

# **TCV 2026.0**

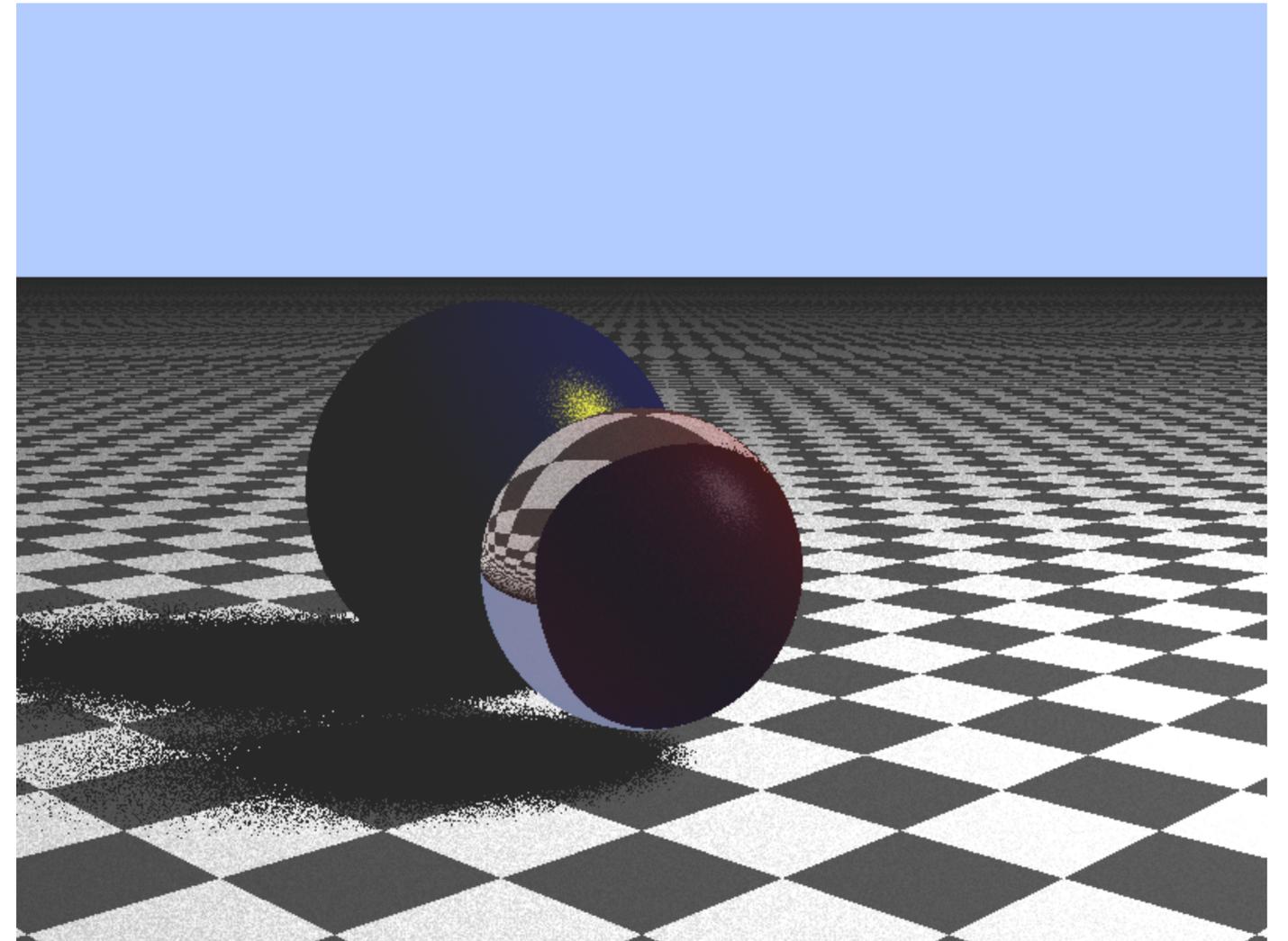
**Images, Amostragem e Reconstrução, Vector Graphics**

**Francisco Ganacim <[ganacim@impa.br](mailto:ganacim@impa.br)>**

# Objetivos



2D



3D

# Código da tarefa 1

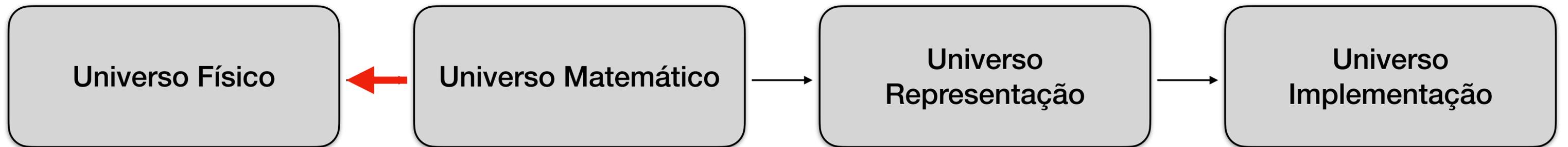


2D



[https://github.com/ganacim/tcv\\_raster\\_2026](https://github.com/ganacim/tcv_raster_2026)

# Paradigma dos 4 universos



[https://en.wikipedia.org/wiki/Pillars\\_of\\_Creation](https://en.wikipedia.org/wiki/Pillars_of_Creation)

$$I(x, y)$$

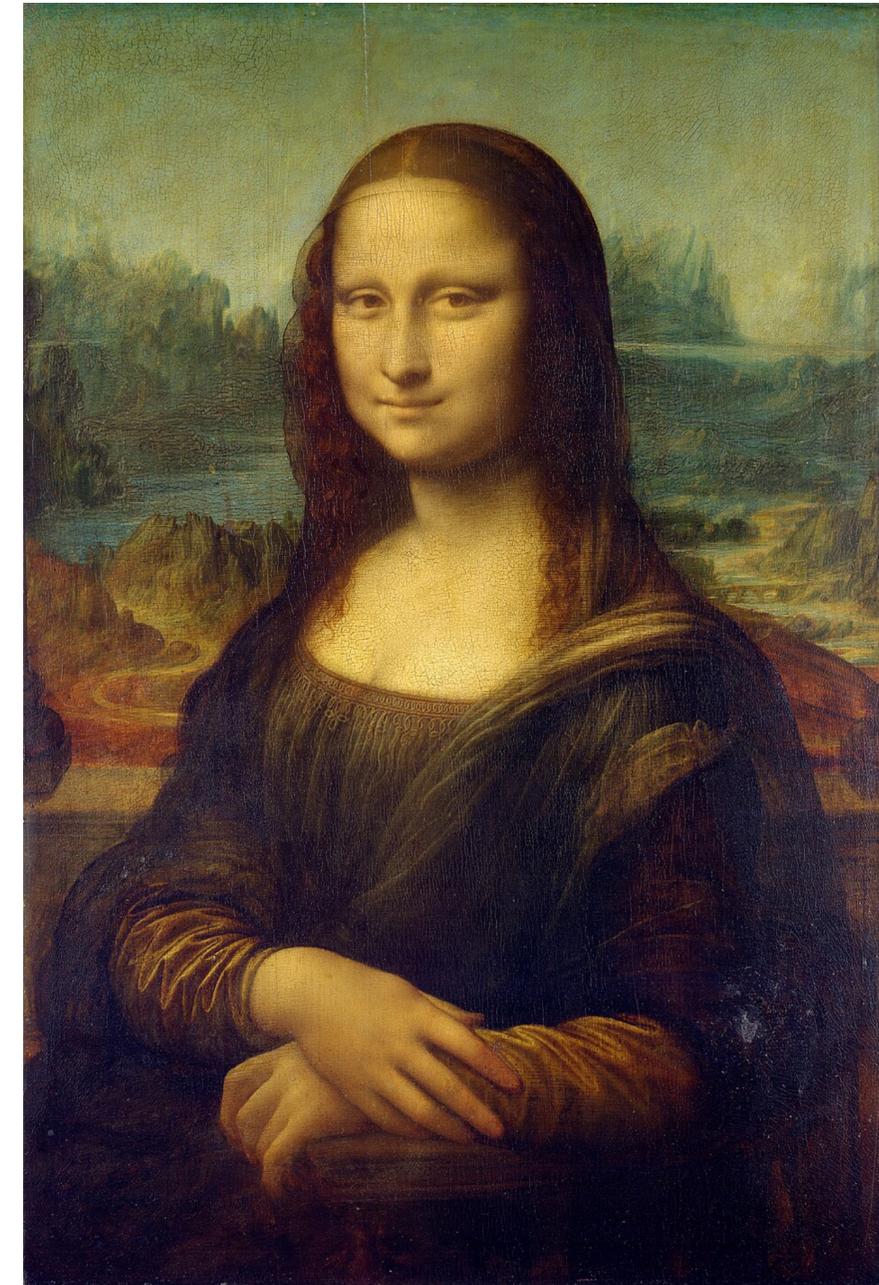
$$\begin{bmatrix} 12 & 44 & \dots & 201 \\ 83 & 156 & \dots & 9 \\ \vdots & \vdots & \ddots & \vdots \\ 230 & 17 & \dots & 98 \end{bmatrix}$$

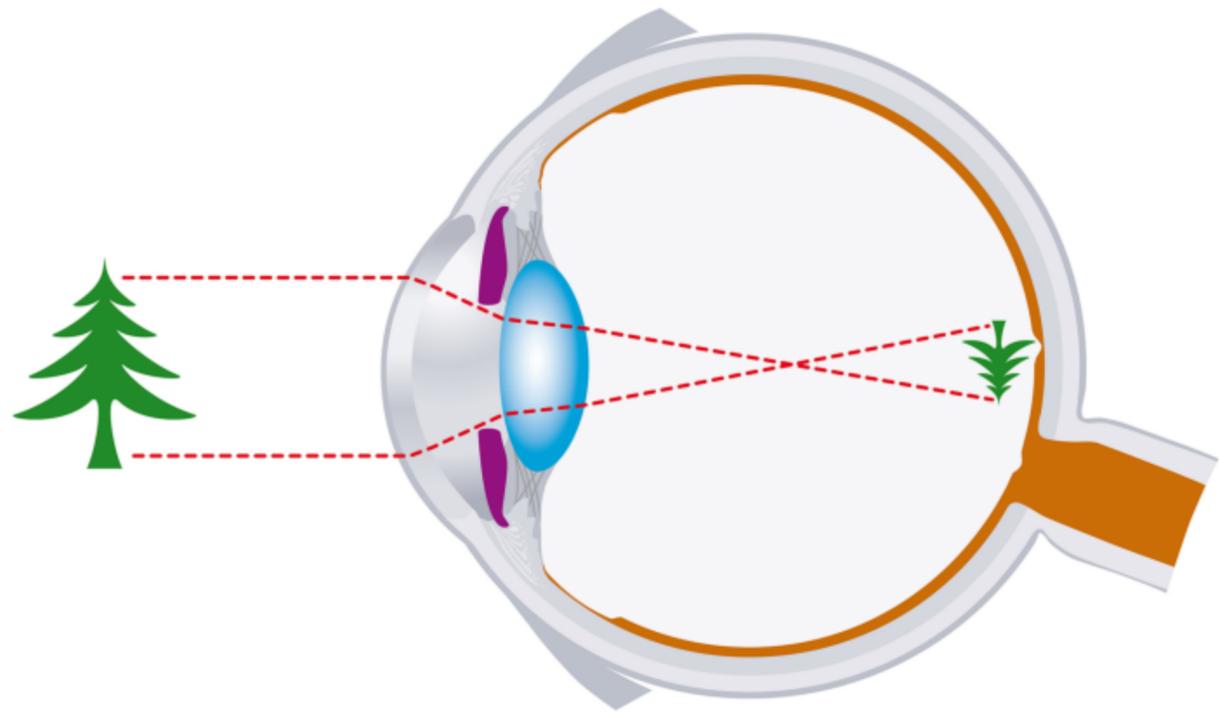
```
P3
# PPM File Format
1024 1024
255
255 0 0
0 255 0
0 0 255
255 255 0
255 255 255
0 0 0
...
```

# Universo Físico

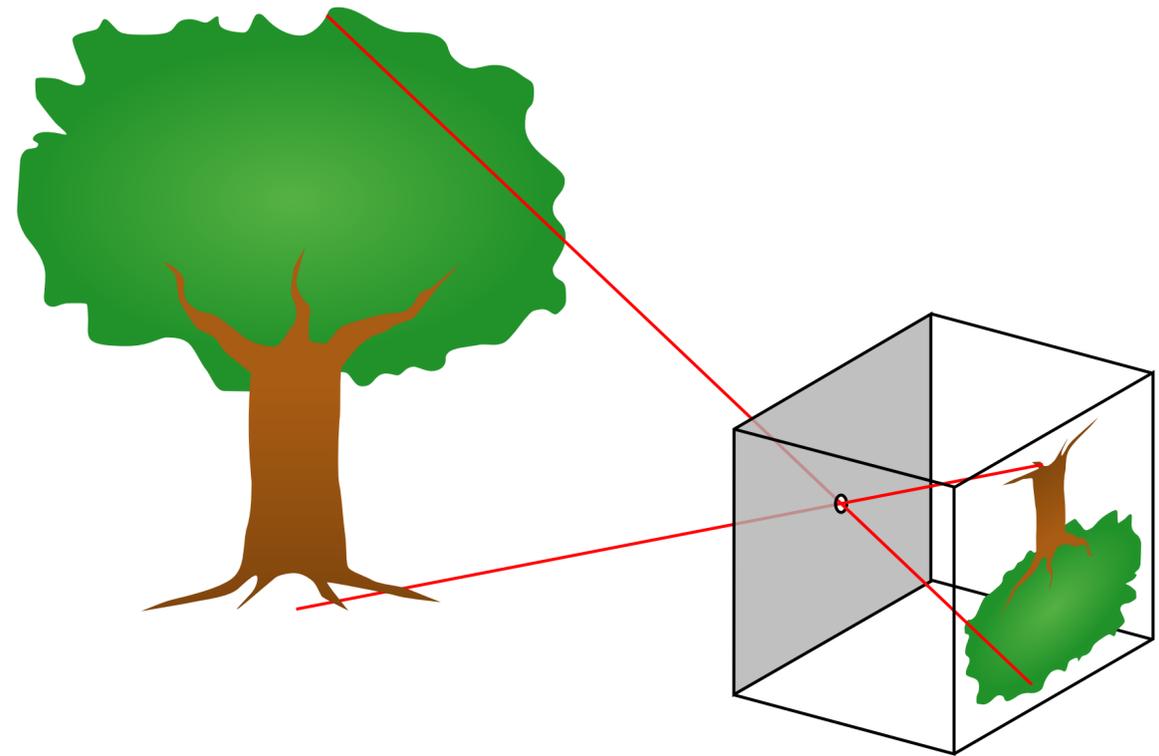


<https://en.wikipedia.org/wiki/File:SantaCruz-CuevaManos-P2210651b.jpg>



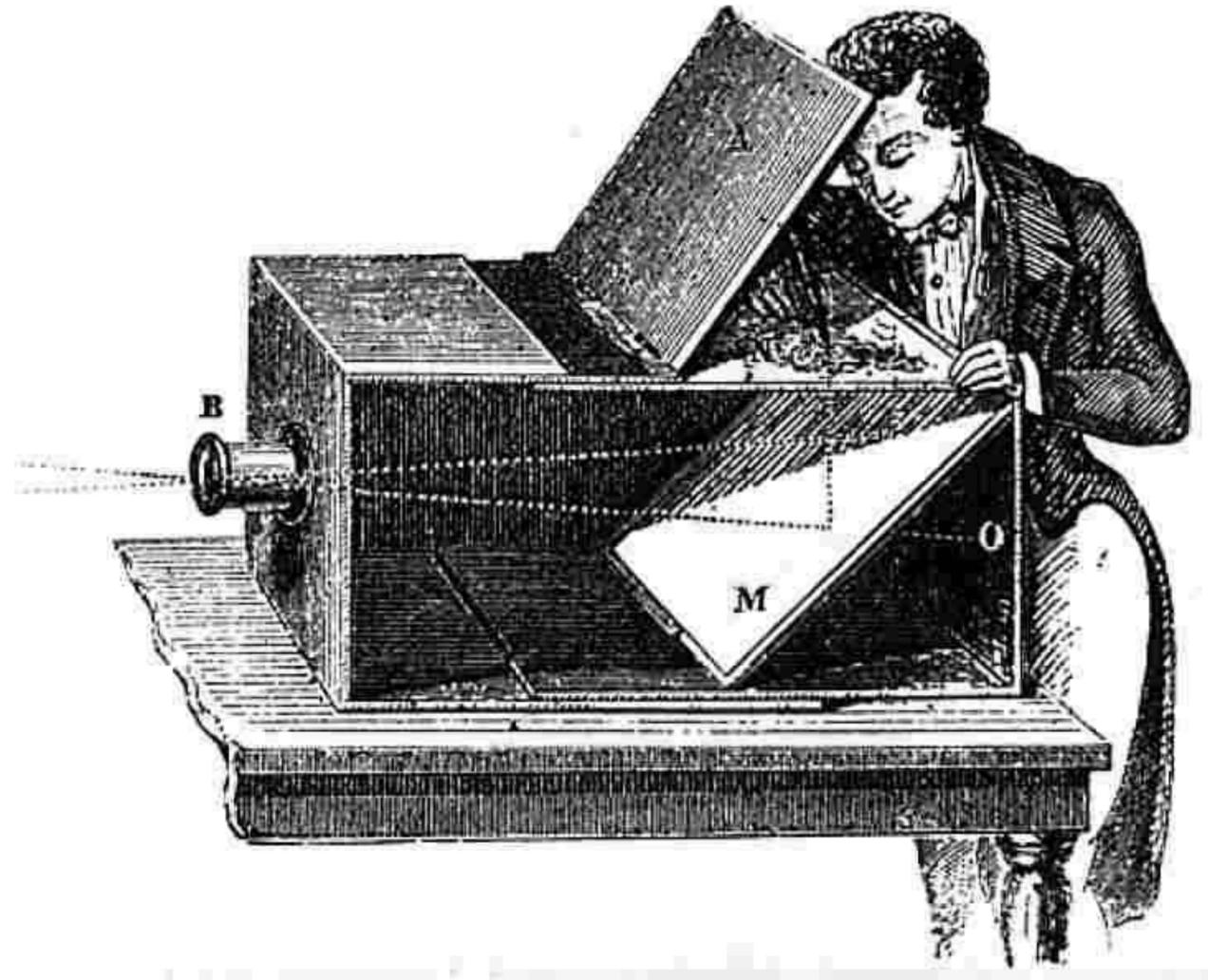


[https://www.edplace.com/worksheet\\_info/science/keystage3/year9/topic/1290/13730/explain-how-lenses-form-images](https://www.edplace.com/worksheet_info/science/keystage3/year9/topic/1290/13730/explain-how-lenses-form-images)



[https://en.wikipedia.org/wiki/Pinhole\\_camera#/media/File:Pinhole-camera.svg](https://en.wikipedia.org/wiki/Pinhole_camera#/media/File:Pinhole-camera.svg)

# Camara Obscura



# Point de vue du Gras (Vista da Janela em Le Gras)



Considerada a fotografia mais antiga ainda existente

# Point de vue du Gras (Vista da Janela em Le Gras)

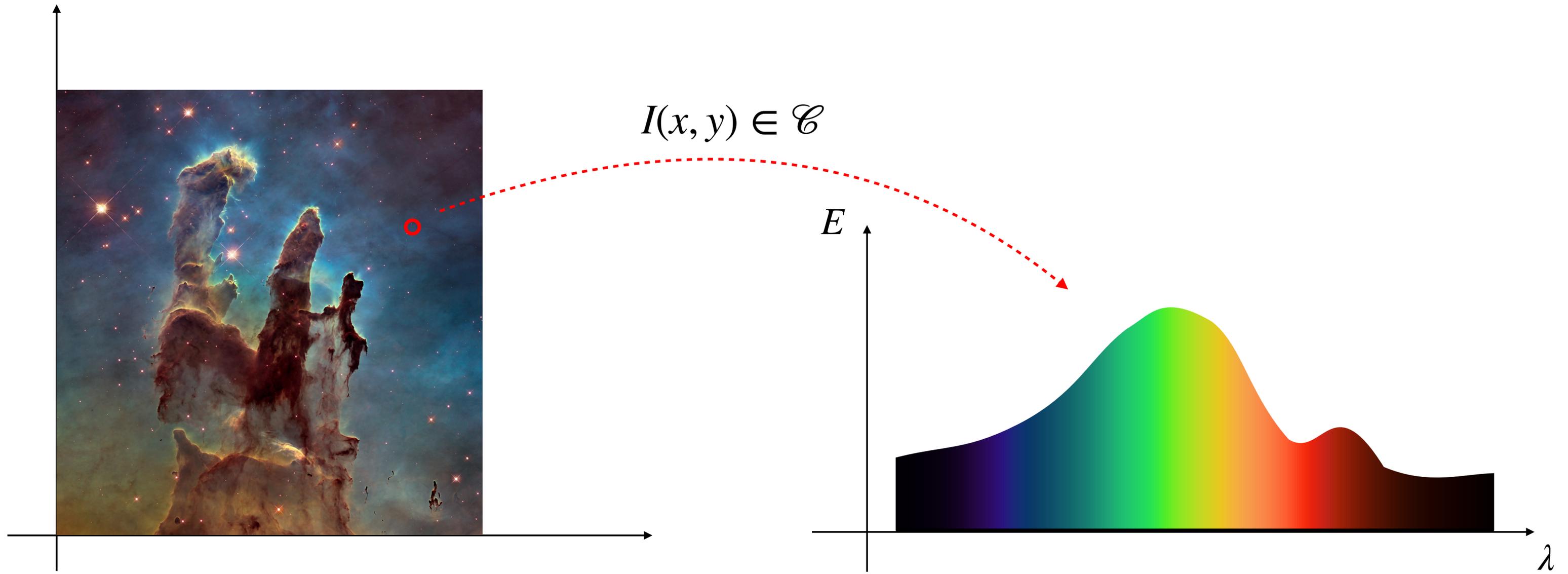


Considerada a fotografia mais antiga ainda existente

# Definição Ingênua

"Imagem é a captura de um sinal luminoso em um anteparo bidimensional"

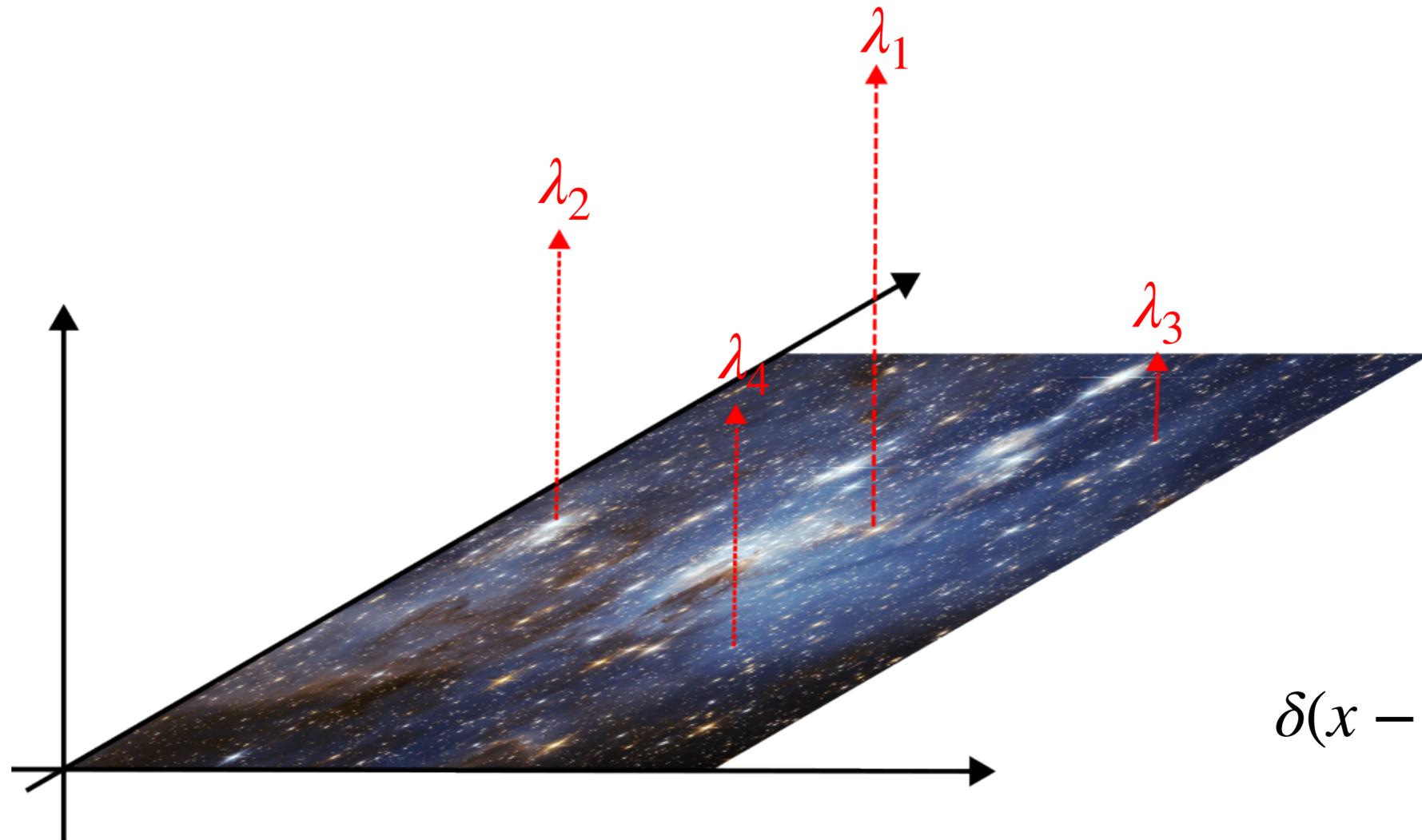
# Universo Matemático



# Onde Isso Falha?



# Estrelas são deltas de Dirac



$$I(x, y) = \sum_{i=1}^N \lambda_i \delta(x - x_i, y - y_i)$$

Onde

$$\{(x_1, y_1), \dots, (x_N, y_N)\}$$

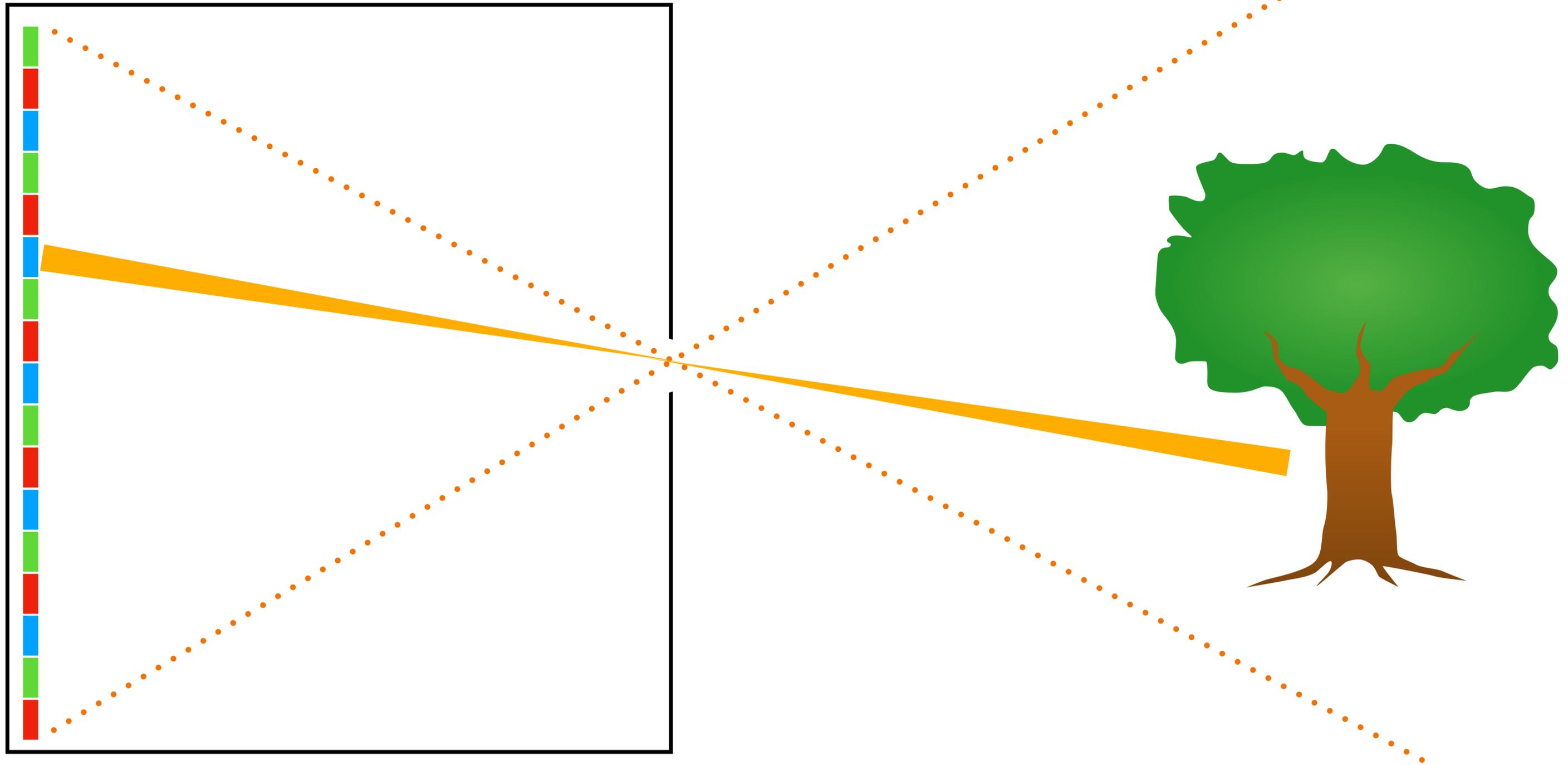
são centros

$$\{\lambda_1, \dots, \lambda_N\}$$

$$\iint_{\mathbb{R}^2} \lambda_i \delta(x - x_i, y - y_i) dx dy = \lambda_i$$

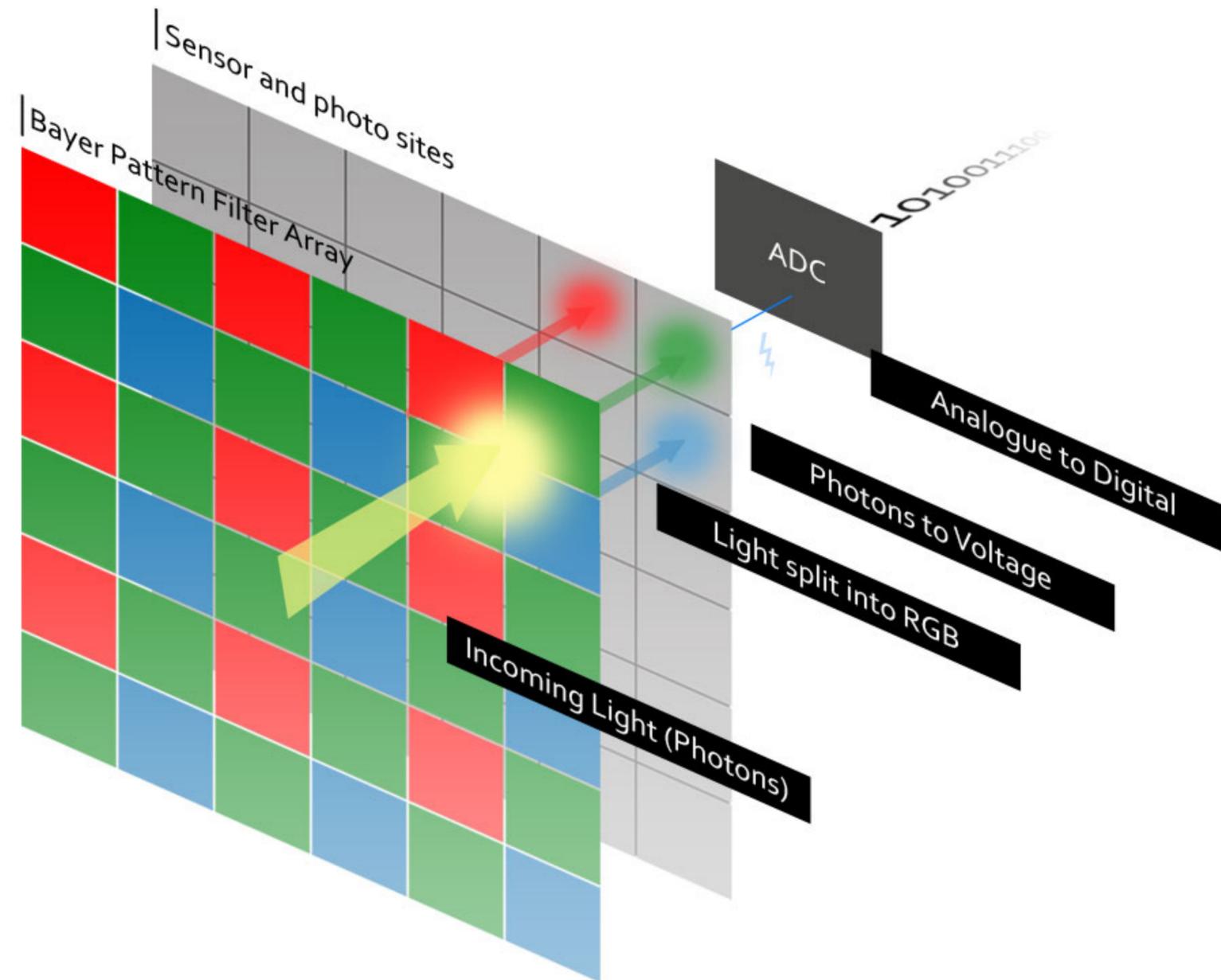
$$\delta(x - x_i, y - y_i) = \begin{cases} +\infty & (x, y) = (x_i, y_i) \\ 0 & (x, y) \neq (x_i, y_i) \end{cases}$$

# Solução: Tirar Médias



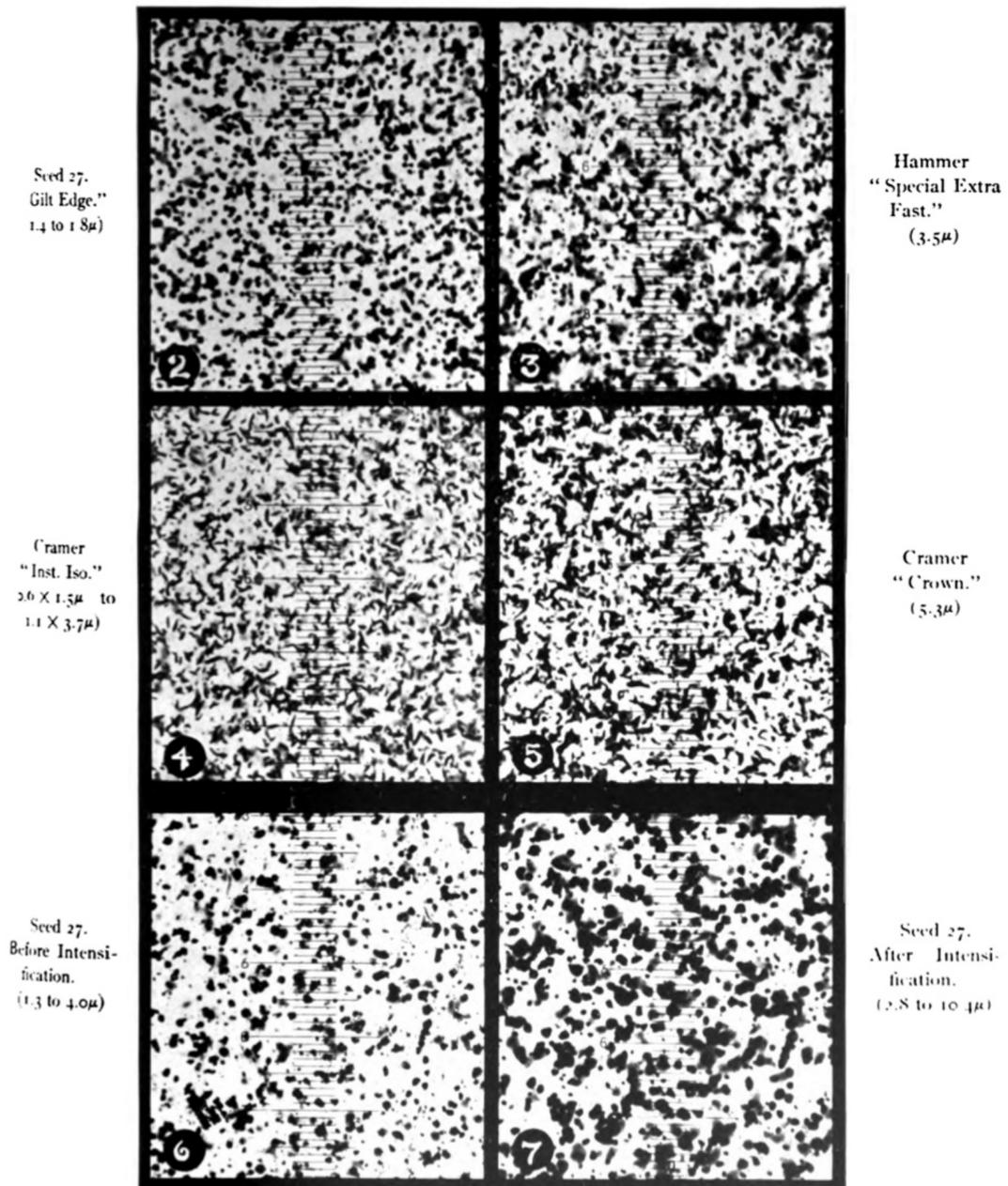
# Solução: Tirar Médias

Camera Imaging Sensor  
with Bayer Filter



# Solução: Tirar Médias

PLATE VII.



PHOTOMICROGRAPHS OF SILVER "GRAIN."



# Qual é o modelo mais adequado?

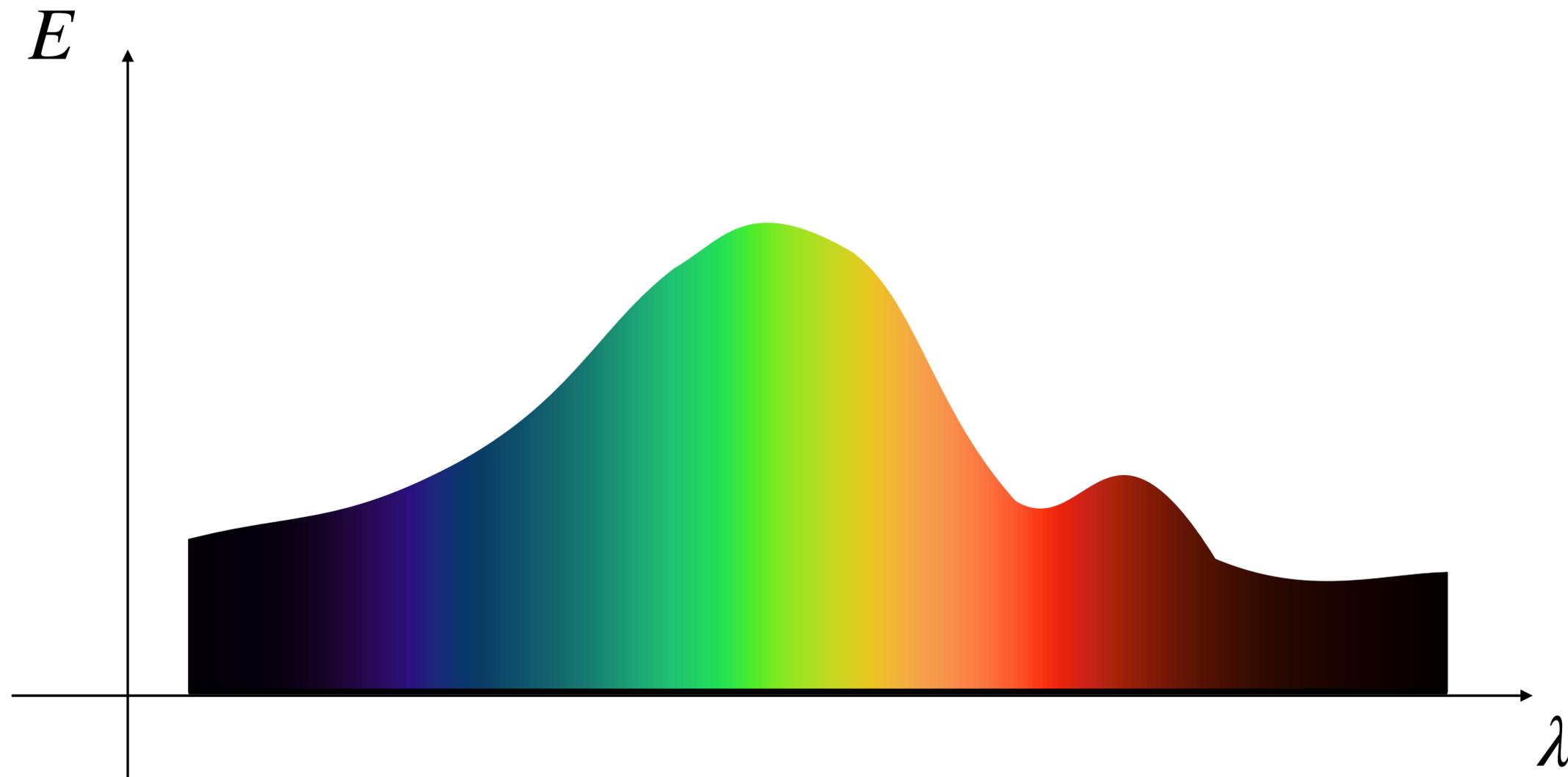


Uma *imagem* é uma função

$$I : \Omega \subset \mathbb{R}^2 \rightarrow \mathcal{C}$$

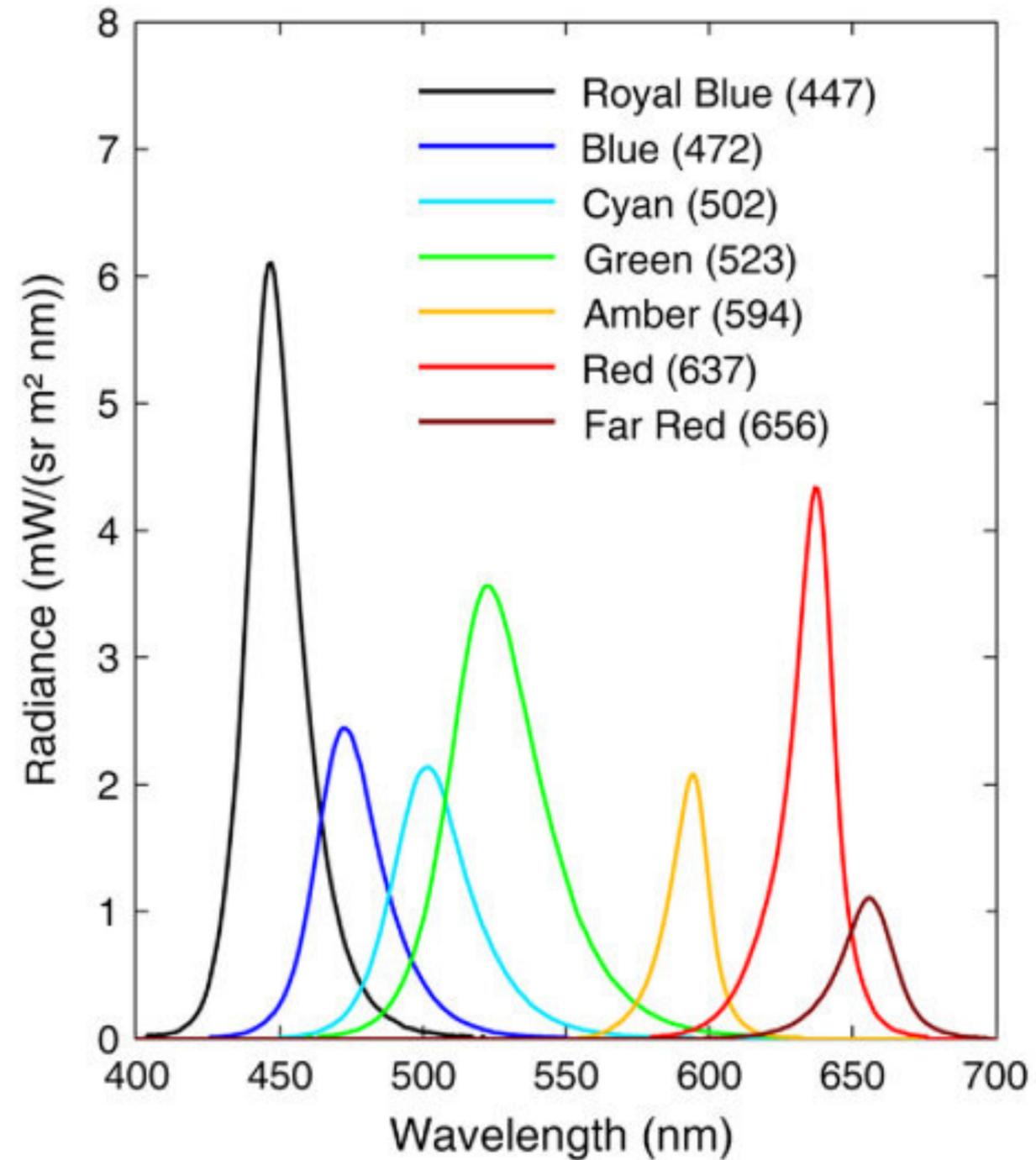
onde  $\Omega$  é o suporte da imagem e  $\mathcal{C}$  é o *conjunto de cores*

# O que é cor?

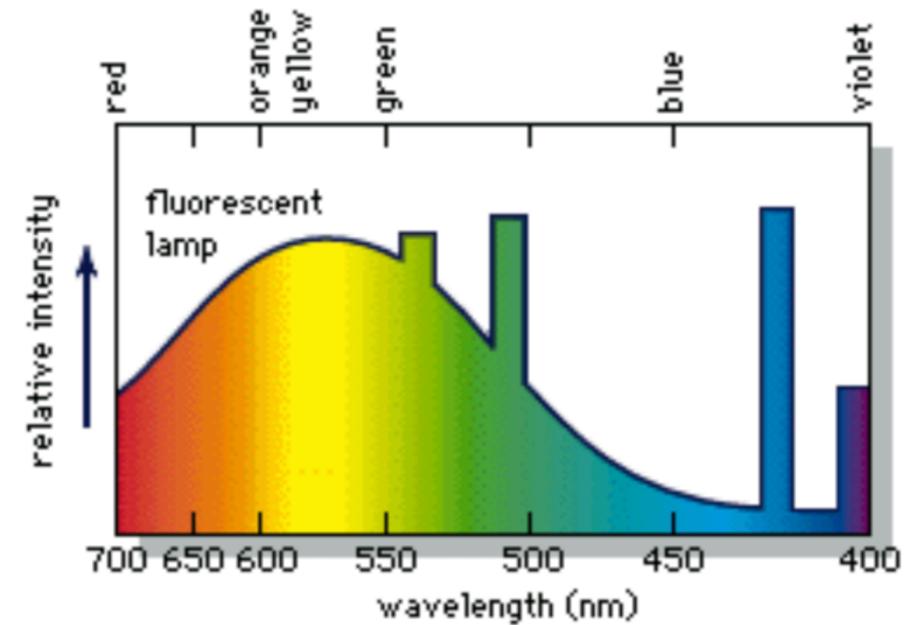
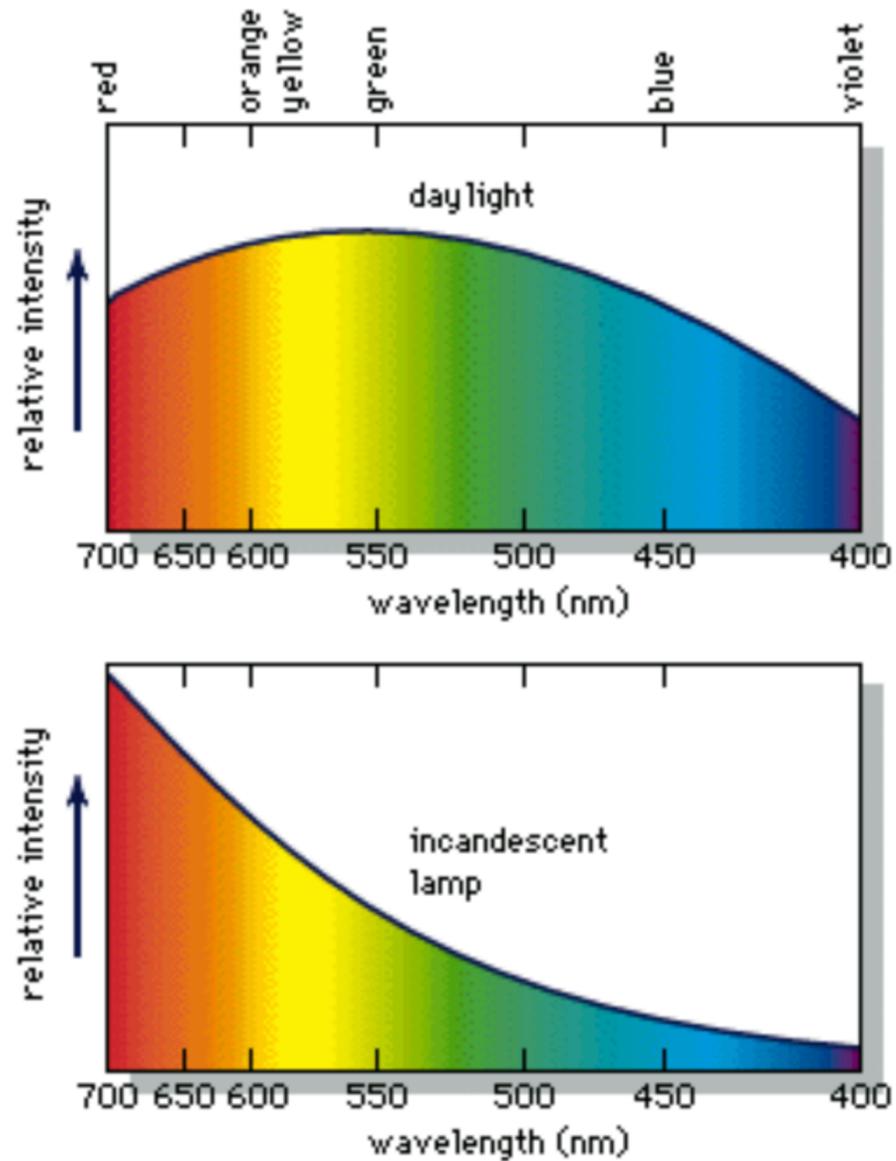


**Distribuição** de energia no espectro eletromagnético visível

# Espectro de algumas cores

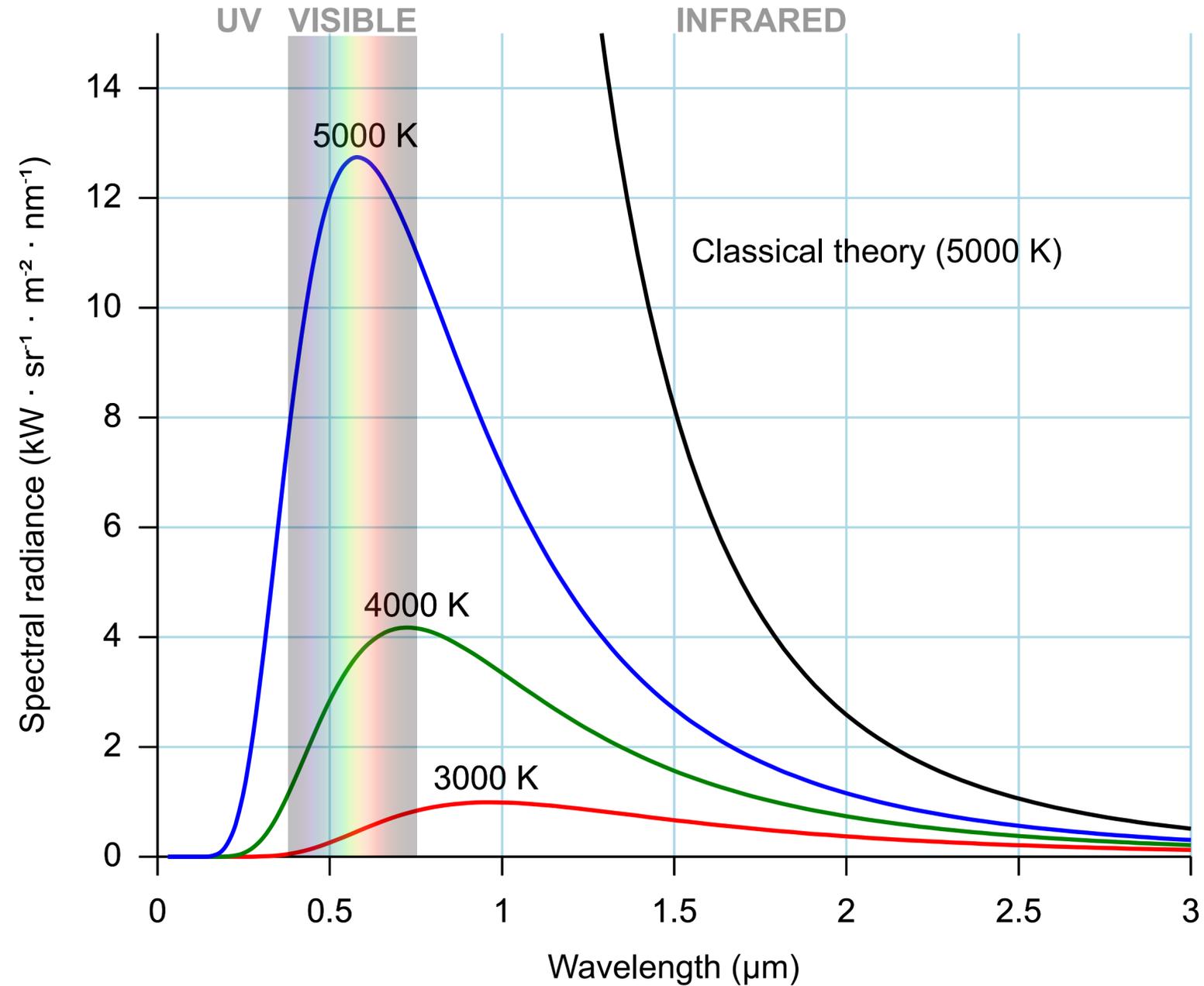


# Espectro de fontes de luz

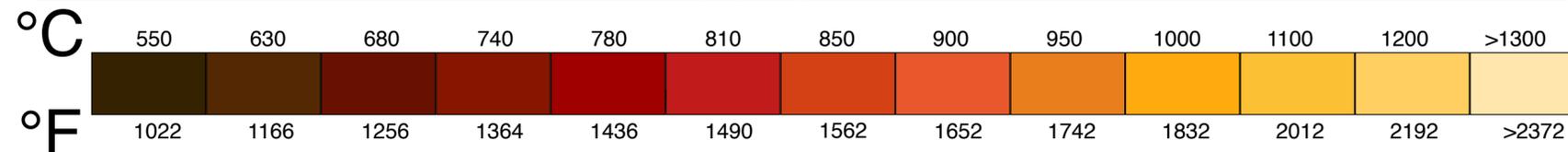


©1994 Encyclopaedia Britannica, Inc.

