Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3**

Вариант 15

Преподаватель \_\_\_\_\_\_\_\_\_\_\_ В.В. Тынченко

подпись, дата инициалы, фамилия

Студент КИ15–16Б, 031510065 \_\_\_\_\_\_\_\_\_\_\_ Т.В. Радионов

номер группы, зачетной книжки подпись, дата инициалы, фамилия

Красноярск 2018

# Задача лабораторной работы

Разработать программу, реализующую метод золотого сечения.

Найти безусловный экстремум функции, выбранной в соответствии с заданием, с использованием разработанной программы.

𝑓(𝑥) = 5𝑥2 − 2𝑥 + 1 → 𝑚𝑖𝑛

Интервал неопределённости [-6,6].

# Листинг программы

using System;

using System.Diagnostics;

namespace ConsoleApp1

{

class Program

{

/// <summary>

/// Function F(x)

/// </summary>

/// <param name="x"></param>

/// <returns></returns>

static double Func(double x) => 5 \* Math.Pow(x, 2) - 2 \* x + 1;

/// <summary>

/// Algorithm

/// </summary>

/// <param name="a"></param>

/// <param name="y"></param>

/// <param name="delta"></param>

/// <returns></returns>

static void Algorithm(double a, double b, double eps)

{

double x = 0;

double y = 0;

double Fy = 0;

double z = 0;

double Fz = 0;

double length = 0;

int k = 0;

// Step 1

length = b - a;

// Step 2

k = 0;

// Step 3

y = a + ((3 - Math.Sqrt(5)) / 2) \* (b - a);

z = a + b - y;

while (true)

{

// Step 4

Fy = Func(y);

Fz = Func(z);

// Step 5

if (Fy <= Fz)

{

// a = a;

b = z;

z = y;

y = a + b - y;

}

else

{

a = y;

// b = b;

y = z;

z = a + b - z;

}

// Step 6

double delta = Math.Abs(a - b);

if (delta <= eps)

{

x = (a + b) / 2;

break;

}

else

{

k++;

}

}

Console.WriteLine("Total steps: {0}", k);

Console.WriteLine("x = {0}", x);

Console.WriteLine("F(x) = {0}", Func(x));

Console.WriteLine("Accuracy: {0:F10}", Math.Abs(Func(0.2) - Func(x)));

}

/// <summary>

/// Main function

/// </summary>

/// <param name="args"></param>

static void Main(string[] args)

{

Stopwatch sw = new Stopwatch(); // Timer for algorithm's speed

sw.Start();

Algorithm(-6, 6, 0.01);

sw.Stop();

Console.WriteLine("Algorithm's speed: {0} ms\n", sw.ElapsedMilliseconds);

sw.Start();

Algorithm(-6, 6, 0.1);

sw.Stop();

Console.WriteLine("Algorithm's speed: {0} ms\n", sw.ElapsedMilliseconds);

sw.Start();

Algorithm(-6, 6, 0.001);

sw.Stop();

Console.WriteLine("Algorithm's speed: {0} ms\n", sw.ElapsedMilliseconds);

sw.Start();

Algorithm(-12, -1, 0.01);

sw.Stop();

Console.WriteLine("Algorithm's speed: {0} ms\n", sw.ElapsedMilliseconds);

sw.Start();

Algorithm(12, 45, 0.01);

sw.Stop();

Console.WriteLine("Algorithm's speed: {0} ms\n", sw.ElapsedMilliseconds);

Console.ReadKey();

}

}

}

# Скриншоты результата выполнения программы

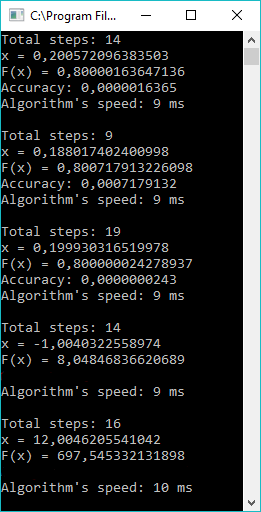


Рисунок 1 – Пример работы программы

# Выводы

В результате проделанной работы была разработана программа, которая выполнила поиск безусловного экстремума заданной функции методом золотого сечения. Причем чем выше задается точность для поиска, тем больше шагов совершает алгоритм. Однако было замечено, что время, затраченное алгоритмом, во всех случаях примерно одинаково и составляет 9-10 мс, что не отличается от результатов метода деления интервала пополам, увеличение и изменение интервала несущественно изменяют количество шагов, осуществляемое алгоритмом, однако количество шагов на 3-6 больше, чем при аналогичных входных данных для метода деления интервала пополам. Наиболее существенное влияние на скорость алгоритма оказывает параметр точности. Метод деления интервала пополам делает меньше шагов для достижения результата по сравнению с методом золотого сечения. Однако, по сравнению с методом деления интервала пополам точность результатов у метода золотого сечения в большинстве (но не во всех) случаях выше.