|  |
| --- |
|  |
| Техническое задание |
| На лабораторную работу №2 «NewtonMethod» по предмету «Основы разработки и анализа требований к ПО» |
|  |
| **Байгазиев Санжар** |
| **ПИ-2-15** |

|  |
| --- |
|  |

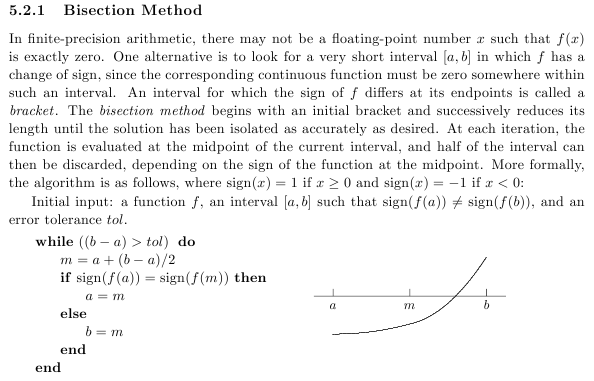
СОДЕРЖАНИЕ

1. [Глава 1: Описание проблемы](#Глава1ОписаниеПроблемы)
2. [Глава 2: Описание Bisection Method](#Глава2ОписаниеBisectionMethod)
3. [Глава 3: Спецификация требований к программному обеспечению](#Глава3СпецификацияТребований_к_ПО)
   1. [Наименование разработки](#НаименованиеРазработки)
   2. [Описание бизнес-процесса](#ОписаниеБизнесПроцесса)
   3. [Бизнес-цель разработки программного обеспечения](#БизнесЦельРазработки)
   4. [Пользовательская история разработки программного обеспечения](#ПользовательскаяИсторияПО)
   5. [Функциональные требования к разработке программного обеспечения](#ФункциональныеТребованияПО)
   6. [Нефункциональные требования на разработку программного обеспечения](#НеФункциональныеТребованияПО)
   7. [Ограничения на разработку программного обеспечения](#ОграниченияНаПО)
   8. [Требования-«бантики», украшательство ~ gold plating](#ТребованияБантикиНаПО)
4. [Глава 4: Interface of Bisection method’s Program](#Глава4ИнтерфейснаяФорма)
5. [Список литературы](#Глава6СписокЛитературы)

Глава 1: Описание проблемы

Найти корень произвольного нелинейного уравнения – ноль нелинейной функции f(x) с заданной допустимой погрешностью Tolerance не менее величины 1E–28 методом деления отрезка пополам (Bisection Method). Нелинейная функция f(x) имеет произвольный аналитический вид, составленный из математических функций (полиномов различных степеней, тригонометрических – sin(x), cos(x), exp(x), ln(x), log(x) и. т. д.), которая имеет математический смысл, и для которой существует хотя бы одно решение задачи. Описание метода деления отрезка пополам – Bisection Method – приведено по книге “SCIENTIFIC COMPUTING. An Introductory Survey. Michael T. Heath. University of Illinois at Urbana-Champaign. 1997 by The McGraw-Hill Companies. ISBN 0-07-027684-6”, стр. 154.

Глава 2: Описание Newton Method



[GoTo Contents](#Contents)

Глава 3: Спецификация требований к программному обеспечению

3.1. Наименование программной разработки

Система поиска корней нелинейного уравнения методом деления отрезка пополам

3.2. Описание бизнес-процесса и анализ расхождения

При решении различных бизнес-задач отделу исследований (пользователям) требуется находить корень произвольного нелинейного уравнения с высокой точностью. Для решения таких задач в этом отделе имеется 5 ставок специалистов. Два из них занимаются формализацией задачи и определением/нахождением вида нелинейной функции, описывающей как можно точнее модель поведения реальной системы, а остальные занимаются решением задачи – поиском корня нелинейного уравнения для найденной нелинейной функции. При больших трудозатратах (заняты два специалиста высокой квалификации), не всегда обеспечивается приемлемая и единообразная/унифицированная точность решения задачи. Бизнесу желательно было бы гарантированно обеспечить любую допустимую погрешность решения задачи при любых видах нелинейной функции f(x) с минимальными трудозатратами.

3.3. 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.

3.4. Пользовательская история разработки программного обеспечения

***Пользователь:***

* должен вводить аналитическое выражение для требуемой нелинейной функции ***f(x)***;
* значения концов интеравала ***[a, b]***;
* значение допустимой погрешности ***Tolerance*** решения задачи;
* допустимое максимальное время поиска решения задачи ***t\_max*** seconds.
* допустимое максимальное количество ***k\_max*** 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 ввести символьные значения концов интервала ***[a, b]*** и преобразовать их в числовые значения концов интеравала;
* должна найти решение задачи – корень нелинейного уравнения ***f(x)=0*** или ноль нелинейной функции ***f(x)*** – по алгоритму Bisection Method;
* должна выводить найденное значение ***x\**** корня заданного нелинейного уравнения;
* должна выводить значение заданной нелинейной функции ***f(x\*)*** в этой точке;
* должна выводить общее затраченное время (в миллисекундах / seconds) на поиск решения ***Elapsed Time*** для оценки производительности программы;
* должна выводить затраченное на поиск решения с заданной погрешностью количество итераций ***Amount Of Iterations***;
* должна выводить достигнутое значение абсолютной ошибки решения ***Abs(b–a)***.

3.5. Функциональные требования к разработке программного обеспечения

* 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 символьного значения*** аргумента этой функции;

В качестве символьного выражения функции так и аргумента функции могут быть заданы ***любые*** символы латинского алфавита как ***строчные*** (lower-case letter) так и ***прописные*** (upper-case letter).

