### РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

### Факультет физико-математических и естественных наук

### Кафедра прикладной информатики и теории вероятностей

### *дисциплина*: Научное программирование

# Лабораторная работа №6

### Выполнил: Маслов Александр

### Группа: НФИмд-02-20

### С/б: 1032202156

#### Москва

#### 2020

### Цель работы:

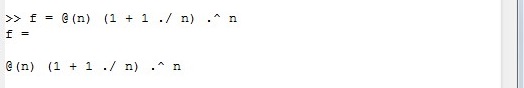
Рассмотреть с помощью Octave пределы, последовательности и ряды, а также два метода оценки функции: с помощью циклов и с помощью вектора входных значений и сравнить их.

### Ход работы:

### Пределы, последовательности и ряды

Рассмотрим предел:

Оценим это выражение. Нужно определить функцию. Есть несколько способов сделать это. Метод, который мы здесь используем, называется анонимной функцией. Это хороший способ быстро определить простую функцию.



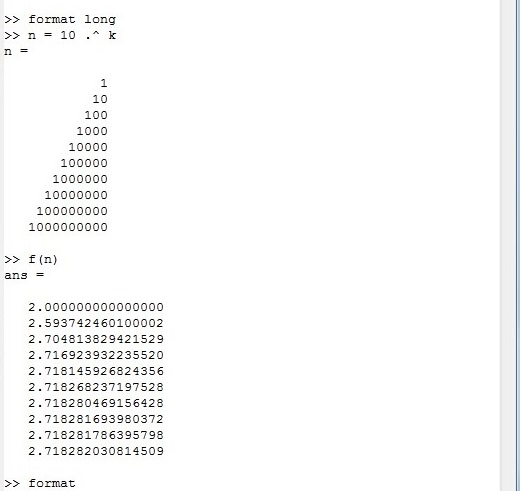
[Рисунок 1. Определяем функцию f](screen1.jpg)

Обратите внимание на использование поэлементных операций. Мы назвали функцию f . Входная переменная обозначается знаком , за которым следует переменная в скобках. Следующее выражение будет использоваться при оценке функции. Теперь f можно использовать как любую функцию в Octave. Далее мы создаём индексную переменную, состоящую из целых чисел от 0 до 9:



[Рисунок 2. Индексная переменная k](screen2.jpg)

Синтаксис [0:1:9] создает вектор строки, который начинается с 0 и увеличивается с шагом от 1 до 9. Использовали операцию транспонирования. Теперь мы возьмём степени 10, которые будут входными значениями, а затем оценим f(n).



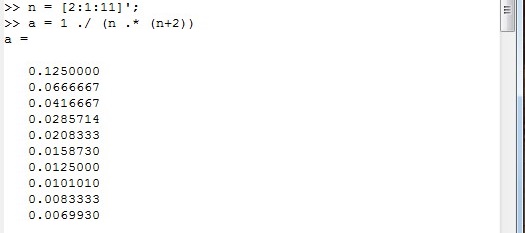
[Рисунок 3. Степени 10](screen3.jpg)

Предел сходится к конечному значению, которое составляет при¬ близительно 2,71828

**Частичные суммы**

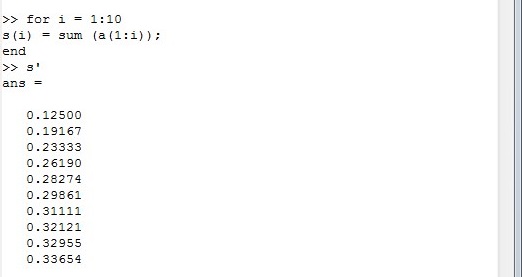
Пусть - ряд, n-й член равен

Для этого мы определим индексный вектор n от 2 до 11, а затем вычислим члены.



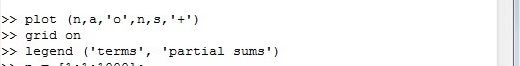
[Рисунок 4. Индексный вектор n и члены a](screen4.jpg)

Если мы хотим знать частичную сумму, нам нужно только написать sum(а).Если мы хотим получить последовательность частичных сумм, нам нужно использовать цикл. Мы будем использовать цикл for с индексом i от 1 до 10. Для каждого i мы получим частичную сумму последовательности от первого слагаемого до -го слагаемого. На выходе получается 10-элементный вектор этих частичных сумм.

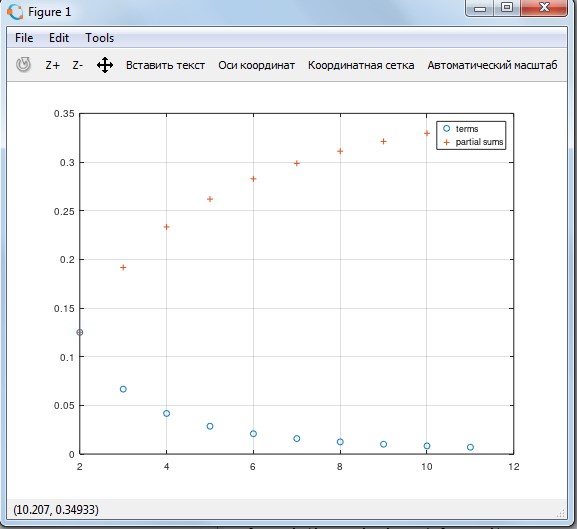


[Рисунок 5. Вектор частичных сумм](screen5.jpg)

Наконец, мы построим слагаемые и частичные суммы для 2 < n < 11.



