L. Magistrale in IA (ML&BD)

# Scientific Computing (part 2 – 6 credits)

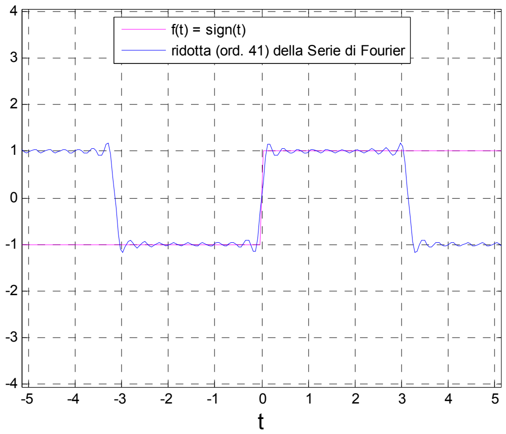

prof. Mariarosaria Rizzardi

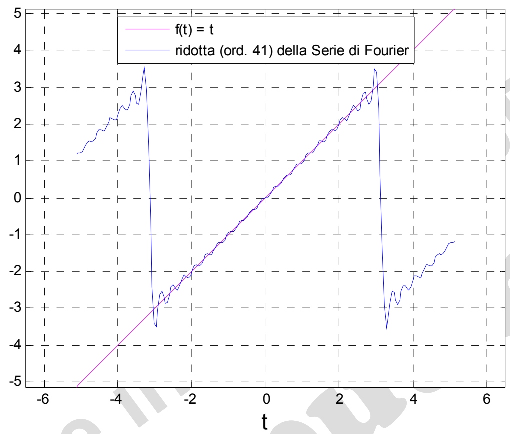
Centro Direzionale di Napoli – Bldg. C4
room: n. 423 – North Side, 4th floor
phone: 081 547 6545
email: mariarosaria.rizzardi@uniparthenope.it

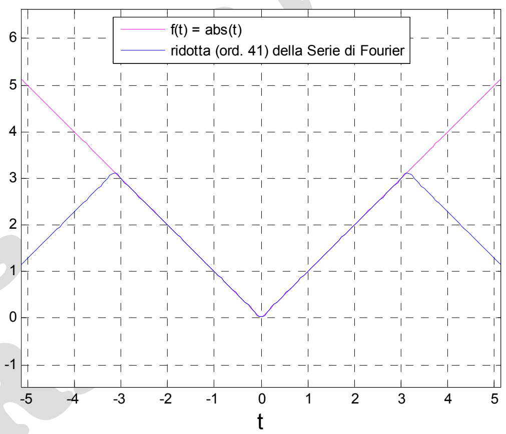
# Contents

- Fourier Transform (FT).
- Examples of Fourier Transforms.
- Fourier Transform of a Comb function.
- Superposition of functions.
- Properties of the Fourier Transform.

# Fourier Transform (FT)




If the ***Fourier Series*** converges to $f$ in an interval $[a,b]$, outside $[a,b]$ the ***Fourier Series*** converges to $f$ only if also $f$ is periodic of period $b-a$. The ***Fourier Transform*** arises from the need to approximate **non-periodic functions** on all $\mathbb{R}$.

**DEF** The ***Fourier Transform*** $F(\omega)$ of $f(t)$ is

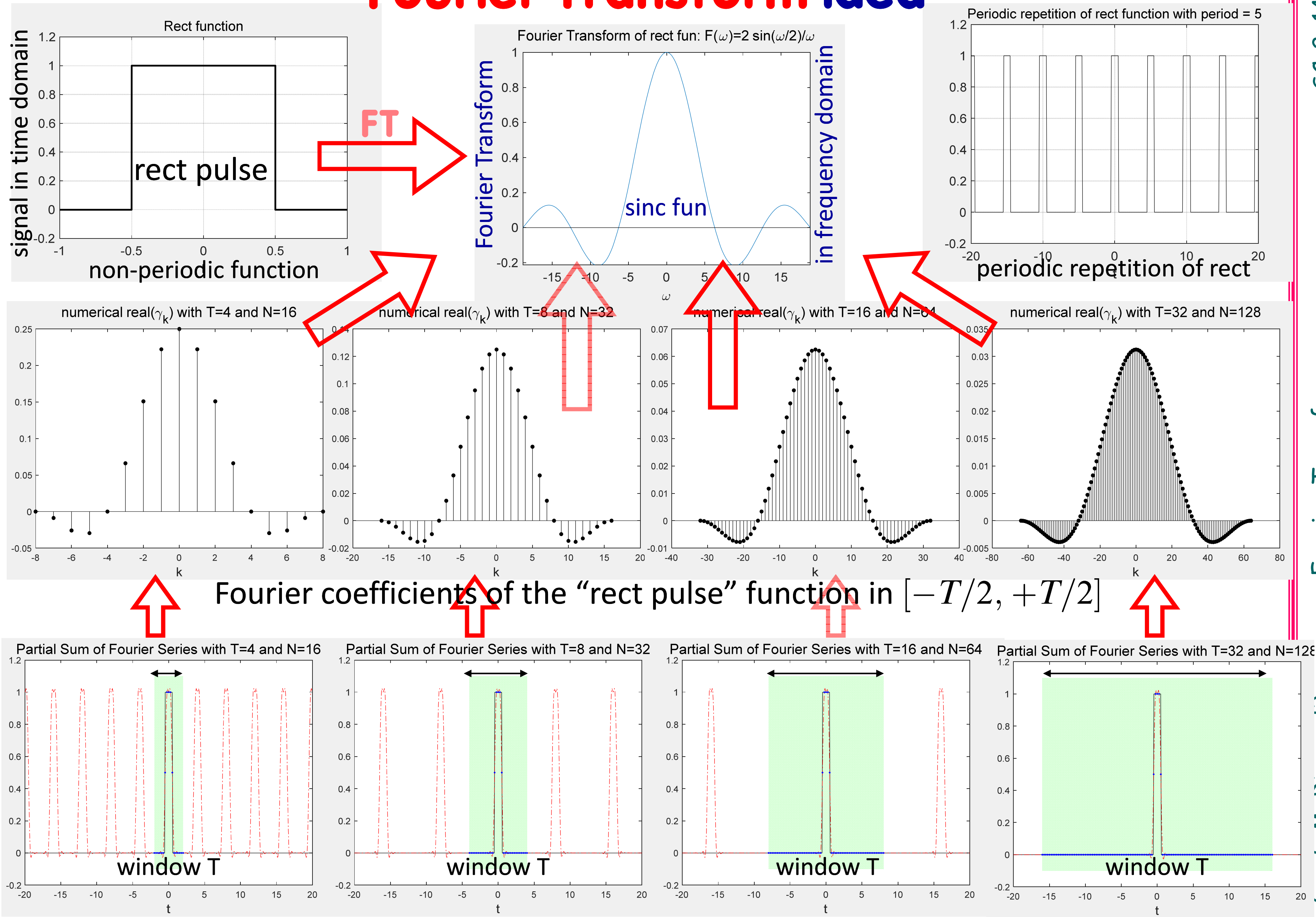
$$F(\omega) = \int_{-\infty}^{+\infty} f(t) e^{-i\omega t} \, \mathrm{d}t$$

$F(\omega)$ is a complex valued function of a real argument $\omega$

when this integral exists (i.e. it is $< \infty$).

The summability of $f$ $[f \in L^1(-\infty,+\infty)]$ represents a sufficient condition, but it is not necessary for the existence of the FT.

# Fourier Transform idea




As the window increases, in the FCs of the “rect pulse” function the frequencies are getting closer and closer together, and it looks as though the coefficients are tracking some definite curve of the FT function.

**Theorem** If $f \in L^1(-\infty,+\infty)$ and satisfies the Dirichlet conditions, then the following formulas hold

| **Fourier Transform (FT)** | complex-valued function of a real argument |
|---|---|
| $F(\omega) = \int_{-\infty}^{+\infty} f(t) e^{-i\omega t} \mathrm{d}t$ | $F(\nu) = \int_{-\infty}^{+\infty} f(t) e^{-2\pi i \nu t} \mathrm{d}t$ |
| **Inverse Fourier Transform (IFT)** | |
| $f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(\omega) e^{i\omega t} \mathrm{d}\omega$ | $f(t) = \int_{-\infty}^{+\infty} F(\nu) e^{2\pi i \nu t} \mathrm{d}\nu$ |

**angular frequency ω** **circular frequency ν**

$$\omega = 2\pi\nu$$

recap

discrete and finite case

## Discrete Fourier Transform (DFT)

$$F_k = \sum_{j=0}^{N-1} f_j \boxed{e^{-i\frac{2\pi}{N}kj}}, \qquad k = 0,\ldots,N-1$$

## Inverse Discrete Fourier Transform (IDFT)

$$f_j = \frac{1}{N}\sum_{k=0}^{N-1} F_k \boxed{e^{+i\frac{2\pi}{N}kj}}, \qquad j = 0,\ldots,N-1$$

discrete and infinite case

## Coefficients of the Fourier Series in $[-\pi,+\pi]$

$$\gamma_k = \frac{1}{2\pi}\int_{-\pi}^{+\pi} f(x) \boxed{e^{-ikx}} \, \mathrm{d}x, \qquad k = -\infty,\ldots,0,\ldots,+\infty$$

## Fourier Series (FS)

$$\sum_{k=-\infty}^{+\infty} \gamma_k \boxed{e^{+ikx}}$$

continuos case

## Fourier Transform (FT)

$$F(\omega) = \int_{-\infty}^{+\infty} f(t) \boxed{e^{-i\omega t}} dt$$

## Inverse Fourier Transform (IFT)

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

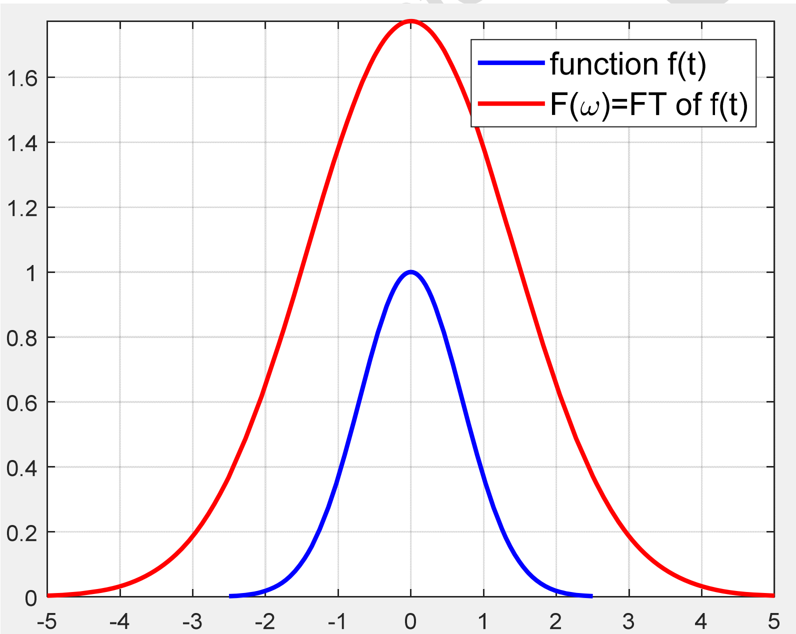
MATLAB Symbolic Math Toolbox provides the functions **fourier(…)** and **ifourier()** for the symbolic expression of the FT and of the IFT respectively.

gaussian

```
syms t real
ft=exp(-t^2);  Fw=fourier(ft)
Fw =
pi^(1/2)/exp(w^2/4)
```

```
syms t real;  ft=exp(-t^2);
Fw=fourier(ft);  Ifw=ifourier(Fw)
IFw =
1/exp(x^2)
```

both FT and IFT are gaussian



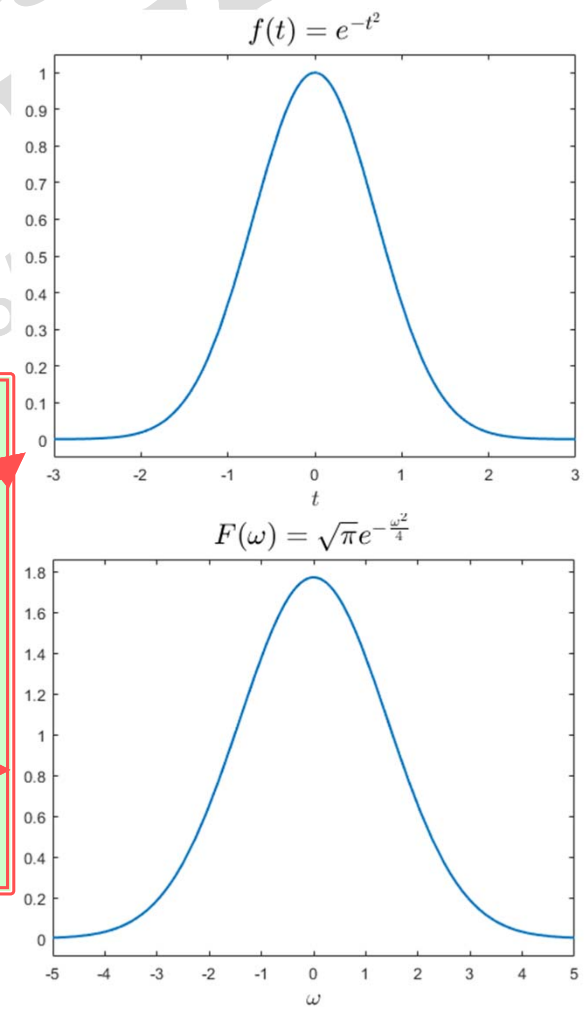

```
fplot(ft,[-5 5],'Color','b','LineWidth',2)
grid on; hold on
fplot(Fw,[-5 5],'Color','r','LineWidth',2)
legend('function f(t)', ...
     'F(\omega)=FT of f(t)','FontSize',14)
```

# Examples of Fourier Transform

"The Fourier Transform of a Gaussian is still a gaussian"

$$f(t) = e^{-|a|t^2} \quad \circ\!\!-\!\!\bullet \quad F(\omega) = \sqrt{\frac{\pi}{|a|}}\, e^{-\frac{\omega^2}{4|a|}}$$

```
syms t real; syms a positive
ft=exp(-a*t^2); % gaussian f(t)
fourier(ft)      % FT of f(t)
ans =
(pi^(1/2)*exp(-w^2/(4*a)))/a^(1/2)
ft1=subs(ft,a,1);  fplot(ft1,[-5 5])
title('$f(t)=e^{-t^2}$','FontWeight','normal','FontSize',18,'Interpreter','LaTeX')
xlabel('$t$','FontSize',14,'Interpreter','LaTeX')
fplot(fourier(ft1),[-5 5])
title('$F(\omega)=\sqrt{\pi}e^{-\frac{\omega^2}{4}}$','FontWeight','normal','FontSize',18, ...
'Interpreter','LaTeX')
xlabel('$\omega$','FontSize',14,'Interpreter','LaTeX')
```



# Example of Fourier Transform (odd function)

$$f(t) = \text{signum}$$

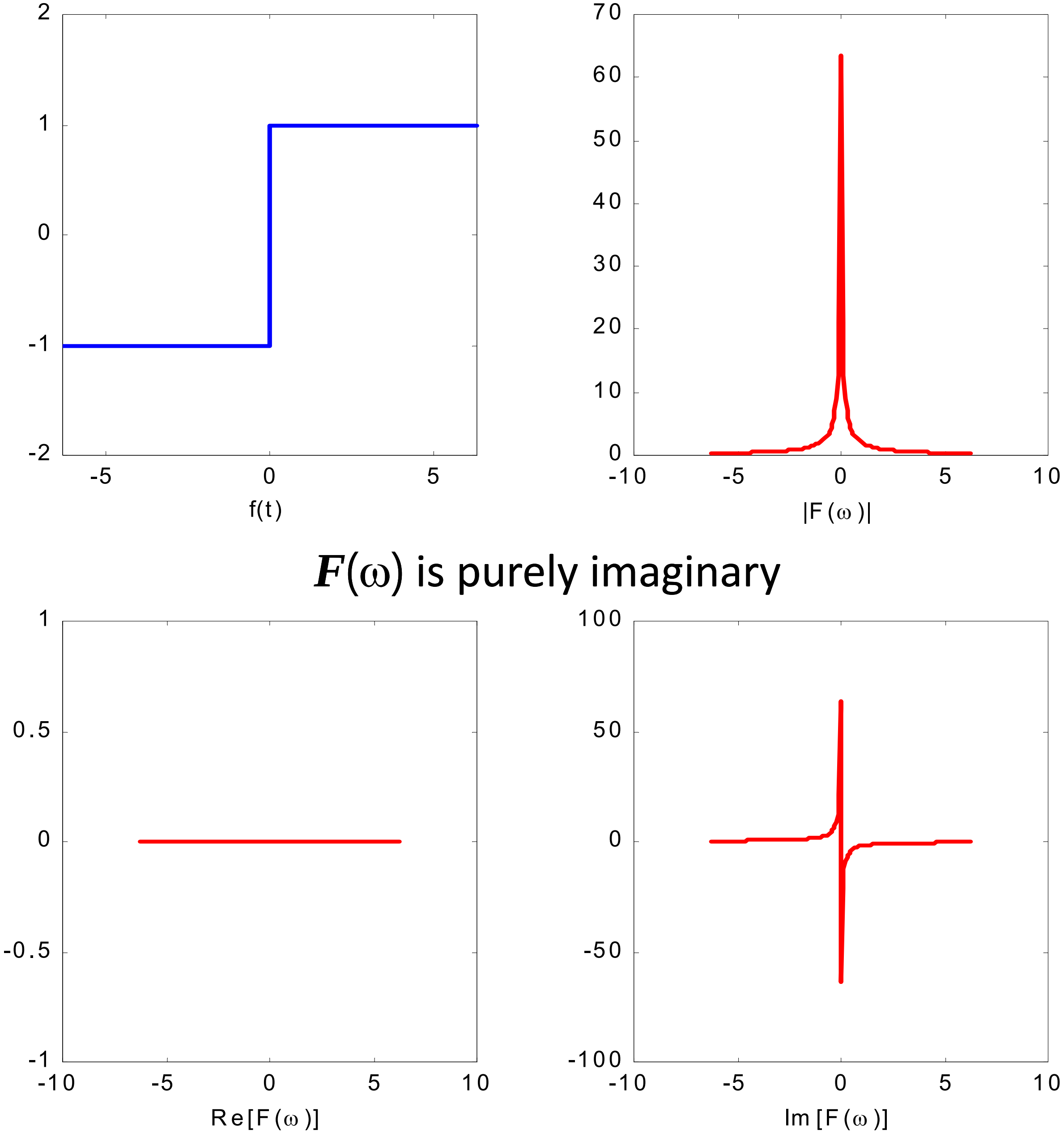
$$f(t) = \begin{cases} +1 & t > 0 \\ -1 & t < 0 \end{cases}$$

```
syms t real
ft=sign(t);
Fw=fourier(ft)
Fw =
-2i/w
```

```
sym(2/i)
ans =
-2i
```

$$F(\omega) = \frac{2}{i\omega}$$

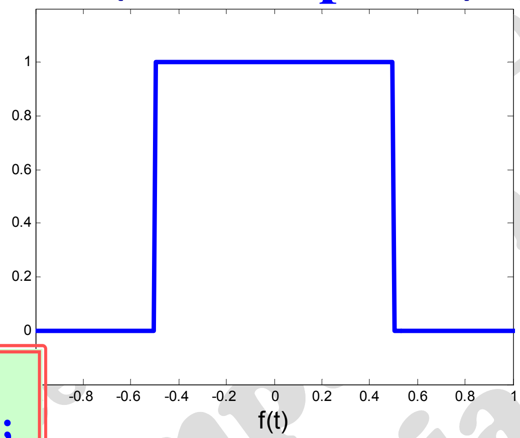
$$F(\nu) = \frac{1}{i\pi\nu}$$

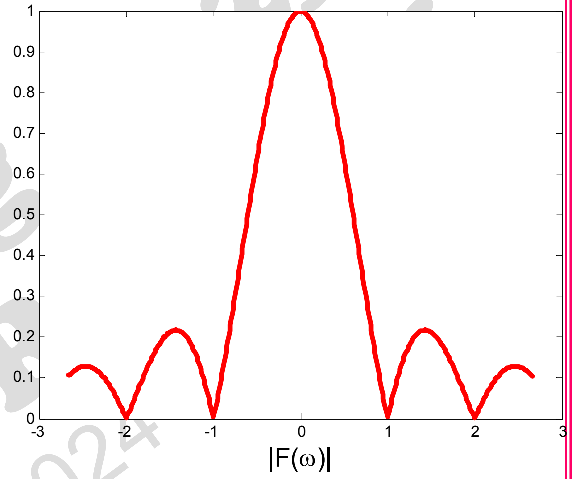


# Example of Fourier Transform (even function)

$\boldsymbol{f}(\boldsymbol{t})$ = **rectangular function** (or **rect pulse**)

$$f(t) = \begin{cases} 1 & |t| < \frac{L}{2} \\ 0 & |t| > \frac{L}{2} \end{cases} \qquad L = 1$$




MATLAB Symbolic Math Toolbox

```
syms L positive; syms t real
ft=rectangularPulse(-L/2,+L/2,t);
Fw=simplify(fourier(ft))
Fw =
(2*sin((L*w)/2))/w
```

```
syms L positive; syms t real; ft=heaviside(t+L/2)-heaviside(t-L/2);
```

**what is heaviside(x)?**

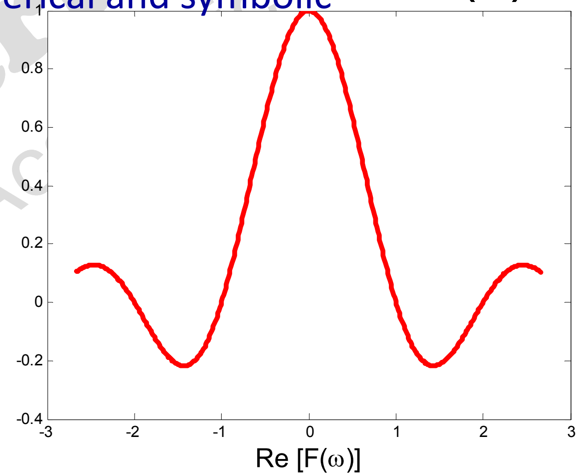
`rectpuls(t)` in MATLAB Signal Toolbox (only numerical)

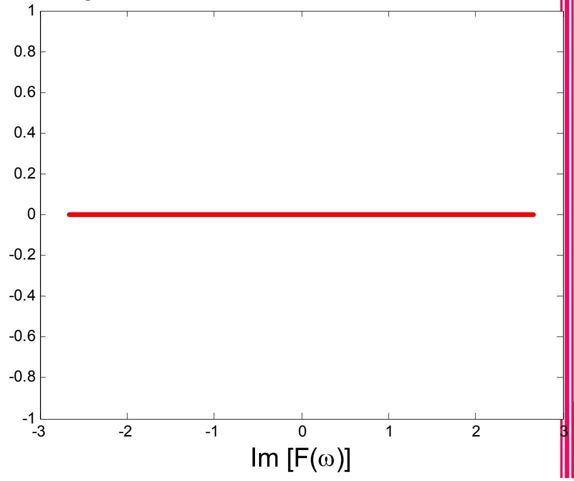
`sinc(v)` sine cardinal or sinc function
`sinc()` is both numerical and symbolic

$F(\omega)$ is purely real

$$F(\omega) = \frac{2\sin\frac{\omega}{2}}{\omega}$$

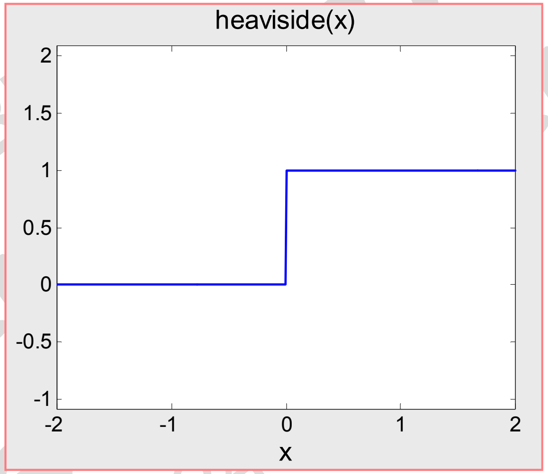
$$F(\nu) = \frac{\sin \pi\nu}{\pi\nu}$$



# what is heaviside(x)?

```
syms x real
ezplot(heaviside(x), [-2 2])
```




Heaviside function
or
Unit step function

$$H(x) = \begin{cases} 1 & x > 0 \\ 0 & x < 0 \end{cases}$$

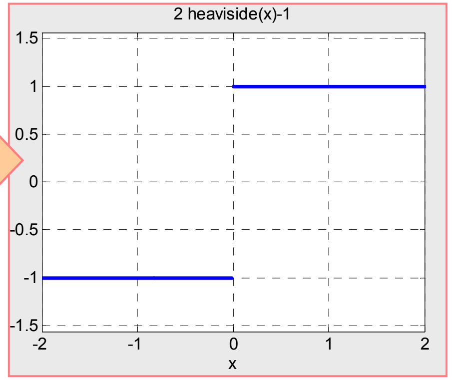
The Dirac delta $\delta$ is the derivative of the heaviside function

```
syms x real
diff(heaviside(x))
ans =
dirac(x)
```

# heaviside: what is it for?

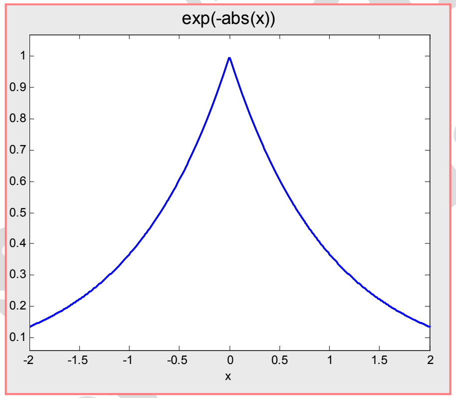
before MATLAB R2014a

```
syms x real
signum=2*heaviside(x)-1;
ezplot(signum, [-2 2])
```

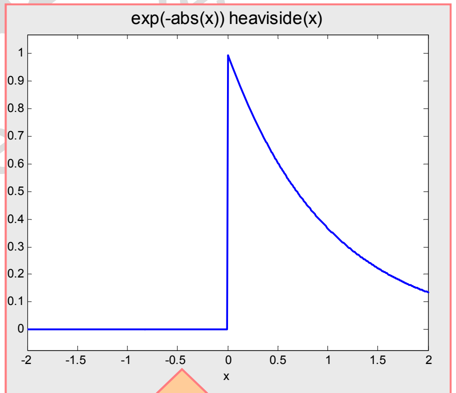



**sign()** function

decay function




decay pulse




```
syms x real; decayP=exp(-abs(x))*heaviside(x); ezplot(decayP, [-2 2])
```

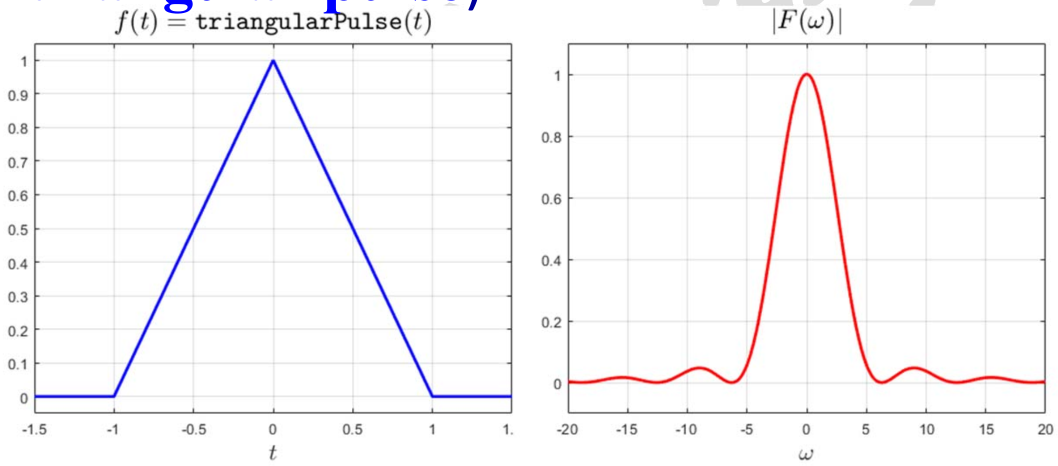
# Example of Fourier Transform (even function)

$f(t)$ = **triangle function** (or **triangular pulse**)

