**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

(СПбГУТ)

ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ СЕТЕЙ И СИСТЕМ **(ИКСС)**

КАФЕДРА ПРОГРАММНОЙ ИНЖЕНЕРИИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ **(ПИ И ВТ)**

Отчет

По практической (лабораторной) работе

Дисциплина: «Алгоритмические основы программной инженерии»

Вариант №17

Выполнил: Студент группы ИКПИ-95

Новиков С.C.

Подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Принял: Преподаватель:

Ерофеев С.А.

Подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_

2020г

Оглавление

Цель работы 3

Таблица идентификаторов 3

Диапазон переменных 3

Ресурсоемкость 3

[Таблица отлова ошибок](#_Toc34827270) 4

[Блок схема](#_Toc34827271) 4

[Проверка на работоспособность](#_Toc34827272) 5

[Код на языке Golang](#_Toc34827273) 6

Вывод 7

# Цель работы

Разработать и написать программу, которая решает методом секущих кубические уравнения вида ax^3+bx^2+cx+d, коэффициенты которого вводит пользователь вручную.

# Таблица идентификаторов

|  |  |  |
| --- | --- | --- |
| N | Идентификатор | Назначение |
| 1 | var a, b, c, d float64 | Коэффициенты кубического уравнения |
| 2 | var x1, x2 float64 | Начальные приближения |
| 3 | const E = 0.01 | Погрешность |
| 4 | var xN float64 | Результат |
| 5 | var n = 0 | Счетчик итераций |
| 6 | var y float64 | Вспомогательная переменная |

# Диапазон переменных

**float64**: представляет число с плавающей точкой от

-1.797693134862315708145274237317043567981e+308 до 1.797693134862315708145274237317043567981e+308 и занимает 8 байт.

**int64**: представляет целое число от –9 223 372 036 854 775 808 до 9 223 372 036 854 775 807 и занимает 8 байт.

# Ресурсоемкость

**float64** занимает в памяти 8 байт. Всего переменных 9. В итоге 72 байт.

**int64** занимает в памяти 8 байт. Всего переменных 1. В итоге 8 байт.

Всего: 80 байт.

# Таблица отлова ошибок

|  |  |  |
| --- | --- | --- |
| Ошибка | Информационное сообщение | Реализация |
| Переполнение и ввод недопустимых аргументов, например, букв. | Неправильный формат данных! | \_, err := fmt.Scan(...)  if err != nil {  fmt.Println("Неправильный формат данных!")  os.Exit(0)  } |
| Пределы переменных | Не получилось посчитать, возможно мы вышли за допустимые пределы | if math.IsNaN(xN) {  fmt.Println("Не получилось посчитать, возможно мы вышли за допустимые пределы")  } |

# Блок схема

Начало

Инициализация переменных

Ввод и проверка на корректность (тип данных)

Ошибка

Ошибка

Считывание значений x1, x2 и проверка на корректность (тип данных)

