

Программа повышения конкурентоспособности

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«Уральский федеральный университет имени

первого Президента России Б. Н. Ельцина»

ОСНОВНЫЕ СРЕДСТВА МАТЛАВ ДЛЯ РАБОТЫ С ВРЕМЕННЫМИ РЯДАМИ

**Методические указания к выполнению
лабораторной работы № 1**

Екатеринбург

2017

Содержание

Цель изучения материала	3
Перечень компетенций, формирующихся или получающих приращение в результате прослушивания лекции	3
Список литературы	4
1. Введение.....	5
2. Краткие основы работы в среде MATLAB	5
3. Арифметика с массивами	8
4. Задание на лабораторную работу	11
5. Требования к оформлению отчета.....	13

Цель изучения материала

Целью данной лабораторной работы является ознакомление с программным пакетом MATLAB, изучение базовых средств работы с массивами и временными рядами, основные функции расчета характеристик временных рядов и средства их визуализации.

Перечень компетенций, формирующихся или получающих приращение в результате прослушивания лекции

Способность анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями.

Умение проводить разработку и исследование теоретических и экспериментальных моделей объектов профессиональной деятельности в областях науки, техники.

Способность осуществлять сбор, анализ научно-технической информации, отечественного и зарубежного опыта по тематике исследования.

Способность к самостоятельной разработке методики исследования.

Список литературы

1. Методы и алгоритмы анализа данных и их моделирование в MATLAB. Учебное пособие / А. Сирота. – БХВ-Петербург. — 2016. — 384 с.
2. Введение в систему MATLAB. Методическое пособие. / Водолазская И.В. . — Астрахань, АГТУ. — 2004. — 48 с.
<http://mathmod.asu.edu.ru/images/File/ebooks/matlab.pdf>
3. Поршнев С. В. Компьютерное моделирование физических процессов в пакете MATLAB. — М.: Горячая линия. — Телеком. — 2003. — 592 с.
4. Садовникова Н.А., Шмойлова Р.А. Анализ временных рядов и прогнозирование. — М.: «Футурис», 2009. — 261 с.
5. George A Duckett. Time Series Analysis: Questions and Answers. — CreateSpace Independent Publishing Platform. — 2016. — 258 с. ISBN 978-1-533-57098-7.

1. Введение

Программный математический комплекс MATLAB является многоцелевым инструментом, который, на сегодняшний день, включает в себя огромное количество инструментов и приложений, посвященных многим актуальным проблемам. Изначально основой этой системы было составление алгоритмов решения линейных уравнений и задач, что требовало создание быстрых и высокоэффективных средств анализа, преобразования и арифметики с матрицами, отсюда и название, которое переводится как «матричная лаборатория». Эта высокая эффективность оказывается весьма актуальной для анализа временных рядов, так как многие методы их анализа оперируют линейными уравнениями, матрицами и другими средствами, которые легко программировать в среде MATLAB.

При работе с математическим пакетом MATLAB предполагается, что у студентов уже есть базовые навыки в области алгоритмизации процессов, языков программирования, а также знания математического анализа, понятия векторов, матриц.

2. Краткие основы работы в среде MATLAB

Среда MATLAB состоит из командного окна (в центре рис. 1), которое появляется при запуске программы, в котором можно выполнять различные интерпретируемые команды на языке высокого уровня, подобного языку C или Java. Все команды, которые вызывались в ходе работы, сохраняются в окне истории команд (*Command History*, справа внизу). Все переменные, объекты, массивы и вектора, которые используются по ходу работы, сохраняются в окне *Workspace* (справа вверху). Слева находится браузер или проводник, позволяющий загружать файлы, читать их, редактировать, сохранять и т.д.

