**Оглавление**

[**Глава 1: Спецификация проблемы** 1](#_Toc492385228)

[**1.1. Задание:** 1](#_Toc492385229)

[**Глава 2: Описание Bisection Method** 2](#_Toc492385230)

[**Глава 3: Спецификация требований к программному обеспечению** 2](#_Toc492385231)

[**3.1. Наименование программной разработки** 2](#_Toc492385232)

[**3.2. Бизнес-цель разработки программного обеспечения** 2](#_Toc492385233)

[**3.5. Функциональные требования к разработке программного обеспечения** 3](#_Toc492385234)

[**3.6. Нефункциональные требования на разработку программного обеспечения** 4](#_Toc492385235)

[**3.7. Ограничения на разработку программного обеспечения** 4](#_Toc492385236)

# **Глава 1: Спецификация проблемы**

Задача: Разработать программу, которая находит корень произвольного нелинейного уравнения – ноль нелинейной функции f(x) с заданной допустимой погрешностью Tolerance не менее 1E–28 методом деления отрезка пополам (Bisection Method).

Проблема данной задачи отражается в затрате времени и денег на дополнительных сотрудников.

# **1.1. Задание:**

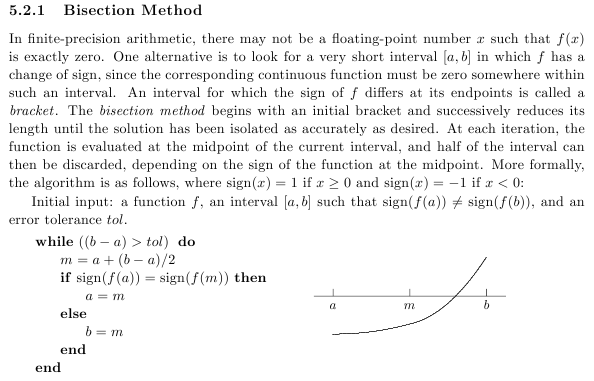
Задача №27:

Изучить тему \_\_ из главы №1 учебника [1, 2].

Разработать программу для определения корня произвольного нелинейного уравнения f(x)=0. Эта программа должна методом Bisection Method найти корень уравнения f(x)=0 с допустимой погрешностью не менее 1E–28. Задание:

1. Разработать входной интерфейс программы.
2. Разработать блок-схему алгоритма вычисления.
3. Разработать выходной интерфейс программы.
4. Продемонстрировать валидность программы.
5. Разработать программу для поиска корней системы нелинейных уравнений.
6. Составить перечень функциональных требований, реализованных в программе.

# **Глава 2: Описание Bisection Method**



# **Глава 3: Спецификация требований к программному обеспечению**

## **3.1. Наименование программной разработки**

Система поиска корней нелинейного уравнения методом деления отрезка пополам (Bisection Method)

## **3.2. Бизнес-цель разработки программного обеспечения**

Минимизировать затраты времени на поиск корня нелинейного уравнения ***до одной минуты*** при максимально высокой точности решения – допустимая погрешность решения должна быть ***не менее 1e-28 (т.е., 0.0000000000000000000000000001)***, что позволит сократить ***две ставки*** специалиста в отделе исследований, поскольку теперь с такой задачей может справиться уже один специалист с помощью разрабатываемой системы.

**3.3. Разработка пользовательских требований**

Польза от компьютерной системы, поиска корней нелинейного уравнения методом деления отрезка пополам, является достижение более высокой точности с минимальной затратой времени и исключает надобность в дополнительных сотрудниках. Пользовательскими требованиями к ПО являются следующее:

* Для достижения бизнес-цели разработки программы нужно добиться необходимой точности решения за заданное время и количество итераций.
* Необходимо удостовериться в достижении заданной точности.

**3.4. Пользовательская история разработки программного обеспечения**

***Пользователь:***

* должен вводить нелинейную функцию ***f(x)***;
* значения концов интервала ***[a, b]***;
* значение допустимой погрешности ***Tolerance*** решения задачи;
* допустимое максимальное количество ***k\_max*** итераций;
* время, за которое производится введённое количество итераций.

***Программа:***

* должна провести parsing – синтаксический анализ выражения введенной нелинейной функции и преобразовать аналитическую запись нелинейной функции в виде символов в вычисляемую процедуру в виде подпрограммы-функции;
* должна дать возможность ввести числовые значения концов интервала ***[a, b]*** и преобразовать их в числовые значения концов интервала;
* должна найти решение задачи – корень нелинейного уравнения ***f(x)=0***;
* должна выводить найденное значение ***х*** корня заданного нелинейного уравнения;
* должна выводить значение заданной нелинейной функции ***f(x)*** в этой точке;
* должна выводить общее затраченное время (в миллисекундах) на поиск решения ***Time(msec)*** для оценки производительности программы;
* должна выводить затраченное на поиск решения с заданной погрешностью количество итераций ***K***;
* должна выводить достигнутое значение абсолютной ошибки решения ***Abs(b–a)*** в формате 0Е0.



