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.

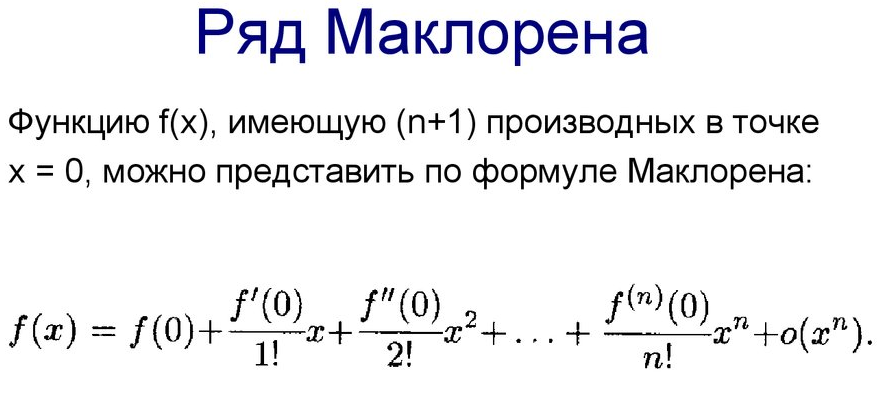
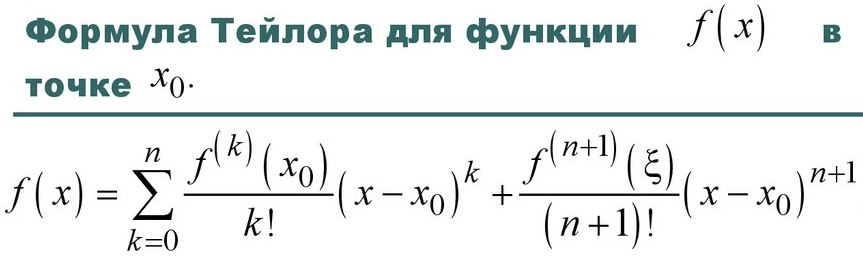
3) Подтвердить корректность данной программы (выявить сходимость функций).

4)Показать результаты экспериментов на графике или в таблицах.

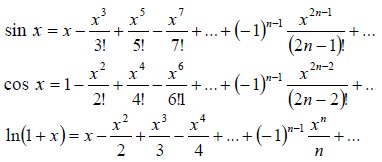
5)Сделать вывод.

# 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 возращать следующий идущий элемент в зависимости от предыдущего. К счастью формула Тейлора позволяет нам это сделать. Реализуем два способа подсчета элементов, первый просто считает следующий элемент (функция необходимая для первого метода подсчета прямой суммы), второй считает следующий элемент в рекурсии ( в отличие от первого метода, второй подсчет через рекурсию или обратную сумму включает в себя способ подсчета элементов). Для третьего метода — попарного подсчета будем использовать элементы первого способа подсчета.

Формулы функций

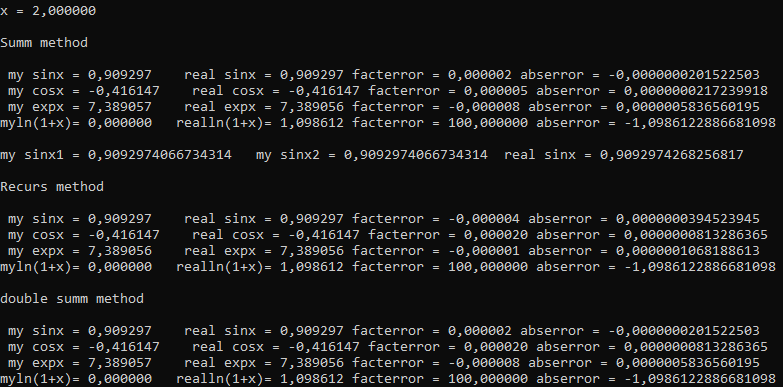


# Руководство пользователя

При запуске компилятора в консоли отобразится следующее:



Программа будет ждать пока вы не введете число соответствующую x которое вы хотите проверить, т.е. если вы хотите например проверить число 2.После того как вы введете число появится следующее:

На экране вывелись результаты подсчета функций тремя методами сначала прямой суммой потом обратной потом попарной вместе с реальными значениями функций и ошибками. Можно заметить что функция ln(1+x) не сработала корректно так как формула Тейлора для данной функции работает лишь на интервале (-1;1], а так как 2 не лежит в этом интервале, то функция вернула 0.

