* Octave
  + [GNU Octave](https://octave.org/) (download/ installation method is shown in the webpage)
* simple tutorial guide of using MATLAB / OCTAVE
* Lession1: basic matrix and vectors
  + Matrices and vectors manipulation, in MATLAB/OCTAVE , following these tests to learn the basic skills.
  + In windows make a directory d:\octave\_test
  + Start matlab or octave , in the command window
  + cd d:\octave\_test   % change directory to d:\octave\_test
  + mkdir test1    %make a directory called test1
  + cd test1          %change directory to test1
  + online interactive tests , you can type in your commands after  ">", e.g.
    - %1)--create  a matrix-----------------------------------------------
    - >a=[1 3 5 ]
    - %This is to create a one row matrix with 3 columns (elements), you will see
    - a
    - 1 3 5
    - %2)--transpose a matrix------------------------------------------------
    - >a'
    - you will see
      * 1
      * 3
      * 5
    - a' will transpose the matrix into a one column matrix, since it has one column, so it can be called a vector
    - %3)--copy a matrix ------------------------------------------------
    - >b=a'
    - copy a' to b
    - you will see,
      * 1
      * 3
      * 5
    - So b is now a vector too.
    - %4)--Access an element of a matrix---------------------------------
    - >b(3)
    - you will see
    - ans= 5,
    - meaning the third element of the vector has a value of 5
    - %5)-- find sub-mtrix------------------------------------------------
    - Create a 4x4 matrix
    - >A=[3 6 7 4;1 2 4 9;5 7 2 1;1 5 8 2]
    - you will see
    - A =
    - 3 6 7 4
    - 1 2 4 9
    - 5 7  2 1
    - 1 5 8 2
    - You can find a sub-matrix inside "A" by typing
    - >SUB\_A=A([1:3,2:3])
    - you will see
    - SUB\_A=
    - 6 7
    - 2 4
    - 7 2
    - For A([1:3,2:3]), you define the range inside the brackets [ ], such as [range\_of\_rows, range\_of\_columns]
    - Here 1:3 means from 1 to 3, 2:3 from column 2 to 3 etc.
    - You have created a sub-matrix SUB\_A from A.  The range of A that this sub-matrix is copied from is from row 1 to 3 and column 2 to 4.
* Lession2: matrix multiplication
  + a=[1 3 5 ]
  + c=[2 4 6; 7 8 9]
  + you just created a matrix with 2 rows and 3 columns, you will see
  + c=
  + 2 4 6
  + 7 8 9
  + >c'
  + you will see
  + 2 7
  + 4 8
  + 6 9
  + you just transposed the matrix c
  + Now try this
  + >a\*c'
  + you will see: 44   76
  + a is 1-by-3 (1 row 3 columns), c is 2-by-3, hence c' is 3-by-2.
  + so when a\*c' is a matrix multiplication of  (1-by3 or 1x3) times (3-by2 or 3x2) , so the result is [ 44 76 ] that is 1-by-2 (1 row 2 columns).
* Lession2: random number generator
  + >d=rand(3,2)  "generate a matrix of 3 rows 2 columns of random numbers."
  + you will see a matrix of 3 rows and 2 columns of random numbers, each random number has a value from 0 to 1.
  + >help rand "to see  more about this function"
* Lesson 3: Use of inverse to solve a matrix equation
  + A=[3 6 7 4;1 2 4 9;5 7 2 1;1 5 8 2] " A is a 4x4 matrix"
  + B=[3 5 2 6]' "don't forget to type in the (') to transport the matrix [3 5 2 6]. then, "B" is a 4x1 matrix"
  + If A\*X=B, find X. we need to solve X of this linear matrix formula; X has 4 unknowns. the solution is
  + Multiply both side with the inverse (A), so
  + Inverse(A)\*A\*X=Inverse(A)\*A, because Inverse(A)\*A is an indenity matrix I=[1 0 0 0; 0 1 0 0 ; 0 0 0 1 ; 0 0 0 1],  hence
  + X=inv(A)\*B, type in the following command
  + >X=inv(A)\*B
  + you will see the 4 unknowns
  + -5.38806
  + 4.46396
  + -1.58209
  + 0.86567
  + Remember "A" must be a square matrix before the inverse can be found. Use pseudo-inverse if "A" is not a square matrix.
  + >help inv
  + or >help pinv   "to learn more."
* Lesson 4: loops and matrix indexing
  + See http://www.cyclismo.org/tutorial/matlab/control.html
  + %----------------------------------------------------  
    disp('----------part A---------------------------')  
    clear %clear everything to make sure variables do not contain values from previous calculations  
    a=[1 4 2 5 7 3 5 8 2] % a(i)=1 because the first index is 1 not 0  
    b=[2 6 0 4 3 8 9 3 1]  
      
