

# Basic Understanding of MATLAB

Michael Dow

Design Team 8

11/07/2008

## **Abstract**

Many problems, when it comes to math and basic theory of engineering, can be solved using MATLAB. MATLAB gives the ability to solve these equation before they are implemented into an actual product or used in the real world. MATLAB can range from basic equation solving, plots, toolboxes and real-time simulink. This will give a basic understanding of how to use MATLAB with equations and plotting graphs.

## **Objective**

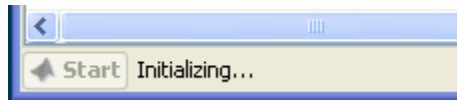
The Objective of this application note is to show the reader how to start to use MATLAB and give a basic understanding of simple commands.

## **Introduction**

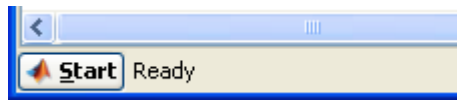
MATLAB requires one to have a basic understanding of programming. The most closely related type of code is C and C++. Majority of the time the code that is formed in MATLAB is not going to be the same that will be implemented into an actual product. MATLAB is just supposed to give you an idea of the right direction that one must go to implement it into the product that they are using. MATLAB is a very powerful tool, but it can only go as far as the user will take it.

## **Getting Started**

MATLAB must first be installed on the computer one will be using. If it is currently not installed, a trial version can be downloaded before a full version is bought. The trial version can be downloaded from MATLAB's website, [http://www.mathworks.com/programs/trials/trial\\_request.html?s\\_cid=cta\\_trials](http://www.mathworks.com/programs/trials/trial_request.html?s_cid=cta_trials). Once MATLAB is installed, open the main application. The MATLAB main window will open and in the bottom left corner you will see Initializing...

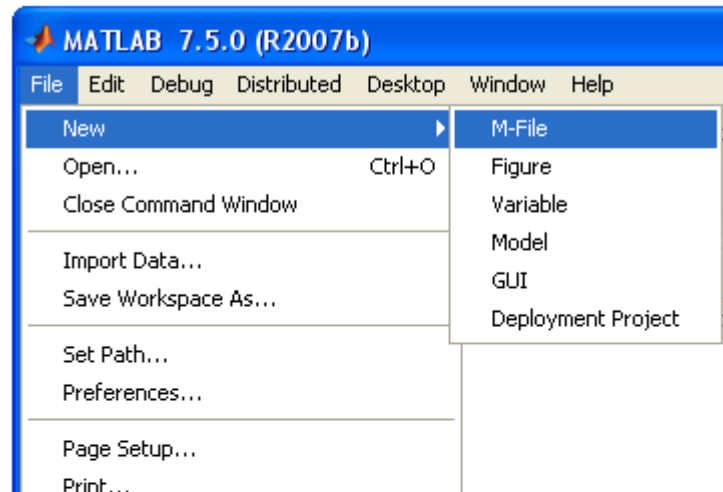


The application will not be ready to use until the bottom left corner displays Ready.



## **Creating a MATLAB file**

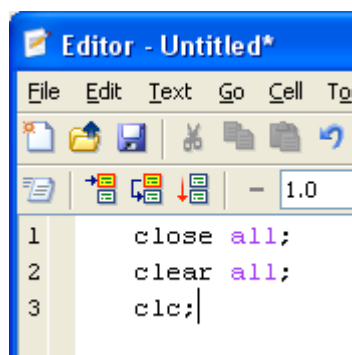
MATLAB has a Command Window that equations and data can be directly entered into. The only issue with directly entering data into the command window is that it is not saved anywhere and if you need to change one piece of information everything needs to be retyped and that is a huge hassle. The command window is great if one needs to see what data is stored in what variable easily. The best way to store a list of commands is to create an M-File. This way to achieve this is by going to File → New → M-File.



This will open up an editor window where the MATLAB code can be entered.

### **MATLAB Code**

It is recommended to start all MATLAB file with three commands close all; clear all; and clc;. These three command make it so past data is not linked with the new file. The command close all; closes all open MATLAB figure windows, the command clear all; clears all data stored to a variable and the command clc; clears the command window, just so everything looks nice.

