

Escola Politécnica-USP
LARC



Seminário NTT

Resumo

O objetivo do seminário é apresentar a NTT e suas vantagens, para isso, parte-se da transformada de Fourier, em seguida, para a sua forma discreta (DFT) e chega-se na NTT propriamente dita. O que almejo com esse trabalho é passar a intuição por trás das ferramentas matemática utilizadas e beleza inerentes a elas.

Contents

1 A Dualidade entre Tempo e Frequência	2
2 Transformada de Fourier Contínua (CFT)	2
3 Transformada Discreta de Fourier (DFT)	2
3.1 Definição da IDFT	3
4 A Multiplicação de Polinômios e a Complexidade Computacional	4
4.1 Teorema da Convolução	5
5 A "Fast Fourier Transform"	5
5.1 Números complexos	5
6 Problemas da FFT	8
7 Number Theoretic Transform (NTT)	8
7.1 Fundamentos	8
7.1.1 Raiz Primitiva e Estrutura Negacíclica	9
7.2 O Isomorfismo via Teorema Chinês dos Restos (CRT)	12
7.3 Transformada Numérica Inversa (INTT)	13
7.4 A NTT como uma FFT: Decomposição Radix-2	14
7.5 A INTT via Gentleman–Sande (GS): Reconstrução	15
8 Testes comparativos	17
9 NTT incompleta e a restrição de parâmetros no caso negacíclico	18
10 Agradecimentos	20
11 Apêndice	21
11.1 DFT como transformação linear	21
11.2 Exemplo de convolução negacíclica	22
11.3 Determinação unívoca do polinômio	24
11.4 Estrutura Cíclica de Grupos Multiplicativos de Corpos Finitos	25

1 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.

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.

2 Transformada de Fourier Contínua (CFT)

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

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

Elá pode ser entendida como um produto interno (uma projeção) do sinal com todas as frequências da reta real $\langle g, e^{-2\pi i ft} \rangle$, que, devido à ortogonalidade das frequências diferentes e que funções bem comportadas podem ser decompostas em séries de autofunções $e^{-2\pi i ft}$, consegue extrair exatamente as frequências do sinal. Apesar de sua elegância teórica, a CFT 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.

3 Transformada Discreta de Fourier (DFT)

Para viabilizar o processamento em computadores, utilizamos a **DFT** (*Discrete Fourier Transform*). Ela opera sobre uma sequência finita de N amostras, mapeando dados discretos no tempo para o domínio da frequência. Segundo o Teorema de Amostragem de Nyquist-Shannon [7], para que um sinal contínuo possa ser unicamente determinado a partir de suas amostras, a frequência de amostragem f_s deve ser superior ao dobro da maior

frequência f_{max} contida no sinal. $f_s > 2f_{max}$. Com essa restrição para a quantidade de amostras N em mente, a DFT é definida da seguinte forma:

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

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 uma transformação linear 11.1, 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:

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

3.1 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:

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

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

Além disso, a DFT pode ser vista de outra óptica, como avaliação de polinômios.

Seja:

$$a(x) = \sum_{n=0}^{N-1} a_n x^n$$

e $\zeta_N = e^{-i \frac{2\pi}{N}}$ Seja uma N -ésima raiz primitiva da unidade. Define

$$A_k := a(\zeta_N^k).$$

Então

$$A_k = \sum_{n=0}^{N-1} a_n (\zeta_N^k)^n = \sum_{n=0}^{N-1} a_n \zeta_N^{kn},$$

o que é exatamente a formulação da DFT, para $a_n \equiv x[n]$ e $A_n \equiv X[n]$, oportamente, a IDFT é interpretada como interpolação de polinômios:

Dados os valores $\{A_k\}_{k=0}^{N-1}$, ela reconstrói os coeficientes $\{a_n\}_{n=0}^{N-1}$ do único polinômio de grau $N - 1$ que satisfaz $a(\zeta_N^k) = A_k$ para todo k . Explicitamente,

$$a_n = \frac{1}{N} \sum_{k=0}^{N-1} A_k \zeta_N^{-kn}.$$

Assim, DFT é *avaliar* em raízes da unidade e IDFT é *interpolar* (recuperar os coeficientes) a partir dessas avaliações. (Essa perspectiva baseia-se na determinação unívoca do polinômio de grau $N - 1$ por N pontos veja a seção 11.3 para demonstração desse fato)

4 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.

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 negacíclica** (de comprimento N), definida por:

$$c_k = \sum_{i=0}^k a_i b_{k-i} - \sum_{i=k+1}^{N-1} a_i b_{N+k-i}, \quad k = 0, \dots, N-1.$$

Diferente da circular, aqui os termos que “ultrapassam” o comprimento N retornam com sinal invertido.

4.1 Teorema da Convolução

Teorema 1. A transformada de uma convolução no domínio do tempo (ou espaço) é o produto ponto a ponto (Hadamard) das transformadas no domínio da frequência:

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

Assim, o cálculo custoso da convolução é convertido em um produto ponto a ponto, pois a base de Fourier apropriada (utilizando as raízes de $x^N + 1$) diagonaliza o operador de convolução negacíclica (matriz negacirculante). Com a transformada direta ingênuia, o custo ainda seria $O(N^2)$, portanto, não há ganho computacional em realizar essa transformação.

Um exemplo para o convencimento do leitor foi disposto no apêndice 11.2.

5 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**.

5.1 Números complexos

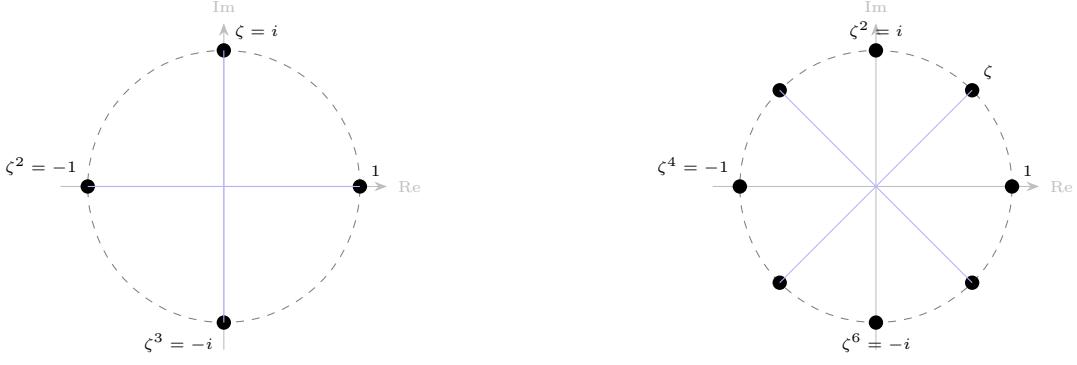
As raízes unitárias possuem propriedades cíclicas e certas simetrias que permitem a economia nos cálculos, vejamos um exemplo.

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

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

$$\zeta^3 = i \qquad \zeta^4 = 1$$

por isso, percebe-se que, a cada 2 ”deslocamentos”, o valor se torna o oposto, como ilustrado na figura 1:



(a) 4-ésimas raízes da unidade

(b) 8-ésimas raízes da unidade

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

De forma mais geral:

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

$$\zeta_N^{a+\frac{N}{2}} = -\zeta_N^a.$$

Além de que, pela periodicidade $\zeta_N^{a+N} = \zeta_N^a$ e $\zeta_N^2 = \zeta_{N/2}$, esta última pode ser vista na figura 1, onde $\zeta_8^2 = \zeta_4$

Para $N = 4$, a DFT é representada desse modo:

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

Voltando à 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:

$$X[k] = \sum_{m=0}^{\frac{N}{2}-1} x[2m] \zeta_N^{2mk} + \sum_{m=0}^{\frac{N}{2}-1} x[2m+1] \zeta_N^{(2m+1)k}$$

$$\text{Usando } \zeta_N^2 = \zeta_{N/2} : \quad \zeta_N^{2mk} = (\zeta_N^2)^{mk} = \zeta_{N/2}^{mk}$$

