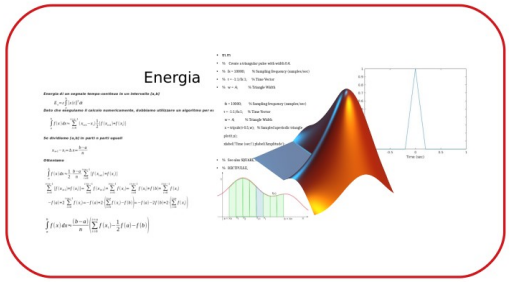
UNIVERSITÀ DEGLI STUDI DI TRIESTE

Programmazione Informatica

Ingegneria Civile e Ambientale
Ingegneria Navale

Programmazione - Matlab 4 – Sistemi Lineari, Algebra lineare, Analisi matematica



Antonio Fiumara
antonio.fiumara@dia.units.it

A. A. 2025-26

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Struttura del Corso

Gli argomenti trattati nel corso saranno:

- Lezione Teoria 1 – Introduzione, presentazione del corso, codifica dell'informazione
- Lezione Teoria 2 – Rappresentazione dell'informazione in un computer
- Lezione Teoria 3 – Algebra booleana, introduzione e applicazioni nella programmazione
- Lezione Teoria 4 – Architettura dei calcolatori. HW, SW, Sistemi Operativi
- Lezione Teoria 5 – Algoritmi e strutture dati - Implementazione di algoritmi notevoli (ordinamento , ricerca, ecc)
- Lezione Programmazione - Matlab 1 – Introduzione, installazione, comandi di base, matrici
- Lezione Programmazione - Matlab 2 – Input/Output (I/O), strutture di controllo
- Lezione Programmazione - Matlab 3 – Funzioni, ciclo for
- **Lezione Programmazione - Matlab 4 – Grafici. Sistemi Lineari, Algebra lineare, Analisi matematica**
- Lezione Programmazione - Matlab 5 – Assegnazione esercitazione di gruppo n.1. Strutture dati avanzate.
- Lezione Programmazione - Matlab 6 – Funzioni nel dominio del tempo e della frequenza. Analisi ed elaborazione dei segnali. Assegnazione esercitazione di gruppo n.2
- Lezione Programmazione - Matlab 7 – Correzione esercitazioni di gruppo. Applicazioni ingegneristiche avanzate
- Lezione Programmazione - Matlab 8 – Ripasso e approfondimenti
- Esonero esame Programmazione Matlab (3/12/25)

Introduzione

Grafici

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Esercizio

Scrivere uno script Matlab che:

- esegua il grafico della funzione $f(x)=4x^2-2x+1$ in un intervallo [a b] forniti in input in “runtime”
- Calcoli la derivata di f(x) e ne esegua il grafico

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Esercizio

Scrivere uno script Matlab che

- esegua il grafico della funzione $f(x)=c_0+c_1x+c_2x^2+...c_nx^n$ in un intervallo [a b].
- Calcoli la derivata di f(x) e ne esegua il grafico
- I coefficienti [c_0 c_1...c_n], [a b] e il numero di punti k sono forniti in input al "runtime"

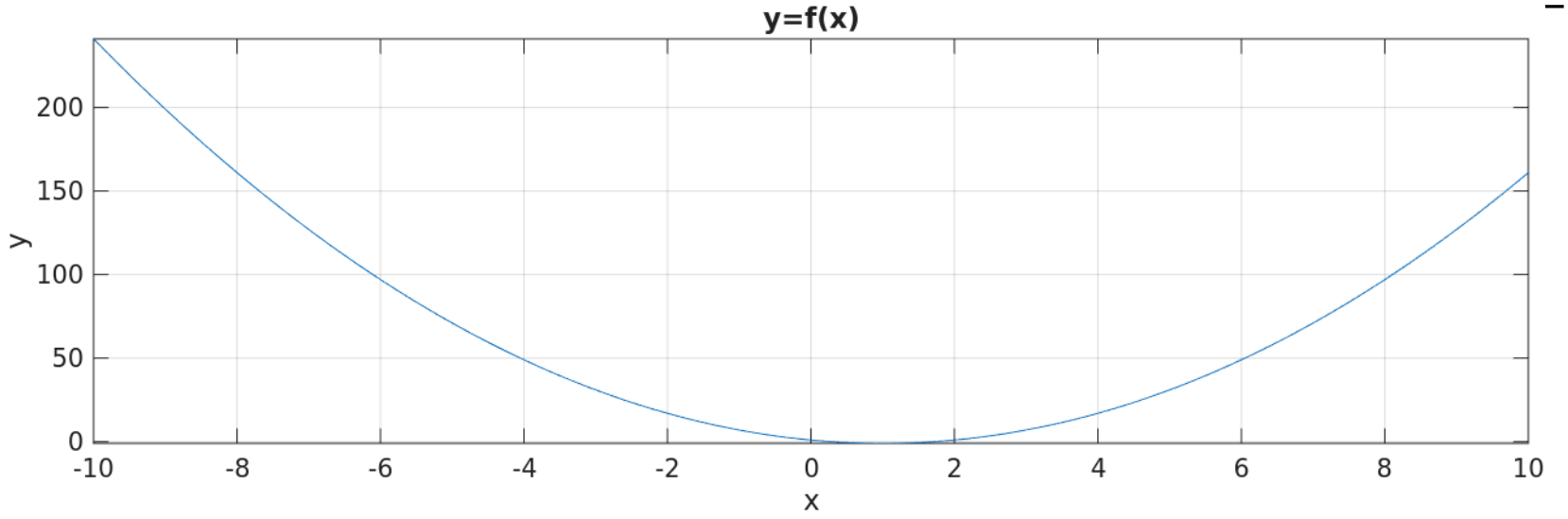
UNIVERSITÀ DEGLI STUDI DI TRIESTE

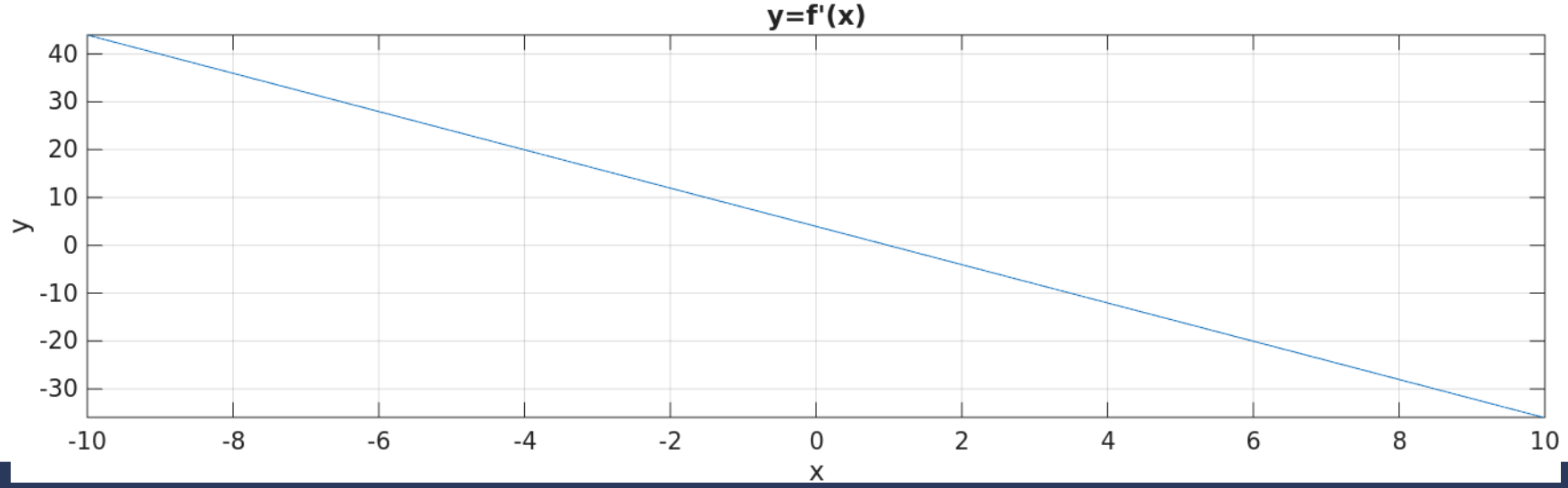
Esercizio

```
clear all
k=input ("insrisci l'intervallo [a b]");
a = k(1);
b = k(2);
coeff=input ('insrisci i coefficinti [a0 a1...an]');
n=length(coeff);
nPunti = input("inserisci numero di punti"); % Define the number of points for the interval
passo=(b-a)/nPunti;
x=a:passo:b;
y = zeros(size(x)); % Initialize y to store computed values
espo = 0; % Initialize exponent variable
for u=1:n
    y=y+coeff(u).*x.^(u-1);
end
figure(1)
plot(x,y)
axis([a b min(y) max(y)])
grid on
title ("y=f(x)")
figure (2)
dy=(y(1:end-1)-y(2:end))/passo;
plot(x(1:end-1),dy)
axis([a b min(dy) max(dy)])
grid on
title ("y=f'(x)")
```

