Командное задание «численное интегрирование» метод Симпсона

Студенты : Грачев Владимир

Быков Георгий

Зубаиров Тимур

Метод Симпсона (парабол)

- Задача нахождения точного значения определенного интеграла не всегда имеет решение. Действительно, первообразную подынтегральной функции во многих случаях не удается представить в виде элементарной функции. В этом случае мы не можем точно вычислить определенный интеграл по формуле Ньютона-Лейбница. Однако есть методы численного интегрирования, позволяющие получить значение определенного интеграла с требуемой степенью точности. Одним из таких методов является метод Симпсона (его еще называют методом парабол). Сначала выясним смысл метода парабол, дадим графическую иллюстрацию и выведем формулу для вычисления приближенного значения интеграла. Далее запишем неравенство для оценки абсолютной погрешности метода Симпсона (парабол).

Метод парабол (Симпсона) - суть метода, формула, оценка погрешности, иллюстрация.

- Пусть функция *y = f(x)* непрерывна на отрезке *[a; b]* и нам требуется вычислить определенный интеграл $\int_a^b f(x)dx$
- Разобьем отрезок *[a; b]* на *n* элементарных отрезков по формуле $h = \frac{(b-a)}{2n}$ – длина каждого из маленьких отрезков или **шаг**
- **Формула Симпсона** для приближенного вычисления определенного интеграла имеет следующий вид:
- – значения подынтегральной функции в точках

Метод парабол (Симпсона) - суть метода, формула, оценка погрешности, иллюстрация.

[illegible] – сумма первого и последнего значения подынтегральной функции;

[illegible] – сумма членов с *чётными* индексами умножается на 2;

[illegible] – сумма членов с *нечётными* индексами умножается на 4.

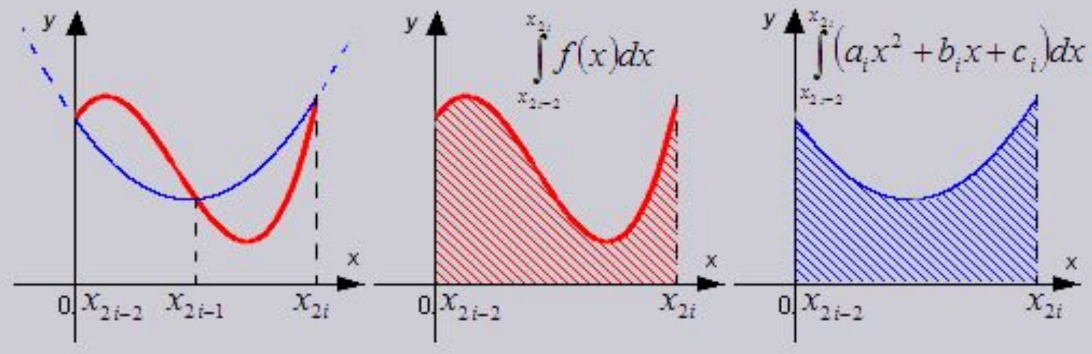
Метод Симпсона (метод парабол)Способ при котором – график подынтегральной функции приближается не ломаной линией, а маленькими параболами. Сколько промежуточных отрезков – столько и маленьких парабол. Если взять те же три отрезка, то метод Симпсона даст ещё более точное приближение, чем метод прямоугольников или метод трапеций.

Суть метода

На каждом интервале $[x_{2i-2}; x_{2i}]$, $i = 1, 2, \ldots, n$ подынтегральная функция приближается квадратичной параболой $y = a_i x^2 + b_i x + c_i$, проходящей через точки $(x_{2i-2}; f(x_{2i-2}))$, $(x_{2i-1}; f(x_{2i-1}))$, $(x_{2i}; f(x_{2i}))$. Отсюда и название метода - метод парабол.

Это делается для того, чтобы в качестве приближенного значения определенного интеграла $\int_{x_{2i-2}}^{x_{2i}} f(x)dx$ взять $\int_{x_{2i-2}}^{x_{2i}} (a_i x^2 + b_i x + c_i)dx$, который мы можем вычислить по формуле Ньютона-Лейбница. В этом и заключается **суть метода парабол**.

