EE 3350
MATLAB INTRODUCTION

Instructor: Dr. Duncan MacFarlane

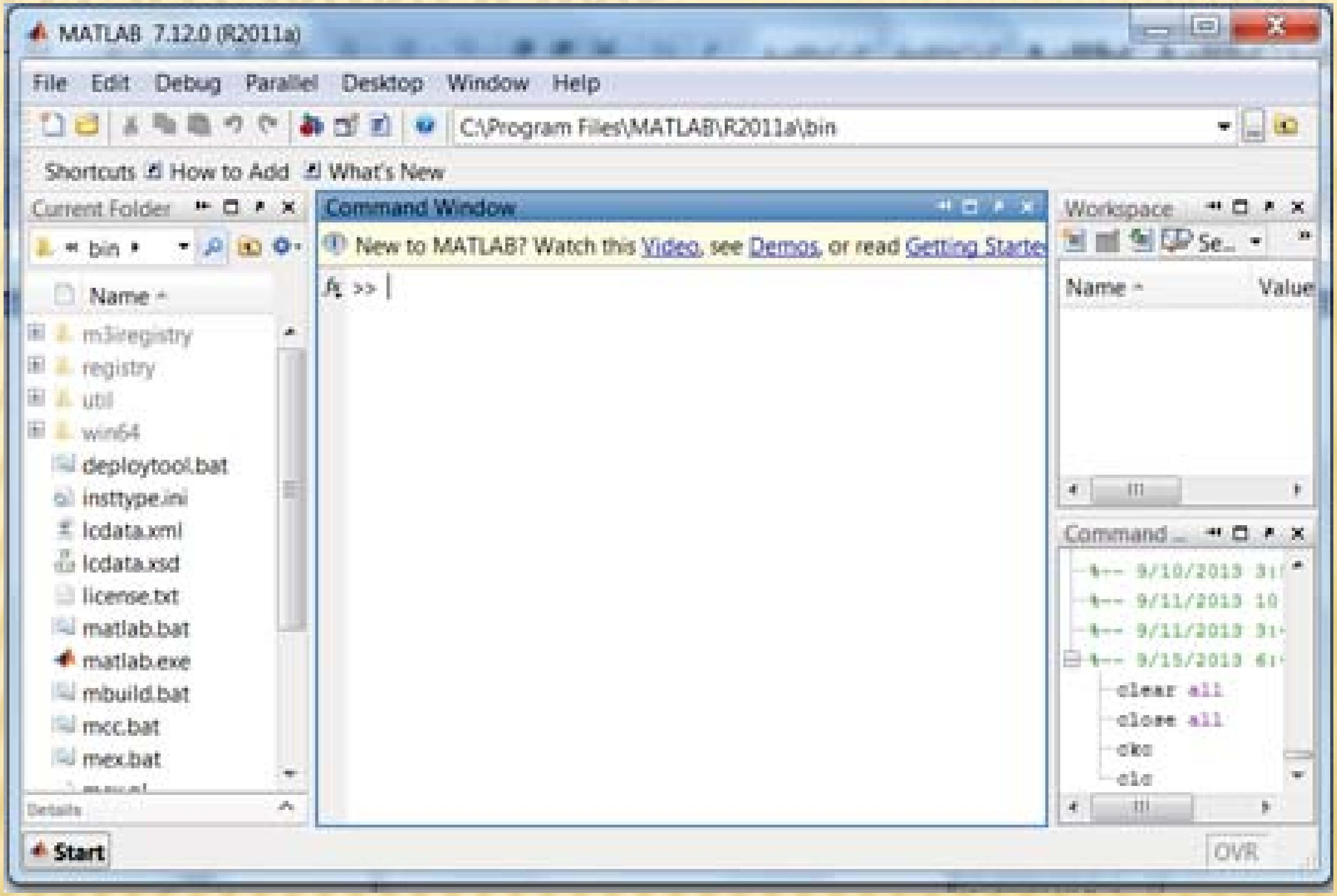
Teaching Assistant: Yuan Hu

UTD

WHAT IS MATLAB?

- MATLAB is a numerical computing environment developed by MathWorks.
- MATLAB allows matrix manipulations, plotting of functions and data, and implementation of algorithms.
- MatLab program and script files always have filenames ending with ".m“.
- The programming language is exceptionally straightforward since almost every data object is assumed to be an array.

MATLAB WINDOWS



C:\Program Files\MATLAB\R2011a\bin

m3iregistry
registry
util
win64
deploytool.bat
insttype.ini
lcdata.xml
lcdata.xsd
license.txt
matlab.bat
matlab.exe
mbuild.bat
mcc.bat
mex.bat

New to MATLAB? Watch this Video, see Demos, or read Getting Starte

```
>> |
```

```
%-- 9/10/2013 3:
%-- 9/11/2013 10
%-- 9/11/2013 3:
%-- 9/15/2013 6:
   clear all
   close all
   ckc
   clc
```

- We can type the commands in the command window.
- As for programming, .m file script is preferred.

File → New → Script

GETTING HELP

- Online help is available from the Matlab prompt (a double arrow)
- **generally** (listing all available commands):

```
>> help
```

[a long list of help topics follows]

- **specific commands:**

```
>> help fft
```

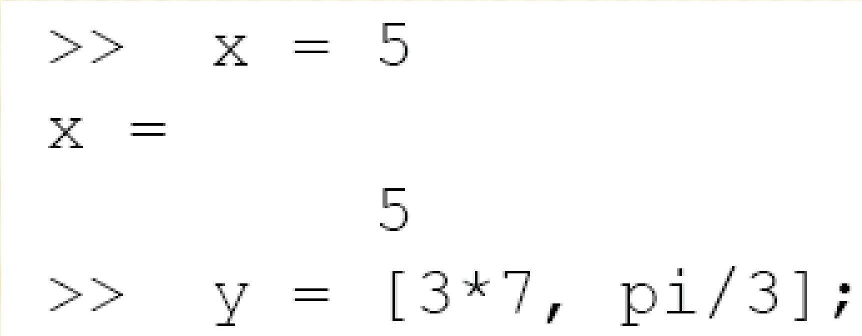
[a help message on the fft function follows].

DATA REPRESENTATIONS IN MATLAB

- **Variables: Variables are defined as the assignment operator “=” . The syntax of** variable assignment is

  variable name = a value (or an expression)

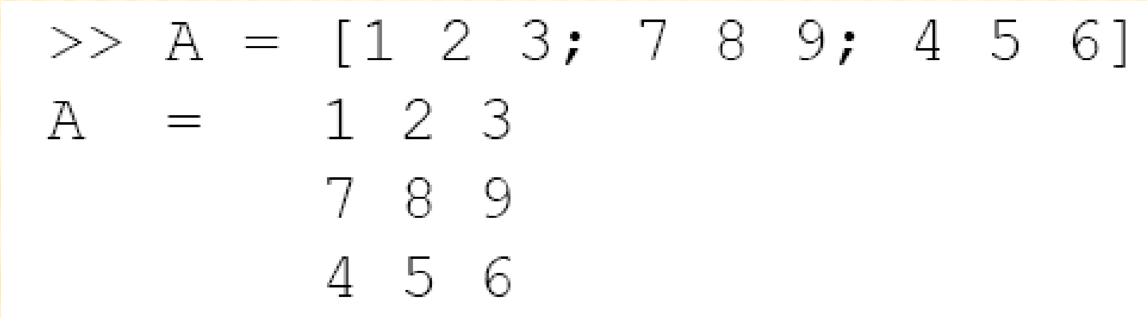
- Example:



```
>>  x = 5
x =
        5
>>  y = [3*7, pi/3];
```

