Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное   
учреждение высшего образования

Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

**Отчет по лабораторной работе**

**«Вычисление математических функций с использованием рядов»**

**Выполнил**:

студент группы 3824Б1ПМ1-1

Коляда В.С.

**Проверил**:

преподаватель каф. ВВСП,

Волокитин В.Д.

Нижний Новгород

2025

# Постановка задачи

Целью данной лабораторной работы является реализация вычислений математических функций, таких как , и с использованием разложения в ряд Тейлора, а также сравнение результатов с библиотечными функциями для оценки точности.

# Метод решения

В лабораторной работе рассматриваются три метода суммирования членов ряда: прямое, обратное и попарное. Для вычисления членов ряда используются рекуррентные соотношения, что позволяет избежать проблем с переполнением, вызванных вычислением больших факториалов, особенно при больших значениях n.

Формулы для вычисления функций:

: ,

: ,

: ,

: ,

**Прямое суммирование**

Этот метод предполагает последовательное сложение элементов ряда в порядке возрастания их индексов, начиная с первого члена. Такой способ удобен для понимания и реализации, однако при большом количестве итераций возможны ошибки округления, особенно если члены ряда существенно отличаются по модулю. Это может привести к снижению точности вычислений.

Функция принимает два аргумента:

* **x** — значение аргумента функции,
* **n** — количество членов ряда, которые необходимо суммировать.

Перед вычислениями проводится проверка входных данных: для тригонометрических функций x предварительно приводится к диапазону [0,2π] с использованием остатка от деления на 2π. Это предотвращает возможные ошибки при работе с большими значениями аргумента.

Члены ряда рассчитываются рекуррентно, что позволяет избежать необходимости явного вычисления факториала, упрощая вычисления. В цикле каждый новый член прибавляется к общей сумме, а затем передается на следующую итерацию.

**Обратное суммирование**

В этом методе сумма членов ряда вычисляется, начиная с последнего элемента и двигаясь к первому. Такой подход может снизить накопление ошибок округления, возникающих при сложении чисел, различающихся по порядку величины, что особенно важно для рядов с большим числом членов.

В отличие от прямого суммирования, здесь сначала вычисляются все члены ряда, но их суммирование выполняется в обратном порядке. Это особенно полезно, когда последние элементы ряда имеют малый порядок величины, так как уменьшает потерю значащих цифр при сложении.

**Попарное суммирование**

Данный метод заключается в группировке элементов ряда по два и сложении их попарно. Это помогает уменьшить накопление ошибок округления, так как разница в порядке величины слагаемых внутри одной пары обычно меньше, чем при последовательном сложении всех членов ряда.

В процессе вычисления элементы ряда группируются, и сначала суммируются попарно. Если количество членов ряда нечётное, то последний элемент прибавляется к сумме отдельно. Такой метод позволяет частично компенсировать ошибки округления, возникающие при работе с числами разного порядка величины.

# Руководство пользователя

Для начала работы с программой необходимо выбрать функцию, которую требуется вычислить, из предложенного списка: , и (см. рис. 1). Затем необходимо указать количество членов ряда n, которые будут использованы для вычисления. (см. рис. 2). Далее требуется ввести значение аргумента x, для которого будет вычислено значение выбранной функции. (см. рис. 3). После ввода всех данных программа отобразит результаты вычислений: значение функции, полученное с использованием реализованных методов, значение, вычисленное с помощью библиотеки math.h, и максимальную абсолютную разницу между ними для оценки точности. (см. рис. 4).

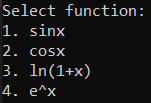


Рисунок 1. Выбор функции



Рисунок 2. Ввод числа членов ряда



Рисунок 3. Ввод значения аргумента

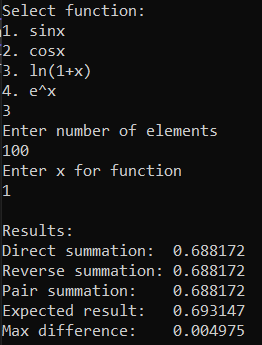


Рисунок 4. Вывод результатов

# Описание программной реализации

Данная программа на языке C реализует вычисление значений математических функций (синус, косинус, натуральный логарифм, экспонента) с использованием рядов Тейлора и сравнение результатов прямого, обратного и попарного суммирования с ожидаемыми значениями из библиотеки math.h.

**Основные функции**

1. **Вычисление членов ряда**
   * sinx, cosx, ln, expx: заполняют массив terms значениями ряда Тейлора для соответствующих функций с нормализацией аргумента x по 2π для периодических функций.
2. **Методы суммирования**
   * direct\_sum: Прямое суммирование от первого к последнему члену.
   * reverse\_sum: Обратное суммирование от последнего к первому.
   * pair\_sum: Попарное суммирование с учетом нечетного числа членов.
3. **Функция обработки**
   * filling: выбирает функцию, заполняет массив terms, вычисляет результаты суммирования и ожидаемое значение.
4. **Главная функция**
   * main: запрашивает у пользователя номер функции, количество членов ряда (n) и аргумент (x), выделяет память, вызывает filling и выводит результаты.

**Используемые библиотеки**

* <stdio.h>: Ввод-вывод (printf, scanf\_s).
* <stdlib.h>: Управление памятью (malloc, free).
* <math.h>: Математические функции и константа M\_PI.

# Подтверждение корректности

Для подтверждения корректности в программе использовались математические функции из библиотеки math.h. В результатах выводится значение, полученное math.h и разница по модулю между этим значением и полученным моими функциями.

# Результаты экспериментов

Чтобы получить результаты экспериментов, я записал погрешность для запусков с каждым n от 1 до 20, x = 5.0 для , и x = 1.0 для . С помощью библиотека matplotlib в питоне построил графики для каждого набора данных.

**Прямое суммирование**

* 1. **sin(x)** (рисунок 6)

Таблица 1. sin(x) прямое суммирование.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | 5 | 10 | 25 | 50 |
| x | **Погрешность** | **Погрешность** | **Погрешность** | **Погрешность** |
| -10 | 0.042962 | 0.000000 | 0.000000 | 0.000000 |
| -9 | 0.001423 | 0.000000 | 0.000000 | 0.000000 |
| -8 | 0.000009 | 0.000000 | 0.000000 | 0.000000 |
| -7 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| -6 | 7.307989 | 0.000398 | 0.000003 | 0.000003 |
| -5 | 1.048554 | 0.000010 | 0.000001 | 0.000001 |
| -4 | 0.095074 | 0.000000 | 0.000000 | 0.000000 |
| -3 | 0.004193 | 0.000000 | 0.000000 | 0.000000 |
| -2 | 0.000050 | 0.000000 | 0.000000 | 0.000000 |
| -1 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 0 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 1 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 2 | 0.000050 | 0.000000 | 0.000000 | 0.000000 |
| 3 | 0.004193 | 0.000000 | 0.000000 | 0.000000 |
| 4 | 0.095074 | 0.000000 | 0.000000 | 0.000000 |
| 5 | 1.048554 | 0.000010 | 0.000001 | 0.000001 |
| 6 | 7.307989 | 0.000398 | 0.000003 | 0.000003 |
| 7 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 8 | 0.000009 | 0.000000 | 0.000000 | 0.000000 |
| 9 | 0.001423 | 0.000000 | 0.000000 | 0.000000 |
| 10 | 0.042962 | 0.000000 | 0.000000 | 0.000000 |

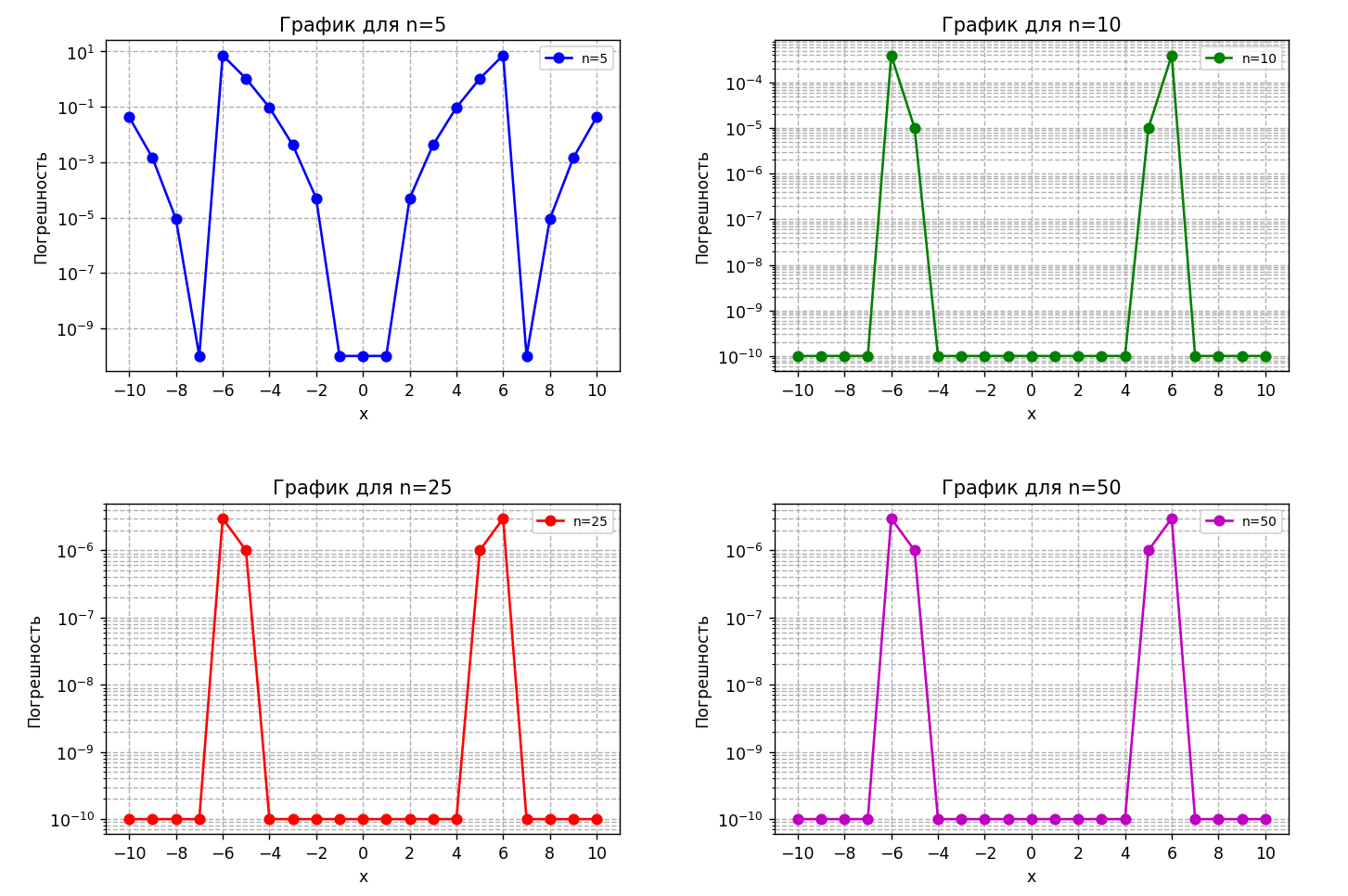
****

Рисунок 6. Графики для прямого суммирования.

