# Introducción a la Teoría del Procesamiento Digital de Señales de Audio

# Transformada de Fourier Discreta

Resumen

- Respuesta en frecuencia de un sistema

- Convolución a través del dominio de la frecuencia
  - Convolución circular
  - Método de solapamiento y suma (overlap-add)
  - Convolución FFT o convolución rápida

- La transformada discreta de Fourier con números complejos

- La transformada rápida de Fourier

# Respuesta en frecuencia

Todo sistema lineal puede caracterizarse **completamente** en términos de cómo cambia la amplitud y la fase de ondas sinusoidales. Esto se denomina r**espuesta en frecuencia**.

En el dominio del tiempo los sistemas se describen en términos de convolución con la **respuesta al impulso**:

$$x[n] * h[n] = y[n]$$

# Respuesta en frecuencia

Todo sistema lineal puede caracterizarse **completamente** en términos de cómo cambia la amplitud y la fase de ondas sinusoidales. Esto se denomina r**espuesta en frecuencia**.

En el dominio del tiempo los sistemas se describen en términos de convolución con la **respuesta al impulso**:

$$x[n] * h[n] = y[n]$$

Ambas representaciones tienen la información completa sobre el sistema, por lo que debe haber una correspondencia entre ellas:

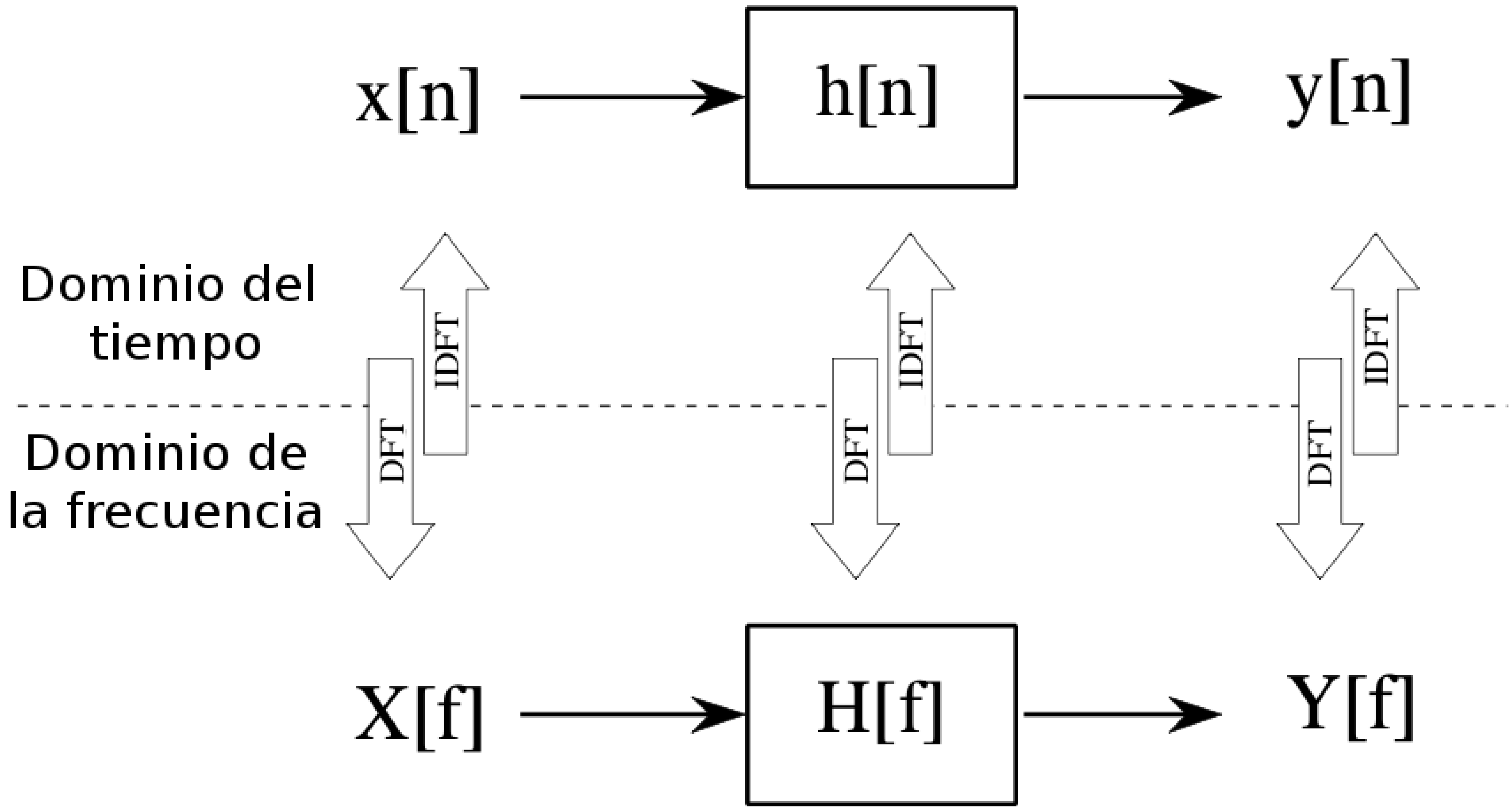
**La respuesta en frecuencia de un sistema es la transformada de Fourier de su respuesta al impulso.**

# Respuesta en frecuencia

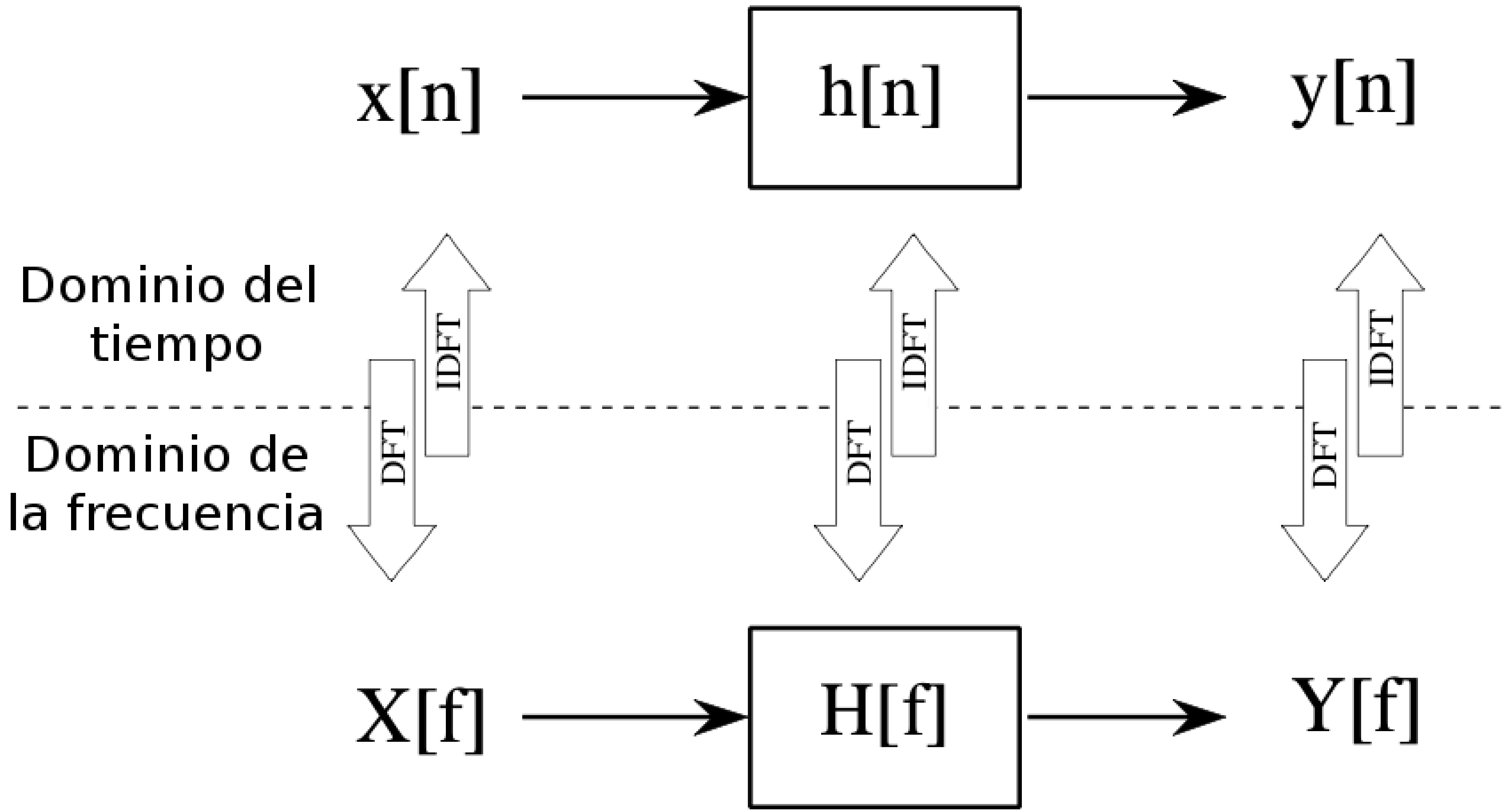
$$x[n] \longrightarrow \boxed{h[n]} \longrightarrow y[n]$$

Dominio del
tiempo

# Respuesta en frecuencia



# Respuesta en frecuencia




La convolución en el tiempo corresponde a la multiplicación en frecuencia

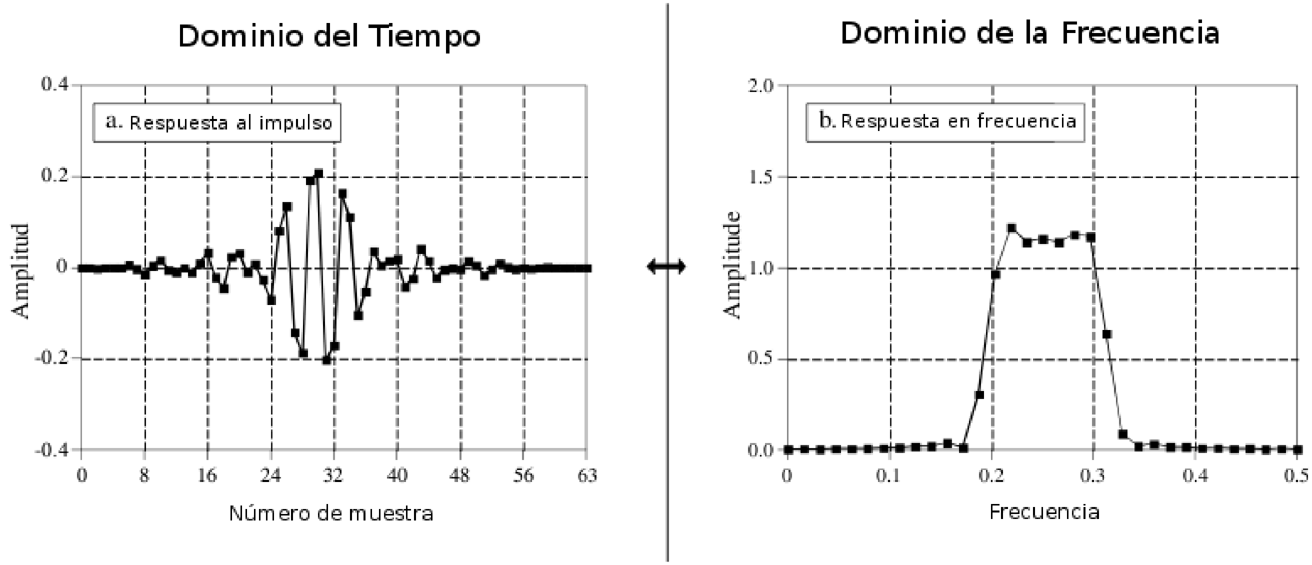
x[n] * h[n] = y[n]

X[f] x H[f] = Y[f]

# Respuesta en frecuencia

Ejemplo

Análisis de un sistema:
- respuesta al impulso
- respuesta en frecuencia
  (por medio de  la DFT)