# Описание программной реализации

Int Main() - помимо инициализации и вывода на консоль, здесь осущевстляется подсчет всех функций методом прямой суммы и попарной через while до 50 000 элементов. Помимо этого есть возможность вычитания периода из x с сохранением верного результата(закоментирована).

double next\_sin1(double pred, double x, int i) — первый способ подсчета следующего элемента у функции синуса. Просто возвращает следующий элемент. На вход принимает предыдущий элемент, число x и номер элемента(начиная с нулевого).

double next\_sin2(double pred, double x, int i) — второй способ подсчета следующего элемента у функции синуса. Возвращает следующий элемент, но в отличие от первого способа используется другая комбинация тех же преобразований что и в первом. На вход принимает предыдущий элемент, число x и номер элемента(начиная с нулевого).

double next\_lnx(double pred, double x, int i) — способ подсчета следующего элемента у функции ln(1+x). Просто возвращает следующий элемент. На вход принимает предыдущий элемент, число x и номер элемента(начиная с нулевого).

double next\_cos(double pred, double x, int i) — способ подсчета следующего элемента у функции косинуса. Просто возвращает следующий элемент. На вход принимает предыдущий элемент, число x и номер элемента(начиная с нулевого).

double next\_exp(double pred, double x, int i) — способ подсчета следующего элемента у функции експоненты. Просто возвращает следующий элемент. На вход принимает предыдущий элемент, число x и номер элемента(начиная с нулевого).

recsin(double x, double xc, int i) — метод обратной суммы для функции синуса. Возвращает значение функции синуса от x.На вход принимает предыдущий элемент, число x и номер элемента(начиная с нулевого).

reccos(double x, double xc, int i)— метод обратной суммы для функции косинуса. Возвращает значение функции косинуса от x. На вход принимает предыдущий элемент, число x и номер элемента(начиная с нулевого).

recexp(double x, double xc, int i)— метод обратной суммы для функции експоненты. Возвращает значение функции експоненты от x. На вход принимает предыдущий элемент, число x и номер элемента(начиная с нулевого).

recln(double x, double xc, int i)— метод обратной суммы для функции ln(1+x). Возвращает значение функции ln(1+x). На вход принимает предыдущий элемент, число x и номер элемента(начиная с нулевого).

# Подтверждение корректности

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

# Результаты экспериментов

График функции синуса

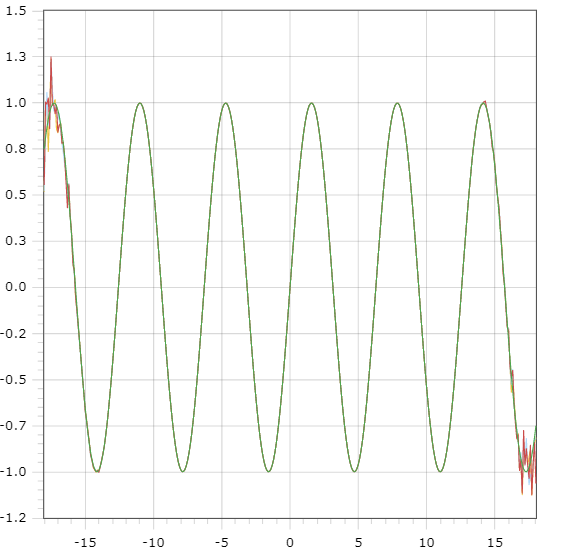


График показанный выше показывает результат вычисления трех методов и настоящий ответ для функции синуса. Содержит вычисления от -18 до 18 с шагом 0.01. Желтая - линия соответствует прямому методу, синяя — обратному, красная — попарному, зеленая настоящему.(Налезают друг на друга в данном порядке).

График функции косинуса

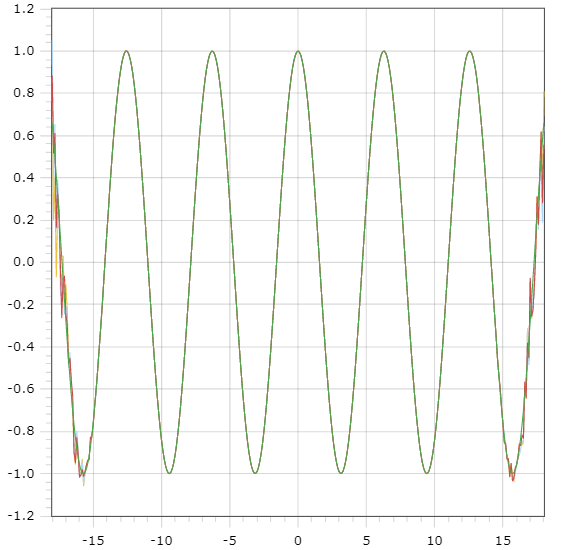


График показанный выше показывает результат вычисления трех методов и настоящий ответ для функции косинуса. Содержит вычисления от -18 до 18 с шагом 0.01. Желтая - линия соответствует прямому методу, синяя — обратному, красная — попарному, зеленая настоящему.(Налезают друг на друга в данном порядке).