* 1. **cos(x)** (рисунок 7)

Таблица 2. cos(x) прямое суммирование.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | 5 | 10 | 25 | 50 |
| x | **Погрешность** | **Погрешность** | **Погрешность** | **Погрешность** |
| -10 | 0.125187 | 0.000001 | 0.000000 | 0.000000 |
| -9 | 0.005713 | 0.000000 | 0.000000 | 0.000000 |
| -8 | 0.000060 | 0.000000 | 0.000000 | 0.000000 |
| -7 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| -6 | 12.896969 | 0.001396 | 0.000003 | 0.000003 |
| -5 | 2.244735 | 0.000037 | 0.000000 | 0.000000 |
| -4 | 0.256818 | 0.000000 | 0.000000 | 0.000000 |
| -3 | 0.015216 | 0.000000 | 0.000000 | 0.000000 |
| -2 | 0.000274 | 0.000000 | 0.000000 | 0.000000 |
| -1 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 0 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 1 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 2 | 0.000274 | 0.000000 | 0.000000 | 0.000000 |
| 3 | 0.015216 | 0.000000 | 0.000000 | 0.000000 |
| 4 | 0.256818 | 0.000000 | 0.000000 | 0.000000 |
| 5 | 2.244735 | 0.000037 | 0.000000 | 0.000000 |
| 6 | 12.896969 | 0.001396 | 0.000003 | 0.000003 |
| 7 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 8 | 0.000060 | 0.000000 | 0.000000 | 0.000000 |
| 9 | 0.005713 | 0.000000 | 0.000000 | 0.000000 |
| 10 | 0.125187 | 0.000001 | 0.000000 | 0.000000 |

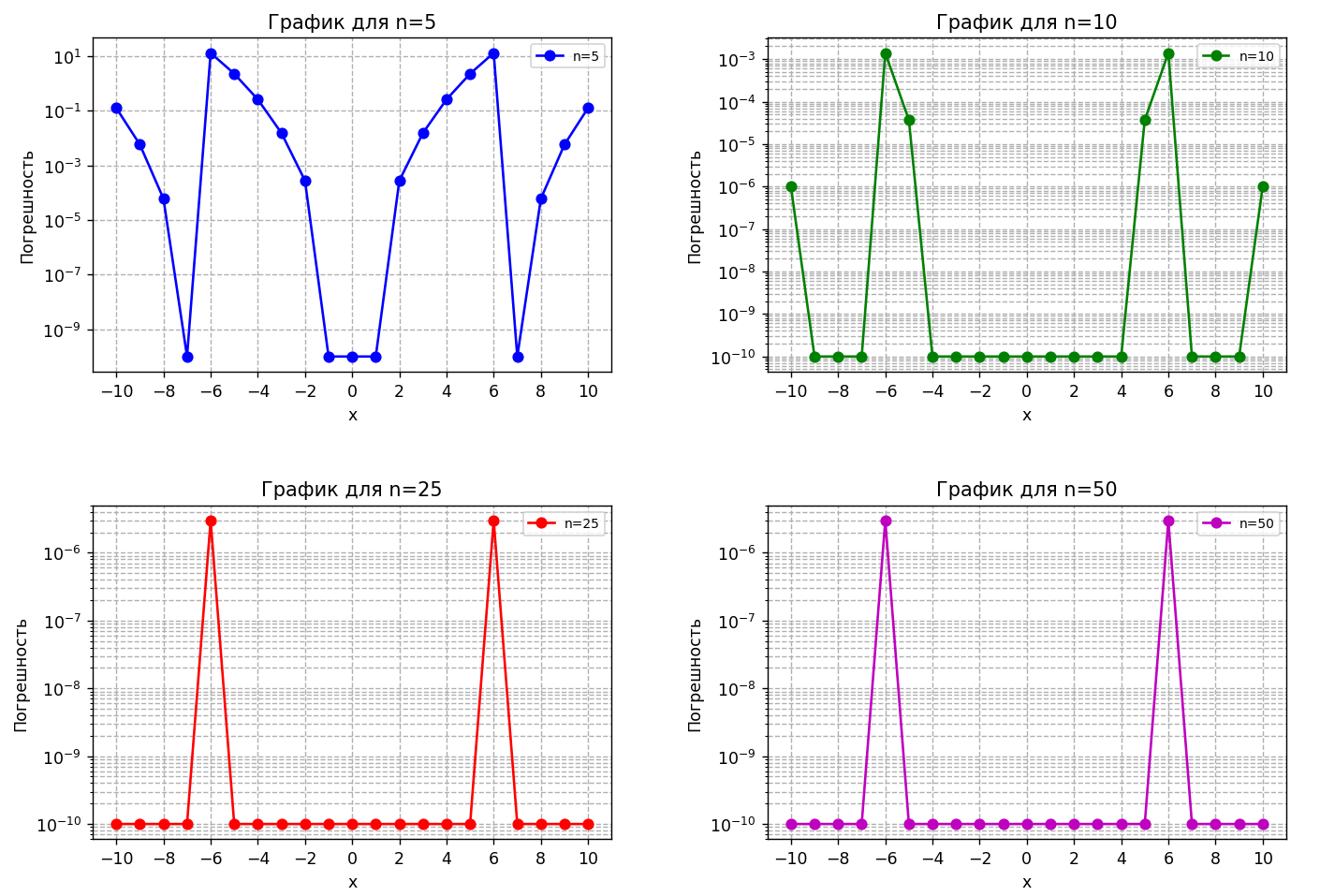
****

Рисунок 7. Графики cos(x) для прямого суммирования.

* 1. **ln(1 + x)** (рисунок 8)

Таблица 3. ln(1 + x) прямое суммирование.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | 5 | 10 | 25 | 50 |
| x | **Погрешность** | **Погрешность** | **Погрешность** | **Погрешность** |
| -0.9 | 0.472462 | 0.183838 | 0.019500 | 0.000789 |
| -0.8 | 0.150835 | 0.030563 | 0.000513 | 0.000001 |
| -0.7 | 0.051000 | 0.005113 | 0.000011 | 0.000000 |
| -0.6 | 0.016339 | 0.000740 | 0.000000 | 0.000000 |
| -0.5 | 0.004605 | 0.000082 | 0.000000 | 0.000000 |
| -0.4 | 0.001044 | 0.000006 | 0.000000 | 0.000000 |
| -0.3 | 0.000164 | 0.000000 | 0.000000 | 0.000000 |
| -0.2 | 0.000013 | 0.000000 | 0.000000 | 0.000000 |
| -0.1 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 0.0 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 0.1 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 0.2 | 0.000009 | 0.000000 | 0.000000 | 0.000000 |
| 0.3 | 0.000097 | 0.000000 | 0.000000 | 0.000000 |
| 0.4 | 0.000509 | 0.000003 | 0.000000 | 0.000000 |
| 0.5 | 0.001827 | 0.000030 | 0.000000 | 0.000000 |
| 0.6 | 0.005148 | 0.000213 | 0.000000 | 0.000000 |
| 0.7 | 0.012294 | 0.001096 | 0.000002 | 0.000000 |
| 0.8 | 0.026016 | 0.004511 | 0.000066 | 0.000000 |
| 0.9 | 0.050219 | 0.015656 | 0.001332 | 0.000048 |
| 1.0 | 0.090186 | 0.047512 | 0.019600 | 0.009900 |

