Modelação Numérica 2017

Aula 6, 7/Mar

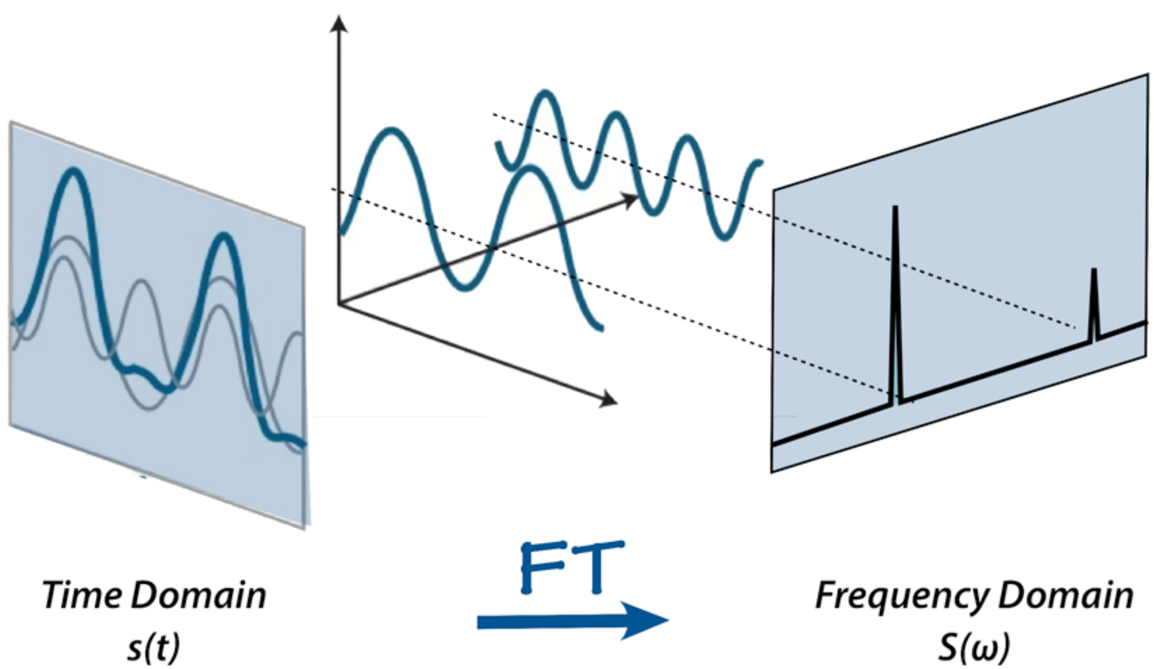
- Propriedades da DFT
- FFT
- Convolução
- Correlação

http://modnum.ucs.ciencias.ulisboa.pt

Transformada de Fourier Discreta

- Qualquer função periódica pode ser representada como uma série de Fourier, i.e.:

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} \left(a_k \cos \frac{2\pi kt}{T} + b_k \sin \frac{2\pi kt}{T} \right)$$



Aula passada

Transformada de Fourier Discreta

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty}\left(a_k \cos\frac{2\pi kt}{T} + b_k \sin\frac{2\pi kt}{T}\right) =$$

$$= \frac{a_0}{2} + \left(a_1 \cos\frac{2\pi t}{T} + b_1 \sin\frac{2\pi t}{T}\right) + \left(a_2 \cos\frac{2\pi t}{T/2} + b_2 \sin\frac{2\pi t}{T/2}\right) + \ldots$$

Primeira harmónica:

$$a_1 \cos\frac{2\pi t}{T} + b_1 \sin\frac{2\pi t}{T} = A \cos\left(\frac{2\pi t}{T} + \phi\right)$$

- Φ: Fase inicial
- A: Amplitude
- T: Período; Frequência f=1/T; Frequência angular: ω=2πf = 2π/T

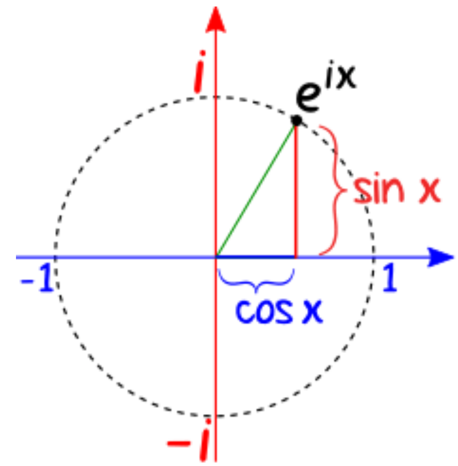
Os coeficientes de Fourier são complexos

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} \left(a_k \cos \frac{2\pi k t}{T} + b_k \sin \frac{2\pi k t}{T} \right) =$$

$$= \frac{a_0}{2} + \sum_{k=1}^{\infty} \frac{a_k - i b_k}{2} e^{i2\pi k t/T} + \sum_{k=1}^{\infty} \frac{a_k + i b_k}{2} e^{i2\pi k t/T} =$$

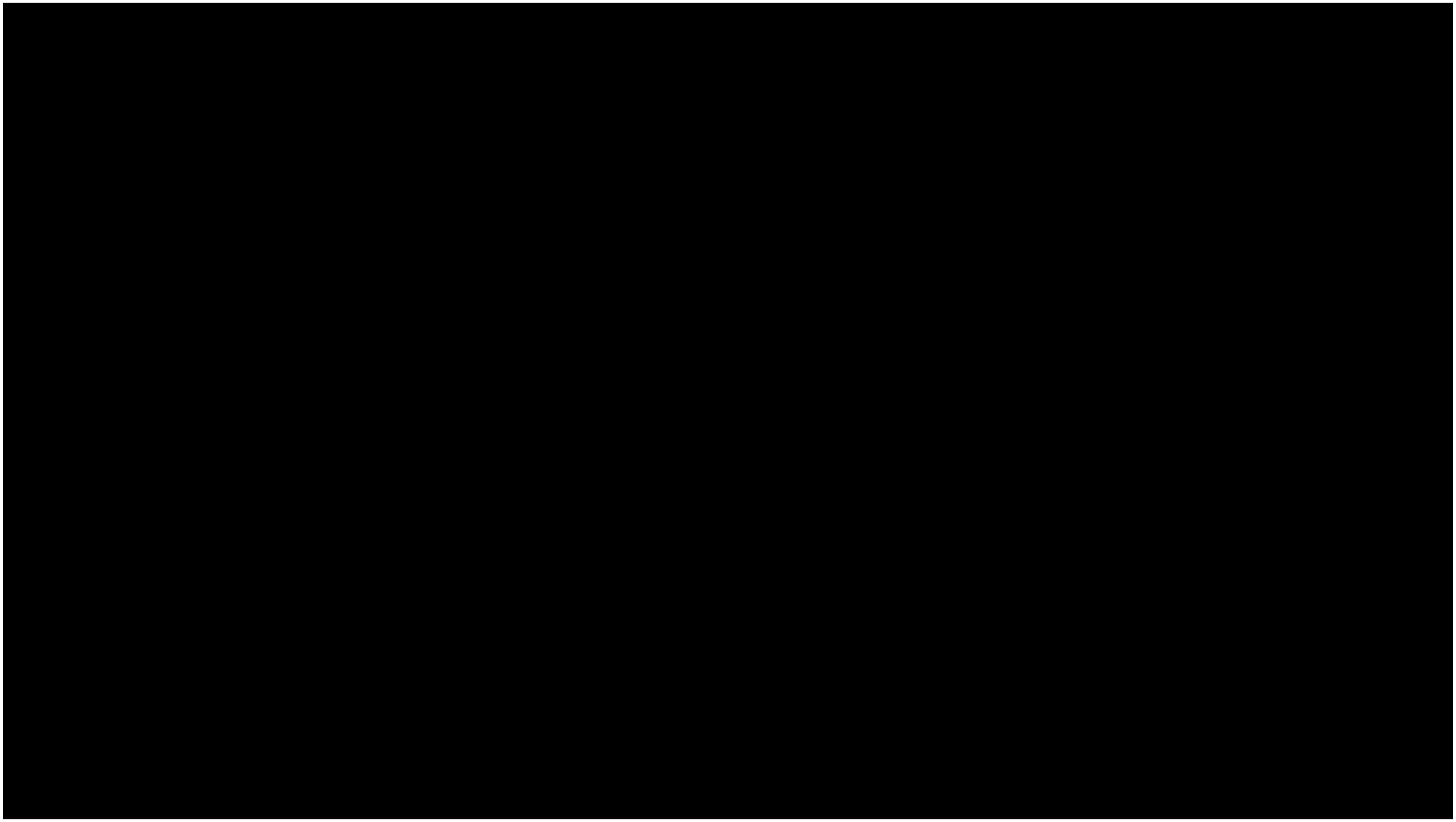
$$= \sum_{k=-\infty}^{\infty} c_k e^{i2\pi k t/T}$$

