

Lezione n° 2

Sommario:

- Determinante, rango e inversa di una matrice
- Interpretazione geometrica dei sistemi lineari
- Il calcolo simbolico _ esempi
- Spazi vettoriali (Combinazioni lineari, lineare dipendenza, basi) _ esempi

I sistemi lineari...

Per introdurre opportunamente i sistemi lineari, è doveroso richiamare i concetti di determinante, rango e inversa di una matrice. Discriminiamo i casi:

Matrici 1×1

Il determinante di una matrice 1×1 coincide con la costante stessa, cioè:

$$\det(c) = c \quad \forall c \in \mathbb{R}$$

Matrici 2×2

Il determinante di una matrice 2×2 , del tipo $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ vale:

$$\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc$$

Matrici $n \times n$

Prima di vedere come avviene il calcolo del determinante di una matrice $n \times n$, occorrono due definizioni.

Minore complementare: data una matrice $n \times n$, si definisce minore complementare dell'elemento di posto ij , e si indica con M_{ij} , il determinante della matrice che si ottiene cancellando l' i -esima riga e la j -esima colonna.

Complemento algebrico: data una matrice $n \times n$, si definisce complemento algebrico dell'elemento di posto ij il termine $C_{ij} = (-1)^{i+j} M_{ij}$, dove M_{ij} è il rispettivo minore complementare.

DETERMINANTE – RANGO - INVERSA

Il *determinante* di una matrice A quadrata di ordine $n \times n$ è definito dalla somma dei prodotti fra gli elementi di una qualunque riga, o colonna, per il rispettivi complementi algebrici.

Se ad esempio si scegliesse di sviluppare rispetto alla riga i -esima il determinante varrebbe:

$$\det(A) = a_{i1}C_{i1} + a_{i2}C_{i2} + \dots + a_{in}C_{in}$$

Se invece si scegliesse di sviluppare rispetto alla colonna j -esima il determinante varrebbe:

$$\det(A) = a_{1j}C_{1j} + a_{2j}C_{2j} + \dots + a_{nj}C_{nj}$$

N.B.: Il determinante non dipende dalla particolare riga o colonna che si sceglie per lo sviluppo.

Esempio: $B = \begin{pmatrix} 5 & 6 & 3 \\ 1 & 2 & 7 \\ 3 & 4 & 5 \end{pmatrix}$

$$\begin{aligned} \det(B) &= 5(-1)^{1+1}\det\begin{pmatrix} 2 & 7 \\ 4 & 5 \end{pmatrix} + 1(-1)^{2+1}\det\begin{pmatrix} 6 & 3 \\ 4 & 5 \end{pmatrix} + 3(-1)^{3+1}\det\begin{pmatrix} 6 & 3 \\ 2 & 7 \end{pmatrix} = \\ &= 5\det\begin{pmatrix} 2 & 7 \\ 4 & 5 \end{pmatrix} - \det\begin{pmatrix} 6 & 3 \\ 4 & 5 \end{pmatrix} + 3\det\begin{pmatrix} 6 & 3 \\ 2 & 7 \end{pmatrix} = \\ &= 5(2 \cdot 5 - 4 \cdot 7) - (6 \cdot 5 - 4 \cdot 3) + 3(6 \cdot 7 - 2 \cdot 3) = -90 - 18 + 108 = 0 \end{aligned}$$

Passiamo a Matlab!

```
>> A = [1 2 -1;3 0 1;4 2 1];  
>> d = det(A); %Determinante di una matrice
```

Definiamo *rango* (o *caratteristica*) di una matrice l'ordine del determinante più alto estraibile che sia diverso da zero.

```
>> r = rank(A); %Rango di una matrice
```

Se una matrice è quadrata, di ordine $n \times n$, ed esiste una matrice B tale che $AB = I$, si dice che B è l'*inversa* di A , e si scrive $B = A^{-1}$.