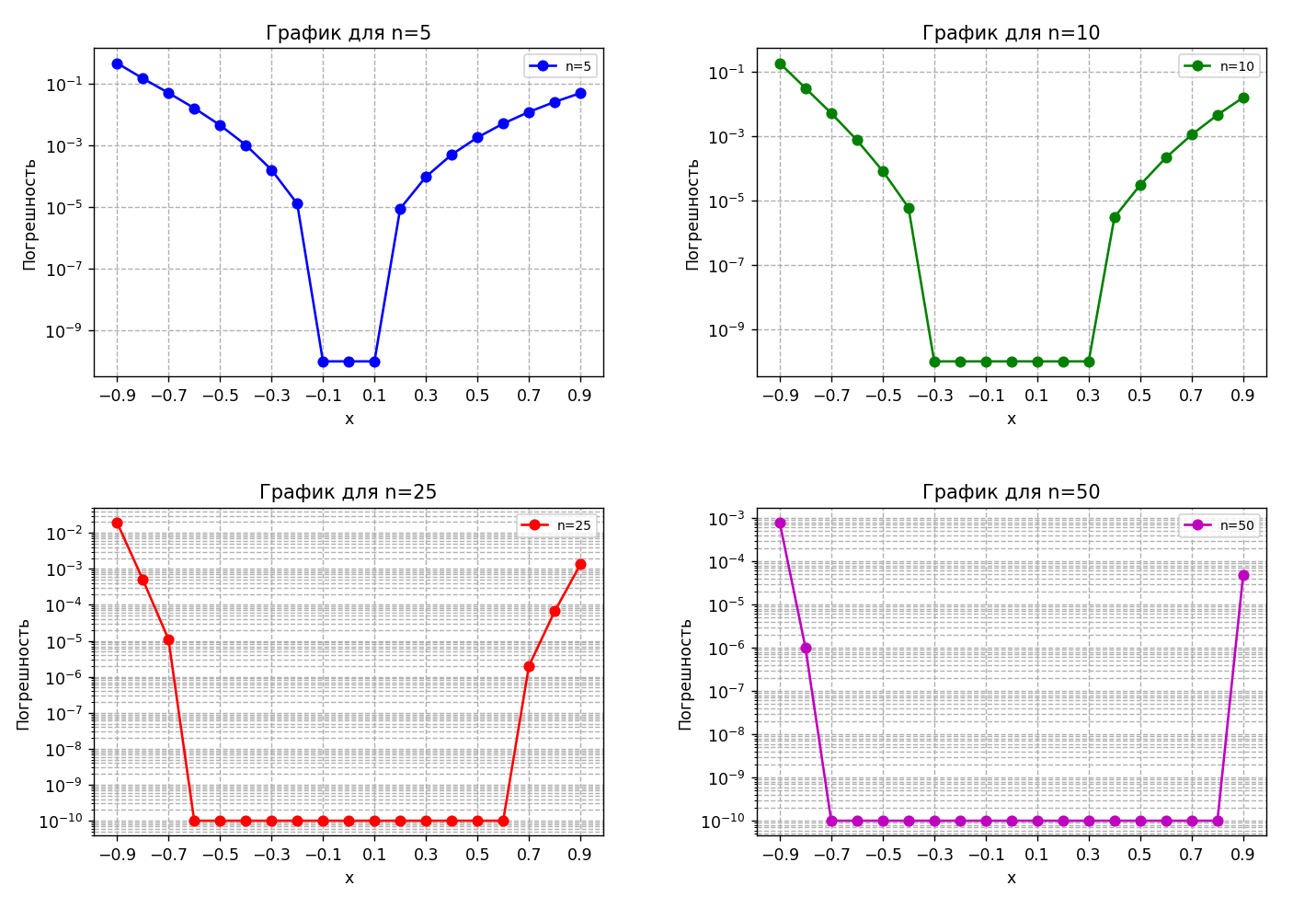
****

Рисунок 8. Графики ln(1 + x) для прямого суммирования.

* 1. (рисунок 9)

Таблица 4. прямое суммирование.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | 5 | 10 | 25 | 50 |
| x | **Погрешность** | **Погрешность** | **Погрешность** | **Погрешность** |
| -10 | 290.999955 | 1413.144821 | 0.464130 | 0.000108 |
| -9 | 184.374877 | 518.589418 | 0.034315 | 0.000021 |
| -8 | 110.333000 | 168.512848 | 0.001872 | 0.000013 |
| -7 | 61.374092 | 46.911755 | 0.000065 | 0.000003 |
| -6 | 30.997521 | 10.659623 | 0.000000 | 0.000002 |
| -5 | 13.701596 | 1.833843 | 0.000000 | 0.000000 |
| -4 | 4.981684 | 0.210556 | 0.000000 | 0.000000 |
| -3 | 1.325213 | 0.012734 | 0.000000 | 0.000000 |
| -2 | 0.197998 | 0.000238 | 0.000000 | 0.000000 |
| -1 | 0.007121 | 0.000000 | 0.000000 | 0.000000 |
| 0 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 1 | 0.009948 | 0.000000 | 0.000000 | 0.000000 |
| 2 | 0.389056 | 0.000343 | 0.000001 | 0.000001 |
| 3 | 3.710537 | 0.022142 | 0.000002 | 0.000002 |
| 4 | 20.264814 | 0.444006 | 0.000002 | 0.000002 |
| 5 | 83.038159 | 4.723675 | 0.000034 | 0.000034 |
| 6 | 288.428793 | 33.857382 | 0.000083 | 0.000083 |
| 7 | 906.924815 | 185.883830 | 0.000142 | 0.000264 |
| 8 | 2683.957987 | 844.731180 | 0.003153 | 0.000265 |
| 9 | 7657.708928 | 3343.265568 | 0.071232 | 0.000920 |
| 10 | 21382.132421 | 11939.891576 | 1.032201 | 0.002955 |

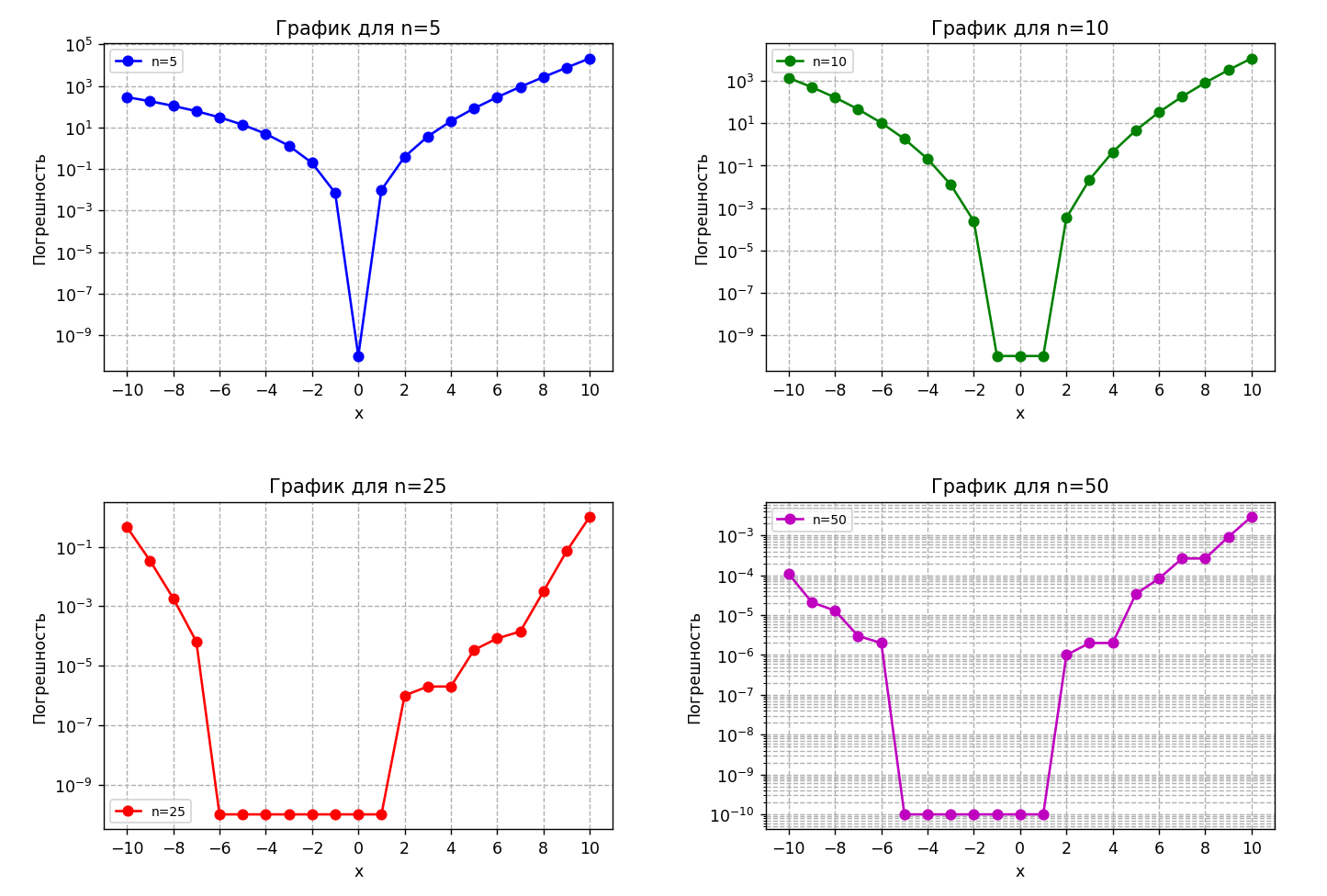
****

Рисунок 9. Графики для прямого суммирования.

**Обратное суммирование**

* 1. **sin(x)** (рисунок 10)

Таблица 5. sin(x) обратное суммирование.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | 5 | 10 | 25 | 50 |
| x | **Погрешность** | **Погрешность** | **Погрешность** | **Погрешность** |
| -10 | 0.042962 | 0.000000 | 0.000000 | 0.000000 |
| -9 | 0.001423 | 0.000000 | 0.000000 | 0.000000 |
| -8 | 0.000009 | 0.000000 | 0.000000 | 0.000000 |
| -7 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| -6 | 7.307988 | 0.000400 | 0.000004 | 0.000004 |
| -5 | 1.048554 | 0.000010 | 0.000001 | 0.000001 |
| -4 | 0.095074 | 0.000000 | 0.000000 | 0.000000 |
| -3 | 0.004193 | 0.000000 | 0.000000 | 0.000000 |
| -2 | 0.000050 | 0.000000 | 0.000000 | 0.000000 |
| -1 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 0 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 1 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 2 | 0.000050 | 0.000000 | 0.000000 | 0.000000 |
| 3 | 0.004193 | 0.000000 | 0.000000 | 0.000000 |
| 4 | 0.095074 | 0.000000 | 0.000000 | 0.000000 |
| 5 | 1.048554 | 0.000010 | 0.000001 | 0.000001 |
| 6 | 7.307988 | 0.000400 | 0.000004 | 0.000004 |
| 7 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 8 | 0.000009 | 0.000000 | 0.000000 | 0.000000 |
| 9 | 0.001423 | 0.000000 | 0.000000 | 0.000000 |
| 10 | 0.042962 | 0.000000 | 0.000000 | 0.000000 |

