Box_Potential_v2

April 21, 2021

[1]:
```
__author__ = "@Tssp"
__date__   = "15/04/21"
import sympy as sp
from sympy.abc import n, x, a, b
import numpy as np
import matplotlib.pyplot as plt
from method import Chebyshev_Expansion, np_Chebyshev
plt.rc('text',usetex=True)
plt.rc('font',family='serif')
ref_ticksize = 18
plt.rcParams['xtick.labelsize']=ref_ticksize
plt.rcParams['ytick.labelsize']=ref_ticksize
plt.rcParams['axes.labelsize']=ref_ticksize * 3/2
plt.rcParams['axes.titlesize']=ref_ticksize * 3/2
aur = (1 + np.sqrt(5)) / 2
aursize = (4.3*aur, 4.3)
```

[2]:
```
Vo_sym, k_sym, xi_sym, L_sym, V_plane_sym = sp.symbols('V_0 k \\xi L V_p')
```

[3]:
```
V = sp.Piecewise((Vo_sym, x<=-xi_sym-L_sym/2), (b*x**12 + a, x<=-L_sym/2), (0,␣
↪x<=L_sym/2),\
                (b*x**12 + a, x<=L_sym/2 + xi_sym), (Vo_sym, x>=xi_sym + L_sym/
↪2))
V
```

[3]:

$$\begin{cases} V_0 & \text{for}\: x \leq - \frac{L}{2} - \xi \\ a + b x^{12} & \text{for}\: x \leq - \frac{L}{2} \\ 0 & \text{for}\: x \leq \frac{L}{2} \\ a + b x^{12} & \text{for}\: x \leq \frac{L}{2} + \xi \\ V_0 & \text{for}\: x \geq \frac{L}{2} + \xi \end{cases}$$

1 Continuity

$$V_0 - a - b\left(-\frac{L}{2} - \xi\right)^{12} = 0$$

$$\frac{L^{12} b}{4096} + a = 0$$

[4]:
```
eq_1 = V.args[0][0] - V.args[1][0].subs({x: -L_sym/2 - xi_sym})
eq_1
```

[4]:

$$V_0 - a - b\left(-\frac{L}{2} - \xi\right)^{12}$$

[5]:
```
eq_2 = V.args[1][0].subs({x: -L_sym/2}) - 0
eq_2
```

[5]:

$$\frac{L^{12}b}{4096} + a$$

[6]:
```
a_temp = sp.solve(eq_2, a)[0]
a_temp
```

[6]:

$$-\frac{L^{12}b}{4096}$$

[7]:
```
b_sol = sp.solve(eq_1.subs({a: a_temp}), b)[0]
b_sol
```

[7]:

$$-\frac{4096V_0}{L^{12} - (L + 2\xi)^{12}}$$

[8]:
```
a_sol = sp.solve(eq_2.subs({b: b_sol}), a)[0]
a_sol
```

[8]:

$$\frac{L^{12}V_0}{L^{12} - (L + 2\xi)^{12}}$$

[9]:
```
V_cont = V.subs({a: a_sol,
       b: b_sol})
V_cont
```

[9]:

$$\begin{cases} V_0 & \text{for } x \le -\frac{L}{2} - \xi \\ \frac{L^{12}V_0}{L^{12}-(L+2\xi)^{12}} - \frac{4096V_0x^{12}}{L^{12}-(L+2\xi)^{12}} & \text{for } x \le -\frac{L}{2} \\ 0 & \text{for } x \le \frac{L}{2} \\ \frac{L^{12}V_0}{L^{12}-(L+2\xi)^{12}} - \frac{4096V_0x^{12}}{L^{12}-(L+2\xi)^{12}} & \text{for } x \le \frac{L}{2} + \xi \\ V_0 & \text{for } x \ge \frac{L}{2} + \xi \end{cases}$$

2 Substituying Parameters

[10]:
```
Vo_value= 1
L_value = 1
xi_value= 0.1
params = {Vo_sym: Vo_value,
         xi_sym: xi_value,
         L_sym: L_value}
```

```python
print(f'''
       Parameters
------------------------
Vo[a.u]:      {Vo_value}
L[a.u]:       {L_value}
xi[a.u]:      {xi_value}
''')
```

```
       Parameters
------------------------
Vo[a.u]:      1
L[a.u]:       1
xi[a.u]:      0.1
```

