

Centro de Ciências Exatas
Departamento de Física
Universidade Estadual de Maringá

Trabalho de Conclusão de Curso

Explorando a Fundamentação Didática da FFT e DFT na Análise de Sinais.

Acadêmico: Mateus Kiyugoro Machado Watanabe

Orientador: Prof. Dr. André Marino Gonçalves

Coorientador: Prof. Dr. Breno Ferraz de Oliveira.

Maringá, 16 de julho de 2024
2024

Centro de Ciências Exatas
Departamento de Física
Universidade Estadual de Maringá

Trabalho de Conclusão de Curso

Explorando a Fundamentação Didática da FFT e DFT na Análise de Sinais.

Trabalho de Conclusão de Curso apresentado ao Departamento de Física da Universidade Estadual de Maringá, sob orientação do Prof. Dr. Breno Ferraz de Oliveira, com parte dos requisitos para obtenção do título de licenciatura em Física.

Acadêmico: Mateus Kiyugoro Machado Watanabe

Orientador: Prof. Dr. André Marino Gonçalves

Coorientador: Prof. Dr. Breno Ferraz de Oliveira

Maringá, 16 de julho de 2024

Conteúdo

| | |
|---|------------|
| Agradecimentos | ii |
| Resumo | iii |
| Introdução | iv |
| 1 Fundamentação Teórica | 1 |
| 1.1 Série de Fourier | 1 |
| 1.2 Transformada de Fourier | 7 |
| 1.3 Transformada Discreta de Fourier | 12 |
| 1.4 Transformada Rápida de Fourier | 20 |
| 2 Resultados e Discussões | 24 |
| 2.1 Transformada Discreta e Rápida de Fourier | 24 |
| 3 Sequência Didática | 38 |
| 3.1 Planejamento | 38 |
| 3.2 Situação Inicial | 41 |
| 3.3 Trabalho de Construção | 44 |
| 3.4 Aplicação | 48 |
| 4 Considerações Finais | 51 |
| Referências | 52 |
| Apêndice | 55 |

Agradecimentos

Meus sinceros agradecimentos ao Professor Doutor André Marino Gonçalves e ao Professor Doutor Breno Ferraz de Oliveira por terem aceito me orientar durante a produção deste trabalho; à Professora Doutora Hatsumi Mukai pela coordenação do curso da disciplina de Monografia. Por último, mas não menos importante, gostaria de agradecer à minha mãe e ao meu pai por todo o apoio e suporte durante esses anos de graduação. Ao meu irmão pela ótima companhia ao longo de todos esses anos, à minha avó e aos meus tios por tudo o que tenho aprendido com eles.

A todos os meus amigos e amigas que estiveram ao meu lado durante a graduação, mas, em especial, a Giovanni Vonsowski Guarido, Laura Candolato, Maicon Luka, Leonardo Hadlich de Oliveira, Tiago Ferreira, Matheus Hodniuk, Fabio Rodrigues, Lucas Lima, Izabella Araújo, Gustavo Henrique Sahd e João Lucas Ferreira, por tornarem o meu dia a dia mais divertido, por me incentivarem e acreditarem em mim.

Por fim, gostaria de agradecer a todas as pessoas que, em algum momento, passaram pela minha vida, seja por um longo ou curto período de tempo, e me ajudaram a me desenvolver como ser humano, tornando-me uma pessoa melhor.

Resumo

Este trabalho tem como objetivo utilizar métodos de remoção de ruídos para promover o ensino e o entendimento das transformadas Rápida de Fourier (FFT) e Discreta de Fourier (DFT). A proposta se baseia em uma sequência didática voltada exclusivamente para o ensino superior, com foco na formação de professores que desejam aprofundar seus conhecimentos sobre essas técnicas e, conseqüentemente, instruir seus alunos de forma eficaz. A sequência didática visa não apenas apresentar os conceitos teóricos das transformadas de Fourier, mas também aplicar abordagens práticas que facilitam a aprendizagem ativa dos estudantes.

A proposta envolve o uso de recursos tecnológicos, como simulações computacionais e atividades experimentais, para que os alunos possam vivenciar de forma prática os conceitos abordados. Durante o processo, serão realizados exercícios e experiências em sala de aula que permitirão aos alunos aplicar a FFT e a DFT em situações reais, como a análise de sinais e imagens, e no estudo de fenômenos relacionados ao processamento de sinais.

A estrutura da sequência didática está organizada de maneira disciplinada, com uma abordagem que guiará o professor na aplicação dos métodos e na gestão do conteúdo durante as aulas. As atividades foram cuidadosamente planejadas para possibilitar um aprendizado significativo, levando em consideração as dificuldades comuns enfrentadas pelos estudantes ao aprender transformadas de Fourier.

Além disso, será fornecido ao professor um conjunto de diretrizes sobre como lidar com as diferentes formas de ensinar o conteúdo, respeitando o ritmo e as dificuldades dos alunos. O objetivo é que, ao final da sequência, tanto os alunos quanto os professores estejam capacitados a utilizar as transformadas de Fourier para resolver problemas reais da área de processamento de sinais, com um entendimento sólido dos conceitos matemáticos e das aplicações práticas dessas ferramentas.

Ao adotar uma metodologia ativa e focada em exemplos práticos, o trabalho visa aprimorar o ensino da FFT e DFT no ensino superior, transformando os alunos em agentes ativos no processo de aprendizagem e promovendo uma compreensão mais profunda e aplicada dos conceitos de Fourier.

Palavras chave: Física, Transformada Rápida de Fourier (FFT), Transformada Discreta de Fourier (DFT), Sequência Didática.

Introdução

Este trabalho tem como objetivo principal explorar a fundamentação teórica e prática das Transformadas Discreta de Fourier (DFT) e Rápida de Fourier (FFT), apresentando uma abordagem voltada para o ensino superior que integra teoria, programação e aplicação prática. Essas transformadas são ferramentas fundamentais no processamento e análise de sinais, permitindo a transição entre os domínios do tempo e da frequência, o que é essencial para diversas áreas como física, engenharia, ciência de dados e telecomunicações.

A sequência didática elaborada faz parte de um curso completo sobre análise e processamento de sinais digitais, sendo esta etapa específica direcionada para a implementação e análise da DFT e FFT. A linguagem C foi escolhida como ferramenta de implementação devido à sua eficiência computacional e à disponibilidade de bibliotecas específicas, como a FFTW. Essas bibliotecas não apenas facilitam a implementação dos algoritmos, mas também otimizam o desempenho computacional, permitindo que os alunos explorem conceitos como complexidade algorítmica e eficiência no processamento de sinais.

Por meio da programação em C, o objetivo é que alunos sejam introduzidos à prática de decompor sinais em suas componentes frequenciais, compreender a relação entre as bases trigonométricas e os sinais analisados, e aplicar essas técnicas em problemas reais. Um exemplo prático para ser abordado é a remoção de ruídos, onde as transformadas podem ser utilizadas para identificar e filtrar frequências indesejadas, destacando a relevância dessas ferramentas na análise de dados reais.

No entanto, é importante destacar que esta sequência didática foi apenas desenvolvida teoricamente e não foi aplicada em sala de aula. O objetivo é contribuir para o ensino acadêmico no nível superior, fornecendo uma base metodológica que possa ser utilizada como referência em cursos voltados para a análise e o processamento de sinais digitais.

Além disso, a sequência didática foi estruturada para oferecer uma abordagem clara e progressiva, começando pela introdução aos conceitos teóricos fundamentais e avançando para atividades práticas. A combinação de atividades teóricas e computacionais busca preparar os alunos para aplicar os conceitos de DFT e FFT em problemas complexos, promovendo uma aprendizagem significativa e duradoura.

A relevância deste trabalho está em sua capacidade de conectar os fundamentos matemáticos às aplicações práticas, demonstrando como as transformadas de Fourier são indispensáveis para a análise e o processamento de sinais em diversos contextos.

Capítulo 1

Fundamentação Teórica

Neste primeira sessão, será explorados o conceito da Série de Fourier, Transformada de Fourier, Transformada Discreta de Fourier (DFT) e a Transformada Rápida de Fourier (FFT). A Série de Fourier é a base para a análise de sinais periódicos, permitindo representá-los como uma soma de funções senoidais de diferentes frequências. A Transformada de Fourier expande esse conceito para sinais não periódicos, fornecendo uma visão contínua de suas componentes espectrais. Para sinais discretos, a Transformada Discreta de Fourier permite a análise no domínio das frequências, sendo amplamente empregada em áreas como processamento digital e compressão de dados. Por fim, a Transformada Rápida de Fourier otimiza os cálculos da TDF, reduzindo o custo computacional e viabilizando sua aplicação em sistemas de alta eficiência e em tempo real.

1.1 Série de Fourier

Série de Fourier é uma soma de funções trigonométricas, como senos e cossenos, que é usada para representar funções periódicas, mesmo que sejam complexas. Ela é amplamente utilizada para descrever fenômenos físicos que se repetem ao longo do tempo, como ondas sonoras, sinais elétricos e vibrações. Facilitando a visualização e manipulação de funções complexas [1]. Uma função periódica complexa é semelhante a uma função periódica real, pois se repete em intervalos iguais. No entanto, seus valores são números complexos, contendo uma parte real e uma parte imaginária. Um exemplo clássico é a função exponencial complexa:

$$g(x) = e^{ix}. \quad (1.1)$$

A Série de Fourier usa números complexos porque facilita a representação de funções periódicas de maneira mais compacta e eficiente. Isso ocorre porque funções senoidais, como seno e cosseno, podem ser expressas usando a fórmula de Euler, que relaciona exponenciais complexas com senos e cossenos:

$$e^{ix} = \cos(x) + i\sin(x). \quad (1.2)$$

Essa relação é eficaz porque permite que a soma de senos e cossenos seja representada como uma soma de exponenciais complexas. A parte real da exponencial representa o cosseno, e a parte imaginária, o seno. Sendo x um número real e i a unidade imaginária, com $i^2 = -1$. Os números complexos permitem que tanto a amplitude quanto a fase de cada componente senoidal sejam expressas de maneira simples. Essa abordagem facilita cálculos matemáticos e unifica o tratamento de fases e amplitudes, especialmente em fenômenos físicos e no processamento de sinais. Assim, a inclusão de números complexos torna a matemática das Série de Fourier mais eficiente e clara na análise de sinais periódicos. Amplitude é o valor máximo que a onda atinge em relação ao seu ponto de equilíbrio. Em termos simples, a amplitude determina o "tamanho" ou a "intensidade" da onda. A fase e amplitude são conceitos fundamentais em ondas e sinais, particularmente no contexto de funções sinusoidais, como seno e cosseno, usadas na Transformada de Fourier. A fase, por sua vez, refere-se ao deslocamento da onda em relação a um ponto de referência no tempo ou no espaço. Ela indica onde a onda está dentro de seu ciclo. É medida em graus ou radianos. Na equação:

$$A \cos(\omega t + \phi) \quad (1.3)$$

A amplitude A é a altura máxima de uma onda, representando a magnitude de sua oscilação vertical e, conseqüentemente, o volume do som associado. A fase ϕ , por outro lado, influencia o deslocamento horizontal da onda, alterando o ponto onde ela inicia seu ciclo. A fase define o atraso ou adiantamento de uma onda em relação a outra, além de indicar como diferentes componentes de frequência estão alinhadas. Quando ondas com frequências ou fases diferentes se encontram, ocorre o fenômeno de superposição. Nesse processo, a amplitude resultante em cada ponto é determinada pela soma das amplitudes das ondas individuais naquele mesmo ponto. Esse processo pode gerar formas de onda mais complexas que as senoidais simples. Na interferência construtiva, as cristas e vales de duas ondas coincidem, somando suas amplitudes e formando uma onda maior. Já na interferência destrutiva, a crista de uma onda coincide com o vale de outra, resultando em cancelamento parcial ou total, o que reduz a amplitude ou a anula completamente.

O deslocamento vertical, por sua vez, é causado por um termo constante adicionado à função, como em:

$$A \cos(\omega t + \phi) + C \quad (1.4)$$

Em que C é o deslocamento vertical. Que move toda a onda verticalmente sem alterar a amplitude ou a frequência. Se C for positivo, a onda se move para cima, se C for negativo, a onda se move para baixo.

A função é periódica com o período 2π , ou seja:

$$e^{i(x+2\pi)} = e^{ix} \quad (1.5)$$

O período 2π está diretamente relacionado à periodicidade das funções trigonométricas seno e cosseno, como indicado pela fórmula de Euler.

Ambas as funções são periódicas com período 2π , o que significa que, ao adicionar 2π a qualquer valor de x , os valores de $\cos(x)$ e $\sin(x)$ retornam ao mesmo ponto. Assim:

$$\cos(x + 2\pi) = \cos(x) \quad (1.6)$$

$$\sin(x + 2\pi) = \sin(x) \quad (1.7)$$

Portanto, a função exponencial complexa também se repete com o mesmo período, porque:

$$e^{i(x+2\pi)} = \cos(x + 2\pi) + i\sin(x + 2\pi) = \cos(x) + i\sin(x) = e^{ix} \quad (1.8)$$

Essa periodicidade é uma propriedade fundamental de funções exponenciais complexas, conforme mostrado na equação (1.1), que envolvem combinações de senos e cossenos. Nesse contexto, a função periódica complexa é formada pela combinação da parte real, $\cos(x)$, e da parte imaginária, $\sin(x)$, ambas também periódicas [2]

Uma característica importante dessas funções é que elas se repetem continuamente, tanto para valores positivos quanto negativos da variável independente, refletindo sua natureza infinita. Essa propriedade fundamenta a forma geral da Série de Fourier, apresentada a seguir:

$$g(x) = a_0 + \sum_{n=1}^{\infty} \left[a_n \cos\left(\frac{2\pi nx}{T}\right) + b_n \sin\left(\frac{2\pi nx}{T}\right) \right] \quad (1.9)$$

Na equação, n é um número inteiro positivo que representa os harmônicos da função periódica $g(x)$. Cada harmônico está associado a uma frequência que é um múltiplo inteiro da frequência fundamental da função. Dessa forma, $n = 1$ corresponde à frequência fundamental, $n = 2$ representa o primeiro harmônico, $n = 3$ o segundo harmônico, e assim por diante.

Já T é o período da função $g(x)$, que é o intervalo de repetição da função no domínio do tempo. Em termos simples, T define a duração de um ciclo completo da função. Os coeficientes $\frac{2\pi n}{T}$ determinam as frequências angulares dos componentes senoidais e cossenoidais da Série de Fourier. O período pode ser escrito como o inverso da frequência:

$$T = \frac{1}{f} \quad (1.10)$$

E a frequência pode ser escrita como o inverso do período:

$$f = \frac{1}{T} \quad (1.11)$$

Os coeficientes a_0 , a_n e b_n são números que dependem da função a ser representada, com o período fundamental $2T$. Esses coeficientes correspondem às amplitudes de cada onda na série e são determinados pelas seguintes fórmulas. O período fundamental é $2T$ porque a Série de Fourier é construída a partir de funções trigonométricas, como seno e cosseno, que possuem periodicidade intrínseca de 2π . Quando essas funções são ajustadas para um intervalo de tempo T , a periodicidade total na Série de Fourier se estende para $2T$, cobrindo o ciclo completo de todas as contribuições harmônicas.

$$a_0 = \frac{1}{T} \int_c^{c+2T} f(x) dx$$

$$a_n = \frac{2}{T} \int_c^{c+2T} f(x) \cos\left(\frac{n\pi x}{T}\right) dx$$

$$b_n = \frac{2}{T} \int_c^{c+2T} f(x) \sin\left(\frac{n\pi x}{T}\right) dx$$

Sendo c qualquer constante real que define o ponto inicial do intervalo de integração. O coeficiente a_0 é o valor médio da função ao longo de um período, ou seja, a parte constante da função, que pode ser vista como a amplitude da componente de frequência zero, essa componente não oscila, ou seja, não tem variação periódica, pois corresponde a uma frequência de zero Hz. Em alguns casos esse valor médio pode ser diferente de zero, isso acontece quando a função tem um deslocamento vertical ao longo do eixo y . Por exemplo, se a função oscila em torno de uma linha horizontal acima do eixo $y = 0$, o valor médio será positivo. Se oscila abaixo dessa linha, o valor será negativo. Um caso típico é da seguinte função:

$$g(t) = \text{sen}(t) + 2. \quad (1.12)$$

A função (1.12) possui valor médio igual a 2, pois todas as oscilações ocorrem em torno de $y = 2$ em vez de $y = 0$. Portanto o coeficiente a_0 neste caso é 2. Isso pode ser observado na figura a seguir.

