La transformada rápida de Fourier

Pablo Ibáñez Trallero

Trabajo de fin de grado en Matemáticas
Universidad de Zaragoza

Directores del trabajo: Ángel Ramón Francés Román,
Julio Bernués Pardo

Mayo de 2022

Abstract

The Fourier Transform has become a powerful analytical tool in several fields of science and applied disciplines. Since its beginnings with the heat equation, the importance of Fourier Analysis has reached areas such as signal processing, quantum mechanics or partial differential equations, solving many problems in classical mathematical physics.

In general terms, Fourier Analysis studies how general functions can be decomposed into trigonometric functions; it searches general conditions to ensure its existence and provides convergence results.

Particularly, if our function $f : \mathbb{R} \longrightarrow \mathbb{C}$ is 1-periodic, Fourier Analysis studies if it is possible to write it as a discrete sum of trigonometric functions:

$$\sum_{n=-\infty}^{\infty} \hat{f}(n) e^{2\pi i n t}, \quad t \in \mathbb{R}$$

This famous series is called the *"Fourier series"*, where each term is composed by the n-th Fourier coefficient, $\hat{f}(n) \in \mathbb{C}$, and the function $e^{2\pi i n t}$, commonly called the n-th "frequency". With this objective in mind, a wide area of study is opened describing function conditions, coefficients properties, relations between spaces or convergence results.

Specifically, if we choose a function f that is continuous and piecewise $\mathscr{C}^{(1}$, we achieve the best situation possible:

$$f(t) = \sum_{n=-\infty}^{\infty} \hat{f}(n) e^{2\pi i n t}, \quad t \in \mathbb{R}$$

At this point, the next question is, how do we compute those Fourier coefficients? Theory gives us a clear answer in terms of f:

$$\hat{f}(n) = \int_{-1/2}^{1/2} f(t) e^{-2\pi i n t} dt, \quad n \in \mathbb{Z}$$

Nevertheless, in practice we rarely know the expression of f. Instead of that, a discrete set of values $f(t_0), \ldots, f(t_n)$ are provided. Consequently, the question changes to, is there any way to compute the Fourier coefficients only with those values?

This project will answer that question. Going through the Fourier Transform we will derive, step by step, a method to obtain approximations of the Fourier coefficients given a certain discrete set of values of f. Moreover, we will bring this method to practice efficiently with one of the most important algorithms in the last century, the Fast Fourier Transform (FFT). We will chop this algorithm developing core understanding of each section that composes it. In more detail:

The first chapter will introduce the Discrete Fourier Transform right from the start. This context will provide a system

$$\sum_{n=0}^{N-1} a_n e^{2\pi i n \frac{k}{N}} = f_k, \quad k = 0, \ldots, N-1$$

where, given a set of N values of f, $[f_0, \cdots, f_{N-1}]$, its solution $[a_0, \cdots, a_{N-1}]$ will be certain N approximated Fourier coefficients.

We will prove that this system has a solution, and we will describe it explicitly with the following product:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{N-1} \end{bmatrix} = \frac{1}{N} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2N-2} & \cdots & \omega^{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_{N-1} \end{bmatrix}, \quad \omega := e^{-2\pi i/N} \tag{1}$$

Moreover, we will ensure that the approximations are good enough for the most common families of functions, and we will study how the method is fixed if the number of values of f is even or if the functions have different periods.

The second chapter will develop the Fast Fourier Transform algorithm itself. Keeping in mind the objetive of computing the previous product (1) and writing the matrix as $\mathbf{F}_N = \frac{1}{N}\left[\omega^{ij}\right]_{0\leq i,j<N}$, we will derive the following factorization for that matrix when the number of values of f is $N = 2^m$:

$$\mathbf{F}_N = \frac{1}{2^m}\mathbf{Q}_{2^m}\mathbf{Q}^{(2)}_{2^{m-1}}\mathbf{Q}^{(4)}_{2^{m-2}}\cdots\mathbf{Q}^{(2^{m-1})}_{2}\mathbf{B}_{2^m}$$

And in general, when $N = P_1\cdots P_m$:

$$\mathbf{F}_N = \frac{1}{P_1\cdots P_m}\cdot\mathbf{Q}_{P_1\cdots P_{m-1},P_m}\cdot\mathbf{Q}^{(P_m)}_{P_1\cdots P_{m-2},P_{m-1}}\cdots\mathbf{Q}^{(P_3\cdots P_m)}_{P_1,P_2}\cdot\mathbf{Q}^{(P_2\cdots P_m)}_{1,P_1}\cdot\mathbf{S}_{P_1,P_2,\ldots,P_{m-1},P_m}$$

In fact, these factorizations and the properties of the matrix involved will derive the general scheme of the FFT algorithm. We will study each part of the algorithm in detail.

Finally, the appendix will contain a brief collection of Fourier Analysis results, gathering the essential ones that will be needed in the first chapter.

Índice general

Capítulo 1

La transformada de Fourier discreta

1.1. Introducción

En este capítulo desarrollaremos un método para obtener coeficientes de Fourier aproximados de ciertos tipos de funciones. Con el objetivo de hallar dichos coeficientes, plantearemos el problema en términos del análisis de Fourier, introduciremos el concepto de la transformada de Fourier discreta y daremos contexto teórico al camino elegido para calcularlos. Además, justificaremos (como cabe esperar de cualquier método) la convergencia de la aproximación a los coeficientes de Fourier deseados. Por otro lado también abordaremos algunas generalizaciones del método relativas al número de elementos del muestreo o al periodo de las funciones, para cubrir así las posibles situaciones que se puedan dar en la práctica.

Con todo este desarrollo trataremos de despejar el origen del producto matriz-vector que inicia el posterior capítulo sobre el algoritmo FFT, ayudando así a una mejor comprensión del significado de los elementos que lo conforman.

1.2. Idea principal

Sea $f : \mathbb{R} \longrightarrow \mathbb{C}$ una función 1-periódica, continua y $C^{(1}$ a trozos. Sean $\{\hat{f}(n)\}_{n\in\mathbb{Z}}$ sus correspondientes coeficientes de Fourier (Def. 9). En particular, por el teorema (A.2) sabemos que su serie de Fourier (Def. 9) cumple que

$$f(t) = \sum_{n=-\infty}^{\infty} \hat{f}(n)e^{2\pi int}, \text{ uniformemente en } \mathbb{R}. \tag{1.1}$$

Que expresado en términos de sumas parciales (Def. 9) significa que

$$S_M f(t) = \sum_{n=-M}^{M} \hat{f}(n)e^{2\pi int}$$

converge uniformemente a $f(t)\ \forall t \in \mathbb{R}$.

Es decir, $\forall \varepsilon > 0$ podemos encontrar un $M \in \mathbb{N}$ lo suficientemente grande tal que:

$$|f(t) - S_M f(t)| < \varepsilon \quad \forall t \in \mathbb{R}$$

Por ello, con un M relativamente grande podemos asumir que $S_M f(t) \approx f(t)\ \forall t \in \mathbb{R}$. No obstante, a partir de ahora, nos fijaremos únicamente en el intervalo $[0,1]$; ya que al ser periódica, reúne toda la información de la función. De hecho, cabe mencionar que en la práctica tendremos una función definida en un intervalo y la extenderemos periódicamente para poder usar los resultados anteriores del análisis de Fourier.

Como bien es sabido en la práctica, generalmente no se conoce la expresión explícita de la función f, sino que lo único que se puede hacer es manejar un muestreo de la función. Con esta restricción, ¿es posible hallar una aproximación de $\hat{f}(n)$ conocidos únicamente los valor de f en un mallado de puntos? Supongamos que tenemos un muestreo de f de $2M+1$ puntos equidistribuidos en el intervalo $[0,1]$:

$$f(\frac{k}{2M+1}), \quad k=0,\dots,2M$$

Entonces, usando las propias definiciones de las sumas parciales

$$\sum_{n=-M}^{M} \hat{f}(n) e^{2\pi i n \frac{k}{2M+1}} = S_M f(\frac{k}{2M+1}), \quad k=0,\dots,2M \tag{1.2}$$

y la aproximación $S_M f(t) \approx f(t)\ \forall t \in [0,1]$, podemos crear el siguiente sistema:

$$\sum_{n=-M}^{M} \hat{f}^*(n) e^{2\pi i n \frac{k}{2M+1}} = f(\frac{k}{2M+1}), \quad k=0,\dots,2M \tag{1.3}$$

Observación. Notemos que las incógnitas $\hat{f}(n)$ se han cambiado por a_n, debido a que al sustituir por la aproximación, el sistema ya no será el mismo. □

La elección de este sistema no es casual, ya que se hace esperando que la convergencia de las series de Fourier se traslade a la solución, si existe, del sistema. Es decir, esperando que las incógnitas $\hat{f}^*(n)$, $n=-M,\dots,M$ también se aproximen a los coeficientes de Fourier $\hat{f}(n)$, $n=-M,\dots,M$. Si esto fuera así, dado el muestreo adecuado de f este sistema obtendría una aproximación de los correspondientes coeficientes de Fourier, respondiendo así a la pregunta planteada.

En las posteriores secciones veremos que efectivamente este sistema tiene solución y que converge a dichos coeficientes. No obstante, para lograr ese objetivo, empezaremos hallando la solución del siguiente sistema:

$$b_k = \sum_{n=0}^{N-1} c_n e^{2\pi i n \frac{k}{N}} \quad k=0,\dots,N-1, \quad [b_0,\dots,b_{N-1}]^T \in \mathbb{C}^N$$
$$[c_0,\dots,c_{N-1}]^T \in \mathbb{C}^N$$

1.3. Existencia y solución explícita

Definamos primeramente la siguiente familia de vectores:

$$w_k = \left[\tau^{k\cdot 0}, \tau^{k\cdot 1}, \dots, \tau^{k\cdot N-1}\right]^T, \quad k=0,\dots,N-1$$

Siendo $\tau = e^{2\pi i/N}$. Nótese que tenemos una familia de vectores $\{w_k\}_{k=0}^{N-1} \subset \mathbb{C}^N$ cuyas componentes son raíces N-ésimas de la unidad. Más aún, podemos formar una base ortonormal con ella:

Lema 1.1. *Sean l,m índices en $0,\dots,N-1$. Entonces,*

$$\langle w_l, w_m \rangle = N\delta_{l,m}$$

Por tanto, la familia $\{\frac{w_k}{\sqrt{N}}\}_{k=0}^{N-1}$ es una base ortonormal en $\mathbb{C}^N$.

Demostración. Dados l,m y sabiendo que $\bar{\tau} = \tau^{-1}$ tenemos que

$$\langle w_l, w_m \rangle = \sum_{n=0}^{N-1} \tau^{nl}\tau^{-nm} = \sum_{n=0}^{N-1} \left(\tau^{l-m}\right)^n = \begin{cases} N & \text{si} \quad l=m \\ \frac{1-r^N}{1-r}, \quad r=\tau^{l-m} & \text{si} \quad l\neq m \end{cases}$$

Y como r es una raíz N-ésima, tenemos que $r^N = 1$, obteniendo así el resultado. La base normalizada completaría el lema. □

Por ello, dado un elemento $b \in \mathbb{C}^N$ tenemos una descomposición única en esa base:

$$b = \sum_{n=0}^{N-1} a_n \frac{w_n}{\sqrt{N}}, \text{ donde } a_n = \langle b, \frac{w_n}{\sqrt{N}} \rangle, \quad n = 0, \ldots, N-1$$

Reescrito de otra forma:

$$b = \frac{1}{N} \sum_{n=0}^{N-1} c_n w_n, \text{ donde } c_n = \langle b, w_n \rangle, \quad n = 0, \ldots, N-1$$

En otros términos, hemos probado el siguiente resultado:

Teorema 1.2. *Sean* $b = [b_0, \ldots, b_{N-1}]^T$, $c = [c_0, \ldots, c_{N-1}]^T \in \mathbb{C}^N$. *Entonces el sistema de ecuaciones*

$$b_k = \sum_{n=0}^{N-1} c_n e^{2\pi i n \frac{k}{N}} = \sum_{n=0}^{N-1} c_n (w_n)_k \quad k = 0, \ldots, N-1$$

Tiene solución y queda determinada por

$$c_n = \frac{1}{N} \langle b, w_n \rangle = \frac{1}{N} \sum_{k=0}^{N-1} b_k e^{-2\pi i n \frac{k}{N}} \quad n = 0, \ldots, N-1$$

Usando los resultados elementales de bases de álgebra lineal hemos hallado la solución explícita del sistema. De hecho, en términos matriciales, hemos obtenido la matriz inversa del sistema. Denotando $\omega := \bar{\tau} = e^{-2\pi i/N}$, la solución queda expresada matricialmente de la siguiente forma:

$$c = \mathbf{F}_N b$$

Donde

$$\mathbf{F}_N = \frac{1}{N} \left[\omega^{ij} \right]_{0 \leq i,j < N} = \frac{1}{N} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2N-2} & \cdots & \omega^{(N-1)(N-1)} \end{bmatrix} \tag{1.4}$$

Cómo realizar el producto $\mathbf{F}_N b$ será el motivo del desarrollo del posterior capítulo. Esta disposición concreta de la matriz y, las raíces N-ésimas de la unidad y su estructura de grupo tendrán un papel crucial.

