1. **Лабораторная работа №1**

**«Асимптотический анализ количественно-зависимых алгоритмов»**

* 1. **Цель работы**

Изучить поведение функций трудоемкости количественно-зависимых алгоритмов в реальных интервалах значений мощности множества исходных данных. На основании этого сделать предпочтительный выбор того или иного алгоритма. Для сравнения функций трудоемкости использовать аппарат интервального анализа, реализованный в виде программы на языке С++.

* 1. **Вариант задания ­– 14**

Требуется для указанных по варианту пары функций трудоёмкости и , целочисленных интервалов {(20; 50), (100; 120), (500; 540)} и значений , определить, каково соотношение между функциями трудоемкости на заданном интервале. Для вычисления значений функций и определения соотношений написать программу на языке С. Результаты работы программы сохранять в текстовые файлы.

Путем подбора значений аргумента определить интервалы, на которых выполняется соотношение: .

Построить графики заданных функций на указанном интервале (вручную или с помощью любого программного обеспечения).

* 1. **Ход работы**
     1. **Текст программы**

В соответствии с вариантом была написана программа на языке си и её текст представлен ниже.

#include <math.h>

#include <stdio.h>

int main() {

double Fn, //F(n)

Gn, //G(n)

ATg\_FG,

ATg\_GF,

pi, // ATg\_FG - ATg\_GF

Nbegin, //Левая граница интервала ->вводим с клавы

Nend, //Правая граница интервала ->вводим с клавы

step, //Шаг изменения аргумента ->вводим с клавы

phi, //Значение угла

k, //Коэффициент кратности ->вводим с клавы

Delta, //Оценка «Дельта»

Theta, //Оценка «Тетта»

O\_large, //Оценка «О-большое»

ii; //Значение аргумента функций трудоемкости (n)

FILE \*stream; //Указатель на файл, в который осуществляется ввод-вывод расчетов

stream = fopen("Example\_TA.TXT", "w+"); // открытие файла для записи

// Ввод значений границ интервалов, шага изменения аргумента внутри интервала, коэффициента кратности

printf("Input Nbegin ->"); scanf("%lf", &Nbegin); //Левая граница, ввод значения

printf("Input Nend ->"); scanf("%lf", &Nend); //Правая граница, ввод значения

printf("Input step ->"); scanf("%lf", &step); //Шаг изменения аргумента

printf("Input koefficient ->"); scanf("%lf", &k); //Коэффициент кратности

phi = M\_PI/k; // Определили угол изменения как П/k, (M\_PI - встроенная константа языка Си = числу Pi)

ii = Nbegin; //Аргумент функций равен левой границе интервала

fprintf(stream, "n, F(n), G(n), ATg\_FG, ATg\_GF, pi, Delta, Theta, O\_large \n\n");

while (ii <= Nend) {

Fn = (pow(2, (ii-4))) / sqrt(ii); //Расчет значения функции F(n)

Gn = pow(1.618, ii); //Расчет значения функции G(n)

ATg\_FG = atan(Fn/Gn);

ATg\_GF = atan(Gn/Fn);

pi = ATg\_FG - ATg\_GF;

Delta = phi - pi; //Оценка «Дельта»

Theta= fabs(pi) - phi; //Оценка «Тетта»

O\_large = pi + phi; //Оценка «О-большое»

//Запись расчетов в файл

fprintf(stream, "%-20.10lf %-20.10lf \t %-20.10lf\t %-20.10lf %-20.10lf %-20.10lf %-20.10lf %-20.10lf %-20.10lf\n", ii, Fn, Gn, ATg\_FG, ATg\_GF, pi, Delta, Theta, O\_large);

ii = ii + step; //Получение следующего значения аргумента

}

fclose(stream); //Закрыли файл

}

* + 1. **Тестирование**

Сперва была написана программа без использования семафоров. Ожидаемо результат выполнения

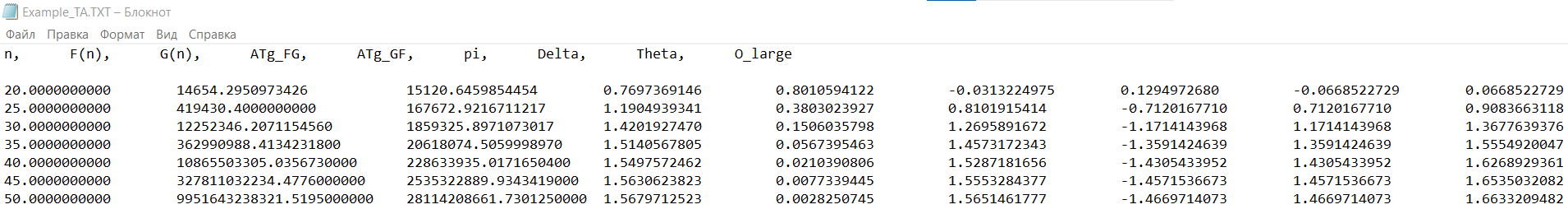


Рисунок 1.1 – Результат выполнения программы при интервале от 20 до 50, значении и шаге изменения аргумента = 5