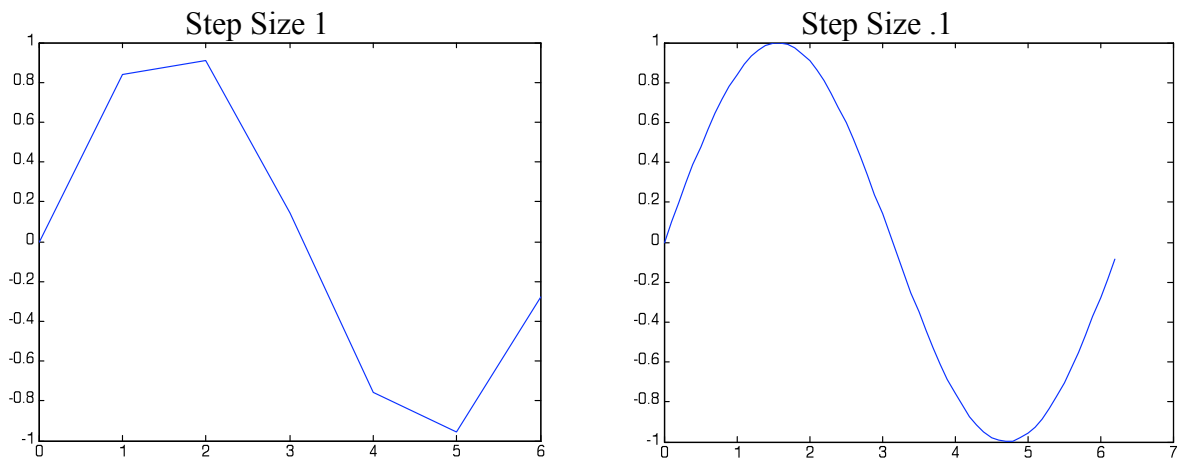


Now that this is done, the actual code that we are trying to solve can be started. A sine wave can easily be plotted using a series of basic commands.

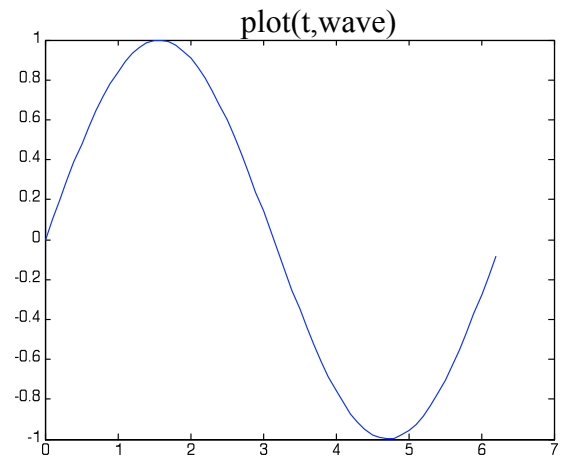
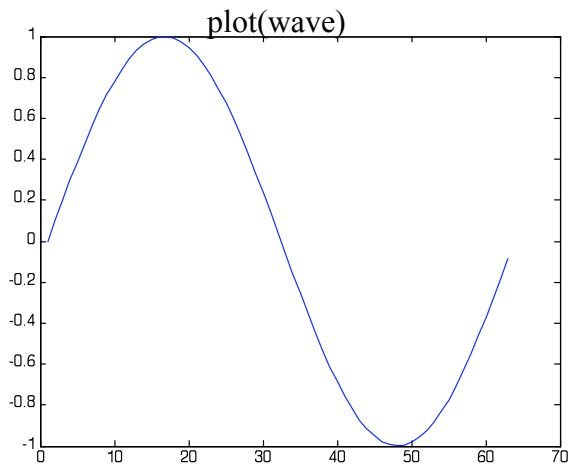
```
close all;  
clear all;
```

```
clc;  
t = 0:.1:2*pi;  
wave = sin(t);  
plot(t, wave);
```

The function we are dealing with is  $\sin(x)$ . Therefore, the first step is to declare what the range is for  $x$ , in this example the name of the variable is called  $t$ , and since  $t$  is a range with multiple values, it becomes an array. The range of  $t$  is from 0 to  $2\pi$  and  $.1$  is the step size, if no step size is declared the value is defaulted to 1 and that can cause a dramatic difference.