1.4. La transformada discreta de Fourier

Si nos fijamos en la formulación del sistema (1.3) que habíamos planteado en el inicio del capítulo, y la comparamos con la del sistema del teorema (1.2), podemos apreciar diferencias en los índices del sumatorio: el primero recorre índices positivos $0, \ldots N-1$ y el otro los recorre de manera simétrica $-M, \ldots M$. Con la transformada discreta veremos que ambos sistemas representan el mismo problema, y que por tanto, el sistema (1.3) en el que estamos interesados tiene solución.

Definición 1. Diremos que una sucesión de números complejos a $= \{a_n\}_{n \in \mathbb{Z}}$ es N-periódica si

$$a_{n+N} = a_n \quad \forall n \in \mathbb{Z}$$

Observación. De manera más general, podemos ver una sucesión periódica como una aplicación:

$$a : \mathbb{Z}/N\mathbb{Z} \longrightarrow \mathbb{C}$$

Por lo que realmente sólo necesitamos un conjunto de representantes módulo N para determinar la sucesión N-periódica. □

En particular, notemos que podemos extender periódicamente los elementos que conforman nuestro problema de forma natural:

Lema 1.3. *Las sucesiones* $\{f(\frac{k}{N})\}_{k\in\mathbb{Z}}$, $\{e^{-2\pi in\frac{k}{N}}\}_{k\in\mathbb{Z}}$ $(\forall n\in\mathbb{Z})$ *y* $\{e^{-2\pi in\frac{k}{N}}\}_{n\in\mathbb{Z}}$ $(\forall k\in\mathbb{Z})$ *son N-periódicas.*

Demostración. Basta recurrir a la extensión periódica de f para ver que la primera lo es. Por la otra parte, la sucesión de las raíces de la unidad es trivialmente N-periódica. □

Con las sucesiones periódicas presentadas, la transformada discreta se define de la siguiente forma:

Definición 2. Sea $b=\{b_k\}_{k\in\mathbb{Z}}$ una sucesión N-periódica, llamamos transformada de Fourier discreta N de b a la sucesión N-periódica $c=\{c_n\}_{n\in\mathbb{Z}}$ generada por la solución del sistema:

$$b_k=\frac{1}{N}\sum_{n=0}^{N-1}c_n e^{2\pi in\frac{k}{N}}, k=0,\dots,N-1 \tag{1.5}$$

Notemos que el sistema tiene solución única por el teorema (1.2). Y efectivamente, dicha solución es N-periódica:

$$c_{n+N}=\frac{1}{N}\sum_{k=0}^{N-1}b_k e^{-2\pi i(n+N)\frac{k}{N}}=\frac{1}{N}\sum_{k=0}^{N-1}b_k e^{-2\pi in\frac{k}{N}}=c_n,\quad \forall n\in\mathbb{Z}$$

Más aún, con la existencia y unicidad de la solución se tiene el siguiente teorema:

Teorema 1.4. *La transformada de Fourier discreta es una biyección en el conjunto de las sucesiones complejas N-periódicas.*

Observación. Nótese que podríamos haber construido la definición con cualquier otro conjunto de representantes módulo N de las sucesiones b y c. En particular, si $N=2M+1$ el sistema se puede reescribir como:

$$\begin{aligned}b_k&=\frac{1}{2M+1}\sum_{n=0}^{2M}c_n e^{2\pi in\frac{k}{2M+1}}=\frac{1}{2M+1}\left(\sum_{n=0}^{M}c_n e^{2\pi in\frac{k}{2M+1}}+\sum_{n=M+1}^{2M}c_n e^{2\pi in\frac{k}{2M+1}}\right)\\&=\frac{1}{2M+1}\left(\sum_{n=0}^{M}c_n e^{2\pi in\frac{k}{2M+1}}+\sum_{n=-M}^{-1}c_n e^{2\pi in\frac{k}{2M+1}}\right)=\frac{1}{2M+1}\sum_{n=-M}^{M}c_n e^{2\pi in\frac{k}{2M+1}}\end{aligned}$$

Este hecho prueba que los sistemas (1.3) y (1.5) representan el mismo problema y que por tanto el sistema (1.3) inicial tiene solución única, como habíamos mencionado al inicio de la sección. No obstante, nótese que aunque ambos sistemas representen la misma transformada discreta, las matrices de los sistemas serán diferentes (se diferenciarán en una permutación de columnas).

Cabe mencionar que el algoritmo FFT usa la formulación del sistema (1.5) para calcular la transformada discreta. Por este motivo, al usar dicha formulación, en la salida del algoritmo se devuelve el vector con los "coeficientes" negativos seguidos de los negativos.

$$[c_0,c_1,\dots,c_M,c_{M+1},\dots,c_{2M}]=[c_0,c_1,\dots,c_M,c_{-M},\dots,c_{-1}]$$

□

En resumen, hemos obtenido explícitamente la solución del sistema (1.3) que nos interesaba, recurriendo a un sistema equivalente (1.5).

Al mismo tiempo y con la vista puesta en la siguiente sección, con este nuevo contexto podemos reinterpretar los sistemas (1.2) y (1.3) que aparecieron al final del desarrollo de la sección (1.2):

Proposición 1.5. *Sean* S_M *la sucesión* $2M+1$*-periódica generada por* $\left[S_M f(\frac{0}{2M+1}),\dots,S_M f(\frac{2M}{2M+1})\right]$ *y* $\hat{f}_M$ *la sucesión* $2M+1$*-periódica generada por* $\left[\hat{f}(-M),\dots,\hat{f}(M)\right]$*, las sucesiones relativas al sistema (1.2). Entonces,*

$$\hat{f}_M \text{ es la transformada de Fourier discreta de } S_M$$

Proposición 1.6. *Sean f_M la sucesión $2M+1$-periódica generada por $\left[f\left(\frac{0}{2M+1}\right),\ldots,f\left(\frac{2M}{2M+1}\right)\right]$ y $\hat{f}_M^*$ la sucesión $2M+1$-periódica generada por $\left[\hat{f}^*(-M),\ldots,\hat{f}^*(M)\right]$, las sucesiones relativas al sistema (1.3). Entonces,*

$$\hat{f}_M^* \text{ es la transformada de Fourier discreta de } f_M$$

1.5. Convergencia de la transformada discreta

Abordada la existencia de la solución del sistema (1.3), nos faltará ver que esta solución se aproxima a los coeficientes de Fourier correspondientes, para completar así el objetivo planteado al final de la sección (1.2).

Más detalladamente, veremos cómo la convergencia de las series de Fourier juega un papel crucial para obtener la convergencia de la transformada discreta de f_M a la correspondiente sucesión periódica de coeficientes de Fourier $\hat{f}_M$.

Proposición 1.7. *Sea $f:\mathbb{R}\longrightarrow\mathbb{C}$, 1-periódica, continua y de clase $C^{(1}$ a trozos. Entonces, $\forall\varepsilon>0$ existe un $M\in\mathbb{N}$ lo suficientemente grande tal que:*

$$\|\hat{f}_M^*-\hat{f}_M\|_\infty<\varepsilon$$

Siendo $\hat{f}_M^*$ y $\hat{f}_M$ las correspondientes transformadas de Fourier discretas de f_M y S_M (definidas en las proposiciones (1.5) y (1.6)).

Demostración. Sea $\varepsilon>0$. Por el teorema (A.2) existe un $M\in\mathbb{N}$ lo suficientemente grande tal que:

$$|f(t)-S_Mf(t)|<\varepsilon\quad\forall t\in\mathbb{R}$$

En particular trasladado a sucesiones:

$$|(f_M)_k-(S_M)_k|=\left|f(\frac{k}{2M+1})-S_Mf(\frac{k}{2M+1})\right|<\varepsilon\quad\forall k\in\mathbb{Z}$$

Si restamos ambas transformadas y aplicamos la desigualdad triangular a cada componente $n=0,\ldots,2M$ tenemos que:

$$|(\hat{f}_M^*)_n-(\hat{f}_M)_n|\leq\frac{1}{2M+1}\sum_{k=0}^{2M}|(f_M)_k-(S_M)_k|\leq\frac{1}{2M+1}(2M+1)\varepsilon=\varepsilon$$

Y finalmente, aplicando supremos tenemos el resultado. □

Nótese que cuanto más grande sea M, además de obtener más coeficientes, se calcularán con mejor aproximación; ya que cuantos más puntos se toman, más completa es la serie truncada y por tanto más certera es la aproximación $S_Mf(t)\approx f(t)$. Además, nótese que para obtener $2M+1$ coeficientes de Fourier (aproximados) necesitamos $2M+1$ muestras equiespaciadas de la función.

No obstante, ¿podemos extender el resultado para otras familias de funciones? En términos prácticos, es difícil encontrarse con casos totalmente continuos. De hecho, solo hay que pensar en una simple melodía; cada cambio de nota crea una discontinuidad. O mismamente si la extensión periódica de la señal que queramos procesar no sea continua.

Por tanto, ¿podemos garantizar igualmente una convergencia para funciones con salto? Veamos que aunque la convergencia de las sumas parciales deje de ser uniforme para estas funciones, los $2M+1$ coeficientes de Fourier obtenidos con la transformada discreta se acercan igualmente a los $2M+1$ coeficientes de Fourier reales.

Teorema 1.8. *Sea $f : \mathbb{R} \longrightarrow \mathbb{C}$, 1-periódica y de clase $C^{(1}$ a trozos salvo en 0 (y por consiguiente en $k \in \mathbb{Z}$). Entonces, $\forall \varepsilon > 0$ existe un $M \in \mathbb{N}$ lo suficientemente grande tal que:*

$$\|\hat{f}_M^* - \hat{f}_M\|_\infty < \varepsilon$$

Demostración. Sea $\varepsilon > 0$ y $\delta := \frac{\varepsilon}{6 \cdot c}$, donde c la constante dependiente del salto del corolario (A.9).
Notemos que por el teorema (A.4), existe un $N_1 \in \mathbb{N}$ lo suficientemente grande tal que:

$$\sup\{|f(t) - S_{N_1} f(t)| : t \notin (-\delta, \delta)\} < \frac{\varepsilon}{2} \tag{1.6}$$

Por otro lado, por el corolario (A.9) también tenemos que existe un $N_2 \in \mathbb{N}$ lo suficientemente grande tal que:

$$\sup\{|f(t) - S_{N_2} f(t)| : t \in (-\delta, \delta)\} \leq c \tag{1.7}$$

Tomemos ahora $M := \max\{N_1, N_2, N_3\}$, donde $N_3 \in \mathbb{N}$ es el menor natural cumpliendo que $\frac{1}{2N_3+1} \leq \delta$.

Restando ambas transformadas y aplicando la desigualdad triangular a cada componente $n = 0, \ldots, 2M$ tenemos que:

$$|(\hat{f}_M^*)_n - (\hat{f}_M)_n| \leq \frac{1}{2M+1} \sum_{k=0}^{2M} |(f_M)_k - (S_M)_k| = \frac{1}{2M+1} \sum_{k=0}^{2M} \left| f(\frac{k}{2M+1}) - S_M f(\frac{k}{2M+1}) \right|$$

Nótese que por la periodicidad de f y $S_M f$ podemos cambiar los índices del sumatorio:

$$\frac{1}{2M+1} \sum_{k=0}^{2M} \left| f(\frac{k}{2M+1}) - S_M f(\frac{k}{2M+1}) \right| = \frac{1}{2M+1} \sum_{k=-M}^{M} \left| f(\frac{k}{2M+1}) - S_M f(\frac{k}{2M+1}) \right|$$

$$= \frac{1}{2M+1} \left[\sum_{\frac{|k|}{2M+1} < \delta} \left| f(\frac{k}{2M+1}) - S_M f(\frac{k}{2M+1}) \right| + \sum_{\frac{|k|}{2M+1} \geq \delta} \left| f(\frac{k}{2M+1}) - S_M f(\frac{k}{2M+1}) \right| \right]$$

Y usando las cotas anteriores (1.6) y (1.7) tenemos que:

$$\leq \frac{1}{2M+1} (2(2M+1)\delta + 1) \cdot c + \frac{1}{2M+1} \cdot (2M+1) \cdot \frac{\varepsilon}{2} \leq 3 \cdot \delta \cdot c + \frac{\varepsilon}{2} = \varepsilon$$

Finalmente, tomando supremos se tiene el resultado. □

Nótese que el resultado general para cualquier función con n discontinuidades en $t_1 \ldots t_n$ tendría una demostración análoga a la anterior tomando $\delta_1, \ldots, \delta_n$ con sus respectivas constantes $c_1, \ldots, c_n$ y eligiendo el máximo de los M_{δ_i} para que se cumpliera la cota de la proposición (A.9) en todos los entornos $(t_i - \delta_i, t_i + \delta)$. La acotación fuera de ellos estaría asegurada igualmente por el teorema (A.4).

1.6. Muestreo par. Suma parcial desviada

No obstante, hay un pequeño detalle que ha pasado desapercibido: hemos desarrollado todo el estudio con un muestreo impar de f. Notemos que hemos hallado un método para obtener buenas aproximaciones de los $2M+1$ coeficientes de Fourier de la suma parcial $S_M f(t) = \sum_{n=-M}^{M} \hat{f}(n) e^{2\pi i n t}$ con un muestreo equiespaciado de f de tamaño $2M+1$.

Como en la práctica los valores de ese muestreo es la única información conocida y, no tendremos control sobre su paridad, tiene sentido adaptar el problema para dar cabida a los muestreos pares. De hecho en un futuro tendrá relativa importancia, ya que el algoritmo que desarrollaremos en el siguiente capítulo alcanza su mayor eficiencia cuando el muestreo es de tamaño $N = 2^m$. Las sumas parciales desviadas solucionarán esta situación.

