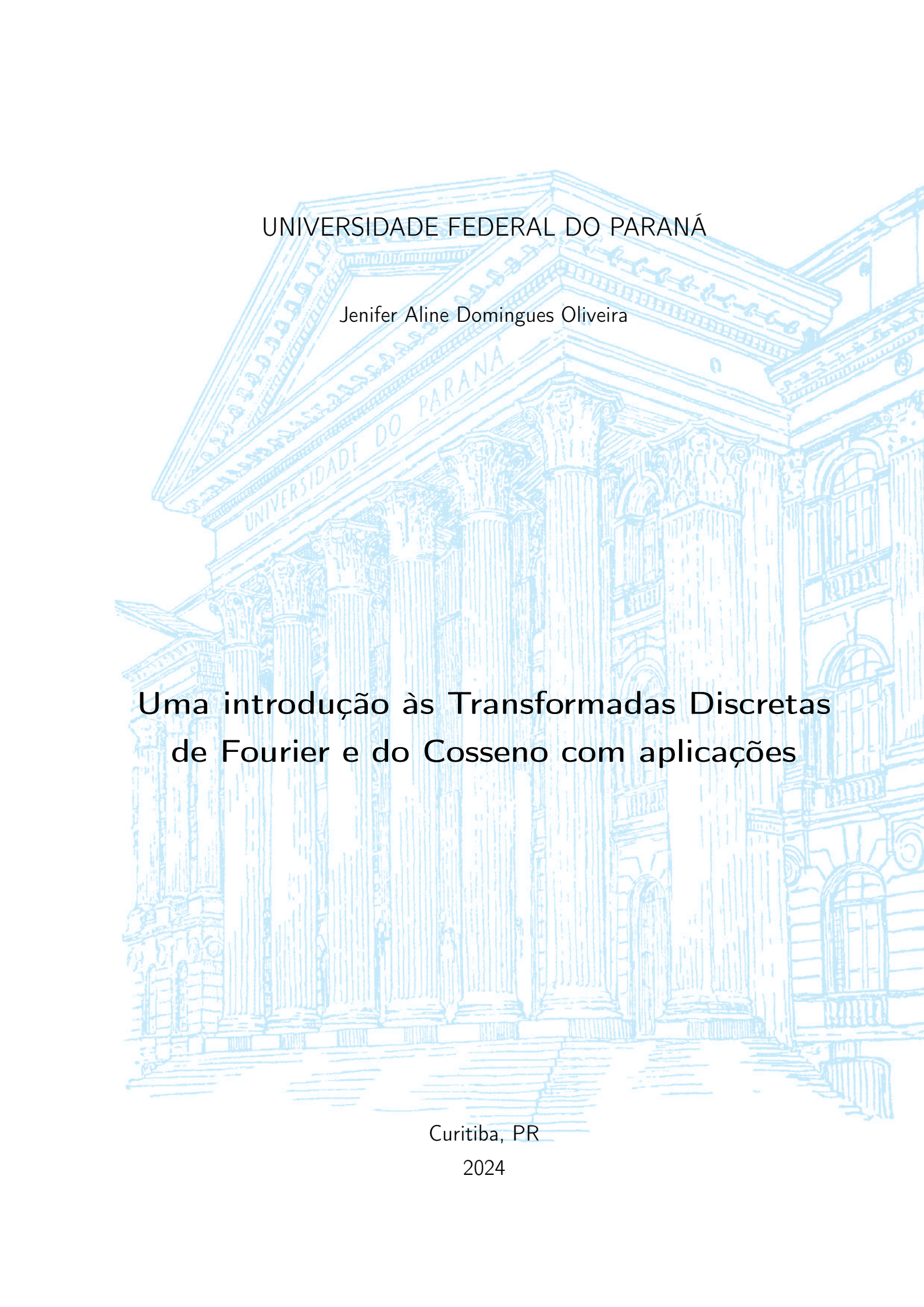




UNIVERSIDADE FEDERAL DO PARANÁ

Jenifer Aline Domingues Oliveira

Uma introdução às Transformadas Discretas de Fourier e do Cosseno com aplicações

Curitiba, PR

2024

Jenifer Aline Domingues Oliveira

Uma introdução às Transformadas Discretas de Fourier e do Cosseno com aplicações

–Trabalho acadêmico apresentado como requisito parcial para aprovação do curso de Matemática Industrial.

Universidade Federal do Paraná (UFPR)
Departamento de Matemática
Bacharelado em Matemática Industrial

Orientador: Prof^a Ana Gabriela Martinez

Curitiba, PR
2024

Agradecimentos

Agradeço à minha família, pelo amor incondicional, pelo apoio inestimável e por sempre acreditarem no meu potencial. Aos meus pais, Geraldo Machado e Elizangela Oliveira, por todo o incentivo e paciência ao longo dessa jornada.

A minha orientadora, Ana Gabriela Martínez, pela orientação, paciência, sabedoria e dedicação. Sua orientação foi essencial para o desenvolvimento deste trabalho. Agradeço também aos professores e funcionários do departamento de Matemática da Unniversidade Federal do Paraná, pelo conhecimento transmitido e pelo suporte durante a minha trajetória acadêmica.

Resumo

A Transformada Discreta de Fourier (DFT) e a Transformada Discreta do Cosseno (DCT) são ferramentas matemáticas fundamentais no campo do processamento de sinais e na análise de frequências. Este trabalho explora algumas das principais propriedades matemáticas das transformadas DFT e DCT, bem como suas aplicações na eliminação de ruídos em sinais unidimensionais e compressão de imagens digitais. Será feita uma comparação entre as duas transformadas, discutindo suas vantagens e desvantagens em diferentes contextos de uso.

Palavras-chave: DFT, DCT, processamento de sinais, compressão de imagens.

Resumo

The Discrete Fourier Transform (DFT) and the Discrete Cosine Transform (DCT) are fundamental mathematical tools in the field of signal processing and frequency analysis. This work explores the mathematical properties of the transformations DFT and DCT, as well as their applications in eliminating noise reduction in one dimensional signal and digital image compression. A comparison between the two transforms, discussing their advantages and disadvantages in different contexts of use.

Keywords: DFT, DCT, signal processing, image compression.

SUMÁRIO

Sumário	5
1 INTRODUÇÃO	7
2 PRELIMINARES	9
2.1 Série de Fourier	9
2.2 Forma Complexa da Série de Fourier	9
2.3 Convergência da série de Fourier	11
2.4 Ortogonalidade das exponenciais complexas	12
3 TRANSFORMADA DISCRETA DE FOURIER	
3.1 Aproximação da DFT para os coeficientes da Série de Fourier	17
3.2 Aproximação da Transformada Discreta de fourier para a Transformada de Fourier	18
3.3 Forma matricial da DFT	21

3.4	A DFT como aproximação trigonométrica	24
3.5	Propriedades da transformada Discreta De Fourier (DFT)	27
3.5.1	Periodicidade:	28
3.5.2	Linearidade:	28
3.5.3	Deslocamento da DFT:	28
3.5.4	Modulação da DFT:	29
3.5.5	Simetria Hermitiana:	29
3.5.6	DFT de uma sequência real:	30
3.5.7	Simetria conjugada na DFT:	30
3.5.8	Convolução Discreta (Cíclica)	30
3.5.9	Relação de Parseval	32
3.6	Uma aplicação da DFT no processamento de sinais	33
4	TRANSFORMADA DISCRETA DO COSSENO DCT	37
4.1	Matriz da DCT	38
4.2	Relação entre DFT e DCT	39
4.3	Ortogonalidade das funções bases	41
4.4	DCT bidimensional	44
4.5	Propriedades da DCT:	44
4.5.1	Decorrelação:	44
4.5.2	Compactação de energia:	46
4.5.3	Separabilidade:	46
4.5.4	Simetria:	47
4.6	Compressão de imagens usando a DCT	47
4.7	Outras variantes para a DCT	49
4.8	As transformadas padrão do cosseno	52
4.9	Uso da DCT para compressão de imagens	53
5	CONCLUSÕES	61
	REFERÊNCIAS	63
A	CÓDIGO DA APLICAÇÃO DFT	65
B	CÓDIGO DA APLICAÇÃO DCT	67

1 INTRODUÇÃO

A Transformada Discreta de Fourier transforma uma sequência de valores amostrais em componentes de frequência, proporcionando uma representação no domínio da frequência que é essencial para entender a estrutura dos sinais e para realizar operações de filtragem e modulação. Por outro lado, a Transformada Discreta do Cosseno, que é uma variação da DFT, é especialmente eficaz na compressão de dados devido à sua capacidade de concentrar a energia do sinal em um pequeno número de coeficientes.

A importância do estudo dessas transformadas é destacada por suas aplicações em tecnologias amplamente utilizadas, como os formatos de compressão de mídia digital como arquivos de áudio e imagens. Compreender as propriedades e as limitações dessas transformadas é crucial para desenvolver e melhorar técnicas de processamento de sinais.

Este trabalho tem como objetivo explorar as propriedades matemáticas das transformadas DFT e DCT, bem como suas aplicações em compressão de imagens. Será feita uma comparação entre as duas transformadas, discutindo suas vantagens e desvantagens em diferentes contextos de uso.

Este TCC está organizado da seguinte forma: no capítulo 2, apresentamos os conceitos preliminares e a fundamentação teórica necessária para o entendimento das séries de Fourier. No capítulo 3, mostramos como a DFT se relaciona com os coeficientes da Série de Fourier e com a Transforma de Fourier. Apresentamos também algumas das principais propriedades da DFT, como é sua forma matricial e aplicações, analisando o uso da DFT na limpeza de sinais contaminados com ruídos. No capítulo 4, focamos na Transformada Discreta do Cosseno, abordando suas propriedades, a relação com a DFT, a ortogonalidade nas funções bases, outras variantes da DCT e por último a apresentação de uma aplicação da DCT na compressão de imagens digitais. A implementação numérica é feita em Python e os códigos se encontram no anexo.

2 PRELIMINARES

2.1 SÉRIE DE FOURIER

A Série de Fourier é uma ferramenta fundamental e amplamente estudada na matemática, utilizada para representar funções periódicas com período A com imagem real por meio de uma soma infinita de funções seno e cosseno do mesmo período. Essa representação descreve como uma função periódica pode ser decomposta em suas componentes de frequência. Esta técnica é de suma importância para a análise e o processamento de sinais. Uma série composta por senos e cossenos é denominada série trigonométrica. Tal série assume a seguinte forma:

$$f(x) \approx \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos\left(\frac{2\pi kx}{A}\right) + b_k \sin\left(\frac{2\pi kx}{A}\right), \quad (2.1)$$

onde $f(x)$ é a função periódica em questão, a_0 , a_k e b_k são os coeficientes da série, k representa a ordem dos termos na série.

Se f for integrável, os coeficientes da Série de Fourier podem ser determinados utilizando integração.

Para uma função periódica $f(x)$ com período A , os coeficientes da Série de Fourier são dados por:

$$a_0 = \frac{2}{A} \int_{-\frac{A}{2}}^{\frac{A}{2}} f(x) dx \quad (2.2)$$

$$a_k = \frac{2}{A} \int_{-\frac{A}{2}}^{\frac{A}{2}} f(x) \cos\left(\frac{2\pi kx}{A}\right) dx \quad (2.3)$$

$$b_k = \frac{2}{A} \int_{-\frac{A}{2}}^{\frac{A}{2}} f(x) \sin\left(\frac{2\pi kx}{A}\right) dx, \quad (2.4)$$

onde k é um número inteiro positivo, com $k = 1, 2, \dots$

2.2 FORMA COMPLEXA DA SÉRIE DE FOURIER

A Série de Fourier pode ser convenientemente representada utilizando funções exponenciais complexas. Essa abordagem complexa oferece várias vantagens e simplificações matemáticas que a tornam extremamente valiosa em diversos contextos.

Primeiramente, a utilização de funções exponenciais complexas facilita a manipulação algébrica das séries. As propriedades das exponenciais complexas permitem transformar somas de senoides em produtos de exponenciais, o que simplifica significativamente as

operações matemáticas envolvidas. Isso é particularmente útil na análise de sinais e sistemas, onde a decomposição de um sinal em suas componentes de frequência é essencial. Além disso, essa formulação permite o uso de ferramentas matemáticas avançadas, como a álgebra complexa e as transformadas de Fourier. A álgebra complexa oferece uma forma compacta e elegante de realizar cálculos que seriam mais complicados usando funções trigonométricas reais. As transformadas de Fourier, por sua vez, são fundamentais para a análise de sinais no domínio da frequência. Elas convertem um sinal temporal em suas componentes de frequência, facilitando a compreensão e a manipulação do sinal.

Ao utilizar funções exponenciais complexas, também é possível simplificar operações de convolução e modulação. A convolução, que no domínio do tempo pode ser uma operação complexa e computacionalmente custosa, torna-se uma simples multiplicação no domínio da frequência. Isso não apenas simplifica a matemática envolvida, mas também melhora a eficiência computacional, tornando essa abordagem ideal para aplicações práticas em processamento de sinais.

Para obter a Forma Complexa da Série de Fourier utilizaremos as relações trigonométricas provenientes da identidade de Euler:

$$\begin{aligned} e^{ix} &= \cos(x) + i \sin(x) \\ \sin(x) &= \frac{e^{ix} - e^{-ix}}{2i} = -i \frac{e^{ix} - e^{-ix}}{2} \\ \cos(x) &= \frac{e^{ix} + e^{-ix}}{2}. \end{aligned}$$

Substituindo-as na forma harmônica da Série de Fourier e supondo que há convergência da série para a função f :

$$\begin{aligned} f(x) &= \frac{a_0}{2} + \sum_{k=1}^{\infty} \left[\frac{a_k}{2} \left(e^{\frac{i2\pi kx}{A}} + e^{-\frac{i2\pi kx}{A}} \right) - \frac{ib_k}{2} \left(e^{\frac{i2\pi kx}{A}} - e^{-\frac{i2\pi kx}{A}} \right) \right] \\ f(x) &= \frac{a_0}{2} + \sum_{k=1}^{\infty} \left[\frac{a_k - ib_k}{2} e^{\frac{i2\pi kx}{A}} + \frac{a_k + ib_k}{2} e^{-\frac{i2\pi kx}{A}} \right]. \end{aligned} \quad (2.5)$$

Considerando a paridade das funções cosseno e seno:

$$\begin{aligned} \cos(-x) &= \cos(x) \\ \sin(-x) &= -\sin(x) \end{aligned}$$

De (2.3) e (2.4) resulta que:

$$\begin{aligned} a_k &= a_{-k} \\ b_k &= -b_{-k} \text{ e } b_0 = 0 \end{aligned}$$

Podemos denotar os coeficientes da seguinte maneira, $c_k = \frac{a_k - ib_k}{2}$, $c_{-k} = \frac{a_k + ib_k}{2}$, sendo $c_0 = \frac{a_0}{2}$, substituindo esses coeficientes na expressão (2.5) obtemos, a seguinte expressão:

$$f(x) = c_0 + \sum_{k=1}^{\infty} c_k e^{\frac{i2\pi kx}{A}} + \sum_{k=1}^{\infty} c_{-k} e^{\frac{-i2\pi kx}{A}}.$$

Com os coeficientes calculados através da expressão envolvendo a função periódica original multiplicada pela função exponencial e integrada ao longo do período A :

$$c_k = \frac{1}{A} \int_{-\frac{A}{2}}^{\frac{A}{2}} f(x) e^{\frac{-i2\pi kx}{A}} dx.$$

Podemos simplificar esta expressão como:

$$f(x) = \sum_{k=-\infty}^{\infty} c_k e^{\frac{i2\pi kx}{A}}.$$

Após estabelecer a expressão dos coeficientes da Série de Fourier, é crucial considerar a questão da convergência dessa série. Deve se estabelecer condições que garantem que a série resultante realmente converge para a função original que estamos tentando representar. A convergência da Série de Fourier está intrinsecamente ligada às propriedades da função em questão.

É importante ressaltar que a convergência da Série de Fourier pode ser afetada por diferentes condições, como a periodicidade e a continuidade da função. Em casos onde essas condições não são satisfeitas, a convergência da série pode ser condicional ou até mesmo divergente, exigindo abordagens adicionais para garantir uma representação precisa da função, que podem ser vistas em [8]

2.3 CONVERGÊNCIA DA SÉRIE DE FOURIER

A seguir são apresentadas algumas definições que serão necessárias para formular o teorema da convergência da Série de Fourier.

Limite à esquerda: O limite lateral de uma função $f(x)$ quando x se aproxima de c pela esquerda é denotado por:

$$f(x^-) = \lim_{x \rightarrow c^-} f(x). \quad (2.6)$$

Limite à direita: O limite lateral de uma função $f(x)$ quando x se aproxima c pela direita é denotado por:

$$f(x^+) = \lim_{x \rightarrow c^+} f(x). \quad (2.7)$$

Estes conceitos são úteis para entender o comportamento de funções em pontos de junção, especialmente em funções definidas por partes.

Função Suave Por Partes

Uma função suave por partes é uma função que é diferenciável em cada um dos intervalos em que é definida, mas pode ter pontos onde a derivada não é contínua.

Uma função $f(x)$ é seccionalmente suave, ou seja suave por partes, quando tanto $f(x)$ quanto sua derivada $f'(x)$ são seccionalmente contínuas. Isso significa que:

$f(x)$ é contínua em cada subintervalo, exceto possivelmente em um número finito de pontos x_n onde existem limites laterais finitos, denotados por $f(x^+)$ e $f(x^-)$.

A derivada $f'(x)$ é contínua em cada subintervalo, exceto possivelmente em um número finito de pontos x_n onde existem os limites laterais finitos, denotados por $f(x^+)$ e $f(x^-)$.

