**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР**

отчет

**по лабораторной работе №3**

**по дисциплине «Программирование»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 1302 |  | Новиков Г.В. |
| Преподаватель |  | Калмычков В.А. |

Санкт-Петербург

2021

Оглавление

[1. Исходная формулировка 3](#_Toc1333552816)

[2. Особенности задания 3](#_Toc2095740380)

[3. Контрольный пример 3](#_Toc1475549438)

[4. Формальная постановка задания 3](#_Toc1261236090)

[5. Формат хранения данных 3](#_Toc657826389)

[6. Ограничения, обусловленные выполнением на компьютере 4](#_Toc2019963878)

[7. Макет ввода и вывода 4](#_Toc1562074842)

[9. Алгоритм решения 4](#_Toc249188223)

[10. Программа 6](#_Toc1025159590)

[12. Выводы о проделанной работе 10](#_Toc1740156452)

# **Исходная формулировка**

Рассматриваются ряды f(x) = Σi = 0...∞ ui. Для каждого индивидуального задания определены вид элемента ряда ui, функция f (x), область сходимости D, если D ≠ R.

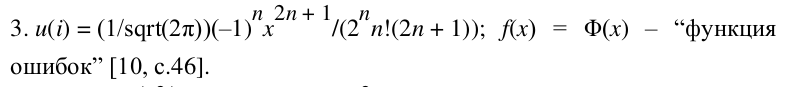
u(n) = (1/sqrt(2π))(–1)nx2n+1 / (2nn!(2n + 1))

f(x) = Φ(x) – “функция ошибок”

Найти первый n, при котором |u(n)| < eps

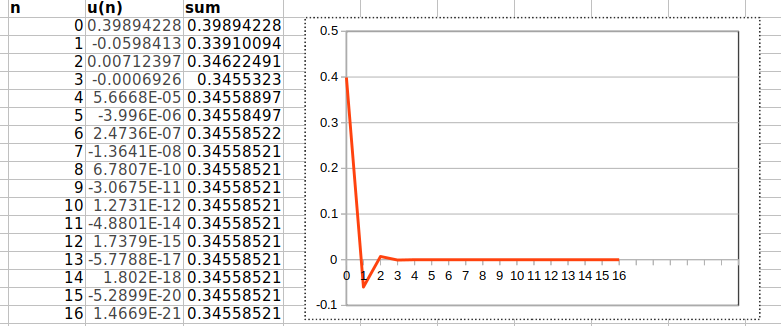
# **Особенности задания**

Двойной вывод: в текстовый файл и в консоль.

В моем задании (3) в формуле используется буква n вместо i, поэтому все i я заменил на n:

# **Контрольный пример**

Для x = 1; eps = 1e-20:



Из чего следует, что n=16.

# **Формальная постановка задания**

Дано eps и x.

0 < eps <= 1e-20

Найти первый n, при котором |u(n)| < eps

# **Формат хранения данных**

|  |  |  |
| --- | --- | --- |
| **Переменная** | **Тип** | **Назначение** |
| x | double | Переменная x |
| eps | double | Значение эпсилон |
| eps\_is\_valid | bool | Правильность введенного эпсилон |
| file | ofstream | Вывод данных в файл |
| PI | double | Число ПИ |
| success | bool | Найдено ли искомое значение |
| Final\_n | double | Искомое значение n |
| n | int | Переменная n |
| u | double | Значение u(n) на текущем шаге |
| sum | double | Сумма u(n) |

# **Ограничения, обусловленные выполнением на компьютере**

Тип double предоставляет доступ к вещественным числам в диапазоне от +/- 1.7E-308 до 1.7E+308.

# **Макет ввода и вывода**

|  |  |
| --- | --- |
| **Приветствие** | Author: Novikov G.  Group: 1302  Start date: 25.10.2021  End date: 25.10.2021  Version: 3.1.01  Formulation: ((1/(sqrt(2\*PI))\*(-1)^n\*x^(2n+1))/(2^n\*n!\*(2\*n + 1)) |
| **Запрос x** | Enter x: |
| **Ввод x** | x |
| **Запрос eps** | Enter eps(between 0 and 1e-20): |
| **Ввод eps** | Eps |
| **Неверное значение eps** | Invalid value(tries left: 2) |
| **Вывод таблицы в консоль** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  n | u(n) | sum |
| **Вывод таблицы в файл** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  n | u(n) | sum |
| **Результат(если найдено)** | First EPS-locality occurenceё is attained at n = 16 |
| **Результат(если не найдено)** | Didn`t find EPS-locality occurencies |

