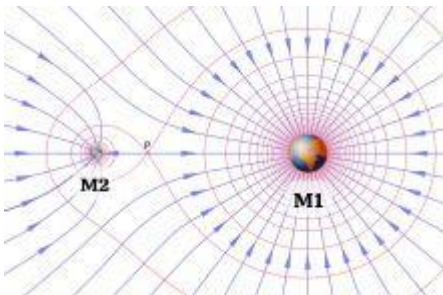


DISTRIBUZIONE DISCRETA E CARICHE

La **distribuzione discreta** possiede posizioni prefissate: ad esempio, le cariche sono localizzate in punti precisi nell'analisi microscopica all'interno dei tessuti (nel corpo umano). Basti pensare come all'interno dell'atomo le cariche positive siano localizzate all'interno del nucleo e le negative come orbite intorno allo stesso.

Il concetto di carica è stato elaborato come modello artificiale per spiegare determinati fenomeni naturali; nasce dunque la carica per **via empirica**, come l'intero elettromagnetismo [*Legge di Ampère e Legge di Gauss*].



FORZA GRAVITAZIONALE

Ogni corpo dotato di massa esercita una forza gravitazionale (attrattiva), dovuta al campo gravitazionale generato dal corpo stesso.

Il corpo di massa m_1 risente del campo gravitazionale provocato dal corpo di massa m_2 , e viceversa.

CAMPO SCALARE E CAMPO VETTORIALE

(Lavoriamo nel campo spaziale xyz)

Per **campo scalare** si intende la grandezza fisica rappresentabile mediante una funzione scalare (che necessita solo di un numero per esprimersi, come la temperatura T e la pressione P).

$$\Phi(x, y, z)$$

RICORDA CHE: In diagnostica una mappatura è una distribuzione di campi.

ESEMPIO CAMPO SCALARE:

$$\Phi(x, y, z) = x^2 + 2x$$

In questo esempio, il campo scalare Φ :

- Non dipende da z .
- Dando un punto P di valore x, y [$P(x, y)$], ritornerà il valore che il punto stesso assume nel campo scalare.

Per **campo vettoriale** si intende la grandezza fisica rappresentabile mediante una funzione vettoriale (un'informazione per ogni direzione d'intensità, necessita di un modulo, una direzione e un verso). *Ogni versore moltiplica la propria funzione scalare* (similare al campo scalare precedente).

$$\underline{A}(x, y, z) = \hat{x}A_x(x, y, z) + \hat{y}A_y(x, y, z) + \hat{z}A_z(x, y, z)$$

VETTORE

Un **vettore** è un ente matematico con un modulo, una direzione e un verso.

$$\underline{a} = \hat{x}a_x + \hat{y}a_y + \hat{z}a_z$$

Nei casi più semplici il modulo di un vettore si calcola:

$$|\underline{a}| = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

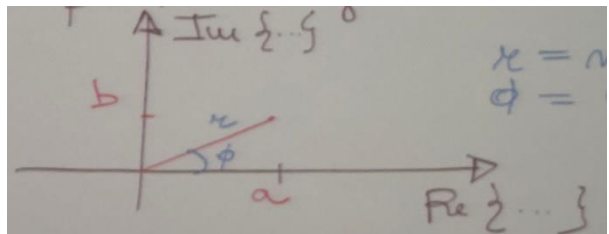
NUMERI COMPLESSI IN FISICA

I **numeri complessi** rappresentano una doppia informazione, si basano su una somma di una parte reale (a) e una parte immaginaria (b). $[j = \sqrt{-1}]$

$$\underline{z} = \underline{a} + j\underline{b}$$

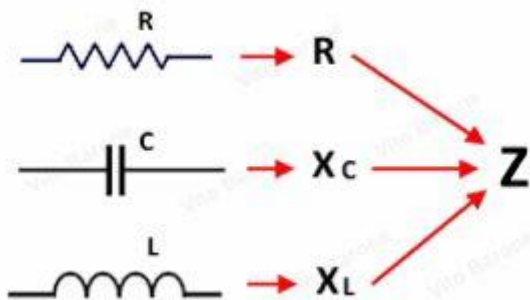
I numeri complessi possono essere espressi sotto forma di due forme: forma cartesiana e forma polare.

Nella **forma polare** \underline{z} è un vettore in un piano cartesiano immaginario. Inoltre \underline{z} possiede un modulo d'intensità r e crea con l'asse x un angolo Φ .



Mentre la **forma cartesiana** viene sfruttata in ambito circuitale, in particolare modo nei casi di impedenza.