DATA REPRESENTATIONS IN MATLAB

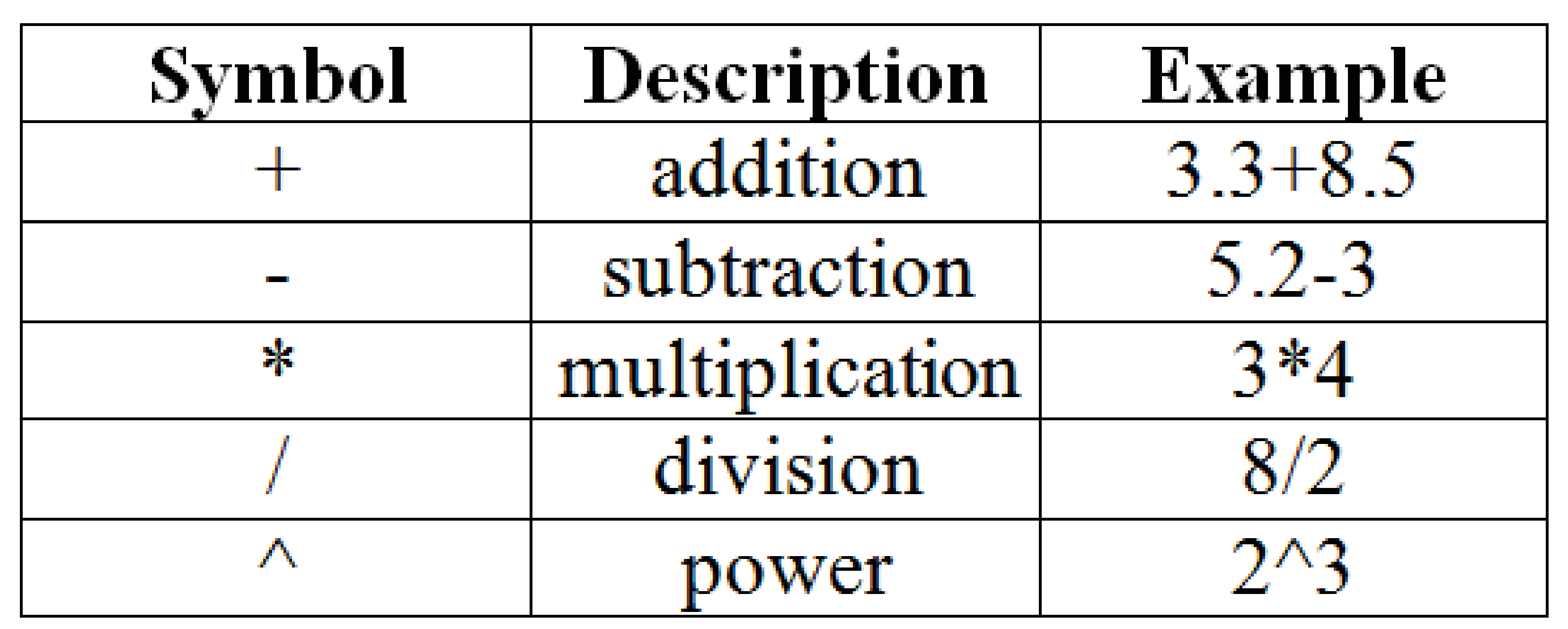
- **Vectors/Matrices :** MATLAB can create and manipulate arrays of 1 (vectors) , 2 (matrices), or more dimensions.
- row vectors: a = [1, 2, 3, 4] is a 1X4 matrix
- column vectors: b = [5; 6; 7; 8; 9] is a 5X1 matrix
- Example:



```
>> A = [1 2 3; 7 8 9; 4 5 6]
A  =    1 2 3
        7 8 9
        4 5 6
```

ARITHMETIC OPERATIONS

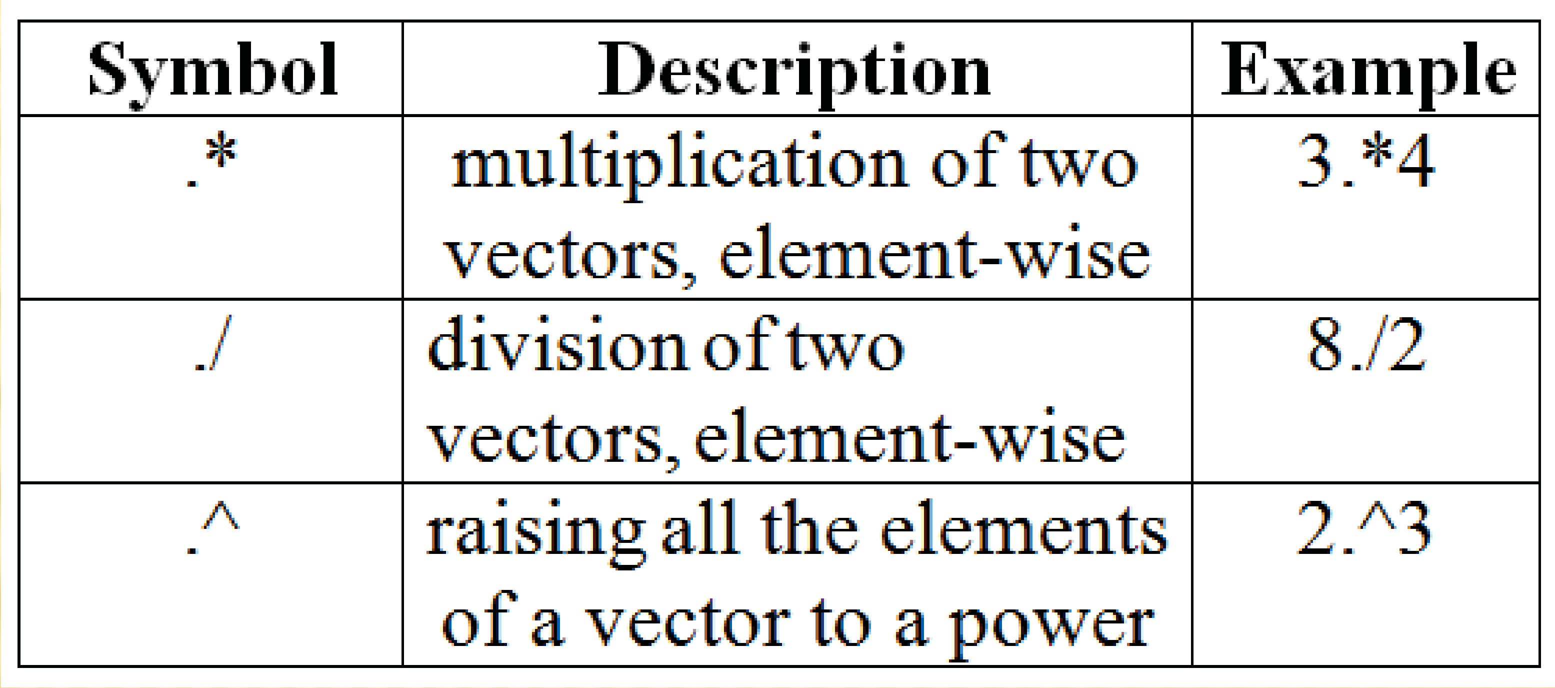
- Five basic arithmetic operators



Symbol	Description	Example
+	addition	3.3+8.5
-	subtraction	5.2-3
*	multiplication	3*4
/	division	8/2
^	power	2^3

ARITHMETIC OPERATIONS

- Three operators working on an element-by element basis.



Symbol	Description	Example
.*	multiplication of two vectors, element-wise	3.*4
./	division of two vectors, element-wise	8./2
.^	raising all the elements of a vector to a power	2.^3

The results of the three examples are the same as those of * / ^. But what will happen when it comes to the vector operations?

VECTOR OPERATIONS

Consider the vectors:

$$\mathbf{x} = [x_1, x_2, \ldots, x_n]$$
$$\mathbf{y} = [y_1, y_2, \ldots, y_n]$$

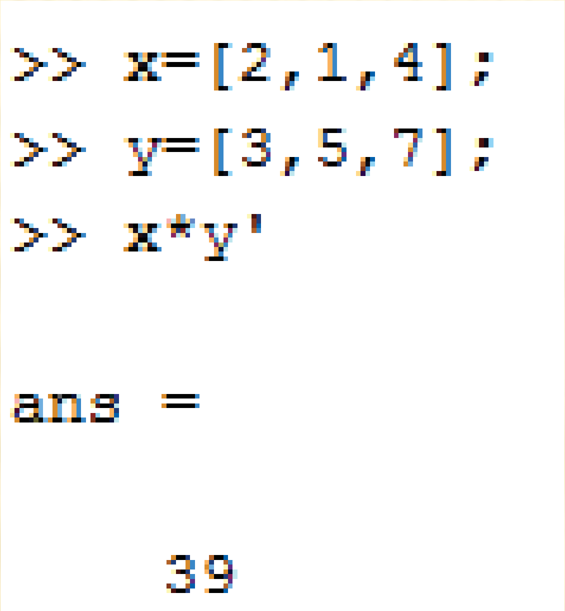
The following operations indicate the resulting vectors:

$$\mathbf{x}.{}^{*}\mathbf{y} = [x_1 y_1, x_2 y_2, \ldots, x_n y_n]$$
$$\mathbf{x}./\mathbf{y} = \left[\frac{x_1}{y_1}, \frac{x_2}{y_3}, \ldots, \frac{x_n}{y_n}\right]$$
$$\mathbf{x}.\hat{}\,\mathbf{p} = [x_1^p, x_2^p, \ldots, x_n^p]$$

VECTOR OPERATIONS

Examples:

(1)Compute the dot product: $\sum_i x_i y_i$

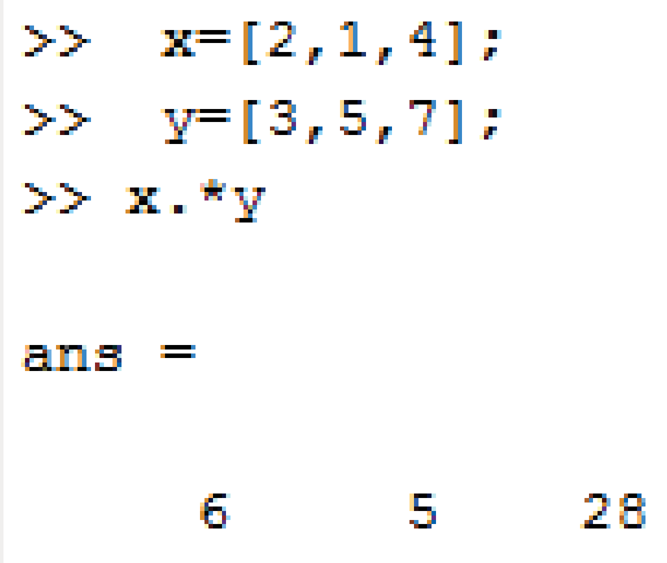


```
>> x=[2,1,4];
>> y=[3,5,7];
>> x*y'

ans =

    39
```

(2) Compute an element-by-element multiplication of two vectors:



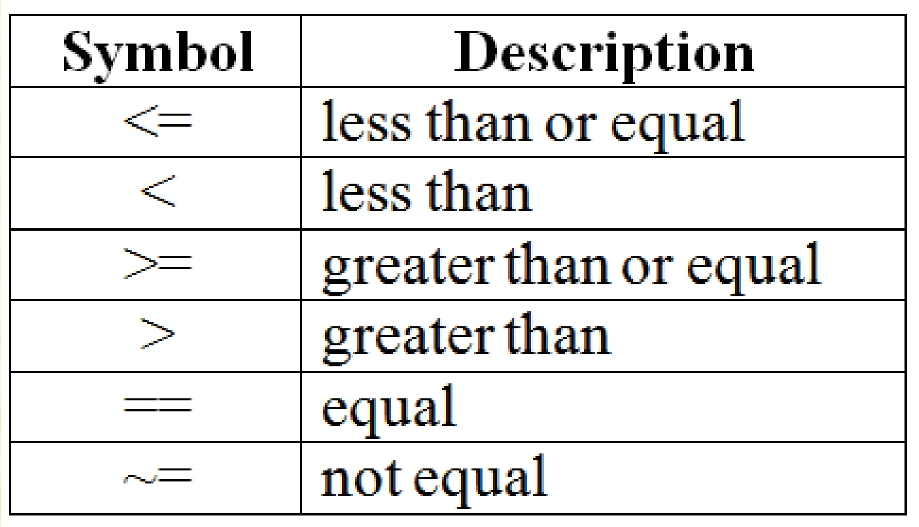
```
>>  x=[2,1,4];
>>  y=[3,5,7];
>> x.*y

ans =

     6     5    28
```

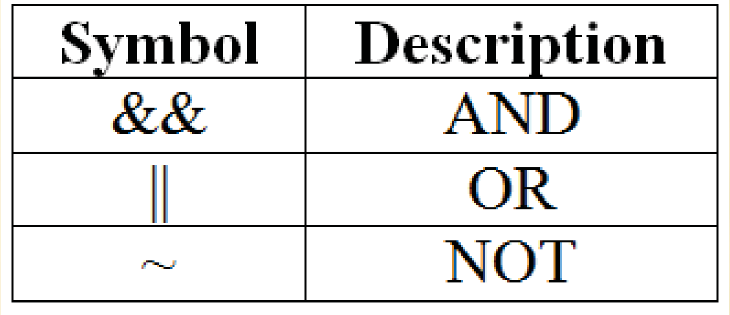
RELATIONAL & LOGICAL OPERATORS

Relational Operators



Symbol	Description
<=	less than or equal
<	less than
>=	greater than or equal
>	greater than
==	equal
~=	not equal

Logical Operators



Symbol	Description
&&	AND
||	OR
~	NOT

CONTROL FLOW

.m files have the following control flow constructs:

- if statements
- for loops
- while loops

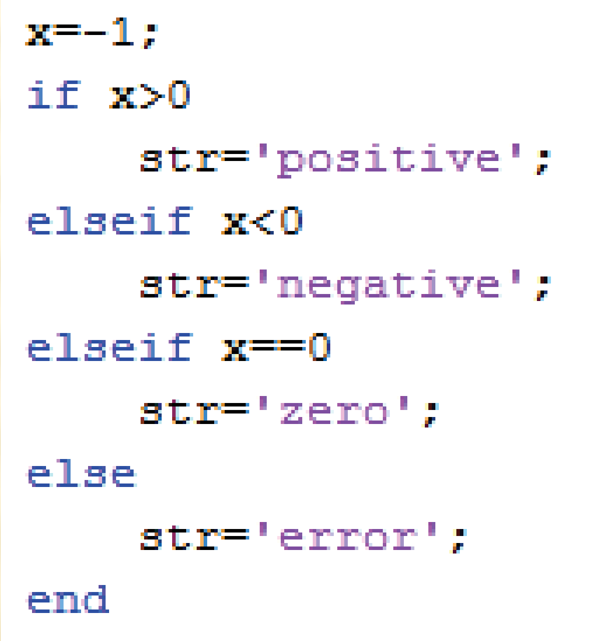
Each of the constructs needs to terminate with an end statement.

CONTROL FLOW

if: conditional execution of certain parts of a code

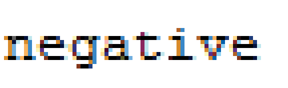
Example:

Matlab code:



```
x=-1;
if x>0
    str='positive';
elseif x<0
    str='negative';
elseif x==0
    str='zero';
else
    str='error';
end
```

Results:



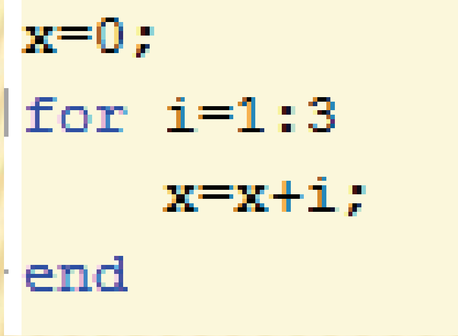
```
negative
```

CONTROL FLOW

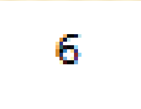
for : repeat certain commands in a predetermined way

Example:

Matlab code:



```
x=0;
for i=1:3
    x=x+i;
end
```

Results: 

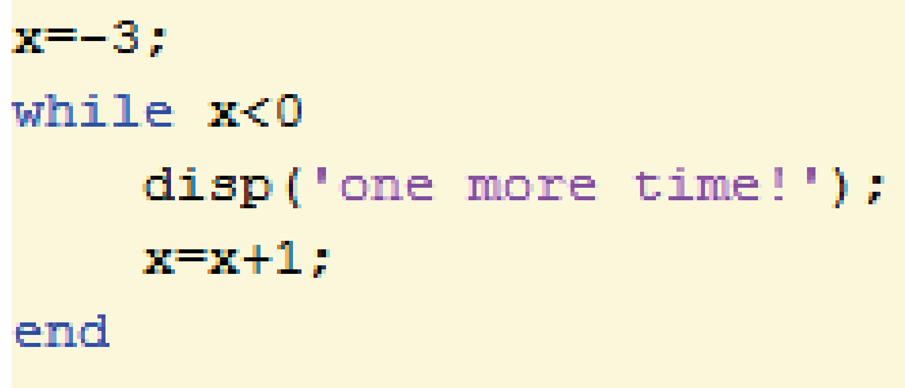
```
6
```

CONTROL FLOW

while: repeats a sequence of commands as long as some condition is met.