Os coeficientes a_n e b_n ajustam a altura de cada onda associada a uma frequência n . Essas amplitudes dizem quanto "peso" ou influência de cada componente senoidal ou cossenoidal na reconstrução da função original.

O cálculo dos coeficientes a_n e b_n por meio de integração mede a correlação entre a função original e as funções seno e cosseno em diferentes frequências. Quando a função original tem uma forte correspondência com uma onda de determinada frequência, os coeficientes a_n ou b_n apresentam valores mais elevados, indicando que essa frequência possui maior amplitude na

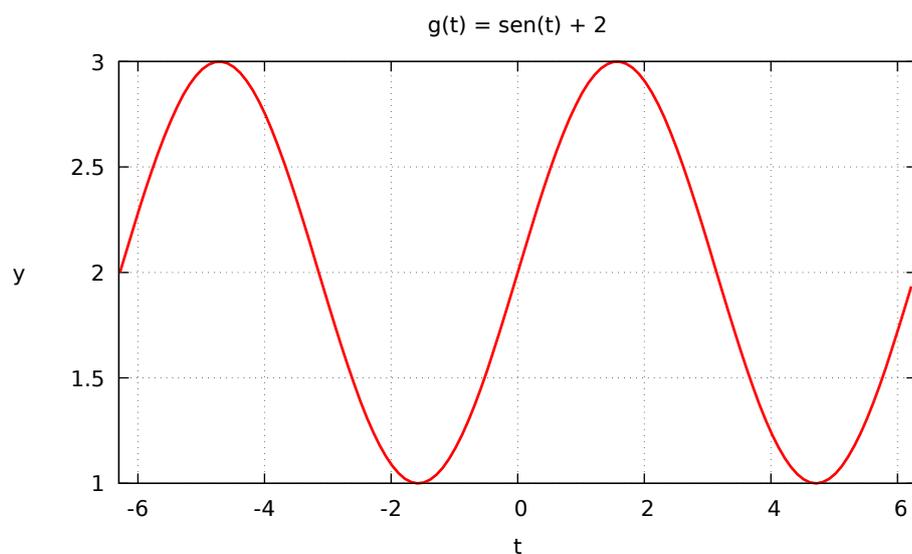


Figura 1.1: Gráfico da função $\text{sen}(t) + 2$ mostrando o deslocamento vertical de 2 unidades.

composição do sinal. Por exemplo:

$$g(t) = 3\text{sen}(t). \quad (1.13)$$

A função (1.13) possui uma amplitude de 3 unidades, e essa informação estaria diretamente associada ao coeficiente b_n na Série de Fourier, dado que a função é representada exclusivamente por um termo seno. Esses coeficientes calculam a contribuição de cada seno ou cosseno na função original [3].

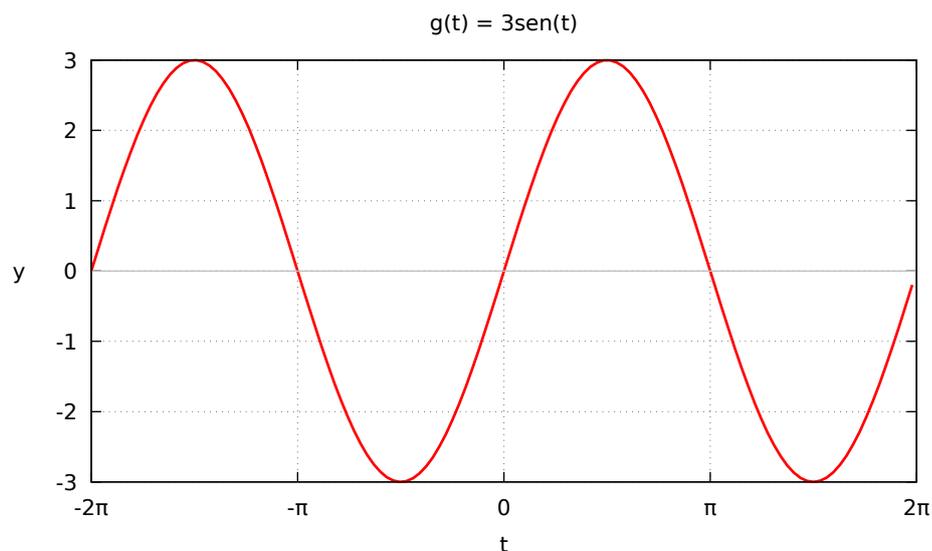


Figura 1.2: Gráfico da função $3\text{sen}(t)$ mostrando a amplitude que varia de 3 à -3.

A Série de Fourier desempenha um papel crucial em diversas áreas de tecnologia, é usada na compactação de dados digitais para reduzir o tamanho de arquivos, removendo informações

menos importantes no domínio da frequência. Em áudio e vídeo, frequências que o ouvido ou os olhos não percebem facilmente são descartadas. No carregamento rápido de imagens, como no formato JPEG, a imagem é transformada em frequências, e apenas as mais importantes são mantidas para reconstrução rápida, reduzindo o tamanho do arquivo e acelerando o processo.

Ruído é qualquer interferência ou variação indesejada em um sinal, que distorce ou dificulta a transmissão ou interpretação correta da informação. Ele pode ocorrer em diversas formas, com sons indesejados e gravações de áudio, distorções visuais em imagens, ou erros em dados transmitidos em sistemas de comunicação.

Essa decomposição facilita a análise, compressão e processamento de sinais ao identificar as frequências dominantes e permitir o descarte de componentes de menor relevância. Uma vez que os componentes fundamentais podem ser manipulados de maneira simples e eficiente. Ao realizar essa decomposição, torna-se possível comprimir dados, Mantendo apenas as frequências com maior contribuição para o sinal original e descartando aquelas de menor relevância ou impacto significativo, que não afetam significativamente a qualidade percebida. Esse processo de compactação não só reduz o volume de dados, mas também acelera o processamento, tornando-o mais eficiente, especialmente em aplicações onde a transmissão rápida e o armazenamento eficiente de dados são essenciais, como em mídias digitais e telecomunicações.

1.2 Transformada de Fourier

A Transformada de Fourier é uma transformada integral que representa uma função em termos de funções de bases sinusoidal. Ao contrário da Série de Fourier, que expressa funções periódicas como uma soma de senos e cossenos com frequências discretas, a Transformada de Fourier utiliza uma integral para decompor funções, periódicas ou não, em componentes frequenciais. Esse processo é especialmente útil para sinais complexos, como o som de um instrumento musical, que pode ser descrito como uma soma de frequências. Diferente de uma função periódica, uma função não periódica não se repete em intervalos regulares, como ilustrado em (1.14), onde a frequência do sinal varia continuamente.

Uma função é não periódica quando não se repete em nenhum intervalo regular. Um exemplo é uma função seno modulado por uma função linear:

$$g(t) = \text{sen}(t^2) \quad (1.14)$$

A função (1.14) não é periódica porque o valor t^2 cresce à medida que t aumenta, fazendo com que a frequência da oscilação varie continuamente. Em $\text{sen}(t)^2$ a frequência muda constantemente, tornando impossível a repetição regular dos ciclos. Assim, a função não possui um período fixo e não se repete em intervalos regulares.

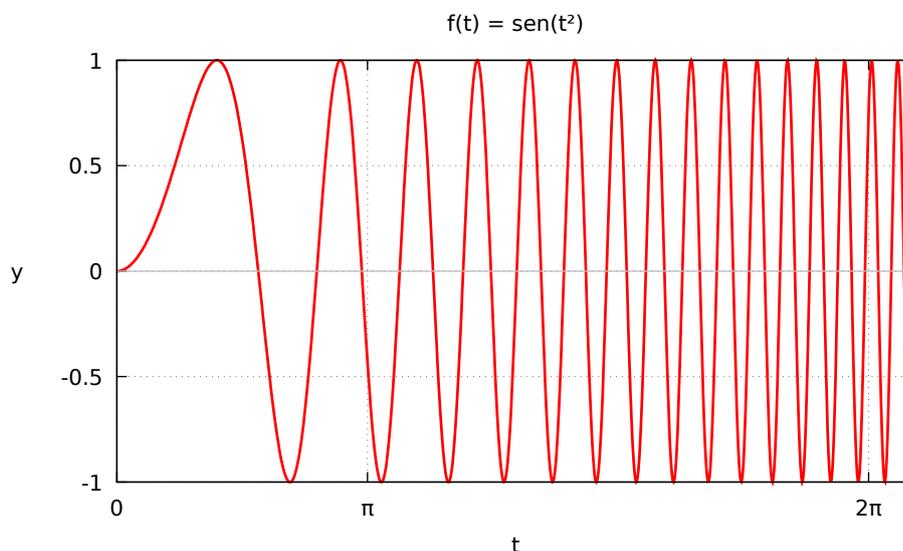


Figura 1.3: Gráfico da função $\text{sen}(t^2)$ mostrando que a função não se repete em intervalos regulares.

Quando combinamos funções cujas frequências não são racionais entre si, ou seja, que não são múltiplos inteiros uma das outras, a soma dessas funções não se repete em intervalos regulares, resultando em função não periódica. Isso ocorre porque os ciclos dessas funções nunca se alinham de forma estável, o que faz com que a função resultante varie continuamente sem repetição.

A Transformada de Fourier é conhecida como representação do domínio da frequência do sinal original. O termo Transformada de Fourier refere-se tanto à forma como a função é representada no domínio das frequências quanto à operação matemática que a converte a partir de uma função temporal. A Transformada de Fourier não se aplica somente a funções temporais; por questões de padronização, é comum referir-se ao domínio original como domínio do tempo.

Uma das vantagens práticas dessa transformada é que algumas operações, como filtragem ou análise de componentes específicas, podem ser mais fáceis de realizar no domínio da frequência do que no domínio do tempo. Por exemplo, identificar e remover ruídos ou destacar certas frequências em um sinal pode ser mais direto ao analisar o espectro de frequências, em vez de manipular diretamente a forma do sinal no tempo. Assim, operações lineares aplicadas em um dos domínios, tempo ou frequência, resultam em efeitos correspondentes no outro, otimizando o processamento e a análise de sinais.

A derivada de uma função no domínio do tempo corresponde à multiplicação de sua Transformada de Fourier pela frequência no domínio da frequência. Além disso a convolução no domínio do tempo condiz à multiplicação no domínio da frequência. Isso significa que qualquer sistema linear e invariável no tempo, como um filtro aplicado a um sinal, pode ser analisado de forma mais simples no domínio da frequência. Nesse contexto, as operações realizadas no domínio do tempo correspondem a manipulações diretas das frequências, facilitando a compreensão e o processamento do sinal.

Após encontrar a frequência almejada, a transformação do resultado alterna para o domínio do tempo. A análise harmônica é o estudo comedido da relação entre os domínios de frequência e tempo, abrangendo diferentes funções ou operações que são mais "simples" em um ou em outro. A Transformada de Fourier de uma função $g(t)$ denotada por $\mathcal{F}\{g(t)\} = g(f)$, é dada por:

$$g(f) = \int_{-\infty}^{\infty} g(t) e^{-i2\pi ft} dt \quad (1.15)$$

onde:

- $g(f)$ é a magnitude da frequência;
- $g(t)$ é a função original no domínio do tempo;
- f é a frequência;
- i é a unidade imaginária, com $i^2 = -1$;
- $e^{-i2\pi ft}$ é o núcleo da transformação, representando componentes senoidais de diferentes frequências.

A Transformada de Fourier é uma ferramenta fundamental na análise de sinais complexos, pois permite decompor qualquer função de tempo em suas frequências componentes.

Esse processo é essencial em diversas áreas, como engenharia, física, e ciência da computação, facilitando a identificação de frequências específicas em sinais que possam conter ruído ou sobreposições, o que simplifica o processamento de áudio e ruído [4].

Nesses processamentos, a Transformada de Fourier facilita a análise de frequências presentes em um sinal, permitindo operações como filtragem e compressão. Ruído, interferência ou informação indesejada que influencia os sinais, e compreender seu impacto é crucial para entender a utilidade da Transformada de Fourier. Os ruídos podem ser isolados e reduzidos, simplificando a análise do sinal, permitindo operações como filtragem. A decomposição do sinal facilita a eliminação de ruídos indesejados. Ruído representa uma forma de ruído se manifesta como componentes adicionais que se misturam ao sinal desejado, tornando difícil observar as características genuínas desse sinal.

A Transformada se torna, então, uma ferramenta valiosa para identificar e isolar o ruído, já que ao decompor um sinal em suas frequências componentes, ela revela as frequências onde o ruído pode estar presente e permite que seja tratado de forma mais eficaz. Ruídos podem ter origens variadas, desde o ambiente até dispositivos eletrônicos e oscilações naturais. Essa diversidade de fontes ajuda a contextualizar o motivo pelo qual ruídos são comuns em sinais e afetam a análise de suas frequências. De todos os tipos de ruído, o ruído branco é um dos mais frequentes.

Em processamento de sinal, o ruído branco é um tipo de interferência que tem a mesma intensidade em todas as frequências, o que significa que ele aparece de maneira uniforme ao longo de todo o espectro de frequências. É o que vemos em uma televisão fora de sinal com aquela imagem de pontos brancos e pretos, essa representação visual tanto visual quanto sonora, ocorre porque o sinal está preenchido de frequências aleatórias e de igual intensidade.

Além do processamento de áudio, a Transformada de Fourier também é amplamente aplicada no processamento de imagens. Nesse contexto, é usada para transformar uma imagem do domínio espacial para o domínio da frequência, observamos *pixels* e disposição de intensidades, para o domínio da frequência, em que cada ponto representa uma frequência específica. Essa transformação é essencial em diversas operações, como remoção de ruídos, compressão, e realce de bordas. A decomposição de uma imagem em frequência permite filtrar componentes de alta ou baixa frequência, facilitando a eliminação de ruídos.

No caso de ruídos de alta frequência, como a granulação de uma foto, o filtro isolará essas frequências para reduzir o efeito e tornar a imagem mais suave. Outra aplicação significativa é a compressão de imagem. O método JPEG, por exemplo, utiliza a Transformada de Fourier para representar as frequências de uma imagem de maneira mais compacta, removendo informações redundantes sem comprometer visivelmente a qualidade, uma técnica vantajosa especialmente para armazenar transmissões de dados. Além disso, a Transformada de Fourier é útil para detecção de bordas e detalhes.

As bordas de uma imagem correspondem a mudanças bruscas de intensidade de cor, que se manifesta como frequências altas no espectro. Ao aplicar a Transformada de Fourier,

conseguimos isolar essas frequências e realçar os contornos, ajudando a destacar objetos ou áreas de interesse em uma imagem. Um conceito relacionado ao uso da Transformada de Fourier é a densidade espectral de potência (PSD), que mede como a potência do sinal se distribui entre as diferentes frequências.

Para um sinal, a PSD indica onde a energia é mais intensa e quais frequências contribuem mais para a composição do sinal. Em uma imagem, por exemplo, a PSD permite identificar texturas finas ou grandes áreas suaves, possibilitando ajustar ou manipular essas regiões com precisão. Essa análise é fundamental para a manipulação eficiente de sinais, oferecendo uma visão detalhada de como cada componente frequencial contribui para o resultado final da imagem. Uma função simples, é a exponencial decrescente. Esse tipo de função é comum em fenômenos que decaem ao longo do tempo, como o decaimento radioativo ou amortecimento de um sinal. A função que vamos usar é:

$$g(t) = (e^{-\alpha t}) \quad (1.16)$$

- para $g(t) = e^{-\alpha t}$ para $t \geq 0$;
- para $g(t) = 0$ para $t < 0$.

em que α é uma constante positiva que controla a taxa de variação. Substituindo a função (1.16) na equação (1.15) temos:

$$g(f) = \int_0^{\infty} e^{-\alpha t} e^{-i2\pi f t} dt \quad (1.17)$$

O limite de integração foi escrito de 0 até infinito porque para todo e qualquer valor negativo de t a função (1.16) tem valor nulo, a base das duas funções são iguais é possível escrever a expressão (1.17) da seguinte maneira:

$$g(f) = \int_0^{\infty} e^{-(\alpha+i2\pi f)t} dt$$

Essa é uma integral exponencial simples, e sabemos que a integral de e^{bt} de zero a infinito é $\frac{1}{b}$, desde que a parte real b seja positiva. Dessa maneira temos:

$$g(f) = \int_0^{\infty} e^{-(\alpha+i2\pi f)t} dt = \left[\frac{1}{\alpha + i2\pi f} \cdot e^{-(\alpha+i2\pi f)t} \right]_0^{\infty}$$

Calculando os limites:

- Quando $t \rightarrow \infty$ o termo $e^{-(\alpha+i2\pi f)t} \rightarrow 0$, pois α é positiva.
- $t = 0$, o termo $e^{-(\alpha+i2\pi f)t} = 1$.