Exponenciais imaginárias



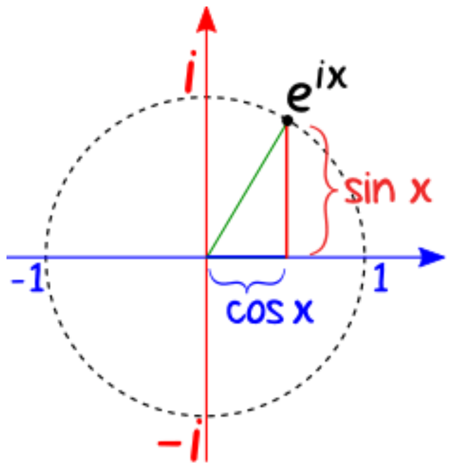
Video: https://www.youtube.com/watch?v=K_C7htSXORY, David Dorran

Exponenciais imaginárias



Video: https://www.youtube.com/watch?v=cUD1gMAl6W4, GLV

Exponenciais imaginárias



Outros vídeos:

- https://www.youtube.com/watch?v=mkGsMWi_j4Q
- https://www.youtube.com/watch?v=DT66OE2JEpU
- https://www.youtube.com/watch?v=GLScx2pWF0A
- https://www.youtube.com/watch?v=k8FXF1KjzY0

Os coeficientes de Fourier são complexos

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} \left(a_k \cos \frac{2\pi kt}{T} + b_k \sin \frac{2\pi kt}{T} \right) =$$

$$= \frac{a_0}{2} + \sum_{k=1}^{\infty} \frac{a_k - ib_k}{2} e^{i2\pi kt/T} + \sum_{k=1}^{\infty} \frac{a_k + ib_k}{2} e^{i2\pi kt/T} =$$

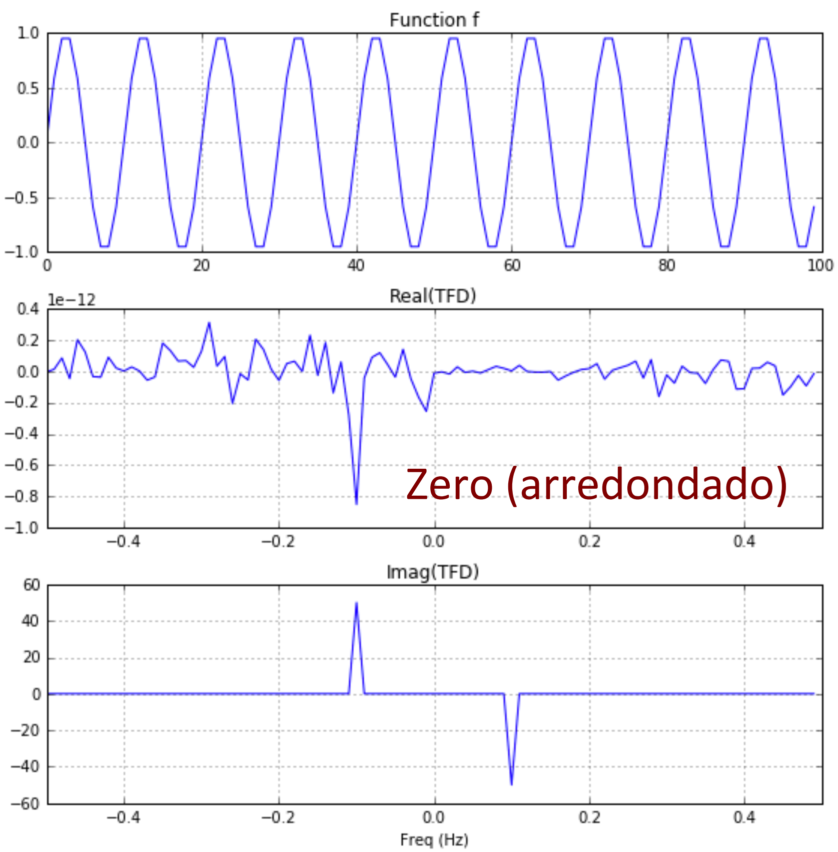
$$= \sum_{k=-\infty}^{\infty} c_k e^{i2\pi kt/T}$$

$$c_0 = \frac{a_0}{2},\ c_n = \frac{a_n - ib_n}{2},\ c_{-n} = \frac{a_n + ib_n}{2}$$

Aula passada

Funções reais: Pares vs Ímpares

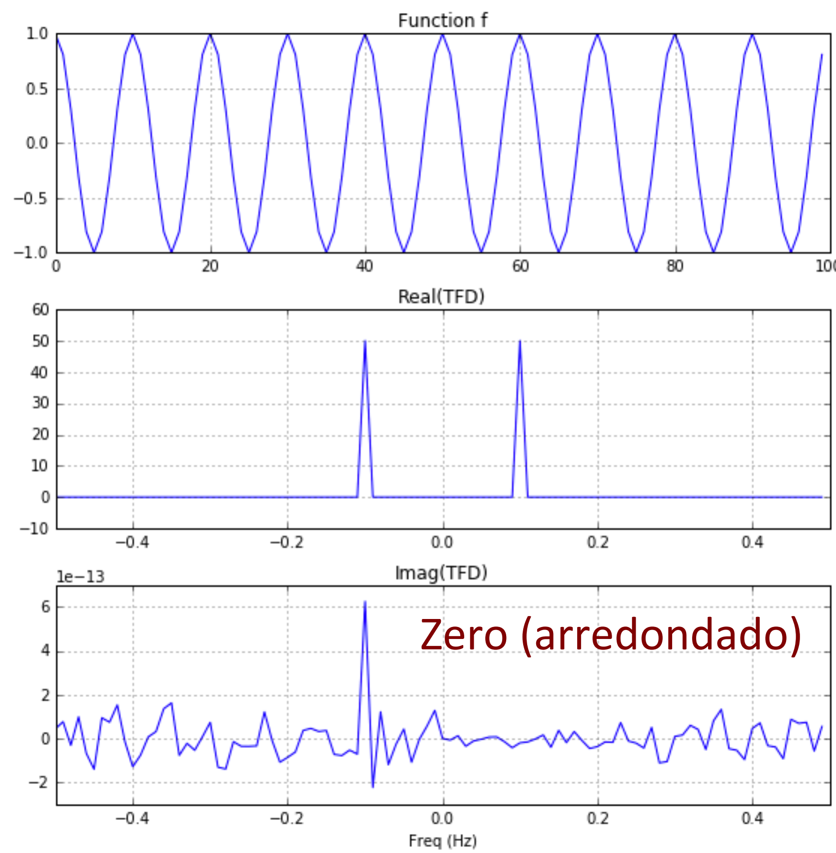
Ímpares: sin(x) = - sin(-x)



Transformada imaginária, anti-simétrica:

$c_k = ir_k = -ir_{-k}$

Pares: cos(x) = cos(-x)



Transformada real, simétrica:

