



Functions & Data Import/Export

Week 4

Loosely follows Chapters 3 & 4



What is a function?

- A computational expression that uses one or more input values to produce an output value.
- MATLAB functions have three components
 - **input**, **output**, and **name**
 - `b = tan(x)`
 - x is the input, b is the output, and tan is the name of a built-in function.
 - This is the tangent function and accepts input arguments in radians.

MATLAB Functions

- Functions take the form:
 - `variable = function(argument, arguments)`
- Many built in functions (`sin()`, `tan()`, `input()`, `fprintf()`, etc)
- You simply need to know the name and what the input values are
- For example, the square root function: `sqrt()`
- To find the square root of 9
 - `a = sqrt(9)`

Rounding Functions

- Many times there are multiple functions that perform similar tasks
- Enter the following into matlab

```
x=16.3  
y=3.9
```

```
round(x)  
floor(x)  
fix(x)  
ceil(x)
```

- What do each of them do?

Discrete Math functions

`factor(10)`

`rats(4.2)`

`factorial(3)`

`gcd(20, 10)`

`lcm(4, 6)`

- What do each of these functions do?
- If you aren't sure, use the help feature (`help <topic>`)

Trigonometric Functions

- MATLAB can compute trig functions in degrees or radians.
- To convert degrees to radians, use the relationship
 - $180 \text{ degrees} = \pi * \text{radians}$
 - Or use `deg2rad(var)`

Exercise

- Open a new M-File
- Use MATLAB to find the sine of 360 degrees
- Use MATLAB to find the arccosine of -1 in degrees (use help if necessary)
- Use MATLAB to find the inverse tangent of x in radians as x ranges from -1 to 1 in increments of 0.1

Data Analysis Functions

- MATLAB has many statistical functions built-in:

max()
min()
mean()
median()
covar()

sum()
prod()
sort()
sortrows()

size()
length()
std()
var()

Exercise

- Open another M-File, given:
 - `x = [5, 3, 7, 10, 4]`
- What is the largest number in vector x and where is it located?
 - `[value, position] = max(x)`
 - value = 10
 - position = 4
- What is the median of the vector x?
 - `median(x)`
 - ans = 5
- What is the sum of vector x?
 - `sum(x)`
 - ans = 29

Exercise

- Open another M-File, given:
 - $v = [2, 24, 53, 7, 84, 9]$
 - $y = [2, 4, 56; 3, 6, 88]$
- Sort v in descending order
- Find the size of y
- Find the standard deviation of v
- Find the cumulative product of v
- Sort the rows of y based on the 3rd column

Generating Random Numbers

- `rand(n)` produces an $n \times n$ matrix of random numbers from 0 to 1
- `rand(n,m)` produces an $n \times m$ matrix of random numbers from 0 to 1
- To produce a random number between x and y use the following formula:
 - $x + (y - x) .* \text{rand}(1)$
- That means to produce a rand between 0 and y use the following formula:
 - $y .* \text{rand}(1)$

Complex Numbers

- Complex numbers are represented by $a+bi$ or $a+bi$
 - a is the real part
 - b is the imaginary part
 - $z = 2 + 3i$ OR $z = 2 + 3i$
- Since ``i`` is used to assign complex numbers, it can be a bad choice for a variable name.
- Complex number can be assigned using the constant `i` or the function for complex numbers.
 - `a = 2; b = 3;` % Given some values `a` and `b`
 - `c = a + b*i;` % Assign `c` with the constant `i`
 - `c = complex(a,b);` % OR Assign `c` with the function `complex()`

More Complex Number Functions

- To find the real and imaginary components of a complex number:
 - `real(c)`
 - `imag(c)`
- To find the absolute value or modulus of a complex number:
 - `abs(c)`
- To find the angle or argument expressed in radians of a complex number:
 - `angle(c)`

Useful Constants

- `clock` produces an array with the year, month, day, hour, min, sec
 - `date` tells the date
 - `pi` the number pi (3.141592653589...)
 - `i` imaginary number ($i = \sqrt{-1}$)
 - `j` imaginary number ($j = \sqrt{-1}$)
-
- Remember, you should not use variable names that share names with constants built into MATLAB.

Importing & Exporting

- Many programs deal with data
- Sometimes data needs to be shared

Importing and Exporting

Typical Data File Types

Binary

- Machine Language
- Fast & Efficient
- Not readable
- Usually proprietary
- Examples
 - .xlsx
 - .docx
 - .mat

ASCII (Plain Text)

- “Text File”
- Easily read in any text reader
- Good for sharing
- Examples
 - .txt
 - .dat
 - .csv

Import Wizard

- Feature that determines
 - The type of data file
 - The way to extract and display information
- Can extract from ASCII and Binary files
- Simply double click on a file in the Directory Window

Import Wizard Functions

- The import wizard can be called using various functions
 - `uiimport('filename.ext')`
 - ~~`xlsread('filename')`~~
 - `csvread('filename')/readmatrix('filename')`
 - `textread('filename')`

NOTES:

- The file must be in the current path in order to simply use the filename
- Excel must be installed for MATLAB to read/write Excel data

Exporting to Excel

- An array in MATLAB can be exported to Excel
- The following is an example
 - ~~`xlswrite("filename.xlsx", someArray)`~~
 - `writematrix(array, "filename.xlsx")`

Exercise

Open an M-file and write a program that does the following

- Create an array of odd numbers from 1 to 19
- Save the array to an excel document
- Clear your workspace variables
- Import the array from the Excel file you just created.

Import Text Files

- Another function that is used to import data is `textread()`.
- `textread()` can only read ASCII files
- The file must be formatted into columns but each column can be different
 - `[a, b, c, d] = textread("filename.ext", "%f %i %i %i", n)`
- `a, b, c, and d` represent the names of each variable
- `Filename` is the name of the file
- `'%f %i %i %i'` is the formatspec string indicating the format of each column
- `n` is the number of rows to be read

NOTE: The formatspec and n are optional (see help for more info)

Example

- Assume a file 'sports.dat' contains:

```
University, Soccer, 12, 7, 2  
University, Hockey, 15, 7, 3
```

- To read it you would enter
 - `[sport, wins, losses, ties] = textread("sports.dat", "%*s, %s, %d, %d, %d", 2)`
- `%s` denotes the column contains strings
- `%d` denotes the column contains integers
- `%*s` or `%*d` means the column will not be read into matlab

Binary Matlab .mat Files

- You may save or load variables into your matlab workspace
 - `save filename var1 var2 var3`
 - `load filename`
- Filename is the name of the file and var1, var2, and var3 are the variables to be saved in a binary MATLAB file.
- If no variables are listed, all workspace variables are saved
- The data is saved into a file with a .mat extension
- Load variables from .mat files with a simple `load` command

Exercise

- In the command window...
- Define x as 6, t as 14.5 and r as 22
- Save these variables into a file titled “work_data”.
- Clear the workspace and reload the variables from the work_data file