L'**impedenza**, in elettrotecnica, è una grandezza fisica che rappresenta l'opposizione di un circuito al passaggio di una corrente elettrica alternata, o, più in generale, di una corrente variabile. È esprimibile come numero complesso ed è data dal rapporto tra tensione e corrente.



Esistono due tipi di impedenza:

1. **Impedenza reale**, con potenza attiva che colpisce la resistenza. L'energia viene dissipata;
2. **Impedenza immaginaria**, con potenza reattiva che colpisce induttori e condensatori. L'energia non verrà trasferita. I condensatori e gli induttori sono componenti reattive: i primi si occupano di immagazzinare energia di tipo elettrico, invece gli induttori immagazzinano energia di tipo magnetico.

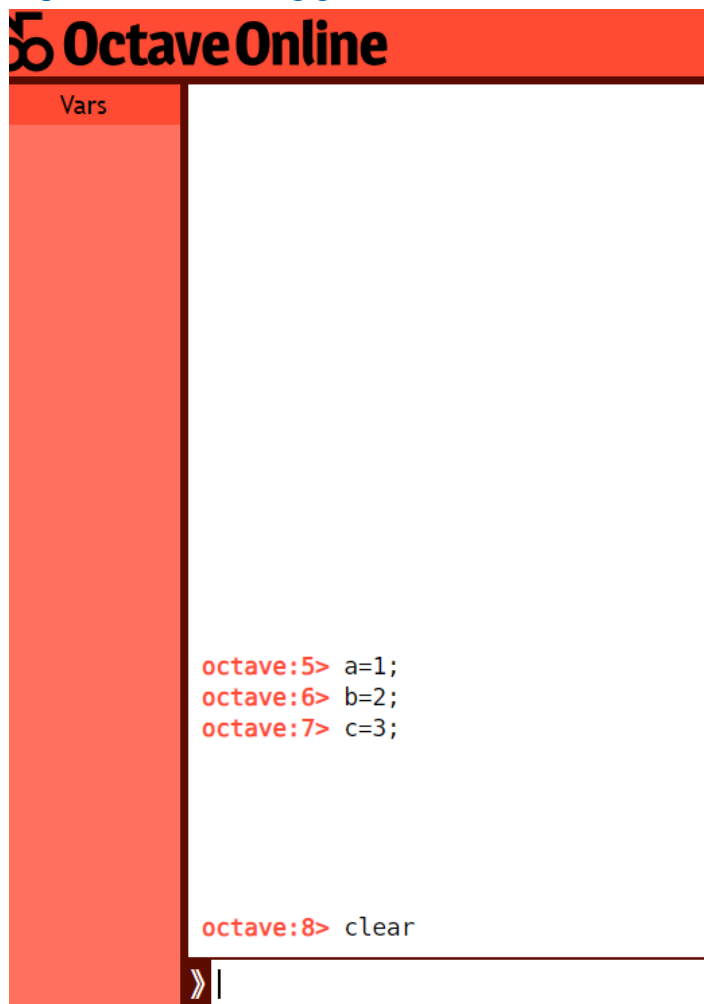
$$\underline{z} = \underline{R} + j\underline{w}$$

$[R = \text{resistore (componente reale)}; w = \text{impedenza}; L = \text{induttore (componente immaginaria)}]$

OCTAVE

I fisici e i matematici utilizzano per la graficizzazione dei dati il programma MATLAB. Per il nostro corso utilizzeremo la forma simile gratuita **Octave**, disponibile al seguente link:

<https://www.octave.org/get-involved.html>



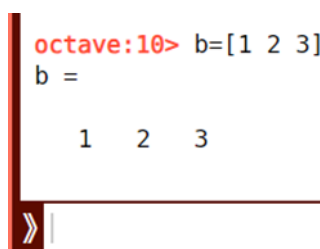
Octave è inoltre disponibile online ed è utilizzabile su ogni dispositivo.

I primi due comandi analizzati sono:

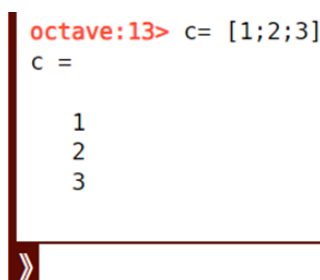
- **clc**: comando che “pulisce” (cela) la schermata;
- **clear**: comando che cancella ciò che si è definito precedentemente, ma non la schermata (da notare la colonna “Vars” vuota delle tre variabili dapprima impostate).

Per eliminare tutto (sia istruzioni che schermata) bisogna utilizzare in sequenza prima clear e poi clc.

L’aggiunta del punto e virgola (;) alla fine di un’istruzione, cela il risultato dell’istruzione stessa e permette anche di dichiarare più istruzioni in una sola riga di codice. Se si digita l’istruzione a=1 e **non** si inserisce il punto e virgola (;) si osserva il risultato dell’istruzione, con il punto e virgola, invece, si nasconde il risultato.