$c_k = r_k = r_{-k}$

Espectro de amplitude

$$c_k = \sum_{t=0}^{N-1} f(t_n) e^{-i2\pi t_n k/N}$$

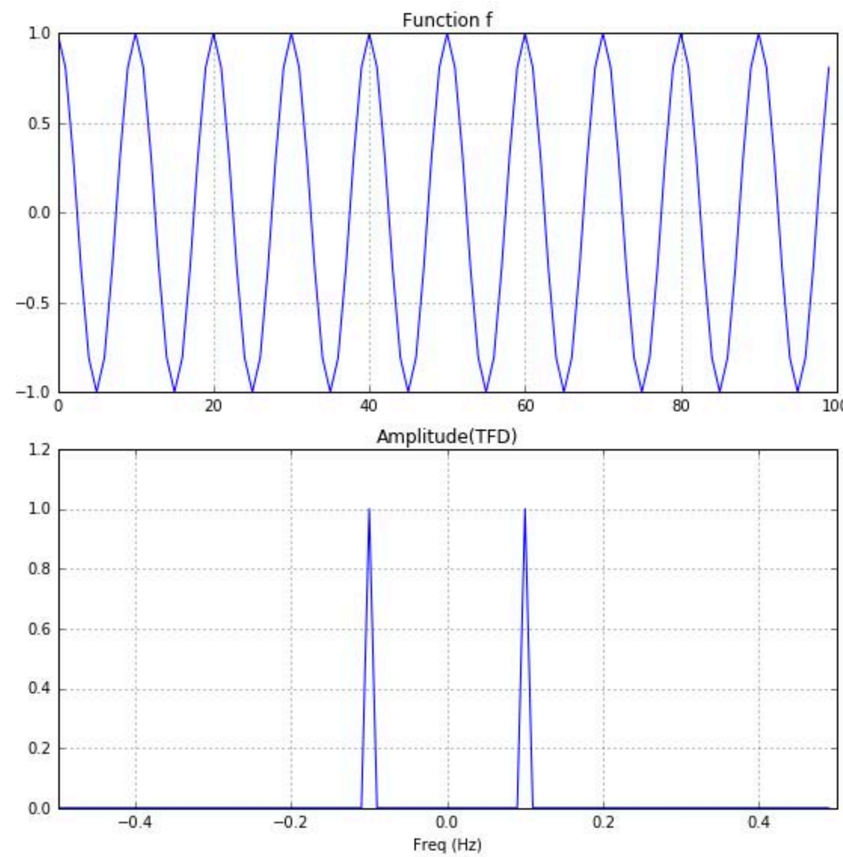
```
import matplotlib.pyplot as plt
import numpy as np
from numpy import pi as pi

#%%
def TFD(f):
    n=len(f);
    c=np.zeros(len(f), dtype=complex)
    for k in range(n):
        for t in range(n):
            c[k] = c[k] + f[t]*np.exp(-2j*pi*t*k/n)
    return c


N=100.; T=10.; dt=1.
t=np.arange(0.,N,dt)
f=np.cos(2*pi * t/T)

tf=TFD(f)
fout=np.concatenate([tf[N/2:], tf[:N/2]])
fNyq=1/(2*dt); df=1/(N*dt);
freq=np.arange(-fNyq, fNyq,df)

plt.close();
plt.subplot(2,1,1); plt.plot(t, f)
plt.title('Function f'); plt.grid();

plt.subplot(2,1,2); plt.plot(freq,np.abs(fout)/(N/2))
plt.title('Amplitude(TFD)'); plt.grid();
plt.xlim([-fNyq, fNyq])
plt.xlabel('Freq (Hz)')
plt.tight_layout()
```

Reordenamento

Vector de frequências

Escalamento

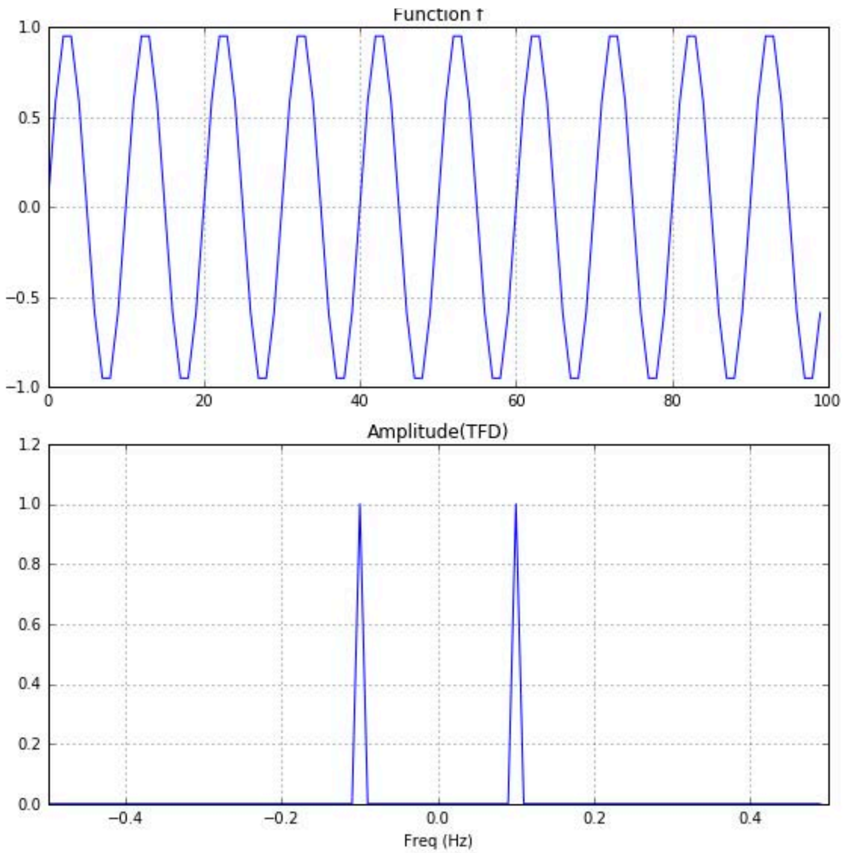
cos(x)



