Introduction for MATLAB

MATLAB is a numerical computing environment developed by MathWorks. MATLAB allows matrix manipulations, plotting of functions and data, and implementation of algorithms

Getting help

You can get help by typing the commands `help` or `lookfor` at the >> prompt, e.g.

```
>> help fft
```

Arithmetic operators

Symbol	Operation	Example
+	Addition	3.1 + 9
-	Subtraction	6.2 – 5
*	Multiplication	2 * 3
/	Division	5 / 2
^	Power	3^2

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)
```

For example,

```
>>  x = 5
x =
        5
>>  y = [3*7, pi/3];    % pi is  π   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, e.g.

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

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;
```

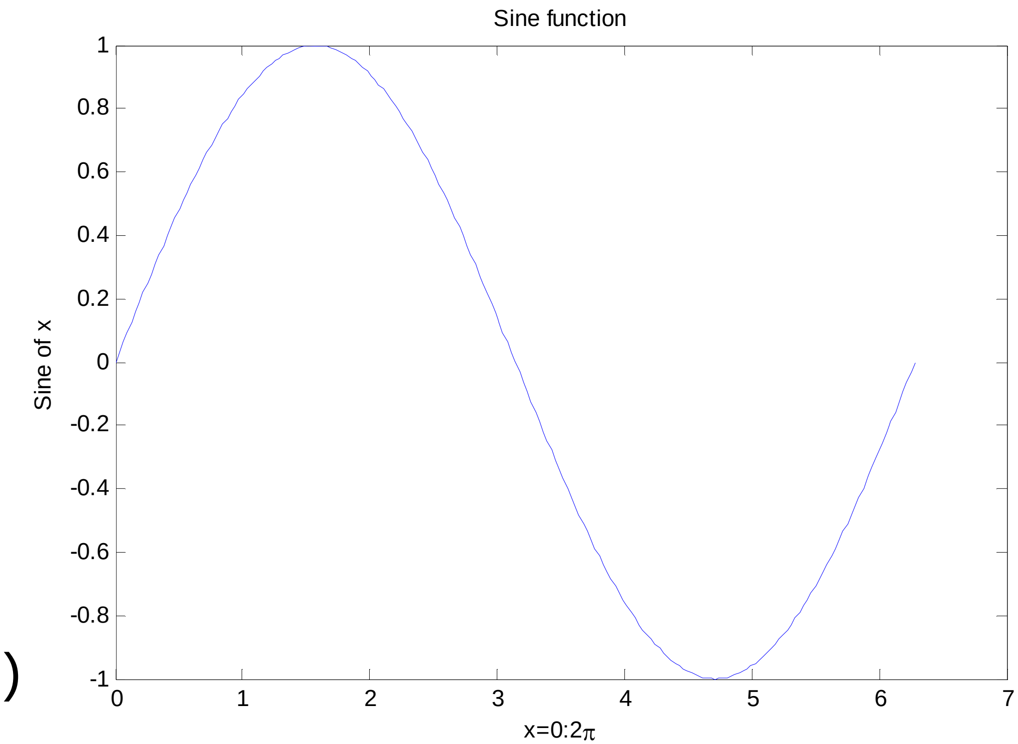
Most of the work you will do in MATLAB will be stored in files called *scripts*, or m-files, containing sequences of MATLAB commands to be executed over and over again

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.

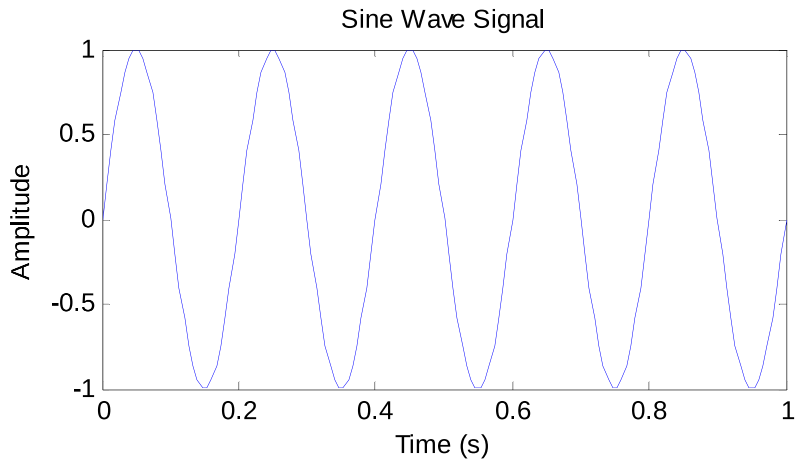
```
>> x = 0:pi/100:2*pi;
>> y = sin(x);
>> plot(x,y)
```

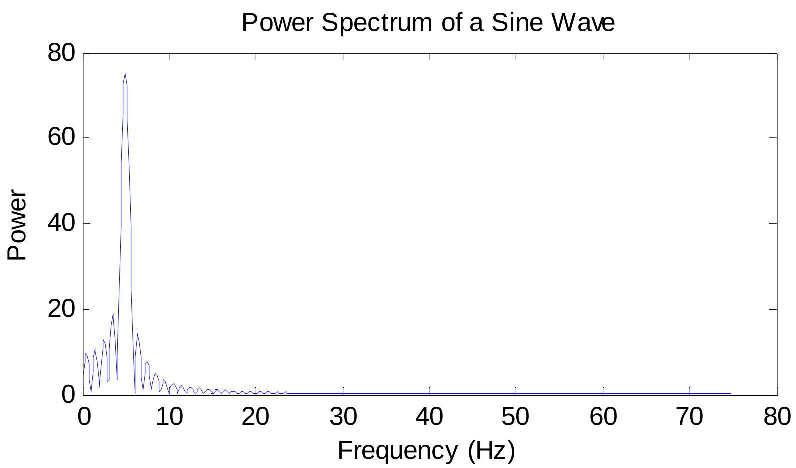


MATLAB enables you to add axis Labels and titles, e.g.

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

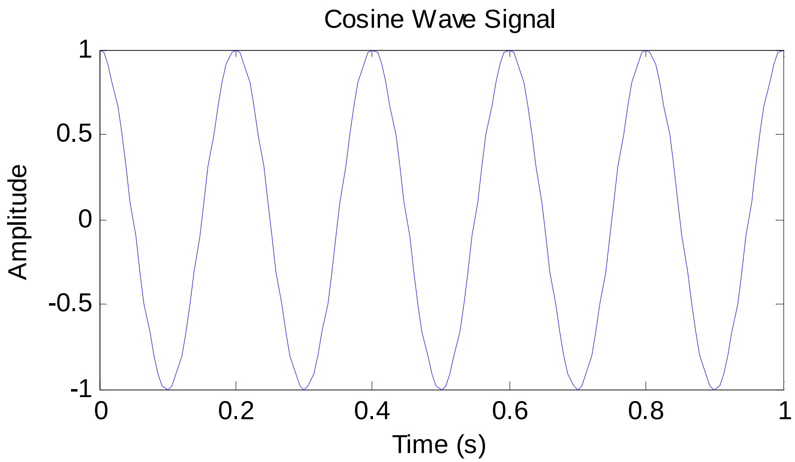
Example 1: Sine Wave

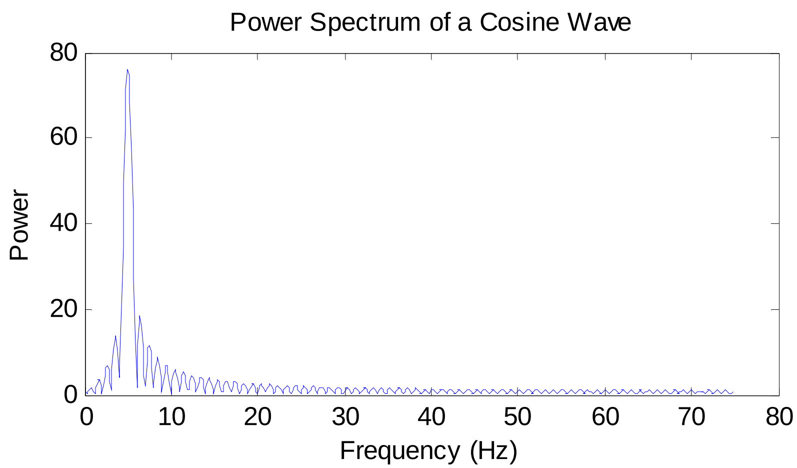





