Modelação Numérica 2017

Aula 3, 21/Fev

- Transformada Discreta de Fourier
- Propriedades da TDF

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

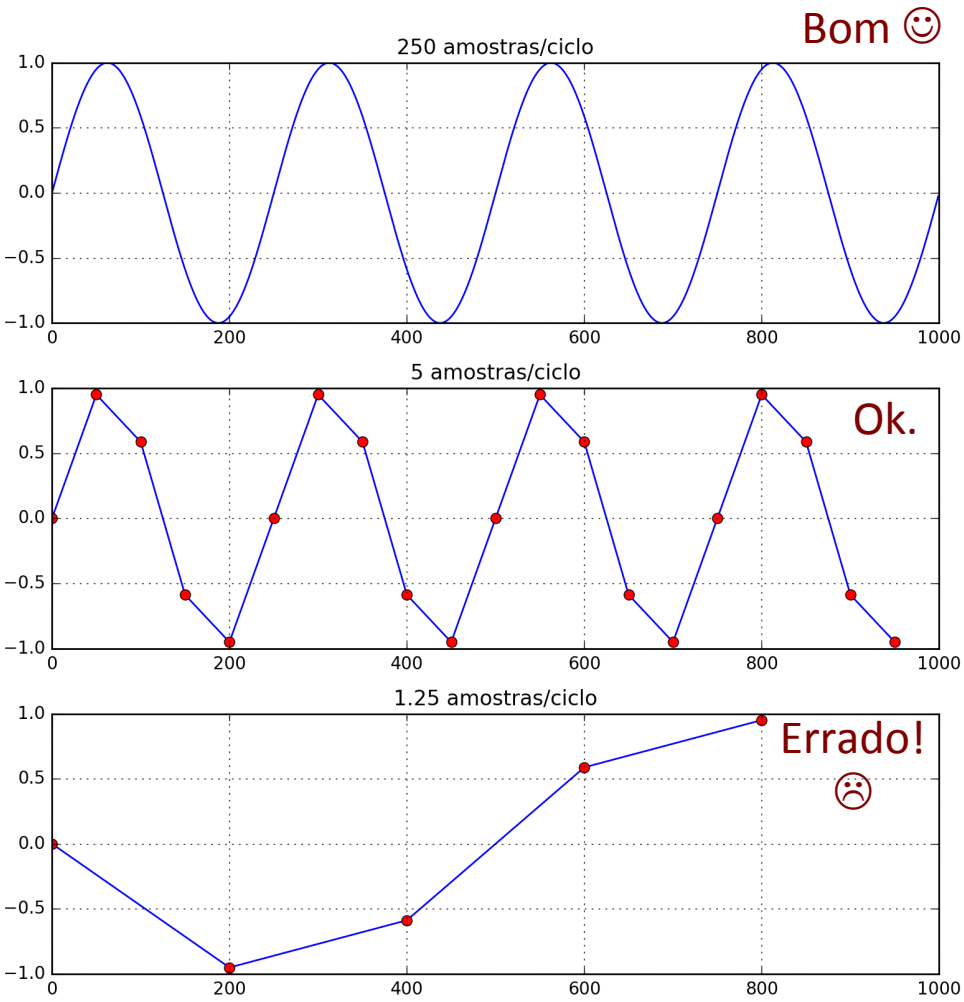
Aula passada

Teorema da amostragem

- Um sinal contínuo será mal representado se for amostrado a uma taxa inferior a 2 amostras por cada uma das oscilações mais rápidas presentes no sinal.

- Frequência de Nyquist:

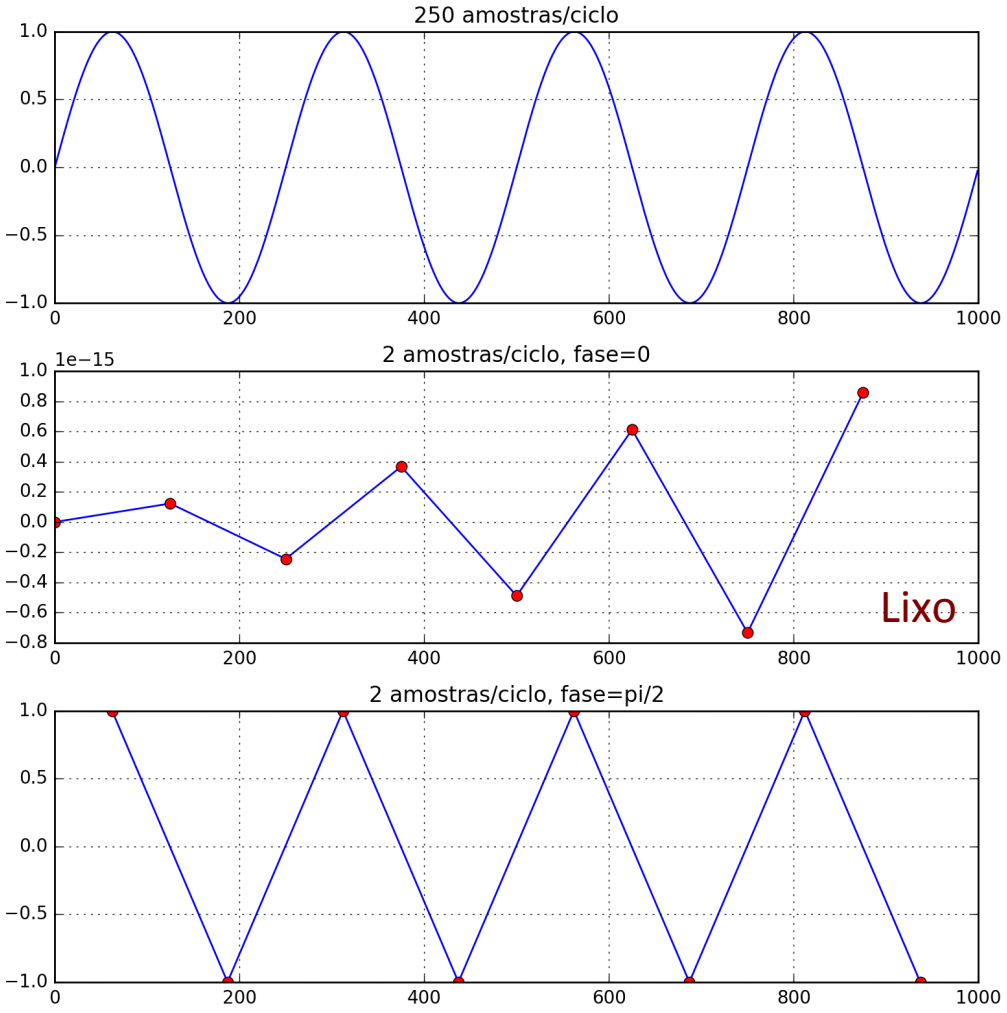
$$f_{Nyquist} = \frac{1}{2\Delta t}$$



Aula passada

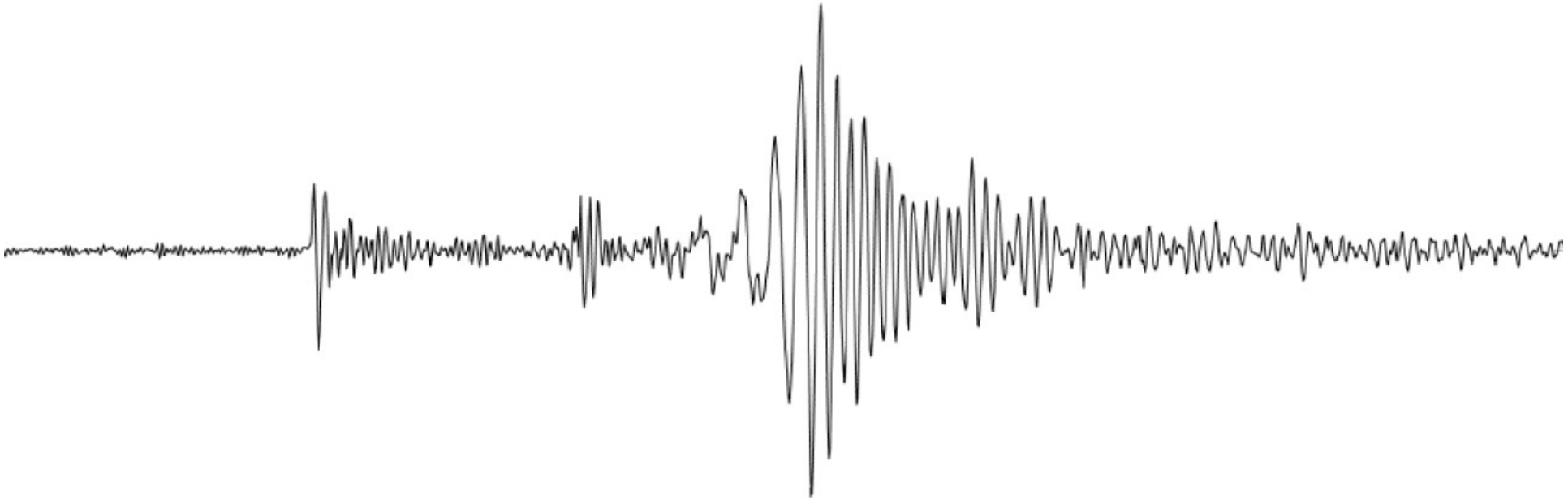
Amostragem no limite (2 amostras/ciclo)

```
t=np.arange(0.,1000.,1.)
T=250.
y=np.sin(2*math.pi * t/T)

plt.close(); plt.subplot(3,1,1)
plt.plot(t, y)
plt.title('250 amostras/ciclo')
plt.grid(); plt.xlim([0,1000])

#%%
dt=125
t2=t[0:len(t):dt];
y2=y[0:len(t):dt];

plt.subplot(3,1,2)
plt.plot(t2, y2, '-bo', markerfacecolor='red')
plt.title('2 amostras/ciclo, fase=0')
plt.grid(); plt.xlim([0,1000])

#%%
dt=125
t3=t[62:len(t):dt];
y3=y[62:len(t):dt];

plt.subplot(3,1,3)
plt.plot(t3, y3, '-bo', markerfacecolor='red')
plt.title('2 amostras/ciclo, fase=pi/2')
plt.grid(); plt.xlim([0,1000])
```



Aula passada

O que fazer com funções não-periódicas?

