

Utilização das Wavelets de Daubechies no estudo de funções analíticas.

Alan Pedro Vasconcelos Martins¹, Paulo César Linhares da Silva².

Resumo: A wavelet é uma ferramenta matemática capaz de gerar, decompor e analisar funções. As suas aplicações estão nas mais diversas áreas do conhecimento científico e tecnológico. Em particular, as wavelets de Daubechies possuem suporte compacto e a possibilidade de operacionalização via métodos numéricos clássicos. As funções analíticas possuem a característica de poder ser deriváveis quantas vezes haja necessidade. Neste trabalho é feita uma análise numérica de funções analíticas via wavelets de Daubechies, fazendo uso da função geradora de momentos. Para tal, são utilizados a linguagem de programação Python e conhecimentos prévios de análise matemática, análise funcional e álgebra linear.

Palavras-chave: *Wavelets de Daubechies; Funções Analíticas; Linguagem Python.*

1. INTRODUÇÃO

No mundo atual, há diversas tecnologias que seriam impossíveis de existir sem a análise de sinais. Por exemplo, qualquer tecnologia que envolva processamento de áudio ou de imagem, como as televisões, computadores e celulares. Sendo um sinal um conjunto de dados ou informação (LATHI, 2008, p. 75), a análise de sinais consiste em representá-los por funções matemáticas, permitindo analisá-los ou modificá-los. A partir disso, é possível criar, por exemplo, sistemas de controle para que as hélices de uma turbina eólica se ajustem para a posição mais adequada diante das condições do vento, ou fazer análises sísmicas precisas, dentre muitas outras aplicações. Diante desse contexto, dada a grande diversidade de aplicações existentes para a análise de sinais, os sinais analisados em cada aplicação têm sua própria natureza e comportamento, razão pela qual não existe um método único que permita representar qualquer sinal por meio de funções matemáticas. É por essa razão que, ao longo dos anos, diversos métodos para análise de sinais foram desenvolvidos. Um deles é a transformada wavelet.

Nascida antes da wavelet, a transformada de Fourier é uma ferramenta matemática que representa uma função como uma combinação de funções senoidais e/ou cossenoidais. Entretanto, o método de Fourier enfrenta um problema ao analisar sinais que apresentam mudanças abruptas ou descontinuidades em seu comportamento, além de não conseguir fornecer informações precisas em cada instante de tempo, problema resultante do fato de as funções seno e cosseno oscilarem infinitamente no tempo. A transformada janelada de Fourier minimiza tal problema, mas torna-se um procedimento caro, conforme dito em Farge (1992). Portanto, para uma análise mais precisa desse tipo de sinal, convém usar uma outra classe de função que seja bem localizada no tempo. A transformada wavelet surge, então, com a possibilidade de lidar com a deficiência da transformada de Fourier. Isso porque as wavelets são funções bem definidas no tempo. Ou seja, elas se tratam de uma função que tem começo e fim bem definido, razão pela qual receberam o nome de wavelet, que pode ser traduzido para o português como ondeleta, que por sua vez significa onda pequena. Entretanto, esse não é, ainda, o conceito matemático de uma wavelet. Tal conceito será apresentado em um momento inicial deste trabalho, bem como o procedimento utilizado funções usando essa transformada. Uma característica muito importante das wavelets é que a definição delas é feita com base em propriedades matemáticas que elas precisam atender, sem definir analiticamente a forma da função de base usada na transformada. Por isso, existem diversas famílias diferentes de wavelets, e para a maioria delas, não existe uma expressão analítica que as descreva. Esse é o caso, por exemplo, de quase todas as Daubechies, incluindo as utilizadas neste trabalho (sendo a Daubechies 1, que não é utilizada neste trabalho, a única que tem expressão analítica). Por isso, diversos pesquisadores buscam formas otimizadas de descrever wavelets numericamente de modo prático, a fim de usufruir dos benefícios mencionados dessa transformada para modelagem de sinais, quando comparada com outros métodos numéricos mais conhecidos. Diante desse contexto, o método demonstrado em Silva (2019) discretiza uma wavelet como uma combinação de funções de base, utilizando um método de Galerkin adaptado para determinar numericamente essas funções de base e, consequentemente, a wavelet, e em seguida utiliza uma função geradora de momentos para representar qualquer polinômio do tipo x^k através de uma combinação de wavelets.

No presente trabalho, realizou-se um estudo a respeito da utilização das wavelets para representar funções analíticas (ou seja, funções que podem ser expandidas localmente em séries de Taylor), a partir da metodologia proposta em Silva (2020), para então fornecer códigos computacionais que permitam realizar esse procedimento e avaliar os resultados. Em seguida, será feita uma análise dos critérios necessários para a eficácia dessa metodologia, e será resolvido um pequeno problema ilustrativo para demonstrar a natureza dos problemas que essa metodologia pode ajudar a resolver. Esse estudo é feito expandindo funções em polinômios de Taylor para aplicar

a transformada wavelet apenas a funções polinomiais. O motivo do uso do polinômio de Taylor como ferramenta auxiliar é que, em Silva, (2019), cujo estudo também se baseia nas wavelets Daubechies, utilizou-se uma função geradora de momentos para representar funções polinomiais do tipo x^k por meio de wavelets. A grande vantagem dessa estratégia é que é muito difícil, se não impossível, criar um método genérico para representar qualquer função por meio de wavelets. Porém, representar apenas funções polinomiais se mostra uma tarefa viável computacionalmente, graças à metodologia utilizada neste trabalho.

Na Seção 3 deste trabalho, será feita uma elucidação sobre a metodologia utilizada para representar numericamente uma wavelet. É importante salientar que este trabalho não pretende se aprofundar nos detalhes matemáticos do desenvolvimento dos teoremas e definições necessárias para essa modelagem numérica. Afinal, o objetivo do trabalho é apresentar, de forma prática, a aplicação numérica de uma técnica já existente para representar funções analíticas via uma combinação de wavelets, bem como avaliar os critérios para a qualidade de tal técnica. Apesar disso, este trabalho ainda fará uma abordagem matemática sobre toda metodologia utilizada, priorizando uma análise qualitativa e clara sobre a mesma, sempre referenciando os trabalhos que apresentam os detalhes matemáticos necessários para demonstrar alguns resultados utilizados.

Após a abordagem introdutória sobre a metodologia e a aplicação desta para representar funções analíticas como uma combinação de wavelets, será feita uma análise de algumas funções analíticas para constatar as condições para a efetividade do método. Por fim, serão apresentados alguns problemas ilustrativos para denotar a aplicabilidade dos estudos realizados neste trabalho. É importante ressaltar que, para aplicar esse método numericamente, é preciso truncar a série de Taylor em algum ponto, evidenciando que haverá um erro tão maior quanto menor for o intervalo de truncamento da série de Taylor. Para medir a eficiência do método, foram avaliados os gráficos de algumas funções escolhidas para análise após ser realizada a transformada wavelet numericamente. Portanto, os resultados deste trabalho poderão ser usados como referência para estudos envolvendo sinais que não são necessariamente representados por uma função polinomial, mas que podem ser representados por uma função analítica.

2. APRESENTAÇÃO DAS WAVELETS

2.1 Conceitos preliminares

Definição 2.1.1. O espaço $L^2(\mathbb{R})$ é um espaço de funções de quadrado integrável, ou seja, de funções cujo módulo ao quadrado admite integral definida com respeito a $x \in \mathbb{R}$. Significa que essa integral tem um valor finito, para quaisquer intervalos de integração reais. O produto interno nesse espaço é definido como $\langle f, g \rangle = \int_{\mathbb{R}} f(x) \overline{g(x)} dx$, em que $\overline{g(x)}$ é o conjugado complexo de $g(x)$.

Definição 2.1.2. Uma base ortogonal para um espaço vetorial V é uma base para V cujos vetores são mutualmente ortogonais. Dessa forma, o produto interno entre os vetores que compõem a base ortogonal é nulo.

Definição 2.1.3. Seja $f: \mathbb{R} \rightarrow \mathbb{C}$ uma função. O suporte de f é o fecho do conjunto de valores $\{x \in \mathbb{R}: f(x) \neq 0\}$. Se os limites desse conjunto são bem definidos, o suporte de f é compacto.

