МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования «Казанский национальный исследовательский

технический университет им. А. Н. Туполева-КАИ»

(КНИТУ-КАИ)

Институт компьютерных технологий и защиты информации

(наименование института (факультета), филиала)

Кафедра Прикладной математики и информатики

(наименование кафедры)

(09.03.04) Программная инженерия

(шифр и наименование направления подготовки (специальности))

Отчет по лабораторным работам по теме «Аналитическое моделирование»

Дисциплина: «Компьютерное моделирование процессов и систем»

Лабораторная работа №2

Вариант 1

Выполнил: студент группы 4411

Хасаншин Ш.Р

Казань, 2024

Лабораторная работа №2

Статичный аналитические модели, описываемые уравнениями

Задание

По экспериментальным данным построена аналитическая модель зависимости содержания цинка в волосах населения (*Zn\_v* мкг/г) от его содержания в питьевой воде (*Zn\_vod* мг/л) и снеге (*Zn\_s/l* мг/л растопленного снега).

*Zn\_v* = 4334.3 – 68193.5 \* *Zn\_vod* + 183.6 \* *Zn\_s/l* + 644657.2 \* *Zn\_vod*2 – 134.2 \* *Zn\_s/l*2 + 806.4 \* *ln*(*Zn\_vod*) – 14.9 \* *ln*(*Zn\_s/l*)

Необходимо рассчитать максимально допустимое содержание цинка в питьевой

воде, если по санитарным нормам уровень цинка в волосах не должен превышать

200 мкг/г, а замер содержания цинка в снеге показал результат 0,375 мг/л.

1. Получить решение уравнения относительно выбранной искомой

переменной символьно.

2. Записать аналитическую модель в виде уравнения вида f(x)=0 с

указанными значениями коэффициентов.

3. Построить график полученного уравнения.

4. Определить примерную область поиска решения по построенному

графику.

5. Получить решение задачи наиболее подходящим методом.

6. Изменить точность вычислений и повторить решение. Сравнить

результаты.

Решение

Преобразуем исходное выражение в уравнение с тремя неизвестными:

*Zn\_v* = *y*

*Zn\_vod* = *x*

*Zn\_s/l* = *z*

*y* = 4334.3 – 68193.5 \* *x* + 183.6 \* *z* + 644657.2 \* *x*2 – 134.2 \* *z*2 + 806.4 \* *ln*(*x*) – 14.9 \* *ln*(*z*)

По условию задачи необходимо найти *xmax*, причем:

*y* ≤ 200

z = 0.375

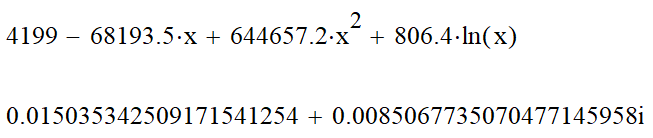
Следовательно, уравнение преобразуется в неравенство:

4334.3 – 68193.5 \* *x* + 183.6 \* 0.375 + 644657.2 \* *x*2 – 134.2 \* 0.3752 + 806.4 \* *ln*(*x*) – 14.9 \* *ln*(0.375) ≤ 200

В итоге:

4199 – 68193.5 \* *x* + 644657.2 \* *x*2 + 806.4 \* *ln*(*x*)

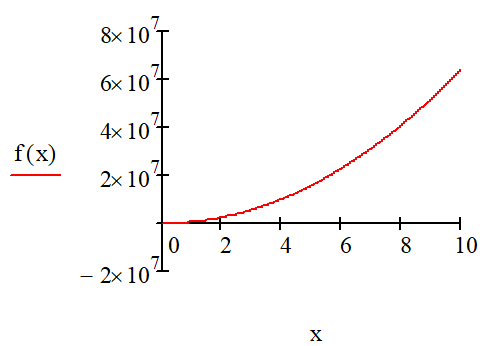
1. Получить решение уравнения относительно выбранной искомой переменной символьно.



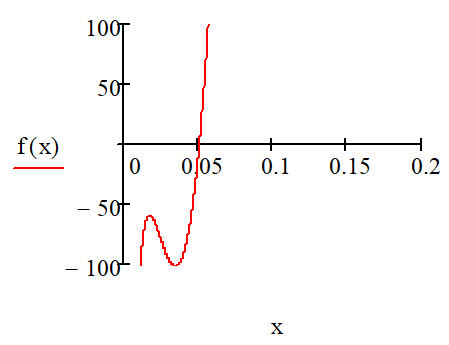
2. Записать аналитическую модель в виде уравнения вида f(x) = 0 с указанными значениями коэффициентов.



3. Построить график полученного уравнения.



4. Определить примерную область поиска решения по построенному графику.

