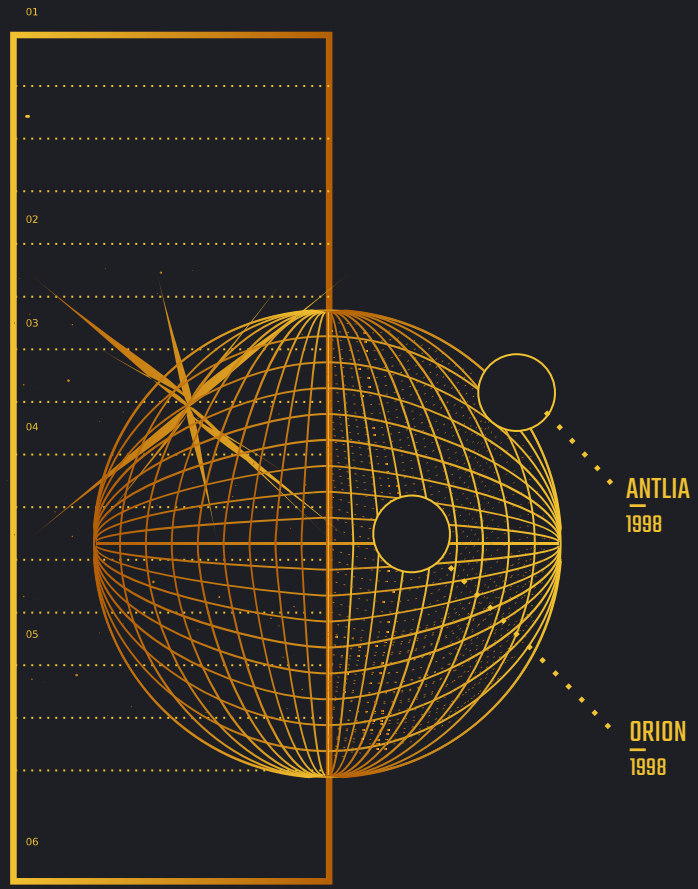
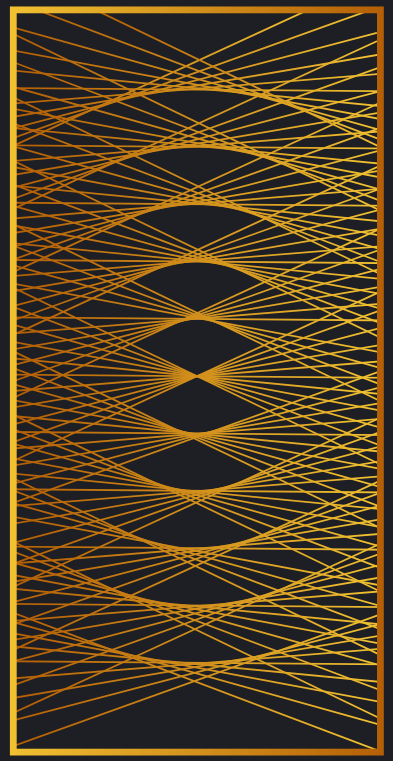


FOURIER
FAST TRASFORM

Gabriel Valdivia
Giulia Naval
Rodrigo Torres

Contenido



01.
INTRODUCCIÓN A LA TRANSFORMADA DE FOURIER

Ecuación de Euler para funciones trigonométricas complejas

$$e^{\pm ix} = \cos(x) \pm isen(x)$$

SERIE DE FOURIER

Fourier plantea que cualquier función periódica puede ser representada como la suma de funciones seno y coseno con amplitud determinada y una frecuencia múltiplo de la frecuencia principal.