Definição 2.1.4. Dado $a > 0$, o operador dilatação, D_a , definido sobre uma função $f(x)$ em L^1 ou L^2 sobre $\mathbb{R}$, é dado por $D_a f(x) = \frac{1}{a} f(ax)$. Este operador faz com que o gráfico da função seja “extendido” ou “contraído” nos eixos x e $f(x)$.

Definição 2.1.5. Dado $b \in \mathbb{R}$, o operador translação, T_b , definido sobre uma função $f(x)$, em L^1 ou L^2 sobre $\mathbb{R}$, é dado por, $T_b f(x) = f(x-b)$. Este operador faz com que o gráfico da função translate no eixo x .

2.2 Introdução às wavelets

As wavelets são definidas matematicamente como a seguir: sendo $L^2(\mathbb{R})$ um espaço de funções de quadrado integrável, $\Psi_j, k(x): L^2 \rightarrow \mathbb{R}^2$ é uma base ortogonal de $L^2(\mathbb{R})$ tal que:

$$\{\Psi_{j,k}(x)\}_{j,k,m \in \mathbb{Z}} = \{D_{2^j} T_k \Psi(x)\}_{j,k \in \mathbb{Z}} = \left\{ 2^{\frac{j}{2}} \Psi(2^j x - 2^j k) \right\}_{j,k \in \mathbb{Z}}. \quad (1)$$

Os termos j e k da Equação (1) são os parâmetros das operações de dilatação e translação, respectivamente. O termo j é chamado de resolução da wavelet. Dessa definição de uma wavelet, é possível extrair duas propriedades relevantes: Seu valor médio é nulo e a energia da função é finita. Essas duas propriedades podem ser expressas, respectivamente, pelas duas equações a seguir:

$$\int_{-\infty}^{\infty} \Psi(x) dx = 0. \quad (2)$$

$$\int_{-\infty}^{\infty} |\Psi(x)|^2 dx < L \quad L \in \mathbb{N}. \quad (3)$$

A Equação (2) se deve a uma wavelet constituir uma base ortogonal do $L^2(\mathbb{R})$. Já ter energia finita (propriedade representada em (3)) é uma expressão oriunda da física. A Equação (3) é consequência do fato das wavelets terem suporte compacto. Por isso, diz-se que a wavelet apresenta duração finita, sendo interpretada como uma “onda pequena”. A transformada wavelet de uma função $f(t)$, por sua vez, é dada por:

$$W_{j,k}(f) = \int_{-\infty}^{\infty} f(t) \Psi_{j,k}(t) dt. \quad (4)$$

Sendo $\Psi_{j,k}(t)$ chamada de wavelet mãe. Na introdução, foi dito que a transformada wavelet permite uma maior flexibilidade para solução de problemas do que a transformada de Fourier. A razão matemática disso se torna, agora, evidente: como as funções de base da série de Fourier não têm energia finita, torna-se difícil adequadamente funções com duração limitada ou com mudanças bruscas no comportamento. Como as wavelets têm duração limitada, torna-se muito mais fácil lidar com esses problemas com precisão.

Utilizar a transformada wavelet para resolver problemas práticos apresenta um problema inicial, no entanto. Entretanto, para vários problemas reais, é pouco prático e virtualmente inviável abordar a transformada wavelet de forma contínua tal qual está exposta em (4), já que a expressão analítica de suas funções de base não é conhecida para a grande maioria das wavelets. Por isso, a wavelet é discretizada e calculada numericamente, e a forma como essa discretização será feita dependerá do tipo de wavelet escolhida para a análise.

2.3 Wavelets de Daubechies

Neste trabalho, a família de wavelets usada é a família das wavelets de Daubechies, que recebem esse nome devido à Ingrid Daubechies, responsável por conceituá-las.

As Wavelets de Daubechies são funções definidas a partir de uma combinação linear de funções de escala com suporte $[0, N - 1]$, em que as translações dessas funções de escala obedecem a uma condição de ortonormalidade. Nessa combinação linear, há N coeficientes de filtro, que representam os pesos da combinação linear. A quantidade N que determina o limite superior do suporte e a quantidade de coeficientes de filtro é definida como o gênero da Daubechies. As funções de escala não apresentam definição analítica, no entanto, conforme mostrado em Silva (2019), as propriedades apresentadas nesta Seção 2.3 são suficientes para determinar numericamente suas formas, com o auxílio de técnicas de interpolação. A maneira como esse processo pode ser feito faz parte da primeira etapa da metodologia deste trabalho e será apresentada na Seção 3.1.

3. METODOLOGIA

A metodologia consistirá em seguir o passo-a-passo proposto em Silva (2020) para representar numericamente qualquer polinômio do tipo x^k em função de wavelets, para em seguida representar numericamente funções analíticas em função de polinômios do tipo x^k (Por meio da expansão em um polinômio de Taylor). Desta maneira, qualquer função analítica pode ser representada em termos de wavelets de Daubechies numericamente. Por fim, será feita uma análise dos resultados e uma reflexão acerca de possíveis aplicações para esta metodologia. Este passo-a-passo pode ser separado nas 4 etapas a seguir (o código computacional para realizar a Etapa 1 está no Apêndice B e os códigos para realizar as Etapa 2 e 3 no Apêndice C):

Etapa 1 (Seção 3.1)- Encontrar numericamente funções de escala que descrevam a wavelet desejada. Para este trabalho, aplicou-se esse método para wavelets Daubechies de gênero 4, 6 e 8.

Etapa 2 (Seção 3.2)- Utilizar as funções escala obtidas na Etapa 1 para representar polinômios do tipo x^k , por meio do uso de uma função geradora de momentos, que será apresentada na Equação (11).

Etapa 3 (Seção 3.3)- Expandir as funções analíticas estudadas neste trabalho em polinômios de Taylor, mediante um truncamento em um certo valor, e representar os polinômios como wavelets por meio dos resultados da Etapa 2.

Etapa 4 (Seção 3.4)- Avaliar os resultados e possíveis aplicações práticas para a metodologia estudada.

3.1 Obtenção numérica de uma Wavelet de Daubechies

Inicialmente, foram desenvolvidos códigos computacionais em Python para representar Daubechies de gênero 4, 6 e 8, e estão apresentadas no Apêndice B deste trabalho. Para mostrar como isso foi feito, será realizada uma explanação da metodologia utilizada em Silva (2019) para permitir a determinação numéricas das wavelets do tipo Daubechies, bem como de detalhes do funcionamento do código computacional desenvolvido neste trabalho, com base nas demonstrações feitas em Burgos (2009).

As Daubechies não são funções que possuem uma forma analítica definida, mas podem ser determinadas numericamente com auxílio do método de Wavelet-Galerkin, que é uma adaptação do método de Galerkin. A

formulação matemática deste método não será abordada neste trabalho, visto que ele é consequência de diversos trabalhos realizados na tentativa de representar wavelets numericamente. Um breve histórico desse desenvolvimento matemático pode ser encontrado em Burgos (2009). De um modo geral, esse método consiste em representar a wavelet por uma combinação linear de uma base ortonormal. Embora essas funções de base não tenham forma conhecida analiticamente (assim como a wavelet), o método consiste em gerar elas de modo que os coeficientes dessa combinação linear possam ser determinados numericamente por alguma outra metodologia e, ainda, satisfazendo as condições de contorno das wavelets. Esse método é possível por conta de propriedades mostradas nas Seções 2.2 e 2.3. Por meio do método de wavelet-Galerkin para discretizar a wavelet de Daubechies, conforme mostrado em Burgos (2009), é obtida a equação a seguir:

$$\phi(x) = \sum_{k=0}^{N-1} a_k \phi(2x - k), \quad (5)$$

em que $\phi(x)$ é a função escala, que representa as funções de base que geram a wavelet. N é o número que determina o gênero da Daubechies e a_k são os coeficientes de Daubechies ou coeficientes de filtro, com valores que dependem apenas do gênero da Daubechies. Ingrid Daubechies (1998) mostrou que, para descrever adequadamente uma função de escala de uma wavelet de Daubechies de gênero N , são necessários N coeficientes de filtro. Para Daubechies de gêneros de índices pequenos, como 4, 6 e 8, que são as únicas wavelets utilizadas neste trabalho, os valores dos coeficientes de filtro já são bem conhecidos na literatura e podem ser encontrados no apêndice deste trabalho. Em Silva (2019), apresenta-se a forma de determinar os coeficientes a_k para Daubechies de qualquer gênero, a partir da Equação (5), mesmo sem determinar numérica ou analiticamente as funções escala.

