Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Петрозаводский государственный университет»  
Физико-технический институт  
Направление Информатика и вычислительная техника. Проектирование информационных систем в экономике

ОТЧЁТ  
по лабораторной работе №1  
**«Интеграл»**

Автор работы:

студентка группы 21218

Э.В. Таничева

«15» сентября 2022 г.

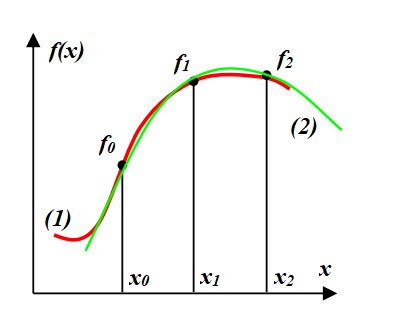
**Формулировка задачи:**

Составить программу на языке Программирования высокого уровня, вычисляющую определенный интеграл методом Симпсона с точностью 0.00001. С помощью любой экспертной системы найти решение этого интеграла и сравнить полученные результаты.

 **ВАРИАНТ 13**

**Сведения о численном методе, которым будет решаться задача:**

Суть метода Симпсона заключается в приближении подынтегральной функции на отрезке [a,b] интерполяционным многочленом второй степени *P(x),*т.е. приближение графика функции на отрезке параболой, проходящей через три узла, например, как показано на рисунке ((1) – функция, (2) ­– полином).



Метод Cимпсона

**Идея метода:**

Рассмотрим три узла *x*0, *x*1, *x*2, и проведем через них параболу, с помощью уравнения Ньютона:

Заменим *x* – *x*0 *на z*

Далее, сосчитаем интеграл по данному интервалу:

Получаем

Для *равномерной сетки* и *четного числа шагов* *n* формула Симпсона имеет вид:

**,** 

**Блок-схема:**

да

нет

да

нет

**Листинг программы:**

**program lab1;**

**uses crt;**

**function F(x:real):real;**

**begin**

**F:=sqrt(1-cos(2\*x));**

**end;**

**var**

**a,pervyi,vtoroi,razbit,i:integer;**

**tochno,shag,slshag,Fun,S,S2: real;**

**begin**

**S2:=0;**

**S:=0 ;**

**pervyi:=0;**

**vtoroi:=314;**

**a:=1;**

**tochno:=0.00001;**

**razbit:=8000;**

**clrscr;**

**while a=1 do**

**begin**

**shag:=(vtoroi-pervyi)/razbit;** //считаем шаг

**i:=1;**

**slshag:=pervyi;**

**Fun:=F(slshag);** //обращаемся к функции

**S:=Fun;**

**while a=1 do** //пока a=1 выполнять

**begin**

**slshag:=slshag+shag;** //следующий шаг

**Fun:=F(slshag);**

**S:=S+4\*Fun;** //наращиваем интеграл

**i:=i+2;** //переход к следующему разбиению

**if i<razbit then** //пока количество разбиений больше текущего шага

**begin**

**slshag:=slshag+shag;**

**Fun:=F(slshag);**

**S:=S+2\*Fun;**

**end**

**else break;**

**end;**

**begin**

**slshag:=vtoroi;** //переходим к последнему пределу интегрирования

**Fun:=f(slshag);**

**S:=(S+Fun)\*(shag/3);** //считаем конечный результат

**if abs(S-S2)>tochno then** //если разность полученного интеграла и предыдущего больше точности

**Begin**

**S2:=s;** //предыдущий становится полученным

**razbit:=razbit+2;** //прибавляем еще 2 разбиения и возвращаемся в цикл

**end**

**else break;** //иначе выход

**end;**

**end;**

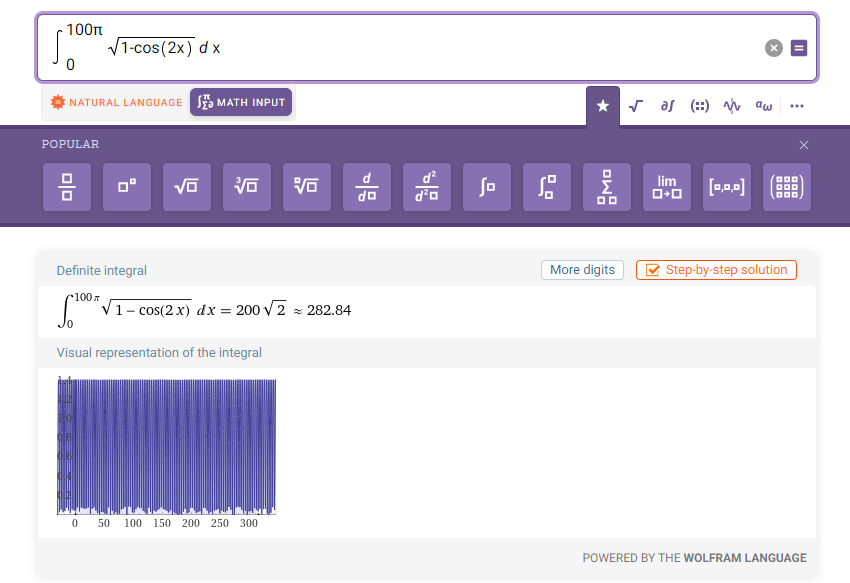
**writeln('Ответ: ',S:0:6);** //вывод ответа

**writeln('Количество разбиений = ',razbit);** //вывод общего количества разбиений

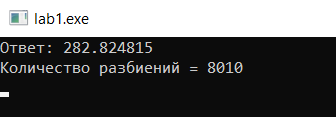
**repeat until keypressed;**

**end.**

**Результаты расчета в среде WolframAlpha:**



**Результаты работы программы:**



**Bывод:**

С помощью программы PascalABC я написала программу, которая считает данный интеграл с точностью 0,00001.

Данные которые получились у меня отличаются от данных полученные в WolframAlpha (в интернете являются более точными). Это связано с заданной точностью, если взять большее количество разбиений, то результаты будут совпадать.