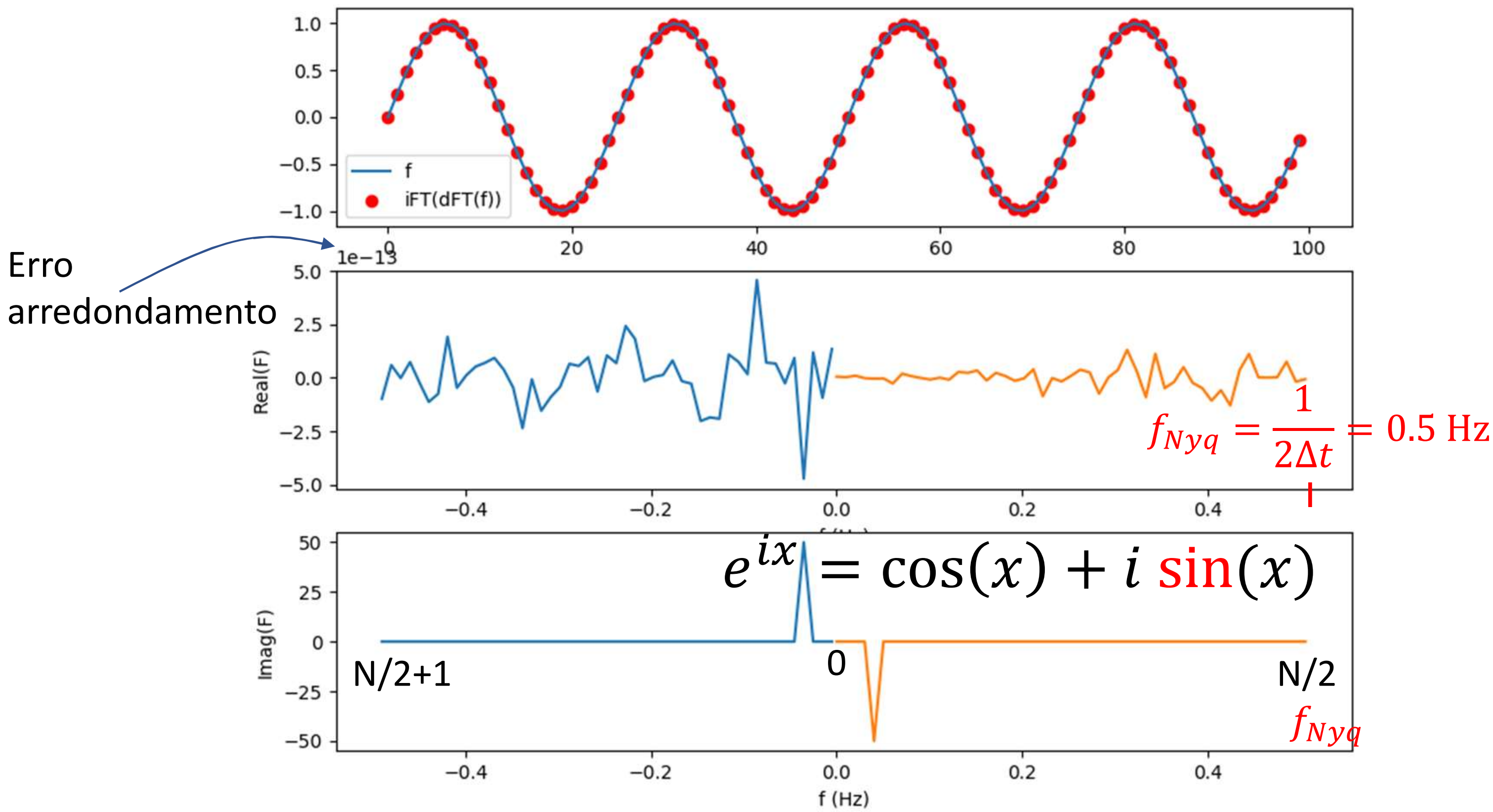
Modelação Numérica
Aula 3

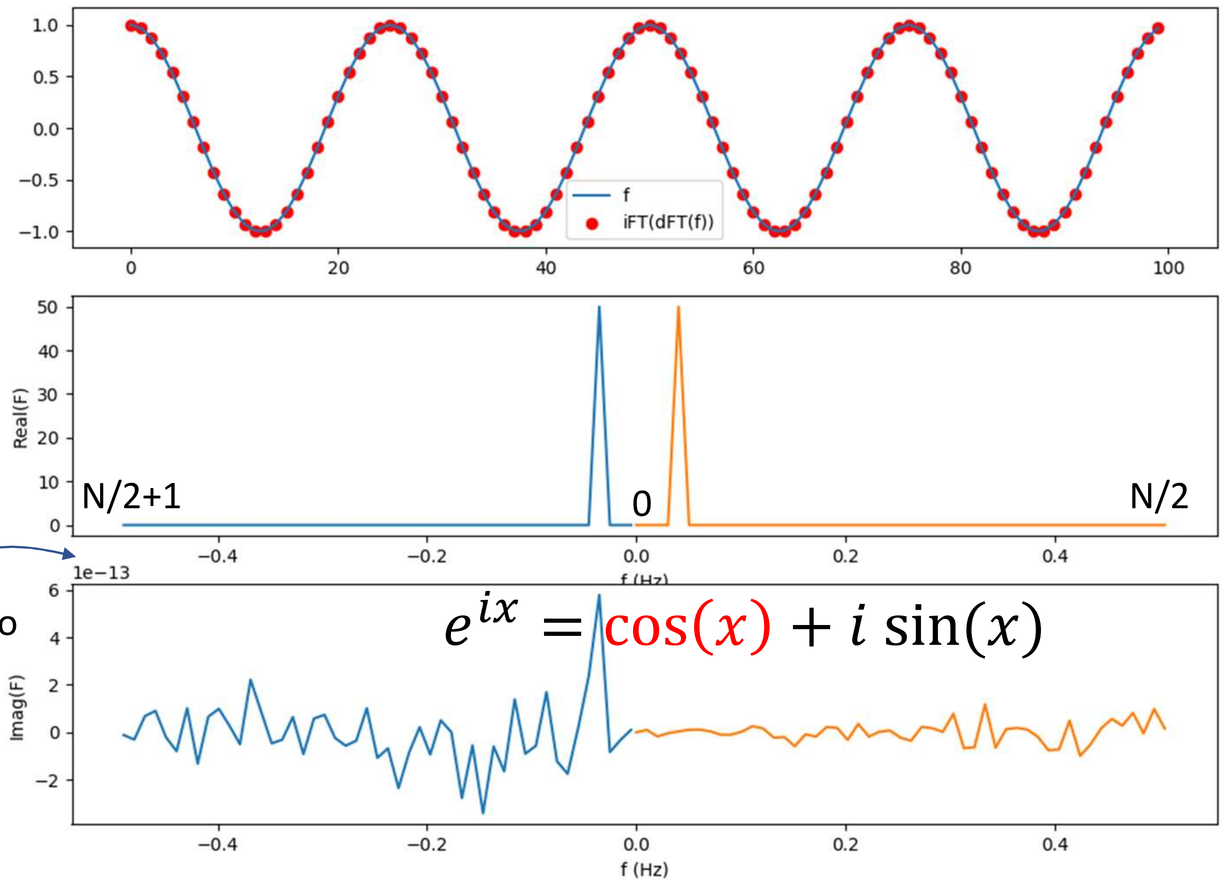
Espectros, FFT

Transformada de $\boldsymbol{sin}\left(\frac{2\pi t}{25}\right)$



Transformada de $\boldsymbol{cos}\left(\frac{\mathbf{2}\boldsymbol{\pi t}}{\mathbf{25}}\right)$

Erro
arredondamento



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

O teorema de Fourier

$$F_k(k) = \sum_{n=0}^{N-1} f_n e^{-2\pi ink/N},\ f_n(n) = \frac{1}{N}\sum_{k=0}^{N-1} F_k e^{2\pi ink/N}$$

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

Diz que eu posso reproduzir uma série temporal com N elementos, pela soma de N sinusoides (algumas talvez com amplitude 0), de forma exata (a menos do erro de arredondamento).

