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**

**«Нахождение значений функций sin x, cos x, ex, ln(1+x) с помощью рядов Тейлора»**

**Выполнила**:

студентка группы 3821Б1ПМ2

Домрачева А.В.

**Проверил**:

преподаватель каф. МОСТ,

Волокитин В.Д.

Нижний Новгород

2021

**Содержание**

[Постановка задачи 3](#_Toc26962562)

[Метод решения 4](#_Toc26962563)

[Руководство пользователя 5](#_Toc26962564)

[Описание программной реализации 6](#_Toc26962565)

[Подтверждение корректности 7](#_Toc26962566)

[Результаты экспериментов 8](#_Toc26962567)

[Заключение 9](#_Toc26962568)

[Приложение 10](#_Toc26962569)

# Постановка задачи

Ряд Тейлора – разложение функции в бесконечную сумму степенных функций. С помощью этого ряда можно найти приближённое значение функции в окрестности заданной точки.

Задачей данной лабораторной работы было нахождение значений для четырёх функций: sin x, cos x, ln(1+x), ex, с помощью суммирования рядов Тейлора.

Разложения по формуле Тейлора для заданных функций выглядят так:

Задача должна быть выполнена с помощью прямого, обратного и попарного суммирования. Каждое следующее слагаемое должно находиться с использованием предыдущего. Для проверки корректности работы программы используются стандартные функции из библиотеки math.h.

После написания программы следует сравнить результаты после прямого, обратного и попарного суммирования с табличными значениями.

# Метод решения

Программа получает на вход значение аргумента функции. По умолчанию ряд в программе состоит из 100 слагаемых. Первое слагаемое каждой функции задано в программе изначально. Затем с помощью цикла for программа находит каждое следующее слагаемое и записывает его в массив. Это происходит как при прямом, так и при обратном и при попарном суммировании. После чего с помощью ещё одного цикла происходит сложение элементов ряда.

В конце работы программы получаются четыре значения для каждой функции: результат прямого суммирования, результат обратного суммирования, результат попарного суммирования и значения стандартной функции в заданной точке, для того, чтобы можно было проверить корректность работы программы.

Также программа считает ошибку для каждого из способов суммирования, Вычисление ошибки происходить вычитанием из полученного результата значения стандартной функции.

# Руководство пользователя

В начале работы программы пользователь видит строку «Введите x:», после чего ему следует ввести число (x имеет тип float).

В завершение работы пользователь может видеть четыре значения для каждой функции.

sin\_прямой – результат прямого суммирования

sin\_обратный – результат обратного суммирования

sin\_попарный – результат попарного суммирования

sin\_math – значение стандартной функции

Далее три строчки со значением ошибок работы программы. Для корректного анализа полученных ошибок следует брать их по модулю.

# Описание программной реализации

Программа считывает введённое пользователем значение x. Далее вызываются соответствующие функции для каждого из случаев. Функции sin\_f, cos\_f, ln\_f, exp\_f выполняют подсчёт элементов ряда, а функции sum\_str, sun\_obr, sum\_par выполняют суммирование элементов прямым способом, обратным или попарным соответственно. Также учитывается, что область определения функции ln(1+x) от -1 до 1 невключая.

При подсчёте синуса и косинуса функции вначале проверяют лежит ли аргумент в диапазоне от 0 до 2π. Если же нет, то приводят его к нужным значениям. В случае с логарифмом и экспонентой необходимости в этом нет.

Первое значение ряда в программах не считается, а задано изначально и записано в массив, поэтому с помощью цикла for находятся слагаемые начиная со второго и также последовательно записываются в массив.

С помощью ещё одного цикла for происходит сложение элементов полученного массива. В случае с прямым суммированием элементы вызываются с 0 до 99. А при обратном суммировании с 99 до 0. Попарное суммирование происходит в два этапа: первый – сложение первого и второго, третьего и четвертого… слагаемого, второй – сложение ранее полученных попарный сумм.

В завершении программа выводит полученные результаты.