[11]:
```python
# Dimensionless
V_d = V_cont.subs({x: x*(L_sym)})
sp.simplify(V_d)
```

[11]:

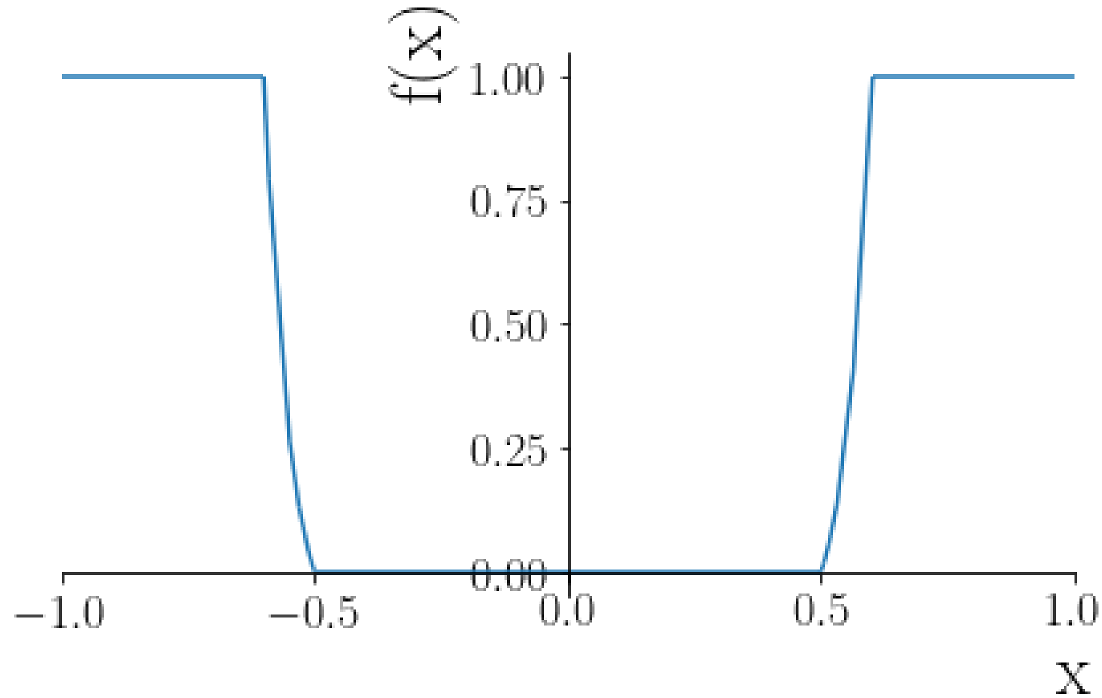
$$\begin{cases} V_0 & \text{for}\ Lx \leq -\frac{L}{2} - \xi \\ \frac{L^{12}V_0\left(1-4096x^{12}\right)}{L^{12}-\left(L+2\xi\right)^{12}} & \text{for}\ \frac{L}{2} \leq -Lx \\ 0 & \text{for}\ \frac{L}{2} \geq Lx \\ \frac{L^{12}V_0\left(1-4096x^{12}\right)}{L^{12}-\left(L+2\xi\right)^{12}} & \text{for}\ Lx \leq \frac{L}{2} + \xi \\ V_0 & \text{for}\ Lx \geq \frac{L}{2} + \xi \end{cases}$$

[12]:
```python
V_f = V_d.subs(params)
sp.simplify(V_f)
```

[12]:

$$\begin{cases} 1 & \text{for}\ x \leq -0.6 \\ 517.426481229454x^{12} - 0.12632482451891 & \text{for}\ x \leq -\frac{1}{2} \\ 0 & \text{for}\ x \leq \frac{1}{2} \\ 517.426481229454x^{12} - 0.12632482451891 & \text{for}\ x \leq 0.6 \\ 1 & \text{for}\ x \geq 0.6 \end{cases}$$

