Министерство науки и высшего образования
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Факультет «Фундаментальные науки»

Кафедра «Высшая математика»

ОТЧЕТ

по технологической практике за 1 семестр 2020-2021 учебного года

Руководитель практики, ст. преп. кафедры ФН1	___________________	Кравченко О. В.
	подпись, дата	*фамилия, инициалы*
Студент группы ФН1-51Б	___________________	Ириневич С. Г.
	подпись, дата	*фамилия, инициалы*

Москва, 2020 г.

Цели и задачи практики

Цель технологической практики: развитие навыков работы с программным обеспечением, необходимых для успешного освоения материала бакалавриата и применяющихся в будущей профессиональной деятельности, развитие навыков решения задач прикладной математики численными методами.

Задачи технологической практики:

- Знакомство с программными средствами, необходимыми в будущей профессиональной деятельности.
- Развитие навыков численного решения прикладных математических задач.
- Знакомство с возможностями системы контроля версий GitHub.
- Развитие умения поиска необходимой информации в специальной литературе и других источниках.
- Развитие навыков составления отчетов и презентации результатов.

В период с сентября 2020 года по январь 2021 года мною была проделана работа по изучению методов вычислительной математики, совершенствованию навыков программирования и созданию программ.

Постановка задачи

Решить сингулярную двухточечную краевую задачу методом конечных разностей

$$\varepsilon y''(x) + p(x)y'(x) + q(x)y(x) + f(x) = 0, \quad x \in [0,1]$$

со смешанными граничными условиями

$$-\alpha_1 y'(0) + \alpha_2 y(0) = \gamma_1,$$

$$\beta_1 y'(1) + \beta_2 y(1) = \gamma_2,$$

при различных значениях параметра

$$\varepsilon = 1,\ 0.1,\ 0.01,\ 0.001.$$

11 вариант:

$$\alpha_1 = 1,$$

$$\alpha_2 = 0,$$

$$\beta_1 = 1,$$

$$\beta_2 = 0,$$

$$\gamma_1 = 3,$$

$$\gamma_2 = 6,$$

$$p(x) = \sin^2(x) + \cos(x),$$

$$q(x) = x^2 + x,$$

$$f(x) = x^3 - x^2 + x.$$

Описание метода

Математическое моделирование задач механики, физики и других отраслей науки и техники сводится к решению дифференциальных уравнений. Необходимость решать двухточечные краевые задачи для обыкновенных дифференциальных уравнений численными методами очень часто возникает в прикладных задачах во многих отраслях науки.

Метод конечных разностей относится к сеточным методам. Основное содержание метода заключается в следующем. Область непрерывного изменения аргумента (например, отрезок) заменяется дискретным множеством точек, которые называются узлами. Эти узлы составляют *разностную сетку*. Функции на непрерывном множестве точек заменяются их проекциями на эту сетку. Это позволяет перейти от пространства непрерывных функций к пространству дискретных функций.

Рассмотрим двухточечную краевую задачу для линейного дифференциального уравнения второго порядка:

$$y''(x) + p(x)y'(x) + q(x)y(x) = f(x), \quad x \in [a;b]$$

$$\alpha_1 y'(a) + \beta_1 y(a) = \gamma_1,$$

$$\alpha_2 y'(b) + \beta_2 y(b) = \gamma_2. \quad (*)$$

Определение производной непрерывной функции $y(x)$ заключается в следующем:

$$\frac{dy(x)}{dx} = \lim_{\Delta x \to 0} \frac{y(x+\Delta x) - y(x)}{\Delta x} = \lim_{\Delta x \to 0} \frac{y(x) - y(x-\Delta x)}{\Delta x} = \lim_{\Delta x \to 0} \frac{y(x+\Delta x) - y(x-\Delta x)}{2\Delta x}$$

В конечно-разностных методах производные и другие члены, входящие в дифференциальное уравнение, аппроксимируются на разностной сетке.

Введём равномерную сетку

$$\Omega_n = \{x_0, x_1, ..., x_n\},\ x_i = x_0 + ih,\ i = 0,...,n,\ h = \frac{b-a}{n}.$$

Размер шага сетки $h = \Delta x = x_{i+1} - x_i$ может быть очень мал, но всё равно остаётся конечным, в отличие от непрерывного случая, когда интервал Δx стремится к нулю.

Пусть точное решение $y(x)$ краевой задачи аппроксимируется на разностную сетку как $\hat{y}_i = y(x_i)$.

