**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 ПВ-21  Браткова Ирина Олеговна |

Белгород 2017

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

**Задания к работе**

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

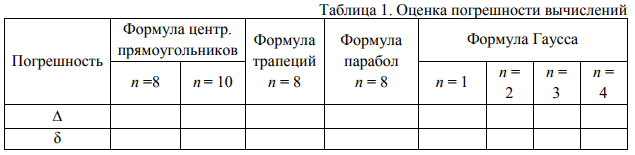
− точно (все 3 интеграла);

− 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.

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

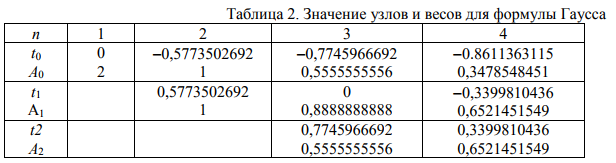
3. Описать в модуле функции, которые возвращают приближенные значения интегралов от функции f(x) с оценкой точности по принципу Рунге для методов центральных прямоугольников, трапеций и парабол.

Реализовать возможность ввода исходных данных (таблица значений подынтегральной функции) с клавиатуры и из файла.

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 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 интегралов (все функции из таблицы вариантов заданий) с использованием формулы Гаусса.



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

**Вариант 4.**

**Задание 1: см тетрадь.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 | 0,209207 | 0,134291 | 3,034 | 0,72213 | 8,2483 | 0 | 9.3207 | 0 |
| δ | 0,021% | 0.013% | 0.3161% | 0.0508% | 1% | 0% | 0.06% | 0% |

**Задание 2:**

**Задание 3:**

**Заголовочный файл:**

|  |  |
| --- | --- |
| typedef float(\*func\_f)(float);  float fun\_1(float x);  float fun\_2(float x);  float fun\_3(float x);  float CentrPry (func\_f f, float a, float b, size\_t n);  float RungeCentrPry (func\_f f, float a, float b, size\_t n);  float Trap (func\_f f, float a, float b, size\_t n);  float RungeTrap (func\_f f, float a, float b, size\_t n);  float Simp (func\_f f, float a, float b, size\_t n);  float RungeSimp (func\_f f, float a, float b, size\_t n);  float Gauss (func\_f f, float a, float b, size\_t n); | Функциональный тип.  Функция интеграла 1  Функция интеграла 2  Функция интеграла 3  Метод центральных прямоугольников  Оценка точности ц.прям.  Метод трапеций  Оценка точности м.трапеций  Метод парабол (Симпсона)  Оценка точности Симпсона  Метод гаусса |

**Файл .c**

#include <stdio.h>#include <stdlib.h>#include <math.h>#include <locale.h>#include "mod.h"

float fun\_1(float x) { return (powf(x, 4)+1);}

float fun\_2(float x){ return (cos(2\*x)); }

float fun\_3(float x){ return (powf(5, x)); }

float CentrPry (func\_f f, float a, float b, size\_t n)

{

float h = (b-a)/n, sum=0, xi2=0; int i;

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

{

xi2 =a+(i+0.5)\*h; sum+=f(xi2);

}

return h\*sum;

}

float RungeCentrPry (func\_f f, float a, float b, size\_t n)

{

float in = CentrPry(f, a, b, n); float in2 = CentrPry(f, a, b, 2\*n);

float ch = fabs(in2-in); return ch/3;

}

float Trap (func\_f f, float a, float b, size\_t n)

{

float h = (b-a)/n; float sum = 0, xi=0; int i;

for (i=1, xi=a+h; i<=n-1; i++)

{

sum+=f(xi); xi +=h;

}

sum += (f(a)+f(b))/2; return h\*sum;

}

float RungeTrap (func\_f f, float a, float b, size\_t n)

{

float in = Trap(f, a, b, n); float in2 = Trap(f, a, b, 2\*n);

float ch = fabs(in2-in); return ch/3;

}

float Simp (func\_f f, float a, float b, size\_t n)

