

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

Факультет прикладной математики, информатики и механики
Кафедра математического и прикладного анализа

ЛАБОРАТОРНАЯ РАБОТА №1

**ЧИСЛЕННОЕ РЕШЕНИЕ СТАЦИОНАРНОГО
УРАВНЕНИЯ ШРЁДИНГЕРА: МЕТОД
ПРИСТРЕЛКИ**

Направление: 01.04.02 - Прикладная математика и информатика
Профиль: Компьютерные технологии в задачах математической физики,
оптимизации и управления

Выполнил:
студент 11 группы 2 курса магистратуры
Маркин Р.О.
Преподаватель:
доктор физ.-мат. наук, профессор
Тимошенко Ю.К.

Воронеж 2024

Содержание

1	Цели и задачи работы	3
2	Одномерное стационарное уравнение Шрёдингера. Математический формализм. Общие свойства решений	4
3	Метод пристрелки. Алгоритм	5
4	Программная реализация алгоритма	8
5	Результаты численных экспериментов и их анализ	9
	Приложение 1. Компьютерный код	13
	Список литературы	17

1. Цели и задачи работы

Цели работы.

Целями лабораторной работы являются практическое освоение информации, полученной при изучении курса «Компьютерное моделирование в математической физике» по теме «Численное решение стационарного уравнения Шрёдингера», а также развитие алгоритмического мышления и приобретение опыта использования знаний и навыков по математике, численным методам и программированию для решения прикладных задач физико-технического характера.

Задачи работы.

Проблема: электрон находится в одномерной потенциальной яме с бесконечными стенками $U(x)$:

$$v(x) = \begin{cases} J_2(x), & x \in (-L, L); \\ \infty, & x \notin (-L, L), \end{cases}$$

где $U(x) = v(x) * V_0$, $V_0 = 25$ эВ, $L = 3 \text{ \AA}$, $J_n(x)$ — функция Бесселя, $n = 2$.

1) Используя метод пристрелки, найти энергии, нормированные волновые функции и плотности вероятности для основного и 2-го возбужденного состояний. Привести как численные значения энергий, так и построить графики волновых функций и плотностей вероятности.

2) Вычислить для этих состояний квантовомеханические средние $\langle p(x) \rangle$ и $\langle p(x^2) \rangle$.

2. Одномерное стационарное уравнение Шрёдингера. Математический формализм. Общие свойства решений

Одномерное стационарное уравнение Шрёдингера[1]

$$\hat{H}\psi(x) = E\psi(x) \quad (1)$$

с математической точки зрения представляет собой задачу определения собственных значений E и собственных функций $\psi(x)$ оператора Гамильтона \hat{H} . Для частицы с массой m , находящейся в потенциальном поле $U(x)$, оператор Гамильтона имеет вид

$$\hat{H} = \hat{T} + U(x), \quad (2)$$

где оператор кинетической энергии

$$\hat{T} = -\frac{\hbar^2}{2m} \frac{d^2}{dx^2}, \quad (3)$$

а \hbar — постоянная Планка. Собственное значение оператора Гамильтона имеет смысл энергии соответствующей изолированной квантовой системы. Собственные функции называются волновыми функциями. Волновая функция однозначна и непрерывна во всем пространстве. Непрерывность волновой функции и её 1-й производной сохраняется и при обращении $U(x)$ в ∞ в некоторой области пространства. В такую область частица вообще не может проникнуть, то есть в этой области, а также на её границе, $\psi(x) = 0$.

Оценим нижнюю границу энергетического спектра. Пусть минимальное значение потенциальной функции равно U_{\min} . Очевидно, что $\langle T \rangle \geq 0$ и $\langle U \rangle \geq U_{\min}$. Потому, из уравнения (1) следует, что

$$E = \langle H \rangle = \int_{-\infty}^{\infty} \psi^*(x) \hat{H} \psi(x) dx = \langle T \rangle + \langle U \rangle > U_{\min}. \quad (4)$$

То есть, энергии всех состояний $> U_{\min}$.

Особый практический интерес представляет случай, когда

$$\lim_{x \rightarrow \pm\infty} U(x) = 0. \quad (5)$$

Потенциал такого типа называется также потенциальной ямой. Для данной $U(x)$ свойства решений уравнения Шрёдингера зависят от знака собственного значения E .

