Laboratorio di Immagini

Esercitazione 3:

Fourier Transform

Mauro Zucchelli
29/03/2016

Chi è Fourier

Fonte: Wikipedia

- Jean Baptiste Joseph Fourier (1768 - 1830)
- Matematico francese
- Ha praticamente fondato l'elaborazione dei segnali (senza saperlo)

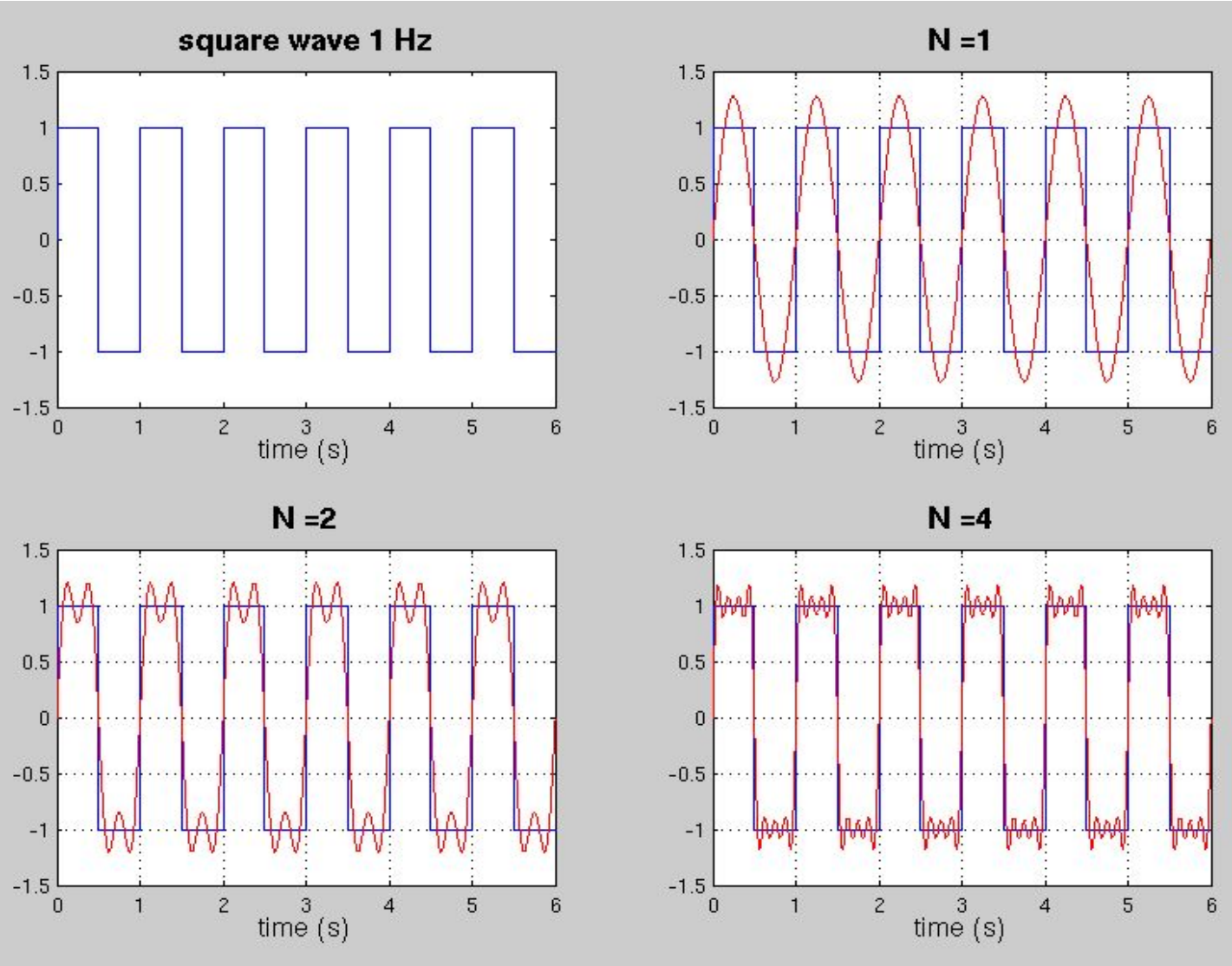
La serie di Fourier

- **Fourier** ebbe la brillante idea di rappresentare funzioni continue e periodiche come una **sommatoria** di funzioni trigonometriche (seni e coseni)

$$f(x) = \frac{a_0}{2} \sum_n \left[a_n cos(nx) + b_n sin(nx) \right]$$

- a_n e b_n sono i **coefficienti** della serie di Fourier

La serie di Fourier: esempio



La serie di Fourier

- Grazie alla Formula di Eulero:

$$e^{ix} = cos(x) + i\ sin(x)$$

- Possiamo scrivere la serie di Fourier in maniera più compatta

$$f(x) = \sum_{n} c_n e^{inx}$$

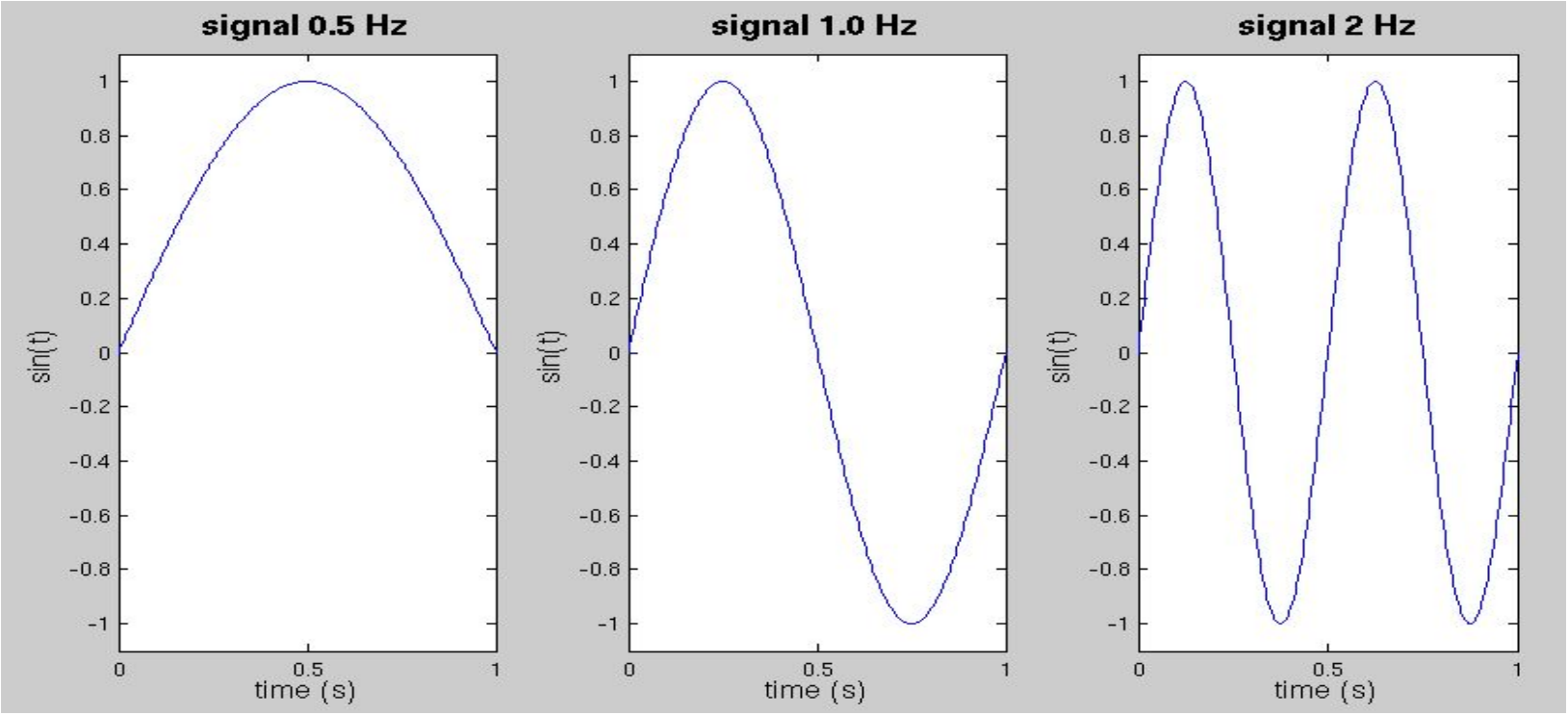
La serie di Fourier

- Per ottenere i coefficienti c_n basta calcolare il prodotto scalare tra la funzione della base di Fourier e la funzione considerata