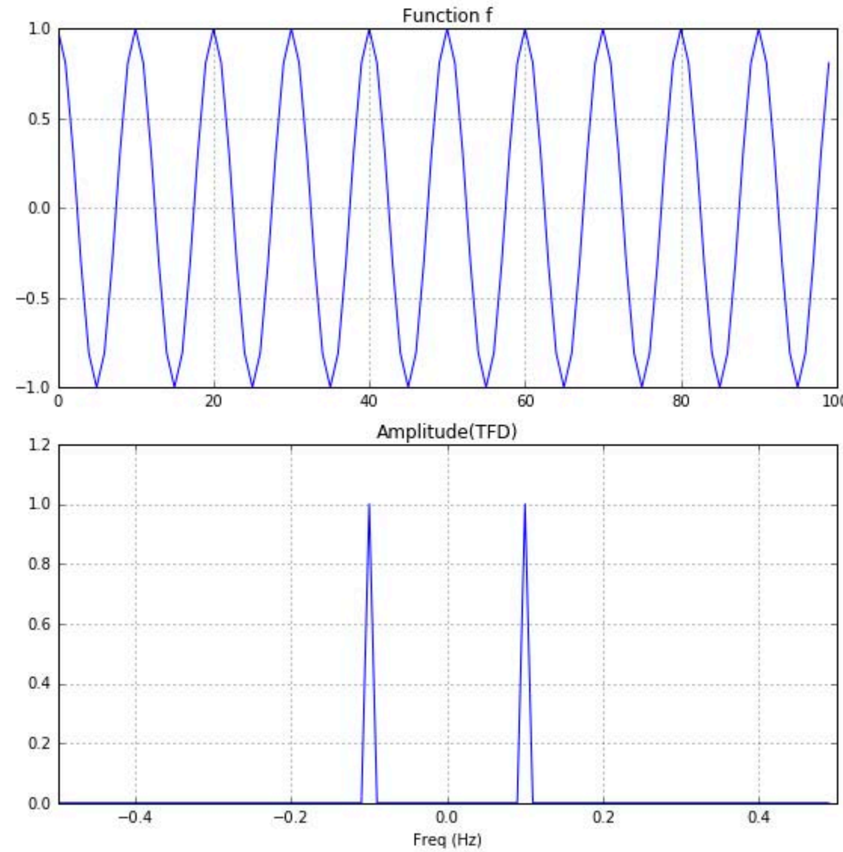
Aula passada

Espectro de amplitude

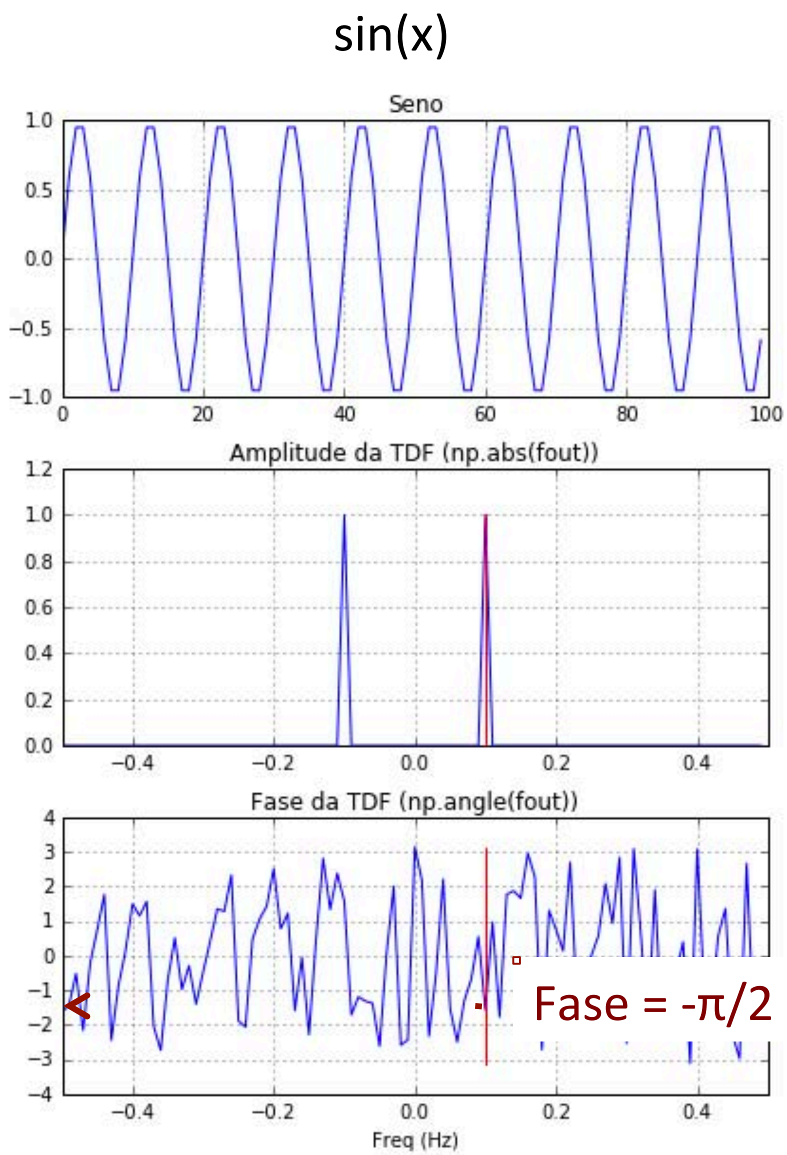
sin(x)

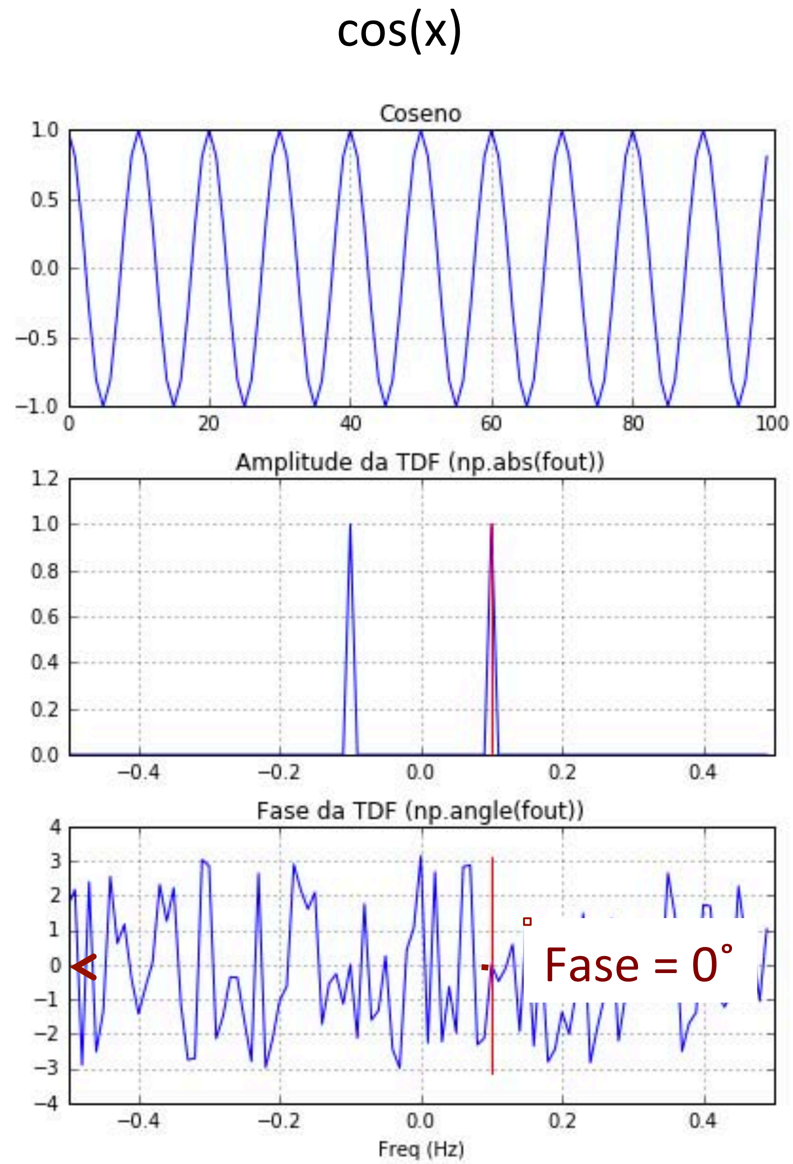


cos(x)



Espectro de amplitude e de fase





Propriedades da TDF

- Linearidade:

$$G = \mathcal{F}(g);\ H = \mathcal{F}(h)$$

$$\Downarrow$$

$$\mathcal{F}(ag+bh) = aG + bH$$

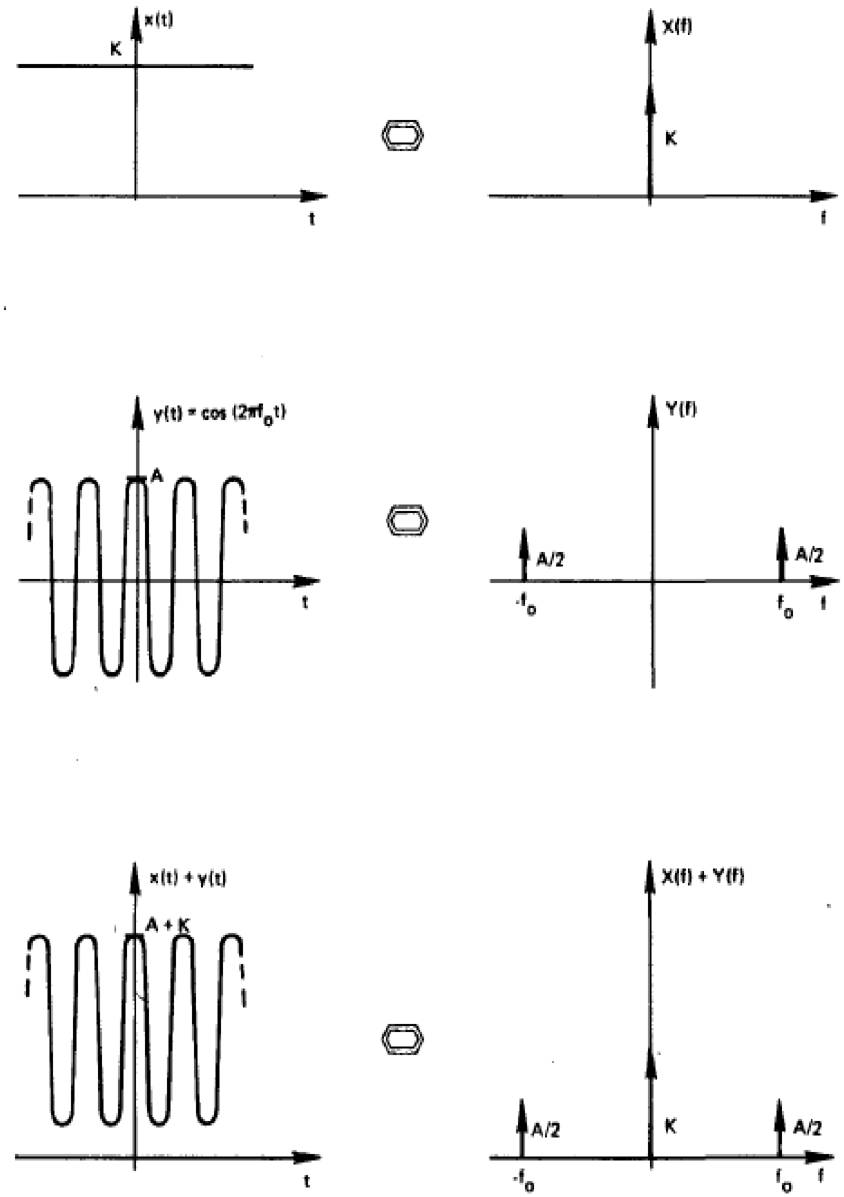


Figure 3-1. The linearity property.

