Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение  
высшего образования   
«Алтайский государственный технический университет им. И. И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Отчет защищен с оценкой \_\_\_\_\_

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Отчет

По лабораторной работе №8

**«Метод Ньютона для решения систем нелинейных уравнений»**

по дисциплине «Вычислительные алгоритмы»

Студент группы ПИ-02

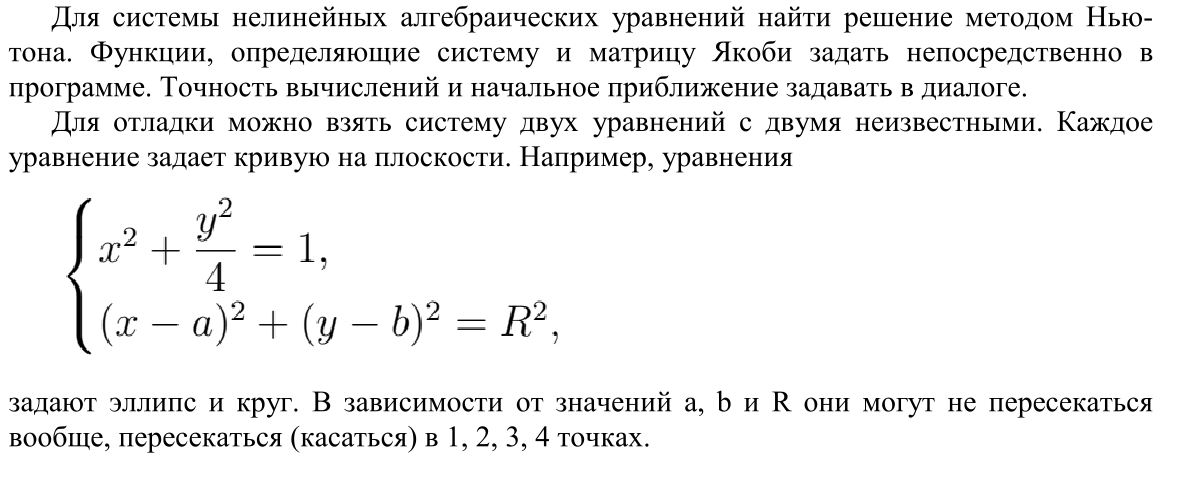
Чередов Р. А.

Преподаватель, к.ф-м.н., доцент,

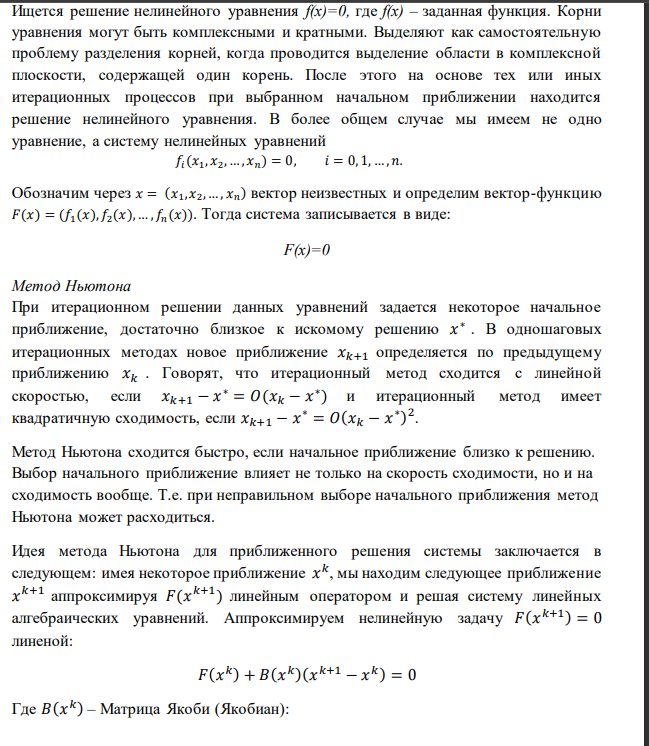
Проскурин А. В.

