

Introduction to Computational Physics – Exercise 4

Simon Groß-Böltung, Lorenz Vogel, Sebastian Willenberg

May 22, 2020

Neutrons in the gravitational field of the Earth

The Numerov algorithm is an efficient approach to numerically solve the time-independet Schrödinger equation. We want to use the Numerov algorithm to calculate the stationary states $\Psi(z)$ of neutrons in the gravitational field of the Earth. For small changes in the vertical amplitude z the potential can be expressed as $V(z) = mgz$ for $z \geq 0$. By placing a perfectly reflecting horizontal mirror at $z = 0$ the potential becomes $V(z) = \infty$ for $z < 0$. Neutrons that fall onto the mirror are reflected upwards, and so we only seek solutions for $z \geq 0$. The stationary Schrödinger equation for this problem is given by

$$\left\{ -\frac{\hbar^2}{2m} \frac{\partial^2}{\partial z^2} + mgz \right\} \Psi(z) = E\Psi(z) , \quad (1)$$

with E being the energy eigenvalue of the Hamilton operator. We can rewrite this equation to:

$$\frac{\partial^2}{\partial z^2} \Psi(z) - \frac{2m}{\hbar^2} (E - mgz) \Psi(z) = 0 \iff \frac{\partial^2}{\partial z^2} \Psi(z) - \frac{2m^2 g}{\hbar^2} \left(\frac{E}{mg} - z \right) \Psi(z) = 0 \quad (2)$$

By looking at the units of the factor in front of the bracket

$$\left[\frac{2m^2 g}{\hbar^2} \right] = \frac{\text{kg}^2 \text{m s}^{-2}}{(\text{J s})^2} = \frac{\text{kg}^2 \text{m}}{\text{s}^4} \left(\frac{\text{kg m}^2}{\text{s}^2} \right)^{-2} = \frac{\text{kg}^2 \text{m}}{\text{s}^4} \frac{\text{s}^4}{\text{kg}^2 \text{m}^4} = \text{m}^{-3} , \quad (3)$$

we find a suitable definition for the characteristic length scale z_0 :

$$\frac{1}{z_0} := \left(\frac{2m^2 g}{\hbar^2} \right)^{1/3} \quad (4)$$

If we use the notation $x := z/z_0$ and $\Psi(z) := \psi(x)/\sqrt{z_0}$, the second order partial derivative becomes:

$$\frac{\partial^2}{\partial z^2} \Psi(z) = \frac{1}{\sqrt{z_0}} \frac{1}{z_0^2} \frac{\partial^2}{\partial x^2} \psi(x) = \frac{1}{z_0^{5/2}} \frac{\partial^2}{\partial x^2} \psi(x) \quad (5)$$

The differential equation then becomes:

$$\frac{1}{z_0^{5/2}} \frac{\partial^2}{\partial x^2} \psi(x) - \frac{1}{z_0^3} \frac{1}{\sqrt{z_0}} \left(\frac{E}{mg} - z \right) \psi(x) = 0 \iff \frac{\partial^2}{\partial x^2} \psi(x) - \frac{1}{z_0} \left(\frac{E}{mg} - z \right) \psi(x) = 0 \quad (6)$$

We now define the characteristic energy scale as $E_0 := mgz_0$. The unit of E_0 is then given by:

$$[E_0] = [mgz_0] = \text{kg m s}^{-2} \text{m} = \text{kg m}^2 \text{s}^{-2} = \text{J} \quad (7)$$

After this proper choice of length and energy units

$$x := \frac{z}{z_0} = z \left(\frac{2m^2 g}{\hbar^2} \right)^{1/3} \quad \text{and} \quad \epsilon := \frac{E}{E_0} = \frac{E}{mgz_0} = \frac{E}{mg} \left(\frac{2m^2 g}{\hbar^2} \right)^{1/3} \quad (8)$$

the Schrödinger equation for neutrons in the gravitational field of the Earth can be rewritten as:

$$\frac{\partial^2}{\partial x^2} \psi(x) + (\epsilon - x) \psi(x) \iff \psi''(x) + (\epsilon - x) \psi(x) = 0 \quad (9)$$

