TRANSFORMADA WAVELET

E SUAS APLICAÇÕES NO PROCESSAMENTO DE IMAGENS

HELDER C. R. DE OLIVEIRA
PROF. MARCELO A. C. VIEIRA

heldercro@usp.br

http://helderc.github.io

SUMÁRIO

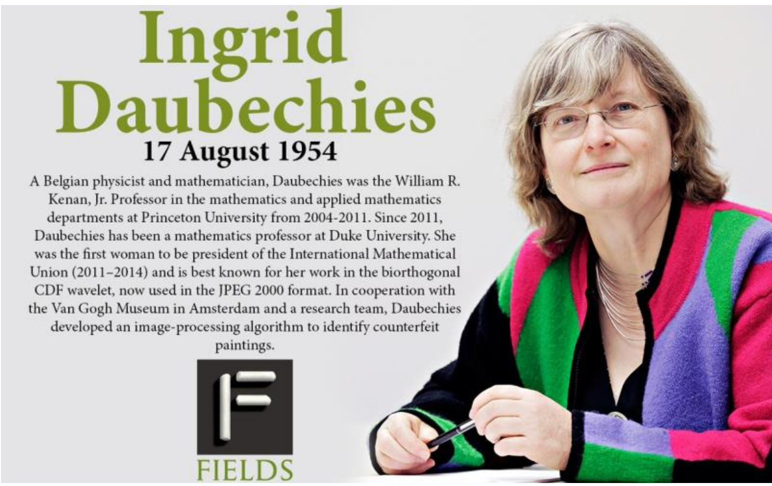
- **Introdução:**
  - Transformada de Fourier;
  - Transformada Janelada de Fourier;
  - Transformada Wavelet:
    - Contínua;
    - Discreta;
    - Análise multiresolução;
  - Bases Wavelet;
  - `WaveletAnalyzer` (MATLAB);

- **Aplicações:**
  - Detecção de arestas;
  - Remoção de ruído.

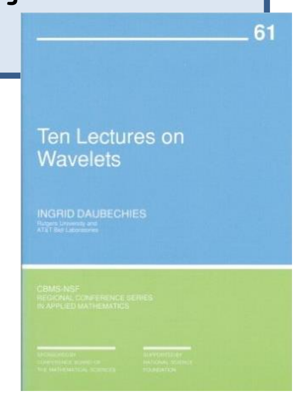
- **Bibliografia e referências.**

POR QUE WAVELETS?

- **Mais uma transformada?**



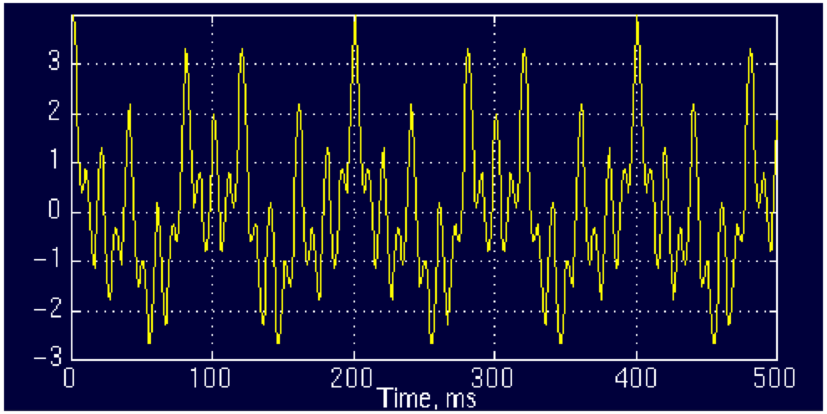
"Assim como um número precisa ser convertido entre diferentes unidades para realização de cálculos, os sinais também precisam dessa mudança."
(I. Daubechies)

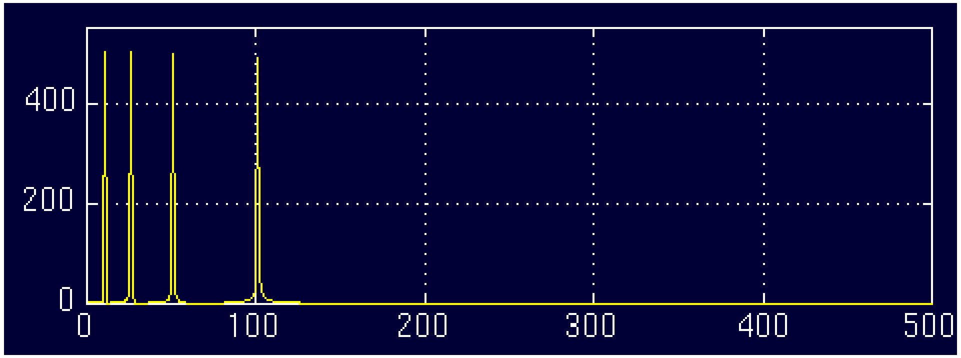


- **Aplicações em diversas áreas:**
  - Mapeamento de tráfego;
  - Remoção de ruído;
  - Compressão de dados (ex.: JPEG-2000);
  - Reconhecimento de padrões, etc...

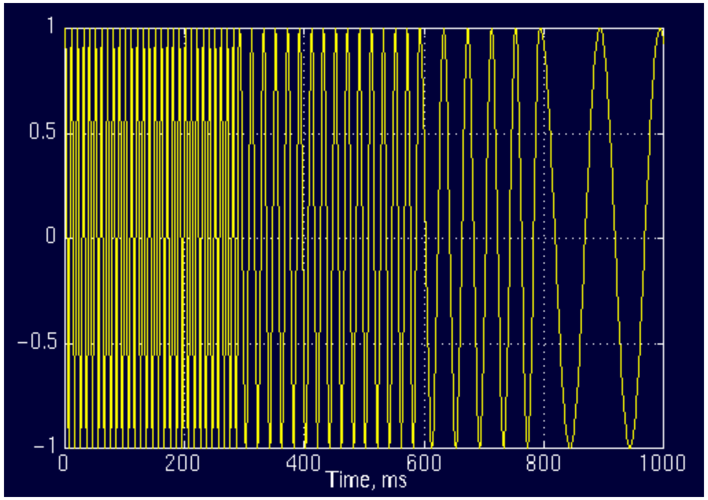
SINAIS ESTACIONÁRIOS *vs.* NÃO-ESTACIONÁRIOS

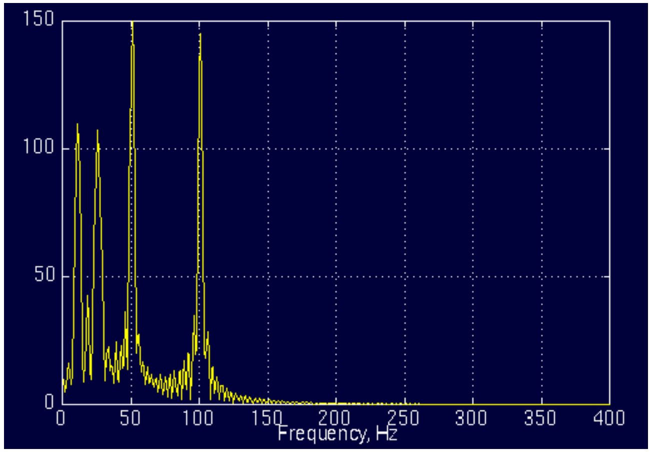
- **Sinais estacionários – o sinal se repete em um curto intervalo de tempo:**





- **Sinais não-estacionários – não há repetição:**

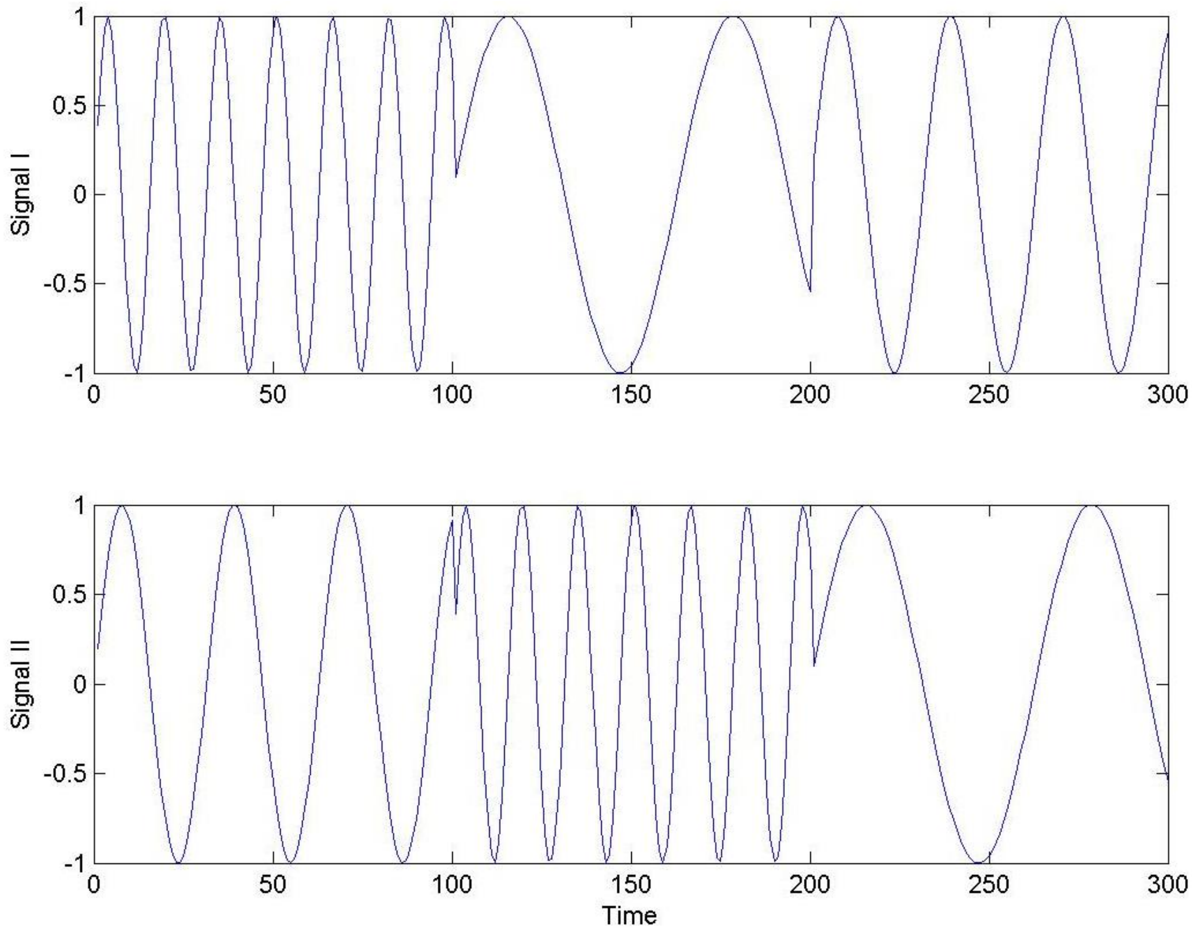




* Fonte das imagens: Polikar, R; **The Wavelet Tutorial**;