Definición 3. Sea $M \in \mathbb{N}$ y $f : \mathbb{R} \longrightarrow \mathbb{C}$, una función 1-periódica, continua y de clase $C^{(1}$ a trozos. Llamaremos suma parcial desviada a:

$$S_M^* f(t) = \sum_{n=-(M-1)}^{M} \hat{f}(n) e^{2\pi i n t}, \quad t \in \mathbb{R}$$

Notemos que, por el lema de Riemann-Lebesgue (A.1), la suma parcial desviada se "acerca" a la suma parcial que ya habíamos definido:

$$|S_M f(t) - S_M^* f(t)| = |\hat{f}(-M)| \xrightarrow{M\to\infty} 0$$

Análogamente, con el propio sistema de sumas parciales desviadas

$$\sum_{n=-(M-1)}^{M} \hat{f}(n) e^{2\pi i n \frac{k}{2M}} = S_M f(\frac{k}{2M}), \quad k = 0, \ldots, 2M \tag{1.8}$$

y la aproximación $S_M^* f(t) \approx f(t)\ \forall t \in [0,1]$, también podemos plantear el sistema análogo a (1.3), ahora con un número par de ecuaciones:

$$\sum_{n=-(M-1)}^{M} \hat{f}^*(n) e^{2\pi i n \frac{k}{2M}} = f(\frac{k}{2M}), \quad k = 0, \ldots, 2M-1 \tag{1.9}$$

Del mismo modo, ambos sistemas encajan en el contexto de la transformada discreta:

Proposición 1.9. *Sean S_M la sucesión $2M$-periódica generada por $\left[S_M^* f(\frac{0}{2M}), \ldots, S_M^* f(\frac{2M-1}{2M})\right]$ y $\hat{f}_M$ la sucesión $2M$-periódica generada por $\left[\hat{f}(-(M-1)), \ldots, \hat{f}(M)\right]$, las sucesiones relativas al sistema (1.8). Entonces,*

$$\hat{f}_M \text{ es la transformada de Fourier discreta de } S_M$$

Proposición 1.10. *Sean f_M la sucesión $2M$-periódica generada por $\left[f\left(\frac{0}{2M}\right), \ldots, f\left(\frac{2M-1}{2M}\right)\right]$ y $\hat{f}_M^*$ la sucesión $2M$-periódica generada por $\left[\hat{f}^*(-(M-1)), \ldots, \hat{f}^*(M)\right]$, las sucesiones relativas al sistema (1.9). Entonces,*

$$\hat{f}_M^* \text{ es la transformada de Fourier discreta de } f_M$$

Notemos que los resultados de convergencia obtenidos en la sección (1.5) también se mantendrán en el problema definido con las sumas parciales desviadas.

Con esta formulación el vector solución del sistema (1.5) será:

$$[c_0, \ldots, c_M, c_{M+1}, \ldots, c_{2M-1}] = [c_0, \ldots, c_M, c_{-M+1}, \ldots, c_{-1}]$$

Observación. Cabe mencionar que podríamos haber definido la suma parcial desviada como

$$S_M^* f(t) = \sum_{n=-M}^{M-1} \hat{f}(n) e^{2\pi i n t}, \quad t \in \mathbb{R}$$

y haber realizado el mismo desarrollo. No obstante, notemos que ambos sistemas darían el mismo resultado, ya que ambos representan el sistema de la misma transformada de Fourier discreta:

$$\begin{aligned} f(\frac{k}{2M}) &= \sum_{n=-(M-1)}^{M} a_n e^{2\pi i n \frac{k}{2M}} = \sum_{n=-(M-1)}^{M-1} a_n e^{2\pi i n \frac{k}{2M}} + a_M e^{2\pi i M \frac{k}{2M}} \\ &= \sum_{n=-(M-1)}^{M-1} a_n e^{2\pi i n \frac{k}{2M}} + a_{-M} e^{2\pi i (-M) \frac{k}{2M}} = \sum_{n=-M}^{M-1} a_n e^{2\pi i n \frac{k}{2M}}, \quad k = 0, \ldots, 2M-1 \end{aligned}$$

Concordando con la biyección obtenida en el teorema (1.4). □

1.7. Funciones de cualquier periodo

Durante el desarrollo de todo el capítulo hemos planteado el problema con funciones de periodo 1. Veamos como podemos generalizarlo a funciones de cualquier periodo.

Sea $f : \mathbb{R} \longrightarrow \mathbb{C}$, L-periódica, continua y de clase $C^{(1}$ a trozos. Del mismo modo, por el teorema (A.2) tenemos que

$$f(t) = \sum_{n=-\infty}^{\infty} \hat{f}(n) e^{2\pi i \frac{n}{L} t}, \text{ uniformemente en } [0,L]$$

Y con la misma idea, tomando un muestreo equiespaciado de f en el intervalo $[0,L]$ y las aproximaciones con sumas parciales obtenemos el sistema:

$$\sum_{n=-M}^{M} \hat{f}^*(n) e^{2\pi i \frac{n}{L} \frac{kL}{2M+1}} = \sum_{n=-M}^{M} \hat{f}^*(n) e^{2\pi i n \frac{k}{2M+1}} = f(\frac{kL}{2M+1}), \quad k = 0, \ldots, 2M$$

Si nos fijamos en el sistema generado por la segunda igualdad, es exactamente el mismo obtenido con las funciones de periodo 1. Únicamente aparece el periodo en el muestreo de la función, que no afecta a la resolución del sistema. Por tanto, la matriz del sistema a resolver será siempre la misma para cualquier periodo. De hecho, para el vector del muestreo únicamente importará que sea equiespaciado en el intervalo.

Capítulo 2

La transformada rápida de Fourier (FFT)

2.1. Introducción

En este capítulo presentaremos el conocido algoritmo de la *Transformada Rápida de Fourier* (FFT), el cual calcula los coeficientes de la transformada discreta de Fourier (Def. 2):

$$\hat{f}_n^* = \frac{1}{N}\sum_{k=0}^{N-1} f_k \omega^{kn}, \quad 0 \le n < N, \quad \omega := e^{-2\pi i/N}$$

Para el estudio del algoritmo, expresaremos el sistema con notación matricial:

$$\hat{f}^* = \mathbf{F}_N f \tag{2.1}$$

En esta ecuación $f = [f_0, \ldots, f_{N-1}]^T \in \mathbb{C}^N$ reúne el conjunto ordenado de los valores de un muestreo de una cierta función f. Dicho vector será parte de la entrada del algoritmo FFT, mientras que la salida será el vector $\hat{f}^* = [\hat{f}_0^*, \ldots, \hat{f}_{N-1}^*]^T \in \mathbb{C}^N$, que reúne los coeficientes de la Transformada Discreta de Fourier de f. Recordemos que la matriz $\mathbf{F}_N$ está definida de la siguiente forma:

$$\mathbf{F}_N = \frac{1}{N}\left[\omega^{ij}\right]_{0\le i,j<N} = \frac{1}{N}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2N-2} & \cdots & \omega^{(N-1)(N-1)} \end{bmatrix}$$

Para analizar el comportamiento asintótico del algoritmo, tanto en tiempo como en espacio, supondremos simplemente que cualquier operación de suma o multiplicación de números complejos se realiza en una unidad de tiempo. Del mismo modo, supondremos que el almacenamiento de un número complejo en la memoria del ordenador requiere una unidad de espacio. Con estas decisiones, la elección natural para el tamaño de un ejemplar de entrada de este problema será N, el número de componentes del vector de entrada f.

El cálculo del tiempo usado por el algoritmo se reducirá a *"contar"* el número de operaciones aritméticas complejas, expresándolo como una función de N. Más precisamente, puesto que sólo nos interesará el comportamiento asintótico, expresaremos estos costes como el orden $O(t(N))$ de una cierta función $t(n)$.

Definición 4. Dada una función $t : \mathbb{N} \longrightarrow \mathbb{R}^+$, el orden de $t(N)$ es el conjunto:

$$O(t(N)) = \{f : \mathbb{N} \longrightarrow \mathbb{R}^+ \,|\, \exists c \in \mathbb{R}^+, n_0 \in \mathbb{N} \text{ tales que } \forall n \ge n_0 \quad f(n) \le c \cdot t(n)\}$$

Obviamente, el algoritmo que realiza directamente el producto matricial dado en la ecuación (2.1) tiene un coste del orden de N^2 operaciones complejas, además de utilizar $O(N^2)$ espacio si tenemos precalculada y almacenada la matriz $\mathbf{F}_N$ en memoria. No obstante, atendiendo a la estructura polinómica en función de ω de las componentes de c

$$\hat{f}_n^* = \frac{1}{N}\left[f_0 + f_1\omega^n + f_2(\omega^n)^2 + \cdots + f_{N-1}(\omega^n)^{N-1}\right]$$

podríamos recurrir al conocido algoritmo de Horner para la evaluación de polinomios, calculando las potencias de ω dentro del algoritmo. Con esto, evitaríamos el almacenamiento en memoria de la matriz $\mathbf{F}_N$, reduciendo el coste en espacio a $O(N)$, aunque manteniendo el coste en tiempo en $O(N^2)$.

A pesar del coste cuadrático, este fue el método generalmente aceptado para el cálculo de la transformada de Fourier discreta hasta la aparición del algoritmo presentado por James W. Cooley y John W. Tukey en 1965. Dicho algoritmo presentó una nueva y eficiente manera de calcular la transformada discreta de Fourier reduciendo el número de operaciones complejas al orden de

$$N \cdot \{(P_1 - 1) + (P_2 - 1) + \cdots + (P_m - 1)\}$$

donde $N = P_1 \cdots P_m$ es una descomposición de N, no necesariamente en factores primos. De hecho, el ahorro más dramático se produce en el caso $N = 2^m$, rebajando el coste hasta el orden de $N\log_2 N$ operaciones complejas, frente a las N^2 de Horner. Esta reducción supuso un punto de inflexión en el uso práctico de la transformada de Fourier discreta. En las siguientes secciones fundamentaremos y presentaremos las piezas que conforman dicho algoritmo.

2.2. La identidad cremallera $\mathbf{F}_{P\cdot Q}$

En el capítulo anterior hemos anunciado que la estructura de la matriz $\mathbf{F}_N$ y el hecho de que sus componentes sean las raíces N-ésimas de la unidad juegan un papel fundamental en el algoritmo de la Transformada Rápida de Fourier (FFT). En esta sección concretaremos esta afirmación.

Más precisamente, dada una factorización $N = P \cdot Q$, obtendremos la *"identidad cremallera"* (*zipper identity*) de $\mathbf{F}_N$. Esta identidad permitirá expresar dicha matriz como producto de tres: una matriz formada por bloques diagonales de tamaño P, una matriz diagonal por bloques formada por Q bloques idénticos a $\mathbf{F}_P$ y, una matriz de permutación. Esencialmente, la identidad vendrá desencadenada como resultado de hacer uso de cierta permutación $\sigma_{P,Q} \in \Sigma_N$, del grupo simétrico de N elementos, y de las propiedades de grupo de las raíces N-ésimas de la unidad.

Para expresar esta identidad de forma compacta, y generalizarla cuando la factorización de N conste de más factores, comenzaremos introduciendo la siguiente notación exponencial para matrices.

Definición 5. Dada una matriz cuadrada $A \in \mathbb{C}^{N\times N}$, para todo entero $n > 0$ definimos la matriz $A^{(n)}$ del siguiente modo, donde 0 denota las matrices de tamaño adecuado cuyos elementos son todos nulos:

$$A^{(1)} := A \quad ; \quad A^{(n+1)} := \left[\begin{array}{c|c} A^{(n)} & 0 \\ \hline 0 & A \end{array}\right]$$

Es decir:

$$A^{(1)} := A, \quad A^{(2)} := \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix}, \quad A^{(3)} := \begin{bmatrix} A & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & A \end{bmatrix}, \quad \ldots$$

Para cualesquiera $p, q \in \mathbb{N}$ y matrices $A, B \in \mathbb{C}^{N\times N}$, es inmediato comprobar las siguientes propiedades:

1. $\left[A^{(p)}\right]^{(q)} = A^{(pq)}$
2. $[AB]^{(p)} = A^{(p)}B^{(p)}$
3. $[\alpha A]^{(p)} = \alpha A^{(p)}, \quad \alpha \in \mathbb{C}$

En el siguiente lema resaltaremos la primera de las piezas fundamentales que forman la base del algoritmo.

Lema 2.1. *Sea $\omega := e^{-2\pi i/N}$ raíz N-ésima de la unidad. Entonces, para cualquier factorización $N = P \cdot Q$, $\tau := \omega^Q$ es raíz P-ésima de la unidad.*

Por otro lado, notemos que por el Teorema de la división entera, para todo índice n tal que $0 \leq n < N = P \cdot Q$, existirán únicos p_1, q_1 cumpliendo que:

$$n = p_1 + q_1 P, \quad 0 \leq p_1 < P \quad 0 \leq q_1 < Q$$

Del mismo modo, intercambiando los papeles de P y Q, también existirán únicos q_2, p_2 tal que:

$$n = q_2 + p_2 Q, \quad 0 \leq q_2 < Q \quad 0 \leq p_2 < P$$

Con estas apreciaciones, podemos definir la segunda de las piezas del algoritmo, la permutación $\sigma_{P,Q}$:

Definición 6. Sea $N = P \cdot Q$. Denotamos como $\sigma_{P,Q} \in \Sigma_N$ a la permutación dada por:

$$\sigma_{P,Q}(p + qP) = q + pQ, \quad p \in \{0, \ldots, P-1\} \quad q \in \{0, \ldots, Q-1\}$$

En particular, usaremos esta permutación actuando sobre vectores y sobre matrices:

Definición 7. Sea $\sigma \in \Sigma_N$.

