

Министерства науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования «Национальный исследовательский университет
ИТМО»

Факультет программной инженерии и компьютерной техники

Вычислительная математика
Лабораторная работа № 2
Вариант: 3

Выполнил:
Вильданов Ильнур Наилевич
Группа №Р3212

Проверила:
Машина Екатерина Алексеевна

Санкт-Петербург
2025 г.

ЦЕЛЬ РАБОТЫ	3
ЗАДАНИЕ	3
ЛИСТИНГ ПРОГРАММЫ	8
ПРИМЕРЫ И РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ	9
ВЫВОД.....	11

Цель работы

Изучить численные методы решения нелинейных уравнений и их систем, найти корни заданного нелинейного уравнения/системы нелинейных уравнений, выполнить программную реализацию методов.

Задание

Вычислительная реализация задачи:

1 Часть. Решение нелинейного уравнения

$$x^3 + 2,84x^2 - 5,606x - 14,766$$

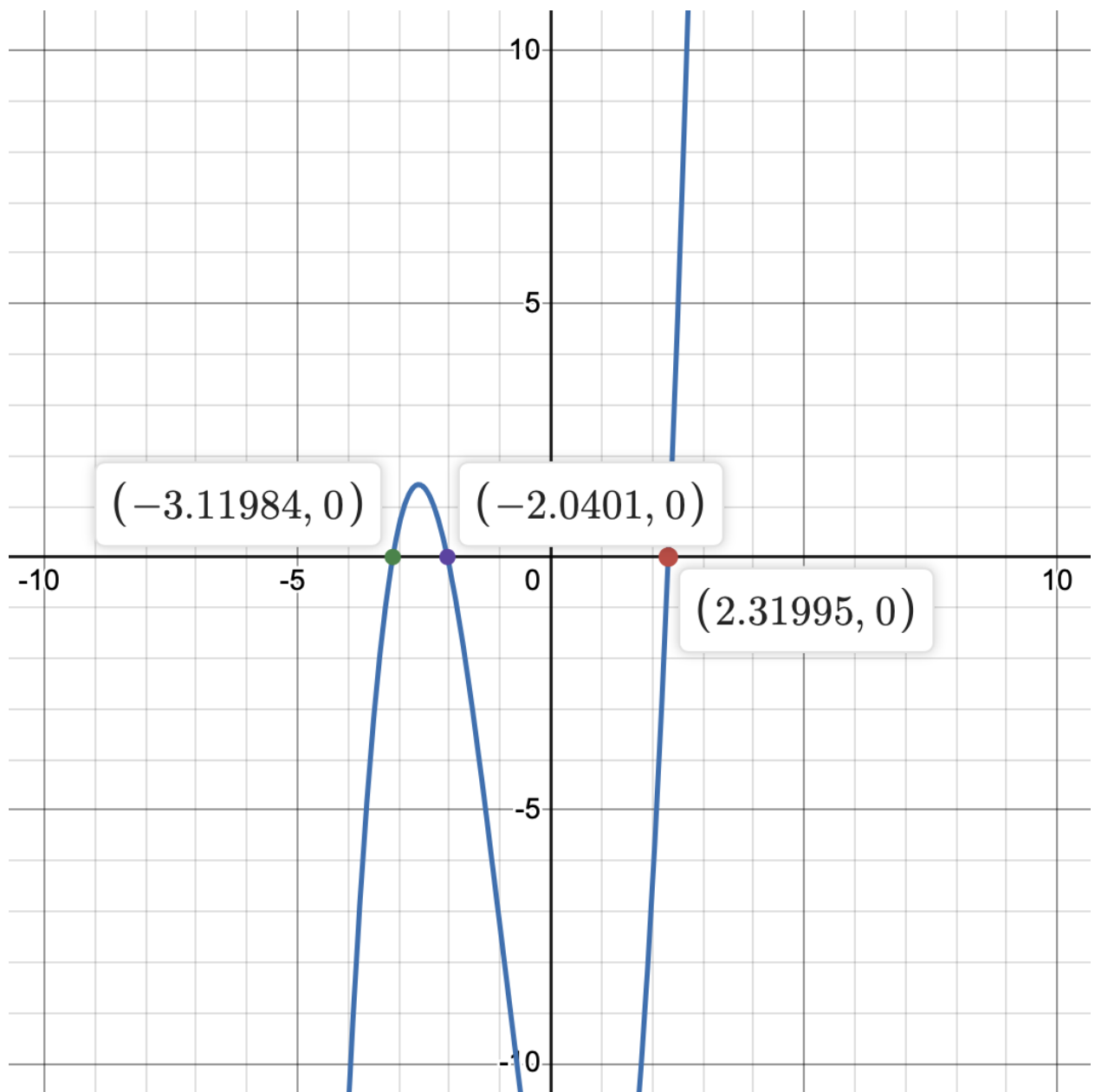


Рисунок 1. Графическое отделение корней нелинейного уравнения

Найдем интервалы изоляции корней:

x	$f(x)$
-4	-10,902
-3	0,61
-2	-0,19
-1	-7,32
0	-14,76
1	-16,53
2	-6,61
3	20,97

Интервалы изоляции корней: [-4; -3] – левый крайний, [-3; -2] – центральный, [2; 3] – правый крайний

Метод половинного деления – крайний правый корень [2; 3]

Рабочая формула:

$$x_i = \frac{a_i + b_i}{2}$$

№ шага	a	b	x	$f(a)$	$f(b)$	$f(x)$	$ a - b $
1	2,000	3,000	2,500	-6,618	20,976	4,594	1,000
2	2,000	2,500	2,250	-6,618	4,594	-1,611	0,500
3	2,250	2,500	2,375	-1,611	4,594	1,336	0,250
4	2,250	2,375	2,313	-1,611	1,336	-0,176	0,125