[13]:
```python
sp.plotting.plot(V_f.subs(params), xlim=(-1, 1))
```



[13]: <sympy.plotting.plot.Plot at 0x1169b3d60>

[14]:
```
xI   = np.linspace(-L_value, -L_value/2 - xi_value, endpoint=False)/(L_value)
xII  = np.linspace(-L_value/2 - xi_value, -L_value/2, endpoint=False)/(L_value)
xIII = np.linspace(-L_value/2, L_value/2, endpoint=False)/(L_value)
xIV  = np.linspace(L_value/2, L_value/2 + xi_value, endpoint=False)/(L_value)
xV   = np.linspace(L_value/2 + xi_value, L_value, endpoint=False)/(L_value)
X    = np.concatenate((xI, xII, xIII, xIV, xV))
```

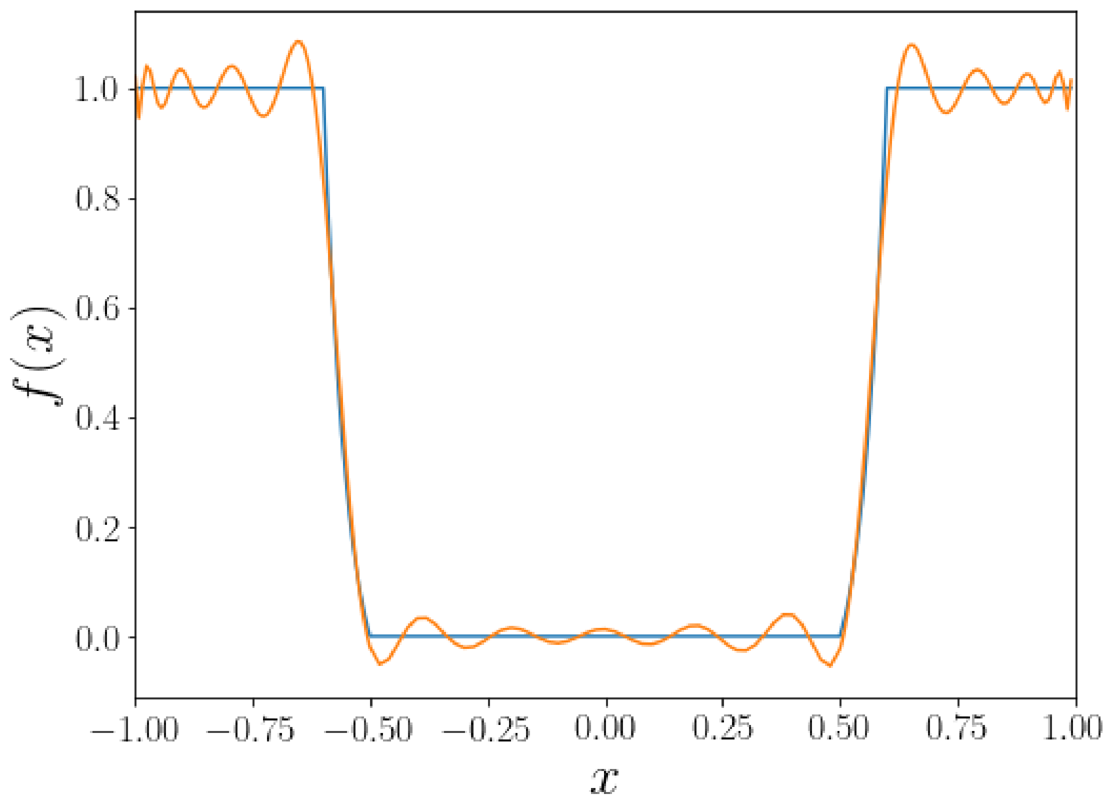
[15]:
```
b = L_value**12*Vo_value/(L_value**12 - (L_value + 2*xi_value)**12)
a = -4096*Vo_value/(L_value**12 - (L_value + 2*xi_value)**12)
```

[16]:
```
VI   = Vo_value*np.ones(xI.size)
VII  = a*(xII*L_value)**12 + b
VIII= np.zeros(xIII.size)
VIV  = a*(xIV*L_value)**12 + b
VV   = Vo_value*np.ones(xV.size)
V = np.concatenate((VI, VII, VIII, VIV, VV))
```