Em outras palavras, a função $f(x)$ e sua derivada $f'(x)$ podem não estar definidas em um número finito de pontos, mas seus limites laterais devem existir nesses pontos.

Teorema 1 (Convergência da Série de Fourier). *Se f for uma função suave por partes e periódica, então a série de Fourier de f converge pontualmente para cada x para o valor*

$$\frac{f(x^+) + f(x^-)}{2}$$

A demonstração do teorema pode ser encontrada no livro do Djairo Guedes [8].

Uma propriedade muito importante dos coeficientes da Série De Fourier tendem a zero quando k tende para infinito, ou seja:

$$a_k = \frac{2}{A} \int_{-\frac{A}{2}}^{\frac{A}{2}} f(x) \cos\left(\frac{2\pi kx}{A}\right) dx \rightarrow 0 \quad \text{quando } k \rightarrow \infty$$

e

$$b_k = \frac{2}{A} \int_{-\frac{A}{2}}^{\frac{A}{2}} f(x) \sin\left(\frac{2\pi kx}{A}\right) dx \rightarrow 0 \quad \text{quando } k \rightarrow \infty.$$

Esse resultado é conhecido pelo nome de Lema de Reimann-Lebesgue, que pode ser encontrado no livro [6]

2.4 ORTOGONALIDADE DAS EXPONENCIAIS COMPLEXAS

A ortogonalidade das funções bases é um conceito fundamental na teoria das séries de Fourier, tanto para a Série de Fourier Real quanto para a Série de Fourier Complexa. Na Série de Fourier Complexa, a ortogonalidade está relacionada às funções exponenciais complexas que são usadas como base para expandir uma função periódica, a base de funções consiste em exponenciais complexas da forma $e^{\frac{i2\pi kx}{A}}$, que são ortogonais umas às outras dentro de um determinado intervalo de integração. A relação de ortogonalidade é expressa da seguinte forma:

$$\int_{-\frac{A}{2}}^{\frac{A}{2}} e^{\frac{i2\pi kx}{A}} e^{\frac{-i2\pi lx}{A}} dx = \delta_N(k - l)$$

Onde $\delta_N(k)$ é a delta de Kronecker, tal que:

$$\delta_N(k) = \begin{cases} 1, & \text{Se } k = 0 \text{ ou múltiplo de } N \\ 0, & k \neq 0. \end{cases}$$

O teorema da ortogonalidade discreta e sua demonstração são do livro [9].

Teorema 2 (Ortogonalidade discreta). *Sejam k e l números inteiros e N um número inteiro positivo, então:*

$$\sum_{n=0}^{N-1} e^{i2\pi \frac{ln}{N}} e^{-i2\pi \frac{kn}{N}} = N\delta_N(k-l). \quad (2.8)$$

Demonstração. Seja $\omega_N = e^{\frac{i2\pi}{N}}$, observe que os N números complexos $\omega_N^k = e^{\frac{i2\pi k}{N}}$ para $k = 0 : N-1$. São chamadas de enésimas raízes da unidade porque satisfazem.

$$(\omega_N^k)^N = \left(e^{\frac{i2\pi k}{N}}\right)^N = e^{i2\pi k} = 1,$$

portanto são zeros do polinômio $z^N - 1$. Fatorando esse polinômio.

$$z^N - 1 = (z - 1)(z^{N-1} + z^{N-2} + \dots + z + 1) = (z - 1) \sum_{n=0}^{N-1} z^n.$$

Há dois casos para considerarmos, se $z = \omega_N^{l-k}$ onde $k-l$ não é um múltiplo de N , então $z \neq 1$ e resulta

$$\sum_{n=0}^{N-1} z^n = \sum_{n=0}^{N-1} (\omega_N^{k-l})^n = \sum_{n=0}^{N-1} \omega_N^{(k-l)n} = 0.$$

Por outro lado, se $k-l$ é um múltiplo de N , então $\omega_N^{k-l} = 1$, assim

$$\sum_{n=0}^{N-1} \omega_N^{(k-l)n} = \sum_{n=0}^{N-1} 1 = N.$$

A propriedade (2.8) segue desses dois casos. ■

3 TRANSFORMADA DISCRETA DE FOURIER

Neste capítulo, discutimos a Transformada Discreta de Fourier, suas propriedades e aplicações.

A Transformada Discreta de Fourier (DFT) é uma ferramenta fundamental na análise de sinais e sistemas discretos. Ela transforma um sinal discreto no domínio do tempo em seu equivalente no domínio da frequência, permitindo a análise das componentes espectrais do sinal. Se $f = (f_n)_{n=-\frac{N}{2}+1}^{\frac{N}{2}}$ denota uma sequência de comprimento finito N par, define-se a DFT de tamanho N pela expressão.

$$F_k = \frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f_n e^{-\frac{i2\pi nk}{N}}, \quad k = -\frac{N}{2} + 1 : \frac{N}{2}. \quad (3.1)$$

Por conveniência, adotaremos a notação

$$\omega_N = e^{\frac{i2\pi}{N}},$$

logo,

$$\omega_N^{-nk} = e^{-\frac{i2\pi nk}{N}}$$

e

$$\omega_N^{nk} = e^{\frac{i2\pi nk}{N}}.$$

Com a notação em questão, podemos reescrever a Definição da Transformada Discreta de Fourier da seguinte maneira:

$$F_k = \frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f_n \omega_N^{-nk}, \quad k = -\frac{N}{2} + 1 : \frac{N}{2} \quad (3.2)$$

Existe uma versão análoga que se aplica quando N é ímpar. Se N é um número inteiro positivo ímpar e f_n uma sequência de números complexos de comprimento N , onde $n = \frac{-(N-1)}{2} : \frac{(N-1)}{2}$, então sua transformada discreta de Fourier é outra sequência de números complexos de comprimento N dado por

$$F_k = \frac{1}{N} \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} f_n \omega_N^{-nk}, \quad k = -\frac{(N-1)}{2} : \frac{(N-1)}{2}.$$

Transformada Discreta De Fourier Inversa IDFT

A Transformada Discreta de Fourier Inversa (IDFT) é uma operação matemática que permite reconstruir um sinal no domínio original a partir de seu espectro no domínio da frequência.

Seja N um número inteiro par positivo e seja F_k uma sequência de N números complexos, onde $k = \frac{-N}{2} + 1 : \frac{N}{2}$. Então a Transformada Discreta de Fourier Inversa é outra sequência de números complexos, dada por.

$$f_n = \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} F_k \omega_N^{nk}, \quad k = \frac{-N}{2} + 1 : \frac{N}{2}$$

.

Assim como na DFT, frequentemente utilizaremos a notação de operador $D^{-1}\{F_k\}$ para representar a IDFT da sequência F_k e $D^{-1}\{F_k\}_n$ para denotar o n -ésimo elemento da transformada inversa, portanto $D^{-1}\{F_k\}_n = f_n$.

A notação e a discussão anteriores sugerem que a DFT e a IDFT são inversas uma da outra, porém este fato ainda não foi formalmente demonstrado. Portanto, a próxima tarefa é estabelecer que os operadores D e D^{-1} satisfazem as relações inversas.

$$D^{-1}\{D\{f_n\}\}_n = f_n$$

e

$$D\{D^{-1}\{F_k\}\}_k = F_k.$$

Com o auxílio da propriedade da ortogonalidade, estamos preparados para investigar a relação inversa entre a DFT e a IDFT.

Teorema 3 (Relação inversa). *Seja f_n uma sequência de número complexos e seja $F_k = D\{f_n\}_k$ a DFT desta sequência. Então $D^{-1}\{D\{f_n\}_k\}_n = f_n$.*

Demonstração.

$$\begin{aligned} D^{-1}\{D\{f_n\}_k\}_n &= \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} F_k \omega_N^{nk} \\ &= \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} \left(\frac{1}{N} \sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} f_j \omega_N^{-jk} \right) \omega_N^{nk} \\ &= \frac{1}{N} \sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} f_j \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} \omega_N^{k(n-j)} \end{aligned}$$

Ao empregar a propriedade (2.8) do teorema 2, observamos que a soma interna nesta equação é não nula apenas quando $j = n$ soma externa, resultando em:

$$D^{-1}\{D\{f_n\}_k\}_n = \frac{1}{N} N f_n = f_n$$

■

A seguir apresentaremos três abordagens distintas para derivar a expressão da Transformada Discreta de Fourier (DFT): a Transformada de Fourier, a Série de Fourier Complexa e a Aproximação Trigonométrica.

3.1 APROXIMAÇÃO DA DFT PARA OS COEFICIENTES DA SÉRIE DE FOURIER

Com a introdução feita na seção anterior à Série de Fourier, estamos prontos para derivar a Transformada Discreta de Fourier (DFT) como uma aproximação da integral que fornece os coeficientes da série de Fourier c_n . Consideramos as aproximações para a integral.

$$c_k = \frac{1}{A} \int_{-A/2}^{A/2} f(x) e^{\frac{i2\pi kx}{A}} dx \quad (3.3)$$

Consideraremos o intervalo de integração subdividido em N subintervalos de igual comprimento, e o espaçamento da grade de $\Delta x = \frac{A}{N}$. Uma grade com $N + 1$ pontos igualmente espaçados no intervalo $[-\frac{A}{2}; \frac{A}{2}]$ é definida pelos pontos $x_n = n\Delta x$ para $n = \frac{-N}{2}, \dots, \frac{N}{2}$. Se chamarmos

$$g(x) = f(x) e^{\frac{-i2\pi kx}{A}},$$

ao aplicar a regra do trapézio, obtemos a seguinte aproximação para a integral

$$c_n = \frac{1}{A} \int_{-\frac{A}{2}}^{\frac{A}{2}} g(x) dx \approx \frac{1}{A} \frac{\Delta X}{2} \left[g\left(-\frac{A}{2}\right) + 2 \sum_{n=\frac{N}{2}+1}^{\frac{N}{2}-1} g(x_n) + g\left(\frac{A}{2}\right) \right]$$

Vimos no teorema 1 que se a extensão periódica de f for descontínua nos pontos finais $x = \pm \frac{A}{2}$ então, quando sua série de Fourier converge, ela converge para o valor médio

$$\frac{1}{2} \left[f\left(-\frac{A}{2}^+\right) + f\left(\frac{A}{2}^-\right) \right].$$

Portanto, é o valor médio de f nos pontos finais que deve ser usado na regra dos trapézio. Observando que o núcleo $e^{\frac{-i2\pi kx}{A}}$ tem o valor $(-1)^k$ em $x = \pm \frac{A}{2}$, vemos que a função g que deve ser usada para a regra do trapézio é

$$g(x) = \begin{cases} f(x) e^{\frac{-i2\pi kx}{A}}, & \text{Se } x \neq \pm \frac{A}{2} \\ \frac{(-1)^k}{2} \left[f\left(-\frac{A}{2}^+\right) + f\left(\frac{A}{2}^-\right) \right], & \text{Se } x = \pm \frac{A}{2}. \end{cases}$$

Deve-se verificar que esta escolha de g determinada pelas propriedades de convergência da Série de Fourier, assegura que

$$g\left(-\frac{A}{2}\right) = g\left(\frac{A}{2}\right)$$

Da mesma forma, é necessário utilizar o valor médio em qualquer ponto da grade onde f apresente descontinuidades.

Usando a definição de g juntamente com a observação de que $\frac{2\pi k x_n}{A} = \frac{2\pi k n}{N}$, conseguimos reduzir a regra do trapézio para

$$\begin{aligned} c_n &= \frac{1}{A} \int_{-\frac{A}{2}}^{\frac{A}{2}} g(x) dx \approx \frac{1}{A} \Delta x \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} g(x_n) \\ &= \frac{1}{A} \frac{A}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f(x_n) e^{\frac{-i2\pi n k}{N}}. \end{aligned}$$

Denotando $f_n = f(x_n)$, observamos que uma estimativa para o coeficiente da Série de Fourier c_n é dada por

$$c_k \approx \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f_n e^{\frac{-i2\pi n k}{N}} = D\{f_n\}_k,$$

para $k = -\frac{N}{2} + 1 : \frac{N}{2}$. Esta definição é precisamente a da Transformada Discreta de Fourier (DFT). Assim, observamos que a DFT fornece estimativas para os primeiros N coeficientes de Fourier de uma função f em um intervalo específico $[-\frac{A}{2}; \frac{A}{2}]$ de maneira natural.

3.2 APROXIMAÇÃO DA TRANSFORMADA DISCRETA DE FOURIER PARA A TRANSFORMADA DE FOURIER

A relação entre a Transformada Discreta de Fourier (DFT) e a Transformada de Fourier (TF) é de grande importância no campo do processamento de sinais. A TF é uma ferramenta utilizada para transformar funções do domínio temporal para o domínio da frequência, sendo definida para funções contínuas por meio de uma integral definida ao longo de todo o domínio. O teorema da amostragem de Nyquist-Shannon estabelece uma conexão crucial entre a DFT e a TF. Essa relação é fundamentada na capacidade de reconstrução completa de um sinal contínuo a partir de suas amostras discretas. Enquanto a TF opera em sinais contínuos e infinitos, a DFT lida com sinais discretos e finitos. No entanto, ambas proporcionam análises detalhadas dos sinais no domínio da frequência. Podemos considerar a DFT como uma forma discreta da TF, permitindo uma abordagem eficiente e computacionalmente viável para analisar sinais no domínio da frequência. Essa relação estabelece uma base sólida para o desenvolvimento de técnicas de processamento de sinais e fornece resultados importantes para uma ampla gama de aplicações práticas, desde comunicações até processamento de áudio e imagem. Suponhamos que f esteja definido no intervalo $(-\infty, \infty)$ e possua algumas propriedades conhecidas, incluindo ser

uma função absolutamente integrável na reta real. Isso implica que:

$$\int_{-\infty}^{\infty} |f(x)| dx \leq \infty.$$

Então podemos definir uma função $\hat{f}(\omega)$ por

$$\hat{f}(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} f(x) e^{-i2\pi\omega x} dx.$$

Pode-se afirmar que a transformada $\hat{f}$ está definida no domínio de frequência, enquanto a função de entrada f é considerada definida no domínio espacial se x for uma coordenada espacial, ou no domínio do tempo se f for uma função dependente do tempo. É de extrema importância ressaltar que existe também uma relação inversa entre f e $\hat{f}$ dada por

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\omega) e^{i2\pi\omega x} d\omega$$

Essa relação estabelece f como a Transformada Inversa de Fourier de $\hat{f}(\omega)$.

Para vermos como a DFT emerge como uma aproximação natural. Primeiramente, uma observação prática é necessária: quando uma função é dada com uma extensão limitada (por exemplo, f pode representar uma imagem com limites bem definidos) ou, por questões de cálculo, f deve ser assumido como zero fora de algum intervalo finito. Por enquanto, suponhamos que $f(x) = 0$ para $|x| > \frac{A}{2}$. A Transformada de Fourier de tal função com extensão limitada é dada por

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi\omega x} dx = \int_{-\frac{A}{2}}^{\frac{A}{2}} f(x) e^{-i2\pi\omega x} dx.$$

Essa é a integral que queremos aproximar numericamente. Para elaborar um método de aproximação, o intervalo de integração $[-\frac{A}{2}, \frac{A}{2}]$ é dividido em N subintervalos de comprimento $\Delta x = \frac{A}{N}$. Supondo, por enquanto, que N é par, uma grade com $N + 1$ pontos igualmente espaçados é definida pelos pontos $x_n = n\Delta x$ para $n = -\frac{N}{2} : \frac{N}{2}$.

Assumimos agora que a função é conhecida nos pontos da grade. Deixando o integrando ser

$$g(x) = f(x) e^{-i2\pi\omega x}$$

Aplicando a regra do trapézio a essa integral, temos a seguinte aproximação

$$\int_{-\frac{A}{2}}^{\frac{A}{2}} g(x) dx \approx \frac{\Delta x}{2} \left[g\left(-\frac{A}{2}\right) + 2 \sum_{n=\frac{N}{2}+1}^{\frac{N}{2}-1} g(x_n) + g\left(\frac{A}{2}\right) \right].$$

Supondo $g(-\frac{A}{2}) = g(\frac{A}{2})$, assim a aproximação da regra do trapézio pode ser escrita

$$\begin{aligned}\hat{f}(\omega) &= \int_{-\frac{A}{2}}^{\frac{A}{2}} g(x)dx \approx \Delta x \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} g(x_n) \\ &= \frac{A}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f(x_n) e^{-i2\pi\omega x_n}.\end{aligned}$$

Esta aproximação pode ser avaliada para qualquer valor de Ω . Para os propósitos da DFT, é razoável escolher N valores para Ω nos quais aproximar f .

A questão de quais valores de frequência usar requer uma discussão de fundamental importância para a DFT, pois leva às relações de reciprocidade.

Precisamos dos valores amostrados $f(x_n)$ para determinar as aproximações para $f(\omega)$ de forma única, e vice-versa. Uma vez que N valores de $f(x_n)$ são usados na aproximação da regra do trapézio.

Relações de reciprocidade

$$A\Omega = N$$

Onde $\frac{\Omega}{2}$ é denominado de frequência de Nyquist.

$$\Delta x \Delta \omega = \frac{1}{N}$$

As duas relações de reciprocidade não são independentes, mas ambas são úteis. A primeira relação nos informa que, se o número de pontos da grade N for mantido fixo, um aumento no comprimento do domínio espacial ocorre às custas de uma diminuição no comprimento do domínio da frequência. Se A for aumentado, isso significa que períodos mais longos são permitidos na grade espacial, o que implica que a frequência fundamental $\Delta\omega$ diminui, levando a uma redução no comprimento do domínio da frequência $\Omega = N\Delta\Omega$. A segunda relação de reciprocidade pode ser interpretada de maneira semelhante. Reduzir ΔX pela metade com N fixo também reduz pela metade o comprimento do domínio espacial. O modo fundamental na grade original tem uma frequência de $\frac{1}{A}$ ciclos por unidade de comprimento, enquanto na nova grade mais curta tem uma frequência de $\frac{1}{(\frac{A}{2})}$ ou $\frac{A}{2}$ ciclos por unidade de comprimento. Assim, $\Delta\omega$ é duplicado no processo.

