МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

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 №7

Выполнила:
Конаныхина Антонина
P3215
Преподаватель:
Малышева Татьяна
Алексеевна

Санкт-Петербург, 2022

Цель работы

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

Задание:

Вычислительная реализация задачи:

Уточнить корни нелинейного уравнения с точностью $\varepsilon = 10^{-2}$. Вычисления оформить в виде таблиц, удержать 3 знака после запятой.

Программная реализация задачи:

Для нелинейных уравнений:

1. Все численные методы должны быть реализованы в виде отдельных подпрограмм или 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. Для метода Ньютона (метода секущих) – выбор начального приближения (a или b). Для метода простой итерации – достаточное условие сходимости метода. Программа должна реагировать на некорректные введенные данные.
5. Предусмотреть вывод результатов (найденный корень уравнения, значение функции в корне, число итераций) в файл или на экран.
6. Организовать вывод графика функции, график должен полностью отображать весь исследуемый интервал (с запасом).

Для систем нелинейных уравнений:

1. Рассмотреть систему двух уравнений.
2. Организовать вывод графика функций.
3. Для метода простой итерации проверить достаточное условие сходимости.
4. Вывод вектора неизвестных: x_1, x_2.
5. Вывод количества итераций, за которое было найдено решение.
6. Вывод вектора погрешностей: $\left|x_i^{(k)} - x_i^{(k-1)}\right|$

Рабочие формулы используемых методов

Вариант №7:

Решение нелинейных уравнений:

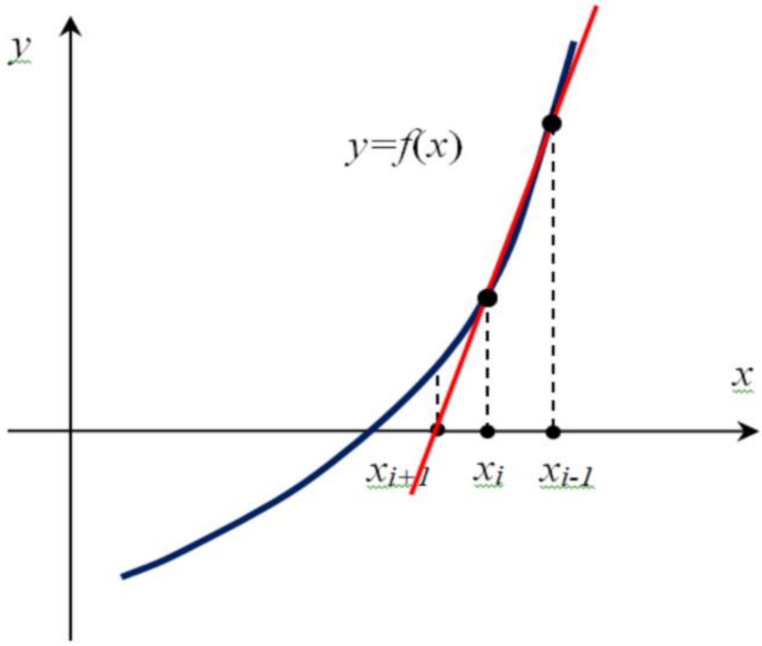
– **Метод секущих**

Рабочая формула: получается из упрощения метода Ньютона заменой производной разностным приближением:

$$x_{i+1} = x_i - \frac{(x_i - x_{i-1})f(x_i)}{f(x_i) - f(x_{i-1})}, \text{где } i = 1, 2, \ldots$$

Метод секущих является двухшаговым, т. е. новое приближение x_{i+1} определяется двумя предыдущими итерациями x_iи x_{i-1}.
Выбор x_0 определяется как и в методе Ньютона, x_1 – выбирается рядом с начальным самостоятельно.

