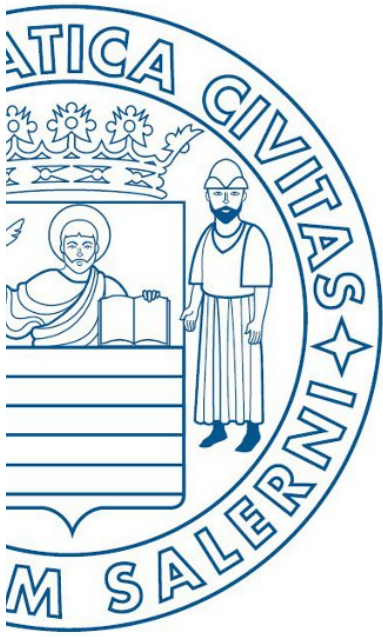




UNIVERSITÀ DEGLI STUDI DI SALERNO

di **in** **Università di Salerno**
Dipartimento di
Ingegneria Industriale



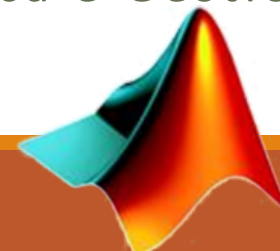
Fondamenti di Informatica

Ripasso Argomenti MATLAB – Possibili Soluzioni

Prof. Christian Esposito

Corso di Laurea in Ingegneria Meccanica e Gestionale (Classe I)

A.A. 2016/17



MATLAB

Parte I

S

<<supermercati.txt>>	Supermercato 1 (Indice 1)	Supermercato 2 (Indice 2)	Supermercato 3 (Indice 3)
Spesa Cliente 1 (indice 1)	210	79	84
Spesa Cliente 2 (indice 2)	129	90	55
Spesa Cliente 3 (indice 3)	229	34	111

P

<<punti_spesa.txt>>	
Supermercato 1 (Indice 1)	25
Supermercato 2 (Indice 2)	15
Supermercato 3 (Indice 3)	12

- In questa esercitazione verranno utilizzati una matrice **S** ed un array colonna **P**
 - L'elemento **S(riga, colonna)** rappresenta l'importo che il cliente, specificato dalla *riga*, ha speso nel supermercato, specificato dalla *colonna*
 - **Esempio:** $S(1, 2) \rightarrow$ indica che l'importo speso dal *Cliente 1* (riga 1), presso il *Supermercato 2* (colonna 2), ammonta a 79€
 - L'array colonna **P** indica l'importo necessario per guadagnare un punto, presso il supermercato specificato dalla *riga*
 - **Esempio:** $P(2) \rightarrow$ indica che un punto viene guadagnato ogni 15€ di spesa effettuata presso il *Supermercato 2* (riga 2)
 - Ad es., su 35€ di spesa vengono guadagnati 2 punti
 - Sopra è mostrato un esempio di **S** e **P**
- Sia la matrice che l'array colonna contengono esclusivamente dati numerici (evidenziati in arancio nell'esempio)

NOTA: Negli esercizi possono essere utilizzate funzioni viste a lezione (negli esempi), funzioni realizzate negli esercizi precedenti e/o funzioni built-in di MATLAB

S

<<supermercati.txt>>	Supermercato 1 (Indice 1)	Supermercato 2 (Indice 2)	Supermercato 3 (Indice 3)
Spesa Cliente 1 (indice 1)	210	79	84
Spesa Cliente 2 (indice 2)	129	90	55
Spesa Cliente 3 (indice 3)	229	34	111

P

<<punti_spesa.txt>>	
Supermercato 1 (Indice 1)	25
Supermercato 2 (Indice 2)	15
Supermercato 3 (Indice 3)	12

Esercizio 1 Scrivere una funzione chiamata `punti_cliente`, che prenda come argomenti di input: la matrice *S* (*supermercati*), l'array *P* (*punti_spesa*) ed un numero intero `indice_cliente`, e restituisca come argomento di output i punti accumulati dal cliente avente come indice il valore di `indice_cliente` (in tutti i supermercati)

N.B.: I punti sono erogati in forma non frazionaria. Ad esempio, se un certo supermercato eroga un punto ogni 12€ di spesa ed un cliente ha effettuato 28€ di spesa, saranno erogati 2 punti per quel cliente (e non 2.33)

- **Esempio:** `punti_cliente(S, P, 2)` → restituisce 15 (5 + 6 + 4)

Possibile Soluzione

```
function [ punti ] = punti_cliente(S, P, indice_cliente)
    punti = sum(floor(S(indice_cliente, :) ./ P'));
end
```

S

<<supermercati.txt>>	Supermercato 1 (Indice 1)	Supermercato 2 (Indice 2)	Supermercato 3 (Indice 3)
Spesa Cliente 1 (indice 1)	210	79	84
Spesa Cliente 2 (indice 2)	129	90	55
Spesa Cliente 3 (indice 3)	229	34	111

P

<<punti_spesa.txt>>	
Supermercato 1 (Indice 1)	25
Supermercato 2 (Indice 2)	15
Supermercato 3 (Indice 3)	12