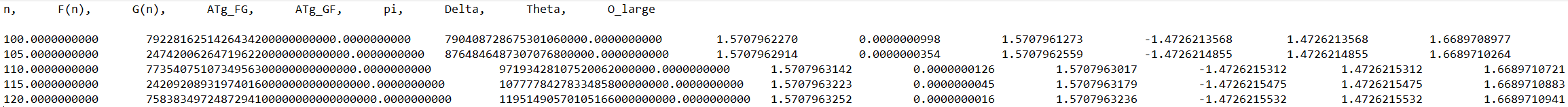


Рисунок 1.2 – Результат выполнения программы при интервале от 100 до 120, значении и шаге изменения аргумента = 5

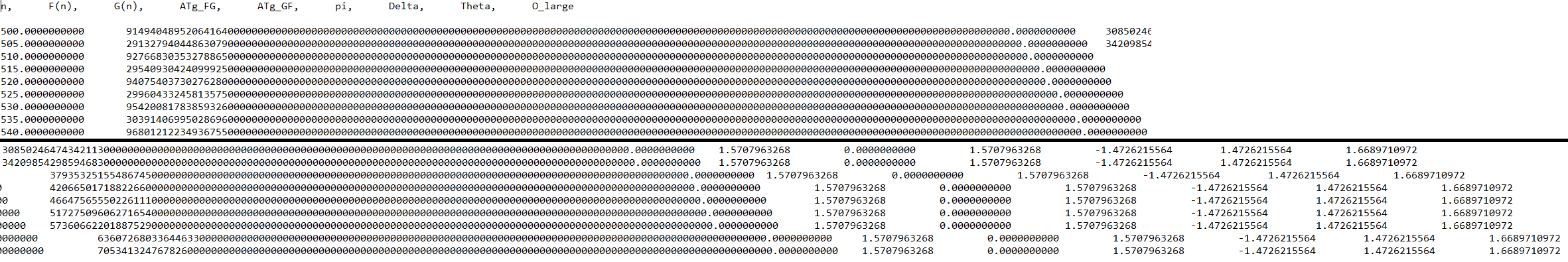


Рисунок 1.3 – Результат выполнения программы при интервале от 500 до 540, значении и шаге изменения аргумента = 5

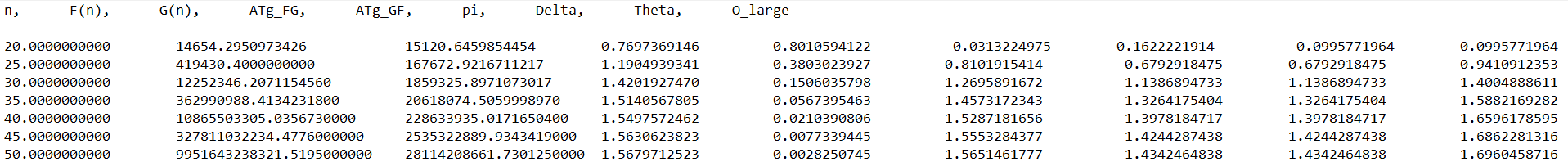


Рисунок 1.4 – Результат выполнения программы при интервале от 20 до 50, значении и шаге изменения аргумента = 5

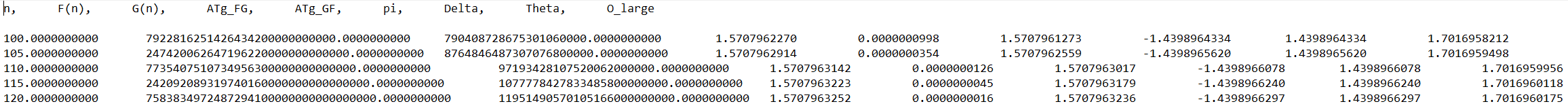


Рисунок 1.5 – Результат выполнения программы при интервале от 100 до 120, значении и шаге изменения аргумента = 5