Если $E < 0$. Частица с отрицательной энергией совершает финитное движение. Оператор Гамильтона имеет дискретный спектр, то есть собственные значения и соответствующие собственные функции можно снабдить номерами. При $E < 0$ уравнение (1) приобретает вид [1]:

$$\hat{H}\psi_k(x) = E_k\psi_k(x). \quad (6)$$

Квантовое состояние, обладающее наименьшей энергией, называется основным. Остальные состояния называют возбуждёнными состояниями. В силу линейности стационарного уравнения Шрёдингера, волновые функции математически определены с точностью до постоянного множителя. Однако, из физических соображений, волновые функции должны быть нормированы следующим образом

$$\int_{-\infty}^{+\infty} |\psi_k(x)|^2 dx = 1. \quad (7)$$

В дальнейшем будет рассматриваться только дискретный спектр. При этом необходимо пользоваться осцилляционной теоремой.

Осцилляционная теорема. Упорядочим собственные значения оператора Гамильтона в порядке возрастания, нумеруя энергию основного состояния индексом «0»: $E_0, E_1, E_2, \dots, E_k, \dots$. Тогда волновая функция $\psi_k(x)$ будет иметь k узлов (то есть, пересечений с осью абсцисс). Исключения: области, в которых потенциальная функция бесконечна.

3. Метод пристрелки. Алгоритм

Прежде чем перейти к алгоритму пристрелки нужно преобразовать уравнение (1). В данном случае граничные условия для волновой функции [1]:

$$\psi(a) = \psi(b) = 0. \quad (8)$$

где в точках а и б по оси абсцисс построены бесконечные потенциальные стенки. Для решения уравнения Шрёдингера удобно использовать атомные единицы Хартри ($e = 1$, $\hbar = 1$ и $m_e = 1$). В этих единицах уравнение (6), предполагая, что $m = m_e$, приобретает вид [1]:

$$\left[-\frac{1}{2} \frac{d^2}{dx^2} + U(x) \right] \psi(x) = E\psi(x). \quad (9)$$

Преобразуем (11) к форме

$$\frac{d^2\psi(x)}{dx^2} + q(E, x)\psi(x) = 0, \quad (10)$$

где

$$q(E, x) = 2[E - U(x)]. \quad (11)$$

Решение стационарного уравнения Шрёдингера сводится к нахождению собственных значений и собственных функций оператора Гамильтона, так как для собственных значений известна оценка снизу (4), то удобно начинать с вычисления энергии и волновой функции основного состояния. Оценим грубо энергию основного состояния $E_0^{(0)} = U_{\min} + \delta$, где δ — малая величина ($\delta > 0$). Подставим значение этой энергии в уравнение (9). Это уравнение теперь становится обыкновенным дифференциальным уравнением 2-го порядка с граничными условиями (8). Рассмотрим алгоритм, использующий эту идею.

Зададим на интервале $[a, b]$ сетку из N узлов с постоянным шагом $h = \frac{(b-a)}{(N-1)}$:

$$x_n = a + (n - 1)h, \quad n = 1, 2, 3, \dots, N. \quad (12)$$

Граничные условия (8) приобретают вид

$$\psi_1 = \psi_N = 0. \quad (13)$$

Задачи Коши для дифференциального уравнения (9) часто решается методом Нумерова. В рамках этого метода значения функции в узле сетки находят интегрируя «вперёд»:

$$\psi_{n+1} = [2(1 - 5cq_n)\psi_n - (1 + cq_{n-1})\psi_{n-1}] (1 + cq_{n+1})^{-1}, \quad (14)$$

либо интегрируя «назад»:

$$\psi_{n-1} = [2(1 - 5cq_n)\psi_n - (1 + cq_{n+1})\psi_{n+1}] (1 + cq_{n-1})^{-1}, \quad (15)$$

Здесь $c = h^2/12$, $q_n = q(E, x_n)$. При использовании формулы (14) необходимо знать ψ_1 и ψ_2 , а формулы (15) — ψ_{N-1} и ψ_N . Значения ψ_1 и ψ_N нам известны (13), а ψ_2 и ψ_{N-1} — нет. Однако, если N достаточно велико, то для простоты можно считать, что $\psi_2 = d_1$, $\psi_{N-1} = d_2$, где d_1, d_2 — малые числа).