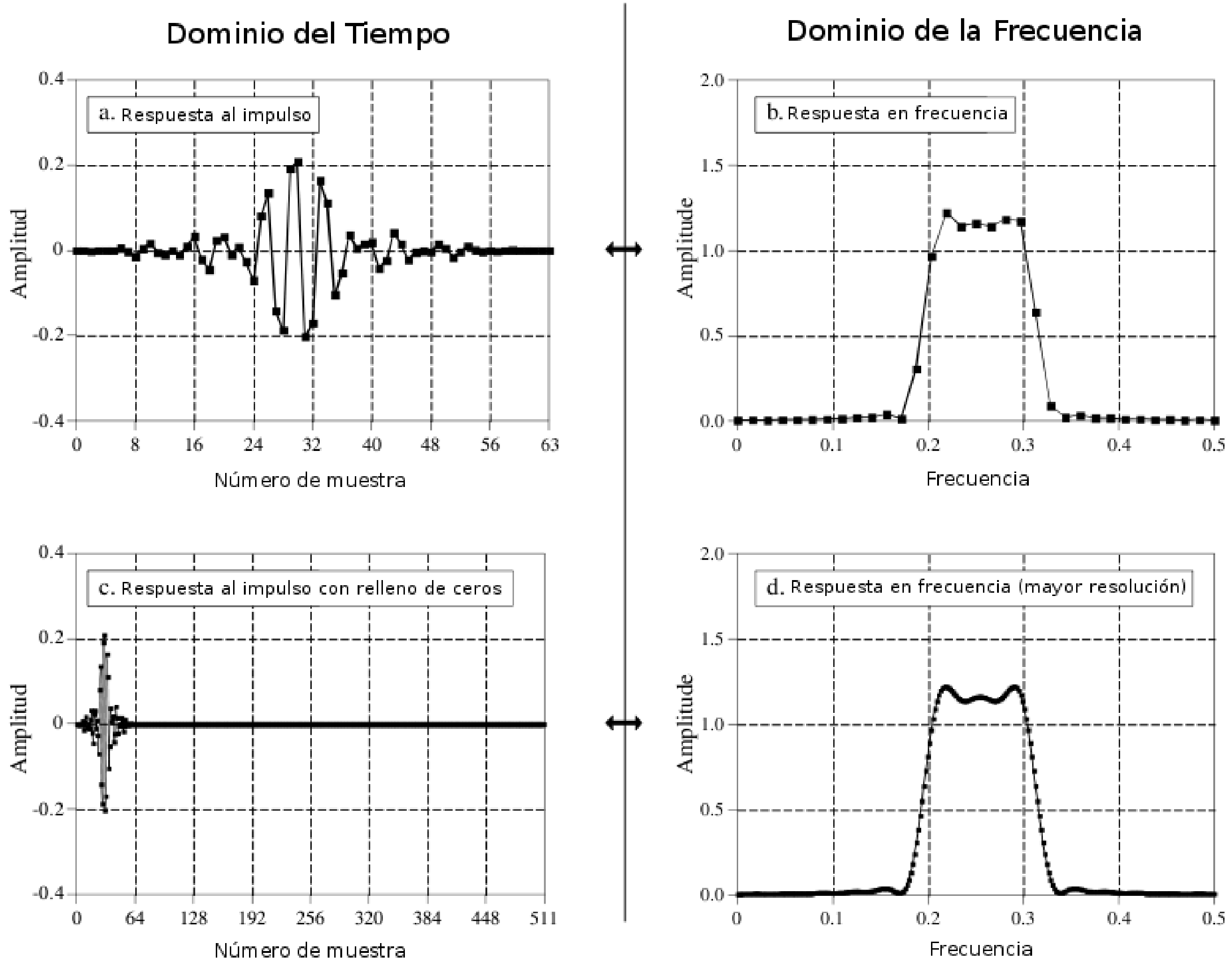
# Respuesta en frecuencia

Ejemplo

Análisis de un sistema:
- respuesta al impulso
- respuesta en frecuencia
  (por medio de la DFT)

Relleno de ceros:
aumento de resolución.

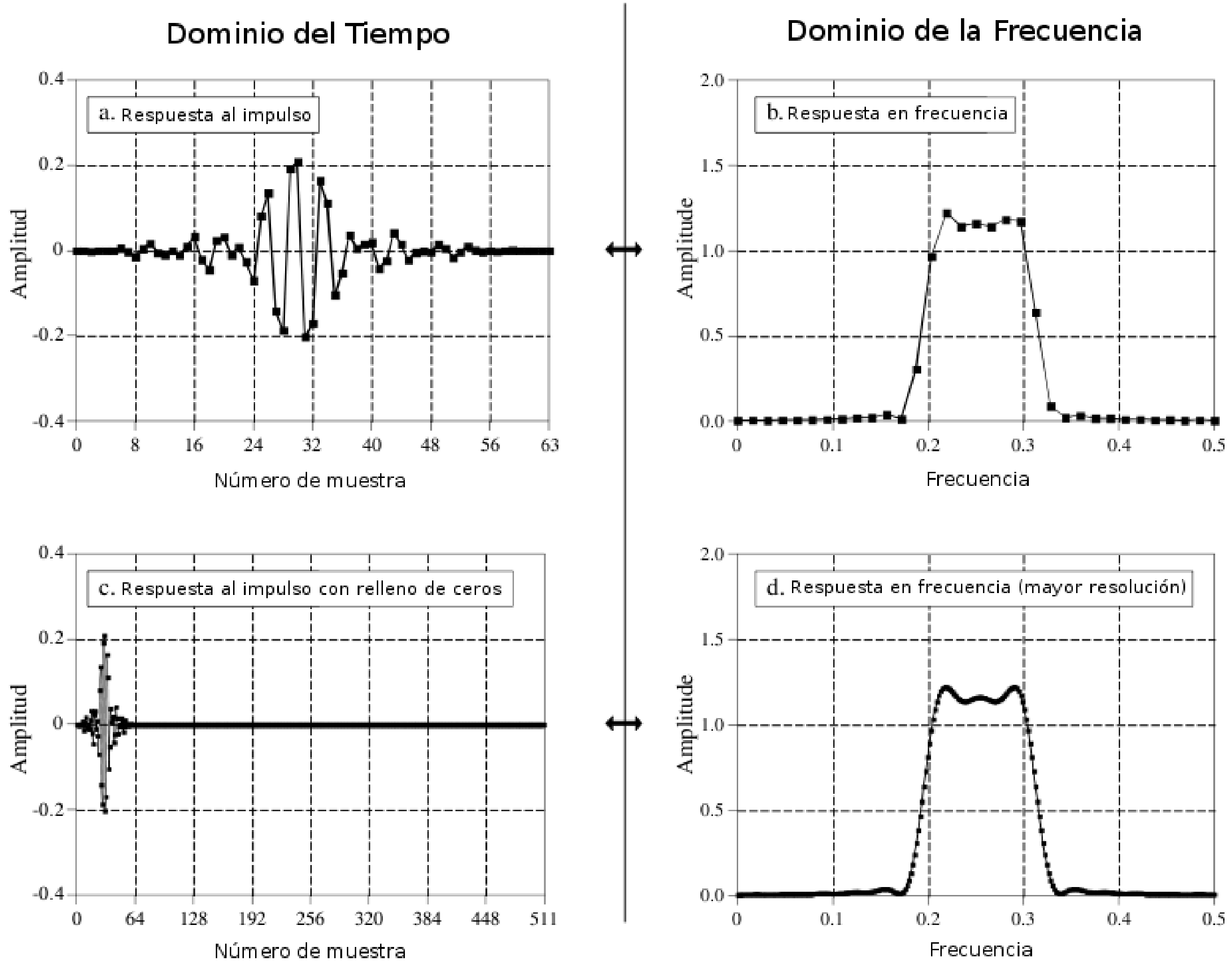
Cuánto puede
aumentarse la
resolución?



# Respuesta en frecuencia

Ejemplo

Análisis de un sistema:
- respuesta al impulso
- respuesta en frecuencia
  (por medio de  la DFT)




Aún cuando la respuesta al impulso es discreta, la respuesta en frecuencia es continua.

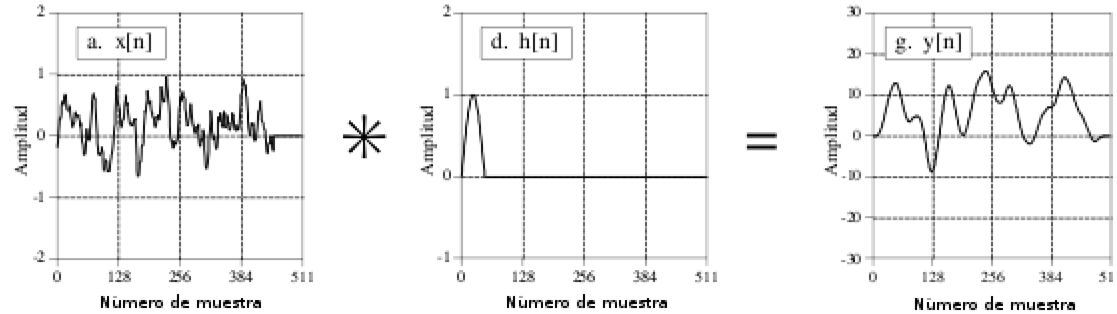
Rellenar con infinitos ceros corresponde a tomar la DTFT

**En términos matemáticos la respuesta en frecuencia de un sistema se obtiene tomando la DTFT de su respuesta al impulso**

# Convolución a través del dominio de la frecuencia

Ejemplo:

Señal y respuesta al impulso: convolución en el tiempo.



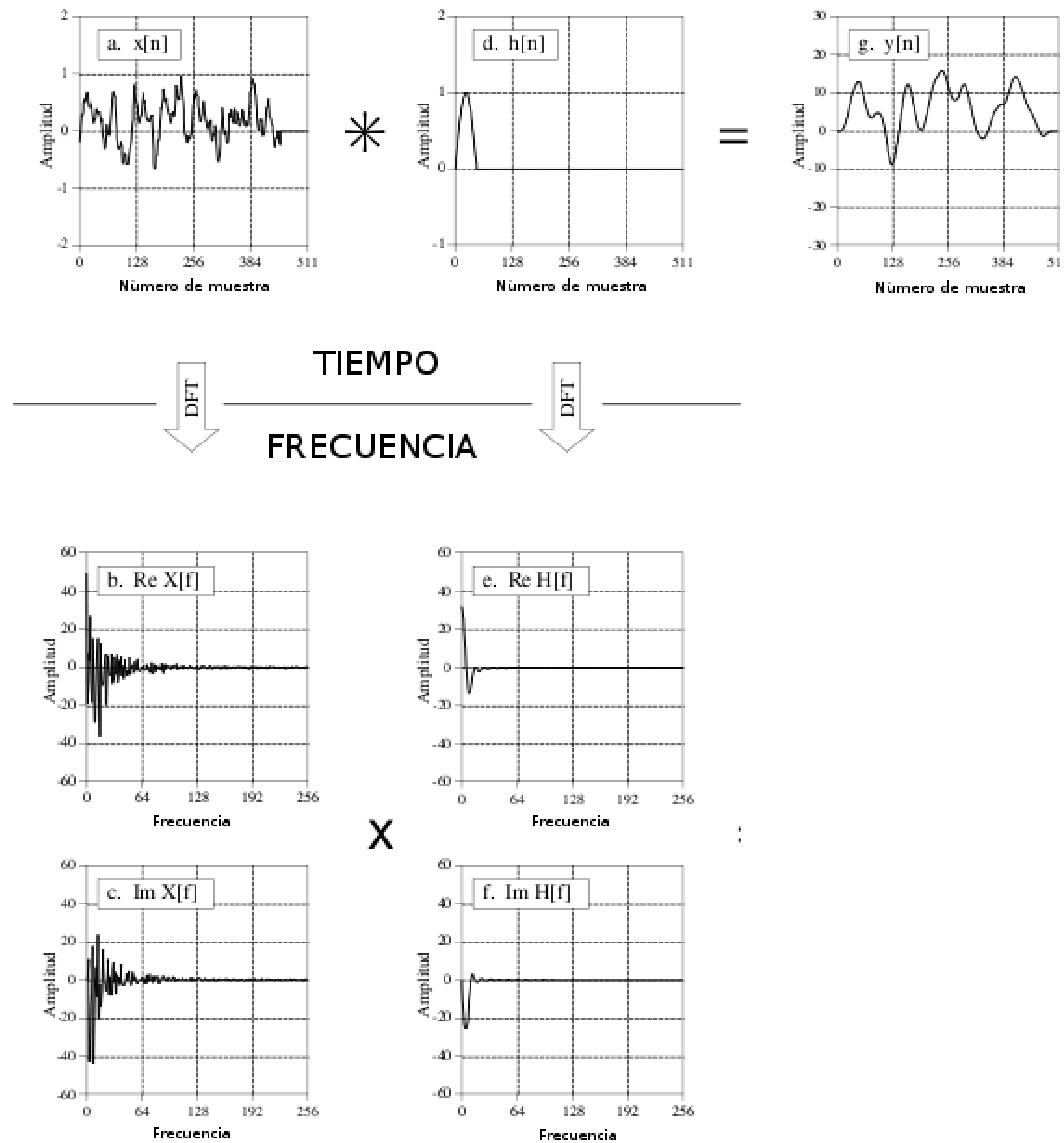
# Convolución a través del dominio de la frecuencia

Ejemplo:

Señal y respuesta al impulso: convolución en el tiempo.

