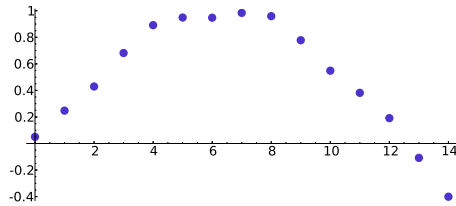


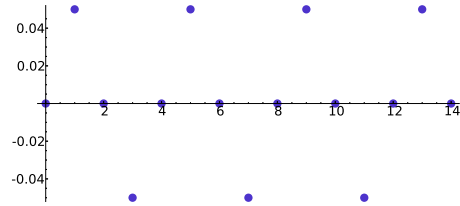
1.3. Análisis armónico discreto

En esta sección “discreto” se refiere más bien a finito, el tema es el análisis armónico en grupos finitos. Aquí nos ocuparemos sólo del caso abeliano, dando un papel protagonista a \mathbb{Z}_N , el grupo aditivo de los enteros módulo N . No obstante, el caso no abeliano tiene una rica teoría y aplicaciones insospechadas [Ter99], [DSV03].

Si consideramos \mathbb{Z}_N como conjunto, son simplemente N elementos y $L^2(\mathbb{Z}_N)$ no es otra cosa que los vectores de \mathbb{C}^N con la norma habitual que viene inducida por el producto escalar $\langle \vec{x}, \vec{y} \rangle = \sum \bar{x}_n y_n$. Intuitivamente es conveniente pensar en la parte real y la parte imaginaria de $f \in L^2(\mathbb{Z}_N)$ como *señales* que en cada tiempo discreto tienen un valor. Por ejemplo, si $f(n) = \text{sen}(n/4) + i^n/20$



$\{\Re f(n)\}_{n=0}^{N-1}$ para $N = 15$



$\{\Im f(n)\}_{n=0}^{N-1}$ para $N = 15$

Esta idea intuitiva no es gratuita. El análisis armónico discreto tiene sus aplicaciones en el tratamiento de señales digitales. La digitalización en la práctica se produce en “tiempo” y se representa en que la variable ahora pertenece al conjunto finito \mathbb{Z}_N , pero también se produce en la propia señal, en la imagen de la función, en el sentido de que el resultado vendrá dado por un número determinado de bytes. Para el modelo matemático es conveniente despreocuparse en primera instancia de esta segunda digitalización y pensar que las imágenes están en \mathbb{R} o en \mathbb{C} .

Un ejemplo más visual son las imágenes digitales, las cuales están compuestas de *pixels*. Cada uno de ellos tiene habitualmente tres valores (canales) correspondientes a los niveles que contiene de rojo, verde y azul (en ciertas situaciones hay un cuarto correspondiente a la transparencia). Éstos son los tres colores básicos en el sentido de que la fisiología de nuestra visión hace que percibamos el resto de estos colores como combinación de ellos. Entonces cada imagen digital de $N \times M$ pixels se puede entender como tres funciones f_R, f_G y f_B de $\mathbb{Z}_N \times \mathbb{Z}_M$ en \mathbb{R} que dan los niveles de estos colores.



$f_R : \mathbb{Z}_{200} \times \mathbb{Z}_{157} \rightarrow \mathbb{R}$



$f_G : \mathbb{Z}_{200} \times \mathbb{Z}_{157} \rightarrow \mathbb{R}$



$f_B : \mathbb{Z}_{200} \times \mathbb{Z}_{157} \rightarrow \mathbb{R}$

1.3.1. Caracteres y transformadas

Una pregunta natural tras la introducción anterior es dónde participa la estructura de grupo. La respuesta es que en la propia construcción de los armónicos. Cuando estudiamos el análisis de Fourier clásico en \mathbb{T} y en \mathbb{R} , en algún momento fue importante que las exponenciales cumplieran $e(x+y) = e(x)e(y)$, es decir, que aplicaran la operación de grupo de \mathbb{T} o de \mathbb{R} , la suma, en la de $\mathbb{C} - \{0\}$, que es la multiplicación. En \mathbb{Z}_N se muestra natural considerar los armónicos $e_j(n) = e(jn/N)$ que tienen una propiedad similar: $e_j(n+m) = e_j(n)e_j(m)$.

Recordemos que podíamos basar el análisis de Fourier básico en las relaciones (1.4) y (1.8). El problema es que caían bajo sospecha por la falta de convergencia. En el mundo finito estos problemas se desvanecen y la relación análoga es sencillamente:

$$(1.28) \quad \delta_D(n) = \sum_{j=0}^{N-1} e(jn/N) \quad \text{donde} \quad \delta_D(n) = \begin{cases} N & \text{si } N|n \\ 0 & \text{en otro caso.} \end{cases}$$

La prueba más pedestre se reduce a sumar la serie geométrica. El segundo miembro es la “delta de Dirac” en \mathbb{Z}_N porque al integrarla con respecto a la medida de contar normalizada (sumar y dividir por N) el resultado es uno o cero dependiendo de si el conjunto de integración contiene a $\bar{0}$ o no.

La *transformada de Fourier discreta* en \mathbb{Z}_N es el análogo de los coeficientes de Fourier y se define como

$$(1.29) \quad \hat{x}_n = \sum_{m=0}^{N-1} x_m e_n(-m) \quad \text{con} \quad e_n(m) = e(nm/N)$$

para cualquier vector $\vec{x} \in \mathbb{C}^N$ que, como hemos visto, es lo mismo que una función $f : \mathbb{Z}_N \rightarrow \mathbb{C}$. Aunque aquí lo evitaremos, es muy habitual designar a la transformada de Fourier discreta mediante sus siglas en inglés DFT.

La relación (1.28) da inmediatamente un desarrollo de Fourier en armónicos (o si lo preferimos, una fórmula de inversión)

$$(1.30) \quad x_n = \frac{1}{N} \sum_{m=0}^{N-1} \hat{x}_m e_m(n).$$

Hay también una identidad de Parseval y buenas propiedades de la convolución que introduciremos en un contexto más general en el que se muestren menos casuales.

Un *carácter* de un grupo abeliano G (no necesariamente finito) es un *homomorfismo* (función que conserva la ley de grupo) $\chi : G \rightarrow S^1$ donde S^1 denota el grupo multiplicativo de números complejos de módulo 1.

Los caracteres forman a su vez un grupo con la multiplicación $(\chi_1\chi_2)(g) = \chi_1(g)\chi_2(g)$ que suele denotarse mediante \widehat{G} . No es difícil comprobar que $\widehat{\mathbb{T}} = \{e(nx)\}_{n=-\infty}^{\infty}$, $\widehat{\mathbb{R}} = \{e(\xi x)\}_{\xi \in \mathbb{R}}$ y $\widehat{\mathbb{Z}}_N = \{e(jn/N)\}_{j=0}^{N-1}$. Por tanto los caracteres generalizan todos los armónicos que hemos visto hasta ahora. Es posible hacer una teoría general en grupos abelianos localmente compactos [Dei05], [Kat76] en la que las transformadas de Fourier continuas y discretas y los coeficientes de Fourier son diferentes aspectos de un mismo fenómeno: las funciones de $L^2(G)$ se expresan como integrales sobre \widehat{G} con respecto a cierta medida. Incluso si nos limitamos al ámbito de la aritmética, esto no es una generalización innecesaria. El análisis armónico en ciertos grupos (también con estructura de anillo) llamados *adeles* [Rob74] asociados a cuerpos de números, permite deducir análogos muy profundos de la Proposición 1.2.10 a partir de una fórmula de sumación de Poisson que incorpora propiedades acerca de la factorización en ideales primos. Genéricamente se llama a este círculo de ideas *tesis de Tate* [RV99] pues fueron introducidas por J.T. Tate en su tesis doctoral.

Por otro lado, en el caso de grupos no abelianos, la definición anterior de caracteres es demasiado restrictiva y hay que considerar en su lugar *representaciones*, una especie de caracteres matriciales, mientras que se llama caracteres a sus trazas [Ter99], [Art00], [Kna96a], [Kna96b].

Volviendo a nuestro caso, dado G abeliano y finito, por el teorema de clasificación debe existir un isomorfismo $\phi : G \longrightarrow \mathbb{Z}_{N_1} \oplus \mathbb{Z}_{N_2} \oplus \cdots \oplus \mathbb{Z}_{N_r}$ y es fácil ver que el grupo de caracteres es

$$(1.31) \quad \widehat{G} = \left\{ \chi : \chi(n) = \prod_{k=1}^r e(j_k \phi(n)/N_k) \quad \text{con } 0 \leq j_k < N_k \right\}.$$

En otras palabras, salvo componer con el isomorfismo, son productos de caracteres de los \mathbb{Z}_{N_k} . Los grupos G y \widehat{G} son duales en el sentido de que los caracteres de \widehat{G} son $\{\chi \mapsto \chi(g)\}_{g \in G}$. Este hecho se extiende al caso infinito y en esta generalidad se llama *teorema de dualidad de Pontryagin*.