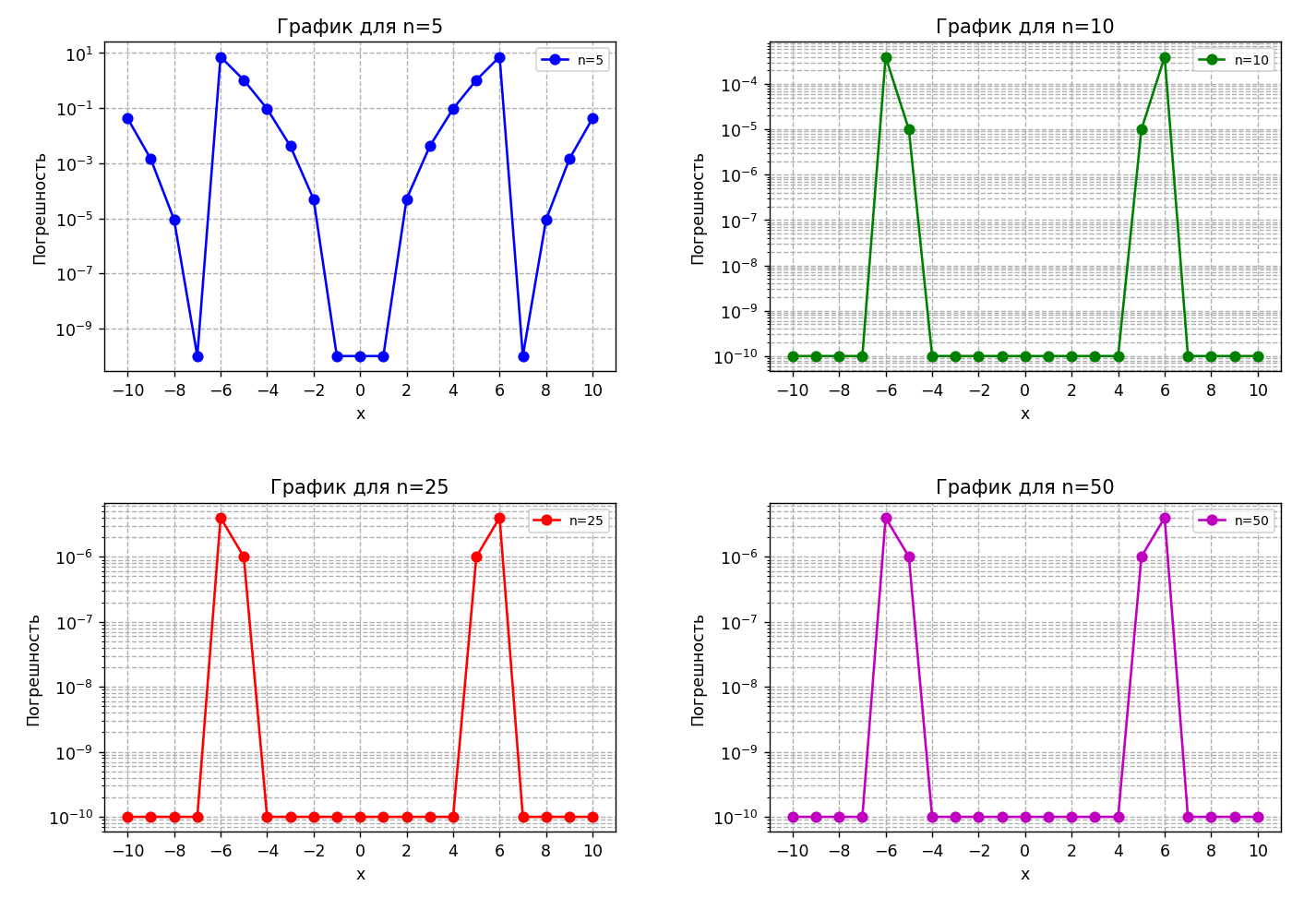
****

Рисунок 10. Графики sin(x) для обратного суммирования.

* 1. **cos(x)** (рисунок 11)

Таблица 6. cos(x) обратное суммирование.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | 5 | 10 | 25 | 50 |
| x | **Погрешность** | **Погрешность** | **Погрешность** | **Погрешность** |
| -10 | 0.125187 | 0.000001 | 0.000001 | 0.000001 |
| -9 | 0.005713 | 0.000000 | 0.000000 | 0.000000 |
| -8 | 0.000060 | 0.000000 | 0.000000 | 0.000000 |
| -7 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| -6 | 12.896969 | 0.001396 | 0.000003 | 0.000003 |
| -5 | 2.244735 | 0.000038 | 0.000000 | 0.000000 |
| -4 | 0.256818 | 0.000000 | 0.000000 | 0.000000 |
| -3 | 0.015216 | 0.000000 | 0.000000 | 0.000000 |
| -2 | 0.000274 | 0.000000 | 0.000000 | 0.000000 |
| -1 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 0 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 1 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 2 | 0.000274 | 0.000000 | 0.000000 | 0.000000 |
| 3 | 0.015216 | 0.000000 | 0.000000 | 0.000000 |
| 4 | 0.256818 | 0.000000 | 0.000000 | 0.000000 |
| 5 | 2.244735 | 0.000038 | 0.000000 | 0.000000 |
| 6 | 12.896969 | 0.001396 | 0.000003 | 0.000003 |
| 7 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 8 | 0.000060 | 0.000000 | 0.000000 | 0.000000 |
| 9 | 0.005713 | 0.000000 | 0.000000 | 0.000000 |
| 10 | 0.125187 | 0.000001 | 0.000001 | 0.000001 |

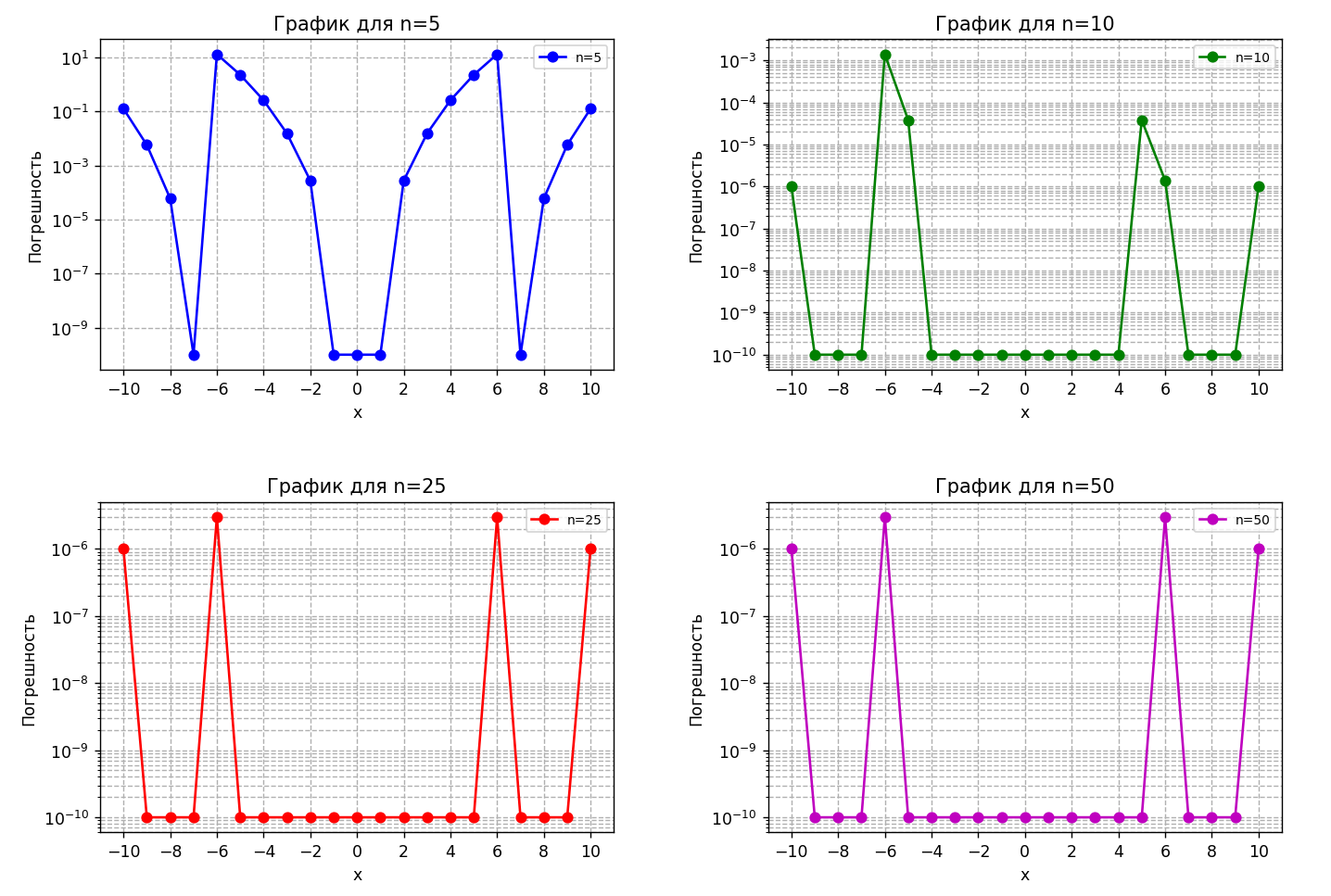
****

Рисунок 11. Графики cos(x) для обратного суммирования.