Propriedades da TDF

- Translação:

$$G(f) = \mathcal{F}(g(t))$$

$$\Downarrow$$

$$\mathcal{F}(g(t-a)) = e^{-ifa}G(f)$$

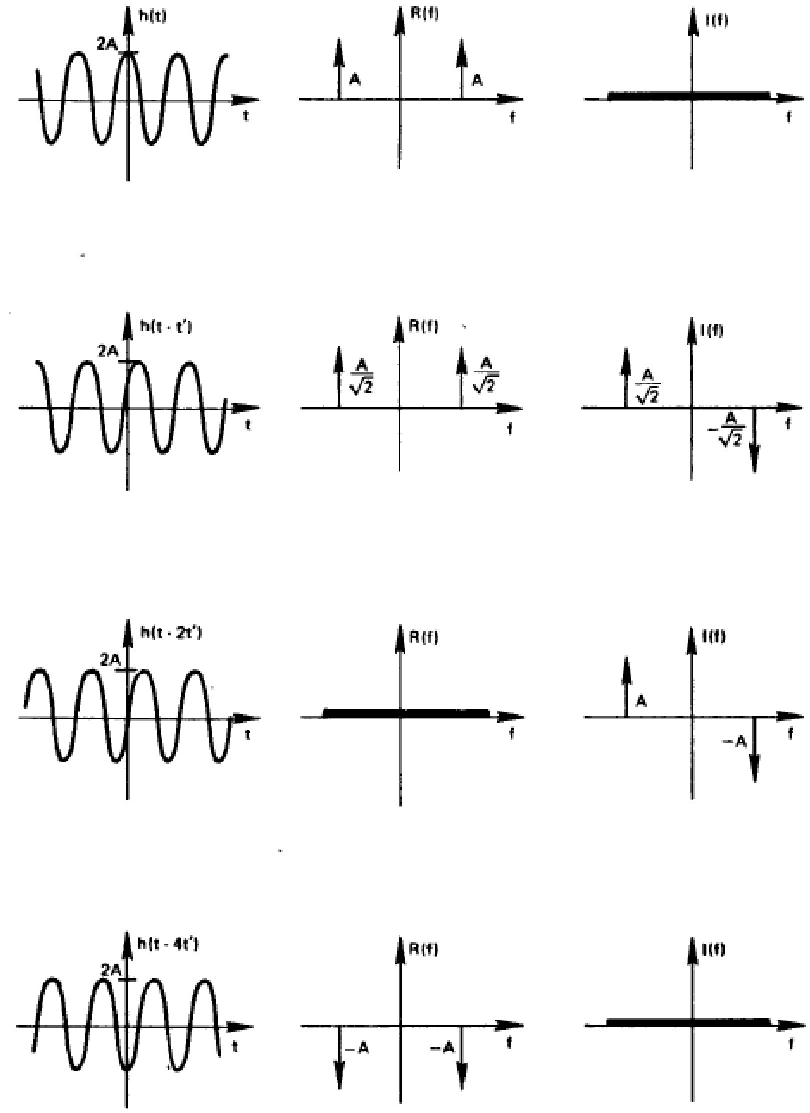


Figure 3-4. Time shifting property.

Propriedades da TDF

- Escalamento:

$$G(f) = \mathcal{F}(g(t))$$

$$\Downarrow$$

$$\mathcal{F}(g(at)) = 1/a\, G(f/a)$$

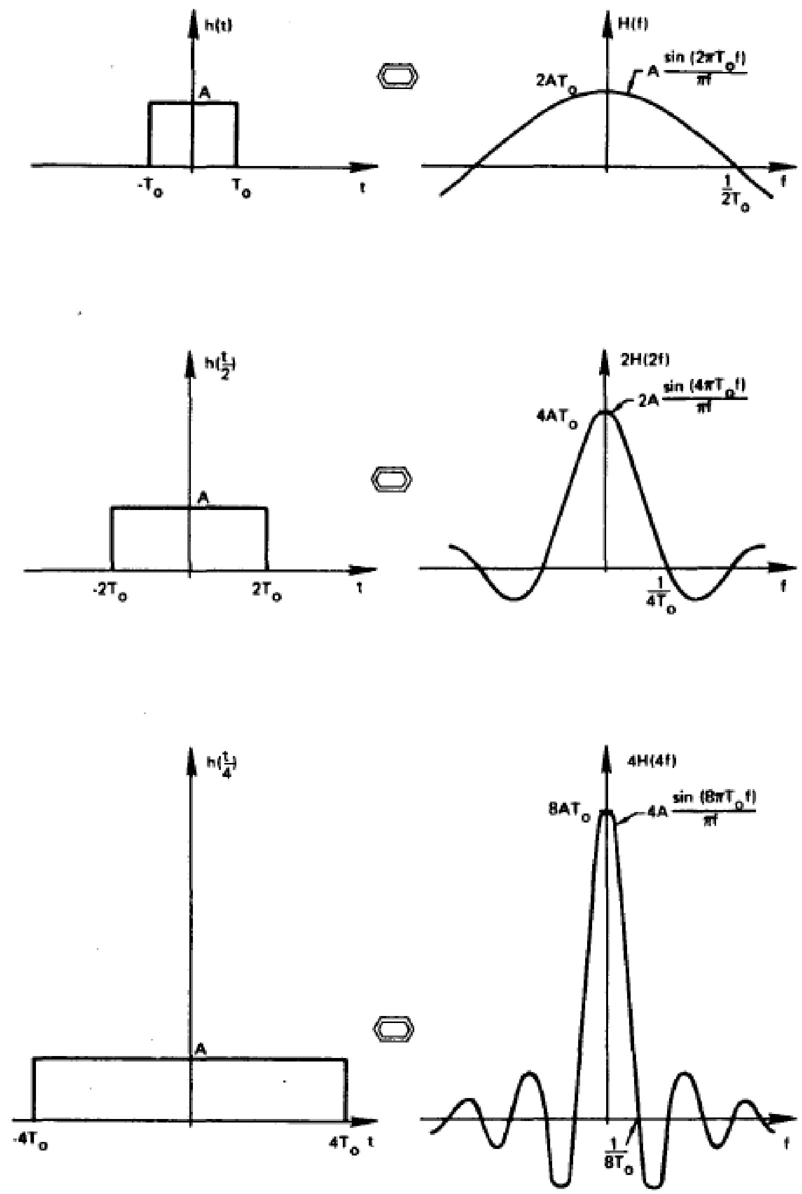
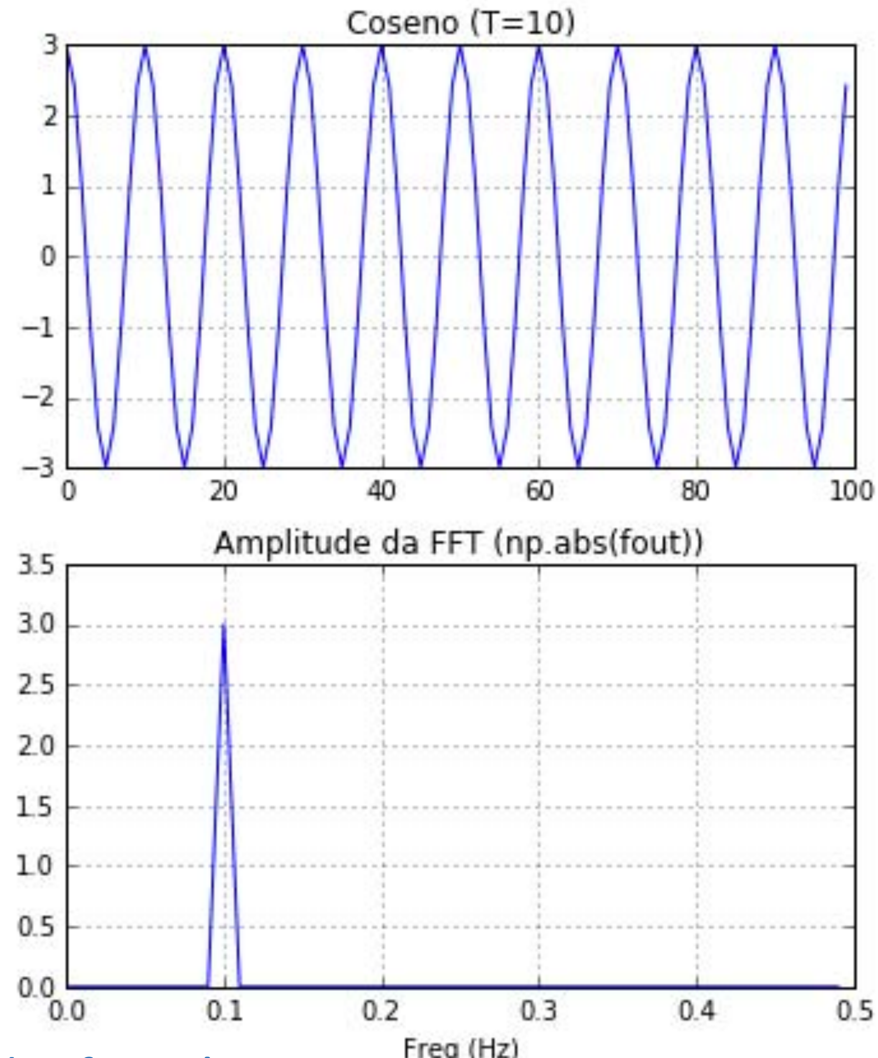


