

TRANSFORMADA DISCRETA DE FOURIER

OBJETIVOS:

- Migrar desde la Transformada Continua de Fourier, la cual sirve fundamentalmente desde un contexto puramente matemático, a la Transformada Discreta de Fourier, que es la adecuada para trabajar con computadoras digitales.
- Resolver la Transformada (y Transformada Inversa) Discreta de señales que pueden ser fácilmente tratadas con la Transformada Continua. Esto es para establecer a esta última como "patrón" de comparación de los resultados obtenidos con la Discreta y marcar pautas a tener en cuenta respecto de los errores que se pueden cometer.
- Generalizar los conceptos anteriores para la resolución de problemas de Análisis y Síntesis Armónicos. Es decir, la aplicación de la Transformada Discreta de Fourier a señales periódicas.
- Evaluar el fenómeno del "leakage" que aparece cuando se "trunca" una señal periódica en un valor distinto de su periodo. Y su atenuación con el uso de "ventanas" adecuadas.

TRANSFORMADA DISCRETA DE FOURIER

La Transformada de Fourier tal como se ha visto en el capítulo I, se aplica a funciones continuas del tiempo. Además, implica el cálculo de una integración con extremos infinitos que no en todos los casos resulta fácil de resolver.

Dado que a través de las computadoras digitales, que abundan como herramientas de proceso en todos los ámbitos académicos, se puede abordar esta temática con muy buenos resultados, es que se utiliza una variante que es la Transformada Discreta de Fourier (DFT).

La idea a seguir a través de este Capítulo es que, si se conoce el resultado de la aplicación de la Transformada Continua de Fourier a una función del tiempo (que se podría calificar de "buena", ya que se puede resolver la integral en forma directa), a esa misma señal, "muestreada", se le aplica la Transformada Discreta de Fourier y se comparan los resultados con los de la Continua.

Si a una señal continua en el tiempo se la muestrea con una cierta tasa, esto es se la discretiza en el tiempo, es posible hallar su **Transformada Discreta de Fourier** mediante la siguiente expresión:

$$\frac{F_n}{N \cdot T} = \sum_{k=0}^{N-1} f_{kT} \cdot \exp\left(-j \cdot 2 \cdot \frac{\pi}{N} \cdot k \cdot n\right) \quad (18-1)$$

con **k** y **n** variando entre 0 y N-1, siendo **N** el número de muestras tomadas de la señal.

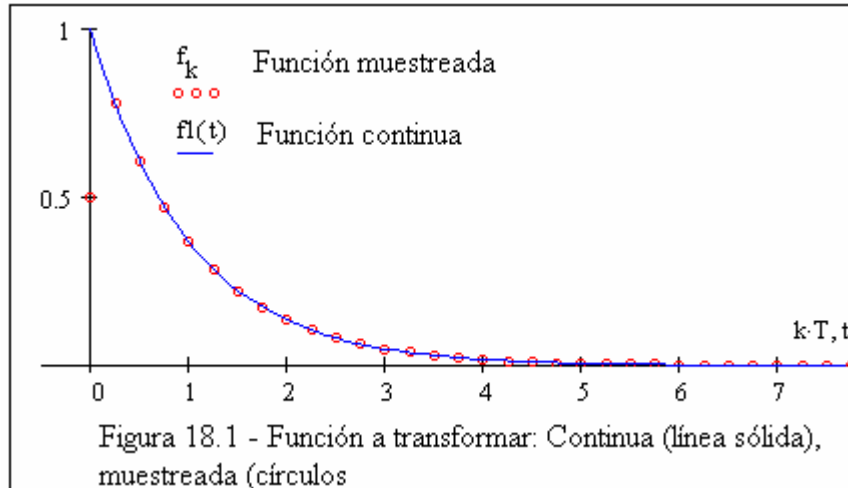
A la función ya muestreada se le llama Secuencia y consiste de N valores.

Para ilustrar la aplicación de la DFT al cómputo de las Transformadas de Fourier, considérese la figura 18.1.

$$N := 32 \quad k := 0 \dots N - 1 \quad T := 0.25$$

$$f_k := \exp(-k \cdot T) \quad f_0 := 0.5$$

$$f_l(t) := \exp(-t) \quad t := 0, 0.1 \dots (N-1) \cdot T$$



Se desea computar por medio de la DFT una aproximación a la Transformada de Fourier de esta función.

El primer paso en la aplicación de la transformada discreta es elegir el número de muestras N y el intervalo de muestreo T . Para $N=32$ y $T=0.25$ se observan las muestras de $\exp(-t)$. Nótese que se ha definido la muestra en el valor $t=0$ de modo de cumplir con la condición que **el valor de la función en una discontinuidad debe ser definido como el valor medio.** [ver Capítulo, expresión (7-3)]

Se computa la DFT como:

$$H\left(\frac{n}{N \cdot T}\right) = T \cdot \sum_{k=0}^{N-1} f_k \cdot \exp\left(\frac{-j \cdot 2 \cdot \pi \cdot n \cdot k}{N}\right) \quad \text{con} \quad n := 0, 1 \dots N-1$$

Se variará ligeramente la notación vista en (18-1), de modo que los subíndices de la Secuencia de la DFT tengan valores enteros positivos. Sin embargo, no se debe perder de vista que el valor discreto $n/(N \cdot T)$ es la referencia en frecuencia de cada elemento del vector que contiene la DFT.

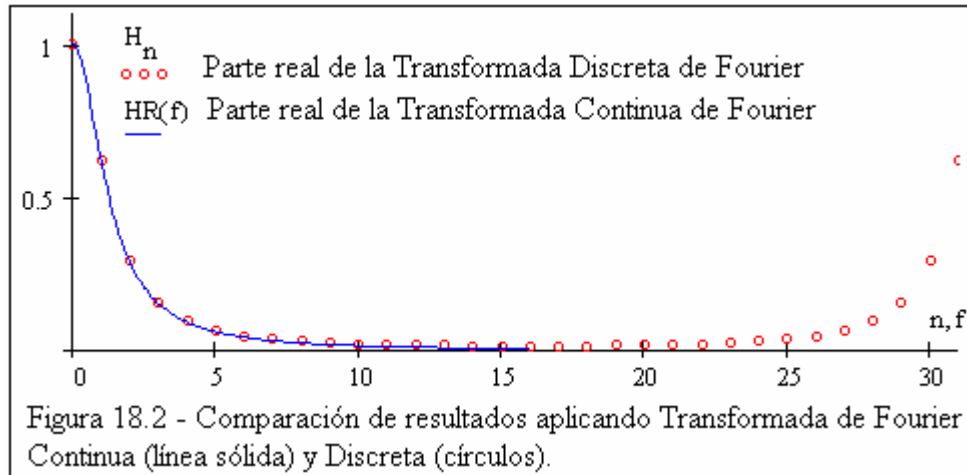
$$H_n := T \cdot \sum_{k=0}^{N-1} f_k \cdot \exp\left(\frac{-j \cdot 2 \cdot \pi \cdot n \cdot k}{N}\right)$$

Resuelta como la Transformada Continua de Fourier, se debe operar como en el ejemplo del párrafo 6 del Capítulo I, cuyos resultados daban la siguiente función como resultado.

$$H(f) = \frac{1}{1 + j \cdot \omega} = \frac{1}{1 + j \cdot 2 \cdot \pi \cdot f} = \frac{1}{1 + (2 \cdot \pi \cdot f)^2} - j \cdot \frac{2 \cdot \pi \cdot f}{1 + (2 \cdot \pi \cdot f)^2}$$

La cual consta de una parte real dada por:

$$HR(f) := \frac{1}{1 + \left(2 \cdot \pi \cdot \frac{f}{N \cdot T}\right)^2} \quad f := 0, 0.1 \dots \frac{N}{2}$$



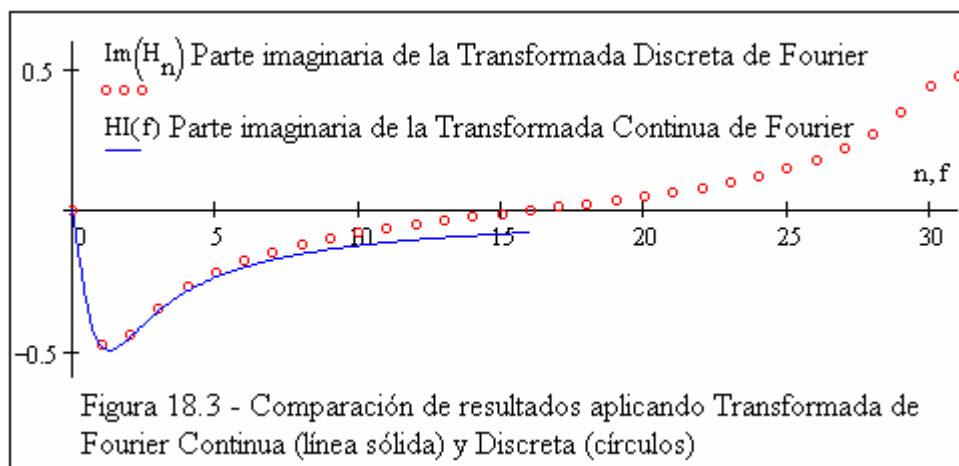
Obsérvese que el factor de escala T se introduce para producir equivalencia entre las transformadas continua y discreta. Nótese que no se ha indicado explícitamente $\text{Re}(H_n)$, ya que Mathcad representa gráficamente sólo la parte real.

De la observación de la figura 18.2 se ve que la TDF es simétrica respecto de $n = N/2$. Esto se deriva de la propiedad vista en el Capítulo I (punto 8.1.1) que dice que la parte real de la transformada de Fourier de una función real es Par.

Los resultados para $n > N/2$ son simplemente valores correspondientes a frecuencias negativas.

Analizando la parte imaginaria.

$$HI(f) := \frac{-\left(2 \cdot \pi \cdot \frac{f}{N \cdot T}\right)}{1 + \left(2 \cdot \pi \cdot \frac{f}{N \cdot T}\right)^2}$$



En la Figura 18.3 se ilustra la parte imaginaria de la DFT y de la CFT. Como se ve, la DFT se aproxima más burdamente a la CFT para las altas frecuencias. Para reducir este error es necesario incrementar la frecuencia de muestreo (reducir T) y N .

Nótese que la TDF en este caso es anti-simétrica respecto de $n = N/2$. Esto se deriva de que la parte imaginaria de la transformada de Fourier de una función real es Impar (Capítulo I, propiedad 8.1.2)

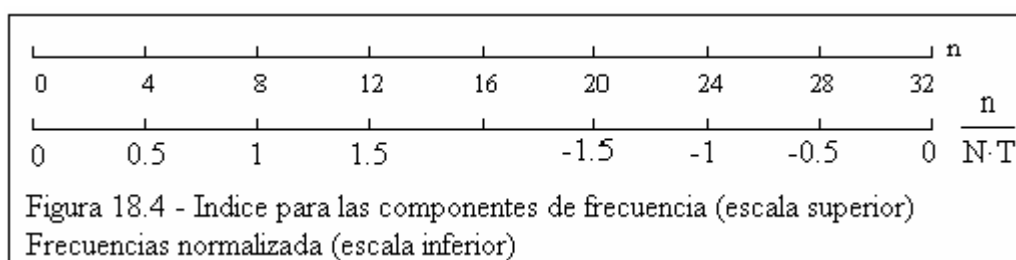
También en este caso, los resultados para $n > N/2$ son simplemente valores correspondientes a frecuencias negativas.

Este último punto es enfatizado representando una verdadera escala de frecuencia debajo de la escala para el parámetro n .

Se podrían haber graficado los datos de la forma convencionalmente usada para representar la transformada continua; esto es, desde $-f_0$ a $+f_0$. Sin embargo, el método convencional de presentar los resultados de la DFT en forma gráfica es como función del parámetro n .

Mientras tanto, recuérdese que los resultados para $n > N/2$ se relacionan con frecuencias negativas.

El eje de abscisas se puede ver de dos maneras:



En resumen, aplicar la DFT para el cómputo de la Transformada de Fourier sólo requiere tener cuidado en la elección del T y del N e interpretar correctamente los resultados.

APROXIMACION A LA TRANSFORMADA INVERSA DE FOURIER

Supóngase están dadas las funciones continuas real e imaginaria de la frecuencia consideradas en la sección previa y que se desea determinar la función del tiempo correspondiente por medio de la transformada discreta de Fourier.

$$h(k \cdot T) = \Delta f \sum_{n=0}^{N-1} (R(n \cdot \Delta f) + j \cdot I(n \cdot \Delta f)) \cdot \exp\left(j \cdot 2 \cdot \pi \cdot k \cdot \frac{n}{N}\right) \quad \text{con } k=0 \dots N-1$$

donde Δf es el intervalo de muestreo en frecuencia.

Sea $HR(f)$ la Transformada Continua de Fourier de la exponencial decreciente tomada como ejemplo y $N = 32$ (Número de muestras) y $\Delta f = 1/8$ (separación entre muestra y muestra)..

Ya que se sabe que $HR(f)$, la parte real de la función de frecuencia compleja, debe ser una función par, entonces se **dobla** (fold) sobre la frecuencia $f = 2.0$, la cual corresponde al punto muestra $n = N/2$.

Como se visualiza abajo, simplemente:

- Se muestrea la función de frecuencia hasta el punto $n = N/2$ (por comodidad se sigue hasta N) según se ve en la ecuación (19-1)
- y luego se "doblan" estos valores alrededor de $n = N/2$ (superponiéndose, en este caso a los calculados en el paso anterior) según se ve en la ecuación (19-2) para obtener las muestras restantes.

$$N := 32 \quad \Delta f := \frac{1}{8} \quad n := 0 \dots N-1$$

$$R_n := \frac{1}{1 + (2 \cdot \pi \cdot n \cdot \Delta f)^2} \quad \text{muestreado} \quad (19-1)$$

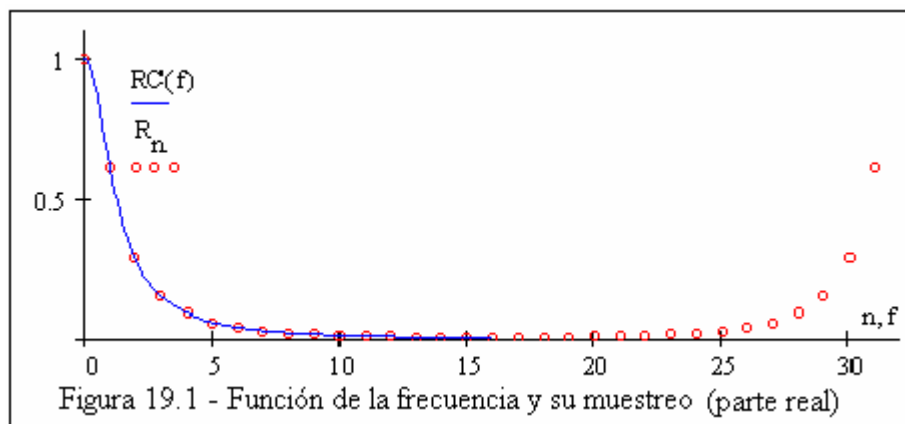
$$R_{N-n} := R_n \quad \text{"doblado (fold)"} \quad (19-2)$$

La función continua es:

$$RC(f) := \frac{1}{1 + (2 \cdot \pi \cdot f \cdot \Delta f)^2}$$

$$f := 0, 1 \dots \frac{N}{2} \quad \text{rango}$$

Todo este proceso se puede visualizar en la Figura 19.1.

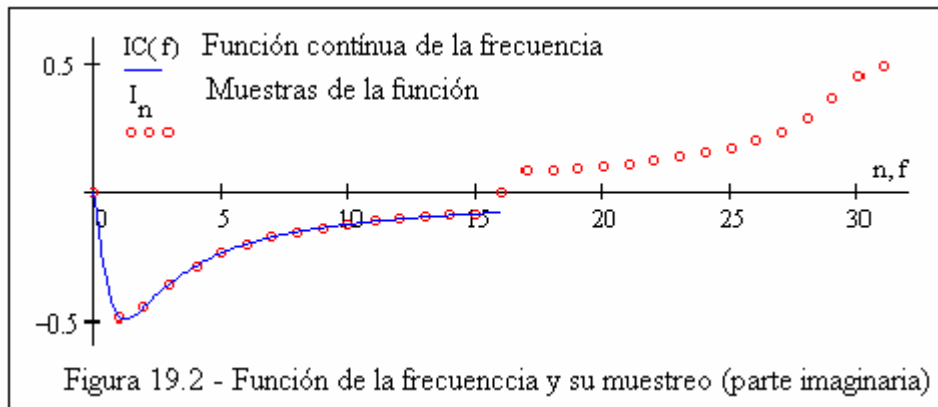


En Figura 19.2 se ilustra el método para determinar las N muestras de la parte imaginaria de la función de frecuencia. Debido a que esta es impar, se deben no sólo "doblar" alrededor de la muestra $N/2$ sino que también se deben "**flippear**" (girar alrededor del eje de abscisas) los resultados.

Para preservar la simetría se pone a cero la muestra en $n = N/2$.

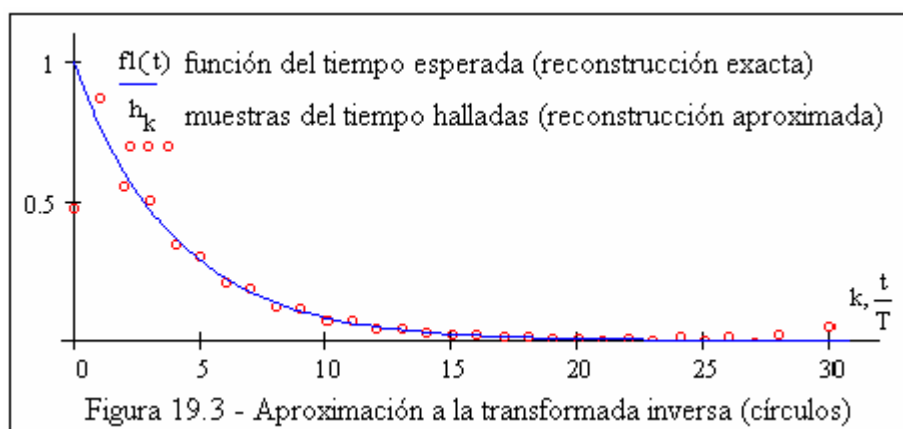
$$I_n := \frac{-2 \cdot \pi \cdot n \cdot \Delta f}{1 + (2 \cdot \pi \cdot n \cdot \Delta f)^2} \quad I_{N-n} := -I_n \quad \text{"flippeado"} \quad I_{\frac{N}{2}} := 0$$

$$IC(f) := \frac{-2 \cdot \pi \cdot f \Delta f}{1 + (2 \cdot \pi \cdot f \Delta f)^2}$$



El cómputo de la expresión $h(k.T)$ de la fórmula general con las funciones muestreadas como se ha visto en las Figuras 19.1 y 19.2, produce la IDFT (transformada discreta inversa de Fourier).

$$h_k := \Delta f \sum_{n=0}^{N-1} (R_n + j \cdot I_n) \cdot \exp\left(j \cdot 2 \cdot \pi \cdot k \cdot \frac{n}{N}\right)$$



El resultado es una función compleja cuya parte imaginaria es aproximadamente cero y cuya parte real se visualiza en la Figura 19.3.

Se nota que en $k=0$ el resultado es aproximadamente igual al valor medio correcto y se obtiene una concordancia razonable para los valores de k . Se pueden obtener mejoras reduciendo Δf e incrementando N .

La clave para usar la IDFT en la aproximación a un resultado continuo es especificar correctamente la función de frecuencia muestreada. Las figuras anteriores muestran ese método correcto.

Se puede observar que se requiere el factor de escala Δf , para dar una aproximación correcta a la CDFT.

Un resultado similar se obtiene usando la forma alternativa de la IDFT:

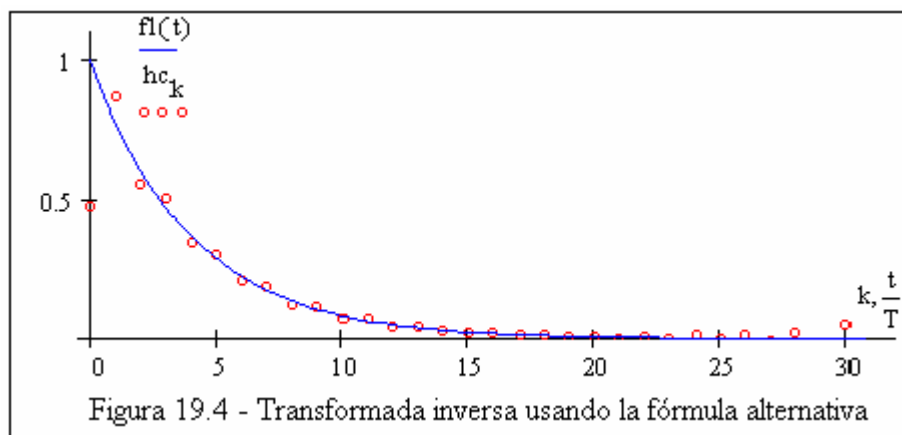
$$h(k) = \frac{1}{N} \sum_{n=0}^{N-1} \overline{H(n)} \cdot \exp\left(-j \cdot 2 \cdot \pi \cdot \frac{k}{N} \cdot n\right)$$

Para usar esta relación se conjuga en primera instancia la función de frecuencia compleja, esto es la función imaginaria muestreada se multiplica por -1.

Ya que la función resultante del tiempo es real, la operación conjugación final puede ser omitida. De aquí se computa:

$$h(k \cdot T) = \Delta f \sum_{n=0}^{N-1} (R(n \cdot \Delta f) + j \cdot (-1) \cdot I(n \cdot \Delta f)) \cdot \exp\left(-j \cdot 2 \cdot \pi \cdot k \cdot \frac{n}{N}\right)$$

$$hc_k := \Delta f \sum_{n=0}^{N-1} (R_n - j \cdot I_n) \cdot \exp\left[-j \cdot 2 \cdot \pi \cdot k \cdot \frac{n}{N}\right]$$



De la observación de la Figura 19.4, se ve que produce el mismo resultado.

ANÁLISIS ARMÓNICO DE LAS SERIES DE FOURIER

Las señales de tiempo continuo que presentan periodicidad, deben ser tratadas con las Series de Fourier, resultando de ello una secuencia infinita de coeficientes que afectan a senos y cosenos de frecuencias múltiples más un valor medio.

Si se aplican los procedimientos vistos hasta aquí, es posible obtener buenos resultados siempre y cuando se tomen los recaudos necesarios, los cuales se expondrán en esta sección.

La aplicación de la DFT al análisis armónico de una forma de onda requiere que se compute:

$$H\left(\frac{n}{N \cdot T}\right) = \frac{T}{N \cdot T} \cdot \sum_{k=0}^{N-1} h(k \cdot T) \cdot \exp\left(\frac{-j \cdot 2 \cdot \pi \cdot n \cdot k}{N}\right)$$

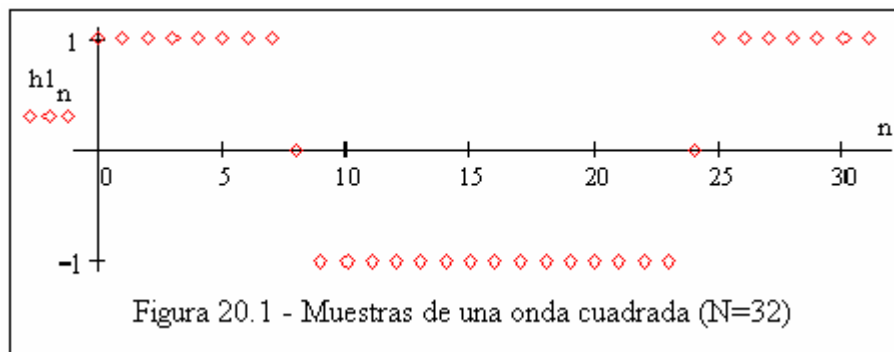
donde el divisor ($N \cdot T$) es la duración en tiempo o **periodo** de la armónica de frecuencia más baja a ser determinada.

Para producir resultados válidos, los valores de las N muestras deben representar exactamente un periodo de la función periódica $h(t)$.

Considérese la onda cuadrada ilustrada en la Figura 20.1, con un periodo de muestreo de 8 seg. De ese modo, si $N = 32$, T (tiempo entre muestra y muestra) debe ser elegido igual a 0.25 para asegurar que las 32 muestras sean exactamente igual a un periodo (periodo de la onda, 8 segundos).

$$N := 32 \quad T := 0.25 \quad T_p := N \cdot T \quad T_p = 8$$

$$h1_n := \text{if}\left(n < \frac{N}{4}, 1, \text{if}\left(n < 3 \cdot \frac{N}{4}, -1, 1\right)\right) \quad h1_{\frac{N}{4}} := 0 \quad h1_{\frac{3 \cdot N}{4}} := 0$$

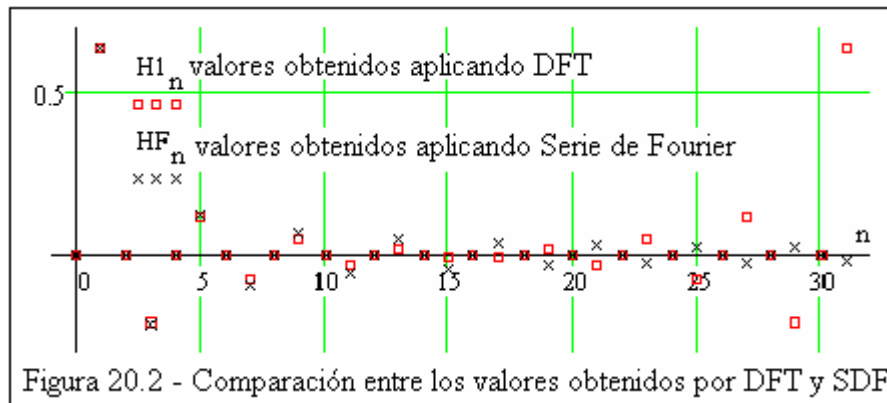


Obsérvese que para atender a las discontinuidades que se producen en $n=8$ y $n=24$, en esos valores se promedia la función.

Calculando la DFT con los procedimientos vistos:

$$H1_n := \frac{1}{N} \cdot \sum_{k=0}^{N-1} h1_k \cdot \exp\left(\frac{-j \cdot 2 \cdot \pi \cdot n \cdot k}{N}\right)$$

$$HF_n := \text{if}\left[n=0, 0, \frac{(-1)^{\frac{n-1}{2}}}{n \cdot \pi} \cdot [1 - (-1)^n]\right] \quad \text{desarrollo en serie de Fourier}$$



Las cruces representan las magnitudes de los coeficientes de armónicos obtenidos del desarrollo en serie de Fourier. Como se esperaba, los resultados son simétricos alrededor del punto $n = N/2$. Se obtienen resultados razonables para los armónicos de orden más bajo. La exactitud puede ser mejorada para los armónicos más altos achicando T e incrementado N .