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

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*):

$$X[k] = E[k] + \zeta_N^k O[k] \quad (1)$$

$$X[k + N/2] = E[k] - \zeta_N^k O[k] \quad (2)$$

como pode ser visto na imagem 2

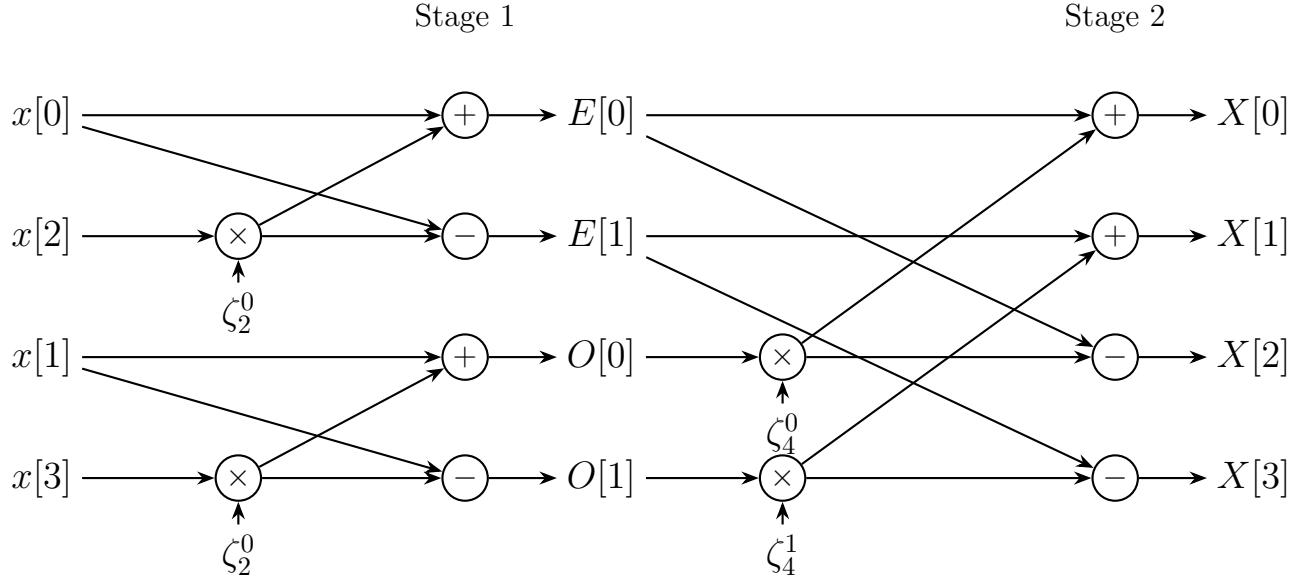


Figure 2: butterfly radix-2

A implementação em pseudocódigo no algoritmo 1

Algorithm 1 Cooley–Tukey FFT

Input: $x \in \mathbb{C}^N$, $N = 2^m$

Output: $X = \text{DFT}(x)$

```

1: if  $N = 1$  then
2:   return  $x$ 
3: end if
4:  $x_e \leftarrow$  elementos pares de  $x$ 
5:  $x_o \leftarrow$  elementos ímpares de  $x$ 
6:  $X_e \leftarrow \text{FFT}(x_e)$ 
7:  $X_o \leftarrow \text{FFT}(x_o)$ 
8: for  $k = 0$  to  $N/2 - 1$  do
9:    $X[k] \leftarrow X_e[k] + \zeta_N^k X_o[k]$ 
10:   $X[k + N/2] \leftarrow X_e[k] - \zeta_N^k X_o[k]$ 
11: end for

```

Como a IDFT é praticamente idêntica à DFT, podemos usar o mesmo algoritmo da FFT para calcular a IFFT

Algorithm 2 IFFT via FFT

Input: $X \in \mathbb{C}^N$ **Output:** $x = \text{IFFT}(X)$

- 1: $Y \leftarrow \overline{X}$ ▷ i.e. o conjugado de X
 - 2: $Z \leftarrow \text{FFT}(Y)$
 - 3: $x \leftarrow \frac{1}{N} \cdot \overline{Z}$
 - 4: **return** x
-

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

6 Problemas da FFT

Um dos problemas da FFT é que ela trabalha com ponto flutuante, o que, para computadores, é um grande problema que pode causar erros de arredondamento ou limitação nos cálculos. Além disso, na convolução linear, o polinômio dobra de tamanho a cada convolução, o que rapidamente torna-se um problema tanto computacional quanto de armazenamento.

Solução: utilizar uma transformada que utiliza apenas números exatos e realiza uma convolução (nega)cíclica

7 Number Theoretic Transform (NTT)

7.1 Fundamentos

As propriedades que usamos na FFT — em especial a existência de uma raiz N -ésima da unidade ζ_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 DFT 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 ζ_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:

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

Assim, o grupo μ_N , gerado pelas potências de ζ_N , é uma realização “multiplicativa” do grupo cíclico $\mathbb{Z}/N\mathbb{Z}$: o isomorfismo $[k] \mapsto \zeta_N^k$ mostra que somar módulo N corresponde exatamente

a multiplicar raízes N -ésimas da unidade. Portanto, estamos apenas vendo o mesmo grupo abstrato em duas roupagens diferentes, uma aditiva e outra multiplicativa.

7.1.1 Raiz Primitiva e Estrutura Negacíclica

Diferente da DFT complexa, onde raízes da unidade sempre existem para qualquer N , a NTT exige que o corpo finito \mathbb{Z}_p , isto é, os inteiros $(\text{mod } p)$, sendo p um primo, suporte a ordem da transformada.

Como p é primo, o conjunto $\mathbb{Z}/p\mathbb{Z}$ forma um corpo, e portanto seus elementos não nulos formam um grupo multiplicativo

$$(\mathbb{Z}/p\mathbb{Z})^\times$$

de cardinalidade $p - 1$. Além disso, esse grupo é cíclico (ver Apêndice 11.4), isto é, existe um gerador g tal que todo elemento não nulo pode ser escrito como uma potência de g .

Nesse contexto, dizer que existe uma raiz $2N$ -ésima primitiva da unidade em $\mathbb{Z}/p\mathbb{Z}$ significa exatamente dizer que existe um elemento $\psi \in (\mathbb{Z}/p\mathbb{Z})^\times$ com ordem $\text{ord}(\psi) = 2N$ tal que $\psi^{2N} \equiv 1 \pmod{p}$ e nenhuma potência menor vale 1.

Como em qualquer grupo finito a ordem de um elemento divide a ordem do grupo, necessariamente deve valer

$$2N \mid |(\mathbb{Z}/p\mathbb{Z})^\times| = p - 1.$$

Reciprocamente, como $(\mathbb{Z}/p\mathbb{Z})^\times$ é cíclico, para todo divisor d de $p - 1$ existe um elemento de ordem exatamente d ; em particular, se $2N \mid (p - 1)$ então existe ψ com $\text{ord}(\psi) = 2N$.

Concluímos, portanto, que existe uma raiz primitiva $2N$ -ésima da unidade em $\mathbb{Z}/p\mathbb{Z}$ se, e somente se, $2N \mid (p - 1)$ (equivalentemente $p \equiv 1 \pmod{2N}$).

Dados esses critérios de existência, trataremos da estrutura algébrica da transformada. A NTT Negacíclica é definida no anel quociente

$$R = \frac{\mathbb{Z}_p[x]}{(x^N + 1)}$$

ou seja, há uma composição de operações modulares. (A imagem 3 ilustra a sequência das duas operações visualmente.)

