Paradigmas de Solução de Problemas

Divisão e Conquista – Transformada Rápida de Fourier

Prof. Edson Alves - UnB/FGA

2020

Sumário

Transformada de Fourier

Série de Fourier

- Uma série de Fourier consiste na expansão de uma função períodica $f(x)$ em termos de senos e cosenos
- Isto possível porque as funções $\sin(mx)$ e $\sin(ny)$ são ortogonais para $m \neq n$ no intervalo $[-\pi, \pi]$:

$$\begin{aligned}\int_{-\pi}^{\pi} \sin(mx)\sin(nx)dx &= \int_{-\pi}^{\pi} \sin(mx)\cos(nx)dx \\ &= \int_{-\pi}^{\pi} \cos(mx)\cos(nx)dx = 0\end{aligned}$$

- Para $m = n$, segue que

$$\int_{-\pi}^{\pi} \sin^2(mx)dx = \int_{-\pi}^{\pi} \cos^2(mx)dx = \pi$$

Série de Fourier

- Deste modo,

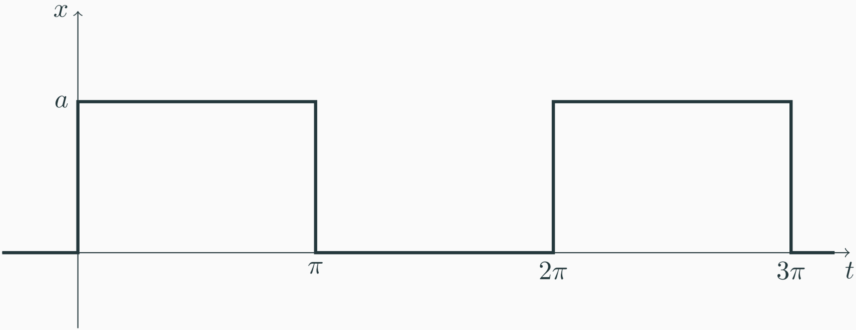
$$f(x) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} a_n \cos(nx) + \sum_{n=1}^{\infty} b_n \sin(nx),$$

onde

$$a_0 = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)dx$$
$$a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\cos(nx)dx$$
$$b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\sin(nx)dx$$

Exemplo: Onda Quadrada

Considere a onda quadrada abaixo:



Exemplo: Onda Quadrada

- O coeficiente a_0 é dado por

$$a_0 = \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)dt = \frac{1}{\pi}\int_{0}^{\pi} a\,dt = a$$

- Os coeficientes a_n, para $n \geq 1$, são todos iguais a zero, pois

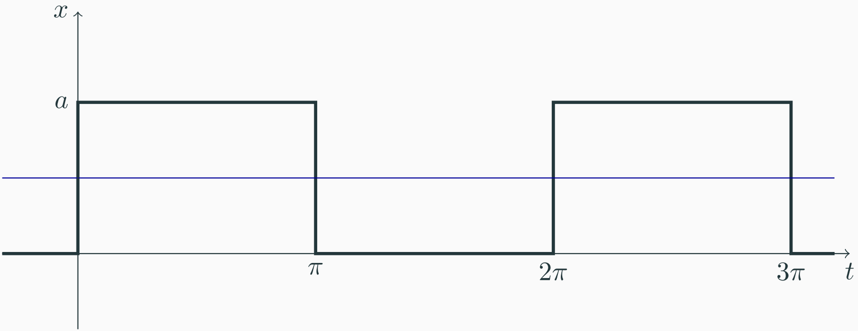
$$a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)\cos(nt)dt = \frac{a}{\pi}\left[\left.\frac{\sin(nt)}{n}\right|_0^{\pi}\right] = 0$$

