# MATLAB Tutorial

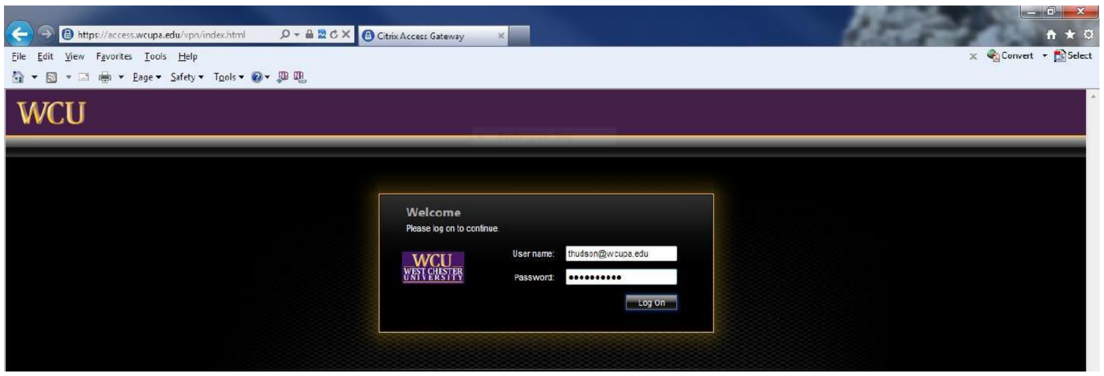
**Primary Author: Shoumik Chatterjee**

**Secondary Author: Dr. Chuan Li**

# Table of Contents

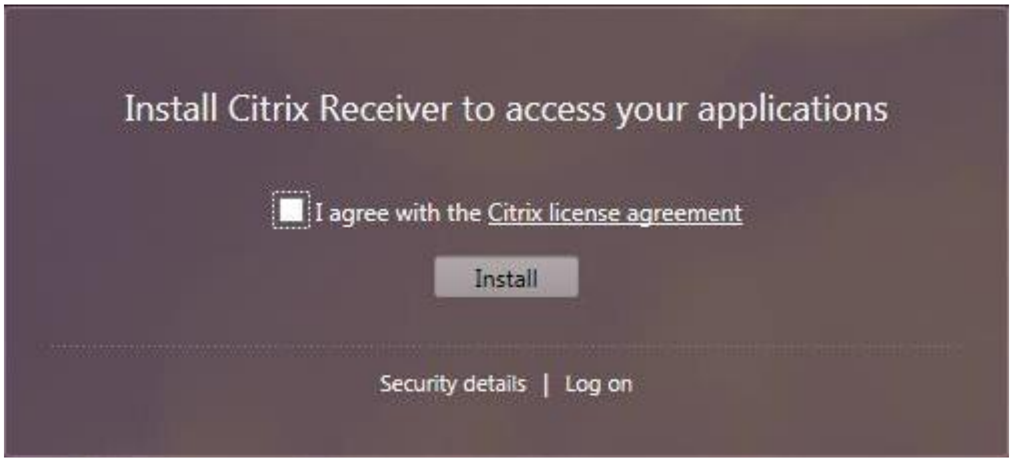
## Section 1: Accessing MATLAB using RamCloud server

