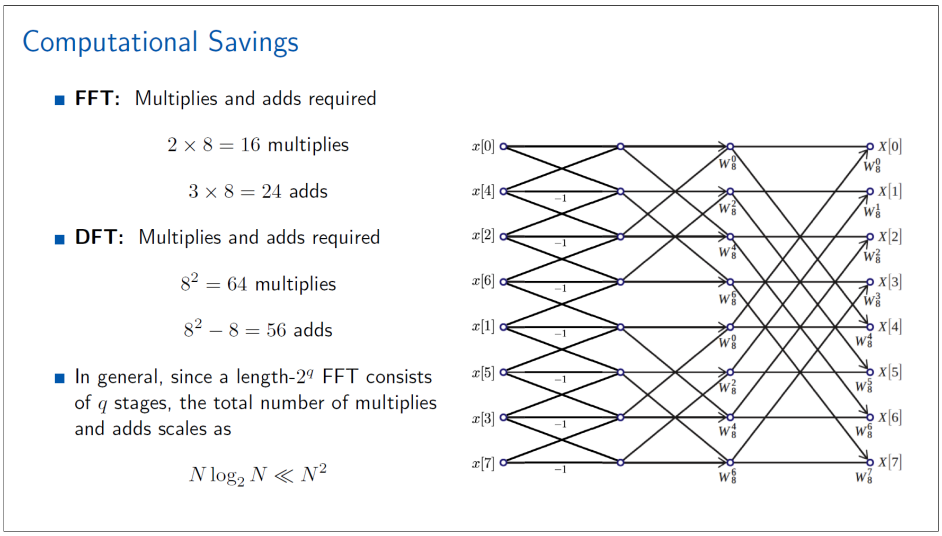
Fast Fourier Transformation (FFT)

Prof. Seungchul Lee
iSystems Design Lab
UNIST
http://isystems.unist.ac.kr/

Table of Contents



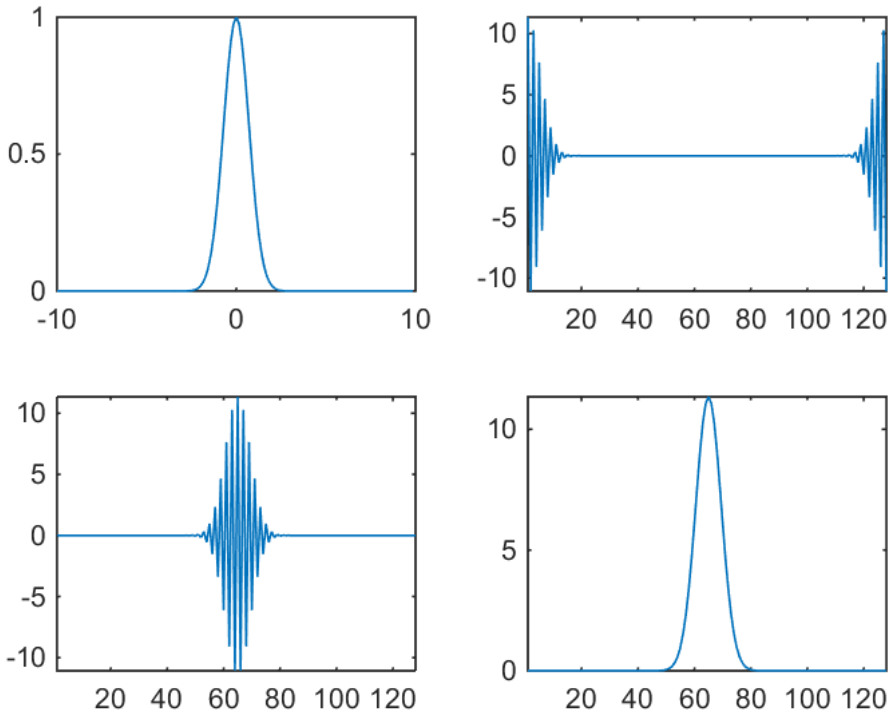
1. Fourier Transformation from the University of Washington

