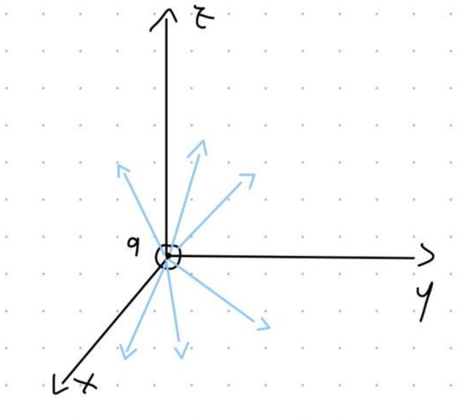
Elettromagnetismo e Teoria dei circuiti

Esercitazione su MATLAB: esempio monodimensionale e esempio bidimensionale del campo generato da una singola carica q

Prof. Sandra Costanzo – Lez.5 – 11/10/2023 - Autori: Rogato, Calisto - Revisionatori: Rogato, Calisto



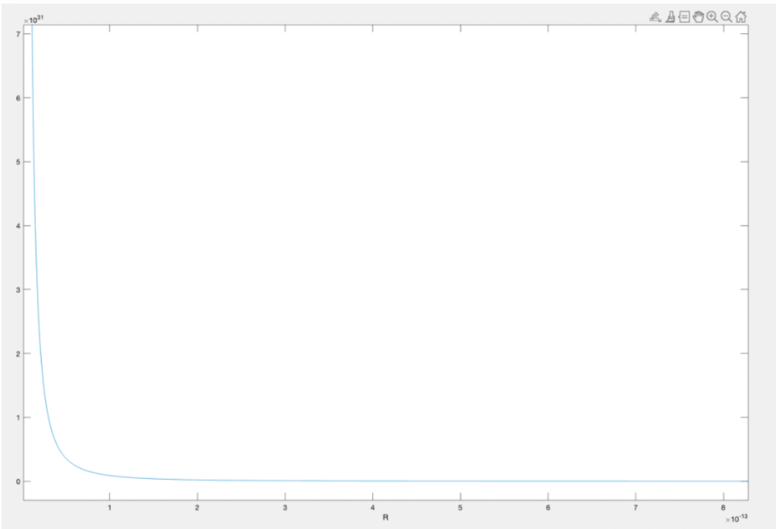
Supponiamo di essere nello spazio *xyz* con una carica positiva *q* che genera un **campo elettrico radiale** (perpendicolare alla carica), che assume in ogni punto dello spazio un valore con una direzione e un verso dettati dal versore $\hat{R}$ (<u>natura vettoriale</u>).

$$\underline{E} = \frac{1}{4\pi\varepsilon_0}\frac{q}{R^2}\hat{R}$$

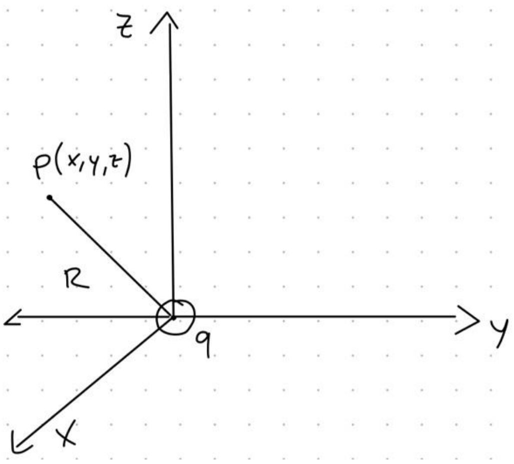
Supponiamo di essere nel vuoto (ε_0).

Esempio 1: graficazione monodimensionale rispetto a x

```
% campo elettrico generato da una carica posta nell'origine degli assi
|
epsilon_0 = 8.85*1e-12;              % costante dielettrica del vuoto
q = 1e-6;                            % quantità di carica positiva
R = linspace(1e-14, 1e-12, 1001);
y = 0;
z = 0;

E_modulo = (1/(4*pi*epsilon_0))*(q./(R.^2));
figure;
plot(R,E_modulo)
xlabel('R')                          % etichetta all'asse delle ascisse
```



Inseriamo *y=0* e *z=0* per supporre che la carica riesca a muoversi solo sull'*asse x*.
Si potevano scegliere anche altri valori.
Quando *R* è piccolissimo il campo procede verso infinito.

