МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет науки и технологий

имени академика М.Ф. Решетнева»

Кафедра высшей математики

Проектная работа

по дисциплине «Математический анализ»

**«Вычисление определённых интегралов»**

**Выполнил:**

студент 1-го курса

гр. БПИ21-01

Кузнецов Д.С.

**Проверил:**

доцент кафедры ВМ

Семёнкина М.Е.

Красноярск, 2022

**1. Краткая постановка задачи**

1.1. Цель работы: изучение возможных путей для вычисления определенных интегралов с помощью компьютера.

1.2. Задания к работе.

1) Реализовать численное нахождение определённых интегралов от произвольной функции (квадратурные формулы треугольников, трапеций, Симпсона).

2) Выполнить тестирование на 8 различных тестовых интегралах. Сравнить погрешности полученных результатов. (Интегралы для точной оценки можно вычислить вручную).

3) Построить графики функций и квадратур. Сравнить погрешности полученных для различного числа точек, разбивающих отрезок.

**2. Краткое описание выполнения задания**

2.1. План выполнения работы:

1. выбрать 8 тестовых интегралов, найти их точные или приближенные значения прямым интегрированием, а также по квадратурным формулам при m=1, m=10;
2. сравнить погрешности вычислений по разным формулам;
3. построить графики функций и квадратур;
4. найти приближенные значения интегралов с помощью программы при m=1, m=10, m=100, m=1000, m=10000 и расчётные погрешности;
5. сравнить погрешности для разного числа точек, разбивающих отрезок;
6. сделать выводы по проделанной работе.

2.2. Квадратурные формулы [h = (b-a)/m]:

1. формула прямоугольников:

a) простая: S = (b-a) \* f((a+b)/2);

б) составная: S = h \* (f(a+h/2) + f(a+3h/2) + ... + f(a+(2m-1)h/2));

1. формула трапеций:

a) простая: S = (b-a)/2 \* (f(a) + f(b));

б) составная: S = h/2 \* (f(a) + 2\*f(a+h) + ... + 2\*f(b-h) + f(b));

1. формула Симпсона:

a) простая: S = (b-a)/6 \* (f(a) + 4\*f((a+b)/2) + f(b));

б) составная: S = h/6 \* (f(a) + f(b) + 2\*(f(a+h)+...+f(b-h)) + 4\*(f(a+h/2)+...+f(b-h/2))).

**3. Графики и результаты вычислений**

3.1. Тестовые интегралы:

1. 0∫20 x dx = x2/2 0|20 = 200
2. -1∫1  (x4 + 3x) dx = (x5/5 + 3x2/2) -1|1 = (1/5 + 3/2) – (-1/5 + 3/2) = 0,4
3. 1∫2 ln x dx = 0∫ln 2 t d(et) = 0∫ln 2 tet dt = tet 0|ln 2 – 0∫ln 2 et dt = 2 ln 2 – 1 ≈ 0,38629436112
4. 0∫π/2 sin x dx = —cos x 0|π/2 = 0 – (-1) = 1
5. -5∫5 arcctgx dx = arcctg(-5)∫arcctg 5 t d(ctg t) = t ctg t arcctg(-5)|arcctg 5 – arcctg(-5)∫arcctg 5 ctg t dt = 15,707963267948967 – ln|sin t| arcctg(-5)|arcctg 5 ≈ 15,70796326795
6. 0∫3 ch x dx = sh x 0|3 = (e3 – e-3)/2 – (1 – 1)/2 ≈ 10,0178749274
7. -1∫1 e-x dx = —-1∫1 e-x d(-x) = —е-х -1|1 = -1/e – (-e) ≈ 2,3504023872876
8. 1∫2 xx dx ≈ 2,050446234534731 (не берётся; вычислен с помощью Интернета).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.2. Результаты приближённых ручных вычислений.

3.2.1. Формула прямоугольников.

3.2.1.1. 0∫20 x dx

3.2.1.1.1. m = 1 → 0∫20 x dx ≈ (20-0)\*10 = 200

3.2.1.1.2. m = 10 → h = (20-0)/10 = 2 → 0∫20 x dx ≈ 2\*(1+3+...+19) = 200

3.2.1.2. -1∫1  (x4 + 3x) dx

3.2.1.2.1. m = 1 → -1∫1  (x4 + 3x) dx ≈ (1-(-1))\*0 = 0

3.2.1.2.2. m = 10 → h = (1-(-1))/10 = 0,2 → -1∫1  (x4 + 3x) dx ≈ 0,2 \* (-2,0439 +(-1,8599) + (-1,4375) + (-0,8919) + (-0,2999) + 0,3001 + 0,9081 + 1,5625 + 2,3401 + 3,3561) = 0,2\*1,9338 = 0,38676

3.2.1.3. 1∫2 ln x dx

3.2.1.3.1. m = 1 → 1∫2 ln x dx ≈ (2-1)\*ln 1,5 ≈ 0,4054651081

3.2.1.3.2. m = 10 → h = (2-1)/10 = 0,1 → 1∫2 ln x dx ≈ 0,1 \* (ln 1,05 + ln 1,15 + … + ln 1,95) ≈ 0,3865024825

3.2.1.4. 0∫π/2 sin x dx

3.2.1.4.1. m = 1 → 0∫π/2 sin x dx ≈ (π/2-0)\*sin π/4 = π/2√2 ≈ 1,1107207

3.2.1.4.2. m = 10 → h = (π/2-0)/10 = π/20 → 0∫π/2 sin x dx ≈ π/20 \* (sin π/40 + sin 3π/40 + … + sin 19π/40) ≈ 1,00102882

3.2.1.5. -5∫5 arcctgx dx

3.2.1.5.1. m = 1 → -5∫5 arcctgx dx ≈ (5-(-5))\*arcctg 0 = 5π ≈ 15,707963268

3.2.1.5.2. m = 10 → h = (5-(-5))/10 = 1 → -5∫5 arcctgx dx ≈ 1\*(arcctg (-4,5) + arcctg (-3,5) + … + arcctg 4,5) ≈ 15,707963268

3.2.1.6. 0∫3 ch x dx

3.2.1.6.1. m = 1 → 0∫3 ch x dx ≈ (3-0)\*ch 1,5 ≈ 7,057228846

3.2.1.6.2. m = 10 → h = (3-0)/10 = 0,3 → 0∫3 ch x dx ≈ 0,3 \* (ch 0,15 + ch 0,45 + … + ch 2,85) ≈ 9,980406276