* 1. **ln(1 + x)** (рисунок 12)

Таблица 7. ln(1 + x) обратное суммирование.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | 5 | 10 | 25 | 50 |
| x | **Погрешность** | **Погрешность** | **Погрешность** | **Погрешность** |
| -0.9 | 0.472462 | 0.183838 | 0.019500 | 0.000789 |
| -0.8 | 0.150835 | 0.030563 | 0.000513 | 0.000001 |
| -0.7 | 0.051000 | 0.005113 | 0.000011 | 0.000000 |
| -0.6 | 0.016339 | 0.000740 | 0.000000 | 0.000000 |
| -0.5 | 0.004605 | 0.000082 | 0.000000 | 0.000000 |
| -0.4 | 0.001044 | 0.000006 | 0.000000 | 0.000000 |
| -0.3 | 0.000164 | 0.000000 | 0.000000 | 0.000000 |
| -0.2 | 0.000013 | 0.000000 | 0.000000 | 0.000000 |
| -0.1 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 0.0 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 0.1 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 0.2 | 0.000009 | 0.000000 | 0.000000 | 0.000000 |
| 0.3 | 0.000097 | 0.000000 | 0.000000 | 0.000000 |
| 0.4 | 0.000509 | 0.000003 | 0.000000 | 0.000000 |
| 0.5 | 0.001827 | 0.000030 | 0.000000 | 0.000000 |
| 0.6 | 0.005148 | 0.000213 | 0.000000 | 0.000000 |
| 0.7 | 0.012294 | 0.001096 | 0.000002 | 0.000000 |
| 0.8 | 0.026016 | 0.004511 | 0.000066 | 0.000000 |
| 0.9 | 0.050219 | 0.015656 | 0.001332 | 0.000048 |
| 1.0 | 0.472462 | 0.183838 | 0.019500 | 0.000789 |

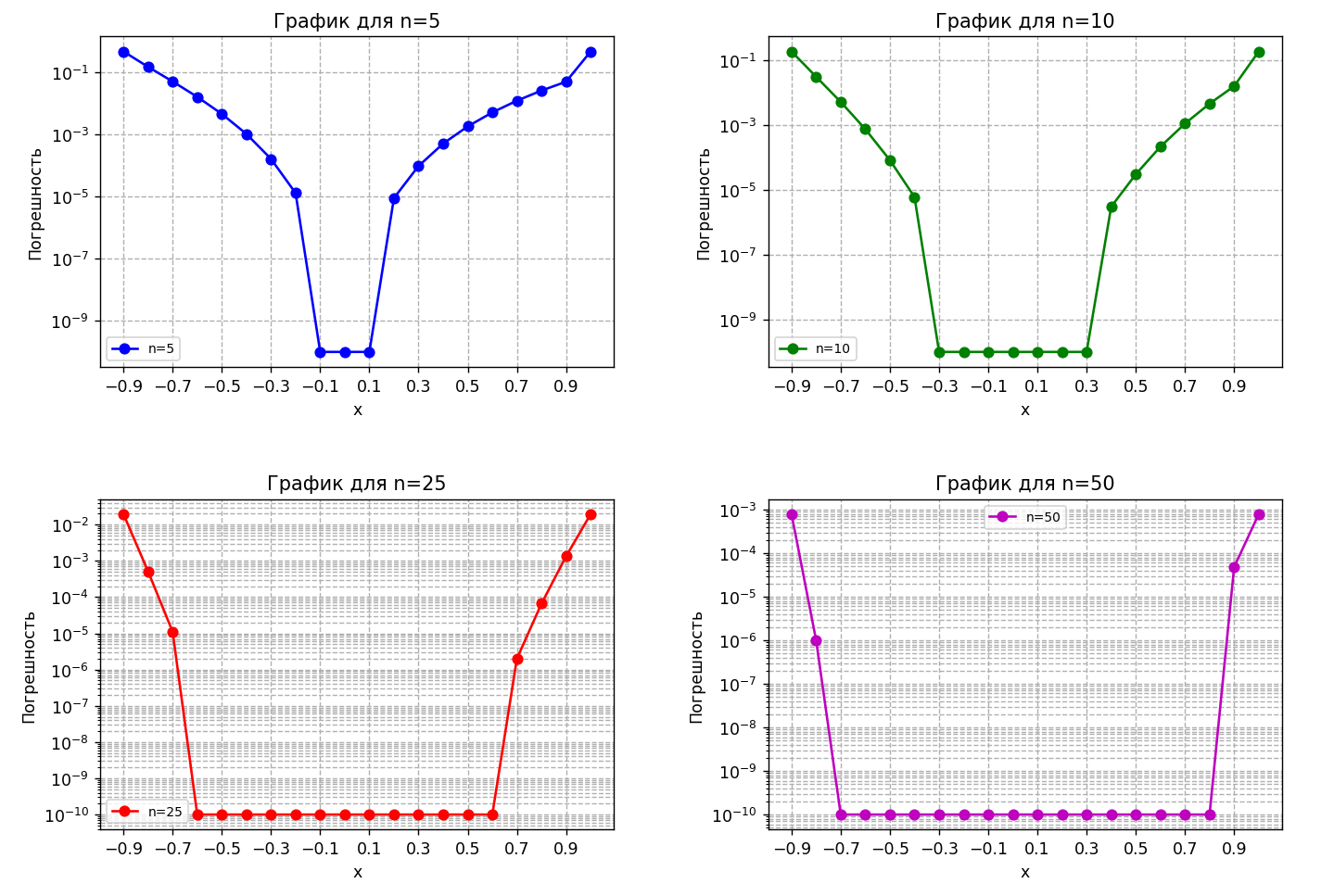
****

Рисунок 12. Графики ln(1 + x) для обратного суммирования.

* 1. (рисунок 13)

Таблица 8. обратное суммирование.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | 5 | 10 | 25 | 50 |
| x | **Погрешность** | **Погрешность** | **Погрешность** | **Погрешность** |
| -10 | 290.999955 | 1413.144699 | 0.464081 | 0.000030 |
| -9 | 184.374877 | 518.589418 | 0.034300 | 0.000060 |
| -8 | 110.333000 | 168.512848 | 0.001854 | 0.000008 |
| -7 | 61.374088 | 46.911755 | 0.000068 | 0.000000 |
| -6 | 30.997521 | 10.659621 | 0.000001 | 0.000001 |
| -5 | 13.701596 | 1.833843 | 0.000001 | 0.000001 |
| -4 | 4.981684 | 0.210556 | 0.000000 | 0.000000 |
| -3 | 1.325213 | 0.012733 | 0.000000 | 0.000000 |
| -2 | 0.197998 | 0.000238 | 0.000000 | 0.000000 |
| -1 | 0.007121 | 0.000000 | 0.000000 | 0.000000 |
| 0 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 1 | 0.009948 | 0.000000 | 0.000000 | 0.000000 |
| 2 | 0.389056 | 0.000344 | 0.000000 | 0.000000 |
| 3 | 3.710537 | 0.022144 | 0.000000 | 0.000000 |
| 4 | 20.264814 | 0.444002 | 0.000002 | 0.000002 |
| 5 | 83.038159 | 4.723675 | 0.000003 | 0.000003 |
| 6 | 288.428793 | 33.857352 | 0.000009 | 0.000009 |
| 7 | 906.924815 | 185.883769 | 0.000102 | 0.000020 |
| 8 | 2683.957987 | 844.731180 | 0.003397 | 0.000021 |
| 9 | 7657.708928 | 3343.265568 | 0.070256 | 0.000057 |
| 10 | 21382.132421 | 11939.891576 | 1.034154 | 0.001002 |

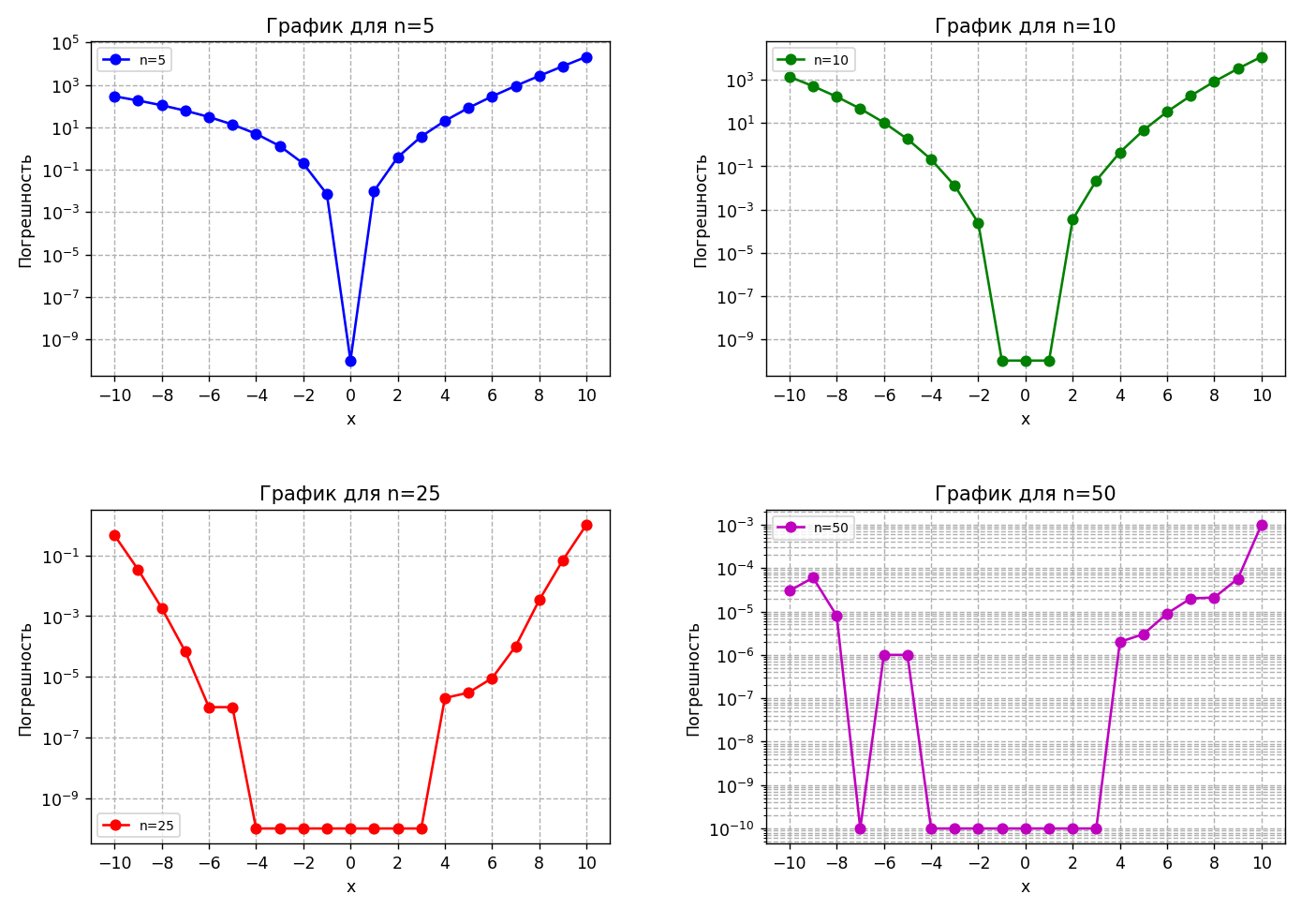


Рисунок 13. Графики для обратного суммирования.

