Университет ИТМО

Физико-технический мегафакультет

Физический факультет

УНИВЕРСИТЕТ ИТМО

Группа М3304 К работе допущен

Студент Васильков Д.А, Лавренов Д.А. Работа выполнена

Преподаватель Шоев В.И. Отчет принят

Рабочий протокол и отчет по компьютерному моделированию №1.

1) Цель работы

1. Изучение квантовых состояний частиц в различных потенциальных системах с применением аналитических и численных подходов к решению стационарного уравнения Шрёдингера.
2. Исследование влияния изменений потенциала на энергетический спектр и свойства собственных функций.

2) Задачи, решаемые при выполнении работы

1. Численное решение уравнения Шрёдингера для частиц в прямоугольной потенциальной яме с использованием дискретной матрицы Гамильтониана.
2. Анализ энергетического спектра и связанных состояний гармонического осциллятора с визуализацией собственных функций.
3. Исследование влияния бесконечно узкой и полупроницаемой перегородки на энергетические уровни и волновые функции.
4. Моделирование туннелирования частицы через потенциальный барьер, вычисление вероятности прохождения и демонстрация квантового эффекта.

3) Объект исследования

Квантовые системы, подчиняющиеся стационарному уравнению Шрёдингера

4) Метод экспериментального исследования

Численное моделирование

5) Задание 1

1. Используя уравнение Шредингера, найти связные состояния и соответствующие им собственные значения в случае прямоугольной потенциальной ямы $V(x) = \begin{cases} -U, & |x| < a, \\ 0, & |x| > a. \end{cases}$.

   Найти также собственные функции и собственные значения для осцилляторного потенциала $V(x) = \frac{1}{2} m\omega^2 x^2$. Построить графически собственные функции. Рассмотреть случай, когда в точке $x = 0$ вводится бесконечно узкая и бесконечная полупроницаемая перегородка. Выявить влияние такой перегородки на стационарные состояния.

1. Уравнение Шрёдингера для одномерной системы:

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

где:

- $\frac{\hbar^2}{2m}\frac{d^2}{dx^2}$ - оператор кинетической энергии
- V(x)- потенциал, который задаёт свойства системы
- E - собственные значения энергии
- $\psi(x)$- волновая функция, соответствующая E

2. В задаче рассматривается прямоугольная потенциальная яма

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

- Внутри ямы $(|x| < a)$: $V(x) = -U$ частица находится в "негативной" потенциальной энергии, где возможны связанные состояния (E < 0)
- Вне ямы $(|x| \geq a)$: $V(x) = 0$ и волновая функция затухает экспоненциально.

Связанные состояния - это такие энергии, при которых частица локализована внутри ямы и не уходит в бесконечность.

3. Для численного решения задача переводится на дискретную сетку. Пространство x дискретизируется с шагом Δx

- $x \in [-2a, 2a]$ чтобы захватить как область ямы, так и окружающее пространство.
- Число точек N выбирается достаточно большим (N=2000) для точности.

Дискретизация приводит к следующему приближению второй производной:

$$\frac{d^2\psi(x)}{dx^2} \approx \frac{\psi_{i+1} - 2\psi_i + \psi_{i-1}}{\Delta x^2}$$

4. Гамильтониан H для дискретного пространства записывается как матрица:

$$H = T + V$$

Где:

- T — матрица кинетической энергии, зависящая от второй разности:

$$T_{i,j} = \begin{cases} -\frac{\hbar^2}{2m\Delta x^2}, & j = i \pm 1, \\ \frac{\hbar^2}{m\Delta x^2}, & j = i, \\ 0, & \text{иначе.} \end{cases}$$

- V — диагональная матрица потенциала

$$V_{i,j} = \begin{cases} V(x_i), & i = j, \\ 0, & i \neq j. \end{cases}$$

5. Решение уравнения Шрёдингера
Численно решается задача на собственные значения для матрицы H:\

$$H\psi = E\psi$$

где:

- E — собственные значения (энергии),

- ψ — собственные векторы (дискретные волновые функции).

Используется функция `numpy.linalg.eigh`

6. Волновые функции ψ(x) нормируются так, чтобы выполнялось условие:

$$\int |\psi(x)|^2 dx = 1$$

Численно это делается с помощью метода трапеций:

$$\psi_n \rightarrow \frac{\psi_n}{\sqrt{\int |\psi_n(x)|^2 dx}}$$

7. Связанные состояния соответствуют собственным значениям E < 0. Волновые функции для связанных состояний имеют вид:
   - Внутри ямы (|x| < a) — синусоидальные колебания.
   - Вне ямы (|x| ≥ a) — экспоненциальное затухание.