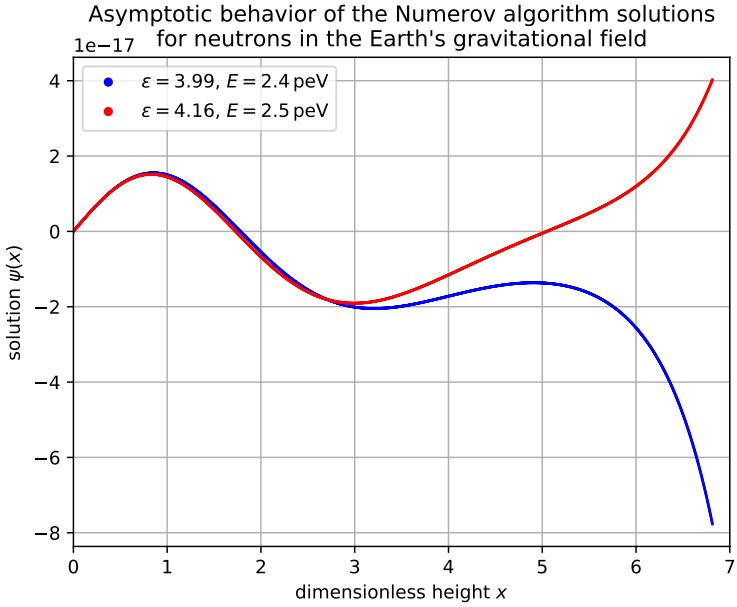


Figure 1: Asymptotic behavior of two solutions of the Numerov algorithm

Eq. (9) is a special variant of Sturm-Liouville differential equations of the type:

$$y''(x) + k(x)y(x) = 0 \quad (10)$$

The Numerov algorithm is a highly accurate discretization method to solve such types of differential equations. It is given by

$$\left(1 + \frac{1}{12}h^2k_{n+1}\right)y_{n+1} = 2\left(1 - \frac{5}{12}h^2k_n\right)y_n - \left(1 + \frac{1}{12}h^2k_{n-1}\right)y_{n-1} + \mathcal{O}(h^6) \quad (11)$$

and provides 6th order accuracy by using the three values y_n , y_{n-1} , y_{n+1} with $k_i = k(x_i)$ and $y_i = y(x_i)$.

Asymptotic behavior of the solutions

We are interested in the asymptotic behavior of the solution $\psi(x)$ for large x , i.e. whether it goes to positive or negative infinity. By using the Numerov method, we numerically calculate the solutions for two different energy values ϵ : One with positive and one with negative asymptotic behavior (see Fig. 1).

Energy eigenvalues

The energy eigenvalues E_n of Schrödinger's equation belong to normalizable eigenfunctions with $\psi(x) \rightarrow 0$ for $x \rightarrow \infty$. It means that while varying E_n (respectively ϵ_n) from smaller to larger values, the function $\psi(x)$ for $x \rightarrow \infty$ changes sign. We use this property to determine the eigenvalues E_n of the first three bound states to two decimals behind the comma (see Table 1 and Fig. 2).

Table 1: energy eigenvalues of the first three bound states

quantum state n	dimensionless energy ϵ_n	energy eigenvalue E_n [peV]
1	2.34	1.42
2	4.09	2.46
3	5.52	3.32