```
Fs = 150; % Sampling frequency
t = 0:1/Fs:1; % Time vector of 1 second
f = 5; % Create a sine wave of f Hz.
x = sin(2*pi*t*f);
nfft = 1024; % Length of FFT
% Take fft, padding with zeros so that length(X)
is equal to nfft
X = fft(x,nfft);
% FFT is symmetric, throw away second half
X = X(1:nfft/2);
% Take the magnitude of fft of x
mx = abs(X);
% Frequency vector
f = (0:nfft/2-1)*Fs/nfft;
% Generate the plot, title and labels.
figure(1);
plot(t,x);
title('Sine Wave Signal');
xlabel('Time (s)');
ylabel('Amplitude');
figure(2);
plot(f,mx);
title('Power Spectrum of a Sine Wave');
xlabel('Frequency (Hz)');
ylabel('Power');
```

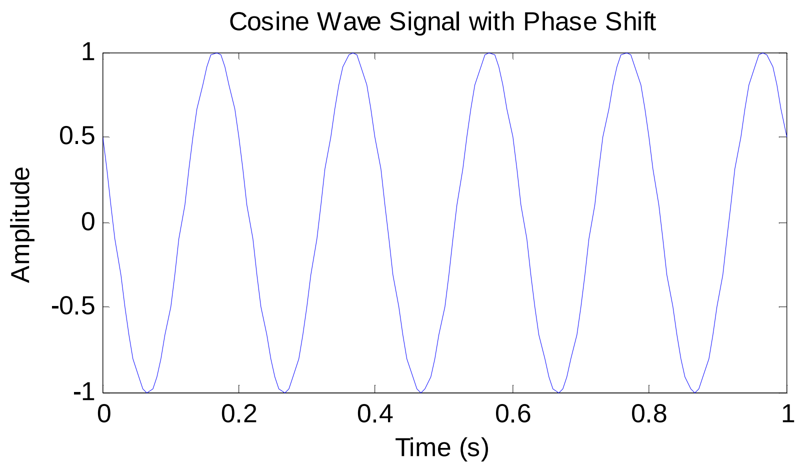
Example 2: Cosine Wave

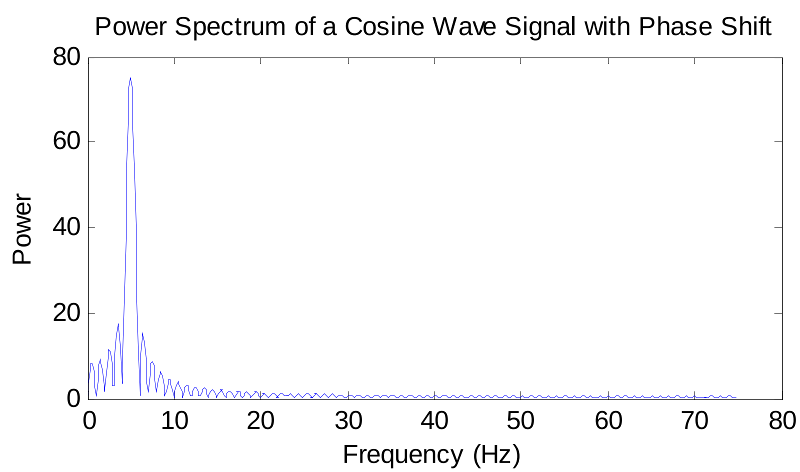