- Dado un vector $a = (a_i)_{0 \leq i < N} \in \mathbb{C}^N$, denotamos como $\sigma(a)$ al vector resultante al aplicar σ sobre sus componentes:
$$\sigma(a) := \left[a_{\sigma(i)}\right]_{0 \leq i < N}$$
- Dada una matriz $A = [A_{ij}]_{0 \leq i,j < N} \in \mathbb{C}^{N \times N}$, denotamos como $\sigma(A)$ a la matriz resultante al aplicar σ sobre sus columnas:
$$\sigma(A) := \left[A_{i\sigma(j)}\right]_{0 \leq i,j < N}$$

Definición 8. Dada una permutación $\sigma \in \Sigma_N$, denotamos como $S_\sigma \in \mathbb{C}^{N \times N}$ a la matriz de permutación asociada a σ:

$$S_\sigma = \begin{bmatrix} \overline{\quad e_{\sigma(0)} \quad} \\ \overline{\quad \vdots \quad} \\ \overline{\quad e_{\sigma(N-1)} \quad} \end{bmatrix}, \qquad e_i = [0, \ldots, \overset{i)}{1}, \ldots, 0] \in \mathbb{C}^N$$

Las matrices de permutación cumplen las siguientes propiedades:

Proposición 2.2. *Sea $a \in \mathbb{C}^N$ y $A \in \mathbb{C}^{N \times N}$. Dada una permutación $\sigma \in \Sigma^N$ se tiene que:*

- $\sigma(a) = S_\sigma a$
- $\sigma(A) = A S_\sigma^T$
- $S_\sigma^T = S_\sigma^{-1}$

Y con ellas es inmediato probar el siguiente lema:

Lema 2.3. *Sea $x \in \mathbb{C}^N$, $A \in \mathbb{C}^{N \times N}$ y $\sigma \in \Sigma_N$. Entonces:*

$$Ax = \sigma(A)\sigma(x)$$

Este hecho nos permitirá reordenar la matriz $\mathbf{F}_N$ y el vector f sin modificar el resultado del producto $\mathbf{F}_N f$:

$$\mathbf{F}_{P\cdot Q} f = \sigma_{P,Q}(\mathbf{F}_{P\cdot Q})\sigma_{P,Q}(f)$$

Por una parte, la aplicación de $\sigma_{P,Q}$ sobre un vector f queda determinada por su matriz de permutación $\mathbf{S}_{\sigma_{P,Q}}$. Como en todo el capítulo haremos uso de la misma permutación, a partir de ahora denotaremos por $\mathbf{S}_{P,Q}$ a la matriz de permutación $\mathbf{S}_{\sigma_{P,Q}}$. Así tenemos que:

$$\sigma_{P,Q}(f) = \mathbf{S}_{P,Q} f$$

Por la otra parte, tenemos a $\sigma_{P,Q}(\mathbf{F}_N)$, que está definido como:

$$\sigma_{P,Q}(\mathbf{F}_N) = \left[(\mathbf{F}_N)_{i\sigma_{P,Q}(j)}\right]_{0\le i,j<N} = \frac{1}{N}\left[\omega^{i\cdot\sigma_{P,Q}(j)}\right]_{0\le i,j<N}$$

No obstante, por motivos posteriores es interesante dividirla en submatrices de tamaño $P\times P$:

$$\sigma_{P,Q}(\mathbf{F}_N) = \frac{1}{P\cdot Q}\begin{bmatrix} B_{0,0} & B_{0,1} & \cdots & B_{0,Q-1} \\ B_{1,0} & B_{1,1} & \cdots & B_{1,Q-1} \\ \vdots & \vdots & & \vdots \\ B_{Q-1,0} & B_{Q-1,1} & \cdots & B_{Q-1,Q-1} \end{bmatrix}$$

Donde cada submatriz queda descrita del siguiente modo:

$$B_{r,s} = \left[\omega^{(i+r\cdot P)\cdot\sigma_{P,Q}(j+s\cdot P)}\right]_{0\le i,j<P} = \left[\omega^{(i+r\cdot P)\cdot(s+j\cdot Q)}\right]_{0\le i,j<P}$$

Proposición 2.4. *Sea $B_{r,s}$, $0\le r,s<Q$, la matriz definida anteriormente. Entonces:*

$$B_{r,s} = P\cdot W_{r,s}\cdot\mathbf{F}_P \quad \textit{siendo} \quad W_{r,s} = \omega^{rsP}\begin{bmatrix} 1 & & & \\ & \omega^s & & \\ & & \ddots & \\ & & & \omega^{(P-1)s} \end{bmatrix}$$

Demostración. Sean r,s índices fijos tal que $0\le r,s<Q$. Recurriendo a la propia definición y operando:

$$B_{r,s} = \left[\omega^{(i+rP)\cdot(s+jQ)}\right]_{0\le i,j<P} = \left[\omega^{is}\cdot\omega^{rsP}\cdot\omega^{ijQ}\cdot\omega^{jrPQ}\right]_{0\le i,j<P}$$

Denotando como $\tau := \omega^Q$, sabiendo que $\omega^{P\cdot Q} = \omega^N = 1$ por ser ω raíz N-ésima de la unidad, y que ω^{rsP} es constante, obtenemos:

$$= \omega^{rsP}\cdot\left[\omega^{is}\cdot\tau^{ij}\cdot(1)^{jr}\right]_{0\le i,j<P} = \omega^{rsP}\cdot\left[\omega^{is}\cdot\tau^{ij}\right]_{0\le i,j<P}$$

Como ω^{is} solo depende de i podemos sacarlo de la matriz como producto por una matriz diagonal:

$$= \omega^{rsP}\begin{bmatrix} 1 & & & \\ & \omega^s & & \\ & & \ddots & \\ & & & \omega^{(P-1)s} \end{bmatrix}\left[\tau^{ij}\right]_{0\le i,j<P} = W_{r,s}\cdot\left[\tau^{ij}\right]_{0\le i,j<P}$$

Y finalmente, como por el lema (2.1) τ es raíz P-ésima de la unidad se tiene que $\left[\tau^{ij}\right]_{0\le i,j<P} = P\cdot\mathbf{F}_P$, se obtiene el resultado buscado:

$$B_{r,s} = W_{r,s}\cdot P\cdot\mathbf{F}_P$$

□

De este modo, con cada bloque de la matriz $\sigma_{P,Q}(\mathbf{F}_N)$ factorizado y la ayuda de la notación dada en la definición (5), ya podemos expresar la identidad cremallera:

Teorema 2.5. (Identidad cremallera). *Dada una factorización $N = P \cdot Q$, se tiene la siguiente identidad:*

$$\mathbf{F}_{P\cdot Q} = \frac{1}{Q}\mathbf{Q}_{P,Q}\mathbf{F}_P^{(Q)}\mathbf{S}_{P,Q} \quad \textit{donde} \quad \mathbf{Q}_{P,Q} := \begin{bmatrix} W_{0,0} & W_{0,1} & \cdots & W_{0,Q-1} \\ W_{1,0} & W_{1,1} & \cdots & W_{1,Q-1} \\ \vdots & \vdots & & \vdots \\ W_{Q-1,0} & W_{Q-1,1} & \cdots & W_{Q-1,Q-1} \end{bmatrix}$$

Demostración. Con la factorización para cada $B_{r,s}$ y la notación exponencial es inmediato ver que:

$$\sigma_{P,Q}(\mathbf{F}_{P\cdot Q}) = \frac{1}{P\cdot Q}\mathbf{Q}_{P,Q}\cdot P\cdot \mathbf{F}_P^{(Q)} = \frac{1}{Q}\mathbf{Q}_{P,Q}\mathbf{F}_P^{(Q)}$$

Así mismo, conocemos que $\sigma(f)$ está descrito por su matriz de permutación:

$$\sigma_{P,Q}(f) = \mathbf{S}_{P,Q}$$

Por lo que uniendo ambas expresiones junto con el lema (2.3) tenemos que:

$$\mathbf{F}_{P\cdot Q}f = \sigma_{P,Q}(\mathbf{F}_{P\cdot Q})\sigma_{P,Q}(f) = \frac{1}{Q}\mathbf{Q}_{P,Q}\mathbf{F}_P^{(Q)}\mathbf{S}_{P,Q}f$$

Como esta igualdad es cierta para cualquier $f \in \mathbb{C}^N$, tenemos finalmente el resultado:

$$\mathbf{F}_{P\cdot Q} = \frac{1}{Q}\mathbf{Q}_{P,Q}\mathbf{F}_P^{(Q)}\mathbf{S}_{P,Q}$$

□

Ejemplo. Tomemos el caso $N = 6$, con los factores $P = 2$ y $Q = 3$. Como hemos podido ver, la permutación $\sigma_{P,Q}$ juega un papel crucial para hallar la identidad cremallera. Primeramente, calculemos explícitamente la permutación $\sigma_{2,3}$:

$$\sigma_{2,3}(p+2\cdot q) = (q+3\cdot p)$$

$$\begin{aligned}
0 &= 0+2\cdot 0 \quad (p,q) = (0,0) \quad \longrightarrow \quad 0+3\cdot 0 = 0 \\
1 &= 1+2\cdot 0 \quad (p,q) = (1,0) \quad \longrightarrow \quad 0+3\cdot 1 = 3 \\
2 &= 0+2\cdot 1 \quad (p,q) = (0,1) \quad \longrightarrow \quad 1+3\cdot 0 = 1 \\
3 &= 1+2\cdot 1 \quad (p,q) = (1,1) \quad \longrightarrow \quad 1+3\cdot 1 = 4 \\
4 &= 0+2\cdot 2 \quad (p,q) = (0,2) \quad \longrightarrow \quad 2+3\cdot 0 = 2 \\
5 &= 1+2\cdot 2 \quad (p,q) = (1,2) \quad \longrightarrow \quad 2+3\cdot 1 = 5
\end{aligned}$$

Con la permutación descrita, podemos ver fácilmente la acción de $\sigma_{2,3}$ sobre el vector f:

$$\sigma_{2,3}(f) = \left[f_{\sigma(i)}\right]_{0\le i<6} = [f_0, f_3, f_1, f_4, f_2, f_5]^T = \mathbf{S}_{2,3}f$$

Así mismo, también podemos expresar la acción de la permutación sobre la matriz $\mathbf{F}_{2\cdot 3}$:

$$\sigma_{2,3}(\mathbf{F}_{2\cdot 3}) = \left[(\mathbf{F}_6)_{i\sigma_{(2,3)}(j)}\right]_{0\le i,j<6} = \frac{1}{6}\left[\begin{array}{cc|cc|cc} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega^3 & \omega & \omega^4 & \omega^2 & \omega^5 \\ \hline 1 & \omega^6 & \omega^2 & \omega^8 & \omega^4 & \omega^{10} \\ 1 & \omega^9 & \omega^3 & \omega^{12} & \omega^6 & \omega^{15} \\ \hline 1 & \omega^{12} & \omega^4 & \omega^{16} & \omega^8 & \omega^{20} \\ 1 & \omega^{15} & \omega^5 & \omega^{20} & \omega^{10} & \omega^{25} \end{array}\right]$$

Y teniendo en cuenta que $\tau := \omega^3$ es una raíz 2-ésima de la unidad y que genera un subgrupo de orden 2, es fácil ver la factorización:

$$\sigma_{2,3}(\mathbf{F}_{2\cdot 3}) = \frac{1}{3}\left[\begin{array}{cc|cc|cc} 1 & & 1 & & 1 & \\ & 1 & & \omega & & \omega^2 \\ \hline 1 & & \omega^2 & & \omega^4 & \\ & 1 & & \omega^3 & & \omega^6 \\ \hline 1 & & \omega^4 & & \omega^8 & \\ & 1 & & \omega^5 & & \omega^{10} \end{array}\right] \left[\begin{array}{c|c|c} \mathbf{F}_2 & \mathbf{0}_2 & \mathbf{0}_2 \\ \hline \mathbf{0}_2 & \mathbf{F}_2 & \mathbf{0}_2 \\ \hline \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{F}_2 \end{array}\right] = \frac{1}{3}\cdot \mathbf{Q}_{2,3}\cdot \mathbf{F}_2^{(3)}$$

Finalmente uniendo ambas expresiones se tiene la identidad obtenida en la sección:

$$\mathbf{F}_{2\cdot 3} f = \sigma_{2,3}(\mathbf{F}_{2\cdot 3})\sigma_{2,3}(f) = \frac{1}{3}\mathbf{Q}_{2,3}\mathbf{F}_2^{(3)}\mathbf{S}_{2,3} f$$

La elección de esta permutación no es casual, en este caso particular se puede ver bien que $\sigma_{2,3}$ reordena la matriz por bloques de columnas cuyos índices pertenecen a la misma clase de equivalencia módulo 3. Esto es lo que permite que aparezca $\tau := \omega^3$ y que obtengamos la sencilla factorización de la proposición (2.4):

$$B_{r,s} = \left[\omega^{(i+2\cdot r)\cdot(s+3\cdot j)}\right]_{0\le i,j<2} = \begin{bmatrix} \omega^{2rs} & \omega^{2rs} \\ \omega^s\cdot\omega^{2rs} & \omega^s\cdot\omega^{2rs}\cdot\omega^3 \end{bmatrix} = \omega^{2rs}\begin{bmatrix} 1 & 0 \\ 0 & \omega^s \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & \tau \end{bmatrix} = W_{r,s}\cdot 2\cdot \mathbf{F}_2$$

□

2.3. Factorización de $\mathbf{F}_N$ en el caso $N = 2^m$

Si se dispone de una factorización de $N = P_1\cdots P_m$ con más de dos factores, el siguiente paso natural sería aplicar recursivamente la identidad cremallera hasta llegar a un caso trivial. A continuación abordaremos esta tarea en el caso más simple, cuando todos los factores sean 2.

