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**

**«Функции sin, cos, exp, ln»**

**Выполнил**:

студент группы 3821Б1ПМ2

Рымарь А. А

**Проверил**:

преподаватель каф. МОСТ,

Волокитин В. Д.

Нижний Новгород

2021

**Содержание**

Постановка задачи……………………………………………………………………...3

Метод решения…………………………………………………………………………4

Руководство пользования……………………………………………………………...5

Описание программной реализации…………………………………………………..6

Подтверждение корректности…………………………………………………………8

Результаты экспериментов…………………………………………………………….9

Заключение…………………………………………………………………………….13

2

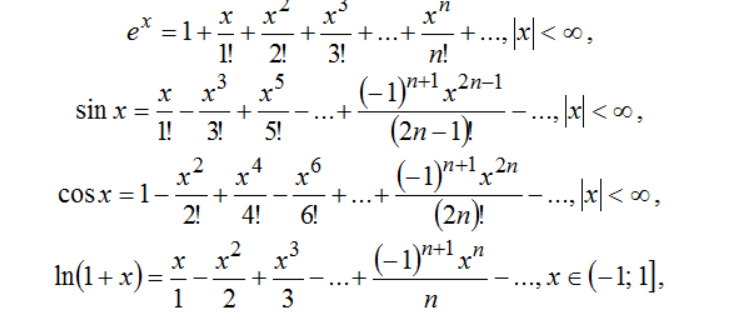
**Постановка задачи**

Целью лабораторной работы являлась реализовать на языке программирования Си следующие функции: sin, cos, exp, ln через ряд Маклорена(Ряд Тейлора вблизи точки x = 0). Необходимо получить три результата, просуммировав элементы ряда Маклорена тремя разными способами: прямым, обратным и попарным суммированием. После этого требуется найти абсолютную и относительную ошибку при вычислении значения всеми видами суммирования. Надо провести эксперименты для подтверждения корректности поиска значений функций с помощью ряда Маклорена, описать способ проведения экспериментов и сделать вывод по полученным результатам.

3

**Метод решения**

Для поиска значений функций в разных точках по заданию мы должны были пользоваться рядом Маклорена:



В моей реализации программы элементы ряда Маклорена помещаются в массив, массив заполняется последовательно, при заполнении каждого элемента массива используется предыдущее слагаемое ряда Маклорена, для этого мной были введены следующие функции: nsin, ncos, nexp, nln(Первый элемент массива задается вручную вне цикла заполнения массива). Этот массив передается в функцию сложения слагаемых ряда Маклорена. Функция stright\_sum суммирует элементы передаваемого в нее массива(слагаемые ряда Маклорена) напрямую, функция back\_sum суммирует элементы массива с конца, а функция couple\_sum суммирует элементы массива попарно, а именно первые два элемента, вторые два элемента и так далее. Разные способы суммирования дают разные результаты из-за особенностей округления числе типа double в языке Си. Функции суммирования возвращают значение функции в конкретной точке, полученное данным способом суммирования. Далее для поиска абсолютной и относительной ошибки в функцию check\_err передается значение, возвращаемое функцией суммирования, и правильное значение функции в данной точке(функция, встроенная в Си).

4

**Руководство пользователя**

В начале программы пользователь вводит значение n(количество слагаемых, используемых в вычислении функций в определенной точке) и меняет значение tos(type of sum) на нужную сумму. Далее пользователь вводит определенную точку в которой он хочет узнать значение функций и цифру от 1 до 4, выбирая нужную ему функцию. Программа выводит в консоль значение выбранной функции в введенной точке, полученное вычислением с помощью ряда Маклорена, значение выбранной функции в введенной точке, полученное встроенной в язык функции и значение абсолютной и относительной ошибок. Абсолютная ошибка высчитывается следующей формулой: |y – y\*|, а относительная ошибка: |y – y\*|/y\*, где y – значение, полученное вычислением через ряд Маклорена, а y\* - значение, полученное через встроенную функцию.

