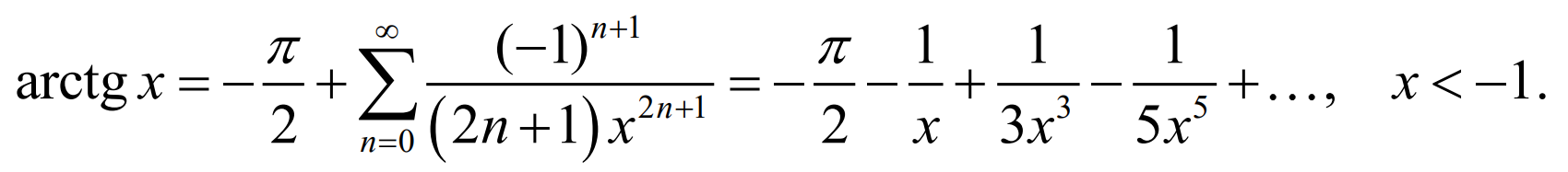
ЗВІТ  
про виконання лабораторної роботи № 5.2  
« Обчислення суми ряду Тейлора за допомогою функцій »  
з дисципліни  
«Алгоритмізація та програмування»  
студентки групи ІК-12  
Пилипів Яни Вікторівни

Умова завдання:

Обчислити і вивести на екран у вигляді таблиці значення функції, заданої за допомогою ряду Тейлора, на інтервалі від хпоч до хкін з кроком dx та точністю . Параметри xпоч, xкін, dx, eps вводяться з клавіатури. Таблиця має містити заголовок та шапку. Кожний рядок таблиці має містити значення аргументу, значення функції, значення суми ряду та кількість порахованих доданків. Суму ряду Тейлора та значення доданку обчислювати за допомогою допоміжних алгоритмів, реалізованих за допомогою окремих функцій. При обчисленні значення доданків використовувати рекурентні співвідношення. Всю необхідну функціям інформацію слід передавати лише за допомогою параметрів.



Блок-схеми алгоритму програми та кожної функції:

UML-діаграма дії програми та кожної функції:

Структурна схема програми:

Текст програми:

1.

#include <iostream>

#include <iomanip>

#include <cmath>

using namespace std;

void S(const double x, const double eps, int &n, double &s);

void A(const double x, const int n, double &a);

int main()

{

double xp, xk, x, dx, eps, s = 0;

int n = 0;

cout << "xp = "; cin >> xp;

cout << "xk = "; cin >> xk;

cout << "dx = "; cin >> dx;

cout << "eps = "; cin >> eps;

cout << fixed;

cout << "------------------------------------------" << endl;

cout << "|" << setw(7) << "x" << " |"

<< setw(10) << "exp(x)" << " |"

<< setw(10) << "S" << " |"

<< setw(5) << "n" << " |"

<< endl;

cout << "------------------------------------------" << endl;

x = xp;

while (x <= xk)

{

S(x, eps, n, s);

cout << "|" << setw(7) << setprecision(2) << x << " |"

<< setw(10) << setprecision(5) << exp(x) << " |"

<< setw(10) << setprecision(5) << s << " |"

<< setw(5) << n << " |"

<< endl;

x += dx;

}

cout << "------------------------------------------" << endl;

return 0;

}

void S(const double x, const double eps, int &n, double &s)

{

n = 0;

double a = 1;

s = -atan(1.0) / 2;

do {

n++;

A(x, n, a);

s += a;

} while (abs(a) >= eps);

}

void A(const double x, const int n, double &a)

{

double R = -(1.0 \* x \* x \* (2 \* n - 1)) / (2 \* n + 1);

a \*= R;

}

2.

#include <iostream>

#include <iomanip>

#include <cmath>

using namespace std;

double S(const double x, const double eps, int &n, double s);

double A(const double x, const int n, double a);

int main2()

{

double xp, xk, x, dx, eps, s = 0;

int n = 0;

cout << "xp = "; cin >> xp;

cout << "xk = "; cin >> xk;

cout << "dx = "; cin >> dx;

cout << "eps = "; cin >> eps;

cout << fixed;

cout << "------------------------------------------" << endl;

cout << "|" << setw(7) << "x" << " |"

<< setw(10) << "exp(x)" << " |"

<< setw(10) << "S" << " |"

<< setw(5) << "n" << " |"

<< endl;

cout << "------------------------------------------" << endl;

x = xp;

while (x <= xk)

{

s = S(x, eps, n, s);

cout << "|" << setw(7) << setprecision(2) << x << " |"

<< setw(10) << setprecision(5) << exp(x) << " |"

<< setw(10) << setprecision(5) << s << " |"

<< setw(5) << n << " |"

<< endl;

x += dx;

}

cout << "------------------------------------------" << endl;

return 0;

}

double S(const double x, const double eps, int &n, double s)

{

n = 0;

double a = 1;

s = -atan(1.0) / 2;

do {

n++;

a = A(x, n, a);

s += a;

} while (abs(a) >= eps);

return s;

}

double A(const double x, const int n, double a)

{

double R = -(1.0 \* x \* x \* (2 \* n - 1)) / (2 \* n + 1);

a \*= R;

return a;

}

Посилання на git-репозиторій з проектом:

https://github.com/pylypivyana/labs\_ap.git

Результати unit-тесту:

Висновки:

В ході даної лабораторної роботи я навчилася використовувати функції, формати виводу, рекурентні співвідношення.