1. Open Web Browser and go to RamCloud.wcupa.edu.



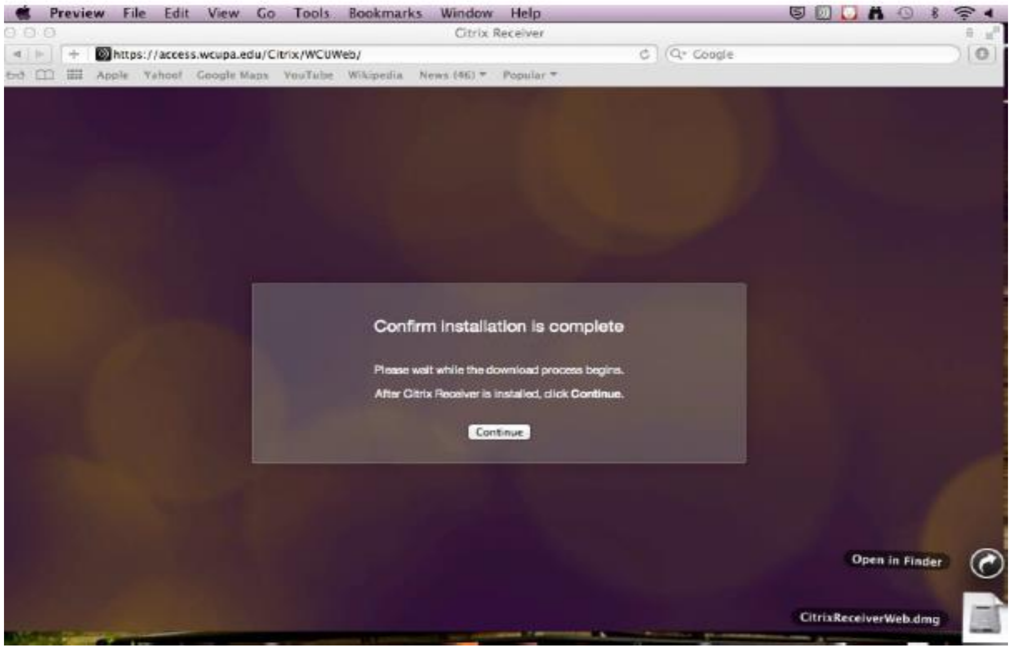
https://access.wcupa.edu/vpn/index.html
Citrix Access Gateway
File Edit View Favorites Tools Help
Page Safety Tools
Convert Select
WCU
Welcome
Please log on to continue
WCU WEST CHESTER UNIVERSITY
User name: thudson@wcupa.edu
Password: ••••••••••
Log On

2. Login with WCU username and password.
3. Click checkbox to agree to Citrix license agreement and click Install.



Install Citrix Receiver to access your applications
I agree with the Citrix license agreement
Install
Security details | Log on

4. Click the downloads icon on the MAC desktop and click the CitrixReceiverWeb.dmg file, which will move it to the desktop.