3.2.1.7. -1∫1 e-x dx

3.2.1.7.1. m = 1 → -1∫1 e-x dx ≈ (1-(-1))\*e0 = 2

3.2.1.7.2. m = 10 → h = (1-(-1))/10 = 0,2 → -1∫1 e-x dx ≈ 0,2\*(e-0,9+ e-0,7 + … + e0,9) ≈ 2,346489615

3.2.1.8. 1∫2 xx dx

3.2.1.8.1. m = 1 → 1∫2 xx dx ≈ (2-1)\*1,51,5 ≈ 1,837117307

3.2.1.8.2. m = 10→ h = (2-1)/10 = 0,1 → 1∫2 xx dx ≈ 0,1\*(1,051,05 + 1,151,15 + … + 1,951,95) 2,0480440927

\_ \_ \_ \_

3.2.2. Формула трапеций

3.2.2.1. 0∫20 x dx

3.2.2.1.1. m = 1 → 0∫20 x dx ≈ (20-0)/2\*(0+20) = 200

3.2.2.1.2. m = 10 → h = (20-0)/10 = 2 → 0∫20 x dx ≈ 2/2\*(0+2\*(2+4...+18)+20) = 200

3.2.2.2. -1∫1  (x4 + 3x) dx

3.2.2.2.1. m = 1 → -1∫1  (x4 + 3x) dx ≈ (1-(-1))/2\*(-2+4) = 0

3.2.2.2.2. m = 10 → h = (1-(-1))/10 = 0,2 → -1∫1  (x4 + 3x) dx ≈ 0,2/2\*(-2+2\*(-1,9904-1,6704+...+2,8096)+4) = 0,42656

3.2.2.3. 1∫2 ln x dx

3.2.2.3.1. m = 1 → 1∫2 ln x dx ≈ (2-1)/2\*(ln 2 + ln 1) = ln 2 / 2 ≈ 0,34657359

3.2.2.3.2. m = 10 → h = (2-1)/10 = 0,1 → 1∫2 ln x dx ≈ 0,1/2 \* (ln 1 + 2 \* (ln 1,1 + … + ln 1,9) + ln 2) ≈ 0,3858779367

3.2.2.4. 0∫π/2 sin x dx

3.2.2.4.1. m = 1 → 0∫π/2 sin x dx ≈ (π/2-0)/2 \* (sin π/2 + sin 0) = π/4 ≈ 0,7853981634

3.2.2.4.2. m = 10 → h = (π/2-0)/10 = π/20 → 0∫π/2 sin x dx ≈ π/40 \* (sin 0 + 2\*(sin π/20 + sin π/10 + … + sin 9π/20) + sin π/2) ≈ 0,99794299

3.2.2.5. -5∫5 arcctgx dx

3.2.2.5.1. m = 1 → -5∫5 arcctgx dx ≈ (5-(-5))/2\*(arcctg (-5) + arcctg 5) ≈ 15,707963268

3.2.2.5.2. m = 10 → h = (5-(-5))/10 = 1 → -5∫5 arcctgx dx ≈ 1/2\*(arcctg (-5) + 2\*(arcctg (-4) + … + arcctg 4) + arcctg 5) ≈ 15,707963268

3.2.2.6. 0∫3 ch x dx

3.2.2.6.1. m = 1 → 0∫3 ch x dx ≈ (3-0)/2\*(ch 0 + ch 3) ≈ 16,60149299

3.2.2.6.2. m = 10 → h = (3-0)/10 = 0,3 → 0∫3 ch x dx ≈ 0,15 \* (ch 0 + 2\*(ch 0,3 +…+ ch 2,7) + ch 3) ≈ 10,0928965

3.2.2.7. -1∫1 e-x dx

3.2.2.7.1. m = 1 → -1∫1 e-x dx ≈ (1-(-1))/2\*(e-1 + e) ≈ 3,08616127

3.2.2.7.2. m = 10 → h = (1-(-1))/10 = 0,2 → -1∫1 e-x dx ≈ 0,1\*(e-1 + 2\*(e-0,8 + … + e0,8) + e) ≈ 2,35823184

3.2.2.8. 1∫2 xx dx

3.2.2.8.1. m = 1 → 1∫2 xx dx ≈ (2-1)/2\*(1+4) = 2,5

3.2.2.8.2. m = 10→ h = (2-1)/10 = 0,1 → 1∫2 xx dx ≈ 0,05\*(1+2\*(1,11,1+1,21,2+…+1,91,9)+4) ≈ 2,05525318

\_ \_ \_ \_

3.2.3. Формула Симпсона

3.2.3.1. 0∫20 x dx

3.2.3.1.1. m = 1 → 0∫20 x dx ≈ (20-0)/6\*(0+4\*10+20) = 200

3.2.3.1.2. m = 10 → h = (20-0)/10 = 2 → 0∫20 x dx ≈ 2/6\*(0+2\*(2+4...+18)+4\*(1+3+...+19)+20) = 200

3.2.3.2. -1∫1  (x4 + 3x) dx

3.2.3.2.1. m = 1 → -1∫1  (x4 + 3x) dx ≈ (1-(-1))/6\*(-2+4\*0+4) = 2/3 = 0,(6)

3.2.3.2.2. m = 10 → h = (1-(-1))/10 = 0,2 → -1∫1  (x4 + 3x) dx ≈ 0,2/6\*(-2+2\*(-1,9904-1,6704+...+2,8096)+4\*(-2,0439+...+3,3561)+4) = 0,40002(6)

3.2.3.3. 1∫2 ln x dx

3.2.3.3.1. m = 1 → 1∫2 ln x dx ≈ (2-1)/6\*(ln 2 + 4\*ln 1,5 + ln 1) ≈ 0,3858346022

3.2.3.3.2. m = 10 → h = (2-1)/10 = 0,1 → 1∫2 ln x dx ≈ 0,1/6 \* (ln 1 + 2 \* (ln 1,1 + … + ln 1,9) + 4 \* (ln 1,05 + ln 1,15 + … + 1,95) + ln 2) ≈ 0,3862943006

3.2.3.4. 0∫π/2 sin x dx

3.2.3.4.1. m = 1 → 0∫π/2 sin x dx ≈ (π/2-0)/6 \* (sin π/2 + 4\*sin π/4 + sin 0) = π/12 \* (1+2√2) ≈ 1,00227987749

3.2.3.4.2. m = 10 → h = (π/2-0)/10 = π/20 → 0∫π/2 sin x dx ≈ π/120 \* (sin 0 + 2\*(sin π/20 + sin π/10 + … + sin 9π/20) + 4\*(sin π/40 + sin 3π/40 + … + sin 19π/40) + sin π/2) ≈ 1,0000002115466