A reconstituição da série temporal inclui uma constante (média) e sinusoides (harmónicas) com frequência múltipla de uma frequência fundamental ($2\pi/T$).

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

Deve notar-se que:

$$a_k \, cos \frac{2\pi k t}{T} + b_k \, sin \frac{2\pi k t}{T} = c_k \sin\left(\frac{2\pi k}{T} + \phi_k\right)$$

Onde c_k é a amplitude da sinusóide, e ϕ_k a sua fase inicial.

Na forma complexa:

$$c_k = |F_k|, \phi_k = \tan^{-1}\left(\frac{\mathrm{Imag}(F_k)}{\mathrm{Real}(F_k)}\right)$$

Espectros de amplitude e de fase

Em vez de analisar a parte Real e a parte Imaginária da transformada de Fourier, podemos analisar o seu módulo, designado por espectro de amplitude:

$$A(\omega) = |F(\omega)|$$

O seu espectro de fase (ângulo no plano complexo)

$$\Phi(\omega) = \tan^{-1}\left(\frac{Imag(F)}{Real(F)}\right)$$

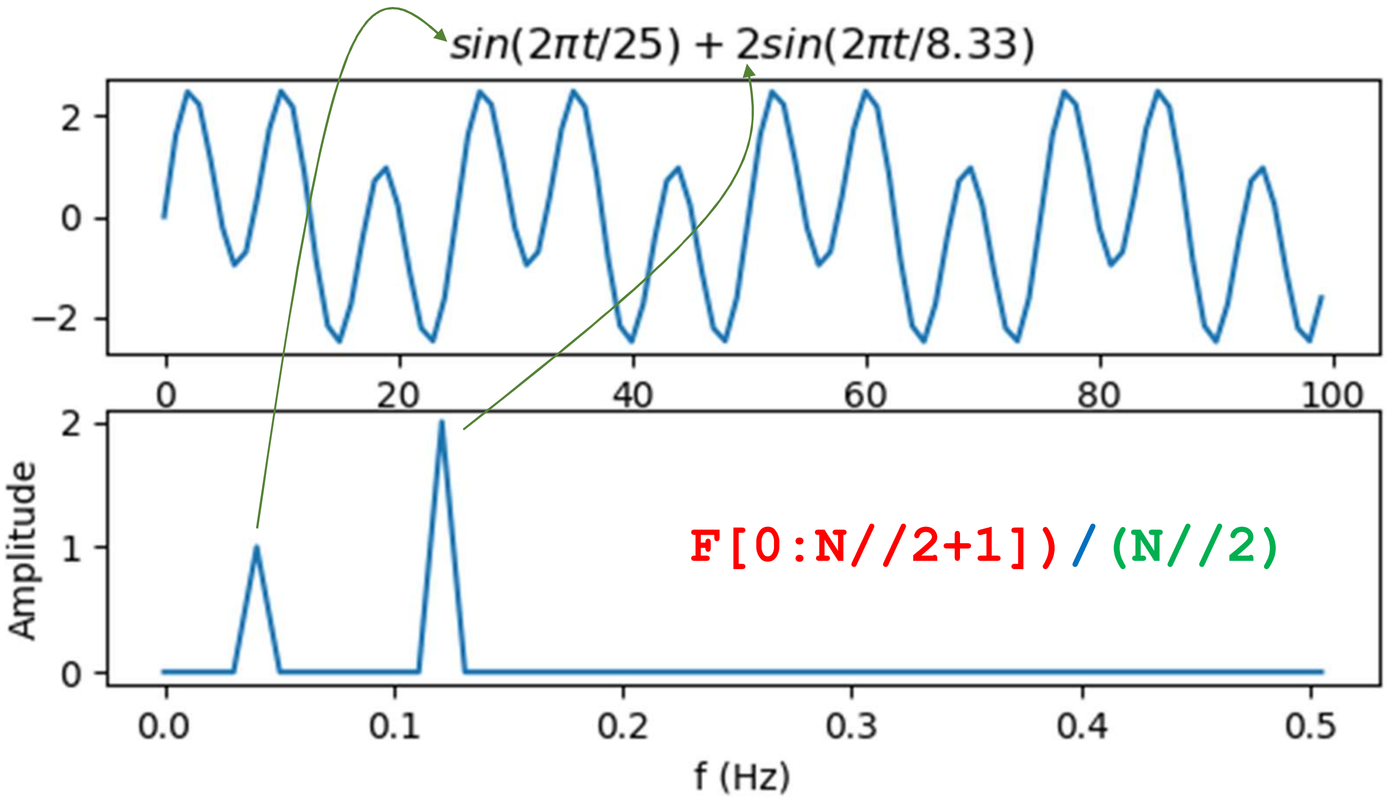
