

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра САПР**

**ОТЧЕТ**  
**по «Практическому заданию №2»**  
**Тема: Итерационные вычисления суммы**  
**с вещественными числами**

Студент гр. 4335

\_\_\_\_\_

Иванов Г.Д.

Руководитель

\_\_\_\_\_

Калмычков В.А.

Санкт-Петербург

2025

**Формулировка задания:**

Написание программы с использованием циклов (итерационные алгоритмы с переходным коэффициентом q)

**Дано:**

Формула для слагаемого частичной суммы:

$$u_i(x) = \frac{(-1)^i * 2^{(2i-1)} * (i-1)! * i! * x^{(2i+1)}}{(2i+1)!}$$

**Найти:**

Значение частичной суммы  $S(\varepsilon, x) = \sum_{i=1}^n u_i(x)$ , где:

$$\begin{cases} |u_{n-1}(x)| > \varepsilon \\ |u_n(x)| \leq \varepsilon \\ n \leq 1000 \end{cases}$$

**Контрольный пример:**

Проведем ручной расчет значения частичной суммы для первых 20 слагаемых. Расчетные данные приведены в виде таблицы:

Номер слагаемого	Значение слагаемого	Значение частичной суммы
1	-0,24300000	-0,24300000
2	0,07873200	-0,16426800
3	-0,03644167	-0,20070967
4	0,01967850	-0,18103117
5	-0,01159243	-0,19262359
6	0,00722297	-0,18540062
7	-0,00468049	-0,19008111
8	0,00312216	-0,18695895
9	-0,00212964	-0,18908859
10	0,00147858	-0,18761001
11	-0,00104143	-0,18865144
12	0,00074233	-0,18790911
13	-0,00053448	-0,18844359
14	0,00038814	-0,18805545
15	-0,00028397	-0,18833942
16	0,00020911	-0,18813031
17	-0,00015486	-0,18828517
18	0,00011526	-0,18816990
19	-0,00008618	-0,18825609
20	0,00006470	-0,18819139

Таблица 1. Ручной расчет значения частной суммы

Как можно видеть из таблицы 1, при увеличении номера слагаемого, его модуль уменьшается, стремясь к 0. Для наглядности, построим график:



Рисунок 1. График зависимости модуля слагаемого частичной суммы от его номера

Из этого следует, что действительно, граничное условие  $|u_n(x)| \leq \varepsilon$  имеет смысл, причем число  $\varepsilon$  должно быть выбрано положительным и близким к 0.

### Способ решения:

Для упрощения организации вычислений сделаем переход от исходного выражения к выражению в виде формулы с переходным коэффициентом:

$$u_{i+1}(x) = q * u_i(x)$$

Для получения переходного коэффициента  $q$  требуется найти частное двух подряд идущих слагаемых  $u_i(x)$  и  $u_{i+1}(x)$ :

$$q = \frac{u_{i+1}(x)}{u_i(x)}$$

Подставим в это выражение нашу исходную формулу:

$$q = \frac{(-1)^{(i+1)} * 2^{(2i+1)} * i! * (i+1)! * x^{(2i+3)}}{(2i+3)!} * \frac{(2i+1)!}{(-1)^i * 2^{(2i-1)} * (i-1)! * i! * x^{(2i+1)}} =$$

$$= \frac{-1 * 2^2 * i * (i + 1) * x^2}{(2i + 2)(2i + 3)} * \frac{(-1)^i * 2^{(2i-1)} * (i - 1)! * i! * x^{(2i+1)}}{(2i + 1)!} * \\ * \frac{(2i + 1)!}{(-1)^i * 2^{(2i-1)} * (i - 1)! * i! * x^{(2i+1)}} = -\frac{4i(i + 1)x^2}{(2i + 2)(2i + 3)} = -\frac{2ix^2}{2i + 3}$$

Таким образом, следующее слагаемое частичной суммы может быть получено умножением предыдущего слагаемого на коэффициент  $-\frac{2ix^2}{2i+3}$ .