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Esercizio





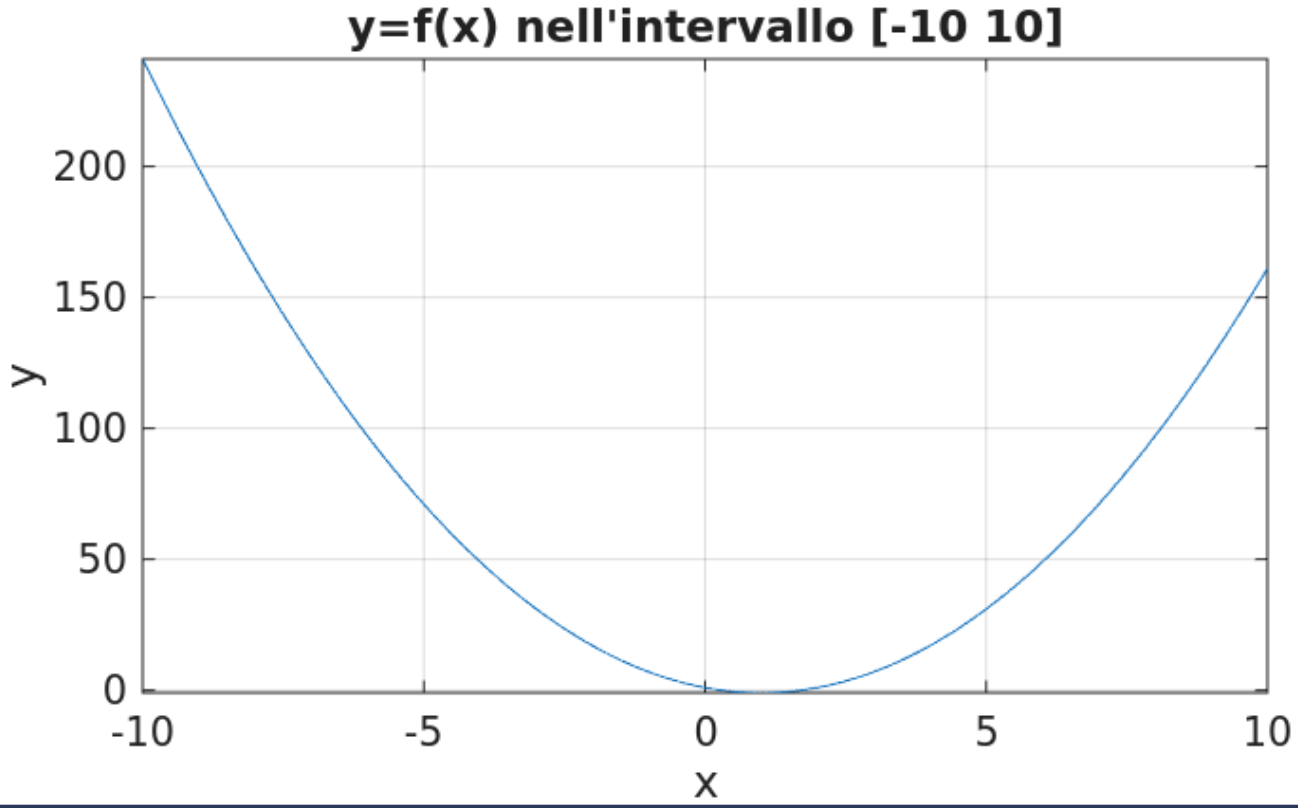
UNIVERSITÀ DEGLI STUDI DI TRIESTE

Esercizio

title (**sprintf**("y=f(x) nell'intervallo [**%d %d**]",a,b))

sprintf è analoga a fprintf() e restituisce in uscita una stringa formattata

>> help sprinf



UNIVERSITÀ DEGLI STUDI DI TRIESTE

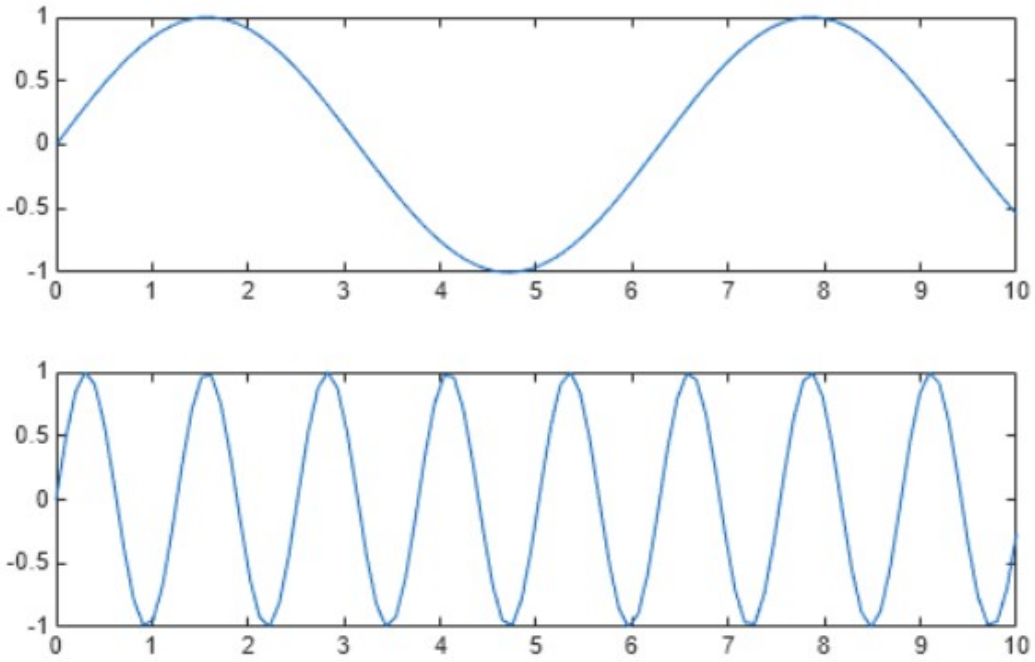
Subplot

subplot(m,n,p) divide la figura corrente in una griglia **m x n** e crea gli assi nella posizione specificata da **p**. MATLAB® numera le posizioni del sub-plot per riga. Il primo sub-plot è la prima colonna della prima riga, il secondo sub-plot è la seconda colonna della prima riga e così via. Se gli assi sono presenti nella posizione specificata, questo comando rende tali assi gli assi correnti.

```
>> help subplot

subplot(2,1,1);
x = linspace(0,10);
y1 = sin(x);
plot(x,y1)

subplot(2,1,2);
y2 = sin(5*x);
plot(x,y2)
```