- Os coeficientes b_n são iguais a zero, para n par, e

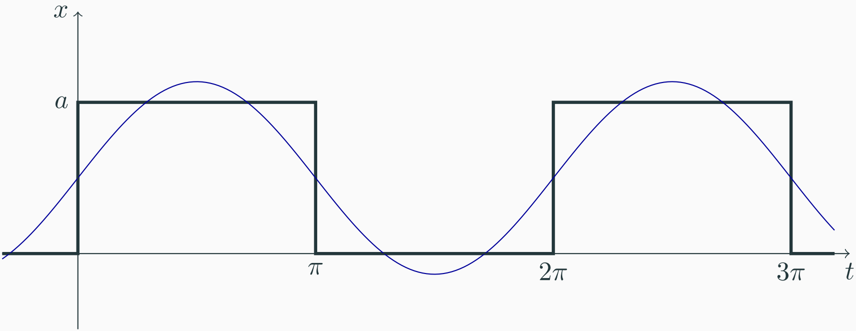
$$b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)\sin(nt)dt = -\frac{a}{\pi}\left[\left.\frac{\cos(nt)}{n}\right|_0^{\pi}\right] = \frac{2a}{m\pi},$$

se n é ímpar

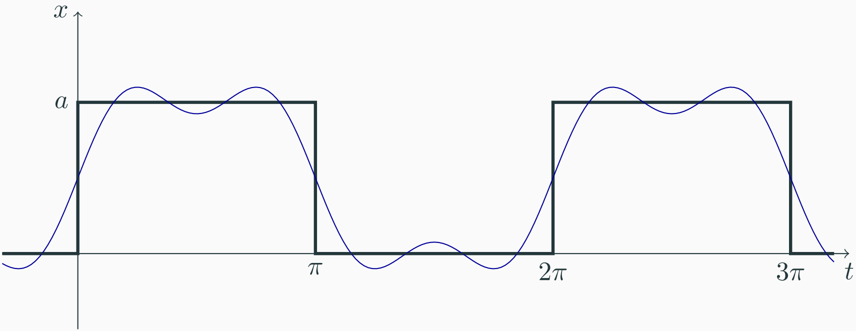
Exemplo: Aproximação da onda quadrada com $n = 0$



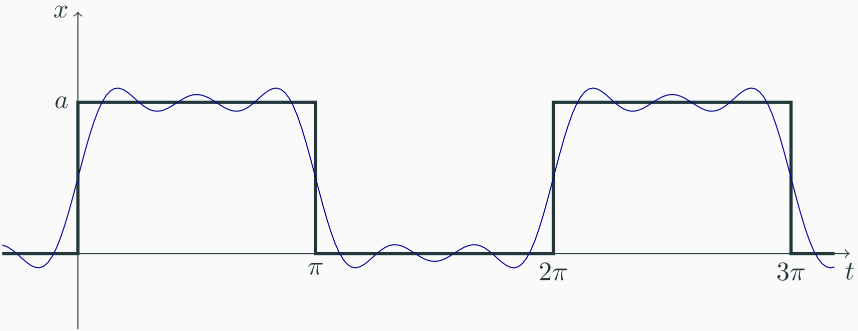
Exemplo: Aproximação da onda quadrada com $n = 1$



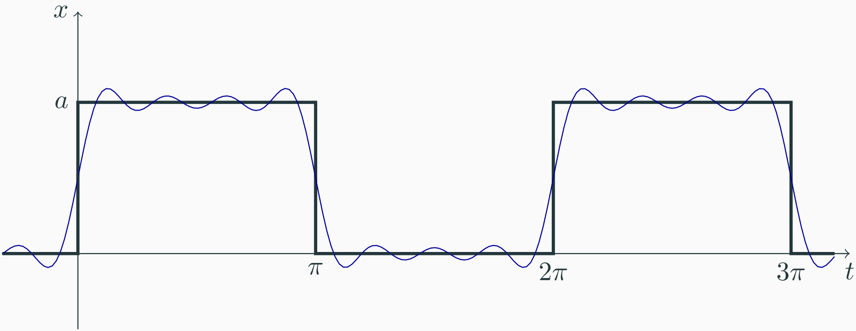
Exemplo: Aproximação da onda quadrada com $n = 3$



