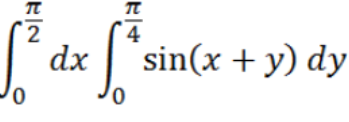
**Лабораторная работа 1.**

**Численное интегрирование.**

Цель работы - создать программу для нахождения интеграла методами: прямоугольников правых и левых частей, трапеции, парабол, двойного пересчета, пересчета. А также поиск кратного интеграла .

Инструментарий: язык си.

Используемые переменные:

|  |  |  |
| --- | --- | --- |
| Имя | Значение | Тип |
| \_A | Левая граница интеграла | const |
| \_B | Правая граница интеграла | const |
| \_N | Количество шагов разбиений | const |
| \_E | Точность вычисления | const |
| a | Левая граница интеграла | float |
| b | Правая граница интеграла | float |
| c | Левая граница интеграла кратного интеграла | float |
| d | Правая граница интеграла кратного интеграла | float |
| x | Счетчик цикла, текущее значение X | float |
| y | Счетчик цикла, текущее значение Y | float |
| h | Шаг движения | float |
| s | Накопленная сумма | float |
| i | Счетчик цикла, текущее значение X | float |
| r | Остаточный член | float |
| in | Значение интеграла | float |
| in1 | Значение первого интеграла | float |
| in2 | Значение второго интеграла | float |
| hv | Шаг вычисления | float |
| hs | Шаг смещения | float |
| hd | Шаг движения | float |
| hx | Шаг по x | float |
| hy | Шаг по y | float |
| sx | Сумма вычислений по x | float |
| sy | Сумма вычислений по y | float |
| nx | Количество шагов разбиений по x | float |
| ny | Количество шагов разбиений по y | float |
| ix | Интеграл после прохода и вычисления по x | float |
| iy | Интеграл после прохода и вычисления по y | float |
| sum1 | Накопленная сумма для первого интеграла | float |
| sum2 | Накопленная сумма для второго интеграла | float |
| choice | Выбор который делает пользователь | int |

Код программы на языке С:

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#define \_A 0

#define \_B 5

#define \_N 200

#define \_E 0.0001

#define e 2.71828

void mainmenu();

void first\_below\_menu();

void second\_below\_menu();

float integr(float x){

return x\*x\*x;

//return sin(pow(x, e))/4; //подинтегральная функция

}

double dif(double x){

//1/4\*e\*pow(x, 0.71828)\*((1.71828)\*cos(pow(x,e))-e\*pow(x,e)\*sin(pow(x,e)))

return 0.471\*(1.6651-0.678\*0.246854);

}

void left\_rectangle(){

float a = \_A, b = \_B, h, n = \_N, s = 0, r;

h = (a+b)/n;

r = fabs(pow(b-a,2)/(2\*n)\*0.4151);

printf("r = %g\n", r);

for(float i = a; i <= b - h; i = i + h)

{

s += integr(i);

}

s \*= h;

printf("%f\n", s);

first\_below\_menu();

}

void right\_rectangle(){

float a = \_A, b = \_B, h, n = \_N, s = 0, r;

r = fabs(pow(b-a,2)/(2\*n)\*0.4151);

printf("r = %g\n", r);

h = (a+b)/n;

for(float i = a+h; i <= b; i = i + h)

{

s += integr(i);

}

s \*= h;

printf("%f\n", s);

first\_below\_menu();

}

void trapeze(){

float a = \_A, b = \_B, h, n = \_N, s = 0, r;

r = fabs(pow(b-a,3)/(12\*n\*n)\*dif(0.6));

printf("r = %g\n", r);

h = (a+b)/n;

for(float i = a+h; i <= b; i = i + h)

{

s += (integr(i) + integr(i-h))/2;

}

s \*= h;

printf("%f\n", s);

first\_below\_menu();

}

void parabol(){

float a = \_A, b = \_B, h, n = \_N, s = 0, r;

r = fabs(pow(b-a,5)/(720\*n\*n)\*0.8775);

printf("r = %g\n", r);

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

n = 0;

for(float i = a+h/2; i <= b-h/2; i = i + h)

{

s += integr(i);

}

s \*= 4;

for(float i = a+h; i <= b-h; i = i + h)

{

n += integr(i);

}

s += integr(a)+integr(b)+2\*n;

s \*= h/6;

printf("%f\n", s);

first\_below\_menu();

}

void double\_count(){

float i, a = \_A, b = \_B, h = sqrt(\_E), n = \_N, in, in2;

double r;

r = fabs(pow(b-a,3)/(12\*n\*n)\*dif(0.6));

printf("r = %g\n", r);

do{

in = 0;

in2 = 0;

for(i= a+h; i <= b; i = i + h)

{

in += (integr(i) + integr(i-h))/2;

}

in \*= h;

h /= 2;

for(i = a+h; i<= b; i = i + h)

{

in2 += (integr(i) + integr(i-h))/2;

}

in2 \*= h;

}while(fabs(in-in2)>\_E);

printf("%f\n", in2);

second\_below\_menu();

}

void count(){

float i, a = \_A, b = \_B, hv = sqrt(\_E), n = \_N, in1, in2, r, hs = 0, hd, sum1, sum2;

r = fabs(pow(b-a,3)/(12\*n\*n)\*dif(0.6));

printf("r = %g\n", r);

do{

sum1 = 0;

sum2 = 0;

hd = hv;

for(i = a + hs; i < b; i = i + hd)

{

sum1 += (integr(i) + integr(i+hv))/2;

}

in1 = sum1 \* hv;

hs = hv / 2;

for(i= a + hs; i < b; i = i + hd)