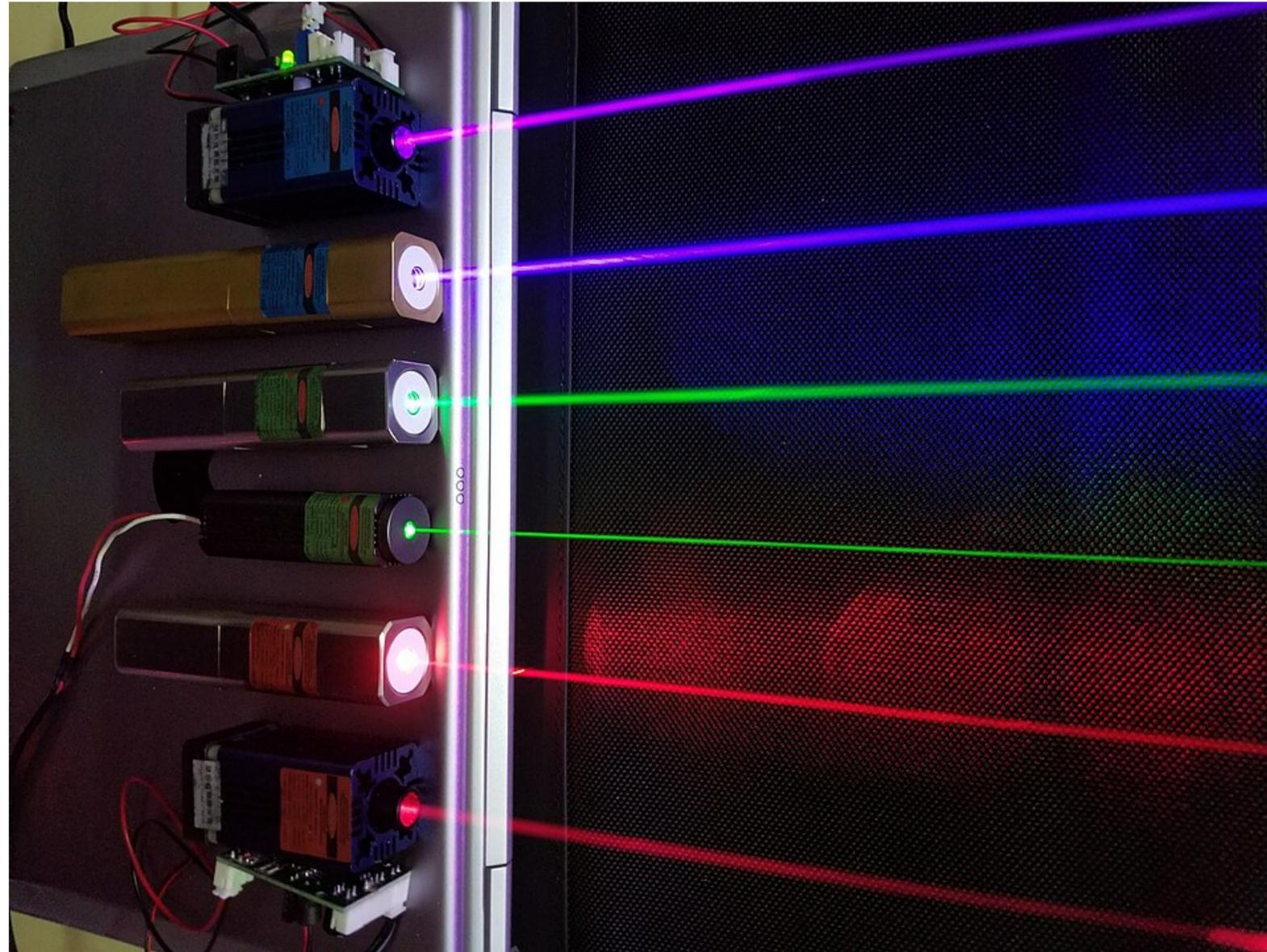
# Radiação de corpo negro



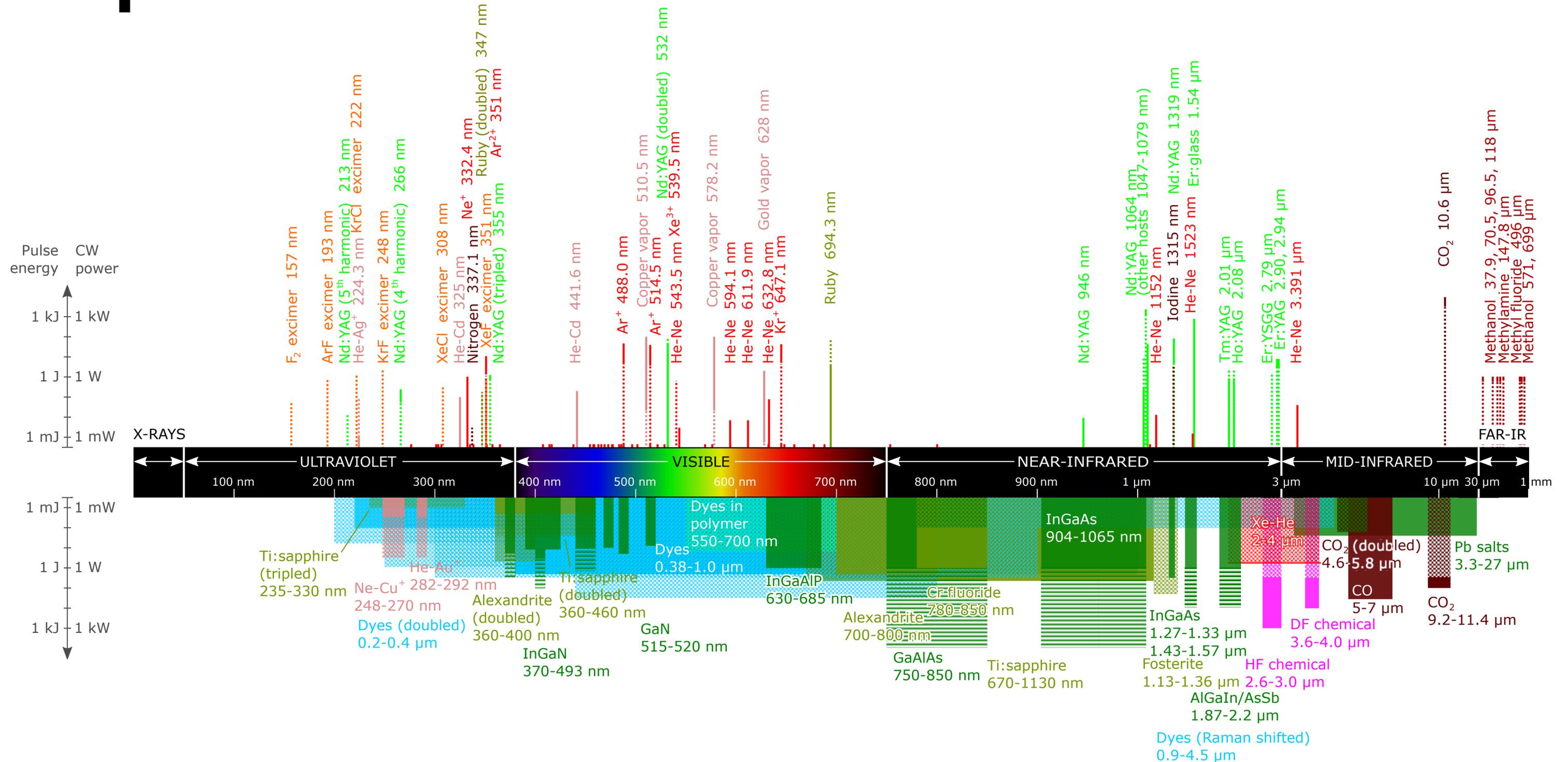
# Radiação de corpo negro



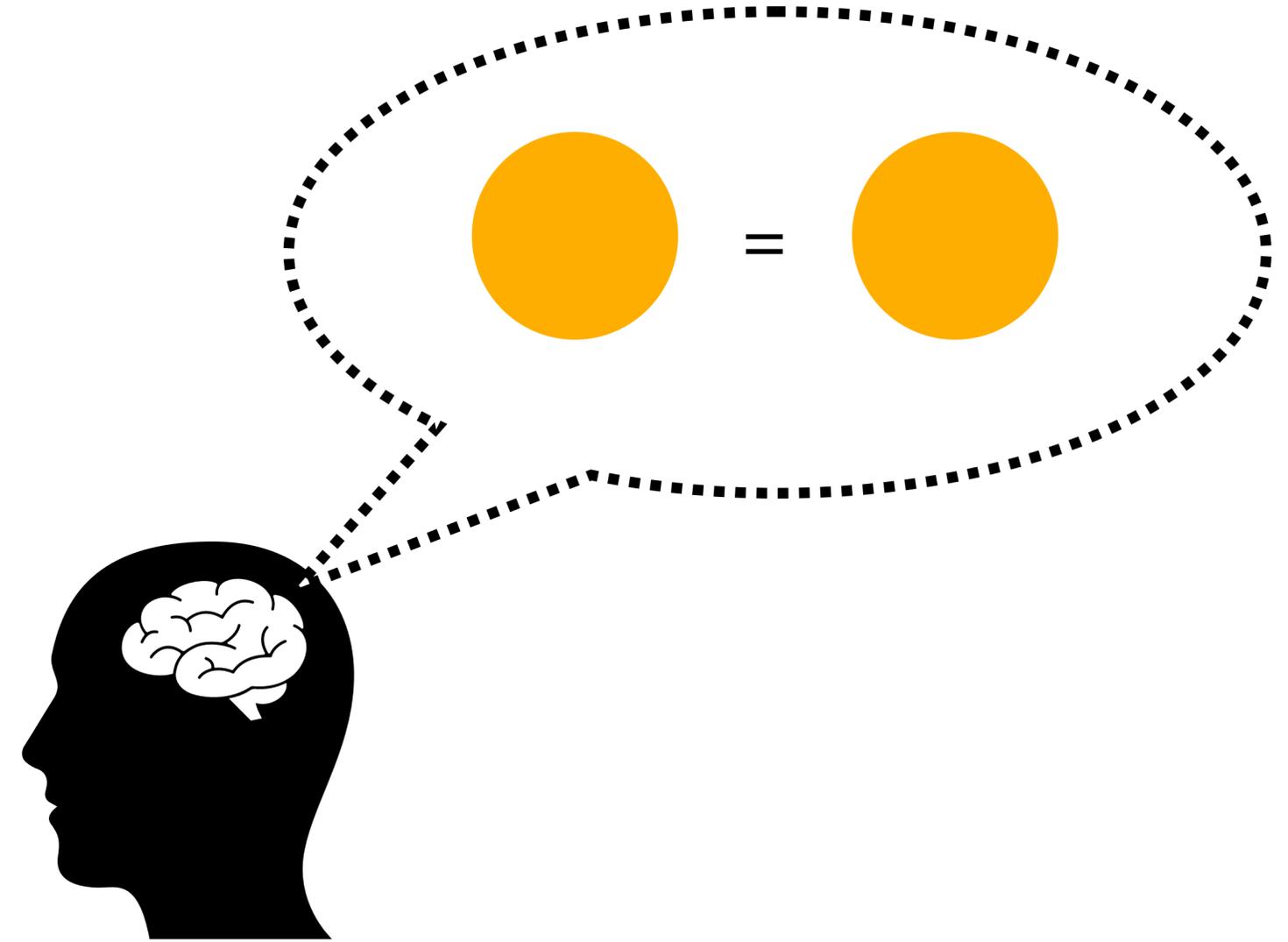
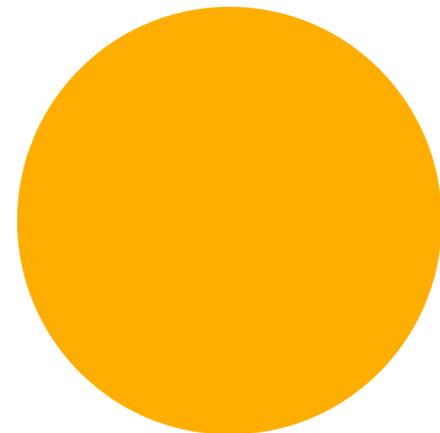
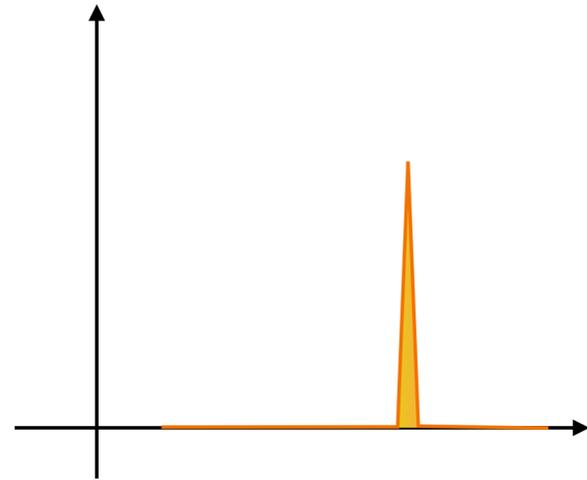
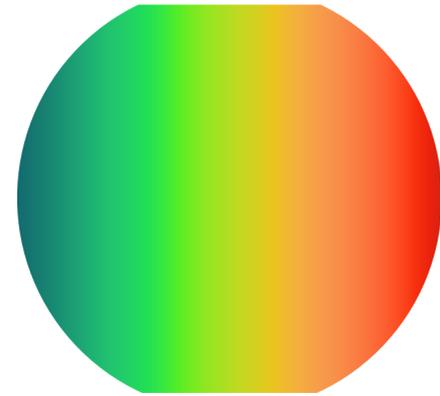
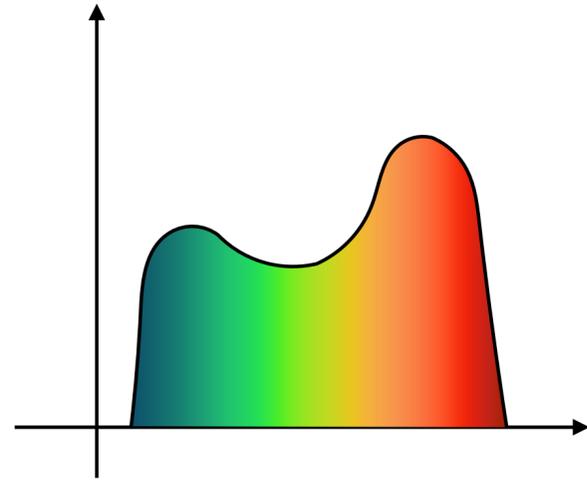
# O que é um Laser?



# O que é um Laser?

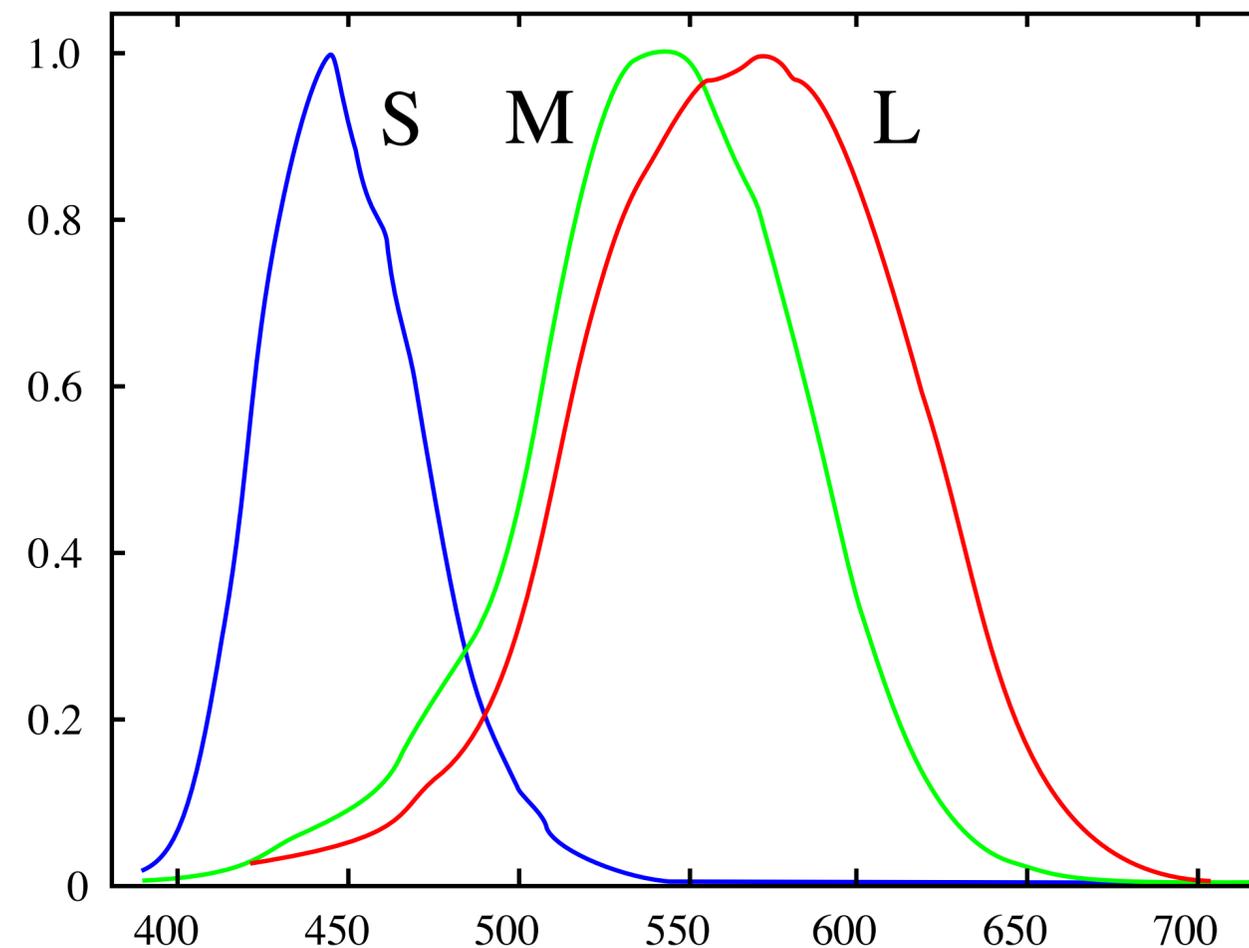


# Cores Metaméricas



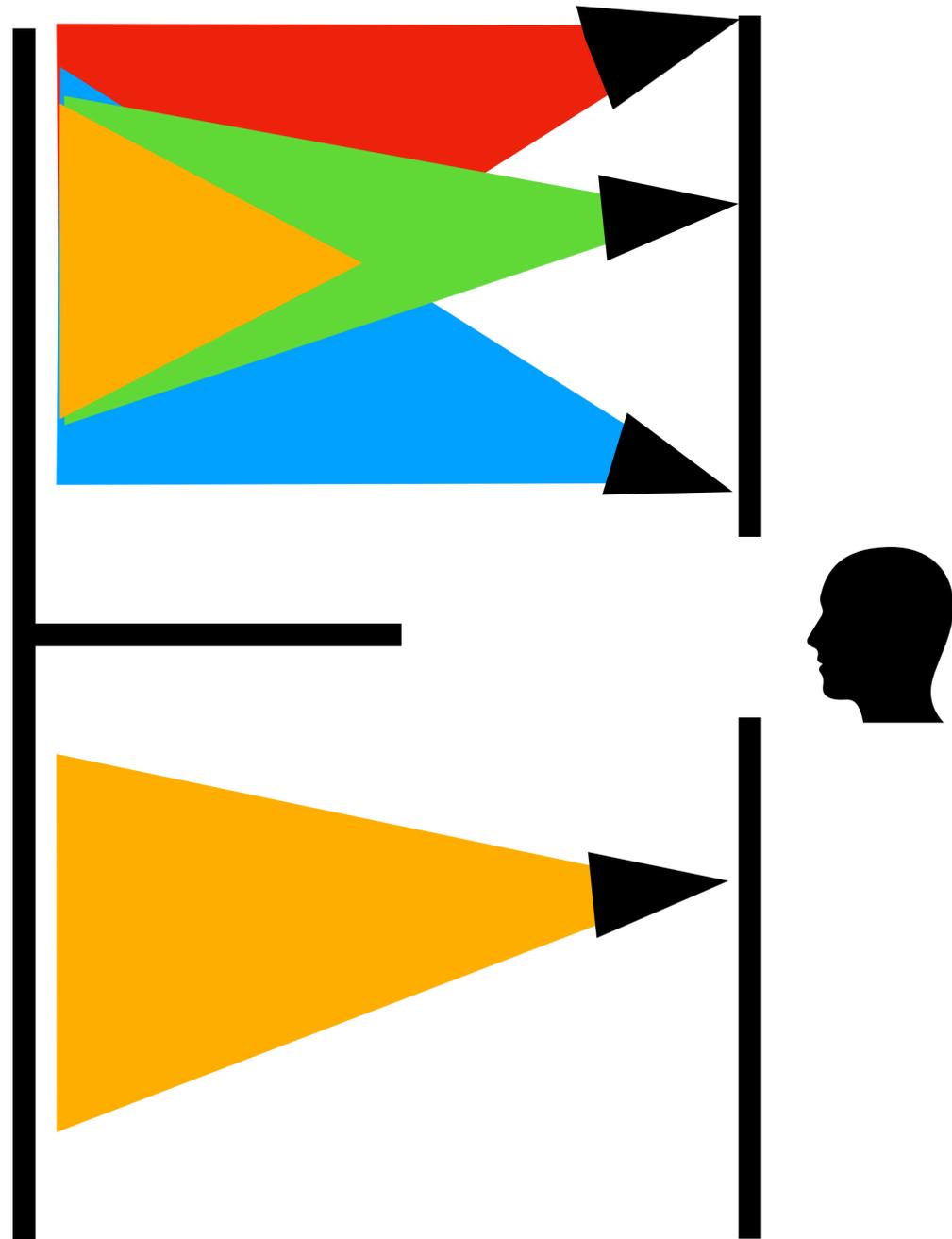
# Cores Metaméricas

Duas cores são metaméricas quando têm espectros distintos, porém são percebidas igualmente

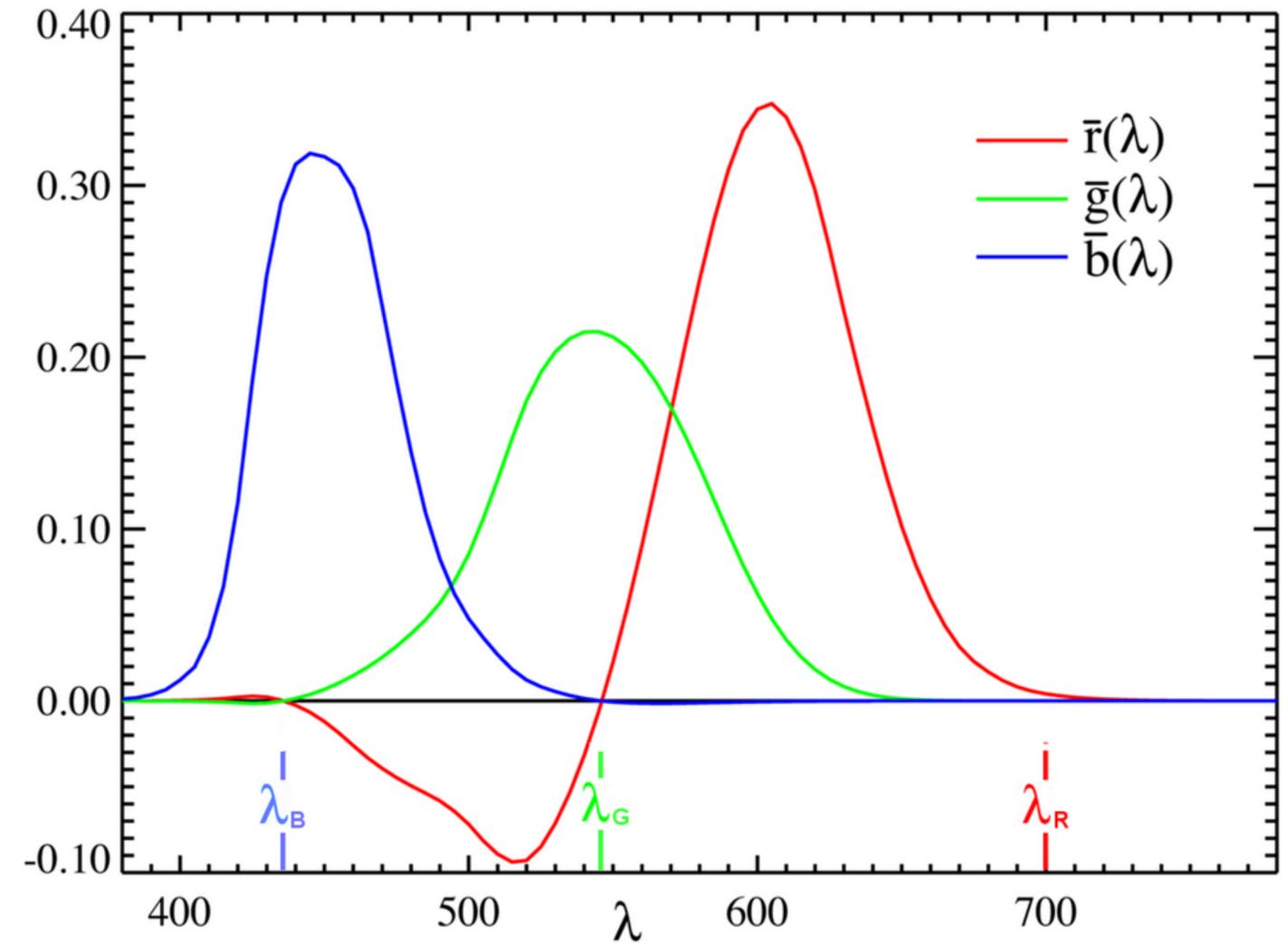


Sensibilidade espectral dos cones humanos

# Color-matching functions



CIE 1931 RGB color-matching functions



# Cores em RGB

Dada um distribuição espectral  $C(\lambda)$ ,  
sua representação tri-estímulo RGB

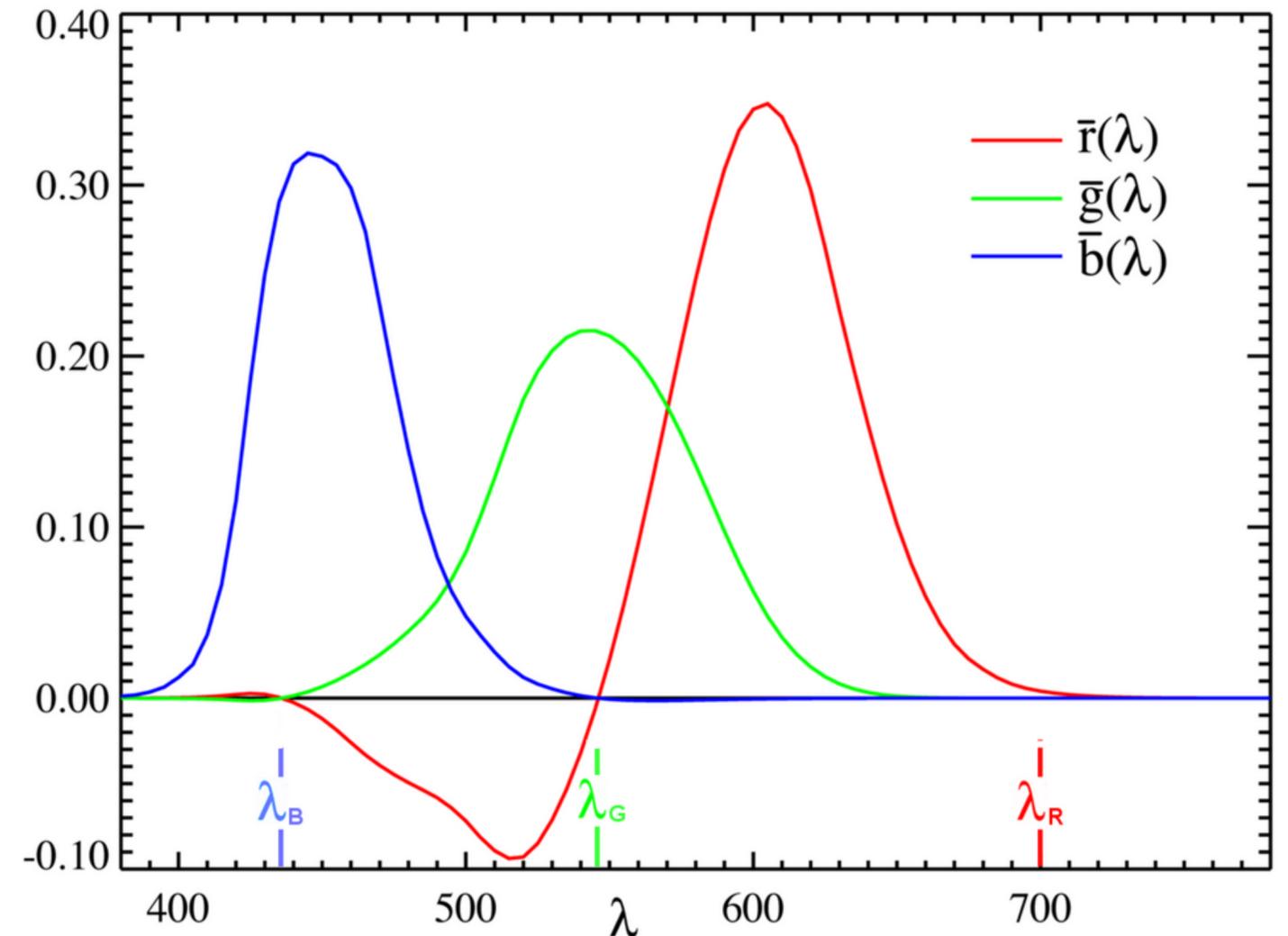
é dada por:

$$R = \int_0^{\infty} C(\lambda) \bar{r}(\lambda) d\lambda$$

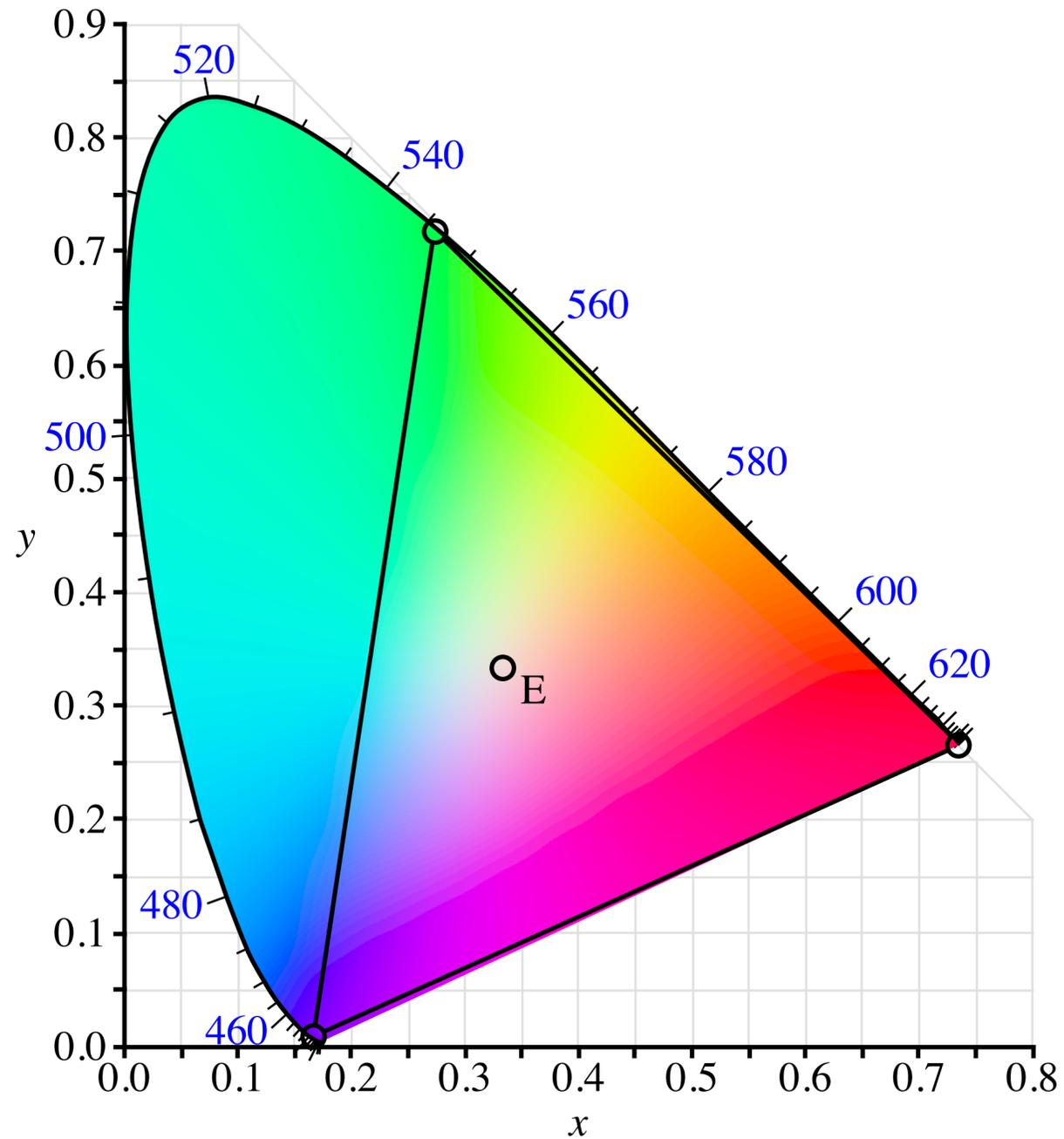
$$G = \int_0^{\infty} C(\lambda) \bar{g}(\lambda) d\lambda$$

$$B = \int_0^{\infty} C(\lambda) \bar{b}(\lambda) d\lambda$$

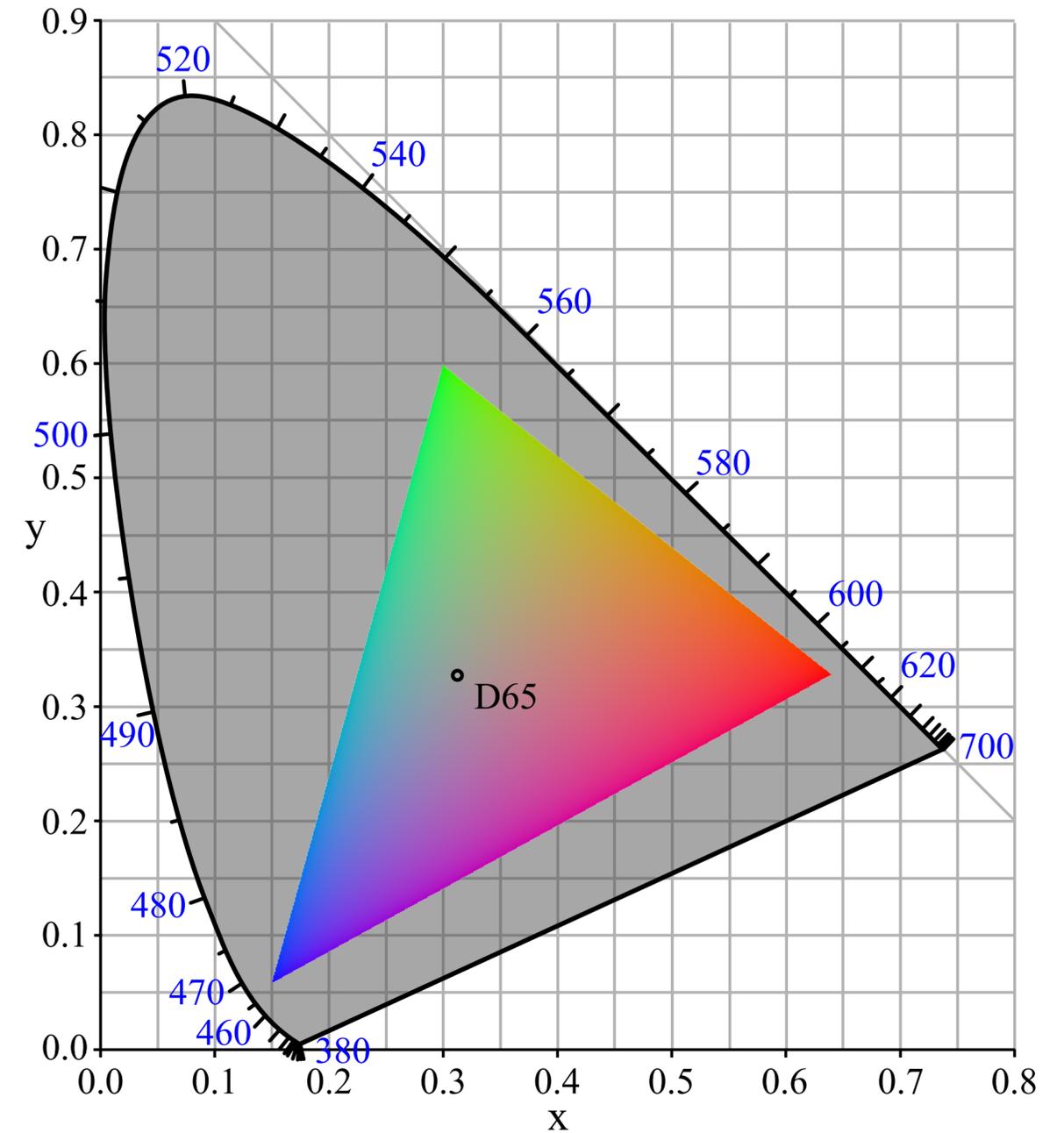
CIE 1931 RGB color-matching functions



# Cores em RGB



[https://en.wikipedia.org/wiki/CIE\\_1931\\_color\\_space#/media/File:CIE1931xy\\_CIERGB.svg](https://en.wikipedia.org/wiki/CIE_1931_color_space#/media/File:CIE1931xy_CIERGB.svg)



[https://en.wikipedia.org/wiki/SRGB#/media/File:SRGB\\_chromaticity\\_CIE1931.svg](https://en.wikipedia.org/wiki/SRGB#/media/File:SRGB_chromaticity_CIE1931.svg)

# Cores em RGB

- $\mathcal{E}$  é o conjunto das distribuições de energia no espectro visível
- $\sim$  é a relação de equivalência:  $c_1 \sim c_2 \iff c_1$  e  $c_2$  são metaméricas
- $\mathcal{C} = \mathcal{E} / \sim$
- $\mathcal{C}$  contém "um" representante de cada cor visível
- $c \in \mathcal{C} \implies c = (r, g, b), r, g, b \in [0, \infty]$

# Checkpoint



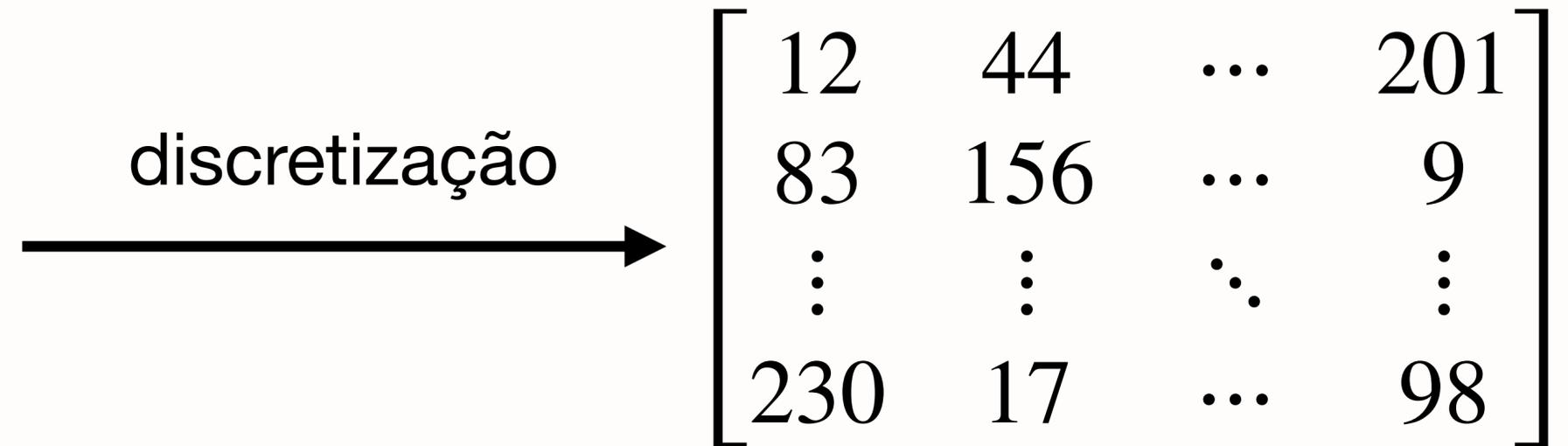
Uma *imagem* é uma função

$$I : \Omega \subset \mathbb{R}^2 \rightarrow \mathcal{C}$$

onde  $\Omega$  é o suporte da imagem e  $\mathcal{C}$  é o *conjunto de cores*

$$c \in \mathcal{C} \implies c = (r, g, b), r, g, b \in [0, \infty]$$

# Universo de Representação



HxWx3

Temos que discretizar no **domínio** e no **contra-domínio**

# Representando Cores RGB

$$c \in \mathcal{C} \implies c = (r, g, b) \text{ onde } r, g, b \in [0, \infty]$$

low dynamic range

high dynamic range

$r, g, b \in \{0, \dots, 255\}$  uint8

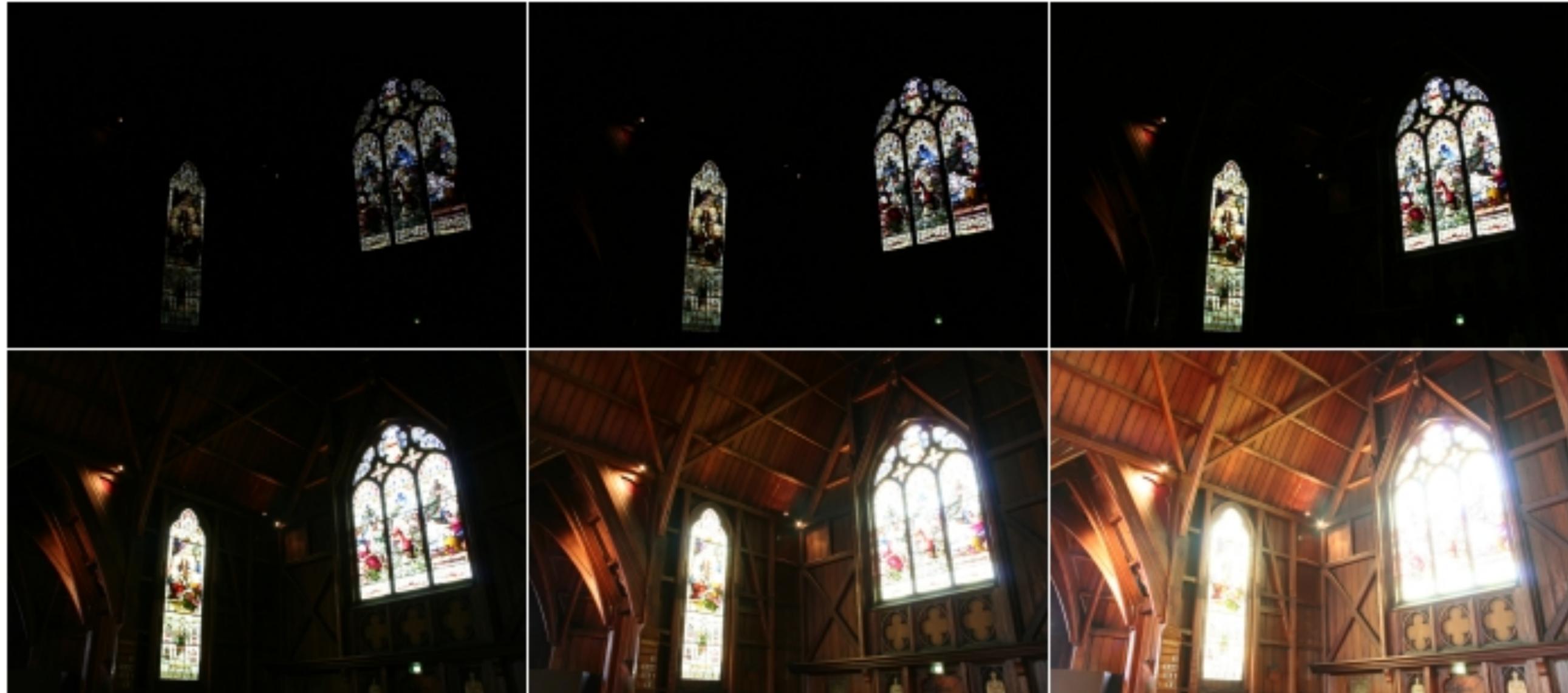
$r, g, b \in [0, \infty]$  float

- Cenas 2D (Vector Graphics)
- Cenas 3D (sem simulação física)

- Cenas 3D (com simulação física)

Tone Mapping

# HDR e Tone Mapping

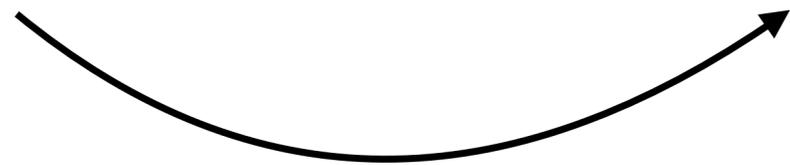
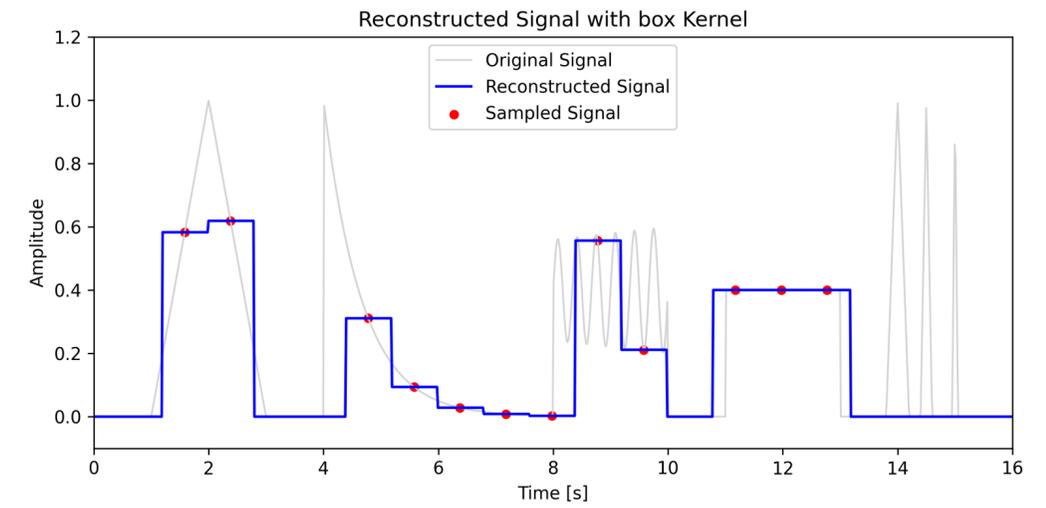
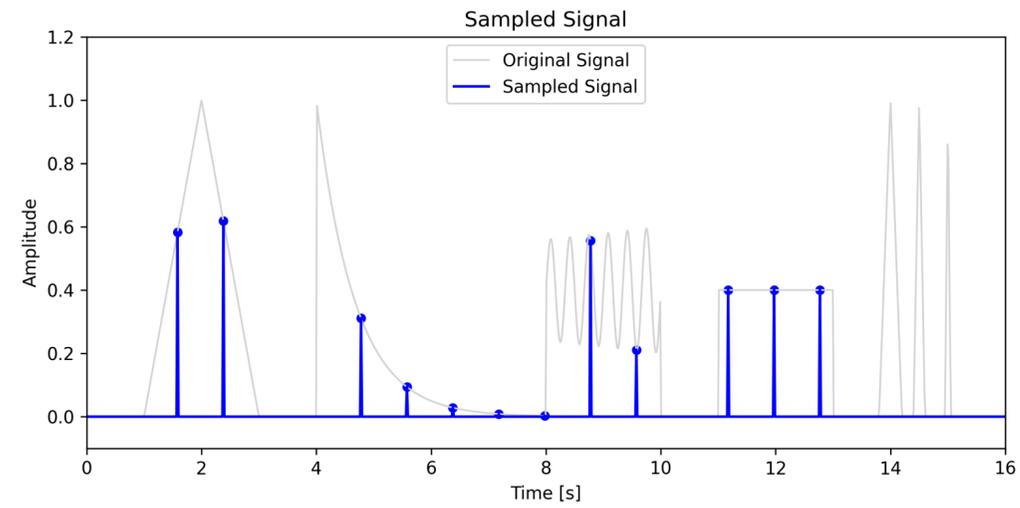
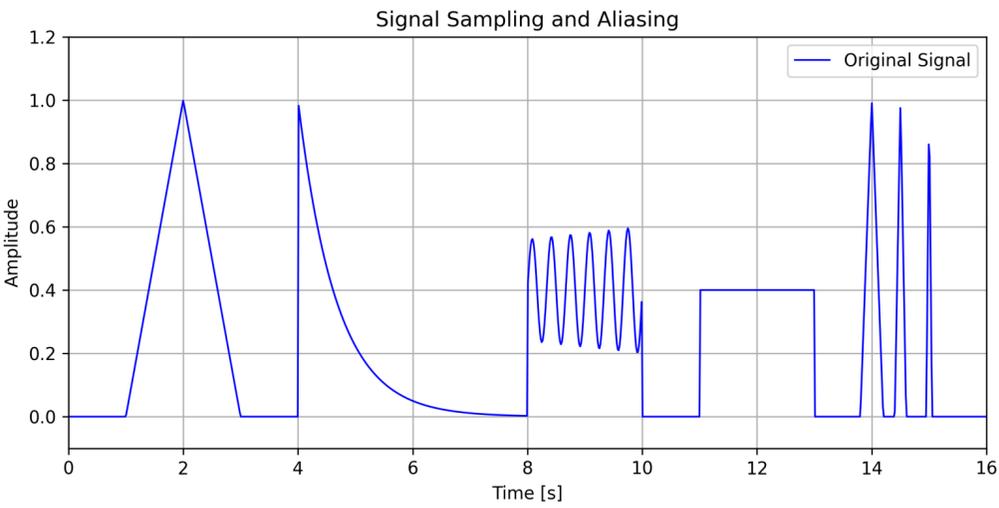