UNIVERSITÀ DEGLI STUDI DI TRIESTE

Subplot

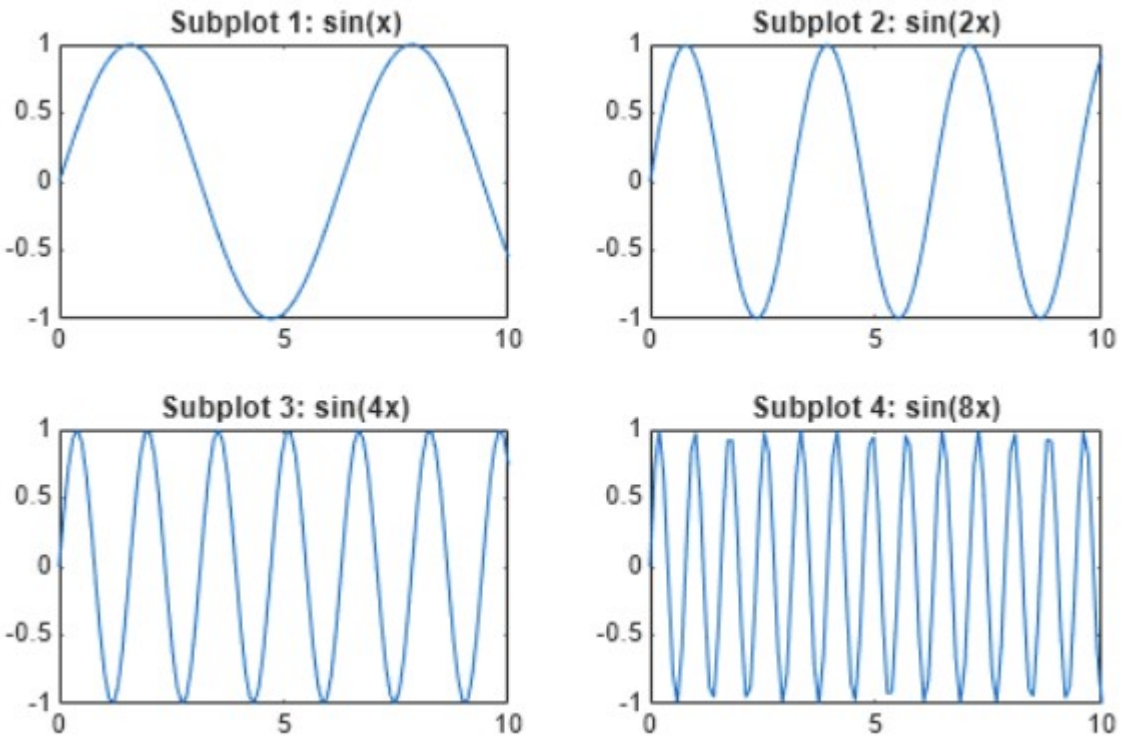
Esempio griglia 2x2

```
subplot(2,2,1)
x = linspace(0,10);
y1 = sin(x);
plot(x,y1)
title('Subplot 1: sin(x)')

subplot(2,2,2)
y2 = sin(2*x);
plot(x,y2)
title('Subplot 2: sin(2x)')

subplot(2,2,3)
y3 = sin(4*x);
plot(x,y3)
title('Subplot 3: sin(4x)')

subplot(2,2,4)
y4 = sin(8*x);
plot(x,y4)
title('Subplot 4: sin(8x)')
```



UNIVERSITÀ DEGLI STUDI DI TRIESTE

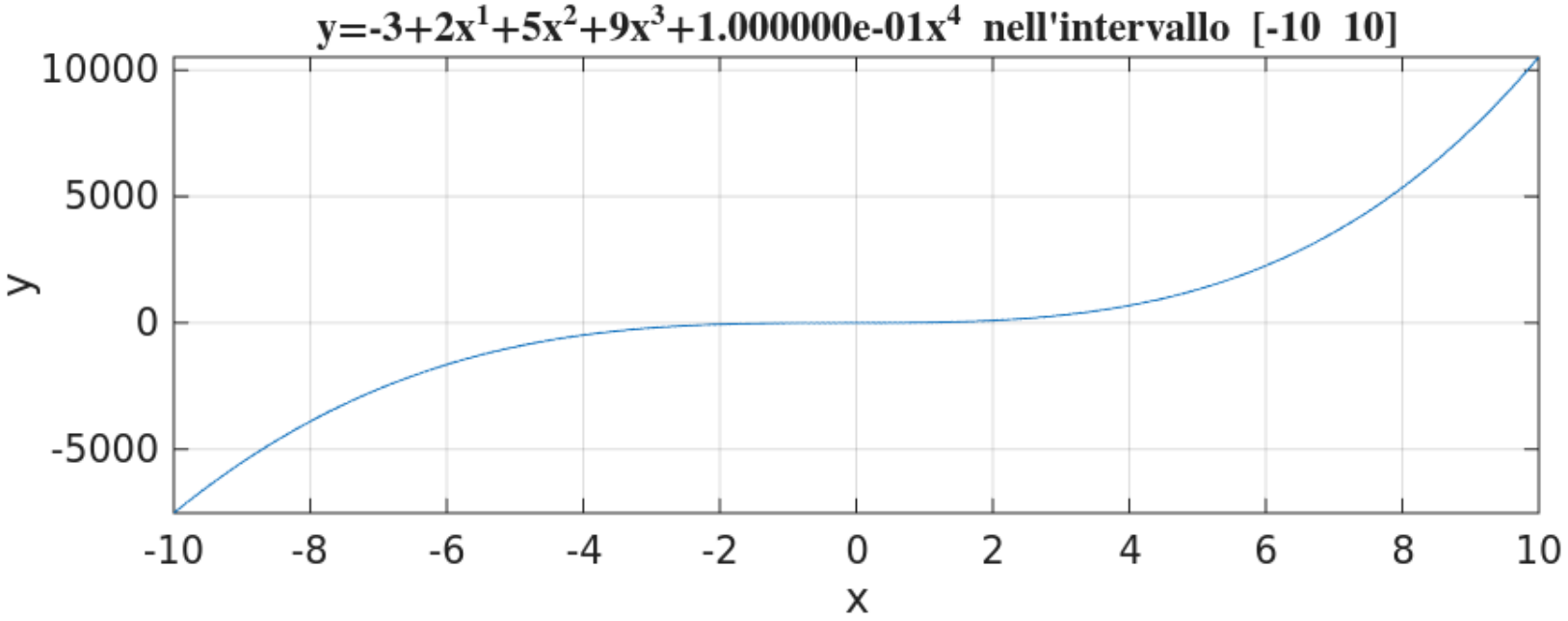
Esercizio

File: /home/arianna/Scaricati/graficoPolinomio3.m Pagina 1 di 1

```
clear all
k=input ("insrisci l'intervallo [a b]");
a = k(1);
b = k(2);
coeff=input ('insrisci i coefficinti [a0 a1...an]');
n=length(coeff);
nPunti = input("inserisci numero di punti"); % Define the number of points for the
interval
passo=(b-a)/nPunti;
x=a:passo:b;
y = zeros(size(x)); % Initialize y to store computed values
titolo=sprintf("y=%d",coeff(1))
for u=1:n
    y=y+coeff(u).*x.^(u-1);
    if (u>1)
        if (coeff(u)<0)
            titolo=titolo+'-'
         else
             titolo=titolo+'+'
         end
         titolo=titolo+sprintf("%dx^%d",abs(coeff(u)),u-1)
    end
end
figure(1)
plot(x,y)
axis([a b min(y) max(y)])
grid on
title (sprintf(titolo+" nell'intervallo [%d %d]",a,b))
xlabel("x")
ylabel("y")
figure (2)
dy=(y(1:end-1)-y(2:end))/passo;
plot(x(1:end-1),dy)
axis([a b min(dy) max(dy)])
grid on
title ("y=f'(x)")
xlabel("x")
ylabel("y")
```

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Esercizio



UNIVERSITÀ DEGLI STUDI DI TRIESTE

Derivata

Calcolo della derivata di y=f(x) nell'intervallo [a b]

$$\frac{df(x)}{dx} = \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x} \qquad \forall\, x \in [a\, b]$$

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Derivata

Calcolo della derivata di y=f(x) nell'intervallo [a b]

$$\frac{df(x)}{dx} = \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x} \qquad \forall x \in [a\, b]$$