Для оценки близости E к собственному значению будем вычислять разность производных волновых функций, полученных интегрированием «вперёд» и «назад», в некотором внутреннем узле сетки x_m :

$$f(E) = \left. \frac{d\psi_>(x)}{dx} \right|_{x_m} - \left. \frac{d\psi_<(x)}{dx} \right|_{x_m}. \quad (16)$$

где

$$\left. \frac{d\psi}{dx} \right|_{x_m} = \frac{\psi(x_m - 2h) - \psi(x_m + 2h) + 8[\psi(x_m + h) - \psi(x_m - h)]}{12h}. \quad (17)$$

Здесь $\psi_>, \psi_<$ — волновые функции, полученные интегрированием «вперёд» и «назад» соответственно; x_m — узел сшивки производных. Естественно, перед вычислением (16) необходимо масштабировать функции $\psi_>$ и $\psi_<$ так, чтобы $\psi_>(x_m) = \psi_<(x_m)$.

Будем увеличивать энергию с шагом ΔE до тех пор, пока величины $f^{(i)}$ на двух соседних шагах i и $i - 1$ не будут иметь различные знаки. Далее для уточнения собственного значения с наперед заданной точностью ϵ используется метод бисекции.

4. Программная реализация алгоритма

В Приложении представлена программа на языке Python 3.0 [2], реализованная в среде разработки PyCharm Community Edition 2024.2.1, численного решения одномерного стационарного уравнения Шредингера для электрона в одномерной потенциальной яме. Программа реализует алгоритм пристрелки, позволяющий находить собственные значения и соответствующие им волновые функции. Потенциальная функция и параметры для нее соответствуют постановке задачи из первой главы. Энергия и длина ямы были переведены в атомные единицы Хартри (строки 143-146)

В строках 142-158 инициализируются параметры из постановки задачи, такие как количество точек системы n , параметры потенциальной функции L , V_0 , задается приближенное значение энергии E_{min} и шаг $step$, также задается узел сшивки g .

В строках 5-9 реализована потенциальная функция $U(x)$ из постановки задачи.

В строках 74-85 реализован метод пристрелки соответствующий алгоритму из 3 главы.

В строках 63-72 и 87-99 реализованы вспомогательные функции `energy scan` и `bisection method`, `energy scan` вычисляет все значения энергий на отрезке $[E_{min}, E_{max}]$ с шагом $step$, функция `bisection method` реализует метод бисекции.

В строках 35-61 реализована функция в которой методом Нумерова вычисляются волновые функции интегрированием вперед и назад, и вычисляется разность производных этих функций в узле сшивки. В строках 11-12 реализована часть уравнения Шредингера(11) и в строках 14-15 реализована формула (17) для вычисления производной в узле сшивки.

В строках 17-19 реализована функция нормировки волновой функции.

В строках 21-33 реализовано вычисление квантовомеханических средних по формуле $\langle P_x \rangle = \int_a^b \Psi_n(x) \hat{P}_x \Psi_n(x) dx = -i\hbar \int_a^b \Psi_n(x) \frac{d\Psi_n(x)}{dx} dx$.

В строках 101-139 реализована функция для вывода графиков и записи данных в файл.

В строке 159 вызывается функция пристрелки и вычисляются все состояния на заданном диапазоне.

5. Результаты численных экспериментов и их анализ

Потенциал из постановки задачи представлен на Рис. 1.