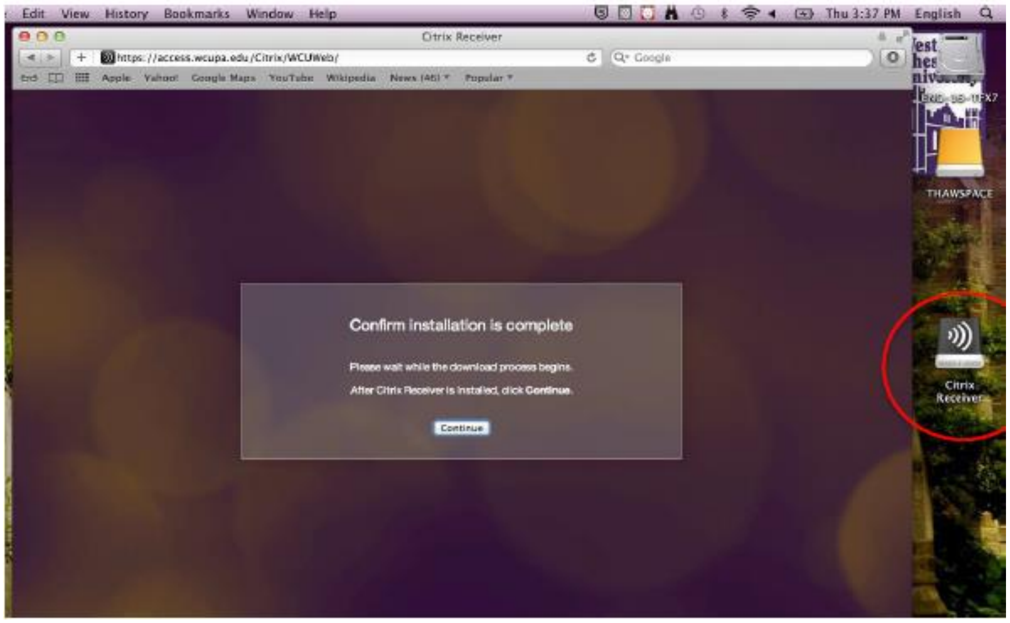

Preview File Edit View Go Tools Bookmarks Window Help
Citrix Receiver
https://access.wcupa.edu/Citrix/WCUWeb/
Open in Finder
CitrixReceiverWeb.dmg

Confirm installation is complete

Please wait while the download process begins.
After Citrix Receiver is installed, click **Continue.**

Continue

5. Click the Citrix Receiver icon to install, and click continue once installed.

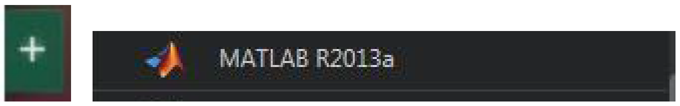

Edit View History Bookmarks Window Help
Thu 3:37 PM English
Citrix Receiver
https://access.wcupa.edu/Citrix/WCUWeb/
THAWSPACE
Citrix Receiver

Confirm installation is complete

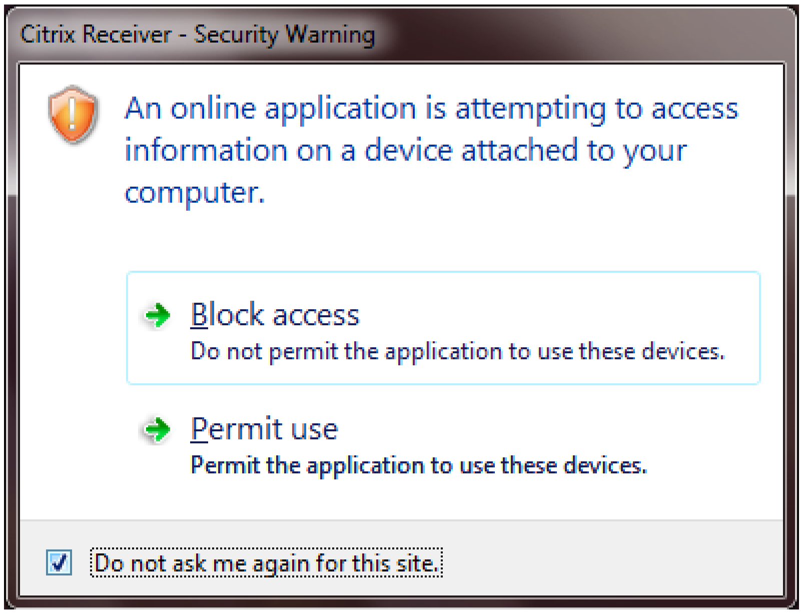
Please wait while the download process begins.
After Citrix Receiver is installed, click **Continue.**

Continue

6. If necessary, refresh browser and login again from RamCloud.wcupa.edu.
7. Click plus sign on the left side, select All APPS, and double-click the MATLAB icon.



8. If security warnings appear, click permit use/access.


Citrix Receiver - Security Warning

An online application is attempting to access information on a device attached to your computer.

Block access
Do not permit the application to use these devices.

