

Aula 15

Séries unidimensionais e análise de Fourier

Nota: reiniciarei a sessão zoom após **40 min.** Terão de se ligar de novo

Funções (contínuas) de uma variável independente

$$V = V(t)$$

t é o tempo, mas pode ser outra variável ($x \dots$)

Amostra regular com N pontos

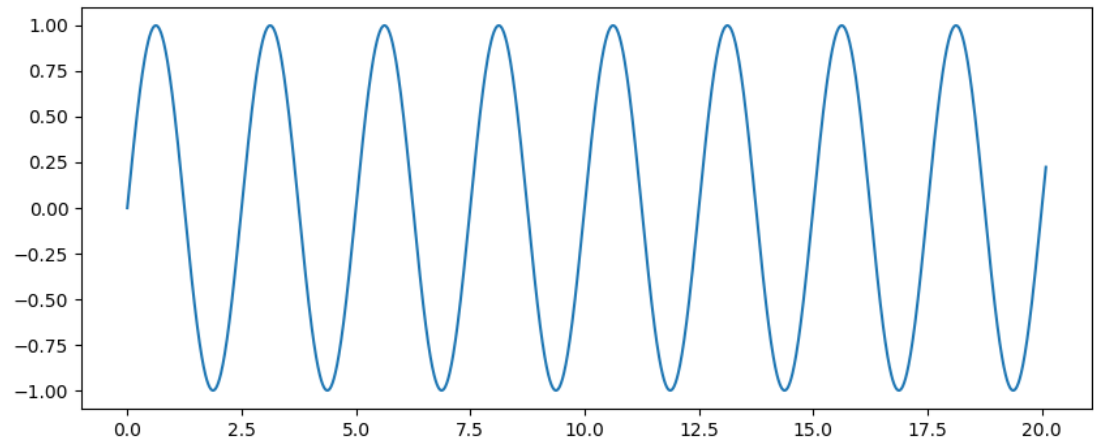
$$V_n = V(t_0 + n\Delta t), n = 0, \dots, N - 1$$

t_0 – fase inicial (amostra 0)

Δt – intervalo de amostragem (step)