Remplazar la convolución por:

 2 DFT directas
 multiplicación de espectros



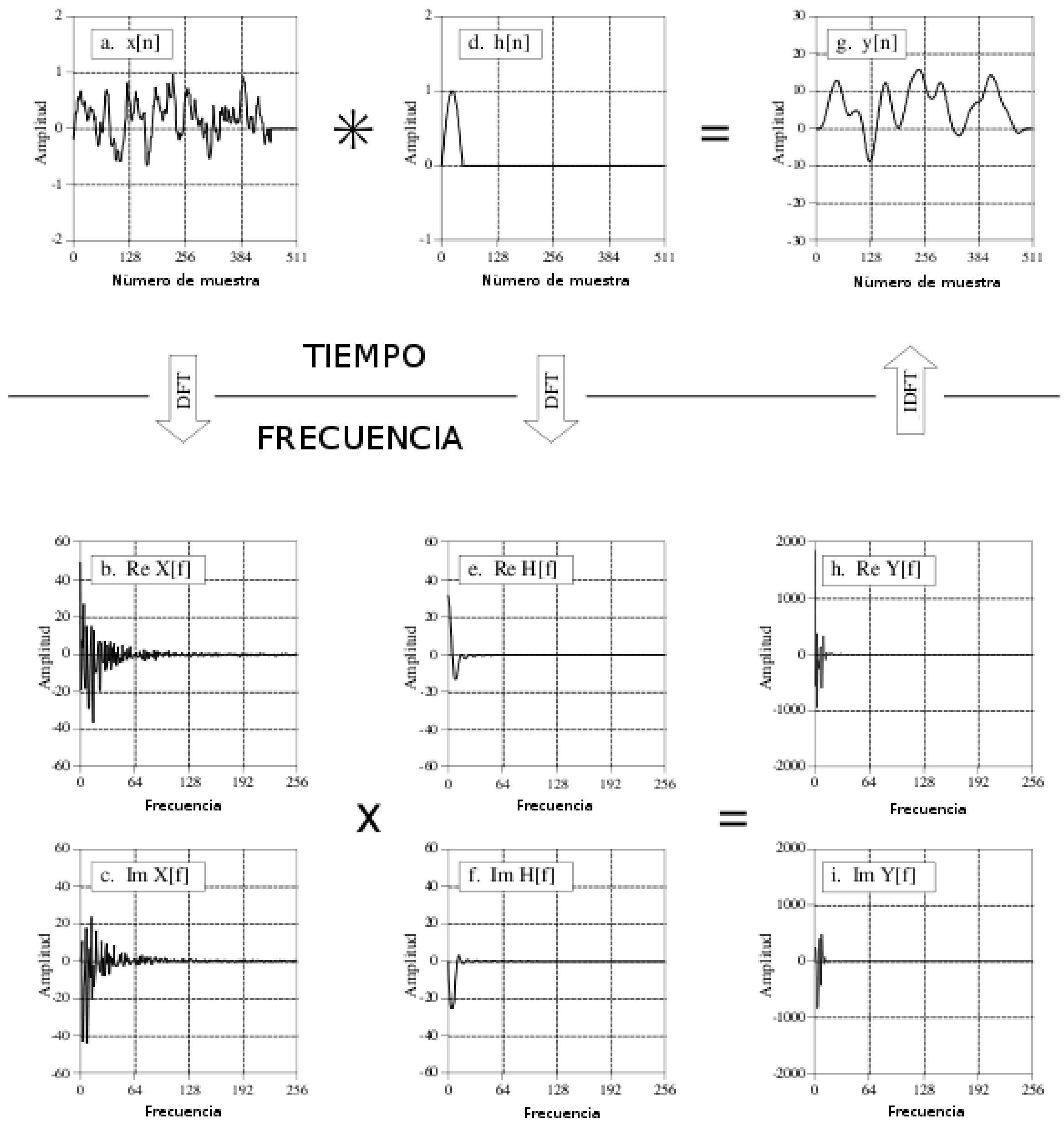
# Convolución a través del dominio de la frecuencia

Ejemplo:

Señal y respuesta al impulso: convolución en el tiempo.

Remplazar la convolución por:

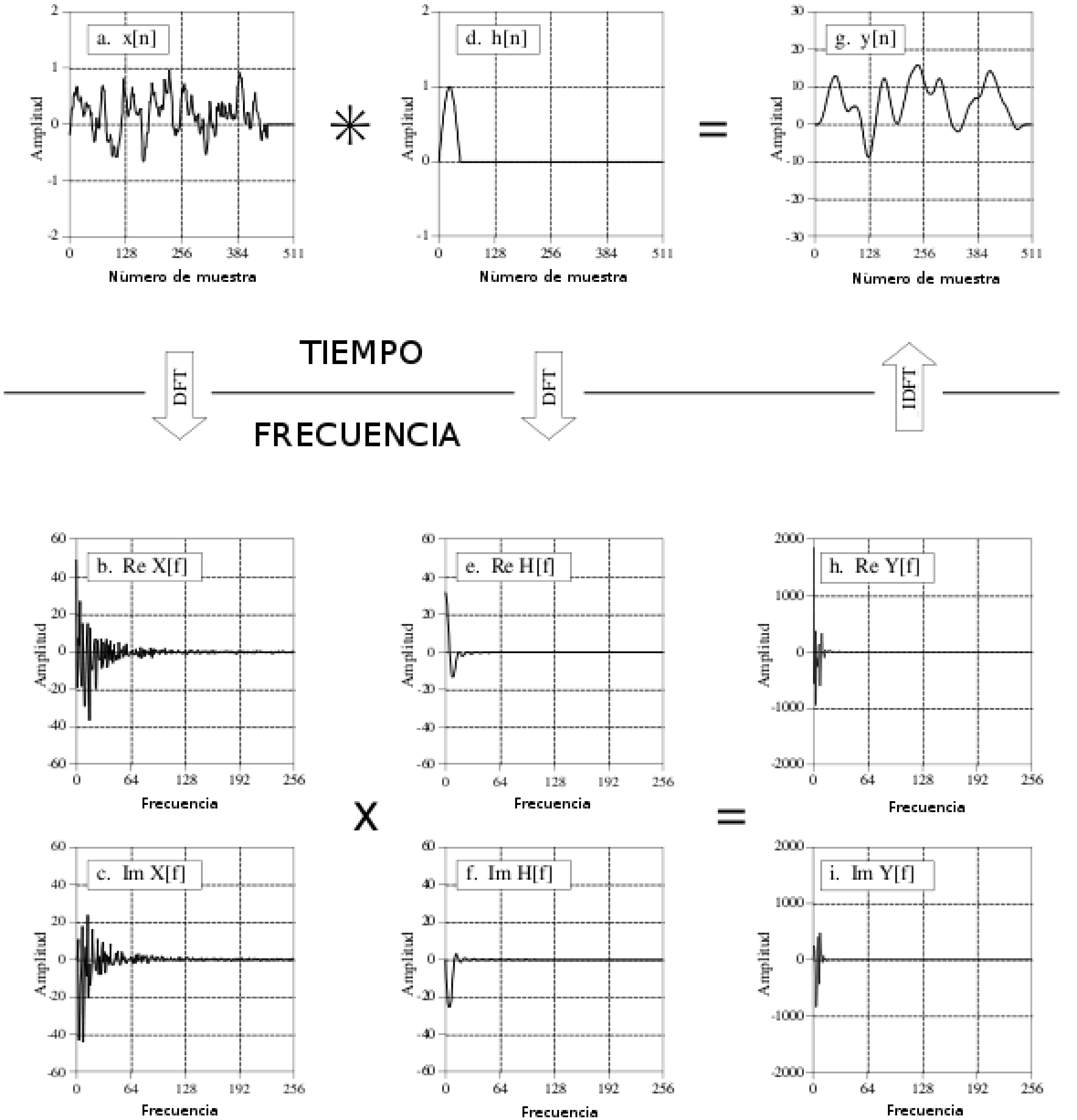
2 DFT directas
multiplicación de espectros
1 DFT inversa



# Convolución a través del dominio de la frecuencia

Ejemplo:

Señal y respuesta al impulso: convolución en el tiempo.




Ventajas?

- facilidad de interpretación en algunos casos (e.g. deconvolución)
- velocidad de cómputo al usar la FFT

# Convolución a través del dominio de la frecuencia

Producto en el dominio de la frecuencia:

$$X[f] \times H[f] = Y[f]$$

# Convolución a través del dominio de la frecuencia

Producto en el dominio de la frecuencia:

**X[f] x H[f] = Y[f]**

En notación polar:

**MagY[f] = MagX[f] x MagH[f], FaseY[f] = FaseX[f] + FaseH[f]**

Se multiplican las magnitudes y se suman las fases.

# Convolución a través del dominio de la frecuencia

Producto en el dominio de la frecuencia:

**X[f] x H[f] = Y[f]**

En notación polar:

**MagY[f] = MagX[f] x MagH[f],    FaseY[f] = FaseX[f] + FaseH[f]**

Se multiplican las magnitudes y se suman las fases.

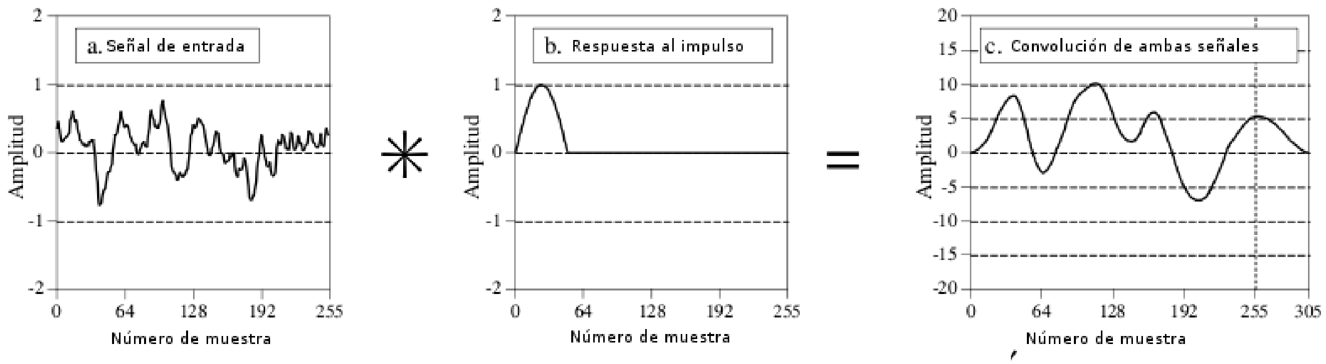
Esto equivale en notación rectangular a:

**ReY[f] = ReX[f] x ReH[f] -  ImX[f] x ImH[f]**
**ImY[f] = ImX[f] x ReH[f] + ReX[f] x ImH[f]**

# Convolución circular

Señal de **N** puntos y respuesta al impulso de **M** puntos: señal resultante de **N+M-1** puntos.

Ejemplo,
N = 256, M = 51,
N+M-1 = 306.