Первое слагаемое  $u_1(x)$  рассчитывается по исходной формуле:

$$u_1(x) = \frac{-1^1 * 2^{(2-1)} * (1 - 1)! * 1! * x^{(2+1)}}{(2 + 1)!} = -\frac{2x^3}{2 * 3} = -\frac{x^3}{3}$$

### Текст программы:

Файл *main.cpp*:

```
#include <iostream>
#include <iomanip>

namespace {
const char credits [] = "=====\\n"
    "Практическая работа №2 по дисциплине \\\"Программирование\\\"\\n"
    "Автор: Иванов Григорий Денисович\\n"
    "Группа: 4335 (подгруппа 1)\\n"
    "Версия: 1\\n"
    "Период выполнения работы: 16.01.2025 - 17.01.2025\\n"
    "=====\\n";

const char description [] = "Описание:\\n"
    "В этой программе будет проведен расчёт частичной суммы\\n"
    "S_n(x) = SUM (u_1(x), u_2(x), ... u_n(x)), номер последнего слагаемого\\n"
    "определяется в соответствии со значением числа e>0.\\n"
    "Формула для слагаемого u_i(x):\\n"
    "    u_i(x)=(-1)^i * 2^(2i-1) * (i-1)! * i! * x^(2i+1) / (2i+1)!\\n"
    "Упрощенная формула (формула с переходным коэффициентом):\\n"
    "    u_{i+1}(x) = (-2x^2*i) / (2i+3) * u_i(x)\\n"
    """;

}

int main()
{
    double x, tmp, sum = 0, ep;
    unsigned int i {1};

    std::cout << credits << std::endl;
    std::cout << description << std::endl;
```

```

do
{
    std::cout << "Введите X из диапазона (-1, 1):";
    std::cin >> x;
}while(x <= -1 || x >= 1);

do
{
    std::cout << "Введите e из диапазона (0, 1):";
    std::cin >> ep;
}while(ep <= 0 || ep >= 1);

tmp = -x*x*x / 3;
std::cout << "|" << std::setw(10) << "i"    //"Шаг"
    << "|" << std::setw(20) << "u_i(x)" //"Слагаемое"
    << "|" << std::setw(20) << "s_i(x)" //"Сумма"
    << "|" << std::endl;

for(;;i++)
{
    sum += tmp;
    std::cout << "|" << std::setw(10) << i
        << "|" << std::setw(20) << std::setprecision(10) << tmp
        << "|" << std::setw(20) << std::setprecision(10) << sum
        << "|" << std::endl;

    tmp *= -2 * x * x * i;
    tmp /= 2 * i + 3;

    if(i >= 1000 || (tmp < 0? -tmp : tmp) < ep)
        break;
}

std::cout << "Результат: " << sum << std::endl;
return 0;
}

```

## Результаты работы программы:

Практическая работа №2 по дисциплине "Программирование"

Автор: Иванов Григорий Денисович

Группа: 4335 (подгруппа 1)

Версия: 1

Период выполнения работы: 16.01.2025 - 17.01.2025

=====

Описание:

В этой программе будет проведен расчёт частичной суммы

$S_n(x) = \text{SUM}(u_1(x), u_2(x), \dots, u_n(x))$ , номер последнего слагаемого определяется в соответствии со значением числа  $e > 0$ .

Формула для слагаемого  $u_i(x)$ :

$$u_i(x) = (-1)^i * 2^{(2i-1)} * (i-1)! * i! * x^{(2i+1)} / (2i+1)!$$

Упрощенная формула (формула с переходным коэффициентом):

$$u_{i+1}(x) = (-2x^{2*i}) / (2i+3) * u_i(x)$$

Введите X из диапазона (-1, 1): 0.9

Введите e из диапазона (0, 1): 0.00001

i	$u_i(x)$	$s_i(x)$
1	-0.243	-0.243
2	0.078732	-0.164268
3	-0.03644166857	-0.2007096686
4	0.01967850103	-0.1810311675
5	-0.01159242606	-0.1926235936
6	0.007222973161	-0.1854006204
7	-0.004680486608	-0.1900811071
8	0.00312215989	-0.1869589472
9	-0.002129641694	-0.1890885889
10	0.001478579804	-0.187610009
11	-0.001041434471	-0.1886514435
12	0.0007423344909	-0.187909109
13	-0.0005344808334	-0.1884435899
14	0.0003881436673	-0.1880554462
15	-0.0002839709153	-0.1883394171
16	0.0002091058558	-0.1881303113
17	-0.0001548578224	-0.1882851691
18	0.000115264444	-0.1881699046
19	-8.618233813e-05	-0.188256087
20	6.469981385e-05	-0.1881913872
21	-4.875055741e-05	-0.1882401377
22	3.68554214e-05	-0.1882032823
23	-2.794738763e-05	-0.1882312297
24	2.12514217e-05	-0.1882099783
25	-1.620108384e-05	-0.1882261793
26	1.23800735e-05	-0.1882137993

Результат: -0.1882137993

**Выводы:**

В этой практической работе мы освоили программирование и организацию разного вида циклов для реализации итерационного алгоритма нахождения частичной суммы математического выражения.