**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №1

по дисциплине: «Компьютерные системы конечноэлементных расчётов»

на тему: **«**Современные численные методы решения граничных задач**»**

Выполнил: студент гр. ИТП-31

Расшивалов Н.И.

Принял: ассистент Точко В.Н.

Гомель 2021

**Цель работы**: на примере решения задачи теплопроводности стержня

научиться использовать следующие численные методы.

**Задание:**

Для выбранного варианта необходимо:

1. Построить математическую модель физической системы. Рассматривается задача моделирования распределения температуры в тонком стержне заданной длины и сечения, изготовленного из различных материалов. На одном конце стержня (левом, согласно рисунка 1) происходит конвективный теплообмен с внешней средой, а на противоположном, подводится тепловой поток заданной интенсивности.

2. Разработать методику исследования математической модели методом

конечных разностей, методом граничных элементов и методом конечных элементов.

3. Разработать программное обеспечение, которое должно удовлетворять

следующим требованиям:

– обеспечить ввод исходных данных с помощью GUI;

– решать задачу методом конечных разностей, методом граничных элементов и методом конечных элементов;

– отобразить в виде графиков (двумерного и трёхмерного) результаты

решения;

– все результаты решения сохранять как в тестовые файлы (для претендующих на оценки 4-5), так и в файлы специальных форматов (для всех остальных).

4. Провести верификацию полученных результатов с помощью конечноэлементного комплекса ANSYS. Вывести график распределения температур по длине стержня.

5. Оформить отчет о проделанной работе.

На рисунке 1 представлена схема нейронной сети согласно варианту.

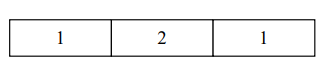


Рисунок 1 – Схема стержня согласно варианту

На рисунке 2 представлено задание к лабораторной работе.

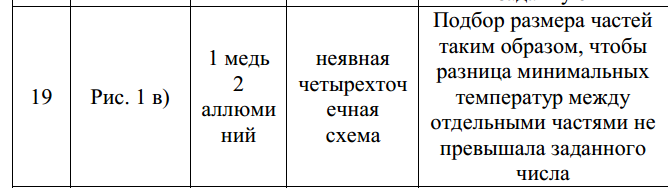


Рисунок 2 – Задание к лабораторной работе

**Ход работы**

На рисунке 3 представлен *GUI* программы.

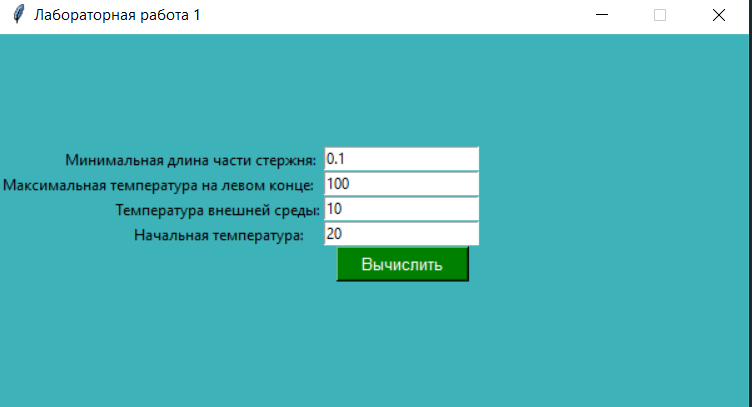


Рисунок 3 – *GUI* программы

На рисунке 4 представлен 2D график температур.

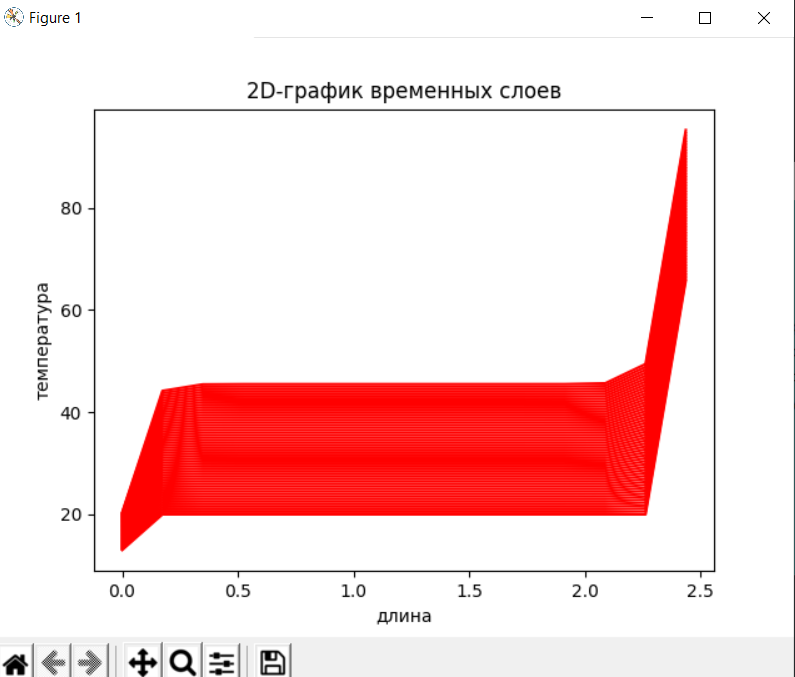


Рисунок 4 – 2D график температур

На рисунке 5 представлен 3D график температур.

