МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Белгородский государственный технологический университет им. В.Г. Шухова»

**ИНСТИТУТ ЭНЕРГЕТИКИ, ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЯЮЩИХ СИСТЕМ Кафедра программного обеспечения вычислительной техники и автоматизированных систем**

Лабораторная работа №3

по дисциплине: Вычислительная математика:

Численное интегрирование

Работу выполнил:

Студент группы ПВ-21:

Донцов Александр Алексеевич

Проверил:

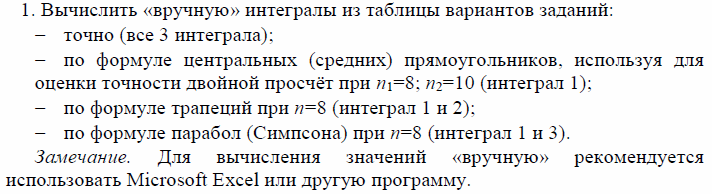
Бондаренко Татьяна Владимировна

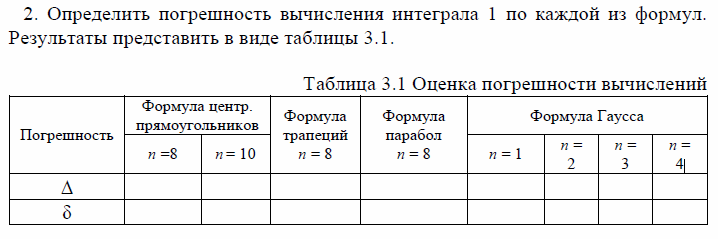
Белгород – 2018

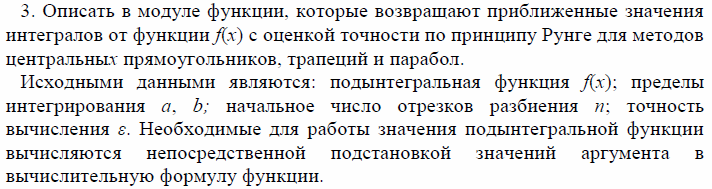
**Численное интегрирование**

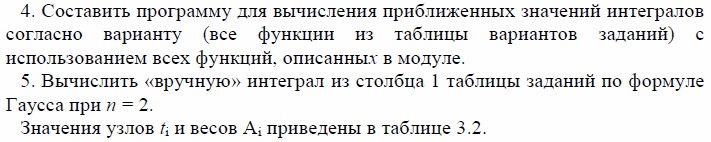
**Цель работы:** изучить понятие квадратурной формулы; изучить основные способы численного интегрирования; получить практические навыки решения задачи численного интегрирования с помощью ЭВМ.

