МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А.И. ГЕРЦЕНА»

Направление подготовки

09.03.01 – Информатика и вычислительная техника

Профиль «Технологии разработки программного обеспечения»

Лабораторная работа №1 и №2

«Численное интегрирование» и «Вычисление кратных интегралов»

Работу выполнили студенты 2 курса 2-1 группы:

Зухир Амира

Крючкова Анастасия

Стецук Максим

Максимова Ангелина

САНКТ-ПЕТЕРБУРГ

2022

СОДЕРЖАНИЕ

Зухир Амира ИВТ 2.1

Лабораторная работа №1

Численное интегрирование

Цель лабораторной работы: вычислить определенный интеграл, используя различные численные методы и алгоритмы их реализации.

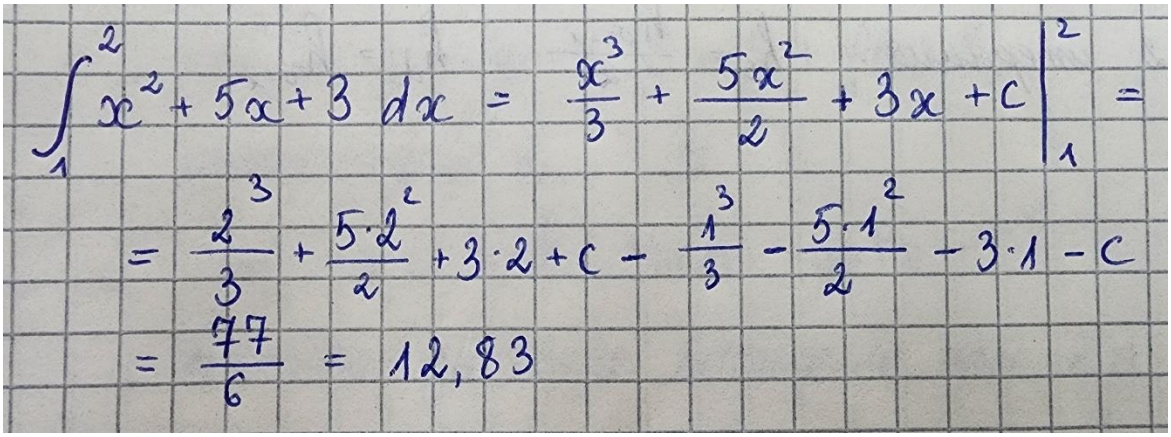
Инструменты: PyCharm, Telegram.

В рамках данной лабораторной работы, был использовал язык программирования Python 3.10

Мы использовали подынтегральную функцию: $x^2 + 5x + 3$.

В качестве примера определенного интеграла возьмем: $\int_1^2 x^2 + 5x + 3\, dx$

Вычислим данный интеграл ручным способом:



При вычислении, получили следующие результаты определенного интеграла с постоянным шагом:

Кол-во разбиений	Метод прямоугольников левых частей	Метод прямоугольников правых частей	Метод трапеций	Метод парабол
100	12.6242	12.7034	12.6642	12.4957
1000	12.8293	12.8373	12.8333	12.7993
10000	12.8329	12.8337	12.8333	12.8333

Консоль и Телеграм-бот выдают одинаковые результаты.

Код и работа Телеграм-бота приведена в приложении 1 и 2, соответственно.(страницы 6 и 21)

Вывод: в ходе сравнения машинных вычислений и ручного вычисления, мы пришли к выводу, что результаты практически идентичны и чем больше количество разбиений, тем точнее выводится результат.

Определенный интеграл с переменным шагом (2 алгоритма)

Точность	Двойной пересчет алгоритм 1	Двойной пересчет алгоритм 2
0.1	12.8255	12.8250
0.001	12.8323	12.8328
0.0001	12.8332	12.8332

Вывод: два данных алгоритма, прекрасно посчитали интеграл и сошли с результатом ручного вычисления, но в данном случае чем меньше точность, тем точнее результат.

Кратный интеграл

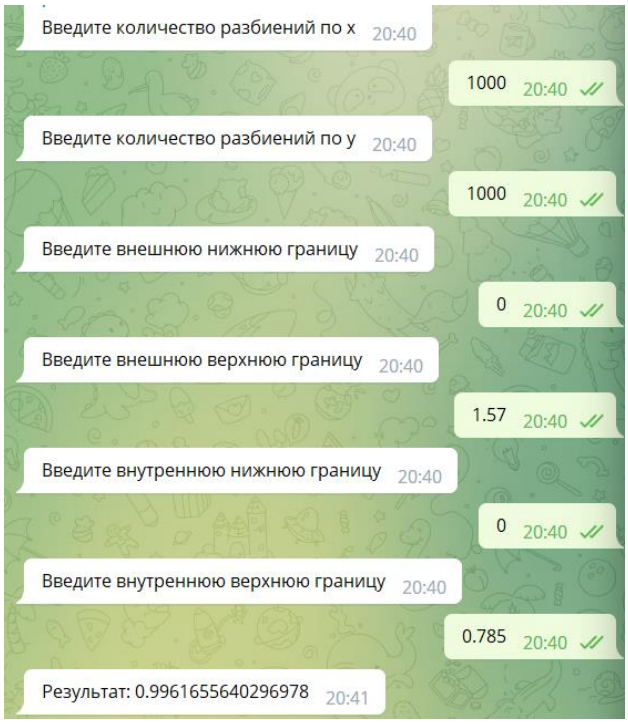
Возьмем кратный интеграл:

$$\int_0^{\pi/2} dx \int_0^{\pi/4} Sin(x+y)dy; \qquad n = 4$$

И вычислим его:

$$\int_0^{\pi/2} \int_0^{\pi/4} Sin(x+y)dxdy \approx \left(\frac{\pi}{24}\right)^2 + [\dots + \dots] = 1{,}00028$$

В нашей программе будет считать, что число пи это 3.14, соответственно пи деленное на 4 это 0.785, а пи деленное на 2 это 1.57.



Вывод: Результаты ручного вычисления и машинного приблизительно равны, погрешность возникла из-за примерного взятия числа пи, а также при увеличении кол-ва разбиений результат будет еще точнее.

Вывод по всей лабораторной работе: Мы научились реализовывать интегрирование и вычислили кратный интеграл, создали программу для данных вычислений и на ее основе создали Телеграм-бота.

Ссылка на Телеграм-бота: ссылка

Приложение 1

Код :

```
from math import sqrt
from math import sin

def task(x: float):
    return x*x+5*x+3

def KratInt(x:float, y:float):
    return sin(x+y)
```

С постоянным шагом:

1. Метод левых прямоугольников:

```
def left(n: int, A:float, B:float):
    h = (B - A) / n
    x = A
    result = 0
    while x <= (B - h):
        result += task(x)
        x += h
    return h * result
```

2. Метод правых прямоугольников:

```
def right(n: int, A:float, B:float):
    h = (B - A) / n
    x = A + h
    result = 0
    while x <= B:
        result += task(x)
        x += h
    return h * result
```

3. Матод трапеций:

```
def trapecia(n: int, A:float, B:float):
    h = (B - A) / n
    x = A + h
    result = 0
    while x <= (B - h):
        result += task(x)
        x += h
    return ((task(A) + task(B)) / 2 + result) * h
```

4. Метод парабол:

```
def parabola(n: int, A:float, B:float):
    h = (B - A) / n
    S1 = 0
    x = A + h
    while x <= (B - h):
        S1 += task(x)
        x += 2 * h
    S2 = 0
    x = A + (2 * h)
    while x <= (B - (2 * h)):
        S2 += task(x)
        x += 2 * h
    return (task(A) + task(B) + (4 * S1) + (2 * S2)) * (h / 3)
```

С переменным шагом:

1- Алгоритм 1:

```
def doubleRecalc(e:float, A:float, B:float):
    n = 2
    f1 = 0
    f2 = left(n, A, B)
    while abs(f1-f2) > e:
        f1 = f2
        n = n * 2
        f2 = left(n, A, B)
    return f2
```

2- Алгоритм 2:

```
def doubleRecalcBetter(e:float, A:float, B:float):
    n = 2
    otst = (B - A)/2
    f1 = left(n, A, B)
    otst = otst / 2
    f2 = left(n, A + otst, B)
    while abs(f1-f2) > e:
        f1 = f2
        n = n * 2
        otst = otst / 2
        f2 = left(n, A+otst, B)
    return f2
```

Кратный интеграл:

```
def kratniy(nx:int, ny:int, A:float, B:float, C:float, D:float):
    hx = (B - A) / nx
    hy = (D - C) / ny
    S = 0
    x = A
    while x <= (B - hx):
        y = C
        while y <= (D - hy):
            S += KratInt(x, y)
            y += hy
        x += hx
    return S*hx*hy
```

Меню:

```
special1 = True
while special1 == True:
    special2 = True
    print("Welcome to MainMenu")
    print("Для выбора задачи нажмите соответствующую кнопку:")
    print('[1]Численное интегрирование\n'
          'more will comming soon')
    mean1 = int(input())

    if mean1 == 1:
        while special2 == True:
            special3 = True
            print("Выберите:\n"
                  "[1] С постоянным шагом\n"
                  "[2] С переменным шагом\n"
                  "[3] Кратный интегралл")
            print("Для выхода в меню введите '4'")
            mean2 = int(input())

            if mean2 == 1:
                print("Выбран постоянный шаг\n")
                while special3 == True:
                    print("Выберите метод:\n"
                          "[1] Метод левых прямоугольников\n"
                          "[2] Метод правых прямоугольников\n"
                          "[3] Метод Симпсона(парабол)\n"
                          "[4] Метод трапеций")
                    print("Для возврата нажмите '5'")
                    mean3 = int(input())

                    if mean3 == 1:
                        print("Выбран 'Метод левых прямоугольников'")
                        print('Введите количество разбиений')
                        n = int(input())
                        print('Введите нижнюю границу')
                        A = float(input())
                        print('Введите верхнюю границу')
                        B = float(input())
                        res = left(n, A, B)
                        print("Результат: ", res, '\n')
```

```
            elif mean3 == 2:
                print("Выбран 'Метод правых прямоугольников'")
                print('Введите количество разбиений')
                n = int(input())
                print('Введите нижнюю границу')
                A = float(input())
                print('Введите верхнюю границу')
                B = float(input())
                res = right(n, A, B)
                print("Результат: ", res, '\n')

            elif mean3 == 3:
                print("Выбран 'Метод Симпсона(парабол)'")
                print('Введите количество разбиений')
                n = int(input())
                print('Введите нижнюю границу')
                A = float(input())
                print('Введите верхнюю границу')
                B = float(input())
                res = parabola(n, A, B)
                print("Результат: ", res, '\n')

            elif mean3 == 4:
                print("Выбран 'Метод трапеций'")
                print('Введите количество разбиений')
                n = int(input())
                print('Введите нижнюю границу')
                A = float(input())
                print('Введите верхнюю границу')
                B = float(input())
                res = trapecia(n, A, B)
                print("Результат: ", res, '\n')

            elif mean3 == 5:
                special3 = False
                print()
```

```
        elif mean2 == 2:
            print("Выбран переменный шаг\n")
            while special3 == True:
                print("Выберите алгоритм:\n"
                      "[1] Двойной пересчёт алгоритм 1\n"
                      "[2] Двойной пересчёт алгоритм 2")
                print("Для возврата нажмите '3'")
                mean4 = int(input())

                if mean4 == 1:
                    print("Двойной пересчёт'\n"
                          "Алгоритм 1\n"
                          "Метод левых прямоугольников \n")

                    print('Введите точность')
                    e = float(input())
                    print('Введите нижнюю границу')
                    A = float(input())
                    print('Введите верхнюю границу')
                    B = float(input())
                    res = doubleRecalc(e, A, B)
                    print("Результат: ", res, '\n')

                elif mean4 == 2:
                    print("Двойной пересчёт\n"
                          "Алгоритм 2\n"
                          "Метод левых прямоугольников\n")
                    print('Введите точность')
                    e = float(input())
                    print('Введите нижнюю границу')
                    A = float(input())
                    print('Введите верхнюю границу')
                    B = float(input())
                    res = doubleRecalcBetter(e, A, B)
                    print("Результат: ", res, '\n')

                elif mean4 == 3:
                    special3 = False
                    print()
```

```
        elif mean2 == 3:
            print("Выбран кратный интеграл\n")
            print("Введите количество разбиений по x")
            nx = int(input())
            print("Введите количество разбиений по y")
            ny = int(input())
            print("Введите внешнюю нижнюю границу")
            A = float(input())
            print("Введите внешнюю верхнюю границу")
            B = float(input())
            print("Введите внутреннюю нижнюю границу")
            C = float(input())
            print("Введите внутреннюю верхнюю границу")
            D = float(input())
            res = kratniy(nx, ny, A, B, C, D)
            print("Результат",res,'\n')

        elif mean2 == 4:
            special2 = False
```

Код для Телеграм-бота:

Код меню:

```python
import telebot
from telebot import types
from math import sin

bot = telebot.TeleBot('TOKEN')

# Команда /start
@bot.message_handler(commands=["start"])
def start(m):
    bot.send_message(m.chat.id, 'Привет! Я бот студентов РГПУ им. Герцена :) Я решаю задание для Лабораторной Работы №1✌ ')
    bot.send_message(m.chat.id, 'Мои команды:\n/start\n/info\n/calculate')
    bot.send_message(m.chat.id, 'Для начала вычисления интеграла выражения x*x+5*x+3 жми  /calculate!')

@bot.message_handler(commands=["info"])
def info(m):
    bot.send_message(m.chat.id, 'Я бот студентов РГПУ им. Герцена :) Моих создателей зовут Максим Стецук, Крючкова Анастасия и Зухир Амира. Они из группы ИВТ 2-1')
    markup = types.InlineKeyboardMarkup()
    item1 = types.InlineKeyboardButton("Максим", url='https://vk.com/makstulenchik')
    item2 = types.InlineKeyboardButton("Анастасия", url='https://vk.com/nestessia')
    item3 = types.InlineKeyboardButton("Амира", url='https://vk.com/amirazhr')
    item4 = types.InlineKeyboardButton("Ангелина", url='https://vk.com/mintange')
    markup.add(item1)
    markup.add(item2)
    markup.add(item3)
    markup.add(item4)
    bot.send_message(m.chat.id, "{0.first_name}, для связи с нами переходи по ссылкам :)".format(m.from_user), reply_markup=markup)

# Команда /calculate
@bot.message_handler(commands=["calculate"])
def calculate(m):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True, one_time_keyboard=True)
    item1 = types.KeyboardButton("Постоянный шаг")
    item2 = types.KeyboardButton("Переменный шаг")
    item3 = types.KeyboardButton("Кратный интеграл")
    markup.add(item1, item2,item3)
    bot.send_message(m.chat.id, text="{0.first_name}, какой способ тебя интересует? ".format(m.from_user), reply_markup=markup)
```

```python
#Выбор метода вычислений
@bot.message_handler(content_types=['text'])
def method(m):
    if(m.text == "Постоянный шаг"):
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True, one_time_keyboard=True)
        item1 = types.KeyboardButton("Метод левых частей")
        item2 = types.KeyboardButton("Метод правых частей")
        item3 = types.KeyboardButton("Метод Симпсона")
        item4 = types.KeyboardButton("Метод трапеций")
        item5 = types.KeyboardButton("Назад")
        markup.add(item1, item2,item3,item4,item5)
        bot.send_message(m.chat.id, text="Какой метод тебя интересует?".format(m.from_user), reply_markup=markup)
    elif(m.text == "Переменный шаг"):
        markup = types.ReplyKeyboardMarkup(row_width=1, one_time_keyboard=True)
        item1 = types.KeyboardButton("Двойной пересчёт алгоритм 1")
        item2 = types.KeyboardButton("Двойной пересчёт алгоритм 2")
        item3 = types.KeyboardButton("Назад")
        markup.add(item1, item2, item3)
        bot.send_message(m.chat.id, text="Выберите алгоритм:".format(m.from_user), reply_markup=markup)

    elif(m.text == "Кратный интеграл"):
        bot.reply_to(m, 'Введите количество разбиений по x')
        bot.register_next_step_handler(m, get_nx)

    elif(m.text == "Метод левых частей"):
        bot.reply_to(m, 'Введи количество разбиений')
        bot.register_next_step_handler(m, get_n_for_left)
    elif(m.text == "Метод правых частей"):
        bot.reply_to(m, 'Введи количество разбиений')
        bot.register_next_step_handler(m, get_n_for_right)
    elif(m.text == "Метод Симпсона"):
        bot.reply_to(m, 'Введи количество разбиений')
        bot.register_next_step_handler(m, get_n_for_parabola)
    elif(m.text == "Метод трапеций"):
        bot.reply_to(m, 'Введи количество разбиений')
        bot.register_next_step_handler(m, get_n_for_trapecia)


    elif(m.text == "Двойной пересчёт алгоритм 1"):
        bot.reply_to(m, 'Введи точность')
        bot.register_next_step_handler(m, get_e_for_recalc1)
    elif(m.text == "Двойной пересчёт алгоритм 2"):
        bot.reply_to(m, 'Введи точность')
        bot.register_next_step_handler(m, get_e_for_recalc2)
```

```
    elif(m.text == "Назад"):
        return calculate(m)
```

```
#Вычисления
def task(x):
    return x*x+5*x+3
```

