harmonic-oscillator-1d

February 28, 2024

[1]:
```python
import scipy
import sympy
from sympy import Function, Symbol, Derivative, diff, integrate, Piecewise
from sympy import sin, pi, sqrt, oo
from sympy.physics.quantum.constants import hbar
from sympy.physics.quantum.operator import DifferentialOperator, Operator
from sympy.physics.quantum.state import Wavefunction
from sympy.physics.quantum.qapply import qapply

import numpy as np
from matplotlib import pyplot as plt
```

[2]:
```python
# Define number of basis
nbasis = 30

# Construct the basis (eigenfuncitons)
x = Symbol('x')
n = Symbol('n')
m = Symbol('m')
ma = hbar**2/2.0
w = sympy.sqrt(1./2. / ma)
L = 20.0

g = Piecewise((0, x < 0), (0, x > L), (sqrt(2./L)*sin(n*pi*x/L), True))
# g = sqrt(2./L)*sin(n*pi*x/L)
phin = Wavefunction(g, x)
phim = phin.subs(n, m)

f = Function('f')
T = DifferentialOperator(-hbar**2/(2*ma) * Derivative(f(x), x, x), f(x))
V = 1./2. * ma * (w**2) * (x - L/2.)**2
```

[3]:
```python
H = T + V
```

[4]:
```python
H
```

[4]:

$$25.0\,(0.1x-1)^2 + DifferentialOperator\left(-1.0\frac{d^2}{dx^2}f(x), f(x)\right)$$

[5]:
```
(phim.expr*(qapply(T * phin).expr)).simplify()
```

[5]:

$$\begin{cases} 0 & \text{for } x > 20.0 \vee x < 0 \\ 0.00025\pi^2 n^2 \sin(0.05\pi m x) \sin(0.05\pi n x) & \text{otherwise} \end{cases}$$

[6]:
```
hmn = (phim.expr*(V * phin.expr + qapply(T * phin).expr)).simplify()
# Exact expression of the integrand hmn, needing to be integrated from -oo to oo
hmn
```

[6]:

$$\begin{cases} 0 & \text{for } x > 20.0 \vee x < 0 \\ \left(0.00025\pi^2 n^2 + 2.5\,(0.1x - 1)^2\right) \sin(0.05\pi m x) \sin(0.05\pi n x) & \text{otherwise} \end{cases}$$

[7]:
```
from sympy import N
tmn = (phim.expr*(qapply(T * phin).expr)).simplify()
N(integrate(tmn.evalf(subs={m: 1, n: 5}), (x, -oo, oo)))
```

[7]:

0

[8]:
```
# Sanity check for h11
integrate(hmn.evalf(subs={m: 1, n: 1}), (x, 0, L))
```

[8]:

$$8.35800734433605 - \frac{50.0}{\pi^2}$$

[9]:
```
from tqdm.auto import tqdm
# Construct the Hamitonian matrix elements
# matrix_elements = []
Hmn = np.zeros((nbasis, nbasis))
for i in tqdm(range(1, nbasis+1)):
    row = []
    for j in range(1, nbasis+1):
        # Substitute m and n values into hmn and integrate
        element = integrate(hmn.evalf(subs={m: i, n: j}), (x, 0, L))
        Hmn[i-1, j-1] = N(element)
        row.append(element)
    # matrix_elements.append(row)
```

```
/home/cyrus/miniconda3/envs/dev-py311/lib/python3.11/site-
packages/tqdm/auto.py:21: TqdmWarning: IProgress not found. Please update
jupyter and ipywidgets. See
https://ipywidgets.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm
100%|                                    | 30/30
[13:29<00:00, 26.99s/it]
```

[10]:
```
# Trim out the contant basis function (n=0)
Hmn_ = Hmn#[1:,1:]
```

[11]:
```python
print(Hmn_.shape)
```

```
(30, 30)
```

[12]:
```python
import scipy.linalg as LA
from sympy import lambdify

# Solve eigenvalues and eigenvectors

eigvalues, eigvectors = LA.eigh(Hmn_)

phi_nx = lambdify((x, n), phin.expr, "numpy")
```

[13]:
```python
# Some sanity check for the coefficient of the first basis function
import math
math.sqrt(2/L)
```

[13]: 0.31622776601683794

[14]:
```python
phin.expr
```

[14]:
$$\begin{cases} 0 & \text{for}\: x > 20.0 \vee x < 0 \\ 0.316227766016838 \sin{\left(0.05 \pi n x \right)} & \text{otherwise} \end{cases}$$

[15]:
```python
# Sort eigenvalues in ascending order
indices = np.argsort(eigvalues)
first = 7
print(eigvalues[indices[:first]])
```

```
[0.5 1.5 2.5 3.5 4.5 5.5 6.5]
```