Ela pode ser vista como a avaliação do polinômio nas raízes da equação $x^N + 1 = 0$, que correspondem às potências ímpares de ψ . A transformada é definida por:

$$X[k] = \sum_{n=0}^{N-1} x[n]\psi^{(2k+1)n} \pmod{p}$$

Na forma matricial:

$$\mathbf{X} = \mathbf{W}_N \mathbf{x}, \quad \mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{bmatrix}$$

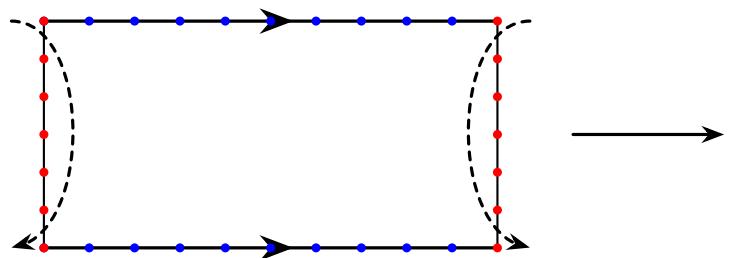
Neste caso, a matriz de transformação \mathbf{W}_N difere da versão cíclica padrão, pois seus coeficientes seguem a estrutura das raízes negacíclicas:

$$\mathbf{W}_N = \begin{bmatrix} 1 & \psi^1 & \psi^2 & \dots & \psi^{(N-1)} \\ 1 & \psi^3 & \psi^6 & \dots & \psi^{3(N-1)} \\ 1 & \psi^5 & \psi^{10} & \dots & \psi^{5(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \psi^{(2k+1)} & \psi^{(2k+1)2} & \dots & \psi^{(2k+1)(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \psi^{(2N-1)} & \psi^{(2N-1)2} & \dots & \psi^{(2N-1)(N-1)} \end{bmatrix}$$

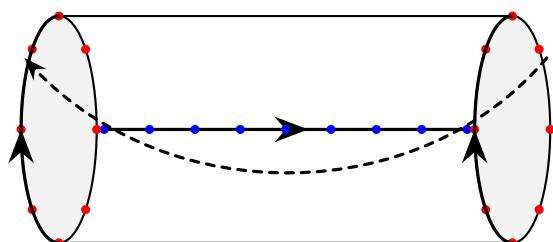
De forma geral, o termo na linha k e coluna n da matriz é dado por:

$$(\mathbf{W}_N)_{k,n} = \psi^{(2k+1)n} \pmod{p}, \quad 0 \leq k, n \leq N-1$$

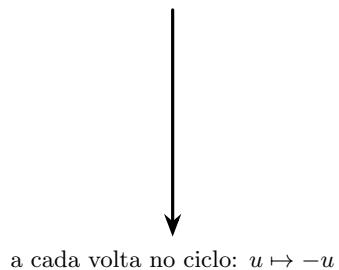
1. Colar lados horizontais (módulo \mathbb{Z}_p)



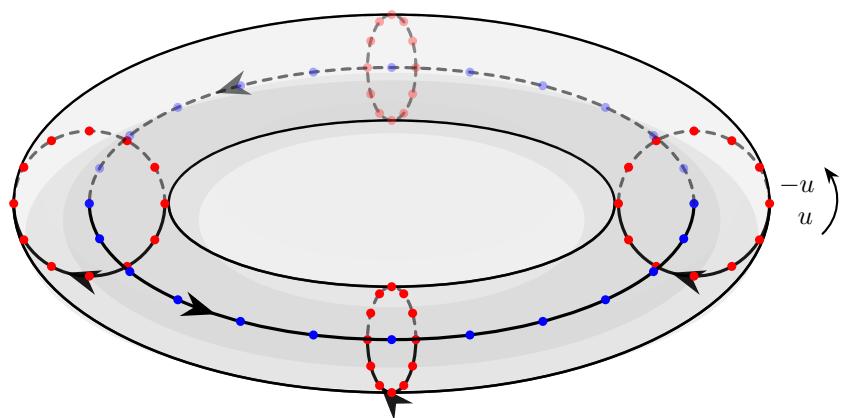
Obtém-se um cilindro



2. Colar lados verticais (módulo $(x^N + 1)$)



a cada volta no ciclo: $u \mapsto -u$



3. Obtém-se um Toro Discreto

Figure 3: representação das transformações

7.2 O Isomorfismo via Teorema Chinês dos Restos (CRT)

A fundamentação algébrica da NTT reside na estrutura do anel quociente $\mathbb{Z}_p[x]/(x^N + 1)$. Já que $x^N + 1$ fatora-se em N fatores lineares distintos e coprimos,

$$x^N + 1 = \prod_{k=0}^{N-1} (x - \psi^{2k+1}),$$

Homomorfismo de avaliação.

Para cada $\alpha \in \mathbb{Z}_p$, considere o mapa de avaliação

$$\text{ev}_\alpha : \mathbb{Z}_p[x] \rightarrow \mathbb{Z}_p, \quad \text{ev}_\alpha(a(x)) = a(\alpha).$$

Esse mapa é um *homomorfismo de anéis* (preserva soma e produto):

$$\text{ev}_\alpha(a + b) = \text{ev}_\alpha(a) + \text{ev}_\alpha(b), \quad \text{ev}_\alpha(ab) = \text{ev}_\alpha(a)\text{ev}_\alpha(b).$$

Além disso, seu núcleo é exatamente o ideal gerado por $(x - \alpha)$:

$$\ker(\text{ev}_\alpha) = (x - \alpha).$$

De fato, pela Divisão Euclidiana, qualquer $a(x)$ pode ser escrito como

$$a(x) = q(x)(x - \alpha) + r$$

com $r \in \mathbb{Z}_p$, e avaliando em $x = \alpha$ obtemos $a(\alpha) = r$, uma vez que $(x - \alpha)$ zera. Logo $a(\alpha) = 0$ se e somente se $(x - \alpha) \mid a(x)$. Pelo **Primeiro Teorema do Isomorfismo** para anéis,

$$\frac{\mathbb{Z}_p[x]}{(x - \alpha)} \cong \text{Im}(\text{ev}_\alpha) = \mathbb{Z}_p, \quad [a(x)] \longmapsto a(\alpha).$$

O isomorfismo global do CRT como produto de avaliações.

Como os fatores $(x - \psi^{2k+1})$ são coprimos dois a dois, o CRT fornece um isomorfismo de anéis

$$\Phi : \frac{\mathbb{Z}_p[x]}{(x^N + 1)} \xrightarrow{\cong} \prod_{k=0}^{N-1} \frac{\mathbb{Z}_p[x]}{(x - \psi^{2k+1})} \cong \underbrace{\mathbb{Z}_p \times \cdots \times \mathbb{Z}_p}_{N \text{ vezes}}.$$

E, crucialmente, Φ é construído a partir dos homomorfismos de avaliação: a k -ésima componente é exatamente o homomorfismo $\text{ev}_{\psi^{2k+1}}$ (passando ao quociente). Assim, para $[a(x)] \in \mathbb{Z}_p[x]/(x^N + 1)$,

$$\Phi([a(x)]) = (a(\psi^1), a(\psi^3), \dots, a(\psi^{2N-1})) \in \mathbb{Z}_p^N.$$

Multiplicação vira produto ponto a ponto. Como cada $\text{ev}_{\psi^{2k+1}}$ é homomorfismo e Φ é isomorfismo de anéis,

$$\Phi([a(x)][b(x)]) = \Phi([a(x)]) \cdot \Phi([b(x)])$$

onde o produto do lado direito é componente a componente. Em particular, se $C = \Phi([a(x)b(x)])$, então

$$C_k = (ab)(\psi^{2k+1}) = a(\psi^{2k+1})b(\psi^{2k+1}).$$

Isso explica formalmente por que a multiplicação no quociente $\mathbb{Z}_p[x]/(x^N + 1)$ (convolução negacíclica dos coeficientes) se traduz em um produto de Hadamard no domínio transformado.

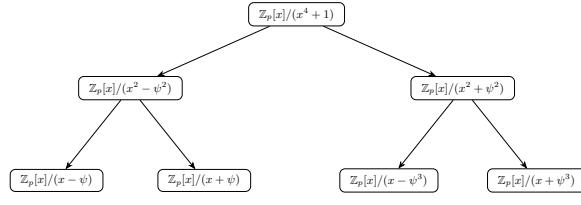


Figure 4: CRT da NTT Negacíclica (Exemplo N=4)

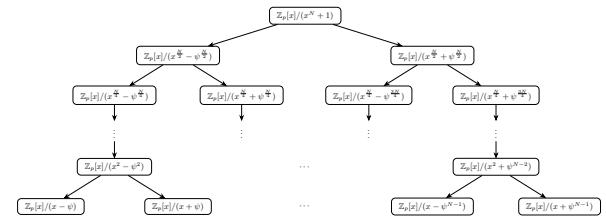


Figure 5: CRT da NTT Negacíclica (Geral)

7.3 Transformada Numérica Inversa (INTT)

Como o Teorema Chinês dos Restos garante que a aplicação da NTT é um isomorfismo bijutor entre o anel de polinômios $\mathbb{Z}_p[x]/(x^N + 1)$ e o domínio da frequência, existe uma transformação inversa única capaz de recuperar os coeficientes originais.

