Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа программной инженерии

**Курсовая работа**

по дисциплине «Математическое моделирование»

**Вариант 61**

Выполнил: Симоненко Иван Сергеевич

Группа: 5130904/20002

№ зач. Книжки: 22350270

Преподаватель: Устинов С. М.

Санкт-Петербург

2024

Оглавление

[1 Глава 3](#__RefHeading___Toc5391_1391575533)

[1.1 Подзаголовок 3](#__RefHeading___Toc5393_1391575533)

[1.2 Подзаголовок 3](#__RefHeading___Toc5395_1391575533)

[2 Глава 3](#__RefHeading___Toc5397_1391575533)

[2.1 Подзаголовок 3](#__RefHeading___Toc5399_1391575533)

[2.2 Подзаголовок 3](#__RefHeading___Toc5401_1391575533)

[3 Глава 3](#__RefHeading___Toc5403_1391575533)

[3.1 Подзаголовок 3](#__RefHeading___Toc5405_1391575533)

[3.2 Подзаголовок 3](#__RefHeading___Toc5407_1391575533)

# Постановка задачи

Изображение выглядит как текст, Шрифт, рукописный текст, снимок экрана

Автоматически созданное описание

# Аналитические преобразования

# Код программы

import numpy as np  
from math import exp  
  
p1 = 0.5  
# p3 = 20  
p4 = 14 # 8, 10, 12, 14  
p5 = 0.8  
p6 = 0  
x2\_start = -1.9  
x2\_end = 4.5  
step = 0.1  
signs = 3  
  
  
# Собственные значения для проверки: -0.846 и -0.365  
  
def calculate\_x1(x2, p1, p4, p5, p6):  
 x1 = (p1 \* x2 + p5 \* (x2 - p6)) / (p1 \* p4)  
 return x1  
  
  
def calculate\_p2(x2, x1, p1, p3):  
 p2 = (p1 \* x1) / ((1 - x1) \* exp(x2 / (1 + x2 / p3)))  
 return p2  
  
  
def calculate\_p2\_var2(x2, x1, p1):  
 p2 = (p1 \* x1) / ((1 - x1) \* exp(x2))  
 return p2  
  
  
x2\_values = []  
x1\_values = []  
p2\_values = []  
  
# Open the file in write mode  
with open(f'B) p4={p4}.txt', 'w') as f:  
 # Write the header  
 f.write("x2|x1|p2|eigv1|eigv2\n")  
  
 for x2 in np.arange(x2\_start, x2\_end + step, step):  
 x2\_values.append(x2)  
  
 x1 = calculate\_x1(x2, p1, p4, p5, p6)  
 x1\_values.append(x1)  
  
 # p2 = calculate\_p2(x2, x1, p1, p3)  
 p2 = calculate\_p2\_var2(x2, x1, p1)  
 p2\_values.append(p2)  
  
 # left\_top = -p1 - p2 \* e \*\* (x2 / (1 + x2 / p3))  
 # left\_btm = -p2 \* p4 \* e \*\* (x2 / (1 + x2 / p3))  
 # right\_top = p2 \* (1 - x1) \* (e \*\* ((p3 \* x2) / (p3 + x2)) \* p3 \*\* 2) / (p3 + x2) \*\* 2  
 # right\_btm = -p1 + p2 \* p4 \* (1 - x1) \* (e \*\* ((p3 \* x2) / (p3 + x2)) \* p3 \*\* 2) / (p3 + x2) \*\* 2 - p5  
   
 left\_top = -p1 - p2 \* exp(x2)  
 left\_btm = -p2 \* p4 \* exp(x2)  
 right\_top = p2 \* (1 - x1) \* exp(x2)  
 right\_btm = -p1 + p2 \* p4 \* (1 - x1) \* exp(x2) - p5  
 A = np.array([[left\_top, right\_top],  
 [left\_btm, right\_btm]])  
 eigenvalues = np.linalg.eigvals(A)  
  
 # Format the output string  
 output\_string = (f"{x2:.{signs}f}|{x1:.{signs}f}|{p2:.{signs}f}"  
 f"|{eigenvalues[0]:.{signs}f}|{eigenvalues[1]:.{signs}f}\n")  
  
 # Write the formatted string to the file  
 f.write(output\_string)