

Лабораторная работа

Численное интегрирование

Цель: составить программу, которая реализует методы численного интегрирования с постоянным и переменным шагом для определенного интеграла, а также программу для вычисления кратного (двойного) интеграла, и оформить меню для этих реализаций

Оборудование: ПК, компилятор Code::Blocks, язык программирования Си.

Определенный интеграл:

1.
$$\int_0^1 e^{-x^2} dx = \frac{1}{2} \sqrt{\pi} \operatorname{erf}(1) \approx 0.746824$$

2.
$$\int_{0.5}^{1.5} x^5 dx = 1.89583$$

Кратный интеграл:

$$\int_0^{\pi} \int_0^{\pi} \sin(x+y) dx dy = 1$$

Имя	Тип	Смысл
x, y	float	переменные подынтегральных функций
n	float	количество разбиений
h, hx, hy, hv, hs	float	шаг
sum, sum1, sum2, sumx, sumy	float	сумма
fa, fb, fx, da, db	float	дополнительные переменные
a, b, c, d	float	пределы
I, In, I2n	float	результатирующая значение интеграла
gl, sh	float	переменные меню
q	float	специальный коэффициент
M1, M2, M	float	коэффициент максимума производной функции
r	float	остаточный член
i	int	параметр цикла
e	float	точность
k	float	счетчик

Код программы:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <locale.h>
float f(float x);
float df(float x);
float ff(float x,float y);
float trapf(float n,float a,float b);
void glm();
void integ1();
void integ2();
void shag1();
void shag2();
void left();
void right();
void trap();
void parab();
void metod1();
void metod2();
int end();

int main()
{
    setlocale(LC_ALL,"");
    glm();
    getchar();
    return 0;
}

float f(float x)
{
    float y;
    y=exp((-1)*pow(x,2)); /*[0;1]*/
    /*y=pow(x,5); /*[0.5;1.5]*/
    return y;
}

float ff(float x,float y)
{
    float r;
    r=sin(x+y); /*[0;1.57],[0;0.785]*/
```

```

    return r;
}
float df(float x)
{
    float h,dy;
    h=pow(10,-6);
    dy=(f(x+h)-f(x))/h;
    return dy;
}
float trapf(float n,float a,float b)
{
    float h,sum=0,x,I,fa,fb,fx;
    h=(b-a)/n;
    fa=f(a);
    fb=f(b);
    fx=(fa+fb)/2;
    x=a+h;
    while (x<=b-h)
    {
        sum=sum+f(x);
        x=x+h;
    }
    I=h*(fx+sum);
    return I;
}
void glm()
{
    int gl;
    printf("Главное меню\n\n1. Определенный интеграл\n2. Кратный
интеграл\n3. Выход из программы\n\nВыберите пункт меню: ");
    scanf("%d",&gl);
    switch(gl)
    {
        case 1: integ1(); break;
        case 2: integ2(); break;
        case 3: end(); break;
    }
}
void integ1()
{
    int gl;

```

```

printf("\nОпределенный интеграл\n\n1. Методы с постоянным шагом\n2.
Методы с переменным шагом\n3. Вернуться в главное меню\n\nВыберите
пункт меню: ");
scanf("%d",&gl);
switch(gl)
{
    case 1: shag1(); break;
    case 2: shag2(); break;
    case 3: glm(); break;
}
}
void integ2()
{
    float x,y,a,b,c,d,n,hx,hy,sumx=0,sumy=0,I,q;
    printf("\nВведите внешние пределы интегрирования\nНижний предел (a):
");
    scanf("%f",&a);
    printf("Верхний предел (b): ");
    scanf("%f",&b);
    printf("Введите внутренние пределы интегрирования\nНижний предел (c):
");
    scanf("%f",&c);
    printf("Верхний предел (d): ");
    scanf("%f",&d);
    printf("Введите количество разбиений: ");
    scanf("%f",&n);
    hx=(b-a)/n;
    hy=(d-c)/n;
    x=a;
    while (x<=b)
    {
        sumy=0;
        y=c;
        while(y<=d)
        {
            if(x==a || x==b)
                if(y==c || y==d)
                    q=0.25;
                else q=0.5;
            else if(y==c || y==d)
                q=0.5;
            else q=1;

```

