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**

**«Подсчет функций»**

**Выполнил**:

студент/ка группы 3821Б1ПМ2

Соколов И. Д.

**Проверил**:

преподаватель каф. МОСТ,

Волокитин В.Д.

Нижний Новгород

2021

Содержание

[Постановка задачи 2](#__RefHeading___Toc711_2420105147)

[Метод решения 3](#__RefHeading___Toc713_2420105147)

[Руководство пользователя 3](#__RefHeading___Toc715_2420105147)

[Описание программной реализации 3](#__RefHeading___Toc717_2420105147)

[Подтверждение корректности 3](#__RefHeading___Toc719_2420105147)

[Результаты экспериментов 3](#__RefHeading___Toc721_2420105147)

[Заключение 3](#__RefHeading___Toc723_2420105147)

# Постановка задачи

Цель лабораторной работы:

1) Реализовать подсчет функций sinx, cosx, expx и ln(1+x) двумя методами.

2) Описать программную реализацию.

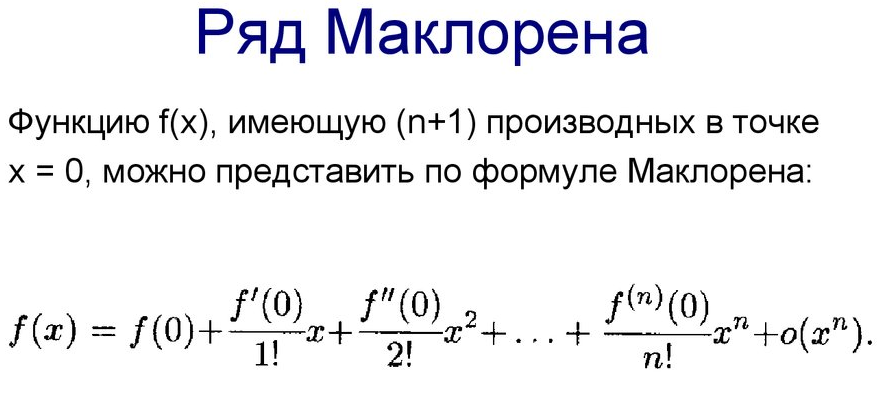
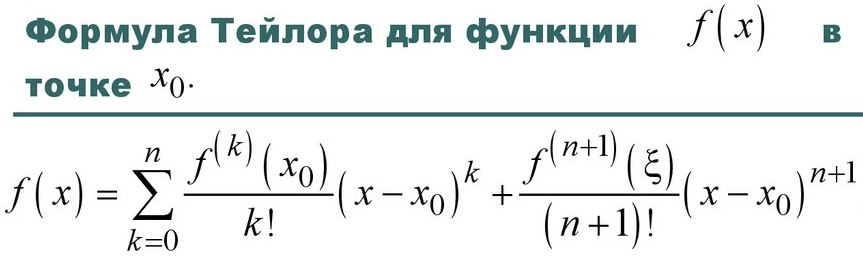
3) Подтвердить корректность данной программы (выявить сходимость функций).

4)Показать результаты экспериментов на графике или в таблицах.

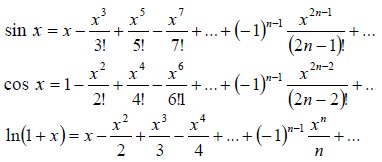
5)Сделать вывод.

# Метод решения

Для всех функций применяется один и тот же подход. Все эти функции можно разложить по формуле Тейлора в ряд Маклорена. Таким образом получим разложение f(x) с которым будет удобно работать.

Записать весь ряд или например сто элементов будет трудоемко, поэтому реализуем код который будет возвращать следующий идущий элемент в зависимости от предыдущего. К счастью формула Тейлора позволяет нам это сделать. Используя данный метод реализуем четыре функции возвращающие массив элементов ряда по одной для каждой функции. Потом реализуем три функции способа подсчета этих элементов — прямой(будет складывать элементы массива по порядку с начала), обратный(будет складывать элементы массива по порядку начиная с конца), попарный(будет складывать элементы по порядку начиная с конца, складывая сразу два элемента).

Формулы функций



# Руководство пользователя

# Как только запустится программа, появится следующее:

# Программа будет ждать пока вы введете какое то число типа double. После этого появится следующее:

# Программа будет ждать пока вы введете какое то число(1,2,3,4) типа int. После этого появится следующее:

# 

# Программа будет ждать пока вы введете какое то число(1,2,3) типа int. После этого появится следующее:

# 

# В конечном итоге программа выведет значение функции подсчитанное данной программой и стандартной программой, а также разницу между ними.