Onde $\omega = 2\pi f$

Se f for real, o espectro de amplitude é simétrico.

fft da soma de dois senos

```
import numpy as np;import matplotlib.pyplot as plt
N=100;dt=1.;T=N*dt/4.;
t=np.linspace(0,dt*(N-1),N);
f=np.sin(2*np.pi*t/T)+2*np.sin(2*np.pi*t/(T/3))
plt.subplot(3,1,1);plt.plot(t,f)
plt.title(r"$sin(2\pi t/25)+2 sin (2\pi t/8.33 )$")
F=np.dFT(f)
fNyq=1/(2*dt); df=2*fNyq/(N-1)
freq=np.arange(0,fNyq+df,df)
plt.subplot(3,1,2)
plt.plot(freq,np.abs(F[0:N//2+1])/(N//2)) #IMPORTANTE
plt.ylabel('Amplitude');plt.xlabel('f (Hz)')
```

dFT da soma de dois senos



Só se mostra para $F > 0$ (é uma função par)

Fast Fourier Transform $\boldsymbol{N = 2^k 3^m 5^n}$

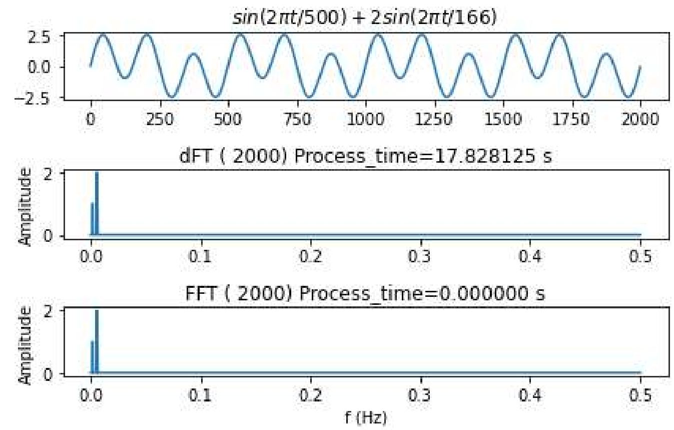
As simetrias dos senos e cosenos podem ser aproveitadas para desenhar algoritmos muito eficientes **se** o Número de pontos da amostra for da forma $N = 2^k 3^m 5^n, (k, n, m \in \mathbb{N})$.