Figure 3-2. Time scaling property.

FFT (iFFT): Fast Fourier Transform

$$c_k = \sum_{t=0}^{N-1} f(t_n) e^{-i2\pi t_n k/N}$$

```
import matplotlib.pyplot as plt
import numpy as np
from numpy import pi as pi
import numpy.fft as fft

plt.rcParams['figure.figsize'] = 5, 6

#%%
N=100.; T=10.; dt=1.
t=np.arange(0.,N,dt)
f=3*np.cos(2*pi * t/T)

tf=fft.fft(f)
fout=tf[:N/2]
fNyq=1/(2*dt); df=1/(N*dt);
freq=np.arange(0, fNyq,df)

plt.close();
plt.subplot(2,1,1); plt.plot(t, f)
plt.title('Coseno (T=10)'); plt.grid();

plt.subplot(2,1,2); plt.plot(freq,np.abs(fout)/(N/2))
plt.title('Amplitude da FFT (np.abs(fout))'); plt.grid();
plt.xlabel('Freq (Hz)')

plt.tight_layout()
```

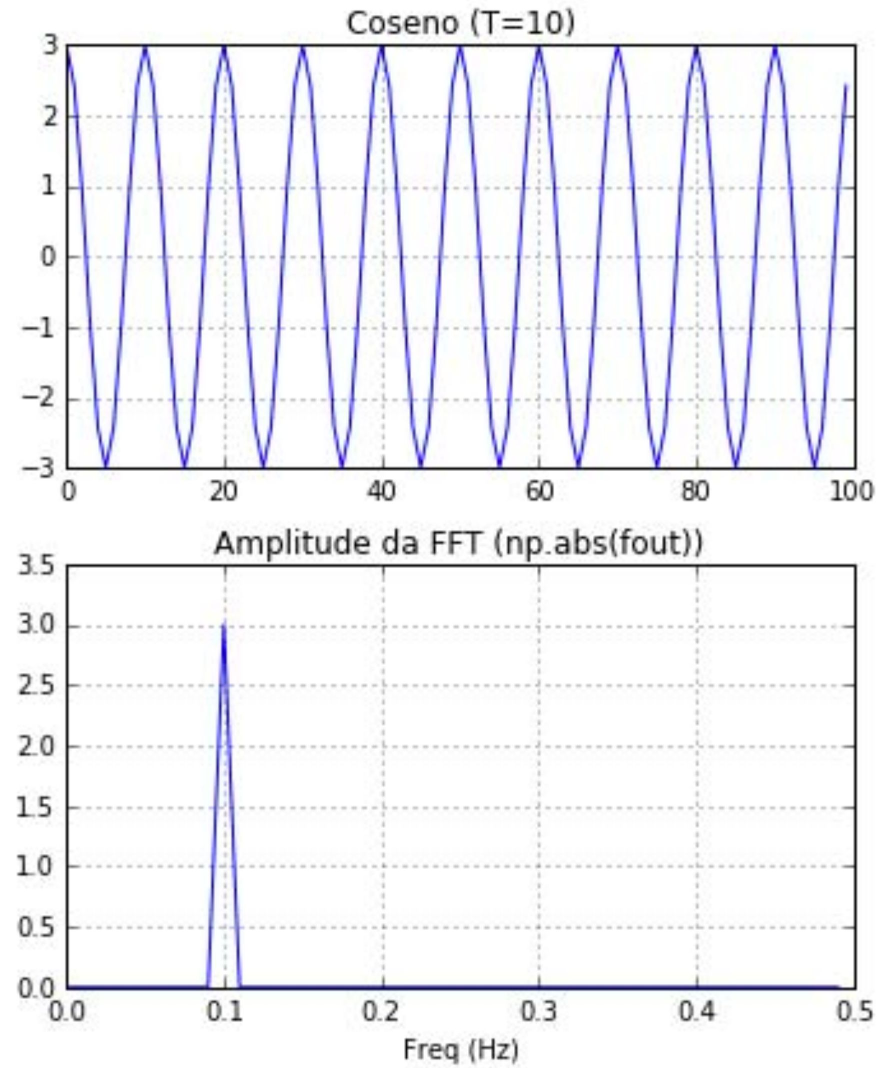


Dada a simetria do espectro de amplitude,
é suficiente representar o semi-eixo positivo das frequências.

FFT (iFFT): Fast Fourier Transform

$$c_k = \sum_{t=0}^{N-1} f(t_n) e^{-i2\pi t_n k/N}$$

```
import matplotlib.pyplot as plt
import numpy as np
from numpy import pi as pi
import numpy.fft as fft

plt.rcParams['figure.figsize'] = 5, 6

#%%
N=100.; T=10.; dt=1.
t=np.arange(0.,N,dt)
f=3*np.cos(2*pi * t/T)

tf=fft.fft(f)
fout=tf[:N/2]
fNyq=1/(2*dt); df=1/(N*dt);
freq=np.arange(0, fNyq,df)

plt.close();
plt.subplot(2,1,1); plt.plot(t, f)
plt.title('Coseno (T=10)'); plt.grid();

plt.subplot(2,1,2); plt.plot(freq,np.abs(fout)/(N/2))
plt.title('Amplitude da FFT (np.abs(fout))'); plt.grid();
plt.xlabel('Freq (Hz)')

plt.tight_layout()
```



A FFT será eficiente se $N=2^k$.

Transformada de Fourier

Domínio físico:

- Tempo (s)
- Espaço (m)

$\mathcal{F}$

Directa >

< Inversa

$\mathcal{F}^{-1}$

Domínio transformado (espectral):

- Frequência (s^{-1} ou Hz)
- Número de onda (m^{-1})

Série temporal:

- Δt (intervalo de amostragem)
- N Δt (dimensão/comprimento da amostra)

Espectro (amplitude, fase):

- $\Delta f = 1/(N\ \Delta t)$ (resolução espectral)
- $f_{Nyquist} = 1/(2\ \Delta t)$ (frequência máxima)

Mesma informação!

Interacções entre duas séries?