Esercizio 2 Scrivere una funzione chiamata `cliente_spesa_maggiore`, che prenda come input: la matrice *S* (*supermercati*) e restituisca l'indice del cliente che ha effettuato la spesa di importo massimo (in tutti i supermercati)

- **Esempio:** `cliente_spesa_maggiore(S)` → restituisce 3

Possibile Soluzione

```
function [ indice_cliente ] = cliente_spesa_maggiore(S)
    [spesa_cliente, indice_cliente] = max(sum(S, 2));
end
```

S

<<supermercati.txt>>	Supermercato 1 (Indice 1)	Supermercato 2 (Indice 2)	Supermercato 3 (Indice 3)
Spesa Cliente 1 (indice 1)	210	79	84
Spesa Cliente 2 (indice 2)	129	90	55
Spesa Cliente 3 (indice 3)	229	34	111

P

<<punti_spesa.txt>>	
Supermercato 1 (Indice 1)	25
Supermercato 2 (Indice 2)	15
Supermercato 3 (Indice 3)	12

Esercizio 3 Scrivere una funzione chiamata `supermercato_punti`, che prenda come input: la matrice *S* (*supermercati*) e l'array *P* (*punti_spesa*), e restituisca l'indice del supermercato che ha erogato meno punti al totale dei clienti

- **Esempio:** `supermercato_punti(S, P) → restituisce 2`

Possibile Soluzione

```
function [ indice_supermercato ] = supermercato_punti(S, P)
    [num_clienti, num_supermercati] = size(S);

    for indice_supermercato = 1:num_supermercati
        somma_punti(indice_supermercato) = sum(floor(S(:, indice_supermercato) / P(indice_supermercato)));
    end

    [min_supermercato_punti, indice_supermercato] = min(somma_punti);
end
```

S

<<supermercati.txt>>	Supermercato 1 (Indice 1)	Supermercato 2 (Indice 2)	Supermercato 3 (Indice 3)
Spesa Cliente 1 (indice 1)	210	79	84
Spesa Cliente 2 (indice 2)	129	90	55
Spesa Cliente 3 (indice 3)	229	34	111

P

<<punti_spesa.txt>>	
Supermercato 1 (Indice 1)	25
Supermercato 2 (Indice 2)	15
Supermercato 3 (Indice 3)	12

Esercizio 4

Scrivere una funzione chiamata `grafico_supermercati`, che prenda come argomenti di input: la matrice `S` (*supermercati*), e mostri un grafico con le seguenti proprietà

- Sull'asse *X*, riporti gli indici di tutti i supermercati
- Sull'asse *Y*, per ogni punto, riporti l'ammontare totale della spesa (di tutti i clienti) effettuata presso il supermercato riportato sull'asse *X*
- *Titolo*: 'Grafico Supermercati'
- *Etichetta Asse X*: 'Supermercati'
- *Etichetta Asse Y*: 'Ammontare Spesa'

Inoltre, la funzione dovrà restituire un array contenente i valori assegnati all'asse *Y* (**Esempio**: Array riga o colonna [568 203 250])

S

<<supermercati.txt>>	Supermercato 1 (Indice 1)	Supermercato 2 (Indice 2)	Supermercato 3 (Indice 3)
Spesa Cliente 1 (indice 1)	210	79	84
Spesa Cliente 2 (indice 2)	129	90	55
Spesa Cliente 3 (indice 3)	229	34	111

P

<<punti_spesa.txt>>	
Supermercato 1 (Indice 1)	25
Supermercato 2 (Indice 2)	15
Supermercato 3 (Indice 3)	12

Possibile Soluzione

Esercizio 4

```
function [ somma ] = grafico_supermercati(S)
    [num_clienti, num_supermercati] = size(S);

    x = [1:1:num_supermercati];
    y = sum(S);

    plot(x, y);

    title('Grafico Supermercati');
    xlabel('Supermercati');
    ylabel('Ammontare Spesa');

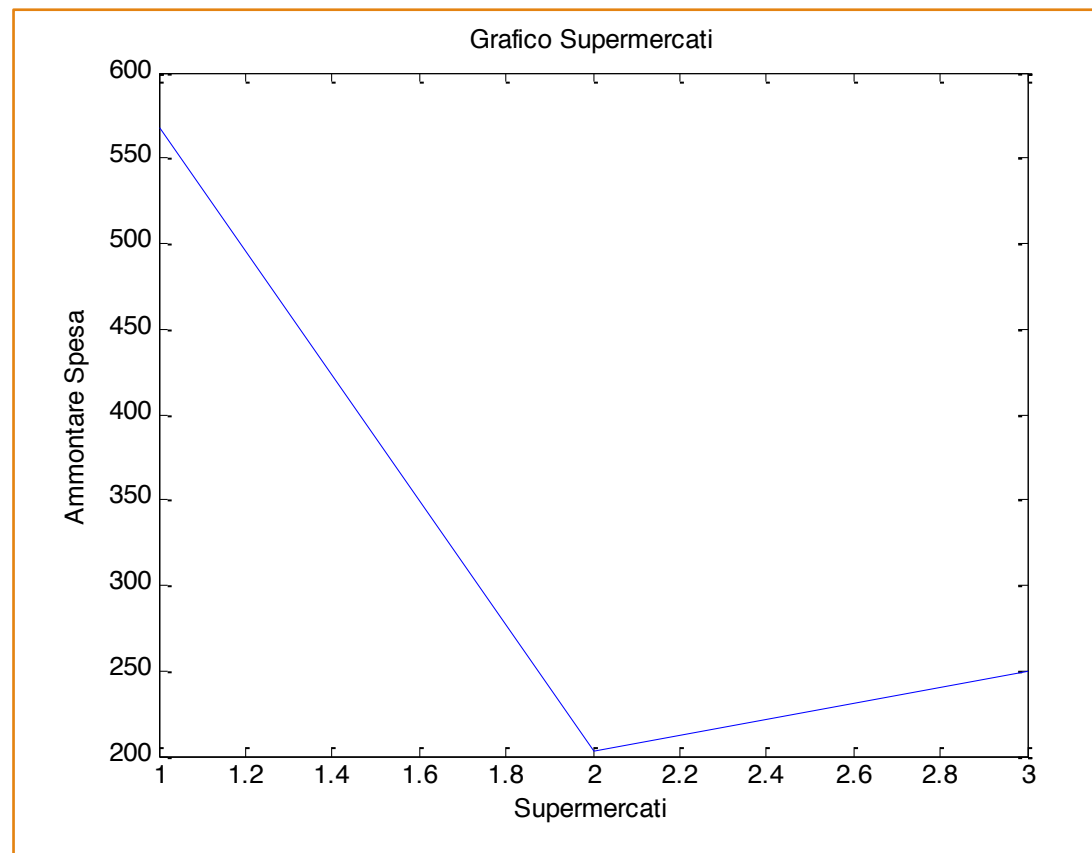
    somma = y;
end
```