С постоянным шагом:

1) Метод левых прямоугольников:

```
#Для Метода левых прямоугольников
def get_n_for_left(m,user_result = None):
    try:
        global user_n
        if user_result == None:
            user_n = int(m.text)
        else:
            user_n = str(user_result)
        bot.send_message(m.chat.id, text='Введи нижнюю границу')
        bot.register_next_step_handler(m, get_a_for_left)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Попробуйте заново. ')

def get_a_for_left(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введи верхнюю границу')
        bot.register_next_step_handler(m, get_b_for_left)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так...  Вы точно ввели число? Проверьте корректност

def get_b_for_left(m):
    try:
        global user_b
        user_b = float(m.text)
        resultPrint_for_left(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность

def left():
    global user_a, user_b, user_n, user_result
    h = (user_b - user_a) / user_n
    x = user_a
    user_result = 0
    while x <= (user_b - h):
        user_result += task(x)
        x += h
    return h * user_result

def resultPrint_for_left(m):
    global user_a, user_b, user_n, user_result
    res = left()
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

2) Метод правых прямоугольников:

```python
#Для Метода правых прямоугольников
def get_n_for_right(m,user_result = None):
    try:
        global user_n

        if user_result == None:
            user_n = int(m.text)
        else:
            user_n = str(user_result)
        bot.send_message(m.chat.id, text='Введи нижнюю границу')
        bot.register_next_step_handler(m, get_a_for_right)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Попробуйте заново.')

def get_a_for_right(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введи верхнюю границу')
        bot.register_next_step_handler(m, get_b_for_right)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность

def get_b_for_right(m):
    try:
        global user_b
        user_b = float(m.text)
        resultPrint_for_right(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность

def right():
    global user_a, user_b, user_n, user_result
    h = (user_b - user_a) / user_n
    x = user_a + h
    result = 0
    while x <= user_b:
        result += task(x)
        x += h
    return h * result

def resultPrint_for_right(m):
    global user_a, user_b, user_n, user_result
    res = right()
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

3) Метод трапеции:

```python
#Для Метода трапеции
def get_n_for_trapecia(m,user_result = None):
    try:
        global user_n

        if user_result == None:
            user_n = int(m.text)
        else:
            user_n = str(user_result)
        bot.send_message(m.chat.id, text='Введи нижнюю границу')
        bot.register_next_step_handler(m, get_a_for_trapecia)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Попробуйте заново.')

def get_a_for_trapecia(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введи верхнюю границу')
        bot.register_next_step_handler(m, get_b_for_trapecia)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность н

def get_b_for_trapecia(m):
    try:
        global user_b
        user_b = float(m.text)
        resultPrint_for_trapecia(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность н

def trapecia():
    global user_a, user_b, user_n, user_result
    h = (user_b - user_a) / user_n
    x = user_a + h
    result = 0
    while x <= (user_b - h):
        result += task(x)
        x += h
    return ((task(user_a) + task(user_b)) / 2 + result) * h

def resultPrint_for_trapecia(m):
    global user_a, user_b, user_n, user_result
    res = trapecia()
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

4) Метод парабол:

```python
#Для Метода парабол
def get_n_for_parabola(m,user_result = None):
    try:
        global user_n

        if user_result == None:
            user_n = int(m.text)
        else:
            user_n = str(user_result)
        bot.send_message(m.chat.id, text='Введи нижнюю границу')
        bot.register_next_step_handler(m, get_a_for_parabola)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так...  Вы точно ввели число? Попробуйте заново.')

def get_a_for_parabola(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введи верхнюю границу')
        bot.register_next_step_handler(m, get_b_for_parabola)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность н

def get_b_for_parabola(m):
    try:
        global user_b
        user_b = float(m.text)
        resultPrint_for_parabola(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность н
```

```python
def parabola():
    global user_a, user_b, user_n, user_result
    h = (user_b - user_a) / user_n
    S1 = 0
    x = user_a + h
    while x <= (user_b - h):
        S1 += task(x)
        x += 2 * h
    S2 = 0
    x = user_a + (2 * h)
    while x <= (user_b - (2 * h)):
        S2 += task(x)
        x += 2 * h
    return (task(user_a) + task(user_b) + (4 * S1) + (2 * S2)) * (h / 3)

def resultPrint_for_parabola(m):
    global user_a, user_b, user_n, user_result
    res = parabola()
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

С переменным шагом:

3- Алгоритм 1

```python
#Двойной перерасчет 1
def get_e_for_recalc1(m,user_result = None):
    try:
        global user_e
        if user_result == None:
            user_e = float(m.text)
        else:
            user_e = str(user_result)
        bot.send_message(m.chat.id, text='Введи нижнюю границу')
        bot.register_next_step_handler(m, get_a_for_recalc1)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте ко

def get_a_for_recalc1(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введи верхнюю границу')
        bot.register_next_step_handler(m, get_b_for_recalc1)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте ко

def get_b_for_recalc1(m):
    try:
        global user_b
        user_b = float(m.text)
        resultPrint_for_recalc1(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте ко
```

```python
def doubleRecalc(user_e,user_a, user_b ):
    n = 2
    f1 = 0
    f2 = left_for_recalc(n, user_a, user_b)
    while abs(f1-f2) > user_e:
        f1 = f2
        n = n * 2
        f2 = left_for_recalc(n, user_a, user_b)
    return f2

def left_for_recalc(n, user_a, user_b):
    h = (user_b - user_a) / n
    x = user_a
    user_result = 0
    while x <= (user_b - h):
        user_result += task(x)
        x += h
    return h * user_result

def resultPrint_for_recalc1(m):
    res = doubleRecalc(user_e,user_a, user_b)
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

4- Алгоритм 2

```python
#Двойной перерасчет 2
def get_e_for_recalc2(m,user_result = None):
    try:
        global user_e
        if user_result == None:
            user_e = float(m.text)
        else:
            user_e = str(user_result)
        bot.send_message(m.chat.id, text='Введи нижнюю границу')
        bot.register_next_step_handler(m, get_a_for_recalc2)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Скорее всего, вы ввели не число. Попробуйте заново ')

def get_a_for_recalc2(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введи верхнюю границу')
        bot.register_next_step_handler(m, get_b_for_recalc2)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность написания веще

def get_b_for_recalc2(m):
    try:
        global user_b
        user_b = float(m.text)
        resultPrint_for_recalc2(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность написания веще
```

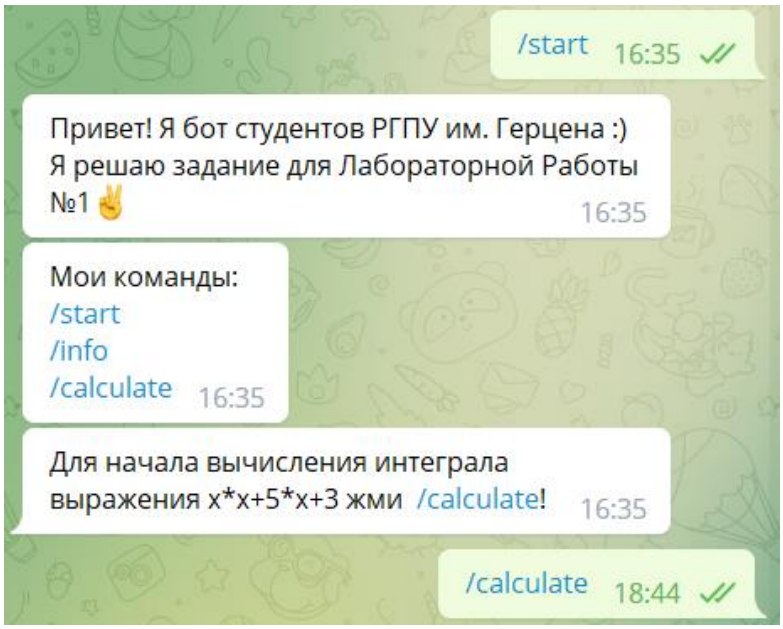
```python
def doubleRecalcBetter(user_e,user_a, user_b):
    n = 2
    otst = (user_b - user_a)/2
    f1 = left_for_recalc(n, user_a, user_b)
    otst = otst / 2
    f2 = left_for_recalc(n, user_a + otst, user_b)
    while abs(f1-f2) > user_e:
        f1 = f2
        n = n * 2
        otst = otst / 2
        f2 = left_for_recalc(n, user_a+otst, user_b)
    return f2

def resultPrint_for_recalc2(m):
    res = doubleRecalcBetter(user_e,user_a, user_b)
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

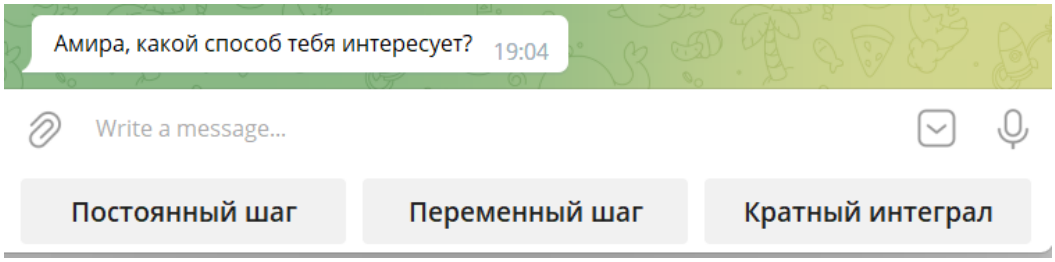
Кратный интеграл:

```python
#Для кратного интеграла
def get_nx(m,user_result = None):
    try:
        global user_nx

        if user_result == None:
            user_nx = float(m.text)
        else:
            user_nx = str(user_result)
        bot.send_message(m.chat.id, text='Введите количество разбиений по y')
        bot.register_next_step_handler(m, get_ny)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корре

def get_ny(m):
    try:
        global user_ny
        user_ny = float(m.text)
        bot.send_message(m.chat.id, text='Введите внешнюю нижнюю границу')
        bot.register_next_step_handler(m, get_a)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корре

def get_a(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введите внешнюю верхнюю границу')
        bot.register_next_step_handler(m, get_b)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корре

def get_b(m):
    try:
        global user_b
        user_b = float(m.text)
        bot.send_message(m.chat.id, text='Введите внутреннюю нижнюю границу')
        bot.register_next_step_handler(m, get_c)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корре
```