Portando, a transformada de Fourier da função (1.16) para para $t \geq 0$ é:

$$g(f) = \frac{1}{\alpha + i2\pi f} \quad (1.18)$$

Esse resultado mostra que a função exponencial decresce é transformada em uma função complexa dependente da frequência f . A parte imaginária indica uma mudança de fase, e a magnitude de f da $g(f)$ descreve a distribuição das frequências no sinal original, ou seja, o domínio do tempo antes de aplicar a transformada. O computador, ao contrário de cálculos analíticos, não consegue lidar com um intervalo que vá de menos infinito até infinito, para isso é feito uma discretização: dividimos o intervalo em um número finito de pontos igualmente espaçados [5].

A DFT então analisa essa série de valores discretos em vez de uma função contínua, permitindo a análise de sinais digitais. Quando se fala ruído em matemática, estamos nos referindo a valores ou variações aleatórios que "contaminam" o sinal principal. Em processamento de sinais, ruído pode representar interferências ou flutuações imprevisíveis. A Transformada Discreta de Fourier é eficaz para diferenciar o sinal principal dos componentes aleatórios, facilitando a filtragem e o processamento de dados.

1.3 Transformada Discreta de Fourier

Em matemática a DFT converte uma sequência finita de amostras igualmente espaçadas no tempo, valores discretos são medições ou amostras específicas do sinal contínuo, capturadas em intervalos regulares do tempo. em uma nova sequência de amostras no domínio da frequência, ambas com o mesmo número de pontos. A Transformada Discreta de Fourier (DFT) produz uma função complexa, que representa os componentes de frequência de sinal original.

O intervalo de amostragem na DFT é inversamente proporcional ao tempo de duração da sequência de entrada. Se a sequência original contém todos os dados os valores diferentes de zero de uma função, a DFT gera uma representação periódica no domínio da frequência, com amostras discretas de um único ciclo. Se a sequência original for um ciclo completo de uma função periódica, a DFT fornece todos os valores diferentes de zero desse ciclo no domínio da frequência.

A DFT é uma das mais importantes, amplamente utilizadas em aplicações práticas de análise de Fourier. No processamento de sinais digitais, ela permite transformar dados que variam ao longo do tempo, como o som, sinais de rádio e medições de temperatura em intervalos indefinidos. Em processamento de imagens, a DFT trabalha com valores de *pixels* de linhas ou colunas da imagem.

Além disso, DFT é usada para resolver equações diferenciais de forma eficiente em operações como convoluções e multiplicação de grandes números inteiros. Por lidar com uma quantidade finita de dados, a Transformada Discreta de Fourier pode ser implementada em computadores usando algoritmos eficientes chamados Transformada Rápida de Fourier (FFT). A DFT tem várias aplicações, incluindo uso puramente matemáticos, sem necessidade de interpretações física [6].

No entanto, no processamento de sinais, no entanto ela pode ser vista como uma versão discreta da Transformada de Fourier em Tempo Discreto, é uma função contínua e não periódica. A DFT calcula N amostras igualmente espaçadas de um ciclo da Transformada de Fourier em Tempo Discreto. A DTFT é uma versão da Transformada de Fourier para sequências discretas, obtidas ao amostrar uma função contínua em intervalos regulares. Ela representa essas sequências em uma faixa finita de frequências, gerando uma função periódica.

Diferente da DFT, que calcula apenas um conjunto finito de amostras, a Transformada de Fourier em Tempo Discreto fornece uma visão contínua e completa no domínio da frequência para dados discretos, sendo útil em área digital e cálculo numérico.

A DTFT transforma uma sequência de N números complexos.

$X_k = X_0, X_1, \dots, X_{N-1}$ em outra sequência de números complexos,

$X_k = \{X_0, X_1, \dots, X_{N-1}\}$, que é definido como:

$$g(f) = \frac{1}{N} \sum_{n=0}^{N-1} g(t_n) e^{-i \frac{ft_n}{N}} \Delta t \quad (1.19)$$

A transformação às vezes é denotada pelo símbolo G , como em $X = GX$, $G(X)$ ou GX . A equação pode ser interpretada de várias maneiras, por exemplo:

- o Δt na equação da DTFT representa o intervalo entre amostras no domínio do tempo. Em muitos casos práticos, quando os dados são amostrados com uma taxa de amostragem uniforme e Δt pode ser simplificado para 1. Ou seja, se as amostras são tomadas a intervalos regulares de tempo, não há necessidade de ajustar o valor de Δt na equação. Isso simplifica a equação. Neste caso, $\Delta t = 1$.
- Descreve a DTFT para uma sequência periódica de N pontos, composta apenas por componentes de frequências discretas;
- Ela também pode fornecer amostras igualmente espaçadas da DTFT contínua de uma sequência finita;
- É a comparação entre a sequência de entrada $x(n)$ e uma senoide complexa de frequência $\frac{k}{N}$. Dessa forma, ela atua como um filtro seletivo para essa frequência;
- É a versão discreta da fórmula dos coeficientes de uma Série de Fourier de uma função periódica $x(t)$ com período P . Esses coeficientes, C_k , representam as amplitudes das componentes de frequência na representação da função $x(t)$ no domínio da frequência, e a expressão é dada por:

$$C_k = \frac{1}{P} \int_P x(t) e^{-i2\pi \frac{k}{P} t}$$

Esses coeficientes permitem decompor $x(t)$ em uma soma de senoides e cossenoides com diferentes frequências e amplitudes, representando a contribuição de cada frequência na função $x(t)$.

A DFT é dada por:

$$g(f) = \sum_{n=0}^{N-1} g(t) e^{-i \frac{2\pi}{N} f t}, \quad (1.20)$$

onde:

- $g(f)$ é a transformada no índice de frequência;
- $g(t)$ é o valor do sinal no domínio do tempo;
- N é o número total de pontos;
- $e^{-i \frac{2\pi}{N} k n}$ é o fator de base exponencial complexo;
- f representa a frequência.

Quando a DFT é usada para análise de sinais, a sequência x_n representa amostras de tempo igualmente espaçadas de um sinal contínuo de $x(t)$ [7]. Essa transformação de dados contínuos para discretos gera uma versão discreta da Transformada de Fourier, chamada de Transformada de Fourier em Tempo Discreto (DTFT). Para compensar as limitações introduzidas pela discretização, fenômenos como *aliasing* e vazamento podem ocorrer durante o processo, tornando essencial o uso de técnicas apropriadas. O *aliasing* acontece quando a taxa de amostragem é insuficiente para capturar todas as informações do sinal original. Para evitar esse problema, é fundamental adotar uma taxa de amostragem adequada, como a taxa de *Nyquist*, que garante uma representação fiel do sinal. O vazamento, por outro lado, surge quando a análise usa uma sequência curta, reduzindo a resolução da Transformada de Fourier em Tempo discreto (DTFT).

A taxa de amostragem indica quantas vezes por segundo um sinal analógico é medido para serem convertidos em sinal digital, sendo expressas em *Hertz* (Hz). Quanto maior a taxa de amostragem, mais detalhado será o registro do sinal, e a representação digital se aproxima mais do sinal original.

O sinal analógico é um sinal contínuo e varia ao longo do tempo sem interrupções. Ele pode ser representado como uma curva que percorre todos os valores em um determinado intervalo. O sinal digital é menos complexo e possui valores discretos tanto no tempo quanto na amplitude. Em vez de variar continuamente, ele assume valores específicos dentro de um intervalo de tempo, sem incluir intermediários. Dessa forma, o sinal digital elimina variações suaves, arredondando os valores fracionados para inteiro mais próximo [7].

O teorema de amostragem de Nyquist-Shannon é um conceito central na teoria de informação, especialmente nas telecomunicações e no processamento de sinais. Ele explica o processo de amostragem, que é quando transformamos um sinal contínuo, como uma onda de som, um sinal de TV ou de rádio, em uma sequência de valores numéricos, criando uma versão discreta do sinal no tempo ou espaço. A versão apresentada por Shannon do teorema diz que:

”Seja um sinal, limitado em banda, e seu intervalo de tempo dividido em partes iguais, de forma que se obtenham intervalos tais que, cada subdivisão compreenda um intervalo com período T segundos, onde T é menor do que $\frac{f_m}{2}$, e se uma amostra é tomada instantânea é tomada arbitrariamente de cada subintervalo, então o conhecimento da amplitude instantânea de cada amostra somando ao conhecimento dos instantes que é tomada a amostra de cada subintervalo contém toda a informação do sinal original.”

O teorema nos diz que um sinal analógico com uma faixa de frequências limitada pode ser recuperado perfeitamente a partir de uma sequência infinita de amostras, desde que a taxa de amostragem seja maior que $2f_m$ amostras por segundo, onde f_m é a frequência mais alta do sinal original. No entanto, se o sinal tiver uma componente exatamente em f_m hertz e for amostrado com intervalos de $\frac{1}{2f}$ segundos, ele não poderá ser totalmente recuperado. Interpretações mais recentes do teorema esclarecem que ele exige que o sinal $x(t)$ não tenha frequências maiores ou iguais a f_m . Essas condições evita problemas quando uma função possui componente senoidal

estável exatamente na frequência f_m , como Shannon observou.

O teorema é uma idealização do mundo real, pois só se aplica a sinais amostrados por tempo infinito [8]. Na prática, um sinal $x(t)$ com duração limitada não pode ser perfeitamente limitado em banda. Ainda assim, é possível recuperar o sinal de forma próxima ao original, o que é suficiente para maioria dos casos.

O teorema também oferece uma fórmula para reconstruir o sinal original. Ele ajuda a entender o fenômeno do *aliasing* que ocorre quando a taxa de amostragem não atende às condições estabelecidas pelo teorema.

Um sinal ou função é considerado limitado em banda quando não possui energia em frequências acima de um determinado limite, chamado de largura de banda B . A variação de um sinal no tempo está diretamente relacionada à sua largura de banda, determinando o nível de detalhes que ele pode transmitir em um intervalo de tempo.

O teorema da amostragem garante que é possível representar completamente um sinal com amostras discretas uniformemente espaçadas, desde que sua largura de banda seja menor que metade de um sinal contínuo no tempo. Seja $g(t)$ a representação de um sinal contínuo no tempo e seja $g(f)$ como sua Transformada de Fourier, o sinal $g(t)$ é limitado em banda, B , se:

$$g(x) = 0 \quad \text{para qualquer } |f| > B$$

Para que seja possível reconstruir um sinal de forma exata a partir de amostras tomadas em uma taxa uniforme de amostragem f_s , em amostras por unidade de tempo, é necessário atender à seguinte condição suficiente:

$$f_s > 2B,$$

ou, de maneira equivalente:

$$B < \frac{f_s}{2}.$$

$2B$ é chamada de Taxa de Nyquist à propriedade de um sinal limitado em banda, enquanto $f_s/2$ é conhecido como Frequência de Nyquist, uma característica do sinal de amostragem. O intervalo entre duas amostras consecutivas é chamado de *intervalo de amostragem*:

$$T = \frac{1}{f_s},$$

e as amostras de $g(t)$ são:

$$g(nT), \quad n \in \mathbb{Z}$$

O Teorema da Amostragem descreve dois processos essenciais no processamento de sinais:

- Amostragem:

Transforma um sinal contínuo no tempo $g(t)$ em um sinal discreto $g[n]$ ao medir seus valores em intervalos regulares T , chamados de intervalos de amostragem.

A frequência de amostragem $f_s = \frac{1}{T}$, expressa em *Hertz* (Hz), representa um número de amostras por segundo. Esse processo gera uma sequência de valores que representam um sinal original.

- Reconstrução:

A partir dessas amostras, o sinal contínuo é recuperado usando interpolação, geralmente com a seguinte função:

$$\text{sinc}(x) = \begin{cases} \frac{\sin(\pi x)}{\pi x}, & \text{se } x \neq 0, \\ 1, & \text{se } x = 0. \end{cases}$$

Esse método combina as amostras com funções *sinc* escaladas e deslocadas. Esse método combina as amostras das funções *sinc* escaladas e deslocadas, cujos picos coincidem com instantes de amostragem, somando-as para formar o sinal contínuo. Apesar de idealizada e inviável na prática por requerer infinitos termos, a reconstrução com a *sinc* é a única que recupera o sinal original exatamente, desde que o sinal seja limitado em banda.

Condições do teorema:

- O sinal contínuo só pode ser recuperado perfeitamente se sua frequência máxima f_m for menor que a metade da frequência de amostragem $\frac{f_s}{2}$, o chamado critério de Nyquist;
- Se houver frequências iguais ou maiores que $\frac{f_s}{2}$, ocorrerá *aliasing*, impossibilitando a recuperação exata do sinal.

Na prática, intervalos de amostragem são pequenos, e a reconstrução ideal é uma aproximação precisa, suficiente para a maioria das aplicações reais.

O teorema tem algumas consequências importantes:

- Se a maior frequência B no sinal original é conhecida, o teorema define a menor frequência de amostragem necessária para garantir a reconstrução perfeita. Essa frequência de amostragem necessária para garantir a reconstrução perfeita. Essa frequência mínima, $2B$, é chamado de Taxa de Nyquist.
- Se a frequência de amostragem f_s é conhecida, o teorema estabelece o limite máximo para as frequências presentes no sinal, $B < \frac{f_s}{2}$, garantindo a reconstrução perfeita. Esse limite é chamado de Frequência de Nyquist (f_N).

Para um sinal que possa ser amostrado corretamente, ele precisa ter suas frequências limitadas, ou seja, qualquer frequência acima de um determinado valor deve ser zero ou suficientemente pequena para ser ignorada. No caso de sinais com um modelo conhecido, é possível garantir essa limitação analisando suas componentes de frequência. Por exemplo, a voz humana normalmente possui frequências abaixo de 10 Hz, o que torna uma frequência de amostragem de 20 Hz suficiente para capturar esse tipo de som. Já para sinais sem um modelo claro, é necessário usar um filtro passa-baixas para limitar as frequências antes da amostragem. Por exemplo, se a amostragem foi feita a 8 Hz, o

sinal deve ser filtrado para remover frequências acima de 4 Hz, garantindo que elas não interfiram na reconstrução do sinal original.

Na prática, é impossível cumprir perfeitamente as condições do teorema ou implementar a fórmula ideal de reconstrução. O processo ideal usa funções *seno* deslocadas e ajustadas, mas isso não é viável porque exige somar infinitos termos, já que cada amostra influencia o sinal em todos os pontos do tempo.

Em vez disso, utiliza-se uma aproximação prática com funções *sinc* de comprimento finito, o que gera um pequeno erro chamado erro de interpolação. Conversores digitais para analógicos reais não produzem funções *sinc* nem impulsos ideais (como os delta de Dirac), mas sim pulsos retangulares ajustados. Esse comportamento pode ser representado por um filtro simples, chamado "filtro *hold* de ordem zero", que é alimentado por uma sequência de pulsos ajustados e atrasados.

Na prática é impossível limitar um sinal perfeitamente em banda, e filtros ideais, conhecido como "muro de tijolos", não podem ser implementados. Filtros reais conseguem apenas reduzir as frequências fora de um intervalo específico, mas não eliminá-las completamente. Além disso, nenhum sinal no domínio do tempo pode ser perfeitamente limitado em banda. Por isso, mesmo que fosse possível realizar uma reconstrução ideal, o sinal resultante ainda seria diferente do original. Esse desvio, causando pela impossibilidade de limitar completamente as frequências, é conhecida como *aliasing*.

O teorema de amostragem não especifica o que acontece quando as condições ideais não são seguidas, mas oferece uma base para analisar esses casos. Quem projeta sistemas que envolvem amostragem e reconstrução precisa entender bem o sinal a ser trabalhado, especialmente suas frequências, a frequência de amostragem, o método de reconstrução (interpolação) e impacto de erros, como *aliasing*, amostragem e interpolação [9]. Todos esses fatores precisam ser ajustados com cuidado para que o sistema funcione de forma eficiente.

Quando a frequência de amostragem não é o suficiente, cópias das frequências originais do sinal acabam se sobrepondo, Isso cria o chamado *aliasing*, em que frequências altas parecem ser frequências baixas. Nesse caso, não é possível recuperar corretamente o sinal original, e a reconstrução resulta em frequências alteradas ou erradas. Frequências alteradas ou erradas. Frequências iguais à metade da taxa de amostragem podem até ser recuperadas, mas com alterações de amplitude de fase.

Para evitar *aliasing*, é necessário garantir que a frequência de amostragem seja pelo menos o dobro da maior frequência do sinal. Caso o sinal não seja limitado em frequência de amostragem adequada. Embora nenhum filtro seja perfeito e sempre permita algum vazamento de altas frequências, esses efeitos podem ser pequenos o suficiente para não causar problemas significativos.

