Содержание

[Введение 4](#_Toc22546896)

[1 Описание метода решения задачи 5](#_Toc22546897)

[1.1 Корни уравнения 5](#_Toc22546898)

[1.2 Отделение корней 6](#_Toc22546899)

[1.3 Численное решение уравнения методом половинного деления 6](#_Toc22546900)

[2 Разработка структур данных 9](#_Toc22546901)

[3 Разработка алгоритмов 10](#_Toc22546902)

[4 Тестирование программы 11](#_Toc22546903)

[5 Разработка документации 12](#_Toc22546904)

[Заключение 13](#_Toc22546905)

[Библиографический список 14](#_Toc22546906)

[Приложение А. Листинг программы 15](#_Toc22546907)

Введение

Целью курсового проекта является разработке программной системы для решения математической задачи уточнения корней уравнения численными методами (методом деления пополам).

Задачами курсового проектирования является:

* изучение метода решения задачи;
* разработка алгоритма и структур данных;
* разработка программы, позволяющей решить задачу уточнения корней уравнения;
* выполнить контрольный расчет в ручном режиме;
* провести тестирование работы программы на контрольном примере;
* выполнить тестирование работы программы на нескольких тестовых примерах;
* разработать документацию к программе.

Пояснительная записка к курсовому проекту состоит из следующих разделов:

* введение;
* описание метода решения задачи;
* разработка структур данных;
* разработка алгоритмов;
* тестирование программы;
* разработка документации;
* заключение;
* библиографический список;
* приложение (листинг программных модулей).

# Описание метода решения задачи

## 1.1 Корни уравнения

Функция называется алгебраической, если для получения ее числового значения по данному значению аргумента х требуется выполнить арифметические операции и возведение в степень с рациональным показателем.

Если в запись уравнения входят только алгебраические функции, то уравнение называется алгебраическим.

Алгебраическое уравнение всегда может быть приведено к виду:

где .

Все неалгебраические функции: показательная *ах*, логарифмическая *logax*, тригонометрические sin *x*, cos *x*, tg *x*, ctg *x* и обратные тригонометрические arcsin *x*, arcos *x*, arctg *x*, arcctg *x* называются трансцендентными.

Если в запись уравнения входят трансцендентные функции, то уравнение называется трансцендентным, например tg *x* = *ax*.

Решение уравнения с одним неизвестным *х* заключается в отыскании корней, то есть тех значений *х*, которые обращают уравнение в тождество.

В общем случае для уравнения отсутствуют аналитические формулы, определяющие его корни.

Задача отыскания корней сводится к нахождению всех точек *xi* пересечения графика функции *f(x)* с осью *x* (см. рисунок 1). Из рисунка видно, что число точек пересечения графика функции с осью *x* может быть несколько. Поэтому в качестве первого шага при решении любого уравнения проводят отделение его корней.

Это означает, что ось *x* разбивают на такие отрезки, что в каждом из них содержится только один корень уравнения. После этого следует уточнить положение каждого корня в пределах допустимой погрешности.

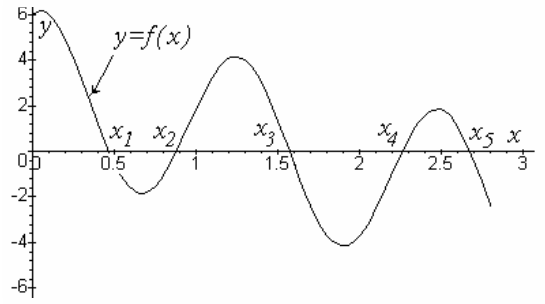


Рисунок 1 – Геометрическая интерпретация корней уравнения f(x)=0

## 1.2 Отделение корней

Для отделения корней полезна следующая теорема: если непрерывная функция f(x) принимает значения разных знаков на концах отрезка [a, b], т.е. , то внутри этого отрезка содержится, по меньшей мере, один корень уравнения . На основе этой же теоремы реализуется самый простой и надежный метод численного определения корней уравнений: метод половинного деления.

Процесс отделения корней начинается с установления знаков *f(x)* в граничных точках интересующего нас отрезка определения переменной *x*: *x = a* и *x = b*.

Затем определяются знаки *f(x)* в ряде промежуточных точек x = α1, α2..., выбор которых должен учитывать особенности функции *f(x)*.

Если окажется, что *f(αi)f(αi+1) < 0*, то в интервале (αi, αi+1) есть корень уравнения . Необходимо убедиться, является ли этот корень единственным на данном интервале. Для отделения корней практически достаточно провести процесс половинного деления, последовательно деля исходный отрезок [a, b] на 2, 4, 8 и т. д. равных частей и определяя знаки *f(x)* в точках деления.

## 1.3 Численное решение уравнения методом половинного деления

Предположим, что процесс отделения корней проведен и на отрезке [a, b] находится ровно один корень ξ уравнения . Необходимо определить его положение с погрешностью ε.

Метод половинного деления заключается в следующем (рисунок 2).

Сначала определяем середину с отрезка [a, b] c = (a+b)/2 и вычисляем значение функции f(c). Далее делаем выбор, какую из двух частей взять для уточнения корня. Очевидно, что корень будет находиться в той половине исходного отрезка, на концах которой функция имеет разные знаки.

На рисунке 2 таким будет правый отрезок – отрезок [a, c]. Для очередного шага уточнения положения корня отрезок [c, b] из рассмотрения исключаем, а с отрезком [a, c] продолжаем процесс деления, как и с первоначальным отрезком [a, b], формально переприсваивая новому значению *b* значение *c*. Если же реализуется ситуация, когда функция имеет разные знаки на концах отрезка [c, b], то из рассмотрения следует исключить отрезок [a, c], формально переприсваивая новому значению *а* значение *c*.