3.2.3.5. -5∫5 arcctgx dx

3.2.3.5.1. m = 1 → -5∫5 arcctgx dx ≈ (5-(-5))/6\*(arcctg (-5) + 4\*arcctg 0 + arcctg 5) ≈ 15,707963268

3.2.3.5.2. m = 10 → h = (5-(-5))/10 = 1 → -5∫5 arcctgx dx ≈ 1/6\*(arcctg (-5) + 2\*(arcctg (-4) + … + arcctg 4) + 4\*(arcctg (-4,5) + arcctg (-3,5) + … + arcctg 4,5) + arcctg 5) ≈ 15,707963268

3.2.3.6. 0∫3 ch x dx

3.2.3.6.1. m = 1 → 0∫3 ch x dx ≈ (3-0)/6\*(ch 0 + 4\*ch 1,5 + ch 3) ≈ 10,2386502284

3.2.3.6.2. m = 10 → h = (3-0)/10 = 0,3 → 0∫3 ch x dx ≈ 0,05 \* (ch 0 + 2 \* (ch 0,3 +…+ ch 2,7) + 4 \* (ch 0,15 + ch 0,45 + … + ch 2,85) + ch 3) ≈ 10,0179030274

3.2.3.7. -1∫1 e-x dx

3.2.3.7.1. m = 1 → -1∫1 e-x dx ≈ (1-(-1))/6\*(e-1 + 4\*e0 + e) ≈ 2,3620537565435

3.2.3.7.2. m = 10 → h = (1-(-1))/10 = 0,2 → -1∫1 e-x dx ≈ 0,2/6\*(e-1 + 2\*(e-0,8 + … + e0,8) + 4\*(e-0,9 + … + e0,9) + e) ≈ 2,35040369151384

3.2.3.8. 1∫2 xx dx

3.2.3.8.1. m = 1 → 1∫2 xx dx ≈ (2-1)/6\*(1+4\*1,51,5+4) ≈ 2,0580782047

3.2.3.8.2. m = 10→ h = (2-1)/10 = 0,1 → 1∫2 xx dx ≈ 0,1/6 \* (1 + 2 \* (1,11,1 + 1,21,2 + … + 1,91,9) + 4 \* (1,051,05 +1,151,15+…+1,951,95) + 4) ≈ 2,050447121

3.3. Погрешности вычислений

3.3.1. Формула прямоугольников

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Инте-грал / n | 0∫20 x dx | -1∫1  (x4 + 3x) dx | 1∫2 ln x dx | 0∫π/2 sin x dx | -5∫5 arcctgx dx | 0∫3 ch x dx | -1∫1 e-x dx | 1∫2 xx dx |
| 1 | 0 | 0,4 | 0,0191707 | 0,1107207 | 0 | 2,96064608 | 0,3504024 | 0,2133289 |
| 10 | 0 | 0,01334 | 0,0002081 | 0,0010288 | 0 | 0,03746865 | 0,0039128 | 0,0024021 |

3.3.2. Формула трапеций

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Инте-грал / n | 0∫20 x dx | -1∫1  (x4 + 3x) dx | 1∫2 ln x dx | 0∫π/2 sin x dx | -5∫5 arcctgx dx | 0∫3 ch x dx | -1∫1 e-x dx | 1∫2 xx dx |
| 1 | 0 | 0,4 | 0,0397207 | 0,2146018 | 0 | 6,58361806 | 0,7357589 | 0,4495538 |
| 10 | 0 | 0,02656 | 0,0004164 | 0,0020570 | 0 | 0,07502157 | 0,0078295 | 0,0048069 |

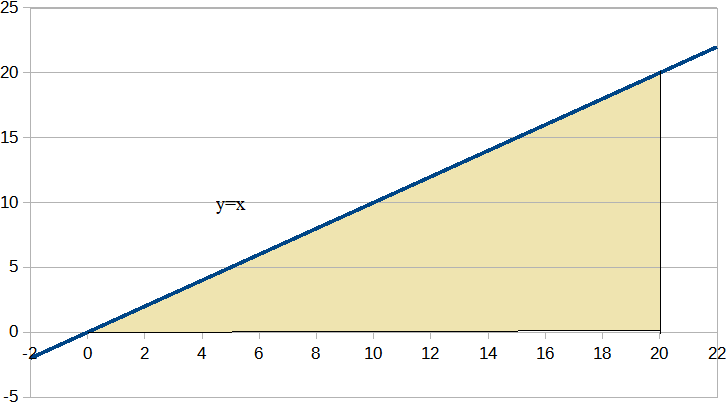
3.3.3. Формула Симпсона

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Инте-грал / n | 0∫20 x dx | -1∫1  (x4 + 3x) dx | 1∫2 ln x dx | 0∫π/2 sin x dx | -5∫5 arcctgx dx | 0∫3 ch x dx | -1∫1 e-x dx | 1∫2 xx dx |
| 1 | 0 | 0,2(6) | 4,598\*10-4 | 0,0022799 | 0 | 0,2207753 | 0,01165137 | 0,00763197 |
| 10 | 0 | 2,(6)\*10-5 | 6,052\*10-8 | 2,1155\*10-7 | 0 | 2,81\*10-5 | 1,3042\*10-6 | 8,8647\*10-7 |

3.4. Графики

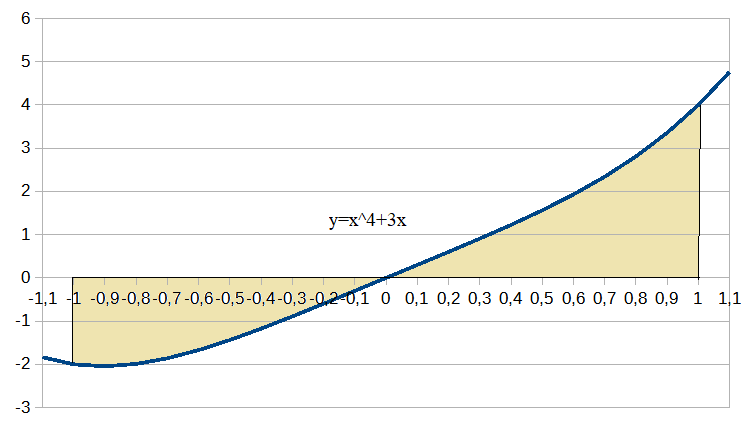
Здесь представлены графики функций и квадратур со значениями m=1, m=10. Значения m=100 или больше значительно приближают график к точному (приведён для каждой функции), так что можно считать их примерно совпадающими начиная с некоторого m.