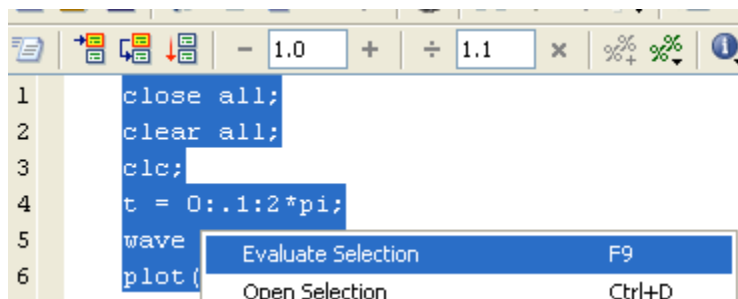


Next the range is taken and input into the  $\sin$  function. The answer to the  $\sin$  function in this situation is called  $wave$  and  $wave$  will have the answers to each value of  $t$  for the  $\sin$  function. To plot the function, we can use the  $\text{plot}$  command. There are many features of the  $\text{plot}$  command, one key feature is using the variables in the right order to receive the proper axis values. These are two examples of the  $\text{plot}$  command. The  $\text{plot}(wave)$  command just uses the values of  $wave$ , which is a  $\sin$  function but the  $x$  axis is the value in the array that it is outputting which means nothing in correlation to  $t$ . The  $\text{plot}(t, wave)$  plots  $wave$  with respect to  $t$ , thus giving us the right data we want.

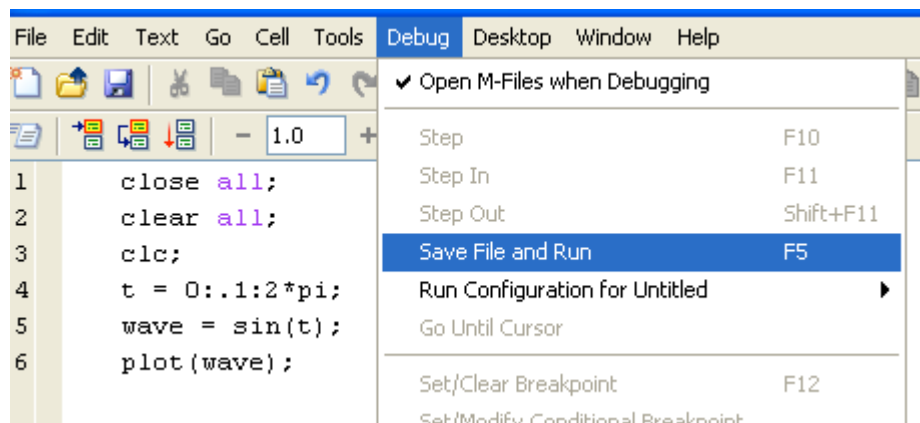


## Running the Code

There are a few different ways to run the code after you finish typing it up. The first way is to highlight the code, right-click it, and select Evaluate Selection or F9. This can be done whether or not the file has been saved.



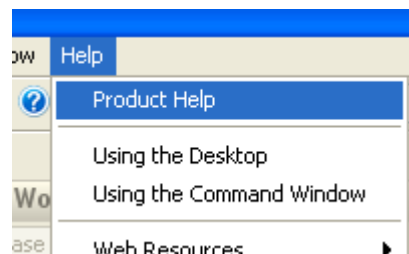
Once the file is saved, one can simply hit F5, to run the command Save File and Run or simply go to Debug → Save File and Run.



Once the code is ran a window will pop-up with the plot you have created.

## **MATLAB Help**

There are two way of getting more information on functions and using MATLAB. The first way is to go to the command window and type 'help' and then the function name, e.g. help plot. Once this is done writing will display in the command window with data on that particular function. Another way is to use the built in help window by going to HELP → Product Help.