```
Fs = 150; % Sampling frequency
t = 0:1/Fs:1; % Time vector of 1 second
f = 5; % Create a sine wave of f Hz.
x = cos(2*pi*t*f);
nfft = 1024; % Length of FFT
% Take fft, padding with zeros so that length(X) is
equal to nfft
X = fft(x,nfft);
% FFT is symmetric, throw away second half
X = X(1:nfft/2);
% Take the magnitude of fft of x
mx = abs(X);
% Frequency vector
f = (0:nfft/2-1)*Fs/nfft;
% Generate the plot, title and labels.
figure(1);
plot(t,x);
title('Sine Wave Signal');
xlabel('Time (s)');
ylabel('Amplitude');
figure(2);
plot(f,mx);
title('Power Spectrum of a Sine Wave');
xlabel('Frequency (Hz)');
ylabel('Power');
```

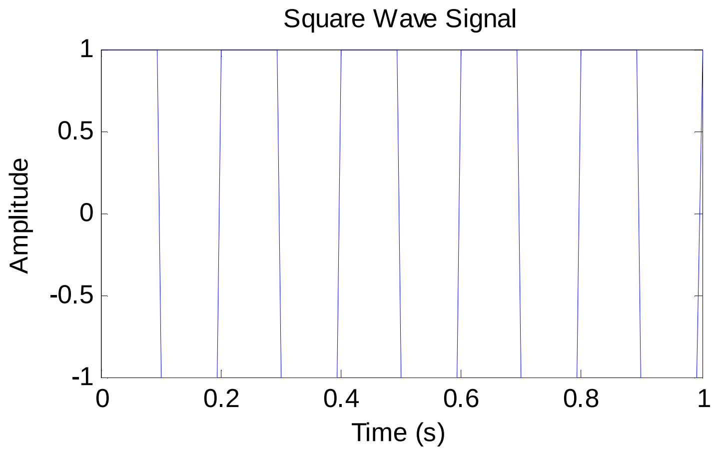
Example 3: Cosine Wave with Phase Shift

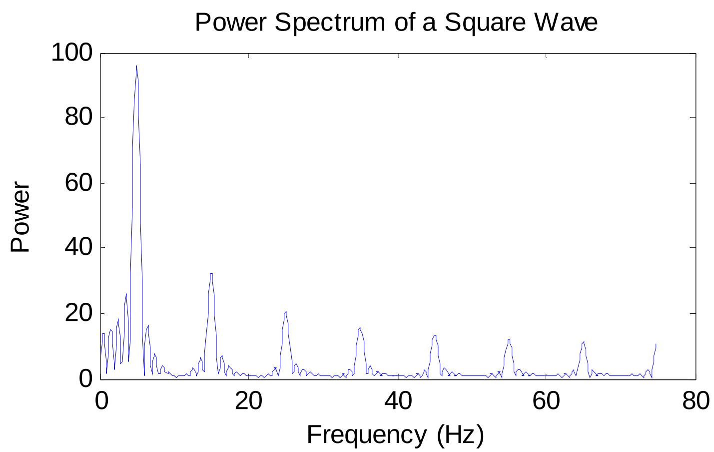





```
Fs = 150; % Sampling frequency
t = 0:1/Fs:1; % Time vector of 1 second
f = 5; % Create a sine wave of f Hz.
pha = 1/3*pi; % phase shift
x = cos(2*pi*t*f + pha);
nfft = 1024; % Length of FFT
% Take fft, padding with zeros so that length(X) is
equal to nfft
X = fft(x,nfft);
% FFT is symmetric, throw away second half
X = X(1:nfft/2);
% Take the magnitude of fft of x
mx = abs(X);
% Frequency vector
f = (0:nfft/2-1)*Fs/nfft;
% Generate the plot, title and labels.
figure(1);
plot(t,x);
title('Sine Wave Signal');
xlabel('Time (s)');
ylabel('Amplitude');
figure(2);
plot(f,mx);
title('Power Spectrum of a Sine Wave');
xlabel('Frequency (Hz)');
ylabel('Power');
```

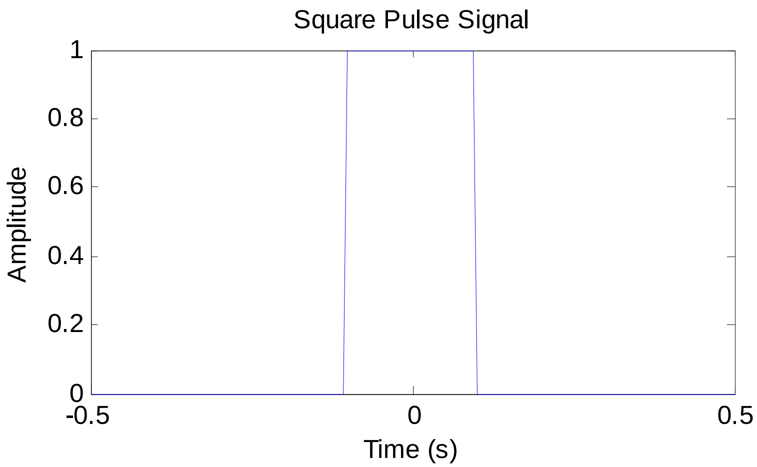
Example 4: Square Wave

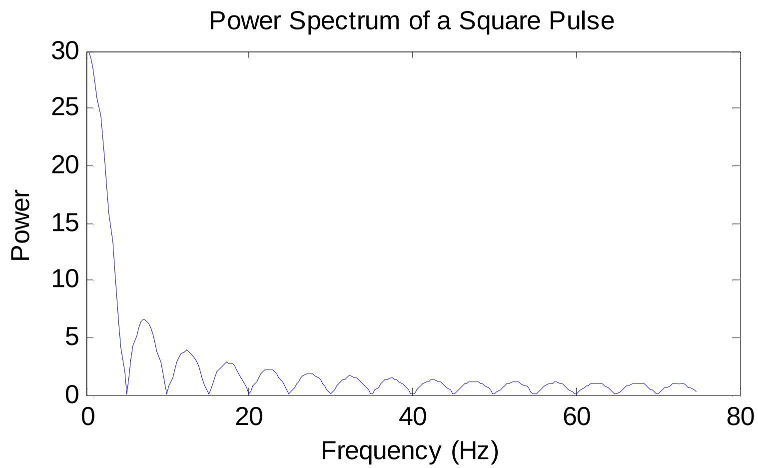