$$c_n = \frac{1}{2\pi} \int f(x) e^{-inx} dx$$

La trasformata di Fourier

- Visto che “n” è argomento di funzioni trigonometriche possiamo considerarlo come se fosse una frequenza
- Le frequenze ovviamente sono continue (es. ω=0.5 Hz)



La trasformata di Fourier

- Studiando i coefficienti della serie di Fourier possiamo vedere quindi quali “frequenze” sono predominanti nel nostro segnale

$$F(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{i\omega x} dx$$

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} F(\omega) e^{-i\omega x} d\omega$$

La trasformata di Fourier in MATLAB

- In MATLAB è implementata la versione “veloce” della trasformata di Fourier discreta (per funzioni campionate)
  - **Fast Fourier Transform** (FFT)

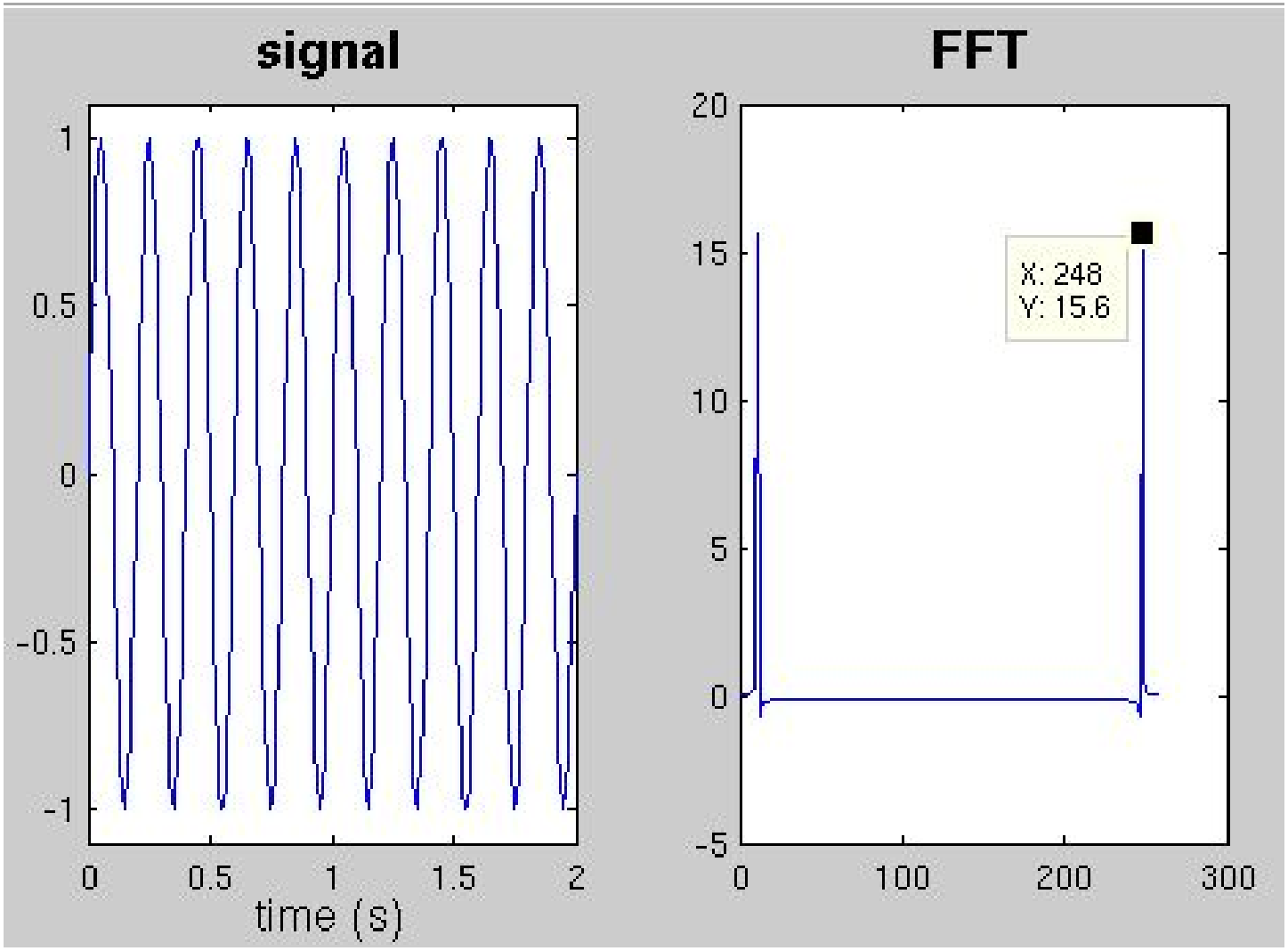
1° esempio: funzioni trigonometriche

- Visto che la trasformata di Fourier ci permette di individuare le **frequenze** di un segnale andando a "fittare" delle funzioni trigonometriche
- Se il nostro segnale è composto **esclusivamente** da funzioni trigonometriche, la nostra trasformata di Fourier sarà diversa da zero **solo** nelle frequenze corrispondenti quelle delle funzioni trigonometriche del segnale

1° esempio: funzioni trigonometriche

```
Fs = 128;       % Sampling frequency
T = 1/Fs;       % Sampling period
t = 0:T:2;      % Time vector

s = sin(2*pi*5*t);
figure
subplot(1,2,1)
plot(t,s);
xlabel('time (s)','FontSize',14)
title('signal','FontSize',14,'fontweight','bold')
ylim([-1.1,1.1])

S = real(fft(s))
subplot(1,2,2)
plot(S);
title('FFT','FontSize',14,'fontweight','bold')
```

1° esempio: funzioni trigonometriche



1° esempio: funzioni trigonometriche

- Per motivi di efficienza la FFT è ordinata in maniera un po' particolare:
  - omega= [0:n/2-1 0 -n/2+1:-1]

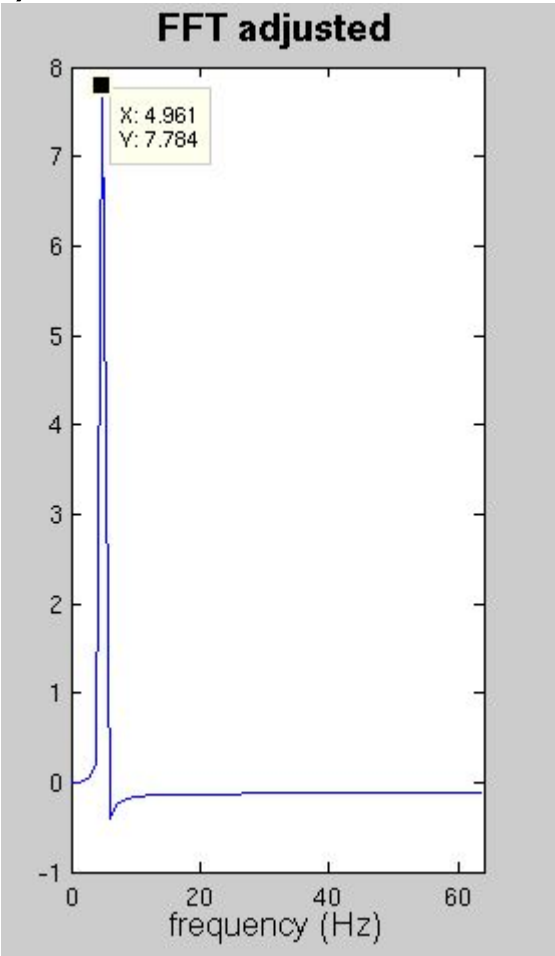
- n/2 corrisponde alla massima frequenza raggiungibile per il **teorema di Nyquist**
  - Frequenza di campionamento / 2

