Harmonic oscillator wave functions L-6.1

Πέτρος Πετρίδης

Α.Π.Θ.

Ιούνιος 2022

Χαμιλτονιανή αρμονικού ταλαντωτή.

Η χαμιλτονιανή κβαντικού αρμονικού ταλαντωτή σε 1D είναι:
$\hat{H} = \frac{\hat{p}^2}{2m} + \hat{V} = -\frac{\hbar^2}{2m}\frac{d^2}{dx^2} + \frac{1}{2}m\omega_o^2 x^2$

Προκύπτει πως οι ενεργειακές καταστάσεις είναι: $E_n = (n + 1/2)\hbar\omega_o$.

Άρα:

$$H\Psi = E_n\Psi \Rightarrow$$

$$-\frac{\hbar^2}{2m}\frac{d^2\Psi}{dx^2} + \frac{1}{2}m\omega_o^2 x^2\Psi = (n + 1/2)\hbar\omega_o\Psi \Rightarrow$$

$$\frac{d^2\Psi}{dx^2} + (2n + 1 - x^2)\Psi = 0,\ \frac{m\omega_o}{\hbar} = 1,\ n = 0, 1, 2, 3...$$

Αριθμητική επίλυση διαφορικής εξίσωσης.

Η διαφορική:
$\frac{d^2\Psi}{dx^2} + (2n+1-x^2)\Psi = 0 \Rightarrow$
$y''(x) + (2n+1-x^2)y(x) = 0$

ανάγεται σε δύο διαφορικές πρώτης τάξης αν:
$y'(x) = z$
$z'(x) = -(2n+1-x^2)y(x)$

Δεξί μέλη εξισώσεων.

```
import numpy as np, matplotlib.pylab as plt
```

$y'(x) = z(x)$
$z'(x) = -(2n + 1 - x^2)y(x)$

όπου:
$y(x_n) = y_n = y[0]$
$y'(x_n) = y[1]$

Χρησιμοποιείται για να επιστρέψει τα δεξιά μέλη των δ.ε.

```
def f(x,y):
    fVec[0]=y[1]
    fVec[1]=-(2*n+1-x**2)*y[0]
    return fVec
```

επιστρέφει [y’(x),y”(x)]

Runge-Kutta 4.

Είναι:

$$\begin{aligned}
\mathrm{k}_1 &= hy'_n & \mathrm{l}_1 &= -h(2n+1-x_n^2)y_n\\
\mathrm{k}_2 &= h(y'_n + l_1/2) & \mathrm{l}_2 &= -h(2n+1-(x_n+h/2)^2)(y_n+k_1/2)\\
\mathrm{k}_3 &= h(y'_n + l_2/2) & \mathrm{l}_3 &= -h(2n+1-(x_n+h/2)^2)(y_n+k_2/2)\\
\mathrm{k}_4 &= h(y'_n + l_3) & \mathrm{l}_4 &= -h(2n+1-(x_n+h)^2)(y_n+k_3)
\end{aligned}$$

Αυτό έχει: $k_1 = [k_1, l_1]$, $k_2 = [k_2, l_2]$,$k_3 = [k_3, l_3]$, $k_4 = [k_4, l_4]$.

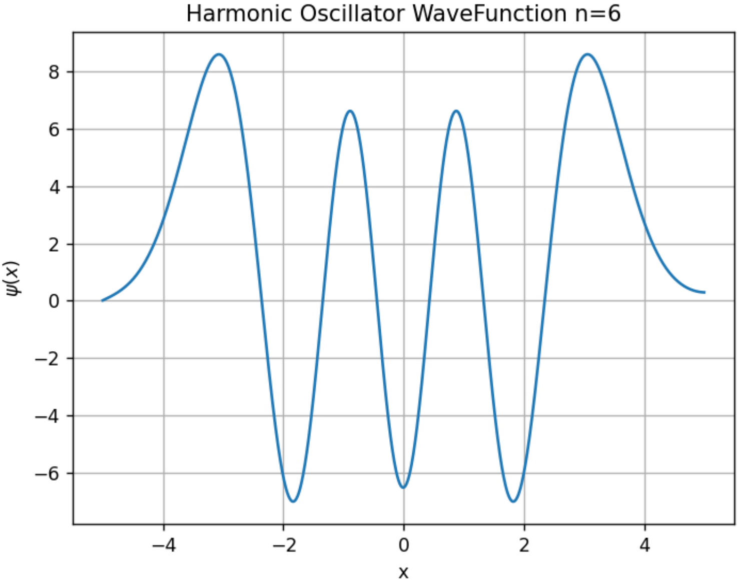
```
def rk4Algor(t,h,N,y,f):
        k1=np.zeros(N)
        k2=np.zeros(N)
        k3=np.zeros(N)
        k4=np.zeros(N)
        k1 = h*f(t,y)
        k2 = h*f(t+h/2.,y+k1/2.)
        k3 = h*f(t+h/2.,y+k2/2.)
        k4 = h*f(t+h,y+k3)
        y=y+(k1+2*(k2+k3)+k4)/6.
        return y
```

επιστρέφει: $[y(x+h), y'(x+h)]$.

RK4 για $x \in [-5, 5)$ $n = 6$.

```
rVec=np.zeros((1000),float)
psiVec=np.zeros((1000),float)
fVec=np.zeros(2)
y=np.zeros((2))
n=6

if (n%2==0): y[0]=1e-8
else: y[0]=-1e-8
y[1]=1; i=0
f(0.0,y)
dr=0.01
for r in np.arange(-5,5,dr):
    rVec[i]=r
    y=rk4Algor(r,dr,2,y,f)
    psiVec[i]=y[0]
    i=i+1

plt.figure()
plt.plot(rVec,psiVec)
plt.grid()
plt.title("Harmonic Oscillator WaveFunction n=6")
plt.xlabel("x")
plt.ylabel("$\psi(x)$")
plt.show()
```

RK4 για $x \in [-5, 5)$ $n = 6$.

