**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Сибирский государственный университет науки и технологий**

**имени академика М.Ф. Решетнёва»**

ИИТК/09.03.01/\Информатика и вычислительная техника/Автоматизированные системы обработки информации и управления

институт/факультет/подразделение

Информатики и вычислительной техники

кафедра

**Определенные интегралы**

Преподаватель \_\_01.04.2022\_\_\_\_\_ Семенкина М.Е.

подпись, дата Фамилия, инициалы

Обучающиеся \_\_БИА21-02 \_ \_\_01.04.2022\_\_\_

номер группы подпись, дата Рау М.А.

Конопченко Л.Б.

Поворов В.С

Турчин П.В

Фамилия, инициалы

Красноярск 2022

Задачи:

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 в ручную)

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

Краткое описание выполнения задания:

1. Вводим параметры интегрирования.
2. Открываем файлы для записи результата и промежуточных значений вычислений
3. Записываем в файл параметры интегрирования
4. Объявляем массив из 8 функций которые нужно проинтегрировать:
   * y=sin(x)
   * y=cos(x)
   * y=2\*x²
   * y=x
   * y=5ᵡ
   * y=1/5²+x²
   * y=eᵡ
   * y=(x+2)³
5. Записываем время начала и завершения алгоритма. Перебирая в цикле массив функций, вызываем функции в соответствии с методом. Используемые методы:
   * Метод трапеций
   * Метод центральных прямоугольников.
   * Метод Симпсона

Записываем полученные результаты в data.tsv, intermediate.tsv. После завершения работы алгоритма, вызываем скрипт обработки данных report.py

Код программы:

С++:

#define \_USE\_MATH\_DEFINES

#include <math.h>

#include <ctime>

#include <fstream>

#include <iomanip>

#include <iostream>

#include "main.h"

using namespace std;

int main() {

setlocale(LC\_ALL, "");

ofstream write\_file("data.tsv");

ofstream write\_integral("intermediate.tsv");

double a, b;

double h;

h = 0.1; // Устанавливаем точность вычислений (чем меньше, тем лучше)

/\*

Запрашивашиваем границы в пределах ожидаемых значений, так как границы общие для всех восьми функций, некоторые интегралы могут не посчитаться,

или посчитаться, но выйти за максимальное или минималное значение long double от ±2.23 x 10^-308 до ±1.80 x 10^308

\*/

cout << "Введите границы интегрирования" << endl;

cout << "Нижняя (a): " << endl;

cin >> a;

cout << "Верхняя (b): " << endl;

cin >> b;

double n = (b - a) / h; // задаем число разбиений

const int f\_num = 8;

double (\*func[f\_num])(double) = {

sin,

cos,

function\_0,

function\_1,

function\_2,

function\_3,

function\_4,

function\_5

};

/\* Записываем в файл параметры интегрирования \*/

write\_file << a << '\t';

write\_file << b << '\t';

write\_file << h << '\n';

clock\_t start, end;

start = clock();

for (int i = 0; i < f\_num; i++) {

**write\_file << trapezoidalIntegral(a, b, h, n, func[i]) << '\t';**

**write\_file << centralrectanglesIntegral(a, b, h, n, func[i], write\_integral) << '\t';**

**write\_file << SimpsonIntegral(a, b, h, n, func[i]) << '\n';**

}

end = clock();

/\*

Выводим информацию о времени выполнения алгорима и о файле отчета

Закрываем потоки ввода файлов

Запускаем скрипт рендеринга отчета

\*/

cout << '|' << setw(10) << "Время интегрирования (с.)" << setw(2) << ':'

<< setw(10) << (end - start) / 1000.0 << setw(5) << '|' << endl << endl;

cout << setw(10) << "Отчет записан в" << setw(2) << ':' << setw(10) << " отчет.xlsx" << endl << endl;

write\_file.close();

write\_integral.close();

system("python report.py");

system("pause");

return 0;

}

/\* y=2\*x² \*/

double function\_0(double x) {

return (2.0 \* (x \* x));

}

/\* y=x \*/

double function\_1(double x) {

return x;

}

/\* y=5ᵡ \*/

double function\_2(double x) {

return pow(5.0, x);