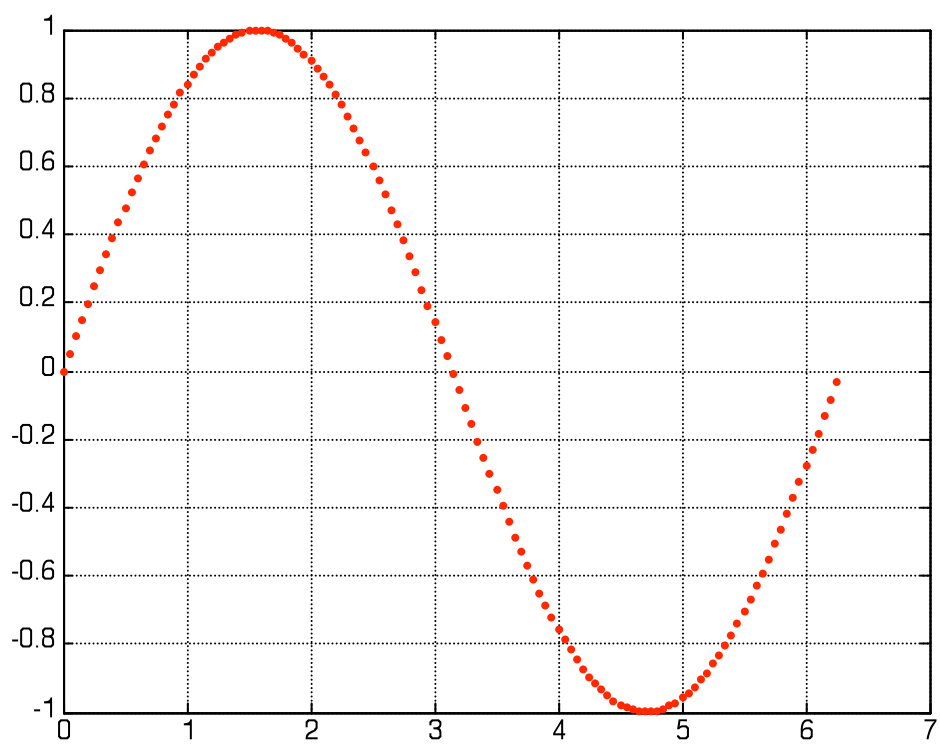


Once the Help window opens there are four different options to choose from to find the answers needed, including search and demos.

## **MATLAB Examples**

Example 1.

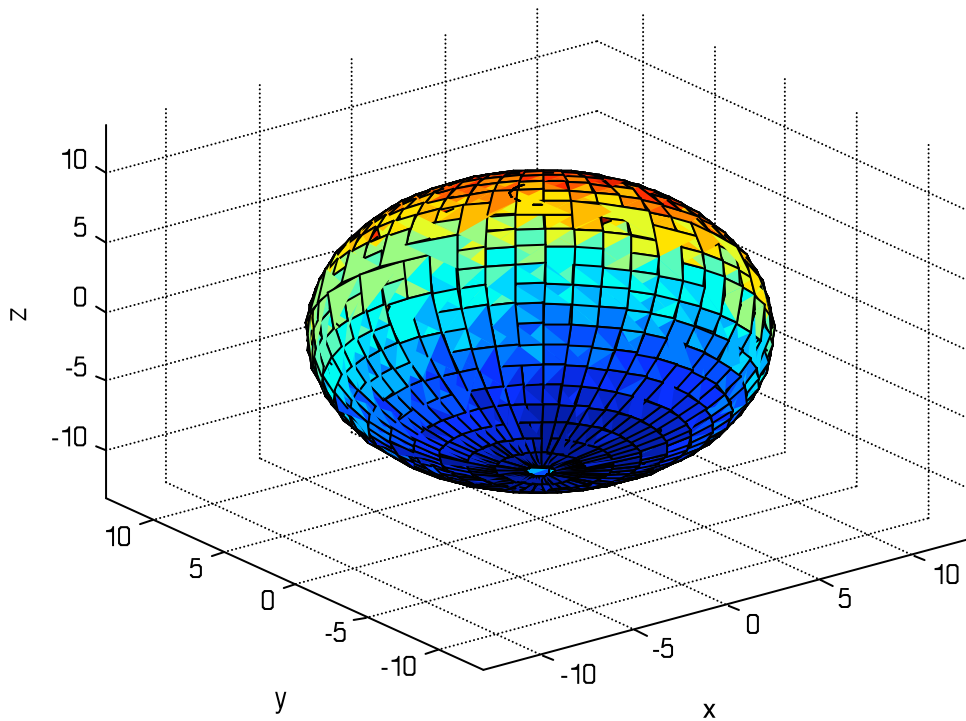
```
close all;  
clear all;  
clc;  
i = 1;  
for t=0:.05:2*pi;  
    x(i) = t;  
    y(i) = sin(t);  
    i = i + 1;  
end  
figure(1)  
plot(x,y,'r')  
grid
```



Example 2.

```
close all;  
clear all;  
clc;  
t2 = [0,pi];  
ezsurf('(\sqrt{10^2 - (10*cos(t2))^2})*cos(t)', '(\sqrt{10^2 - (10*cos(t2))^2})*sin(t)', '10*cos(t2)')
```

$$x = (\sqrt{10^2 - (10 \cos(t_2))^2}) \cos(t), \quad y = (\sqrt{10^2 - (10 \cos(t_2))^2}) \sin(t), \quad z = 10 \cos(t_2)$$