El siguiente lema es heredero de (1.28) y admite una demostración directa similar. Vamos, sin embargo, a deducirlo básicamente de las propiedades de grupo.

Lema 1.3.1 (relaciones de ortogonalidad). *Sea G un grupo abeliano finito y $\chi_1, \chi_2 \in \widehat{G}$, entonces*

$$\sum_{g \in G} \overline{\chi_1(g)} \chi_2(g) = \begin{cases} |G| & \text{si } \chi_1 = \chi_2 \\ 0 & \text{si } \chi_1 \neq \chi_2 \end{cases} \quad y \quad \sum_{\chi \in \widehat{G}} \chi(g) = \begin{cases} |G| & \text{si } g = e \\ 0 & \text{si } g \neq e \end{cases}$$

donde $e \in G$ es elemento neutro.

Demostración. Se tiene $\overline{\chi_1(g)}\chi_2(g) = \chi(g)$ con $\chi = \chi_1^{-1}\chi_2$ porque $\chi_1(g) \in S^1$. Si $\chi_1 = \chi_2$ el resultado es trivial y si $\chi_1 \neq \chi_2$ existe $h \in G$ con $\chi(h) \neq 1$, en este caso

$$\sum_{g \in G} \overline{\chi_1(g)}\chi_2(g) = \sum_{g \in G} \chi(g) = \sum_{g \in G} \chi(gh) = \chi(h) \sum_{g \in G} \overline{\chi_1(g)}\chi_2(g)$$

y entonces la suma tiene que ser cero. La relación recién probada cambiando G por \widehat{G} y tomando como χ_1 el elemento neutro, da la segunda fórmula del enunciado. \square

En este contexto se define la *transformada de Fourier discreta* para cada $\chi \in \widehat{G}$ como

$$(1.32) \quad \widehat{f}(\chi) = \sum_{g \in G} f(g)\overline{\chi(g)}.$$

Nótese que para $G = \mathbb{Z}_N$ esto es lo mismo que la definición (1.29) salvo el abuso de notación de representar el carácter por su índice en el argumento de la transformada.

El producto escalar en $L^2(G)$ y la *convolución* vienen definidos respectivamente por

$$\langle f_1, f_2 \rangle = \sum_{g \in G} \overline{f_1(g)}f_2(g) \quad \text{y} \quad (f_1 * f_2)(g) = \sum_{h \in G} f_1(h)f_2(gh^{-1}).$$

Con estas definiciones y el Lema 1.3.1 se prueban en un instante las propiedades cuyo análogo conocíamos en \mathbb{T} y \mathbb{R} . También hay una fórmula de sumación de Poisson [Ter99, §12], aunque no la veremos aquí.

Teorema 1.3.2. *Para G abeliano y finito y $f, f_1, f_2 : G \rightarrow \mathbb{C}$, se verifican la fórmula de inversión*

$$f(g) = \frac{1}{|G|} \sum_{\chi \in \widehat{G}} \widehat{f}(\chi)\chi(g),$$

la identidad de Parseval

$$\langle f, f \rangle = \frac{1}{|G|} \langle \widehat{f}, \widehat{f} \rangle, \quad \text{en general} \quad \langle f_1, f_2 \rangle = \frac{1}{|G|} \langle \widehat{f}_1, \widehat{f}_2 \rangle,$$

y además la propiedad de la convolución

$$\widehat{f_1 * f_2}(\chi) = \widehat{f}_1(\chi)\widehat{f}_2(\chi).$$

Por ejemplo, los elementos invertibles (con la multiplicación) módulo 10, $\mathcal{U}(\mathbb{Z}_{10}) = \{\overline{1}, \overline{3}, \overline{7}, \overline{9}\}$, forman un grupo isomorfo a \mathbb{Z}_4 con el isomorfismo $\phi : \mathcal{U}(\mathbb{Z}_{10}) \rightarrow \mathbb{Z}_4$, dado por $\phi(\overline{1}) = \overline{0}$, $\phi(\overline{3}) = \overline{1}$, $\phi(\overline{7}) = \overline{3}$, $\phi(\overline{9}) = \overline{2}$, que se dice que es un *logaritmo discreto* porque pasa multiplicaciones en sumas. Según el Teorema 1.3.2, los armónicos vienen dados por (1.31) y son

$$\{\chi_0, \chi_1, \chi_2, \chi_3\} \quad \text{con} \quad \chi_j(n) = e(j\phi(n)/4).$$

Todo esto parece demasiado fácil como para ser interesante, pero veremos a continuación que sirve de manera fundamental para resolver un problema difícil en teoría de números.

Empecemos por otro más sencillo. Todos sabemos que hay infinitos primos. La prueba de Euler parte de la importantísima *fórmula producto de Euler*

$$(1.33) \quad \boxed{\zeta(s) = \prod_{p \text{ primo}} (1 - p^{-s})^{-1}} \quad \text{para } \Re(s) > 1,$$

que se sigue de $(1 - p^{-s})^{-1} = 1 + p^{-s} + (p^2)^{-s} + (p^3)^{-s} + \dots$ aplicando el teorema fundamental de la aritmética. Si $s \rightarrow 1^+$, $\zeta(s) \rightarrow +\infty$ y por tanto el producto no puede ser finito.

El problema difícil es probar que hay infinitos primos que acaban por ejemplo en 1. Uno estaría tentado de escribir un análogo de (1.33) con $\sum (10n+1)^{-s}$ en lugar de $\zeta(s)$ pero tal cosa no funciona porque aunque un número entero acabe en 1, sus factores primos pueden no hacerlo, por ejemplo $221 = 13 \cdot 17$. Extendamos nuestros armónicos a \mathbb{Z} definiendo $\chi_j(n)$ como $\chi_j(\bar{n})$ si $\bar{n} \in \mathcal{U}(\mathbb{Z}_{10})$, esto es si n y 10 son coprimos, y $\chi_j(n) = 0$ en otro caso. En general, a este tipo de extensiones a \mathbb{Z} de los armónicos de $\mathcal{U}(\mathbb{Z}_N)$, se les llama *caracteres de Dirichlet*. Con ello los armónicos se vuelven multiplicativos en \mathbb{Z} . Dada una factorización en primos, $n = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_k^{\alpha_k}$ se tiene $\chi_j(n) = \chi_j^{\alpha_1}(p_1) \chi_j^{\alpha_2}(p_2) \dots \chi_j^{\alpha_k}(p_k)$ y la prueba de (1.33) se repite exactamente para obtener

$$L(s, \chi_j) = \prod_{p \text{ primo}} (1 - \chi_j(p)p^{-s})^{-1} \quad \text{para } \Re(s) > 1, \quad \text{donde } L(s, \chi_j) = \sum_{n=1}^{\infty} \frac{\chi_j(n)}{n^s}.$$

Se dice que $L(s, \chi_j)$ es una *función L de Dirichlet*. Utilizando que los armónicos representan cualquier función, seleccionaremos los primos que acaban en 1 y todo el resultado volverá a reducirse a qué ocurre con $L(s, \chi_j)$ cuando $s \rightarrow 1^+$, lo cual en este caso es algo puramente computacional. Los detalles están en la demostración siguiente.

Proposición 1.3.3. *Hay infinitos primos que acaban en 1. En general hay infinitos primos que acaban en cualquier cifra impar distinta de 5.*

Demostración. Tomando logaritmos y usando el desarrollo de Taylor de $-\log(1-x)$, se tiene

$$\log L(s, \chi_j) = \sum_{p \text{ primo}} \sum_{k=1}^{\infty} \frac{\chi_j^k(p)}{k p^s} = \sum_{p \text{ primo}} \frac{\chi_j(p)}{p^s} + O(1) \quad \text{para } s > 1,$$

donde simplemente se ha empleado que $\sum p^{-k} < \sum_{n=2}^{\infty} n^{-k} \ll 2^{-k}$. Ahora analizamos la “delta de Dirac” en $\bar{1}$ con los armónicos χ_j (Teorema 1.3.2 o Lema 1.3.1)

$$\chi_0(n) + \chi_1(n) + \chi_2(n) + \chi_3(n) = \begin{cases} 4 & \text{si } n \equiv 1 \pmod{10} \\ 0 & \text{en otro caso.} \end{cases}$$

Así pues

$$(1.34) \quad \log(L(s, \chi_0)L(s, \chi_1)L(s, \chi_2)L(s, \chi_3)) = 4 \sum_{p \equiv 1 \pmod{10}} \frac{1}{p^s} + O(1) \quad \text{para } s > 1.$$

Como $\chi_0(n) = 1$ si n y 10 son coprimos y 0 en el resto, $L(s, \chi_0) > \sum (10n + 1)^{-s} \rightarrow \infty$ cuando $s \rightarrow 1^+$. Por otro lado, tomando 10 términos en las sumas que definen $L(1, \chi_1)$ y $L(1, \chi_2) = \overline{L(1, \chi_3)}$, se tiene

$$L(1, \chi_1) = 0,63 + \epsilon_1 \quad \text{y} \quad \Re(L(1, \chi_2)) = 0,88 + \epsilon_2$$

con $|\epsilon_1|, |\epsilon_2| < 1/10$. Esta cota proviene de que las series son alternadas y el error es menor que el primer término despreciado [Spi84, p.600]. Por tanto $L(1, \chi_1)L(1, \chi_2)L(1, \chi_3)$ es un número real positivo y de (1.34) con $s \rightarrow 1^+$ se deduce que hay infinitos primos que acaban en 1. La prueba es similar para otra cifra admisible k , analizando la “delta de Dirac” en \overline{k} . \square

La generalización natural de este resultado es conocida:

Teorema 1.3.4 (Teorema de Dirichlet). *Si $q \in \mathbb{Z}^+$ y $a \in \mathbb{Z}$ son coprimos, entonces la progresión aritmética $\{qn + a\}_{n=1}^{\infty}$ contiene infinitos primos.*

La demostración anterior permite reducir el resultado a un trabajo computacional para cada q concreto, pero el enunciado general exige que ninguna de las funciones L de Dirichlet tenga un cero en $s = 1$. La prueba de ello es artificiosa (aunque no larga) y puede consultarse por ejemplo en [SS03] o [Dav80].