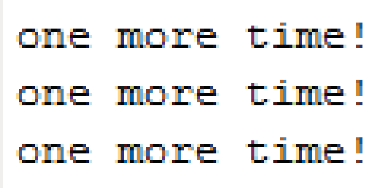
Example:

Matlab code:



```
x=-3;
while x<0
    disp('one more time!');
    x=x+1;
end
```

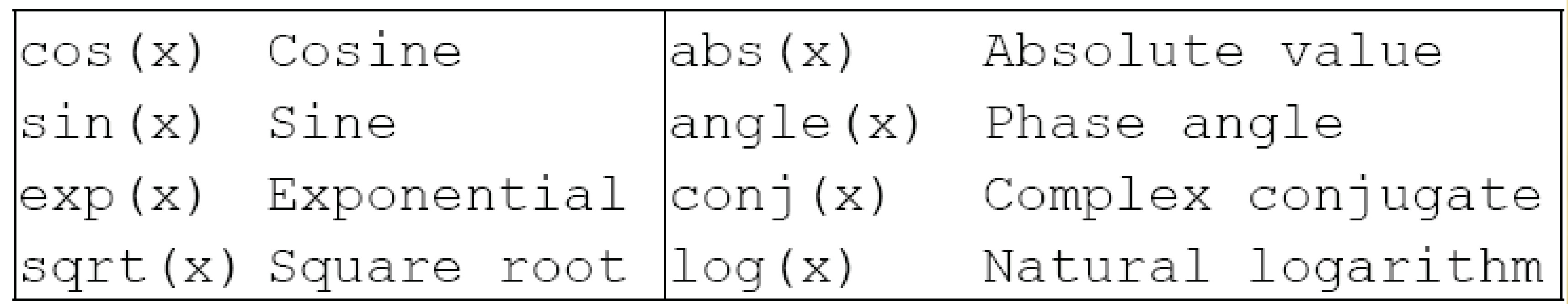
Results:



```
one more time!
one more time!
one more time!
```

MATHEMATICAL FUNCTIONS IN MATLAB

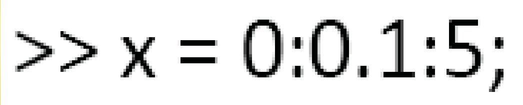
- MATLAB offers many predefined mathematical functions for technical computing, e.g.



`cos(x)`	Cosine	`abs(x)`	Absolute value
`sin(x)`	Sine	`angle(x)`	Phase angle
`exp(x)`	Exponential	`conj(x)`	Complex conjugate
`sqrt(x)`	Square root	`log(x)`	Natural logarithm

- **Colon operator (:)**

  Suppose we want to enter a vector x *consisting of points* (0,0.1,0.2,0.3,…,5).We can use the command



```
>> x = 0:0.1:5;
```

FFT FUNCTION IN MATLAB

- fft function uses the Fast Fourier transform algorithm to compute Discrete Fourier transform.
- **Syntax**

  Y=fft(x)

  Y=fft(x,n)
- **Definition**

  The functions Y=fft(x) and y=ifft(X) implement the transform and inverse transform pair given for vectors of length N by:

$$X(k) = \sum_{j=1}^{N} x(j)\omega_N^{(j-1)(k-1)}$$

$$x(j) = (1/N)\sum_{k=1}^{N} X(k)\omega_N{}^{-(j-1)(k-1)}$$

$\omega_N = e^{(-2\pi i)/N}$ Is an Nth root of unity

FFT FUNCTION IN MATLAB

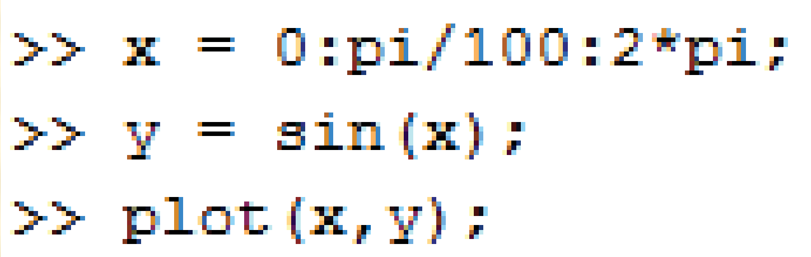
- Description

Y =fft(x) returns the DFT of vector x, computed with a fast Fourier transform algorithm.

Y=fft(x,n) returns the n-point DFT. If the length of x is less than n, x is padded with trailing zeros to length n. If the length of x is greater than n, the sequence x is truncated.

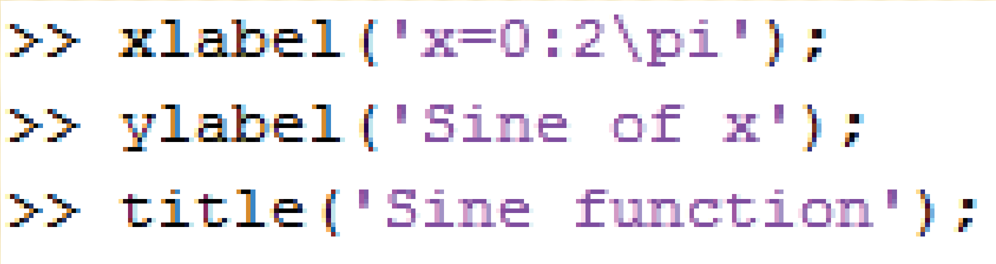
BASIC PLOTTING IN MATLAB

- MATLAB has an excellent set of graphic tools. Plotting a given data set or the results of computation is possible with very few commands .
- The MATLAB command to plot a graph is **plot(x,y),** e.g.



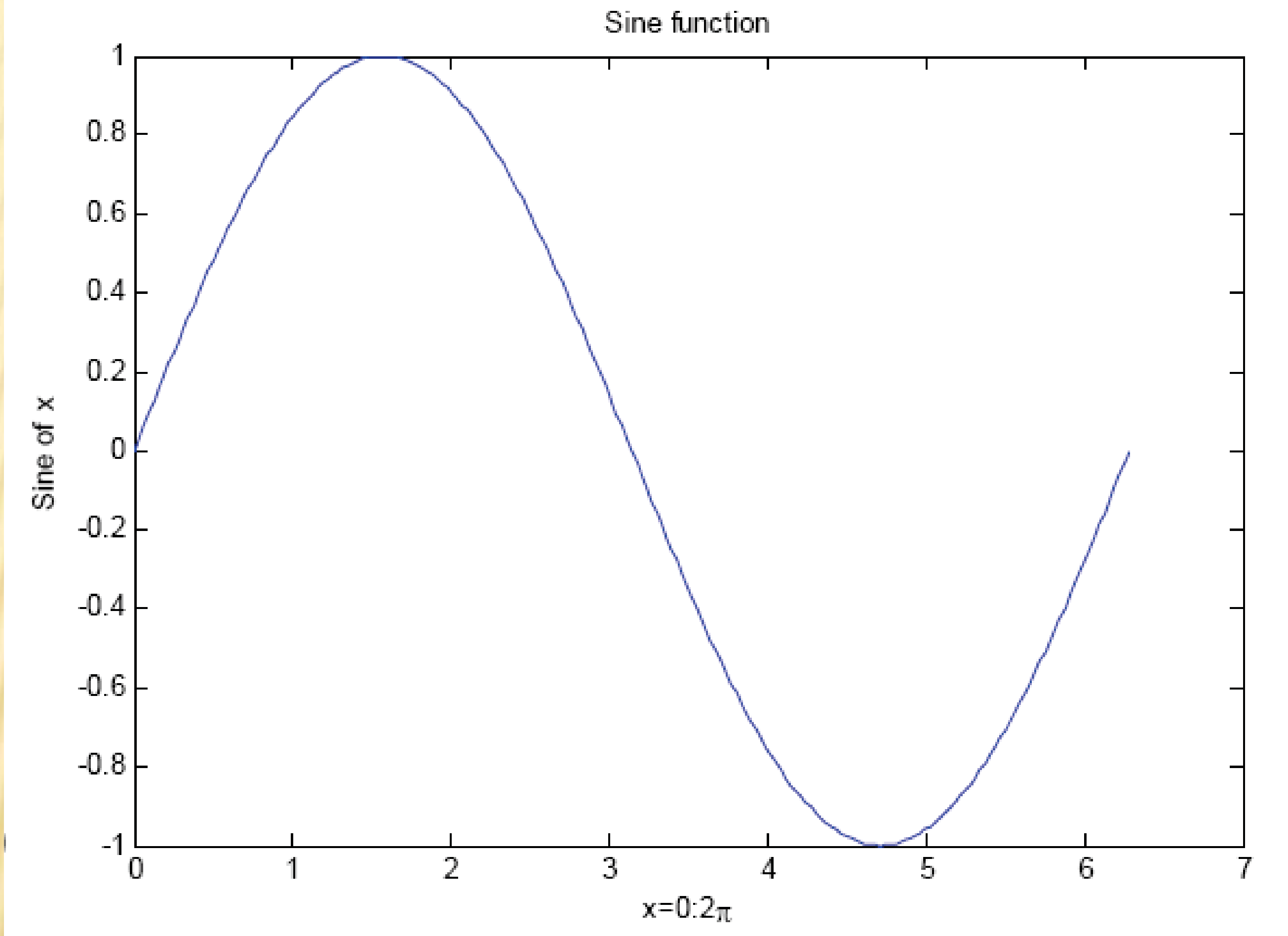
```
>> x = 0:pi/100:2*pi;
>> y = sin(x);
>> plot(x,y);
```

- MATLAB enables you to add axis labels and titles, e.g.



```
>> xlabel('x=0:2\pi');
>> ylabel('Sine of x');
>> title('Sine function');
```