Геометрически это выглядит так:



Задание

- Интеграл $$\int_{0}^{2.5} \frac{1}{\sqrt[4]{625+x^4}} dx$$

- Решить его методом Симпсона с приближенным к 0.001 значению

Формула Симпсона:

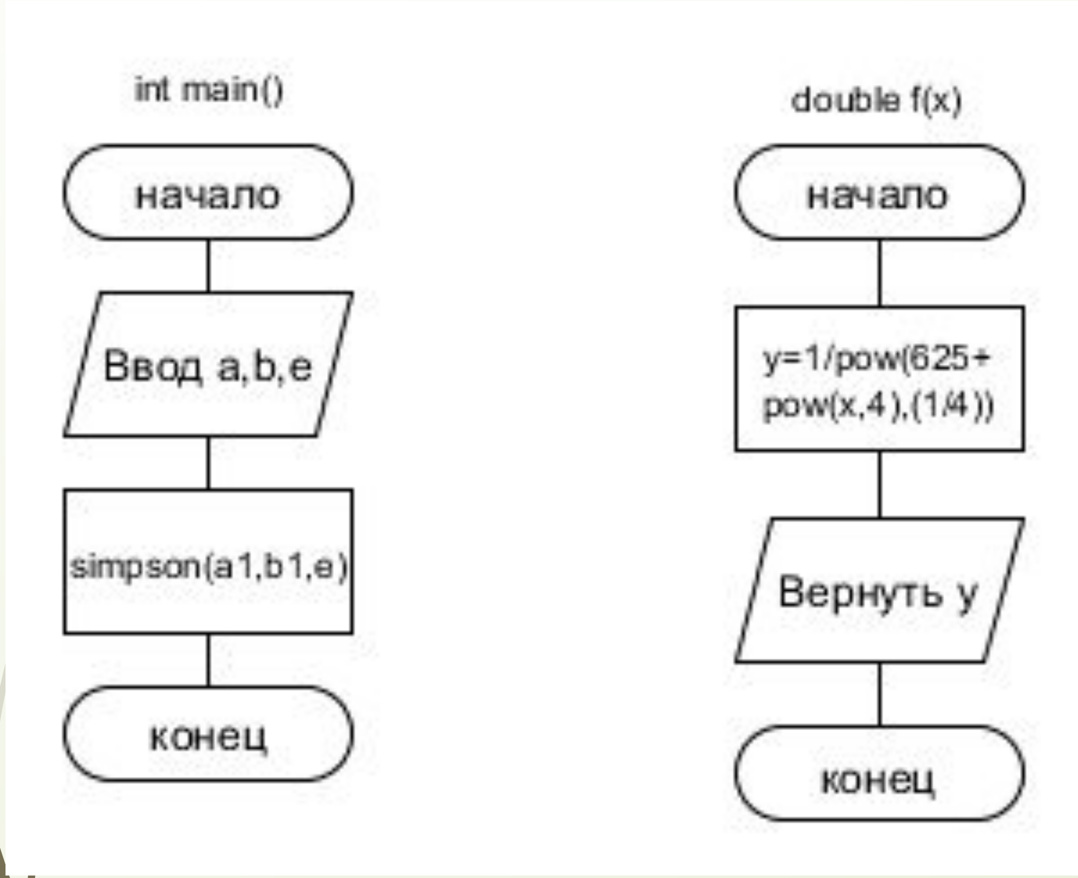
$$\int_a^b f(x)dx \approx \frac{2h}{6}[y_0 + y_n + 4(y_1 + y_3 + ... + y_{n-1}) + 2(y_2 + y_4 + ... + y_{n-2})]$$