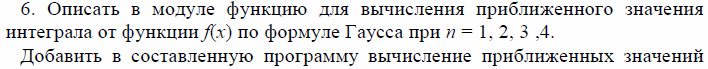
**Задания к работе**

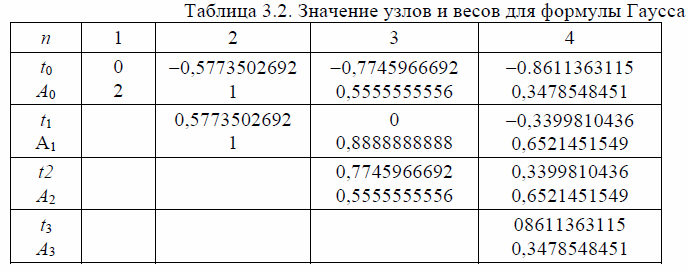






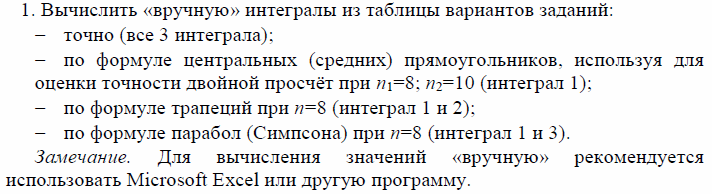


\





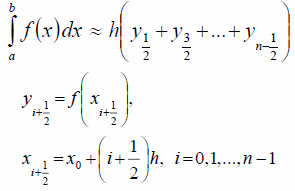
**Ход работы**



**точно**



**Формула центральных прямоугольников**



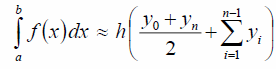
При n = 8

= 0.333

При n = 10

= 0.3

**По формуле трапеций**



n = 8



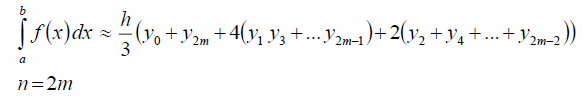
= 0.333

0,333\*(27,3

= 0.393

0,393\*(1,89

**По формуле парабол**



n = 8

h = 0,333



0,111\*(2+29+4(2,1+3,4+8,6+20,1)+2(2,4+5,4+13,4+29))=167,8

0,111\*(0,5+1,4+4(0,41+0,52+0,67+0,9)+2(0,6+0,8+1,1+1,4)) = 1

**№2**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
|  |  |  |  |  | 0,25 | 0,25 | 0,25 | - |
|  |  |  |  |  | 0,9% | 0,9% | 0,9% | - |

**№5**

n = 2

*26,25*

**Реализация кода**

#include "pch.h"

#include <stdio.h>

#include <stdlib.h>

#include <Math.h>

#include <locale.h>

#include <conio.h>

#define PI 3.14

#define N 4

typedef struct {//элемент матрицы

double t;

double a;

} point;

point matrConst[N][N];// матрица структур

void initMatrConst(point m[N][N]);//функция инициализации матрицы структур

double function1(double);//входная подинтегральная функция 1

double function2(double);//входная подинтегральная функция 2

double function3(double);//входная подинтегральная функция 3

double methodSenterRectagle(double a, double b, int n, double(\*f)(double));//метод центральных прямоугольников

double methodTrapezium(double a, double b, int n, double(\*f)(double));//метод трапеций

double methodParabola(double a, double b, int n, double(\*f)(double));//метод парабол

double methodGauss(double a, double b, int n, double(\*f)(double));//метод Гауса

/\*

int main() {

setlocale(0, "Rus");

initMatrConst(matrConst);

//инициализация указателя на функции

double(\*f1) (double) = &function1;

double(\*f2) (double) = &function2;

double(\*f3) (double) = &function3;

printf("n = 8\n метод центральных прямоугольников\n");

double res1 = methodSenterRectagle(0, 3, 8, f1);

double res2 = methodSenterRectagle(0, PI, 8, f2);

double res3 = methodSenterRectagle(0, 1, 8, f3);

printf("1) %lf\n2) %lf\n3) %lf\n\n", res1, res2, res3);

printf("метод трапеций\n");

res1 = methodTrapezium(0, 3, 8, f1);

res2 = methodSenterRectagle(0, PI, 8, f2);

res3 = methodSenterRectagle(0, 1, 8, f3);

printf("1) %lf\n2) %lf\n3) %lf\n\n", res1, res2, res3);

printf("метод парабол\n");

res1 = methodParabola(0, 3, 8, f1);

res2 = methodSenterRectagle(0, PI, 8, f2);

res3 = methodSenterRectagle(0, 1, 8, f3);

printf("1) %lf\n2) %lf\n3) %lf\n\n", res1, res2, res3);

printf("n = 10\n метод центральных прямоугольников\n");

res1 = methodSenterRectagle(0, 3, 10, f1);

res2 = methodSenterRectagle(0, PI, 10, f2);

res3 = methodSenterRectagle(0, 1, 10, f3);

printf("1) %lf\n2) %lf\n3) %lf\n\n", res1, res2, res3);

printf("метод трапеций\n");

res1 = methodTrapezium(0, 3, 10, f1);

res2 = methodSenterRectagle(0, PI, 10, f2);

res3 = methodSenterRectagle(0, 1, 10, f3);

printf("1) %lf\n2) %lf\n3) %lf\n\n", res1, res2, res3);

printf("метод парабол\n");

res1 = methodParabola(0, 3, 10, f1);

res2 = methodSenterRectagle(0, PI, 10, f2);

res3 = methodSenterRectagle(0, 1, 10, f3);

printf("1) %lf\n2) %lf\n3) %lf\n\n", res1, res2, res3);

return 0;