```

        sumy+=q*ff(x,y);
        y+=hy;
    }
    sumx+=sumy;
    x+=hx;
}
I=hx*hy*sumx;
printf("Значение двойного интеграла: %.8f\n\n",I);
glm();
}
void shag1()
{
    int sh;
    printf("\nМетоды с постоянным шагом\n\n1. Метод левых частей
прямоугольников\n2. Метод правых частей прямоугольников\n3. Метод
трапеции\n4. Метод парабол\n5. Вернуться в главное меню\n\nВыберите
пункт меню: ");
    scanf("%d",&sh);
    switch(sh)
    {
        case 1: left(); break;
        case 2: right(); break;
        case 3: trap(); break;
        case 4: parab(); break;
        case 5: glm(); break;
    }
}
void shag2()
{
    int sh;
    printf("\nМетоды с переменным шагом\n\n1. Метод 1\n2. Метод 2\n3.
Вернуться в главное меню\n\nВыберите пункт меню: ");
    scanf("%d",&sh);
    switch(sh)
    {
        case 1: metod1(); break;
        case 2: metod2(); break;
        case 3: glm(); break;
    }
}
void left()
{

```

```

float h,sum=0,x,I,M1,M2,M,r,a,b,n;
printf("\nВведите пределы интегрирования\nНижний предел: ");
scanf("%f",&a);
printf("Верхний предел: ");
scanf("%f",&b);
printf("Введите количество разбиений: ");
scanf("%f",&n);
M1=fabs(df(a));
M2=fabs(df(b));
if (M1>M2)
    M=M1;
else M=M2;
r=fabs((M*pow(b-a,2))/(2*n));
h=(b-a)/n;
x=a;
while (x<=b-h)
{
    sum=sum+f(x);
    x=x+h;
}
I=h*sum;
printf("Значение интеграла: %.8f\nЗначение остаточного члена: %g\n\n",I,r);
glm();
}
void right()
{
    float n,h,sum=0,x,I,M1,M2,M,r,a,b;
    printf("\nВведите пределы интегрирования\nНижний предел: ");
    scanf("%f",&a);
    printf("Верхний предел: ");
    scanf("%f",&b);
    printf("Введите количество разбиений: ");
    scanf("%f",&n);
    M1=fabs(df(a));
    M2=fabs(df(b));
    if (M1>M2)
        M=M1;
    else M=M2;
    r=fabs((M*pow(b-a,2))/(2*n));
    h=(b-a)/n;
    x=a+h;
    while (x<=b)

```

```

    {
        sum=sum+f(x);
        x=x+h;
    }
    I=h*sum;
    printf("Значение интеграла: %.8f\nЗначение остаточного члена: %g\n\n",I,r);
    glm();
}
void trap()
{
    float n,h,sum=0,x,I,fa,fb,fx,da,db,M1,M2,M,r,a,b;
    printf("\nВведите пределы интегрирования\nНижний предел: ");
    scanf("%f",&a);
    printf("Верхний предел: ");
    scanf("%f",&b);
    printf("Введите количество разбиений: ");
    scanf("%f",&n);
    da=df(a);
    M1=fabs(df(da));
    db=df(b);
    M2=fabs(df(db));
    if (M1>M2)
        M=M1;
    else M=M2;
    r=sqrt(fabs((M*pow(b-a,3))/(12*n)));
    h=(b-a)/n;
    fa=f(a);
    fb=f(b);
    fx=(fa+fb)/2;
    x=a+h;
    while (x<=b-h)
    {
        sum=sum+f(x);
        x=x+h;
    }
    I=h*(fx+sum);
    printf("Значение интеграла: %.8f\nЗначение остаточного члена: %g\n\n",I,r);
    glm();
}
void parab()
{
    float h,sum1=0,sum2=0,x,I,fa,fb,n,da,db,M,r,a,b;

```