```
Fs = 150; % Sampling frequency
t = 0:1/Fs:1; % Time vector of 1 second
f = 5; % Create a sine wave of f Hz.
x = square(2*pi*t*f);
nfft = 1024; % Length of FFT
% Take fft, padding with zeros so that length(X) is
equal to nfft
X = fft(x,nfft);
% FFT is symmetric, throw away second half
X = X(1:nfft/2);
% Take the magnitude of fft of x
mx = abs(X);
% Frequency vector
f = (0:nfft/2-1)*Fs/nfft;
% Generate the plot, title and labels.
figure(1);
plot(t,x);
title('Square Wave Signal');
xlabel('Time (s)');
ylabel('Amplitude');
figure(2);
plot(f,mx);
title('Power Spectrum of a Square Wave');
xlabel('Frequency (Hz)');
ylabel('Power');
```

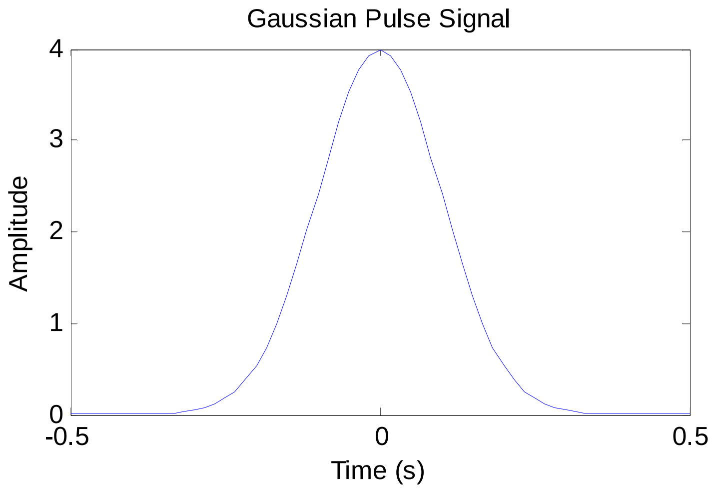
Example 5: Square Pulse

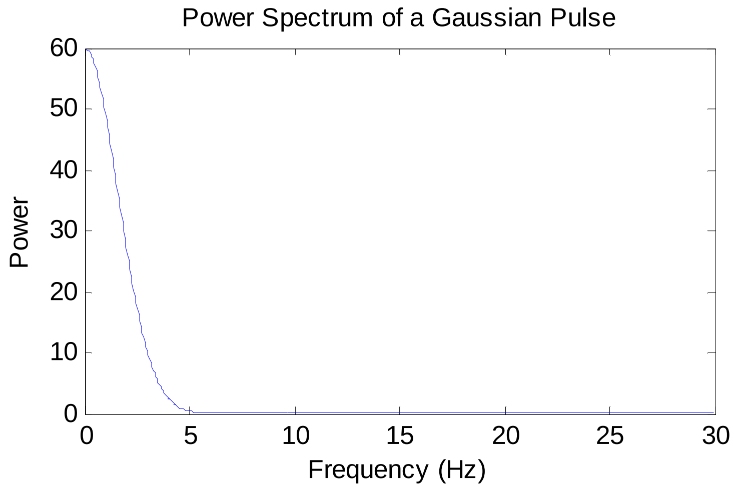





