

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА №2

по дисциплине

«Вычислительная математика»
Вариант № 13

Выполнил работу:

Студент группы Р3218
Рамеев Тимур Ильгизович

Преподаватель:

Бострикова Дарья
Константиновна

Содержание

Цель	2
Задание	2
Для вычислительной реализации задачи	2
Для программной реализации задачи	3
Рабочие формулы используемых методов	4
Метод половинного деления	4
Метод хорд	4
Метод Ньютона	4
Метод секущих	4
Метод простой итерации	4
Решение нелинейного уравнения	5
График функции	5
Определение промежутков изоляции корней	6
Метод хорд	6
Метод Ньютона	6
Метод простой итерации	7
Решение системы нелинейных уравнений	8
График функций	8
Определение областей изоляции корней	8
Метод простой итерации	9
Программная реализация задачи	10
Ссылка на исходный код к лабораторной работе	10
Основные методы программы	10
Вывод	12

Цель

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

Задание

Для вычислительной реализации задачи

1 Часть. Решение нелинейного уравнения

1. Отделить корни заданного нелинейного уравнения графически (вид уравнения представлен в табл. 6)
2. Определить интервалы изоляции корней.
3. Уточнить корни нелинейного уравнения (см. табл. 6) с точностью $\epsilon = 10^{-2}$
4. Используемые методы для уточнения каждого из 3-х корней многочлена представлены в таблице 7.
5. Вычисления оформить в виде таблиц (1-5), в зависимости от заданного метода. Для всех значений в таблице удерживать 3 знака после запятой.
 - 5.1 Для метода половинного деления заполнить таблицу 1.
 - 5.2 Для метода хорд заполнить таблицу 2.
 - 5.3 Для метода Ньютона заполнить таблицу 3.
 - 5.4 Для метода секущих заполнить таблицу 4.
 - 5.5 Для метода простой итерации заполнить таблицу 5. Проверить условие сходимости метода на выбранном интервале.
6. Заполненные таблицы отобразить в отчете.

2 Часть. Решение системы нелинейных уравнений

1. Отделить корни заданной системы нелинейных уравнений графически (вид системы представлен в табл. 8).
2. Используя указанный метод, решить систему нелинейных уравнений с точностью до 0,01.
3. Для метода простой итерации проверить условие сходимости метода.
4. Подробные вычисления привести в отчете.

Для программной реализации задачи

1 Часть. Решение нелинейного уравнения

1. Все численные методы (см. табл. 9) должны быть реализованы в виде отдельных подпрограмм/методов/классов.
2. Пользователь выбирает уравнение, корень/корни которого требуется вычислить (3-5 функций, в том числе и трансцендентные), из тех, которые предлагает программа.
3. Предусмотреть ввод исходных данных (границы интервала/начальное приближение к корню и погрешность вычисления) из файла или с клавиатуры по выбору конечного пользователя.
4. Выполнить верификацию исходных данных. Необходимо анализировать наличие корня на введенном интервале. Если на интервале несколько корней или они отсутствуют – выдавать соответствующее сообщение. Программа должна реагировать на некорректные введенные данные.
5. Для методов, требующих начальное приближение к корню (методы Ньютона, секущих, хорд с фиксированным концом, простой итерации), выбор начального приближения (а или b) вычислять в программе.
6. Для метода простой итерации проверять достаточное условие сходимости метода на введенном интервале.
7. Предусмотреть вывод результатов (найденный корень уравнения, значение функции в корне, число итераций) в файл или на экран по выбору конечного пользователя.
8. Организовать вывод графика функции, график должен полностью отображать весь исследуемый интервал (с запасом).

2 Часть. Решение системы нелинейных уравнений

1. Пользователь выбирает предлагаемые программой системы двух нелинейных уравнений (2-3 системы).
2. Организовать вывод графика функций.
3. Начальные приближения ввести с клавиатуры.
4. Для метода простой итерации проверить достаточное условие сходимости.
5. Организовать вывод вектора неизвестных: x_1, x_2 .
6. Организовать вывод количества итераций, за которое было найдено решение.
7. Организовать вывод вектора погрешностей: $|x_i^{(k)} - x_i^{(k-1)}|$
8. Проверить правильность решения системы нелинейных уравнений.