Exemplo: Aproximação da onda quadrada com $n = 5$



Exemplo: Aproximação da onda quadrada com $n = 7$



Série de Fourier com coeficientes complexos

- A série de Fourier pode ser estendida para coeficientes complexos a partir da observação que

$$e^{bi} = \cos b + i \sin b$$

- Seja $f(x)$ uma função nos reais. Faça

$$f(x) = \sum_{-\infty}^{\infty} A_n e^{inx}$$

- Assim, vale que

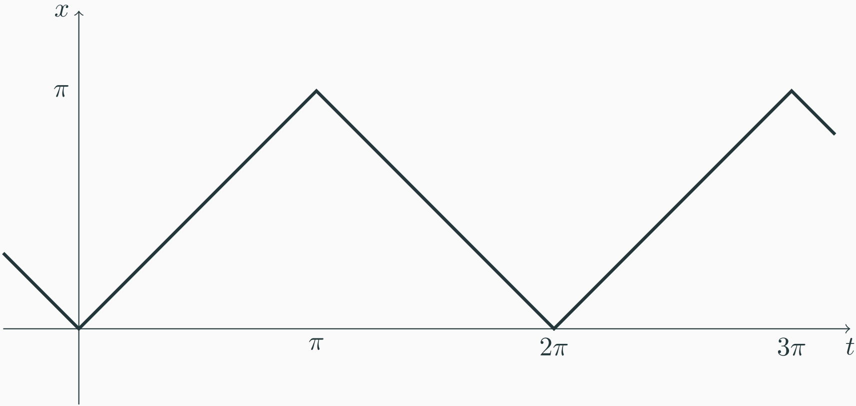
$$\int_{-\pi}^{\pi} f(x) e^{-imx} dx = 2\pi A_m,$$

de modo que

$$A_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{-inx} dx$$

Exemplo: Onda Triangular

Considere a onda triangular abaixo:



Exemplo: Onda Triangular

- No intervalo $[-\pi, \pi]$ temos que

$$f(x) = \begin{cases} -x, & \text{se } x \leq 0, \\ x, & \text{caso contrário} \end{cases}$$

- Daí

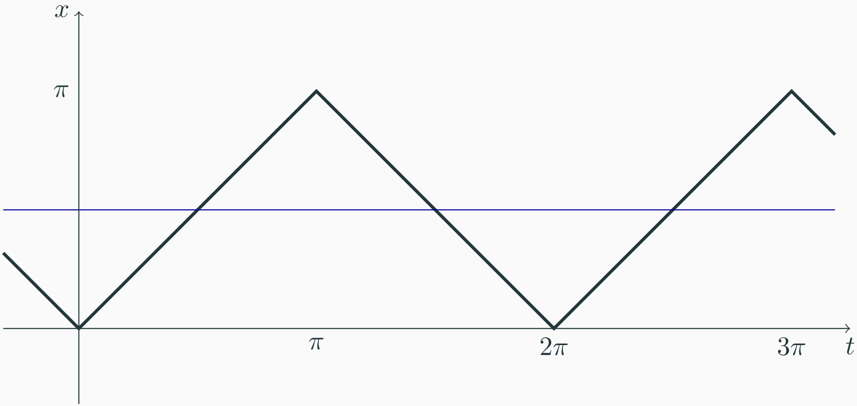
$$A_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) dx = \frac{\pi}{2}$$