Визуализация:

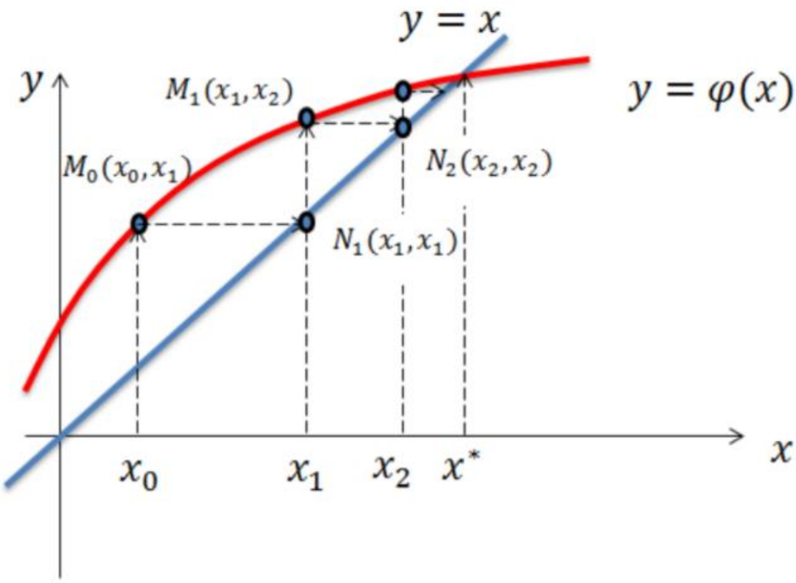


Критерий окончания итерационного процесса:
$|x_n - x_{n-1}| \leq \ \varepsilon$

– **Метод простой итерации**

<u>Рабочая формула метода:</u>
$x_{i+1} = \ \varphi(x_i)$

Геометрический смысл:



Уравнение $f(x) = 0$ приводится к эквивалентному виду: $x = \varphi(x)$, выражая x из исходного уравнения.
Через начальное приближение: $x_0 \in$ a, b, находятся очередные приближения:

$$x_1 = \varphi(x_0) \to x_2 = \varphi(x_1) \dots$$

Достаточное условие сходимости метода:

$\varphi'(x) \leq q \leq 1,$ где q – некоторая константа.

Критерий окончания итерационного процесса:

$|x_n - x_{n-1}| \leq \varepsilon$ (при $0 < q \leq 0{,}5$)

Решение систем нелинейных уравнений:

– **Метод Ньютона**

К основе метода лежит использование разложения функций $F_i(x_1, x_2, \dots, x_n)$ в окрестности некоторой фиксированной точки в ряд Тейлора, причем члены, содержащие вторые (и более высоких порядков) производные, отбрасываются.

Приведем систему уравнений к эквивалентному виду:

$$\begin{cases} F_1(x_1, x_2, \dots, x_n) = 0 \\ F_2(x_1, x_2, \dots, x_n) = 0 \\ \dots\dots\dots\dots\dots\dots\dots. \\ F_n(x_1, x_2, \dots, x_n) = 0 \end{cases} \qquad \begin{cases} x_1 = \varphi_1(x_1, x_2, \dots, x_n) \\ x_2 = \varphi_2(x_1, x_2, \dots, x_n) \\ \dots\dots\dots\dots\dots\dots\dots. \\ x_n = \varphi_n(x_1, x_2, \dots, x_n) \end{cases}$$

Или, в векторной форме: $\boldsymbol{X} = \boldsymbol{\varphi}(\boldsymbol{X}) \quad \boldsymbol{\varphi}(\boldsymbol{X}) = \begin{pmatrix} \varphi_1(\boldsymbol{X}) \\ \varphi_2(\boldsymbol{X}) \\ \dots\dots \\ \varphi_n(\boldsymbol{X}) \end{pmatrix}$

Если выбрано начальное приближение: $\boldsymbol{X}^{(0)} = x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}$, последующие приближения находятся по формулам:

$$\begin{cases} x_1^{(k+1)} = \varphi_1(x_1^k, x_2^k, \dots, x_n^k) \\ x_2^{(k+1)} = \varphi_2(x_1^k, x_2^k, \dots, x_n^k) \\ \dots\dots\dots\dots\dots\dots\dots. \\ x_n^{(k+1)} = \varphi_n(x_1^k, x_2^k, \dots, x_n^k) \end{cases} \qquad k = 0, 1, 2, \dots$$