Usamos f_n para denotar os valores amostrados $f(x_n)$ para $n = -\frac{N}{2} + 1 : \frac{N}{2}$. Aproximando $\hat{f}$ nos pontos da grade de frequência $\omega_k = k\Delta\omega = \frac{k}{A}$ notamos que

$$x_n \omega_k = (n\Delta x)(k\Delta\omega) = \frac{nA}{N} \frac{k}{A} = \frac{nk}{N}.$$

Assim a soma da regra do trapézio torna-se

$$\hat{f}(\omega_k) \approx \frac{A}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f(x_n) e^{-i2\pi\omega_k x_n}$$

$$= \frac{A}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f_n e^{\frac{-i2\pi nk}{N}},$$

Portanto as aproximações para a Transformada de Fourier nos pontos de grade são dados por:

$$\hat{f}(\omega_k) = \hat{f}\left(\frac{k}{A}\right) = \int_{-\frac{A}{2}}^{\frac{A}{2}} f(x) e^{\frac{-i2\pi kx}{A}} dx \approx A \frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f_n e^{\frac{-i2\pi nk}{N}}.$$

A expressão acima representa a definição da DFT. Dado o conjunto de N valores amostrais f_n , a DFT consiste nos N coeficientes.

$$F_k = \frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f_n e^{\frac{-i2\pi nk}{N}}.$$

Podemos concluir que as aproximações para a transformada de Fourier são dadas por $\hat{f}(\omega_k) = AF_k$.

3.3 FORMA MATRICIAL DA DFT

Vimos anteriormente que a DFT da sequência (f_n) pode ser expressa da seguinte maneira:

$$F_k = \frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f_n e^{\frac{-i2\pi nk}{N}}, \quad k = -\frac{N}{2} + 1, \dots, \frac{N}{2}$$

Considerando a raiz da unidade:

$$\omega_N^{\frac{-i2\pi}{N}} = e^{\frac{-i2\pi}{N}}$$

Assim podemos reescrever a DFT da seguinte maneira:

$$\begin{aligned} F_k &= \frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f_n \omega_N^{-nk} \\ &= \frac{1}{N} \left[\sum_{n=0}^{\frac{N}{2}} f_n \omega_N^{-nk} + \sum_{n=-\frac{N}{2}+1}^{-1} f_n \omega_N^{-nk} \right]. \end{aligned} \tag{3.4}$$

Considerando a extensão periódica de período N e se $n = j - N$ ou equivalentemente

$j = n + N$ tem se que:

$$\sum_{n=-\frac{N}{2}+1}^{-1} f_n \omega_N^{-nk} = \sum_{j=-\frac{N}{2}+1}^{N-1} f_{j-N} \omega_N^{-(j-N)k}.$$

o fator $\omega_N^{-(j-N)k} = \omega_N^{-jk} \omega_N^{Nk}$, sendo $\omega_N^{Nk} = 1$ assim,

$$\begin{aligned} \sum_{n=-\frac{N}{2}+1}^{-1} f_n \omega_N^{-nk} &= \sum_{j=-\frac{N}{2}+1}^{N-1} f_{j-N} \omega_N^{-jk} \quad \text{por periodicidade} \\ &= \sum_{j=-\frac{N}{2}+1}^{N-1} f_j \omega_N^{-jk} \end{aligned}$$

Voltando na equação (3.4) obtemos:

$$F_k = \frac{1}{N} \left[\sum_{n=0}^{\frac{N}{2}} f_n \omega_N^{-nk} + \sum_{n=-\frac{N}{2}+1}^{N-1} f_n \omega_N^{-nk} \right] = \sum_{n=0}^{N-1} f_n \omega_N^{-nk}.$$

Usaremos esta forma alternativa de representar da DFT para introduzir a formulação matricial da transformada, isto é, podemos representa-la como o produto de uma matriz $N \times N$ e um vetor coluna de tamanho N .

Se f representa o vetor de tamanho N :

$$f = (f_0, f_1, f_2, \dots, f_{N-1})^T$$

e F representa o vetor de saída

$$F = (F_0, F_1, F_2, \dots, F_{N-1})^T,$$

podemos escrever a DFT da seguinte maneira:

$$F = W_N f,$$

ou seja

$$\begin{bmatrix} F_0 \\ F_1 \\ \vdots \\ F_{N-1} \end{bmatrix} = \frac{1}{N} \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & \omega_N^{-1} & \dots & \omega_N^{-(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N^{-(N-1)} & \dots & \omega_N^{-(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_{N-1} \end{bmatrix}.$$

Essa matriz tem algumas propriedades importantes, que veremos a seguir:

1. A matriz W_N é uma matriz quadrada não singular.

2. O inverso da matriz é um múltiplo da matriz adjunta.

$$W_N^{-1} = NW_N^*$$

Usando a inversa da matriz W_N , podemos obter a forma matricial da IDFT.

$$F = W_N f$$

$$f = W_N^{-1} F = NW_N^* F = W_N^{-1}.$$

3. Temos que a matriz é simétrica,

$$W_N^T = W_N$$

Para $N > 4$ a matriz terá 4 autovalores distintos:

$$\lambda_1 = \sqrt{N}, \lambda_2 = -\sqrt{N}, \lambda_3 = 0 - i\sqrt{N}, \lambda_4 = 0 + i\sqrt{N}$$

As multiplicidade dos autovalores estão relacionados a ordem N da matriz.

Agora apresentaremos um exemplo de obter a DFT usando a forma matricial para $N = 4$ com $f = (0,1,2,3)^t$

As entradas da matriz DFT F de tamanho $n \times n$ é definida como:

$$F_{jk} = \frac{1}{N} e^{-2\pi i(j-1)(k-1)/n}$$

onde j, k variam de 1 até n . Para $N = 4$, temos:

$$F = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{bmatrix}$$

E a sequência f é:

$$f = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \end{bmatrix}.$$

Para calcular a DFT de f , multiplicamos a matriz F pela sequência f :

$$Ff = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & i \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \end{bmatrix}.$$

Multiplicando essas matrizes, obtemos a seguinte DFT:

$$Ff = \frac{1}{4} \begin{bmatrix} 6 \\ -2 + 2i \\ -2 \\ -2 - 2i \end{bmatrix}.$$

Custo operacional

Para calcularmos cada elemento de F realizaremos N multiplicações complexas de cada elemento de W por cada elemento de f , assim teremos um total de $N \times N$ multiplicações complexas, ou seja, N^2 operações, tendo um custo operacional de $O(N^2)$.

Embora a DFT como produto matricial seja útil para compreensão teórica e implementação em pequenos conjuntos de dados, pelo seu custo operacional de $O(N^2)$ torna-se menos prática para aplicações em grandes conjuntos de dados. Para esses casos é mais vantajoso utilizar o algoritmo FFT que tem custo operacional de $O(N \log N)$, sendo mais eficiente para o cálculo da DFT do que utilizando o produto matricial.

Transformada Rápida de Fourier (FFT)

A Transformada Rápida de Fourier é um algoritmo muito eficiente que calcula a transformada Discreta De Fourier (DFT) e sua inversa. A análise de Fourier converte um sinal do domínio original para uma representação no domínio da frequência e vice-versa. Uma Transformada Rápida de Fourier calcula rapidamente essas transformações fatorizando a matriz da Transformada discreta de Fourier em um produto de fatores esparsos (principalmente zero). Como resultado, ele consegue reduzir a complexidade de calcular a Transformada discreta de Fourier de $O(N^2)$ operações para $O(N \log N)$ operações, onde N é o tamanho dos dados.

Para a aplicação da Transformada Rápida de Fourier (FFT), o tamanho da amostra N deve ser uma potência de 2. Essa exigência simplifica a implementação do algoritmo e otimiza seu desempenho. Portanto, o valor de N é escolhido como um número par. A DFT, é útil porque permite converter um problema complexo no domínio espacial (ou temporal) em um problema mais simples em outro domínio. Posteriormente, a solução obtida nesse segundo domínio precisa ser revertida para o domínio original. Para essa finalidade, é requerida uma transformação inversa. Mais detalhes do algoritmo FFT podem ser encontrados no capítulo 10 do livro de [9].

3.4 A DFT COMO APROXIMAÇÃO TRIGONOMÉTRICA

Uma abordagem alternativa para derivar a Transformada Discreta de Fourier (DFT) surge da tentativa de ajustar um conjunto de dados com uma função conhecida como polinômio trigonométrico. O objetivo é encontrar uma combinação linear de senos e

cossenos que forneça a melhor aproximação para os dados. Esta abordagem estabelece uma conexão elegante entre o problema de ajuste de dados e a DFT.

Matematicamente, essa abordagem envolve a minimização da diferença entre os dados originais e a função ajustada, expressa como um polinômio trigonométrico. A solução para este problema leva à descoberta da DFT.

Suponha que tenhamos N pares de dados denotados como (x_n, f_n) onde n varia de $n = \frac{-(N-1)}{2} : \frac{(N-1)}{2}$. Essa escolha de N como ímpar é conveniente para simplificar o tratamento, embora os princípios também se apliquem para N par com ajustes mínimos. Os x_n são números reais e são considerados pontos igualmente espaçados em um intervalo $[-\frac{A}{2}, \frac{A}{2}]$, ou seja, $x_n = n\Delta x$, onde $\Delta x = \frac{A}{N}$. Os f_n podem assumir valores complexos. Esses pares de dados podem ser interpretados como amostras de uma função contínua f que foi amostrada nos pontos da grade x_n .

No entanto, também é igualmente válido considerar que os pares são oriundos de um conjunto discreto de dados coletados.

Estamos buscando a melhor aproximação para os dados utilizando o polinômio trigonométrico na forma:

$$\psi_N(x) = \sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} \alpha_k e^{\frac{i2\pi kx}{A}}$$

Podemos impor várias condições para determinar a melhor aproximação para um conjunto de dados. Utilizaremos o critério dos mínimos quadrados e exigiremos que a soma dos quadrados das diferenças entre os valores dos dados e a função de aproximação ψ nos pontos x_n seja minimizada. Em outras palavras, procuramos escolher os coeficientes α_k para minimizar o erro discreto ao quadrado.

$$E(\alpha_1, \dots, \alpha_N) = E = \sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} |f_n - \psi_N(x_n)|^2$$

Uma condição necessária para a minimização de E é que as primeiras derivadas em relação a cada um dos N coeficientes sejam zero, isto é $\frac{\partial}{\partial \alpha_k} E = 0$, $k = 0, 1, N$. Observe que

$$|f_n - \psi_N(x_n)|^2 = (f_n - \psi_N(x_n))(f_n - \psi_N(x_n))^*,$$

logo

$$\frac{\partial E}{\partial \alpha_k} = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \left[e^{-\frac{i2\pi kn}{N}} \left(f_n - \sum_{p=-\frac{N-1}{2}}^{\frac{N-1}{2}} \alpha_p e^{\frac{i2\pi pn}{N}} \right) \right] = 0.$$

Observe que

$$\sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} f_n e^{-\frac{i2\pi kn}{N}} = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \left(\sum_{p=-\frac{N-1}{2}}^{\frac{N-1}{2}} \alpha_p e^{\frac{i2\pi pn}{N}} \right) e^{-\frac{i2\pi kn}{N}}.$$

Essas expressões podem ser ainda mais simplificadas utilizando a notação $\omega_N = e^{\frac{i2\pi}{N}}$:

$$\sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} f_n \omega_N^{-nk} = \sum_{p=-\frac{N-1}{2}}^{\frac{N-1}{2}} \alpha_p \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \omega_N^{(p-k)n}.$$

Usando a relação de ortogonalidade $\sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \omega_N^{(p-k)n} = N\delta(p-k)$ temos

$$\sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} f_n \omega_N^{-nk} = N\alpha_k, \quad k = -\frac{N}{2} + 1 : \frac{N}{2}.$$

A partir dessa última expressão, torna-se evidente que o erro dos mínimos quadrados é minimizado quando os coeficientes no polinômio aproximado são dados pela Transformada Discreta de Fourier (DFT) dos dados:

$$\alpha_k = \frac{1}{N} \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} f_n \omega_N^{-nk}.$$

Inicialmente, demonstramos que a melhor aproximação para os dados, expressa como polinômio trigonométrico, é dada pela Transformada Discreta de Fourier (DFT). Em seguida, analisaremos o erro associado a essa aproximação.

$$\begin{aligned} E &= \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} |f_n - \psi_N(x_n)|^2 \\ &= \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} (f_n - \psi_N(x_n))(f_n - \psi_N(x_n))^* \\ &= \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} |f_n|^2 - \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} f_n \psi_N(x_n)^* \\ &\quad - \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} f_n^* \psi_N(x_n) + \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} |\psi_N(x_n)|^2. \end{aligned}$$

Usando ortogonalidade e o fato de $\alpha_k = D(f)_k$.

$$\begin{aligned} \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} |\psi_N(x_n)|^2 &= \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} f(x_n) \psi_N(x_n)^* \\ &= \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} f(x_n)^* \psi_N(x_n) \end{aligned}$$

$$= N \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} |\alpha_n|^2.$$

Portanto,

$$E = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} |f_n|^2 - N \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} |\alpha_n|^2.$$

Observando $|\alpha_n|^2 = \alpha_n \alpha_n^*$ e usando α_k obtido anteriormente temos:

$$\begin{aligned} E &= \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} |f_n|^2 - \frac{N}{N^2} \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \sum_{p=-\frac{N-1}{2}}^{\frac{N-1}{2}} \sum_{m=-\frac{N-1}{2}}^{\frac{N-1}{2}} (f_p f_m^* e^{\frac{i2\pi n(m-p)}{N}}) \\ &= \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} |f_n|^2 - \frac{1}{N} \sum_{p=-\frac{N-1}{2}}^{\frac{N-1}{2}} \sum_{m=-\frac{N-1}{2}}^{\frac{N-1}{2}} f_p f_m^* \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} e^{\frac{i2\pi n(m-p)}{N}}. \end{aligned}$$

Dado que a última soma envolve N termos, podemos aplicar o conceito de ortogonalidade para concluir que é nula para todos os valores de p distintos de m . Quando p é igual a m , a soma assume o valor de N . Portanto, podemos observar que

$$\begin{aligned} E &= \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} |f_n|^2 - \frac{1}{N} \sum_{p=-\frac{N-1}{2}}^{\frac{N-1}{2}} \sum_{m=-\frac{N-1}{2}}^{\frac{N-1}{2}} f_p f_m^* N \delta_N(m-p) \\ &= \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} |f_n|^2 - \sum_{m=-\frac{N-1}{2}}^{\frac{N-1}{2}} |f_m|^2 = 0. \end{aligned}$$

Dado que a soma dos quadrados dos erros individuais nos pontos da grade é zero, decorre que os próprios erros individuais devem ser nulos. Em outras palavras, a função de aproximação ψ_N deve passar exatamente por cada ponto de dados, representado por $\psi_N(x_n) = f_n$ em todos os pontos da grade. Isso implica que ψ_N representa a aproximação por mínimos quadrados e atua como uma função de interpolação para os dados. Portanto, concluímos que a técnica de mínimos quadrados produz uma função interpolante. Na verdade, a Transformada Discreta de Fourier (DFT) pode ser derivada diretamente exigindo que o polinômio ψ_N interpole os dados.

3.5 PROPRIEDADES DA TRANSFORMADA DISCRETA DE FOURIER (DFT)

No que segue consideraremos f_n uma sequência finita de comprimento N par. A DFT de comprimento N e sua inversa foram definida no capítulo 3 como:

$$D\{F_n\}_k = F_k = \frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f_n \omega_N^{-nk}$$

e sua inversa como

$$D^{-1}\{F_k\}_n = f_n = \frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} F_k \omega_N^{nk}.$$

3.5.1 PERIODICIDADE:

A propriedade de periodicidade surge naturalmente usando a sequência complexa de f_n e F_k , assim temos que:

$$f_{n+N} = f_n \quad \text{e} \quad F_{k+N} = F_k \quad \text{para } n \text{ e } k \text{ inteiros.}$$

Essa propriedade decorre imediatamente do fato de que:

$$\omega_N^{-n(k+N)} = \omega_N^{-nN} \quad \text{e} \quad \omega_N^{k(n+N)} = \omega_N^{nk}.$$

3.5.2 LINEARIDADE:

Se F_n e G_n são sequências de valores complexos, e α e β números complexos temos que:

$$D\{\alpha F_n + \beta G_n\}_k = \alpha D\{F_n\}_k + \beta D\{G_n\}_k$$

Demonstração.

$$D\{\alpha F_n + \beta G_n\}_k = \alpha D\{F_n\}_k + \beta D\{G_n\}_k$$

$$\begin{aligned} \text{Pois : } D\{\alpha F_n + \beta G_n\}_k &= \frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} (\alpha F_n + \beta G_n) \omega_N^{-nk} \\ &= \alpha \frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} F_n \omega_N^{-nk} + \beta \frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} G_n \omega_N^{-nk} \\ &= \alpha D\{F_n\}_k + \beta D\{G_n\}_k. \end{aligned}$$

■

3.5.3 DESLOCAMENTO DA DFT:

Seja f_{n-j} uma sequência que foi deslocada j unidades para a direita. Utilizando a Inversa da Transformada Discreta de Fourier IDFT podemos expressar f_{n-j} como:

$$f_{n-j} = \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} F_k \omega_N^{(n-j)k}.$$

Demonstração.

$$\begin{aligned}
 f_{n-j} &= \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} F_k \omega_N^{(n-j)k} \\
 &= \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} \left[F_k \omega_N^{-jk} \right] \omega_N^{nk} \\
 &= D^{-1} \{ F_k \omega_N^{-jk} \}_n \\
 D \{ f_{n-j} \}_k &= F_k \omega_N^{-jk}
 \end{aligned}$$

■

3.5.4 MODULAÇÃO DA DFT:

A propriedade resulta na modulação da sequência de entrada, ou seja, na multiplicação dos elementos da sequência de entrada f_n por ω_N^{nj} onde j é um número inteiro fixo. De maneira simétrica e previsível, isso resulta em uma sequência DFT que é deslocada em relação à DFT da sequência não modulada F_N . Um argumento breve pode demonstrar o efeito da modulação:

$$\begin{aligned}
 D(f_n \omega_N^{nj})_k &= \frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} (f_n \omega_N^{nj}) \omega_N^{-nk} \\
 D\{f_n \omega_N^{nj}\}_k &= \frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f_n \omega_N^{-n(k-j)} \\
 D\{f_n \omega_N^{nj}\}_k &= D\{f_n\}_{k-j} = F_{k-j}.
 \end{aligned}$$

3.5.5 SIMETRIA HERMITIANA:

A relação entre a Transformada Discreta de Fourier (DFT) e a sua inversa, a IDFT, é notavelmente próxima, como evidenciado em suas definições. Na verdade, as transformações são tão similares que é possível calcular uma a partir da outra, com uma simples modificação nos dados de entrada.

