



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA



DIPARTIMENTO  
DI INGEGNERIA  
INDUSTRIALE



DIPARTIMENTO  
**MATEMATICA**  
DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA"

## Laboratorio di Calcolo Numerico LAB 7

### Approssimazione ai minimi quadrati

Docenti: E. Bachini, L. Bruni

Email: [elena.bachini@unipd.it](mailto:elena.bachini@unipd.it) Email: [bruni@math.unipd.it](mailto:bruni@math.unipd.it)

17 aprile 2024

# Outline

- 1 Richiami comandi Interpolazione/Approssimazione
- 2 Esercizi

# Interpolazione/Aprossimazione polinomiale

## Input del problema:

$$x = (x_0, \dots, x_n), \quad y = (y_0, \dots, y_n)$$

## Problema:

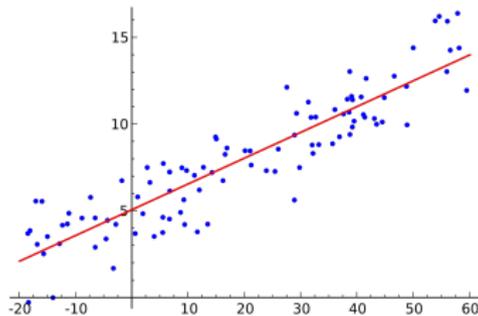
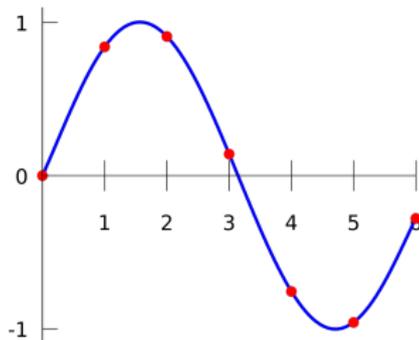
Trovare il polinomio  $P(x) = \sum_{i=0}^m a_i x^i$  tale che valga:

- Caso interpolazione ( $m = n$ ):

$$P(x_i) = y_i \quad \forall i \in [0, \dots, n]$$

- Caso approssimazione ( $m < n$ ):

$$\min \sum_{i=0}^n |P(x_i) - y_i|^2$$



## MATLAB - functions `polyfit()`, `polyval()`

### Calcolo del polinomio interpolante/approssimante

- INPUT: vettori  $x = (x_0, \dots, x_n)$ ,  $y = (y_0, \dots, y_n)$  e grado  $m$
- MATLAB: `a = polyfit(x,y,m)`
- OUTPUT: vettore  $a = (a_0, \dots, a_m)$  dei coefficienti di  $P(x) = \sum_{i=0}^m a_i x^i$ , polinomio cercato

### Valutazione del polinomio

- INPUT: vettore  $a = (a_0, \dots, a_m)$  dei coefficienti, vettore  $z = (z_0, \dots, z_k)$  dei punti di valutazione (per una buona visualizzazione:  $k$  grande)
- MATLAB: `p = polyval(a,z)`
- OUTPUT: vettore  $p = (P(z_0), \dots, P(z_k))$ , dove  $P(x) = \sum_{i=0}^m a_i x^i$

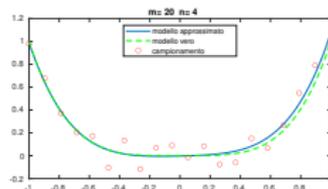
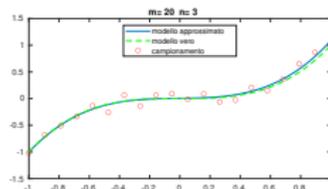
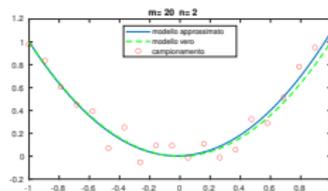
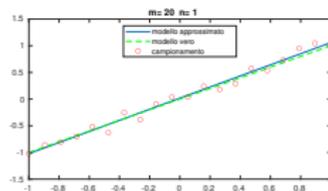
## Esempio di fitting

Proviamo a fittare dati derivanti da polinomi di grado 1, 2, 3, 4 ma campionati con un errore (grande!) di  $10^{-1}$ .

```
m = 20;
x = linspace(-1,1,m);
xplot = linspace(-1,1);
err = 10^-1*randn(1,m);
for n = 1:4
    y = x.^n+err;
    a = polyfit(x,y,n);
    yplot = polyval(a,xplot);
    subplot(2,2,n); plot(xplot,yplot); hold on
    plot(xplot,xplot.^n,'--g')
    plot(x,y,'or')
    title(['m= ' num2str(m) ' n= ' num2str(n)]);
    legend('modello approssimato','modello vero', ...
        'campionamento','Location','North')
end
```

Vediamo i risultati graficamente.

# l'errore di campionamento viene filtrato



## Esercizio 1 - Funzioni da approssimare

Considerare il problema di approssimazione per le seguenti funzioni, con  $n = 10, 20, 50$  e diversi valori di  $m < n$  ( $m = 1, 2, 3, 4, \dots$ ):

①  $f(x) = \cos(x - 2) - e^{1-x}$  con  $x \in [0, 8]$

vettori (punti sample):  $\hat{x} = \text{linspace}(0, 8, n + 1)$ ,  $\hat{y} = f(\hat{x})$

②  $f(x) = e^{-x}(1 + 0.3 \cos(4x^2))$  con  $x \in [0, 5]$

vettori:  $\hat{x} = \text{linspace}(0, 5, n + 1)$ ,  $\hat{y} = f(\hat{x})$

③  $f(x) = \frac{1}{x^2 + 1}$  con  $x \in [-5, 5]$  (funzione di Runge)

vettori:  $\hat{x} = \text{linspace}(-5, 5, n + 1)$ ,  $\hat{y} = f(\hat{x})$

## Esercizio 2 - Approssimazione di dati di input

- Leggere i dati importando il file `dati.txt` (vedi function `load()`)
- Creare i vettori con le coordinate  $x$  e  $y$  dalle colonne dei dati letti e salvarne la lunghezza ( $n + 1 =$  numero punti)
- Approssimare i dati e visualizzare il polinomio ottenuto utilizzando le funzioni `polyfit()` e `polyval()`, per diversi valori di  $m < n$ , per esempio  $m = 1, 2, 5$
- Confrontarli col polinomio interpolatore (cioè l'oggetto che si ottiene per  $m = n$ )

### MATLAB - function `load()`

- INPUT: file di testo (es. `dati.txt`) con valori numerici disposti in colonne
- SINTASSI: `mat = load('dati.txt');`
- OUTPUT: matrice `mat` contenente i dati del file

## Esercizi - consegna

Si crei un file .zip con i codici creati e con i relativi grafici. Si rinomini lo zip (**cognome\_nome\_lab7.zip**) e si carichi su moodle.