5	2,313	2,375	2,344	-0,176	1,336	0,570	0,063
6	2,313	2,344	2,328	-0,176	0,570	0,195	0,031
7	2,313	2,328	2,320	-0,176	0,195	0,009	0,016
8	2,313	2,320	2,316	-0,176	0,009	-0,084	0,008

Таблица 1. Уточнение правого крайнего корня уравнения методом половинного деления

Метод Ньютона – центральный корень [-3; -2]

Рабочая формула:

$$x_i = x_{i-1} - \frac{f(x_{i-1})}{f'(x_{i-1})}$$

№ итерации	x_k	$f(x_k)$	$f'(x_k)$	x_{k+1}	$ x_{k+1} - x_k $
1	-2,000	-0,194	-4,966	-2,039	0,039
2	-2,039	-0,016	-4,720	-2,042	0,003

Таблица 2. Уточнение центрального корня уравнения методом Ньютона

Метод простой итерации - крайний левый корень [-4; -3]

Рабочая формула:

$$x_{i+1} = \varphi(x_i)$$

Найдем производную функции $f(x) = x^3 + 2,84x^2 - 5,606x - 14,766$

$$f'(x) = 3x^2 + 5,68x - 5,606$$

На границах интервала производная равна:

$$f'(-4) = 19,674$$

$$f'(-3) = 4,354$$

$$\lambda = \frac{1}{\max(|f'(-4)|, |f'(-3)|)} = -\frac{1}{19,674}$$

$$\varphi(x) = x + \lambda f(x) = x + \frac{x^3 + 2,84x^2 - 5,606x - 14,766}{19,674}$$

$$\varphi'(x) = 1 + \lambda f'(x) = 1 + \frac{3x^2 + 5,68x - 5,606}{19,674}$$

$$\varphi'(-4) = 0$$

$$\varphi'(-3) = 0,778$$

№ итерации	x_k	x_{k+1}	$f(x_{k+1})$	$ x_{k+1} - x_k $
1	-3,8	-3,63	-4,824	0,170
2	-3,63	-3,506	-3,254	0,124
3	-3,506	-3,416	-2,29	0,090
4	-3,416	-3,337	-1,598	0,079
5	-3,337	-3,28	-1,096	0,057
6	-3,28	-3,23	-0,764	0,050
7	-3,23	-3,20	-0,512	0,030
8	-3,20	-3,17	-0,36	0,030
9	-3,17	-3,15	-0,196	0,020
10	-3,15	-3,14	-0,141	0,010
11	-3,14	-3,13	-0,021	0,010