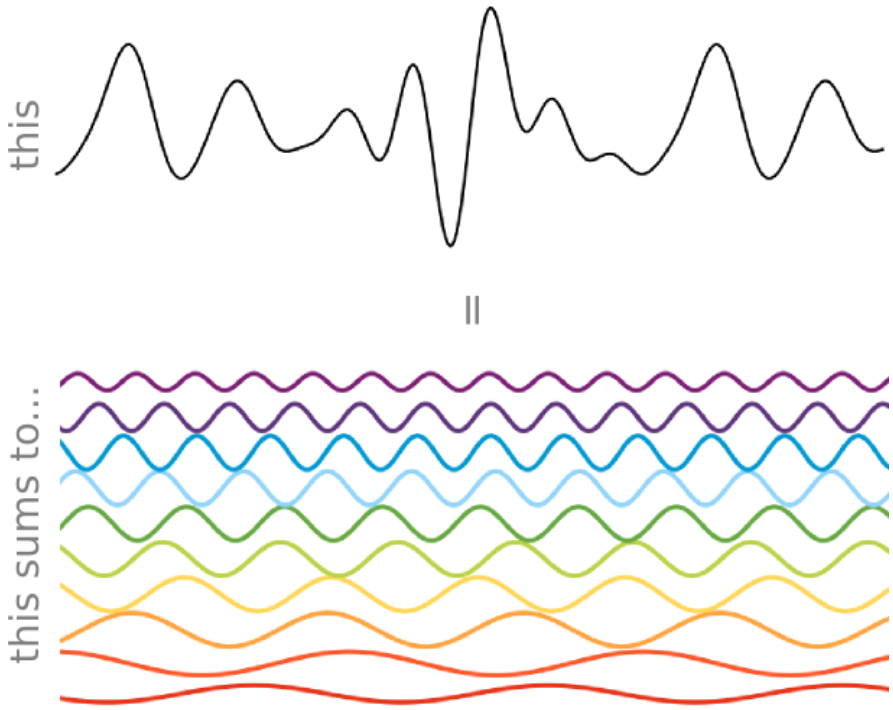


Aula passada

Generalização a qualquer função (não senos)

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

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



Aula passada

Série de Fourier

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

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

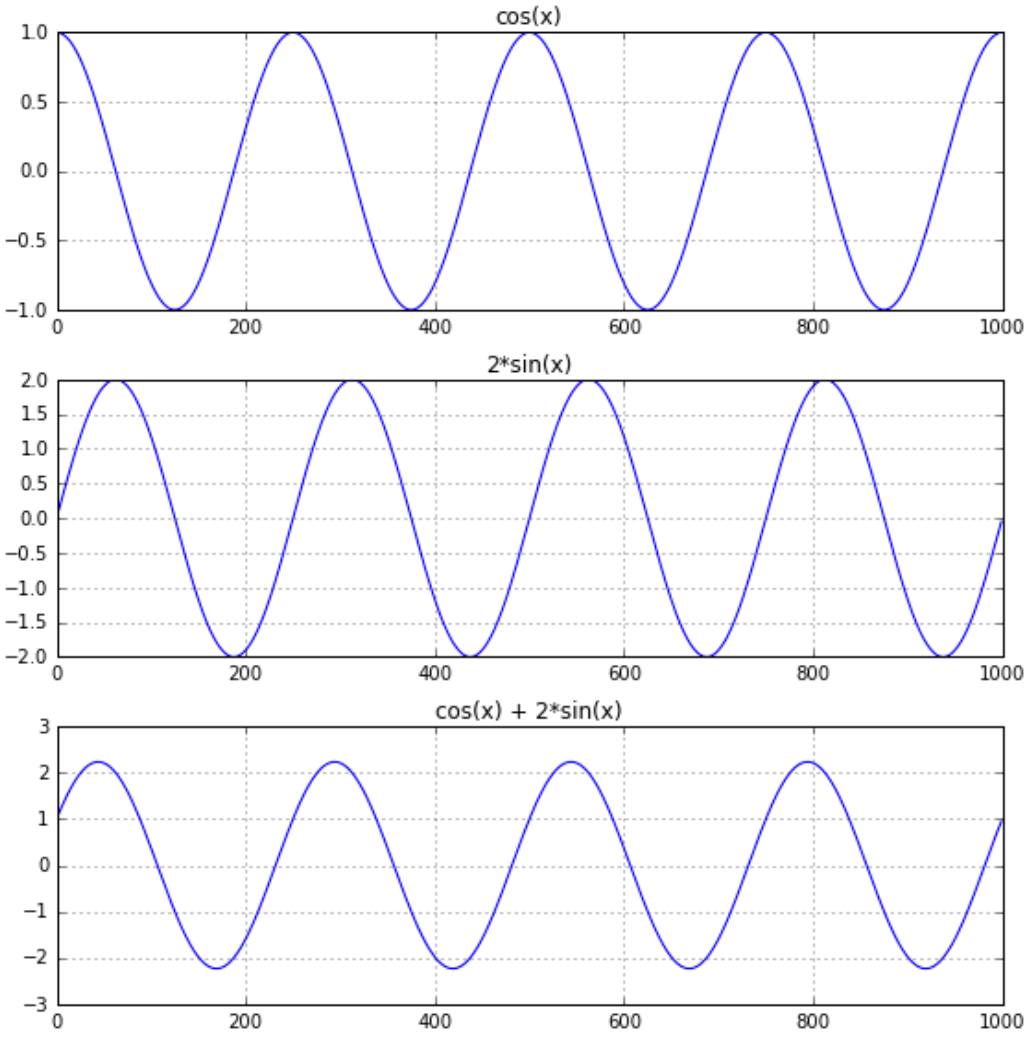
$$a_1 \cos \frac{2\pi t}{T} + b_1 \sin \frac{2\pi t}{T} = A \cos \left(\frac{2\pi t}{T} + \phi \right)$$

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

Aula passada

Série de Fourier

```
t=np.arange(0.,1000.,1.)
T=250.
y1=np.cos(2*math.pi * t/T)
y2=2*np.sin(2*math.pi * t/T)
y3=y1+y2

plt.close(); plt.subplot(3,1,1)
plt.plot(t, y1)
plt.title('cos(x)')

plt.subplot(3,1,2)
plt.plot(t, y2)
plt.title('2*sin(x)')

plt.subplot(3,1,3)
plt.plot(t, y3)
plt.title('cos(x) + 2*sin(x)')

for i in range(3):
    plt.subplot(3,1,i+1)
    plt.grid(); plt.xlim([0,1000])
plt.tight_layout()
```



Aula passada

Os coeficientes de Fourier são complexos

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

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

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

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

!! Série de Fourier: aplicável a funções complicadas, mas periódicas.