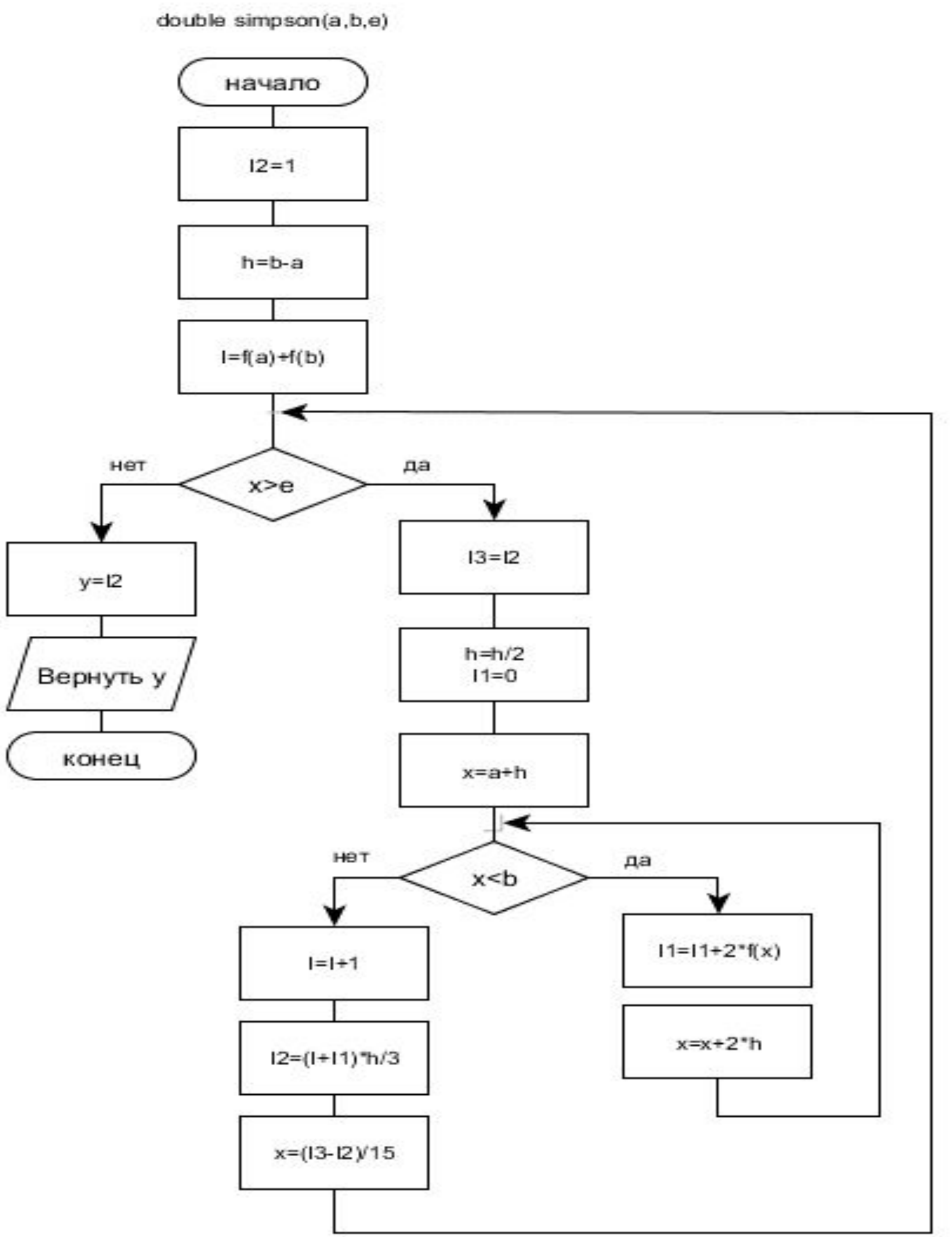
$$h = \frac{b-a}{n} = \frac{2.5-0}{10} = 0.25$$

i	x_i	y_i
0	0	0.2
1	0.25	0.2
2	0.5	0.2
3	0.75	0.2
4	1	0.1999
5	1.25	0.1998
6	1.5	0.1996
7	1.75	0.1993
8	2	0.1987
9	2.25	0.198
10	2.5	0.197

$$\int_0^{2.5} \frac{1}{(625+x^4)^{1/4}}dx \approx \frac{2 \cdot 0.25}{6}(0.2 + 0.197 + 4 \cdot 0.997 + 2 \cdot 0.798) = \frac{0.25}{3} \cdot 5.982 = 0.498$$