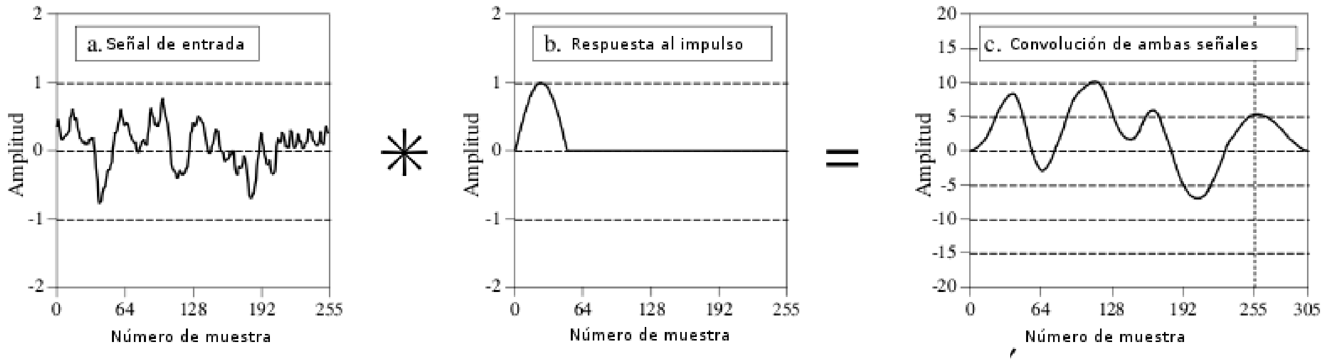


# Convolución circular

Señal de **N** puntos y respuesta al impulso de **M** puntos: señal resultante de **N+M-1** puntos.

Ejemplo,
N = 256, M = 51,
N+M-1 = 306.

Qué ocurre si se usan DFTs de 256 puntos?



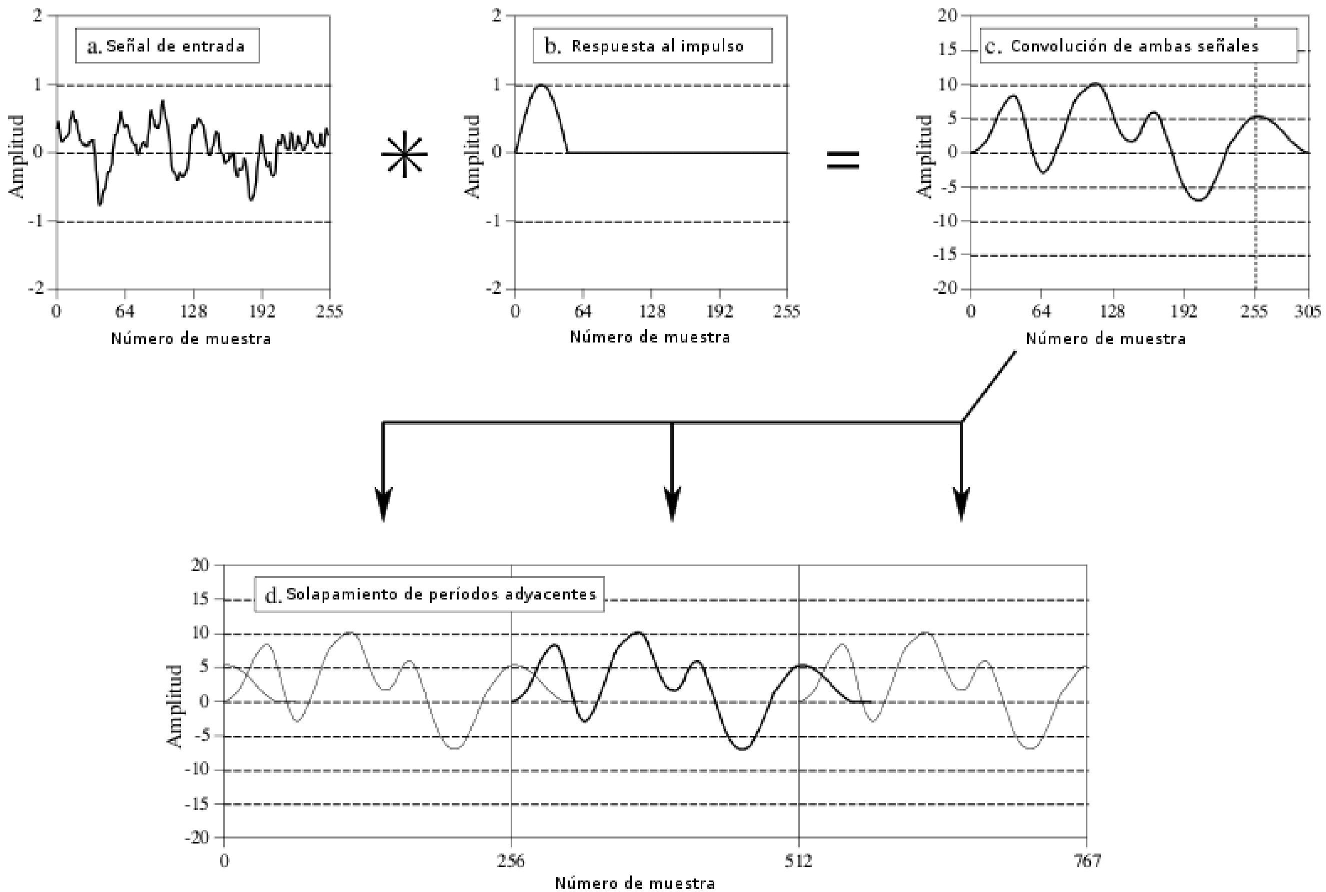
# Convolución circular

Señal de **N** puntos y respuesta al impulso de **M** puntos: señal resultante de **N+M-1** puntos.

Ejemplo,
N = 256, M = 51,
N+M-1 = 306.

Qué ocurre si se usan DFTs de 256 puntos?

La señal se expande de un período a otro, distorsionando el resultado correcto.



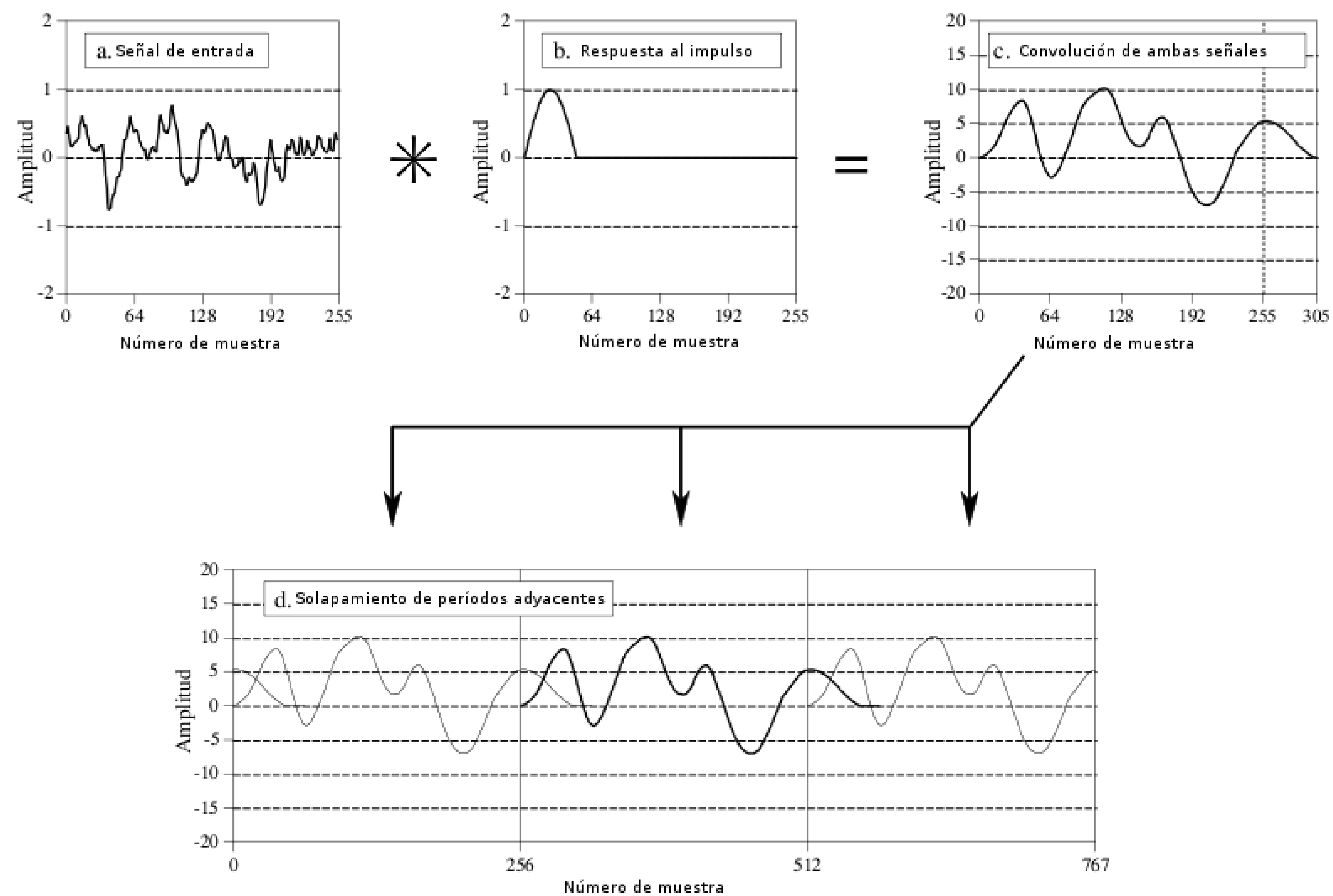
# Convolución circular

Señal de **N** puntos y respuesta al impulso de **M** puntos: señal resultante de **N+M-1** puntos.

Ejemplo,
N = 256, M = 51,
N+M-1 = 306.

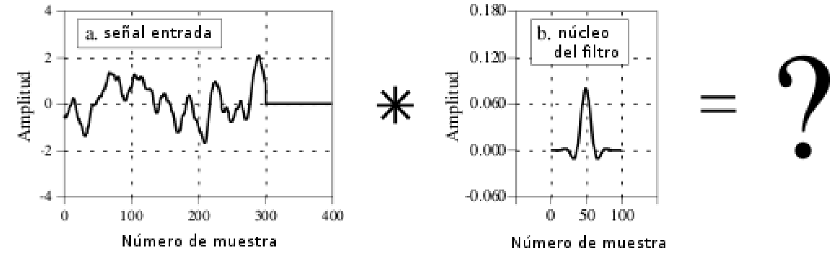
Qué ocurre si se usan DFTs de 256 puntos?

La señal se expande de un período a otro, distorsionando el resultado correcto.