Amostras de Sinusóides

$$t_0 = 0$$

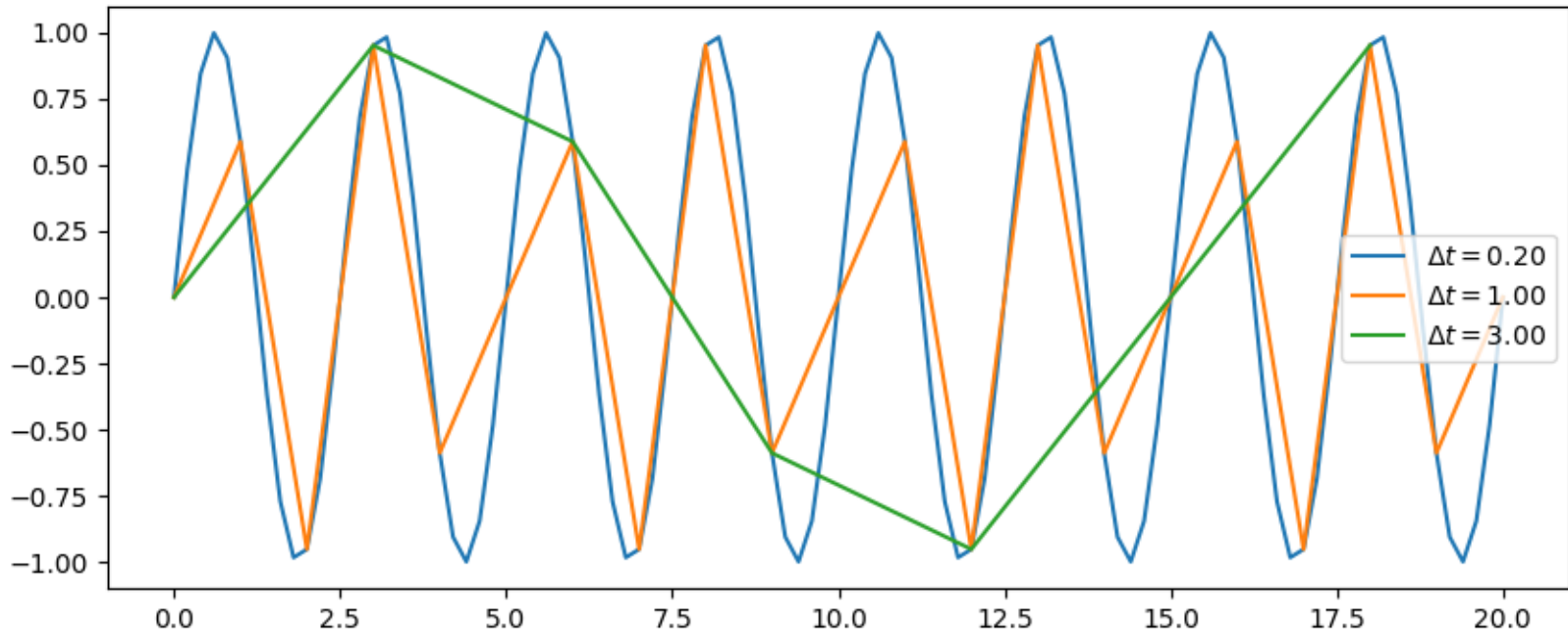


12.5 pontos por período: razoável

2.5 pontos por período: mau

0.8 pontos por período: péssimo

$$f = \sin(2\pi t/2.5)$$



Teorema da amostragem

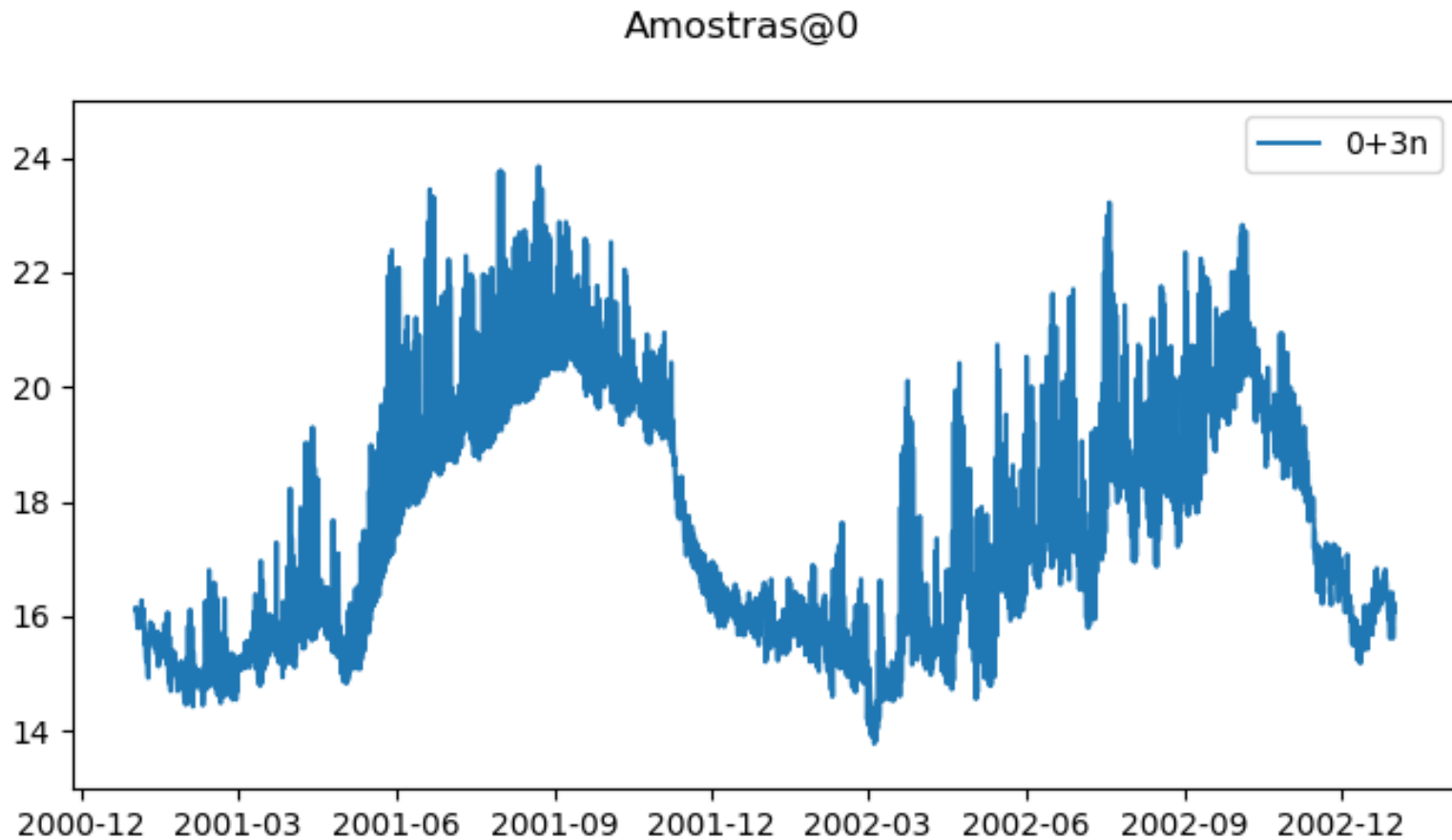
O exemplo anterior mostra que a operação de amostragem implica sempre um **erro**, mas em certos casos altera completamente a função.