N.B.: Una matrice A $n \times n$ ammette un'inversa se e solo se $\det(A) \neq 0$, e in tal caso si dice *invertibile* (una matrice con determinante nullo si dice *singolare*). Quando una matrice ammette un'inversa, essa è unica. Dalla geometria:

$$A^{-1} = \frac{1}{\det(A)} M^T$$

dove M^T è la trasposta della matrice dei complementi algebrici, e presenta l'elemento di posto ij di M coincidente con il complemento algebrico dell'elemento di posto ij di A .

```
>> I = inv(A); %Inversa di una matrice  
>> J = A^-1; %Metodo alternativo per calcolare l'inversa
```

```
-----clear-----  
-----clc-----
```

Abbiamo già visto che si possono utilizzare i metodi della sostituzione e della combinazione lineare (o riduzione) per la risoluzione di un sistema di equazioni.

Se il sistema è lineare (cioè se le equazioni sono tutte di I° grado) si può utilizzare anche il metodo di **Cramer**, che consiste nel risolvere il sistema attraverso i valori dei determinanti delle matrici ottenute dai coefficienti e dai termini noti delle equazioni stesse.

In particolare é necessario trovare innanzitutto il valore del determinante della matrice dei coefficienti; se il sistema è di due equazioni in due incognite

$\begin{cases} ax + by = c \\ dx + ey = f \end{cases}$, allora ne vengono calcolate le incognite in questa maniera:

$$\rightarrow \text{la } x \text{ si calcola dal rapporto } \frac{\begin{vmatrix} c & b \\ f & e \end{vmatrix}}{\begin{vmatrix} a & b \\ d & e \end{vmatrix}}, \text{ mentre la } y \text{ da } \frac{\begin{vmatrix} a & c \\ d & f \end{vmatrix}}{\begin{vmatrix} a & b \\ d & e \end{vmatrix}}$$

Consideriamo il seguente sistema lineare $\mathbf{A} \mathbf{x} = \mathbf{b}$:

$$\begin{cases} 3x_1 + 5x_2 = 1 \\ x_1 + 2x_2 = 2 \end{cases}$$

```
>> A = [3 5;1 2];
```

```
>> B = [1 2]';
```

```
%In Matlab non c'è bisogno di calcolare il determinante di A, perché la
%risoluzione secondo Cramer non differisce dai metodi utilizzati nella
%riduzione
```

```
>> X = inv(A)*B; %Primo metodo di risoluzione di sistemi lineari
```

```
>> X1 = A\B; %Secondo metodo di risoluzione di sistemi lineari (già visto)
```

... e l'interpretazione geometrica

Ora vogliamo rappresentare il nostro sistema dal punto di vista geometrico:

```
>> x = [-50:50]; %Costruisco un range di valori di x
```

La grafica 2-D è introdotta dall'istruzione `plot(x,y)` :

```
>> plot(x, (1-3*x)/5, 'r') %Disegno la prima retta: 3x+5y=1 -> y=(1-3x)/5
```

N.B.: I vettori x e y devono avere le stesse dimensioni!

Attenzione alla sovrascrittura dei grafici!

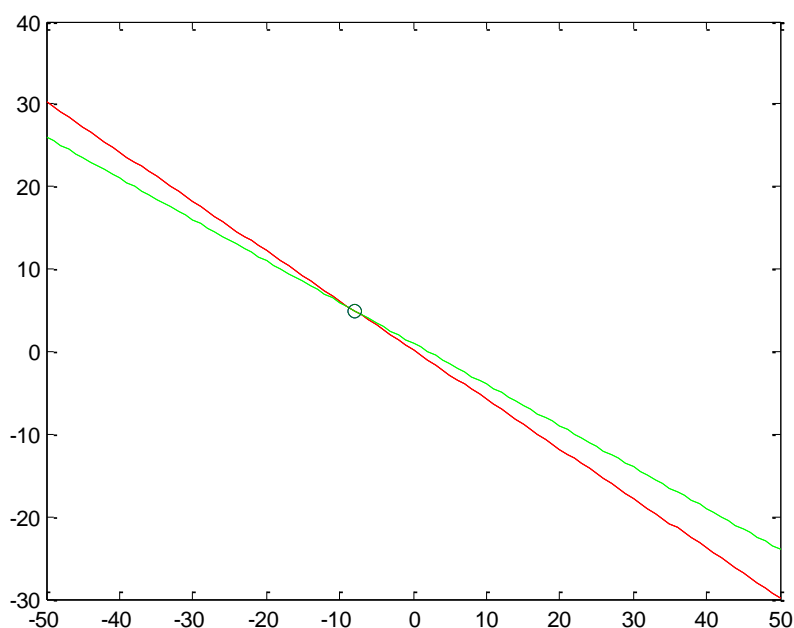
```
>> hold on %"hold on" congela ogni figura ad ogni plot
```