$$\Delta x = x_{k+1} - x_k = passo \qquad (x = a : passo : b)$$

$$f(x+\Delta x) - f(x) = y_{k+1} - y_k$$

$$y_{k+1} - y_k = y(2:end) - y(1:end-1)$$

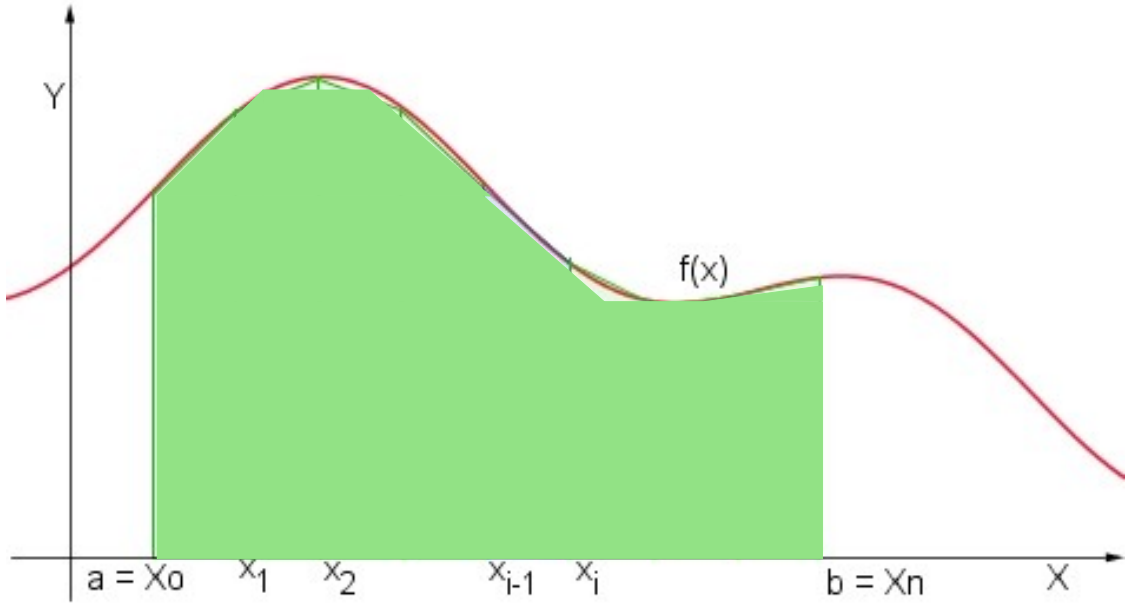
$$\frac{df(x)}{dx} = \frac{y(2:end) - y(1:end-1)}{passo}$$

Esiste anche una funzione di libreria **diff()**

Integrale definito in [a b]

Calcolo dell'integrale di y=f(x) nell'intervallo [a b]. Dobbiamo calcolare l'area sottesa alla curva tra x=a e x=b.

$$\int_a^b f(x)\,dx$$



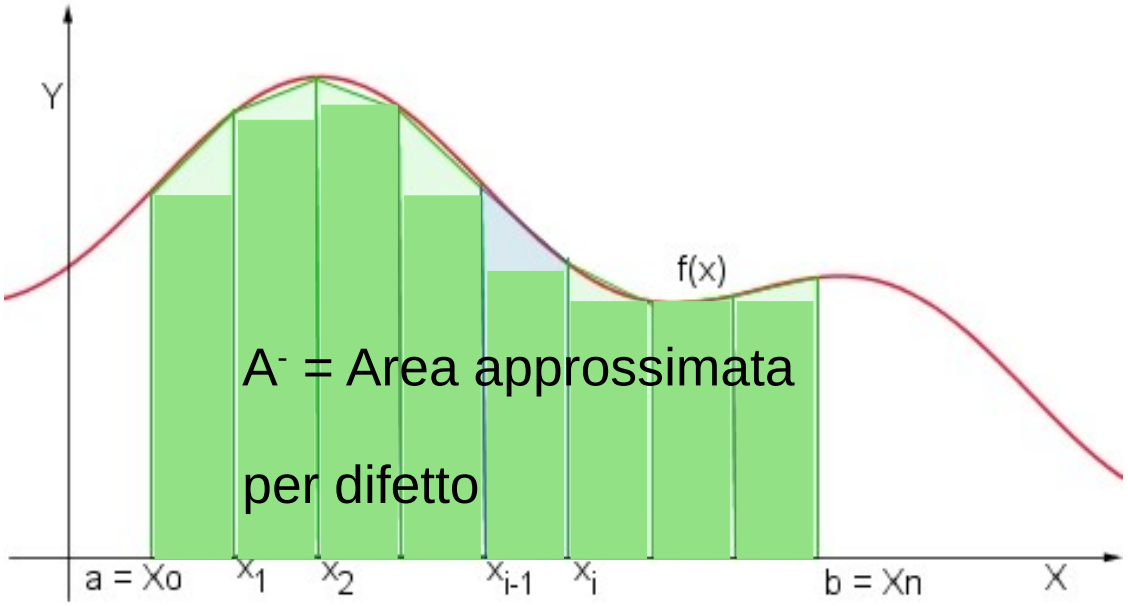
UNIVERSITÀ DEGLI STUDI DI TRIESTE

Integrale definito in [a b]

$$\int_a^b f(x)\,dx$$

Calcolo dell'integrale di y=f(x) nell'intervallo [a b].

Possiamo suddividere l'intervallo [a b] in n parti (uguali) e calcolare la somma delle aree dei rettangoli (vedi figura)

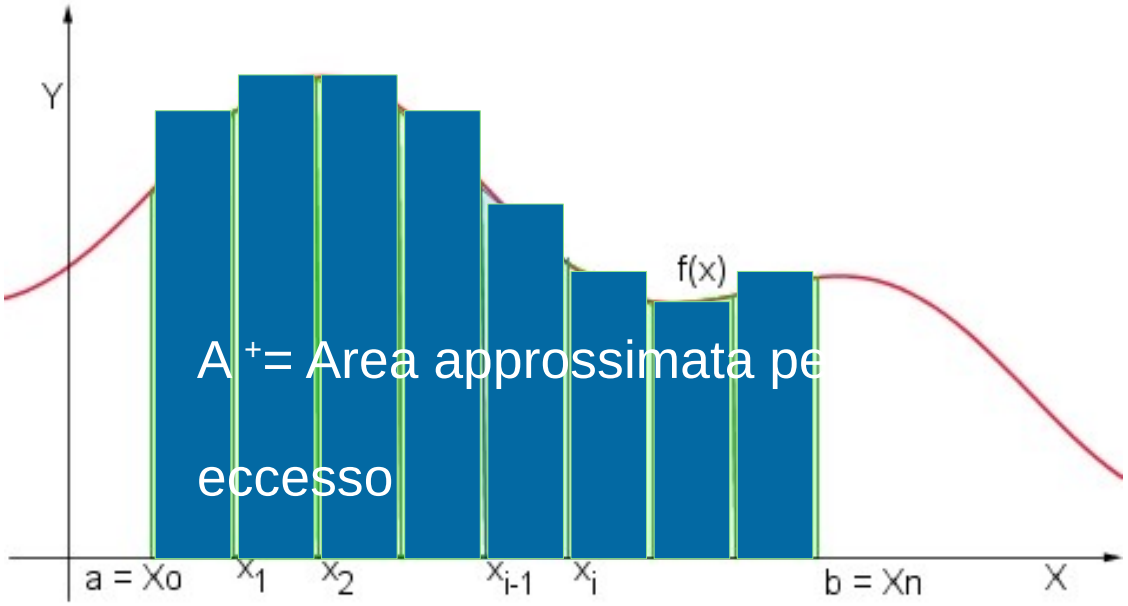


UNIVERSITÀ DEGLI STUDI DI TRIESTE

Integrale definito in [a b]

$$\int_a^b f(x)\,dx$$

Calcolo dell'integrale di y=f(x) nell'intervallo [a b].

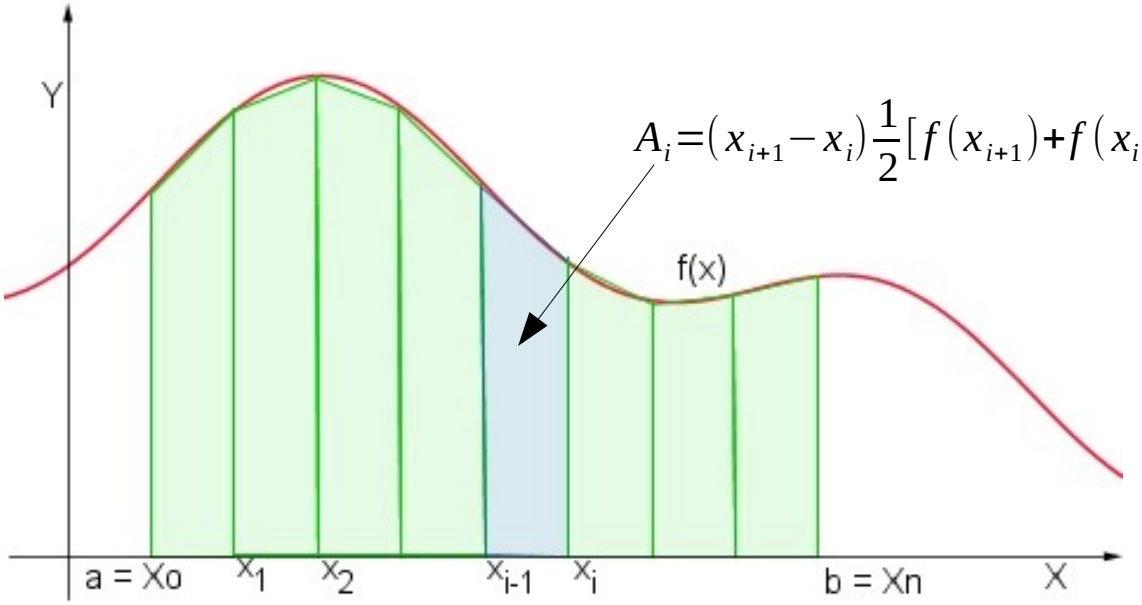


Integrale definito in [a b]

Calcolo dell'integrale di y=f(x) nell'intervallo [a b]

Per migliorare l'approssimazione possiamo ridurre il passo x_{i+1}-x_i e fare la media tra i due valori A^+ e A^- .