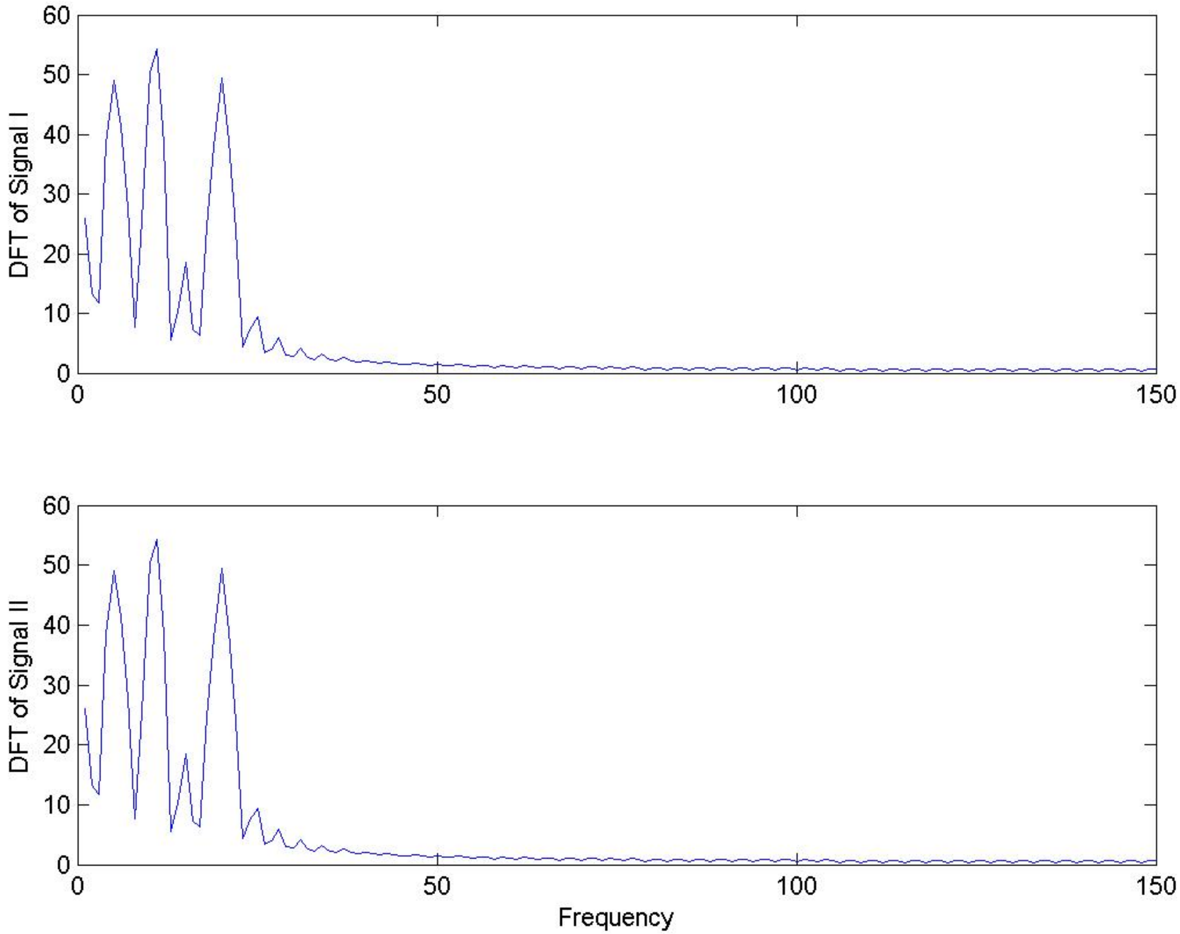
TRANSFORMADA DE FOURIER

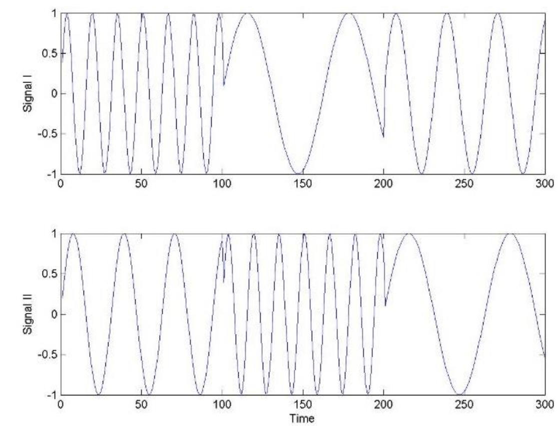
- **Consideremos os seguintes sinais:**



TRANSFORMADA DE FOURIER

- **Espectro de Fourier dos mesmos sinais:**



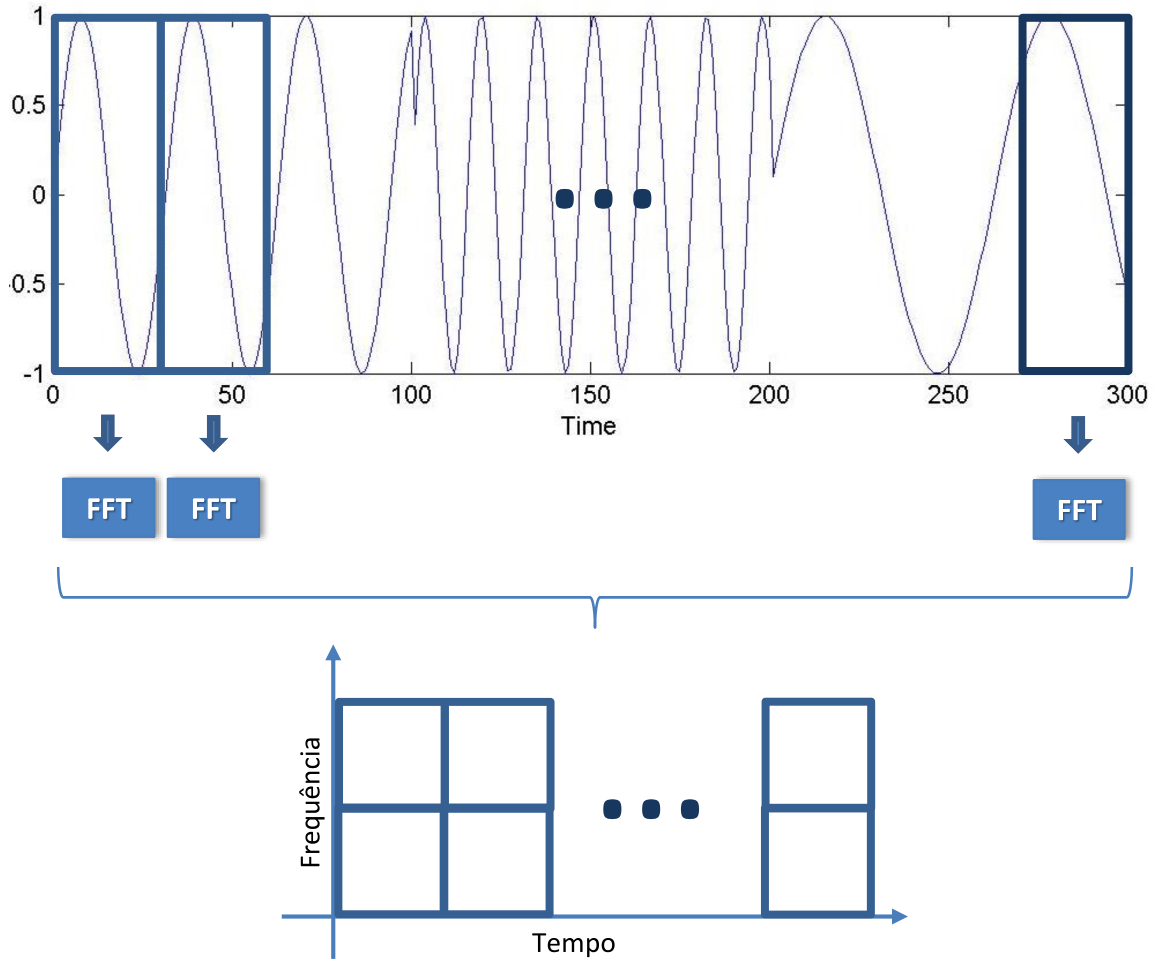


- **Não há informação espacial!**

* Fonte das imagens: **Biomedical Signal and Image Processing**, 2012;

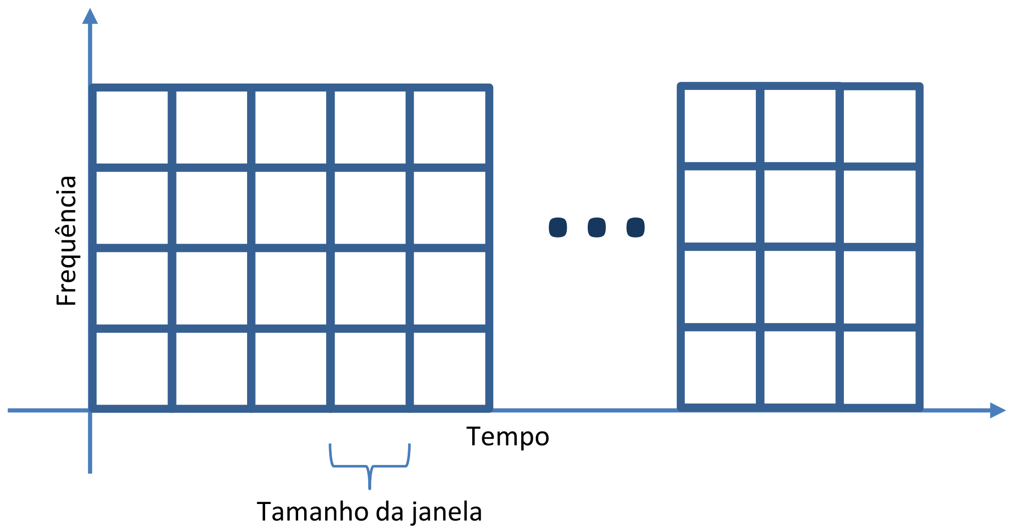
SHORT-TIME FOURIER TRANSFORM

- **FFT não é adequada para sinais não-estacionários;**
- **STFT pode ser uma solução:**



SHORT-TIME FOURIER TRANSFORM

- **STFT pode ser uma solução!**

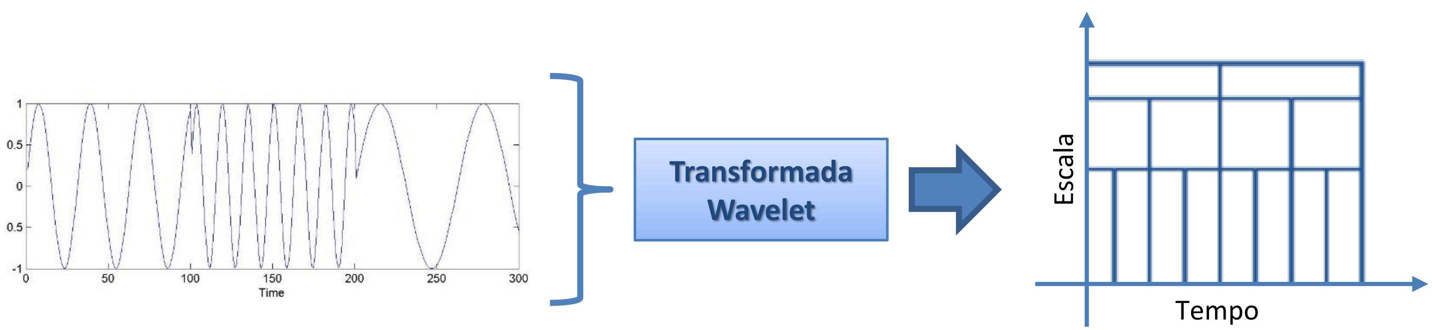


- **Tamanho da janela:**
  - Invariante;
  - Como definir?
    - Janela pequena:
      - **Pouca informação sobre o sinal;**
      - Muito processamento;
    - Janela grande:
      - **Aumenta o erro ao considerar o sinal ser estacionário.**

CENÁRIO IDEAL PARA ANÁLISE DE SINAIS

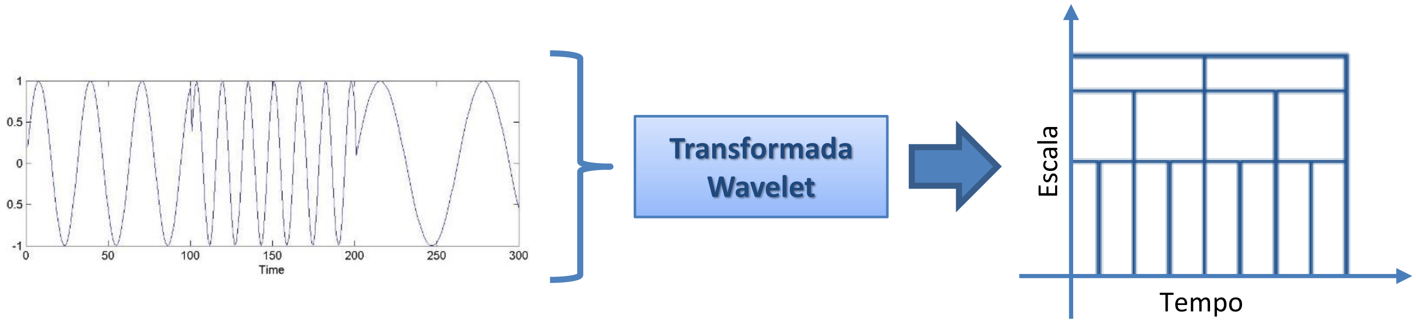
- **Transformada Janelada:**
  - Janelas de tamanho variável:
    - Intervalos maiores → maior precisão sobre baixas frequências;
    - Intervalos menores → maior precisão sobre altas frequências;

- **Transformada Wavelet:**
  - Representação do sinal:
    - Função base não-periódica (*suporte compacto*);
    - Tempo e escala ao mesmo tempo:
      - Escala *vs.* Frequência?



TRANSFORMADA WAVELET

- **Decomposição do sinal no domínio Wavelet:**



- **Matematicamente (*Continuous Wavelet Transform - CWT*):**

$$W_{\Psi,X}(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} x(t)\Psi^{*}\left(\frac{t-b}{a}\right)dt, \qquad a \neq 0.$$

na qual:

- $x(t)$: sinal a ser decomposto;
- $\Psi(t)$: função base com duração limitada no tempo (*Mother Wavelet*);
- a: parâmetro de escala (contração/dilatação);
- b: parâmetro de deslocamento.