1. **Средства обеспечения ввода и вывода**

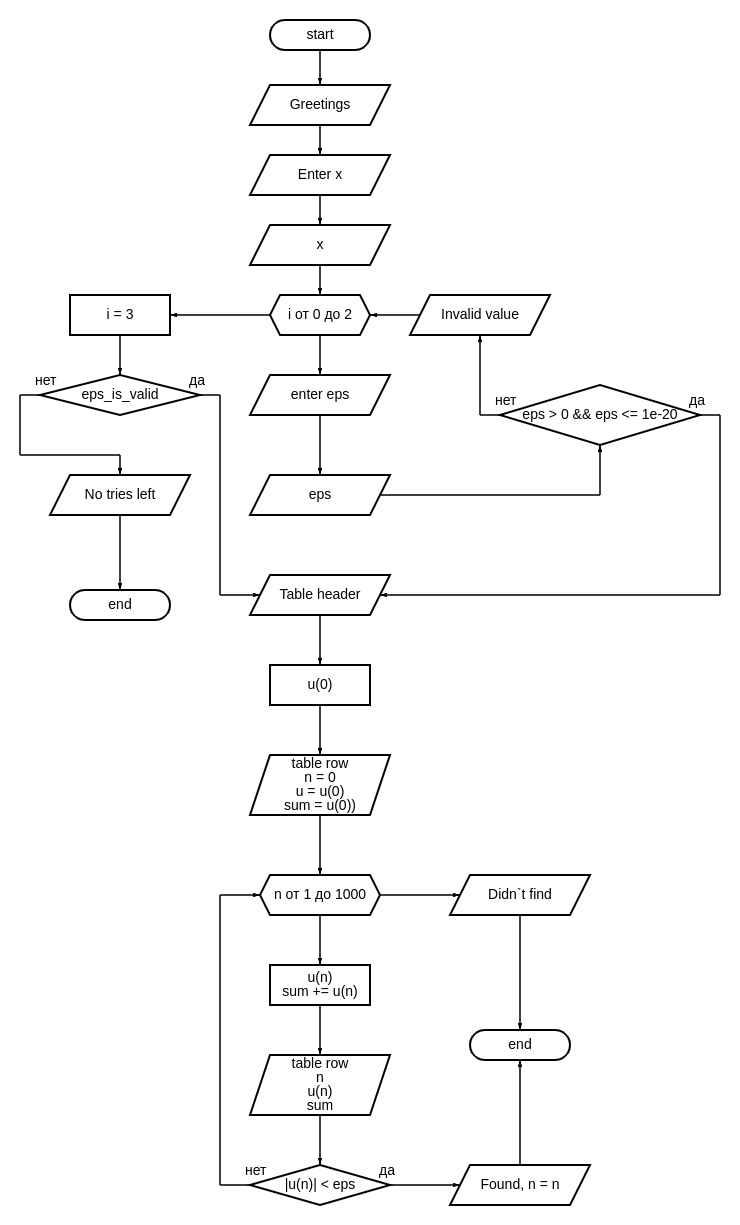
Библиотеки: iostream, fstream, iomanip

Средство обеспечения ввода: cin>>

Средство обеспечения вывода в консоль: cout <<

Средство обеспечения вывода в файл: file <<

# **Алгоритм решения**



# **Программа**

// Автор: Новиков Г.В.

// Группа: 1302

// Дата начала: 25.10.2021

// Дата окончания: 25.10.2021

// Версия: 3.1.01

#include <iostream>

#include <iomanip>

#include <fstream>

using namespace std;

void print\_line(double n, double u, double sum) {

ofstream file;

file.open("out.txt", ios::app);

cout << setw(9) << n;

cout << "|";

cout << setw(23) << u;

cout << "|";

cout << setw(23) << sum << endl;

file << setw(9) << n;

file << "|";

file << setw(23) << u;

file << "|";

file << setw(23) << sum << endl;

file.close();

}

double absolute(double num) {

if (num < 0) {

num = -num;

}

return num;