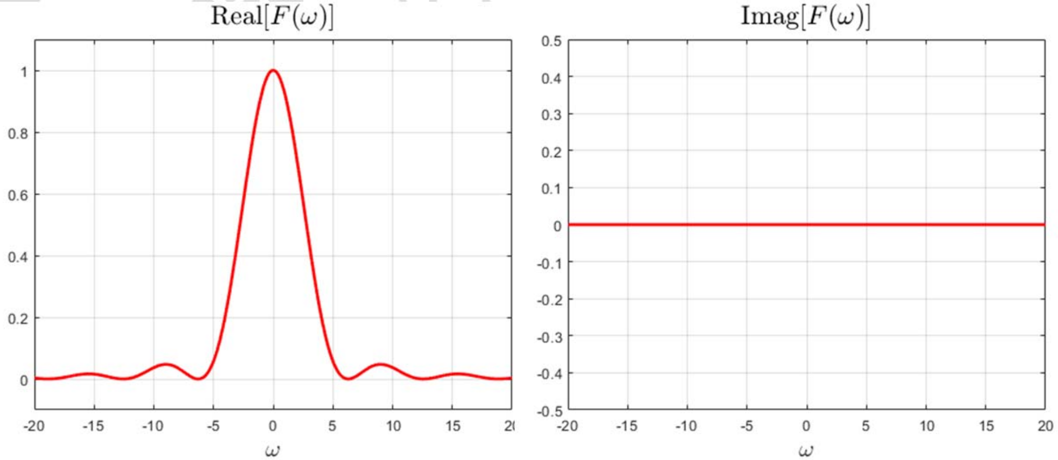
$$f(t)=\begin{cases}1+t & t\in[-L,0] \\ 1-t & t\in[0,L] \\ 0 & |t|>L\end{cases} \qquad L=1$$

MATLAB Symbolic Math Toolbox

```
syms t real
ft=triangularPulse(t);
Fw=simplify(fourier(ft),100)
Fw =
-(2*(cos(w) - 1))/w^2
syms v w
Fv=simplify(subs(Fw,w,2*pi*v),100)
Fv =
sin(pi*v)^2/(v^2*pi^2)
```




$F(\omega)$ is purely real




$$F(\omega)=2\frac{1-\cos\omega}{\omega^2}$$

$$F(\nu)=\left(\frac{\sin\pi\nu}{\pi\nu}\right)^2$$

`sinc(v)²`

sine cardinal or sinc function

**MATLAB** `sinc()` is both numerical and symbolic





(prof. M. Rizzardi)

# Example of Fourier Transform (even function)

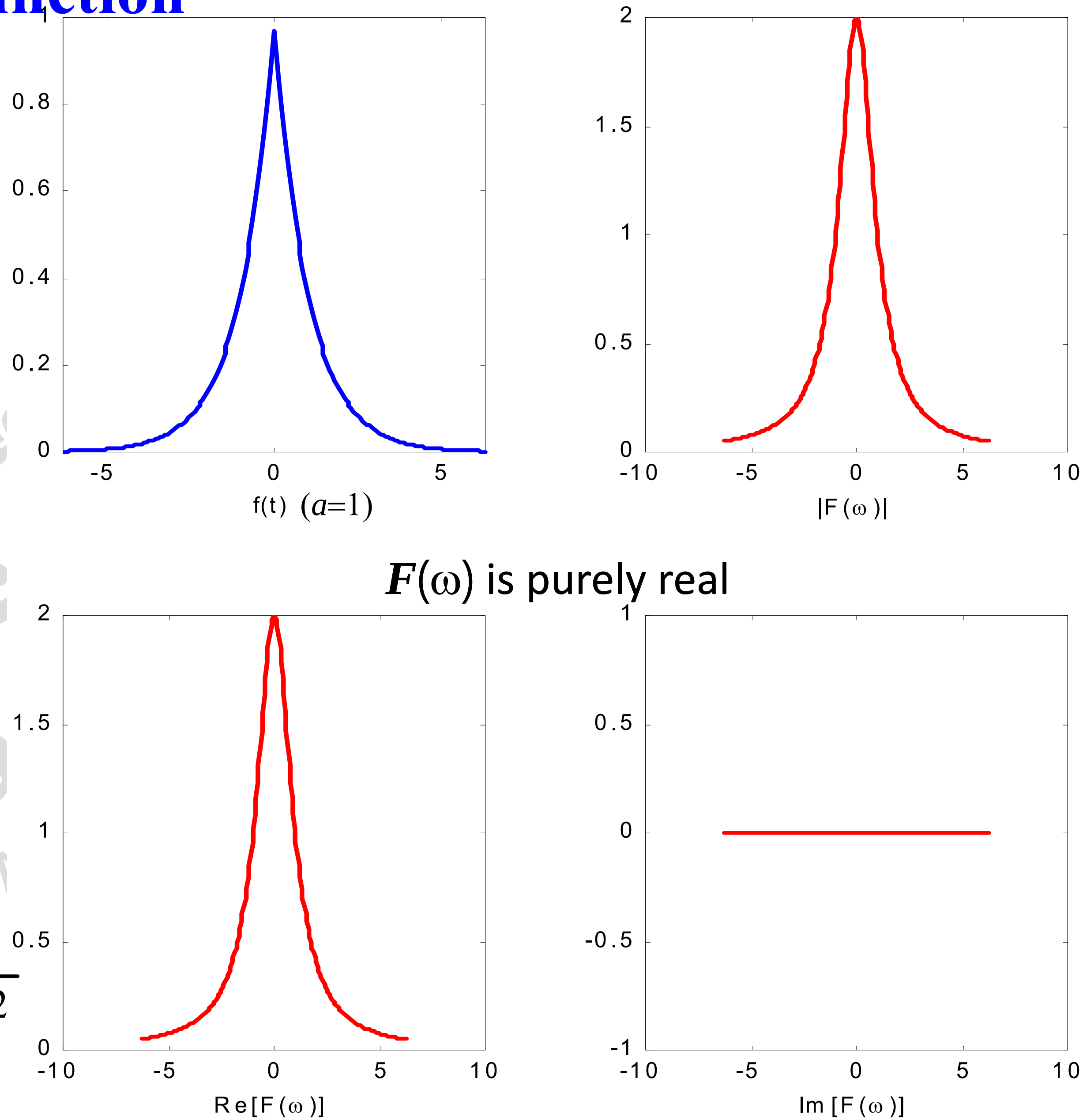
$f(t)$ = **even decay function**

$$f(t) = e^{-a|t|}, \quad a > 0$$

```
syms a t real
syms a positive
ft=exp(-a*abs(t));
Fw=fourier(ft)
Fw =
(2*a)/(a^2 + w^2)
```

$$F(\omega) = \frac{2a}{a^2 + \omega^2}$$

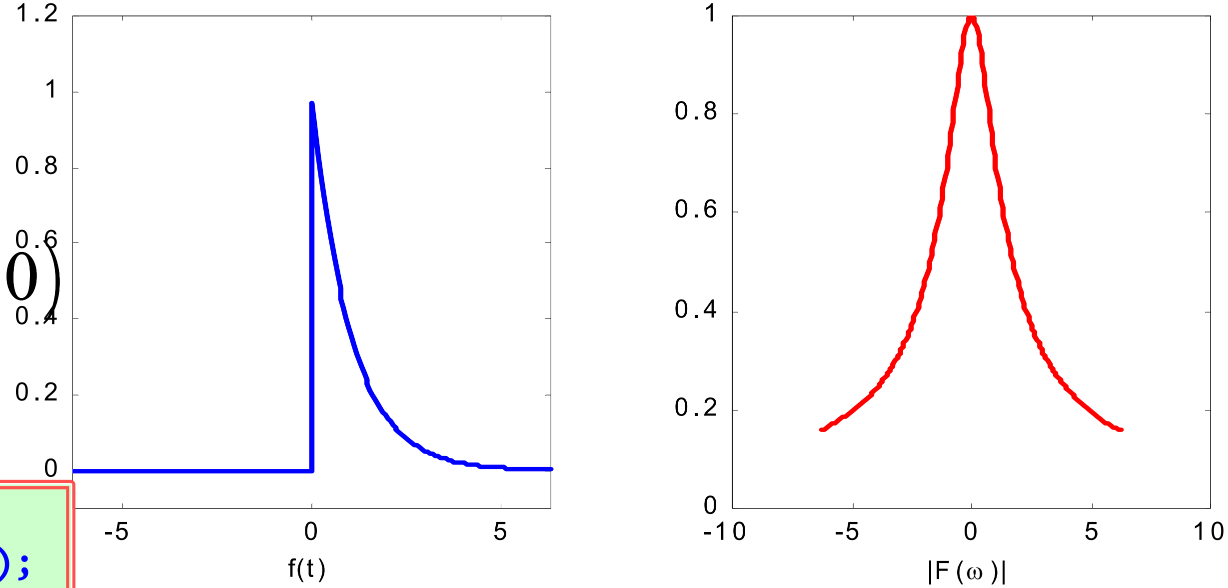
$$F(\nu) = \frac{2a}{a^2 + (2\pi\nu)^2}$$



# Example of Fourier Transform (neither even nor odd function)

***f*(*t*) = decay pulse**

$$f(t)=\begin{cases}e^{-at} & t>0\\ 0 & t<0\end{cases}\quad(a>0)$$

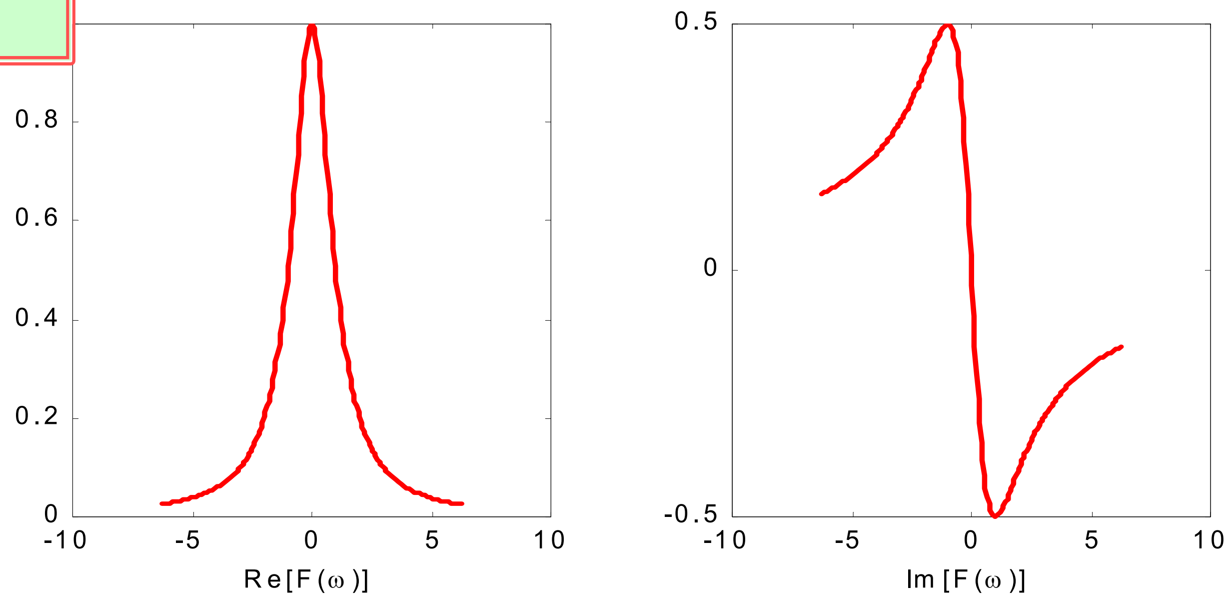



```
syms t real; syms a positive
ft=exp(-a*abs(t))*heaviside(t);
Fw=simplify(fourier(ft))
Fw =
1/(a + w*1i)
```

***F*(ω) is complex**

$$F(\omega)=\frac{1}{a+i\omega}$$

$$F(\nu)=\frac{1}{a+i2\pi\nu}$$

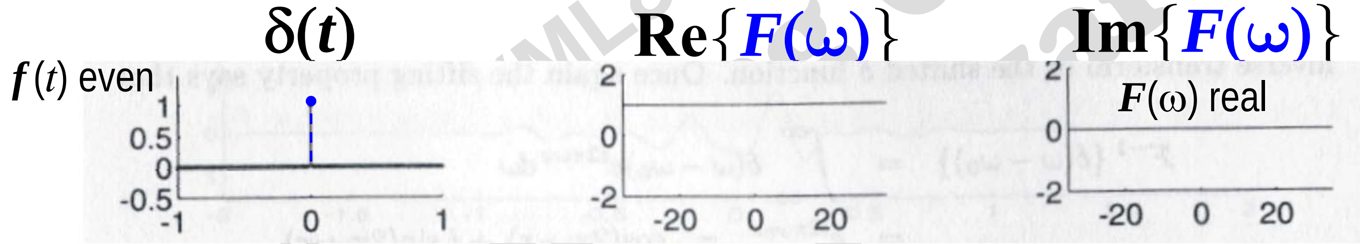


# Examples of Fourier Transform

## Dirac delta function

$$f(t)=\delta(t) \circ\!\!-\!\!\bullet F(\omega)=1$$

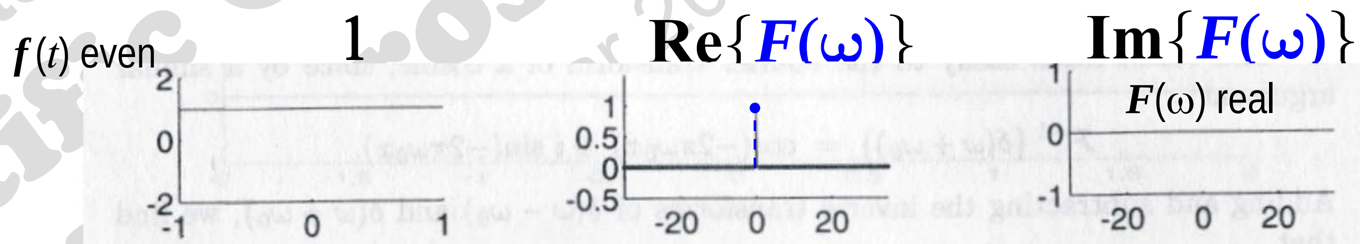
```
syms t; f=dirac(t);
F = fourier(f)
 F =
 1
```




## constant function 1

$$f(t)=1 \circ\!\!-\!\!\bullet F(\omega)=2\pi\delta(\omega)$$

```
fourier(sym(1))
ans =
2*pi*dirac(w)
```




## Heaviside function

$$f(t)=\begin{cases}1 & t>0\\ 0 & t<0\end{cases} \qquad F(\omega)=\pi\,\delta(\omega)-\frac{i}{\omega}$$

```
syms t real; fourier(heaviside(t))
ans =
pi*dirac(w) - 1i/w
```

# Examples of Fourier Transform

## trigonometric functions

f(t) even

$$f(t) = \cos \alpha t \quad \circ\!\!-\!\!\bullet \quad F(\omega) = \pi[\delta(\omega+\alpha)+\delta(\omega-\alpha)]$$

F(ω) is real

f(t) odd

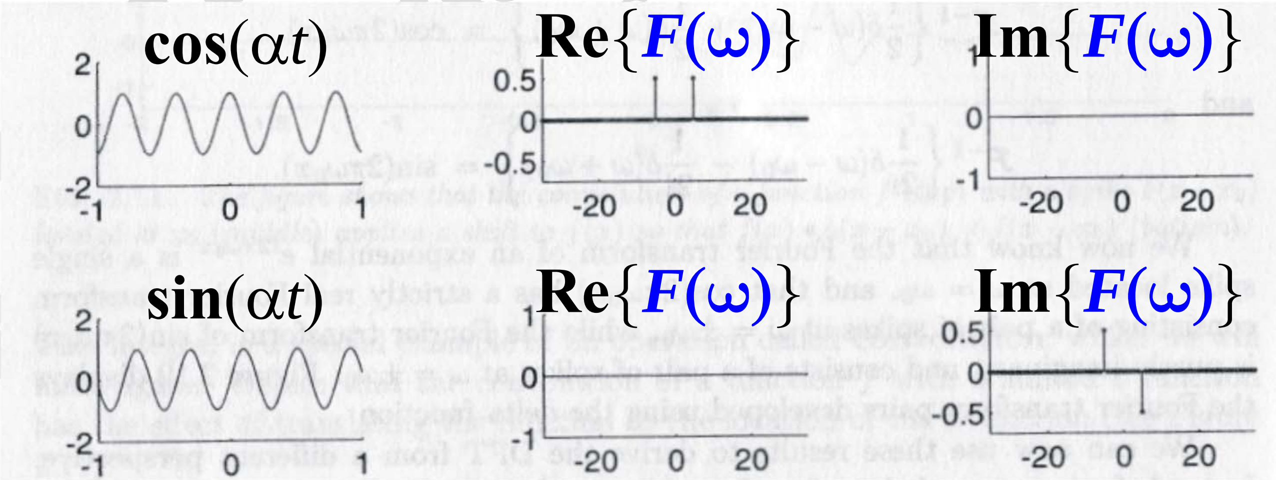
$$f(t) = \sin \alpha t \quad \circ\!\!-\!\!\bullet \quad F(\omega) = i\pi[\delta(\omega+\alpha)-\delta(\omega-\alpha)]$$

F(ω) is imaginary

$$f(t) = e^{i\alpha t} \quad \circ\!\!-\!\!\bullet \quad F(\omega) = 2\pi\delta(\omega-\alpha)$$

```
syms t a real
disp(fourier(cos(a*t)))
pi*(dirac(w-a)+dirac(w+a))
disp(fourier(sin(a*t)))
pi*(-dirac(w-a)+dirac(w+a))*1i
```

```
syms t a real
disp(fourier(exp(i*a*t)))
2*pi*dirac(w-a)
```



# Examples of Fourier Transform

## Comb function (pulse train or sampling function)

$$f(t) = \delta_T(t) = \sum_{k=-\infty}^{+\infty} \delta(t - kT) \quad \circ\!\!-\!\!\bullet \quad F(\omega) = \frac{2\pi}{T} \sum_{k=-\infty}^{+\infty} \delta\left(\omega - k\frac{2\pi}{T}\right) = \frac{1}{2\pi} \sum_{k=-\infty}^{+\infty} e^{i\omega kT}$$

$$F(\nu) = \frac{1}{T} \sum_{k=-\infty}^{+\infty} \delta\left(\nu - \frac{k}{T}\right) = \sum_{k=-\infty}^{+\infty} e^{i2\pi\nu kT}$$

pulse train

$$\delta_T(t) \quad \circ\!\!-\!\!\bullet \quad \frac{1}{T}\delta_{\frac{1}{T}}(\nu)$$

period $T$ — period $1/T$

The Fourier Transform of a comb function is still a comb function, but with a period equal to the reciprocal of the period of the original function.

$\delta_T$ is a series of $T$-shifted "δ functions", called as **superposition** or **periodic replication** of $\delta(t)$; $\delta_T$ is characterized by

$$\langle \delta_T, g \rangle = \int_{-\infty}^{+\infty} g(x)\delta_T(x)dx = \sum_{k=-\infty}^{+\infty} g(kT)$$

(it describes the sampling of $g$)

by the Sifting Property of the Dirac delta

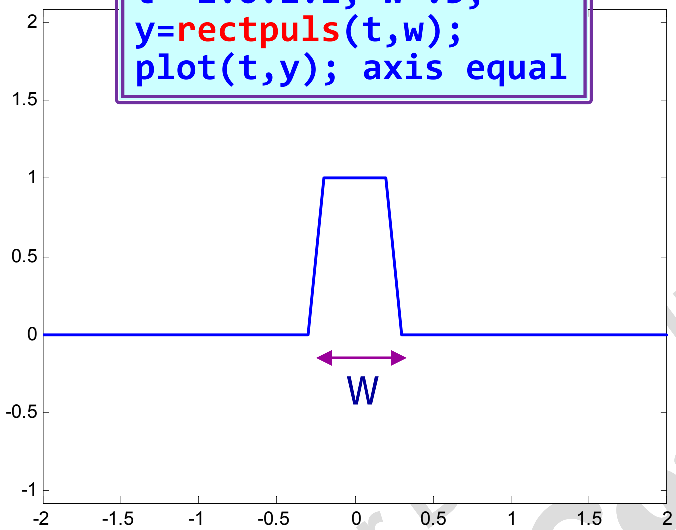
$$\int_{-\infty}^{\infty} f(x)\delta(x - x_0)\mathrm{d}x = f(x_0)$$

in MATLAB Signal Toolbox `pulstran()`

# pulstran(): in MATLAB Signal Toolbox

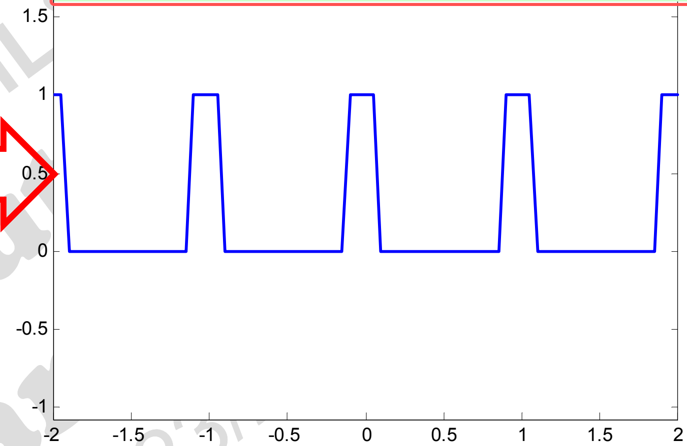
**window functions**

```
t=-2:0.1:2; w=.5;
y=rectpuls(t,w);
plot(t,y); axis equal
```

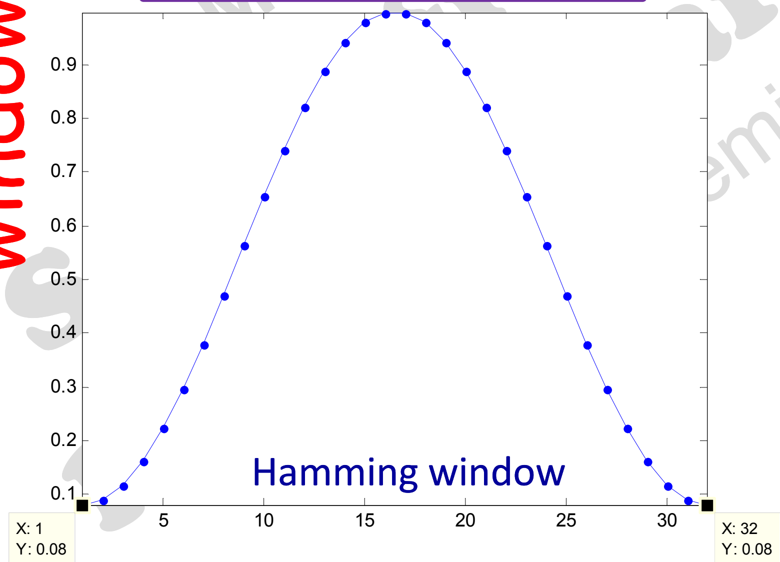



**superposition**

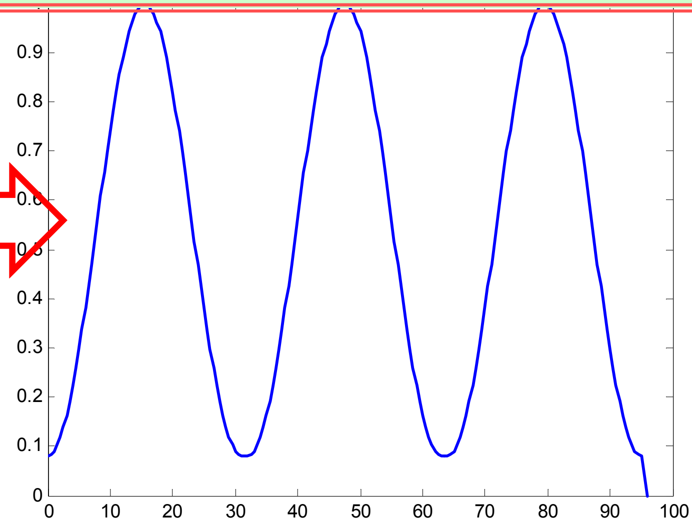
```
t=-2:0.05:2; w=0.2;  d=-2:2;
y=pulstran(t,d,'rectpuls',w);
plot(t,y); axis equal; grid on
```



```
y=hamming(32); plot(y)
```




```
W=32; N=3; d=(0:N-1)'*W; t=0:0.5:N*W;
p=hamming(H); y=pulstran(t,d,p);
plot(t,y)
```



# What is a **superposition** $\varphi_T(x) = \sum_{k=-\infty}^{+\infty} \varphi(x-kT)$ ?

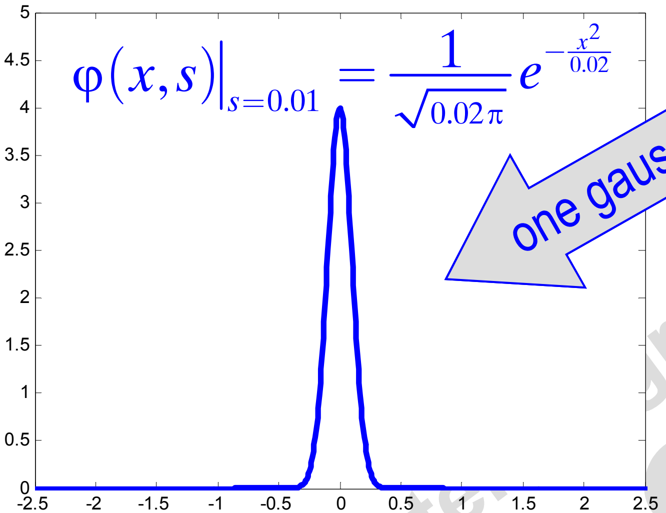
periodic replication

$$p(x,\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

for $\mu=0$ and $s=\sigma^2$ it becomes $\varphi(x,s) = \frac{1}{\sqrt{2\pi s}} e^{-\frac{x^2}{2s}}$

one gaussian

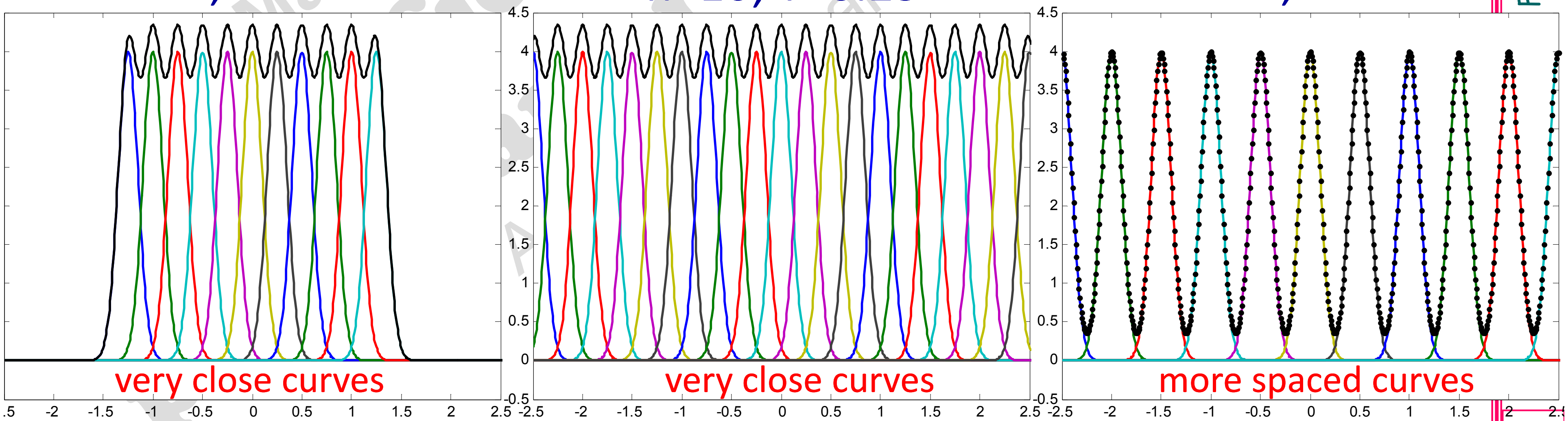
$$\varphi(x,s)\big|_{s=0.01} = \frac{1}{\sqrt{0.02\pi}} e^{-\frac{x^2}{0.02}}$$



We give the idea of a superposition, but in a finite case

```
f=@(x,k,t)exp(-(x-k*t).^2/0.02)/sqrt(pi*0.02);
n=5;   k=-n:n;  x=linspace(-4,4,1001);
[K,X]=meshgrid(k,x); T=0.25;
F=f(X,K,T); plot(x',F, x',(sum(F,2)),'k')
axis([-2.5 2.5 -.5 4.5])
```

n=5; T=0.25 | n=10; T=0.25 | n=5; T=0.5

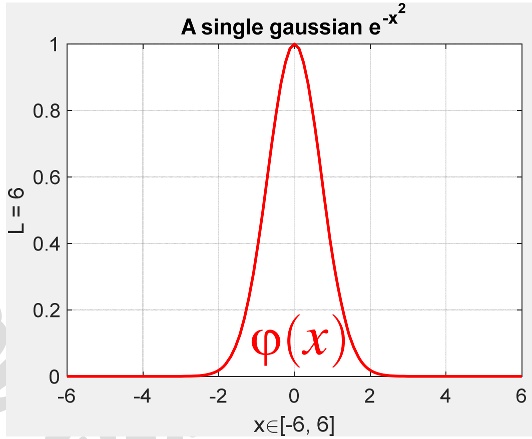



The **black curve** represents the (finite) superposition of n gaussians with a T period