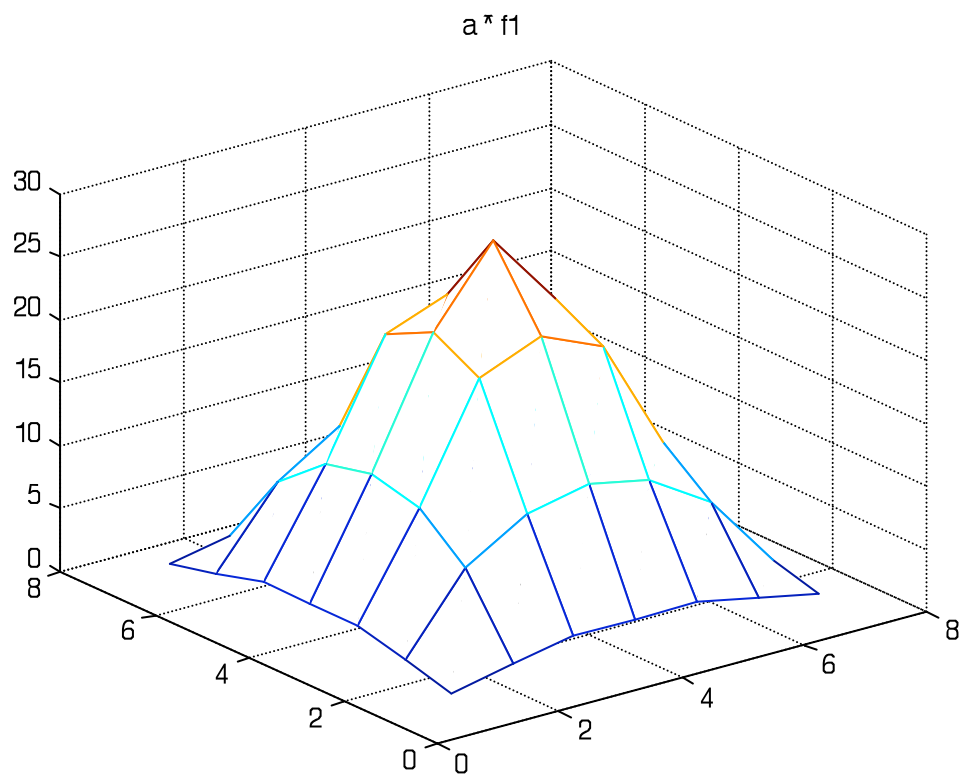
Example 3.

```
clc  
clear all  
close all  
a = [1 1 1 1 1 ; 1 2 2 2 1 ; 1 2 3 2 1 ; 1 2 2 2 1 ; 1 1 1 1 1];  
f1 = [1 1 1 ; 1 4 1 ; 1 1 1];  
f2 = [-1 -1 -1 ; -1 9 -1 ; -1 -1 -1];  
C1 = conv2(a,f1);  
C2 = conv2(a,f2);  
C = conv2(C1,f2);  
mesh(C1)  
title 'a * f1';  
figure;  
mesh(C2)
```