Note que se ha introducido un aliasing significativo evidenciado por el hecho que los valores de los coeficientes verdaderos tienen magnitud apreciable en el número de muestra $n = N/2$.

SINTESIS DE ARMONICOS DE LAS SERIES DE FOURIER

La síntesis de armónicos se refiere al procedimiento de calcular una forma de onda periódica dados los coeficientes de la serie de Fourier. Para llevar a cabo esta tarea usando la DFT, simplemente se computa:

$$h(k \cdot T) = \Delta f \sum_{n=0}^{N-1} H(n \cdot \Delta f) \cdot \exp\left(j \cdot 2 \cdot \pi \cdot n \cdot \frac{k}{N}\right)$$

donde Δf debe ser elegido como un entero múltiplo de la armónica fundamental.

Para aplicar la expresión anterior se deben muestrear los coeficientes reales e imaginarios consistentes con los procedimientos discutidos previamente.

Si se considera el ejemplo previo, entonces sólo los coeficientes reales deben ser muestreados.

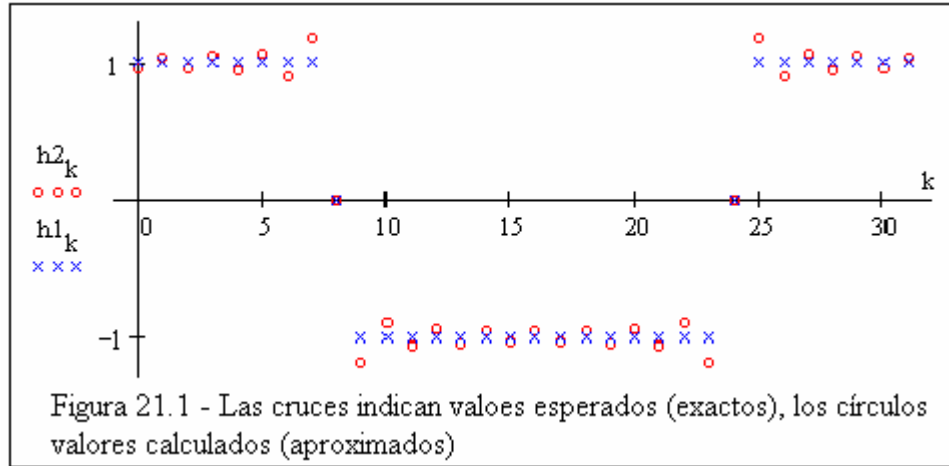
Por supuesto, estas muestras deben ser "dobladadas" alrededor del punto $N/2$. Nótese que se tiene de hecho truncada la serie de Fourier, porque los valores muestras cercanos a $N/2$ tienen aún apreciable magnitud.

El cómputo de la expresión de referencia con los valores muestras hallados en el ejemplo anterior, llevan a la forma de onda sintetizada que se calcula del siguiente modo:

$$n1 := 0 \dots \frac{N}{2} \quad HF1_{n1} := HF_{n1} \quad HF1_{N-n1} := HF_{n1}$$

$$h2_k := \sum_{n=0}^{N-1} HF1_n \cdot \exp\left(j \cdot 2 \cdot \pi \cdot k \cdot \frac{n}{N}\right)$$

En la Figura 21.1 se puede apreciar gráficamente este resultado.



Como se ve, los resultados tienden a oscilar alrededor del valor correcto. Estas oscilaciones son debidas al bien conocido fenómeno de Gibbs el cual establece que el truncamiento en un dominio lleva a oscilaciones en el otro dominio. Para disminuir la magnitud de estas oscilaciones es necesario considerar más coeficientes armónicos; esto es, incrementar N.

REDUCCION DEL LEAKAGE

El **Leakage** es un efecto inherente a la DFT debido al truncamiento requerido en el dominio del tiempo.

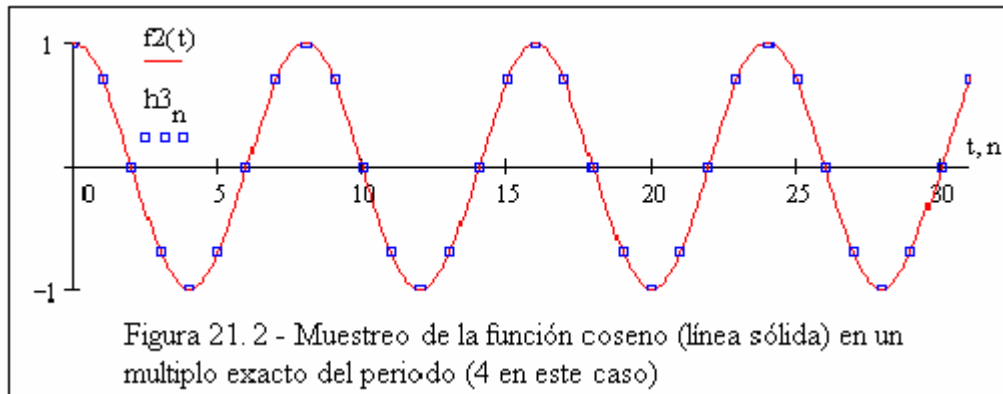
Recuérdese que el truncamiento de una función periódica para producir otra que es un múltiplo del periodo, resulta en una aguda discontinuidad en el dominio del tiempo, o equivalentemente produce lóbulos laterales en el dominio de frecuencia. Estos lóbulos laterales son los responsables de componentes de frecuencia adicionales a los cuales se le denomina **leakage**.

Para ilustrar este punto se computará la DFT de la función coseno.

Para un intervalo de muestreo $T=1$ y $N=32$ muestras.

$$\begin{aligned} f0 &:= \frac{1}{8} & N &:= 32 & T &:= 1 \\ n &:= 0..N-1 & t &:= 0..10 \cdot \pi \\ f2(t) &:= \cos(2 \cdot \pi \cdot f0 \cdot t) & h3_n &:= \cos(2 \cdot \pi \cdot f0 \cdot n) \end{aligned}$$

Esto se ilustra en la Figura 21.2.

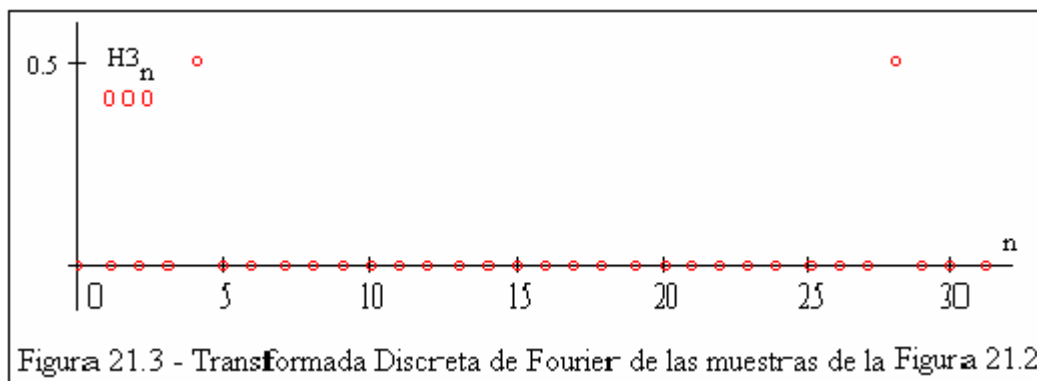


Nótese que las 32 muestras definen exactamente cuatro periodos de la forma de onda periodica.

Aplicando los conceptos vistos en cuanto a DFT:

$$H3_n := \frac{1}{N} \cdot \sum_{k=0}^{N-1} h3_k \cdot \exp\left(\frac{-j \cdot 2 \cdot \pi \cdot n \cdot k}{N}\right)$$

Lo que lleva al resultado mostrado en la Figura 21.3.



Todos los resultados son cero, excepto en la frecuencia de la señal (n=4 y su imagen n=28).

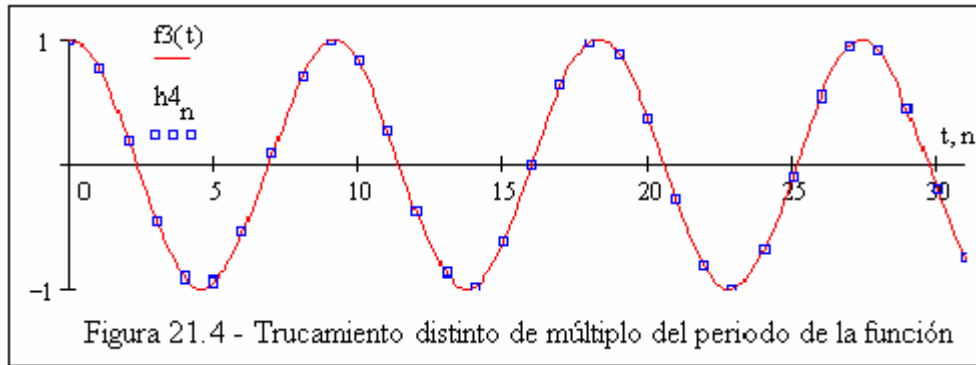
Esto era lo esperado, la Transformada de Fourier de una función coseno está dada por dos impulsos de altura igual a la mitad de la amplitud de dicha función y ubicados en la frecuencia de la misma (positiva y negativa).

Si el intervalo de truncamiento no es elegido igual a un múltiplo del periodo, las características de lóbulo lateral de la función en frecuencia **sin(f)/f** genera una considerable diferencia en los resultados de Transformada Discreta y Continua de Fourier.

Para ilustrar este efecto, considérese:

$$f0 := \frac{1}{9.143} \quad N := 32 \quad T := 1$$

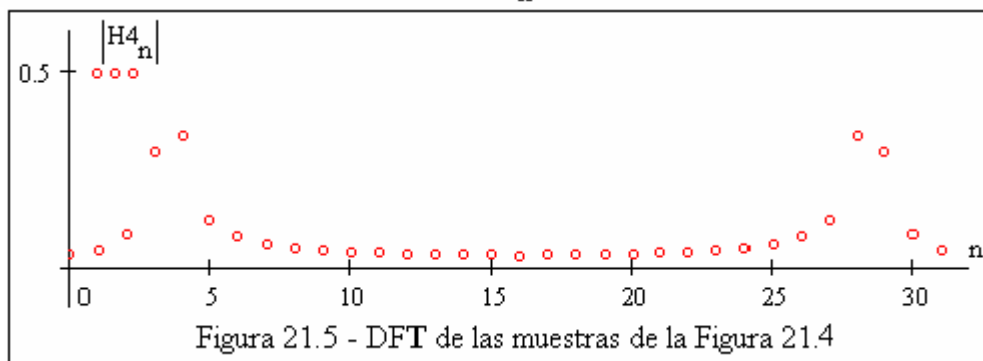
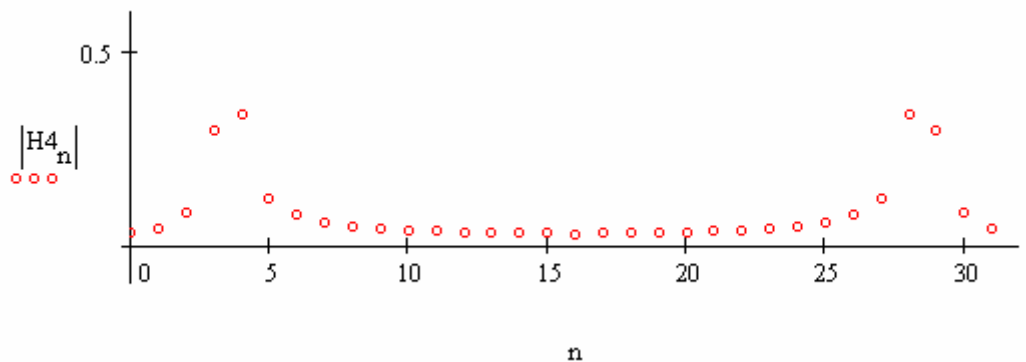
$$n := 0..N-1 \quad f3(t) := \cos(2 \cdot \pi \cdot f0 \cdot t) \quad h4_n := \cos(2 \cdot \pi \cdot f0 \cdot n)$$



Nótese que los 32 puntos no definen un múltiplo del periodo y como resultado se introduce una aguda discontinuidad.

Calculando la DFT:

$$H4_n := \frac{1}{N} \cdot \sum_{k=0}^{N-1} h4_k \cdot \exp\left(\frac{-j \cdot 2 \cdot \pi \cdot n \cdot k}{N}\right)$$



En la Figura 21.5 se grafica el módulo de la DFT de las muestras de la Figura 21.4, de más arriba. Aquí existen componentes de frecuencia no nulas en todas las frecuencia discretas de la DFT.

Como se indicó previamente, los componentes de frecuencia adicionales se denominan **leakage** y son un resultado de las características lóbulo lateral de la función $\sin(f)/f$. Para reducir el leakage es necesario emplear una función de truncamiento en el dominio del tiempo la cual tenga características glóbulo lateral que sean de magnitud más pequeña que aquellas de la función $\sin(f)/f$.

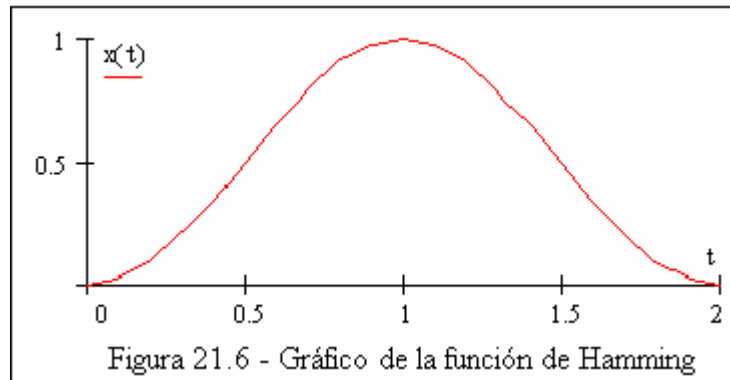
A menor lóbulo lateral, menor leakage afectará a los resultados de la DFT. Afortunadamente, existen funciones de truncamiento que exhiben exactamente las características deseadas.

Una función de truncamiento particularmente buena es la función de Hamming:

$T_c := 2$ Parámetro, Intervalo de Truncamiento

$$x(t) := \frac{1}{2} - \frac{1}{2} \cdot \cos\left(2 \cdot \frac{\pi}{T_c} \cdot t\right) \quad \text{con} \quad 0 \leq t \leq T_c$$

$t := 0, 0.1 \dots 3 \cdot T_c$



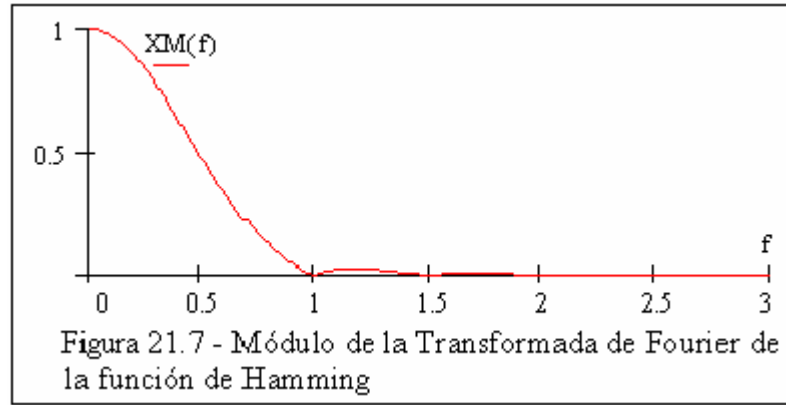
El módulo de la Transformada de Fourier de la función de Hamming está dada por:

$$|X(f)| = \frac{1}{2} \cdot Q(f) + \frac{1}{4} \cdot \left(Q\left(f + \frac{1}{T_c}\right) + Q\left(f - \frac{1}{T_c}\right) \right) \quad \text{donde} \quad Q(f) = \frac{\sin(\pi \cdot T_c \cdot f)}{\pi \cdot f}$$

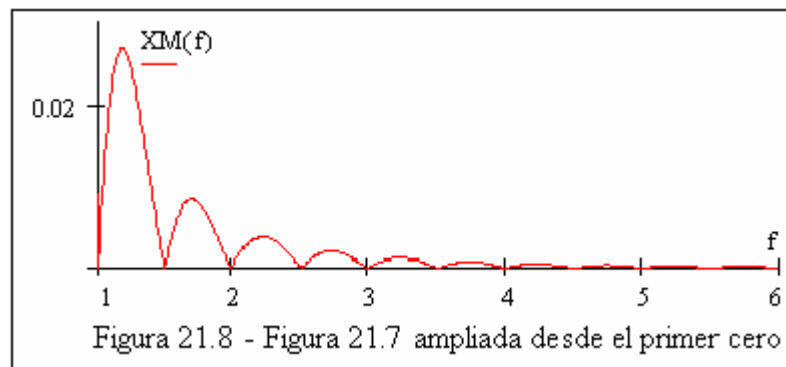
Para representa esto gráficamente, se instrumentan los siguientes pasos que conducen a la figura 21.7.

$$XM(f) := \left| \frac{1}{2} \cdot \frac{\sin(\pi \cdot T_c \cdot f)}{\pi \cdot f} + \frac{1}{4} \cdot \left[\frac{\sin\left[\pi \cdot T_c \cdot \left(f + \frac{1}{T_c}\right)\right]}{\pi \cdot \left(f + \frac{1}{T_c}\right)} + \frac{\sin\left[\pi \cdot T_c \cdot \left(f - \frac{1}{T_c}\right)\right]}{\pi \cdot \left(f - \frac{1}{T_c}\right)} \right] \right|$$

$f := 0.01, 0.02 \dots 3 \cdot T_c$



En la figura 21.8 se puede apreciar la misma Figura 21., pero ampliada desde el primer cero



Como se muestra en la figura 21.8, esta función de la frecuencia tiene varios pequeños lóbulos laterales. Otras funciones de truncamiento tienen propiedades similares, sin embargo se elige la de Hamming por su simplicidad.

Debido a las características de bajo lóbulo lateral de esta función, se espera que su utilización reduzca significativamente el leakage que resulta del truncamiento en el dominio del tiempo.

En la Figura 21.9 se muestra la función coseno, vista anteriormente, multiplicada por la función de truncamiento de Hamming.

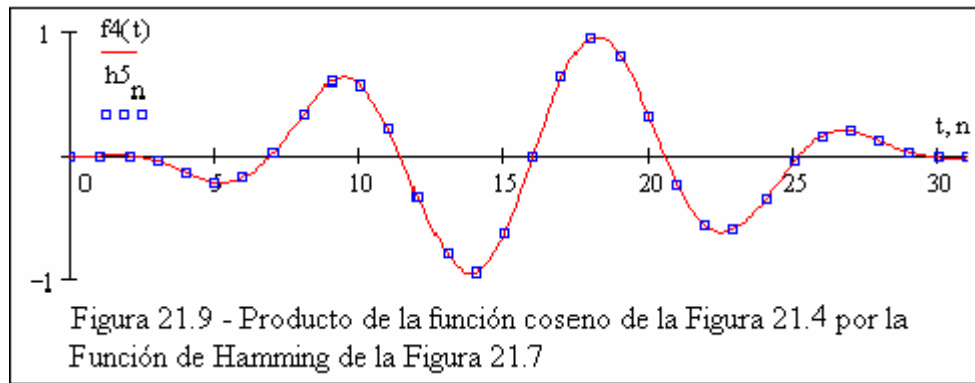
$$f_0 := \frac{1}{9.143} \quad N := 32 \quad T := 1$$

$$n := 0..N - 1$$

$$f_4(t) := \cos(2 \cdot \pi \cdot f_0 \cdot t) \cdot \left[\frac{1}{2} - \frac{1}{2} \cdot \cos \left[2 \cdot \frac{\pi}{T_c} \cdot \frac{t}{\left(\frac{N}{2} \right)} \right] \right]$$

$$t := 0, 0.1 .. 10 \cdot \pi \quad T_c := N$$

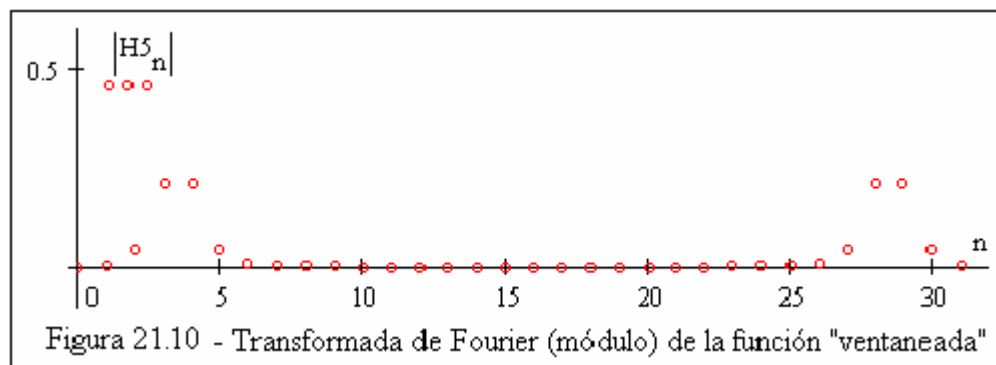
$$h_{5_n} := \cos(2 \cdot \pi \cdot f_0 \cdot n) \cdot \left(\frac{1}{2} - \frac{1}{2} \cdot \cos \left(2 \cdot \frac{\pi}{T_c} \cdot n \right) \right)$$



Nótese que el efecto de la función de Hamming es reducir la discontinuidad que resulta de la función truncamiento rectangular.

La Figura 21.10 muestra el módulo de la transformada de Fourier para las muestras indicadas arriba.

$$H5_n := \frac{1}{N} \cdot \sum_{k=0}^{N-1} h5_k \cdot \exp\left(\frac{-j \cdot 2 \cdot \pi \cdot n \cdot k}{N}\right)$$



Como se esperaba, el leakage ha sido reducido debido a las características lóbulo laterales de la función de truncamiento de Hamming. Las componentes de frecuencia no nulas son considerablemente "ensanchadas" con respecto a la función impulsiva deseada.

Recuérdese que esto es lo que se espera, ya que el efecto del truncamiento en el dominio del tiempo es convolucionar la función impulsiva de frecuencia con la FT de la función truncamiento. En general, mientras más se reduce el leakage, aparece una DFT con resultados más "ensanchados". La función de Hamming resulta ser un compromiso aceptable.

PROPIEDADES DE LA TRANSFORMADA DE FOURIER

Funciones reales en general

Si $f(t)$ es una **función real** la transformada de Fourier tiene una parte real y una parte imaginaria.

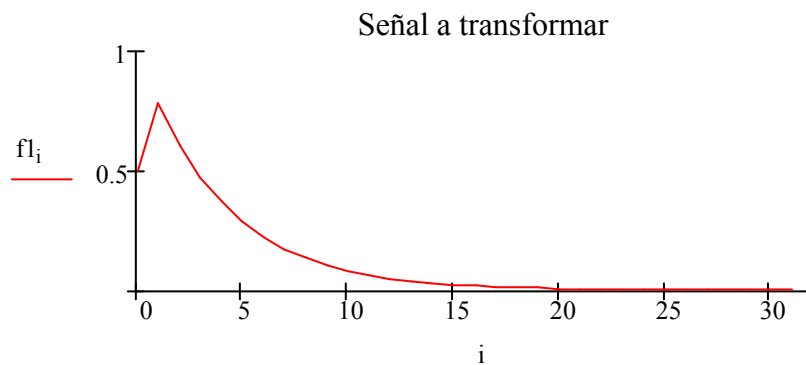
Ejemplo:

$f(t) = \exp(-t)$ $N := 32$ número de muestras

$i := 0..N - 1$ índice $T := 0.25$ tiempo de muestreo

$fl_i := \exp(-T \cdot i)$ función (muestreada)

$fl_0 := \frac{fl_0}{2}$ ajuste en la discontinuidad

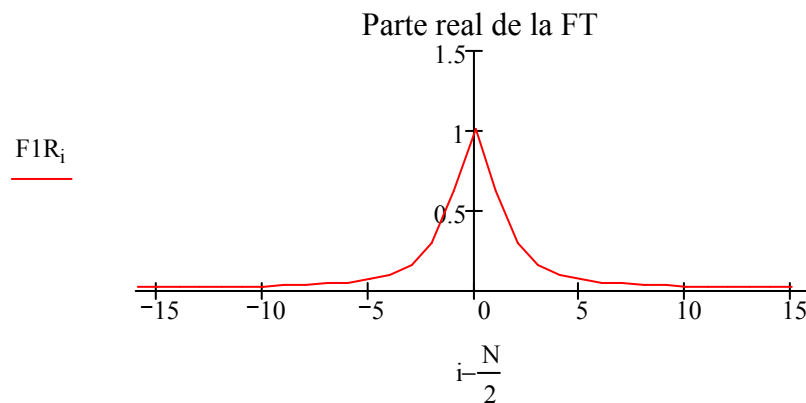


$F1 := \text{cff}(fl) \cdot \sqrt{N} \cdot T$ transformada de Fourier

$i1 := 0.. \frac{N}{2}$ índice auxiliar

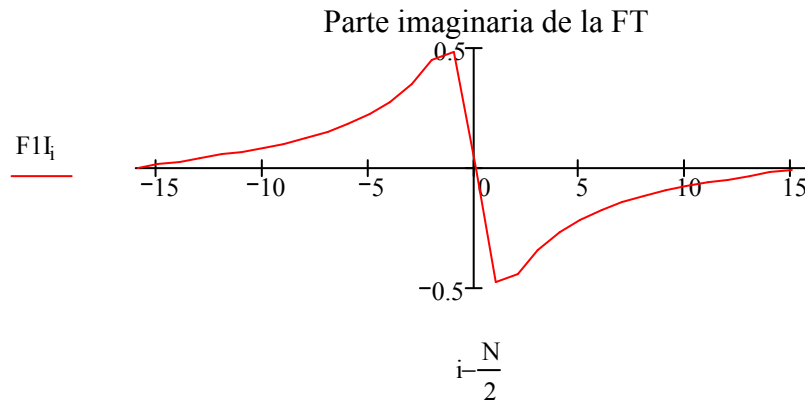
Para reconstruir la "auténtica" parte real de la transformada (que se extiende desde -0.5 a 0.5, en frecuencias normalizadas, o bien $-N/2$ a $+ N/2$) se debe realizar un "fold" (plegado) de la segunda mitad del resultado.