1.3.2. Implementación del análisis de Fourier discreto

En el mundo real hay transformadas de Fourier por todas partes, sin embargo el mundo virtual es digital y debe reemplazarlas por transformadas de Fourier discretas. Un primer problema es cómo pasar del caso continuo al discreto, en cierto modo (aunque no exactamente), de analógico a digital. Otro problema de índole más técnica es cómo llevar a cabo los cálculos con transformadas discretas de manera eficiente.

En la práctica, nuestros sentidos o nuestros instrumentos de medida sólo operan en un rango de frecuencias. Sería por ejemplo absurdo comprar un altavoz para nuestra cadena de música que reprodujera frecuencias por encima de los 20000 hercios (20000 oscilaciones por segundo), ya que tales sonidos son inaudibles. Por otro lado, es mejor que el mando a distancia de nuestra televisión tenga una frecuencia de funcionamiento por debajo de la de la luz visible para que no se vea afectado cuando encendamos una lámpara.

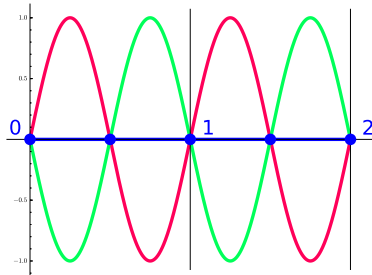
De esta forma, en la práctica es natural considerar que las funciones que representan señales tienen un contenido en frecuencias limitado. Para el caso de una onda periódica,

esto es lo mismo que decir que la serie de Fourier es una suma finita, un polinomio trigonométrico, mientras que para una señal genérica, la hipótesis natural es que sea una *función de banda limitada*, lo que significa que satisface para cierto B

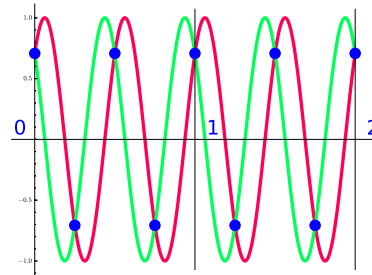
$$(1.35) \quad \widehat{f}(\xi) = 0 \quad \text{cuando } |\xi| \geq B.$$

Recordemos que, a través de (1.7), el valor de $\widehat{f}(\xi)$ indica de alguna forma el “coeficiente” con el que aparece el armónico $e(\xi x)$ de frecuencia ξ al analizar f . Si a uno no le dan mucho miedo las deltas de Dirac, de hecho esta definición recoge también el caso periódico. Dicho sea de paso, el *teorema de Paley-Wiener* describe en cierto modo las funciones de banda limitada como funciones de variable compleja.

Para transformar una señal en un conjunto discreto de datos, digamos que tomamos una muestra (un valor de la función) cada T segundos. En esta situación se dice que la *frecuencia de muestreo* es $\nu_s = 1/T$. Matemáticamente lo que se tiene es una función f y una sucesión $\{f(nT)\}_{n=-\infty}^{\infty}$. Está claro que este conjunto discreto no permite caracterizar la función en general:



$T = 1/2, \nu_s = 2$ no distingue $0, \text{sen}(2\pi x)$ y $-\text{sen}(2\pi x)$



$T = 1/4, \nu_s = 4$ no distingue $\text{sen}(4\pi x + \frac{\pi}{4})$ y $\cos(4\pi x + \frac{\pi}{4})$

Pensemos en una onda 1-periódica f con frecuencias limitadas al rango $(-B, B)$ y supongamos para simplificar que $B \in \mathbb{Z}^+$. Esta f debe ser un polinomio trigonométrico con a lo más $2B - 1$ coeficientes no nulos. Si $\nu_s \geq 2B$, entonces habrá $2B$ muestras en cada periodo y eso es suficiente para recuperar todos los coeficientes resolviendo un sistema lineal. Lo que dice el *teorema de muestreo de Shannon* es que tal cosa también se cumple para señales no periódicas, de hecho hay una forma fácil de recuperar la señal a partir de la muestra. Por simplicidad, omitiremos las consideraciones con respecto a la regularidad de f en el siguiente resultado, aunque está claro que con $f \in L^1$ continua y $\widehat{f} \in C^2$, se asegura la convergencia absoluta de las series que aparecen en la prueba.

Teorema 1.3.5 (Teorema de muestreo de Shannon). *Sea f satisfaciendo (1.35) con $2B \leq \nu_s$. Entonces*

$$f(x) = \sum_{n=-\infty}^{\infty} f(n/\nu_s) \text{sinc}(\nu_s x - n)$$

donde $\text{sinc}(x) = (\pi x)^{-1} \text{sen}(\pi x)$ si $x \neq 0$ y $\text{sinc}(0) = 1$.

Es decir, dada una frecuencia de muestreo ν_s , la máxima frecuencia que debe contener una función de banda limitada para que la podamos reconstruir es $\nu_s/2$, esto es lo que se llama *frecuencia de Nyquist*. Véase en [Wal96, §5.11] una versión aproximada bajo hipótesis más débiles que (1.35).

Demostración. Por la hipótesis $2B \leq \nu_s$, es posible definir una función g como la extensión ν_s -periódica de \hat{f} de manera que $g(\xi) = \hat{f}(\xi)$ para $|\xi| < B$. Su desarrollo de Fourier será, según (1.5)

$$g(\xi) = \nu_s^{-1} \sum_{n=-\infty}^{\infty} e(n\xi/\nu_s) \int_{-\nu_s/2}^{\nu_s/2} \hat{f}(x) e(-nx/\nu_s) dx.$$

La integral se extiende a todo \mathbb{R} por la condición (1.35), y por la fórmula de inversión y cambiando n por $-n$ se sigue

$$g(\xi) = \nu_s^{-1} \sum_{n=-\infty}^{\infty} f(n/\nu_s) e(-n\xi/\nu_s).$$

Ahora bien, $f(x) = \int \hat{f}(\xi) e(x\xi) d\xi = \int_{-\nu_s/2}^{\nu_s/2} g(\xi) e(x\xi) d\xi$, también se sigue de la fórmula de inversión, y sustituyendo g en esta última integral se concluye la prueba con un sencillo cálculo. \square

Evidentemente en la práctica la muestra no es una sucesión infinita sino que confiamos en que la señal tenga un principio y un fin (localización en tiempo) o que decaiga hasta límites despreciables. Una manifestación del *principio de incertidumbre* hace que lo primero sea teóricamente imposible para funciones de banda limitada [DM72, §2.9]. Por otro lado, hay un antiguo trabajo poco conocido de A.E. Ingham en el que se dan condiciones necesarias y suficientes para la existencia de una función de banda limitada con decaimiento prefijado [Ing].

Está claro que desde el punto de vista práctico, no podemos guardar todo el desarrollo de Fourier de una función 1-periódica arbitraria en una máquina. En principio sí podríamos hacerlo con los desarrollos de Fourier en \mathbb{Z}_N porque son finitos pero en ambos casos, con vistas a la compresión, sólo deseáramos guardar la información significativa. La mejor situación en \mathbb{T} es cuando tenemos coeficientes de Fourier que decaen rápidamente y basta con almacenar los primeros de ellos. Sin embargo, si tomamos un fragmento de una señal, típicamente cuando la hacemos periódica el resultado es discontinuo en \mathbb{T} y la baja regularidad impide la convergencia rápida de la serie de Fourier. En el caso discreto ocurre algo similar. Sumando por partes (Lema 1.2.1) en (1.29), se tiene para $n \neq 0$

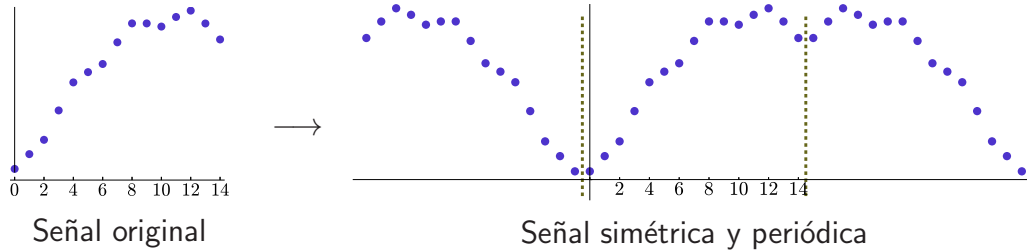
$$\hat{x}_n = \sum_{m=0}^{N-2} (x_m - x_{m+1}) s_m \quad \text{con} \quad s_m = \sum_{k=0}^m e_n(-k).$$

Los s_m oscilan pero se puede comprobar que su promedio no es cero, de hecho se cumple $N^{-1} \sum_{m=0}^{N-1} s_m = 1/2$, por consiguiente esperamos una contribución típica de $\frac{1}{2} \sum_{m=0}^{N-2} (x_m - x_{m+1}) = (x_0 - x_{N-1})/2$ en \hat{x}_n .