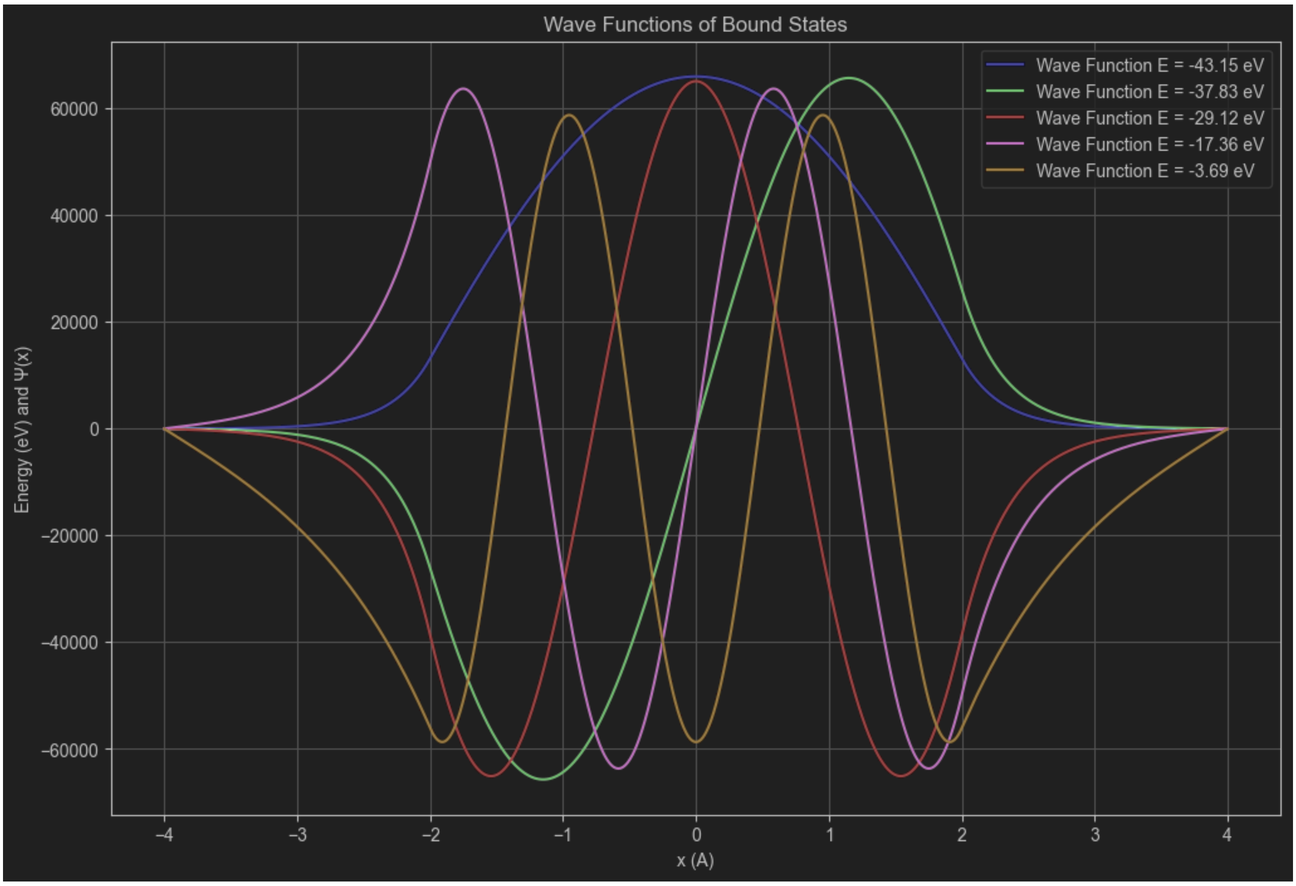
8. Визуализация

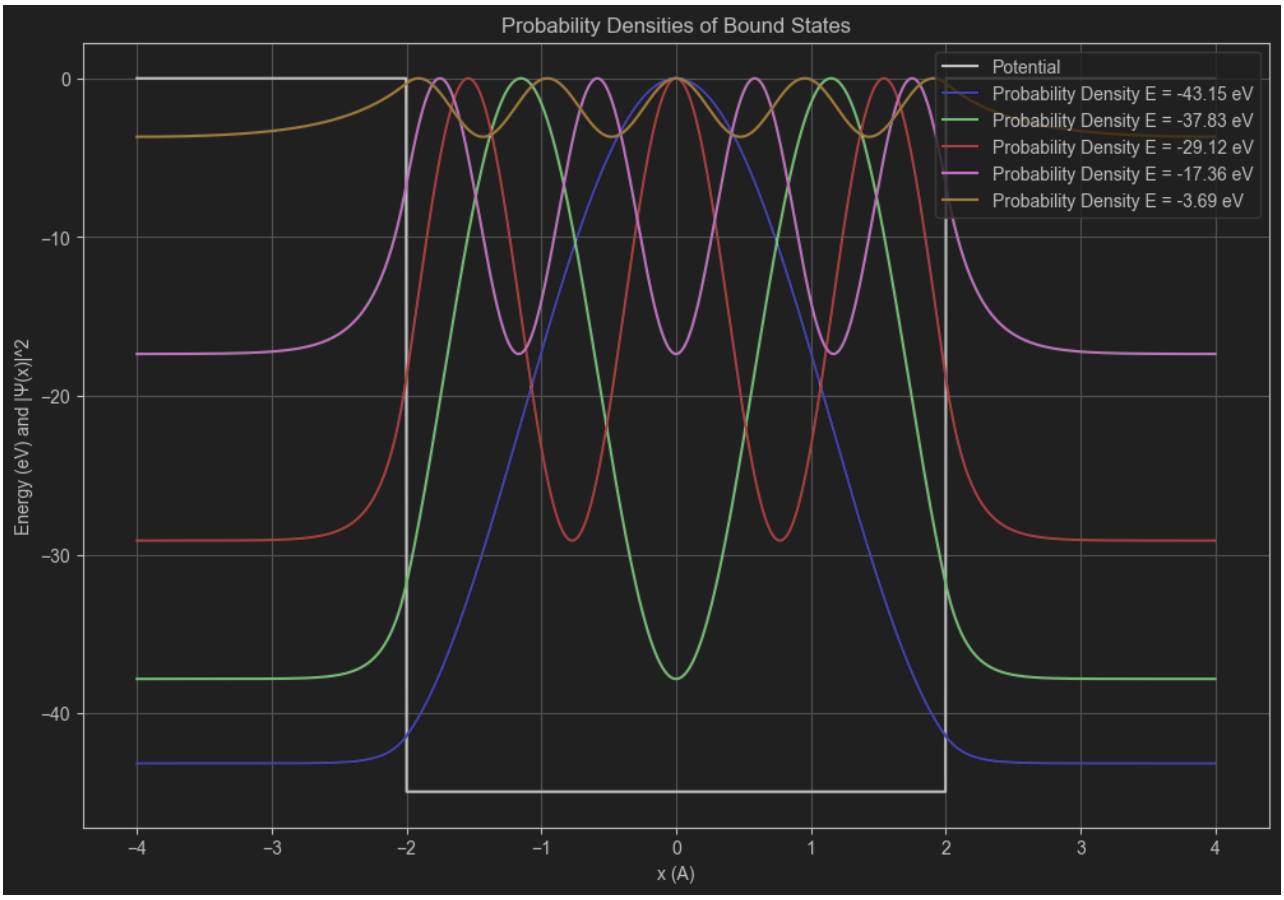
- **Волновые функции**: Каждая волновая функция $\psi_n(x)$ отображается на графике вместе с соответствующей энергией E_n. Это позволяет визуально оценить форму связанных состояний.
- **Плотности вероятности**: Плотность вероятности $|\psi_n(x)|^2$ отображает вероятность нахождения частицы в данной области пространства. Она смещается на уровень энергии E_n для наглядности.
- **Потенциал**: На графике также отображается профиль потенциала V(x), чтобы показать, как волновые функции распределены относительно ямы.