[Рисунок 6. Построение графика](screen6.jpg)

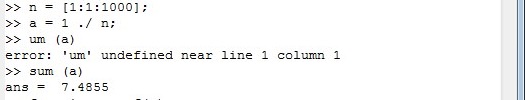


[Рисунок 7. График](graph1.jpg)

**Сумма ряда**

Найдём сумму первых 1000 членов гармонического ряда:

Нам нужно только сгенерировать члены как ряда вектор, а затем взять их сумму.



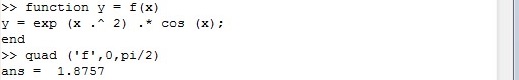
[Рисунок 8. Сумма ряда](screen7.jpg)

### Численное интегрирование

**Вычисление интегралов**

Octave имеет несколько встроенных функций для вычисления определённых интегралов. Мы будем использовать команду quad (сокращение от слова квадратура). Вычислим интеграл:

Синтаксис команды - quad( ’ f ’ , а , b). Нам нужно сначала определить функцию.



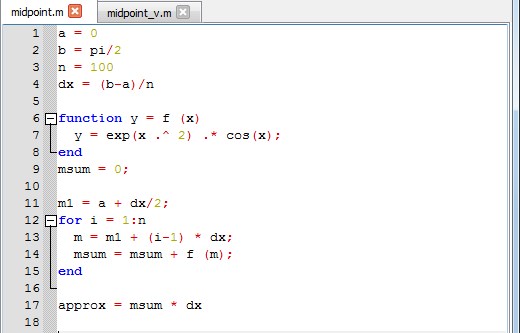
[Рисунок 9. Вычисление интеграла](screen8.jpg)

Обратите внимание, что функция exp (х) используется для .Мы использовали конструкцию function . . . end.

**Аппроксимирование суммами**

Правило средней точки, правило трапеции и правило Симпсона являются общими алгоритмами, используемыми для численного интегрирования. Напишем скрипт,чтобы вычислить интеграл

по правилу средней точки для n = 100. Стратегия заключается в использовании цикла,который добавляет значение функции к промежуточной сумме с каждой итерацией.В конце сумма умножается на . Введём код в текстовом файле и назовём его midpoint.m.



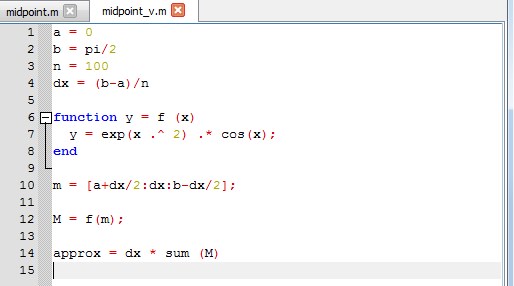
[Рисунок 10. Код файла mibpoint.m](screen9.jpg)

Ондолжен бытьпомещён вваш рабочий каталог,азатем его можно запустить,набрав midpoint в командной строке.



[Рисунок 11. Запуск файла mibpoint.m](screen10.jpg)

Традиционный код работает хорошо,но поскольку Octave является векторным языком,также можно писать векторизованный код, который не требует каких-либо циклов. Создадим вектор x-координат средних точек. Затем мы оцениваем f по этому вектору средней точки,чтобы получить вектор значений функции.Аппроксимация средней точки - это сумма компонент вектора,умноженная на .



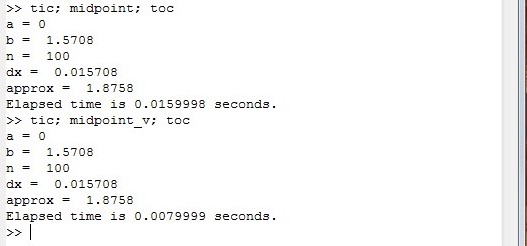
[Рисунок 12. Код файла mibpoint\_v.m](screen11.jpg)

Запустим его:



[Рисунок 13. Запуск файла mibpoint\_v.m](screen12.jpg)

Сравним результаты и время выполнения для каждой реализации.



[Рисунок 13. Результаты и время выполнения](screen13.jpg)

Результаты обоих методов получились одинаковыми, но метод оценки функции с помощью цикла выполняется намного медленнее.

### Вывод:

В процессе выполнения лабораторной работы с помощью Octave мы рассмотрели пределы, последовательности и ряды, а также два метода оценки функции: с помощью циклов и с помощью вектора входных значений и сравнили их.