

# 1. FFT

## 1.1. Enfoque

Se decidió desarrollar un algoritmo fft, ya que se contó con conocimiento del campo de ciencias de la computación. En las optimizaciones progresivas se llegó a un algoritmo con  $\frac{n}{2} \log_2 n$  multiplicaciones,  $6n$  memoria, en otras palabras,  $\Theta(n \log(n))$  tiempo,  $\Theta(n)$  memoria

## 1.2. El problema

El cálculo de la dft consiste en producir una la salida  $X[k]$  de coeficientes mediante la entrada  $x[n]$ , ambos arreglos de igual longitud  $N$ . La fórmula que describe la dft que utilizamos fue

$$DFT[X[n]] = X[k] = \sum_{n=0}^{N-1} x(n) e^{-i2\pi kn/N} \quad (1)$$

Se necesita resolver la dft en tiempo subcuadrático, es decir en menos de  $\Theta(n^2)$ , que es la solución trivial.

## 1.3. Primera observación de interés

Si la longitud de la suceción,  $N$  es par se puede escribir  $X[k]$  como

$$DFT[X[n]] = X[k] = \underbrace{\sum_{n=0}^{\frac{N}{2}-1} x(2n) e^{-i2\pi kn/(N/2)}}_{DFT[X[2n]=Y[n]]} + e^{i2\pi k/N} \overbrace{\sum_{n=0}^{\frac{N}{2}-1} x(2n+1) e^{-i2\pi kn/(N/2)}}^{DFT[X[2n+1]=Z[n]]} \quad (2)$$

$$DFT[\underbrace{X[n]}_{0 \leq n < N}] = DFT[\underbrace{Y[n]}_{0 \leq n < N/2}] + e^{i2\pi k/N} DFT[\underbrace{Z[n]}_{0 \leq n < N/2}] \quad (3)$$

Donde  $X[2n] = Y[n]$ ,  $X[2n+1]$ ,  $Y[2n+1]$  son sub-sucesiones de  $X[n]$  tomando los índices pares e impares respectivamente. Podemos decir que descompusimos el problema de la DFT como dos problemas de DFT con sucesiones de la mitad de tamaño, es decir, conseguimos una solución recursiva. Es importante observar que la unica forma de que la solución sea viable es que  $N$  sea una potencia de dos, para poder garantizar que al dividir los subproblemas siempre  $N_i$  sea par.

## 1.4. Formalización

Ahora que ya se mostró la primera observación de interés, la cual mostró el caracter recursivo del problema, escribiremos dicha formulación de una forma precisa. En primer lugar a partir de ahora

$$e^{i2\pi kn/N} = W[k][N] \quad (4)$$

Además, cuando  $N$  sea igual a la longitud de la subsecuencia más grande (la raíz del problema)

$$W[k][N] = W[k] \quad (5)$$