Санкт-Петербургский Национальный Исследовательский Университет

ИТМО

МФКТиУ, факультет ПИиКТ

**Лабораторная работа №2 по**

**«Вычислительной Математике»**

«Численное решение нелинейных уравнений и систем»

**Преподаватель**: Малышева Татьяна Алексеевна

**Выполнил:** Стефан Лабович

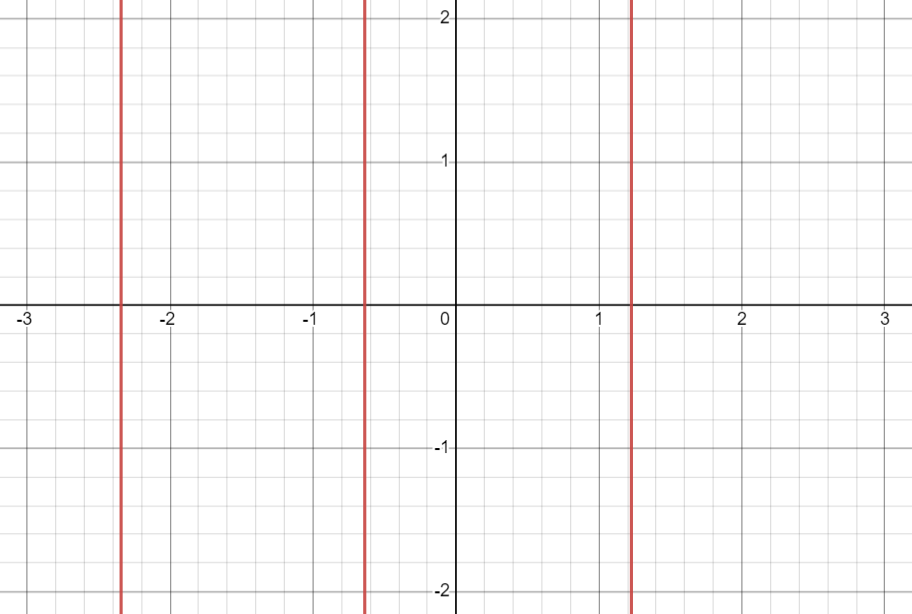
**Группа:** Р3210

**Вариант:** 11

**Цель работы:**

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

**Вычислительная реализация задачи:**

Функция:

Метод половинного деления (*крайний правый корень*):

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № шага |  |  |  |  |  |  |  |
| 1. | 1 | 2 | 1.5 | -5.53 | 39.43 | 9.99125 | 1 |
| 2. | 1 | 1.5 | 1.25 | -5.53 | 9.99125 | 0.69953125 | 0.5 |
| 3. | 1 | 1.25 | 1.125 | -5.53 | 0.69953125 | -2.771933594 | 0.25 |
| 4. | 1.125 | 1.25 | 1.1875 | -2.771933594 | 0.69953125 | -1.128635254 | 0.125 |
| 5. | 1.1875 | 1.25 | 1.21875 | -1.128635254 | 0.69953125 | -0.2380679321 | 0.0625 |
| 6. | 1.21875 | 1.25 | 1.234375 | -0.2380679321 | 0.69953125 | 0.2248017502 | 0.03125 |
| 7. | 1.21875 | 1.234375 | 1.2265625 | -0.2380679321 | 0.2248017502 | -0.008109202385 | 0.015625 |
| 8. | 1.2265625 | 1.234375 | 1.23046875 | -0.008109202385 | 0.2248017502 | 0.1079764503 | 0.0078125 |

Метод хорд (*крайний левый корень*):

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № шага |  |  |  |  |  |  |  |
| 1. | -3 | -2 | -2.187012263 | -29.17 | 6.71 | 3.675109992 | 1 |
| 2. | -3 | -2.187012263 | -2.277979209 | -29.17 | 3.675109994 | 1.668877721 | 0.812987737 |
| 3. | -3 | -2.277979209 | -2.317052110 | -29.17 | 1.668877728 | 0.693480469 | 0.722020791 |
| 4. | -3 | -2.317052110 | -2.332911314 | -29.17 | 0.693480470 | 0.277524892 | 0.682947890 |
| 5. | -3 | -2.332911314 | -2.339198216 | -29.17 | 0.277524888 | 0.109388940 | 0.667088686 |
| 6. | -3 | -2.339198216 | -2.341666997 | -29.17 | 0.109388949 | 0.042858398 | 0.660801784 |
| 7. | -3 | -2.341666997 | -2.342632842 | -29.17 | 0.042858411 | 0.016752320 | 0.658333003 |
| 8. | -3 | -2.342632842 | -2.343010151 | -29.17 | 0.016752332 | 0.006542050 | 0.657367158 |