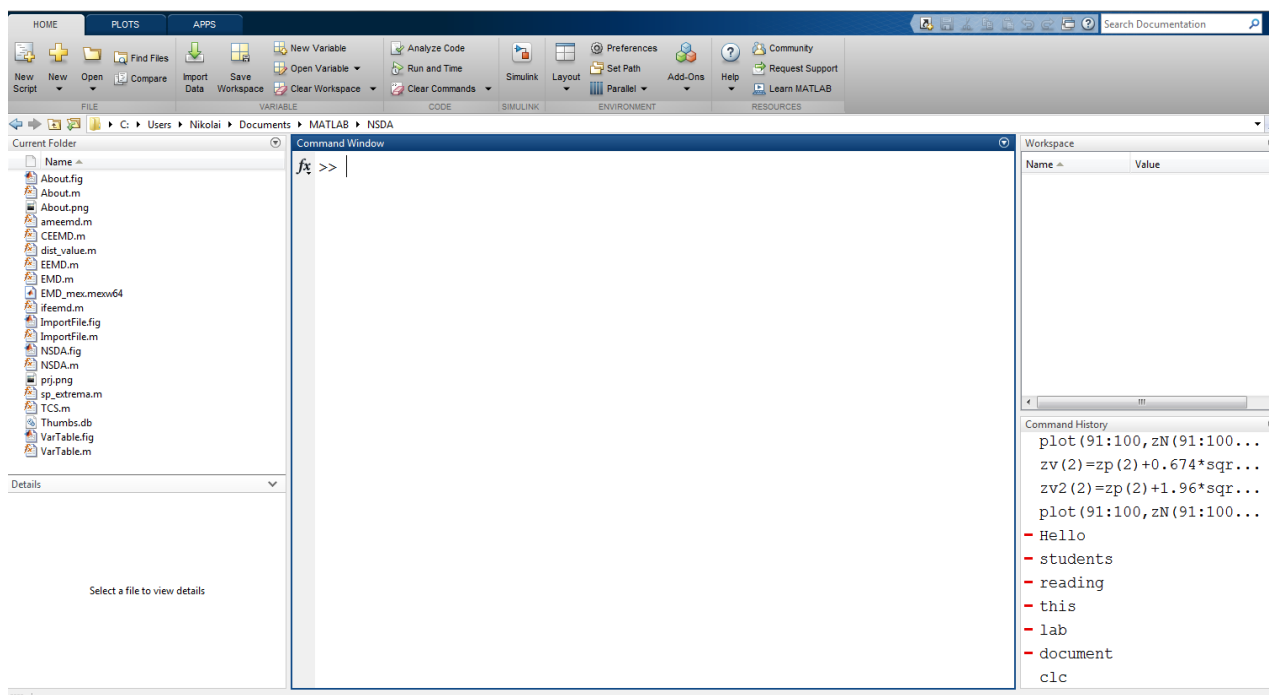


Рисунок 1. – Среда MATLAB

В командном окне в режиме диалога проводятся вычисления. Пользователь вводит команды или запускает на выполнение файлы с текстами на языке MATLAB. Интерпретатор обрабатывает введенное выражение и выдает результаты: числовые и строковые данные, предупреждения и сообщения об ошибках. Строка ввода помечена знаком «>>». В командном окне показываются вводимые с клавиатуры числа, переменные, а также результаты вычислений. Имена переменных должны начинаться с буквы. Знак «=» соответствует операции присваивания. Нажатие клавиши *Enter* заставляет систему вычислить выражение и показать результат. Например, если набрать $a=23+35$, то после нажатия клавиши *Enter* будет получен результат, представленный на рисунке 2.

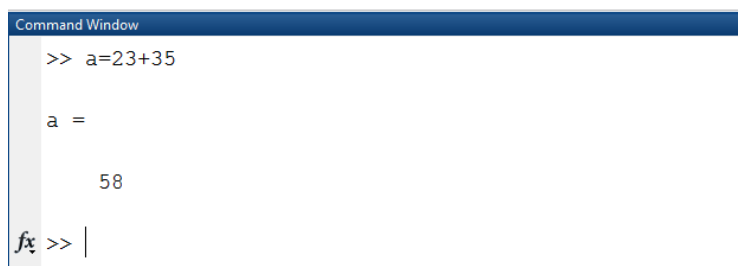


Рисунок 2. Пример расчета выражения.

Если командное окно нужно очистить, то можно использовать команду ***clc***, которая очищает экран. Стандарты записи чисел, операторы условий сравнения оказываются теми же самыми, что и для Си-подобных языков программирования, за исключением операции «**Не равно**», которое в MATLAB оформляется как «**~=**». Кроме того, важным отличием является то, что здесь не требуется указывать тип переменной, так как ее тип интерпретируется в зависимости от ее формы записи. Тем не менее, здесь есть все те же ключевые слова, когда тип переменной хочется указать самостоятельно.