Por exemplo, a inversa da transformada $f_n = D^{-1}(F_k)_n$ pode ser computada tomando o conjugado da Transformada Discreta de Fourier de uma versão ligeiramente modificada da sequência F_k . Da mesma forma podemos calcular $D(f_n)_k$.

A propriedade que captura a semelhança entre a DFT e IDFT é chamada de Simetria Hermitiana e se expressa matematicamente como:

$$f_n = D\{NF_k^*\}^* \text{ e } F_k = D^{-1}\left\{\frac{1}{N}f_n^*\right\}^*.$$

Essa expressão resulta diretamente da propriedade de simetria Hermitiana da matriz da DFT.

Essa simetria é fundamental para entender a relação entre as transformadas e é amplamente explorada em diversas aplicações de processamento de sinais.

3.5.6 DFT DE UMA SEQUÊNCIA REAL:

Suponha que tem uma sequência f_n real tal que $f_n^* = f_n$.

A Transformada Discreta de Fourier (DFT) da sequência f_n é dada por:

$$F_k = \left(\frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f_n \omega_N^{-nk} \right), \quad \text{onde } \omega_N \text{ é a raiz } N\text{-ésima da unidade.}$$

Tomando o conjugado da DFT, obtemos:

$$F_k^* = \left(\frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f_n \omega_N^{-nk} \right)^* = \frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f_n^* (\omega_N^{-nk})^* = \frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f_n \omega_N^{-n(-k)} = F_k$$

3.5.7 SIMETRIA CONJUGADA NA DFT:

Toda sequência f_n que satisfaz $f_n^* = f_{-n}$ é dita conjugada simétrica. Como observado na propriedade anterior, a DFT de uma sequência simétrica conjugada é real. Portanto, podemos estabelecer a seguinte relação:

$$(f_n)^* = f_{-n} \implies F_k^* = F_k.$$

Demonstração:

$A \implies B$

$$\begin{aligned} F_k^* &= \left(\frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f_n \omega_N^{-nk} \right)^* = \frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} (f_n)^* (\omega_N^{-nk})^* = \frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f_{-n} \omega_N^{nk} = \\ &= \frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f_{-n} \omega_N^{n(-k)} = \frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f_{-n} \omega_N^{-nk} = F_k. \end{aligned}$$

3.5.8 CONVOLUÇÃO DISCRETA (CÍCLICA)

O teorema da convolução discreta é uma das propriedades mais importantes da Transformada Discreta de Fourier (DFT), muito utilizada no processamento de sinais. Vamos começar definindo a noção de convolução discreta.

Sejam duas sequências periódicas f_n e g_n de comprimento N , sua convolução discreta será outra sequência h_n , definida como:

$$h_n = f_n * g_n = \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} F_n g_{n-j}. \quad (3.5)$$

Cada um dos termos da sequência h_n é o produto escalar entre a sequência f_n e outra sequência que é uma versão deslocada de g_n .

Teorema Da Convolução

Suponha que f_n e g_n sejam duas sequências periódicas de tamanho N , e suas respectivas Transformadas Discretas de Fourier (DFTs) sejam F_k e G_K , então a DFT da convolução de f_n e g_n é dada por:

$$H_k = D\{f_n * g_n\}_k = NF_k G_k$$

Demonstração:

$$\begin{aligned} f_n * g_n &= \sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} F_n g_{n-j} \\ &= \sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} \left[\left(\sum_{p=-\frac{N}{2}+1}^{\frac{N}{2}} F_p \omega_N^{jp} \right) \left(\sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} [G_k \omega_N^{-jk}] \omega_N^{nk} \right) \right] \\ &= \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} G_k \omega_N^{nk} \sum_{p=-\frac{N}{2}+1}^{\frac{N}{2}} F_p \left(\sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} \omega_N^{j(p-k)} \right). \end{aligned}$$

Usando a propriedade de ortogonalidade

$$\left(\sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} \omega_N^{j(p-k)} \right) = N \delta_N(p-k),$$

obtemos que

$$f_n * g_n = N \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} F_k G_k \omega_N^{nk}$$

Mostra-se que $f_n * g_n$ é IDFT de $NF_k G_k$. Em outras palavras, podemos expressar esta relação da seguinte forma

$$f_n * g_n = ND^{-1}\{F_k G_k\}_n$$

ou

$$D\{f_n * g_n\} = NF_k G_k.$$

■

Convolução de frequência

Da mesma forma que a Transformada Discreta de Fourier (DFT) de uma convolução entre duas sequências é o produto das suas respectivas DFTs, pode-se demonstrar que a DFT do produto de duas sequências é equivalente à convolução das DFTs dessas sequências.

$$D\{f_n g_n\} = F_k * G_k.$$

Considerando a transformada inversa da convolução conseguimos mostrar que

$$\begin{aligned} D^{-1}\{F_k * G_k\}_n &= \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} \left(\sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} F_j G_{k-j} \right) \omega_N^{nk} \\ &= \sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} F_j \left(\sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} G_{k-j} \omega_N^{nk} \right). \end{aligned}$$

Usando a propriedade do Deslocamento da DFT

$$\left(\sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} G_{k-j} \omega_N^{nk} \right) = g_n \omega_N^{-jn}.$$

logo

$$\begin{aligned} D^{-1}\{F_k * G_k\}_n &= \sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} F_j g_n \omega_N^{nj} \\ &= g_n \sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} F_j \omega_N^{nj} \\ &= g_n f_n. \end{aligned}$$

3.5.9 RELAÇÃO DE PARSEVAL

A relação de Parseval, também conhecida como Teorema de Energia de Parseval, estabelece a equivalência de energia entre uma sequência f_n e sua Transformada Discreta de Fourier F_k . Se f_n é estendida periodicamente, equivale a:

$$\frac{1}{N} \sum_{n=0}^{N-1} |f_n|^2 = \sum_{k=0}^{N-1} |F_k|^2.$$

Demonstração. Usando a forma matricial vista na seção 3.3 temos:

$$\begin{aligned} \sum_{n=0}^{N-1} |F_k|^2 &= \|W_N f_n\|_2^2 \\ &= (W_N f_n)^* (W_N f_n) \\ &= f_n^* W_N^* W_N f_n \\ &= f_n^* \frac{1}{N} I f_n = \frac{1}{N} f_n^* f_n \\ &= \frac{1}{N} |f_n|^2 = \frac{1}{N} \sum_{n=0}^{N-1} |f_n|^2. \end{aligned}$$

■

3.6 UMA APLICAÇÃO DA DFT NO PROCESSAMENTO DE SINAIS

Nesta seção apresentaremos um exemplo de aplicação da DFT para eliminar ruídos em sinais de 1D. A DFT é uma ferramenta matemática que transforma um sinal do domínio do tempo para o domínio da frequência, permitindo a análise das componentes de frequência presentes no sinal. Isso é útil para identificar e remover ruídos, que geralmente se apresentam como componentes de alta frequência com baixa amplitude no espectro de frequência.

Para ilustrarmos essa propriedade da DFT consideramos um sinal $f(t) = \sin(2k_1t) + \sin(2k_2t)$ com $k_1 = 250$ e $k_2 = 350$.

Sinais no mundo real frequentemente são contaminados por ruídos indesejados, o que pode dificultar a análise e processamento dos mesmos. No exemplo apresentado, perturbamos um sinal sinusoidal puro com ruído aleatório e aplicamos a DFT para remover esses ruídos.

DENSIDADE ESPECTRAL DE POTÊNCIA (PSD)

A densidade espectral de potência (PSD, do inglês Power Spectral Density) é uma função que descreve como a potência de um sinal (ou série temporal) está distribuída ao longo das diferentes frequências. Ela fornece uma medida de quanta energia ou potência está presente em cada componente de frequência de um sinal. Segue como PSD e definida:

$$PSD(f) = \frac{1}{N} ||(F_k)_k||^2$$

METODOLOGIA

- Utilizando o software Python, amostramos a função $f(t)$ no intervalo $[0,1]$ com $\Delta t = 10^{-3}$, que resulta em $N = 1000$ amostras;
- Realizamos a perturbação do sinal com ruído, adicionando um ruído aleatório de tamanho N ao sinal;
- Realizamos o cálculo e filtragem da energia espectral. Para isso calculamos a DFT do sinal perturbado e a densidade espectral de potência (PSD);
- Efetuamos a aplicação do filtro no espaço de Fourier, eliminando as componentes do espectro com $PSD < 100$;

- Em seguida foi realizada a recuperação do sinal original. Para transformar o sinal filtrado de volta ao domínio do tempo utilizamos a DFT inversa;
- Por fim realizamos a visualização dos resultados, graficamos o sinal original, sinal perturbado, a densidade espectral de potência e o sinal recuperado.

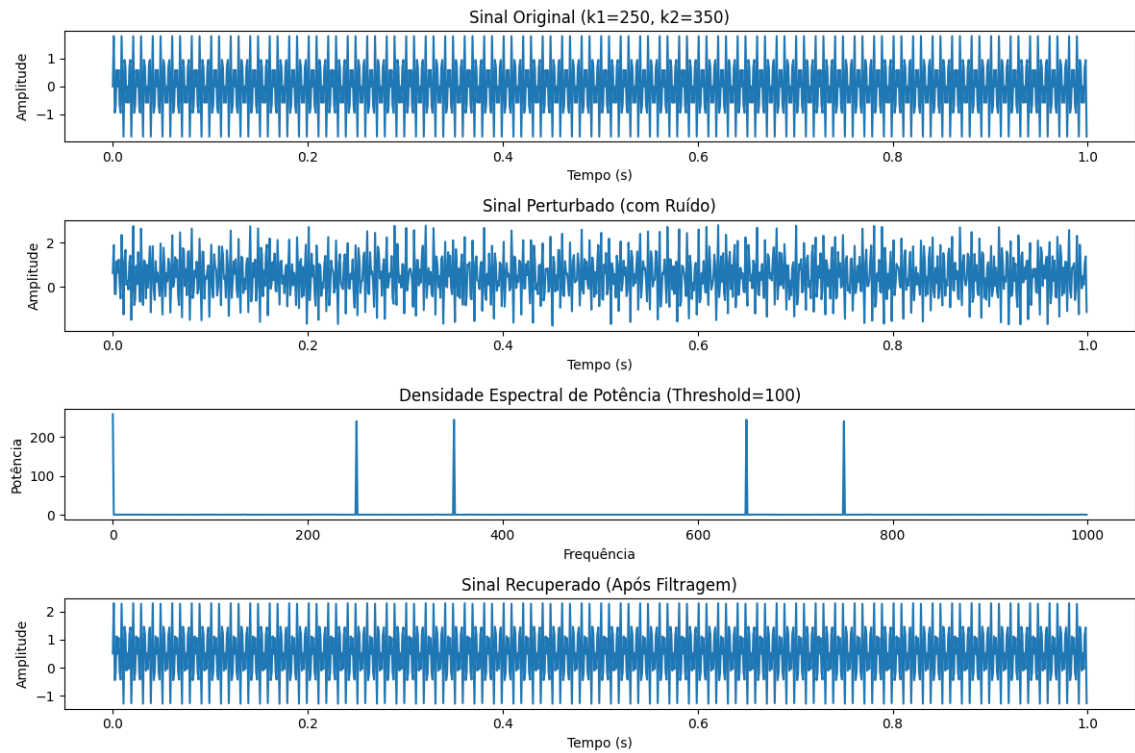


Figura 1 – Limpeza de Sinais Contaminados com Ruídos utilizando a DFT.

Resultados

Nos gráficos apresentados na figura (1) conseguimos visualizar a eficácia da DFT na remoção de ruídos. Utilizando a densidade espectral de potência que nos permite identificar e eliminar componentes de ruído que tem sinais contaminados, resultando em um sinal recuperado com uma melhor qualidade, que se aproxima bastante do sinal original.

4 TRANSFORMADA DISCRETA DO COS- SENO DCT

A Transformada Discreta do Cosseno (DCT) foi introduzida em 1974 por Nasir Ahmed, como uma versão discreta da Transformada Contínua do Cosseno, mas detalhes podem ser encontrados em [7]. A DCT é uma transformada real amplamente utilizada para compressão de imagens.

Dado um sinal discreto $f = (f_0, \dots, f_{N-1})$ define-se a DCT da seguinte forma:

$$C(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \cos \left(\frac{\pi}{N} \left(x + \frac{1}{2} \right) u \right), \quad u = 0, 1, \dots, N-1. \quad (4.1)$$

De maneira análoga a inversa é definida como:

$$f(x) = \sum_{u=0}^{N-1} \alpha(u) C(u) \cos \left(\frac{\pi}{N} \left(x + \frac{1}{2} \right) u \right), \quad x = 0, 1, \dots, N-1, \quad (4.2)$$

onde em ambas equações (4.1) e (4.2) α é definido como:

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}}, & \text{se } u = 0 \\ \sqrt{\frac{2}{N}}, & \text{se } u \neq 0 \end{cases} \quad (4.3)$$

Mostraremos na seção (4.3) que o conjunto dos N vetores $(V_u)_x = \cos \left(\frac{\pi}{N} \left(x + \frac{1}{2} \right) u \right)$ forma uma base ortogonal de $\mathbb{R}^N$, assim da expressão (4.2) pode-se interpretar a DCT como as componentes do sinal $f(x)$ nessa nova base.

Na figura (2) podem ser observados os gráficos dos vetores da base $\{\cos \left(\frac{\pi}{N} \left(x + \frac{1}{2} \right) u \right)\}_{u=0}^7$ para $N = 8$. Também pode-se observar na figura (2) que o vetor V_0 representa valor constante, e nos demais vetores de V_u fornecem formas de onda em frequências progressivamente crescentes. Essas formas de onda são denominadas funções base de cosseno. A DCT pode ser vista como uma DFT de aproximadamente o dobro do comprimento, operando em dados reais com simetria par. Isso ocorre porque a DFT de uma função real e par resulta em uma sequência real e par.

Existem 8 variações da DCT, a mais comum para ser utilizada em processamento de sinais e imagens é a DCT-II.

A DCT-II converte um sinal de domínio no tempo equivalente ao seu domínio da frequência. Uma característica importante é sua capacidade de concentrar a maior parte da energia do sinal nos primeiros coeficientes, tornando-a útil para a compressão dos dados. Nesse trabalho usaremos a DCT do tipo II.

Observações:

Se uma sequência de entrada tiver mais de N pontos de amostra, ela pode ser dividida em

subsequências de comprimento N , e a Transformada Discreta de Cosseno (DCT) pode ser aplicada a essas subsequências de forma independente.

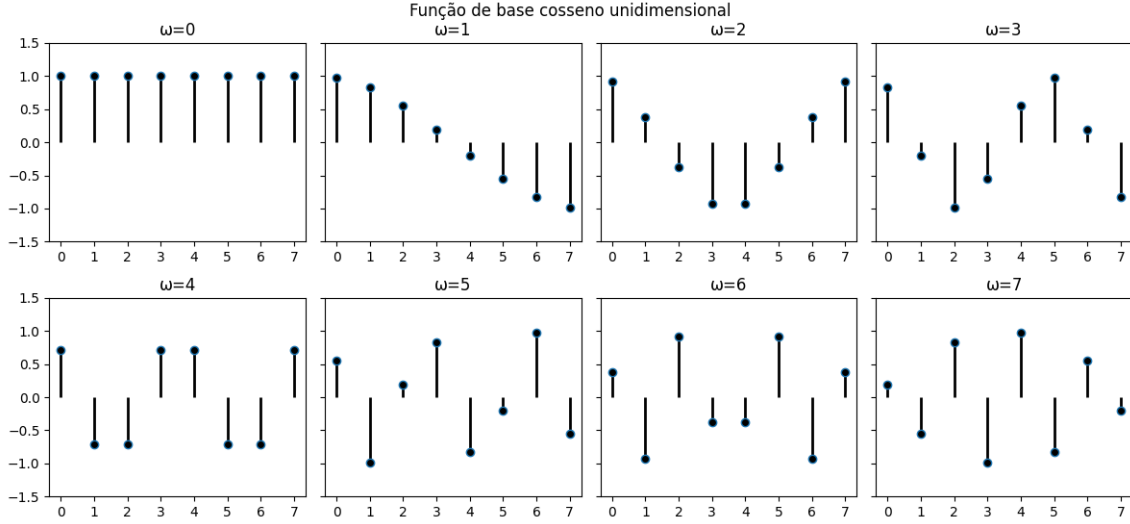


Figura 2 – Função de base cosseno unidimensional

4.1 MATRIZ DA DCT

A matriz da Transformada Discreta do Cosseno (DCT) é uma matriz que permite calcular a DCT de um vetor de amostras através da multiplicação matricial. Vamos definir explicitamente a matriz da DCT, que é a forma mais comumente usada. Seja $\mathbf{x}$ o vetor coluna de tamanho N contendo os elementos $f(x)$ e $\mathbf{X}$ o vetor coluna contendo os coeficientes $C(u)$. A relação pode ser escrita como:

$$\mathbf{X} = \mathbf{D}\mathbf{x}, \quad (4.3)$$

onde $\mathbf{D}$ é a matriz de transformada DCT de tamanho $N \times N$, onde cada elemento D_{ij} de $\mathbf{D}$ é dado pela expressão:

$$D_{ij} = \alpha(j) \cos\left(\frac{\pi}{N} \left(i + \frac{1}{2}\right) j\right),$$

onde os índices i e j variam de 0 a $N - 1$.