# HDR e Tone Mapping

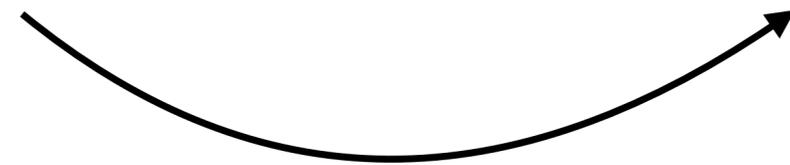


# Amostragem e Reconstrução

# Amostragem e Reconstrução

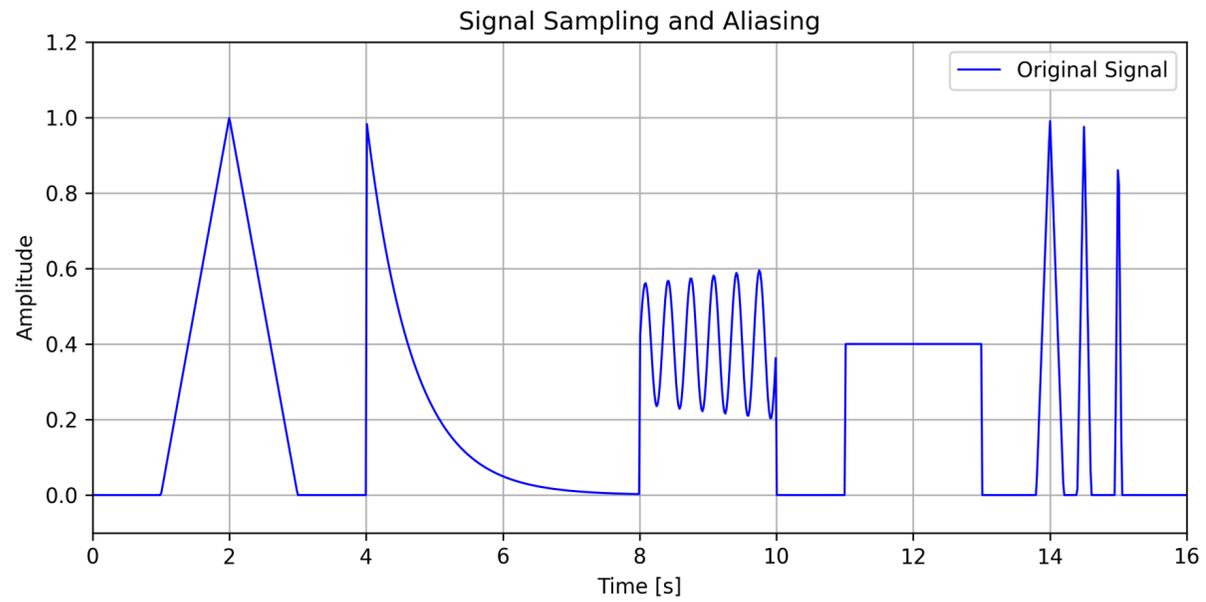


amostragem

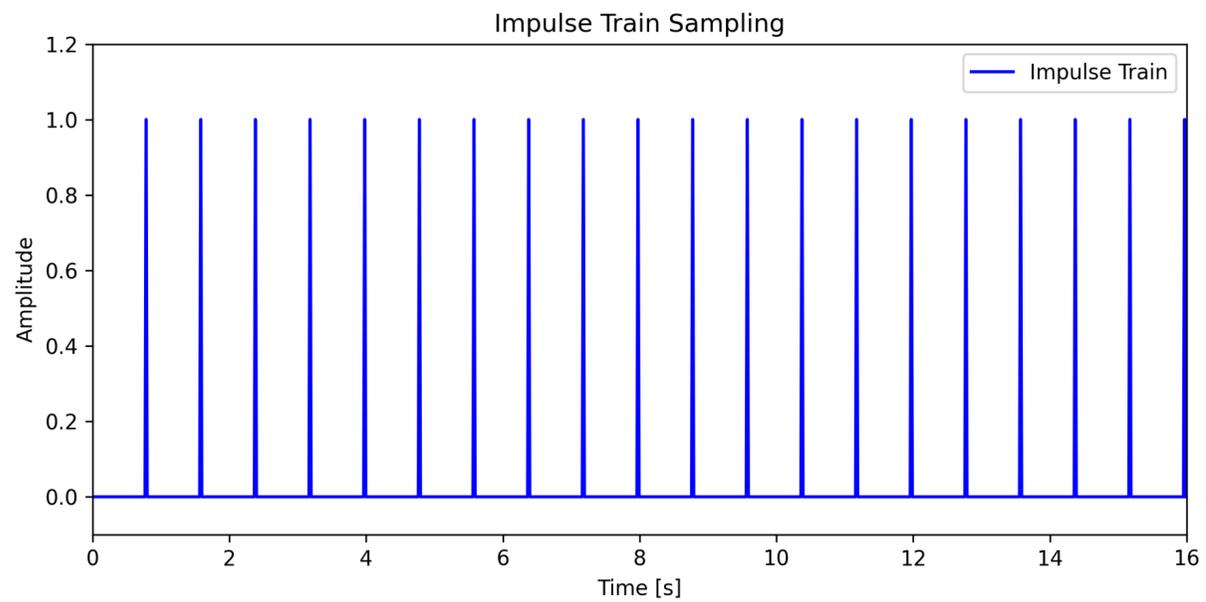


reconstrução

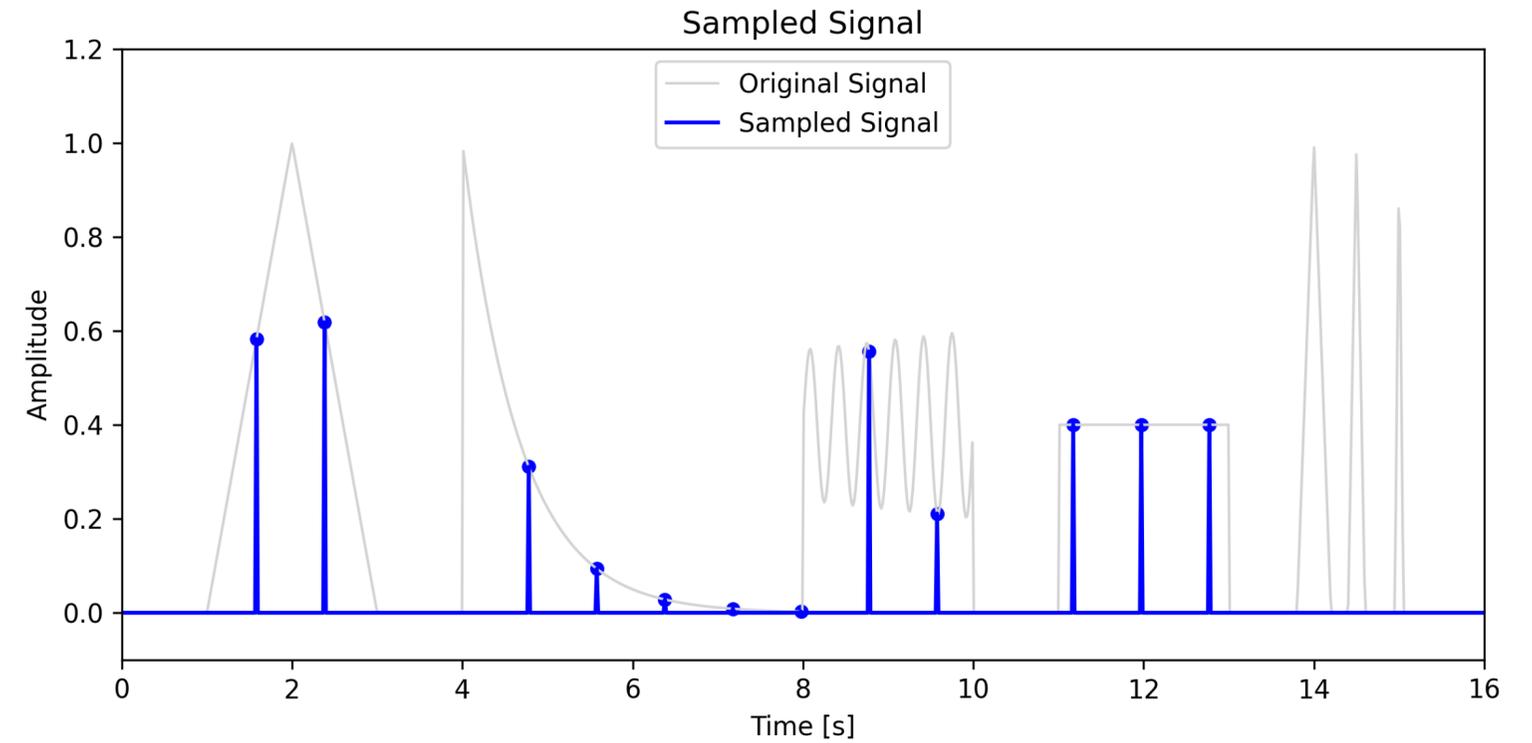
# Amostrar é multiplicar



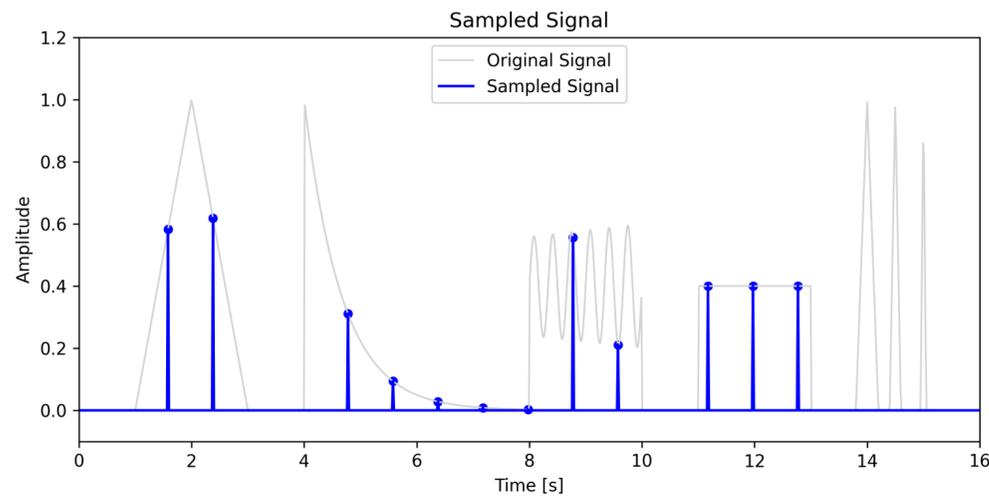
**X**



**=**



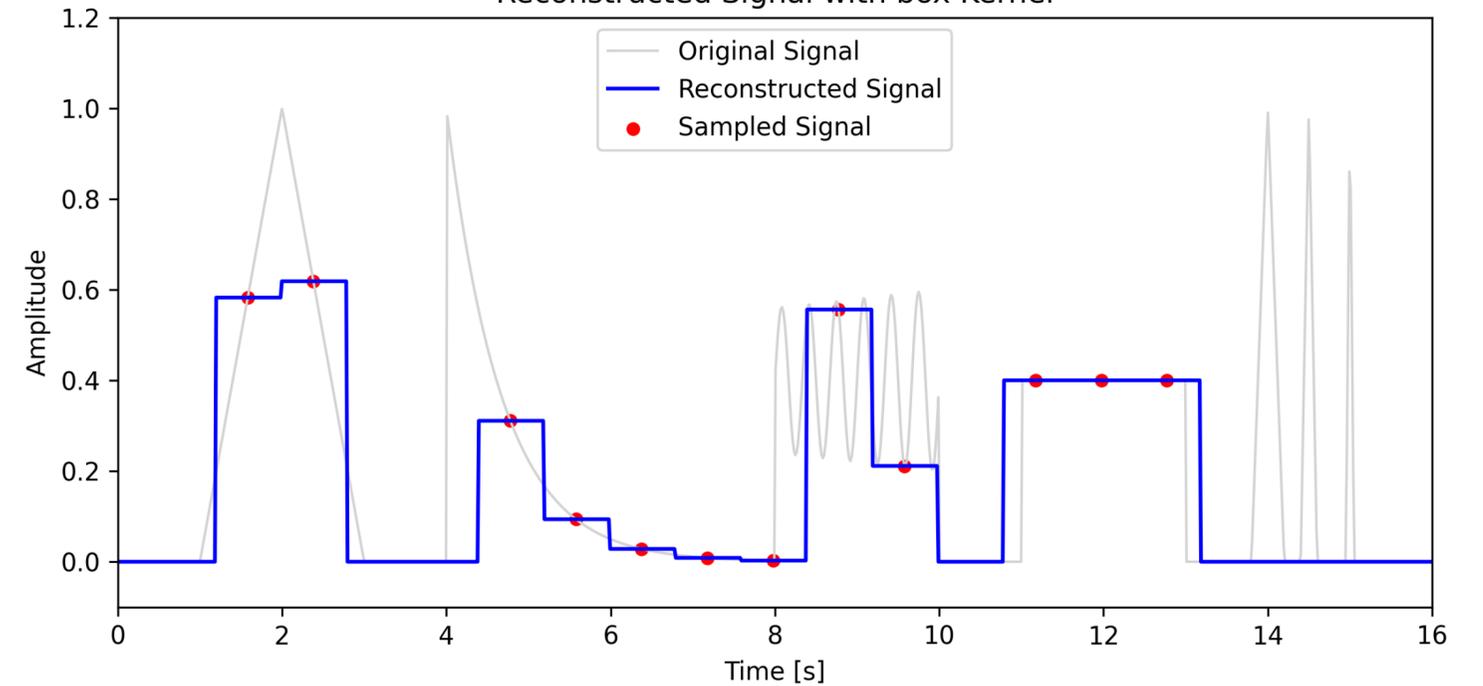
# Reconstruir é interpolar



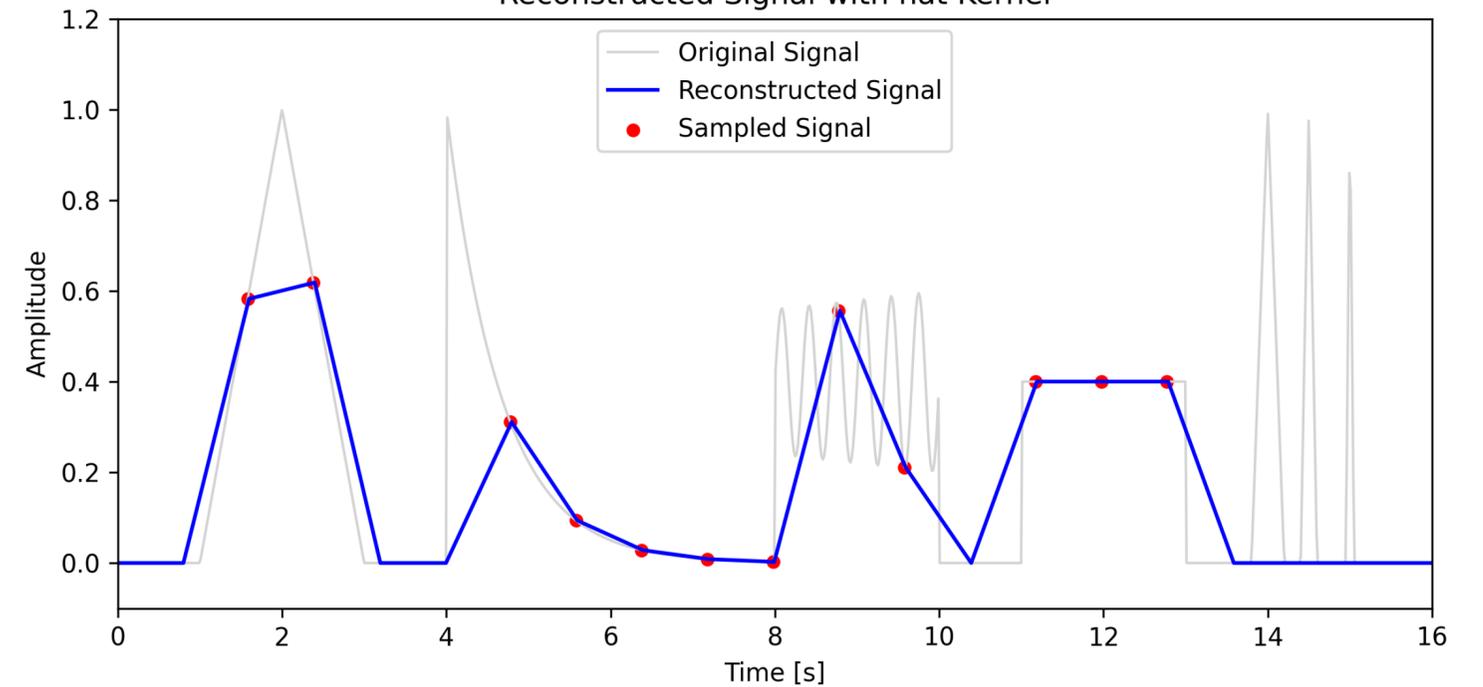
nearest

linear

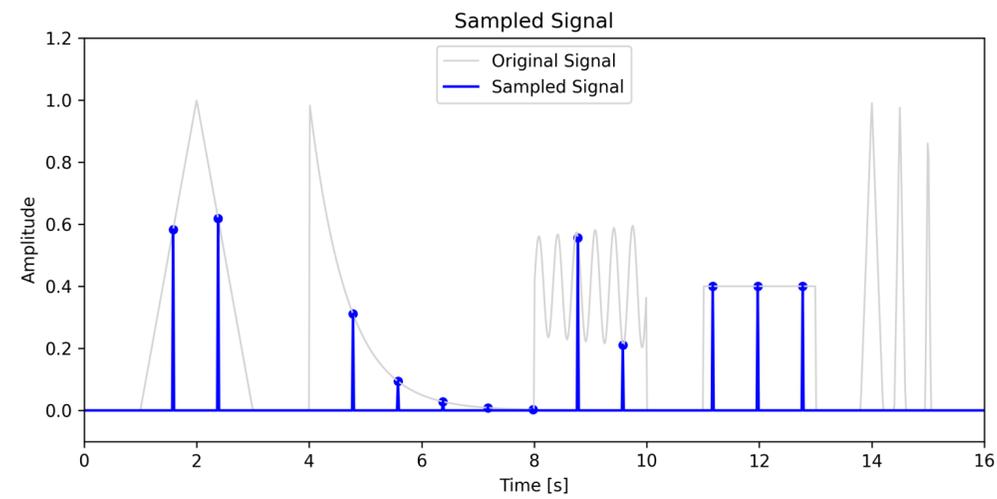
Reconstructed Signal with box Kernel



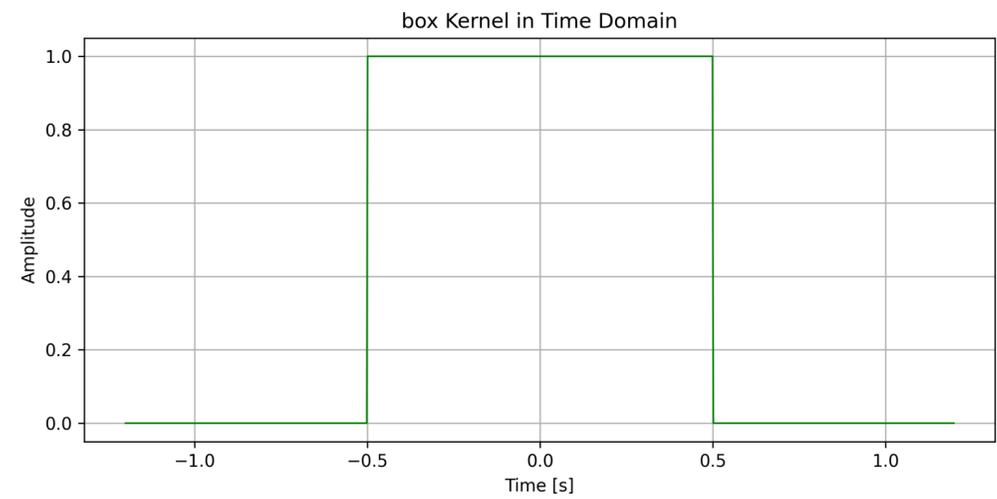
Reconstructed Signal with hat Kernel



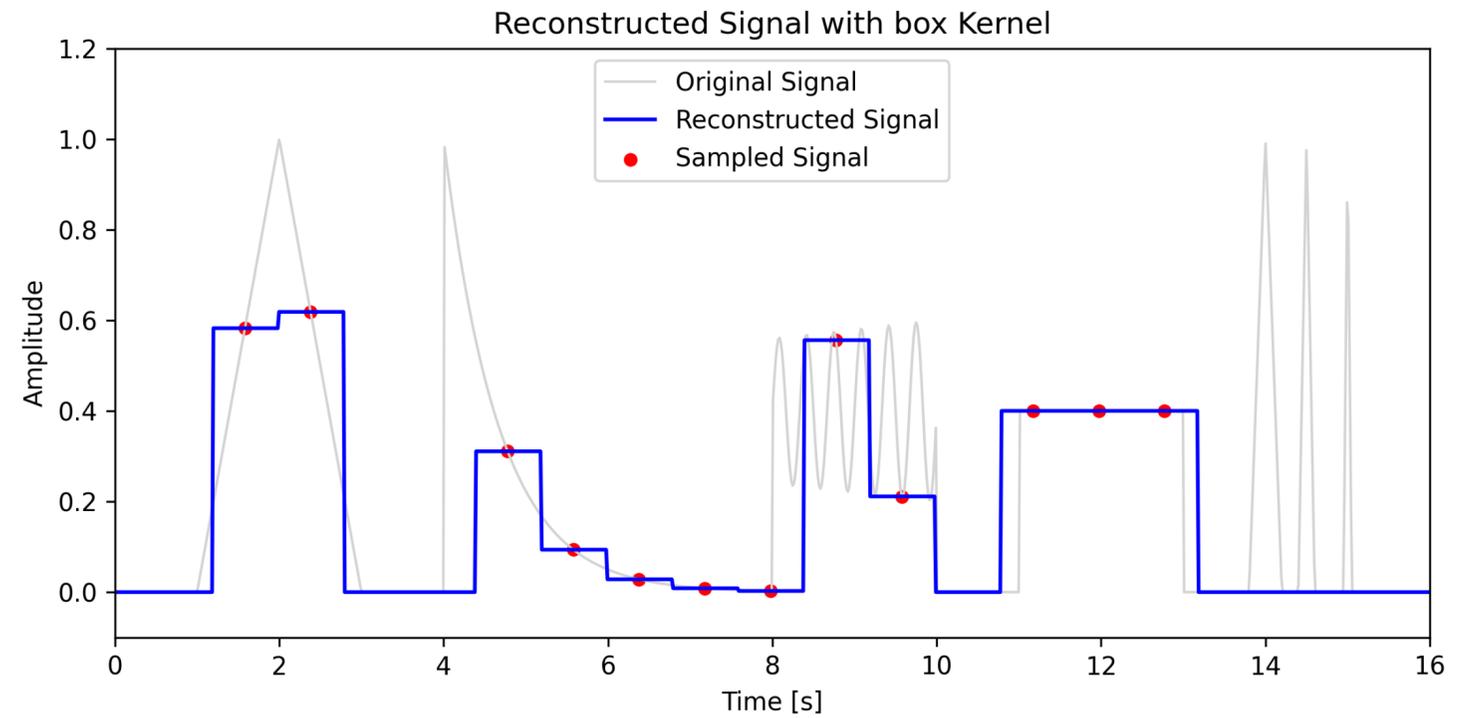
# Interpolação por convolução



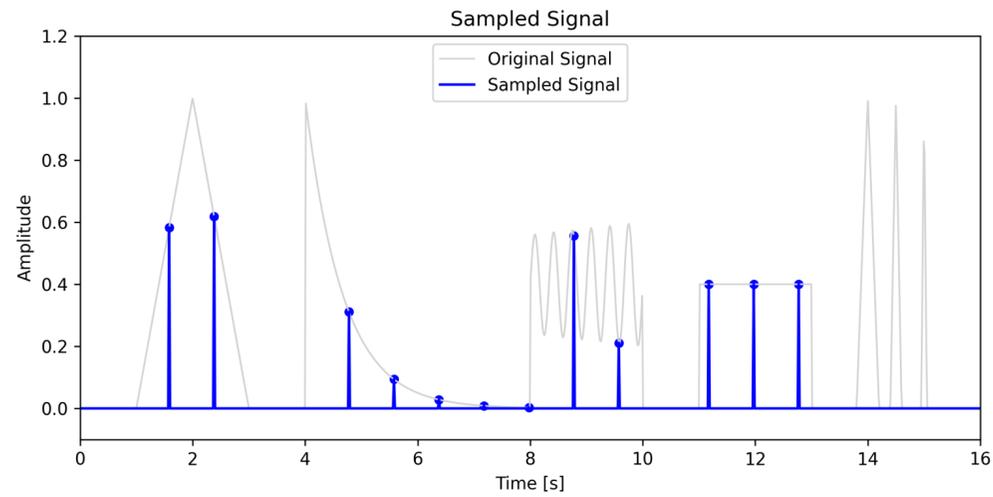
\*



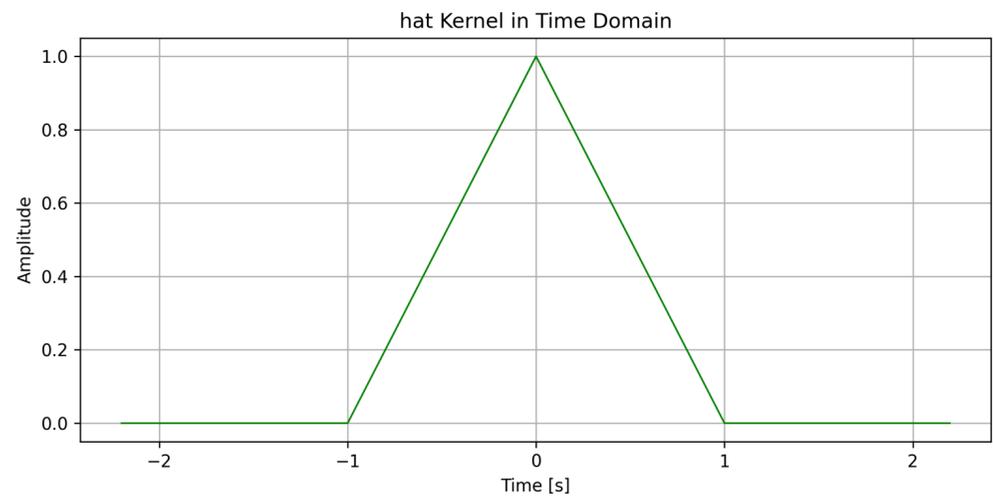
=



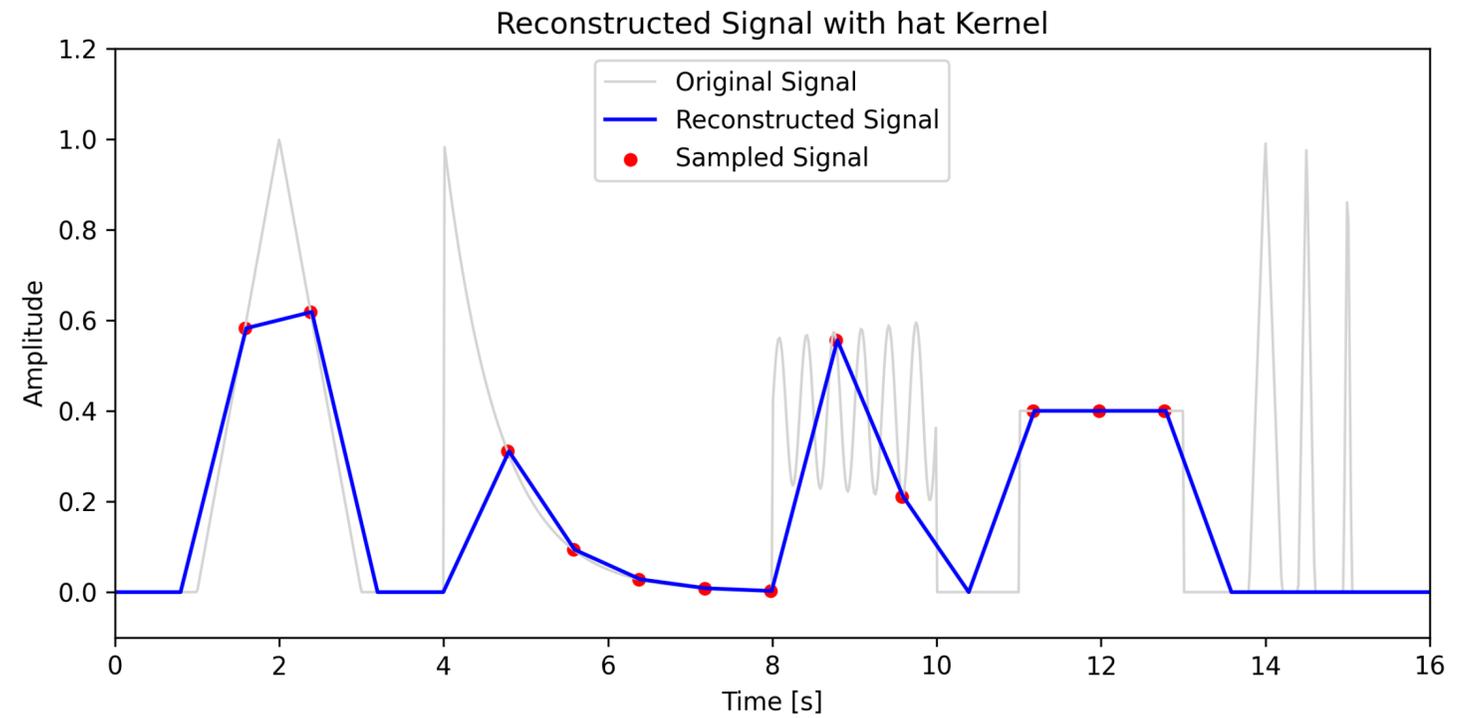
# Interpolação por convolução



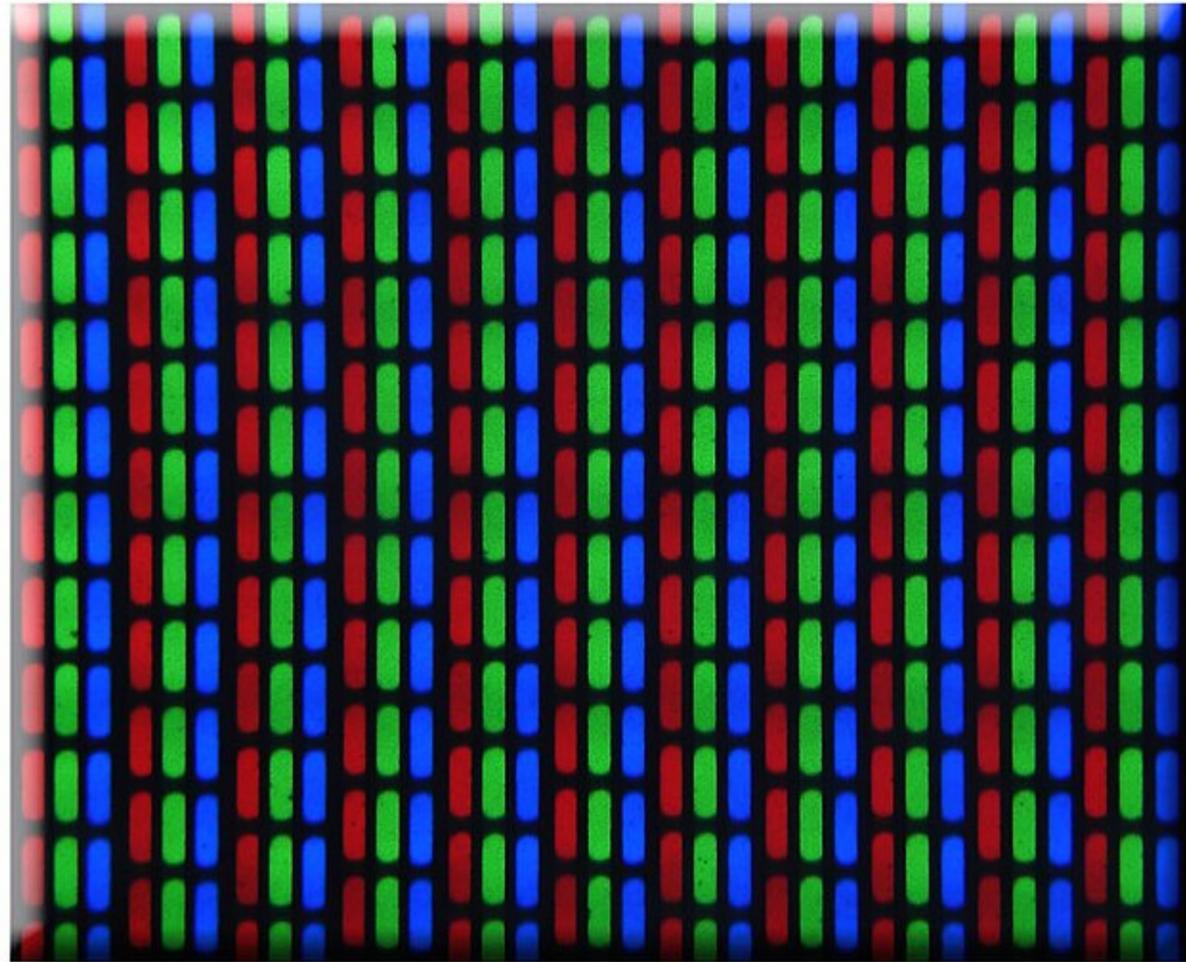
\*



=



# Reconstrução no CRT

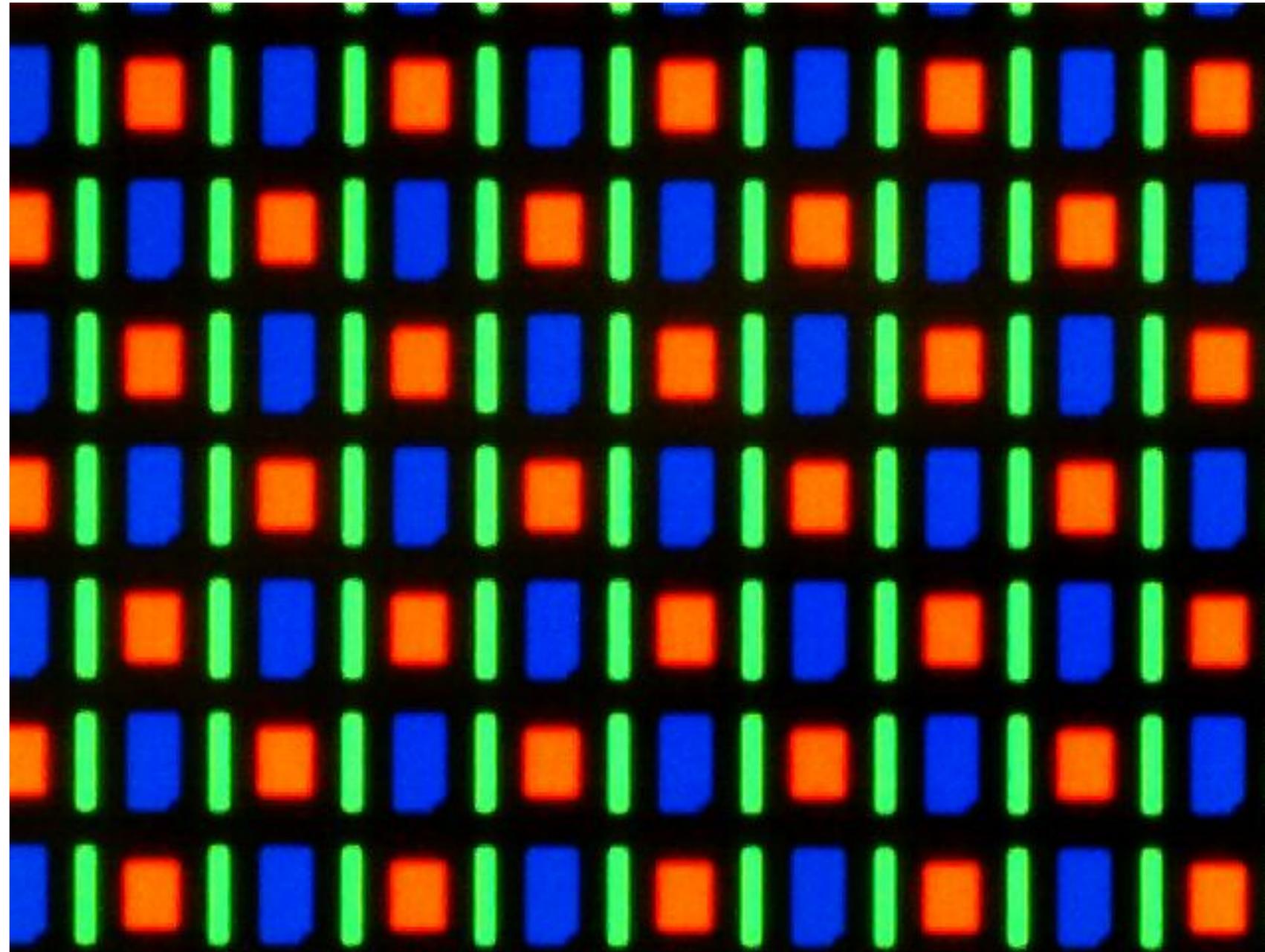


21" TV CRT Display



17" PC CRT Display

# Reconstrução no OLED

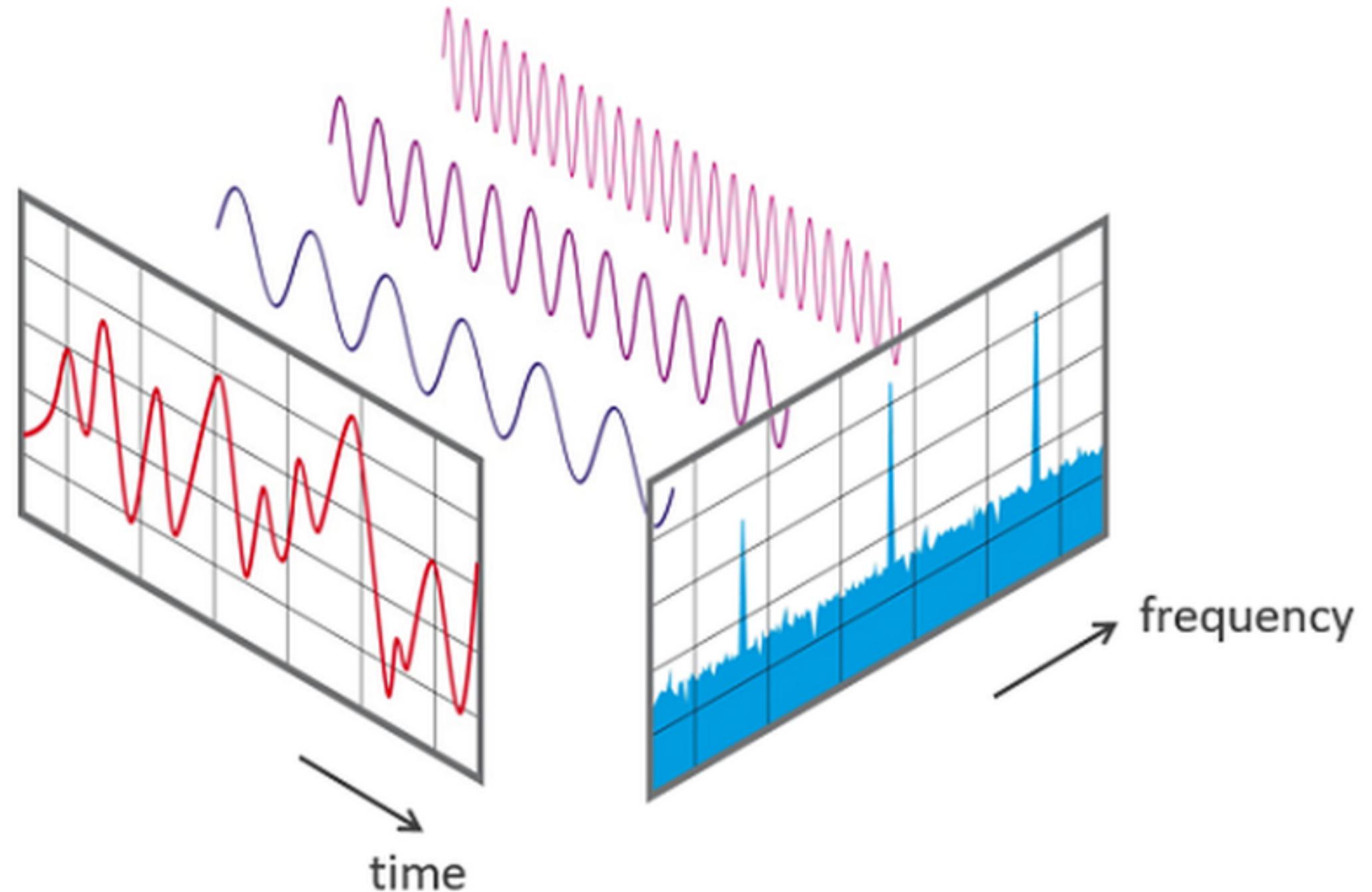


[https://en.wikipedia.org/wiki/OLED#/media/File:Nexus\\_one\\_screen\\_microscope.jpg](https://en.wikipedia.org/wiki/OLED#/media/File:Nexus_one_screen_microscope.jpg)

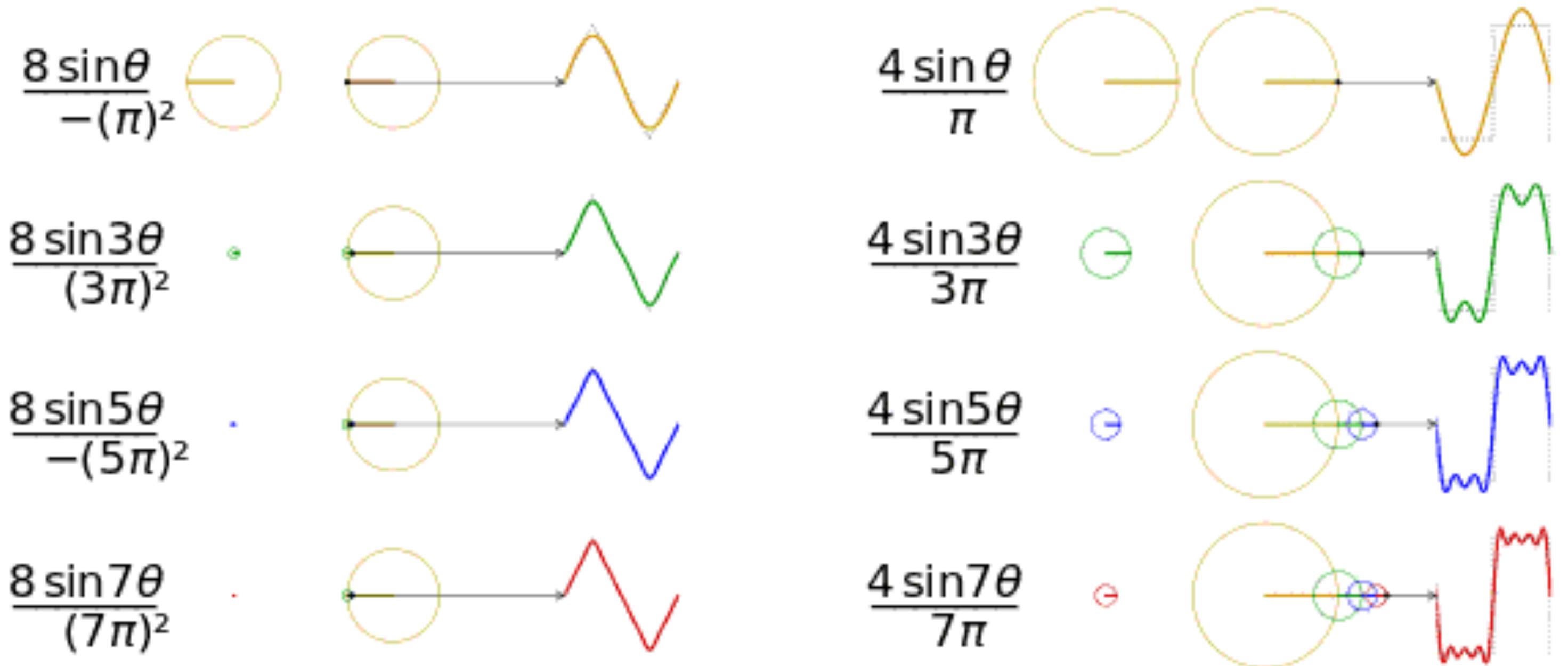
# Check Point

- temos um modelo matemático de imagem
- sabemos o que é amostragem
- sabemos o que é reconstrução
- como fazer isso direito?

# Transformada de Fourier



# Transformada de Fourier



# Transformada de Fourier

Definição:

$$\mathcal{F}(f)(\omega) = \hat{f}(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi\omega x} dx, \quad \forall \omega \in \mathbb{R}$$

- $\hat{f}(\omega) \in \mathbb{C}, \forall \omega \in \mathbb{R}$
- $|\hat{f}(\omega)|$  é a **amplitude** da componente de frequência  $\omega$
- $\theta(\hat{f}(\omega))$  é a **fase** da componente de frequência  $\omega$

Inversa:

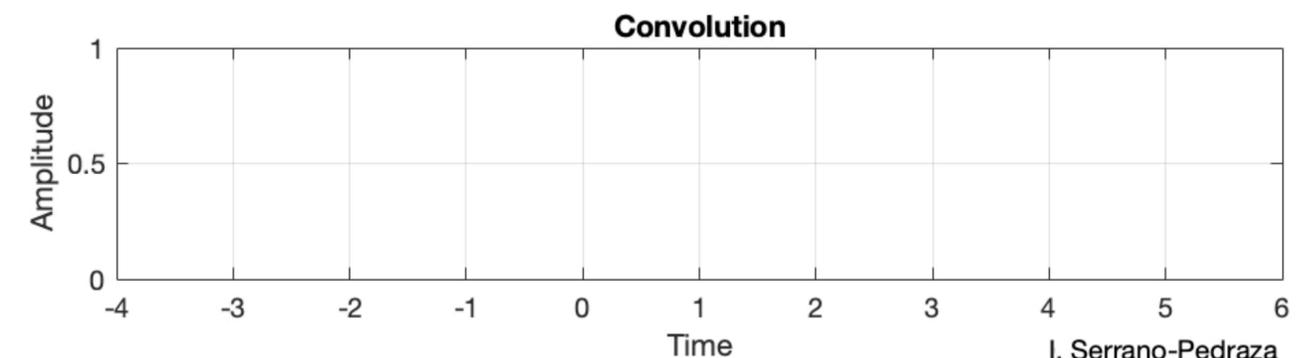
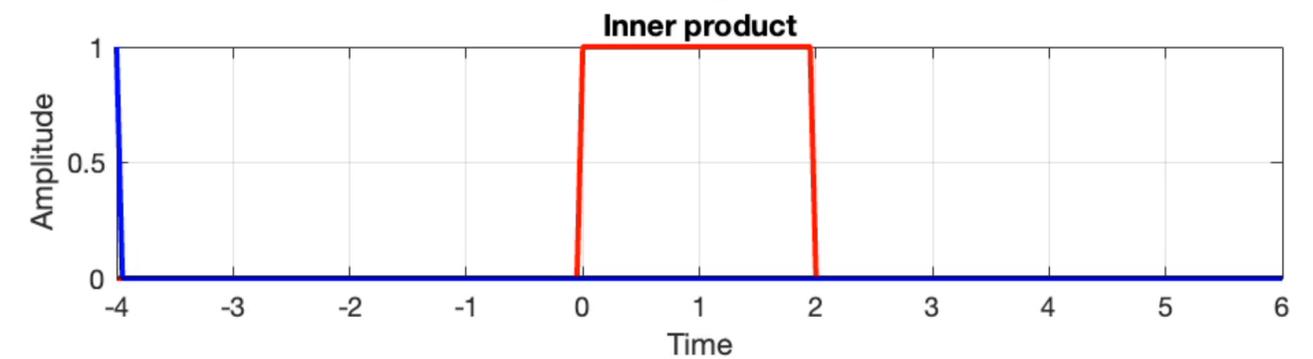
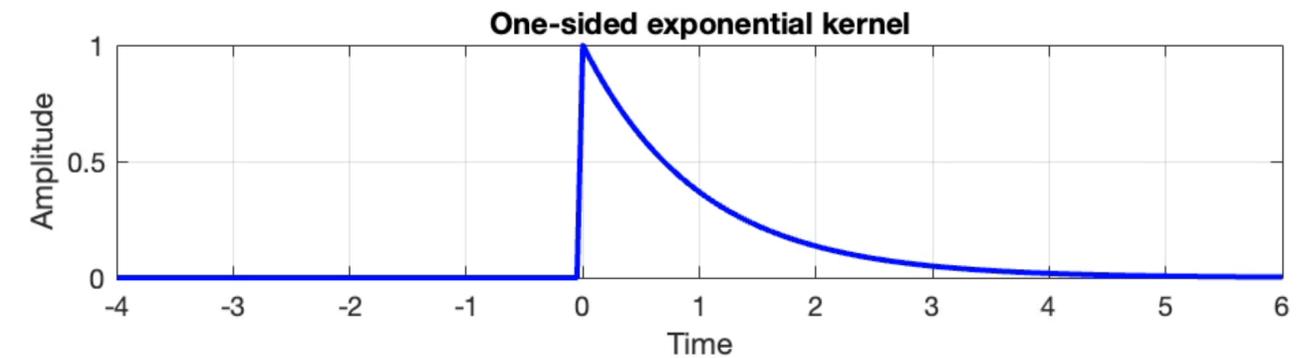
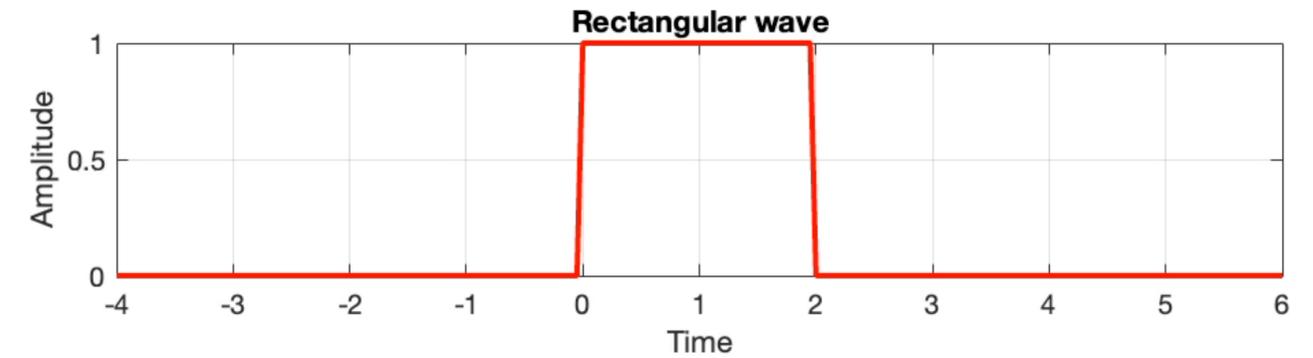
$$\mathcal{F}^{-1}(\hat{f})(x) = f(x) = \int_{-\infty}^{\infty} \hat{f}(\omega) e^{i2\pi\omega x} d\omega, \quad \forall x \in \mathbb{R}$$

# Convolução

Definição:

dadas  $f: \mathbb{R} \rightarrow \mathbb{R}$  e  $g: \mathbb{R} \rightarrow \mathbb{R}$

$$(f * g)(t) := \int_{-\infty}^{\infty} f(\tau)g(t - \tau) d\tau$$



I. Serrano-Pedraza

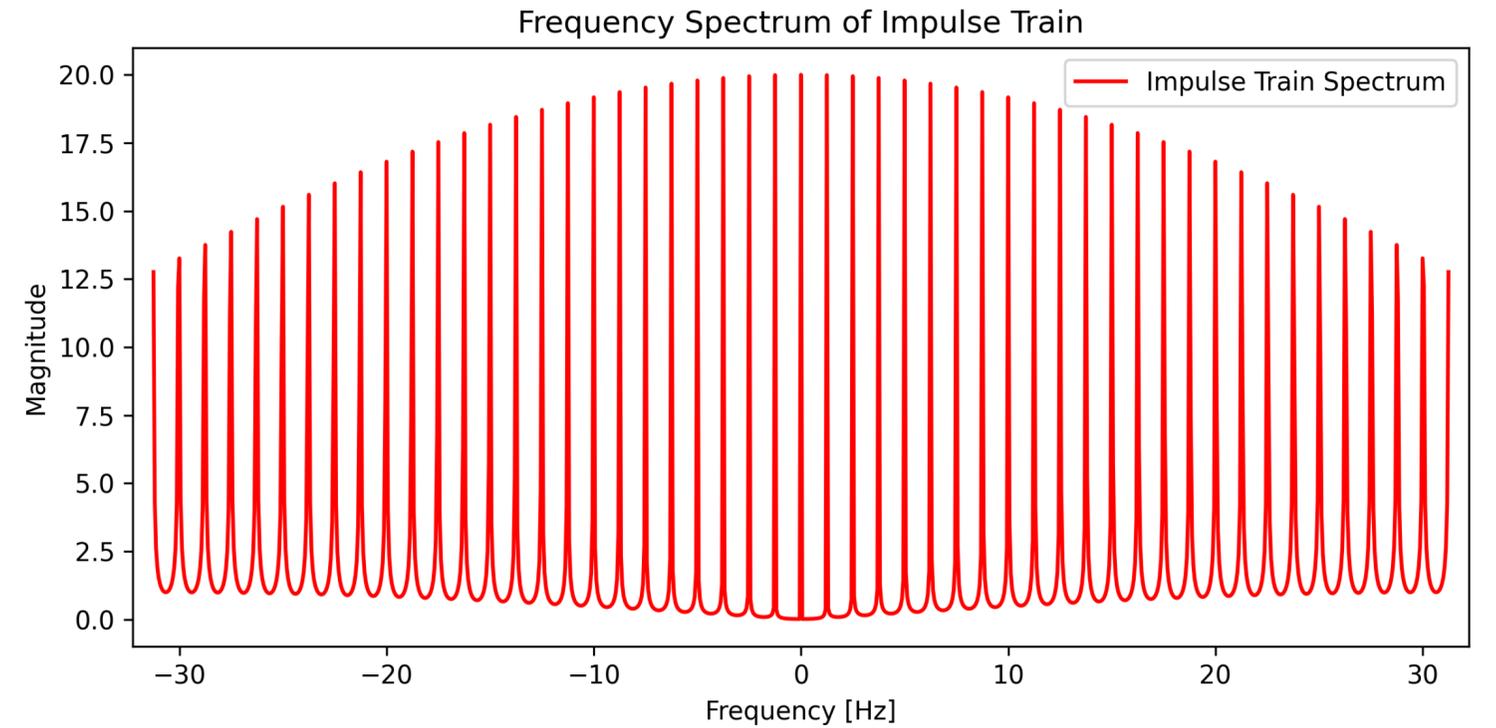
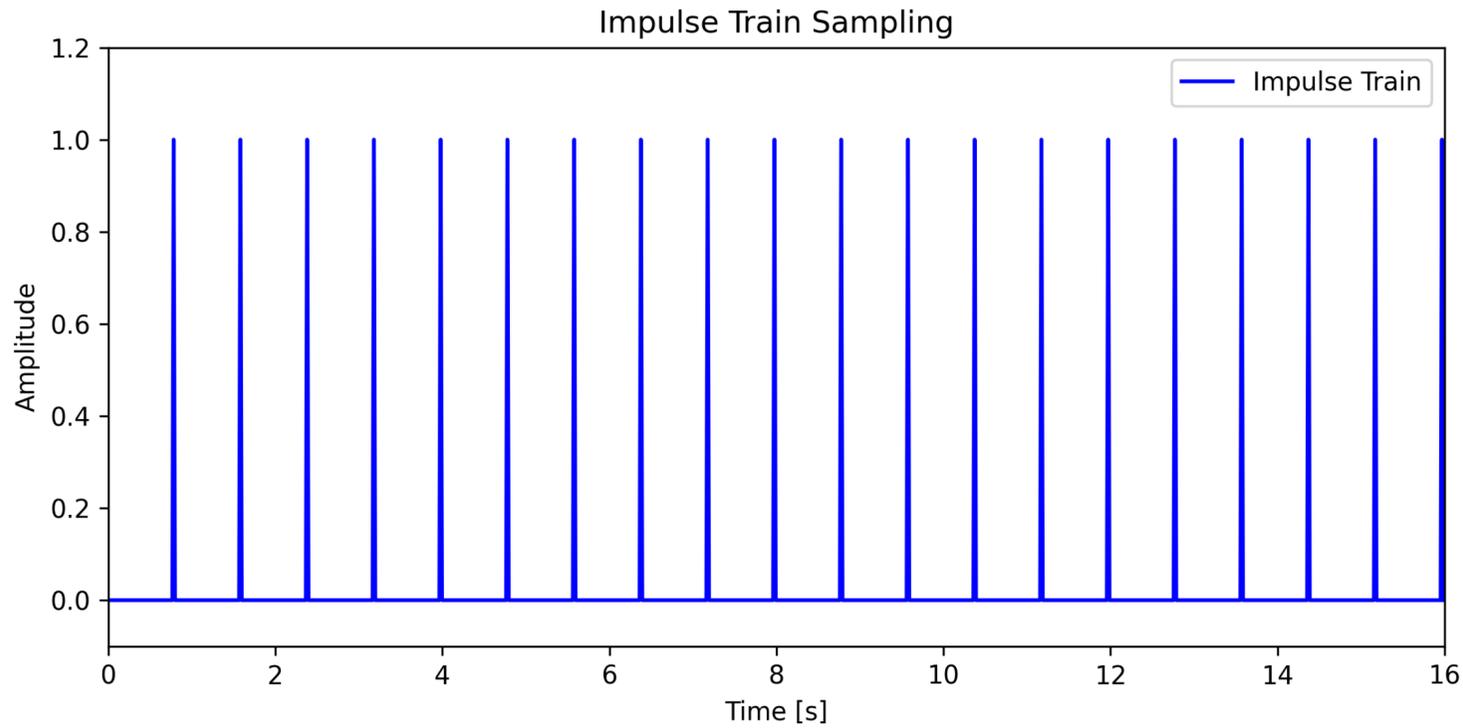
# Teorema da Convolução

Dadas  $f: \mathbb{R} \rightarrow \mathbb{R}$  e  $g: \mathbb{R} \rightarrow \mathbb{R}$ , temos que

$$\mathcal{F}(f * g) = \mathcal{F}(f) \cdot \mathcal{F}(g)$$

$$\mathcal{F}(f \cdot g) = \mathcal{F}(f) * \mathcal{F}(g)$$

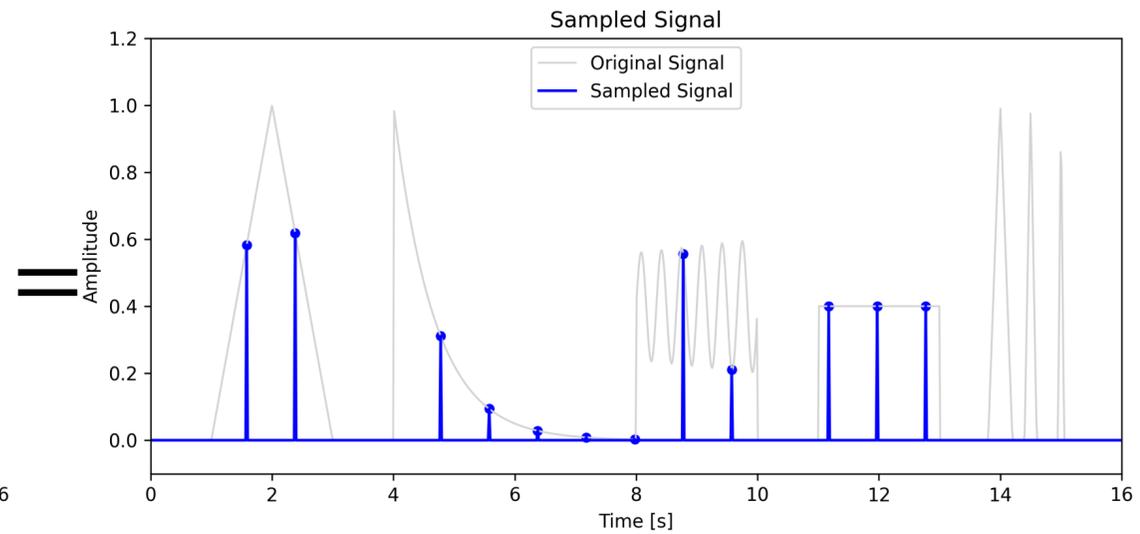
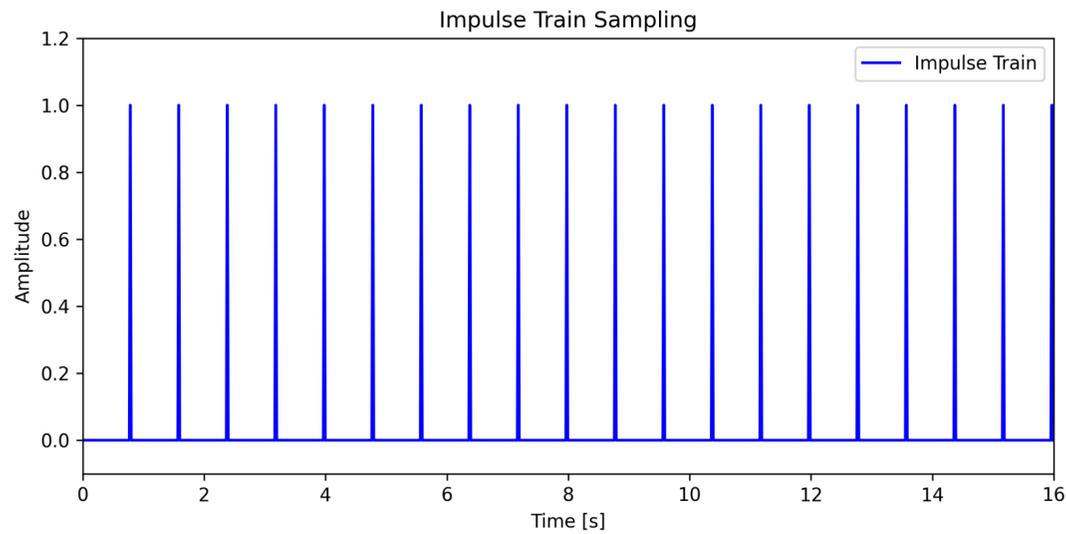
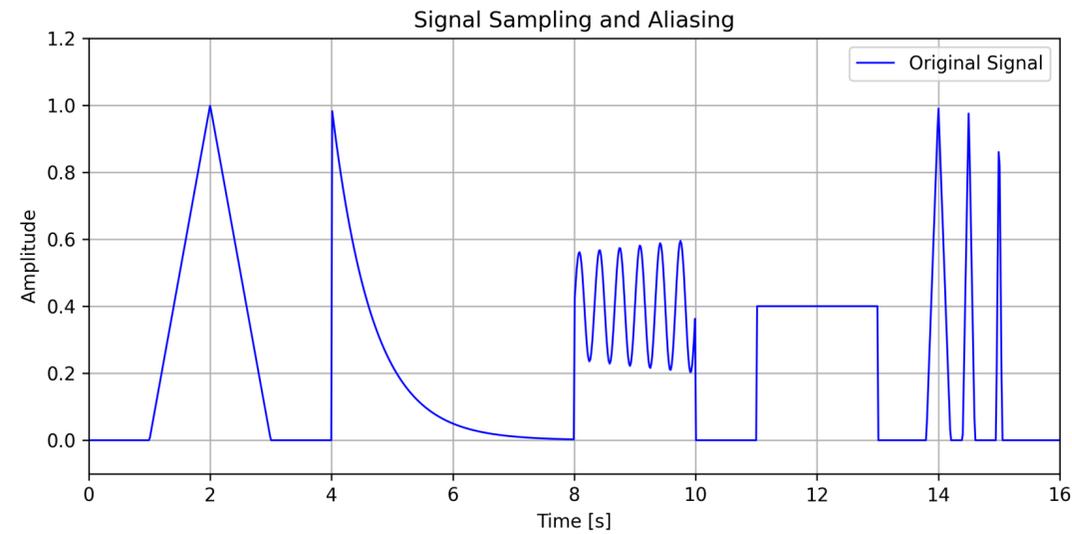
# Trem de impulsos



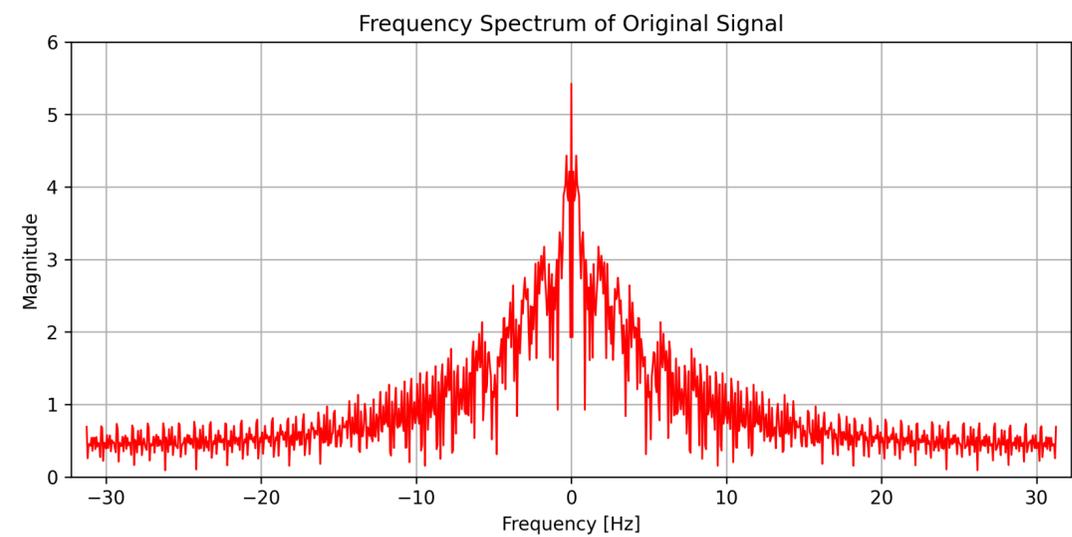
$$\mathbf{\Psi}_T(t) := \sum_{k=-\infty}^{\infty} \delta(t - kT)$$

$$\mathcal{F} [\mathbf{\Psi}_T](\omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} \delta(\omega - k\frac{1}{T}) = \frac{1}{T} \mathbf{\Psi}_{\frac{1}{T}}(\omega)$$

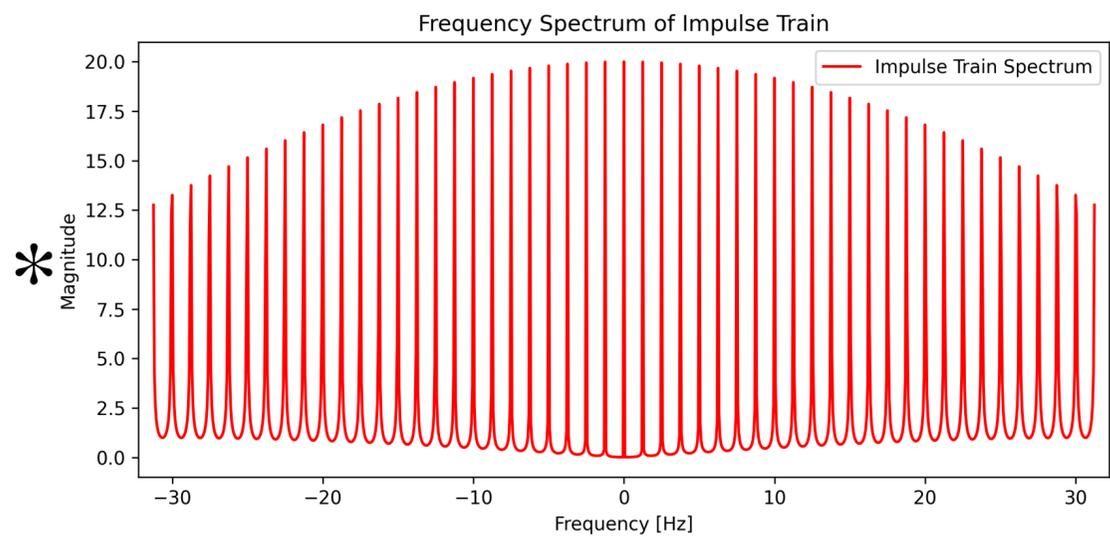
# Exemplo: amostragem



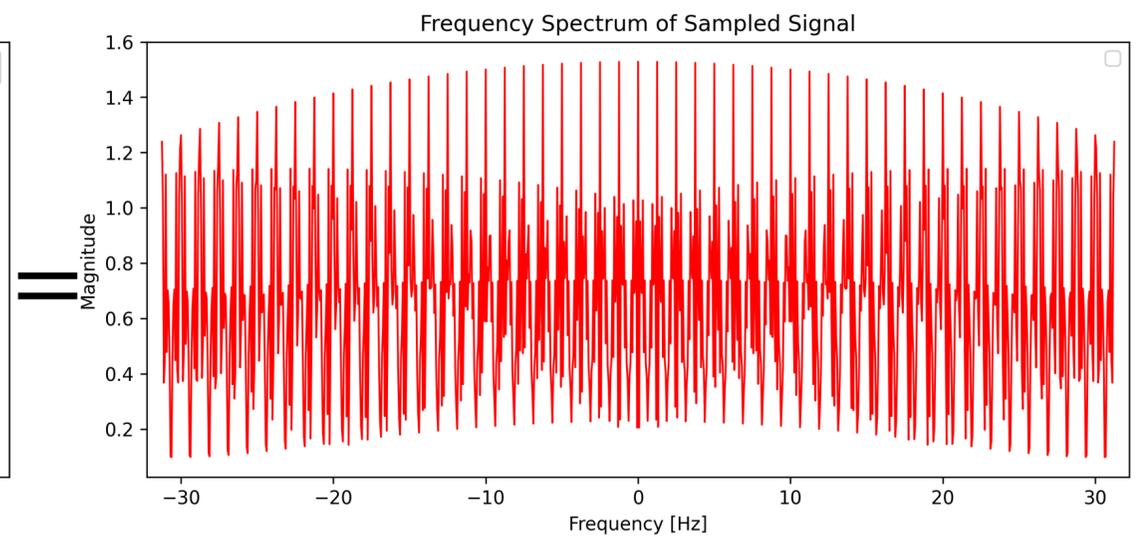
$\mathcal{F}$



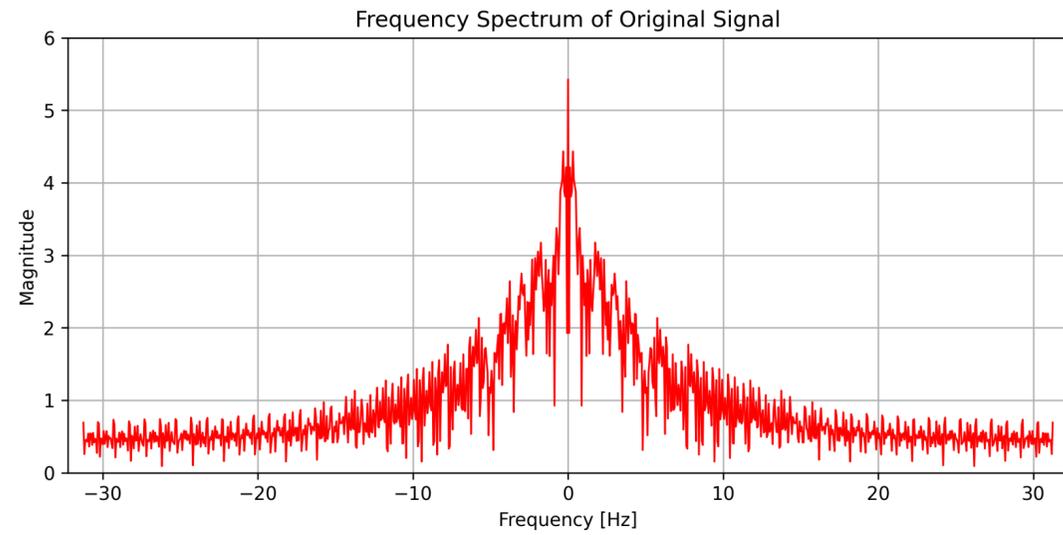
$\mathcal{F}$



$\mathcal{F}$

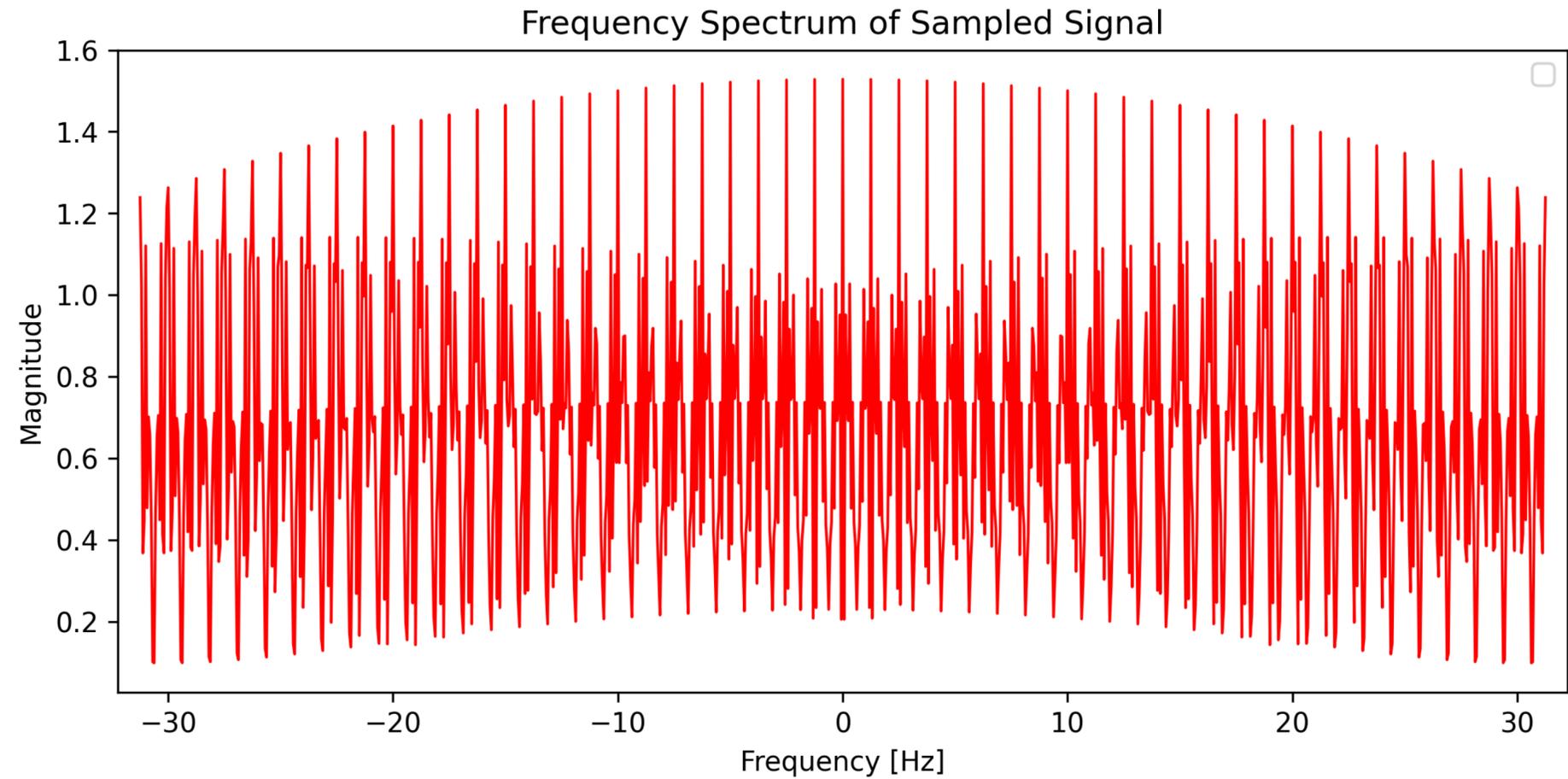
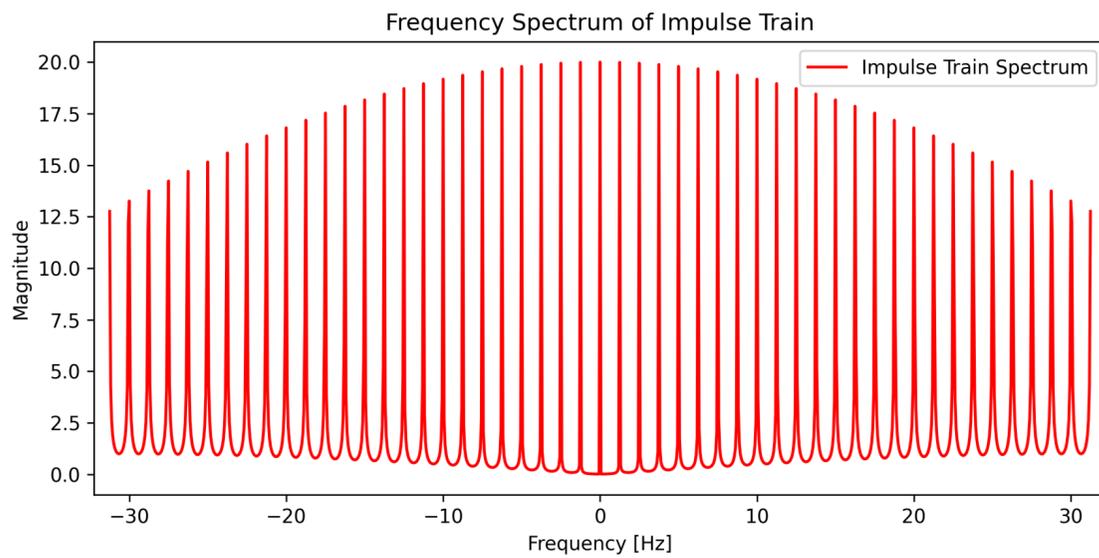