Denotamos a inversa multiplicativa de N no corpo \mathbb{Z}_p por N^{-1} , tal que $N \cdot N^{-1} \equiv 1 \pmod{p}$. A NTT Negacíclica Inversa (INTT) é definida formalmente por:

$$x[n] = N^{-1} \sum_{k=0}^{N-1} X[k] \psi^{-(2k+1)n} \pmod{p}$$

Note que o termo $\psi^{-(2k+1)n}$ refere-se à potência do inverso multiplicativo da raiz.

Representação Matricial

Na forma matricial, a operação de inversão corresponde à resolução do sistema linear $\mathbf{X} = \mathbf{W}_N \mathbf{x}$. A solução é dada por:

$$\mathbf{x} = \mathbf{W}_N^{-1} \mathbf{X}$$

Onde a matriz inversa \mathbf{W}_N^{-1} é definida como o inverso modular das potências ímpares de ψ , escalada pelo fator N^{-1} :

$$\mathbf{W}_N^{-1} = N^{-1} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ \psi^{-1} & \psi^{-3} & \psi^{-5} & \dots & \psi^{-(2N-1)} \\ \psi^{-2} & \psi^{-6} & \psi^{-10} & \dots & \psi^{-2(2N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \psi^{-n} & \psi^{-3n} & \psi^{-5n} & \dots & \psi^{-n(2k+1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \psi^{-(N-1)} & \psi^{-3(N-1)} & \psi^{-5(N-1)} & \dots & \psi^{-(N-1)(2N-1)} \end{bmatrix}$$

De modo geral, o termo na linha n e coluna k da matriz inversa é:

$$(\mathbf{W}_N^{-1})_{n,k} = N^{-1} \psi^{-(2k+1)n} \pmod{p}$$

7.4 A NTT como uma FFT: Decomposição Radix-2

A eficiência da NTT negacíclica baseia-se na estratégia *dividir para conquistar*, análoga à FFT clássica (algoritmo de Cooley-Tukey). Em vez de avaliar diretamente um polinômio $A(x)$ em N pontos, decomponemos o problema em dois subproblemas de tamanho $N/2$ através da separação dos coeficientes em índices pares e ímpares.

1. Definições e Propriedades

Seja $N = 2^m$ e p um primo tal que exista uma raiz primitiva $\psi \in \mathbb{Z}_p^*$ de ordem $2N$. As propriedades fundamentais para a aritmética negacíclica são:

$$\psi^{2N} \equiv 1 \pmod{p} \quad \text{e} \quad \psi^N \equiv -1 \pmod{p}.$$

A NTT negacíclica consiste na avaliação do polinômio $A(x)$ nas N raízes ímpares da unidade:

$$x_k = \psi^{2k+1}, \quad \text{para } k = 0, 1, \dots, N-1.$$

2. Decomposição do Polinômio (Decimation-in-Time)

Podemos reescrever $A(x)$ separando os termos com potências pares e ímpares de x :

$$\begin{aligned} A(x) &= \sum_{i=0}^{N-1} a[i]x^i \\ &= \sum_{i=0}^{N/2-1} a[2i]x^{2i} + \sum_{i=0}^{N/2-1} a[2i+1]x^{2i+1} \\ &= \sum_{i=0}^{N/2-1} a[2i](x^2)^i + x \cdot \sum_{i=0}^{N/2-1} a[2i+1](x^2)^i. \end{aligned}$$

Definindo os polinômios auxiliares $A_{\text{par}}(y)$ e $A_{\text{ímpar}}(y)$ como as partes pares e ímpares, respectivamente, obtemos a relação recursiva:

$$A(x) = A_{\text{par}}(x^2) + x \cdot A_{\text{ímpar}}(x^2).$$

3. A Recursão e o Fator ψ^2

Ao avaliarmos essa expressão nos pontos $x_k = \psi^{2k+1}$, observamos o comportamento do termo quadrático:

$$x_k^2 = (\psi^{2k+1})^2 = \psi^{4k+2} = (\psi^2)^{2k+1}.$$

Isso revela que, para os subproblemas de tamanho $N/2$, a base da transformação torna-se ψ^2 . Como a ordem de ψ é $2N$, a ordem de ψ^2 é N , satisfazendo o requisito de raiz primitiva para o subgrupo de tamanho $N/2$.

Logo, os valores retornados pelas chamadas recursivas são:

$$Y_{\text{par}}[k] = A_{\text{par}}(x_k^2) \quad \text{e} \quad Y_{\text{ímpar}}[k] = A_{\text{ímpar}}(x_k^2).$$

4. A Borboleta (Butterfly) Negacíclica

Para recombinar os resultados e obter $Y[k] = A(x_k)$, exploramos a simetria das raízes. Lembre que, após a decomposição $A(x) = A_{\text{par}}(x^2) + x A_{\text{impar}}(x^2)$, avaliamos em

$$x_k = \psi^{2k+1}, \quad k = 0, 1, \dots, N-1,$$

de modo que

$$Y_{\text{par}}[k] = A_{\text{par}}(x_k^2), \quad Y_{\text{impar}}[k] = A_{\text{impar}}(x_k^2), \quad (0 \leq k < N/2).$$

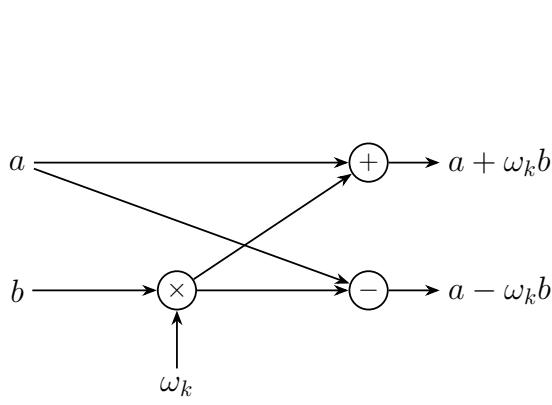
Para a primeira metade ($0 \leq k < N/2$), a recombinação é direta. Para a segunda metade, usamos a simetria negacíclica das raízes ímpares:

$$x_{k+N/2} = \psi^{2(k+N/2)+1} = \psi^{2k+1} \cdot \psi^N = -\psi^{2k+1} = -x_k.$$

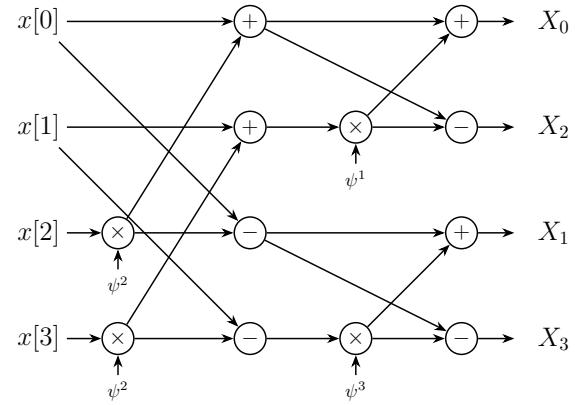
Assim, o termo $x \cdot A_{\text{impar}}(x^2)$ apenas troca de sinal na segunda metade. As equações da borboleta tornam-se:

$$\boxed{\begin{aligned} Y[k] &= Y_{\text{par}}[k] + x_k \cdot Y_{\text{impar}}[k] \pmod{p} \\ Y[k+N/2] &= Y_{\text{par}}[k] - x_k \cdot Y_{\text{impar}}[k] \pmod{p} \end{aligned}}$$

Nota de Implementação: Na prática, os valores $x_k = \psi^{2k+1}$ não são recalculados por exponenciação. Percorremos as raízes ímpares multiplicando por ψ^2 a cada passo: inicialize $x \leftarrow \psi$ e atualize $x \leftarrow x \cdot \psi^2$, gerando $(\psi^1, \psi^3, \psi^5, \dots)$.



(a) Butterfly NTT



(b) Butterfly NTT 4

7.5 A INTT via Gentleman–Sande (GS): Reconstrução

Enquanto a transformação direta (NTT) utiliza *dividir para conquistar* separando coeficientes pares e ímpares, a transformação inversa (INTT) percorre o caminho oposto: combina resultados de subproblemas menores para recuperar o vetor original. O esquema de Gentleman–Sande pode ser visto como a versão “transposta” (em ordem de operações) da decomposição de Cooley–Tukey.

