Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное

образовательное учреждение высшего образования

Санкт-Петербургский университет аэрокосмического приборостроения КАФЕДРА № 2

ОТЧЕТ

ЗАЩИЩЕН С ОЦЕНКОЙ ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доц., канд. тех. наук |  |  |  | С.Л. Козенко |
| должность, уч. степень,  звание |
|  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ №4 |
| *“Численное интегрирование”* |
| по курсу: ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |
| --- | --- | --- |
| СТУДЕНТ ГР. № 4134к | подпись, дата | Столяров Н.С.  инициалы, фамилия |

Санкт-Петербург 2023

# Цель работы

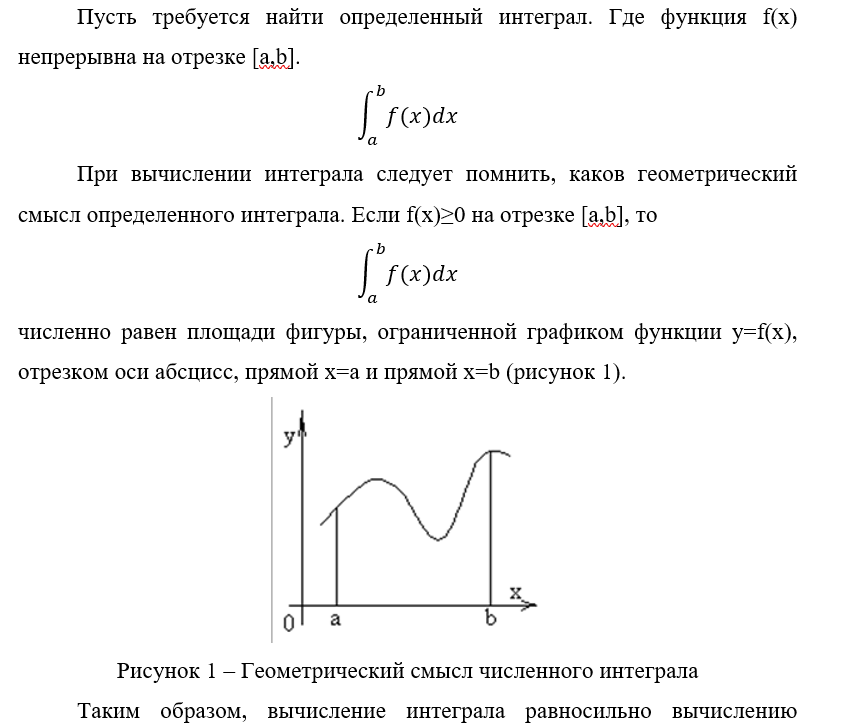
а) Освоение методов численного интегрирования;

б) совершенствование навыков по алгоритмизации и программированию вычислительных задач.

# Постановка задачи

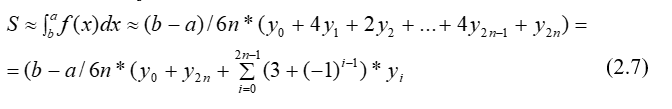


# Математическая часть



Геометрически иллюстрация формулы Симпсона состоит в том, что на каждом из сдвоенных частичных отрезков заменяем дугу данной кривой дугой графика квадратного тричлена.

Разобьем отрезок интегрирования *[a,b]* на *2n* равных частей длины *h=(b-a)/2n*. Обозначим точки разбиения *x0=a, x1=x0+h, … , xi=x0+i\*h, … , x2n=b*. Значения функции *f* в точках *xi* обозначим *yi, т.е. yi=f(xi).* Тогда согласно методу Симпсона:



формулаПусть функция  *y = f(x)* непрерывна на отрезке *[a,b]* и нам требуется вычислить определенный интеграл .

формулаформулаРазобьем отрезок *[a,b]* на *n* элементарных

отрезков   длины

точками . Пусть точки являются серединами отрезков  соответственно. В этом случае все "узлы" определяются из равенства .



### ***Суть метода парабол***

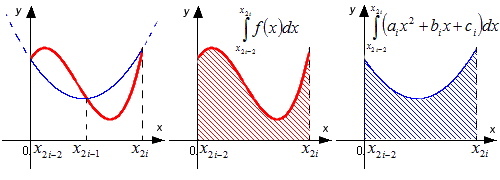
На каждом интервале  подынтегральная функция приближается квадратичной параболой , проходящей через точки . Отсюда и название метода - метод парабол.



формулаформулаЭто делается для того, чтобы в качестве приближенного значения определенного интеграла взять  , который мы

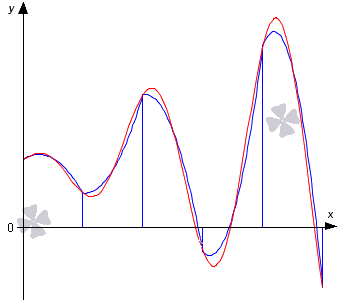
можем вычислить по формуле Ньютона-Лейбница. В этом и заключается  ***суть метода парабол***.

Геометрически это выглядит так (рис.2.4):



### **Рис. 2.4. Интервальное приближение**

Красной линией изображен график функции *y=f(x)*, синей линией показано приближение графика функции *y=f(x)* квадратичными параболами на каждом элементарном отрезке разбиения (рис.2.5).



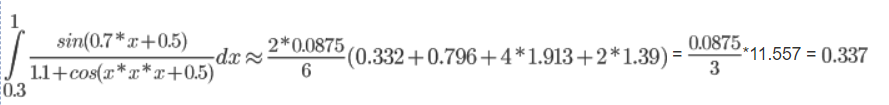
### **Рис. 2.5. Графическая иллюстрация метода парабол (Симпсона)**

# 4Аналитические расчеты

Для проведения аналитических расчетов был использован онлайн-калькулятор. На рисунке 2 представлен результат расчетов. На рисунке 3 представлена графическая иллюстрация решения.

