i pesi w_i, memorizzati in 3 vettori colonna distinti di nome `x, y` e `w`, restituisca i coefficienti della soluzione `a` (vettore colonna di due componenti).
  Tale funzione deve controllare che le lunghezze dei tre vettori dati siano le stesse (altrimenti si esca con un errore).
  L'algoritmo, e quindi la funzione, dovranno preventivamente costruire il sistema ovvero creare la matrice `M` ed i vettore `z` (si usa la vettorizzazione Matlab) e poi risolvere il sistema, restituendo la soluzione (comando `a = M \ z;`. La function avrà quindi la seguente intestazione

```
function a = regre (x,y,w)
% REGRE calcola i coefficienti della retta di regressione,
%       soluzione del sistema ai minimi quadrati, a partire
%       da tabulazione di dati (punti e pesi).
%
% Uso:
%   a = regre (x,y,w);
%
% Dati di ingresso:
%   x    vettore colonna con ascisse dei punti
%   y    vettore colonna con ordinate dei punti
%   w    vettore colonna con i pesi
%
% Dati di uscita:
%   a    vettore colonna che contiene i due coefficienti della
%        retta di regressione (a_1 e a_2).
```

- Si scriva uno script di nome `regrescript.m` che
  - legga da file esterno, chiamato `dati.dat`, i dati della tabulazione disposti in 3 colonne (comando `load` che memorizza una matrice);
  - assegni le colonne di tale matrice, separatamente, a tre vettori colonna di nome `xv, fxv` e `wv`;
  - visualizzi il numero di punti dati;
  - calcoli e visualizzi l'intervallo di approssimazione $[\min x_i, \max x_i]$;
  - chiami la function `regre` per ottenere la soluzione nel vettore `av`;
  - calcoli e visualizzi l'errore quadratico (trattandosi di una retta, può essere calcolato direttamente, utilizzando la vettorizzazione; in alternativa si può utilizzare opportunamente il comando `polyval` visto nei laboratori precedenti, riordinando i coefficienti del polinomio);
  - calcoli e visualizzi il baricentro (`xbari,ybari`) dei punti della tabulazione;
  - disegni sulla stessa figura, nell'intervallo di interpolazione $I = [\min x_i, \max x_i]$, i 16 punti di coordinate (x_i, y_i) (rappresentati con un `*`), la retta di regressione (disegnata con soli DUE punti) ed il baricentro (rappresentato con il carattere `o`), inserendo titolo, scritte sugli assi cartesiani, cognome e nome.

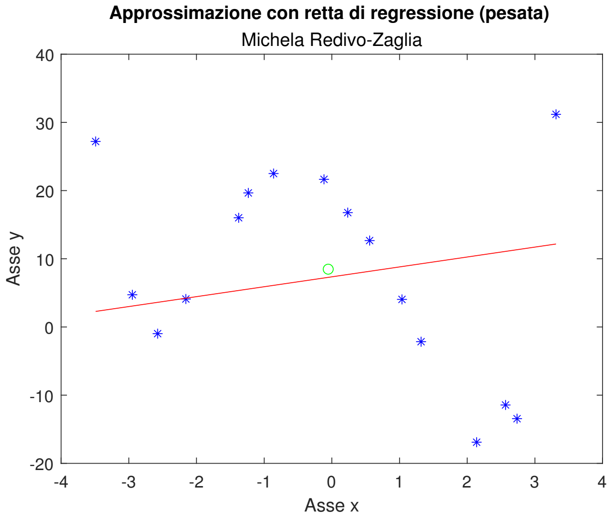
La visualizzazione dati e risultati dovrebbe essere

```
Numero punti = 16
Intervallo di approssimazione [-3.49,3.312]

L'errore quadratico e`
   2.5342e+04

Baricentro
   -0.0536    8.4738
```

e una figura simile a


Approssimazione con retta di regressione (pesata)
Michela Redivo-Zaglia
Asse y
Asse x
40
30
20
10
0
-10
-20
-4
-3
-2
-1
0
1
2
3
4