No caso de uma função sinusoidal é fácil perceber que tudo depende do Número de amostras por período:

Com **menos de 2 amostras por período a função é falseada** (*aliasing*) sendo vista como um seno com um período muito mais longo : uma amostra só pode representar períodos **$T \geq 2\Delta t$**

Com pouco mais de 2 amostras por período a série apresenta **artefactos**. A **fase** também é importante.

Várias formas de amostrar uma série: **start** (n_0 , fase), **step** (Δt)



E se a função não for uma senoide?

O teorema de Fourier garante que qualquer **função periódica** pode ser obtida pela soma de sinusoides:

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

Onde (a_k, b_k) são as **amplitudes** associadas à **harmónica** k

Em geral, precisamos de **infinitas** harmónicas!

Série de Fourier na forma complexa

Utilizando a formula de Euler pode mostrar-se que:

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

Ou

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

Com os coeficientes (complexos):

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

Transformada discreta de Fourier (1)

No caso geral o teorema de Fourier não é **computável**, pois requer infinitos coeficientes, e os coeficientes são calculados por meio de um integral.

Se a função for representada exatamente com um número finito de harmônicas (**função de banda limitada**), se o intervalo de amostragem satisfizer o **teorema da amostragem** $\left(\Delta t < \frac{T_{Min}}{2} \Leftrightarrow \Delta t < \frac{1}{2f_{Max}} \right)$ e se a **amostra for suficientemente longa** para conter um período fundamental (o mais longo), a série de Fourier é **computável** e **exata**.

Transformada discreta de Fourier (2)