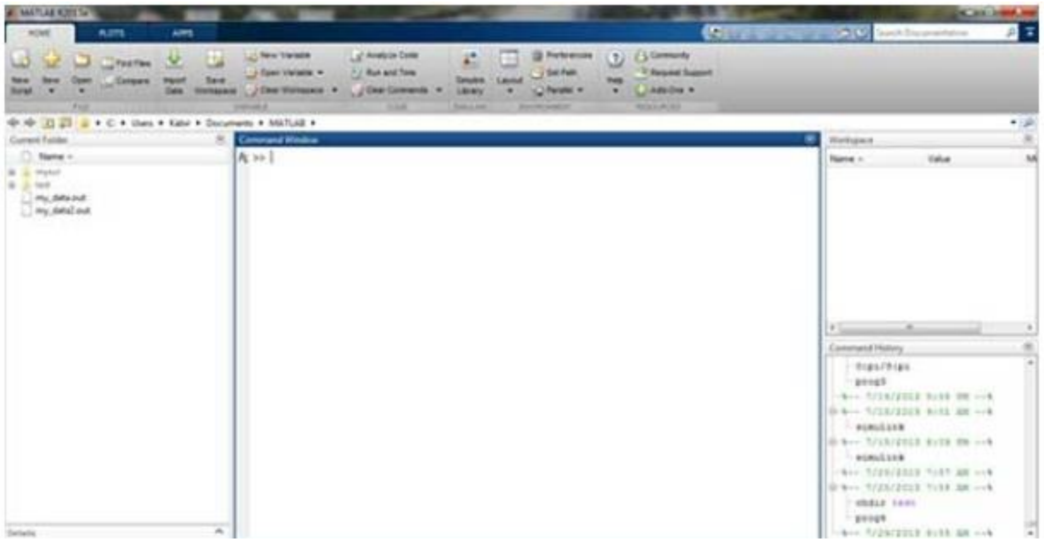
Permit use
Permit the application to use these devices.

☑ Do not ask me again for this site.

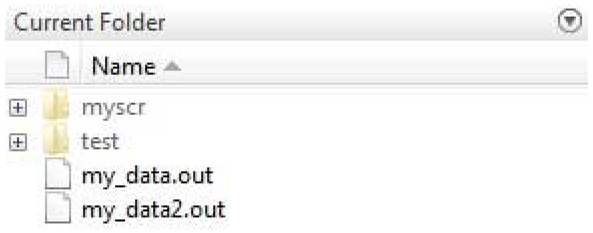
9. Save all files to local computer hard drive by clicking File, Save As, Computer, C: drive, etc.

**Section 2: MATLAB GUI Basics**

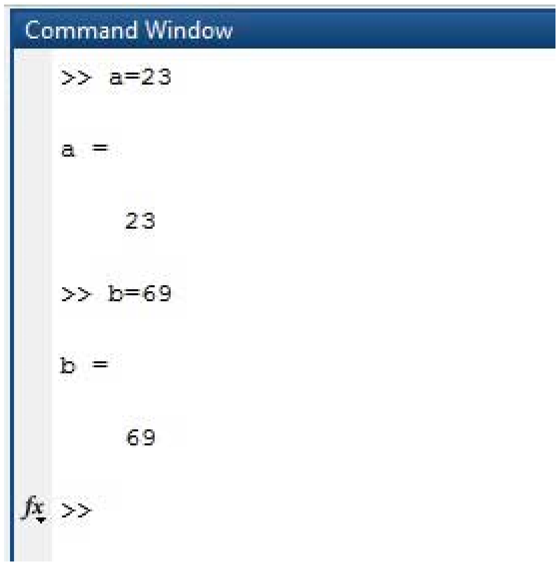
1. When MATLAB is opened, the following desktop appears (in some instances, the two rightmost panels may be located as separate tabs underneath the left panel):



2. The desktop has four panels:
   i. The left panel is the current folder and allows you to access the project folders and files.

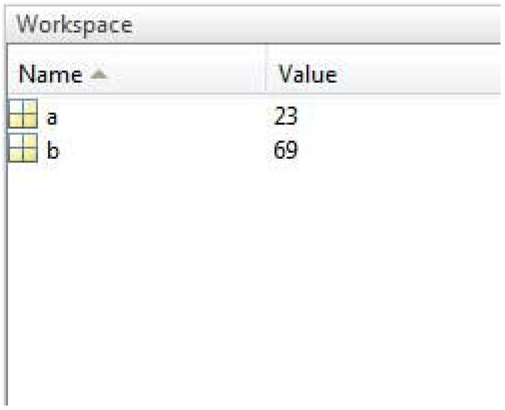


ii. The middle panel is the **Command Window** and it is where commands can be entered at the command line. It is indicated by the command prompt (>>).



