Basics of MATLAB-I

Dr.K.Srinivasa Rao
Professor & Head
Department of Mathematics
Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya
Kanchipuram, Tamilnadu

Material Description

Material Description

The material provides a gentle introduction to the MATLAB computing environment, and designed to give a basic understanding of MATLAB. No prior programming experience or knowledge of MATLAB is assumed.

Objectives

The main objectives are:

* Understanding the MATLAB environment
* Being able to do simple calculations using MATLAB
* Being able to carry out simple numerical computations and analyses using MATLAB

Prerequisites

Prerequisites

* No specific prerequisites are needed.
* It is advisable to have a have a good familiarity with PC operations.
* Basic knowledge of computer programming and an understanding of matrix and linear algebra are highly beneficial.

Features of MATLAB

* MATLAB is a high-level programming language with data structures, control flow statements, functions, output/input, and object-oriented programming
* It permits both, rapidly creating speedy throw-away programs, and creating complete, complex and large application programs.
* It provides an interactive environment that allows iterative exploration, design, and problem-solving.
* It is a bunch of tools that a programmer can use. It includes abilities for handling the variables in the workspace & importing/exporting data
* It also contains tools for development, handling, debugging, and profiling MATLAB files and offers built-in graphics useful for data visualizing, and tools for generating custom plots.

Features of MATLAB

* MATLAB is a high-level programming language with data structures, control flow statements, functions, output/input, and object-oriented programming
* It permits both, rapidly creating speedy throw-away programs, and creating complete, complex and large application programs.
* It provides an interactive environment that allows iterative exploration, design, and problem-solving.
* It is a bunch of tools that a programmer can use. It includes abilities for handling the variables in the workspace & importing/exporting data
* It also contains tools for development, handling, debugging, and profiling MATLAB files and offers built-in graphics useful for data visualizing, and tools for generating custom plots.

Features of MATLAB

* MATLAB is a high-level programming language with data structures, control flow statements, functions, output/input, and object-oriented programming
* It permits both, rapidly creating speedy throw-away programs, and creating complete, complex and large application programs.
* It provides an interactive environment that allows iterative exploration, design, and problem-solving.
* It is a bunch of tools that a programmer can use. It includes abilities for handling the variables in the workspace & importing/exporting data
* It also contains tools for development, handling, debugging, and profiling MATLAB files and offers built-in graphics useful for data visualizing, and tools for generating custom plots.

Features of MATLAB

* MATLAB is a high-level programming language with data structures, control flow statements, functions, output/input, and object-oriented programming
* It permits both, rapidly creating speedy throw-away programs, and creating complete, complex and large application programs.
* It provides an interactive environment that allows iterative exploration, design, and problem-solving.
* It is a bunch of tools that a programmer can use. It includes abilities for handling the variables in the workspace & importing/exporting data
* It also contains tools for development, handling, debugging, and profiling MATLAB files and offers built-in graphics useful for data visualizing, and tools for generating custom plots.

Features of MATLAB

* MATLAB is a high-level programming language with data structures, control flow statements, functions, output/input, and object-oriented programming
* It permits both, rapidly creating speedy throw-away programs, and creating complete, complex and large application programs.
* It provides an interactive environment that allows iterative exploration, design, and problem-solving.
* It is a bunch of tools that a programmer can use. It includes abilities for handling the variables in the workspace & importing/exporting data
* It also contains tools for development, handling, debugging, and profiling MATLAB files and offers built-in graphics useful for data visualizing, and tools for generating custom plots.

Features of MATLAB

* It offers a huge library of mathematical functions needed for computing statistics, linear algebra, numerical integration, filtering, Fourier analysis, optimization and solving regular differential equations.
* MATLAB Application Program Interfaces (APIs) allow users to write C/C++ and Fortran programs that directly interact with MATLAB.
* A Toolbox is a set of functions designed for a specific purpose and compiled as a package. These Toolboxes include MATLAB code, apps, data, examples and the documentation which helps users to utilize each Toolbox. Users can compile MATLAB files to create toolboxes if they require sharing with others.
* There are separate Toolboxes available from Mathworks, to be used for specific purposes, for example, text analytics, image processing, signal processing, deep learning, statistic & machine learning, and many more.

Features of MATLAB

* It offers a huge library of mathematical functions needed for computing statistics, linear algebra, numerical integration, filtering, Fourier analysis, optimization and solving regular differential equations.
* MATLAB Application Program Interfaces (APIs) allow users to write C/C++ and Fortran programs that directly interact with MATLAB.
* A Toolbox is a set of functions designed for a specific purpose and compiled as a package. These Toolboxes include MATLAB code, apps, data, examples and the documentation which helps users to utilize each Toolbox. Users can compile MATLAB files to create toolboxes if they require sharing with others.
* There are separate Toolboxes available from Mathworks, to be used for specific purposes, for example, text analytics, image processing, signal processing, deep learning, statistic & machine learning, and many more.