Transformada discreta de Fourier

$$F_k = \sum_{n=0}^{N-1} f_n e^{-2\pi i n k / N}$$

*transformada discreta **inversa** de Fourier*

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

Código naïve (lento)

```
import numpy as np;import matplotlib.pyplot as plt
def dFT(y): #transformada discreta de Fourier direta
    i=complex(0,1.);pi=np.pi
    N=len(y)
    z=np.zeros(y.shape,dtype=complex)
    for k in range(N):
        for j in range(N):
            z[k]=z[k]+y[j]*np.exp(-2*pi*i*j*k/N)
    return z
def iFT(y): #transformada discreta de Fourier inversa
    i=complex(0,1.);pi=np.pi
    N=len(y)
    z=np.zeros(y.shape,dtype=complex)
    for k in range(N):
        for j in range(N):
            z[k]=z[k]+y[j]*np.exp(2*pi*i*j*k/N)
    return z/N
```

```

N=100; dt=1.; T=N*dt/4.;
t=np.linspace(0, dt*(N-1), N);
f=np.sin(2*np.pi*t/T)
plt.subplot(3,1,1); plt.plot(t, f, label='f')
F=dFT(f);
fNyq=1/(2*dt) #frequência de Nyquist
df=2*fNyq/(N-1) #resolução espectral
freq=np.zeros(t.shape)
freq[0:N//2+1]=np.arange(0, fNyq+df, df)
if N%2==0:
    freq[N//2+1:N]=np.arange(-fNyq+df, 0, df)
else:
    freq[N//2+1:N]=np.arange(-fNyq, 0, df)
left=range(N//2+1, N)
right=range(0, N//2+1)

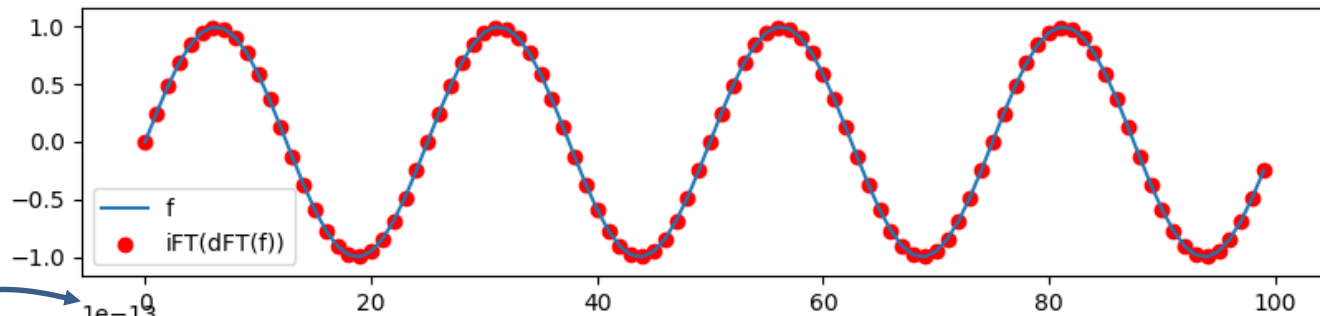
```

```

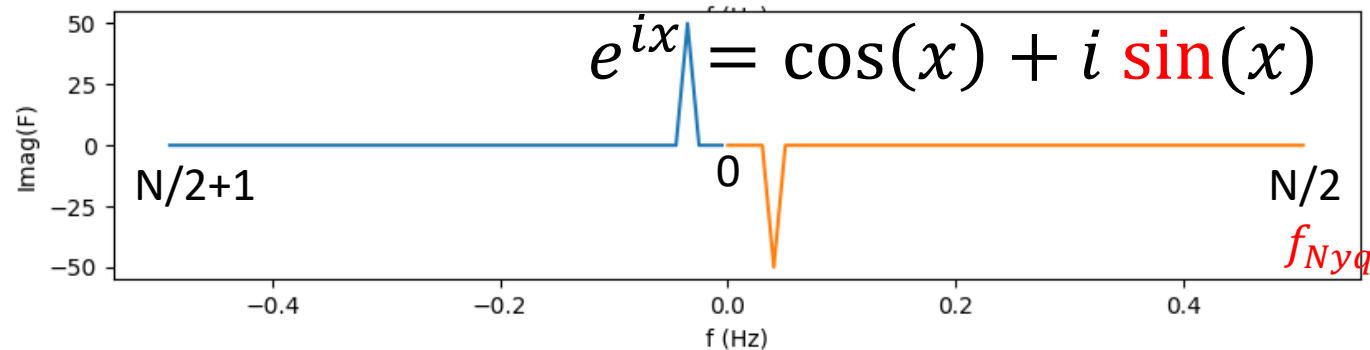
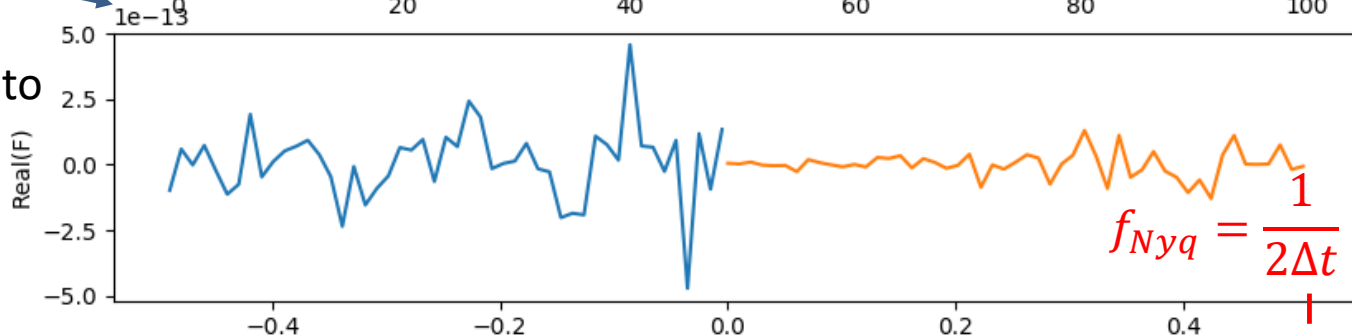
plt.subplot(3,1,2);
plt.plot(freq[left],np.real(F[left]))
plt.plot(freq[right],np.real(F[right]))
plt.ylabel('Real(F)');plt.xlabel('f (Hz)')
plt.subplot(3,1,3);
plt.plot(freq[left],np.imag(F[left]))
plt.plot(freq[right],np.imag(F[right]));
plt.ylabel('Imag(F)'); plt.xlabel('f (Hz)')
ff=iFT(F)
plt.subplot(3,1,1);
plt.scatter(t,np.real(ff),color='red',\
            label='iFT(dFT(f))')
plt.legend()