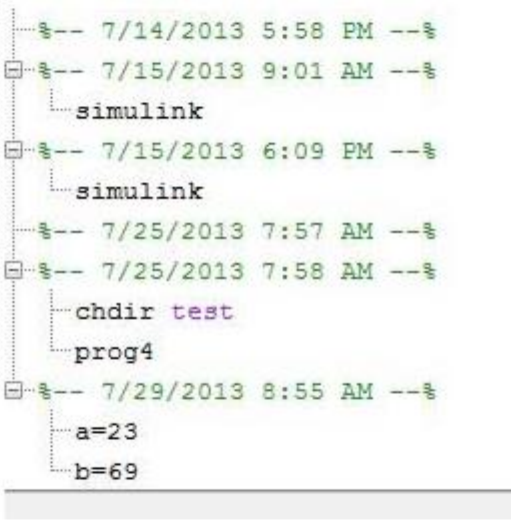
```
>> a=23

a =

    23

>> b=69

b =

    69

>>
```

iii. The panel in the top right is the workspace and shows all the variables created and/or imported from files.



| Name | Value |
|---|---|
| a | 23 |
| b | 69 |

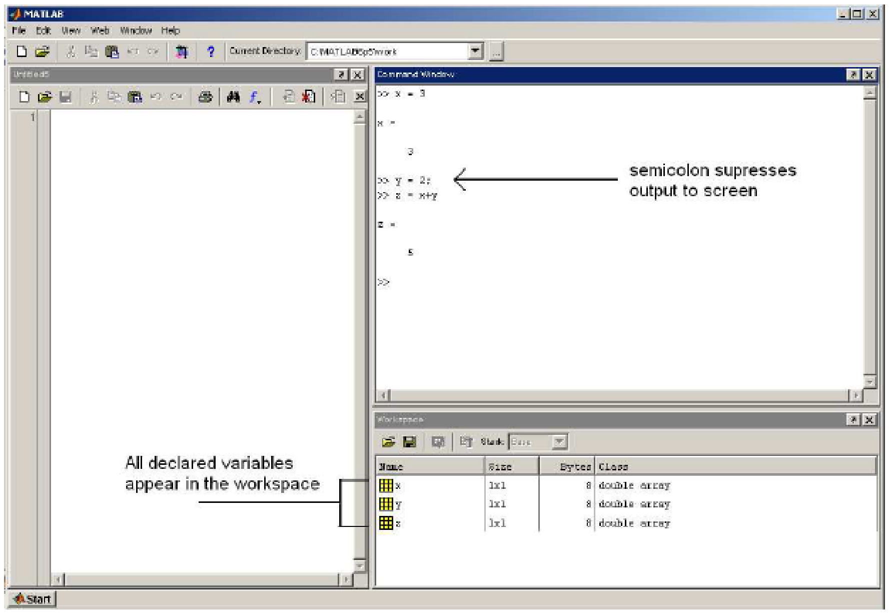
iv. The panel in the bottom right is the command history and shows or rerun commands that are entered at the command line.



```
%-- 7/14/2013 5:58 PM --%
%-- 7/15/2013 9:01 AM --%
   simulink
%-- 7/15/2013 6:09 PM --%
   simulink
%-- 7/25/2013 7:57 AM --%
%-- 7/25/2013 7:58 AM --%
   chdir test
   prog4
%-- 7/29/2013 8:55 AM --%
   a=23
   b=69
```

## Section 3: MATLAB Commands

1. How to enter simple commands and operations:
   - i. Commands are entered in the command window.
   - ii. The result of a variable or operation appears when Enter is pressed.
   - iii. To suppress the result of a variable or operation, enter ";" at line end.