- Computational Methods for Data Analysis
- By Nathan Kutz At University of Washington
- Should watch: https://youtu.be/EIFrNGgbilE?list=PLBD_gON7g_m0HNsf7G3dzow2mDi5_Vy2H (https://youtu.be/EIFrNGgbilE?list=PLBD_gON7g_m0HNsf7G3dzow2mDi5_Vy2H)

1.1. Implementing FFT routine

In [1]:

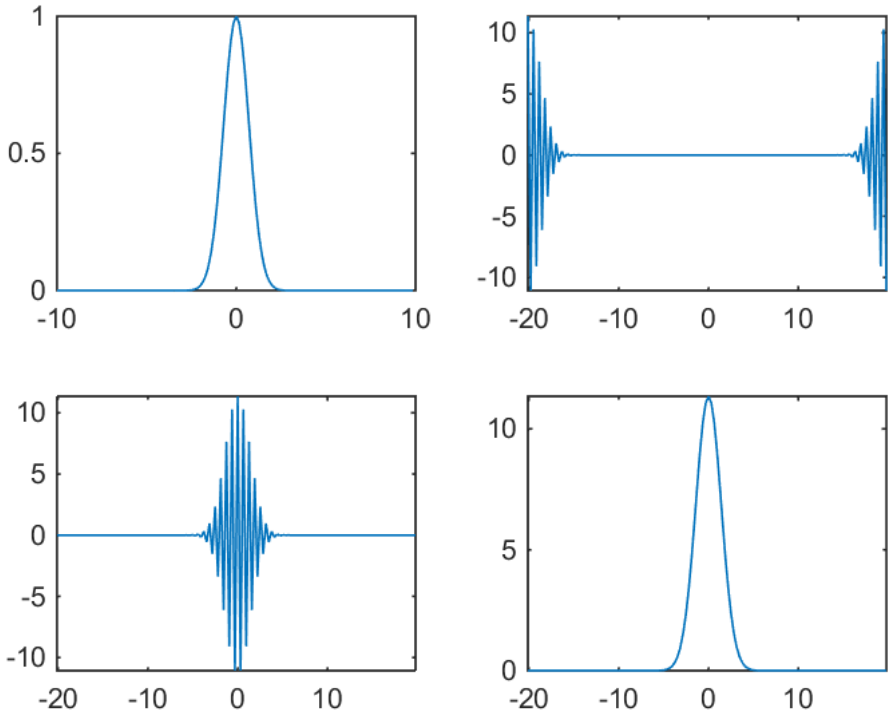
```
%plot -s 800,600
L = 20;
n = 2^7;

x2 = linspace(-L/2,L/2,n+1);
x = x2(1:n);

u = exp(-x.*x);
ut = fft(u);
utshift = fftshift(ut);

subplot(2,2,1), plot(x,u)
subplot(2,2,2), plot(real(ut)), axis tight
subplot(2,2,3), plot(real(utshift)), axis tight
subplot(2,2,4), plot(abs(utshift)), axis tight
```



Out[1]:

In [2]:

```
%plot -s 800,600
L = 20;
n = 2^7;

x2 = linspace(-L/2,L/2,n+1);
x = x2(1:n);

k = (2*pi/L)*[0:n/2-1 -n/2:-1];

u = exp(-x.*x);
ut = fft(u);
utshift = fftshift(ut);

subplot(2,2,1), plot(x,u)
subplot(2,2,2), plot(fftshift(k),real(ut)), axis tight
subplot(2,2,3), plot(fftshift(k),real(utshift)), axis tight
subplot(2,2,4), plot(fftshift(k),abs(utshift)), axis tight
```