A Transformada Discreta de Fourier desempenha um papel fundamental no processamento digital de imagens, especialmente no reconhecimento facial. Esse processo começa com a captura de imagens faciais, que, ao serem convertidas para o domínio da frequência por meio da Transformada Discreta de Fourier, permite uma análise mais eficaz das características faciais. A Transformada Discreta de Fourier transforma a imagem que originalmente está no domínio espacial (onde as informações estão disponíveis em pixels), para o domínio da frequência, onde ela é representada por suas componentes de frequências, que facilita a identificação e comparação de padrões faciais.

Ao aplicar a Transformada Discreta de Fourier a uma imagem de um rosto, a decomposição resulta em componentes de diferentes frequências. As frequências mais baixas capturam as formas gerais do rosto, como a posição dos olhos, a linha da mandíbula e o formato do nariz, enquanto as frequências mais altas são responsáveis pelos detalhes finos, como texturas da pele, rugas e contornos detalhados. Isso ocorre porque as características mais suaves e amplas do rosto estão associadas a frequências baixas, enquanto os detalhes mais nítidos e as variações rápidas da imagem, como as linhas finas ao redor dos olhos ou boca, estão nas frequências altas.

Antes da aplicação da DFT, a imagem facial passa por um pré-processamento. Esse processo pode incluir redimensionamento, conversão para escala de cinza e ajuste de iluminação. Essas etapas são importantes para garantir que a análise seja realizada de maneira eficiente, evitando que elementos indesejados ou variações de iluminação afetem os resultados.

Após a conversão para o domínio da frequência, as componentes extraídas podem ser analisadas para extrair características relevantes para o reconhecimento facial. Por exemplo, as informações sobre a morfologia do rosto, como a distância entre os olhos ou o formato da mandíbula, são representadas pelas frequências baixas. Detalhes sobre a textura da pele, como poros, rugas e outros pequenos detalhes, são capturados pelas frequências mais altas. A partir dessas características extraídas, a imagem do rosto é representada de forma compacta, facilitando o processo de comparação com outras imagens.

No processo de reconhecimento facial, as características extraídas de uma imagem facial são comparadas com um banco de dados de rostos previamente registrados. A comparação pode ser feita tanto no domínio da frequência quanto no domínio espacial. No domínio da frequência, os coeficientes de Fourier das imagens faciais são comparados diretamente. Essa abordagem é vantajosa porque permite comparar apenas as características mais importantes do rosto, que são extraídas de maneira eficiente pela DFT.

Além disso, o uso da DFT no reconhecimento facial apresenta várias vantagens. A principal delas é a redução de dimensionalidade, que permite representar uma imagem facial com um número reduzido de coeficientes, mantendo as informações essenciais. Isso acelera a comparação e torna o processo de reconhecimento mais eficiente. A DFT também

é robusta ao ruído. Como ela separa as frequências de alta e baixa energia, o sistema pode ser mais resistente a distorções e variações na imagem, desde que as frequências relevantes sejam preservadas. Além disso, ela permite analisar a textura da pele, o que pode ajudar a distinguir um rosto de outro.

No entanto, apesar das vantagens, a DFT também tem limitações. Por exemplo, ela pode ser sensível a variações de iluminação, o que pode dificultar a comparação de rostos em diferentes condições de luz. Além disso, mudanças na pose do rosto, como a rotação da cabeça ou o ângulo do olhar, também podem afetar a precisão do reconhecimento. Isso ocorre porque a DFT depende das características espaciais da imagem, que podem ser alteradas por esses fatores.

Em resumo, a DFT é uma ferramenta poderosa para o reconhecimento facial, pois permite decompor a imagem facial em componentes de frequência que capturam as informações essenciais, como a forma geral do rosto e os detalhes da textura da pele. Ela facilita a extração de características faciais e permite a comparação eficiente entre imagens, tornando o processo de reconhecimento mais rápido e preciso.

A DFT é uma ferramenta fundamental para a análise de sinais no domínio de frequência, permitindo a decomposição de sinais complexos em componentes simples. Apesar da sua versatilidade, o elevado custo computacional limita sua aplicação em casos com grandes volumes de dados ou que exigem processamento em tempo real. Sua importância, no entanto, reside na base teórica que fornece para diversas aplicações, como compressão de dados, análise espectral de sinais e imagens, e reconhecimento de padrões [8]. A DFT, portanto é essencial para entender e manipular sinais digitais, sendo frequentemente utilizada em conjunto com abordagens mais eficientes, como a FFT, para viabilizar aplicações práticas em larga escala [10].

1.4 Transformada Rápida de Fourier

A Transformada Rápida de Fourier (FFT), do inglês *Fast Fourier Transform* é um algoritmo usado para calcular a Transformada Discreta de Fourier (DFT) de forma muito mais rápida. Sua grande vantagem está na redução significativa da complexidade computacional. Enquanto o cálculo direto da DFT complexidade de $O(N^2)$, crescendo rapidamente com o tamanho do conjunto de dados, a FFT reduz isso para $O(N \log N)$, tornando-a muito mais rápida e mais prática para conjuntos grandes de dados.

A DFT decompõe uma sequência de valores diferentes componentes de frequência, sendo amplamente utilizada em várias áreas. No entanto, realizar o cálculo pela definição pode ser muito lento. A FFT é um algoritmo que realiza o mesmo cálculo que a DFT de forma mais eficiente. Enquanto a abordagem direta requer um número de operações proporcional a $O(N^2)$. Isso significa que o tempo de execução do algoritmo cresce proporcionalmente ao número de n . Ou seja se o número de pontos ou dados for 10, o algoritmo precisaria de 100 operações, e assim por diante. Quando se realiza uma DFT manualmente a complexidade é $O(N^2)$, pois é necessário calcular cada ponto da Transformada de Fourier. O que envolve calcular soma de produtos entre todos os ponto de dados. A DFT requer n somas para cada um dos n pontos, resultando em $O(N^2)$. A FFT tem complexidade de $O(N \log N)$, o que a torna muito mais rápida que a DFT, especialmente para grandes números de pontos. Isso significa que o tempo de execução do algoritmo cresce de forma mais lenta que o $O(N^2)$, ele cresce em função de n o logaritmo de n , que é a taxa de crescimento mais eficiente, especialmente quando n é grande.

Essa diferença de eficiência é significativa, especialmente para conjunto de dados grandes, onde N pode ser da ordem de milhares ou milhões. Na prática, o tempo de processamento pode ser reduzido em ordens de magnitude, tornando o uso da DFT viável em diversas aplicações, como o processamento digital de sinais, resolução de equações diferenciais e algoritmos para multiplicação rápida de grandes inteiros [11].

Quando falamos em ordem de magnitude referem-se ao fatores de dez. Por exemplo: Se o tempo de execução de um algoritmo é reduzido de 100 segundos para 1 segundo, a redução foi de duas ordens de magnitude ($100 \div 10 \div 10$). Em termos simples, cada ordem de magnitude representa multiplicar ou dividir por 10.

O algoritmo da FFT usa método de dobramentos sucessivos para acelerar os cálculos. A Transformada de Fourier pode ser escrita como:

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) W_N^{ux}$$

onde:

$$W_N^{ux} = e^{-j \frac{2\pi ux}{N}}$$

Assumimos que $N = 2^n$, sendo n um número inteiro positivo. Assim, N pode ser rescrito como $N = 2M$, com M também sendo um número inteiro positivo.

Logo a Transformada de Fourier poderia ser escrita como:

$$F(u) = \frac{1}{2M} \sum_{x=0}^{2-1} f(x) W_{2M}^{ux}$$

Com isso, a Transformada de Fourier original pode ser reformulada como:

$$F(u) = \frac{1}{2} \left[\frac{1}{M} \sum_{x=0}^{M-1} f(2x) W_M^{ux} + \frac{1}{M} \sum_{x=0}^{M-1} f(2x+1) W_{2M}^{ux} \right].$$

Denominamos as duas somas acima como:

$$F_{\text{par}}(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(2x) W_M^{ux}, \quad F_{\text{impar}}(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(2x+1) W_{2M}^{ux}.$$

Com isso, a Transformada de Fourier pode ser reescrita como:

$$F(u) = F_{\text{par}}(u) + F_{\text{impar}}(u) W_{2M}^u.$$

Além disso, utilizando as propriedades de W :

$$W_M^{u+M} = W_M^u, \quad W_{2M}^{u+M} = -W_{2M}^u,$$

podemos expressar:

$$F(u + M) = F_{\text{par}}(u) - F_{\text{impar}}(u)W_{2M}^u. \quad (1.21)$$

Essas equações mostram que uma Transformada de Fourier com N pontos pode ser dividida em duas partes, cada uma com $M = N/2$ pontos. Esse processo é iterado até que as somas finais possam ser calculadas diretamente, resultando em uma redução significativa no número de operações.

O "dobramento sucessivo" é usado para dividir o cálculo de uma Transformada de Fourier em etapas menores e mais simples. Em vez de calcular a transformada de todos os pontos de uma vez, o processo divide os dados em duas partes: uma parte para os índices pares e outra para os índices ímpares.

Esses dados são então processados separadamente, e em seguida, os resultados são confirmados de forma eficiente para obter o resultado final. Esse "dobramento" reduz o número de cálculos, tornando o algoritmo muito rápido do que o cálculo direto da Transformada de Fourier.

A FFT não se refere a um único algoritmo, mas a vários algoritmos desenvolvidos para calcular a DFT de maneira mais eficiente [12]. Esses algoritmos exploram propriedades matemáticas como simetria e periodicidade para reduzir a quantidade de cálculos necessários. Dependendo da situação, diferentes variações da FFT são utilizadas.

a fórmula geral da FFT é dada por:

$$g(f) = \sum_{n=0}^{N-1} g(t_n) e^{-i \frac{2\pi f t}{N}} \quad (1.22)$$

- $g(f)$: Valor do sinal no domínio da frequência;
- $g(t)$: Valor do sinal no domínio do tempo no instante t ;
- $e^{-i \frac{2\pi f t}{N}}$: Termo que realiza a transformação de cada ponto do domínio do tempo para o domínio da frequência;
- f : Representa a frequência;
- N : Número total de pontos amostrados no sinal.

A equação que trabalhamos representa o princípio fundamental por trás da FFT mostrando como a DFT (DFT) pode ser dividida em somas menores, separando os dados em partes

par e ímpar. Esse conceito é a essência da maioria dos algoritmos, sendo adaptado em cada caso para entender a requisitos como tamanho de dados não convencionais.

A equação não é um único algoritmo, é um modelo matemático que guia a construção de algoritmos otimizados, cada um adaptado a diferentes situações e tecnologias. Essa ferramenta tornou a FFT uma ferramenta valiosa em áreas de telecomunicações, processamento de sinais, compressão de dados e até mesmo transmissões esportivas, como a Copa do Mundo.

Durante a copa do Mundo de 2010, o som das vuvuzelas se tornou um problema nas transmissões, por causa do seu ruído permanente e alto. A FFT foi usada para remover esse problema, permitindo desagregar as frequências específicas do som das vuvuzelas do restante do áudio. Ao detectar essas frequências, foi possível aplicar filtros para diminuir ou remover esse ruído sem comprometer outros sons importantes, como os comentários e os sons do jogo.

Cálculos numéricos envolvem o desenvolvimento de algoritmos para encontrar soluções aproximadas de problemas matemáticos. A DFT e a FFT se tratam desses cálculos, em vez de trabalhar com integrais contínuas como na Transformada de Fourier tradicional, esses métodos utilizam somatórios para processar valores em um conjunto finito com um número de pontos N espaçados de formas iguais dentro de um certo intervalo de tempo. Essa discretização é essencial para computadores, que não conseguem trabalhar com dados infinitos. Dessa maneira a DFT e a FFT convertem informações matemáticas em operações numéricas práticas, permitindo a análise eficiente de sinais reais em aplicações digitais [13].

Capítulo 2

Resultados e Discussões

Neste capítulo, será apresentada uma análise detalhada dos resultados obtidos da Transformada de Fourier, com foco na versão discreta, a DFT e a rápida, a FFT. O objetivo é interpretar os dados da maneira objetiva, relacionando os conceitos teóricos à aplicação prática apresentada nos experimentos e nos gráficos obtidos. Serão demonstrado os métodos aplicados para analisar sinais periódicos, identificar ruídos com filtragem. Além disso, será discutido como as diferença entre a DFT e FFT influenciam na eficiência computacional e no desempenho em aplicações práticas. Também abordará ainda as limitações de possíveis aplicações desses métodos em diferentes contextos.

2.1 Transformada Discreta e Rápida de Fourier

Neste trabalho, analisamos sinais periódicos simples compostos por combinações de cossenos, como:

$$g(t) = A\cos(\omega t) \tag{2.1}$$

Em que ω é a frequência angular, é a medida de rotação, a velocidade angular, descreve a taxa com que sua variação muda, é expressa da seguinte forma:

$$\omega = 2\pi f$$

Além dos sinais principais, adicionamos ruídos a cada um deles para simular cenários reais em que sinais são corrompidos por interferências externas. Esses ruídos são componentes de altas frequências, superiores a 6 Hz, que podem dificultar a análise do comportamento dos sinais originais. No contexto deste trabalho, esses ruídos foram gerados por meio de um comando presente na linguagem C que cria números aleatórios, os quais foram somados aos sinais no domínio do tempo, introduzindo variações inesperadas nos dados. Os ruídos foram escolhidos para ter frequências acima de 6 Hz, enquanto as frequências dominantes dos sinais originais são menores do que 1. Essa escolha estratégica permite uma separação clara entre as componentes desejadas e os ruídos. Como resultado, ao aplicar um filtro para remover frequências acima de 6 Hz, é possível eliminar os ruídos sem comprometer as frequências dominantes do sinal original, garantindo uma reconstrução precisa [14]. Esse procedimento é relevante tanto em aplicações práticas, onde ruídos de alta frequência podem representar interferências externas, quanto em contextos educacionais, pois facilita a visualização e compreensão do impacto do ruído e da eficiência do filtro utilizado.

Para analisar e eliminar os ruídos, utilizamos a DFT e sua versão eficiente, a FFT. Essas transformadas permitem converter os sinais do domínio do tempo para o domínio da frequência, onde as componentes indesejadas podem ser facilmente identificadas. Após a identificação, aplicamos um filtro para remover as frequências acima de 6 Hz. Finalmente, utilizamos a Transformada Inversa de Fourier para reconstruir os sinais no domínio do tempo, eliminando os ruídos e preservando as características essenciais dos sinais originais.

Os sinais utilizados apresentam diferentes frequências e amplitudes, os sinais são:

$$1\cos(1t) \tag{2.2}$$

$$4\cos(1t) \tag{2.3}$$

$$1\cos(5t) \tag{2.4}$$

$$4\cos(5t) \tag{2.5}$$

A figura (2.1) apresenta os sinais originais com maior e menor amplitude no domínio do tempo, uma grandeza que determina altura máxima e mínima de oscilações. No contexto dos sinais que analisamos, a amplitude controla a intensidade do sinal e influencia sua resiliência frente ao ruído, tornando os sinais de maior amplitude mais perceptíveis, mesmo com interferência. Na figura (2.1) os sinais (2.3) e (2.5) mantêm suas características mais perceptíveis, mesmo com a interferência do ruído. Porém, os sinais (2.2) e (2.4) apresentam uma amplitude menor e são mais afetados, dificultando uma observação clara das oscilações. A amplitude desempenha um papel fundamental na visibilidade e resiliência do sinal em meio ao ruído.

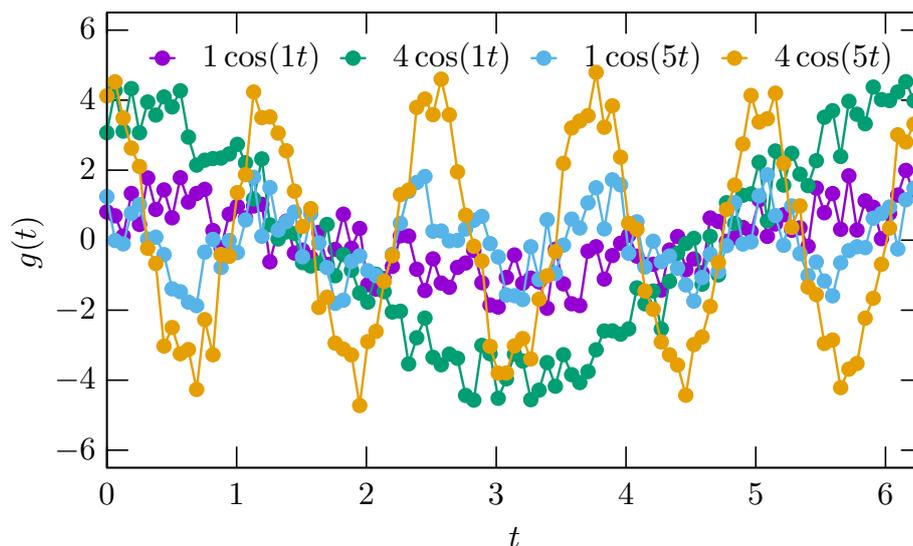


Figura 2.1: Sinais com interferência do ruído antes da DFT e da FFT.

