% script: fftex2.m

%------------------------------------------------------------------

%

% PURPOSE

% Example of fft of timesignal y(t)- discrete Fourier transform

%

% REFERENCES

% P-E Austrell 2018-02-01

%-------------------------------------------------------------------

dt=1/100;endt=4;

t=0:dt:endt;

y=12\*sin(2\*pi\*t)-6\*sin(2\*2\*pi\*t)+4\*sin(3\*2\*pi\*t)-3\*sin(4\*2\*pi\*t);

[freq,yfft]=fftoperator(t,y);

% Plot time signal

subplot (2,1,1)

plot(t,y);

grid on

xlabel('Time [s]')

ylabel('Amplitude')

% Plot FFT

subplot(2,1,2)

plot(freq,yfft)

grid on

xlabel('Frequency [Hz]')

ylabel('Amplitude')

axis([0 10 0 14])

%-------- end----------------

 Results:

function [freq,Yfft]=fftoperator(time,y)

%

%[freq,Yfft]=fftoperator(time,y)

% --------------------------------------------------------------------------

%

% PURPOSE

% Obtain fft of timesignal y(t)- discrete fourier transform

%

% INPUT:

% time: row or column vector with n elements with constant increment

% y: corresponding row or column vector with n elements of signal

%

% OUTPUT:

% freq: row or column vector with n elements with constant

% increment [Hz]

% yfft: corresponding row or column vector with n elements of

% amplitudes

%

% REM.

% Only the unique points are returned i.e. f lies in 0<=f<=Fs/2

%

% REFERENCES

% Juan Negreira; May 2011 mod. P-E Austrell 2018-02-01

%--------------------------------------------------------------------------

%

% Introducing the time signal

%

xData=time;

yData=y;

%Calculating the FFT

%

%Number of points in input data

NFFT=length(yData);

%Nyquist frequency

Fn=1/(xData(2)-xData(1))/2;

%Absolute value of the FRF

FFTY=abs(fft(yData));

NumUniquePts=ceil((NFFT+1)/2 ) ;

% fft s~~tric, t hrow away second half

FFTY=FFTY(1:NumUniquePts) ;

% Take magnitude of Y

Yfft=abs(FFTY);

% Mult1ply by 2 to take into account the fact that we

% threw out second half of FFTY above

Yfft=Yfft\*2 ;

% Account for endpoint uniqueness

Yfft(1)=Yfft(1)/2;

% We know NFFT is even

Yfft(length(Yfft))=Yfft(length(Yfft))/2;

% Scale the FFT so that it is not a function of the length of y.

Yfft=Yfft/length(yData);

%Frequencies

freq=(0:NumUniquePts-1)\*2\*Fn/NFFT;