**El agregado inicial de suficientes ceros para contener N+M-1 puntos soluciona el problema.**

# Método overlap-add

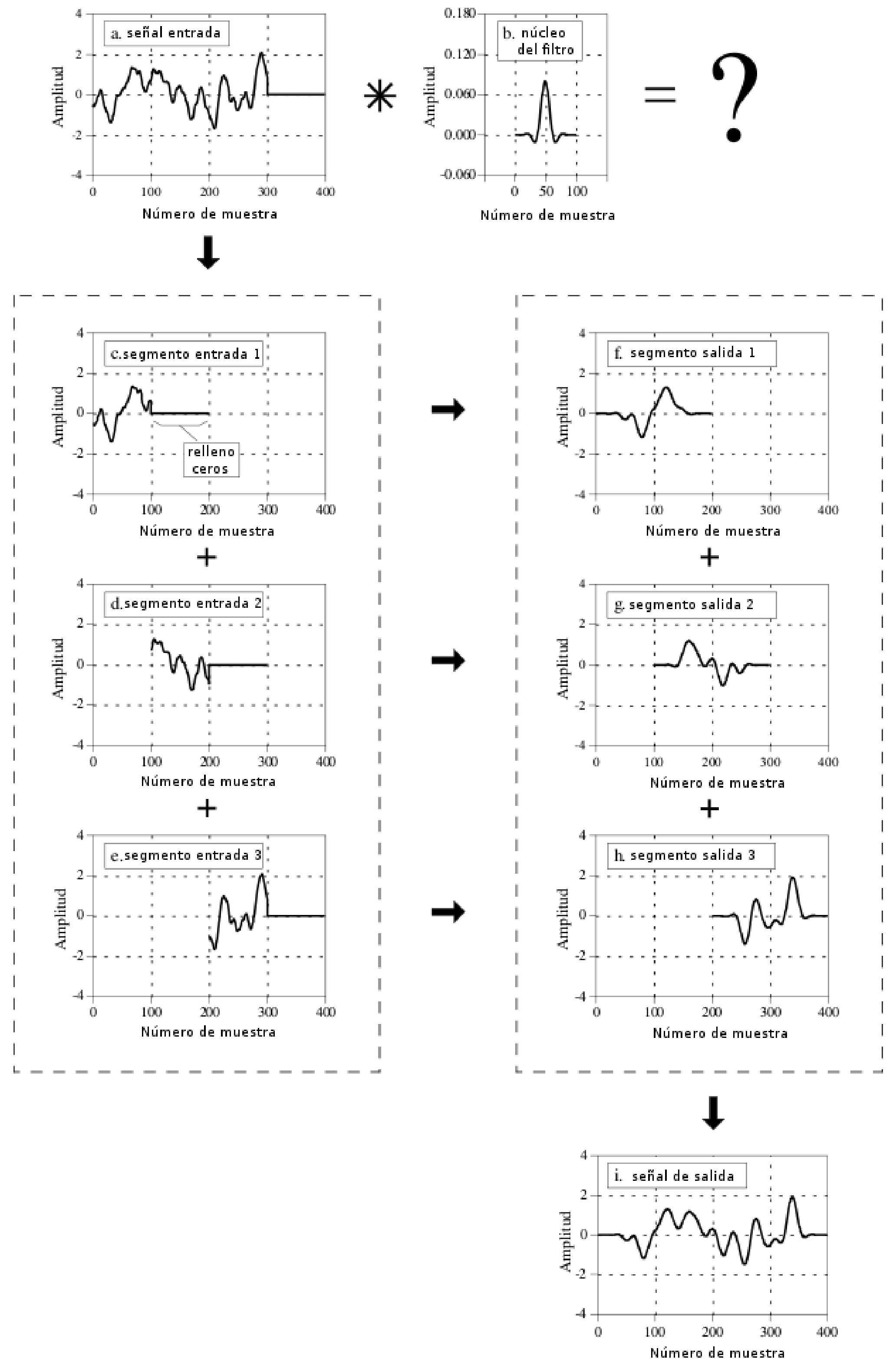



Procesar señal larga en segmentos
(e.g. tiempo real, memoria insuficiente)

# Método overlap-add

Procesar señal larga en segmentos
(e.g. tiempo real, memoria insuficiente)

1. descomponer señal en segmentos
2. procesar independientemente c/u
3. combinar resultado en señal final

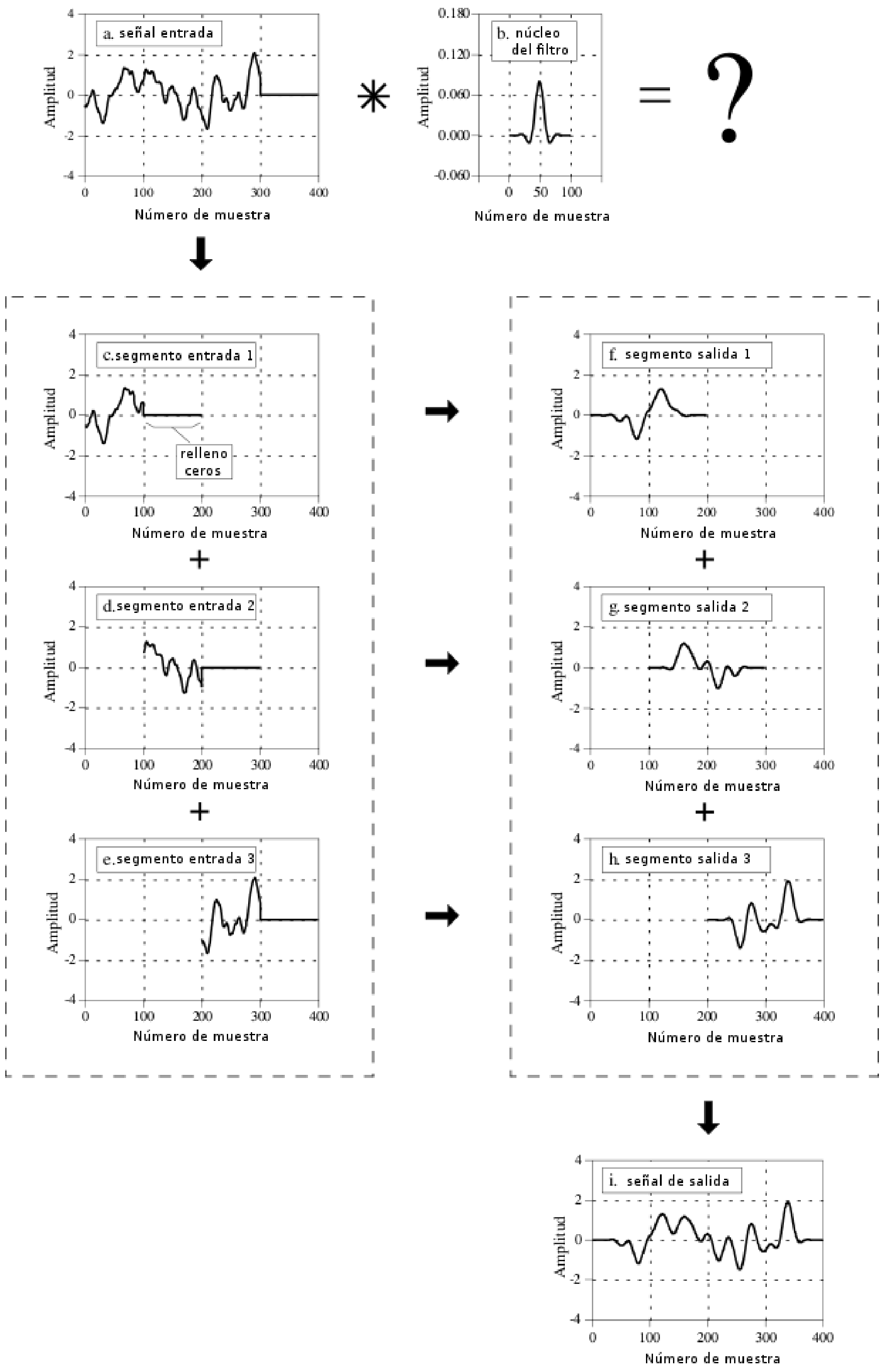


# Método overlap-add

Procesar señal larga en segmentos
(e.g. tiempo real, memoria insuficiente)

1. descomponer señal en segmentos
2. procesar independientemente c/u
3. combinar resultado en señal final

Tener en cuenta el largo resultante:
N_total = N + M – 1
Cada segmento se rellena con ceros:
N_segmento = n + M – 1



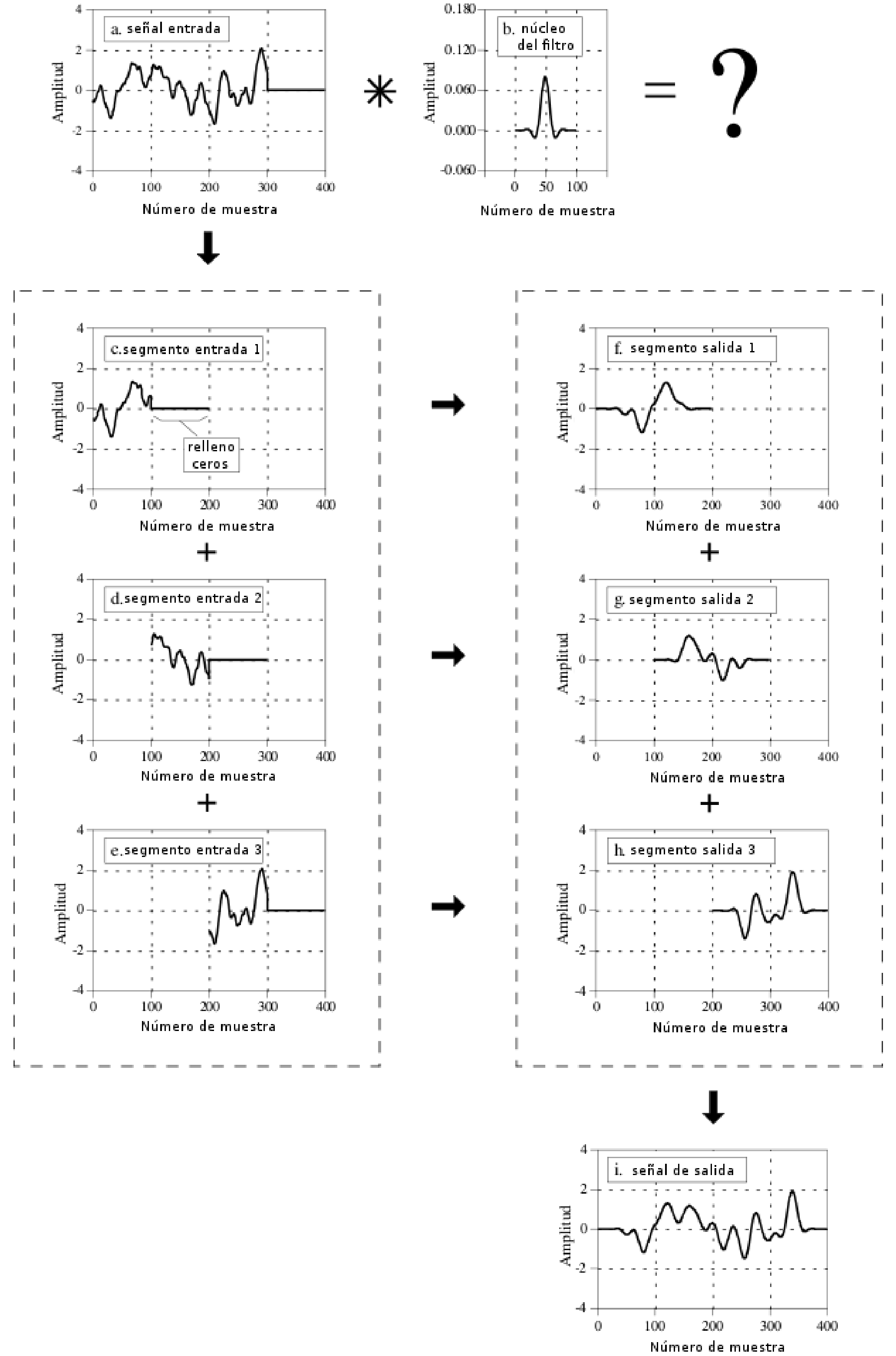
# Método overlap-add

Procesar señal larga en segmentos
(e.g. tiempo real, memoria insuficiente)

1. descomponer señal en segmentos
2. procesar independientemente c/u
3. combinar resultado en señal final

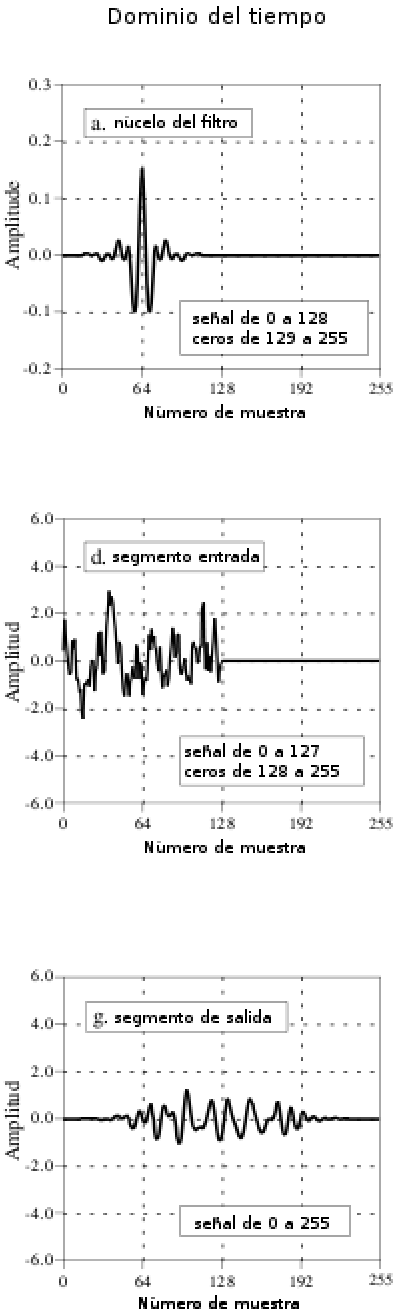
Tener en cuenta el largo resultante:
 N_total = N + M – 1
Cada segmento se rellena con ceros:
 N_segmento = n + M – 1

El resultado de procesar cada segmento se solapa con el resultado de procesar los segmentos adyacentes y la suma produce la señal de salida.