- Vengono incluse anche le **frequenze negative**, quindi lo spettro di Fourier viene "duplicato"

1° esempio: funzioni trigonometriche

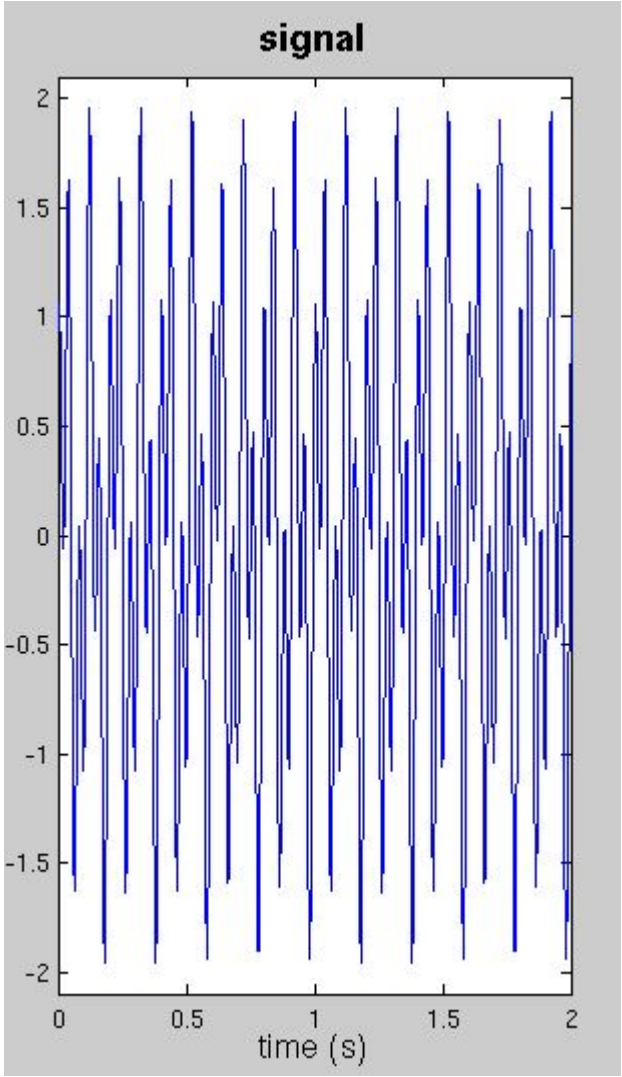
- Il grafico corretto si ottiene considerando solo la prima metà dello spettro (frequenze positive) e sistemando opportunamente l'asse delle x

```
l = length(t);
S = S(1:floor(l/2)+1);
f_max = Fs/2;
N = l/2 +1;
f_sampling = Fs/l;
f = 0:f_sampling:f_max;
subplot(1,3,3)
plot(f,S);
title('FFT adjusted','FontSize',14,'fontweight','bold')
xlabel('frequency (Hz)','FontSize',14)
xlim([0,f_max])
```



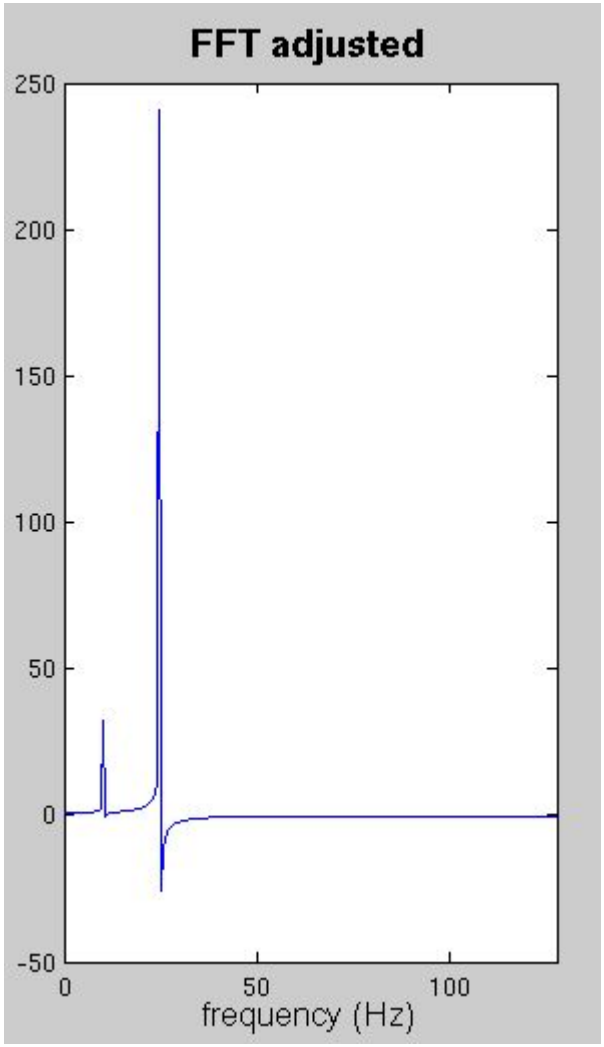
2º esempio: funzioni trigonometriche

```
Fs = 256;        T = 1/Fs;
t = 0:T:2;
s = sin(2*pi*10*t) + cos(2*pi*25*t);

figure
plot(t,s);
xlabel('time (s)','FontSize',14)
title('signal','FontSize',14,'fontweight','bold')
ylim([-1.1,1.1])
```



