# Processamento Digital de Imagens

**Tema:**

Filtragem no Domínio da Frequência

**Apresentador:**

Deivison Oliveira Costa

# Sumário

- Transformada de Fourier.
- Filtragem usando Transformada de Fourier.
- Filtro ideal.
- Filtro Gaussiano.

# Transformada de Fourier

- Introdução
  - Definição e objetivo;

# Transformada de Fourier

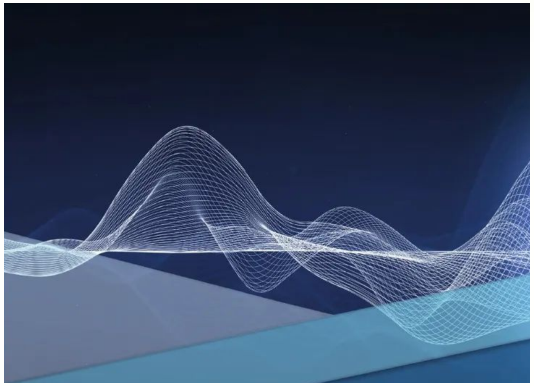
- Definição Formal:
  - A Transformada de Fourier de f(t), denotada como F(ω), é uma função complexa de uma variável ω no domínio da frequência, e é definida como:

$$\hat{f}(\omega) \equiv F(\omega) \equiv \mathcal{F}\{f(t)\} \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(t)\, e^{-i\omega t}\, dt$$

- *F(ω)*: Transformada de Fourier da função f(t) no domínio da frequência;
- *ω*: Frequência angular.

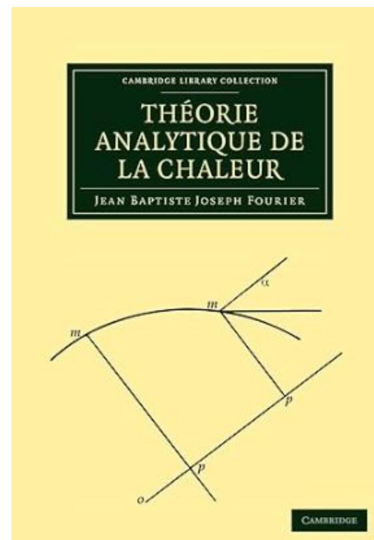
# Transformada de Fourier

- Introdução
  - História;
  - Importância na análise de sinais.



# Transformada de Fourier

- Jean-Baptiste Joseph Fourier (1768-1830)
  - ***Théorie Analytique de la Chaleur*** (1822).

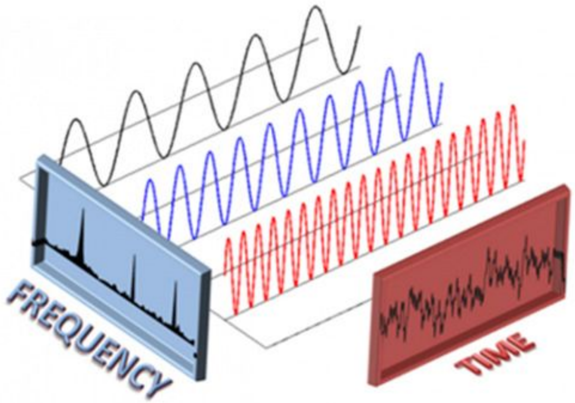


# Transformada de Fourier

- Peter Gustav Lejeune Dirichlet (1805-1859)
  - Teoria de Séries Trigonométricas;
  - Melhoria da representação de **Séries de Fourier.**

# Transformada de Fourier

- Domínio do Tempo vs Domínio da Frequência
  - Principais diferenças;
  - Comparação.



# Transformada de Fourier

- Domínio no Tempo:
  - Um sinal é representado como uma função que varia com o tempo;
  - Descreve como o sinal muda ao longo do tempo.
- Domínio na Frequência:
  - Um sinal é decomposto em suas componentes de frequência;
  - Descreve as diferentes frequências que compõem o sinal e suas amplitudes relativas.

# Transformada de Fourier

- Ilustração:

# Transformada de Fourier

- Transformada de Fourier Contínua (CFT)
  - Explicação;
  - Formulação.

# Transformada de Fourier

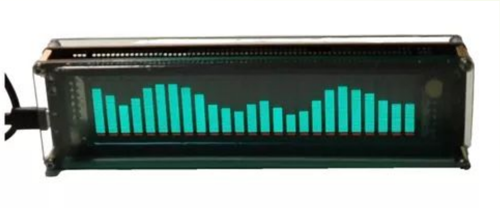
- Transformada de Fourier Contínua (CFT)

$$X(f) = \int_{-\infty}^{+\infty} x(t)\, e^{-j2\pi ft} dt$$

  - *X(f)*: representação no domínio da frequência do sinal contínuo x(t) na frequência f;
  - *x(t)*: sinal contínuo no domínio do tempo;
  - *f*: frequência em Hertz (Hz);
  - *j*: unidade imaginária

# Transformada de Fourier

- Onde usamos isso??
  - Análise de espectro de áudio;
  - Processamento de sinais de comunicação;
  - Processamento de imagens;
  - Análise de vibrações mecânicas;
  - Análise de sinais biomédicos;
  - Síntese de sinais.




