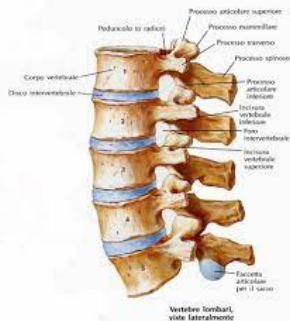


Approssimazione nel senso dei minimi quadrati

Problema 1.

Nella Tabella vengono riportati i risultati di un esperimento eseguito per individuare il legame fra lo sforzo e la relativa deformazione di un campione di tessuto biologico (un disco intervertebrale). Partendo dai dati riportati in tabella si vuole stimare la deformazione corrispondente ad uno sforzo $\sigma = 0.2$ MPa ($\text{MPa} = 100 \text{ N/cm}^2$) utilizzando un'approssimazione nel senso dei minimi quadrati con un polinomio di grado 1 (retta di regressione).



| Test | Sforzo σ | Deformazione ϵ |
|------|-----------------|-------------------------|
| 1 | 0.00 | 0.00 |
| 2 | 0.04 | 0.03 |
| 3 | 0.06 | 0.08 |
| 4 | 0.10 | 0.08 |
| 5 | 0.14 | 0.14 |
| 6 | 0.25 | 0.20 |
| 7 | 0.31 | 0.23 |

Risoluzione del problema

Cerco un polinomio di grado 1 che approssima i dati nel senso dei minimi quadrati: $p_1(x) = a_1x + a_2$, dove $x = \sigma$.
2 incognite, $n = 7$ equazioni.

Il comando `a=polyfit(x,y,m)` calcola il vettore `a` dei coefficienti del polinomio di grado m che approssima i dati x_i, y_i nel senso dei minimi quadrati (al suo interno costruisce il sistema delle equazioni normali, fa la fattorizzazione QR, risolve il sistema triangolare).

```
x=[ ]; % (colonna) conterra' i valori di sigma  
y=[ ]; % (colonna) conterra' i valori di epsilon  
a=polyfit(x,y,1); % 1= grado del polinomio da cercare
```

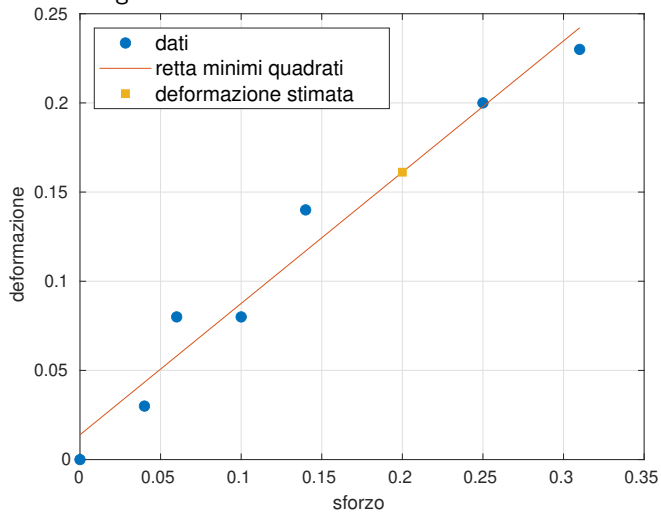
Troviamo `a=[0.7361 0.0139]`

Una volta trovato a , rappresentare i punti sul piano cartesiano e disegnare la retta calcolata. Quindi valutare la retta quando $x = 0.02$.

```
figure(1); clf
p1=plot(x,y,'o');
p1.MarkerFaceColor=p1.Color; % per colorare i marker
% al loro interno
hold on
yr=polyval(a,x); % valuto il polinomio (i cui
% coefficienti sono salvati nel vettore a)
% nei punti di ascissa x
plot(x,yr,'r','Linewidth',2);
xs=0.02;
ys=polyval(a,xs);
p2=plot(xs,ys,'s');
p2.MarkerFaceColor=p2.Color;
```

Quando $\sigma = 0.2$ troviamo sforzo $\epsilon \simeq p_1(0.2) = 0.16115$.