}

/\* y=1/5²+x² \*/

double function\_3(double x) {

return (1 / (5.0 \* 5.0 + x \* x));

}

/\* y=eᵡ \*/

double function\_4(double x) {

return pow(M\_E, x);

}

/\* y=(x+2)³ \*/

double function\_5(double x) {

return ((x + 2.0) \* (x + 2.0) \* (x + 2.0));

}

/\* Метод трапеций \*/

long double trapezoidalIntegral(double a, double b, double h, double n, double(\*f)(double)) {

long double result = 0.0;

result = h \* (f(a) + f(b)) / 2.0;

for (int i = 1; i <= n - 1; i++) {

result = result + h \* f(a + h \* i);

}

return round(result\* 1000000)/ 1000000;

}

/\*Метод центральных прямоугольников

Записываем в файл intermediate.tsv промежуточные результаты вычислений интегралов, для построения графиков \*/

void centralrectanglesIntegral(double a, double b, double h, double n, double(\*f)(double), ofstream& write\_integral) {

long double result = 0.0;

for (int j = 1; j <= n; j++) {

result = result + h \* f(a + h \* (j - 0.5));

write\_integral << round(result \* 1000000) / 1000000 << '\t';

}

write\_integral << endl;

return round(result \* 1000000) / 1000000;

}

/\* Метод Симпсона \*/

void SimpsonIntegral(double a, double b, double h, double n, double(\*f)(double)) {

long double result = h \* (f(a) + f(b)) / 6.0;

for (int i = 1; i <= n; i++)

result = result + 4.0 / 6.0 \* h \* f(a + h \* (i - 0.5));

for (int i = 1; i <= n - 1; i++)

result = result + 2.0 / 6.0 \* h \* f(a + h \* i);

return round(result \* 1000000) / 1000000;

}

Python:

# Подключаем библеотеки для работы с excel и массивами данных

import xlsxwriter

import numpy as np

import os

import math

import pandas as pd

# Создаем файл xlsx с листами для данных и для графиков

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

worksheet = workbook.add\_worksheet('Данные')

worksheet2 = workbook.add\_worksheet('Графики')

# Считываем данные в массивы из файлов data.tsv и intermediate.tsv, удаляем nan элементы из данных intermediate.tsv

data = np.loadtxt("data.tsv", delimiter='\t', dtype=np.longdouble )

data\_integral = np.genfromtxt("intermediate.tsv", delimiter='\t', dtype=np.longdouble )

data\_integral[np.isnan(data\_integral)] = 0.0

data\_integral = [x[:-1].tolist() for x in data\_integral]

# Заполняем шапку и данные главной таблицы в виде массива

main\_table\_data = {'Интеграл от:': ['Метод трапеций', 'Метод центральных прямоугольников' , 'Метод Симпсона'],

'sin(x) dx': data[1],

'cos(x) dx': data[2],

'2\*x² dx': data[3],

'x dx': data[4],

'5ᵡ dx': data[5],

'1/5²+x² dx': data[6],

'eᵡ dx': data[7],

'(x+2)³ dx': data[8],

}

limits\_table\_data = {'Границы': ['Нижняя(a)', 'Верхняя(b)', 'Точность'],

'':data[0],

}

# Просчитываем значения функций и их первообразных в заданных границах интегрирования

y = []

F\_y=[]

x = []

for j in range(8):

y.append([])

F\_y.append([])

for i in np.arange(data[0][0], data[0][1], data[0][2]):

y[0].append(round(math.sin(round(i,1)), 6))

F\_y[0].append(round(-math.cos(round(i,1)),6))

y[1].append(round(math.cos(round(i,1)),6))

y[2].append(round(2\*(round(i,1)\*round(i,1)),6))

F\_y[2].append(round((2\*(round(i,1)\*\*3))/3,6))

y[3].append(round(i,1))

F\_y[3].append(round((round(i,1)\*round(i,1))/2,6))

y[4].append(round(5\*\*round(i,1),6))

F\_y[4].append(round((5\*\*round(i,1))/math.log(5),6))

y[5].append(round(1/(5\*5+round(i,1)\*round(i,1)),6))