Features of MATLAB

* It offers a huge library of mathematical functions needed for computing statistics, linear algebra, numerical integration, filtering, Fourier analysis, optimization and solving regular differential equations.
* MATLAB Application Program Interfaces (APIs) allow users to write C/C++ and Fortran programs that directly interact with MATLAB.
* A Toolbox is a set of functions designed for a specific purpose and compiled as a package. These Toolboxes include MATLAB code, apps, data, examples and the documentation which helps users to utilize each Toolbox. Users can compile MATLAB files to create toolboxes if they require sharing with others.
* There are separate Toolboxes available from Mathworks, to be used for specific purposes, for example, text analytics, image processing, signal processing, deep learning, statistic & machine learning, and many more.

Features of MATLAB

* It offers a huge library of mathematical functions needed for computing statistics, linear algebra, numerical integration, filtering, Fourier analysis, optimization and solving regular differential equations.
* MATLAB Application Program Interfaces (APIs) allow users to write C/C++ and Fortran programs that directly interact with MATLAB.
* A Toolbox is a set of functions designed for a specific purpose and compiled as a package. These Toolboxes include MATLAB code, apps, data, examples and the documentation which helps users to utilize each Toolbox. Users can compile MATLAB files to create toolboxes if they require sharing with others.
* There are separate Toolboxes available from Mathworks, to be used for specific purposes, for example, text analytics, image processing, signal processing, deep learning, statistic & machine learning, and many more.

What Is MATLAB?

MATLAB (MATrix LABoratory) is an efficient user-friendly interactive software package, which is very effective for solving engineering, mathematical, and system problems.

MATLAB Windows

The assumption here is that the reader is sitting in front of an active computer and MATLAB is installed. To begin MATLAB, double click the MATLAB icon on the computer's desktop or select MATLAB from the Start or Program menu. Immediately a special window called the MATLAB desktop appears as below.

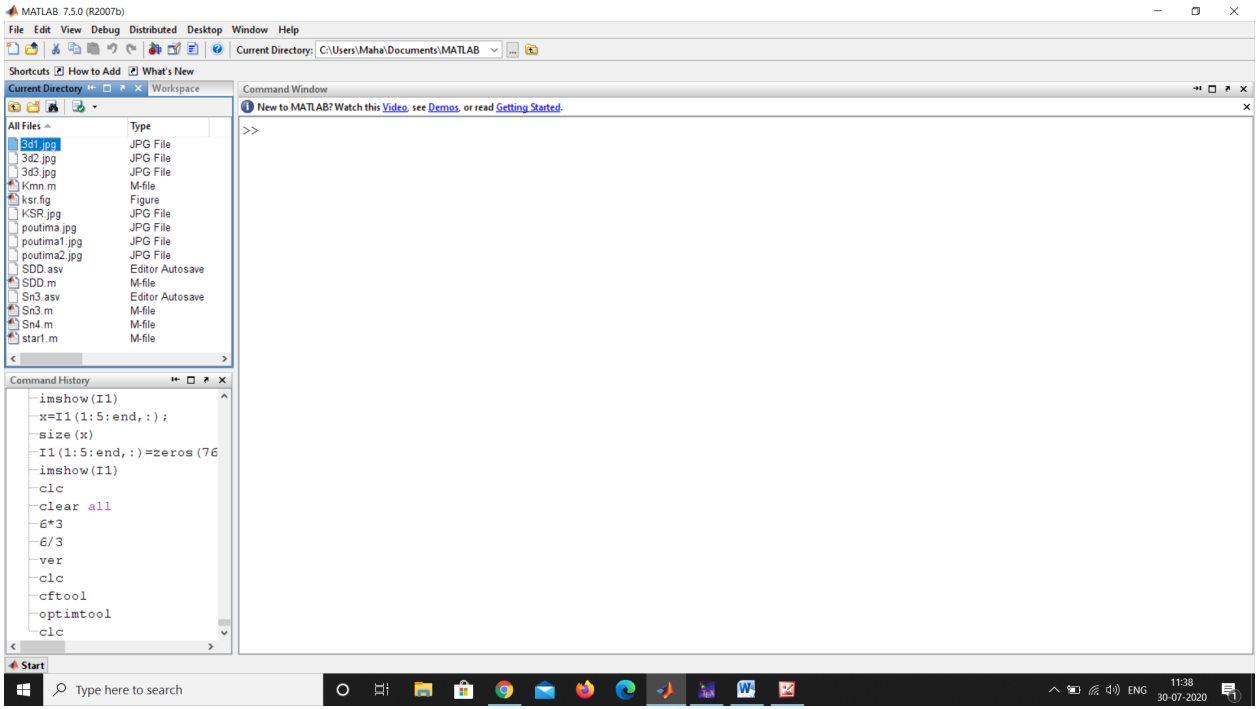

Figure: MATLAB Desktop

Componets of MATLAB Desktop

The MATLAB desktop consists of the following components:

1. Command Window
2. Command History
3. Workspace
4. Current Directory
5. Help Browser
6. Start Button

The command window is a place where certain basic operations like simple mathematical calculations can be easily performed. The difficulty with command prompt is that previously typed lines cannot be modified.