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

plt.rcParams['font.family'] = 'sans-serif'
plt.rcParams['font.sans-serif'] = ['Arial']
plt.rcParams['mathtext.fontset'] = 'dejavusans'

def define_constants():
    constants = {
        "m_electron": 9.10938e-31,  # Масса электрона (кг)
        "hbar": 1.054571e-34,       # Постоянная Планка (Дж·с)
        "U_eV": 45,                    # Глубина потенциальной ямы (эВ)
        "U0": 45 * 1.6e-19,         # Глубина потенциальной ямы (Дж)
        "a": 2e-10                     # Полуширина ямы (м)
    }
    return constants

def discretize_space(a, N_points=2000):
    """Создаёт дискретизированное пространство."""
    x_min, x_max = -2 * a, 2 * a
    x = np.linspace(x_min, x_max, N_points)
    dx = x[1] - x[0]
    return x, dx

def define_potential(x, a, U0):
    """Определяет потенциал в пространстве."""
```

```python
    V = np.zeros_like(x)
    V[np.abs(x) < a] = -U0  # Потенциал внутри ямы
    V[np.abs(x) >= a] = 0   # Потенциал вне ямы
    return V

def build_hamiltonian(N_points, dx, V, m_electron, hbar):
    """Создаёт гамильтониан."""
    H = np.zeros((N_points, N_points))
    coeff = -(hbar**2) / (2 * m_electron * dx**2)  # Коэффициент при второй производной

    for i in range(N_points):
        H[i, i] = -2 * coeff + V[i]
        if i > 0:
            H[i, i - 1] = coeff
        if i < N_points - 1:
            H[i, i + 1] = coeff

    return H

def solve_schrodinger(H):
    """Решает уравнение Шрёдингера."""
    E_values, psi_vectors = np.linalg.eigh(H)
    return E_values, psi_vectors

def get_bound_states(E_values, psi_vectors, x):
    """Находит связанные состояния и нормирует волновые функции."""
    bound_states = E_values < 0
    E_bound = E_values[bound_states]
    psi_bound = psi_vectors[:, bound_states]
    psi_bound_norm = psi_bound / np.sqrt(np.trapezoid(psi_bound**2, x, axis=0))
    return E_bound, psi_bound_norm

def plot_wavefunctions(x, E_bound, psi_bound_norm):
    """Строит график волновых функций."""
    plt.figure(figsize=(12, 8))
    colors = ['blue', 'green', 'red', 'purple', 'orange']

    for n, E in enumerate(E_bound):
        E_eV = E / 1.60218e-19  # Перевод энергии в эВ
        psi = psi_bound_norm[:, n]
        plt.plot(
            x * 1e10,
            psi + E_eV,
            label=f'Wave Function E = {E_eV:.2f} eV',
            color=colors[n % len(colors)]
        )

    plt.title('Wave Functions of Bound States')
    plt.xlabel('x (A)')
    plt.ylabel('Energy (eV) and Ψ(x)')
    plt.legend()
    plt.grid()
    plt.show()

def plot_probability_densities(x, E_bound, psi_bound_norm, V):
    """Строит график плотностей вероятностей."""
    plt.figure(figsize=(12, 8))
    colors = ['blue', 'green', 'red', 'purple', 'orange']

    plt.plot(x * 1e10, V / 1.60218e-19, 'k-', label='Potential')  # Потенциал в эВ

    for n, E in enumerate(E_bound):
        E_eV = E / 1.60218e-19
        psi = psi_bound_norm[:, n]
```

```
        density = psi**2
        scaled_density = density / np.max(density) * abs(E_eV)  # Масштабирование
        plt.plot(
            x * 1e10,
            scaled_density + E_eV,
            label=f'Probability Density E = {E_eV:.2f} eV',
            color=colors[n % len(colors)]
        )

    plt.title('Probability Densities of Bound States')
    plt.xlabel('x (A)')
    plt.ylabel('Energy (eV) and |Ψ(x)|^2')
    plt.legend(loc='upper right')
    plt.grid()
    plt.show()


def main():
    # Определение параметров и пространства
    constants = define_constants()
    x, dx = discretize_space(constants["a"])
    V = define_potential(x, constants["a"], constants["U0"])