# Convolução com trem de impulsos



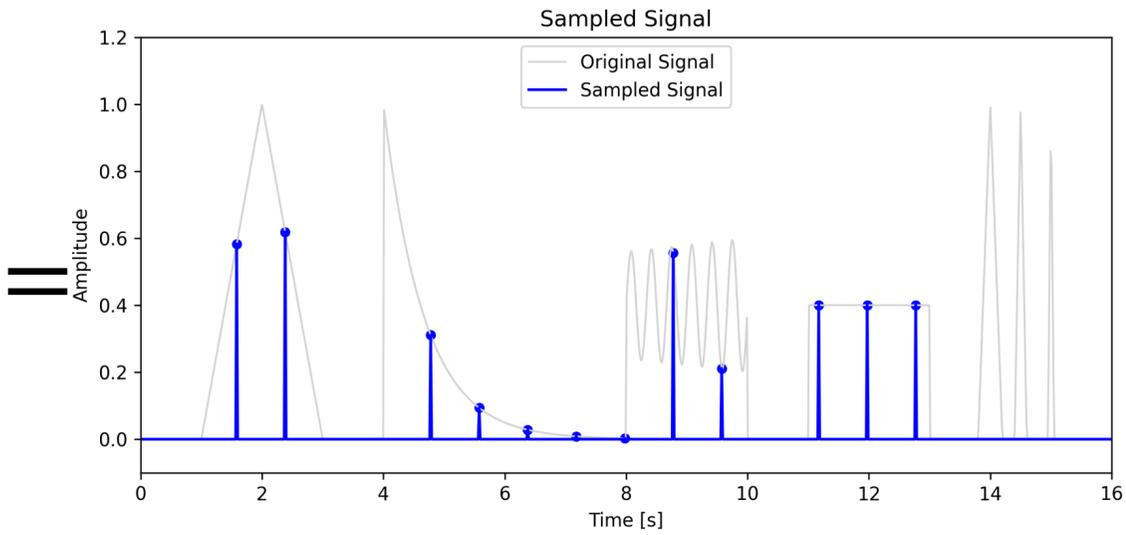
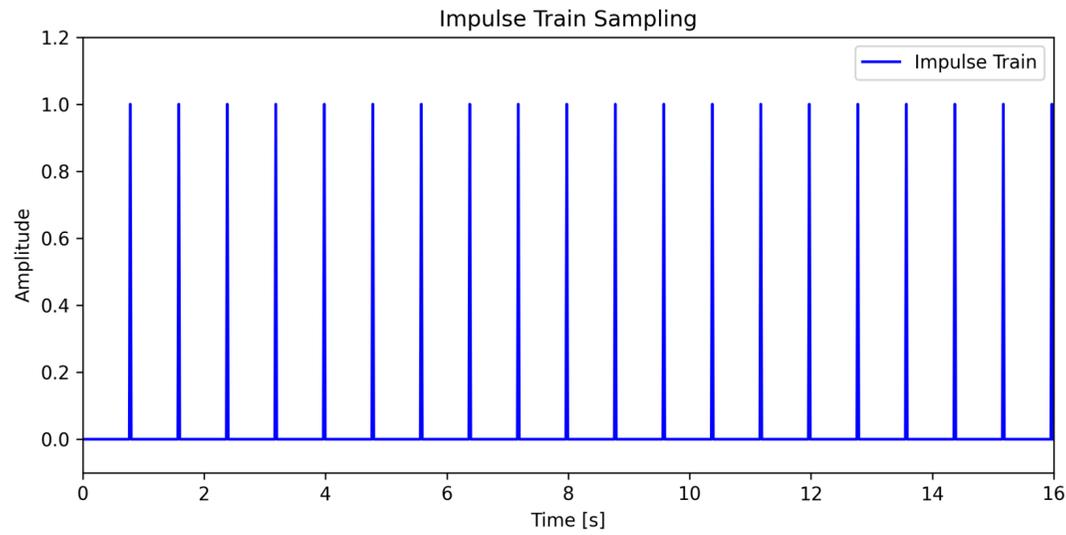
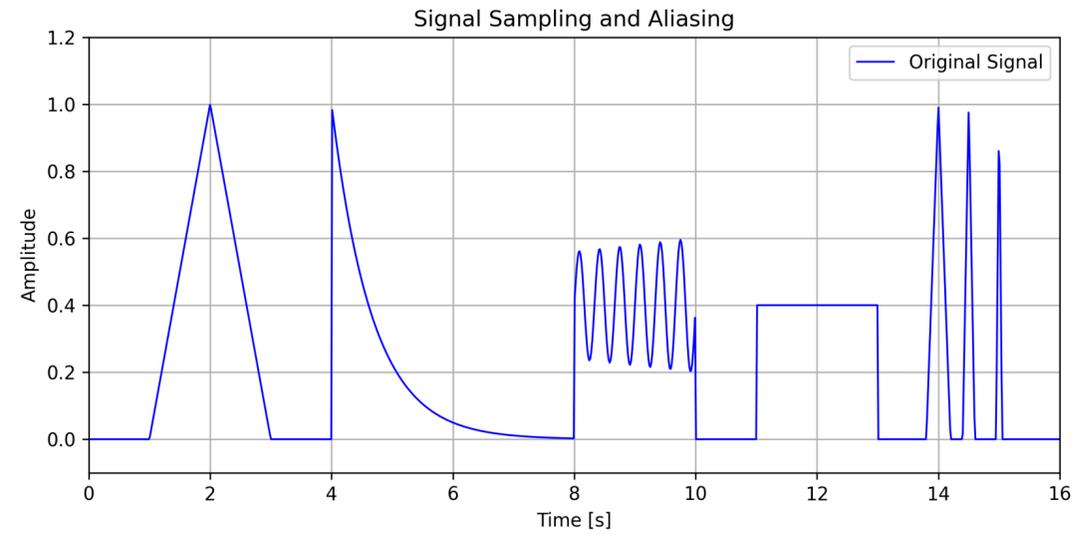
\*

=

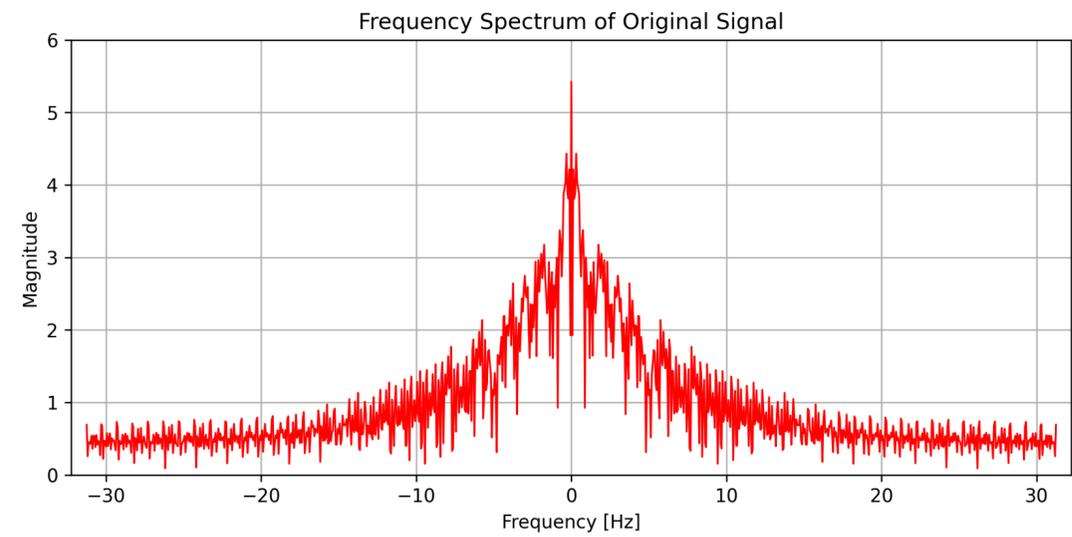


# Amostragem

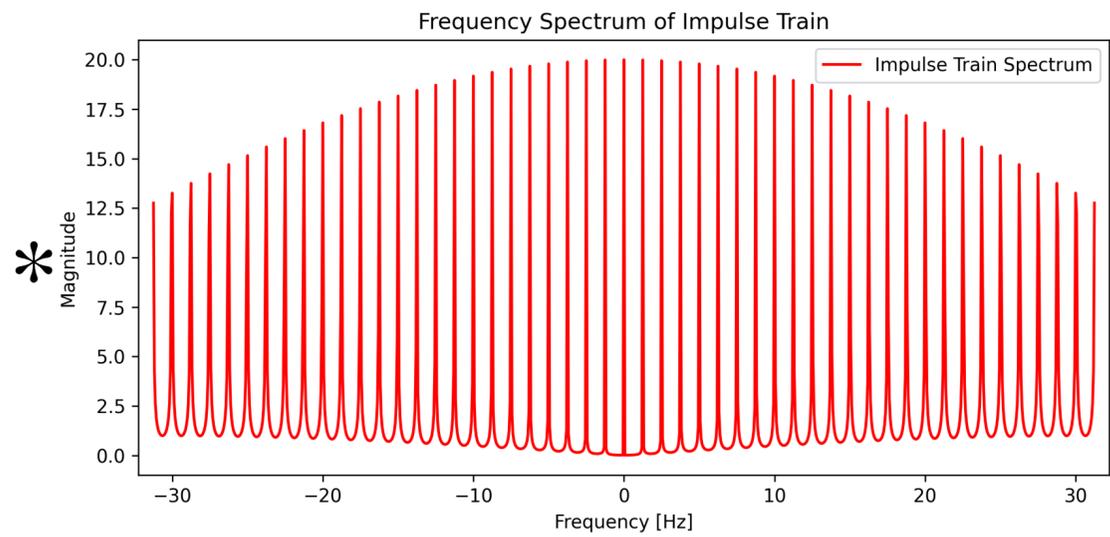
1:50



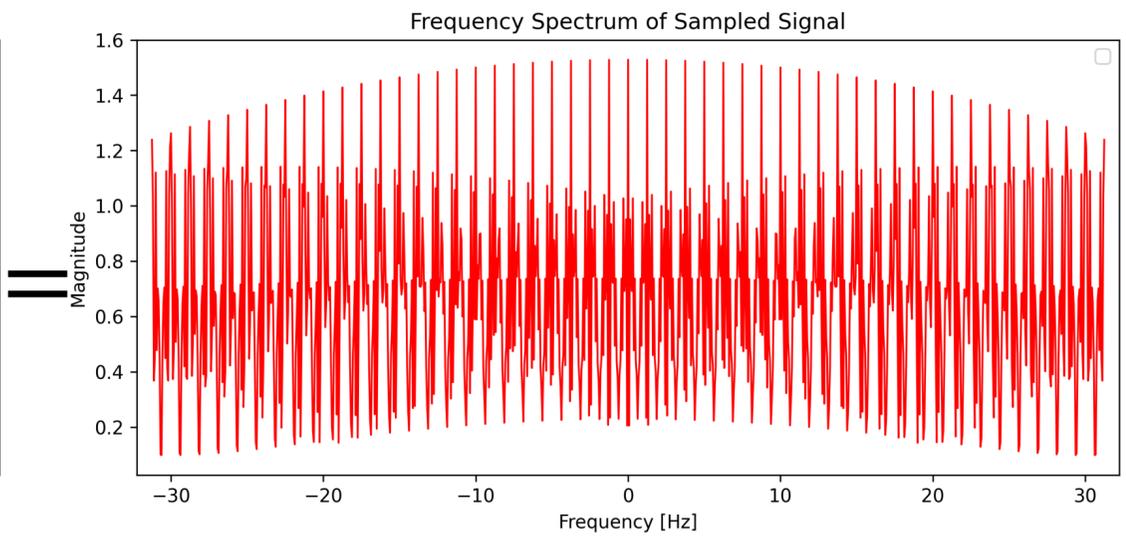
$\mathcal{F}$



$\mathcal{F}$



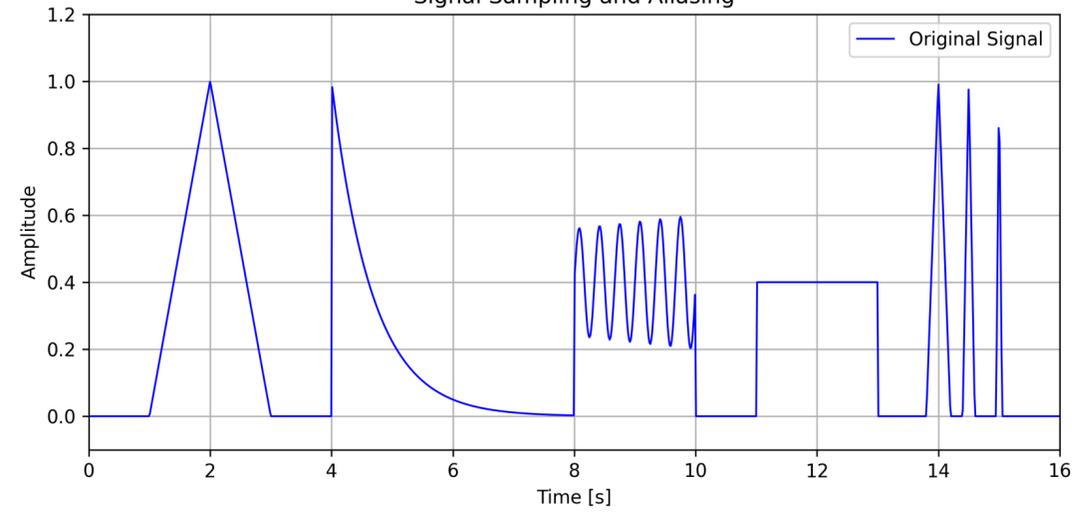
$\mathcal{F}$



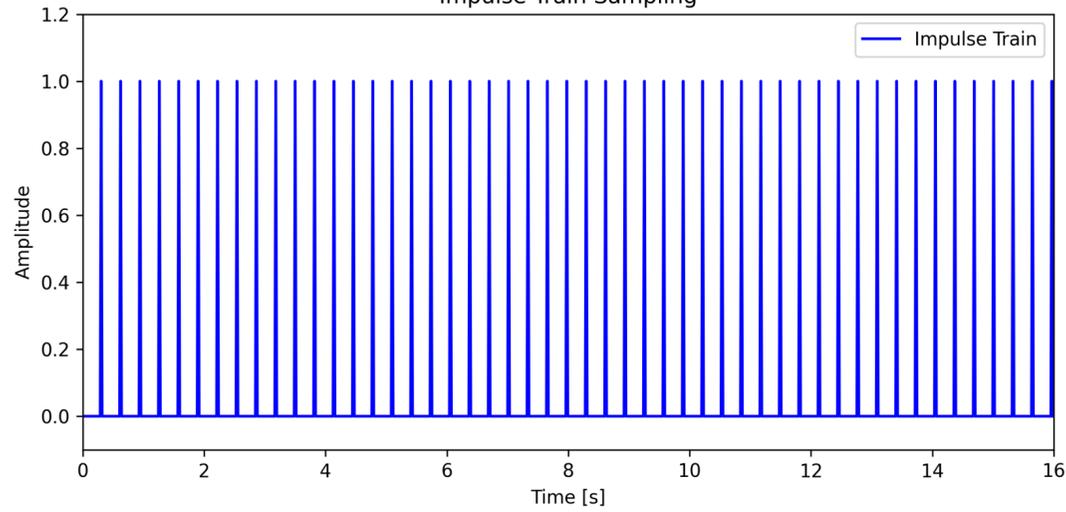
# Amostragem

1:20

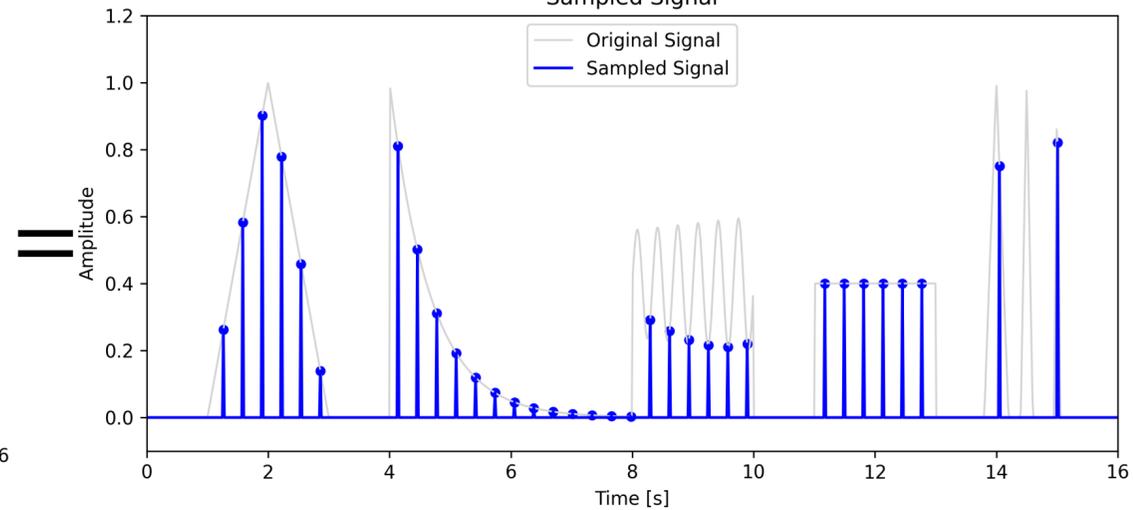
Signal Sampling and Aliasing



Impulse Train Sampling

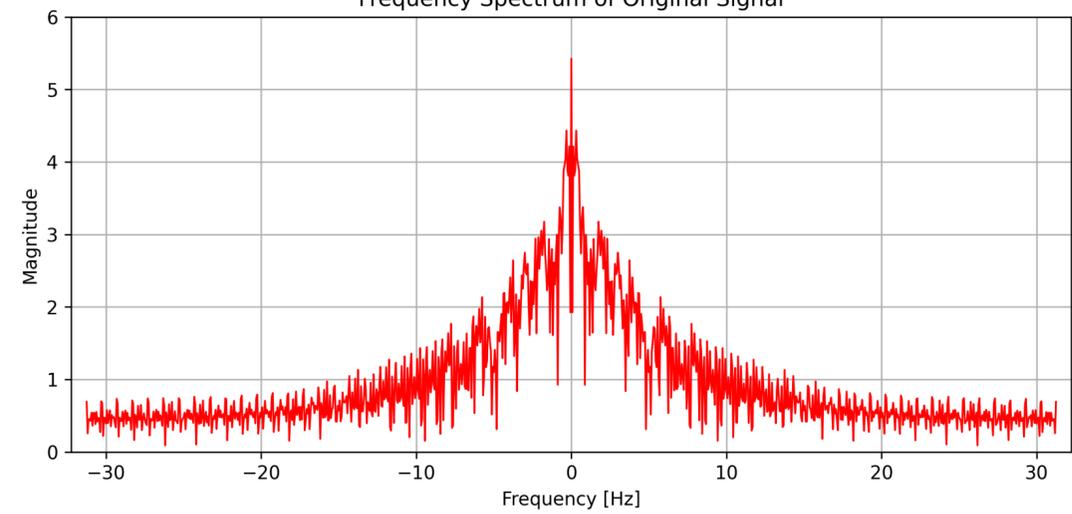


Sampled Signal



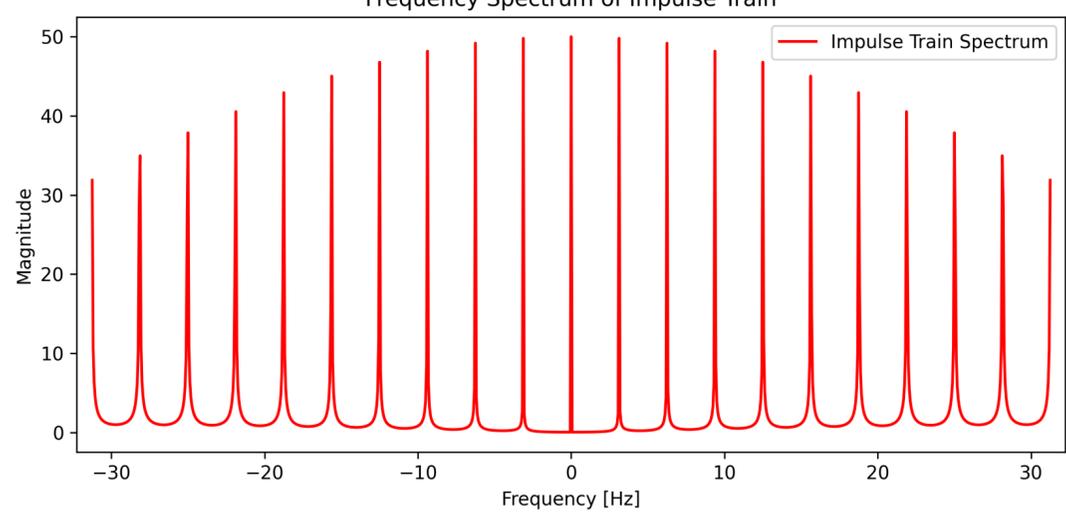
$\mathcal{F}$

Frequency Spectrum of Original Signal



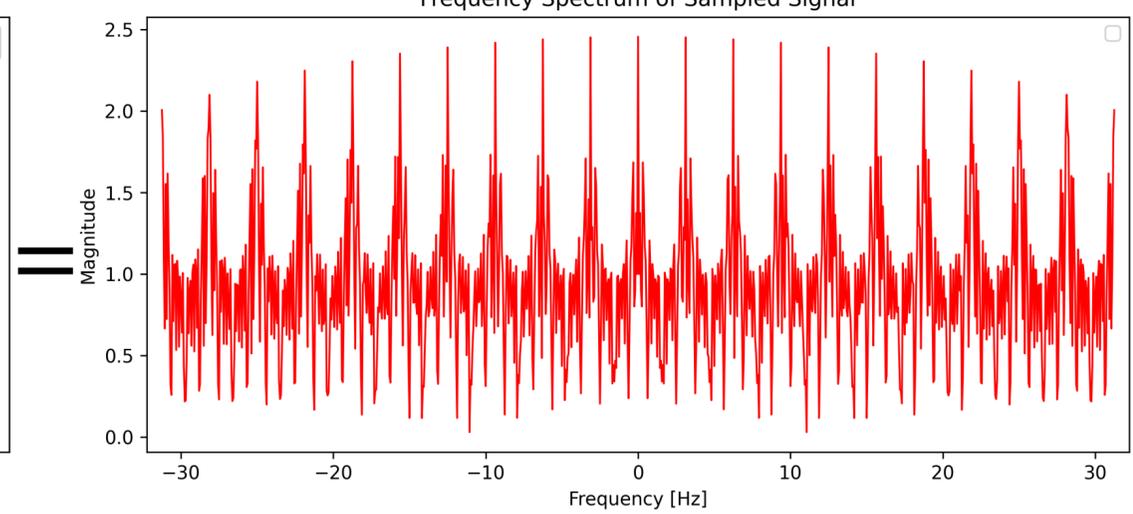
$\mathcal{F}$

Frequency Spectrum of Impulse Train



$\mathcal{F}$

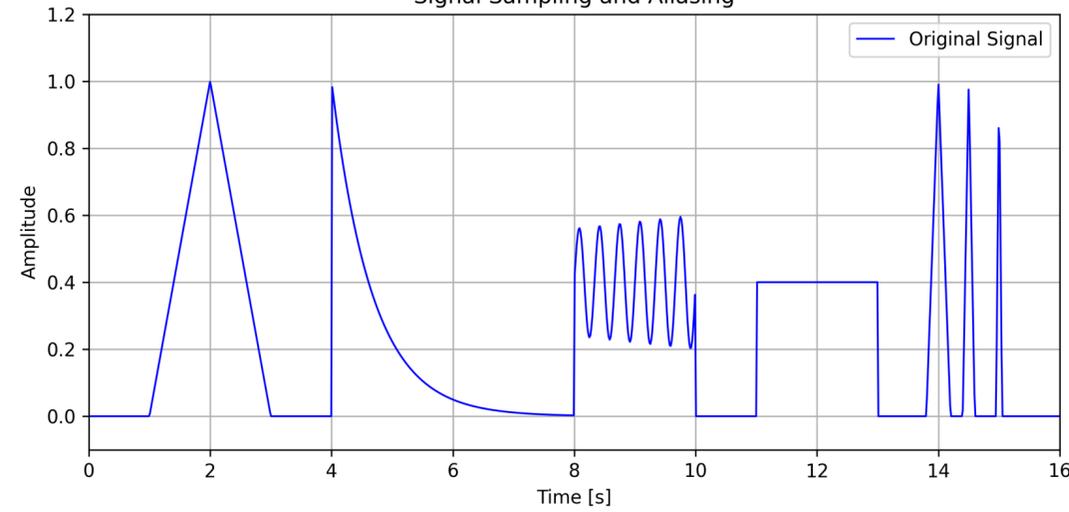
Frequency Spectrum of Sampled Signal



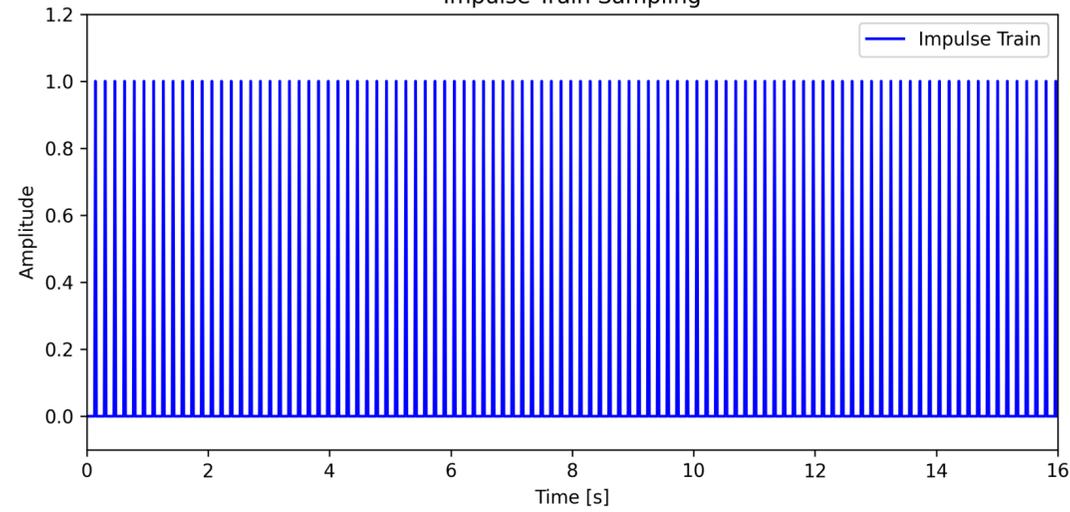
# Amostragem

1:10

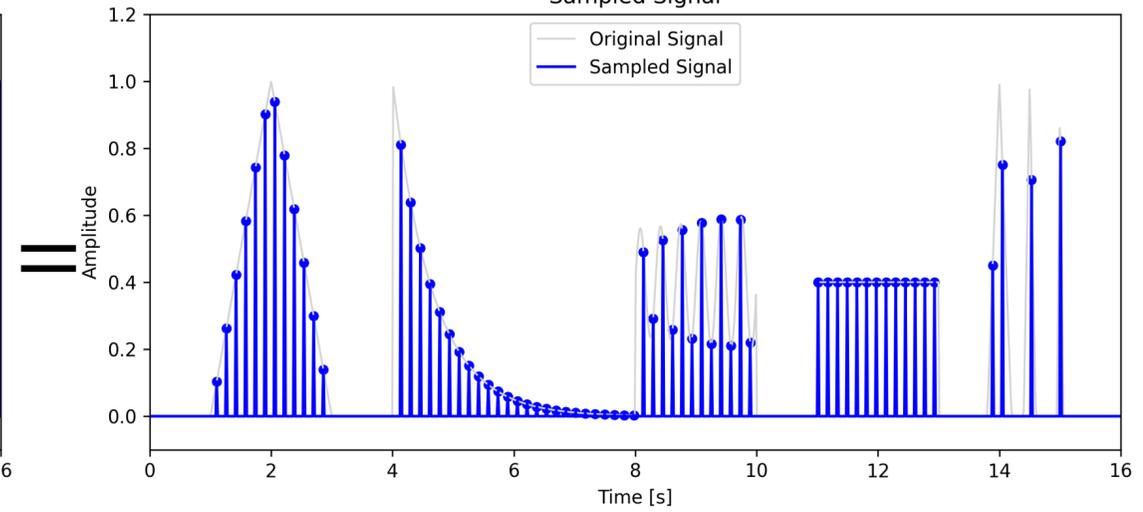
Signal Sampling and Aliasing



Impulse Train Sampling

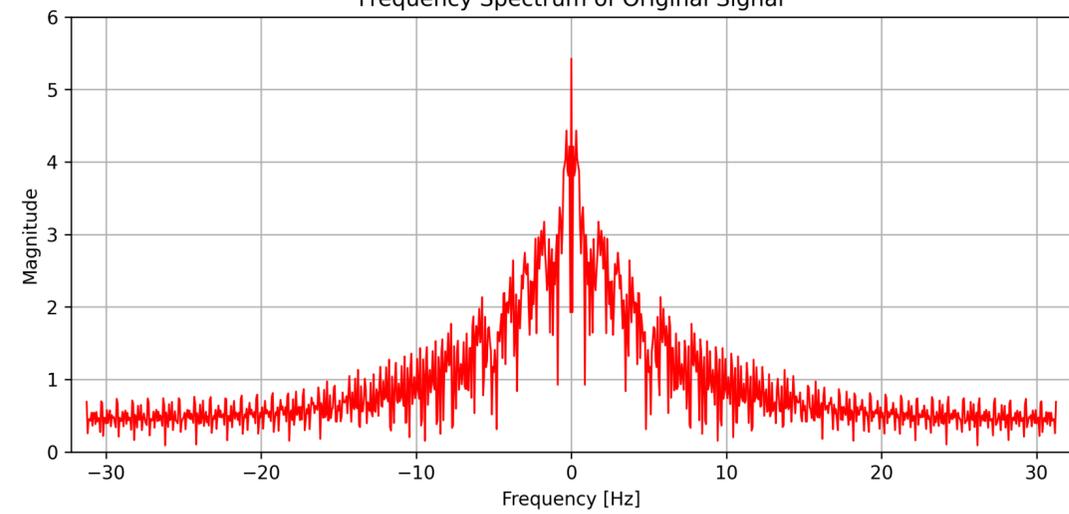


Sampled Signal



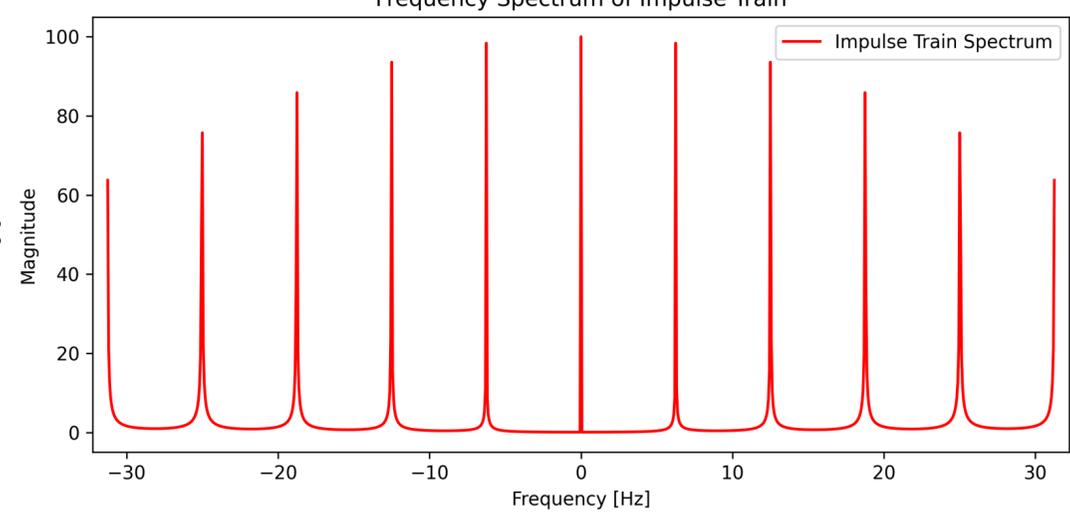
$\mathcal{F}$

Frequency Spectrum of Original Signal



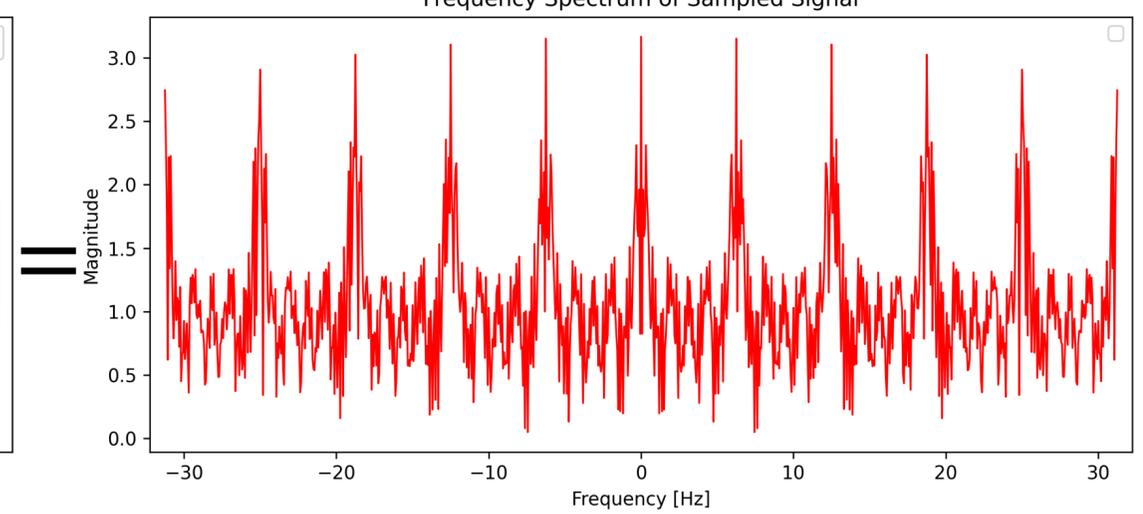
$\mathcal{F}$

Frequency Spectrum of Impulse Train



$\mathcal{F}$

Frequency Spectrum of Sampled Signal



•

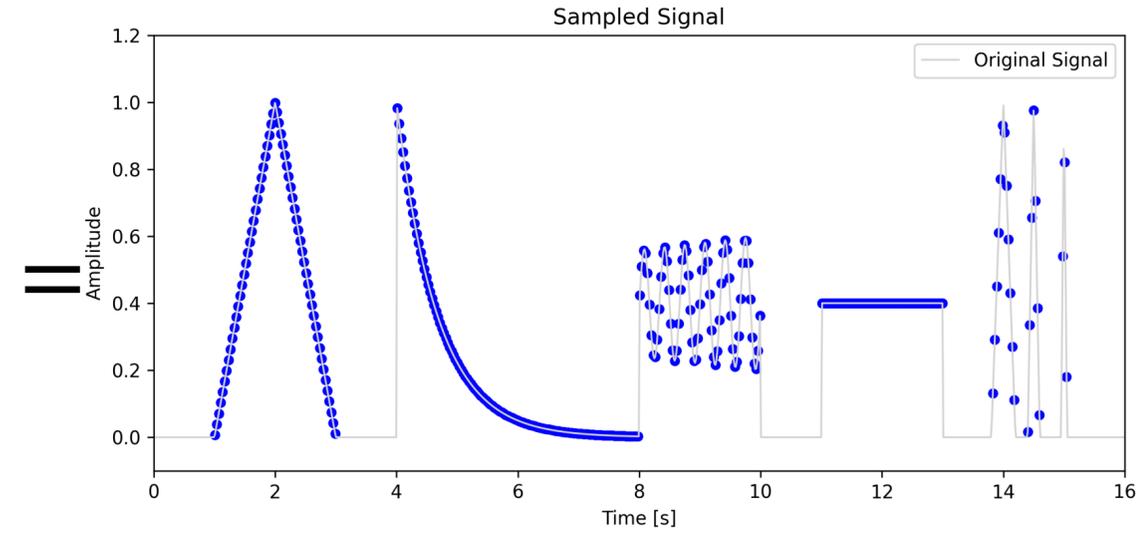
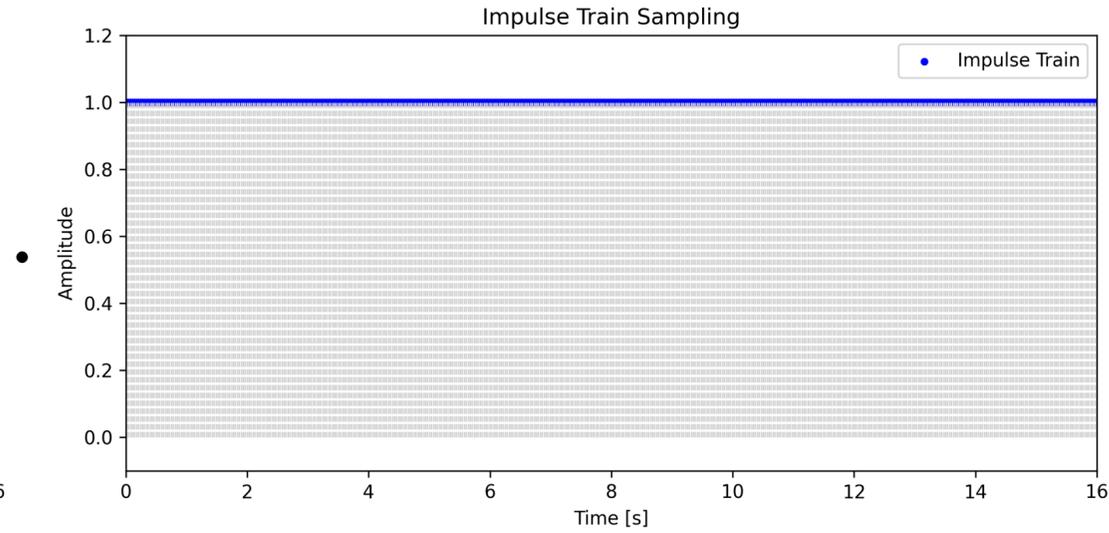
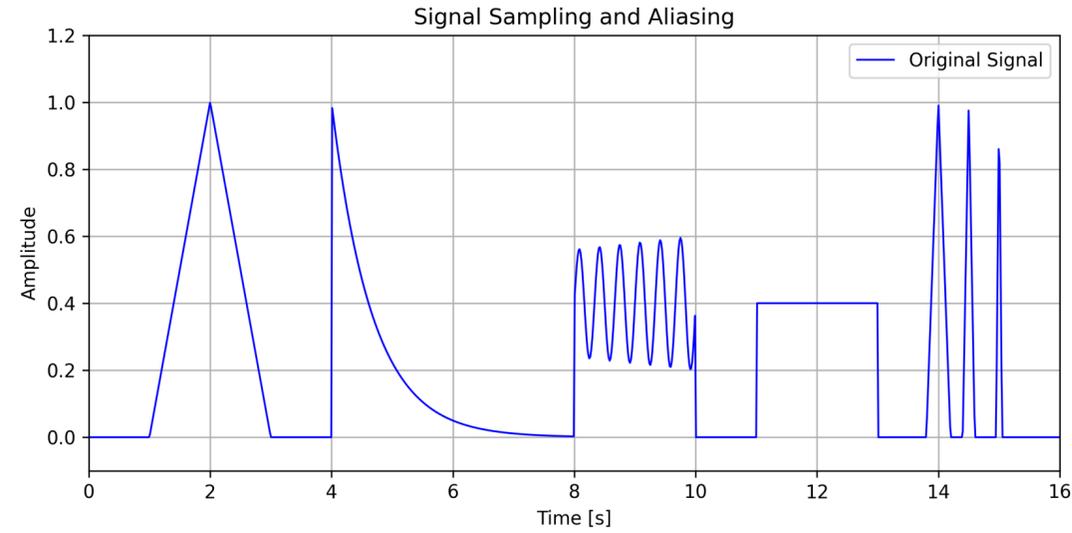
\*

||

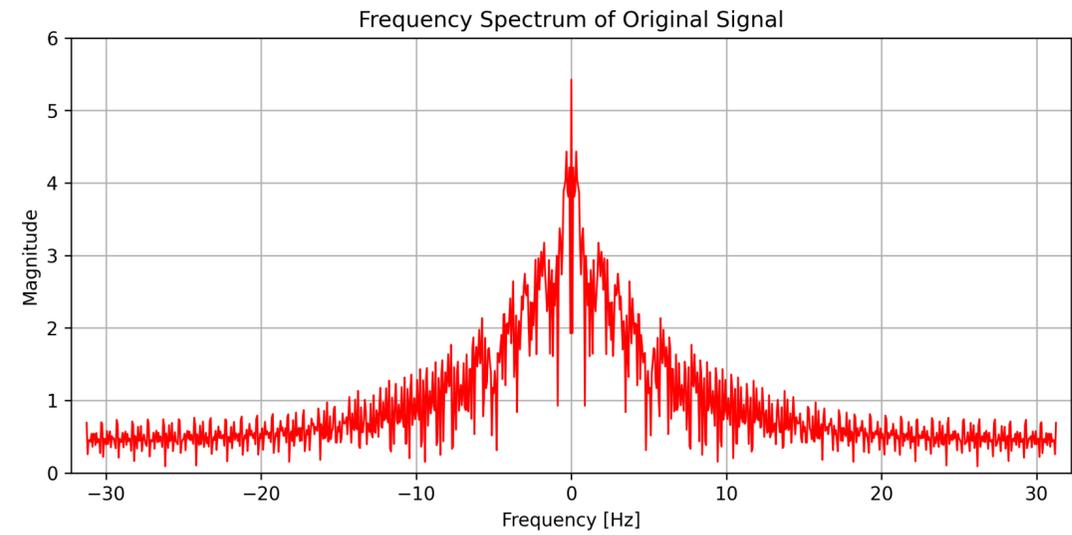
||

# Amostragem

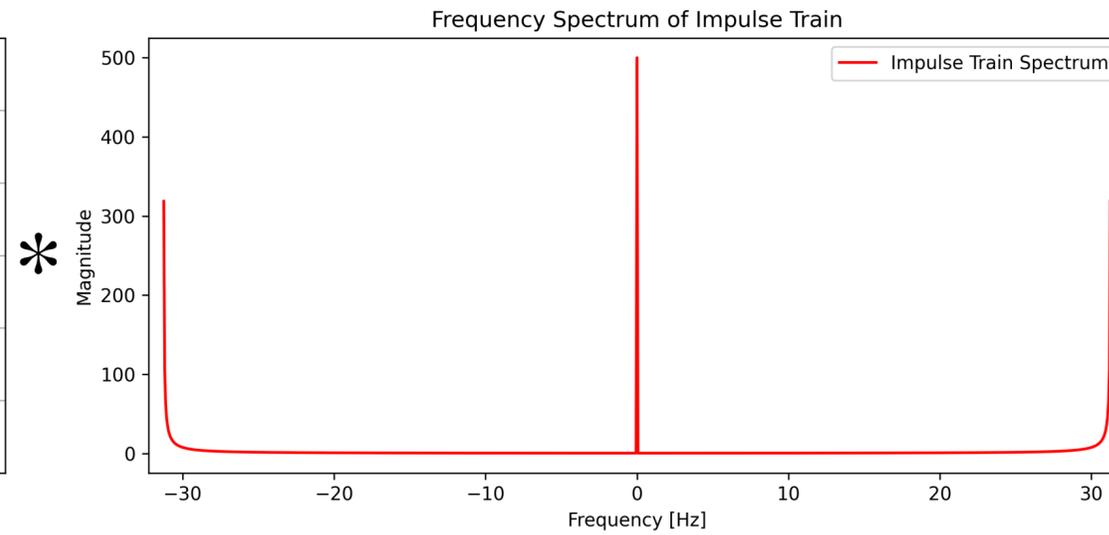
1:2



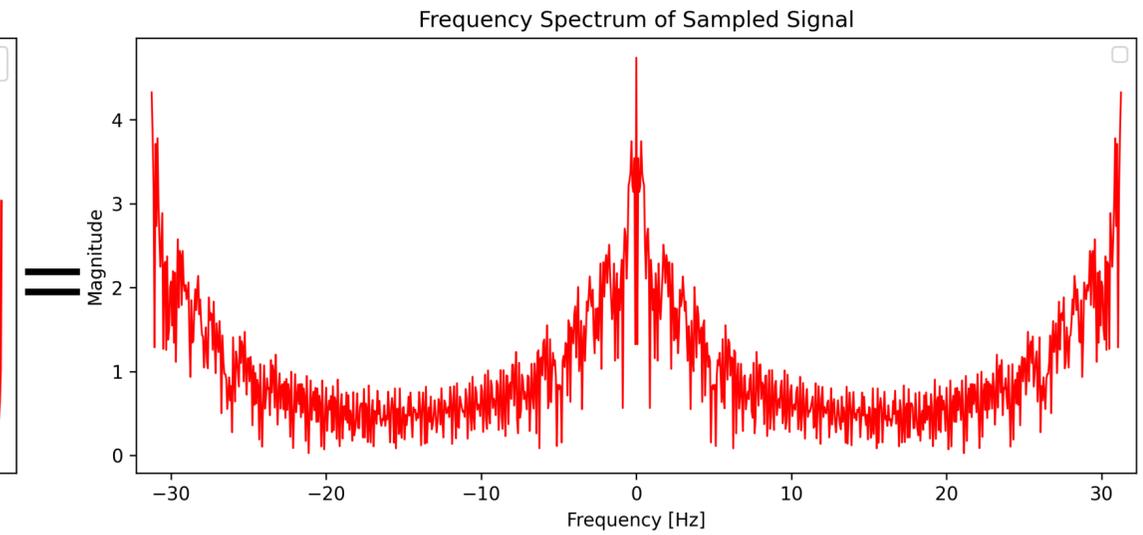
$\mathcal{F}$



$\mathcal{F}$

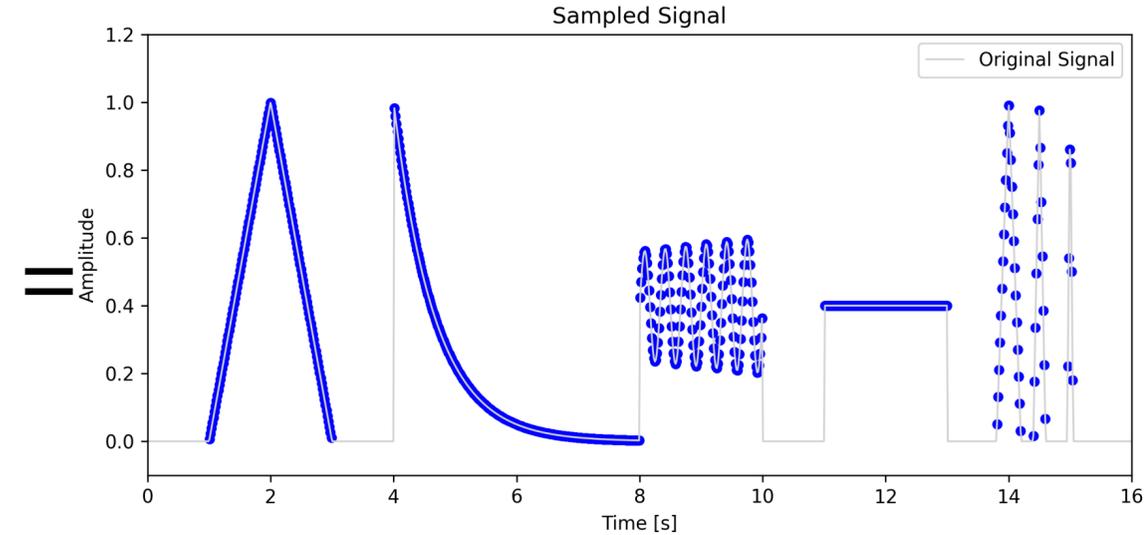
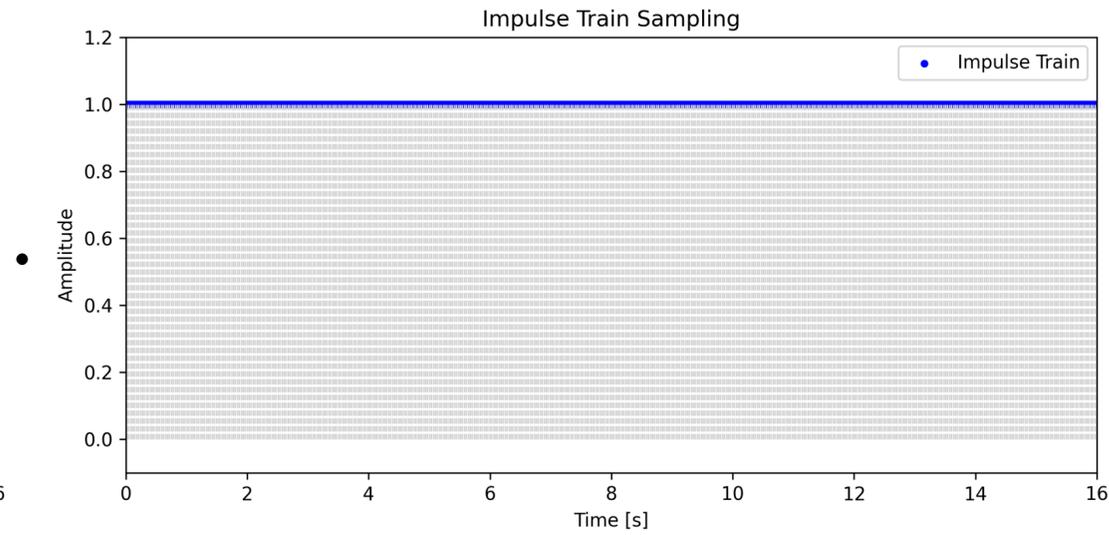
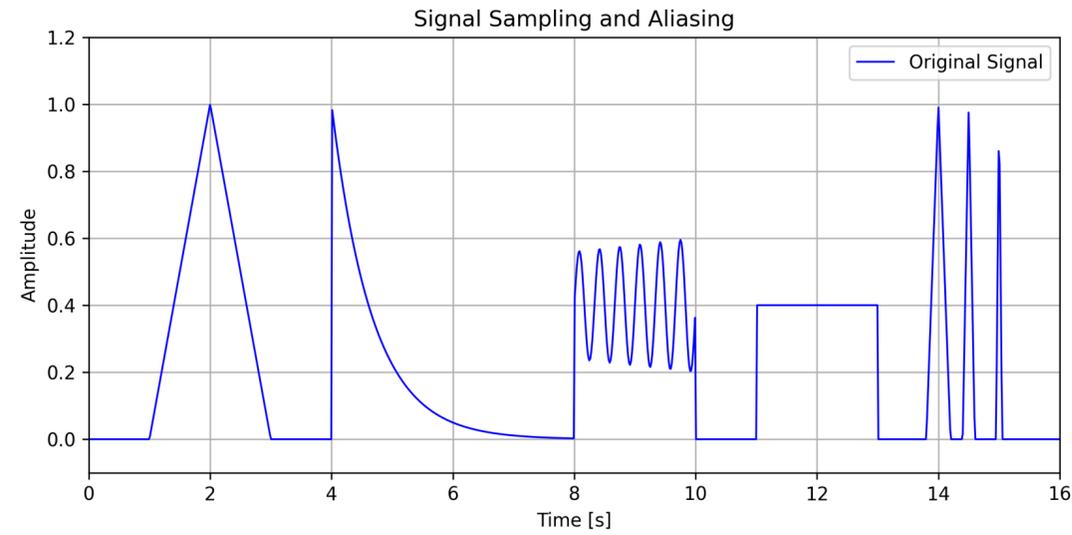