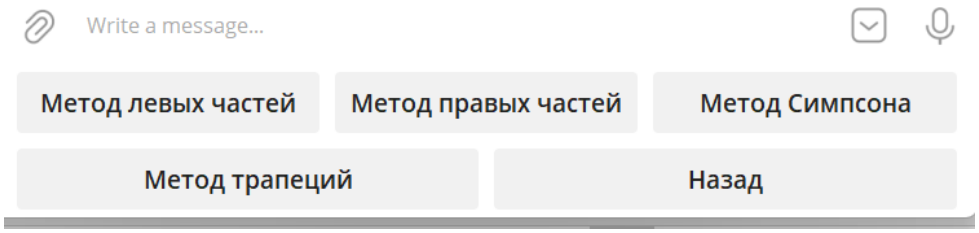
```python
def get_c(m):
    try:
        global user_c
        user_c = float(m.text)
        bot.send_message(m.chat.id, text='Введите внутреннюю верхнюю границу')
        bot.register_next_step_handler(m, get_d)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректно

def get_d(m):
    try:
        global user_d
        user_d = float(m.text)
        resultPrint_for_KratInt(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректно

def KratInt(x,y):
    return sin(x+y)

def calc_kratniy():
    global user_nx, user_ny, user_a, user_b, user_c, user_d
    hx = (user_b - user_a) / user_nx
    hy = (user_d - user_c) / user_ny
    S = 0
    x = user_a
    while x <= (user_b - hx):
        y = user_c
        while y <= (user_d - hy):
            S += KratInt(x, y)
            y += hy
        x += hx
    return S*hx*hy

def resultPrint_for_KratInt(m):
    global user_nx, user_ny, user_a, user_b, user_c, user_d, user_result
    res = calc_kratniy()
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

Приложение 2

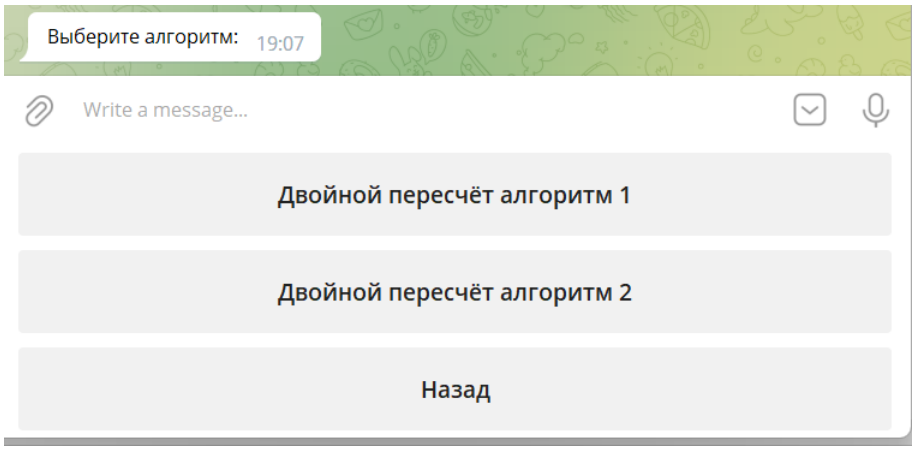




При выборе “постоянный шаг” нам предлагаются следующие методы:



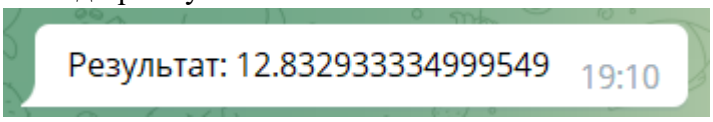
При выборе “переменный шаг” нам предлагаются 2 метода:



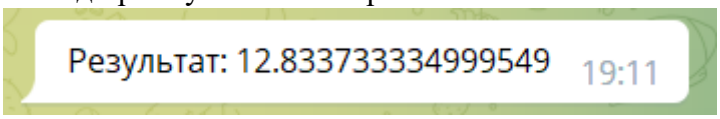
Результаты вычисления интеграла:

С постоянным шагом (с количеством разбиений 10000):

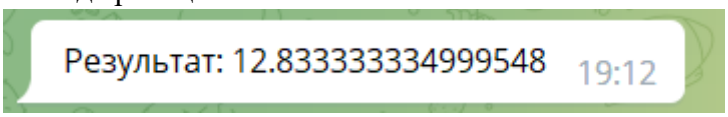
1- Метод прямоугольников левых частей



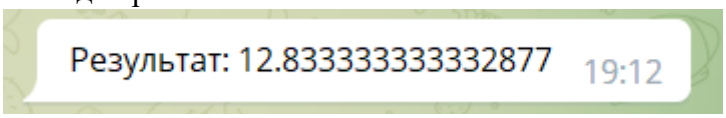
2- Метод прямоугольников правых частей



3- Метод трапеций

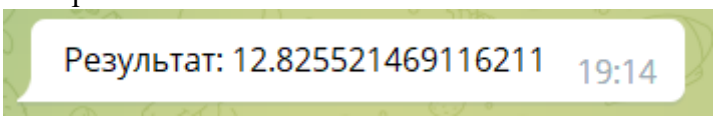


4- Метод парабол

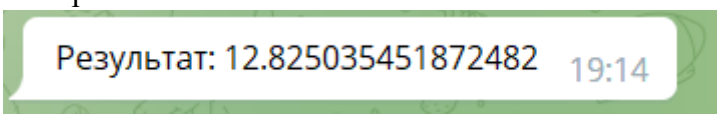


С переменным шагом (с точностью 0.01):

1- Алгоритм 1



2- Алгоритм 2



Крючкова Анастасия ИВТ 2.1

Лабораторная работа №1

Численное интегрирование

Цель лабораторной работы: вычислить определенный интеграл, используя различные численные методы и алгоритмы их реализации.

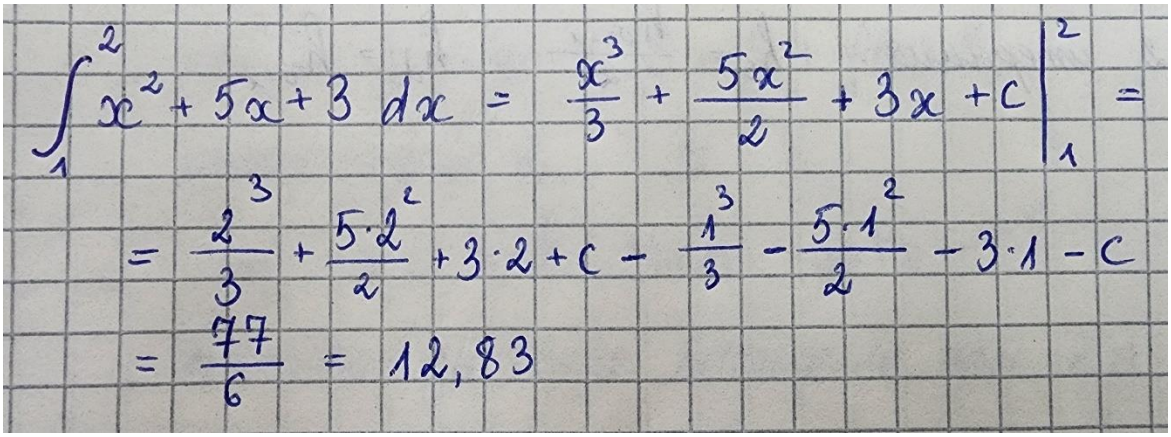
Инструменты: PyCharm, Telegram.

В рамках данной лабораторной работы, был использовал язык программирования Python 3.10

Мы использовали подынтегральную функцию: $x^2 + 5x + 3$.

В качестве примера определенного интеграла возьмем: $\int_1^2 x^2 + 5x + 3\ dx$

Вычислим данный интеграл ручным способом:



При вычислении, получили следующие результаты определенного интеграла с постоянным шагом:

Кол-во разбиений	Метод прямоугольников левых частей	Метод прямоугольников правых частей	Метод трапеций	Метод парабол
100	12.6242	12.7034	12.6642	12.4957
1000	12.8293	12.8373	12.8333	12.7993
10000	12.8329	12.8337	12.8333	12.8333

Консоль и Телеграм-бот выдают одинаковые результаты.

Код и работа Телеграм-бота приведена в приложении 1 и 2, соответственно.(страницы 26 и 41)

Вывод: в ходе сравнения машинных вычислений и ручного вычисления, мы пришли к выводу, что результаты практически идентичны и чем больше количество разбиений, тем точнее выводится результат.

Определенный интеграл с переменным шагом (2 алгоритма)

Точность	Двойной пересчет алгоритм 1	Двойной пересчет алгоритм 2
0.1	12.8255	12.8250
0.001	12.8323	12.8328
0.0001	12.8332	12.8332

Вывод: два данных алгоритма, прекрасно посчитали интеграл и сошли с результатом ручного вычисления, но в данном случае чем меньше точность, тем точнее результат.

Кратный интеграл

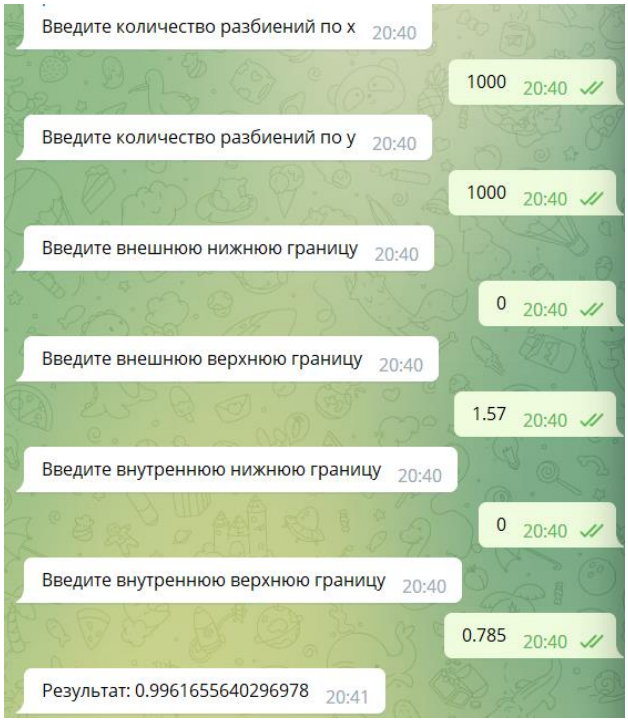
Возьмем кратный интеграл:

$$\int_0^{\pi/2} dx \int_0^{\pi/4} Sin(x+y)dy; \qquad n = 4$$

И вычислим его:

$$\int_0^{\pi/2} \int_0^{\pi/4} Sin(x+y)dxdy \approx \left(\frac{\pi}{24}\right)^2 + [\dots + \dots] = 1{,}00028$$

В нашей программе будет считать, что число пи это 3.14, соответственно пи деленное на 4 это 0.785, а пи деленное на 2 это 1.57.



Вывод: Результаты ручного вычисления и машинного приблизительно равны, погрешность возникла из-за примерного взятия числа пи, а также при увеличении кол-ва разбиений результат будет еще точнее.

Вывод по всей лабораторной работе: Мы научились реализовывать интегрирование и вычислили кратный интеграл, создали программу для данных вычислений и на ее основе создали Телеграм-бота.

Ссылка на Телеграм-бота: ссылка

Приложение 1

Код :

```
from math import sqrt
from math import sin

def task(x: float):
    return x*x+5*x+3

def KratInt(x:float, y:float):
    return sin(x+y)
```

С постоянным шагом:

1. Метод левых прямоугольников:

```
def left(n: int, A:float, B:float):
    h = (B - A) / n
    x = A
    result = 0
    while x <= (B - h):
        result += task(x)
        x += h
    return h * result
```

2. Метод правых прямоугольников:

```
def right(n: int, A:float, B:float):
    h = (B - A) / n
    x = A + h
    result = 0
    while x <= B:
        result += task(x)
        x += h
    return h * result
```

3. Матод трапеций:

```
def trapecia(n: int, A:float, B:float):
    h = (B - A) / n
    x = A + h
    result = 0
    while x <= (B - h):
        result += task(x)
        x += h
    return ((task(A) + task(B)) / 2 + result) * h
```

4. Метод парабол:

```
def parabola(n: int, A:float, B:float):
    h = (B - A) / n
    S1 = 0
    x = A + h
    while x <= (B - h):
        S1 += task(x)
        x += 2 * h
    S2 = 0
    x = A + (2 * h)
    while x <= (B - (2 * h)):
        S2 += task(x)
        x += 2 * h
    return (task(A) + task(B) + (4 * S1) + (2 * S2)) * (h / 3)
```

С переменным шагом:

1. Алгоритм 1:

```
def doubleRecalc(e:float, A:float, B:float):
    n = 2
    f1 = 0
    f2 = left(n, A, B)
    while abs(f1-f2) > e:
        f1 = f2
        n = n * 2
        f2 = left(n, A, B)
    return f2
```

2. Алгоритм 2:

```
def doubleRecalcBetter(e:float, A:float, B:float):
    n = 2
    otst = (B - A)/2
    f1 = left(n, A, B)
    otst = otst / 2
    f2 = left(n, A + otst, B)
    while abs(f1-f2) > e:
        f1 = f2
        n = n * 2
        otst = otst / 2
        f2 = left(n, A+otst, B)
    return f2
```

Кратный интеграл:

```
def kratniy(nx:int, ny:int, A:float, B:float, C:float, D:float):
    hx = (B - A) / nx
    hy = (D - C) / ny
    S = 0
    x = A
    while x <= (B - hx):
        y = C
        while y <= (D - hy):
            S += KratInt(x, y)
            y += hy
        x += hx
    return S*hx*hy
```

Меню:

```
special1 = True
while special1 == True:
    special2 = True
    print("Welcome to MainMenu")
    print("Для выбора задачи нажмите соответствующую кнопку:")
    print('[1]Численное интегрирование\n'
          'more will comming soon')
    mean1 = int(input())

    if mean1 == 1:
        while special2 == True:
            special3 = True
            print("Выберите:\n"
                  "[1] С постоянным шагом\n"
                  "[2] С переменным шагом\n"
                  "[3] Кратный интегралл")
            print("Для выхода в меню введите '4'")
            mean2 = int(input())

            if mean2 == 1:
                print("Выбран постоянный шаг\n")
                while special3 == True:
                    print("Выберите метод:\n"
                          "[1] Метод левых прямоугольников\n"
                          "[2] Метод правых прямоугольников\n"
                          "[3] Метод Симпсона(парабол)\n"
                          "[4] Метод трапеций")
                    print("Для возврата нажмите '5'")
                    mean3 = int(input())

                    if mean3 == 1:
                        print("Выбран 'Метод левых прямоугольников'")
                        print('Введите количество разбиений')
                        n = int(input())
                        print('Введите нижнюю границу')
                        A = float(input())
                        print('Введите верхнюю границу')
                        B = float(input())
                        res = left(n, A, B)
                        print("Результат: ", res, '\n')
```

```
            elif mean3 == 2:
                print("Выбран 'Метод правых прямоугольников'")
                print('Введите количество разбиений')
                n = int(input())
                print('Введите нижнюю границу')
                A = float(input())
                print('Введите верхнюю границу')
                B = float(input())
                res = right(n, A, B)
                print("Результат: ", res, '\n')

            elif mean3 == 3:
                print("Выбран 'Метод Симпсона(парабол)'")
                print('Введите количество разбиений')
                n = int(input())
                print('Введите нижнюю границу')
                A = float(input())
                print('Введите верхнюю границу')
                B = float(input())
                res = parabola(n, A, B)
                print("Результат: ", res, '\n')

            elif mean3 == 4:
                print("Выбран 'Метод трапеций'")
                print('Введите количество разбиений')
                n = int(input())
                print('Введите нижнюю границу')
                A = float(input())
                print('Введите верхнюю границу')
                B = float(input())
                res = trapecia(n, A, B)
                print("Результат: ", res, '\n')

            elif mean3 == 5:
                special3 = False
                print()
```

```
        elif mean2 == 2:
            print("Выбран переменный шаг\n")
            while special3 == True:
                print("Выберите алгоритм:\n"
                      "[1] Двойной пересчёт алгоритм 1\n"
                      "[2] Двойной пересчёт алгоритм 2")
                print("Для возврата нажмите '3'")
                mean4 = int(input())

                if mean4 == 1:
                    print("Двойной пересчёт'\n"
                          "Алгоритм 1\n"
                          "Метод левых прямоугольников \n")

                    print('Введите точность')
                    e = float(input())
                    print('Введите нижнюю границу')
                    A = float(input())
                    print('Введите верхнюю границу')
                    B = float(input())
                    res = doubleRecalc(e, A, B)
                    print("Результат: ", res, '\n')

                elif mean4 == 2:
                    print("Двойной пересчёт\n"
                          "Алгоритм 2\n"
                          "Метод левых прямоугольников\n")
                    print('Введите точность')
                    e = float(input())
                    print('Введите нижнюю границу')
                    A = float(input())
                    print('Введите верхнюю границу')
                    B = float(input())
                    res = doubleRecalcBetter(e, A, B)
                    print("Результат: ", res, '\n')

                elif mean4 == 3:
                    special3 = False
                    print()
```

```
        elif mean2 == 3:
            print("Выбран кратный интеграл\n")
            print("Введите количество разбиений по x")
            nx = int(input())
            print("Введите количество разбиений по y")
            ny = int(input())
            print("Введите внешнюю нижнюю границу")
            A = float(input())
            print("Введите внешнюю верхнюю границу")
            B = float(input())
            print("Введите внутреннюю нижнюю границу")
            C = float(input())
            print("Введите внутреннюю верхнюю границу")
            D = float(input())
            res = kratniy(nx, ny, A, B, C, D)
            print("Результат",res,'\n')

        elif mean2 == 4:
            special2 = False
```

Код для Телеграм-бота:

Код меню:

```python
import telebot
from telebot import types
from math import sin

bot = telebot.TeleBot('TOKEN')

# Команда /start
@bot.message_handler(commands=["start"])
def start(m):
    bot.send_message(m.chat.id, 'Привет! Я бот студентов РГПУ им. Герцена :) Я решаю задание для Лабораторной Работы №1✌ ')
    bot.send_message(m.chat.id, 'Мои команды:\n/start\n/info\n/calculate')
    bot.send_message(m.chat.id, 'Для начала вычисления интеграла выражения x*x+5*x+3 жми  /calculate!')

@bot.message_handler(commands=["info"])
def info(m):
    bot.send_message(m.chat.id, 'Я бот студентов РГПУ им. Герцена :) Моих создателей зовут Максим Стецук, Крючкова Анастасия и Зухир Амира. Они из группы ИВТ 2-1')
    markup = types.InlineKeyboardMarkup()
    item1 = types.InlineKeyboardButton("Максим", url='https://vk.com/makstulenchik')
    item2 = types.InlineKeyboardButton("Анастасия", url='https://vk.com/nestessia')
    item3 = types.InlineKeyboardButton("Амира", url='https://vk.com/amirazhr')
    item4 = types.InlineKeyboardButton("Ангелина", url='https://vk.com/mintange')
    markup.add(item1)
    markup.add(item2)
    markup.add(item3)
    markup.add(item4)
    bot.send_message(m.chat.id, "{0.first_name}, для связи с нами переходи по ссылкам :)".format(m.from_user), reply_markup=markup)

# Команда /calculate
@bot.message_handler(commands=["calculate"])
def calculate(m):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True, one_time_keyboard=True)
    item1 = types.KeyboardButton("Постоянный шаг")
    item2 = types.KeyboardButton("Переменный шаг")
    item3 = types.KeyboardButton("Кратный интеграл")
    markup.add(item1, item2,item3)
    bot.send_message(m.chat.id, text="{0.first_name}, какой способ тебя интересует? ".format(m.from_user), reply_markup=markup)
```

```python
#Выбор метода вычислений
@bot.message_handler(content_types=['text'])
def method(m):
    if(m.text == "Постоянный шаг"):
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True, one_time_keyboard=True)
        item1 = types.KeyboardButton("Метод левых частей")
        item2 = types.KeyboardButton("Метод правых частей")
        item3 = types.KeyboardButton("Метод Симпсона")
        item4 = types.KeyboardButton("Метод трапеций")
        item5 = types.KeyboardButton("Назад")
        markup.add(item1, item2,item3,item4,item5)
        bot.send_message(m.chat.id, text="Какой метод тебя интересует?".format(m.from_user), reply_markup=markup)
    elif(m.text == "Переменный шаг"):
        markup = types.ReplyKeyboardMarkup(row_width=1, one_time_keyboard=True)
        item1 = types.KeyboardButton("Двойной пересчёт алгоритм 1")
        item2 = types.KeyboardButton("Двойной пересчёт алгоритм 2")
        item3 = types.KeyboardButton("Назад")
        markup.add(item1, item2, item3)
        bot.send_message(m.chat.id, text="Выберите алгоритм:".format(m.from_user), reply_markup=markup)

    elif(m.text == "Кратный интеграл"):
        bot.reply_to(m, 'Введите количество разбиений по x')
        bot.register_next_step_handler(m, get_nx)

    elif(m.text == "Метод левых частей"):
        bot.reply_to(m, 'Введи количество разбиений')
        bot.register_next_step_handler(m, get_n_for_left)
    elif(m.text == "Метод правых частей"):
        bot.reply_to(m, 'Введи количество разбиений')
        bot.register_next_step_handler(m, get_n_for_right)
    elif(m.text == "Метод Симпсона"):
        bot.reply_to(m, 'Введи количество разбиений')
        bot.register_next_step_handler(m, get_n_for_parabola)
    elif(m.text == "Метод трапеций"):
        bot.reply_to(m, 'Введи количество разбиений')
        bot.register_next_step_handler(m, get_n_for_trapecia)


    elif(m.text == "Двойной пересчёт алгоритм 1"):
        bot.reply_to(m, 'Введи точность')
        bot.register_next_step_handler(m, get_e_for_recalc1)
    elif(m.text == "Двойной пересчёт алгоритм 2"):
        bot.reply_to(m, 'Введи точность')
        bot.register_next_step_handler(m, get_e_for_recalc2)
```