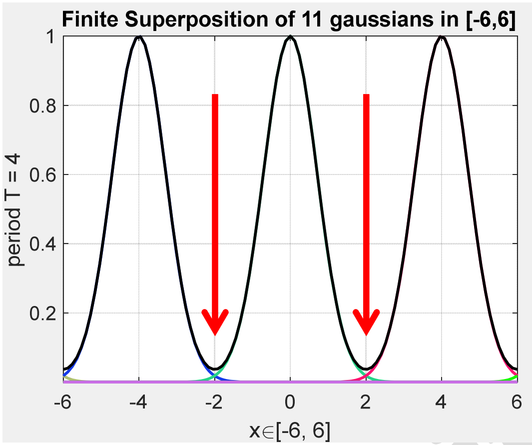
Download: `finite_superposition.p`

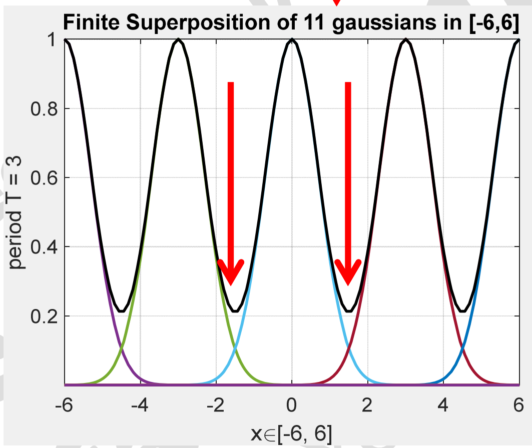
$$\varphi_T(x) = \sum_{k=-\infty}^{+\infty} \varphi(x - kT)$$

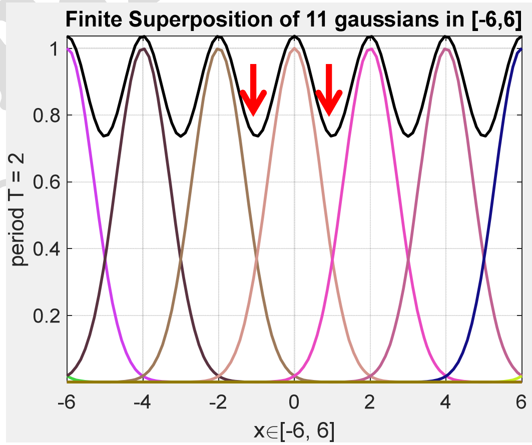
(infinite) superposition



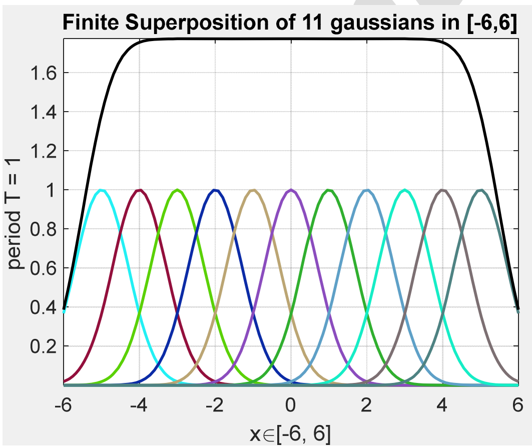

Nyquist frequency



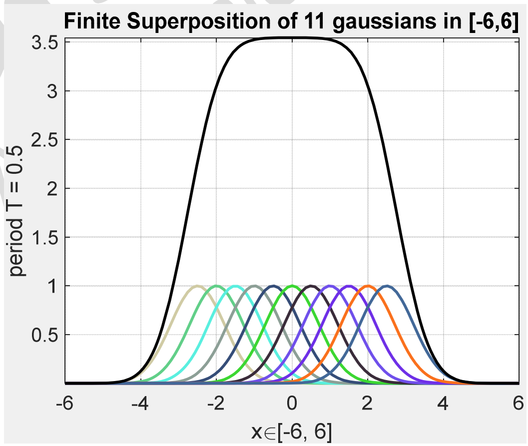



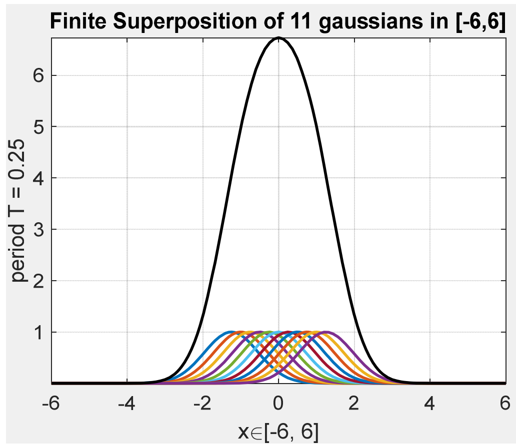

Nyquist frequency? Lost!




Nyquist frequency? Lost!



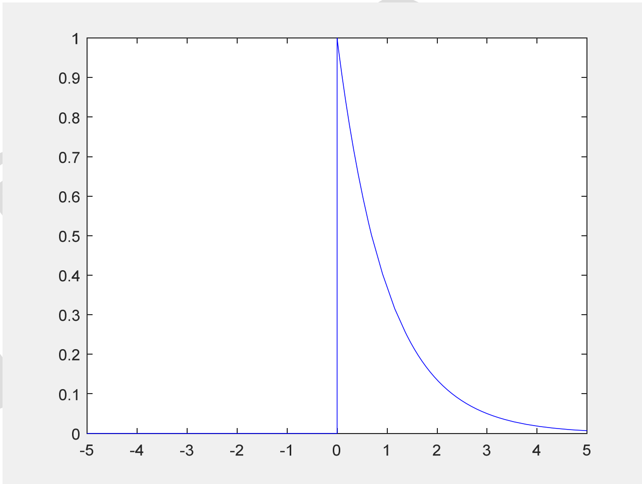

Nyquist frequency? Lost!



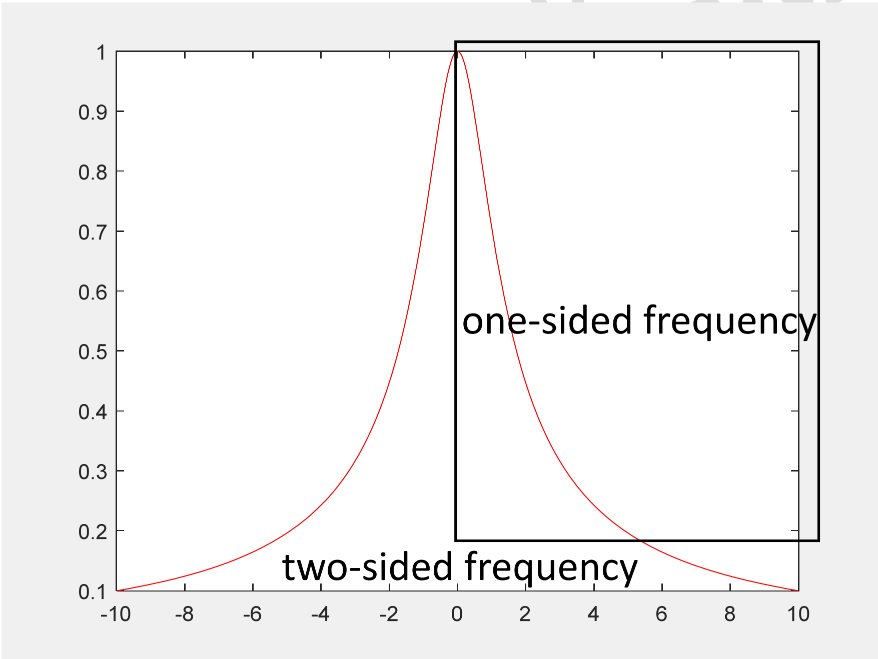
# a bit of terminology …

```
syms t real
ft=exp(-abs(t))*heaviside(t);
figure; fplot(ft,[-5 5]); axis tight
Fw=simplify(fourier(ft));
figure; fplot(abs(Fw), [-10 10])
figure; fplot(abs(Fw)^2, [-10 10])
```

signal: decay pulse $e^{-|t|}$

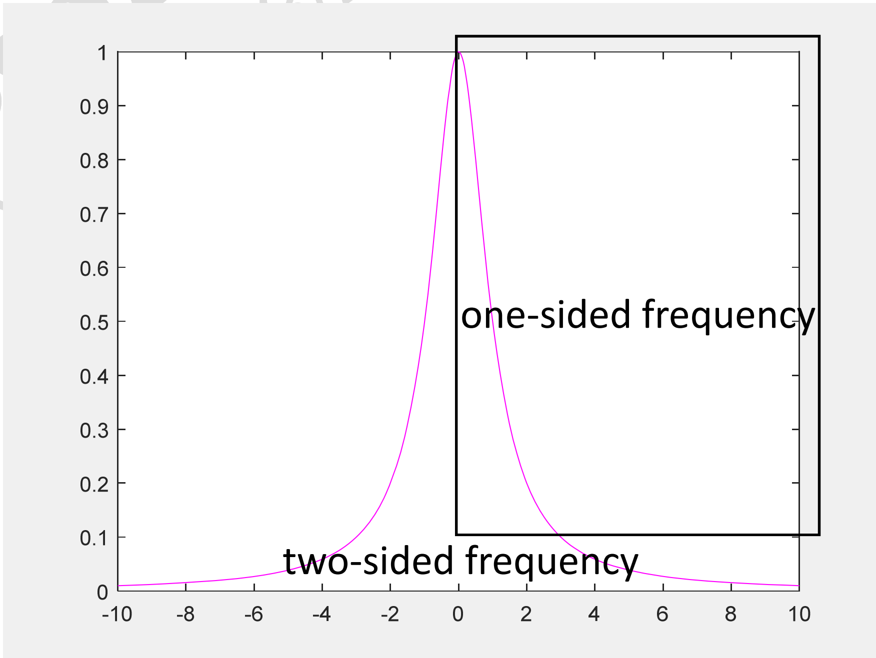



Fourier Spectrum: $\left|F(\omega)\right|$




frequency ω

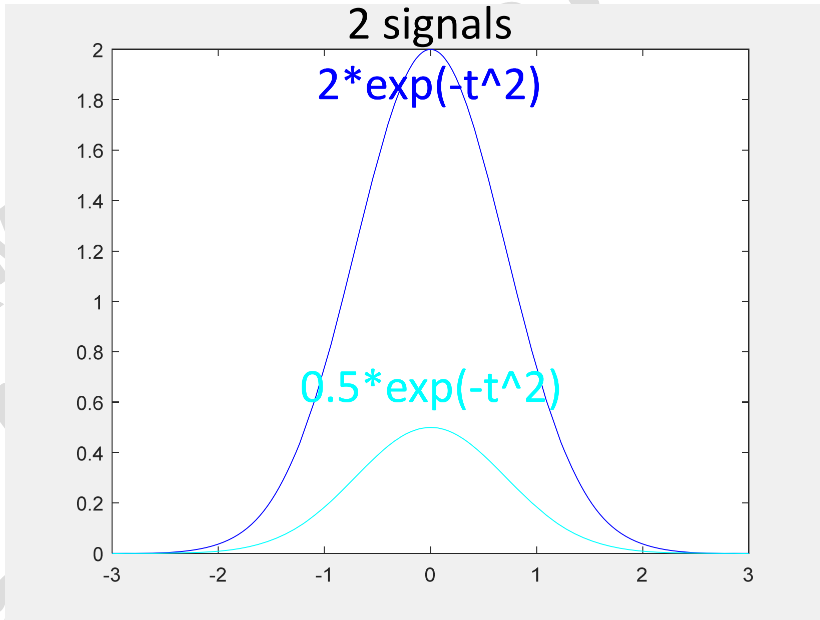
Power Spectrum: $\left|F(\omega)\right|^2$

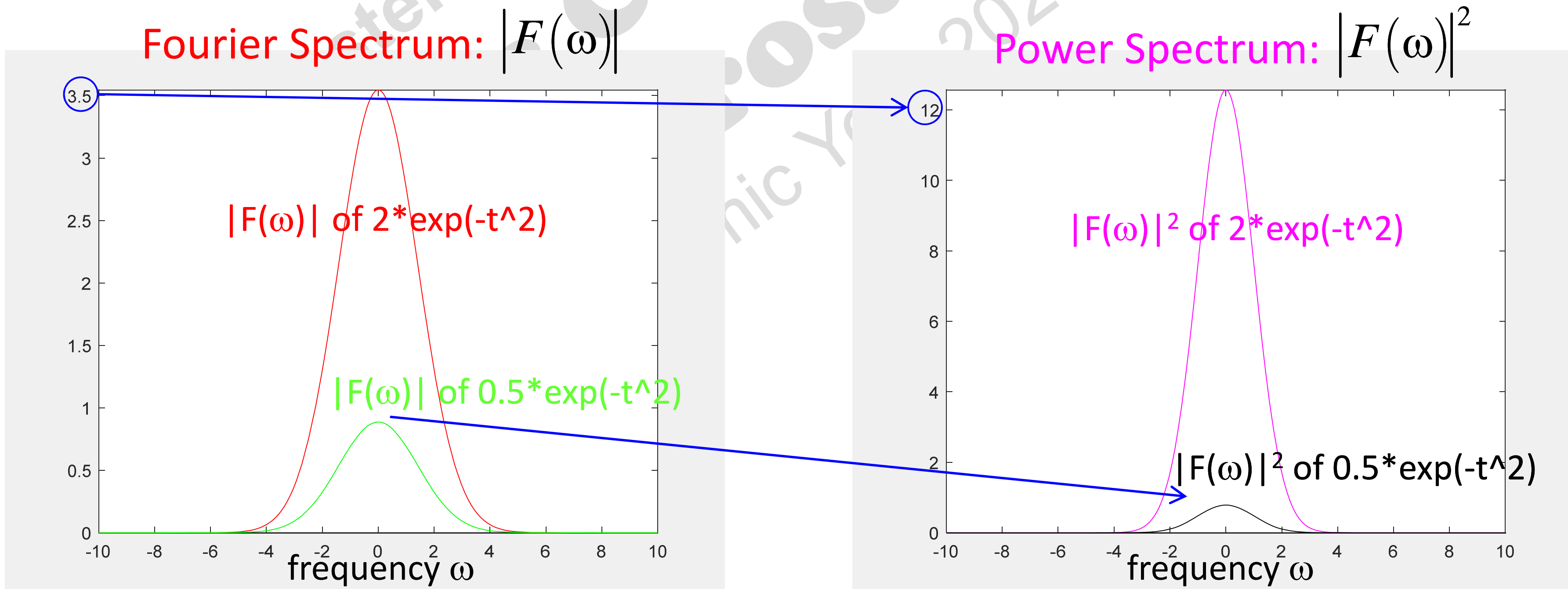



frequency ω

# What is the Power Spectrum $|F(\omega)|^2$ for?

The Power Spectrum amplifies the large values of the Fourier Spectrum ($|F(\omega)|>1$) and shrinks its small values ($|F(\omega)|<1$).



# Main properties of the Fourier Transform [1]



Let $\mathscr{F}$ be the map which transforms a function $f(x) \in L^1(-\infty, +\infty)$ into its Fourier Transform $F(\omega)$

$$\mathscr{F} : f \longrightarrow \mathscr{F}[f,\omega] = F(\omega) \qquad \boxed{f(t) \bullet\!\!-\!\!-\!\!\circ\, F(\omega)}$$

- $\mathscr{F}[f,\omega]$ is a ***linear operator***: $\mathscr{F}[\alpha f + \beta g, \omega] = \alpha F(\omega) + \beta G(\omega)$.
- If $f$ is an ***even function***, then $F(\omega)$ is ***real*** and it is in turn an ***even function***.
- If $f$ is an ***odd function***, then $F(\omega)$ is ***purely imaginary*** and it is in turn an ***odd function***.
- If $f$ is a ***real-valued function***, then $F(\omega)$ is a ***complex valued function***, and

$$F(-\omega) = \overline{F(\omega)}$$

- ***Shift Properties***: *shifting* (or *translating*) a function *in the time domain* $t \pm h$ corresponds to a *rotation* by an angle $\pm h\omega$ *in the frequency domain*, i.e.

$$f(t \pm h) \bullet\!\!-\!\!-\!\!\circ\, e^{\pm ih\omega} F(\omega) \qquad \boxed{\text{Time shift property}}$$

and *shifting* (or *translating*) *in the frequency domain* $\omega \mp \lambda$ corresponds to a *rotation* by an angle $\pm \lambda t$ *in the time domain*, i.e.

$$e^{\pm i\lambda t} f(t) \bullet\!\!-\!\!-\!\!\circ\, F(\omega \mp \lambda) \qquad \boxed{\text{Frequency shift property}}$$

# Main properties of the Fourier Transform [2]



- ***Time scaling*** (or ***Similarity Property***): if $f(t) \bullet\!\!-\!\!\!-\!\!\!-\!\!\circ \mathscr{F}[f,\omega] = F(\omega)$ and $c \in \mathbb{R} - \{0\}$, then

$$f(ct) \bullet\!\!-\!\!\!-\!\!\!-\!\!\circ F(\omega/c)/|c|$$

- ***Convolution Property***: if $f, g \in L^1(-\infty, +\infty)$ also $f * g \in L^1$, then *the convolution of signals* in the time domain will be transformed into *the multiplication of their Fourier Transforms* in the frequency domain, and, conversely, *the multiplication of signals* in the time domain will be transformed into *the convolution of their Fourier Transforms* in the frequency domain:

$$\mathscr{F}[f * g, \omega] = \frac{1}{2\pi} F(\omega) \cdot G(\omega) \quad \text{and} \quad \mathscr{F}[f \cdot g, \omega] = F(\omega) * G(\omega)$$

$$\mathscr{F}[f * g, \nu] = F(\nu) \cdot G(\nu) \quad \text{and} \quad \mathscr{F}[f \cdot g, \nu] = F(\nu) * G(\nu)$$

where the ***convolution*** between $f, g \in L^1(-\infty, +\infty)$ is defined as

$$[f * g](\tau) = \int_{-\infty}^{+\infty} f(t) g(\tau - t) \, \mathrm{d}t$$

- ***Parseval's Identity*** (or ***Rayleigh's Energy Theorem***)

$$\int_{-\infty}^{+\infty} |f(t)|^2 \, \mathrm{d}t = \frac{1}{2\pi} \int_{-\infty}^{+\infty} |F(\omega)|^2 \, \mathrm{d}\omega$$

# Main properties of the Fourier Transform [3]

- ***Differentiation of f(t)***: If $f$ is absolutely continuous and summable, and $f'$ is summable, then

$$\mathscr{F}[f',\omega] = i\omega\,\mathscr{F}[f,\omega]$$

More generally, if $f$ is absolutely continuous and summable with its first $k$–1 derivatives, and $f^{(k)}$ is summable, then

$$\mathscr{F}[f^{(k)},\omega] = (i\omega)^k\,\mathscr{F}[f,\omega]$$

- ***Differentiation of F(ω)***: If $f(t)$, $tf(t) \in L^1(-\infty,+\infty)$, then $F$ has a continuous derivative, and

$$F'(\omega) = \mathscr{F}[-itf(t),\omega]$$

More generally, if $f(t)$, $tf(t)$, …, $t^k f(t) \in L^1(-\infty,+\infty)$, then $F$ has continuous derivatives up to order $k$, and

$$F^{(k)}(\omega) = \mathscr{F}[-(it)^k f(t),\omega]$$

- ***Symmetry*** (or ***Duality***) ***Property***:

$$\begin{cases} \mathscr{F}[F(\omega),y] = \int_{-\infty}^{+\infty} F(\omega)e^{-i\omega y}\,\mathrm{d}\omega = 2\pi f(-y) \\ \mathscr{F}[F(\nu),y] = \int_{-\infty}^{+\infty} F(\nu)e^{-2\pi i\nu y}\,\mathrm{d}\nu = f(-y) \end{cases}$$

reversed signal

- ***Riemann-Lebesgue Lemma***: If $f(t) \in L^1(-\infty,+\infty)$, then $F(\omega)$ is a continuous function and it is infinitesimal at ∞.

## Example of the Time shift property (w.r.t. the angular frequency)

$$f(t-h) \bullet\!\!-\!\!\circ\; e^{-ih\omega} F(\omega)$$

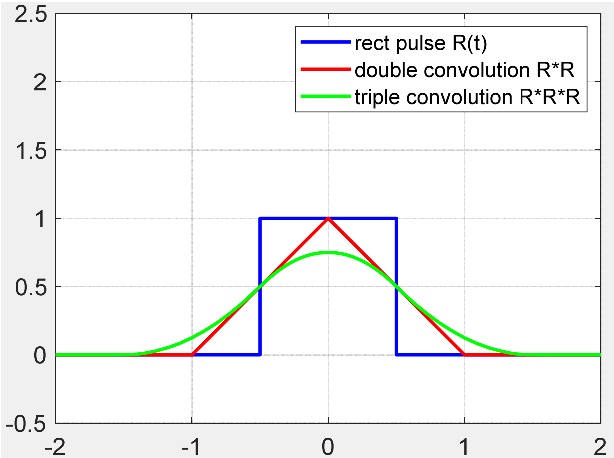
```
syms t real
syms a positive
ft=exp(-a*abs(t));
Fw=fourier(ft)
Fw =
(2*a)/(a^2 + w^2)
syms h real
f1t=exp(-a*abs(t-h));
F1w=simplify(fourier(f1t))
F1w =
(2*a*exp(-h*w*1i))/(a^2 + w^2)
F1w/Fw
ans =
exp(-h*w*1i)
```

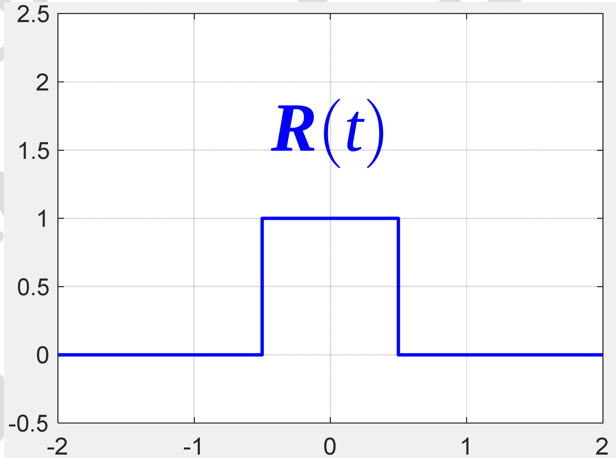
# Application of Convolution Property (w.r.t. the circular freq)

$$[f*g](\tau)=\int_{-\infty}^{+\infty} f(t)g(\tau-t)\mathrm{d}t \qquad \boldsymbol{f*g} = \mathscr{F}^{-1}\{\mathscr{F}[\boldsymbol{f}]\cdot\mathscr{F}[\boldsymbol{g}]\}$$

rect pulse

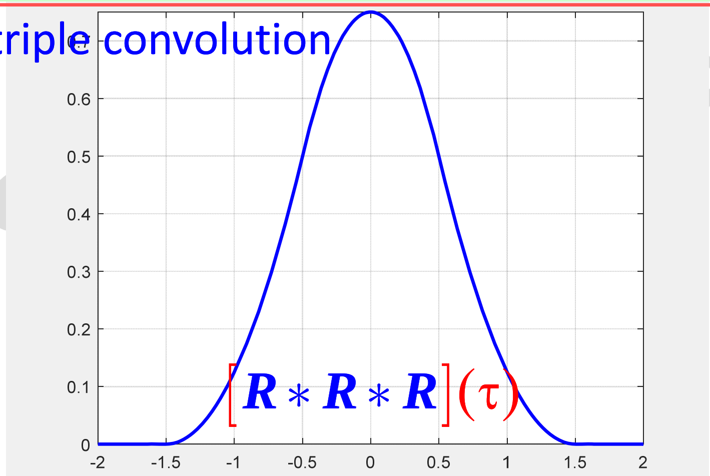
$$\boldsymbol{R}(t)=\begin{cases}1 & |t|<\dfrac{1}{2}\\ 0 & |t|>\dfrac{1}{2}\end{cases}$$



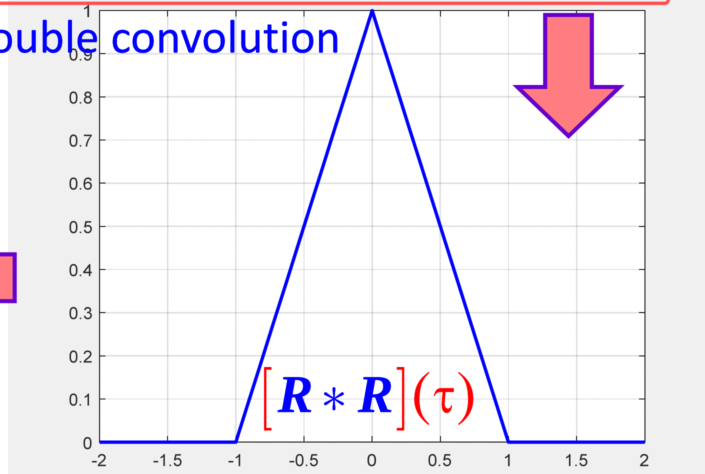




```
syms t w real;  Rt=rectangularPulse(t);  Fw=simplify(fourier(Rt))
fplot(Rt,[-2 2],'Color','b','LineWidth',2); hold on
F2=Fw*Fw; R2=simplify(ifourier(F2),100); % double convolution R(t)*R(t)
fplot(R2,[-2 2],'Color','r','LineWidth',2)
F3=F2*Fw; R3=simplify(ifourier(F3),100); % triple convolution R(t)*R(t)*R(t)
fplot(R3,[-2 2],'Color','g','LineWidth',2)
```

triple convolution




double convolution

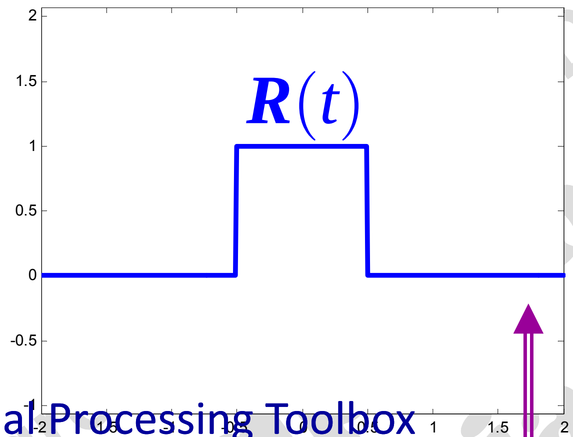


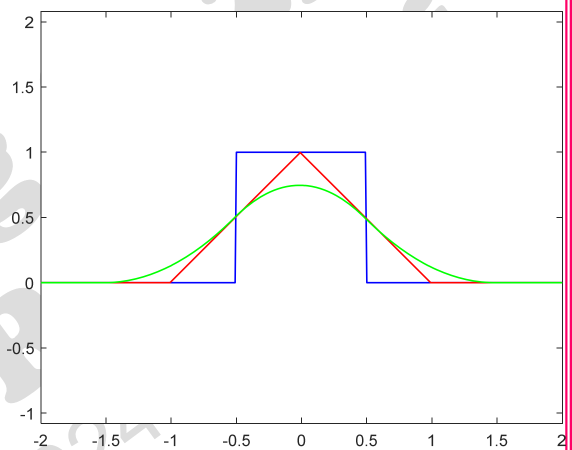
# Numerical application of Convolution Property (w.r.t. the circular freq)

Equivalently by conv() ... `RR=conv(R,R,'same')*T/N;`

rect pulse

$$R(t) = \begin{cases} 1 & |t| < \frac{1}{2} \\ 0 & |t| > \frac{1}{2} \end{cases}$$