$F1R_{i+\frac{N}{2}} := \text{Re}(F1)_i$ $F1R_{i1} := \text{Re}(F1)_{\frac{N}{2}-i1}$ "folding" del resultado



En cambio, para reconstruir la "auténtica" parte imaginaria de la transformada (que se extiende desde -0.5 a 0.5, en frecuencias normalizadas, o bien -N/2 a + N/2) se debe realizar un "fold" (plegado) y un "flip" de la segunda mitad del resultado.

$$F1I_{i+\frac{N}{2}} := -\text{Im}(F1)_i \quad F1I_{i1} := \text{Im}(F1)_{\frac{N}{2}-i1}$$
 "flip" y "folding" del resultado



PROPIEDAD: La transformada de Fourier de una función real es una función compleja cuya parte real es PAR y cuya parte imaginaria es IMPAR.

Ejercicio N° 1 - Plantéese una función real, cuya transformada de Fourier continua pueda calcular, muestreéela y hállele la DFT (transformada discreta de Fourier). Compare los resultados con los de la continua y verifique la propiedad enunciada arriba.

Funciones reales y pares, desplazamiento (shift)

Sea $f(t)$ una **función real**.

Ejemplo:

$$f(t) := \text{if}(0 \leq |t| < .2, 1, 0)$$

$$F(\omega) = \int_0^{.2} f(t) \cdot \exp(-j \cdot \omega \cdot t) dt = 2 \cdot \exp\left(-j \cdot \frac{\omega \cdot 0.2}{2}\right) \cdot \frac{\sin\left(\omega \cdot \frac{0.2}{2}\right)}{\omega}$$

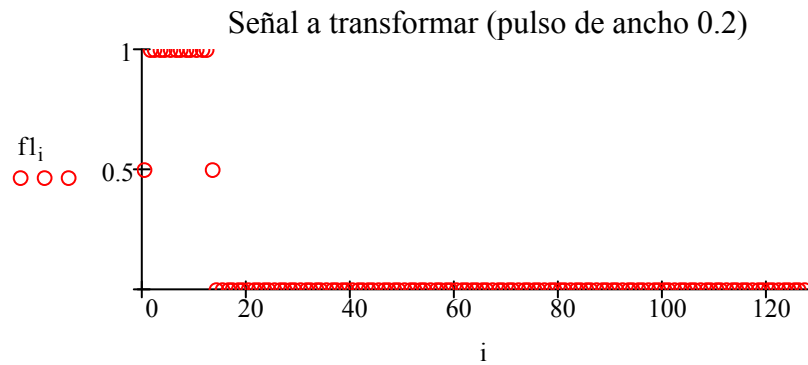
$$N := 128 \quad \text{número de muestras}$$

$$T := \frac{2}{N} \quad \text{tiempo de muestreo}$$

$$i := 0..N-1 \quad \text{índice}$$

$$f_{l_i} := f(-T \cdot i) \quad \text{función (muestreada)}$$

$$f_{l_0} := \frac{f_{l_0}}{2} \quad f_{l_{13}} := \frac{f_{l_{12}}}{2} \quad \text{ajuste en las discontinuidades}$$



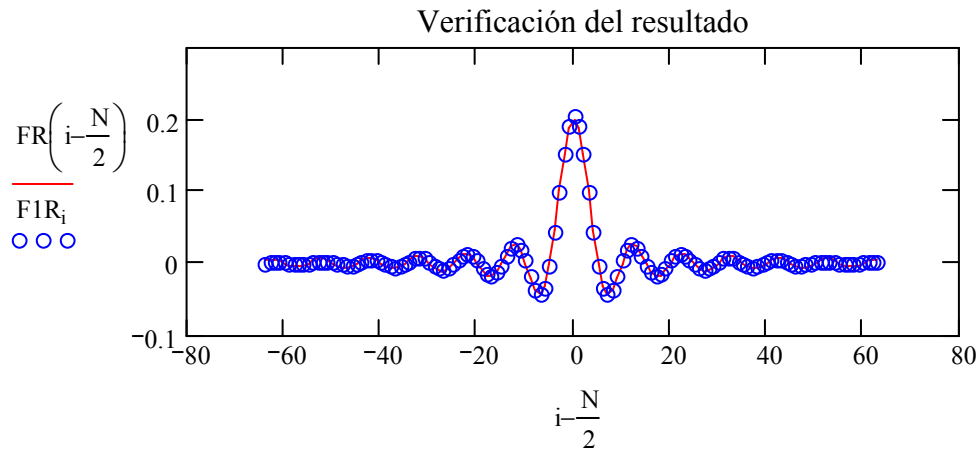
$$F1 := \text{cff}(f1) \cdot \sqrt{N} \cdot T \quad \text{transformada de Fourier}$$

$$i1 := 0.. \frac{N}{2} \quad \text{índice auxiliar}$$

$$F1R_{i+\frac{N}{2}} := \text{Re}(F1)_i \quad F1R_{i1} := \text{Re}(F1)_{\frac{N}{2}-i1} \quad \text{"folding" del resultado}$$

La transformada continua (parte real) se puede representar (a modo comparativo) mediante la siguiente expresión:

$$FR(i) := \text{if } i = 0, 0.2, \text{Re} \left[2 \cdot \exp \left[-j \cdot 0.1 \cdot \left(2 \cdot \frac{\pi \cdot i}{N \cdot T} \right) \right] \cdot \frac{\sin \left(2 \cdot \frac{\pi \cdot i}{N \cdot T} \cdot 0.1 \right)}{2 \cdot \frac{\pi \cdot i}{N \cdot T}} \right]$$

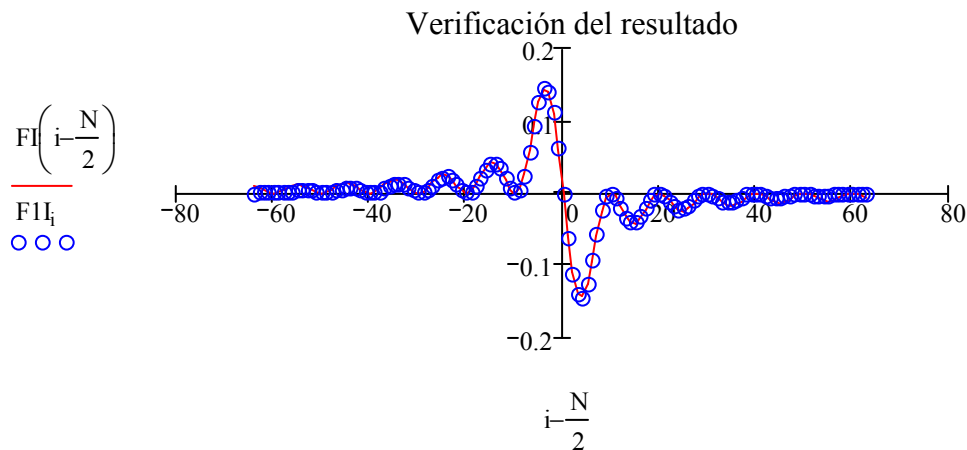


$F1I_{i+\frac{N}{2}} := -\text{Im}(F1)_i$ $F1I_{i1} := \text{Im}(F1)_{\frac{N}{2}-i1}$ "flip" y "folding" del resultado

La transformada continua (parte imaginaria) se puede representar (a modo comparativo) mediante la siguiente expresión:

$$FI(i) := \text{Im} \left[2 \cdot \exp \left[-j \cdot 0.1 \cdot \left(2 \cdot \frac{\pi \cdot i}{N \cdot T} \right) \right] \cdot \frac{\sin \left(2 \cdot \frac{\pi \cdot i}{N \cdot T} \cdot 0.1 \right)}{2 \cdot \frac{\pi \cdot i}{N \cdot T}} \right]$$

Al no haber discontinuidad en $i=0$, no se recurre al if.



Si se analiza el resultado de la transformada continua de Fourier del mismo pulso, pero centrado en $t=0$ (**función real y par**), la resolución de la integral da:

$$F(\omega) = \int_{-0.1}^{0.1} f(t) \cdot \exp(-j \cdot \omega \cdot t) dt = 2 \cdot \frac{\sin \left(\omega \cdot \frac{0.2}{2} \right)}{\omega}$$

Es posible observar que la parte imaginaria del resultado es nula. De aquí se deriva:

PROPIEDAD: Si $f(t)$ es una función real y par la transformada de Fourier tiene sólo parte real y la parte imaginaria es nula.

En cambio, en todo el análisis anterior, se trató con la misma función pero desfasada medio ancho de pulso (0.1) hacia la derecha. Esto es necesario porque la primera muestra se toma en $t=0$ y no en $t=-0.1$. A consecuencia de esto se deriva una nueva propiedad:

PROPIEDAD: Si $f(t)$ sufre un desplazamiento (en este caso a la derecha, de 0.1) su transformada de Fourier queda automáticamente multiplicada por el factor $\exp(-j\omega.c/2)$ siendo c el ancho del pulso.

Ejercicio N° 2 - Plantéese una función real y par, cuya transformada de Fourier continua pueda calcular, muéstréela y hállele la DFT (transformada discreta de Fourier). Compare los resultados con los de la continua y verifique las propiedades enunciadas arriba.

Funciones reales e impares, desplazamiento (shift)

Sea $f(t)$ una **función real**.

Ejemplo:

$$f(t) := \text{if}(t < 0, 0, \text{if}(t < 0.1, -1, \text{if}(t < 0.2, 1, 0)))$$

$$F(\omega) = \int_0^{0.2} f(t) \cdot \exp(-j \cdot \omega \cdot t) dt = \frac{j}{\omega} \cdot [1 - \exp(-j \cdot \omega \cdot 0.1)]^2$$

$$F(\omega) = -j \cdot 4 \cdot \frac{(\sin(0.05 \cdot \omega))^2}{\omega} \cdot \exp(-j \cdot \omega \cdot 0.1)$$

$$N := 128 \quad \text{número de muestras}$$

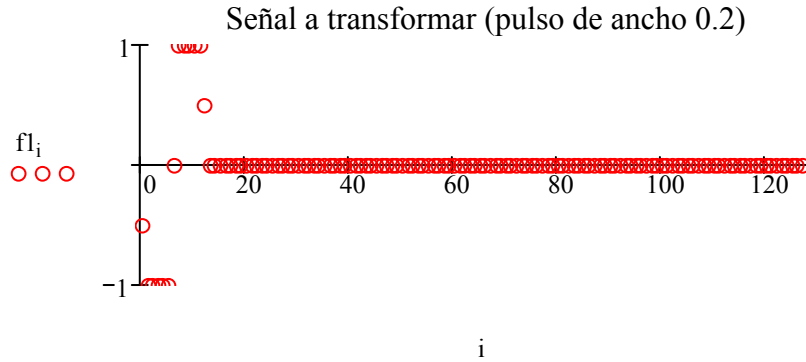
$$T := \frac{2}{N} \quad \text{tiempo de muestreo}$$

$$i := 0 .. N - 1 \quad \text{índice}$$

$$fl_i := f(T \cdot i) \quad \text{función (muestreada)}$$

$$fl_0 := \frac{fl_0}{2} \quad fl_6 := \frac{fl_6 + fl_7}{2} \quad fl_{12} := \frac{fl_{12}}{2} \quad \text{ajuste} \quad \text{en} \quad \text{las}$$

discontinuidades



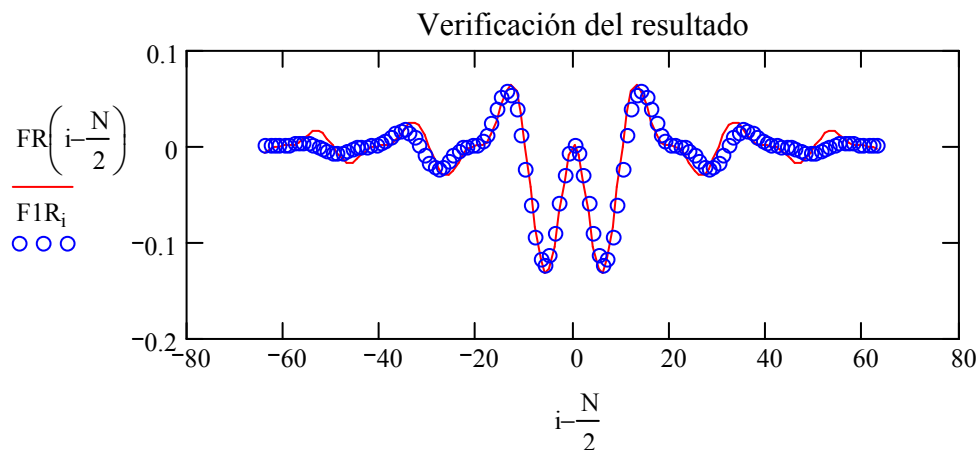
$$F1 := \text{cfft}(f1) \cdot \sqrt{N} \cdot T \quad \text{transformada de Fourier}$$

$$i1 := 0.. \frac{N}{2} \quad \text{índice auxiliar}$$

$$F1R_{i+\frac{N}{2}} := \text{Re}(F1)_i \quad F1R_{i1} := \text{Re}(F1)_{\frac{N}{2}-i1} \quad \text{"folding" del resultado}$$

La transformada continua (parte real) se puede representar (a modo comparativo) mediante la siguiente expresión:

$$FR(i) := \text{if } i = 0, 0, \text{Re} \left[\frac{j}{2 \cdot \frac{\pi \cdot i}{N \cdot T}} \cdot \left[1 - \exp \left[-j \cdot \left(2 \cdot \frac{\pi \cdot i}{N \cdot T} \right) \cdot 0.1 \right] \right]^2 \right] \right]$$

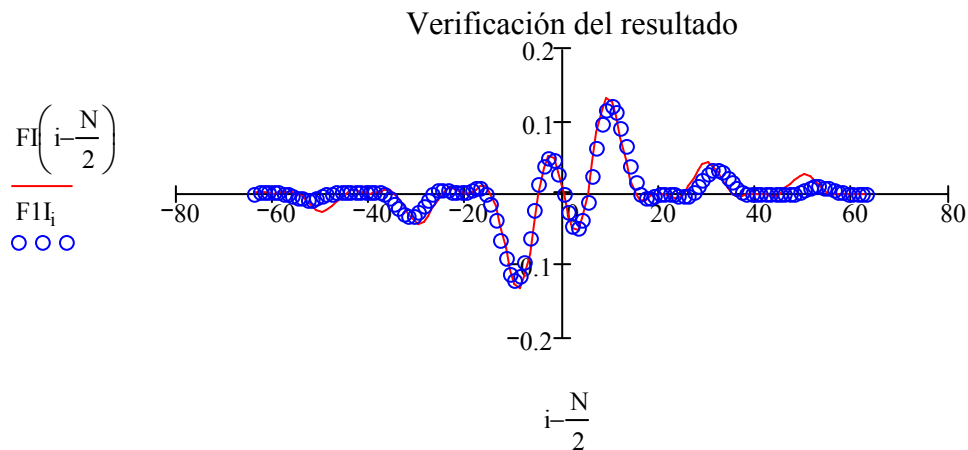


$$F1I_{i+\frac{N}{2}} := -\text{Im}(F1)_i \quad F1I_{i1} := \text{Im}(F1)_{\frac{N}{2}-i1} \quad \text{"flip" y "folding" del resultado}$$

La transformada continua (parte imaginaria) se puede representar (a modo comparativo) mediante la siguiente expresión:

$$FI(i) := \text{if } i = 0, 0, \text{Im} \left[\frac{j}{2 \cdot \frac{\pi \cdot i}{N \cdot T}} \cdot \left[1 - \exp \left[-j \cdot \left(2 \cdot \frac{\pi \cdot i}{N \cdot T} \right) \cdot 0.1 \right] \right]^2 \right] \right]$$

Al no haber discontinuidad en $i=0$, no se recurre al if.



Si se analiza el resultado de la transformada continua de Fourier del mismo pulso, pero centrado en $t=0$ (**función real e impar**), la resolución de la integral da:

$$F(\omega) = \int_{-0.1}^{0.1} f(t) \cdot \exp(-j \cdot \omega \cdot t) dt = -j \cdot 4 \cdot \frac{\sin\left(\omega \cdot \frac{0.2}{2}\right)^2}{\omega}$$

Es posible observar que la parte real del resultado es nula. De aquí se deriva:

PROPIEDAD: Si $f(t)$ es una función real e impar la transformada de Fourier tiene sólo parte imaginaria y la parte real es nula.

En cambio, en todo el análisis anterior, se trató con la misma función pero desfasada medio ancho de pulso (0.1) hacia la derecha. Esto es necesario porque la primera muestra se toma en $t=0$ y no en $t=-0.1$. Como en el caso anterior se verifica la propiedad:

PROPIEDAD: Si $f(t)$ sufre un desplazamiento (en este caso a la derecha, de 0.1) su transformada de Fourier queda automáticamente multiplicada por el factor $\exp(-j\omega \cdot c/2)$ siendo c el ancho del pulso.

Ejercicio N° 3 - Plantéese una función real e impar, cuya transformada de Fourier continua pueda calcular, muestreela y hállele la DFT (transformada discreta de Fourier). Compare los resultados con los de la continua y verifique las propiedades enunciadas arriba.

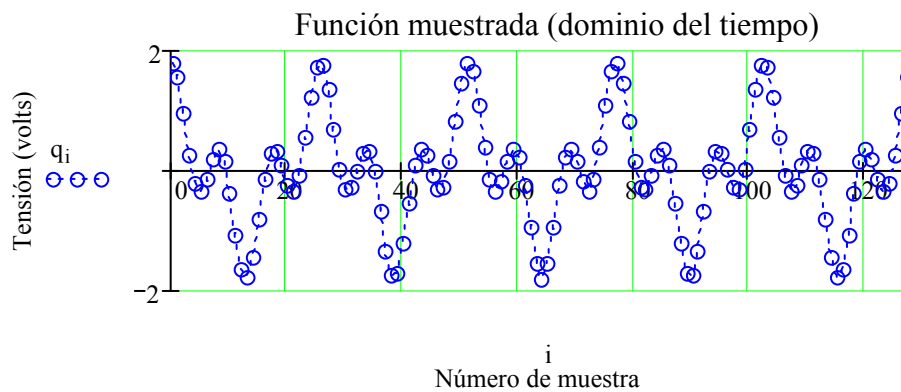
Reconstrucción de una función par en el dominio del tiempo mediante sus componentes de Fourier en el dominio de frecuencia.

Se parte de la función en el dominio del tiempo dada por la suma de dos cosenos de diferente frecuencia (Función par).

$f_0 := 5$ $N := 128$ $i := 0..N-1$ $T := \frac{1}{N}$ tiempo de muestreo (en seg)

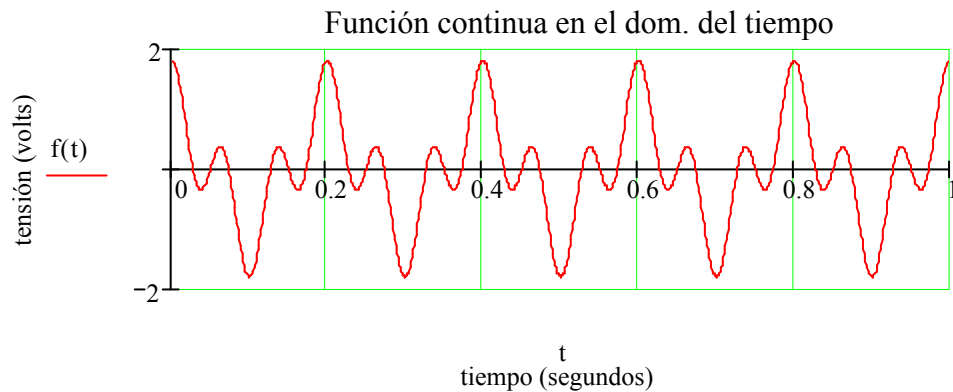
$T = 7.813 \times 10^{-3}$ $f_s := \frac{1}{T}$ frecuencia de muestreo (en Hz) $f_s = 128$

$f(t) := \cos(2 \cdot \pi \cdot f_0 \cdot t) + 0.8 \cdot \cos(2 \cdot \pi \cdot 3 \cdot f_0 \cdot t)$ $q_i := f(i \cdot T)$



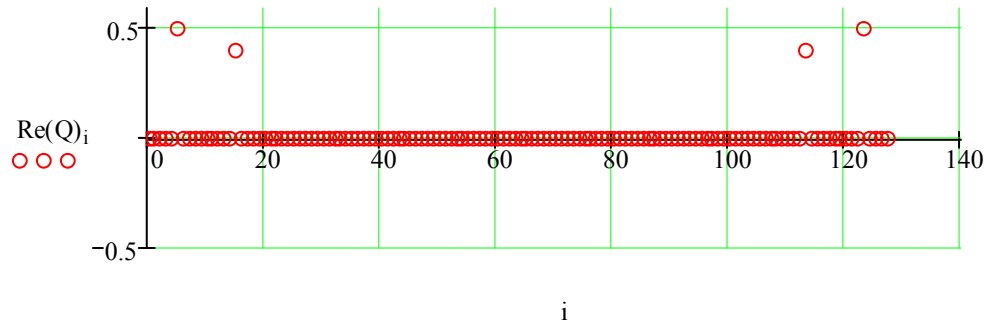
Para representar gráficamente la señal continua, se da un rango para t.

$t := 0, 0.001 .. N \cdot T$



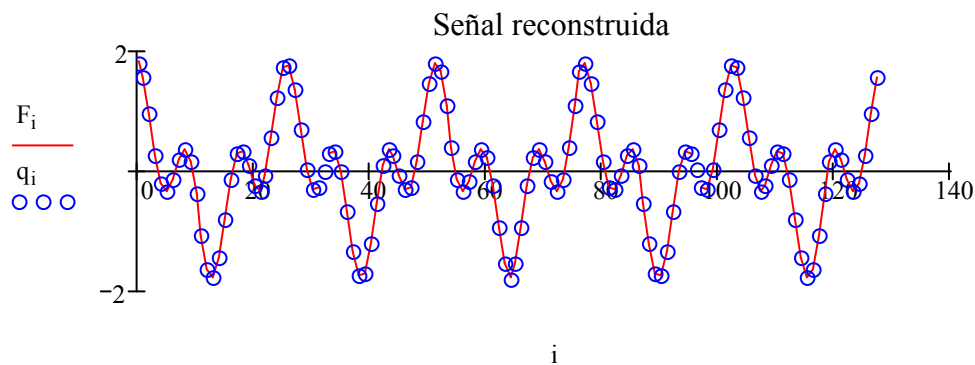
Hallando la transformada de Fourier para la señal discretizada

$Q := \text{cff}(q) \cdot T \cdot \sqrt{N}$



Es posible reconstruir la señal (dominio del tiempo) desde las muestras de la transformada de Fourier (dominio de frecuencia), aplicando la siguiente expresión.

$$F_i := \frac{1}{N \cdot T} \cdot \sum_{k=0}^{N-1} Q_k \cdot \left(\cos\left(2 \cdot \pi \cdot \frac{k \cdot i}{N}\right) + \sin\left(2 \cdot \pi \cdot \frac{k \cdot i}{N}\right) \right)$$



Supóngase ahora una **función par, pero no periódica**. Para hacer el muestreo con un sistema de procesamiento, lógicamente no se pueden tomar muestras de índice negativo. Por lo tanto la forma de solucionar este inconveniente es indicar los valores de la zona negativa como muestras con índice entre $N/2$ y $N-1$, mientras que los de la zona par se adjudican a los índice entre 0 y $N/2$.

En forma más práctica, a las muestras de la zona positiva se les hace un "fold" alrededor de $N/2$ previo a su análisis de Fourier.