S

<<supermercati.txt>>	Supermercato 1 (Indice 1)	Supermercato 2 (Indice 2)	Supermercato 3 (Indice 3)
Spesa Cliente 1 (indice 1)	210	79	84
Spesa Cliente 2 (indice 2)	129	90	55
Spesa Cliente 3 (indice 3)	229	34	111

P

<<punti_spesa.txt>>	
Supermercato 1 (Indice 1)	25
Supermercato 2 (Indice 2)	15
Supermercato 3 (Indice 3)	12

Esercizio 4**Esempio Esercizio 4**

S

<<supermercati.txt>>	Supermercato 1 (Indice 1)	Supermercato 2 (Indice 2)	Supermercato 3 (Indice 3)
Spesa Cliente 1 (indice 1)	210	79	84
Spesa Cliente 2 (indice 2)	129	90	55
Spesa Cliente 3 (indice 3)	229	34	111

P

<<punti_spesa.txt>>	
Supermercato 1 (Indice 1)	25
Supermercato 2 (Indice 2)	15
Supermercato 3 (Indice 3)	12

Esercizio 5 Scrivere un M-File Script chiamato `supermercati_script.m` che effettui le seguenti operazioni

1. Importi la matrice `S` dal file `supermercati.txt`
2. Importi la matrice `P` dal file `punti_spesa.txt`
3. Invochi la funzione dell'Esercizio 3 (chiamata `supermercato_punti`) con gli argomenti di input: `S` e `P`, ed infine mostri a video il risultato della funzione stessa

NOTA: I file `supermercati.txt` e `punti_spesa.txt` contengono solo dati numerici. È utilizzato il separatore virgola (,) per separare le colonne (**suggerimento:** utilizzare la funzione `importdata`). Si assuma che i file siano memorizzati all'interno della **Current Directory**

Possibile Soluzione (contenuto del file `supermercati_script.m`)

```
S = importdata('supermercati.txt');
P = importdata('punti_spesa.txt');
supermercato_punti(S, P)
```

Parte II

Un fisico effettua una misurazione della distanza coperta da una palla di cannone sparata dalla sommità di una torre. Regularmente, il fisico effettua la misurazione dell'altezza raggiunta dal proiettile rispetto al livello del mare, considerando che all'istante 8 la palla di cannone finisce in mare.

	0	1	2	3	4	5	6	7	8
Altezza	40.12	66.78	80.17	86.71	80.77	66.78	44.41	10.51	-32.60

Esercizio 1 Trovare i coefficienti del polinomio di interpolazione lineare per approssimare i dati in tabella per mezzo di una retta. Calcolare la pendenza e l'intersezione con l'asse delle ordinate della retta interpolante. Graficare per mezzo della funzione `plot()` i punti della tabella e la retta interpolante ottenuta. Determinare se la retta rappresenta la migliore interpolazione dei punti della tabella.

Lo svolgimento di questo esercizio deve essere implementato per mezzo di una funzione MATLAB, chiamata `interp_lineare`, che accetta in ingresso gli array `t` e `a`, con `t = 0:8`, mentre `a` contiene i valori della tabella.