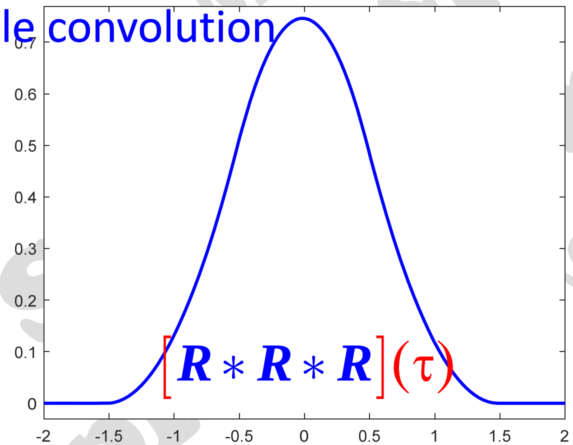


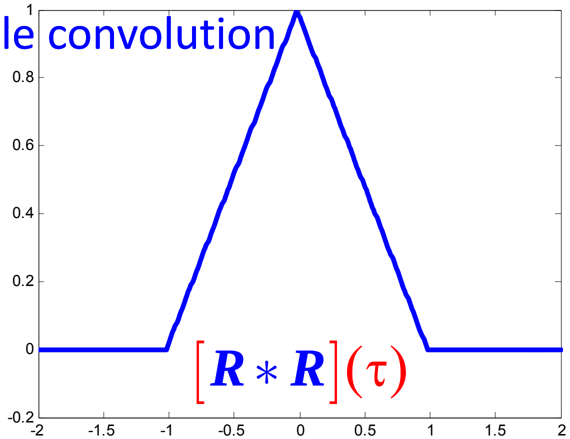
**rectpuls()**: in MATLAB Signal Processing Toolbox

```
a=-2; b=2; T=b-a; t=a:.01:b; N=numel(t)-1; R=rectpuls(t); plot(t,R,'b')
axis equal; hold on
RR=conv(R,R,'same')*T/N; plot(t,RR,'r')      % double convolution of R
RRR=conv(RR,R,'same')*T/N; plot(t,RRR,'g')  % triple convolution of R
```

triple convolution



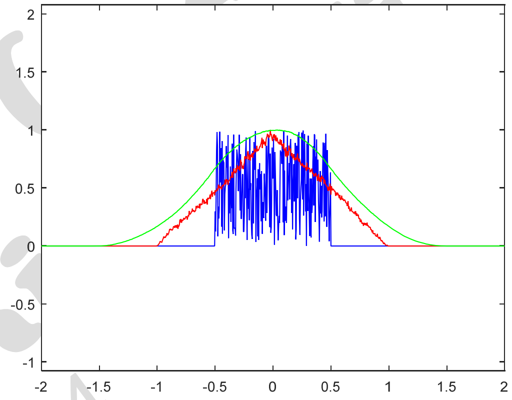

double convolution



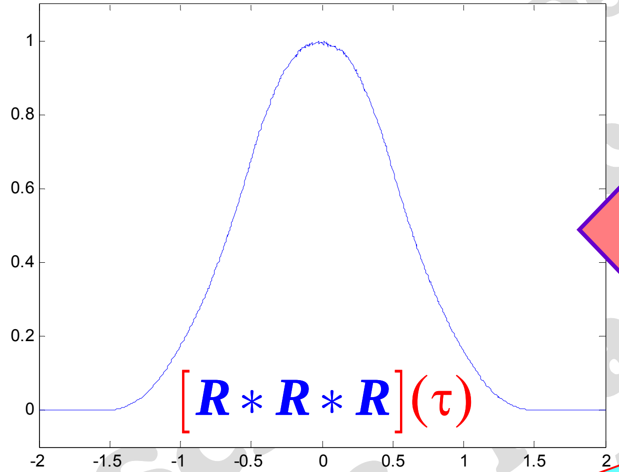

**Exercise**: repeat with **DFT** (**fft**) and with **cconv(R,R,N)**

# Numerical application of Convolution Property (w.r.t. the circular freq)

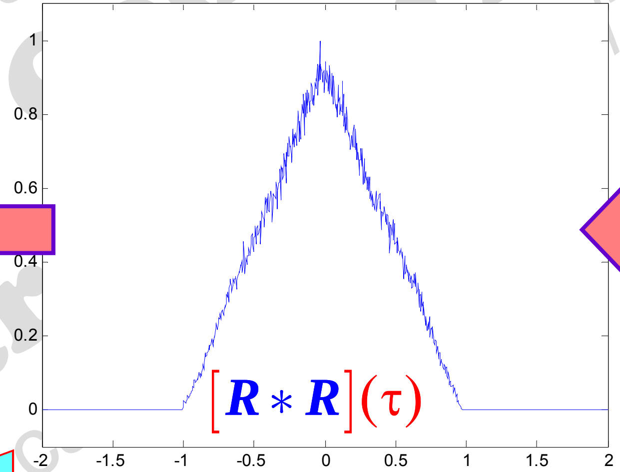
```
a=-2; b=2; T=b-a;
t=a:.01:b; N=numel(t)-1;
rng('default');
R=rectpuls(t).*rand(size(t)); % with uniform random noise
RR=conv(R,R,'same')*T/N;     % RR double convolution R(t)*R(t)
RRR=conv(RR,R,'same')*T/N;  % RRR triple convolution R(t)*R(t)*R(t)
plot(t,R,'b'); axis equal; hold on
plot(t,RR/max(RR),'r')
plot(t,RRR/max(RRR),'g')
```

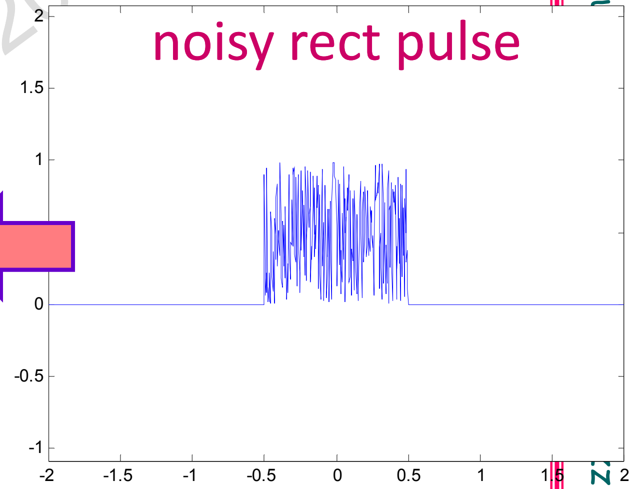


triple convolution




double convolution




the noise has been abated! (convolution as data filtering)

**Exercise**: repeat with **DFT** (**fft**) and with **cconv(R,R,N)**

From the previous example, we get the following:

From $\mathrm{sinc}(\omega)$ being the Fourier Transform of the rect pulse:

$$\mathscr{F}[\text{rect pulse},\omega] = \mathrm{sinc}(\omega)$$

it follows that the Fourier Transform of the triangular function is $\mathrm{sinc}^2(\omega)$:

$$\mathscr{F}[\text{triangular pulse},\omega] = \mathrm{sinc}^2(\omega)$$

**Quiz:** ... Why?
Explain your answer

# Contents

- Windowing and Aliasing errors.
- The Nyquist-Shannon Sampling Theor.
- Connection between FT and FC.
- Algorithm for the numerical approximation of the FT.
- Some applications of the Fourier Transform.
- Short Time Fourier Transform (STFT).
- Some applications to sounds of the Fourier Transform.

# Windowing effect

When performing the spectral analysis of a sample from a physical phenomenon $f(t)$, we are actually using an observation of the phenomenon obtained over a finite interval of time.

From a mathematical point of view, this process is equivalent to the multiplication of the function $f(t)$, which describes the phenomenon, by a rectangular function $w(t)$ (rectangular window function of width $L$); thus the actually "observed" function is:

$$g(t) = f(t) \cdot w(t)$$

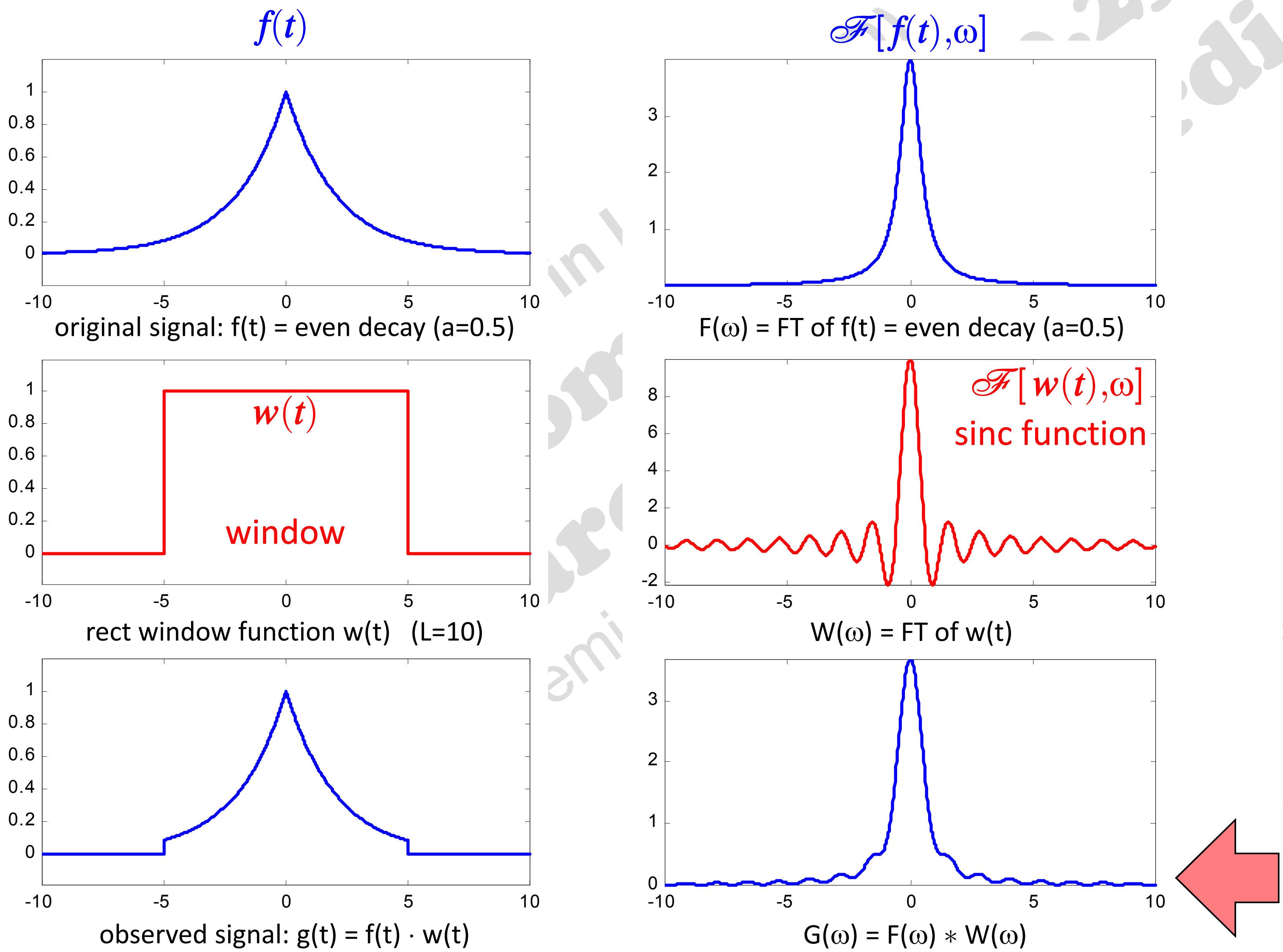
$$F(\nu) = \mathscr{F}[f(t),\nu]$$
$$W(\nu) = \mathscr{F}[w(t),\nu]$$

so that the ***Fourier Transform G(ν)*** of $g(t)$ is then given by:

$$G(\nu) = \mathscr{F}[g(t),\nu] = \mathscr{F}[f(t),\nu] * \mathscr{F}[w(t),\nu] = F(\nu) * W(\nu)$$

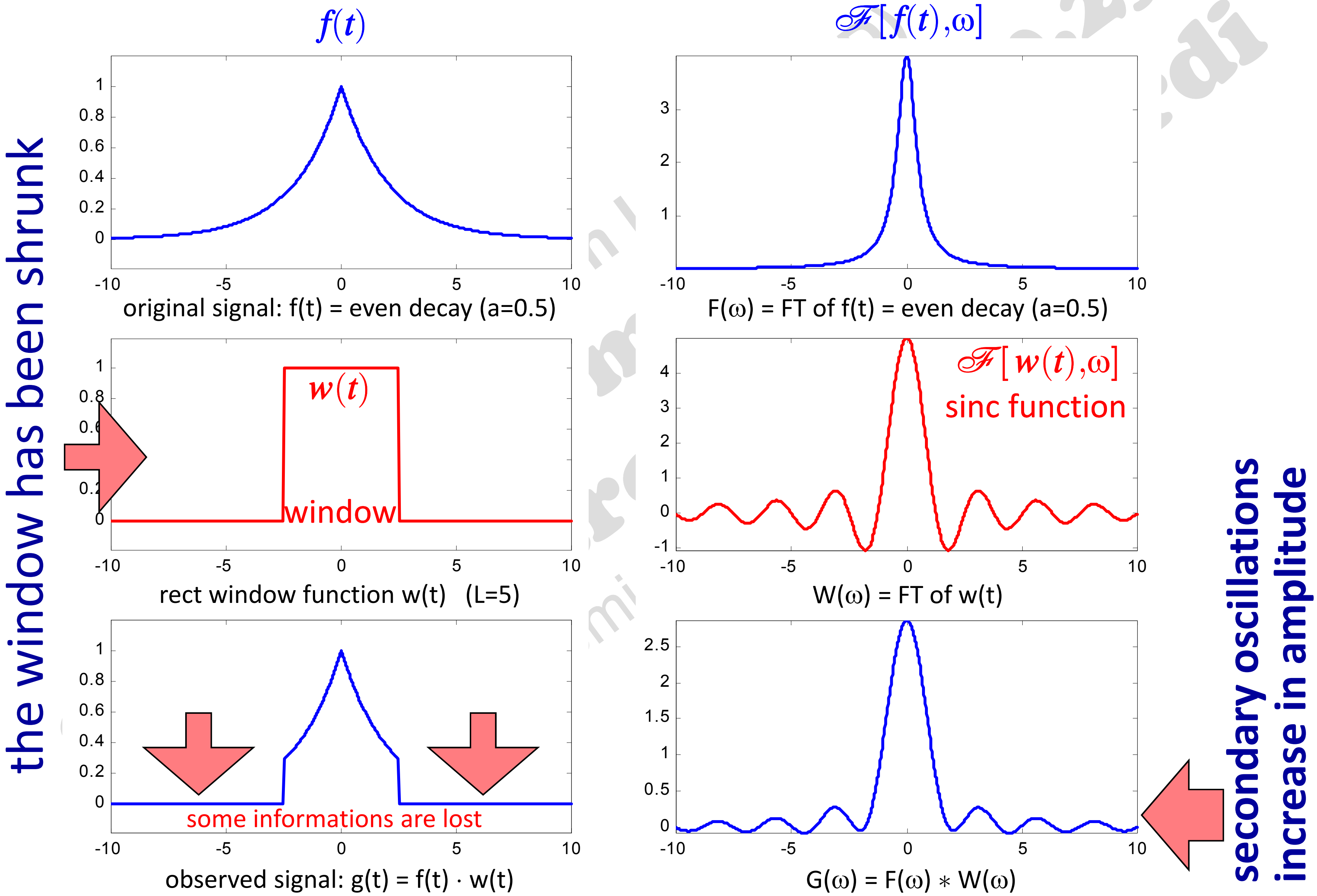
where $F(\nu)*W(\nu)$ represents the convolution product of $F$ and $W$. Remember that $W$ is a `sinc()`.

# Windowing effect: example



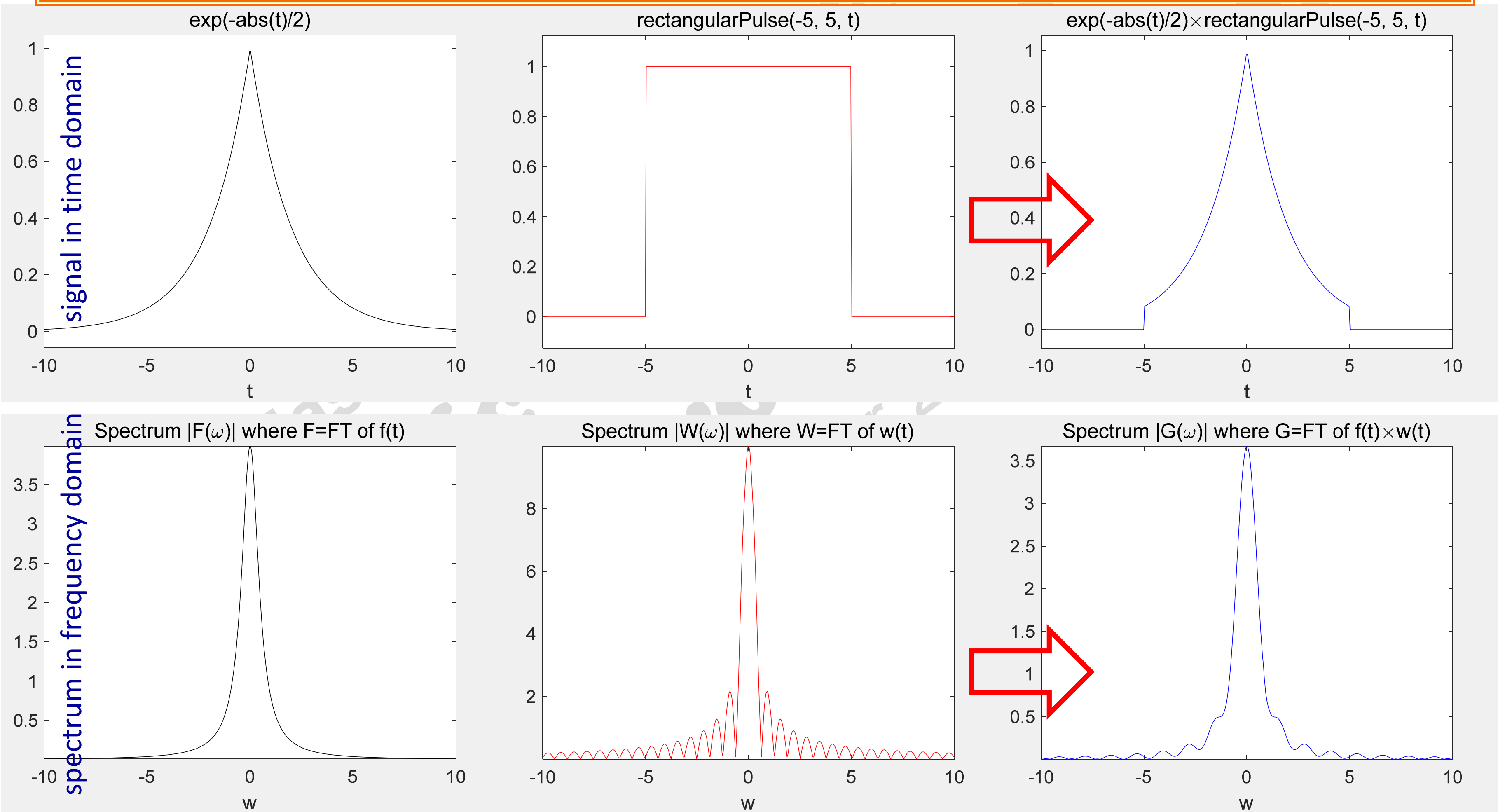

secondary oscillations

# Windowing effect: example (cont.)



# Example by means of Symbolic Math Toolbox

```
syms t real;  f=exp(-abs(t)/2);            figure; ezplot(f,[-10 10])
L=10;  w=rectangularPulse(-L/2,+L/2,t);  figure; ezplot(w,[-10 10])
figure; ezplot( f*w, [-10 10])
F=fourier(f);    figure; h=ezplot(abs(F),[-10 10]);  set(h,'Color','k')
W=fourier(w);    figure; h=ezplot(abs(W),[-10 10]); set(h,'Color','r')
G=fourier(f*w); figure; h=ezplot(abs(G),[-10 10]);  set(h,'Color','b')
```




**Exercise:** What happens for L=5?

# Aliasing effect

When performing the spectral analysis of a sample from a physical phenomenon $f(t)$, we are actually using a discrete set of observations of the phenomenon obtained by equispaced time series.

From a mathematical point of view, this process is equivalent to the multiplication of the function $f(t)$, which describes the phenomenon, by a Comb function $\delta_T(t)$ (of step $T$); thus the "observed" function is:

$$h(t) = f(t)\cdot\delta_T(t)$$

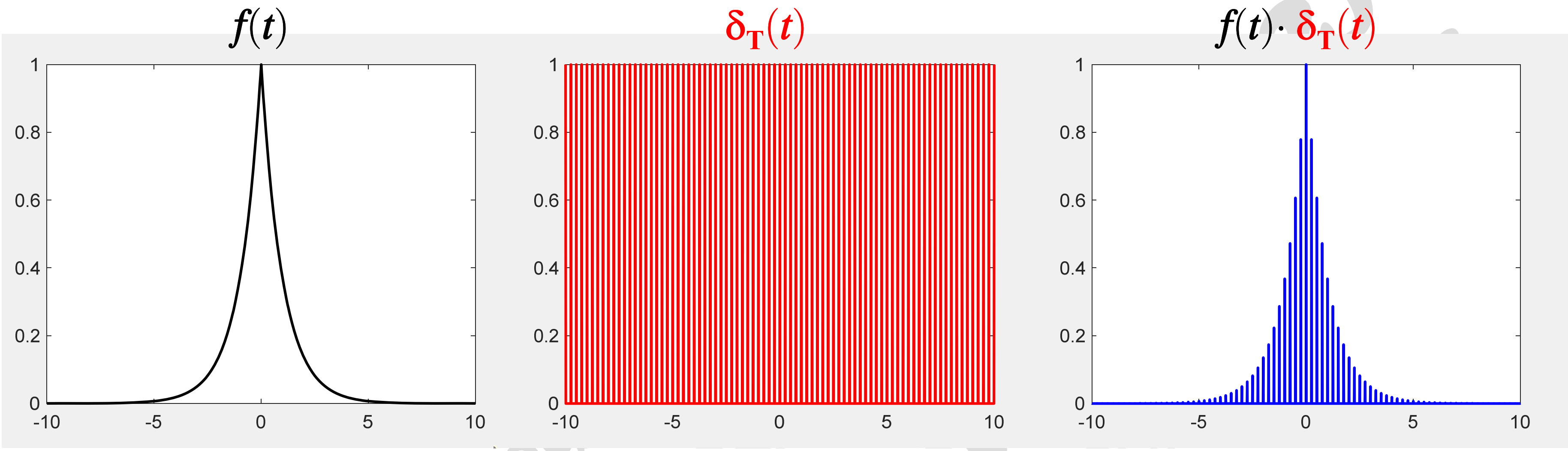
$$F(\nu) = \mathscr{F}[f(t),\nu]$$
$$\delta_{1/T}(\nu) = \mathscr{F}[\delta_T(t),\nu]$$

so that the ***Fourier Transform*** $H(\nu)$ of $h(t)$ is then given by:

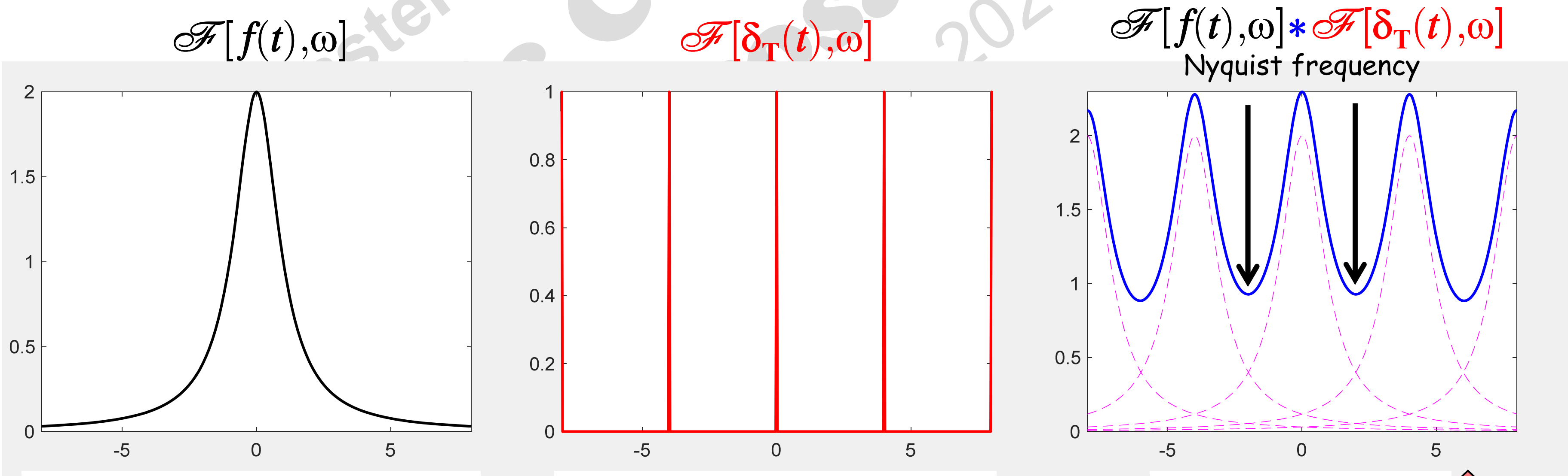
$$H(\omega) = \mathscr{F}[h(t),\omega] = \mathscr{F}[f(t),\omega] * \mathscr{F}[\delta_T(t),\omega] = F(\omega) * \delta_{1/T}(\omega)/T$$

where $F(\nu) * \delta_{1/T}(\nu)/T$ represents the convolution product of $F$ and $\delta_{1/T}/T$. Remember that the Fourier Transform of a comb function is still a comb function, but with a period equal to the reciprocal of the period of the original comb function.