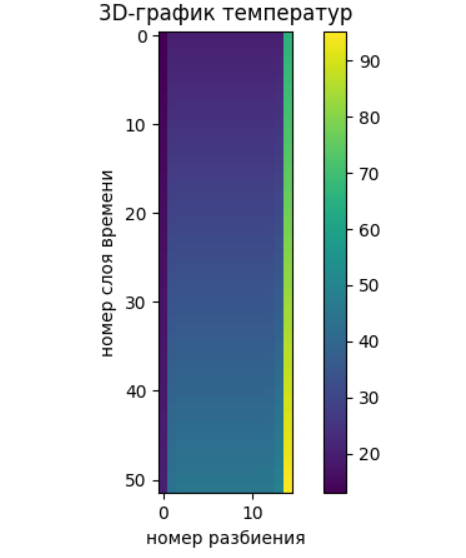


Рисунок 5 – 3D график температур

Текстовый файл с результатами вычислений представлен на рисунке 6.

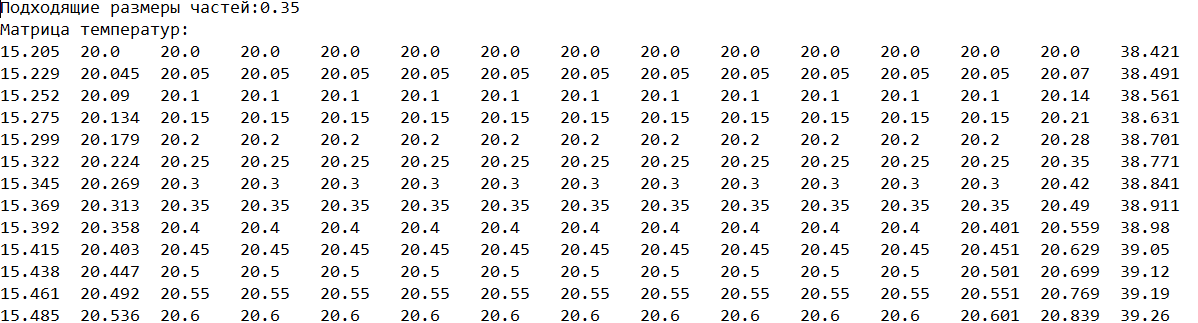


Рисунок 6 – Текстовый файл с результатами вычислений

Листинг программы представлен в приложении А.

**Вывод:** в ходе выполнения лабораторной работы изучено решение задачи тепловодности с визуализацией результатов с помощью графиков, а также сохранением результатов в различные файлы.

**ПРИЛОЖЕНИЕ А**

(обязательное)

**Листинг программы**

from tkinter import \*

import math

import matplotlib.pyplot as plt

import numpy as np

import xlsxwriter

def main():

    mainWindow = Tk()

    mainWindow.resizable(width=False, height=False)

    mainWindow.title("Лабораторная работа 1")

    screenWidth = mainWindow.winfo\_screenwidth()

    screenHeight = mainWindow.winfo\_screenheight()

    Lmin = StringVar(mainWindow) #минимальная длина части стержня

    LminBox = Entry(mainWindow,textvariable=Lmin)

    Lminlabel = Label(mainWindow, text="Минимальная длина части стержня:" , bg="#3db3b9")

    LminBox.insert(END, "0.1")

    Tmax = StringVar(mainWindow) #плотность теплового потока

    TmaxBox = Entry(mainWindow,textvariable=Tmax)

    Tmaxlabel = Label(mainWindow, text="Максимальная температура на левом конце:" , bg="#3db3b9")

    TmaxBox.insert(END, "100")

    Te = StringVar(mainWindow) #температура внешней среды

    TeBox = Entry(mainWindow,textvariable=Te)

    Telabel = Label(mainWindow, text="Температура внешней среды:" , bg="#3db3b9")

    TeBox.insert(END, "10")

    T0 = StringVar(mainWindow) #начальная температура

    T0Box = Entry(mainWindow,textvariable=T0)

    T0Label = Label(mainWindow, text="Начальная температура:" , bg="#3db3b9")

    T0Box.insert(END, "20")

    calcButton = Button(mainWindow,text="Вычислить", command=lambda:calc(int(N),float(tEnd),float(L),float(q),float(a),float(Te.get()),float(T0.get()),lLambda,mLambda,lRo,mRo,lC,mC,float(Lmin.get()),float(Tmax.get())),bg="green",fg="white",width="12", height="1",font=("MS Sans Serif", 10))