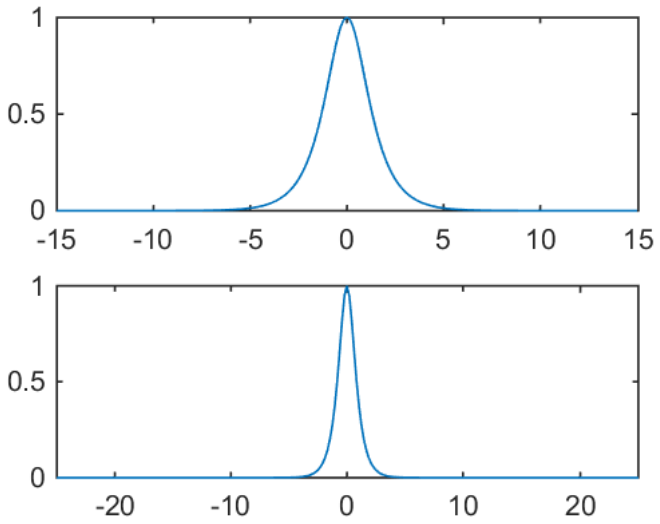


Out[2]:

2. FFT for Radar

In [3]:

```
%plot -s 560,420
T = 30;
n = 2^9;

t2 = linspace(-T/2,T/2,n+1); t = t2(1:n);

k = (2*pi/T)*[0:n/2-1 -n/2:-1]; ks = fftshift(k);

u = sech(t);
ut = fft(u);
utshift = fftshift(ut);

subplot(2,1,1), plot(t,u);
subplot(2,1,2), plot(ks,abs(utshift)/max(abs(utshift)));
axis([-25 25 0 1])
```

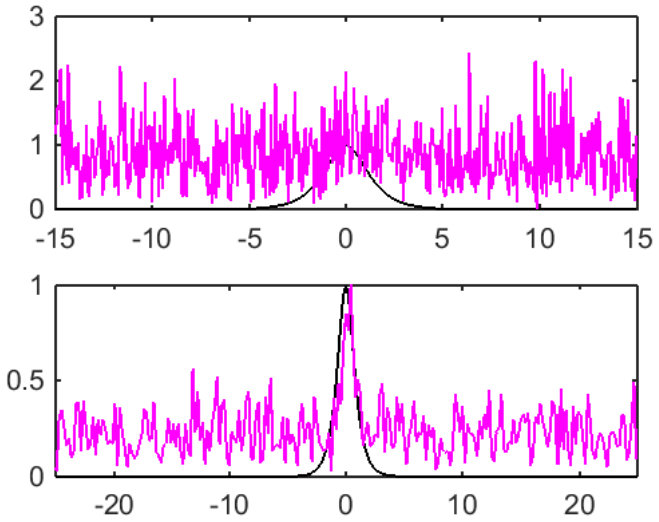


Out[3]:

2.1. White noise on signals

In [2]:

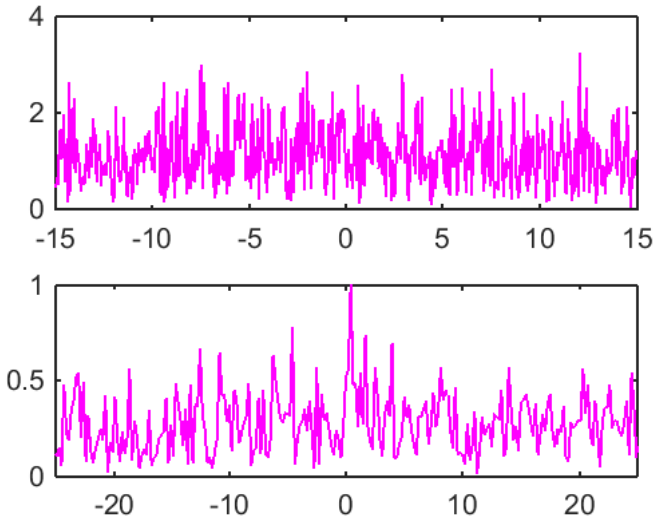
```
%plot -s 560,420
T = 30;
n = 2^9;

t2 = linspace(-T/2,T/2,n+1); t = t2(1:n);

k = (2*pi/T)*[0:n/2-1 -n/2:-1]; ks = fftshift(k);

u = sech(t);
ut = fft(u);

noise = 15;
utn = ut + noise*(randn(1,n) + 1j*randn(1,n)); % in Fourier domain
un = ifft(utn);

utshift = fftshift(ut);
utnshift = fftshift(utn);

subplot(2,1,1), plot(t,u,'k',t,abs(un),'m')
subplot(2,1,2), plot(ks,abs(utshift)/max(abs(utshift)),'k',ks,abs(utnshift)/max(abs(utnshift)),'m');
axis([-25 25 0 1])
```