Графическая схема



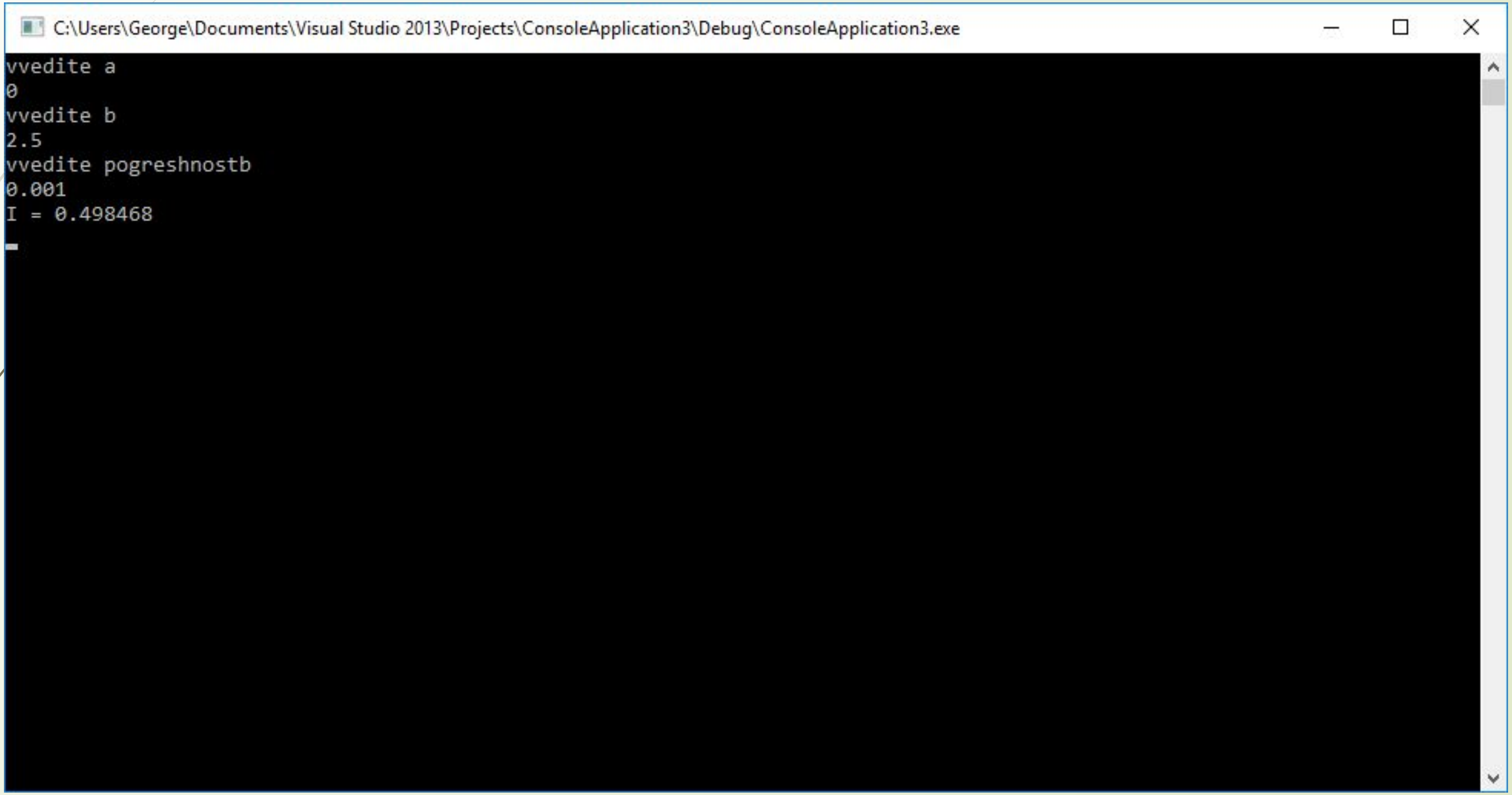


Код программы

```
#include<iostream>
#include<cmath>
#include<conio.h>
using namespace std;

//Производим описание нашей подинтегральной функции

double f(double x)
{
    double y;
    y = 1. / pow(625. + pow(x, 4.), (1. / 4.));
    return y;
}


double simpson(double a, double b, double e)
{
    double y, h, I, I1, I2, I3, x; // шаг, конечный результат, переменные
    I2 = 1.; //вводим начальные значения интегральной суммы
    h = b - a;
    I = f(a) + f(b);//вводим начальное значение шага

    do
    {
        I3 = I2;//вычисляем
        h = h / 2.;//уменьшаем шаг вдвое
        I1 = 0;
        x = a + h;//получаем новый x, идя с границы "a" с шагом h
        do
        {
            //подсчитаем сумму при определенном разбиении
            I1 = I1 + 2. * f(x);
            x = x + 2. * h;
        } while (x<b);//пока промежуточное значение меньше верхнего предела
        I = I + I1;
        I2 = (I + I1)*h / 3.;
        x = fabs(I3 - I2) / 15.;
    } while (x>e);//если промежуточное значение больше погрешности, то
    y = I2;
```