A matriz da DCT é dada por:

$$D = \alpha(j) \begin{bmatrix} \frac{1}{\sqrt{2}} & \cos\left(\frac{\pi}{N} \left(\frac{1}{2}\right)\right) & \cdots & \cdots & \cos\left(\frac{\pi}{N} \left(\frac{1}{2}\right) N - 1\right) \\ \frac{1}{\sqrt{2}} & \cos\left(\frac{\pi}{N} \left(\frac{3}{2}\right)\right) & \cdot & \cdots & \cos\left(\frac{\pi}{N} \left(\frac{3}{2}\right) N - 1\right) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\sqrt{2}} & \cos\left(\frac{\pi}{N} \left(\frac{2N-1}{2}\right)\right) & \cdots & \cdots & \cos\left(\frac{\pi}{N} \left(\frac{1}{2} + N - 1\right) N - 1\right) \end{bmatrix}$$

Custo Operacional

Quando realizamos o cálculo de $\mathbf{X} = \mathbf{D}\mathbf{x}$, são feitas N multiplicações de cada elemento de $\mathbf{x}$ por cada elemento da matriz $\mathbf{D}$, tendo um total de $N \times N$ multiplicações, ou seja, custo operacional de $O(N^2)$, assim é mais eficiente utilizar o algoritmo FFT que tem custo operacional $O(N \log N)$.

4.2 RELAÇÃO ENTRE DFT E DCT

Nesta seção se apresenta a relação existente entre a Transformada Discreta de Fourier (DFT) e a Transformada Discreta de Cosseno (DCT) de um sinal real $\{f_j\}_j^{N-1}$. Para isso, construímos a partir de $f = (f_0, f_1, \dots, f_{N-1})$ um novo vetor f_c de comprimento igual ao dobro de f e que consiste em uma reflexão do sinal em $N - \frac{1}{2}$, sendo N o tamanho de f ,

$$\text{especificamente, } f_c(x) = \begin{cases} f(x), & x = 0, \dots, N-1 \\ f(2N-1-x), & x = N, \dots, 2N-1. \end{cases}$$

A partir da equação (3.1) conseguimos escrever a versão unidimensional da DFT de $f_c(x)$ com comprimento $2N$.

$$DFT\{f_c(x)\}(k) = \sum_{x=0}^{2N-1} f_c(x) e^{\frac{-i2\pi xk}{2N}}. \quad (4.4)$$

Vamos dividir a soma global em uma soma para as duas partes do sinal composto.

$$DFT\{f_c(x)\}(k) = \sum_{x=0}^{N-1} f_c(x) e^{\frac{-i2\pi xk}{2N}} + \sum_{x=N}^{2N-1} f_c(x) e^{\frac{-i2\pi xk}{2N}}. \quad (4.5)$$

Podemos notar que $f_c(x) = f(x)$ quando $x \in \{0, 1, \dots, N-1\}$. Para a segunda soma uma mudança de variável é feita de x para x' , com $x' = 2N-1-x$ assim podemos rescrever-lá da seguinte maneira,

$$\begin{aligned} \sum_{x=N}^{2N-1} f_c(x) e^{\frac{-i2\pi xk}{2N}} &= \sum_{x'=N-1}^0 f_c(2N-1-x') e^{\frac{-i2\pi k(2N-1-x')}{2N}} \\ &= \sum_{x'=0}^{N-1} f_c(2N-1-x') e^{\frac{i2\pi k(x'+1)}{2N}}. \end{aligned} \quad (4.6)$$

Reescrevendo com um novo índice x a equação (4.6) temos:

$$\sum_{x=N}^{2N-1} f_c(x) e^{\frac{-i2\pi xk}{2N}} = \sum_{x=0}^{N-1} f(x) e^{\frac{i2\pi k(x+1)}{2N}}. \quad (4.7)$$

Substituindo (4.7) na equação (4.5) temos:

$$\begin{aligned} DFT\{f_c(x)\}(k) &= \sum_{x=0}^{N-1} f(x) e^{\frac{-i2\pi xk}{2N}} + \sum_{x=0}^{N-1} f(x) e^{\frac{i2\pi k(x+1)}{2N}} \\ &= \sum_{x=0}^{N-1} f(x) \left[e^{\frac{-i2\pi xk}{2N}} + e^{\frac{i2\pi k(x+1)}{2N}} \right]. \end{aligned} \quad (4.8)$$

assim, pode-se aplicar a propriedade

$$e^{i\lambda(a+1)} + e^{-i\lambda a} = e^{i\frac{\lambda}{2}} \left[e^{i\lambda(a+\frac{1}{2})} + e^{-i\lambda(a+\frac{1}{2})} \right] = 2e^{i\frac{\lambda}{2}} \cos \left(\lambda \left(a + \frac{1}{2} \right) \right),$$

com $a = x$ e $\lambda = \frac{2\pi k}{2N}$ em (4.8). Resulta que:

$$\text{DFT} \{f_c(x)\} (k) = e^{i\frac{\pi k}{2N}} \sum_{x=0}^{N-1} 2f(x) \cos \left[\frac{\pi}{2N} \left(x + \frac{1}{2} \right) k \right]. \quad (4.9)$$

Sendo que $\sum_{x=0}^{N-1} f(x) \cos \left[\frac{\pi}{N} \left(x + \frac{1}{2} \right) k \right]$ corresponde a DCT do sinal original f , assim temos que

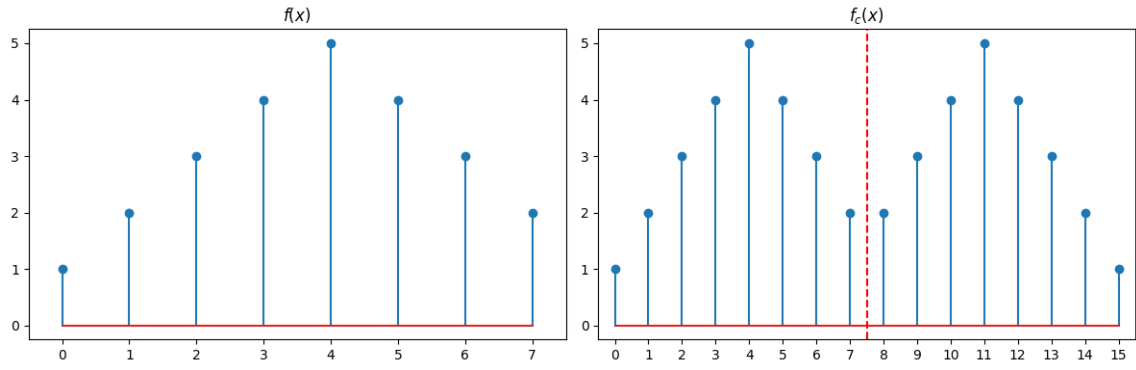
$$\text{DFT} \{f_c(x)\} (k) = e^{i\frac{\pi k}{N}} \text{DCT} \{f(x)\} \quad (4.10)$$

Pode-se observar que existe uma relação entre a Transformada Discreta de Fourier (DFT) e a Transformada Cosseno Discreta (DCT), onde um fator complexo de proporcionalidade aparece. No entanto, o módulo desse fator é unitário, o que significa que as propriedades de compactação da energia não são afetadas. Portanto, podemos destacar que:

Uma DCT é equivalente a uma DFT aplicada a uma sequência composta pelo sinal original e sua reflexão. Isso implica que a largura da janela pela qual um sinal de comprimento infinito teria sido multiplicado para obter o segundo seria o dobro do tempo que no caso do primeiro. Consequentemente, a dispersão produzida pela convolução com o espectro de tal janela seria menor do que aquela produzida no caso de uma janela com metade do comprimento.

Dada a relação entre a DFT e a DCT, podemos afirmar que o desenvolvimento periódico de $f_c(x)$ é uma expansão com transições menos abruptas do que a expansão periódica de $f(x)$. Isso ocorre porque $f_c(x)$ é composto pelo sinal original e $f(x)$ e sua reflexão, o que elimina possíveis descontinuidades nos extremos quando o sinal é periódico, como podemos ver na figura (3).

Conclui-se que a DCT concentra melhor a energia em baixas frequências do que a DFT. A DFT, ao se basear em uma extensão periódica do sinal, considera os pixels nas extremidades opostas da imagem como "vizinhos", o que provoca transições abruptas. Em contraste, a DCT, ao se basear em uma extensão simétrica do sinal, mantém a suavidade nas bordas. Essa característica da DCT resulta em uma melhor compactação da energia nas baixas frequências.

Figura 3 – Ilustração da criação de $f_c(x)$ a partir de $f(x)$

4.3 ORTOGONALIDADE DAS FUNÇÕES BASES

A matriz da DCT é dada pela inversa de D e como D é **ortogonal** resulta que $D^{-1} = D^T$ logo temos

$$\text{DCT}(f) = D^T(F_c)$$

$$F_c = \text{IDCT}(f) = D(f)$$

Agora precisamos mostrar que a matriz D é ortogonal, para a demonstração iremos mostrar que as colunas da matriz correspondem aos autovetores de uma matriz simétrica. Vamos considerar a matriz tridiagonal A_N real e simétrica de tamanho $N \times N$:

$$A_N = \begin{pmatrix} 1 & -1 & 0 & \cdots & \cdots & 0 \\ -1 & 2 & -1 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \\ \cdots & \cdots & \cdots & -1 & 2 & -1 \\ \cdots & \cdots & \cdots & \cdots & -1 & 1 \end{pmatrix}.$$

O nosso objetivo aqui é mostrar que as colunas de D são autovetores da matriz A_N , com V_k denotando a k -ésima coluna de D .

Seja $V_1 = \frac{1}{\sqrt{N}}(1, 1, \dots, 1)^t$, a primeira coluna de D

$$\Rightarrow A_N V_1 = 0 \Rightarrow V_1 \text{ é autovetor associado a } \Rightarrow \lambda_1 = 0$$

Seja, para cada $k = 1, 2, \dots, N - 2$, se $\alpha = \frac{k\pi}{N}$

$$\begin{aligned}
\vec{V}^{(k)} &= \sqrt{\frac{2}{N}} \left(\cos \left(\alpha \left(\frac{1}{2} + 1 \right) \right), \cos \left(\alpha \left(\frac{1}{2} + 2 \right) \right), \dots, \cos \left(\alpha \left(\frac{1}{2} + (N-1) \right) \right) \right) \\
&= \left(A\vec{V}^{(k)} \right)_j = \sum_{s=0}^{N-1} a_{js} \vec{V}_s^{(k)} \\
&= -V_{j-1}^{(k)} + 2V_j^{(k)} - V_{j+1}^{(k)} \\
&= \sqrt{\frac{2}{N}} \left(-\cos \left(\alpha \left(\frac{1}{2} + j - 1 \right) \right) + 2\cos \left(\alpha \left(\frac{1}{2} + j \right) \right) - \cos \left(\alpha \left(\frac{1}{2} + j + 1 \right) \right) \right) \\
&= \sqrt{\frac{2}{N}} \left(-\cos \left(\alpha \left(j - \frac{1}{2} \right) \right) + 2\cos \left(\alpha \left(j + \frac{1}{2} \right) \right) - \cos \left(\alpha \left(j + \frac{3}{2} \right) \right) \right) \\
&= \sqrt{\frac{2}{N}} \left(-\cos \left(\alpha \left(j + \frac{1}{2} \right) - \alpha \right) + 2\cos \left(\alpha \left(j + \frac{1}{2} \right) \right) - \cos \left(\alpha \left(j + \frac{3}{2} \right) \alpha \right) \right).
\end{aligned}$$

Considerando $\alpha \left(j + \frac{1}{2} \right) = \theta$, temos que:

$$\begin{aligned}
\vec{V}^{(k)} &= \sqrt{\frac{2}{N}} (-\cos(\theta - \alpha) + 2\cos \theta - \cos(\theta + \alpha)) \\
&= \sqrt{\frac{2}{N}} (-\cos \theta \cos \alpha - \sin \theta \sin \alpha + 2\cos \theta - \cos \theta \cos \alpha + \sin \theta \sin \alpha) \\
&= \sqrt{\frac{2}{N}} (-2\cos \theta \cos \alpha + 2\cos \theta) \\
&= \sqrt{\frac{2}{N}} (2\cos \theta (1 - \cos \alpha)) \\
&= \sqrt{\frac{2}{N}} (-2\cos \theta \cos \alpha + 2\cos \theta) \\
&= \sqrt{\frac{2}{N}} (2\cos \theta (1 - \cos \alpha)).
\end{aligned}$$

$$\Rightarrow \left(A\vec{V}^{(k)} \right)_j = 2 \left(1 - \cos \frac{k\pi}{N} \right) \sqrt{\frac{2}{N}} \cos \left(\frac{k\pi}{N} \left(j + \frac{1}{2} \right) \right).$$

sendo $\lambda_k = 2 \left(1 - \cos \frac{k\pi}{N} \right)$ e $V_j^{(k)} = \sqrt{\frac{2}{N}} \cos \left(\frac{k\pi}{N} \left(j + \frac{1}{2} \right) \right)$, para $j = 1, 2, \dots, N-1$.

Resta mostrar para $j = 0$ e $j = N-1$.

Para $j = 0$:

$$\begin{aligned}
 (AV^{(k)})_j &= V_0^{(k)} - V_1^{(k)} \\
 &= \sqrt{\frac{2}{N}} \left(\cos\left(\frac{1}{2}\alpha\right) - \cos\left(\frac{3}{2}\alpha\right) \right) \\
 &= \sqrt{\frac{2}{N}} \left(\cos\frac{1}{2}\alpha - \cos\frac{\alpha}{2} \cos\alpha + \sin\frac{\alpha}{2} \sin\alpha \right) \\
 &= \sqrt{\frac{2}{N}} \cos\frac{\alpha}{2} \left(1 - \cos\alpha + 2\sin^2\frac{\alpha}{2} \right), \quad +2\sin^2\frac{\alpha}{2} = 1 - \cos\alpha \\
 &= \sqrt{\frac{2}{N}} \cos\frac{\alpha}{2} (2(1 - \cos\alpha)) \\
 &= \lambda_k V_0^{(k)}
 \end{aligned}$$

Para $j = N - 1$

$$\begin{aligned}
 (AV^{(k)})_{j=N-1} &= V_{N-2}^{(k)} + V_{N-1}^{(k)} = V_{N-1}^{(k)} - V_{N-2}^{(k)} \\
 &= \sqrt{\frac{2}{N}} \cos\left(\frac{\pi k}{N} \left(N - 1 + \frac{1}{2}\right)\right) - \cos\left(\frac{\pi k}{N} \left(N - 2 + \frac{1}{2}\right)\right) \\
 &= 2 \left(1 - \cos\left(\frac{\pi k}{N}\right)\right) \cos\left(\frac{\pi k}{N} \left(N - 1 + \frac{1}{2}\right)\right)
 \end{aligned}$$

ou

$$\begin{aligned}
 \cos\left(\alpha \left(N - \frac{1}{2}\right)\right) - \cos\left(\alpha \left(N - \frac{3}{2}\right)\right) &= 2(1 - \cos\alpha) \cos\left(\alpha \left(N - \frac{1}{2}\right)\right), \quad \alpha = \frac{\pi k}{N} \\
 \Leftrightarrow \cos\left(\alpha \left(N - \frac{1}{2}\right)\right) - \cos\left(\alpha \left(N - \frac{3}{2}\right)\right) &= 2 \cos\left(\alpha \left(N - \frac{1}{2}\right)\right) - 2 \cos(\alpha) \cos\left(\alpha \left(N - \frac{1}{2}\right)\right) \\
 \Leftrightarrow \cos\left(\alpha \left(N - \frac{1}{2}\right)\right) + \cos\left(\alpha \left(N - \frac{3}{2}\right)\right) &= +2 \cos\alpha \cos\left(\alpha \left(N - \frac{1}{2}\right)\right) \\
 \Leftrightarrow \cos\left(\alpha \left(N - \frac{1}{2}\right)\right) + \cos\left(\alpha \left(N - \frac{3}{2}\right)\right) &= \cos\left(\alpha \left(N - \frac{1}{2}\right) + \alpha\right) + \cos\left(\alpha \left(N - \frac{1}{2}\right) - \alpha\right) \\
 \Leftrightarrow \cos\left(\alpha \left(N - \frac{1}{2}\right)\right) - \cos\left(\alpha \left(N - \frac{3}{2}\right)\right) &= \cos\left(\alpha \left(N + \frac{1}{2}\right)\right) + \cos\left(\alpha \left(N - \frac{3}{2}\right)\right) \\
 \Leftrightarrow \cos\left(\alpha \left(N - \frac{1}{2}\right)\right) &= \cos\left(\alpha \left(N + \frac{1}{2}\right)\right)
 \end{aligned}$$

Esta última relação é verdadeira pois:

$$\cos(\alpha N - \alpha/2) = \cos \alpha N \cos \alpha/2 + \sin \alpha N \sin \alpha/2 = \cos \alpha N \cos \alpha/2$$

$$\cos(\alpha N + \alpha/2) = \cos \alpha N \cos \alpha/2 - \sin \alpha N \sin \alpha/2 = \cos \alpha N \cos \alpha/2$$

pois $\sin(\alpha N) = \sin\left(\frac{\pi k}{N} N\right) = \sin k\pi = 0$

Mostramos assim que $\forall j = 0, 1, \dots, N-1 \quad \left(A\vec{V}^{(k)}\right)_j = \lambda_k \vec{V}_j^{(k)}$.