**Попарное суммирование**

* 1. **sin(x)** (рисунок 14)

Таблица 9. sin(x) попарное суммирование.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | 5 | 10 | 25 | 50 |
| x | **Погрешность** | **Погрешность** | **Погрешность** | **Погрешность** |
| -10 | 0.042962 | 0.000000 | 0.000000 | 0.000000 |
| -9 | 0.001423 | 0.000000 | 0.000000 | 0.000000 |
| -8 | 0.000009 | 0.000000 | 0.000000 | 0.000000 |
| -7 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| -6 | 7.307989 | 0.000398 | 0.000003 | 0.000003 |
| -5 | 1.048554 | 0.000010 | 0.000001 | 0.000001 |
| -4 | 0.095074 | 0.000000 | 0.000000 | 0.000000 |
| -3 | 0.004193 | 0.000000 | 0.000000 | 0.000000 |
| -2 | 0.000050 | 0.000000 | 0.000000 | 0.000000 |
| -1 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 0 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 1 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 2 | 0.000050 | 0.000000 | 0.000000 | 0.000000 |
| 3 | 0.004193 | 0.000000 | 0.000000 | 0.000000 |
| 4 | 0.095074 | 0.000000 | 0.000000 | 0.000000 |
| 5 | 1.048554 | 0.000010 | 0.000001 | 0.000001 |
| 6 | 7.307989 | 0.000398 | 0.000003 | 0.000003 |
| 7 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 8 | 0.000009 | 0.000000 | 0.000000 | 0.000000 |
| 9 | 0.001423 | 0.000000 | 0.000000 | 0.000000 |
| 10 | 0.042962 | 0.000000 | 0.000000 | 0.000000 |

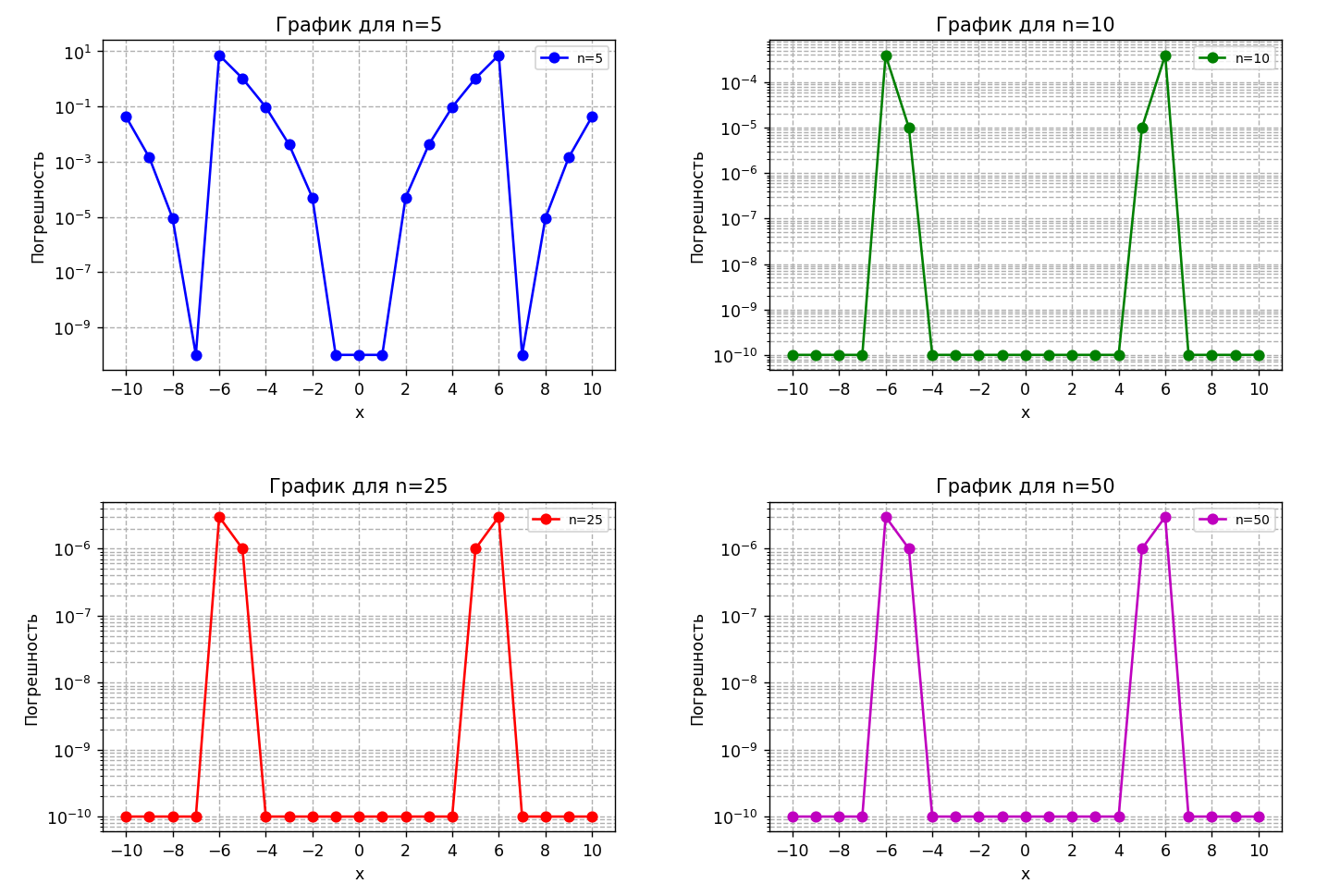
****

Рисунок 14. Графики sin(x) для попарного суммирования.

* 1. **cos(x)** (рисунок 15)

Таблица 10. cos(x) попарное суммирование.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | 5 | 10 | 25 | 50 |
| x | **Погрешность** | **Погрешность** | **Погрешность** | **Погрешность** |
| -10 | 0.125187 | 0.000001 | 0.000000 | 0.000000 |
| -9 | 0.005713 | 0.000000 | 0.000000 | 0.000000 |
| -8 | 0.000060 | 0.000000 | 0.000000 | 0.000000 |
| -7 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| -6 | 12.896969 | 0.001396 | 0.000003 | 0.000003 |
| -5 | 2.244735 | 0.000037 | 0.000000 | 0.000000 |
| -4 | 0.256818 | 0.000000 | 0.000000 | 0.000000 |
| -3 | 0.015216 | 0.000000 | 0.000000 | 0.000000 |
| -2 | 0.000274 | 0.000000 | 0.000000 | 0.000000 |
| -1 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 0 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 1 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 2 | 0.000274 | 0.000000 | 0.000000 | 0.000000 |
| 3 | 0.015216 | 0.000000 | 0.000000 | 0.000000 |
| 4 | 0.256818 | 0.000000 | 0.000000 | 0.000000 |
| 5 | 2.244735 | 0.000037 | 0.000000 | 0.000000 |
| 6 | 12.896969 | 0.001396 | 0.000003 | 0.000003 |
| 7 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 8 | 0.000060 | 0.000000 | 0.000000 | 0.000000 |
| 9 | 0.005713 | 0.000000 | 0.000000 | 0.000000 |
| 10 | 0.125187 | 0.000001 | 0.000000 | 0.000000 |

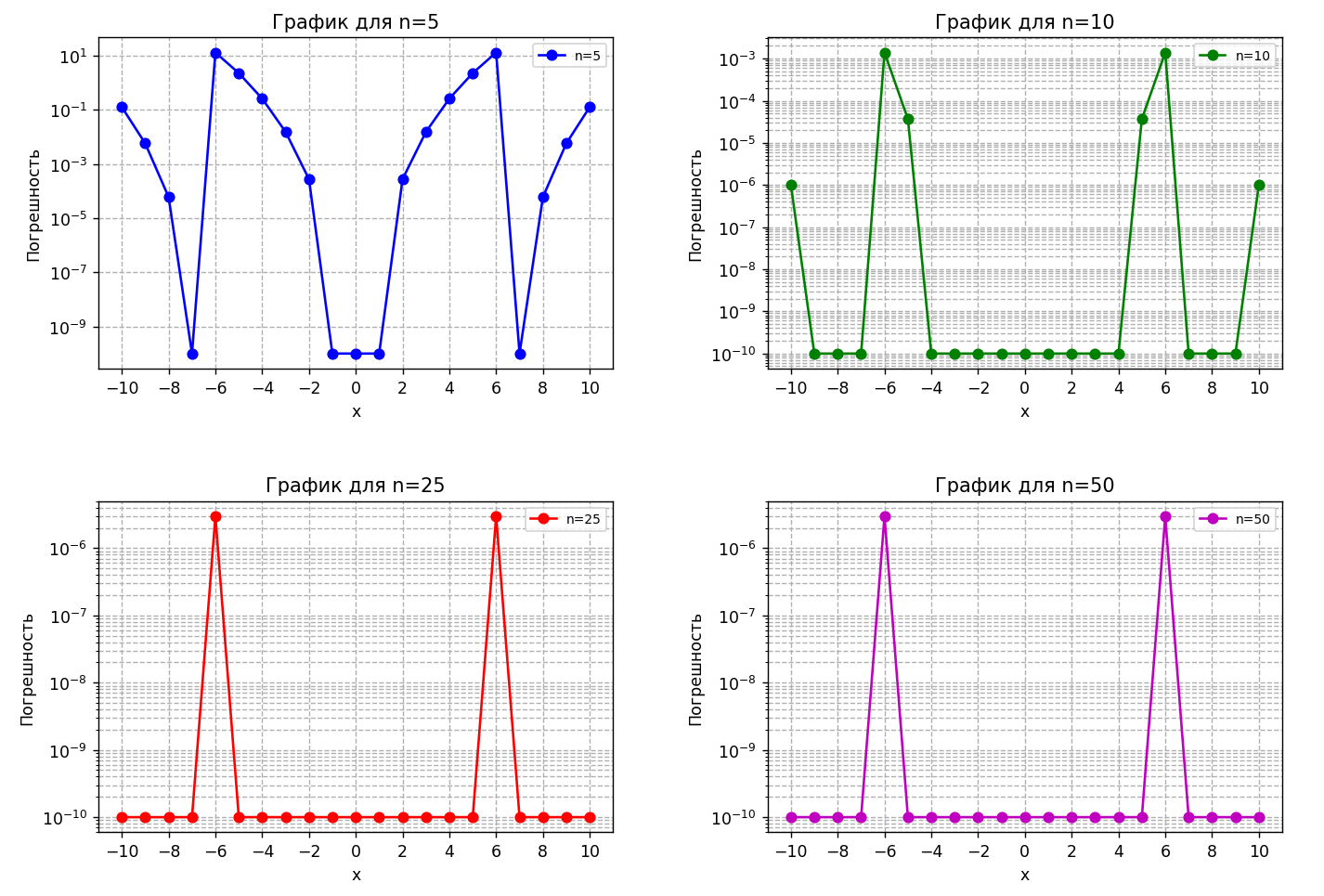
****

Рисунок 15. Графики cos(x) для попарного суммирования.