$\mathcal{F}$

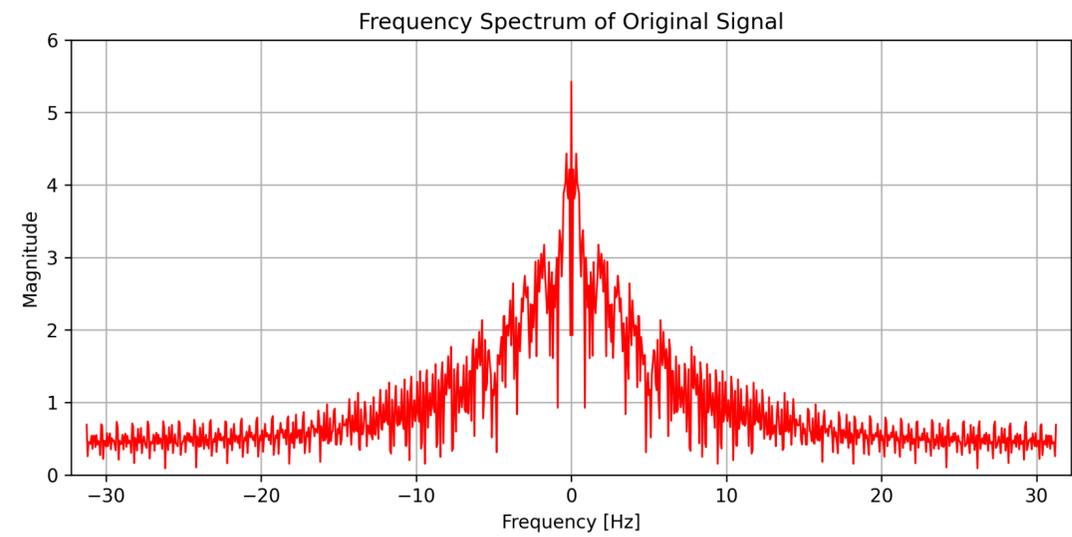


# Amostragem

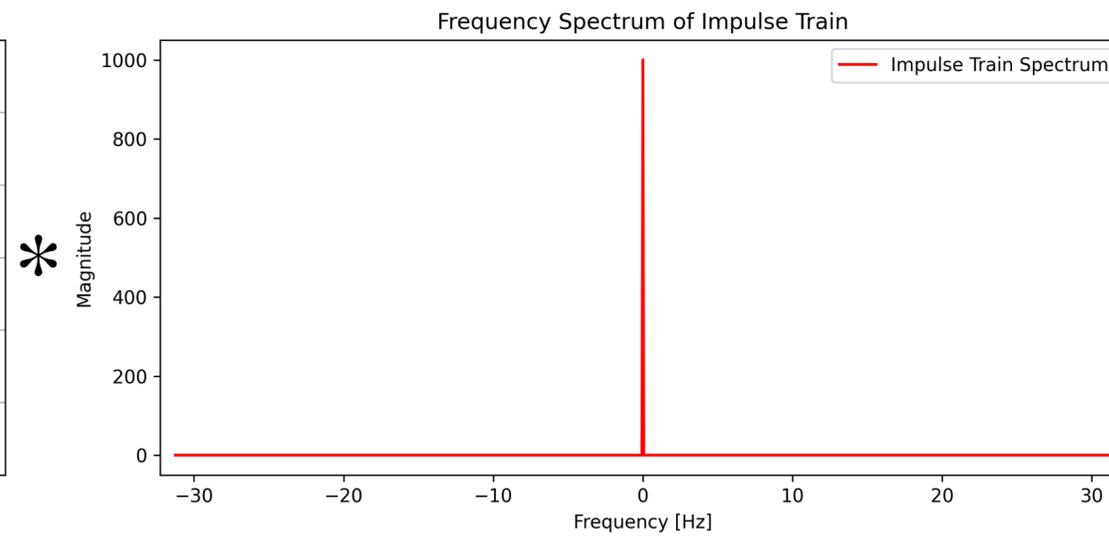
1:1



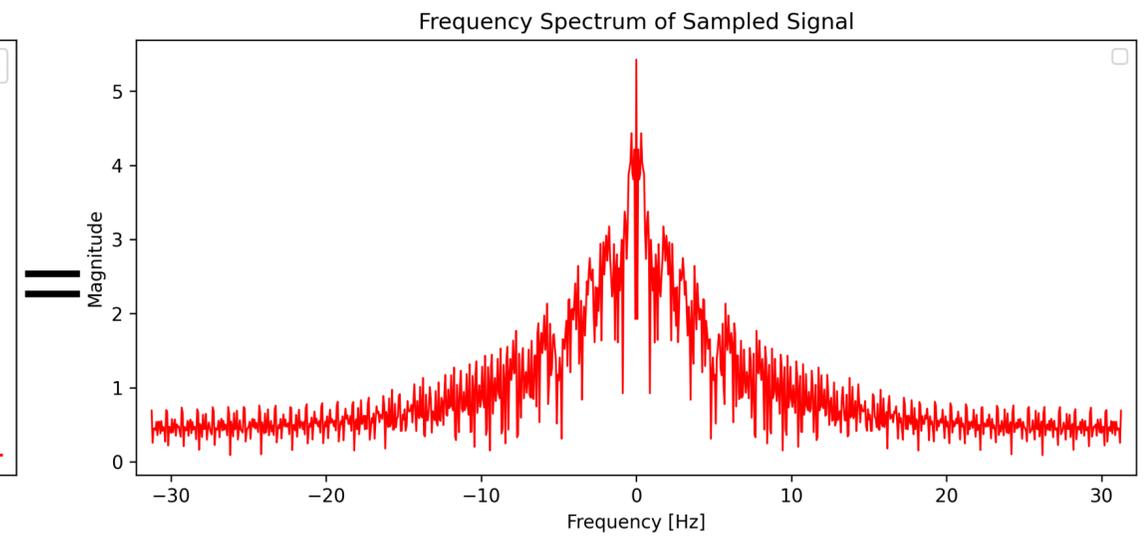
$\mathcal{F}$



$\mathcal{F}$

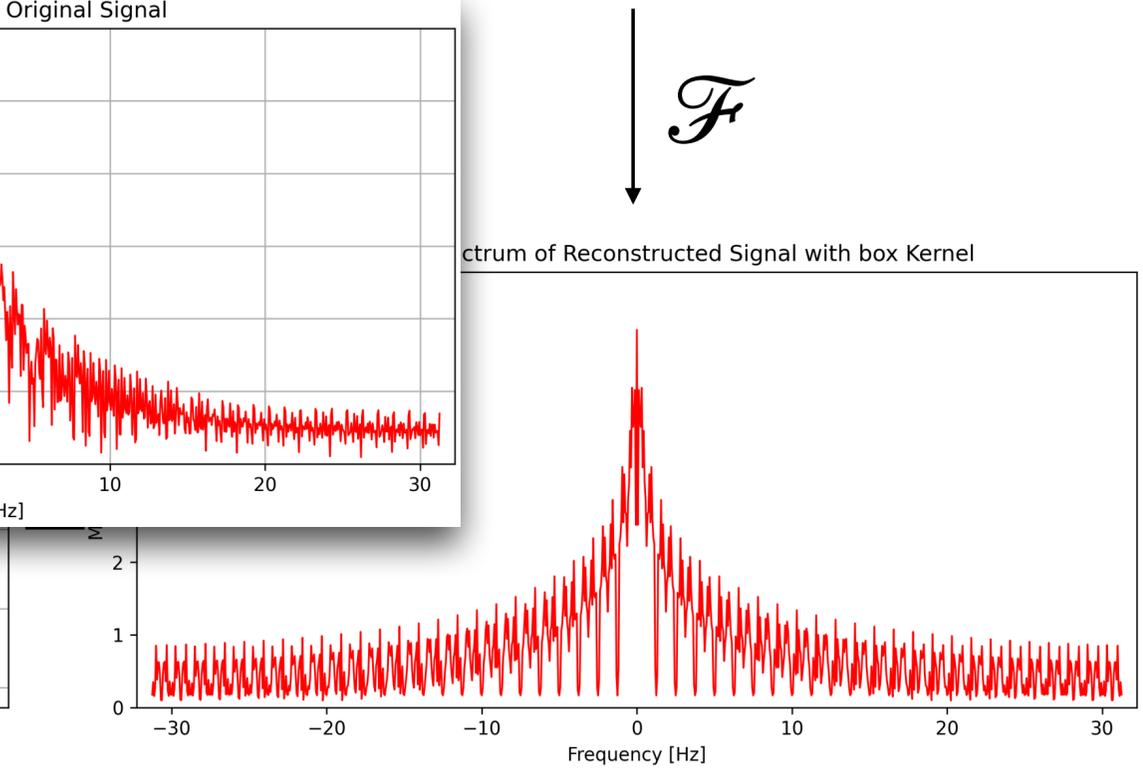
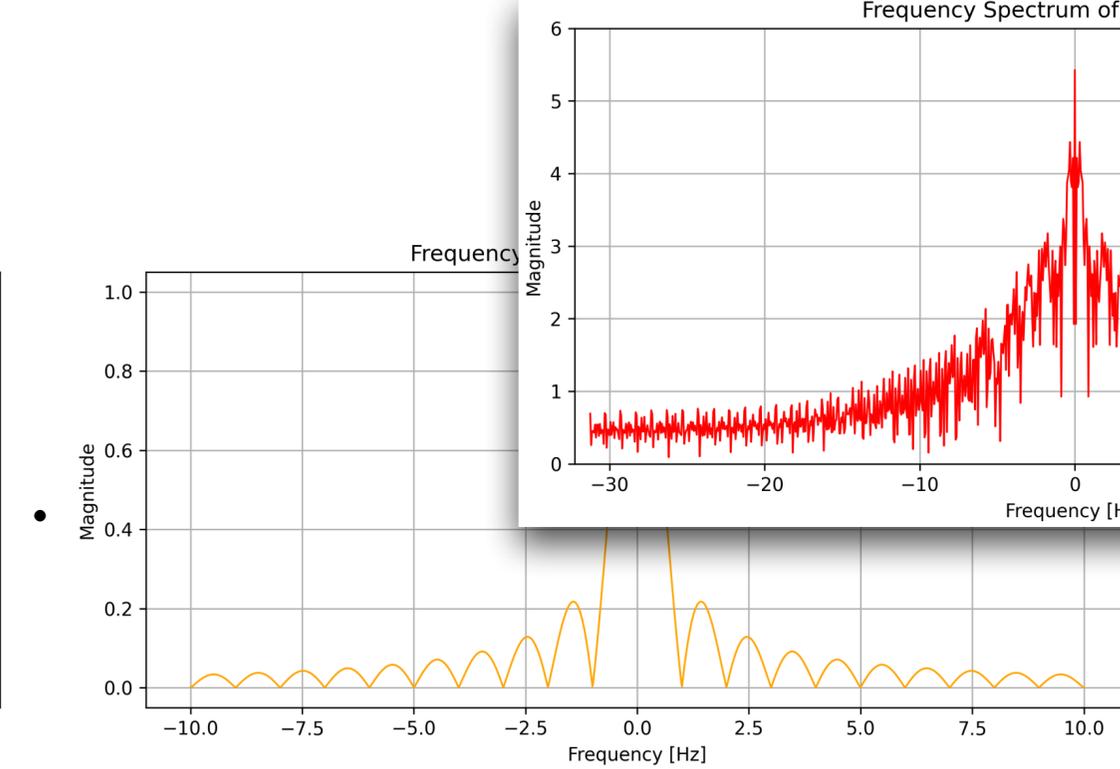
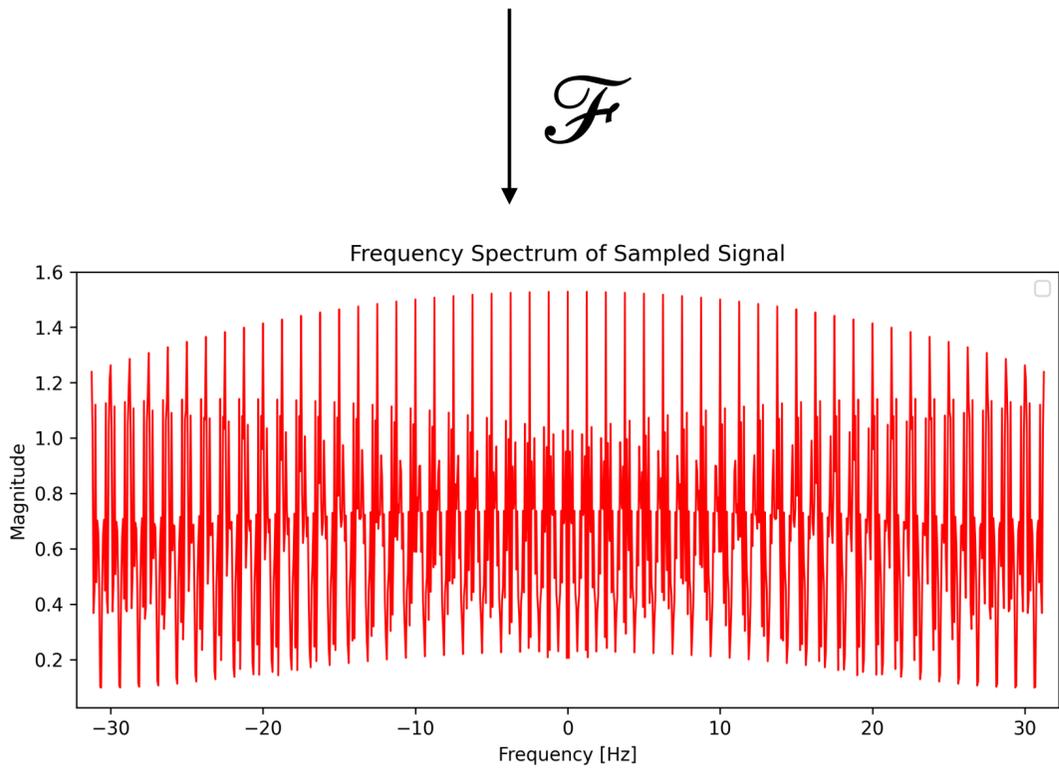
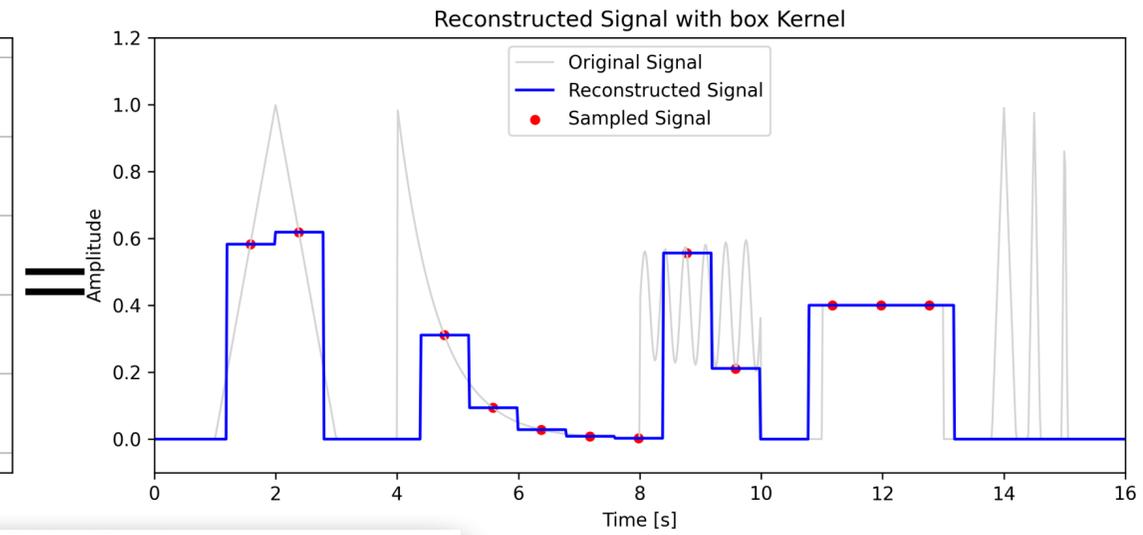
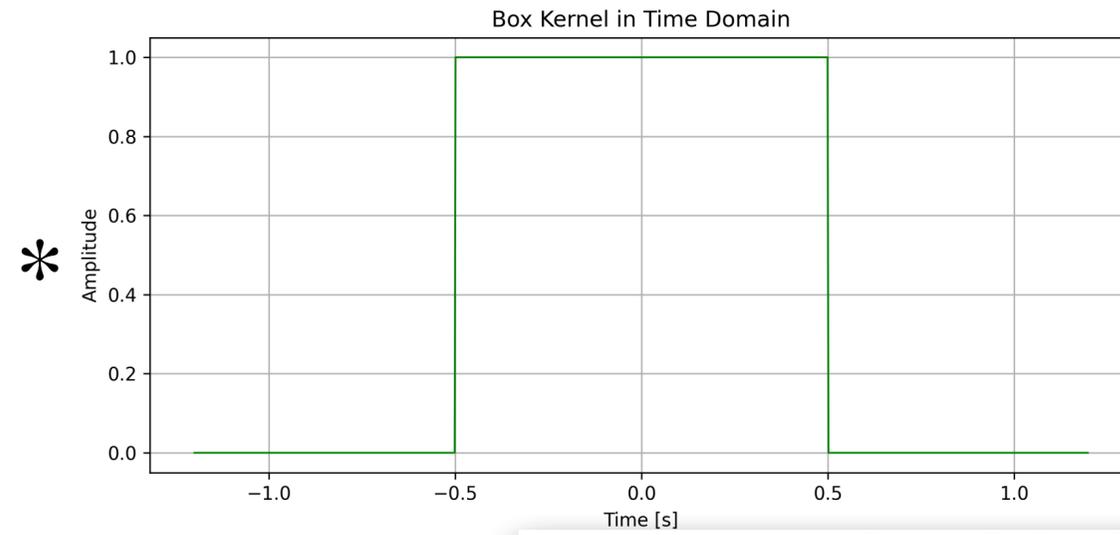
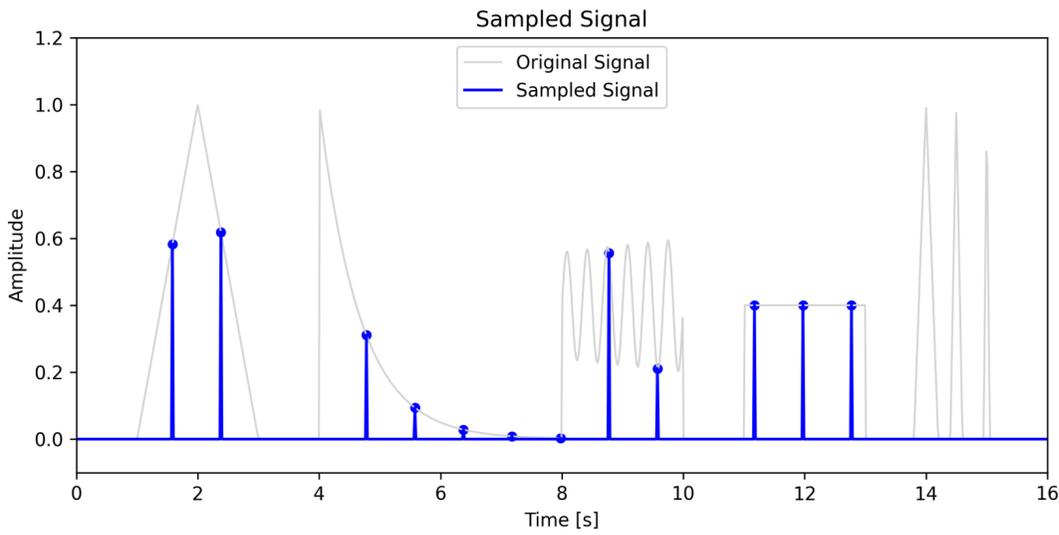


$\mathcal{F}$



# Reconstrução

1:50 box



$\mathcal{F}$

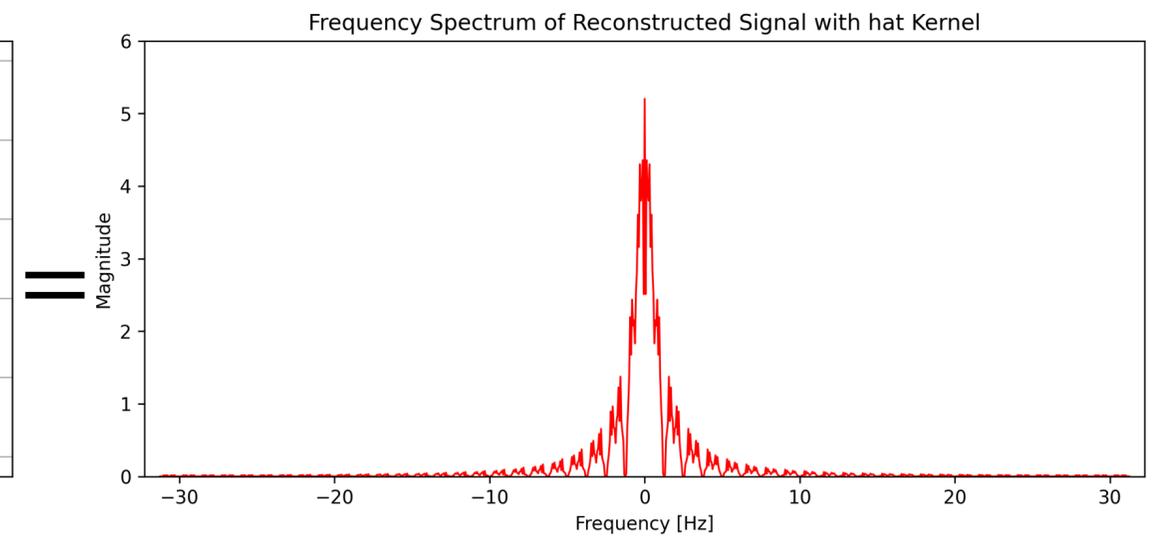
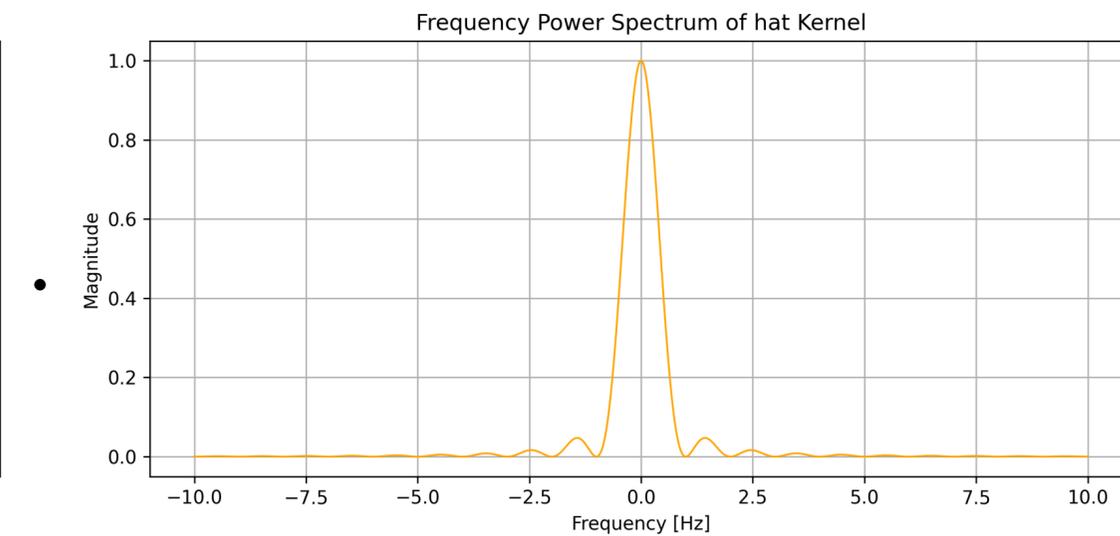
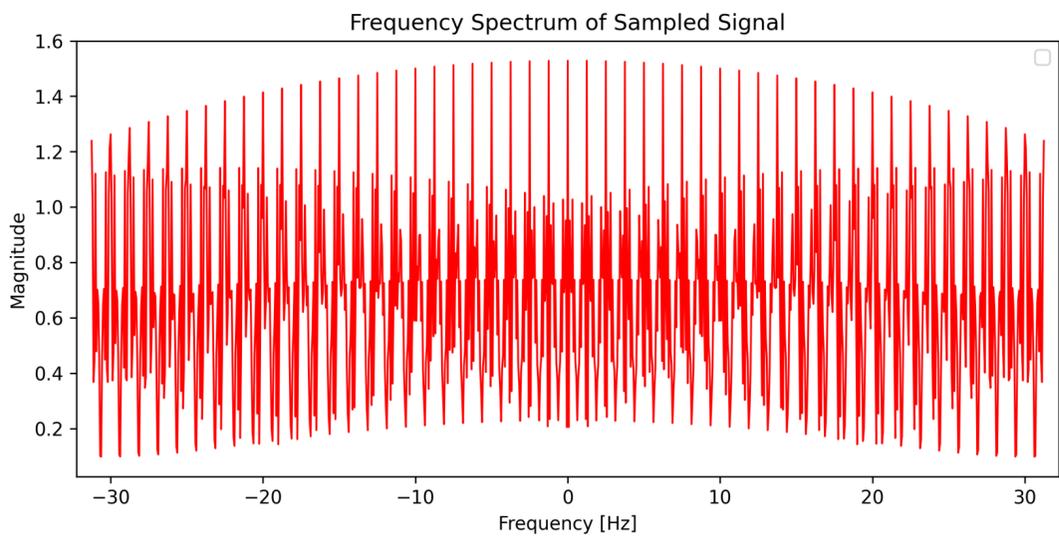
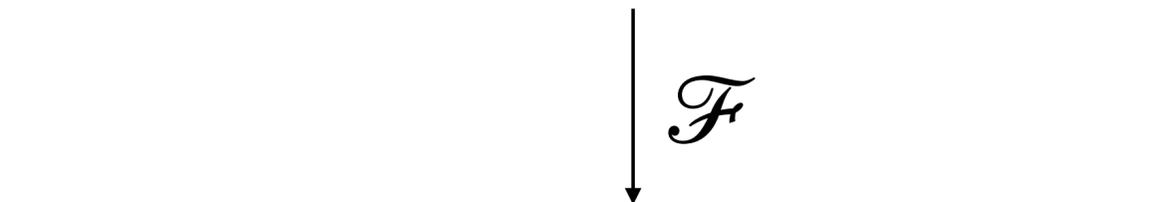
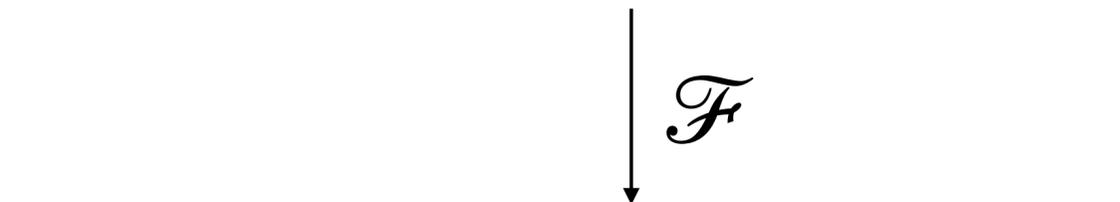
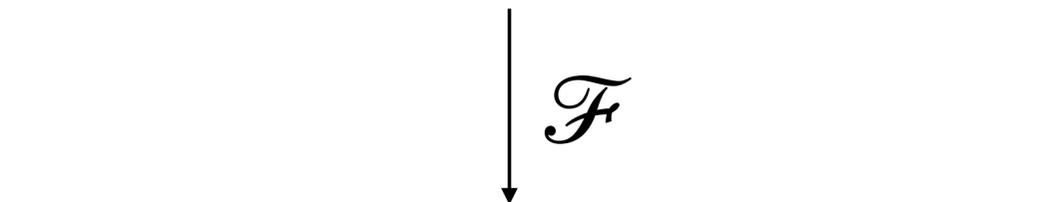
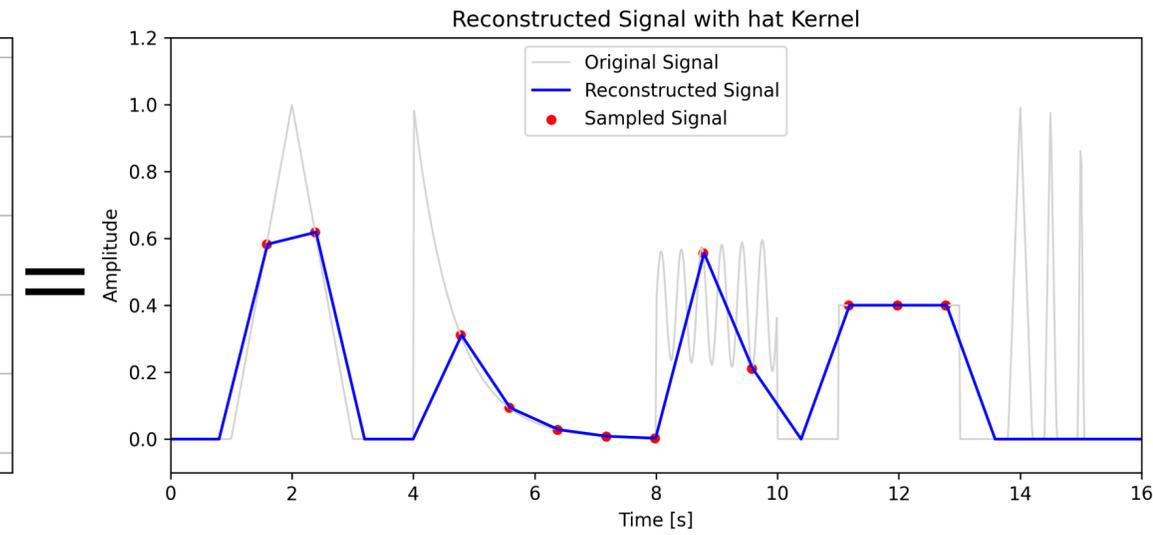
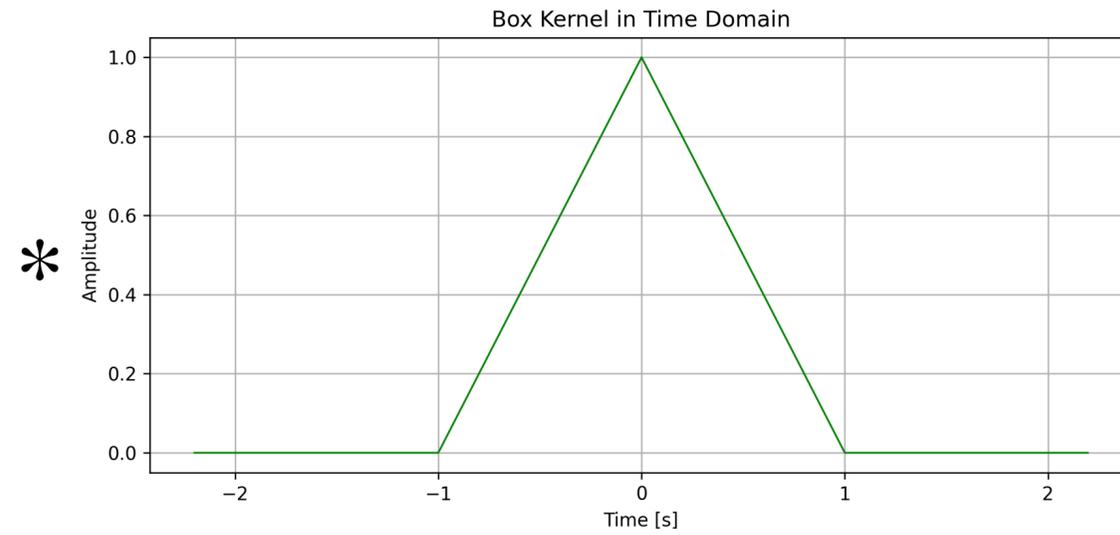
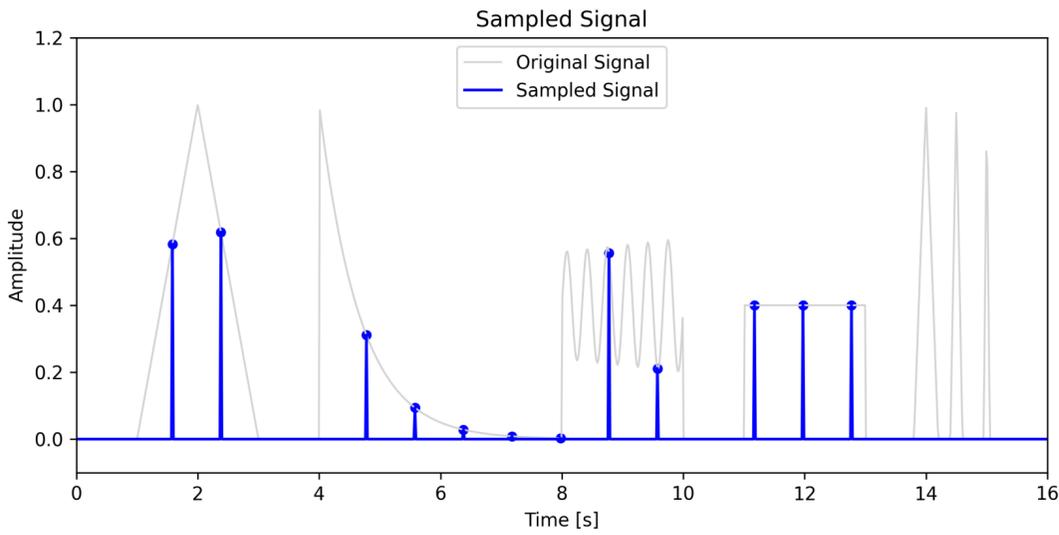
$\mathcal{F}$

\*

=

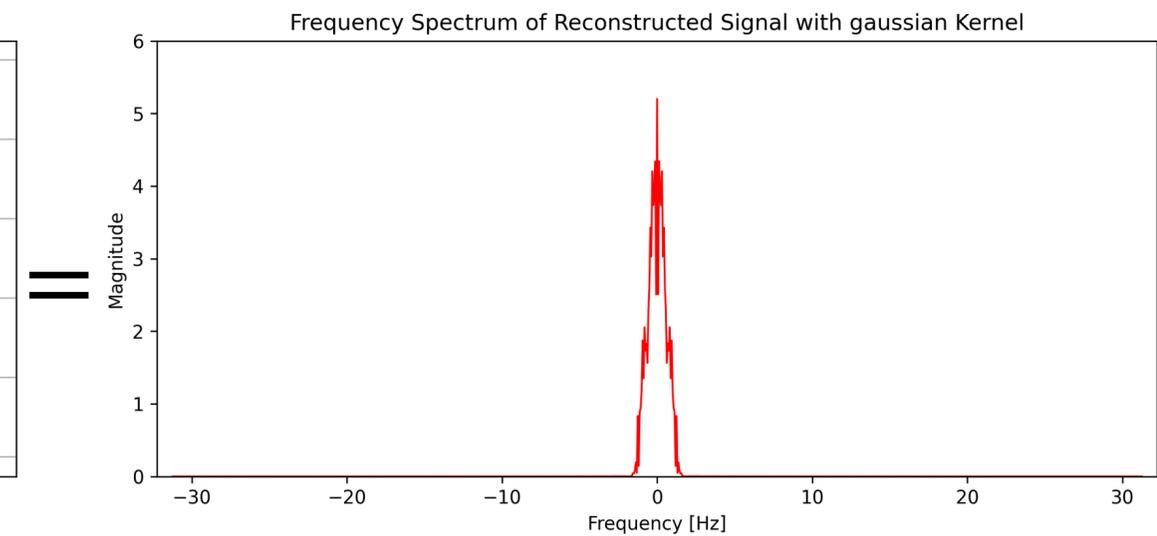
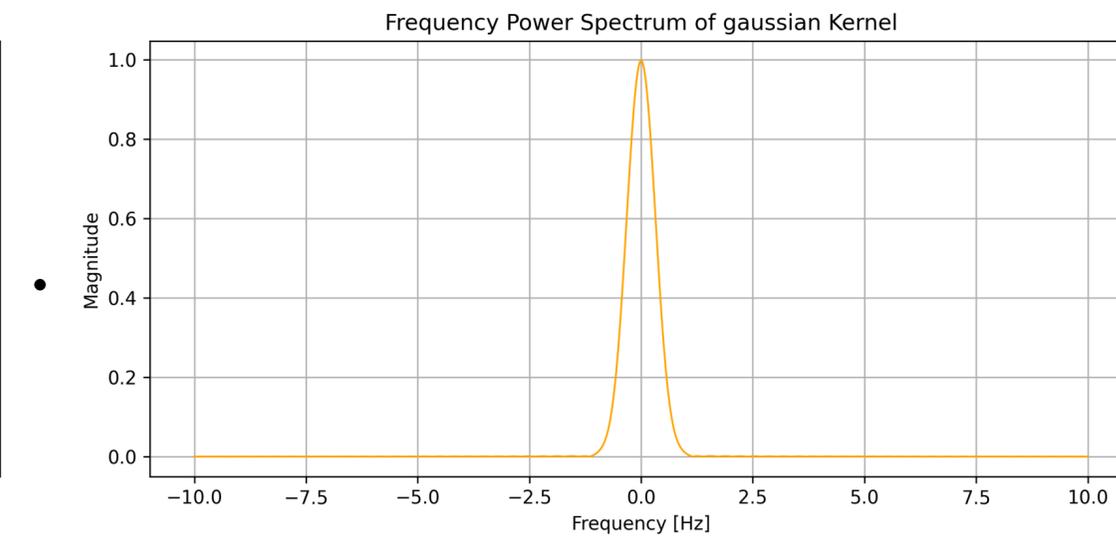
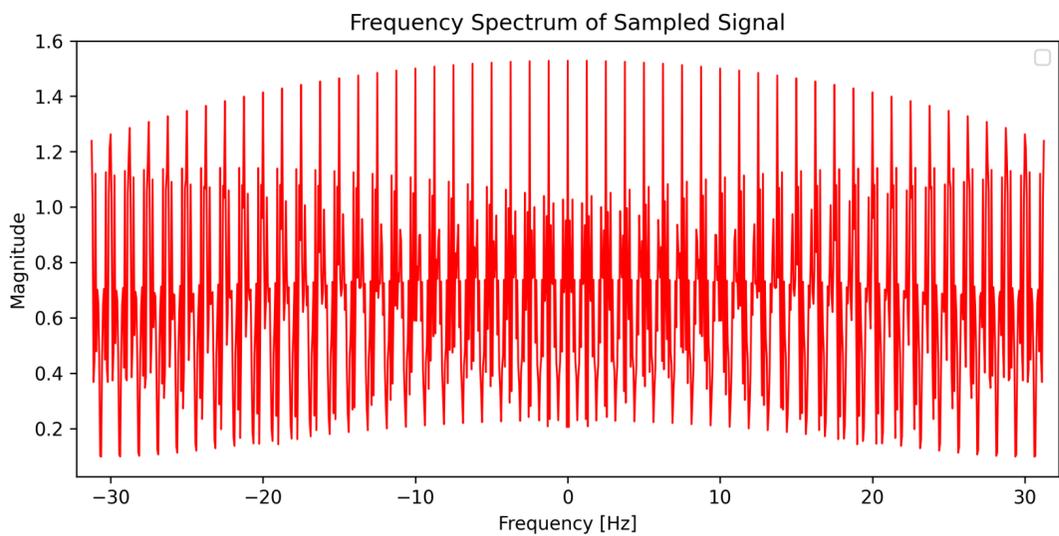
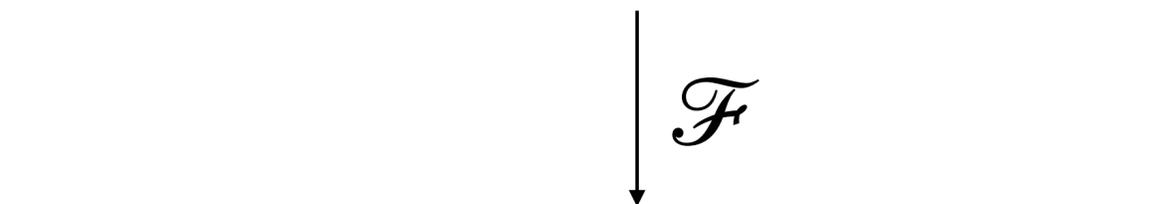
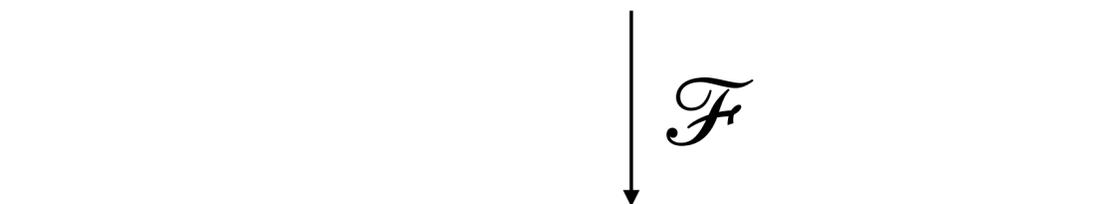
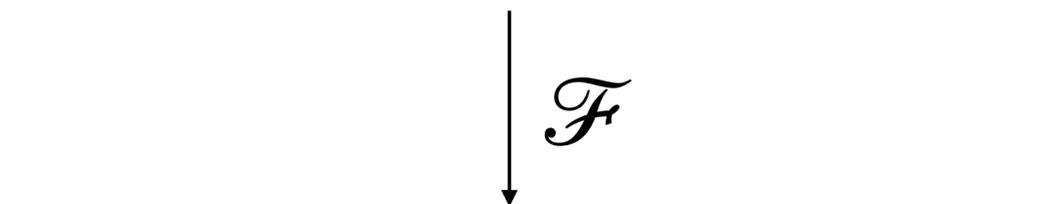
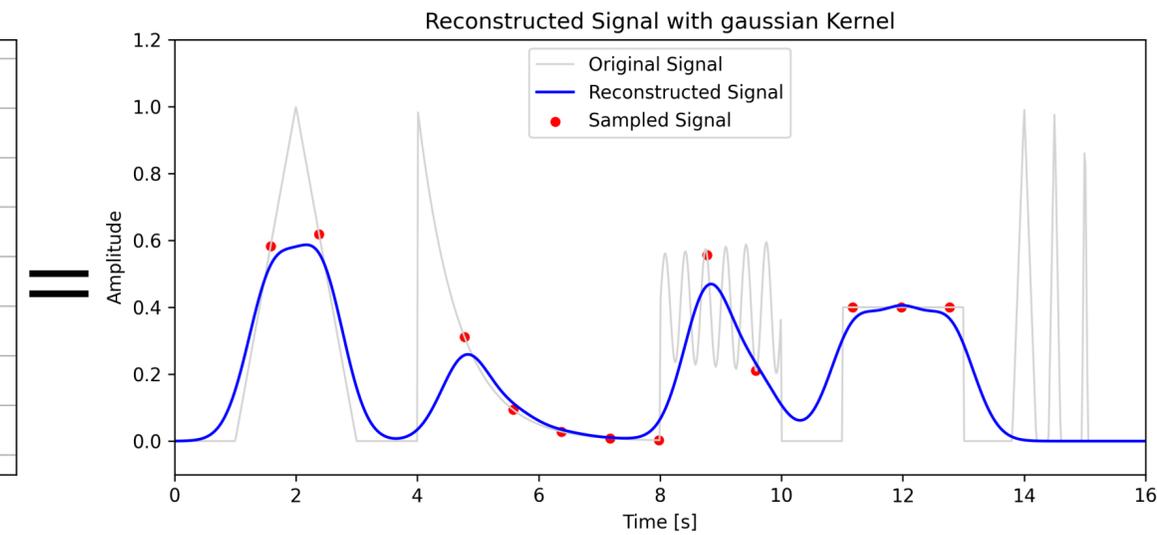
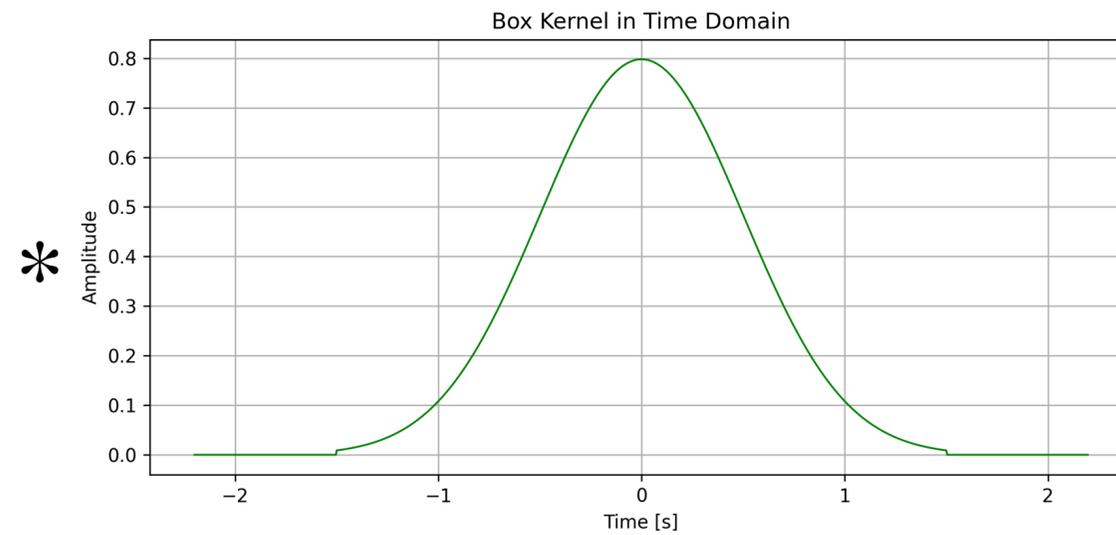
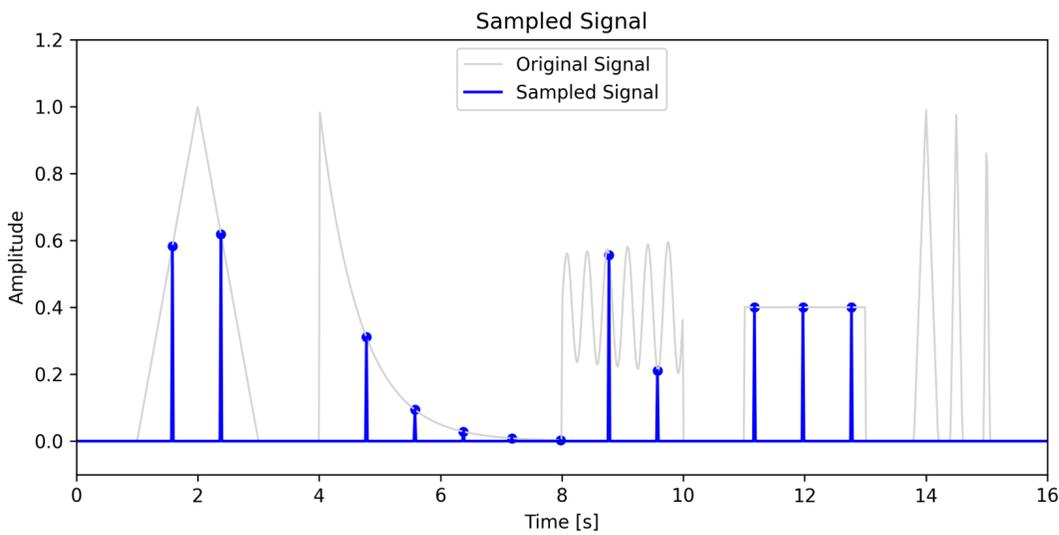
# Reconstrução

1:50 hat



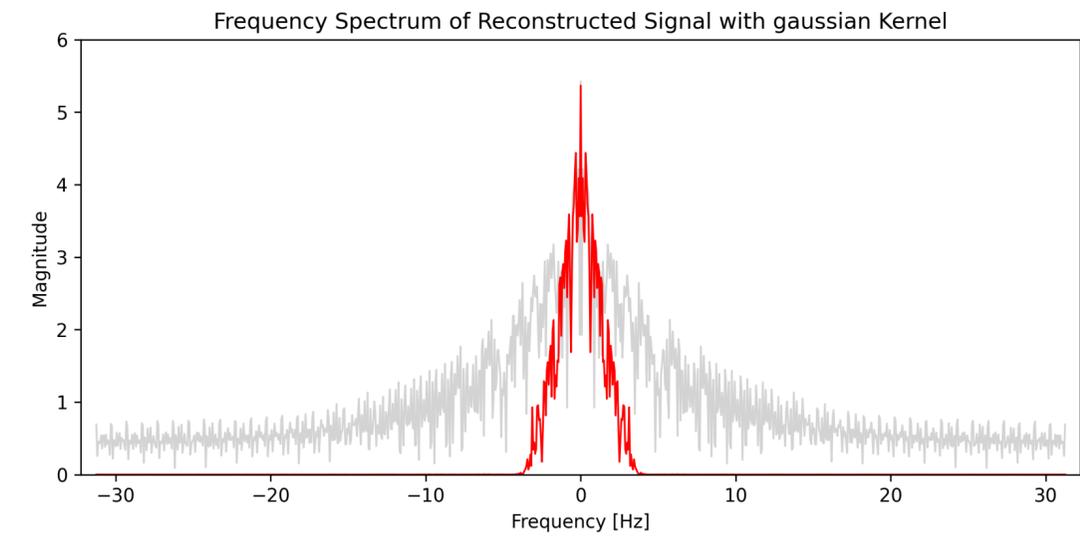
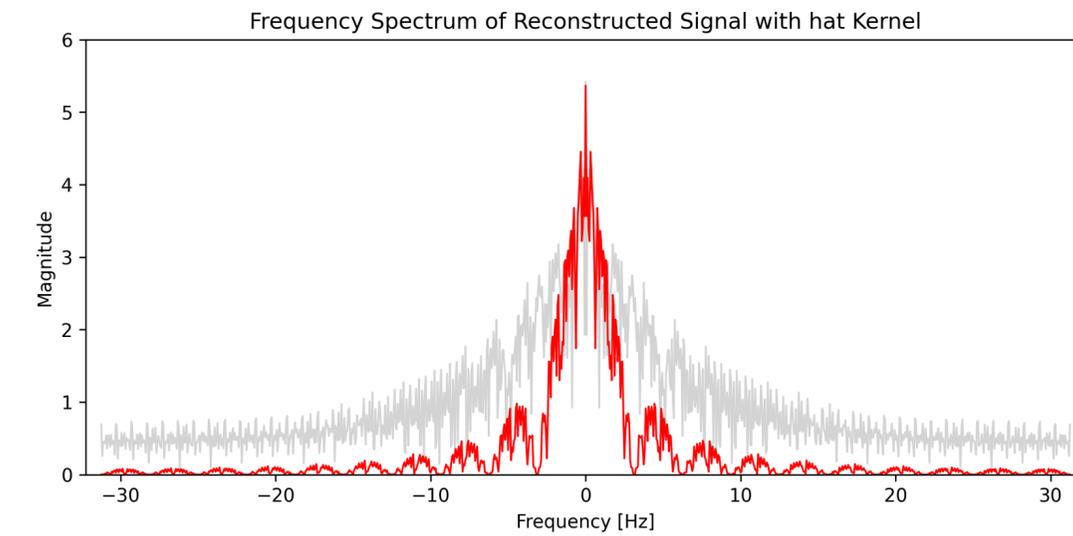
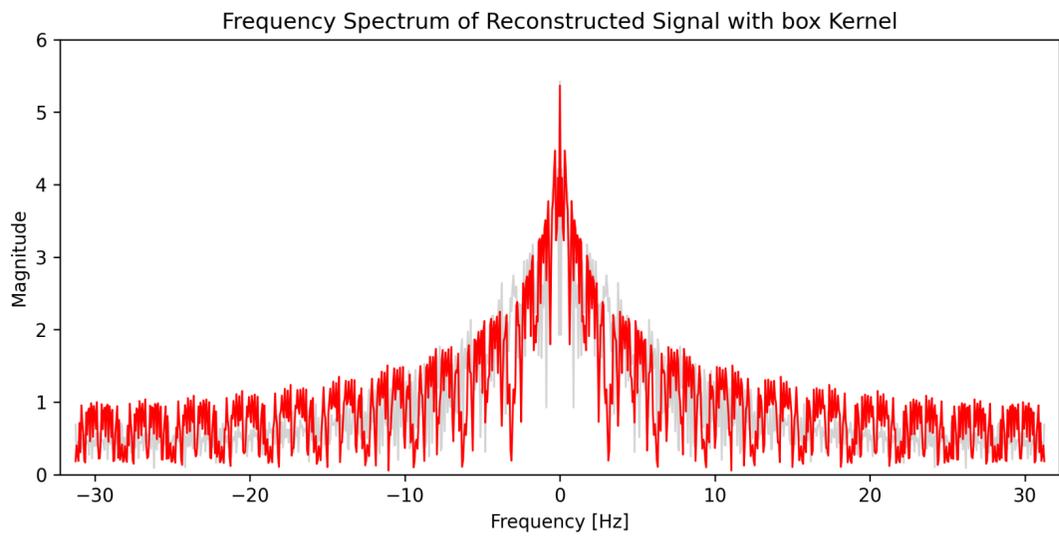
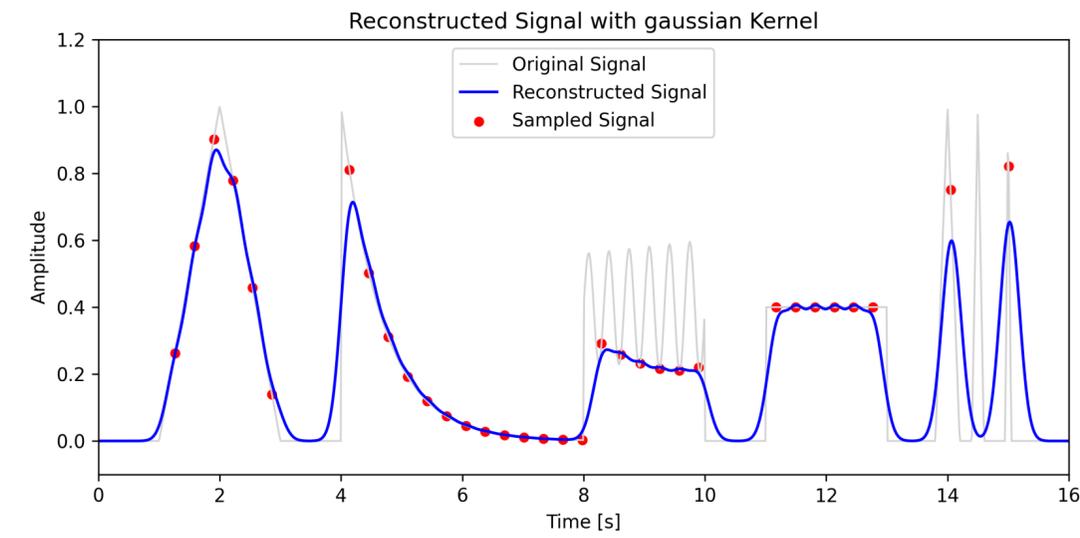
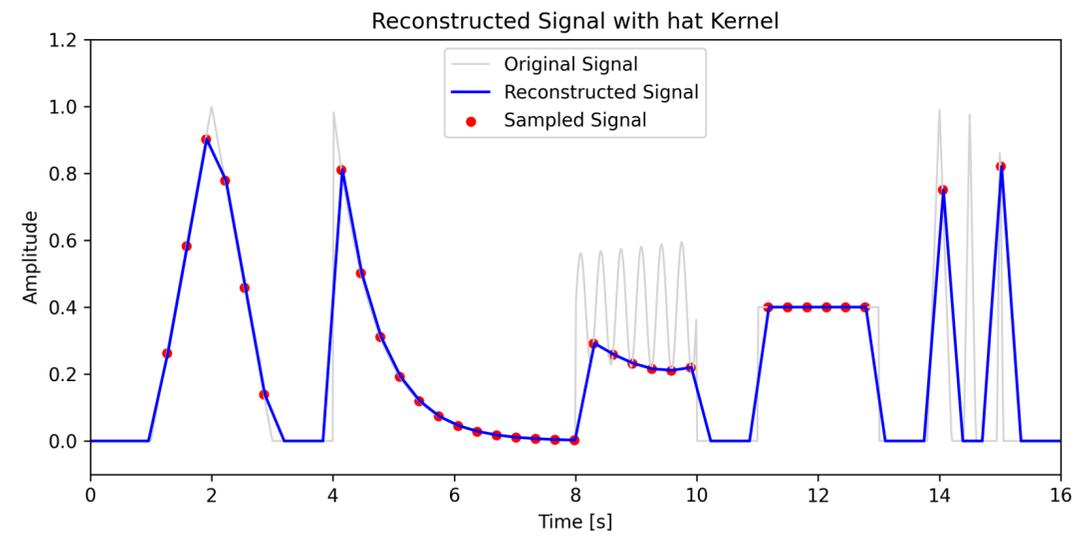
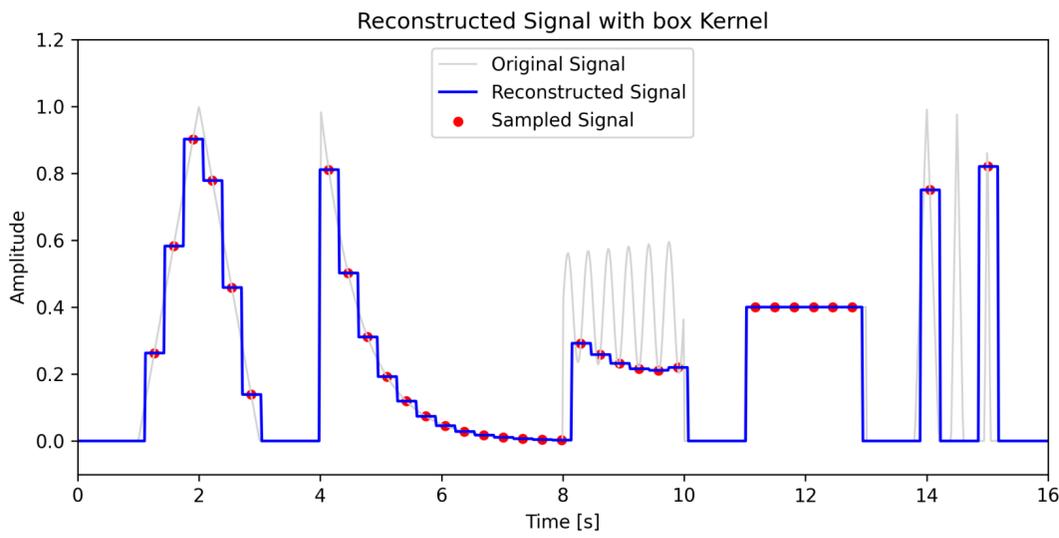
# Reconstrução

1:50 gaussian



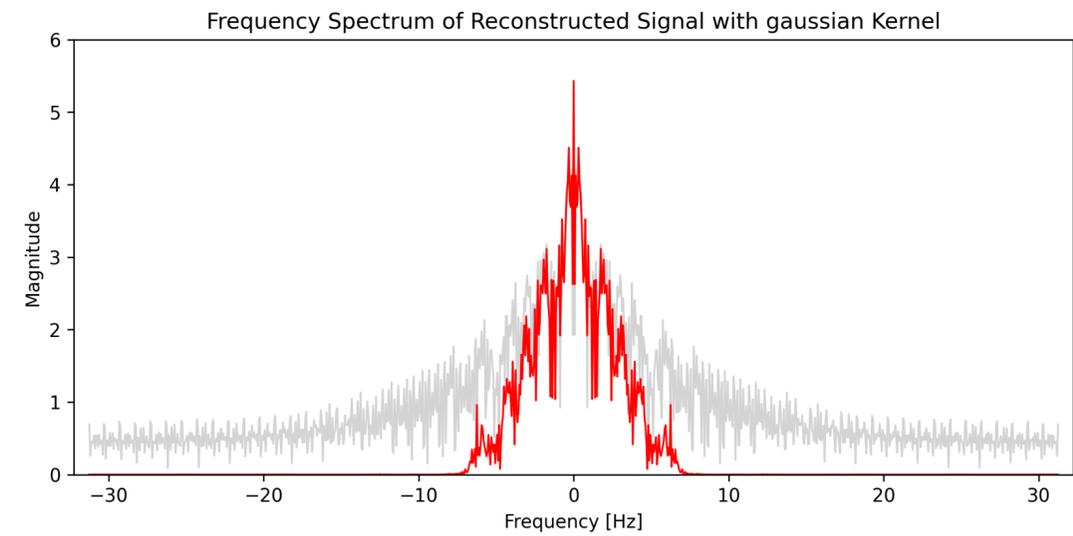
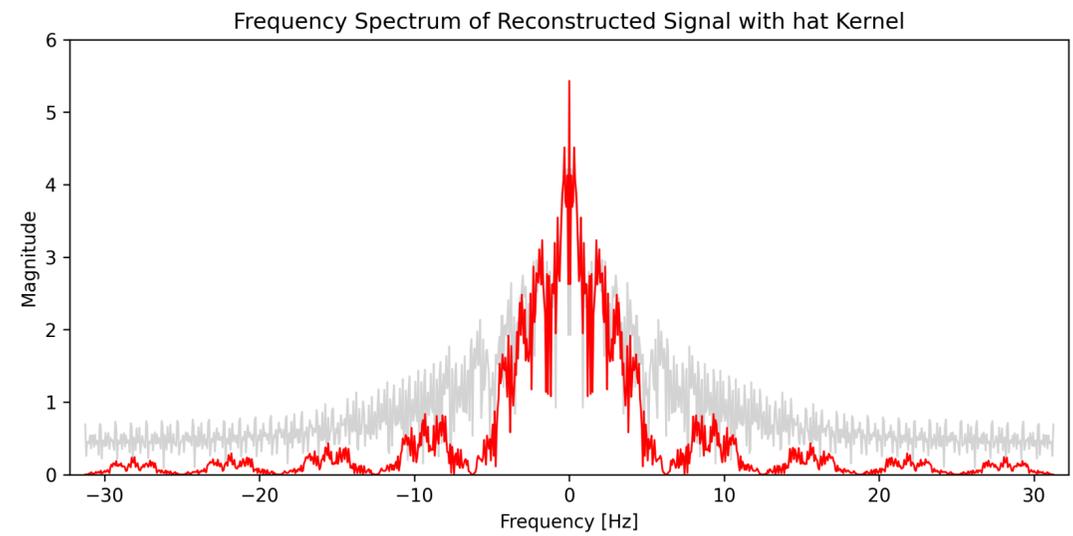
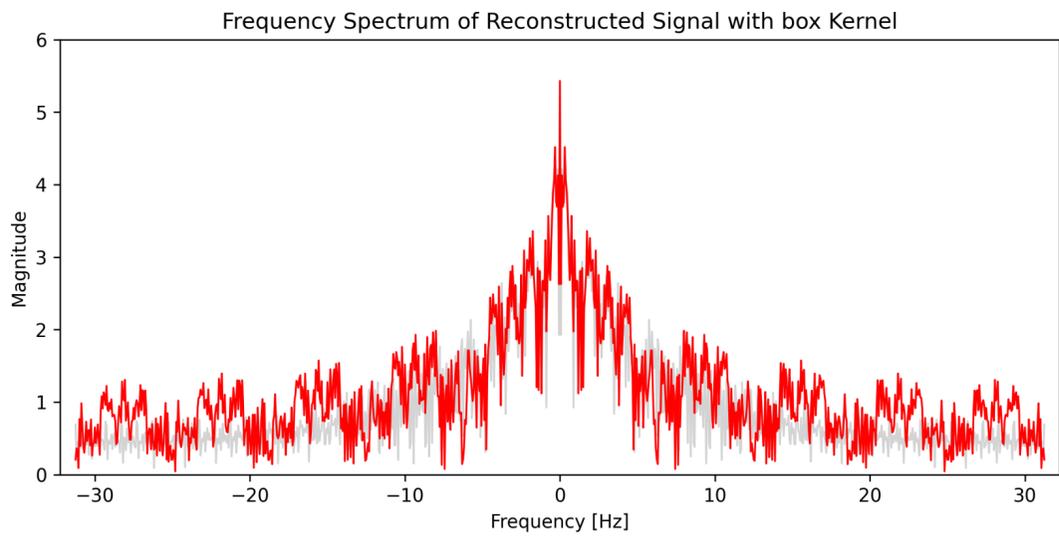
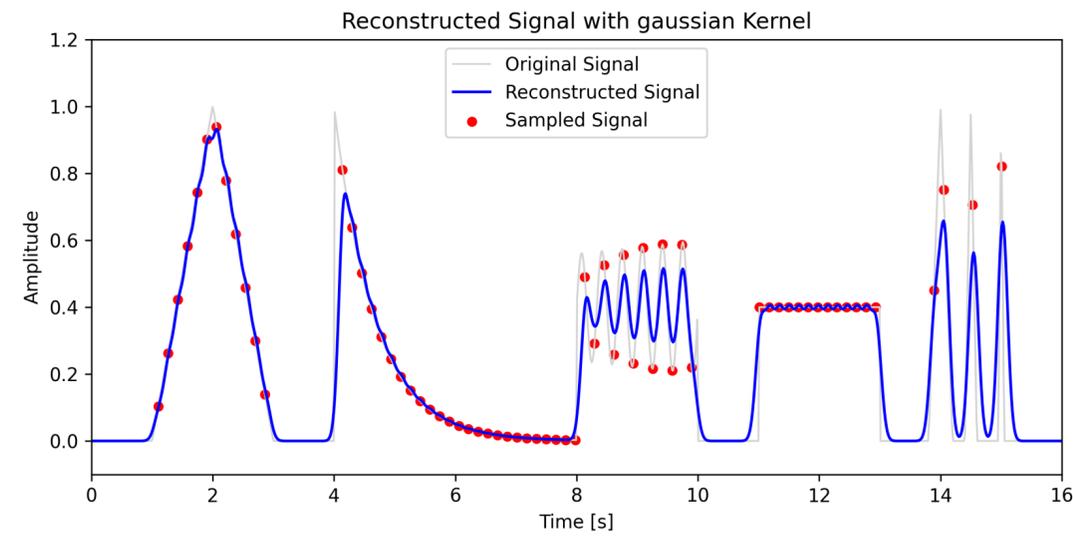
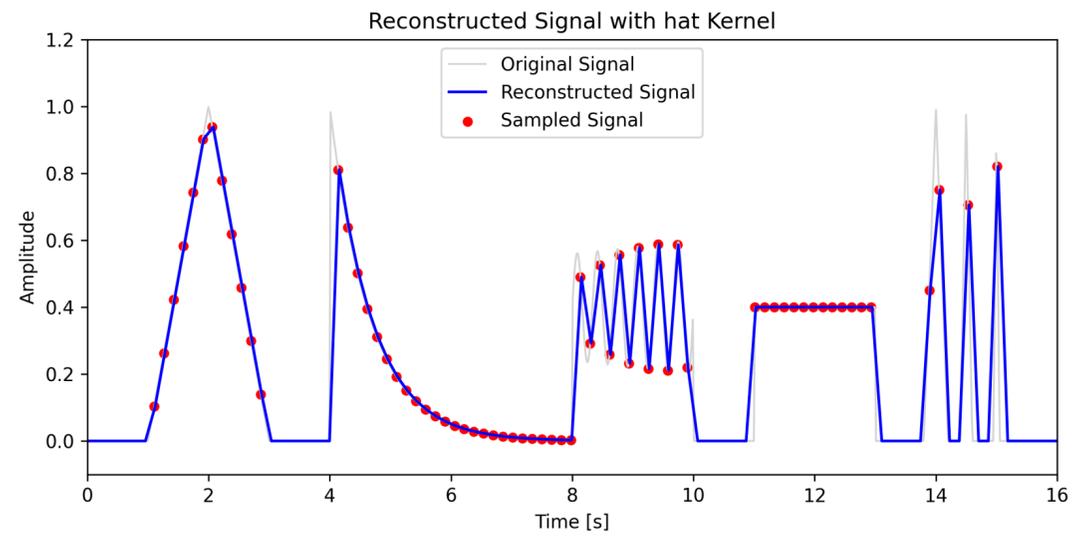
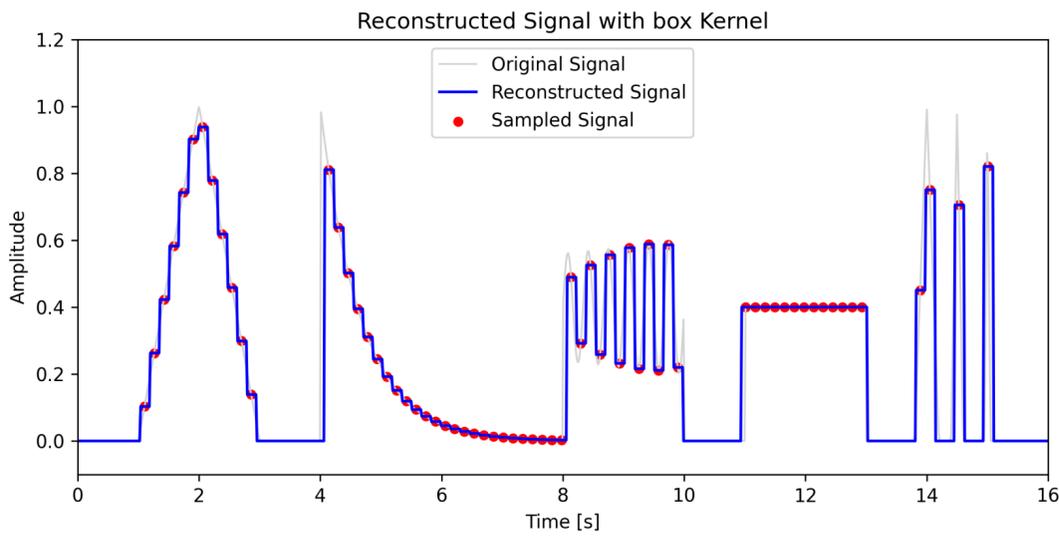
# Reconstrução