2º esempio: funzioni trigonometriche

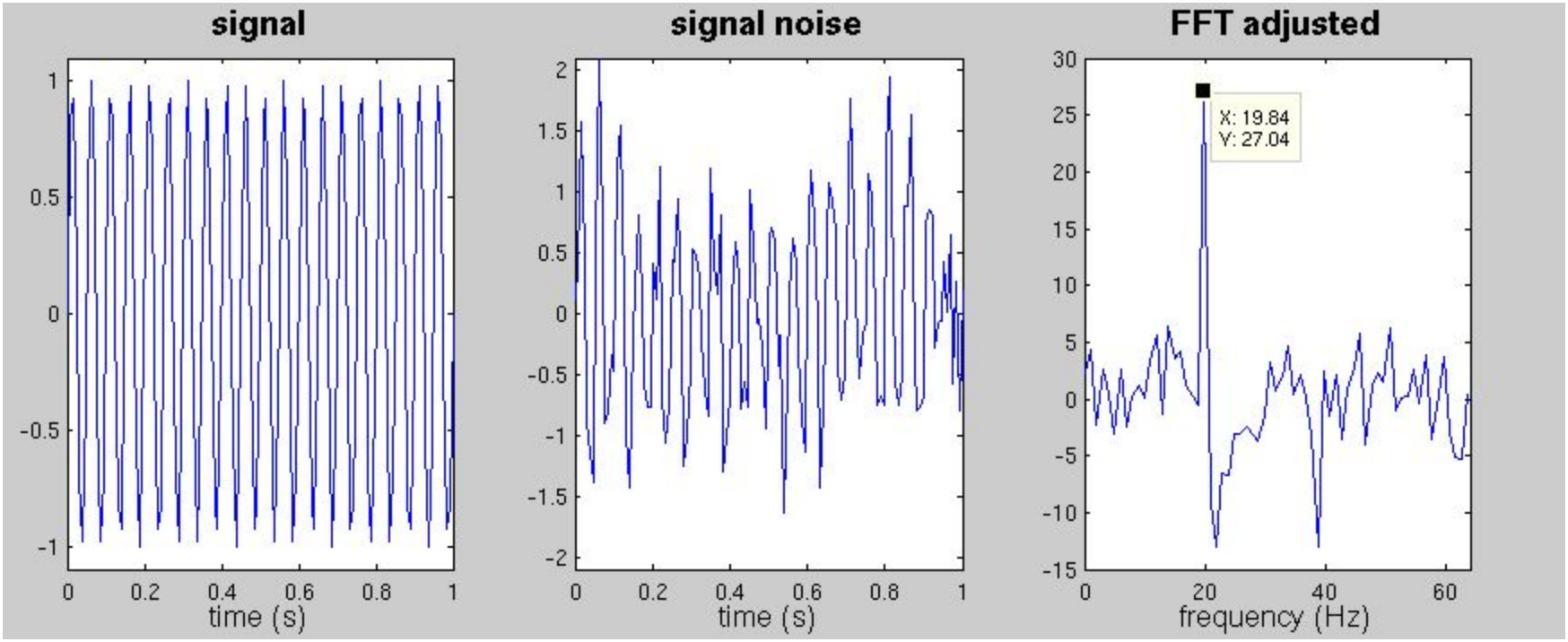
```
S = real(fft(s))

l = length(t);
S = S(1:floor(l/2)+1);
f_max = Fs/2;
N = l/2 +1;
f_sampling = Fs/l;
f = 0:f_sampling:f_max;

figure
plot(f,S);
title('FFT adjusted','FontSize',
14,'fontweight','bold')
xlabel('frequency (Hz)','FontSize',14)
xlim([0,f_max])
```



2º esempio: funzioni trigonometriche

- Una delle cose migliori della trasformata di Fourier è la sua robustezza al rumore

```
s = sin(2*pi*20*t);
s_noise = s + randn(1,length(t))*0.5;
```

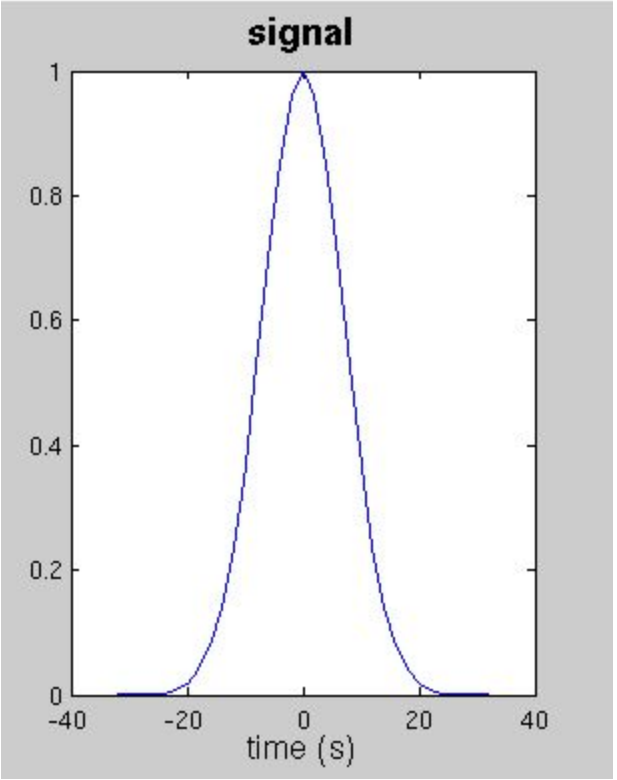


3º esempio: funzioni Gaussiane

- Una funzione Gaussiana è definita come

$$f(x) = e^{-\alpha x^2}$$

```
Fs = 0.5;
T = 1/Fs;
t = -32:T:32;
l = length(t);

alpha = 0.01;
g = exp(-alpha*t.^2 );
figure;
plot(t,g)
xlabel('time (s)','FontSize',14)
title('signal','FontSize',14,'fontweight','bold')
```



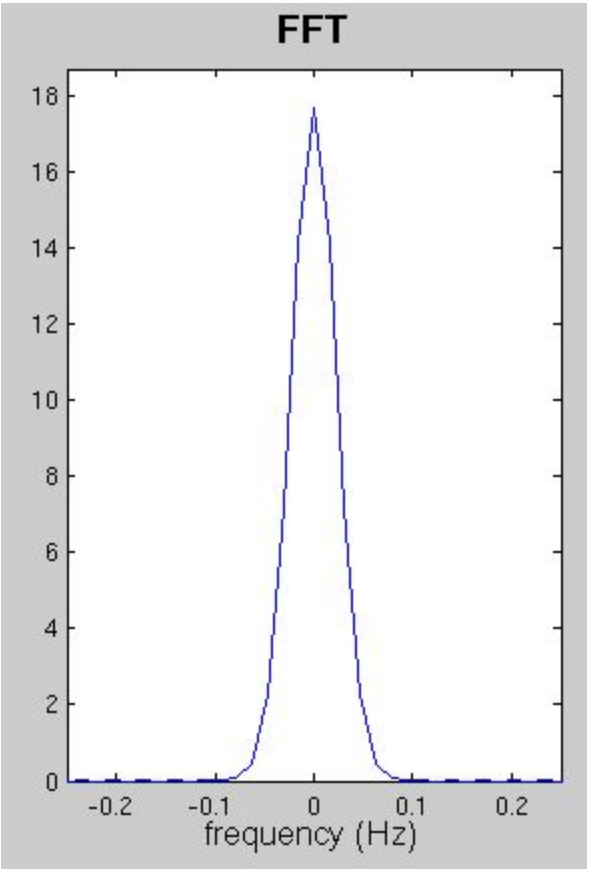
3º esempio: funzioni Gaussiane

- A causa del particolare ordinamento delle frequenze del comando FFT è necessario “riordinarle” affinché la trasformata di **funzioni simmetriche** sia calcolata in maniera corretta

- I comandi FFTSHIFT e IFFTSHIFT ricentrano e decentrano la funzione Gaussiana correttamente

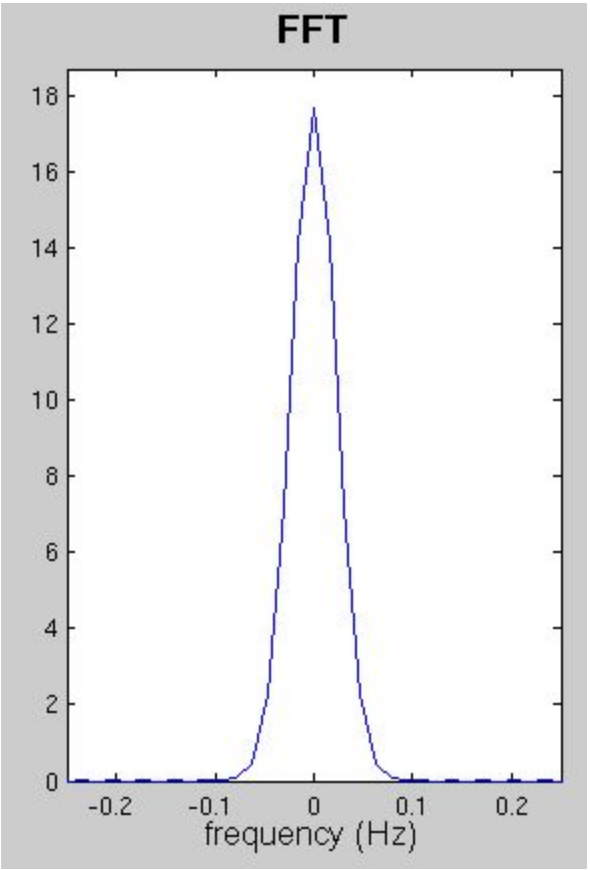
3º esempio: funzioni Gaussiane

```
G = fftshift(real(fft(ifftshift(g)))) / Fs;
f_max = Fs/2.0;
f_sampling = Fs/(l-1);
f = -f_max:f_sampling:f_max;

figure
plot(f,G)
xlabel('frequency (Hz)','FontSize',14)
title('FFT','FontSize',14,'fontweight','bold')
ylim([0,max(G)+1])
xlim([-f_max,f_max])
```