Тогда первая и вторая производная аппроксимируются на разностную сетку следующим образом:

$$y'(x) = \lim_{\Delta x \to 0} \frac{y(x+\Delta x) - y(x-\Delta x)}{2\Delta x} \approx \frac{\hat{y}_{i+1} - \hat{y}_{i-1}}{2h} = \hat{y}_i^{'}$$

$$y''(x) \approx \frac{\dfrac{\hat{y}_{i+1} - \hat{y}_i}{h} - \dfrac{\hat{y}_i - \hat{y}_{i-1}}{h}}{h} = \frac{\hat{y}_{i+1} - 2\hat{y}_i + \hat{y}_{i-1}}{h^2} = \hat{y}_i^{''}$$

Данные формулы для вычисления производных имеют *второй порядок точности* относительно размера шага h.

Функции $p(x)$, $q(x)$ и $f(x)$ заменяются их проекциями на разностную сетку $p_i = p(x_i)$, $q_i = q(x_i)$, $f_i = f(x_i)$.

Для краевых условий используются аппроксимационные формулы вида

$$\hat{y}_0^{'} = \frac{\hat{y}_1 - \hat{y}_0}{h},$$

$$\hat{y}_n^{'} = \frac{\hat{y}_n - \hat{y}_{n-1}}{h}.$$

Эти формулы имеют *первый порядок точности* относительно длины шага h, поэтому метод конечных разностей в целом в этом случае также будет иметь *первый порядок точности*. Из этих формул можно легко получить выражения для $\hat{y}_0$ и $\hat{y}_n$.

После подстановок от краевой задачи (*) переходим к *разностной схеме*, представляющей собой СЛАУ с трёхдиагональной матрицей:

$$\frac{\hat{y}_{i+1} - 2\hat{y}_i + \hat{y}_{i-1}}{h^2} + p_i \frac{\hat{y}_{i+1} - \hat{y}_{i-1}}{2h} + q_i \hat{y}_i = f_i, i = 1, \dots, n-1$$

$$c_i^1 \hat{y}_{i-1} + c_i^2 \hat{y}_i + c_i^3 \hat{y}_{i+1} = f_i, i = 1, \dots, n-1$$

$$c_i^1 = \frac{1}{h^2} - \frac{p_i}{2h},$$

$$c_i^2 = -\frac{2}{h^2} + q_i,$$

$$c_i^3 = \frac{1}{h^2} + \frac{p_i}{2h}.$$

Эту СЛАУ можно решить методом прогонки.

Реализация на языке Python

Краевые условия для данной задачи:

$$\begin{cases} -y'(0) = 3, \\ y'(1) = 6. \end{cases}$$

$$\begin{cases} y'(0) = -3, \\ y'(1) = 6. \end{cases}$$

$$\begin{cases} \dfrac{y_1 - y_0}{h} = -3, \\ \dfrac{y_n - y_{n-1}}{h} = 6. \end{cases}$$

```
from math import *
from numpy import linspace
import matplotlib.pyplot as plt
def p_x(x, e):
    return ((sin(x))**2+cos(x))/e

def q_x(x, e):
    return (x**2+x)/e

def f_x(x, e):
    return (-x**3+x**2-x)/e

def Solve_3diag(a, b, c, d):
    n = len(b)
    a.insert(0, 0)
    P = [0]*(n-1)
    Q = [0]*n
    P[0] = -c[0]/b[0]
    Q[0] = d[0]/b[0]
    for i in range(1, n-1):
        P[i] = c[i]/(-b[i] - a[i]*P[i-1])
        Q[i] = (a[i]*Q[i-1] - d[i])/(-b[i] - a[i]*P[i-1])
    x = [0]*n
    x[n-1] = (a[n-1]*Q[n-2] - d[n-1])/(-b[n-1] - a[n-1]*P[n-2])
    for i in range(n-2, -1, -1):
        x[i] = P[i]*x[i+1] + Q[i]
    return x

def Fin_Diff_Method(x, h, e, a, b):
    n = round((b - a)/h)
    c1, c2, c3, c4 = [], [], [], []
    c2.append(-1/h)
    c3.append(1/h)
    c4.append(-3)
```

```
    for i in range(1, n):
        p = p_x(x[i], e)
        q = q_x(x[i], e)
        f = f_x(x[i], e)
        c1.append(1/(h**2)-p/(2*h))
        c2.append(-2/(h**2)+q)
        c3.append(1/(h**2)+p/(2*h))
        c4.append(f)
    c1.append(-1/h)
    c2.append(1/h)
    c4.append(6)
    y = Solve_3diag(c1, c2, c3, c4)
    return y

def Delta(y1, y2):
    n = len(y1)
    dif = []
    for i in range(n):
        dif.append(abs(y1[i]-y2[2*i]))
    return max(dif)

def Error1(h, e, a, b): #Погрешность
    n = round((b - a)/h) #Целый шаг
    x = linspace(a, b, n+1)
    y11 = Fin_Diff_Method(x, h, e, a, b)
    h /= 2 #Половинный шаг
    n = round((b - a)/h)
    x = linspace(a, b, n+1)
    y12 = Fin_Diff_Method(x, h, e, a, b)
    h /= 2 #Четверть шага
    n = round((b - a)/h)
    x = linspace(a, b, n+1)
    y13 = Fin_Diff_Method(x, h, e, a, b)
    h /= 2 #Восьмая шага
    n = round((b - a)/h)
    x = linspace(a, b, n+1)
    y14 = Fin_Diff_Method(x, h, e, a, b)
    delta1 = Delta(y11, y12)
    delta2 = Delta(y12, y13)
    delta3 = Delta(y13, y14)
    print('e = ', e)
    print(delta1, delta2, delta3, sep='\n')
    print()

def Demonstrate(h, a=0, b=1):
    n = round((b - a)/h)
    x = linspace(a, b, n+1)
    y1 = Fin_Diff_Method(x, h, 1, a, b)
    y2 = Fin_Diff_Method(x, h, 0.1, a, b)
    y3 = Fin_Diff_Method(x, h, 0.01, a, b)
    y4 = Fin_Diff_Method(x, h, 0.001, a, b)
    Error1(h, 1, a, b)
    Error1(h, 0.1, a, b)
```

```
    Error1(h, 0.01, a, b)
    Error1(h, 0.001, a, b)
    plt.subplot(2, 3, 1)
    plt.plot(x, y1, color='red')
    plt.plot(x, y2, color='blue')
    plt.plot(x, y3, color='green')
    plt.plot(x, y4, color='orange')
    plt.legend(['eps = 1', 'eps = 0.1', 'eps = 0.01', 'eps =
0.001'])
    plt.subplot(2, 3, 2)
    plt.plot(x, y1)
    plt.title('eps = 1')
    plt.subplot(2, 3, 3)
    plt.plot(x, y2)
    plt.title('eps = 0.1')
    plt.subplot(2, 3, 4)
    plt.plot(x, y3)
    plt.title('eps = 0.01')
    plt.subplot(2, 3, 5)
    plt.plot(x, y4)
    plt.title('eps = 0.001')
    plt.show()
```

Результаты

Графики решений представлены на рисунках 1-5 (длина шага h=1e-4)