In alternativa, possiamo consideriamo i trapezi rettangoli di altezza x_{i+1}-$x_{i,}$ e basi $f(x_{i+1})$ e $f(x_i)$



UNIVERSITÀ DEGLI STUDI DI TRIESTE

Integrale definito in [a b]

Calcolo dell'integrale di y=f(x) nell'intervallo [a b]. Metodo dei trapezi

$$\int_a^b f(x)\,dx \approx \sum_{i=0}^{i=n-1} (x_{i+1}-x_i)\frac{1}{2}[f(x_{i+1})+f(x_i)]$$

Se dividiamo l'intervallo [a b] in parti uguali $x_{i+1}-x_i=\Delta x=\frac{b-a}{n}$ otteniamo

$$\int_a^b f(x)\,dx \approx \frac{1}{2}\frac{b-a}{n}\sum_{i=0}^{i=n-1}[f(x_{i+1})+f(x_i)]=\frac{1}{2}\frac{b-a}{n}\left(\sum_{i=0}^{i=n-1} f(x_{i+1})+\sum_{i=0}^{i=n-1} f(x_i)\right)$$

$$\sum_{i=0}^{i=n-1} f(x_{i+1})=\sum_{i=1}^{i=n} f(x_i)=\sum_{i=0}^{i=n} f(x_i)-f(a)$$

$$\sum_{i=0}^{i=n-1} f(x_i)=\sum_{i=0}^{i=n} f(x_i)-f(b)$$

$$\int_a^b f(x)\,dx \approx \frac{1}{2}\frac{b-a}{n}\left(2\sum_{i=0}^{i=n} f(x_i)-f(a)-f(b)\right)= \frac{b-a}{n}\left(\sum_{i=0}^{i=n} f(x_i) - \frac{(f(a)+f(b))}{2}\right)$$

Integrale definito in [a b]

Calcolo dell'integrale di y=f(x) nell'intervallo [a b]. Metodo dei trapezi

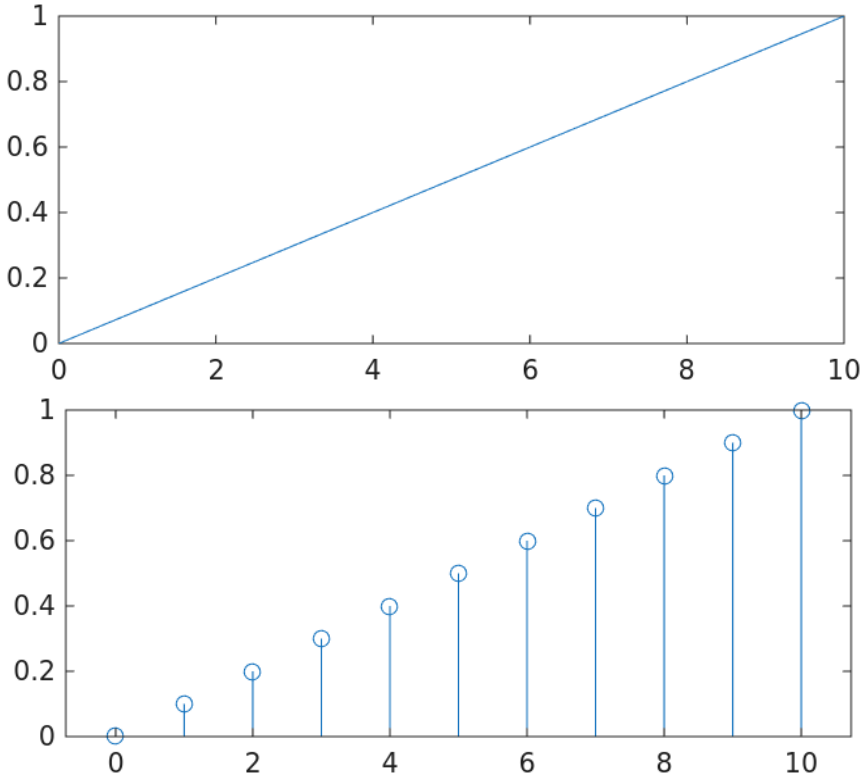
$$\int_a^b f(x)\,dx \approx \frac{b-a}{n}\left(\sum_{i=0}^{i=n} f(x_i) - \frac{(f(a)+f(b))}{2}\right)$$

y=[0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1]

x=0:10

figure(1)
stem(x,y)
a=y(1)
b=y(end)
n=size(y,2)-1
h=(b-a)/n
integral=h*(sum(y)-0.5*(y(1)+y(end)))
figure(2)
plot(x,y)

integral=5



UNIVERSITÀ DEGLI STUDI DI TRIESTE

Integrale definito in [a b]

Calcolo dell'integrale di y=f(x) nell'intervallo [a b]. Metodo di Simpson o delle parabole. Formula di Simpson. L'integrale si può approssimare con l'area delle parabole che passano per ogni terna di punti successivi. I punti devono quindi essere dispari (n pari)

$$\int_a^b f(x)\,\mathrm{d}x \approx 2\Delta x \sum_{i=1}^{n/2} \frac{f(x_{2i-2}) + 4f(x_{2i-1}) + f(x_{2i})}{6} =$$

$$= \frac{\Delta x}{3}\Big(f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) \cdots + 4f(x_{n-1}) + f(x_n)\Big)$$

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Esercizio

Scrivere uno script Matlab che permetta il calcolo dell’integrale definito di una funzione data (a vostra scelta) in un intervallo [a b] con il metodo di Simpson

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Sistemi Lineari

Uno dei problemi più importanti nel calcolo tecnico è la risoluzione dei sistemi di equazioni lineari.

Date due matrici A e b, esiste una matrice x unica, per cui **Ax= b** o xA= b?

È utile considerare un esempio 1x1. Ad esempio, l'equazione

7x = 21

presenta una soluzione unica?

La risposta ovviamente è sì. L'equazione presenta la soluzione unica x = 3. La soluzione viene ottenuta facilmente tramite la divisione:

x = 21/7 = 3.

La soluzione non viene ottenuta ordinariamente calcolando l'inverso di 7, cioè 7^{-1}= 0,142857..., e quindi moltiplicando 7^{-1} per 21. Sarebbe più laborioso e, rappresentando 7^{-1} con un numero finito di cifre, meno accurato. Considerazioni simili si applicano anche alle serie di equazioni lineari con più di un'incognita; **MATLAB® le risolve senza calcolare l'inverso della matrice.**

Benché non si tratti di notazione matematica standard, MATLAB utilizza la familiare terminologia delle divisioni nel caso scalare per descrivere la soluzione di un sistema generale di equazioni simultanee. I due simboli della divisione, la **barra, */**, e la **barra rovesciata, \,** corrispondono alle due funzioni di MATLAB **mrdivide e mldivide**. Questi operatori sono utilizzati per le situazioni in cui la matrice incognita appare sulla sinistra o sulla destra della matrice dei coefficienti:

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Sistemi Lineari

x = b/A	Denota la soluzione dell'equazione matriciale xA = b, ottenuta con mrdivide.
x = b\A	Denota la soluzione dell'equazione matriciale Ax = b, ottenuta con mldivide.

Si pensi alla "divisione" dei due lati dell'equazione Ax = b o xA = b per A. La matrice di coefficienti A si trova sempre nel "denominatore".

Le condizioni di compatibilità delle dimensioni per **x = A\b** richiedono che le due matrici A e b presentino lo stesso numero di righe. La soluzione x quindi presenta lo stesso numero di colonne di b e la dimensione delle sue righe è uguale alle dimensioni delle colonne di A.

Per **x = b/A** i ruoli di righe e colonne risultano scambiati.

In pratica, le equazioni lineari della forma **Ax = b** sono più frequenti rispetto a quelle della forma xA = b. La **barra rovesciata** è quindi molto più utilizzata rispetto alla barra dritta.