Por ejemplo, para la señal $x_n = n - 4,5$ en \mathbb{Z}_{10} , se tiene $\hat{x}_0 = 0$,

$\hat{x}_1 = -5 + 15,3884i$, $\hat{x}_2 = -5 + 6,8819i$, $\hat{x}_3 = -5 + 3,6327i$, $\hat{x}_4 = -5 + 1,6245i$, $\hat{x}_5 = -5$ y $\hat{x}_n = \overline{\hat{x}_{10-n}}$ para $5 < n < 10$.

El problema se soluciona en la práctica sustituyendo la señal por otra de periodo doble obtenida por simetría a través de $-1/2$ (o equivalentemente de $N - 1/2$).



La simetría por $n = -1/2$ sugiere que estamos trasladando los armónicos, y como duplicamos el periodo, deberían pasar a ser $\tilde{e}_j((n+1/2)/2N)$. Por otra parte, la simetría, hace que las partes imaginarias se cancelen, entonces desde el punto de vista teórico todo funciona como si simplemente hubiéramos cambiado los armónicos exponenciales de (1.29) por unos nuevos armónicos $\cos(\pi(m+1/2)n/N)$. Por las razones antes indicadas, la nueva transformada sustituye en muchas aplicaciones a la transformada de Fourier discreta. El análogo de la definición (1.29) y de (1.30) está incluido en el siguiente resultado:

Proposición 1.3.6. Dado $\vec{x} = (x_n)_{n=0}^{N-1} \in \mathbb{C}^N$ se define su transformada coseno discreta como

$$\hat{x}_n^c = \sum_{m=0}^{N-1} x_m \cos\left(\frac{\pi n}{N}\left(m + \frac{1}{2}\right)\right).$$

Entonces se tiene la fórmula de inversión

$$x_m = \frac{\hat{x}_0^c}{N} + \frac{2}{N} \sum_{n=1}^{N-1} \hat{x}_n^c \cos\left(\frac{\pi n}{N}\left(m + \frac{1}{2}\right)\right).$$

Demostración. Como las fórmulas son lineales, basta probar la fórmula de inversión en el caso en que $x_{m_0} = 1$ y $x_m = 0$ para $m \neq m_0$. Por sustitución directa y empleando $2 \cos \alpha \cos \beta = \cos(\alpha + \beta) + \cos(\alpha - \beta)$, la fórmula se reduce en este caso, con la notación de (1.28), a

$$\delta_D(m - m_0) = 1 + \sum_{n=1}^{N-1} \left(\cos\left(\frac{\pi n}{N}(m_0 + m + 1)\right) + \cos\left(\frac{\pi n}{N}(m - m_0)\right) \right) \quad \text{para } 0 \leq m < N.$$

La suma de cosenos es par en n y se anula para $n = N$, por tanto se reescribe como

$$\delta_D(m - m_0) = \frac{1}{2} \sum_{n=-N}^{N-1} \left(\cos\left(\frac{\pi n}{N}(m_0 + m + 1)\right) + \cos\left(\frac{\pi n}{N}(m - m_0)\right) \right) \quad \text{para } 0 \leq m < N,$$

que equivale a

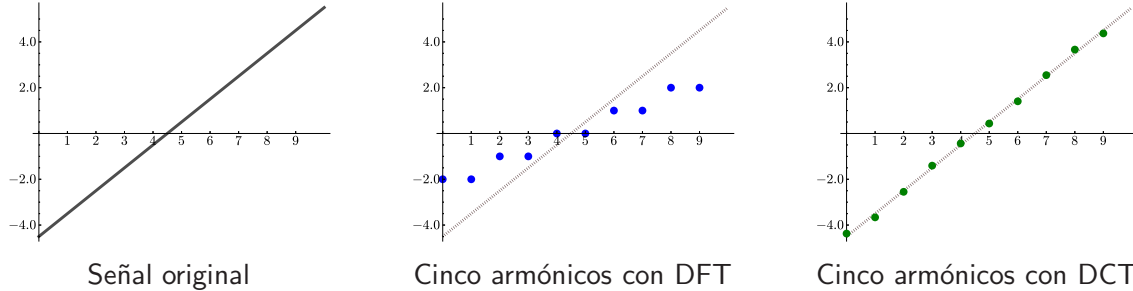
$$\delta_D(m - m_0) = \frac{1}{2} \Re \sum_{n=0}^{2N-1} \left(e\left(\frac{n(m_0 + m + 1)}{2N}\right) + e\left(\frac{n(m - m_0)}{2N}\right) \right) \quad \text{para } 0 \leq m < N,$$

y esto se sigue de (1.28) con $2N$ en lugar de N , notando que $0 < m_0 + m + 1 < 2N$. \square

Para ver que la idea funciona, tomemos de nuevo la señal $x_n = n - 4,5$ en \mathbb{Z}_{10} . Los valores de la transformada coseno discreta son $\hat{x}_n = 0$ si n es par y

$$\hat{x}_1^c = -20,1801, \quad \hat{x}_3^c = -2,1615, \quad \hat{x}_5^c = -0,7071, \quad \hat{x}_7^c = -0,2859, \quad \hat{x}_9^c = -0,0801.$$

Por si esto no fuera suficientemente ilustrativo, veamos lo que ocurre al reconstruir la señal original empleando sólo la primera mitad de los armónicos:



No es ya sólo que la transformada coseno discreta (DCT) da una aproximación casi perfecta con la mitad de la información, es que la transformada de Fourier discreta apenas aproxima en la misma situación.

Uno de los algoritmos que sustentan la importancia del análisis de Fourier abstracto en la práctica, y uno de los más importantes en todo el análisis numérico, es la *transformada rápida de Fourier*, más conocida por sus siglas en inglés FFT. Es un algoritmo de tipo *divide y vencerás* (lo que en Matemáticas se llamaría inducción, se reduce recursivamente un problema a otros con parámetros menores) que sirve para calcular transformadas de Fourier discretas en \mathbb{Z}_N mucho más rápido de lo que cabría esperar. Por supuesto, también sirve para calcular transformadas coseno discretas pues éstas se expresan fácilmente en términos de aquellas.

Nuestro objetivo es calcular las sumas

$$y_n = \sum_{m=0}^{N-1} x_m e\left(\frac{mn}{N}\right) \quad \text{para } 0 \leq n < N$$

suponiendo que, como casi siempre en la práctica, N es una potencia de dos.

Si están almacenadas en memoria las raíces N -ésimas de la unidad y no cuesta trabajo calcular o recuperar de la memoria $e(mn/N)$, cada y_n requiere N multiplicaciones, y si queremos hallar todos los y_n , se requieren N^2 . Si uno tiene que hacer estas operaciones “a mano” no es difícil percatarse de que se reducen los cálculos usando la periodicidad de las funciones trigonométricas. No es por ello de extrañar que aunque se suele dar a [CT65] como referencia para la creación del método, algunos autores han señalado que se remonta incluso a Gauss.

La idea clave de la FFT se sintetiza en un línea:

$$(1.36) \quad y_n = \sum_{m=0}^{N'-1} x_{2m} e\left(\frac{mn}{N'}\right) + e\left(\frac{n}{N}\right) \sum_{m=0}^{N'-1} x_{2m+1} e\left(\frac{mn}{N'}\right) \quad \text{con } N' = \frac{N}{2}.$$

Esta igualdad es una obviedad. Lo que hay que comprobar es que con ella es posible reducir el número de operaciones. Llamemos y_n^{par} al primer sumatorio e y_n^{impar} al segundo.

Evidentemente, como sólo involucra la mitad de los términos, cada y_n^{par} requiere N' multiplicaciones, pero además sólo hace falta calcular y_n^{par} para $n = 0, 1, 2, \dots, N' - 1$ porque $y_n^{\text{par}} = y_{n+N'}^{\text{par}}$. En definitiva, tomando la mitad de los términos el número de multiplicaciones por fuerza bruta pasa de N^2 a $(N')^2 = N^2/4$. ¡En vez de dividirse por dos se divide por cuatro! De la misma manera, los y_n^{impar} requieren en total $N^2/4$ multiplicaciones mientras que multiplicar por $e(n/N)$ requiere otras N operaciones. (De hecho con $N/2$ es suficiente por la antisimetría).