Os sinais analisados possuem N pontos porque foram discretizados, ou seja, convertidos de um formato contínuo para um formato discreto, permitindo seu processamento em um computador. Esse processo de discretização consiste em amostrar o sinal em intervalos de tempo igualmente espaçados dentro do intervalo $[0, 2\pi)$.

O ruído não é apenas um valor isolado, ele pode ser visto como uma combinação de diversas senoides com frequências, amplitudes e fases aleatórias [15]. Assim, o ruído tem uma forma oscilatória própria, impactando para os espectros de frequência dos sinais com pequenas instabilidades espalhadas, geralmente em faixas de baixa frequência.

Na figura (2.1), o tempo e a frequência estão relacionados pelo comportamento periódico dos sinais. Ou seja, o período de cada sinal. A frequência f determina quantas oscilações acontecem em 1 segundo, enquanto o tempo é o domínio onde observamos essas variações, ou seja, as oscilações ao longo do tempo.

No domínio do tempo, é possível observar as oscilações do sinal ao longo de um intervalo de tempo, que são caracterizadas por amplitudes e períodos. No domínio da frequência, é possível observar como essas oscilações podem ser decompostas em diferentes frequências, cada uma com uma magnitude associada.

Para sinais como os que estamos analisando, o valor de f determina a rapidez com que o sinal oscila em 1 segundo. Quanto maior f , mais rápido o sinal oscila. Se f for pequeno, o sinal vai oscilar de forma mais lenta ao longo do tempo. A figura (2.1) apresenta a evolução dos sinais originais ao longo do tempo t , onde as oscilações estão diretamente relacionadas aos períodos dos sinais. As frequências dos sinais que estamos trabalhando são 0,159 Hz e 0,79 Hz, elas são afetadas pelos ruídos, que correspondem as frequências acima de 6 Hz. Esse tipo de filtro permite a passagem de frequências baixas, enquanto atenua ou elimina frequências mais altas. Dessa forma, o sinal recuperado mantém suas características essenciais, refletindo padrões periódicos do sinal original, agora sem perturbações causadas pelo ruído.

A Figura (2.2) mostra que todos os sinais foram recuperados de forma evidente após a filtragem, os sinais agora estão mais claros e fáceis de interpretar, sem instabilidades ou perturbações. Isso foi possível por ao aplicar a DFT e a FFT, os sinais foram convertidos do domínio do tempo para o domínio da frequência, permitindo identificar as frequências dominantes e os ruídos. Frequências indesejadas, maiores que 6 Hz, foram então removidas com uso de filtros apropriados, preservando apenas as frequências necessárias. Os sinais (2.3) e (2.5) possuem maior amplitude, permanecem dominantes e têm suas características mais evidente. Foi utilizado o filtro passa-baixa para preservar as frequências dominantes de de 0,159 Hz

Os sinais (2.2) e (2.5) também foram recuperados, entretanto o impacto do ruído era mais perceptível antes da filtragem, tornando-os mais sensíveis ao processo. A DFT (1.20) e a FFT (1.22) foram aplicadas aos sinais, convertendo-os do domínio do tempo para o domínio da frequência. Após a transformada ser realizada e o sinal ser decomposto, tornou-se possível identificar tanto as frequências dominantes quanto os ruídos que acompanham o

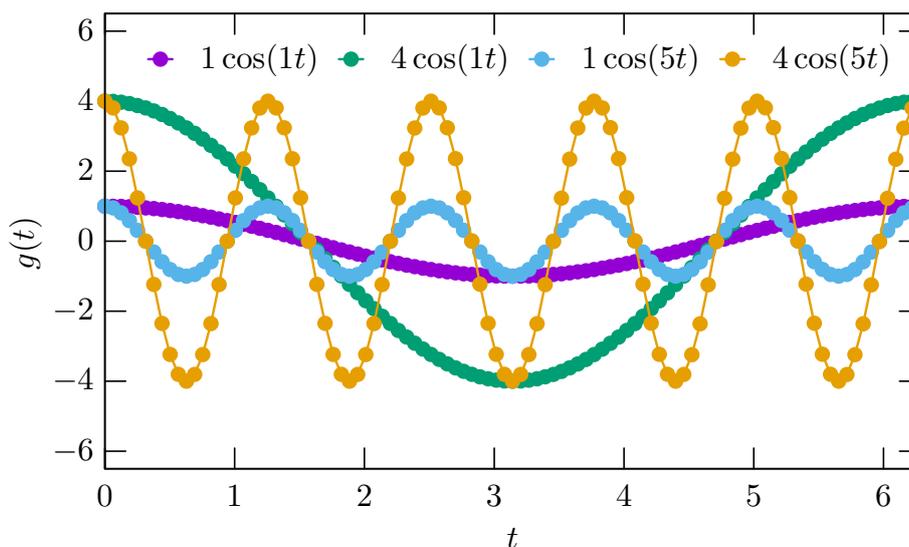


Figura 2.2: Sinais sem interferência do ruído após a DFT e FFT.

sinal. Nesse caso, os ruídos correspondem às frequências maiores que 6 Hz, enquanto as frequências dominantes, de 0,159 Hz e 0,79 Hz, refletem os padrões periódicos associados aos sinais originais [16].

No espectro de frequência, as frequências dominantes são identificadas como picos bem definidos, com magnitudes significativamente maiores em comparação às demais componentes. Isso indica que essas frequências têm maior contribuição para o sinal original. Por outro lado, os ruídos aparecem como componentes de baixa intensidade, espalhadas de maneira mais irregular pelo espectro. No caso analisado, as frequências dominantes de 0,159 Hz e 0,79 Hz apresentaram magnitudes muito mais altas, enquanto as frequências maiores que 6 Hz, de menor intensidade, foram reconhecidas como ruídos. Essa diferenciação é possível porque as frequências dominantes refletem os padrões periódicos principais do sinal, enquanto os ruídos se manifestam como variações não desejadas e menos intensas. É possível observar isso na figura (2.3)

Após a aplicação do filtro no domínio da frequência, a Transformada Inversa de Fourier (IFT) reconstrói o sinal no domínio do tempo sem as componentes indesejadas, resultando em um sinal mais limpo e próximo ao original, isso pode ser observado na figura (2.4).

A figura (2.2) mantém N pontos no domínio do tempo, satisfazendo à discretização do sinal original no intervalo $[0, 2\pi)$. No domínio da frequência, apenas $\frac{N}{2}$ pontos foram analisados.

As frequências positivas contêm todas as informações necessárias sobre o sinal, enquanto

as frequências negativas são redundantes. Essas frequências representam sinais no domínio da frequência, mas em sinais reais, elas não trazem novas informações, pois têm a mesma magnitude que as positivas, com diferença apenas na fase. Como as partes real e imaginária do espectro de frequência se complementam, não há necessidade de duplicação de informações. Por isso, apenas as frequências positivas são utilizadas na análise, sendo as negativas um reflexo das positivas.

A figura (2.4) apresenta os resultados da DFT e da FFT. O eixo horizontal f representa a frequência do sinal. Cada posição ao longo desse eixo corresponde a uma frequência específica que compõe o sinal original. O eixo vertical exibe o espectro de potência, que está relacionada à intensidade ou energia associada a cada frequência, que é resultado da DFT e FFT aplicada ao sinal. Esse espectro é calculado após a aplicação das Transformadas, convertendo o sinal do domínio do tempo para o domínio da frequência, decompondo-o em suas componentes de diferentes frequências. O espectro de frequência é a representação completa de um sinal no domínio da frequência. Pode ser demonstrado como:

$$g(f) = Re(g(f)) + Im(g(f))$$

A magnitude espectral indica a intensidade com que cada frequência contribui para o sinal total. Quanto maior o valor da magnitude em um ponto específico, maior é a contribuição daquela frequência para o sinal. Ou seja, a magnitude reflete o quanto uma frequência específica parece ou influencia o sinal original.

Na figura (2.4) as posições antes do pico e após o pico representam frequências que não contribuem significativamente para o sinal. Essas frequências têm magnitudes muito baixas ou nulas, o que indica que não são componentes principais do sinal.

A magnitude é o valor absoluto de uma componente de frequência no espectro. Ela mede a intensidade ou energia associada a uma frequência específica, sem considerar a fase. É calculada como raiz quadrada da soma dos quadrados das partes real e imaginária do espectro.

Além de que, as frequências negativas não são necessárias, porque o espectro de sinais reais apresenta a simetria Hermitiana. Isso quer dizer que:

$$g(-f) = g^*(f)$$

Onde $g^*(f)$ é o complexo conjugado de $g(f)$. Isso significa que a parte real de $g(f)$, o cosseno, é igual para frequências f e $-f$:

$$Re(g(-f)) = Re(g(f))$$

A parte imaginária de $g(f)$, o seno, muda de sinal para frequências f e $-f$:

$$Im(g(f)) = -Im(g(-f))$$

Se $g(f)$ for um número complexo como:

$$g(f) = Re(g(f)) + j \cdot Im(g(f))$$

O complexo conjugado de $g^*(f)$ é:

$$g^*(f) = Re(g(f)) - j \cdot Im(g(f))$$

A parte real não sofre alterações, mas a parte imaginária troca de sinal.

significa que as frequências negativas têm o mesmo módulo que as frequências positivas, mas a parte imaginária muda de sinal. Dessa maneira, as frequências positivas são suficientes para descrever completamente o sinal do domínio da frequência [17]. isso elimina redundâncias e melhora a eficiência computacional, isso economiza tempo e memória.

A magnitude é calculada como:

$$|g(f)| = \sqrt{Re(g(f))^2 + Im(g(f))^2}$$

Essa operação elimina qualquer diferença de sinal, tanto na parte real quanto na parte imaginária, e resulta em valor positivo tanto para f quanto para $-f$, ou seja, valores idênticos para as duas frequências.

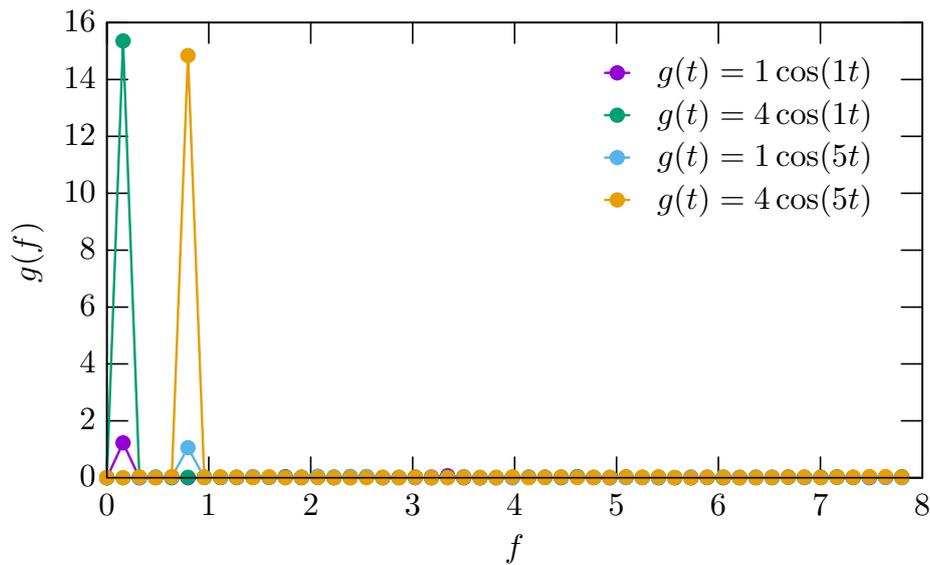


Figura 2.3: Espectro de frequências das funções senoidais analisadas antes do filtro.

Na figura (2.4), as posições antes do pico e após o pico representam frequências que não contribuem significativamente para o sinal.

A figura mostra o espectro de frequência dos sinais antes do filtro ser aplicado para remover os ruídos.

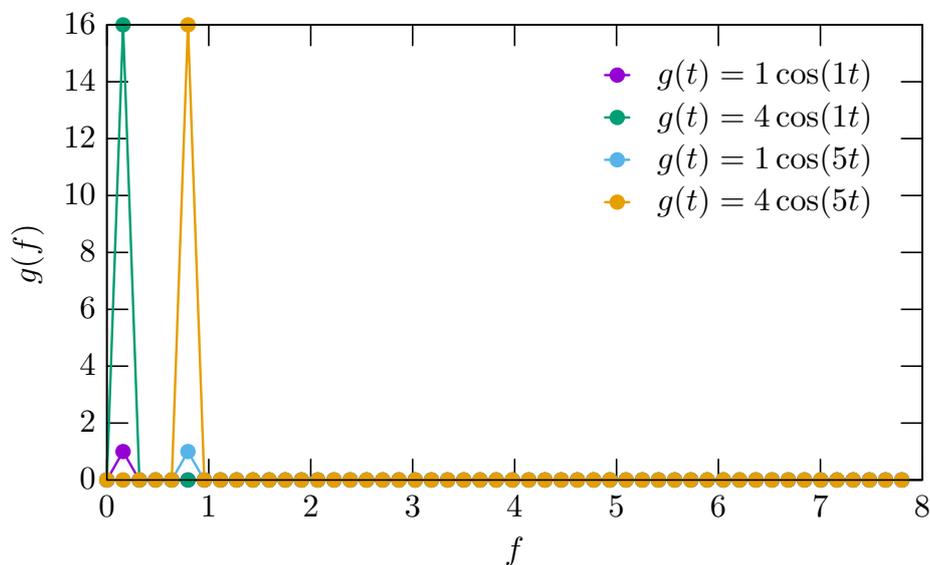


Figura 2.4: Espectro de frequências das funções senoidais analisadas após o filtro.

Mostrando como as instabilidades implicam as magnitudes esperadas das frequências dominantes. No espectro, as frequências dominantes deveriam ter magnitudes exatamente iguais a 1 para os sinais (2.2) e (2.4), e 16 para os sinais (2.3) e (2.5).

Porém, devido à presença do ruído, essas magnitudes não atingem os valores exatos es-

perados, mas aparecem deslocados ao redor de 1 e 16. Isso acontece porque o ruído reduz a precisão das frequências dominantes e na amplitude. Além disso, os ruídos espalham energia para frequências próximas, causando pequenas flutuações ao longo do espectro, essa dispersão de energia desmancha a intensidade concentrada nos picos dominantes, resultando em uma leve atenuação de suas magnitudes.

Outro fator que influencia essas variações é a natureza do cálculo prático da Transformada de Fourier. A discretização do sinal no domínio do tempo, realizada usando N , combinada com a interferência do ruído, introduz pequenas imprecisões nos valores obtidos no domínio da frequência.

A figura (2.5) mostra o espectro de frequência em escala logarítmica dos sinais analisados, incluindo o impacto do ruído. O eixo horizontal f exibe as frequências presentes no sinal, enquanto o eixo vertical $g(f)$ mostra a magnitude associada a cada frequência em uma escala logarítmica, variando de 10^{-14} a 10^{12} .

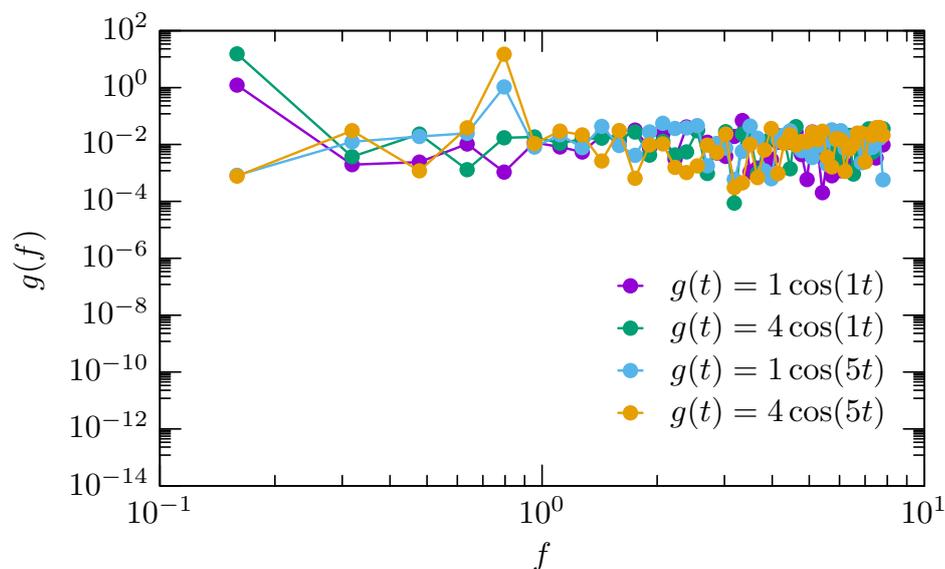


Figura 2.5: Espectro de frequências das funções senoidais analisadas antes do filtro em escala logarítmica.