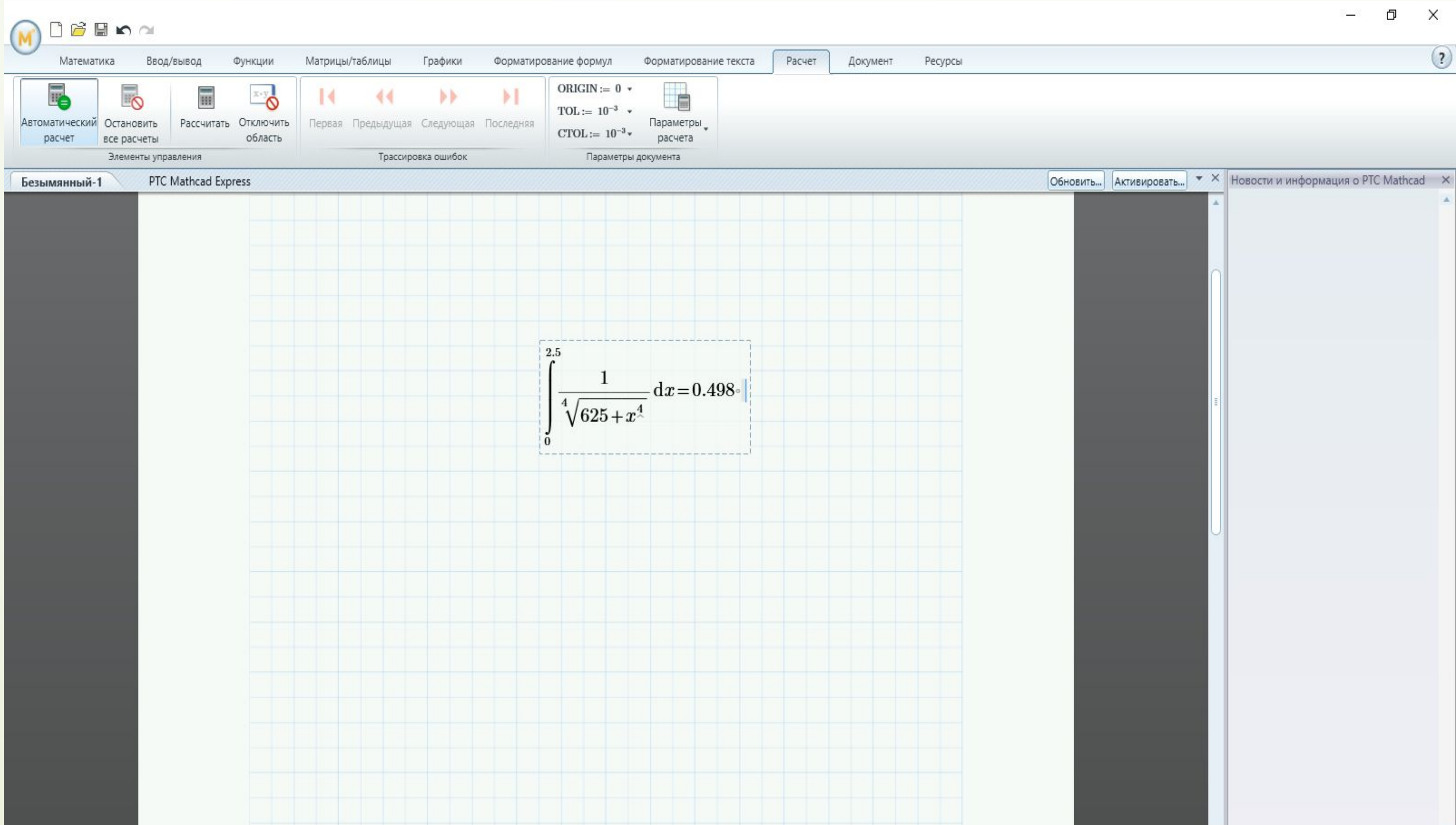
```
    return y;
}

//Описываем главную функцию
int main()
{
    double a1;//переменная для нижнего предела
    double b1;//переменная верхнего предела
    double e;//переменная для погрешности
    cout << "vvedite a" << endl;
    cin >> a1;
    cout << "vvedite b" << endl;
    cin >> b1;
    cout << "vvedite pogreshnostb" << endl;
    cin >> e;
    cout << "Integral Summa ravna I = " << simpson(a1, b1, e) << endl;
    getch();
    return 0;
```

Результат работы программы



Проверка с помощью сторонних программ

- MatCad:



Источники

- Википедия
- MathProfi.ru

Спасибо за внимание