# Transformada de Fourier

- Implementando em Python
  - Exemplo de CFT

INSTITUTO FEDERAL
Minas Gerais

# Transformada de Fourier

- Transformada de Fourier Discreta (DFT)
  - Explicação;
  - Formulação.

# Transformada de Fourier

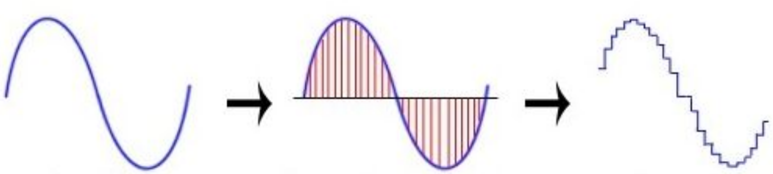
- Transformada de Fourier Discreta (DFT)

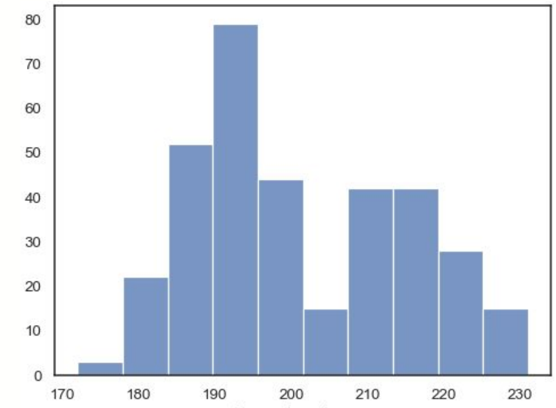
$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\left(\frac{2\pi}{N}\right)kn}$$

  - *X[k]*: componente no domínio da frequência no índice k.
  - *x[n]*: amostra do sinal no domínio do tempo no índice n.
  - *N*:  número total de amostras no sinal.
  - *j*: unidade imaginária
  - *k*: índice da frequência desejada, variando de 0 a N - 1.

# Transformada de Fourier

- Discretização de Sinais e Amostragem.
  - Processamento Digital de Sinais;
  - Análise de Histogramas;
  - Análise de Dados Temporais.






INSTITUTO FEDERAL
Minas Gerais

# Transformada de Fourier

- Implementando em Python
  - Exemplo de DFT

# Transformada de Fourier

- Transformada Rápida de Fourier (FFT)
  - Ideia do Algoritmo:
    - Cooley e Tukey (1965);
    - Divisão e Conquista.
  - Eficiência Computacional:
    - N: tamanho da sequência de entrada;
    - DFT: *O(N^2)*;
    - FFT: *O(N log N)*.
  - Aplicações:
    - Redução de ruído;
    - Compressão de dados;

# Transformada de Fourier

- Exemplo:
  - **Sequência Original:**
    - x[0], x[1], x[2], x[3], x[4], x[5], x[6], x[7]
  - **Divisão:**
    - Subsequência 1: x[0], x[2], x[4], x[6]
    - Subsequência 2: x[1], x[3], x[5], x[7]
  - **Cálculo das FFTs das Subsequências:**
    - FFT(Subsequência 1)
    - FFT(Subsequência 2)
  - **Combinação das FFTs das Subsequências:**
    - FFT(Sequência Original) = Combinar(FFT(Subsequência 1), FFT(Subsequência 2))

# Transformada de Fourier

- Foco agora em processamento digital de imagens:
    - Onde usamos FFT?
        - Filtragem no Domínio da Frequência;
        - Análise Espectral;
        - Compressão de Imagens;
        - Correção de Distorção;
        - Análise de Textura;
        - Super-resolução;
        - Detecção de Movimento.

# Transformada de Fourier

- Implementando em Python
  - Exemplo de FFT

# Transformada de Fourier

- Propriedades Gerais da Transformada de Fourier
  - Linearidade e convolução no domínio da frequência;
  - Teorema da Convolução.

# Transformada de Fourier

- Princípio da Linearidade
  - Definição Formal:
    - Na existência de um sinal composto por uma combinação linear de outros sinais, a Transformada de Fourier deste sinal composto será a combinação linear das Transformadas de Fourier dos sinais individuais.
  - Utilidade:
    - Possibilita a análise de sinais compostos por várias componentes.