Logo, os V^k que denotam as colunas da matriz D são autovetores da matriz simétrica A_N . Logo, resultam ser ortogonais, como queríamos mostrar.

4.4 DCT BIDIMENSIONAL

Um dos objetivos desse trabalho é aplicar a DCT na compreensão de imagens digitais. Ao aplicar a DCT-2D a uma imagem, a informação espacial dos pixels é convertida em informação de frequência, resultando em uma matriz de coeficientes que refletem as diferentes frequências presentes na imagem. Para isso utilizaremos a extensão das ideias apresentadas na DCT unidimensional para um espaço bidimensional, assim a DCT bidimensional é uma extensão direta da DCT unidimensional e é dada por, para cada $u, v = 0, 1, \dots, N-1$:

$$C(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos\left(\frac{\pi}{N} \left(x + \frac{1}{2}\right) u\right) \cos\left(\frac{\pi}{N} \left(y + \frac{1}{2}\right) v\right), \quad (4.11)$$

A Transformada inversa é dada de forma análoga, para cada $x, y = 0, 1, \dots, N-1$

$$f(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \alpha(u)\alpha(v) C(u, v) \cos\left(\frac{\pi}{N} \left(x + \frac{1}{2}\right) u\right) \cos\left(\frac{\pi}{N} \left(y + \frac{1}{2}\right) v\right), \quad (4.12)$$

As funções base da DCT bidimensional podem ser geradas multiplicando as funções base da DCT unidimensional orientadas horizontalmente mostrada na figura 2 com o conjunto das mesmas funções orientadas verticalmente.

Pode-se notar que as funções base exibem um aumento progressivo de frequência tanto na direção vertical quanto na horizontal.

4.5 PROPRIEDADES DA DCT:

4.5.1 DECORRELAÇÃO:

A propriedade de decorrelação refere-se à capacidade da DCT de transformar dados correlacionados em dados que são, em grande parte, não correlacionados. Em outras palavras, quando um sinal que contém redundâncias (correlações) é transformado pela DCT, os coeficientes resultantes são aproximadamente independentes uns dos outros.

Em muitos sinais reais, como imagens e áudio, os valores consecutivos são frequentemente

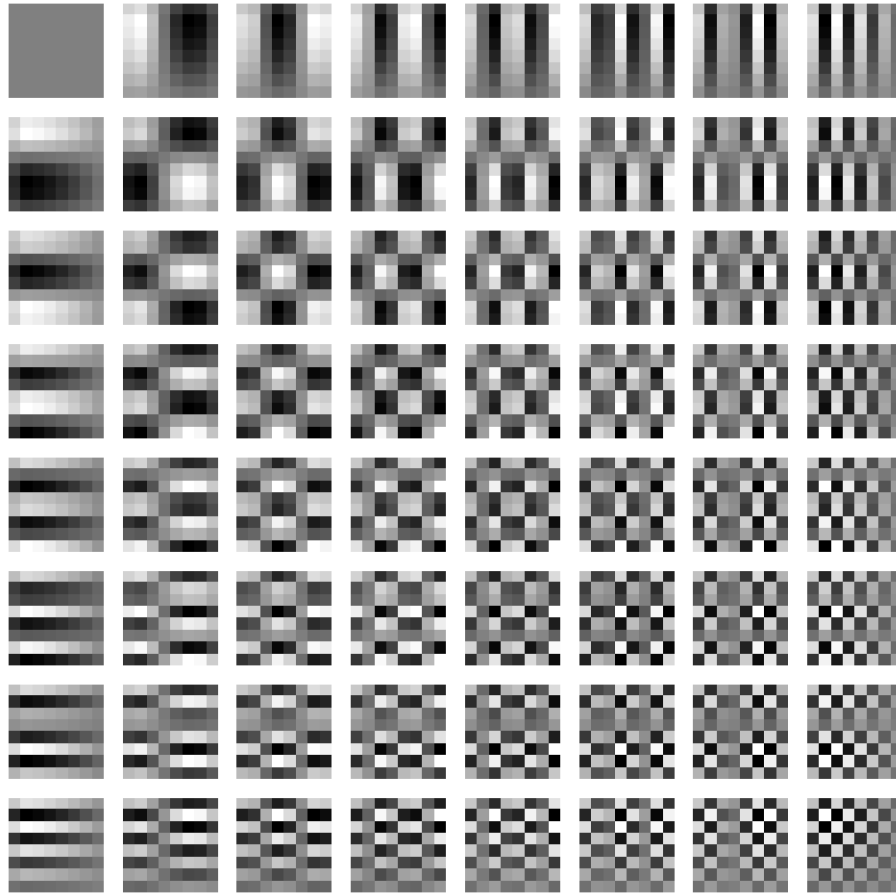


Figura 4 – Funções base da DCT bidimensional, para $N = 8$. O cinza representa $u = 0$ e $v = 0$, o branco representa amplitudes positivas e o preto representa amplitudes negativas.

correlacionados. A DCT ajuda a reduzir essa redundância, resultando em coeficientes que representam a energia do sinal de forma mais compacta. Devido à redução da correlação, muitos dos coeficientes da DCT podem ser zero ou muito pequenos, o que permite uma compressão mais eficiente. Apenas os coeficientes de maior magnitude são necessários para reconstruir o sinal original com alta qualidade.

Considere uma imagem onde os pixels vizinhos têm valores similares, indicando alta correlação espacial. Quando a DCT é aplicada a essa imagem, os coeficientes da transformada refletem a energia concentrada em componentes de baixa frequência, onde a correlação é alta, e os componentes de alta frequência tendem a ser pequenos ou zero, indicando menos correlação.

Para entender a decorrelação, precisamos analisar as propriedades estatísticas do sinal transformado. Vamos assumir que $\mathbf{x}$ é um vetor aleatório com uma matriz de covariância $\mathbf{R}_{\mathbf{x}}$, a matriz de covariância é dada por:

$$\mathbf{R}_{\mathbf{x}} = E[\mathbf{x}\mathbf{x}^T].$$

A matriz de covariância do vetor transformado $\mathbf{X}$ é:

$$\mathbf{R}_{\mathbf{X}} = E[\mathbf{X}\mathbf{X}^T].$$

Substituindo a equação (4.3), temos:

$$\mathbf{R}_{\mathbf{X}} = E[\mathbf{D}\mathbf{x}(\mathbf{D}\mathbf{x})^T] = E[\mathbf{D}\mathbf{x}\mathbf{x}^T\mathbf{D}^T].$$

A decorrelação ocorre porque a matriz de transformação da DCT é ortogonal, o que garante que os coeficientes resultantes não correlacionados.

4.5.2 COMPACTAÇÃO DE ENERGIA:

A propriedade de compactação de energia refere-se à capacidade da DCT de concentrar a maior parte da energia do sinal original nos primeiros e poucos coeficientes de $C(u)$ no domínio da frequência. Isso significa que, após a transformação, os coeficientes de baixa frequência tendem a conter a maior parte da informação útil, enquanto os coeficientes de alta frequência tendem a ser pequenos ou zero. Podemos visualizar essa compactação de energia no gráfico 5, que nos mostra a maior parte de informação útil nos coeficientes de baixa frequência.

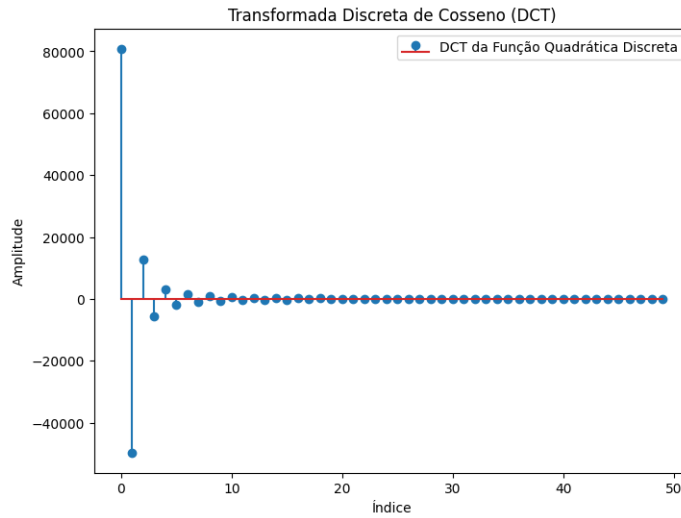


Figura 5 – DCT da Função Quadrática discreta $y(n) = n^2$

4.5.3 SEPARABILIDADE:

A principal vantagem da propriedade, conhecida como separabilidade, é que $C(u,v)$ pode ser calculado em duas etapas por operações unidimensionais sucessivas nas linhas e colunas de uma imagem.

A equação da DCT bidimensional (4.11) pode ser escrita como.

$$C(u,v) = \alpha(u)\alpha(v) \sum_{i=0}^{N-1} \cos\left(\frac{\pi}{N}\left(i + \frac{1}{2}\right)u\right) \sum_{y=0}^{N-1} f(i,y) \cos\left(\frac{\pi}{N}\left(y + \frac{1}{2}\right)v\right), \quad (4.13)$$

Onde $\alpha(u)$ e $\alpha(v)$ são definidos na equação (4.3) onde $u, v = 0, 1, \dots, N - 1$:

4.5.4 SIMETRIA:

Outra análise das operações de linha e coluna na equação (4.13) revela que essas operações são funcionalmente idênticas. Tal transformação é chamada de transformação simétrica. Uma transformação separável e simétrica pode ser expressa na forma.

$$T = \mathbf{D}f\mathbf{D}, \quad (4.14)$$

onde $\mathbf{D}$ é uma matriz de transformação simétrica $N \times N$ com entradas $d(i,j)$ de $\mathbf{D}$ dado por.

$$d(i,j) = \alpha(j) \sum_{i=0}^{N-1} \cos\left(\frac{\pi}{N}\left(j + \frac{1}{2}\right)i\right). \quad (4.15)$$

4.6 COMPRESSÃO DE IMAGENS USANDO A DCT

A compressão de imagens utilizando a Transformada Discreta do Cosseno (DCT) é uma técnica amplamente utilizada em sistemas de compressão de imagem, como o JPEG. A JPEG (Joint Photographic Experts Group) é um formato de compressão de imagens digitais. Este formato é amplamente utilizado para armazenar e transmitir fotografias digitais devido à sua capacidade de reduzir o tamanho do arquivo, mantendo uma qualidade de imagem aceitável. A DCT transforma a imagem do domínio espacial para o domínio da frequência, permitindo que os componentes de frequência menos importantes sejam descartados, resultando em uma compressão eficiente.

Uma imagem é representada por uma matriz, se a imagem é de resolução 256×256 , a matriz será de ordem 256. Nesse trabalho a imagem será convertida para a escala cinza, removendo assim os canais de cores RGB da imagem.

A conversão para escala de cinza na compressão de imagens é feita principalmente para simplificar o processo de compressão e reduzir a quantidade de dados. Cada pixel em uma imagem colorida contém informações de cor para cada canal (R, G e B), aumentando a quantidade total de dados. Ao converter para escala de cinza, essas informações são reduzidas a um único valor de intensidade, diminuindo a quantidade de informação a ser comprimida. A imagem é dividida em blocos de 8×8 pixels. Isso é feito porque a DCT é mais eficiente quando aplicada a blocos menores. Cada bloco de 8×8 é transformado

usando a DCT. A fórmula da DCT para um bloco 8×8 é, com $u, v = 0, 1, \dots, N - 1$:

$$C(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos\left(\frac{\pi}{8} \left(x + \frac{1}{2}\right) u\right) \cos\left(\frac{\pi}{8} \left(y + \frac{1}{2}\right) v\right). \quad (4.16)$$

Os coeficientes resultantes da DCT são quantizados. Isso significa que cada coeficiente da DCT é dividido pelo respectivo coeficiente da matriz de quantização e arredondados para o inteiro mais próximo. A matriz de quantização determina o nível de compressão e a perda de qualidade. Os coeficientes quantizados são então codificados usando técnicas como codificação de Huffman ou codificação RLE (Run-Length Encoding).

Para visualizar a imagem, o processo é revertido. Os dados são decodificados, desquantizados e a DCT inversa é aplicada para reconstruir a imagem.

Na compressão de imagens temos a compressão com perda e sem perda, a seguir apresentaremos informações sobre elas.

Compressão com perda

Quando fazermos a compressão da imagem usando a DCT iremos ter perdas de algumas informações, pois quando aplicamos a quantização os coeficientes de frequência são divididos e arredondados, e as informações originais são truncadas. Essa redução de precisão resulta na perda de detalhes finos na imagem, que estão apresentados nos coeficientes de alta frequência. Como esses detalhes são menos perceptíveis ao olho humano, eles são mais quantizados (ou até zerados) durante a compressão, resultando na perda de detalhes finos. Esse é o motivo pelo qual áreas da imagem com muitos detalhes podem parecer borradas após a compressão.

Em uma imagem comprimida, áreas suaves podem aparecer quase idênticas à imagem original, mas áreas com muitos detalhes podem apresentar borrões.

Após a descompressão, a imagem reconstruída não será exatamente igual à original devido à perda de detalhes durante a quantização.

Partes da informação que não são essenciais para a percepção visual são removidas, o que reduz a quantidade de dados necessária para representar a imagem.

Compressão sem perda

Compressão sem perda (ou compressão lossless) é uma técnica que reduz o tamanho dos arquivos sem eliminar nenhum dado da informação original. Isto significa que a imagem comprimida pode ser descomprimida de volta ao seu estado original, sem nenhuma perda de qualidade ou detalhes. A compressão sem perda é crucial em aplicações onde a integridade dos dados é fundamental. Alguns dos métodos usados são, Run-Length Encoding (RLE) e Huffman Coding. A vantagem da compressão sem perda é não ter perda de qualidade, a imagem descomprimida é idêntica à original. A desvantagem está na taxa de compressão que é menor em comparação com a compressão com perda.

A Matriz de quantização utilizada nesse trabalho é a seguinte:

$$Q_{50} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}.$$

A qualidade da compressão é melhor nos blocos da matriz onde os valores da matriz de quantização são menores. Esses valores de baixa frequência estão localizados no canto superior esquerdo da matriz de quantização. Estes valores correspondem aos detalhes mais significativos da imagem, como variações suaves de brilho e cores que o olho humano percebe facilmente, eles também resultam em uma quantização menos agressiva, preservando melhor os detalhes nesses blocos.

Os valores de alta frequência estão localizados no canto inferior direito da matriz de quantização. Estes valores correspondem aos detalhes mais finos e variações rápidas, que o olho humano é menos sensível a perceber. Valores maiores na matriz de quantização resultam em uma quantização mais agressiva, reduzindo mais esses coeficientes, o que pode resultar em perda de qualidade, mas é menos perceptível.

4.7 OUTRAS VARIANTES PARA A DCT

Seja $u = (u_0, u_1, \dots, u_{n-1})$ um vetor em $\mathbb{R}^n$ estendido periodicamente e considere as equações das segundas diferenças $y_j = -u_{j-1} + 2u_j - u_{j+1}$ que matricialmente pode ser escrito como $y = A_0 u$, onde A_0 é a matriz circulante

$$A_0 = \begin{pmatrix} 2 & -1 & \dots & \dots & -1 \\ -1 & 2 & -1 & & \\ \vdots & & \ddots & & \\ & & -1 & 2 & -1 \\ -1 & & & -2 & 2 \end{pmatrix}.$$

Essa escolha do sinal $-u''$ garante que cada matriz seja positiva definida (ou pelo menos semidefinida), assegurando que nenhum autovalor seja negativo.

Note que nas primeiras e últimas linhas ($j = 0$ e $j = N - 1$), essa segunda diferença envolve u_{-1} e u_N , ultrapassando os limites. Assim, usando a periodicidade temos que $u_N = u_0$ e $u_{N-1} = u_{-1}$ isto se reflete nas entradas -1 que aparecem nos cantos de A_0 .

Vamos verificar que $v_k = (1, w_k, w_{2k}, \dots, w_{(N-1)k})$ é um autovetor de A_0 . Este vetor é periódico porque $w^N = 1$, onde $w = e^{\frac{2\pi i}{N}}$. A j -ésimo componente de $A_0 v_k = \lambda_k v_k$ é a segunda diferença:

$$\begin{aligned} -w^{(j-1)k} + 2w^{jk} - w^{(j+1)k} &= (-w^{-k} + 2 - w^k)w^{jk} \\ &= (-e^{-2\pi i k/N} + 2 - e^{2\pi i k/N})w^{jk} \\ &= \left(2 - 2\cos\left(\frac{2k\pi}{N}\right)\right)w^{jk}. \end{aligned} \quad (4.17)$$

A matriz A_0 é simétrica, e seus autovalores $\lambda_k = 2 - 2\cos\left(\frac{2k\pi}{N}\right)$ são reais. O menor autovalor é $\lambda_0 = 0$, correspondente ao autovetor $v_0 = (1, 1, \dots, 1)$. É muito útil contar com esse vetor na base pois em processamento de imagens corresponde a um nível de cinza constante.