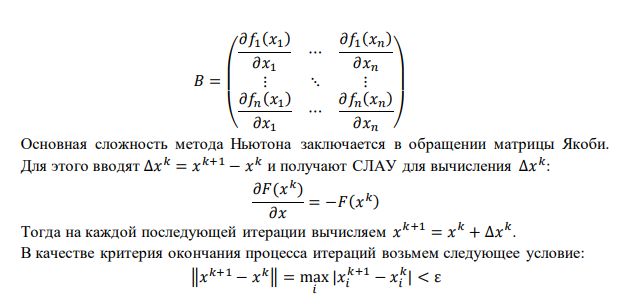
Барнаул 2023

**Задание к лабораторной работе:**



**Описание метода:**



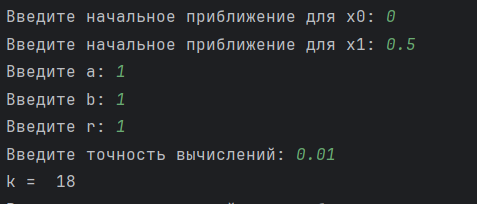


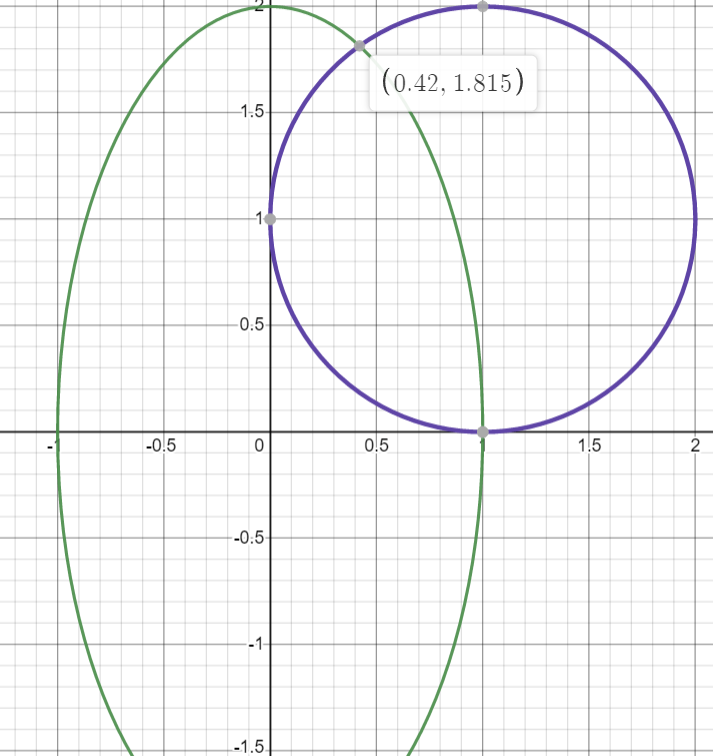
**Программа:**

import numpy as np  
  
# Функции, определяющие систему  
def f(x):  
 return np.array([x[0] \*\* 2 + x[1] \*\* 2 / 4 - 1, (x[0] - a) \*\* 2 + (x[1] - b) \*\* 2 - R \*\* 2])  
  
  
# Матрица Якоби  
def jacobian(x):  
 return np.array([[2 \* x[0], x[1] / 2], [2 \* (x[0] - a), 2 \* (x[1] - b)]])  
  
  
# Метод Ньютона  
def newton\_method(f, jacobian, x0, eps):  
 k = 0  
 xn = x0  
 fn = f(xn)  
 jn = jacobian(xn)  
 delta = np.linalg.solve(jn, -fn)  
  
 while np.linalg.norm(delta) > eps:  
 xn = xn + delta  
 fn = f(xn)  
 jn = jacobian(xn)  
 delta = np.linalg.solve(jn, -fn)  
 k += 1  
 print("k = ", str(k))  
 return xn  
  
  
# Точность вычислений и начальное приближение, а также необходимые переменные  
x0 = np.array([float(input("Введите начальное приближение для x0: ")),  
 float(input("Введите начальное приближение для x1: "))])  
a = float(input("Введите a: "))  
b = float(input("Введите b: "))  
R = float(input("Введите r: "))  
eps = float(input("Введите точность вычислений: "))  
  
# Вычисление решения системы  
x = newton\_method(f, jacobian, x0, eps)  
print("Решение системы нелинейных алгебраических уравнений методом Ньютона: ", np.around(x, 5))