3º esempio: funzioni Gaussiane

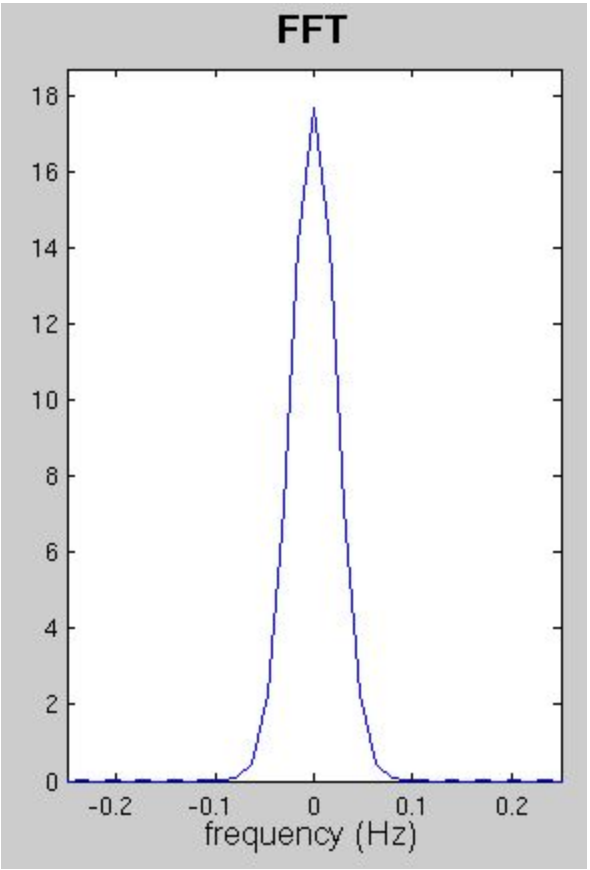
```
G = fftshift(real(fft(ifftshift(g)))) / Fs;
f_max = Fs/2.0;
f_sampling = Fs/(l-1);
f = -f_max:f_sampling:f_max;

figure
plot(f,G)
xlabel('frequency (Hz)','FontSize',14)
title('FFT','FontSize',14,'fontweight','bold')
ylim([0,max(G)+1])
xlim([-f_max,f_max])
```



La divisione per Fs è solo una correzione causata dalla discretizzazione della trasformata

3º esempio: funzioni Gaussiane

```
G = fftshift(real(fft(ifftshift(g)))) / Fs;
f_max = Fs/2.0;
f_sampling = Fs/(l-1);
f = -f_max:f_sampling:f_max;

figure
plot(f,G)
xlabel('frequency (Hz)','FontSize',14)
title('FFT','FontSize',14,'fontweight','bold')
ylim([0,max(G)+1])
xlim([-f_max,f_max])
```



La divisione per Fs è solo una correzione causata dalla discretizzazione della trasformata

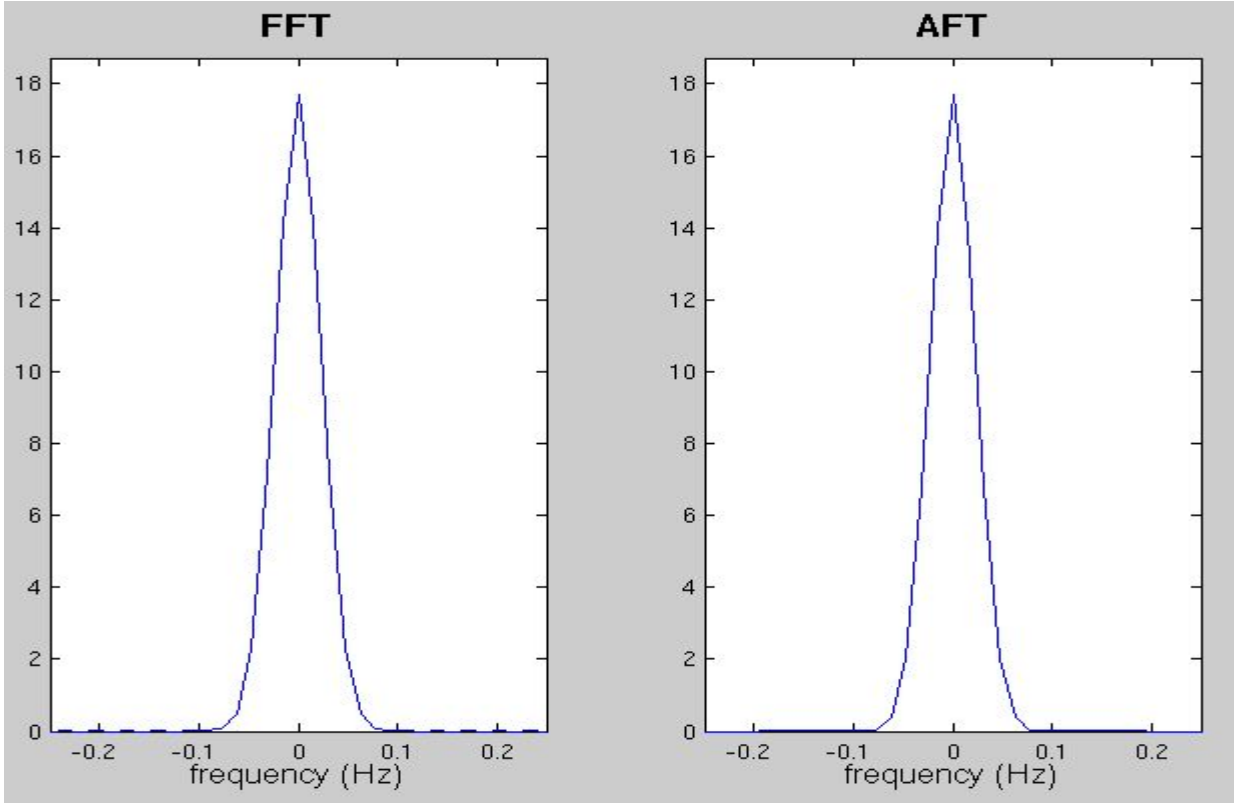
3º esempio: funzioni Gaussiane

- Possiamo confrontare il risultato della FFT con la trasformata di Fourier analitica del segnale:

$$F(\omega) = \sqrt{\frac{\pi}{\alpha}} e^{-\frac{\pi^2 \omega^2}{\alpha}}$$

3º esempio: funzioni Gaussiane

```
G2 = sqrt(pi/alpha)*exp(-pi^2 *f.^2 / alpha);
figure
plot(f,G2)
xlabel('frequency (Hz)','FontSize',14)
title('AFT','FontSize',14,'fontweight','bold')
ylim([0,max(G)+1])
xlim([-f_max,f_max])
```