TRANSFORMADA WAVELET CONTÍNUA (CWT)

- ***Continuous Wavelet Transform***:

$$W_{\Psi,X}(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} x(t)\Psi^*\left(\frac{t-b}{a}\right) dt, \qquad a \neq 0.$$

  - Com $\frac{t-b}{a}$, Ψ^* é uma função **deslocada** e/ou **escalada** de Ψ;
  - FFT decompõe o sinal em senóides (função periódica):
    - Com CWT a decomposição é com uma Wavelet (*pequena onda*)!

- **Transformada inversa:**

$$x(t) = \frac{C_\Psi^{-1}}{a^2} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} W_{\Psi,X}(a,b)\Psi\left(\frac{t-b}{a}\right) da\, db, \qquad a \neq 0.$$

na qual:

- C_Ψ^{-1}: é uma constante que depende da função Wavelet (Ψ) escolhida;

TRANSFORMADA WAVELET DISCRETA (DWT)

- **A CWT supõe um deslocamento e escala em toda reta real:**
  - Inviável para a maioria dos problemas práticos;
  - Computacionalmente caro;

- **Transformada Wavelet Discreta:**
  - A formação das bases discretizadas foi proposta por Mallat:
    - *Pyramidal Algorithm* ou *Quadrature Mirror Filter* (QMF);
  - Assim como na CWT deve ser escolhido uma Wavelet mãe:
    - Sua discretização será um filtro passa-baixa, $h(n)$;
    - A partir de $h(n)$, um outro filtro é definido:

$$g(n) = h(2N - 1 - n)$$

$h(n)$ → passa-baixa ⇔ $g(n)$ → passa-alta;

TRANSFORMADA WAVELET DISCRETA (DWT)

- $h(n)$ **do QMF é baseado na Wavelet mãe;**

  - Função Wavelet (*Mother Wavelet*):

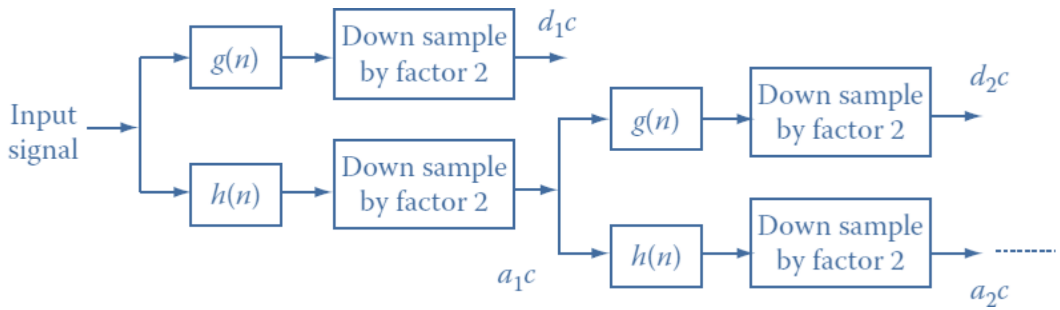
$$\Psi(n) = \sum_{i=0}^{N-1} g(i)\Phi(2n - i)$$

  - Função de Escala (*Scaling function*):

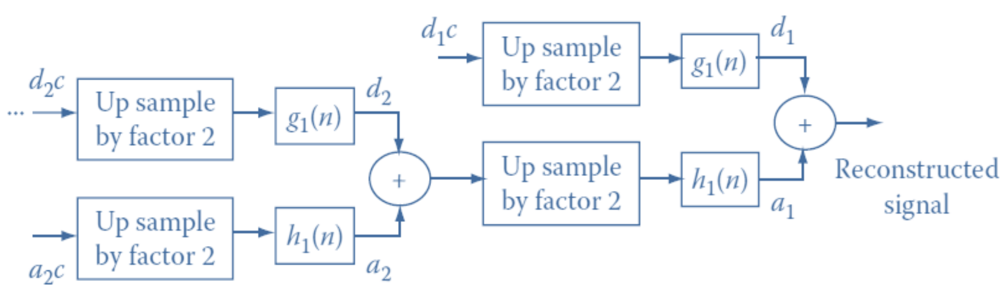
$$\Phi(n) = \sum_{i=0}^{N-1} h(i)\Phi(2n - i)$$

TRANSFORMADA WAVELET DISCRETA (DWT)

- **Sinal 1D:**
  - Decomposição:



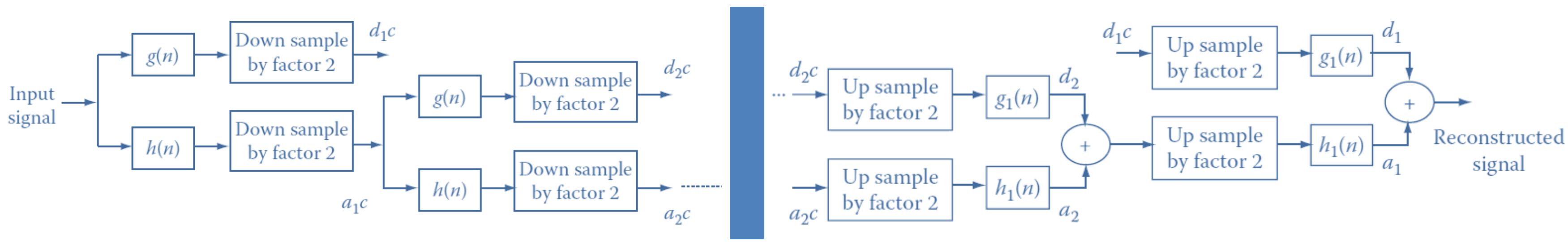
  - Reconstrução:

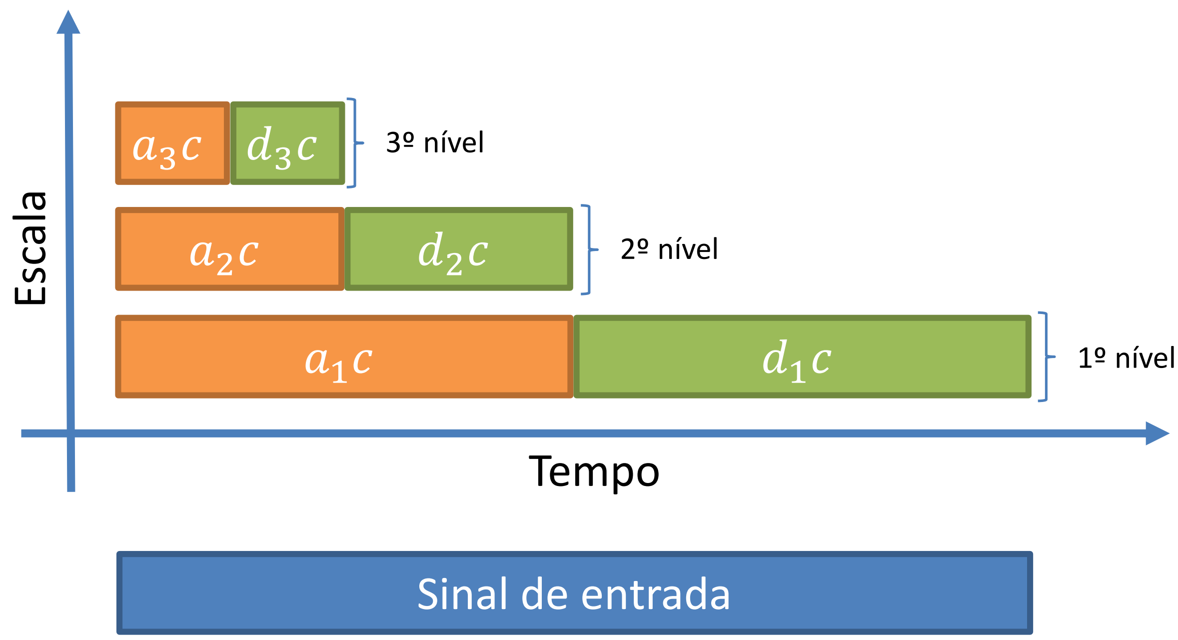


* Fonte das imagens: **Biomedical Signal and Image Processing**, 2012;

TRANSFORMADA WAVELET DISCRETA (DWT)

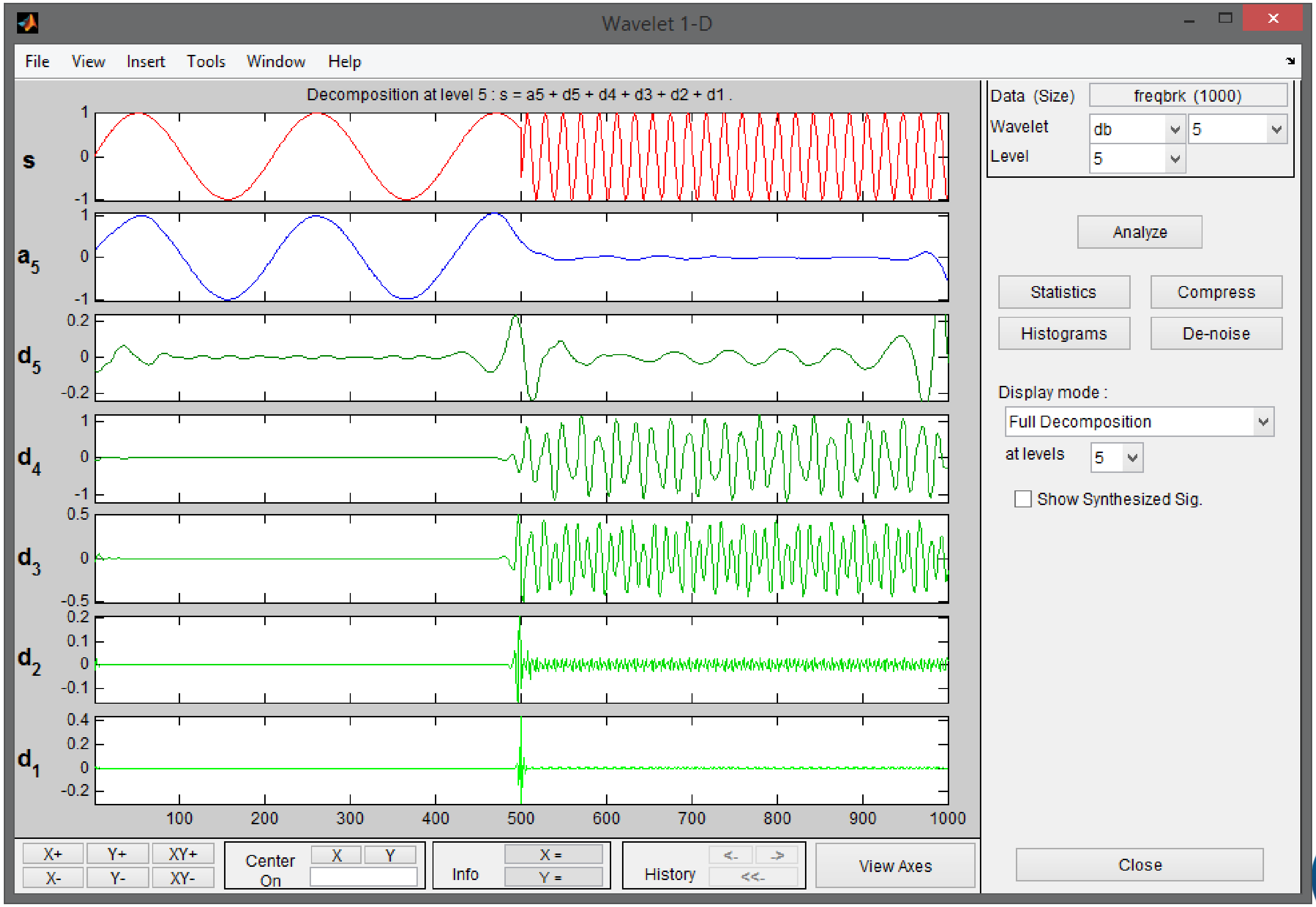
- **Decomposição e Reconstrução:**





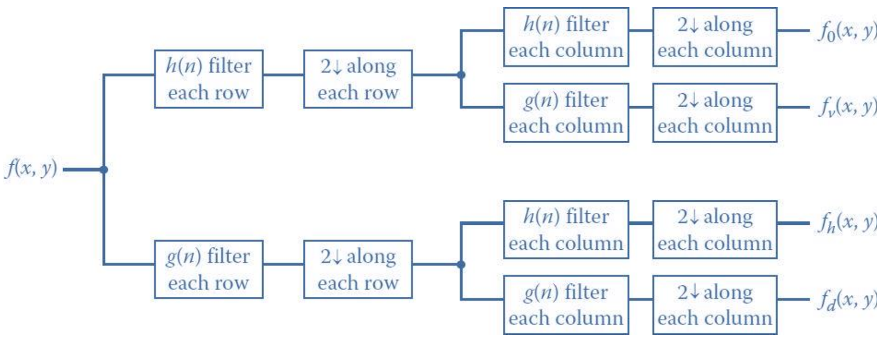
EXEMPLO DE DECOMPOSIÇÃO DE SINAL 1D

- **Utilizando o *Wavelet Toolbox* do MATLAB:**



TRANSFORMADA WAVELET EM IMAGENS

- **Mesma ideia, aplicada 2x:**



Filtragem

L: *Low-pass* – $h(x)$.
H: *High-pass* – $g(x)$.