Os picos nas frequências 0,159 Hz e 0,79 Hz são as contribuições principais, correspondentes as senoides dos sinais analisados. Mesmo valores muito baixos de magnitude são exibidos no gráfico, mostrando a pequena contribuição de frequências f , o que ajuda a observar o espectro de maneira mais clara, instabilidades adicionais.

A figura (2.6) mostra o gráfico do espectro de frequência dos sinais analisados sem a interferência dos ruídos, em escala logarítmica, para destacar as pequenas contribuições

espectrais, aproximadamente 10^{-14} , em contraste com as grandes magnitudes, em torno de 10^{12} . Na figura (2.4) o gráfico está em escala linear, e valores muito pequenos podem ficar difíceis de serem visualizados. Com a escala é possível visualizar claramente as diferenças entre ordens de magnitude.

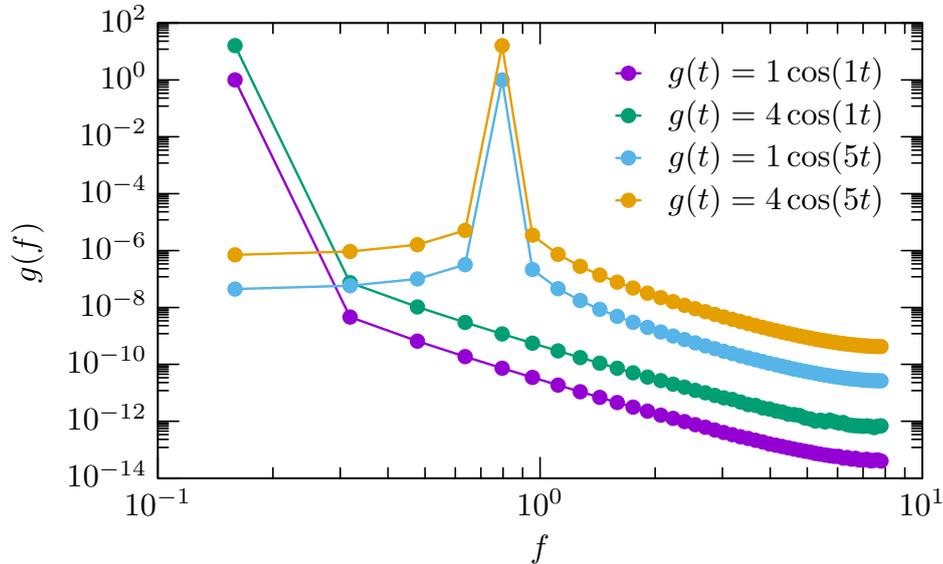


Figura 2.6: Espectro de frequências das funções senoidais analisadas após o filtro em escala logarítmica.

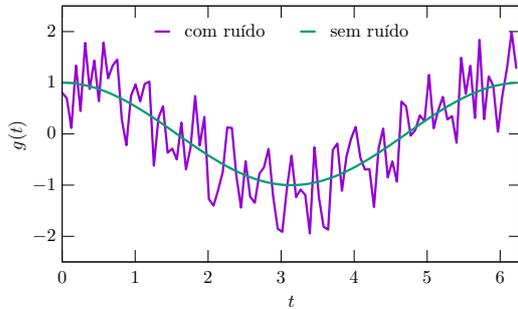
Os principais picos acontecem na frequência $f = 0,159$ Hz e $f = 0,79$ Hz, que correspondem às quatro senoides originais. Esses picos demonstram as contribuições dominantes dos sinais analisados, enquanto as magnitudes menores em frequências intermediárias e em frequências baixas são atribuídas a presença de ruídos.

A Figura (2.5) mostra como o ruído afeta o espectro ao introduzir pequenas contribuições adicionais em frequências que não são dominantes. Essas contribuições, mesmo que muito pequenas, são visíveis por conta da escala logarítmica, que amplia a diferença entre valores pequenos, em torno de 10^{-4} e valores altos, próximos de 10^2 .

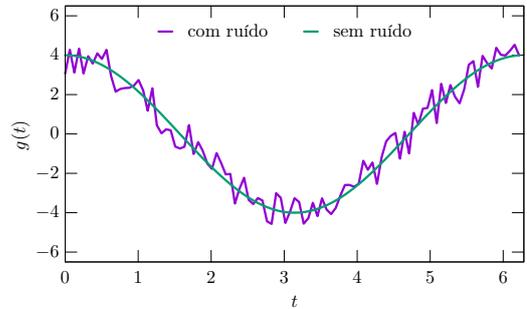
Portanto, o gráfico destaca a composição do sinal no domínio da frequência, mostrando as contribuições principais das frequências dominantes e a interferência adicional do ruído. Essa visualização ajuda a compreender como as frequências originais ainda são reconhecíveis, mesmo com a influência do ruído, e a importância de usar escala logarítmica para capturar variações sutis no espectro.

A figura (2.7) tem como objetivo comparar os sinais originais com a interferência dos ruídos e os sinais reconstruídos após o processo de filtragem [18]. Esse procedimento é

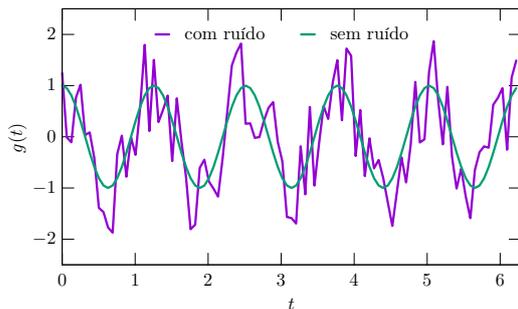
aplicado aos quatro sinais analisados, os sinais com ruídos, obtidos antes da aplicação da DFT e FFT exibem distorções introduzidas por componentes de baixa frequência. Após a aplicação da DFT e FFT as frequências indesejadas são removidas, e o sinal filtrado é reconstruído no domínio do tempo usando a Inversa da DFT e a inversa da FFT.



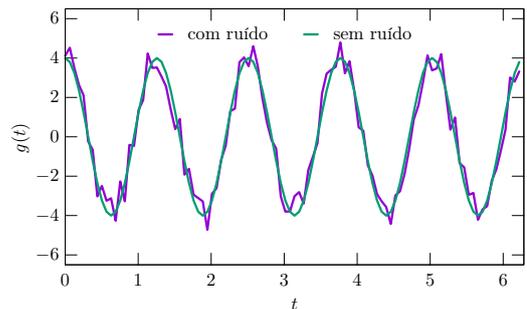
(a) Espectro do sinal $g(t) = \cos(1t)$ com e sem ruído.



(b) Espectro do sinal $g(t) = 4 \cos(1t)$ com e sem ruído.



(c) Espectro do sinal $g(t) = \cos(5t)$ com e sem ruído.



(d) Espectro do sinal $g(t) = 4 \cos(5t)$ com e sem ruído.

Figura 2.7: Comportamento dos quatro sinais analisados, comparando as versões originais e com ruído.

Nos gráficos analisados, o eixo horizontal t representa o tempo, enquanto o eixo vertical $g(t)$ mostra o valor absoluto do sinal para cada instante de tempo, a linha distorcida exibe o comportamento dos sinais influenciados pelo ruído. Enquanto a linha sem distorções apresenta o sinal recuperado, com suas características dominantes preservadas.

O sinal (2.2) apresenta amplitude igual a 1, uma frequência angular baixa, igual a 1, no gráfico (2.7a) a linha que representa o sinal influenciado pelo ruído exibe distorções significativas, causadas pela interferência de componentes de baixa frequência, que dificultam uma visualização mais clara das oscilações periódicas características do sinal. Esse comportamento é esperado, já que sinais com menor amplitude são mais suscetíveis ao impacto do ruído.

Isso acontece porque o ruído, sendo de intensidade semelhante para todos os sinais, pode esconder as oscilações do sinal menor, tornando difícil notar o comportamento original do sinal.

Após a aplicação da DFT, FFT e do filtro, o sinal recuperado mostra uma oscilação regular e limpa, exibindo a preservação de sua amplitude e frequências dominantes. As frequências abaixo do limite de corte ($f < 6$ Hz) foram eliminados de forma eficiente, permitindo uma visualização clara do comportamento periódico do sinal. Essa recuperação ressalta a eficiência do filtro aplicado no sinal (2.2), especialmente em sinais com baixa amplitude.

O sinal (2.3), com amplitude igual a 4 e frequência angular baixa, igual a 1. A linha influenciada que representa o sinal influenciado pelo ruído exibe oscilações mais perceptíveis em comparação aos sinais de menor amplitude, por mais que apresente distorções causadas pela interferência do ruído, o gráfico (2.7b) mostra oscilações mais perceptíveis em comparação aos sinais de menor amplitude. Essa resistência aos ruídos acontece porque o sinal (2.3) apresenta uma amplitude maior.

Após a aplicação da DFT e FFT e do filtro, o sinal reconstruído apresenta uma oscilação regular e limpa. O processo de filtragem removeu eficientemente as frequências abaixo do limite de corte ($f < 6$ Hz), mantendo as características dominantes do sinal (2.3).

O sinal (2.4) apresenta amplitude igual a 1 e frequência angular igual a 5, exibem oscilações rápidas que são menos evidentes devido às distorções do ruído. Após a DFT e FFT e a aplicação do filtro, as oscilações periódicas características do sinal são restauradas, com frequência abaixo de 6 Hz removidas, dando ênfase para as frequências dominantes, mesmo para baixa amplitude.

E por fim, o sinal (2.5) apresenta amplitude igual a 4 e frequência angular igual a 5, mantém suas oscilações rápidas e suas principais características, mesmo com a interferência do ruído. Após a DFT (1.20) e FFT (1.22) e a aplicação da filtragem, o sinal mostra uma oscilação regular e sem interferência do ruído ($f < 6$ Hz).

Ao observar os quatro gráficos da figura (2.7) é evidente que os sinais com amplitude mais baixa sofrem mais interferência. Isso acontece porque o ruído, em todos esses casos, de intensidade semelhante, interfere mais proporcionalmente nos sinais de menor amplitude, dificultando a identificação de suas oscilações características.

Na DFT, o cálculo de cada componente de frequência é realizado diretamente, seguindo

a definição matemática da transformada. Esse processo envolve um laço duplo, o laço externo percorre todas as frequências f , enquanto o laço interno percorre todos os pontos do sinal no tempo t . Para cada f , é somado a contribuição de todos os t , o que resulta em um custo computacional de $O(N^2)$. Esse método, embora direto e simples de compreender, é ineficiente para sinais com muitos pontos, devido ao grande número de operações necessárias.

A FFT, elimina a necessidade desses laços explícitos ao utilizar algoritmos otimizados, como o *Cooley-Tukey*, que empregam a técnica de divisão e conquista. Esses algoritmos decompõem o cálculo da transformada em problemas menores, reutilizando resultados intermediários. Além do mais, eles exploram propriedades como a simetria e a periodicidade das exponenciais complexas, reduzindo o número de cálculos necessários. O custo computacional da FFT é $O(N \log N)$, tornando-a muito mais eficiente do que a DFT direta.

Cooley-Tukey é um algoritmo usado na FFT que divide o cálculo da Transformada de Fourier em partes menores, organizando os dados de forma eficiente para reutilizar cálculos, ele explora simetria e periodicidade das funções senos e cossenos, reduzindo o número de operações necessárias.

Algoritmos otimizados são métodos matemáticos ou computacionais feitos para resolver problemas de maneira mais eficiente, reduzindo o tempo de execução e uso de recursos. $O(N^2)$ significa que o número de operações cresce com o quadrado da quantidade de pontos (N). É mais devagar para sinais lento para sinais grandes. $O(N \log N)$ significa que o número de operações cresce proporcionalmente ao número de pontos multiplicados pelo algoritmo de N , sendo mais rápido e eficiente.

Por exemplo, vamos imaginar que temos um sinal discretizado dentro de um intervalo de tempo com 1024 pontos, o cálculo é feito de duas maneiras diferentes [19].

Cálculo com $O(n^2)$:

O número de operações será:

$$N^2 = 1024^2 = 1048576 \text{ Operações.}$$

Cálculo com $O(N \log N)$:

Usando $\log_{10}(1024)$, aproximadamente, 3,01:

$$O(N \log N) = 1024 \cdot 3,01 \approx 3,084.$$

Capítulo 3

Sequência Didática

Neste capítulo, será apresentada uma sequência didática elaborada com o objetivo de promover o ensino integrado da DFT e da FFT para o nível superior. A proposta se estrutura com um planejamento e apresenta três momentos de aprendizagem: planejamento, situação inicial, trabalho de construção e aplicação [20]. A sequência didática conecta os fundamentos teóricos das transformadas com suas aplicações práticas computacionais para a construção do conhecimento. Este capítulo detalha cada etapa da sequência, desde a introdução ao tema até a avaliação final.

É importante destacar que esta sequência didática não foi aplicada em sala de aula, sendo desenvolvida exclusivamente como um exercício teórico e metodológico. Além disso, trata-se de uma proposta voltada para um curso completo, considerando a complexidade e a amplitude do tema. Assim, o conteúdo apresentado não seria possível de ser ensinado de maneira integral em poucas aulas, exigindo um cronograma mais amplo e detalhado para sua implementação efetiva.

3.1 Planejamento

O planejamento desta sequência didática é importante para pronunciar um aprendizado significativo e estruturado, conectando ideias teóricas e aplicações práticas ao contexto real da análise de sinais digitais. Esta etapa organiza os objetivos educacionais, conteúdos e atividades, garantindo que o ensino seja progressivo e adaptado às necessidades dos alunos.

O principal objetivo da sequência didática é contribuir para que os alunos construam o seu entendimento teórico e na aplicação prática das DFT e a FFT, utilizando a linguagem de programação C como ferramenta principal. Para isso, é necessário:

- Proporcionar uma ótima compreensão teórica sobre os fundamentos da DFT e FFT.
- Diferenciar a DFT e FFT, mostrando suas características, vantagens e limitações.
- Desenvolver habilidades práticas na implementação dessas transformadas em C, com ênfase no uso de bibliotecas específicas
- Preparar os alunos a interpretar gráficos no domínio do tempo e da frequência, e aplicar as transformadas em tempos reais.

As atividades devem ser organizadas progressivas, focando na relação entre a teoria e a prática. A introdução trabalha com os conceitos fundamentais dos domínios do tempo e da frequência, contextualizando a necessidade de transformar sinais demonstrando como isso é útil em diversas áreas. Exemplos visuais e simulações interativas podem ser utilizados para ilustrar os conceitos e estimular o interesse dos alunos.

Na prática, os alunos serão orientados a implementar a DFT em C, analisando sua complexidade computacional $O(N^2)$. Em seguida, será introduzida a FFT, dando destaque para a sua otimização computacional, por meio do algoritmo *Cooley-Tukey*, que reduz a complexidade para $O(N \log N)$. Nesta fase, será explorado o uso da biblioteca FFTW, que facilita a aplicação da FFT em sistemas computacionais. Por fim, os alunos vão lidar com desafios práticos, como a remoção de ruídos de sinais e interpretação de gráficos espectrais.

Para que as atividades sejam bem executadas, é necessário laboratórios de informática, equipamentos com compiladores da linguagem C são necessários, conjunto de dados simulados, gráficos e computação técnica sobre as bibliotecas. O conteúdo abordado neste trabalho é voltado para um curso completo. A complexidade dos conceitos e das aplicações práticas, como a implementação em linguagem C e a análise detalhada de sinais no domínio da frequência, exige uma abordagem aprofundada e contínua. Esse formato permite integrar de maneira eficaz os fundamentos teóricos às práticas computacionais, proporcionando aos participantes uma formação sólida e abrangente sobre as transformadas Discreta e Rápida de Fourier, além de suas aplicações em diferentes áreas do conhecimento.

O curso foi planejado com uma carga horária total de 60 horas, distribuídas em 2 encontros semanais de 2 horas cada, ao longo de 15 semanas. Essa estrutura proporciona tempo suficiente para uma abordagem detalhada e progressiva dos conceitos teóricos e práticos, garantindo que os participantes tenham uma experiência de aprendizado completa e integrada.

