**МИНОБРНАУКИ РОССИИ**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»  
(БГТУ им. В.Г.Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №3  
дисциплина: Вычислительная математика  
тема: «Численное интегрирование»

Выполнил: ст. группы ВТ-21  
Сидорова Ангелина Сергеевна  
Проверил: Бондаренко Т.В.

Белгород 2018

Цель работы: изучить понятие квадратурной формулы; изучить основные способы численного интегрирования; получить практические навыки решения задачи численного интегрирования с помощью ЭВМ.

Вариант 10.

Безымянный.png

Задания к работе

1. Вычислить «вручную» интегралы из таблицы вариантов заданий:

− точно (все 3 интеграла);

− по формуле центральных (средних) прямоугольников, используя для оценки точности двойной просчёт при n1=8; n2=10 (интеграл 1);

− по формуле трапеций при n=8 (интеграл 1 и 2);

− по формуле парабол (Симпсона) при n=8 (интеграл 1 и 3).

Замечание. Для вычисления значений «вручную» рекомендуется использовать Microsoft Excel или другую программу.

2. Определить погрешность вычисления интеграла 1 по каждой из формул. Результаты представить в виде таблицы 3.1.

Таблица 3.1 Оценка погрешности вычислений

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Погрешность | Формула центр. прямоугольников | | Формула трапеций n = 8 | Формула парабол n = 8 | Формула Гаусса | | | |
| n =8 | n = 10 | n = 1 | n = 2 | n = 3 | n = 4 |
| Δ | 0.083 | 0.03 | 0.1667 | 2.32\*10^-6 | 5,33 | 0.333 | 0.333336 |  |
| δ | 0.12% | 0.04% | 0.24% | 0 | 7,69% | 0.483084% | 0.483095% |  |

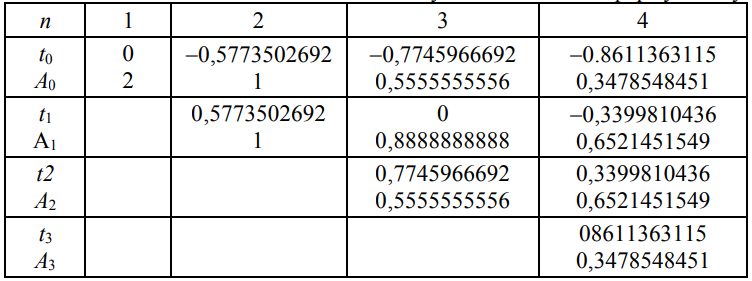
3. Описать в модуле функции, которые возвращают приближенные значения интегралов от функции f(x) с оценкой точности по принципу Рунге для методов центральных прямоугольников, трапеций и парабол. Исходными данными являются: подынтегральная функция f(x); пределы интегрирования a, b; начальное число отрезков разбиения n; точность вычисления ε. Необходимые для работы значения подынтегральной функции вычисляются непосредственной подстановкой значений аргумента в вычислительную формулу функции.

4. Составить программу для вычисления приближенных значений интегралов согласно варианту (все функции из таблицы вариантов заданий) с использованием всех функций, описанных в модуле.

