Coherent states L-6.20

Πέτρος Πετρίδης

Α.Π.Θ.

Ιούνιος 2022

Χαμιλτονιανή αρμονικού ταλαντωτή.

Η χαμιλτονιανή κβαντικού αρμονικού ταλαντωτή σε 1D είναι:

$$\begin{aligned}
\hat{H} &= \frac{\hat{p}^2}{2m} + \hat{V} \\
&= \left(-\frac{\hbar}{\sqrt{2m}}\frac{d}{dx} + \sqrt{\frac{m}{2}}\omega_o x\right)\left(\frac{\hbar}{\sqrt{2m}}\frac{d}{dx} + \sqrt{\frac{m}{2}}\omega_o x\right) + \frac{\hbar\omega_o}{2} \\
&= \hbar\omega_o\left[\left(-i\sqrt{\frac{1}{2m\hbar\omega_o}}\hat{p} + \sqrt{\frac{m}{2\hbar}}\sqrt{\omega_o}x\right)\left(i\sqrt{\frac{1}{2m\hbar\omega_o}}\hat{p} + \sqrt{\frac{m}{2\hbar}}\sqrt{\omega_o}x\right) + \frac{1}{2}\right]
\end{aligned}$$

Ορίζονται οι τελεστές δημιουργείας και καταστροφής:

$$\hat{a}^+ = \frac{1}{\sqrt{2m\hbar\omega_o}}(-i\hat{p} + m\omega_o x)$$

και

$$\hat{a} = \frac{1}{\sqrt{2m\hbar\omega_o}}(i\hat{p} + m\omega_o x)$$

άρα:

$$\hat{H} = \hbar\omega_o\left(\hat{a}^+\hat{a} + \frac{1}{2}\right)$$

Ορισμός coherent states.

Coherent state είναι το κβαντομηχανικό αντίστοιχο του μονοχρωματικού κλασικού ηλεκτρομαγνητικού κύματος και συμβολίζεται ως $|\alpha >$ και είναι ίσο με:
$|\alpha >= e^{-|a|^2/2} \sum_{n=0}^{\infty} \frac{a^n}{\sqrt{n!}} |n >.$
Όπου $|n >$ οι ιδιοκαταστάσεις του αρμονικού ταλαντωτή.

Χρονική εξέλιξη coherent states.

Για αρμονικό ταλαντωτή $E_n = \hbar\omega_o(n + 1/2)$.
$|\alpha, t >= T(t)|\alpha >= e^{-iE_n t/\hbar}|\alpha >= e^{-|a|^2/2} \sum_{n=0}^{\infty} e^{-i\omega_o(n+1/2)t} \frac{a^n}{\sqrt{n!}} |n >$

όπου $|n >= (\frac{m\omega_o}{\hbar\pi})^{1/4} \frac{1}{\sqrt{2^n n!}} H_n(\sqrt{\frac{m\omega_o}{\hbar}} x) e^{-\frac{m\omega_o}{2\hbar} x^2}$
οι ιδιοκαταστάσεις του αρμονικού ταλαντωτή.

Πολυόνυμα Hermite:
$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n} e^{-x^2}$
όπου ισχύει:
$H_0(x) = 1$
$H_1(x) = 2x$
$H_n(x) = 2xH_{n-1}(x) - 2(n-1)H_{n-2}(x), n \geq 2$

Υπολογισμός των μέτρων συναρτήσει του χρόνου.

Θεώρηση $m = \hbar = \omega_o = k = 1$:

$|a, t >= e^{-|a|^2/2} \sum_{n=0}^{\infty} e^{-i(n+1/2)t} \frac{a^n}{\sqrt{n!}} \frac{1}{\sqrt{2^n n! \sqrt{\pi}}} H_n(x) e^{-\frac{x^2}{2}}$

Σύμφωνα με την εκφώνηση:
$E_a = |a|^2 + 1/2$
Οπότε $|a|^2 = E_a - 1/2$.