3.4.1. 0∫20 x dx



Точный

|  |  |  |  |
| --- | --- | --- | --- |
| n\формула | прямоугольники | трапеции | Симпсон |
| 1 |  |  |  |
| 10 |  |  |  |

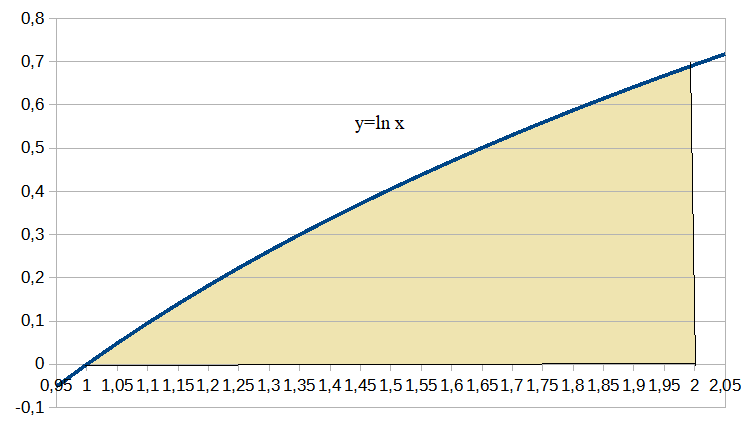


3.4.2. -1∫1  (x4 + 3x) dx

Точный

|  |  |  |  |
| --- | --- | --- | --- |
| n\формула | прямоугольники | трапеции | Симпсон |
| 1 |  |  |  |
| 10 |  |  |  |

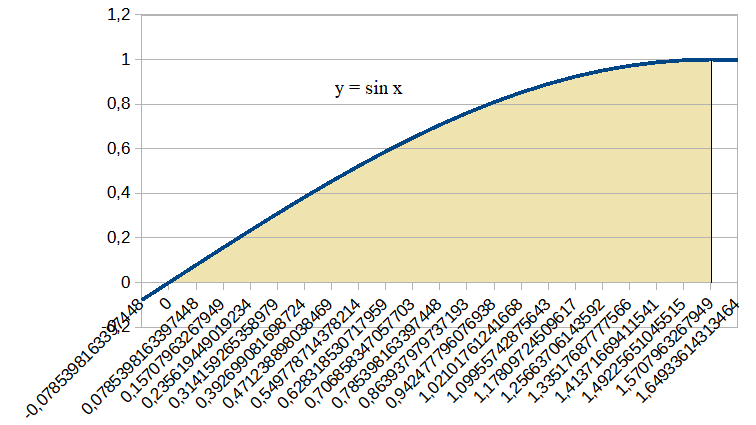
3.4.3. 1∫2 ln x dx



Точный

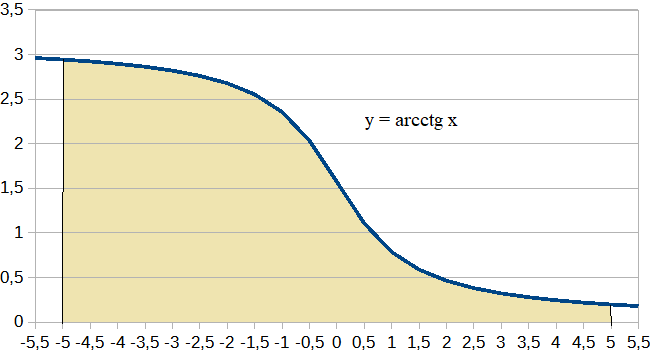
|  |  |  |  |
| --- | --- | --- | --- |
| n\формула | прямоугольники | трапеции | Симпсон |
| 1 |  |  |  |
| 10 |  |  |  |

3.4.4. 0∫π/2 sin x dx



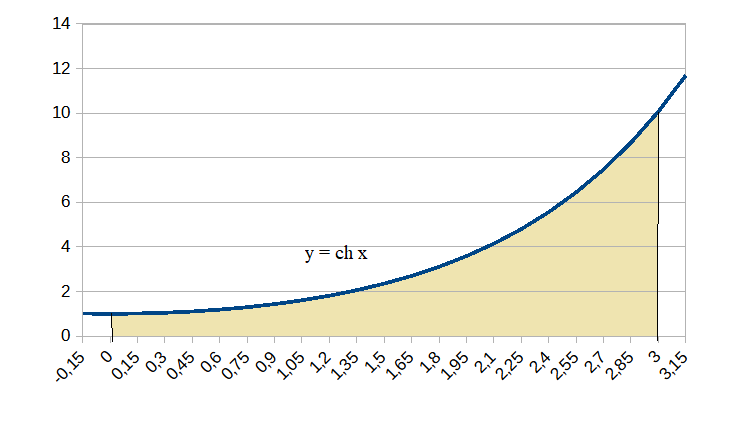
|  |  |  |  |
| --- | --- | --- | --- |
| n\формула | прямоугольники | трапеции | Симпсон |
| 1 |  |  |  |
| 10 |  |  |  |

3.4.5. -5∫5 arcctg x dx



Точный

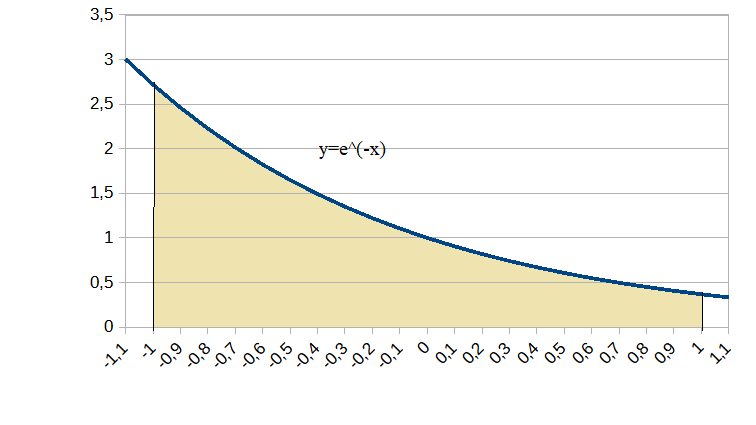
|  |  |  |  |
| --- | --- | --- | --- |
| n\формула | прямоугольники | трапеции | Симпсон |
| 1 |  |  |  |
| 10 |  |  |  |