Os limites do somatório da Equação (5) têm como objetivo obter o valor da função escala $\phi(x)$ para tantos valores de x quanto possível, uma vez que quanto mais pontos, mais conhecido o comportamento da função será. Tais limites são considerados finitos na análise, visto que a wavelet possui valor nulo para valores fora dos seus suportes. Isto porque o suporte inferior de uma função de escala de uma wavelet do tipo Daubechies é $x = 0$, enquanto o superior é $x = N - 1$, em que N é o gênero da wavelet, conforme demonstrado em Burgos (2009). Por exemplo, a função de escala da Daubechies 6 tem imagem nula para $x < 0$ e para $x > 5$. Por isso, na análise numérica, os limites inferior e superior do somatório passam a ser, respectivamente, $k = 0$ e $k = N - 1$, conforme mostrado na Equação (5). Para dar prosseguimento ao cálculo computacional, é preciso considerar duas propriedades importantes das funções de escala das Daubechies:

$$\phi(0) = \phi(N - 1) = 0, \quad (6)$$

$$\sum_{k=0}^{N-2} \phi(x - k) = 1. \quad (7)$$

A Equação (6) é consequência das funções escala das Daubechies serem contínuas e, para $x < 0$ e para $x > N - 1$, nulas. Já a Equação (7) é consequência da condição de normalização das funções de escala das Daubechies, mencionada na seção 2.3.

Considerando apenas valores inteiros de x pertencentes ao suporte de $\phi(x)$, já que outros valores de x geram implicam em $\phi(x) = 0$, a Equação (5) é representada no âmbito computacional como um sistema matricial, em que é preciso resolver para a matriz de $\phi(x)$. Expandindo o somatório representado em (5), substituindo nele a equação (6) e agrupando os termos de forma conveniente, obtém-se a matriz a seguir:

$$\begin{bmatrix} a_1 & a_0 & 0 & \cdots & 0 & 0 & 0 \\ a_3 & a_2 & a_1 & \cdots & 0 & 0 & 0 \\ a_5 & a_4 & a_3 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{n-4} & a_{n-5} & a_{n-6} \\ 0 & 0 & 0 & \cdots & a_{n-2} & a_{n-3} & a_{n-4} \\ 0 & 0 & 0 & \cdots & 0 & a_{n-1} & a_{n-2} \end{bmatrix} \begin{pmatrix} \phi(1) \\ \phi(2) \\ \phi(3) \\ \vdots \\ \phi(N-4) \\ \phi(N-3) \\ \phi(N-2) \end{pmatrix} = \begin{pmatrix} \phi(1) \\ \phi(2) \\ \phi(3) \\ \vdots \\ \phi(N-4) \\ \phi(N-3) \\ \phi(N-2) \end{pmatrix}. \quad (8)$$

Sistemas de equações no mesmo modelo que o da Equação (8) são problemas de autovalor de solução já conhecida. Neste caso, há um autovalor unitário correspondente a um autovetor associado a uma certa matriz A . O autovetor é a matriz das funções escala, enquanto a matriz A é a matriz dos coeficientes de filtro. Por isso, a Equação (8) pode ser reescrita, obtendo-se a expressão a seguir:

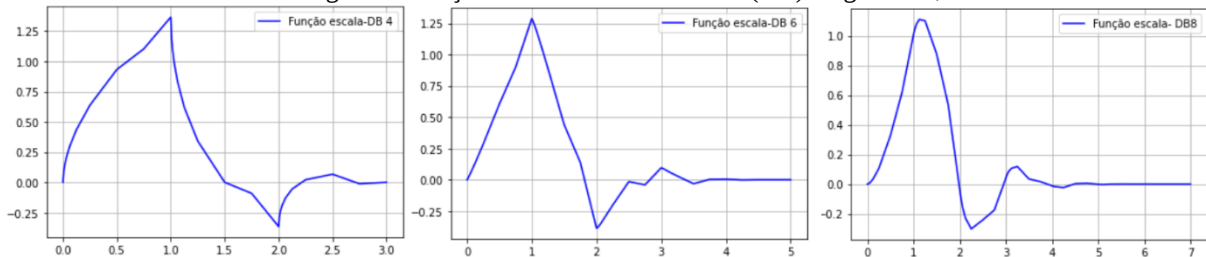
$$\begin{bmatrix} a_1 - 1 & a_0 & 0 & \dots & 0 & 0 & 0 \\ a_3 & a_2 - 1 & a_1 & \dots & 0 & 0 & 0 \\ a_5 & a_4 & a_3 - 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & a_{n-4} - 1 & a_{n-5} & a_{n-6} \\ 0 & 0 & 0 & \dots & a_{n-2} & a_{n-3} - 1 & a_{n-4} \\ 0 & 0 & 0 & \dots & 0 & a_{n-1} & a_{n-2} - 1 \end{bmatrix} \begin{pmatrix} \phi(1) \\ \phi(2) \\ \phi(3) \\ \vdots \\ \phi(N-4) \\ \phi(N-3) \\ \phi(N-2) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{pmatrix}. \quad (9)$$

É possível observar que a Equação (9) trata-se de um sistema homogêneo, pois o determinante da matriz dos coeficientes não é nulo, razão pela qual não apresenta solução única. Para contornar esse problema, recorre-se à Equação (7), que é incluída como uma nova equação no sistema e permite torná-lo possível e determinado. Feita a inclusão desta nova equação, o sistema pode ser resolvido facilmente por uma funcionalidade disponível na biblioteca do Python para resolver equações matriciais desse tipo. No código mostrado no Apêndice B, é possível visualizar esse procedimento ao se gerar cada Daubechies.

É importante salientar que essa análise é feita apenas com valores inteiros de k , de modo que a solução numérica obtida será apenas uma matriz com termos $\phi(x)$ com x inteiro variando de 0 a $N - 1$, conforme mostrado. No entanto, esses pontos ainda não são suficientes para obter uma expressão próxima de uma função contínua para $\phi(x)$ que a represente fielmente. Para obter o valor de $\phi(x)$ para outros valores não inteiros de x entre 0 e $N - 1$, é preciso realizar mais um processo iterativo, resolvendo a Equação (5) numericamente para outros valores de x . Porém, os valores de x utilizados para determinar os pontos da wavelet deverão ser múltiplos de $\frac{1}{2^q}$, sendo q inteiro. Essa necessidade é decorrente da Equação (5), pois após os resultados obtidos em (9), se conhece o valor de $\phi(x)$ para qualquer valor inteiro e desconhece para qualquer outro número. Logo, quando se aplica um valor de x não inteiro à equação (5), a função duplicará esse x e subtrairá o valor de m , e checará se esse argumento é inteiro. Caso não, a função precisará aplicar esse argumento nela mesma, aplicando o mesmo procedimento até encontrar um valor inteiro, cuja imagem é conhecida. Por isso, caso se tente aplicar a Equação (5) computacionalmente para valores de x não múltiplos de $\frac{1}{2^q}$, o computador ficará calculando infinitamente, sendo incapaz de determinar valores para $\phi(x)$ partindo apenas dos dados dos resultados do sistema obtido na Equação (9). Por exemplo, se $x = 0,3$, não importa quantas vezes se duplique este valor, nunca se obterá um número inteiro e, portanto, nunca será possível achar a imagem de $\phi(2 * 0,3 - m)$ a partir de (5). Esse cuidado com a escolha dos valores de x para determinar a função escala pode ser observado no código exibido nos Apêndices B e C, em que todas as listas de valores de x apresentam apenas valores no formato apropriado.

Com isso, determina-se computacionalmente o valor de $\phi(x)$ para vários valores de x no intervalo $0 \leq x \leq N - 1$. Quanto mais valores de x forem considerados no programa, mais precisa será a construção das funções de escala. O código em Python criado neste trabalho para determinar as Daubechies 4, 6 e 8 pode ser encontrado no apêndice. Os gráficos de suas funções escala são apresentados na figura a seguir:

Figura 1- Funções escala das Daubechies (DB) de gênero 4, 6 e 8.