}

\*/

void initMatrConst(point m[N][N]) {//функция инициализации матрицы структур

//N = 1

m[0][0].t = 0;

m[0][0].a = 2;

//N = 2

m[0][1].t = -0.5773502692;

m[0][1].a = 1;

m[1][1].t = 0.5773502692;

m[1][1].a = 1;

//N = 3

m[0][2].t = -0.7745966692;

m[0][2].a = 0.5555555556;

m[1][2].t = 0;

m[1][2].a = 0.8888888888;

m[2][2].t = 0.7745966692;

m[2][2].a = 0.5555555556;

//N = 4

m[0][3].t = -0.8611363115;

m[0][3].a = 0.3478548451;

m[1][3].t = -0.3399810436;

m[1][3].a = 0.6521451549;

m[2][3].t = 0.3399810436;

m[2][3].a = 0.6521451549;

m[3][3].t = 0.8611363115;

m[3][3].a = 0.3478548451;

}

double function1(double x) {//входная подинтегральная функция 1

return pow(x,3) + 2;

}

double function2(double x) {//входная подинтегральная функция 2

return sin(x);

}

double function3(double x) {//входная подинтегральная функция 3

return exp(x) / 2;

}

double methodSenterRectagle(double a, double b, int n, double(\*f)(double)) {//метод центральных прямоугольников

double h = (b - a) / n;

double sumFunc = 0;

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

sumFunc += f(a + h \* i + h / 2);

return h \* sumFunc;

}

double methodTrapezium(double a, double b, int n, double(\*f)(double)) {//метод трапеций

double h = (b - a) / n;

double sumFunc = 0;

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

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

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

}

double methodParabola(double a, double b, int n, double(\*f)(double)) {//метод парабол

double h = (b - a) / (2 \* n);

double sumFunc1 = 0, sumFunc2 = 0;

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

sumFunc1 += f(a + (2 \* i - 1) \* h);

sumFunc2 += f(a + 2 \* i \* h);

}

sumFunc1 += f(a + (2 \* n - 1) \* h);

return h / 3 \* (f(a) + 4 \* sumFunc1 + 2 \* sumFunc2 + f(b));

}

double methodGauss(double a, double b, int n, double(\*f)(double)) {//метод Гауса

double h = (b - a) / 2;

double h0 = (b + a) / 2;

double sumFunc = 0;

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

sumFunc += matrConst[i][n].a \* f(h0 + h \* matrConst[i][n].t);

return h \* sumFunc;

}

int main() {

setlocale(0, "Rus");

initMatrConst(matrConst);

//инициализация указателя на функции

double(\*f1) (double) = &function1;

double(\*f2) (double) = &function2;

double(\*f3) (double) = &function3;

printf("метод Гаусса\n");

for (int i = 1; i < 4; i++) {

printf("n = %d\n", i);

double res1 = methodGauss(0, 3, i, f1);

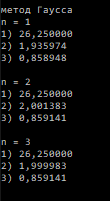
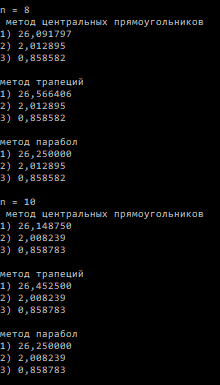
double res2 = methodGauss(0, PI, i, f2);

double res3 = methodGauss(0, 1, i, f3);

printf("1) %lf\n2) %lf\n3) %lf\n\n", res1, res2, res3);

}

return 0;

}