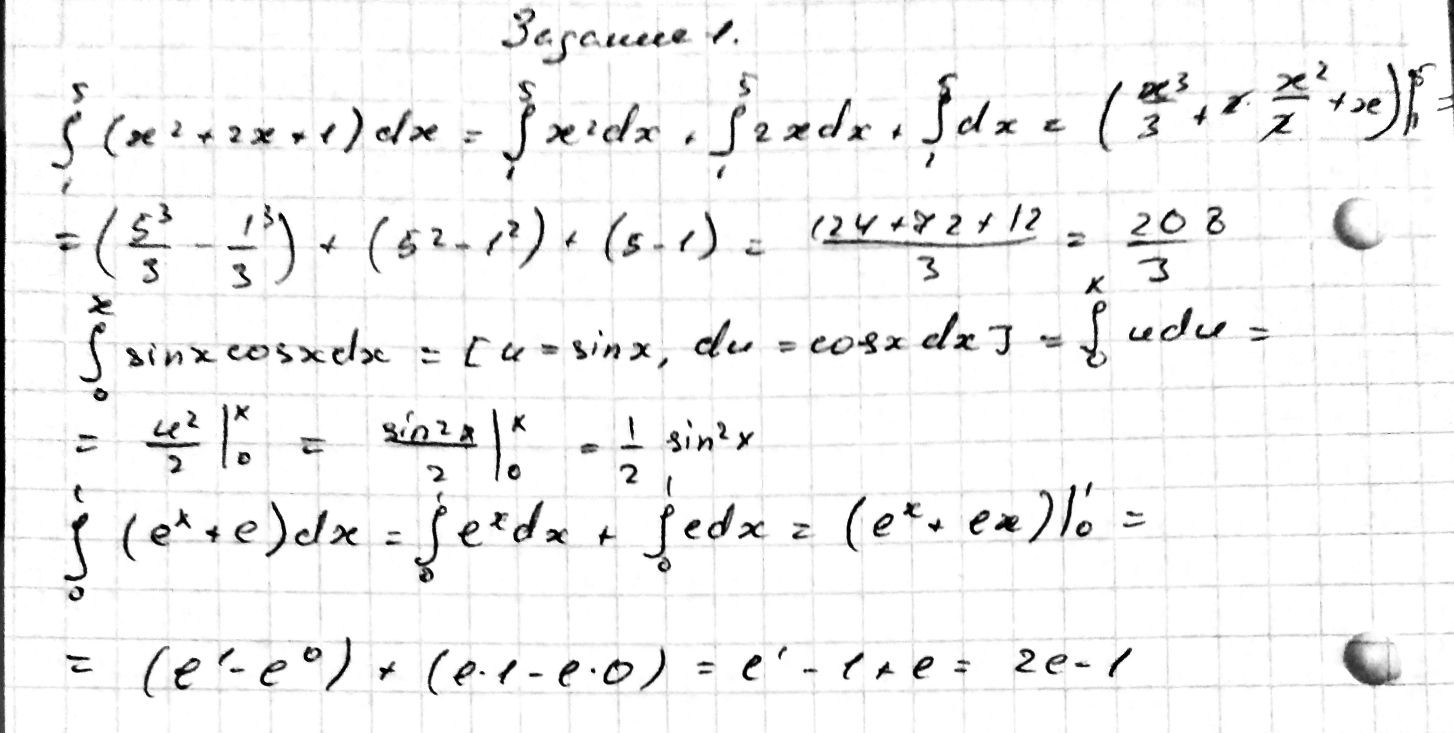
5. Вычислить «вручную» интеграл из столбца 1 таблицы заданий по формуле Гаусса при n = 2. Значения узлов ti и весов Аi приведены в таблице 3.2.

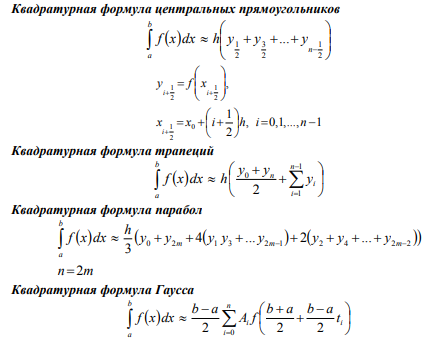
6. Описать в модуле функцию для вычисления приближенного значения интеграла от функции f(x) по формуле Гаусса при n = 1, 2, 3 ,4. Добавить в составленную программу вычисление приближенных значений 25 интегралов (все функции из таблицы вариантов заданий) с использованием формулы Гаусса.

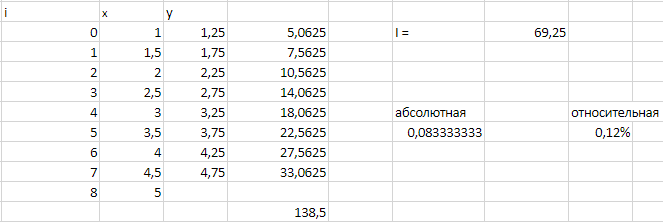
Таблица 3.2. Значение узлов и весов для формулы Гаусса