MATLAB
File Edit View Web Window Help
Current Directory: C:\MATLAB6p5\work
Command Window
Workspace
Stack: Base
semicolon supresses output to screen
All declared variables appear in the workspace
Start

```
>> x = 3

x =

     3

>> y = 2;
>> z = x+y

z =

     5

>>
```

| Name | Size | Bytes | Class |
|---|---|---|---|
| x | 1x1 | 8 | double array |
| y | 1x1 | 8 | double array |
| z | 1x1 | 8 | double array |

2. How to enter matrices:
   - i. MATLAB stores numerical results as matrices (even a number is 1 X 1 matrix).
   - ii. Use "[" to denote the start of a matrix and "]" to denote the end of the matrix.
   - iii. Use blank spaces to separate row entries.
   - iv. Use semicolons (;) to separate columns.

```
a = [ 1 2 3 4 5; 2 3 4 5 6; 3 4 5 6 7; 4 5 6 7 8]
```

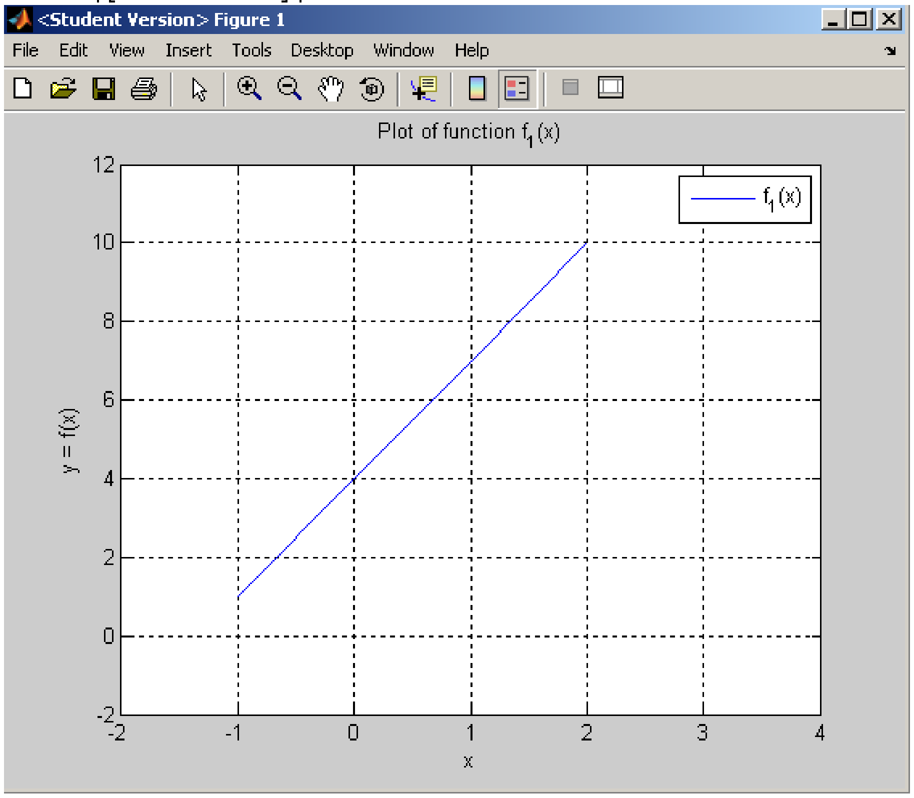
MATLAB will execute the above statement and return the following result

```
a =
      1      2      3      4      5
      2      3      4      5      6
      3      4      5      6      7
      4      5      6      7      8
```