```

int i;
printf("\nВведите пределы интегрирования\nНижний предел: ");
scanf("%f",&a);
printf("Верхний предел: ");
scanf("%f",&b);
printf("Введите количество разбиений: ");
scanf("%f",&n);
da=f(a);
db=f(b);
for (i=0;i<4;i++)
{
    da=df(da);
    db=df(db);
}
if (fabs(da)>fabs(db))
    M=da;
else M=db;
r=sqrt(sqrt(fabs((pow(b-a,5)*M)/(2880*n))));
h=(b-a)/(2*n);
fa=f(a);
fb=f(b);
for (i=1;i<=(2*n-1);i+=2)
{
    x=a+i*h;
    sum1=sum1+f(x);
}
for (i=2;i<=(2*n-2);i+=2)
{
    x=a+i*h;
    sum2=sum2+f(x);
}
I=(h/3)*(fa+fb+4*sum1+2*sum2);
printf("Значение интеграла: %.8f\nЗначение остаточного члена: %g\n\n",I,r);
glm();
}
void metod1()
{
    float n=2,In,I2n,e,a,b,k=1;
    printf("\nВведите пределы интегрирования\nНижний предел: ");
    scanf("%f",&a);
    printf("Верхний предел: ");
    scanf("%f",&b);

```



```

printf("Введите точность для вычисления интеграла: ");
scanf("%f",&e);
In=trapf(n,a,b);
I2n=trapf(2*n,a,b);
while (fabs(I2n-In)>e)
{
    n*=2;
    In=trapf(n,a,b);
    I2n=trapf(2*n,a,b);
    k++;
}
printf("Значение интеграла: %.8f\nКоличество итераций: %g\n\n",I2n,k);
glm();
}
void metod2()
{
    float a,b,e,n=2,In,I2n,fa,fb,fx,hv,hs,x,sum=0,k=1;
    printf("\nВведите пределы интегрирования\nНижний предел: ");
    scanf("%f",&a);
    printf("Верхний предел: ");
    scanf("%f",&b);
    printf("Введите точность для вычисления интеграла: ");
    scanf("%f",&e);
    fa=f(a);
    fb=f(b);
    fx=(fa+fb)/2;
    hv=(b-a)/n;
    x=a+hv;
    while (x<=b-hv)
    {
        sum+=f(x);
        x+=hv;
    }
    In=hv*(fx+sum);
    hv=hv/2;
    x=a+hv;
    while (x<=b-hv)
    {
        sum+=f(x);
        x+=hv;
    }
    I2n=hs*(fx+sum);