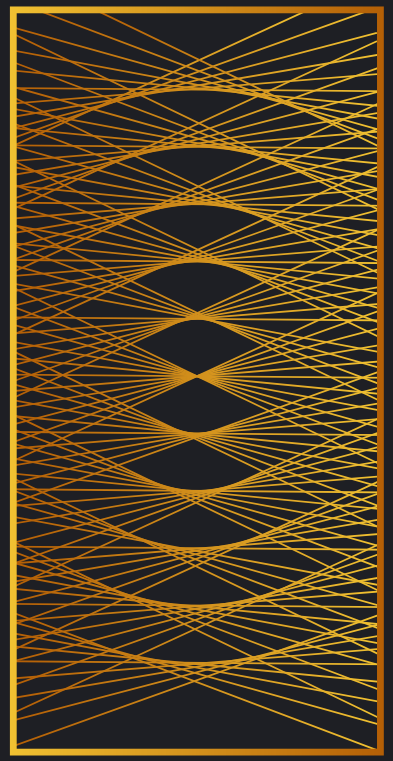
$$P(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n cos(\omega nt) + b_n sen(\omega nt)]$$

$$\omega = 2\pi f_0$$

$$P(t) = \sum_{n=-\infty}^{\infty} C_n e^{i\omega nt}$$

01 02 03 04 05 06

01 02 03 04 05 06



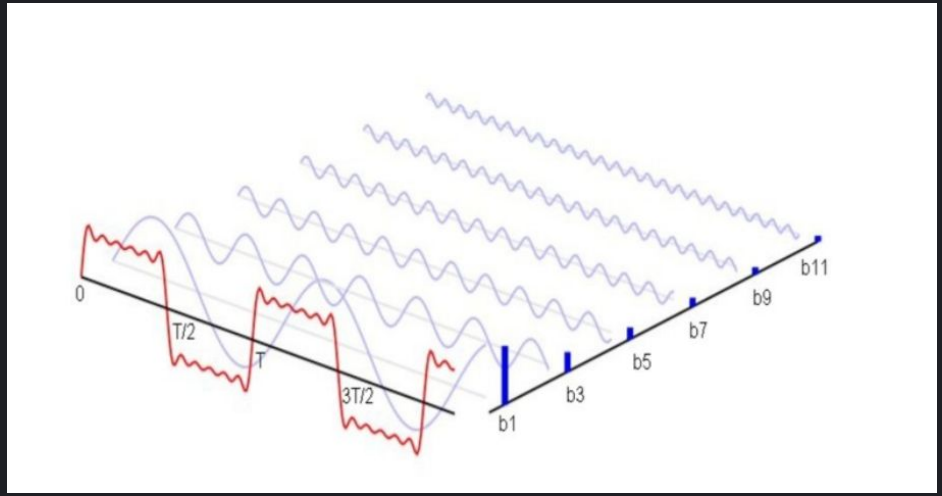
02.

TRANSFORMADA DE FOURIER

¿QUÉ ES LA TF?

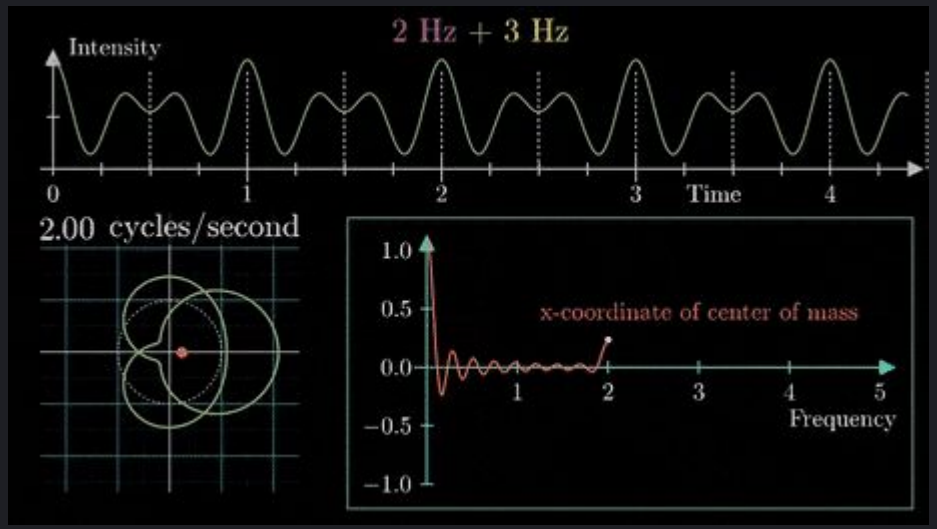
La transformada de Fourier nos permite cambiar una señal recibida en base al tiempo a otra en base a frecuencia.