Fonte: Autoria própria (2021).

3.2 Aplicação numérica da transformada wavelet a x^k

Em posse dessas funções, é possível representar qualquer monômio do tipo x^k como uma soma discreta de wavelets, conforme a equação a seguir:

$$x^k = \sum_{m=-\infty}^{\infty} c_k \phi(2^j x - m), \quad (10)$$

sendo m a quantidade de translações da wavelet e sendo os valores de c_k os coeficientes das funções de escala. As translações $\phi(2^j x - m)$ podem facilmente ser encontradas numericamente, uma vez que já se foram obtidas as funções de escala na Sessão 3.1. No Apêndice B, o código mostrado também demonstra esse processo. É importante salientar que ϕ , da maneira mostrada em (10), poderia ser a função de escala de qualquer gênero de

Daubechies. Entretanto, como será mostrado posteriormente nesta sessão, não se pode representar qualquer função com qualquer gênero de Daubechies com precisão e será preciso determinar o gênero apropriado para a análise.

Para determinar os coeficientes c_k , será utilizada a função geradora de momentos em relação ao monômio x^k , definida em Silva (2019). Sendo W o espaço das wavelets, a função geradora de momentos é definida como $M_m^k : W \rightarrow \mathbb{R}$ e calculada pela Equação (11) a seguir, que é desenvolvida em Silva (2019) para obter a Equação (12) e construir uma expressão para x^k , mostrada na Equação (13) logo em seguida:

$$M_m^k = \int_{-\infty}^{\infty} x^k \phi(x - m) dx \quad (11)$$

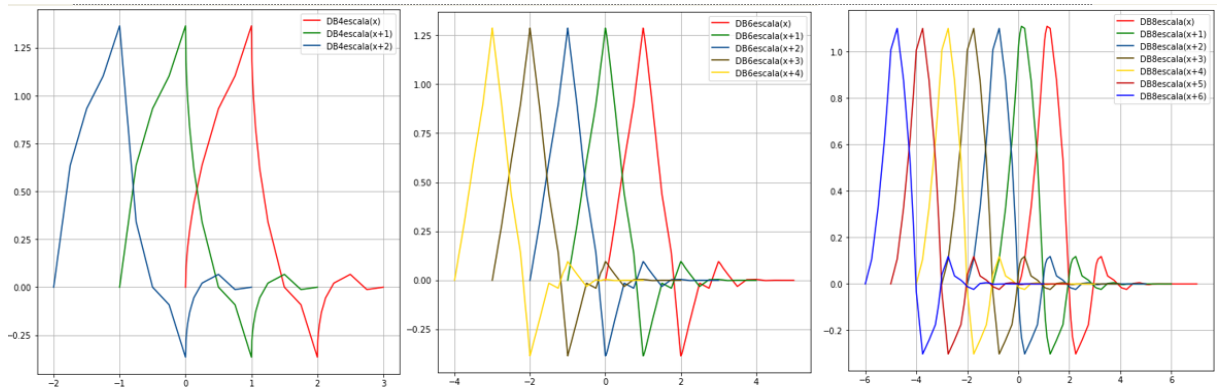
$$M_m^k = \begin{cases} \sum_{r=0}^{k-1} \binom{k}{r} \sum_{s=0}^{N-1} a_s s^{k-r} M_0^r, & \text{se } m = 0; k \neq 0 \\ \sum_{r=0}^{k-1} \binom{k}{r} m^{k-r} M_0^r, & \text{se } m \neq 0; k \neq 0 \\ 1, & \text{se } m = k = 0, \end{cases} \quad (12)$$

$$x^k = \sum_{m=-\infty}^{+\infty} \frac{M_m^k}{2^{jk}} \phi(2^j x - m), \quad (13)$$

A Equação (12) dá como retorno números que serão divididos por 2^{jk} para obter os coeficientes c_k das funções de escala. Os valores obtidos a partir de (12) que foram utilizados para os estudos deste trabalho se encontram na Tabela 2, no Apêndice A. Com a Equação (13), é possível encontrar o valor de $f(x) = x^k$ ponto a ponto, substituindo valores de x que sejam da forma $\frac{1}{2^q}$. Quanto mais valores considerados, melhor a aproximação. Porém, antes disso, é preciso escolher qual gênero de wavelet utilizar. Logo, é preciso investigar se qualquer wavelet pode representar qualquer polinômio a partir da Equação (13). Para responder isso, é preciso saber que uma base formada por funções de escala de Daubechies de ordem N é capaz de representar exatamente polinômios de grau até $\frac{N}{2} - 1$. Isto se deve à propriedade de que wavelets de gênero N têm apenas $\frac{N}{2}$ momentos nulos, conforme demonstrado em Uzinski (2013). Logo, o polinômio x , por exemplo, precisa ser representado por Daubechies de gênero de no mínimo 4, pois $\frac{4}{2} - 1 \geq k = 1$. Por sua vez, x^2 precisaria de Daubechies de gênero 6 (pois $\frac{6}{2} - 1 \geq k = 2$), x^3 precisaria de Daubechies de gênero 8 (pois $\frac{8}{2} - 1 \geq k = 3$) e assim por diante.

Um fator importante a ser considerado ao se aplicar a Equação (12) numericamente é que a quantidade mínima de translações necessárias da função escala para representar um polinômio x^k com uma Daubechies de gênero N é $N - 1$. Esta propriedade é demonstrada em Silva (2019). Todas as translações devem ter m inteiro e devem ser feitas de modo a interceptar o intervalo de representação de x . Por exemplo, se o objetivo é determinar um polinômio x utilizando DB4 (Daubechies 4) com resolução $j = 0$ entre $x = 0$ e $x = 1$, a função de escala com translação nula para as DB4 ficaria no formato $\phi(x)$, com suporte $[0,3]$. Logo, as translações apropriadas seriam $\phi(x)$, $\phi(x + 1)$ e $\phi(x + 2)$. Já para representar, por exemplo, x^2 via DB6 com $j = 0$, as translações apropriadas seriam $\phi(x)$, $\phi(x + 1)$, $\phi(x + 2)$, $\phi(x + 3)$ e $\phi(x + 4)$. Para representar x^3 via DB8 sob as mesmas condições, as translações seriam $\phi(x)$, $\phi(x + 1)$, $\phi(x + 2)$, $\phi(x + 3)$, $\phi(x + 4)$, $\phi(x + 5)$ e $\phi(x + 6)$, e assim por diante. A figura a seguir ilustra as translações das funções de escala mencionadas, evidenciando que elas são as únicas translações (com m inteiro) de $\phi(x)$ que passam pelo intervalo $[0,1]$ para cada gênero de Daubechies citado:

Figura 2- Translações das funções de escala para as DB 4, DB 6 e DB6, respectivamente



Fonte: Autoria própria (2021).

É importante estabelecer o valor da resolução j , pois ela influencia o argumento da função de escala e o coeficiente das translações em (13). Se o objetivo fosse representar uma função polinomial em um intervalo entre

valores de x quaisquer, a resolução precisaria ser alterada de modo que todas a função pudesse ser estendida lateralmente, para que suas translações pudessem ter valor não nulo dentro de todo o intervalo.

Convém destacar que o somatório em (13) inclui todas as translações $\phi(2^j x - m)$. Porém, quando as translações estiverem fora do suporte de $\phi(2^j x)$, sua imagem será nula e não contribuirá para o resultado do somatório. Esse fato é levado em consideração no código gerado e pode ser percebido no Apêndice B.

Os valores da função geradora de momentos, por sua vez, são determinados computacionalmente, obtendo-se uma expressão para x^k em função de wavelets determinadas numericamente, utilizando a wavelet de gênero adequado. O código realizado em Python neste trabalho retorna, portanto, um vetor com valores de x^k correspondente a um intervalo pré-definido de valores discretos de x . Com isso, tem-se o necessário para realizar a última etapa da metodologia, que é utilizar esses códigos computacionais para gerar funções analíticas em termos de x^k representado como wavelets. Isso será possível com o auxílio da expansão de Taylor.