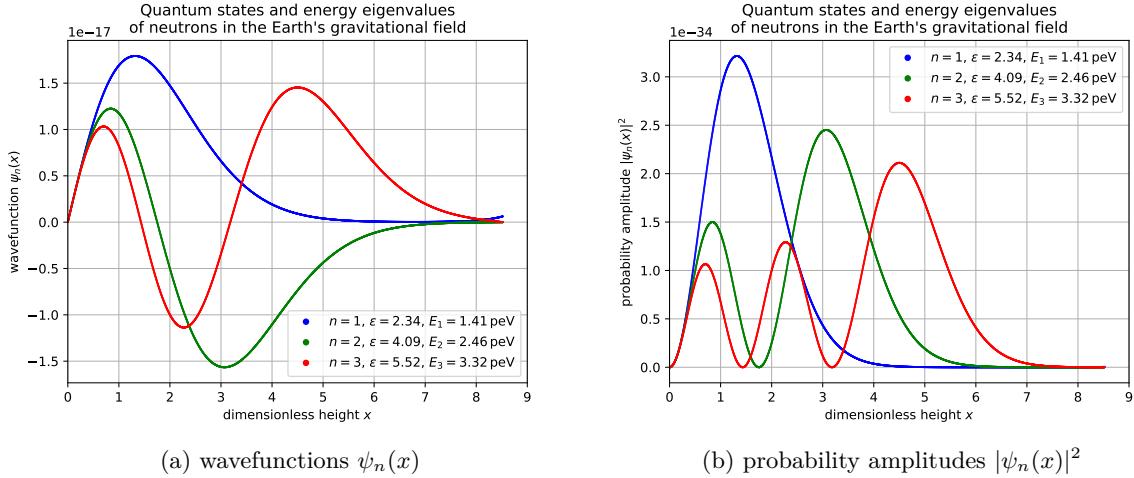


Figure 2: Wavefunctions and probability amplitudes of the quantum states of neutrons in the potential formed by the Earth's gravitational field and the horizontal mirror

Python-Code 1: Numerical solution of the Schrödinger equation using the Numerov method

```

1 # -*- coding: utf-8 -*-
2 """
3 Introduction to Computational Physics
4 - Exercise 04: Numerov Algorithm for the Schrödinger Equation
5           Neutrons in the Gravitational Field of the Earth
6 - Group: Simon Groß-Böltig, Lorenz Vogel, Sebastian Willenberg
7 """
8
9 import numpy as np; import matplotlib.pyplot as plt
10 import scipy.constants as const
11
12 class Particle:
13     def __init__(self, mass, energy=0):
14         ''' Input: mass [kg] and energy [eV] of the particle '''
15         self.mass = mass          # mass [kg] of the particle
16         self.energy = energy       # energy [eV] of the particle
17
18     def calculate_wavefunction(self, zrange, N):
19         ''' Function to compute the wavefunction for a particle in the
20             gravitational field of the Earth using the Numerov algorithm
21             Input: particle of the class "Particle" with mass [kg] and energy [eV]
22             Output: dimensionless length range and wavefunction '''
23
24     # compute the characteristic length scale
25     z0 = (const.hbar**2/(2*particle.mass**2*const.g))**(1/3)
26
27     # compute the scaled length and energy units (dimensionless)
28     x = zrange/z0; epsilon = particle.energy*const.e/(particle.mass*const.g*z0)
29
30     k = epsilon-x
31     h_sq = (abs(max(x)-min(x))/N)**2      # squared step size
32     psi = np.zeros(N)                      # empty array for the wavefunction
33     psi[0] = 0; psi[1] = 1e-20              # set conditions at the mirror
34
35     for i in range(2,N): # compute the wavefunction using the Numerov method
36         psi[i] = (2.*((1.-(5./12.)*h_sq*k[i-1])*psi[i-1]
37                     -(1.+(1./12.)*h_sq*k[i-2])*psi[i-2]))/(1.+(1./12.)*h_sq*k[i])
38     return (x, psi, epsilon)
39
40     def find_eigenvalue(self, energy_step, precision):
41         ''' Function to compute the energy eigenvalues using sign changes
42             of the asymptotic behavior for different trial energies
43             Input: particle of the class "Particle" with a trial energy [eV],
44                   energy step size [eV] and precision [eV]
45             Output: energy eigenvalue [eV] '''
46     psil = calculate_wavefunction(self, z, N)[1][N-1]
47