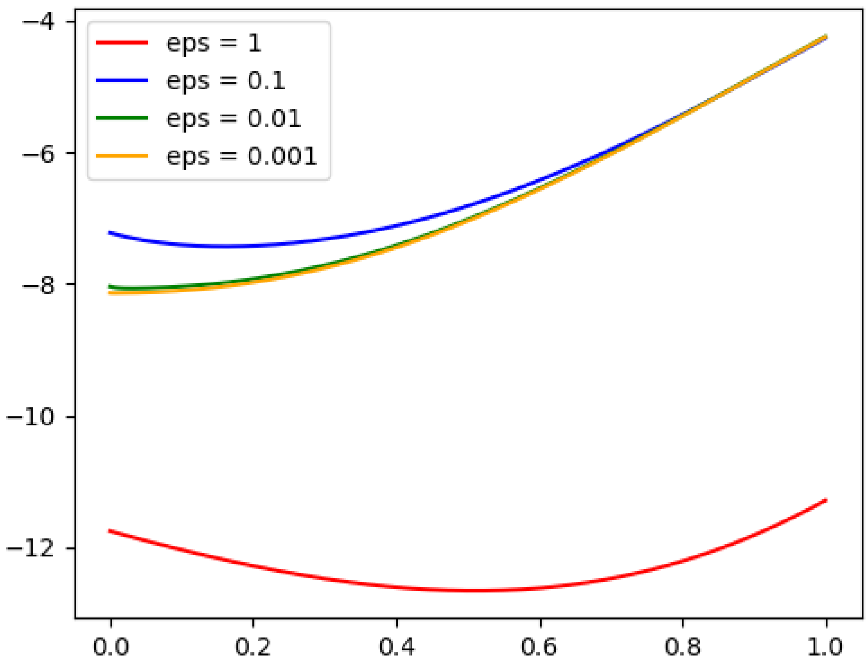

Рисунок 1

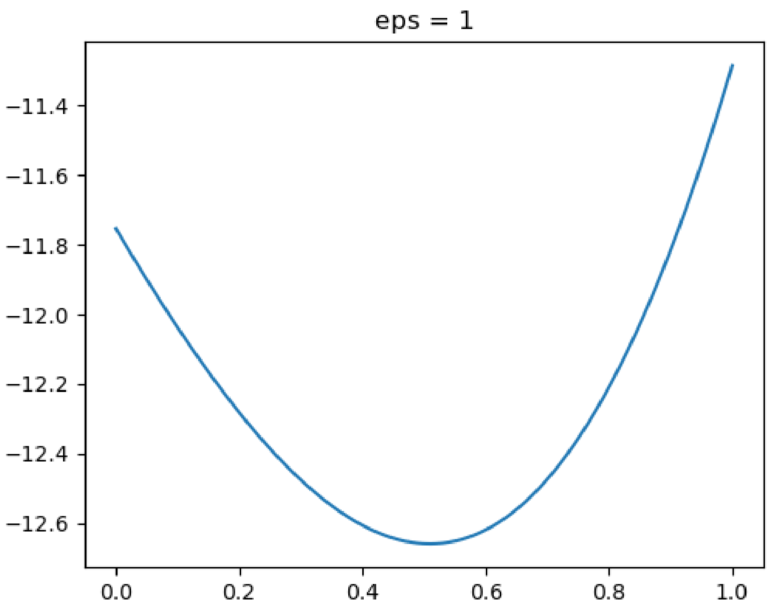

Рисунок 2

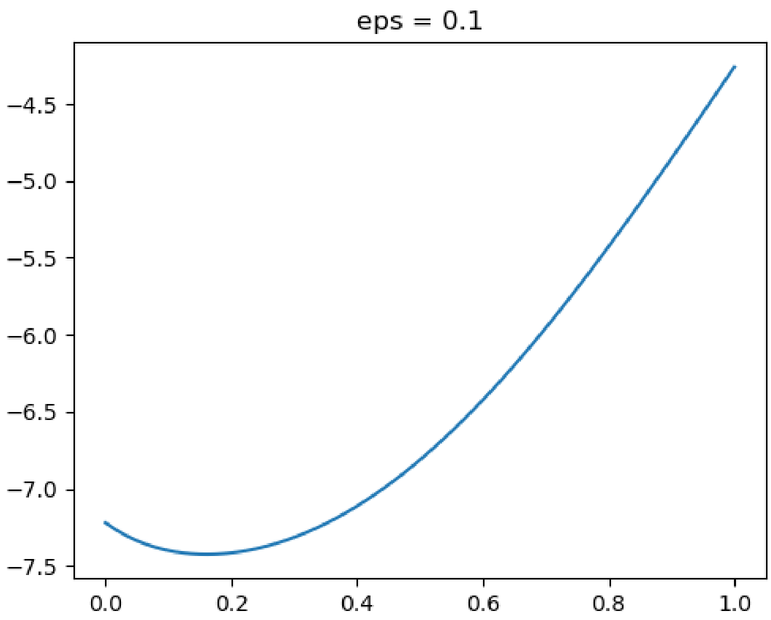

Рисунок 3

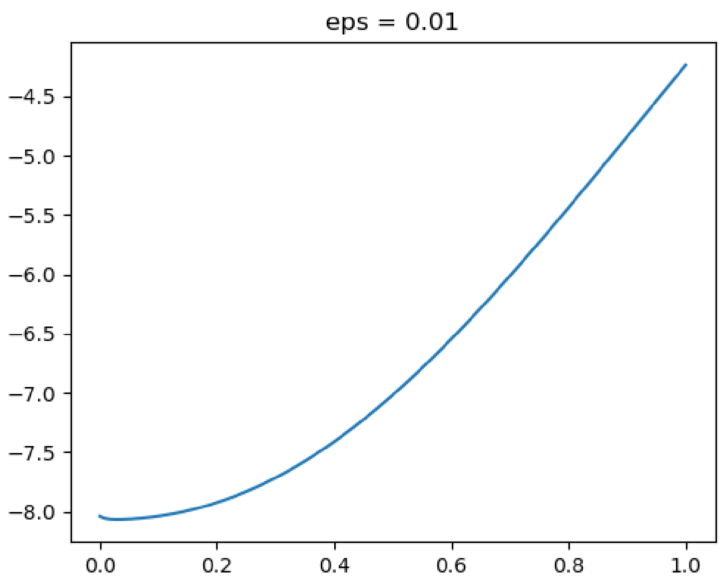

Рисунок 4

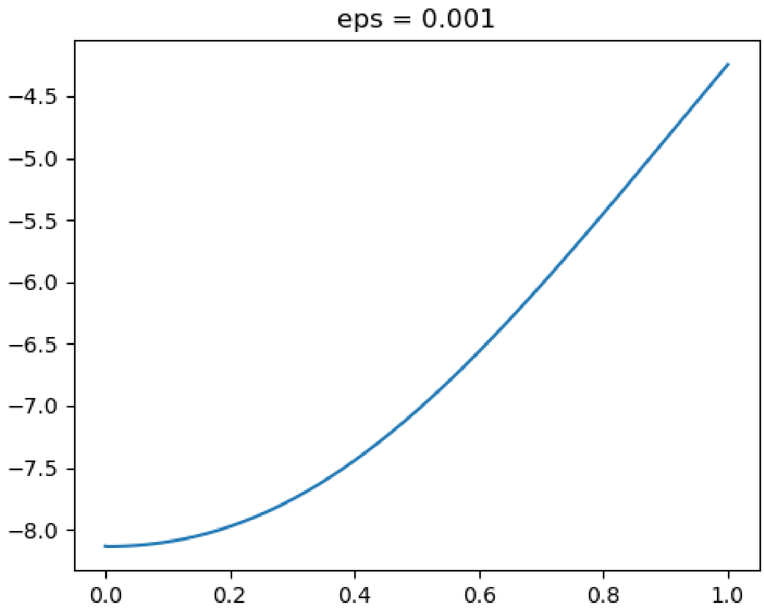

Рисунок 5

Оценка погрешности решения по *правилу Рунге* (правилу двойного пересчёта):

ε	1	0.1	0.01	0.001
h = 1e-4	0.00067028	6.6864e-05	0.00011347	0.00011721
h/2	0.00033511	3.3434e-05	5.6736e-05	5.8607e-05
h/4	0.00016751	1.6717e-05	2.8370e-05	2.9305e-05

Мы видим, что погрешность линейно зависит от длины шага (при уменьшении шага в 2 раза погрешность также уменьшается приблизительно в 2 раза).

Вывод

Необходимость решать дифференциальные уравнения численночень часто возникает в прикладных задачах во многих отраслях науки и техники.

В данной работе был реализован метод конечных разностей решения двухточечной краевой задачи для линейного дифференциального уравнения 2-го порядка. В этом методе отрезок (непрерывная область значений аргумента) заменяется дискретным множеством точек – сеткой, а функции и их производные аппроксимируются своими проекциями на эту сетку. Данная реализация имеет первый порядок точности.

Список литературы