En definitiva, al escribir (1.36) hemos transformado las N^2 operaciones que aparecen en la definición de los y_n en $N + 2(N')^2 = N + N^2/2$. Un factor dos no parece una gran cosa pero la fuerza del método está en iterar acumulando muchos de estos doses. Aplicando (1.36) a y_n^{par} e y_n^{impar} , que tienen longitud N' , para descomponerlas en sumas de longitud $N'' = N'/2$, la evolución del número de operaciones tras los dos pasos sería

$$N^2 \xrightarrow{\text{paso 1}} N + 2(N')^2 \xrightarrow{\text{paso 2}} N + 2(N' + 2(N'/2)^2) = 2N + 4(N'')^2.$$

Repetiendo el proceso $k = \log_2 N$ veces se llegará a $kN + 2^k$, con lo que el número total de multiplicaciones es comparable a kN , y esto es muchísimo menor que N^2 cuando N es grande. Por ejemplo $N = 2^{25}$ implica $N^2 = 1,12 \cdot 10^{15}$ y $kN = 8,38 \cdot 10^8$, un valor más de un millón de veces menor. En ordenador actuales (rápidos) el primer número de operaciones podría llevar dos semanas y la FFT reduciría el tiempo a un segundo.

Es fácil entender que tal diferencia es crucial en ingeniería y que un bien de tan primera necesidad y a la vez tan sencillo, no debe pertenecer a nadie. En [Ger99] hay

un extraño juicio acerca de ello: *When Cooley (then at IBM) first presented the FFT, IBM concluded that it was so significant it should be put in the public domain to prevent anyone from trying to patent it, and so it was published openly. Ironically, its very success has made this kind of behavior much less common now.*

En esta exposición de la FFT hemos supuesto que $e(mn/N)$ están dados. Un análisis similar prueba que disponiendo sólo de $e(1/N)$ el número de operaciones sigue siendo comparable al anterior [SS03, §7.13], [Str86, §5.5].

La transformada rápida de Fourier es tan rápida que a veces es conveniente calcular la convolución $a * b$ con la igualdad $a * b = (\widetilde{ab})$ en vez de directamente. Esto tiene una sorprendente consecuencia a la hora de multiplicar números muy grandes. Dados dos enteros N y M escritos en base B , al multiplicarlos aparece una convolución:

$$N = \sum_{r=0}^R a_r B^r, \quad M = \sum_{r=0}^R b_r B^r \quad \text{implica} \quad NM = \sum_{r=0}^{2R} \sum_{s=0}^r a_s b_{r-s} B^r = \sum_{r=0}^{2R} (a * b)_r B^r.$$

Con el método descrito, $a * b$ se calcula bastante rápido. En la práctica B está formado por muchos *bytes*, es como sino nosotros multiplicásemos en base 10 a mano tomando por ejemplo grupos de tres cifras y teniendo cuidado con las llevadas. Elaborar esta idea e incorporar algunas sutilezas da lugar al *algoritmo de Schönhage-Strassen* [SS71] que permite multiplicar números de n cifras en $O(n \log n \log \log n)$ pasos. La constante O es suficientemente grande como para que esto empiece a ser eficiente con respecto a otros algoritmos conocidos sólo para números con al menos decenas de miles de cifras decimales.

1.3.3. Análisis de Fourier discreto en la vida cotidiana

Hoy en día es difícil exagerar la importancia práctica del análisis de Fourier discreto y esto no es una novedad. En [Str86], escrito hace más de 25 años, leemos:

We come now to reality. The truth is that the digital computer has totally defeated the analog computer. The input is a sequence of numbers and not a continuous function. The output is another sequence of numbers, whether it comes from a digital filter or a finite element stress analysis or an image processor. The question is whether the special ideas of Fourier analysis still have a part to play, and the answer is absolutely *yes*. The discrete Fourier transform takes n numbers f_0, \dots, f_{n-1} into n coefficients c_0, \dots, c_{n-1} , and the patterns that are hidden in f stand out more distinctly in c .

Aquí vamos a ver algunas de las aplicaciones sin más propósito que ilustrar que el análisis de Fourier discreto tiene una fuerte motivación práctica.

Comencemos con unas consideraciones sencillas acerca de *filtros lineales*. Matemáticamente no son más que convoluciones y su importancia radica en la propiedad básica que se cumple tanto en \mathbb{R} como en grupos abelianos finitos (véase (1.12) y el Teorema 1.3.2)

$$F = f * g \quad \Rightarrow \quad \widehat{F} = \widehat{f}\widehat{g}.$$

La utilidad práctica es que en teoría de la señal uno quiere habitualmente actuar sobre las frecuencias (por ejemplo para eliminar ruido o crear algunos efectos o reducir el ancho de banda) y la relación anterior dice que actuar sobre las frecuencias es lo mismo que convolver la señal. Muchos matemáticos teóricos posiblemente se sorprenderían de la cantidad de libros de texto para ingenieros basados en esta fórmula que estudian mil y una formas de diseñar e implementar convoluciones útiles.

En el caso de \mathbb{Z}_N una señal es lo mismo que un vector $\vec{x} \in \mathbb{C}^N$. Si lo convolvemos con otro vector \vec{y} que sólo tiene las K primeras coordenadas no nulas, las coordenadas de la convolución serán:

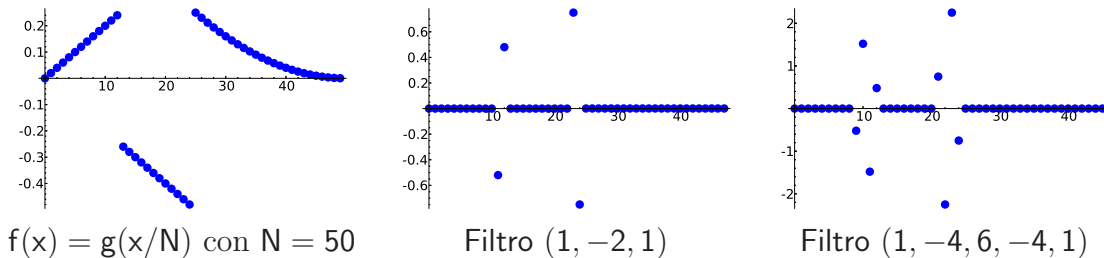
$$(1.37) \quad (\vec{x} * \vec{y})_n = \sum_{k=0}^{K-1} y_k x_{n-k}$$

donde hay que entender $n-k$ módulo N , aunque en algunas aplicaciones conviene definir $x_{n-k} = 0$ si $n-k \notin [0, N)$. Diremos que hemos aplicado el filtro $(y_0, y_1, \dots, y_{K-1})$. En teoría de la señal esto es lo que se llama un **FIR** (del inglés *Finite Impulse Response*). Un filtro muy sencillo para detectar la “regularidad” de una señal es $(1, -2, 1)$. La idea es que una señal suave satisface $f(n) \approx (f(n-1) + f(n+1))/2$ y por tanto el resultado de aplicar este filtro debe ser cero excepto en los puntos donde hay un problema. Matemáticamente, se tiene

$$\lim_{h \rightarrow 0} \frac{f(x-h) - 2f(x) + f(x+h)}{h^2} = f''(x) \quad \text{y} \quad \lim_{h \rightarrow 0} \frac{f(x-2h) - 4f(x-h) + 6f(x) - 4f(x+h) + f(x+2h)}{h^4} = f^{(4)}(x),$$

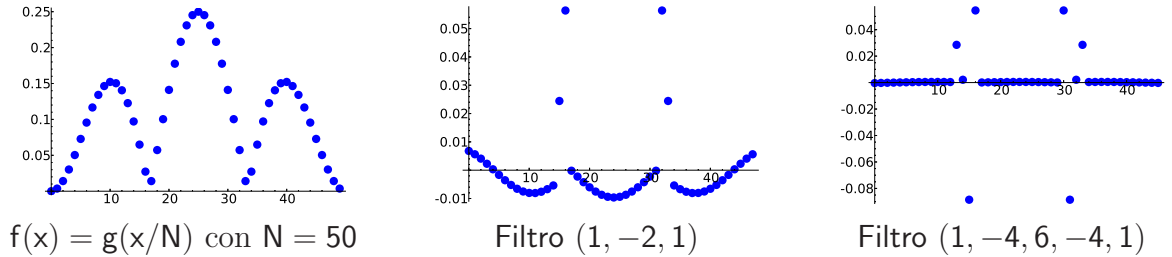
por tanto seguramente $(1, -4, 6, -4, 1)$ es un filtro más sensible para detectar faltas de regularidad.

