Exercise No. 4

David Bubeck, Pascal Becht, Patrick Nisblè

May 18, 2017

2 - Neutrons in the gravitional field

1)

The time - independent Schrödinger equation reads

$$\Psi''(x) + \frac{2m}{\hbar^2}(E - V(x))\Psi(x) = 0 \tag{1}$$

To get the equation dimensionless we use the following

$$r := \sqrt[3]{\frac{\hbar^2}{2m^2 g}} \tag{2}$$

$$\Rightarrow x := \frac{z}{r} \tag{3}$$

$$\epsilon := \frac{E}{mgr} \tag{4}$$

We obtain the dimensionless Schrödinger equation

$$\Psi''(x) + (\epsilon - x)\Psi(x) = 0 \tag{5}$$

To solve this equation we use the Numerov algorithm and the following code.

Listing 1: Exercise04.py

```python
import numpy as np
import matplotlib.pyplot as plt


def k(x):
    return epsilon - x

h = 0.001
epsilon = 0.0
n = 0                   #starting point
x = 15

y_n = 0                 #seed values
y_plus_1 = 0.1

x_out = []
y_out = []

while(n < x):
    y_minus_1 = y_n
    y_n = y_plus_1

    y_first_part = 2 * (1 - (5 / 12) * (h**2) * k(n)) * y_n
    y_sec_part = (1 + ((h**2) * k(n - h)) / 12) * y_minus_1
    y_third_part = 1 + ((h**2) * k(n + h)) / 12
    y_plus_1 = (y_first_part - y_sec_part) / y_third_part

    x_out.append(n)
    y_out.append(y_plus_1)

    n += h


plt.plot(x_out, y_out)
plt.xlabel('x')
plt.ylabel('$\psi(x)$')
plt.title('$\epsilon=$' + str(epsilon))
plt.xlim(0.0, x * 1.2)
plt.savefig('plot_' + str(epsilon) + '.png', dpi = 300)
plt.show()
```

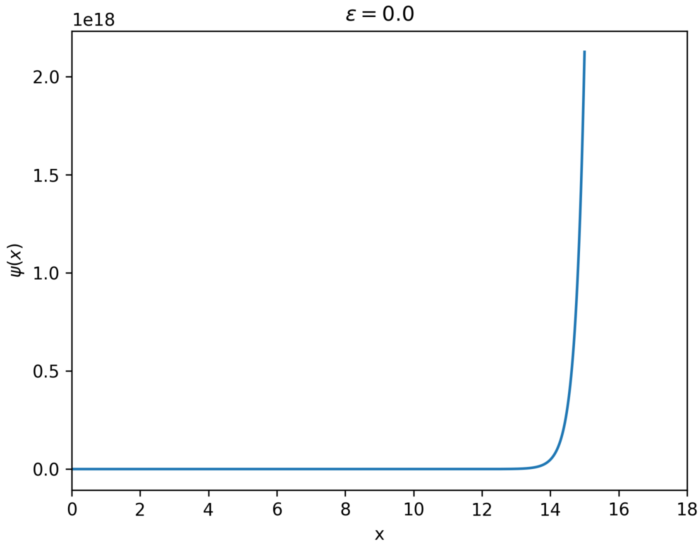

Figure 1: positive asymptotic behaviour ($\epsilon = 0.0$)

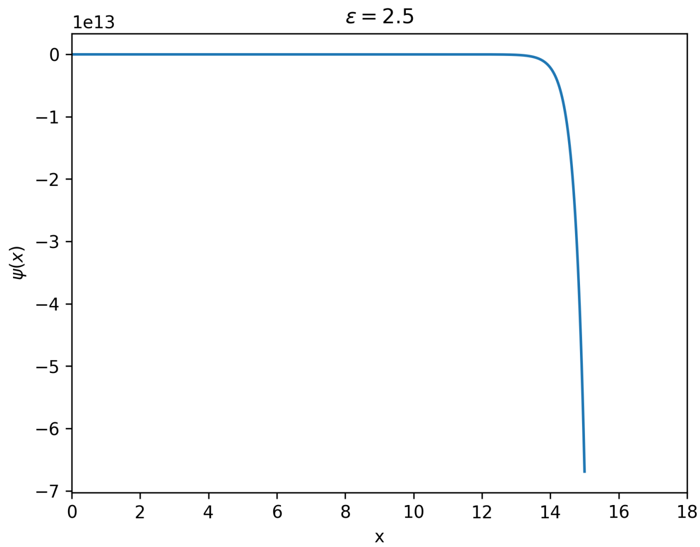

Figure 2: negative asymptotic behaviour ($\epsilon = 2.5$)