Transformada de Fourier:

Transformada de Fourier:

Contínua:

$$F(f) = \int_{-\infty}^{\infty} f(t) e^{-i2\pi ft} dt$$

Transformada de Fourier Inversa:

Contínua:

$$f(t) = \int_{-\infty}^{\infty} F(f) e^{i2\pi ft} df$$

Transformada de Fourier: Extensão da série de Fourier quando o período tende para infinito...

Transformada de Fourier:

- A transformada de Fourier pode ser generalizada a funções não periódicas...
- A transformada de Fourier pode ser definida com ligeiras diferenças:
  - O expoente das exponenciais pode ser simétrico (mudando os coeficientes c_k em conformidade);
  - O período por ser substituído pelo semi-período (desaparece o fator 2 no expoente, notem que o período mais grave que pode ser tocado num instrumento de cordas corresponde à oscilação que tem como semi-comprimento de onda a dimensão da corda);
  - Pode existir um fator multiplicativo diferente (que compensará nos coeficientes). Pode ser feita no espaço (x) e não no tempo (t) (ou até noutras dimensões...)
- Falta discretizar, truncar e arredondar.

Discretizar, truncar, arredondar

- Os resultados anteriores referem-se a uma função contínua mas periódica.
- Quando a função é discretizada (amostrada a intervalo regular) existe um período mínimo (ou frequência máxima $f_{Nyquist} = 1/(2\Delta t)$) que pode ser representado. Logo temos uma série discreta e finita (e com erro de arredondamento).
- O processo de discretização não introduz erro se a função for de banda limitada e se Δt for suficiente pequeno, i.e. $f_{max} < f_{Nyquist}$ e o número de pontos suficientemente grande para conter pelo menos 1 período dos mais longos.
- Na prática, a análise numérica de dados reais refere-se sempre a esse tipo de série. Nesse caso tanto a representação da função (transformada inversa) como o cálculo dos coeficientes (transformada) envolve somatórios (não integrais) com um número finito de termos: Tansformada Discreta de Fourier (TDF).

Transformada de Fourier:

Transformada de Fourier:

Contínua:

$$F(f) = \int_{-\infty}^{\infty} f(t) e^{-i2\pi ft} dt$$

Discreta:

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

Transformada de Fourier Inversa:

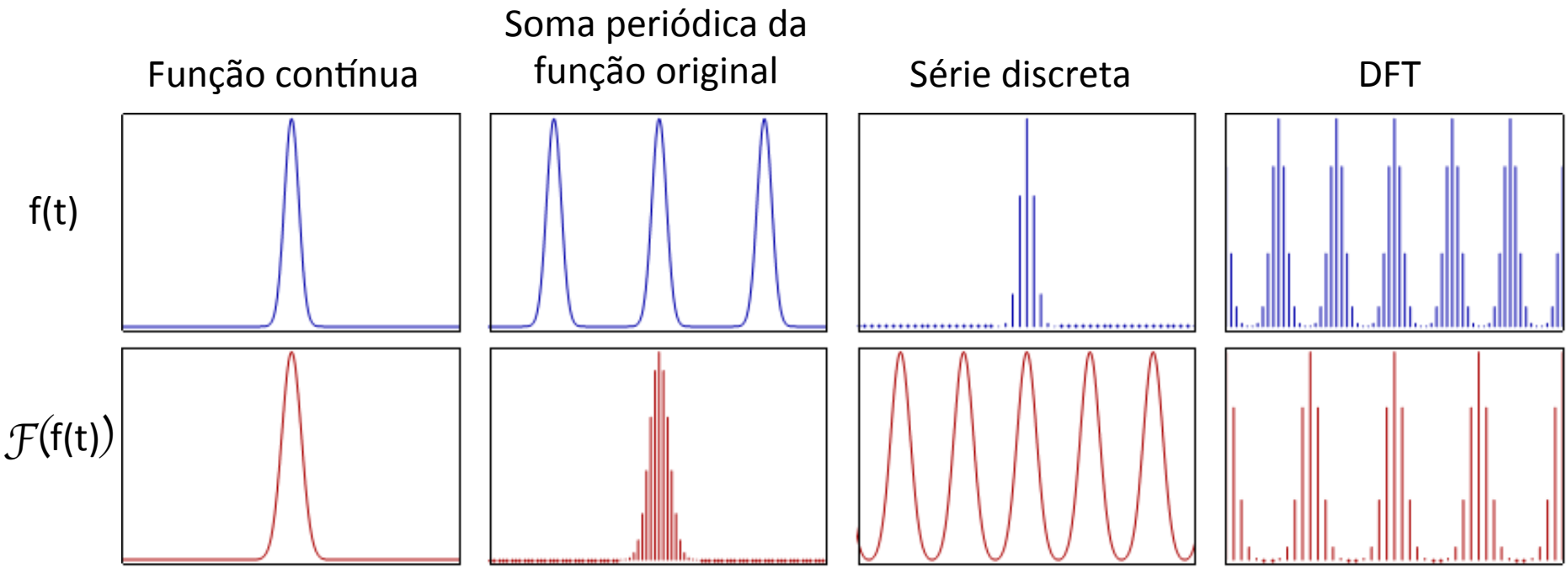
Contínua:

$$f(t) = \int_{-\infty}^{\infty} F(f) e^{i2\pi ft} df$$

Discreta:

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

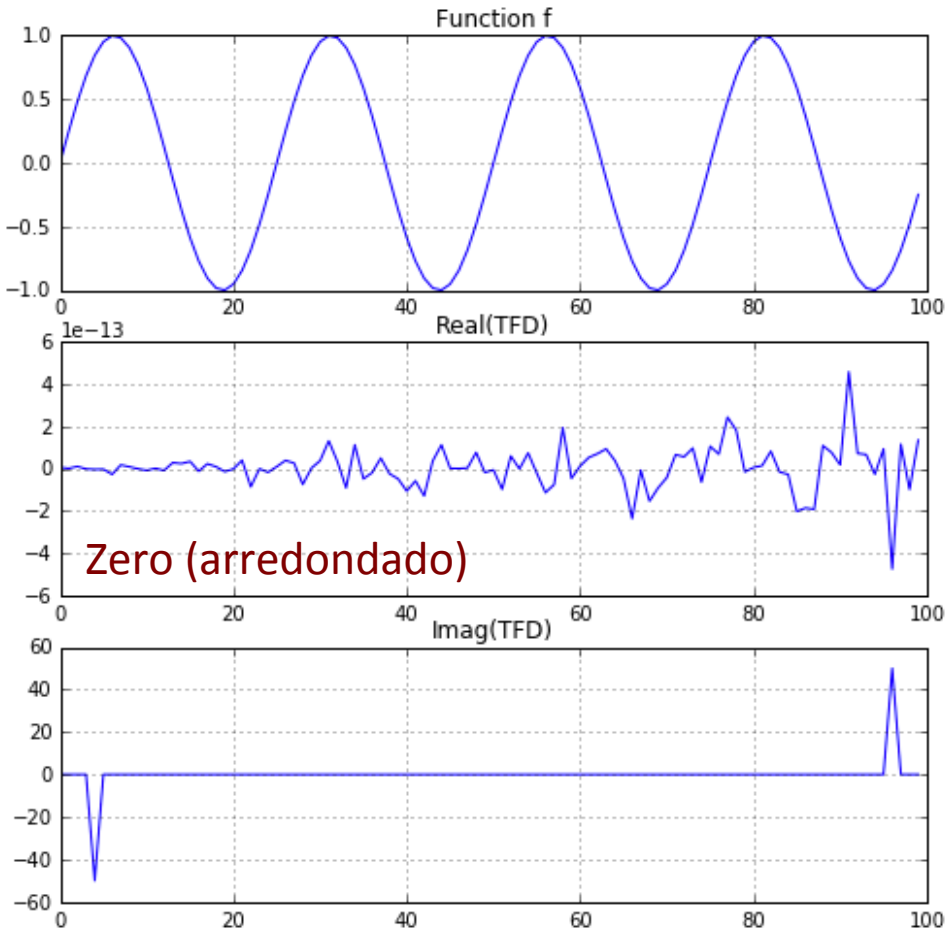
Transformada de Fourier contínua e discreta



Transformada Discreta de Fourier (TDF)

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

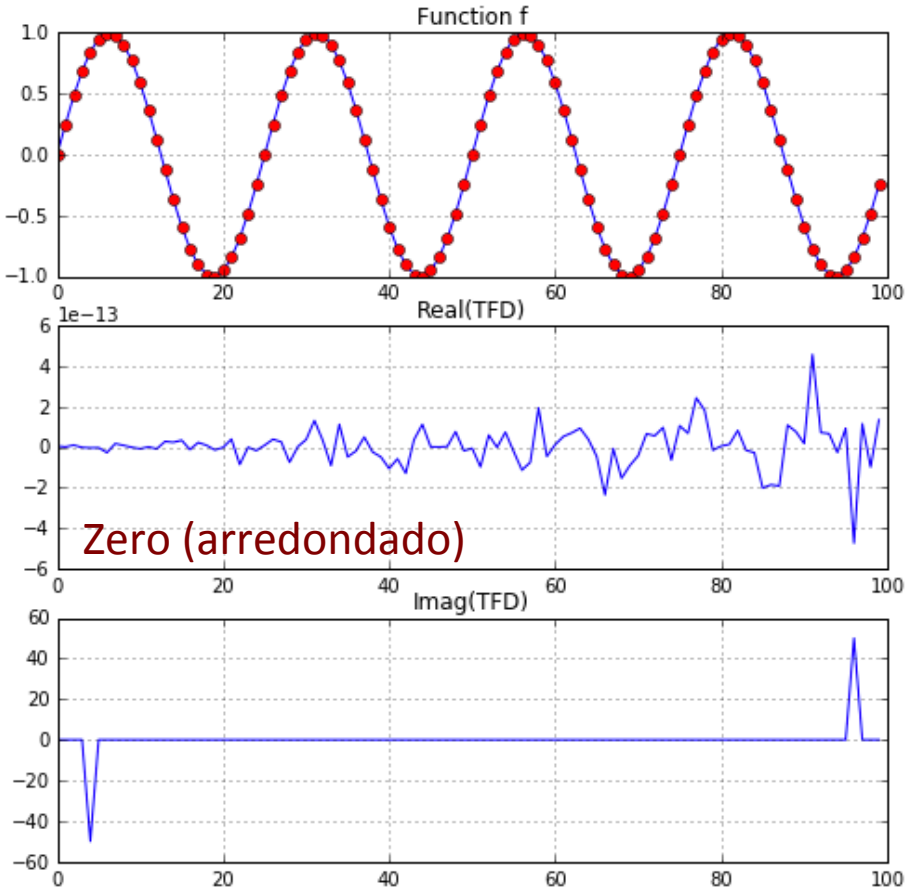
```
import matplotlib.pyplot as plt
import numpy as np
from numpy import pi as pi

#%%
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=25;
t=np.arange(0.,N,1.)
f=np.sin(2*pi * t/T)

tf=TFD(f)

plt.close();
plt.subplot(3,1,1); plt.plot(t, f)
plt.title('Function f'); plt.grid();

plt.subplot(3,1,2); plt.plot(np.arange(0,N), np.real(tf))
plt.title('Real(TFD)'); plt.grid();

plt.subplot(3,1,3); plt.plot(np.arange(0,N), np.imag(tf))
plt.title('Imag(TFD)'); plt.grid();
```