# Aliasing effect: example




original signal: f(t)=even decay $e^{-|t|}$ pulse train $\delta_T(t)$, T = 0.25 observed signal: $h(t) = f(t) \cdot \delta_T(t)$



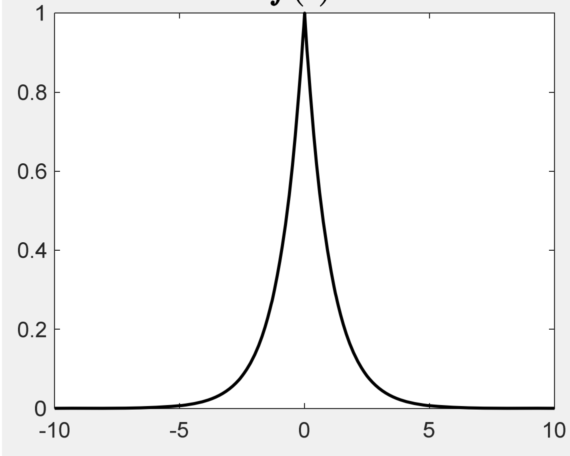

$F(\omega) = 2/(1+\omega^2)$ = FT of $f(t) = e^{-|t|}$ $\delta_P(\omega)$ = FT of $\delta_T(t)$, P = 1/T = 4 $H(\omega) = F(\omega) * \delta_{1/T}(\omega)$

Superposition of $F(\omega)$

# Aliasing effect: example (cont.)

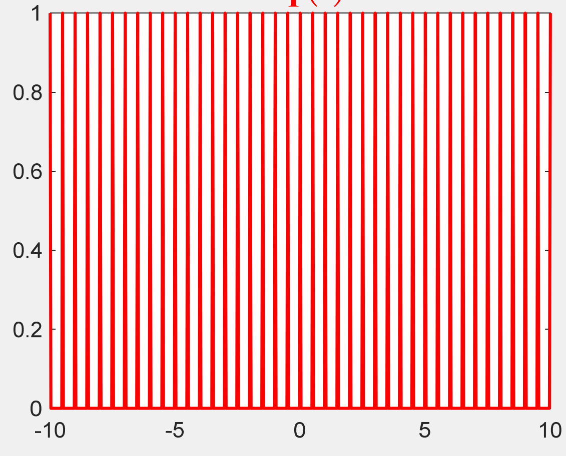
Now, let's double the period T of the Comb Function $\delta_T$

$f(t)$

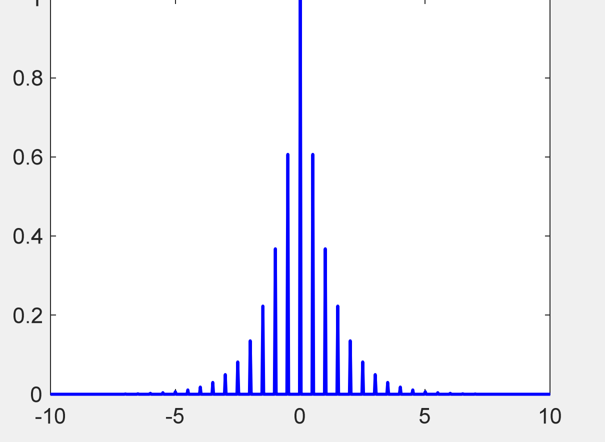


original signal: f(t)=even decay $e^{-|t|}$

$\delta_T(t)$

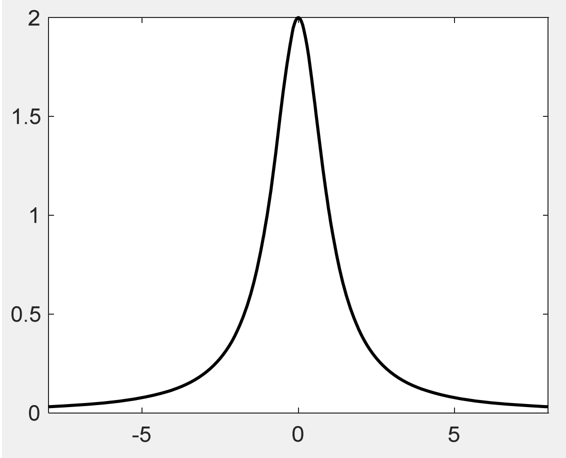


pulse train $\delta_T(t)$, T = 0.5



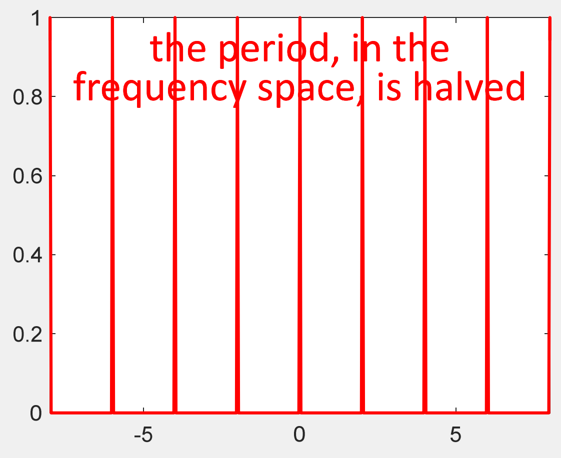

observed signal: $h(t) = f(t) \cdot \delta_T(t)$

$\mathscr{F}[f(t),\omega]$



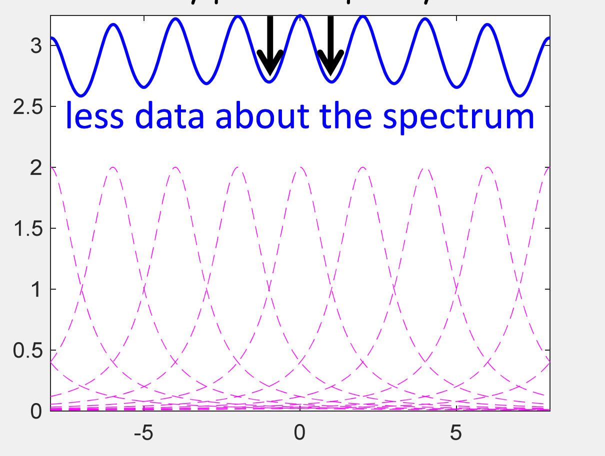
$F(\omega) = 2/(1+\omega^2)$ = FT of $f(t) = e^{-|t|}$

$\mathscr{F}[\delta_T(t),\omega]$




$\delta_P(\omega)$ = FT of $\delta_T(t)$, P = 1/T = 2

$\mathscr{F}[f(t),\omega] * \mathscr{F}[\delta_T(t),\omega]$

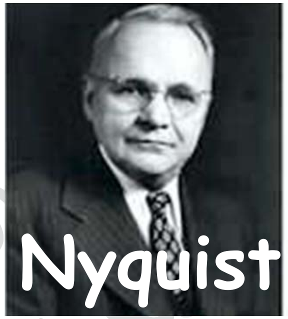


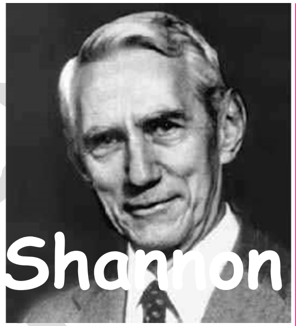

$H(\omega) = F(\omega) * \delta_{1/T}(\omega)$

Superposition of $F(\omega)$

# The Nyquist-Shannon Sampling Theorem

stated by Harry Nyquist and proved by Claude Shannon in 1948




w.r.t. the circular frequency $\nu$

If the *FT* of $f(t)$, $F(\nu)$ $[\omega=2\pi\nu]$, is such that

$$F(\nu)=0,\ |\nu|>H/2$$

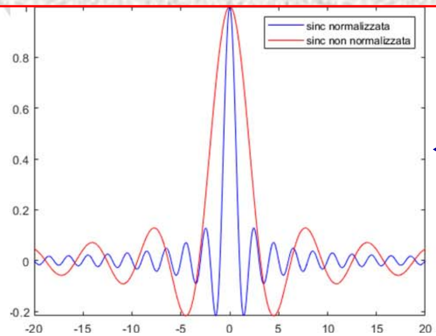
($f$ is a **band limited function** of bandwidth $H$)

then the values of $f$ at the sampling points $t_k=k\Delta t$, taken with frequency $f_s=1/\Delta t$, if $f_s>H$ (greater than twice the maximum frequency) allow $f$ to be fully restored, i.e.:

$$f(t)=\sum_{k=-\infty}^{+\infty} f(k\Delta t)\ \mathrm{sinc}\left(\frac{t}{\Delta t}-k\right)$$

$$\mathrm{sinc}(x)=\frac{\sin(\pi x)}{\pi x}$$

normalized sinc: $\sin(\pi x)/(\pi x)$
unnormalized sinc $\sin(x)/(x)$




**sinc:
"sine cardinal" function**

new expansion basis

The **Aliasing Error** is governed by the Sampling Theorem.

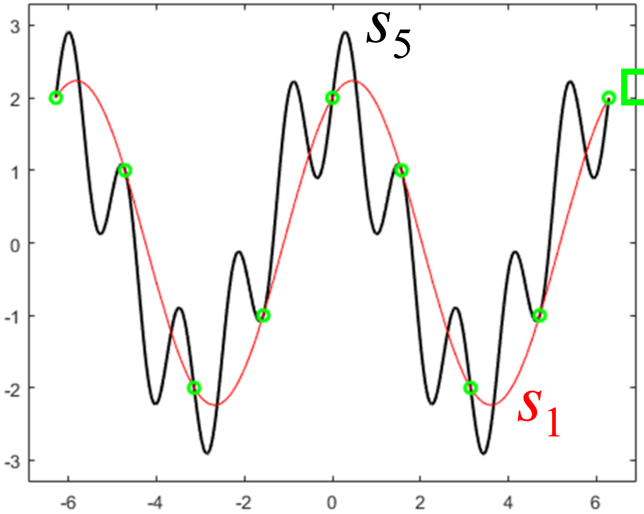
The **Sampling Theorem** can be thought of as the conversion of an analog signal into a discrete form by taking the sampling frequency at least as twice the maximum frequency of the input analog signal.

This **Theorem** is the basis of digital recordings able to reach a degree of fidelity much higher than analog ones.

# Example: application of the Sampling Theorem

Let us consider **two signals**: $s_5 = 2\cos(t) + \sin(5t)$ and $s_1 = 2\cos(t) + \sin(t)$.

We want to reconstruct $s_5$ starting from a sequence of its equispaced samples.
We sample $s_5$ with a step $h = \pi/2$ in the interval $[-2\pi, +2\pi]$ getting 9 samples.



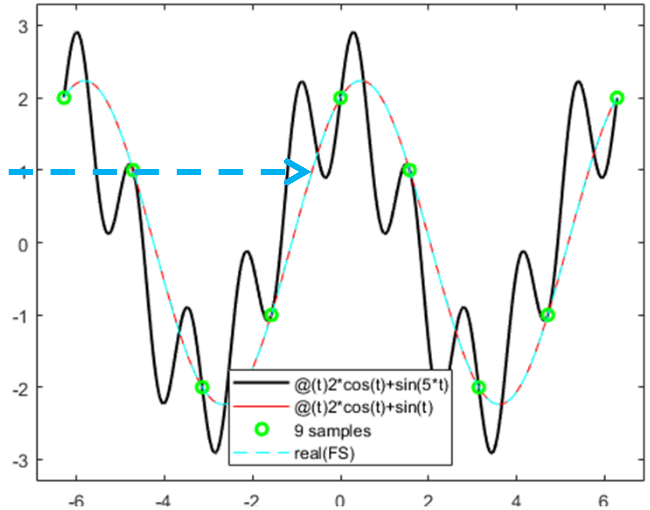

But $s_5$ and $s_1$ share these samples and we are not able to distinguish the two signals starting from these 9 samples, since the sampling rate **does not satisfy** the Sampling Theorem.

max freq in $s_5$: $\omega_{max} = 5$ ⇨ $\nu = \omega/(2\pi)$

min useful freq for $s_5$: $f_s > 2\nu_{max} = 5/\pi$ ⇨

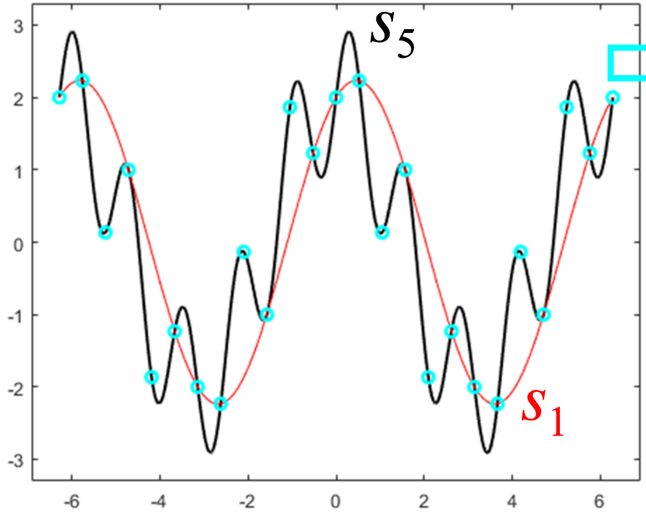
According to the Sampling Theor. we should use a step: $\Delta t < \pi/5$ but used step = $\pi/2 > \pi/5$

In facts, if we reconstruct a signal by means of the Fourier Series starting from the 9 samples (whose frequency is $f_s = 2/\pi < 5/\pi = 2\nu_{max}$), we get $s_1$ and not $s_5$ (aliasing effect).



# **Example:** application of the Sampling Theorem (cont.)

Now we sample $s_5$ with a step $h=\pi/6<\pi/5$, that satisfies the Sampling Theor.: we get 25 samples.



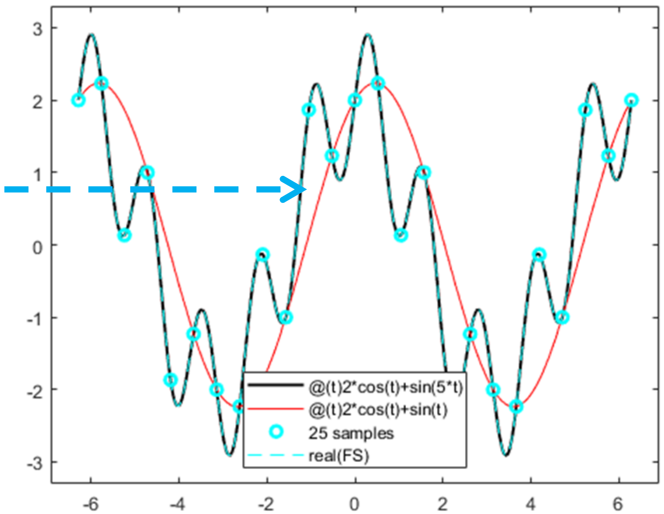

The figure shows that some of the 25 samples of $s_5$ no longer fall on $s_1$, so that we are able to distinguish the two signals starting from these 25 samples.

max freq in $s_5$: $\omega_{max}=5$ ⇨ $\nu=\omega/(2\pi)$

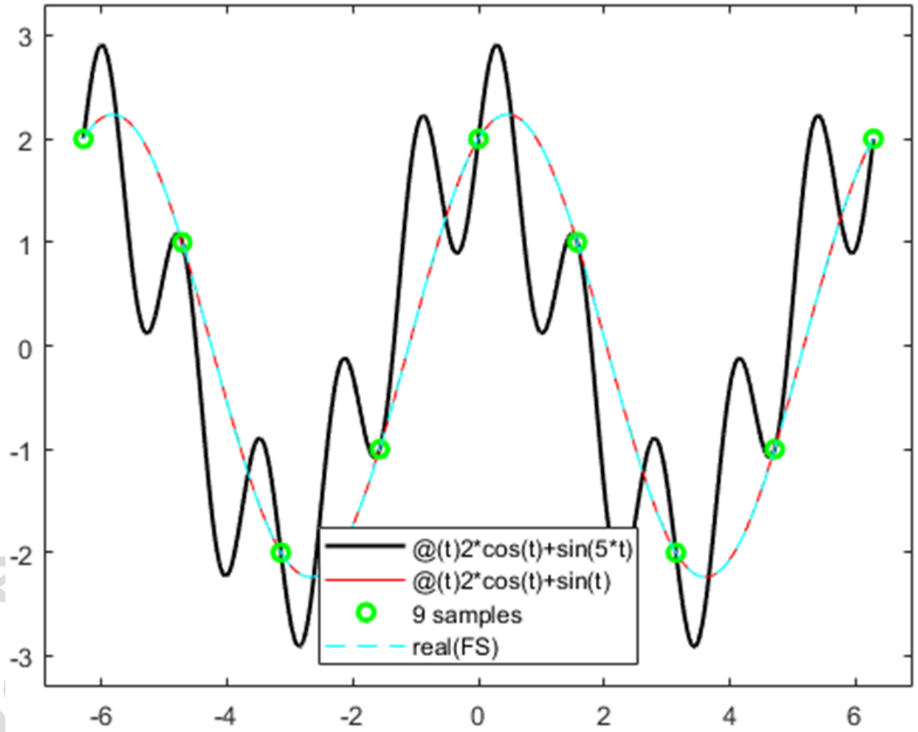
min useful freq for $s_5$: $f_s>2\nu_{max}=5/\pi$ ⇨

According to the Sampling Theor. we should use a step: $\Delta t<\pi/5$ and used step = $\pi/6<\pi/5$

In facts, if we reconstruct a signal by means of the Fourier Series starting from the 25 samples (whose frequency is $f_s=6/\pi>5/\pi=2\nu_{max}$), now we get $s_5$ (no aliasing effect).



# Quiz




Why, starting from the samples of $s_5=2\cos(t)+\sin(5t)$ taken with frequency $f_s=2/\pi<5/\pi=2\nu_{max}$, are we able to reconstruct the signal $s_1=2\cos(t)+\sin(t)$ with angular frequency $=1$, and not:

$$s_k=2\cos(t)+\sin(kt),\ k=2,3,4$$

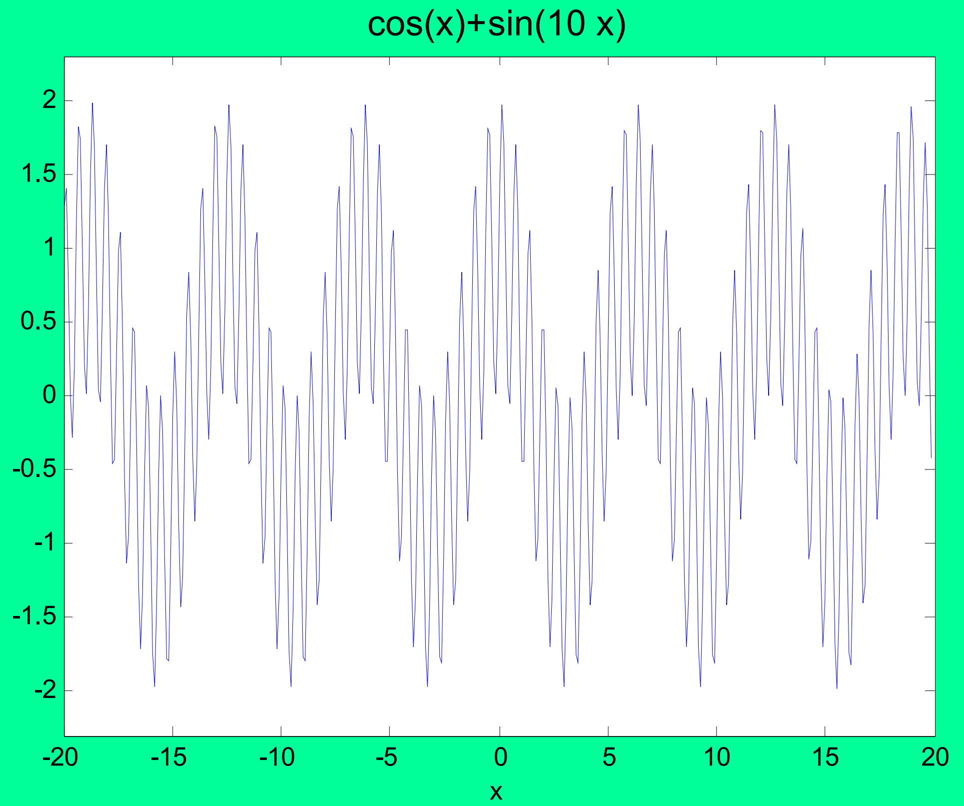
with maximum frequency $\omega=k<5$?



(prof. M. Rizzardi)

# Exercise

What is the Nyquist frequency of the following signal?




Check it by means the Sampling Theor. and display what happens with a lower frequency.

# Connection between the Fourier Transform and Fourier Series coefficients

FT

$$F(\nu) = \int_{-\infty}^{+\infty} f(t) e^{-2\pi i \nu t} \mathrm{d}t \quad \Rightarrow \quad F\left(\frac{k}{T}\right) = \int_{-\infty}^{+\infty} f(t) e^{-2\pi i \frac{k}{T} t} \mathrm{d}t$$

$$\nu = \frac{k}{T}$$

$$f(t) = 0 \quad |t| > \frac{T}{2} \quad \Rightarrow \quad \frac{1}{T} F\left(\frac{k}{T}\right) = \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} f(t) e^{-2\pi i \frac{k}{T} t} \mathrm{d}t$$

$f$: time-limited function

$$\gamma_k = \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} f(t) e^{-2\pi i \frac{k}{T} t} \mathrm{d}t$$

$\gamma_k$: $k^{\text{th}}$ Fourier coefficient of $f$ in $[-T/2, +T/2]$

for a time-limited function $f(t)$:

$$F(\nu_k) = F\left(\frac{k}{T}\right) = T \cdot \gamma_k$$

We can apply the same approximation **algorithm** of Fourier coefficients, with an extra step.

# Algorithm for the numerical approx. of $F(\nu_k)$



Input: $N+1$ equispaced samples $f_j = f(t_j)$ ($N$: even) in $[-T/2, +T/2]$.

Output: $N+1$ equispaced samples $F_k \approx F(\nu_k)$ in $[-\Omega/2, +\Omega/2]$, where

$$N = \Omega\, T$$

1. Define the vector of samples **f**: $\begin{cases} \mathbf{f}_0 = \frac{1}{2}\left[ f(t_0) + f(t_N) \right] \\ \mathbf{f}_j = f(t_j), \qquad j = 1, \ldots, N-1 \end{cases}$

2. Compute the **DFT** (MATLAB `fft()`).

3. Reorder the vector (MATLAB `fftshift()`).

4. Add the last component* and the scale factors $\left( (-1)^k T/N, \;\; k = -N/2, \ldots, +N/2 \right)$.

   * equal to the first

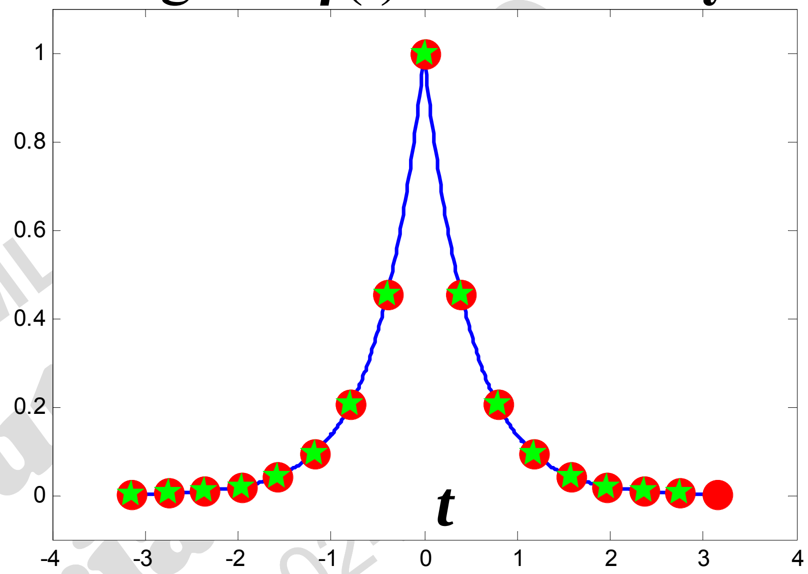
new 5. Compute the frequencies: $\nu_k = \dfrac{k}{T}, \quad k = -\frac{N}{2}, \ldots, 0, \ldots, +\frac{N}{2}$.

# MATLAB example: FT

```
pf=@(t) exp(-2*abs(t));
T=2*pi; N=16;   tj in [−T/2, +T/2]
tj=T/N*(-N/2:N/2)'; fj=pf(tj);          samples
f=[.5*(fj(1)+fj(end)); fj(2:end-1)];    FT
F=fftshift(fft(f));     F=[F; F(1)]*T/N;
F(2:2:end)=-F(2:2:end); nu=(-N/2:N/2)'/T;
G=F(1:end-1);                            IFT
G(2:2:end)=-G(2:2:end);
g=ifft(fftshift(G)); g=[g; g(1)]/T*N;
x=linspace(-T/2,T/2,499); y=pf(x);
```

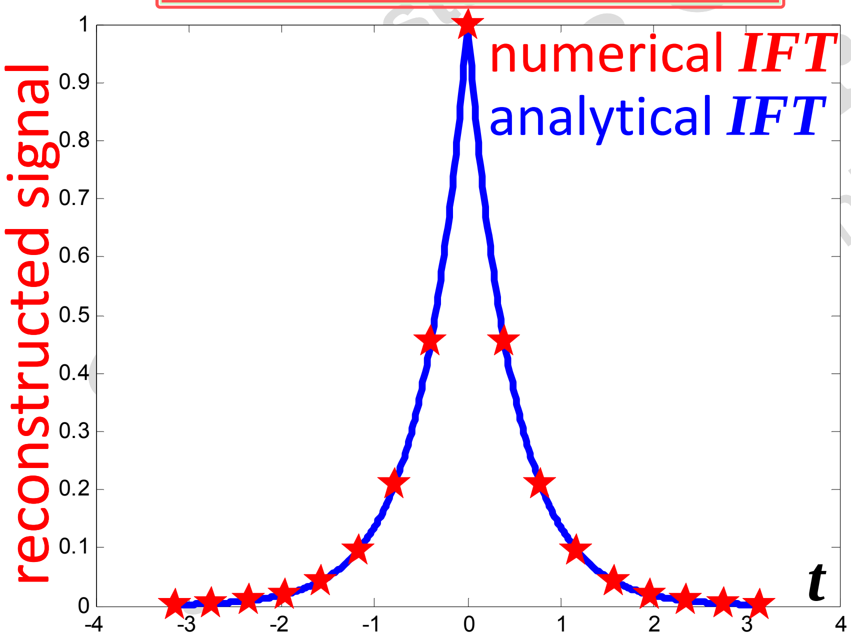
the same algorithm as for Fourier Synthesis with series

signal: *f*(*t*)=even decay

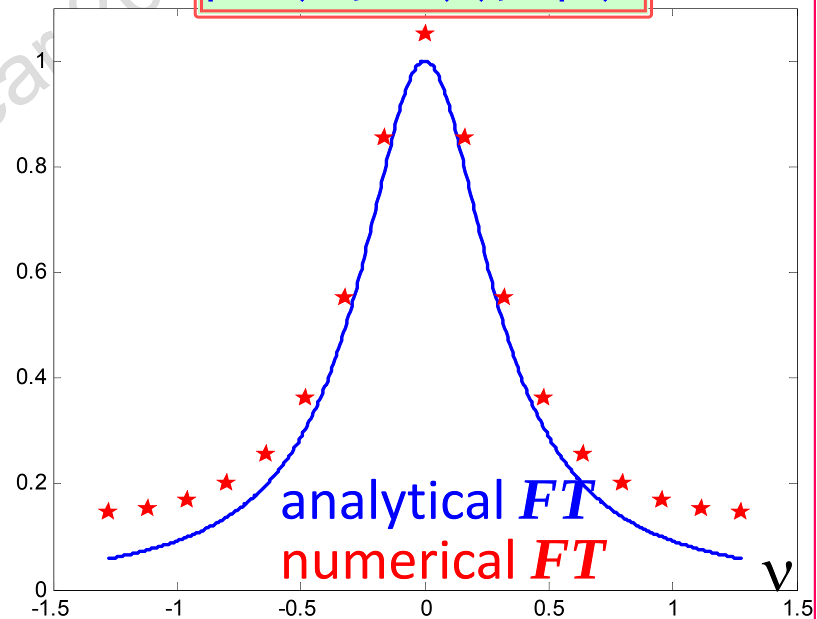



```
plot(x,y,tj,fj,'ro',tj(1:end-1),f,'gp')
```

```
plot(x,y,'b',tj,real(g),'pr')
```




```
plot(nu,abs(F),'rp')
```








(prof. M. Rizzardi)

# How can we add the analytical Fourier Transform for a comparison with its numerical samples?



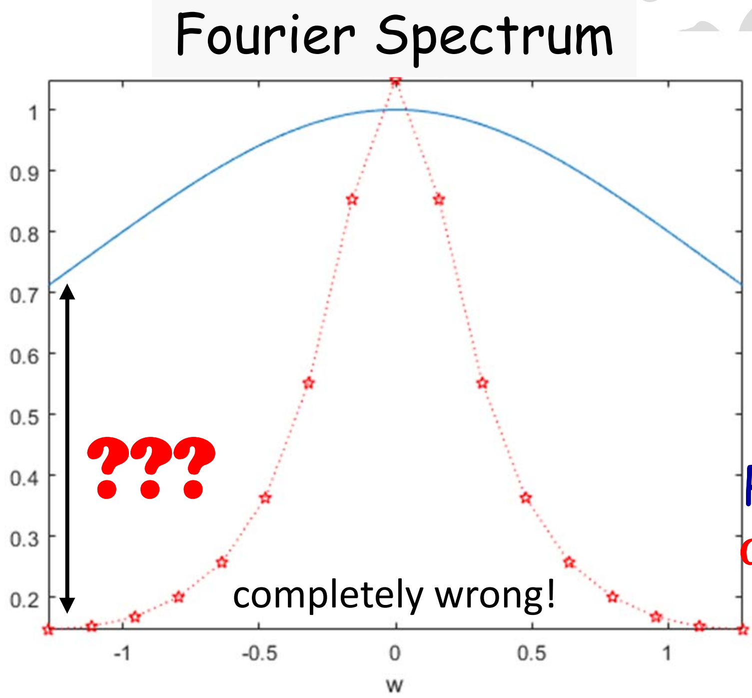
```
pf=@(t)exp(-2*abs(t));  T=2*pi; N=16;  tj=T/N*(-N/2:N/2)'; fj=pf(tj);
f=[.5*(fj(1)+fj(end)); fj(2:end-1)]; F=fftshift(fft(f)); F=[F; F(1)]*T/N;
F(2:2:end)=-F(2:2:end); nu=(-N/2:N/2)'/T;
plot(nu,abs(F),'pr:')
```

numerical FT

```
syms t real
Fw=fourier(sym(pf));
hold on; fplot(abs(Fw),[-1.5,1.5])
axis tight
```

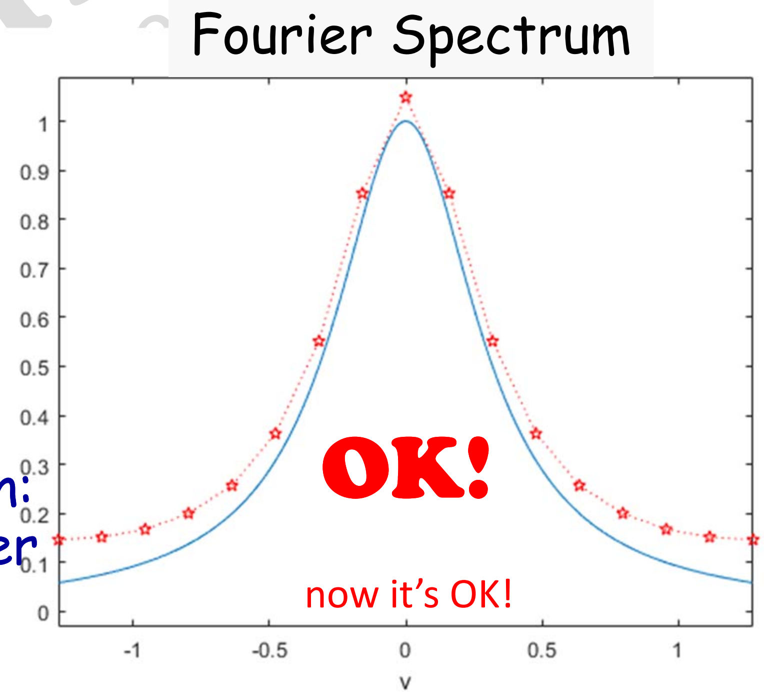
```
syms t w v real; O=N/T;
Fw=fourier(sym(pf));
Fv=subs(Fw,w,2*pi*v);
hold on; fplot(abs(Fv),[-O/2,O/2])
```