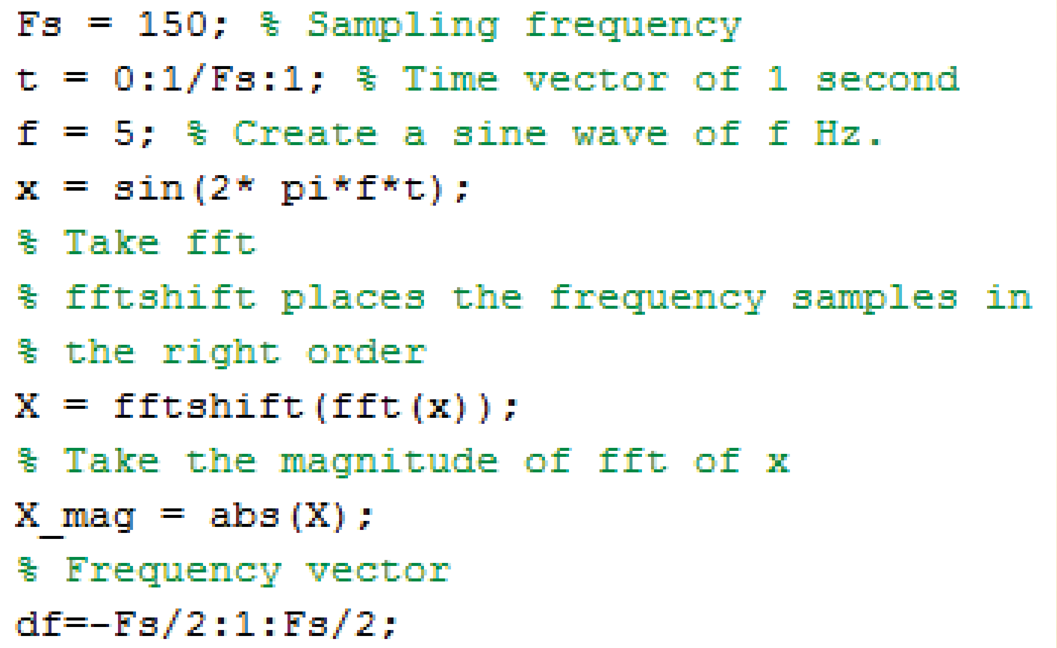
BASIC PLOTTING IN MATLAB



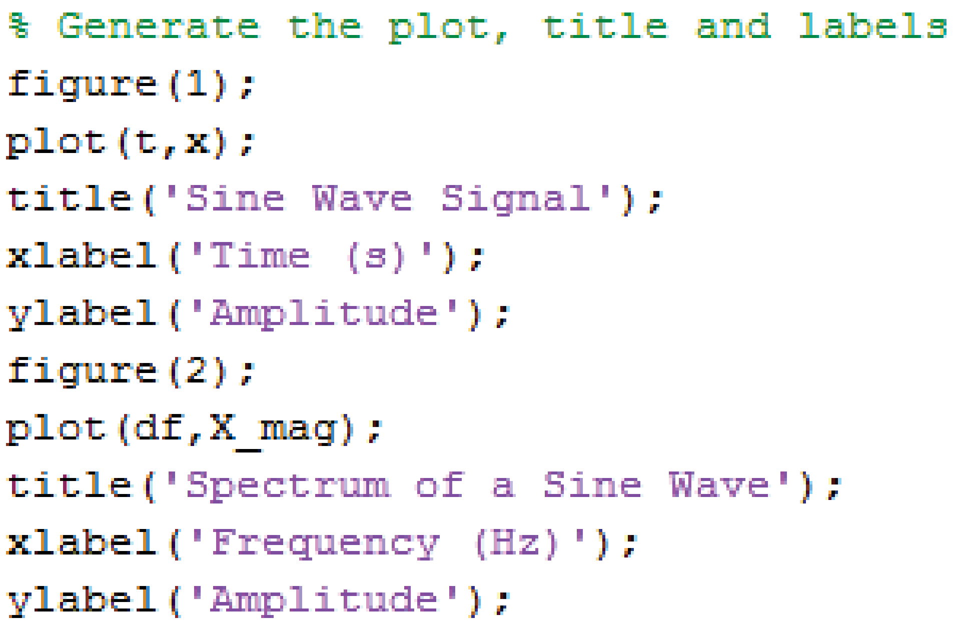
BASIC PLOTTING IN MATLAB

Example 1: Sine Wave

Matlab code:



```
Fs = 150; % Sampling frequency
t = 0:1/Fs:1; % Time vector of 1 second
f = 5; % Create a sine wave of f Hz.
x = sin(2* pi*f*t);
% Take fft
% fftshift places the frequency samples in
% the right order
X = fftshift(fft(x));
% Take the magnitude of fft of x
X_mag = abs(X);
% Frequency vector
df=-Fs/2:1:Fs/2;
```

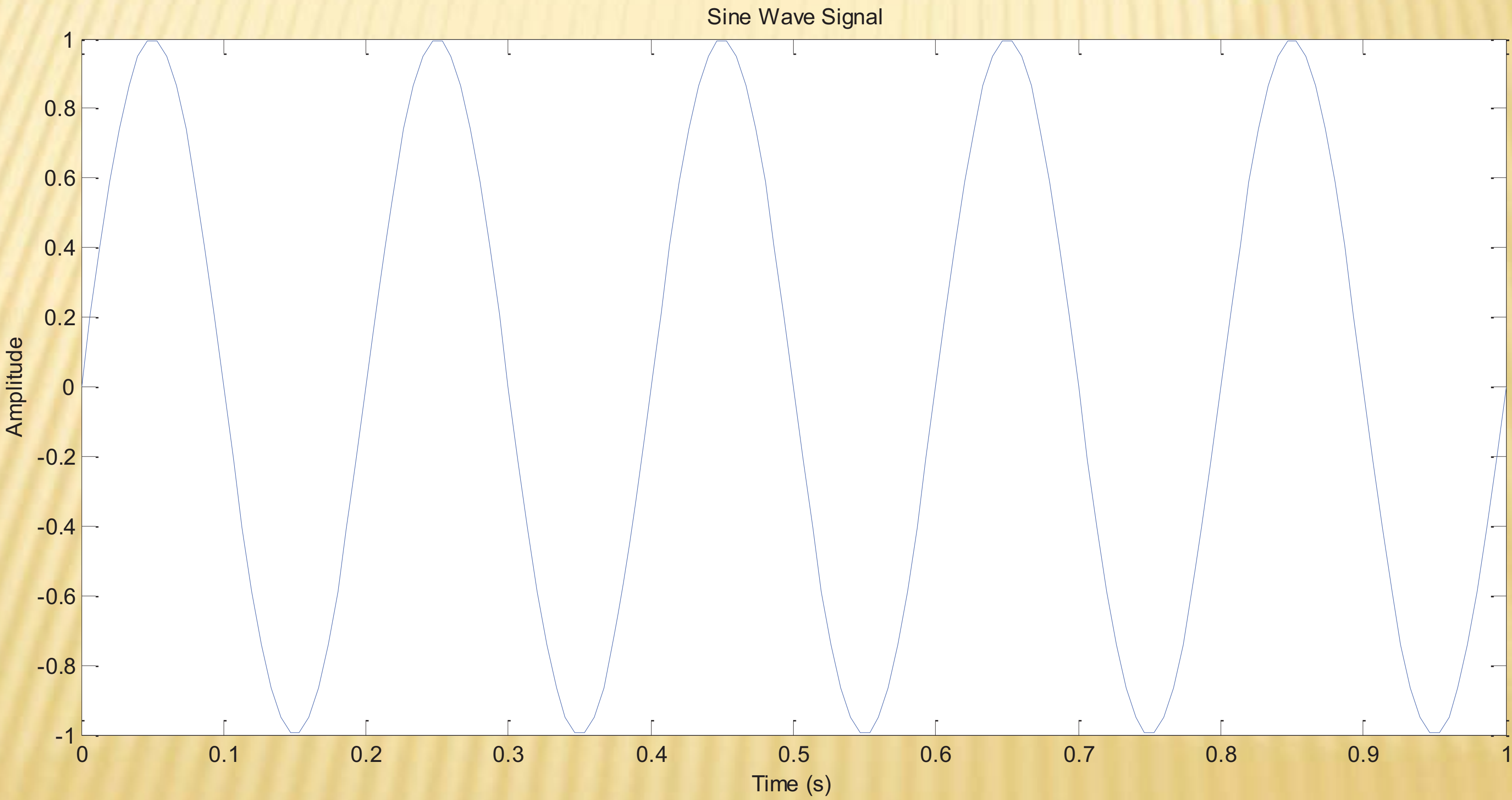


```
% Generate the plot, title and labels
figure(1);
plot(t,x);
title('Sine Wave Signal');
xlabel('Time (s)');
ylabel('Amplitude');
figure(2);
plot(df,X_mag);
title('Spectrum of a Sine Wave');
xlabel('Frequency (Hz)');
ylabel('Amplitude');
```