Найдем максимальное значение четвертой производной функции на интервале [0.3;1].

|  |  |  |
| --- | --- | --- |
| i | xi | yi |
| 0 | 0.3 | 0.3318 |
| 1 | 0.388 | 0.3578 |
| 2 | 0.475 | 0.385 |
| 3 | 0.563 | 0.4148 |
| 4 | 0.65 | 0.4498 |
| 5 | 0.738 | 0.4941 |
| 6 | 0.825 | 0.5548 |
| 7 | 0.913 | 0.6459 |
| 8 | 1 | 0.7961 |



Найдем максимальное значение четвертой производной функции на интервале [0.3;1].

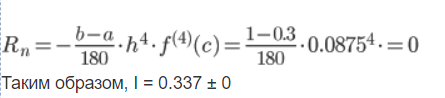


Рисунок 2 - результат расчетов

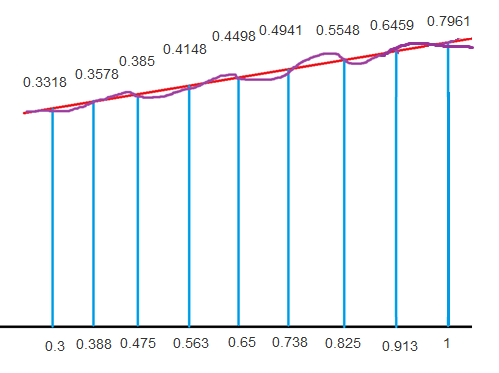


Рисунок 3 – графическая иллюстрация

Расчёты были проведены на сайте planetcalc.ru

# C:\Users\nikit\AppData\Local\Microsoft\Windows\INetCache\Content.Word\diagram.pngСхема алгоритма

# Текст программы на С++

#include<iostream>

#include<cmath>

**using** **namespace** std**;**

double f**(**double x**)** **{**

**return** **(**sin**(**0.7 **\*** x **+** 0.5**))** **/** **(**1.1 **+** cos**(**x **\*** x **\*** x **+** 0.5**));**

**}**

double simpsonIntegral**(**double a**,** double b**,** int n**)** **{**

const double width **=** **(**b **-** a**)** **/** n**;**

double simpson\_integral **=** 0**;**

**for** **(**int step **=** 0**;** step **<** n**;** step**++)** **{**

const double x1 **=** a **+** step **\*** width**;**

const double x2 **=** a **+** **(**step **+** 1**)** **\*** width**;**

simpson\_integral **+=** **(**x2 **-** x1**)** **/** 6.0 **\*** **(**f**(**x1**)** **+** 4.0 **\*** f**(**0.5 **\*** **(**x1 **+** x2**))** **+** f**(**x2**));**

**}**

**return** simpson\_integral**;**

**}**

// a = 0.3

// b = 1.0

// n = 8

int main**()** **{**

system**(**"chcp 65001"**);**

double a**,** b**;**

int n**;**

cout **<<** "Введите а: "**;**

cin **>>** a**;**

cout **<<** "Введите b: "**;**

cin **>>** b**;**

cout **<<** "Введите n: "**;**

cin **>>** n**;**

cout **<<** "Метод Симпсона" **<<** endl**;**

cout **<<** simpsonIntegral**(**a**,** b**,** n**);**

**return** 0**;**

**}**

Скриншоты с результатами программных расчетов

На рисунке 5 приведен результат работы программы.

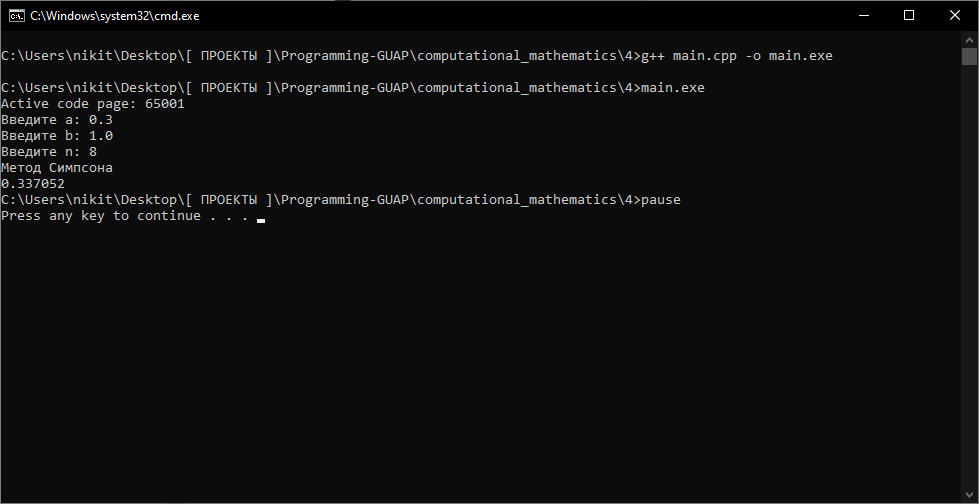


Рисунок 5 – Результат работы программы

Значение полученное аналитически не отличаются от значений, полученных программным путем.

# Сравнение результатов программных и аналитических расчётов

|  |  |
| --- | --- |
| Результат программы | Результат онлайн калькулятора |
| 0.337052 | 0.33707 |

Как видим значения не отличаются, значит программа работает корректно!

# Выводы

Я освоил методы численного интегрирования , усовершенствовал свои навыки по алгоритмизации и программированию вычислительных задач.