```

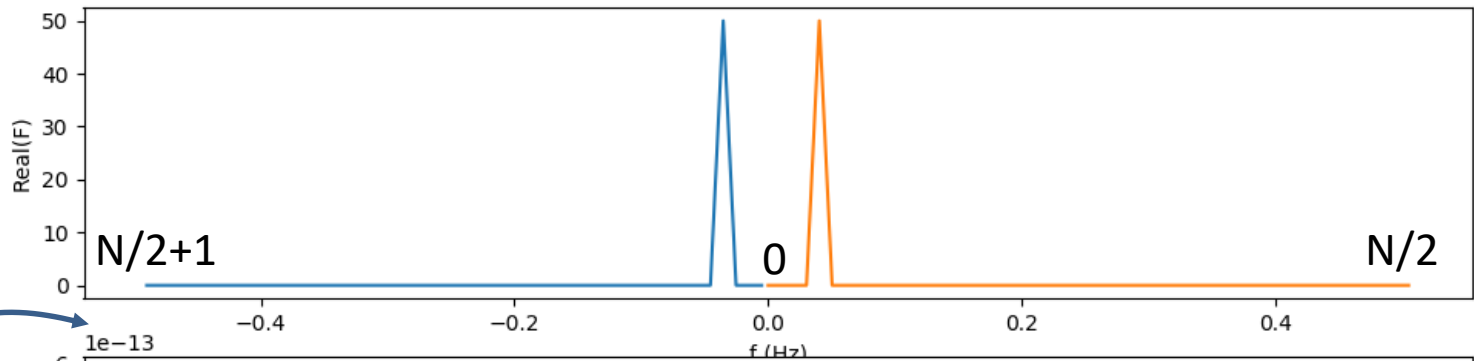
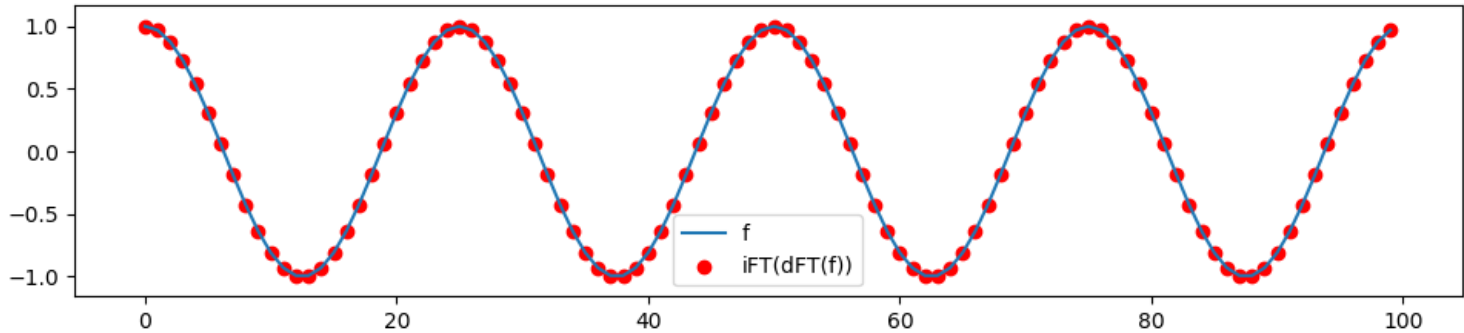
Transformada de $\sin\left(\frac{2\pi t}{25}\right)$