График функции експоненты

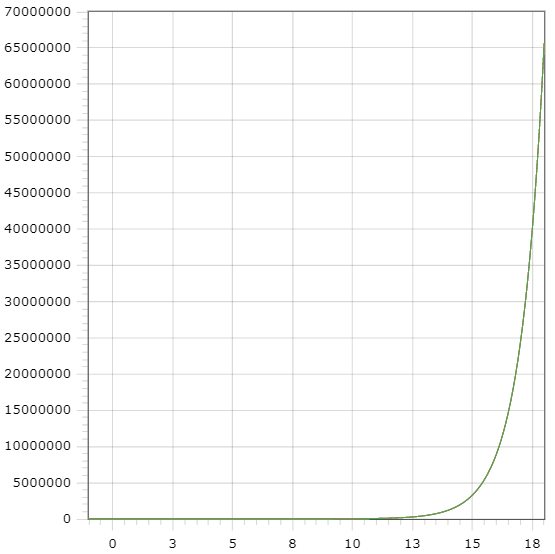


График показанный выше показывает результат вычисления трех методов и настоящий ответ для функции експоненты. Содержит вычисления от 0 до 18 с шагом 0.1. Желтая - линия соответствует прямому методу, синяя — обратному, красная — попарному, зеленая настоящему.(Налезают друг на друга в данном порядке).

График функции ln(1+x)

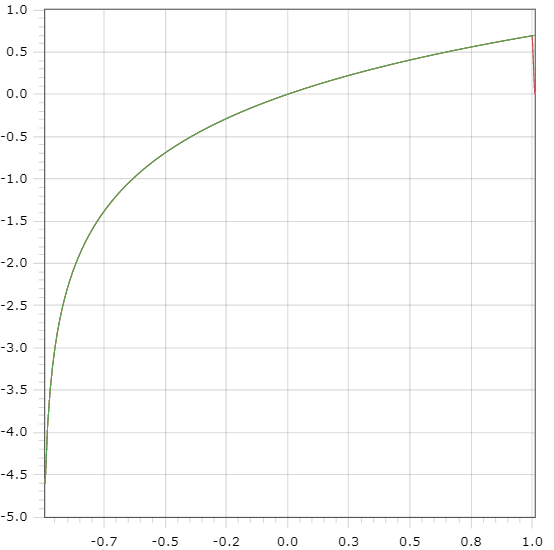


График показанный выше показывает результат вычисления трех методов и настоящий ответ для функции ln(1+x). Содержит вычисления от -0,99 до 1 с шагом 0.01. Желтая - линия соответствует прямому методу, синяя — обратному, красная — попарному, зеленая настоящему.(Налезают друг на друга в данном порядке).

График ошибок функции синуса

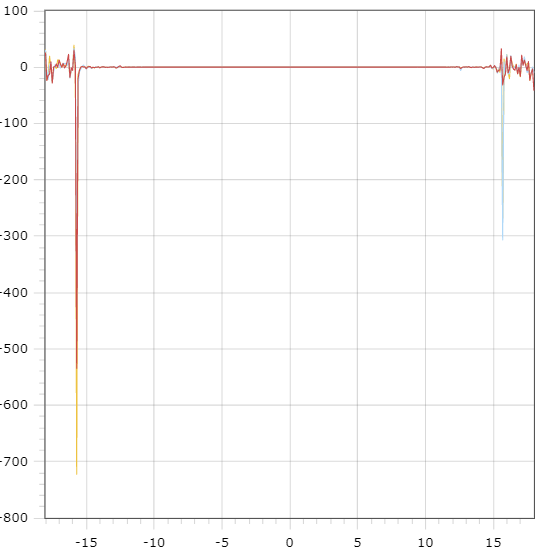


График показанный выше показывает результат вычисления трех методов и настоящий ответ для функции синуса. Содержит вычисления от 0 до 18 с шагом 0.1. Желтая - линия соответствует прямому методу, синяя — обратному, красная — попарному.(Налезают друг на друга в данном порядке).