Se relaciona con la Serie de Fourier, ya que puede representar todas las amplitudes de las curvas encontradas en esta serie.



$$X(f) = \int_{-\infty}^{\infty} X(t) \cdot e^{-2\pi i f t} dt$$

¿Cómo funciona la transformada?



Si tomamos una función finita que establece la intensidad en función del tiempo y la “enrollamos” en otro plano, aumentando progresivamente la frecuencia de enrollamiento, podemos ver a la intensidad como la distancia que hay desde el origen hasta el centro de masa de la gráfica.

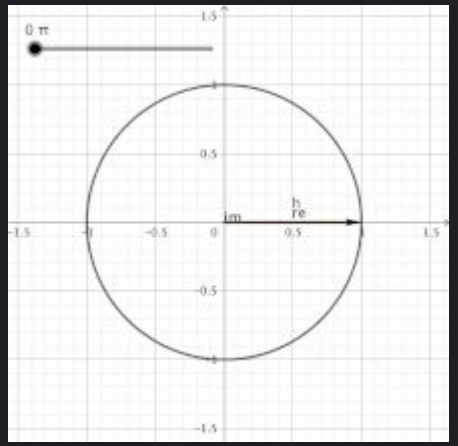
Si llevamos esta frecuencia al eje x de otro plano y la distancia al eje y, obtendremos así la idea de la Transformada de Fourier.

Las coordenadas en las que se encuentra el centro de masa pueden ser expresadas como números complejos, y el módulo de la distancia desde el origen al punto en un círculo unitario sería $e^{i\varphi}$

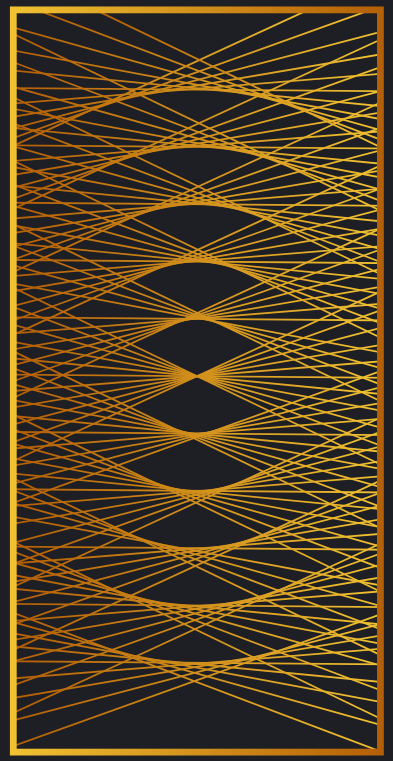
Si queremos describir la rotación que se hace en un determinado tiempo y con una determinada frecuencia la fórmula quedaría como $e^{2\pi ift}$

Si tomamos en cuenta que en la transformada de Fourier se considera esta rotación en el sentido del reloj, tendríamos que cambiar de signo a la expresión.

Por último, como queremos que esta rotación se adecúe al comportamiento de una determinada función, tendríamos que reemplazar al radio del círculo por ella, multiplicando a la expresión. Y para hacer un seguimiento del centro de masa a lo largo de todo el intervalo, integrar todo llegando finalmente a la expresión:



$$\int_{t1}^{t2} g(x)e^{-2\pi ift}dt$$

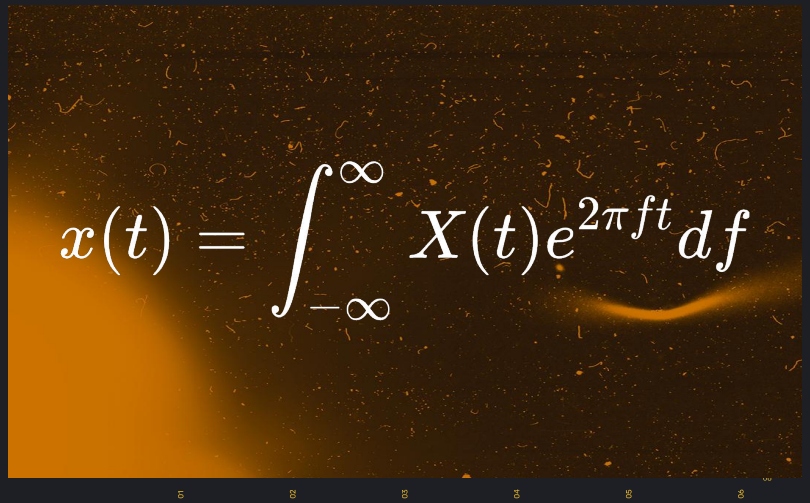


03.

TRANSFORMADA INVERSA DE FOURIER

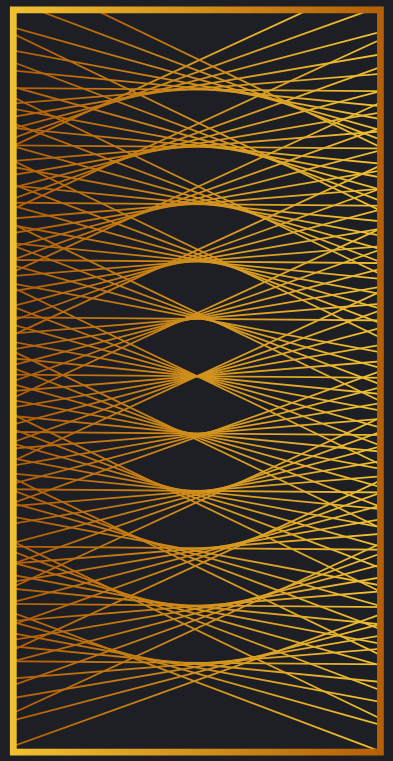
Revierte el estado en que se encuentra la representación de una onda luego de haberle aplicado la Transformada de Fourier normal.

¿Qué hace?



¿Cómo lo hace?

Convierte una serie de tamaño potencia de 2 números complejos, puntos en el espectro de frecuencia, en una serie del mismo tamaño en un dominio de tiempo.

