

# 1 semanario NTT

## 2 A Dualidade entre Tempo e Frequência

A Transformada de Fourier é uma operação matemática que mapeia uma função do domínio do tempo (ou espaço) para o seu domínio dual: a frequência. Essa transição é extremamente útil, pois propriedades que são complexas de analisar no tempo tornam-se claras no espectro de frequências.

Além da análise, essa mudança de espaço possibilita a simplificação de operações fundamentais.

## 3 Aplicabilidade e Importância

Esta ferramenta é um pilar fundamental em diversas áreas do conhecimento:

- **Matemática Pura:** Essencial na Teoria Analítica dos Números e no estudo de Equações Diferenciais Parciais (EDPs).
- **Física Moderna:** É a base matemática do **Princípio da Incerteza de Heisenberg** na Mecânica Quântica, onde a posição e o momento de uma partícula formam um par de variáveis conjugadas de Fourier.
- **Engenharia:** Processamento de sinais, compressão de dados (MP3, JPEG) e telecomunicações.

## 4 Transformada de Fourier Contínua (CTFT)

Para uma função contínua  $f(t)$ , a transformada é definida pela integral:

$$F(f) = \int_{-\infty}^{\infty} g(t)e^{-2\pi ift} dt$$

Ela pode ser entendida como um produto interno (uma projeção) do sinal com todos as frequências da reta real  $\langle g, e^{-2\pi ift} \rangle$ , que, devido a ortogonalidade das frequências diferentes e que funções bem comportadas podem ser decompostas em séries de autofunções  $e^{-ift}$ , consegue extrair exatamente as frequências do sinal. Apesar de sua elegância teórica, a CTFT apresenta desafios para a aplicação prática em sistemas digitais:

1. **Natureza Analítica:** A resolução de integrais impróprias exige uma manipulação simbólica que é difícil de implementar em computadores comuns.
2. **Limite Infinito:** A definição pressupõe que conhecemos o sinal de  $-\infty$  a  $+\infty$ , o que é impossível em cenários reais.
3. **Amostragem Finita:** Na prática, os sinais são capturados de forma discreta (amostras) e por um tempo limitado, o que torna a integral contínua inaplicável.

## 5 Transformada Discreta de Fourier (DFT)

Para viabilizar o processamento em computadores, utilizamos a **DFT**. Ela opera sobre uma sequência finita de  $N$  amostras, mapeando dados discretos no tempo para dados discretos na frequência:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-i \frac{2\pi}{N} kn}$$

Para  $k = 0, 1, \dots, N-1$ .

Diferente da versão contínua, a DFT lida com somatórios e vetores numéricos, permitindo que a teoria de Fourier seja aplicada em qualquer dispositivo digital. É possível provar que a DFT é um transformação linear, logo, pode ser representada matricialmente.

Seja  $\zeta_N = e^{-i \frac{2\pi}{N}}$ . A representação matricial da DFT para  $n = 0, 1, \dots, N-1$  é:

$$\begin{bmatrix} X[0] \\ X[1] \\ \vdots \\ X[N-1] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \zeta_N^1 & \zeta_N^2 & \dots & \zeta_N^{N-1} \\ 1 & \zeta_N^2 & \zeta_N^4 & \dots & \zeta_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \zeta_N^{N-1} & \zeta_N^{2(N-1)} & \dots & \zeta_N^{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix}$$

### 5.1 Definição da IDFT

A reconstrução do sinal original no domínio do tempo a partir de suas amostras de frequência é realizada pela IDFT:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \zeta_N^{-nk}, \quad n = 0, 1, \dots, N-1$$

Onde  $\zeta_N^{-nk} = e^{i \frac{2\pi}{N} nk}$ . Matricialmente, a IDFT é dada por:

$$\begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ \vdots \\ x[N-1] \end{bmatrix} = \frac{1}{N} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \zeta_N^{-1} & \zeta_N^{-2} & \dots & \zeta_N^{-(N-1)} \\ 1 & \zeta_N^{-2} & \zeta_N^{-4} & \dots & \zeta_N^{-2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \zeta_N^{-(N-1)} & \zeta_N^{-2(N-1)} & \dots & \zeta_N^{-(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ \vdots \\ X[N-1] \end{bmatrix}$$