Out[2]:

In [3]:

```
%plot -s 560,420
T = 30;
n = 2^9;

t2 = linspace(-T/2,T/2,n+1); t = t2(1:n);

k = (2*pi/T)*[0:n/2-1 -n/2:-1]; ks = fftshift(k);

u = sech(t);
ut = fft(u);

noise = 20;
utn = ut + noise*(randn(1,n) + 1j*randn(1,n)); % in Fourier domain
un = ifft(utn);

utshift = fftshift(ut);
utnshift = fftshift(utn);

subplot(2,1,1), plot(t,abs(un),'m')
subplot(2,1,2), plot(ks,abs(utnshift)/max(abs(utnshift)),'m');
axis([-25 25 0 1])
```

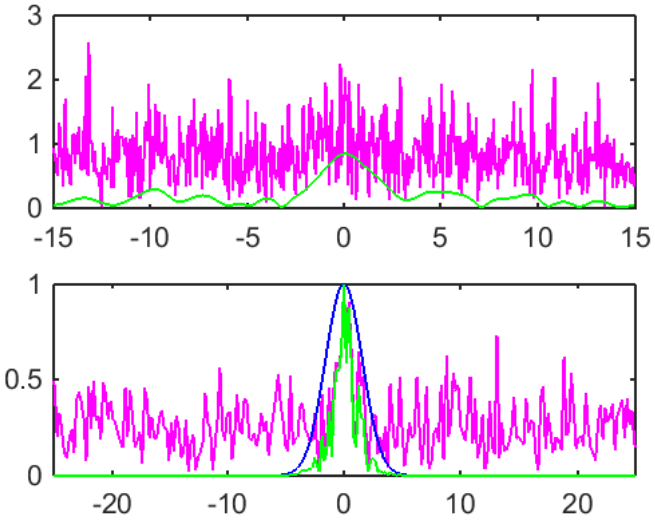


Out[3]:

2.2. Denoising via filtering

In [4]:

```
%plot -s 560,420
T = 30;
n = 2^9;

t2 = linspace(-T/2,T/2,n+1); t = t2(1:n);

k = (2*pi/T)*[0:n/2-1 -n/2:-1]; ks = fftshift(k);

u = sech(t);
ut = fft(u);

noise = 15;
utn = ut + noise*(randn(1,n) + 1j*randn(1,n)); % in Fourier domain
un = ifft(utn);

filter = exp(-0.2*k.^2);
filtershift = fftshift(filter);

utnf = filter.*utn;
unf = ifft(utnf);

utshift = fftshift(ut);
utnshift = fftshift(utn);
utnfshift = fftshift(utnf);

subplot(2,1,1), plot(t,abs(un),'m',...
                     t,abs(unf),'g')
subplot(2,1,2), plot(ks,abs(utnshift)/max(abs(utnshift)),'m',...
                     ks,filtershift,'b',...
                     ks,abs(utnfshift)/max(abs(utnfshift)),'g');
axis([-25 25 0 1])
```

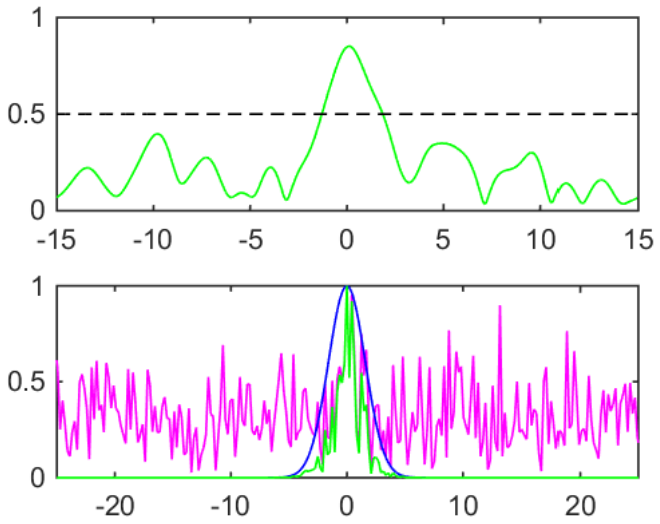


Out[4]:

2.3. Signal detection and thresholding

In [7]:

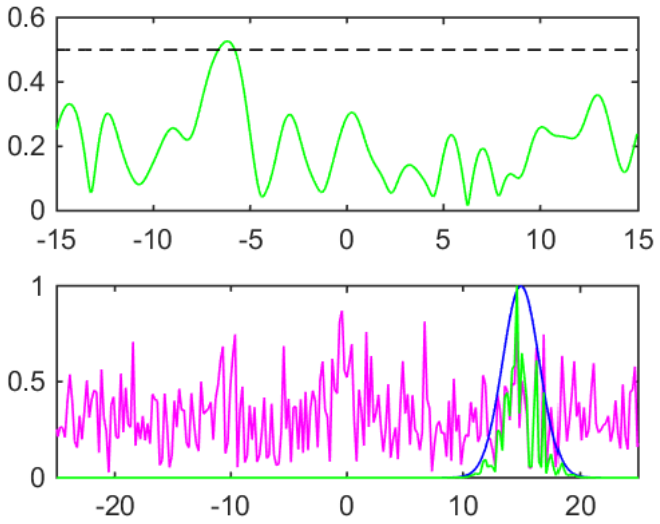
```
%plot -s 560,420
T = 30;
n = 2^9;

t2 = linspace(-T/2,T/2,n+1); t = t2(1:n);

k = (2*pi/T)*[0:n/2-1 -n/2:-1]; ks = fftshift(k);

u = sech(t);
ut = fft(u);

noise = 20;
utn = ut + noise*(randn(1,n) + 1j*randn(1,n)); % in Fourier domain
un = ifft(utn);

filter = exp(-0.2*k.^2);
filtershift = fftshift(filter);

utnf = filter.*utn;
unf = ifft(utnf);

utshift = fftshift(ut);
utnshift = fftshift(utn);
utnfshift = fftshift(utnf);

subplot(2,1,1), plot(t,abs(unf),'g',t,0*t+0.5,'k--')
subplot(2,1,2), plot(ks,abs(utnshift)/max(abs(utnshift)),'m',...
                     ks,filtershift,'b',...
                     ks,abs(utnfshift)/max(abs(utnfshift)),'g');
axis([-25 25 0 1])
```



Out[7]:

In [8]:

```
%plot -s 560,420
T = 30;
n = 2^9;

t2 = linspace(-T/2,T/2,n+1); t = t2(1:n);

k = (2*pi/T)*[0:n/2-1 -n/2:-1]; ks = fftshift(k);

u = sech(t);
ut = fft(u);

noise = 20;
utn = ut + noise*(randn(1,n) + 1j*randn(1,n)); % in Fourier domain
un = ifft(utn);

filter = exp(-0.2*(k-15).^2);
filtershift = fftshift(filter);

utnf = filter.*utn;
unf = ifft(utnf);

utshift = fftshift(ut);
utnshift = fftshift(utn);
utnfshift = fftshift(utnf);

subplot(2,1,1), plot(t,abs(unf),'g',t,0*t+0.5,'k--')
subplot(2,1,2), plot(ks,abs(utnshift)/max(abs(utnshift)),'m',...
                     ks,filtershift,'b',...
                     ks,abs(utnfshift)/max(abs(utnfshift)),'g');
axis([-25 25 0 1])
```