Break

```
% PAUSA

pause(10*60);
```

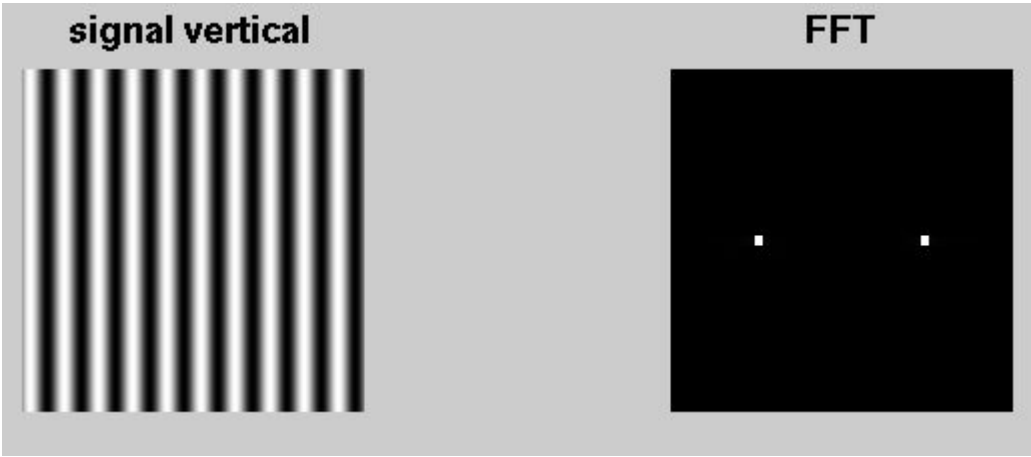
Trasformata di Fourier 2D

- La trasformata di Fourier in 2D non è altro che la trasformata di Fourier calcolata per le x (lungo le righe) e lungo le y (lungo le colonne) applicate in serie

- Per le immagini le frequenze sono intese come frequenze spaziali (**variazione di contrasto**)

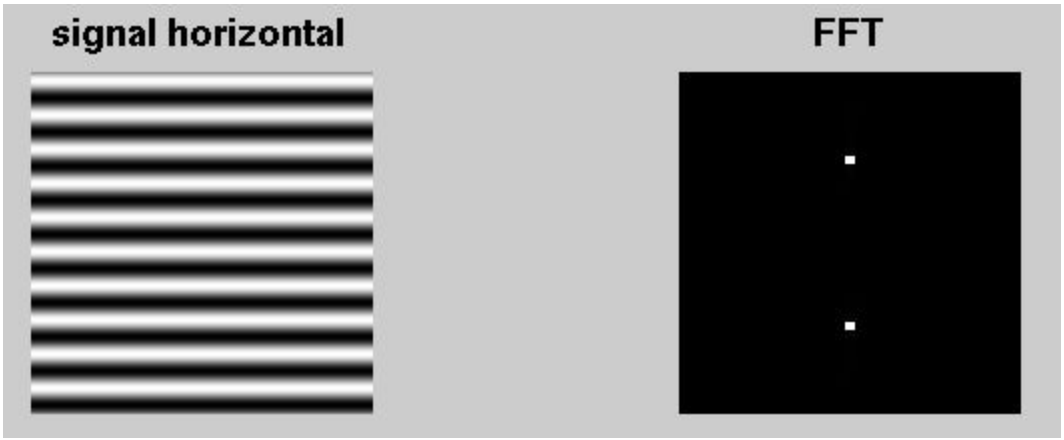
Trasformata di Fourier 2D

```
Fs = 256;
T = 1/Fs;
t = -1:T:1;

s = sin(2*pi*5*t);
s_2d = repmat(s,length(s),1);

figure;
subplot(2,2,1)
imshow(s_2d,[])
title('signal vertical','FontSize',
14,'fontweight','bold')

S_2D =abs(fftshift(fft2(ifftshift(s_2d))));
subplot(2,2,2)
imshow(S_2D(237:277,237:277),[])
title('FFT','FontSize',14,'fontweight','bold')
```



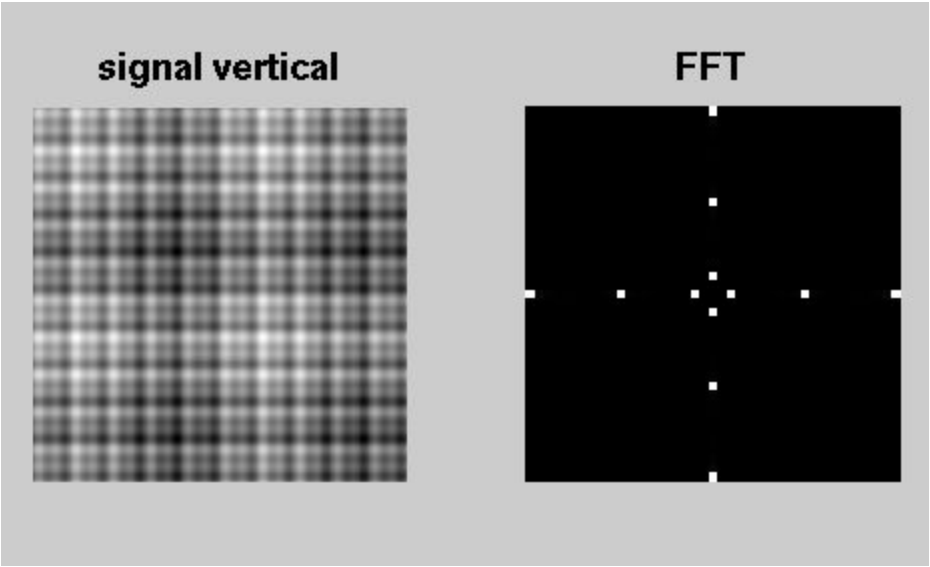
Trasformata di Fourier 2D

```
subplot(2,2,3)
imshow(s_2d',[])
title('signal horizontal','FontSize',
14,'fontweight','bold')

S_2D =abs(fftshift(fft2(ifftshift(s_2d'))));

subplot(2,2,4)
imshow(S_2D(237:277,237:277),[])
title('FFT','FontSize',14,'fontweight','bold')
```



Trasformata di Fourier 2D

```
s = sin(2*pi*5*t) + sin(2*pi*1*t) + sin(2*pi*10*t);
s_2d = repmat(s,length(s),1);
s_2d = s_2d+s_2d';
subplot(1,2,1)
imshow(s_2d,[])
title('signal vertical','FontSize',14,'fontweight','bold')
S_2D =abs(fftshift(fft2(ifftshift(s_2d))));
subplot(1,2,2)
imshow(S_2D(237:277,237:277),[])
```

