Научное програмирование

Лабораторная работа № 6

Кейела Патачона, НПМмд-02-21

Содержание

[Цель работы 1](#_Toc90685508)

[Пределы, последовательности и ряды 1](#_Toc90685509)

[Пределы 1](#_Toc90685510)

[Частичные суммы 3](#_Toc90685511)

[Сумма ряда 5](#_Toc90685512)

[Численное интегрирование 6](#_Toc90685513)

[Встроенная функция 6](#_Toc90685514)

[Правило средней точки 6](#_Toc90685515)

[Сравнение методов 8](#_Toc90685516)

[Вывод 9](#_Toc90685517)

[Список литературы 9](#_Toc90685518)

# Цель работы

Вычисление пределов, сумм рядов и интегралов

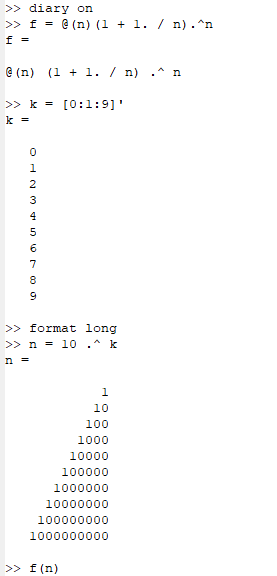
# Пределы, последовательности и ряды

Octave - полноценный язык программирования, поддерживающий множество типов циклов и условных операторов. Однако, поскольку то векторный язык, многие вещи, которые можно было бы сделать с помощью циклов, можно векторизовать. Под векторизованным кодом мы понимаем следующее: вместо того, чтобы писать цикл для многократной оценки функции, мы сгенерируем вектор входных значений, а затем оценим функцию с использованием векторного ввода. В результате получается код, который легче читать и понимать, и он выполняется быстрее благодаря эффективным алгоритмам для матричных операций.

## Пределы

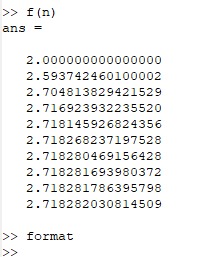
Рассмотрим предел:

Мы оценим это выражение. Сначала определим функцию. Есть несколько способов сделать это. Метод, который мы здесь используем, называется анонимной функцией. Это хороший способ быстро определить простую функцию.



Нахождение предела 1

Обращаем внимание на использование поэлементных операций. Мы назвали функцию . Входная переменная обозначается знаком @, за которым следует переменная в скобках. Следующее выражение будет использоваться при оценке функции. Теперь можно использовать как любую функцию в Octave. Далее мы создали индексную переменную, состоящую из целых чисел от 0 до 9 и взяли степени 10, которые будут входными значениями, а затем оценили .



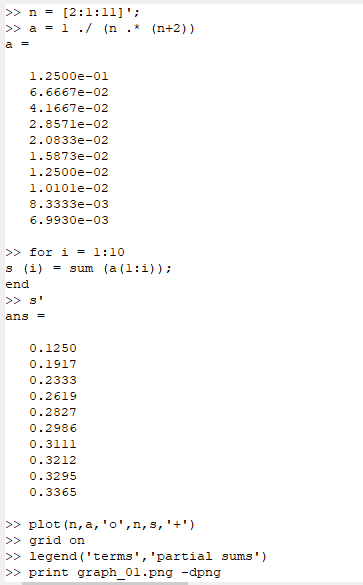
Нахождение предела 2

Предел сходится к конечному значению, которое составляет приблизительно Подобные методы могут быть использованы для численного исследования последовательностей и рядов.