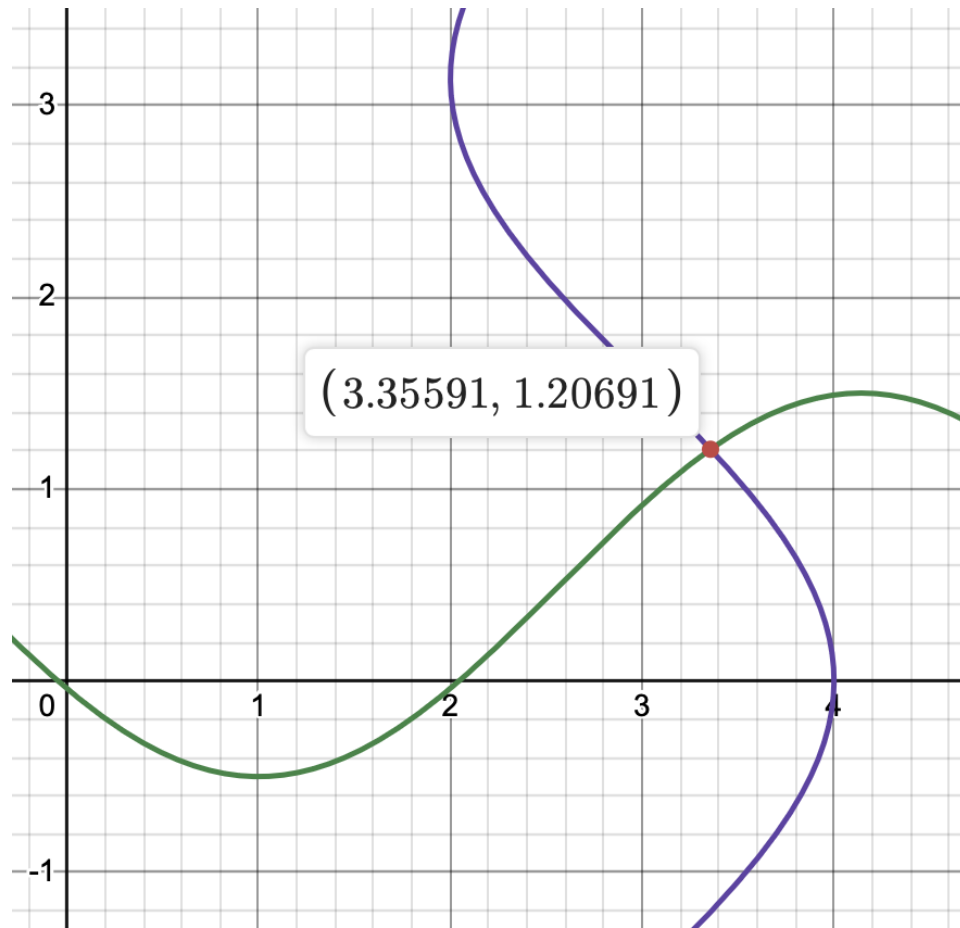
Таблица 3. Уточнение левого крайнего корня уравнения методом простой итерации.

2 часть. Решение системы нелинейных уравнений

$$\begin{cases} \cos(x - 1) + y = 0,5 \\ x - \cos(y) = 3 \end{cases} \quad \varepsilon = 0,01 \quad \text{Метод простой итерации}$$

Приведем систему уравнений к эквивалентному виду:

$$\begin{cases} y = 0,5 - \cos(x - 1) \\ x = 3 + \cos(y) \end{cases}$$



Области изоляции корней:

$$G \begin{cases} 3,2 \leq x \leq 3,5 \\ 1 \leq y \leq 1,3 \end{cases}$$

Проверим условие сходимости. В области G имеем:

$$\begin{aligned} \frac{\partial \varphi_1}{\partial x} &= 0 & \frac{\partial \varphi_1}{\partial y} &= -\sin(y) \\ \frac{\partial \varphi_2}{\partial x} &= \sin(x-1) & \frac{\partial \varphi_2}{\partial y} &= 0 \end{aligned}$$

$$\begin{aligned} \left| \frac{\partial \varphi_1}{\partial x_1} \right| + \left| \frac{\partial \varphi_1}{\partial x_1} \right| &= |0 + (-\sin(y))| \leq |\sin(1,3)| \approx 0,964 \\ \left| \frac{\partial \varphi_2}{\partial x} \right| + \left| \frac{\partial \varphi_2}{\partial y} \right| &= |\sin(x-1) + 0| \leq |\sin(2,5)| \approx 0,598 \end{aligned}$$

$$\max |\varphi'(x)| \leq 0,964 < 1 \quad \rightarrow \quad \text{Процесс сходящийся}$$