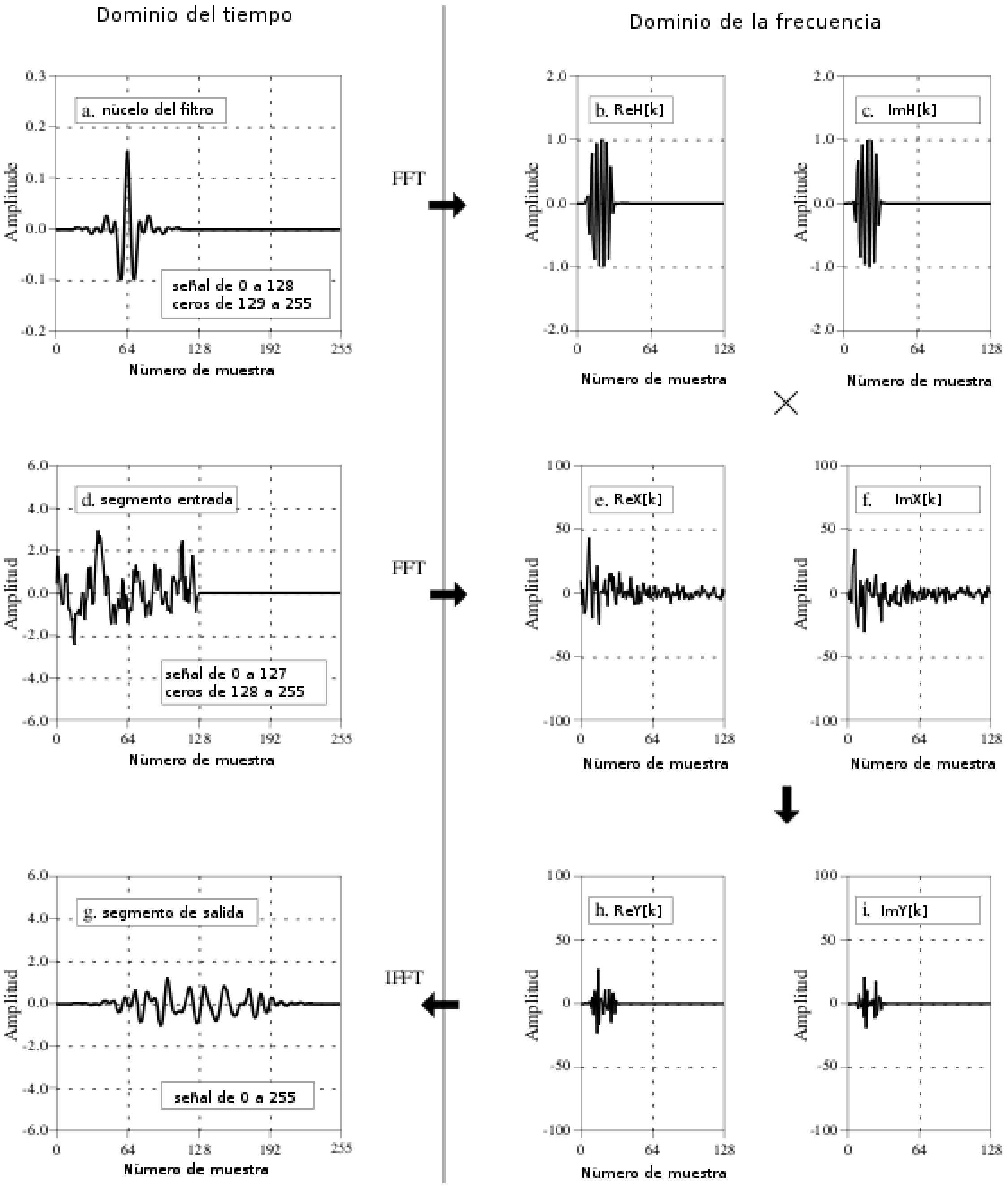
# Convolución rápida

La convolución rápida hace uso del método overlap-add y de la FFT para calcular espectros y antitransformar.



# Convolución rápida

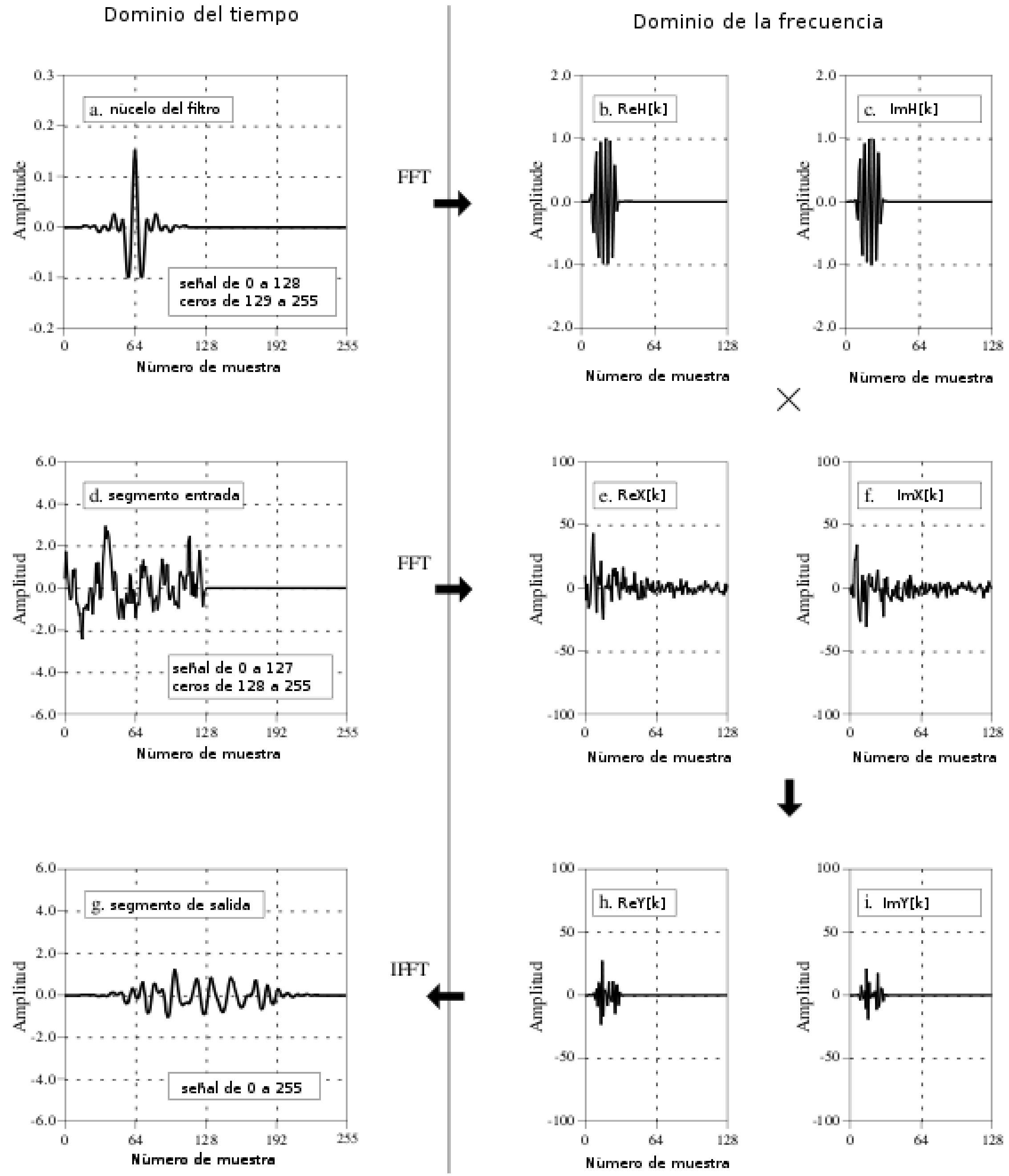
La convolución rápida hace uso del método overlap-add y de la FFT para calcular espectros y antitransformar.



# Convolución rápida

La convolución rápida hace uso del método overlap-add y de la FFT para calcular espectros y antitransformar.

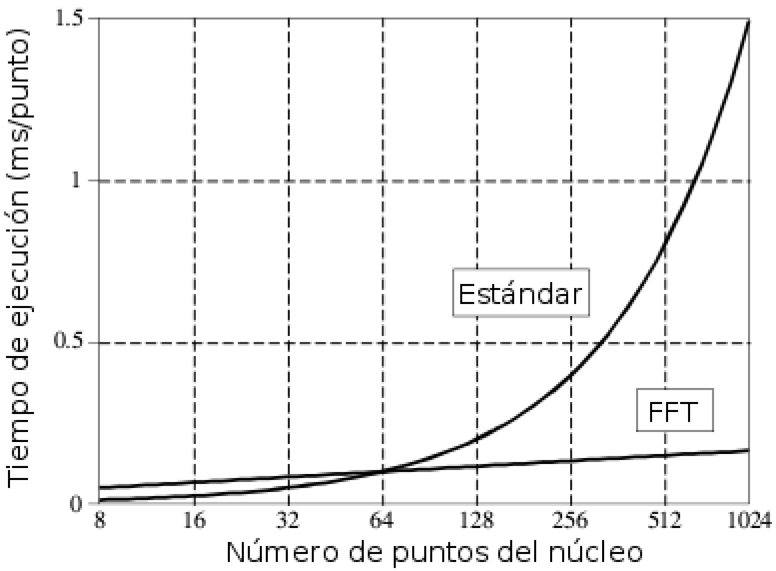
El largo de bloque de la FFT debe ser lo suficientemente largo para que no se produzca la convolución circular.