- Convolução:

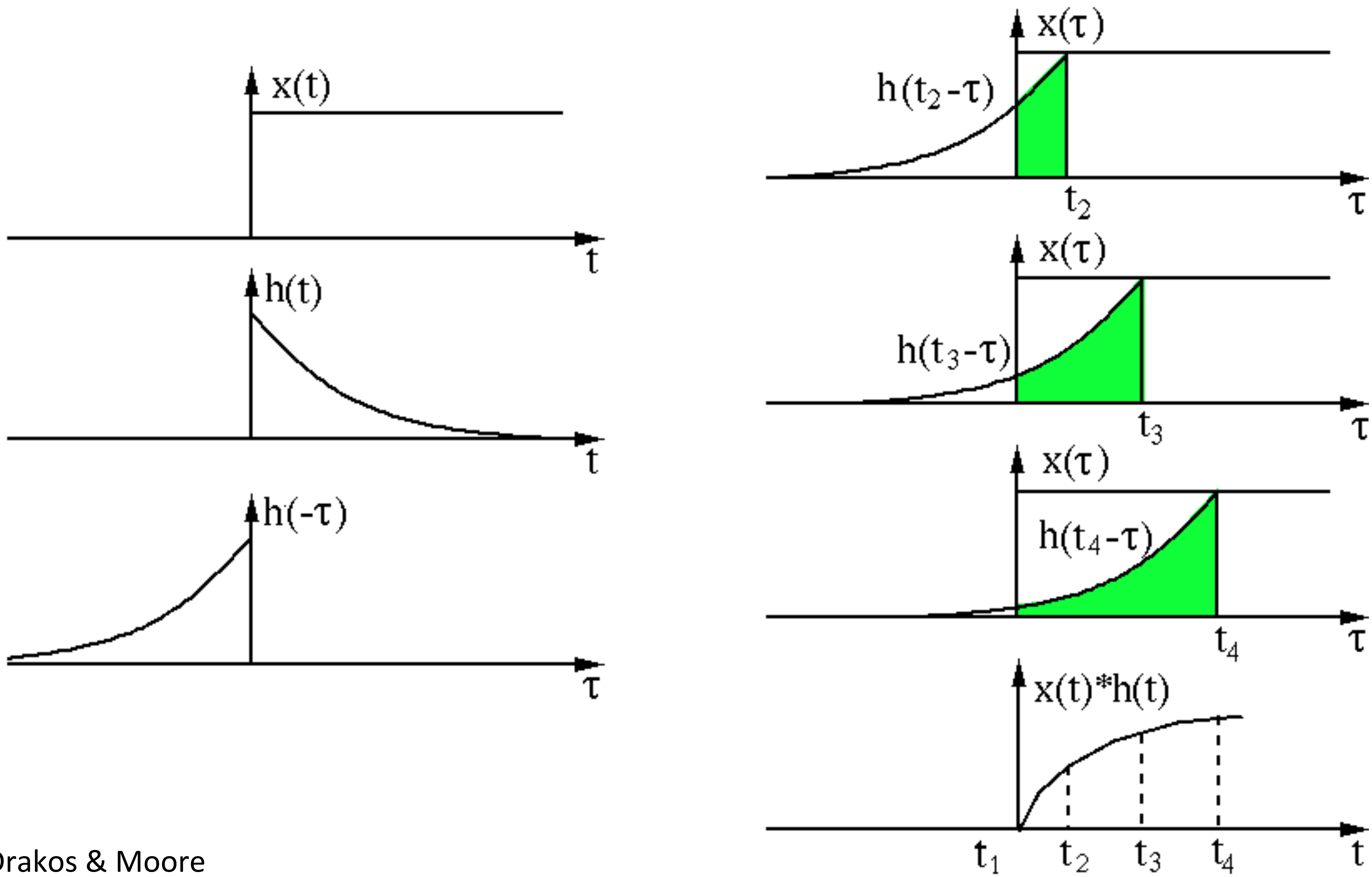
$$y(k) = x * h = conv(x,h) = \sum_{n=-\infty}^{+\infty} x(n)h(k-n)$$

- Correlação:

$$y(k) = corr(x,h) = \sum_{n=-\infty}^{+\infty} x(n)h(n+k)$$

Convolução

$$y(k) = x * h = conv(x,h) = \sum_{n=-\infty}^{+\infty} x(n)h(k-n)$$



Drakos & Moore

Convolução

$$y(k) = x * h = conv(x,h) = \sum_{n=-\infty}^{+\infty} x(n)h(k-n)$$

Inductiveload

Teorema da convolução

$$\mathcal{F}(xy) = \mathcal{F}(x) * \mathcal{F}(y)$$

$$\mathcal{F}(x*y) = \mathcal{F}(x)\ \mathcal{F}(y)$$

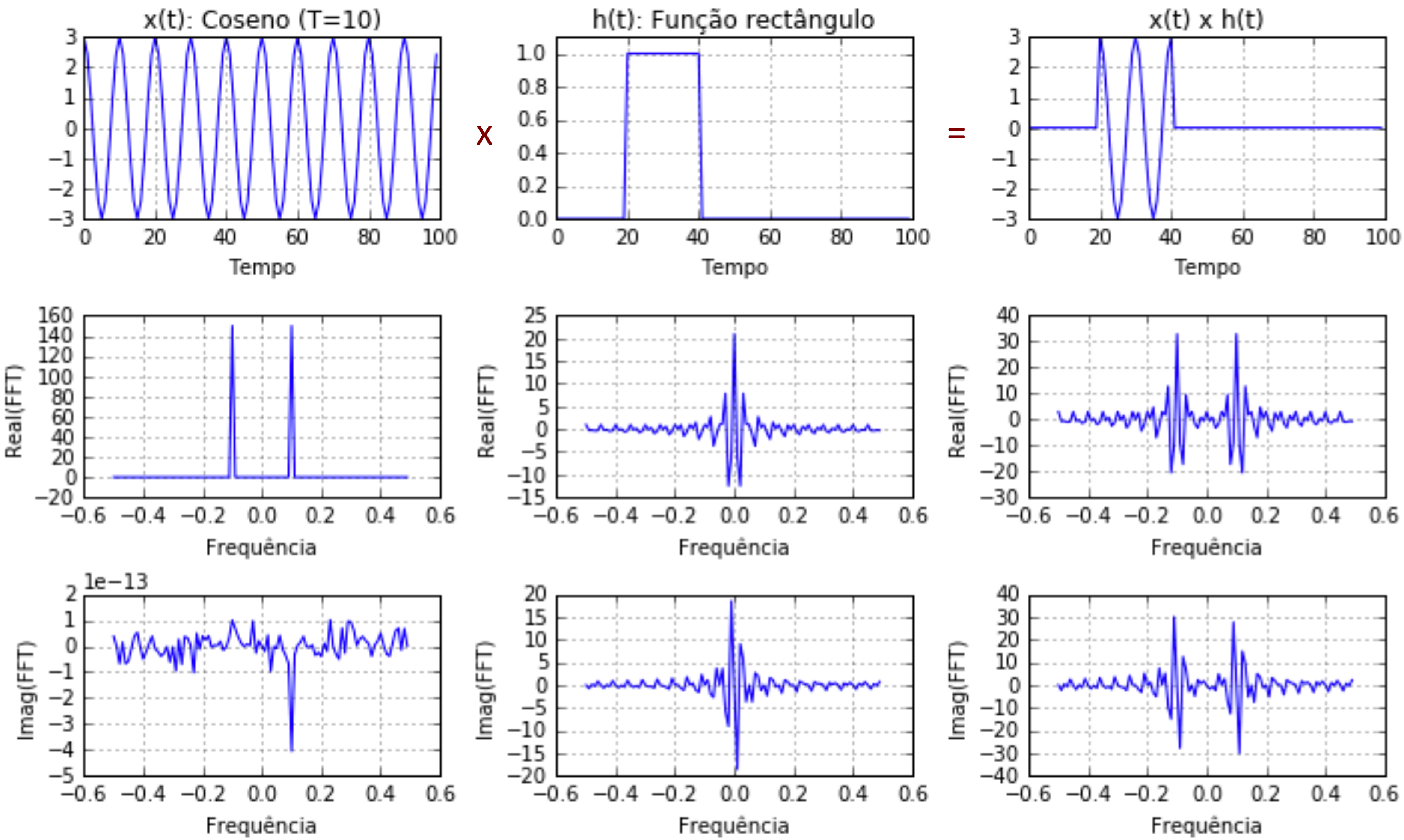
A transformada de Fourier do produto de duas funções é a convolução das suas transformadas (e vice versa).

Aplicações da convolução

- Os filtros lineares não recursivos (e.g. medias móveis pesadas) são convoluções no domínio do tempo. Logo, no domínio transformado, teremos o produto do espectro do input pelo espectro do filtro.