{

float h = (b-a)/n; float xi=0, sum=0; int i;

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

{

xi= a+i\*h; sum+=f(xi-h)+4\*f(xi)+f(xi+h);

}

return sum\*h/3;

}

float RungeSimp (func\_f f, float a, float b, size\_t n)

{

float in = Simp(f, a, b, n); float in2 = Simp(f, a, b, 2\*n);

float ch = fabs(in2-in); return ch/15;

}

float Gauss (func\_f f, float a, float b, size\_t n)

{

float A[n]; float t[n];

switch (n)

{

case 1:

t[0] = 0; A[0] = 2;

break;

case 2:

t[0] = -0.5773503; A[0] = 1; t[1] = 0.5773503; A[1] = 1;

break;

case 3:

t[0] = -0.7745967; A[0] = 0.5555556; t[1] = 0; A[1] = 0.8888889; t[2] = 0.7745967;

A[2] = 0.5555556;

break;

case 4:

t[0] = -0.8611363; A[0] = 0.3478548;

t[1] = -0.3399810; A[1] = 0.6521451;

t[2] = 0.3399810; A[2] = 0.6521451;

t[3] = 0.8611363; A[3] = 0.3478548;

break;

}

float h=(b-a)/n; float sum=0, xi=0, tmp1=(b+a)/2, tmp2=(b-a)/2; int i;

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

{

sum+=A[i]\*f(tmp1+tmp2\*t[i]);

}

return sum\*tmp2;

}

**Главная функция:**

#include <stdio.h>#include <stdlib.h>#include <math.h>#include <locale.h>#include "mod.h"

void main()

{

setlocale(LC\_ALL, "Rus");

printf("Интеграл (x^4+1); [-1, 2]\n"); printf("Центральные прямоугольники\n");

printf(" Значение: %f #", CentrPry(fun\_1, -1, 2, 8));

printf(" Точность: %f \n", RungeCentrPry(fun\_1, -1, 2, 8));

printf("Трапеции\n"); printf(" Значение: %f #", Trap(fun\_1, -1, 2, 8));

printf(" Точность: %f \n", RungeTrap(fun\_1, -1, 2, 8));

printf("Параболы (Симпсона)\n"); printf(" Значение: %f #", Simp(fun\_1, -1, 2, 4));

printf(" Точность: %f \n", RungeSimp(fun\_1, -1, 2, 4)); printf("\n");

printf("Интеграл cos(2x); [0, pi/3]\n"); printf("Центральные прямоугольники\n");

printf(" Значение: %f #", CentrPry(fun\_2, 0, M\_PI/3, 8));

printf(" Точность: %f \n", RungeCentrPry(fun\_2, 0, M\_PI/3, 8));

printf("Трапеции\n"); printf(" Значение: %f #", Trap(fun\_2, 0, M\_PI/3, 8));

printf(" Точность: %f \n", RungeTrap(fun\_2, 0, M\_PI/3, 8));

printf("Параболы (Симпсона)\n"); printf(" Значение: %f #", Simp(fun\_2, 0, M\_PI/3, 4));

printf(" Точность: %f \n", RungeSimp(fun\_2, 0, M\_PI/3, 4)); printf("\n");

printf("Интеграл 5^x; [0, 2]\n"); printf("Центральные прямоугольники\n");

printf(" Значение: %f #", CentrPry(fun\_3, 0, 2, 8));

printf(" Точность: %f \n", RungeCentrPry(fun\_3, 0, 2, 8));

printf("Трапеции\n"); printf(" Значение: %f #", Trap(fun\_3, 0, 2, 8));

printf(" Точность: %f \n", RungeTrap(fun\_3, 0, 2, 8));

printf("Параболы (Симпсона)\n"); printf(" Значение: %f #", Simp(fun\_3, 0, 2, 4));

printf(" Точность: %f \n", RungeSimp(fun\_3, 0, 2, 4));

}

**Выполнение:**