3.4.6. 0∫3 ch x dx

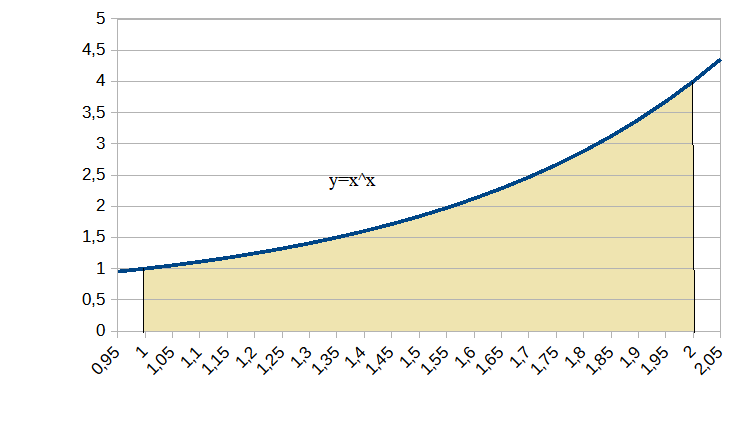
Точный

|  |  |  |  |
| --- | --- | --- | --- |
| n\формула | прямоугольники | трапеции | Симпсон |
| 1 |  |  |  |
| 10 |  |  |  |

3.4.7. -1∫1 e-x dx

Точный

|  |  |  |  |
| --- | --- | --- | --- |
| n\формула | прямоугольники | трапеции | Симпсон |
| 1 |  |  |  |
| 10 |  |  |  |

3.4.8. 1∫2 xx dx

Точный

|  |  |  |  |
| --- | --- | --- | --- |
| n\формула | прямоугольники | трапеции | Симпсон |
| 1 |  |  |  |
| 10 |  |  |  |

**4. Листинг кода и результатов работы программы**

4.1. Текст программы (на языке Python 3.9)

import math

from math import \*

from tkinter import \*

from tkinter import scrolledtext

from tkinter import messagebox

from tkinter import filedialog as fd

from tkinter.ttk import Checkbutton

import matplotlib.pyplot as plt

import numpy as np

# преобразование текстового выражения в математическое

def eval\_expression(s, x):

allowed\_names = {"x": x, "п": math.pi, "pi": math.pi,

"е": math.e, "e": math.e, "sqrt": sqrt,

"ln": log, "lg": log10, "log": log,

"sin": sin, "cos": cos, "tg": tan, # ctg = 1 / tg

"sh": sinh, "ch": cosh, "th": tanh, # cth = 1 / th

"arcsin": asin, "arccos": acos, "arctg": atan, # arcctg = pi / 2 - arctg

"arsh": asinh, "arch": acosh, "arth": atanh}

code = compile(s, "<string>", "eval")

for name in code.co\_names:

if name not in allowed\_names:

messagebox.showerror('Внимание!', 'Ошибка при вводе данных!')

return eval(code, {"\_\_builtins\_\_": {}}, allowed\_names)

# вставка функции из файла

def insert\_expression():

try:

file\_name = fd.askopenfilename()

f = open(file\_name)

s = f.read()

txt\_int.insert(1.0, s)

f.close()

except Exception:

messagebox.showerror("Внимание!", "Ошибка при загрузке файла!")

# формула прямоугольников

def rectangles(a, b, m, h):

s = txt\_int.get()

try:

# значения для построения графика

xs = [a]

j = a + h

while j < b:

xs += [j] \* 2

j += h

xs += [b]

if m == 1:

integral = (b-a) \* eval\_expression(s, (a+b)/2)

ys = [eval\_expression(s, (a+b)/2)] \* 2

return [integral, xs, ys]

elif m in range(2, 10001):

integral = 0

ys = []

x = a + h/2

while x < b:

y = eval\_expression(s, x)

integral += y

ys += [y] \* 2

x += h

integral \*= h

return [integral, xs, ys]

except Exception:

messagebox.showerror('Внимание!', 'Ошибка при вычислении!')

# формула трапеций

def trapecies(a, b, m, h):

s = txt\_int.get()

try:

# значения для построения графика

xs = [a]

j = a + h

while j < b:

xs += [j] \* 2

j += h

xs += [b]

if m == 1:

fa = eval\_expression(s, a)

fb = eval\_expression(s, b)

integral = (b-a) \* (fa + fb) / 2

ys = [fa, fb]

return [integral, xs, ys]

elif m in range (2, 10001):

y = eval\_expression(s, a)

integral = y

ys = [y]

x = a + h

while x < b:

y = eval\_expression(s, x)

integral += 2\*y

ys += [y] \* 2

x += h

y = eval\_expression(s, b)

integral += y

ys += [y]

integral \*= (h/2)

return [integral, xs, ys]

except Exception:

messagebox.showerror('Внимание!', 'Ошибка при вычислении!')

# формула Симпсона

def simpson(a, b, m, h):

s = txt\_int.get()

h /= 2 # для удобства разбиваем шаг деления ещё надвое

try:

if m == 1:

# значения для построения графика

xs = []

j = a

while j < b:

xs += [(j + h\*i/10) for i in range (0, 21)]

j += 2\*h

y0 = eval\_expression(s, a)

y1 = eval\_expression(s, (a+b)/2)

y2 = eval\_expression(s, b)

integral = (b-a) / 6 \* (y0 + 4\*y1 + y2)

alpha = (y0 - 2\*y1 + y2) / (2 \* h\*\*2)

beta = (y2 - y0 - 4\*alpha\*((a+b)/2)\*h) / (2 \* h)

gamma = y1 - alpha\*((a+b)/2)\*\*2 - beta\*(a+b)/2

ys = [(alpha\*i\*\*2 + beta\*i + gamma) for i in xs]

return [integral, xs, ys]

elif m in range (2, 10001):

integral = 0

xs = []

ys = []

x = a + h

while x < b:

y0 = eval\_expression(s, x - h)

y1 = eval\_expression(s, x)

y2 = eval\_expression(s, x + h)

integral += (y0 + 4\*y1 + y2)

alpha = (y0 - 2\*y1 + y2) / (2 \* h\*\*2)

beta = (y2 - y0 - 4\*alpha\*x\*h) / (2 \* h)

gamma = y1 - alpha\*x\*\*2 - beta\*x

xs1 = [(x + h\*i/10) for i in range(-10, 11)]

ys += [(alpha\*i\*\*2 + beta\*i + gamma) for i in xs1]

x += 2\*h

xs += xs1

integral \*= (b-a)/6

return [integral, xs, ys]

except Exception:

messagebox.showerror('Внимание!', 'Ошибка при вычислении!')