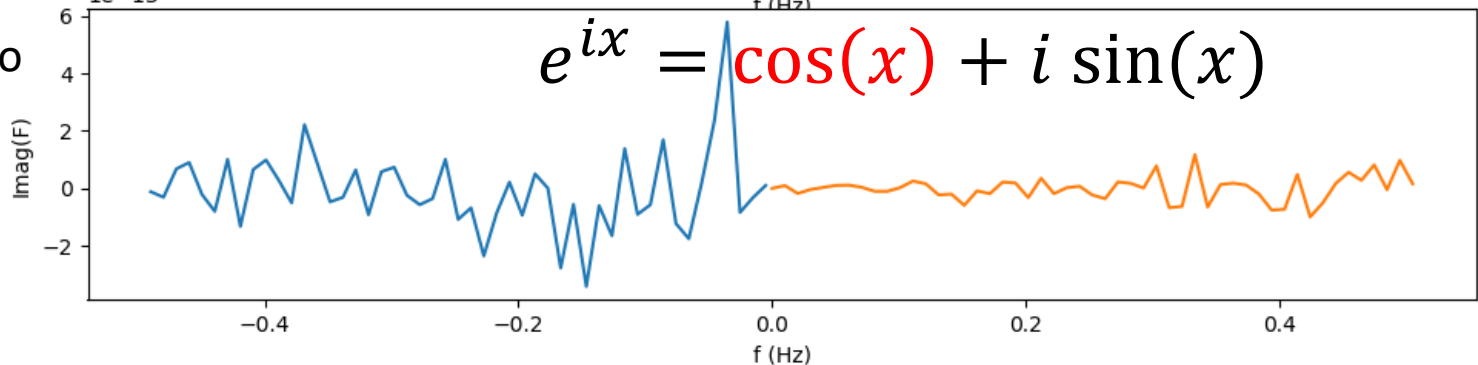
Erro arredondamento



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



Erro arredondamento



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)