    # Построение и решение уравнения
    H = build_hamiltonian(len(x), dx, V, constants["m_electron"], constants["hbar"])
    E_values, psi_vectors = solve_schrodinger(H)
    E_bound, psi_bound_norm = get_bound_states(E_values, psi_vectors, x)

    plot_wavefunctions(x, E_bound, psi_bound_norm)
    plot_probability_densities(x, E_bound, psi_bound_norm, V)

if __name__ == "__main__":
    main()
```





6) Задание 2.

1.Гармонический потенциал описывается выражением:

$$V(x)=\frac{1}{2}m\omega^2x^2$$

Где:

- m — масса частицы
- ω — угловая частота
- x — координата частицы

Задача состоит в нахождении собственных функций $\psi_n(x)$ и собственных значений энергии E_n для уравнения Шрёдингера:

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

2. Решение уравнения Шрёдингера для гармонического осциллятора:
Решение уравнения Шрёдингера для гармонического осциллятора приводит к квантованию энергии:

$$E_n=\hbar\omega\left(n+\frac{1}{2}\right),\quad n=0,1,2,\dots$$

где n — квантовое число, определяющее уровень энергии

Собственные функции (волновые функции) имеют вид:

$$\psi_n(x) = \left(\frac{\alpha}{\pi}\right)^{1/4} \frac{1}{\sqrt{2^n n!}} H_n(\xi) e^{-\xi^2/2}$$

Где:

- $\alpha = \sqrt{\frac{m\omega}{\hbar}}$ - параметр нормировки
- $\xi = \alpha x$ - безразмерная координата
- $H_n(\xi)$ - полином Эрмита n-го порядка

3. В условии задачи добавляется **бесконечно узкая и бесконечно высокая перегородка** в точке x=0. Это означает:

- Волновая функция должна быть равна нулю для $x < 0$
- Для $x \geq 0$ волновая функция сохраняет стандартный вид, но требует перенормировки.

Математически это выражается так:

$$\psi_n(x) = \begin{cases} 0, & x < 0, \\ C \cdot \psi_n(x), & x \geq 0, \end{cases}$$

где C — нормировочный коэффициент, обеспечивающий выполнение условия нормировки:

$$\int_0^\infty |\psi_n(x)|^2 dx = 1$$

4. Влияние перегородки на стационарные состояния

Влияние перегородки не изменяет уровни энергии, так как гармонический осциллятор сохраняет свои квантованные значения:

$$E_n = \hbar\omega\left(n + \frac{1}{2}\right)$$

Форма волновых функций

Волновые функции изменяются:

- Для $x < 0$: $\psi_n(x) = 0$
- Для $x \geq 0$: волновая функция перенормируется

Это приводит к изменению плотности вероятности $|\psi_n(x)|^2$ так как частица ограничена правой половиной пространства ($x \geq 0$).

5. Численное решение

(1) Расчет потенциала V(x)

Гармонический потенциал вычисляется по формуле