$\omega = 2\pi\nu$




numerical FT uses circular frequency

Pay attention: $\omega$ and $\nu$ differ




$\omega \longleftrightarrow \nu$

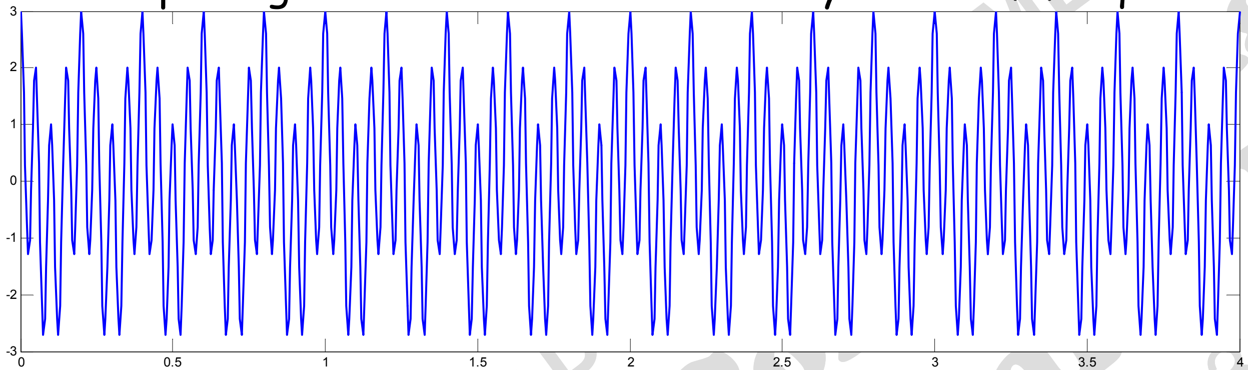


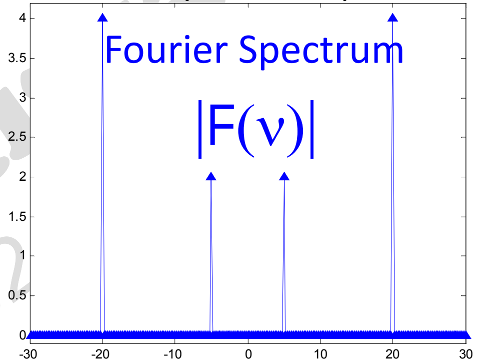
(prof. M. Rizzardi)

Applications **example 1:** ideal filter

```
dt=.005; t=(0:dt:4)'; N=numel(t)-1; T=4;
fj=cos(2*pi*5*t)+2*cos(2*pi*20*t); plot(t,fj); axis([0 4 -3 3]);
f=[.5*(fj(1)+fj(end));fj(2:end-1)];                          FT
F=fftshift(fft(f));F=[F;F(1)]*T/N; F(2:2:end)=-F(2:2:end);
nu=(-N/2:N/2)'/T;
figure; plot(nu,abs(F),'^-'); axis([-30 30 -.1 4.2])
```

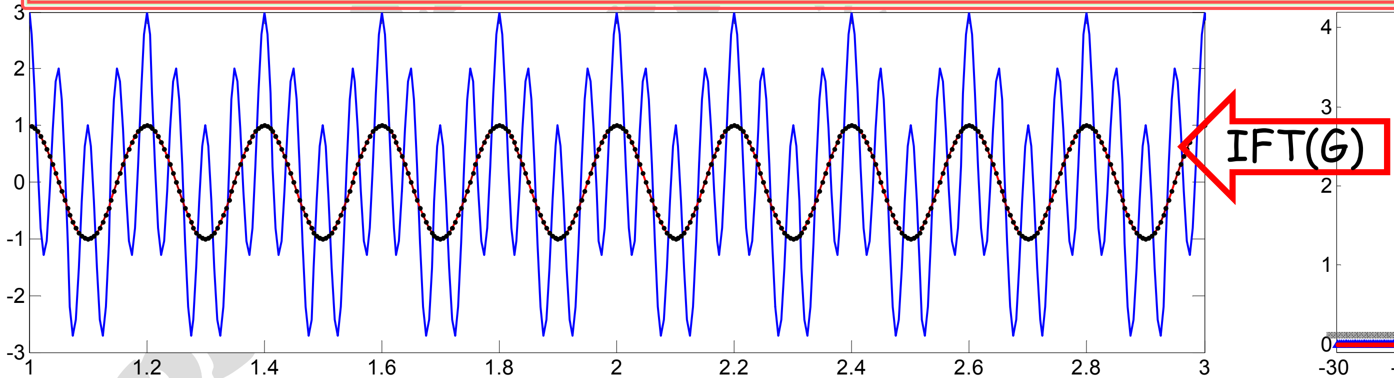
The input signal contains two elementary waves of frequencies 5 and 20 respectively

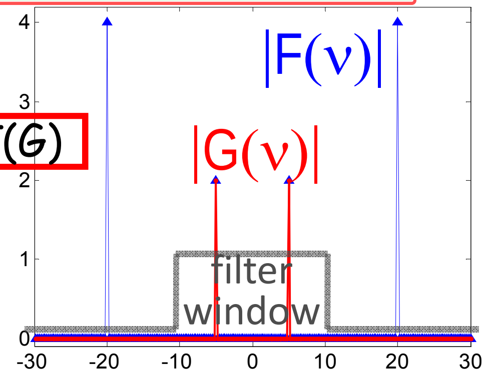





We apply a "low-pass filter" to the input signal, i.e. freqs ν : |ν|>W/2=10 are set to 0

```
G=zeros(size(F)); k=find(abs(nu)<10); G(k)=F(k); hold on; plot(nu,abs(G),'r.-')
H=G(1:end-1); H(2:2:end)=-H(2:2:end);
h=ifft(fftshift(H)); h=[h;h(1)]/T*N;     IFT
figure; plot(t,fj,t,real(h),'r', t,cos(2*pi*5*t),'.k--'); axis([1 3 -3 3])
```



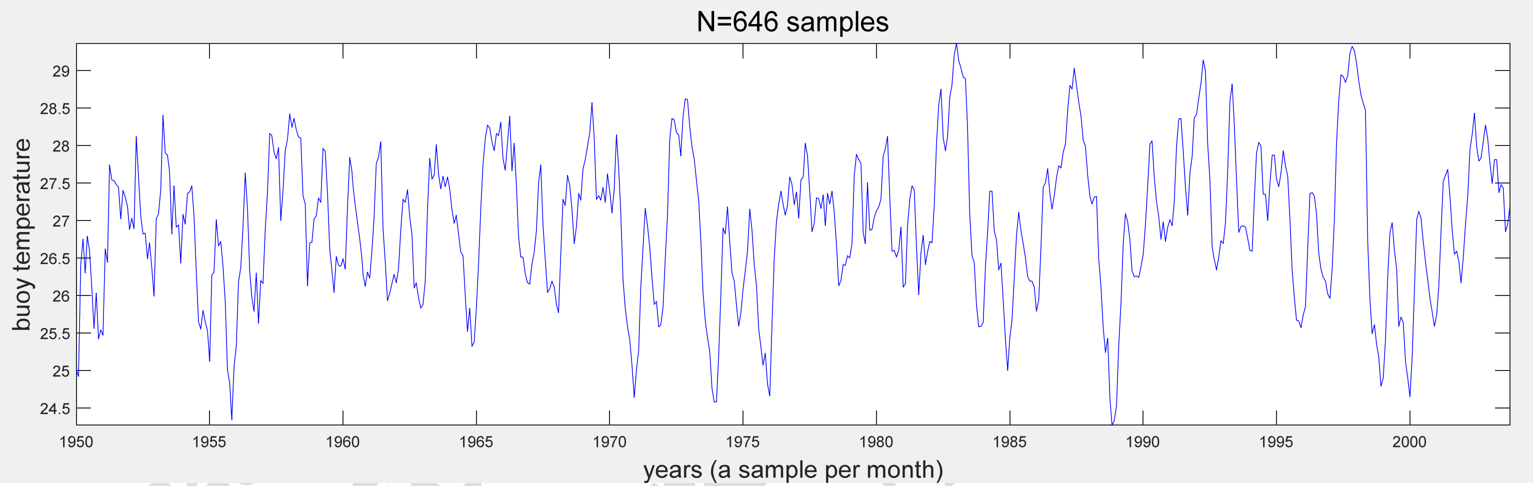

The two signals **IFT(G)** and cos(2π5t) overlap perfectly. The "noise" 2cos(2π20t) has been totally removed.

# Applications example 2

## Study of a possible periodicity of equispaced data: how to find the period of "El Ninho"?

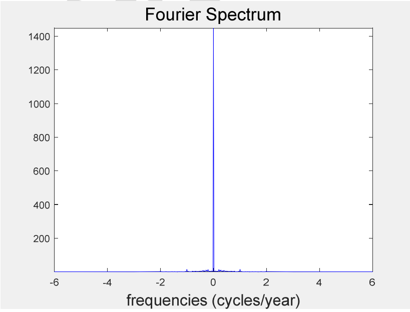
```
load ninho; N=numel(fj);
tj=(0:N-1)'/12 + 1950; % time in years
plot(tj,fj,'b'); ylabel('temperatura boe'); xlabel('anni (un campione per mese)')
```

download ninho.mat

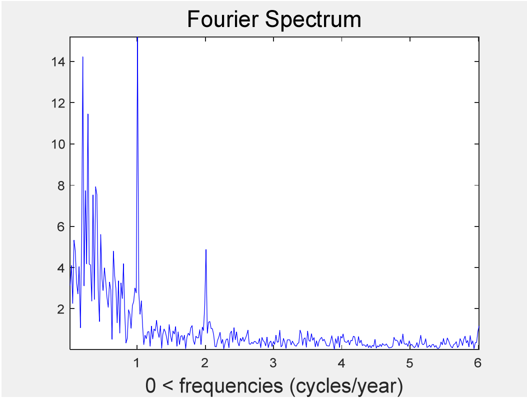



```
T=tj(end)-tj(1); Dt=T/N; % about 1 sample per month
Dnu=1/T; nu=(-N/2:N/2)'*Dnu;
F=fftshift(fft(fj)); F=[F; F(1)]*T/N;
F(1:2:end)=-F(1:2:end);
plot(nu,abs,'b'); axis tight
title('Spettro di Fourier')
xlabel('frequenze (cicli/anno)')
```

```
mid=N/2+2; plot(nu(mid:end),abs(F(mid:end)),'b')
axis tight; title('Spettro','FontSize',18)
xlabel('0<frequencies(cycles/year)','FontSize',16)
```




only positive frequencies

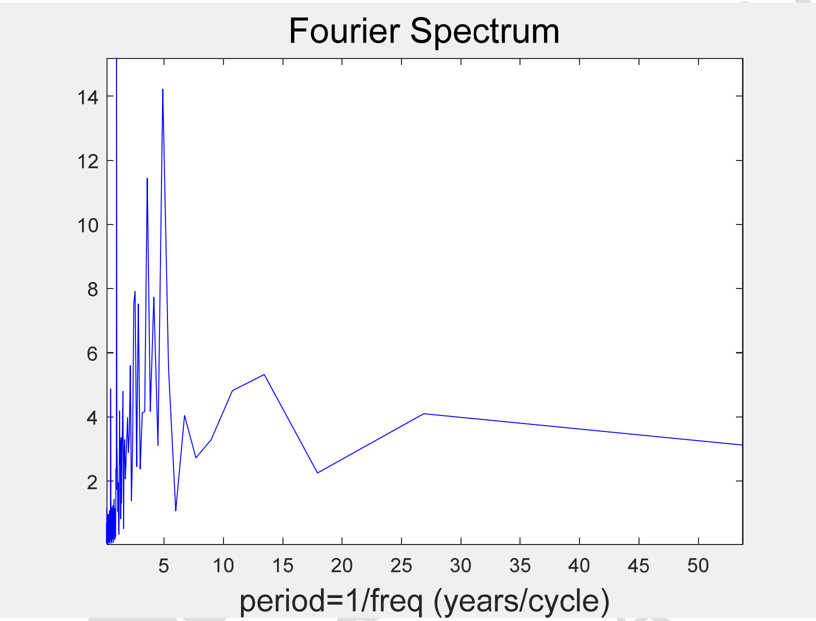


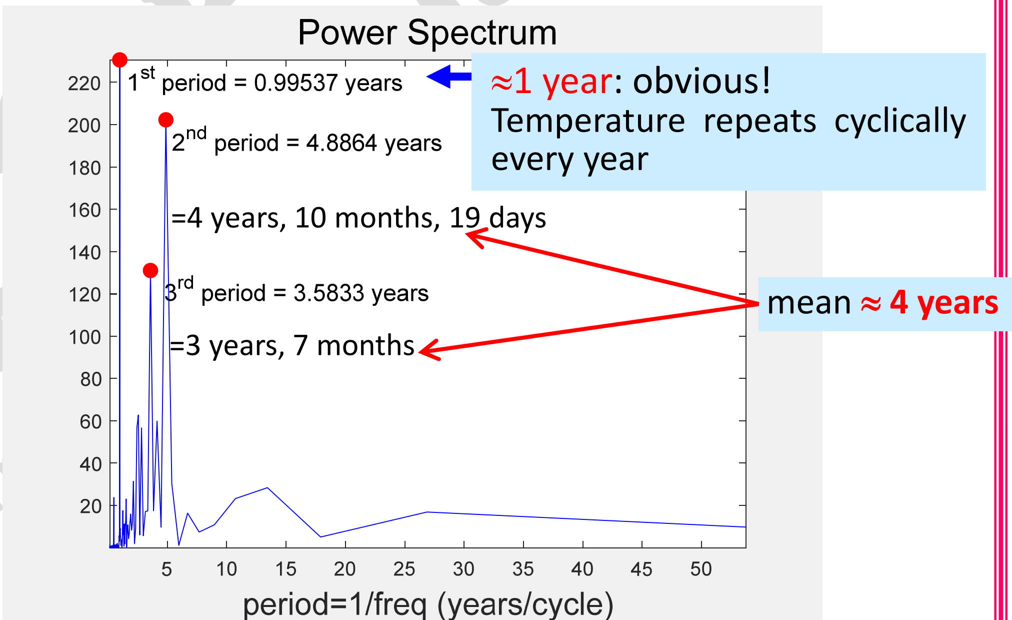
Applications
**example 2**

# period of "El Ninho"

```
mid=N/2+2;  % only freq > 0
period=1./nu(mid:end);  % instead of frequencies, on abscissas we use periods (in years)
plot(period,abs(F(mid:end)),'b'); axis tight
xlabel('period=1/freq (years/cycle)','FontSize',16)
title('Spectrum','FontSize',18)
power=abs(F(mid:end)).^2;
plot(period,power,'b'); axis tight
title('Power Spectrum','FontSize',18); xlabel(…); hold on
```

Power Spectrum = $\{|F_k|^2\}_k$



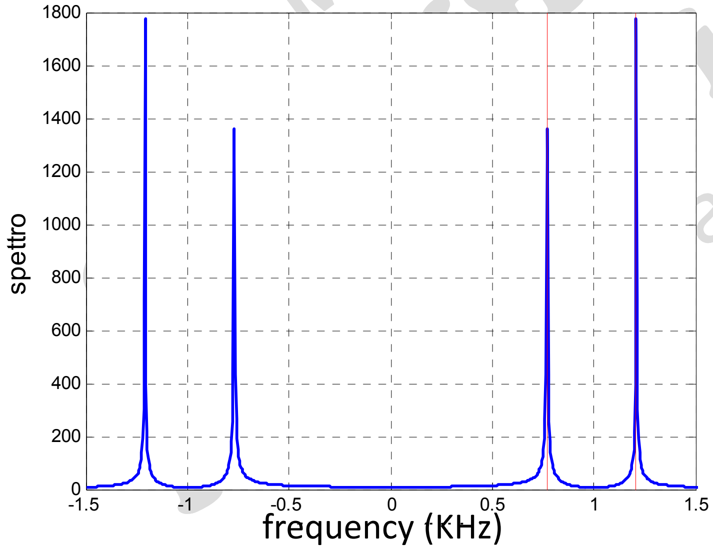

```
%% find the index of the 1st maximum in the Power Spectrum
index=find(power == max(power));  % 1st
p1=period(index); plot(p1,power(index),'r.', 'MarkerSize',25)
text(period(index),power(index),[' 1^{st} period = ' num2str(period(index)) ' years'])
```

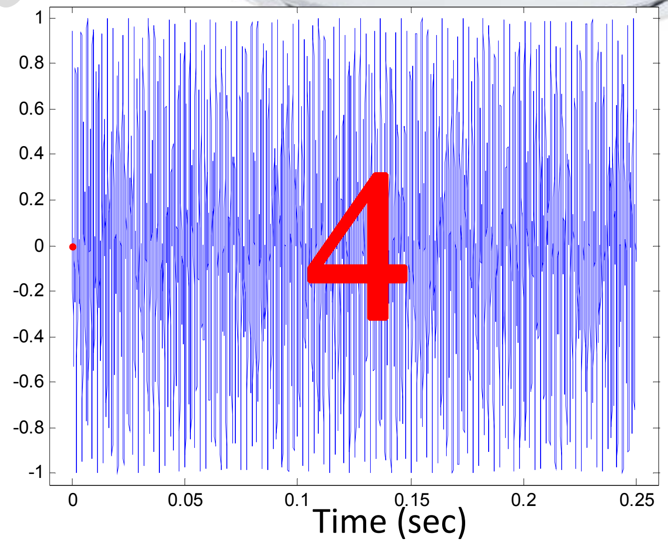
Applications example 3

# Dual-tone multi-frequency (DTMF) phone keypad

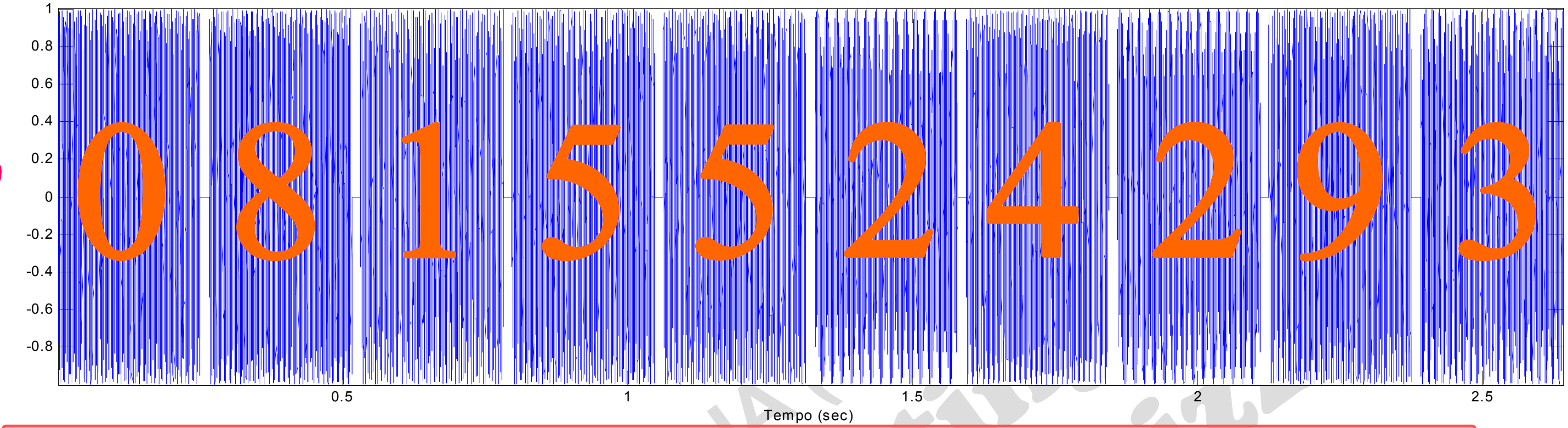
The sound ***y*** of each key is the sum of two “tones”, i.e. two sinusoids of suitable frequencies:

$$y = \frac{\sin(2\pi\phi_{row}t) + \sin(2\pi\phi_{col}t)}{2}$$

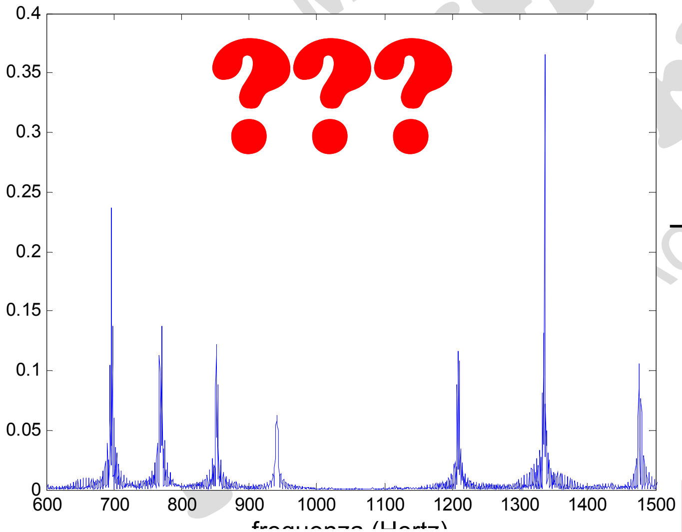



# example 3

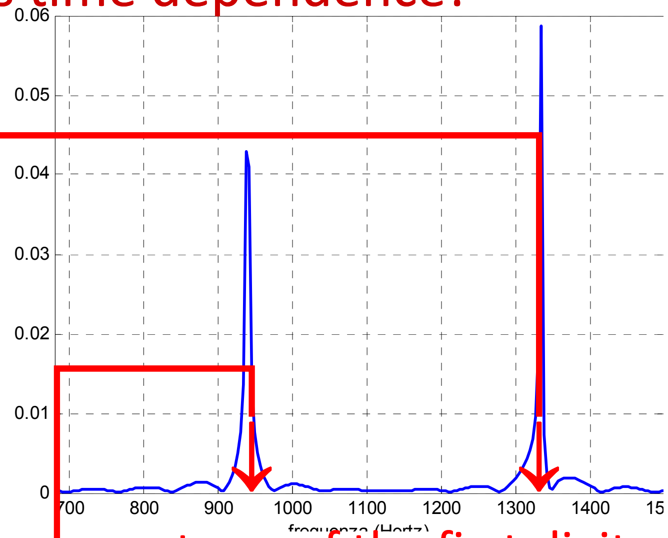



```
info=audioinfo('PhoneNumber.wav')
[y,fs]=audioread('PhoneNumber.wav'); sound(y,fs) % fs: sample rate
N=numel(y); Dt=1/fs; tj=Dt*(1:N)'; T=N*Dt;
figure(1); plot(tj,y,'b'); axis tight; xlabel('Tempo (sec)')
Y=fftshift(fft(y)); Y=[Y;Y(1)]*T/N; Dnu=1/T; nuk=(-N/2:N/2)'/T;
figure(2); plot(nuk,abs(Y),'b');
xlabel('frequency (Hertz)'); ylabel('Spectrum of the whole phone number')
```

the sound varies with time: the spectrum of the whole signal contains all the component frequencies, without showing its time dependence!



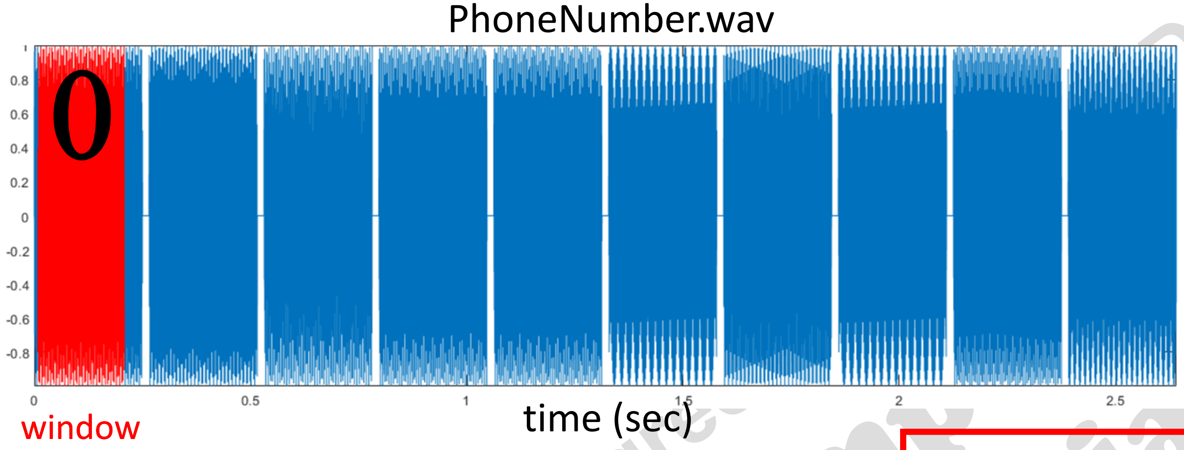

| Hz | 1209 | 1336 | 1477 |
|---|---|---|---|
| 697 | 1 | 2 | 3 |
| 770 | 4 | 5 | 6 |
| 852 | 7 | 8 | 9 |
| 941 | * | 0 | # |



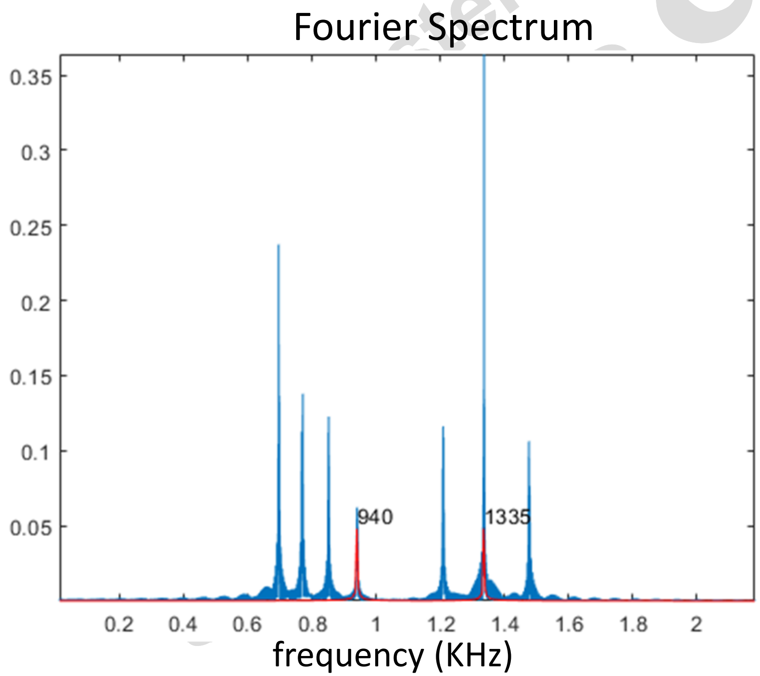
spectrum of the first digit

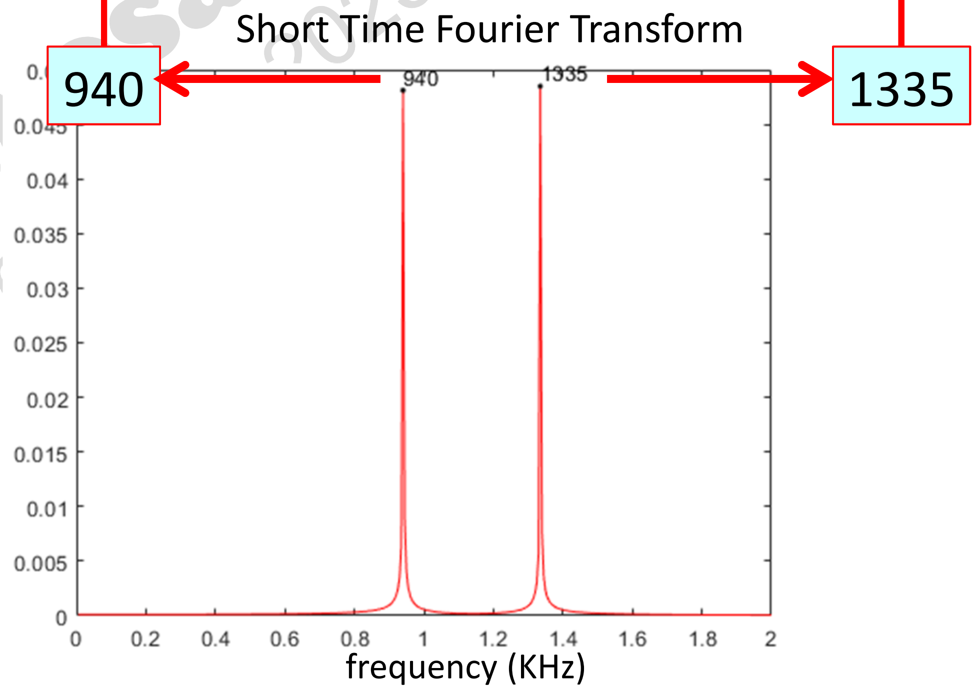
# A **moving window** should be applied to the signal: "Short Time Fourier Transform (STFT)"