Como A_0 é uma matriz real simétrica, seus autovetores ortogonais também podem ser escolhidos como vetores reais. Na verdade, as partes real e imaginária de v_k devem ser autovetores,

$$\begin{aligned} c_k &= \Re v_k = \left(1, \cos\left(\frac{2k\pi}{N}\right), \cos\left(\frac{4k\pi}{N}\right), \dots, \cos\left(\frac{2(N-1)k\pi}{N}\right)\right), \\ s_k &= \Im v_k = \left(0, \sin\left(\frac{2k\pi}{N}\right), \sin\left(\frac{4k\pi}{N}\right), \dots, \sin\left(\frac{2(N-1)k\pi}{N}\right)\right). \end{aligned}$$

Os pares de autovalores iguais λ_k e $\lambda_{(N-k)}$ fornecem os dois autovetores c_k e s_k . As exceções são $k = 0$ com o autovetor $c_0 = (1, 1, \dots, 1)$, e, para N par, $k = \frac{N}{2}$ com $c_{\frac{N}{2}} = (1, -1, \dots, 1, -1)$.

Considerando apenas os cossenos ao longo de meio período, isto é, no intervalo $[0, \pi]$. A periodicidade é perdida porque $\cos(0) \neq \cos(\pi)$. A equação diferencial ainda é $-u'' = k^2 u$, a condição de contorno que fornece cossenos é $u'(0) = 0$, quando considerada a condição $u'(\pi) = 0$ obtém-se $v_k = \cos(kx)$. Assim restando duas possibilidades na outra fronteira: Neumann e Dirichlet.

Para o caso discreto, cada condição de contorno escrita de duas formas diferentes usando o valor de u sobre os pontos de malha ou sobre os pontos médios, assim temos 2 variantes de discretização para $u'(0) = 0$, para $u'(\pi) = 0$ e para $u(\pi) = 0$ resultando em 8 possibilidade. Essas oito combinações resultam em oito transformadas de cosseno diferentes.

Cada uma das oito matrizes tem a forma tridiagonal:

$$A = \begin{pmatrix} \otimes & \otimes & & & \\ -1 & 2 & -1 & & \\ \vdots & & \ddots & & \\ & & & -1 & 2 & -1 \\ - & & & \otimes & \otimes \end{pmatrix}.$$

As condições de contorno determinam os autovetores, com quatro possibilidades em cada extremidade: Dirichlet ou Neumann, centrado em um ponto de malha ou no ponto médio. Pode-se argumentar que essa simetria exige a presença da diagonal -1 na primeira e na última linha. No entanto, a condição de Neumann no ponto de malha gera -2 . Por isso, admitimos que os autovetores nesse caso necessitam de um reescalonamento final (envolvendo apenas $\sqrt{2}$) para serem ortogonais. O resultado é um conjunto de vetores base simples e elegante. Analisaremos a seguir as diferentes possibilidades das condições de contorno

Existem duas escolhas para $u'(0) = 0$:

- Simetria em torno do ponto de malha: $j = 0$: $u_{-1} = u_1$;
- Simetria em torno do ponto médio: $j = -\frac{1}{2}$: $u_{-1} = u_0$.

Substituindo as duas opções de u_{-1} na segunda diferença resulta $u_1 + 2u_0 - u_{-1}$.

- Simetria no ponto de malha: $u_1 = u_1$: $2u_0 - 2u_1$;
- Simetria no ponto médio: $u_1 = u_0$: $u_0 - u_1$.

Logo as duas linha superiores possíveis para a matriz A:

Ponto de Malha: $\begin{bmatrix} \otimes & \otimes \end{bmatrix} = 2 \ - \ 2$ e Ponto Médio: $\begin{bmatrix} \otimes & \otimes \end{bmatrix} = 1 \ - \ 1$.

- No outro extremo vão existir as mesmas opções na aproximação de $u'(\pi) = 0$. Fazendo a substituição de $u_N = u_{N-2}$ ou $u_N = u_{N-1}$ na segunda diferença $-u_{N-2} + 2u_{N-1} + u_N$, que nos dá as duas formas para a última linha da matriz A:

Ponto de Malha: $\begin{bmatrix} \otimes & \otimes \end{bmatrix} = -2 \ 2$ e Ponto Médio: $\begin{bmatrix} \otimes & \otimes \end{bmatrix} = -1 \ 1$.

- Na condição de Dirichlet $u(\pi) = 0$. Na condição de pontos de malha $u_N = 0$ remove o último termo de $-u_{N-2} + 2u_{N-1} + u_N$:

Ponto de Malha: $\begin{bmatrix} \otimes & \otimes \end{bmatrix} = -1 \ 2$ e Ponto Médio: $\begin{bmatrix} \otimes & \otimes \end{bmatrix} = -1 \ 3$.

Agora, temos $2 \times 4 = 8$ combinações. Quatro delas correspondem às funções básicas padrão de cossenos, listadas anteriormente como DCT-1 até DCT-4. Essas combinações surgem quando a centralização ocorre nos dois extremos: os dois extremos com simetria centrada nos pontos de malha ou outros extremos com simetria centrados nos ponto médio. Podemos prontamente expressar as matrizes A_1 a A_4 , que têm esses cossenos como autovetores.

4.8 AS TRANSFORMADAS PADRÃO DO COSSENO

DCT-1: Com a escolha da condição de contorno de Dirichlet, ou seja, $u'(0) = 0$, $u'(\pi) = 0$, com simetria em torno do ponto de malha, obtem-se a seguinte matriz A_1 :

$$A_1 = \begin{pmatrix} 2 & -2 & & & \\ -1 & 2 & -1 & & \\ \vdots & & \ddots & & \\ & & & -1 & 2 & -1 \\ & & & & -2 & 2 \end{pmatrix}$$

$D_1 = \text{diag}(\sqrt{2}, 1, \dots, 1, \sqrt{2})$ e centros $j = 0$ e $N - 1$.

Os autovetores da matriz correspondem as funções bases da DCT-1, os vetores da base são da forma $\cos(jk \frac{\pi}{N-1})$.

DCT-2: Condições de contorno $u'(0) = 0$, $u'(\pi) = 0$, com simetria em torno do ponto médio, resultando na matriz A_2 :

$$A_2 = \begin{pmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ \vdots & & \ddots & & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 1 \end{pmatrix}$$

$D_2 = I$ e centros $j = \frac{1}{2}$ e $N - \frac{1}{2}$.

Os autovetores da matriz correspondem as funções bases da DCT-2, com vetores de base da forma $\cos((j + \frac{1}{2})k \frac{\pi}{N})$.

DCT-3: Escolhendo as condições de contorno $u'(0) = 0$, $u(\pi) = 0$, com simetria em torno do ponto de malha, resultando a matriz A_3 .

$$A_3 = \begin{pmatrix} 2 & -2 & & & \\ -1 & 2 & -1 & & \\ \vdots & & \ddots & & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{pmatrix}$$

$D_3 = \text{diag}(\sqrt{2}, 1, \dots, 1)$ e centros $j = 0$ e N .

Os autovetores da matriz correspondem as funções bases da DCT-3, com vetores de base da forma $\cos(j(k + \frac{1}{2}) \frac{\pi}{N})$.

DCT-4: Escolhendo as condições de contorno $u'(0) = 0$, $u(\pi) = 0$, e simetria em torno do ponto médio, obtemos a seguinte matriz.

$$A_4 = \begin{pmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ \vdots & & \ddots & & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 3 \end{pmatrix}$$

$D_4 = I$ e centros $j = -\frac{1}{2}$ e $N - \frac{1}{2}$.

Os autovetores da matriz correspondem as funções bases da DCT-4, com vetores de base da forma $\cos\left((j + \frac{1}{2})\left(k + \frac{1}{2}\right)\frac{\pi}{N}\right)$.

4.9 USO DA DCT PARA COMPRESSÃO DE IMAGENS

A Transformada Discreta de Cosseno (DCT) é uma ferramenta matemática utilizada em compressão de imagens, especialmente em formatos de imagem como JPEG.

A DCT transforma os valores dos pixels do domínio espacial para o domínio da frequência, tendo a capacidade de concentrar a maior parte da energia de uma imagem em um pequeno número de coeficientes de baixa frequência. Isso significa que muitos dos coeficientes de alta frequência (que representam detalhes finos) podem ser descartados sem uma perda perceptível de qualidade, resultando em uma compressão eficiente. Muitos coeficientes da DCT são pequenos ou zero, o que facilita a compactação usando técnicas como quantização e codificação de entropia.

A DCT é eficaz em manter a qualidade visual da imagem mesmo após a compressão. A percepção humana é menos sensível a erros em altas frequências, permitindo uma compressão significativa sem perda perceptível de qualidade.

ÍNDICE DE SIMILARIDADE (SSIM)

O Índice de Similaridade Estrutural (SSIM) é uma métrica usada para medir a similaridade entre duas imagens. Diferente de outras métricas como o Erro Quadrático Médio (MSE), que avaliam diferenças absolutas em termos de intensidade de pixel, o SSIM leva em conta a percepção visual, considerando fatores como luminância, contraste e estrutura. O valor do SSIM varia entre -1 e 1, onde 1 indica correspondência perfeita entre as duas imagens, e valores próximo de 0 indica baixa similaridade entre as imagens, mais detalhes sobre SSIM pode ser encontrados em [10].

METODOLOGIA

- Utilizando o software Python, realizamos o Pré-processamento, que é a divisão da imagem em blocos menores (8×8 pixels). Isso é feito para facilitar a aplicação da DCT e para localizar a compressão em pequenas regiões da imagem;
- Imagem original será convertida para escala cinza, removendo assim os canais de cores RGB da imagem;
- Aplicamos a DCT a cada bloco de 8×8 pixels, convertendo os valores dos pixels (domínio espacial) para coeficientes de frequência;
- Após a aplicação da DCT realizamos a quantização dos coeficientes, que é o processo de reduzir a precisão dos coeficientes de frequência. Isso é feito dividindo cada coeficiente por um valor específico de uma matriz de quantização e arredondando o resultado. Esse passo é crucial para a compressão, pois introduz perdas controladas que reduzem a quantidade de dados. A matriz de quantização usada nesse trabalho é a Q_{50} ;
- Realizamos a desquantização, onde os coeficientes quantizados são multiplicados pelos valores da matriz de quantização para aproximar os coeficientes DCT originais;
- A Transformada Discreta de Cosseno Inversa (IDCT) é aplicada aos blocos de 8×8 para transformar os dados do domínio da frequência de volta ao domínio espacial;
- Realizamos a descompressão da imagem, onde os blocos de 8×8 são recombinaados para formar a imagem descomprimida final;
- Para avaliarmos a qualidade e compressão da imagem utilizamos o Erro Médio Quadrático (MSE) que mede a diferença média dos quadrados dos valores de pixel entre a imagem original e a imagem comprimida e o Índice de Similaridade Estrutural (SSIM) que avalia a similaridade entre duas imagens. Para aplicarmos a compressão usando DCT iremos utilizar a imagem (6) de tamanho (722×720).



Figura 6 – Imagem Original de tamanho (722×720) .

Resultados

Ao analisar a imagem (13) que é a reconstrução com 100% de coeficientes, percebemos que, mesmo após os processos de compressão e descompressão, a qualidade visual da imagem permanece muito próxima da imagem (7). Essa característica é uma demonstração eficaz da Transformada Discreta do Cosseno (DCT) na compressão de imagens. Para avaliar quantitativamente essa eficiência, calculamos o Erro Médio Quadrático (MSE) e o Índice de Similaridade Estrutural (SSIM) para cada porcentagem, resultados estão na tabela 1, assim conseguimos ver que conforme aumentamos o percentual de coeficientes o erro diminuiu e o índice de similaridade aumenta, valores do índice próximo a 1 indicam alta similaridade entre as imagens, isso significa que a qualidade da imagem está ficando melhor e a imagem está muito similar com a imagem original.

Segue tabela 1 com os erros e os índices de similaridade para diferentes porcentagens de coeficientes:



Figura 7 – Imagem na escala cinza

Porcentagem	Erro	Índice de similaridade
5%	6361.21	0.0996
10%	5724.34	0.1847
25%	3978.76	0.4433
40%	2551.01	0.6403
70%	644.28	0.8595
100%	6.56	0.9666

Tabela 1 – Erros e índice de similaridade para cada porcentagem de coeficientes.

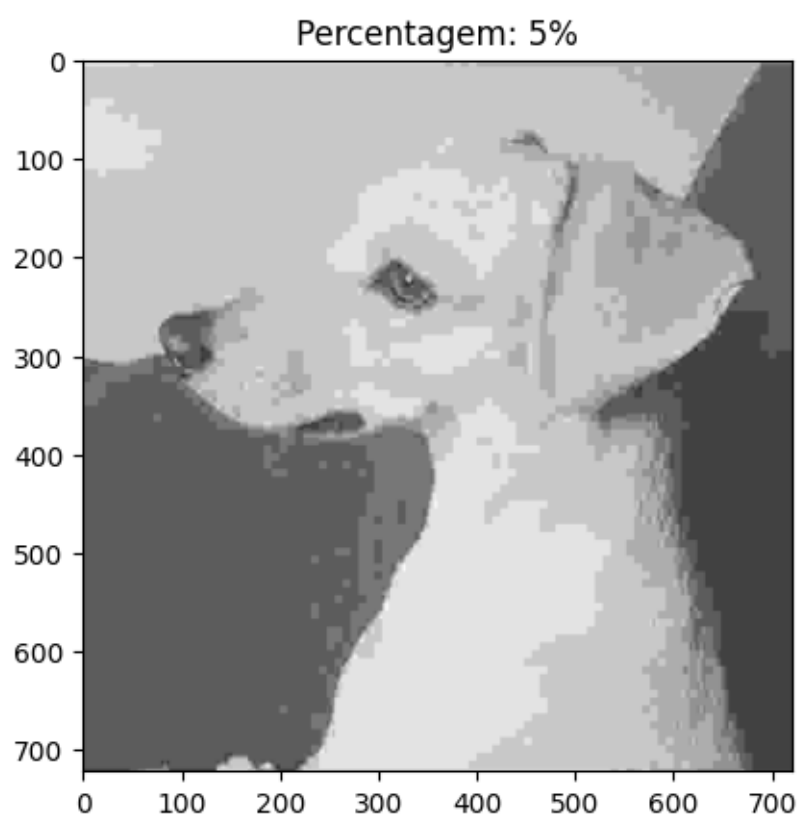


Figura 8 – Imagem com 5% de coeficientes.

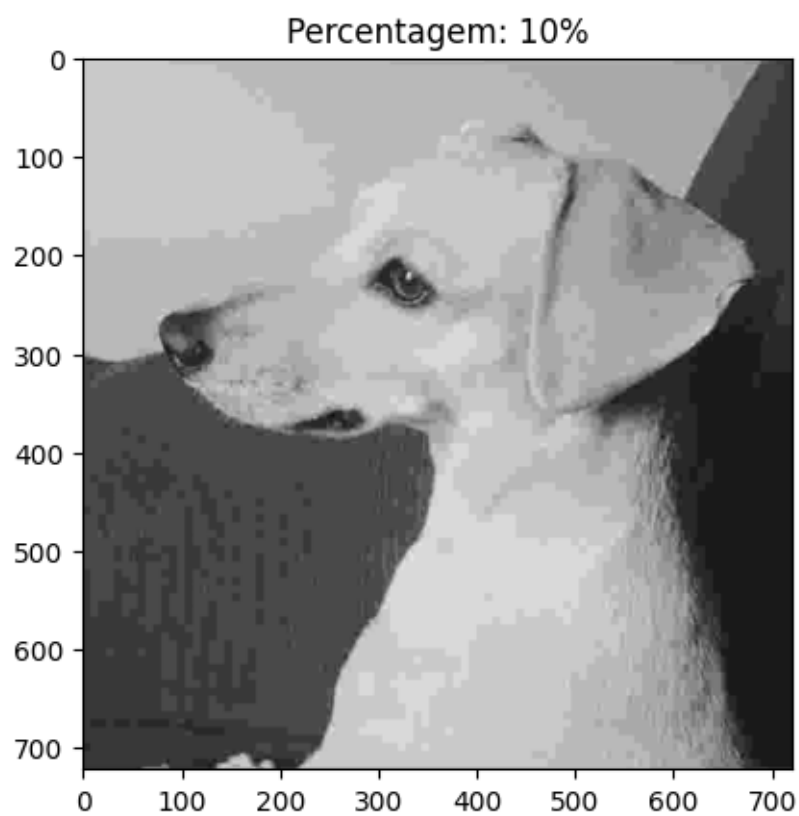


Figura 9 – Imagem com 10% de coeficientes.

