**Министерство науки и высшего образования РФ**

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»**

**(ВлГУ)**

УДК 004.8

Срок хранения 2 года

ВЛГУ.10.05.04.04.05.00 ПЗ

**ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ**

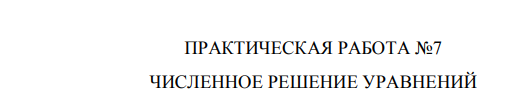
«Численные методы»

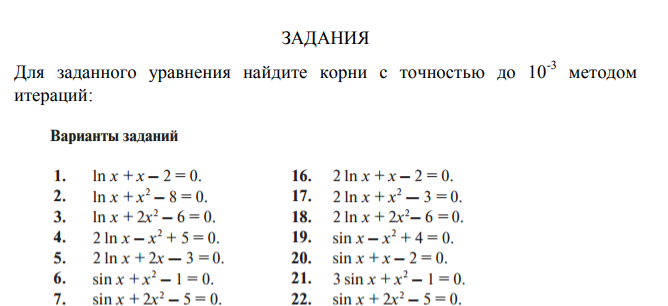
Специальность 10.05.04 – «Информационно-аналитические системы безопасности»

|  |  |  |
| --- | --- | --- |
| Руководитель |  | доцент кафедры ИЗИ Т. В. Спирина |
|  |
| Исполнитель |  | студент группы ИСБ-119 Д. А. Журавлев |
|  |  |  |  |

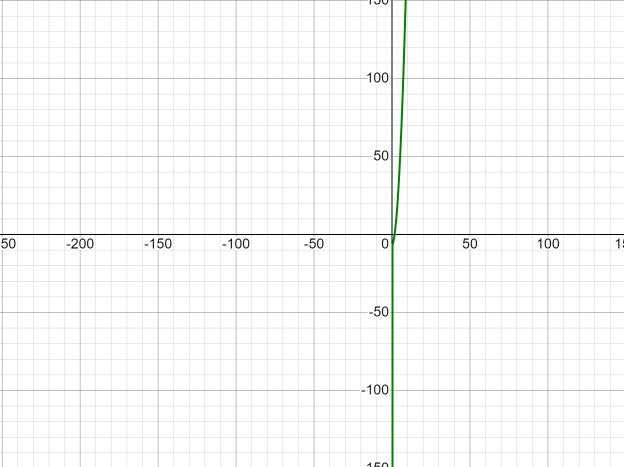
Владимир 2023 г.

Вариант 3

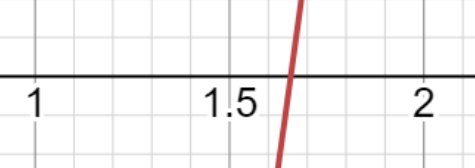


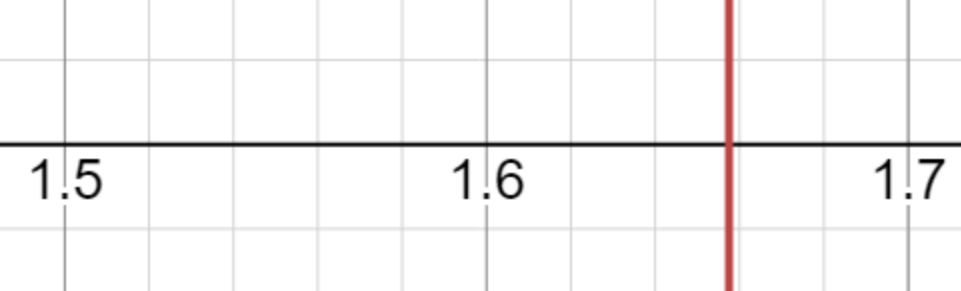


С помощью графического метода найдем количество корней и отрезки их изоляции



Из графика видно, что уравнение имеет 1 корень и 1 отрезок изоляции [1;2]

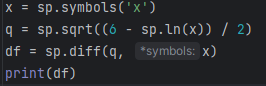




Удобный вид для итераций



Вычислим производную:



Производная q’ =



Значение в точках:



Максимальное в точке x=1, коэффиц сжатия 0.14

**В результате всех итераций получаем:**



Значение функции при этом корне:



**Код программы:**

import math

from sympy.solvers import solve

import sympy

import math

import sympy as sp

def q(x):

return sp.sqrt((6 - sp.log(x, sp.exp)) / 2)

x = sp.symbols('x')

q = sp.sqrt((6 - sp.ln(x)) / 2)

q\_sp = sp.diff(q, x)

print(q\_sp)

def q\_r(x):

return -1/(4\*x\*math.sqrt(3 - math.log(x)/2))

x\_1\_2 = [1, 2]

print(abs(q\_r(x\_1\_2[0])), abs(q\_r(x\_1\_2[1])))

def iterative\_method(p0, epsilon):

p = p0

while True:

p\_new = math.exp(-2 \* p\*\*2 + 6)

if abs(p\_new - p) < epsilon:

return p\_new

p = p\_new

# Задаем начальное приближение и требуемую точность

p0 = 1.5

epsilon = 1e-3

# Применяем метод итераций

root = iterative\_method(p0, epsilon)

print("Корень уравнения: ", root)