Quindi grafichiamo la seconda retta, senza sovrascrivere la prima:

```
>> plot(x, (2-x)/2, 'g') %Disegno la seconda retta: x+2y=2 -> y=(2-x)/2
```

Per selezionare e mettere in risalto il punto di intersezione tra le due rette si utilizza il comando `scatter`, che prende in ingresso l'ascissa e l'ordinata desiderate:

```
>> scatter(x(1,1), x(2,1)) %Marco la soluzione del sistema
```



OSSERVAZIONE

Matlab offre la possibilità di poter personalizzare i grafici nella maniera più disparata: si consulti il comando `plot` con l'help.

Ogni grafico nello stesso sistema di riferimento compare con un colore diverso: è possibile specificare per ogni grafico una stringa da uno a tre caratteri che suggerisce il colore, il tipo di simbolo da usare per i punti di tabulazione e il tipo di linea.

`plot(x1,y1,s1,x2,y2,s2,...)` con `s1` e `s2` stringhe di al più tre caratteri, date da **'colore, simbolo, tipo_linea'**; se si specifica solo il simbolo e non il tipo di linea, la spezzata non viene disegnata e solo i punti di tabulazione vengono segnati.

```
%Esempio di grafica personalizzata
```

```
>> plot(x, (1-3*x)/5, 'rd', x, (2-x)/2, 'g.--')
```

```
-----clear-----  
-----clc-----
```

```
%////////////////////////////////////  
% Risolvere e graficare  
% x1 - x2 +3x3 = 8  
% 2x1 - x2 +4x3 = 11  
% - x1 +2x2 -4x3 = -11  
%////////////////////////////////////
```

```
-----clear-----  
-----clc-----
```

```
%Eliminazione di Gauss-Jordan
```

```
>> A = [ 1 2 0; 4 3 -1; 5 5 -1];
```

```
>> C = [A [2 1 3]'];
```

Risolviamo il sistema:

```
>> R = rref(C) %Risoluzione di Gauss-Jordan
```

```
R =
```

```
1.0000    0   -0.4000   -0.8000  
    0    1.0000    0.2000    1.4000  
    0    0         0         0
```

L'obiettivo è, quindi, **risolvere il sistema lineare equivalente che ha come equazioni le righe non nulle di R!**

Se r è il rango della matrice R ridotta “a gradini” allora risolvo un sistema di Cramer in n incognite ed $n-r$ parametri (o *variabili libere*).

In questo caso $n=3$, $r=2$ quindi abbiamo una sola variabile libera x_3 .

Il problema si restringe al fatto di come poter scrivere dei sistemi a più variabili in Matlab, e poi racchiudere queste in un'unica variabile: la semplificazione, di tipo algebrico, viene risolta dal **calcolo simbolico**. Il concetto deriva dall'inconsistenza del sistema perché la soluzione si può effettuare anche con l'operatore “\”, ma non risulterà essere quella effettiva:

```
>> B = C(:,4); %Matrice dei termini noti
```

```
>> x = A\B;
```

```
Warning: Matrix is close to singular or badly scaled.  
Results may be inaccurate. RCOND = 3.469447e-018.
```

Nel caso di sistemi che ammettono infinite soluzioni l'operatore *backslash* si comporta in modo diverso a seconda delle situazioni: se la matrice è quadrata ci segnala il fatto che il determinante è nullo e non calcola nessuna soluzione; se invece la matrice è rettangolare ci segnala un'eventuale deficienza del rango rispetto al rango massimo possibile e poi calcola una sola soluzione del sistema.