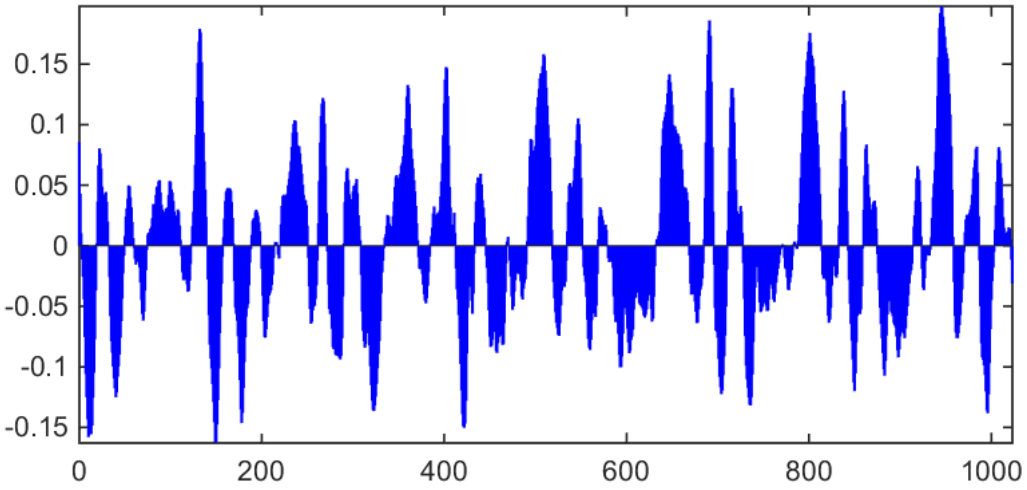
Out[8]:

3. DFT (or FFT) Example

- edX

In [9]:

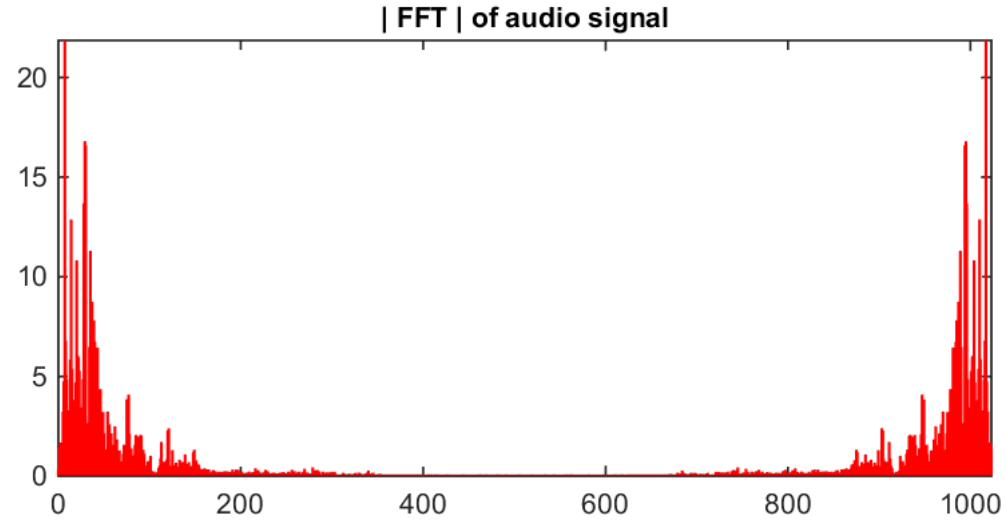
```
%plot -s 900,400

% DFTexamples_MatlabDEMO.m
% Matlab generated figures for the lecture "DFT examples"
% richb, February 2014

load hamlet
x = hamlet(58000:58000+1023);

% EXAMPLE -- DFT of a speech signal

n = 0:length(x)-1;
stem(n,x,'b','Marker','none','LineWidth',1); axis tight
```



Out[9]:

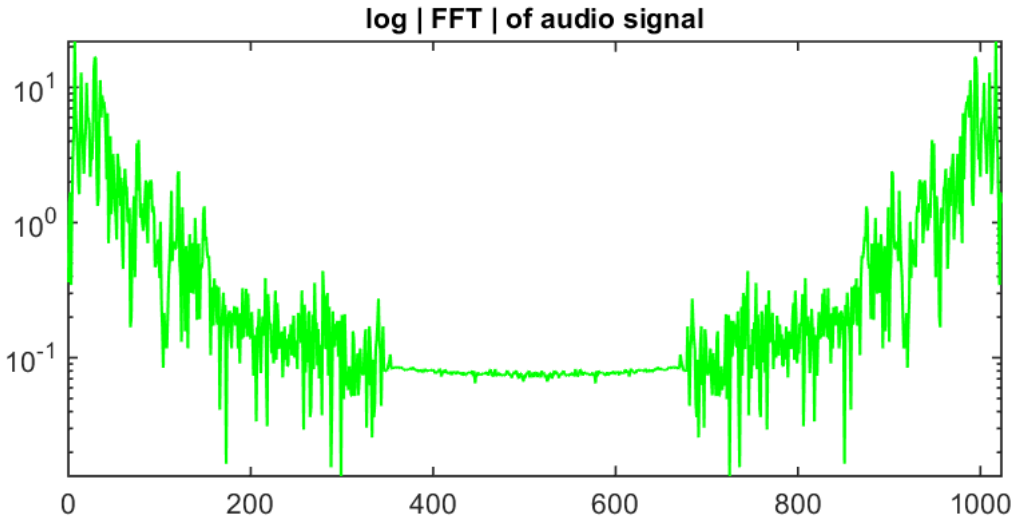
In [10]:

```
X = fft(x);
stem(n,abs(X),'r','Marker','none','LineWidth',1); axis tight
title('| FFT | of audio signal','fontsize',10)
```



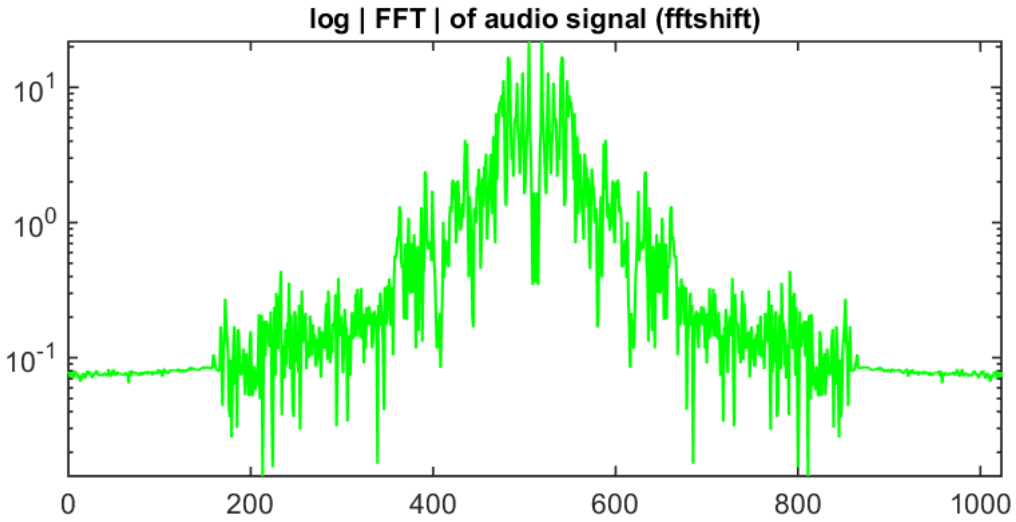
Out[10]:

In [11]:
```
semilogy(n,abs(X),'g','Marker','none','LineWidth',1); axis tight
title('log | FFT | of audio signal','fontsize',10)
```



Out[11]:

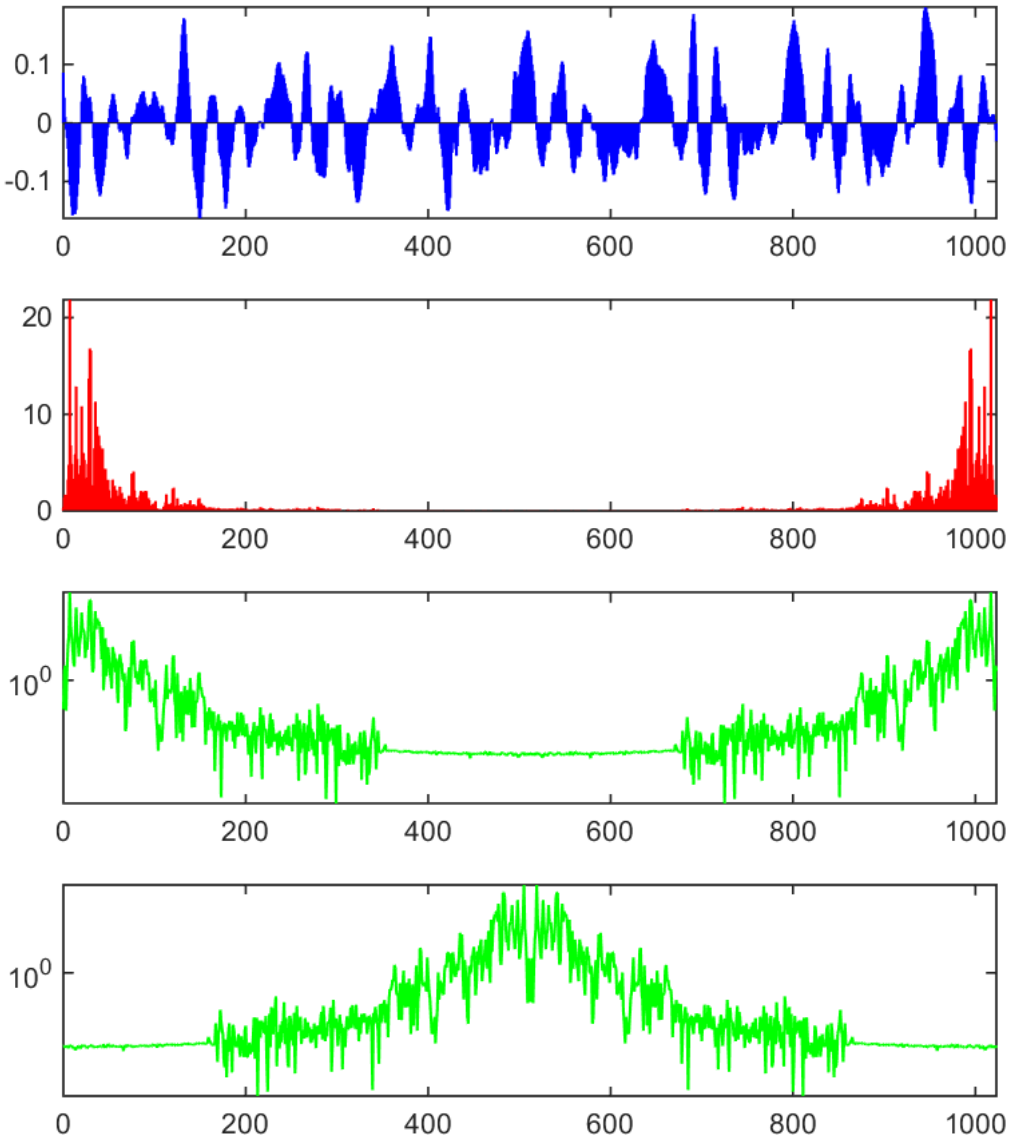
In [12]:
```
semilogy(n,fftshift(abs(X)),'g','Marker','none','LineWidth',1); axis tight
title('log | FFT | of audio signal (fftshift)','fontsize',10)
```



Out[12]:

In [13]:

```
%plot -s 900,1000

subplot(411)
stem(n,x,'b','Marker','none','LineWidth',1); axis tight
% title('Snippet of audio signal','fontsize',18)

X = fft(x);

subplot(412)
stem(n,abs(X),'r','Marker','none','LineWidth',1); axis tight
%title('| FFT | of audio signal','fontsize',18)

subplot(413)
semilogy(n,abs(X),'g','Marker','none','LineWidth',1); axis tight
%title('log | FFT | of audio signal','fontsize',18)

subplot(414)
semilogy(n,fftshift(abs(X)),'g','Marker','none','LineWidth',1); axis tight
%title('log | FFT | of audio signal (fftshift)','fontsize',18)
```



Out[13]:

In [1]:

```
%%javascript
$.getScript('https://kmahelona.github.io/ipython_notebook_goodies/ipython_notebook_toc.js')
```