Vale la seguente identità

(b / A) ' = (A' \ b')

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Sistemi Lineari

- Il sistema di m equazioni lineari in n incognite $x_1, \ldots, x_n$:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \quad \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{cases}$$

può essere scritto in forma matriciale come:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{Bmatrix}$$

$$\mathbf{Ax} = \mathbf{b}$$

  - $m = n$ e $\det(\mathbf{A}) \neq 0$, ossia $\text{rank}(\mathbf{A}) = m = n$, $\Rightarrow$ sistema determinato;
  - $m > n \Rightarrow$ sistema sovradeterminato;
  - $m < n \Rightarrow$ sistema sottodeterminato.

- In MATLAB il precedente sistema si risolve con il comando *backslash*:

```
x=A\b
```

che sceglie automaticamente che tipo di soluzione cercare in funzione delle dimensioni e delle proprietà della matrice $\mathbf{A}$.

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Sistemi Lineari

La soluzione generale di un sistema di equazioni lineari Ax= b descrive tutte le soluzioni possibili.

È possibile trovare la soluzione generale come segue:

1) Risolvere il sistema omogeneo corrispondente **Ax = 0.**
A questo scopo utilizzare il la funzione null, digitando **null(A**). Questo restituisce una base per lo spazio della soluzione verso Ax = 0. Qualsiasi soluzione è una combinazione lineare di vettori della base.

2) Trovare una soluzione particolare al sistema non omogeneo **Ax = b**.

È poi possibile scrivere qualsiasi soluzione per Ax = b come somma della soluzione particolare verso Ax =b, dal punto 2, oltre una combinazione lineare dei vettori di base dal punto 1.

Sistemi Lineari

Sistemi quadrati

La situazione più comune interessa una matrice quadrata dei coefficienti A e un singolo vettore colonna sul lato destro b.

Matrice **non singolare** dei coefficienti

Se la matrice A è non singolare, allora la soluzione, **x = A\b**, presenta le stesse dimensioni di b.

Ad esempio:

A = pascal(3); (sicuramente non singolare)

b = [3; 1; 4];

x = A\b

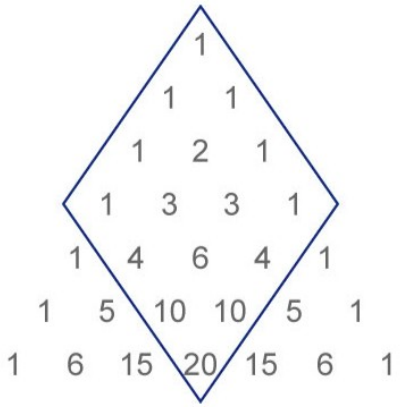
x =

10

-12

5

Verificare che A*x è pari a b.



Triangolo di Pascal e Matrice di Pascal

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Sistemi Lineari

Se A e b sono quadrate e delle stesse dimensioni, anche x= A\b presenta le stesse dimensioni:

```
b = magic(3);

X = A\b

X =

     19    -3    -1

    -17     4    13

      6     0    -6
```

N.B. M = magic(n) restituisce una matrice n x n costruita dai numeri interi da 1 a n^2 nella quale la somma degli elementi delle righe e delle colonne sono uguali.

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Sistemi Lineari

Matrice singolare dei coefficienti

Una matrice quadrata A è singolare se non presenta colonne linearmente indipendenti. Se A è singolare, la soluzione di Ax = b **non esiste oppure non è unica**. L'operatore barra rovesciata, A\b, genera un avviso se A è quasi singolare o se rileva una singolarità esatta.

Se A è singolare e Ax = b ha una soluzione, è possibile trovare una soluzione particolare non unica digitando

P = pinv(A)*b

pinv(A) è una pseudo-inversa di A. Se Ax = b non presenta una soluzione esatta, pinv(A) restituisce una soluzione basata sui minimi quadrati.

Ad esempio:

```
A = [ 1     3     7
     -1     4     4
      1    10    18 ]
```

è singolare, come è possibile verificare digitando

```
rank(A)
ans =
     2
```

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Sistemi Lineari

Sistemi sovradeterminati

Questo esempio dimostra quanto siano frequenti i sistemi sovradeterminati in diversi tipi di adattamento delle curve a dati sperimentali.

Una quantità y viene misurata in diversi valori di tempo t per produrre le osservazioni sotto riportate. Le seguenti dichiarazioni consentono di inserire i dati e di visualizzarli in una tabella.

```
t = [0 .3 .8 1.1 1.6 2.3]';
y = [.82 .72 .63 .60 .55 .50]';
B = table(t,y)
```

Provare a modellare i dati con una funzione di decadimento esponenziale

$y(t)=c1+c2e^{-t}$

L'equazione precedente dichiara che il vettore y dovrebbe essere approssimato da una combinazione lineare di altri due vettori. Uno è un vettore costante contenente tutti uno, l'altro è il vettore con i componenti exp(-t). È possibile calcolare i coefficienti incogniti c1 e c2 con un adattamento basato sui minimi quadrati, che minimizza la somma dei quadrati delle deviazioni dei dati dal modello. Vi sono sei equazioni in due incognite, rappresentate da una matrice 6x2.

```
E = [ones(size(t)) exp(-t)]
```

Sistemi Lineari

Utilizzare l'operatore barra rovesciata per ottenere la soluzione basata sui minimi quadrati.

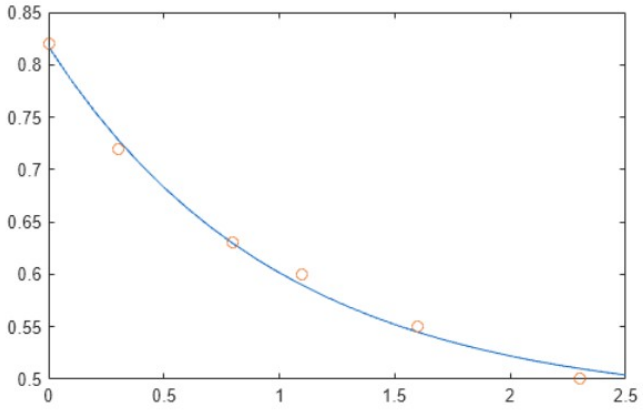
```
c = E\y
c = 2×1

    0.4760
    0.3413
```

In altri termini, l'adattamento basato sui minimi quadrati dei dati è $y(t)=0.4760+0.3413e-t$

Le dichiarazioni seguenti valutano il modello in corrispondenza di incrementi con spaziamento regolare in t, quindi tracciano il risultato insieme ai dati originali:

```
T = (0:0.1:2.5)';
Y = [ones(size(T)) exp(-T)]*c;
plot(T,Y,'-',t,y,'o')
```



E*c non è esattamente uguale a y, ma la differenza potrebbe risultare inferiore rispetto agli errori di misurazione nei dati originali.

Una matrice rettangolare A presenta una carenza di rank se non ha colonne linearmente indipendenti. Se A presenta una carenza di rank, la soluzione basata sui minimi quadrati di AX = B non è unica. A\B genera un errore se A presenta una carenza di rank e genera una soluzione basata sui minimi quadrati. È possibile utilizzare **lsqminnorm** per trovare la soluzione di X con norma minima tra tutte le soluzioni.

https://it.mathworks.com/help/matlab/math/systems-of-linear-equations.html

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Distanza tra un punto e una retta

Calcolare la distanza tra un punto $\mathbf{P}_0$ nello spazio ed una retta r nella forma esplicita

$$\mathbf{r}(\lambda) = \mathbf{r}_0 + \lambda \mathbf{v}$$

Metodo A (minimizzazione esplicita): si determina esplicitamente il valore di λ che minimizza la distanza punto-retta, equivalente ad imporre $\mathbf{P}_0 - \mathbf{r}(\lambda)$ ortogonale a $\mathbf{v}$:

$$\min_{\lambda \in \mathbb{R}} \|\mathbf{P}_0 - \mathbf{r}(\lambda)\|^2 \quad \Rightarrow \quad \bar{\lambda} = \frac{(\mathbf{P}_0 - \mathbf{r}_0)^T \mathbf{v}}{\mathbf{v}^T \mathbf{v}}$$

$$dist = \|\mathbf{P}_0 - \mathbf{r}(\bar{\lambda})\|$$

```
% Valori particolari
P0 = [ 1 ; 1 ; 0 ] ; % punto

r0 = [ 0 ; 0 ; 0 ] ; % retta, punto di partenza
v  = [ 1 ;-1 ; 0 ] ; % retta, vettore direzione
r  = @(l) r0 + l*v ; % funzione anonima della retta

% Metodo A
lambda = ( (P0-r0)' * v ) / ( v'*v ) ;
dist_A = norm( P0 - r(lambda) )
```