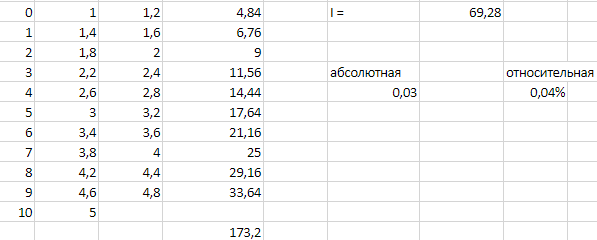


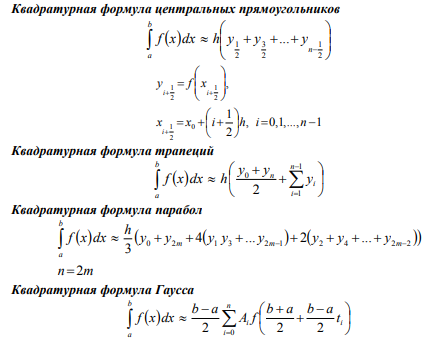
1. Заполнить значения погрешности вычисления интеграла 1 в таблице 3.1 для формулы Гаусса при n = 1, 2, 3, 4.

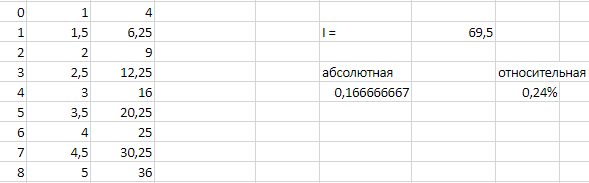


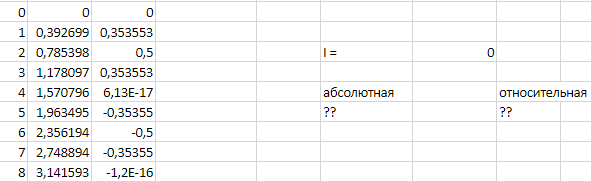


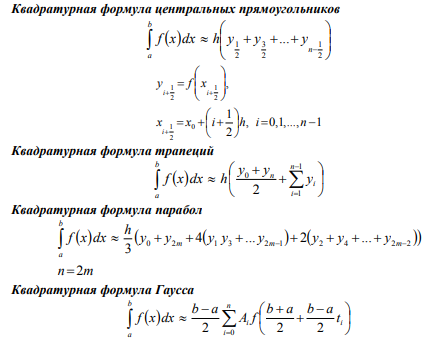


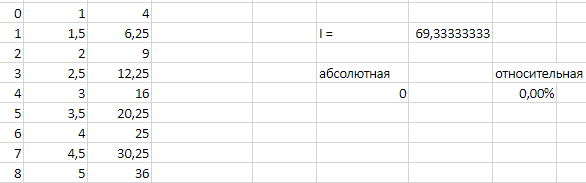


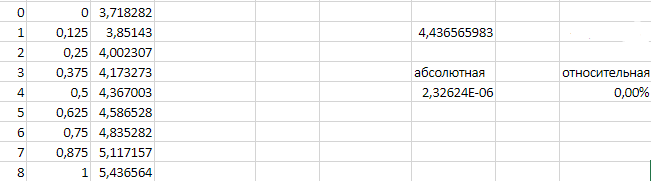


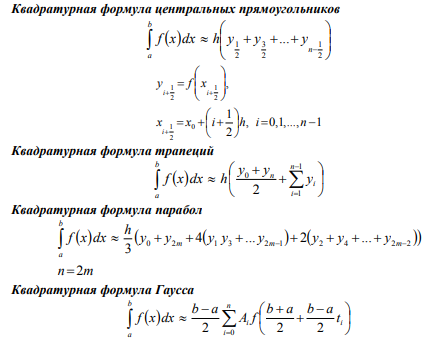












I=69.3

Vm.h

#ifndef VM\_H\_INCLUDED

#defineVM\_H\_INCLUDED

float f\_pryam(float (\*funct)(float x),float a,float b,int n, float e);

float f\_trap(float (\*funct)(float x),float a,float b,int n,float e);

float f\_parab(float (\*funct)(float x),float a,float b,int n,float e);

float f\_gauss(float (\*funct)(float x),float a, float b, int n);

#include "integral.c"

#endif // VM\_H\_INCLUDED

Vm.c

#include <math.h>

#include "integral.h"

float f\_pryam(float (\*funct)(float x),float a,float b,int n,float e) /\*возвращает значение интеграла функции funct(x) на промежутке [a;b] с начальным числом разбиения на отрезки n вычисленного с помощью метода центральных прямоугольников с точностью e \*/