Можно заметить что на данном интервале самым точным является обратный метод. Интервалы смены точности методов от худщего к лучшему лежит в отдельном txt точность синуса.

График ошибок функции косинуса

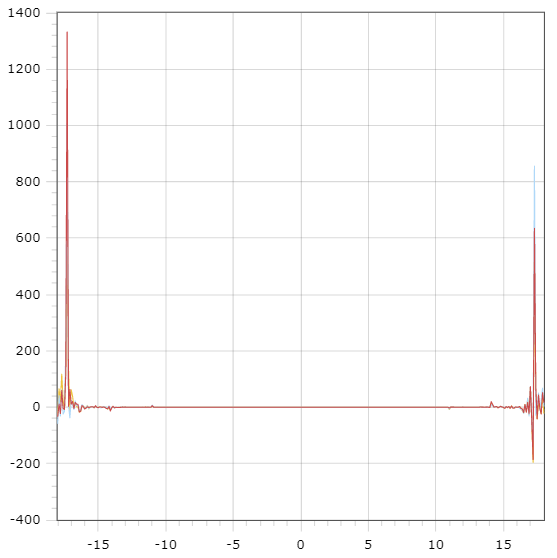


График показанный выше показывает результат вычисления трех методов и настоящий ответ для функции косинуса. Содержит вычисления от -18 до 18 с шагом 0.1. Желтая - линия соответствует прямому методу, синяя — обратному, красная — попарному.(Налезают друг на друга в данном порядке).

Можно заметить что на данном интервале самым точным является прямой метод. Интервалы смены точности методов от худщего к лучшему лежит в отдельном txt точность косинуса.

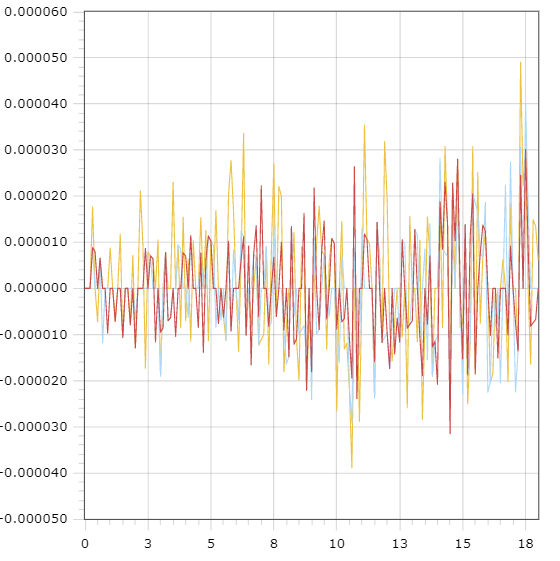
График ошибок функции експоненты

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. Интервалы смены точности методов от худщего к лучшему лежит в отдельном txt точность експоненты.

График ошибок функции ln(1+x)

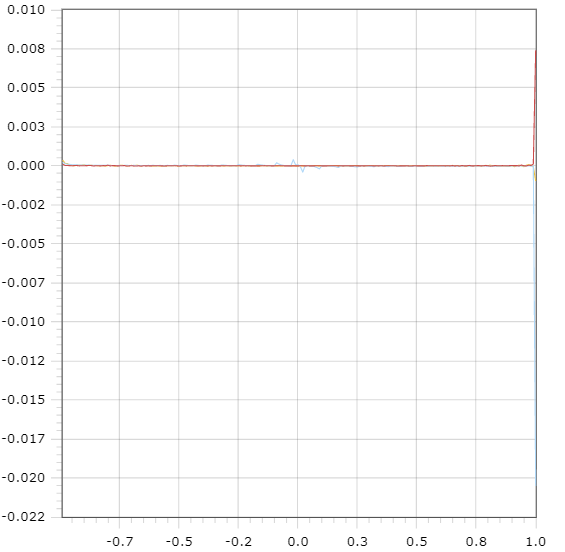


График показанный выше показывает результат вычисления трех методов и настоящий ответ для функции ln(1+x). Содержит вычисления от -0,99 до 1 с шагом 0.01. Желтая - линия соответствует прямому методу, синяя — обратному, красная — попарному.(Налезают друг на друга в данном порядке).

Рекурсия отстает по точности при числе приближенному к 1. Интервалы смены точности методов от худщего к лучшему лежит в отдельном txt точность ln(1+x).

# 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.