1:20



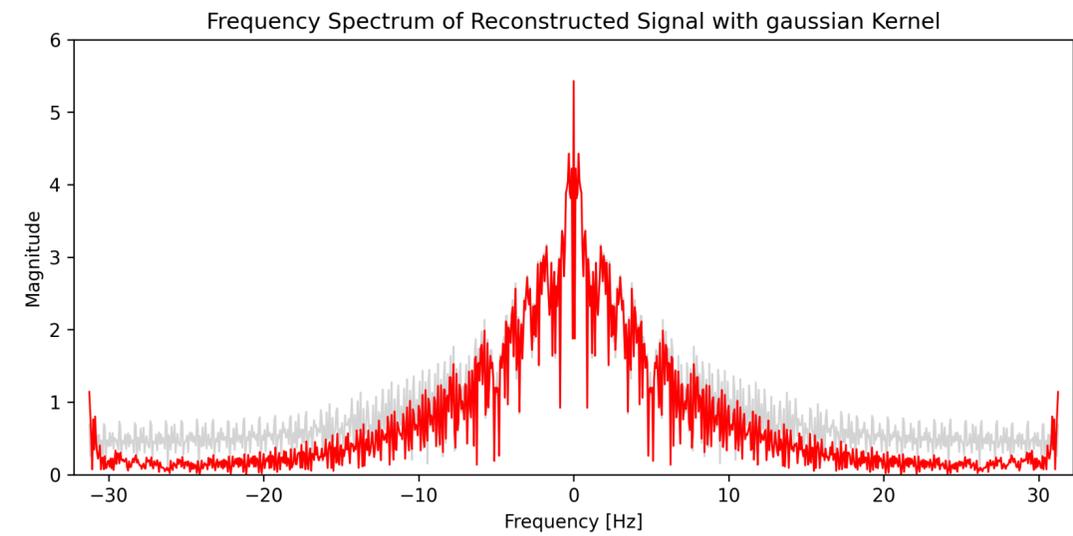
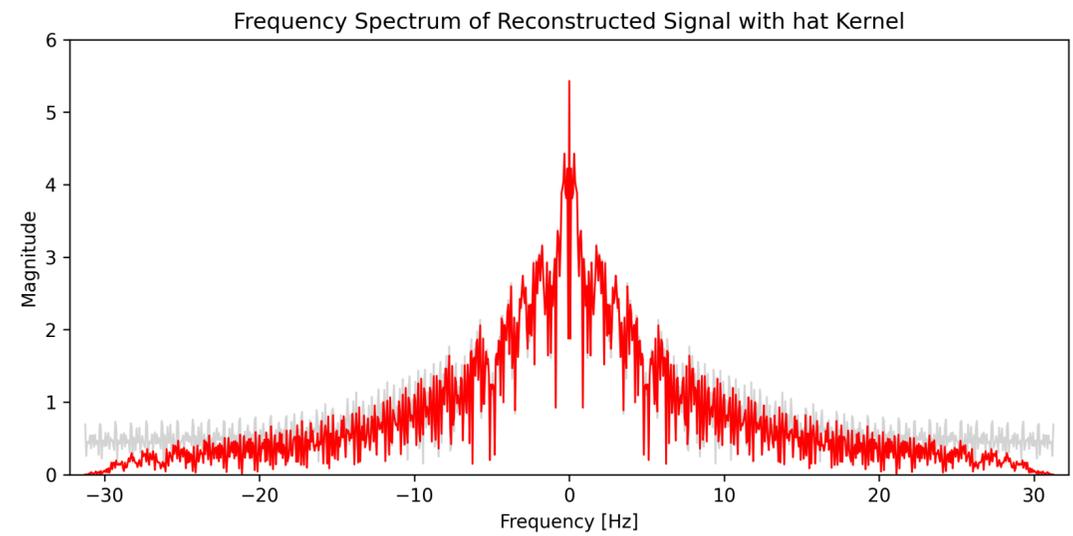
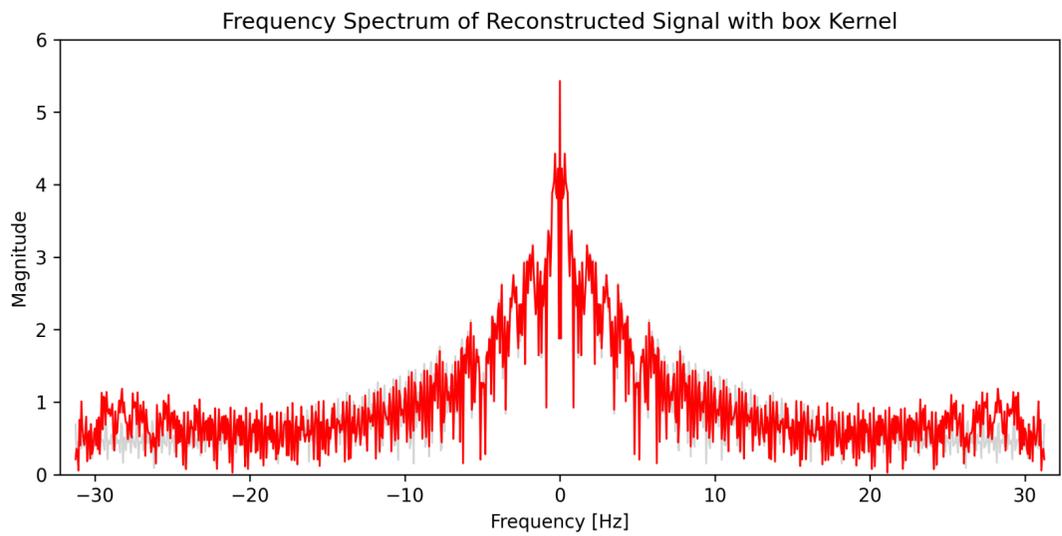
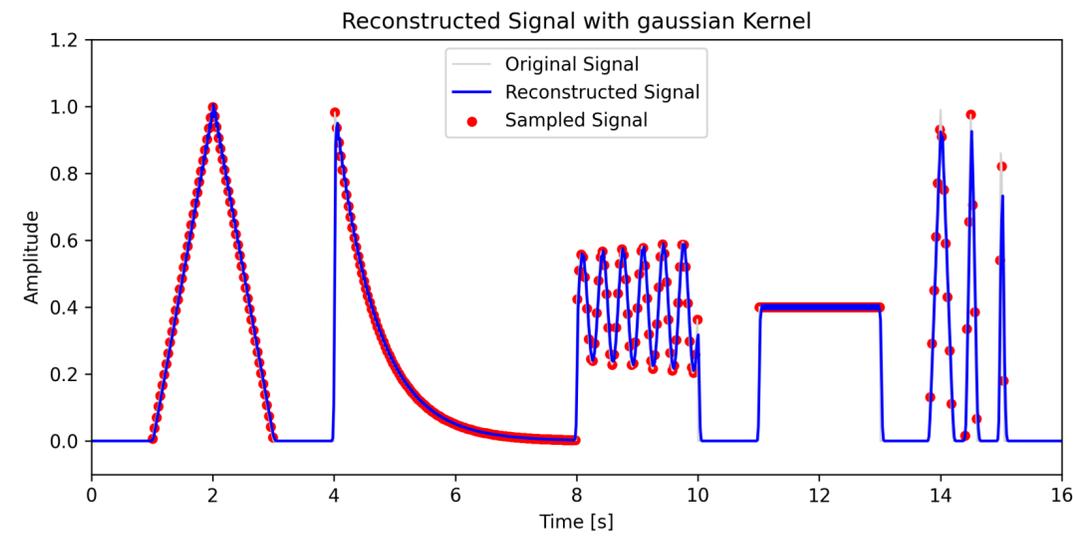
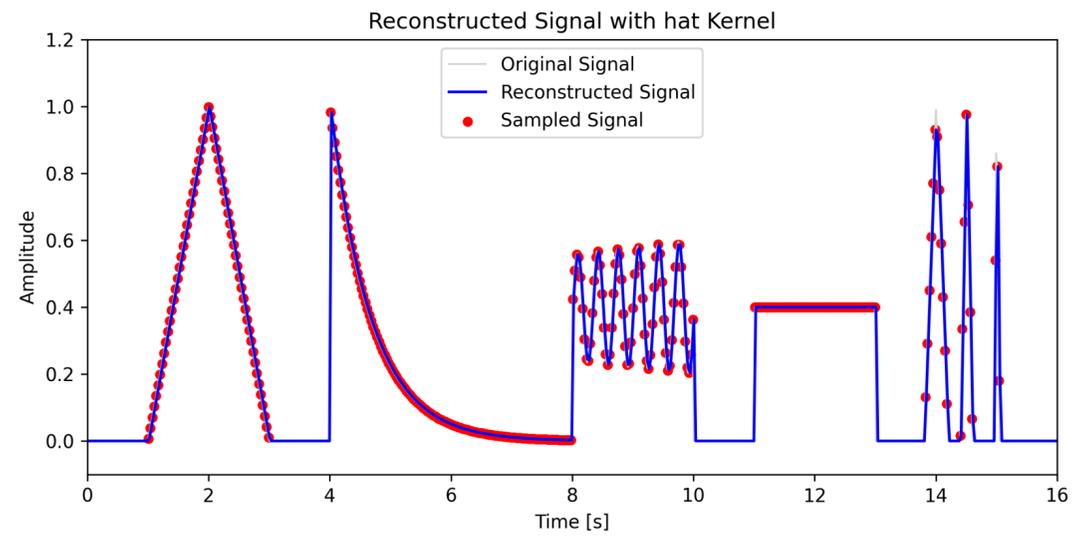
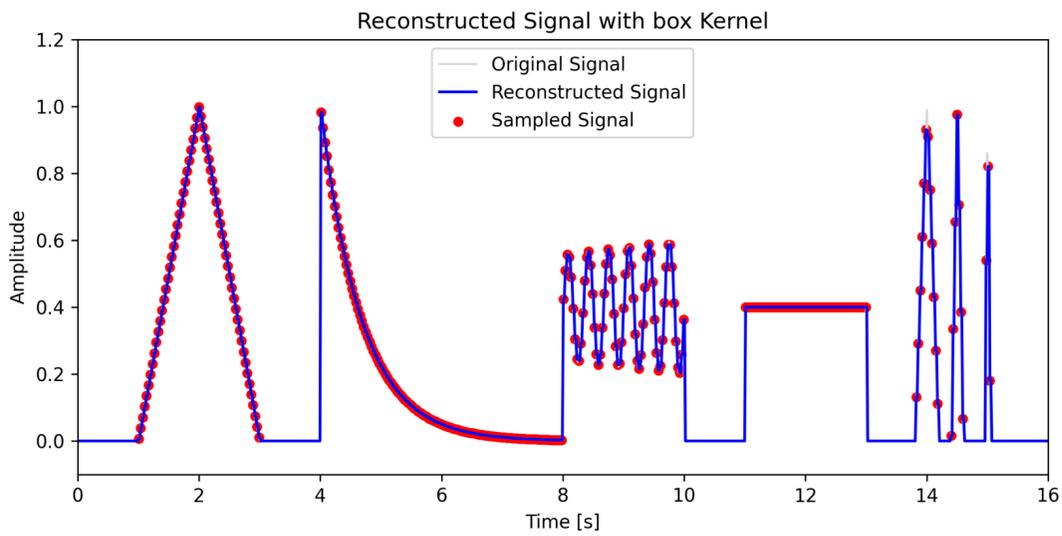
# Reconstrução

1:10

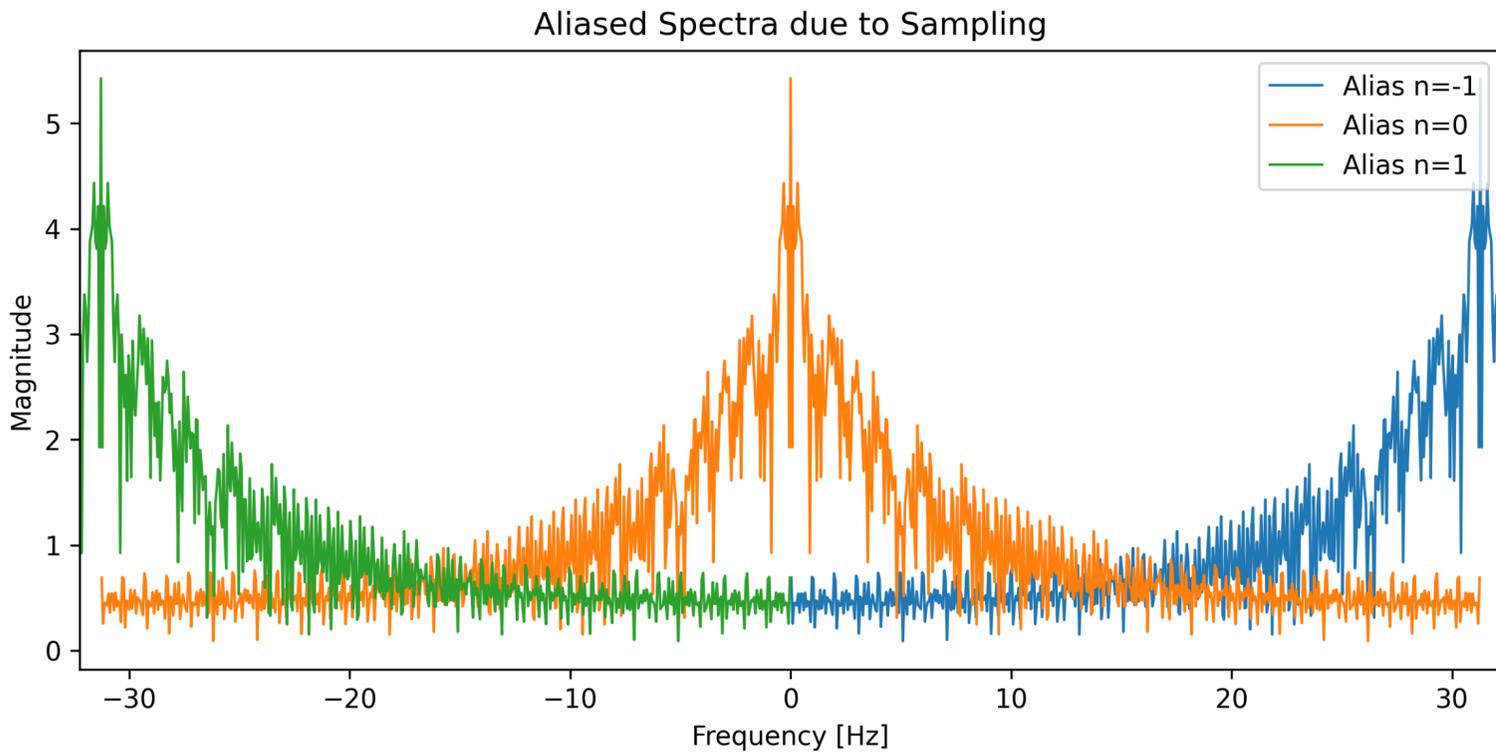


# Reconstrução

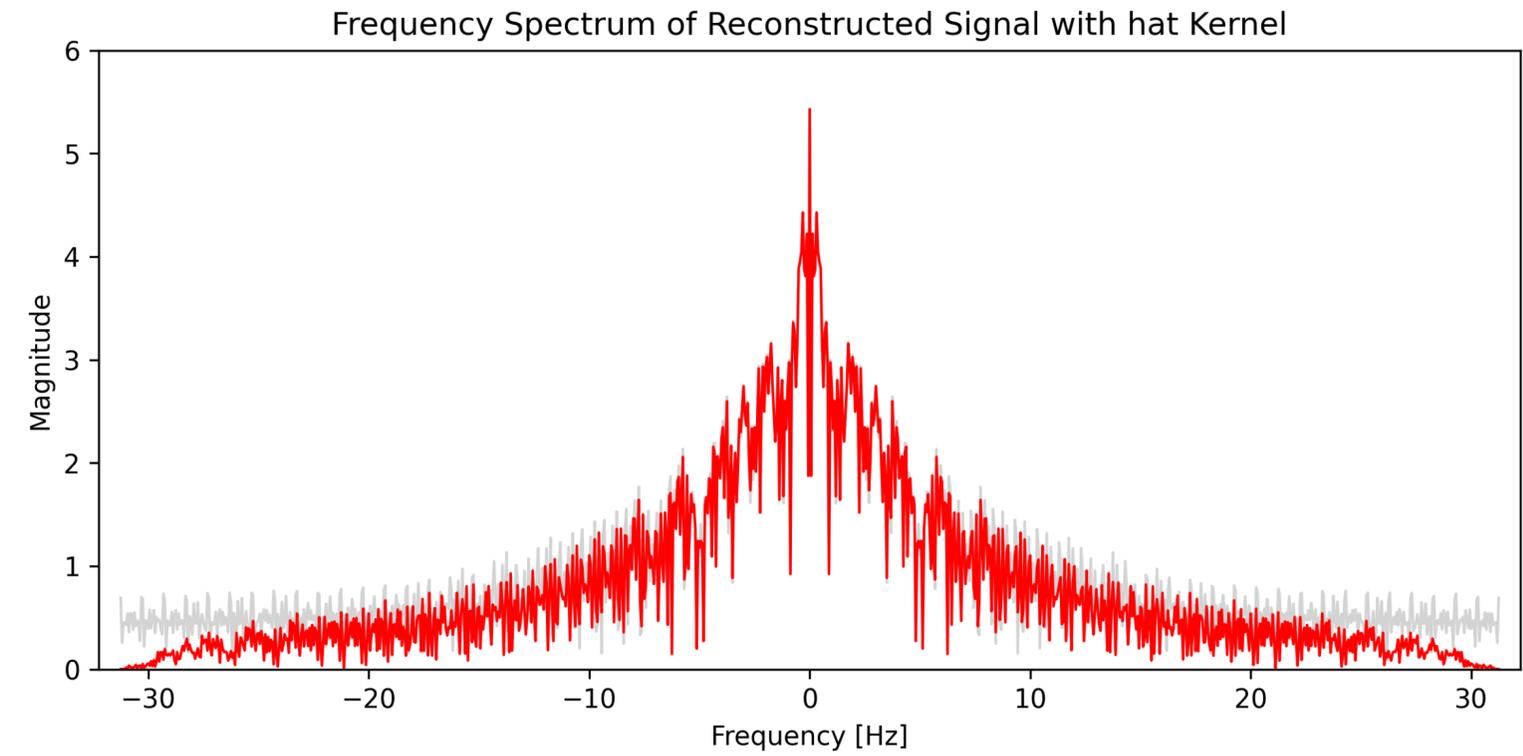
1:2



# Aliasing



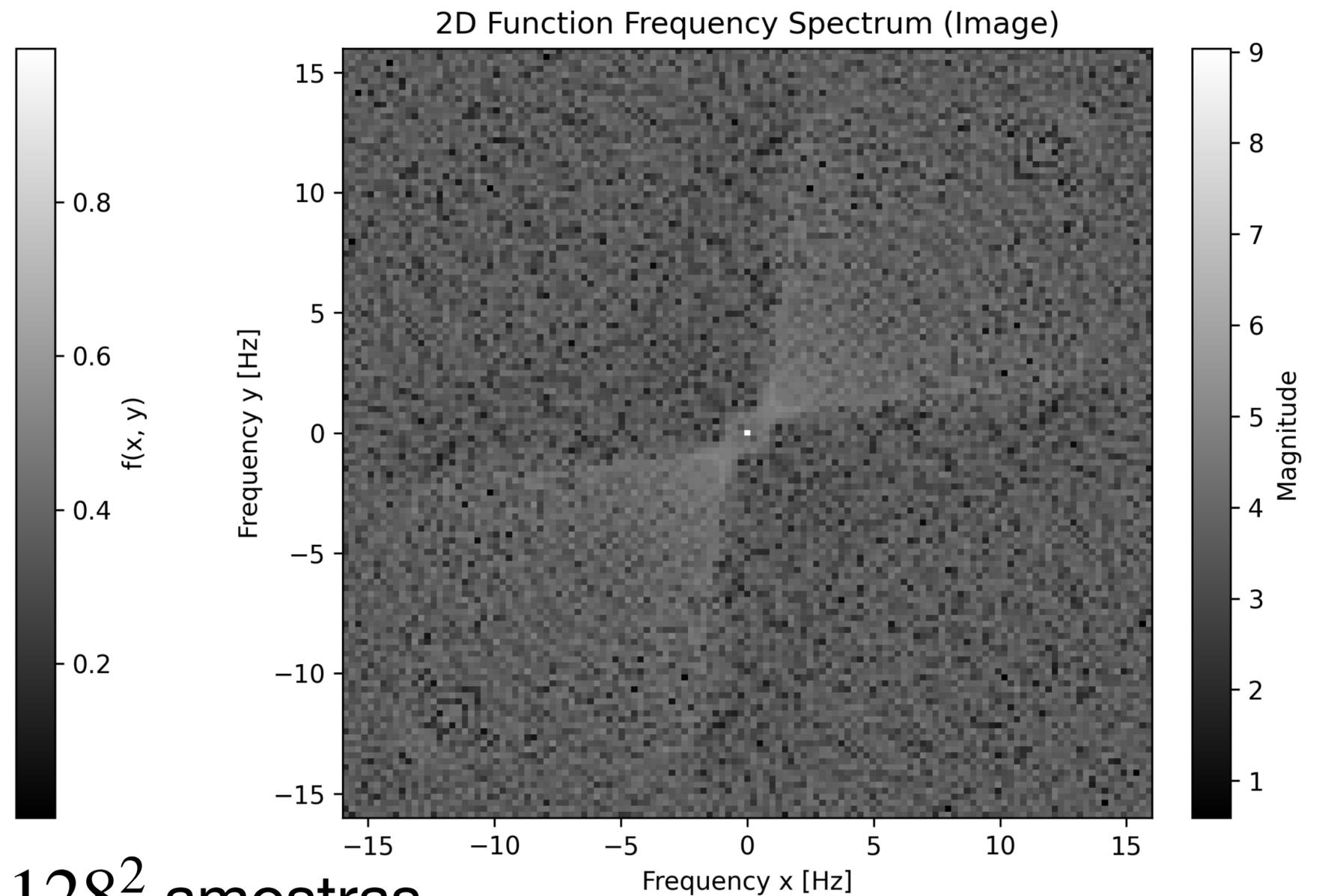
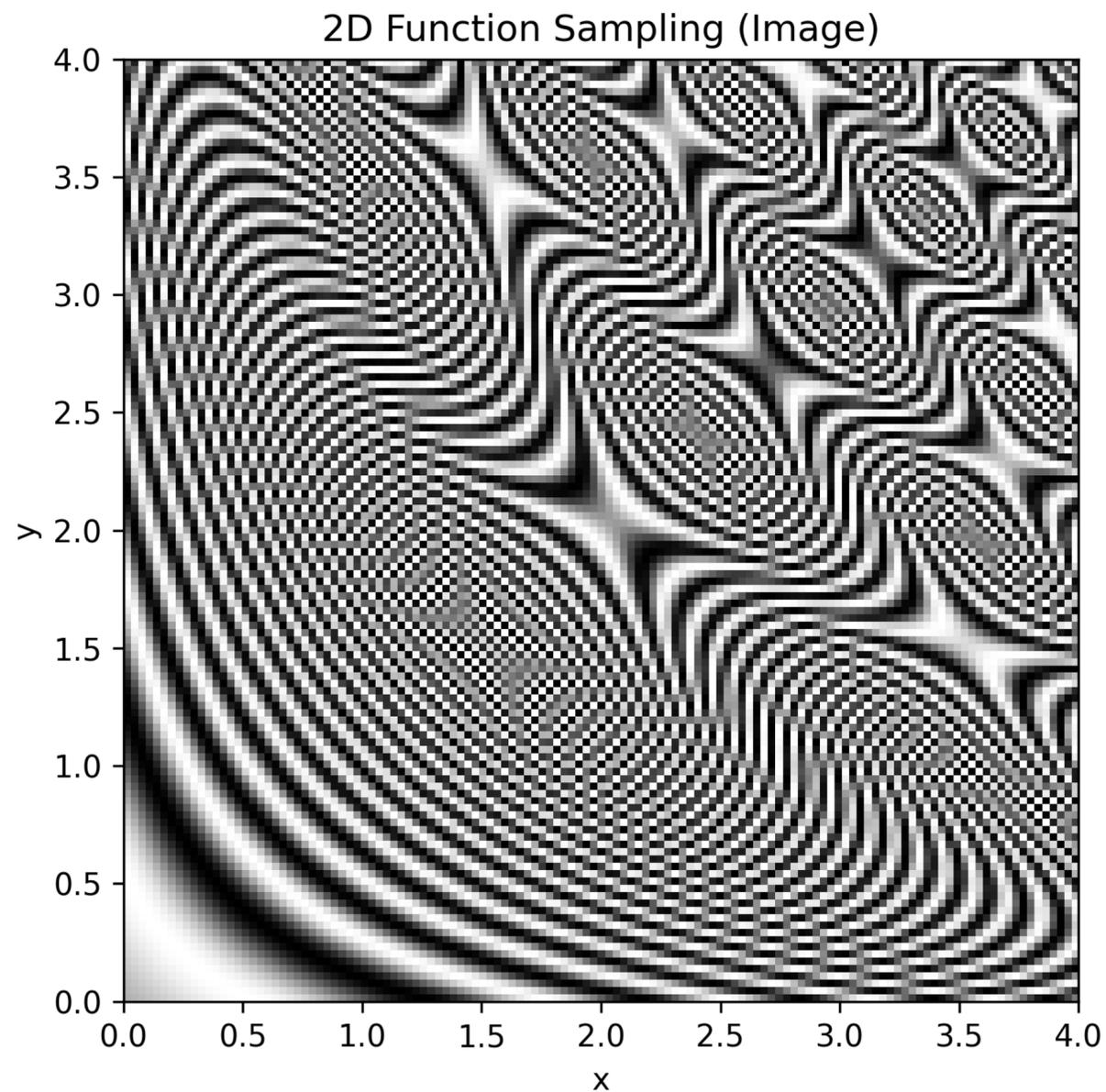
pre-aliasing



post-aliasing

# Exemplo 2D

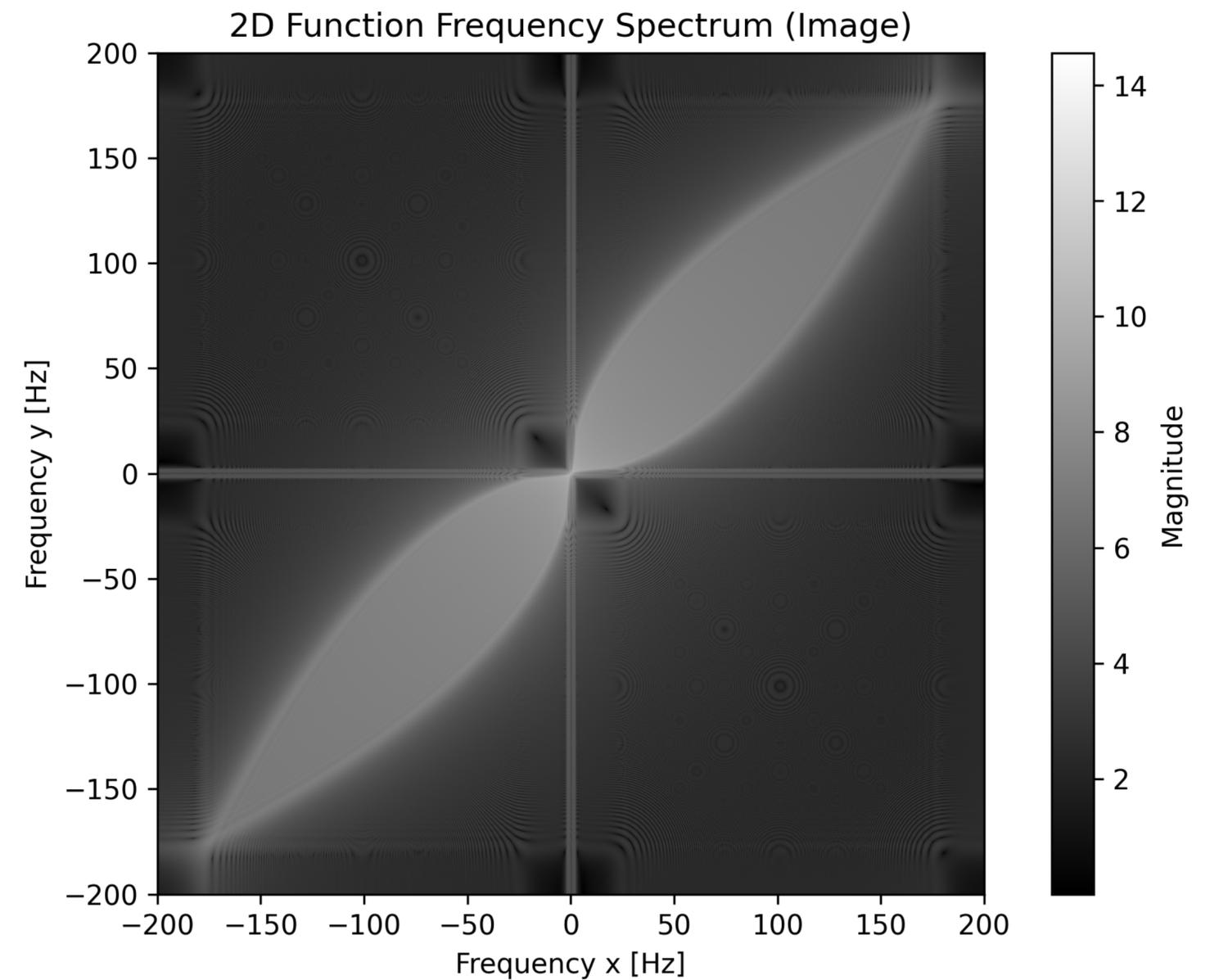
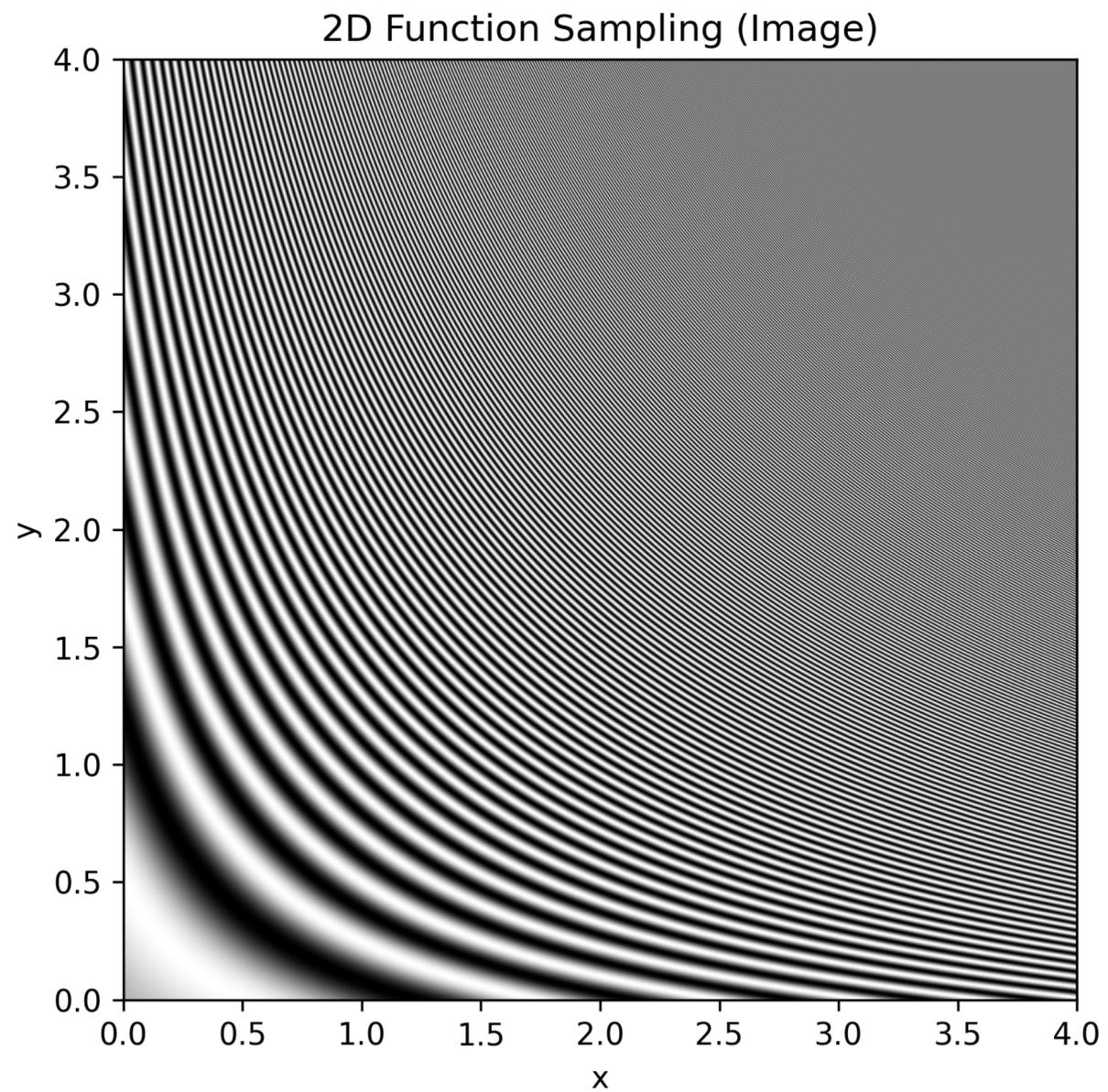
$$f(x, y) = \frac{1}{2} + \frac{\sin(2\pi x^2 y^2)}{2}$$



128<sup>2</sup> amostras

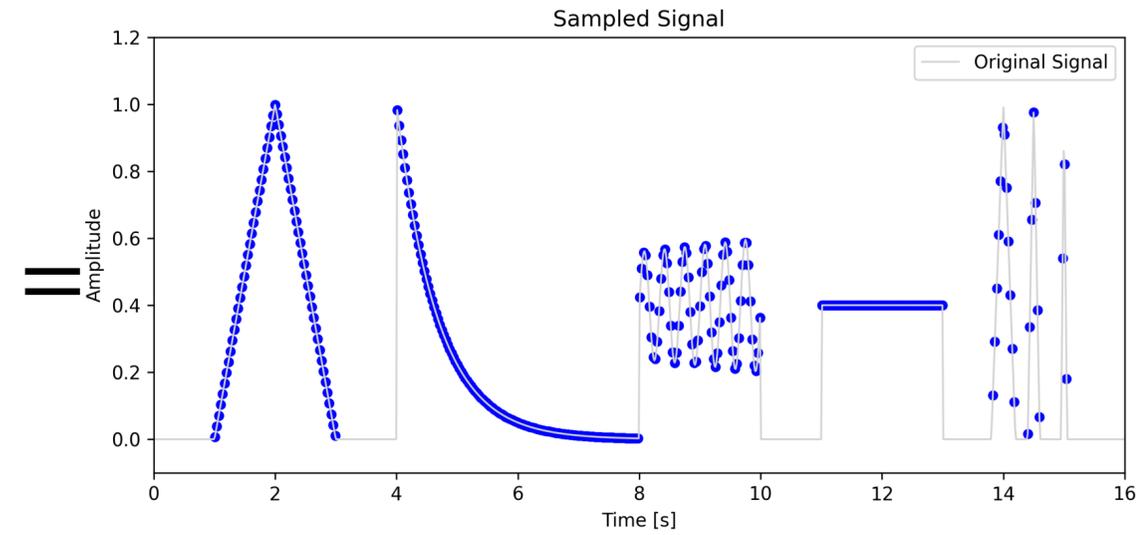
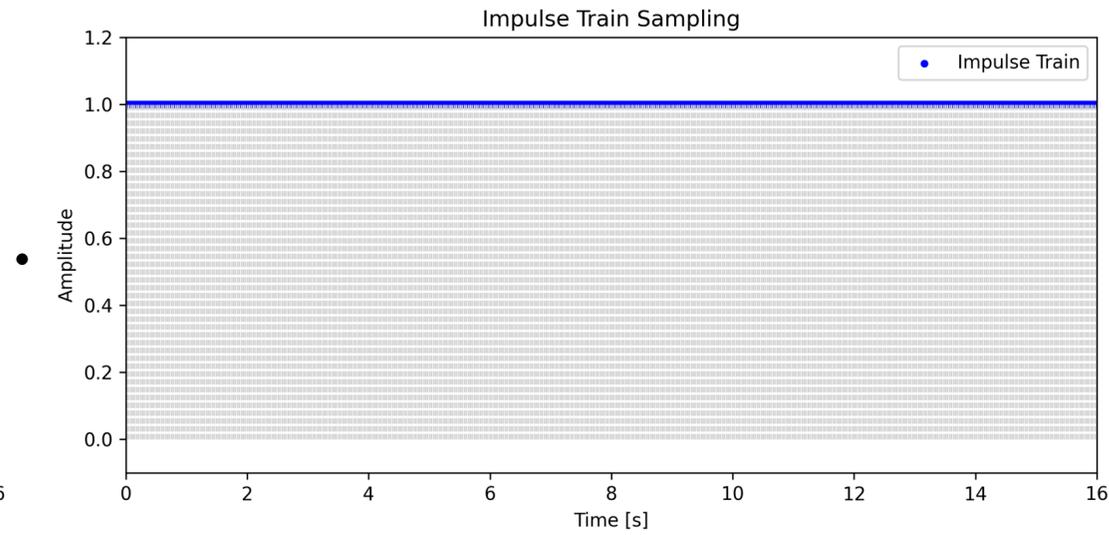
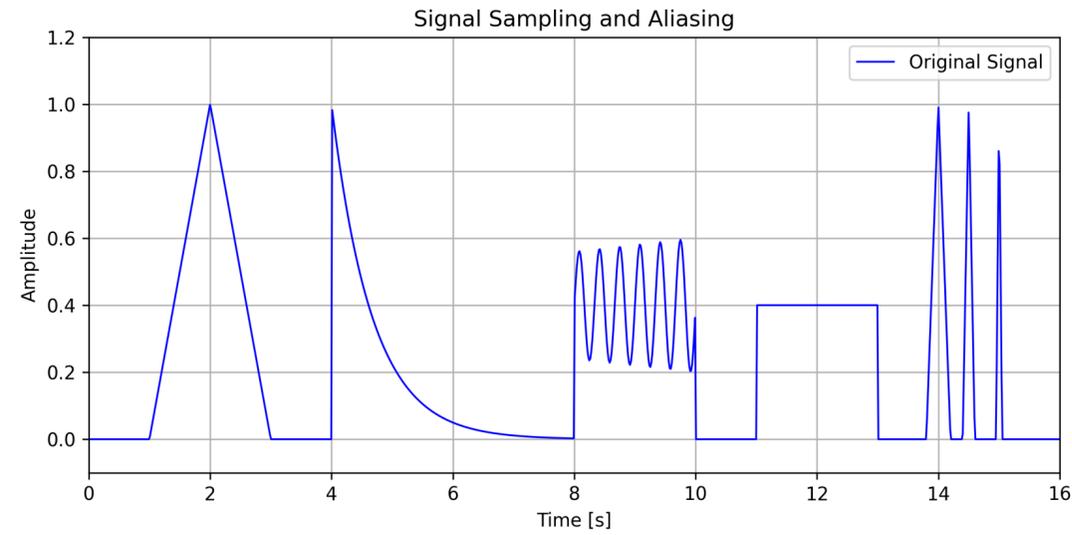
# Exemplo 2D

$$f(x, y) = \frac{1}{2} + \frac{\sin(2\pi x^2 y^2)}{2}$$

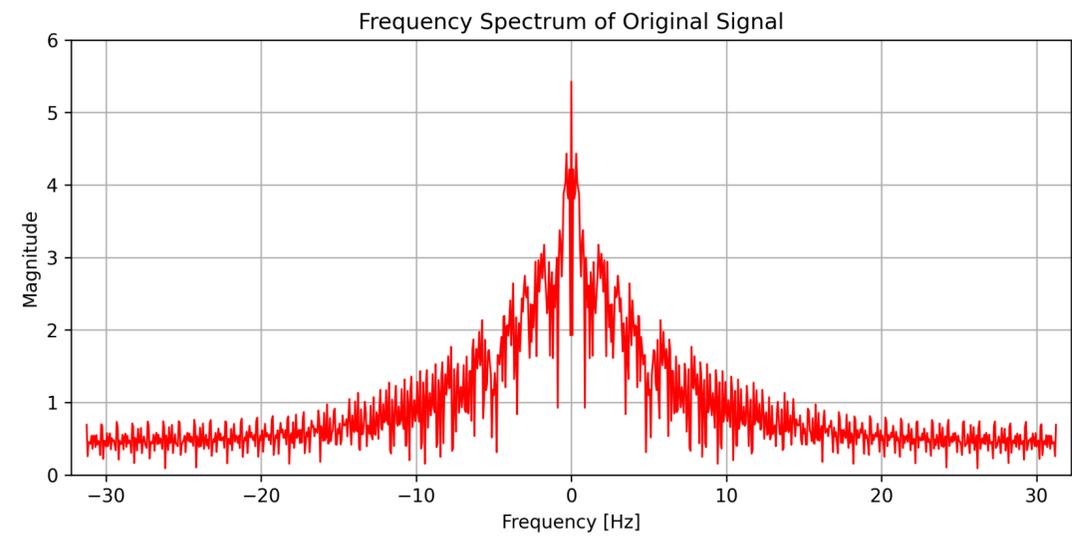


# Anti-aliasing

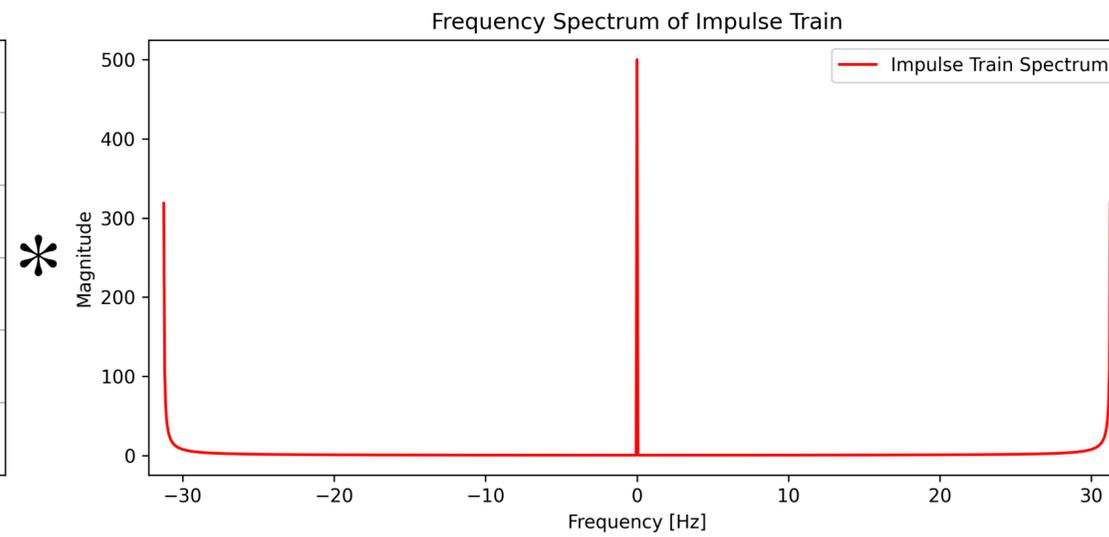
1:2



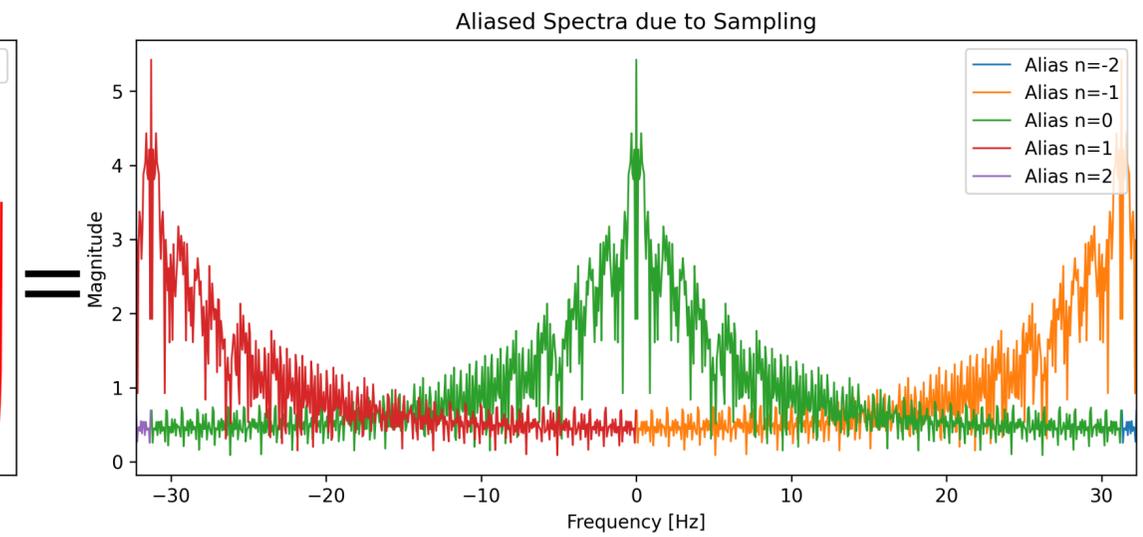
$\mathcal{F}$



$\mathcal{F}$

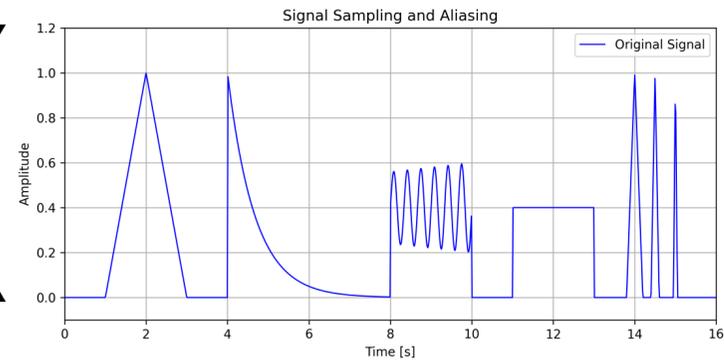


$\mathcal{F}$

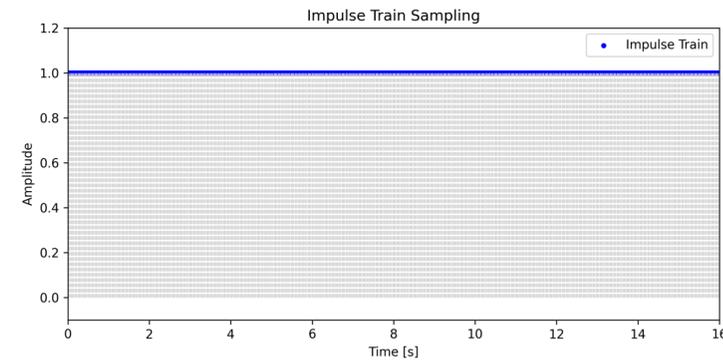


# Anti-aliasing

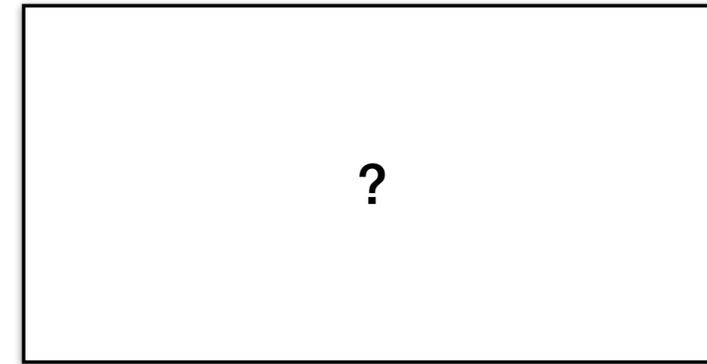
1:2



\*



=

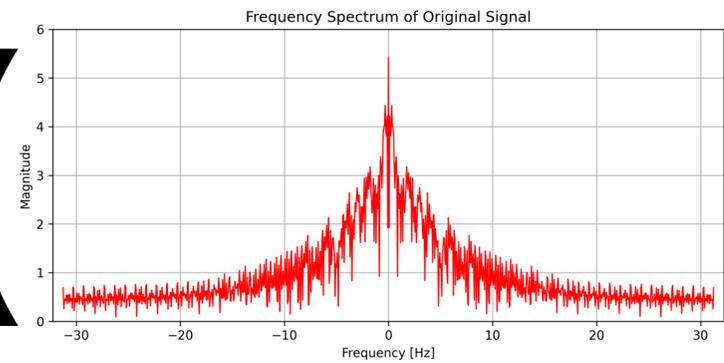


$\mathcal{F}$

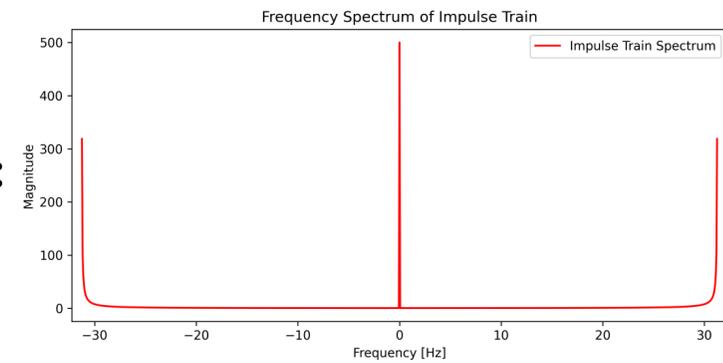
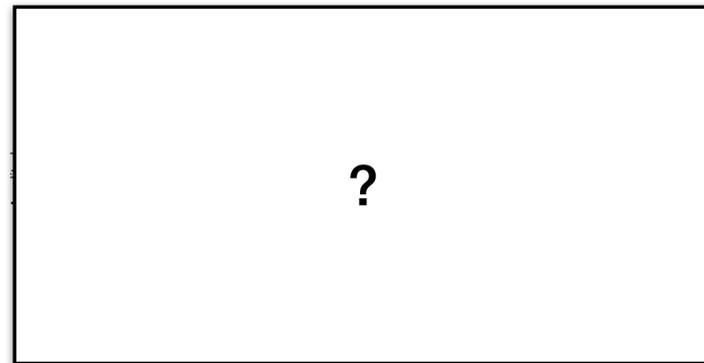
$\mathcal{F}$

$\mathcal{F}$

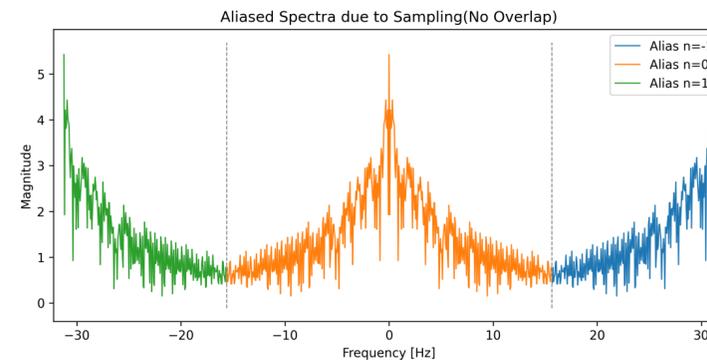
$\mathcal{F}^{-1}$



\*

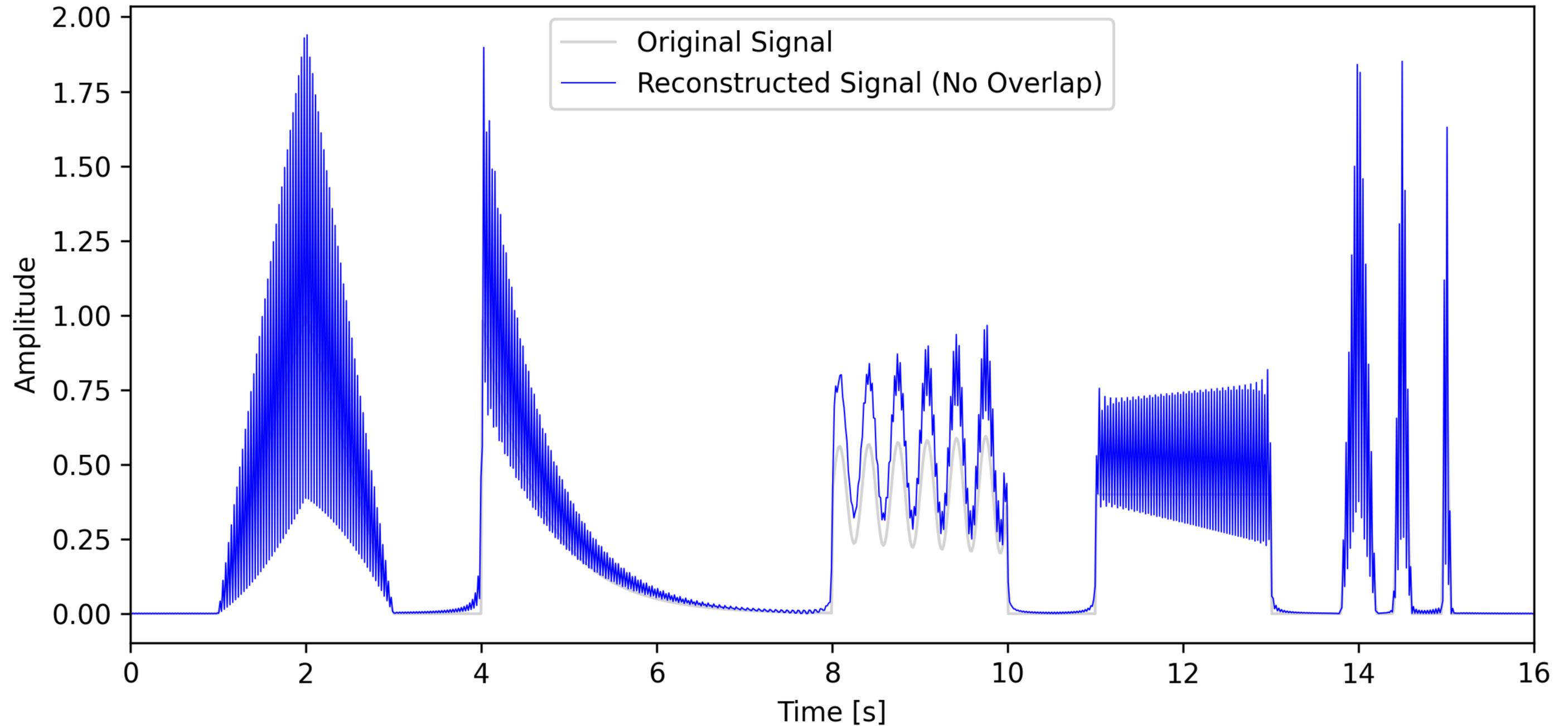


=



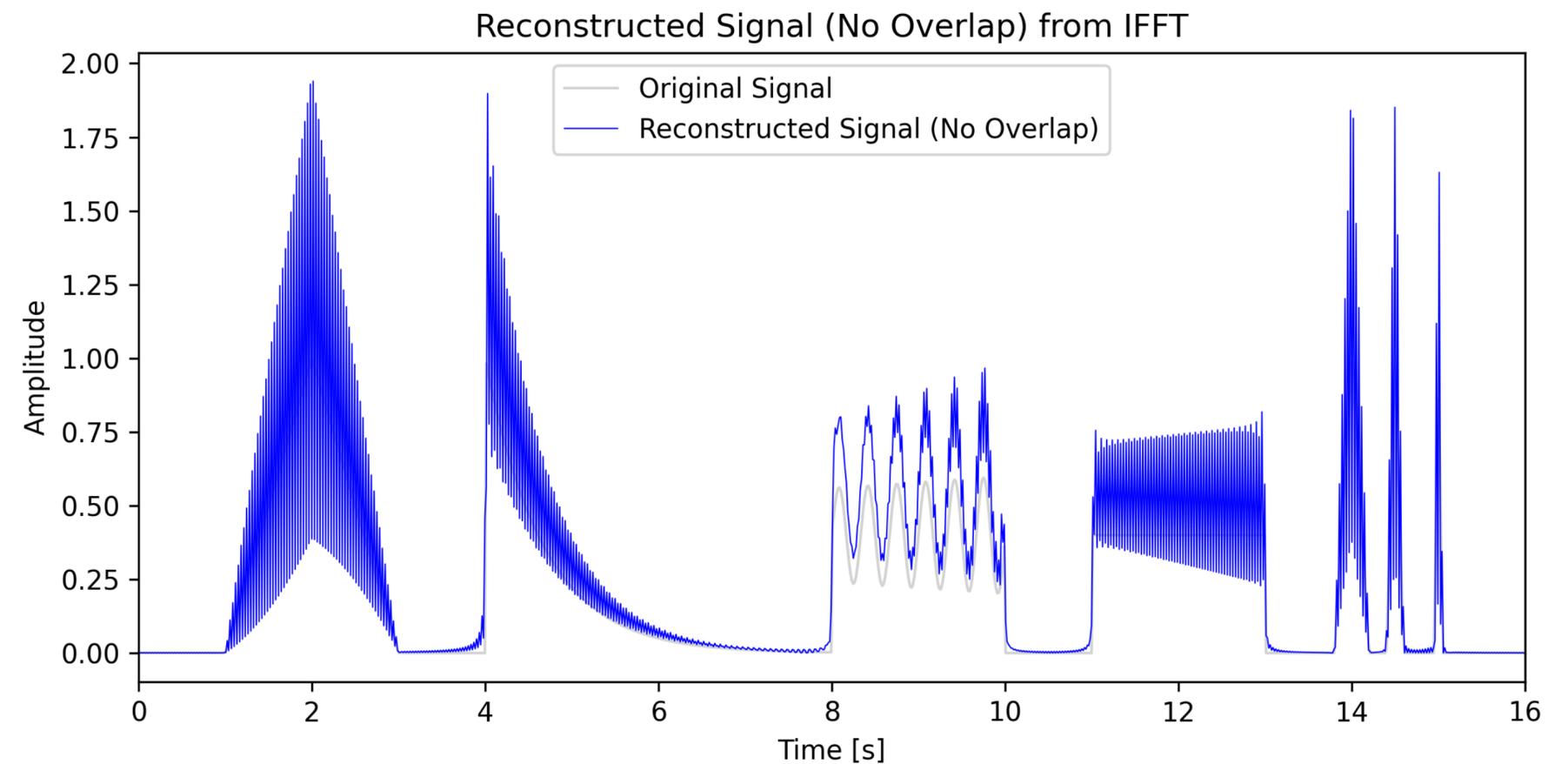
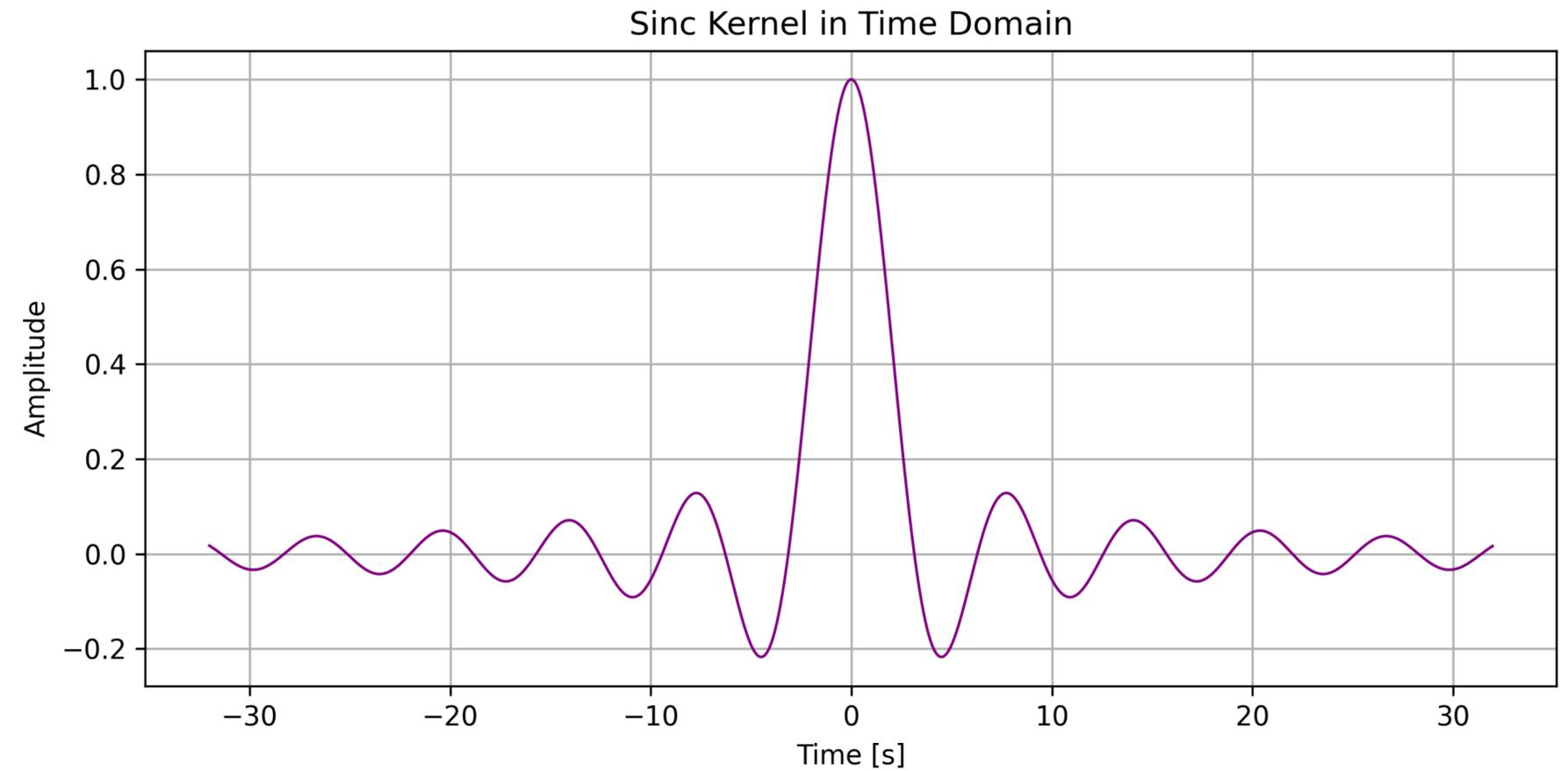
# Anti aliasing