Con la función $g(x) = x$ en $[0, 1/4)$, $g(x) = -x$ en $x \in [1/4, 1/2)$ y $g(x) = (1-x)^2$ en $[1/2, 1]$, vemos que realmente estos filtros cumplen su cometido.



Por cierto, la multiplicidad de los puntos “detectores” está relacionada con la longitud del filtro y desde el punto de vista práctico no es un inconveniente porque hace más notoria la falta de suavidad.

El segundo filtro no parece dar más información pero eso es debido a que el ejemplo es demasiado simple. Tomando $g(x) = x(1 - x)|\text{sen}(3\pi x)|$, ya notamos la diferencia:



La falta de regularidad es un enemigo fundamental para el éxito del análisis de Fourier, como vimos al introducir la transformada coseno discreta, por tanto detectar los puntos malos es interesante de por sí. Sin embargo vamos a aplicar esta idea en unos ejemplos más llamativos que posiblemente nos hayamos encontrado alguna vez sin saber la teoría que escondían.

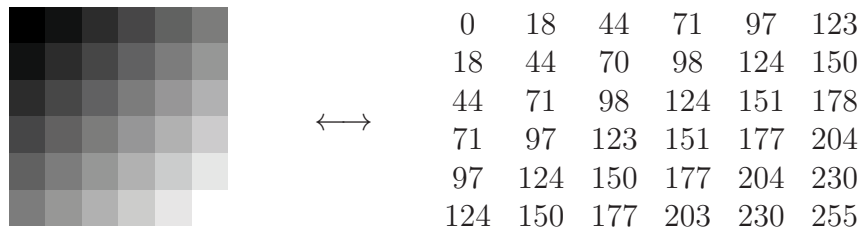
Lo que hemos hecho en \mathbb{Z}_N tiene su traducción inmediata a $\mathbb{Z}_N \times \mathbb{Z}_M$. Los filtros y las señales (las funciones) en $\mathbb{Z}_N \times \mathbb{Z}_M$ los podemos ver como matrices. Un filtro (y_{jk}) transforma la matriz (x_{nm}) haciendo que su elemento nm sea $\sum y_{jk}x_{n-j}m-k$, es decir utilizando (1.37) en cada uno de los índices.

Consideremos los filtros dados por las matrices

$$H = \begin{pmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \quad HV = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad Ob = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

Para detectar faltas de regularidad en las horizontales podemos usar H , y H^t para las verticales, mientras que $HV = H + H^t$ tiene en cuenta ambas direcciones. Si además introducimos las direcciones oblicuas noreste y sudeste, el filtro natural es Ob .

Una imagen en blanco y negro (se sobreentiende que en tonos de gris) de tamaño $N \times M$ viene representada por una función $f : \mathbb{Z}_N \times \mathbb{Z}_M \rightarrow \mathbb{R}$ (porque $f_R = f_G = f_B$ con la notación de la introducción). De hecho su imagen se representa con un solo byte, es decir, $\text{Im } f = \{0, 1, \dots, 255\}$ con 0 correspondiendo al negro y 255 al blanco.



Los filtros HV y Ob aplicados a una imagen f deberían detectar bordes, y lo confirman las siguientes figuras, donde se ha representado el resultado en negativo para que sea más fácil visualizarlo (0 a blanco y 255 a negro). Para los valores de la convolución fuera del rango $[0, 255]$ hay que tomar una decisión, aquí se ha seguido la habitual de asignar 0 a los negativos y 255 a los que son mayores que éste.



Imagen original



Filtro HV (negativo)



Filtro Ob (negativo)

El resultado parece poco denso y un poco aleatorio porque estos filtros detectan variaciones en un rango de 3 pixels, que es una unidad pequeña (en un monitor típicamente $0.75mm$). Trabajando en bloques de tamaño doble se obtiene:

Filtro HV' (negativo)Filtro Ob' (negativo) $\frac{1}{8}Ob'$ (negativo) umbral 240

Aquí HV' y Ob' son como HV y Ob sustituyendo cada elemento por una celda 2×2 con todos sus valores iguales. De esta forma, HV' y Ob' son matrices 6×6 . La mayor norma de Ob' hace que demasiados valores sobrepasen el umbral 255 (negro en el negativo). Para evitarlo, se ha dividido en la última figura el filtro por 8 y se ha bajado el umbral a 240. Con ello hemos conseguido imitar el efecto de transformar una foto en un dibujo a lápiz (véase una variante en [BB09, §4.4]).

La mayor parte de los filtros de los programas de retoque fotográfico están basados en convoluciones.

En el conocido programa libre GIMP, es posible elegir una matriz de convolución para crear un filtro propio (en el menú `Filters>Generic>Convolution Matrix`). En la documentación correspondiente se lee “*Here is a mathematician’s domain. [...] What is a convolution matrix? It’s possible to get a rough idea of it without using mathematical tools that only a few ones know.*” Lo curioso es que a los matemáticos en formación que conocen sobradamente las herramientas, rara vez se les explica su utilidad.

El formato más difundido en la fotografía digital amateur es el formato JPEG. Estas siglas se refieren al comité que introdujo sus especificaciones (*Joint Photographic Experts Group*) pero ha quedado como nombre del propio formato. De nuevo consideraremos una imagen en blanco y negro (tonos de gris), representada por una función $f : \mathbb{Z}_N \times \mathbb{Z}_M \rightarrow \mathbb{R}$ cuya imagen está en realidad en $\{0, 1, \dots, 255\}$. Por cierto, en las fotografías en color, el formato JPEG no trata cada uno de los canales f_R , f_G y f_B por separado sino que hace un cambio simple de coordenadas [Sal02, p.145] porque los experimentos prueban que el ojo humano presenta distinta sensibilidad a los colores dependiendo de su brillo. Después de este cambio de coordenadas todo es similar al caso de blanco y negro.

El objetivo del formato es comprimir la foto perdiendo totalmente cierto número de coeficientes de Fourier y parte de la precisión en el resto. Es por ello un formato de compresión con pérdidas, eso explica por qué las carpetas repletas de ficheros `.jpg` apenas sufren cambios en su tamaño cuando se les aplican programas de compresión y también por qué editar muchas veces una foto (abrir un JPEG y grabarlo como JPEG) reduce su calidad.

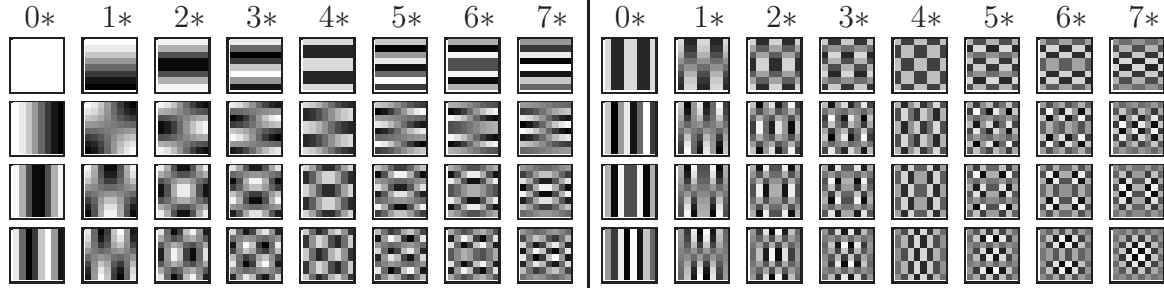
Una foto típica corresponde a una función $f : \mathbb{Z}_N \times \mathbb{Z}_M \rightarrow \mathbb{R}$ nada suave (hay cambios abruptos de color al pasar de un objeto a otro), por ello el análisis de Fourier discreto se muestra poco útil de cara a una aplicación directa. Lo que se hace es descomponer la imagen $N \times M$ en porciones de 8×8 pixels (si $8 \nmid N$ o si $8 \nmid M$, se agranda artificialmente la imagen). Cada porción 8×8 tendrá asignada su función $F : \mathbb{Z}_8 \times \mathbb{Z}_8 \rightarrow \mathbb{R}$ y para una foto típica la inmensa mayoría de estas funciones F será bastante suave, una gradación de color, porque los cambios abruptos de color ocurren en una minoría de los MN pixels de la imagen original (los contornos de los objetos son un conjunto unidimensional). Utilizando la transformada coseno discreta, ya hemos visto que se eliminan los problemas derivados de que después del 7 viene el 0 en \mathbb{Z}_8 . Por ello los armónicos naturales para analizar F son

$$\phi_{nm}(k, l) = \cos\left(\frac{\pi n}{16}(2k + 1)\right) \cos\left(\frac{\pi m}{16}(2l + 1)\right).$$

Por la Proposición 1.3.6 (la fórmula de inversión) aplicada en cada uno de los índices, se deduce que cualquier $F : \mathbb{Z}_8 \times \mathbb{Z}_8 \rightarrow \mathbb{R}$ admite un desarrollo de Fourier discreto

$$(1.38) \quad F = \sum_{n=0}^7 \sum_{m=0}^7 a_{nm} \phi_{nm}.$$

De alguna forma tenemos representada cualquier imagen 8×8 como superposición de imágenes básicas 8×8 . Asignando al negro el -1 y al blanco el 1 , el aspecto de estas imágenes básicas es:



En la mitad izquierda están los armónicos ϕ_{nm} con $0 \leq m < 4$ y en la derecha, los armónicos ϕ_{nm} con $4 \leq m < 8$.