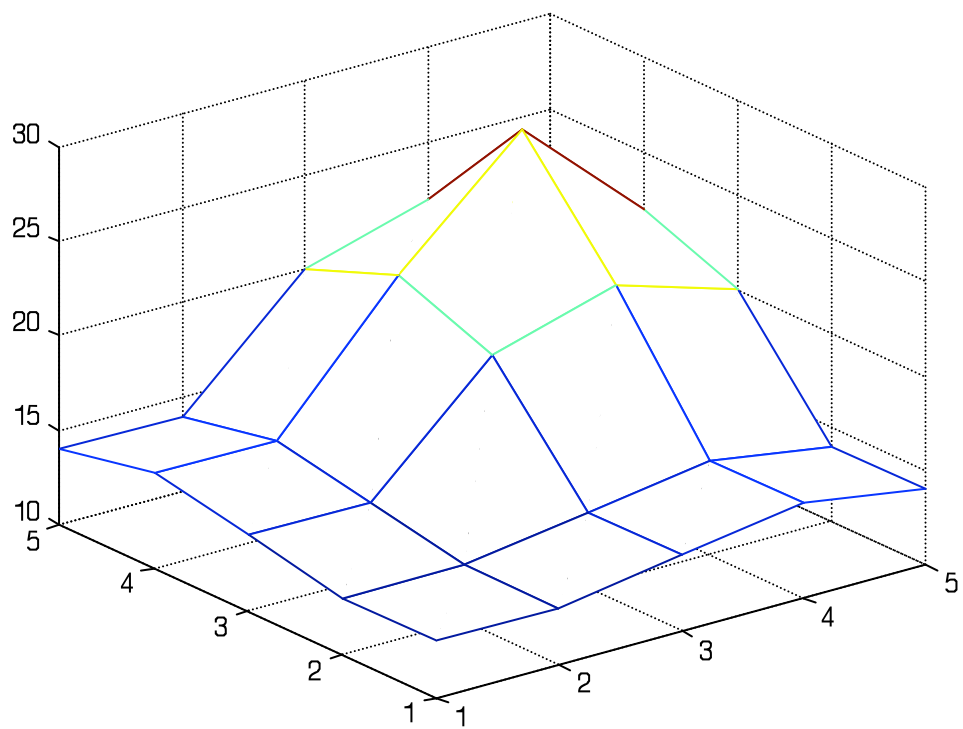
```

title 'a * f2';
a = [1 1 1 1 1 ; 1 2 2 2 1 ; 1 2 3 2 1 ; 1 2 2 2 1 ; 1 1 1 1 1];
f1 = [1 1 1; 1 4 1; 1 1 1];
f1(5,5) = 0;
g1 = f1;
f2 = [-1 -1 -1; -1 9 -1; -1 -1 -1];
f2(5,5) = 0;
g2 = f2;
sol1 = ifft2(fft2(a).*fft2(g1));
sol2 = ifft2(fft2(a).*fft2(g2));
mesh(sol1)
title 'ifft2(fft2(a)fft2(f1))'
figure;
mesh(sol2)
title 'ifft2(fft2(a)fft2(f2))'

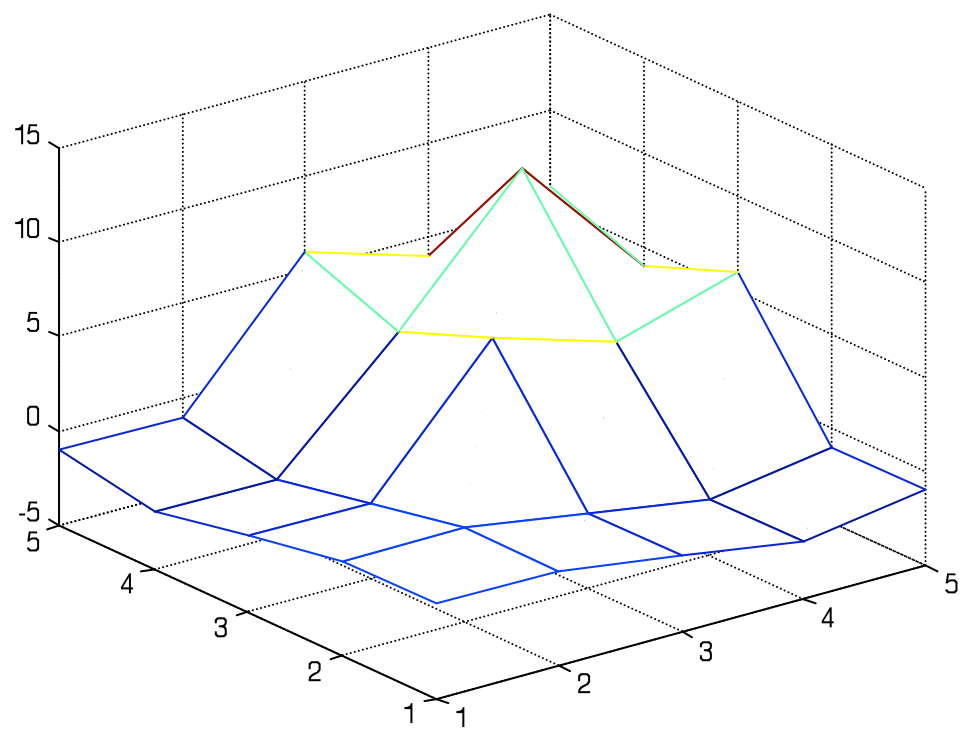
```



$\text{ifft2}(\text{fft2}(a)\text{fft2}(f1))$



$\text{ifft2}(\text{fft2}(a)\text{fft2}(f2))$



## **Conclusion**

This was a very basic introduction to the workings of MATLAB. For someone to explain everything that MATLAB has to offer, one would require an endless amount of time. MATLAB has something for everyone and information on that topic to make it easier for them to solve their problem. From a range of simple mathematics to complex calculus, MATLAB is a tool that does it all.