Risultato grafico:



Problema 2

Nella seguente tabella sono riportati i tempi di CPU (in sec) misurati durante l'esecuzione di un programma che ordina un vettore di lunghezza L .

| | | | | | |
|----------------------------|-------|-------|-------|-------|-------|
| $x = L$ (dim. del vettore) | 10000 | 20000 | 30000 | 40000 | 50000 |
| $y = \text{CPUtime}$ (sec) | 0.31 | 0.95 | 2.45 | 4.10 | 6.46 |

Determinare la parabola che meglio approssima i dati nel senso dei minimi quadrati e stimare quanto tempo di CPU serve per ordinare un vettore di 55218 elementi. Utilizzare prima la function `polyfit` di MATLAB e poi la fattorizzazione QR sul sistema delle equazioni normali.

Svolgimento

Devo calcolare i coefficienti del polinomio $p_2(x) = a_1x^2 + a_2x + a_3$ che approssima i dati nel senso dei minimi quadrati.

Con il comando `polyfit`.

```
x=[ ]; % dimensioni dei vettori
y=[ ]; % tempi di CPU
a=polyfit(x,y,2); % 2= grado del polinomio da cercare
```

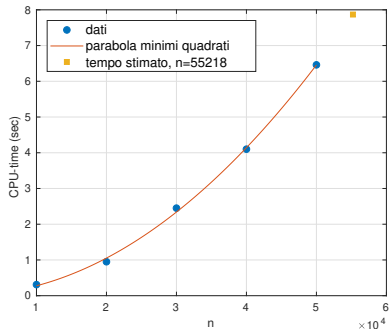
Per disegnare:

```
% disegno i punti
figure(1); clf
plot(x,y,'o','Displayname','dati');
legend('-dynamiclegend')
hold on; grid on;
% valuto la parabola in piu' punti tra x(1) e x(end)
x1=linspace(x(1),x(end),100)';
y1=polyval(a,x1); % valuto p_2(x)
plot(x1,y1,'Linewidth',2,...
      'Displayname','parabola minimi quadrati');
xlabel('L'); ylabel('CPUtime')
```

Per stimare il tempo di CPU necessario per ordinare un vettore di $L = 55218$ elementi:

```
Ls=55218;  
ys=polyval(a,Ls)  
plot(Ls,ys,'sy','Displayname','valore stimato')
```

Quindi la risposta finale sarà $t = 7.8681\text{sec}$



Risoluzione delle equazioni normali con fatt. QR

Devo calcolare i coefficienti del polinomio $p_2(x) = a_1x^2 + a_2x + a_3$ che approssima i dati nel senso dei minimi quadrati.

Il sistema delle equazioni normali $X^T X \mathbf{a} = X^T \mathbf{y}$ ha matrice

$$X = \begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ \vdots & \vdots & \vdots \\ x_n^2 & x_n & 1 \end{bmatrix}$$

Le istruzioni sono:

```
x=[ ]; y=[ ]; % vettori colonna
X=[x.^2, x, ones(n,1)];
[Qtilde,Rtilde]=qr(X,"econ");
% risolvo \tilde{R}=(\tilde{Q})^T y
a=Rtilde\'*(Qtilde*y)
```

Troviamo $\mathbf{a}=[2.5643\text{e-}09 \ 6.4286\text{e-}07 \ 1.4000\text{e-}02]$, come con `polyfit`.

Base di monomi e base di sin/cos

Problema 3. Nella tabella alla pagina seguente sono riportate le misurazioni di concentrazione di monossido di azoto (NO) nel periodo di 24h rilevate in una strada trafficata di una grande città. Poiché la concentrazione di NO è principalmente dovuta alle auto, ha valori massimi al mattino e al pomeriggio, quando il traffico è più intenso. Per un'ulteriore analisi dell'inquinamento atmosferico abbiamo bisogno di costruire una funzione che approssimi i dati nel senso dei minimi quadrati, in modo da poter calcolare la concentrazione in un tempo arbitrario tra 0 e 24 ore.