A modo de ejemplo, el siguiente es un par transformado:

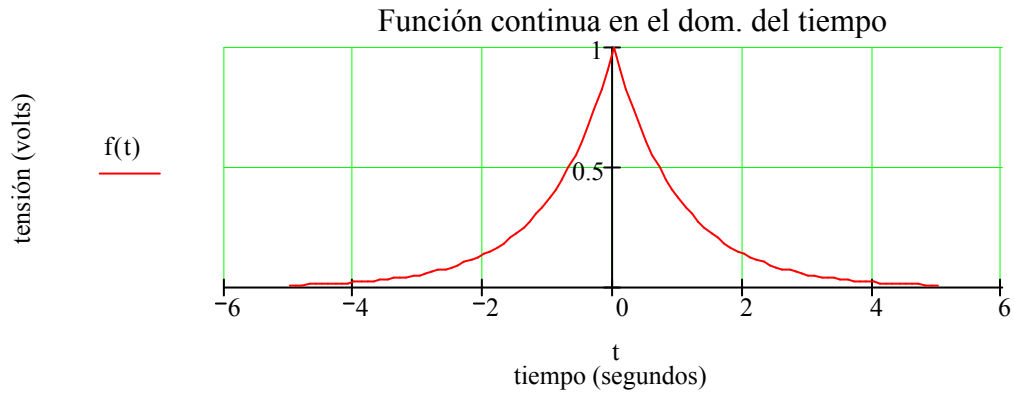
$$\exp(-|t|) \quad <===== > \quad \frac{2}{1 + \omega^2}$$

como es fácilmente verificable resolviendo:

$$\int_{-\infty}^0 \exp(t) \cdot \exp(-j \cdot \omega \cdot t) dt + \int_0^{\infty} \exp(-t) \cdot \exp(-j \cdot \omega \cdot t) dt = \frac{2}{1 + \omega^2}$$

$$N := 64 \quad k1 := 0.. \frac{N}{2} - 1 \quad T := 0.25$$

$$f(t) := \exp(-|t|) \quad t := -5, -4.9..5$$

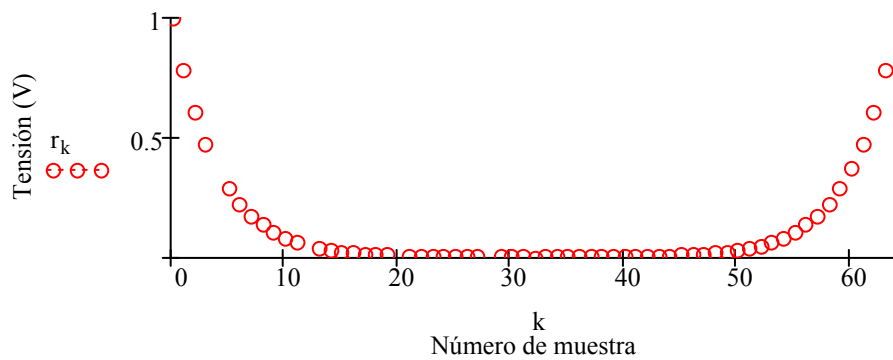


$$r_{k1} := f(k1 \cdot T)$$

$$r_{k1 + \frac{N}{2}} := r_{\frac{N}{2} - k1}$$

función muestreada y

"foldeada"

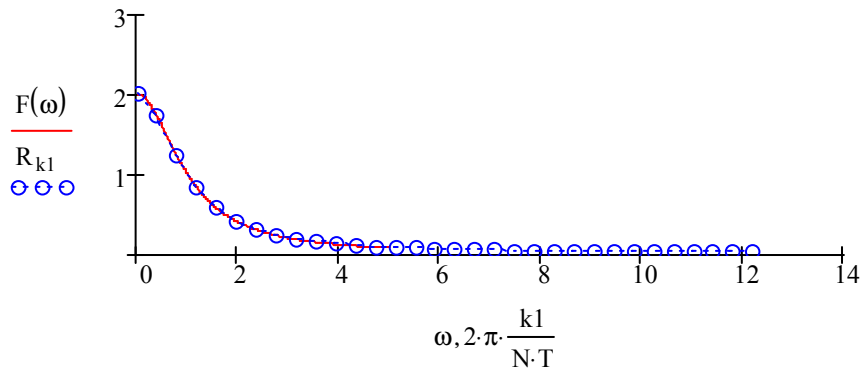


La transformada discreta de Fourier de esta función se halla de la siguiente forma:

$$R := \text{cfft}(r) \cdot T \cdot \sqrt{N}$$

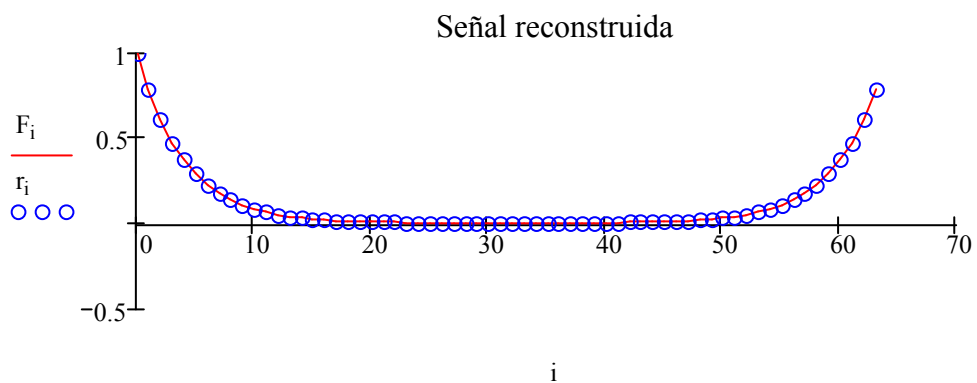
y comparada con el resultado continuo (referencia):

$$\omega := 0, 0.01..5 \quad F(\omega) := \frac{2}{1 + \omega^2}$$



Es posible reconstruir la señal (dominio del tiempo) desde las muestras de la transformada de Fourier (dominio de frecuencia), aplicando la siguiente expresión.

$$i := 0..N - 1 \quad F_i := \frac{1}{N \cdot T} \cdot \sum_{k=0}^{N-1} R_k \cdot \left(\cos\left(2 \cdot \pi \cdot \frac{k \cdot i}{N}\right) + \sin\left(2 \cdot \pi \cdot \frac{k \cdot i}{N}\right) \right)$$



Efecto del agregado de ceros a una función

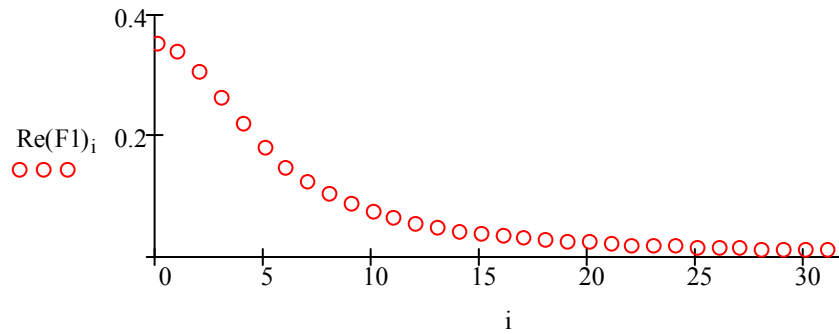
Si al conjunto de datos al que se le ha de aplicar la transformada de Fourier se le agregan ceros, se pueden observar los siguientes efectos:

$$f(t) := \exp(-t) \quad N := 32 \quad \text{número de muestras}$$

$$i := 0..N - 1 \quad \text{índice} \quad T := 0.25 \quad \text{tiempo de muestreo}$$

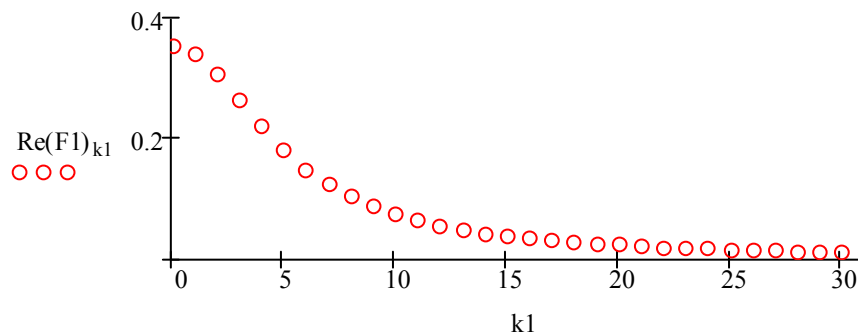
$$f_{l_i} := \exp(-T \cdot i) \quad \text{función (muestreada)} \quad f_{l_0} := \frac{f_{l_0}}{2}$$

$$F1 := \text{cff}(f1)$$



$$k := N..2 \cdot N - 1 \quad f1_k := 0 \quad k1 := 0..2 \cdot N - 1$$

$$F2 := \text{cff}(f1)$$



Como se puede observar, no hay efecto alguno.

Multiplicación de señales (Modulación de Amplitud)

Como una importante propiedad de la transformada de Fourier es que la convolución en un dominio corresponde a la multiplicación en el otro dominio. La modulación de amplitud es un ejemplo de la situación inversa, **la multiplicación en el dominio del tiempo corresponde a la convolución en el dominio de frecuencia**. Además, la modulación en amplitud provee un excelente ejemplo de cómo las elusivas frecuencias negativas entran en la ciencia diaria y en los problemas de ingeniería.

Las señales de audio son adecuadas para la comunicación a corta distancia; cuando se habla en una habitación se escucha desde otra. En el otro extremo, las radio frecuencias son muy buenas en la propagación a grandes distancias. Por ejemplo, si una onda seno de 100 V, 1 Mhz se alimenta a una antena, la onda de radio resultante puede ser detectada en la próxima habitación, el próximo país y aún en todo el planeta.

La modulación es un proceso de mezclado de dos señales para formar una tercera con las características deseables de ambas. Esto siempre involucra procesos no lineales tales como la multiplicación. En radio comunicación, la modulación resulta en señales de radio que se pueden propagar a grandes distancias y transportando audio u otra información.

La radio comunicación es una disciplina extremadamente desarrollada, y se han diseñado muchos esquemas de modulación. Una de los más simples es la llamada **Modulación de Amplitud**.

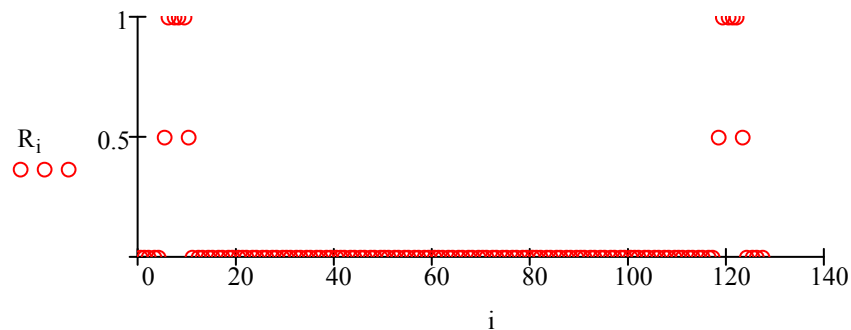
Supóngase una señal **moduladora** que se obtiene por un proceso de síntesis, esto es partiendo de la transformada de Fourier de la misma, la cual tiene una banda determinada de frecuencias.

$$N := 128 \quad i := 0..N - 1$$

$$R_i := \text{if}(5 < i < 10, 1, 0) \quad R_i := \text{if}(N - 10 < i < N - 5, 1, R_i)$$

$$R_5 := \frac{(R_5 + R_6)}{2} \quad R_{10} := \frac{(R_9 + R_{10})}{2}$$

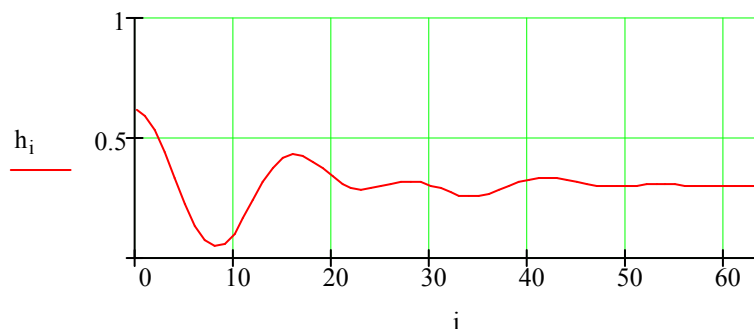
$$R_{118} := \frac{(R_{118} + R_{119})}{2} \quad R_{123} := \frac{(R_{122} + R_{123})}{2}$$



Luego la señal moduladora (en el dominio del tiempo) se puede reconstruir desde estas componentes de Fourier.

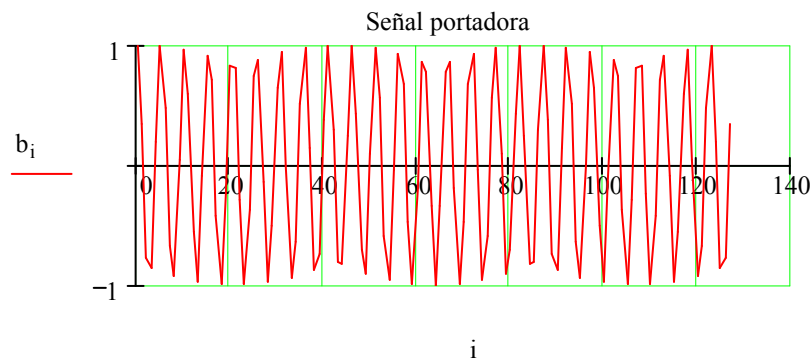
$$h_i := 0.3 + \frac{1}{N \cdot T} \cdot \sum_{k=0}^{N-1} R_k \cdot \left(\cos\left(2 \cdot \pi \cdot \frac{k \cdot i}{N}\right) + \sin\left(2 \cdot \pi \cdot \frac{k \cdot i}{N}\right) \right)$$

Se le ha agregado un valor de continua para que la señal sea siempre positiva.



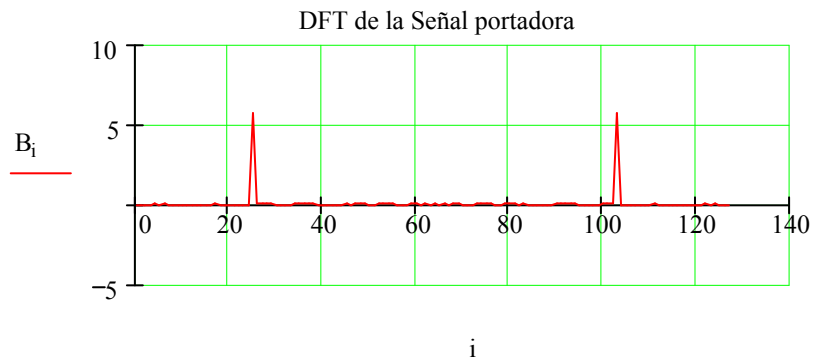
Si se incorpora ahora una señal **portadora**

$$f_0 := 25 \quad b_i := \cos\left(2 \cdot \pi \cdot \frac{i \cdot f_0}{N}\right)$$



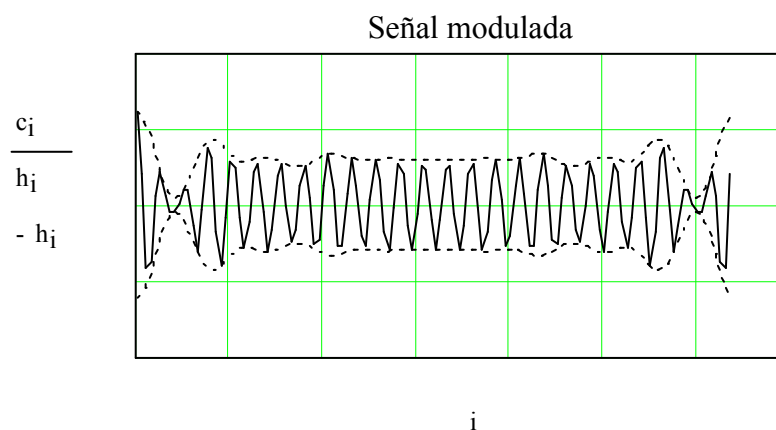
Su espectro de frecuencia estará dado por:

$$B := \text{Re}(\text{cff}(b))$$



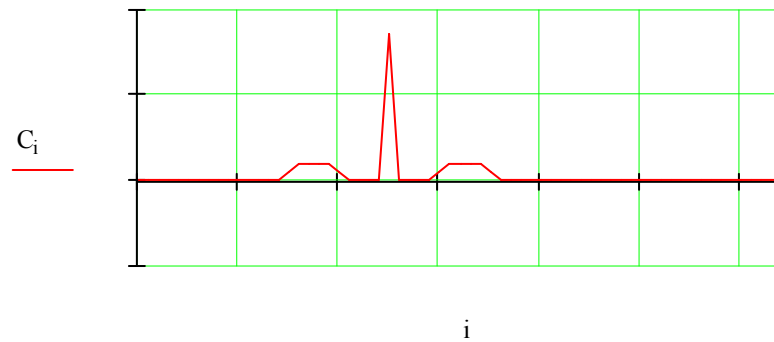
El producto de ambas da la **señal Modulada**

$$c_i := h_i \cdot b_i$$



Si ahora se analizan la transformada de Fourier de esta señal:

$$C := \text{Re}(\text{cff}(c))$$



En las figuras anteriores se muestra cómo la modulación en amplitud aparece tanto en los dominios del tiempo y la frecuencia. El proceso entero se ha hecho con señales discretas.

Si se hablara en términos de frecuencias, la señal moduladora tiene componentes entre 5 y 10 Hz, en cambio la portadora tiene 25 Hz. En cambio en la señal modulada se aprecia la portadora de 25 Hz y a ambos lados de ella el espectro de la moduladora repetido (banda lateral superior, entre $25-10=15$ y $25-5=20$ Hz, y la banda lateral superior, entre $25+5=30$ y $25+10=35$ Hz).

Resumiendo esta propiedad, queda el par:

$$f(t) \cdot \cos(\omega_0 t) \longleftrightarrow \frac{1}{2} \cdot (F(\omega + \omega_0) + F(\omega - \omega_0))$$

Ejercicio N° 4 - Repetir el mismo procedimiento para cuando la moduladora es una onda cosenoidal pura.

Como se ha visto más arriba, existe el siguiente par transformado:

$$f(t) = \exp(-|t|) \longleftrightarrow F(\omega) = \frac{2}{1 + \omega^2}$$

Si se permuta t por ω , es posible verificar la siguiente propiedad:

$$\frac{2}{1 + \omega^2} \longleftrightarrow 2 \cdot \pi \cdot f(-\omega) \quad \text{propiedad de simetría}$$

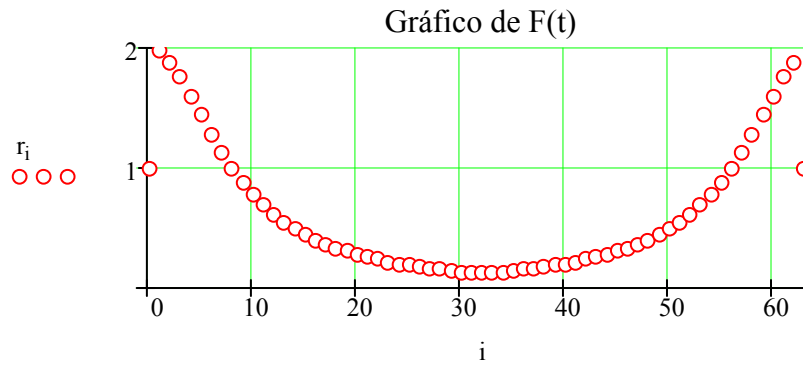
Para su verificación, se permuta t por ω :

$$F(t) = \frac{2}{1 + t^2}$$

$$N := 64 \quad T = 0.125 \quad k1 = 0 \dots \frac{N}{2}$$

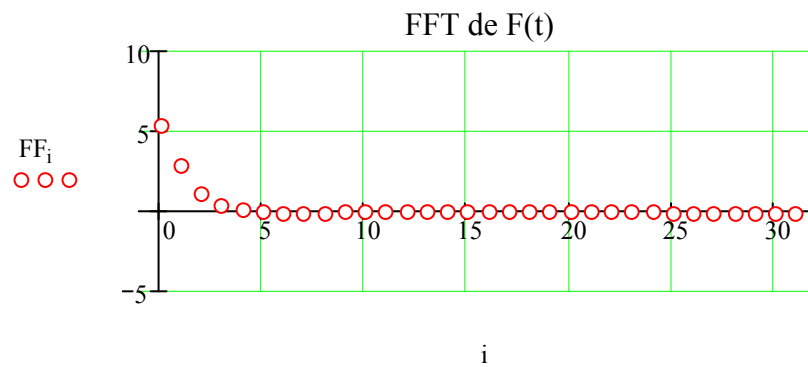
$$r_{k1} = F(-k1 \cdot T) \quad r_{k1 + \frac{N}{2}} := r_{\frac{N}{2} - k1} \quad \text{función muestreada y "foldeada"}$$

$$r_0 = \frac{r_0}{2} \quad r_{N-1} = \frac{r_{N-1}}{2}$$



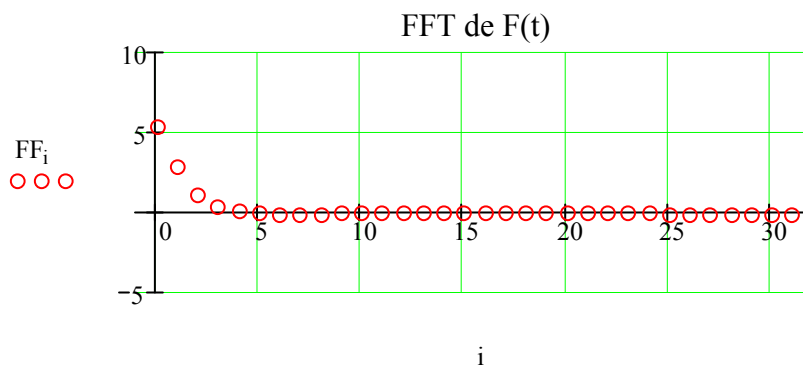
Si se le halla la transformada discreta de Fourier:

$$FF_i = \text{Re}(\text{cfft}(r)) \cdot T \cdot N$$



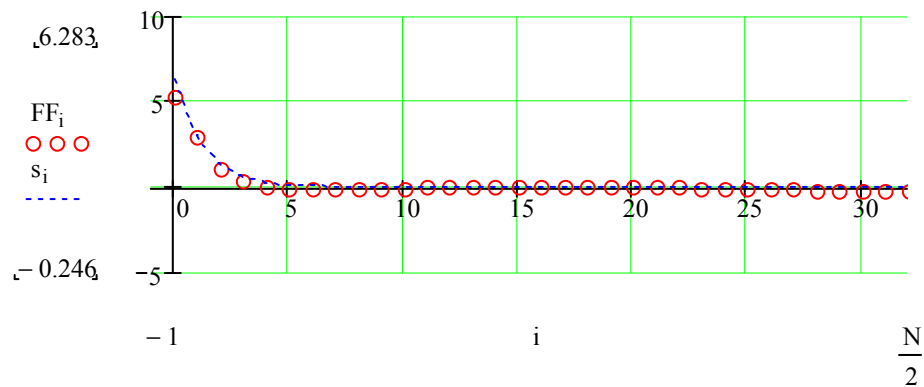
Si se le halla la transformada discreta de Fourier:

$$FF_i := \text{Re}(\text{cfft}(r)) \cdot T \cdot \sqrt{N}$$



Por otro lado, si se permuta ω por t :

$$f(-\omega) = \exp(-j\omega) \quad s_i := 2 \cdot \pi \cdot \exp\left(-2 \cdot \frac{\pi \cdot i}{N \cdot T}\right)$$



Con lo cual se verifica la propiedad de simetría.

Ejercicio N° 5 - Verificar la propiedad de simetría para la transformada de $f(t)=\cos(t)$.

Otra propiedad de la transformada de Fourier establece que:

$$f(at) \longleftrightarrow \frac{1}{|a|} F\left(\frac{\omega}{a}\right)$$

Si: $a := 0.5$

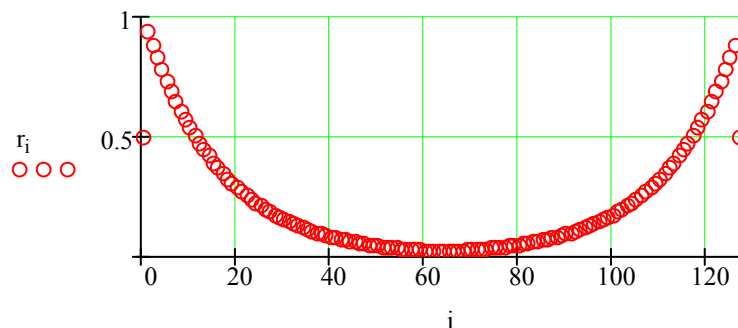
$f(t) := \exp(-a \cdot t)$ $N := 128$ $T := 0.125$

$k1 := 0.. \frac{N}{2}$ $i := 0..N-1$

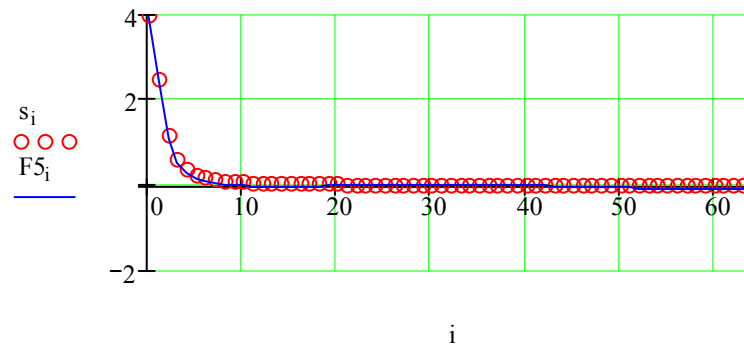
$r_{k1} := f(k1 \cdot T)$ $r_{k1 + \frac{N}{2}} := r_{\frac{N}{2} - k1}$ función muestreada y

"foldeada"

$r_0 := \frac{r_0}{2}$ $r_{N-1} := r_0$



$$F5 := \text{Re}(\text{cff}(r)) \cdot T \cdot \sqrt{N} \quad s_i := \frac{1}{|a|} \cdot \frac{2}{1 + \left(2 \cdot \pi \cdot \frac{i}{N \cdot T \cdot a}\right)^2}$$



Ejercicio N° 6 - Considerar otra función (distinta a la del modelo) y seguir los pasos para la verificación de la propiedad.

La Transformada de Fourier de $f(-t)$ es, según la definición:

$$\int_{-\infty}^{\infty} f(-t) \cdot \exp(-j \cdot \omega \cdot t) dt = \int_{-\infty}^{\infty} f(t) \cdot \exp(j \cdot \omega \cdot t) dt = F(-\omega)$$

Siendo $f(t)$ real:

$$\begin{aligned} F(-\omega) &= R(\omega) - j \cdot X(\omega) \\ f(t) &\longleftrightarrow R(\omega) + j \cdot X(\omega) & f(-t) &\longleftrightarrow R(\omega) - j \cdot X(\omega) \end{aligned}$$

Definiendo:

$$\begin{aligned} f_p(t) &= \frac{f(t) + f(-t)}{2} && \text{parte par de } f(t) \\ f_i(t) &= \frac{f(t) - f(-t)}{2} && \text{parte impar de } f(t) \end{aligned}$$

Resulta:

$$f_p(t) \longleftrightarrow R(\omega) \quad f_i(t) \longleftrightarrow j \cdot X(\omega)$$

Si además $f(t)=0$ para $t<0$, para $t>0$ se cumple $f(t) = 2 \cdot f_p(t) = 2 \cdot f_i(t)$.

LA TRANSFORMADA RAPIDA DE FOURIER

OBJETIVOS:

- Replantear el mecanismo de cálculo de la Transformada (y Transformada Inversa) Discreta de Fourier, de modo de hacer más eficiente su cómputo. Esto es, establecer los fundamentos del algoritmo de la Transformada Rápida de Fourier (FFT).
- Elaborar un Diagrama de Flujo que permita un mecanizado racional del algoritmo que sea lo más general posible. Pasando por el análisis, utilizando la teoría de grafos, desde un ejemplo particular.

22 - LA TRANSFORMADA RAPIDA DE FOURIER

Considérese la Transformada Discreta de Fourier (DFT):

$$X(n) = \sum_{k=0}^{N-1} x_0(k) \cdot \exp\left(\frac{-j \cdot 2 \cdot \pi \cdot n \cdot k}{N}\right) \quad n=0..N-1 \quad (22-1)$$

donde se ha reemplazado $k \cdot T$ por k y $n/N \cdot T$ por n por conveniencia de notación. Se percibe que la ecuación (22-1) describe el cómputo de N ecuaciones. Por ejemplo, si $N = 4$ y si se hace:

$$W = \exp\left(\frac{-j \cdot 2 \cdot \pi}{N}\right) \quad (22-2)$$

la expresión (22.1) puede ser escrita como:

$$\begin{aligned} X(0) &= x_0(0) \cdot W^0 + x_0(1) \cdot W^0 + x_0(2) \cdot W^0 + x_0(3) \cdot W^0 \\ X(1) &= x_0(0) \cdot W^0 + x_0(1) \cdot W^1 + x_0(2) \cdot W^2 + x_0(3) \cdot W^3 \\ X(2) &= x_0(0) \cdot W^0 + x_0(1) \cdot W^2 + x_0(2) \cdot W^4 + x_0(3) \cdot W^6 \\ X(3) &= x_0(0) \cdot W^0 + x_0(1) \cdot W^3 + x_0(2) \cdot W^6 + x_0(3) \cdot W^9 \end{aligned} \quad (22-3)$$

Las ecuaciones anteriores pueden ser más fácilmente representadas en forma matricial:

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 \\ W^0 & W^2 & W^4 & W^6 \\ W^0 & W^3 & W^6 & W^9 \end{bmatrix} \cdot \begin{bmatrix} x_0(0) \\ x_0(1) \\ x_0(2) \\ x_0(3) \end{bmatrix} \quad (22-4)$$

o más compactamente:

$$X(n) = W^{n \cdot k} \cdot x_0(k) \quad (22-5)$$

El examen de (22-4) revela que ya que W y posiblemente $x_0(k)$ sean complejas, entonces son necesarias N^2 multiplicaciones complejas y $N \cdot (N-1)$ adiciones para realizar el cómputo matricial requerido.