Рабочие формулы используемых методов

Метод половинного деления

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

Метод хорд

$$x_i = \frac{a_i f(b_i) + b_i f(a_i)}{f(b_i) - f(a_i)}$$

Метод Ньютона

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

Метод секущих

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

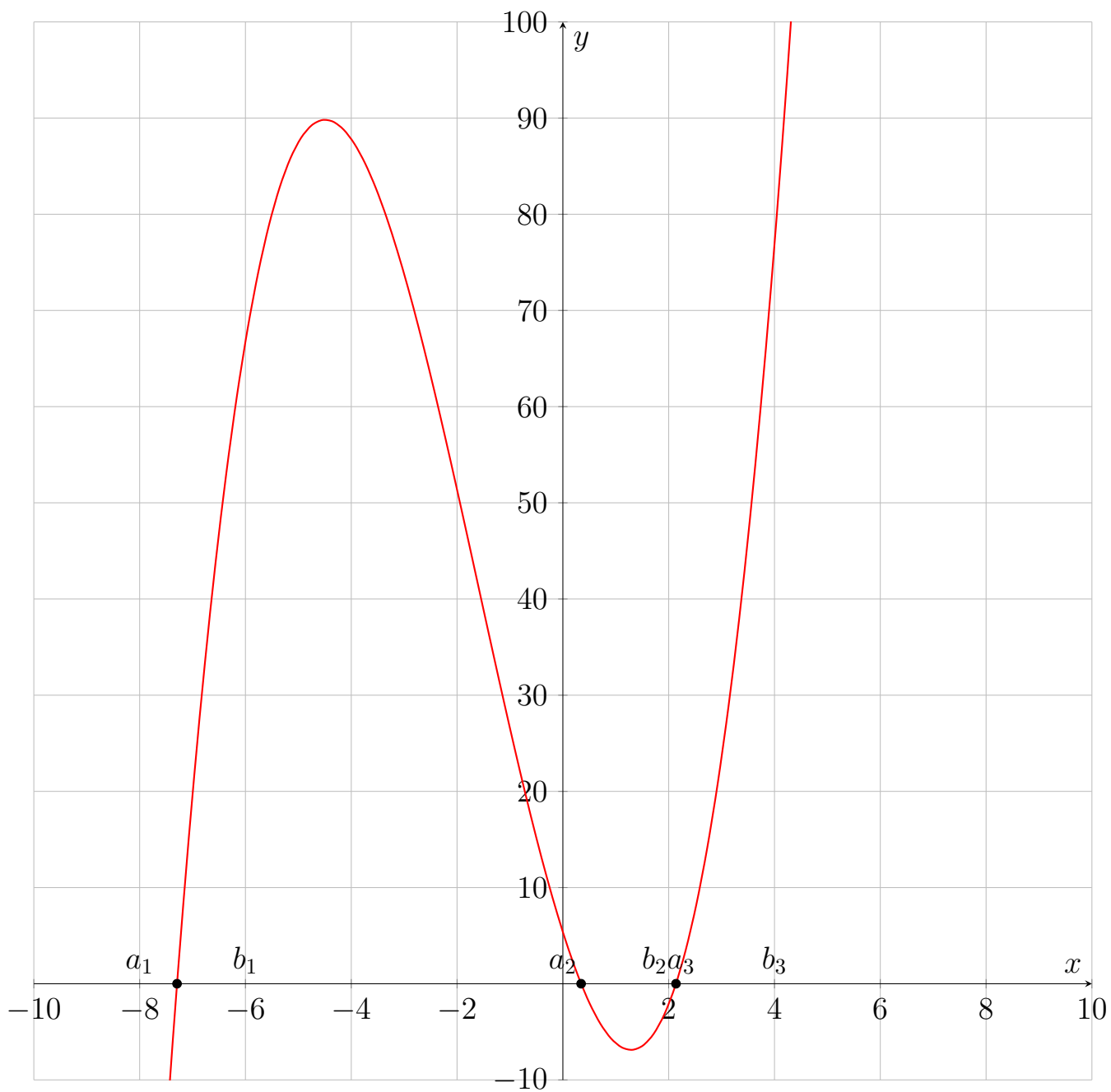
Метод простой итерации

$$x_i = \phi(x_{i-1})$$

Решение нелинейного уравнения

График функции

$$f(x) = x^3 + 4.81x^2 - 17.37x + 5.38$$



Определение промежутков изоляции корней

Определим интервалы изоляции корней по графику "на глаз"