5

**Описание программной реализации**

double stright\_sum(double\* mas) – функция прямого суммирования. Функция принимает на вход массив с слагаемыми ряда Маклорена, элементы массива складываются в переменную res типа double поочередно с первого элемента до последнего, функция возвращает переменную res.

double back\_sum(double\* mas) – функция обратного суммирования. Функция принимает на вход массив с слагаемыми ряда Маклорена, элементы массива складываются в переменную res типа double поочередно с последнего элемента до первого, функция возвращает переменную res.

double couple\_sum(double\* mas) – функция попарного суммирования. Функция принимает на вход массив с слагаемыми ряда Маклорена, элементы массива складываются в переменную res типа double попарно, а именно: первые два элемента массива + вторые два элемента и тд, функция возвращает переменную res.

double nsin(double prev, int i, double x) – функция поиска следующего слагаемого ряда Маклорена для синуса. Функция принимает на вход предыдущий элемент ряда Маклорена, номер искомого элемента, точка, в которой будет производиться поиск синуса. Функция возвращает искомый элемент ряда Маклорена.

double ncos(double prev, int i, double x) – функция поиска следующего слагаемого ряда Маклорена для косинус. Функция принимает на вход предыдущий элемент ряда Маклорена, номер искомого элемента, точка, в которой будет производиться поиск косинуса. Функция возвращает искомый элемент ряда Маклорена.

double nexp(double prev, int i, double x) – функция поиска следующего слагаемого ряда Маклорена для экспоненты. Функция принимает на вход предыдущий элемент ряда Маклорена, номер искомого элемента, точка, в которой будет производиться поиск экспоненты. Функция возвращает искомый элемент ряда Маклорена.

double nln(double prev, int i, double x) – функция поиска следующего слагаемого ряда Маклорена для натурального логарифма. Функция принимает на вход предыдущий элемент ряда Маклорена, номер искомого элемента, точка, в которой будет производиться поиск натурального логарифма. Функция возвращает искомый элемент ряда Маклорена.

double check\_err(double a, double b) – функция поиска относительной и абсолютной ошибки. На вход функция принимает найденное значение функция в

6

заданной точке через ряд Маклорена, значение функции в заданной точке,

найденное встроенной функции. Функция выводит в консоль найденное значение функции в заданной точке через ряд Маклорена, значение функции в заданной точке, найденное встроенной функции, абсолютную и относительную ошибку.

В int main() организовано интерактивное меню для выбора функции.

**7**

**Подтверждение корректности**

Для подтверждения корректности были использованы стандартные функции синуса, косинуса, экспоненты и натурального логарифма из стандартной библиотеки <math.h>. Для проверки корректности были высчитаны относительная и абсолютные ошибки для всех функций и для разных типов сложения. По результатам экспериментам все ошибки всех функций всех видов сумм очень маленькие, из чего можно сделать вывод, что реализация поиска синуса, косинуса, экспоненты и натурального логарифма через ряд Маклорена работает корректно.

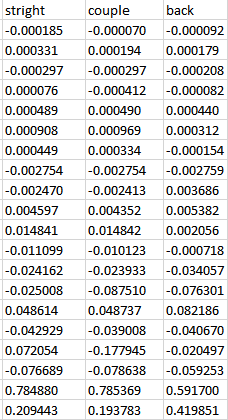
8

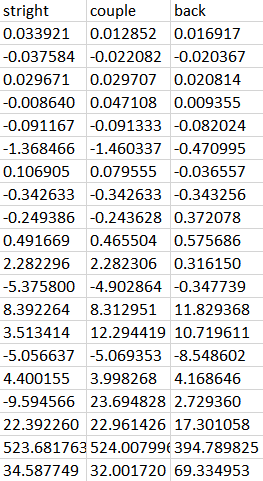
**Результаты экспериментов**

Программа вычисляет относительную погрешность всех функций для

x € [10; 20] с интервалом 0.5 для косинуса, синуса и экспоненты и в точке (1 + x) для натурального логарифма, где x € [-1; 1.6] c интервалом 0.2 отдельно для каждого способа суммирования.

1. Sin:

Абсолютная ошибка Относительная ошибка



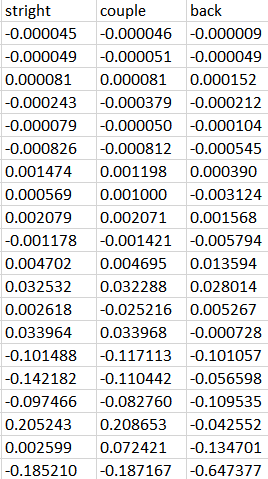
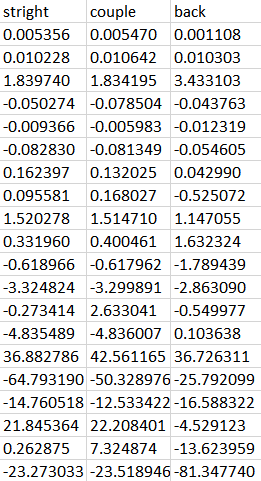
Данные были получены в промежутке x € [10; 20] с интервалом 0.5. Надо учитывать, что ряд Маклорена работает при значениях x близких к нулю. По таблице можно заметить, что при вычислении обратным суммированием, ошибка меньше, чем при использовании попарного и прямого суммирования, что значит, что обратный ход суммирования более точно считает синус через ряд Маклорена.

