Introduction to Computational Physics – Exercise 4

Simon Groß-Bölting, 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

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

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

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

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

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

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

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

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

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

The differential equation then becomes:

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

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

$$[E_0] = [mgz_0] = \mathrm{kg}\,\mathrm{m}\,\mathrm{s}^{-2}\,\mathrm{m} = \mathrm{kg}\,\mathrm{m}^2\,\mathrm{s}^{-2} = \mathrm{J} \tag{7}$$

After this proper choice of length and energy units

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

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

$$\frac{\partial^2}{\partial x^2}\psi(x) + (\epsilon - x)\psi(x) \qquad \Longleftrightarrow \qquad \psi''(x) + (\epsilon - x)\psi(x) = 0 \tag{9}$$

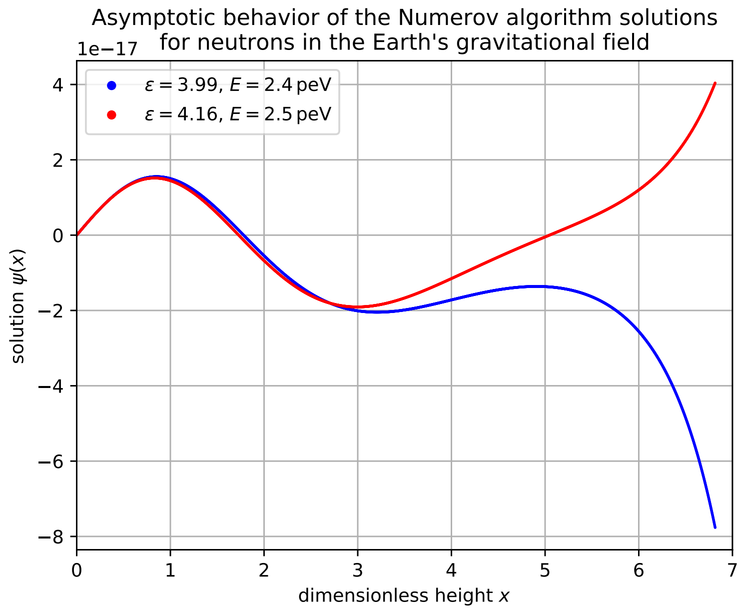


Figure 1: Asymptotic behavior of two solutions of the Numerov algorithm

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

$$y''(x) + k(x)y(x) = 0 \tag{10}$$

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

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

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 beavior of the solutions

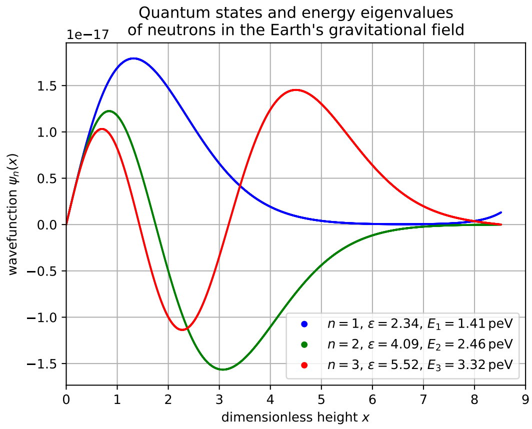
We are interested in the asymptotic behavior of the solution $\psi(x)$ for large x, i.e. whether it goes to positive or negative infinity. By using the Numerov method, we numerically calculate the solutions for two different energy values ϵ: One with positive and one with negative asymptotic behavior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

Energy eigenvalues

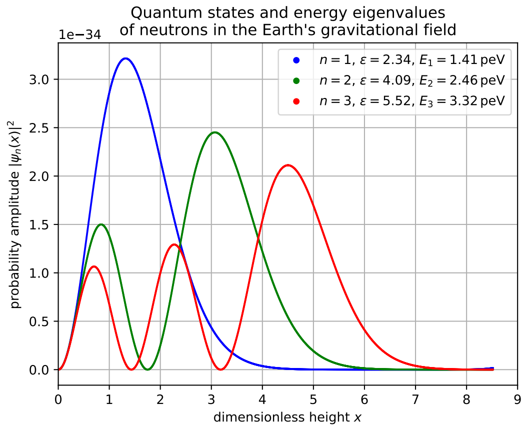
The energy eigenvalues E_n of Schrödinger's equation belong to normalizable eigenfunctions with $\psi(x) \to 0$ for $x \to \infty$. It means that while varying E_n (respectively ϵ_n) from smaller to larger values, the function $\psi(x)$ for $x \to \infty$ changes sign. We use this property to determine 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