F\_y[5].append(round(math.atan(round(i,1)/5)/5,6))

y[6].append(round(math.exp(round(i,1)),6))

y[7].append(round((round(i,1)+2)\*\*3,6))

F\_y[7].append(round(((round(i,1)+2)\*\*4)/4,6))

x.append(round(i,1))

F\_y[1]=y[0]

F\_y[6]=y[6]

# Заполняем шапку и данные таблицы построения графиков в виде массива

graph\_table = { 'x': x,

'y=sin(x)': y[0],

'sin(x) dx': data\_integral[0],

'F(sin(x))': F\_y[0],

'y=cos(x)': y[1],

'cos(x) dx': data\_integral[1],

'F(cos(x))': F\_y[1],

'y=2\*x²': y[2],

'2\*x² dx': data\_integral[2],

'F(2\*x²)': F\_y[2],

'y=x': y[3],

'x dx': data\_integral[3],

'F(x)': F\_y[3],

'y=5ᵡ': y[4],

'5ᵡ dx': data\_integral[4],

'F(5ᵡ)': F\_y[4],

'y=1/5²+x²': y[5],

'1/5²+x² dx': data\_integral[5],

'F(1/5²+x²)': F\_y[5],

'y=eᵡ': y[6],

'eᵡ dx': data\_integral[6],

'F(eᵡ)': F\_y[6],

'y=(x+2)³': y[7],

'(x+2)³ dx': data\_integral[7],

'F((x+2)³)': F\_y[7],

}

# Рендерим главную таблицу результатов

row = 0

col = 0

worksheet.set\_column(0, 16, 40)

worksheet.set\_row(0, 20)

cell\_format = workbook.add\_format()

cell\_format.set\_align('center')

cell\_format.set\_align('vcenter')

cell\_format.set\_border()

cell\_format.set\_font\_size(12)

col\_num = 0

for key, value in main\_table\_data.items():

worksheet.write(0, col\_num, key, cell\_format)

worksheet.write\_column(1, col\_num, value, cell\_format)

col\_num += 1

col\_num = 0

worksheet.merge\_range('A6:B6',list(limits\_table\_data.keys())[0], cell\_format)

for key, value in limits\_table\_data.items():

worksheet.write\_column(6, col\_num, value, cell\_format)

col\_num += 1

col\_num = 0

# Рендерим таблицу графиков

for key, value in graph\_table.items():

worksheet.write(11, col\_num, key, cell\_format)

worksheet.write\_column(12, col\_num, value, cell\_format)

col\_num += 1

# Определяем количество строк и колонок занятых данными (с учетом отступов)

df1 = pd.DataFrame(graph\_table)

df2 = pd.DataFrame(main\_table\_data)

df3 = pd.DataFrame(limits\_table\_data)

max\_row1, max\_col1 = df1.shape

max\_row2, max\_col2 = df2.shape

max\_row3, max\_col3 = df3.shape

max\_row = max\_row1+max\_row2+max\_row3+5

# Рендерим график y=sin(x

sin\_x= workbook.add\_chart({'type': 'line'})

sin\_x.add\_series({

'name': '=Данные!$B$12',

'categories': ['Данные', 12, 0, max\_row, 0],

'values': ['Данные', 12, 1, max\_row, 1],

})

sin\_x\_dx = workbook.add\_chart({'type': 'column'})

sin\_x\_dx.add\_series({

'name': '=Данные!$C$12',

'categories': ['Данные', 12, 0, max\_row, 0],

'values': ['Данные', 12, 2, max\_row, 2],

})

sin\_x.add\_series({

'name': '=Данные!$D$12',

'categories': ['Данные', 12, 0, max\_row, 0],

'values': ['Данные', 12, 3, max\_row, 3],

})

sin\_x\_dx.combine(sin\_x)

sin\_x\_dx.set\_title({ 'name': 'График sin(x) /sin(x) dx / F(sin(x))'})

worksheet2.insert\_chart('B2', sin\_x\_dx)