Componets of MATLAB Desktop

The Command history provides all the details of the functions typed and codes executed from the command prompt in the recent past.

The current directory shows the location where the program is held or the directory from where the current execution is running. Work folder in the drive where MATLAB is installed is the default folder where programs are stored.

Help browser is the way to go through the details of the documentation made available with MATLAB.

The prompt >> is the program prompt indicating that you are in the MATLAB environment. Each instruction line in the command window begins with a prompt (>>), which is automatically inserted by MATLAB. An instruction is executed after pressing the enter key. The result of a command appears on the next line. The result can be

- A MATLAB output
- A MATLAB prompt, meaning that the instruction was executed and MATLAB is waiting for the next command
- An error message

Input-Output

The following examples indicates the input-output relations of MATLAB and its response

$>> a = 6 * 3$ ← Input by user

$a =$

18 ← Output by MATLAB

$>>$ ← Waiting for the next command

A MATLAB prompt, meaning that the instruction was executed and MATLAB is waiting for the next command

$>> a = 6 * 3;$ ← Input by user

$>>$ ← Waiting for the next command (meaning that command is executed)

Input-Output

This activity of entering and executing commands is carried out in the main window (called command window) and is used to enter single line commands only. Besides the main window, there are two more windows are interest that are defined as follows:

- The figure window, which is used to display graphs and plots executed by a program entered at the command window
- The editor or debugger window is the place where program are created and modified. These programs can be saved in the form of files

When the user first enters MATLAB, the main program window or command window is active. The edit window is used only when a program is created or modified and then stored in a file. The graphic or figure window is created when plots are generated as a result of executing a set of instructions.

MATLAB Operations

Symbol	Operation	Example	Answer
+	Addition	6+3	9
-	Subtraction	6-3	3
*	Multiplication	6*3	18
/	Right Division	6/3	2
\	Left Division	3\6	2
^	Exponentation	2^3	8

Mathematical Functions

MATLAB Command	Function	Example
sqrt(x)	$\sqrt{x}$	sqrt(3)
log(x)	$log_e(x)$	log(2)
log10(x)	$log_{10}(x)$	log10(2)
exp(x)	e^x	exp(2)
sin(x)	$sinx$	sin(pi/3)
cos(x)	$cosx$	cos(pi/4)
acos(x)	$cos^{-1}x$	acos(0.2)

home, *clc*, *clear* commands

>> home

home moves the cursor to the upper-left corner of the Command Window. You can use the scroll bar to see the history of previous functions.

>> clc

clc clears all input and output from the Command Window display, giving you a "clean screen". After using clc, you cannot use the scroll bar to see the history of functions, but you still can use the up arrow to recall statements from the command history.

>> clear

clear removes all variables from the workspace. This frees up system memory.

Different *formats*

MATLAB always attempts to display integers (whole numbers) exactly. However, if the integer is too large, it is displayed in scientific notation with five significant digits, e.g. 1234567890 is displayed as 1.2346e+009 (i.e.1.234610^9). Check this by first entering 123456789 at the command line, and then 1234567890.
Numbers with decimal parts are displayed with four significant digits. If the value $x$is in the range $0.001 < x \leq 1000$ it is displayed in fixed point form, otherwise scientific (floating point) notation is used, in which case the mantissa is between 1and 9.9999, e.g. 1000.1 is displayed as 1.0001e+003. Check this by entering following numbers at the prompt (on separate lines): 0.0011, 0.0009, 1/3, 5/3, 2999/3, 3001/3

Different *formats*

This is what is called the default format, i.e. what normally happens. However, you can change from the default with variations on the format command, as follows. If you want values displayed in scientific notation (floating point form) whatever their size, enter the command

>> format short e

All output from subsequent display statements will be in scientific notation with five significant digits, until the next format command is issued. Enter this command and check it with the following values: 0.0123456, 1.23456, 123.456 (all on separate lines). If you want more accurate output, you can use

>> format long e

This also gives scientific notation, but with 15 significant digits. Try it out on 1/7.

format short, format long, format bank, and format rat

The commonly used formats are format short, format long, format bank, format rat

We can change the format to format long by typing

>> format long

View the result of value of $\sqrt{3}$ by typing

>> sqrt(3)

ans =

1.732050807568877

When the format is set to format short

>>format short

>> sqrt(3)

ans =

1.7321

format short, format long, format bank, and format rat

When the format is set to format bank

```
>>format bank
>> sqrt(3)
ans =
      1.73
```

When the format is set to format rat

```
>> format rat
>> sqrt(3)
ans =
      1351/780
```

Note :The format function affects only how numbers are displayed, not how MATLAB computes or saves them.