Percebe-se que nesse caso discreto, a DFT e IDFT atuam como uma matriz mudança de base, saindo da base do tempo e indo para base das raízes unitárias

## 6 A Multiplicação de Polinômios e a Complexidade Computacional

Um problema simplificado pela mudança de domínio é a multiplicação de polinômios. Tome os polinômios  $f(x)$  e  $g(x)$  de grau  $n-1$ :

$$f(x) = \sum_{i=0}^{n-1} a_i x^i, \quad g(x) = \sum_{j=0}^{n-1} b_j x^j$$

Na abordagem clássica, o produto  $h(x) = f(x) \cdot g(x)$  é obtido distribuindo-se cada termo de  $f$  sobre todos os termos de  $g$ . Este processo resulta em um novo polinômio de grau  $2n - 2$ :

$$h(x) = \sum_{k=0}^{2n-2} c_k x^k$$

onde  $c_k = \sum_{i+j=k} a_i b_j$ .

Nesta metodologia, o cálculo de cada coeficiente  $c_k$  exige múltiplas operações de produto e soma, resultando em uma complexidade assintótica  $O(n^2)$ . Para polinômios com grandes volumes de coeficientes, este custo computacional torna o método inviável.

## 7 O Teorema da Convolução

A conexão entre a multiplicação de polinômios e a análise de Fourier vem do fato de que, se  $h(x) = f(x)g(x)$ , então os coeficientes  $c_k$  de  $h$  são dados pela **convolução linear** dos coeficientes de  $f$  e  $g$ :

$$c_k = \sum_{i+j=k} a_i b_j.$$

Neste trabalho, devido ao foco na NTT e ao anel quociente  $\mathbb{Z}_p[x]/(x^n - 1)$ , trabalhamos com a **convolução circular** (de comprimento  $n$ ), definida por

$$c_k = \sum_{i=0}^{n-1} a_i b_{(k-i) \bmod n}, \quad k = 0, \dots, n-1.$$

**Teorema da Convolução:** a transformada de uma convolução no domínio do tempo (ou espaço) é o produto ponto a ponto (Hadamard) das transformadas no domínio da frequência:

$$\mathcal{F}(f * g) = \mathcal{F}(f) \odot \mathcal{F}(g).$$

Assim, o cálculo custoso da convolução é convertido em um produto ponto a ponto, pois a base de Fourier diagonaliza o operador de convolução (circulante). Com a transformada direta ingênua o custo ainda é  $O(n^2)$ , não há ganho da multiplicação clássica de polinômios.

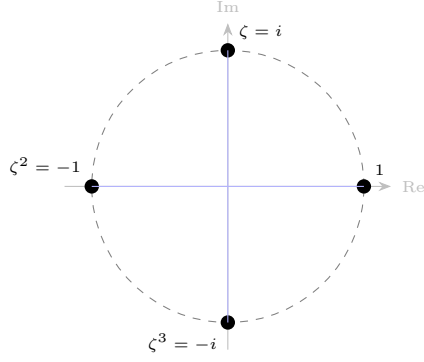
## 8 A Fast Fourier Transform

A FFT (Fast Fourier Transform) é uma maneira de otimizar o cálculo da DFT.

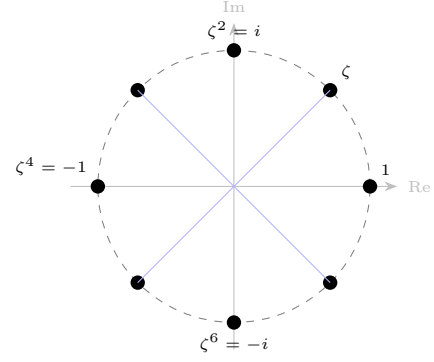
O algoritmo da FFT foi redescoberto por Cooley e Tukey em 1965, uma vez que Gauss já tinha utilizado um algoritmo semelhante para calcular as órbitas de asteroides em 1805.

