МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

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

по дисциплине: «Вычислительная математика»

Выполнил: ст. группы ПВ-211
Медведев Дмитрий Сергеевич

Проверила:
Бондаренко Т.В.

Белгород 2023 г.

Численное интегрирование

Вариант 8

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

Ход работы

8	$\int_0^3 (x^3+x^2)dx$	$\int_0^{\pi/3} \cos(3x)dx$	$\int_0^2 (3^x+2)dx$

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

– точно (все 3 интеграла);

№1.

а) $\int_0^3 (x^3+x^2)\,dx = \int_0^3 x^3dx + \int_0^3 x^2dx = \frac{x^4}{4}\Big|_0^3 + \frac{x^3}{3}\Big|_0^3 = \frac{81}{4}+\frac{27}{3} = 29,25$

б) $\int_0^{\pi/3} \cos 3x\,dx = \frac{1}{3}\int_0^{\pi/3} \cos(3x)\,d3x = \frac{1}{3}\sin 3x\Big|_0^{\pi/3} = \frac{1}{3}(\sin\pi - \sin 0) = 0$

в) $\int_0^2 (3^x+2)\,dx = \int_0^2 3^x dx + 2\int_0^2 dx = \frac{3^x}{\ln 3}\Big|_0^2 + x\Big|_0^2 = \left(\frac{3^2}{\ln 3} - \frac{3^0}{\ln 3}\right) + 2(2-0) =$

$= \frac{8}{\ln 3} + 4 = 11,282$

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

n	0	1	2	3	4	5	6	7		
x	0	0,375	0,75	1,125	1,5	1,875	2,25	2,625		
y(x+1/2)	0,041748	0,494385	1,702881	3,983643	7,653076	13,02759	20,42358	30,15747		
							Сумма:	29,05664		
n	0	1	2	3	4	5	6	7	8	9
x	0	0,3	0,6	0,9	1,2	1,5	1,8	2,1	2,4	2,7
y(x+1/2)	0,025875	0,293625	0,984375	2,260125	4,282875	7,214625	11,21738	16,45313	23,08388	31,27163
									Сумма:	29,12625

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

n	0	1	2	3	4	5	6	7	8
x	0	0,375	0,75	1,125	1,5	1,875	2,25	2,625	3
y	0	0,193359	0,984375	2,689453	5,625	10,10742	16,45313	24,97852	36
									29,63672
n	0	1	2	3	4	5	6	7	8
x	0	0,392699	0,785398	1,178097	1,570796	1,963495	2,356194	2,748894	3,141593
y	1	0,382683	-0,70711	-0,92388	-1,8E-16	0,92388	0,707107	-0,38268	-1
									0

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

n	0	1	2	3	4	5	6	7	8
x	0	0,375	0,75	1,125	1,5	1,875	2,25	2,625	3
y	0	0,193359	0,984375	2,689453	5,625	10,10742	16,45313	24,97852	36
								Сумма:	29,25
n	0	1	2	3	4	5	6	7	8
x	0	0,25	0,5	0,75	1	1,25	1,5	1,75	2
y	3	3,316074	3,732051	4,279507	5	5,948222	7,196152	8,838521	11
								Сумма:	11,28214

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 табл. 3.1.

Погрешность	Формула центральных прямоугольников		Формула трапеций	Формула парабол n = 8	Формула Гаусса			
	n = 8	n = 10	n = 8		n = 1	n = 2	n = 3	n = 4
Δ1	0,193359375	0,12375	0,38671875	0				
δ1	0,006610577	0,004230769	0,013221154	0				
Δ2			0	0,000141965				
δ2			0	1,25833E-05				