Per definire un **vettore riga** non si utilizzano virgole fra i valori nelle parentesi.



Per definire un **vettore colonna** si utilizza il punto e virgola (;) fra i valori nelle parentesi.

Con la funzione **linspace**(*START*, *END*, *N*) si crea una spaziatura lineare, dove i primi due numeri inseriti rappresenteranno il numero iniziale e finale del mio range, mentre l'ultimo i punti che dividono la mia linea temporale in maniera tale che la variazione tra due numeri consecutivi rimanga lineare (costante).

```
octave:18> t= linspace(0,10,11)
t =
    0    1    2    3    4    5    6    7    8    9   10
```

```
octave:23> t=0:0.5:10
t =
```

Columns 1 through 7:

```
    0    0.5000    1.0000    1.5000    2.0000    2.5000
3.0000
```

Columns 8 through 14:

```
    3.5000    4.0000    4.5000    5.0000    5.5000    6.0000
6.5000
```

Columns 15 through 21:

```
    7.0000    7.5000    8.0000    8.5000    9.0000    9.5000
10.0000
```

```
>> |
```

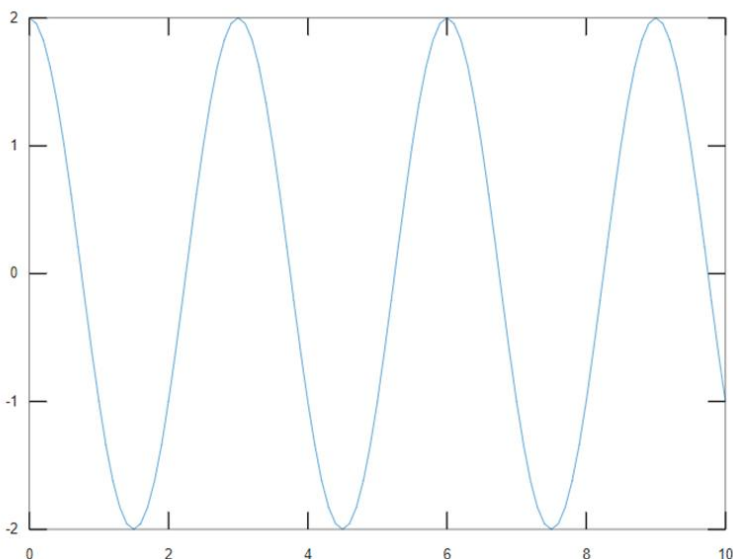
Per cancellare una variabile si utilizza la funzione **clear** *seguita da variabile*. (ES: clear t #cancello la variabile t).

Se si vuole invece definire la spaziatura, e dunque lasciare la scelta dei punti di divisione al sistema, basta dichiarare la variabile nella seguente modalità, **START:SPACE:END** dove i numeri agli estremi rappresentano il mio range e il mio passo (spaziatura) è il valore centrale.

Per definire una funzione variabile nel tempo, ad esempio una cosinusoidale, onde t è il vettore temporale, A è l'ampiezza, f la frequenza, π è definita e rappresenta il valore di 3.14, le **funzioni figure e plot** consentono di visualizzare l'onda.

```
octave:35> t=linspace(0,10,101);
octave:36> f=1/3;
octave:37> s=A*cos(2*pi*f*t);
octave:38> figure;
```

```
octave:39> plot(t,s)
```



LA MODULAZIONE

Modulare in fisica vuol dire variare in un certo modo voluto il valore di una definita grandezza.

La principale modulazione utilizzata è l'AM.

AM (amplitude modulation) è la modulazione in ampiezza, in cui si modifica l'ampiezza del segnale nel tempo e si associa ad ogni valore un certa approssimazione. Essa viene utilizzata in ambito biomedico per il tracciato dell'ECG.

$$s(t) = A \cos(2\pi f t)$$

La formula sovrastante descrive la **funzione non modulata** del tracciato, essa rappresenta una semplice sinusoide con una frequenza costante e un'ampiezza costante. Di seguito, invece, verrà riportata la **funzione modulata** del tracciato in cui si osserva una $s_m(t)$ che avrà un'ampiezza variabile in base all'andamento del segnale $A(t)$, ma manterrà la stessa frequenza del segnale:

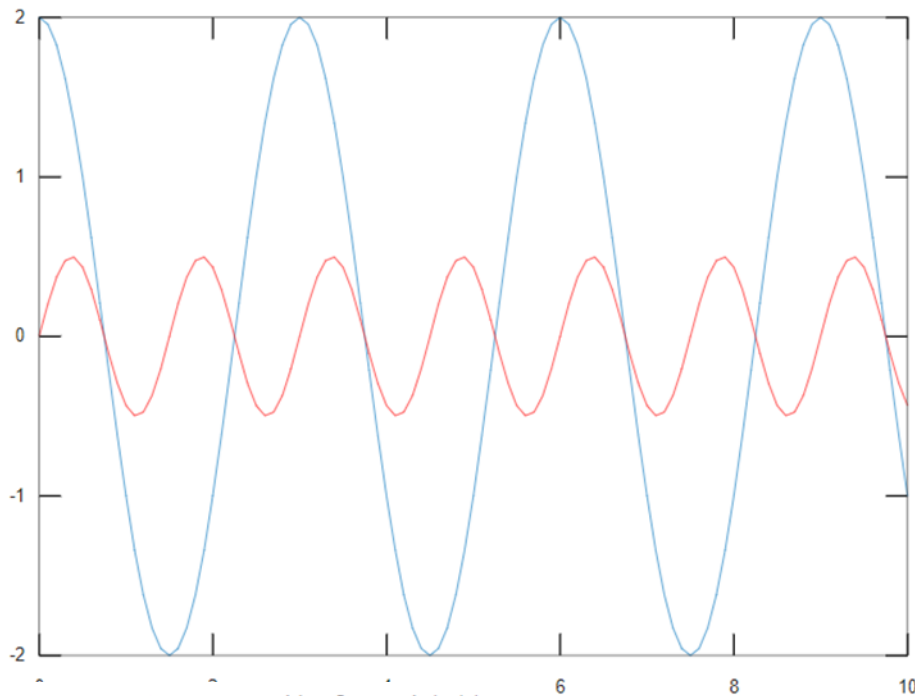
$$s_m(t) = A(t)\cos(2\pi ft)$$

L'ampiezza $A(t)$ nell'AM dipenderà dunque dal tempo in cui si è effettuata la modulazione.

```
octave:52> A=sin(2*pi*f*t);
octave:53> s=A.*cos(2*pi*f*t);
octave:54> sm=A.*cos(2*pi*f*t);
octave:55> s=2.*cos(2*pi*f*t);
octave:56> figure;plot(t,s)
```

A dunque diviene un vettore che può essere costituito da numeri che già sono conosciuti o con espressioni matematiche variabili. Per la moltiplicazione tra due vettori si utilizza “.”

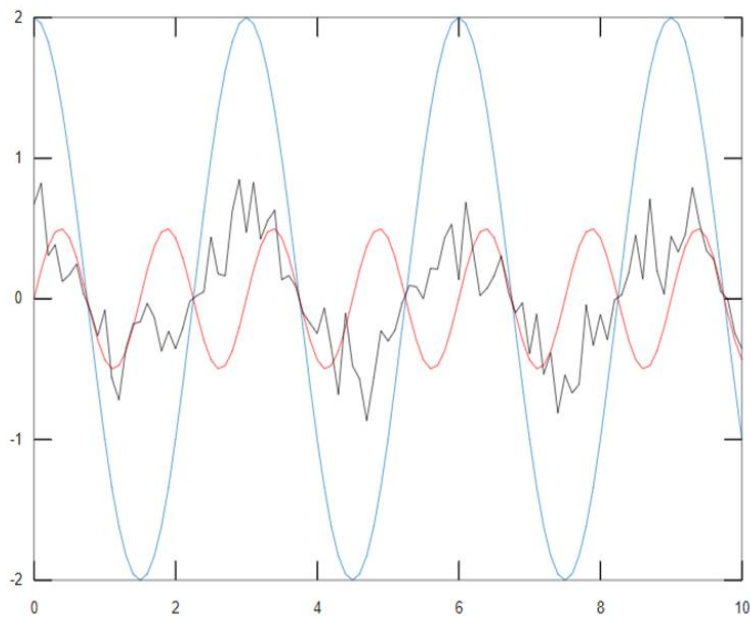
```
octave:57> hold on;plot(t,sm,'r')
```



La funzione **hold on** consente di sovrapporre due grafici e quella lettera fra virgolette (**'r'**) è semplicemente il colore che viene assegnato alla seconda onda.

```
octave:61> A=rand(1,length(t));
octave:62> smr=A.*cos(2*pi*f*t);
```

```
octave:66> hold on; plot(t, smr, 'k')
```



Per creare un vettore in maniera casuale basta utilizzare invece la funzione **rand**(*START*, *END*).

In questa ultima immagine, l'onda nera (*smr*) ha un andamento anomale che ricorda in molti aspetti un ECG.