Transformada Discreta de Fourier Inversa (TDFi)

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

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

#%%
N=100.; T=25;
t=np.arange(0.,N,1.)
f=np.sin(2*pi * t/T)
tf=TFD(f)

plt.close();
plt.subplot(3,1,1); plt.plot(t, f)
plt.title('Function f'); plt.grid();

plt.subplot(3,1,2); plt.plot(np.arange(0,N), np.real(tf))
plt.title('Real(TFD)'); plt.grid();

plt.subplot(3,1,3); plt.plot(np.arange(0,N), np.imag(tf))
plt.title('Imag(TFD)'); plt.grid();

ff=TFDi(tf)
plt.subplot(3,1,1); plt.plot(t, ff, 'or')
```



Transformada Discreta de Fourier Inversa (TDFi)

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

A transformada não perde informação.

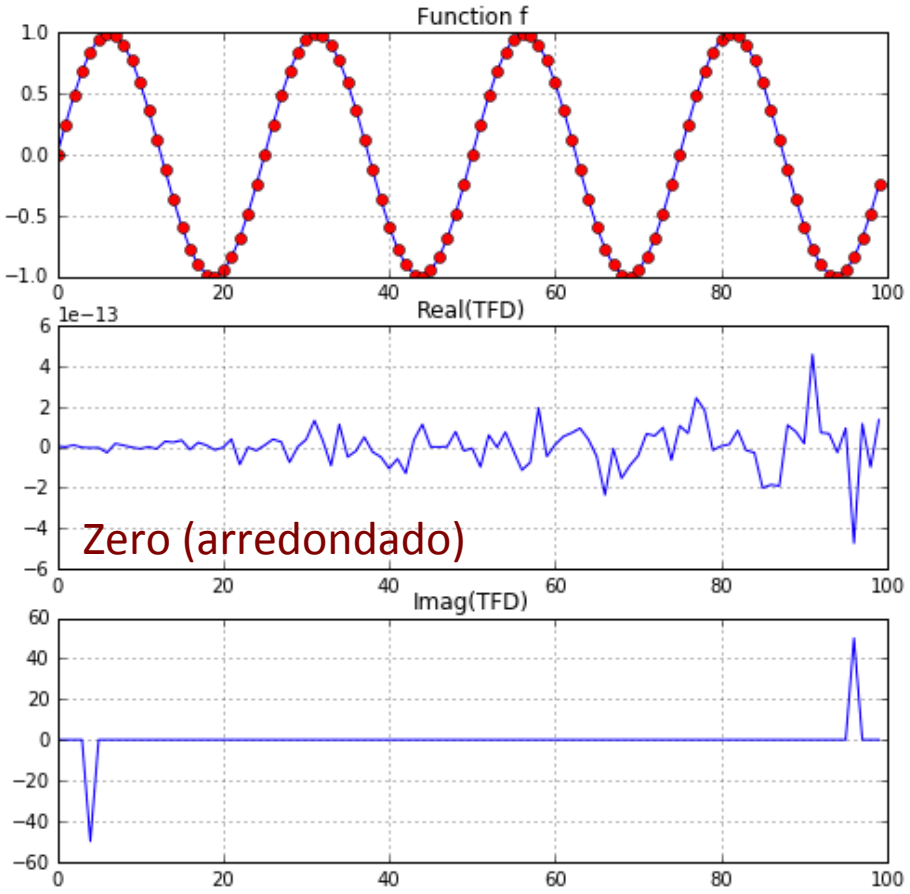
Pode-se recuperar a série original com a transformada inversa.

$\mathcal{F}$: Transformada de Fourier

$\mathcal{F}^{-1}$: Transformada Inversa de Fourier

F = $\mathcal{F}$(f)

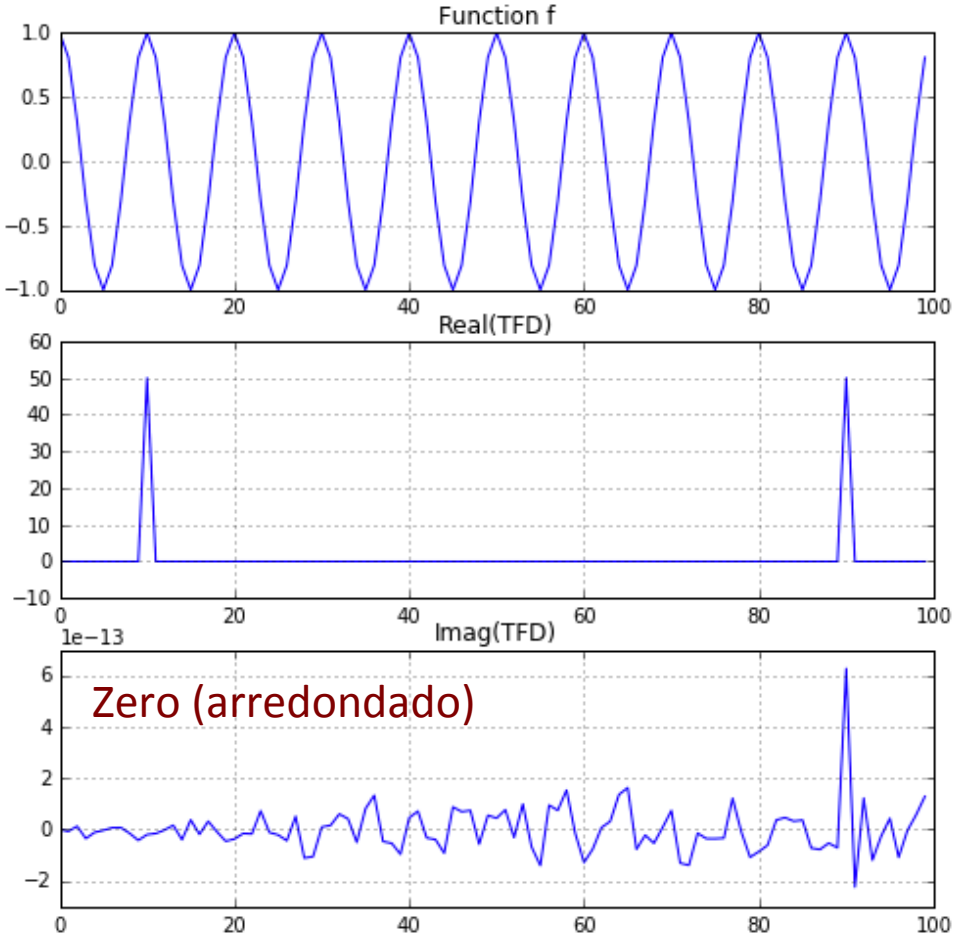
f = $\mathcal{F}^{-1}$(F) = $\mathcal{F}^{-1}$($\mathcal{F}$(f))



Interpretação do espectro: Frequência

Coseno, 10 ciclos exactos, T = 10 s, f = 0.1 Hz

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

tf=TFD(f)

plt.close();
plt.subplot(3,1,1); plt.plot(t, f)
plt.title('Function f'); plt.grid();

plt.subplot(3,1,2);
plt.plot(np.arange(0,N), np.real(tf))
plt.title('Real(TFD)'); plt.grid();

plt.subplot(3,1,3);
plt.plot(np.arange(0,N), np.imag(tf))
plt.title('Imag(TFD)'); plt.grid();
```