Difference between fix, round, ceil and floor commands

```
>> round(6.628)
ans =
      7
```

round command rounds the element of x to towards nearest integer

```
>> fix(6.628)
ans =
      6
```

fix command rounds the element x to nearest integer towards zero

fix, ceil, floor, round commands

```
>> ceil(6.628)
ans =
       7
```

ceil command rounds the element x to nearest integer towards infinity

```
>> floor(6.628)
ans =
       6
```

floor command rounds the element x to nearest integer towards minus infinity

Arrays & Matrices

Row Vectors

Arrays are the backbone of MATLAB computation. A 2-D array is a list of numbers arranged in rows and columns. If you form an array by writting numbers in rows, all rows must have the same number of entities. Same is true for columns. An array with m rows and n columns is called $m \times n$ array and it has a total of $m.n$ entries.

Array Creation:

To create an array with six elements in a single row, separate the elements with either a comma (,) or a space.

```
>> a=[3 5 7 -8 5 2]
a=
     3 5 7 -8 5 2
```

This type of array is called *row vector*

Commands-Outputs

Let a, b be two row vectors of same size.

Command	**Output**
size(a)	We get size of the array
$a + b$	Performs point-wise addition
$a - b$	Performs point-wise subtraction
a.*b	Performs point-wise multiplication
a./b	Performs point-wise division
sum(a)	We get sum of all the elements of a
prod(a)	We get product of all the elements of a
a(i)	i^{th} element of a will be displayed
a(end)	Last element of a will be displayed
a(i)=y	i^{th} element of a will be replaced by y
max(a)	maximum element of a will be replaced

Creating Row Vectors

Very often we need to create a vector of numbers over a given range with specified increment.The general command do this in MATLAB is

>> v=a:h:b

Where a is the initial value , h is the increment value, b is the final value. If no increment is specified, MATLAB uses the default increament of 1.

Examples:

```
>> a=1:2:13
a=
     1 3 5 7 9 11 13
>> b=1:8
b=
     1 2 3 4 5 6 7 8
```

Two frequently used built-in-functions to generate vectors:

>> v=linspace(a,b,n)

It generates a linearly spaced vector of length n from a to b i.e., It divides [a,b] into n-1 equally spaced sub-intervals

Examples:

>> a=linspace(0,20,5)

a=

0 5 10 15 20

Thus,v=linspace(a,b,n) is the same as $v = a : \frac{b-a}{n-1} : b$

Also v=linspace(a,b) produce a vector of length 100 from a to b

Relational Operators

Relational Operators

Relational operators can also work on both scalar and non-scalar data. Relational operators for arrays perform element-by-element comparisons between two arrays and return a logical array of the same size, with elements set to logical 1 (true) where the relation is true and elements set to logical 0 (false) where it is not. The following table shows the relational operators available in MATLAB.

Operator	Description
$<$	less than
$>$	greater than
$<=$	less than or equal to
$>=$	greater than or equal to
$==$	equal to
$\sim=$	not equal to

For example,

```
>> a=[2 4 2 5 7 8 9 10]; b=[5 7 3 5 3 2 6 3];
>> a>4
ans =
     0   0   0   1   1   1   1   1
>> a==b
ans =
     0   0   0   1   0   0   0   0
```

Syntax of find command: ind = find(X)

ind = find(X) locates all nonzero elements of array X, and returns the indices of those elements in vector **ind**. If X is a row vector, then ind is a row vector; otherwise, ind is a column vector. If X contains no nonzero elements or is an empty array, then ind is an empty array.

```
>> find(a> 4)

ans =

     4   5   6   7   8

>> find(a==b)

ans =

     4
```

Sorting

For vectors, **sort(X)** sorts the elements of X in ascending order and for matrices,**sort(X)** sorts each column of X in ascending order.

```
>> m=[ 28 3 5 -10 0 6 4 3];

>> sort(m)

ans =

    -10   0   3   3   5   6   28

>>[n ind]=sort(m)

n =

    -10   0   3   3   5   6   28

ind =

     4   5   2   7   3   6   1
```

```
>> sort(m,'descend')

ans =

28   6   5   3   3   0   -10
```

Problems

Sum of first 'n' natural numbers

Find the sum of first 100 natural numbers?

OR

Find the value of $\sum_{n=1}^{100} n$

Sum of first 'n' natural numbers

Find the sum of first 100 natural numbers?

OR

Find the value of $\sum_{n=1}^{100} n$

We need a sequence $S = \{1, 2, 3, \cdots, 100\}$

Sum of first 'n' natural numbers

Find the sum of first 100 natural numbers?

OR

Find the value of $\sum_{n=1}^{100} n$

We need a sequence $S = \{1, 2, 3, \cdots, 100\}$

Next, we have to add all the numbers in the set S

Sum of first 'n' natural numbers

Find the sum of first 100 natural numbers?