[17]:
```
fig, ax = plt.subplots(figsize=(8, 6))
P = np.polynomial.Chebyshev.fit(X, V, 28)
plt.plot(X, V)
plt.xlim(-1, 1)
```

```
plt.plot(X, P(X))
#plt.xticks([-xi_value, -L_value/2, 0, L_value/2, xi_value], [r'$-\xi$', r'$-L/
↪2$', r'$0$', r'$L/2$', r'$\xi$'])
plt.xlabel('$x$')
plt.ylabel('$f(x)$')
#plt.legend([r'$V(x)$', 'Chebyshev fit'], fontsize=15)
#plt.savefig('Box_Interpolation.png', dpi=200)
```

[17]: Text(0, 0.5, '$f(x)$')



3 Real Parameters

[18]:
```
kb = 1.3806488e-23          # a.u
ao = 5.2917720859e-11       # a.u
Vo_value = kb * 2e-6/4.35974417e-18    # a.u
L_value  = 70e-6/ao              # a.u
wL = 532e-9/ao
k = 2*np.pi/wL     # a.u
xi_value = 6e-6/ao# a.u
params = {Vo_sym: Vo_value,
```

```
        xi_sym: xi_value,
        L_sym: L_value}
print(f'''
       Parameters
------------------------
Vo[a.u]:      {Vo_value}
L[a.u]:       {L_value}
wL[nm]:       {wL*ao*1e9}
k[a.u]:       {k}
xi[a.u]:      {xi_value}
''')
```

```
       Parameters
------------------------
Vo[a.u]:      6.333623011645659e-12
L[a.u]:       1322808.29301995
wL[nm]:       532.0
k[a.u]:       0.0006249846732907886
xi[a.u]:      113383.56797313859
```

[19]:
```
V_f = V_d.subs(params)
sp.simplify(V_f)
```

[19]:

$$\begin{cases} 6.33362301164566 \cdot 10^{-12} & \text{for } x \leq -0.585714285714286 \\ 4.56965503905668 \cdot 10^{-9} x^{12} - 1.1156384372697 \cdot 10^{-12} & \text{for } x \leq -0.5 \\ 0 & \text{for } x \leq 0.5 \\ 4.56965503905668 \cdot 10^{-9} x^{12} - 1.1156384372697 \cdot 10^{-12} & \text{for } x \leq 0.585714285714286 \\ 6.33362301164566 \cdot 10^{-12} & \text{for } x \geq 0.585714285714286 \end{cases}$$

4 With arrays

[21]:
```
xI   = np.linspace(-L_value, -L_value/2 - xi_value, endpoint=False, num=100)/
 ↪(L_value)
xII  = np.linspace(-L_value/2 - xi_value, -L_value/2, endpoint=False, num=100)/
 ↪(L_value)
xIII = np.linspace(-L_value/2, L_value/2, endpoint=False, num=100)/(L_value)
xIV  = np.linspace(L_value/2, L_value/2 + xi_value, endpoint=False, num=100)/
 ↪(L_value)
xV   = np.linspace(L_value/2 + xi_value, L_value, endpoint=False, num=100)/
 ↪(L_value)
X    = np.concatenate((xI, xII, xIII, xIV, xV))
```

[22]:
```
b = L_value**12*Vo_value/(L_value**12 - (L_value + 2*xi_value)**12)
a = -4096*Vo_value/(L_value**12 - (L_value + 2*xi_value)**12)
```

[23]:
```
VI   = Vo_value*np.ones(xI.size)
VII = a*(xII*L_value)**12 + b
VIII= np.zeros(xIII.size)
VIV = a*(xIV*L_value)**12 + b
VV   = Vo_value*np.ones(xV.size)
V = np.concatenate((VI, VII, VIII, VIV, VV))
```

[24]:
```
fig, ax = plt.subplots(figsize=(8, 6))
P = np.polynomial.Chebyshev.fit(X, V, 30)
plt.plot(X, V)
plt.xlim(-1, 1)
plt.plot(X, P(X))
#plt.xticks([-xi_value, -L_value/2, 0, L_value/2, xi_value], [r'$-\xi$', r'$-L/
 ↪2$', r'$0$', r'$L/2$', r'$\xi$'])
plt.xlabel('$x$')
plt.ylabel('$f(x)$')
plt.legend([r'$V(x)$', 'Chebyshev fit'], fontsize=15)
plt.savefig('Box_Interpolation.png', dpi=200)
```