- Para $n > 1$ ímpar vale que

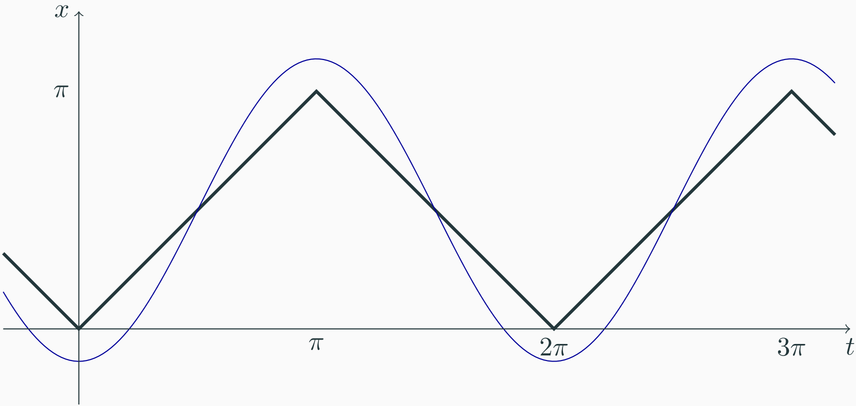
$$A_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{inx} dx = -\frac{4}{n^2}$$

- $A_n = 0$, se n é par

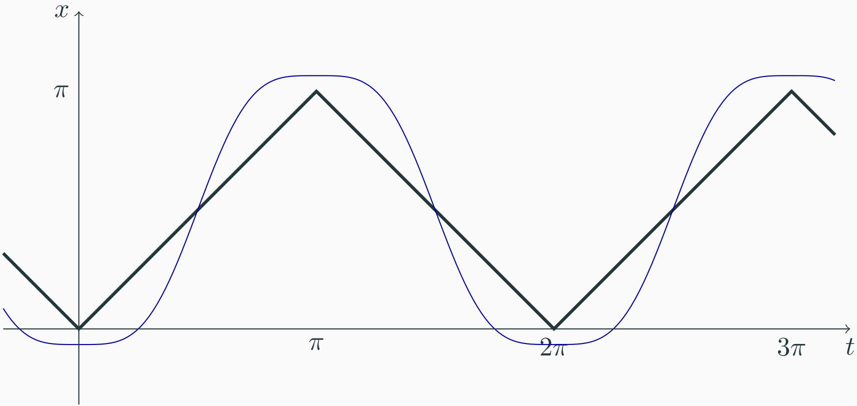
Exemplo: Aproximação com $n = 0$



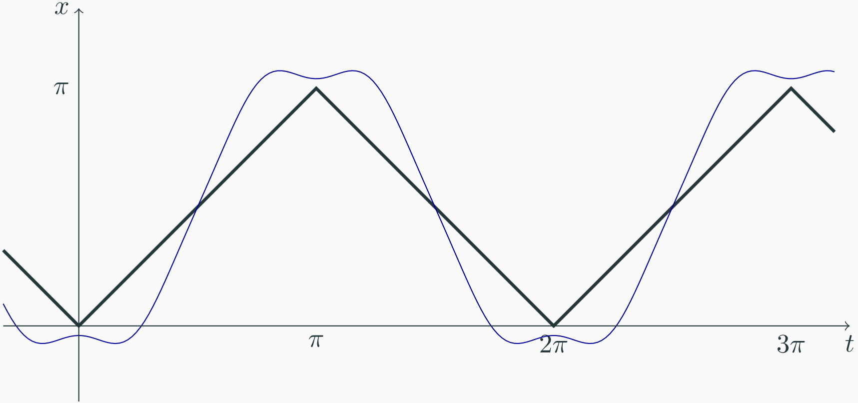
Exemplo: Aproximação com $n = 1$



Exemplo: Aproximação com $n = 3$



Exemplo: Aproximação com $n = 5$



Transformada de Fourier

- A transformada de Fourier é uma generalização das séries de Fourier com coeficientes complexos quando o período tende ao infinito
- Seja $f(x)$ uma função com um número finito de descontinuidades e tal existe a integral
$$\int_{-\infty}^{\infty} |f(x)| dx$$
- A Transformada de Fourier $\mathcal{F}$ de $f(x)$ é dada por
$$\mathcal{F}[f(x)] = F(k) = \int_{-\infty}^{\infty} f(x) e^{-2\pi ikx} dx$$
- A Transformada Inversa $\mathcal{F}^{-1}$ é dada por
$$\mathcal{F}^{-1}[F(k)] = f(x) = \int_{-\infty}^{\infty} F(k) e^{2\pi ikx} dk$$