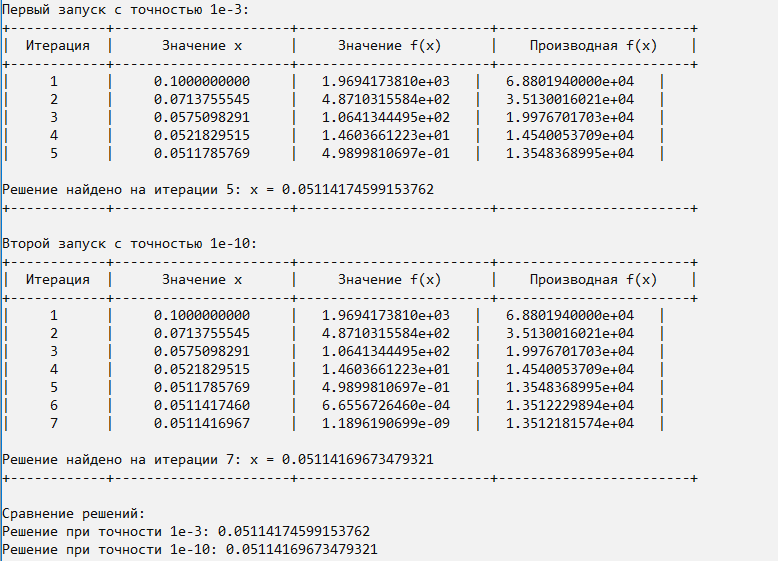


Примерная область: (0; 0.1).

5. Получить решение задачи наиболее подходящим методом.

6. Изменить точность вычислений и повторить решение. Сравнить результаты.

Изменим TOL с 10-3 на 10-10:



Вывод: содержание цинка в питьевой воде находится в интервале (0; 0.05114], следовательно, максимально допустимое содержание цинка в питьевой воде равно 0.05114.

Приложение 1

Код программы:

import math

# Определяем функцию для содержания цинка

def zinc\_function(x):

return 4199 - 68193.5 \* x + 644657.2 \* x\*\*2 + 806.4 \* math.log(x)

# Производная функции для метода Ньютона

def zinc\_function\_derivative(x):

return -68193.5 + 2 \* 644657.2 \* x + 806.4 / x

# Функция для форматирования вывода итераций

def display\_iteration(iteration, x, fx, fpx):

print(f"| {iteration:^10} | {x:^20.10f} | {fx:^20.10e} | {fpx:^20.10e} |")

# Реализация метода Ньютона

def newton\_method(func, func\_derivative, start\_value, tolerance=1e-6, max\_iterations=100):

x\_current = start\_value

iteration\_count = 0

# Заголовок таблицы для вывода

print(f"+{'-'\*12}+{'-'\*22}+{'-'\*24}+{'-'\*24}+")

print(f"| {'Итерация':^10} | {'Значение x':^20} | {'Значение f(x)':^22} | {'Производная f(x)':^22} |")

print(f"+{'-'\*12}+{'-'\*22}+{'-'\*24}+{'-'\*24}+")

while iteration\_count < max\_iterations:

fx\_current = func(x\_current)

fpx\_current = func\_derivative(x\_current)

# Вывод значений текущей итерации

display\_iteration(iteration\_count + 1, x\_current, fx\_current, fpx\_current)

# Проверка на деление на ноль или очень малое значение производной

if abs(fpx\_current) < 1e-12:

print("Ошибка: производная слишком близка к нулю, метод не может продолжиться.")

return None

# Вычисление нового значения x

x\_next = x\_current - fx\_current / fpx\_current

# Проверка критерия сходимости

if abs(x\_next - x\_current) < tolerance:

print(f"\nРешение найдено на итерации {iteration\_count + 1}: x = {x\_next:.17f}")

print(f"+{'-'\*12}+{'-'\*22}+{'-'\*24}+{'-'\*24}+")

return x\_next

# Переход к следующей итерации

x\_current = x\_next

iteration\_count += 1

print(f"+{'-'\*12}+{'-'\*22}+{'-'\*24}+{'-'\*24}+")

print("Максимальное количество итераций достигнуто.")

return x\_current

# Начальное приближение

initial\_guess = 0.1

# Первый запуск с точностью 1e-3

print("Первый запуск с точностью 1e-3:")

solution\_1 = newton\_method(zinc\_function, zinc\_function\_derivative, initial\_guess, tolerance=1e-3)

# Второй запуск с точностью 1e-10

print("\nВторой запуск с точностью 1e-10:")

solution\_2 = newton\_method(zinc\_function, zinc\_function\_derivative, initial\_guess, tolerance=1e-10)

# Сравнение результатов

print("\nСравнение решений:")

print(f"Решение при точности 1e-3: {solution\_1}")

print(f"Решение при точности 1e-10: {solution\_2}")