Coeficientes

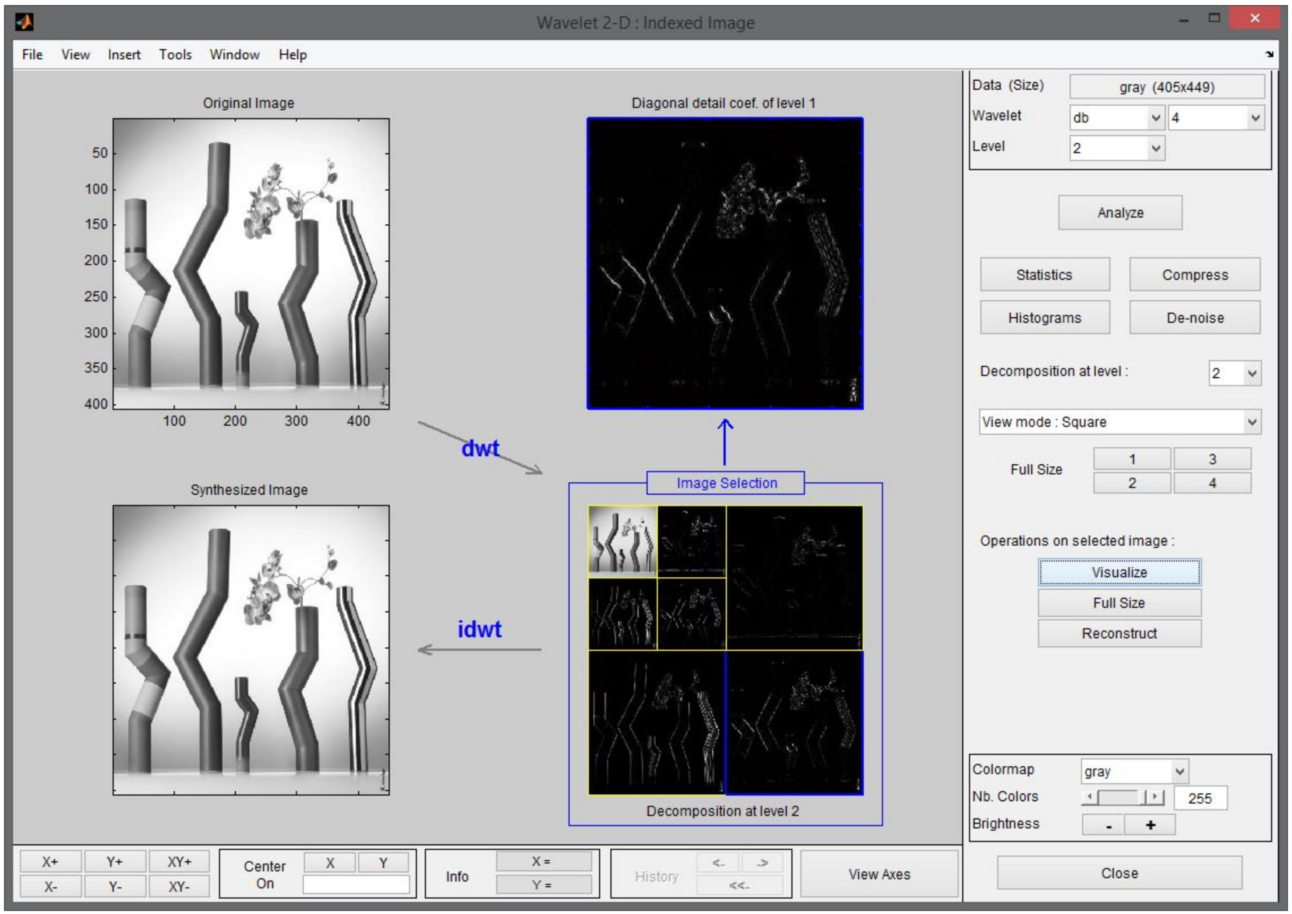
CA: *Aproximação.*
DH: *Detalhes Horizontais.*
DV: *Detalhes Verticais.*
DD: *Detalhes Diagonais.*

- **Exemplo:**

* Fonte da imagem superior: **Biomedical Signal and Image Processing**, 2012;

TRANSFORMADA WAVELET EM IMAGENS

- *Wavelet Toolbox* do MATLAB:



Filtragem

L: *Low-pass* – $h(x)$.
H: *High-pass* – $g(x)$.

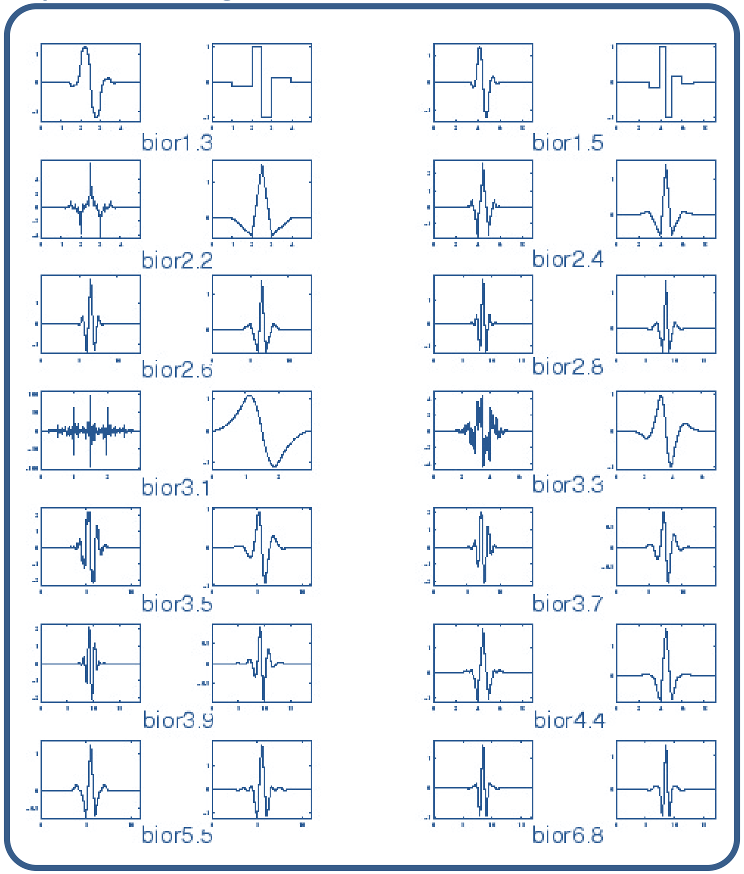
Coeficientes

CA: *Aproximação.*
DH: *Detalhes Horizontais.*
DV: *Detalhes Verticais.*
DD: *Detalhes Diagonais.*

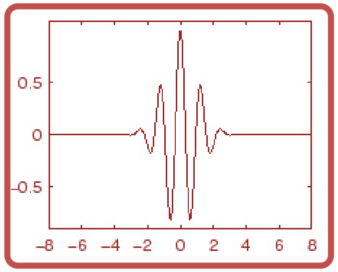
BASES WAVELET

- **Famílias de funções:**

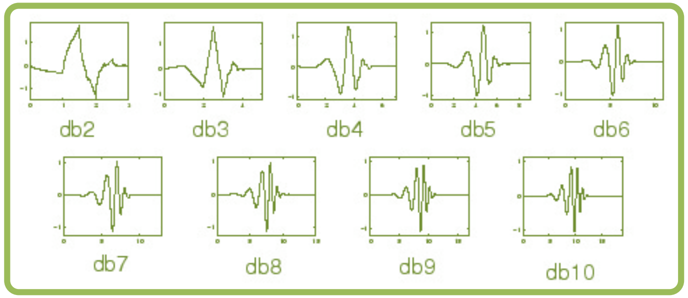
Splines bi-ortogonais:



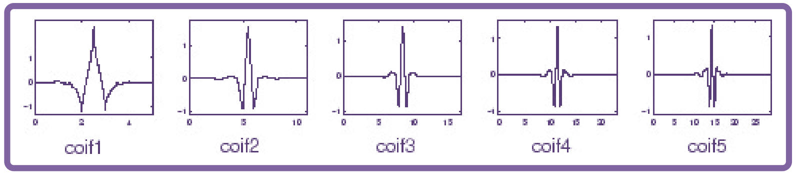
Molet:



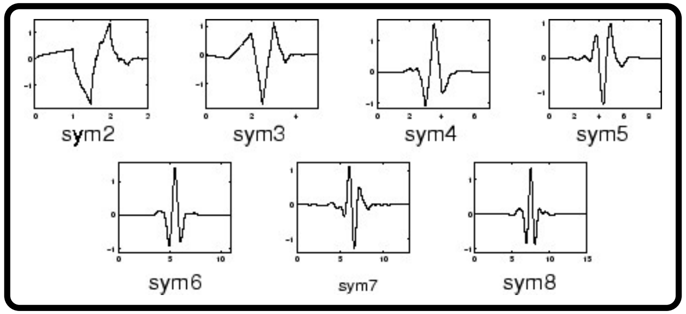
Daubechies:



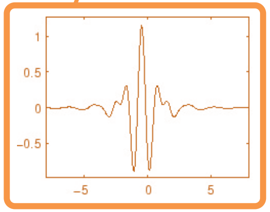
Coiflets:



Symlets:



Meyer:



* Fonte das imagens: **Help do MATLAB R2013b**;

BASES WAVELET

- **Cada Wavelet resulta em uma resposta diferente da transformada;**

- **Propriedades de uma função Wavelet Ψ:**

1. Área zero (condição de admissibilidade):
   - Garante a existência da inversa:

$$\int_{-\infty}^{\infty} \Psi(t)dt = 0 \quad \Rightarrow \quad C_g = \int_{0}^{\infty} \frac{\left|\widehat{\Psi}(f)\right|^2}{f} df < \infty$$

na qual, $\widehat{\Psi}$ é a FFT de $\Psi(t)$:

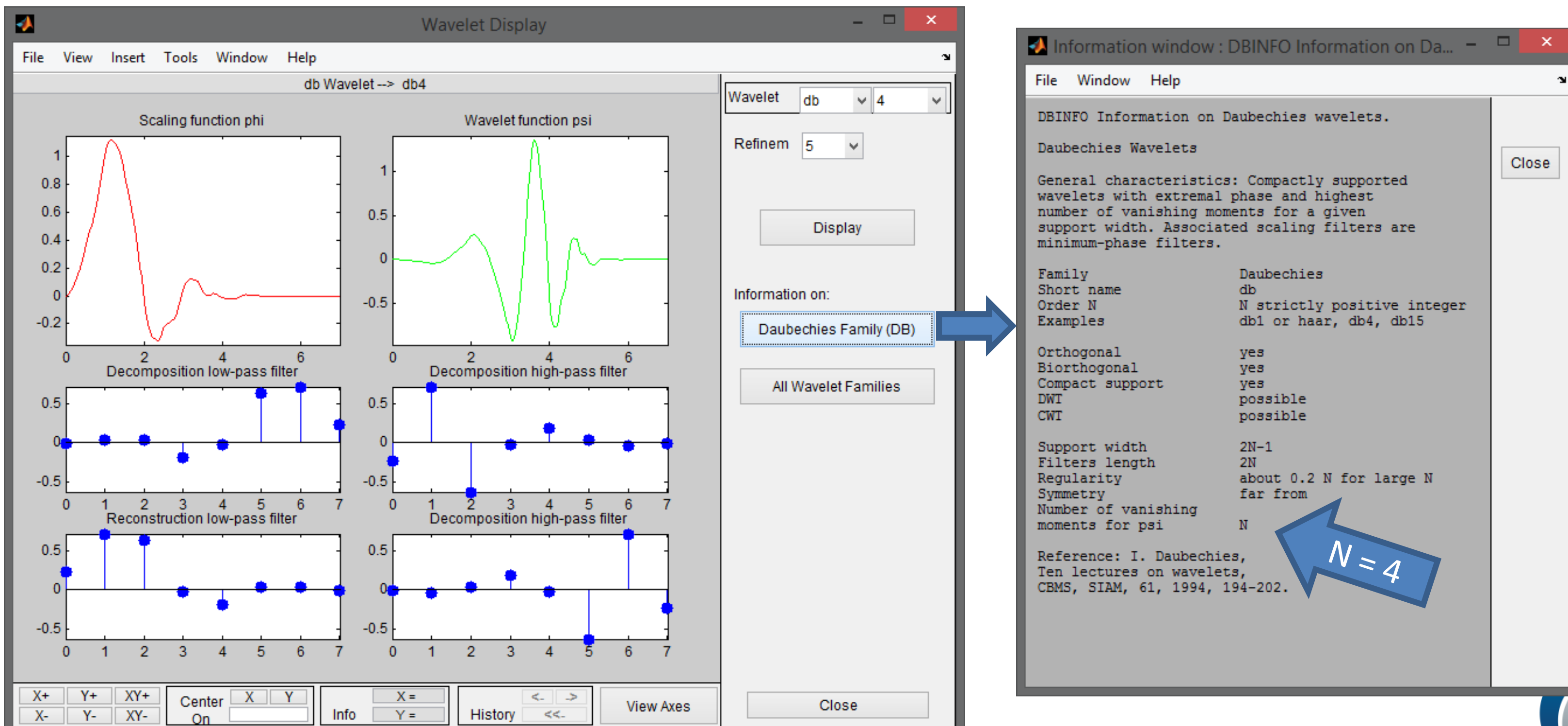
$$\widehat{\Psi} = \int_{-\infty}^{\infty} \Psi(t)e^{-i(2\pi f)t}dt$$