Выберем начальное приближение внутри области G: $x^{(0)} = 3, y^{(0)} = 0,5$

$$\begin{cases} x^{(k+1)} = \varphi_1(x, y) \\ y^{(k+1)} = \varphi_2(x, y) \end{cases}$$

№ итерации	x_k	x_{k+1}	$ x_{k+1} - x_k $	y_k	y_{k+1}	$ y_{k+1} - y_k $
1	3,878	3,609	0,269	0,916	1,465	0,549
2	3,609	3,105	0,504	1,465	1,361	0,104
3	3,105	3,208	0,103	1,361	1,009	0,352
4	3,208	3,532	0,325	1,009	1,095	0,085
5	3,532	3,458	0,074	1,095	1,320	0,225
6	3,458	3,248	0,210	1,320	1,275	0,045
7	3,248	3,291	0,043	1,275	1,127	0,149
8	3,291	3,430	0,138	1,127	1,160	0,033
9	3,430	3,400	0,030	1,160	1,257	0,098
10	3,400	3,309	0,091	1,257	1,237	0,020
11	3,309	3,327	0,019	1,237	1,173	0,065
12	3,327	3,388	0,060	1,173	1,186	0,014
13	3,388	3,375	0,013	1,186	1,229	0,043
14	3,375	3,335	0,040	1,229	1,220	0,009
15	3,335	3,343	0,008	1,220	1,192	0,028
16	3,343	3,370	0,026	1,192	1,198	0,006
17	3,370	3,364	0,005	1,198	1,217	0,019
18	3,364	3,347	0,017	1,217	1,213	0,003
19	3,347	3,350	0,004	1,212	1,200	0,012
20	3,350	3,361	0,011	1,200	1,203	0,002
21	3,361	3,360	0,002	1,203	1,211	0,008

Листинг программы

```
def chord_method(f, a: float, b: float, tol: float, max_iters: int = 1000,
fixed_end: str = 'left'):
    fa = f(a)
    fb = f(b)
    if fa * fb > 0:
        raise ValueError("На данном отрезке нет единственного корня.")

    if fixed_end == 'left':
        x_prev = b
        fixed_val = a
    elif fixed_end == 'right':
        x_prev = a
        fixed_val = b
    else:
        raise ValueError("fixed_end должен быть 'left' или 'right'")

    for k in range(1, max_iters + 1):
        ff = f(fixed_val)
        fx = f(x_prev)
        denom = fx - ff
        if denom == 0:
            raise ZeroDivisionError("Деление на ноль при вычислении хорд.")
        x_next = x_prev - (x_prev - fixed_val) * fx / denom

        if math.isnan(x_next) or math.isinf(x_next):
            raise ArithmeticError("Хорд: вычислен NaN/Inf.")

        if abs(x_next - x_prev) < tol or abs(f(x_next)) < tol:
            return x_next, f(x_next), k

        x_prev = x_next

    raise RuntimeError("Хорды: не сошлось за max_iters.")

def secant_method(f, a: float, b: float, tol: float, max_iters: int = 1000):
    x_prev, x_cur = a, b
    f_prev, f_cur = f(x_prev), f(x_cur)

    if f_prev == f_cur:
        raise ZeroDivisionError("Секущие: f(a) == f(b), деление на ноль.")

    for k in range(1, max_iters + 1):
        x_next = x_cur - (x_cur - x_prev) * f_cur / (f_cur - f_prev)
        if math.isnan(x_next) or math.isinf(x_next):
            raise ArithmeticError("Секущие: получили NaN/Inf.")
        if abs(x_next - x_cur) < tol or abs(f(x_next)) < tol:
            return x_next, f(x_next), k
        x_prev, f_prev = x_cur, f_cur
        x_cur, f_cur = x_next, f(x_next)

    raise RuntimeError("Секущие: не сошлось за max_iters.")

def simple_iteration_method(f, phi, phi_prime, a: float, b: float, tol:
float, max_iters: int = 1000):
    xs = np.linspace(a, b, 200)
    q = max(abs(phi_prime(x)) for x in xs)
    if q >= 1:
        raise ValueError(f"Условие сходимости не выполнено: max|φ'| = {q:.4f}
≥ 1")

    x = (a + b) / 2
    for k in range(1, max_iters + 1):
```



```

    x_next = phi(x)
    if abs(x_next - x) < tol:
        return x_next, f(x_next), k
    x = x_next
raise RuntimeError("Простая итерация: не сошлось за max_iters.")

