Matlab Tutorial

- Create your own project directory
- Run Maltab by clicking Matlab icon,
  or type “matlab” in the command shell
- Change directory to your directory
  - “**pwd**”: show current directory
  - “**cd [where-you-wanna-go]**”: change directory
  - “**dir, ls**”: show the files under current directory
  - “**path**”: show the search paths of Matlab (usually need to **addpath** when your functions are not under the current directory)
  - “**help, doc**”: show introduction of built-in functions

Matlab Tutorial

- Some useful links for quick-start:
  - http://de.mathworks.com/help/matlab/getting-started-with-matlab.html (**tutorials**, interactive learning, and **videos**)
  - http://de.mathworks.com/products/matlab/examples.html some basic examples, please run through the following demos:
    - ***Mathematics***
      - *Basic Matrix Operation*
      - *Matrix Manipulation*
    - ***Graphics***
      - *2-D Plots*
      - *3-D Plots*
    - ***Images and Matrices***
      - *Programming*
      - *Manipulating Multidimensional Arrays*
      - *Function Functions*
  - http://www.ml.uni-saarland.de/MatlabTutorial/MatlabTutorial.pdf from Prof. Matthias Hein

Matlab Tutorial

- common syntax:
  - >> **expression**
    - >> a = 10+4
  - >> **function(parameter1, parameter2)**
    - >> a = sum(b, 1)
  - >> **[output_variable1, output_variable2] = function(parameter1, parameter2)**
    - >> [ia, ic] = sort(b, 1)
  - **%** for comments
- Add semicolon **;** if you don't wanna see the results directly on the screen
- functions/variables are **case-sensitive**:
  - helloworld <-> HELLOWORLD are 2 different variables

Matlab Tutorial

- Workspace:
  - No need to pre-allocate or declare variables
    - but sometimes it will speed up the code (http://www.slideshare.net/jbhuang/writing-fast-matlab-code)
  - variables will be automatically stored in the workspace:
    - "**who**": to list all the variables existing in your workspace
    - "**whos**": similar to "**who**", but will all details such like size, class
    - "**clear**": to clear all the variables in the workspace
    - "**clear name-of-variable**": to clear a specific variable

Matlab Tutorial

- Data types:
  - http://de.mathworks.com/help/matlab/data-types_data-types.html
- Basic operators:
  - **+**, **-**, *, **/**, \, **^**, ‘
  - **elementwise**:
    - with “dot” in front of operators:
      ex: [2, 3, 4] **.*** [3, 4, 5] = [6, 12, 20]
- Relational operators:
  - **<**, **<=**, **>**, **>=**, **==**, **~=**
- logical operators:
  - **&**, **|**, **~**, **xor**

Matlab Tutorial

- Some algorithmic functions:
  - Trigonometric functions:
    - **sin** / **cos** / **tan** / **cot** / **asin** / **acos** / **atan** / **atan2** / …
  - Exponentials and Logarithms:
    - **sqrt** / **exp** / **log** / **log10**
  - Elemental functions:
    - **ceil** / **floor** / **round** / **sign**

Matlab Tutorial

- Matrix:
  - a = [1 2 3; 4 5 6; 7 8 9]
    use **space or comma** to separate elements on the same row, **semicolon** to next row
    - a =
      1 2 3
      4 5 6
      7 8 9
  - to index the element in a matrix (**row index first**, then column index):
    - >> a(2,1)
      ans = 4
    - use **colon to index all elements**
      >> a(2,:)
      ans = 4 5 6
  - **equidistant** elements
    - **start_value : interval : end_value**
      >> x = 0:2:10
      x = 0 2 4 6 8 10
    - **default interval to be 1**:
      >> x = 2:8
      x = 2 3 4 5 6 7 8

Matlab Tutorial

- Some basic operators for matrix:
  - **size(A)**: get the size of A
  - **abs(A)**: get the absolute values of all elements in A
  - **find(A)**: find the positions of non-zero elements in A
  - **max(A)**:
  - **min(A)**:
  - **mean(A)**:
  - **sum(A)**:
  - **sort(A)**:
  - **cat(2, A, B)**: concatenate matrix A and matrix B on 2nd dimension
    - >> A = [1 2; 3 4]; B = [5 6; 7 8]; cat(2, A, B)
      ans = 1 2 5 6
      3 4 7 8
    - same results can be gotten by >> [A, B]
  - please see the Matlab documentations for details

Matlab Tutorial

- special matrices:
  - **eye(n)**: to create a n x n identity matrix
  - **zeros(n1, n2)**: to create a n1 x n2 zero matrix
  - **ones(n1, n2)**: to create a n1 x n2 matrix with all elements = 1
  - **rand(n1, n2)**: to create a n1 x n2 matrix with all elements randomized b/w 0 to 1

- matrix reshape:
  - reshape: first parameter is the input matrix, the others are the size of target matrix

```
>> A = reshape(2:13, 3, 4)
    ans  = 2    5    8   11
           3    6    9   12
           4    7   10   13
```

Matlab Tutorial

- Flow control:
  - **if** condition:

```
if conditional_expression
        execute_command;
end
```

  - **if else** condition:

```
if conditional_expression_1
    execute_command_1;
elseif conditional_expression_2
    execute_command_2;
else
    execute_command_3;
end
```

Matlab Tutorial

- Flow control:
  - **for** loop:

```
for var = expression
   execute_command;
end
```

```
>> x=0;
for i = 1:5
    x(i) = i^2;
end
x = 1     4     9    16    25
```

  - **while** loop:

```
while expression
   execute_command;
end
```

Matlab Tutorial

- Flow control:
  - **switch** condition:

```
switch switch_expr
    case case_expr,
      statements
    case {case_expr1, case_expr2, case_expr3,...}
      statements
    otherwise,
      statements
end
```

  - **break**, **continue**, **return** (usually used in the function)

Matlab Tutorial

- Scripts files and function files:
  - **scripts** file:
    - sequentially execute all the statements in the script file
    - only call by its name, no input parameters
  - **function** file:
    - start with:

function [output_variables] = Name_of_function(input_variables)

```
function bmi = computeBMI(mass_in_kg, height_in_m)
    bmi = mass / height^2;
```

Matlab Tutorial

- Some functions for plotting and image read/write/show will be parts of the exercise.