La FFT debe su éxito al hecho que el algoritmo reduce el número de multiplicaciones y adiciones requeridas en el cálculo de (22-4). Ahora se discutirá, a nivel intuitivo, como se lleva a cabo esta reducción.

23 - DESARROLLO INTUITIVO

Para ilustrar el algoritmo FFT, es conveniente elegir el número de puntos muestras de $x_0(k)$ de acuerdo a la relación $N = 2^Y$, donde Y es un entero. Recuérdese que la ecuación (22-4) resulta de la elección $N = 4 = 2^2$ por lo tanto, se puede aplicar la FFT al cómputo de (22-4).

El primer paso en el desarrollo del algoritmo FFT para este ejemplo es reescribir (22.4) como:

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W^1 & W^2 & W^3 \\ 1 & W^2 & W^0 & W^2 \\ 1 & W^3 & W^2 & W^1 \end{bmatrix} \cdot \begin{bmatrix} x_0(0) \\ x_0(1) \\ x_0(2) \\ x_0(3) \end{bmatrix} \quad (23-1)$$

La matriz de la ecuación (23-1) se deriva de la (22-4) usando la relación:

$$W^{n \cdot k} = W^{\text{mod}(n \cdot k, N)}$$

Recuérdese que $\text{mod}(n \cdot k, N)$ es el resto de la división entre $n \cdot k$ y N , por lo tanto si $N=4$, $n=2$ y $k=3$ entonces:

$$W^6 = W^2 \quad (23-2)$$

ya que

$$\begin{aligned} W^{n \cdot k} &= W^6 = \exp\left[\frac{-j \cdot 2 \cdot \pi}{4} \cdot (6)\right] = \exp(-j \cdot 3 \cdot \pi) = \exp(-j \cdot \pi) \\ &= \exp\left[\frac{-j \cdot \pi \cdot 2}{4} \cdot (2)\right] = W^2 = W^{\text{mod}(n \cdot k, N)} \end{aligned} \quad (23-3)$$

El segundo paso en el desarrollo es factorizar la matriz cuadrada en (23-1) como sigue:

$$\begin{bmatrix} X(0) \\ X(2) \\ X(1) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & W^0 & 0 & 0 \\ 1 & W^2 & 0 & 0 \\ 0 & 0 & 1 & W^1 \\ 0 & 0 & 1 & W^3 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & W^0 & 0 \\ 0 & 1 & 0 & W^0 \\ 1 & 0 & W^2 & 0 \\ 0 & 1 & 0 & W^2 \end{bmatrix} \cdot \begin{bmatrix} x_0(0) \\ x_0(1) \\ x_0(2) \\ x_0(3) \end{bmatrix} \quad (23-4)$$

Se puede demostrar fácilmente que el producto de las dos matrices cuadradas da por resultado la matriz cuadrada de (23-1). **Con la excepción que las filas 1 y 2 han sido intercambiadas.** Note que el intercambio ha sido tomado en cuenta en (23-4) reescribiendo el vector columna $X(n)$, se denota el vector con la fila intercambiada por:

$$X_d(n) = \begin{bmatrix} X(0) \\ X(2) \\ X(1) \\ X(3) \end{bmatrix} \quad (23-5)$$

Esta factorización es la clave de la eficiencia del algoritmo FFT.

Habiendo aceptado el hecho que (23-4) es correcta, aunque los resultados están "revueltos", se puede observar el número de multiplicaciones requeridas para computar la ecuación. Hágase primero:

$$\begin{bmatrix} x_1(0) \\ x_1(1) \\ x_1(2) \\ x_1(3) \end{bmatrix} = \begin{bmatrix} 1 & 0 & W^0 & 0 \\ 0 & 1 & 0 & W^0 \\ 1 & 0 & W^2 & 0 \\ 0 & 1 & 0 & W^2 \end{bmatrix} \cdot \begin{bmatrix} x_0(0) \\ x_0(1) \\ x_0(2) \\ x_0(3) \end{bmatrix} \quad (23-6)$$

Esto es, el vector columna $x_1(k)$ es igual al producto entre el vector y la matriz a la derecha de la ecuación (23-4).

El elemento $x_1(0)$ es computado por una multiplicación compleja y una adición compleja (W^0 no es reducido a la unidad para desarrollar un resultado general).

$$x_1(0) = x_0(0) + W^0 \cdot x_0(2) \quad (23-7)$$

El elemento $x_1(1)$ está también determinado por una multiplicación compleja y una adición. Sólo una adición compleja se realiza para computar $x_1(2)$. Esto deviene del hecho que $W^0 = -W^2$.

Por lo tanto

$$x_1(2) = x_0(0) + W^2 \cdot x_0(2) = x_0(0) - W^0 \cdot x_0(2) \quad (23-8)$$

donde la multiplicación compleja $\mathbf{W}^0 \cdot \mathbf{x}_0(2)$ ya ha sido computada en la determinación de $\mathbf{x}_1(0)$.

Por el mismo razonamiento, $\mathbf{x}_1(3)$ es computada por sólo por una adición compleja y ninguna multiplicación. El vector intermedio $\mathbf{x}_1(\mathbf{k})$ es entonces determinado por cuatro adiciones y dos multiplicaciones complejas.

$$\begin{bmatrix} X(0) \\ X(2) \\ X(1) \\ X(3) \end{bmatrix} = \begin{bmatrix} x_2(0) \\ x_2(1) \\ x_2(2) \\ x_2(3) \end{bmatrix} = \begin{bmatrix} 1 & W^0 & 0 & 0 \\ 1 & W^2 & 0 & 0 \\ 0 & 0 & 1 & W^1 \\ 0 & 0 & 1 & W^3 \end{bmatrix} \cdot \begin{bmatrix} x_1(0) \\ x_1(1) \\ x_1(2) \\ x_1(3) \end{bmatrix} \quad (23-9)$$

El elemento $\mathbf{x}_2(0)$ está determinado por una multiplicación y una adición complejas:

$$x_2(0) = x_1(0) + W^0 \cdot x_1(1) \quad (23-10)$$

El elemento $\mathbf{x}_2(1)$ está determinado por una adición compleja porque

$$W^0 = -W^2.$$

Con un razonamiento similar $\mathbf{x}_2(2)$ está determinado por una multiplicación compleja y una adición y $\mathbf{x}_2(3)$ por sólo una adición.

Luego el cómputo de (23-4) requiere un total de cuatro multiplicaciones complejas y ocho sumas complejas. En cambio el cómputo de (22-4) requiere dieciséis multiplicaciones complejas y 12 sumas complejas.

Nótese que el proceso de factorización de la matriz introduce ceros en las matrices factoreadas y, como resultado, reduce el número requerido de multiplicaciones. Para este ejemplo, el proceso de factorización de matriz ha reducido el número requerido de multiplicaciones en un factor de dos.

Ya que el tiempo de cómputo es fuertemente gobernado por el número de multiplicaciones, se ve la razón de la eficiencia del algoritmo FFT.

Para $N = 2^{\gamma}$ el algoritmo FFT es entonces simplemente un proceso de factorización de una matriz $N \times N$ en γ matrices (cada una de $N \times N$) de modo tal que cada una de las matrices factoreadas tenga una propiedad especial de minimizar el número de multiplicaciones y adiciones.

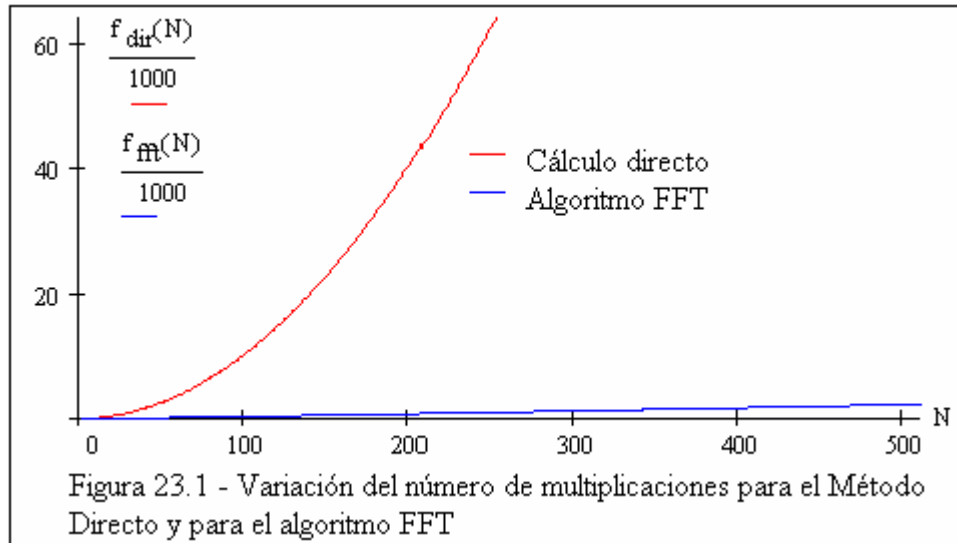
Si se extiende el resultado del ejemplo previo, se nota que la FFT requiere $N \cdot \gamma / 2 = 4$ multiplicaciones complejas y $N \cdot \gamma = 8$ adiciones complejas, mientras que el método directo requiere N^2 multiplicaciones complejas y $N \cdot (N-1)$ adiciones complejas.

Si se supone que el tiempo de cómputo es proporcional al número de multiplicaciones, entonces la razón aproximada de tiempo de cómputo del método directo al de la FFT está dado por:

$$\frac{N^2}{N \cdot \frac{\gamma}{2}} = \frac{2 \cdot N}{\gamma} \quad (23-11)$$

la cual para $N = 1024 = 2^{10}$ es una reducción computacional de más de 200 a 1. La Figura 23.1 ilustra la relación entre el número de multiplicaciones requeridas usando el algoritmo FFT comparada con el número de multiplicaciones del método directo.

$$f_{\text{dir}}(N) := N^2 \quad f_{\text{fft}}(N) := \frac{N}{2} \cdot \frac{\ln(N)}{\ln(2)} \quad N := 2 \dots 1024$$



El procedimiento de factorización introduce una discrepancia. Recuérdese que el cómputo de (23-4) produce:

$$X_d(n) \quad \text{en vez de} \quad X(n)$$

este reordenamiento es inherente al proceso de factorización de la matriz y es un problema menor debido a que es sencillo generalizar una técnica de "reordenamiento" de $X_d(n)$ para obtener $X(n)$.

Reescribiendo $X_d(n)$ reemplazando el argumento n con sus equivalentes binarios:

$$\begin{bmatrix} X(0) \\ X(2) \\ X(1) \\ X(3) \end{bmatrix} \quad \text{se convierte en} \quad \begin{bmatrix} X(00) \\ X(10) \\ X(01) \\ X(11) \end{bmatrix} \quad (23-12)$$

Obsérvese que si los argumentos binarios de (18) son "flippeados" o "bit reversed" (es decir, 10 se transforma en 01, etc.) En otras palabras, la imagen del espejo del binario. entonces

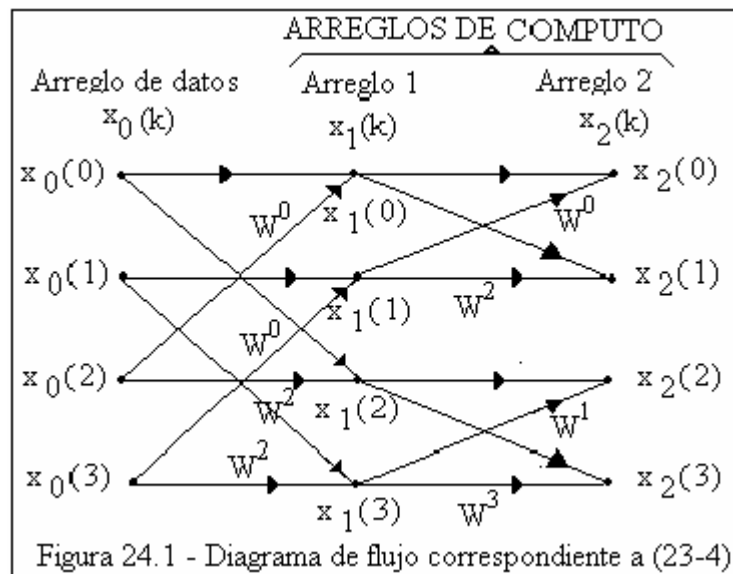
$$X_d(n) = \begin{bmatrix} X(00) \\ X(10) \\ X(01) \\ X(11) \end{bmatrix} \quad \text{"flikea a"} \quad \begin{bmatrix} X(00) \\ X(01) \\ X(10) \\ X(11) \end{bmatrix} = X(n) \quad (23-13)$$

Es sencillo desarrollar un resultado generalizado para "reordenar" la FFT.

Para N mayor que 4, es voluminoso describir el proceso de factorización de la matriz análoga a (23-4). Por esta razón es conveniente interpretar a (23-4) de manera gráfica. Usando esta formulación gráfica se puede describir en forma suficientemente generalizada un diagrama de flujo para un programa de computadora.

24 - GRAFICO DE FLUJO DE SEÑAL

Se convertirá la expresión (23-4) en un gráfico de flujo como el de la Figura 24.1.



Como se muestra, se representa el vector de datos o arreglo $x_0(k)$ por una columna vertical de nodos sobre la izquierda del gráfico. El segundo arreglo vertical de nodos es el vector $x_1(k)$ computado en la ecuación (23-6), y el próximo arreglo corresponde el vector $x_0(k) = X_d(n)$, ecuación (23-9). En general, se computarán γ arreglos, donde

$$N = 2^{\gamma}$$

El flujo de señal se interpreta como sigue. A cada nudo ingresan dos líneas sólidas representando "sendas de transmisión" desde nudos previos. Una senda transmite o trae una cantidad desde un nodo en un arreglo, multiplica la cantidad por W^p , y saca el resultado por el nudo del próximo arreglo. El factor W^p aparece cerca de la flecha de la senda de transmisión, la ausencia de este factor implica que $W^p = 1$.

Los resultados entrantes en un nudo desde dos sendas de transmisión son combinados aditivamente.

Para ilustrar la interpretación del gráfico de flujo de señal, considérese el nudo $x_1(2)$ de la Figura 24.1.

De acuerdo a las reglas para interpretar el gráfico de flujo de señal:

$$x_1(2) = x_0(0) + W^2 \cdot x_0(2) \quad (24.1)$$

lo cual es simplemente la ecuación (23-8). Cada nodo del gráfico de flujo de señal es expresado similarmente.

Este método es entonces un método conciso para representar los cálculos requeridos en la matriz factorizada del algoritmo FFT (23-4). Cada columna del gráfico corresponde a una matriz factorizada; se requieren γ arreglos verticales de N puntos cada uno ($N=2^Y$). La utilización de esta presentación gráfica permite fácilmente describir el proceso de factorización para N grande.

Se muestra en gráfico aparte (Figura 24.2) el diagrama de flujo de señal para $N=16$.

Con un diagrama de flujo de señal de este tamaño, es posible desarrollar las propiedades concernientes al proceso de factorización de la matriz y así proveer de un marco para el desarrollo de un diagrama de flujo para un programa de computadora que aplique el algoritmo FFT.

25 - NODOS DUALES

La inspección de la Figura 24.2 revela que en cada arreglo siempre se pueden encontrar dos nodos cuyas entradas de senda de transmisión provienen del mismo par de nudos en el arreglo previo. Por ejemplo, los nodos $x_1(0)$ y $x_1(8)$ son computados en términos de los nodos $x_0(0)$ y $x_0(8)$. Note que los nodos $x_0(0)$ y $x_0(8)$ no entran en el cómputo de cualquier otro nodo. Se define a tales nodos como **Par de Nodos Duales**.

Ya que el cómputo de un par de nodos duales es independiente de los otros nodos, es posible realizar cómputo "in-place". Para ilustrar, nótese de la Figura 24.2 que se pueden calcular simultáneamente $x_1(0)$ y $x_1(8)$ en términos de $x_0(0)$ y $x_0(8)$ y retornar el resultado a las locaciones de almacenamiento previamente ocupadas por $x_0(0)$ y $x_0(8)$. Los requerimientos de almacenaje están entonces limitados solamente al arreglo de datos $x_0(k)$. A medida que cada arreglo es computado, los resultados son retornados a este arreglo.

26 - ESPACIAMIENTO DE NODOS DUALES

Se investigará ahora el espaciamiento (medido verticalmente en términos del índice k) entre un par de nodos duales. La siguiente discusión se referirá a la Figura 24.2.

Primero, en el arreglo $I=1$, un par de nodos duales, como $x_1(0)$ y $x_1(8)$, están separados por $k=8=N/2^I=2^1$.

En el arreglo $I=2$, un par de nodos duales, como $x_2(8)$ y $x_2(2)$, están separados por

$$k=4=N/2^I=2^2$$

Similarmente, un par de nodos duales, como $x_3(4)$ y $x_3(6)$, en el arreglo $I=3$, están separados $k=2=N/2^I=2^3$.

Generalizando estos resultados, se observa que el espaciamiento entre nodos duales en el arreglo I está dado por $N/2^I$. Así, si se considera un nodo en particular $x_1(k)$, entonces su nodo dual es $x_1(k+N/2)$. Esta propiedad permite fácilmente identificar un par de nodos duales.

27 - COMPUTO DE NODO DUAL

El cómputo de un par de nodos duales requiere sólo una multiplicación compleja.

Para clarificar este punto, considérese el nodo $x_2(8)$ y su dual $x_2(12)$ ilustrados en la Figura 24.2. Las sendas de transmisión provenientes del nodo $x_1(12)$ están multiplicadas por W^4 y W^{12} antes de ingresar en los nudos $x_2(8)$ y $x_2(12)$, respectivamente.

Es importante notar que $W^4 = -W^{12}$ y que sólo una de las multiplicaciones requeridas desde el mismo dato $x_1(12)$ está para ser multiplicada por estos factores. En general, si el factor ponderante en el nodo es $W^p = -W^{p-N/2}$, sólo se requiere una multiplicación en el cómputo de un par de nodos duales. El cómputo de cualquier par de nodos duales está dado por el par de ecuaciones:

$$\begin{aligned} x_1(k) &= x_{1-1}(k) + W^p \cdot x_{1-1}\left(k + \frac{N}{2}\right) \\ x_1\left(k + \frac{N}{2}\right) &= x_{1-1}(k) - W^p \cdot x_{1-1}\left(k + \frac{N}{2}\right) \end{aligned} \quad (27-1)$$

En el cómputo de un arreglo, normalmente se comienza con el nodo $k=0$ y secuencialmente se trabaja hacia abajo del arreglo, computando el par de ecuaciones (27-1). Como se estableció previamente, el dual de cualquier nodo en el l -ésimo arreglo está siempre $N/2^l$ abajo en el arreglo. Ya que el espaciamiento es $N/2$, entonces se deduce que se deben saltar después cada $N/2^l$.

En la Figura 27.1, si se comienza con el nodo $k=0$, entonces de acuerdo a la discusión previa, el nodo dual está localizado en $k=N/2^1=4$ lo cual puede ser verificado por inspección de la figura. Procediendo hacia abajo en este arreglo, se nota que el nodo dual está siempre localizado 4 posiciones por debajo en el arreglo hasta alcanzar el nodo 4.

En este punto se ha ingresado un conjunto de nodos previamente encontrados; esto es, estos nodos son los nodos duales $k=0, 1, 2$ y 3 . Es necesario saltar sobre los nodos $k=4, 5, 6$ y 7 . Los nodos $8, 9, 10$ y 11 siguen la convención original del nodo dual localizado 4 posiciones más abajo en el arreglo.

En general, si se trabaja desde la cima al fondo en el arreglo l , entonces se computará la ecuación (27-1) para los primeros $N/2^l$ nodos, salta a los próximos $N/2^l$, etc. Se detienen los "saltos" cuando se alcanza un índice de nodo superior a $N-1$.

28 - DETERMINACION DE WP

$$W^6$$

En base a las discusiones precedentes, se han definido las propiedades de cada arreglo con la excepción de los valores de p en (27-1). El valor p se determina p con el siguiente procedimiento:

- (a) Se escribe el índice k en forma binaria con γ bits,

- (b) Se desliza el número binario encintrado en a) $\gamma - 1$ bits a la derecha, llenándose las nuevas posiciones de bit abiertas a la izquierda con ceros y
- (c) Finalmente, se invierte el orden de los bits.

Este número "bit-reversed" es el p .

Para ilustrar el procedimiento, refiérase a la Figura 24.2 y considérese el nodo $x_3(8)$.

Ya que $\gamma = 4$, $k = 8$ y $l = 3$, entonces k en binario es 1000. Se desplaza este número $\gamma - l = 4 - 3 = 1$ lugares a la derecha llenando con ceros los lugares libres, resultando 0100. Entonces se revierte el orden de los bits produciéndose 0010 o el entero 2. Finalmente, el valor de p es 2.

Considérese ahora el procedimiento para implementar esta operación de reversión de bits. Se sabe que un número binario, por ejemplo $a_4a_3a_2a_1$ puede ser escrito en base 10 como $a_4 \cdot 2^3 + a_3 \cdot 2^2 + a_2 \cdot 2^1 + a_1 \cdot 2^0$. El número bit-reversed que se está intentando describir está dado entonces por $a_1 \cdot 2^3 + a_2 \cdot 2^2 + a_3 \cdot 2^1 + a_4 \cdot 2^0$. Si se encuentra una técnica para determinar los bits binarios a_4, a_3, a_2, a_1 , entonces se habrá definido la operación "bit-reversing".

Supóngase que M es un número binario igual a $a_4 a_3 a_2 a_1$. Divídase a M por 2, trúnquese y multiplíquese el resultado truncado por 2. Luego compútase $[a_4a_3a_2a_1 - 2(a_4a_3a_2)]$. Si el bit a_1 es cero, entonces esta diferencia será cero porque la división por 2, el truncamiento, y la subsecuente multiplicación por 2 no altera M .

Sin embargo, si el bit a_1 es 1, el truncamiento cambia el valor de M y la expresión diferencia de arriba no será cero. Se observa que con esta técnica se puede determinar si el bit a_1 es 0 ó 1.

Se puede verificar el bit a_2 de una forma similar. La expresión diferencia apropiada es $[a_4a_3a_2 - 2(a_4a_3)]$. Si esta diferencia es cero, entonces a_2 es cero. Los bits a_3 y a_4 se determinan de la misma manera. Este procedimiento formará las bases del desarrollo de un programa de computadora para bit-reversing.

29 - REORDENAMIENTO DE LA FFT

El paso final en el cómputo de la FFT es "reordenar" los resultados análogos a la ecuación (23-13). Recuérdese que el procedimiento para reordenar el vector $X_d(n)$ es para escribir n en binario y revertir o "flipear" el número binario. Se muestra en la Figura 29.1 el resultado de la operación bit-reversing; los términos $x_4(k)$ y $x_4(i)$ han sido simplemente intercambiados donde i es el entero obtenido por bit-reversing del entero k .

Nótese que una situación similar al concepto de nodo dual existe cuando se "reordena" el arreglo de salida. Si se procede hacia abajo en el arreglo, intercambiando $x(k)$ con el apropiado $x(i)$, eventualmente se encuentra un nodo el cual ha sido previamente intercambiado.

Por ejemplo, en la Figura 29.1, el nodo $k=0$ permanece en su ubicación, los nodos $k=1, 2$, y 3 son intercambiados con los nodos $8, 4$ y 12 respectivamente. El próximo nodo a ser considerado es el 4 , pero este nodo fue previamente intercambiado con el 2 .

Para eliminar la posibilidad de considerar un nodo que ha sido previamente intercambiado, simplemente chequéese para ver si i (el entero obtenido por bit-reversing de k) es menor que k . Si es así, esto implica que el nodo ha sido intercambiado por una

operación previa. Con este chequeo se puede asegurar un sencillo procedimiento de reordenamiento.

30 - DIAGRAMA DE FLUJO PARA EL COMPUTO DE LA FFT

Usando las propiedades discutidas para el gráfico de flujo de señal de la FFT, se puede fácilmente desarrollar una diagrama de flujo para programar el algoritmo sobre una computadora digital. Se sabe de las discusiones previas que primero se computa el arreglo $l=1$ arrancando por el nodo $k=0$ y trabajando hacia abajo en el arreglo. En cada nodo k se computa el par de ecuaciones (27-1) donde p está determinado por el procedimiento descripto. Se continúa hacia abajo en el arreglo computando el par de ecuaciones (27-1) hasta alcanzar la región de los nodos que deben ser saltadas.

Se salta sobre los nodos apropiados y se continúa hasta haber computado el arreglo entero. Luego se procede a computar los arreglos remanentes usando el mismo procedimiento. Finalmente se reordena el arreglo final para obtener los resultados deseados.

El diagrama de flujo de la Figura 30.1 sirve para elaborar un programa de computadora que realiza el algoritmo FFT.

La caja 1 describe los datos de entrada necesarios. El vector de datos $x_0(k)$ se supone que es un complejo y está indexado como $k = 0, 1, \dots, N-1$. Si $x_0(k)$ es real., entonces la parte imaginaria debería ser puesta a cero. El número de puntos muestra N debe satisfacer la relación $N = 2^Y$, con Y entero.

La inicialización de varios parámetros del programa se lleva a cabo en la caja 2. El parámetro l es el número de arreglo que está siendo considerado. Se arranca con el arreglo $l=1$.

El espaciamento entre los nodos duales está dado por el parámetro $N2$; para el arreglo $l=1$, $N2=N/2$ y es inicializado como tal. El parámetro $NU1$ es el desplazamiento a derecha requerido en la determinación del valor de p en la ecuación (27-1); es inicializado a $Y-1$. El índice k del arreglo es inicializado a $k=0$, así trabajará desde la cima y progresará hacia abajo en el arreglo.

