Санкт-Петербургский Государственный Электротехнический Университет "ЛЭТИ"

кафедра физики

Задание №4 по дисциплине

"Физические основы информационных технологий"

Название: Туннелирование электронов сквозь потенциальный барьер

|  |  |
| --- | --- |
| Фамилия И.О.: | Калмак Д.А. |
| группа: | 0303 |
| Преподаватель: | Альтмарк А.М. |
| Итоговый балл: |  |
|  |  |

Крайний срок сдачи: 10.12.22

Санкт-Петербург 2022

Условие задания 4

Электрон c энергией  туннелирует сквозь потенциальный барьер шириной *d*, форма потенциальной энергии  которого задана на рисунке в соответствии с вариантом. Определить коэффициент прохождения *D* электрона сквозь барьер. Исходные данные брать из таблицы. В качестве результата также необходимо представить график волновой функции для трех областей.

Коэффициент прохождения *D* записать в файл result.txt в папке LR4. Помимо текстового файла result.txt в папке LR4 должен находиться Word-файл с отчетом, а также файл с кодом (Python, Mathcad, Mathematica). Для лучшего понимания отчетности смотрите папку “Пример организации яндекс-папки студентов”.

Пример содержания файла result.txt:

4.53258

Таблица 1 - Исходные данные

|  |  |  |  |
| --- | --- | --- | --- |
| Вариант |  | *d* | рисунок |
| 9. | 0.1 | 0.5 | Рис.1 |

|  |
| --- |
|  |
| Рис. 1. Парабола |

**Выполнение работы.**

Волновая функция: , где – волновая функция для i-ой зоны, , – энергия электрона, – потенциальная энергия среды, – координата электрона. До начала потенциального барьера , а после потенциального барьера .

Для решения задачи туннелирования выполняются следующие равенства, если - постоянна:

Разобьем наш барьер на 100 частей, тогда можно взять постоянной. Решая систему уравнений, найдем . Коэффициент прохождения *D* электрона сквозь барьер находится по формуле . D = 0.274680903113051. Решая систему уравнений с известными и , найдем . Для графика волновой функции до потенциального барьера необходимы и . Для графика волновой функции после потенциального барьера необходимы и . График волновой функции представлен на рис. 2.

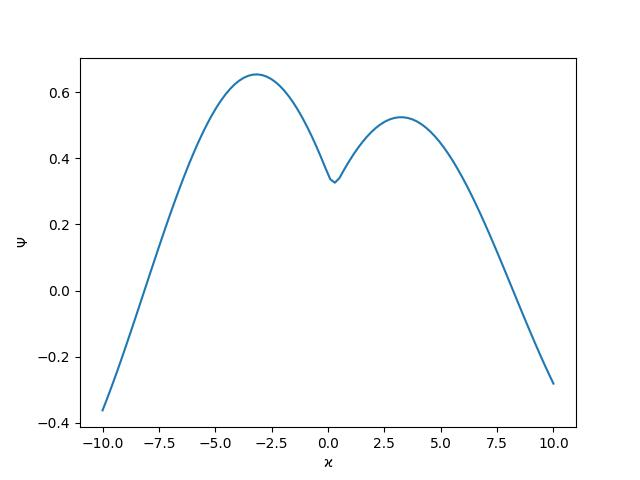


Рисунок 2 – График волновой функции

Разработанный код представлен в Приложении А.

**Вывод.**

Таким образом, был вычислен коэффициент прохождения D электрона сквозь барьер и построен график волновой функции.

**ПРИЛОЖЕНИЕ А  
ИСХОДНЫЙ КОД ПРОГРАММЫ**

import matplotlib.pyplot as plt

import numpy as np

import cmath

import sympy as sp

e = 0.1

d = 0.5

N = 100

delta = d / N

ae = np.linspace(-10, 10, 100)

zones = N + 1

barrs = [i \* delta for i in range(zones)]

def f(ae):

return ae \*\* 2 \* 0.1 + 1.9

def e\_p(ae):

return f(ae) if 0 < ae <= d else 0

def gamma(ae):

return 1j \* cmath.sqrt(e - e\_p(ae))

a0, b0, aI, bI, aI1, bI1, aN, bN = sp.symbols('a0, b0, aI, bI, aI1, bI1, aN, bN')

a0 = aI

b0 = bI

for i in range(zones):

aei = barrs[i]

g = gamma(aei - delta / 2)

g1 = gamma(aei + delta / 2)

eq1 = aI \* sp.exp(g \* aei) + bI \* sp.exp(-g \* aei) - aI1 \* sp.exp(g1 \* aei) - bI1 \* sp.exp(-g1 \* aei)

eq2 = g \* aI \* sp.exp(g \* aei) - g \* bI \* sp.exp(-g \* aei) - g1 \* aI1 \* sp.exp(g1 \* aei) + g1 \* bI1 \* sp.exp(-g1 \* aei)

answ = sp.solvers.solve((eq1, eq2), (aI, bI))

a0 = a0.subs(answ)

a0 = sp.simplify(a0.subs({aI1: aI, bI1: bI}))

b0 = b0.subs(answ)

b0 = sp.simplify(b0.subs({aI1: aI, bI1: bI}))

resA0 = sp.simplify(a0.subs({aI: aN, bI: bN}))

resB0 = sp.simplify(b0.subs({aI: aN, bI: bN}))

a0, b0, aN, bN = sp.symbols('a0, b0, aN, bN')

A\_0 = 1

B\_N = 0

eq1 = (resA0 - a0).subs({a0: A\_0, bN: B\_N})

eq2 = (resB0 - b0).subs({a0: A\_0, bN: B\_N})

answ = sp.solvers.solve((eq1, eq2), (b0, aN))

B\_0 = answ[b0]

A\_N = answ[aN]

D = (abs(A\_N)) \*\* 2

file = open("./result.txt", "w")

file.write(str(D))

file.close()

AB = [[A\_0, B\_0]]

for i in range(zones):

aei = barrs[i]

g = gamma(aei - delta / 2)

g1 = gamma(aei + delta / 2)

aI = AB[i][0]

bI = AB[i][1]

eq1 = aI \* sp.exp(g \* aei) + bI \* sp.exp(-g \* aei) - aI1 \* sp.exp(g1 \* aei) - bI1 \* sp.exp(-g1 \* aei)

eq2 = g \* aI \* sp.exp(g \* aei) - g \* bI \* sp.exp(-g \* aei) - g1 \* aI1 \* sp.exp(g1 \* aei) + g1 \* bI1 \* sp.exp(-g1 \* aei)

answ = sp.solvers.solve((eq1, eq2), (aI1, bI1))

AB.append([answ[aI1], answ[bI1]])

def psi(ae, d, AB, delta):

if ae < 0:

return sp.re(AB[0][0] \* sp.exp(gamma(ae) \* ae) + AB[0][1] \* sp.exp(-gamma(ae) \* ae))

elif ae > d:

return sp.re(AB[-1][0] \* sp.exp(gamma(ae) \* ae) + AB[-1][1] \* sp.exp(-gamma(ae) \* ae))

else:

return sp.re(AB[sp.floor(ae / delta) + 1][0] \* sp.exp(gamma(ae) \* ae) + AB[sp.floor(ae / delta) + 1][1] \* sp.exp(-gamma(ae) \* ae))

fig, ax = plt.subplots()

psi = [psi(ae, d, AB, delta) for ae in ae]

ax.plot(ae, psi)

ax.set\_xlabel(r'ϰ')

ax.set\_ylabel(r'$\Psi$')

fig.savefig('Graphic wave function.jpeg')