Il pacchetto **Symbolic Math Toolbox** definisce un tipo di variabile differente da quelle viste in precedenza, che sono *double* e *char*.

Tale variabile non è di tipo numerico, e pertanto non è soggetta ad errori di approssimazioni da parte del calcolatore: il prezzo da pagare per ottenere questo ricade sulla velocità d'esecuzione e sulla necessità di particolari strutture dati.

Tramite il calcolo simbolico possiamo risolvere anche sistemi nei quali la soluzione dipende da uno o più parametri.

N.B.: Matlab, ovvero il calcolatore, non lavora con i numeri in sé, ma con una loro opportuna e definita conversione in relativi numeri binari, serie di 0 e 1 (il più piccolo numero utilizzabile da Matlab può essere visualizzato per mezzo del comando *eps*).

Detto ciò, il software commette un errore di approssimazione nella conversione di un numero (intero, razionale o reale che sia) in binario, e le operazioni che risultano possono essere condizionate da tale approssimazione: bisogna, quindi, fare attenzione quando si lavora con numeri dell'ordine di grandezza della precisione di macchina.

In alcuni casi anche l'utilizzo di *det* e *rank* può essere interpretato ed eseguito secondo un certo grado di approssimazione da parte del software: un determinante prossimo allo 0 è come se fosse nullo!

```
>> syms x3 %Dichiarazione di variabile simbolica
```

Nel Workspace →  x3 <1x1 sym>

```
%Risolvi un sistema A*X=B e selezioniamo le sottomatrici tali da poter  
%isolare la variabile libera dalle altre variabili
```

```
>> A1 = R(1:2,1:2);  
>> B1 = [-x3*R(1:2,3)+R(1:2,4)];
```

```
%Poichè in questo caso A1 è la matrice identità di dimensione 2 risulta che  
%X1==B1 (basta dividere B1 per l'inversa dell'identità, semplice  
%risoluzione di sistema X1 = A1\B1, e verificare l'uguaglianza logica)
```

In realtà il sistema di partenza aveva 3 incognite quindi dobbiamo aggiungere ad x1 anche la terza soluzione, che consiste in x3:

```
>> X = [B1;x3]
```

```
X =
```

```
(2*x3)/5 - 4/5  
7/5 - x3/5  
x3
```

```
-----clear-----  
-----clc-----
```



```

%Esempio di funzione che calcola Cramer per una matrice 3x3

%Nota: File->New->Script

%
                                Cramer3.m

function X = Cramer3(A,B)
%Risoluzione di un sistema lineare di 3 equazioni in 3 incognite con il Metodo
%di Cramer

X = []; %Inizializzo X come matrice vuota
a = det(A); %Calcolo il determinante di A

%Se il determinante di A è nullo la funzione si arresta
if (a==0)
    display('Errore! La regola di Cramer non è applicabile: det(A)=0');
    return %(return fa uscire dalla funzione); in caso contrario la funzione
        %prosegue
end

%Ricavo le tre matrici di cui poi dovrò calcolare il determinante
A1 = [B A(:,2:3)];
A2 = [A(:,1) B A(:,3)];
A3 = [A(:,1:2) B];
a1 = det(A1);
a2 = det(A2);
a3 = det(A3);

%Calcolo le xi come rapporto fra determinante di Ai e di A
x1 = a1/a;
x2 = a2/a;
x3 = a3/a;

%Costruisco il vettore X da fornire in uscita
X = [x1 x2 x3]';
end

```

```

-----clear-----
-----clc-----

```

```
%Esempio - Sistema impossibile
```

Si considera la stessa matrice A (di partenza) vista nell'esempio di Gauss-Jordan:

```
>> A = [ 1 2 0;4 3 -1;5 5 -1];
```

Segue una rappresentazione leggermente diversa del vettore dei termini noti:

```
>> B = [2 1 1]';
```

```
>> C = [A B];
```