(a) wavefunctions $\psi_n(x)$



(b) probability amplitudes $|\psi_n(x)|^2$

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

```python
# -*- coding: utf-8 -*-
"""
Introduction to Computational Physics
- Exercise 04: Numerov Algorithm for the Schrödinger Equation
               Neutrons in the Gravitational Field of the Earth
- Group: Simon Groß-Bölting, Lorenz Vogel, Sebastian Willenberg
"""

import numpy as np; import matplotlib.pyplot as plt
import scipy.constants as const

class Particle:
    def __init__(self,mass,energy=0):
        ''' Input: mass [kg] and energy [eV] of the particle '''
        self.mass = mass          # mass [kg] of the particle
        self.energy = energy      # energy [eV] of the particle

def calculate_wavefunction(particle,z,dz,N):
    ''' Function to compute the wavefunction for a particle in the
        gravitational field of the Earth using the Numerov algorithm
        Input:  particle of the class "Particle" with mass [kg] and energy [eV]
        Output: dimensionless length range and wavefunction '''

    # compute the characteristic length scale
    z0 = (const.hbar**2/(2*particle.mass**2*const.g))**(1/3)

    # compute the scaled length and energy units (dimensionless)
    x = z/z0; epsilon = particle.energy*const.e/(particle.mass*const.g*z0)

    k = epsilon-x
    h_sq = (dz/z0)**2                    # squared step size
    psi = np.zeros(N)                    # empty array for the wavefunction
    psi[0] = 0; psi[1] = 1e-20           # set conditions at the mirror

    for i in range(2,N): # compute the wavefunction using the Numerov method
        psi[i] = (2.*(1.-(5./12.)*h_sq*k[i-1])*psi[i-1]
                  -(1.+(1./12.)*h_sq*k[i-2])*psi[i-2])/(1.+(1./12.)*h_sq*k[i])
    return (x,psi,epsilon)

def find_eigenvalue(particle,z,dz,N,energy_step,precision):
    ''' Function to compute the energy eigenvalues using sign changes
        of the asymptotic behavior for different trial energies
        Input:  particle of the class "Particle" with a trial energy [eV],
                energy step size [eV] and precision [eV]
        Output: energy eigenvalue [eV] '''
    psi1 = calculate_wavefunction(particle,z,dz,N)[1][N-1]

```

```python
    while abs(energy_step) > precision:
        particle.energy += energy_step
        psi2 = calculate_wavefunction(neutron,z,dz,N)[1][N-1]

        if psi1*psi2<0: # check for sign change
            energy_step = -energy_step/2    # reduce energy step
        psi1 = psi2
    return particle.energy


# Exercise 1: We use the Numerov algorithm to solve the differential equation
N = int(2e4)                                  # set number of iterations
z, dz = np.linspace(0,40e-6,N,retstep=True) # set z-range [0 to 40 micrometre]
energy = 1e-12*np.array([2.4,2.5])            # set energy values
color = ['blue','red']                        # set color of the graphs

# plot the corresponding wavefunctions for two energy values:
# one with positive and one with negative asymptotic behavior
fig, ax = plt.subplots()
ax.set_title('Asymptotic behavior of the Numerov algorithm solutions\n'+
             'for neutrons in the Earth\'s gravitational field')
ax.set_xlabel(r'dimensionless height $x$')
ax.set_ylabel(r'solution $\psi(x)$')

for i in range(0,len(energy)):
    neutron = Particle(const.neutron_mass, energy[i])
    x, psi, eps = calculate_wavefunction(neutron,z,dz,N)
    ax.plot(x[::6], psi[::6], '.', markersize=1, color=color[i],
            label=r'$\epsilon={eps}$, $E={e}\,\mathrm{{peV}}$'
            .format(e=round(neutron.energy*1e12,2),eps=round(eps,2)))

ax.grid(); ax.legend(loc='best', markerscale=8)
ax.set_xlim((int(round(min(x))),int(round(max(x)))))
fig.savefig('figures/Asymptotic-Behavior.pdf',format='pdf')


# Exercise 2: We compute the stationary states of neutrons in the gravitational
# field of the Earth and the energy eigenvalues of the first three bound states
N = int(2e4)                                  # set number of iterations
z, dz = np.linspace(0,50e-6,N,retstep=True) # set z-range [0 to 50 micrometre]
trial_energy = 1e-12*np.array([1.,2.,3.])    # set trial energy values
color = ['blue','green','red']                # set color of the graphs
energy_step = 0.25e-12                        # set energy step
precision = 1e-18                             # set precision limit

# compute the energy eigenvalues and plot the corresponding wavefunctions
# as well as the probability amplitude
fig1, ax1 = plt.subplots(); fig2, ax2 = plt.subplots()
ax1.set_ylabel(r'wavefunction $\psi_n(x)$')
ax2.set_ylabel(r'probability amplitude $|\psi_n(x)|^2$')

for i in range(0,len(trial_energy)):
    neutron = Particle(const.neutron_mass, trial_energy[i])
    neutron.energy = find_eigenvalue(neutron,z,dz,N,energy_step,precision)
    x, psi, eps = calculate_wavefunction(neutron,z,dz,N)
    ax1.plot(x[::6], psi[::6], '.', markersize=1, color=color[i],
             label=r'$n={n}$, $\epsilon={eps}$, $E_{n}={e}\,\mathrm{{peV}}$'
             .format(n=i+1,e=round(neutron.energy*1e12,2),eps=round(eps,2)))
    ax2.plot(x[::6], abs(psi[::6])**2, '.', markersize=1, color=color[i],
             label=r'$n={n}$, $\epsilon={eps}$, $E_{n}={e}\,\mathrm{{peV}}$'
             .format(n=i+1,e=round(neutron.energy*1e12,2),eps=round(eps,2)))

for ax in [ax1,ax2]:
    ax.set_title('Quantum states and energy eigenvalues\n'+
                 'of neutrons in the Earth\'s gravitational field')
    ax.set_xlabel(r'dimensionless height $x$')
    ax.grid(); ax.legend(loc='best', markerscale=8)
    ax.set_xlim((int(round(min(x))),int(round(max(x)))))

fig1.savefig('figures/Eigenvalues_Wavefunction.pdf', format='pdf')
fig2.savefig('figures/Eigenvalues_Probability-Amplitude.pdf', format='pdf')
plt.show(); plt.clf(); plt.close()
```