Comenzaremos particularizando la identidad cremallera obtenida en la sección anterior al caso $N = 2\cdot M$, $M \ge 1$. Como en esta sección tomaremos $P = M$ y $Q = 2$, escribiremos $\mathbf{Q}_{2M} := \mathbf{Q}_{M,2}$ y $\mathbf{S}_{2M} := \mathbf{S}_{M,2}$ para simplificar la notación. Así, la identidad cremallera se reescribe como:

$$\mathbf{F}_{2M} = \frac{1}{2}\mathbf{Q}_{2M}\left[\begin{array}{c|c} \mathbf{F}_M & \mathbf{0}_M \\ \hline \mathbf{0}_M & \mathbf{F}_M \end{array}\right]\mathbf{S}_{2M} = \frac{1}{2}\mathbf{Q}_{2M}\mathbf{F}_M^{(2)}\mathbf{S}_{2M} \tag{2.2}$$

Por otro lado, teniendo en cuenta que $\omega^M = -1$, ya que $(\omega^M)^2 = \omega^N = 1$ y $\omega \ne 1$, la matriz $\mathbf{Q}_{2M}$ también se reescribe de la siguiente forma más simple:

$$\mathbf{Q}_{2M} := \left[\begin{array}{ccc|ccc} 1 & & & 1 & & \\ & \ddots & & & \ddots & \\ & & 1 & & & \omega^{M-1} \\ \hline 1 & & & -1 & & \\ & \ddots & & & \ddots & \\ & & 1 & & & -\omega^{M-1} \end{array}\right] \tag{2.3}$$

Nótese que la identidad (2.2) relaciona $\mathbf{F}_N$ con $\mathbf{F}_{N/2}$. Por lo que, si nuestro N inicial se factoriza como $N = 2^m$, podremos aplicar esta relación que nos aporta la identidad cremallera recursivamente hasta llegar al caso trivial $\mathbf{F}_1 = [1]$. Dicha recursión es la que nos lleva a la factorización de $\mathbf{F}_{2^m}$.

Teorema 2.6. *Sea $N = 2^m$, $m = 1, 2, \ldots$. Entonces:*

$$\mathbf{F}_{2^m} = \frac{1}{2^m}\mathbf{Q}_{2^m}\mathbf{Q}^{(2)}_{2^{m-1}}\mathbf{Q}^{(4)}_{2^{m-2}}\cdots\mathbf{Q}^{(2^{m-1})}_{2}\mathbf{B}_{2^m}, \quad \textit{donde} \quad \mathbf{B}_{2^m} := \mathbf{S}^{(2^{m-1})}_{2}\mathbf{S}^{(2^{m-2})}_{4}\ldots\mathbf{S}^{(2)}_{2^{m-1}}\mathbf{S}_{2^m}$$

Demostración. Aplicando la identidad cremallera m veces con $Q = 2$ y $P = 2^k$, $k = m-1, \ldots, 0$, y usando las propiedades de la notación exponencial (5) tenemos que:

$$\begin{aligned}
\mathbf{F}_{2^m} &= \frac{1}{2}\mathbf{Q}_{2^m}\mathbf{F}^{(2)}_{2^{m-1}}\mathbf{S}_{2^m} \\
&= \frac{1}{2}\mathbf{Q}_{2^m}\left[\frac{1}{2}\mathbf{Q}_{2^{m-1}}\mathbf{F}^{(2)}_{2^{m-2}}\mathbf{S}_{2^{m-1}}\right]^{(2)}\mathbf{S}_{2^m} \\
&= \frac{1}{4}\mathbf{Q}_{2^m}\mathbf{Q}^{(2)}_{2^{m-1}}\mathbf{F}^{(4)}_{2^{m-2}}\mathbf{S}^{(2)}_{2^{m-1}}\mathbf{S}_{2^m} \\
&= \cdots = \frac{1}{2^m}\mathbf{Q}_{2^m}\mathbf{Q}^{(2)}_{2^{m-1}}\mathbf{Q}^{(4)}_{2^{m-2}}\ldots\mathbf{F}^{(2^m)}_{1}\mathbf{S}^{(2^{m-1})}_{2}\mathbf{S}^{(2^{m-2})}_{4}\ldots\mathbf{S}^{(2)}_{2^{m-1}}\mathbf{S}_{2^m}
\end{aligned}$$

Finalmente, teniendo en cuenta que

$$\mathbf{F}^{(2^m)}_{1} = [1]^{(2^m)} = \mathbf{I}_{2^m}$$

y reuniendo todas las matrices de permutación en

$$\mathbf{B}_{2^m} := \mathbf{S}^{(2^{m-1})}_{2}\mathbf{S}^{(2^{m-2})}_{4}\ldots\mathbf{S}^{(2)}_{2^{m-1}}\mathbf{S}_{2^m}$$

Obtenemos el resultado:

$$\mathbf{F}_{2^m} = \frac{1}{2^m}\mathbf{Q}_{2^m}\mathbf{Q}^{(2)}_{2^{m-1}}\ldots\mathbf{Q}^{(2^{m-1})}_{2}\mathbf{B}_{2^m}$$

□

Notemos que las matrices $\mathbf{S}^{(2^k)}_{2^{m-k}}$, $k = 1, \ldots, m-1$ son matrices de permutación. Por ello, como el producto de matrices de permutación es también una matriz de permutación, B_{2^m} será igualmente una matriz de permutación. Que hayamos definido la matriz $\mathbf{B}_{2^m}$ no es casual; en la siguiente sección describiremos la permutación que representa.

2.3.1. Acción de $\mathbf{B}_{2^m}$

Como hemos definido en la sección anterior, la matriz de permutación $\mathbf{B}_{2^m}$ está formada por un producto de matrices de permutación. Por ello, el guión que seguiremos para conocer su acción sobre un vector será describir las acciones de cada una de las matrices de permutación que la conforman, y hallar la acción global componiéndolas ordenadamente.

Antes de comenzar con ello, notemos que las matrices involucradas son del tipo $\mathbf{S}^{(2^k)}_{2^{m-k}}$, donde donde k es un entero tal que $0 \leq k < m$. Es decir, matrices diagonales por bloques formadas por 2^k matrices $\mathbf{S}_{2^{m-k}}$ de tamaño $2^{m-k} \times 2^{m-k}$. Recordemos que estas matrices $\mathbf{S}_{2^{m-k}} := \mathbf{S}_{2^{m-k-1},2}$ vienen expresadas por la permutación:

$$\sigma_{2^{m-k-1},2}(p + q\cdot 2^{m-k-1}) = q + p\cdot 2, \quad 0 \leq p < 2^{m-k-1} \quad 0 \leq q < 2$$

A su vez, nótese que si dividimos el vector en 2^k bloques de tamaño 2^{m-k}, las matrices $\mathbf{S}_{2^{m-k}}$ que conforman $\mathbf{S}^{(2^k)}_{2^{m-k}}$ actuarán idénticamente sobre cada uno de estos bloques. Con ello, si indexamos los índices en 2^k bloques (es decir $n = n' + r\cdot 2^{m-k}$, $0 \leq r < 2^k$), es sencillo ver que podemos expresar la permutación asociada a la matriz $\mathbf{S}^{(2^k)}_{2^{m-k}}$ en términos de $\sigma_{2^{m-k-1},2}$:

$$\sigma^{(2^k)}_{2^{m-k-1},2}(n) = \sigma^{(2^k)}_{2^{m-k-1},2}(p + q\cdot 2^{m-k-1} + r\cdot 2^{m-k}) = \sigma_{2^{m-k-1},2}(p + q\cdot 2^{m-k-1}) + r\cdot 2^{m-k},$$
$$0 \leq p < 2^{m-k-1} \quad 0 \leq r < 2^k \quad 0 \leq q < 2$$

Proposición 2.7. *Sea $x \in \mathbb{C}^N$ y $\mathbf{S}^{(2^k)}_{2^{m-k}}$ la matriz de permutación asociada a $\sigma^{(2^k)}_{2^{m-k-1},2}$. Entonces:*

$$(\mathbf{S}^{(2^k)}_{2^{m-k}}x)_{p+q\cdot 2^{m-k-1}+r\cdot 2^{m-k}} = x_{q+p\cdot 2+r\cdot 2^{m-k}}, \quad 0 \le p < 2^{m-k-1} \quad q = 0,1 \quad 0 \le r < 2^k$$

Es decir, la acción de la permutación $\sigma^{(2^k)}_{2^{m-k-1},2}$ sobre las componentes del vector x mueve la componente de índice $q + p\cdot 2 + r\cdot 2^{m-k}$ a la componente de índice $p + q\cdot 2^{m-k-1} + r\cdot 2^{m-k}$. Denotaremos dicho movimiento con la siguiente notación:

$$s_k\left[q + p\cdot 2 + r\cdot 2^{m-k}\right] = p + q\cdot 2^{m-k-1} + r\cdot 2^{m-k}$$

Demostración. Recurriendo a la definición (7) de la acción de una permutación sobre un vector, a la primera de las propiedades de la proposición (2.2) y a la descripción de la permutación $\sigma^{(2^k)}_{2^{m-k-1},2}$ dada, se tiene el resultado:

$$\begin{aligned}(\mathbf{S}^{(2^k)}_{2^{m-k}}x)_{p+q\cdot 2^{m-k-1}+r\cdot 2^{m-k}} &= x_{\sigma^{2^k}_{2^{m-1},2}(p+q\cdot 2^{m-k-1}+r\cdot 2^{m-k})} \\ &= x_{\sigma_{2^{m-1},2}(p+q\cdot 2^{m-k-1})+r\cdot 2^{m-k}} = x_{q+p\cdot 2+r\cdot 2^{m-k}}\end{aligned}$$

□

Observación. Nótese que si $k = 0$ se tiene que la permutación resultante, asociada a $\mathbf{S}_{2^m}$, es $\sigma_{2^{m-1},2}$:

$$\begin{gathered}\sigma^{(1)}_{2^{m-1},2}(p + q\cdot 2^{m-1} + r\cdot 2) = q + p\cdot 2 + r\cdot 2^m, \quad 0 \le p < 2^{m-1} \quad 0 \le q < 2 \quad 0 \le r < 1 \\ \sigma_{2^{m-1},2}(p + q\cdot 2^{m-1}) = q + r\cdot 2, \quad 0 \le p < 2^{m-1} \quad 0 \le q < 2\end{gathered}$$

Por lo que el movimiento asociado a $\mathbf{S}_{2^m}$ queda expresado como:

$$s_0\left[q + p\cdot 2\right] = p + q\cdot 2^{m-1}, \quad 0 \le p < 2^{m-1} \quad q = 0,1 \tag{2.4}$$

Por otro lado, si $k = m-1$ se tiene que la permutación resultante es la identidad:

$$\begin{gathered}\sigma^{(2^{m-1})}_{1,2}(p + q\cdot 1 + r\cdot 2) = q + p\cdot 2 + r\cdot 2, \quad 0 \le p < 1 \quad 0 \le q < 2 \quad 0 \le r < 2^{m-1} \\ \sigma^{(2^{m-1})}_{1,2}(q + r\cdot 2) = q + r\cdot 2, \quad 0 \le q < 2 \quad 0 \le r < 2^{m-1}\end{gathered}$$

Es decir, esta acción no mueve ninguna componente. En particular se tiene que:

$$\mathbf{S}^{(2^{m-1})}_2 = \mathbf{I}_{2^m} \tag{2.5}$$

□

Para llegar a nuestro objetivo y poder componer todas las matrices necesitaremos el siguiente resultado, que también es consecuencia del Teorema de la división entera:

Proposición 2.8. *(Expresión binaria). Para todo entero n tal que $0 \le n < N = 2^k$ existen únicos $\{b_i\}_{i=0}^{k-1}$ cumpliendo:*

$$n = b_0 + b_1\cdot 2 + b_2\cdot 2^2 + \cdots + b_{k-2}\cdot 2^{k-2} + b_{k-1}\cdot 2^{k-1} := (b_{k-1}b_{k-2}\cdots b_1b_0)_{2)}$$

Lema 2.9. *Sea $N = 2^m$ y k un entero tal que $0 \le k < m-1$. La acción de la permutación asociada a la matriz $\mathbf{S}^{(2^k)}_{2^{m-k}} \cdot \mathbf{S}^{(2^{k-1})}_{2^{m-k+1}} \cdots \mathbf{S}_{2^m}$ está descrita por el siguiente movimiento:*

$$r_k[(b_{m-1}\cdots b_0)_{2)}] = (b_{m-1}\cdots b_{k+1})_{2)} + (b_k\cdots b_0)_{2)}\cdot 2^{m-k-1}$$

Donde $n = (b_{m-1}\cdots b_0)_{2)}$ es la expresión binaria de cada índice $0 \le n < 2^m$.