BASIC PLOTTING IN MATLAB

Example 1: Sine Wave

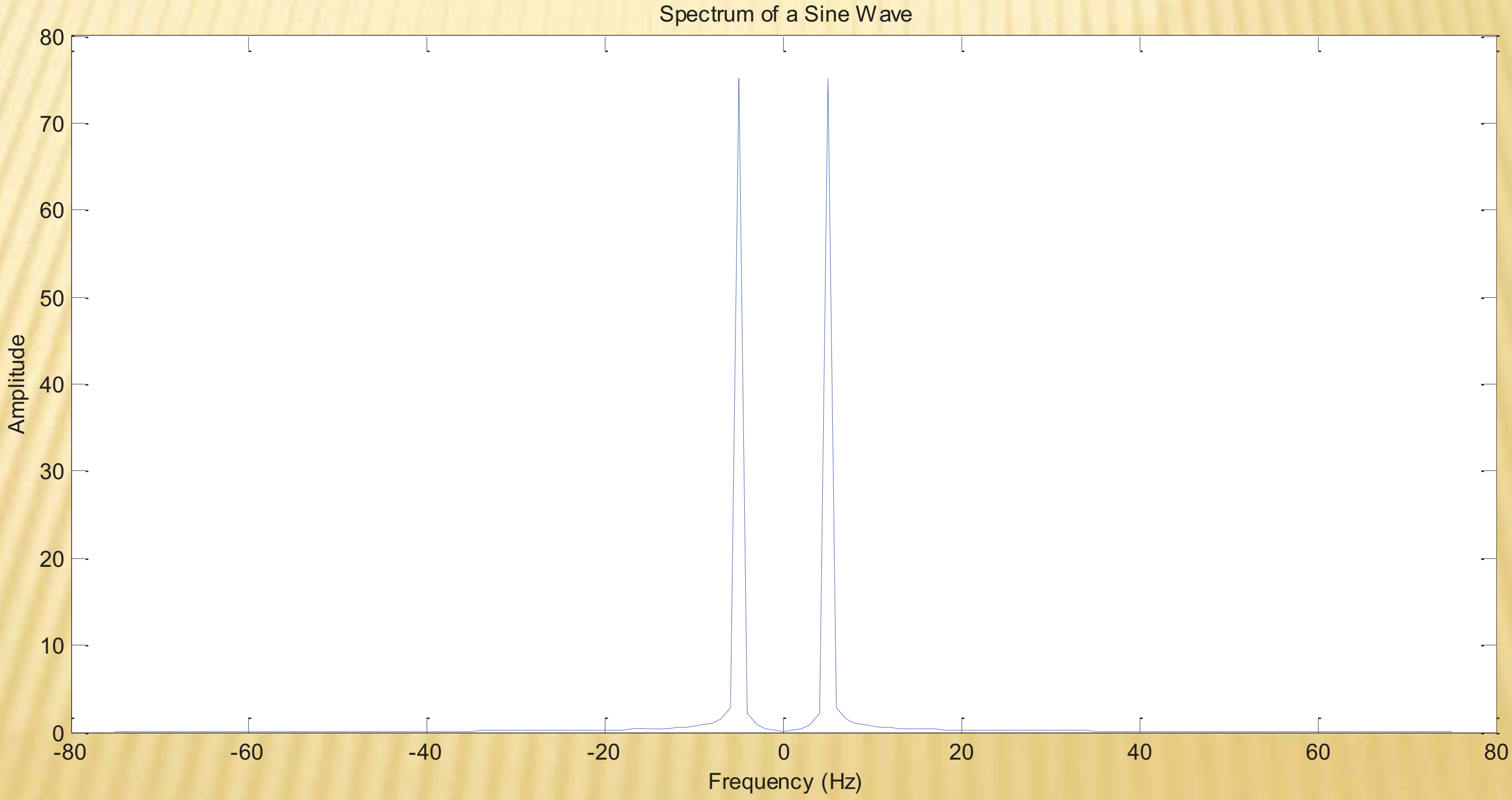
Plot: signal in time domain



BASIC PLOTTING IN MATLAB

Example 1: Sine Wave

Plot: signal in frequency domain

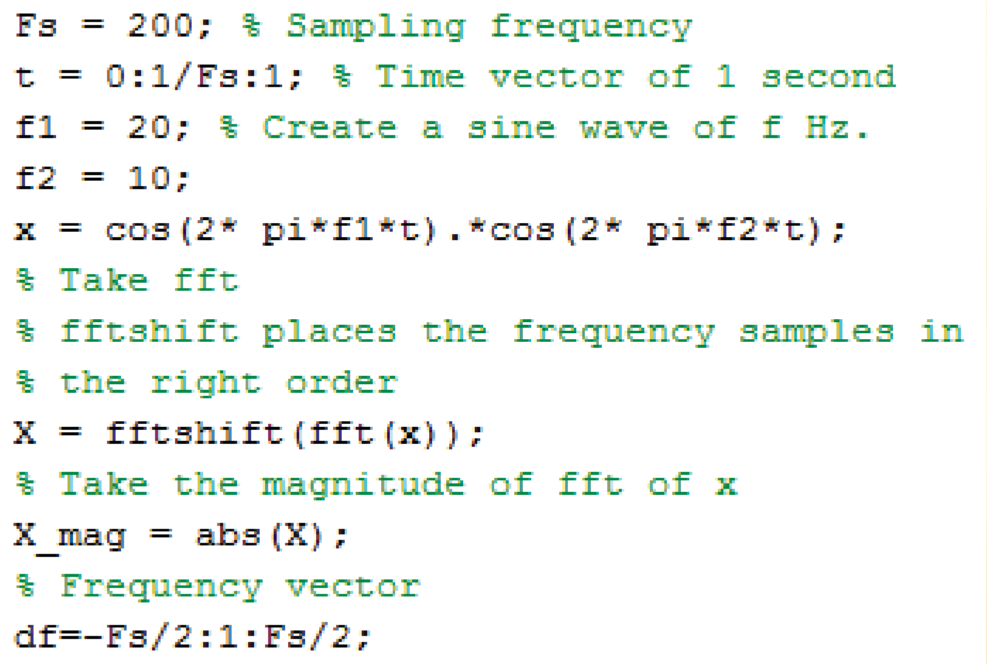


BASIC PLOTTING IN MATLAB

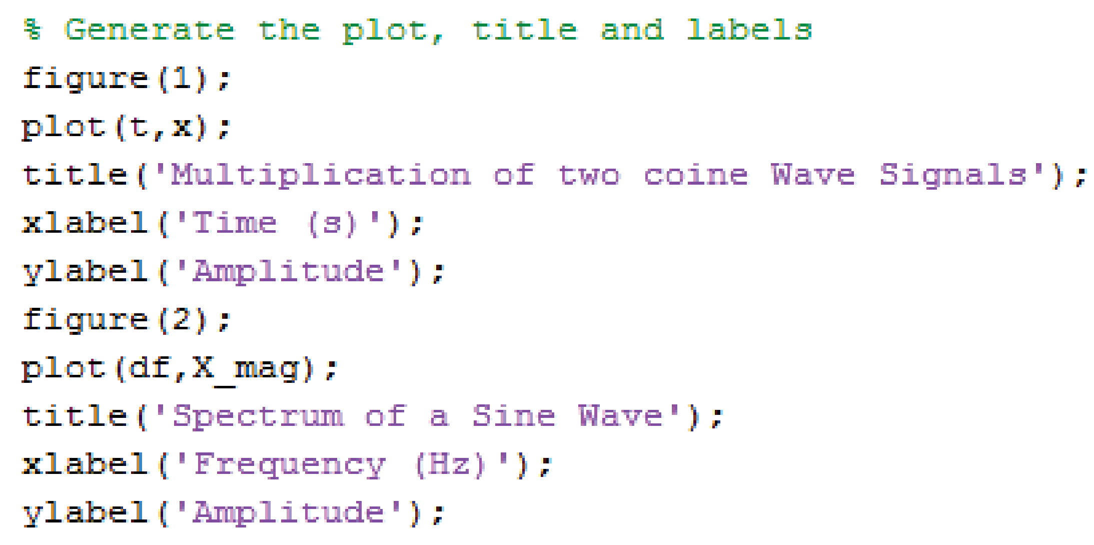
Example 2:Multiplication of two Cosine Waves