2. Energia finita:

$$\int_{-\infty}^{\infty} |\Psi(t)|^2 dt < \infty$$

BASES WAVELET

- Momentos nulos (*Vanishing Moments*):
    - Maior grau do polinômio interpolador do sinal que pode ser representado:
- Exemplo, Wavelet **Daubechies-4 (db4)**:



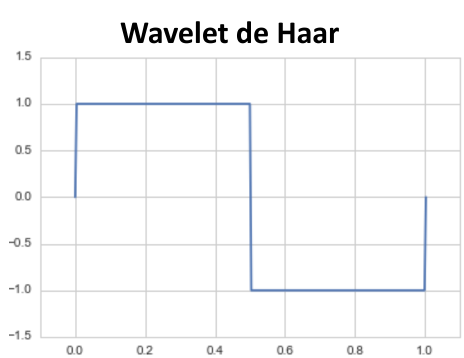
FAQ – *FREQUENTLY ASKED QUESTIONS*

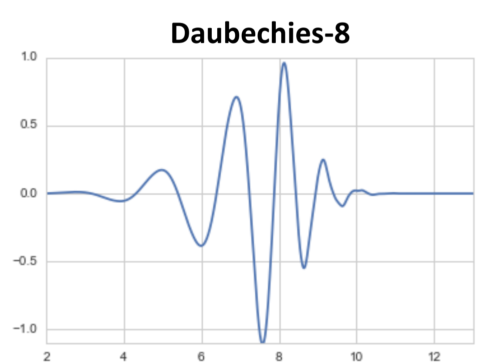
- **Perguntas que valem ouro:**
  - Qual Wavelet usar?
  - Quantos níveis de decomposição?

- **Recomendações:**
  - Se o sinal é complexo → use Wavelet complexa;

  - A forma do seu sinal é importante!
    - Use uma Wavelet que mais se parece com a característica do sinal*;

  - Para sinal com mudanças bruscas, **Haar** é melhor que **Daubechies**;
    - Se quiser um maior refinamento, Daubechies é recomendada;

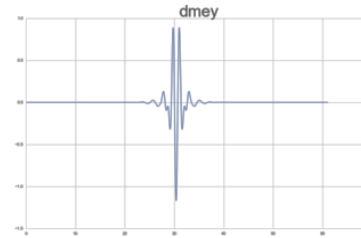


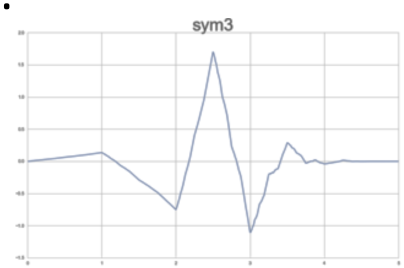


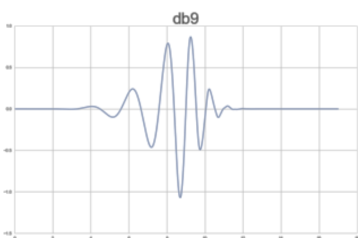
* Wavelets suportadas pela biblioteca PyWavelets: https://goo.gl/hhZcAX

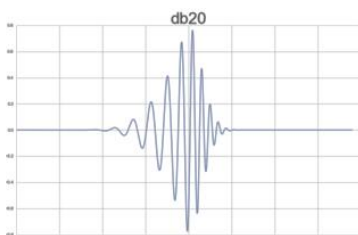
FAQ – *FREQUENTLY ASKED QUESTIONS*

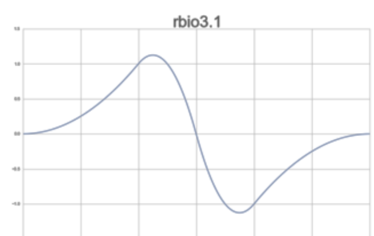
- **Recomendações:**
  - Sinal com variações suaves:
    - Morlet (Mexican-hat): Permite analisar fase e módulo;
    - Ex.: Sistemas geofísicos;
  - Necessidade de sintetizar o sinal e/ou fazer compressão:
    - Wavelets ortogonais, ex: Daubechies ou Meyer;
  - Segmentação de imagens:
    - Wavelets de Gabor (ou Morlet);











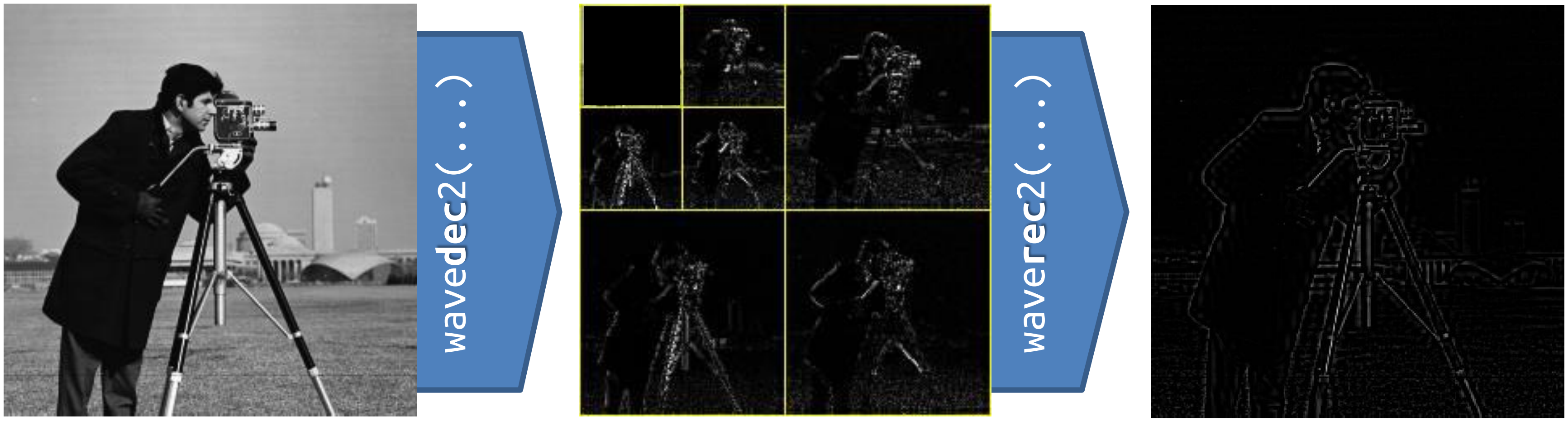
FERRAMENTA *WAVEMENU* (MATLAB)

`>> wavemenu`

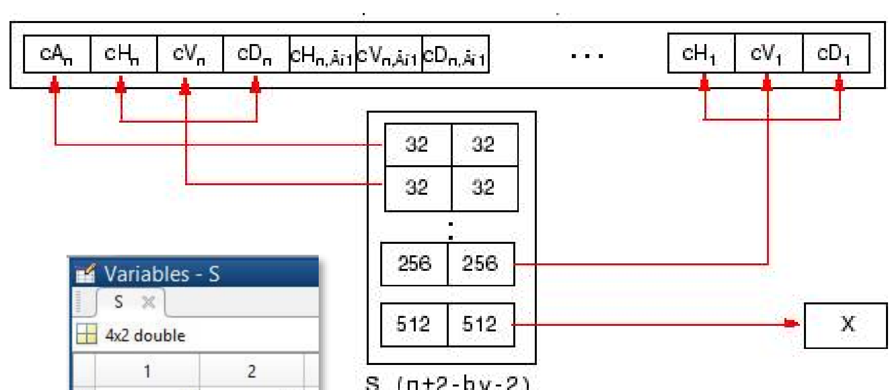
Novas (> R2017b) versões do MATLAB: `waveletanalyzer`

APLICAÇÕES: DETECÇÃO DE ARESTAS

- **Passo-a-passo:**



- **MATLAB:**




```
img = double(imread('Cameraman256.png'));
[C,S] = wavedec2(img, 2, 'db1');
C2 = C;
C2(1:64*64) = 0;
img_rec = waverec2(C2, S, 'db1');
imshow(img_rec,[])
```

Variables - S

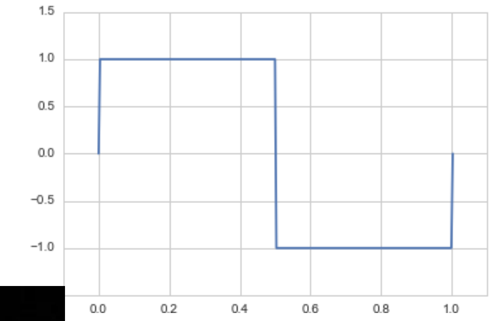
S

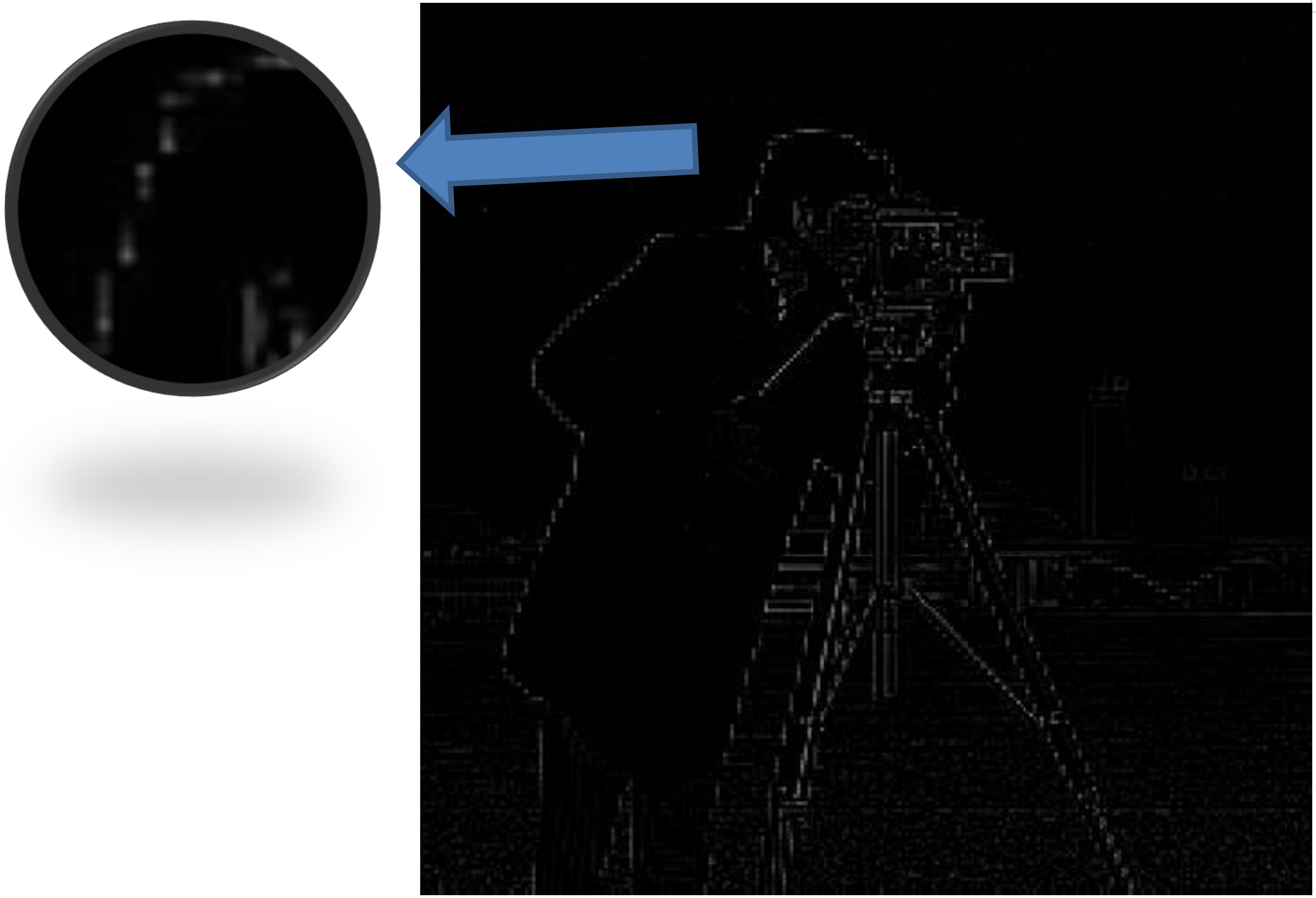
4x2 double

	1	2
1	64	64
2	64	64
3	128	128
4	256	256

APLICAÇÕES: DETECÇÃO DE ARESTAS

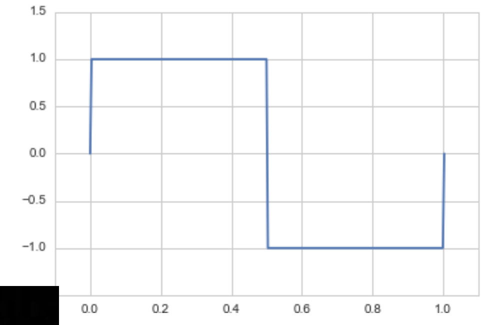
- **Exemplo 1:**
  - Daubechies-1 (Haar);
  - 1 Nível de decomposição;