Download `STFT.p` and PhoneNum.zip



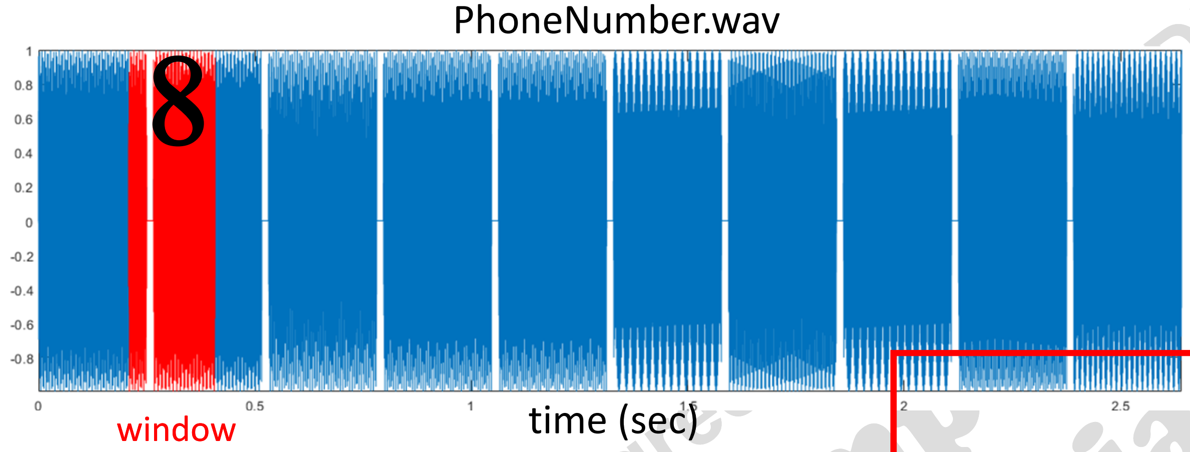

| Hz | 1209 | 1336 | 1477 |
|---|---|---|---|
| 697 | 1 | 2 | 3 |
| 770 | 4 | 5 | 6 |
| 852 | 7 | 8 | 9 |
| 941 | * | 0 | # |



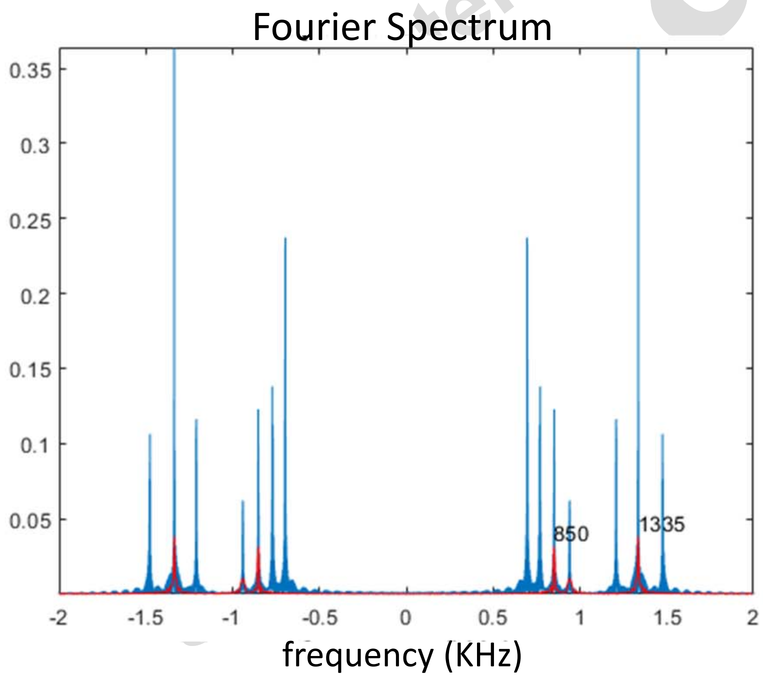



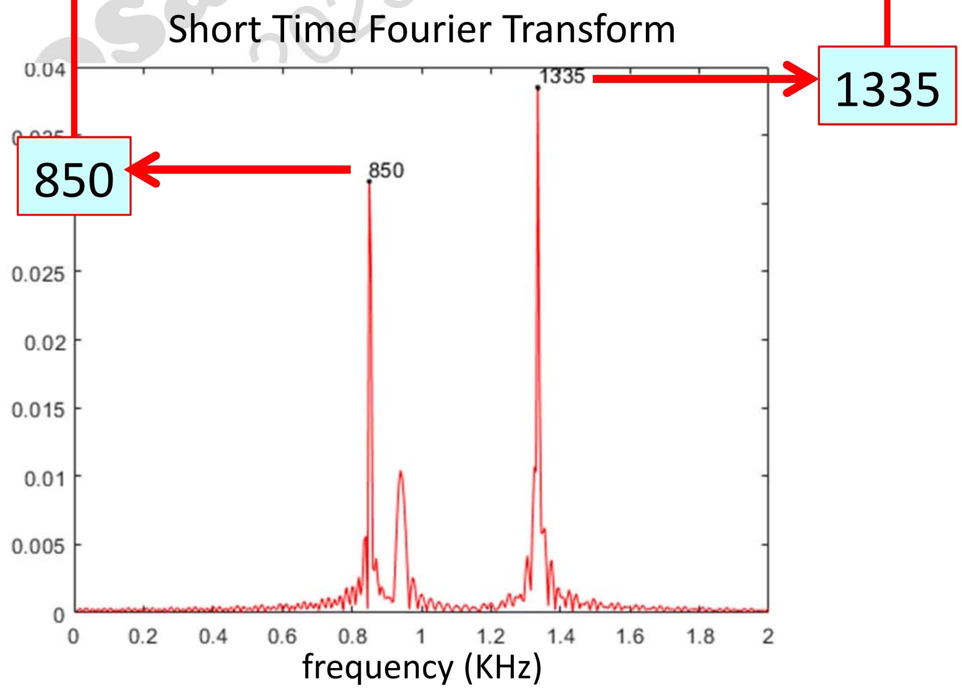
# A moving window should be applied to the signal: "Short Time Fourier Transform (STFT)"

Download **STFT.p** and PhoneNum.zip



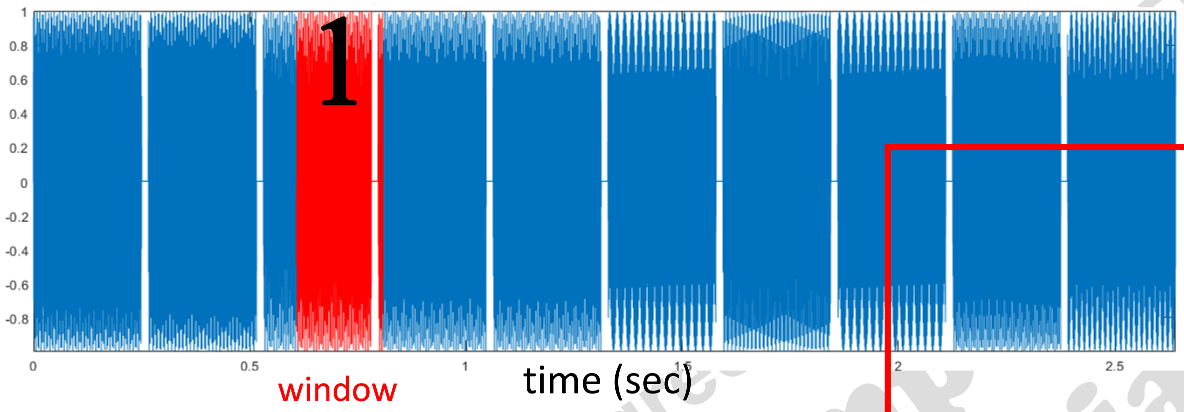

| Hz | 1209 | 1336 | 1477 |
|---|---|---|---|
| 697 | 1 | 2 | 3 |
| 770 | 4 | 5 | 6 |
| 852 | 7 | 8 | 9 |
| 941 | * | 0 | # |



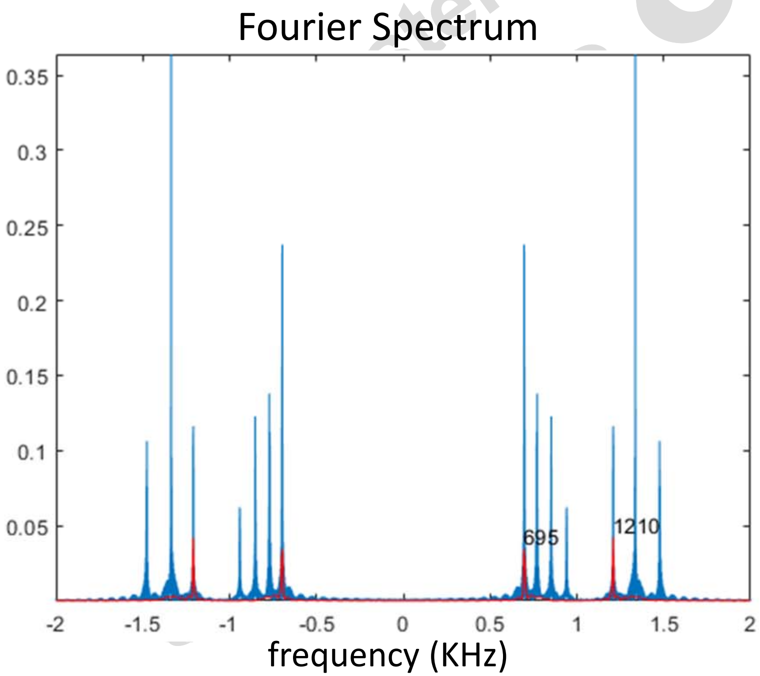



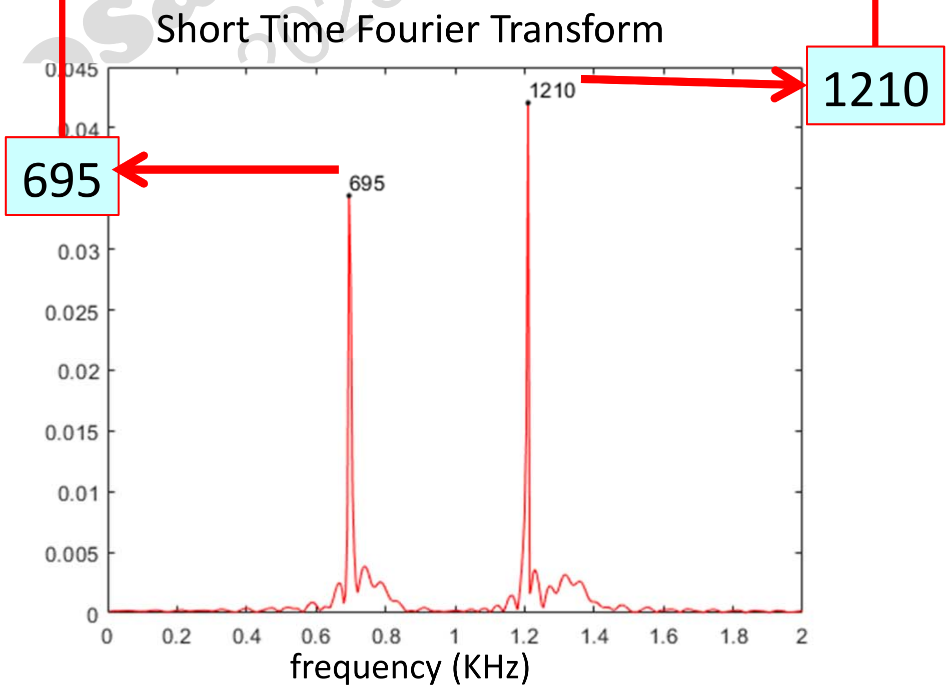
## A moving window should be applied to the signal: "Short Time Fourier Transform (STFT)"

Download **STFT.p** and PhoneNum.zip




| Hz | 1209 | 1336 | 1477 |
|---|---|---|---|
| 697 | 1 | 2 | 3 |
| 770 | 4 | 5 | 6 |
| 852 | 7 | 8 | 9 |
| 941 | * | 0 | # |

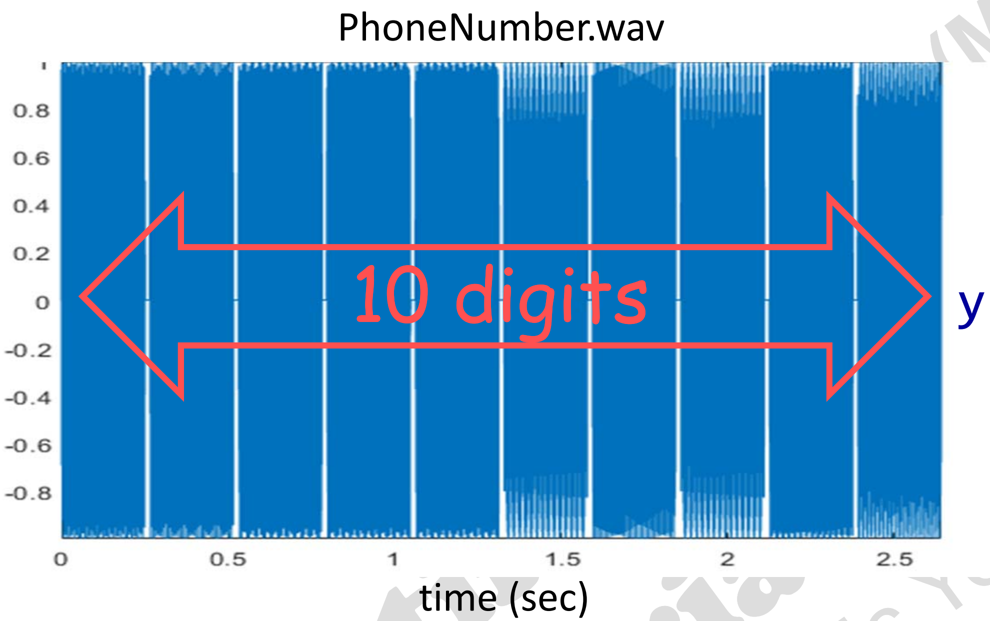


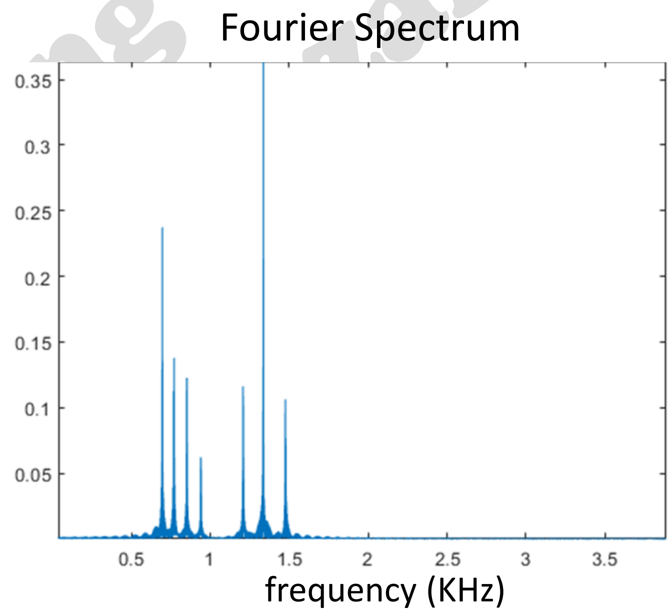



A **moving window** should be applied to the signal:
"Short Time Fourier Transform (STFT)"

Download
**STFT.p**
and PhoneNum.zip

Put the code STFT.p in the same folder as files in PhoneNum.zip




Run: **STFT.p** and answer as follows to the questions

```
Total Duration (sec) = 2.6409
Reduce the window? [y/n]: y
T0: origin of the new window (0 <= t0 < 2.6409) = 0
Width T of the new window = numel(y)/10*Dt
```

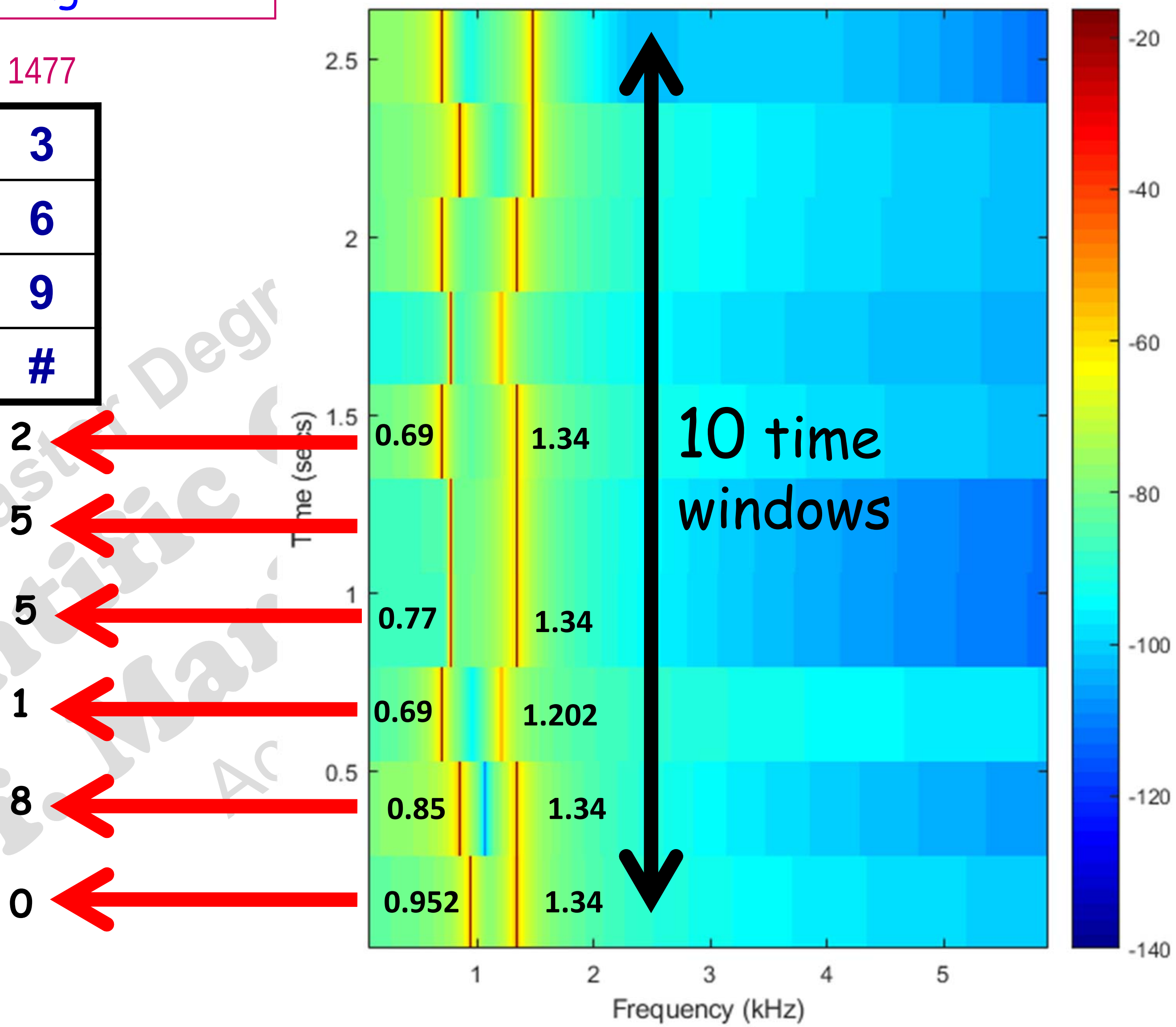
← the best answer

# STFT in MATLAB

```
[y,fs]=audioread('PhoneNumber.wav');
M=fix(numel(y)/10);
spectrogram(y,M,0,[ ],fs); colormap('jet')
```

in Signal Processing Toolbox

| Hz | 1209 | 1336 | 1477 |
|---|---|---|---|
| 697 | 1 | 2 | 3 |
| 770 | 4 | 5 | 6 |
| 852 | 7 | 8 | 9 |
| 941 | * | 0 | # |



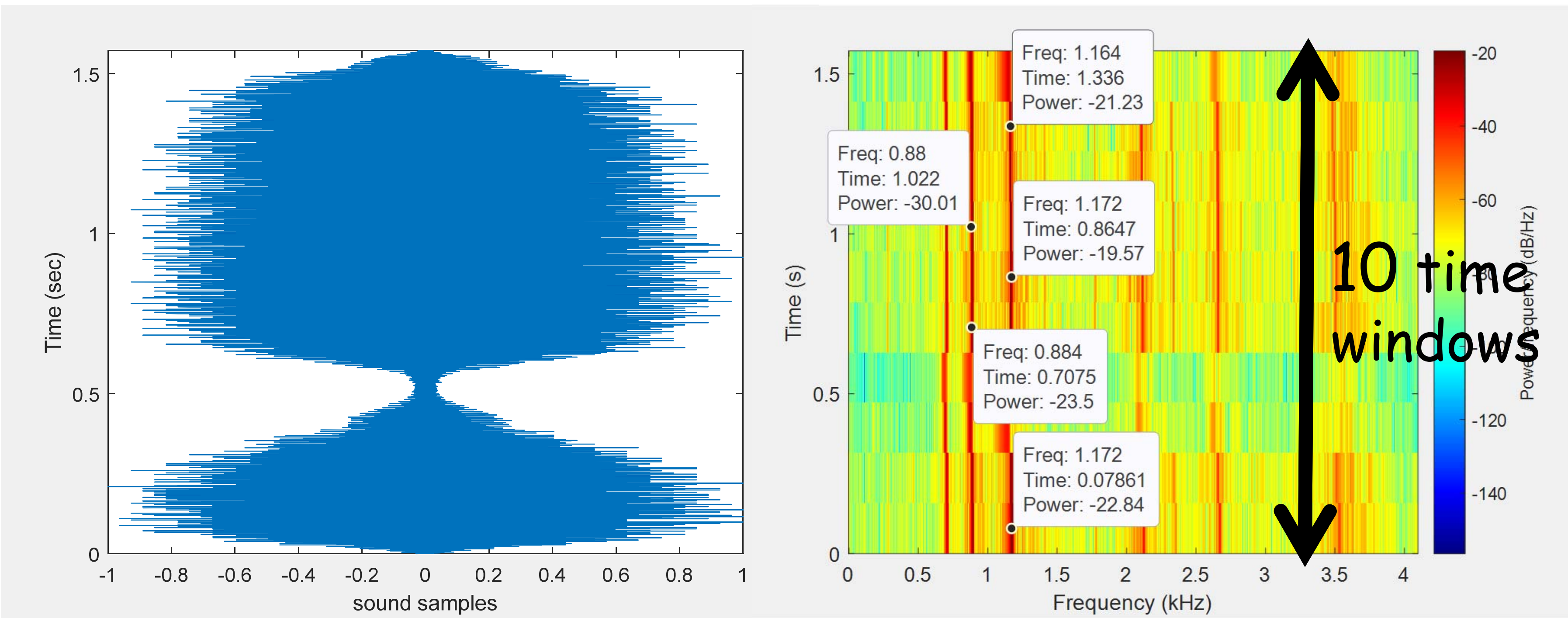



(prof. M. Rizzardi)

# STFT in MATLAB

```
load train; sound(y,Fs) % [y: samples, Fs: sample rate]
Dt=1/Fs  % Period = 1/circ.frequency
Duration=Dt*numel(y); tj=linspace(0,Duration,numel(y))';
plot(y,tj); axis tight; ylabel('time (sec)')
M=fix(numel(y)/10); spectrogram(y,M,0,[ ],Fs); colormap('jet')
```

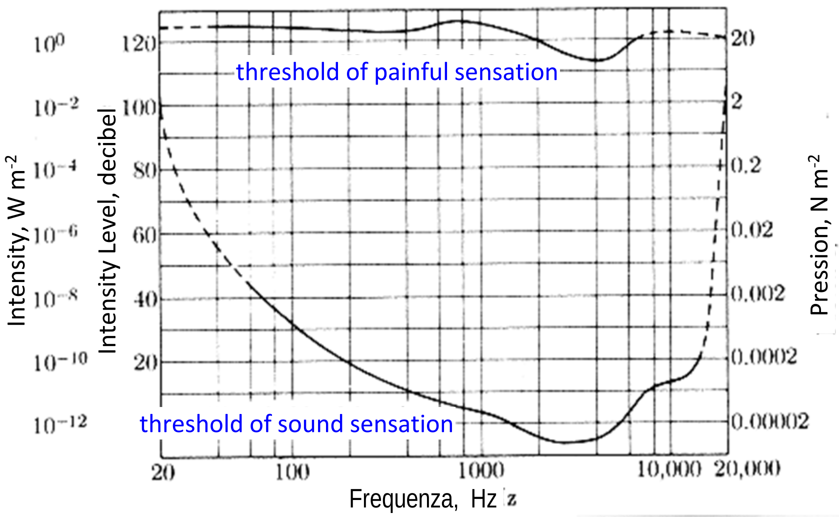
Compare with frequency table in the file: **`Musical_Note_Frequencies.pdf`**



**Example 4**

# Application to music

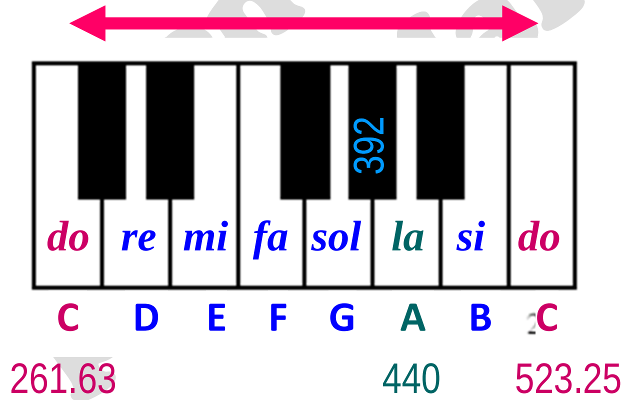
The sensitivity of human ears is such that for each frequency there is a minimal intensity, or threshold of sound sensation, below which sound is inaudible, and a maximal intensity, or threshold of painful sensation, above which sound produces discomfort or pain.