Propriedades

- A Transformada de Fourier é linear:

$$\mathcal{F}[af(x) + bg(x)] = a\mathcal{F}[f(x)] + b\mathcal{F}[g(x)],$$

onde a e b são constantes

- A transformada da derivada da função está diretamente relacionada com a transformada da função

$$\mathcal{F}[f^{(n)}(x)](k) = (2\pi ik)^n \mathcal{F}[f(x)](k)$$

- **Teorema da Convolução**:

$$\mathcal{F}[f * g] = \mathcal{F}[f]\mathcal{F}[g]$$

Exemplo: Exponencial Descrescente

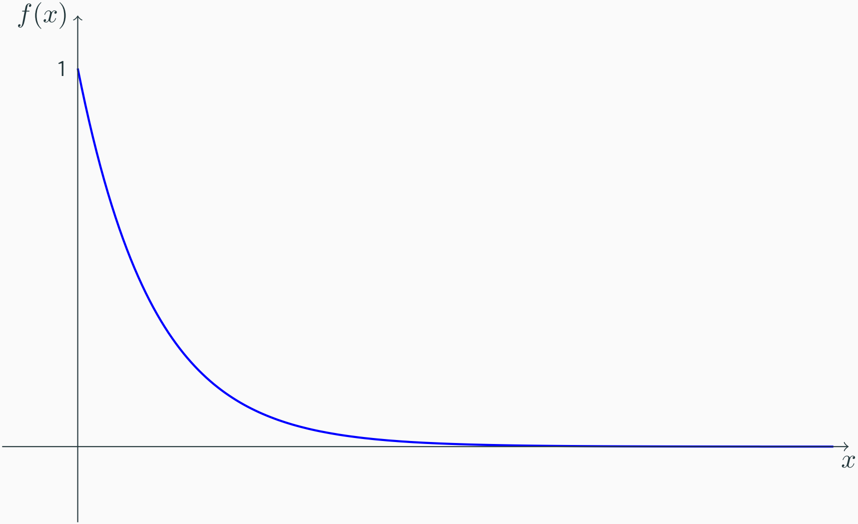
- Seja

$$f(x) = \begin{cases} e^{-x}, & \text{se } x \geq 0, \\ 0, & \text{caso contrário} \end{cases}$$

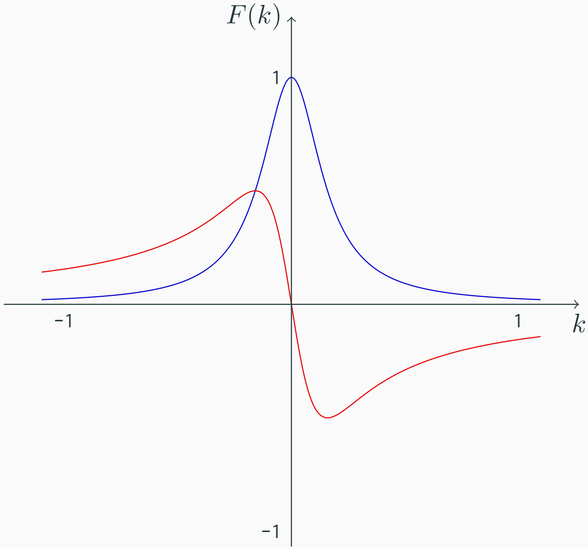
- Temos que

$$\begin{aligned} \mathcal{F}[f(x)] = F(k) &= \int_{-\infty}^{\infty} f(x) e^{-2\pi i k x} dx \\ &= \int_{0}^{\infty} e^{-x} e^{-2\pi i k x} dx = \int_{0}^{\infty} e^{-(1+2\pi i k)x} dx \\ &= -\frac{e^{-(1+2\pi i k)x}}{1+2\pi i k}\bigg|_{0}^{\infty} \\ &= \frac{1}{1+2\pi i k} \end{aligned}$$