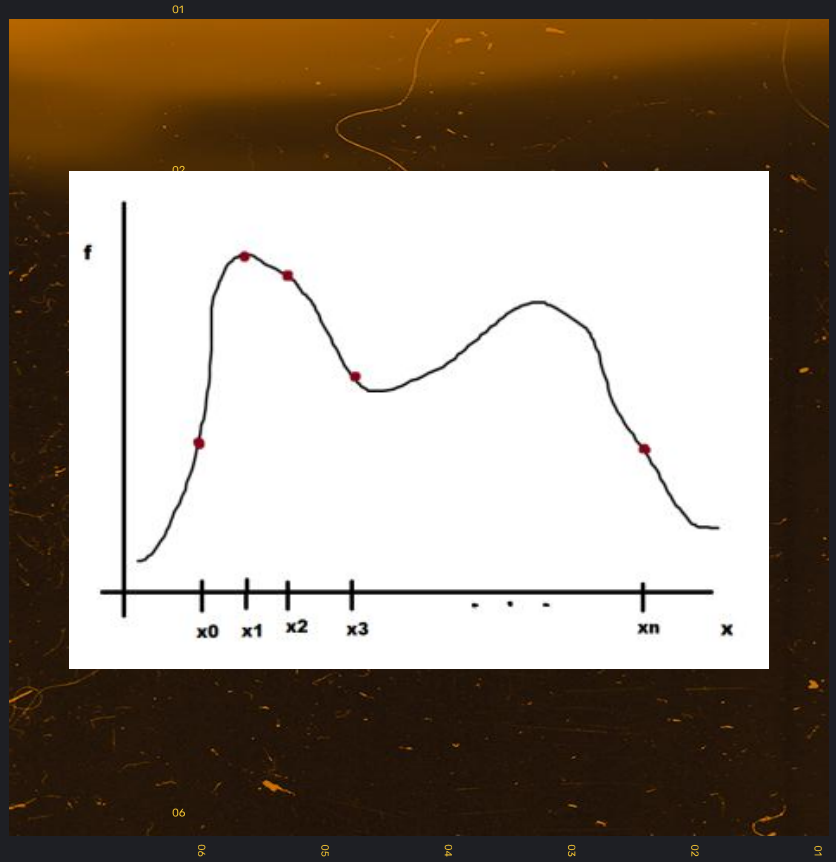


04.

TRANSFORMADA DE FOURIER DISCRETA

- Trabaja con un conjunto finito de puntos de una función, y los transforma a puntos complejos que nos darán información de la función.

- Sus resultados indican el grado de relación de una determinada función sinusoidal con la función principal. Por ello el DFT se relaciona mucho con la serie Fourier.



¿Cómo lo calculamos?

$$\hat{f}_k = \sum_{j=0}^{n-1} f_j \cdot e^{-i2\pi j \frac{k}{n}}$$

Donde ^fk hace referencia al punto transformado y fj al punto no transformado

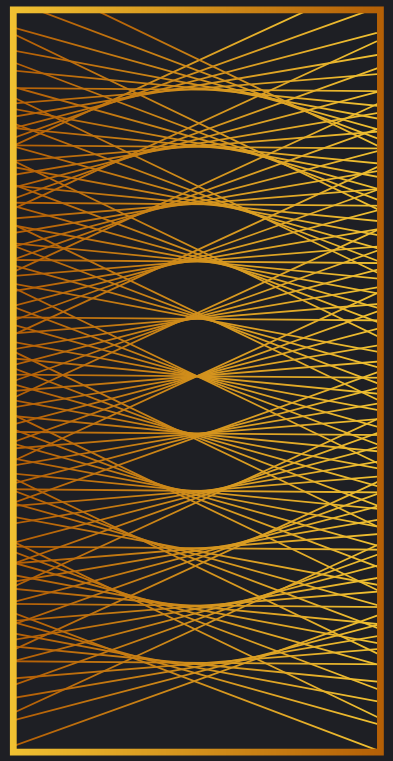
$$\{f_1, f_2, \dots f_n\} \xrightarrow{DFT} \{\hat{f}_1, \hat{f}_2, \dots, \hat{f}_n\}$$

La exponencial compleja se puede expresar en términos de sen y cos de acuerdo a la fórmula de Euler, por lo que representa una función sinusoidal con características dependientes del punto en cuestión (j) que se comparará para encontrar su relación con la función original.

Todas las transformadas están añadiendo múltiplos a esta frecuencia fundamental. Usando esto podemos mapear todas las operaciones de la ecuación en una matriz de la siguiente forma:

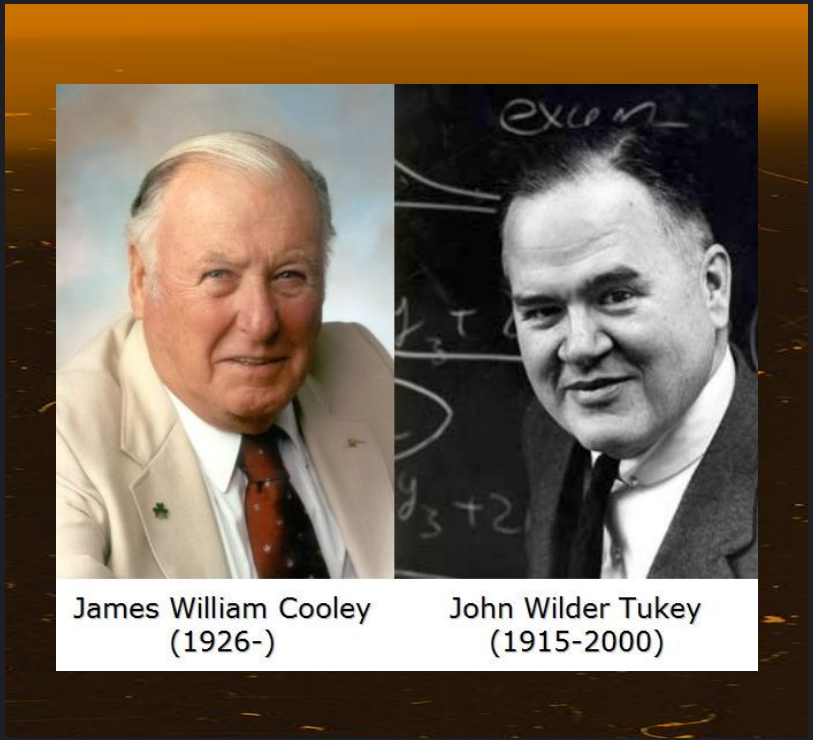
$$\begin{bmatrix} \hat{f}_1 \\ \hat{f}_2 \\ \hat{f}_3 \\ \dots \\ \hat{f}_n \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega_n & \omega_n^2 & \dots & \omega_n^{n-1} \\ 1 & \omega_n^2 & \omega_n^4 & \dots & \omega_n^{2(n-1)} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \dots & \omega_n^{(n-1)^2} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \dots \\ f_n \end{bmatrix}$$

Como el DFT trabaja con números complejos, sus resultados también serán complejos. La magnitud de cada número representará qué tanto se tiene de esa frecuencia, y su ángulo representará su fase (desplazamiento del origen).



05.

ALGORITMO FFT (COOLEY TUKEY)



Pensando en formas de implementar y mejorar la TFD, James W. Cooley y John W. Tukey publicaron en su paper insigne en 1965 una implementación de la transformada utilizando el paradigma ”divide y vencerás”, denominada como Transformada Rápida de Fourier. En dicho trabajo, se destaca la ventaja que presenta el algoritmo en arreglos de tamaño N potencia de 2 sobre otros con N distintos, al lograr una complejidad de O(Nlog(N)) para el primer caso.

Este algoritmo consigue reducir el número de cálculos al dividir el arreglo de tamaño N del polinomio en 2 de tamaño N/2, uno con el contenido de los índices impares y el otro con el de los índices pares.

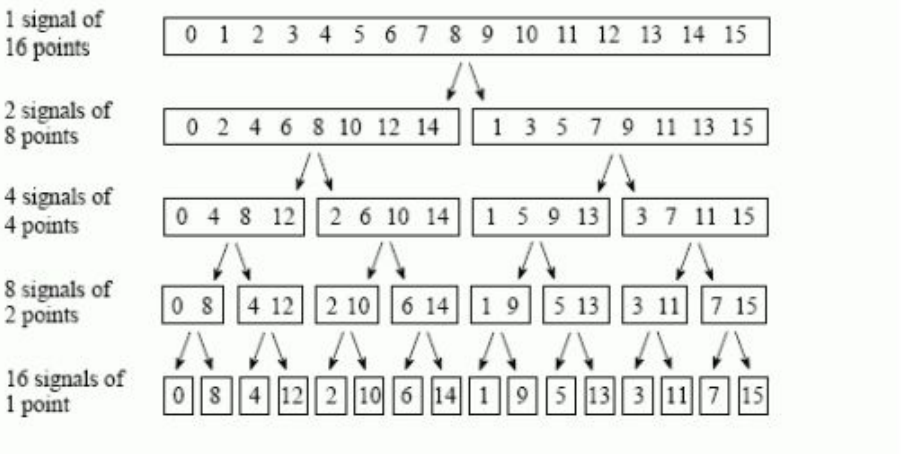


FIGURE 12-2
The FFT decomposition. An N point signal is decomposed into N signals each containing a single point. Each stage uses an *interlace decomposition*, separating the even and odd numbered samples.