1. Paridade e o sistema linear local

Na NTT negacíclica por partição par/ímpar, escrevemos

$$A(x) = A_e(x^2) + x A_o(x^2), \quad A_e(t) = \sum_{i=0}^{N/2-1} a[2i]t^i, \quad A_o(t) = \sum_{i=0}^{N/2-1} a[2i+1]t^i.$$

A avaliação ocorre nos pontos ímpares $x_k = \psi^{2k+1}$, $k = 0, \dots, N-1$. Denotando por $Y_e[k] = A_e(x_k^2)$ e $Y_o[k] = A_o(x_k^2)$ (retornados pela recursão), a combinação (borboleta direta) é:

$$\begin{cases} Y[k] = Y_e[k] + x_k Y_o[k], \\ Y[k+N/2] = Y_e[k] - x_k Y_o[k], \end{cases} \quad 0 \leq k < N/2.$$

Na etapa inversa, conhecemos $Y[k]$ e $Y[k+N/2]$ e queremos recuperar $Y_e[k]$ e $Y_o[k]$.

2. A lógica da recuperação (inversão do 2×2)

Somando e subtraindo as equações acima, obtemos:

$$Y[k] + Y[k+N/2] = 2Y_e[k], \quad Y[k] - Y[k+N/2] = 2x_k Y_o[k].$$

Como p é primo ímpar, 2 é invertível em \mathbb{Z}_p e podemos escrever:

$$\boxed{\begin{aligned} Y_e[k] &= 2^{-1}(Y[k] + Y[k+N/2]) \pmod{p}, \\ Y_o[k] &= 2^{-1}(Y[k] - Y[k+N/2]) x_k^{-1} \pmod{p}. \end{aligned}}$$

Assim, a borboleta inversa recupera os dois vetores de entrada do nível, que então alimentam as recursões de tamanho $N/2$.

3. Por que a recursão usa ψ^2 novamente

Na ida, ao avaliar em $x_k = \psi^{2k+1}$, temos

$$x_k^2 = (\psi^{2k+1})^2 = \psi^{4k+2} = (\psi^2)^{2k+1},$$

ou seja, os subproblemas são novamente NTTs negacíclicas em $N/2$ pontos ímpares, mas agora com parâmetro ψ^2 . Na volta, o mesmo vale: após aplicar a borboleta inversa no nível corrente, chamamos recursivamente a INTT nos blocos correspondentes usando ψ^{-2} .

4. Ordem das operações: CT vs GS

A diferença estrutural entre as duas direções pode ser resumida assim:

1. Na **Ida (CT / DIT)**, primeiro resolvemos recursivamente e *depois* combinamos com o fator x_k .
2. Na **Volta (GS)**, primeiro desfazemos a combinação local (somar/subtrair e multiplicar por x_k^{-1}) e *depois* prosseguimos recursivamente.

Nota de implementação: o fator 2^{-1} pode ser acumulado ao longo das camadas e aplicado como um único fator global N^{-1} ao final, reduzindo o número de multiplicações escalares. Além disso, para evitar criar uma variável dedicada ao *twiddle factor*, usa-se diretamente x_k^{-1} (ou o valor já pré-computado em uma tabela de potências) no ponto em que a multiplicação é necessária.

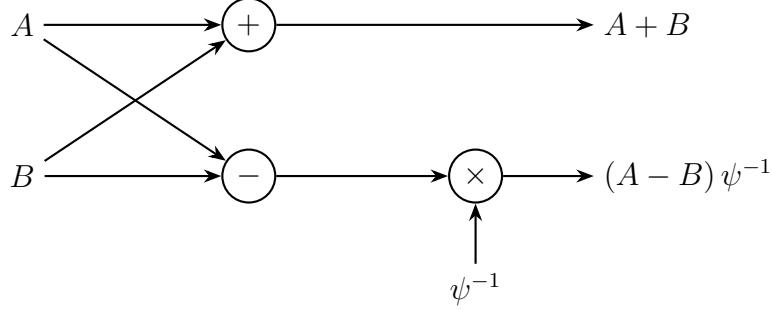


Figure 7: Butterfly INTT

A implementação da NTT negacíclica está no pseudocódigo a seguir

Algorithm 3: NTT Negacíclica (CT)

Input: $a \in \mathbb{Z}_p^N$ (Coeficientes de entrada)
Input: $N = 2^m$ e Módulo p
Input: ψ : Raiz primitiva $2N$ -ésima da unidade
1: $len \leftarrow N/2$
2: **for** $l \leftarrow m - 1$ **down to** 0 **do**
3: $num_gr \leftarrow 2^{m-1-l}$
4: **for** $i \leftarrow 0$ **to** $num_gr - 1$ **do**
5: $\zeta \leftarrow \psi^{\text{brv}_m(num_gr+i)}$
6: **for** $j \leftarrow i \cdot 2^{l+1}$ **to** ... + $2^l - 1$ **do**
7: $t_0 \leftarrow a[j]$
8: $t_1 \leftarrow \zeta \cdot a[j + 2^l] \pmod p$
9: $a[j] \leftarrow t_0 + t_1 \pmod p$
10: $a[j + 2^l] \leftarrow t_0 - t_1 \pmod p$
11:
12: **end for**
13:
14: **end for**
15:
16: **end for**
17: **return** a (*bit-reversed*)

Algorithm 4: INTT (GS)

Input: \hat{a} (Vetor em ordem *bit-reversed*)
Input: $N = 2^m$ e Módulo p
Input: ψ : Raiz primitiva $2N$ -ésima da unidade
1: **for** $l \leftarrow 0$ **to** $m - 1$ **do**
2: $num_gr \leftarrow 2^{m-1-l}$
3: **for** $i \leftarrow 0$ **to** $num_gr - 1$ **do**
4: $\zeta \leftarrow \psi^{-\text{brv}_m(num_gr+i)}$
5: **for** $j \leftarrow i \cdot 2^{l+1}$ **to** ... + $2^l - 1$ **do**
6: $t_0 \leftarrow a[j]$
7: $t_1 \leftarrow a[j + 2^l]$
8: $a[j] \leftarrow t_0 + t_1 \pmod p$
9: $a[j + 2^l] \leftarrow (t_0 - t_1)\zeta \pmod p$
10: **end for**
11: **end for**
12: **end for**
13: $invN \leftarrow N^{-1} \pmod p$
14: **for** $i \leftarrow 0$ **to** $N - 1$ **do**
15: $a[i] \leftarrow a[i] \cdot invN$
16: **end for**
17: **return** a (Ordem normal)

8 Testes comparativos

A comparação computacional dessas ferramentas pode ser vista na tabela a seguir, onde calculou-se o maior número de Fibonacci obtido em menos de um segundo. Todos os testes foram executados em single-core.

Nos algoritmos da FFT e da NTT, o n -ésimo número de Fibonacci foi calculado pela seguinte relação:

$$\begin{bmatrix} F_{n+1} \\ F_n \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^n \cdot \begin{bmatrix} F_1 \\ F_0 \end{bmatrix}$$

sendo a multiplicação que ocorre tanto na exponenciação quanto no produto interno foram otimizadas com o devido método.

Além disso, como o foco foi apenas em comparar a velocidade dos métodos, utilizou-se de scripts para a geração de macros com os valores das raízes primitivas, raízes da unidade, entre outras constantes; para mais informações, veja o repositório.

(GMP é uma biblioteca otimizada para operações com inteiros extremamente grandes)

Setup:

CPU: 13th Gen Intel(R) Core(TM) i7-1365U (12) @ 5.20 GHz

Table 1: Comparaçāo de Desempenho dos Algoritmos

Algoritmo	Quantidade n calculados (%)
Algoritmo Naive	44
Algoritmo Iterado	566.053
Algoritmo FFT	3.145.816
Algoritmo NTT	24.178.839
Algoritmo GMP	238.961.323

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

9 NTT incompleta e a restrição de parâmetros no caso negacíclico

Na multiplicação negacíclica de polinômios no anel

$$\mathbb{Z}_p[x]/(x^N + 1),$$

com N potência de dois, a utilização de uma **NTT completa** exige a existência de uma raiz primitiva de ordem $2N$ módulo p . Quando p é primo, essa condição é equivalente a

$$p \equiv 1 \pmod{2N}.$$

Em aplicações criptográficas baseadas em reticulados, em especial em esquemas de Fully Homomorphic Encryption, o valor de N é tipicamente muito grande. Como consequência, a condição acima impõe fortes restrições sobre a escolha do primo p , frequentemente forçando o uso de módulos grandes ou pouco flexíveis, o que impacta tanto a eficiência quanto o ajuste fino de parâmetros de segurança.