La caja 3 chequea para ver si el arreglo l a ser computado es mayor que Y . De ser así, el programa bifurca a la caja 13 para reordenar los resultados computados por inversión de bit. Si todos los arreglos no han sido computados, se procede a la caja 4.

La caja 4 pone un contador $I=1$. Este contador monitorea el número de pares de nodos duales que han de ser considerados. Recuérdese que es necesario saltar ciertos nodos con el objeto de asegurar que los nodos previamente considerados no son encontrados una segunda vez. El contador I es el control para determinar cuándo el programa debe saltar.

Las cajas 5 y 6 realizan el cómputo de la ecuación (27-1). Ya que K y l han sido inicializados a 0 y 1 respectivamente, el nodo inicial considerado es el primer nodo del primer arreglo. Para determinar el factor p para este nodo, recuérdese que primero se debe escalar el número binario k a la derecha $Y-l$ bits. Para llevar a cabo esto, se computa el valor entero de $k/2^{Y-l} = k/2^{NU1}$, y se pone el resultado en M como se muestra en la caja 5. De acuerdo a los procedimientos para la determinación de p , se debe bit-reverse M , donde M está representado por $Y = NU$ bits.

La función $IBR(M)$ denotada en la caja 5 es una rutina especial para reversión de bit y será descripta más adelante.

La caja 6, es el cómputo de $(27-1)$. Se computa el producto $WP \cdot (x(k+N/2))$ y se asigna el resultado a una locación de almacenamiento temporario. Después se suma y se resta este término de acuerdo con $(27-1)$. El resultado es la salida del nodo dual.

Luego se procede hacia abajo en el arreglo en el próximo nodo. Como se muestra en la caja 7, k se incrementa en 1.

Para evitar el cómputo de un nodo dual que ha sido considerado previamente, se chequea en la caja 8 para determinar si el contador I es igual a $N/2$. Para el arreglo 1, el número de nodos que pueden ser considerados consecutivamente sin saltar es igual a $N/2=N2$. La caja 8 determina esta condición. Si I no es igual a $N2$, entonces se procede hacia abajo en el arreglo y se incrementa el contador I como se muestra en la caja 9. Recuérdese que k ya ha sido incrementado en la caja 7.

Las cajas 5 y 6 son entonces repetidas para el nuevo valor de k .

Si $I = N/2$ en la caja 8, entonces se sabe que se ha alcanzado un nodo previamente considerado. Luego se saltan $N2$ nodos poniendo $k=k+N2$. Debido a que k ya ha sido incrementado en 1 en la caja 7, es suficiente saltar a los nodos previamente considerados incrementando a k en $N2$.

Antes de realizar los cálculos indicados en las cajas 5 y 6 para el nuevo nodo $k=k+N2$, se debe chequear primero para ver que no se haya excedido el tamaño del arreglo. Como se muestra en la caja 11, si k es menor que $N-1$ (recuérdese que k está indexado desde 0 a $N-1$), se resetea el contador I a 1 en la caja 4 y se repiten las cajas 5 y 6.

Si $k > N - 1$ en la caja 11, se sabe que se debe proceder al próximo arreglo. Por lo tanto, como se muestra en la caja 12, I es indexado por 1. El nuevo espaciamiento $N2$ es simplemente $N2/2$ (recuérdese que el espaciamiento es $N/2^I$, $NU1$ es decrementado en 1 ($NU1$ es igual a $\gamma - 1$) y k es reseteado a 0. Luego se chequea la caja 3 para ver si todos los arreglos han sido computados. Si es así, se procede a reordenar el resultado final. Esta operación es realizada en las cajas 13 a 17.

La caja 13 revierte los bits del entero k para obtener el entero i . De nuevo, se usa la función de reversión de bit $IBR(k)$ la cual se explicará posteriormente. Recuérdese que para reordenar la FFT simplemente se intercambian $x(k)$ con $x(i)$. Esta manipulación es realizada por las operaciones indicadas en la caja 15. Sin embargo, antes de ingresar en la caja 15 es necesario determinar, como se muestra en la caja 14, si i es menor que o igual a k . Este paso es necesario para prohibir la alteración de los nodos previamente reordenados.

La caja 16 determina cuándo todos los nodos han sido reordenados y la caja 17 es simplemente un índice para k .

En la caja 18, se describe la lógica de la función de reversión de bit $IBR(k)$. Se ha implementado el procedimiento de reversión de bit descrito en un párrafo anterior.

Cuando se procede a implementar el diagrama de flujo de la figura adjunta en un programa de computadora, es necesario considerar las variables $x(k)$ y WP como números complejos y ellas deben ser manejadas acordeamente.

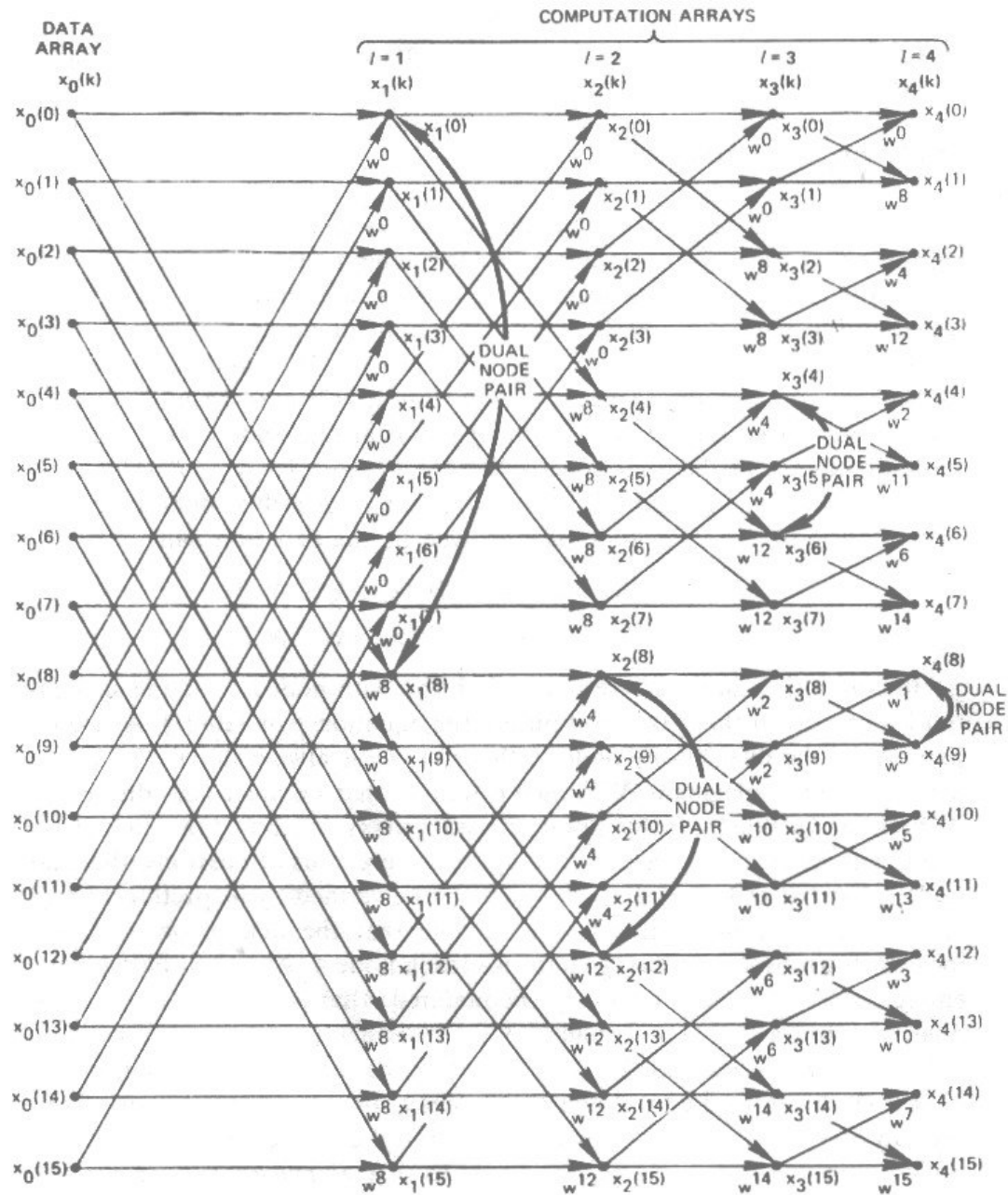


Figura 24.2

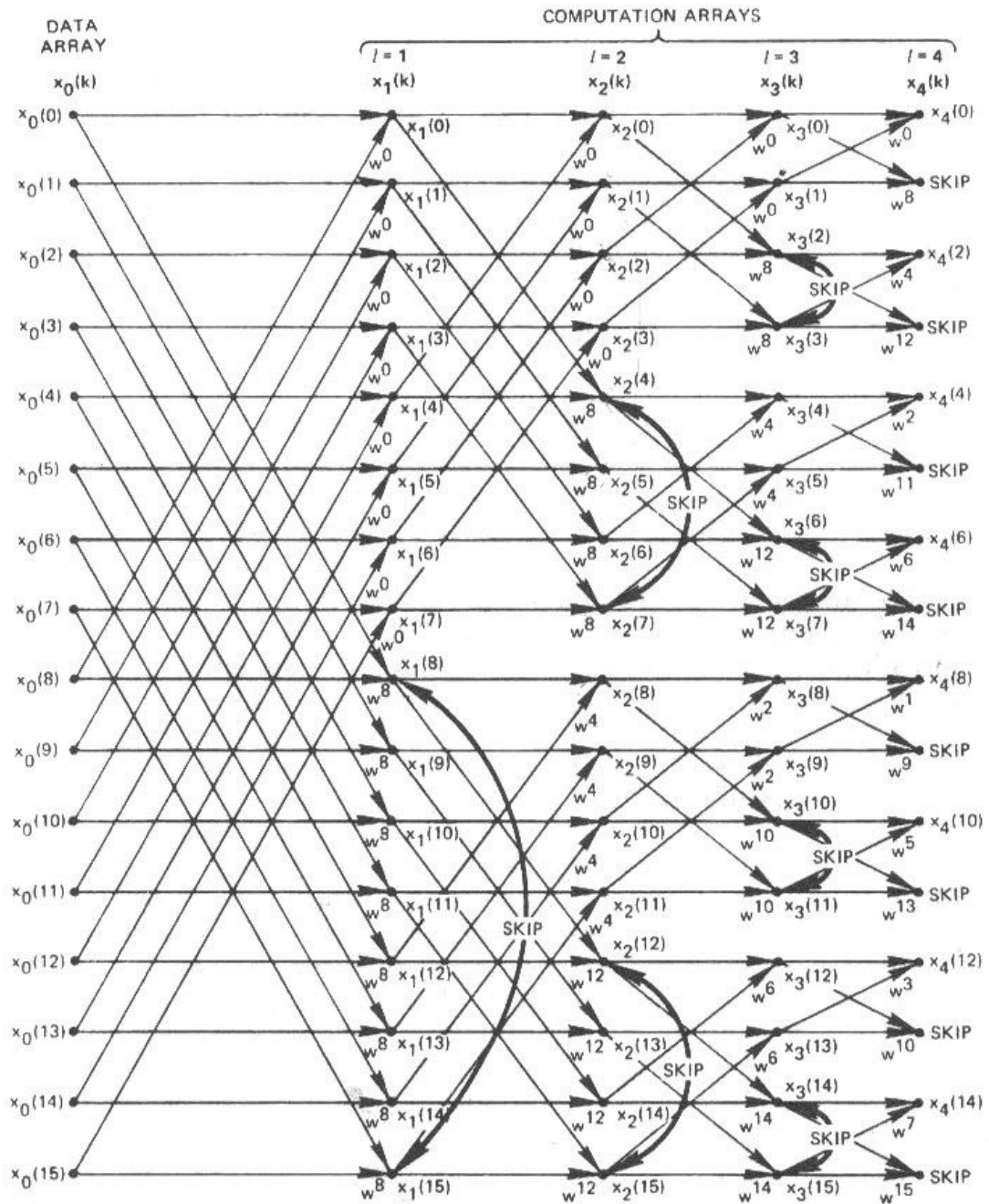


Figura 27.1

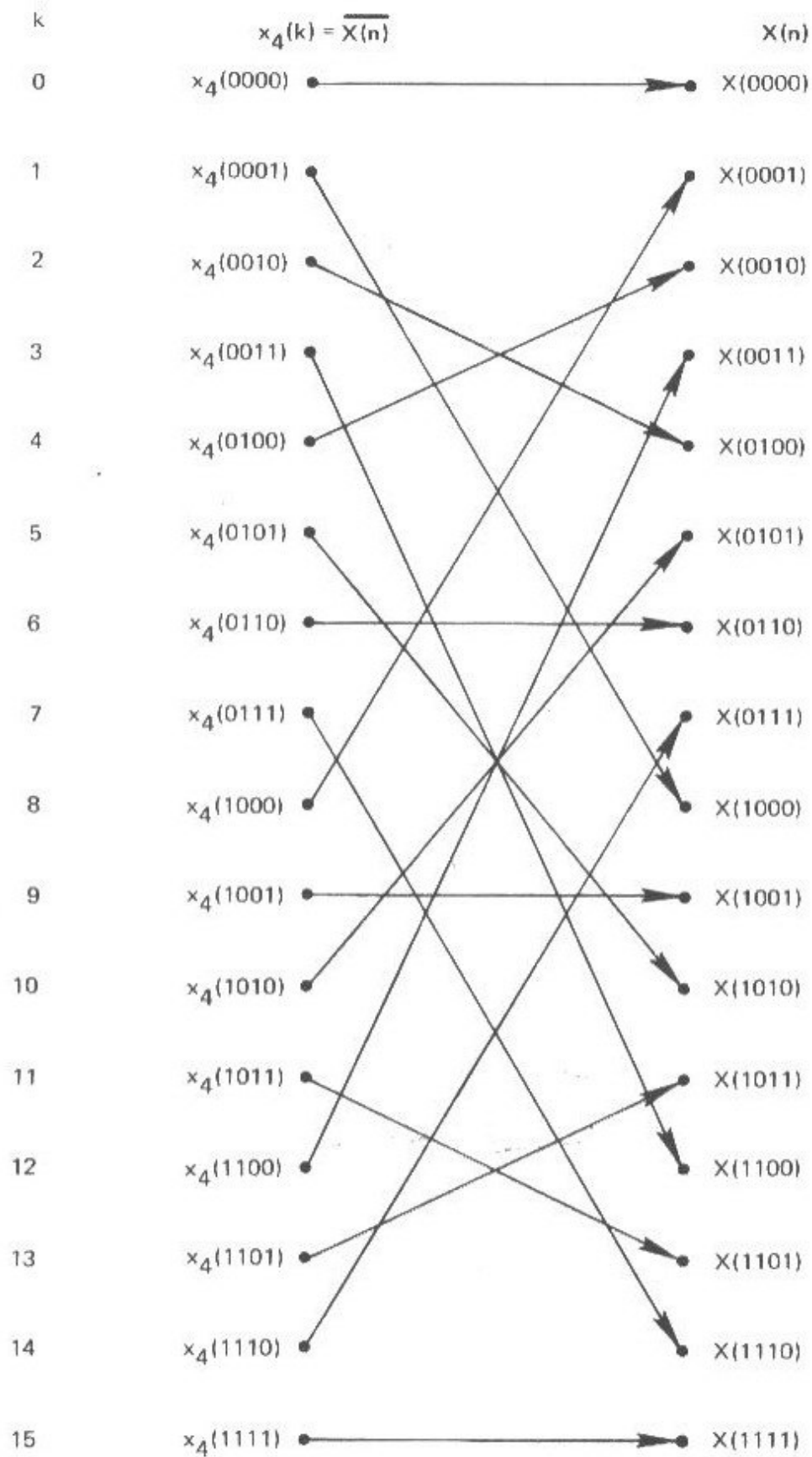


Figura 29.1

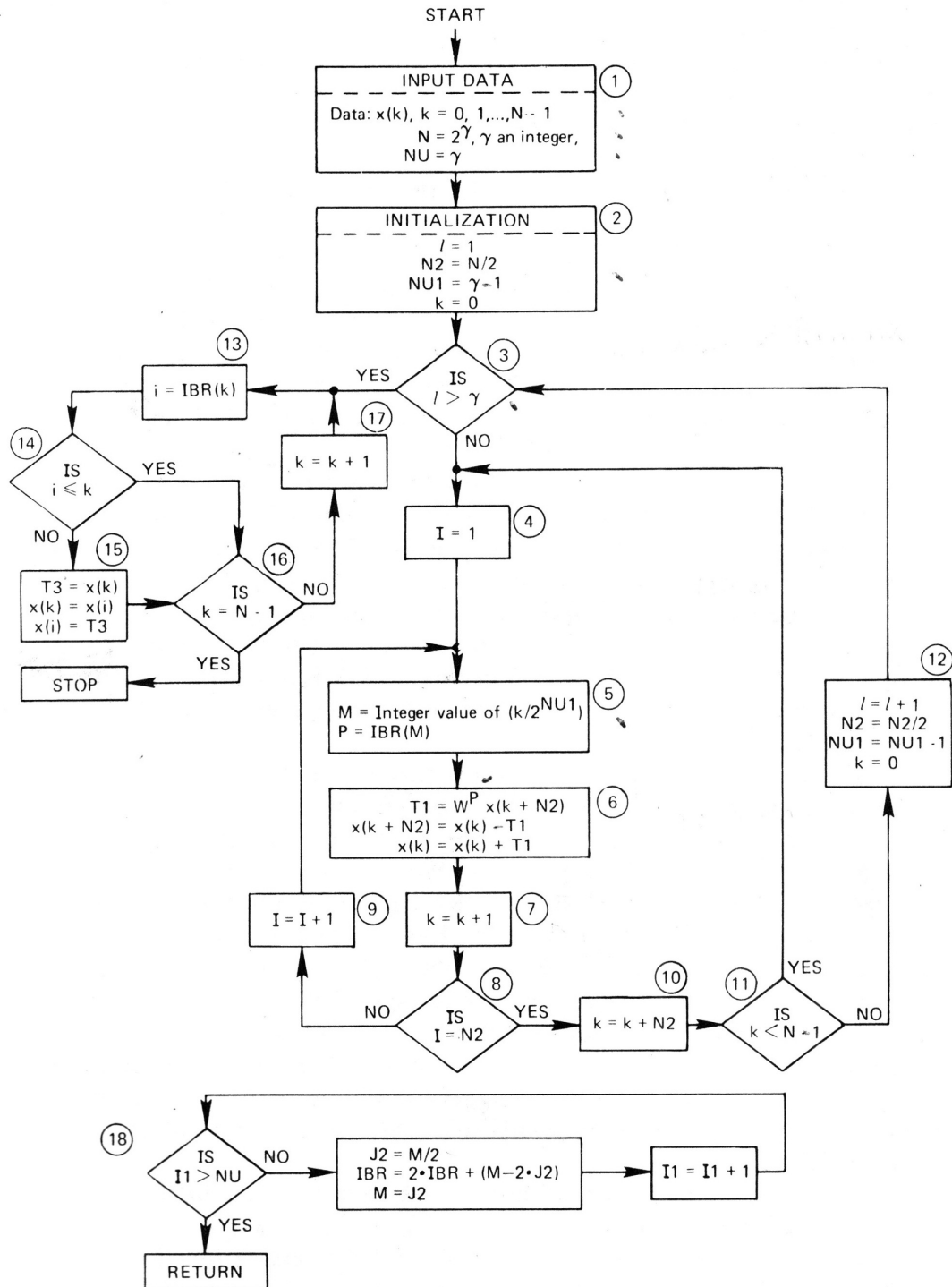


Figura 30.1

OBJETIVOS:

- Escribir un programa de computadora, en un lenguaje de Alto Nivel (en este caso, Basic) que permita aplicar el algoritmo de la Transformada Rápida de Fourier en trabajos de procesamiento de señales provenientes del mundo exterior, en tiempo real.

- Analizar las distintas funciones que tiene instrumentadas el Software Mathcad para resolver el algoritmo de la Transformada Rápida de Fourier, comparándolas con los resultados que entrega el programa producido en el objetivo anterior.

31 . PROGRAMA BASIC PARA CALCULO DE LA FFT

Llevando a cabo las indicaciones del diagrama de flujo de la Figura 30.1, se llega al siguiente programa BASIC que se combina con Mathcad para evaluación.

Se va a trabajar con el ejemplo de la exponencial decreciente a la que se le hallará la Transformada de Fourier.

$N := 32$	Número de muestras
$T := 0.25$	Periodo de muestreo
$k := 0..N - 1$	Índice
$b_k := \exp(-k \cdot T)$	$b_0 := 0.5$ Vector con la parte real de cada muestra
$c_k := 0$	Vector con la parte imaginaria de cada muestra

WRITEPRN(ent_real) := b
 Archivo ASCII que contiene la parte real de los datos

WRITEPRN(ent_imag) := c
 Archivo ASCII que contiene la parte imaginaria de los datos

Una vez que los datos se han incorporado en los archivos **ent_real.prn** y **ent_imag.prn** se ejecuta el siguiente programa BASIC:

```

5 pi = 4 * ATN(1): GOSUB 1000 'Pantalla
10 INPUT "Número de Muestras (N): "; N
15 INPUT "Periodo de muestreo (T): "; T
20 nu = INT(LOG(N) / LOG(2) + .0001)
30 DIM xr(N + 1), xi(N + 1), yr(N + 1), yi(N + 1)
40 ON a GOTO 2000, 3000, 4000, 5000
1000 CLS
1010 PRINT "Elija:  1 - Para fft"
1011 PRINT "      2 - Para ifft"
1012 PRINT "      3 - Para Convolución"
1013 PRINT "      4 - Para Correlación"
1020 INPUT a
1030 RETURN
2000 REM Rutina fft
2010 GOSUB 6000 'Ingreso de valores
2020 GOSUB 7000 'Núcleo fft
2030 GOTO 8000 'Graficación
3000 REM Rutina ifft
3010 GOSUB 6000
3020 FOR k = 0 TO N - 1: xi(k) = -xi(k): NEXT k

```

```
3030 GOSUB 7000
3040 FOR k = 0 TO N - 1: xr(k) = xr(k) / N: xi(k) = xi(k) / N: NEXT k
3050 GOTO 8000
4000 REM Rutina convolución
4010 GOSUB 6000 'Ingreso de valores de la primera Función
4020 GOSUB 7000 'Núcleo fft
4030 FOR k = 0 TO N - 1: yr(k) = xr(k): yi(k) = xi(k): NEXT k
4045 FOR k = 0 TO N - 1: xr(k) = 0: xi(k) = 0: NEXT k
4047 GOSUB 6000 'Ingreso de valores de la segunda Función
4050 GOSUB 7000 'Núcleo fft
4060 FOR k = 0 TO N - 1
4062 c1 = xr(k): xr(k) = xr(k) * yr(k) - xi(k) * yi(k)
4064 xi(k) = -c1 * yi(k) - yr(k) * xi(k)
4066 NEXT k
4070 GOSUB 7000
4080 FOR k = 0 TO N - 1: xr(k) = xr(k) / N: xi(k) = xi(k) / N: NEXT k
4090 GOTO 8000
5000 REM Rutina correlación
5010 GOSUB 6000 'Ingreso de valores de la primera Función
5020 GOSUB 7000 'Núcleo fft
5030 FOR k = 0 TO N - 1: yr(k) = xr(k): yi(k) = xi(k): NEXT k
5040 FOR k = 0 TO N - 1: xr(k) = 0: xi(k) = 0: NEXT k
5050 GOSUB 6000 'Ingreso de valores de la segunda función
5060 GOSUB 7000 'Núcleo fft
4075 FOR k = 0 TO N - 1: yi(k) = -xi(k): NEXT k
5080 FOR k = 0 TO N - 1
5082 c1 = xr(k): xr(k) = xr(k) * yr(k) - xi(k) * yi(k)
5084 xi(k) = -c1 * yi(k) - yr(k) * xi(k)
5086 NEXT k
5110 GOSUB 7000 'Halla ifft
5120 FOR k = 0 TO N - 1: xr(k) = xr(k) / N: xi(k) = xi(k) / N: NEXT k
5130 GOTO 8000 'Rutina Graficación
6000 REM Subrutina de ingreso de valores reales
6004 OPEN "c:\winmcad\ent_real.prn" FOR INPUT AS #1
6005 FOR k = 0 TO N - 1: INPUT #1, xr(k): NEXT k
6008 CLOSE #1
6010 REM Subrutina de ingreso de valores imaginarios
6020 OPEN "c:\winmcad\ent_imag.prn" FOR INPUT AS #1
6030 FOR k = 0 TO N - 1: INPUT #1, xi(k): NEXT k
6040 CLOSE #1
'6010 FOR k = 0 TO N - 1
'6020 PRINT "valor real "; k; " de la entrada": INPUT xr(k)
'6030 NEXT k
'6040 FOR k = 0 TO N - 1
'6050 PRINT "valor imaginario "; k; " de la entrada": INPUT xi(k)
'6060 NEXT k
6070 RETURN
7000 REM Núcleo fft
7010 L = 1: n2 = N / 2: n1 = nu - 1: k = 0
7020 IF L > nu THEN 7140
```

```

7030 i = 1
7040 M = INT(k / 2 ^ n1)
7050 i2 = 0: ib = 0: z = M: GOSUB 7200: P = ib
7060 ar = 2 * pi * P / N: c = COS(ar): s = SIN(ar): ag = k + n2
7070 tr = xr(ag) * c + xi(ag) * s: ti = xi(ag) * c - xr(ag) * s
7080 xr(ag) = xr(k) - tr: xi(ag) = xi(k) - ti
7090 xr(k) = xr(k) + tr: xi(k) = xi(k) + ti: k = k + 1
7100 IF i = n2 THEN 7110 ELSE i = i + 1: GOTO 7040
7110 k = k + n2
7120 IF k < N - 1 THEN 7030
7130 k = 0: n1 = n1 - 1: n2 = n2 / 2: L = L + 1: GOTO 7020
7140 i2 = 0: ib = 0: z = k: GOSUB 7200: i = ib
7150 IF i <= k THEN 7180
7160 t3 = xr(k): t4 = xi(k): xr(k) = xr(i): xi(k) = xi(i)
7170 xr(i) = t3: xi(i) = t4
7180 IF k = N - 1 THEN 7190 ELSE k = k + 1: GOTO 7140
7190 RETURN
7200 IF i2 < nu THEN 7210 ELSE RETURN
7210 j2 = INT(z / 2): ib = 2 * ib + z - 2 * j2: z = j2: i2 = i2 + 1: GOTO 7200
8000 REM Rutina Graficación
8002 OPEN "c:\winmcad\salida.prn" FOR OUTPUT AS #1
8004 FOR k = 0 TO N - 1: PRINT #1, xr(k) * T, xi(k) * T: NEXT k
8006 CLOSE #1: END
'8003 FOR k = 0 TO N - 1
'8005   yr(k) = SQR(xr(k) ^ 2 + xi(k) ^ 2)
'8006   PRINT USING "##.##"; xr(k); : PRINT " "; : PRINT USING "##.##"; xi(k)
'8007 NEXT k: END
8008 SCREEN 2: M = xr(0)
8010 FOR k = 0 TO N - 1
8020   IF M >= xr(k) THEN 8040
8030   M = xr(k)
8040 NEXT k
8050 q = 90 / M
8060 FOR k = 0 TO 255: PSET (k, 90): NEXT k
8070 FOR k = 0 TO N - 1: PSET (k * 245 / N, q * xr(k)): NEXT k
8105 j$ = INKEY$
8110 IF j$ = "" THEN 8105
8120 FOR k = 0 TO N - 1: xr(k) = yr(k): NEXT k: GOTO 8008

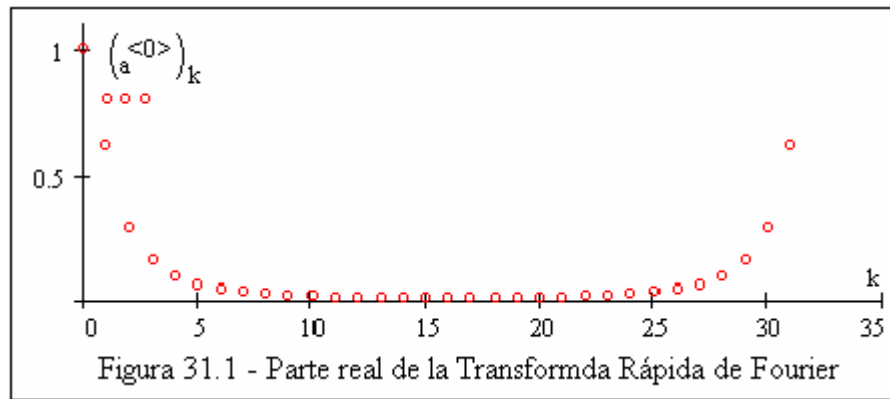
```

Una vez corrido el programa, devuelve el archivo **salida.prn** una matriz de dos columnas, donde la primera corresponde a la parte real de la transformada y la segunda a la parte imaginaria.

