**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Отчет**

по моделированию №1 **«Расчет молекулы бензола методом Хюккеля»**

по дисциплине «**Дополнительные главы физики**»

Авторы: Войнов и Чечулин Львы

Факультет: ФИТИП

Группа: M32021

Преподаватель:

Шоев Владислав Иванович



Санкт-Петербург 2022

**Цель:** рассчитать уровни энергии и волновые функции молекулы бензола

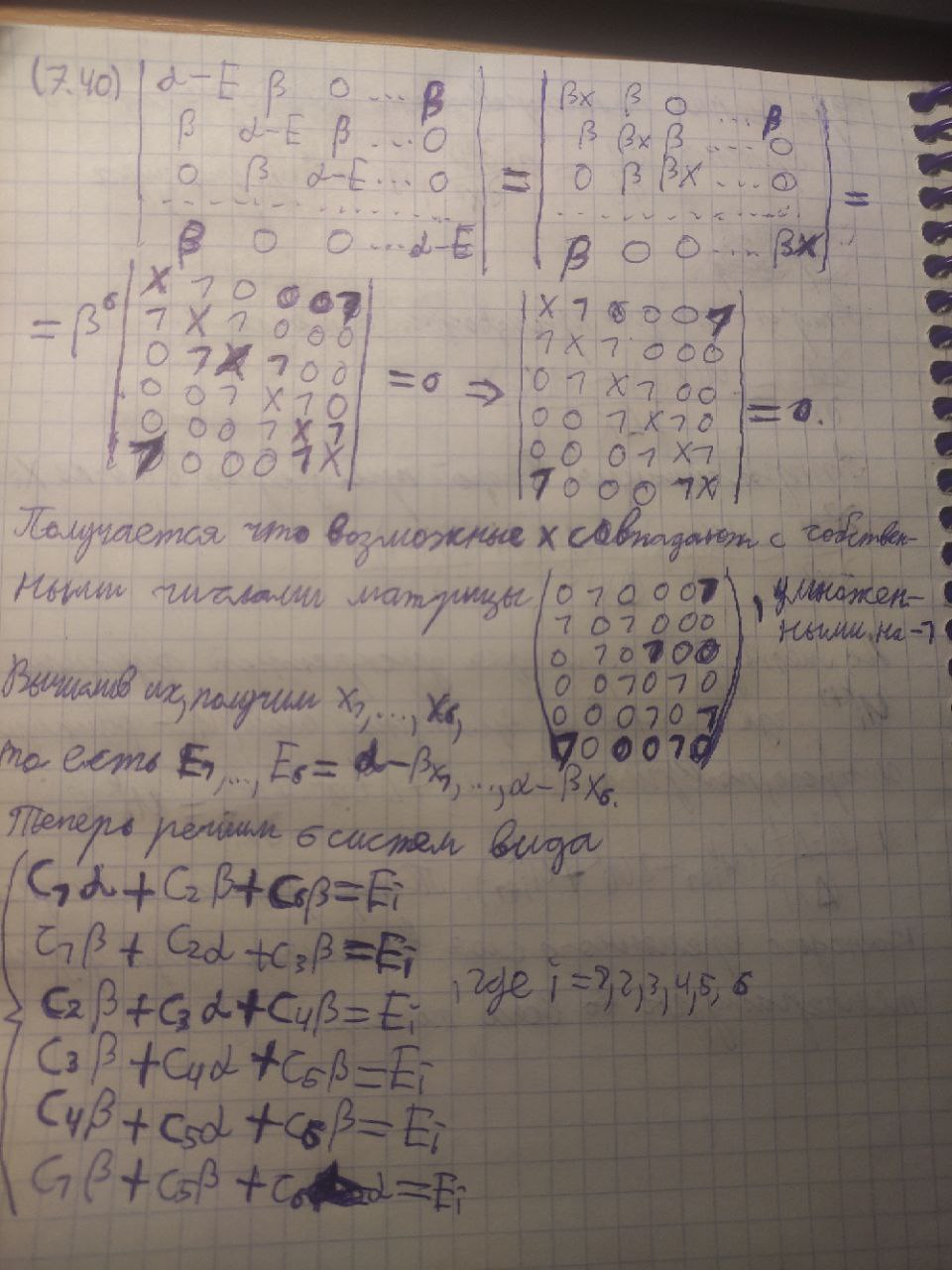
**Задачи:**

Составить алгоритм решения уравнения

Найти уровни энергии

Найти 6 наборов коэффициентов при волновой функции

**Теоретические вычисления:**



**Код:**

import numpy as np  
import matplotlib.pyplot as plt  
  
  
def diag(A):  
 n = A.shape[0]  
 L = np.zeros(n)  
 for i in range(n):  
 L[i] = A[i, i]  
 return L  
  
  
def Zero(n):  
 return np.empty((n, n), dtype=float)  
  
  
def One(n):  
 E = Zero(n)  
 for i in range(n):  
 E[i, i] = 1  
 return E  
  
  
def LU(A, L, U):  
 n = A.shape[0]  
 for i in range(n):  
 for j in range(n):  
 if i <= j:  
 s = 0.0  
 for k in range(i):  
 s += L[i, k] \* U[k, j]  
 U[i, j] = A[i, j] - s  
 else:  
 s = 0.0  
 for k in range(j):  
 s += L[i, k] \* U[k, j]  
 L[i, j] = (A[i, j] - s) / U[j, j]  
  