Reconstructed Signal (No Overlap) from IFFT

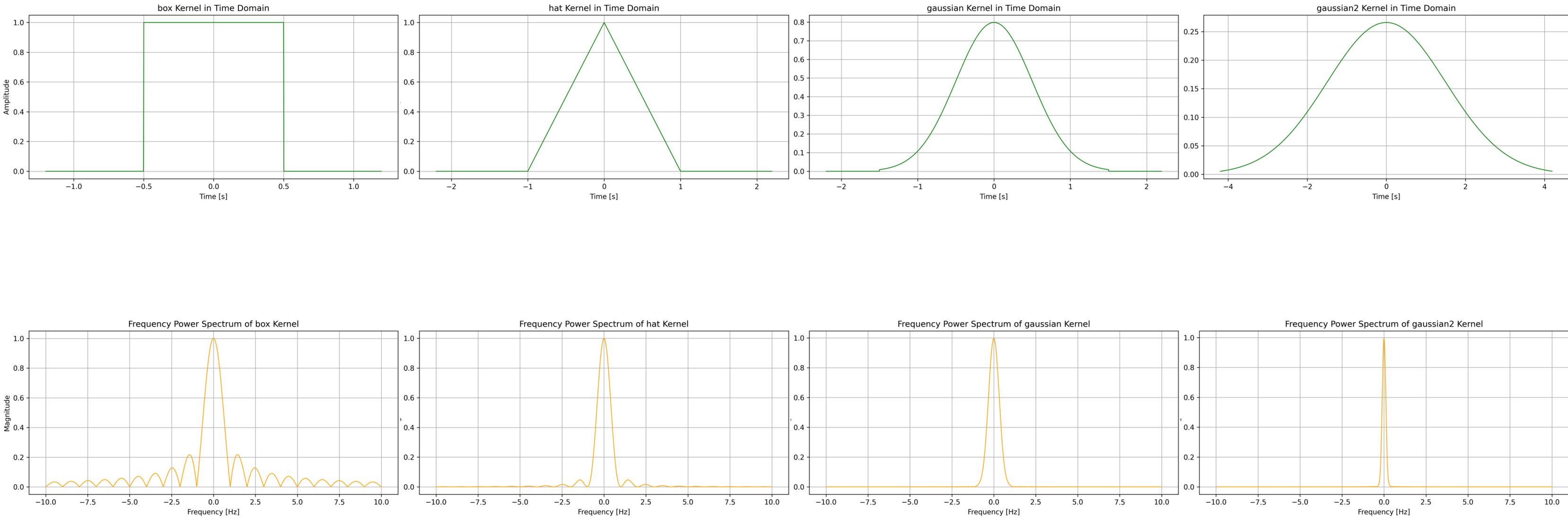


# Filtro `sinc()` é ruim

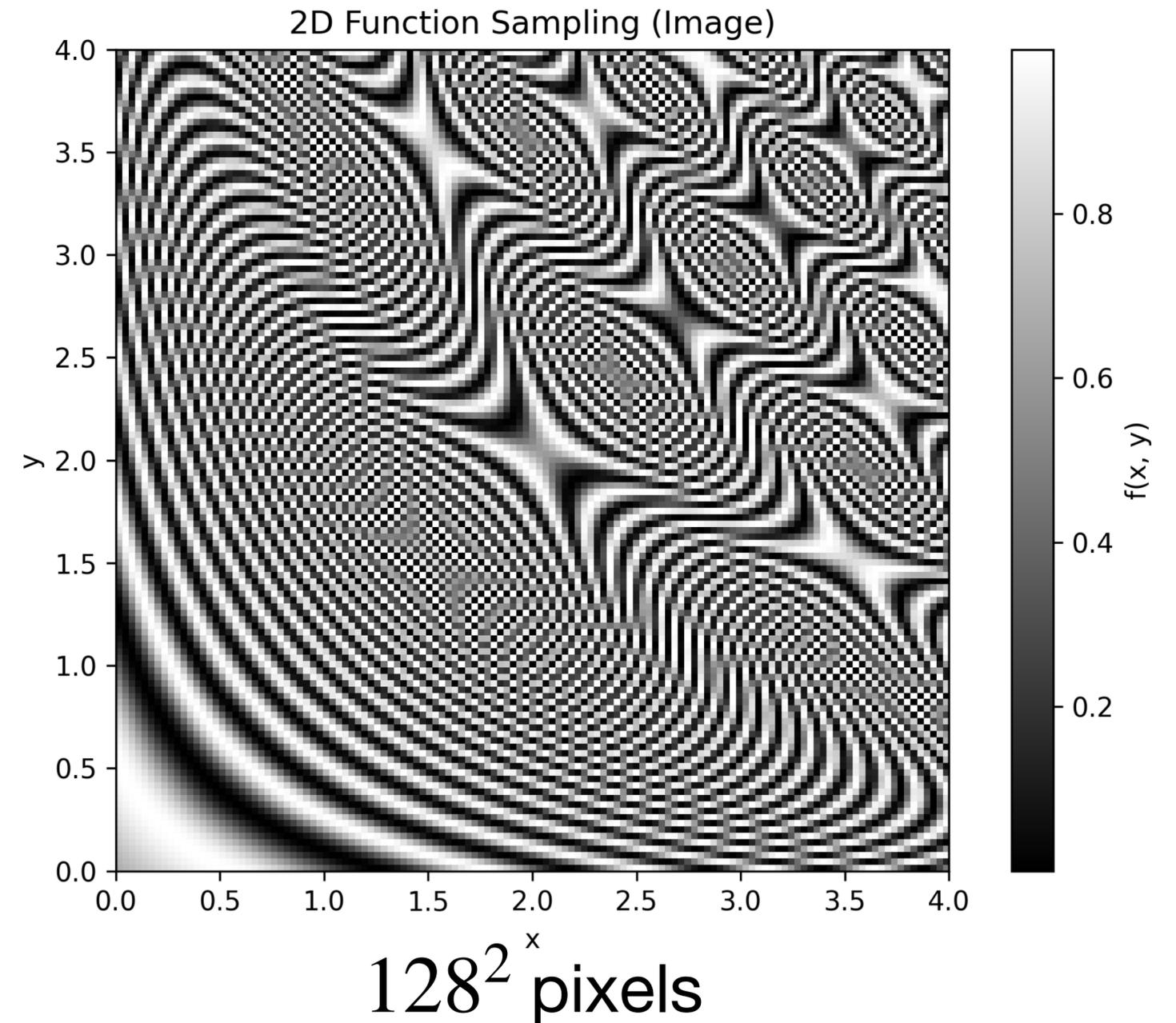
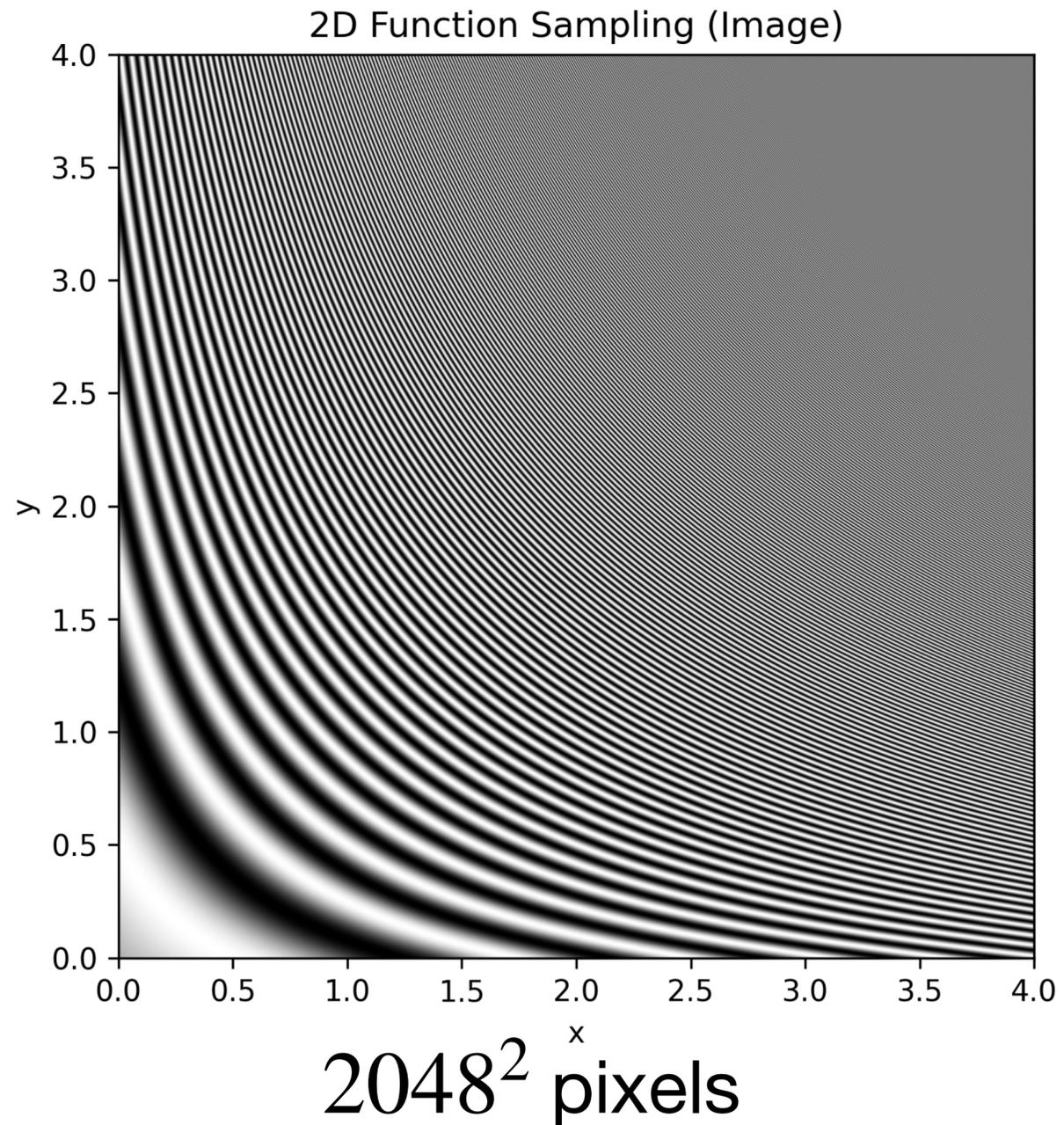
- suporte infinito no espaço
- cria ondulações no sinal



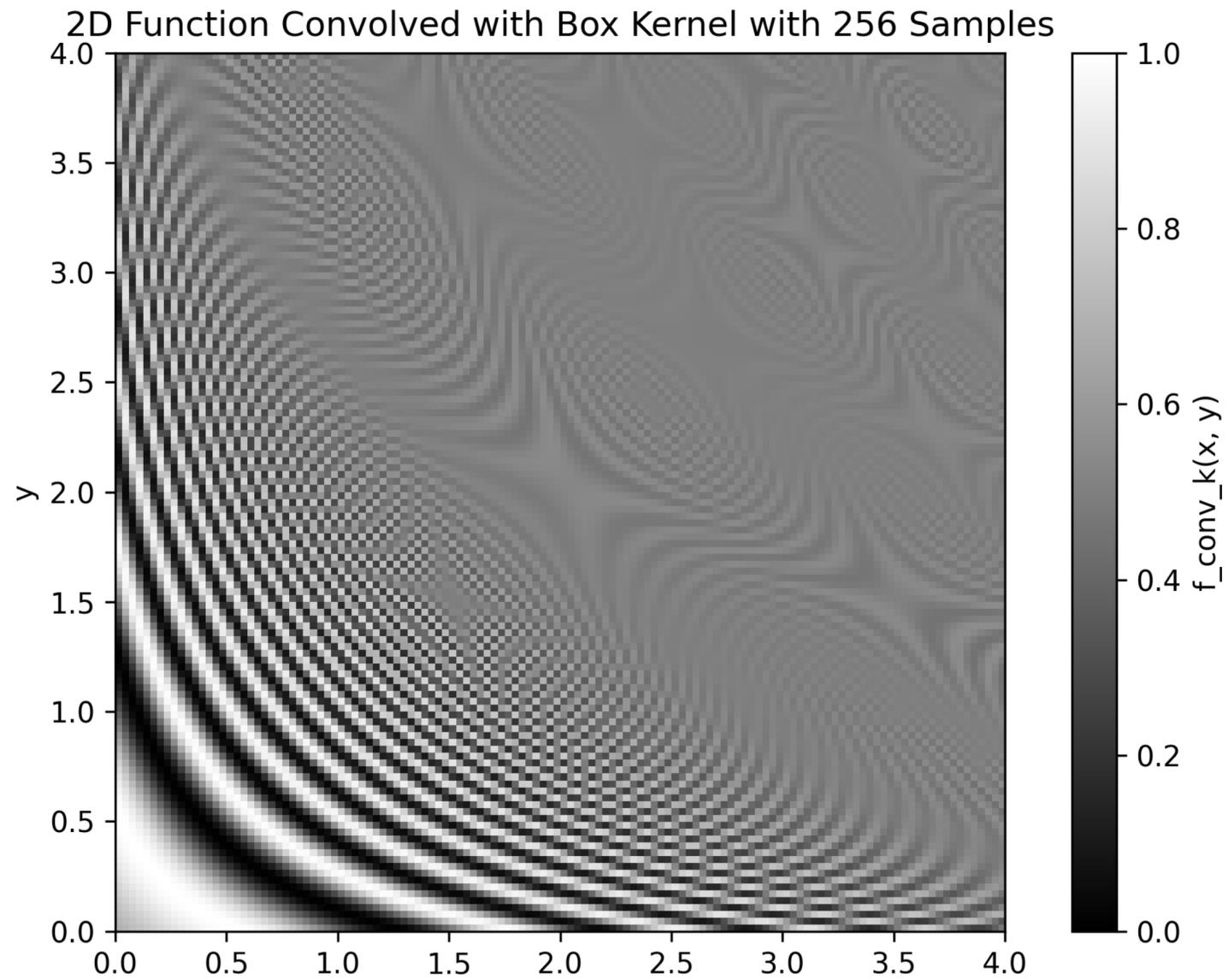
# Kernels de suporte compacto



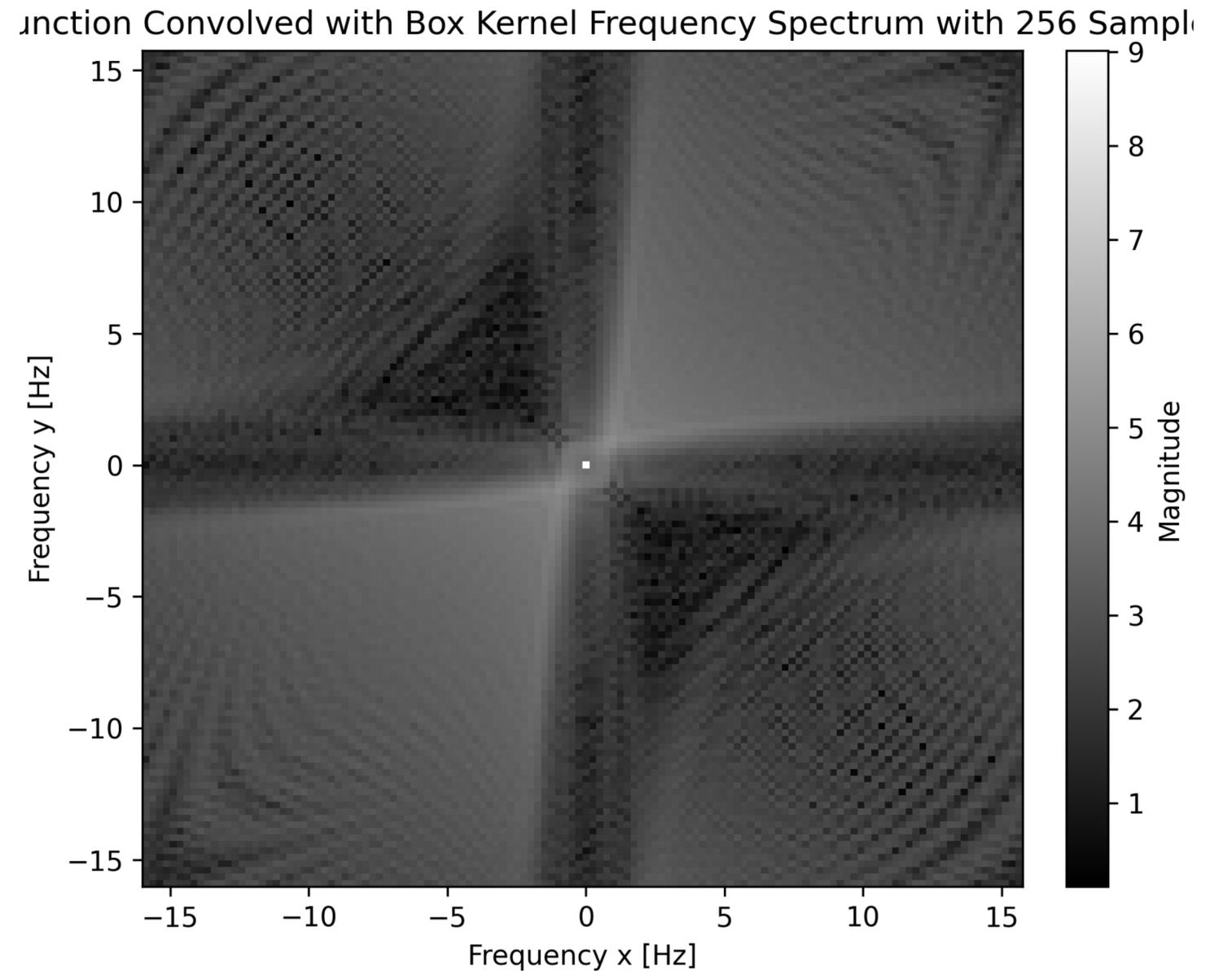
# De volta ao 2D



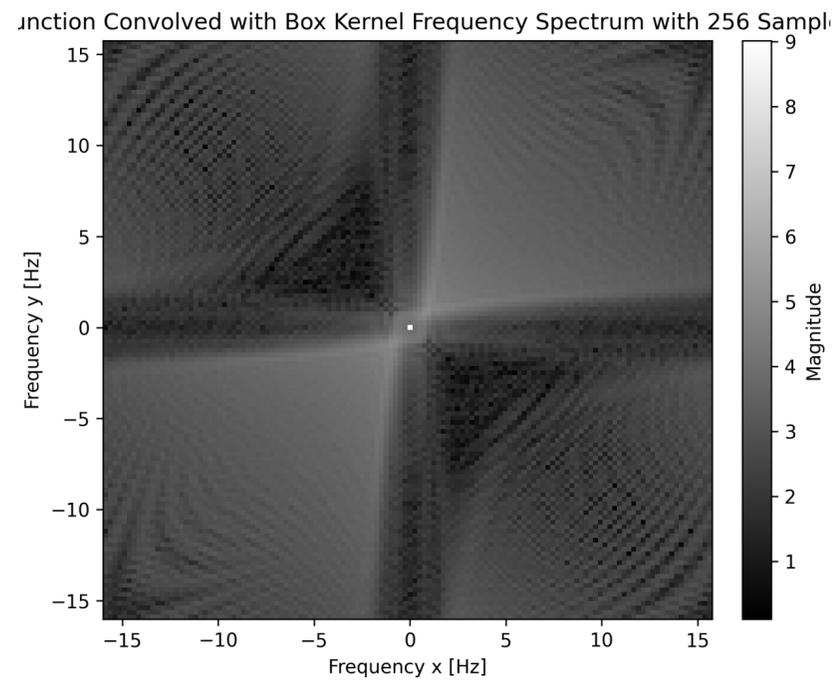
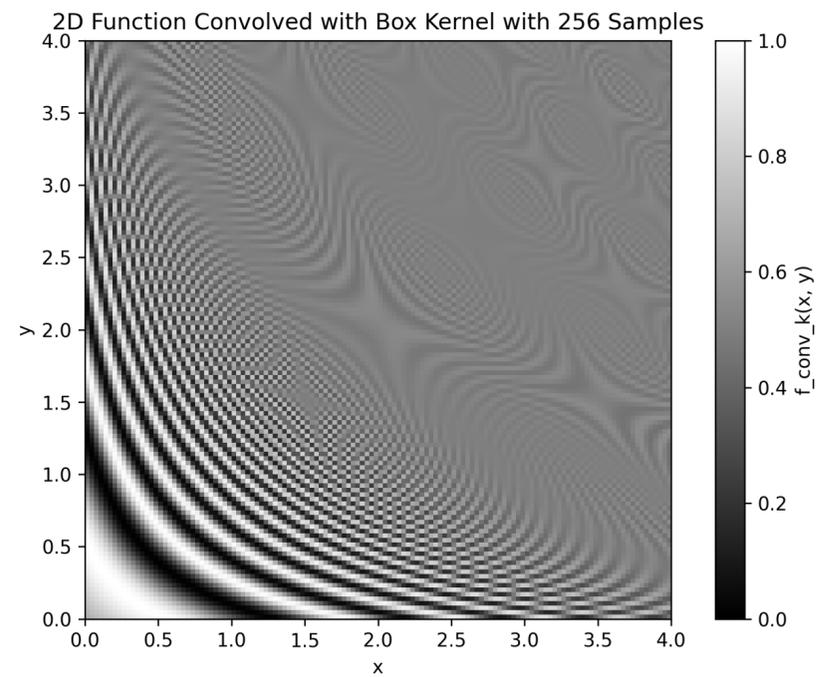
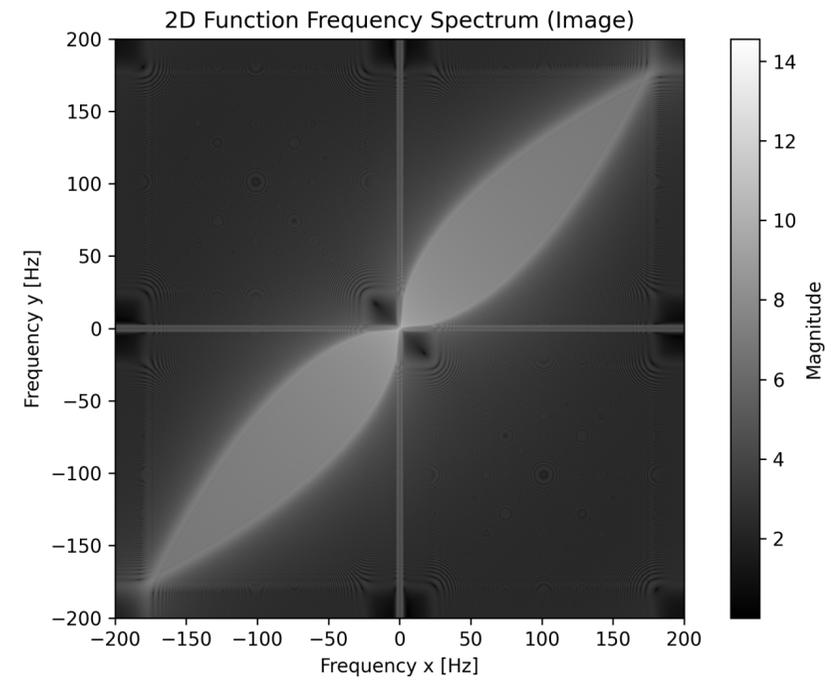
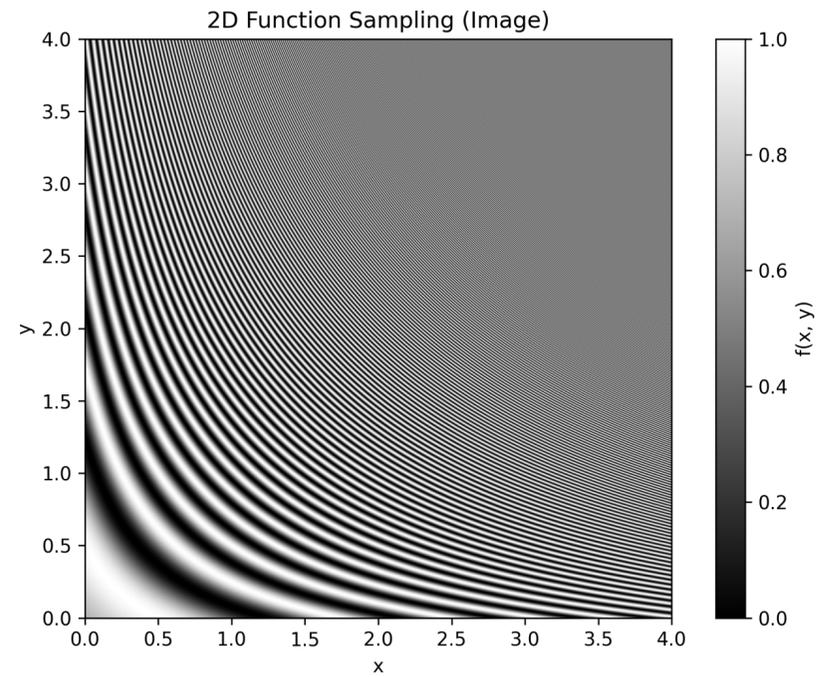
# De volta ao 2D



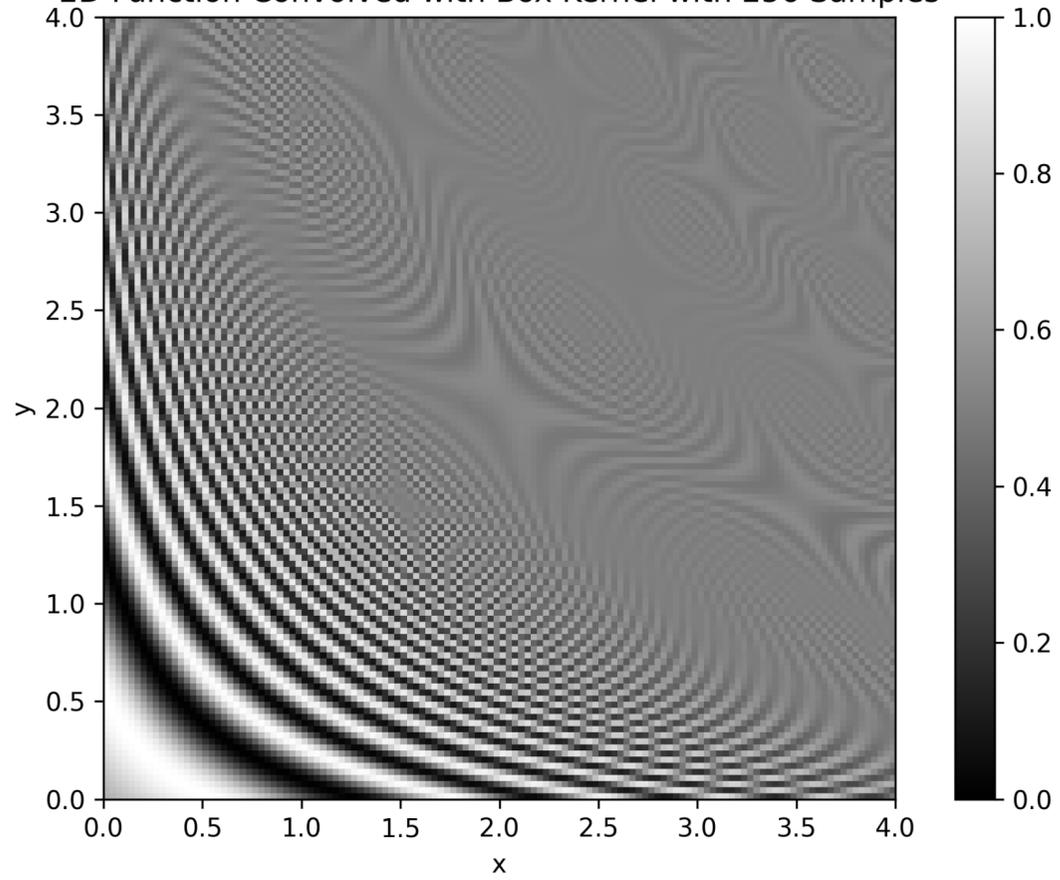
$128^2$  pixels (256 amostras/pixel)



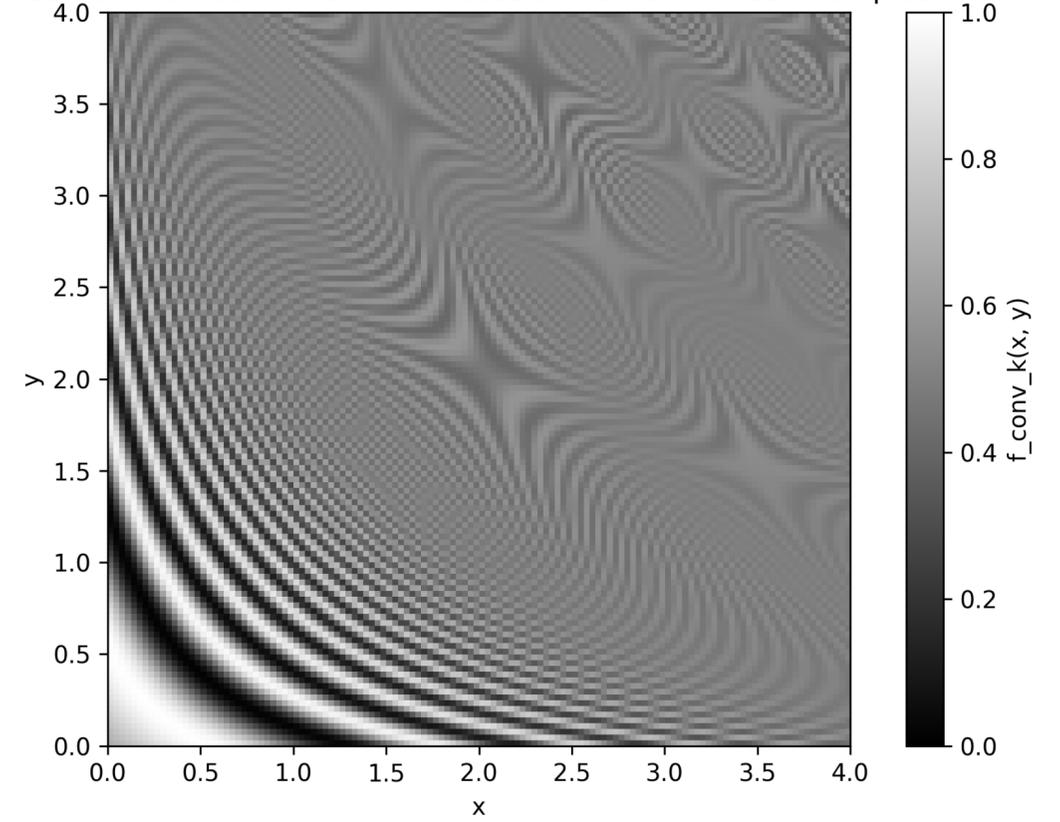
# De volta ao 2D



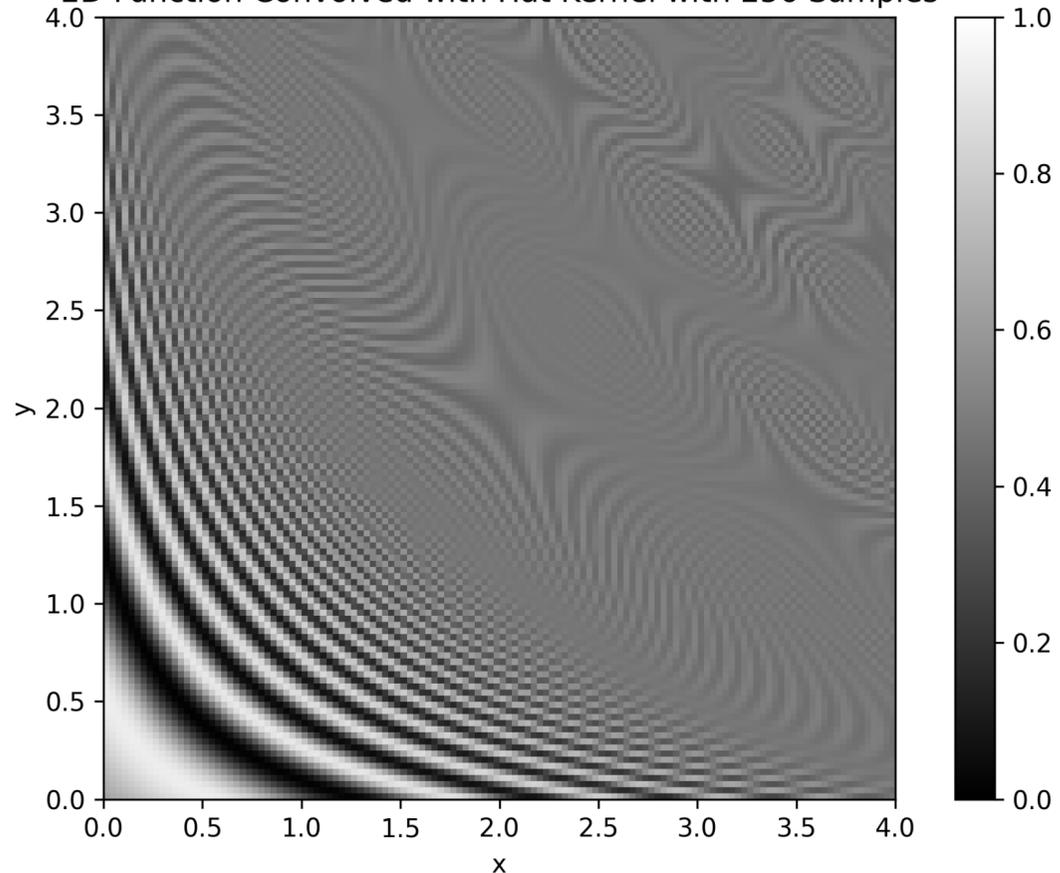
2D Function Convolved with Box Kernel with 256 Samples



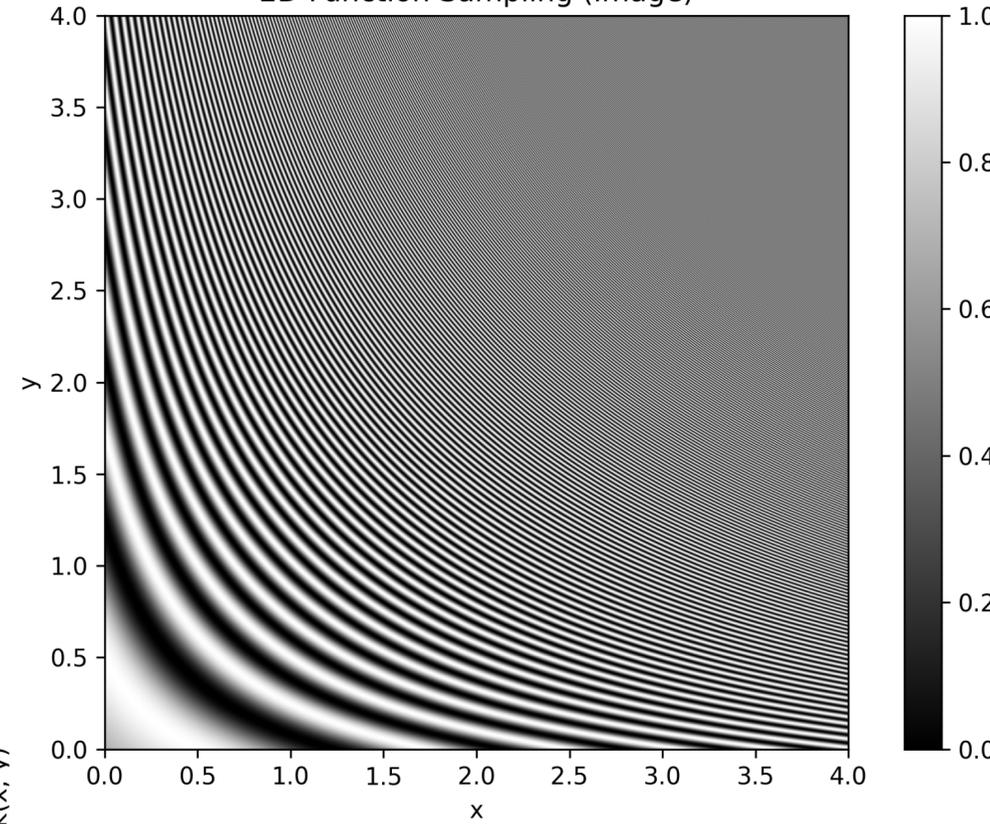
2D Function Convolved with Gaussian Kernel with 256 Samples



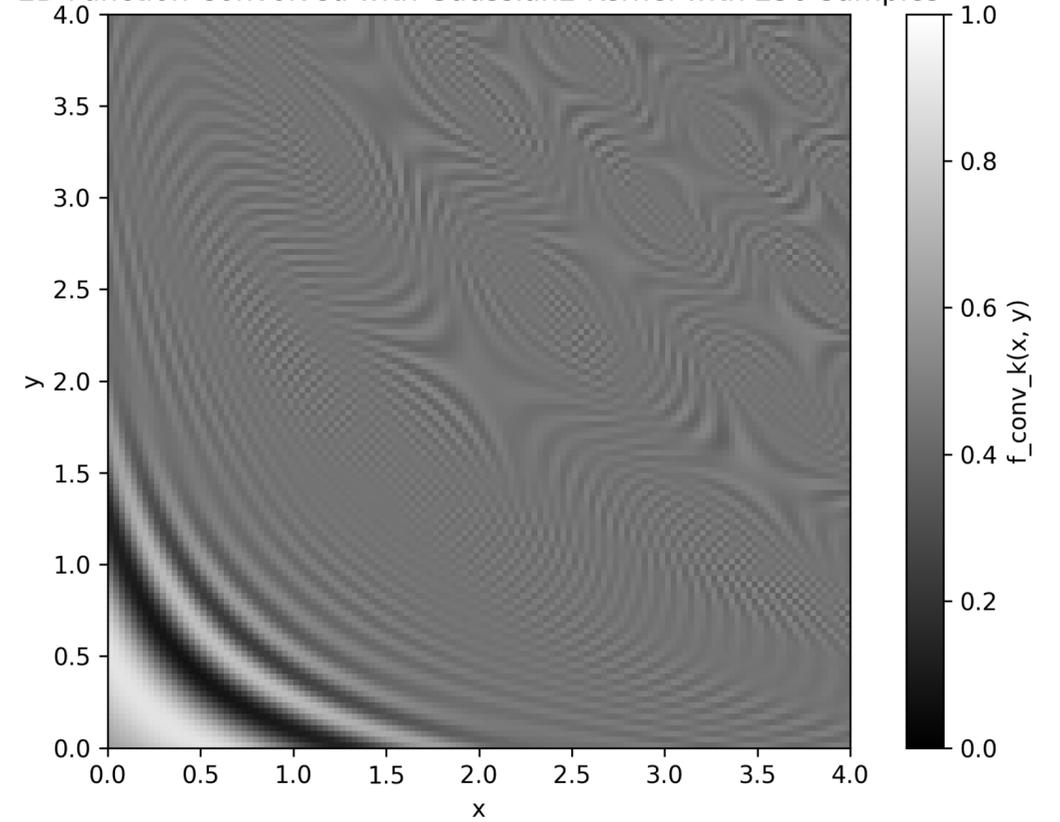
2D Function Convolved with Hat Kernel with 256 Samples



2D Function Sampling (Image)



2D Function Convolved with Gaussian2 Kernel with 256 Samples



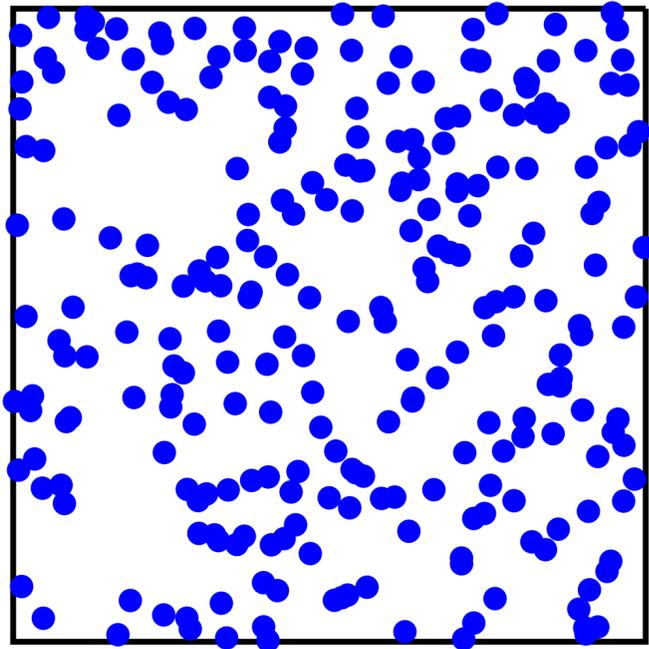
# Como calcular a convolução?

Integração Monte Carlo

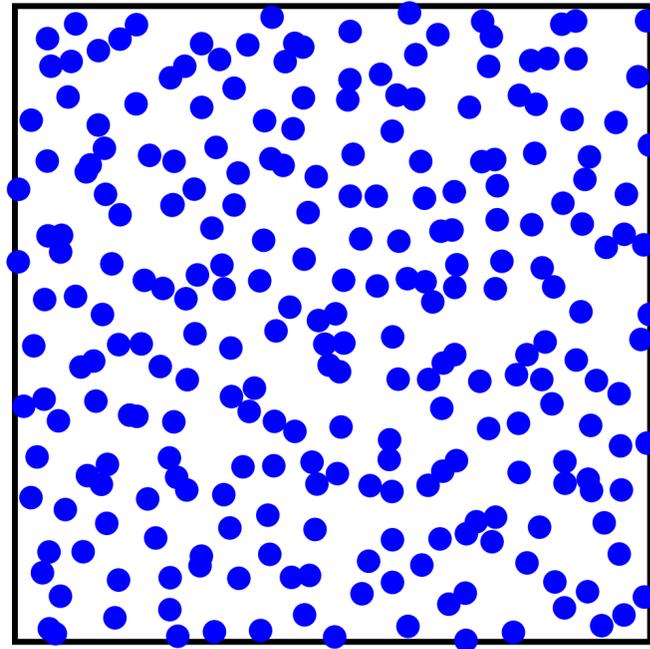
$$\int_a^b f(t) dt \approx E_N := \frac{b-a}{N} \sum_{i=1}^N f(t_i) \text{ onde } t_i \sim U([a, b])$$

Cuja variância  $Var[E_N]$  é proporcional a  $\frac{1}{\sqrt{N}}$

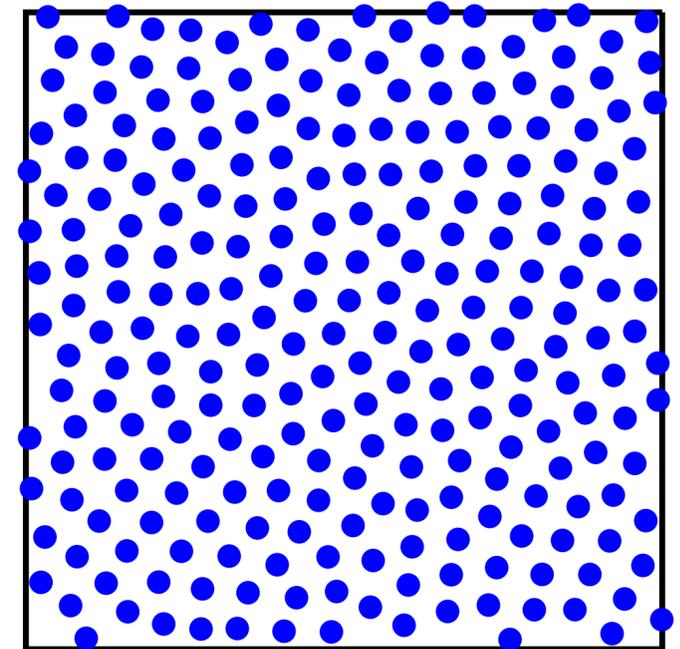
# Amostragens uniformes em 2D



uniform



stratified



blue noise

# Algoritmo de amostragem de images

- definir imagem (coordenadas de pixel)
- definir janela (coordenadas do mundo)
- encontrar transformação da imagem para a janela
- encontrar posição dos pixels
- escolher o kernel (filtro) de amostragem
- amostrar kernel e imagem usando monte carlo

# Amostragem de imagens

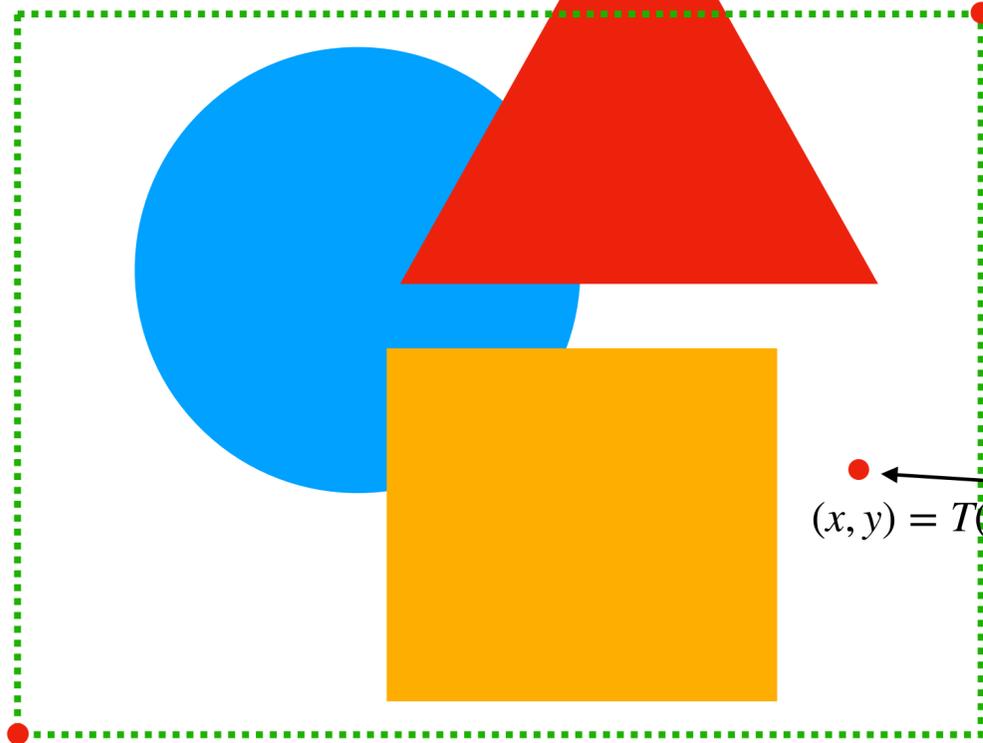
$$(x, y) = T(u, v)$$

$$(c_j, c_i) = (0.5 + j, 0.5 + i)$$

$I(x, y)$

$(x_{max}, y_{max})$

$H$

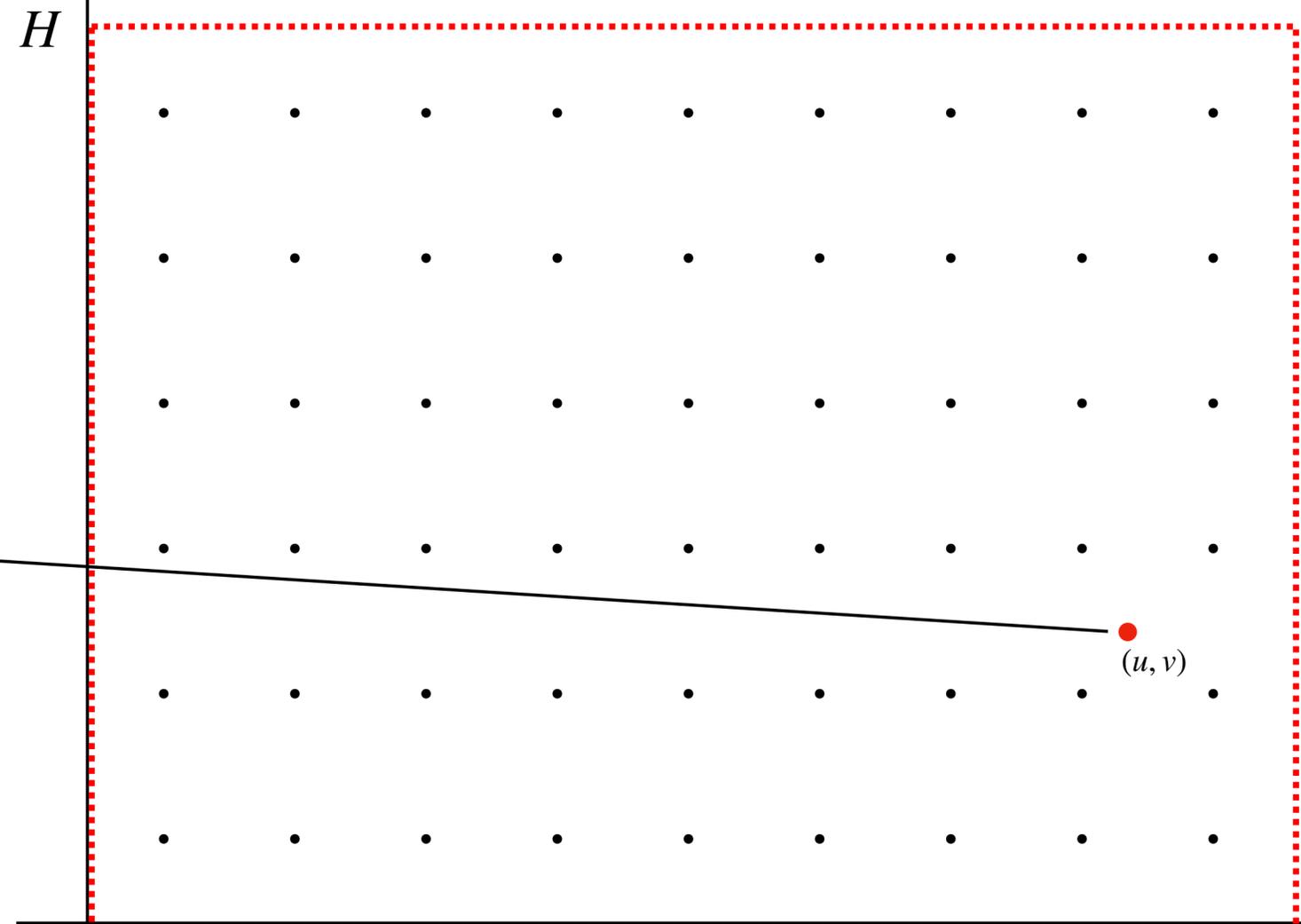


$(x, y) = T(u, v)$

$(u, v)$

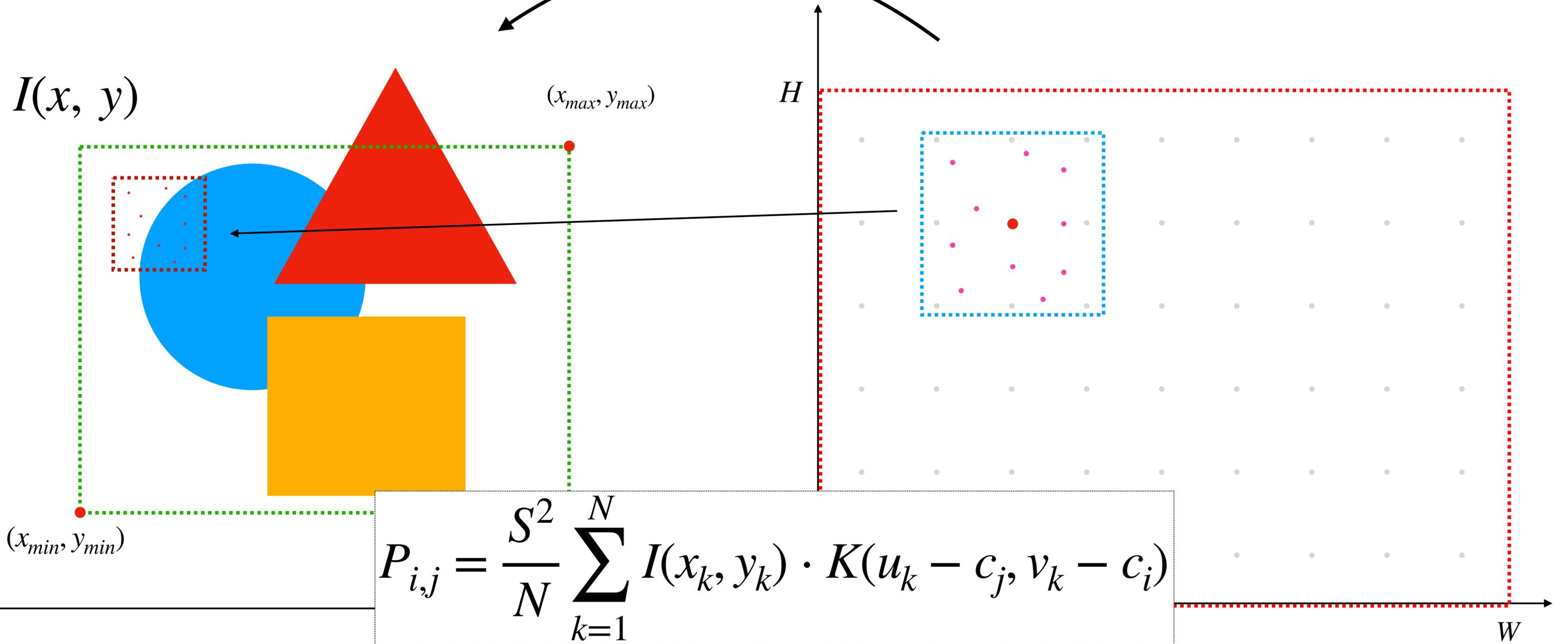
$(x_{min}, y_{min})$

$W$



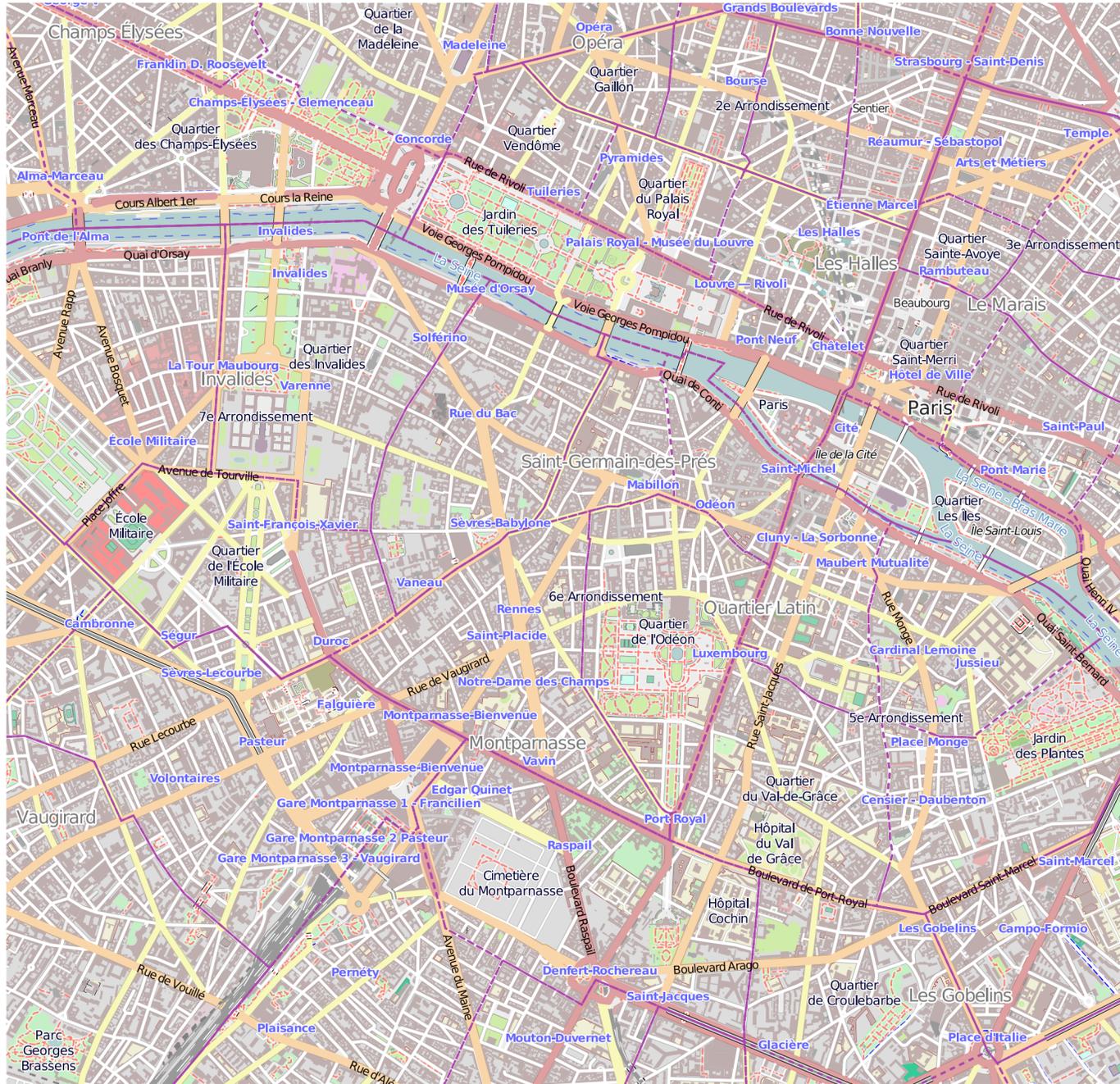
# Amostragem de imagens

$$(x, y) = T(u, v)$$

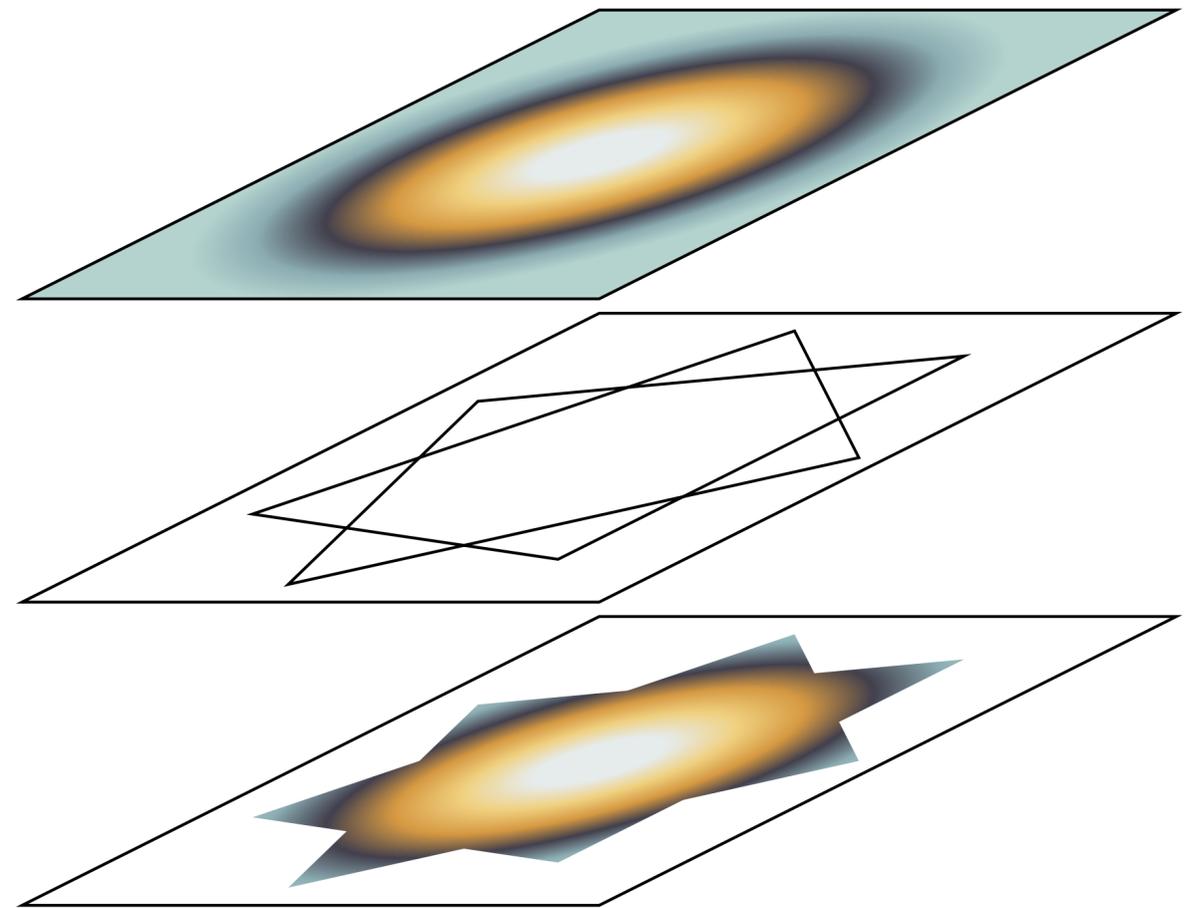
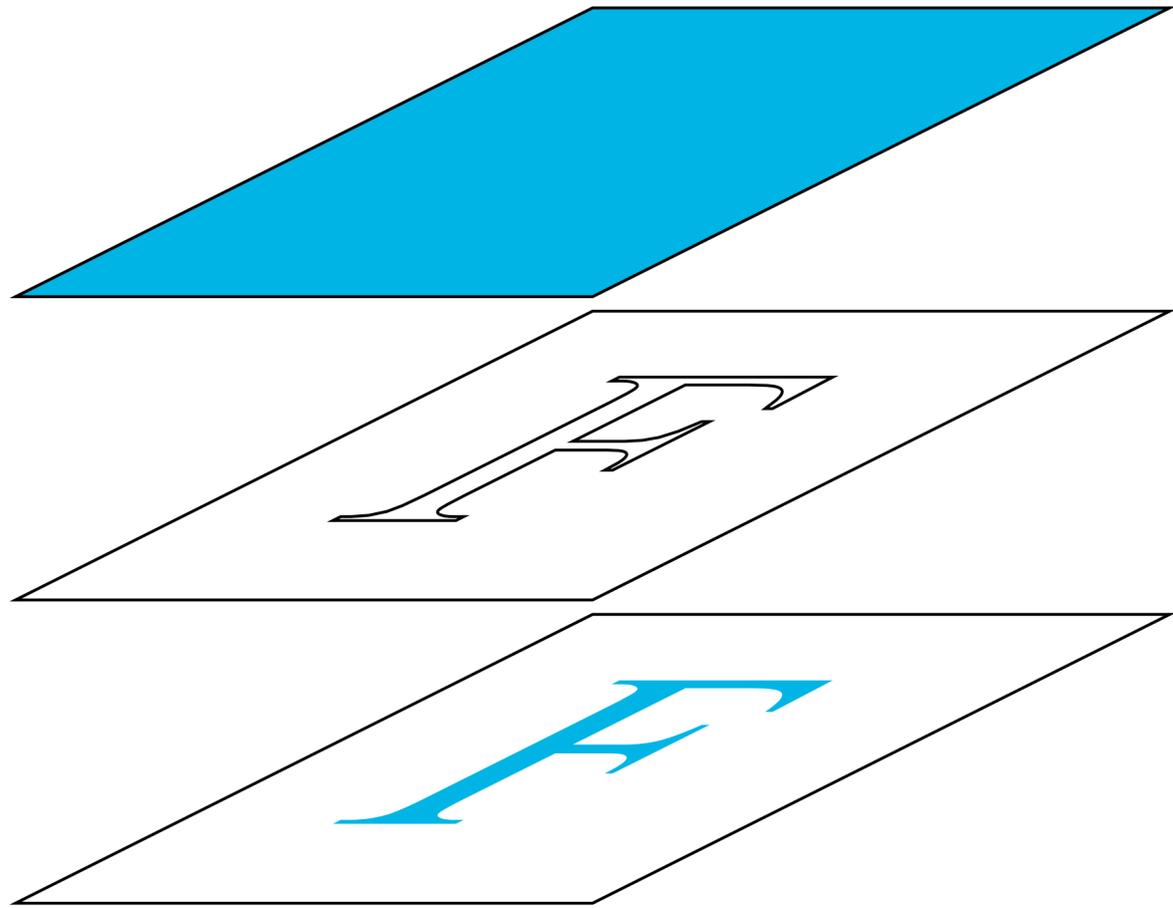