Possibile Soluzione

```
function [ coefficienti ] = interp_lineare(t, a)
    coefficienti = polyfit(t,a,1);
    disp(['Pendenza pari a ' num2str(coefficienti(1))]);
    disp(['y-Intercetta pari a ' num2str(coefficienti(2))]);
    subplot(1,2,1); plot(t,a,'rs');
    hold on;
    tt = [0 10];
    plot(tt,polyval(coefficienti,tt),'b-');
    legend('Punti','Interpolazione Lineare');
    res = polyval(coefficienti, t) - a;
    subplot(1,2,2); plot(t, res);
    hold off;
end
```

Un fisico effettua una misurazione della distanza coperta da una palla di cannone sparata dalla sommità di una torre. Regularmente, il fisico effettua la misurazione dell'altezza raggiunta dal proiettile rispetto al livello del mare, considerando che all'istante 8 la palla di cannone finisce in mare.

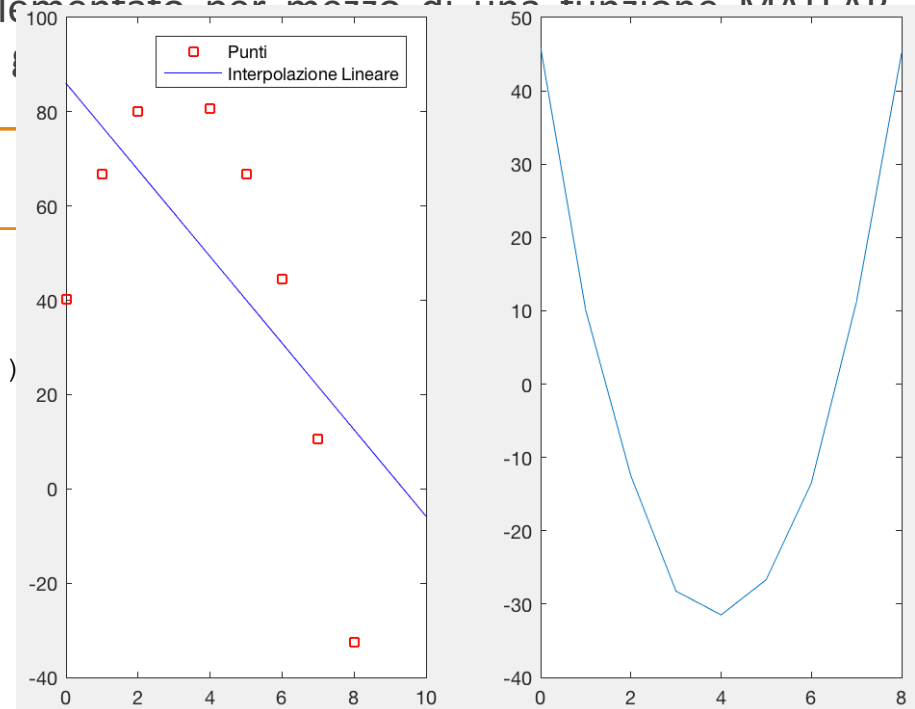
	0	1	2	3	4	5	6	7	8
Altezza	40.12	66.78	80.17	86.71	80.77	66.78	44.41	10.51	-32.60

Esercizio 1 Trovare i coefficienti del polinomio di interpolazione lineare per approssimare i dati in tabella per mezzo di una retta. Calcolare la pendenza e l'intersezione con l'asse delle ordinate della retta interpolante. Graficare per mezzo della funzione `plot()` i punti della tabella e la retta interpolante ottenuta. Determinare se la retta rappresenta la migliore interpolazione dei punti della tabella.

Lo svolgimento di questo esercizio deve essere implementato per mezzo di una funzione MATLAB chiamata `interp_lineare`, che accetta in ingresso i valori della tabella.

Possibile Soluzione

```
function [ coefficienti ] = interp_lineare(t, a)
    coefficienti = polyfit(t,a,1);
    disp(['Pendenza pari a ' num2str(coefficienti(1))]);
    disp(['y-Intercetta pari a ' num2str(coefficienti(2))]);
    subplot(1,2,1); plot(t,a,'rs');
    hold on;
    tt = [0 10];
    plot(tt,polyval(coefficienti,tt),'b-');
    legend('Punti','Interpolazione Lineare');
    res = polyval(coefficienti, t) - a;
    subplot(1,2,2); plot(t, res);
    hold off;
end
```



Un fisico effettua una misurazione della distanza coperta da una palla di cannone sparata dalla sommità di una torre. Regularmente, il fisico effettua la misurazione dell'altezza raggiunta dal proiettile rispetto al livello del mare, considerando che all'istante 8 la palla di cannone finisce in mare.