# Описание программной реализации

double\* line\_sin(double x) — создает ряд элементов функции синуса и возвращает их в виде массива. В качестве аргумента для функции взят x принимаемый на вход.

double\* line\_cos(double x) — создает ряд элементов функции косинуса и возвращает их в виде массива. В качестве аргумента для функции взят x принимаемый на вход.

double\* line\_exp(double x) — создает ряд элементов функции експоненты и возвращает их в виде массива. В качестве аргумента для функции взят x принимаемый на вход.

double\* line\_ln(double x) — создает ряд элементов функции ln(1+x) и возвращает их в виде массива. В качестве аргумента для функции взят x принимаемый на вход.

float error\_sin(double sinx, double x) — вычисляет разницу между реальным значением синуса и его функцией данной программы. На вход принимается значение функции вычисленное данной программой и число x.

float error\_cos(double sinx, double x) — вычисляет разницу между реальным значением косинуса и его функцией данной программы. На вход принимается значение функции вычисленное данной программой и число x.

float error\_exp(double sinx, double x) — вычисляет разницу между реальным значением експоненты и его функцией данной программы. На вход принимается значение функции вычисленное данной программой и число x.

float error\_ln(double sinx, double x) — вычисляет разницу между реальным значением ln(1+x) и его функцией данной программы. На вход принимается значение функции вычисленное данной программой и число x.

double forward\_sum(double\* mas) – возвращает значение функции прямым методом. На вход принимает ряд элементов.

double reverse\_sum(double\* mas) – возвращает значение функции обратным методом. На вход принимает ряд элементов.

double in\_pair\_sum(double\* mas) – возвращает значение функции попарным методом. На вход принимает ряд элементов.

# Подтверждение корректности

Для подтверждения корректности вычислений была использована функция double sin(double x) из библиотеки <math.h>.

# Результаты экспериментов

График функции синус

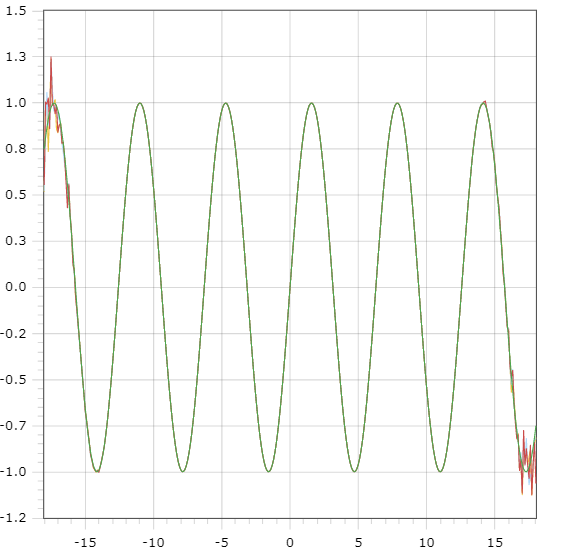
График показанный выше показывает результат вычисления трех методов и настоящий ответ для функции синуса. Содержит вычисления от -18 до 18 с шагом 0.01. Желтая - линия соответствует прямому методу, синяя — обратному, красная — попарному, зеленая настоящему.(Налезают друг на друга в данном порядке).

График функции косинуса

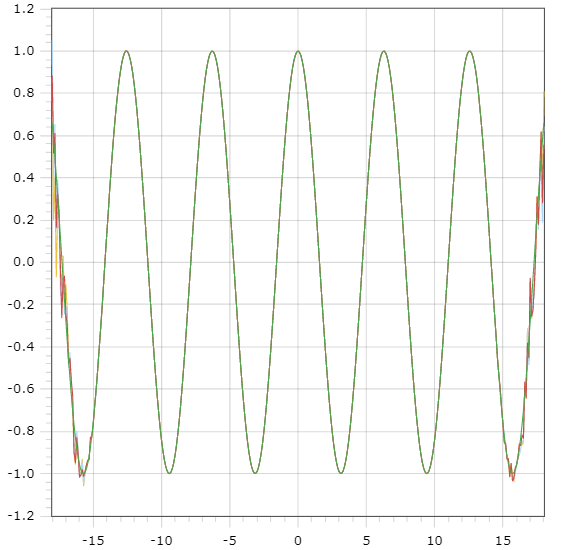
График показанный выше показывает результат вычисления трех методов и настоящий ответ для функции косинуса. Содержит вычисления от -18 до 18 с шагом 0.01. Желтая - линия соответствует прямому методу, синяя — обратному, красная — попарному, зеленая настоящему.(Налезают друг на друга в данном порядке).