OR

Find the value of $\sum_{n=1}^{100} n$

We need a sequence $S = \{1, 2, 3, \cdots, 100\}$

Next, we have to add all the numbers in the set S

* >> S=1:100;

Sum of first 'n' natural numbers

Find the sum of first 100 natural numbers?

OR

Find the value of $\sum_{n=1}^{100} n$

We need a sequence $S = \{1, 2, 3, \cdots, 100\}$

Next, we have to add all the numbers in the set S

* >> S=1:100;
* >> value =sum(S)

Sum of squares of first 'n' natural numbers

Find the sum of the squares of first 100 natural numbers?

OR

Find the value of $\sum_{n=1}^{100} n^2$

Sum of squares of first 'n' natural numbers

Find the sum of the squares of first 100 natural numbers?

OR

Find the value of $\sum_{n=1}^{100} n^2$

We need a sequence $S = \{1^2, 2^2, 3^2, \cdots, 100^2\}$

Sum of squares of first 'n' natural numbers

Find the sum of the squares of first 100 natural numbers?

OR

Find the value of $\sum_{n=1}^{100} n^2$

We need a sequence $S = \{1^2, 2^2, 3^2, \cdots, 100^2\}$

Next, we have to add all the numbers in the set S

Sum of squares of first 'n' natural numbers

Find the sum of the squares of first 100 natural numbers?

OR

Find the value of $\sum_{n=1}^{100} n^2$

We need a sequence $S = \{1^2, 2^2, 3^2, \cdots, 100^2\}$

Next, we have to add all the numbers in the set S

* >> S=1:100;

Sum of squares of first 'n' natural numbers

Find the sum of the squares of first 100 natural numbers?

OR

Find the value of $\sum_{n=1}^{100} n^2$

We need a sequence $S = \{1^2, 2^2, 3^2, \cdots, 100^2\}$

Next, we have to add all the numbers in the set S

* >> S=1:100;
* >> value = sum(S.^ 2)

Sum of first 'n' odd numbers

Find the sum of first 50 odd numbers?

OR

Find the value of $\sum_{n=1}^{50}(2n-1)$

Sum of first 'n' odd numbers

Find the sum of first 50 odd numbers?

OR

Find the value of $\sum_{n=1}^{50}(2n-1)$

We need a sequence $S = \{1, 3, 5 \cdots, 99\}$

Sum of first 'n' odd numbers

Find the sum of first 50 odd numbers?

OR

Find the value of $\sum_{n=1}^{50}(2n-1)$

We need a sequence $S = \{1, 3, 5 \cdots, 99\}$

Next, we have to add all the numbers in the set S

Sum of first 'n' odd numbers

Find the sum of first 50 odd numbers?

OR

Find the value of $\sum_{n=1}^{50}(2n-1)$

We need a sequence $S = \{1, 3, 5 \cdots, 99\}$

Next, we have to add all the numbers in the set S

* >> S=1:2:100;

Sum of first 'n' odd numbers

Find the sum of first 50 odd numbers?

OR

Find the value of $\sum_{n=1}^{50}(2n-1)$

We need a sequence $S = \{1, 3, 5 \cdots, 99\}$

Next, we have to add all the numbers in the set S

* >> S=1:2:100;
* >> value = sum(S)

Write a MATLAB program that adds all elements of array S with even indices

Indexing

Write a MATLAB program that adds all elements of array S with even indices

Let S=[3 6 2 7 5 -10 4 92 3 44 76]

Write a MATLAB program that adds all elements of array S with even indices

Let S=[3 6 2 7 5 -10 4 92 3 44 76]

We need S(2)+S(4)+S(6)+S(8)+S(10)

Write a MATLAB program that adds all elements of array S with even indices

Let S=[3 6 2 7 5 -10 4 92 3 44 76]

We need S(2)+S(4)+S(6)+S(8)+S(10)

* >> P=S(2:2:end)
* >> value = sum(P)

Problem

Find the value of $\sum_{n=1}^{N} a^N$, for $a = \frac{1}{2}$ and $N = 50$

Problem

Find the value of $\sum_{n=1}^{N} a^N$, for $a = \frac{1}{2}$ and $N = 50$

We need a sequence $S = \{0.5^1, 0.5^2, 0.5^3 \cdots, (0.5)^{50}\}$

Problem

Find the value of $\sum_{n=1}^{N} a^N$, for $a = \frac{1}{2}$ and $N = 50$

We need a sequence $S = \{0.5^1, 0.5^2, 0.5^3 \cdots, (0.5)^{50}\}$

Next, we have to add all the numbers in the set S

Problem

Find the value of $\sum_{n=1}^{N} a^N$, for $a = \frac{1}{2}$ and $N = 50$

We need a sequence $S = \{0.5^1, 0.5^2, 0.5^3 \cdots, (0.5)^{50}\}$

Next, we have to add all the numbers in the set S

* >> N=1:50;