```
    elif(m.text == "Назад"):
        return calculate(m)
```

```
#Вычисления
def task(x):
    return x*x+5*x+3
```

С постоянным шагом:

1) Метод левых прямоугольников:

```
#Для Метода левых прямоугольников
def get_n_for_left(m,user_result = None):
    try:
        global user_n
        if user_result == None:
            user_n = int(m.text)
        else:
            user_n = str(user_result)
        bot.send_message(m.chat.id, text='Введи нижнюю границу')
        bot.register_next_step_handler(m, get_a_for_left)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Попробуйте заново. ')

def get_a_for_left(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введи верхнюю границу')
        bot.register_next_step_handler(m, get_b_for_left)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так...  Вы точно ввели число? Проверьте корректност

def get_b_for_left(m):
    try:
        global user_b
        user_b = float(m.text)
        resultPrint_for_left(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность

def left():
    global user_a, user_b, user_n, user_result
    h = (user_b - user_a) / user_n
    x = user_a
    user_result = 0
    while x <= (user_b - h):
        user_result += task(x)
        x += h
    return h * user_result

def resultPrint_for_left(m):
    global user_a, user_b, user_n, user_result
    res = left()
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

2) Метод правых прямоугольников:

```python
#Для Метода правых прямоугольников
def get_n_for_right(m,user_result = None):
    try:
        global user_n

        if user_result == None:
            user_n = int(m.text)
        else:
            user_n = str(user_result)
        bot.send_message(m.chat.id, text='Введи нижнюю границу')
        bot.register_next_step_handler(m, get_a_for_right)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Попробуйте заново.')

def get_a_for_right(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введи верхнюю границу')
        bot.register_next_step_handler(m, get_b_for_right)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность

def get_b_for_right(m):
    try:
        global user_b
        user_b = float(m.text)
        resultPrint_for_right(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность

def right():
    global user_a, user_b, user_n, user_result
    h = (user_b - user_a) / user_n
    x = user_a + h
    result = 0
    while x <= user_b:
        result += task(x)
        x += h
    return h * result

def resultPrint_for_right(m):
    global user_a, user_b, user_n, user_result
    res = right()
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

3) Метод трапеции:

```python
#Для Метода трапеции
def get_n_for_trapecia(m,user_result = None):
    try:
        global user_n

        if user_result == None:
            user_n = int(m.text)
        else:
            user_n = str(user_result)
        bot.send_message(m.chat.id, text='Введи нижнюю границу')
        bot.register_next_step_handler(m, get_a_for_trapecia)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Попробуйте заново.')

def get_a_for_trapecia(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введи верхнюю границу')
        bot.register_next_step_handler(m, get_b_for_trapecia)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность н

def get_b_for_trapecia(m):
    try:
        global user_b
        user_b = float(m.text)
        resultPrint_for_trapecia(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность н

def trapecia():
    global user_a, user_b, user_n, user_result
    h = (user_b - user_a) / user_n
    x = user_a + h
    result = 0
    while x <= (user_b - h):
        result += task(x)
        x += h
    return ((task(user_a) + task(user_b)) / 2 + result) * h

def resultPrint_for_trapecia(m):
    global user_a, user_b, user_n, user_result
    res = trapecia()
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

4) Метод парабол:

```python
#Для Метода парабол
def get_n_for_parabola(m,user_result = None):
    try:
        global user_n

        if user_result == None:
            user_n = int(m.text)
        else:
            user_n = str(user_result)
        bot.send_message(m.chat.id, text='Введи нижнюю границу')
        bot.register_next_step_handler(m, get_a_for_parabola)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так...  Вы точно ввели число? Попробуйте заново.')

def get_a_for_parabola(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введи верхнюю границу')
        bot.register_next_step_handler(m, get_b_for_parabola)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность н

def get_b_for_parabola(m):
    try:
        global user_b
        user_b = float(m.text)
        resultPrint_for_parabola(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность н
```

```python
def parabola():
    global user_a, user_b, user_n, user_result
    h = (user_b - user_a) / user_n
    S1 = 0
    x = user_a + h
    while x <= (user_b - h):
        S1 += task(x)
        x += 2 * h
    S2 = 0
    x = user_a + (2 * h)
    while x <= (user_b - (2 * h)):
        S2 += task(x)
        x += 2 * h
    return (task(user_a) + task(user_b) + (4 * S1) + (2 * S2)) * (h / 3)

def resultPrint_for_parabola(m):
    global user_a, user_b, user_n, user_result
    res = parabola()
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

С переменным шагом:

1- Алгоритм 1

```python
#Двойной перерасчет 1
def get_e_for_recalc1(m,user_result = None):
    try:
        global user_e
        if user_result == None:
            user_e = float(m.text)
        else:
            user_e = str(user_result)
        bot.send_message(m.chat.id, text='Введи нижнюю границу')
        bot.register_next_step_handler(m, get_a_for_recalc1)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте ко

def get_a_for_recalc1(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введи верхнюю границу')
        bot.register_next_step_handler(m, get_b_for_recalc1)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте ко

def get_b_for_recalc1(m):
    try:
        global user_b
        user_b = float(m.text)
        resultPrint_for_recalc1(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте ко
```

```python
def doubleRecalc(user_e,user_a, user_b ):
    n = 2
    f1 = 0
    f2 = left_for_recalc(n, user_a, user_b)
    while abs(f1-f2) > user_e:
        f1 = f2
        n = n * 2
        f2 = left_for_recalc(n, user_a, user_b)
    return f2

def left_for_recalc(n, user_a, user_b):
    h = (user_b - user_a) / n
    x = user_a
    user_result = 0
    while x <= (user_b - h):
        user_result += task(x)
        x += h
    return h * user_result

def resultPrint_for_recalc1(m):
    res = doubleRecalc(user_e,user_a, user_b)
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

2- Алгоритм 2

```python
#Двойной перерасчет 2
def get_e_for_recalc2(m,user_result = None):
    try:
        global user_e
        if user_result == None:
            user_e = float(m.text)
        else:
            user_e = str(user_result)
        bot.send_message(m.chat.id, text='Введи нижнюю границу')
        bot.register_next_step_handler(m, get_a_for_recalc2)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Скорее всего, вы ввели не число. Попробуйте заново ')

def get_a_for_recalc2(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введи верхнюю границу')
        bot.register_next_step_handler(m, get_b_for_recalc2)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность написания веще

def get_b_for_recalc2(m):
    try:
        global user_b
        user_b = float(m.text)
        resultPrint_for_recalc2(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность написания веще
```

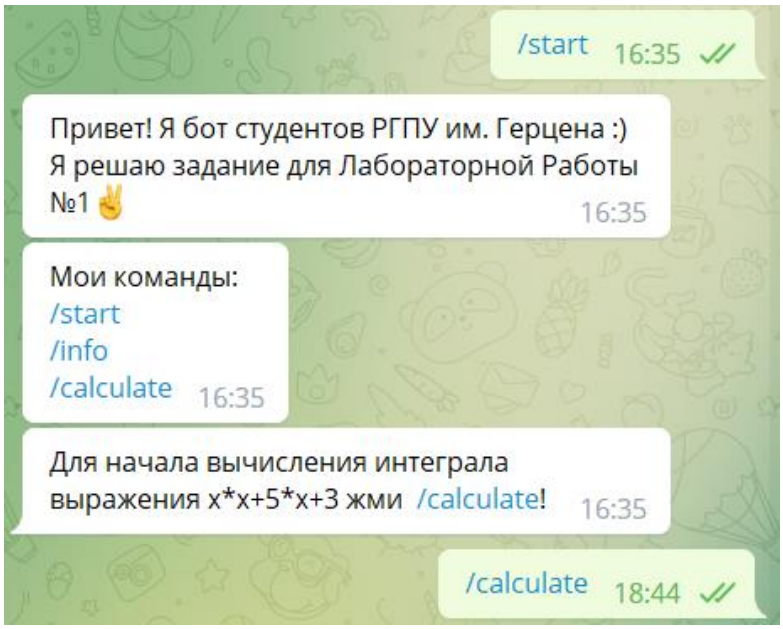
```python
def doubleRecalcBetter(user_e,user_a, user_b):
    n = 2
    otst = (user_b - user_a)/2
    f1 = left_for_recalc(n, user_a, user_b)
    otst = otst / 2
    f2 = left_for_recalc(n, user_a + otst, user_b)
    while abs(f1-f2) > user_e:
        f1 = f2
        n = n * 2
        otst = otst / 2
        f2 = left_for_recalc(n, user_a+otst, user_b)
    return f2

def resultPrint_for_recalc2(m):
    res = doubleRecalcBetter(user_e,user_a, user_b)
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

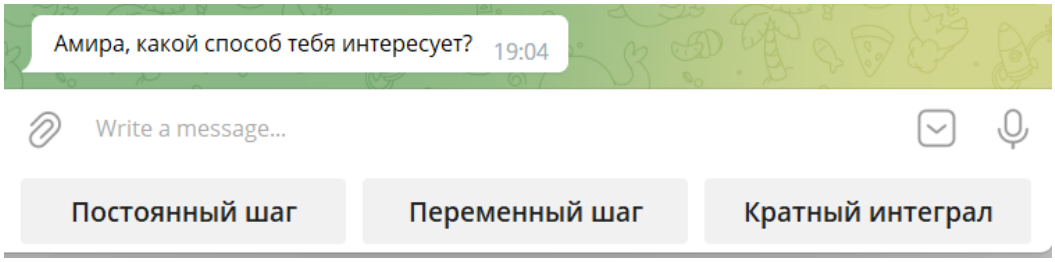
Кратный интеграл:

```python
#Для кратного интеграла
def get_nx(m,user_result = None):
    try:
        global user_nx

        if user_result == None:
            user_nx = float(m.text)
        else:
            user_nx = str(user_result)
        bot.send_message(m.chat.id, text='Введите количество разбиений по y')
        bot.register_next_step_handler(m, get_ny)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корре

def get_ny(m):
    try:
        global user_ny
        user_ny = float(m.text)
        bot.send_message(m.chat.id, text='Введите внешнюю нижнюю границу')
        bot.register_next_step_handler(m, get_a)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корре

def get_a(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введите внешнюю верхнюю границу')
        bot.register_next_step_handler(m, get_b)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корре

def get_b(m):
    try:
        global user_b
        user_b = float(m.text)
        bot.send_message(m.chat.id, text='Введите внутреннюю нижнюю границу')
        bot.register_next_step_handler(m, get_c)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корре
```

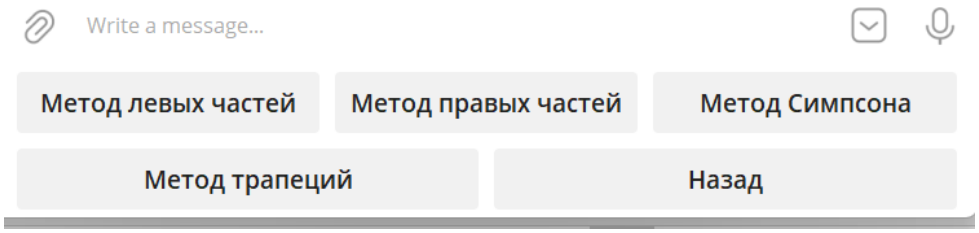
```python
def get_c(m):
    try:
        global user_c
        user_c = float(m.text)
        bot.send_message(m.chat.id, text='Введите внутреннюю верхнюю границу')
        bot.register_next_step_handler(m, get_d)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректно

def get_d(m):
    try:
        global user_d
        user_d = float(m.text)
        resultPrint_for_KratInt(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректно

def KratInt(x,y):
    return sin(x+y)

def calc_kratniy():
    global user_nx, user_ny, user_a, user_b, user_c, user_d
    hx = (user_b - user_a) / user_nx
    hy = (user_d - user_c) / user_ny
    S = 0
    x = user_a
    while x <= (user_b - hx):
        y = user_c
        while y <= (user_d - hy):
            S += KratInt(x, y)
            y += hy
        x += hx
    return S*hx*hy

def resultPrint_for_KratInt(m):
    global user_nx, user_ny, user_a, user_b, user_c, user_d, user_result
    res = calc_kratniy()
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

Приложение 2

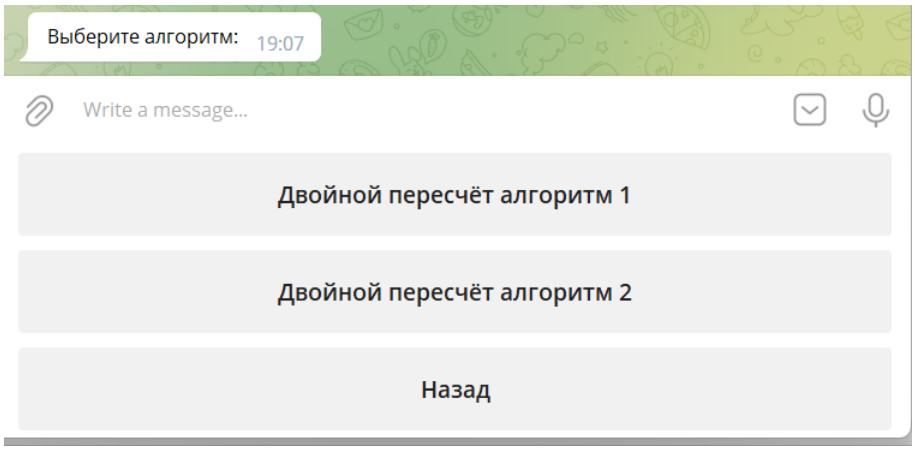




При выборе “постоянный шаг” нам предлагаются следующие методы:



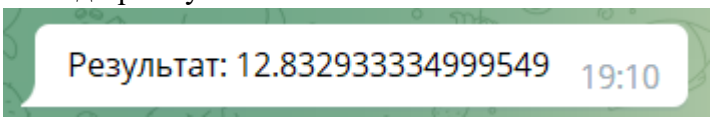
При выборе “переменный шаг” нам предлагаются 2 метода:



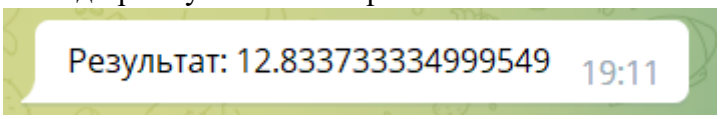
Результаты вычисления интеграла:

С постоянным шагом (с количеством разбиений 10000):

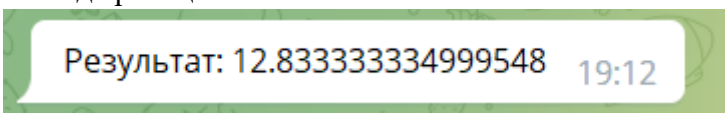
1- Метод прямоугольников левых частей



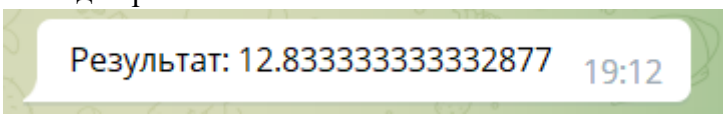
2- Метод прямоугольников правых частей



3- Метод трапеций

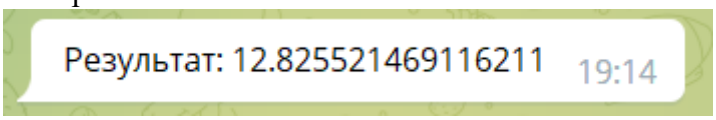


4- Метод парабол

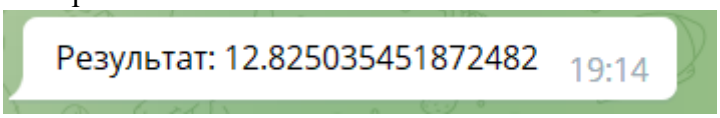


С переменным шагом (с точностью 0.01):

1- Алгоритм 1



2- Алгоритм 2



Стецук Максим ИВТ 2.1

Лабораторная работа №1

Численное интегрирование

Цель лабораторной работы: вычислить определенный интеграл, используя различные численные методы и алгоритмы их реализации.