# Transformada de Fourier

- Exemplo:
  - Seja $a \cdot x(t) + b \cdot y(t)$ a representação de um sinal:

$$F\left(a \cdot x(t) + b \cdot y(t)\right) = a \cdot X(f) + b \cdot Y(f)$$

  - *F(*sinal*):* Transformada de Fourier do sinal $a \cdot x(t) + b \cdot y(t)$.
  - *X(f)*: Transformada de Fourier de *x(t)*.
  - *Y(f)*: Transformada de Fourier de *y(t)*.
  - *a* e *b*: constantes reais.

# Transformada de Fourier

- Princípio da Convolução:
  - O que significa convolução?
  - Definição Formal:
    - A convolução no domínio do tempo se traduz na multiplicação no domínio da frequência;
  - Utilidade:
    - Possibilita a análise do comportamento de sistemas complexos no domínio da frequência de maneira mais eficiente.

# Transformada de Fourier

- Exemplo:
  - Sejam x(t) e h(t) dois sinais no domínio do tempo:

$$F\big(x(t)\cdot h(t)\big) = X(f)\times H(f)$$

  - *F(x(t) * h(t))*: representa a Transformada de Fourier do sinal resultante da convolução entre *x(t)* e *h(t)*.
  - *X(f)*: Transformada de Fourier de *x(t).*
  - *H(f)*:*Transformada de Fourier de h(t).*
  - *'*x*'*: denota multiplicação no domínio da frequência.

# Filtragem Usando Fourier

- **Transformada de Fourier 2D;**
- ~~Transformada de Fourier 3D.~~

# Filtragem Usando Fourier

- Transformada de Fourier 2D
  - Qual a finalidade?
    - Analisar a distribuição das frequências em uma imagem bidimensional.
  - O que posso fazer com isso?
    - Filtragem de imagens;
    - Compressão;
    - Detecção de bordas;
    - Análise espectral de texturas.

# Filtragem Usando Fourier

- Formulação

$$F(u,v) = \iint_{-\infty}^{+\infty} f(x,y)\, e^{-j2\pi(ux+vy)}\, dxdy$$

  - *F(u, v)*: Representação no domínio da frequência da imagem original *f(x,y)*;
  - *f(x, y)*: Imagem original que você deseja analisar no domínio espacial. Ela é uma função de duas variáveis, *x* e *y*, que representam as coordenadas no plano 2D.
  - *u* e *v*: São as variáveis no domínio da frequência e correspondem às coordenadas no espaço da frequência.

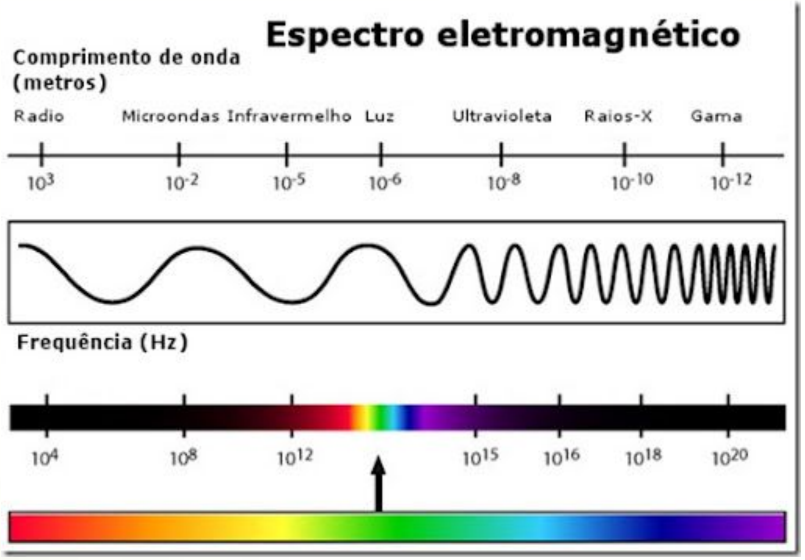
# Filtragem Usando Fourier

$$F(u,v) = \iint_{-\infty}^{+\infty} f(x,y)\, e^{-j2\pi(ux+vy)}\, dxdy$$

$$F(u,v,w) = \iiint_{-\infty}^{+\infty} f(x,y,z)\, e^{-j2\pi(ux+vy+wz)}\, dxdydz$$

# Filtragem Usando Fourier

- Ideia Geral:
  - Decompor a imagem f(x, y) em componentes de frequência.