* 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 обеспечивать проверку условия . Выполнение этого условия гарантирует корректность значений введенных начальных числовых значений концов интервала ***[a; b]***, которые необходимы для правильной работы Bisection Method;
  + Система должна распознавать ситуации , которые недопустимы для корректного применения метода – Bisection Method – поиска корней нелинейного уравнения, и сообщать пользователю о возникновении подобной ситуации;
  + Система должна обеспечить для пользователя возможность продолжения поиска решения задачи с последних достигнутых до выполнения условия ***ElapsedTime>=t\_max*** концов интервала ***[a(k), b(k)]***, приняв их за начальные значения концов интервала {т.е., приняв a(0)=a(k) и b(0)=b(k) система должна заново затратить не менее ***t\_max*** времени}, если за данное количество времени ***t\_max*** не достигнуто решение с требуемой погрешностью ***Tolerance***;
  + Система должна предупреждать пользователя о том, что полученное решение задачи не удовлетворяет требуемой точности решения по причине заданного ограничения на длительность времени работы программы;
  + Система должна обеспечить для пользователя возможность продолжения поиска решения задачи с последних достигнутых на шаге ***k\_max*** концов интервала ***[a(k\_max), b(k\_max)]***, приняв их за начальные значения концов интервала {т.е., приняв a(0)=a(k\_max) и b(0)=b(k\_max) система должна заново выполнить не менее ***k\_max*** итераций}, если за данное количество итераций ***k\_max*** не достигнуто решение с требуемой погрешностью ***Tolerance***;
  + Система должна предупреждать пользователя о том, что полученное решение задачи не удовлетворяет требуемой точности решения по причине заданного ограничения по количеству итераций на поиск решения задачи;
* Система должна обеспечить ввод любого целого числового значения для максимального допустимого количества итераций ***k\_max***;
* Система должна обеспечевать возможность ввода любых числовых значений допустимой погрешности ***Tolerance*** как в формате с фиксированной точностью “0.000000000000001”, так и в экпоненциальном формате “1e-28”;
* Система должна обеспечить вывод результата решения задачи x\*;
* Система должна обеспечить вывод значения функции в этой точке f(x\*);
* Система должна обеспечить оценку производительности программы при решении задачи заданнымитерационным методом (***Bisection Method***) и реализации программы на языке ***Visual Basic / C# / C++ в среде Visual Studio 2012-2017*** путем измерения и вывода времени, затрачиваемой на поиск решения с заданной погрешностью;

3.6. Нефункциональные требования на разработку программного обеспечения

* Программа в процессе поиска решения должна отображать индикатор состояния процесса выполнения операции ***Progressive 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;
* 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;
* Система должна обеспечить возможность измерения ***скорости сходимости*** итерационного метода (линейная, суперлинейная или квадратичная скорости сходимости).

3.7. Ограничения на разработку программного обеспечения

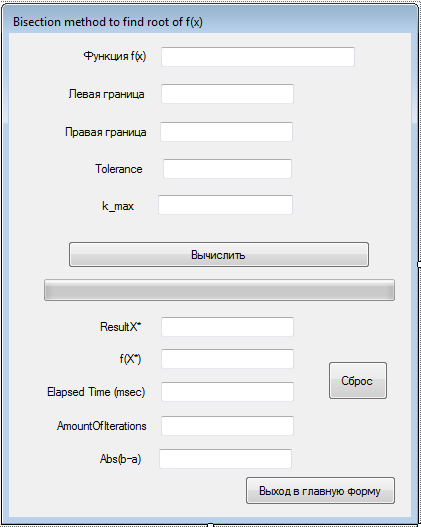
* Нелинейная функция ***f(x)*** должна быть непрерывной на интервале ***[a; b]***;
* Разработать ПО на основе IDE (integrated development environment – интегрированная среда разработки) – Visual Studio 2017 / Qt-Creator;
* Разработать ПО на языке C#;
* Система должна решать задачи с допустимой погрешностью 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 соответсвующую документацию (техническое задание на ПО, руководство пользователя и руководство программиста) не должны превышать $1000;
* Система должна быть разработана для эксплуатации на компъютерах типа пентиум с оперативной памятью не более 2 гигобайт;
* Разработка программы должна начаться не позже 16 сентября 2018 года и система должна быть передана заказчику не позднее 20 октября 2018 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 аналитические выражения для нелинейной функции, которые включают следующие математические символы:
  + Cтандартных функций – *sin, cos, tg, ctg, arcsin, arccos, arctg, arcctg, sh, ch, th, cth, exp, lg, ln, sqrt*;
  + Основных вычислительных операций: сложение (+), вычитание (-), умножение (\*), деление (/), остаток от деления (%) и возведение в степень (^);
  + круглых и квадратных скобок любой вложенности;
  + вектора переменных, на которые можно ссылаться, используя запись вида x[N] или x(N), где N - индекс переменной.
* Система не предназначена для ввода логических и алгоритмически заданных нелинейных функций.

3.8. Требования-«бантики», украшательство ~ gold plating

* Система должна играть марсельезу, когда количество итераций превысит максимально заданное число ***k\_max***;
* Система должна обеспечить решение задачи для любого значения заданной допустимой погрешности ***Tolerance***, независимо от разрядности памяти (memory width, digit capacity, ) процессора компьютора.

[GoTo Contents](#Contents)

Глава 4: Interface of Bisection method’s Program



[GoTo Contents](#Contents)