Che fornisce il risultato:

```
dist_A =
       1.4142
```

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Distanza tra un punto e una retta

Metodo B (sottrazione componente parallela): si scompone la differenza $\mathbf{d} = \mathbf{P}_0 - \mathbf{r}_0$ in una componente parallela e in una ortogonale a $\mathbf{v}$:

$$\mathbf{d} = \mathbf{d}_\mathbf{v} + \mathbf{d}_\perp$$

Fissato $\mathbf{e}_\mathbf{v} = \mathbf{v}/\|\mathbf{v}\|$ il vettore di lunghezza unitaria in direzione $\mathbf{v}$, la componente parallela si ottiene direttamente dal prodotto scalare lungo $\mathbf{e}_\mathbf{v}$ in quanto $\mathbf{d}_\perp \cdot \mathbf{e}_\mathbf{v} = 0$:

$$\mathbf{d} \cdot \mathbf{e}_\mathbf{v} = \mathbf{d}_\mathbf{v} \cdot \mathbf{e}_\mathbf{v} \quad \Rightarrow \quad \mathbf{d}_\mathbf{v} = (\mathbf{d} \cdot \mathbf{e}_\mathbf{v})\mathbf{e}_\mathbf{v}$$

Si sottrae quindi a $\mathbf{d}$ la sua componente parallela a $\mathbf{v}$, lasciando quindi la componente ortogonale $\mathbf{d}_\perp$ la cui lunghezza è proprio la distanza cercata:

$$dist = \|\mathbf{d} - \mathbf{d}_\mathbf{v}\| = \|\mathbf{d}_\perp\|$$

```
% Metodo B
e_v      = v / norm(v) ;        % vettore unitario lungo v
d        = P0 - r0 ;            % differenza punto-partenza retta
d_v      = dot(d, e_v) * e_v ; % componente parallela
d_ortho = d - d_v ;            % componente ortogonale
dist_B   = norm( d_ortho )      % distanza
```

Che fornisce il risultato:

```
dist_B =
       1.4142
```

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Distanza tra un punto e una retta

Metodo C (matriciale): si scrivono le condizioni di ortogonalità in forma esplicita:

$$\mathbf{d} = \mathbf{d_v} + \mathbf{d}_\perp = \lambda \mathbf{v} + I_3 \mathbf{d}_\perp$$

$$\mathbf{d}_\perp \cdot \mathbf{v} = 0$$

Si assumono come incognite il vettore ortogonale $\mathbf{d}_\perp$ ed il coefficiente λ, per un totale di 3+1 scalari, da esprimere mediante il seguente sistema lineare:

$$\begin{bmatrix} I_3 & \mathbf{v} \\ \mathbf{v}^T & 0 \end{bmatrix} \begin{Bmatrix} \mathbf{d}_\perp \\ \lambda \end{Bmatrix} = \begin{Bmatrix} \mathbf{d} \\ 0 \end{Bmatrix}$$

$$\mathbf{M}\mathbf{x} = \mathbf{b}$$

```
% Metodo C (esagerato)
I3 = eye(3,3) ;
M  = [ I3 v ; ...
       v' 0 ] ;     % costruzione matrice
d = P0 - r0 ;       % differenza punto-partenza retta
b = [ d ; 0 ] ;     % vettore termini noti
x = M\b ;           % soluzione mediante backslash
d_ortho = x(1:3) ; % vettore ortogonale = prime 3 incognite
dist_C  = norm( d_ortho )
```

Che fornisce il risultato corretto dist_C = 1.4142

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Distanza tra un punto e una retta

Metodo D (Pitagora): si fa il prodotto scalare

$$\mathbf{d}\cdot\mathbf{d} = \mathbf{d}\cdot\mathbf{d}_{\mathbf{v}} + \mathbf{d}\cdot\mathbf{d}_{\perp} = (\mathbf{d}\cdot\mathbf{e}_{\mathbf{v}})^2 + dist^2$$

$$dist = \sqrt{\mathbf{d}\cdot\mathbf{d} - (\mathbf{d}\cdot\mathbf{e}_{\mathbf{v}})^2}$$

con $\mathbf{e}_{\mathbf{v}} = \mathbf{v}/\|\mathbf{v}\|$ il vettore di lunghezza unitaria in direzione $\mathbf{v}$.

```
% Metodo D
d    = P0 - r0 ;        % differenza punto-partenza retta
dd   = dot(d, d) ;      % ipotenusa^2
e_v  = v / norm(v) ;    % vettore unitario lungo v
L_v  = dot(d, e_v)^2 ; % cateto^2 lungo v
dist_D = sqrt( dd - L_v )
```

Che fornisce il risultato corretto dist_D = 1.4142

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Distanza tra un punto P_0 nello spazio ed un piano q

$$\mathbf{q}(\lambda_1, \lambda_2) = \mathbf{q}_0 + \lambda_1 \mathbf{v}_1 + \lambda_2 \mathbf{v}_2 = \mathbf{q}_0 + [\mathbf{v}_1 \ \mathbf{v}_2] \begin{Bmatrix} \lambda_1 \\ \lambda_2 \end{Bmatrix} = \mathbf{q}_0 + \mathbf{V}\boldsymbol{\lambda}$$

Metodo A (ortogonalità): si impone $\mathbf{P}_0 - \mathbf{q}(\lambda_1, \lambda_2)$ ortogonale al piano:

$$\mathbf{v}_i \cdot \big(\mathbf{P}_0 - \mathbf{q}(\lambda_1, \lambda_2)\big) = 0 \qquad i = 1, 2$$

$$\mathbf{V}^T \big(\mathbf{P}_0 - \mathbf{q}_0 - \mathbf{V}\boldsymbol{\lambda}\big) = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \quad \Rightarrow \quad \mathbf{V}^T\mathbf{V}\boldsymbol{\lambda} = \mathbf{V}^T(\mathbf{P}_0 - \mathbf{q}_0)$$

$$dist = \|\mathbf{P}_0 - \mathbf{q}(\lambda_1, \lambda_2)\|$$

```
% Valori particolari
P0 = [ 3 ; 3 ; 5 ] ; % punto

q0 = [ 0 ; 4 ; 2 ] ; % piano, punto di partenza
v1 = [ 1 ; 3 ; 0 ] ; % piano, primo vettore direzione
v2 = [ 0 ; 2 ; 5 ] ; % piano, secondo vettore direzione

V = [ v1 v2 ] ;                % matrice V = [ v1 v2 ]
q = @(lambda) q0 + V*lambda ; % funzione piano

% Metodo A
A = V' * V ;         % matrice quadrata dei coefficienti
b = V' * (P0-q0) ; % vettore dei termini noti
lambda = A\b ;       % soluzione, lambda
dist_A = norm( P0 - q(lambda) )
```

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Distanza tra un punto P_0 nello spazio ed un piano q

- Metodo B (prodotto vettoriale): si calcola il prodotto vettoriale

$$\mathbf{c} = \mathbf{v}_1 \times \mathbf{v}_2 = \det \begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ v_{1,x} & v_{1,y} & v_{1,z} \\ v_{2,x} & v_{2,y} & v_{2,z} \end{bmatrix}$$

e lo si normalizza mediante $\mathbf{e_c} = \mathbf{c}/\|\mathbf{c}\|$. Il prodotto vettoriale $\mathbf{c}$ è quindi ortogonale al piano, essendo ortogonale sia a $\mathbf{v}_1$ che a $\mathbf{v}_2$.
La distanza cercata è quindi data dalla componente di $\mathbf{P}_0 - \mathbf{q}_0$ lungo $\mathbf{e_c}$, senza segno:

$$dist = |(\mathbf{P}_0 - \mathbf{q}_0) \cdot \mathbf{e_c}|$$

```
% Metodo B
c   = cross( v1 , v2 ) ; % prodotto vettoriale, vettore ortognale al piano
e_c = c / norm( c ) ;    % normalizzazione (lunghezza unitaria)
dist_B = abs( dot(P0-q0, e_c) )
```

- Dall'esecuzione dei precedenti script si ottiene:

```
dist_A =
        3.5138

dist_B =
        3.5138
```