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

Onde $\omega = 2\pi f$

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

Fast Fourier Transform $N = 2^k 3^m 5^n$

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)
```

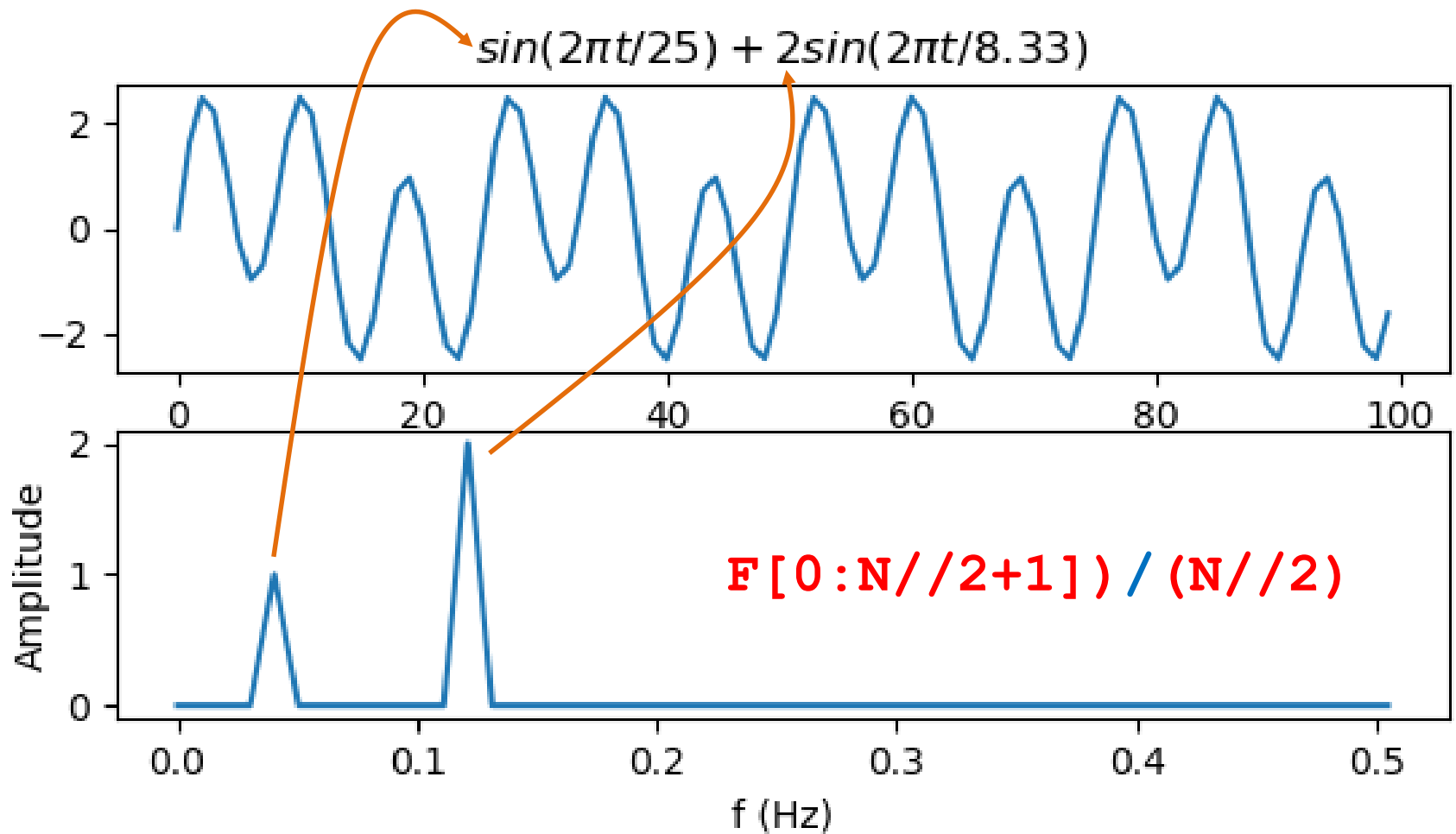
Transformada inversa

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


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.fft.fft(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))
plt.ylabel('Amplitude');plt.xlabel('f (Hz)')
```

fft da soma de dois senos



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 $f(k_x)$).

Notas

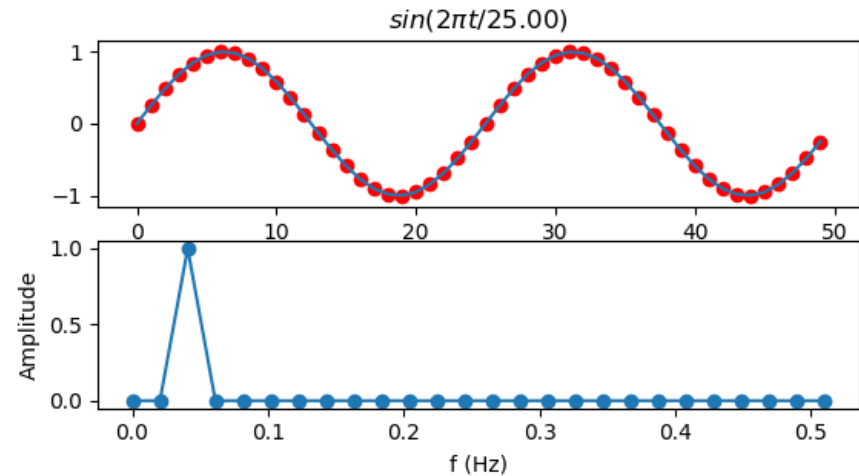
O facto de a transformada discreta ter a mesma informação do que a série discreta finita, não implica que ela tenha toda a informação da série original (contínua e infinita). Isso só acontecerá nas condições indicadas anteriormente.