Ошибка

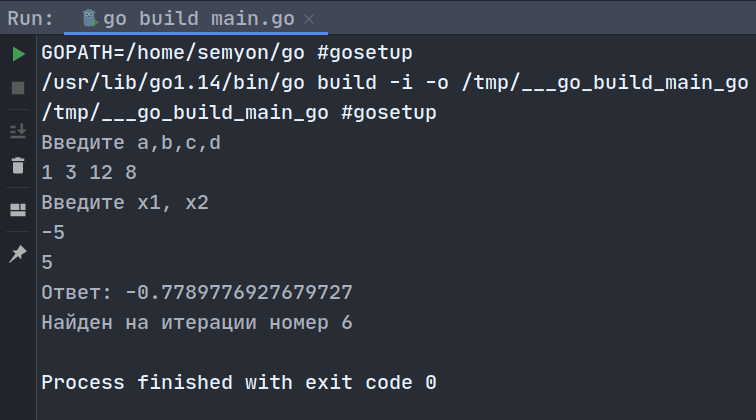
Решение уравнения

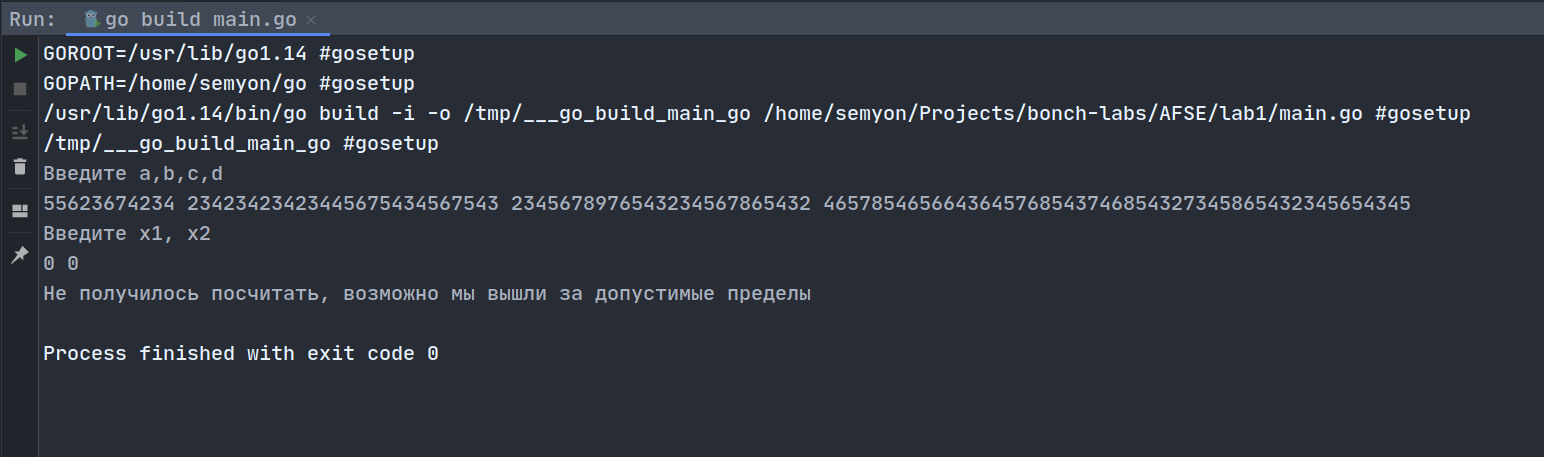
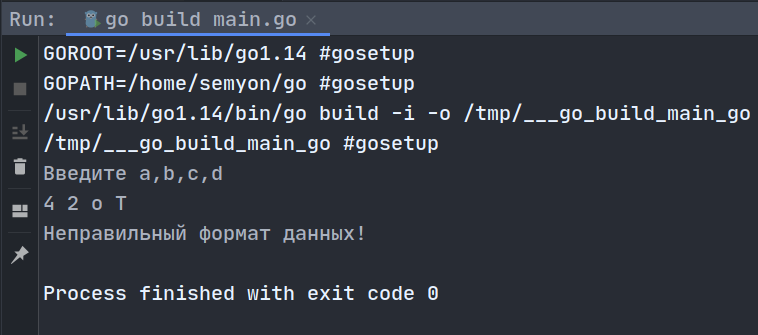
Проверка результата

Вывод результата

Конец

# Проверка на работоспособность





# Код на языке Golang

Файл main.go

package main

import (

"fmt"

"math"

"os"

)

var a, b, c, d, x1, x2 float64

const E = 0.01 // погрешность

func f(x float64) float64 {

return a\*math.Pow(x, 3) + b\*math.Pow(x, 2) + c\*x + d

}

func main() {

fmt.Println("Введите a,b,c,d")

\_, err := fmt.Scan(&a, &b, &c, &d)

if err != nil {

fmt.Println("Неправильный формат данных!")

os.Exit(0)

}

if a == 0 && b == 0 && c == 0 {

if d == 0 {

fmt.Println("x - любое")

os.Exit(0)

} else if d != 0 {

fmt.Println("нет решений")

os.Exit(0)

}

}

fmt.Println("Введите x1, x2") // начальное приближение x1, начальное приближение x2

\_, err = fmt.Scan(&x1, &x2)

if err != nil {

fmt.Println("Неправильный формат данных!")

os.Exit(0)

}

var y, xN float64

var n = 0

for {

n++

y = xN

xN = x2 - ((x2-x1)/(f(x2)-f(x1)))\*f(x2)

x1 = x2

x2 = xN

if !(math.Abs(y-xN) >= E) {

break

}

}

if math.IsNaN(xN) {

fmt.Println("Не получилось посчитать, возможно мы вышли за допустимые пределы")

} else {

fmt.Println("Ответ:", xN)

fmt.Println("Найден на итерации номер", n)

}

}

# Вывод

Поставлена задача вывода в консоль результата решения кубического уравнения методом секущих. Данная задача реализована на языке Go. В этом алгоритме реализовано решение, постановленной задачи с контролем входных данных. В контроле входных данных реализована проверка данных на некорректный ввод и переполнение переменной.