Transformada discreta de Fourier em numpy

```
F=np.fft.fft(f)
```

Transformada inversa

```
f=np.fft.ifft(F)
```

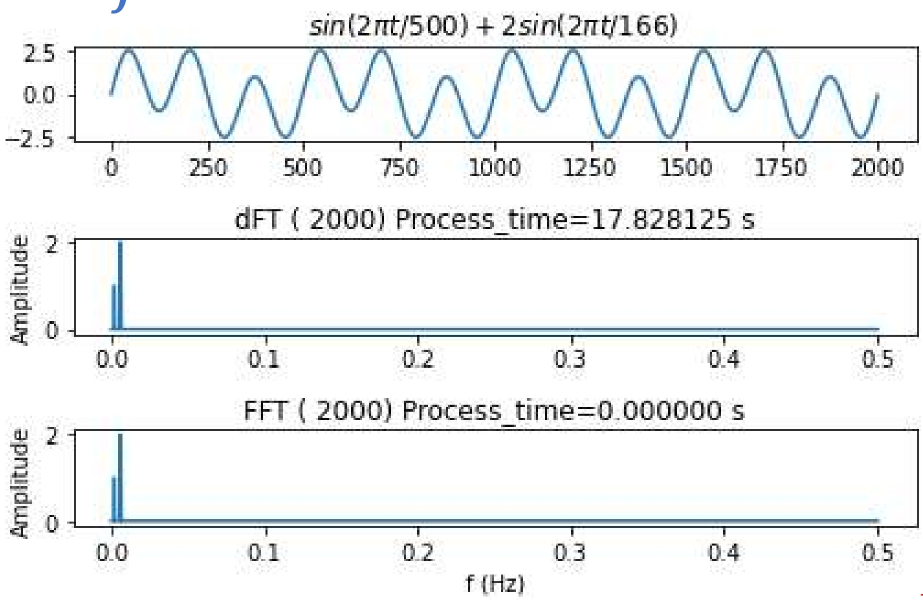


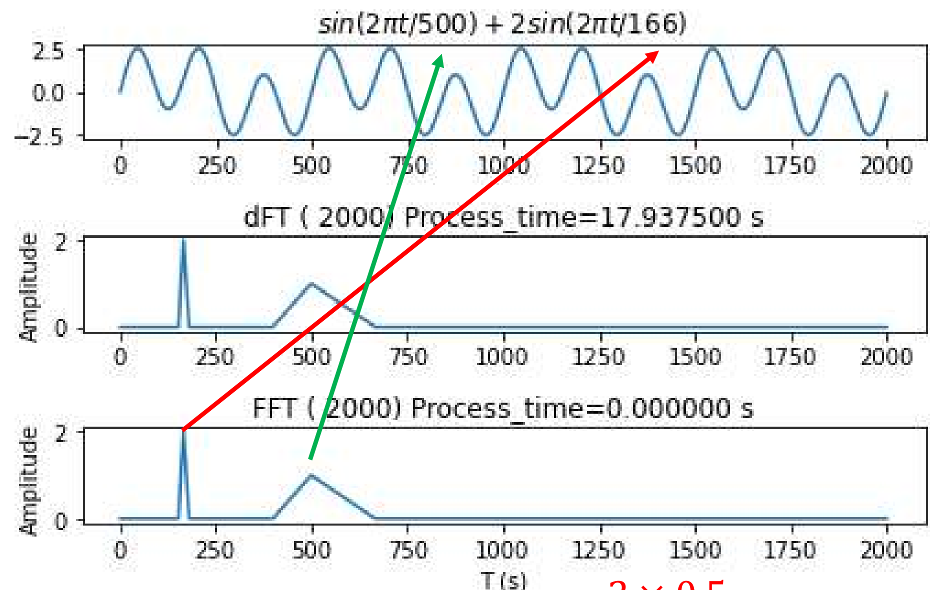

```
from scipy import fft
N=2000;dt=1.;T=N*dt/4.
t=np.linspace(0,dt*(N-1),N);

f=np.sin(2*np.pi*t/T)+2*np.sin(2*np.pi*t/(T/3))
fig,ax=plt.subplots(nrows=3)
ax[0].plot(t,f)
ax[0].set_title(r"$sin(2\pi t/%3i)+2 sin (2\pi t/%3i )$" %(T,T/3))
t0=process_time();F=dFT(f);t1=process_time()
fNyq=1/(2*dt); df=2*fNyq/(N-1)
freq=np.arange(0,fNyq+df,df)
ax[1].plot(freq,np.abs(F[0:N//2+1])/(N//2))
ax[1].set_ylabel('Amplitude')
ax[1].set_title('dFT (%5i) Process_time=%8.6f s' % (N,t1-t0))
t0=process_time();F=fft.fft(f);t1=process_time()
ax[2].plot(freq,np.abs(F[0:N//2+1])/(N//2))
ax[2].set_ylabel('Amplitude');plt.xlabel('f (Hz)')
ax[2].set_title('FFT (%5i) Process_time=%8.6f s' % (N,t1-t0))

fig.tight_layout()
```