De esta forma, se puede simplificar el tiempo que toma la multiplicación de matrices en la DFT segmentando esta matriz en otras más pequeñas.

Suponemos que tenemos un ^f = F1024. Es decir, una matriz de 1024 * 1024. Para segmentar esta matriz se usa el criterio de índices pares o impares, quedando lo siguiente:

$$\hat{f} = F_{1024} \cdot f = \begin{bmatrix} I_{512} & D_{512} \\ I_{512} & D_{512} \end{bmatrix} \begin{bmatrix} F_{512} & 0 \\ 0 & F_{512} \end{bmatrix} \begin{bmatrix} f_{even} \\ f_{odd} \end{bmatrix}$$

Donde f(even) son los índices pares, f(odd) los índices impares. I(512) es la matriz identidad de tamaño 512 * 512, y D(512) está dado por:

$$D_{512} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & \omega & 0 & \dots & 0 \\ 0 & 0 & \omega^2 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \omega^{511} \end{bmatrix}$$

Esta distribución también se puede lograr usando inversión de bits.

Sample numbers in normal order			Sample numbers after bit reversal	
Decimal	*Binary*		*Decimal*	*Binary*
0	0000		0	0000
1	0001		8	1000
2	0010		4	0100
3	0011		12	1100
4	0100		2	0010
5	0101		10	1010
6	0110	⇨	6	0100
7	0111		14	1110
8	1000		1	0001
9	1001		9	1001
10	1010		5	0101
11	1011		13	1101
12	1100		3	0011
13	1101		11	1011
14	1110		7	0111
15	1111		15	1111

FIGURE 12-3
The FFT bit reversal sorting. The FFT time domain decomposition can be implemented by sorting the samples according to bit reversed order.

Una vez dividido el arreglo en partes más pequeñas, el algoritmo busca encontrar su valor en el espectro de frecuencia, donde obviamente, la frecuencia de las divisiones más pequeñas, es decir de 1 solo punto, será igual a sí mismo. Luego de esto, tendrá que regresar y juntar los resultados en el orden inverso al que se hicieron las divisiones.

Para sumar los puntos de cada una, también se tienen que diluir la señal completando con 0s las posiciones pares o impares, dependiendo del sub-arreglo, lo que equivale a una duplicación de la data de la señal en el dominio de la frecuencia. Esta discordancia en la separación con 0s, o de forma más precisa, este corrimiento del segundo sub-arreglo corresponde a la multiplicación del mismo por un sinusoide.