Risolviamo il sistema, focalizzando l'attenzione sui ranghi della matrici ridotte a partire da A e da C !

```
>> Ra = rref(A);
```

```
>> rangoA = rank(Ra);
```

```
>> Rc = rref(C);
```

```
>> rangoC = rank(Rc);
```

Confronto i due ranghi calcolati:

```
>> rangoA == rangoC;
```

```
%Dal confronto logico si osserva che il sistema non è risolvibile, oppure si  
%può notare ciò semplicemente dall'ultima riga di Rc, che rappresenta  
%un'equazione impossibile
```

Si noti l'inconsistenza del sistema, verificandone banalmente la risoluzione anche con l'utilizzo dell'operatore “\”.

```
-----clear-----  
-----clc-----
```

```
%Esercizio 1 - Il curve fitting
```

```
%Determinare l'equazione del polinomio di grado 2 il cui grafico passa per i  
%seguenti 3 punti: (1,6) (2,3) (3,2)
```

Il **curve fitting** è il metodo per determinare una curva, che abbia la migliore corrispondenza ad una serie di punti assegnati, possibilmente soggetti a limitazioni.

Avendo a disposizione dei dati, che rappresentano solo alcuni valori di una funzione, possiamo considerare una approssimazione con un polinomio di un certo grado.

L'equazione che si cerca è del tipo: $y = a_0 + a_1x + a_2x^2$

Il curve fitting pretende di voler trovare il polinomio che passi esattamente per (1,6) (2,3) (3,2): si parla quindi di *interpolazione*, dove è richiesta un'esatta corrispondenza con i punti dati, e il grado del polinomio deve essere uguale al numero di punti meno uno.

Inoltre tutte le ascisse dei punti devono essere differenti per punti diversi, perché in corrispondenza di una ascissa non possiamo pretendere che il polinomio assuma due valori differenti).

```
%Soluzione
```

```
>> P = [1 6;2 3;3 2]; %Imponiamo il passaggio per i tre punti dati
```

```
>> for i=1:3, %Costruiamo la matrice dei coefficienti A  
A(i,:) = [1 P(i,1) P(i,1)^2];  
end
```

```
>> A; B = P(:,2); %N.B.: è possibile scrivere più istruzioni, se separate da  
%un ';' , sulla stessa linea di comando
```

```
%Abbiamo ricavato il sistema da risolvere: procediamo utilizzando Gauss-  
%Jordan o Cramer
```

```
>> C = [A B];
```

```
>> R = rref(C); %PRIMO METODO: Risoluzione con Gauss-Jordan
```

Poichè C ha rango 3, `rref` restituisce una matrice R composta dalla matrice identità nelle prime 3 colonne (matrice A del nuovo sistema), unita al vettore colonna delle soluzioni. Si nota che $X = \text{inv}(I) * B = B$ (dove $I = \text{eye}(3)$).

```
>> Xgj = R(:,4); %Soluzioni del sistema
```

```
>> Xc = Cramer3(A,B); %SECONDO METODO: Risoluzione con Cramer3
```

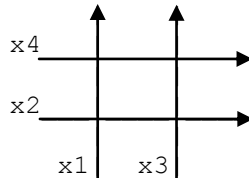
```
>> Xgj == Xc; %Uguaglianza fra matrici
```

```
-----clear-----  
-----clc-----
```

%Esercizio 2 - Traffico di una rete

%Consideriamo 4 strade a senso unico che si intersecano come da schema.
%Ogni intersezione rappresenta un incrocio.

%
%
%
%
%
%
%



Vogliamo costruire un modello matematico che analizzi il traffico della rete; l'unica regola è il principio di conservazione del flusso: il traffico entrante in un incrocio è pari al traffico uscente (si noti l'analogia con la legge di Kirchhoff per le correnti).