3.3 Aplicação da transformada wavelet discreta à funções analíticas

Até então, foi feita uma introdução a respeito das wavelets e dos cálculos realizados em Silva (2019). Esse momento era importante, pois com base nesses conhecimentos, é possível analisar funções analíticas mediante a transformada wavelet. Uma função analítica $f(x)$ qualquer pode ser expandida em uma série de Taylor em torno de um ponto. Neste caso, o polinômio de Taylor equivalente, desenvolvido em torno do ponto $x_0 = 0$, é dado pela Equação (13) a seguir:

$$f(x) = \sum_{k=0}^{+\infty} \frac{f^{(k)}(0)}{k!} x^k, \quad (14)$$

Sendo $f^{(k)}(0)$ a k -ésima derivada $f(x)$ com respeito a x , para $x = 0$. Substituindo (14) em (13), obtém-se uma expressão aproximada para qualquer função analítica em termos de wavelets, utilizando os cálculos feitos em Silva (2019) para representar funções polinomiais em termos de wavelets:

$$f(x) = \sum_{k=0}^{+\infty} \sum_{m=-\infty}^{+\infty} \frac{f^{(k)}(0)}{k!} \frac{M_m^k}{2^{jk}} \Phi(2^j x - m). \quad (15)$$

A partir de (15), é possível representar diferentes funções analíticas por meio uma expansão em polinômio de Taylor com polinômios representados por wavelets, bem como avaliar a eficácia dessa metodologia em representar as funções com fidelidade. As funções estudadas estão exibidas a seguir da Equação (16) até a (21), bem como suas equivalentes expansões em polinômio de Taylor centrado em 0:

$$f(x) = x, \quad (16)$$

$$f(x) = x^2, \quad (17)$$

$$f(x) = x^3, \quad (18)$$

$$f(x) = e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}, \quad (19)$$

$$f(x) = \text{sen}(x) = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} x^{2k+1}, \quad (20)$$

$$f(x) = \text{cos}(x) = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k)!} x^{2k}. \quad (21)$$

O intervalo analisado para todas essas funções foi $0 < x < 1$. Para as Equações (16), (17) e (18), foram utilizadas wavelets de gênero 4, 6 e 8, respectivamente, todas com resolução 0. Os coeficientes foram calculados a partir dos resultados a Equação (12) e aplicados à Equação (13) com as devidas translações para o intervalo de estudo, obtendo-se:

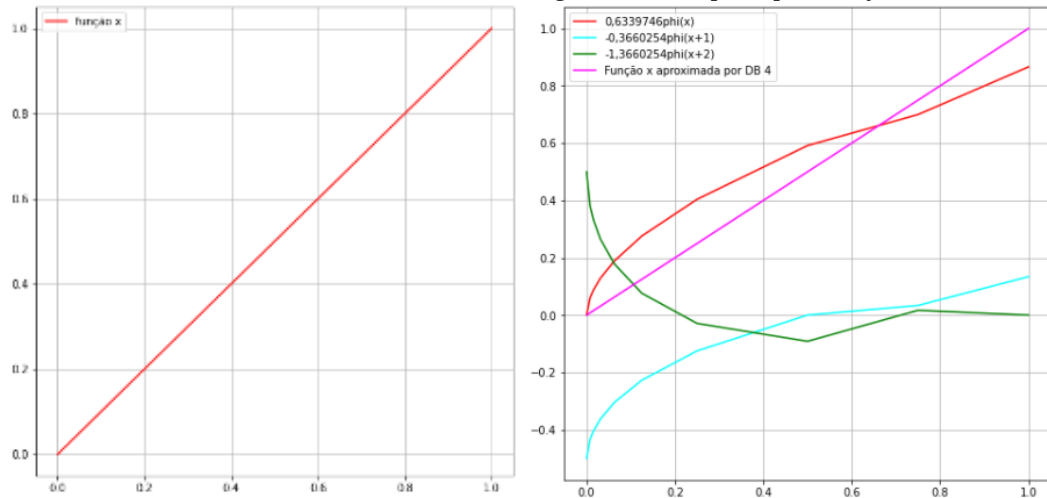
$$x = 0,6339746\phi(x) - 0,3660254\phi(x+1) - 1,3660254\phi(x+2), \quad (22)$$

$$x^2 = 0,6681447\phi(x) + 0,03334233\phi(x+1) + 1,398539\phi(x+2) + 4,763737\phi(x+3) + 10,12893\phi(x+4), \quad (23)$$

$$x^3 = 0,9073604\phi(x) - 0,1089065\phi(x+1) - 1,092814\phi(x+2) - 8,044362\phi(x+3) - 26,9636\phi(x+4) - 63,8503\phi(x+5) - 124,7048\phi(x+6). \quad (24)$$

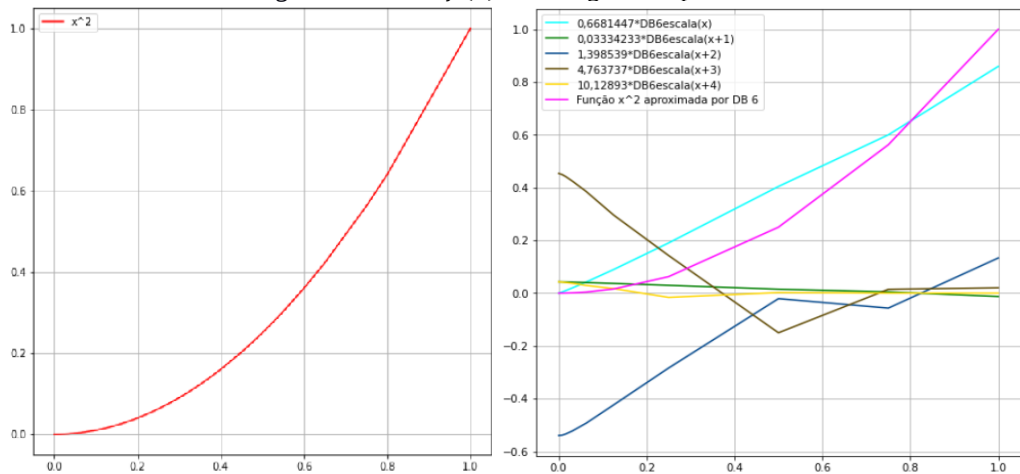
As figuras a seguir comparam os gráficos gerados com os gráficos esperados para as 3 funções acima. O Apêndice C mostra o código usado para gerar tais funções.

Figura 3- Gráfico $f(x) = x$ e gráfico obtido pela aproximação



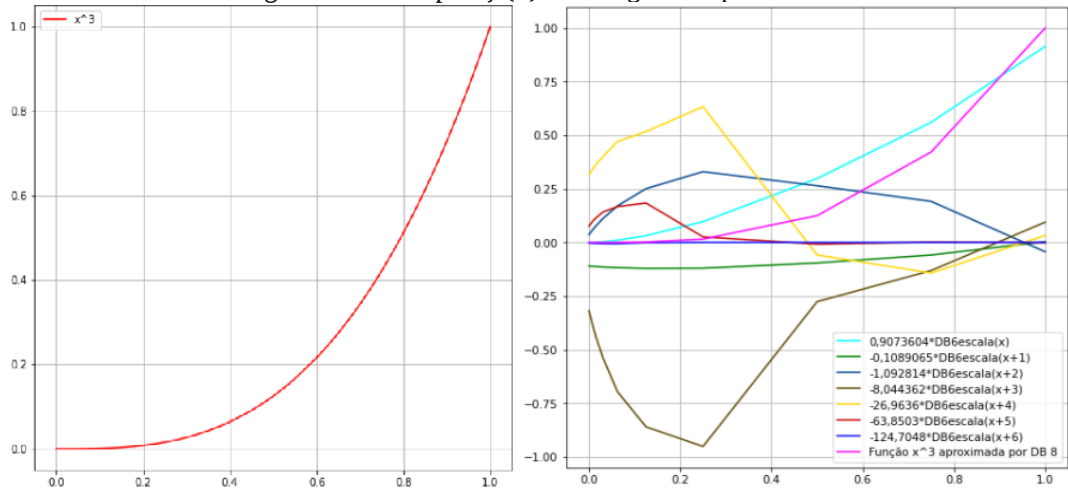
Fonte: Autoria própria (2021).