График функции експоненты

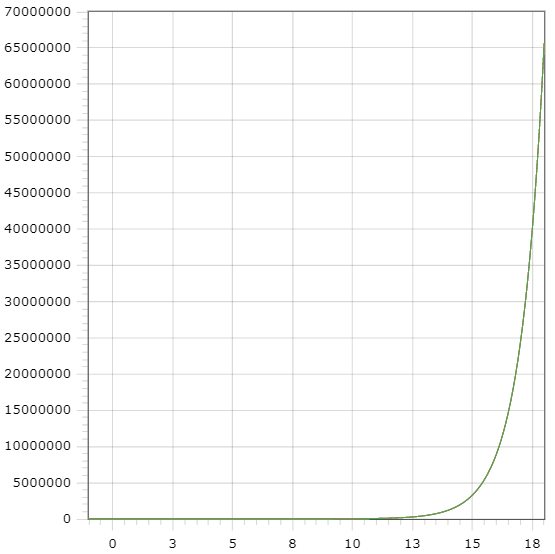
График показанный выше показывает результат вычисления трех методов и настоящий ответ для функции експоненты. Содержит вычисления от 0 до 18 с шагом 0.1. Желтая - линия соответствует прямому методу, синяя — обратному, красная — попарному, зеленая настоящему.(Налезают друг на друга в данном порядке).

График функции ln(1+x)

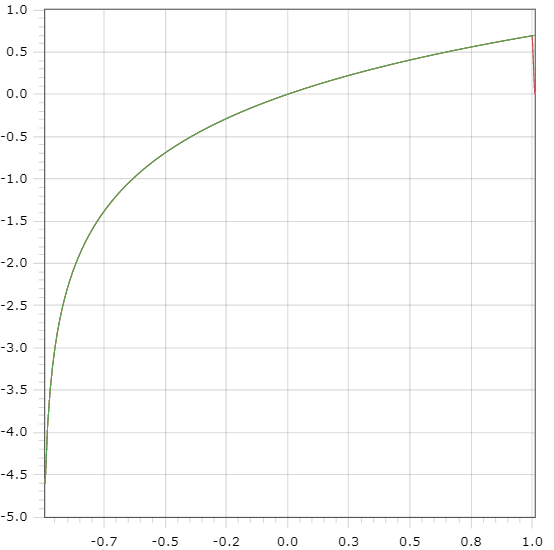
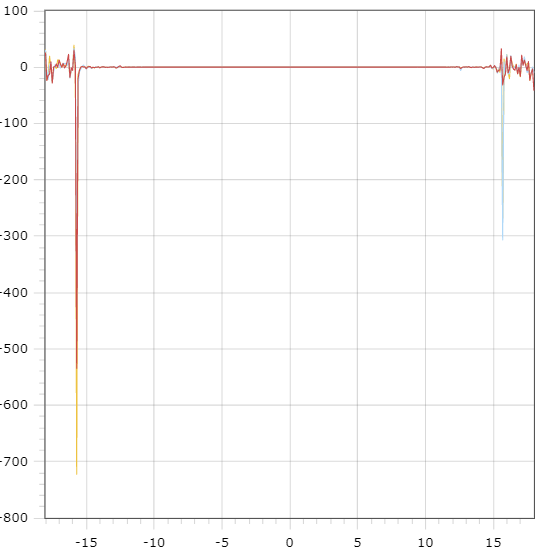
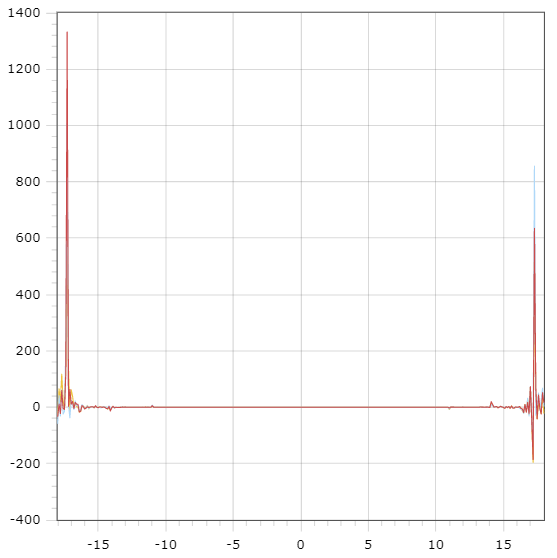
График показанный выше показывает результат вычисления трех методов и настоящий ответ для функции ln(1+x). Содержит вычисления от -0,99 до 1 с шагом 0.01. Желтая - линия соответствует прямому методу, синяя — обратному, красная — попарному, зеленая настоящему.(Налезают друг на друга в данном порядке).