$$V(x) = \frac{1}{2} m\omega^2 x^2$$

Где:

- m — масса частицы
- ω — угловая частота

- x — массив координат.

(2) Волновая функция $\psi_n(x)$

Общая формула для волновой функции гармонического осциллятора:

$$\psi_n(x) = \left(\frac{\alpha}{\pi}\right)^{1/4} \frac{1}{\sqrt{2^n n!}} H_n(\xi) e^{-\xi^2/2}$$

(3) Построение графиков без барьера

(a) Волновые функции

График строится для нескольких значений квантового числа n. Волновая функция сдвигается на уровень энергии E_n:

$$E_n = \hbar\omega \left(n + \frac{1}{2}\right)$$

Сдвиг выполняется так:

$$\psi_n(x) \to \psi_n(x) + E_n$$

(b) Плотность вероятности

Плотность вероятности вычисляется как квадрат модуля волновой функции:

$$|\psi_n(x)|^2$$

На графике она также сдвигается на уровень энергии:

$$|\psi_n(x)|^2 \to |\psi_n(x)|^2 + E_n.$$

(4) Построение графиков с барьером

Когда добавляется перегородка в точке x=0, волновая функция изменяется:

$$\psi_n(x) = \begin{cases} 0, & x < 0, \\ \psi_n(x), & x \geq 0. \end{cases}$$

После введения барьера волновую функцию нужно перенормировать:

$$\int_0^\infty |\psi_n(x)|^2 dx = 1$$

Сдвиг на уровень энергии выполняется так же, как и без барьера:

$$\psi_n(x) + E_n, \quad |\psi_n(x)|^2 + E_n.$$

(5) Энергии

Собственные значения энергии не изменяются при добавлении барьера:

$$E_n = \hbar\omega \left(n + \frac{1}{2}\right)$$

(6) Результаты

Графики без барьера:

- Волновые функции симметричны относительно x= 0.
- Плотности вероятности также симметричны.

Графики с барьером:

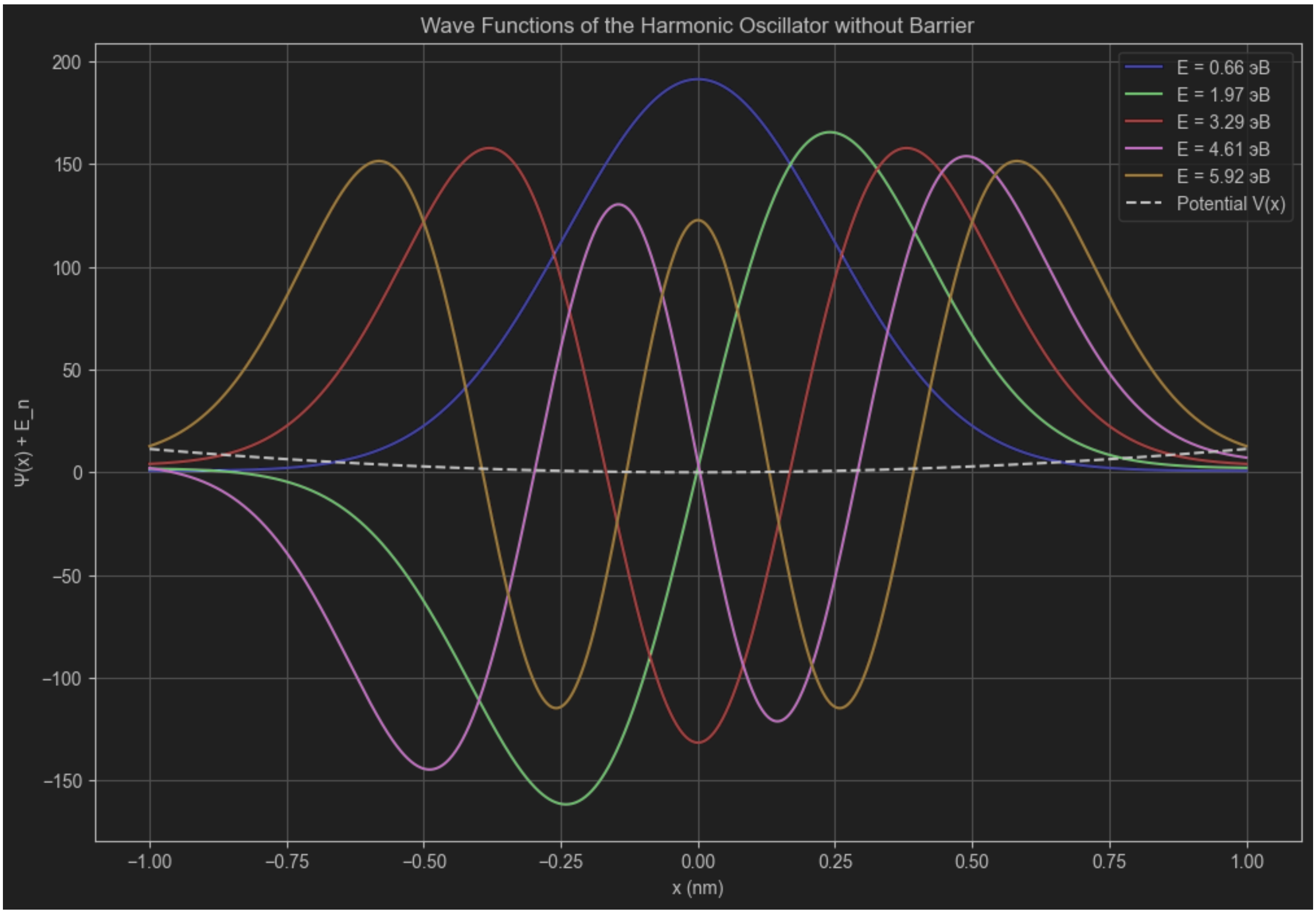
- Волновые функции обнуляются для x < 0, теряя симметрию.
- Плотности вероятности смещаются в область x ≥ 0.

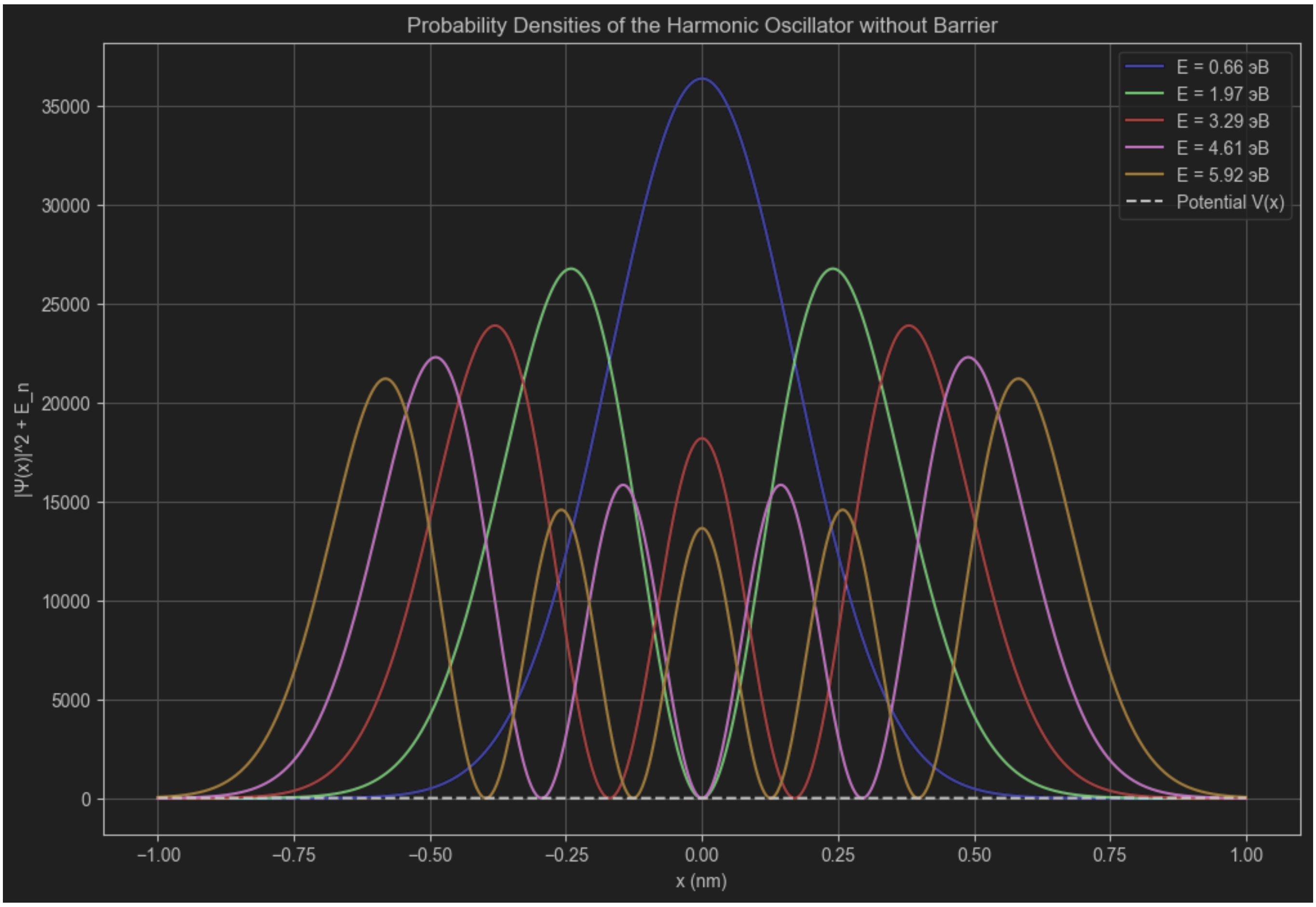
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.special import hermite
from math import factorial

# Константы
m = 9.10938e-31  # масса электрона
hbar = 1.054571e-34  # постоянная Планка
omega = 2e15  # угловая частота
alpha = np.sqrt(m * omega / hbar)  # параметр нормировки

def calculate_potential(x):
    """
    Вычисляет гармонический потенциал для заданных координат x.

    Аргументы:
    x -- массив координат (в метрах).

    Возвращает:
    Потенциал в электронвольтах (эВ).
    """
    V = 0.5 * m * omega**2 * x**2  # Гармонический потенциал в Джоулях
    return V / 1.60218e-19  # Перевод в электронвольты

def psi_n(n, x):
    """
    Вычисляет волновую функцию для заданного квантового числа n и координат x.

    Аргументы:
    n -- квантовое число.
    x -- массив координат (в метрах).

    Возвращает:
    Волновую функцию Ψ_n(x).
```

```
    """
    xi = alpha * x  # Безразмерная координата
    Hn = hermite(n)(xi)  # Полином Эрмита
    # Нормировочный коэффициент
    normalization = (1.0 / np.sqrt(2**n * factorial(n))) * (alpha / np.pi)**0.25
    psi = normalization * Hn * np.exp(-xi**2 / 2)  # Волновая функция
    return psi

def plot_wavefunctions(x, n_max, V_eV):
    """
    Строит графики волновых функций для гармонического осциллятора.

    Аргументы:
    x -- массив координат (в метрах).
    n_max -- максимальное квантовое число.
    V_eV -- потенциал в электронвольтах (эВ).
    """
    plt.figure(figsize=(12, 8))
    colors = ['blue', 'green', 'red', 'purple', 'orange']  # Цвета для графиков
    for n in range(0, n_max + 1):
        psi = psi_n(n, x)  # Волновая функция
        E_n = hbar * omega * (n + 0.5)  # Энергия уровня
        E_eV = E_n / 1.60218e-19  # Перевод энергии в эВ
        # Сдвиг волновой функции на уровень энергии
        plt.plot(
            x * 1e9,
            psi + E_eV,
            label=f'E = {E_eV:.2f} эВ',
            color=colors[n % len(colors)]
        )
    # График гармонического потенциала
    plt.plot(x * 1e9, V_eV, 'k--', label='Potential V(x)')
    plt.title('Wave Functions of the Harmonic Oscillator without Barrier')
    plt.xlabel('x (nm)')
    plt.ylabel('Ψ(x) + E_n')
    plt.legend()
    plt.grid(True)
    plt.show()

def plot_probability_densities(x, n_max, V_eV):
    """
    Строит графики плотностей вероятностей для гармонического осциллятора.

    Аргументы:
    x -- массив координат (в метрах).
    n_max -- максимальное квантовое число.
    V_eV -- потенциал в электронвольтах (эВ).
    """
    plt.figure(figsize=(12, 8))
    colors = ['blue', 'green', 'red', 'purple', 'orange']  # Цвета для графиков
    for n in range(0, n_max + 1):
        psi = psi_n(n, x)  # Волновая функция
        prob_density = np.abs(psi)**2  # Плотность вероятности
        E_n = hbar * omega * (n + 0.5)  # Энергия уровня
        E_eV = E_n / 1.60218e-19  # Перевод энергии в эВ
        # Сдвиг плотности вероятности на уровень энергии
        plt.plot(
            x * 1e9,
            prob_density + E_eV,
            label=f'E = {E_eV:.2f} эВ',
            color=colors[n % len(colors)]
        )
    # График гармонического потенциала
    plt.plot(x * 1e9, V_eV, 'k--', label='Potential V(x)')
```