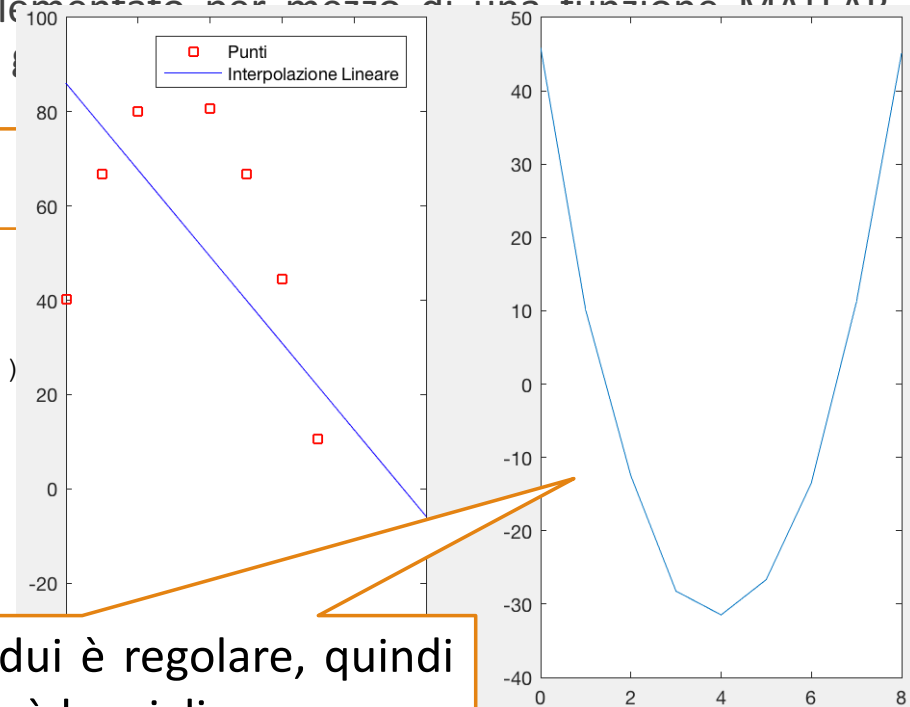
	0	1	2	3	4	5	6	7	8
Altezza	40.12	66.78	80.17	86.71	80.77	66.78	44.41	10.51	-32.60

Esercizio 1 Trovare i coefficienti del polinomio di interpolazione lineare per approssimare i dati in tabella per mezzo di una retta. Calcolare la pendenza e l'intersezione con l'asse delle ordinate della retta interpolante. Graficare per mezzo della funzione `plot()` i punti della tabella e la retta interpolante ottenuta. Determinare se la retta rappresenta la migliore interpolazione dei punti della tabella.

Lo svolgimento di questo esercizio deve essere implementato per mezzo di una funzione MATLAB chiamata `interp_lineare`, che accetta in ingresso i valori della tabella.

Possibile Soluzione

```
function [ coefficienti ] = interp_lineare(t, a)
    coefficienti = polyfit(t,a,1);
    disp(['Pendenza pari a ' num2str(coefficienti(1))]);
    disp(['y-Intercetta pari a ' num2str(coefficienti(2))]);
    subplot(1,2,1); plot(t,a,'rs');
    hold on;
    tt = [0 10];
    plot(tt,polyval(coefficienti,tt),'b-');
    legend('Punti','Interpolazione Lineare');
    res = polyval(coefficienti, t) - a;
    subplot(1,2,2); plot(t, res);
    hold off;
end
```



Il diagramma dei residui è regolare, quindi l'approssimazione non è la migliore.

Un fisico effettua una misurazione della distanza coperta da una palla di cannone sparata dalla sommità di una torre. Regularmente, il fisico effettua la misurazione dell'altezza raggiunta dal proiettile rispetto al livello del mare, considerando che all'istante 8 la palla di cannone finisce in mare.

	0	1	2	3	4	5	6	7	8
Altezza	40.12	66.78	80.17	86.71	80.77	66.78	44.41	10.51	-32.60

Esercizio 2 Determinare la migliore interpolazione possibile tra una polinomiali lineare e una non lineare. Confrontare per mezzo dei residui tale interpolazione con una spline e determinare quale delle due offre la migliore approssimazione. Data l'interpolazione prescelta, determinare l'altezza del proiettile all'istante $t = 5,5$ e all'istante $t = 9$.

Lo svolgimento di questo esercizio deve essere implementato per mezzo di una funzione MATLAB, chiamata `interpolazione`, che accetta in ingresso gli array riga t e a , come in precedenza.

Possibile Soluzione

```
function [ risultato ] = interpolazione(t, a)
    c1 = polyfit(t,a,1);      c2 = polyfit(t,a,2);
    res1 = polyval(c1, t) - a;    res2 = polyval(c2, t) - a;
    plot(t,res1,'r',t,res2,'b');
    disp('Interpolazione polinomiale di grado 2 è la migliore');
    yy = spline(t,a,t);      res_spline = yy - a;
    if abs(mean(res_spline)) < abs(mean(res2))
        disp('La spline ottiene dei residui minori');
        risultato = spline(t, a, [5.5 9]);
    else
        disp('Interpolazione polinomiale di secondo grado ottiene dei residui minori');
        risultato = polyval(c2, [5.5 9]);
    end
end
```

Un fisico effettua una misurazione della distanza coperta da una palla di cannone sparata dalla sommità di una torre. Regularmente, il fisico effettua la misurazione dell'altezza raggiunta dal proiettile rispetto al livello del mare, considerando che all'istante 8 la palla di cannone finisce in mare.