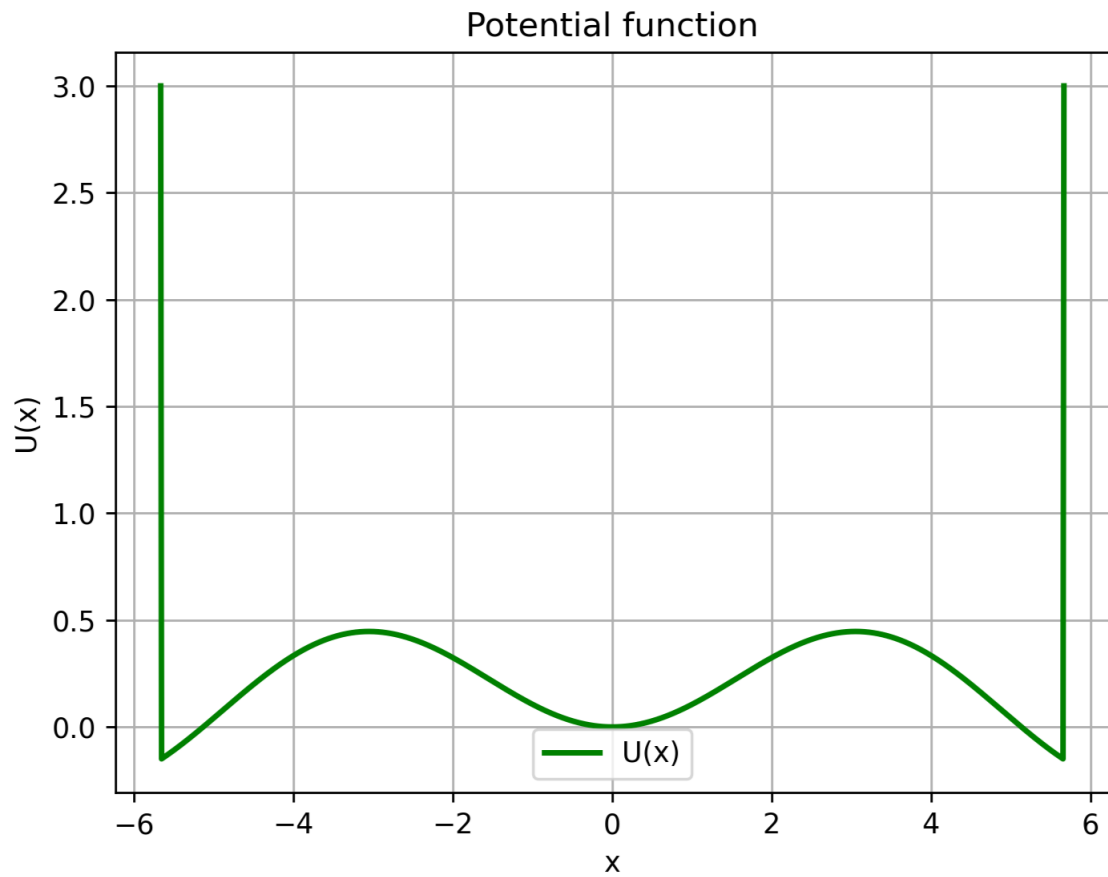


Рис. 1. Потенциальная функция

Для основного состояния я получил энергию $E=0.2005$ и следующую волновую функцию и плотность вероятности::