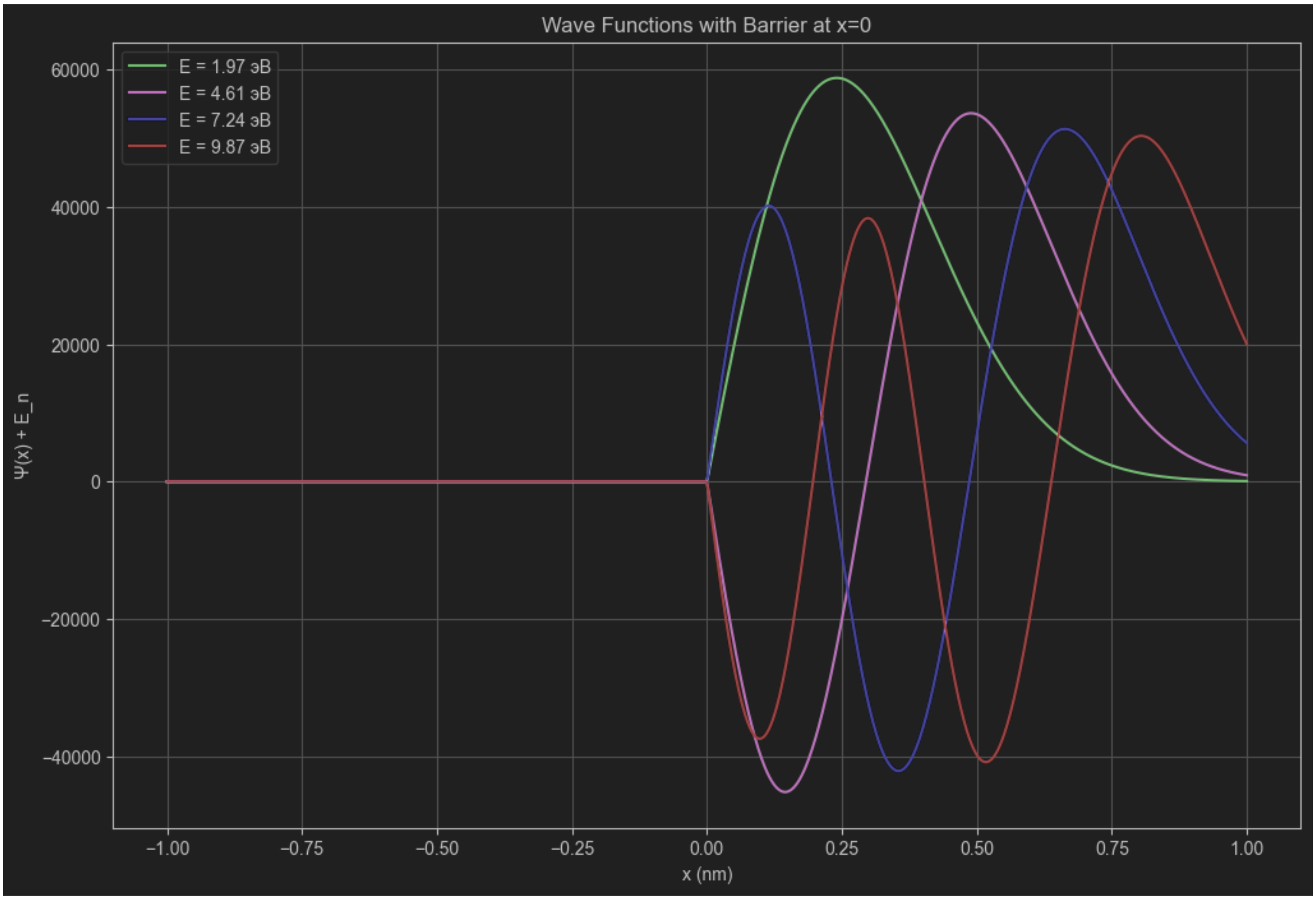
```
    plt.title('Probability Densities of the Harmonic Oscillator without Barrier')
    plt.xlabel('x (nm)')
    plt.ylabel('|Ψ(x)|^2 + E_n')
    plt.legend()
    plt.grid(True)
    plt.show()

def plot_wavefunctions_with_barrier(x, n_values):
    """
    Строит волновые функции с перегородкой в x=0.

    Аргументы:
    x -- массив координат (в метрах).
    n_values -- список квантовых чисел.
    """
    plt.figure(figsize=(12, 8))
    colors = ['blue', 'green', 'red', 'purple', 'orange']  # Цвета для графиков
    for n in n_values:
        psi = psi_n(n, x)  # Волновая функция
        # Установка волновой функции равной 0 для x < 0 (перегородка)
        psi_modified = np.where(x >= 0, psi, 0)
        # Нормировка волновой функции
        norm_factor = np.sqrt(np.trapezoid(np.abs(psi_modified)**2, x))
        psi_modified /= norm_factor
        E_n = hbar * omega * (n + 0.5)  # Энергия уровня
        E_eV = E_n / 1.60218e-19  # Перевод энергии в эВ
        # Сдвиг волновой функции на уровень энергии
        plt.plot(
            x * 1e9,
            psi_modified + E_eV,
            label=f'E = {E_eV:.2f} эВ',
            color=colors[n % len(colors)]
        )
    plt.title('Wave Functions with Barrier at x=0')
    plt.xlabel('x (nm)')
    plt.ylabel('Ψ(x) + E_n')
    plt.legend()
    plt.grid(True)
    plt.show()

def plot_probability_densities_with_barrier(x, n_values):
    """
    Строит плотности вероятности с перегородкой в x=0.

    Аргументы:
    x -- массив координат (в метрах).
    n_values -- список квантовых чисел.
    """
    plt.figure(figsize=(12, 8))
    colors = ['blue', 'green', 'red', 'purple', 'orange']
    for n in n_values:
        psi = psi_n(n, x)  # Волновая функция
        psi_modified = np.where(x >= 0, psi, 0)  # Волновая функция с перегородкой
        norm_factor = np.sqrt(np.trapezoid(np.abs(psi_modified)**2, x))  # Нормировка
        psi_modified /= norm_factor
        prob_density = np.abs(psi_modified)**2  # Плотность вероятности
        E_n = hbar * omega * (n + 0.5)  # Энергия уровня
        E_eV = E_n / 1.60218e-19  # Перевод энергии в эВ
        # Сдвиг плотности вероятности на уровень энергии
        plt.plot(
            x * 1e9, prob_density + E_eV,
            label=f'E = {E_eV:.2f} эВ',
            color=colors[n % len(colors)]
        )
```

```
    plt.title('Probability Densities with Barrier at x=0')
    plt.xlabel('x (nm)')
    plt.ylabel('|Ψ(x)|^2 + E_n')
    plt.legend()
    plt.grid(True)
    plt.show()

def main():
    x = np.linspace(-1e-9, 1e-9, 1000)  # Массив координат (от -1 до 1 нм)
    V_eV = calculate_potential(x)  # Гармонический потенциал
    n_max = 4  # Максимальное квантовое число для графиков
    n_values = [1, 3, 5, 7]  # Квантовые числа для графиков с перегородкой

    plot_wavefunctions(x, n_max, V_eV)  # Волновые функции без перегородки
    plot_probability_densities(x, n_max, V_eV)  # Плотности вероятности без перегородки
    plot_wavefunctions_with_barrier(x, n_values)  # Волновые функции с перегородкой
    plot_probability_densities_with_barrier(x, n_values)  # Плотности вероятности с
перегородкой


if __name__ == "__main__":
    main()
```