1. Крайний левый корень - $[-8 : -6]$
2. Центральный корень - $[0 : 2]$
3. Крайний правый корень - $[2 : 4]$

Метод хорд

№ Шага	a	b	x	$f(a)$	$f(b)$	$f(x)$	$ x_{k+1} - x_k $
1	-8,000	-6,000	-7,055	-59,820	66,760	16,185	-
2	-8,000	-7,055	-7,256	-59,820	16,185	2,636	0,201
3	-8,00	-7,256	-7,287	-59,820	2,636	0,426	0,031
4	-8,00	-7,287	-7,292	-59,820	0,426	0,066	0,005

Таблица 1: Для крайне левого корня

Метод Ньютона

Найдем производную исходной функции:

$$f'(x) = 3x^2 + 9,62x - 17,37$$

№ Итерации	x_k	$f(x_k)$	$f'(x_k)$	x_{k+1}	$ x_{k+1} - x_k $
1	1,000	-6,180	-4,750	-0,301	1,301
2	-0,301	11,017	-19,994	0,250	0,551
3	0,250	1,354	-14,778	0,342	0,092
4	0,342	0,042	-13,729	0,345	0,003

Таблица 2: Для центрального корня

Метод простой итерации

Проверка сходимости

Преобразуем уравнение к виду $x = \phi(x)$

$$\phi(x) = \frac{x^3 + 4,81x^2 + 5,38}{17,37}$$

$$\phi'(x) = \frac{3x^2 + 9,62x}{17,37}$$

$|\phi'(2)| \approx 1,9 > 1$ - условие сходимости не выполняется

Попробуем выразить x другим способом

$$\phi(x) = \sqrt{\frac{17,37x - x^3 - 5,38}{4,81}}$$

$|\phi'(2)| \approx 2,1 > 1$ - условие сходимости не выполняется

Попробуем выразить x другим способом

$$\phi(x) = \sqrt[3]{17,37x - 4,81x^2 - 5,38}$$

$|\phi'(2)| \approx 2,2 > 1$ - условие сходимости не выполняется

Выразить x так, чтобы коэффициент сжатия был меньше единицы не получается, поэтому применим прием введения параметра λ

$$\begin{aligned} f'(x) &= 3x^2 + 9,62x - 17,37 \\ f'(2) &= 13,87 \quad f'(4) = 69,11 \quad \lambda = -\frac{1}{\max_{[a,b]} |f'(x)|} = -\frac{1}{69,11} \\ \phi(x) &= x + \lambda f(x) = x - \frac{1}{69,11}(x^3 + 4,81x^2 - 17,37x + 5,38) \\ \phi(x) &= -0,014x^3 - 0,070x^2 + 1,251x - 0,078 \quad x_0 = 3 \end{aligned}$$