Κώδικας.

```
from numpy import *
import numpy as np , matplotlib.pyplot as plt, math
```

Ορισμός σταθερών:

```
sqpi=np.sqrt(np.pi)#ρίζα π
E=3.0; alpha=np.sqrt(E-0.5) #|α|
factr=np.exp(0.5*alpha*alpha); nmax=20
```

Ορισμός συνάρτησης που υπολογίζει τα πολυόνυμα Hermite:

```
def Hermite(x,n):
    if(n==0): #Μηδενικού βαθμού
        p=1.0
    elif(n==1): #Πρώτου βαθμού
        p=2*x
    else: #Βαθμού >=2
        p0=1
        p1=2*x
        for i in range(1,n):
            p2=2*x*p1-2*i*p0
            p0=p1
            p1=p2
            p=p2
    return p
```

Κώδικας.

$$|a,t>=e^{-|a|^2/2}\sum_{n=0}^{\infty}e^{-i(n+1/2)t}\frac{a^n}{\sqrt{n!}}\frac{1}{\sqrt{2^n n!\sqrt{\pi}}}H_n(x)e^{-\frac{x^2}{2}}$$

```
def glauber(x,t,nmax):
    Reterm=0.0
    Imterm=0.0

    factr=np.exp(-0.5*alpha*alpha) #exp(-|α|^2/2)

    for n in range(0,nmax+1):
        fact=np.sqrt(1.0/(math.factorial(n)*sqpi*(2**n))) #1/ρίζα[2^n * n! * ρίζα(π)] = c2
        psin=fact*Hermite(x,n)*np.exp(-0.5*x*x) #c2 * Hn(x) * exp(-x^2/2)

        den=np.sqrt(math.factorial(n)) # ρίζα(n!)

        num=factr*(alpha**n)*psin #c1 * a^n * c2 * Hn(x) * exp(-x^2/2)

        #To exp[-i(n+1/2)]=cos(n+1/2) -isin(n+1/2)
        Reterm+=num*(np.cos((n+0.5)*t))/den
        Imterm+=num*(np.sin((n+0.5)*t))/den

        #|Ψ|^2=Re^2+Im^2
        phi=Reterm*Reterm+Imterm*Imterm
    return phi
```

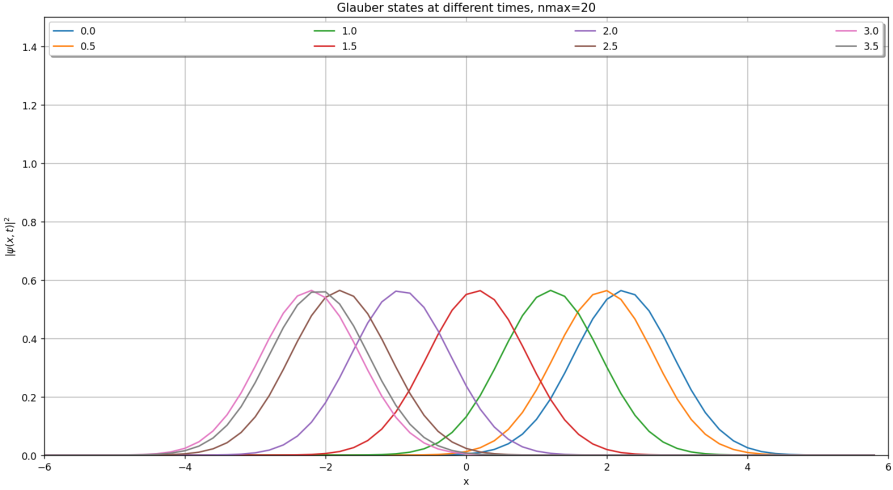
Κώδικας.

Η συνάρτηση animate, δέχεται την παράμετρο t, που είναι κάποια χρονική στιγμή και υπολογίζει την πιθανότητα του |a,t>, για κάθε θέση x από τον array xx.

```
#plot |Ψ|^2 για δεδομένα x & t
def animate(t):
    y=glauber(xx,t,nmax)
    s=str(t)
    plt.plot(xx,y,label=s)
    leg=plt.legend(loc="best",ncol=4,mode="expand",shadow=True)


fig=plt.figure()
ax=fig.add_subplot(111,autoscale_on=True,xlim=(-6,6),ylim=(0,1.5))
ax.grid()


plt.title(f"Glauber states at different times, nmax=%d" %nmax)
plt.xlabel("x")
plt.ylabel("$|\psi(x,t)|^2$")

xx=np.arange(-6.0,6.0,0.2)

for t in np.arange(0,3.6,0.5):
    animate(t)

plt.show()
```

Πιθανότητα για nmax=20, $t \in [0, 3.5]$ seconds.

Με τον έλεγχο των πιθανοτήτων φαίνεται και η ταλάντωση της πιθανότητας γύρω από την "θέση ισορροπίας" x=0.



Τέλος παρουσίασης.