

Laboratorio Calcolo Numerico

Esercizio 1

Si vogliono approssimare i seguenti dati sperimentali (con pesi tutti unitari)

n	x_n	$f(x_n)$
1	-3.490	27.200
2	-2.948	4.720
3	-2.574	-0.978
4	-2.157	4.100
5	-1.377	16.013
6	-1.234	19.656
7	-0.861	22.498
8	-0.116	21.650
9	0.235	16.770
10	0.558	12.671
11	1.036	4.042
12	1.318	-2.158
13	2.139	-16.901
14	2.566	-11.437
15	2.736	-13.449
16	3.312	31.184

con il **criterio dei minimi quadrati**, con un *polinomio al più di quarto grado*

$$P_4(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0.$$

Per trovare il vettore $\mathbf{a} = (a_0, \dots, a_4)$ dei coefficienti, si devono creare la matrice

$$A = \begin{pmatrix} 1 & x_1 & x_1^2 & x_1^3 & x_1^4 \\ 1 & x_2 & x_2^2 & x_2^3 & x_2^4 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{16} & x_{16}^2 & x_{16}^3 & x_{16}^4 \end{pmatrix},$$

il vettore $\mathbf{b} = (f(x_1), \dots, f(x_{16}))^T$, e si deve risolvere **nel senso dei minimi quadrati** il *sistema rettangolare*

$$A\mathbf{a} = \mathbf{b}.$$

Attenzione: Per costruire A si può utilizzare il comando

```
A = [ones(m1,1) x x.^2 x.^3 x.^4 ]
```

dove $m1$ rappresenta il numero di punti $m + 1$. A è una matrice particolare di Vandermonde (ne parleremo in seguito). Esistono comandi Matlab per costruirla facilmente qualsiasi sia il numero di colonne desiderato (comandi **vander** e **fliplr**. Solo per i più esperti in Matlab!)

Per trovare la soluzione si risolve il sistema delle *equazioni normali*

$$\underbrace{A^T A}_M \mathbf{a} = \underbrace{A^T \mathbf{b}}_{\mathbf{z}}.$$

Si può dimostrare che la matrice M è simmetrica e definita positiva, quindi il sistema $M\mathbf{a} = \mathbf{z}$ potrebbe essere risolto tramite la fattorizzazione di Cholesky (ma lo vedremo in seguito).

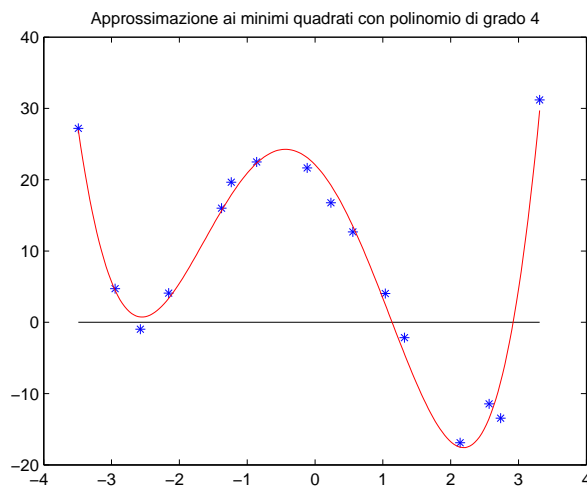
Si costruisca uno script `minquadscript.m` che

- Definisca le 16 coppie di dati, memorizzandole nei due vettori **colonna** **x** e **b**;
- costruisca la matrice **A**
- costruisca la matrice **M** ed il vettore **z**;
- risolva il sistema $M\mathbf{a} = \mathbf{z}$ con il comando

`a = M\z;`

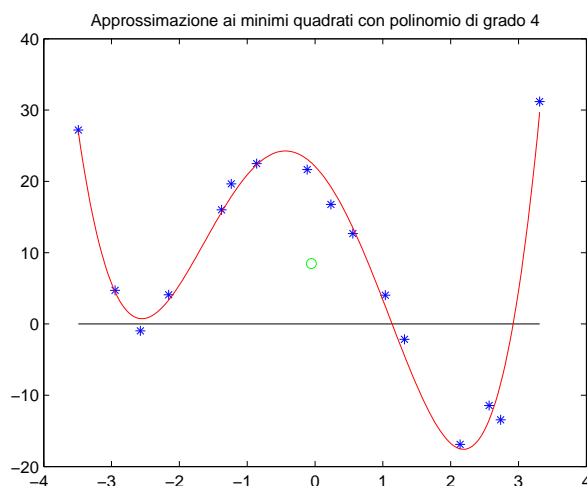
- disegni su di un grafico l'asse x, i 16 punti di coordinate $(x_i, f(x_i))$ ed il polinomio approssimante ai minimi quadrati, valutandolo su 200 punti dell'intervallo di interpolazione $I = [\min x_i; \max x_i]$ (si usino i comandi Matlab `min` e `max`). Per disegnare il polinomio si usino i comandi `linspace` e `polyval` (si ricordi che tale comando prevede in ingresso di avere i coefficienti del polinomio in ordine decrescente delle potenze di x , mentre il vettore **a**, soluzione del sistema, li contiene in ordine crescente; si usi quindi anche il comando `flipud`).

Si dovrebbe ottenere una figura simile a



Esercizio 2

Si calcoli e si rappresenti nella stessa figura anche il baricentro dei punti con un circoletto verde. Si dovrebbe ottenere una figura simile a



Esercizio 3 (facoltativo)

Si generalizzi la soluzione precedente creando una function che, avuti in ingresso la tabulazione dei punti, i pesi ed il grado n del polinomio, restituisca i coefficienti della soluzione \mathbf{a} del sistema ai minimi quadrati. Attenzione che la costruzione di A e di \mathbf{b} deve essere modificata per tener conto dei pesi. Coloro che hanno usato i comandi `vander` e `fliplr`, devono anche usare il comando `repmat` (solo per i più esperti in Matlab!)

Si provi tale function scrivendo uno script che utilizzi la stessa tabulazione dell'Esercizio 1, con pesi unitari, ma con un polinomio di grado al più 5. Si confronti il vettore soluzione \mathbf{a} della propria function con quello ottenuto con il comando `polyfit` e si producano con `subplot` due figure che contengano le due curve separate (quella ottenuta con la propria function e quella ottenuta con il comando Matlab).

Ovviamente l'aspetto delle curve deve essere uguale! La figura risultante dovrà essere simile alla seguente.