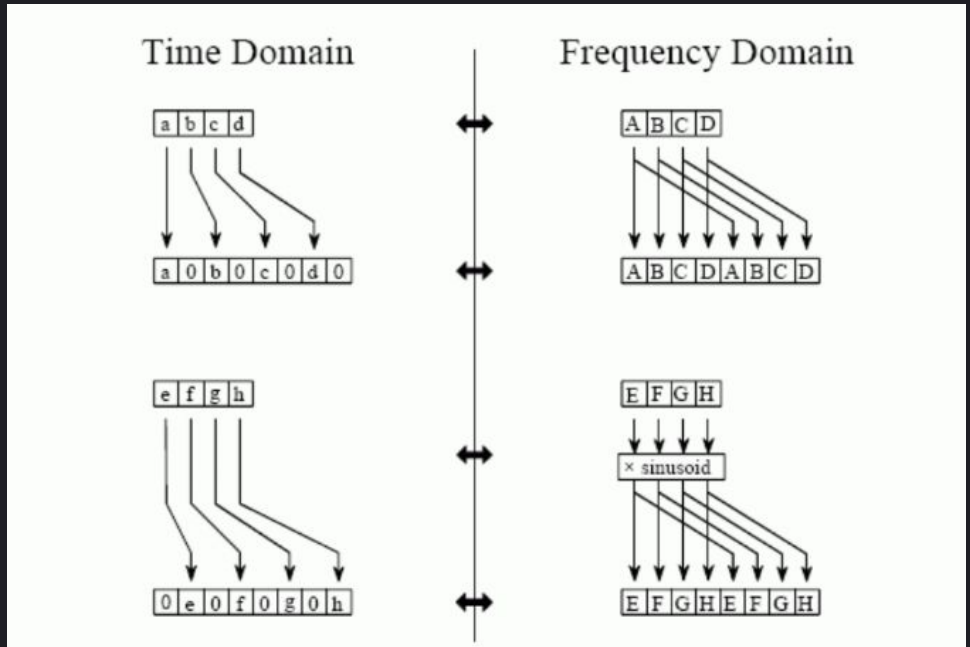


FIGURE 12-4
The FFT synthesis. When a time domain signal is diluted with zeros, the frequency domain is duplicated. If the time domain signal is also shifted by one sample during the dilution, the spectrum will additionally be multiplied by a sinusoid.

MARIPOSA

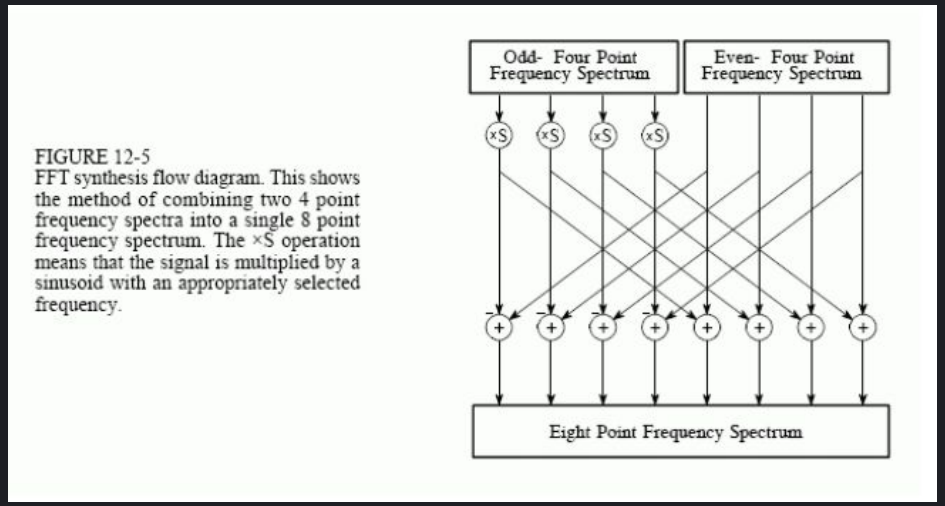


FIGURE 12-5
FFT synthesis flow diagram. This shows the method of combining two 4 point frequency spectra into a single 8 point frequency spectrum. The ×S operation means that the signal is multiplied by a sinusoid with an appropriately selected frequency.

Esta forma de sumar los puntos del espectro mientras se los reordena, tiene el nombre de mariposa y es la operación base de toda la TFR.

ALGORITMO FFT

```
void fft(vector<complex<float>>& v)
{
    int N = v.size();
    if (N <= 1) return;


    vector<complex<float>> impar;
    vector<complex<float>> par;
    for (int i=0; i<N-1; i+=2) {
        par.push_back(v[i]);
        impar.push_back(v[i+1]);
    }

    fft(par);
    fft(impar);
    for (int k = 0; k < N/2; ++k)
    {
        complex<float> t = std::polar(1.0,
-2 * PI * k / N) * impar[k];
        v[k] = par[k] + t;
        v[k+N/2] = par[k] - t;
    }
}
```

01 02 03 04 05 06

ALGORITMO IFFT

```
void ifft(vector<complex<float>>& v)
{
    for (auto& x : v)
        x = conj(x);
    fft(v);
    complex<float> s(v.size(),0);
    for (auto& x : v)
        x = conj(x) / s;

}
```

GRACIAS

003-1040559 1250 003-77156.8 1760 0009-14563.7 73273