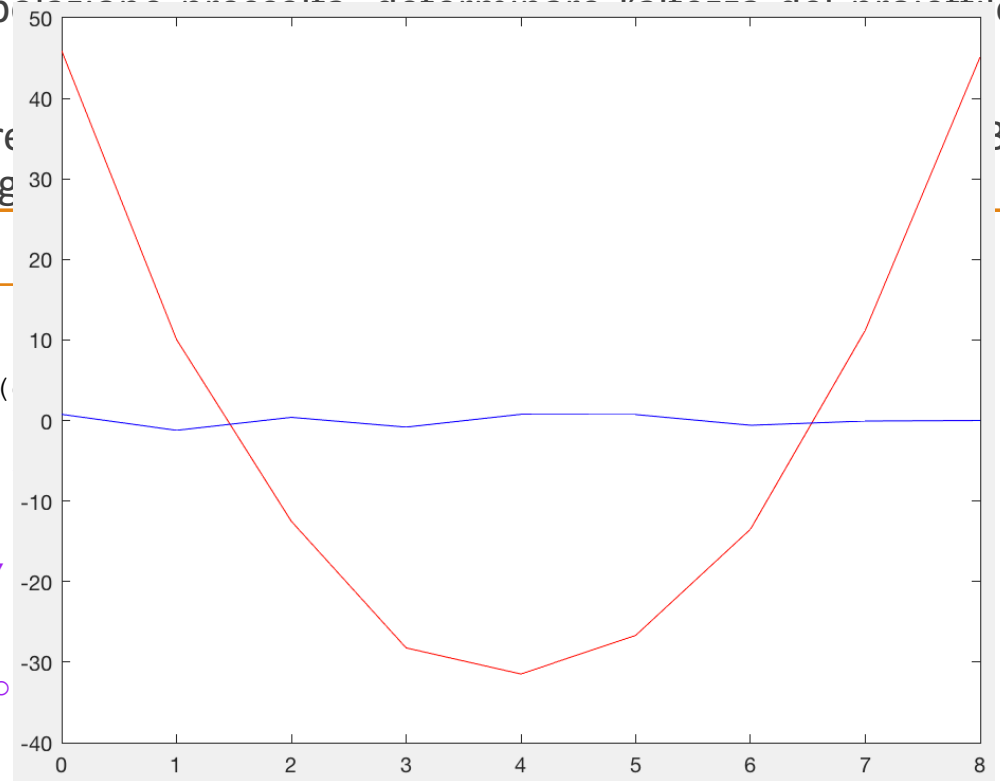
	0	1	2	3	4	5	6	7	8
Altezza	40.12	66.78	80.17	86.71	80.77	66.78	44.41	10.51	-32.60

Esercizio 2 Determinare la migliore interpolazione possibile tra una polinomiali lineare e una non lineare. Confrontare per mezzo dei residui tale interpolazione con una spline e determinare quale delle due offre la migliore approssimazione. Data l'interpolazione, determinare l'altezza del proiettile all'istante $t = 5,5$ e all'istante $t = 9$.

Lo svolgimento di questo esercizio deve essere chiamata interpolazione, che accetta in ing

Possibile Soluzione

```
function [ risultato ] = interpolazione(t, a)
    c1 = polyfit(t,a,1);      c2 = polyfit(t,a,2);
    res1 = polyval(c1, t) - a;      res2 = polyval(
    plot(t,res1,'r',t,res2,'b');
    disp('Interpolazione polinomiale di grado 2 è
    yy = spline(t,a,t);      res_spline = yy - a;
    if abs(mean(res_spline)) < abs(mean(res2))
        disp('La spline ottiene dei residui minori'
        risultato = spline(t, a, [5.5 9]);
    else
        disp('Interpolazione polinomiale di secondo
        risultato = polyval(c2, [5.5 9]);
    end
end
```



Un fisico effettua una misurazione della distanza coperta da una palla di cannone sparata dalla sommità di una torre. Regularmente, il fisico effettua la misurazione dell'altezza raggiunta dal proiettile rispetto al livello del mare, considerando che all'istante 8 la palla di cannone finisce in mare.

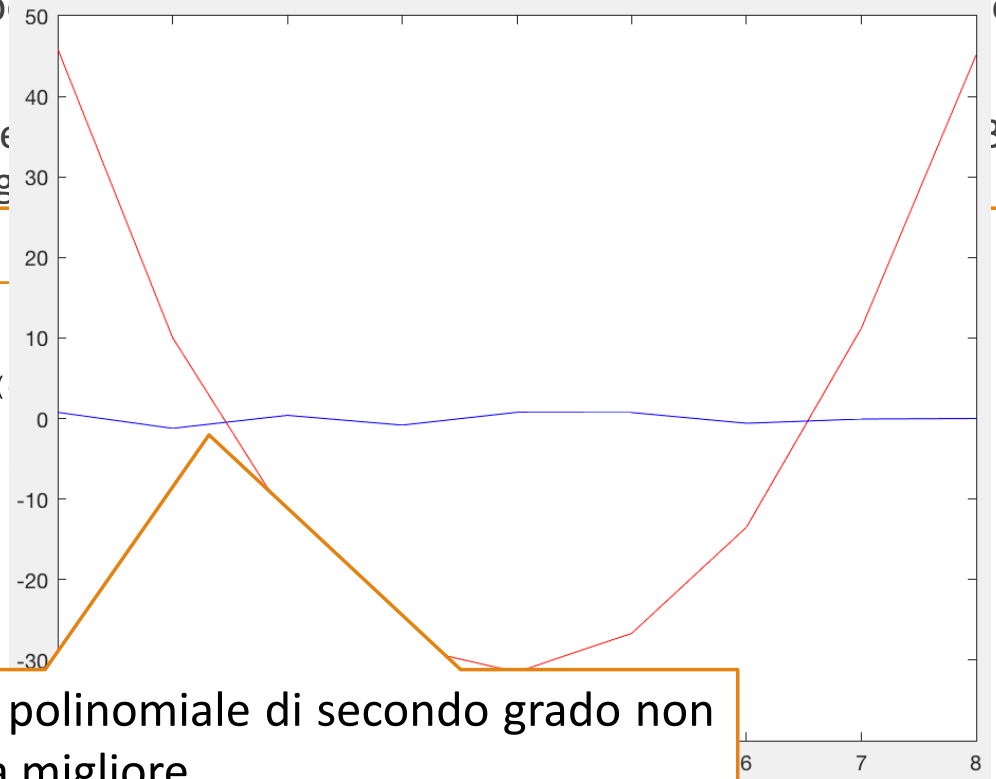
	0	1	2	3	4	5	6	7	8
Altezza	40.12	66.78	80.17	86.71	80.77	66.78	44.41	10.51	-32.60