# порядок погрешности

def mistake(a, b, h):

variant = selected.get()

if variant == 1:

return (b-a) / 24 \* (h\*\*2)

elif variant == 2:

return (b-a) / 12 \* (h\*\*2)

elif variant == 3:

return (b-a) / 180 \* (h\*\*4)

def graphic(a, b, m, xs, ys): # построение графиков

s = txt\_int.get()

x = np.linspace(a, b, m\*100)

y = [eval\_expression(s, i) for i in x]

variant = selected.get()

if variant == 1:

plt.title("Квадратура прямоугольников")

elif variant == 2:

plt.title("Квадратура трапеций")

elif variant == 3:

plt.title("Квадратура Симпсона")

plt.plot(x, y, xs, ys)

plt.axis('square')

plt.grid(True)

plt.fill\_between(xs, 0, ys, color="r", alpha=0.3)

plt.show()

# кнопка выполнения действий

def clicked():

variant = selected.get()

txt1.delete(1.0, END)

txt2.delete(1.0, END)

try:

a = float(eval\_expression(lim\_down.get(), 0))

b = float(eval\_expression(lim\_up.get(), 0))

m = int(eval\_expression(spin.get(), 0))

h = (b - a) / m

if variant == 1:

res = rectangles(a, b, m, h)

result = '{:.10f}'.format(res[0])

elif variant == 2:

res = trapecies(a, b, m, h)

result = '{:.10f}'.format(res[0])

elif variant == 3:

res = simpson(a, b, m, h)

result = '{:.10f}'.format(res[0])

txt1.insert(INSERT, result)

txt2.insert(INSERT, '{:.10g}'.format(mistake(a, b, h)))

# передача данных для графика

xs = res[1]

ys = res[2]

if chk\_state:

graphic(a, b, m, xs, ys)

except Exception:

pass

# информация о программе

def info():

messagebox.showinfo('О программе', '''Данная программа является учебной и предназначена для приближённого вычисления определённых собственных интегралов тремя различными способами.

\nФункции реализованы с помощью языка Python 3.9 на основании методических указаний в задании и могут некорректно работать при попытке вычислить интеграл иного типа.

\nНе рекомендуется вводить выражения, не являющиеся математическими, в том числе команды для командной строки Windows и интерпретатора Python.

\nВ программе реализована защита от подобных выражений, однако создатель не несёт ответственности за некорректное использование приложения.''')

# справка для пользователя

def help():

messagebox.showinfo('Помощь', '''Чтобы вычислить интеграл, необходимо:

\n1) выбрать формулу для подсчёта;

\n2) указать точность вычисления - число частей, на которые разбивается отрезок;

\n3) нажать кнопку "Строить график" для вывода окна с графическим сравнением функции и выбранной квадратуры;

\n4) ввести функцию для интегрирования:

\n a) с клавиатуры;

\n б) нажать на кнопку и выбрать текстовый файл (\*.txt, \*.doc) для чтения;

\n5) ввести пределы интегрирования (числа или выражения);

\n6) нажать на кнопку "Вычислить".\n

\nО точности вычисления - см. в разделе "Формулы".

\nНе вводите выражения, не являющиеся математически! Это может быть небезопасны для работы программы и компьютера.

\nИспользуйте "\*\*" в качестве символа возведения в степень.

\nПредельно доступное число шагов разбиения - 10000.''')

# список формул

def formuls():

messagebox.showinfo('Формулы', '''a, b - пределы интегрирования

\nm - количество частей отрезка интегрирования

\nh = (b-a) / m - шаг интегрирования\n

\nКвадратура прямоугольников:

\nm = 1 ∫ f(x) dx ~ (b-a) \* f((a+b)/2)

\nm > 1 ∫ f(x) dx ~ h \* (f(a+h/2) + f(a+3h/2) + ... + f(a+(2m-1)h/2))\n

\nКвадратура трапеций:

\nm = 1 ∫ f(x) dx ~ (b-a)/2 \* (f(a) + f(b))

\nm > 1 ∫ f(x) dx ~ h/2 \* (f(a) + 2\*f(a+h) + ... + 2\*f(b-h) + f(b))\n

\nКвадратура Симпсона:

\nm = 1 ∫ f(x) dx ~ (b-a)/6 \* (f(a) + 4\*f((a+b)/2) + f(b))

\nm > 1 ∫ f(x) dx ~ h/6 \* (f(a) + f(b) + 2\*(f(a+h)+...+f(b-h)) + 4\*(f(a+h/2)+...+f(b-h/2)))\n

\nПорядок погрешности - выражение, позволяющее оценить возможное отклонение значения от точного. В данном приложении равен:

\nпрямоугольники: (b-a) / 24 \* h^2

\nтрапеции: (b-a) / 12 \* h^2

\nСимпсон: (b-a) / 180 \* h^4\n

\nТочное значение погрешности можно получить, если домножить порядок на max|f''(x)| для формул прямоугольников и трапеций или на max|f(IV) (x)| для квадратуры Симпсона. Значение x при этом лежит на отрезке [a;b].''')

# выход

def away():

exit()

# графический интерфейс

window = Tk()

window.title("Определённые интегралы")

window.geometry('650x350')

lbl1 = Label(window, text="Формула для вычисления:", font=("Calibri", 11))

lbl1.grid(column=0, row=1, columnspan=4)

selected = IntVar()

rad1 = Radiobutton(window, text='прямоугольников', font=("Calibri", 11), value=1, variable=selected)

rad2 = Radiobutton(window, text='трапеций', font=("Calibri", 11), value=2, variable=selected)

rad3 = Radiobutton(window, text='Симпсона', font=("Calibri", 11), value=3, variable=selected)

rad1.grid(column=1, row=2)

rad2.grid(column=2, row=2)

rad3.grid(column=3, row=2)

lbl2 = Label(window, text=". . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .")

lbl2.grid(column=0, row=3, columnspan=4)

lbl3 = Label(window, text="Разбиение на ", font=("Calibri", 11))

lbl3.grid(column=1, row=4)

var = IntVar()

var.set(1)

spin = Spinbox(window, from\_=1, to=10000, width=5, textvariable=var)

spin.grid(column=2, row=4)

lbl4 = Label(window, text="частей", font=("Calibri", 11))

lbl4.grid(column=3, row=4)

chk\_state = BooleanVar()

chk\_state.set(True)

chk = Checkbutton(window, text='Строить график', var=chk\_state)

chk.grid(column=0, row=4)

lbl5 = Label(window, text="(При разбиении по Симпсону отрезок автоматически делится на 2m частей.)", font=("Calibri", 8))

lbl5.grid(column=1, row=5, columnspan=4)

lbl6 = Label(window, text=". . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .")