Figura 4- Gráfico $f(x) = x^2$ e gráfico aproximado



Fonte: Autoria própria (2021).

Figura 5- Gráfico para $f(x) = x^3$ e gráfico aproximado



Fonte: Autoria própria (2021).

Escrevendo as séries de Taylor das Equações (19), (20) e (21) até, respectivamente, $k = 3$, $k = 1$ e $k = 1$, tem-se:

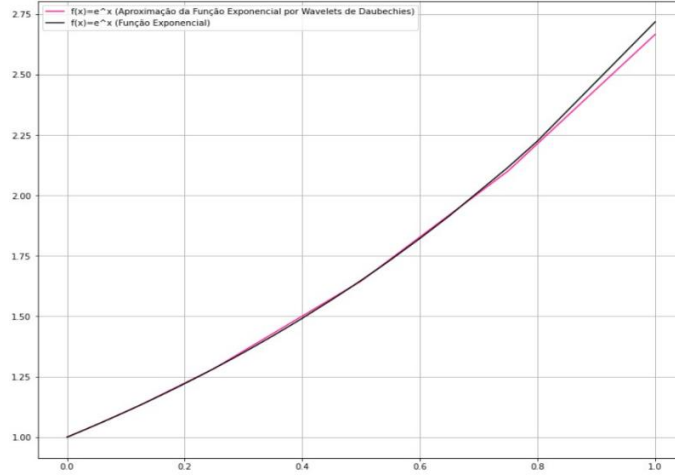
$$e^x \cong 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!}, \quad (25)$$

$$\text{sen}(x) \cong x - \frac{x^3}{3!}, \quad (26)$$

$$\cos(x) \cong 1 - \frac{x^2}{2!}. \quad (27)$$

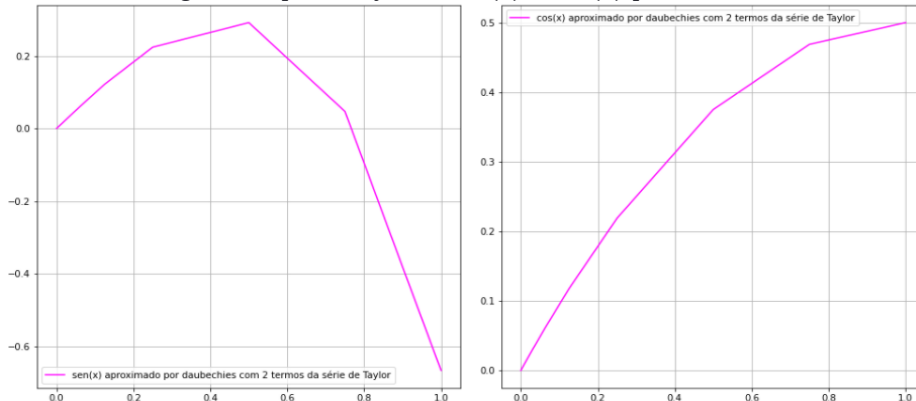
Em seguida, basta substituir as Equações (22), (23) e (24) nas Equações (25), (26) e (27). As figuras a seguir mostram os gráficos obtidos:

Figura 6- Função $f(x) = e^x$ e sua aproximação por wavelets



Fonte: Autoria própria (2021)

Figura 7- Aproximação de $\text{sen}(x)$ e $\cos(x)$ por wavelets



Fonte: Autoria própria (2021)

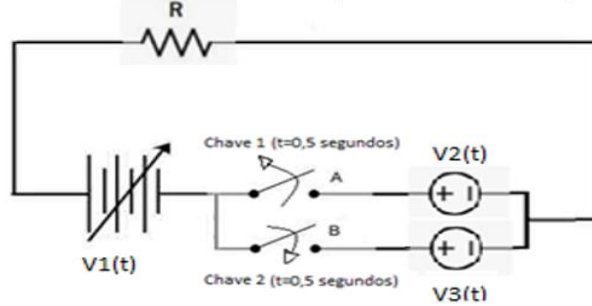
Como esperado, as aproximações feitas para $\text{sen}(x)$ e $\cos(x)$ apresentaram grandes erros, pois foram considerados apenas 2 termos do somatório, e suas séries de Taylor oscilam entre termos positivos e negativos, aumentando ainda mais a necessidade de mais termos para uma aproximação mais fidedigna. Esta aproximação ruim foi feita propositalmente, a fim de demonstrar que, para algumas funções, esta metodologia requer que mais termos da série de Taylor sejam considerados. Para considerar mais termos, seriam necessárias potências maiores de x e, portanto, seria preciso gerar Daubechies de gênero maior para representar essas potências, utilizando o mesmo procedimento apresentado neste trabalho. Já para $f(x) = e^x$, a aproximação no intervalo analisado se mostrou satisfatória. Isso é consequência de terem sido considerados mais termos da série de Taylor e de seus polinômios de Taylor não mudarem de sinal periodicamente.

3.4 Ideias de possíveis aplicações

Para demonstrar a utilização do que foi estudado, será resolvido um problema ilustrativo de circuitos elétricos. Segue o problema:

Considere o circuito da Figura 8 a seguir. Para $t=0$ s, a chave A está fechada e a chave B aberta. Em $t = 0,5$ s, a chave A abre e a chave B fecha, desligando a fonte V_2 do circuito e conectando a fonte V_3 , e a resistência interna de todas as fontes é desprezível. Tem-se que $V_1(t) = 3t$ V, $V_2(t) = e^t$ V e $V_3(t) = e^{-0,5t}$ V. Imediatamente após $t=1$ s, todas as fontes geradoras são desligadas. Determine a tensão elétrica entre os terminais do resistor R para $0s \leq t \leq 1s$ utilizando wavelets de Daubechies.

Figura 8- Circuito elétrico (Problema ilustrativo)



Autoria própria (2021).

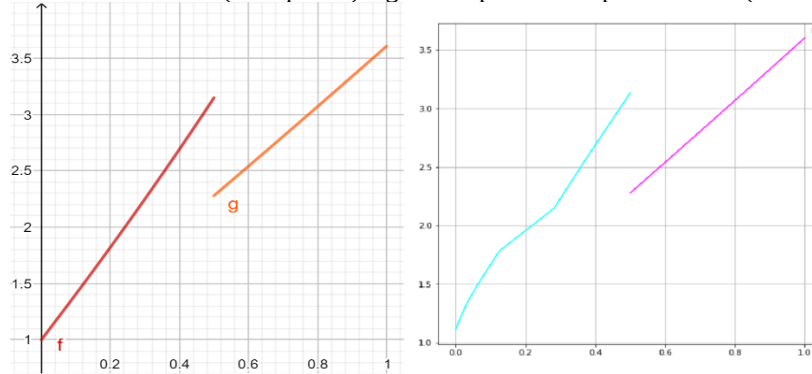
Solução: Pela lei de Kirchhoff para tensões, sabe-se que a tensão $V_R(t)$ sobre o resistor será a soma das tensões das fontes. Assim, tem-se:

$$V_R(t) = \begin{cases} 3t + e^t, & \text{se } 0s \leq t \leq 0,5s \\ 3t + e^{-0,5t}, & \text{se } 0,5s < t \leq 1s \end{cases}$$

Neste contexto, a variável t será a abscissa, sendo equivalente ao x . A tensão $V_1(t)$ será substituída pela Equação (21) multiplicada por 3. A tensão $V_2(t)$, por sua vez, será aproximada pela Equação (24). Já a tensão $V_3(t)$, expandida em uma série de Taylor centrada em 0 e considerando os 4 primeiros termos do somatório é dada por:

$$e^{-0,5t} \cong 1 - 0,5t + \frac{0,25t^2}{2!} - \frac{0,125t^3}{3!}$$

Figura 9- Gráfico exato (À esquerda) e gráfico aproximado por wavelets (À direita)



Fonte: Autoria própria (2021).

Na Figura 9, o gráfico à esquerda foi gerado no geogébra. É possível perceber que a ondulação que deveria ocorrer em torno de $x = 0,2$ deveria ser levemente mais suave. A diferença se deve ao fato de terem sido considerados apenas 10 valores de x no momento de discretização. Quanto mais pontos forem considerados, melhor será a fidelidade do gráfico.