2)

To determine the eigenvalues ϵ_n we use the property of the sign change of the function $\Psi(x)$. We have modefied our code from before. You can choose a startenergy, a stepzise which you vary epsilon and how many sign changes you will search. The value of epsilon of a sign change will be printed. We start at $\epsilon = 0$ and chose our stepsize as the wanted decimals behind the comma.

Listing 2: Exercise04signChange.py

```python
import numpy as np
import matplotlib.pyplot as plt


def k(x):
    return epsilon - x

epsilon = 0          #start value of epsilon
stepsize = 0.0001     #stepsize to increas epsilon
signChanges = 3      #counts of sign changes (3 for the first 3 eigenvalues)

h = 0.001
n = 0                #starting point
x = 15

y_n = 0              #seed values
y_plus_1 = 0.1

while(n < x):
    y_minus_1 = y_n
    y_n = y_plus_1

    y_first_part = 2 * (1 - (5 / 12) * (h**2) * k(n)) * y_n
    y_sec_part = (1 + ((h**2) * k(n - h)) / 12) * y_minus_1
    y_third_part = 1 + ((h**2) * k(n + h)) / 12
    y_plus_1 = (y_first_part - y_sec_part) / y_third_part

    y_sign = y_plus_1
    n += h

if((y_sign * y_plus_1) > 0):
    i = 0
    while(i < signChanges):
        y_sign = y_plus_1
        y_plus_1 = 1
        y_n = 0
        n = 0
        n += h

        while(n < x):
            y_minus_1 = y_n
            y_n = y_plus_1

            y_first_part = 2 * (1 - (5 / 12) * (h**2) * k(n)) * y_n
            y_sec_part = (1 + ((h**2) * k(n - h)) / 12) * y_minus_1
            y_third_part = 1 + ((h**2) * k(n + h)) / 12
            y_plus_1 = (y_first_part - y_sec_part) / y_third_part

            n += h

        if((y_sign * y_plus_1) <= 0):
            print()
            print(epsilon - stepsize)
            print(epsilon)
            i += 1

        epsilon += stepsize


#2.338
#4.088
#5.521
```

The eigenvalues we found are:

$$\epsilon_0 = 2.34$$
$$\epsilon_1 = 4.09$$
$$\epsilon_2 = 5.52$$

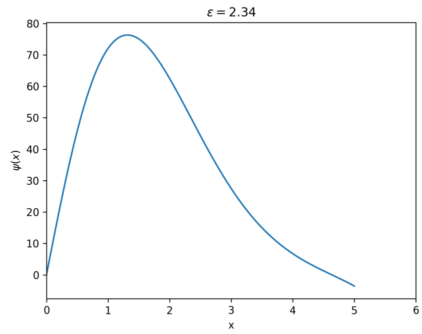

Figure 3: eigenvalue ($\epsilon_0 = 2.34$)

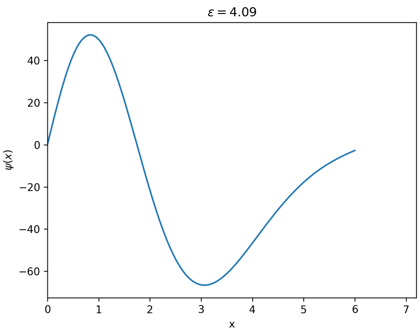

Figure 4: eigenvalue ($\epsilon_0 = 4.09$)

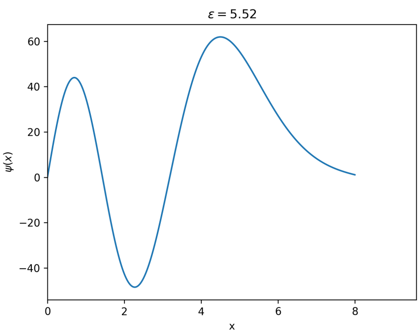

Figure 5: eigenvalue ($\epsilon_0 = 5.52$)