# Vector Graphics

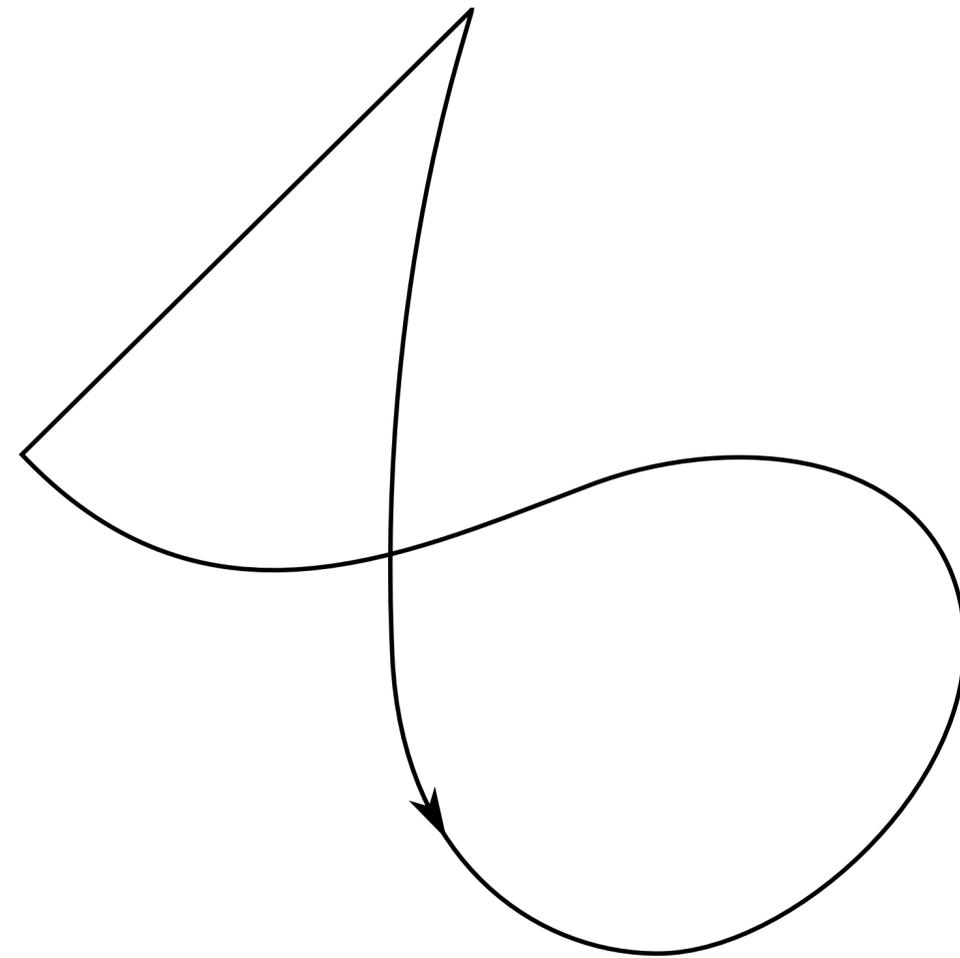
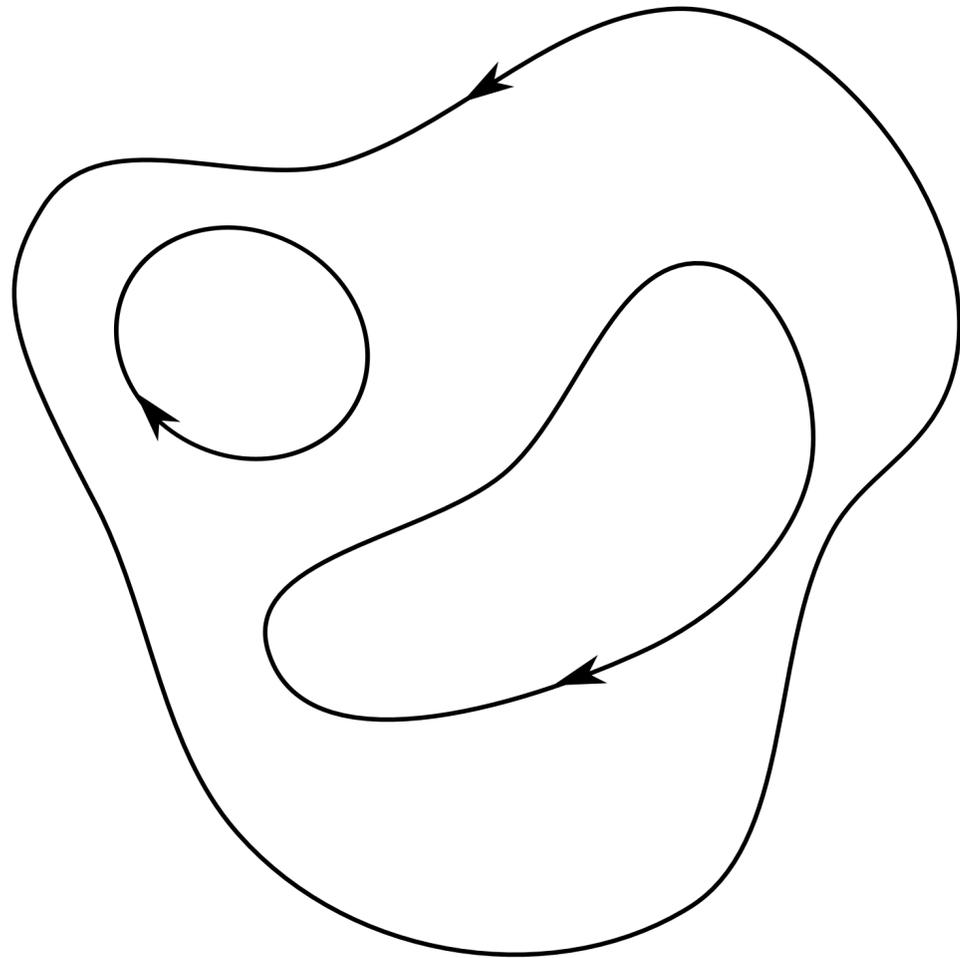
# Vector Graphics



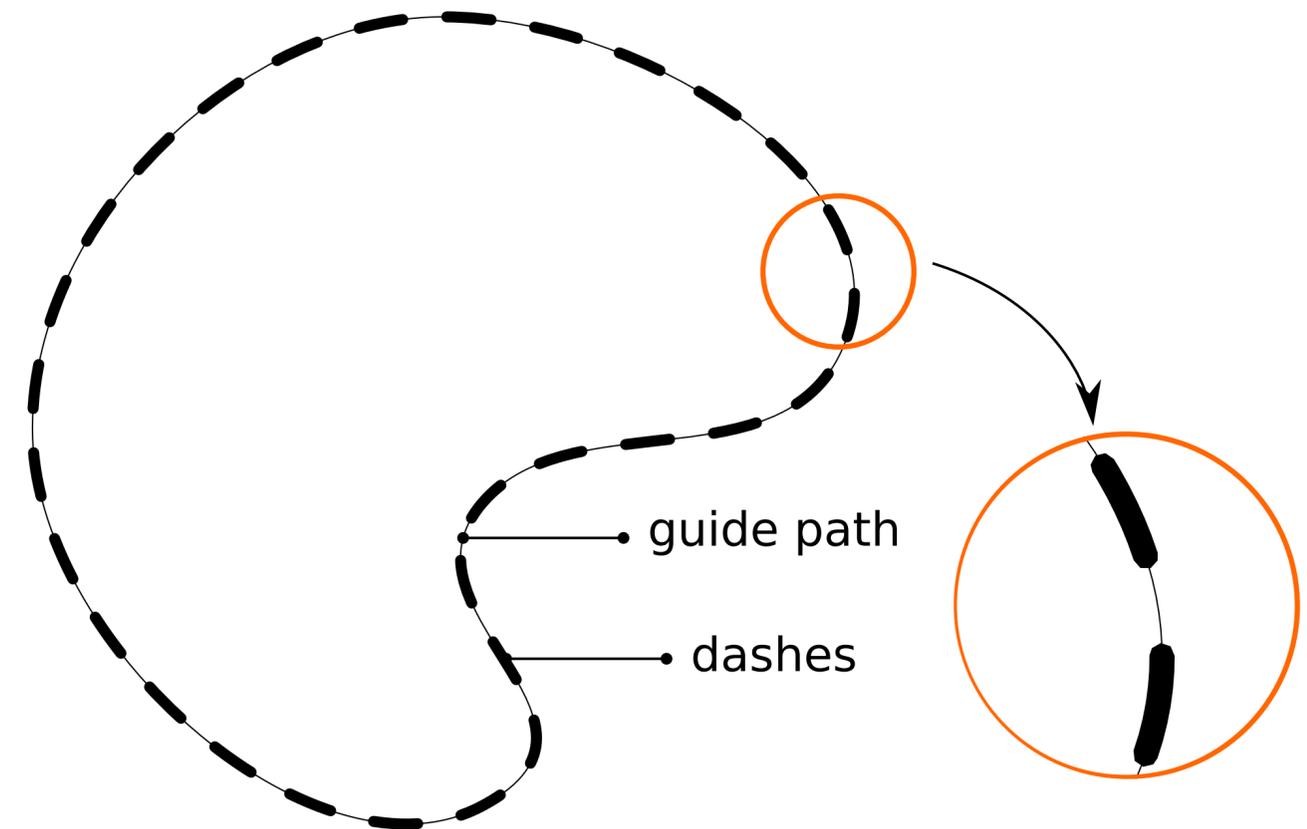
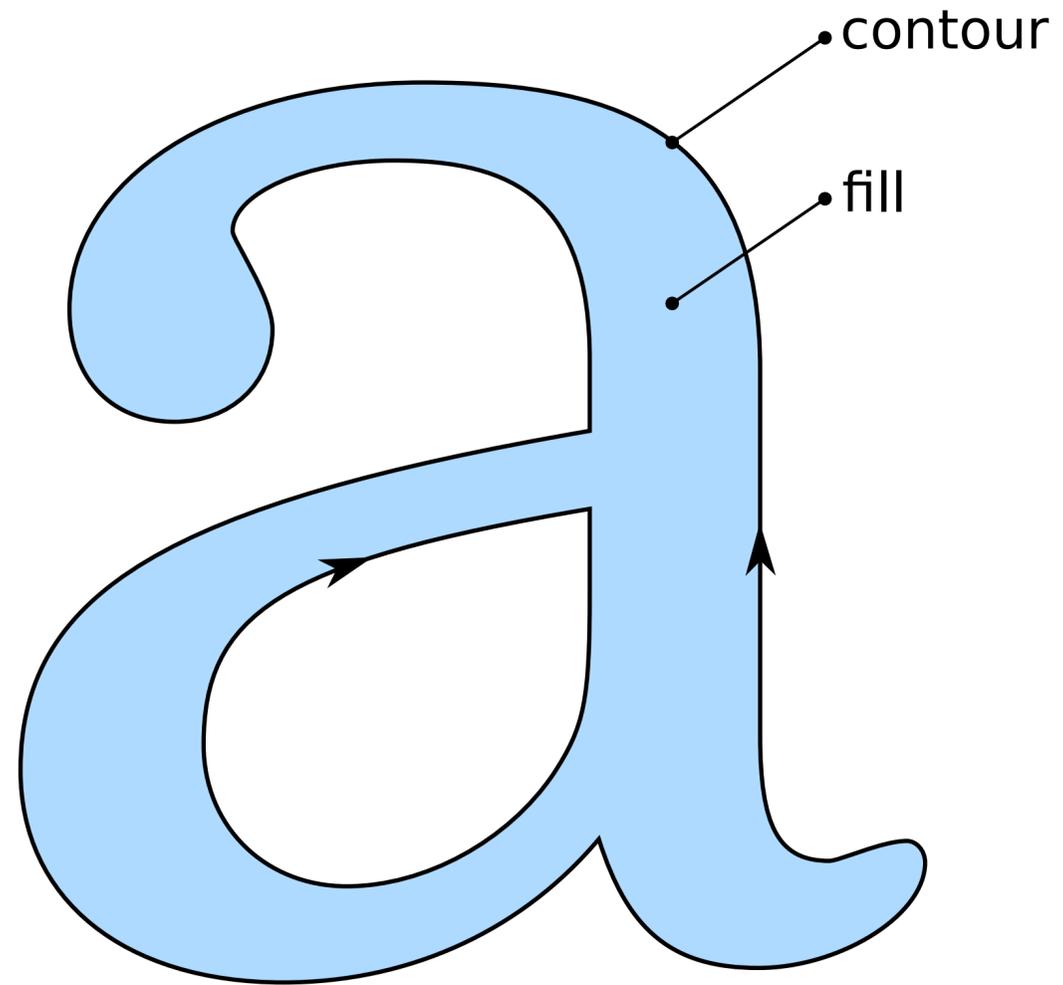
# Metáfora de silkscreen



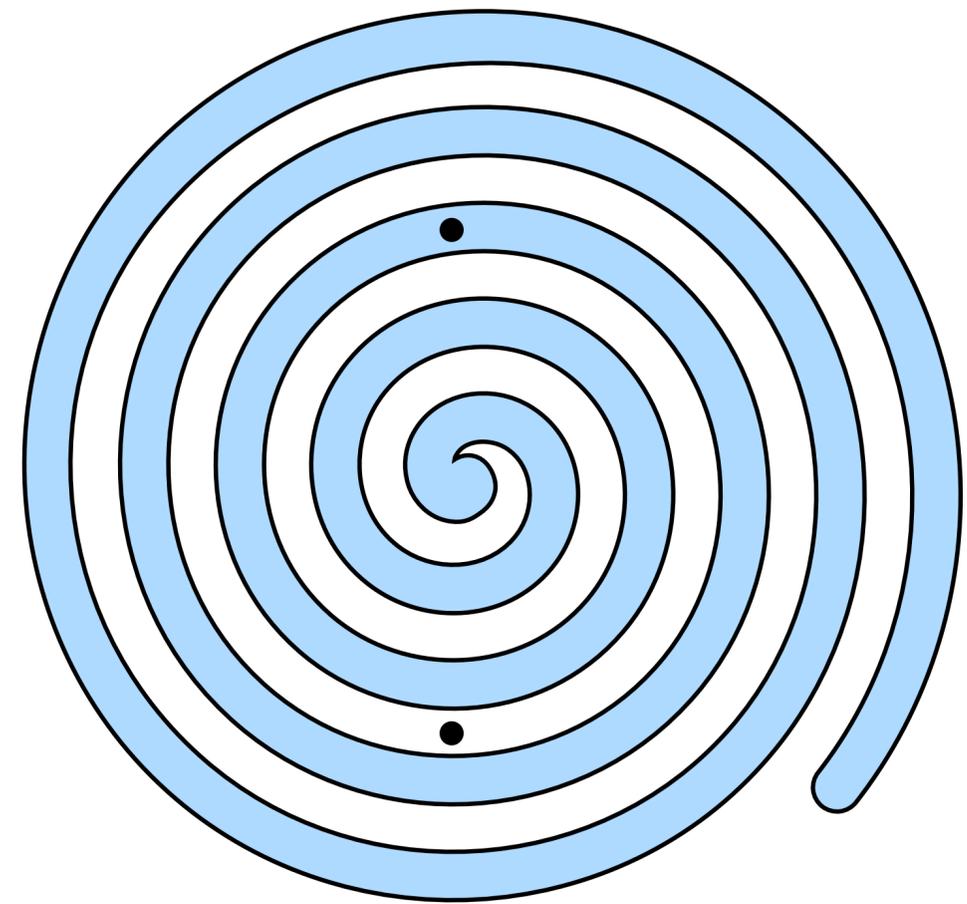
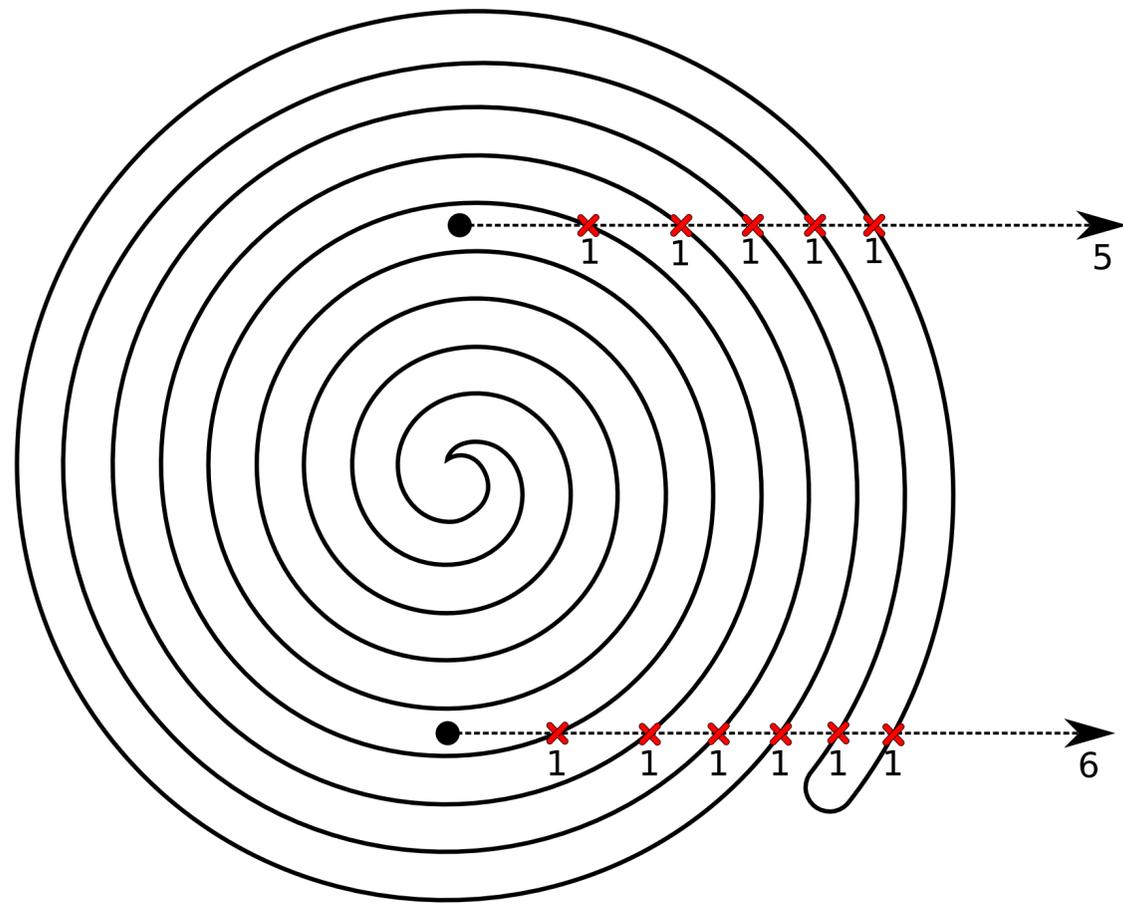
# Contornos



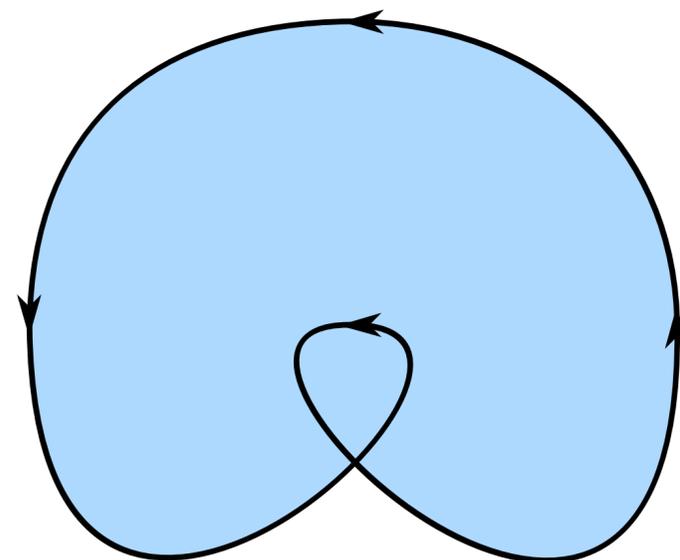
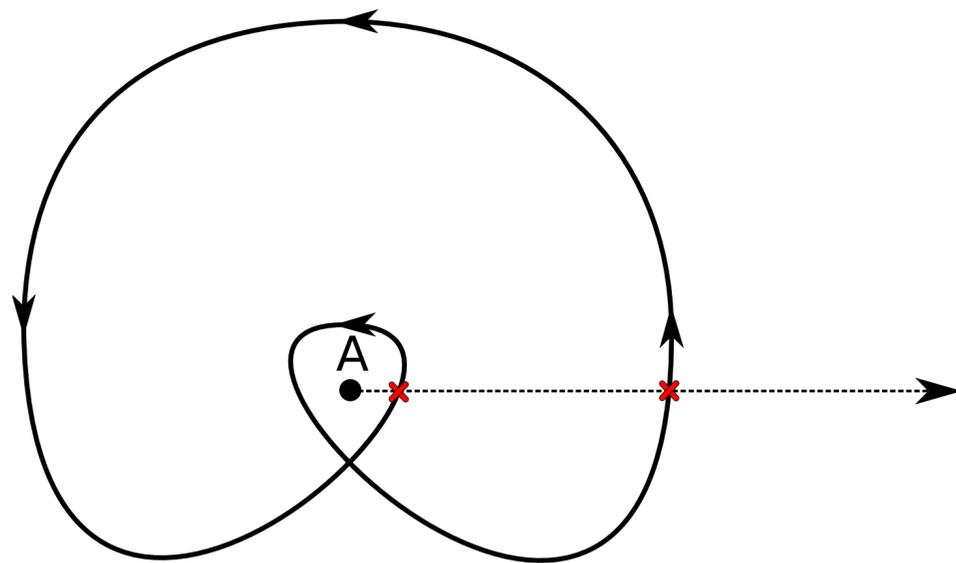
# Fill vs Stroke



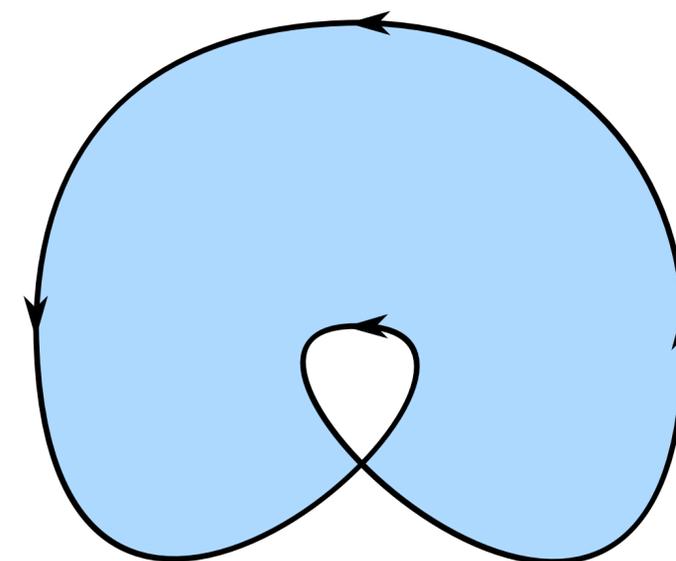
# Point in shape test



# Point in shape test

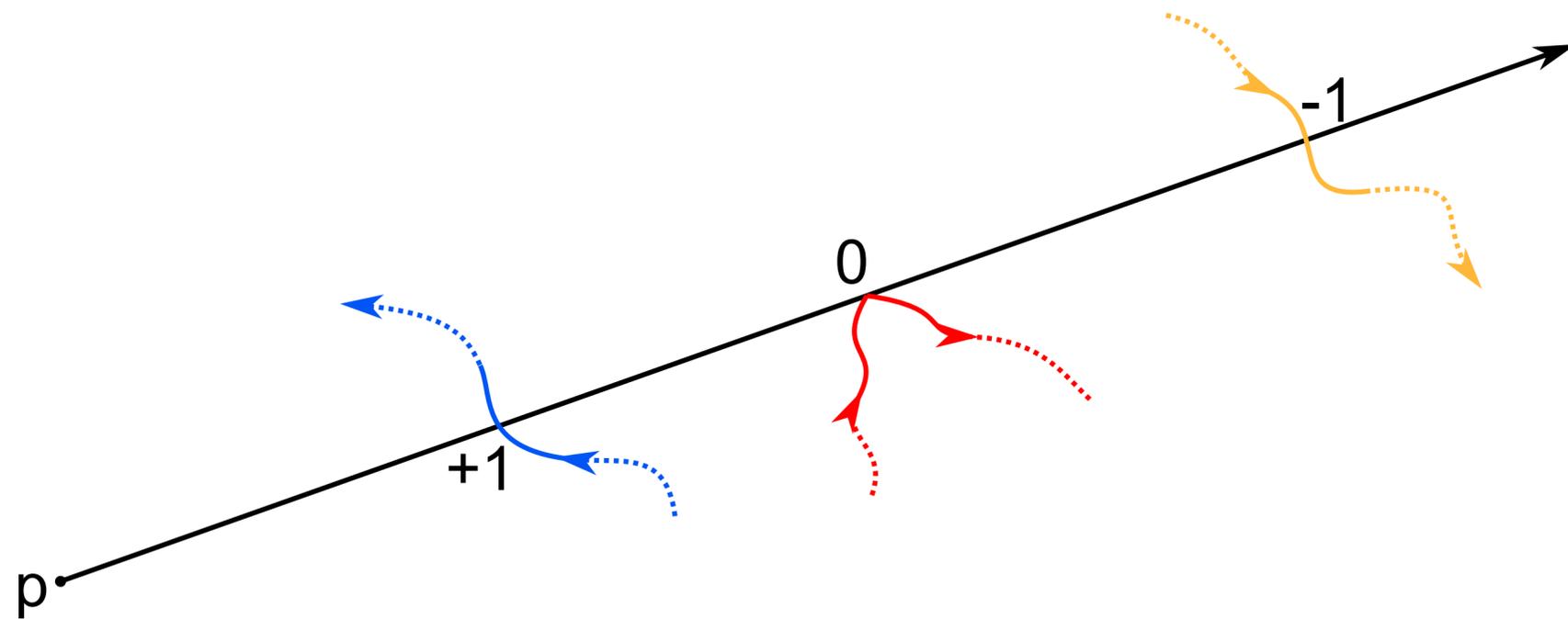


non-zero



even-odd

# Point in shape test

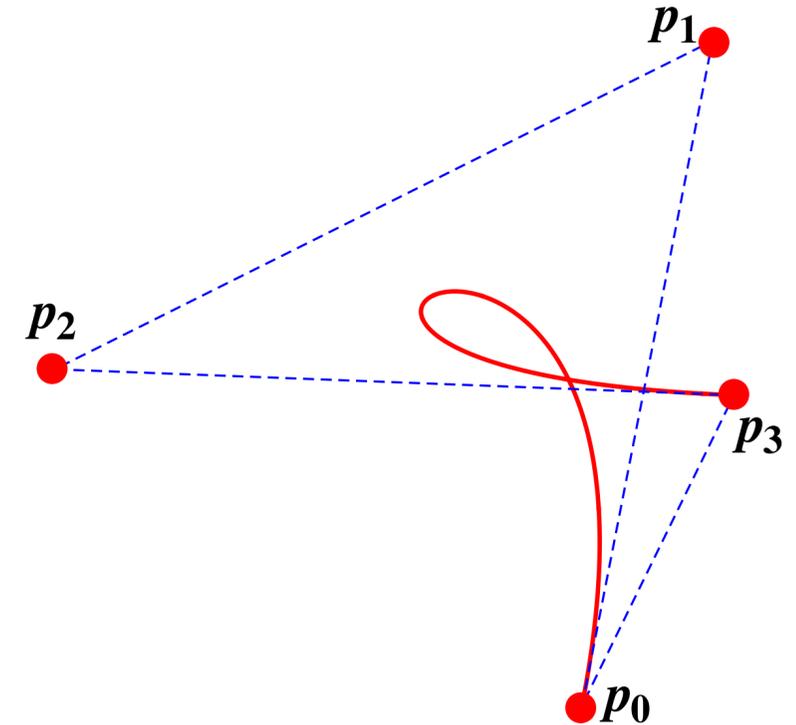
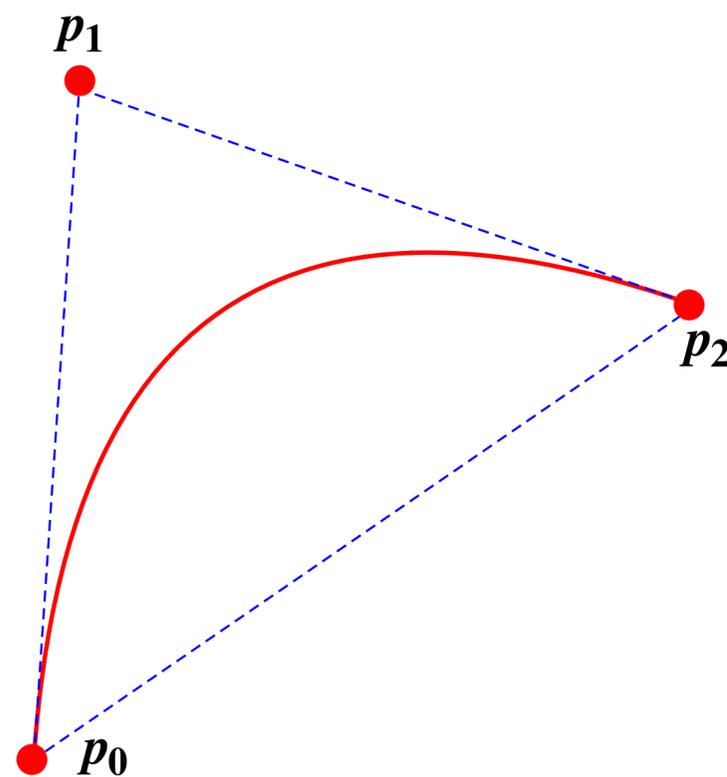
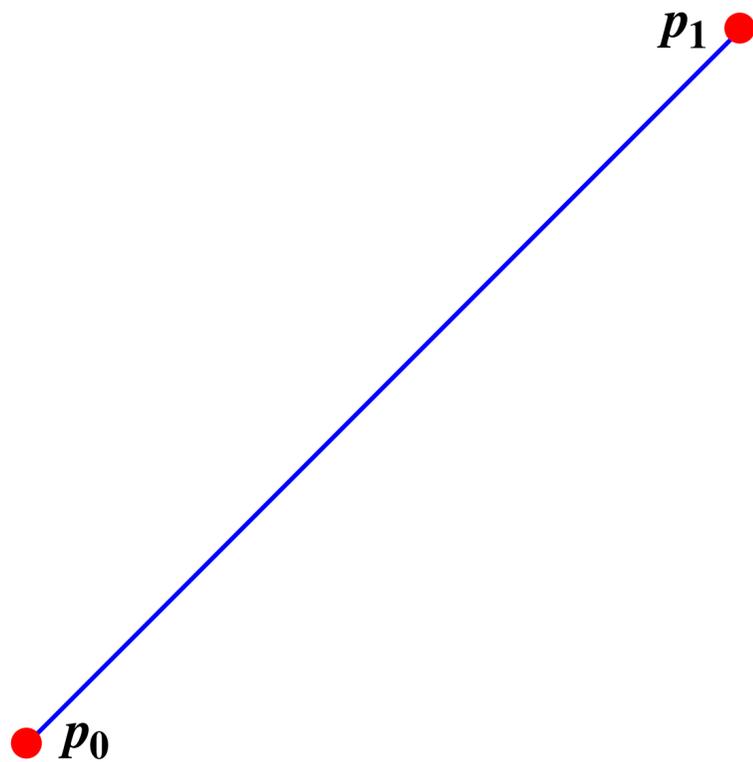


# Bezier segments

$$b_1(t) = p_0(1 - t) + p_1t,$$

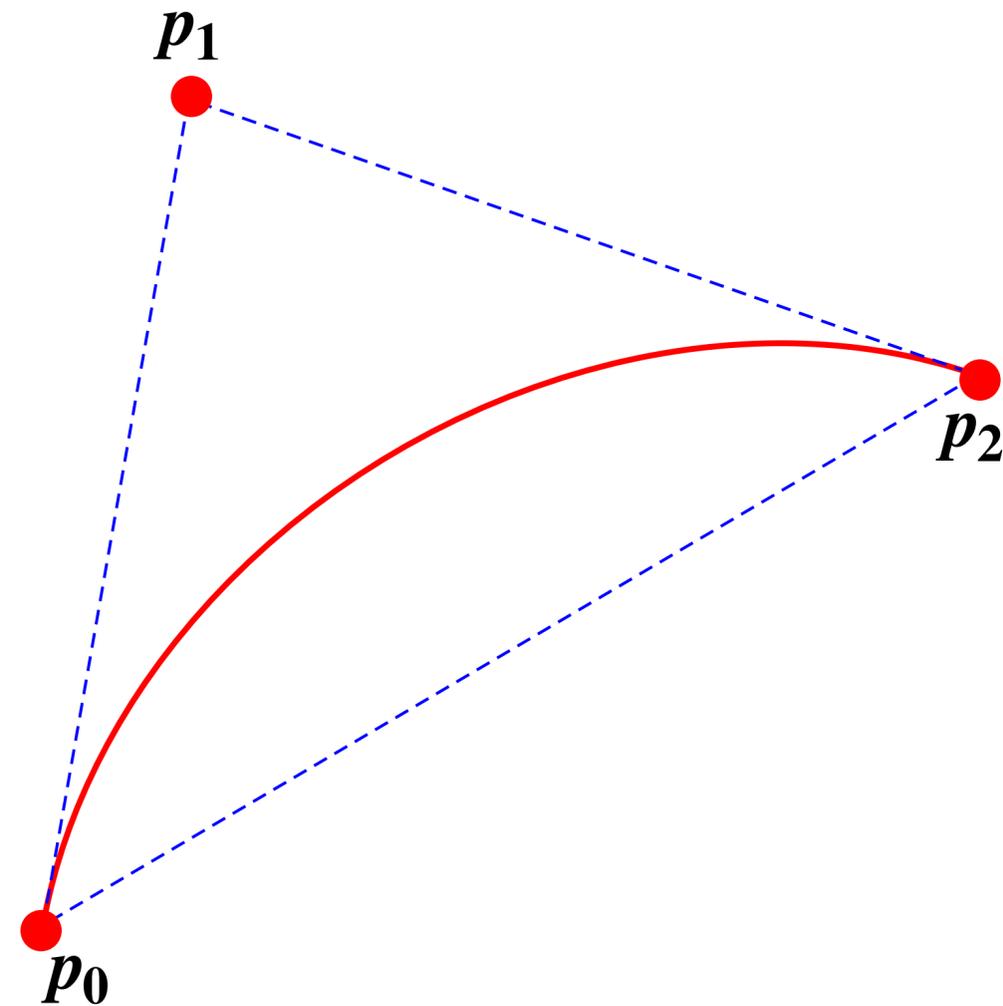
$$b_2(t) = p_2t^2 + p_0(1 - t)^2 + 2p_1t(1 - t),$$

$$b_3(t) = p_3t^3 + 3p_2t^2(1 - t) + p_0(1 - t)^3 + 3p_1t(1 - t)^2.$$



# Rational Bezier Quadratic

$$\begin{aligned} a_2(t) &= \frac{b_2(t)}{\sum_{i=0}^2 B_{i,n}(t)w_i} \\ &= \frac{p_0(1-t)^2 + 2p_1t(1-t) + p_2t^2}{(1-t)^2w_0 + 2t(1-t)w_1 + t^2w_2}. \end{aligned}$$

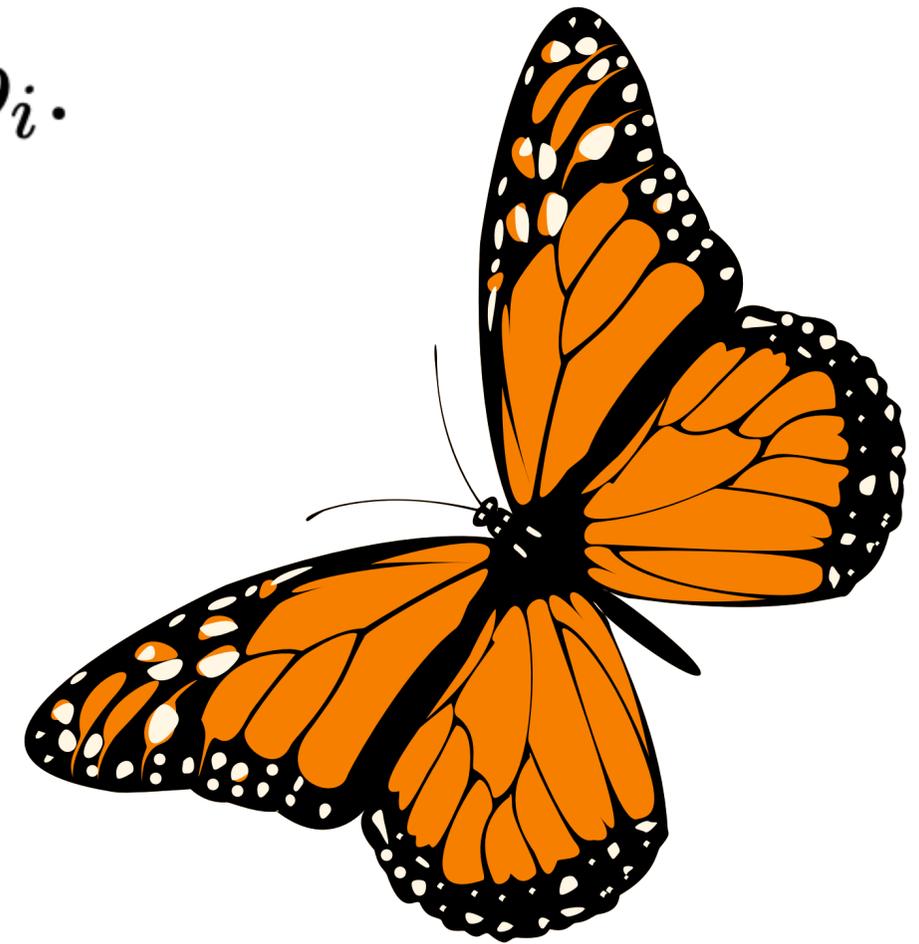
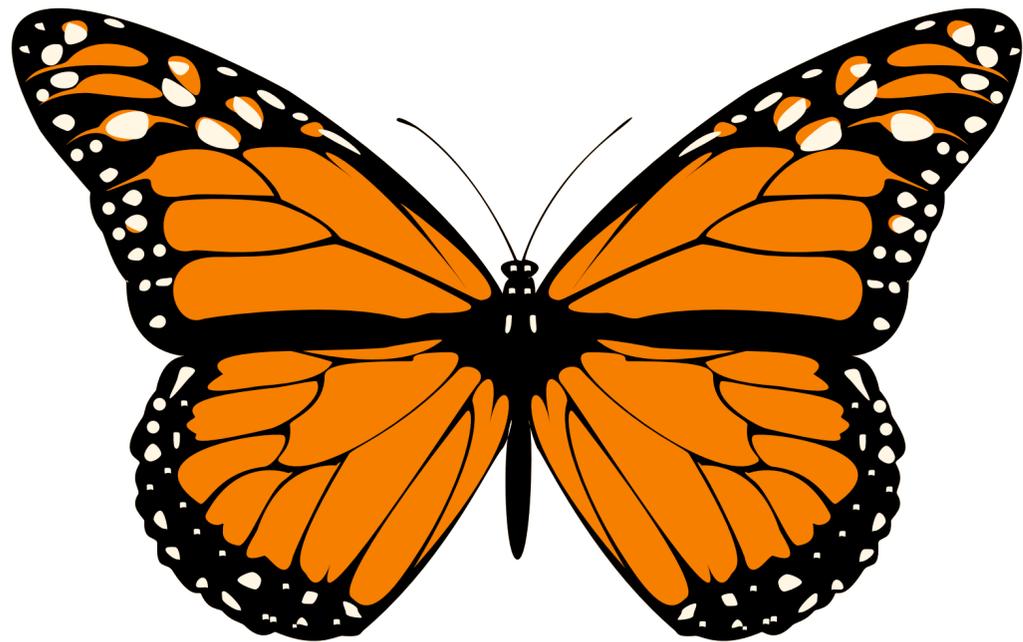


# Point in shape test

```
function is_inside( p, S )
  -- Receives: the point 'p' in  $\mathbb{R}^2$ , and the path 'S' defining a shape.
  -- Returns: true if p is inside the shape, false otherwise.
  r = ray( p, {1, 0} ) -- creates a ray in the direction (1,0).
  wn = 0 -- sets winding number to zero.
  for c in contours( S ) do -- iterates over all contours in S.
    I = intersections( r, c ) -- finds the intersections between r and c
    for {u, t} in I do -- iterates over the intersections
      if u > 0 and t  $\in$  [a,b] then
        -- sum the sign to the intersection
        wn = wn + sign( r, u, c, t )
      end
    end
  end
  -- apply the path's rule: even/odd or non-zero
  return apply_rule( C, wn )
end
```

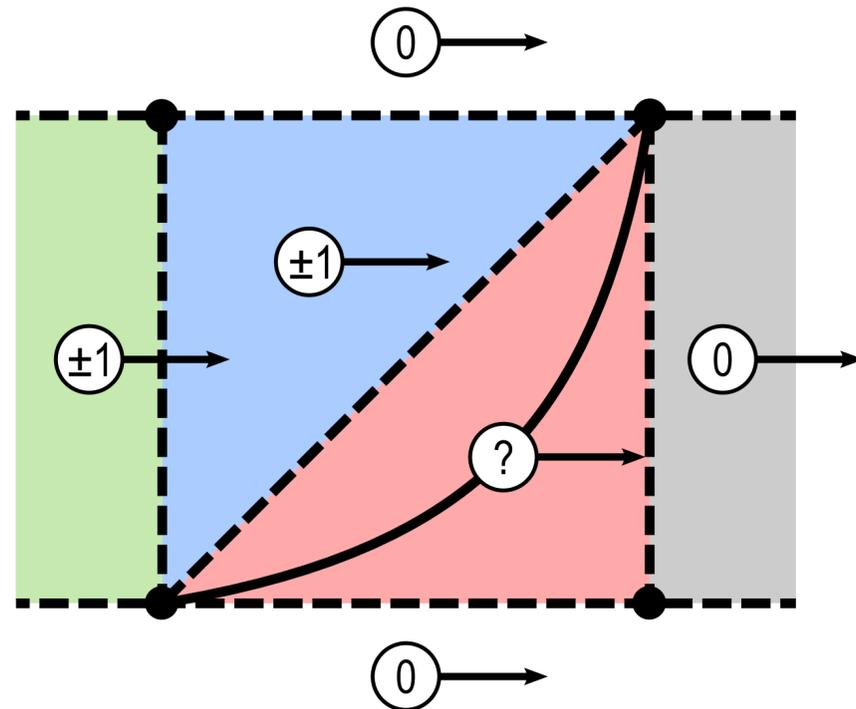
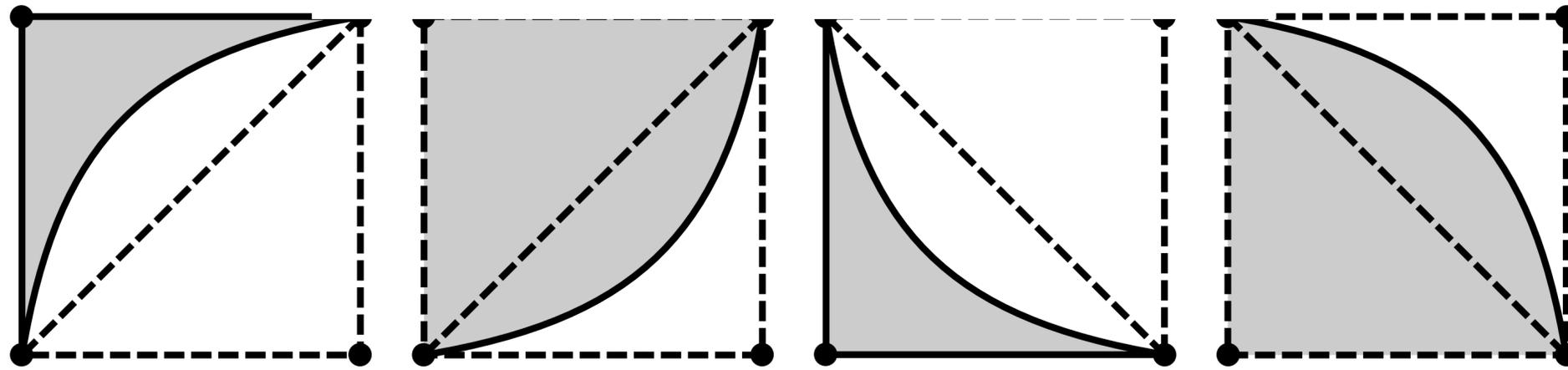
# Propriedade importante de segmentos de Bezier

$$(Ab_n)(t) = \sum_{i=0}^n B_{i,n}(t) Ap_i.$$

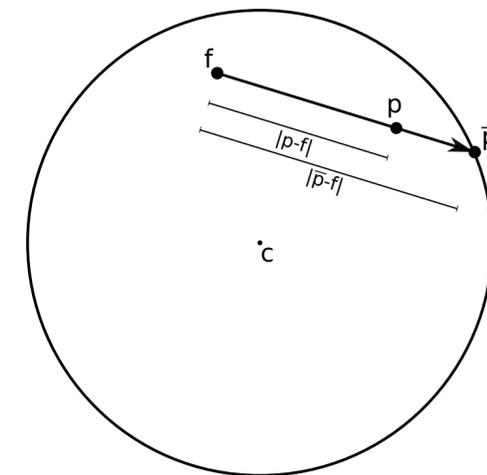
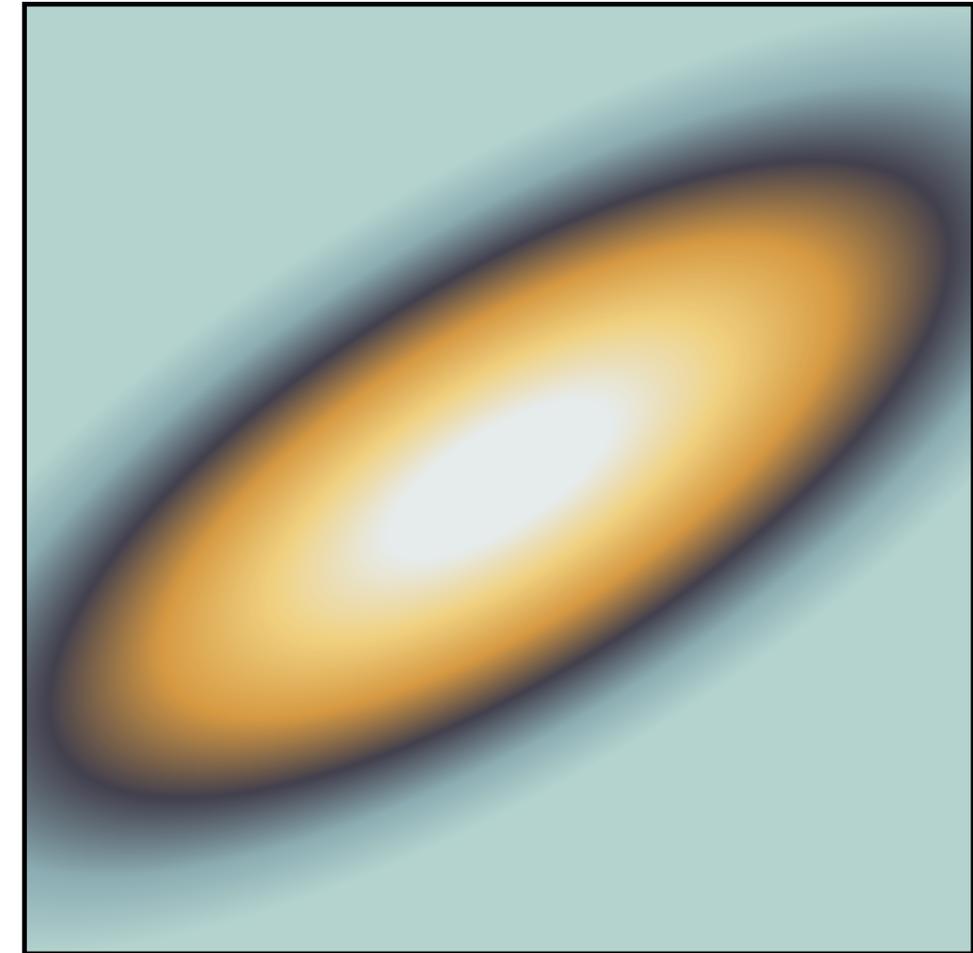
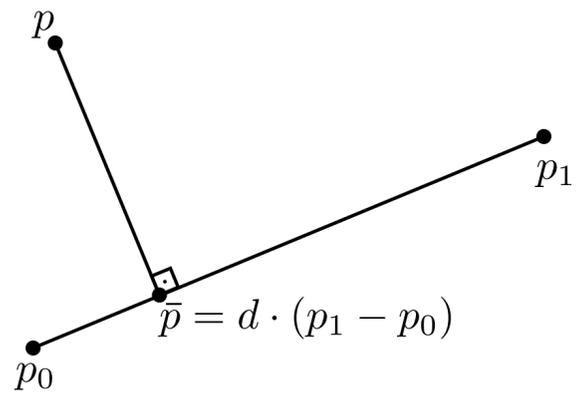
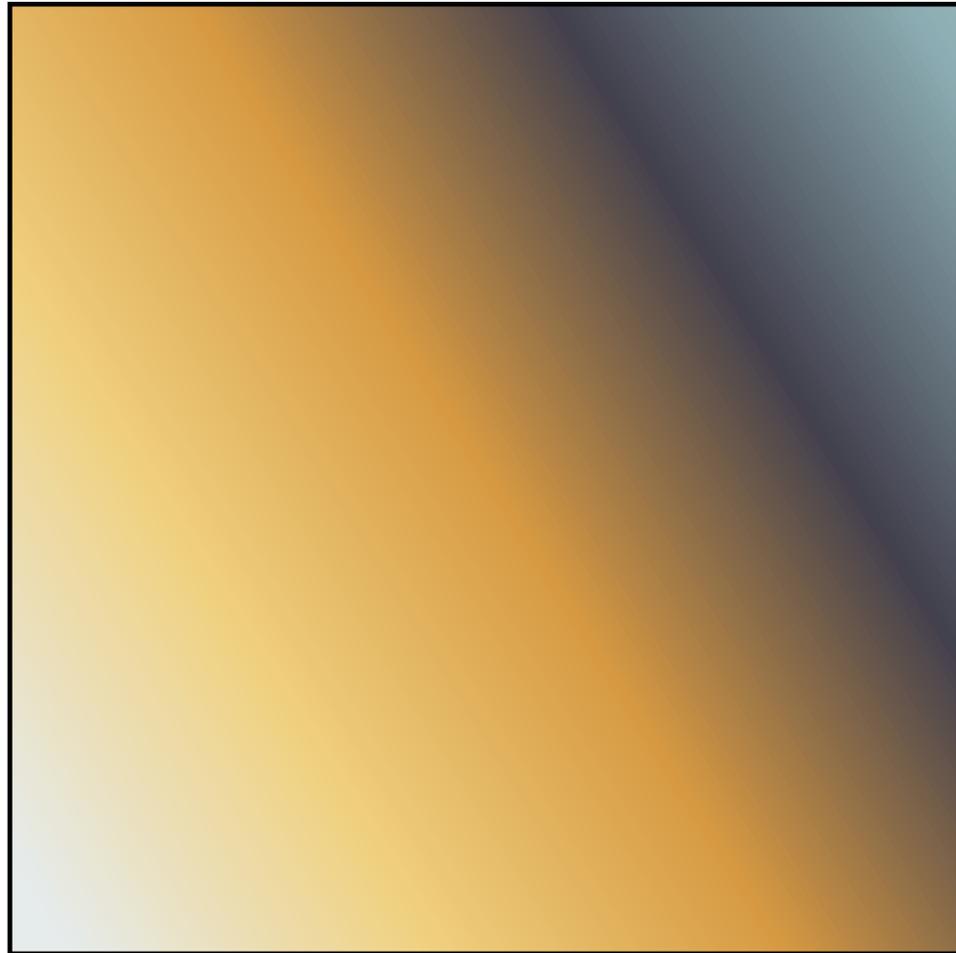


# Otimização: monotonização de segmentos

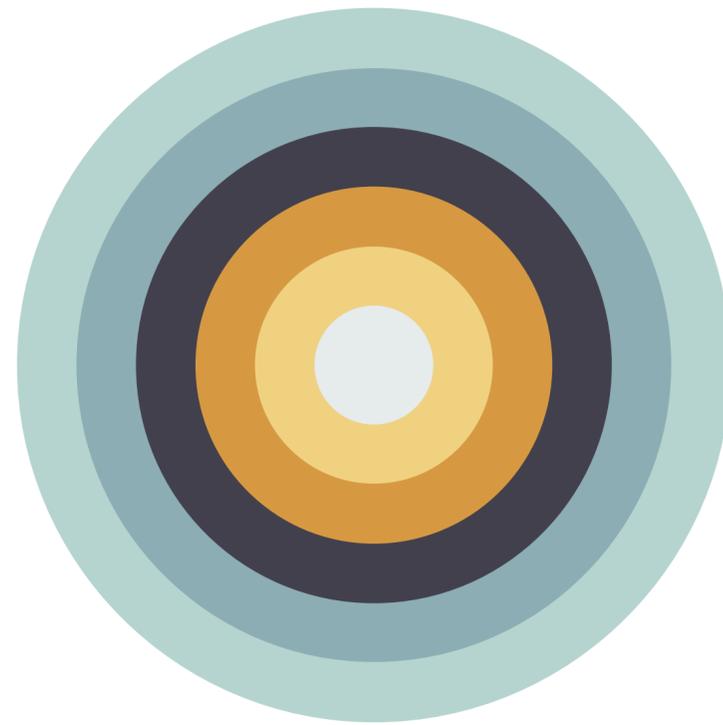
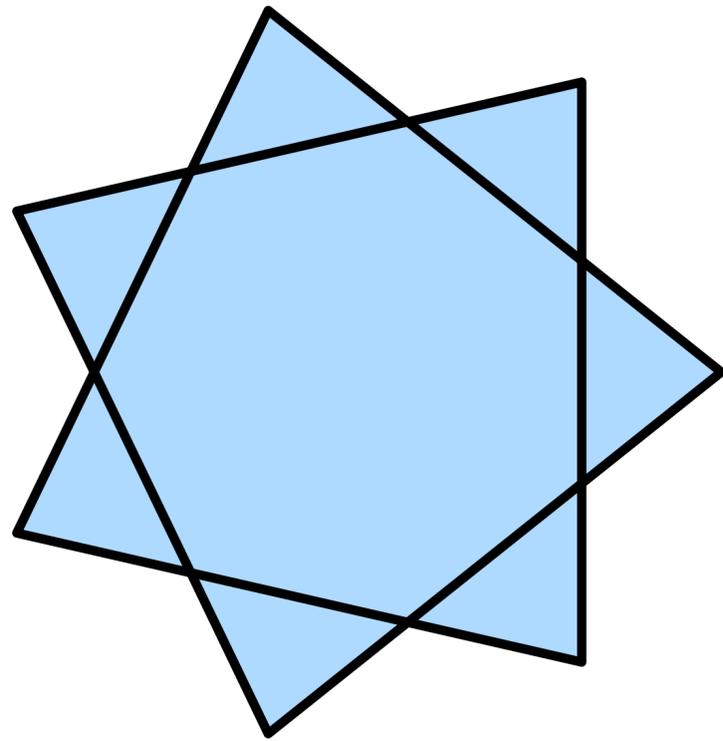
$$x'(t_j) = 0 \quad \text{or} \quad y'(t_j) = 0.$$



# Gradientes



# Clip Path



# Geometria Afim