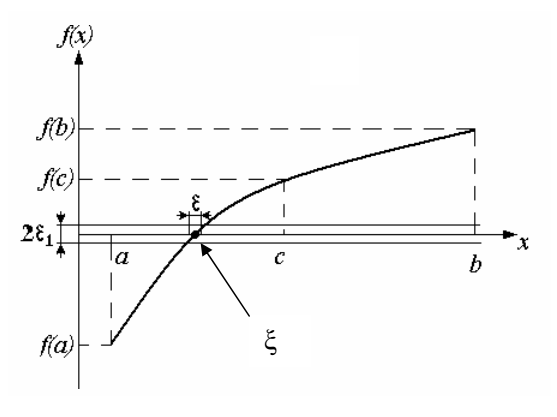


Рисунок 2 – Графическая интерпретация метода половинного деления

В результате мы получим последовательность вложенных друг в друга отрезков все уменьшающейся длины: [a1, b1], [a2, b2], ... [an, bn]. Этот повторяющийся (итерационный) процесс будем продолжать до тех пор, пока длина отрезка [an, bn] не станет меньше заданной погрешности ε вычислений.

Тогда искомый корень

ξ ≈ an ≈ bn ≈ (an + bn)/2.

Следует учитывать, что функция *f(x)* вычисляется с некоторой абсолютной погрешностью ε1. Вблизи корня значения функции *f(x)* малы по абсолютной величине и могут оказаться сравнимыми с погрешностью ее вычисления. Другими словами, при подходе к корню мы можем попасть в “полосу шумов” 2ε1 (рисунок 2) и дальнейшее уточнение корня становится бессмысленным. Поэтому надо задать ширину “полосы шумов” и прекратить итерационный процесс при попадании в нее. Также необходимо иметь в виду, что при уменьшении длины интервала [an, bn] увеличивается погрешность вычисления его длины an−bn за счет вычитания двух близких чисел.

Метод половинного деления обладает довольно большой скоростью сходимости. Так как за каждую итерацию интервал, где расположен корень, уменьшается в два раза, то через *n* итераций длина интервала будет равна (b−a)/2n. За 10 итераций интервал уменьшится в 210 ≈1024 ≈103 раз, а за 20 итераций – в 220≈106 раз.

# Разработка структур данных

Для решения задачи необходимо определить типы данных, которые будут использоваться для хранения исходных и промежуточных данных, а также результатов вычислений.

Для реализации подсчета числа итераций необходимы данные целого типа. Язык *Free Pascal* предоставляет возможность использования следующих целочисленных типов (Таблица 1):

Таблица 1 – Целые типы данных

|  |  |  |
| --- | --- | --- |
| **Тип** | **Диапазон значений** | **Размер, байт** |
| Byte | 0 .. 255 | 1 |
| Shortint | -128 .. 127 | 1 |
| Smallint | -32768 .. 32767 | 2 |
| Word | 0 .. 65535 | 2 |
| Integer | smallint or longint | 2 или 4 |
| Cardinal | longword | 4 |
| Longint | -2147483648 .. 2147483647 | 4 |
| Longword | 0..4294967295 | 4 |
| Int64 | -9223372036854775808 .. 9223372036854775807 | 8 |
| QWord | 0 .. 18446744073709551615 | 8 |

Диапазона значений типов *Byte*, *ShortInt* может не хватить для подсчета количества итераций. Диапазон значений типов *Cardinal*, *Longint*, *Longword*, *Int64*, *QWord* явно избыточен для решения поставленной задачи. Количество итераций не может быть отрицательным, поэтому тип *Word* предпочтительнее, чем *Smallint*.

Для реализации математических вычислений необходимы данные вещественного типа. Язык *Free Pascal* предоставляет возможность использования вещественных типов данных:

Таблица 2 – Вещественные типы данных

|  |  |  |  |
| --- | --- | --- | --- |
| **Тип** | **Диапазон значений** | **Значимых разрядов** | **Размер, байт** |
| Real | Зависит от платформы | Зависит от платформы | 4 or 8 |
| Single | 1.5E-45 .. 3.4E38 | 7-8 | 4 |
| Double | 5.0E-324 .. 1.7E308 | 15-16 | 8 |
| Extended | 1.9E-4932 .. 1.1E4932 | 19-20 | 10 |
| Comp | -2E64+1 .. 2E63-1 | 19-20 | 8 |
| Currency | -922337203685477.5808 | 922337203685477.5807 | 8 |

Тип данных *Real* зависит от платформы, и использовать его нежелательно. Тип *Currency* предназначен для операций с числами с фиксированной точностью (операции с денежными величинами). Тип *Comp* трактуется как вещественное число без дробной части.

Точность данных типа *Single* недостаточна для решения задач вычислительной математики, поэтому предпочтение следует отдать типам *Double* или *Extended*.

# Разработка алгоритмов

# Тестирование программы

# Разработка документации

Заключение

Библиографический список

1. Самарский А.А., Гулин А.В. Численные методы. - М.: 1989.

2. Калиткин Н.Н. Численные методы. - М.: Наука, 1978.

3. Демидович Б.П., Марон И.А. Основы вычислительной математики. - М.: 1970.

4. Копченова Н.В., Марон И.А. Вычислительная математика в примерах и задачах . - М.: Наука, 1972.

5. Мудров А.Е. Численные методы для ПЭВМ на языках Бейсик, Фортран, Паскаль. – Томск: МП «Раско», 1991.

Приложение А. Листинг программы