## Section 4: MATLAB Plots

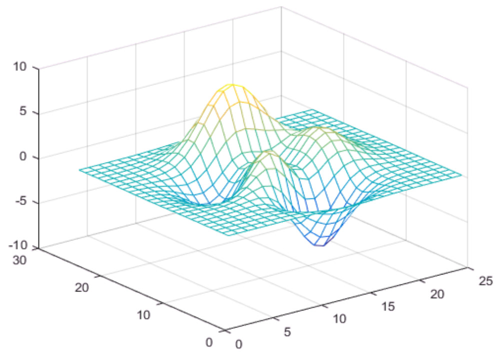
1. How to plot 2-D function:
   - i. Create a vector of x-values to evaluate the function.
   - ii. Write equation for y in terms of x.
   - iii. Include the figure keyword and the plot (x,y) function.
   - iv. Include the xlabel('x') and ylabel('y = f(x)') to label the axes.
   - v. Include grid on so grid appears.
   - vi. Include title (' ') so title appears on top of graph.
   - vii. Include legend (' ') so that the legend is shown.
   - viii. Use axis([ ]) so that the axis is set to the desired range.

```
x = [-1 0 1 2];
y = 3*x + 4;
figure
plot(x,y)
xlabel('x')
ylabel('y = f(x)')
grid on
title('Plot of function f_1(x)')
legend('f_1(x)')
axis([-2 4 -2 12])
```

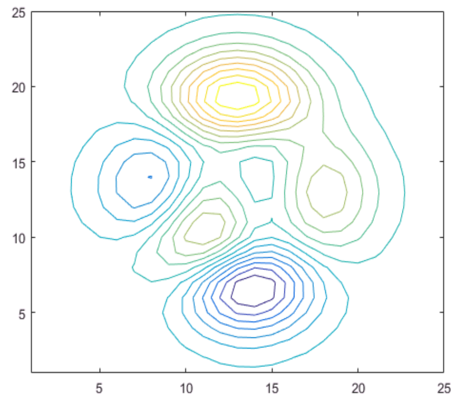

<Student Version> Figure 1
File Edit View Insert Tools Desktop Window Help
Plot of function $f_1(x)$
$f_1(x)$
y = f(x)
x
12
10
8
6
4
2
0
-2
-2
-1
0
1
2
3
4

2. How to plot 3-D function:
   i. Set up vectors that represent the range of *x* and *y* values, in the following manner: x = $x_{min}$:$x_{increment}$:$x_{max}$;   y = $y_{min}$:$y_{increment}$:$y_{max}$;
   ii. Make a grid of points over which to evaluate the heights of the surface, via the meshgrid function, as follows:   [X,Y] = meshgrid(x,y);
   iii. Express z as a function of x and y.
   iv. To make a surface plot, type surf(X,Y,Z).
   v. To make a contour plot, type contour(X,Y,Z,N), where N is number of contour levels.

```
 x = [0:5:25];              y = [0:5:30]);
[X,Y] = meshgrid(x,y);
z = x.^2 - y.^2;
surf(X,Y,Z)
```

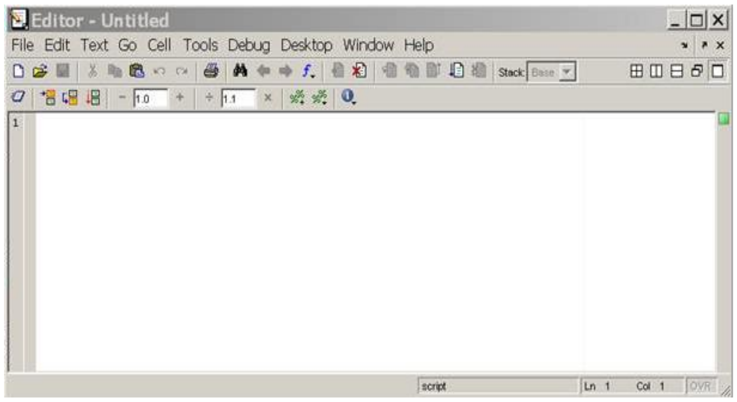


```
contour(X,Y,Z,10)
```



## Section 5: MATLAB Script Files

1. Type edit or edit <filename>.m in the command window to open editor.



2. Type code in editor window.

```
NoOfStudents = 6000;
TeachingStaff = 150;
NonTeachingStaff = 20;
Total = NoOfStudents + TeachingStaff ...
   + NonTeachingStaff;
disp(Total);
```