Problem

Find the value of $\sum_{n=1}^{N} a^N$, for $a = \frac{1}{2}$ and $N = 50$

We need a sequence $S = \{0.5^1, 0.5^2, 0.5^3 \cdots, (0.5)^{50}\}$

Next, we have to add all the numbers in the set S

* >> N=1:50;
* >> S=(0.5).^ N

Problem

Find the value of $\sum_{n=1}^{N} a^N$, for $a = \frac{1}{2}$ and $N = 50$

We need a sequence $S = \{0.5^1, 0.5^2, 0.5^3 \cdots, (0.5)^{50}\}$

Next, we have to add all the numbers in the set S

* >> N=1:50;
* >> S=(0.5).^ N
* >> value = sum(S)

Problem

Find the value of $\sum_{n=1}^{9} n^n$

Problem

Find the value of $\sum_{n=1}^{9} n^n$

We need a sequence $S = \{1^1, 2^2, 3^3 \cdots, 9^9\}$

Problem

Find the value of $\sum_{n=1}^{9} n^n$

We need a sequence $S = \{1^1, 2^2, 3^3 \cdots, 9^9\}$

Next, we have to add all the numbers in the set S

Problem

Find the value of $\sum_{n=1}^{9} n^n$

We need a sequence $S = \{1^1, 2^2, 3^3 \cdots, 9^9\}$

Next, we have to add all the numbers in the set S

* >> n=1:9;

Problem

Find the value of $\sum_{n=1}^{9} n^n$

We need a sequence $S = \{1^1, 2^2, 3^3 \cdots, 9^9\}$

Next, we have to add all the numbers in the set S

* >> n=1:9;
* >> S=n.^ n

Problem

Find the value of $\sum_{n=1}^{9} n^n$

We need a sequence $S = \{1^1, 2^2, 3^3 \cdots, 9^9\}$

Next, we have to add all the numbers in the set S

* >> n=1:9;
* >> S=n.^ n
* >> value = sum(S)

Roots of a polynomial equation

Find the roots of the equation $x^5 - 3x^3 + 5x + 1 = 0$

Roots of a polynomial equation

Find the roots of the equation $x^5 - 3x^3 + 5x + 1 = 0$

* >> **a=[1 0 -3 0 5 1]**

Roots of a polynomial equation

Find the roots of the equation $x^5 - 3x^3 + 5x + 1 = 0$

* >> **a=[1 0 -3 0 5 1]**
* >> **roots(a)**

Roots of a polynomial equation

Find the roots of the equation $x^5 - 3x^3 + 5x + 1 = 0$

* >> **a=[1 0 -3 0 5 1]**
* >> **roots(a)**

1.4116 + 0.6474i
1.4116 - 0.6474i
-1.3090 + 0.5550i
-1.3090 - 0.5550i
-0.205

Lab Assignment-I

Lab-Assignment-I

The following exercises are meant to be answered by a single MATLAB command. The command may be involved (i.e., it may use a number of parentheses or calls to functions) but can, in essence, be solved by the execution of a single command. If the command is too complicated, feel free to break it up over two or more lines.

1. Create a vector of the even numbers between 31 and 75.
2. Let x = [2 5 1 6].

   Add 16 to each element
   Add 3 to just the odd-index elements
   Compute the square root of each element
   Compute the square of each element

Lab-Assignment-I

3. Let x = [3 2 6 8]' and y = [4 1 3 5]' (NB. x and y should be column vectors).

   (a). Add the sum of the elements in x to y

   (b). Raise each element of x to the power specified by the corresponding element in y.

   (c). Divide each element of y by the corresponding element in x

   (d). Multiply each element in x by the corresponding element in y, calling the result z.

   (e). Add up the elements in z and assign the result to a variable called w.

   (f). Compute x'*y - w and interpret the result

Lab-Assignment-I

4. Create a vector x with the elements (don't manually assign values; use MATLAB commands).
   (a). 2, 4, 6, 8
   (b). 10, 8, 6, 4, 2, 0, -2, -4
   (c). 1, 1/2, 1/3, 1/4, 1/5
   (d). 0, 1/2, 2/3, 3/4, 4/5
5. Given a vector, t, of length n, write down the MATLAB expressions that will correctly compute the following:
   (a). $ln(2 + t + t^2)$
   (b). $e^t(1 + cos3t)$
   (c). $cos^2 t + sin^2 t$

Test that your solution works for t = 1:0.2:2

Lab Assignment-II

Lab Assignment-II

The following exercises are meant to be answered by a single MATLAB command. The command may be involved (i.e., it may use a number of parentheses or calls to functions) but can, in essence, be solved by the execution of a single command. If the command is too complicated, feel free to break it up over two or more lines.

1. Given x = [3 1 5 7 9 2 6], explain what the following commands 'mean' by by summarizing the net result of the command.