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.

```
#include "integrals.h"
#include "utility"
#include "vector"
#include "stdlib.h"


typedef std::pair<double, double> GridElement;

typedef std::vector<GridElement> Grid;
```

```
double calculateIntegralByCentralRectangle(Function function, double lowBorder,
                                           double highBorder, int nParts) {
    Grid grid(nParts);

    double step = (highBorder - lowBorder) / nParts;
    double sum = 0;
    for (int i = 0; i < nParts; i++) {
        grid[i].first = lowBorder + (i + 0.5) * step;
        grid[i].second = function(grid[i].first);
        sum += grid[i].second;
    }

    return sum * step;
}

double calculateIntegralByTrapezoid(Function function, double lowBorder,
                                    double highBorder, int nParts) {
    Grid grid(nParts + 1);

    grid[0] = {lowBorder, function(lowBorder)};
    grid[grid.size() - 1] = {highBorder, function(highBorder)};

    double step = (highBorder - lowBorder) / nParts;
    double sum = (grid[0].second + grid[grid.size() - 1].second) / 2;
    for (int i = 1; i < nParts; i++) {
        grid[i].first = lowBorder + i * step;
        grid[i].second = function(grid[i].first);
        sum += grid[i].second;
    }

    return step * sum;
}

double calculateIntegralByParable(Function function, double lowBorder,
                                  double highBorder, int nParts) {
    Grid grid(nParts + 1);

    grid[0] = {lowBorder, function(lowBorder)};
    grid[grid.size() - 1] = {highBorder, function(highBorder)};

    double step = (highBorder - lowBorder) / nParts;
    double sum = grid[0].second + grid[grid.size() - 1].second;

    for (int i = 1; i < grid.size() - 1; i += 2) {
        grid[i].first = lowBorder + i * step;
        grid[i].second = function(grid[i].first);
        sum += 4 * grid[i].second;
    }

    for (int i = 2; i < grid.size() - 1; i += 2) {
        grid[i].first = lowBorder + i * step;
        grid[i].second = function(grid[i].first);
        sum += 2 * grid[i].second;
    }

    return step / 3 * sum;
}

double calculateIntegralByCentralRectangleWithEps(Function function, double lowBorder,
                                                  double highBorder, int nParts,
double &eps) {
    double integralValueNParts = calculateIntegralByCentralRectangle(function,
lowBorder, highBorder, nParts);
    double difference = abs(integralValueNParts -
calculateIntegralByCentralRectangle(function, lowBorder,

highBorder, nParts / 2));
    eps = (double) 1 / 3 *difference;
    return integralValueNParts;
}
```

```
double calculateIntegralByTrapezoidWithEps(Function function, double lowBorder,
                                           double highBorder, int nParts, double &eps)
{
    double integralValueNParts = calculateIntegralByTrapezoid(function, lowBorder,
highBorder, nParts);
    double difference = abs(integralValueNParts -
calculateIntegralByTrapezoid(function, lowBorder,

highBorder, nParts / 2));
    eps = (double) 1 / 3 * difference;
    return integralValueNParts;
}

double calculateIntegralByParableWithEps(Function function, double lowBorder,
                                         double highBorder, int nParts, double &eps) {
    double integralValueNParts = calculateIntegralByParable(function, lowBorder,
highBorder, nParts);
    double difference = abs(integralValueNParts - calculateIntegralByParable(function,
lowBorder,

highBorder, nParts / 2));
    eps = (double) 1 / 15 * difference;
    return integralValueNParts;
}
```

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

```
#include <iostream>
#include "libs/integrals/integrals.h"
#include "cmath"
#include "windows.h"

double f1(double x) {
    return pow(x, 3) + pow(x, 2);
}

double f2(double x) {
    return cos(3 * x);
}

double f3(double x) {
    return pow(3, x) + 2;
}

int main() {
    SetConsoleOutputCP(CP_UTF8);

    std::cout << "Вычисление по формуле центральных прямоугольников\n" <<
              "Интеграл 1, n = 8: " << calculateIntegralByCentralRectangle(f1, 0, 3,
8) <<
              "\nИнтеграл 1, n = 10: " << calculateIntegralByCentralRectangle(f1, 0,
3, 10) <<
              "\n\nВычисление по формуле трапеции\n" <<
              "Интеграл 1, n = 8: " << calculateIntegralByTrapezoid(f1, 0, 3, 8) <<
              "\nИнтеграл 2, n = 8: " << calculateIntegralByTrapezoid(f2, 0,
std::numbers::pi / 3, 8) <<
              "\n\nВычисление по формуле парабол\n" <<
              "Интеграл 1, n = 8: " << calculateIntegralByParable(f1, 0, 3, 8) <<
              "\nИнтеграл 3, n = 8: " << calculateIntegralByParable(f3, 0, 3, 8);
}
```

```
C:\Users\dimam\Desktop\4Semester\VM\LR3\Code\cmake-build-debug\Code.exe
Вычисление по формуле центральных прямоугольников
Интеграл 1, n = 8: 29.0566
Интеграл 2, n = 8: 1.30795e-16
Интеграл 3, n = 8: 11.2591

Вычисление по формуле трапеций
Интеграл 1, n = 8: 29.6367
Интеграл 2, n = 8: 8.71967e-17
Интеграл 3, n = 8: 11.3276

Вычисление по формуле парабол
Интеграл 1, n = 8: 29.25
Интеграл 2, n = 8: 7.75082e-17
Интеграл 3, n = 8: 11.2821
Process finished with exit code 0
```

5. Вычислить «вручную» интеграл из столбца 1 табл. 3.1 по формуле Гаусса при n = 2. Значения узлов ti и весов Ai приведены в табл. 3.2.

