


Computational quantum mechanics

▼ Status	Started
👤 Assign	 Paramesh Chandra
🕒 Created time	@April 4, 2022 2:15 PM
🕒 Last edited	@April 21, 2022 7:29 PM
▼ Priority	
☰ Subject	Physics

[Finding eigenstates solving the transcendental equation](#)

[Introduction](#)

[Properties of One-Dimensional Motion](#)

[Discrete Spectrum \(Bound States\)](#)

[Continuous Spectrum \(Unbound States\)](#)

[Mixed Spectrum](#)

[Symmetric Potentials and Parity](#)

[Summary](#)

[List of one-dimensional motions we can solve analytically](#)

[The Finite Square Well Potential](#)

[Code](#)

[Using shooting method](#)

[Time Evaluation of Wave Packet](#)

Finding eigenstates solving the transcendental equation

Introduction

We need to make something clear before starting this study of transcendental equations related to quantum mechanics. Ok, let's start with the equation defining quantum mechanics, i.e. **Schrodinger's equation**.

The Schrödinger equation describes the dynamics of a microscopic particle of mass m in

a one-dimensional time-independent potential $V(x)$ is given by

$$-\frac{\hbar^2}{2m} \frac{d^2\psi(x)}{dx^2} + V(x)\psi(x) = E\psi(x)$$

where E is the total energy of the particle. That's all we need to solve to know about the quantum system. This is the time-independent Schrodinger's equation that we will be dealing with in the upcoming section. Let's talk about the solutions first, we can expect two kinds of solutions from the above equation.

Properties of One-Dimensional Motion

Discrete Spectrum (Bound States)

Bound states occur whenever the particle cannot move to infinity. That is, the particle is confined or bound at all energies to move within a finite and limited region of space which is delimited by two classical turning points. The Schrödinger equation in this region admits only solutions that are discrete. The infinite square well potential and the harmonic oscillator are typical examples that display bound states.

Continuous Spectrum (Unbound States)

Unbound states occur in those cases where the motion of the system is not confined; a typical example is a free particle.

Mixed Spectrum

Potentials that confine the particle for only some energies give rise to mixed spectra; the motion of the particle for such potentials is confined for some energy values only.

Symmetric Potentials and Parity

- Non-degenerate spectrum
- Degenerate spectrum

Summary

- The energy spectrum of a bound state system is discrete and non-degenerate.
- The bound state wave function has: (a) n nodes if $n=0$ corresponds to the ground state and (b) $(n-1)$ nodes if $n=1$ corresponds to the ground state.

- The bound state eigenfunctions in an even potential have definite parity.
- The eigenfunctions of a degenerate spectrum in an even potential do not have definite parity.

List of one-dimensional motions we can solve analytically

- The Free Particle: Continuous States
- The Potential Step
- The Potential Barrier and Well
- The Infinite Square Well Potential

We are close...

The Finite Square Well Potential

Consider a particle of mass m moving in the following symmetric potential:

$$V(x) = \begin{cases} V_0, & x < -a/2. \\ 0, & -a/2 < x < a/2. \\ V_0 & x > a/2 \end{cases}$$

The two physically interesting cases are $E > V_0$ and $0 < E < V_0$. We expect the solutions to yield a continuous doubly-degenerate energy spectrum for $E > V_0$ and a discrete nondegenerate spectrum for $0 < E < V_0$.

Two possible solutions are,

- The Scattering Solutions ($E > V_0$)
- The Bound State Solution ($0 < E < V_0$)


Solutions are,

$$-\alpha_n \cot \alpha_n = \sqrt{R^2 - \alpha_n^2}, (\text{odd states})$$

$$-\alpha_n \tan \alpha_n = \sqrt{R^2 - \alpha_n^2}, (\text{even states})$$

Optimization and root finding (scipy.optimize) - SciPy v1.8.0 Manual

SciPy provides functions for minimizing (or maximizing) objective functions, possibly subject to constraints. It includes solvers for nonlinear problems (with support for both local and global optimization algorithms), linear programming, constrained and nonlinear least-squares, root finding, and curve fitting.

 <https://docs.scipy.org/doc/scipy/reference/optimize.html>

Code

```
import numpy as np
from matplotlib import pyplot as plt
from scipy.optimize import newton

def leftfunc(x):
    return(x*np.tan(x))

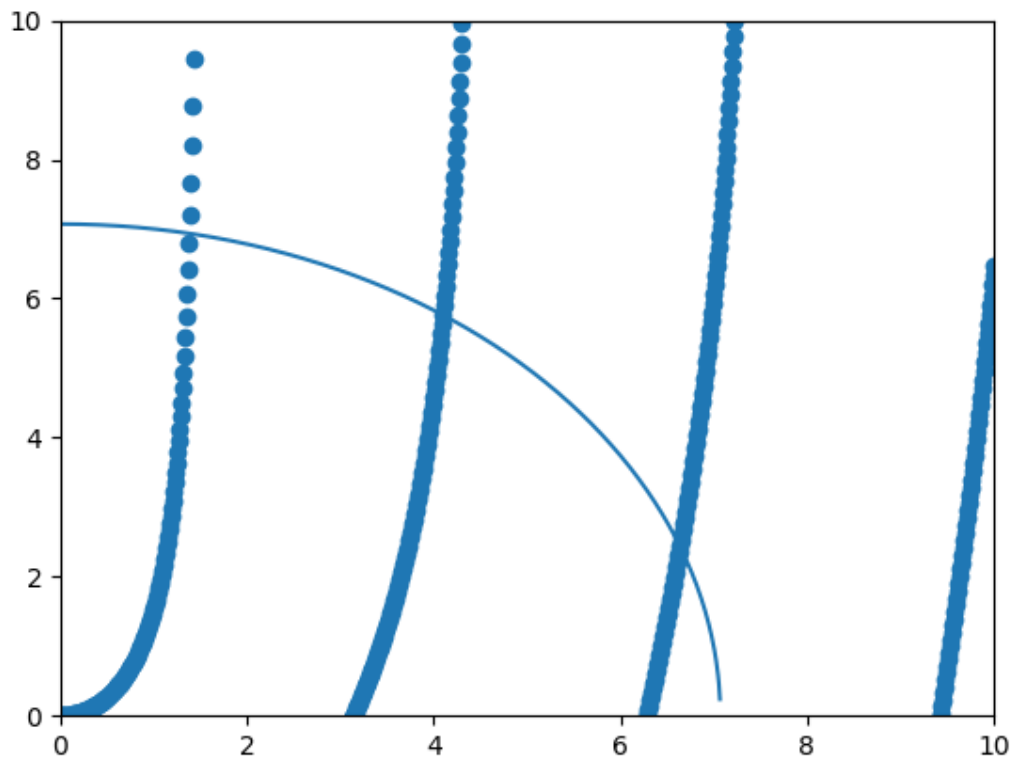
def rightfunc(x):
    return(np.sqrt(50-x**2))

def your_funcs(x):

    f = leftfunc(x)-rightfunc(x)
    return f

x = np.linspace(0,10,1000)
plt.scatter(x, leftfunc(x))
plt.plot(x, rightfunc(x))
plt.axis([0, 10, 0, 10])
plt.show()

initial_guess = float(input("Input the initial guess: "))
sol2 = newton(your_funcs, initial_guess)
print(sol2)
```



We can predict the value close to the solution and find the root from that value.

Using shooting method

Time Evaluation of Wave Packet