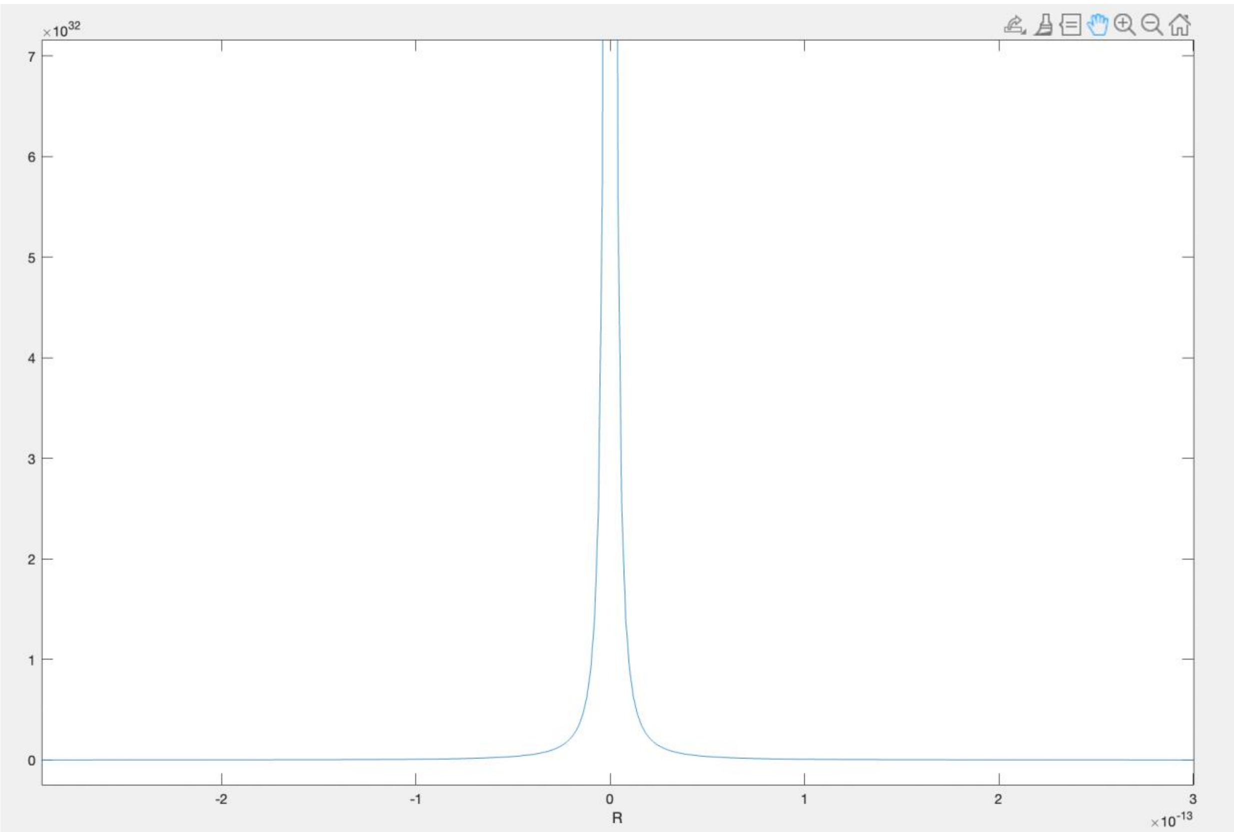


Per dividere le varie sezioni utilizziamo la doppia percentuale %%.
Dati x,y,z identifico in maniera inequivocabile un punto *P* nello spazio. La **distanza *R*** tra il punto e la carica sarà pari a:

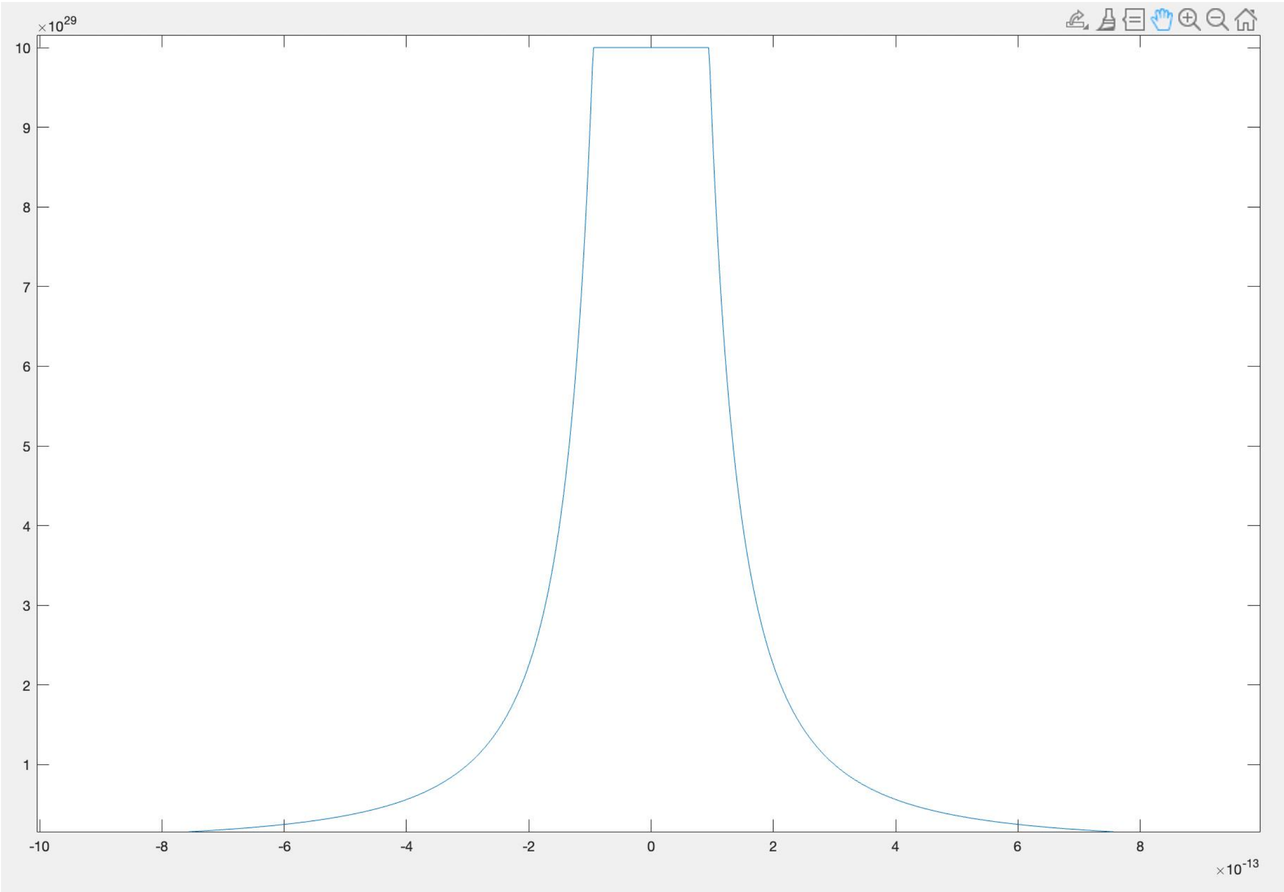
$$\boldsymbol{R = \sqrt{x^2 + y^2 + z^2}}$$

Possiamo dunque saturare il campo come artificio (non è la realtà), per prefissare un valore massimo di campo nonostante le variazioni di R

```
% saturazione del campo come artificio grafico

epsilon_0 = 8.85*1e-12;              % costante dielettrica del vuoto
q = 1e-6;                            % quantità di carica positiva
x = linspace(-1e-12, 1e-12, 1001);    % varia solo sull'asse x
y = 0;
z = 0;
E_modulo = (1/(4*pi*epsilon_0))*(q./(x.^2+y.^2+z.^2));

% x.^2+y.^2+z.^2 -- radice quadrata della distanza tra P e la carica

figure;
plot(x,E_modulo)
xlabel('R')                          % etichetta all'asse delle ascisse
```



```
% con il ciclo for si fa la saturazione (si chiudono le linee)

for n = 1:length(E_modulo)              % impongo un valore massimo
    if ge(E_modulo(n), 1e30)            % E_modulo(n) è >= a 1e30 ?
        E_modulo(n) = 1e30;             % se si, il massimo a cui arriva è 1e30
                                        % e riporta 1 come risposta affermativa
    end                                 % se no, esce 0
end
figure;
plot(x,E_modulo)
```



Il campo di una singola carica in corrispondenza della carica stessa tende all'infinito, pertanto, come mostrato nella figura sovrastante, abbiamo operato una saturazione per chiudere il grafico.