  
def LU\_Gauss(A, b):  
 n = A.shape[0]  
 L = One(n)  
 U = Zero(n)  
 LU(A, L, U)  
  
 y = np.zeros(n)  
 y[0] = b[0]  
 for i in range(1, n):  
 s = b[i]  
 for j in range(i):  
 s -= L[i, j] \* y[j]  
 y[i] = s  
  
 x = np.zeros(n)  
 nn = n - 1  
 x[nn] = y[nn] / U[nn, nn]  
 for i in range(1, n):  
 s = y[nn - i]  
 for j in range(i):  
 s -= U[nn - i, nn - j] \* x[nn - j]  
 x[nn - i] = s / U[nn - i, nn - i]  
 return x  
  
  
d = np.array([[0, 1, 0, 0, 0, 1],  
 [1, 0, 1, 0, 0, 0],  
 [0, 1, 0, 1, 0, 0],  
 [0, 0, 1, 0, 1, 0],  
 [0, 0, 0, 1, 0, 1],  
 [1, 0, 0, 0, 1, 0]])  
x = -np.linalg.eigh(d)[0]  
alpha = -11  
beta = -2.4  
e = alpha - beta \* x  
print("Energy levels:", e)  
  
A = np.array([[alpha, beta, 0, 0, 0, beta],  
 [beta, alpha, beta, 0, 0, 0],  
 [0, beta, alpha, beta, 0, 0],  
 [0, 0, beta, alpha, beta, 0],  
 [0, 0, 0, beta, alpha, beta],  
 [beta, 0, 0, 0, beta, alpha]])  
print("Wave functions:")  
for j in range(6):  
 ei = np.array([e[j], e[j], e[j], e[j], e[j], e[j]])  
 psi = LU\_Gauss(A, ei)  
 print("ψ =", psi[0], "ψ1 +", psi[1], "ψ2 +",  
 psi[2], "ψ3 +", psi[3], "ψ4 +",  
 psi[4], "ψ5 +", psi[5], "ψ6")  
  
for j in range(6):  
 plt.plot((j % 2, j % 2 + 1), (e[j], e[j]))  
plt.show()

**Результаты:**

Energy levels: [ -6.2 -8.6 -8.6 -13.4 -13.4 -15.8]

Wave functions:

ψ = 0.3924050632911392 ψ1 + 0.3924050632911393 ψ2 + 0.39240506329113917 ψ3 + 0.39240506329113933 ψ4 + 0.3924050632911393 ψ5 + 0.3924050632911394 ψ6

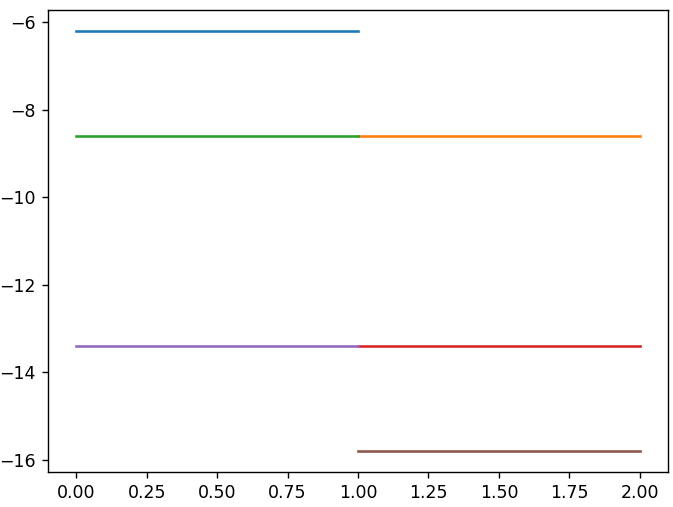
ψ = 0.5443037974683542 ψ1 + 0.5443037974683543 ψ2 + 0.5443037974683542 ψ3 + 0.5443037974683543 ψ4 + 0.5443037974683542 ψ5 + 0.5443037974683543 ψ6

ψ = 0.5443037974683544 ψ1 + 0.5443037974683544 ψ2 + 0.5443037974683543 ψ3 + 0.5443037974683544 ψ4 + 0.5443037974683544 ψ5 + 0.5443037974683544 ψ6

ψ = 0.8481012658227848 ψ1 + 0.8481012658227847 ψ2 + 0.8481012658227847 ψ3 + 0.8481012658227849 ψ4 + 0.8481012658227848 ψ5 + 0.8481012658227848 ψ6

ψ = 0.848101265822785 ψ1 + 0.848101265822785 ψ2 + 0.8481012658227849 ψ3 + 0.8481012658227849 ψ4 + 0.8481012658227849 ψ5 + 0.848101265822785 ψ6

ψ = 1.0 ψ1 + 1.0 ψ2 + 1.0 ψ3 + 1.0000000000000002 ψ4 + 1.0 ψ5 + 1.0000000000000002 ψ6



**Вывод:** Изучен метод Хюккеля, найдены и изображены уровни энергии, записаны волновые функции молекулы бензола.