Demostración. En primer lugar, fijemos un índice con expresión binaria $n = (b_{m-1}\cdots b_0)_{2)}$, definida en la proposición anterior (2.8). Probemos el lema por inducción:

- $k = 0$. Se reduce a expresar el movimiento de $\mathbf{S}_{2^m}$, reescribiendo la expresión binaria:

$$r_0\left[(b_{m-1}\cdots b_0)_{2)}\right] = s_0\left[(b_{m-1}\cdots b_0)_{2)}\right] = s_0\left[(b_0)_{2)} + (b_{m-1}\cdots b_1)_{2)}\cdot 2\right]$$

  Notando que $0 \le b_0 < 2$ y que $0 \le (b_{m-1}\cdots b_1)_{2)} < 2^{m-1}$ y usando la expresión (2.4) del movimiento s_0 se tiene el resultado:

$$s_0\left[(b_0)_{2)} + (b_{m-1}\cdots b_1)_{2)}\cdot 2\right] = (b_{m-1}\cdots b_1)_{2)} + (b_0)_{2)}\cdot 2^{m-1}$$

- $0 < k < m-1$. Supongamos que es cierto para $k-1$. Entonces:

$$\begin{aligned} r_k\left[(b_{m-1}\cdots b_0)_{2)}\right] &= s_k\left[r_{k-1}\left[(b_{m-1}\cdots b_0)_{2)}\right]\right] = s_k\left[(b_{m-1}\cdots b_k)_{2)} + (b_0\cdots b_{k-1})_{2)}\cdot 2^{m-k}\right] \\ &= s_k\left[(b_k)_{2)} + (b_{m-1}\cdots b_{k+1})_{2)}\cdot 2 + (b_0\cdots b_{k-1})_{2)}\cdot 2^{m-k}\right] \end{aligned}$$

  Notando que $0 \le b_k < 2$, $0 \le (b_{m-1}\cdots b_{k+1})_{2)} < 2^{m-k-1}$ y $0 \le (b_0\cdots b_{k-1})_{2)} < 2^k$, y usando la expresión del movimiento s_k obtenido en la proposición (2.7) se tiene la inducción:

$$\begin{aligned} &= (b_{m-1}\cdots b_{k+1})_{2)} + (b_k)_{2)}\cdot 2^{m-k-1} + (b_0\cdots b_{k-1})_{2)}\cdot 2^{m-k} \\ &= (b_{m-1}\cdots b_{k+1})_{2)} + \left[(b_k)_{2)} + (b_0\cdots b_{k-1})_{2)}\cdot 2\right]\cdot 2^{m-k-1} = (b_{m-1}\cdots b_{k+1})_{2)} + (b_0\cdots b_k)_{2)}\cdot 2^{m-k-1} \end{aligned}$$

□

Teorema 2.10. *Sea* $\mathbf{B}_{2^m} := \mathbf{S}_2^{(2^{m-1})}\mathbf{S}_4^{(2^{m-2})}\ldots\mathbf{S}_{2^{m-1}}^{(2)}\mathbf{S}_{2^m}$ *y* $f \in \mathbb{C}^{2^m}$. *Entonces se tiene que:*

$$(\mathbf{B}_{2^m} f)_{(b_0 b_1\cdots b_{m-2}b_{m-1})_2} = f_{(b_{m-1}b_{m-2}\cdots b_1 b_0)_2}$$

Es decir, la acción de la permutación asociada a $\mathbf{B}_{2^m}$ *sobre las componentes del vector* f *mueve la componente de índice* $(b_{m-1}b_{m-2}\cdots b_1 b_0)_{2)}$ *a la componente de índice* $(b_0 b_1\cdots b_{m-2}b_{m-1})_{2)}$:

$$r_{m-1}\left[(b_{m-1}b_{m-2}\cdots b_1 b_0)_{2)}\right] = (b_0 b_1\cdots b_{m-2}b_{m-1})_{2)}$$

Demostración. Sea n un índice con expresión binaria $n = (b_{m-1}\cdots b_0)_{2)}$. Notemos que por la observación (2.5) se tiene que:

$$\mathbf{B}_{2^m} = S_2^{(2^{m-1})}\mathbf{S}_4^{(2^{m-2})}\ldots\mathbf{S}_{2^{m-1}}^{(2)}\mathbf{S}_{2^m} = \mathbf{S}_4^{(2^{m-2})}\ldots\mathbf{S}_{2^{m-1}}^{(2)}\mathbf{S}_{2^m}$$

Que en términos de movimientos se traslada a la siguiente igualdad:

$$r_{m-1}\left[(b_{m-1}b_{m-2}\cdots b_1 b_0)_{2)}\right] = r_{m-2}\left[(b_{m-1}b_{m-2}\cdots b_1 b_0)_{2)}\right]$$

Y gracias al lema anterior (2.9) se tiene el resultado:

$$r_{m-2}\left[(b_{m-1}b_{m-2}\cdots b_1 b_0)_{2)}\right] = (b_{m-1})_{2)} + (b_0\cdots b_{m-2})_{2)}\cdot 2 = (b_0\cdots b_{m-1})_{2)}$$

□

Observación. Nótese que la permutación representada por $\mathbf{B}_{2^m}$ es una transposición:

$$r_{m-1}^2\left[(b_{m-1}\cdots b_0)_{2)}\right] = r_{m-1}\left[(b_0\cdots b_{m-1})_{2)}\right] = (b_{m-1}\cdots b_0)_{2)}$$

En la literatura, este movimiento de componentes recibe el nombre de *"bit reversal permutation"*.

2.3.2. Algoritmo de Bracewell-Buneman

En esta sección deduciremos uno de los múltiples algoritmos existentes para calcular el *"bit revesal"* de los índices $0 \leq i < N = 2^m$. Este algoritmo tomará como entrada un entero $m \geq 0$, para devolver el vector r de tamaño 2^m cuya componente i-ésima contiene la representación binaria de m bits invertida de $i = (b_{m-1} \cdots b_0)_{2)}$, es decir:

$$r = [r_i]_{0 \leq i < 2^m}, \quad \text{donde} \quad r_i = r_{m-1}[i] = r_{m-1}[(b_{m-1} \cdots b_0)_{2)}]$$

Observación. Nótese que aunque las expresiones $(0b_{m-1}b_{m-2} \cdots b_1 b_0)_2$ y $(b_{m-1}b_{m-2} \cdots b_1 b_0)_2$ representan el mismo número, con $m+1$ y m bits respectivamente, en general se tiene que:

$$r_m[(0b_{m-1}b_{m-2} \cdots b_1 b_0)_2] \neq r_{m-1}[(b_{m-1}b_{m-2} \cdots b_1 b_0)_2]$$

Sin embargo, se cumple la siguiente relación:

$$\begin{aligned} r_m[b_m b_{m-1} \cdots b_1 b_0)_2] &= (b_0 b_1 \cdots b_{m-1} b_m)_2 \\ &= 2 \cdot (b_0 b_1 \cdots b_{m-1})_2 + b_m \\ &= 2 \cdot r_{m-1}[(b_{m-1} \cdots b_1 b_0)_2] + b_m \end{aligned}$$

Que expresado de forma compacta:

$$r_m[n] = \begin{cases} 2 \cdot r_{m-1}[n] & \text{si} \quad n = 0, 1, \ldots, 2^{m-1} - 1 \\ 2 \cdot r_{m-1}[n - 2^{m-1}] + 1 & \text{si} \quad n = 2^{m-1}, 2^{m-1} + 1, \ldots, 2^m - 1 \end{cases}$$

Esta expresión será la célula que usará el algoritmo. De manera ilustrativa, el proceso iterativo se puede ver en la siguiente tabla:

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$r_0[n]$	0	1														
$r_1[n]$	0	2	1	3												
$r_2[n]$	0	4	2	6	1	5	3	7								
$r_3[n]$	0	8	4	12	2	10	6	14	1	9	5	13	3	11	7	15

La mitad izquierda de la fila m se obtiene doblando los valores de la fila anterior $m-1$, y la mitad derecha, sumando uno a los elementos de la mitad izquierda obtenida.

Con ello, el algoritmo de Bracewell-Buneman es el siguiente:

Algoritmo 1: Algoritmo de Bracewell-Buneman

Entrada: Entero $m > 0$ (número de bits)
Salida : El vector r: $r[i] = r_{m-1}[i]$, $0 \leq i < 2^m$

```
1 r[0] := 0 ; M := 1
2 for i = 0, 1, ..., m-1 do
3     for k = 0, 1, ..., M-1 do
4         r[k] := 2 · r[k]
5         r[k+M] := r[k] + 1
6     M := 2M
```

Dado que el número de operaciones que se realizan en el bucle interno (líneas 4-5) es constante y que $M = 2^i$ en cada iteración del bucle externo (líneas 2-6), el coste en tiempo de este algoritmo es del orden de $\sum_{i=0}^{m-1} 2^i = 2^m - 1$. Así, si suponemos que $N = 2^m$, entonces el coste es $O(N)$.

No obstante, notemos que al tratarse de una transposición, realmente no es necesario calcular todos los índices $r_{m-1}[n]$, $0 \leq n < N$, ya que n se intercambia con $r_{m-1}[n]$ y $r_{m-1}[n]$ con n. Con un estudio de este hecho, se puede modificar el algoritmo para calcular solamente los pares $(r_{m-1}[n], n)$ tales que $r_{m-1}[n] > n$, reduciendo el coste en tiempo y con un coste en espacio de a lo sumo $\sqrt{2N}$.

2.3.3. Descripción del algoritmo en el caso $N = 2^m$

Con todo el desarrollo realizado y los resultados obtenidos, ya nos encontramos en situación para presentar el esquema del algoritmo de la Transformada Rápida de Fourier (FFT), cuando $N = 2^m$. Dicho algoritmo tomará como entrada el valor m y un vector $f \in \mathbb{C}$, para devolver el vector resultante del producto $F_N f$.

El cálculo del producto se basa en la factorización obtenida en el teorema (2.6), realizando los productos de derecha a izquierda:

$$\frac{1}{2^m}\mathbf{Q}_{2^m}\mathbf{Q}_{2^{m-1}}^{(2)}\dots\mathbf{Q}_{2}^{(2^{m-1})}\mathbf{B}_{2^m}f = \mathbf{F}_N f \tag{2.6}$$

Antes de presentar el algoritmo completo, desgranaremos y detallaremos los trozos del algoritmo que calculan estos productos.

Como hemos visto en el teorema (2.10), la matriz $\mathbf{B}_{2^m}$ actúa sobre el vector f moviendo su componente de índice $n = (b_{m-1}b_{m-2}\cdots b_1 b_0)_2$ a la componente de índice $r[n] = (b_0 b_1 \cdots b_{m-2}b_{m-1})_2$. Así, con el *"bit reversal"* de cada índice precalculado (Alg. (1)) podemos realizar el producto $\mathbf{B}_{2^m}f$ intercambiando pares de componentes de f, por ser una transposición (Obs. (2.3.1)).

Algoritmo 2: Cálculo de $B_{2^m}f$

Entrada: Entero $m > 0$ y $f \in \mathbb{C}^N$, $N = 2^m$
Salida : El vector permutado $\mathbf{B}_{2^m}f$

```
1 Alg. (1) "bit reversal" → r
2 for n = 1,2,...,N−2 do
3     if r[n] > n then
4         ⟨f_n, f_{r[n]}⟩ := ⟨f_{r[n]}, f_n⟩
```

Notemos que es innecesario intercambiar las componentes 0, $N-1$-ésimas debido a que $r[0] = 0$ y $r[N-1] = N-1$.

Como ya hemos visto, el cálculo del "bit reversal"(línea 1) tiene un coste en tiempo de $O(N)$. Igualmente, el bucle del algortimo (líneas 2-4) tiene un coste $O(N)$. Por tanto, el coste en tiempo de este algoritmo es $O(N)$.

Por el otro lado tenemos las matrices de la forma $\mathbf{Q}_{2M}^{(K)}$, matrices diagonales por bloques donde cada bloque es la matriz $\mathbf{Q}_{2M}$, cuya expresión simplificamos en la ecuación (2.3). Aprovechando esa repetición en los bloques, el siguiente trozo de algoritmo calcula de forma inteligente el producto $\mathbf{Q}_{2M}^{(K)}f$:

Algoritmo 3: Cálculo de $\mathbf{Q}_{2M}^{(K)}f$

Entrada: Enteros $M, K > 0$ y $f \in \mathbb{C}^{2MK}$
Salida : El vector $\mathbf{Q}_{2M}^{(K)}f$

```
1 ω := e^{−πi/M}
2 for λ = 0,1,...,M−1 do
3     for κ = 0,1,...,K−1 do
4         [ f_{κ·2M+λ}     ]   [ 1   ω^λ  ] [ f_{κ·2M+λ}     ]
          [ f_{κ·2M+λ+M}   ] := [ 1  −ω^λ  ] [ f_{κ·2M+λ+M}   ]
```

Como el número de operaciones del bucle interno (líneas 3-4) es constante y, por otro lado, M y K son valores también constantes, el coste en tiempo de este algoritmo es $O(MK)$ (sin tener en cuenta el cálculo de las potencias de ω).