Determinare le funzioni

$$g_1(t) = \sum_{j=0}^8 a_j t^j$$

$$g_2(t) = a_0 + \sum_{j=1}^4 a_j \sin(\omega_j t) + \sum_{j=1}^4 a_{4+j} \cos(\omega_j t)$$

dove t è il tempo, mentre $\omega = \frac{2\pi}{24}$. Confrontare quindi i risultati.

Dati del problema

| t_i | y_i | t_i | y_i |
|-------|--------|-------|--------|
| 1 | 73.72 | 13 | 239.48 |
| 2 | 23.39 | 14 | 236.52 |
| 3 | 17.11 | 15 | 245.04 |
| 4 | 20.31 | 16 | 286.74 |
| 5 | 29.37 | 17 | 304.78 |
| 6 | 74.74 | 18 | 288.76 |
| 7 | 117.02 | 19 | 247.11 |
| 8 | 298.04 | 20 | 216.73 |
| 9 | 348.13 | 21 | 185.78 |
| 10 | 294.75 | 22 | 171.19 |
| 11 | 253.78 | 23 | 171.73 |
| 12 | 250.48 | 24 | 164.05 |

y_i = misurazione della concentrazione di monossido di azoto (NO) (in $\mu\text{g}/\text{m}^3$) al tempo t_i (in ore).

Il file `monossido.mat` contiene i dati del problema. Download dalla pagina del corso: <https://paola-gervasio.unibs.it/CS/MATLAB/monossido.mat>

Osserviamo che g_1 è espressa rispetto alla base dei monomi, g_2 rispetto ad una base di seni e coseni.

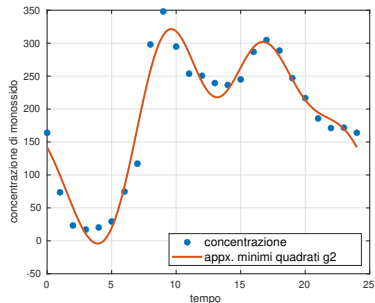
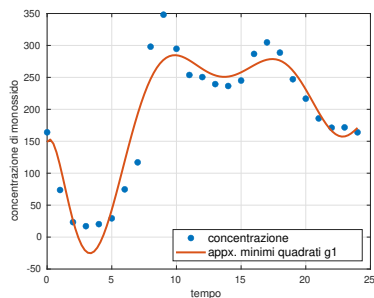
Nel primo caso possiamo richiamare la function `polyfit`:

```
a1=polyfit(t,y,8);
```

Nel secondo caso invece non possiamo usare `polyfit` perché la base non è polinomiale. Dobbiamo costruire la matrice X ed utilizzare la fattorizzazione QR.

$$X = \begin{bmatrix} 1 & \sin(\omega t_1) & \cos(\omega t_1) & \dots & \sin(4\omega t_1) & \cos(4\omega t_1) \\ 1 & \sin(\omega t_2) & \cos(\omega t_2) & \dots & \sin(4\omega t_2) & \cos(4\omega t_2) \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & \sin(\omega t_{24}) & \cos(\omega t_{24}) & \dots & \sin(4\omega t_{24}) & \cos(4\omega t_{24}) \end{bmatrix}$$

I risultati



Un indicatore della bontà dell'approssimazione è il residuo pesato sulla media dei dati:

$$r_k = \frac{\|\mathbf{y} - g_k(\mathbf{t}, \mathbf{a})\|_2}{\text{mean}(\mathbf{y})}$$

Si ottiene nei due casi $r_1 = 1.06318$, $r_2 = 0.59241$.