```
Fs = 150; % Sampling frequency
t = -0.5:1/Fs:0.5; % Time vector of 1 second
w = .2; % width of rectangle
x = rectpuls(t, w); % Generate Square Pulse
nfft = 512; % Length of FFT
% Take fft, padding with zeros so that length(X) is
equal to nfft
X = fft(x,nfft);
% FFT is symmetric, throw away second half
X = X(1:nfft/2);
% Take the magnitude of fft of x
mx = abs(X);
% Frequency vector
f = (0:nfft/2-1)*Fs/nfft;
% Generate the plot, title and labels.
figure(1);
plot(t,x);
title('Square Pulse Signal');
xlabel('Time (s)');
ylabel('Amplitude');
figure(2);
plot(f,mx);
title('Power Spectrum of a Square Pulse');
xlabel('Frequency (Hz)');
ylabel('Power');
```

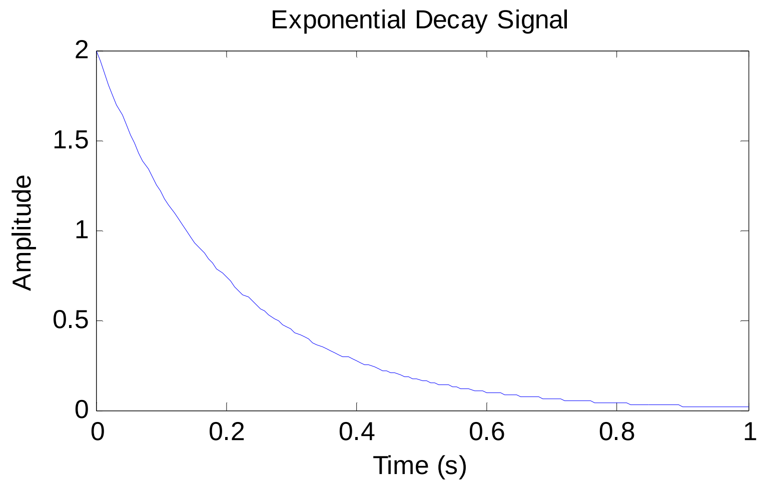
Example 6: Gaussian Pulse

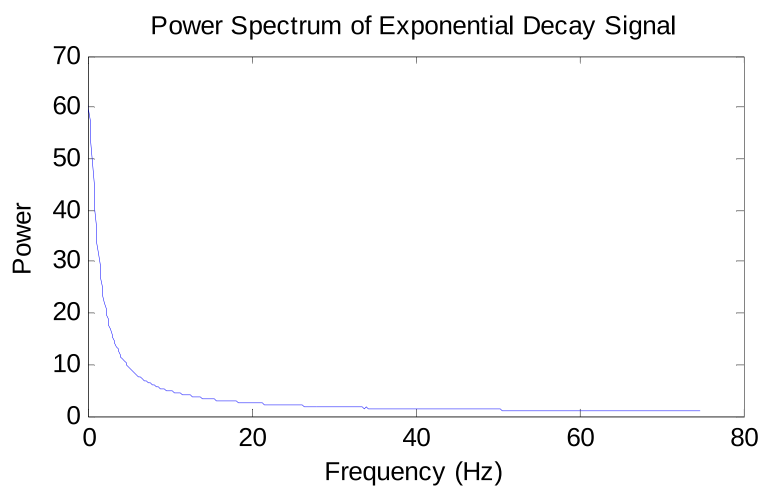