На значениях x больше 15, результат, полученный вычислениями с помощью ряда Маклорена начинает сильно отличаться от результата, полученного встроенной функцией.

9

2. Cos:

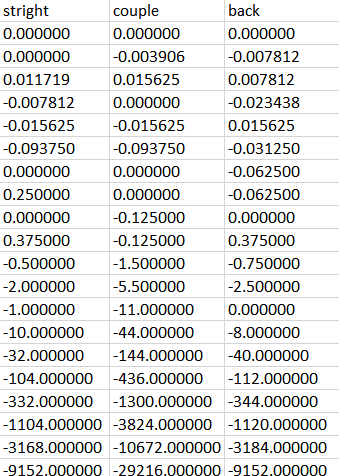
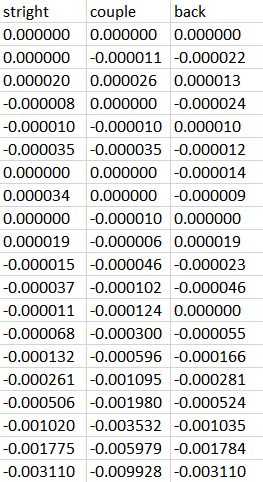
Абсолютная ошибка Относительная ошибка



Данные были получены в промежутке x € [10; 20] с интервалом 0.5. Надо учитывать, что ряд Маклорена работает при значениях x близких к нулю. По таблице можно заметить, что при вычислении обратным суммированием, ошибка меньше, чем при использовании попарного и прямого суммирования, это значит, что обратный ход суммирования более точно считает косинус через ряд Маклорена.

10

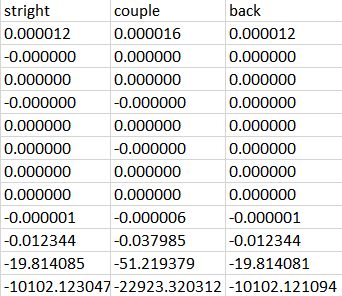
3. Exp

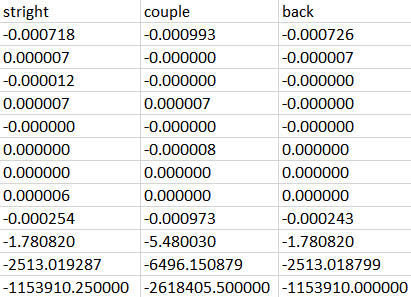
Абсолютная ошибка Относительная ошибка

Данные были получены в промежутке x € [10; 20] с интервалом 0.5. Надо учитывать, что ряд Маклорена работает при значениях x близких к нулю. По таблице можно заметить, что попарное суммирование наиболее невыгодное для вычисления экспоненты. При вычислении обратным суммированием, ошибка меньше, чем при использовании прямого суммирования, что значит, что обратный ход суммирования более точно считает синус через ряд Маклорена.

11

4. Ln:

Абсолютная ошибка Относительная ошибка



Данные были получены в промежутке x € [-1; 1.6] с интервалом 0.2. Надо учитывать, что ряд Маклорена работает при значениях x близких к нулю. По таблице можно заметить, что попарное суммирование наиболее невыгодное для вычисления натурального логарифма, ошибки при вычислении прямым и обратным суммированием почти одинаковые, но все же у обратного суммирования меньше.

12

**Заключение**

Я реализовал нахождение 4 функций(sin, cos, ln, exp) в заданных точках с помощью ряда Маклорена. Описал алгоритм их работы, вычислил погрешность для разных типов сумм и показал, как отличается относительная ошибка при выборе разных способов суммирования, с помощью таблиц. По результатам экспериментов можно утверждать, что для вычисления sin, cos, ln и exp с помощью ряда Маклорена, выгоднее всего использовать обратный ход суммирования.

13