# Рендерим график y=cos(x)

cos\_x= workbook.add\_chart({'type': 'line'})

cos\_x.add\_series({

'name': '=Данные!$E$12',

'categories': ['Данные', 12, 0, max\_row, 0],

'values': ['Данные', 12, 4, max\_row, 4],

})

cos\_x\_dx = workbook.add\_chart({'type': 'column'})

cos\_x\_dx.add\_series({

'name': '=Данные!$F$12',

'categories': ['Данные', 12, 0, max\_row, 0],

'values': ['Данные', 12, 5, max\_row, 5],

})

cos\_x.add\_series({

'name': '=Данные!$G$12',

'categories': ['Данные', 12, 0, max\_row, 0],

'values': ['Данные', 12, 6, max\_row, 6],

})

cos\_x\_dx.combine(cos\_x)

cos\_x\_dx.set\_title({ 'name': 'График cos(x) / cos(x) dx / F(cos(x))'})

worksheet2.insert\_chart('K2', cos\_x\_dx)

# Рендерим график y=2\*x²

f3\_x= workbook.add\_chart({'type': 'line'})

f3\_x.add\_series({

'name': '=Данные!$H$12',

'categories': ['Данные', 12, 0, max\_row, 0],

'values': ['Данные', 12, 7, max\_row, 7],

})

f3\_x\_dx = workbook.add\_chart({'type': 'column'})

f3\_x\_dx.add\_series({

'name': '=Данные!$I$12',

'categories': ['Данные', 12, 0, max\_row, 0],

'values': ['Данные', 12, 8, max\_row, 8],

})

f3\_x.add\_series({

'name': '=Данные!$J$12',

'categories': ['Данные', 12, 0, max\_row, 0],

'values': ['Данные', 12, 9, max\_row, 9],

})

f3\_x\_dx.combine(f3\_x)

f3\_x\_dx.set\_title({ 'name': 'График 2\*x² / 2\*x² dx / F(2\*x²)'})

worksheet2.insert\_chart('B18', f3\_x\_dx)

# Рендерим график y=x

f4\_x= workbook.add\_chart({'type': 'line'})

f4\_x.add\_series({

'name': '=Данные!$K$12',

'categories': ['Данные', 12, 0, max\_row, 0],

'values': ['Данные', 12, 10, max\_row, 10],})

f4\_x\_dx = workbook.add\_chart({'type': 'column'})

f4\_x\_dx.add\_series({

'name': '=Данные!$L$12',

'categories': ['Данные', 12, 0, max\_row, 0],

'values': ['Данные', 12, 11, max\_row, 11],

})

f4\_x.add\_series({

'name': '=Данные!$M$12',

'categories': ['Данные', 12, 0, max\_row, 0],

'values': ['Данные', 12, 12, max\_row, 12],

})

f4\_x\_dx.combine(f4\_x)

f4\_x\_dx.set\_title({ 'name': 'График x / x dx / F(x)'})

worksheet2.insert\_chart('K18', f4\_x\_dx)

# Рендерим график y=5ᵡ

f5\_x= workbook.add\_chart({'type': 'line'})

f5\_x.add\_series({

'name': '=Данные!$N$12',

'categories': ['Данные', 12, 0, max\_row, 0],

'values': ['Данные', 12, 13, max\_row, 13],

})

f5\_x\_dx = workbook.add\_chart({'type': 'column'})

f5\_x\_dx.add\_series({

'name': '=Данные!$O$12',

'categories': ['Данные', 12, 0, max\_row, 0],

'values': ['Данные', 12, 14, max\_row, 14],

})

f5\_x.add\_series({

'name': '=Данные!$P$12',

'categories': ['Данные', 12, 0, max\_row, 0],

'values': ['Данные', 12, 15, max\_row, 15],

})

f5\_x\_dx.combine(f5\_x)

f5\_x\_dx.set\_title({ 'name': 'График 5ᵡ / 5ᵡ dx / F(5ᵡ)'})

worksheet2.insert\_chart('B34', f5\_x\_dx)