Problema do domínio

Exatamente dois períodos
fundamentais.

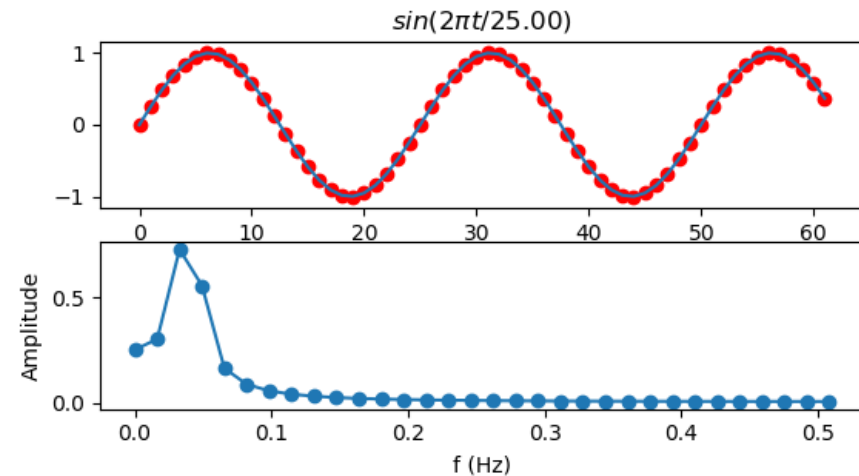
Espectro OK

Pico em $0.04\text{Hz}=1/T$



2.5 períodos fundamentais.

Espectro poluído



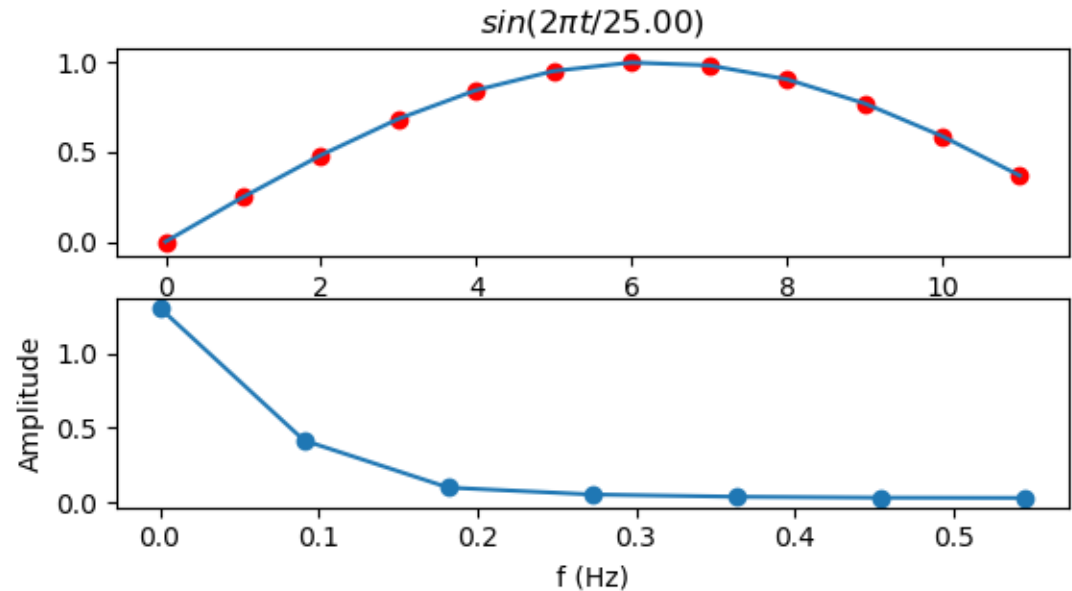
Mas, em ambos os casos

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

Domínio inferior ao período fundamental

Mas,

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



Pico em 0Hz (média
mão nula)

Com dados reais

Existem muitas harmónicas e podemos não saber qual a **harmónica fundamental**.

É preciso ter bom senso e ensaiar diferentes aproximações.

Por exemplo, se se tratar de uma série climática (em que existe um período anual obrigatório) deve analisar-se um período que corresponda a anos completos, ou a dias completos (não cortar o período diurno)