Con le funzioni ***min(x)*** e ***max(x)*** ritornano gli estremi dell'asse x.
Per scoprire il delta, ovvero la spaziatura costante tra i 1001 punti del nostro range basta porre la differenza tra due punti consecutivi:

x(b)-x(a), *a,b>0 e b>a*

Dettagli nella graficazione: sull'asse delle x si osserveranno dei punti che si collocheranno in corrispondenza dei valori di x e lo spazio tra due valori (tra due punti) corrisponde alla differenza tra di essi, ossia la regione in cui non si visualizza il campo. Perciò in questi punti non siamo a conoscenza dei valori del campo, però il programma è in grado di determinarlo in maniera approssimativa.

Esempio 2: graficazione bidimensionale dipendente da x e y

```
%Graficazione 2D del modulo del campo elettrico
epsilon_0=8.85*1e-12; %costante dialettrica nel vuoto
q=1e-6;
x=linspace(-1e-12, 1e-12,100);
y=linspace(-1e-12, 1e-12,100);
z=0;
[X,Y]=meshgrid(x,y);
Z=z

E_modulo=(1/(4*pi*epsilon_0))*(q./((X.^2+Y.^2+Z.^2)));
figure; surf(X,Y,E_modulo)
```

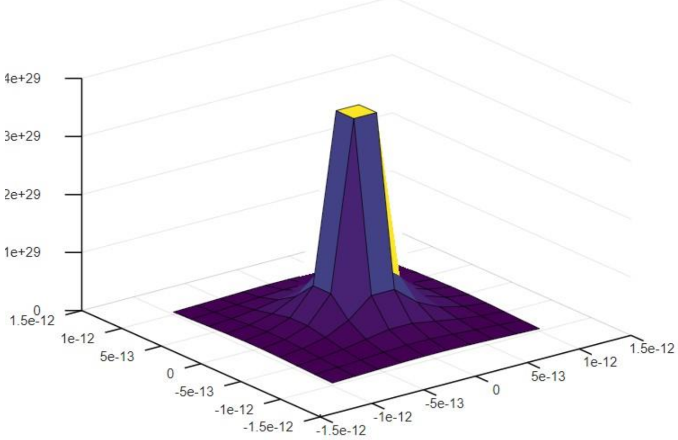
In questo esempio ci collochiamo sul piano x,y. Individuiamo la variazione di x e la variazione di y, il numero di punti in questo caso sarà 100 punti per x e 100 punti per y, in modo da ottenere una superficie pari a 100x100. Si definisce anche z pari a 0.
A questo punto si andrà a realizzare il grigliato [X,Y], in cui non si andranno a prendere tutti i punti.

Il grigliato si forma dalla funzione ***meshgrid***. Inoltre si osserva che *x* e *y* sono dei numeri mentre *X* e *Y* sono delle matrici.

Oltre a queste caratteristiche, si può far utilizzo della funzione ***size,*** ossia la dimensione per x, y, X e Y:

- size(x): ans= 1 100
- size(y): ans= 1 100
- size(X): ans=100 100
- size(Y): ans=100 100

Nei primi due casi (x e y) si parla di un ***vettore riga*** (prima riga e cento colonne). Negli altri due casi (X e Y) si parla di una ***matrice***, che identifica un punto specifico nello spazio.



Quello che si ottiene dal codice sovrastante e il seguente grafico il quale può essere visualizzato dal piano X-Y,Y-Z e X-Z.
Esempio nel campo medico in cui viene utilizzata questa tipologia di graficazione è la TAC bidimensionale.

Di seguito vengono riportati i passaggi della saturazione per tagliare e rendere "finito" il campo.

```
%artificio con saturazione
for n=1:size(E_modulo,1)
  for m=1:size(E_modulo,2)
    if ge(E_modulo(n,m),1e30)
      E_modulo(n,m)=1e30;
    endif
  endfor
endfor
figure;surf(X,Y,Z,E_modulo)
```

Sono presenti due cicli di for: il primo descrittivo delle righe e il secondo per le colonne.