График ошибок функции синуса

График показанный выше показывает результат вычисления трех методов и настоящий ответ для функции синуса. Содержит вычисления от 0 до 18 с шагом 0.1. Желтая - линия соответствует прямому методу, синяя — обратному, красная — попарному.(Налезают друг на друга в данном порядке).

Можно заметить что на данном интервале самым точным является обратный метод.

График ошибок функции косинуса

График показанный выше показывает результат вычисления трех методов и настоящий ответ для функции косинуса. Содержит вычисления от -18 до 18 с шагом 0.1. Желтая - линия соответствует прямому методу, синяя — обратному, красная — попарному.(Налезают друг на друга в данном порядке).

Можно заметить что на данном интервале самым точным является прямой метод.

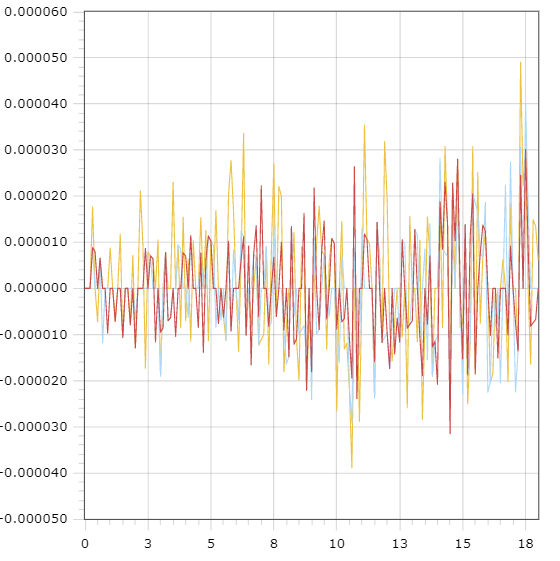
График ошибок функции експоненты

График показанный выше показывает результат вычисления трех методов и настоящий ответ для функции експоненты. Содержит вычисления от 0 до 18 с шагом 0.1. Желтая - линия соответствует прямому методу, синяя — обратному, красная — попарному.(Налезают друг на друга в данном порядке).

Наглядно видно что в общем случае попарное суммирование самое точное.

График ошибок функции ln(1+x)

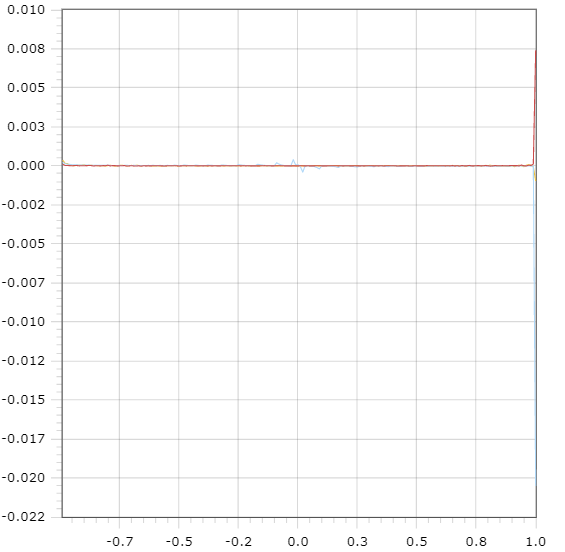


График показанный выше показывает результат вычисления трех методов и настоящий ответ для функции ln(1+x). Содержит вычисления от -0,99 до 1 с шагом 0.01. Желтая - линия соответствует прямому методу, синяя — обратному, красная — попарному.(Налезают друг на друга в данном порядке).

Рекурсия отстает по точности при числе приближенному к 1.

# Заключение

В результате проведенной лабораторной работы было реализовано 3 метода для подсчета 4ех функций на языке СИ . Была описана программная реализация. Была подтверждена корректность работы функций. И были проведены эксперименты выявляющие абсолютную ошибку методов.