NTNU – Trondheim
Norwegian University of
Science and Technology

NTNU, Department of Physics

FY2045 Solutions problem set 9 fall 2023

By Henning G. Hugdal

November 3, 2023

Problem 1

Inserting

$$x = x_0 + \lambda x_1 + \lambda^2 x_2 + \dots, \tag{1}$$

into $x^3 - 4x + \lambda = 0$ and collecting terms which are the same order in λ, we get

$$\lambda^0 : \quad x_0^3 - 4x_0 = 0, \tag{2a}$$
$$\lambda^1 : \quad 3x_0^2 x_1 - 4x_1 + 1 = 0, \tag{2b}$$
$$\lambda^2 : \quad 3x_0^2 x_2 + 3x_0 x_1^2 - 4x_2 = 0, \tag{2c}$$
$$\vdots$$

Solving the equations iteratively, we get

$$x_0 = 0, \quad x_0 = \pm 2,$$
$$x_1 = -\frac{1}{3x_0^2 - 4} \Rightarrow x_1 = \begin{cases} \frac{1}{4}, & x_0 = 0, \\ -\frac{1}{8}, & x_0 = \pm 2, \end{cases}$$
$$x_2 = -\frac{3x_0 x_1^2}{3x_0^2 - 4} \Rightarrow x_2 = \begin{cases} 0, & x_0 = 0, \\ \mp\frac{3}{256}, & x_0 = \pm 2. \end{cases}$$

Hence, to second order in λ, we find the approximate solutions

$$x \approx \frac{\lambda}{4} + \mathcal{O}(\lambda^3),$$
$$x \approx \pm 2 - \frac{\lambda}{8} \mp \frac{3\lambda^2}{256} + \mathcal{O}(\lambda^3).$$

For $\lambda = 0.1$, we then get $x \approx 0.025$, $x \approx 1.9874$ and $x \approx -2.0124$, which is in good agreement with numerical results.

Problem 2

a) To express the matrix element in the position representation, we insert completeness relations $1 = \int dx|x\rangle\langle x|$,

$$\langle m|\delta V|n\rangle = \int dx \int dx' \ \langle m|\ x\rangle \langle x|\delta V|x'\rangle \left\langle x'\right|\ n\rangle\,, \tag{3}$$

and use the fact that $\langle x|\ n\rangle = \psi_n^0(x) = \sqrt{\frac{2}{L}} \sin\frac{\pi n x}{L}$ for $0 < x < L$ and zero otherwise. Moreover, we have previously found that

$$\langle x|\delta V(\hat{x})|x'\rangle = \delta V(x)\delta(x - x'). \tag{4}$$

Inserting these results, we get

$$\begin{aligned}\langle m|\delta V|n\rangle &= \int dx \int dx' \ \psi_m^{0*}(x)\delta V(x)\psi_n^0(x')\delta(x-x') = \frac{2}{L}\int_0^L dx \ \delta V(x) \sin\frac{\pi m x}{L} \cdot \sin\frac{\pi n x}{L} \\ &= \frac{2V_0}{L}\int_0^{\frac{L}{2}} dx \ \ \sin\frac{\pi m x}{L} \cdot \sin\frac{\pi n x}{L}.\end{aligned} \tag{5}$$

b) The perhaps simplest way to calculate the integral is to use the formula $2\sin\frac{x+y}{2}\sin\frac{x-y}{2} = \cos y - \cos x$, such that

$$\begin{aligned}\langle m|\delta V|n\rangle &= \frac{V_0}{L}\int_0^{\frac{L}{2}} dx \ \ \left[\cos\frac{\pi(m-n)x}{L} - \cos\frac{\pi(m+n)x}{L}\right] \\ &= \frac{V_0}{L}\left[\frac{L}{\pi(m-n)}\sin\frac{\pi(m-n)x}{L} - \frac{L}{\pi(m+n)}\sin\frac{\pi(m+n)x}{L}\right]\Big|_0^{\frac{L}{2}} \\ &= \frac{1}{\pi}\left[\frac{\sin\frac{\pi(m-n)}{2}}{m-n} - \frac{\sin\frac{\pi(m+n)}{2}}{m+n}\right].\end{aligned}$$

c) If m and n are both even or both odd, $m \pm n$ is an even number, and we therefore get $\sin[\pi(m \pm n)/2] = \sin N\pi = 0$, with N an integer.