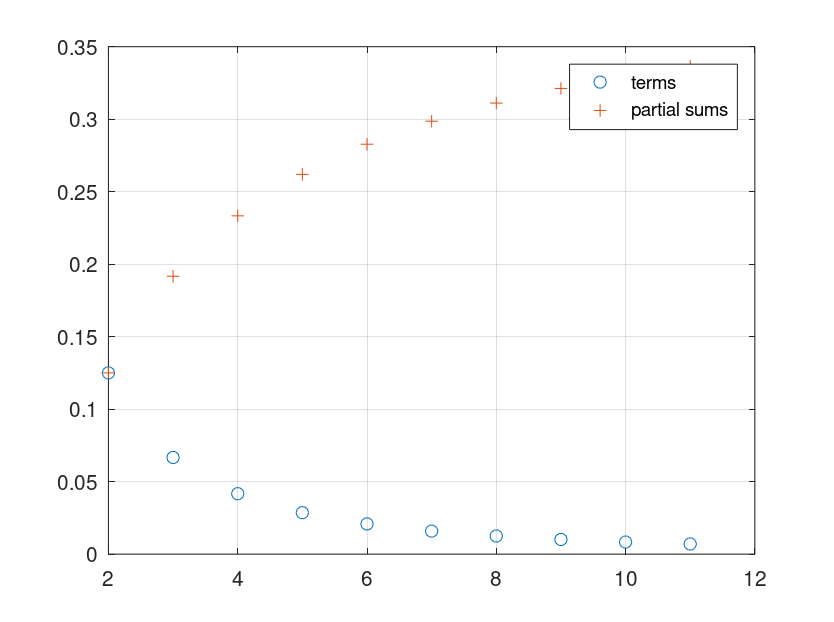
## Частичные суммы

Пусть ряд, й член равен

Для этого мы определим индексный вектор пот 2 до 11, а затем вычислим члены. Если мы хотим знать частичную сумму, нам нужно только написать . Если мы хотим получить последовательность частичных сумм, нам нужно использовать цикл. Мы будем использовать цикл for c индексом і от 1 до 10. Для каждого і мы получим частичную сумму последовательности а от первого слагаемого до і-го слагаемого. На выходе получается 10-элементный вектор этих частичных сумм.



Частичные суммы

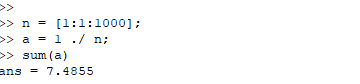


Частичные суммы - график

## Сумма ряда

Найдём сумму первых 1000 членов гармонического ряда:

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



Сумма ряда

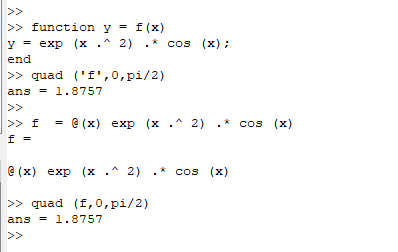
# Численное интегрирование

## Встроенная функция

Octavе имеет несколько встроенных функций для вычисления определённых интегралов. Мы будем использовать команду (сокращение от слова квадратура).

Вычислим интеграл:

Синтаксис команды — . Нам нужно сначала определить функцию.



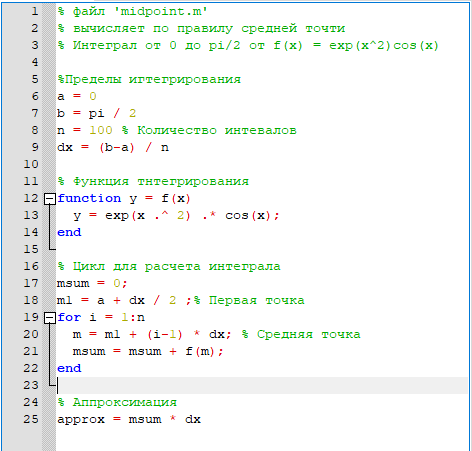
Определенный интеграл

Обращаем внимание, что функция используется для . Мы использовали конструкцию . Кавычки вокруг имени f не используются, если используется анонимная функция.

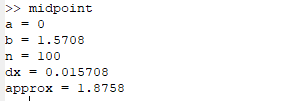
## Правило средней точки

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 . Стратегия заключается в использовании цикла, который добавляет значение функции к промежуточной сумме с каждой итерацией. В конце сумма умножается на

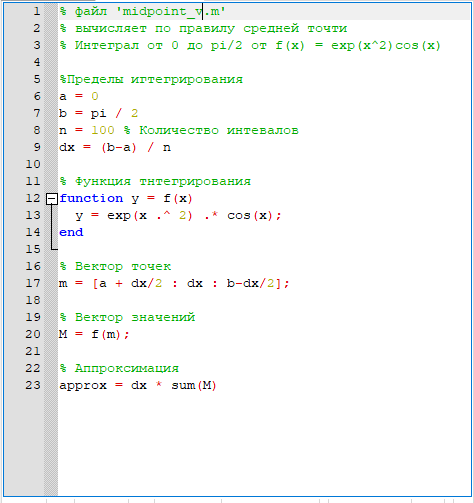


Метод средней точки - код

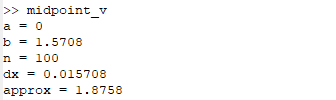


Метод средней точки - результат

Традиционный код работает хорошо, но поскольку Octave является векторным языком, также можно писать векторизованный код, который не требует каких-либо циклов. Создадим вектор r-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 .



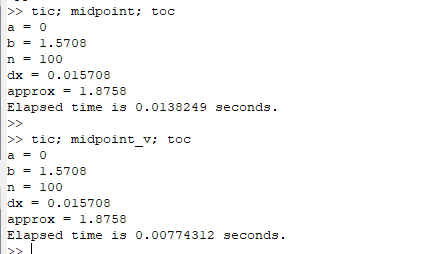
Векторный метод средней точки



Векторный результат

## Сравнение методов

Сравнили время выполнения для каждой реализации, и векторная реализация работает быстрее.



Сравнение реализаций

# Вывод

В ходе выполнения данной работы мы ознакомились с вычислением пределов, с работой с последавательностями и с рядами и научились посчитать определенные интеграли с помощью различных методов на языке Octave.

# Список литературы

1. [Инструкция к лабораторной работе №6](https://esystem.rudn.ru/mod/resource/view.php?id=795800)