Лабораторная работа №6

Дисциплина: Научное программирование

Аветисян Давид Артурович

Содержание

# 1 Цель работы

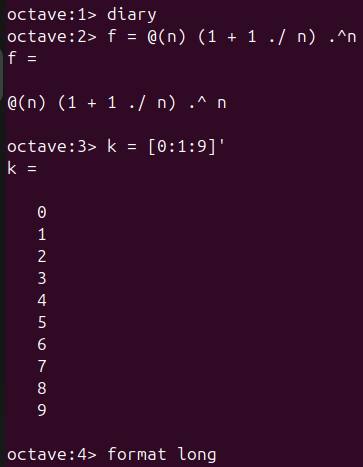
Познакомиться со сложными алгоритмами в Octave, которые были встроены для работы с пределами, последовательностями и рядами.

# 2 Задание

1. Познакомиться с пределами.
2. Познакомиться с частичными суммами.
3. Познакомиться с суммой ряда.
4. Познакомиться с вычислением интегралов.
5. Познакомиться с аппроксимированием суммами.

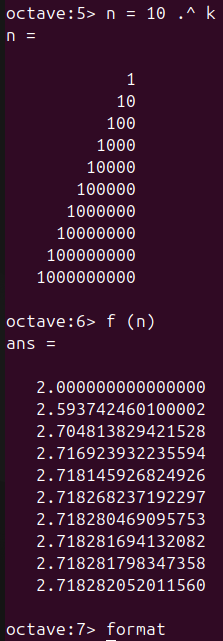
# 3 Выполнение лабораторной работы

1. Первым делом я научился работать с пределами. Для это я рассмотрел предел функции при . Сначала я определил данную функцию и создал индексную переменую, состоящую из целых чисел от до . Затем я задал длинный формат чисел.



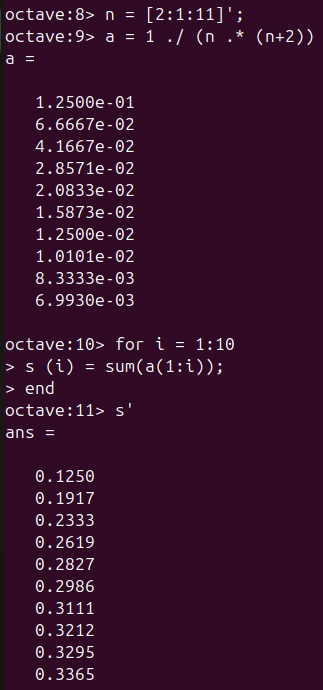
Определение функции и индексной переменной

После этого я взял степени числа , которые стали удобным входным значением и оценил . Предел сходится к конечному значению, которое приблизительно равно .



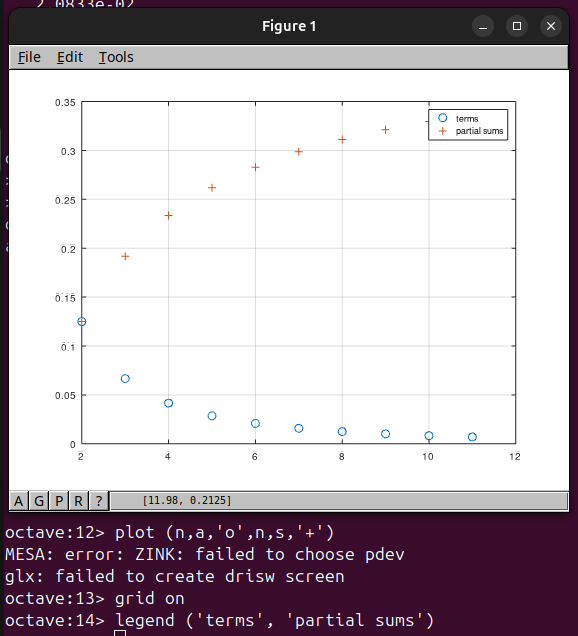
Оценка предела

1. Далее я познакомился с частичными суммами. Для этого я рассмотрел сумму ряда от при от до , где . Сначала я определил индексный вектор от до . Затем я вычислил члены ряда . После чего при помощи цикла я посчитал частичную сумму для каждого ряда от от первого слагаемого до -го слагаемого.