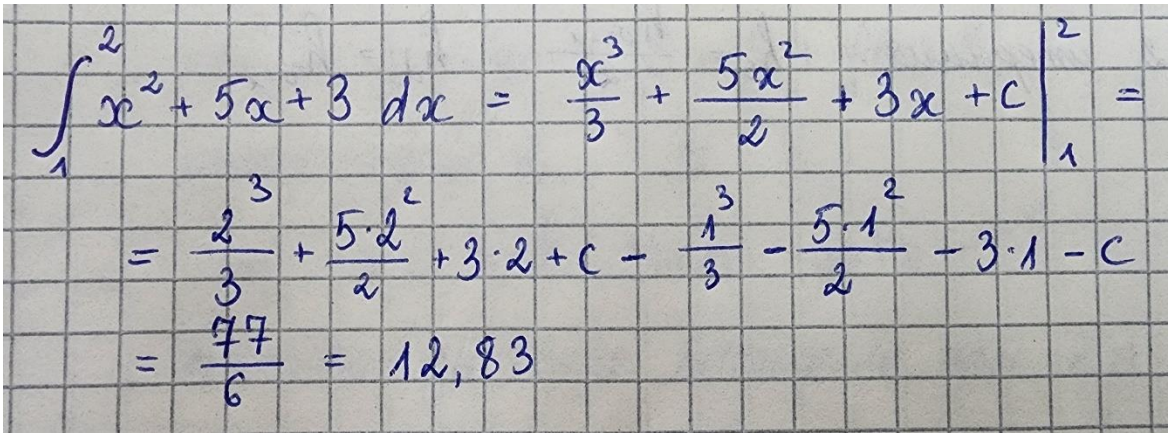
Инструменты: PyCharm, Telegram.

В рамках данной лабораторной работы, был использовал язык программирования Python 3.10

Мы использовали подынтегральную функцию: $x^2 + 5x + 3$.

В качестве примера определенного интеграла возьмем: $\int_1^2 x^2 + 5x + 3\ dx$

Вычислим данный интеграл ручным способом:



При вычислении, получили следующие результаты определенного интеграла с постоянным шагом:

Кол-во разбиений	Метод прямоугольников левых частей	Метод прямоугольников правых частей	Метод трапеций	Метод парабол
100	12.6242	12.7034	12.6642	12.4957
1000	12.8293	12.8373	12.8333	12.7993
10000	12.8329	12.8337	12.8333	12.8333

Консоль и Телеграм-бот выдают одинаковые результаты.

Код и работа Телеграм-бота приведена в приложении 1 и 2, соответственно.(страницы 46 и 61)

Вывод: в ходе сравнения машинных вычислений и ручного вычисления, мы пришли к выводу, что результаты практически идентичны и чем больше количество разбиений, тем точнее выводится результат.

Определенный интеграл с переменным шагом (2 алгоритма)

Точность	Двойной пересчет алгоритм 1	Двойной пересчет алгоритм 2
0.1	12.8255	12.8250
0.001	12.8323	12.8328
0.0001	12.8332	12.8332

Вывод: два данных алгоритма, прекрасно посчитали интеграл и сошли с результатом ручного вычисления, но в данном случае чем меньше точность, тем точнее результат.

Кратный интеграл

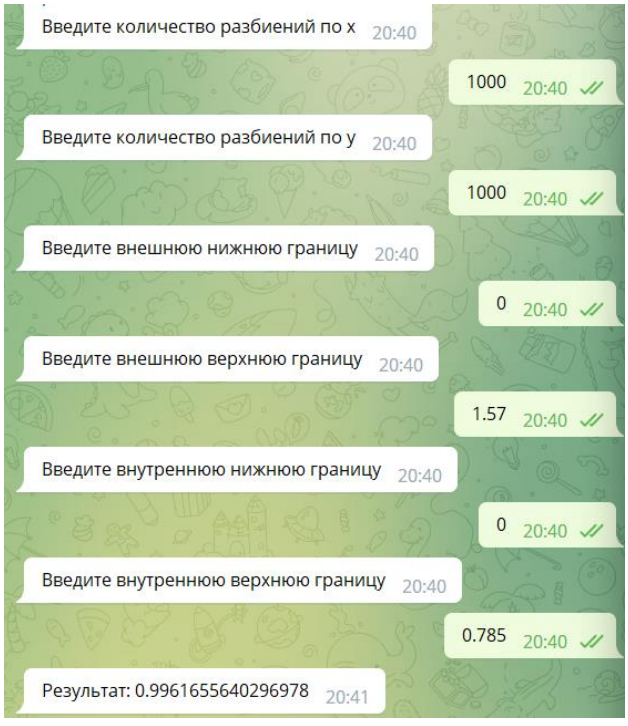
Возьмем кратный интеграл:

$$\int_0^{\pi/2} dx \int_0^{\pi/4} Sin(x+y)dy; \qquad n = 4$$

И вычислим его:

$$\int_0^{\pi/2} \int_0^{\pi/4} Sin(x+y)dxdy \approx \left(\frac{\pi}{24}\right)^2 + [\dots + \dots] = 1{,}00028$$

В нашей программе будет считать, что число пи это 3.14, соответственно пи деленное на 4 это 0.785, а пи деленное на 2 это 1.57.



Вывод: Результаты ручного вычисления и машинного приблизительно равны, погрешность возникла из-за примерного взятия числа пи, а также при увеличении кол-ва разбиений результат будет еще точнее.

Вывод по всей лабораторной работе: Мы научились реализовывать интегрирование и вычислили кратный интеграл, создали программу для данных вычислений и на ее основе создали Телеграм-бота.

Ссылка на Телеграм-бота: ссылка

Приложение 1

Код :

```
from math import sqrt
from math import sin

def task(x: float):
    return x*x+5*x+3

def KratInt(x:float, y:float):
    return sin(x+y)
```

С постоянным шагом:

1. Метод левых прямоугольников:

```
def left(n: int, A:float, B:float):
    h = (B - A) / n
    x = A
    result = 0
    while x <= (B - h):
        result += task(x)
        x += h
    return h * result
```

2. Метод правых прямоугольников:

```
def right(n: int, A:float, B:float):
    h = (B - A) / n
    x = A + h
    result = 0
    while x <= B:
        result += task(x)
        x += h
    return h * result
```

3. Матод трапеций:

```
def trapecia(n: int, A:float, B:float):
    h = (B - A) / n
    x = A + h
    result = 0
    while x <= (B - h):
        result += task(x)
        x += h
    return ((task(A) + task(B)) / 2 + result) * h
```

4. Метод парабол:

```
def parabola(n: int, A:float, B:float):
    h = (B - A) / n
    S1 = 0
    x = A + h
    while x <= (B - h):
        S1 += task(x)
        x += 2 * h
    S2 = 0
    x = A + (2 * h)
    while x <= (B - (2 * h)):
        S2 += task(x)
        x += 2 * h
    return (task(A) + task(B) + (4 * S1) + (2 * S2)) * (h / 3)
```

С переменным шагом:

1- Алгоритм 1:

```
def doubleRecalc(e:float, A:float, B:float):
    n = 2
    f1 = 0
    f2 = left(n, A, B)
    while abs(f1-f2) > e:
        f1 = f2
        n = n * 2
        f2 = left(n, A, B)
    return f2
```

2- Алгоритм 2:

```
def doubleRecalcBetter(e:float, A:float, B:float):
    n = 2
    otst = (B - A)/2
    f1 = left(n, A, B)
    otst = otst / 2
    f2 = left(n, A + otst, B)
    while abs(f1-f2) > e:
        f1 = f2
        n = n * 2
        otst = otst / 2
        f2 = left(n, A+otst, B)
    return f2
```

Кратный интеграл:

```
def kratniy(nx:int, ny:int, A:float, B:float, C:float, D:float):
    hx = (B - A) / nx
    hy = (D - C) / ny
    S = 0
    x = A
    while x <= (B - hx):
        y = C
        while y <= (D - hy):
            S += KratInt(x, y)
            y += hy
        x += hx
    return S*hx*hy
```

Меню:

```
special1 = True
while special1 == True:
    special2 = True
    print("Welcome to MainMenu")
    print("Для выбора задачи нажмите соответствующую кнопку:")
    print('[1]Численное интегрирование\n'
          'more will comming soon')
    mean1 = int(input())

    if mean1 == 1:
        while special2 == True:
            special3 = True
            print("Выберите:\n"
                  "[1] С постоянным шагом\n"
                  "[2] С переменным шагом\n"
                  "[3] Кратный интегралл")
            print("Для выхода в меню введите '4'")
            mean2 = int(input())

            if mean2 == 1:
                print("Выбран постоянный шаг\n")
                while special3 == True:
                    print("Выберите метод:\n"
                          "[1] Метод левых прямоугольников\n"
                          "[2] Метод правых прямоугольников\n"
                          "[3] Метод Симпсона(парабол)\n"
                          "[4] Метод трапеций")
                    print("Для возврата нажмите '5'")
                    mean3 = int(input())

                    if mean3 == 1:
                        print("Выбран 'Метод левых прямоугольников'")
                        print('Введите количество разбиений')
                        n = int(input())
                        print('Введите нижнюю границу')
                        A = float(input())
                        print('Введите верхнюю границу')
                        B = float(input())
                        res = left(n, A, B)
                        print("Результат: ", res, '\n')
```

```
            elif mean3 == 2:
                print("Выбран 'Метод правых прямоугольников'")
                print('Введите количество разбиений')
                n = int(input())
                print('Введите нижнюю границу')
                A = float(input())
                print('Введите верхнюю границу')
                B = float(input())
                res = right(n, A, B)
                print("Результат: ", res, '\n')

            elif mean3 == 3:
                print("Выбран 'Метод Симпсона(парабол)'")
                print('Введите количество разбиений')
                n = int(input())
                print('Введите нижнюю границу')
                A = float(input())
                print('Введите верхнюю границу')
                B = float(input())
                res = parabola(n, A, B)
                print("Результат: ", res, '\n')

            elif mean3 == 4:
                print("Выбран 'Метод трапеций'")
                print('Введите количество разбиений')
                n = int(input())
                print('Введите нижнюю границу')
                A = float(input())
                print('Введите верхнюю границу')
                B = float(input())
                res = trapecia(n, A, B)
                print("Результат: ", res, '\n')

            elif mean3 == 5:
                special3 = False
                print()
```

```
        elif mean2 == 2:
            print("Выбран переменный шаг\n")
            while special3 == True:
                print("Выберите алгоритм:\n"
                      "[1] Двойной пересчёт алгоритм 1\n"
                      "[2] Двойной пересчёт алгоритм 2")
                print("Для возврата нажмите '3'")
                mean4 = int(input())

                if mean4 == 1:
                    print("Двойной пересчёт'\n"
                          "Алгоритм 1\n"
                          "Метод левых прямоугольников \n")

                    print('Введите точность')
                    e = float(input())
                    print('Введите нижнюю границу')
                    A = float(input())
                    print('Введите верхнюю границу')
                    B = float(input())
                    res = doubleRecalc(e, A, B)
                    print("Результат: ", res, '\n')

                elif mean4 == 2:
                    print("Двойной пересчёт\n"
                          "Алгоритм 2\n"
                          "Метод левых прямоугольников\n")
                    print('Введите точность')
                    e = float(input())
                    print('Введите нижнюю границу')
                    A = float(input())
                    print('Введите верхнюю границу')
                    B = float(input())
                    res = doubleRecalcBetter(e, A, B)
                    print("Результат: ", res, '\n')

                elif mean4 == 3:
                    special3 = False
                    print()
```

```
        elif mean2 == 3:
            print("Выбран кратный интеграл\n")
            print("Введите количество разбиений по x")
            nx = int(input())
            print("Введите количество разбиений по y")
            ny = int(input())
            print("Введите внешнюю нижнюю границу")
            A = float(input())
            print("Введите внешнюю верхнюю границу")
            B = float(input())
            print("Введите внутреннюю нижнюю границу")
            C = float(input())
            print("Введите внутреннюю верхнюю границу")
            D = float(input())
            res = kratniy(nx, ny, A, B, C, D)
            print("Результат",res,'\n')

        elif mean2 == 4:
            special2 = False
```

Код для Телеграм-бота:

Код меню:

```python
import telebot
from telebot import types
from math import sin

bot = telebot.TeleBot('TOKEN')

# Команда /start
@bot.message_handler(commands=["start"])
def start(m):
    bot.send_message(m.chat.id, 'Привет! Я бот студентов РГПУ им. Герцена :) Я решаю задание для Лабораторной Работы №1✌ ')
    bot.send_message(m.chat.id, 'Мои команды:\n/start\n/info\n/calculate')
    bot.send_message(m.chat.id, 'Для начала вычисления интеграла выражения x*x+5*x+3 жми  /calculate!')

@bot.message_handler(commands=["info"])
def info(m):
    bot.send_message(m.chat.id, 'Я бот студентов РГПУ им. Герцена :) Моих создателей зовут Максим Стецук, Крючкова Анастасия и Зухир Амира. Они из группы ИВТ 2-1')
    markup = types.InlineKeyboardMarkup()
    item1 = types.InlineKeyboardButton("Максим", url='https://vk.com/makstulenchik')
    item2 = types.InlineKeyboardButton("Анастасия", url='https://vk.com/nestessia')
    item3 = types.InlineKeyboardButton("Амира", url='https://vk.com/amirazhr')
    item4 = types.InlineKeyboardButton("Ангелина", url='https://vk.com/mintange')
    markup.add(item1)
    markup.add(item2)
    markup.add(item3)
    markup.add(item4)
    bot.send_message(m.chat.id, "{0.first_name}, для связи с нами переходи по ссылкам :)".format(m.from_user), reply_markup=markup)

# Команда /calculate
@bot.message_handler(commands=["calculate"])
def calculate(m):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True, one_time_keyboard=True)
    item1 = types.KeyboardButton("Постоянный шаг")
    item2 = types.KeyboardButton("Переменный шаг")
    item3 = types.KeyboardButton("Кратный интеграл")
    markup.add(item1, item2,item3)
    bot.send_message(m.chat.id, text="{0.first_name}, какой способ тебя интересует? ".format(m.from_user), reply_markup=markup)
```

```python
#Выбор метода вычислений
@bot.message_handler(content_types=['text'])
def method(m):
    if(m.text == "Постоянный шаг"):
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True, one_time_keyboard=True)
        item1 = types.KeyboardButton("Метод левых частей")
        item2 = types.KeyboardButton("Метод правых частей")
        item3 = types.KeyboardButton("Метод Симпсона")
        item4 = types.KeyboardButton("Метод трапеций")
        item5 = types.KeyboardButton("Назад")
        markup.add(item1, item2,item3,item4,item5)
        bot.send_message(m.chat.id, text="Какой метод тебя интересует?".format(m.from_user), reply_markup=markup)
    elif(m.text == "Переменный шаг"):
        markup = types.ReplyKeyboardMarkup(row_width=1, one_time_keyboard=True)
        item1 = types.KeyboardButton("Двойной пересчёт алгоритм 1")
        item2 = types.KeyboardButton("Двойной пересчёт алгоритм 2")
        item3 = types.KeyboardButton("Назад")
        markup.add(item1, item2, item3)
        bot.send_message(m.chat.id, text="Выберите алгоритм:".format(m.from_user), reply_markup=markup)

    elif(m.text == "Кратный интеграл"):
        bot.reply_to(m, 'Введите количество разбиений по x')
        bot.register_next_step_handler(m, get_nx)

    elif(m.text == "Метод левых частей"):
        bot.reply_to(m, 'Введи количество разбиений')
        bot.register_next_step_handler(m, get_n_for_left)
    elif(m.text == "Метод правых частей"):
        bot.reply_to(m, 'Введи количество разбиений')
        bot.register_next_step_handler(m, get_n_for_right)
    elif(m.text == "Метод Симпсона"):
        bot.reply_to(m, 'Введи количество разбиений')
        bot.register_next_step_handler(m, get_n_for_parabola)
    elif(m.text == "Метод трапеций"):
        bot.reply_to(m, 'Введи количество разбиений')
        bot.register_next_step_handler(m, get_n_for_trapecia)


    elif(m.text == "Двойной пересчёт алгоритм 1"):
        bot.reply_to(m, 'Введи точность')
        bot.register_next_step_handler(m, get_e_for_recalc1)
    elif(m.text == "Двойной пересчёт алгоритм 2"):
        bot.reply_to(m, 'Введи точность')
        bot.register_next_step_handler(m, get_e_for_recalc2)
```

```
    elif(m.text == "Назад"):
        return calculate(m)
```

```
#Вычисления
def task(x):
    return x*x+5*x+3
```