Los coeficientes de Fourier se calculan con cierta precisión (al guardar un fichero JPEG con GIMP, en `Advanced Options > DCT method` hay cierto control sobre ello). Parece claro, y las experiencias lo avalan, que el ojo es poco sensible a cambios rápidos de brillo en áreas pequeñas, es decir, que ve peor los armónicos con frecuencias mayores. La *matriz de cuantificación* es una matriz entera $C = (c_{nm})_{n,m=0}^7$ de naturaleza experimental tal que c_{nm} está en relación inversa a la “visibilidad” del armónico ϕ_{nm} . La matriz más empleada es

$$C = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}$$

En vez de los coeficientes de Fourier a_{nm} , lo que se almacena son los enteros λ_{nm} más cercanos a a_{nm}/c_{nm} . Si F viene dada por 64 valores enteros y ahora almacenamos los λ_{nm} que son otros 64 enteros, cabe preguntarse dónde está la compresión. La respuesta es que típicamente los λ_{nm} son muchas veces cero o valores pequeños, y en estas condiciones, la lista de todos los λ_{nm} para todas las porciones 8×8 de la foto original es fácil de comprimir (sin pérdidas) con métodos habituales (que no dependen del análisis de Fourier y no veremos aquí [Sal02, §3]). Cada vez que F sea suave (una gradación de color) bastarán los primeros coeficientes de Fourier para reconstruir muy bien F con (1.38) y el resto serán prácticamente nulos. Además, la matriz de cuantificación penaliza las frecuencias grandes y hace que los λ_{nm} estén sesgados hacia valores pequeños.

Cuando se abre una imagen JPEG, internamente en el ordenador se descomprime el fichero y se obtienen los λ_{nm} , con ellos se aproximan los a_{nm} por medio de $a_{nm} \approx c_{nm}\lambda_{nm}$. Finalmente se utiliza (1.38) para reconstruir cada porción 8×8 de la imagen.

A pesar de que el formato JPEG logra habitualmente una compresión notoria sin pérdida de calidad aparente, está claro que tiene ciertas deficiencias. La más clara es que la división en porciones 8×8 es arbitraria. Una foto puede tener zonas mucho mayores con gradaciones de color en las que el análisis de Fourier discreto funciona bien.

Lo ideal sería que hubiera un proceso adaptativo en el que el tamaño de los armónicos se ajustase a las diferentes zonas de la imagen. Esta idea se manifiesta en gran medida en el nuevo formato JPEG 2000 que emplea unos armónicos bien diferentes de los que hemos visto hasta ahora y que estudiaremos a continuación.

Las *wavelets* (a veces *ondículas* en español) son unos armónicos que han causado una revolución. Sobre todo en sus comienzos, se resaltó su importancia en algunas cuestiones teóricas pero claramente su mayor influencia ha sido en el campo de las aplicaciones a la hora de tratar y comprimir diversos tipos de datos. Las ventajas de las wavelets en estas aplicaciones son por un lado que la implementación es bastante sencilla porque se basa en un algoritmo iterativo simple y, en contraposición con el análisis de Fourier, que los armónicos están localizados. Pensemos por ejemplo que la lentitud de la convergencia de la primera serie considerada por Fourier (1.2) se debe a que hay una discontinuidad. Todos los armónicos sienten esta discontinuidad que afecta globalmente a la convergencia. Las wavelets permiten un análisis más local que el análisis de Fourier. Aunque el ámbito discreto es el que más aparece en las aplicaciones, las wavelets no están limitadas a él, de hecho nacieron en \mathbb{R} y su propio nombre deriva de que en su forma continua son “ondas breves”. Hay muchas monografías matemáticas acerca de los aspectos teóricos de las wavelets en su forma continua, por ejemplo [HW96].

En esta descripción breve (seguimos en parte [Ger99]), sólo describiremos un miembro sencillo pero destacado de la tribu de las wavelets. Lo que se llama a veces D4 (o Daub4 [Wal08]), donde la nomenclatura hace referencia a I. Daubechies, la campeona de las wavelets.

Vamos a diseñar un filtro (f_0, f_1, f_2, f_3) para detectar la falta de regularidad que normalizaremos de manera que $f_0^2 + f_1^2 + f_2^2 + f_3^2 = 1$ (más adelante se verá la utilidad de ello). Si queremos que anule todas las rectas $f(n) = \alpha n + \beta$, basta con imponer $f_0 + f_1 + f_2 + f_3 = 0$ y $0f_0 + 1f_1 + 2f_2 + 3f_3 = 0$. Además añadimos la condición extraña de que sea ortogonal consigo mismo cuando se traslada dos unidades, esto es, $f_2f_0 + f_3f_1 = 0$. Este sistema no lineal de 4 ecuaciones con 4 incógnitas tiene la solución:

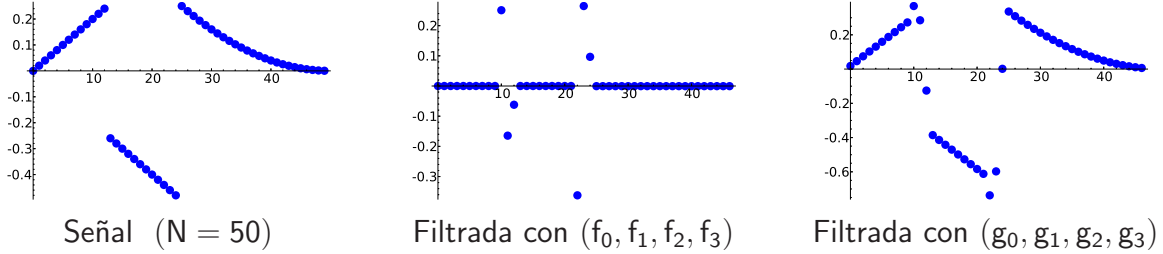
$$f_0 = \frac{1 - \sqrt{3}}{4\sqrt{2}}, \quad f_1 = \frac{\sqrt{3} - 3}{4\sqrt{2}}, \quad f_2 = \frac{3 + \sqrt{3}}{4\sqrt{2}}, \quad f_3 = -\frac{1 + \sqrt{3}}{4\sqrt{2}}.$$

Consideremos ahora el filtro (g_0, g_1, g_2, g_3) más simple que a uno se le ocurriría que sea ortogonal a (f_0, f_1, f_2, f_3) y a su trasladado dos veces:

$$g_0 = \frac{1 + \sqrt{3}}{4\sqrt{2}}, \quad g_1 = \frac{3 + \sqrt{3}}{4\sqrt{2}}, \quad g_2 = \frac{3 - \sqrt{3}}{4\sqrt{2}}, \quad g_3 = \frac{1 - \sqrt{3}}{4\sqrt{2}}.$$

Simplemente se han intercambiado f_0 con f_2 y f_1 con f_3 , modificando los signos para forzar la ortogonalidad. Este nuevo filtro no detecta bordes sino que hace un promedio escalado por $\sqrt{2}$ porque $g_0 + g_1 + g_2 + g_3 = \sqrt{2}$.

Sobre la señal que ya habíamos filtrado antes, vemos que estas ideas se ajustan a la realidad.



Por construcción, la matriz circulante

$$T = \begin{pmatrix} g_0 & g_1 & g_2 & g_3 & 0 & 0 & \dots & 0 & 0 \\ f_0 & f_1 & f_2 & f_3 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & g_3 & & & \\ 0 & 0 & f_0 & f_1 & f_2 & f_3 & & & \\ \vdots & \vdots & & & & & \ddots & & \\ g_2 & g_3 & 0 & 0 & 0 & 0 & \dots & g_0 & g_1 \\ f_2 & f_3 & 0 & 0 & 0 & 0 & \dots & f_0 & f_1 \end{pmatrix} \quad \text{cumple} \quad T^t T = I,$$

es decir, es ortogonal, para ello impusimos algunas de las condiciones sobre los filtros. Cuando se la aplicamos a una señal $\vec{x} \in \mathbb{R}^N$, las coordenadas en lugar impar dan promedios y las de lugar par, faltas en la suavidad. Tras aplicar T , la ortogonalidad implica que la “energía” de la señal, su norma la cuadrado $\|\vec{x}\|^2$, se conserva. Repitiendo el proceso con el vector de promedios en vez de con toda la señal, conseguimos detalles más finos sobre la falta de suavidad (porque los promedios involucran más puntos). Supongamos $N = 2^k$ y definamos la recurrencia partiendo de $\vec{a}_0 = \vec{x}$,

$$(1.39) \quad (\vec{a}_{n+1}, \vec{d}_{n+1}) = \text{dentr}(T\vec{a}_n), \quad n = 0, 1, \dots, k-2$$