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Piano in forma canonica

Scrivere una funzione che, dato il piano q (nello spazio) nella forma

$$\mathbf{q}(\lambda_1, \lambda_2) = \mathbf{q}_0 + \lambda_1 \mathbf{v}_1 + \lambda_2 \mathbf{v}_2,$$

ne determini i coefficienti della forma canonica

$$c_1 x + c_2 y + c_3 z + d = \mathbf{c} \cdot \mathbf{q} + d = 0$$

Il piano nella forma canonica può essere scritto come prodotto scalare

$$\mathbf{c} \cdot (\mathbf{q} - \mathbf{q}_0) = 0 \quad \Rightarrow \quad d = -\mathbf{c} \cdot \mathbf{q}_0$$

Il vettore $\mathbf{c}$ è ortogonale a $\mathbf{v}_1$ e $\mathbf{v}_2$:

$$\mathbf{c} = \mathbf{v}_1 \times \mathbf{v}_2$$

piano_canonico.m

```
% Input: q0, vettore colonna 3x1 di un punto nel piano
%        v1 e v2, vettori colonna 3x1 delle direzioni lungo il piano
% Output: c, vettore colonna 3x1 dei coefficienti della forma canonica
%         d, costante della forma canonica
function [ c , d ] = piano_canonico(q0, v1, v2)
  c = cross(v1, v2) ;
  d = -dot(c, q0) ;
end
```

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Regressione lineare

Regressione lineare. Data una serie di n valori x_i della variabile indipendente in corrispondenza dei quali sono dati n valori $\bar{y}_i$ della variabile dipendente, ad esempio delle misurazioni, ci si pone il problema di determinare la retta $y = c_1 + c_2 x$ che "meglio" approssima i dati forniti.

Soluzione ai minimi quadrati: c_1 e c_2 si determinano minimizzando la somma dei quadrati degli scarti:

$$\min_{c_1,c_2\in\mathbb{R}} \sum_{i=1}^{n} \delta_i^2 = \min_{c_1,c_2\in\mathbb{R}} \sum_{i=1}^{n} \Big((c_1 + c_2 x_i) - \bar{y}_i \Big)^2$$

Organizzando i dati nei vettori colonna $\mathbf{x} = (x_i)$ e $\bar{\mathbf{y}} = (\bar{y}_i)$, il vettore colonna degli scarti è

$$\boldsymbol{\delta} = (c_1 + c_2\mathbf{x}) - \bar{\mathbf{y}} = \mathbf{A}\mathbf{c} - \bar{\mathbf{y}}$$

dove

$$\mathbf{A} = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{x} \end{bmatrix}, \quad \mathbf{c} = \begin{Bmatrix} c_1 \\ c_2 \end{Bmatrix}$$

Il problema si può quindi risolvere calcolando la soluzione ai minimi quadrati del sistema lineare sovradeterminato $\mathbf{A}\mathbf{c} = \bar{\mathbf{y}}$, dove la matrice dei coefficienti $\mathbf{A}$ ha dimensione $n \times 2$.

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Regressione lineare

```
% Numero di valori
n = 50 ;

% Valori variabile indipendente
x = linspace(0, 1, n)' ;

% Valori variabile dipendente, misura
c1 = 2.0 ;
c2 = 3.0 ;
r  = 0.1 ;
y = c1 + c2 * x ;
rumore = rand(size(x)) - .5 ;
y = y + r * rumore .* y ;

% Minimi quadrati
A = [ ones(n,1) x ] ;
c = A\y ;
```

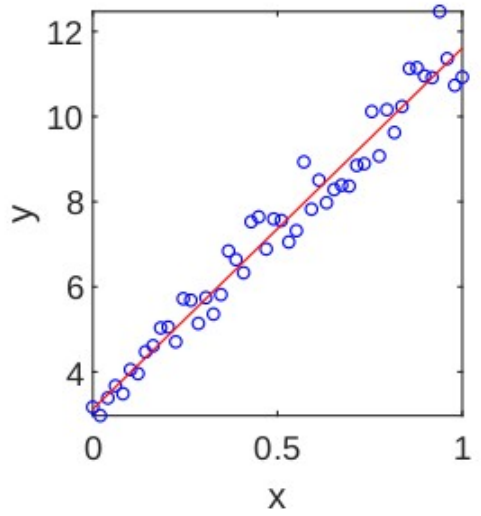
UNIVERSITÀ DEGLI STUDI DI TRIESTE

Regressione lineare

```
% Plot
plot( x , y , 'ob' ) ; % dati
hold on ;
plot( x , c(1)+c(2)*x , '-r' ) ; % retta di regressione
hold on ;
plot( x , c1 + c2*x , '-b' ) ;

% Output testuale dei coefficienti
testo = [ 'Regressione lineare di %d valori:\n' ...
          ' c      = (%.2f,%.2f)\n' ...
          ' c_regr = (%.2f,%.2f)\n' ] ;
fprintf(testo, n,          ...
               c1  , c2 , ...
               c(1), c(2) ) ;

% Minimi quadrati espliciti
c_lsq = (A'*A)\(A'*y) ;
```



- L'output testuale che si ottiene nella command window è

```
Regressione lineare di 50 valori:
 c      = (3.00,9.00)
 c_regr = (2.96,9.06)
```

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Regressione lineare

Alcune funzioni utili

- **norm**(v) per la norma 2 di vettori (e matrici) ||v||,
- **dot**(a,b) per il prodotto scalare a · b,
- **cross**(a,b) per il prodotto vettoriale a × b,
- **null**(A) per il ker, o spazio nullo ker(A),
  Vi sono altre funzioni particolarmente utili per l'algebra lineare:
- **det**(A) per calcolare il determinante di una matrice quadrata A;
- **inv**(A) per calcolare l'inversa di una matrice quadrata invertibile A;
- **rank**(A) per calcolare il rango di una matrice A;
- R=**orth**(A) per calcolare una base ortonormale del range di A: R ha lo stesso numero di righe di A ed un numero di colonne pari al rango di A; ogni colonna è un vettore della base ortonormale;
- [V,D]=**eig**(A) per calcolare autovettori (colonne di V) ed autovalori (elementi diagonali di D) di una matrice quadrata A. Se l'argomento richiesto in uscita alla funzione è unico, d=eig(A), gli autovalori sono disposti in un vettore colonna, d.

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Energia

Energia di un segnale tempo-continuo in un intervallo [a,b]

$$E_x = \varepsilon \int_a^b |x(t)|^2 dt$$

Dato che eseguiamo il calcolo numericamente, dobbiamo utilizzare un algoritmo per e:

$$\int_a^b f(x)dx \approx \sum_{i=0}^{i=n-1} (x_{i+1}-x_i)\frac{1}{2}[f(x_{i+1})+f(x_i)]$$

Se dividiamo [a,b] in parti n parti uguali

$$x_{i+1}-x_i = \Delta x = \frac{b-a}{n}$$

Otteniamo

$$\int_a^b f(x)dx \approx \frac{1}{2}\cdot\frac{b-a}{n}\sum_{i=0}^{i=n-1}[f(x_{i+1})+f(x_i)]$$

$$\sum_{i=0}^{i=n-1}[f(x_{i+1})+f(x_i)] = \sum_{i=0}^{i=n-1} f(x_{i+1}) + \sum_{i=0}^{i=n-1} f(x_i) = \sum_{i=0}^{i=n-1} f(x_i) + f(b) + \sum_{i=0}^{i=n-1} f(x_i)$$

$$-f(a)+2\sum_{i=0}^{i=n-1} f(x_i) = -f(a)+2\left(\sum_{i=0}^{i=n} f(x_i) - f(b)\right) = -f(a)-2f(b)+2\left(\sum_{i=0}^{i=n} f(x_i)\right)$$

$$\int_a^b f(x)dx \approx \frac{(b-a)}{n}\left(\sum_{i=0}^{i=n} f(x_i) - \frac{1}{2}f(a) - f(b)\right)$$

- tri.m
- % Create a triangular pulse with width 0.4.
- % fs = 10000; % Sampling frequency (samples/sec)
- % t = -1:1/fs:1; % Time Vector
- % w = .4; % Triangle Width

```
fs = 10000;      % Sampling frequency (samples/sec)
t = -1:1/fs:1;   % Time Vector
w = .4;          % Triangle Width
x = tripuls(t-0.5,w);  % Sampled aperiodic triangle
plot(t,x);
xlabel('Time (sec)');ylabel('Amplitude')
```

- % See also SQUARE, S
- % RECTPULSE,