Visualização da função $f(x)$



Visualização da parte real (azul) e imaginária (vermelha) da função $F(k)$



Transformada Discreta de Fourier

- Uma série $x_i = \{x_0, x_1, \ldots, x_{N-1}$ de N amostras de um sinal, igualmente espaçadas ao longo do tempo, pode ser interpretada como uma função y_i períodica de período N
- Para isso, defina $y(j) = x_i$, onde j é um inteiro tal que $j = N * q + i$, e $y(t) = 0$, se t não é inteiro
- Contudo, ao invés de fazer esta adaptação e utilizar a transformada de Fourier, é melhor utilizar a Transformada Discreta de Fourier (DFT):

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi ikn/N}$$

- A Transformada Discreta Inversa de Fourier (IDFT) é dada por:

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} F_k e^{-2\pi ikn/N}$$

Implementação da DFT e da IDFT em C++ em $O(N^2)$

```cpp
#include <bits/stdc++.h>
#include <complex>

using namespace std;

const double PI { acos(-1.0) };

template<typename T>
vector<complex<T>> dft(const vector<T>& xs)
{
    int N = (int) xs.size();
    vector<complex<T>> F(N, 0);

    for (int k = 0; k < N; ++k)
        for (int i = 0; i < N; ++i)
            F[k] += xs[i]*exp(complex<T>(0, -2*PI*i*k/N));

    return F;
}

```

Implementação da DFT e da IDFT em C++ em $O(N^2)$

```
template<typename T>
vector<T> idft(const vector<complex<T>>& Fs)
{
    int N = (int) Fs.size();
    vector<T> f(N, 0);

    for (int x = 0; x < N; ++x)
        for (int k = 0; k < N; ++k)
            f[x] += (1.0/N)*(Fs[k]*exp(complex<T>(0, 2*PI*x*k/N))).real();

    return f;
}
```

Aplicação da DFT: Multiplicação de Polinômios

- A convolução entre duas funções $f(x)$ e $g(x)$ é uma função $h(x) = f(x) * g(x)$ que representa como a forma de uma função é modificada pela outra
- Ela é a integral do produto de ambas funções, sendo que uma delas é invertida e deslocada:

$$h(x) = f(x) * g(x) = \int_{-\infty}^{\infty} f(\tau)(t - \tau)\, d\tau$$

- A convolução discreta de f e g é dada por

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n-m]$$

- Se $f(x)$ e $g(x)$ são sequências de coeficientes de dois polinômios, a convolução de ambas será igual ao produto destes polinômios

Visualização da multiplicação de polinômios

$$f(x) = x^2 - 5x + 6$$
$$g(x) = x^3 - 2x^2 + 3x - 1$$

6	−5	1	
−1	3	−2	1

Visualização da multiplicação de polinômios

$$f(x) = x^2 - 5x + 6$$
$$g(x) = x^3 - 2x^2 + 3x - 1$$
$$h(x) = -\mathbf{6}$$

			6	−5	1
1	−2	3	−1		

Visualização da multiplicação de polinômios

$$f(x) = x^2 - 5x + 6$$
$$g(x) = x^3 - 2x^2 + 3x - 1$$
$$h(x) = \mathbf{23}x - 6$$

		6	−5	1
1	−2	3	−1	

Visualização da multiplicação de polinômios

$$f(x) = x^2 - 5x + 6$$
$$g(x) = x^3 - 2x^2 + 3x - 1$$
$$h(x) = -\mathbf{28}x^2 + 23x - 6$$

	6	−5	1
1	−2	3	−1

Visualização da multiplicação de polinômios

$$f(x) = x^2 - 5x + 6$$
$$g(x) = x^3 - 2x^2 + 3x - 1$$
$$h(x) = \mathbf{19}x^3 - 28x^2 + 23x - 6$$

