МИНОБРНАУКИ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

Тульский государственный университет

Кафедра прикладной математики и информатики

Лабораторная работа № 2

по дисциплине: «Численные методы»

на тему:

«Приближённое вычисление кратных интегралов методом последовательного интегрирования »

Выполнил студент группы 221241: Новиков А.В.

Принял проф. каф. ПМиИ : Баранов В.П.

Тула 2017

I. ЦЕЛЬ РАБОТЫ

Приобретение навыков приближенного вычисления кратных интегралов.

II. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Построим кубатурную формулу, предназначенную для приближенного вычисления двойных интегралов



Рассмотрим метод последовательного интегрирования.

Пусть область интегрирования  представляет собой прямоугольник  Разобьем область  на прямоугольные ячейки (рис. 1).



Рис. 1.

оо

ю

**ююю**

Интеграл  вычислим последовательным интегрированием:



где



Каждый однократный интеграл вычисляется на выбранной сетке по какой-либо квадратурной формуле типа



Последовательное интегрирование по обоим направлениям приводит к кубатурным формулам, которые являются прямым произведением одномерных квадратурных формул



или



где 

Для разных направлений можно использовать квадратурные формулы одного или разных порядков точности.

Если область  имеет произвольную форму, то в  следует провести хорды, параллельные оси  (рис. 2).



Рис. 2

На каждой хорде выбираются узлы так, как нам требуется. Интеграл  представим в виде



Сначала вычислим интеграл по  вдоль каждой хорды по какой-либо квадратурной формуле, используя введенные узлы. Затем вычислим интеграл по , используя какую-либо квадратурную формулу. При этом узлами будут служить проекции хорд на ось ординат.

III. ЗАДАНИЕ

Методом последовательного интегрирования вычислить интеграл  по области , изображенной на рис. 3.



Рис. 3

|  |  |
| --- | --- |
| 1 |  |

**Текст программы:**

#include <conio.h>

#include <iostream>

#include <cmath>

#include <windows.h>

#include <stdio.h>

#include <math.h>

using namespace std;

int nIn = 0;

int nOut = 0;

double X1 = 0;

double X2 = 1;

double Y1 = 0;

double Y2 = 1;

void readN()

double f(double x, double y){

return x\*x\*x\*sin(x+y);

}

double solveBy\_Simpson(double n, double a, double b, double x0){

double h = (b-a)/n;

double summ = 0;

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

double pLeft = a+(h\*(double)i);

double pRight = a+(h\*(double)(i+1));

double fun = (pLeft+pRight)/2;

double c = f(x0,fun)\*h;

summ += c;

}

return summ;

}

double solveBy\_StepStrip(double nOuter, int nInner){

double stepX = (X2-X1)/nOuter;

double summ = 0.0;

double x0 = X1;

for(int i = 0; i <= nOuter-1; i++){

summ += solveBy\_Simpson(1,Y1,Y2,x0);

x0 += stepX;

}

return summ;

}

int main(){

readN();

printf("Value is : %f",solveBy\_StepStrip(nOut,nIn));

\_getch();

return 0;

}

**Результат работы программы:**