I dati ci vengono riportati in questo sistema di equazioni:

$$\begin{cases} x_1 + x_2 = 625 \\ x_1 + x_4 = 475 \\ x_3 + x_4 = 900 \\ x_2 + x_3 = 1050 \end{cases}$$

%Soluzione

```
>> C = [ %Matrice completa  
1 1 0 0 625;  
1 0 0 1 475;  
0 0 1 1 900;  
0 1 1 0 1050;  
1];
```

```
>> R = rref(C); %Risoluzione del sistema
```

Da $R=rref(C)$ risulta che $\text{rank}(R)=3$ quindi avremo una variabile libera x_4 : si arriverà alla soluzione attraverso il calcolo simbolico.

```
>>syms x4 %variabile libera
```

```
>> X = [-x4*R(1:3,4)+R(1:3,5);x4];
```

```
-----clear-----  
-----clc-----
```

SPAZI VETTORIALI

Una nuova visione, che permetterà di cogliere meglio il significato dei sistemi lineari $\mathbf{A} \mathbf{x} = \mathbf{b}$, è data dal concetto di *spazio vettoriale*, che ci permette di vedere le operazioni tra matrici e vettori, matrici e matrici, come operazioni su *vettori* e non come operazioni su “molti numeri”.

Definiamo lo **spazio vettoriale** \mathbb{R}^n come l'insieme di tutti i vettori colonna (riga) di n componenti reali.

Si definisce **sottospazio U di uno spazio vettoriale** \mathbb{R} un insieme di vettori (compreso il vettore nullo, $0 \in U$) che verificano le seguenti 2 proprietà:

1. $\forall x, y \in U \rightarrow x + y \in U$
2. $\forall x \in U, \forall \alpha \in \mathbb{R} \rightarrow \alpha x \in U$

Con questi presupposti, data una matrice A e considerati i vettori che formano le sue colonne, introduciamo lo *spazio delle colonne*.

Lo **spazio delle colonne**, $R(A)$, di una matrice A è dato dalle combinazioni lineari delle colonne di A . (*Tutti i possibili vettori Ax*).

È chiaro quindi che un sistema lineare $\mathbf{A} \mathbf{x} = \mathbf{b}$ ammette soluzione se e solo se il vettore $b \in R(A)$: che la soluzione sia poi unica, o meno, dipenderà da un altro spazio vettoriale caratterizzante la matrice A , in particolare

- Lo *spazio nullo*, $N(A)$, di una matrice A è dato da tutti i vettori x_n per cui $Ax_n = 0$.

COMBINAZIONI LINEARI, LINEARE DIPENDENZA, BASI

Dati m vettori colonna a n componenti, consideriamo la matrice A che ha come colonne i vettori dati.

→ Le colonne di A (gli m vettori dati) sono *linearmente indipendenti* se e solo se il sistema $\mathbf{A} \mathbf{x} = \mathbf{0}$ ammette la sola soluzione $\mathbf{x} = \mathbf{0}$. (*Nessuna altra combinazione lineare delle colonne di A può dare il vettore nullo*)

Per la definizione di $N(A)$ ne segue che se $N(A) \equiv 0$ allora le colonne di A sono linearmente indipendenti, mentre se $N(A) \neq 0$ le colonne sono *linearmente dipendenti*. Analogamente per i vettori riga (basta considerare la matrice A^T , dove A ha per righe i vettori dati).

Se si considerano tutte le possibili combinazioni lineari di m vettori dati si ottiene lo spazio generato dai vettori stessi (si pensi allo spazio delle colonne $R(A)$ di A).

Definiamo **base** di uno spazio vettoriale l'insieme dei vettori che godono delle seguenti due proprietà:

1. I vettori sono linearmente indipendenti
2. I vettori generano tutto lo spazio vettoriale

È importante osservare che uno stesso spazio vettoriale può essere generato da differenti basi, per esempio tutti i vettori di \mathbb{R}^2 si possono ottenere dalle 2 basi:

$$\begin{array}{ll} (1,0) \text{ e } (0,1) & \text{(assi cartesiani)} \\ (1,1) \text{ e } (-1,1) & \text{(assi ruotanti di } 45^\circ) \end{array}$$

ma un dato vettore è esprimibile, una volta fissata la base, in un unico modo. Per esempio il vettore $(2,4) = 2(1,0) + 4(0,1)$ oppure $(2,4) = 3(1,1) + 1(-1,1)$.