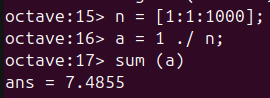
Определение частичной суммы ряда

И в конце я построил слагаемые и частичные суммы для .



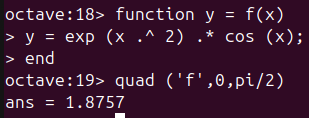
Построение слагаемых и частичных сумм для

1. После я познакомился с нахождением суммы первых членов гармонического ряда . Для этого я сгенерировал члены как вектор ряда, а затем взял их сумму.



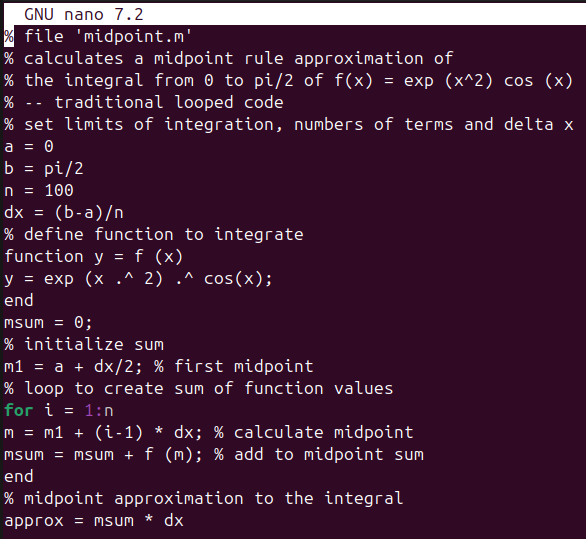
Нахождение суммы первых членов гармонического ряда

1. Потом я познакомился с вычислением интегралов. Для этого я взял интеграл от до функции . Для вычисления я определил функцию и использовал команду *quad*.

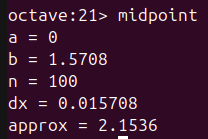


Вычисление интеграла

1. В конце лабораторой работы я научился аппроксимировать суммы. Сначала я написал скрипт чтобы вычислить предыдущий интеграл по правилу средней точки для и запустил его.

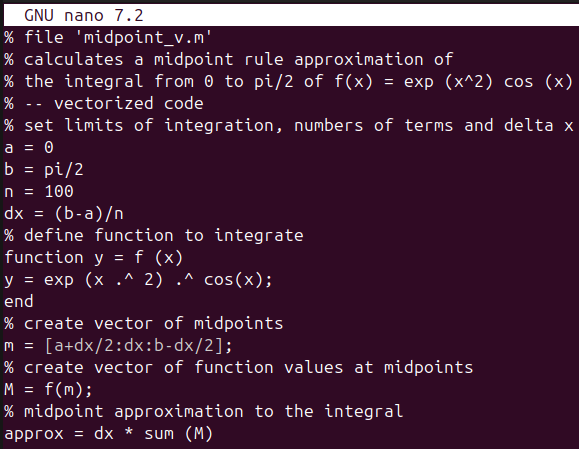


Скрипт midpoint.m

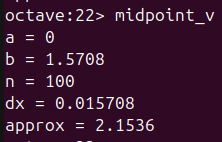


Работа скрипта midpoint.m

Далее я написал новый скрипт, но использовал векторизованный код, который не требует каких-либо циклов. После чего я его запустил.

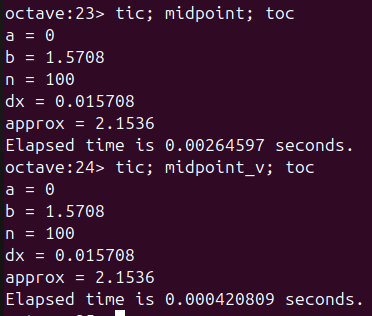


Скрипт midpoint\_v.m



Работа скрипта midpoint\_v.m

В конце я сравнил результаты и время выполнения каждого скрипта. Результаты были одинаковыми, но традиционный код выполнял почти в раз медленнее, чем векторизованный код.



Сравнение результатов и времени выполнения

# 4 Выводы

Я познакомился со сложными алгоритмами в Octave, которые были встроены для работы с пределами, последовательностями и рядами.