Больше всего нас будет интересовать работа с массивами, так как на их основе и строятся временные ряды, представляющие собой упорядоченные вектора некоторых отсчетов. Массивы в среде MATLAB создаются с помощью оператора квадратных скобок []. Элементы массива помещаются между скобками, отделенные пробелом или запятой. Например, выражение **a=[1 2 3]** создаст **массив-строку** из трех элементов. Если же надо создать **массив-столбец**, то запись будет в виде **a=[1; 2; 3]**. Если хочется сразу преобразовать строку в столбец, то есть **транспонировать** ее, то есть оператор транспонирования матриц и векторов «**'**», поэтому данное выражение **[1; 2; 3]' == [1 2 3]** будет истинным.

Если есть массивы, то можно обращаться к их элементам. **Нумерация в массивах идет с единицы!** То есть первый элемент есть **a(1)**. Обращение к элементам массива происходит через круглые скобочки. При этом массивы можно как угодно менять, дополнять и изменять прямо по номеру их элемента. Если хочется найти **длину массива**, то есть функция **length()**, которая ее для Вас найдет. Если массив многомерный, то лучше использовать функцию **size(A, 1)**, которая вторым аргументом принимает номер размерности, по которой надо вернуть размер массива. Либо, если просто передать ей массив, то она вернет 2 и более значений.

Отсюда можно понять, что MATLAB полностью поддерживает ООП парадигму. То есть можно использовать и писать функции с переменным числом аргументов, делать наследование классов и т.д. Но все это нам пригодится в малой степени. Но вернемся к массивам.

Работа с многомерными массивами выглядит следующим образом: если ввести $a = [1 \ 2; 3 \ 4; 5 \ 6]$, то получится массив из 3 строчек и 2 столбцов. По другому тот же массив можно записать, как $a = [[1 \ 2]; [3 \ 4]; [5 \ 6]]$; Если надо обратиться к элементу такого массива, то используется нотация вида $a(1,1)$, $a(1,2)$, $a(2,1)$ и т.д., где идет сначала номер строки, затем номер столбца. Что будет если написать $a(3)$? Тогда MATLAB выведет число **5**, то есть нумерация **линеаризуется по столбцам**.

Если хочется инициализировать массив нужного размера, то используется функция **zeros(M,N)**, которой передается число строк и столбцов, а сама матрица будет полностью состоять из нулей. Другой способ инициализации массива – в виде операций **среза**. Это выражения вида **$A = a:i:b$** . Если написать $A = 0:1:5$, то такая запись создаст массив от **0 до 5 с шагом 1**: $[0 \ 1 \ 2 \ 3 \ 4 \ 5]$.

3. Арифметика с массивами

В традиционных языках программирования вычисления с массивами осуществляются поэлементно в том смысле, что нужно запрограммировать каждую отдельную операцию над отдельным элементом массива. В М-языке системы MATLAB допускаются мощные групповые операции над всем массивом сразу. Именно групповые операции системы MATLAB позволяют чрезвычайно компактно задавать выражения, при вычислении которых реально выполняется гигантский объем работы.

Операции сложения и вычитания обозначаются обычными «+» и «-». Они определяются для массивов однозначно. Но вот операции **умножения, деления и возведения в степень** для массивов воспринимаются в двух

видах: если операции работают, как матричные, то они пишутся, как просто знаки «*», «/», «^». Но если массивы перемножаются, делятся и возводятся в степень **поэлементно**, то к операциям впереди добавляется точка: «.*», «./», «.^». Например, $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$; $B = \begin{bmatrix} 0 & 1 \end{bmatrix}$; то умножение $A*B$ невыполнимо (размерности не совпадают), а вот $A.*B$ даст матрицу $\begin{bmatrix} 0 & 2 \\ 0 & 4 \\ 0 & 6 \end{bmatrix}$.

Почти все функции MATLAB написаны таким образом, что умеют работать, как со **скалярными** величинами, так и с **массивами**. Но нас больше будут интересовать функции, которые специально предназначены для массивов.