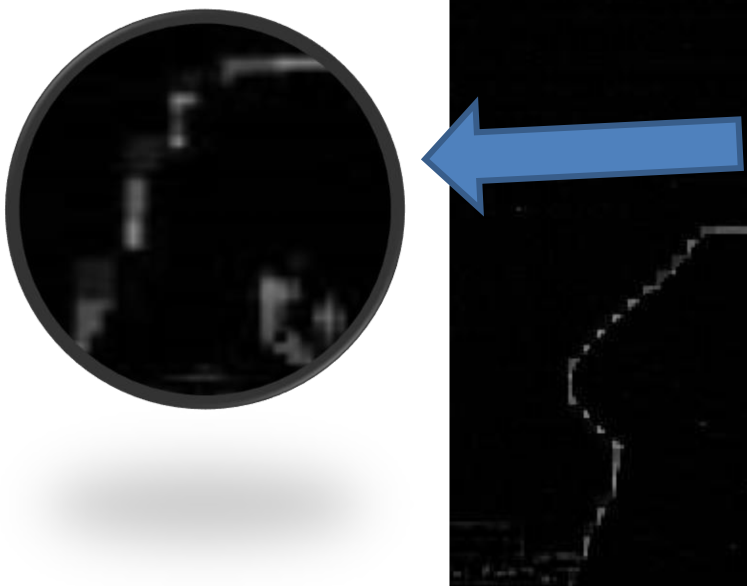




APLICAÇÕES: DETECÇÃO DE ARESTAS

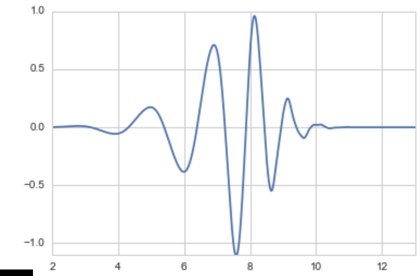
- **Exemplo 2:**
  - Daubechies-1 (Haar);
  - 2 Níveis de decomposição;





APLICAÇÕES: DETECÇÃO DE ARESTAS

- **Exemplo 3:**
  - Daubechies-8;
  - 2 Níveis de decomposição;



APLICAÇÕES: REMOÇÃO DE RUÍDO

- **Motivação:**

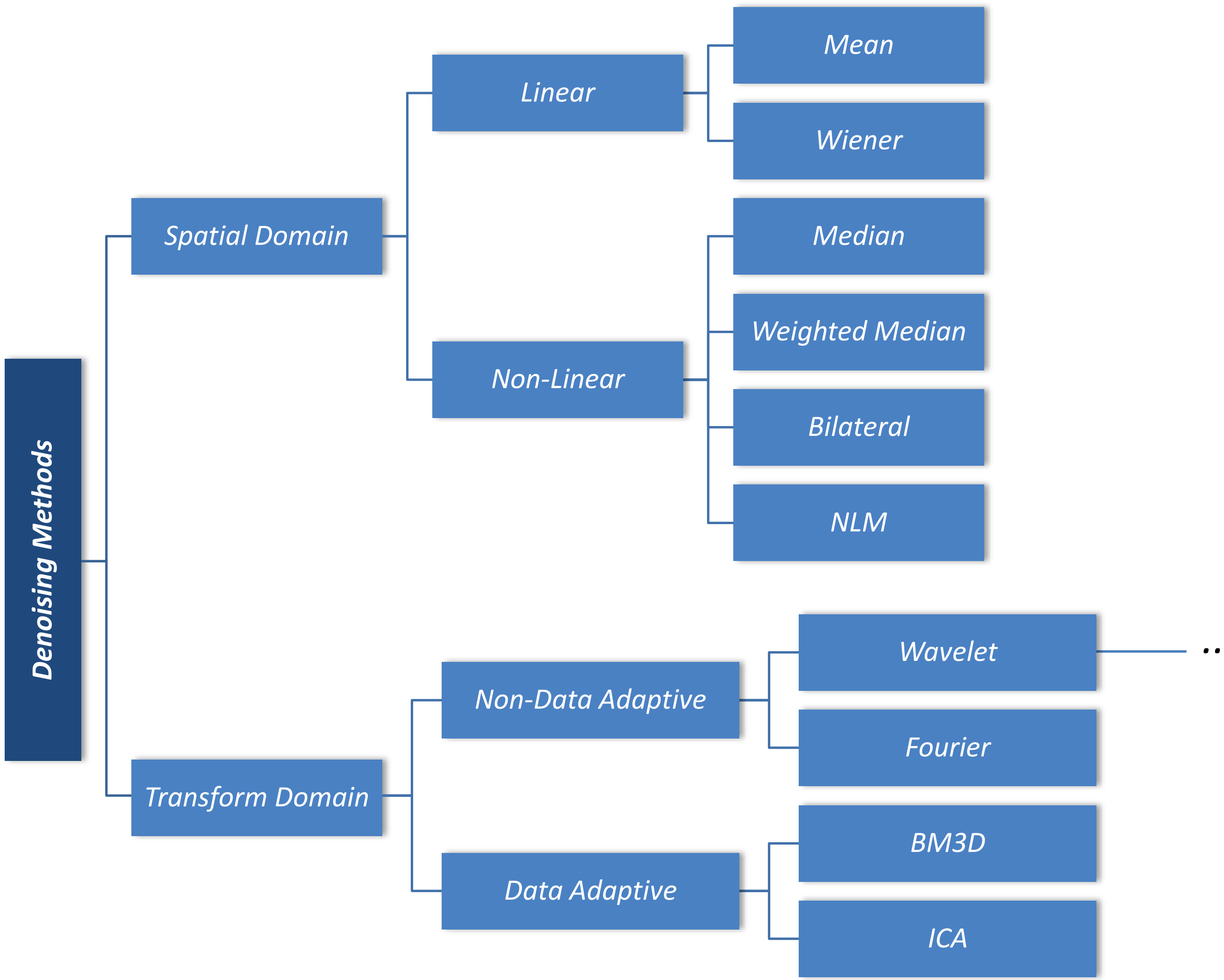
  No domínio Wavelet, para outros níveis de decomposição, o ruído é representado por coeficientes menores do que aqueles do sinal original*.

- **Em geral assume-se que:**
  - O ruído é Independente e Identicamente distribuído pelo sinal:
    - Ex.: Ruído Gaussiano (*Aditive White Gaussian Noise* - AWGN);
  - O ruído existe, mas ainda é possível identificar o sinal;

- **Objetivo:**
  - Remover o ruído mantendo o sinal original causando o mínimo de borramento possível;

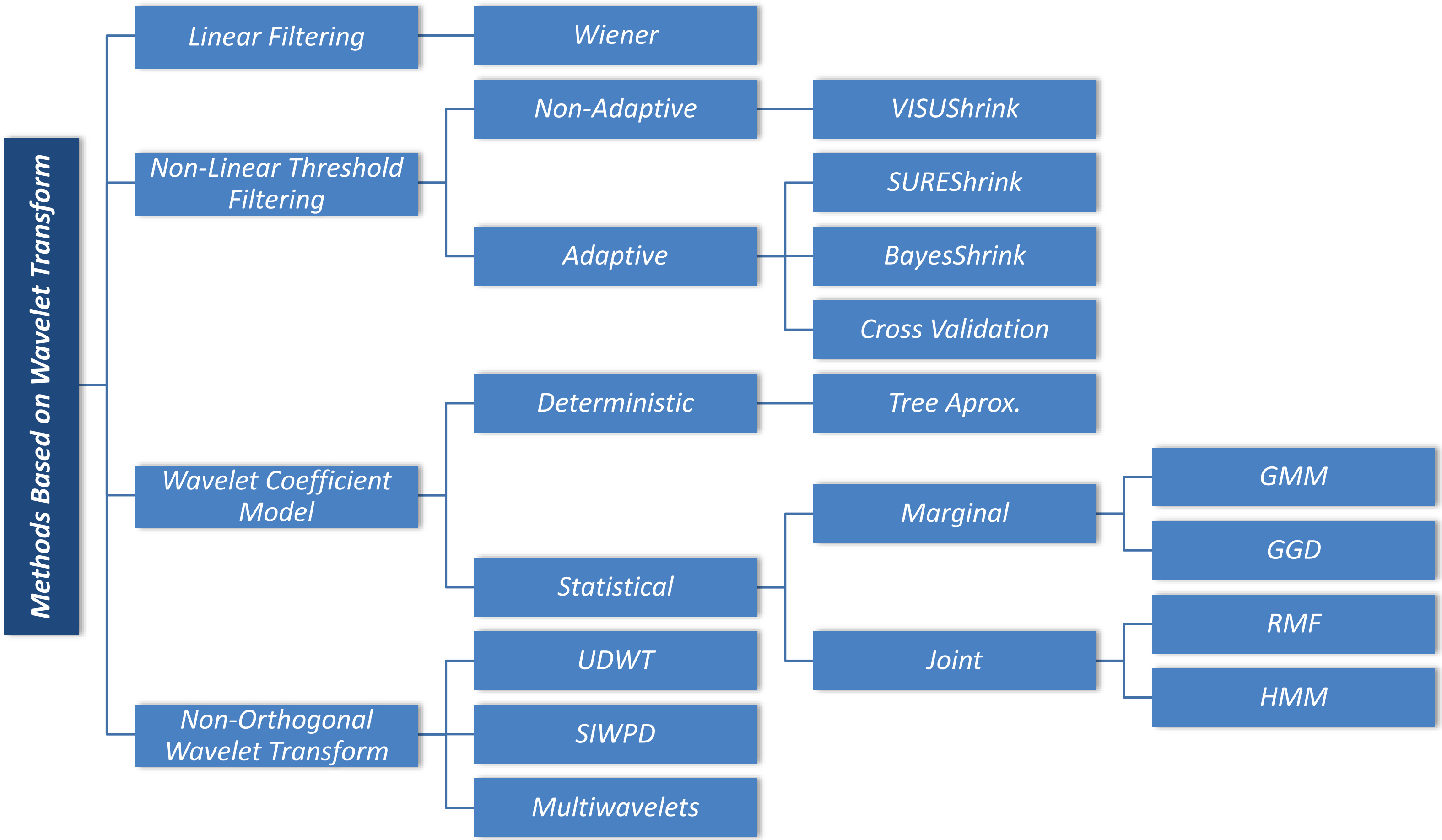
* Donoho, DL; Johnstone, IM; **Ideal Spatial Adaptation via Wavelet Shrinkage**. Stanford Univ., 1992a;

APLICAÇÕES: REMOÇÃO DE RUÍDO



* Graph adapted from: Motwani M., et al.; **Survey of Image Denoising Techniques** in Proc. of GSPx, 2004.

APLICAÇÕES: REMOÇÃO DE RUÍDO



* Graph adapted from: Motwani M., et al.; **Survey of Image Denoising Techniques** in Proc. of GSPx, 2004.