The reason the matrix element is zero if m and n are both even or odd is the overall symmetry of the system and the symmetry of the perturbing potential. The unperturbed system is symmetric about $x = L/2$, and all wavefunctions are therefore either even or odd under the transformation $x \to L - x$: $\psi_n^{(0)}(x - L) = (-1)^{n-1}\psi_n^{(0)}(x)$. This ensures that the probability density has the same symmetry as the unperturbed system.

However, the perturbing potential breaks this symmetry. The perturbing potential can be written as a constant part and an odd function of x:

$$\delta V = V_0\Theta(L/2 - x) = \frac{V_0}{2}\Theta(L/2 - x) - \frac{V_0}{2}\Theta(x - L/2) + \frac{V_0}{2} \equiv \delta V_{\text{odd}}(x) + \frac{V_0}{2}.$$

When written in this way, the odd part does not contribute to the first order correction, but the constant part gives $V_0/2$, as shown in the lectures. However, the constant part does not contribute to higher order, since $\langle m|\ n\rangle = 0$ if $m \neq n$. Hence, we have

$$\begin{aligned}\langle m|\delta V|n\rangle = \langle m|\delta V_{\text{odd}}|n\rangle &= \int_0^L dx\ \psi_m^{(0)}(x)\delta V_{\text{odd}}(x)\psi_n^{(0)}(x) \\ &= \frac{V_0}{2}\int_0^{\frac{L}{2}} dx\ \psi_m^{(0)}(x)\psi_n^{(0)}(x) - \frac{V_0}{2}\int_{\frac{L}{2}}^L dx\ \psi_m^{(0)}(x)\psi_n^{(0)}(x).\end{aligned}$$

In the last term, we define $x' = L - x$, such that

$$\begin{aligned}\langle m|\delta V_{\text{odd}}|n\rangle &= \frac{V_0}{2}\int_0^{\frac{L}{2}} dx\ \psi_m^{(0)}(x)\psi_n^{(0)}(x) - \frac{V_0}{2}\int_0^{\frac{L}{2}} dx'\ (-1)^{m-1}\psi_m^{(0)}(x')(-1)^{n-1}\psi_n^{(0)}(x') \\ &= \frac{V_0}{2}[1 - (-1)^{m+n}]\int_0^{\frac{L}{2}} dx\ \psi_m^{(0)}(x)\psi_n^{(0)}(x).\end{aligned}$$

Since the perturbing potential is odd, $\psi_m^{(0)}$ and $\psi_n^{(0)}$ cannot have the same parity to get a finite result, meaning that m and n cannot both be odd or even, as seen from the last equality. We say that the perturbing potential only couples states with different parity.

d) From the previous question we know that $\langle m|\delta V|1\rangle = 0$ for $m = 3, 5, 7, 9$. For the remaining values of $m \leq 10$ we get

$$\begin{aligned}
\langle 2|\delta V|1\rangle &= \frac{V_0}{\pi}\left[1 + \frac{1}{3}\right] = \frac{4}{3\pi}V_0, \\
\langle 4|\delta V|1\rangle &= \frac{V_0}{\pi}\left[-\frac{1}{3} - \frac{1}{5}\right] = -\frac{8}{15\pi}V_0, \\
\langle 6|\delta V|1\rangle &= \frac{V_0}{\pi}\left[\frac{1}{5} + \frac{1}{7}\right] = \frac{12}{35\pi}V_0, \\
\langle 8|\delta V|1\rangle &= \frac{V_0}{\pi}\left[-\frac{1}{7} - \frac{1}{9}\right] = -\frac{16}{63\pi}V_0, \\
\langle 10|\delta V|1\rangle &= \frac{V_0}{\pi}\left[\frac{1}{9} + \frac{1}{11}\right] = \frac{20}{99\pi}V_0.
\end{aligned}$$