    LminBox.place(x=260, y=90)

    Lminlabel.place(x=50, y=90)

    TmaxBox.place(x=260, y=110)

    Tmaxlabel.place(x=0, y=110)

    TeBox.place(x=260, y=130)

    Telabel.place(x=90, y=130)

    T0Box.place(x=260, y=150)

    T0Label.place(x=105, y=150)

    calcButton.place(x=270, y=170)

    N = 5 #количество разбиений одной части

    tEnd = 30 #окончание по времени

    q = -100000 #плотность теплового потока

    a = 5000 #коэффицент теплообмена

    L = 0.3 #начальный размер стержня

    lLambda = 110 #коэффицент теплопроводности алюминия

    mLambda = 380 #коэффицент теплопроводности меди

    lRo = 8600 #плотность алюминия

    mRo = 8930 #плотность меди

    lC = 380 #теплоемкость алюминия

    mC = 400 #теплоемкость меди

    x = (screenWidth -600)/ 2

    y = (screenHeight -500)/ 2

    mainWindow.geometry('%dx%d+%d+%d' % (600, 300, x, y))

    mainWindow.configure(background="#3db3b9")

    mainWindow.mainloop()

def calc(N,tEnd,L,q,a,Te,T0,lLambda,mLambda,lRo,mRo,lC,mC,Lmin,Tmax):

    isSuit = False

    while isSuit == False:

        h = (Lmin\*3)/(N\*3)

        t = tEnd/60.0

        T = []

        lA = lLambda/(lRo\*lC)

        mA = mLambda/(mRo\*mC)

        amount = N\*3

        T.append([T0]\*amount) #начальный слой

        T[0][0] = (a\*Te+mLambda/h\*T[0][1])/(a+mLambda/h) #граничное условие 3 рода

        T[0][amount-1]=T[0][amount-2]-q\*h/mLambda #граничное условие 2 рода

        Yl = lA\*t/h\*\*2

        Ym = mA\*t/h\*\*2

        currentTime = t

        while currentTime < tEnd:

            T.append([T0]\*amount)

            n = len(T)-1

            for i in range(1,N):

                T[n][i] = Ym \* T[n-1][i-1] + (1 - 2\*Ym)\* T[n-1][i] + Ym \* T[n-1][i+1] + t

            for i in range(N,N\*2):

                T[n][i] = Yl \* T[n-1][i-1] + (1 - 2\*Yl)\* T[n-1][i] + Yl \* T[n-1][i+1] + t

            for i in range(N\*2,amount-1):

                T[n][i] = Ym \* T[n-1][i-1] + (1 - 2\*Ym)\* T[n-1][i] + Ym \* T[n-1][i+1] + t

            T[n][0] = (a\*Te+mLambda/h\*T[n][1])/(a+mLambda/h)

            T[n][amount-1] = T[n][amount-2]-q\*h/mLambda

            if n > 50 and math.fabs(T[n][4] - T[n][5]) < 0.0001  and math.fabs(T[n][9] - T[n][10]) < 0.0001:

                print("fabs "+str(T[n][4] - T[n][5])+" fabs2 "+str(T[n][9] - T[n][10]))

                isSuit = True

                break

            currentTime += t

        for i in range(len(T)):

            for j in range(len(T[0])):

                T[i][j] = round(T[i][j],3)

        if isSuit:

            print("Подходящие размеры частей:"+str(Lmin))

            f = open("results.txt", 'w')

            resultStr = "Подходящие размеры частей:"+str(round(Lmin,2))+"\n"

            resultStr += "Матрица температур:\n"

            workbook = xlsxwriter.Workbook('results.xlsx')

            worksheet = workbook.add\_worksheet()

            worksheet.write(0, 1,"Подходящие размеры частей:"+str(round(Lmin,2)))

            for i in range(len(T)):

                for j in range(len(T[0])):

                    resultStr += '%-8s' % str(T[i][j])

                    worksheet.write(i+1, j,T[i][j])

                resultStr += "\n"

            f.write(resultStr)

            f.close()

            workbook.close()

            create2DGraphic(T,N\*3,h)

            create3DGraphic(T)

            isSuit = True

        else:

            Lmin += 0.01

def create2DGraphic(T, N, h):

    lengths = []

    for i in range(N):

        lengths.append(h\*i)

    fig, ax = plt.subplots()

    for i in range(len(T)):

        ax.plot(lengths,T[i],'r')

    ax.set\_xlabel('длина')

    ax.set\_ylabel('температура')

    plt.title('2D-график временных слоев')

    plt.show()

def create3DGraphic(T):

    fig, ax = plt.subplots()

    plt.imshow(T, cmap='viridis')

    plt.colorbar()

    plt.title('3D-график температур')

    ax.set\_xlabel('номер разбиения')

    ax.set\_ylabel('номер слоя времени')

    plt.show()

main()