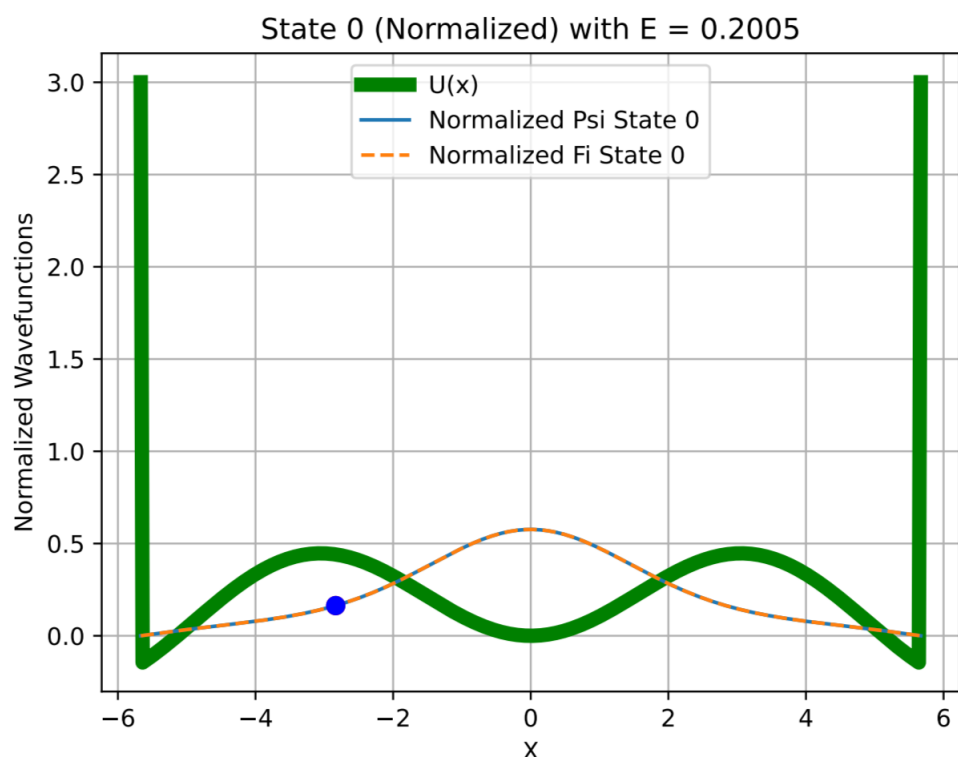


Рис. 2. Волновая функция основного состояния

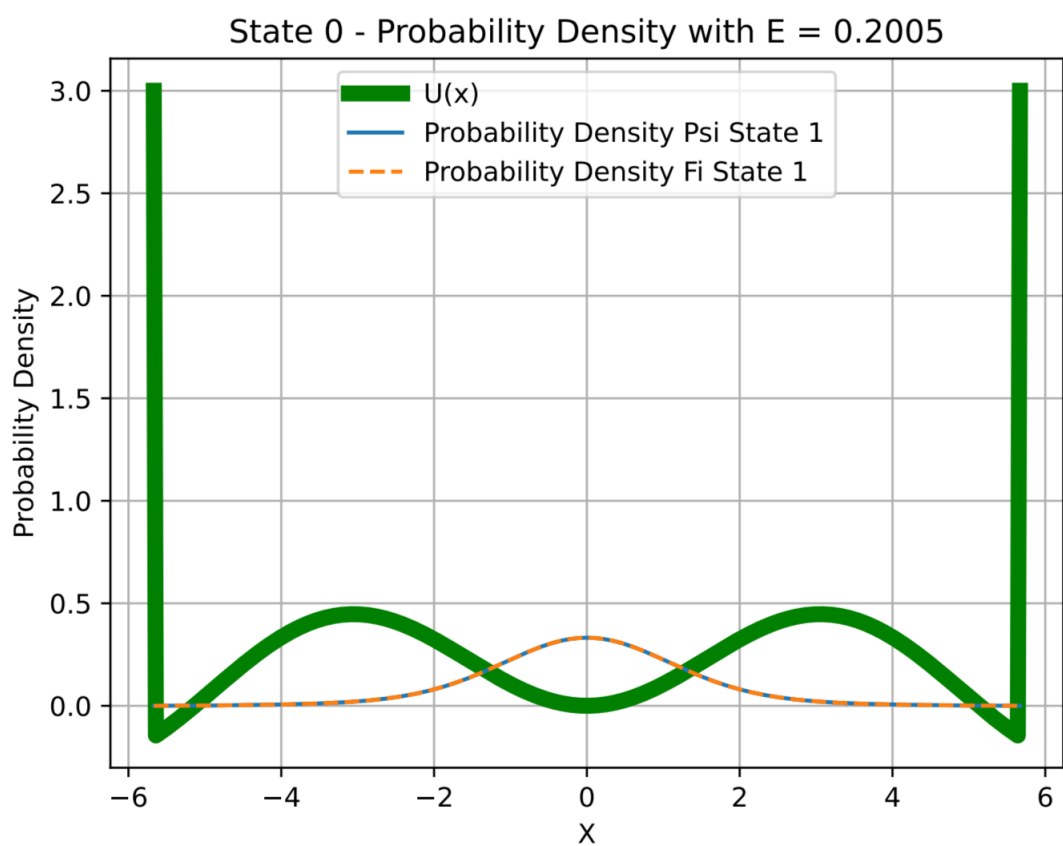


Рис. 3. Плотность вероятности основного состояния

На графиках можно видеть, что функции согласно осциляционной тео-

реме соответствуют основному состоянию так как нету пересечений с осью абсцисс, кроме бесконечных потенциальных стенок которые не учитываются.

Для второго возбужденного состояния я получил энергию $E=0.6141$ и следующую волновую функцию и плотность вероятности:

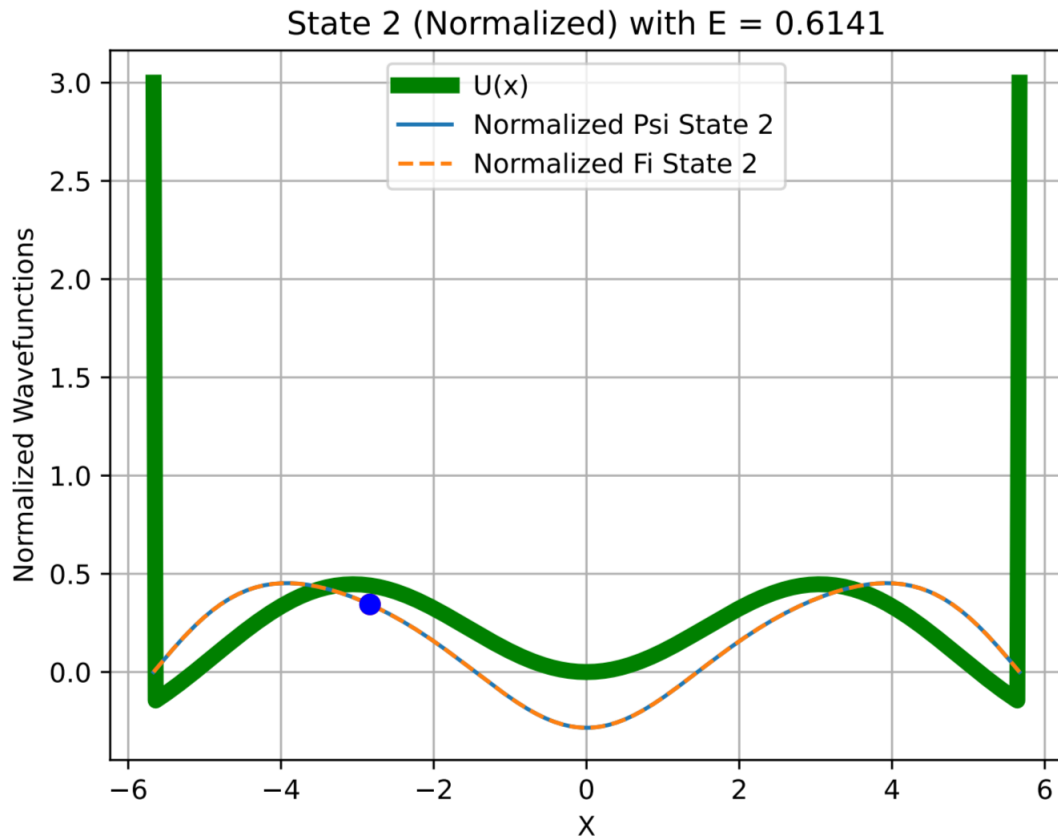


Рис. 4. Волновая функция второго возбужденного состояния

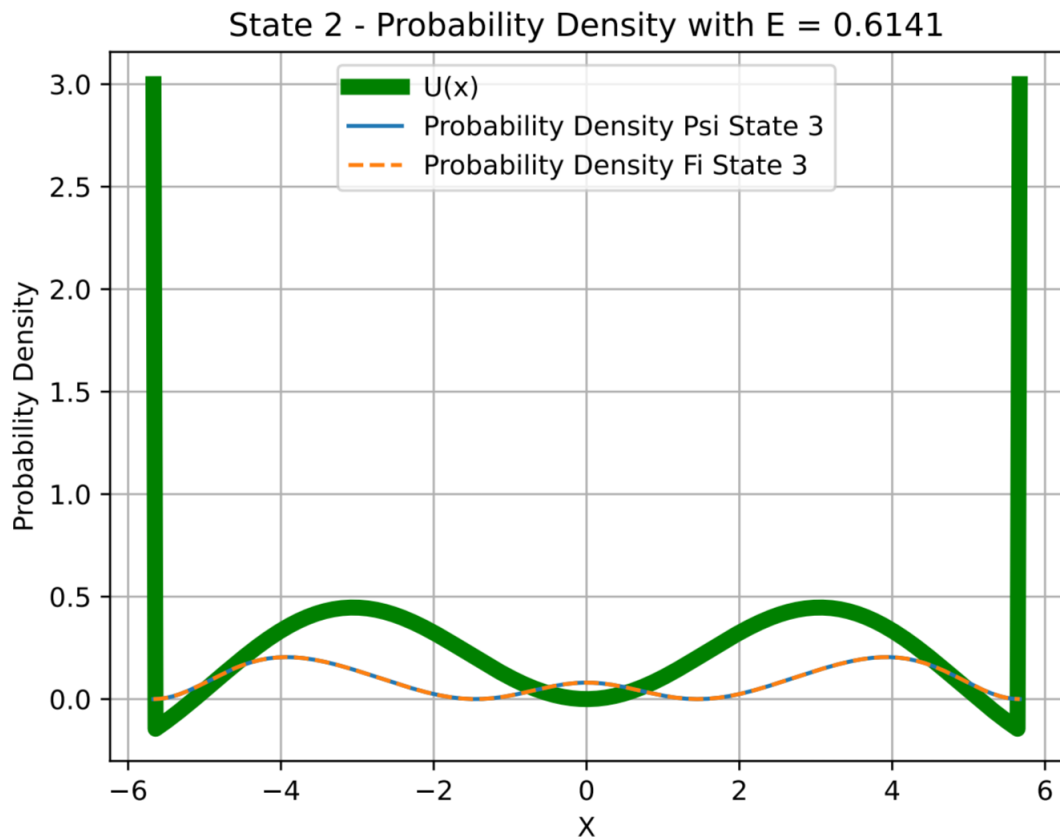


Рис. 5. Плотность вероятности второго возбужденного состояния

На графиках можно видеть, что функции согласно осциляционной теореме соответствуют второму возбужденному состоянию так как есть два пересечения с осью абцисс, кроме бесконечных потенциальных стенок которые не учитываются.