`% Esercizio 1 - lineare dipendenza`

Verificare se i seguenti tre vettori di \mathbb{R}^3

$$v_1 = [1 \ 2 \ 3] \quad v_2 = [-2 \ 1 \ 1] \quad v_3 = [8 \ 6 \ 10]$$

sono linearmente dipendenti o indipendenti

`%Soluzione`

`%Consideriamo l'equazione seguente`

$$\% \quad a(1,2,3) + b(-2,1,1) + c(8,6,10) = 0$$

`%con a,b,c coefficienti del sistema omogeneo che andiamo a rappresentare in
%forma matriciale`

`>> C = [[1 2 3]' [-2 1 1]' [8 6 10]' [0 0 0]'];`

`>> R = rref(C); %Risoluzione con Gauss-Jordan`

Si utilizza la variabile simbolica:

`>> syms x3`

`%Verifichiamo in seguito l'uguaglianza logica tra matrici; in questo caso
%vengono rispecchiate le notazioni richiamate nell'esempio del sistema
%ridotto con Gauss-Jordan per introdurre le variabili simboliche -> X1 == B1`

`>> R(1:2,1:2)\B1 == [-x3*R(1:2,3)+R(1:2,4)];`

`>> X = [-x3*R(1:2,3)+R(1:2,4);x3]; %Vettore colonna simbolico delle
%soluzioni ammesse`

Si conclude che i vettori sono linearmente dipendenti poiché il sistema ammette infinite soluzioni.

-----clear-----
-----clc-----

```
% Esercizio 2 - R^4
```

Determinare una base per il sottospazio di \mathbb{R}^4 generato dai seguenti vettori:

(1,2,3,4) (0,-1,2,3) (2,3,8,11) (2,3,6,8)

```
%Soluzione
```

```
%Prendo come matrice la matrice che ha per righe/colonne i vettori dati
```

```
>> C = [1 2 3 4; 0 -1 2 3; 2 3 8 11; 2 3 6 8];
```

Risolviamo il sistema:

```
>> R = rref(C);
```

La dimensione dello spazio generato coincide col rango quindi in questo caso equivale a 3, ovvero i vettori formano un sottospazio di dimensione 3; una base si può ottenere prendendo le prime tre righe di R. Quindi:

```
>> base = R(1:3,:);
```

```
-----clear-----  
-----clc-----
```

```
% Esercizio 3 - coefficienti di una combinazione lineare
```

Determinare se il vettore $(-1, 1, 5)$ è combinazione lineare dei seguenti tre vettori:

(1,2,3) (0,1,4) (2,3,6)

```
%Soluzione
```

```
%Dobbiamo scrivere il problema come un sistema lineare che ha per colonne  
%della matrice A dei coefficienti i tre vettori dati e come vettore B dei  
%termini noti il vettore da verificare
```

```
>> A = [[1 2 3]' [0 1 4]' [2 3 6]'];
```

```
>> B = [-1 1 5]';
```

```
>> C = [A B];
```

Risolviamo il sistema:

```
>> R = rref(C);
```

```
>> X = R(:,4); %Soluzione del sistema
```

Si osserva che il rango di R è 3 quindi i tre vettori costituiscono una base e il vettore dato è combinazione lineare degli altri 3 con coefficienti a, b, c corrispondenti agli elementi della soluzione X .

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Esercizi %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Dati i sistemi, determinarne le soluzioni tramite il metodo opportuno:
%
% 3*x1-x3 = -4
% 2*x2-x3 = -1
% -x1-x2+2*x3 = 3
%
% soluzione: X=[-1 0 1]'
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% -2*x1 +x3= 1
% 12*x1 -4*x2 +5*x3= 2
% -4*x1 +4*x2 -9*x3= 3
%
% soluzione: X=[x3/2 (11*x3)/4 x3]'
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Implementare una funzione Matlab che, dati come input la matrice dei
% coefficienti e il vettore dei termini noti,
% dia in output il vettore delle soluzioni e un messaggio di testo che
% indica il metodo utilizzato per trovare le soluzioni.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```