Министерство образования Республики Беларусь

Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Информационных технологий и управления

Кафедра Интеллектуальных информационных технологий

**ОТЧЁТ**

по лабораторной работе №4

по дисциплине “Численные методы”

Выполнил:

Р. В. Липский, гр. 121701

Проверил:

П. А. Самосонов

**Вариант**



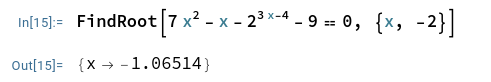
Перенесем все слагаемые в левую часть уравнения:

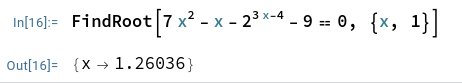
7x^2-x-2^(3x-4)+9 = 0

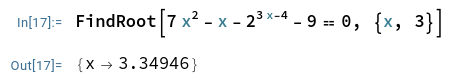
решим методом Ньютона:

R = { -1.065, 1.26, 3.349 }, где R - множество корней уравнения.

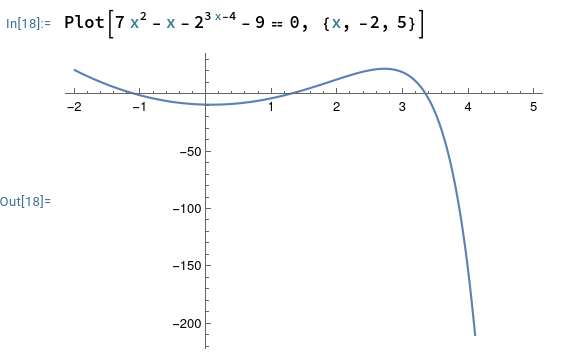
корни, найденные функцией FindRoot пакета математика:







Подтвердим, что корня действительно три:



Исходный код решения:

#include <iostream>

#include <cmath>

#include <set>

#define \_A -2.

#define \_B 4.

#define \_eps 0.001

#define \_step 0.1

double function(const double & param) {

return 7\*param\*param-param-pow(2, 3\*param - 4) - 9;

}

double derivative(const double & param) {

return 14\*param-1-3\*log(2)\*pow(2, 3\*param - 4);

}

double findRoot(const double & near) {

if (fabs(function(near)) < \_eps)

return near;

return findRoot(

near - ( function(near) / derivative(near) )

);

}

double roundToEps(double d) {

return round(d/\_eps)\*\_eps;

}

int main() {

double x = \_A - 1;

std::set<double> roots;

for (double i = \_A; i <= \_B; i += \_step)

roots.emplace(roundToEps(findRoot(i)));

std::cout << "R = { ";

for (auto d : roots) {

std::cout << d << " ";

}

std::cout << "}, где R - множество корней уравнения." << std::endl;

return 0;

}