Квантовомеханические средние $\langle p(x) \rangle$ и $\langle p(x^2) \rangle$ для основного, первого и второго возбужденного состояний:

```
State 0: E = 0.200481, <p_x> = 0.000000e+00, <p_x^2> = 1.551882e-01
State 1: E = 0.476680, <p_x> = 0.000000e+00, <p_x^2> = 3.352033e-01
State 2: E = 0.614087, <p_x> = 0.000000e+00, <p_x^2> = 6.514036e-01
```

Закключение.

Таким образом, было получено численное решение для задачи о частице в одномерной квантовой яме с бесконечными стенками при помощи метода пристрелки. Были получены значения энергий и волновые функции основного и второго возбужденного состояний. Полученные волновые функции соответствуют осциляционной теореме. Кроме того для каждого состояния были вычислены квантовомеханические средние $\langle p(x) \rangle$, $\langle p(x^2) \rangle$

Приложение 1. Компьютерный код

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3  from scipy.special import jn
4
5  def U(x):
6      if abs(x) < L:
7          return V0 * jn(2, x)
8      else:
9          return W
10
11  def q(e, x):
12      return 2.0 * (e - U(x))
13
14  def deriv(Y, h, m):
15      return (Y[m - 2] - Y[m + 2] + 8.0 * (Y[m + 1] - Y[m - 1])) / (12.0 * h)
16
17  def normalize_wavefunction(Y):
18      norm = np.sqrt(np.trapz(Y**2, X))
19      return Y / norm
20
21  def mean_momentum(Psi, X):
22      hbar = 1.0
23      dPsi_dx = np.gradient(Psi, X)
24      integrand = Psi.conj() * dPsi_dx
25      mean_Px = -1j * hbar * np.trapz(integrand, X)
26      return mean_Px.real
27
28  def mean_square_momentum(Psi, X):
29      hbar = 1.0
30      d2Psi_dx2 = np.gradient(np.gradient(Psi, X), X)
31      integrand = Psi.conj() * d2Psi_dx2
32      mean_Px2 = -hbar**2 * np.trapz(integrand, X)
33      return mean_Px2.real
34
35  def f_fun(e, n):
36      F = np.array([c * q(e, X[i]) for i in np.arange(n)])
37      Psi[0] = 0.0
38      Fi[n - 1] = 0.0
39      Psi[1] = d1
40      Fi[n - 2] = d2
41
42      for i in np.arange(1, n - 1, 1):
```

```

43         p1 = 2.0 * (1.0 - 5.0 * F[i]) * Psi[i]
44         p2 = (1.0 + F[i - 1]) * Psi[i - 1]
45         Psi[i + 1] = (p1 - p2) / (1.0 + F[i + 1])
46
47     for i in np.arange(n - 2, 0, -1):
48         f1 = 2.0 * (1.0 - 5.0 * F[i]) * Fi[i]
49         f2 = (1.0 + F[i + 1]) * Fi[i + 1]
50         Fi[i - 1] = (f1 - f2) / (1.0 + F[i - 1])
51
52     p1 = np.abs(Psi).max()
53     p2 = np.abs(Psi).min()
54     big = p1 if p1 > p2 else p2
55
56     Psi[:] = Psi[:] / big
57
58     coef = Psi[r] / Fi[r]
59     Fi[:] = coef * Fi[:]
60
61     return deriv(Psi, h, r) - deriv(Fi, h, r)
62
63 def energy_scan(E_min, E_max, step):
64     energies = []
65     values = []
66     E = E_min
67     while E <= E_max:
68         f_value = f_fun(E, n)
69         energies.append(E)
70         values.append(f_value)
71         E += step
72     return energies, values
73
74 def find_exact_energies(E_min, E_max, step, tol):
75     energies, values = energy_scan(E_min, E_max, step)
76     exact_energies = []
77     for i in range(1, len(values)):
78         Log1 = values[i] * values[i - 1] < 0.0
79         Log2 = np.abs(values[i] - values[i - 1]) < porog
80         if Log1 and Log2:
81             E1, E2 = energies[i - 1], energies[i]
82             exact_energy = bisection_method(E1, E2, tol)
83             f_fun(exact_energy, n)
84             exact_energies.append(exact_energy)
85     return exact_energies
86
87 def bisection_method(E1, E2, tol):

```