O algoritmo se baseia em **dividir para conquistar**.

Relembrando



(a) 4-ésimas raízes da unidade



(b) 8-ésimas raízes da unidade

Figura 1: Comparação entre as raízes da unidade no plano complexo.

As raízes unitárias possuem propriedades cíclicas e certas simetrias que permitem a economia nos cálculos, vejamos um exemplo.

$$\zeta_4^1 = e^{i\frac{2\pi}{4}} = e^{i90^\circ} = i$$

$$\zeta^1 = i \quad \zeta^2 = -1$$

$$\zeta^3 = -i \quad \zeta^4 = 1$$

por isso, percebe-se que, a cada 2 "deslocamentos", o valor se torna o oposto, como ilustrado na figura:

De forma mais geral:

$$\zeta_N = e^{\frac{2\pi i}{N}}, \text{ uma raiz } N\text{-ésima primitiva da unidade. Então para todo inteiro } a, \\ \zeta_N^{a+\frac{N}{2}} = -\zeta_N^a.$$

Alem de que, pela periodicidade  $\zeta_N^{a+N} = \zeta_N^a$ . Para esse caso a DFT eh representada desse modo:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{bmatrix}$$

Voltando a FFT, o algoritmo decompõe uma DFT de tamanho  $N$  em duas sub-transformadas de tamanho  $N/2$ , separando os índices pares e ímpares da sequência original:

$$\begin{aligned} X[k] &= \sum_{m=0}^{\frac{N}{2}-1} x[2m] \zeta_N^{2mk} + \sum_{m=0}^{\frac{N}{2}-1} x[2m+1] \zeta_N^{(2m+1)k} \\ &= \sum_{m=0}^{\frac{N}{2}-1} x[2m] \zeta_{N/2}^{mk} + \zeta_N^k \sum_{m=0}^{\frac{N}{2}-1} x[2m+1] \zeta_{N/2}^{mk} \\ &= E[k] + \zeta_N^k O[k], \quad k = 0, \dots, \frac{N}{2} - 1. \end{aligned}$$

Esta estrutura permite calcular dois valores de saída ( $X[k]$  e  $X[k + N/2]$ ) utilizando os mesmos resultados intermediários, através da denominada **operação borboleta** (*butterfly operation*):

1.  $X[k] = E[k] + \zeta_N^k O[k]$
2.  $X[k + N/2] = E[k] - \zeta_N^k O[k]$

como pode ser visto na imagem

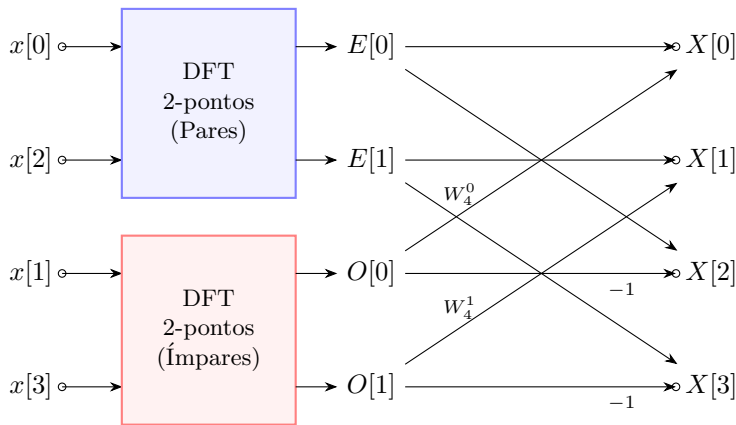


Figura 2: butterfly radix-2

o a implementação em sage esta no codigol

```

1 def fft(a, omega):
2     n = len(a)
3     if n == 1:
4         return a[:]
5
6
7     a_par = fft(a[0::2], omega^2)
8     a_impar = fft(a[1::2], omega^2)
9
10    A = [0] * n
11    w = 1
12    half = n // 2
13    for k in range(half):
14        t = w * a_impar[k]
15        A[k] = a_par[k] + t
16        A[k + half] = a_par[k] - t
17        w *= omega
18    return A