3. Save editor file (1.e. File->Save).
4. Type filename (without .m extension) in command window (NOT editor) to run script file.
5. Command window displays result(s) of script file.

```
6170
```

## Section 6: MATLAB Functions

1. Functions allow the user to reuse sequences of commands by storing them in program files.
2. The format of a MATLAB functions is as follows:

```
function [output1,…,outputN] = functionName (input1,…,inputM)
        MATLAB commands
output1 = …
…
outputN = …
end
```

```
function a = triarea(b,h)
a = 0.5*(b.*h);
end
```

3. Call function from command line with different input parameters.

```
a1 = triarea(1,5)
a2 = triarea(2,10)
a3 = triarea(3,6)
```

4. Command window displays function results.

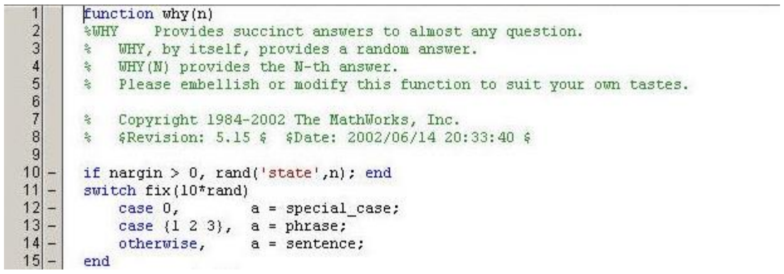
```
a1 =
    2.5000
a2 =
    10
a3 =
     9
```

5. Commonly used commands are as follows:
   - dir- lists all files in current directory
   - format- controls screen display format
   - help- searches for help topic
   - quit- stops MATLAB

6. Commonly used (built-in) functions are as follows:
   - det – matrix determinant
   - inv – matrix inverse
   - eig - matrix eigenvalues
   - rank – matrix rank

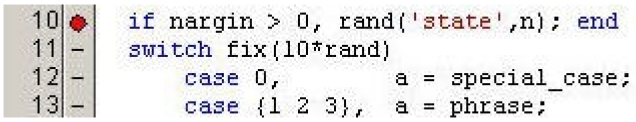
**Section 7: Debugging MATLAB Scripts**

1. If MATLAB script is producing result(s) which are not correct or are unexpected, it is oftentimes useful to stop the script execution in the middle to isolate the specific variable(s) causing problem(s).
2. To stop program execution, implement a breakpoint at the start of the line by clicking the horizontal line immediately to the right of the line number on the command window.



```
 1    function why(n)
 2    %WHY    Provides succinct answers to almost any question.
 3    %   WHY, by itself, provides a random answer.
 4    %   WHY(N) provides the N-th answer.
 5    %   Please embellish or modify this function to suit your own tastes.
 6
 7    %   Copyright 1984-2002 The MathWorks, Inc.
 8    %   $Revision: 5.15 $  $Date: 2002/06/14 20:33:40 $
 9
10 -  if nargin > 0, rand('state',n); end
11 -  switch fix(10*rand)
12 -      case 0,          a = special_case;
13 -      case {1 2 3},    a = phrase;
14 -      otherwise,       a = sentence;
15 -  end
```

Implementing a breakpoint in front of line 10 stops the program in front of line 10.



```
10 ●  if nargin > 0, rand('state',n); end
11 -  switch fix(10*rand)
12 -      case 0,          a = special_case;
13 -      case {1 2 3},    a = phrase;
```

3. Once a breakpoint is implemented, run the program

```
why(37)
```

4. Once the command prompt appears (K>>), type a variable and press enter.

```
n
```

5. If the value returned by MATLAB is not correct, the error (bug) is located before the breakpoint.