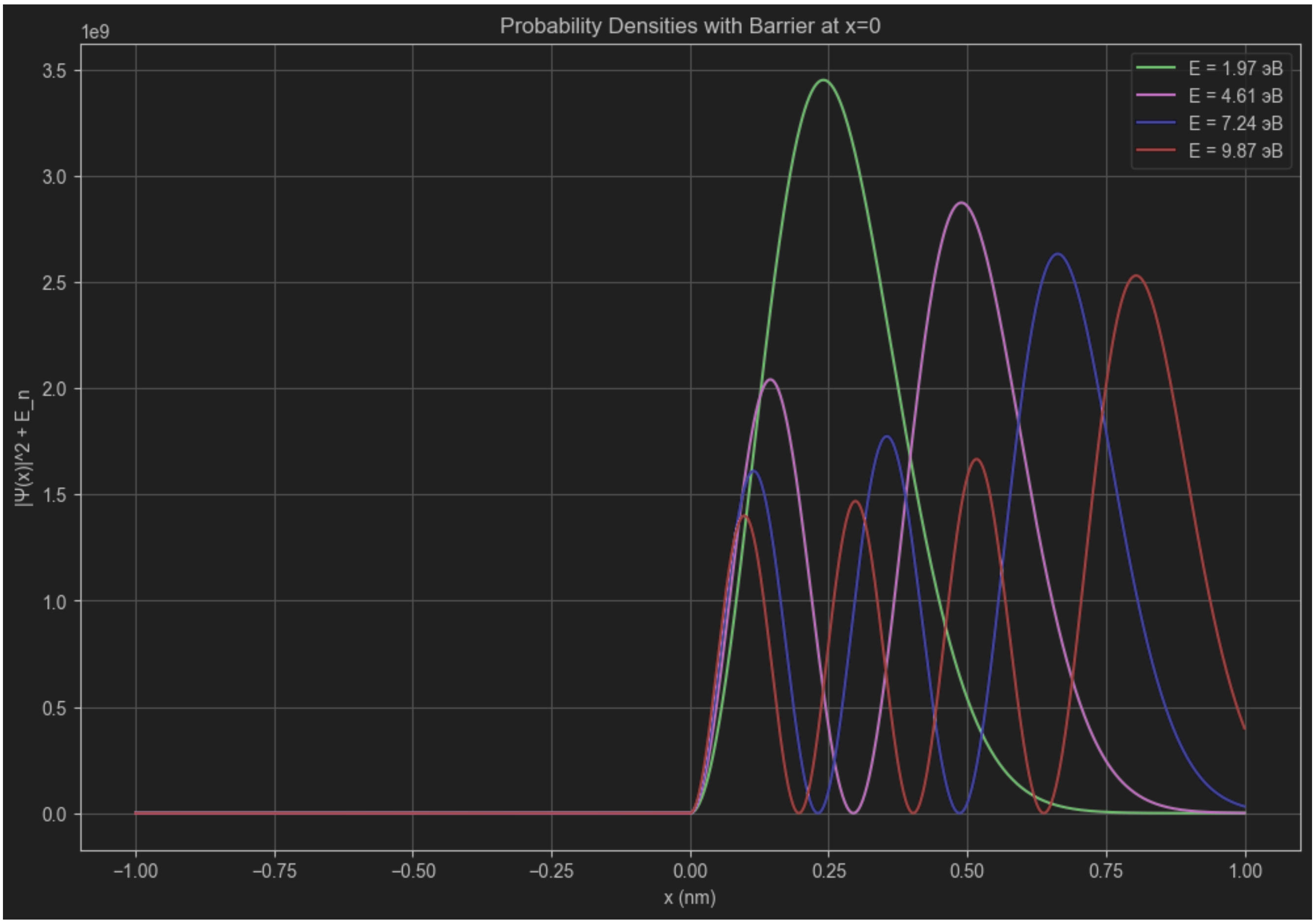




Probability Densities of the Harmonic Oscillator without Barrier
E = 0.66 эВ
E = 1.97 эВ
E = 3.29 эВ
E = 4.61 эВ
E = 5.92 эВ
Potential V(x)
|Ψ(x)|^2 + E_n
x (nm)



Wave Functions with Barrier at x=0
E = 1.97 эВ
E = 4.61 эВ
E = 7.24 эВ
E = 9.87 эВ
Ψ(x) + E_n
x (nm)



7) Выводы:

Прямоугольная потенциальная яма:

- Были найдены собственные значения энергии для связанных состояний частицы в прямоугольной потенциальной яме. Собственные значения зависят от параметров U, a, и массы частицы m.
- Для данного потенциала определены собственные функции, которые удовлетворяют граничным условиям, связанным с непрерывностью функции и её производной на границах ямы.

Гармонический осциллятор:

- Для гармонического потенциала $V(x) = \frac{1}{2}m\omega^2x^2$ были определены собственные значения энергии, имеющие вид $E_n = \hbar\omega\left(n + \frac{1}{2}\right)$ где n — квантовое число.
- Собственные функции являются полиномами Эрмита, умноженными на гауссовый множитель. Их графическое построение подтверждает ожидаемую симметрию и форму.

Влияние перегородки в гармоническом осцилляторе:

- При добавлении бесконечно узкой и полупроницаемой перегородки в точке x=0 наблюдается изменение энергетического спектра и собственных функций.
- Установлено, что перегородка приводит к разделению волновых функций на чётные и нечётные, влияя на распределение вероятности нахождения частицы.

Данное исследование демонстрирует основные свойства квантовых систем с разными потенциальными барьерами, включая возможность аналитического и численного анализа спектров энергии и форм волновых функций.