   (a). x(3) (b). x(1:3) (c). x(1:end) (d). x(6:-2:1)
   (e). x([1 6 2 1 1]) (f) sum(x) and prod(x)

2. Given the array A = [2 4 1 6 7 2 3 5 9], provide the commands needed to

   (a). replace the second element by 0
   (b). replace the even indexed elements by [0 4 8 12]
   (c). remove third element

Lab-Assignment-II

3. Create a vector x with the elements, $x_n = \frac{(-1)^n + 1}{2n - 1}$. Add up the elements of the version of this vector that has 100 elements.
4. Write a MATLAB program to find a factorial of n for given n value
5. Given the arrays A = [2 3 7 8 4 1 9] and B=[2 4 6] provide the commands needed to
   (a). merge A and B horizontally
   (b). insert the value 10 between A(4) and A(5)
   (c). insert an array [2 5 10] between A(3) and A(4)
   (d). merge A and B vertically

Lab-Assignment-III

1. Report the outputs and your comments from the following:
   a. [4 -1 7 5 3].*[5 -9 6 5 -3]
   b. [4 -1 7 5 3] >[5 -9 6 5 -3] and [4 -1 7 5 3] > 2
   c. What is(are) your comment(s)/observation(s) on (a) and (b)
   d. [4 -1 7 5 3] <=4
   e. sum([4 -1 7 5 3] <=4)
   f. sum([4 -1 7 5 3]>2)+2
   g. sum([4 -1 7 5 3]>2+2)
   h. What is (are) your comment(s)/observation(s) on (f) and (g)
   i. find([4 -1 7 5 3])
   j. find([4 -1 7 5 3]==7)
   k What is (are) your comment(s)/observation(s) on (i) and (j)

Lab-Assignment-III

2. a=[16 97 96 49 80 14 42 92 79 96], provide the command needed to get [80 49 96 97 16 96 79 92 42 14]
3. Add just -100 to even indexed elements of a
4. Let x=[1 2 3 4], write are the sizes of the arrays (i) [x x x x], (ii) [x x ; x x], and (iii) [x; x; x; x] ? what about the out puts of [x; x x x], [x x x; x]?

Matrices

A matrix is entered row-wise, with consecutive elements of a row separated by a space or a comma, and the rows separated by semi-colons or carriage returns. The entire matrix must be encloed with square brackets. For example,

```
>> a=[1 2 3; 6 4 5; 1 4 2]
a=
     1   2   3
     6   4   5
     1   4   2
>> size(a)
ans=
     3   3
```

Commands-Outputs

Let a, b be two matrices of same size.

Command	**Output**
a'	trancepose of a
$a + b$	Performs addition of two matrices
$a - b$	Performs subtraction of two matrices
a*b	Performs matrix multiplication
a.*b	Performs point-wise multiplication
diag(a)	We get the diagonal elements of a
rank(a)	We get the rank of the matrix a
det(a)	We get the determinant of a
inv(a)	inverse of a provided a is non-singular
[a b]	a and b will be merged horizontally
[a;b]	a and b will be merged vertically
a(i,j)	Displays i^{th}-row and j^{th}- column element of a

Command	Output
a(i,j)	Displays i^{th}-row and j^{th}- column element of a
a(i,j)=5	Displays i^{th}-row and j^{th}- column element of a by 5
a(1,:)	Displays first row of a
a(:,2)	Displays second column of a
a([2 3],:)	Displays 2nd and 3rd rows of a
a(:,[1 3])	Displays first and 3rd colums of a
a(2,:)=[]	Removes the 2nd row of a
a(:)	Displays all the elements of a in column-wise
poly(a)	We get characteristic polynomial of a
roots(poly(a))	We get the eigenvalues of a
eig(a)	We get the eigenvalues of a
fliplr(a)	Flips the elements of a from left to right
diag(fliplr(a))	Displays the second diagonal elements of a

Eigenvalues & Eigen Vectors

>> [V,D]=eig(a) produces a diagonal matrix D of eigenvalues of *a* and a full matrix V whose columns are the corresponding eigen vectors

```
>> a=[1 1 3;1 5 1;3 1 1];
>> [eigvec eigval]=eig(a)
eigvec=
      -0.7071    0.5774    0.4082
      -0.0000   -0.5774    0.8165
       0.7071    0.5774    0.4082
eigval=
      -2    0    0
       0    3    0
       0    0    6
```

Special Matrices

To aid matrix generation and manipulation, MATLAB provides utility matrices. For example

Command	**Output**
eye(m,n)	Returns $m \times n$ matrix with ones on the main diagonal
zeros(m,n)	Returns $m \times n$ matrix of zeros
ones(m,n)	Returns $m \times n$ matrix of ones
rand(m,n)	Returns $m \times n$ matrix of random numbers
randn(m,n)	Returns $m \times n$ matrix of normally distributed numbers
diag(v)	Returns a diagonal matrix with vector v on the diagonal
diag(a,1)	Extracts the first upper-diagonal vector of matrix a
diag(a,-1)	Extracts the first lower-diagonal vector of matrix a

The first five commands with single argument, i.e., eye(m), returns unit matrix of order m. For example, eye(3) returns unit matrix of order 3. ones(4) returns matrix of ones of order 4.