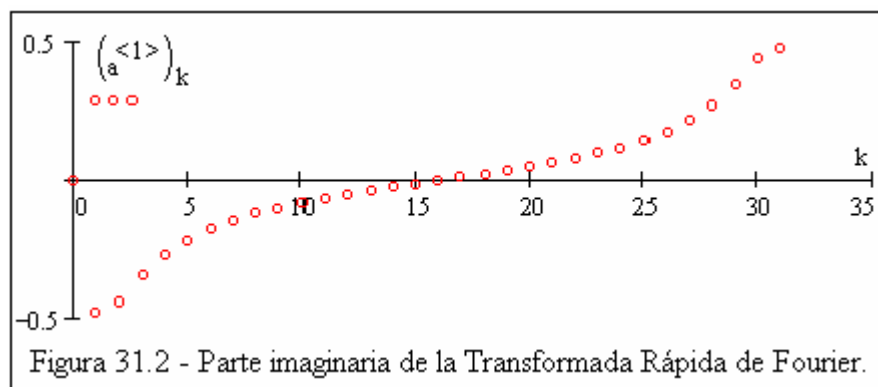
a := READPRN(salida)

Lectura de la matriz

En la Figura 31.1 se observa la parte real:



Y en la Figura 31.2, la parte imaginaria



Lo que coincide plenamente con el cálculo directo de la Transformada de Fourier.

TRANSFORMADA DE FOURIER CON MATHCAD

fft(v)

Retorna la TF de un vector. El resultado es un vector de $1+2^{m-1}$ elementos cuyo n-ésimo elemento está dado por

$$c_n = \frac{1}{\sqrt{N}} \cdot \sum_{k=0}^{N-1} v_k \cdot \exp\left(j \cdot 2 \cdot \pi \cdot \frac{k}{N} \cdot n\right)$$

Argumentos:

- v debe tener 2^m elementos ($m > 2$).
- Todos los elementos en v son reales.

Para vectores con valores complejos o con cualquier número de elementos, use **cfft** en lugar de **fft**.

Para TF bi-dimensionales , use **cfft**.

ifft(v)

Retorna la IFT de un vector. El resultado es un vector de $1+2^{m-1}$ elementos cuyo n-ésimo elemento está dado por

$$c_n = \frac{1}{\sqrt{N}} \cdot \sum_{k=0}^{N-1} v_k \cdot \exp\left(-j \cdot 2 \cdot \pi \cdot \frac{k}{N} \cdot n\right)$$

cfft(A)

Retorna la FT de un vector o matriz. El resultado tiene el mismo número de filas y columnas que A. Si A es un vector, el resultado está dado por:

$$c_n = \frac{1}{\sqrt{N}} \cdot \sum_{k=0}^{N-1} v_k \cdot \exp\left(j \cdot 2 \cdot \pi \cdot \frac{k}{N} \cdot n\right)$$

Argumentos:

- A puede ser un vector o una matriz.

Si...

- **A** es un vector tiene 2^m elementos, y..
- Todos los elementos en el vector son reales

...use **fft** en lugar de **cfft**. Para tales vectores, la segunda mitad del espectro es la imagen del espejo de la primera mitad y no necesitan ser calculados.

Para argumentos matriciales, **cfft** retorna la FT bidimensional:

icfft(B)

Retorna la IFT correspondiente a la **fft**.

Argumentos:

B puede ser una matriz o un vector.

FFT(v)

Retorna la FT de un vector. El resultado es un vector de $1+2^{m-1}$ elementos cuyo n-ésimo elemento está dado por

$$c_n = \sum_{k=0}^{N-1} v_k \cdot \exp\left(-j \cdot 2 \cdot \pi \cdot \frac{k}{N} \cdot n\right)$$

Argumentos:

- **v** debe tener 2^m elementos ($m > 2$).
- Todos los elementos en **v** son reales.

Para vectores con valores complejos o con cualquier número de elementos, use **CFFT** en lugar de **FFT**.

Para FT bi-dimensionales , use **CFFT**.

IFFT(u)

Retorna la IFT de un vector correspondiente a **FFT**. El resultado es un vector de 2^m elementos cuyo n-ésimo elemento está dado por

$$c_n = \sum_{k=0}^{N-1} v_k \cdot \exp\left(j \cdot 2 \cdot \pi \cdot \frac{k}{N} \cdot n\right)$$

Argumentos:

- **u** es un vector con $1+2^{m-1}$ elementos ($m > 2$).
- Todos los elementos en **u** son reales.

Se debe usar esta función sólo cuando los datos sean puramente reales. Si los datos son complejos se debe usar **ICFFT**.

CFFT(A)

Retorna la FT de un vector o matriz. El resultado tiene el mismo número de filas y columnas que **A**. Si **A** es un vector, el resultado está dado por:

$$c_n = \frac{1}{N} \cdot \sum_{k=0}^{N-1} v_k \cdot \exp\left(-j \cdot 2 \cdot \pi \cdot \frac{k}{N} \cdot n\right)$$

Argumentos:

- **A** puede ser un vector o una matriz.

Si...

- **A** es un vector tiene 2^m elementos, y..
- Todos los elementos en el vector son reales

...use **FFT** en lugar de **CFFT**. Para tales vectores, la segunda mitad del espectro es la imagen del espejo de la primera mitad y no necesitan ser calculados. Para argumentos matriciales, **CFFT** retorna la FT bidimensional:

ICFFT(B)

Retorna la IFT correspondiente a la CFFT.

Argumentos:

B puede ser una matriz o un vector.

32.1 - **Frecuencias correspondientes a los coeficientes:**

Tanto **fft** como **cfft** retornan vectores cuyos elementos son las amplitudes complejas de las distintas frecuencias comprendidas en la señal original. Para recuperar dichas frecuencias, se debe conocer:

- La frecuencia de muestreo de la señal original.
- El número de muestras en la señal original.

Dados estos parámetros, la frecuencia asociada con el n-ésimo elemento está dada por:

$$c_n = \frac{n}{N} \cdot f_s$$

Donde: f_s es la frecuencia de muestreo y N es el número de muestras.

32.2 - Aliasing:

Debido a que la fft es una aproximación discreta a la transformada continua, se pueden encontrar armónicas espúreas. Esto se llama Aliasing. Para evitar el aliasing:

- Asegurarse que la señal tiene ancho de banda finito
- Asegurarse que la frecuencia de muestreo es al menos el doble del ancho de banda (la frecuencia de Nyquist.)

32.3 - **Espectro de Potencia y de Fase:**

El Espectro de Potencia en el dominio de frecuencia está dado por:

$$(|F(\omega)|)^2 = \left(F(\omega) \cdot \overline{F(\omega)} \right)$$

El Espectro de Fase en el dominio de frecuencia está dado por:

$$\arg(|F(\omega)|) = \text{atan} \left(\frac{\text{Im}(F(\omega))}{\text{Re}(F(\omega))} \right)$$

Las funciones fft y cfft retornan las partes reales e imaginarias de la Transformada de Fourier. Se puede recuperar el espectro de potencia y el de fase a partir de estas usando el operador vectorización:

$$\overrightarrow{(|F(v)|)^2} \quad \text{espectro de potencia}$$

$$\overrightarrow{\arg(|F(v)|)} \quad \text{espectro de fase}$$

32.4 - fft/iff contra cfft/cifft:

Usar el par **fft/iff** si ambas de las siguientes sentencias son ciertas:

- Los datos en el dominio del tiempo son reales.
- El vector de datos tiene 2^m elementos. ($m > 2$).

Usar el par **cfft/icfft** si cualquiera de las siguientes sentencias son ciertas:

- Los datos en el dominio dle tiempo son complejos.

- El vector de datos no tiene 2^m elementos.
- Los datos son una matriz, no un vector.

Para datos reales en el dominio del tiempo, la FT tiene simetría conjugada. Debido a esto, **fft** caerá en una segunda mitad redundante. Este es el porqué el vector retornado por **fft** es de la mitad de tamaño del vector original.

La función **ifft** reconstruye esta mitad redundante. Por esta razón, se debería usar **fft** e **ifft** juntas. El par **cff/iccft** no hace tales suposiciones acerca de simetría. Aunque estas funciones no son tan eficientes para datos reales, no obstante pueden ser usadas.

33 - OPERACIONES POSIBLES DE REALIZAR CON EL ALGORITMO FFT

33.1 - Ejemplo de Transformada de Fourier de una señal

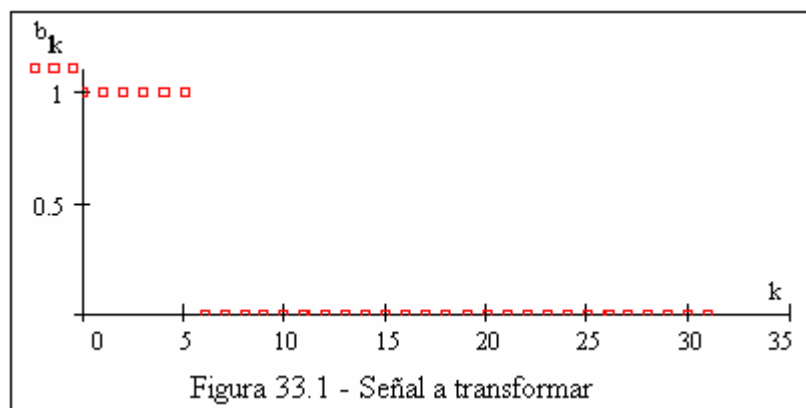
$N := 32$ Número de muestras

$k := 0..N - 1$ Índice

$b_k := \text{if}(k < 6, 1, 0)$ Vector con las muestras reales

$b1_k := 0$ Vector con las muestras imaginarias

Gráficamente, esta señal se observa en la Figura 33.1.



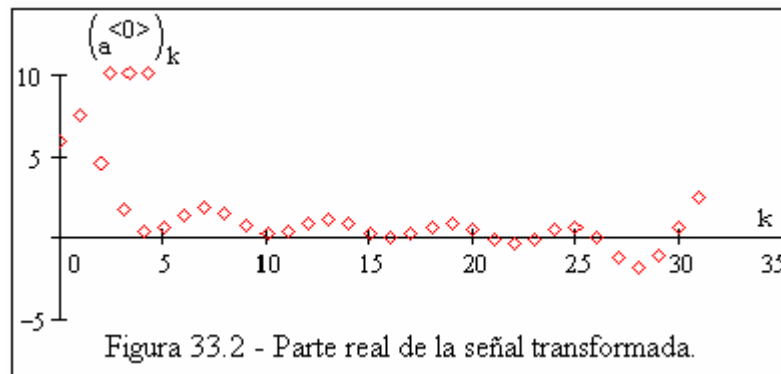
WRITEPRN(ent_real) := b Archivo ASCII que contiene los datos reales de la función

WRITEPRN(ent_imag) := b1 Archivo ASCII que contiene los datos imaginarios de la función

Una vez corrido el programa, devuelve el archivo **sal_fft.prn** una matriz de dos columnas, donde la primera corresponde a la parte real de la transformada y la segunda a la parte imaginaria.

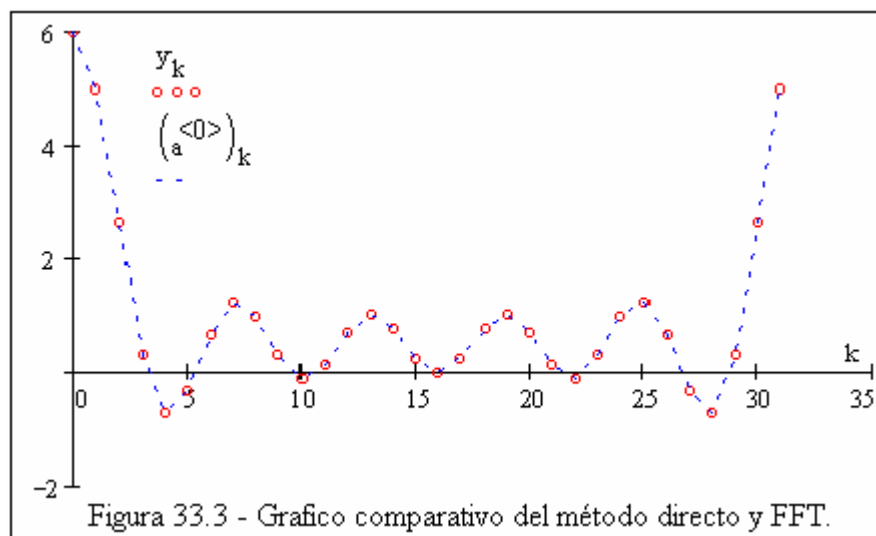
a := READPRN(sal_fft) Lectura de la matriz

En la Figura 33.2 se grafica la parte real de la Transformada.

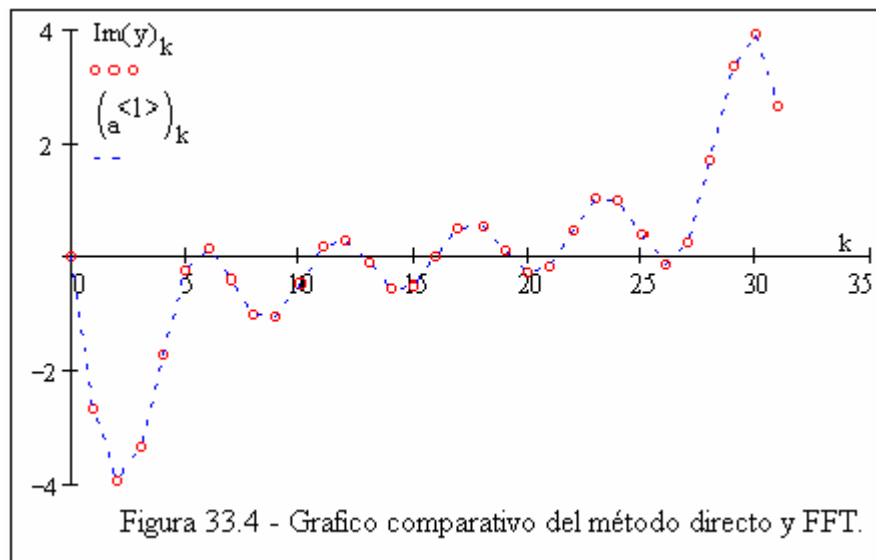


Lo que coincide plenamente con el cálculo directo de la Transformada de Fourier:

$$y_k := \sum_{i=0}^{N-1} b_i \cdot \exp\left(-j \cdot 2 \cdot \pi \cdot i \cdot \frac{k}{N}\right) \quad \text{Transformada de Fourier (directa)}$$



Si se analiza la parte imaginaria, Figura 33.4.



33.2 - Ejemplo de Transformada Inversa de Fourier de una señal

$N := 32$

Número de muestras

$k := 0..N - 1$

Índice

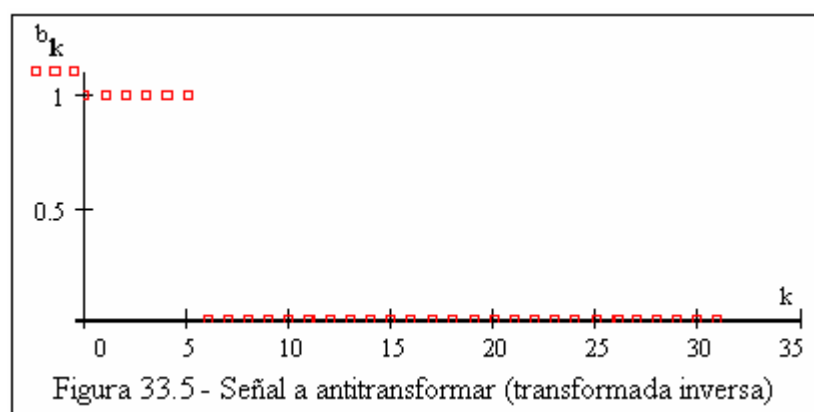
$b_k := \text{if}(k < 6, 1, 0)$

Vector con las muestras reales

$b1_k := 0$

Vector con las muestras imaginarias

La señal se puede apreciar gráficamente en la Figura 33.5.



`WRITEPRN(ent_real) := b`

Archivo ASCII que contiene los datos reales de la función

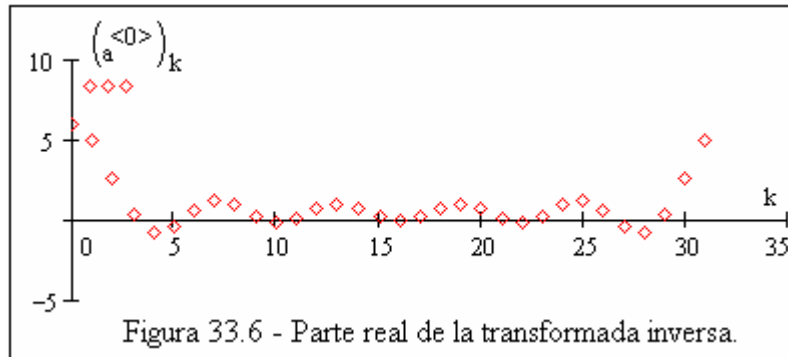
`WRITEPRN(ent_imag) := b1`

Archivo ASCII que contiene los datos imaginarios de la función

Una vez corrido el programa, devuelve el archivo **sal_ifft.prn** una matriz de dos columnas, donde la primera corresponde a la parte real de la transformada Inversa y la segunda a la parte imaginaria.

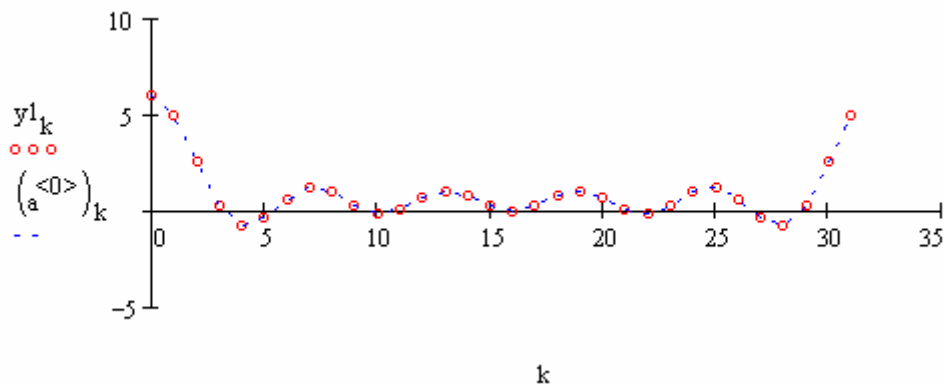
$a := \text{READPRN}(\text{sal_iff})$ Lectura de la matriz

En la Figura 33.6 se observa la parte real de la transformada inversa.

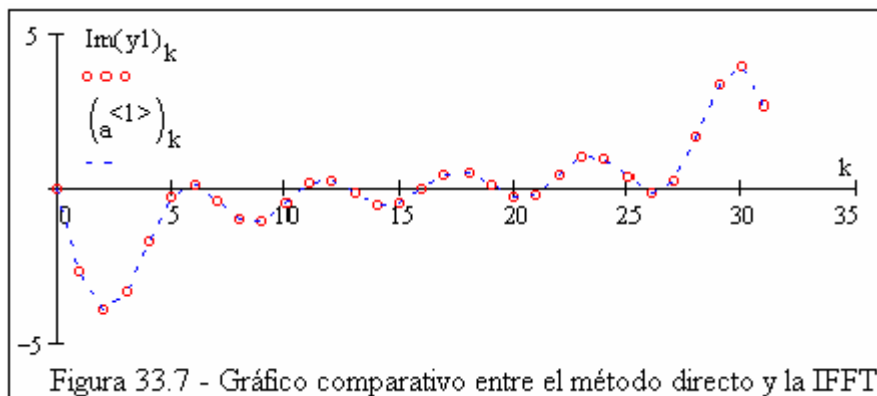


Lo que coincide plenamente con el cálculo directo de la Transformada Inversa de Fourier, como se ve en la Figura 33.7.

$$y1_k := \sum_{i=0}^{N-1} \overline{b_i} \cdot \exp\left(-j \cdot 2 \cdot \pi \cdot i \cdot \frac{k}{N}\right) \quad \text{Transformada Inversa de Fourier (directa)}$$



Si se analiza la parte imaginaria:



33.3 - Ejemplo de Convolución de dos señales

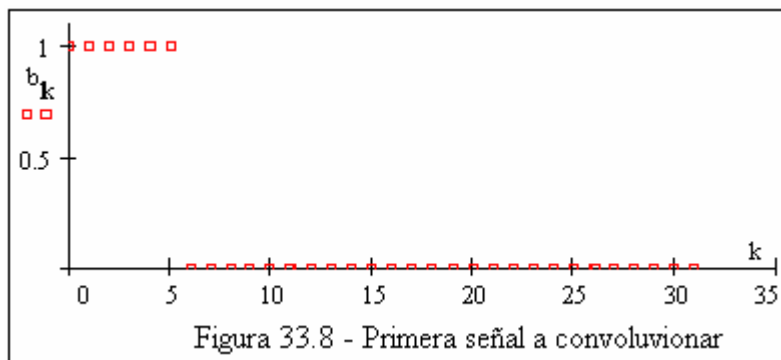
$N := 32$ Número de muestras

$k := 0..N - 1$ Índice

$b_k := \text{if}(k < 6, 1, 0)$ Vector con las muestras reales de la primera señal.

$b1_k := 0$ Vector con las muestras imaginarias de la primera señal

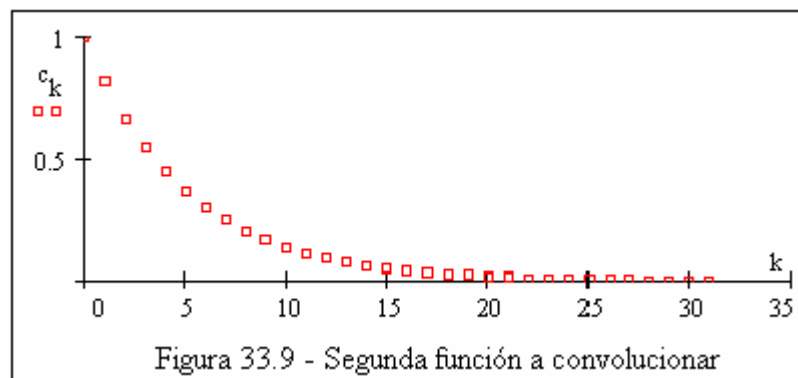
La primera señal a convolucionar se observa en la Figura 33.8.



$c_k := \exp(-k \cdot T)$ Vector con las muestras reales de la segunda señal.

$c1_k := 0$ Vector con las muestras imaginarias de la segunda señal.

La segunda señal a convolucionar se observa en la Figura 33.9.



WRITEPRN(ent_rea1) := b Archivo ASCII que contiene los datos reales de la primera señal

WRITEPRN(ent_ima1) := b1 Archivo ASCII que contiene los datos imaginarios de la primera señal

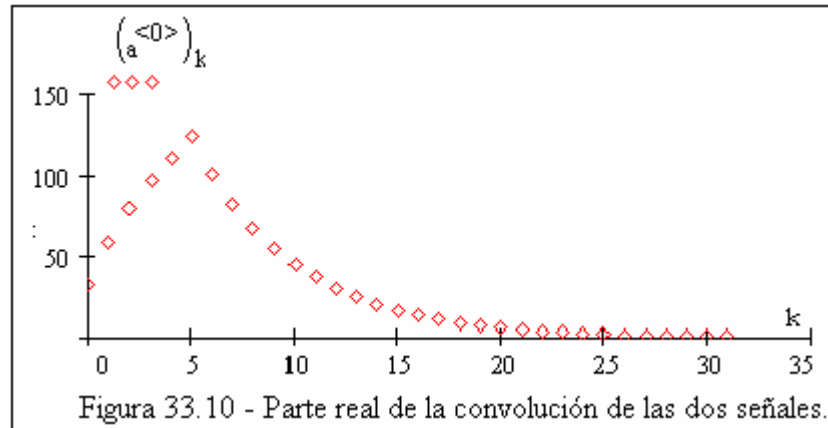
WRITEPRN(ent_rea2) := c Archivo ASCII que contiene los datos reales de la segunda señal

WRITEPRN(ent_ima2) := c1 Archivo ASCII que contiene los datos
imaginarios de la segunda señal

Una vez ejecutado el programa, devuelve el archivo **sal_conv.prn.** una matriz de dos columnas, donde la primera corresponde a la parte real de la transformada y la segunda a la parte imaginaria.