1. ГОСТ 7.32-2001. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления.
2. Python 3.9.1 documentation. URL: https://docs.python.org/3/ (дата обращения: 04.02.2021).
3. Асташова И. В., Никишкин В. А. Практикум по курсу «Дифференциальные уравнения». Учебное пособие. Изд. 3-е, исправл. – М.: Изд. центр ЕАОИ, 2010. – 94 с., ил.
4. Гусев Е. В., Федотов А. А. Разностные методы решения гиперболических уравнений: Метод. указания к лабораторным работам по курсу «Численные методы» - М.: Изд-во МГТУ, 1991, 44 с., ил.
5. Киреев В. И., Пантелеев А. В. Численные методы в примерах и задачах: Учеб. пособие – 3-е изд., стер. – М.: Высш. шк., 2008. – 480 с.: ил.
6. Крайнов А. Ю., Моисеева К. М. Численные методы решения краевых задач для обыкновенных дифференциальных уравнений: учеб. пособие. – Томск: STT, 2016. – 44 с.
7. Мареев В. В., Станкова Е. Н. Основы методов конечных разностей. – СПб.: Изд-во С.-Петерб. ун-та, 2012. 64 с.
8. Морозова В.Д. Теория функций комплексного переменного: Учеб. для вузов / Под ред. В.С. Зарубина, А.П. Крищенко. — 3-е изд., исправл. — М.: Изд-во МГТУ им. Н.Э. Баумана, 2009. — 520 с.
9. Филиппов А. Ф. Сборник задач по дифференциальным уравнениям. – Ижевск: НИЦ «Регулярная и хаотическая динамика», 2000, 176 стр.