$$F(freq) \rightarrow F(T)$$

$\bar{f} = 0$

Notas: $\bar{f} = 0;\ f(0) = 0; F[0] = 0; T_{MAX} = \infty$





$f_{MAX} = 0.5Hz \Longrightarrow T = 2s$

$f_{MIN} = \dfrac{2 \times 0.5}{2000} Hz \Longrightarrow T = 2000s$

Nota: $\dfrac{1}{freq[0]} = \infty$

```
ax[2].plot(1/freq,np.abs(F[0:N//2+1])/(N//2))
ax[2].set_ylabel('Amplitude');plt.xlabel('T (s)')
ax[2].set_title('FFT (%5i) Process_time=%8.6f s' % (N,t1-t0))
```

RESUMO: Uma série finita (amostra) ...

Tem uma transformada discreta de Fourier

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

e existe uma transformada discreta inversa tal que:

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

Trata-se de **operações exatas** (**a menos do erro de arredondamento**).

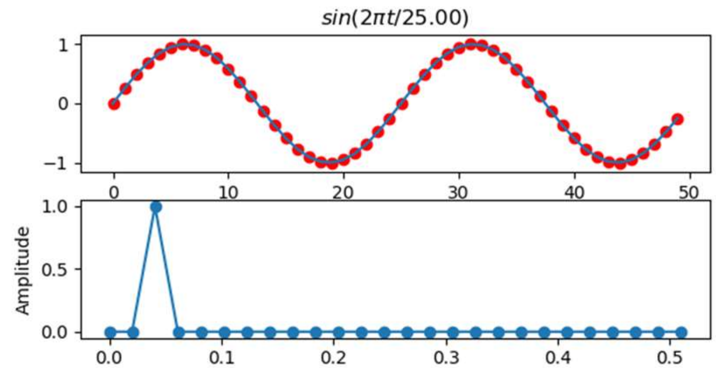
Em consequência, a função f (no domínio físico $f(t)$ ou $f(x)$) tem a mesma informação que o seu espectro F (no domínio transformado $f(\omega)$ ou $f(k_x)$).

Nota

O facto de a transformada discreta ter a mesma informação do que a série discreta finita, não implica que ela tenha toda a informação da série original (contínua e infinita). Isso só acontecerá nas condições indicadas anteriormente.

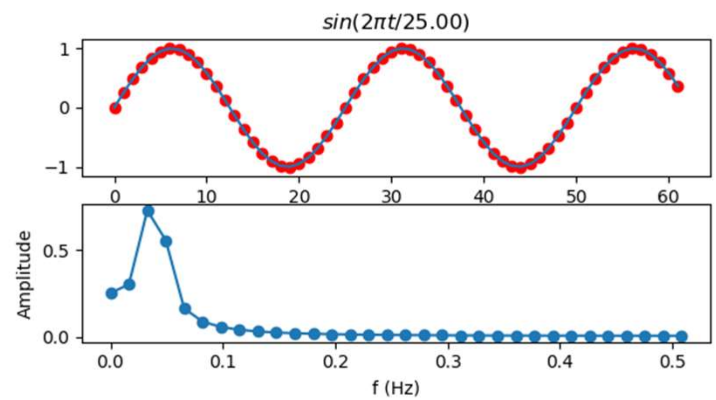
Problema do domínio

Exatamente dois periodos fundamentais.
Espectro OK
Pico em 0.04Hz=1/T



0.04

2.5 periodos fundamentais.
Espectro poluído



Mas, em ambos os casos
$$\mathcal{F}^{-1}\big(\mathcal{F}(f)\big) = f$$