КЫРГЫЗСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. И. Раззакова

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Кафедра «ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРНЫХ

СИСТЕМ»

Дисциплина

Методы оптимизации

Отчет

по практической работе №1

«Разработка ПО для поиска ***корней нелинейного уравнения***

на основе итерационного метода: ***Bisection method***»

Выполнил: студент группы ПИ-2-16

Калыков Алишер

Проверил: кандидат технических наук, профессор

Тен Иосиф Григорьевич

Оглавление

[Раздел №1 Наименование работы 3](#_Toc532776548)

[Раздел №2 Спецификация проблемы 3](#_Toc532776549)

[Раздел №3 Спецификация метода 3](#_Toc532776550)

[Раздел №4 Стадии проектирования системы 4](#_Toc532776551)

[Раздел №5 Документирования этапов проектирования интерфейсной формы системы 6](#_Toc532776552)

[Раздел №6 Стадии конструирования ПО 9](#_Toc532776553)

[Раздел №7 Тестирование 12](#_Toc532776554)

# Раздел №1 Наименование работы

Разработка ПО для поиска ***корней нелинейного уравнения*** на основе итерационного метода: ***Bisection method.***

# Раздел №2 Спецификация проблемы

Найти корень произвольного нелинейного уравнения – ноль нелинейной функции ***f(x)*** с заданной допустимой погрешностью ***Tolerance*** 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.), который имеет математический смысл, и для которой существует хотя бы одно решение задачи. Полное описание метода деления отрезка пополам – Bisection Method – приведено по книге [3] “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.

# Раздел №3 Спецификация метода

In ﬁnite-precision arithmetic, there may not be a ﬂoating-point number x such that f(x) is exactly zero. One alternative is to look for a very short interval [a, b] in which f has a change of sign, since the corresponding continuous function must be zero somewhere within such an interval. An interval for which the sign of diﬀers at its endpoints is called a ***bracket***. The ***bisection method*** begins with an initial bracket and successively reduces its length until the solution has been isolated as accurately as desired. At each iteration the function is evaluated at the midpoint of the current interval, and half of the interval can then be discarded, depending on the sign of the function at the midpoint. More formally, the algorithm is as follows, where

Initial input: a function, an interval [a, b] such that and an error tolerance.

|  |  |
| --- | --- |
|  |  |

# Раздел №4 Стадии проектирования системы

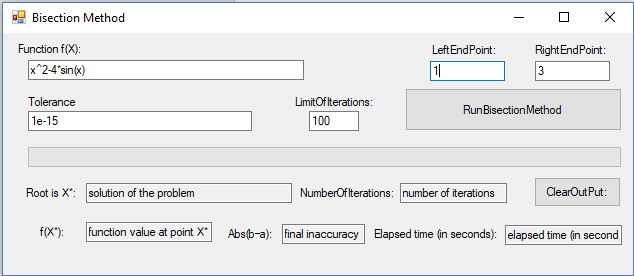
1. Разработка блок-схемы Bisection Method:



Figure 1: Flowchart of the Bisection Method with a design of the input and output interfaces

1. Разработка пользовательского интерфейса.

Форма для взаимодействия пользователя с программой представлена ниже:



# Раздел №5 Документирования этапов проектирования интерфейсной формы системы

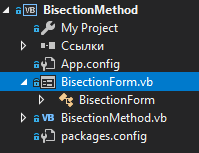
**ПРОЕКТИРОВАНИЕ По**: ДОКУМЕНТИРОВАНИЕ ЭТАПОВ ПРОЕКТИРОВАНИЯ ИнтерфейснОЙ формЫ системЫ, реализующЕЙ итерационный метод – Bisection Method – ДЛЯ поиска корнЕЙ нелинейного уравнения

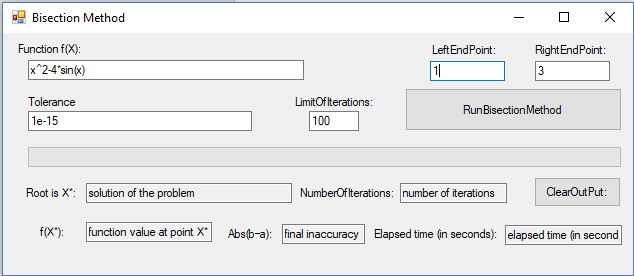
1. Таблица 1: Настройки элементов управления интерфейсной формы системы