# Рендерим график y=1/5²+x²

f6\_x= workbook.add\_chart({'type': 'line'})

f6\_x.add\_series({

'name': '=Данные!$Q$12',

'categories': ['Данные', 12, 0, max\_row, 0],

'values': ['Данные', 12, 16, max\_row, 16],

})

f6\_x\_dx = workbook.add\_chart({'type': 'column'})

f6\_x\_dx.add\_series({

'name': '=Данные!$R$12',

'categories': ['Данные', 12, 0, max\_row, 0],

'values': ['Данные', 12, 17, max\_row, 17],

})

f6\_x.add\_series({

'name': '=Данные!$S$12',

'categories': ['Данные', 12, 0, max\_row, 0],

'values': ['Данные', 12, 18, max\_row, 18],

})

f6\_x\_dx.combine(f6\_x)

f6\_x\_dx.set\_title({ 'name': 'График 1/5²+x² / 1/5²+x² dx / F(1/5²+x²)'})

worksheet2.insert\_chart('K34', f6\_x\_dx)

# Рендерим график y=eᵡ

f7\_x= workbook.add\_chart({'type': 'line'})

f7\_x.add\_series({

'name': '=Данные!$T$12',

'categories': ['Данные', 12, 0, max\_row, 0],

'values': ['Данные', 12, 19, max\_row, 19],

})

f7\_x\_dx = workbook.add\_chart({'type': 'column'})

f7\_x\_dx.add\_series({

'name': '=Данные!$U$12',

'categories': ['Данные', 12, 0, max\_row, 0],

'values': ['Данные', 12, 20, max\_row, 20],

})

f7\_x.add\_series({

'name': '=Данные!$V$12',

'categories': ['Данные', 12, 0, max\_row, 0],

'values': ['Данные', 12, 21, max\_row, 21],

})

f7\_x\_dx.combine(f7\_x)

f7\_x\_dx.set\_title({ 'name': 'График eᵡ / eᵡ dx / F(eᵡ)'})

worksheet2.insert\_chart('B50', f7\_x\_dx)

# Рендерим график y=(x+2)³

f8\_x= workbook.add\_chart({'type': 'line'})

f8\_x.add\_series({

'name': '=Данные!$W$12',

'categories': ['Данные', 12, 0, max\_row, 0],

'values': ['Данные', 12, 22, max\_row, 22],

})

f8\_x\_dx = workbook.add\_chart({'type': 'column'})

f8\_x\_dx.add\_series({

'name': '=Данные!$X$12',

'categories': ['Данные', 12, 0, max\_row, 0],

'values': ['Данные', 12, 23, max\_row, 23],

})

f8\_x.add\_series({

'name': '=Данные!$Y$12',

'categories': ['Данные', 12, 0, max\_row, 0],

'values': ['Данные', 12, 24, max\_row, 24],

})

f8\_x\_dx.combine(f8\_x)