Finalmente, uniendo el algoritmo (2) y añadiendo un bucle para las matrices $\mathbf{Q}$, obtenemos el algoritmo completo que realiza todos los productos matriciales de la factorización (2.6):

Algoritmo 4: Algoritmo FFT cuando $N = 2^m$

Entrada: Entero $m > 0$ y $f \in \mathbb{C}^N$
Salida : El vector $\mathbf{F}_{2^m} f$

```
1  f := B_{2^m} f    (Alg. 2)
2  for μ = 1,2,...,m do
3      ω := e^{-2πi/2^μ} ; U := 1
4      for λ = 0,1,...,2^{μ-1} - 1 do
5          for κ = 0,1,...,2^{m-μ} - 1 do
6              g := f_{κ·2^μ+λ}
7              f_{κ·2^μ+λ} := g + U · f_{κ·2^μ+λ+2^{μ-1}}
8              f_{κ·2^μ+λ+2^{μ-1}} := g - U · f_{κ·2^μ+λ+2^{μ-1}}
9          U := ωU
10 for k = 0,1,...,N-1 do
11     f_k := f_k / 2^m
```

Por un lado, nótese que el bucle interno (líneas 3-9) se corresponde al algoritmo (3) con $M = 2^{\mu-1}$ y $K = 2^{m-\mu}$ como valores de entrada. Como hemos visto, el coste en tiempo de dicho bucle es $O(MK) = O(2^{m-1})$, es decir, $O(N)$. No es de extrañar que este coste no dependa de m, ya que las matrices $\mathbf{Q}_{2M}^{(K)}$ del producto (2.6) cumplen que $2MK = N$. Así, como este coste es constante en cada iteración del bucle externo (líneas 2-9), el coste en tiempo del bucle es $O(mN)$, $m = \log_2 N$.

Por otro lado, es sabido ya que el coste en tiempo del cálculo de $\mathbf{B}_{2^m} f$ (línea 1) es $O(N)$, calculado previamente en el estudio del algoritmo (2). El bucle final (líneas 10-11) también tiene coste $O(N)$.

Con este estudio, el coste en tiempo del algoritmo FFT cuando $N = 2^m$ es $O(N \log_2 N)$. Cabe mencionar que el coste en espacio es $O(N)$, al estar únicamente almacenado el vector $f \in \mathbb{C}^N$ y el vector "bit reversal" $r \in \mathbb{Z}^N$ involucrado en el cálculo de $\mathbf{B}_{2^m} f$ (línea 1).

2.4. Factorización de $\mathbf{F}_N$ en el caso $N = P_1 \cdots P_N$

En esta última sección generalizaremos la factorización de la matriz $\mathbf{F}$ para cualquier factorización $N = P_1 \cdot P_2 \cdots P_m$. Del mismo modo que en el caso $N = 2^m$, aplicaremos recursivamente la identidad cremallera con los factores de la descomposición de N, relacionando $\mathbf{F}_{P_1 \cdots P_N}$ con $\mathbf{F}_{P_1 \cdots P_{N-1}}$ y sucesivamente, reduciendo los factores hasta llegar al caso trivial $\mathbf{F}_1 = [1]$. Esta factorización se recoge en el siguiente resultado:

Teorema 2.11. *Sea $N = P_1 \cdots P_m$. Entonces:*

$$\mathbf{F}_N = \frac{1}{P_1 \cdots P_m} \cdot \mathbf{Q}_{P_1 \cdots P_{m-1}, P_m} \cdot \mathbf{Q}_{P_1 \cdots P_{m-2}, P_{m-1}}^{(P_m)} \cdots \mathbf{Q}_{P_1, P_2}^{(P_3 \cdots P_m)} \cdot \mathbf{Q}_{1, P_1}^{(P_2 \cdots P_m)} \cdot \mathbf{S}_{P_1, P_2, \ldots, P_{m-1}, P_m}$$

$$\textit{donde} \quad \mathbf{S}_{P_1, P_2, \ldots, P_{m-1}, P_m} := \mathbf{S}_{P_1, P_2}^{(P_3 \cdots P_m)} \cdot \mathbf{S}_{P_1 P_2, P_3}^{(P_4 \cdots P_m)} \cdots \mathbf{S}_{P_1 \cdots P_{m-2}, P_{m-1}}^{(P_m)} \cdot \mathbf{S}_{P_1 \cdots P_{m-1}, P_m}$$

Demostración. La demostración seguirá el mismo esquema usado en la factorización del caso $N = 2^m$ 2.6. Aplicaremos la identidad cremallera tomando $P = P_1 \cdots P_{k-1}$, $Q = P_k$, $k = m, \ldots, 1$, hasta llegar al

caso trivial $\mathbf{F}_1 = 1$.

$$\begin{aligned}
\mathbf{F}_{P_1\cdots P_m} &= \frac{1}{P_m}\cdot\mathbf{Q}_{P_1\cdots P_{m-1},P_m}\mathbf{F}^{(P_m)}_{P_1\cdots P_{m-1}}\mathbf{S}_{P_1\cdots P_{m-1},P_m}\\
&= \frac{1}{P_{m-1}P_m}\cdot\mathbf{Q}_{P_1\cdots P_{m-1},P_m}\cdot\left[\mathbf{Q}_{P_1\cdots P_{m-2},P_{m-1}}\cdot\mathbf{F}^{(P_{m-1})}_{P_1\cdots P_{m-2}}\cdot\mathbf{S}_{P_1\cdots P_{m-2},P_{m-1}}\right]^{(P_m)}\cdot\mathbf{S}_{P_1\cdots P_{m-1},P_m}\\
&= \frac{1}{P_{m-1}P_m}\cdot\mathbf{Q}_{P_1\cdots P_{m-1},P_m}\cdot\mathbf{Q}^{(P_m)}_{P_1\cdots P_{m-2},P_{m-1}}\cdot\mathbf{F}^{(P_{m-1}\cdot P_m)}_{P_1\cdots P_{m-2}}\cdot\mathbf{S}^{(P_m)}_{P_1\cdots P_{m-2},P_{m-1}}\cdot\mathbf{S}_{P_1\cdots P_{m-1},P_m}\\
&\vdots\\
&= \frac{1}{P_1\cdots P_m}\cdot\mathbf{Q}_{P_1\cdots P_{m-1},P_m}\cdots\mathbf{Q}^{(P_3\cdots P_m)}_{P_1,P_2}\cdot\mathbf{Q}^{(P_2\cdots P_m)}_{1,P_1}\cdot\mathbf{F}^{(P_1\cdots P_m)}_{1}\cdot\mathbf{S}^{(P_2\cdots P_m)}_{1,P_1}\cdot\mathbf{S}^{(P_3\cdots P_m)}_{P_1,P_2}\cdots\mathbf{S}_{P_1\cdots P_{m-1},P_m}
\end{aligned}$$

Teniendo en cuenta que:

$$\mathbf{F}^{(P_1\cdots P_m)}_1 = [1]^{(P_1\cdots P_m)} = \mathbf{I}_{P_1\cdots P_m}$$

$$\sigma_{1,P_1}(q) = q,\ 0\le q<P_1 \implies \mathbf{S}^{(P_2\cdots P_m)}_{1,P_1} = \mathbf{I}_{P_1\cdots P_m}$$

Y reuniendo en $\mathbf{S}_{P_1,\ldots,P_m}$ todas las matrices de permutación generadas:

$$\mathbf{S}_{P_1,\ldots,P_m} := \mathbf{S}^{(P_3\cdots P_m)}_{P_1,P_2}\cdot\mathbf{S}^{(P_4\cdots P_m)}_{P_1P_2,P_3}\cdots\mathbf{S}^{(P_m)}_{P_1\cdots P_{m-2},P_{m-1}}\cdot\mathbf{S}_{P_1\cdots P_{m-1},P_m}$$

Se tiene el resultado:

$$\mathbf{F}_{P_1\cdots P_m} = \frac{1}{P_1\cdots P_m}\cdot\mathbf{Q}_{P_1\cdots P_{m-1},P_m}\cdot\mathbf{Q}^{(P_m)}_{P_1\cdots P_{m-2},P_{m-1}}\cdots\mathbf{Q}^{(P_3\cdots P_m)}_{P_1,P_2}\cdot\mathbf{Q}^{(P_2\cdots P_m)}_{1,P_1}\cdot\mathbf{S}_{P_1,\ldots,P_m}$$

□

Nótese que esta factorización será diferente según los divisores que consideremos en la descomposición de N y del orden en el que los tomemos. Como cabe esperar, la factorización coincidirá con la obtenida en (2.6) cuando todos los factores sean $P_k = 2$.

2.4.1. Acción de $\mathbf{S}_{P_1,\ldots,P_m}$

Análogamente a la sección (2.3.1), describiremos cada uno de los movimientos de las matrices que conforman la matriz de permutación $\mathbf{S}_{P_1,\ldots,P_m}$ para posteriormente componerlos y hallar el movimiento de las componentes que realiza el producto total. Dichos movimientos, expresados con la misma notación que en la sección (2.3.1), están reunidos en los siguientes resultados:

Proposición 2.12. *La acción representada por la matriz $\mathbf{S}_{P_1\cdots P_{m-1},P_m}$ está descrita por el movimiento:*

$$s_{P_m}[q+p\cdot P_m] = p+q\cdot(P_1\cdots P_{m-1}),\quad 0\le p<(P_1\cdots P_{m-1})\quad 0\le q<P_m$$

Es decir, al actuar sobre un vector, $\mathbf{S}_{P_1\cdots P_{m-1},P_m}$ mueve la componente de índice $q+p\cdot P_m$ a la componente de índice $p+q\cdot(P_1\cdots P_{m-1})$.

Proposición 2.13. *Sea k un entero tal que $2\le k<m$. La acción de la matriz $\mathbf{S}^{(P_{k+1}\cdots P_m)}_{P_1\cdots P_{k-1},P_k}$, está descrita por el movimiento:*

$$\begin{aligned}
&s_{P_k}[q+p\cdot P_k+r\cdot(P_1\cdots P_k)] = p+q\cdot(P_1\cdots P_{k-1})+r\cdot(P_1\cdots P_k),\\
&\textit{donde}\quad 0\le p<(P_1\cdots P_{k-1})\quad 0\le q<P_k\quad 0\le r<P_{k+1}\cdots P_m
\end{aligned}$$

Es decir, al actuar sobre un vector, $\mathbf{S}^{(P_{k+1}\cdots P_m)}_{P_1\cdots P_{k-1},P_k}$ mueve la componente de índice $q+p\cdot P_k+r\cdot(P_1\cdots P_k)$ a la componente de índice $p+q\cdot(P_1\cdots P_{k-1})+r\cdot(P_1\cdots P_k)$.

De manera análoga al resultado (2.8), necesitaremos una descomposición de los índices particular para poder deducir el movimiento de $\mathbf{S}_{P_1,\dots,P_m}$:

Proposición 2.14. *Sea $N = P_1 \cdots P_m$. Entonces, para todo entero n tal que $0 \leq n < N$ existen únicos $\{p_i\}_{i=1}^m$ cumpliendo:*

$$n = \sum_{i=1}^{m} p_i \left(\prod_{j>i}^{m} P_j \right) = p_m + p_{m-1} \cdot P_m + p_{m-2} \cdot P_{m-1} \cdot P_m + \cdots + p_1 \cdot (P_2 \cdots P_m)$$

$$\textit{donde} \quad 0 \leq p_i < P_i, \quad 1 \leq i \leq m$$

De este modo, con los movimientos y la descomposición de índices, ya podemos describir el movimiento de $\mathbf{S}_{P_1,\dots,P_N}$:

Teorema 2.15. *La acción de la matriz $\mathbf{S}_{P_1,\cdots,P_m}$ está descrita por el movimiento:*

$$s\left[\sum_{i=1}^{m} p_i \left(\prod_{j>i}^{m} P_j \right) \right] = \sum_{i=1}^{m} p_{m+1-i} \left(\prod_{j>i}^{m} P_{m+1-j} \right), \quad 0 \leq p_i < P_i, \quad 1 \leq i \leq m$$

Es decir, al actuar sobre un vector, $\mathbf{S}_{P_1,\dots,P_N}$ mueve la componente de índice:

$$n = p_m + p_{m-1} \cdot P_m + p_{m-2} \cdot P_{m-1} \cdot P_m + \cdots + p_1 \cdot (P_2 \cdots P_m)$$

a la componente de índice:

$$s[n] = p_1 + p_2 \cdot P_1 + p_3 \cdot P_1 \cdot P_2 + \cdots + p_m \cdot (P_{m-1} \cdots P_1)$$

Demostración. Basta aplicar ordenadamente los movimientos involucrados, descritos en las proposiciones (2.12) y (2.13), para obtener el resultado. □

Notemos que en general, no podemos asegurar que $\mathbf{S}_{P_1,\dots,P_N}$ sea una transposición como lo era $\mathbf{S}_{2^m}$. Por ello, a la hora de mover las componentes, ya no se tratarán de intercambios entre pares de componentes.

2.4.2. Descripción del algoritmo en el caso $N = P_1 \cdots P_m$

Finalmente, en esta sección presentaremos el esquema del algoritmo de la Transformada Rápida de Fourier (FFT) en el caso general, cuando $N = P_1 \cdots P_m$. Dicho algoritmo tomará como entrada la factorización $N = P_1 \cdots P_m$ y un vector $f \in \mathbb{C}$, para devolver el vector resultante del producto $F_N f$.