Looking at the contribution to the energy correction, we get for $m = 2$

$$\frac{16V_0^2}{9\pi^2 E_1^0(1-4)} \sim -0.06\frac{V_0^2}{E_1^0}$$

and for $m = 10$

$$\left(\frac{20V_0}{99\pi}\right)^2 \frac{1}{E_1^0(1-100)} \sim -4 \times 10^{-5}\frac{V_0^2}{E_1^0},$$

a reduction of three orders of magnitude, thanks partly due to the fast increase in energy when increasing m, $E_m^0 \propto m^2$. It therefore does not seem necessary to include many more terms in the sum for the energy correction. In fact, depending on the ratio V_0/E_1^0, terms of higher order in the perturbation expansion might be more important.

e) See the attached code for one way of solving the problem. The results are shown in fig. 1. We see that the second order perturbation results are in good agreement for $V_0/E_1^0 \lesssim 2$.

f) See the attached code for one way of solving the problem. The results are shown in fig. 1. We see that the first order perturbation results for the ground state wavefunction are in good agreement for $V_0/E_1^0 \lesssim 1$. The peak of the wavefunction is shifted towards larger x as V_0 increases, which is to be expected since having a large wavefunction at the left part becomes more costly.

g) In the limit $V_0 \to \infty$, we approach an infinite square well with length $L/2$, and the ground state energy is therefore $E_1 = \frac{4\hbar^2\pi^2}{2mL^2} = 4E_1^0$.