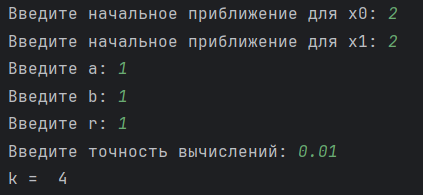
**Тесты:**

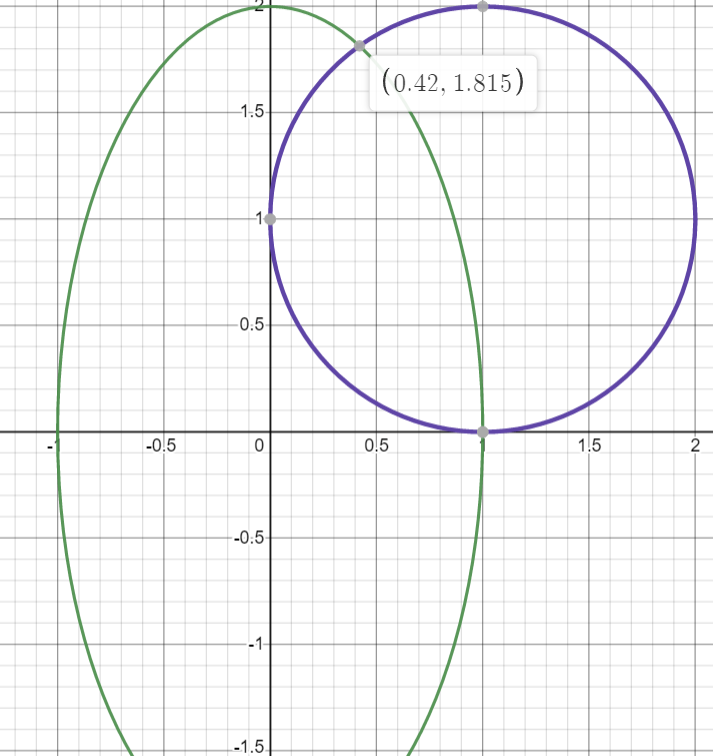
1. Решение системы уравнения, предложенной для тестирования с начальным приближением x = 0 и y = 0.5, a = 1, b = 1, r = 1 и точностью eps = 0.01.



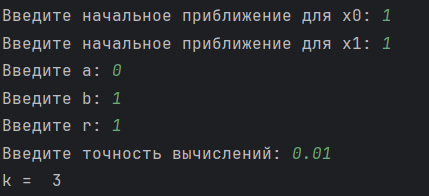
****

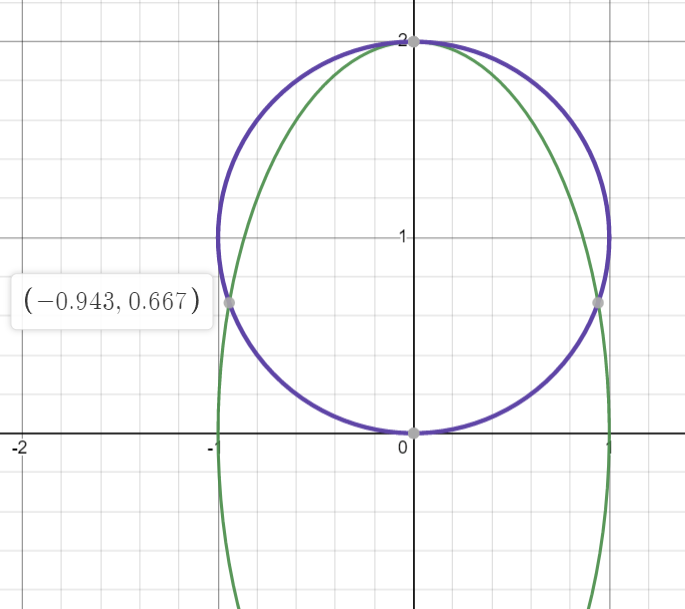
1. Решение системы уравнения, предложенной для тестирования с начальным приближением x = 2 и y = 2, a = 1, b = 1, r = 1 и точностью eps = 0.01.