```

88         while abs(E2 - E1) > tol:
89             Emid = (E1 + E2) / 2.0
90             f1, f2, fmid = f_fun(E1, n), f_fun(E2, n), f_fun(Emid, n)
91             if f1 * fmid < 0.0:
92                 E2 = Emid
93             else:
94                 E1 = Emid
95             if f2 * fmid < 0.0:
96                 E1 = Emid
97             else:
98                 E2 = Emid
99         return (E1 + E2) / 2.0
100
101     def plot_wavefunctions(Energies):
102         for i, E in enumerate(Energies):
103             f_fun(E, n)
104             Psi_norm = normalize_wavefunction(Psi.copy())
105             Fi_norm = normalize_wavefunction(Fi.copy())
106             mean_Px = mean_momentum(Fi_norm, X)
107             mean_Px2 = mean_square_momentum(Fi_norm, X)
108             file1 = open("result.txt", "a")
109             print(f"State {i}: E = {E:.6f}, <p_x> = {mean_Px:.6e}, <p_x^2> = {
110                 mean_Px2:.6e}")
111             print(f"State {i}: E = {E:.6f}, <p_x> = {mean_Px:.6e}, <p_x^2> = {
112                 mean_Px2:.6e}", file = file1)
113
114             plt.figure(figsize=(15, 11))
115             plt.scatter(X[r], Psi_norm[r], color='red', s=50, zorder=5)
116             plt.scatter(X[r], Fi_norm[r], color='blue', s=50, zorder=5)
117             plt.plot(X, [U(x) for x in X], 'g-', linewidth=6.0, label="U(x)")
118             plt.plot(X, Psi_norm, label=f"Normalized Psi State {i}")
119             plt.plot(X, Fi_norm, '--', label=f"Normalized Fi State {i}")
120             plt.title(f"State {i} (Normalized) with E = {E:.4f}")
121             plt.xlabel("X")
122             plt.ylabel("Normalized Wavefunctions")
123             plt.grid(True)
124             plt.legend()
125             plt.savefig(f"State {i} (Normalized).pdf", dpi=300)
126             plt.show()
127
128             prob_density_psi = Psi_norm**2
129             prob_density_fi = Fi_norm**2
130             plt.figure(figsize=(15, 11))
131             plt.plot(X, [U(x) for x in X], 'g-', linewidth=6.0, label="U(x)")

```

```

131         plt.plot(X, prob_density_psi, label=f"Probability Density Psi State
           {i+1}")
132         plt.plot(X, prob_density_fi, '--', label=f"Probability Density Fi
           State {i+1}")
133         plt.title(f"State {i} - Probability Density with E = {E:.4f}")
134         plt.xlabel("X")
135         plt.ylabel("Probability Density")
136         plt.grid(True)
137         plt.legend()
138         plt.savefig(f"State {i} (Probability Density).pdf", dpi=300)
139         plt.show()
140
141
142     U_min = -0.149124
143     cenergy = 27.212
144     clength = 0.5292
145     V0 = 25.0 / cenergy
146     L = 3.0 / clength
147     A, B = -L, L
148     n = 401
149     h = (B - A) / (n - 1)
150     c, W = h ** 2 / 12.0, 3.0
151     Psi, Fi, X = np.zeros(n), np.zeros(n), np.linspace(A, B, n)
152     r = (n-1)//2 - 100
153     porog = 4.0
154
155     d1, d2 = 1.e-09, 1.e-09
156     tol = 1e-6
157
158     E_min, E_max, step = U_min + 0.01, 3.0, 0.01
159     exact_energies = find_exact_energies(E_min, E_max, step, tol)
160
161     if len(exact_energies) == 0:
162         print("Error: not find energies.")
163     else:
164         print("Energies:")
165         for i, E in enumerate(exact_energies):
166             print(f"State {i}: Energy = {E:.6f}")
167
168     plot_wavefunctions(exact_energies)

```


Список литературы

1. Тимошенко Ю.К. Численное решение стационарного уравнения Шредингера: метод пристрелки. Учебное пособие. Воронеж: Научная книга, 2019. 35 с.
2. Доля П.Г. Введение в научный Python. Харьков: ХНУ, 2016. 265 с.