* 1. **ln(1 + x)** (рисунок 16)

Таблица 11. ln(1 + x) попарное суммирование.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | 5 | 10 | 25 | 50 |
| x | **Погрешность** | **Погрешность** | **Погрешность** | **Погрешность** |
| -0.9 | 0.472462 | 0.183838 | 0.019500 | 0.000789 |
| -0.8 | 0.150835 | 0.030563 | 0.000513 | 0.000001 |
| -0.7 | 0.051000 | 0.005113 | 0.000011 | 0.000000 |
| -0.6 | 0.016339 | 0.000740 | 0.000000 | 0.000000 |
| -0.5 | 0.004605 | 0.000082 | 0.000000 | 0.000000 |
| -0.4 | 0.001044 | 0.000006 | 0.000000 | 0.000000 |
| -0.3 | 0.000164 | 0.000000 | 0.000000 | 0.000000 |
| -0.2 | 0.000013 | 0.000000 | 0.000000 | 0.000000 |
| -0.1 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 0.0 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 0.1 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 0.2 | 0.000009 | 0.000000 | 0.000000 | 0.000000 |
| 0.3 | 0.000097 | 0.000000 | 0.000000 | 0.000000 |
| 0.4 | 0.000509 | 0.000003 | 0.000000 | 0.000000 |
| 0.5 | 0.001827 | 0.000030 | 0.000000 | 0.000000 |
| 0.6 | 0.005148 | 0.000213 | 0.000000 | 0.000000 |
| 0.7 | 0.012294 | 0.001096 | 0.000002 | 0.000000 |
| 0.8 | 0.026016 | 0.004511 | 0.000066 | 0.000000 |
| 0.9 | 0.050219 | 0.015656 | 0.001332 | 0.000048 |
| 1.0 | 0.090186 | 0.047512 | 0.019600 | 0.009900 |

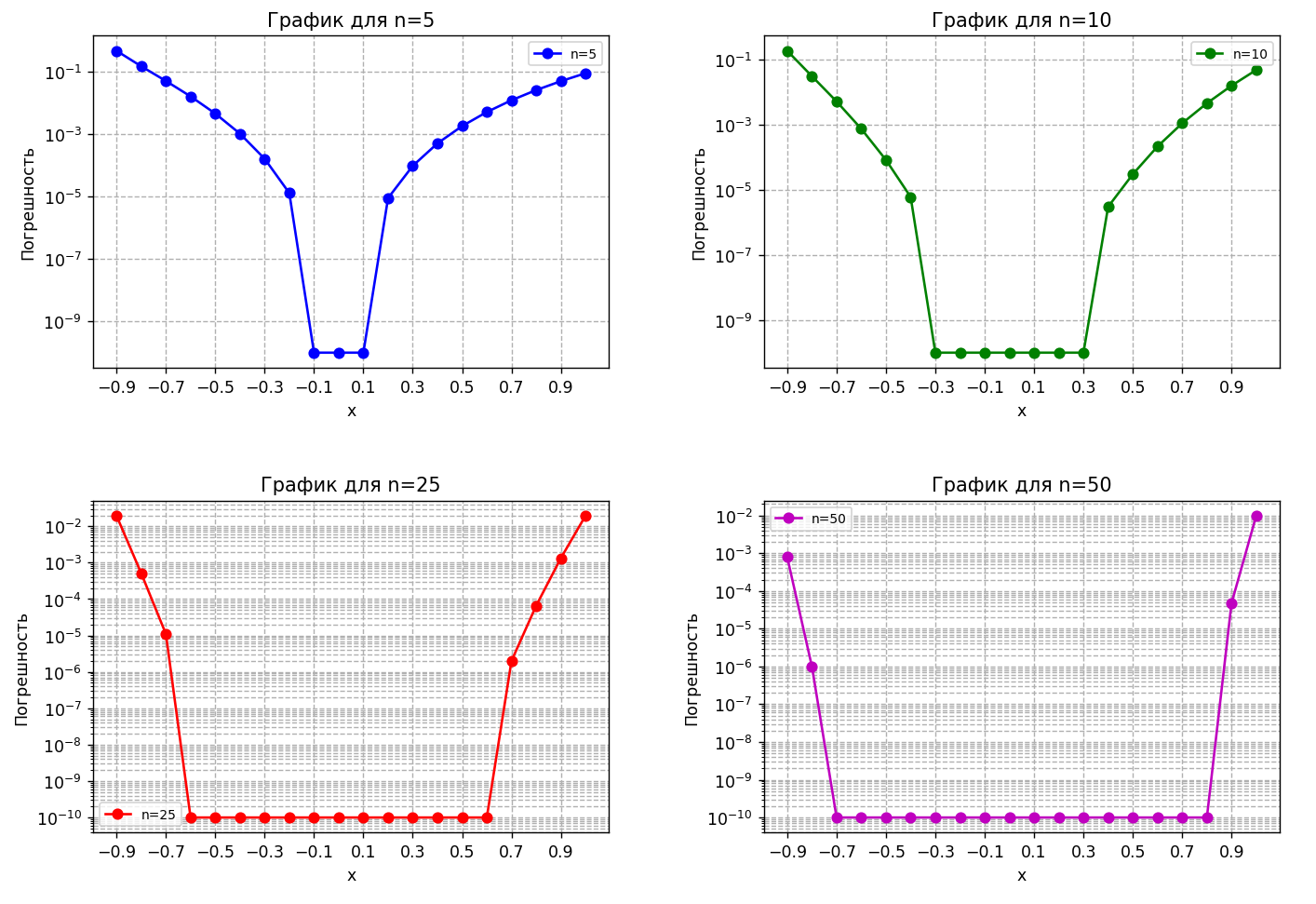
****

Рисунок 16. Графики ln(1 + x) для попарного суммирования.

* 1. n от 1 до 20, x = 5.0 (рисунок 17)

Таблица 12. попарное суммирование.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | 5 | 10 | 25 | 50 |
| x | **Погрешность** | **Погрешность** | **Погрешность** | **Погрешность** |
| -10 | 290.999955 | 1413.144821 | 0.464130 | 0.000108 |
| -9 | 184.374877 | 518.589418 | 0.034315 | 0.000021 |
| -8 | 110.333000 | 168.512848 | 0.001872 | 0.000013 |
| -7 | 61.374092 | 46.911755 | 0.000065 | 0.000003 |
| -6 | 30.997521 | 10.659623 | 0.000000 | 0.000002 |
| -5 | 13.701596 | 1.833843 | 0.000000 | 0.000000 |
| -4 | 4.981684 | 0.210556 | 0.000000 | 0.000000 |
| -3 | 1.325213 | 0.012734 | 0.000000 | 0.000000 |
| -2 | 0.197998 | 0.000238 | 0.000000 | 0.000000 |
| -1 | 0.007121 | 0.000000 | 0.000000 | 0.000000 |
| 0 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 1 | 0.009948 | 0.000000 | 0.000000 | 0.000000 |
| 2 | 0.389056 | 0.000343 | 0.000001 | 0.000001 |
| 3 | 3.710537 | 0.022142 | 0.000002 | 0.000002 |
| 4 | 20.264814 | 0.444006 | 0.000002 | 0.000002 |
| 5 | 83.038159 | 4.723675 | 0.000034 | 0.000034 |
| 6 | 288.428793 | 33.857382 | 0.000083 | 0.000083 |
| 7 | 906.924815 | 185.883830 | 0.000142 | 0.000264 |
| 8 | 2683.957987 | 844.731180 | 0.003153 | 0.000265 |
| 9 | 7657.708928 | 3343.265568 | 0.071232 | 0.000920 |
| 10 | 21382.132421 | 11939.891576 | 1.032201 | 0.002955 |