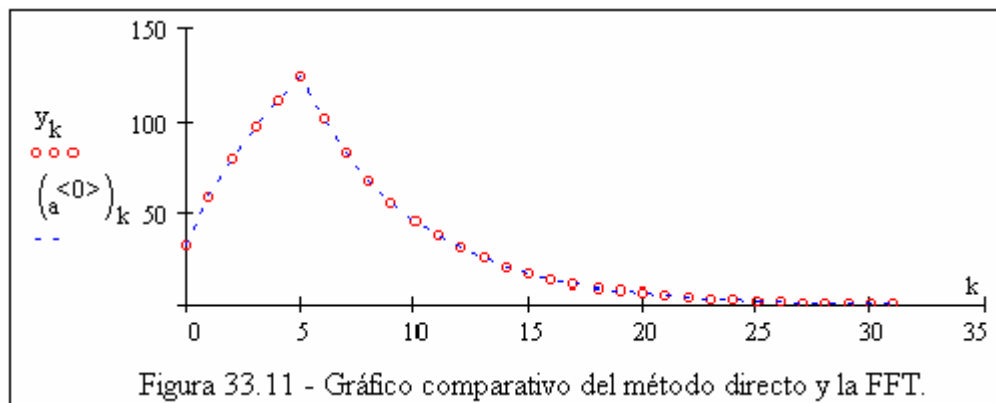
a := READPRN(sal_conv) Lectura de la matriz

Si se observa la parte real, figura 33.10.



Lo que coincide plenamente con el cálculo directo de la Convolución, Figura 33.12.

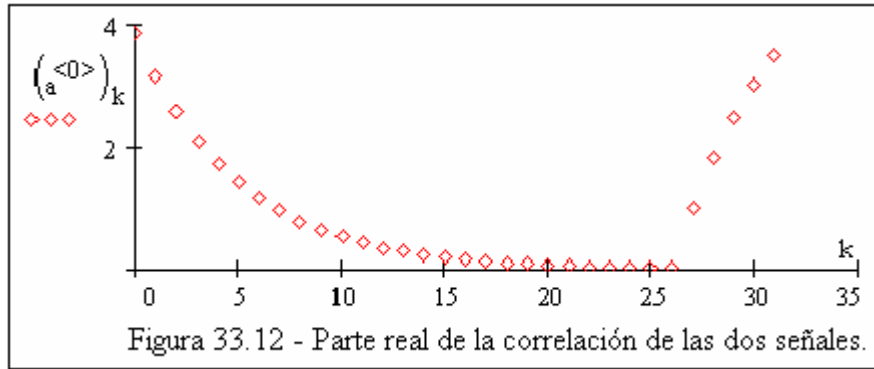
$$y_k := N \cdot \sum_{i=0}^{N-1} b_i \cdot c_{\text{mod}((k-i)+N,N)} \quad \text{Convolución directa}$$



33. 4 - Ejemplo de Correlación

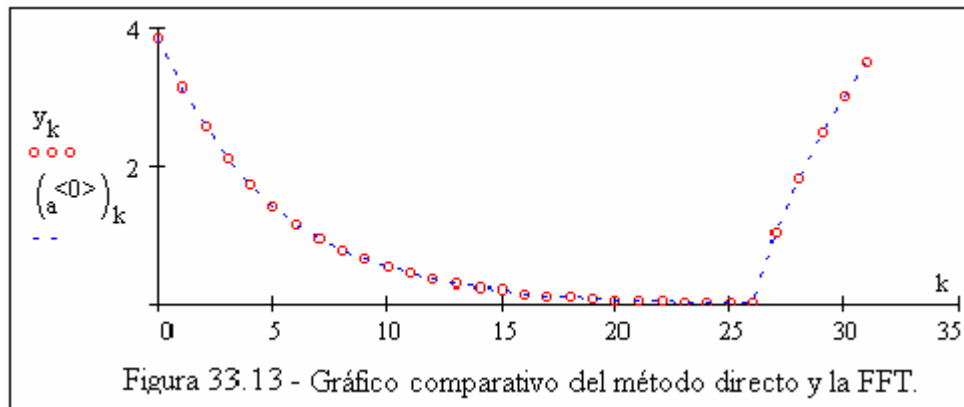
a := READPRN(sal_corr) Lectura de la matriz

En la Figura 33.12 se observa la parte real de la correlación.



Lo que coincide plenamente con el cálculo directo de la Correlación, Figura 33.13

$$y_k := \sum_{i=0}^{N-1} b_i \cdot c_{\text{mod}((k+i)+N,N)} \quad \text{Correlación directa}$$



Si se tienen dos secuencias reales $g(k.T)$ y $h(k.T)$, ambas de longitud N , se puede encontrar las FFT simultáneamente formando una secuencia compleja $a(k.T) = g(k.T) + j h(k.T)$. Usando esta secuencia como entrada al programa BASIC de cálculo, se obtienen las correspondientes FFT, $A(k)$ la cual es $G(k) + j H(k)$. Después, la secuencia $a^*(k.T) = g(k.T) - j h(k.T)$ se forma y su FFT se determina. Esto está dado por:

$$A^*(N-k) = G(k) - j H(-k)$$

donde $G(k)$ y $H(k)$ se extraen de $A(k)$ y $A^*(N-k)$ formando las siguientes ecuaciones:

CONVOLUCION DISCRETA

La convolución discreta está definida por la sumatoria:

$$y(k \cdot T) = \sum_{i=0}^{N-1} x(i \cdot T) \cdot h((k-i) \cdot T) \quad (1)$$

donde tanto $x(k \cdot T)$ como $h(k \cdot T)$ son funciones periódicas con período N ,

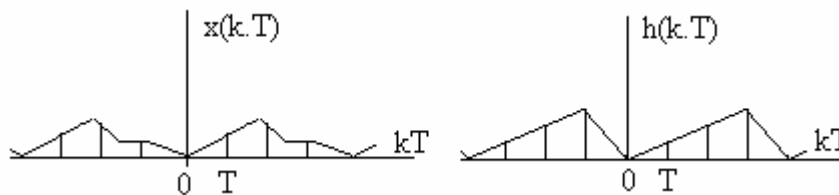
$$x(k \cdot T) = x((k + r \cdot N) \cdot T) \quad r = 0, \pm 1, \pm 2, \dots \quad (2)$$

$$h(k \cdot T) = h((k + r \cdot N) \cdot T) \quad r = 0, \pm 1, \pm 2, \dots$$

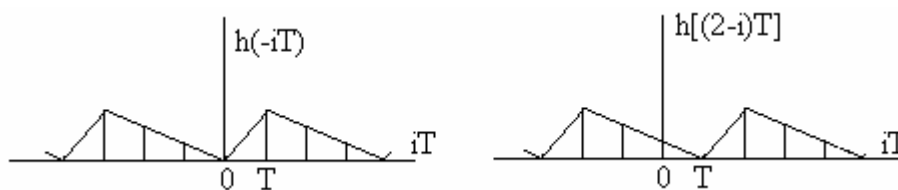
Por conveniencia de notación, la convolución discreta es escrita como:

$$y(k \cdot T) = x(k \cdot T) * h(k \cdot T) \quad (3)$$

Para examinar la ecuación de la convolución discreta, considérese las siguientes ilustraciones:



Ambas funciones, $x(k \cdot T)$ como $h(k \cdot T)$ son periódicas con periodo $N = 4$. A partir de la ecuación (1) las funciones $x(i \cdot T)$ y $h[(k-i) \cdot T]$ son requeridas. La función $h(-i \cdot T)$ es la imagen de $h(i \cdot T)$ alrededor del eje de ordenadas como se ilustra en la primera de las siguientes figuras:

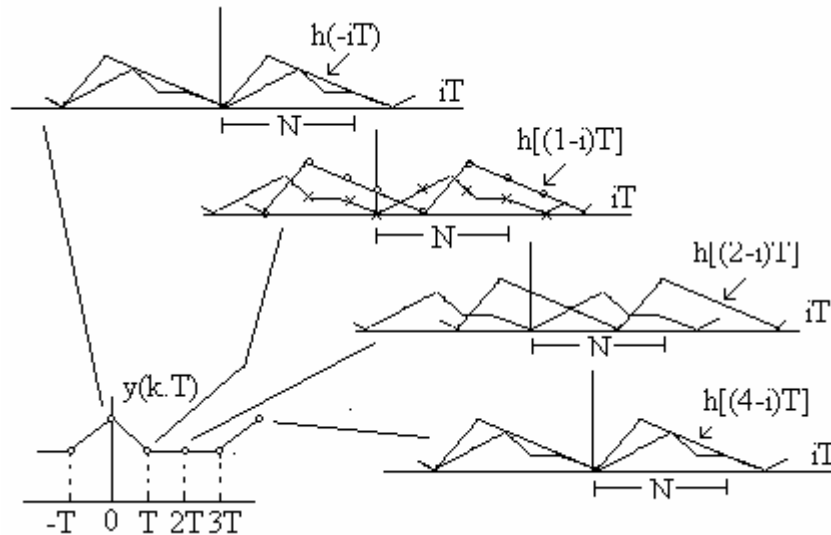


La función $h[(k-i) \cdot T]$ es simplemente la función $h(-i \cdot T)$ desplazada en una cantidad $k \cdot T$.

En la segunda figura se ve $h[(k-i) \cdot T]$ para el desplazamiento T . La ecuación (1) es evaluada para cada desplazamiento $k \cdot T$ realizando las requeridas multiplicaciones y adiciones.

CONVOLUCION DISCRETA GRAFICA

El proceso de convolución discreta se ilustra gráficamente en la siguiente figura:



Los valores muestra de $x(k.T)$ y $h(k.T)$ son denotados con puntos y cruces.

En la primera figura se ilustra el cálculo deseado para $k=0$. El valor de cada punto es multiplicado por el valor de cada cruz lo cual ocurre para el mismo valor de abscisa; estos productos son sumados sobre los $N=4$ valores discretos indicados. En la segunda figura se observa el cómputo de la convolución para $k=1$; la multiplicación y adición es sobre los N puntos indicados. Luego se observa el resultado para $k=2$. Note que para $k=4$, los términos multiplicados y sumados son idénticos a los de la primera figura. Esto era esperado porque tanto $x(kT)$ como $h(kT)$ son periódicas con periodo de cuatro términos. Por lo tanto:

$$y(k.T) = y((k + r.N).T) \quad r = 0, \pm 1, \pm 2, \dots$$

los pasos para el cómputo gráfico de la convolución discreta difieren de la continua sólo en que la integración es reemplazada por la sumatoria. Para la convolución discreta estos pasos son: (1) Plegado (folding), (2) desplazamiento (shifting), (3) multiplicación y (4) sumatoria. Como en la convolución continua, una u otra de las secuencias $x(k.T)$ como $h(k.T)$ pueden ser seleccionadas para desplazamiento. Luego (1) puede ser escrita equivalentemente como:

$$y(k.T) = \sum_{i=0}^{N-1} x((k-i).T) \cdot h(i.T) \quad (5)$$

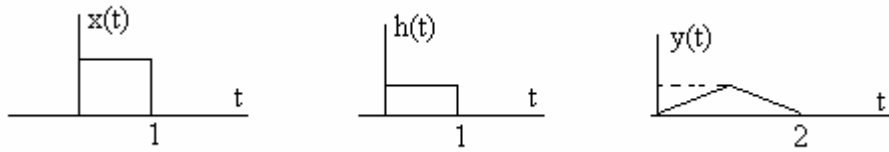
RELACION ENTRE LA CONVOLUCION DISCRETA Y LA CONTINUA

Si sólo se consideran funciones periódicas representadas por funciones impulso igualmente espaciados, la convolución discreta relaciona idénticamente a su equivalente continua.

La aplicación más importante de la convolución discreta no es a funciones muestreadas periódicas, sino más bien para aproximar a las convoluciones continuas de formas de ondas generales. Por esta razón, se explorará en detalle la relación entre la convolución discreta y la continua.

CONVOLUCION DISCRETA DE FORMAS DE ONDA DE DURACION FINITA

Considérense las funciones $x(t)$ y $h(t)$ de la figura.



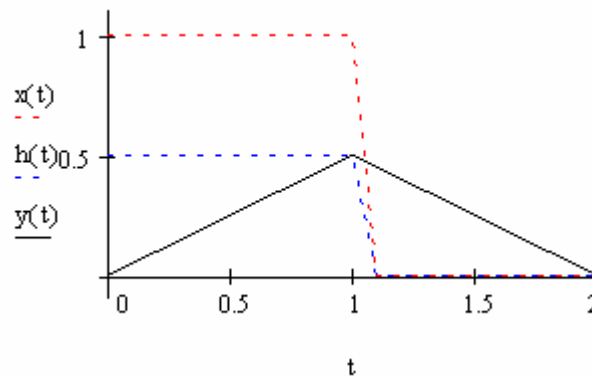
Se desea convolucionar estas dos funciones continua y discretamente, y comparar estos resultados. La convolución continua $y(t)$ de las dos funciones se muestra a la derecha.

$x(t) := \text{if}(t < 0, 0, \text{if}(t < 1, 1, 0))$ primera función a convolucionar

$h(t) := \text{if}(t < 0, 0, \text{if}(t < 1, \frac{1}{2}, 0))$ segunda función a convolucionar

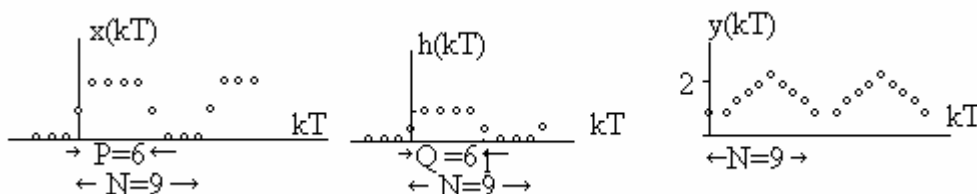
$t := 0, 0.1 \dots 2$ rango (para representación gráfica)

$$y(t) := \int_0^1 x(\tau) \cdot h(t - \tau) d\tau$$



Para evaluar la convolución discreta $y(t)$ de dos funciones, se muestrean tanto $x(t)$ como $h(t)$ con un intervalo de muestreo T y se supone que ambas funciones muestreadas son periódicas con periodo N .

Como se muestra en la siguiente figura, el periodo ha sido elegido como $N=9$ y tanto $x(kT)$ y $h(kT)$ están representadas por $P=Q=6$ muestras; las muestras remanentes definiendo un periodo son puestas a cero.



En la figura de la derecha se ve la convolución discreta $y(k.T)$ para el periodo $N=9$.

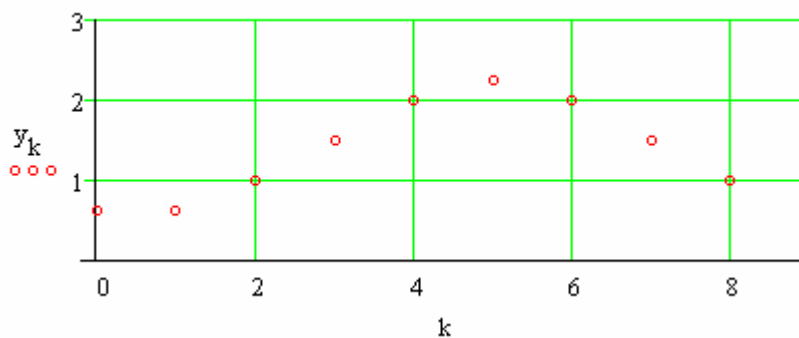
$$x := (.5 \ 1 \ 1 \ 1 \ 1 \ .5 \ 0 \ 0 \ 0) \quad x := x^T$$

$$h := (.25 \ .5 \ .5 \ .5 \ .5 \ .25 \ 0 \ 0 \ 0) \quad h := h^T$$

$$N := \text{length}(x) \quad \text{Periodo de la función}$$

$$k := 0..N - 1$$

$$y_k := \sum_{i=0}^{N-1} x_i \cdot h_{\text{mod}((k-i)+N,N)}$$



Para esta elección de N , la convolución discreta es muy mala aproximación del caso continuo porque la periodicidad constriñe los resultados en un solapamiento de la salida periódica deseada.

Esto es, no se elige el periodo suficientemente grande de modo que la convolución resulte de un periodo que no interfiriera o solaparía la convolución resultante del periodo sucesivo. Es obvio que si se desea la convolución discreta para aproximar a la continua es necesario que el periodo sea elegido de modo que no haya solapamiento.

Elíjase el periodo de acuerdo a la relación:

$$N = P + Q - 1 \quad (6)$$

Esta situación se ilustra del siguiente modo, donde $N=P+Q-1=11$:

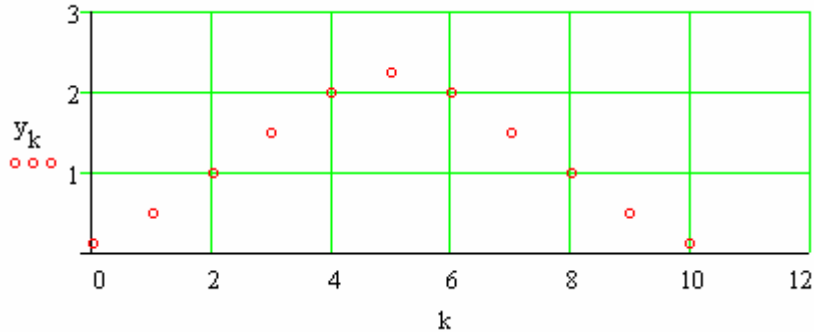
$$x := (.5 \ 1 \ 1 \ 1 \ 1 \ .5 \ 0 \ 0 \ 0 \ 0 \ 0) \quad x := x^T$$

$$h := (.25 \ .5 \ .5 \ .5 \ .5 \ .25 \ 0 \ 0 \ 0 \ 0 \ 0) \quad h := h^T$$

$$N := \text{length}(x) \quad \text{Periodo de la función}$$

$$k := 0..N - 1$$

$$y_k := \sum_{i=0}^{N-1} x_i \cdot h_{\text{mod}((k-i)+N,N)}$$



Nótese que esta elección de N no tiene sobrelape en la convolución resultante. La ecuación (6) está basada en el hecho que la convolución de una función representada por P muestras y una función representada por Q muestras es una función descrita por P+Q-1 muestras.

No hay ventajas en elegir $N > P+Q-1$, como se muestra a continuación:

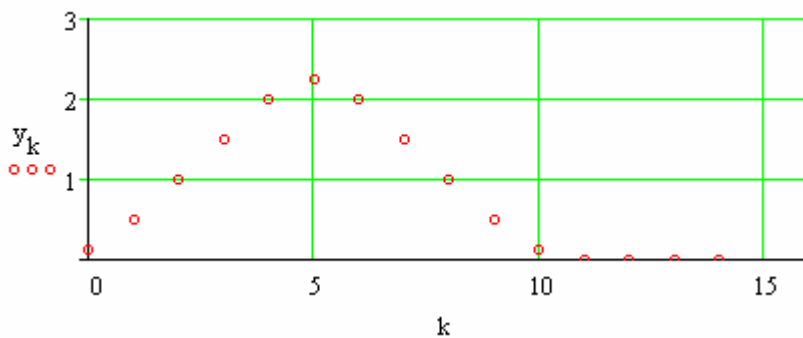
$$x := (.5 \ 1 \ 1 \ 1 \ 1 \ .5 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0) \quad x := x^T$$

$$h := (.25 \ .5 \ .5 \ .5 \ .5 \ .25 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0) \quad h := h^T$$

$$N := \text{length}(x) \quad \text{Periodo de la función}$$

$$k := 0..N-1$$

$$y_k := \sum_{i=0}^{N-1} x_i \cdot h_{\text{mod}((k-i)+N,N)}$$



Para $N=15$ los valores no nulos de la convolución discreta son idénticos a los de la figura anterior. Mientras que N sea elegido de acuerdo a (6) la convolución discreta

resulta ser una función periódica donde cada periodo se aproxima a los resultados de la convolución continua.

En las figuras se ve que la convolución discreta está escalada de distinta forma que la de la continua.

Esta constante de escalamiento es T; modificando la convolución discreta de la ecuación (1) queda:

$$y(k \cdot T) = T \cdot \sum_{i=0}^{N-1} x(i \cdot T) \cdot h((k-i) \cdot T) \quad (7)$$

La relación (7) es simplemente la integral de convolución continua para funciones de tiempo-limitado evaluada por integración rectangular. Así, para funciones de tiempo de longitud finita, la convolución discreta aproxima a la continua dentro del error que introduce la integración rectangular.

Si el intervalo de tiempo T es hecho suficientemente pequeño, entonces el error introducido por la convolución discreta es despreciable.

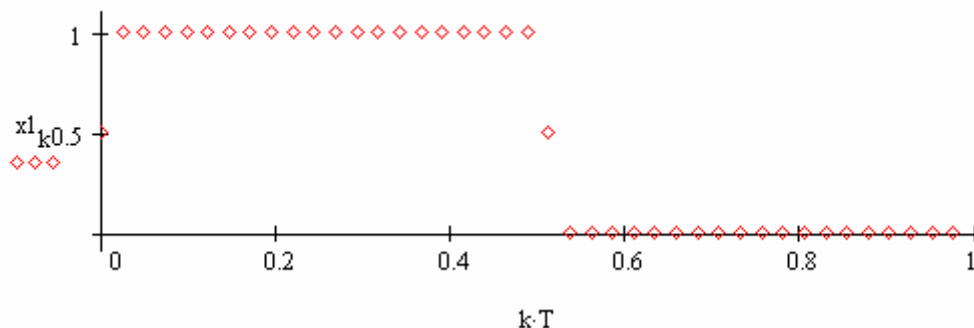
$$T_d := 1 \quad N := 41 \quad T := \frac{T_d}{N} \quad k := 0..N-1$$

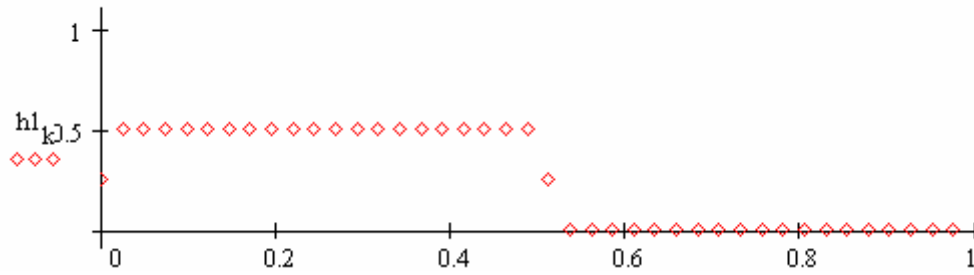
Primera función:

$$x1(t) := \text{if}\left(t < \frac{T_d}{2}, 1, 0\right) \quad x1_k := x1(k \cdot T) \quad x1_0 := \frac{1}{2} \quad x1_{\frac{N+1}{2}} := \frac{1}{2}$$

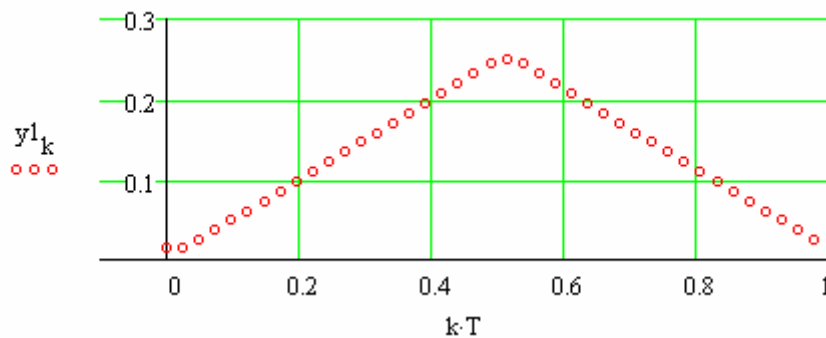
Segunda función:

$$h1(t) := \text{if}\left(t < \frac{T_d}{2}, \frac{1}{2}, 0\right) \quad h1_k := h1(k \cdot T) \quad h1_0 := \frac{1}{4} \quad h1_{\frac{N+1}{2}} := \frac{1}{4}$$





$$y1_k := T \cdot \sum_{i=0}^{N-1} x1_i \cdot h1_{\text{mod}((k-i)+N,N)} \quad \text{Convolución}$$



CORRELACION DISCRETA

La correlación discreta se define como:

$$z(k \cdot T) = \sum_{i=0}^{N-1} x(i \cdot T) \cdot h((k+i) \cdot T)$$

donde $x(kT)$, $h(kT)$, $z(kT)$ son funciones periódicas.

$$x(k \cdot T) = x((k + r \cdot N) \cdot T) \quad r = 0, \pm 1, \pm 2, \dots$$

$$h(k \cdot T) = h((k + r \cdot N) \cdot T) \quad r = 0, \pm 1, \pm 2, \dots$$

$$z(k \cdot T) = z((k + r \cdot N) \cdot T) \quad r = 0, \pm 1, \pm 2, \dots$$

Como en el caso continuo, la correlación discreta difiere de la convolución en que no hay operación de "plegado" (folding). Por lo tanto, las reglas para desplazamiento, multiplicación y sumatoria son realizadas exactamente como para el caso de la convolución discreta.

El Teorema de la correlación discreta está dado por:

$$\sum_{i=0}^{N-1} x(i \cdot T) \cdot h((k+i) \cdot T) \longleftrightarrow \overline{X\left(\frac{n}{N \cdot T}\right)} \cdot H\left(\frac{n}{N \cdot T}\right)$$

A modo de ejemplo correlacionar:

$$\begin{aligned} Td &:= 1 & N &:= 41 & T &:= \frac{Td}{N} & k &:= 0..N-1 \\ x2(t) &:= \text{if}\left(t < \frac{Td}{2}, 1, 0\right) & x2_k &:= x2(k \cdot T) & x2_0 &:= \frac{1}{2} & x2_{\frac{N+1}{2}} &:= \frac{1}{2} \\ h2(t) &:= \text{if}\left(t < \frac{Td}{2}, 1, 0\right) & h2_k &:= h2(k \cdot T) & h2_0 &:= \frac{1}{4} & h2_{\frac{N+1}{2}} &:= \frac{1}{4} \end{aligned}$$

$$z2_k := T \cdot \sum_{i=0}^{N-1} x2_i \cdot h2_{\text{mod}((k+i) + N, N)} \quad \text{Correlación}$$