Uma forma natural de relaxar essa restrição é empregar a chamada **NTT ℓ -incompleta**. A ideia consiste em interromper o algoritmo da NTT antes de executar todos os $\log_2 N$ estágios do esquema radix-2. Mais precisamente, ao parar após $\log_2 N - \ell$ estágios, a existência

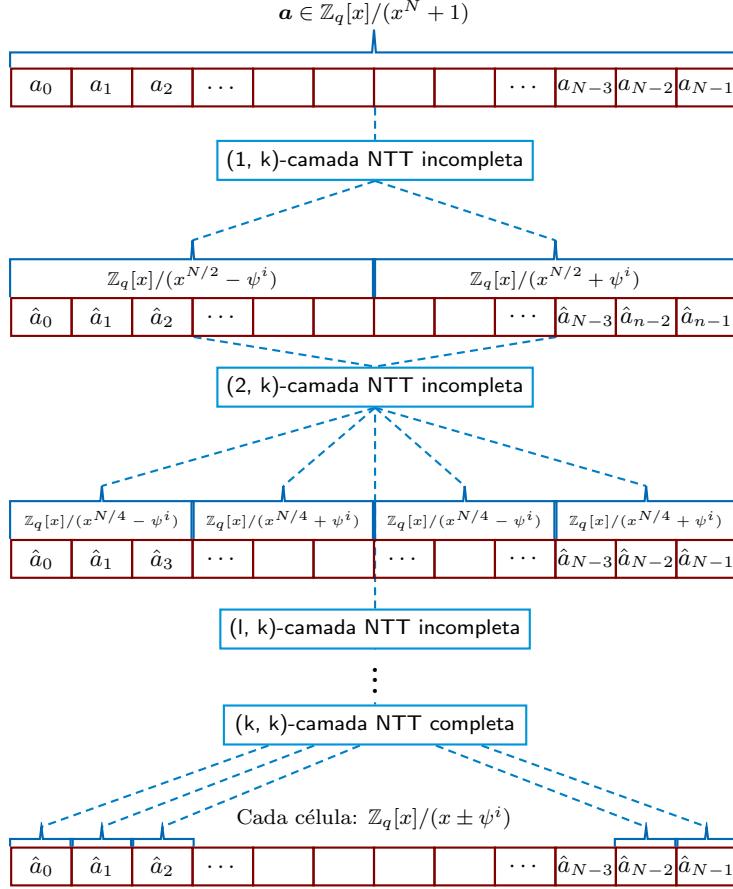


Figure 8: árvore de fracionamento da NTT

da transformada passa a requerer apenas uma raiz da unidade de ordem $2N/2^\ell$ em \mathbb{Z}_p , o que resulta na condição mais fraca

$$p \equiv 1 \pmod{2N/2^\ell}.$$

Dessa forma, o conjunto de primos admissíveis torna-se significativamente maior, permitindo escolhas de parâmetros mais flexíveis.

Do ponto de vista algébrico, a NTT ℓ -incompleta não avalia o polinômio em todas as N raízes da unidade, mas o mapeia para um vetor de $N/2^\ell$ polinômios de menor grau, cada um pertencente a um quociente do tipo

$$\mathbb{Z}_p[x]/(x^{2^\ell} - \psi_i),$$

onde ψ_i são potências apropriadas da raiz disponível. A multiplicação passa então a ser realizada componente a componente nesses anéis menores, seguida de uma transformada inversa incompleta.

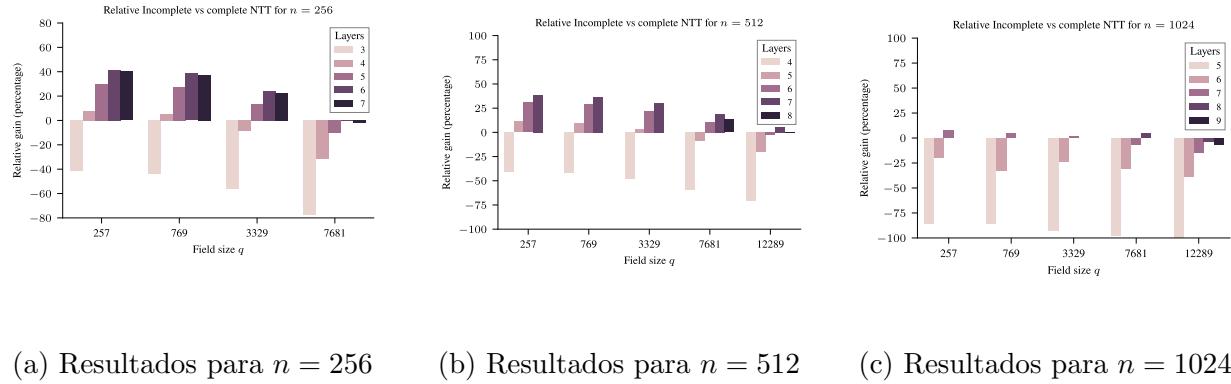
Para demonstrar o impacto prático dessa flexibilidade algébrica, analisa-se o cenário de *bootstrapping* amortizado em FHE. Como a complexidade das operações aritméticas depende diretamente do tamanho do módulo p , a capacidade de selecionar primos significativamente menores — viabilizada pela incompletude da NTT — traduz-se em ganhos de desempenho

mensuráveis. A Tabela 2 quantifica essa melhoria, contrastando os custos computacionais e a escolha ótima de parâmetros para diferentes níveis de segurança em comparação com a restrição rígida imposta pela NTT completa.

Table 2: Comparação de Desempenho: Bootstrapping com NTT Incompleta vs. Completa

Nível de Segurança (λ)	Grau Ótimo de Incompletude (ℓ)	Módulo Ótimo (p , Incompleta)	Módulo NTT Completa ($\ell = 10$)	Ganho de Desempenho
128	6	3.329	12.289	33%
192	8	7.681	12.289	29%
256	8	12.289	12.289	6%
512	7	18.433	40.961	42%

Nota: Para $\lambda = 256$, o módulo ótimo permanece o mesmo ($q = 12.289$) em ambos os cenários, resultando em um ganho de desempenho menor comparado aos outros níveis onde a redução do módulo é significativa.



(a) Resultados para $n = 256$ (b) Resultados para $n = 512$ (c) Resultados para $n = 1024$

Figure 9: Imagens retiradas do artigo [3]

Os resultados experimentais apresentados na Figura 9 sugerem que a adoção de valores maiores de n em esquemas FHE (por exemplo, $n \geq 1024$) pode limitar a melhoria de desempenho obtida pela incompletude da NTT. Observa-se que o número de instâncias onde as configurações de NTT incompleta apresentam melhor desempenho diminui à medida que n aumenta. Frequentemente, as restrições de segurança para esses casos exigem a utilização de módulos primos maiores (como $q = 12289$), cenário no qual a NTT completa acaba por ser a escolha ótima, anulando os ganhos que seriam obtidos através do uso de módulos menores permitidos pela técnica incompleta.

10 Agradecimentos

Gostaria de agradecer ao professor doutor *Thales Paiva* pela chance de apresentar o seminário e pelos ensinamentos, gostaria de agradecer também o professor doutor *João Fernando da*

Cunha Nariyoshi por esclarecer algumas dúvidas sobre a parte da matemática pura. Foi realmente divertido pesquisar, escrever, desenhar e animar para esse seminário.