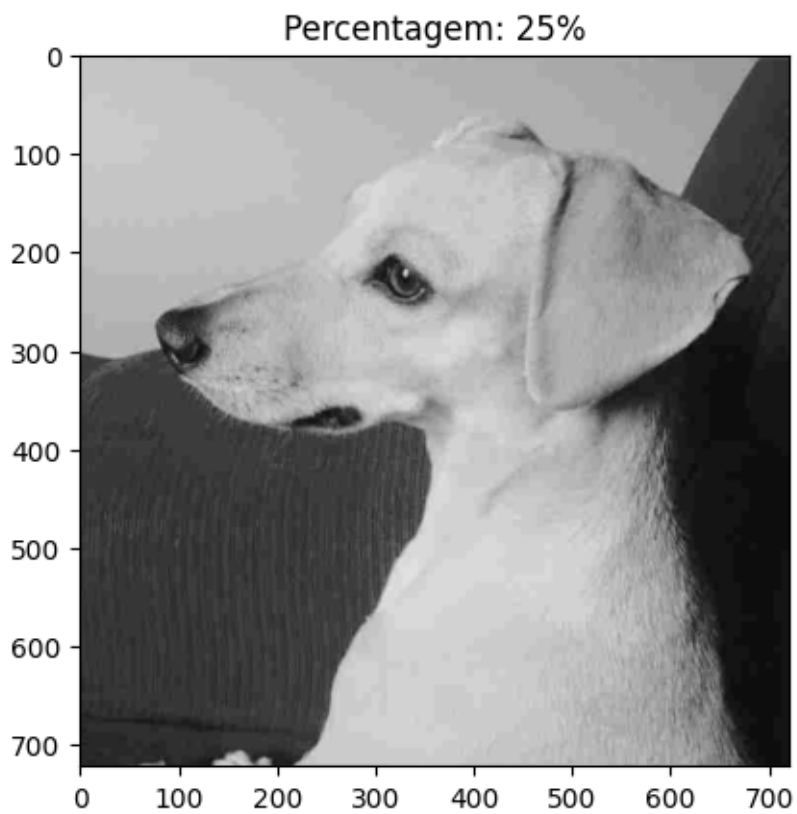


Figura 10 – Imagem com 25% de coeficientes.

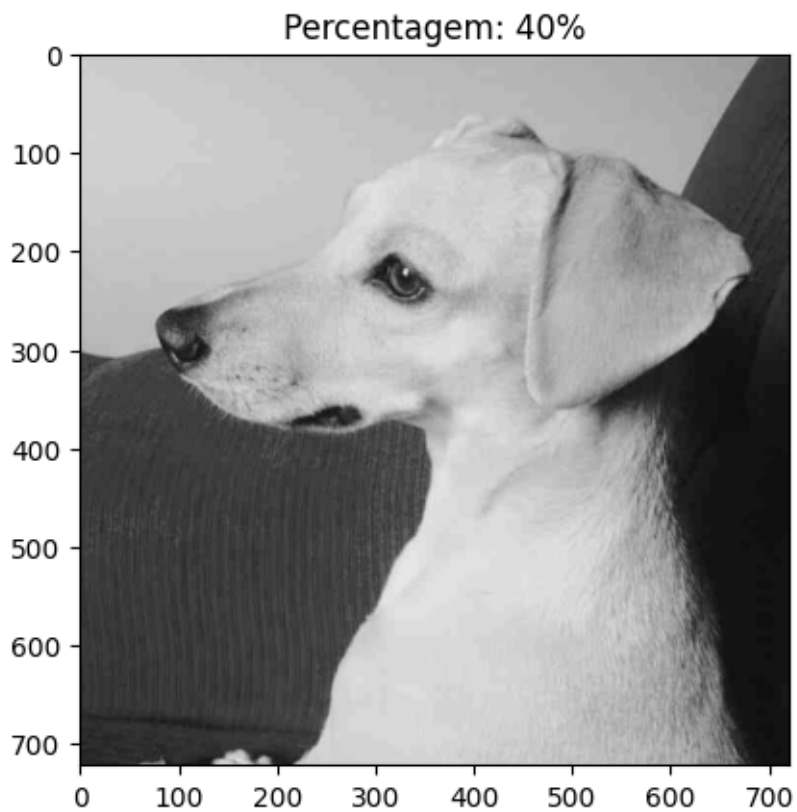


Figura 11 – Imagem com 40% de coeficientes.

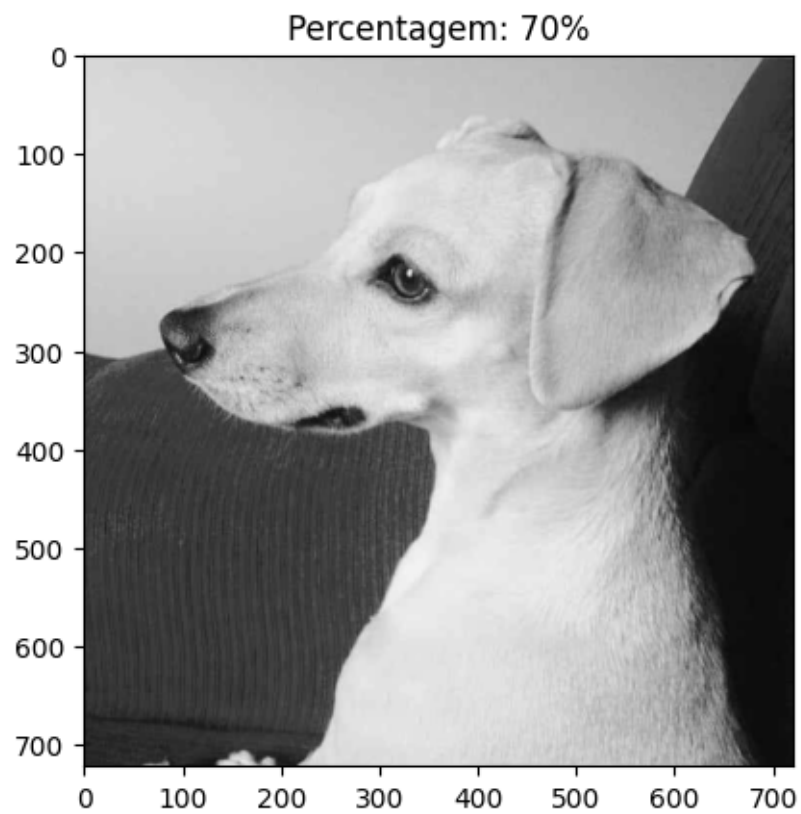


Figura 12 – Imagem com 70% de coeficientes.

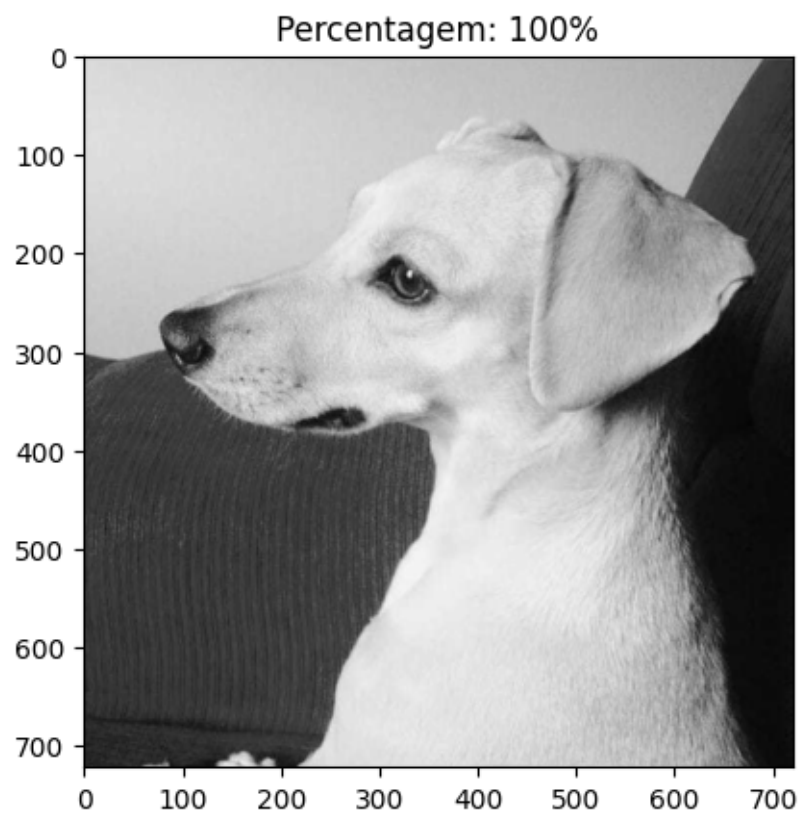


Figura 13 – Imagem com 100% de coeficientes.

5 CONCLUSÕES

Este trabalho apresentou um estudo detalhado sobre as transformadas de Fourier (DFT) e do Cosseno (DCT), explorando suas propriedades matemáticas e aplicações práticas no processamento de sinais e compressão de imagens. A DFT revelou-se uma ferramenta eficaz para a análise de sinais no domínio da frequência, permitindo a identificação e eliminação de ruídos. Por outro lado, a DCT mostrou-se particularmente eficiente na compressão de dados, como imagens digitais, ao concentrar a energia do sinal em um número reduzido de coeficientes.

A análise comparativa entre DFT e DCT destacou as vantagens e desvantagens de cada transformada em diferentes contextos. Enquanto a DFT é amplamente utilizada para a filtragem e modulação de sinais, a DCT é preferida em aplicações de compressão de mídia devido à sua capacidade de compactação de energia.

Além disso, a implementação prática das transformadas em Python demonstrou a aplicabilidade dos conceitos teóricos, proporcionando uma compreensão mais profunda de suas funcionalidades e limitações. As simulações realizadas confirmaram a eficácia da DFT na remoção de ruídos e a superioridade da DCT na compressão de imagens, com resultados que corroboram com a literatura existente.

REFERÊNCIAS

- [1] Syed Ali Khayam. *The Discrete Cosine Transform (DCT): Theory and Application*. Michigan State University, 2003.
- [2] Timothy Sauer. *Numerical Analysis. Compression*. 2005. Cap. 1.
- [3] Gilbert Strang. “The Discrete Cosine Transform”. Em: *Siam publications library* (1999). URL: <https://epubs.siam.org/doi/pdf/10.1137/S0036144598336745>.
- [4] *mcap03, Um capítulo sobre Serie de Fourier*. URL: <https://www.feis.unesp.br/Home/departamentos/engenhariaeletrica/mcap03.pdf>.
- [5] Miguel Frasson. *Transformada Discreta de Cosseno: uma aplicação da Algebra Linear na compressão de imagens do formato JPEG*. 2013. URL: <https://sites.icmc.usp.br/frasson/jpeg/jpeg.html>.
- [6] Reginaldo Santos. *Equações Diferenciais Parciais, uma introdução*. Ufmg, 2011.
- [7] T. Natarajan N. Ahmed e K. R. Rao. “Discrete Cosine Transform”. Em: *IEEE Transactions on Computers*, vol. C-23, no. 1. (1974), pp. 90–93.
- [8] Djairo Guedes de Figueiredo. *Análise de Fourier e Equações Diferenciais Parciais*. Coleção euclides, IMPA/CNPq, Rio de Janeiro, 1986.
- [9] William L. Briggs e Van Emden Henson. *The DFT: An Owner’s Manual for the Discrete Fourier Transform*. Philadelphia: Society for Industrial e Applied Mathematics, 1995.
- [10] Rafael Borges De Souza. *Métricas em Processamento de Imagens*. 2020. URL: <http://mtm.ufsc.br/~leo/Rafael-2019-2020.pdf>.

A CÓDIGO DA APLICAÇÃO DFT

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # Parâmetros do sinal
5 k1 = 250
6 k2 = 350
7 t = np.arange(0, 1, 0.001)
8 N = len(t)
9
10 # Função original
11 f = np.sin(2 * np.pi * k1 * t) + np.sin(2 * np.pi * k2 * t)
12
13 # Adicionando ruído
14 noise = np.random.rand(N)
15 f_noisy = f + noise
16
17 # Calculando a DFT
18 F = np.fft.fft(f_noisy)
19 PSD = (1 / N) * np.abs(F) ** 2
20
21 # Aplicando filtro
22 threshold = 100
23 F_filtered = F.copy()
24 F_filtered[PSD < threshold] = 0
25
26 # Recuperando o sinal filtrado
27 f_filtered = np.fft.ifft(F_filtered)
28
29 # Plotando os resultados
30 plt.figure(figsize=(12, 8))
31
32 plt.subplot(4, 1, 1)
33 plt.plot(t, f)
34 plt.title('Sinal Original')
35
36 plt.subplot(4, 1, 2)
37 plt.plot(t, f_noisy)
38 plt.title('Sinal Perturbado')
39
40 plt.subplot(4, 1, 3)
41 plt.plot(PSD)
42 plt.title('Densidade Espectral de Potência')
```

```
43
44 plt.subplot(4, 1, 4)
45 plt.plot(t, np.real(f_filtered))
46 plt.title('Sinal Recuperado')
47
48 plt.tight_layout()
49 plt.show()
```

B CÓDIGO DA APLICAÇÃO DCT

```

1 import numpy as np
2 import cv2
3 from scipy.fftpack import dct, idct
4 import matplotlib.pyplot as plt
5 from skimage.metrics import structural_similarity as ssim
6
7 # Função para padronizar a imagem
8 def pad_image(img):
9     h, w = img.shape
10    m = max(h, w)
11    flag = 3
12    while m > (2**flag):
13        flag += 1
14    padded_img = np.zeros((2**flag, 2**flag))
15    padded_img[:h, :w] = img
16    return padded_img
17
18 # Matriz de quantização
19 Q50 = np.array([
20     [16, 11, 10, 16, 24, 40, 51, 61],
21     [12, 12, 14, 19, 26, 58, 60, 55],
22     [14, 13, 16, 24, 40, 57, 69, 56],
23     [14, 17, 22, 29, 51, 87, 80, 62],
24     [18, 22, 37, 56, 68, 109, 103, 77],
25     [24, 35, 55, 64, 81, 104, 113, 92],
26     [49, 64, 78, 87, 103, 121, 120, 101],
27     [72, 92, 95, 98, 112, 100, 103, 99]
28 ])
29
30 # Função para quantizar a matriz
31 def quantize(D, Q):
32     return np.round(D / Q)
33
34 # Função para desquantizar a matriz
35 def dequantize(C, Q):
36     return C * Q
37
38 # Função para criar a matriz DCT
39 def dct_matrix(n):
40     mat = np.zeros((n, n))
41     for i in range(n):
42         for j in range(n):

```

```

43         if i == 0:
44             mat[i, j] = 1 / np.sqrt(n)
45         else:
46             mat[i, j] = np.sqrt(2 / n) * np.cos((2 * j + 1) * i * np
47             .pi / (2 * n))
48     return mat
49 # Função para aplicar a DCT em blocos 8x8
50 def apply_dct(img, block_size, quantize_func=None, Q=None, percentage
51     =100):
52     h, w = img.shape
53     T = dct_matrix(block_size)
54     dct_img = np.zeros_like(img)
55     Q = Q * (100 / percentage) # Ajustar a matriz de quantiza o
56     conforme o percentual
57     for i in range(0, h, block_size):
58         for j in range(0, w, block_size):
59             block = img[i:i + block_size, j:j + block_size]
60             dct_block = T @ block @ T.T
61             if quantize_func and Q is not None:
62                 dct_block = quantize_func(dct_block, Q)
63             dct_img[i:i + block_size, j:j + block_size] = dct_block
64     return dct_img
65 # Função para aplicar a IDCT em blocos 8x8
66 def apply_idct(dct_img, block_size, dequantize_func=None, Q=None):
67     h, w = dct_img.shape
68     T = dct_matrix(block_size)
69     idct_img = np.zeros_like(dct_img)
70     for i in range(0, h, block_size):
71         for j in range(0, w, block_size):
72             block = dct_img[i:i + block_size, j:j + block_size]
73             if dequantize_func and Q is not None:
74                 block = dequantize_func(block, Q)
75             idct_block = T.T @ block @ T
76             idct_img[i:i + block_size, j:j + block_size] = idct_block
77     return idct_img
78 # Função para calcular o MSE
79 def calculate_mse(original, compressed):
80     return np.mean((original - compressed) ** 2)
81
82 # Função para calcular o SSIM
83 def calculate_ssim(original, compressed):
84     return ssim(original, compressed, data_range=compressed.max() -
85     compressed.min())

```

```

86 # Função para manter uma porcentagem específica dos coeficientes
87 def keep_percentage(dct_img, percentage):
88     flattened = np.abs(dct_img.flatten())
89     threshold = np.percentile(flattened, 100 - percentage)
90     dct_img[np.abs(dct_img) < threshold] = 0
91     return dct_img
92
93 def process_image(image_path):
94     img = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
95     img = img.astype(np.float32) - 128
96     og_h, og_w = img.shape
97     img_padded = pad_image(img)
98
99     # Exibir imagem original
100    plt.figure()
101    plt.title('Imagem na escala cinza')
102    plt.imshow(img + 128, cmap='gray')
103    plt.show()
104
105    percentages = [5, 10, 25, 40, 70, 100]
106    mse_values = []
107    ssim_values = []
108
109    for p in percentages:
110        dct_img_with_quant = apply_dct(img_padded, 8, quantize, Q50, p)
111
112        # Normalizar para visualização
113        dct_img_with_quant_norm = (dct_img_with_quant - np.min(
dct_img_with_quant)) / (np.max(dct_img_with_quant) - np.min(
dct_img_with_quant)) * 255
114        plt.figure()
115        plt.title(f'DCT with Quantization (p={p}%)')
116        plt.imshow(dct_img_with_quant_norm[:og_h, :og_w], cmap='gray')
117        plt.show()
118
119        idct_img = apply_idct(dct_img_with_quant, 8, dequantize, Q50)[:
og_h, :og_w] + 128
120
121        mse = calculate_mse(img + 128, idct_img)
122        ssim_index = calculate_ssim(img + 128, idct_img)
123        mse_values.append(mse)
124        ssim_values.append(ssim_index)
125
126        print(f"Porcentagem: {p}% - MSE: {mse:.2f} - SSIM: {ssim_index
:.4f}")
127
128    # Visualizar a imagem comprimida

```

```
129     plt.figure()
130     plt.title(f'Percentagem: {p}%')
131     plt.imshow(idct_img, cmap='gray')
132     plt.show()
133
134 # Nome do arquivo de entrada
135 image_path = input('File name + extension (must be in current dir.): ')
136 process_image(image_path)
```