Кроме того, М-язык, будучи языком программирования, имеет средства ветвления программ, циклы и др. механизмы стандартных языков программирования. **Циклы** описываются в виде:

for i=start:step:end

внутренность цикла

end

или через

while (условие)

внутренность цикла

end

Операторы ветвления:

if (условие)

команды

else

команды

end

Итак, в среде MATLAB мы будем записывать временные ряды (ВР) в виде **векторов**, то есть одномерных матриц (строк или столбцов), за исключением редких случаев, где мы будем это оговаривать. Вот только чаще всего для ВР будет **два вектора** – один содержит **отсчеты уровня** ряда y , а другой ряд содержит **отсчеты времени** t , в которых эти отсчеты наблюдались. Тогда для построения такого ВР визуально достаточно будет вызвать функцию **plot (t, y)**.

Еще одним важным инструментом М-языка являются создание скриптов и м-функций. Скрипты представляют собой просто набор команд, который можно сохранить и затем вызвать для моментального выполнения друг за другом. Функции – это те же скрипты, но которые можно вызывать с разными параметрами, при этом внутренность функции будет скрыта, а результатом ее работы будут ее выходные аргументы. Обе эти вещи создаются с помощью меню в левом верхнем углу окна MATLAB-среды. Перейдем к самой работе, в которой станет понятней, как работать с ВР в MATLAB.

4. Задание на лабораторную работу

Результатом выполнения лабораторной работы является оформленный отчет, в котором должны быть представлены и отражены все нижеперечисленные пункты:

- 1) Создать ВР, являющийся выборкой случайной величины с нормальным распределением

x = randn(1, 10000);

где данная функция генерирует случайные величины из нормального распределения размером 1x10000.

Если нужно использовать равномерное распределение, то используется функция **rand(m, n)**.

- 2) Создать для него ряд временных отсчетов, на которых он будет определен: **t = linspace(3,5,10000);**

Где данная функция создает линейный равномерный массив из 10000 элементов в интервале от 3 до 5.

- 3) Построить ВР на временной сетке с помощью функции **plot(t, x)**.
- 4) В отчете привести получившееся изображение ВР и также написать, к какому виду ВР он относится по классификации рядов из **лекции 1**.
Указать, почему было принято такое решение о классификации.

- 5) Найти мат. ожидание данного ВР двумя способами:

С помощью функции **mean(x)**, которая считает среднее значение ряда;

С помощью **собственных расчетов** на основе формулы (2.7) из лекции 2.

Внимание! Нужно писать собственные расчеты над временными рядами в виде операций над массивами, то есть **не использовать циклы**.

Сравнить полученные результаты в отчете.

- б) Найти дисперсию (*variance*) данного ВР двумя способами:

С помощью функции **var(x)**, которая считает дисперсию ряда;

С помощью **собственных расчетов** на основе формулы (2.8) из лекции 2.

Сравнить полученные результаты в отчете.

7) Найти асимметрию ВР по формуле (2.9).

8) Найти эксцесс ВР по формуле (2.10).

9) Построить оценку выборочной автокорреляции ВР двумя способами и построить ее графически функцией **plot()**:

С помощью функции **autocorr(x)**;

С помощью **собственных расчетов** на основе формулы (2.17) до 20 лага.

Здесь разрешается использовать циклы для расчетов.

Сравнить полученные результаты в отчете. Пояснить, почему автокорреляционная функция ряда (нормального шума) имеет такой вид.

10) Написать **м-сценарий**, который бы выполнял все вышеперечисленные пункты.

11) Написать **м-функцию**, которая имеет **один входной параметр** – это массив временного ряда, и не имеет выходных.

Функция должна выполнять все перечисленные действия (кроме 1, 2, 10 пунктов) для того ВР, что был передан ей в качестве параметра.

12) Получить у преподавателя **mat-файлы**, содержащие массивы некоторых ВР, **по вариантам**. Номер варианта определяется по последним двум цифрам студенческого билета.

13) Загрузить из этих **mat-файлов** массив ВР. Используйте уже написанную функцию от этого ВР, для того, чтобы изучить его базовые характеристики.

14) Представить в отчете подробный анализ данного ВР для своего варианта.

15) Пояснить, с чем могут быть связаны особенности поведения или значений тех или иных характеристик.

5. Требования к оформлению отчета

Отчет должен обязательно содержать: постановку задачи, результаты выполнения пункта 4, графики соответствующих зависимостей с пояснениями, объяснения (которые требовались в ходе работы), заключение. Также весь код функций и сценариев добавляется в приложение в конце. У отчета должен быть оформлен грамотный титульный лист с указанием названия дисциплины, номера работы, фамилии преподавателя и ученика.