lbl6.grid(column=0, row=6, columnspan=4)

lbl7 = Label(window, text="Интеграл:", font=("Calibri", 13))

lbl7.grid(column=0, row=7, columnspan=4)

btn = Button(window, text="Вычислить", command=clicked)

btn.grid(column=3, row=7)

lbl8 = Label(window, text="∫", font=("Calibri", 13))

lbl8.grid(column=0, row=9)

lbl9 = Label(window, text=" dx = ", font=("Calibri", 13))

lbl9.grid(column=2, row=9)

lim\_up = Entry(window, width=5)

lim\_up.grid(column=0, row=8, padx=5)

lim\_down = Entry(window, width=5)

lim\_down.grid(column=0, row=10, padx=5)

txt\_int = Entry(window,width=40)

txt\_int.grid(column=1, row=9)

txt1 = scrolledtext.ScrolledText(window, width=20, height=1)

txt1.grid(column=3, row=9)

filename = Button(window, text="Выбрать файл", command=insert\_expression)

filename.grid(column=0, row=11)

lbl10 = Label(window, text="Порядок погрешности:", font=("Calibri", 12))

lbl10.grid(column=1, row=11, columnspan=2)

txt2 = scrolledtext.ScrolledText(window, width=20, height=1)

txt2.grid(column=3, row=11)

menu = Menu(window)

new\_item = Menu(menu, tearoff=0)

new\_item.add\_command(label="О программе", command=info)

new\_item.add\_command(label="Помощь", command=help)

new\_item.add\_command(label="Формулы", command=formuls)

new\_item.add\_separator()

new\_item.add\_command(label="Выход", command=away)

menu.add\_cascade(label='Справка', menu=new\_item)

window.config(menu=menu)

window.mainloop()

4.2. Значения, вычисленные с помощью программы.

4.2.1. Формула прямоугольников

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Интеграл \ n** | **1** | **10** | **100** | **1000** | **10000** |
| 0∫20 x dx | 200 | 200 | 200 | 200 | 200 |
| -1∫1  (x4 + 3x) dx | 0 | 0,38676 | 0,399866676 | 0,3999986667 | 0,3999999867 |
| 1∫2 ln x dx | 0,4054651081 | 0,3865024825 | 0,3862964444 | 0,386294382 | 0,3862943613 |
| 0∫π/2 sin x dx | 1,110720735 | 1,001028824 | 1,000010281 | 1,000000103 | 1,000000001 |
| -5∫5 arcctg x dx | 15,70796327 | 15,70796327 | 15,70796327 | 15,70796327 | 15,70796327 |
| 0∫3 ch x dx | 7,057228846 | 9,980406276 | 10,01749927 | 10,01787117 | 10,01787489 |
| -1∫1 e-x dx | 2 | 2,346489615 | 2,350363214 | 2,350401996 | 2,350402383 |
| 1∫2 xx dx | 1,837117307 | 2,048044093 | 2,050422182 | 2,050445994 | 2,050446232 |

4.2.2. Формула трапеций

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Интеграл \ n** | **1** | **10** | **100** | **1000** | **10000** |
| 0∫20 x dx | 200 | 200 | 200 | 200 | 200 |
| -1∫1  (x4 + 3x) dx | 2 | 0,42656 | 0,400266656 | 0,4000026667 | 0,4000000267 |
| 1∫2 ln x dx | 0,3465735903 | 0,3858779367 | 0,3862901945 | 0,3869874666 | 0,3863636754 |
| 0∫π/2 sin x dx | 0,7853981634 | 0,9979429864 | 1,015687402 | 1,001570591 | 0,9999999979 |
| -5∫5 arcctg x dx | 15,70796327 | 15,70796327 | 15,72770282 | 15,70993722 | 15,70796327 |
| 0∫3 ch x dx | 16,60149299 | 13,11319513 | 10,32065612 | 10,01788244 | 10,017875 |
| -1∫1 e-x dx | 3,08616127 | 2,358231844 | 2,350480734 | 2,350403171 | 2,350475971 |
| 1∫2 xx dx | 2,5 | 2,055253178 | 2,050494339 | 2,050446716 | 2,050446239 |

4.2.3. Формула Симпсона

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Интеграл \ n** | **1** | **10** | **100** | **1000** | **10000** |
| 0∫20 x dx | 200 | 200 | 200 | 200 | 200 |
| -1∫1  (x4 + 3x) dx | 0,6666666667 | 0,4000266667 | 0,4000000027 | 0,4 | 0,4 |
| 1∫2 ln x dx | 0,3858346022 | 0,3862943006 | 0,3862943611 | 0,3869873416 | 0,3863636742 |
| 0∫π/2 sin x dx | 1,002279877 | 1,000000212 | 1,01570764 | 1,001570796 | 1 |
| -5∫5 arcctg x dx | 15,70796327 | 15,70796327 | 15,70796327 | 15,70796327 | 15,70796327 |
| 0∫3 ch x dx | 10,23865023 | 12,75923184 | 10,31692197 | 10,01787493 | 10,01787493 |
| -1∫1 e-x dx | 2,362053757 | 2,350403692 | 2,350402387 | 2,350402387 | 2,350402387 |
| 1∫2 xx dx | 2,058078205 | 2,050447121 | 2,050446235 | 2,050443978 | 2,050446212 |

4.3. Расчётные максимальные значения погрешностей

4.3.1. Формула прямоугольника

R <= |(b-a)3\*f’’(n)/24m2|, a<=n<=b, |f’’(n)| → max

x’’ = 0

(x4+3x)’’ = 12x2; max = 12

(ln x)’’ = -1/x2; |max| = 1

(sin x)’’ = -sin x; |max| = 1

(arcctg x)’’ = 2x/(x2+1)2; |max| = 3(√3)/8 ≈ 0,649519

(ch x)’’ = ch x; max ≈ 10,067661996

(e-x)’’ = e-x; max = e

(xx)’’ = xx(ln x +1)2 + xx-1; max ≈ 13,4669895

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Интеграл \ n** | **1** | **10** | **100** | **1000** | **10000** |
| 0∫20 x dx | 0 | 0 | 0 | 0 | 0 |
| -1∫1  (x4 + 3x) dx | 4 | 0,04 | 0,0004 | 0,000004 | 0,00000004 |
| 1∫2 ln x dx | 0,041666667 | 0,000416667 | 0,000004167 | 0,000000042 | 0,0000000004 |
| 0∫π/2 sin x dx | 0,161491024 | 0,001614910 | 0,000016149 | 0,000000161 | 0,0000000016 |
| -5∫5 arcctg x dx | 27,063293868 | 0,270632939 | 0,002706329 | 0,000027063 | 0,0000002706 |
| 0∫3 ch x dx | 11,326119746 | 0,113261197 | 0,001132612 | 0,000011326 | 0,0000001133 |
| -1∫1 e-x dx | 0,906093943 | 0,009060939 | 0,000090609 | 0,000000906 | 0,0000000091 |
| 1∫2 xx dx | 0,561124563 | 0,005611246 | 0,000056112 | 0,000000561 | 0,0000000056 |