Este exemplo, embora simples, teve como objetivo mostrar que a metodologia deste trabalho pode ser útil para modelar funções que apresentam duração limitada e/ou mudança de comportamento abrupta (no caso do problema ilustrado, ambos os casos ocorrem). Embora seja uma metodologia restrita a funções analíticas, muitos problemas práticos são descritos como funções analíticas. Por exemplo, fontes geradoras de tensão elétrica em geral fornecem, na prática, tensões com comportamento linear, exponencial ou alternado (senoidal/cossenoidal),

comportamentos correspondentes a funções analíticas. Uma forma de complicar o problema ilustrado seria incluir mais estados transitórios, e trocar o resistor por uma ou mais cargas com impedância que não seja puramente resistiva, dificultando ainda mais a análise desses regimes transientes por métodos analíticos. Em geral, muitos problemas da física, ou mesmo engenharia, consistem em modelagem de funções que mudam de comportamento ao cruzar uma fronteira, desde exemplos mais simples até problemas mais complexos, como determinar a função de onda de um elétron ao atravessar regiões sujeitas a diferentes energias potenciais. Vale salientar que mesmo na física quântica, área em que se trabalha com uma matemática relativamente avançada, muitas vezes as soluções da equação de Schrödinger para uma determinada região do espaço com energia potencial conhecida são funções analíticas. Na área da mecânica, por exemplo, também há problemas envolvendo funções analíticas e descontinuidades. Por exemplo, ao se investigar a tensão de cisalhamento em uma viga irregular, a tensão cisalhante pode ser modelada por funções analíticas diferentes para diferentes regiões da viga, podendo haver mudanças bruscas de comportamento dessa grandeza ao redor das fronteiras entre regiões.

4. CONSIDERAÇÕES FINAIS

Com este trabalho, foi feito um estudo da metodologia proposta em Silva (2020) para representar polinômios como wavelets de Daubechies e, a partir disso, representar qualquer função analítica por wavelets de Daubechies, ao transformar a função em uma soma de polinômios através da expansão em série de Taylor. Então, foi desenvolvido um programa de computador na linguagem Python para aplicar esta metodologia, e investigar suas vantagens, desvantagens e possíveis aplicações práticas.

Como resultado desse estudo, foi possível averiguar que a eficácia dessa metodologia está sujeita a 3 fatores principais: O primeiro fator é que funções com termos da série de Taylor que oscilam muito a cada termo, como a função $\sin(x)$, requerem que mais termos sejam considerados na análise. Por outro lado, funções cujo polinômio de Taylor varia pouco a cada termo podem ser aproximadas satisfatoriamente com bem menos termos. O segundo fator é que, quanto mais valores de abscissas forem considerados ao se gerar as funções de escala, mais fiel à função original a aproximação será.

Uma vez que muitos problemas da física ou engenharia que são difíceis de resolver analiticamente envolvem funções analíticas e mudanças bruscas de comportamento ou curta duração, é possível modelar tais problemas mediante a transformada discreta de wavelet a partir da metodologia estudada, de forma prática e, sob as devidas condições, satisfatória. Essa possibilidade permite que a análise de tais problemas possa usufruir dos benefícios da análise de wavelets, que consiste principalmente em obter informações precisas sobre a função em diferentes escalas de tempo e frequência.

5. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] SILVA, Paulo César Linhares da. Use of Daubechies Wavelets in the Representation of Analytical Functions. **IntechOpen**, 2020. Disponível em <https://api.intechopen.com/chapter/pdf-preview/73428>. Acesso em 21. mai 2021. DOI: <http://dx.doi.org/10.5772/intechopen.93885>.
- [2] SILVA, Paulo César Linhares da. **Aplicação das Wavelets de Daubechies em Conjunto com o Método de Propagação Vetorial de Feixes na Análise de Estruturas Fotônicas**. Natal-RN: UFRN, 2019. 121p. (Tese de Doutorado em Programa de Pós-Graduação em Engenharia e de Computação) Disponível em <https://repositorio.ufm.br/jspui/handle/123456789/27573>. Acesso em 21. mai 2021.
- [3] BURGOS, Rodrigo Bird. **Análise de Estruturas Utilizando Wavelets de Daubechies e Interpolets de Deslauriers-Dubuc**. Rio de Janeiro-RJ: Pontifícia Universidade Católica do Rio de Janeiro. 2009, 185p. (Tese de Doutorado em Engenharia Civil). Disponível em https://www.maxwell.vrac.puc-rio.br/15505/15505_1.PDF. Acesso em 21. mai 2021.
- [4] UZINSKI, Julio Cezar. **Momentos nulos e regularidade wavelet na detecção de falhas em sinais**. Ilha Solteira-SP: Universidade Estadual Paulista “Júlio de Mesquita Filho”, 2013. 103p. (Dissertação de Mestrado em Engenharia Elétrica). Disponível em https://www.feis.unesp.br/Home/departamentos/engenhariaeletrica/pos-graduacao/julio-cezar-uzinski_final.pdf. Acesso em 21. mai 2021.
- [5] SILVEIRA, Thiago da.; KOZAKEVICIUS, Alice J. **Transformada wavelet de haar: conceitos, formulações e aplicações**. Rio Grande: FURG. 2016. Disponível em https://www.sbm.org.br/wp-content/uploads/2016/06/transformada-wavelet-de-haar_ebook.pdf. Acesso em 21. mai 2021.
- [6] SILVA, P. C. L.; MELO, R. R.; DA SILVA, J. P. Optical Fiber Coupler Analysis Using Daubechies Wavelets. **Journal of Microwaves, Optoelectronics and Electromagnetic Applications**, São Caetano do Sul, v. 19, n. 3, p. 294-300, Set. 2020. Disponível em

http://www.scielo.br/scielo.php?script=sci_arttext&pid=S2179-10742020000300294&lng=en&nrm=iso. Acesso em 21. Mai 2021. Epub Sep 16, 2020. <https://doi.org/10.1590/2179-10742020v19i3825>.

[7] DAUBECHIES, Ingrid. Recent results in wavelets applications. Journal of Electronic Imaging, vol.7, 1998.

[8] FARGE, Marie. “Wavelet transforms and their applications to turbulence”. Annu. Rev. Fluid Mech, 1992. Disponível em <https://www.annualreviews.org/doi/10.1146/annurev.fl.24.010192.002143>. Acesso em 21. Mai 2021.

[9] LATHI, B. P. Sinais e sistemas lineares. 2º edição. Porto Alegre: Bookman, 2007.

[10] YOUNG, H. D.; Freedman. R. A. Física IV: Óptica e física moderna. 12º edição. São Paulo: PEARSON 2009. Disponível em <https://docero.com.br/doc/s1snn8>. Acesso em 21, Mai 2021.

APÊNDICE A – COEFICIENTES DE FILTRO E MOMENTOS DAS FUNÇÕES ESCALA

Tabela 1- Coeficientes de filtro a_k das funções escala para Daubechies 4, 6 e 8

Daubechies 4	Daubechies 6	Daubechies 8
$a_0 = 0,683012701892219$	$a_0 = 0,4704672077841637$	$a_0 = 0,3258034280512984$
$a_1 = 1,183012701892219$	$a_1 = 1,141116915831444$	$a_1 = 1,010945715091829$
$a_2 = 0,316987298107781$	$a_2 = 0,6503650005262326$	$a_2 = 0,8922001382467597$
$a_3 = -0,183012701892219$	$a_3 = -0,1909344155683274$	$a_3 = -0,03957502623564464$
-	$a_4 = -0,1208322083103962$	$a_4 = -0,2645071673690398$
-	$a_5 = 0,04981749973688374$	$a_5 = 0,04361630047417725$
-	-	$a_6 = 0,04650360107098176$
-	-	$a_7 = -0,01498698933036147$

Tabela 2- Momentos das funções de escala para Daubechies 4, 6 e 8, considerando os respectivos monômios analisados com cada uma (Respectivamente, x , x^2 e x^3) e $j = 0$:

Daubechies 4	Daubechies 6	Daubechies 8
$M_0^1 = 0,6339746$	$M_0^2 = 0,6681447$	$M_0^3 = 0,9073604$
$M_{-1}^1 = -0,3660254$	$M_{-1}^2 = 0,03334233$	$M_{-1}^3 = -0,1089065$
$M_{-2}^1 = -1,3660254$	$M_{-2}^2 = 1,398539$	$M_{-2}^3 = -1,092814$
-	$M_{-3}^2 = 4,763737$	$M_{-3}^3 = -8,044362$
-	$M_{-4}^2 = 10,12893$	$M_{-4}^3 = -26,9636$
-	-	$M_{-5}^3 = -63,8503$
-	-	$M_{-6}^3 = -124,7048$

APÊNDICE B – CÓDIGOS EM PYTHON PARA ENCONTRAR AS FUNÇÕES DE ESCALA E SUAS TRANSLAÇÕES, COM $j = 0$ e $0 \leq x \leq 1$

```
#_____ DETERMINANDO FUNÇÃO ESCALA DAS DAUBECHIES 4 _____
import numpy as np
import matplotlib.pyplot as plt
a=(0.683012701892219,1.183012701892219,0.316987298107781,-0.183012701892219) #
Coeficientes de Daubechies de gênero 4
A = np.array(((a[1]-1, a[0]), (1, 1)))
b = np.array((0, 1))
c = np.linalg.solve(A, b) #Matriz com os valores de Phi(1),...,Phi(N-2)
def DB4escala(x):
    w4_escalaValor=0
    if (x==3 or x==0):
```

```

        return 0
    elif (x==1):
        return c[0]
    elif (x==2):
        return c[1]
    elif (x<0 or x>3):
        return 0
    elif (x>0 and x<3):
        for m in range(0,4):
            w4_escalaValor += a[m]*DB4escala(2*x-m)
        return w4_escalaValor
#_____ DETERMINANDO TRANSLAÇÕES DA DAUBECHIES 4 (Para 0<x<1) _____
def DB4escala1(x):    #Translação Phi(x+1)
    return DB4escala(x+1)
def DB4escala2(x):    #Translação Phi(x+2)
    return DB4escala(x+2)

#_____ DETERMINANDO FUNÇÃO ESCALA DAS DAUBECHIES 6 _____
import numpy as np
import matplotlib.pyplot as plt
l=(0.4704672077841637,1.141116915831444,0.6503650005262326,-0.1909344155683274,-
0.1208322083103962,0.04981749973688374) # Coeficientes de Daubechies de gênero 6
L = np.array(((l[1]-1,l[0],0,0), (l[3],l[2]-1,l[1],l[0]), (l[5], l[4], l[3]-
1,l[2]), (1,1,1,1)))
h = np.array((0,0,0,1))
u = np.linalg.solve(L, h)    #Matriz com os valores de Phi(1),...,Phi(N-2)
def DB6escala(x):
    w6_escalaValor=0
    if (x==5 or x==0):
        return 0
    elif (x==1):
        return u[0]
    elif (x==2):
        return u[1]
    elif (x==3):
        return u[2]
    elif (x==4):
        return u[3]
    elif (x<0 or x>5):
        return 0
    elif (x>0 and x<5):
        for m in range(0,6):
            w6_escalaValor += l[m]*DB6escala(2*x-m)
        return w6_escalaValor

#_____ DETERMINANDO TRANSLAÇÕES DA DAUBECHIES 6 (Para 0<x<1) _____
def DB6escala1(x):    #Phi(x+1)
    return DB6escala(x+1)
def DB6escala2(x):    #Phi(x+2)
    return DB6escala(x+2)

```

```

def DB6escala3(x):          #Phi(x+3)
    return DB6escala(x+3)
def DB6escala4(x):          #Phi(x+4)
    return DB6escala(x+4)

#_____ DETERMINANDO FUNÇÃO ESCALA DAS DAUBECHIES 8 _____
import numpy as np
import matplotlib.pyplot as plt
d=(0.3258034280512984,1.010945715091829,0.8922001382467597,-0.03957502623564464,-
0.2645071673690398,0.04361630047417725,0.04650360107098176,-0.01498698933036147) #
Coeficientes de Daubechies de gênero 8
D = np.array(((d[1]-1,d[0],0,0,0,0), (d[3],d[2]-1,d[1],d[0],0,0), (d[5], d[4],
d[3]-1,d[2],d[1],d[0]), (d[7],d[6],d[5],d[4]-1,d[3],d[2]), (0,0,d[7],d[6],d[5]-
1,d[4]), (1,1,1,1,1,1)))
f = np.array((0,0,0,0,0,1))
g = np.linalg.solve(D, f)
def DB8escala(x):
    w8_escalaValor=0
    if (x==7 or x==0):
        return 0
    elif (x==1):
        return g[0]
    elif (x==2):
        return g[1]
    elif (x==3):
        return g[2]
    elif (x==4):
        return g[3]
    elif (x==5):
        return g[4]
    elif (x==6):
        return g[5]
    elif (x<0 or x>7):
        return 0
    elif (x>0 and x<7):
        for m in range(0,8):
            w8_escalaValor += d[m]*DB8escala(2*x-m)
        return w8_escalaValor
#_____ DETERMINANDO TRANSLAÇÕES DA DAUBECHIES 8 (Para 0<x<1) _____
def DB8escala1(x):          #Phi(x+1)
    return DB8escala(x+1)
def DB8escala2(x):          #Phi(x+2)
    return DB8escala(x+2)
def DB8escala3(x):          #Phi(x+3)
    return DB8escala(x+3)
def DB8escala4(x):          #Phi(x+4)
    return DB8escala(x+4)
def DB8escala5(x):          #Phi(x+5)
    return DB8escala(x+5)
def DB8escala6(x):          #Phi(x+6)

```

```
return DB8escala(x+6)
```

APÊNDICE C – CÓDIGOS EM PYTHON PARA ENCONTRAR OS POLINÔMIOS x , x^2 e x^3 para $0 < x < 1$. (OS CÓDIGOS DO APÊNDICE B PRECISAM SER INSERIDOS ANTES PARA OS DO APÊNDICE C FUNCIONAREM)

```
# _____ DETERMINANDO APROXIMAÇÃO PARA  $f(x)=x$ , COM  $0 < x < 1$  _____
x=[0.0,0.0078125,0.015625,0.03125,0.0625,0.125,0.25,0.5,0.75,1.0]
y=[0.6339746*DB4escala(i)-0.3660254*DB4escala1(i)-1.3660254*DB4escala2(i) for i in
x]
plt.figure(figsize=(8, 8))
plt.plot(x,y, color='#ff02ff', label='Polinômio x aproximado')
plt.grid()
plt.legend()
plt.show()
plt.figure(figsize=(8, 8))
plt.plot(x,y0,color='#FFD700', label='função x')
plt.grid()
plt.legend()
plt.show()

# _____ DETERMINANDO APROXIMAÇÃO PARA  $f(x)=x^2$ , COM  $0 < x < 1$  _____
x2=[0.0,0.0078125,0.015625,0.03125,0.0625,0.125,0.25,0.5,0.75,1.0]
y2=[0.6681446691385907*DB6escala(i)+0.03334233351683025*DB6escala1(i)+1.39853999789
5070*DB6escala2(i)+4.763737662273309*DB6escala3(i)+10.12893532665155*DB6escala4(i)
for i in x2]
plt.figure(figsize=(8,8))
plt.plot(x2,y2, color='#FF0202', label='x^2')
plt.grid()
plt.legend()
plt.show()

# _____ DETERMINANDO APROXIMAÇÃO PARA  $f(x)=x^3$ , COM  $0 < x < 1$  _____
x3=[0.0,0.0078125,0.015625,0.03125,0.0625,0.125,0.25,0.5,0.75,1.0]
y3=[0.9073603672021224*DB8escala(i)-0.1089065333813635*DB8escala1(i)-
1.092814153305344*DB8escala2(i)-8.044362492569816*DB8escala3(i)-
26.96355155117479*DB8escala4(i)-63.85038132912025*DB8escala5(i)-
124.7048518264062*DB8escala6(i) for i in x]
plt.figure(figsize=(8,8))
plt.plot(x3,y3,color='#FF02FF',label='x^3 aproximado por DB8')
plt.grid()
plt.legend()
plt.show()
```