    % a simple test  
    for i=1:1:length(a) %usage of for: for start\_value: step: end\_value  
      c(i)=a(i)-b(i)  
    end  
    c   
    %you will see c=[-1 -2 1 4 -5 -4 5 1]  
    %  
      
    disp('----------part B---------------------------')  
    clear %clear everything to make sure variables do not contain values from previous calculations
  + a=[1 4 2 5 7 3 5 8 2] % a(i)=1 because the first index is 1 not 0  
    b=[2 6 0 4 3 8 9 3 1]  
    a  
    b  
    pause %it is a known bug that "pause" may not work in octave
  + %repeat the test with printout on screen  
    for i=1:1:length(a) %usage of for: for start\_value: step: end\_value  
      i %display the value of index i  
      sprintf('a(%d)=%d',i,a(i)')  
      sprintf('b(%d)=%d',i,b(i)')  
      sprintf('a(%d)-b(%d)=%d',a(i),b(i),a(i)-b(i)')  
      disp('press any key to continue');  
     pause %it is a known bug that "pause" may not work in octave
  + end   
      
    disp('----------part c--shifted-subtraction-------------------------')  
    clear %clear everything to make sure variables do not contain values from previous calculations
  + a=[1 4 2 5 7 3 5 8 2] % a(i)=1 because the first index is 1 not 0  
    b=[2 6 0 4 3 8 9 3 1]  
    offset=2  
    for i=1:1:length(a)-offset   
      c(i)=a(i)-b(i+offset)  
    end  
    a  
    b  
    c %you should see c=[1 0 -1 -3 -2 0 4]
* Lesson 5: sine, cosine, absolute\_value (abs), log/exp  functions
  + sine, cosine:
    - [http://www.ece.rice.edu/~dhj/courses/elec241/matlab.html#functions](http://www.ece.rice.edu/%7Edhj/courses/elec241/matlab.html#functions)
    - <http://www.mathworks.com/help/matlab/ref/sin.html>
  + Absolute value (abs): <http://www.mathworks.com/help/signal/ref/abs.html>
  + log/exp: <http://www.mathworks.com/help/matlab/ref/exp.html>
* Lesson 6: IF-THEN-ELSE
  + - <http://www.mathworks.com/help/matlab/ref/if.html>
* Lesson 7: Play with images , use of image (display images) , imread (read jpg files)
  + >a=[10 48 26 55 37 13 25 48 52]
  + >b=[2 6 0 4 3 8 9 3 1]
  + >c=a’\*b % to create a 9x9 matrix
  + >figure(1) % create a figure called 1
  + > clf %clear the figure first
  + >image(c) %display the matrix in pseudo color
  + >colormap(gray256) %change the display to gray level, the display may change
  + >help imagesc %scale up the gary levels to create a better image
  + >%----------------
  + >img1=imread(‘house\_out.jpg’); %assume you have a jpg called house\_out.jpg in the current dirctory
  + >figure(1) % create a figure called 1
  + > clf %clear the figure first
  + >image(img1) %display the image
* Lesson 8: read sound files, wavread (read sound files)
  + >[x,fs]=wavread(‘sor1.wav’)
  + >plot(x) %plot the sound file
  + >sound(x,fs) >play sound , matlab only, not implemented in octave yet!!
* Lesson 9, random, number
  + >rand(3,4) %to generate a 3x4 matrix of random numbers(range from 0 to 1, mean around 0.5)
  + >randn(3,4) %to generate a 3x4 matrix of normally distributed random numbers (range from -1 to 1, mean around 0)
* Lesson 10, keyboard input and for command building and text display
  + >help input %(for keyboard input ) to learn how to use this command
  + >help ginput %(for mouse input ) to learn how to use this command
  + >help sprint % for how to use text display
* Note:
  + If the display window cannot show all results, it will just show part of it and enter a mode that with a prompt ':'. The system is waiting for your input. Use 'enter' (for one line) or 'space' (for a page) to scroll down the display. If you want to quit, just type 'q'.