[16]:
```python
import numpy as np

# Check orthonormality
cmn = eigvectors[:,indices].T @ eigvectors[:,indices]
print(cmn.diagonal())
print(cmn.diagonal(offset=1))
print(np.allclose(cmn.diagonal(offset=1), 0))
```

```
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1.]
[ 9.68225696e-17  3.25390442e-16 -1.04257663e-16  1.04377536e-16
  5.31805617e-17  1.35580515e-16  1.04828517e-16 -4.16519336e-18
  5.86811706e-17  3.21962095e-17  4.58976868e-17 -9.93490585e-18
  2.20686357e-18 -8.09338637e-19  4.53015898e-17 -8.17387374e-17
 -1.54104356e-17 -1.07213839e-16  1.17675343e-16 -5.64796639e-17
  5.42391927e-17 -8.86429672e-17  1.75530692e-17 -2.10964552e-17
 -3.87624604e-18  8.53248781e-18 -5.37288618e-18 -1.18464447e-17
```

```
  3.94411170e-19]
True
```

[17]:

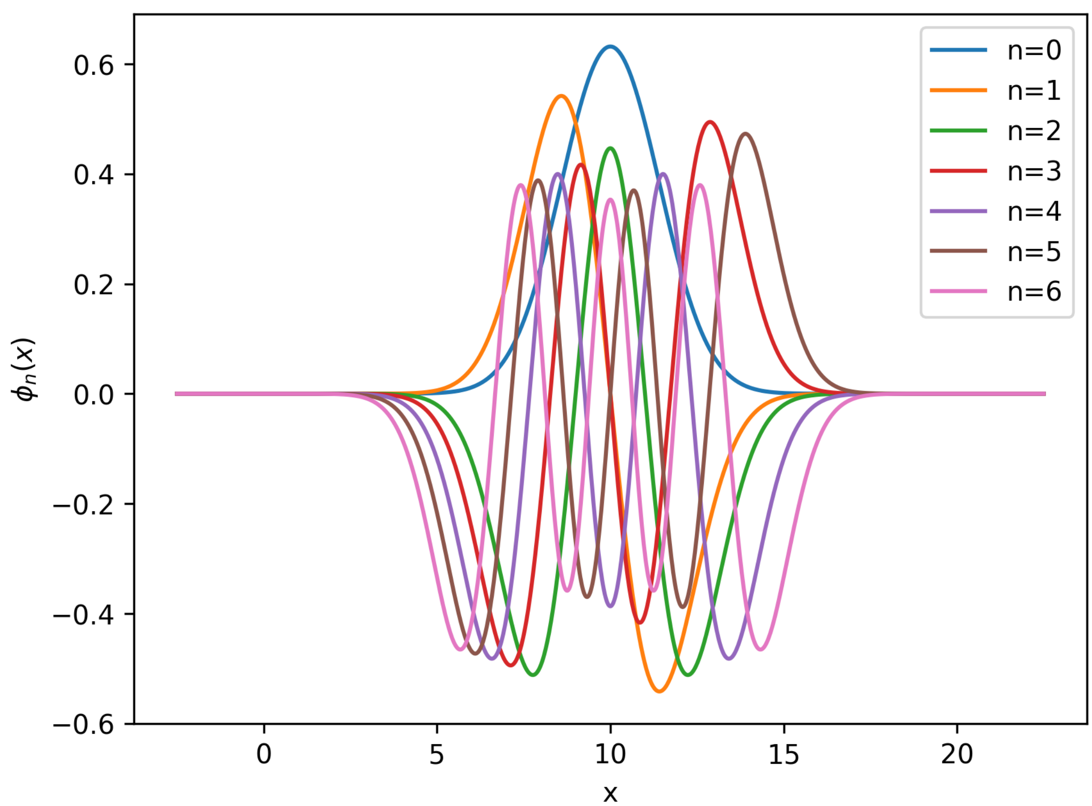
```
from matplotlib import pyplot as plt

xs = np.linspace(-0.125*L, 1.125*L, int(1e3))
phis_x = np.vstack(tuple(map(lambda j: phi_nx(xs, j+1), range(nbasis))))

with plt.style.context('default'):
    fig, ax = plt.subplots(dpi=300)

    for i, ys in enumerate(np.dot(eigvectors[:, indices[:first]].T, phis_x)):
        ax.plot(xs, ys, label=f'n={i}')

    ax.legend()
    ax.set(xlabel='x', ylabel='$\phi_n(x)$')
    plt.savefig('ho-sine-approx.png')
    plt.show()
```



[ ]:

```

```