## **3.5. Функциональные требования к разработке программного обеспечения**

* Система должна использовать ***Bisection Method для поиска корней*** нелинейного уравнения:
  + Система должна искать решение задачи по следующему алгоритму (Bisection algorithm):
  + Система должна обеспечивать ввод любых начальных числовых значений концов интервала ***[a; b]***;
  + Система должна обеспечить проверку условия, что левый конец интервала меньше правого конца, то есть a < b;
  + Система должна обеспечивать проверку условия . Выполнение этого условия гарантирует корректность значений, введенных начальных числовых значений концов интервала ***[a; b]***, которые являются необходимым условием для правильной работы Bisection Method;
* Система должна обеспечивать возможность ввода исходных данных имеющих следующие особенности:
  + Система должна обеспечивать возможность ввода аналитического выражения для любой нелинейной функции;
  + Система должна проводить parsing – синтаксический анализ аналитического выражения нелинейной функции f(x) – и автоматически конструировать подпрограмму для вычисления значения этой функции для любого значения аргумента ***x*** этой функции;
  + Система должна обеспечить ввод любого целого числового значения в качестве максимального допустимого количества итераций k\_max;
  + Система должна обеспечивать возможность ввода любых числовых значений допустимой погрешности Tolerance как в формате с фиксированной точностью “0,000000000000001”, так и в экспоненциальном формате “1e-15”;
  + Система должна быть способна проверять валидность формата введенных исходных числовых данных и сообщать пользователю об ошибке в формате исходных данных;
* Система должна обеспечить вывод следующих результатов решения задачи:
  + Система должна обеспечить вывод результата решения задачи в виде числа x – корня нелинейного уравнения или нуля нелинейной функции;
  + Система должна обеспечить вывод значения функции в этой точке f(x);
  + Система должна обеспечить вывод количества итераций, которые потребовались системе, чтобы найти решение задачи с требуемой погрешностью в качестве критерия производительности системы;
  + Система должна обеспечить вывод достигнутого значения погрешности решения задачи в качестве оценки точности этого метода (Bisection method);
  + Система должна обеспечить невозможность какого-либо исправления вручную выходных данных системы;
* Система должна обеспечить оценку производительности программы при решении задачи заданным методом (Bisection Method).

## **3.5. Нефункциональные требования на разработку программного обеспечения**

* Программа в процессе поиска решения должна отображать индикатор состояния процесса выполнения операции Progress Bar, чтобы пользователь понимал, что система не зависла, а находится в рабочем состоянии;
* В случае ошибки во введенном аналитическом выражении нелинейной функции f(x) система должна предоставлять достаточно подробную информацию об ошибке, которая будет облегчать поиск источника ошибки;
* Система должна обеспечить возможность очищения текстовых полей выходного интерфейса от результатов предыдущих вычислений при новом запуске вычислений. При этом данные в текстовых полях входного интерфейса должны сохраняться. Это позволит пользователю увидеть более наглядно, что получены новые результаты, соответствующие измененным входным данным;
* Сбой системы не должен происходить чаще чем один раз на тысячу запусков программы;
* Система должна обеспечить оценку качества решения задачи заданным методом (Bisection Method) путем измерения и вывода количества итераций, потраченных на поиск решения с заданной погрешностью;

## **3.6. Ограничения на разработку программного обеспечения**

* Нелинейная функция ***f(x)*** должна быть непрерывной на интервале [a; b];
* Разработать ПО на языке C++ / C# / Visual Basic;
* Система должна решать задачи с допустимой погрешностью не более 1e-15;
* Система должна решать задачи за время не более одной минуты;
* Временные затраты на разработку ПО не должны превышать одного месяца;
* Финансовые затраты на разработку ПО и соответствующую документацию (техническое задание на ПО, руководство пользователя и руководство программиста) не должны превышать $1000;
* Система должна быть разработана для эксплуатации на компьютерах типа пентиум с оперативной памятью не более 2 гигабайт;
* Разработка программы должна начаться не позже 15 июня 2017 года, и система должна быть передана заказчику не позднее 1июля 2017 года;
* Система должна позволять пользователю решать не менее 100 задач поиска корня нелинейного уравнения за один час для любой нелинейной функции, описываемой в виде аналитического выражения, корректной с точки зрения математики;
* Система должна понимать и обрабатывать любые аналитические выражения для нелинейной функции, которые включают следующие математические символы:
  + Стандартных функций *– sin, cos, tg, ctg, arcsin, arccos, arctg, arcctg, sh, ch, th, cth, exp, lоg, ln, sqrt, fac, sinh, cosh, tanh, abs, ceil, floor, sfac, round, fpart.*
  + Основных вычислительных операций: сложение (+), вычитание (-), умножение (\*), деление (/) и возведение в степень (^);
  + круглых и квадратных скобок любой вложенности;
* Система не предназначена для ввода логических и алгоритмически заданных нелинейных функций, потому что понятие «корень нелинейного уравнения» относится к понятию числа, который несовместим с понятием логических значений «ложь-истина» вышеуказанных нелинейных функций.

Алгоритм формирования система поиска корней нелинейного уравнения методом деления отрезка пополам (Bisection Method)