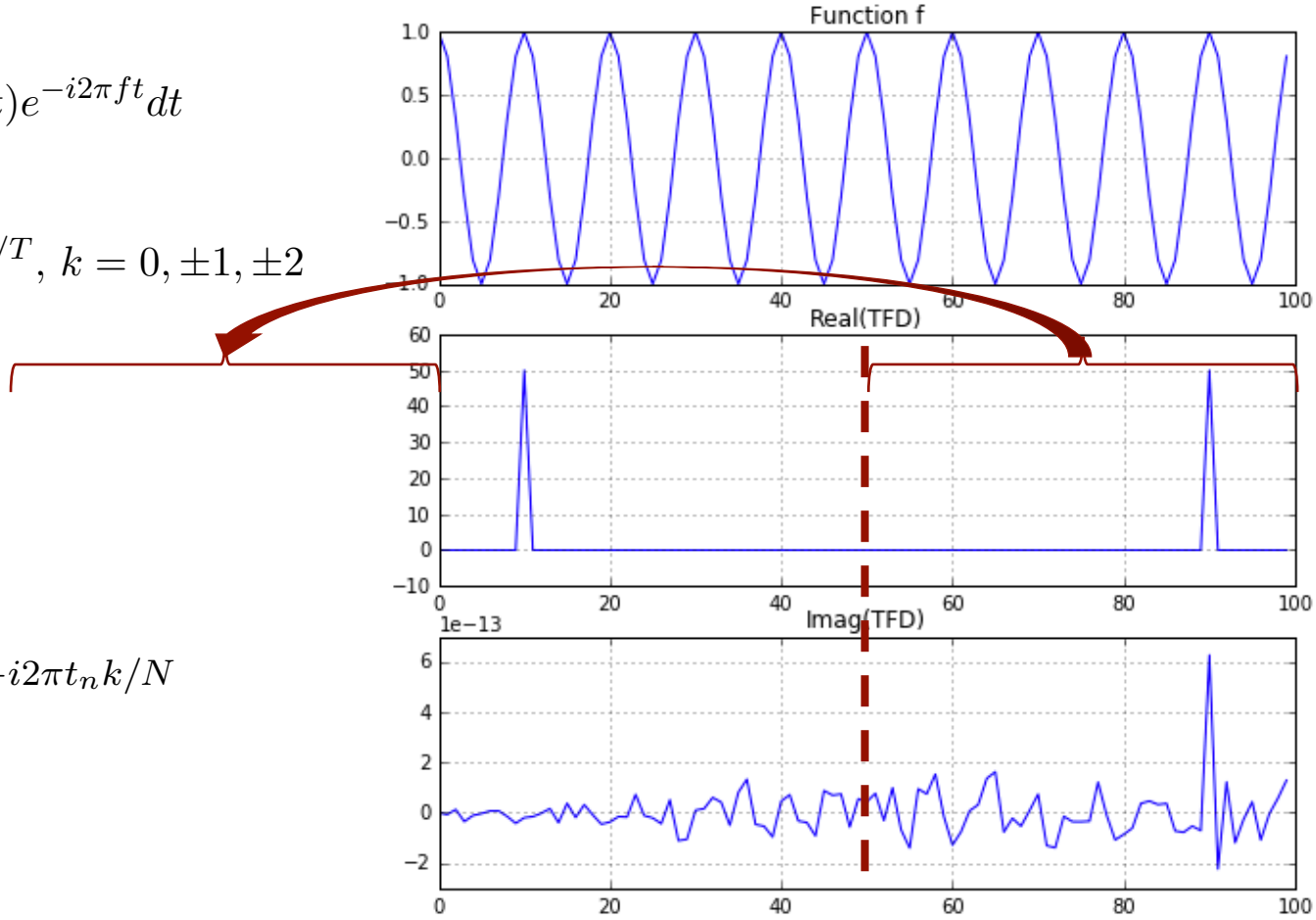
Interpretação do espectro: Frequência

Coseno, 10 ciclos exactos, T = 10 s, f = 0.1 Hz

$$F(f) = \int_{-\infty}^{\infty} f(t) e^{-i2\pi ft} dt$$

$$c_k = \frac{1}{2\pi} \int_{-T/2}^{T/2} f(t) e^{-i2\pi kt/T},\ k = 0, \pm 1, \pm 2$$

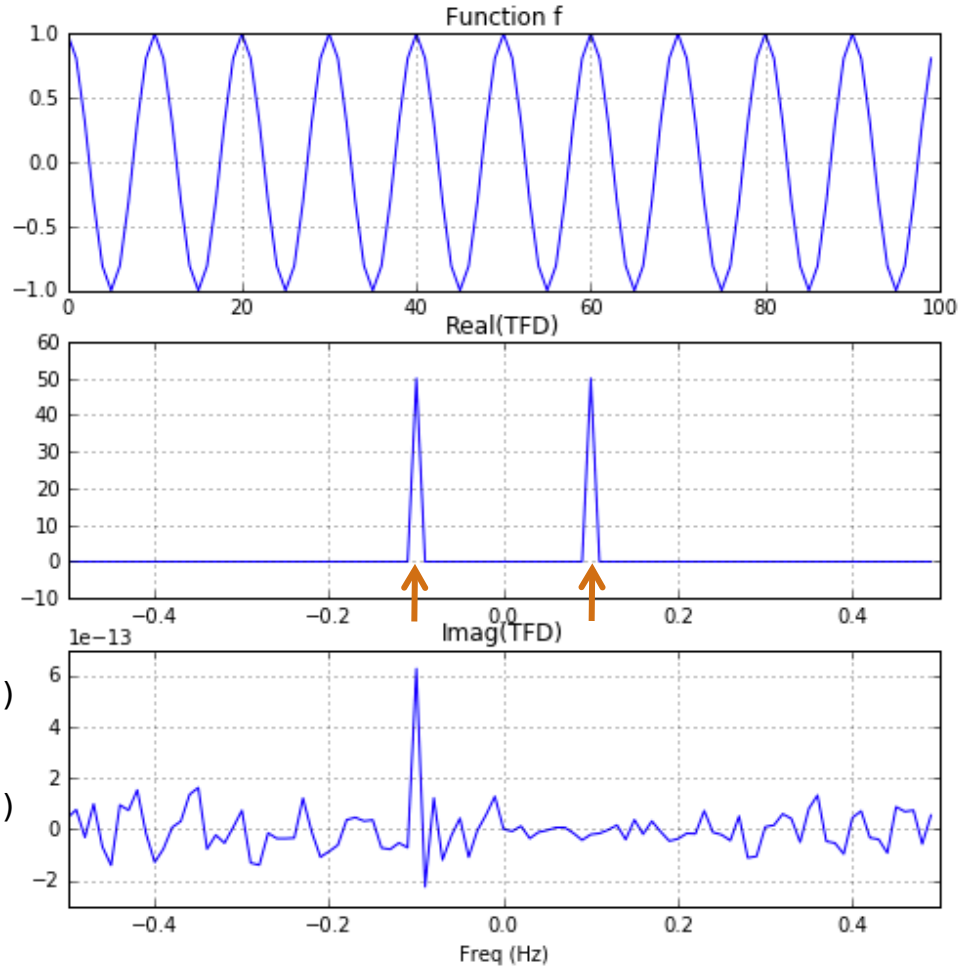
$$c_k = \sum_{t=0}^{N-1} f(t_n) e^{-i2\pi t_n k/N}$$



Interpretação do espectro: Frequência

Coseno, 10 ciclos exactos, T = 10 s, f = 0.1 Hz

```
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(3,1,1); plt.plot(t, f)
plt.title('Function f'); plt.grid();

plt.subplot(3,1,2); plt.plot(freq,np.real(fout))
plt.title('Real(TFD)'); plt.grid();

plt.subplot(3,1,3); plt.plot(freq,np.imag(fout))
plt.title('Imag(TFD)'); plt.grid();
```



Re-ordenando, obtém-se o espectro habitual.

Transformada de Fourier

https://en.wikipedia.org/wiki/File:Fourier_transform_time_and_frequency_domains_(small).gif