APLICAÇÕES: REMOÇÃO DE RUÍDO

- **Donoho & Johnstone, propuseram a metodologia*:**

Decomposição do Sinal
- Aplicar a Transformada Wavelet (DWT) ao sinal ruidoso (I);

Estimativa do Ruído
- Estimar (E) o valor de *threshold* (λ) para os coeficientes (Y);

Remoção do Ruído
- Realizar o encurtamento (*thresholding* ou *shrinkage*) (D) dos coeficientes abaixo de λ;

Reconstrução do Sinal
- Reconstruir o sinal a partir dos coeficientes alterados ($\breve{Z}$).

$$\breve{Y} = \boldsymbol{DWT}(I)$$

$$\lambda = \boldsymbol{E}(\breve{Y}(i))$$

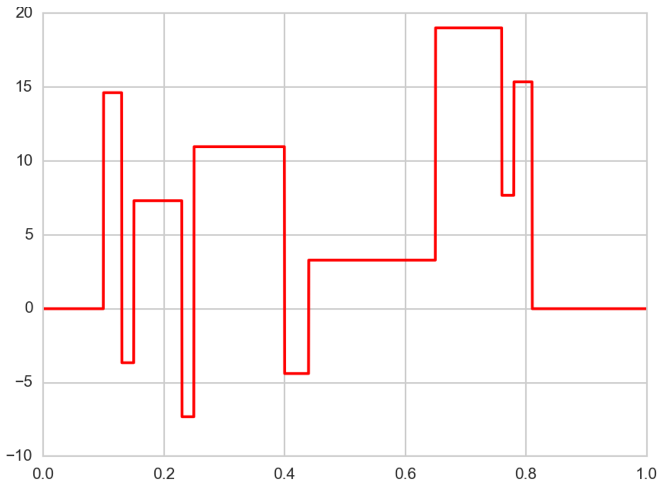
$$\breve{Z} = \boldsymbol{D}(\breve{Y}(i), \lambda)$$

$$\hat{I} = \boldsymbol{IDWT}(\breve{Z})$$

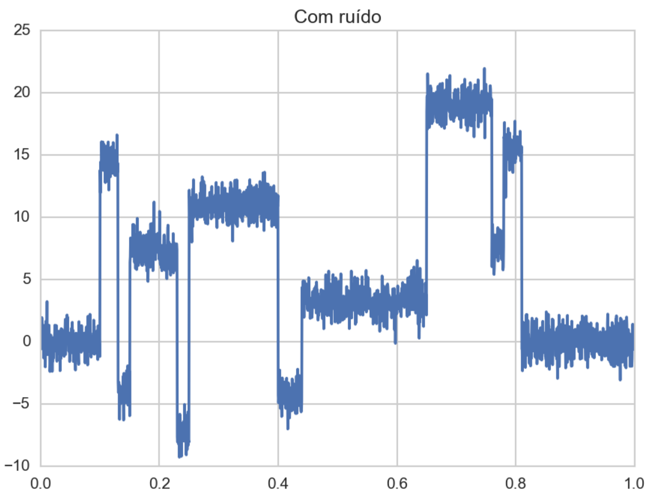
* Donoho, DL; Johnstone, IM; **Ideal Spatial Adaptation via Wavelet Shrinkage**. Stanford Univ., 1992a;

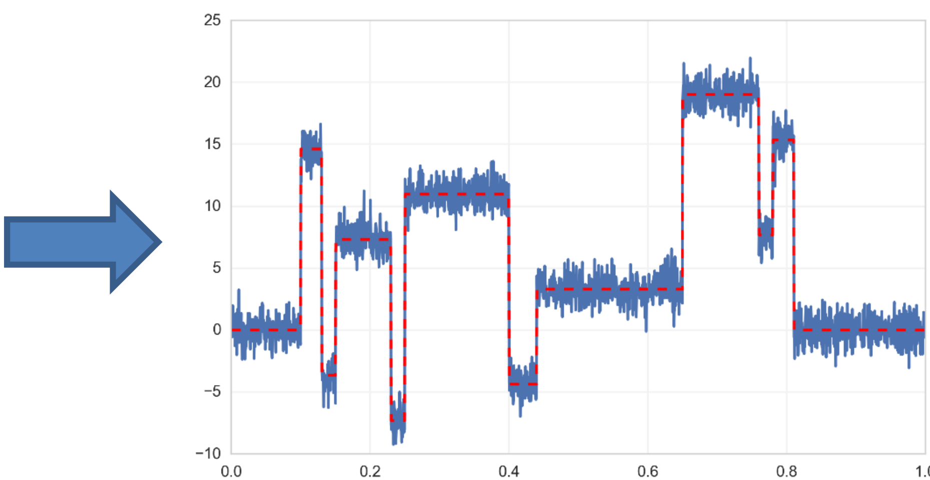
APLICAÇÕES: REMOÇÃO DE RUÍDO

- **Sinal sintético, livre de ruído:**



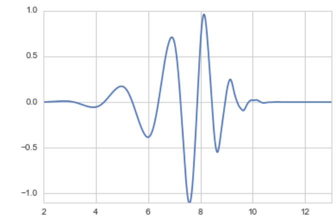
- **Corrompido por ruído Gaussiano:**

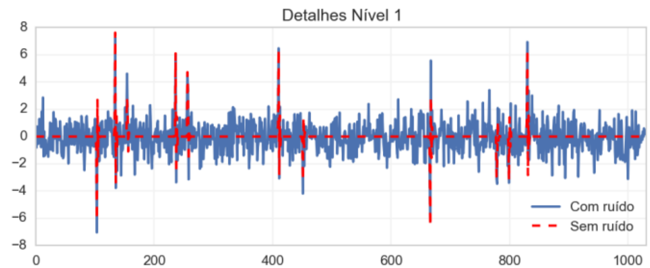


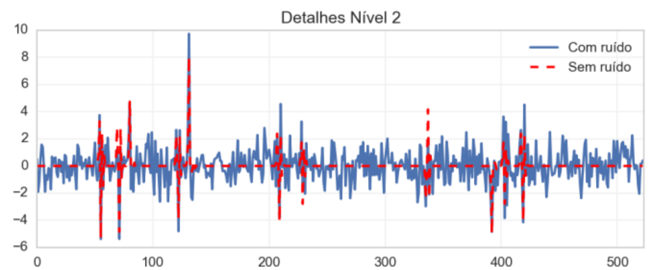


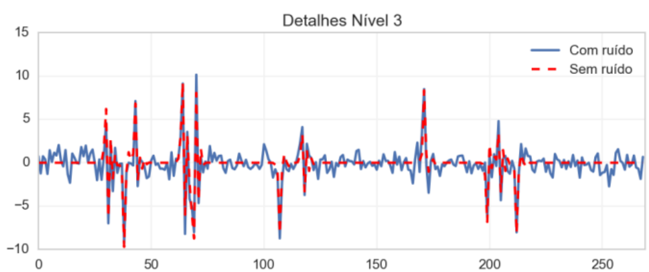
APLICAÇÕES: REMOÇÃO DE RUÍDO

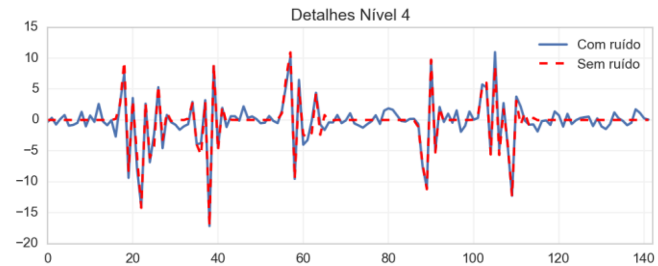
- **Composição com db-8:**

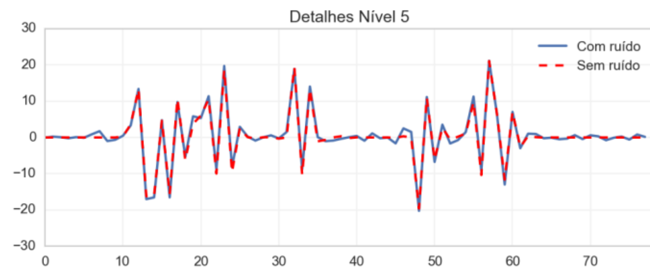


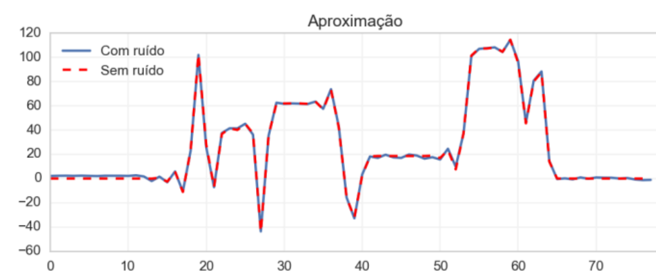






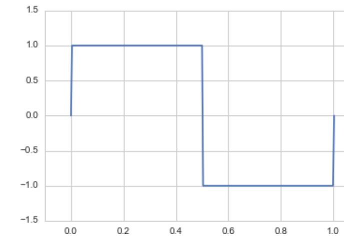


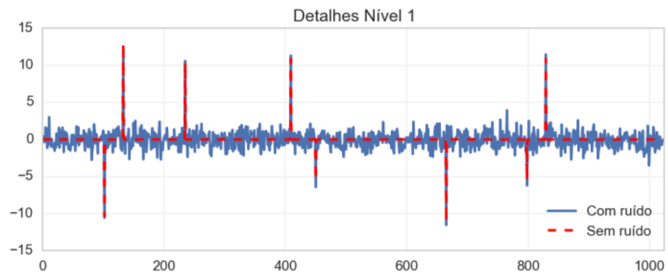


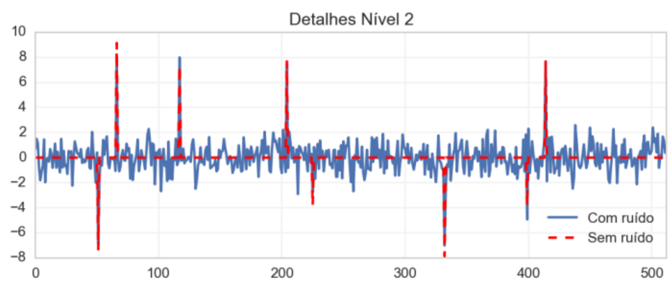


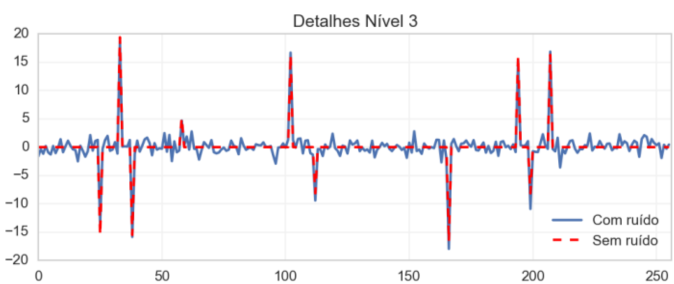
APLICAÇÕES: REMOÇÃO DE RUÍDO

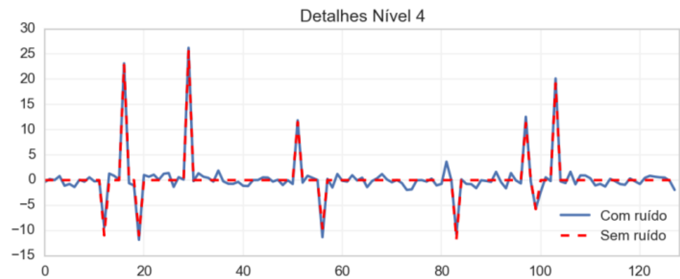
- **Decomposição com Haar (db1):**

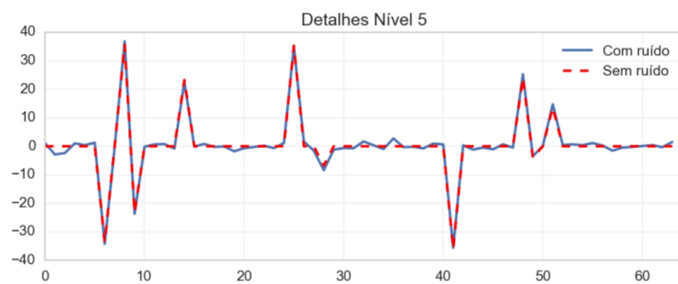


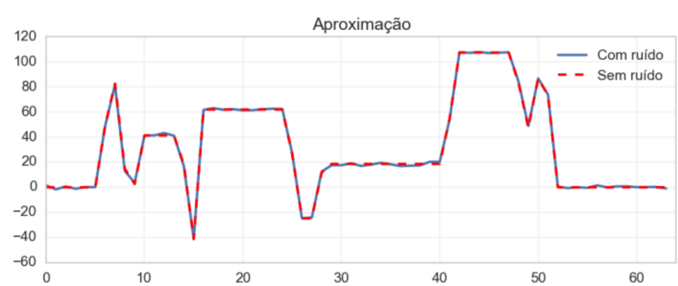












APLICAÇÕES: REMOÇÃO DE RUÍDO

- **Estimativa do valor de threshold (λ):**
  - Visu-shrink*:

$$\lambda_{vs} = \sqrt{2\ln(n)}\sigma$$

σ: desvio padrão do ruído;
n: tamanho do sinal;

  - Bayes-shrink**:
    - Parte do modelo tradicional de observação: $Y = X + N \Rightarrow \sigma_Y^2 = \sigma_X^2 + \sigma^2$:

$$\lambda_{bs} = \frac{\hat{\sigma}^2}{\hat{\sigma}_x}$$

na qual:

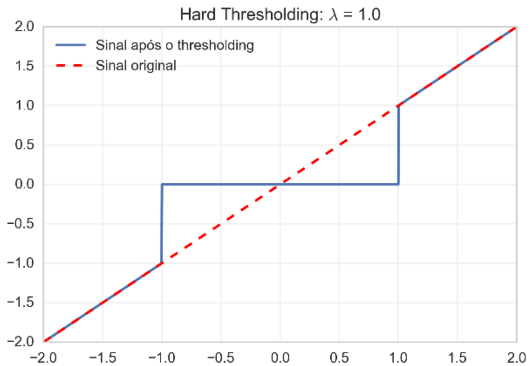
$$\hat{\sigma}_X^2 = \max(\hat{\sigma}_Y^2 - \hat{\sigma}^2, 0) \qquad \hat{\sigma}_Y^2 = \frac{1}{N}\sum_{n=1}^{N} y_n^2$$

e $\hat{\sigma}^2$ estimado pelo *Median Absolute Deviation* (MAD).