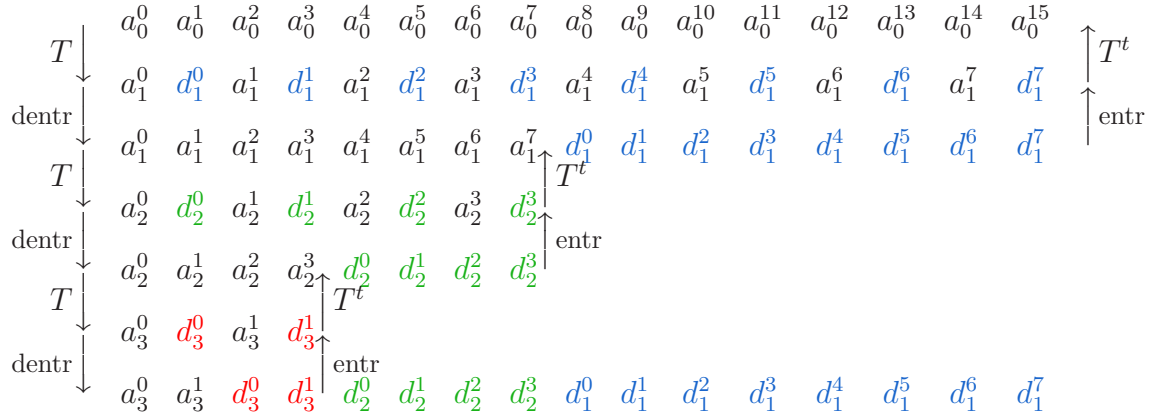
donde dentr indica el operador (lineal) que “desentrelaza” un vector poniendo primero las coordenadas que estaban en lugar impar y después las que estaban en lugar par. Se dice que los \vec{a}_j son los *vectores de aproximación* y que los \vec{d}_j son los *vectores de detalle*. Como $T^{-1} = T^t$, el proceso se puede invertir mediante

$$(1.40) \quad \vec{a}_n = T^t \text{entr}(\vec{a}_{n+1}, \vec{d}_{n+1}), \quad n = 0, 1, \dots, k-2$$

donde $\text{entr}(\vec{a}_{n+1}, \vec{d}_{n+1})$ es el resultado de entrelazar \vec{a}_{n+1} y \vec{d}_{n+1} , poniendo el primer vector en los lugares impares y el segundo en los pares.

Por iteraciones sucesivas, (1.40) permite recuperar la señal \vec{x} a partir del último vector de aproximación $\vec{a}_{k-1} \in \mathbb{R}^2$, que da una especie de promedio general, y de todos los vectores de detalle $\vec{d}_1, \vec{d}_2, \dots, \vec{d}_{k-1}$ que tienen respectivamente $N/2, N/4, \dots, 2$ coordenadas. En total hay $N = N/2 + N/4 + \dots + 2 + 2$ grados de libertad, como es natural porque $\vec{x} \in \mathbb{R}^N$.

Por ejemplo, para una señal $\vec{x} = \vec{a}_0 \in \mathbb{R}^{16}$, las iteraciones de (1.39) y de (1.40) corresponden, respectivamente, a seguir este esquema en sentido descendente y ascendente:

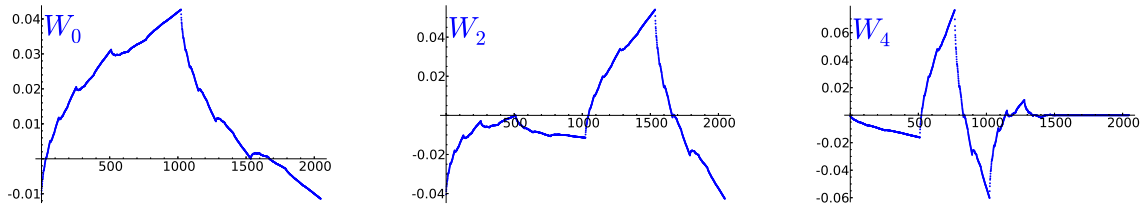


La aplicación W que dada una señal \vec{x} , le asigna $(\vec{a}_{k-2}, \vec{d}_{k-1}, \vec{d}_{k-2}, \dots, \vec{d}_1)$ por medio de (1.39), $N = 2^k$, se llama *transformada wavelet discreta* y a W^{-1} se le llama *transformada wavelet discreta inversa*. Si $\{\vec{e}_0, \vec{e}_1, \dots, \vec{e}_{N-1}\}$ es la base usual de \mathbb{R}^N , entonces por la linealidad

$$(1.41) \quad W(\vec{x}) = \sum_{j=0}^{N-1} \lambda_j \vec{e}_j \quad \implies \quad \vec{x} = \sum_{j=0}^{N-1} \lambda_j W_j \quad \text{con} \quad W_j = W^{-1}(\vec{e}_j).$$

Hemos descompuesto \vec{x} en términos de unos armónicos W_j que llamamos *wavelets*.

Es instructivo, ver el aspecto de algunas de las wavelets cuando N es muy grande. La sorpresa es que en cierto modo sólo hay dos, W_0 y W_2 (a veces llamadas padre y madre de las wavelets) porque W_1 es un trasladado de W_0 y a partir de W_2 el resto son escalados y trasladados; cuanto mayor es el índice mayor es la localización. Por ejemplo, para $N = 2048$,



Otra sorpresa es que estos armónicos no son muy suaves en el límite $N \rightarrow \infty$, más bien tienen un aspecto fractal. Hay otras variantes más complicadas que tienen mayor regularidad [Wal08].

Vamos a ilustrar la utilidad de las wavelets para reducir el ruido de una señal. Digamos que a nuestra señal le sumamos una muestra de la misma longitud correspondiente a una distribución normal $N(0, \sigma)$. Esto es lo que se llama *ruido blanco gaussiano*. En el

lado de la transformada de Fourier, es decir, en cuanto a las frecuencias se refiere, este tipo de ruido se manifiesta como una perturbación aleatoria de todos los coeficientes de Fourier. Bajo la hipótesis de que el ruido no es muy acusado (σ pequeño), una manera de reducirlo es eliminar los coeficientes de Fourier con tamaño menor que cierta cantidad t .

Con la Proposición 1.3.6, este proceso de reducción de ruido sería

$$(1.42) \quad \vec{x} \mapsto \left(\frac{c_0}{N} + \frac{2}{N} \sum_{n=1}^{N-1} c_n \cos \left(\frac{\pi n}{N} \left(m + \frac{1}{2} \right) \right) \right)_{n=0}^{N-1} \quad \text{donde } c_n = \begin{cases} 0 & \text{si } |\hat{x}_n^c| < t \\ \hat{x}_n^c & \text{si } |\hat{x}_n^c| \geq t \end{cases}$$

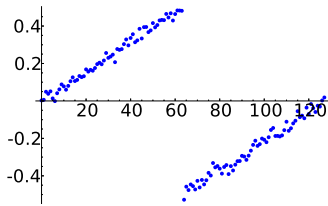
El problema es que si la señal de partida tiene muchos coeficientes de Fourier pequeños que contribuyen efectivamente a ella, estamos perdiendo también la señal. Esto ocurrirá necesariamente cuando la suavidad de la señal es poca (porque los coeficientes de Fourier decaen lentamente).

El análogo con wavelets está basado en (1.41) y es

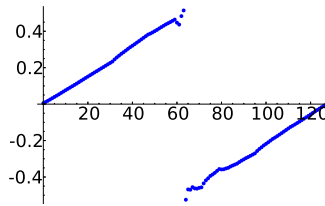
$$(1.43) \quad \vec{x} \mapsto \sum_{j=0}^{N-1} c_j W_j \quad \text{donde } c_j = \begin{cases} 0 & \text{si } |\lambda_j| < t \\ \lambda_j & \text{si } |\lambda_j| \geq t \end{cases}$$

La ventaja de las wavelets es que como sus armónicos están localizados, son menos sensibles a singularidades locales. Desde un punto de vista más teórico, el ruido blanco gaussiano sigue siendo ruido blanco gaussiano del mismo tipo al aplicar W (porque T es ortogonal).

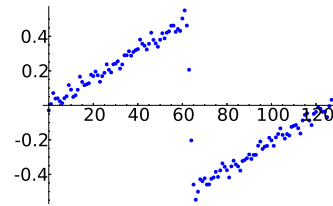
Veamos un ejemplo. Contaminamos la señal $f(n) = \{n/N\}$, $N = 128$ con ruido gaussiano $N(0, 1/50)$, y ponemos como umbral $t = 4,5\sigma = 0,09$ en (1.43) [Wal08], la reducción de ruido es bastante buena. Sin embargo con (1.42) el resultado no es satisfactorio para $t = 0,9$



Señal con ruido (N = 128)



(1.43), $t = 0,09$



(1.42), $t = 0,9$

Con otros valores de t , la reducción de ruido con (1.42) no mejora. Esencialmente si t es pequeño se obtiene un resultado como el indicado y si no lo es, se ven unas ondas regulares que oscilan alrededor de la señal.