Si richiede di calcare l'indice di condizionamento della valutazione della soluzione dell'equazione quadratica assegnata, mediante l'algoritmo $x_1 = -p + \sqrt{p^2 + q}$, p=10^5 e q =10^(-i), i=0,..,10

Dobbiamo quindi valutare l'indice di condizionamento del problema di valutare la funzione

$$f(q) = -p + \sqrt{p^2 + q}$$

L'indice di condizionamento del problema di valutare la f è dato da

$$K = \frac{|f'(q)| \ |q|}{|f(q)|}$$

Affinchè il problema sia ben condizionato, bisogna escludere i valori di q per cui f(q) tende a zero, cioè per cui $-p + \sqrt{p^2 + q}$ vada a zero.

Dai dati del problema risulta $p^2 = 10^{10} = 10\,000\,000\,000$

p^2 è contenuto nell'intervallo [2^33,2^34], cioè in [8 589 934 592, 17 179 869 184].

Lo spacing in [2^33,2^34] è dato da:

$$s = 2^{33+1-53} = 1.9073486328125e - 06$$

Per valori di q piccoli di $1.9073486328125e - 06$, il problema risulta mal condizionato.

Risultati in Python:

q= array([1.e+00, 1.e-01, 1.e-02, 1.e-03, 1.e-04, 1.e-05, 1.e-06, 1.e-07,
 1.e-08, 1.e-09, 1.e-10])

x= array([4.99999442e-06, 5.00003807e-07, 5.00003807e-08, 4.99130692e-09,
 4.94765118e-10, 4.36557457e-11, 1.45519152e-11, 0.00000000e+00,
 0.00000000e+00, 0.00000000e+00, 0.00000000e+00])

econda formula risolutiva

$x \quad = \frac{q}{p+\sqrt{p^2+q}}$, funzione di cui studiare l'indice di condzionamento $f(q) \ = \frac{q}{p+\sqrt{p^2+q}}$

$$K = \frac{|f \ '(q)| \ \ |q|}{|f \ \ (q)|}$$

In questo caso il problema risulta ben condizionato per tutti i valori di q assegnati, in quanto il denominatore non si annulla mai.

Risultati in python

```
array([5.e-06, 5.e-07, 5.e-08, 5.e-09, 5.e-10, 5.e-11, 5.e-12, 5.e-13,
    5.e-14, 5.e-15, 5.e-16])
```

```
# -*- coding: utf-8 -*-
"""
```

```
Created on Sat Jun  5 09:55:06 2021

@author: damia
"""
import numpy as np
p=10.0**5
q=10.0**(-np.arange(0.0,11.0))
x=-p+np.sqrt(p**2+q)


xa= q/(p+np.sqrt(p**2+q))
```

```
# -*- coding: utf-8 -*-
"""
Filtraggio di un segnale nel dominio di FOurier
"""

from scipy.fft  import fft, ifft
from scipy.fftpack import fftshift, ifftshift
import math
import numpy as np
import matplotlib.pyplot as plt



f= lambda x: np.sin(2*math.pi*5*x)+np.sin(2*math.pi*10*x)
noise= lambda x: 2*np.sin(2*math.pi*30*x)

T=2     #Durata del segnale
Fs=100  # Frequenza di campionamento nel dominio del tempo: Numero di campioni al secondo (maggiore uguale del doppio della freqeunza massima nel dominio delle frequenze
        #(wmax) presente nel segnale)
dt=1/Fs # Passo di campionamento nel dominio del tempo
N=T*Fs  #Numero di campioni: durata in secondi per numero di campioni al secondo

#Campionamento del dominio temporale
t=np.linspace(0,T,N)

#Campionamento del segnale rumoroso
y=f(t)+noise(t)
plt.plot(t,y,'r-')
plt.title('Segnale rumoroso')
plt.show()
plt.plot(t,f(t),'b-')
plt.title('Segnale esatto')
```

```
plt.show()


#Passo di campionamento nel dominio di Fourier (si ottiene dividendo per N l'ampiezza del range che contiene le frequenze)

delta_u=Fs/N

freq=np.arange(-Fs/2,Fs/2,delta_u)  #Il range delle frequenza varia tra -fs/2 ed fs/2

c= fftshift(fft(y))




plt.plot(freq,np.abs(c))

plt.title('Spettro Fourier segnale rumoroso')

plt.show()

ind= np.abs(freq)> 10.0


#Annulliamo i coefficienti di Fourier esterni all'intervallo di frequenze [-10,10]

c[ind]=0

plt.plot(freq,np.abs(c))

plt.title('Spettro Fourier segnale Filtrato')

plt.show()

#Ricostruiamo il segnale a partire dai coefficienti du Fourier filtrati

rec=ifft(ifftshift(c))

plt.plot(t,rec,t,f(t))

plt.legend(['Segnale filtrato', 'Segnale originale'])
```