{

sum2 += (integr(i) + integr(i+hv))/2;

}

in2 = sum2 \* hv;

hs = hv /2;

hv = hv / 2;

}while(fabs(in2-in1)>=\_E);

printf("%f\n", in2);

second\_below\_menu();

}

void crat(){

float b = 3.141592/2, a = 0, d = 3.141592/4, c = 0,

nx = 10000, ny = 10000, hx, hy, sx, sy, iy, ix, x, y;

hx = (b-a)/nx;

hy = (d-c)/ny;

printf("hx = %g, hy = %g\n", hx, hy);

sx = 0;

for(x = a; x<=b-hx; x += hx){

sy = 0;

for(y = c; y<=d-hy; y += hy){

sy += fabs(sinf(x+y));

// printf("%f\n", sy);

}

iy = hy\*sy;

sx += iy;

}

ix = hx\*sx;

printf("%f\n", ix);

mainmenu();

}

void first\_below\_menu(){

int choice;

printf("Методы с постоянным шагом:\n 1. Метод правых частей прямоугольников\n 2. Метод левых частей прямоугольников\n 3. Метод трапеций\n 4. Метод парабол\n 5. Возврат в главное меню\n");

scanf("%d", &choice);

switch (choice){

case 1: right\_rectangle();

break;

case 2: left\_rectangle();

break;

case 3: trapeze();

break;

case 4: parabol();

break;

case 5: mainmenu();

break;

default: printf("err: try again\n");

first\_below\_menu();

}

}

void second\_below\_menu(){

int choice;

printf("Методы с переменным шагом:\n 1. Метод двойного пересчета\n 2. Метод пересчета\n 3. Возврат в главное меню\n");

scanf("%d", &choice);

switch (choice){

case 1: double\_count();

break;

case 2: count();

break;

case 3: mainmenu();

break;

default: printf("err: try again\n");

second\_below\_menu();

}

}

void mainmenu(){

int choice;

printf("Главное меню:\n 1. Методы с постоянным шагом\n 2. Методы с переменным шагом\n 3. Вычисление кратного интеграла\n 4. Выход из программы\n");

scanf("%d", &choice);

switch (choice){

case 1: first\_below\_menu();

break;

case 2: second\_below\_menu();

break;

case 3: crat();

break;

case 4: exit(0);

break;

default: printf("err: try again\n");

mainmenu();

}

}

int main(){

mainmenu();

return 0;

}

Приведенные в таблицах ниже значения были получены с Ep = 0.0001 и n = 200.

Табличный интеграл = 4:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Метод прямоугольников правых частей | Метод прямоугольников левых частей | Метод трапеции | Метод парабол | Метод двойного пересчета | Метод пересчета |
| 4.040091 | 3.960091 | 4.000092 | 3.999993 | 4.000015 | 3.996167 |

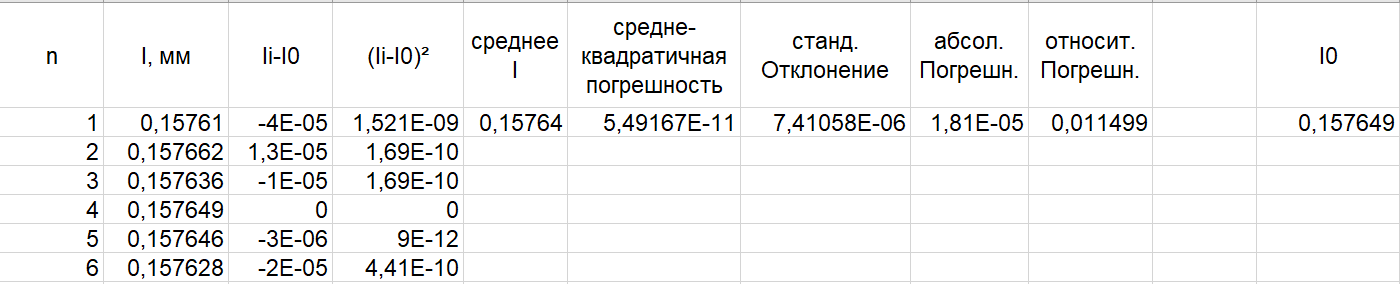
Наиболее точным методом при заданных Ep и n оказался метод парабол с погрешностью в . Хуже всего себя показал метод прямоугольников с погрешностью в 0.04.

Таблица полученных результатов для интеграла

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Метод прямоугольников правых частей | Метод прямоугольников левых частей | Метод трапеции | Метод парабол | Метод двойного пересчета | Метод пересчета |
| 0.157610 | 0.157662 | 0.157636 | 0.157649 | 0.157646 | 0.157628 |

Результат всех измерений оказался точен до . Наиболее точными при заданных значениях Ep и n оказались методы парабол и двойного пересчета. Для достижения наибольшей точность необходимо правильно подобрать Ep и увеличить количество разбиений заданного участка интегрирования.

Таблица погрешности измерений:

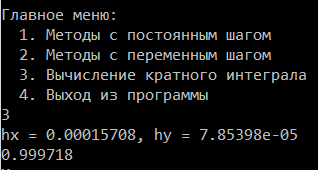


Остаточный член r для метода прямоугольниками: 0.00103775.

Остаточный член r для метода трапециями: 1.46965e-06.

Остаточный член r для метода параболами: 3.04688e-08.

Вычисление кратного интеграла:



Вывод: в результате выполнения лабораторной работы был подсчитан интеграл и погрешность измерения, результат работы представлен программой, позволяющий найти интеграл различными методами.