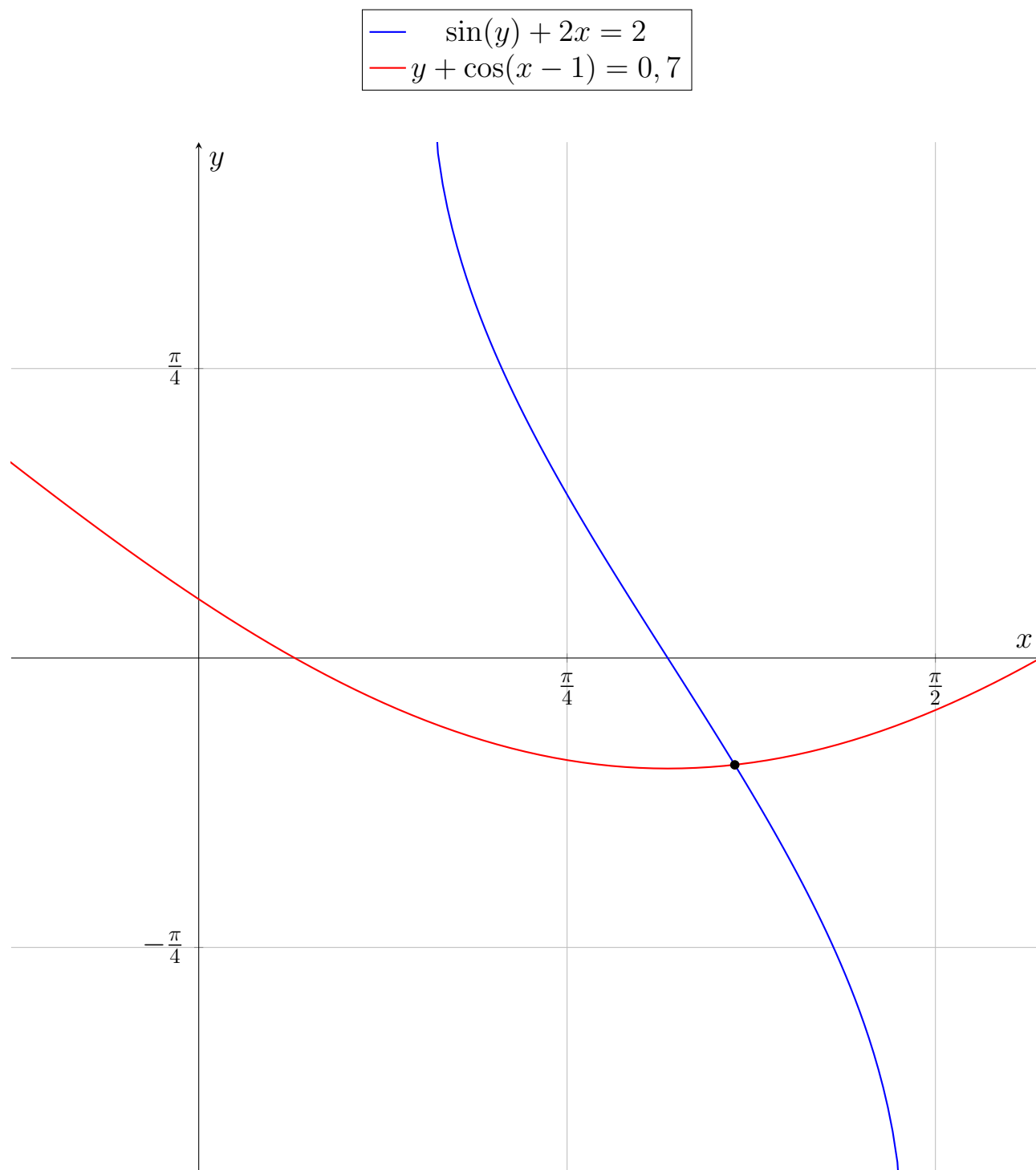
Решение

№ Итерации	x_k	x_{k+1}	$f(x_{k+1})$	$ x_{k+1} - x_k $
1	3	2,667	12,237	0,333
2	2,667	2,495	7,516	0,172
3	2,495	2,390	4,993	0,105
4	2,390	2,320	3,458	0,070
5	2,320	2,273	2,493	0,053
6	2,273	2,239	1,826	0,034
7	2,239	2,215	1,372	0,024
8	2,215	2,197	1,040	0,018
9	2,197	2,184	0,804	0,013
10	2,184	2,174	0,626	0,010

Таблица 3: Для крайне правого корня

Решение системы нелинейных уравнений

График функций



Определение областей изоляции корней

По графику видно, что искомый корень находится в области:

$$\begin{cases} 0 < x < \frac{\pi}{2} \\ -\frac{\pi}{4} < y < 0 \end{cases}$$

Метод простой итерации

Проверка сходимости

$$\begin{cases} x = 1 - \frac{\sin(y)}{2} \\ y = 0,7 - \cos(x - 1) \end{cases}$$

$$\begin{aligned} \frac{\partial \phi_1}{\partial x} &= 0 & \frac{\partial \phi_1}{\partial y} &= -\frac{\cos(y)}{2} \\ \frac{\partial \phi_2}{\partial x} &= \sin(x - 1) & \frac{\partial \phi_2}{\partial y} &= 0 \end{aligned}$$

$$\left| \frac{\partial \phi_1}{\partial x} \right| + \left| \frac{\partial \phi_1}{\partial y} \right| < 1$$

$$\left| \frac{\partial \phi_2}{\partial x} \right| + \left| \frac{\partial \phi_2}{\partial y} \right| < 1 \text{ - следовательно процесс сходящийся}$$

Решение

Выберем начальное приближение $x^{(0)} = \frac{\pi}{2} = 1,571$; $y^{(0)} = -\frac{\pi}{4} = -0,785$