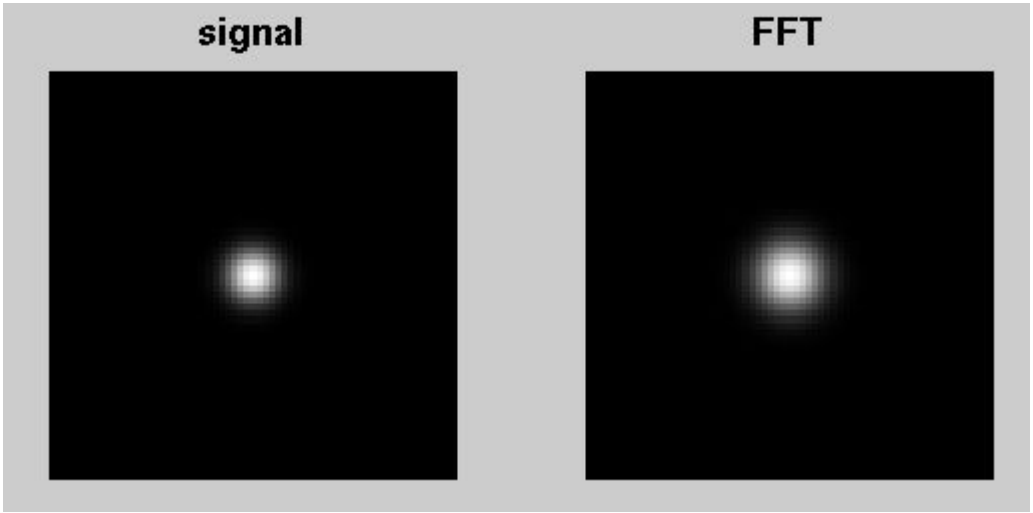


Trasformata di Fourier 2D: Gaussiana

```
function G = my_gaussian(r_max,r_sampling,alpha)

t = -r_max:r_sampling:r_max;
l = length(t);

G = zeros(l,l);

for i=1:l
    for j=1:l
        r = sqrt(t(i)^2 + t(j)^2);
        G(i,j) = exp(-alpha*r^2) ;
    end
end

end
```

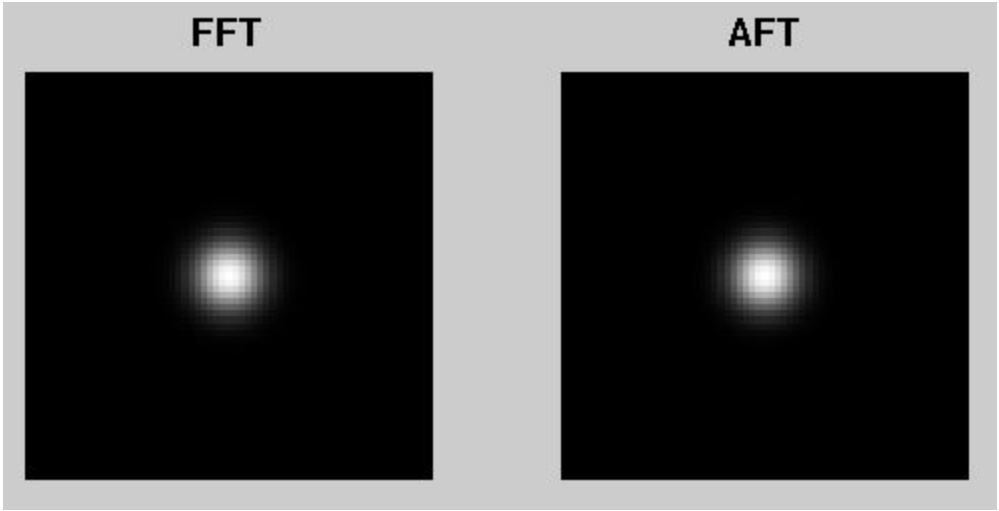
Trasformata di Fourier 2D: Gaussiana

```
t_max = 8;
Fs = 4;
alpha = 1;
g = my_gaussian(t_max, 1.0/Fs, alpha);

G = fftshift(fft2(ifftshift(g)))/Fs^2;

figure;
subplot(1,3,1)
imshow(g,[])
title('signal')

subplot(1,3,2)
imshow(G,[])
title('FFT')
```



Trasformata di Fourier 2D: Gaussiana

```
f_max = Fs/2.0;
f_sampling = Fs/(length(G)-1);

G2 = my_gaussian(f_max,f_sampling, pi^2/alpha) *
(pi/alpha);
subplot(1,3,3)
imshow(G2,[])
title('AFT','FontSize',14,'fontweight','bold')
```



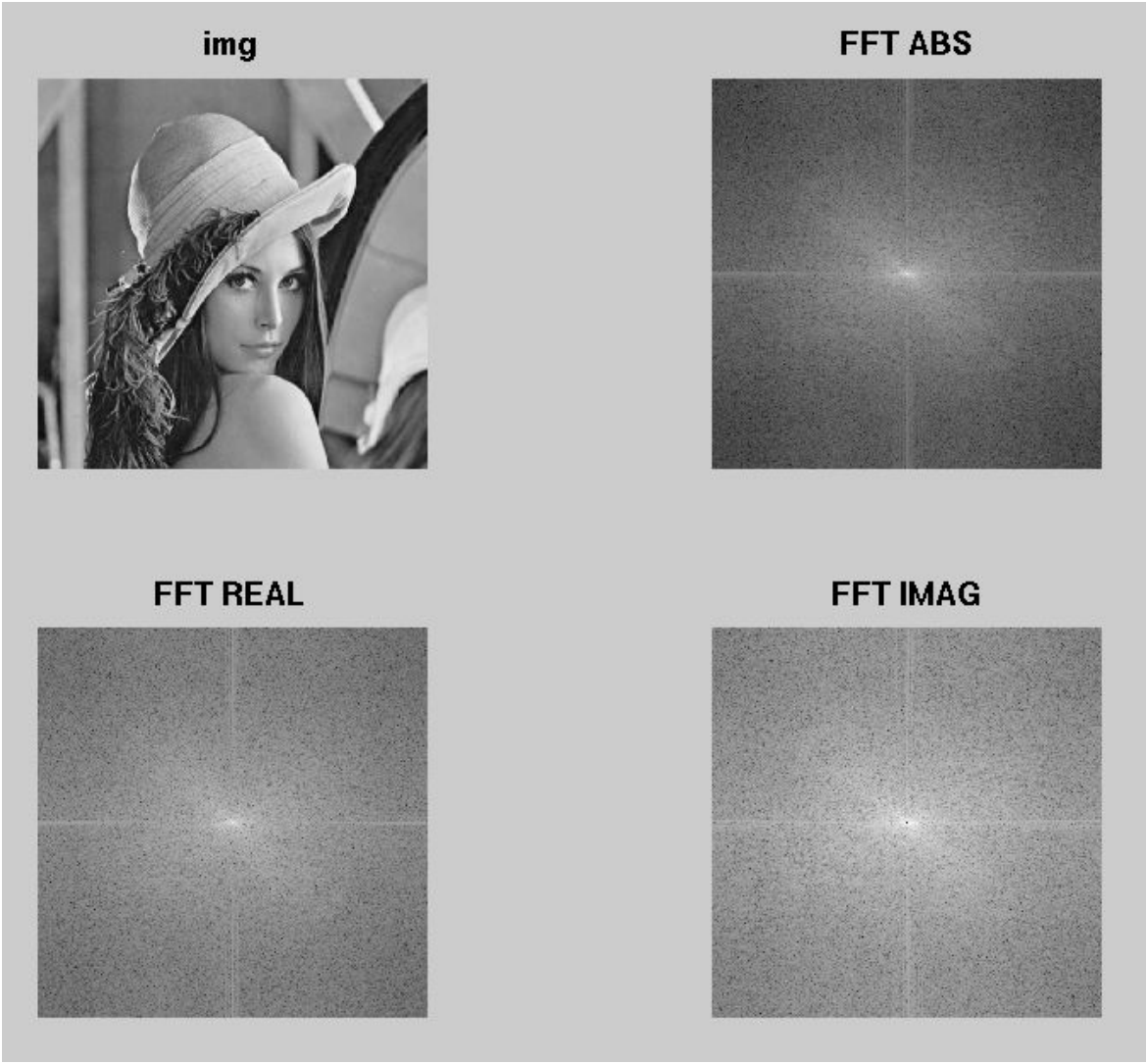
FFT per Immagini

- Caricate l'immagine 'lena.jpg' (o la vostra preferita)
- Convertitela in scale di grigio
- Provate a visualizzare:
  - Il modulo della trasformata di Fourier (*abs*)
  - La parte reale (comando *real*)
  - La parte immaginaria (comando *imag*)
- Attenzione a volte è meglio prendere il logaritmo della trasformata per visualizzare meglio il risultato

FFT per Immagini

```
img = rgb2gray(imread('lena.jpg'));
I =fftshift(fft2(img));

figure;
subplot(2,2,1)
imshow(img,[])
title('img')

subplot(2,2,2)
imshow(log(abs(I)),[])
title('FFT ABS')

subplot(2,2,3)
imshow(log(abs(real(I))),[])
title('FFT REAL')
subplot(2,2,4)

imshow(log(abs(imag(I))),[])
title('FFT IMAG')
```

FFT per Immagini



FFT per Immagini

- A cosa corrispondono la parte immaginaria e la parte reale?
- Provate ad antitrasformare l'immagine:
  - Mettendo a zero la parte immaginaria
  - Mettendo a zero la parte reale
- P.s. Ricordatevi l'ifftshift se avete fatto l'fftshift della trasformata!