Метод простой итерации (*центральный корень*):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № шага |  |  |  |  |  |
| 1. | -0.6 | -0.5476 | -0.656923077 | -0.656923077 | 0.056923077 |
| 2. | -0.656923077 | 0.258441945 | -0.63005801 | -0.63005801 | 0.026865067 |
| 3. | -0.63005801 | -0.121498617 | -0.642687804 | -0.642687804 | 0.012629794 |
| 4. | -0.642687804 | 0.057258579 | -0.636735769 | -0.636735769 | 0.005952035 |
| 5. | -0.636735769 | -0.026956902 | -0.639537941 | -0.639537941 | 0.002802173 |
| 6. | -0.639537941 | 0.012697541 | -0.638218031 | -0.638218031 | 0.001319911 |
| 7. | -0.638218031 | -0.00597955 | -0.638839605 | -0.638839605 | 0.000621575 |

**Программная реализация задачи:**

Методы используемые в задачедля **нелинейных уравнений:**

**Метод секащих:**

Является двухшаговым, т.е. новое приближение 𝑥𝑖+1 определяется двумя предыдущими итерациями 𝑥𝑖 и 𝑥𝑖−1. Выбор 𝑥0 определяется как и в методе Ньютона, 𝑥1 выбирается рядом с начальным самостоятельно.

Критерий окончания итерационного процесса:

Рабочая формула:

**Метод простой итерации:**

Уравнение 𝑓(𝑥) = 0 приведем к эквивалентному виду: 𝑥 = 𝜑(𝑥), выразив 𝑥 из исходного уравнения. Зная начальное приближение: 𝑥0 ∈ [𝑎, 𝑏], найдем очередные приближения: 𝑥1 = 𝜑(𝑥0) → 𝑥2 = 𝜑(𝑥1) …

Критерий окончания итерационного процесса:

Достаточное условие сходимости метода:

Рабочая формула:

Метод используемый в задаче для **системы нелинейных уравнений:**

**Метод простой итерации:**

Критерий окончания итерационного процесса:

Достаточное условие сходимости метода:

Рабочая формула:

**Код методов:**

Методы используемые в задачедля **нелинейных уравнений:**

**Метод секащих:**

def secant(f, x1, x2, accuracy):  
 diff = fabs(x2 - x1)   
 iters = 0  
 while diff > accuracy:  
 x\_new = x2 - (x2 - x1) / (f(x2) - f(x1)) \* f(x2)  
 x1 = x2  
 x2 = x\_new  
 iters += 1  
 diff = fabs(x2 - x1)

**Метод простой итерации:**

def fixed\_point\_iteration(f, f\_der, x0, e):   
 x = x0  
 l = -1 / f\_der(x0)   
 iters = 0  
 while fabs(f(x)) > e:  
 x0 = x  
 x = x0 + l \* f(x0)  
 iters += 1

Метод используемый в задаче для **системы нелинейных уравнений:**

**Метод простой итерации:**

sum\_der\_x = f1\_der\_x(x) + f1\_der\_y(y)  
sum\_der\_y = f2\_der\_x(x, y) + f2\_der\_y(y)  
  
if sum\_der\_x < 1 and sum\_der\_y < 1:  
 condition = True  
 while condition:  
 temp\_x = f1(x, y)  
 temp\_y = f2(x, y)  
  
 condition = fabs(temp\_x - x) > e and fabs(temp\_y - y) > e  
  
 x = temp\_x  
 y = temp\_y  
else:  
 print("Convergence condition has not been met.")

Весь код доступен по ссылке:

<https://github.com/SteLaba/CompMath2>

**Пример работы метода:**

Решение методом секащих:

x1 x2 f(x1) f(x2) |f(x1)-f(x2)|

1.0000 3.0000 2.2000 111.6000 2.0000

3.0000 0.9598 111.6000 1.5902 2.0402

0.9598 0.9303 1.5902 1.1701 0.0295

0.9303 0.8481 1.1701 0.1166 0.0822

0.8481 0.8390 0.1166 0.0102 0.0091

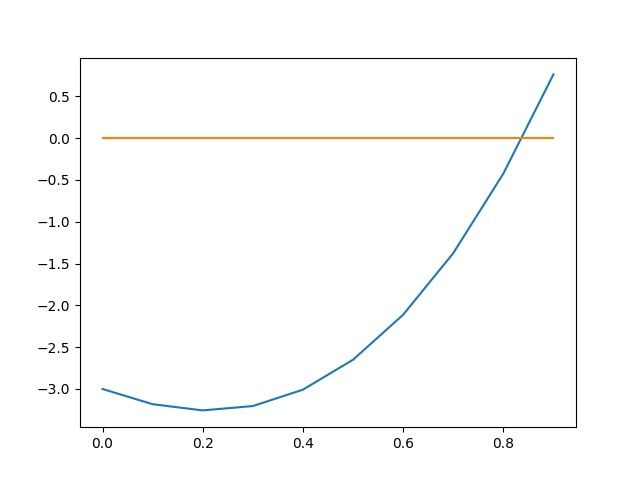
0.8390 0.8382 0.0102 0.0001 0.0009

x = 0.8381700119477727

f(x)= 0.00010547553995188252

Accuracy: 0.0008761370734038376

Iterations: 5



**Выводы:**

В процессе выполнения данной лабораторной работы я ознакомился с разными численными методами решения нелинейных уравнений, я выучил как они работают и сделал программную реализацию метода секущих и метода простой итерации.