11 Apêndice

11.1 DFT como transformação linear

Seja $\zeta = e^{-2\pi i/N}$ uma raiz primitiva N -ésima da unidade. Definimos a Transformada Discreta de Fourier (DFT) como a operação que mapeia um sinal discreto $x[n]$ de comprimento N em uma sequência $X[k]$ no domínio da frequência, dada por:

$$X[k] = \sum_{n=0}^{N-1} x[n] \zeta^{kn}, \quad k = 0, 1, \dots, N-1.$$

Teorema 2. A Transformada Discreta de Fourier é uma transformação linear. Ou Seja, para quaisquer sinais $x[n], y[n]$ e escalares $a, b \in \mathbb{C}$, vale:

$$\text{DFT}(ax + by) = a \text{DFT}(x) + b \text{DFT}(y).$$

Demonstração. Sejam $x[n]$ e $y[n]$ dois sinais e $a, b \in \mathbb{C}$ constantes. Definimos a combinação linear $z[n] = a x[n] + b y[n]$. Aplicando a definição da DFT ao sinal $z[n]$, obtemos $Z[k]$:

$$Z[k] = \sum_{n=0}^{N-1} z[n] \zeta^{kn} = \sum_{n=0}^{N-1} (a x[n] + b y[n]) \zeta^{kn}.$$

Pela propriedade distributiva do somatório e pela linearidade da soma em \mathbb{C} :

$$Z[k] = \sum_{n=0}^{N-1} a x[n] \zeta^{kn} + \sum_{n=0}^{N-1} b y[n] \zeta^{kn}.$$

Como os escalares a e b não dependem do índice da soma n , podemos fatorá-los para fora dos somatórios:

$$Z[k] = a \sum_{n=0}^{N-1} x[n] \zeta^{kn} + b \sum_{n=0}^{N-1} y[n] \zeta^{kn}.$$

Reconhecendo as expressões dentro dos somatórios como as definições de $X[k]$ e $Y[k]$, respectivamente, concluímos que:

$$Z[k] = a X[k] + b Y[k].$$

Portanto, a relação de linearidade é satisfeita para cada componente k , o que prova que a DFT é uma transformação linear. \square

11.2 Exemplo de convolução negacíclica

Sejam dois sinais x e y de comprimento $N = 3$:

$$x = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}, \quad y = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

Neste contexto, a frequência fundamental não é a raiz da unidade, mas sim a raiz de $x^N + 1 = 0$. Seja ψ tal que $\psi^N = -1$. Para $N = 3$, temos $\psi^3 = -1$.

Método 1: Convolução no Tempo (Matriz Negacirculante)

A convolução negacíclica $z = x *_{\text{neg}} y$ equivale à multiplicação de uma matriz negacirculante H_x pelo vetor y . Note que os elementos que “dão a volta” (acima da diagonal principal) têm sinal invertido:

$$H_x = \begin{bmatrix} x[0] & -x[2] & -x[1] \\ x[1] & x[0] & -x[2] \\ x[2] & x[1] & x[0] \end{bmatrix} = \begin{bmatrix} 1 & 0 & -2 \\ 2 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}$$

Calculando $z = H_x y$:

$$z = \begin{bmatrix} 1 & 0 & -2 \\ 2 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1(1) + 0 - 2(1) \\ 2(1) + 0 + 0 \\ 0 + 0 + 1(1) \end{bmatrix} = \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix}$$

Método 2: Diagonalização (Domínio da Frequência)

A matriz negacirculante H_x é diagonalizada por uma variante da DFT que utiliza potências ímpares de ψ (raízes de $x^N + 1$). Matematicamente:

$$H_x = \mathcal{F}^{-1} \Lambda_x \mathcal{F}$$

onde Λ_x contém os autovalores de H_x , que correspondem à avaliação do polinômio $x(z)$ nas raízes de $z^N = -1$.

Para verificar os autovalores (o espectro de x), calculamos o polinômio característico de H_x :

$$\det(H_x - \lambda I) = 0 \implies \det \begin{bmatrix} 1 - \lambda & 0 & -2 \\ 2 & 1 - \lambda & 0 \\ 0 & 2 & 1 - \lambda \end{bmatrix} = 0$$

Expandindo o determinante (regra de Sarrus):

$$(1 - \lambda)^3 - (2)(2)(2) = 0 \implies (1 - \lambda)^3 = 8$$

As raízes para $(1 - \lambda)$ são as raízes cúbicas de 8. Sabemos que $8 = 8 \cdot 1$, mas no contexto complexo as raízes são $2, 2\zeta_3$ e $2\zeta_3^2$. Logo:

$$\begin{aligned} 1 - \lambda_0 &= 2 \implies \lambda_0 = -1 \\ 1 - \lambda_1 &= 2\zeta_3 \implies \lambda_1 = 1 - 2\zeta_3 \\ 1 - \lambda_2 &= 2\zeta_3^2 \implies \lambda_2 = 1 - 2\zeta_3^2 \end{aligned}$$

Estes valores λ_k correspondem à Transformada Negacíclica (NTT sobre $x^N + 1$) do vetor x , permitindo calcular a convolução via produto ponto a ponto.

Verificação do Autovetor

Verificamos agora se o autovetor v_1 da base Negacíclica (coluna de \mathcal{F}^{-1} associada à raiz ψ), dado por $v_1 = [1, \psi^{-1}, \psi^{-2}]^T = [1, -\psi^2, -\psi]^T$, satisfaz $H_x v_1 = \lambda_1 v_1$.

Lado esquerdo ($H_x v_1$):

$$\begin{bmatrix} 1 & 0 & -2 \\ 2 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -\psi^2 \\ -\psi \end{bmatrix} = \begin{bmatrix} 1 + 2\psi \\ 2 - \psi^2 \\ -2\psi^2 - \psi \end{bmatrix}$$

Lado direito ($\lambda_1 v_1$), onde o autovalor é $\lambda_1 = x(\psi) = 1 + 2\psi$:

$$(1 + 2\psi) \begin{bmatrix} 1 \\ -\psi^2 \\ -\psi \end{bmatrix} = \begin{bmatrix} 1 + 2\psi \\ -\psi^2 - 2\psi^3 \\ -\psi - 2\psi^2 \end{bmatrix}$$

Utilizando a propriedade $\psi^3 = -1$, simplificamos o termo do meio: $-\psi^2 - 2(-1) = 2 - \psi^2$. A igualdade é satisfeita, confirmando que as raízes de $x^N + 1$ geram a base natural de H_x .

Representação Matricial da Diagonalização

Primeiro, calculamos os vetores transformados $X = \text{NTT}(x)$ e $Y = \text{NTT}(y)$ avaliando os polinômios nas raízes de $z^3 = -1$:

$$X = \begin{bmatrix} x(-1) \\ x(\psi) \\ x(\psi^2) \end{bmatrix} = \begin{bmatrix} -1 \\ 1 + 2\psi \\ 1 + 2\psi^2 \end{bmatrix}, \quad Y = \begin{bmatrix} y(-1) \\ y(\psi) \\ y(\psi^2) \end{bmatrix} = \begin{bmatrix} 2 \\ 1 + \psi^2 \\ 1 + \psi^4 \end{bmatrix}$$

*Nota: $y(\psi) = 1 + 0\psi + 1\psi^2$. Para o terceiro termo, usamos $\psi^4 = -\psi$, logo $1 - \psi$.

Agora, construímos a matriz diagonal $\Lambda_x = \text{diag}(X)$. A operação de convolução no domínio da frequência ($Z = \Lambda_x Y$) torna-se:

$$\mathbf{Z} = \underbrace{\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 + 2\psi & 0 \\ 0 & 0 & 1 + 2\psi^2 \end{bmatrix}}_{\text{Matriz Diagonal } (\Lambda_x)} \begin{bmatrix} 2 \\ 1 + \psi^2 \\ 1 - \psi \end{bmatrix}$$

Executando o produto ponto a ponto:

$$\mathbf{Z} = \begin{bmatrix} -2 \\ (1 + 2\psi)(1 + \psi^2) \\ (1 + 2\psi^2)(1 - \psi) \end{bmatrix} = \begin{bmatrix} -2 \\ 1 + \psi^2 + 2\psi + 2\psi^3 \\ 1 - \psi + 2\psi^2 - 2\psi^3 \end{bmatrix} = \begin{bmatrix} -2 \\ -1 + 2\psi + \psi^2 \\ 3 - \psi + 2\psi^2 \end{bmatrix}$$

*Simplificações usando $\psi^3 = -1$.