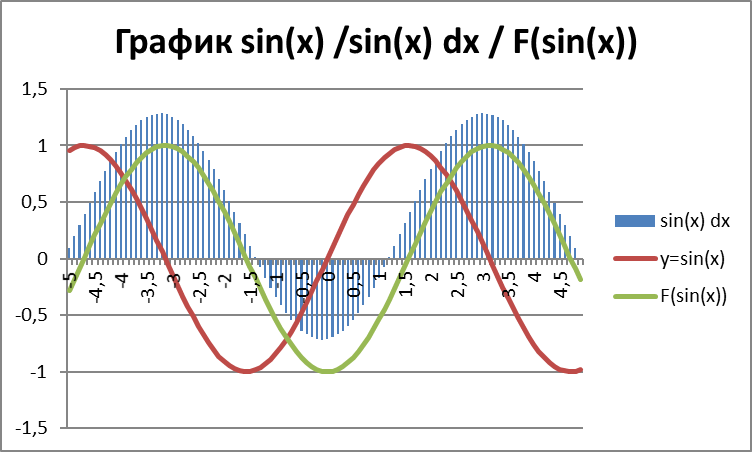
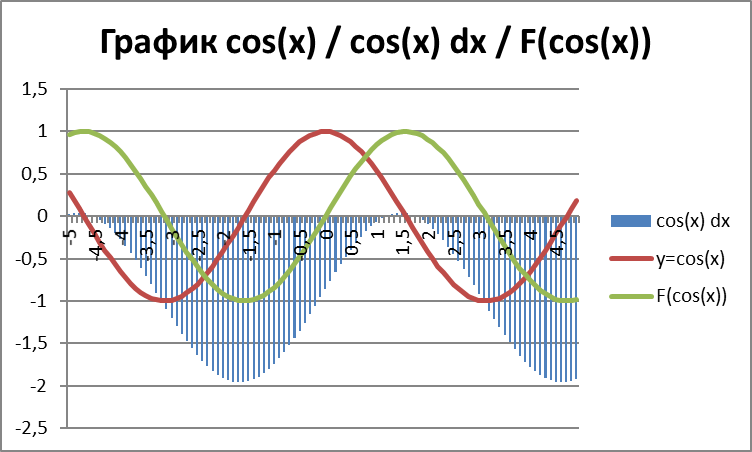
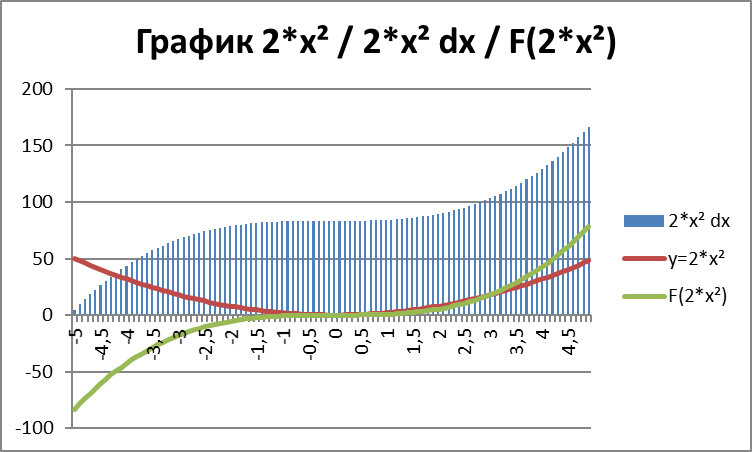
f8\_x\_dx.set\_title({ 'name': 'График (x+2)³ / (x+2)³ dx / F((x+2)³)'})

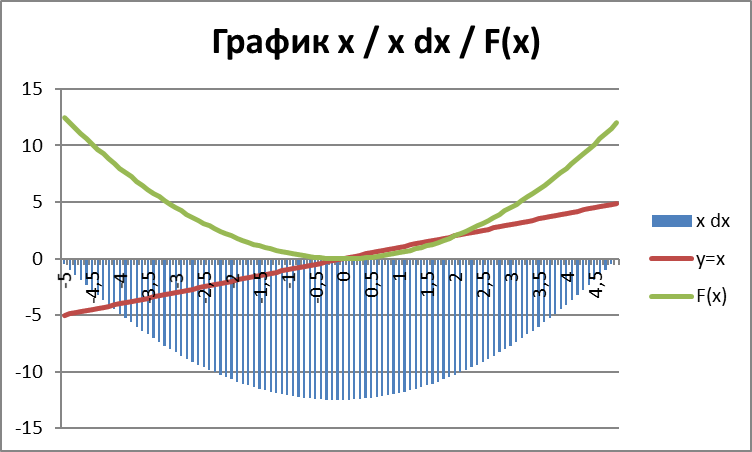
worksheet2.insert\_chart('K50', f8\_x\_dx)

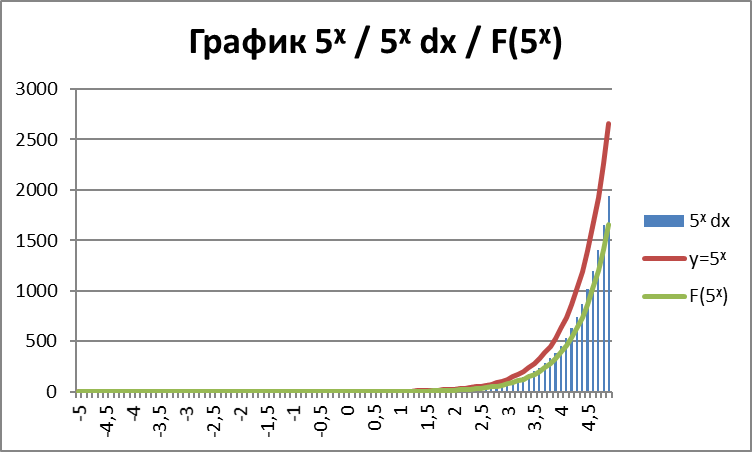
workbook.close()