|  |  |  |  |
| --- | --- | --- | --- |
| **Number of control** | **Control** | **Property** | **Setting** |
| 1 | Label | Appearance (Text) | Function f(x): |
| Design (Name) | Label1 |
| 2 | Textbox | Appearance (Text) | x^2-4\*sin(x) |
| Design (Name) | funcBox |
| 3 | Label | Appearance (Text) | LeftEndPoint: |
| Design (Name) | LeftEndPointBox |
| 4 | Textbox | Appearance (Text) | 1 |
| Design (Name) | LeftEndPointBox |
| 5 | Label | Appearance (Text) | RightEndPoint: |
| Design (Name) | LabelRightEndPoint |
| 6 | Textbox | Appearance (Text) | 3 |
| Design (Name) | RightEndPointBox |
| 7 | Label | Appearance (Text) | LimitOfIterations: |
| Design (Name) | LabelLimitOfIterations |
| 8 | Textbox | Appearance (Text) | 100 |
| Design (Name) | k\_maxBox |
| 9 | Label | Appearance (Text) | Tolerance |
| Design (Name) | LabelTolerance |
| 10 | Textbox | Appearance (Text) | 1e-15 |
| Design (Name) | ToleranceBox |
| 11 | Button | Appearance (Text) | RunBisectionMethod |
| Design (Name) | ButtonRunBisectionMethod |
| 12 | ProgressBar | Behavior (Visible) | False |
| Design (Name) | ProgressBar1 |
| 13 | Label | Appearance (Text) | Root is X\*: |
| Design (Name) | LabelRootOfEquation |
| 14 | Textbox | Design (Name) | SolutionOfTaskBox |
| Appearance (Text) | Solution of the problem |
| **Behavior (ReadOnly)** | True |
| 15 | Label | Appearance (Text) | f(X\*): |
| Design (Name) | LabelFunctionValue |
| 16 | Textbox | **Behavior (ReadOnly)** | True |
| Appearance (Text) | Function value at point X\* |
| Design (Name) | ValueOfFunctionBox |
| 17 | Label | Appearance (Text) | NumberOfIterations: |
| Design (Name) | Label NumberOfIterations |
| 18 | Textbox | **Behavior (ReadOnly)** | True |
| Appearance (Text) | Number of iterations |
| Design (Name) | NumberOfIterationsBox |
| 19 | Label | Appearance (Text) | Abs(b–a): |
| Design (Name) | LabelAbsError |
| 20 | Textbox | **Behavior (ReadOnly)** | True |
| Appearance (Text) | Final inaccurancy |
| Design (Name) | AbsErrorBox |
| 21 | Button | Appearance (Text) | ClearOutPut: |
| Design (Name) | ButtonClearOutPut |
| 22 | Textbox | **Behavior (ReadOnly)** | True |
| Appearance (Text) | Elapsed time(in second) |
| Design (Name) | elapsedTime |
| 23 | Label | Appearance (Text) | Elapsed time (in seconds): |
| Design (Name) | Label3 |
| 24 | Label | Appearance (Text) | empty |
| Design (Name) | Label4 |

# Раздел №6 Стадии конструирования ПО

1. Код программы на Visual Basic.NET, ***ассоцированный с интерфейсной формой*** “BisectionForm.vb”, который ***реализует функции ввода и вывода данных*** и составляет Public Class “BisectionForm”.





\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Outset of the “Public Class BisectionForm”

Option Explicit On

Imports System.Math

Imports info.lundin.math

Imports System.Threading

Public Class BisectionForm

Sub Clean()

SolutionOfTaskBox.Text = "Solution Of Task"

ValueOfFunctionBox.Text = "Value Of Function"

NumberOfIterationsBox.Text = "Number Of Iterations"

AbsErrorBox.Text = "final inaccuracy"

elapsedTime.Text = "Elapsed time (in seconds)"

Label4.Text = ""

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles ButtonRunBisectionMethod.Click

Dim started As DateTime = Now

Dim finished As DateTime

ProgressBar1.Value = 0

Try

If (funcBox.Text = "" Or LeftEndPointBox.Text = "" Or RightEndPointBox.Text = "" \_

Or ToleranceBox.Text = "" Or k\_maxBox.Text = "") Then

MsgBox("Input textboxes are empty! Enter the data")

Else

Clean()

Dim BM As BisectionMethod = New BisectionMethod()

Label2.Text = "Analytical expression of the function is: f(x) = " & funcBox.Text

BM.start(funcBox, LeftEndPointBox, RightEndPointBox, ToleranceBox,

k\_maxBox, ProgressBar1, Label4)

finished = Now

elapsedTime.Text = finished.Subtract(started).Seconds

BM.out(SolutionOfTaskBox, ValueOfFunctionBox, NumberOfIterationsBox, AbsErrorBox)

End If

Catch ex As ParserException

MsgBox("A mistake is in the analytical expression of the function f(x)")

Catch ef As FormatException

MsgBox("A mistake is in the format of the input data")

End Try

End Sub

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles ButtonClearOutPut.Click

Clean()

End Sub

Private Sub Form2\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Label2.Text = ""

SolutionOfTaskBox.Text = "solution of the problem"

ValueOfFunctionBox.Text = "function value at point X\*"

NumberOfIterationsBox.Text = "number of iterations"

AbsErrorBox.Text = "final inaccuracy"

elapsedTime.Text = "elapsed time (in seconds)"

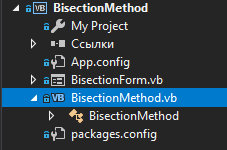
Label4.Text = ""

End Sub

End Class

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Ending of the “Public Class BisectionForm”

1. Код класса “BisectionMethod.vb”