С постоянным шагом:

1) Метод левых прямоугольников:

```
#Для Метода левых прямоугольников
def get_n_for_left(m,user_result = None):
    try:
        global user_n
        if user_result == None:
            user_n = int(m.text)
        else:
            user_n = str(user_result)
        bot.send_message(m.chat.id, text='Введи нижнюю границу')
        bot.register_next_step_handler(m, get_a_for_left)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Попробуйте заново. ')

def get_a_for_left(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введи верхнюю границу')
        bot.register_next_step_handler(m, get_b_for_left)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так...  Вы точно ввели число? Проверьте корректност

def get_b_for_left(m):
    try:
        global user_b
        user_b = float(m.text)
        resultPrint_for_left(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность

def left():
    global user_a, user_b, user_n, user_result
    h = (user_b - user_a) / user_n
    x = user_a
    user_result = 0
    while x <= (user_b - h):
        user_result += task(x)
        x += h
    return h * user_result

def resultPrint_for_left(m):
    global user_a, user_b, user_n, user_result
    res = left()
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

2) Метод правых прямоугольников:

```python
#Для Метода правых прямоугольников
def get_n_for_right(m,user_result = None):
    try:
        global user_n

        if user_result == None:
            user_n = int(m.text)
        else:
            user_n = str(user_result)
        bot.send_message(m.chat.id, text='Введи нижнюю границу')
        bot.register_next_step_handler(m, get_a_for_right)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Попробуйте заново.')

def get_a_for_right(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введи верхнюю границу')
        bot.register_next_step_handler(m, get_b_for_right)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность

def get_b_for_right(m):
    try:
        global user_b
        user_b = float(m.text)
        resultPrint_for_right(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность

def right():
    global user_a, user_b, user_n, user_result
    h = (user_b - user_a) / user_n
    x = user_a + h
    result = 0
    while x <= user_b:
        result += task(x)
        x += h
    return h * result

def resultPrint_for_right(m):
    global user_a, user_b, user_n, user_result
    res = right()
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

3) Метод трапеции:

```python
#Для Метода трапеции
def get_n_for_trapecia(m,user_result = None):
    try:
        global user_n

        if user_result == None:
            user_n = int(m.text)
        else:
            user_n = str(user_result)
        bot.send_message(m.chat.id, text='Введи нижнюю границу')
        bot.register_next_step_handler(m, get_a_for_trapecia)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Попробуйте заново.')

def get_a_for_trapecia(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введи верхнюю границу')
        bot.register_next_step_handler(m, get_b_for_trapecia)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность н

def get_b_for_trapecia(m):
    try:
        global user_b
        user_b = float(m.text)
        resultPrint_for_trapecia(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность н

def trapecia():
    global user_a, user_b, user_n, user_result
    h = (user_b - user_a) / user_n
    x = user_a + h
    result = 0
    while x <= (user_b - h):
        result += task(x)
        x += h
    return ((task(user_a) + task(user_b)) / 2 + result) * h

def resultPrint_for_trapecia(m):
    global user_a, user_b, user_n, user_result
    res = trapecia()
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

4) Метод парабол:

```python
#Для Метода парабол
def get_n_for_parabola(m,user_result = None):
    try:
        global user_n

        if user_result == None:
            user_n = int(m.text)
        else:
            user_n = str(user_result)
        bot.send_message(m.chat.id, text='Введи нижнюю границу')
        bot.register_next_step_handler(m, get_a_for_parabola)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так...  Вы точно ввели число? Попробуйте заново.')

def get_a_for_parabola(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введи верхнюю границу')
        bot.register_next_step_handler(m, get_b_for_parabola)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность н

def get_b_for_parabola(m):
    try:
        global user_b
        user_b = float(m.text)
        resultPrint_for_parabola(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность н
```

```python
def parabola():
    global user_a, user_b, user_n, user_result
    h = (user_b - user_a) / user_n
    S1 = 0
    x = user_a + h
    while x <= (user_b - h):
        S1 += task(x)
        x += 2 * h
    S2 = 0
    x = user_a + (2 * h)
    while x <= (user_b - (2 * h)):
        S2 += task(x)
        x += 2 * h
    return (task(user_a) + task(user_b) + (4 * S1) + (2 * S2)) * (h / 3)

def resultPrint_for_parabola(m):
    global user_a, user_b, user_n, user_result
    res = parabola()
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

С переменным шагом:

1- Алгоритм 1

```python
#Двойной перерасчет 1
def get_e_for_recalc1(m,user_result = None):
    try:
        global user_e
        if user_result == None:
            user_e = float(m.text)
        else:
            user_e = str(user_result)
        bot.send_message(m.chat.id, text='Введи нижнюю границу')
        bot.register_next_step_handler(m, get_a_for_recalc1)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте ко

def get_a_for_recalc1(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введи верхнюю границу')
        bot.register_next_step_handler(m, get_b_for_recalc1)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте ко

def get_b_for_recalc1(m):
    try:
        global user_b
        user_b = float(m.text)
        resultPrint_for_recalc1(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте ко
```

```python
def doubleRecalc(user_e,user_a, user_b ):
    n = 2
    f1 = 0
    f2 = left_for_recalc(n, user_a, user_b)
    while abs(f1-f2) > user_e:
        f1 = f2
        n = n * 2
        f2 = left_for_recalc(n, user_a, user_b)
    return f2

def left_for_recalc(n, user_a, user_b):
    h = (user_b - user_a) / n
    x = user_a
    user_result = 0
    while x <= (user_b - h):
        user_result += task(x)
        x += h
    return h * user_result

def resultPrint_for_recalc1(m):
    res = doubleRecalc(user_e,user_a, user_b)
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

2- Алгоритм 2

```python
#Двойной перерасчет 2
def get_e_for_recalc2(m,user_result = None):
    try:
        global user_e
        if user_result == None:
            user_e = float(m.text)
        else:
            user_e = str(user_result)
        bot.send_message(m.chat.id, text='Введи нижнюю границу')
        bot.register_next_step_handler(m, get_a_for_recalc2)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Скорее всего, вы ввели не число. Попробуйте заново ')

def get_a_for_recalc2(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введи верхнюю границу')
        bot.register_next_step_handler(m, get_b_for_recalc2)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность написания веще

def get_b_for_recalc2(m):
    try:
        global user_b
        user_b = float(m.text)
        resultPrint_for_recalc2(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность написания веще
```

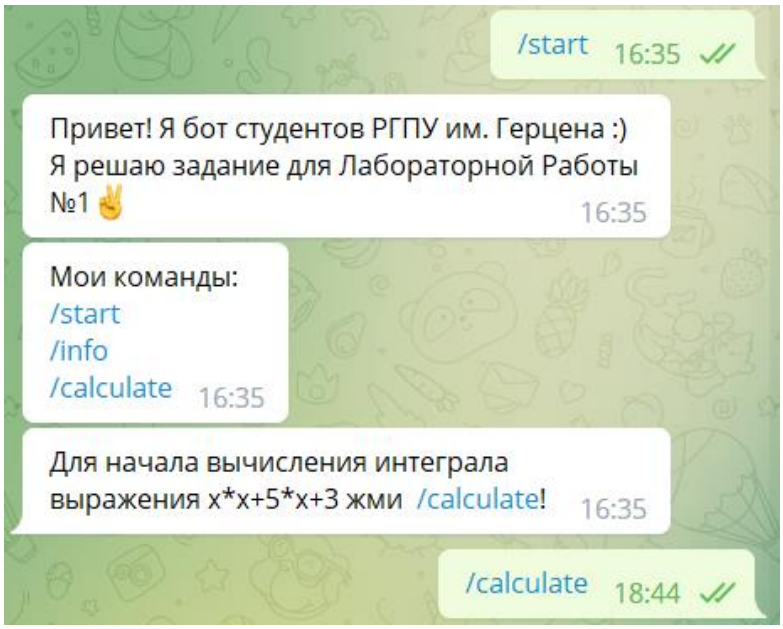
```python
def doubleRecalcBetter(user_e,user_a, user_b):
    n = 2
    otst = (user_b - user_a)/2
    f1 = left_for_recalc(n, user_a, user_b)
    otst = otst / 2
    f2 = left_for_recalc(n, user_a + otst, user_b)
    while abs(f1-f2) > user_e:
        f1 = f2
        n = n * 2
        otst = otst / 2
        f2 = left_for_recalc(n, user_a+otst, user_b)
    return f2

def resultPrint_for_recalc2(m):
    res = doubleRecalcBetter(user_e,user_a, user_b)
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

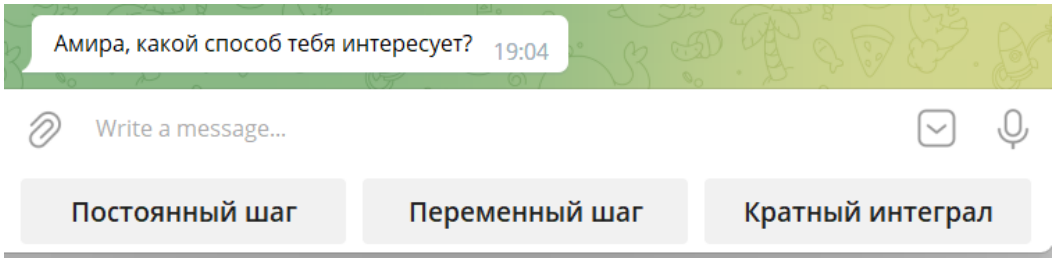
Кратный интеграл:

```python
#Для кратного интеграла
def get_nx(m,user_result = None):
    try:
        global user_nx

        if user_result == None:
            user_nx = float(m.text)
        else:
            user_nx = str(user_result)
        bot.send_message(m.chat.id, text='Введите количество разбиений по y')
        bot.register_next_step_handler(m, get_ny)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корре

def get_ny(m):
    try:
        global user_ny
        user_ny = float(m.text)
        bot.send_message(m.chat.id, text='Введите внешнюю нижнюю границу')
        bot.register_next_step_handler(m, get_a)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корре

def get_a(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введите внешнюю верхнюю границу')
        bot.register_next_step_handler(m, get_b)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корре

def get_b(m):
    try:
        global user_b
        user_b = float(m.text)
        bot.send_message(m.chat.id, text='Введите внутреннюю нижнюю границу')
        bot.register_next_step_handler(m, get_c)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корре
```