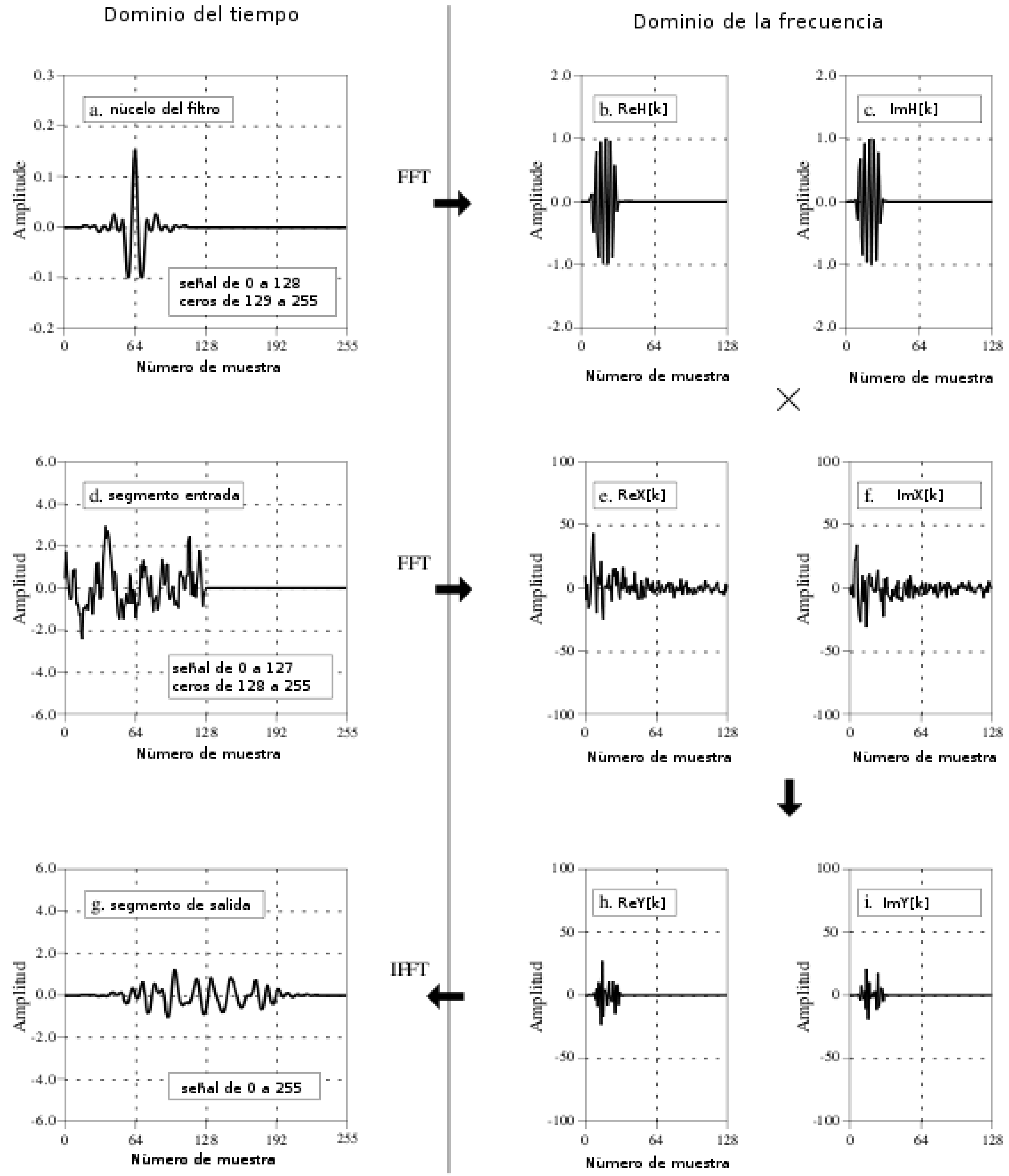


# Convolución rápida

La convolución rápida hace uso del método overlap-add y de la FFT para calcular espectros y antitransformar.

Cuándo esto es más rápido que la convolución estándar?



# DFT compleja

La DFT real:

$$Re\,X[k] = \frac{2}{N}\sum_{n=0}^{N-1} x[n]\cos(2\pi kn/N)$$

$$Im\,X[k] = \frac{-2}{N}\sum_{n=0}^{N-1} x[n]\sin(2\pi kn/N)$$

ReX[k] – número real
ImX[k] – número real

# DFT compleja

La DFT real:

$$Re\,X[k] = \frac{2}{N}\sum_{n=0}^{N-1} x[n]\cos(2\pi kn/N)$$

ReX[k] – número real
ImX[k] – número real

$$Im\,X[k] = \frac{-2}{N}\sum_{n=0}^{N-1} x[n]\sin(2\pi kn/N)$$

Desventajas:

- no representa frecuencias negativas
- maneja como excepción ReX[0] y ReX[N/2]
- menos elegante

# DFT compleja

La DFT real:

$$Re\,X[k] = \frac{2}{N}\sum_{n=0}^{N-1} x[n]\cos(2\pi kn/N)$$

ReX[k] – número real
ImX[k] – número real

$$Im\,X[k] = \frac{-2}{N}\sum_{n=0}^{N-1} x[n]\sin(2\pi kn/N)$$

Desventajas:

$$e^{jx} = \cos(x) + j\sin(x)$$

- no representa frecuencias negativas
- maneja como excepción ReX[0] y ReX[N/2]
- menos elegante

# DFT compleja

La DFT real:

$$Re\,X[k] = \frac{2}{N}\sum_{n=0}^{N-1} x[n]\cos(2\pi kn/N)$$

ReX[k] – número real
ImX[k] – número real

$$Im\,X[k] = \frac{-2}{N}\sum_{n=0}^{N-1} x[n]\sin(2\pi kn/N)$$

Desventajas:

- no representa frecuencias negativas
- maneja como excepción ReX[0] y ReX[N/2]
- menos elegante

$$e^{jx} = \cos(x) + j\sin(x)$$

$$\cos(x) = \frac{e^{jx} + e^{-jx}}{2} \qquad \sin(x) = \frac{e^{jx} - e^{-jx}}{2j}$$

# DFT compleja

La DFT real:

$$Re\,X[k] = \frac{2}{N}\sum_{n=0}^{N-1} x[n]\cos(2\pi kn/N)$$

$$Im\,X[k] = \frac{-2}{N}\sum_{n=0}^{N-1} x[n]\sin(2\pi kn/N)$$

ReX[k] – número real
ImX[k] – número real

Desventajas:

- no representa frecuencias negativas
- maneja como excepción ReX[0] y ReX[N/2]
- menos elegante

$$e^{jx} = \cos(x) + j\sin(x)$$

$$\cos(x) = \frac{e^{jx} + e^{-jx}}{2} \qquad \sin(x) = \frac{e^{jx} - e^{-jx}}{2j}$$

$$\cos(\omega t) = \frac{1}{2}e^{j(-\omega)t} + \frac{1}{2}e^{j\omega t}$$

$$\sin(\omega t) = \frac{1}{2}je^{j(-\omega)t} - \frac{1}{2}je^{j\omega t}$$

# DFT compleja

La DFT compleja:

$$X[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]\, e^{-j2\pi kn/N}$$

x[n] – número complejo
X[k] – número complejo

# DFT compleja

La DFT compleja:

$$X[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]\, e^{-j2\pi kn/N}$$

x[n] – número complejo
X[k] – número complejo

Usando Euler:

$$X[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]\left(\cos(2\pi kn/N) - j\sin(2\pi kn/N)\right)$$

# DFT compleja

La DFT compleja:

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

x[n] – número complejo
X[k] – número complejo

Usando Euler:

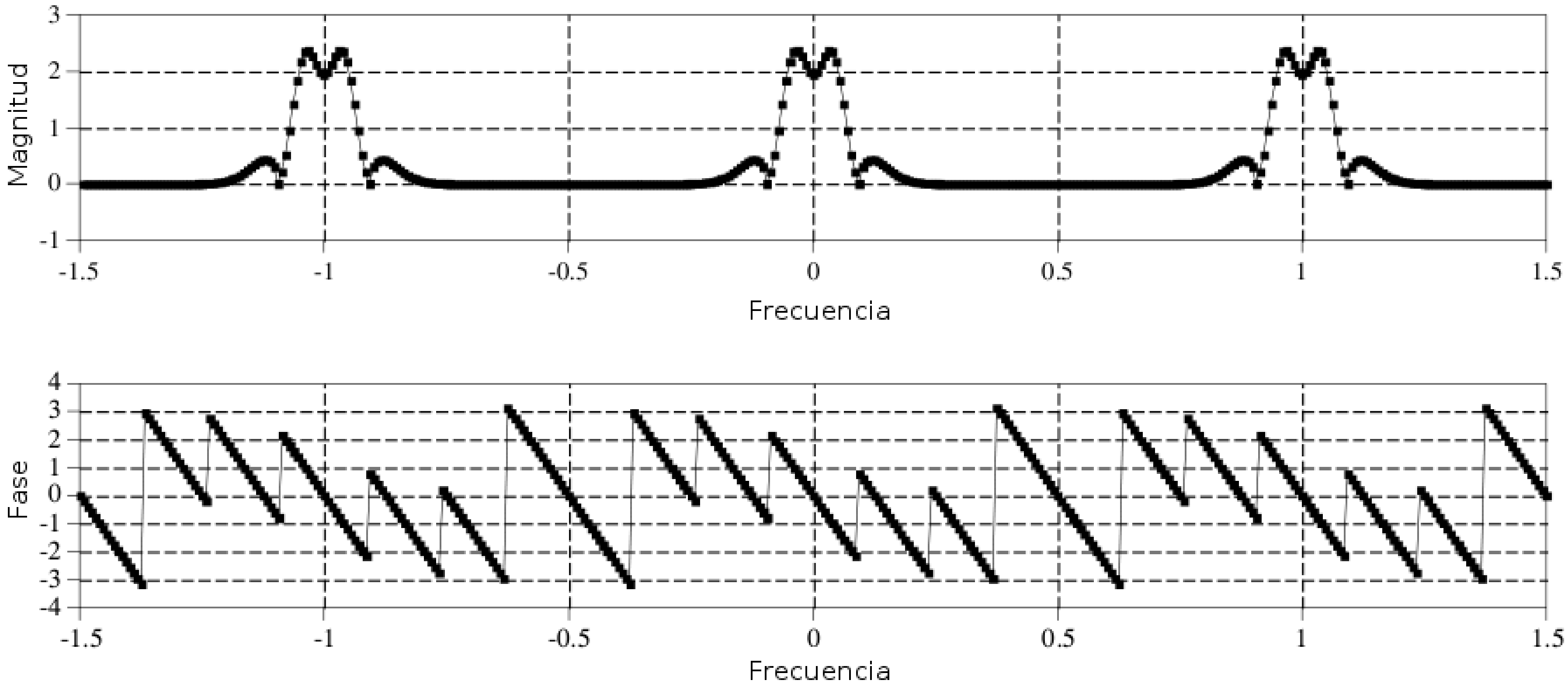
$$X[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] \left( \cos(2\pi k n / N) - j \sin(2\pi k n / N) \right)$$

Ventajas:
representa frecuencias negativas (k va de 0 a N-1)

frecuencias de 0 a N/2: positivas

frecuencias de N/2+1 a N: negativas

# DFT compleja



# Transformada rápida de Fourier (FFT)

1965 – Cooley & Tukey

Algoritmo eficiente para el cálculo de la DFT.

Se basa en descomponer el cálculo en transformadas de menos puntos.

$$X[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]\, e^{-j2\pi kn/N}$$

# Transformada rápida de Fourier (FFT)

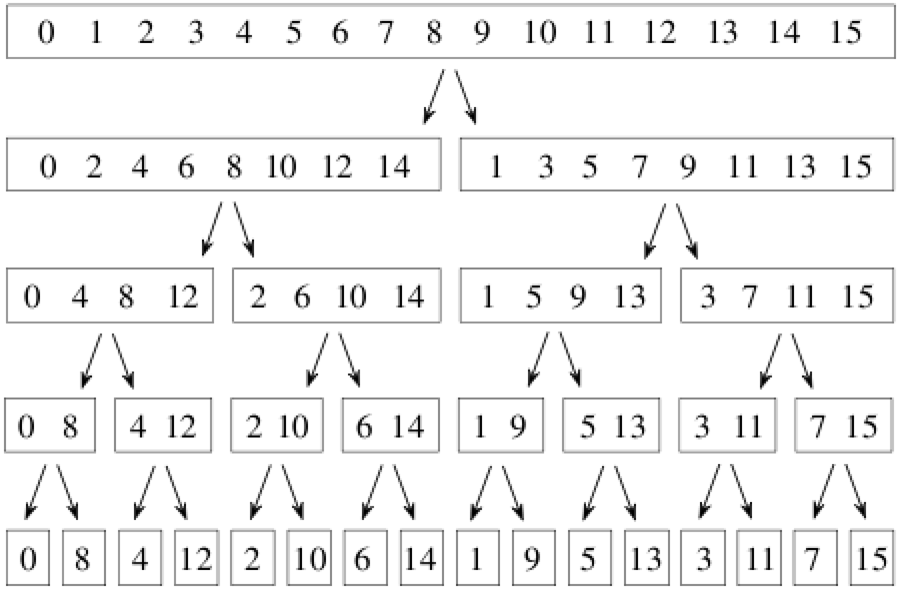
1965 – Cooley & Tukey

Algoritmo eficiente para el cálculo de la DFT.

$$X[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]\, e^{-j2\pi kn/N}$$

Se basa en descomponer el cálculo en transformadas de menos puntos.