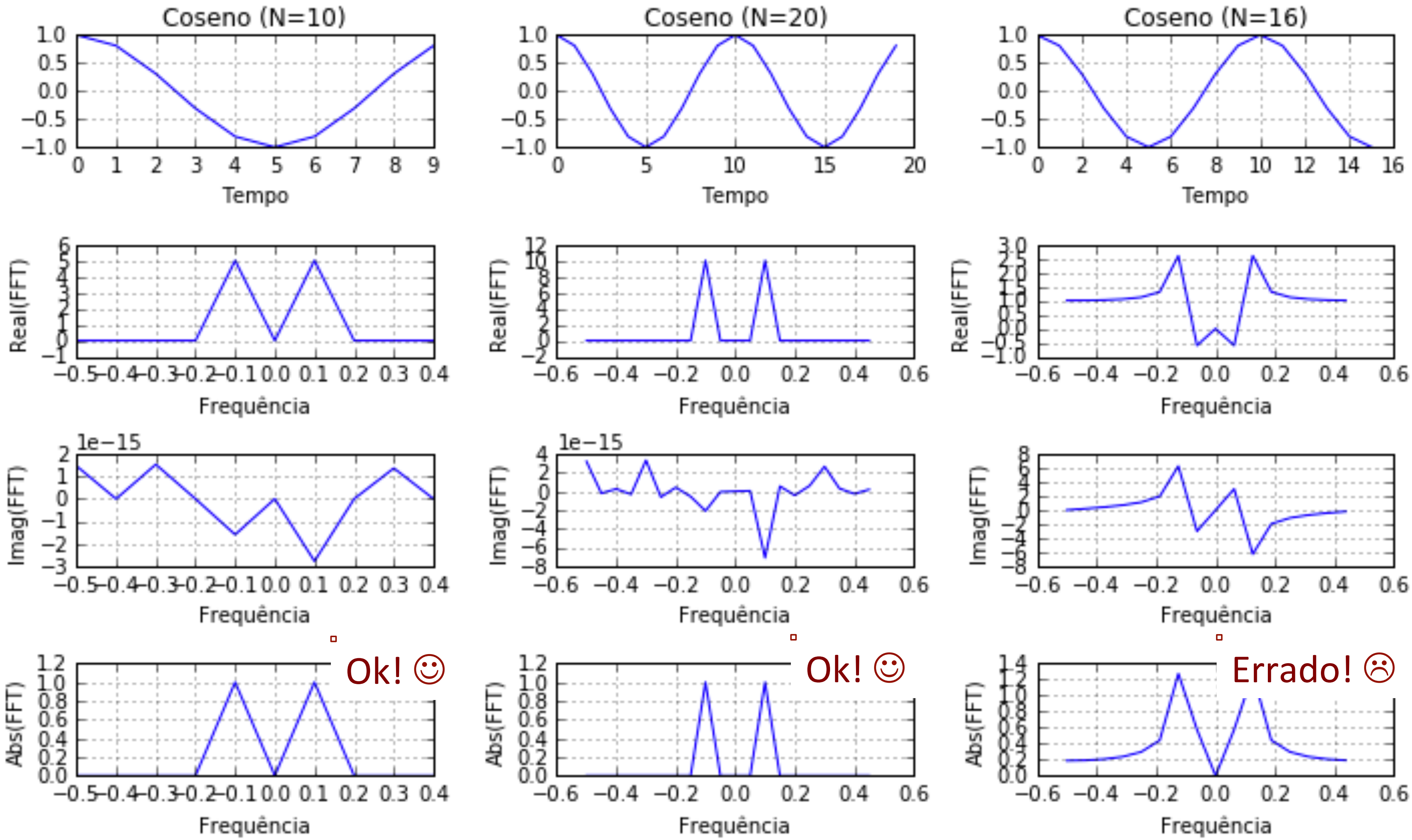
- Truncar uma série é multiplicá-la por uma função rectangular. Logo, no domínio transformado, teremos a convolução do espectro do input pelo espectro do filtro.

Truncamento



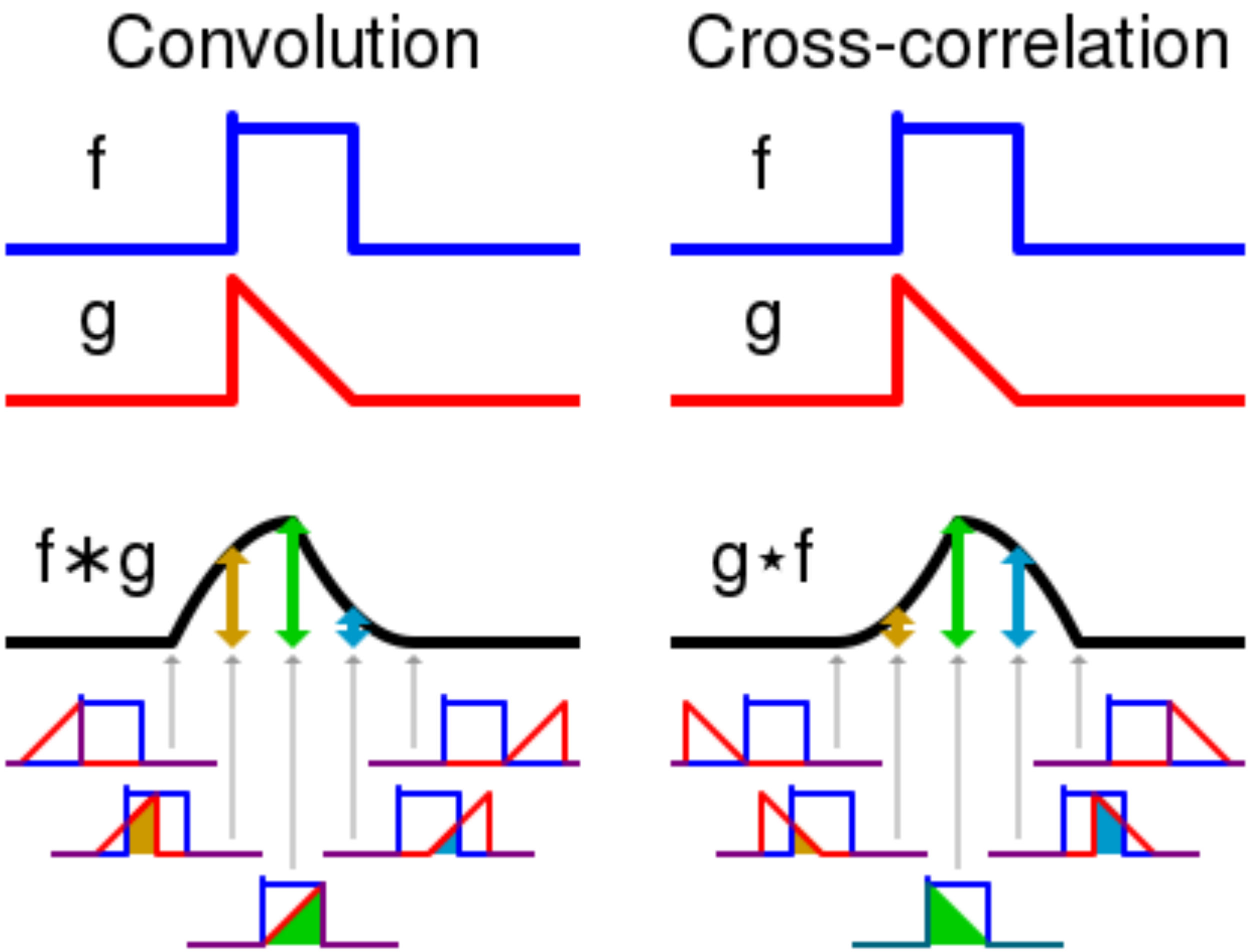
Truncar uma série com fracções do período

Mesmo sinal: T=10, Δt=1, mas N variável



Correlação

$$y(k) = corr(x,h) = \sum_{n=-\infty}^{+\infty} x(n)h(n+k)$$



Cmglee

Correlação

$$y(k) = corr(x, h) = \sum_{n=-\infty}^{+\infty} x(n)h(n+k)$$

UCBerkeley

Teorema da correlação

X=$\mathcal{F}$(x), Y=$\mathcal{F}$(y), Y* complexo conjugado de Y

Então:

$$\mathcal{F}(corr(x,y)) = X\,Y^*$$

$$\mathcal{F}(corr(x,x)) = X\,X^* \quad = \text{espectro de potência}$$
$$= (\text{espectro de amplitude})^2$$