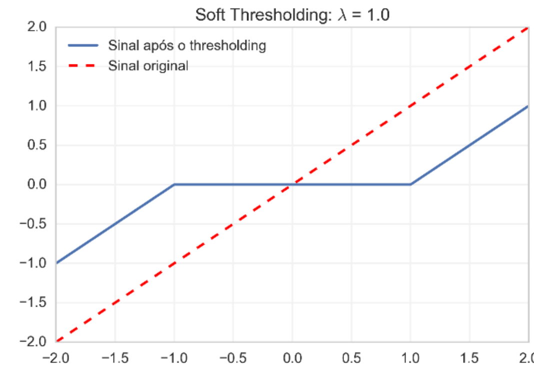
* Donoho, DL; Johnstone IM; **Ideal Spatial Adaptation via Wavelet Shrinkage**. Biometrika, v81(3), 1994;
** Chang, S.; Yu, B.; Vetterli, M. **Adaptive wavelet thresholding for image denoising and compression**. Image Proc., IEEE Trans. on, v. 9, n. 9, p. 1532–1546, Sep 2000.

APLICAÇÕES: REMOÇÃO DE RUÍDO

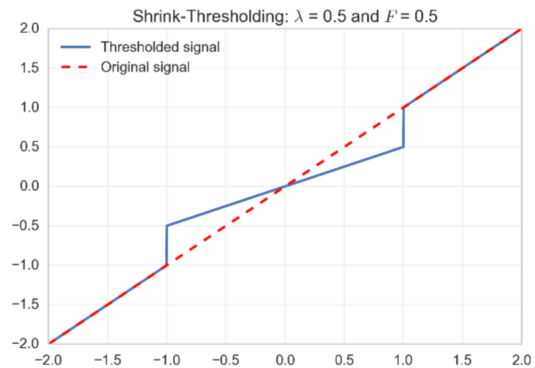
- **Métodos de Thresholding (D):**



$$D_h(Y,\lambda) = \begin{cases} Y, & se\ |Y| > \lambda \\ 0, & c.c. \end{cases}$$



$$D_s(Y,\lambda) = \begin{cases} sgn(Y)(|Y| - \lambda), & se\ |Y| > \lambda \\ 0, & c.c. \end{cases}$$

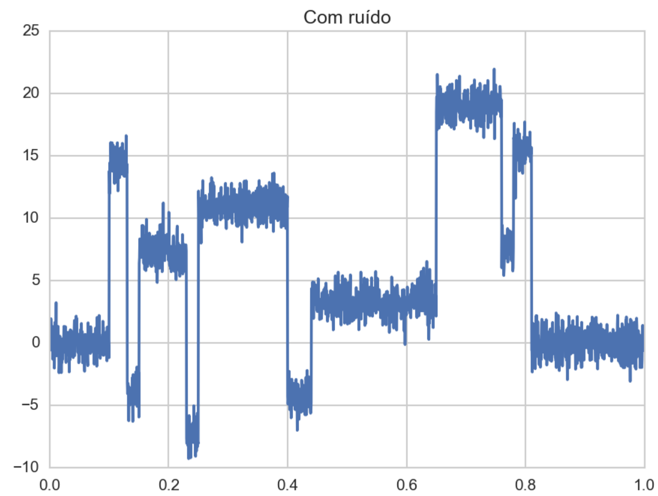


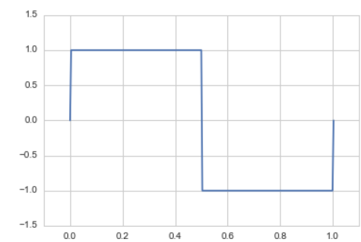
$$D_{st}(Y,\lambda,F) = \begin{cases} Y, & se\ |Y| > \lambda \\ Y \cdot F, & c.c. \end{cases}$$

* Jansen, M; **Noise Reduction by Wavelet Thresholding**, 2001;
** Oliveira, HCR et. al; **Use of wavelet multiresolution analysis to reduce radiation dose in digital mammography**, Proc. of 28th IEEE Int. Symp. on Comp. Based Medical-Systems (CBMS2015), 2015.

APLICAÇÕES: REMOÇÃO DE RUÍDO

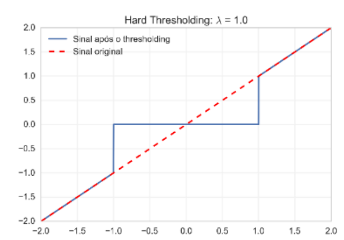
- **Voltando ao exemplo:**





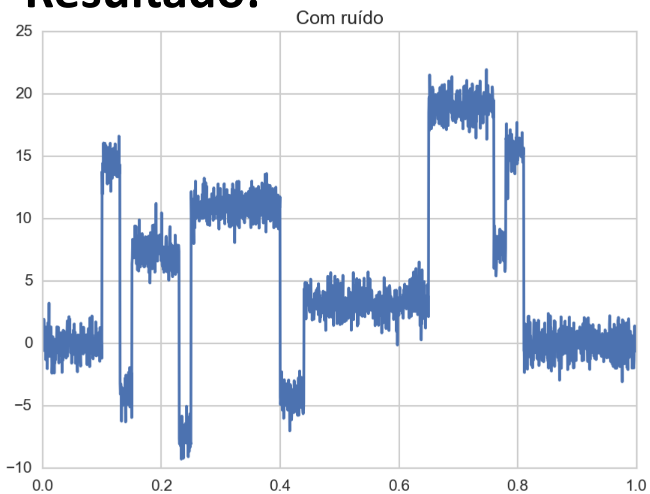
$$\lambda_{vs} = \sqrt{2\ln(n)}\sigma$$

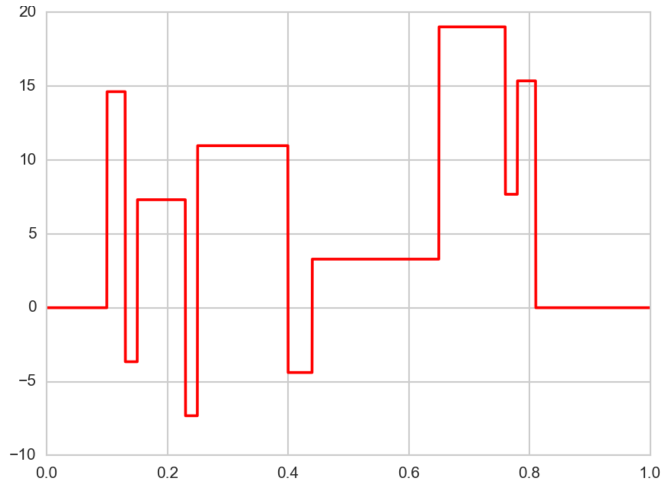
- Wavelet definida: **Haar (db1)**;
- Níveis de decomposição: **5**;
- Estimador de *threshold* (λ): **VisuShrink**;
- Função de *thresholding* (D): **Hard-thresholding**;

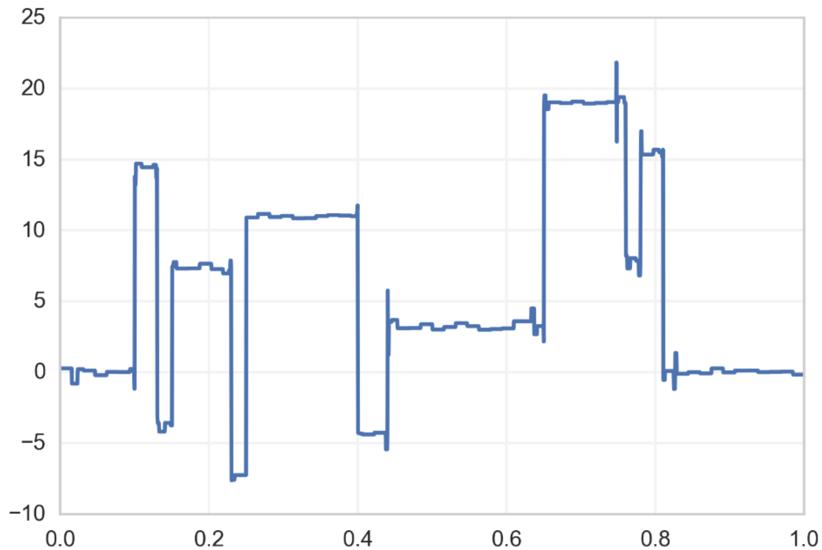


APLICAÇÕES: REMOÇÃO DE RUÍDO

- **Resultado:**







	SNR (dB)
Antes	19.08
Depois	**30.28**

APLICAÇÕES: REMOÇÃO DE RUÍDO

- **Exemplo 2:**
  - Imagem corrompida por ruído Poisson:
    - Dependente do sinal;

  - Metodologia:
    - Wavelet: **Daubechies-8**;
    - Níveis de decomposição: **5**;
    - Estimador de *threshold* (λ): **MiniMax** (MATLAB);
    - Funções de *thresholding* (D): ***Hard***, ***Soft***, ***Shrink***;

APLICAÇÕES: REMOÇÃO DE RUÍDO

- **Resultados:**

Hard-thresholding

Soft-thresholding

Shrink-thresholding

	SNR (dB)
Antes	21.82
Hard	24.68
Soft	21.11
Shrink	**25.41**

CONCLUSÕES

- **Transformada Wavelet é a solução para os meus problemas?**
- **Sempre vou usar Wavelet ao invés de Fourier?**
- **Perguntas de que valem ouro:**
  - Qual a melhor wavelet?
  - Qual o melhor nível de decomposição?

- **Processamento/Análise de sinais com Wavelet é um problema aberto!**
  - Várias funções bases e metodologias de processamento são propostas;

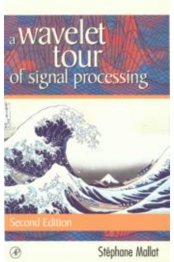
- **Transformadas não apresentadas:**
  - Transformada Não-decimada;
  - *Wavelet Packets*;
  - Curvelets;
  - 3D Wavelet

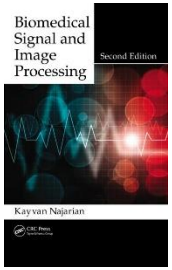
  ...

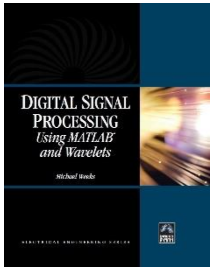
REFERÊNCIAS

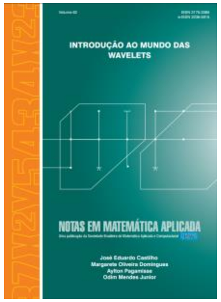
- **Tutorial:**
  - **The Wavelet Tutorial** (Robi Polikar);

- **Livros:**
  - **Processamento Digital de Imagens**, 3ª Ed., 2010 (R. Gonzales; Woods);
  - **A Wavelet Tour of Signal Processing**, 2ª Ed., 1999 (S. Mallat);
  - **Biomedical Signal and Image Processing**, 2ª Ed., 2012 (K. Najarian; R. Splinter);
  - **Digital Signal Processing Using MATLAB and Wavelets**, 1ª Ed., 2007 (M. Weeks);
  - **Introdução ao Mundo das Wavelets**, Notas em Mat. Aplicada, (Castilho; Domingues; Pagamisse; Mendes Jr.), vol.62, 2012.









Helder C. R. de Oliveira

heldercro@usp.br

Departamento de Engenharia Elétrica

Universidade de São Paulo - USP – São Carlos

EESC · USP
Escola de Engenharia de São Carlos
Universidade de São Paulo

Laboratório de Visão Computacional
LAVI
EESC-USP