Esercizio 2 Determinare la migliore interpolazione possibile tra una polinomiali lineare e una non lineare. Confrontare per mezzo dei residui tale interpolazione con una spline e determinare quale delle due offre la migliore approssimazione. Data l'interpolazione, determinare l'altezza del proiettile all'istante $t = 5,5$ e all'istante $t = 9$.

Lo svolgimento di questo esercizio deve essere chiamata interpolazione, che accetta in ing

Possibile Soluzione

```
function [ risultato ] = interpolazione(t, a)
    c1 = polyfit(t,a,1);      c2 = polyfit(t,a,2);
    res1 = polyval(c1, t) - a;      res2 = polyval(
    plot(t,res1,'r',t,res2,'b');
    disp('Interpolazione polinomiale di grado 2 è
    yy = spline(t,a,t);      res_spline = yy - a;
    if abs(mean(res_spline)) < abs(mean(res2))
        disp('La spline ottiene dei residui minori'
        risultato = spline(t, a, [5.5 9]);
    else
```



Il diagramma dei residui dell'interpolazione polinomiale di secondo grado non è regolare come di quella lineare, quindi è la migliore.

Un fisico effettua una misurazione della distanza coperta da una palla di cannone sparata dalla sommità di una torre. Regularmente, il fisico effettua la misurazione dell'altezza raggiunta dal proiettile rispetto al livello del mare, considerando che all'istante 8 la palla di cannone finisce in mare.

	0	1	2	3	4	5	6	7	8
Altezza	40.12	66.78	80.17	86.71	80.77	66.78	44.41	10.51	-32.60

Esercizio 3 Determinare numericamente la velocità del proiettile nel punto apicale del suo moto di caduta, ovvero con $t = 3$.
Lo svolgimento deve essere implementato per mezzo di una funzione MATLAB, chiamata `calc_velocita`, che accetta in ingresso gli array riga `t` e `a`, come in precedenza, e il valore 3.

Possibile Soluzione

```
function [ risultato ] = calc_velocita(t, a, valore)
    x_inf = valore - 2;
    x_sup = valore + 2;
    intervallo = x_inf:.1:x_sup;
    yy_intervallo = polyval(c2, intervallo);
    velocita = diff(yy_intervallo)/diff(intervallo);
end
```

Parte III

Data la seguente funzione: $f(x) = \sqrt{4 - x^2}$

Esercizio 1 Calcolare la funzione derivata di $f(x)$ impiegando le funzioni per il calcolo simbolico di MATLAB:

$$f'(x) = \frac{df(x)}{dx}$$

e determinare il valore numerico assunto da tale funzione per $x = 5$.

Lo svolgimento di questo esercizio deve essere implementato per mezzo di una funzione MATLAB, chiamata `derivata_in_x`, che accetta in ingresso o la funzione e il valore 5, oppure solo il valore 5.

Possibile Soluzione

```
function [ risultato ] = derivata_in_x(fun, valore)
    syms x;
    derivata = diff(fun, x);
    disp(['Derivata pari a ' char(derivata)]);
    risultato = vpa(subs(derivata, valore), 3);
end
```

Data la seguente funzione: $f(x) = \sqrt{4 - x^2}$

Esercizio 1 Calcolare la funzione derivata di $f(x)$ impiegando le funzioni per il calcolo simbolico di MATLAB:

$$f'(x) = \frac{df(x)}{dx}$$

e determinare il valore numerico assunto da tale funzione per $x = 5$.

Lo svolgimento di questo esercizio deve essere implementato per mezzo di una funzione MATLAB, chiamata `derivata_in_x`, che accetta in ingresso o la funzione e il valore 5, oppure solo il valore 5.