{

float I = 0,I1 = 0,I2 = 0,i,y;

float h = (b - a)/n;

do

{ I2 = I1;

I1 = 0;

i = a + h/2;

while(i < b)

{

y = funct(i);

I1 = I1 + y;

i += h;

}

I1 = I1\*h;

h = h/2;

}

while (fabs(I1 - I2)/3 > e);

I = I2 + (I2- I1)/3;

return I;

}

float f\_trap(float (\*funct)(float x),float a,float b,int n,float e) /\*возвращает значение интеграла функции funct(x) на промежутке [a;b] с начальным числом разбиения на отрезки n вычисленного с помощью метода трапеций точностью e \*/

{

float I = 0,I1 = 0,I2 = 0,i,y;

float h = (b - a)/n;

do

{ I2 = I1;

I1 = 0;

i = a + h;

float s = 0;

while(i < b)

{

y = funct(i);

s = s + y;

i += h;

}

I1 = h\*((funct(a)+funct(b))/2+s);

h = h/2;

}

while (fabs(I1 - I2)/3 > e);

I = I2 + (I2- I1)/3;

return I;

}

float f\_parab(float (\*funct)(float x),float a,float b,int n,float e) /\*возвращает значение интеграла функции funct(x) на промежутке [a;b] с начальным числом разбиения на отрезки n вычисленного с помощью метода параболы с точностью e \*/

{

float I = 0,I1 = 0,I2 = 0,i,y;

float h = (b - a)/n;

do

{ I2 = I1;

I1 = 0;

i = a+h;

float s = funct(a) + funct(b);

int k = 1;

while(i < b)

{

y = funct(i);

if (k%2 == 0)

s = s + 2\*y;

else

s = s + 4\*y;

i += h;

k++;

}

I1 = h/3\*s;

h = h/2;

}

while (fabs(I1 - I2)/15 > e);

I = I2 + (I2- I1)/15;

return I;

}

float f\_gauss(float (\*funct)(float x),float a, float b, int n) /\*возвращает значение интеграла функции funct(x) на промежутке [a;b] с начальным числом разбиения на отрезки n вычисленного с помощью метода Гаусса \*/

{

float \*\*A = calloc(4, sizeof(float \*));

float \*\*t = calloc(4, sizeof(float \*));

for (int i = 0; i < 4; i++)

{

A[i] = calloc(i+1,sizeof(float));

t[i] = calloc(i+1,sizeof(float));

}

A[0][0] = 2; A[1][0] = 1; A[2][0] = 0.555555556; A[3][0] = 0.347854845;

A[1][1] = 1; A[2][1] = 0.888888889; A[3][1] = 0.652145155;

A[2][2] = 0.555555556; A[3][2] = 0.652145155;

A[3][3] = 0.347854845;

t[0][0] = 0; t[1][0] = -0.577350269; t[2][0] = -0.774596669; t[3][0] = -0.861136312;

t[1][1] = 0.577350269; t[2][1] = 0; t[3][1] = -0.339981044;

t[2][2] = 0.774596669; t[3][2] = 0.339981044;

t[3][3] = 0.861136312;

float I = (b-a)/2,s = 0, x = (b+a)/2,y;

for (int i = 0; i < n; i++)

s += A[n-1][i]\*funct(x+(b-a)/2\*t[n-1][i]);

I = I\*s;

printf("\n\nI = %f", I);

return I;

}

main.c

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

#include "vm.h"

float fun1(float x)

{

return (x\*x+2\*x+1);

}

int main()

{

float Ipr, Itr, Ipar, I;

int n;

setlocale(LC\_ALL, "Rus");

Ipr = f\_praym(fun1,1,5,8,0.01);

Itr = f\_trap(fun1,1,5,8,0.01);

Ipar = f\_parab(fun1,1,5,8,0.01);

printf("%f\n%f\n%f",Ipr,Itr,Ipar);

printf("введите n : ");

scanf("%i",&n);

float f = 208/3;

I = f\_gauss(fun1,1,5,4);

printf("I = %f\n",I);

printf("абсолютная погрешность : %f\n",fabs(I-f));

printf("относительная погрешность : %f%%\n",fabs(I-f)/f\*100);

}