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Outset of the “Public Class BisectionMethod”

Option Explicit On

Imports System.Math

Imports info.lundin.math

Imports System.Threading

Public Class BisectionMethod

Dim func As String

Dim k As Integer

Dim SolutionOfTask As Decimal

Dim ValueOfFunction As Decimal

Dim FinalInaccuracy As Decimal

Public Sub out(SolutionOfTaskBox As TextBox, ValueOfFunctionBox As TextBox,

NumberOfIterationsBox As TextBox, AbsErrorBox As TextBox)

SolutionOfTaskBox.Text = SolutionOfTask

ValueOfFunctionBox.Text = ValueOfFunction.ToString("0E0")

NumberOfIterationsBox.Text = k

AbsErrorBox.Text = FinalInaccuracy.ToString("0E0")

End Sub

Function F(par As Double) As Double

Dim Parser As New ExpressionParser()

Parser.Values.Add("x", par)

Return Parser.Parse(func)

End Function

Public Sub start(funcBox As TextBox, LeftEndPointBox As TextBox,

RightEndPointBox As TextBox,

ToleranceBox As TextBox, k\_maxBox As TextBox,

ByRef ProgressBar1 As ProgressBar, ByRef Label4 As Label)

Dim Tolerance As Double

Dim a, b As Decimal

Dim mid As Decimal

Dim i\_Max, k\_max As Integer

func = funcBox.Text

a = Decimal.Parse(LeftEndPointBox.Text)

b = Decimal.Parse(RightEndPointBox.Text)

Tolerance = Double.Parse(ToleranceBox.Text)

k\_max = Integer.Parse(k\_maxBox.Text)

k = 0

If Math.Sign(F(a)) = Math.Sign(F(b)) Then

MsgBox("Sign(F(a))= " & Sign(F(a)) & " and Sign(F(b))= " &

Sign(F(b)) & " must be opposite! Check endpoints of the interval [a, b]!")

Else

While k < k\_max

mid = a + (b - a) / 2

If Math.Sign(F(a)) = Math.Sign(F(mid)) Then

a = mid

Else

b = mid

End If

FinalInaccuracy = Abs(b - a)

k = k + 1

i\_Max = CInt(Math.Log((b - a) / Tolerance))

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

If k >= k\_max Then

Label4.Text = "Attention: It isn't possible to find a solution with the given Tolerance = " & Tolerance

End If

If FinalInaccuracy <= CDec(Tolerance) Then

Label4.Text = "The root of a nonlinear equation found with the given Tolerance = " & Tolerance

Exit While

End If

ProgressBar1.Visible = True

ProgressBar1.Maximum = k + 0.00000001

ProgressBar1.Value = k

Thread.Sleep(50)

End While

ProgressBar1.Visible = False

SolutionOfTask = mid

ValueOfFunction = F(mid)

End If

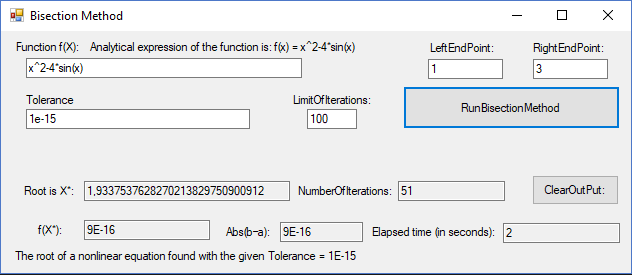
End Sub

End Class

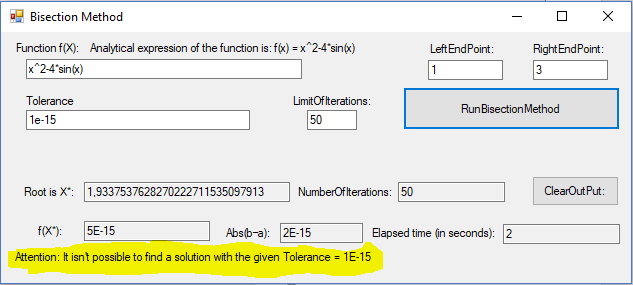
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Ending of the “Public Class BisectionMethod”

# Раздел №7 Тестирование

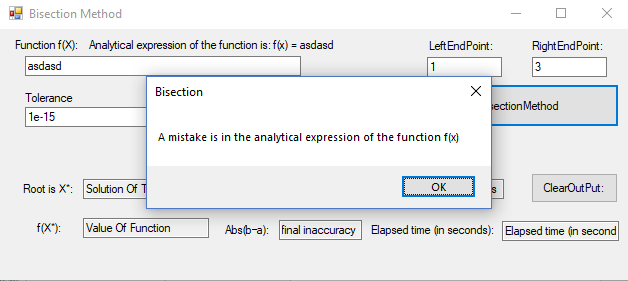
1. Тест №1 функция: x^2-4\*sin(x)
   1. Найдено корректное решение



* 1. Не найдено решение за указанное количество итерации.



* 1. Ошибка в выражении функции f(x)



* 1. Ошибка в формате ввода