"The sensation of **musical sound** is given by rapid, periodic vibrations; that of **noise** by non-periodic vibrations"

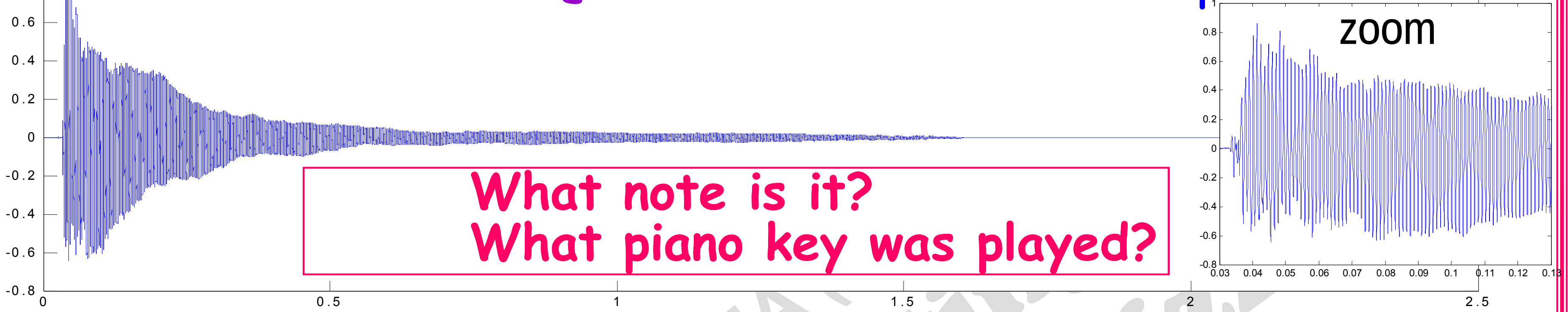
*Sensations of Tone,* Hermann von Helmholtz (1821-1894)
(German physician)

**octave**



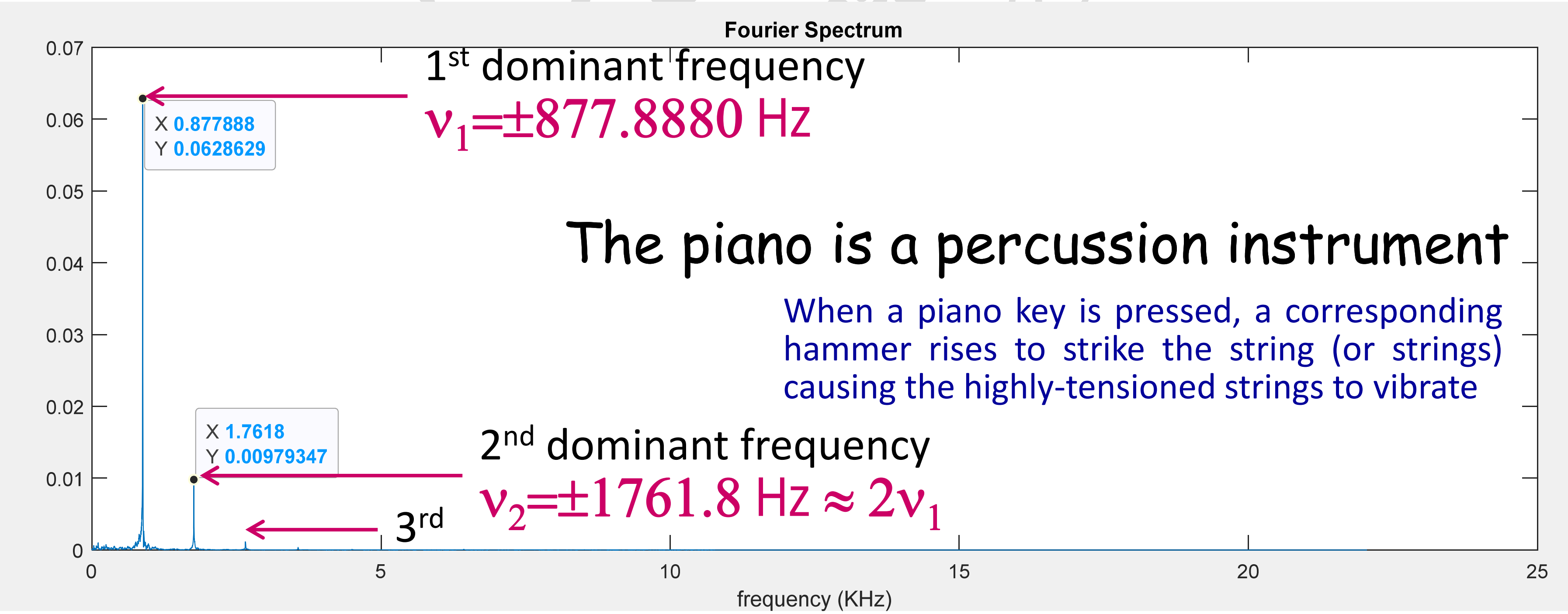

| sound | frequency |
|---|---|
| lowest note of a piano | 27.187 |
| higher note of a piano | 4180 |
| Middle C of the piano | 261.63 |
| A above middle C | 440 |

# *MATLAB* **Example**: reconstruction of a pure sound




**What note is it?**
**What piano key was played?**

```
[y,fs]=audioread('PIANO.wav'); sound(y,fs)
N=numel(y); Dt=1/fs; tj=Dt*(1:N)'; T=N*Dt; Dnu=1/T;
figure(1); plot(tj,y); xlabel('Time (sec)')
Y=fftshift(fft(y));   Y=[Y;Y(1)]*T/N;   nu=(-N/2:N/2)'/T;
figure(2); plot(1e-3*nu,abs(Y)) % KHz
xlabel('frequency (KHz)'); title('Fourier Spectrum')
```




1st dominant frequency

$\nu_1 = \pm 877.8880$ Hz

2nd dominant frequency

$\nu_2 = \pm 1761.8$ Hz $\approx 2\nu_1$

3rd

## The piano is a percussion instrument

When a piano key is pressed, a corresponding hammer rises to strike the string (or strings) causing the highly-tensioned strings to vibrate

# frequencies and musical notes

| | | |
|---|---|---|
| DO5 | C5 | 523 Hz |
| DO#5 | C#5 | 554 Hz |
| RE5 | D5 | 587 Hz |
| RE#5 | D#5 | 622 Hz |
| MI5 | E5 | 659 Hz |
| FA5 | F5 | 698 Hz |
| FA#5 | F#5 | 740 Hz |
| SOL5 | G5 | 784 Hz |
| SOL#5 | G#5 | 831 Hz |
| LA5 | A5 | 880 Hz |
| LA#5 | A#5 | 932 Hz |
| SI5 | B5 | 988 Hz |
| | | |
| DO6 | C6 | 1046 Hz |
| DO#6 | C#6 | 1109 Hz |
| RE6 | D6 | 1175 Hz |
| RE#6 | D#6 | 1245 Hz |
| MI6 | E6 | 1319 Hz |
| FA6 | F6 | 1397 Hz |
| FA#6 | F#6 | 1480 Hz |
| SOL6 | G6 | 1568 Hz |
| SOL#6 | G#6 | 1661 Hz |
| LA6 | A6 | 1760 Hz |
| LA#6 | A#6 | 1865 Hz |
| SI6 | B6 | 1976 Hz |

$\nu$=877.8880 Hz

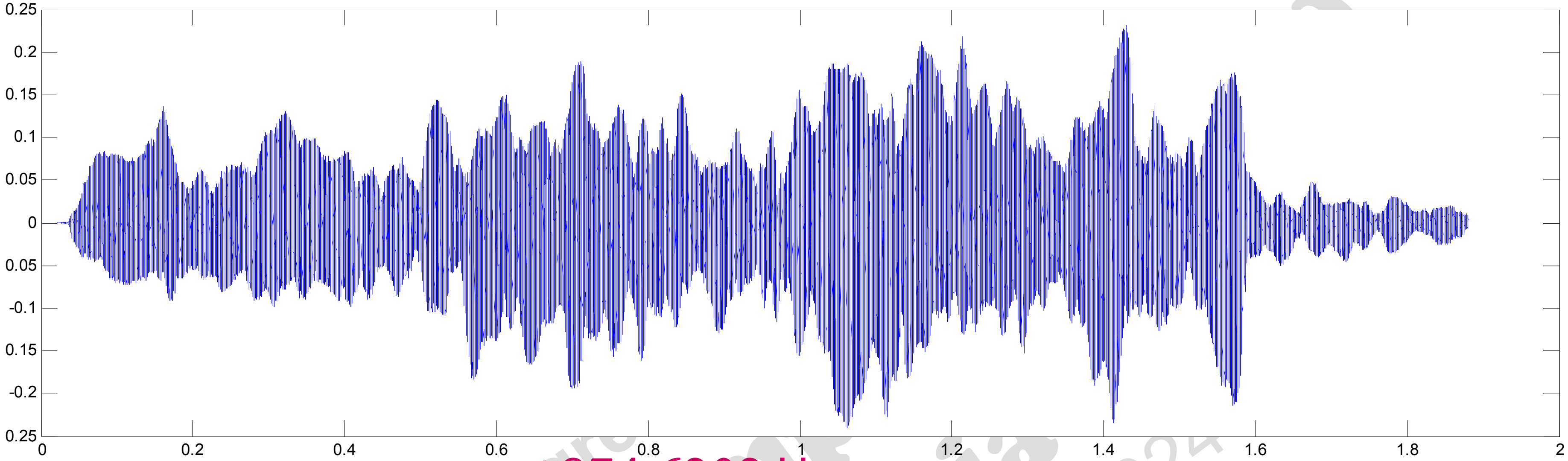
$\approx$ 880 Hz

1 octave

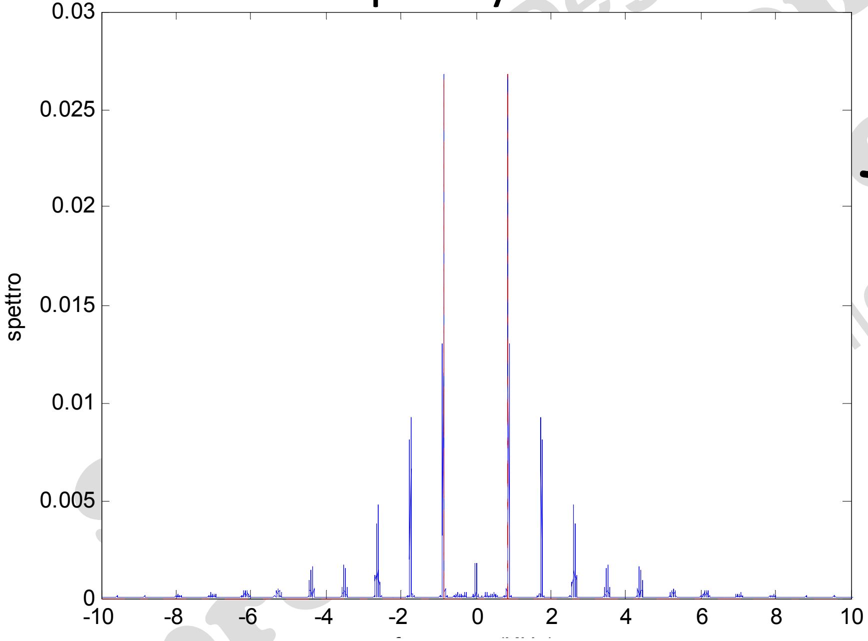
$\approx 1760 = 2\times880$

$\nu$=1761 Hz

# If we repeat the experiment with a violin … (file "VIOLIN.wav")



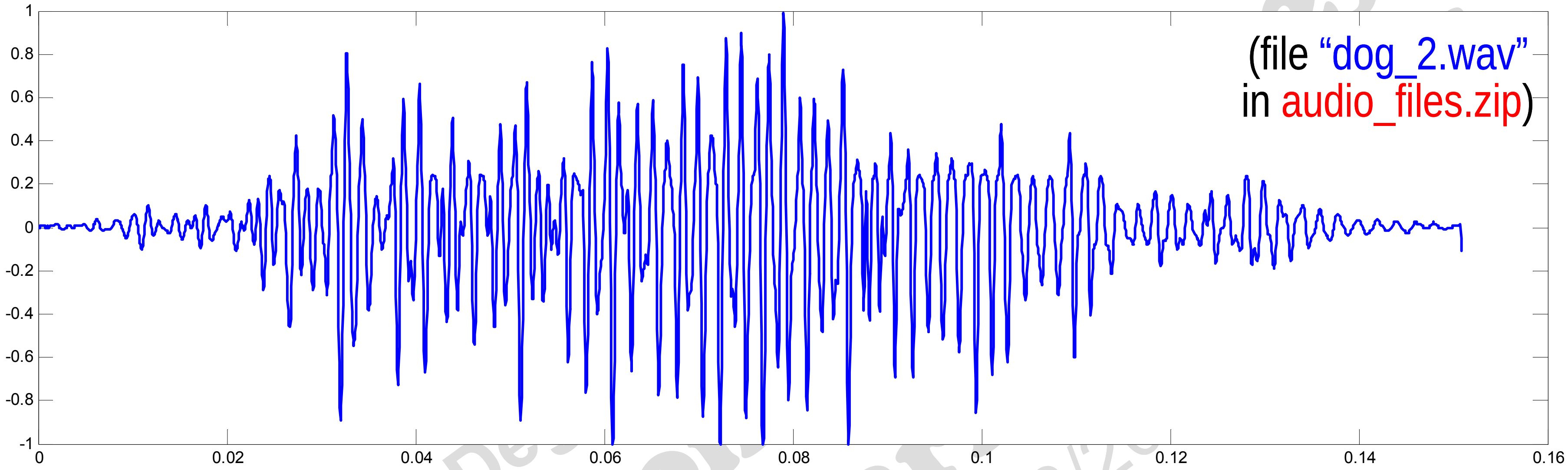

dominant frequency $\nu = \pm 874.6202$ Hz

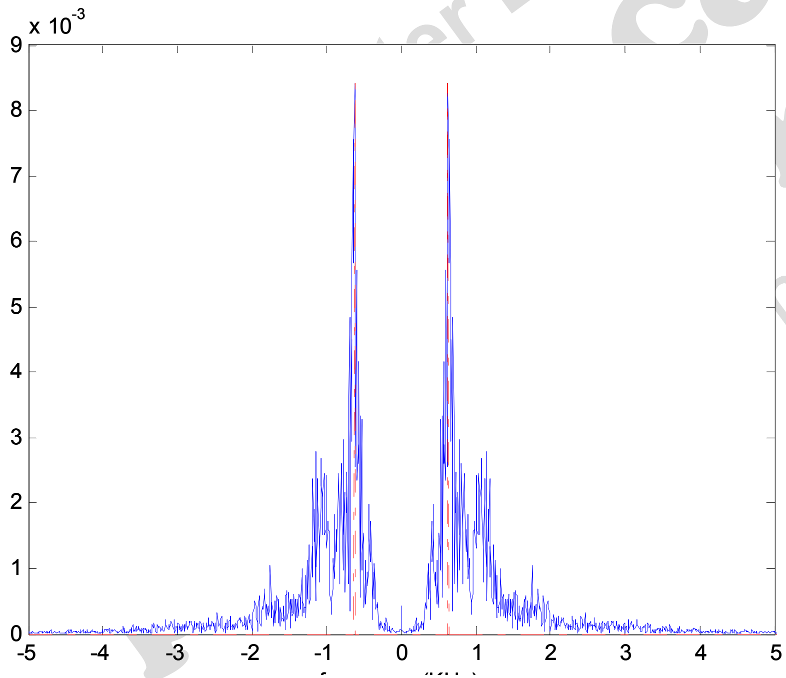



The violin is a string instrument

a given note on a violin will have several frequencies vibrating at once, since the violin bow touches the chords

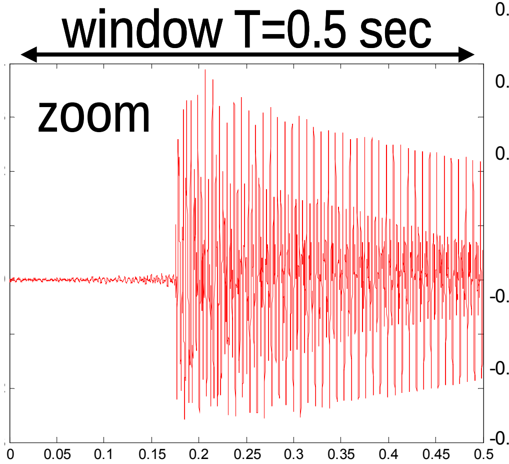
# If we repeat the experiment with a non-periodic sound (noise)

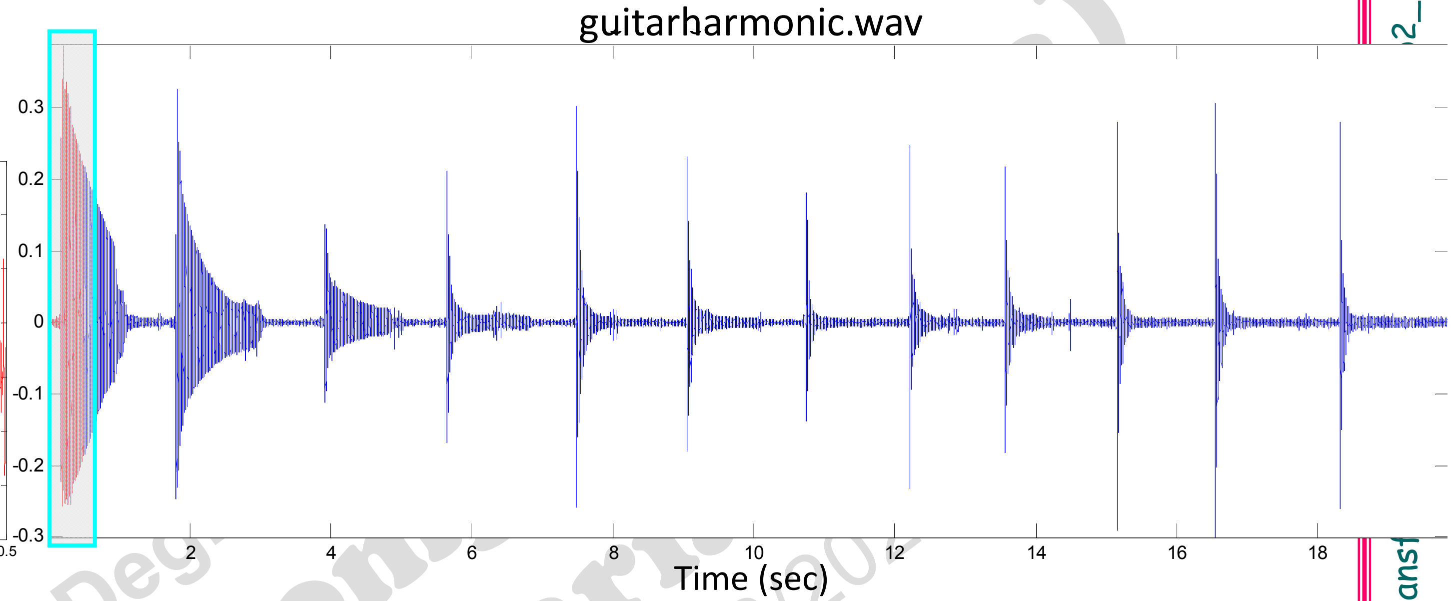


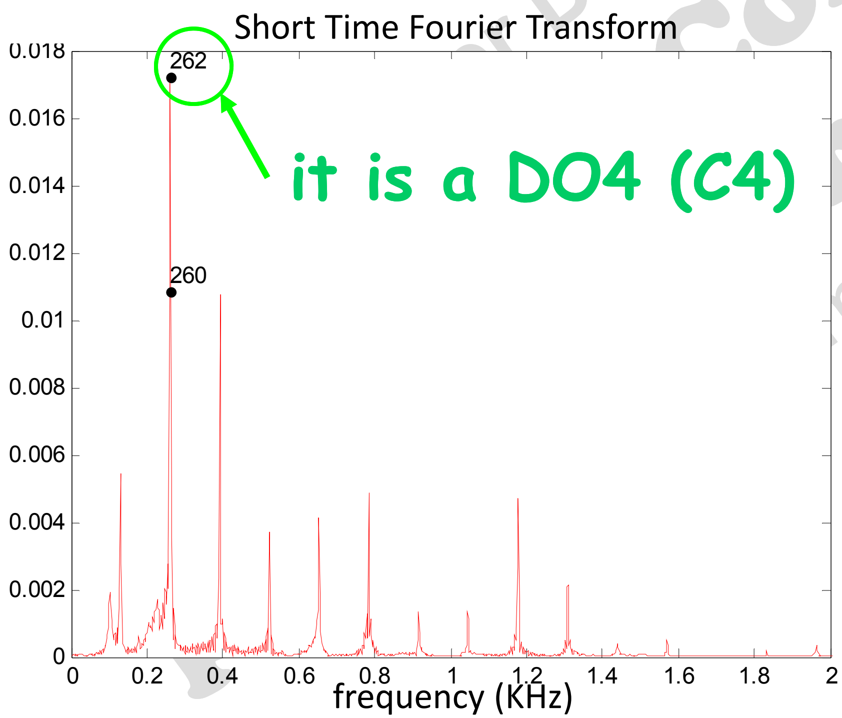



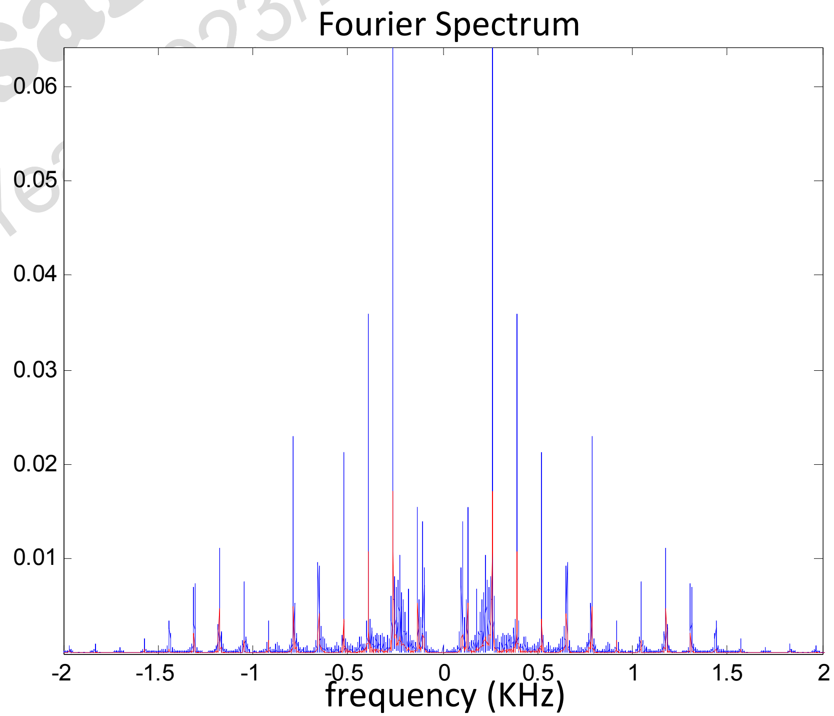
infinitely many frequencies
for a non-periodic sound

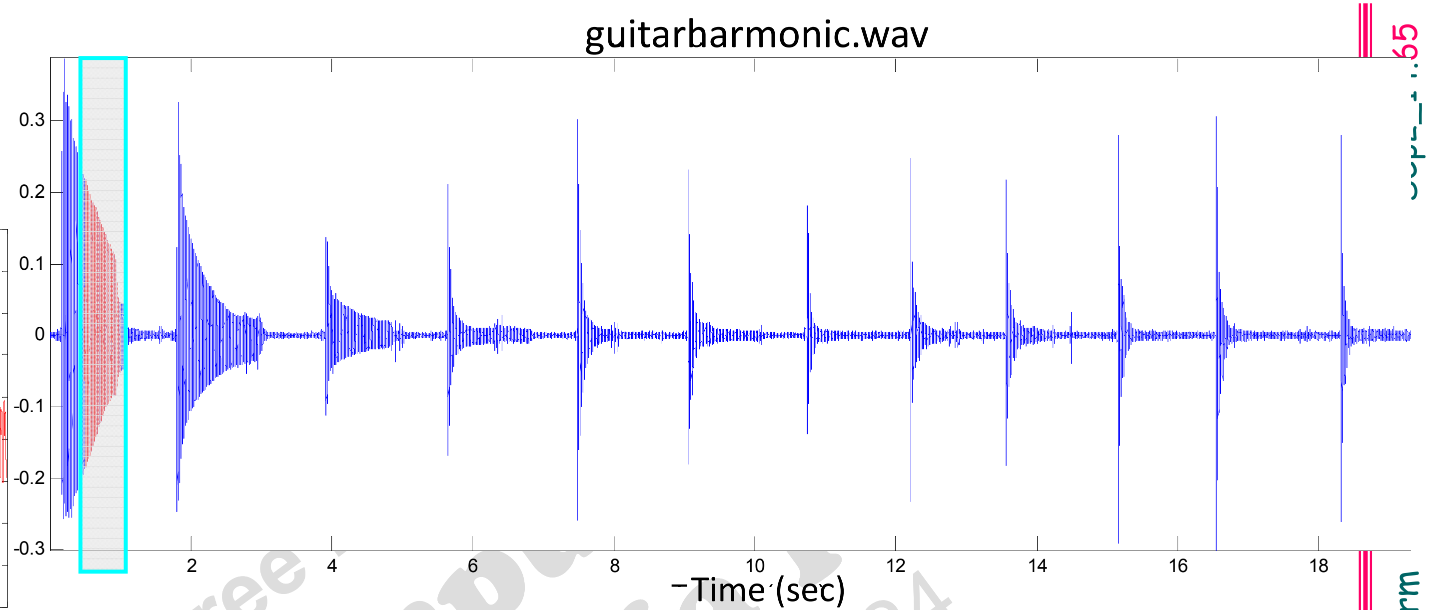
# by means of Short Time Fourier Transform ...

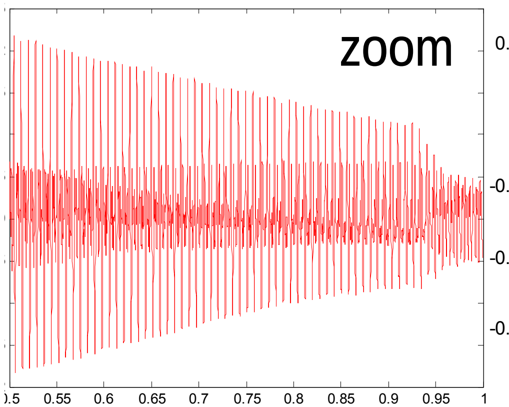


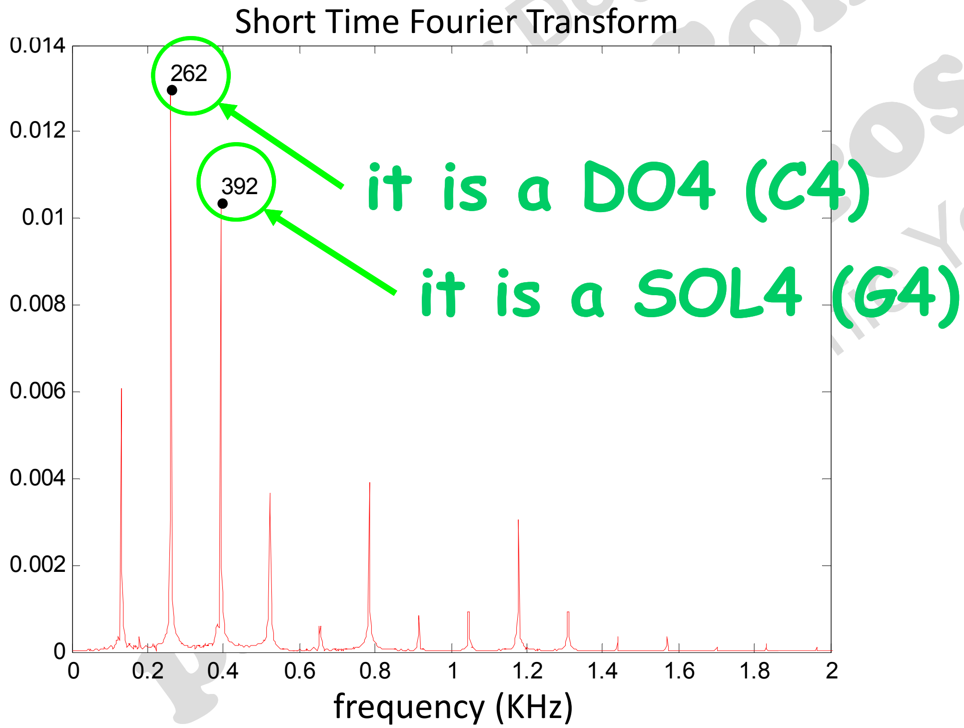



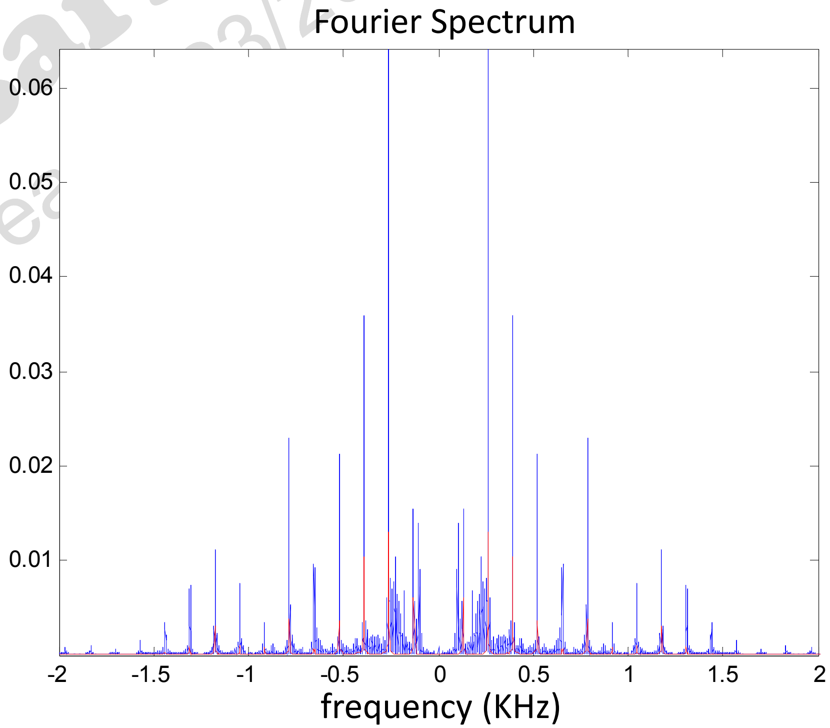




guitarbarmonic.wav
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
2 4 6 8 10 12 14 16 18
Time (sec)


zoom
0.55 0.6 0.65 0.7 0.75 0.8 0.85 0.9 0.95 1


Short Time Fourier Transform
262
392
it is a DO4 (C4)
it is a SOL4 (G4)
0.014
0.012
0.01
0.008
0.006
0.004
0.002
0
0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2
frequency (KHz)


Fourier Spectrum
0.06
0.05
0.04
0.03
0.02
0.01
-2 -1.5 -1 -0.5 0 0.5 1 1.5 2
frequency (KHz)