```
Fs = 60; % Sampling frequency
t = -.5:1/Fs:.5;
x = 1/(sqrt(2*pi*0.01))*(exp(-t.^2/(2*0.01)));
nfft = 1024; % Length of FFT
% Take fft, padding with zeros so that
length(X) is equal to nfft
X = fft(x,nfft);
% FFT is symmetric, throw away second half
X = X(1:nfft/2);
% Take the magnitude of fft of x
mx = abs(X);
% This is an evenly spaced frequency vector
f = (0:nfft/2-1)*Fs/nfft;
% Generate the plot, title and labels.
figure(1);
plot(t,x);
title('Gaussian Pulse Signal');
xlabel('Time (s)');
ylabel('Amplitude');
figure(2);
plot(f,mx);
title('Power Spectrum of a Gaussian Pulse');
xlabel('Frequency (Hz)');
ylabel('Power');
```

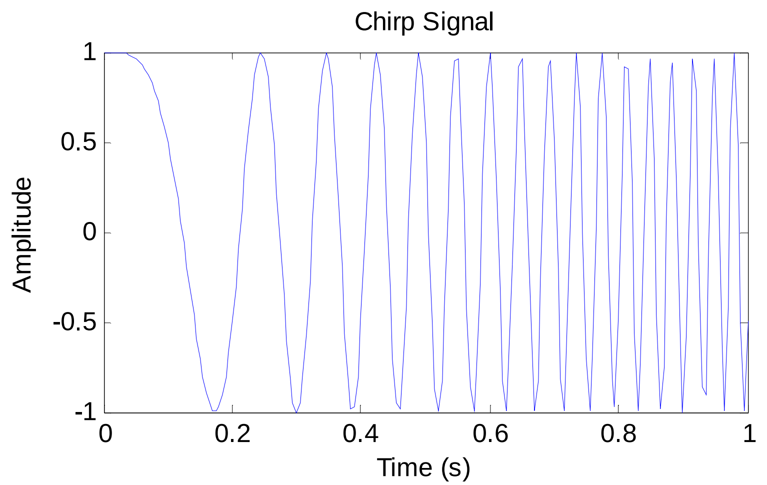
Example 7: Exponential Decay

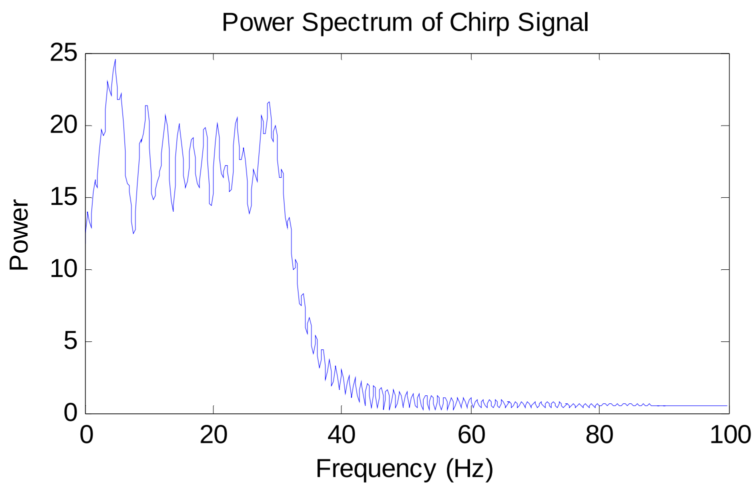





```
Fs = 150; % Sampling frequency
t = 0:1/Fs:1; % Time vector of 1 second
x = 2*exp(-5*t);
nfft = 1024; % Length of FFT
% Take fft, padding with zeros so that
length(X) is equal to nfft
X = fft(x,nfft);
% FFT is symmetric, throw away second
half
X = X(1:nfft/2);
% Take the magnitude of fft of x
mx = abs(X);
% This is an evenly spaced frequency
vector
f = (0:nfft/2-1)*Fs/nfft;
% Generate the plot, title and labels.
figure(1);
plot(t,x);
title('Exponential Decay Signal');
xlabel('Time (s)');
ylabel('Amplitude');
figure(2);
plot(f,mx);
title('Power Spectrum of Exponential
Decay Signal');
xlabel('Frequency (Hz)');
ylabel('Power');
```

Example 8: Chirp Signal






```
Fs = 200; % Sampling frequency
t = 0:1/Fs:1; % Time vector of 1 second
x = chirp(t,0,1,Fs/6);
nfft = 1024; % Length of FFT
% Take fft, padding with zeros so that
length(X) is equal to nfft
X = fft(x,nfft);
% FFT is symmetric, throw away second half
X = X(1:nfft/2);
% Take the magnitude of fft of x
mx = abs(X);
% This is an evenly spaced frequency
vector
f = (0:nfft/2-1)*Fs/nfft;
% Generate the plot, title and labels.
figure(1);
plot(t,x);
title('Chirp Signal');
xlabel('Time (s)');
ylabel('Amplitude');
figure(2);
plot(f,mx);
title('Power Spectrum of Chirp Signal');
xlabel('Frequency (Hz)');
ylabel('Power');
```