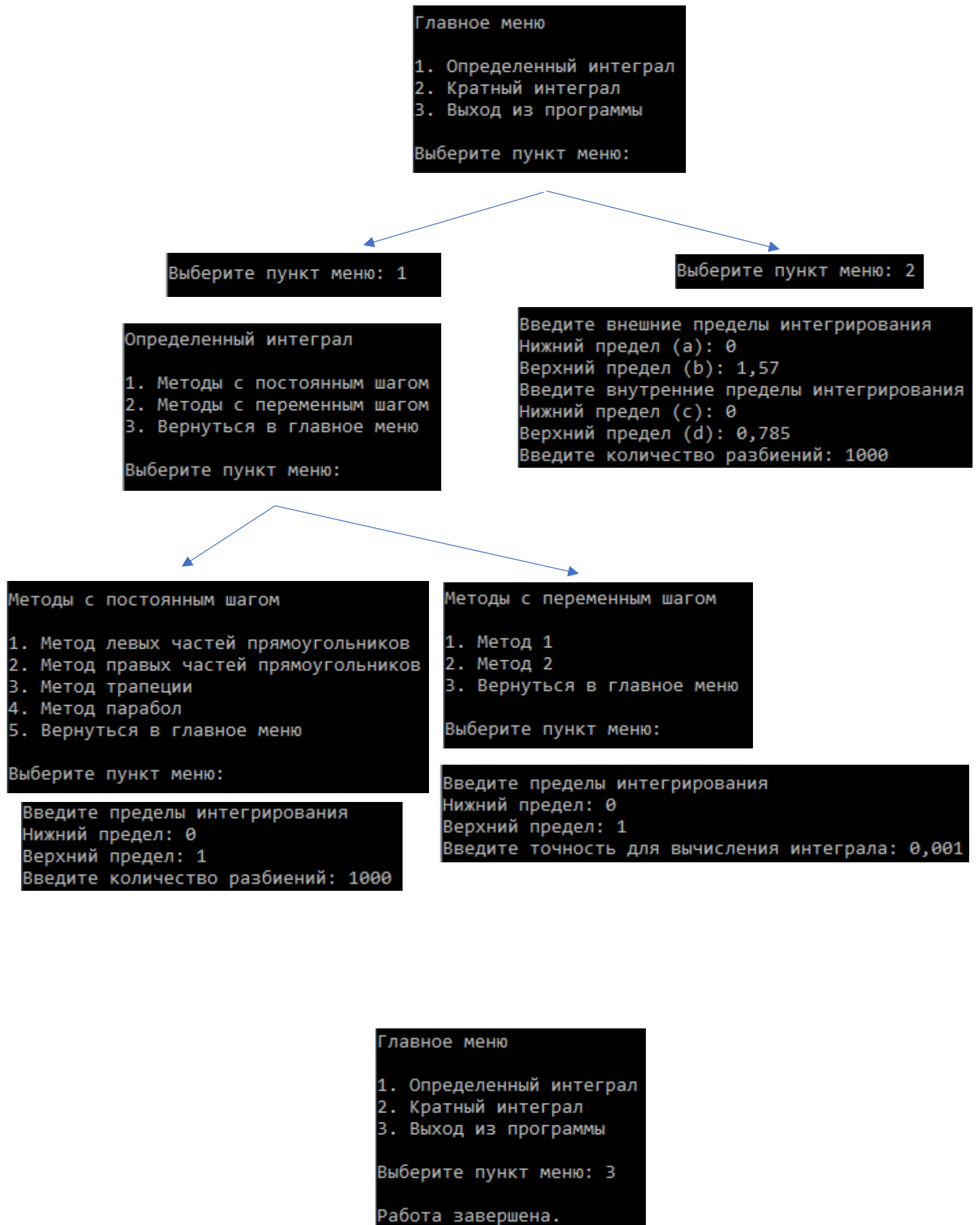
```

```

while (fabs(I2n-In)>e)
{
    hv=hs;
    In=I2n;
    hs=hv/2;
    x=a+hs;
    while (x<=b-hs)
    {
        sum+=f(x);
        x+=hv;
    }
    I2n=hs*(fx+sum);
    k++;
}
printf("Значение интеграла: %.8f\nКоличество итераций: %g\n\n",I2n,k);
glm();
}
int end()
{
    printf("\nРабота завершена.\n");
    return 0;
}

```

Реализация меню.



Результаты.

Метод	Результат
Определенный интеграл 1	
<i>Методы с постоянным шагом (количество разбиений - 1000)</i>	
Метод левых частей прямоугольников	Значение интеграла: 0,74714291 Значение остаточного члена: 0,000357628
Метод правых частей прямоугольников	Значение интеграла: 0,74651080 Значение остаточного члена: 0,000357628
Метод трапеций	Значение интеграла: 0,74682689 Значение остаточного члена: 0,00863167
Метод парабол	Значение интеграла: 0,74682295 Значение остаточного члена: 0,0232009
<i>Методы с переменным шагом (точность – 0,001)</i>	
Метод 1	Значение интеграла: 0,74658459 Количество итераций: 3
Метод 2	Значение интеграла: 0,74658459 Количество итераций: 3
Определенный интеграл 2	
<i>Методы с постоянным шагом (количество разбиений - 1000)</i>	
Метод левых частей прямоугольников	Значение интеграла: 1,88453484 Значение остаточного члена: 0,0121593
Метод правых частей прямоугольников	Значение интеграла: 1,89207244 Значение остаточного члена: 0,0121593
Метод трапеций	Значение интеграла: 1,88831615 Значение остаточного члена: 15,8114
Метод парабол	Значение интеграла: 1,89583409 Значение остаточного члена: 2,42746e-008
<i>Методы с переменным шагом (точность – 0,001)</i>	
Метод 1	Значение интеграла: 1,89596021 Количество итераций: 6
Метод 2	Значение интеграла: 1,89596009 Количество итераций: 6
Кратный (двойной) интеграл	
	Значение двойного интеграла: 0,99998343

Вывод.

Каждый из численных методов интегрирования имеет свои достоинства и недостатки, и для каждого интеграла есть более подходящие и менее подходящие методы. Для данных определенных интегралов наилучшим методом с постоянным шагом по результатам вычислений является метод парабол. А в методах с переменным шагом была выдержана заданная небольшая точность, поэтому здесь было важно время выполнения программы. Но в данных случаях мы не сможем его оценить, так как количество итераций довольно мало.

Также можно оценить точность вычислений с помощью остаточного члена: чем он меньше, тем точнее результат. С его помощью можно улучшить каждый из методов с постоянным шагом и добиться достаточно высокой точности вычислений.