# Filtragem Usando Fourier

- Implementando em Python com NumPy e OpenCV
  - Exemplo de FFT 2D

# Filtragem Ideal

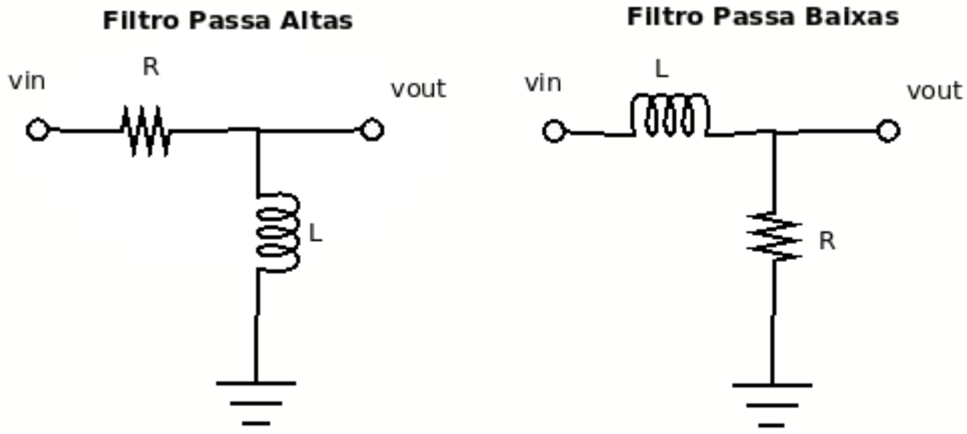
- Filtro Passa-Baixa
  - Função de transferência
    - 1 - para frequências abaixo da frequência de corte;
    - 0 - para frequências acima da frequência de corte.
  - O que podemos fazer na imagem?
    - Reduzir o ruído em imagens ou aplicar efeitos de borramento.

# Filtragem Ideal

- Filtro Passa-Alta
  - Função de transferência
    - 0 - para frequências abaixo da frequência de corte;
    - 1 - para frequências acima da frequência de corte.
  - O que podemos fazer na imagem?
    - Realçar as bordas em uma imagem.

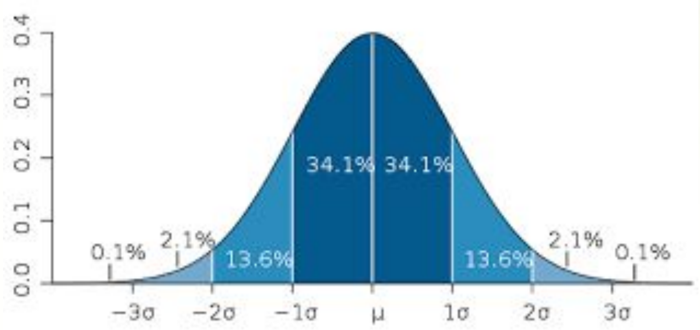
# Filtragem Ideal

- Comparação:



# Filtragem Gaussiana

- Definição
  - Função Gaussiana
    - Estatística;
    - Processamento de sinais.
  - O que podemos fazer na imagem?
    - Redução de ruído;
    - Suavização de imagens;
    - Detecção de bordas.




$$G(x) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{(x-\mu)^2}{2\sigma}}$$

# Filtragem Gaussiana

- Fórmula para o plano 2D:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

- Onde:
  - *G(x,y)*: valor do filtro Gaussiano em uma determinada posição (*x, y*);
  - *σ*: desvio padrão (ou escala) da função Gaussiana;
  - << *σ* → << suavização.

# Filtragem Gaussiana

- Implementando em Python com NumPy e OpenCV
  - Exemplo de Filtragem Gaussiana;
  - ~~***cv2.GaussianBlur(image, (0, 0), sigma)***~~

# Referências


- **Transformada de Fourier:**
    - FOLLAND, Gerald B. ***Fourier Analysis and Its Applications***. [Inglês]. Belmont, Califórnia: Brooks/Cole, 1992.
- **Processamento Digital de Imagens:**
    - GONZALES, Rafael C.; WOODS, Richard E. ***Digital Image Processing.*** [Inglês]. 4. ed. Upper Saddle River, NJ: Prentice Hall, 2018.
    - SONKA, Milan; HLAVAC, Vaclav; BOYLE, Roger. ***Image Processing, Analysis, and Machine Vision.*** [Inglês]. 4. ed. Cengage Learning, 2014.
- **Python:**
    - MCKINNEY, Wes. ***Python for Data Analysis***. [Inglês]. O'Reilly Media, 2017.
    - Doc Numpy - FFT;
    - Doc OpenCV - Processamento/Tratamento de Imagens.
- **Filtros Passa-Baixa/Alta e Gaussiano:**
    - GONZALES, Rafael C.; WOODS, Richard E. ***Digital Image Processing***. [Inglês]. 4. ed. Upper Saddle River, NJ: Prentice Hall, 2018.
    - PROAKIS, John G.; MANOLAKIS, Dimitris G. ***Digital Signal Processing: Principles, Algorithms, and Applications.*** [Inglês]. 4. ed. Prentice Hall, 2006.