```

```

def newton_system_method(system: dict, x0: float, y0: float, tol: float,
max_iters: int = 100):
    flf2 = system['functions']
    jac = system['jacobian']

    x, y = x0, y0
    errors = []

    for k in range(1, max_iters + 1):
        f1_val, f2_val = flf2(x, y)
        J11, J12 = jac(x, y)[0]
        J21, J22 = jac(x, y)[1]
        det = J11 * J22 - J12 * J21
        if abs(det) < 1e-14:
            raise ValueError("Якобиан вырождён (det≈0).")

        # dx dy через формулу Крамера
        dx = (-f1_val * J22 + f2_val * J12) / det
        dy = (J11 * (-f2_val) + f1_val * J21) / det

        x_new = x + dx
        y_new = y + dy
        err = math.hypot(dx, dy)
        errors.append(err)

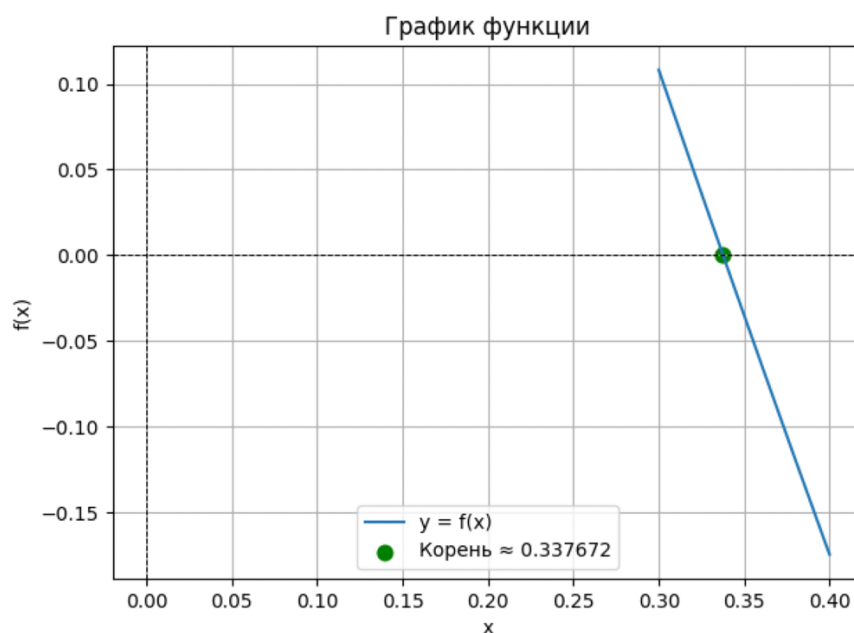
        if err < tol:
            return x_new, y_new, errors, k

        x, y = x_new, y_new

    raise RuntimeError("Ньютон (система): не сошлось за max_iters.")