№ Шага	$x^{(i)}$	$y^{(i)}$	$ x^{(i)} - x^{(i-1)} $	$ y^{(i)} - y^{(i-1)} $
1	1,354	-0,141	0,217	0,644
2	1,070	-0,238	0,284	0,097
3	1,118	-0,298	0,048	0,060
4	1,147	-0,293	0,029	0,005
5	1,144	-0,289	0,003	0,004

Таблица 4: Система нелинейных уравнений

Программная реализация задачи

Ссылка на исходный код к лабораторной работе

[Git Hub](#)

Основные методы программы

Метод половинного деления

```
1 def half_division(left, right, equation, accuracy):
2     counter = 0
3     search_value = define_method(equation)
4     while right - left > accuracy:
5         x = (right + left) / 2
6         if search_value(x) * search_value(left) > 0:
7             left = x
8         else:
9             right = x
10        counter += 1
11    return {"status" : 1, "root" : x, "value" : search_value(x),
12           "number_of_iterations" : counter}
```

Метод секущих

```
1 def secant(left, right, approach, equation, accuracy):
2     counter = 0
3     search_value = define_method(equation)
4     search_value_second_derivative =
5     define_method_for_second_derivative(equation)
6     preprevious = None
7     previous = None
8     if search_value(left) * search_value_second_derivative(left) > 0:
9         preprevious = left
10    else:
11        preprevious = right
12
13    if approach != "" and abs(preprevious - approach) > accuracy:
14        previous = approach
15    elif preprevious == left:
16        previous = preprevious + 0.2
17    else:
18        previous = preprevious - 0.2
19
20    while abs(preprevious - previous) > accuracy:
21        x = previous - (previous - preprevious) * search_value(previous)
22        / (search_value(previous) - search_value(preprevious))
23        counter += 1
24        preprevious = previous
25        previous = x
26
27    return {"status" : 1, "root" : previous, "value" :
28    search_value(preprevious), "number_of_iterations" : counter}
```

Метод простых итераций

```
1 def simple_iterations_equation(left, right, approach, equation, accuracy):
2     search_value = define_method(equation)
3     search_first_derivative = define_method_for_first_derivative(equation)
4     counter = 1
5     x = None
6     previous = None
7     if approach != "":
8         previous = approach
9         if search_first_derivative(previous) > 0:
10             gamma = -1 / search_max_derivative(left, right, equation)
11             .get('max_value')
12             else:
13                 gamma = 1 / search_max_derivative(left, right, equation)
14             .get('max_value')
15             else:
16                 if abs(search_first_derivative(left)) >=
17 abs(search_first_derivative(right)):
18                 previous = left
19                 else:
20                 previous = right
21                 if search_first_derivative(previous) > 0:
22                 gamma = -1 / search_max_derivative(left, right, equation)
23             .get('max_value')
24             else:
25                 gamma = 1 / search_max_derivative(left, right, equation)
26             .get('max_value')
27             x = gamma * search_value(previous) + previous
28             if abs(gamma * search_first_derivative(left) + 1) > 1
29 or abs(gamma * search_first_derivative(right) + 1) > 1:
30                 return {"status" : 0, "error" : "К сожалению этот метод не сходится,
31 попробуйте уменьшить промежуток поиска корня"}
32             while abs(x - previous) > accuracy:
33                 previous = x
34                 x = gamma * search_value(previous) + previous
35                 counter += 1
36             return {"status" : 1, "root" : x, "value" : search_value(x),
37 "number_of_iterations" : counter}
```

Вывод

В ходе выполнения лабораторной работы было изучено несколько итерационных методов решения нелинейных уравнений, а также два метода для решения системы нелинейных уравнений. Была реализована программа на языках программирования Python и Java Script. Были проанализированы условия возможности применения тех или иных численных методов, их достоинства и недостатки.

Анализ методов

Метод Половинного деления - прост в реализации, однако имеет линейную сходимость.

Метод Хорд - прост в реализации. Порядок сходимости метода хорд выше, чем у метода половинного деления.

Метод Ньютона - использует касательную для нахождения корня. Скорость сходимости: быстрая, квадратичная.

Метод Половинного деления - скорость сходимости зависит от выбора функции $\phi(x)$ и начального приближения.

Метод Секущих - имеет меньший объем вычислений по сравнению с методом Ньютона, т. к. не нужно вычислять производную, однако порядок сходимости ниже, чем у метода касательных и равен золотому сечению.