Más precisamente, descompone una señal en el tiempo de N puntos en N señales de 1 punto cada una. Luego se calcula el espectro de cada señal y se los combina para obtener el espectro de la señal original.



$$X[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}$$

$$X[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}$$

considerando $N = 2^n$, puede escribirse $N = 2M$

$$X[k] = \frac{1}{2M} \sum_{n=0}^{2M-1} x[n] e^{-j2\pi kn/(2M)}$$

$$X[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N}$$

considerando $N = 2^n$, puede escribirse $N = 2M$

$$X[k] = \frac{1}{2M}\sum_{n=0}^{2M-1} x[n]e^{-j2\pi kn/(2M)}$$

lo que es equivalente a:

$$X[k] = \frac{1}{2}\left[\frac{1}{M}\sum_{n=0}^{M-1} x[2n]e^{-j2\pi k(2n)/(2M)} + \frac{1}{M}\sum_{n=0}^{M-1} x[2n+1]e^{-j2\pi k(2n+1)/(2M)}\right]$$

$$X[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}$$

considerando $N = 2^n$, puede escribirse $N = 2M$

$$X[k] = \frac{1}{2M} \sum_{n=0}^{2M-1} x[n] e^{-j2\pi kn/(2M)}$$

lo que es equivalente a:

$$X[k] = \frac{1}{2} \left[ \frac{1}{M} \sum_{n=0}^{M-1} x[2n] e^{-j2\pi k(2n)/(2M)} + \frac{1}{M} \sum_{n=0}^{M-1} x[2n+1] e^{-j2\pi k(2n+1)/(2M)} \right]$$

usando que:

$$e^{-j2\pi k(2n)/(2M)} = e^{-j2\pi kn/M}$$

$$e^{-j2\pi k(2n+1)/(2M)} = e^{-j2\pi kn/M} e^{-j2\pi k/(2M)}$$

$$X[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N}$$

considerando $N = 2^n$, puede escribirse $N = 2M$

$$X[k] = \frac{1}{2M}\sum_{n=0}^{2M-1} x[n]e^{-j2\pi kn/(2M)}$$

lo que es equivalente a:

$$X[k] = \frac{1}{2}\left[\frac{1}{M}\sum_{n=0}^{M-1} x[2n]e^{-j2\pi k(2n)/(2M)} + \frac{1}{M}\sum_{n=0}^{M-1} x[2n+1]e^{-j2\pi k(2n+1)/(2M)}\right]$$

usando que:

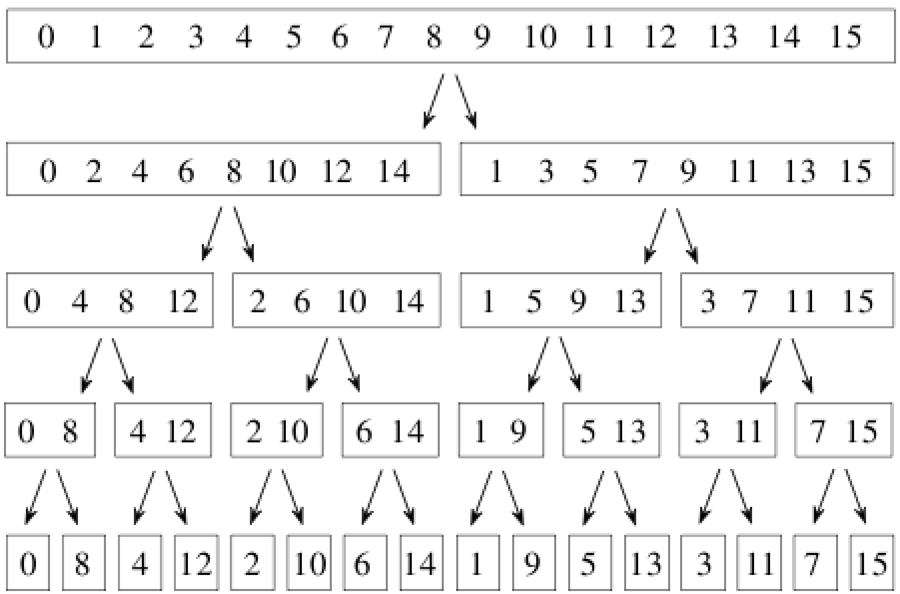
$$e^{-j2\pi k(2n)/(2M)} = e^{-j2\pi kn/M}$$

$$e^{-j2\pi k(2n+1)/(2M)} = e^{-j2\pi kn/M}e^{-j2\pi k/(2M)}$$

se obtiene:

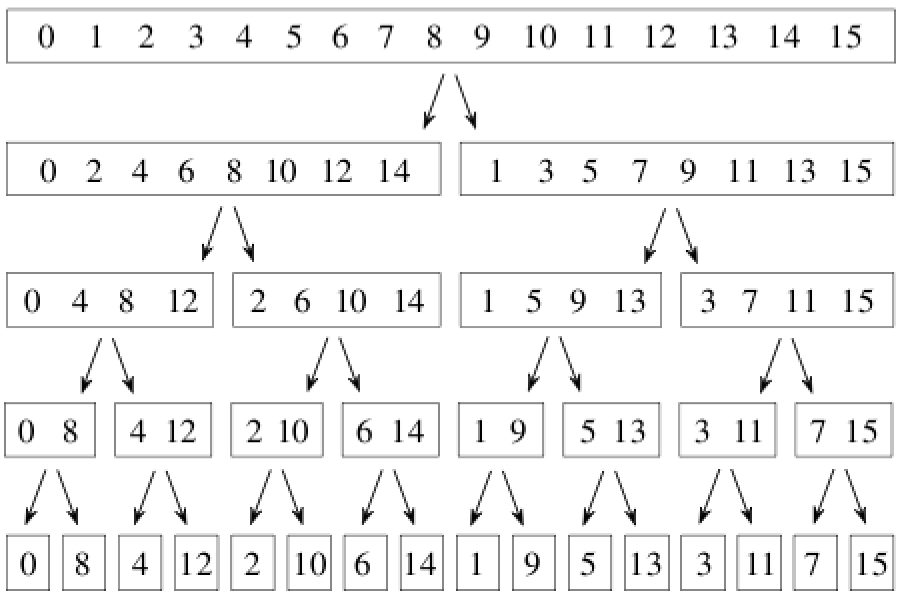
$$X[k] = \frac{1}{2}\left[\frac{1}{M}\sum_{n=0}^{M-1} x[2n]e^{-j2\pi kn/M} + \frac{1}{M}\sum_{n=0}^{M-1} x[2n+1]e^{-j2\pi kn/M}e^{-j2\pi k/(2M)}\right]$$

# Transformada rápida de Fourier (FFT)




La descomposición en el tiempo se denomina interlaceado y corresponde a un ordenamiento de las muestras.
Involucra log2(N) niveles, e.g. 2^4 = 16 -> 4 niveles.

# Transformada rápida de Fourier (FFT)




| | | | |
|---|---|---|---|
| 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 |

La descomposición en el tiempo se denomina interlaceado y corresponde a un ordenamiento de las muestras.
Involucra log2(N) niveles, e.g. 2^4 = 16 -> 4 niveles.

Lo interesante es que puede calcularse en representación binaria de forma muy eficiente: mediante el reverso de cada número.

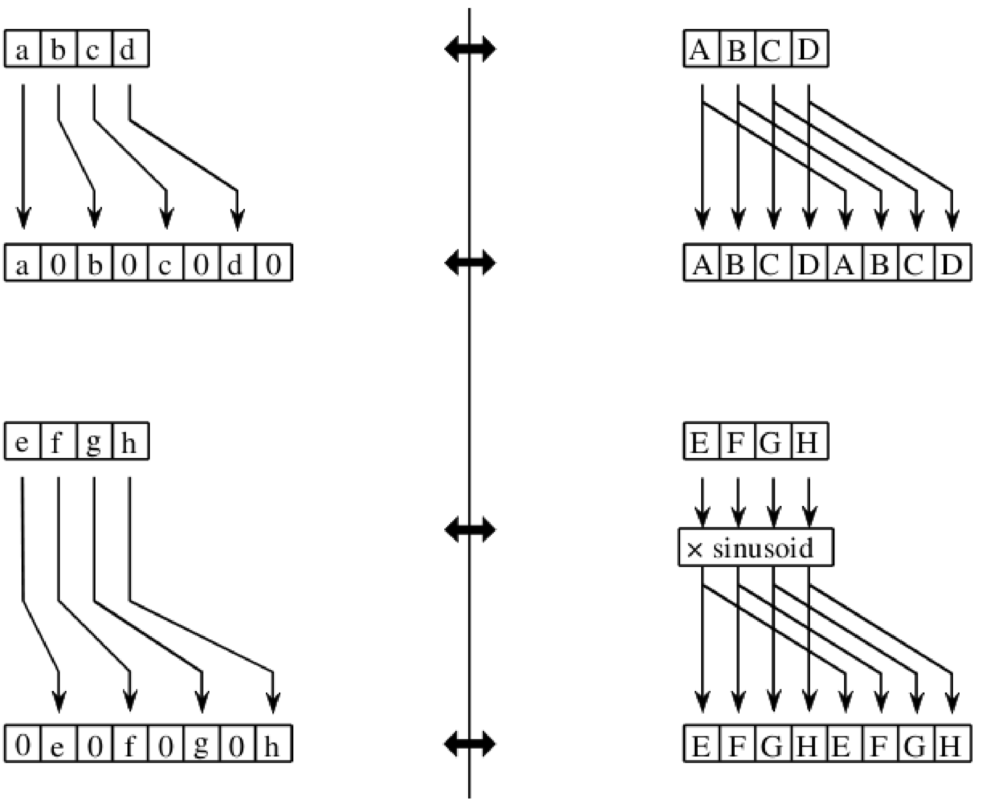
# Transformada rápida de Fourier (FFT)

Una vez hecha la descomposición, el calculo de la DFT de cada punto es trivial.

Luego queda combinar las transformadas, pero teniendo en cuenta las operaciones que se hicieron durante el ordenamiento.

El intercalado de ceros corresponde a una repetición del espectro.

El corrimiento temporal corresponde a multiplicar el espectro por una sinusoide.

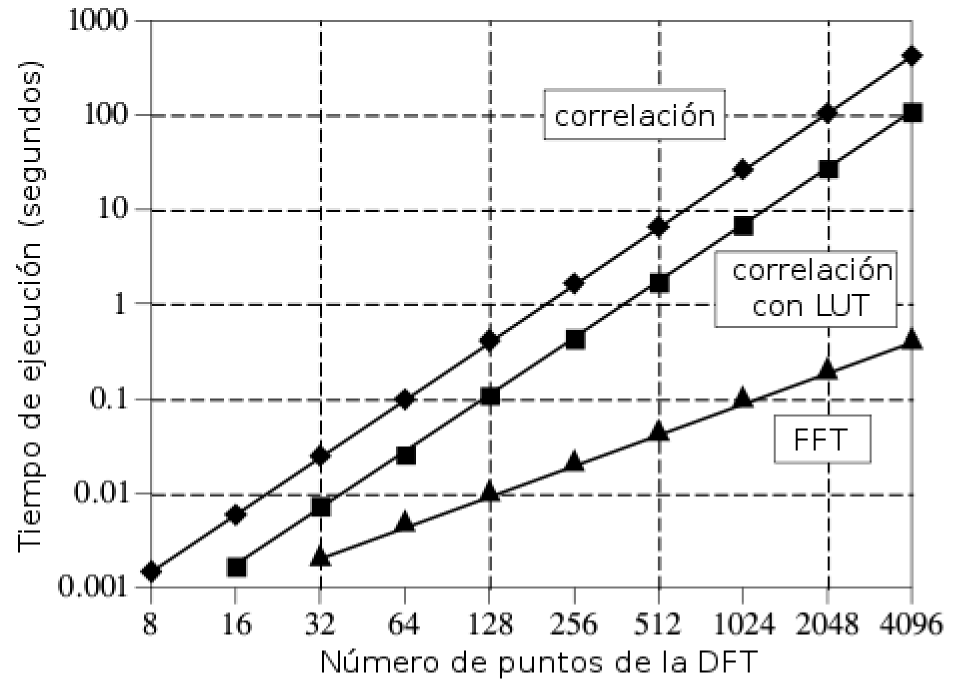


# Transformada rápida de Fourier (FFT)

Desempeño y precisión:

DFT: orden de N^2
FFT: orden de N log2(N)

A medida que N crece se hace cientos de veces más rápida.



# Transformada rápida de Fourier (FFT)

Desempeño y precisión:

DFT: orden de N^2
FFT: orden de N log2(N)

A medida que N crece se hace cientos de veces más rápida.

Un aspecto importante es que un menor número de operaciones redunda en menor error de redondeo, por lo que el cálculo usando la FFT es además más preciso.