```

```

48     while abs(energy_step) > precision:
49         particle.energy += energy_step
50         psi2 = calculate_wavefunction(neutron, z, N)[1][N-1]
51
52         if psi1*psi2<0: # check for sign change
53             energy_step = -energy_step/2      # reduce energy step
54         psi1 = psi2
55     return particle.energy
56
57
58 # Exercise 1: We use the Numerov algorithm to solve the differential equation
59 N = int(2e4)                                # set number of iterations
60 z = 1e-6*np.linspace(0,40,N)                 # set z-range [0 to 50 micrometre]
61 energy = 1e-12*np.array([2.4,2.5])           # set energy values
62 color = ['blue','red']                       # set color of the graphs
63
64 # plot the corresponding wavefunctions for two energy values:
65 # one with positive and one with negative asymptotic behavior
66 fig, ax = plt.subplots()
67 ax.set_title('Asymptotic behavior of the Numerov algorithm solutions\n'+
68             'for neutrons in the Earth\'s gravitational field')
69 ax.set_xlabel(r'dimensionless height $x$')
70 ax.set_ylabel(r'solution $\psi(x)$')
71
72 for i in range(0,len(energy)):
73     neutron = Particle(const.neutron_mass, energy[i])
74     x, psi, eps = calculate_wavefunction(neutron, z, N)
75     ax.plot(x[::6], psi[::6], '.', markersize=1, color=color[i],
76             label=r'$\epsilon=$' + str(eps) + ' eV',
77             format(e=round(neutron.energy*1e12,2),eps=round(eps,2)))
78
79 ax.grid(); ax.legend(loc='best', markerscale=8)
80 ax.set_xlim((int(round(min(x))),int(round(max(x))))) 
81 fig.savefig('figures/Asymptotic-Behavior.pdf',format='pdf')
82
83
84 # Exercise 2: We compute the stationary states of neutrons in the gravitational
85 # field of the Earth and the energy eigenvalues of the first three bound states
86 N = int(2e4)                                # set number of iterations
87 z = 1e-6*np.linspace(0,50,N)                 # set z-range [0 to 50 micrometre]
88 trial_energy = 1e-12*np.array([1.,2.,3.])    # set trial energy values
89 color = ['blue','green','red']                # set color of the graphs
90 energy_step = 0.25e-12                         # set energy step
91 precision = 1e-18                            # set precision limit
92
93 # compute the energy eigenvalues and plot the corresponding wavefunctions
94 # as well as the probability amplitude
95 fig1, ax1 = plt.subplots(); fig2, ax2 = plt.subplots()
96 ax1.set_ylabel(r'wavefunction $\psi_n(x)$')
97 ax2.set_ylabel(r'probability amplitude $|\psi_n(x)|^2$')
98
99 for i in range(0,len(trial_energy)):
100    neutron = Particle(const.neutron_mass, trial_energy[i])
101    neutron.energy = find_eigenvalue(neutron,energy_step,precision)
102    x, psi, eps = calculate_wavefunction(neutron, z, N)
103    ax1.plot(x[::6], psi[::6], '.', markersize=1, color=color[i],
104              label=r'$n=' + str(i+1) + ', \epsilon=' + str(neutron.energy*1e12) + ' eV',
105              format(n=i+1,e=round(neutron.energy*1e12,2),eps=round(eps,2)))
106    ax2.plot(x[::6], abs(psi[::6])**2, '.', markersize=1, color=color[i],
107              label=r'$n=' + str(i+1) + ', \epsilon=' + str(neutron.energy*1e12) + ' eV',
108              format(n=i+1,e=round(neutron.energy*1e12,2),eps=round(eps,2)))
109
110 for ax in [ax1,ax2]:
111     ax.set_title('Quantum states and energy eigenvalues\n'+
112                 'of neutrons in the Earth\'s gravitational field')
113     ax.set_xlabel(r'dimensionless height $x$')
114     ax.grid(); ax.legend(loc='best', markerscale=8)
115     ax.set_xlim((int(round(min(x))),int(round(max(x))))) 
116
117 fig1.savefig('figures/Eigenvalues_Wavefunction.pdf', format='pdf')
118 fig2.savefig('figures/Eigenvalues_Probability-Amplitude.pdf', format='pdf')
119 plt.show(); plt.clf(); plt.close()

```