```

Listing 1: Implementação do algoritmo FFT em SageMath

Desse modo, reduzimos a complexidade da transformada de  $O(n^2)$  para  $O(n \cdot \log n)$ . Por causa disso, podemos utilizar a FFT, junto com o **teorema da convolucao**, para multiplicar polinomios em  $O(n \cdot \log n)$

## 9 Problemas da FFT

Uns dos problemas da FFT eh que ela trabalha com ponto flutuante, o que, para computadores, eh um grande problemas que pode causar erro de arredondamentos e, assim causar um falha nos esquemas criptograficos. Alem disso, o polinomio dobram de tamanho a cada concolucao o que rapidamente torna-se um problema tanto computacional quanto de armazenamento .

Solucao: utilizar um transformada que utiliza apenas numeros exatos

## 10 Number Theoretic Transform (NTT)

### 10.1 Fundamentos

As propriedades que usamos na FFT — em especial a existência de uma raiz  $N$ -ésima da unidade  $\zeta_N$  e o fato de que suas potências percorrem uniformemente o círculo — têm um análogo perfeito em teoria dos números, dentro de corpos (ou anéis) finitos. Isso não é coincidência: a FFT nada mais é do que a transformada de Fourier no grupo cíclico  $\mathbb{Z}/N\mathbb{Z}$ , e a mesma construção existe em outros contextos algébricos.

Mais formalmente, se  $\zeta_N$  é uma raiz  $N$ -ésima primitiva da unidade, então o conjunto de todas as  $N$ -ésimas raízes

$$\mu_N = \{1, \zeta_N, \zeta_N^2, \dots, \zeta_N^{N-1}\}$$

forma um grupo multiplicativo cíclico de ordem  $N$ . Existe um isomorfismo natural de grupos

$$\varphi : \mathbb{Z}/N\mathbb{Z} \rightarrow \mu_N, \quad \varphi([k]) = \zeta_N^k,$$

onde o lado esquerdo usa a soma módulo  $N$  e o lado direito usa multiplicação:

$$\varphi([k + \ell]) = \zeta_N^{k+\ell} = \zeta_N^k \zeta_N^\ell = \varphi([k]) \varphi([\ell]).$$

**Raiz primitiva** Diferente da DFT complexa, onde raízes da unidade sempre existem para qualquer  $n$ , a NTT exige que o corpo finito  $\mathbb{Z}_p$  suporte a ordem da transformada.

Para que exista uma raiz primitiva  $n$ -ésima da unidade em  $\mathbb{Z}_p$ , o tamanho da sequência  $n$  deve dividir a ordem do grupo multiplicativo do corpo:

$$n \mid (p - 1)$$

Agora, iremos tratar da estrutura aritimetica, na qual a NTT ocorre.

Ela eh definida da seguinte forma:

$$X[k] = \sum_{n=0}^{N-1} x[n] \omega^{nk} \pmod{p}$$

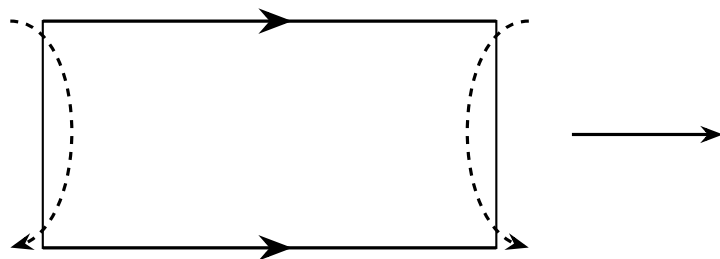
Ela opera no anel quociente

$$R = \mathbb{Z}_p[x]/(x^n - 1)$$

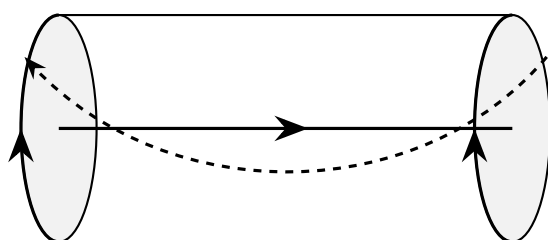
onde  $\mathbb{Z}_p[x]$  representa o coeficientes do polinomio *mod*  $p$  e o  $/(x^n - 1)$  faz com que a cadeia longa de polinomio dobre em si mesma formando um ciclo.

a imagem 3 ilustra a sequencia das duas operacoes visualmente.