Possibile Soluzione

```
function [ risultato ] = derivata_in_x(fun, valore)
    syms x;
    derivata = diff(fun, x);
    disp(['Derivata pari a ' char(derivata)]);
    risultato = vpa(subs(derivata, valore), 3);
end
```

Esecuzione

```
>> fun = @(x) sqrt(4-x^2)
fun =      @(x) sqrt(4-x^2)
>> derivata_in_x(fun, .1)
Derivata pari a -x/(4 - x^2)^(1/2)
ans = -0.0501
```

Data la seguente funzione: $f(x) = \sqrt{4 - x^2}$

Esercizio 2 Determinare la funzione data dall'integrazione indefinita di $f(x)$:

$$g(x) = \int f(x)dx$$

e determinare la curva dei valori della funzione $g(x)$ nell'intervallo $[0, 2]$.

Lo svolgimento deve essere implementato per mezzo di una funzione MATLAB, chiamata `integrale_in_x`, che accetta in ingresso o la funzione e i valori 0 e 2, oppure solo i due valori.

Possibile Soluzione

```
function integrale_in_x(fun, valore1, valore2)
    syms x;
    integrale = int(fun, x);
    disp(['Integrale pari a ' char(integrale)]);
    if valore1 < valore2
        intervallo = valore1:.1:valore2;
    else
        intervallo = valore2:.1:valore1;
    end
    int_val = vpa(subs(integrale, intervallo), 3);
    plot(intervallo, int_val);
end
```

Data la seguente funzione: $f(x) = \sqrt{4 - x^2}$

Esercizio 2 Determinare la funzione data dall'integrazione indefinita di $f(x)$:

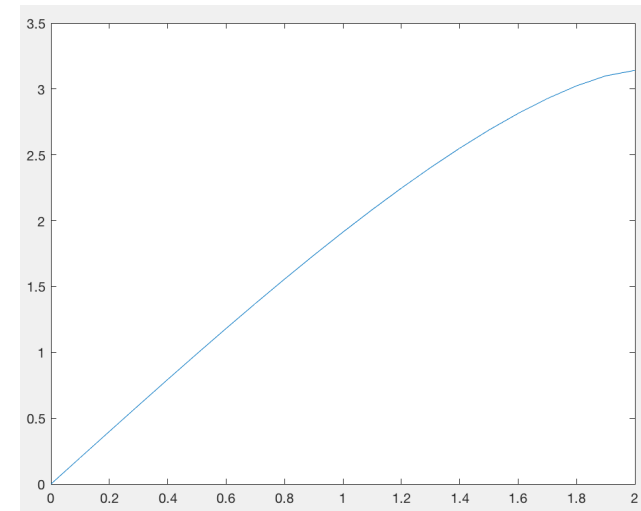
$$g(x) = \int f(x) dx$$

e determinare la curva dei valori della funzione $g(x)$ nell'intervallo $[0, 2]$.

Lo svolgimento deve essere implementato per mezzo di una funzione MATLAB, chiamata `integrale_in_x`, che accetta in ingresso o la funzione e i valori 0 e 2, oppure solo i due valori.

Possibile Soluzione

```
function integrale_in_x(fun, valore1, valore2)
    syms x;
    integrale = int(fun, x);
    disp(['Integrale pari a ' char(integrale)]);
    if valore1 < valore2
        intervallo = valore1:.1:valore2;
    else
        intervallo = valore2:.1:valore1;
    end
    int_val = vpa(subs(integrale, intervallo), 3);
    plot(intervallo, int_val);
end
```



Esecuzione

```
>> fun
fun = (4 - x^2)^(1/2)
>> integrale_in_x(fun, 0, 2)
Integrale pari a 2*asin(x/2) + (x*(4 -x^2)^(1/2))/2
```

Data la seguente funzione: $f(x) = \sqrt{4 - x^2}$

Esercizio 3 Ottenere le uscite della funzione nei seguenti punti [0 0,5 1 1,5 2], disegnare il diagramma xy di questi punti, ed effettuare la quadratura della funzione f(x) nell'intervallo [0 2].

Lo svolgimento di questo esercizio deve essere implementato per mezzo di una funzione MATLAB, chiamata `quadratura`, che accetta in ingresso o la funzione e i valori, oppure solo i cinque valori.

Possibile Soluzione

```
function quadratura(fun, valori)
    y = fun(valori);
    plot(valori, y);
    ris = quad(fun, valori(1), valori(length(valori)));
    disp(['Quadratura in [0 6] pari a ' mat2str(ris, 5)]);
end
```

Data la seguente funzione: $f(x) = \sqrt{4 - x^2}$

Esercizio 3 Ottenere le uscite della funzione nei seguenti punti [0 0,5 1 1,5 2], disegnare il diagramma xy di questi punti, ed effettuare la quadratura della funzione $f(x)$ nell'intervallo [0 2].

Lo svolgimento di questo esercizio deve essere implementato per mezzo di una funzione MATLAB, chiamata `quadratura`, che accetta in ingresso o la funzione e i valori, oppure solo i cinque valori.

Possibile Soluzione

```
function quadratura(fun, valori)
    y = fun(valori);
    plot(valori, y);
    ris = quad(fun, valori(1), valori(length(valori)));
    disp(['Quadratura in [0 6] pari a ' mat2str(ris, 5)]);
end
```

Esecuzione

```
>> fun = @(x) sqrt(4-x.^2)
fun =      @(x) sqrt(4-x.^2)
>> val = [0 .5 1 1.5 2];
>> quadratura(fun, val)
Quadratura in [0 6] pari a 3.1416
```