Sub-Matrices

Consider a matrix

>> s=round(100*rand(6))

s=

81	28	96	79	68	71
91	55	49	96	76	3
13	96	80	66	74	28
91	96	14	4	39	5
63	16	42	85	66	10
10	97	92	93	17	82

The following command returns the 3rd and 4th column of s

>> s(:,[3 4])

The following command returns the 1st and 2nd row of s

>> s([1 2],:)

Also we can replace the two rows by new rows with the following command

>> s([1 2],:) =[1 3 2 1 3 2;0 9 8 9 8 8]

Sub-Matrices

The following command returns 2×2 matrix which is situated at the centre of S

>> s([3 4],[3 4])

Reshaping Matrices

Matrices can be reshaped into a vector or any other appropriately sized matrices:

As a vector: All the elements of matrix A can be strung into a single-column vector B by the command B=A(:) (matrix A is stacked in vector B columnwise)

As a differently sized matrix: If matrix A is a $m \times n$ matrix, it can be reshaped into a $p \times q$ matrix , as long as m.n=p.q, with the command **reshape(A,p,q)**.

For example, if A is 6×6 matrix, then **reshape(A,9,4)** transforms A into 9×4 matrix and **reshape(A,3,12)** transforms A into a 3×12 matrix

Lab Assignment-IV

Assignment-IV

The following exercises are meant to be answered by a single MATLAB command. The command may be involved (i.e., it may use a number of parentheses or calls to functions) but can, in essence, be solved by the execution of a single command. If the command is too complicated, feel free to break it up over two or more lines.

1. Given the array A = [2 4 1; 6 7 2; 3 5 9], provide the commands needed to
    a. assign the first row of A to a vector called x1
    b. assign the last 2 rows of A to an array called y
    c. compute the sum over the columns of A
    d. compute the sum over the rows of A
    e. compute the standard error of the mean of each column of A (NB. the standard error of the mean is defined as the standard deviation divided by the square root of the number of elements used to compute the mean.)

Lab-Assignment-IV

2. Given the arrays x=[1 4 8], y=[2 1 5] and A=[3 1 6 ; 5 2 7]. Determine which of the following statements will correctly execute and provide the result. If the command will not correctly execute, state why it will not.

   a. x + y b. x + A c. x' +y d. A - [x' y']
   e. [x ; y'] f. [x ; y] g. A - 3

3. Given the array A = [2 7 9 7 ; 3 1 5 6 ; 8 1 2 5], explain the results of the following commands:

   a. A' b. A(:,[1 4]) c. A([2 3],[3 1])
   d. reshape(A,2,6) e. A(:) f. flipud(A)
   g. fliplr(A) h. [A A(end,:)] i. A(1:3,:)
   j. [A ; A(1:2,:)] k. sum(A) l. sum(A')
   m. sum(A,2) n. [[A ; sum(A)] [sum(A,2) ; sum(A(:))]]

Lab-Assignment-IV

4. Given the array A from problem 3, above, provide the command that will
    a. assign even-numbered columns of A to array called B
    b. assign the odd-numbered rows to an array called C
    c. convert A into a 4-by-3 array
    d. compute the reciprocal of each element of A
    e. compute the square-root of each element of A
5. Write a MATLAB program that creates matrix A composed of four sub-matrices B,C D and E arranged forming the structure indicated as A= $\begin{pmatrix} B & C \\ D & E \end{pmatrix}$ where B,C,D and E are 5×5 matrices, with the following characteristics: B is an identity matrix; C consists of ones in the first two columns and rest elements are zeros; D consists of zeros in the first three columns and remaining two columns are ones; E consists of the sequence of integers 1 through 25 in ascending order, column by column, from left to right.

Lab Assignment-V

Assignment-V

1. Write a MATLAB program that returns 6x6 random matrix C with random elements consists of integers between 1 and 100
2. Determine the main diagonal of C
3. The sum and product of the main diagonal of C
4. The average of the elements of the main diagonal of C
5. The maximum and minimum values of the elements and corresponding indices on the main diagonal of C
6. The maximum and minimum values of the elements and corresponding indices on the main diagonal of C
7. Find rank and determinant of C
8. Find the Inverse, transpose and Square each element of C
9. Display second fourth and fifth rows
10. Reshape the matrix C into a 4X9 and 9x4 matrices
11. Replace the 5th row by [1 0 1 0 1] and name it as E

Useful Youtube links:

www.youtube.com/watch?v=O41BWhXFu8E

www.youtube.com/watch?v=83S48Fs9WhY

Useful Books:

Getting Started with MATLAB by Rudra Pratap, Oxford University Press

Practical MATLAB-Basics for Engineers, CRC Press, Misza Kalechman

Useful Website:

www.mathworks.com