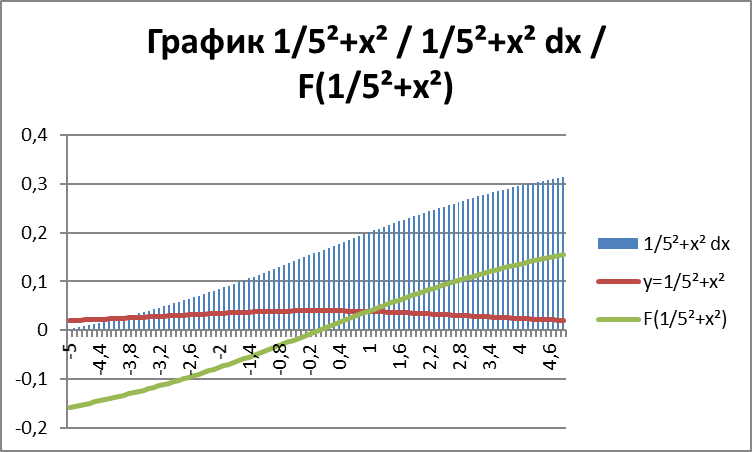
Графики:

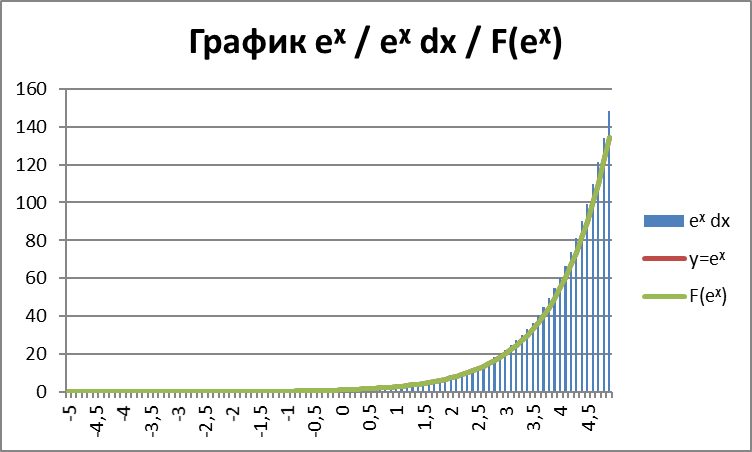
Данные графики построены по интегралам в границах от -5 до 5.

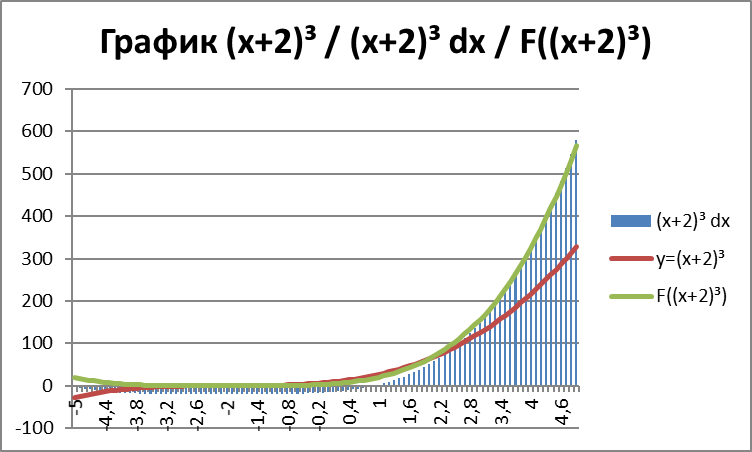
  











Выводы:  
На выходе получаем программу, которая различными методами высчитывает значения интегралов для 8 заданных функций, и на их основе строит графики.