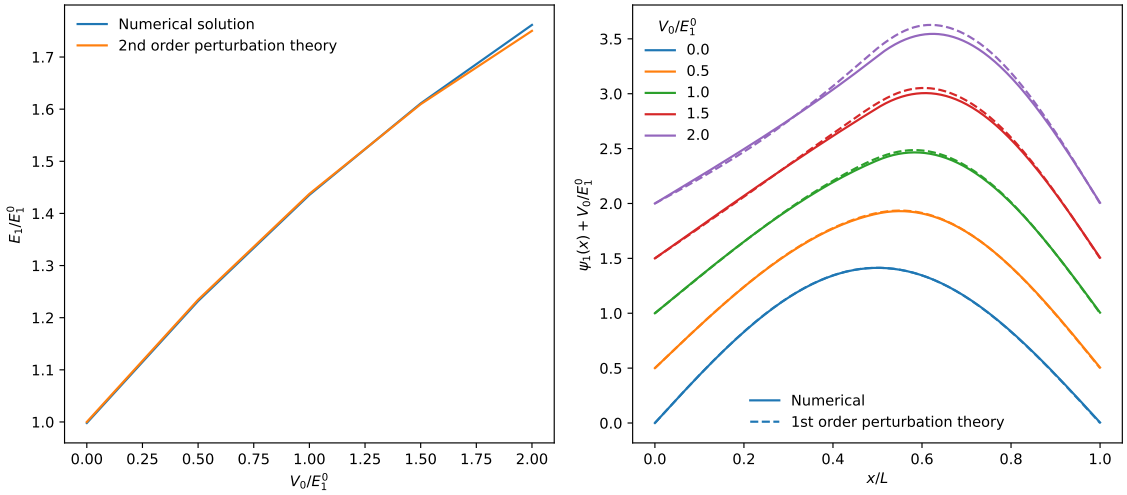

Figure 1: (left) Ground state energy as a function of V_0/E_1^0 found numerically and to second order in perturbation theory. (right) Ground state wavefunction for different V_0/E_1^0 found numerically (solid lines) and from first order perturbation theory (dashed line). The lines are shifted vertically by V_0/E_1^0 to improve visibility.

```
"""
Suggested solution to problem set 9, FY2045 Quantum Mechanics I

By Henning G. Hugdal

Oct. 30th, 2023
"""

import numpy as np
import numpy.linalg as linalg
import matplotlib.pyplot as plt


def Hamiltonian(N, V, *args):
    """Function constructing the discrete Hamiltonian matrix.

    Input:
        N:              number of grid points on the interval [0, 1]
        V:              function defining the potential, V(x, args)
        args:           tuple of arguments to be passed to the potential
                        function besides the position x

    """

    # Grid points and grid spacing
    xs, dx = np.linspace(0, 1, N, retstep=True)
```

```
27
28      # We define a N-by-N zero matrix:
29      H = np.zeros([N, N])
30
31      # Set the non-zero elements of the matrix:
32      for i, x in enumerate(xs):
33          # if i > 0 and i < N-1:
34          H[i, i] = 1/dx**2 + V(x, *args)
35          if i > 1:
36              H[i, i-1] = -1/(2*dx**2)
37          if i < N-1:
38              H[i, i+1] = -1/(2*dx**2)
39
40      return H
41
42
43  def V(x, v0):
44      """Returns perturbing potential at position x."""
45
46      return v0*np.heaviside(1/2 - x, 0)
47
48
49  def Vmn(m, n):
50      """Matrix element <m|V|n> between states m and n."""
51
52      return (np.sin(np.pi*(m-n)/2)/(m-n) - np.sin(np.pi*(m+n)/2)/(m+n))/np.pi
53
54  def En0(n):
55      """Unperturbed energy eigenvalue for level n."""
56
57      return np.pi**2/2*n**2
58
59  def psi_n0(x, n):
60      """Unperturbed wavefunction for level n."""
61
62      return np.sqrt(2)*np.sin(np.pi*n*x)
63
64  def E1approx(v0, m):
65      """
66      Second order approximation for ground state energy for potential
67      strength v0 and sum up to index m.
68      """
69
70      E10 = En0(1)
71      E11 = v0/2
72      E12 = 0
73
74      for i in np.arange(2, m+1):
75          E12 += v0**2*abs(Vmn(i, 1))**2/(E10 - En0(i))
76
77
78      return E10 + E11 + E12
```

```
79
80
81  def psi_1approx(x, v0, m):
82      """
83      First order approximation for ground state wavefunction for potential
84      strength v0 and sum up to index m.
85      """
86
87      psi_1 = psi_n0(x, 1)
88      E10 = En0(1)
89
90      for i in np.arange(2, m+1):
91          psi_1 += v0*Vmn(i, 1)/(E10 - En0(i))*psi_n0(x, i)
92
93      return psi_1
94
95
96
97
98  # Number of grid points
99  N = 1000
100
101 # System length
102 L = 1
103
104 # Unit energy
105 eps = np.pi**2/(2*L**2)
106
107 # Perturbing potential strengths
108 vs = eps*np.linspace(0, 2, 5)
109
110 # Empty list of energy eigenvalues
111 Es = np.zeros_like(vs)
112 psi_s = np.zeros([len(vs), N])
113
114 fig, ax = plt.subplots(1, 2, figsize=(11, 5))
115
116 xs = np.linspace(0, 1, N)
117
118 for i, v in enumerate(vs):
119     H = Hamiltonian(N, V, v)
120     # print(H)
121
122     # Find eigenvalue and eigenvector for potential strength v.
123     eigvals, eigvecs = linalg.eigh(H)
124
125     Es[i] = eigvals[0]
126     psi_s[i] = eigvecs[:, 0]
127
128
129 # Normalize eigenvectors
130 for i, v in enumerate(vs):
```

```
    psi_s[i] = psi_s[i]/np.sqrt(np.trapz(np.abs(psi_s[i])**2, x=xs))


    ax[1].plot(xs, v/eps + np.sign(psi_s[i, int(N/2)])*psi_s[i],
                "C{}".format(i), label=r'${value:.1f}$'.format(value = v/eps))

    ax[1].plot(xs, v/eps + psi_1approx(xs, v, 10), "C{}--".format(i))



ax[0].plot(vs/eps, Es/eps)
ax[0].plot(vs/eps, E1approx(vs, 10)/eps)

ax[0].set_xlabel(r'$V_0/E_1^0$')
ax[1].set_xlabel(r'$x/L$')

ax[0].set_ylabel(r'$E_1/E_1^0$')
ax[1].set_ylabel(r'$\psi_1(x) + V_0/E_1^0$')

lines = ax[1].get_lines()[0:2]
legend = ax[1].legend(lines, ['Numerical', '1st order perturbation theory'],
                        loc=8, frameon=False)

ax[0].legend(['Numerical solution', '2nd order perturbation theory'],
                frameon=False)
ax[1].legend(title=r'$V_0/E_1^0$', frameon=False, loc=2)
ax[1].add_artist(legend)


plt.tight_layout()

plt.savefig('perturbed_well.pdf')
plt.show()
```