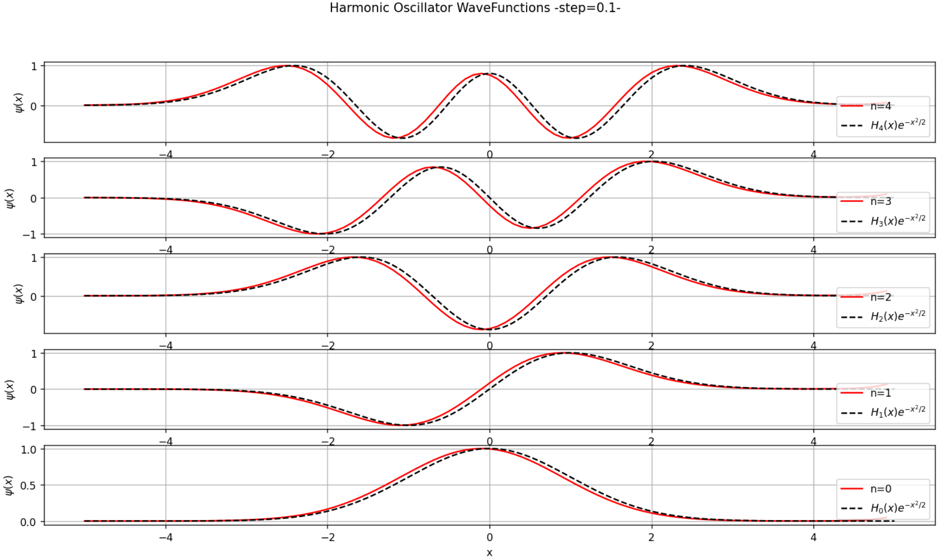


Σύγκριση ιδιοσυναρτήσεων με $H_n(x)e^{-x^2/2}$.

```
rVec=np.zeros((1000),float);psiVec=np.zeros((1000),float)
fVec=np.zeros(2);y=np.zeros((2))

ns=[0,1,2,3,4]
fig,ax=plt.subplots(len(ns))
j=len(ns)-1
for n in ns:
    if (n%2==0):
        y[0]=10**-8
        y[1]=10**-8
    else:
        y[0]=-10**-8
        y[1]=-10**-8

    i=0
    f(0.0,y)
    dr=0.01

    for r in np.arange(-5,5,dr):
        rVec[i]=r
        y=rk4Algor(r,dr,2,y,f)
        psiVec[i]=y[0]
        i=i+1

    psiVec=psiVec/max(abs(psiVec))

    ax[j].plot(rVec,psiVec, label="n=%d"%n,color="red")
    ax[j].grid()

    hermite=s.hermite(n,monic=False)
    rng=np.linspace(-5,5,1000)

    ax[j].plot(rng,hermite(rng)*np.exp(-rng*rng/2)/max(abs(hermite(rng)*np.exp(-rng*rng/2))),
                "k--", label="$H_%d(x)e^{-x^2/2}$"%n)
    ax[j].legend(loc="lower right")
    j-=1
fig.suptitle("$\Psi_n(x)$"" vs ""$H_n(x)$")
for axis in ax.flat:
    axis.set(xlabel="x", ylabel="$\psi(x)$")

plt.show()
```

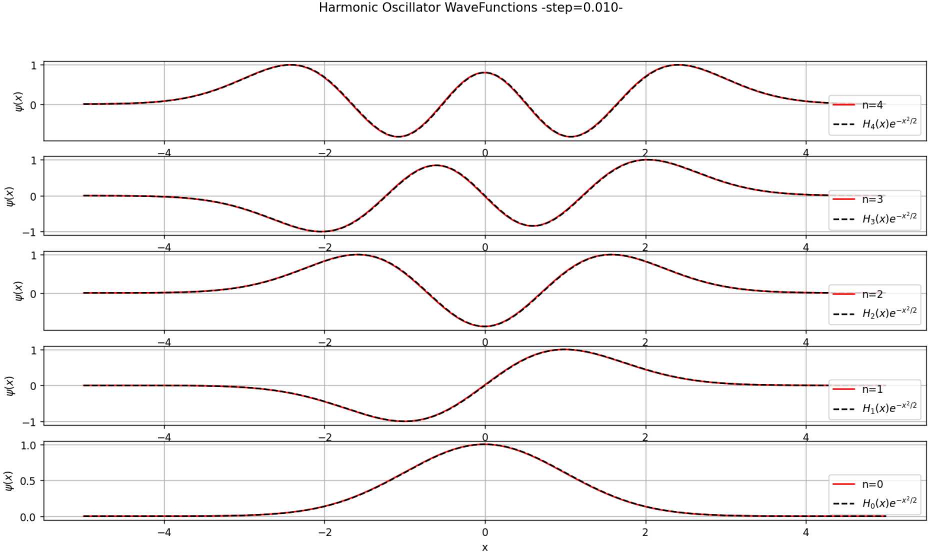
Σύγκριση ιδιοσυναρτήσεων με $H_n(x)e^{-x^2/2}$.

Βήμα Runge-Kutta 4 ίσο με 0.1



Σύγκριση ιδιοσυναρτήσεων με $H_n(x)e^{-x^2/2}$.

Βήμα Runge-Kutta 4 ίσο με 0.01



ΤΕΛΟΣ ΠΑΡΟΥΣΙΑΣΗΣ