```

Примеры и результаты работы программы



```

=== Решение нелинейных уравнений и систем нелинейных уравнений ===

1) Нелинейное уравнение
2) Система нелинейных уравнений
Выберите (1/2): 1
Доступные одиночные уравнения:
1)  $f_1(x) = x^4 - 3 \cdot x + 1$ 
2)  $f_2(x) = e^{-x} - x^2 + 2$ 
3)  $f_3(x) = x \cdot \sin(x) - 1$ 
4)  $f_4(x) = \ln(x + 2) + x^2 - 3$ 
Выберите номер уравнения: 1

Способ ввода:
1) С клавиатуры
2) Из файла
Ваш выбор (1/2): 1
Введите a (левая граница, float): 0.3
Введите b (правая граница, float): 0.4
Введите ε (точность, float > 0): 0.001

Методы уточнения корня:
1) Метод хорд
2) Метод секущих
3) Метод простой итерации
Выберите метод (1-3): 1

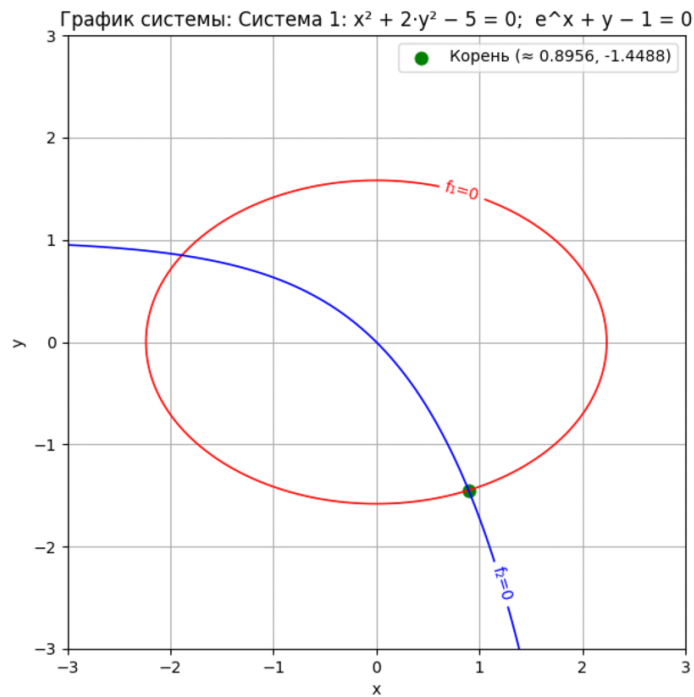
---- Метод хорд ----
Выберите форму:
a) фиксируем левый конец ( $x_0 = b$ )
b) фиксируем правый конец ( $x_0 = a$ )
Ваш выбор (a/b): a

Вывод результата:
1) В консоль
2) В файл
Ваш выбор (1/2): 1
=== Результат одиночного уравнения ( $f_1(x) = x^4 - 3 \cdot x + 1$ ) ===
Корень  $x^* = 0.337672$ 
 $f(x^*) = -1.421294e-05$ 
Число итераций: 2
Точность  $\varepsilon = 0.001$ 
=====

(Строим график функции...)

Process finished with exit code 0

```



```

=== Решение нелинейных уравнений и систем нелинейных уравнений ===

1) Нелинейное уравнение
2) Система нелинейных уравнений
Выберите (1/2): 2
Доступные системы:
  1) Система 1:  $x^2 + 2 \cdot y^2 - 5 = 0$ ;  $e^x + y - 1 = 0$ 
  2) Система 2:  $\sin(x) + x \cdot y - 0.5 = 0$ ;  $x + \cos(y) - 2 = 0$ 
Выберите номер системы (1 или 2): 1

Введите начальные приближения для  $x_0$  и  $y_0$ :
Введите начальное приближение  $x_0$ : 0.9
Введите начальное приближение  $y_0$ : -1.44
Введите  $\varepsilon$  (точность для системы, float > 0): 0.001

Вывод результата:
  1) В консоль
  2) В файл
Ваш выбор (1/2): 1
=== Результат для системы (Система 1:  $x^2 + 2 \cdot y^2 - 5 = 0$ ;  $e^x + y - 1 = 0$ ) ===
Найденное решение:  $x^* = 0.895589$ ,  $y^* = -1.448779$ 
Число итераций: 2
Вектор погрешностей: [0.009837197711075461, 3.095167884212234e-05]
Проверка:  $f_1(x^*, y^*) = 1.54e-09$ ,  $f_2(x^*, y^*) = 4.65e-10$ 
=====

(Строим график системы...)

```

Вывод

В результате выполнения данной лабораторной работы были изучены численные методы решения нелинейных уравнений и систем нелинейных уравнений. Для нелинейных уравнений реализованы методы хорд, секущих и простых итераций, а для систем нелинейных уравнений метод Ньютона на языке программирования Python.