A intervenção pedagógica será planejada para entender a diversidade de níveis de conhecimento dos alunos. O professor estará disponível para oferecer suporte individualizado e ajustará o ritmo das atividades conforme necessário. O trabalho em grupo também vai ser incentivado, em algumas atividades, promovendo a colaboração e troca de ideias.

3.2 Situação Inicial

Esse momento de aprendizagem tem o objetivo de introduzir o tema de modo envolvente, desafiadora e relevante para os alunos, a situação inicial deve ser feita com muito cuidado para gerar interesse e motivação para os alunos. No caso da DFT e FFT gerar uma motivação para os alunos para estudarem essas ferramentas.

A situação inicial pode começar com a apresentação de um problema real, que mostre a necessidade de transformar sinais entre o domínio do tempo e o domínio da frequência.

O uso de exemplos práticos e próximos da realidade dos alunos é muito importante para envolver os estudantes com o conteúdo que está sendo ensinado, o professor pode, por exemplo, mostrar aos alunos que um sinal simples como $g(t) = A\cos(\omega t)$, enfatizando como esse sinal no domínio do tempo representa uma variação suave e periódica ao longo de um intervalo. Porém, ao analisar esse mesmo sinal em função da frequência, é possível decompor em suas componentes espectrais, mostrando que ele é composto por uma frequência dominante que representa a taxa de variação dessa onda ao longo do tempo. Esse simples exemplo prepara os alunos para a próxima discussão sobre a importância de transformar sinais no domínio do tempo para o domínio da frequência, pois essa transformação facilita a análise e o processamento, especialmente quando se trabalha com sinais mais complexos.

Após essa introdução, o professor pode introduzir um cenário mais complexo, como o processamento de sinais de áudio ou de imagem, que são bastante utilizados em diversas áreas. Mostrar um sinal acompanhado com informações indesejáveis, e explicar que, para isolar o som desejado, seja uma fala ou uma música, é necessário aplicar a Transformada de Fourier para decompor um sinal e as frequências que formam esse sinal. Dessa maneira, é possível questionar os alunos: "De que maneira é possível isolar o som de uma música em uma gravação quando este está mesclado a ruídos indesejáveis?"

A partir dessa pergunta, o professor pode apresentar a necessidade da DFT para analisar sinais mais complexos. A DFT serve para transformar um sinal discreto no domínio do tempo para o domínio da frequência, o que permite visualizar e analisar as frequências que formam o sinal. O professor pode ressaltar que após a transformada ser feita é possível reconhecer as frequências dominantes e as indesejadas, como o ruído, o que possibilita

manipular o sinal de maneira eficiente.

Aqui, o professor pode também discutir a complexidade computacional envolvida nesse processo. A DFT é embora muito útil, tem uma complexidade de $O(N)^2$, quando está trabalhando com sinais de grande porte, como gravações de áudio ou imagens, isso pode se tornar um problema, porque o tempo de processamento se torna muito alto. Nesse momento, o professor introduz a FFT mostrando que essa é uma versão otimizada da DFT, diminuindo sua complexidade para $O(N \log N)$. A FFT, portanto, é a otimização da DFT, uma versão mais rápida, que lida com os mesmos desafios, alcançando o mesmo resultado que a DFT, mas com um menor tempo computacional.

Neste momento, o professor faz uma análise mais detalhada sobre a importância da DFT e da FFT no contexto real. Por exemplo, ele pode mostrar que, ao aplicar a FFT em um sinal de áudio, é possível atenuar o ruído de fundo e destacar determinados componentes, como instrumentos musicais ou a voz. No entanto, esse processo não é trivial e depende de técnicas avançadas de filtragem e processamento de sinais. Isolar completamente um instrumento musical ou uma voz requer um espectro de frequência bem definido e pouco sobreposto com outras frequências no sinal. Em muitos casos, a presença de frequências compartilhadas entre diferentes componentes do áudio torna o isolamento completo um desafio, exigindo abordagens sofisticadas. Ao fazer essa conexão como um exemplo do cotidiano dos alunos, o professor facilita a compreensão dos conceitos que serão abordados na sequência didática.

Para deixar a discussão mais completa o professor pode mostrar gráficos que mostram a diferença entre o sinal original e o espectro de frequências antes e depois da aplicação da DFT e da FFT. Isso possibilita uma visualização concreta de como os sinais são representados no domínio da frequência e ajuda os alunos a entenderem a importância de trabalhar com as transformadas.

O papel do professor neste momento não é apenas transmitir o conhecimento, mas também incentivar os alunos a desenvolverem pensamentos críticos sobre as aplicações das Transformadas de Fourier. Deixando o seguinte questionamento:

”Porque a DFT é tão importante para análise de sinais? Como a FFT resolve problemas práticos, como é o tempo de processamentos em sinais reais?”

Essa é uma maneira de ativar o interesse dos alunos sobre sinais e sistemas e desenvolve

uma base sólida para estudo teórico e prático das Transformadas de Fourier, com ênfase na DFT e FFT, despertando o interesse e curiosidade para a próxima fase de construção de conhecimento.

3.3 Trabalho de Construção

Esta sessão é a etapa central da sequência didática, onde os alunos exploram ativamente os conteúdos, desenvolvem habilidades práticas e consolidam o aprendizado por meio de atividades desafiadoras e significativas [21]. Nesta fase, o foco está em proporcionar aos alunos uma compreensão profunda e aplicada da DFT e, incluindo teoria, prática computacional e análise crítica.

Essa etapa começa com uma exploração teórica e detalhada sobre as Transformadas de Fourier. O professor revisa conceitos fundamentais, explicando como sinais podem ser representados no domínio do tempo e da frequência, e a importância dessa transformação para decompor sinais complexos e suas componentes básicas. Dessa maneira, os alunos devem:

Entender a DFT, para que isso aconteça, é necessário explicar matematicamente, de forma clara, a DFT como uma soma ponderada de senoides e cossenoides, mostrando como ela transforma um sinal discreto no domínio do tempo para o domínio da frequência. A fórmula da DFT deve ser apresentada, destacando o significado de cada termo.

Aqui o professor deve explicar a relação entre a soma exponencial e as representações em senoides, utilizando gráficos como as componentes da frequência são extraídas. Após isso ser feito, a FFT deve ser explorada, introduzindo o algoritmo *Cooley-Tukey*, que otimiza a DFT ao dividir o cálculo em problemas menores, organizador de forma hierárquica. O professor dá ênfase como essa abordagem reduz a complexidade computacional para $O(N \log N)$, mostrar que a FFT alcança os mesmos resultados que a DFT de forma mais rápida.

Durante essa etapa teórica, o professor vai utilizar exemplos visuais e gráficos para mostrar o comportamento de sinais no domínio do tempo e sua decomposição no domínio da frequência. Os alunos são incentivados a anotar dúvidas e reflexões, relacionando os conceitos teóricos à aplicação prática.

Após a revisão teórica, os alunos são direcionados para a prática computacional, implementando a DFT e explorando a FFT com o uso de bibliotecas específicas. Essa parte é organizada em três etapas:

- Implementação da DFT em Linguagem C: Os alunos desenvolvem o código para

calcular a DFT de sinais simples, como $g(t) = A\cos(2\pi ft)$. Eles implementam o cálculo de cada termo da fórmula da DFT. O professor orienta os alunos a analisar a complexidade computacional do código, dando ênfase para os limites da DFT quando aplicada a sinais com muitos pontos N .

- Exploração da FFT com a FFTW: Os alunos são introduzidos à biblioteca FFTW (*Fast Fourier Transform in the West*) uma ferramenta muito utilizada para calcular a FFT. Aqui o objetivo é mostrar como bibliotecas otimizadas automatizam cálculos complexos, permitindo a análise de sinais de grandes dimensões em menos tempo. O professor pode apresentar um exemplo prático, como a análise de um sinal de áudio ou de imagem, onde a FFT é usada para identificar frequências dominantes.
- Análise de Sinais com Ruído: Os alunos aplicam a DFT em sinais com ruído, observando como o espectro de frequência revela tanto as frequências dominantes quanto as indesejadas, no caso, os ruídos. Os alunos, para solucionar estes problemas são desafiados a aplicar filtros para remover esses ruídos, sejam eles de frequências mais altas ou mais baixas, após fazerem isso, devem reconstruir o sinal original no domínio do tempo. Isso conecta os conceitos teóricos às suas aplicações em problemas reais.

Após a implementação e realização das atividades em linguagem C, os alunos são incentivados a refletir sobre os resultados obtidos. Eles comparam a eficiência computacional da DFT e da FFT, discutindo a diferença de tempo e precisão dos resultados obtidos. O professor promove uma discussão em grupo, fazendo questionamentos, como:

- "Como a DFT e a FFT se diferem em termos de desempenho?"
- "Por que a FFT é mais adequada para sinais de alta densidade de dados?"
- "Quais são as limitações de cada abordagem em aplicações reais?"

Os alunos também vão discutir os desafios enfrentados na implementação dos códigos e na análise de sinais, identificando estratégias para melhorar a precisão e eficiência de suas soluções.

Após realizar as atividades os alunos vão fazer um relatório sobre as atividades que foram feitas em sala. Vai servir como uma ferramenta essencial para cimentar o aprendizado e

avaliar a capacidade dos alunos de organizar e comunicar os conceitos e práticas desenvolvidas durante o estudo da DFT e da FFT. Deve ser desenvolvido de forma simples e objetiva, contendo:

- **Introdução:** Uma breve explicação sobre a relevância das Transformadas de Fourier dando ênfase para a sua aplicação em situações reais, como análises de sinais com ruído. De acordo com a forma como eles compreenderam o conteúdo.
- **Fundamentação Teórica:** Escrever de forma objetiva sobre a DFT e FFT, de preferência incluindo as expressões matemáticas de todas as transformadas e suas aplicações da transição do domínio de frequência para o domínio do tempo.
- **Descrição Prática:** Uma apresentação de códigos utilizados para resolver as atividades e gráficos gerados durante a análise, incluindo os sinais originais, espectro de frequência e sinais filtrados. Os alunos devem interpretar os resultados, dando ênfase para as frequências dominantes e os impactos dos filtros aplicados.
- **Metodologia:** Este trabalho integra teoria e prática, com foco na compreensão e aplicação das Transformadas Discreta de Fourier (DFT) e Rápida de Fourier (FFT). As atividades foram estruturadas em etapas que incluíram a introdução teórica ao tema, implementação computacional em linguagem C e análise prática de sinais reais e simulados. Os alunos foram guiados em atividades práticas, utilizando códigos específicos para a DFT e FFT, com o objetivo de identificar frequências dominantes, remover ruídos e interpretar espectros de frequência. Durante o processo, foram elaborados relatórios técnicos para consolidar o aprendizado e promover a reflexão crítica sobre os métodos utilizados e os resultados obtidos. A linguagem C foi escolhida pela sua eficiência em cálculos numéricos e pela disponibilidade de bibliotecas específicas, como a FFTW, que simplificam a aplicação prática das transformadas. Os gráficos e os dados analisados foram gerados para simular cenários reais, permitindo a conexão entre os fundamentos teóricos e suas aplicações práticas.
- **Considerações Finais:** Um resumo dos principais aprendizados e desafios enfrentados, conectando os resultados, conectando os resultados obtidos ao contexto das aplicações práticas.

Esse relatório é aplicado para desenvolver a capacidade dos alunos de refletir sobre o aprendizado, relatar como foi o processo de estudo e aprendizado e interpretar o resultado

de forma crítica. Além do mais, é algo que incentiva a organização das ideias e desenvolvimento de habilidades de comunicação científica. Porém este relatório não precisa ter informações extremamente específicas, deve ter as informações essenciais e necessárias de maneira objetiva.

A relação entre teoria e prática é fundamental nos estudos da DFT e na FFT, pois permite que alunos compreendam não apenas os conceitos matemáticos que sustentam essas transformadas, mas também como aplicá-las em problemas reais. A teoria fornece a base necessária para entender a transformação de sinais no domínio do tempo para o domínio da frequência, o papel das frequências dominantes e importância dos algoritmos eficientes como a FFT. A prática estabiliza esse conhecimento ao desafiar os alunos a implementar essas transformadas, utilizando ferramentas como linguagem C e bibliotecas especializadas, e analisar sinais reais.

Essa relação ocorre de forma natural quando os alunos utilizam a teoria para interpretar os resultados obtidos na prática, como reconhecer frequências dominantes em espectros ou avaliar eficácia de filtros. Ao mesmo tempo, a prática tem impacto significativo na teoria, permitindo que os alunos explorem conceitos como simetria hermitiana e complexidade computacional de maneira aplicada. Esse equilíbrio entre teoria e prática garante um ótimo aprendizado, relacionando os conteúdos estudados às aplicações reais e preparando os alunos para enfrentar desafios em contextos acadêmicos e profissionais.

3.4 Aplicação

A situação final é a etapa da sequência didática em que os alunos vão resumir o conhecimento adquirido, aplicando-o problemas práticos que refletem o uso da DFT e FFT em contextos reais. Nesta fase, o objetivo é avaliar o aprendizado de maneira abrangente, promovendo a integração dos conceitos teóricos e das habilidades práticas desenvolvidas ao longo do trabalho.

Os alunos são desafiados a resolver problemas que simulam situações reais em diferentes áreas, como processamento de áudio, análise de imagens e remoção de ruídos. Por exemplo:

- **Áudio com Interferência:** Um arquivo de áudio contendo ruídos de baixa frequência é analisado pelos alunos. Utilizando a DFT, eles reconhecem as frequências dominantes relacionadas ao sinal principal e ao ruído, é aplicado um filtro e o sinal original é reconstruído no domínio do tempo, avaliando a qualidade do filtro. O mesmo é feito com a FFT.
- **Sinais Senoidais com Ruído Aleatório:** Um conjunto de funções senoidais como:

$$g(t) = \cos(\omega t + \text{Ruído})$$

é fornecido. Os alunos fazem uma análise com a DFT e a FFT para identificar frequências dominantes e aplicar filtros para remover ruídos específicos, como frequências abaixo de um limite de corte.

Após a realização das atividades, os alunos participam de uma discussão contanto com a participação do professor e avaliada por ele. Nesse momento eles compartilham os gráficos feitos e os resultados obtidos, avaliando a eficácia das transformadas aplicadas e dos filtros utilizados. A discussão envolve interpretação dos gráficos, como identificar frequências dominantes e ruídos no espectro de frequência, serão feitos gráficos em escala linear e na escala logarítmica.

Eficácia dos filtros, comparação de diferentes métodos de filtragem e análise de como o limite de corte impacta o sinal reconstruído. E uma reflexão do impacto da complexidade

computacional, uma reflexão sobre a eficiência da FFT em comparação com a DFT para sinais maiores ou com maior densidade de dados.

O compartilhamento de experiências aumenta a compreensão dos alunos, permitindo que aprendam uns com os outros, deixando suas análises mais elaboradas.

A avaliação final de uma sequência didática será composta por duas atividades complementares: a elaboração de um relatório final e uma apresentação em slides. Essas atividades tem objetivo de firmar o aprendizado, permitindo que os alunos relacionem a teoria e a prática e para desenvolverem habilidades da comunicação escrita e oral.

O relatório final deverá ser um trabalho feito de maneira detalhada, onde os alunos deverão, um documento em que os alunos deverão apresentar o processo completo da análise de sinais utilizando a DFT e FFT. O trabalho deve conter uma introdução ao conteúdo, fundamentação teórica, descrição detalhada da metodologia empregada, análise dos sinais considerados, apresentação dos gráficos gerados com e sem a presença de ruídos, bem como uma interpretação abrangente dos resultados obtidos. É importante que os alunos reflitam sobre as dificuldades que eles enfrentaram ao longo das aulas, os métodos utilizados para remover ruídos e as relações entre a teoria e a prática. As considerações finais deve sintetizar os principais aprendizados.

A apresentação em slides tem o objetivo de desenvolver a comunicação oral e a visualização de dados. Nela os alunos devem expor os principais resultados do relatório, destacando os objetivos, os métodos aplicados, os gráficos e reflexões mais importantes. A apresentação deve ser clara e objetiva, com tempo limite de 15 minutos, seguida de um breve momento para perguntas e respostas.

Essas atividades serão avaliadas levando em consideração critérios como organização, profundidade de análise, clareza na interpretação dos resultados, qualidade dos gráficos e dos slides, e a qualidade da oratória dos alunos. A combinação dessas duas maneiras de avaliar garante uma abordagem completa, sendo possível compreender a parte teórica e sua aplicação.