4.3.2. Формула трапеций

R <= |(b-a)3\*f’’(n)/12m2|, a<=n<=b, |f’’(n)| → max

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Интеграл \ n** | **1** | **10** | **100** | **1000** | **10000** |
| 0∫20 x dx | 0 | 0 | 0 | 0 | 0 |
| -1∫1  (x4 + 3x) dx | 8 | 0,08 | 0 | 0 | 0 |
| 1∫2 ln x dx | 0,083333333 | 0,000833333 | 0,000008333 | 0,000000083 | 0,0000000008 |
| 0∫π/2 sin x dx | 0,322982049 | 0,003229820 | 0,000032298 | 0,000000323 | 0,000000003 |
| -5∫5 arcctg x dx | 54,126587737 | 0,541265877 | 0,005412659 | 0,000054127 | 0,000000541 |
| 0∫3 ch x dx | 22,652239491 | 0,226522395 | 0,002265224 | 0,000022652 | 0,000000227 |
| -1∫1 e-x dx | 1,812187886 | 0,018121879 | 0,000181219 | 0,000001812 | 0,000000018 |
| 1∫2 xx dx | 1,122249125 | 0,011222491 | 0,000112225 | 0,000001122 | 0,000000011 |

4.3.3. Формула Симпсона

R <= |((b-a)/2)5\*fIV (n)/90m4|, a<=n<=b, |f[IV](n)| → max

(x)IV = 0

(x4+3x)IV = 24

(ln x)IV = -6/x4; |max| = 6

(sin x)IV = sin x; |max| = 1

(arcctg x)IV = 24x(x2-1)/(x2+1)4; |max| ≈ 4,668559284

(ch x)IV = ch x; max ≈ 10,067661996

(e-x)IV = e-x; max = e

(xx)IV = xx(ln x +1)2(ln x (ln x +2) +1) + xx-1(5ln x (ln x +2) + (ln x +1)2)– xx-2(4ln x +1) + 2xx-3; max ≈ 54,5013418273685

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Интеграл \ n** | **1** | **10** | **100** | **1000** | **10000** |
| 0∫20 x dx | 0 | 0 | 0 | 0 | 0 |
| -1∫1  (x4 + 3x) dx | 0,2666666667 | 0,0000266667 | 2,666667\*10-9 | 2,666667\*10-13 | 2,666667\*10-17 |
| 1∫2 ln x dx | 0,0020833333 | 2,083333\*10-7 | 2,083333\*10-11 | 2,083333\*10-15 | 2,083333\*10-19 |
| 0∫π/2 sin x dx | 3,320526\*10-3 | 3,320526\*10-7 | 3,320526\*10-11 | 3,320526\*10-15 | 3,320526\*10-19 |
| -5∫5 arcctg x dx | 162,10275292 | 1,6210275\*10-2 | 1,6210275\*10-6 | 1,6210275\*10-10 | 1,6210275\*10-14 |
| 0∫3 ch x dx | 0,8494589809 | 8,4945898\*10-5 | 8,4945898\*10-9 | 8,4945898\*10-13 | 8,4945898\*10-17 |
| -1∫1 e-x dx | 3,0203131\*10-2 | 3,0203131\*10-6 | 3,0203131\*10-10 | 3,0203131\*10-14 | 3,0203131\*10-18 |
| 1∫2 xx dx | 1,8924077\*10-2 | 1,8924077\*10-6 | 1,8924077\*10-10 | 1,8924077\*10-14 | 1,8924077\*10-18 |

**5. Инструкция для пользователя**

5.1. О программе

Данная программа является учебной и предназначена для приближённого вычисления определённых собственных интегралов тремя различными способами.

Функции реализованы с помощью языка Python 3.9 на основании методических указаний в задании и могут некорректно работать при попытке вычислить интеграл иного типа.

Не рекомендуется вводить выражения, не являющиеся математическими, в том числе команды для командной строки Windows и интерпретатора Python. В программе реализована защита от подобных выражений, однако создатель не несёт ответственности за некорректное использование приложения.

5.2. Помощь

Чтобы вычислить интеграл, необходимо:

1. выбрать формулу для подсчёта;
2. указать точность вычисления — число частей, на которые разбивается отрезок;
3. нажать кнопку «Строить график» для вывода окна с графическим сравнением функции и выбранной квадратуры;
4. ввести функцию для интегрирования: a) с клавиатуры; б) нажать на кнопку и выбрать текстовый файл (\*.txt, \*.doc) для чтения;
5. ввести пределы интегрирования (числа или выражения);
6. нажать на кнопку "Вычислить".

О точности вычисления — см. в разделе "Формулы".

Не вводите выражения, не являющиеся математически! Это может быть небезопасны для работы программы и компьютера.

Используйте "\*\*" в качестве символа возведения в степень.

Предельно доступное число шагов разбиения — 10000.

(Исполняемый файл прилагается к отчёту).

**6. Выводы**

По проделанной работе, направленной на реализацию численного нахождения определённых интегралов программными методами, можно сделать следующие выводы:

* квадратурные формулы прямоугольников, трапеций и Симпсона применимы к собственным определённым интегралам от различных функций (линейных, степенных, трансцендентных);
* с увеличением количества частей, на которые разбивается отрезок интегрирования, возрастает точность вычисления;
* значения фактических погрешностей вычислений вручную не превосходят максимальных расчётных;
* при увеличении точности (количества частей, на которые разбивается отрезок) в 10 раз погрешность для квадратурных формул прямоугольников и трапеций уменьшается в среднем на 2 порядка, для квадратурной формулы Симпсона — на 4 порядка;
* графики квадратур при увеличении точности неограниченно приближаются к графикам исходных функций;
* реализация численного нахождения определённых интегралов программными методами возможна и успешно осуществима на практике;
* наименьший порядок погрешности, который возможно получить при разбиении отрезка на 10000 частей, равен ~10-19 (правда, большая точность существенно замедляет работу программы).

Таким образом, цель работы можно считать достигнутой.