Критерий окончания итерационного процесса:

$$\max_{1<i<n} \left| x_i^{(k+1)} - x_i^k \right| \leq \varepsilon$$

Заполненные таблицы

№ итерации	x_k	$f(x_k)$	x_{k+1}	φ(xk)	$\lvert x_k - x_{k+1} \rvert$
1	-2.500	-0.447	-2.417	-2.417	0.0825
2	-2.417	-0.035	-2.411	-2.411	0.006

Таблица 1 - Уточнение крайнего правого корня методом половинного деления

№ шага	a	b	x	f(a)	f(b)	f(x)	|a-b|
1	1.000	1.500	1.250	-2.561	1.697	-0.800	0.500
2	1.250	1.500	1.375	-0.809	1.697	0.344	0.250
3	1.250	1.375	1.313	-0.809	0.344	-0.257	0.125
4	1.313	1.375	1.344	-0.257	0.344	0.037	0.060
5	1.316	1.344	1.328	-0.257	0.037	-0.111	0.031
6	1.328	1.344	1.336	-0.111	0.037	-0.037	0.016

Таблица 2 - Уточнение крайнего левого корня методом простой итерации

№ итерации	x_k	$f(x_k)$	$f'(x_k)$	x_{k+1}	$\lvert x_k - x_{k+1} \rvert$
1	-1.000	-0.693	-3.494	-1.198	0.198
2	-1.198	-0.036	-3.090	-1.210	0.012
3	-1.210	-0.001	-3.059	-1.210	0.001

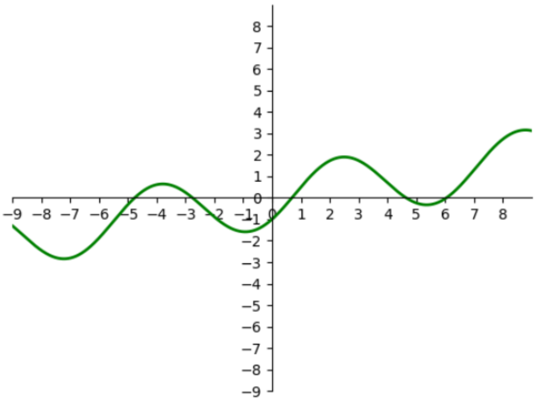
Таблица 3 - Уточнение центрального корня методом Ньютона

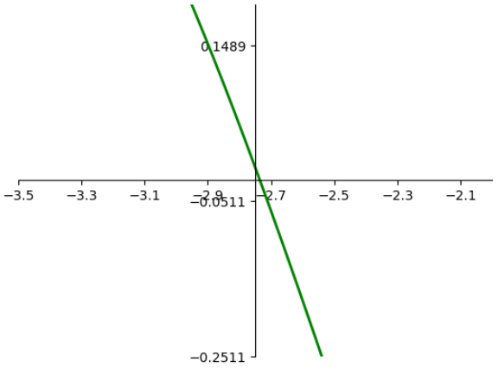
Листинг программы

https://github.com/tchn11/ITMO-labs/tree/main/2nd%20year/computational%20mathematics/nonlinear_equations

Результаты выполнения программы:

```
Введите 1, чтобы выбрать одно уравнение, 2, чтобы выбрать систему уравнений: 1
1 - x^3 + 2.28x^2 - 1.934x - 3.907
2 - x^2 - 3x - 2
3 - sin(x) - cos(x) + 0.2x
Введите номер желаемой функции: 3
Введите 1, чтобы ввести интервал, 2, чтобы запустить автоматический поиск интервала: 2
Найден интервал: [-5.0, -4.5]
Введите 1, чтобы найти следующий интервал, 2, чтобы выбрать этот интервал: 1
Найден интервал: [-3.0, -2.5]
Введите 1, чтобы найти следующий интервал, 2, чтобы выбрать этот интервал: 2
a = -3.0, b = -2.5
Введите точность: 0.001
Введите 1, чтобы выбрать метод секущих, 2 чтобы выбрать метод простой итерации: 2
Корень: -2.756818112848826 найден за 4 итераций, f(x) = 0.00016928791576364954
```





Вывод:

В результате выполнения данной лабораторной работой были изучены численные методы решения нелинейных уравнений и реализованы метод секущих, метод простой итерации и метод Ньютона на языке программирования Python.