O encerramento das aulas marca o fim de uma sequência didática, os alunos são convidado a refletirem sobre os conhecimentos adquiridos, estabelecendo relações entre os conceitos teóricos abordados e as atividades práticas realizadas. . O professor apresenta de maneira bem objetiva os principais pontos trabalhados, com as diferenças entre a DFT e a FFT e

suas aplicações práticas, e a remoção de ruídos de sinais.

Esse momento também vai contar um um rápido *feedback*, dando ênfase para os avanços da turma dando sugestões de melhorias de próximas atividades para futuras aulas. As considerações finais das aulas reforça a importância do tema estudado, preparando os alunos para aplicar o conhecimento adquirido em novas situações e encerra a sequência de maneira clara e objetiva [22].

Capítulo 4

Considerações Finais

O presente trabalho alcançou o objetivo de explorar a fundamentação didática da DFT e da FFT, destacando sua aplicação prática na análise e no processamento de sinais. O desenvolvimento de uma sequência didática para o ensino superior possibilitou conectar teoria e prática de maneira estruturada e significativa. Contudo, é importante destacar que este trabalho não foi aplicado na prática, tratando-se de uma proposta teórica elaborada para fins de planejamento e estudo.

A DFT e a FFT foram analisadas tanto do ponto de vista teórico quanto prático. A implementação em linguagem C possibilitou uma compreensão mais aprofundada das transformadas e seus algoritmos, demonstrando como a complexidade computacional afeta o desempenho no processamento de sinais. O uso de exemplos reais, como a remoção de ruídos e identificação de frequências dominantes, reforçou a relevância dessas ferramentas em diversas áreas, como física, engenharia e ciência de dados.

A sequência didática seguiu as três etapas de aprendizagem: situação inicial, trabalho de construção e aplicação. A introdução de situações-problemas reais, como a filtragem de ruídos, tem a intenção de despertar o interesse dos alunos e promover a contextualização do conteúdo. No trabalho de construção, as atividades práticas visam a implementação e análise das transformadas, enquanto a situação final busca consolidar o aprendizado por meio de relatórios e apresentações.

Em resumo, este trabalho oferece uma abordagem teórica que une teoria e prática para o ensino de conceitos avançados, como a DFT e a FFT, apresentando-se como uma proposta viável para futuros cursos ou projetos de extensão no ensino superior.

Referências

- [1] GANDULFO, Roberto Oscar. *Série de Fourier e Convergência*. Revista Matemática Universitária, v. 11, p. 27-52, 1990.
- [2] BARROS, Lucas S.; CAVALCANTE, Marcos; SOLEDADE, Paulo; SBRISSA NETO, Antônio; SIQUEIRA, Marcelo. Oscilador Harmônico Simples, Série e Transformada de Fourier e Operador de Sturm-Liouville-Uma Breve Discussão. Revista Brasileira de Ensino de Física, v. 45, 2023.
- [3] PRANDINI RICIEFI, Aguinaldo. Construindo a Série de Fourier. São Paulo: Edições Prandiano, 2006.
- [4] CERQUEIRA, Eduardo O.; POPPI, Ronei J.; KUBOTA, Lauro T.; MELLO, Cesar. Utilização de Filtro de Transformada de Fourier Para a Minimização de Ruídos em Sinais Analíticos. Química Nova, v. 23, n. 5, p. 690-698, 2000.
- [5] GALVÃO, Roberto Kawakami Harrop; ARAÚJO, Mário César Ugulino de; SALDANHA, Teresa Cristina Bezerra; VISANI, Valeria; PIMENTEL, Maria Fernanda. Estudo Comparativo Sobre Filtragem de Sinais Instrumentais Usando Transformada de Fourier e Wavelet. Química Nova, v. 24, n. 6, p. 874-884, 2001.
- [6] OPPENHEIM, Alan V.; SCHAFER, Ronald W. Discrete-time Signal Processing. 3. ed. Pearson Education Limited, 2014.
- [7] WINOGRAD, Shmuel. On Computing The Discrete Fourier Transform. Mathematics of Computation, v. 32, n. 141, p. 175-199, 1978.
- [8] POAZ, Borat. A Course in Digital Signal Processing. 1. ed. John Wiley & Sons Inc., 1997.
- [9] SOUZA, José Luiz da Silva; COSTA, Reynerth Pereira da; MIRANDA, Francisco Otávio. Utilização da Transformada Rápida de Fourier e Wavelet Para Análise e Di-

mensionamento de Vórtice em Séries Temporais Reais Obtidas na Atmosfera Tropical. *Brazilian Journal of Development*, v. 5, n. 6, p. 5112-5127, 2019.

[10] PAPLINSKI, Janus P.; CARIOW, Aleksandr. 10-Point DFT Algorithm for Power System Harmonic Analysis. *Applied Sciences*, v. 11, n. 15, p. 7007-7021, 2021.

[11] OPPENHEIM, Alan V.; SCHAFER, Ronald W. *Discrete-time Signal Processing*. 3. ed. Pearson Education Limited, 2014.

[12] SIEBERT, William McC. *Circuits, Signals, and Systems*. 1. ed. The MIT Press McGraw-Hill Book Company, 1986.

[13] POAZ, Borat. *A Course in Digital Signal Processing*. 1. ed. John Wiley & Sons Inc., 1997.

[14] OLIVEIRA, José R. de; LIMA, Juliano B. Cálculo da Transformada Discreta de Fourier Fracionária com Complexidade Aritmética Reduzida. In: SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES E PROCESSAMENTO DE SINAIS, 37., 2019, Rio de Janeiro. *Anais [...]*. Rio de Janeiro: Sociedade Brasileira de Telecomunicações, p. 1-5.

[15] WINOGRAD, Shmuel. On Computing The Discrete Fourier Transform. *Mathematics of Computation*, v. 32, n. 141, p. 175-199, 1978.

[16] TOLIMIERI, Richard; AN, Myoung; LU, Chao. *Algorithms for Discrete Fourier Transform and Convolution*. 1. ed. Springer Science+Business Media, LLC, 1989.

[17] WANG, Zhongde. Fast Algorithms for the Discrete W Transform and for the Discrete Fourier Transform. *Speech*, v. 32, n. 4, p. 803-816, 1984.

[18] HEIDEMAN, Michael T.; JOHNSON, Don H.; BURRUS, C. Sidney. Gauss and the History of the Fast Fourier Transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. 34, n. 3, p. 265-277, 1985.

[19] SCHOUKENS, J.; PINTELON, R.; VAN HAMME, H. The Interpolated Fast Fourier Transform: A Comparative Study. *IEEE Transactions on Instrumentation and Measurement*, v. 41, n. 2, p. 226-232, 1992.

[20] ZABALA, Antoni. *A prática educativa: Como Ensinar*. Artmed, edição digital, 2014.

[21] ARAÚJO, Denise Lino. O que é (e como faz) sequência didática? *Entrepalavras*, v. 3, n. 1, p. 322-334, 2013.

[22] UGALDE, Maria Cecília Pereira; ROWDER, Charlys. Sequência Didática: Uma Proposta Metodológica de Ensino-Aprendizagem. Revista de Estudos e Pesquisa sobre Ensino Tecnológico, v. 6, Edição Especial, p. e099220, 2020.

Apêndice

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>
4 #include <time.h>
5 #include <gsl/gsl_rng.h>
6
7 int main(int argc, char **argv){
8     if(argc != 4){
9         printf("%s A f n \n \n nem que A é amplitude, f é a frequência e n
10             é o número de pontos, todos devem ser números inteiros \n",
11             argv[0]);
12         exit(1);
13     }
14     int i, A, f, n;
15     double t, Delta;
16     A = atoi(argv[1]); //amplitude da função cosseno
17     f = atoi(argv[2]); //freq. da função cosseno
18     n = atoi(argv[3]); //número de pontos da função cosseno
19
20     char nome[100];
21     sprintf(nome, "%d-cos-%d-noise.dat", A, f);
22     FILE *arquivo;
23     arquivo = fopen(nome, "w");
24
25     gsl_rng_default_seed = time(NULL);
26     gsl_rng *rand= gsl_rng_alloc(gsl_rng_taus);
27
28     t = 0.0;
29     Delta = (2.0*M_PI - (2.0*M_PI/(n-1)))/(n-1); //Delta é o intervalo
30         entre dois tempos t. Note que reduzimos um valor de Delta para
```

```

        que termine antes do valor 2*pi.
28     for(i = 0; i < n; i++){
29         fprintf(arquivo, "%+e\n", t, A*cos(t*f) + (2.0*
        gsl_rng_uniform(rand)-1.0));
30         t += Delta;
31     }
32     fclose(arquivo);
33     gsl_rng_free(rand);
34     return 0;
35 }

```

Listing 4.1: Código para Geração do Gráfico dos Quatro Sinais com Ruído Antes da Filtragem e Transformada

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4
5  int main(int argc, char **argv){
6      if(argc != 4){
7          printf("%s A f n \n\n nem que A é amplitude, f é a
              frequência e n é o número de pontos, todos devem
              ser números inteiros\n", argv[0]);
8          exit(1);
9      }
10     int i, A, f, n;
11     double t, Delta;
12     A = atoi(argv[1]); //amplitude da função cosseno
13     f = atoi(argv[2]); //freq. da função cosseno
14     n = atoi(argv[3]); //número de pontos da função cosseno
15
16     char nome[100];
17     sprintf(nome, "%d-cos-%d.dat", A, f);
18     FILE *arquivo;
19     arquivo = fopen(nome, "w");
20
21     t = 0.0;
22     Delta = (2.0*M_PI - (2.0*M_PI/(n-1)))/(n-1); //Delta é o
        intervalo entre dois tempos t. Note que reduzimos um valor

```

```

        de Delta para que termine antes do valor 2*pi.
23     for(i = 0; i < n; i++){
24         fprintf(arquivo, "%+e\n", t, A*cos(t*f));
25         t += Delta;
26     }
27     fclose(arquivo);
28     return 0;
29 }

```

Listing 4.2: Código para Geração do Gráfico dos Quatro Sinais com Ruído Após da Filtragem e Transformada

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <math.h>
5  #include <complex.h>
6
7  int main(int argc, char **argv){
8      if(argc != 3){
9          printf("%s\nnome_do_arquivo_de_dados.dat\nn_de_linhas\n"
10             , argv[0]);
11          exit(1);
12      }
13
14      int f, n, t;
15      double Delta_t;
16      double *t_t;
17      double *g_t;
18      double complex *g_f;
19      char nome[100];
20      FILE *arquivo;
21
22      n = atoi(argv[2]);
23      t_t = (double *) calloc(n, sizeof(double));
24      g_t = (double *) calloc(n, sizeof(double));
25      g_f = (double complex *) calloc(n/2, sizeof(double complex));
26
27      if(!(arquivo = fopen(argv[1], "r"))){

```

```

27         printf("não foi possível abrir o arquivo %s\n", argv
28                [1]);
29         exit(1);
30     }
31     for(t = 0; t < n; t++){
32         fscanf(arquivo, "%lf%lf", &t_t[t], &g_t[t]);
33     }
34     fclose(arquivo);
35
36     for(f = 0; f < n/2; f++){
37         for(t = 0; t < n; t++){
38             g_f[f] += g_t[t]*cexp(-2.0*M_PI*I*f*t/n);
39         }
40         g_f[f] *= 2.0/n;
41     }
42     g_f[0] *= 0.5;
43
44     char *base = strtok(argv[1], ".");
45     strcpy(nome, base);
46     strcat(nome, "-g_f-dft.dat");
47     arquivo = fopen(nome, "w");
48     Delta_t = t_t[1] - t_t[0];
49     for(f = 0; f < n/2; f++){
50         fprintf(arquivo, "%d%e%e%e%e\n", f, f/(Delta_t*n),
51                cabs(g_f[f]*g_f[f]), creal(g_f[f]), cimag(g_f[f]))
52                ;
53     }
54     fclose(arquivo);
55
56     for(t = 0; t < n; t++){
57         g_t[t] = 0.0;
58         for(f = 0; f < n/2; f++){
59             if(f < 6){
60                 g_t[t] += creal(g_f[f]*cexp(2.0*M_PI*I
61                        *f*t/n));

```

```

62     strcpy(nome, base);
63     strcat(nome, "-g_i-dft.dat");
64     arquivo = fopen(nome, "w");
65     for(t = 0; t < n; t++){
66         fprintf(arquivo, "%e\te\n", t_t[t], g_t[t]);
67     }
68     fclose(arquivo);
69
70     free(t_t);
71     free(g_t);
72     free(g_f);
73     return 0;
74 }

```

Listing 4.3: Implementação da DFT em Linguagem C

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <math.h>
5  #include <complex.h>
6  #include <fftw3.h>
7
8  int main(int argc, char **argv) {
9      if(argc != 3) {
10         printf("%s\tnome_do_arquivo_de_dados.dat\tn_de_linhas\n"
11             , argv[0]);
12         exit(1);
13     }
14
15     int f, n, t;
16     double Delta_t;
17     double *inp, *t_t;
18     char nome[100];
19     FILE *arquivo;
20
21     n = atoi(argv[2]);
22     inp = (double *) calloc(n, sizeof(double));
23     t_t = (double *) calloc(n, sizeof(double));

```

```

23
24     if(!(arquivo = fopen(argv[1], "r"))){
25         printf("não foi possível abrir o arquivo %s\n", argv
26             [1]);
27         exit(1);
28     }
29     for(t = 0; t < n; t++){
30         if(fscanf(arquivo, "%lf%lf", &t_t[t], &inp[t]) != 2){
31             printf("não foi possível ler o arquivo\n");
32             exit(1);
33         }
34     }
35     fclose(arquivo);
36
37     fftw_complex *g_t = (fftw_complex *) fftw_malloc(sizeof(
38         fftw_complex)*n);
39     fftw_complex *g_f = (fftw_complex *) fftw_malloc(sizeof(
40         fftw_complex)*n);
41
42     for(t = 0; t < n; t++) {
43         g_t[t] = inp[t] + 0.0*I;
44     }
45     free(inp);
46
47     fftw_plan FTF = fftw_plan_dft_1d(n, g_t, g_f, FFTW_FORWARD,
48         FFTW_ESTIMATE);
49     fftw_execute(FTF);
50     fftw_destroy_plan(FTF);
51
52     g_f[0] /= n;
53     for(f = 1; f < n; f++){
54         g_f[f] /= (n/2);
55     }
56
57     char *base = strtok(argv[1], ".");
58     strcpy(nome, base);
59     strcat(nome, "-g_f-fft.dat");
60     arquivo= fopen(nome, "w");
61     Delta_t = t_t[1] - t_t[0];

```

```

58     for(f = 0; f < n/2; f++){
59         fprintf(arquivo, "%d□e□e□e□e\n", f, f/(Delta_t*n),
60             cabs(g_f[f]*g_f[f]), creal(g_f[f]), cimag(g_f[f]))
61             ;
62     }
63     fclose(arquivo);
64
65     for(f = 0; f < n; f++){
66         if(f < 6){
67             g_f[f] = 0.0 + 0.0*I;
68         }
69     }
70     fftw_plan FTB = fftw_plan_dft_1d(n, g_f, g_t, FFTW_BACKWARD,
71         FFTW_ESTIMATE);
72     fftw_execute(FTB);
73     fftw_destroy_plan(FTB);
74
75     strcpy(nome, base);
76     strcat(nome, "-g_i-fft.dat");
77     arquivo= fopen(nome, "w");
78     for(t = 0; t < n; t++){
79         fprintf(arquivo, "%e□e\n", t_t[t], creal(g_t[t]));
80     }
81     fclose(arquivo);
82
83     free(t_t);
84     fftw_free(g_t);
85     fftw_free(g_f);
86     fftw_cleanup();
87
88     return 0;
89 }

```

Listing 4.4: Implementação da FFT utilizando a biblioteca FFTW

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>

```

```

4 #include <gsl/gsl_rng.h>
5 #include <time.h>
6
7 double f(double x){
8     return cos(x);
9 }
10
11 int main(int argc, char **argv){
12     if(argc != 5){
13         printf("Digite\n\n%s x_i x_f n nome.dat\n\n", argv[0]);
14         exit(1);
15     }
16     int i, n;
17     double x, Delta;
18     FILE *xr;
19     n = atoi(argv[3]);
20     xr = fopen(argv[4], "w"); // no lugar de w - write, pode ser
        r - read ou a - append
21     x = atof(argv[1]);
22     Delta = (atof(argv[2]) - x)/(n-1);
23     for(i = 0; i < n; i++){
24         fprintf(xr, "%e %e\n", x, f(x));
25         x += Delta;
26     }
27     fclose(xr);
28     return 0;
29 }

```

Listing 4.5: Código para geração de pontos de uma função coseno em um intervalo definido, com valores igualmente espaçados