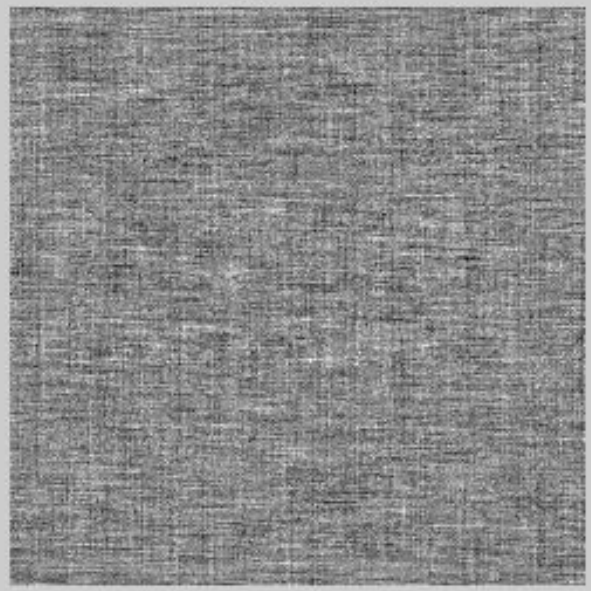
FFT per Immagini

```
img_real = ifft2(ifftshift(real(I)));

figure;
subplot(1,2,1)
imshow(img_real,[])
title('real part','FontSize',14,'fontweight','bold')


img_imag = ifft2(ifftshift(imag(I)));

subplot(1,2,2)
imshow(img_imag,[])
title('immaginary part','FontSize',
14,'fontweight','bold')
```

FFT per Immagini

real part

immaginary part



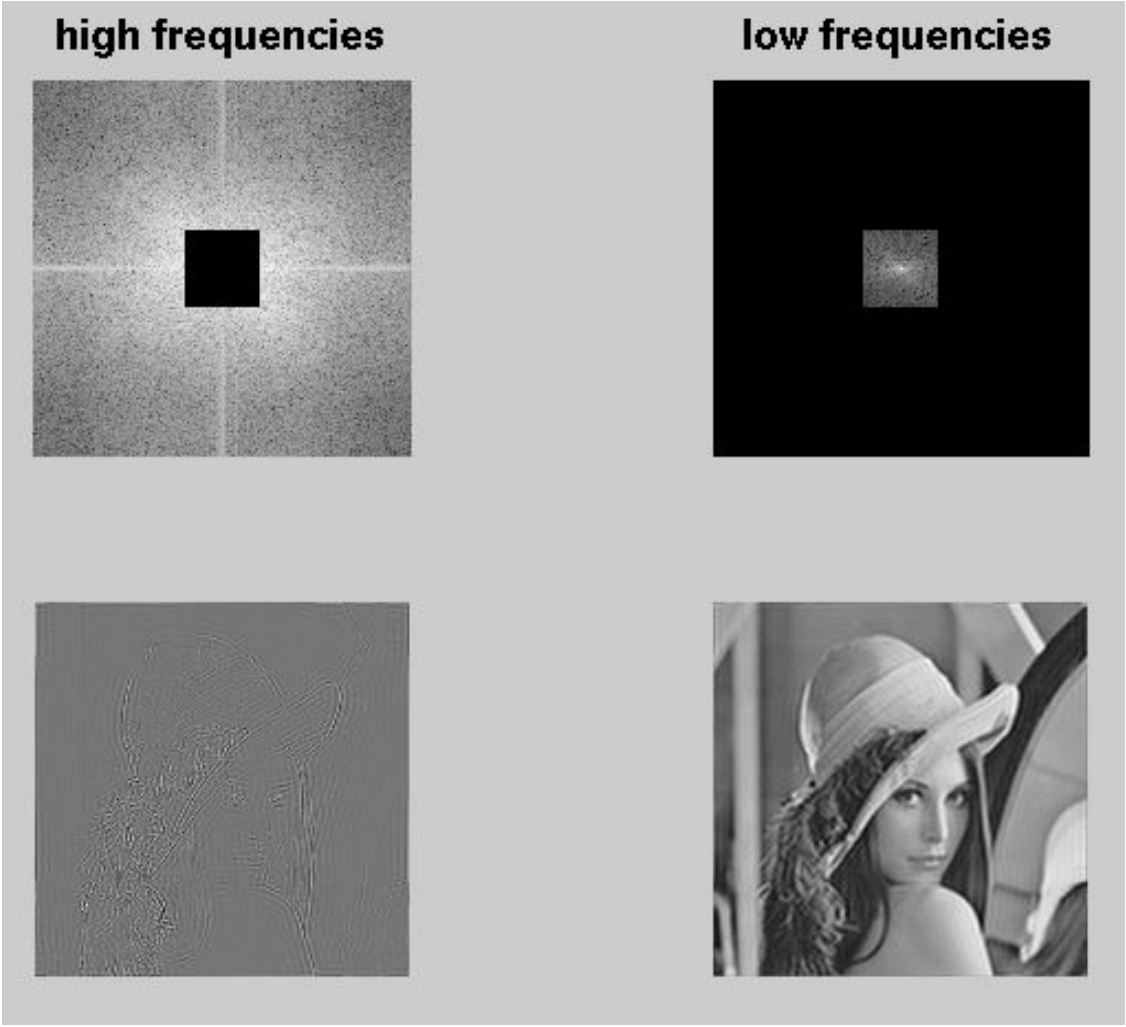
FFT per Immagini: esercizio finale!

- Se eseguite l'fftshift della trasformata al **centro** dell' immagine ci sono le basse frequenze, e verso i **bordi** le alte frequenze
- A cosa corrispondono nelle immagini? Provate a:
  - Mettere a zero il centro della trasformata (con un quadrato di lato 50)
  - Mettete a zero il contorno (il complementare del quadrato di prima)

FFT per Immagini: esercizio finale!

```
square = ones(size(img));
square(257-51:257+50,257-51:257+50)=0;

I_high = I.*square;
figure;
subplot(2,2,1)
imshow(log(abs(I_high)),[])
title('high frequencies','FontSize',14,'fontweight','bold')

I_low = I.*(1-square);
subplot(2,2,2)
imshow(log(abs(I_low)),[])
title('low frequencies','FontSize',14,'fontweight','bold')

img_high = real(ifft2(ifftshift(I_high)));
img_low= real(ifft2(ifftshift(I_low)));
subplot(2,2,3)
imshow(img_high,[])
subplot(2,2,4)
imshow(img_low,[])
```

FFT per Immagini: esercizio finale!



FFT per Immagini: esercizio finale!

FINE!

BONUS

http://9gag.com/gag/aGxOYw6