Retorno ao Tempo (INTT)

Finalmente, aplicamos a transformada inversa. O vetor resultante z deve coincidir com o cálculo temporal $[-1, 2, 1]^T$. Ao reconstruir o polinômio $z(\omega)$ a partir dos valores em \mathbf{Z} , obtemos os coeficientes:

$$z = \text{INTT}(\mathbf{Z}) = \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix}$$

Isso verifica que:

$$z(\psi) = -1 + 2\psi + \psi^2$$

O que coincide exatamente com o segundo elemento do vetor \mathbf{Z} calculado acima, fechando o ciclo da prova numérica.

11.3 Determinação unívoca do polinômio

Teorema 3 (Unicidade via matriz de Vandermonde). *Sejam x_0, \dots, x_{N-1} escalares dois a dois distintos em um corpo \mathbb{K} (e.g., $\mathbb{R}, \mathbb{C}, \mathbb{F}_p$), e Sejam $y_0, \dots, y_{N-1} \in \mathbb{K}$. Existe um único polinômio $p(x) = a_0 + a_1x + \dots + a_{N-1}x^{N-1} \in \mathbb{K}[x]$ tal que $p(x_i) = y_i$ para todo $i = 0, \dots, N-1$.*

Demonstração. Escreva $p(x) = \sum_{k=0}^{N-1} a_k x^k$. Impor as condições $p(x_i) = y_i$ para $i = 0, \dots, N-1$ gera o sistema linear

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{N-1} \\ 1 & x_1 & x_1^2 & \cdots & x_1^{N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N-1} & x_{N-1}^2 & \cdots & x_{N-1}^{N-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{N-1} \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-1} \end{bmatrix}.$$

Denote essa matriz por V (matriz de Vandermonde, i.e. $V_{i,j} = x_i^{j-1}$ para todo os índices i,j), o vetor de coeficientes por a e o vetor de valores por y ; então o sistema é

$$Va = y.$$

O determinante de Vandermonde é dado por

$$\det(V) = \prod_{0 \leq i < j \leq N-1} (x_j - x_i).$$

Como os x_i são dois a dois distintos, temos $x_j - x_i \neq 0$ para $i \neq j$, logo $\det(V) \neq 0$. Portanto, V é invertível e o sistema $Va = y$ tem **solução única**, dada por

$$a = V^{-1}y.$$

Concluímos que existe um único vetor de coeficientes (a_0, \dots, a_{N-1}) , isto é, um **único** polinômio $p(x)$ de grau $\leq N-1$ que interpola os N pontos. \square

11.4 Estrutura Cíclica de Grupos Multiplicativos de Corpos Finitos

Neste apêndice, justificamos a afirmação de que o grupo multiplicativo $(\mathbb{Z}/p\mathbb{Z})^\times$ é cíclico e analisamos a existência de elementos com ordens específicas, fundamentando a existência de raízes primitivas da unidade necessárias para a definição da Transformada.

Ciclicidade de $(\mathbb{Z}/p\mathbb{Z})^\times$

Seja \mathbb{K} um corpo finito. O teorema a seguir estabelece que seu grupo multiplicativo é cíclico. No contexto do trabalho, aplicamos isso para $\mathbb{K} = \mathbb{Z}/p\mathbb{Z}$.

Teorema 4. *Seja \mathbb{K} um corpo finito. Então o grupo multiplicativo $\mathbb{K}^\times = \mathbb{K} \setminus \{0\}$ é um grupo cíclico.*

Demonstração. Seja $n = |\mathbb{K}^\times|$ a ordem do grupo. Sendo \mathbb{K}^\times um grupo abeliano finito, Seja m a ordem máxima dentre os elementos de \mathbb{K}^\times . Uma propriedade fundamental de grupos abelianos finitos garante que a ordem de qualquer elemento do grupo divide a ordem máxima m .

Portanto, para todo $x \in \mathbb{K}^\times$, temos que:

$$x^m = 1.$$

Considere agora o polinômio $f(x) = x^m - 1$ com coeficientes no corpo \mathbb{K} . Sabemos que um polinômio de grau m sobre um corpo possui, no máximo, m raízes distintas.

No entanto, acabamos de ver que todos os n elementos de \mathbb{K}^\times satisfazem a equação $x^m - 1 = 0$. Logo, o polinômio tem n raízes. Para que isso não viole o limite de raízes, devemos ter necessariamente:

$$n \leq m.$$

Por outro lado, pelo Teorema de Lagrange, a ordem de qualquer elemento (incluindo o elemento de ordem máxima m) deve dividir a ordem do grupo (n). Logo, $m \leq n$.

Concluímos que $m = n$. Isso significa que existe um elemento em \mathbb{K}^\times cuja ordem é igual à ordem do grupo. Tal elemento é, por definição, um gerador de \mathbb{K}^\times . Portanto, o grupo é cíclico. \square

Existência de Elementos de Ordem d

Tendo estabelecido que $G = (\mathbb{Z}/p\mathbb{Z})^\times$ é cíclico de ordem $p - 1$, justificamos a recíproca mencionada no texto principal: para todo divisor da ordem do grupo, existe um elemento com aquela ordem.

Proposição 1. *Seja G um grupo cíclico finito de ordem M , gerado por g . Se d é um divisor de M , então existe um elemento em G com ordem exatamente d .*

Demonstração. Como $d \mid M$, podemos escrever $M = d \cdot k$ para algum inteiro k . Considere o elemento $h = g^k$. Calculando as potências de h :

$$h^d = (g^k)^d = g^{kd} = g^M = 1_G.$$

Portanto, a ordem de h divide d . Para ver que a ordem é exatamente d , suponha que $h^r = 1$ para $0 < r < d$. Então:

$$(g^k)^r = g^{kr} = 1.$$

Como a ordem de g é M , isso implicaria que $M \mid kr$, ou seja, $dk \mid kr$, o que implica $d \mid r$. Isso contradiz $r < d$.

Logo, $\text{ord}(h) = d$. \square

Aplicando ao contexto do texto principal: como $G = (\mathbb{Z}/p\mathbb{Z})^\times$ tem ordem $M = p - 1$, se impusermos a condição $2N \mid (p - 1)$, a proposição acima garante a existência de um elemento ψ de ordem $2N$ (uma raiz primitiva $2N$ -ésima da unidade).

References

- [1] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra*. 3rd. Cambridge: Cambridge University Press, 2013.
- [2] Syed Mahbub Hafiz et al. “Incompleteness in Number-Theoretic Transforms: New Trade-offs and Faster Lattice-Based Cryptographic Applications”. In: *2025 IEEE 10th European Symposium on Security and Privacy (EuroS&P)*. 2025, pp. 565–584. DOI: [10.1109/EuroSP63326.2025.00039](https://doi.org/10.1109/EuroSP63326.2025.00039).
- [3] Syed Mahbub Hafiz et al. “Incompleteness in Number-Theoretic Transforms: New Trade-offs and Faster Lattice-Based Cryptographic Applications”. en. In: () .
- [4] Matthias J. Kannwischer. “Polynomial Multiplication for Post-Quantum Cryptography”. Master’s Thesis. Tallinn University of Technology, 2018.
- [5] Patrick Longa and Michael Naehrig. “Speeding up the Number Theoretic Transform for Faster Ideal Lattice-Based Cryptography”. en. In: *Cryptology and Network Security*. Ed. by Sara Foresti and Giuseppe Persiano. Vol. 10052. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016, pp. 124–139. ISBN: 978-3-319-48964-3. DOI: [10.1007/978-3-319-48965-0_8](https://doi.org/10.1007/978-3-319-48965-0_8). URL: https://link.springer.com/10.1007/978-3-319-48965-0_8.
- [6] Henri J. Nussbaumer. *Fast Fourier Transform and Convolution Algorithms*. 2nd ed. Springer Berlin, Heidelberg, 1982.
- [7] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing*. 3rd. Prentice Hall, 2009.
- [8] Thales B. Paiva et al. *Faster amortized bootstrapping using the incomplete NTT for free*. IACR Cryptology ePrint Archive, Report 2025/696. 2025. URL: <https://eprint.iacr.org/2025/696>.
- [9] Ardianto Satriawan, Rella Maretta, and Hanho Lee. “A Complete Beginner Guide to the Number Theoretic Transform (NTT)”. In: *arXiv preprint arXiv:2307.01944* (2023).