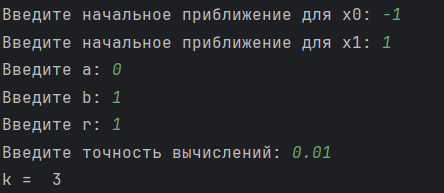
****

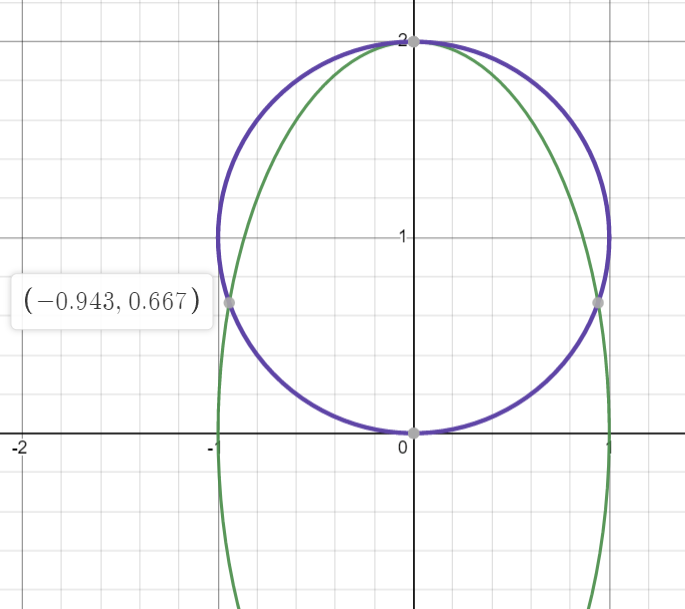
1. Решение системы уравнения, предложенной для тестирования с начальным приближением x = 1 и y = 1, a = 0, b = 1, r = 1 и точностью eps = 0.01.



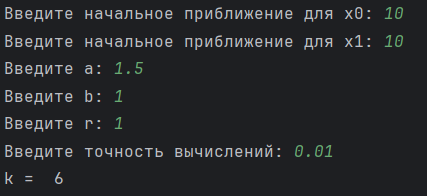
****

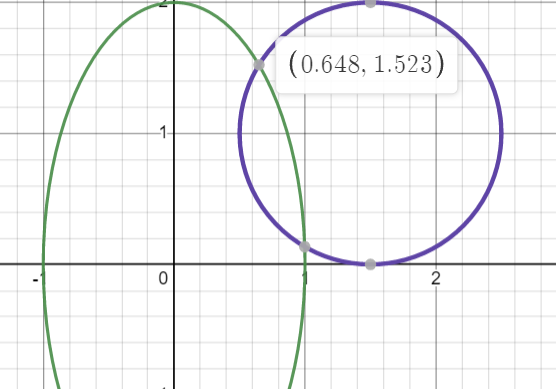
1. Решение системы уравнения, предложенной для тестирования с начальным приближением x = -1 и y = 1, a = 0, b = 1, r = 1 и точностью eps = 0.01.



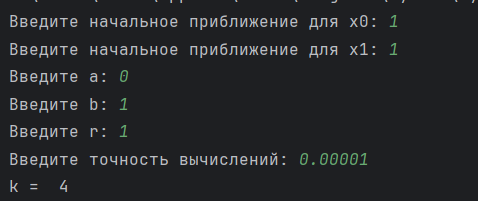
****

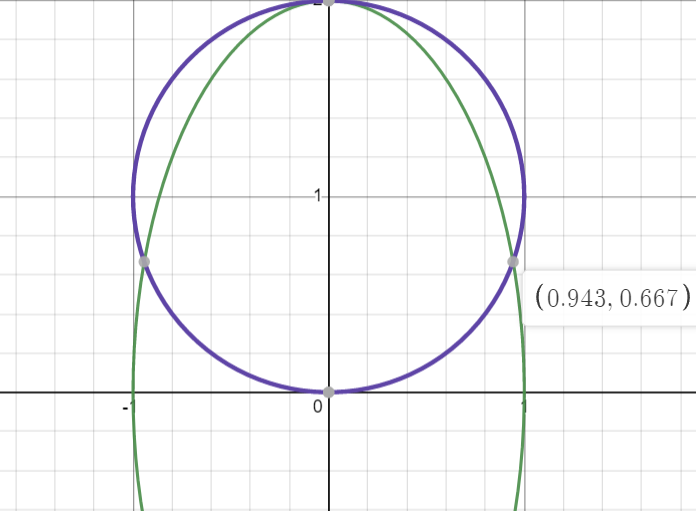
1. Решение системы уравнения, предложенной для тестирования с начальным приближением x = 10 и y = 10, a = 1.5, b = 1, r = 1 и точностью eps = 0.01.



****

1. Решение системы уравнения, предложенной для тестирования с начальным приближением x = 1 и y = 1, a = 0, b = 1, r = 1 и точностью eps = 0.001.



****

**Вывод:** Исходя из результатов вычислительных экспериментов, метод Ньютона быстро сходится в случае, когда начальное приближение близко к решению. Кроме того, при выборе “неудачного” начального приближения метод может и вовсе не сойтись. Количество шагов, что естественно, увеличивается при уменьшении eps, то есть увеличении точности.