N 5

$$\int_0^3 (x^3+x^2)\,dx \approx \frac{3-0}{2}\cdot\sum_{i=0}^{1} A_i f\left(\frac{3+0}{2}+\frac{3-0}{2}t_i\right) =$$

$$= \frac{3}{2}\left(1\cdot\left(\left(\frac{3}{2}+\frac{3}{2}\cdot\left(-\frac{1}{\sqrt{3}}\right)\right)^3+\left(\frac{3}{2}+\frac{3}{2}\cdot\left(-\frac{1}{\sqrt{3}}\right)\right)^2\right)+1\cdot\left(\left(\frac{3}{2}+\frac{3}{2}\cdot\left(\frac{1}{\sqrt{3}}\right)\right)^3+\right.\right.$$

$$\left.\left.+\left(\frac{3}{2}+\frac{3}{2}\left(\frac{1}{\sqrt{3}}\right)\right)^2\right)\right) = 28,92$$

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.

```
struct gaussCoefficient {
    double t;
    double A;
};


std::vector<std::vector<gaussCoefficient>> gaussCoefficients{
        {},
        {{0,        2}},
        {{-0.57735, 1},       {0.57735,  1}},
        {{-0.77459, 0.55555}, {0,        0.88888}, {0.77459, 0.55555}},
        {{-0.86114, 0.34785}, {-0.33998, 0.65215}, {0.33998, 0.65215},
{0.861136, 0.34785}}
};
```

```
double calculateIntegralByGauss(Function function, double lowBorder,
                                double highBorder, int nParts) {
    double sum = 0;
    for (int i = 0; i < nParts; i++) {
        double value = gaussCoefficients[nParts][i].A;
        value *= function((highBorder + lowBorder) / 2 +
                          (highBorder - lowBorder) / 2 *
gaussCoefficients[nParts][i].t);
        sum += value;
    }

    return (highBorder - lowBorder) / 2 * sum;
}

int main() {
    SetConsoleOutputCP(CP_UTF8);

    std::cout << "Вычисление интеграла по формуле Гаусса\n";

    for (int i = 1; i <= 4; i++) {
        std::cout << "Интеграл 1, n = " << i << ": " <<
                  calculateIntegralByGauss(f1, 0, 3, i) << "\n";
    }

    std::cout << "Интеграл 2, n = 4: " <<
              calculateIntegralByGauss(f2, 0, std::numbers::pi / 3, 4) << "\
n";
    std::cout << "Интеграл 3, n = 4: " <<
              calculateIntegralByGauss(f3, 0, 2, 4) << "\n";
}
```

```
C:\Users\dimam\Desktop\4Semester\VM\LR3\Code\cmake-build-debug\Code.exe
Вычисление интеграла по формуле Гаусса
Интеграл 1, n = 1: 16.875
Интеграл 1, n = 2: 29.25
Интеграл 1, n = 3: 29.2495
Интеграл 1, n = 4: 29.2499
Интеграл 2, n = 4: 2.47642e-07
Интеграл 3, n = 4: 11.2819

Process finished with exit code 0
```

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

Погрешность	Формула центральных прямоугольников		Формула трапеций	Формула парабол n = 8	Формула Гаусса			
	n = 8	n = 10	n = 8		n = 1	n = 2	n = 3	n = 4
Δ1	0,193359375	0,12375	0,38671875	0				
δ1	0,006610577	0,004230769	0,013221154	0				
Δ2			0	0,000141965	12,375	0	0,0005	0,0001
δ2			0	1,25833E-05	0,423077	0	1,71E-05	3,42E-06

Вывод: в ходе лабораторной работы мы изучили понятие квадратурной формулы; изучили основные способы численного интегрирования; получили практические навыки решения задачи численного интегрирования с помощью ЭВМ.