}

double sqrt(double num) {

double res = num / 2;

for (int i = 0; i < 1000; i++) {

res = (res + num / res) / 2;

};

return res;

}

double next\_u(double u, int n, double x) {

return u \* (-(x \* x \* (2 \* n + 1)) / (2 \* (n + 1) \* (2 \* n + 3)));

};

int main(int argc, char const \*argv[])

{

cout << "Author: Novikov G. \n"

"Group: 1302 \n"

"Start date: 25.10.2021 \n"

"End date: 25.10.2021 \n"

"Version: 3.1.01 \n"

"Formulation: ((1/(sqrt(2\*PI))\*(-1)^n\*x^(2n+1))/(2^n\*n!\*(2\*n + 1))" << endl << endl;

double eps, x;

bool eps\_is\_valid = false;

cout << "Enter x: ";

cin >> x;

for (int i = 0; i < 3; i++) {

cout << "Enter eps(between 0 and 1e-20): ";

cin >> eps;

if (eps > 0 && eps <= 1e-20) {

eps\_is\_valid = true;

break;

};

cout << "Invalid value(tries left: " << 2 - i << ")" << endl;

};

if (!eps\_is\_valid) {

cout << "No tries left" << endl;

return 0;

};

ofstream file;

file.open("out.txt", ios::trunc);

for (int i = 0; i < 57; i++) {

cout << "\_";

}

cout << endl;

cout << setw(5) << "n";

cout << setw(5) << "|";

cout << setw(13) << "u(n)";

cout << setw(11) << "|";

cout << setw(13) << "sum";

cout << setw(10) << endl;

for (int i = 0; i < 57; i++) {

file << "\_";

}

file << endl;

file << setw(5) << "n";

file << setw(5) << "|";

file << setw(13) << "u(n)";

file << setw(11) << "|";

file << setw(13) << "sum";

file << setw(10) << endl;

file.close();

double PI = 3.141592653589793;

bool success = false;

int final\_n;

int n = 0;

double u = 1 / sqrt(2 \* PI);

double sum = u;

print\_line(n, u, sum);

for (int n = 1; n <= 1000; n++) {

u = next\_u(u, n, x);

sum += u;

print\_line(n, u, sum);

if (absolute(u) < eps) {

success = true;

final\_n = n;

break;

}

}

file.open("out.txt", ios::app);

if (success) {

cout << "First EPS-locality occurence is attained at n = " << final\_n << endl;

file << "First EPS-locality occurence is attained at n = " << final\_n << endl;

} else {

cout << "Didn`t find EPS-locality occurencies" << endl;

file << "Didn`t find EPS-locality occurencies" << endl;

}

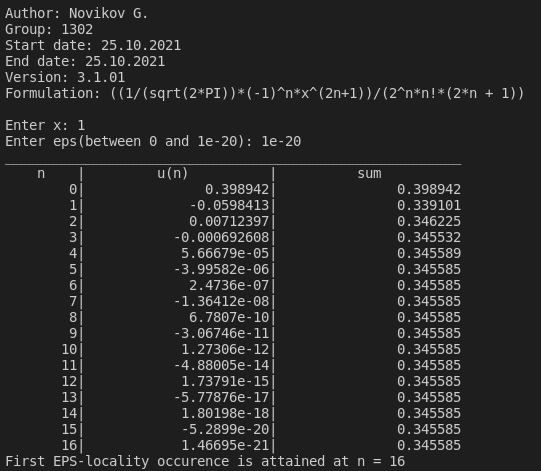
file.close();

return 0;

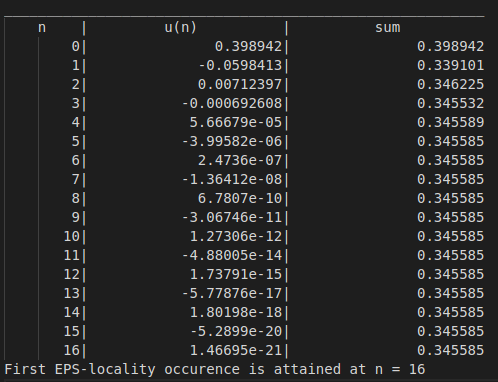
}

1. **Результаты работы программы**

Консоль:



Файл:



# **Выводы о проделанной работе**

Узнал о выводе результата программы в текстовый файл.