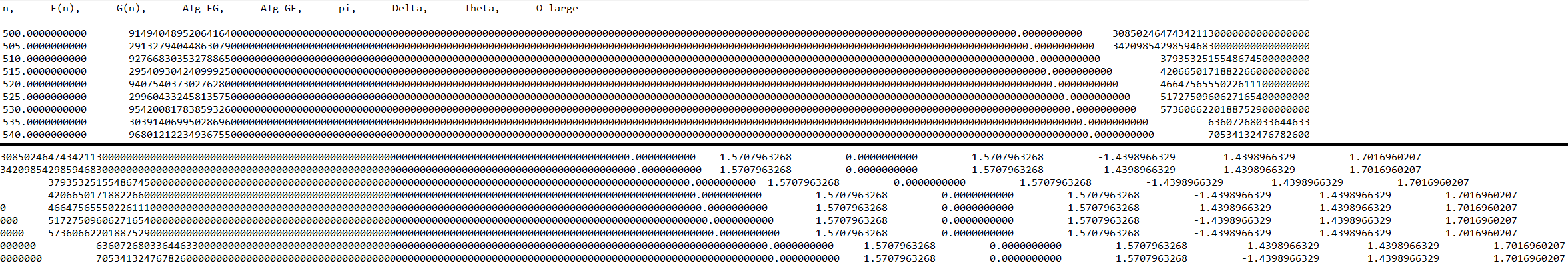


Рисунок 1.6 – Результат выполнения программы при интервале от 500 до 540, значении и шаге изменения аргумента = 5

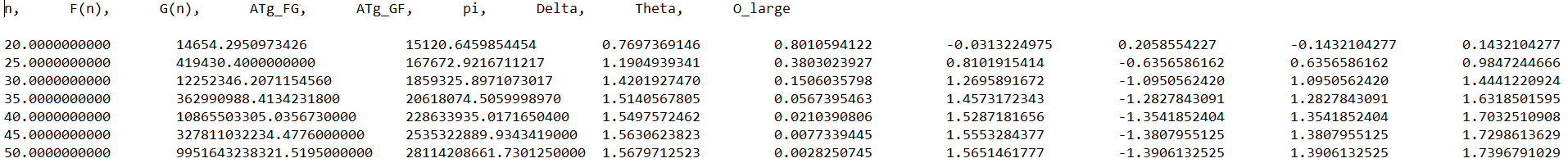


Рисунок 1.7 – Результат выполнения программы при интервале от 20 до 50, значении и шаге изменения аргумента = 5

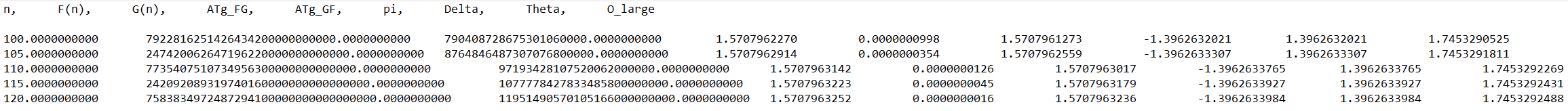


Рисунок 1.8 – Результат выполнения программы при интервале от 100 до 120, значении и шаге изменения аргумента = 5

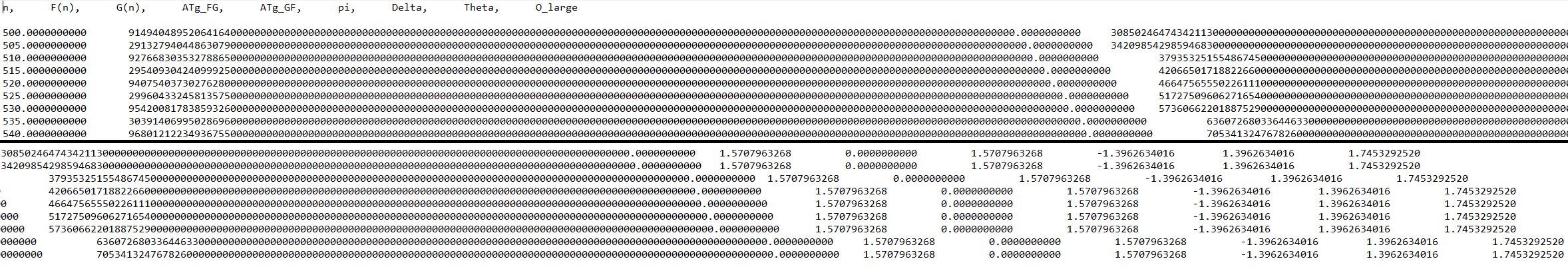


Рисунок 1.9 – Результат выполнения программы при интервале от 500 до 540, значении и шаге изменения аргумента = 5

**Выводы**

При выполнении данной лабораторной работы были получены навыки разработки программ с семафорами с помощью средств WinAPI. Определено, что в WinAPI аналогом операции P(S) являются WaitForSingleObject и WaitForMultipleObjects, а операции V(S) – ReleaseSemaphore. Получены практические навыки работы с семафорами. Также были повторно закреплены навыки использования многопоточности. Полученные знания позволят создавать многопоточные приложения, которые на многоядерных системах будут работать эффективнее.