# Подтверждение корректности

Для подтверждения корректности работы программы стоит рассмотреть абсолютные и относительные ошибки результатов экспериментов.

В таблицах значения относительных ошибок представлены в %

y = sin x

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | -4 | -0,5 | 0,3 | 11 | 105 | 1.000 | 12.000 |
| Прямой | 0 | 2\*10-4 | 0 | 0 | 3\*10-4 | 0,06 | 2,5 |
| Обратный | 0 | 4\*10-4 | 0 | 0 | 2\*10-4 | 0,06 | 2,5 |
| Попарный | 0 | 0 | 0 | 0 | 3\*10-4 | 0,06 | 2,5 |

y = cos x

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | -4 | -0,5 | 0,3 | 11 | 105 | 1.000 | 12.000 |
| Прямой | 0 | 3,4\*10-4 | 0 | 0,02 | 4\*10-3 | 0,13 | 3,9 |
| Обратный | 0 | 2,3\*10-4 | 0 | 0,02 | 4\*10-3 | 0,13 | 3,9 |
| Попарный | 0 | 4,5\*10-4 | 0 | 0,02 | 4\*10-3 | 0,13 | 3,9 |

y = ex

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | -4 | -0,5 | 0,3 | 11 | 38 | 50 | 70 |
| Прямой | 0 | 0 | 0 | 2\*10-5 | 1,3\*10-5 | 8,6\*10-5 | 0,04 |
| Обратный | 0 | 0 | 0 | 1,3\*10-5 | 6,7\*10-6 | 7,6\*10-5 | 0,04 |
| Попарный | 0 | 0 | 0 | 1,3\*10-5 | 0 | 4,3\*10-5 | 0,04 |

y = ln (1+x)

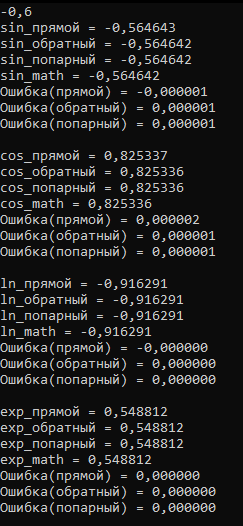
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | -0,9 | -0,5 | -0,3 | 0,01 | 0,3 | 0,6 | 0,98 |
| Прямой | 1,3\*10-4 | 0 | 0 | 0 | 0 | 0 | 0,09 |
| Обратный | 8,6\*10-5 | 0 | 0 | 0 | 0 | 0 | 0,09 |
| Попарный | 8,6\*10-5 | 0 | 0 | 0 | 0 | 0 | 0,09 |

Таким образом, мы видим, что с увеличением значения x уменьшается точность результата, но несмотря на это значения погрешности остаётся достаточно мало. Наиболее точный результат получается при попарном суммировании.

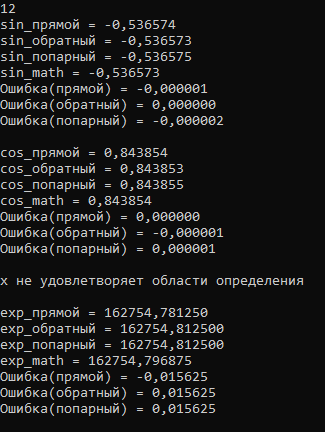
# Результаты экспериментов

В качестве эксперимента рассмотрим два случая.

x = - 0.6



x = 12



Получаем, что программа корректно учитывает область определения функции ln(1+x) и считает с допустимой неточностью значения остальных функций.

# Заключение

В результате выполненной лабораторной работы был реализован алгоритм поиска значений функций sin x, cos x, ln (1+x), ex с помощью рядов Тейлора.

По данным из таблиц, приведённых выше, можно сделать вывод, что с увеличением числа x увеличивается ошибка программы. В случае с логарифмом ошибка тем больше, чем ближе число к границам области определения функции.

В результате экспериментов также получается, что наиболее точный подсчёт результата получается при попарном суммировании.

# Приложение