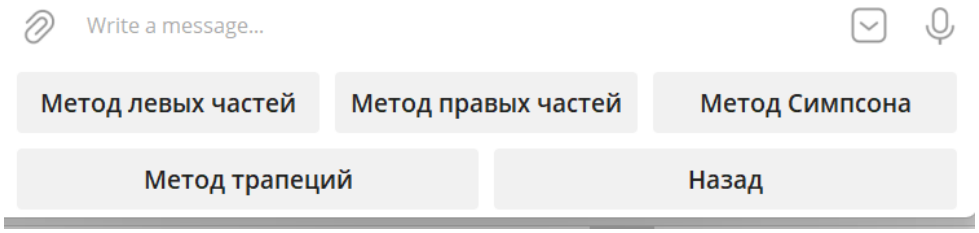
```python
def get_c(m):
    try:
        global user_c
        user_c = float(m.text)
        bot.send_message(m.chat.id, text='Введите внутреннюю верхнюю границу')
        bot.register_next_step_handler(m, get_d)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректно

def get_d(m):
    try:
        global user_d
        user_d = float(m.text)
        resultPrint_for_KratInt(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректно

def KratInt(x,y):
    return sin(x+y)

def calc_kratniy():
    global user_nx, user_ny, user_a, user_b, user_c, user_d
    hx = (user_b - user_a) / user_nx
    hy = (user_d - user_c) / user_ny
    S = 0
    x = user_a
    while x <= (user_b - hx):
        y = user_c
        while y <= (user_d - hy):
            S += KratInt(x, y)
            y += hy
        x += hx
    return S*hx*hy

def resultPrint_for_KratInt(m):
    global user_nx, user_ny, user_a, user_b, user_c, user_d, user_result
    res = calc_kratniy()
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

Приложение 2

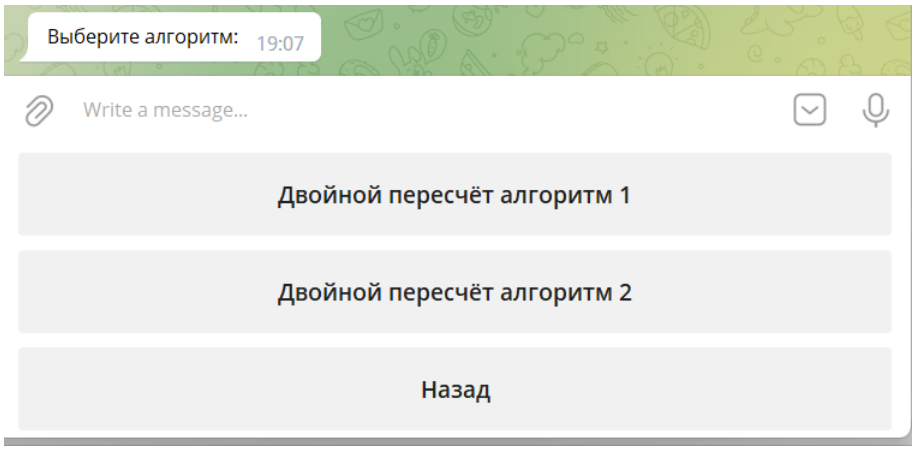




При выборе “постоянный шаг” нам предлагаются следующие методы:



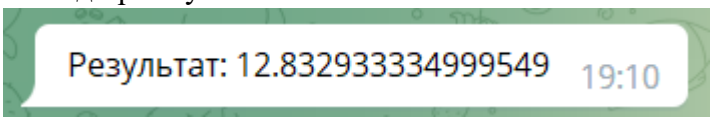
При выборе “переменный шаг” нам предлагаются 2 метода:



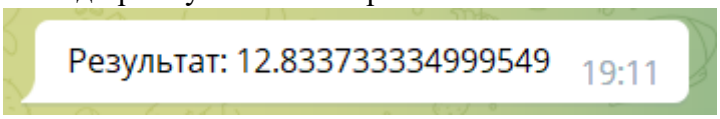
Результаты вычисления интеграла:

С постоянным шагом (с количеством разбиений 10000):

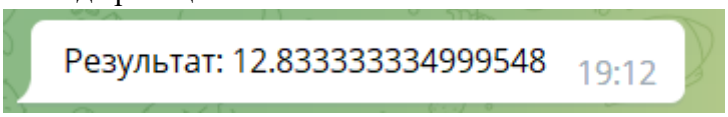
1- Метод прямоугольников левых частей



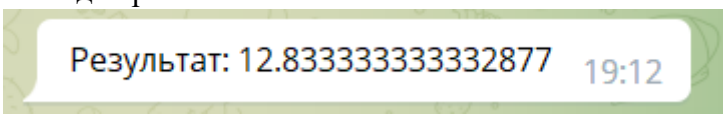
2- Метод прямоугольников правых частей



3- Метод трапеций

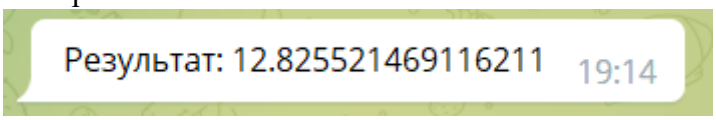


4- Метод парабол

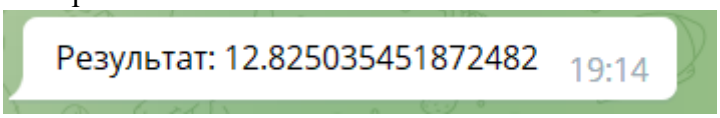


С переменным шагом (с точностью 0.01):

1- Алгоритм 1



2- Алгоритм 2



Максимова Ангелина ИВТ 2.1

Лабораторная работа №1

Численное интегрирование

Цель лабораторной работы: вычислить определенный интеграл, используя различные численные методы и алгоритмы их реализации.

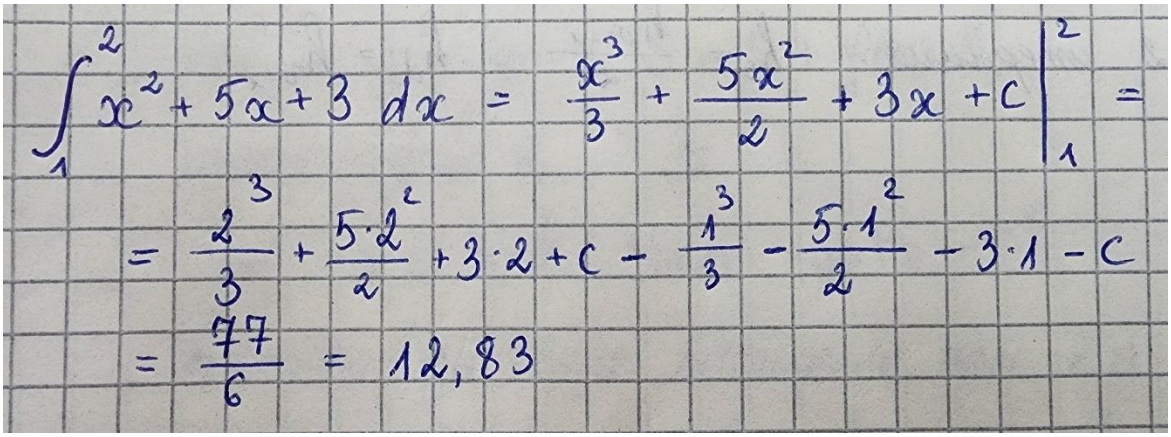
Инструменты: PyCharm, Telegram.

В рамках данной лабораторной работы, был использовал язык программирования Python 3.10

Мы использовали подынтегральную функцию: $x^2 + 5x + 3$.

В качестве примера определенного интеграла возьмем: $\int_1^2 x^2 + 5x + 3\ dx$

Вычислим данный интеграл ручным способом:



При вычислении, получили следующие результаты определенного интеграла с постоянным шагом:

Кол-во разбиений	Метод прямоугольников левых частей	Метод прямоугольников правых частей	Метод трапеций	Метод парабол
100	12.6242	12.7034	12.6642	12.4957
1000	12.8293	12.8373	12.8333	12.7993
10000	12.8329	12.8337	12.8333	12.8333

Консоль и Телеграм-бот выдают одинаковые результаты.

Код и работа Телеграм-бота приведена в приложении 1 и 2, соответственно.(страницы 66 и 81)

Вывод: в ходе сравнения машинных вычислений и ручного вычисления, мы пришли к выводу, что результаты практически идентичны и чем больше количество разбиений, тем точнее выводится результат.

Определенный интеграл с переменным шагом (2 алгоритма)

Точность	Двойной пересчет алгоритм 1	Двойной пересчет алгоритм 2
0.1	12.8255	12.8250
0.001	12.8323	12.8328
0.0001	12.8332	12.8332

Вывод: два данных алгоритма, прекрасно посчитали интеграл и сошли с результатом ручного вычисления, но в данном случае чем меньше точность, тем точнее результат.

Кратный интеграл

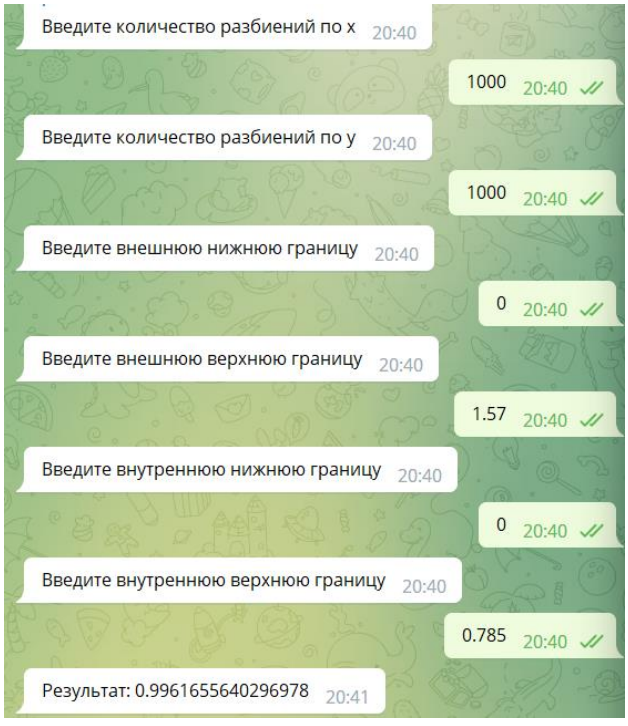
Возьмем кратный интеграл:

$$\int_0^{\pi/2} dx \int_0^{\pi/4} Sin(x+y)dy; \qquad n = 4$$

И вычислим его:

$$\int_0^{\pi/2} \int_0^{\pi/4} Sin(x+y)dxdy \approx \left(\frac{\pi}{24}\right)^2 + [\dots + \dots] = 1{,}00028$$

В нашей программе будет считать, что число пи это 3.14, соответственно пи деленное на 4 это 0.785, а пи деленное на 2 это 1.57.



Вывод: Результаты ручного вычисления и машинного приблизительно равны, погрешность возникла из-за примерного взятия числа пи, а также при увеличении кол-ва разбиений результат будет еще точнее.

Вывод по всей лабораторной работе: Мы научились реализовывать интегрирование и вычислили кратный интеграл, создали программу для данных вычислений и на ее основе создали Телеграм-бота.

Ссылка на Телеграм-бота: ссылка

Приложение 1

Код :

```
from math import sqrt
from math import sin

def task(x: float):
    return x*x+5*x+3

def KratInt(x:float, y:float):
    return sin(x+y)
```

С постоянным шагом:

1. Метод левых прямоугольников:

```
def left(n: int, A:float, B:float):
    h = (B - A) / n
    x = A
    result = 0
    while x <= (B - h):
        result += task(x)
        x += h
    return h * result
```

2. Метод правых прямоугольников:

```
def right(n: int, A:float, B:float):
    h = (B - A) / n
    x = A + h
    result = 0
    while x <= B:
        result += task(x)
        x += h
    return h * result
```

3. Матод трапеций:

```
def trapecia(n: int, A:float, B:float):
    h = (B - A) / n
    x = A + h
    result = 0
    while x <= (B - h):
        result += task(x)
        x += h
    return ((task(A) + task(B)) / 2 + result) * h
```

4. Метод парабол:

```
def parabola(n: int, A:float, B:float):
    h = (B - A) / n
    S1 = 0
    x = A + h
    while x <= (B - h):
        S1 += task(x)
        x += 2 * h
    S2 = 0
    x = A + (2 * h)
    while x <= (B - (2 * h)):
        S2 += task(x)
        x += 2 * h
    return (task(A) + task(B) + (4 * S1) + (2 * S2)) * (h / 3)
```

С переменным шагом:

1- Алгоритм 1:

```
def doubleRecalc(e:float, A:float, B:float):
    n = 2
    f1 = 0
    f2 = left(n, A, B)
    while abs(f1-f2) > e:
        f1 = f2
        n = n * 2
        f2 = left(n, A, B)
    return f2
```

2- Алгоритм 2:

```
def doubleRecalcBetter(e:float, A:float, B:float):
    n = 2
    otst = (B - A)/2
    f1 = left(n, A, B)
    otst = otst / 2
    f2 = left(n, A + otst, B)
    while abs(f1-f2) > e:
        f1 = f2
        n = n * 2
        otst = otst / 2
        f2 = left(n, A+otst, B)
    return f2
```

Кратный интеграл:

```
def kratniy(nx:int, ny:int, A:float, B:float, C:float, D:float):
    hx = (B - A) / nx
    hy = (D - C) / ny
    S = 0
    x = A
    while x <= (B - hx):
        y = C
        while y <= (D - hy):
            S += KratInt(x, y)
            y += hy
        x += hx
    return S*hx*hy
```

Меню:

```
special1 = True
while special1 == True:
    special2 = True
    print("Welcome to MainMenu")
    print("Для выбора задачи нажмите соответствующую кнопку:")
    print('[1]Численное интегрирование\n'
          'more will comming soon')
    mean1 = int(input())

    if mean1 == 1:
        while special2 == True:
            special3 = True
            print("Выберите:\n"
                  "[1] С постоянным шагом\n"
                  "[2] С переменным шагом\n"
                  "[3] Кратный интегралл")
            print("Для выхода в меню введите '4'")
            mean2 = int(input())

            if mean2 == 1:
                print("Выбран постоянный шаг\n")
                while special3 == True:
                    print("Выберите метод:\n"
                          "[1] Метод левых прямоугольников\n"
                          "[2] Метод правых прямоугольников\n"
                          "[3] Метод Симпсона(парабол)\n"
                          "[4] Метод трапеций")
                    print("Для возврата нажмите '5'")
                    mean3 = int(input())

                    if mean3 == 1:
                        print("Выбран 'Метод левых прямоугольников'")
                        print('Введите количество разбиений')
                        n = int(input())
                        print('Введите нижнюю границу')
                        A = float(input())
                        print('Введите верхнюю границу')
                        B = float(input())
                        res = left(n, A, B)
                        print("Результат: ", res, '\n')
```

```
            elif mean3 == 2:
                print("Выбран 'Метод правых прямоугольников'")
                print('Введите количество разбиений')
                n = int(input())
                print('Введите нижнюю границу')
                A = float(input())
                print('Введите верхнюю границу')
                B = float(input())
                res = right(n, A, B)
                print("Результат: ", res, '\n')

            elif mean3 == 3:
                print("Выбран 'Метод Симпсона(парабол)'")
                print('Введите количество разбиений')
                n = int(input())
                print('Введите нижнюю границу')
                A = float(input())
                print('Введите верхнюю границу')
                B = float(input())
                res = parabola(n, A, B)
                print("Результат: ", res, '\n')

            elif mean3 == 4:
                print("Выбран 'Метод трапеций'")
                print('Введите количество разбиений')
                n = int(input())
                print('Введите нижнюю границу')
                A = float(input())
                print('Введите верхнюю границу')
                B = float(input())
                res = trapecia(n, A, B)
                print("Результат: ", res, '\n')

            elif mean3 == 5:
                special3 = False
                print()
```

```
        elif mean2 == 2:
            print("Выбран переменный шаг\n")
            while special3 == True:
                print("Выберите алгоритм:\n"
                      "[1] Двойной пересчёт алгоритм 1\n"
                      "[2] Двойной пересчёт алгоритм 2")
                print("Для возврата нажмите '3'")
                mean4 = int(input())

                if mean4 == 1:
                    print("Двойной пересчёт'\n"
                          "Алгоритм 1\n"
                          "Метод левых прямоугольников \n")

                    print('Введите точность')
                    e = float(input())
                    print('Введите нижнюю границу')
                    A = float(input())
                    print('Введите верхнюю границу')
                    B = float(input())
                    res = doubleRecalc(e, A, B)
                    print("Результат: ", res, '\n')

                elif mean4 == 2:
                    print("Двойной пересчёт\n"
                          "Алгоритм 2\n"
                          "Метод левых прямоугольников\n")
                    print('Введите точность')
                    e = float(input())
                    print('Введите нижнюю границу')
                    A = float(input())
                    print('Введите верхнюю границу')
                    B = float(input())
                    res = doubleRecalcBetter(e, A, B)
                    print("Результат: ", res, '\n')

                elif mean4 == 3:
                    special3 = False
                    print()
```

```
        elif mean2 == 3:
            print("Выбран кратный интеграл\n")
            print("Введите количество разбиений по x")
            nx = int(input())
            print("Введите количество разбиений по y")
            ny = int(input())
            print("Введите внешнюю нижнюю границу")
            A = float(input())
            print("Введите внешнюю верхнюю границу")
            B = float(input())
            print("Введите внутреннюю нижнюю границу")
            C = float(input())
            print("Введите внутреннюю верхнюю границу")
            D = float(input())
            res = kratniy(nx, ny, A, B, C, D)
            print("Результат",res,'\n')

        elif mean2 == 4:
            special2 = False
```

Код для Телеграм-бота:

Код меню:

```python
import telebot
from telebot import types
from math import sin

bot = telebot.TeleBot('TOKEN')

# Команда /start
@bot.message_handler(commands=["start"])
def start(m):
    bot.send_message(m.chat.id, 'Привет! Я бот студентов РГПУ им. Герцена :) Я решаю задание для Лабораторной Работы №1✌ ')
    bot.send_message(m.chat.id, 'Мои команды:\n/start\n/info\n/calculate')
    bot.send_message(m.chat.id, 'Для начала вычисления интеграла выражения x*x+5*x+3 жми  /calculate!')

@bot.message_handler(commands=["info"])
def info(m):
    bot.send_message(m.chat.id, 'Я бот студентов РГПУ им. Герцена :) Моих создателей зовут Максим Стецук, Крючкова Анастасия и Зухир Амира. Они из группы ИВТ 2-1')
    markup = types.InlineKeyboardMarkup()
    item1 = types.InlineKeyboardButton("Максим", url='https://vk.com/makstulenchik')
    item2 = types.InlineKeyboardButton("Анастасия", url='https://vk.com/nestessia')
    item3 = types.InlineKeyboardButton("Амира", url='https://vk.com/amirazhr')
    item4 = types.InlineKeyboardButton("Ангелина", url='https://vk.com/mintange')
    markup.add(item1)
    markup.add(item2)
    markup.add(item3)
    markup.add(item4)
    bot.send_message(m.chat.id, "{0.first_name}, для связи с нами переходи по ссылкам :)".format(m.from_user), reply_markup=markup)

# Команда /calculate
@bot.message_handler(commands=["calculate"])
def calculate(m):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True, one_time_keyboard=True)
    item1 = types.KeyboardButton("Постоянный шаг")
    item2 = types.KeyboardButton("Переменный шаг")
    item3 = types.KeyboardButton("Кратный интеграл")
    markup.add(item1, item2,item3)
    bot.send_message(m.chat.id, text="{0.first_name}, какой способ тебя интересует? ".format(m.from_user), reply_markup=markup)
```

```python
#Выбор метода вычислений
@bot.message_handler(content_types=['text'])
def method(m):
    if(m.text == "Постоянный шаг"):
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True, one_time_keyboard=True)
        item1 = types.KeyboardButton("Метод левых частей")
        item2 = types.KeyboardButton("Метод правых частей")
        item3 = types.KeyboardButton("Метод Симпсона")
        item4 = types.KeyboardButton("Метод трапеций")
        item5 = types.KeyboardButton("Назад")
        markup.add(item1, item2,item3,item4,item5)
        bot.send_message(m.chat.id, text="Какой метод тебя интересует?".format(m.from_user), reply_markup=markup)
    elif(m.text == "Переменный шаг"):
        markup = types.ReplyKeyboardMarkup(row_width=1, one_time_keyboard=True)
        item1 = types.KeyboardButton("Двойной пересчёт алгоритм 1")
        item2 = types.KeyboardButton("Двойной пересчёт алгоритм 2")
        item3 = types.KeyboardButton("Назад")
        markup.add(item1, item2, item3)
        bot.send_message(m.chat.id, text="Выберите алгоритм:".format(m.from_user), reply_markup=markup)

    elif(m.text == "Кратный интеграл"):
        bot.reply_to(m, 'Введите количество разбиений по x')
        bot.register_next_step_handler(m, get_nx)

    elif(m.text == "Метод левых частей"):
        bot.reply_to(m, 'Введи количество разбиений')
        bot.register_next_step_handler(m, get_n_for_left)
    elif(m.text == "Метод правых частей"):
        bot.reply_to(m, 'Введи количество разбиений')
        bot.register_next_step_handler(m, get_n_for_right)
    elif(m.text == "Метод Симпсона"):
        bot.reply_to(m, 'Введи количество разбиений')
        bot.register_next_step_handler(m, get_n_for_parabola)
    elif(m.text == "Метод трапеций"):
        bot.reply_to(m, 'Введи количество разбиений')
        bot.register_next_step_handler(m, get_n_for_trapecia)


    elif(m.text == "Двойной пересчёт алгоритм 1"):
        bot.reply_to(m, 'Введи точность')
        bot.register_next_step_handler(m, get_e_for_recalc1)
    elif(m.text == "Двойной пересчёт алгоритм 2"):
        bot.reply_to(m, 'Введи точность')
        bot.register_next_step_handler(m, get_e_for_recalc2)
```