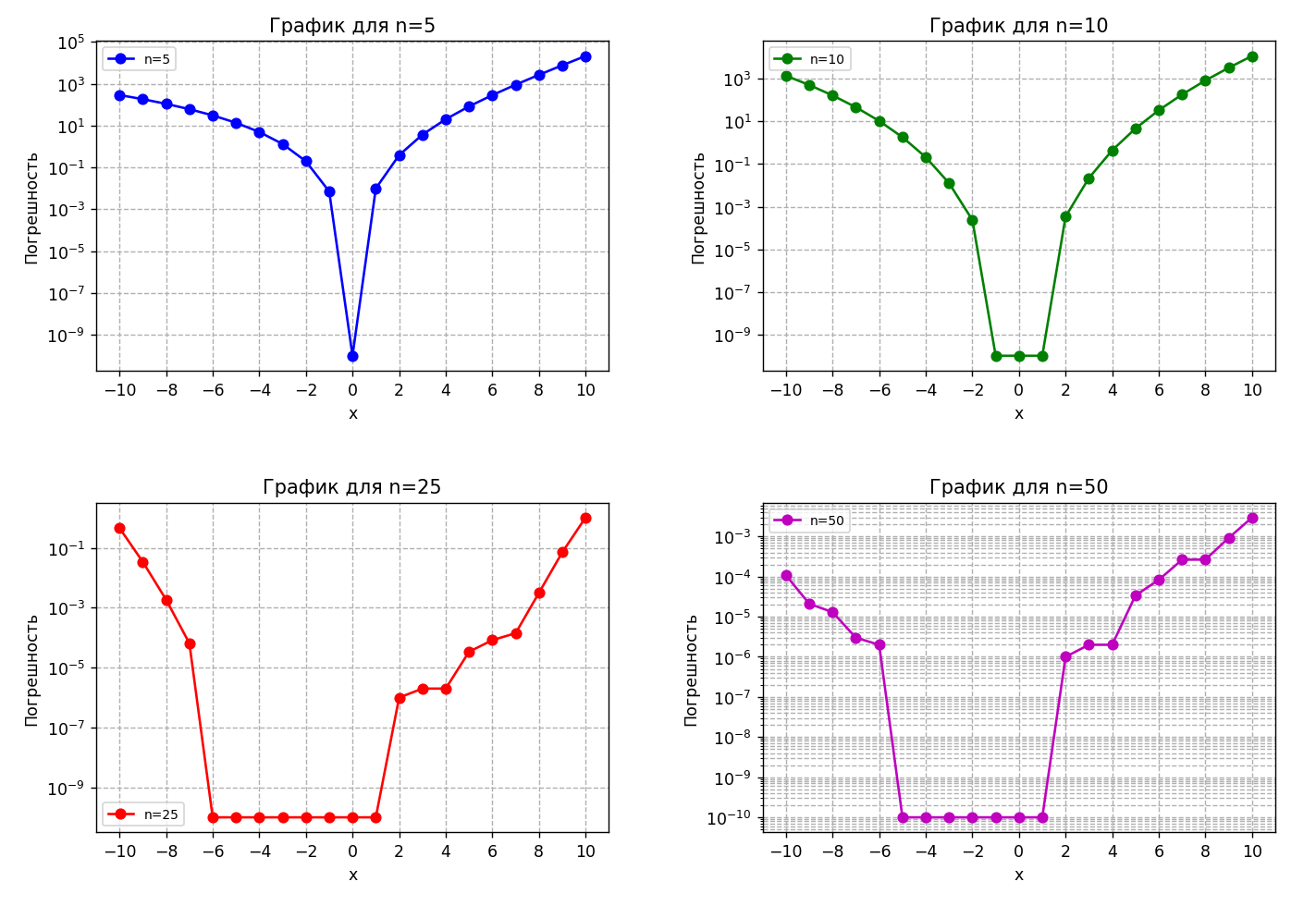


Рисунок 17. Графики для попарного суммирования.

### **Вывод из результатов экспериментов**

На основе проведенных в рамках лабораторной работы экспериментов по вычислению функций **sin(x)**, **cos(x)**, **ln(1 + x)** и с использованием разложения в ряд Тейлора и применения трех методов суммирования (прямого, обратного и попарного), можно сформулировать следующие ключевые выводы:

#### **Точность вычислений и количество членов ряда**

Для достижения приемлемой точности вычислений необходимо учитывать количество членов ряда **n**, особенно при увеличении абсолютного значения аргумента **|x|**. Увеличение числа членов **n** существенно снижает погрешность.

* Например, для функции **sin(x)** при **x = 6** погрешность уменьшается с **7.307989** при **n = 5** до **0.000003** при **n = 25**.
* Аналогичная тенденция наблюдается для при **x = 10**: погрешность сокращается с **21382.132421** при **n = 5** до **0.002955** при **n = 50**.

Эти данные свидетельствуют о том, что для больших значений **|x|** требуется использовать большее число членов ряда (например, **n ≥ 25**), чтобы обеспечить высокую точность результатов, что является важным аспектом при выполнении вычислений в рамках лабораторной работы.

#### **Отличия методов суммирования**

Сравнительный анализ прямого, обратного и попарного методов суммирования показал, что их влияние на точность вычислений минимально и в большинстве случаев не является определяющим фактором.

* Для функций **sin(x)**, **cos(x)** и **ln(1 + x)** результаты всех трех методов практически идентичны. Различия в погрешности становятся пренебрежимо малыми при увеличении **n**. Например, для **sin(x)** при **x = 6** и **n = 25** погрешность составляет **0.000003** независимо от выбранного метода суммирования.
* Для функции при больших **|x|** и малом числе членов ряда (**n**) наблюдаются незначительные различия. Так, при **x = 10** и **n = 25** погрешность для обратного суммирования составляет **1.034154**, тогда как для прямого и попарного — **1.032201**. Однако при увеличении **n** до **50** эти различия нивелируются, и погрешность стабилизируется на уровне **0.002955** для всех методов.

Таким образом, метод суммирования не оказывает существенного влияния на итоговую точность при достаточном количестве членов ряда. Незначительные расхождения проявляются лишь в специфических условиях (малые **n**, большие **|x|**) и не имеют критического значения для целей лабораторной работы.

# Заключение

В ходе выполнения данной лабораторной работы была успешно достигнута поставленная цель — реализованы вычисления математических функций , и с использованием разложения в ряд Тейлора. Разработанная программа на языке C позволила вычислить значения указанных функций с применением трёх методов суммирования: прямого, обратного и попарного. Для повышения эффективности вычислений были использованы рекуррентные формулы, что исключило проблемы переполнения при больших значениях факториалов.

Результаты, полученные с использованием реализованных методов, были сравнены с эталонными значениями, вычисленными с помощью стандартных функций из библиотеки math.h. Проведённое сравнение показало, что точность вычислений напрямую зависит от количества членов ряда n: при увеличении n погрешность уменьшается, особенно для больших значений аргумента ∣x∣.

Анализ методов суммирования выявил, что различия между прямым, обратным и попарным подходами минимальны при достаточном количестве членов ряда, что подтверждает их сопоставимую эффективность в данном контексте. Таким образом, цель работы — реализация вычислений и оценка точности по сравнению с библиотечными функциями — была выполнена. Полученные результаты демонстрируют корректность разработанной программы и её применимость для численных вычислений.

# Приложение

#include <stdio.h>

#include <stdlib.h>

#define \_USE\_MATH\_DEFINES

#include <math.h>

#define PI M\_PI

void sinx(float\* terms, int n, float x)

{

x = fmodf(x, 2.0f \* (float)PI);

terms[0] = x;

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

terms[i] = (terms[i - 1] \* -1 \* x \* x) / ((2 \* i + 1) \* (2 \* i));

}

}

void cosx(float\* terms, int n, float x)

{

x = fmodf(x, 2.0f \* (float)PI);

terms[0] = 1.0f;

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

terms[i] = (terms[i - 1] \* -1 \* x \* x) / ((2 \* i - 1) \* (2 \* i));

}

}

void ln(float\* terms, int n, float x)

{

terms[0] = x;

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

terms[i] = (terms[i - 1] \* -1 \* x \* i) / (i + 1);

}

}

void expx(float\* terms, int n, float x)

{

terms[0] = 1.0f;

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

terms[i] = (terms[i - 1] \* x) / i;

}

}

// Прямое суммирование

float direct\_sum(float\* terms, int n)

{

float fx = 0.0f;

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

fx += terms[i];

}

return fx;

}

// Обратное суммирование

float reverse\_sum(float\* terms, int n)

{

float fx = 0.0f;

for (int i = n - 1; i >= 0; i--) {

fx += terms[i];

}

return fx;

}

// Попарное суммирование

float pair\_sum(float\* terms, int n)

{

float fx = 0.0f;

for (int i = 0; i < n / 2; i++) {

fx += terms[2 \* i];

if (2 \* i + 1 < n) {

fx += terms[2 \* i + 1];

}

}

if (n % 2 != 0) {

fx += terms[n - 1];

}

return fx;

}

void filling(float\* terms, int n, int func, float x,

float\* direct\_fx, float\* reverse\_fx,

float\* pair\_fx, double\* expected\_res)

{

if (func == 1) {

sinx(terms, n, x);

\*expected\_res = sin(x);

}

else if (func == 2) {

cosx(terms, n, x);

\*expected\_res = cos(x);

}

else if (func == 3) {

ln(terms, n, x);

\*expected\_res = log(1 + x);

}

else {

expx(terms, n, x);

\*expected\_res = exp(x);

}

\*direct\_fx = direct\_sum(terms, n);

\*reverse\_fx = reverse\_sum(terms, n);

\*pair\_fx = pair\_sum(terms, n);

}

int main()

{

int n, func;

float x, direct\_fx, reverse\_fx, pair\_fx;

double expected\_res;

char end = 'r';

while (end != 'e')

{

printf("Select function:\n");

printf("1. sinx\n");

printf("2. cosx\n");

printf("3. ln(1+x)\n");

printf("4. e^x\n");

scanf\_s("%d", &func);

printf("Enter number of elements\n");

scanf\_s("%d", &n);

if (n <= 0)

{

printf("Incorrect 'n'. Program ends.");

break;

}

printf("Enter x for function\n");

scanf\_s("%f", &x);

float\* terms = (float\*)malloc(sizeof(float) \* n);

filling(terms, n, func, x, &direct\_fx, &reverse\_fx, &pair\_fx, &expected\_res);

printf("\nResults:\n");

printf("Direct summation: %f\n", direct\_fx);

printf("Reverse summation: %f\n", reverse\_fx);

printf("Pair summation: %f\n", pair\_fx);

printf("Expected result: %f\n", expected\_res);

printf("Max difference: %f\n", fmaxf(fmaxf(fabs(expected\_res - direct\_fx), fabs(expected\_res - reverse\_fx)), fabs(expected\_res - pair\_fx)));

free(terms);

printf("End or repeat program? (e/r)\n");

scanf\_s(" %c", &end);

}

return 0;

}