Matlab code:

f1 = 20; f2=10



```
Fs = 200; % Sampling frequency
t = 0:1/Fs:1; % Time vector of 1 second
f1 = 20; % Create a sine wave of f Hz.
f2 = 10;
x = cos(2* pi*f1*t).*cos(2* pi*f2*t);
% Take fft
% fftshift places the frequency samples in
% the right order
X = fftshift(fft(x));
% Take the magnitude of fft of x
X_mag = abs(X);
% Frequency vector
df=-Fs/2:1:Fs/2;
```



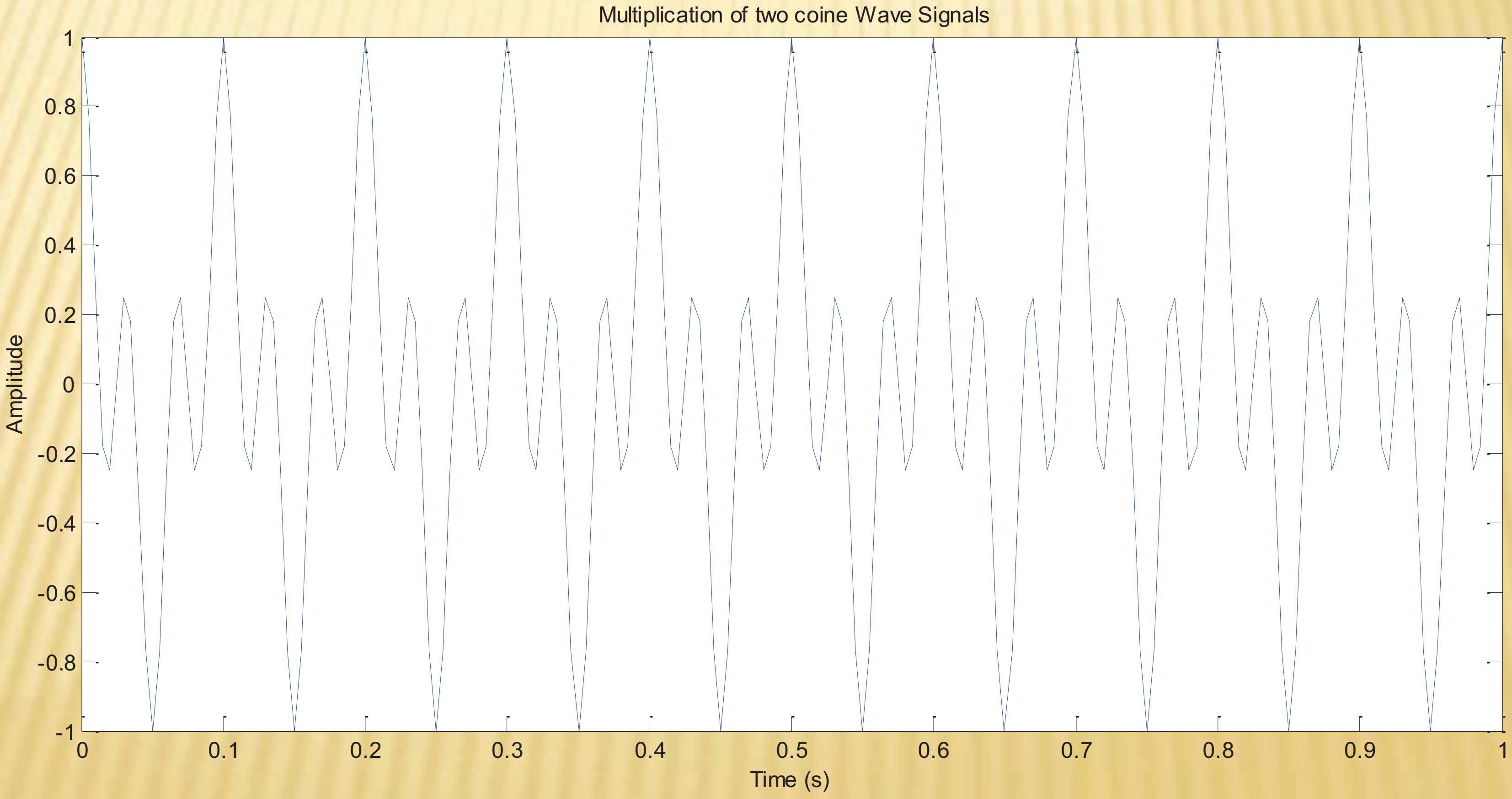
```
% Generate the plot, title and labels
figure(1);
plot(t,x);
title('Multiplication of two coine Wave Signals');
xlabel('Time (s)');
ylabel('Amplitude');
figure(2);
plot(df,X_mag);
title('Spectrum of a Sine Wave');
xlabel('Frequency (Hz)');
ylabel('Amplitude');
```