6	−5	1	
1	−2	3	−1

Visualização da multiplicação de polinômios

$$f(x) = x^2 - 5x + 6$$
$$g(x) = x^3 - 2x^2 + 3x - 1$$
$$h(x) = -\mathbf{7}x^4 + 19x^3 - 28x^2 + 23x - 6$$

6	−5	1		
	1	−2	3	−1

Visualização da multiplicação de polinômios

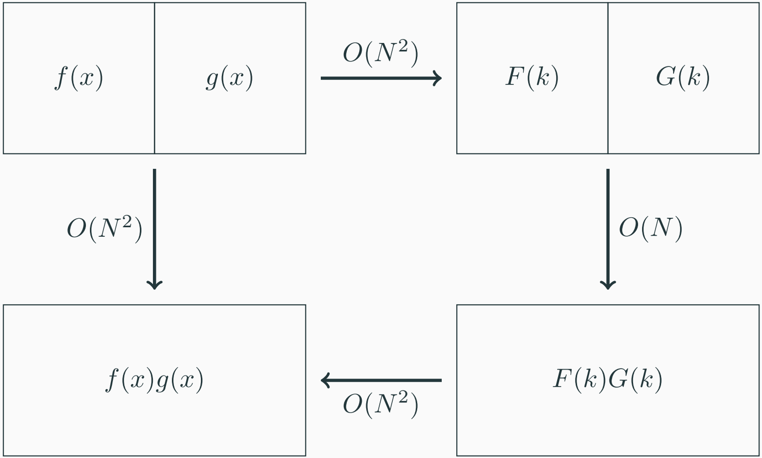
$$f(x) = x^2 - 5x + 6$$
$$g(x) = x^3 - 2x^2 + 3x - 1$$
$$h(x) = x^5 - 7x^4 + 19x^3 - 28x^2 + 23x - 6$$

6	−5	1			
		1	−2	3	−1

Aplicação da DFT: Multiplicação de Polinômios

- Considere as transformadas $F(k)$ e $G(k)$ dos polinômios $f(x)$ e $g(x)$
- Pelo Teorema da Convolução, a transformada do produto será $H(k) = F(k)G(k)$, onde a multiplicação, neste caso, é termo a termo
- No domínio do tempo, onde estão os polinômios, a multiplicação polinomial é convolução, com complexidade $O(N^2)$, onde N é o maior dentre os graus
- No domínio das frequências, onde estão as transformadas, a convolução se torna uma multiplicação termo a termo, com complexidade $O(N)$
- Assim, é possível realizar a multiplicação de polinômios indiretamente, computando as transformadas $F(k)$ e $G(k)$, fazendo a multiplicação termo a termo, e computando a inversa de $H(k)$

Visualização da multiplicação indireta de polinômios



Implementação da multiplicação indireta de polinômios

```
vector<double>
operator*(const vector<double>& fx, const vector<double>& gx)
{
    auto n = fx.size() - 1, m = gx.size() - 1;
    vector<double> xs(n + m + 1), ys(n + m + 1);

    copy(fx.begin(), fx.end(), xs.begin());
    copy(gx.begin(), gx.end(), ys.begin());

    auto Fk = dft(xs), Gk = dft(ys), Hk(Fk);

    for (size_t i = 0; i < Hk.size(); ++i)
        Hk[i] *= Gk[i];

    return idft(Hk);
}
```

Transformada Rápida de Fourier

Referências

Referências

1. **CHEEVER**, Erick. The Fourier Series, acesso em 12/08/2020.
2. CP Algorithms. Fast Fourier Transform, acesso em 13/08/2020.
3. Standford. Lecture 11 – The Fourier Transform, acesso em 13/08/2020.
4. Wikipédia. Discrete Fourier Transform, acesso em 13/08/2020.
5. Wolfram. Fourier Series, acesso em 12/08/2020.
6. Wolfram. Fourier Transform, acesso em 13/08/2020.