```
    elif(m.text == "Назад"):
        return calculate(m)
```

```
#Вычисления
def task(x):
    return x*x+5*x+3
```

С постоянным шагом:

1) Метод левых прямоугольников:

```
#Для Метода левых прямоугольников
def get_n_for_left(m,user_result = None):
    try:
        global user_n
        if user_result == None:
            user_n = int(m.text)
        else:
            user_n = str(user_result)
        bot.send_message(m.chat.id, text='Введи нижнюю границу')
        bot.register_next_step_handler(m, get_a_for_left)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Попробуйте заново. ')

def get_a_for_left(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введи верхнюю границу')
        bot.register_next_step_handler(m, get_b_for_left)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так...  Вы точно ввели число? Проверьте корректност

def get_b_for_left(m):
    try:
        global user_b
        user_b = float(m.text)
        resultPrint_for_left(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность

def left():
    global user_a, user_b, user_n, user_result
    h = (user_b - user_a) / user_n
    x = user_a
    user_result = 0
    while x <= (user_b - h):
        user_result += task(x)
        x += h
    return h * user_result

def resultPrint_for_left(m):
    global user_a, user_b, user_n, user_result
    res = left()
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

2) Метод правых прямоугольников:

```python
#Для Метода правых прямоугольников
def get_n_for_right(m,user_result = None):
    try:
        global user_n

        if user_result == None:
            user_n = int(m.text)
        else:
            user_n = str(user_result)
        bot.send_message(m.chat.id, text='Введи нижнюю границу')
        bot.register_next_step_handler(m, get_a_for_right)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Попробуйте заново.')

def get_a_for_right(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введи верхнюю границу')
        bot.register_next_step_handler(m, get_b_for_right)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность

def get_b_for_right(m):
    try:
        global user_b
        user_b = float(m.text)
        resultPrint_for_right(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность

def right():
    global user_a, user_b, user_n, user_result
    h = (user_b - user_a) / user_n
    x = user_a + h
    result = 0
    while x <= user_b:
        result += task(x)
        x += h
    return h * result

def resultPrint_for_right(m):
    global user_a, user_b, user_n, user_result
    res = right()
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

3) Метод трапеции:

```python
#Для Метода трапеции
def get_n_for_trapecia(m,user_result = None):
    try:
        global user_n

        if user_result == None:
            user_n = int(m.text)
        else:
            user_n = str(user_result)
        bot.send_message(m.chat.id, text='Введи нижнюю границу')
        bot.register_next_step_handler(m, get_a_for_trapecia)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Попробуйте заново.')

def get_a_for_trapecia(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введи верхнюю границу')
        bot.register_next_step_handler(m, get_b_for_trapecia)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность н

def get_b_for_trapecia(m):
    try:
        global user_b
        user_b = float(m.text)
        resultPrint_for_trapecia(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность н

def trapecia():
    global user_a, user_b, user_n, user_result
    h = (user_b - user_a) / user_n
    x = user_a + h
    result = 0
    while x <= (user_b - h):
        result += task(x)
        x += h
    return ((task(user_a) + task(user_b)) / 2 + result) * h

def resultPrint_for_trapecia(m):
    global user_a, user_b, user_n, user_result
    res = trapecia()
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

4) Метод парабол:

```python
#Для Метода парабол
def get_n_for_parabola(m,user_result = None):
    try:
        global user_n

        if user_result == None:
            user_n = int(m.text)
        else:
            user_n = str(user_result)
        bot.send_message(m.chat.id, text='Введи нижнюю границу')
        bot.register_next_step_handler(m, get_a_for_parabola)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так...  Вы точно ввели число? Попробуйте заново.')

def get_a_for_parabola(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введи верхнюю границу')
        bot.register_next_step_handler(m, get_b_for_parabola)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность н

def get_b_for_parabola(m):
    try:
        global user_b
        user_b = float(m.text)
        resultPrint_for_parabola(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность н
```

```python
def parabola():
    global user_a, user_b, user_n, user_result
    h = (user_b - user_a) / user_n
    S1 = 0
    x = user_a + h
    while x <= (user_b - h):
        S1 += task(x)
        x += 2 * h
    S2 = 0
    x = user_a + (2 * h)
    while x <= (user_b - (2 * h)):
        S2 += task(x)
        x += 2 * h
    return (task(user_a) + task(user_b) + (4 * S1) + (2 * S2)) * (h / 3)

def resultPrint_for_parabola(m):
    global user_a, user_b, user_n, user_result
    res = parabola()
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

С переменным шагом:

1- Алгоритм 1

```python
#Двойной перерасчет 1
def get_e_for_recalc1(m,user_result = None):
    try:
        global user_e
        if user_result == None:
            user_e = float(m.text)
        else:
            user_e = str(user_result)
        bot.send_message(m.chat.id, text='Введи нижнюю границу')
        bot.register_next_step_handler(m, get_a_for_recalc1)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте ко

def get_a_for_recalc1(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введи верхнюю границу')
        bot.register_next_step_handler(m, get_b_for_recalc1)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте ко

def get_b_for_recalc1(m):
    try:
        global user_b
        user_b = float(m.text)
        resultPrint_for_recalc1(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте ко
```

```python
def doubleRecalc(user_e,user_a, user_b ):
    n = 2
    f1 = 0
    f2 = left_for_recalc(n, user_a, user_b)
    while abs(f1-f2) > user_e:
        f1 = f2
        n = n * 2
        f2 = left_for_recalc(n, user_a, user_b)
    return f2

def left_for_recalc(n, user_a, user_b):
    h = (user_b - user_a) / n
    x = user_a
    user_result = 0
    while x <= (user_b - h):
        user_result += task(x)
        x += h
    return h * user_result

def resultPrint_for_recalc1(m):
    res = doubleRecalc(user_e,user_a, user_b)
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

2- Алгоритм 2

```python
#Двойной перерасчет 2
def get_e_for_recalc2(m,user_result = None):
    try:
        global user_e
        if user_result == None:
            user_e = float(m.text)
        else:
            user_e = str(user_result)
        bot.send_message(m.chat.id, text='Введи нижнюю границу')
        bot.register_next_step_handler(m, get_a_for_recalc2)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Скорее всего, вы ввели не число. Попробуйте заново ')

def get_a_for_recalc2(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введи верхнюю границу')
        bot.register_next_step_handler(m, get_b_for_recalc2)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность написания веще

def get_b_for_recalc2(m):
    try:
        global user_b
        user_b = float(m.text)
        resultPrint_for_recalc2(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректность написания веще
```

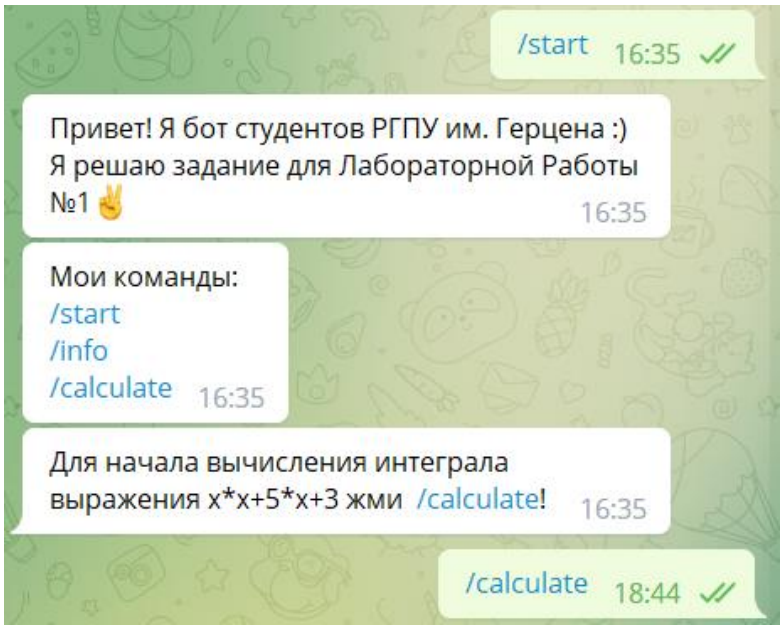
```python
def doubleRecalcBetter(user_e,user_a, user_b):
    n = 2
    otst = (user_b - user_a)/2
    f1 = left_for_recalc(n, user_a, user_b)
    otst = otst / 2
    f2 = left_for_recalc(n, user_a + otst, user_b)
    while abs(f1-f2) > user_e:
        f1 = f2
        n = n * 2
        otst = otst / 2
        f2 = left_for_recalc(n, user_a+otst, user_b)
    return f2

def resultPrint_for_recalc2(m):
    res = doubleRecalcBetter(user_e,user_a, user_b)
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

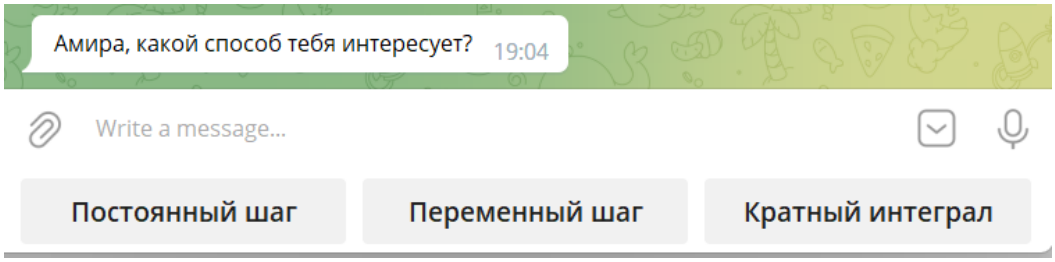
Кратный интеграл:

```
#Для кратного интеграла
def get_nx(m,user_result = None):
    try:
        global user_nx

        if user_result == None:
            user_nx = float(m.text)
        else:
            user_nx = str(user_result)
        bot.send_message(m.chat.id, text='Введите количество разбиений по y')
        bot.register_next_step_handler(m, get_ny)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корре

def get_ny(m):
    try:
        global user_ny
        user_ny = float(m.text)
        bot.send_message(m.chat.id, text='Введите внешнюю нижнюю границу')
        bot.register_next_step_handler(m, get_a)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корре

def get_a(m):
    try:
        global user_a
        user_a = float(m.text)
        bot.send_message(m.chat.id, text='Введите внешнюю верхнюю границу')
        bot.register_next_step_handler(m, get_b)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корре

def get_b(m):
    try:
        global user_b
        user_b = float(m.text)
        bot.send_message(m.chat.id, text='Введите внутреннюю нижнюю границу')
        bot.register_next_step_handler(m, get_c)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корре
```

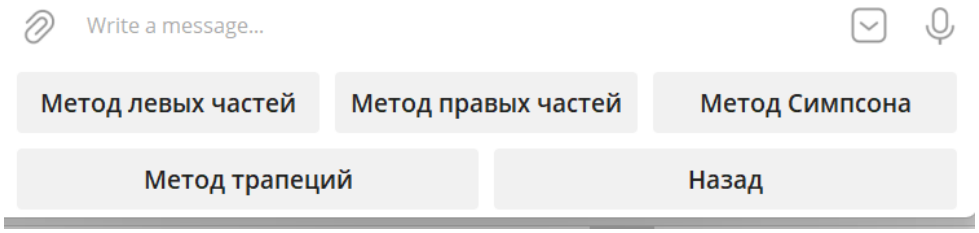
```
def get_c(m):
    try:
        global user_c
        user_c = float(m.text)
        bot.send_message(m.chat.id, text='Введите внутреннюю верхнюю границу')
        bot.register_next_step_handler(m, get_d)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректно

def get_d(m):
    try:
        global user_d
        user_d = float(m.text)
        resultPrint_for_KratInt(m)
    except Exception as e:
        bot.reply_to(m, 'Что-то пошло не так... Вы точно ввели число? Проверьте корректно

def KratInt(x,y):
    return sin(x+y)

def calc_kratniy():
    global user_nx, user_ny, user_a, user_b, user_c, user_d
    hx = (user_b - user_a) / user_nx
    hy = (user_d - user_c) / user_ny
    S = 0
    x = user_a
    while x <= (user_b - hx):
        y = user_c
        while y <= (user_d - hy):
            S += KratInt(x, y)
            y += hy
        x += hx
    return S*hx*hy

def resultPrint_for_KratInt(m):
    global user_nx, user_ny, user_a, user_b, user_c, user_d, user_result
    res = calc_kratniy()
    bot.send_message(m.chat.id, "Результат: " + (str(res)))
```

Приложение 2

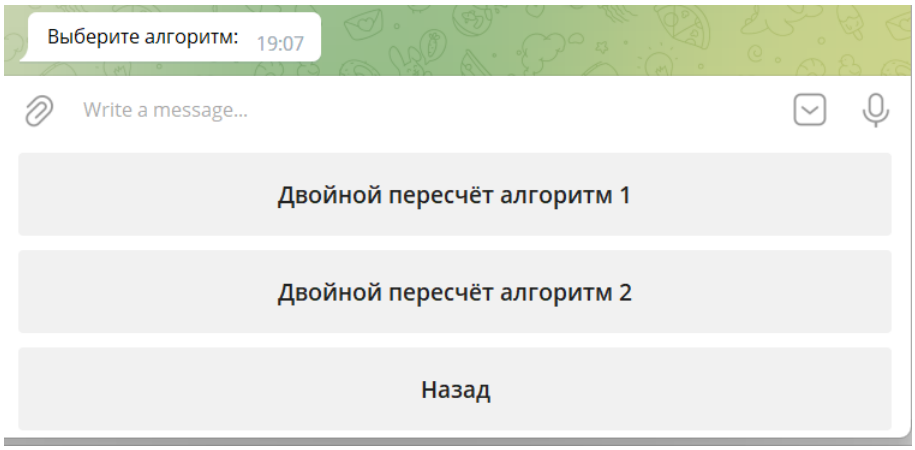




При выборе “постоянный шаг” нам предлагаются следующие методы:



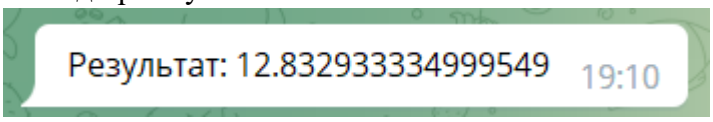
При выборе “переменный шаг” нам предлагаются 2 метода:



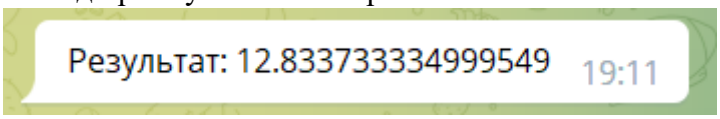
Результаты вычисления интеграла:

С постоянным шагом (с количеством разбиений 10000):

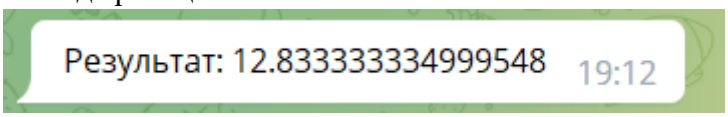
1- Метод прямоугольников левых частей



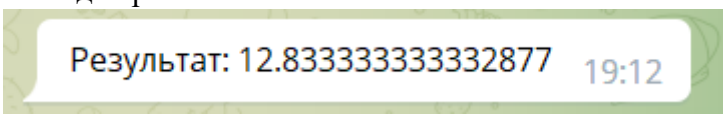
2- Метод прямоугольников правых частей



3- Метод трапеций

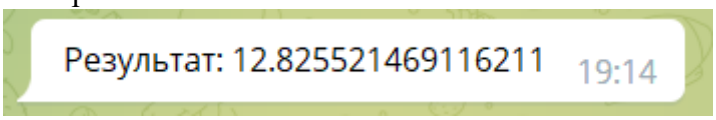


4- Метод парабол



С переменным шагом (с точностью 0.01):

1- Алгоритм 1



2- Алгоритм 2