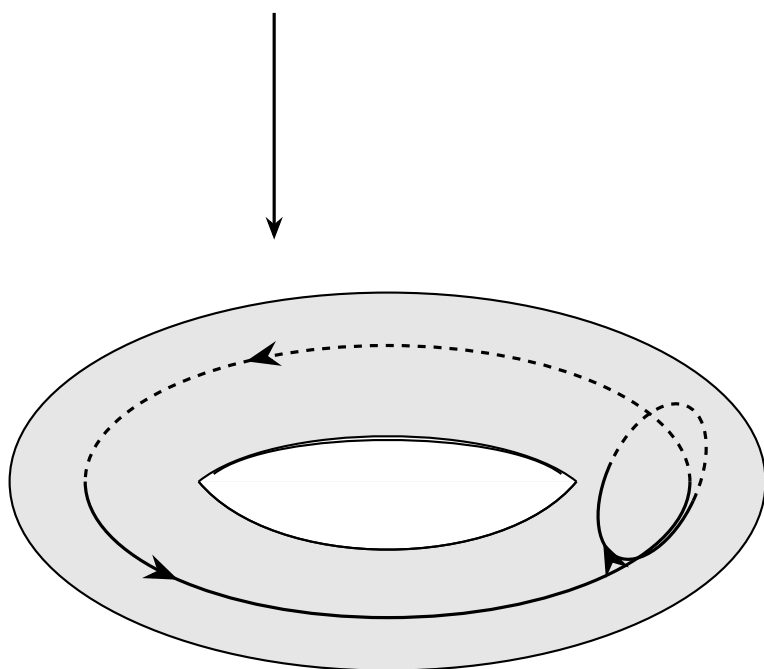
1. Colar lados horizontais (Passo  $\mathbb{Z}_p$ )



Obtém-se um cilindro



2. Colar lados verticais (Passo  $/(x^n - 1)$ )



3. Obtém-se um Toro

Figura 3: representação das transformacoes



Escolhemos modulo de um primo para que  $\mathbb{Z}_p$  seja um field, i.e, para que a aritmedica tenha propiedades agradaveis e escolhemos  $(x^n - 1)$  para que haja raizes distinta e se preserve as "informacoes" indepentens o que garante que ele possa ser decomposto completamente.

## 11 O Isomorfismo via Teorema Chinês dos Restos (CRT)

A fundamentação algébrica da NTT reside na decomposição do anel de polinômios. Se  $\omega$  é uma raiz primitiva  $n$ -ésima da unidade no corpo finito  $\mathbb{Z}_p$ , o polinômio  $x^n - 1$  pode ser fatorado completamente em binômios lineares distintos:

$$x^n - 1 = \prod_{i=0}^{n-1} (x - \omega^i)$$

Como cada termo  $(x - \omega^i)$  é irredutível e todos são coprimos entre si, o **Teorema Chinês dos Restos (CRT)** garante a existência de um isomorfismo de anéis:

$$\frac{\mathbb{Z}_p[x]}{(x^n - 1)} \cong \frac{\mathbb{Z}_p[x]}{(x - \omega^0)} \times \frac{\mathbb{Z}_p[x]}{(x - \omega^1)} \times \dots \times \frac{\mathbb{Z}_p[x]}{(x - \omega^{n-1})}$$

Este isomorfismo é o que permite interpretar a NTT não apenas como uma transformação de vetores, mas como uma mudança de representação.

fonte: <https://github.com/SheafificationOfG/Fibsonisheaf>

Algorithm	Fibonacci index
Algoritmo Naive	44
Algoritmo Linear	566'053
Algoritmo FFT	3'145'816
Algoritmo NTT	24'178'839
Algoritmo GMP	238'961'323