BASIC PLOTTING IN MATLAB

Example 2:Multiplication of two Cosine Waves

Plot: signal in time domain

f1 = 20; f2=10

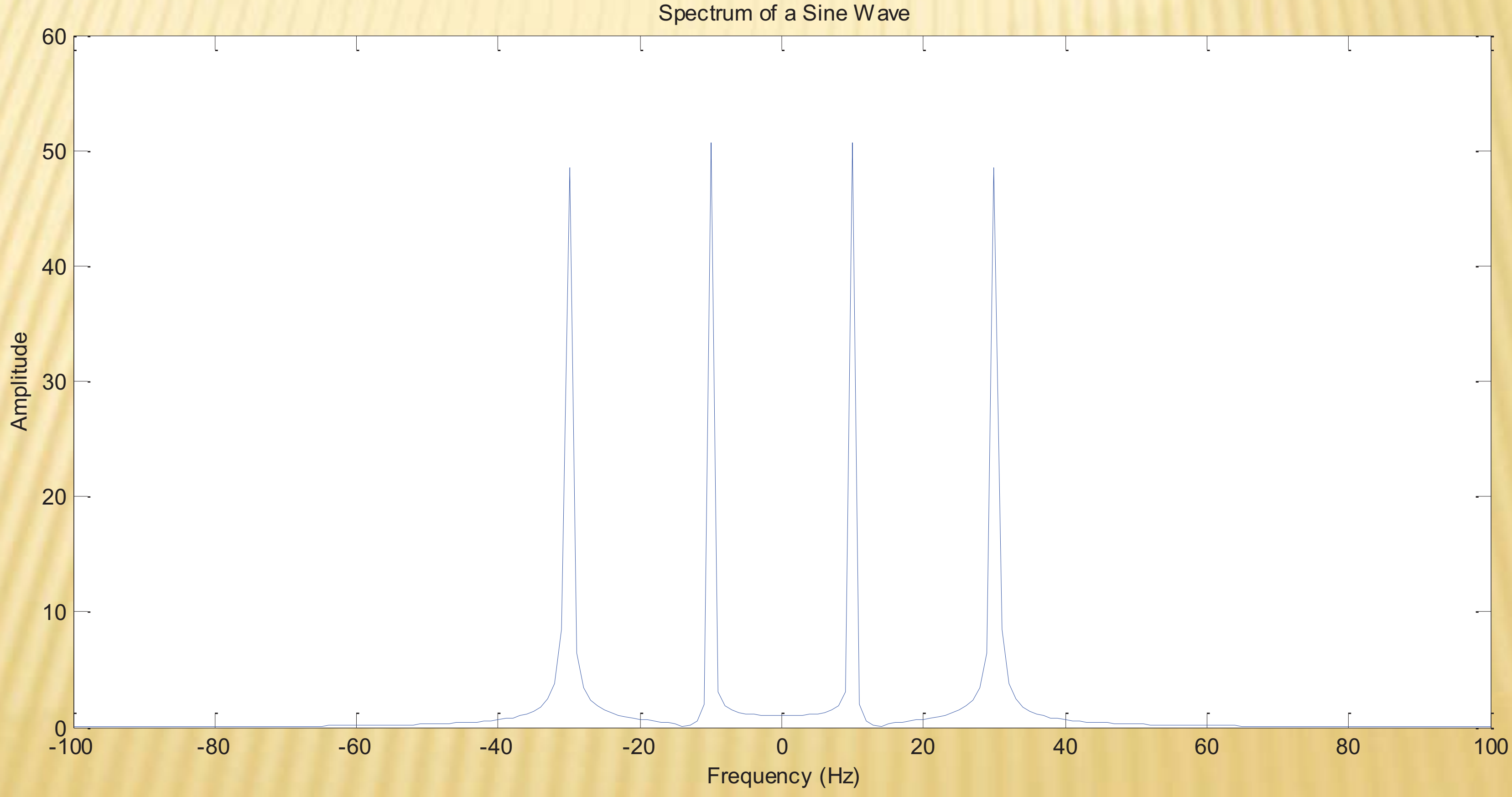


BASIC PLOTTING IN MATLAB

Example 2:Multiplication of two Cosine Waves

Plot: signal in frequency domain

f1 = 20; f2=10



BASIC PLOTTING IN MATLAB

Example 2:Multiplication of two Cosine Waves

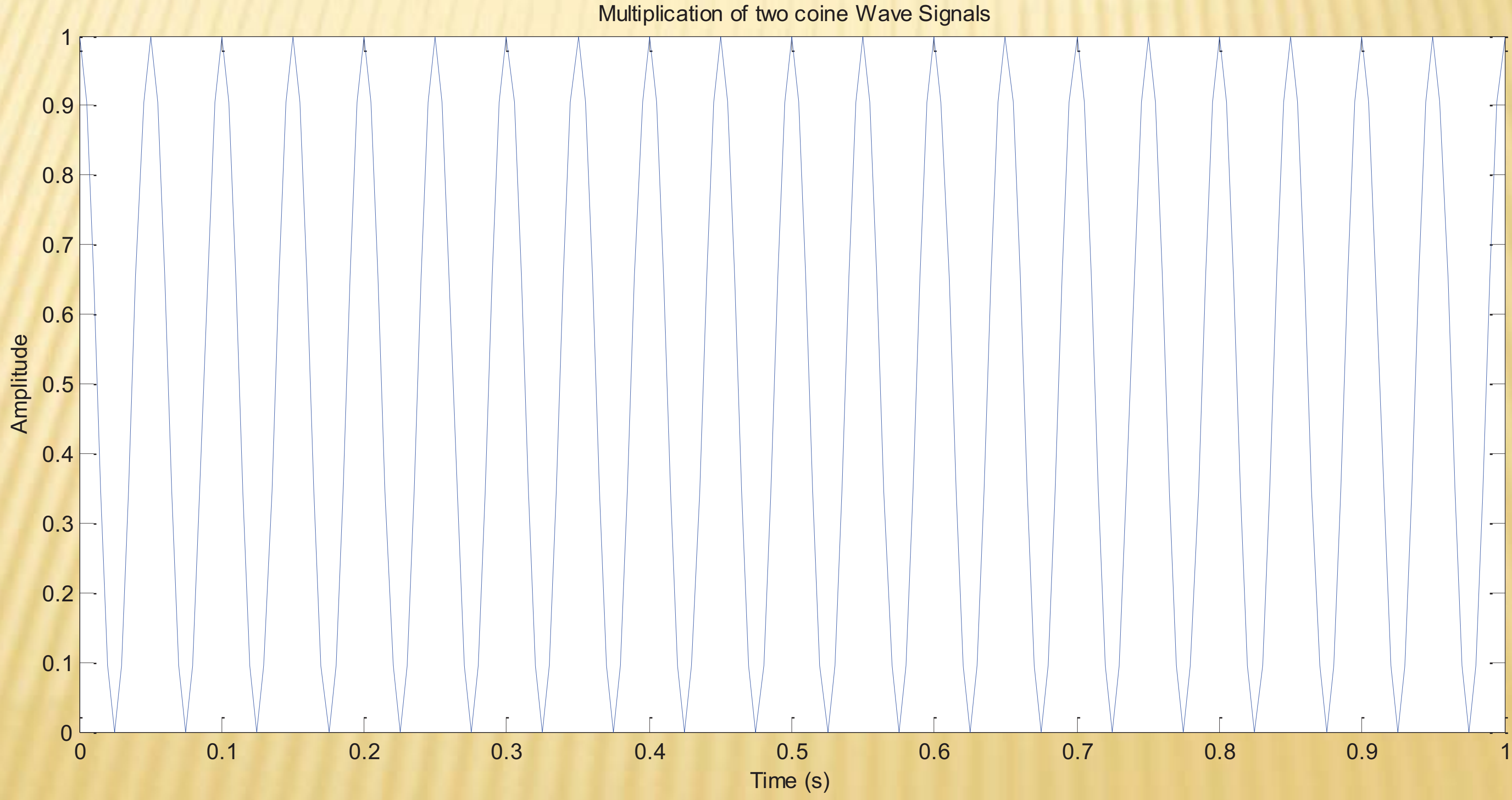
What happens when f1=f2?

BASIC PLOTTING IN MATLAB

Example 2:Multiplication of two Cosine Waves

Plot: signal in time domain

f1 = 10; f2=10

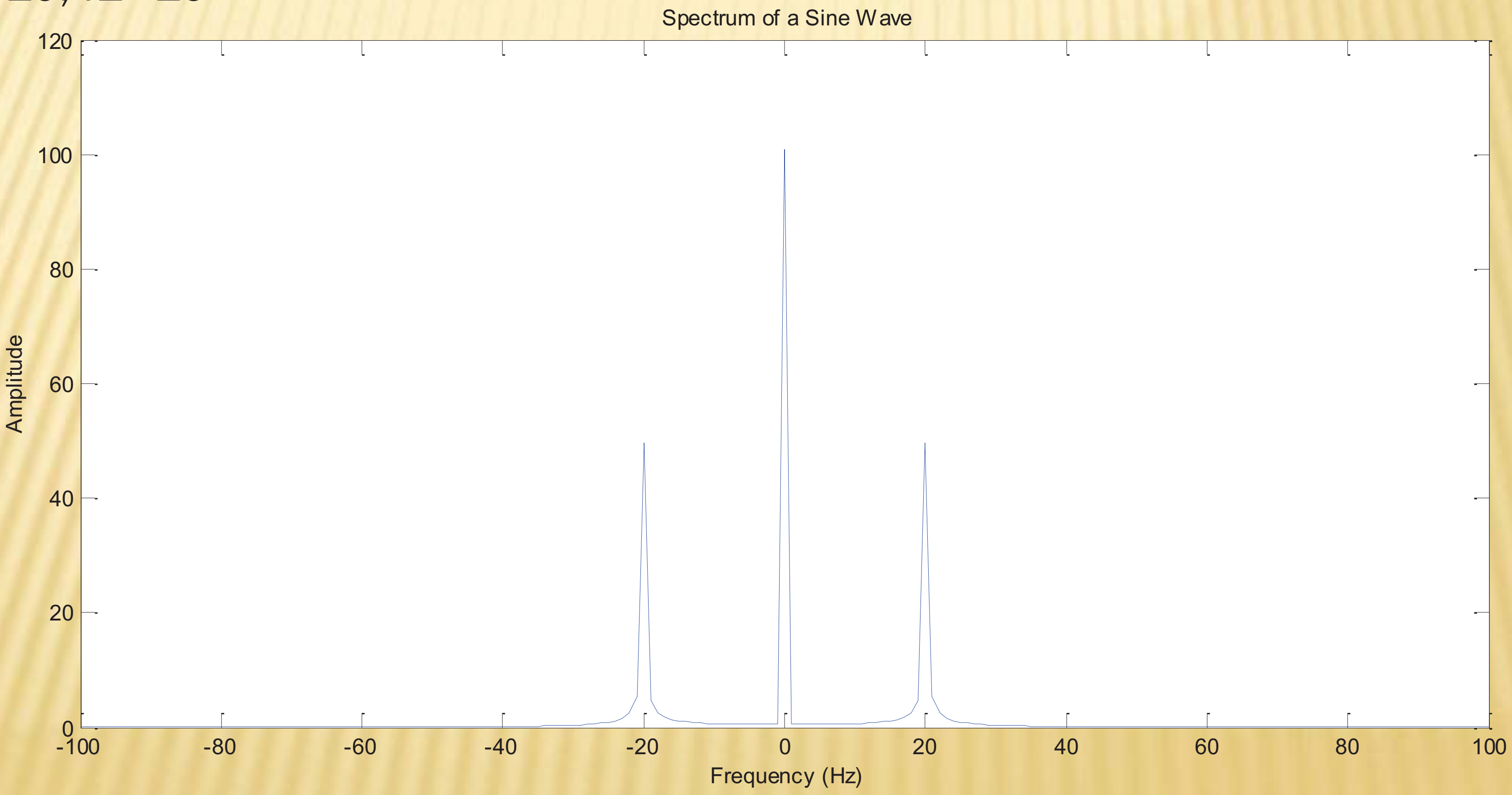


BASIC PLOTTING IN MATLAB

Example 2:Multiplication of two Cosine Waves

Plot: signal in frequency domain

f1 = 10; f2=10



It has 3 spectrum components instead of 4. Why?

BASIC PLOTTING IN MATLAB

Example 2:Multiplication of two Cosine Waves

Plot: signal in frequency domain

f1 = 10; f2=10

It has 3 spectrum components instead of 4. Why?

Hints:

$$\cos(2\pi f_1 t)\cdot\cos(2\pi f_2 t)=\frac{\cos(2\pi(f_1+f_2)t)+\cos(2\pi(f_1-f_2)t)}{2}$$

Thank You