Análogamente a la descripción del caso $N = 2^m$ en la sección (2.3.3), el cálculo del vector se basa en la factorización obtenida en el teorema (2.11), realizando los productos de derecha a izquierda:

$$\frac{1}{P_1 \cdots P_m} \cdot \mathbf{Q}_{P_1 \cdots P_{m-1}, P_m} \cdots \mathbf{Q}_{P_1, P_2}^{(P_3 \cdots P_m)} \cdot \mathbf{Q}_{1, P_1}^{(P_2 \cdots P_m)} \cdot \mathbf{S}_{P_1, P_2, \dots, P_{m-1}, P_m} f$$

Como hemos visto en la sección anterior (2.4.1), particularmente en el teorema (2.15), la matriz $\mathbf{S}_{P_1,\dots,P_m}$ actúa sobre el vector f moviendo su componente de índice:

$$n = p_m + p_{m-1} \cdot P_m + p_{m-2} \cdot P_{m-1} \cdot P_m + \cdots + p_1 \cdot (P_2 \cdots P_m)$$

a la componente de índice:

$$s[n] = p_1 + p_2 \cdot P_1 + p_3 \cdot P_1 \cdot P_2 + \cdots + p_m \cdot (P_{m-1} \cdots P_1)$$

De este modo, podemos realizar el producto $\mathbf{S}_{P_1,\dots,P_m} f$ calculando el movimiento de cada índice (con el algoritmo de la división) y recurriendo a un vector auxiliar para reordenar las componentes:

Algoritmo 5: Cálculo de $\mathbf{S}_{P_1,\ldots,P_m} f$

Entrada: $m > 0$, $P_1, \ldots, P_m \geq 2$ y
$f \in \mathbb{C}^N$, con $N = P_1 \cdots P_m$
Salida : El vector $\mathbf{S}_{P_1,\ldots,P_m} f$

1 **for** $n = 0, 1, \ldots, N-1$ **do**
2 $s := 0$; $d := n$
3 **for** $\mu = m, m-1, \ldots, 1$ **do**
4 $q := \lfloor d/P_\mu \rfloor$
5 $p := d - P_\mu \cdot q$
6 $r := p + P_\mu \cdot r$
7 $d := q$
8 $g[s] := f[n]$
9 **for** $n = 0, 1, \ldots, N-1$ **do**
10 $f[n] := g[n]$

Notemos que el número de operaciones que se realizan en el bucle interno (líneas 3-7) es constante y que en cada iteración del bucle externo (líneas 1-8) únicamente se realizan asignaciones, por lo que el coste en tiempo del bucle es del orden de mN, el número de iteraciones realizadas. Trivialmente, el coste en tiempo del último bucle (líneas 9-10) es $O(N)$. Por tanto, el coste en tiempo del algoritmo completo es $O(mN)$.

El algoritmo requiere en espacio a los m factores $P_1, \ldots, P_m$, al vector f y al vector auxiliar g, estos dos últimos de tamaño N. Nótese que $m \leq N$, ya que el número de factores que descomponen un número es menor que dicho número. Por tanto el coste en espacio del algoritmo es $O(\text{máx}\{m, N\}) = O(N)$.

Análogamente, nos faltará ver los productos de las matrices $\mathbf{Q}_{P_1\cdots P_{k-1}, P_k}^{P_{k+1}\cdots P_m}$. Este algoritmo sigue la misma idea que el algoritmo (3) del caso $N = 2^m$:

Algoritmo 6: Cálculo de $\mathbf{Q}_{M,P}^{(K)} f$

Entrada: Enteros $M, P, K > 0$ y $f \in \mathbb{C}^{MPK}$
Salida : El vector $\mathbf{Q}_{M,P}^{(K)} f$

1 **for** $\lambda = 0, 1, \ldots, M-1$ **do**
2 **for** $\kappa = 0, 1, \ldots, K-1$ **do**
3 $$\begin{bmatrix} f_{\lambda+\kappa MP} \\ f_{\lambda+M+\kappa MP} \\ \vdots \\ f_{\lambda+(P-1)M+\kappa MP} \end{bmatrix} := \Omega_{\lambda,M,P} \begin{bmatrix} f_{\lambda+\kappa MP} \\ f_{\lambda+M+\kappa MP} \\ \vdots \\ f_{\lambda+(P-1)M+\kappa MP} \end{bmatrix}$$

Donde

$$\Omega_{\lambda,M,P} := \begin{bmatrix} 1 & \omega^{\lambda} & \omega^{2\lambda} & \cdots & \omega^{(P-1)\lambda} \\ 1 & \omega^{\lambda+M} & \omega^{2(\lambda+M)} & \cdots & \omega^{(P-1)(\lambda+M)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^{\lambda+(P-1)M} & \omega^{2[\lambda+(P-1)M]} & \cdots & \omega^{(P-1)[\lambda+(P-1)M]} \end{bmatrix}, \quad \omega := e^{-2\pi i/MP}$$

Notemos que el producto matricial de la línea 3 del algoritmo anterior requiere $P-1$ sumas y solamente $P-1$ productos de números complejos para actualizar cada una de las P componentes del vector, ya que todos los elementos de la primera columna de $\Omega_{\lambda,M,P}$ son 1. De este modo, sin tener en cuenta el cálculo de las potencias de ω, como el número de operaciones es constante en cada iteración del bucle interno, y los valores M, K también son constantes, el coste en tiempo del algoritmo es $O(MK(P-1)P)$.

Juntando ambos algoritmos y añadiendo un bucle para calcular todos los productos de las matrices $\mathbf{Q}_{P_1\cdots P_{k-1},P_k}^{P_{k+1}\cdots P_m}$, el algoritmo general completo es el siguiente:

Algoritmo 7: Algoritmo FFT cuando $N = P_1 \cdots P_m$

Entrada: $m > 0$, $P_1, \ldots, P_m \geq 0$ y $f \in \mathbb{C}^N$, con $N = P_1 \cdots P_m$
Salida : El vector $\mathbf{F}_N f$

```
1  f := S_{P_1,...,P_m} f    (Alg. 5)
2  M := 1 ; P := 1 ; K := N
3  for μ = 1,2,...,m do
4      P := P_μ ; K := K/P
5      ω := e^{-2πi/MP}
6      W := 1 ; V := ω^M
7      g := copy(f)
8      for λ = 0,1,...,M-1 do
9          Λ := W
10         for κ = 0,1,...,K-1 do
11             for i = 0,...,P-1 do
12                 u := g_{λ+κMP} ; Π := Λ
13                 for j = 1,...,P-1 do
14                     u := u + Π · g_{λ+jM+κMP}
15                     Π := Π · Λ
16                 f_{λ+iM+κMP} := u
17                 Λ := Λ · V
18         W := W · ω
19     M := M · P
20 for λ = 0,1,...,N-1 do
21     f_k := f_k/N
```

Notemos que el bucle interno (líneas 8-18) se corresponde al algoritmo (6) con $M = P_1 \cdots P_{\mu-1}$, $P = P_\mu$ y $K = P_{\mu+1} \cdots P_m$ como valores de entrada; salvo el caso particular $\mu = 1$, donde toma los valores $M = 1$, $P = P_1$ y $K = P_2 \cdots P_m$. Por lo ya analizado, el coste en tiempo de dichas líneas es $O(MK(P-1)P)$. Aprovechando que para estos valores de M, P, K se cumple que $MPK = N$, lo podemos reescribir como $O(N(P_\mu - 1))$. Conocido el coste del bucle interno para cada iteración del bucle externo (líneas 3-19), el coste en tiempo del bucle es $O\left(N \cdot \sum_{\mu=1}^{m}(P_\mu - 1)\right)$. Por otra parte, gracias al estudio previo, también conocemos que el coste en tiempo del cálculo de $\mathbf{S}_{P_1,\ldots,P_m} f$ (línea 1) es $O(mN)$. Finalmente, el último bucle (líneas 20-21) tiene coste $O(N)$.

Con esto y teniendo en cuenta la siguiente desigualdad

$$\sum_{\mu=1}^{m}(P_\mu - 1) \geq m \quad (\text{ya que } P_1, \ldots, P_m \geq 2)$$

el coste en tiempo del algoritmo FFT cuando $N = P_1 \cdots P_m$, es el anunciado al inicio del capítulo:

$$O(N \cdot \{(P_1 - 1) + (P_2 - 1) + \cdots + (P_m - 1)\})$$

Para acabar, hemos visto que el cálculo de $\mathbf{S}_{P_1,\ldots,P_m} f$ tiene un coste en espacio $O(N)$ y en el resto del algoritmo únicamente requieren espacio la copia de f en en vector g, del mismo tamaño N, y las variables definidas. Por tanto, el coste en espacio del algoritmo es $O(N)$.

Bibliografía

[1] DAVID W. KRAMMER, *A First Course in Fourier Analysis*, Cambridge University Press, 2007.

[2] FRANCISCO J. RUIZ BLASCO, *Análisis de Fourier*, cap. 3-4, Colección Textos Docentes, Prensas Universitarias de Zaragoza.

[3] J.W. COOLEY AND J.W. TUKEY, *An algorithm for the machine computation of complex Fourier series*, Math. Comp. 19(1965), 297-301.

[4] E.O. BRIGHAM, *The Fast Fourier Transform and Its Applications*, cap. 8-9, Prentice Hall, Englewood Cliffs, NJ, 1988.

[5] J. DUOANDIKOETXEA, *Análisis de Fourier*, cap. 6, Addison-Wesley Iberoamericana, 1995.

Apéndice A

La transformada de Fourier

En este apéndice presentamos una colección de resultados de Análisis de Fourier. Únicamente se recogen los resultados necesarios para el estudio y desarrollo de la transformada discreta de Fourier realizados en el primer capítulo (1). El contexto y desarrollo de este área se ha estudiado en mayor profundidad en la asignatura Ánalisis de Fourier del Grado de Matemáticas.

A partir de ahora asumiremos que $f : \mathbb{R} \to \mathbb{C}$ es una función $2L$-periódica. No obstante, cabe mencionar que en el capítulo de la transformada discreta se toman por comodidad funciones 1-periódicas.

Definición 9. Sea $f \in L^1([-L,L])$.

a) Llamaremos coeficiente de Fourier n-ésimo de f a

$$\hat{f}(n) = \frac{1}{2L}\int_{-L}^{L} f(t)e^{-in\frac{\pi}{L}t}dt, \quad n \in \mathbb{Z}$$

b) Llamaremos serie de Fourier de f a la serie

$$\sum_{n=-\infty}^{\infty} \hat{f}(n)e^{in\frac{\pi}{L}t}$$

c) Dado $N \in \mathbb{N} \cup \{0\}$, llamaremos suma parcial N-ésima de la serie de Fourier a

$$S_N f(t) = \sum_{n=-N}^{N} \hat{f}(n)e^{i\frac{\pi}{L}t}, \quad t \in \mathbb{R}$$

Nótese que los coeficientes de Fourier están bien definidos porque $f \in L^1([-L,L])$. Más aún, cumplirán la siguiente propiedad:

Teorema A.1. *(Lema de Riemann-Lebesgue). Sea $f \in L^1([-L,L])$. Entonces, $\{\hat{f}(n)\}_{n\in\mathbb{Z}} \in c_0(\mathbb{Z})$. Es decir,*

$$\lim_{|n|\to\infty} \hat{f}(n) = 0$$

En cambio, no podemos asegurar la convergencia de la serie de Fourier en general. No obstante, sí que tendremos convergencia para las familias de funciones de interés en el capítulo (1).

Teorema A.2. *Sea f una función continua y de clase $C^{(1}$ a trozos en $[-L,L]$. Entonces, $S_N f$ converge uniformemente a f en $\mathbb{R}$.*

Lema A.3. *Sea $f \in L^1([-\pi,\pi])$, $[a,b]$ un intervalo y $\phi \in \mathscr{C}^{(1}([a,b])$. Entonces,*

$$\lim_{\lambda\to\infty} \int_a^b f(x-t)\phi(t)\sin(\lambda t)dt = 0, \text{ uniformemente en } x \in \mathbb{R}$$

Teorema A.4. *(Convergencia uniforme local). Sea $f \in L^1([-L,L])$ y $[a,b]$ un intervalo tal que f es continua y de clase $\mathscr{C}^1$ a trozos en él. Entonces, $\forall \delta > 0$ tal que $a+\delta < b-\delta$,*

$$S_N f(t) \xrightarrow{N\to\infty} f(t) \text{ uniformemente, } t \in [a+\delta, b-\delta]$$

Demostración. Demostraremos el resultado para $L = \pi$. Supongamos que la longitud de $[a,b]$ es menor que 2π, ya que en otro caso, por el teorema anterior, la serie de Fourier converge uniformemente en todo $\mathbb{R}$. Para demostrar el resultado nos apoyaremos en una función $F \in \mathscr{C}$ y de clase $\mathscr{C}^1$ a trozos en $[-\pi,\pi]$, tal que

$$F(x) = f(x), \quad x \in [a,b]$$

Por ejemplo, la función extendida que une los valores $f(b+k\cdot 2\pi)$ y $f(a+(k+1)\cdot 2\pi)$, $k \in \mathbb{Z}$, con líneas rectas. Es claro que dicha función es continua y de clase $\mathscr{C}^1$ a trozos.
Denotemos como $G(x) = f(x) - F(x)$ a la diferencia. Es claro que esta función es $L^1([-\pi,\pi])$, cumpliendo que $G(x) = 0$ si $x \in [a,b]$. Entonces se tiene que:

$$S_N f = S_N F + S_N G$$

Como por el teorema (A.2) $S_N F$ converge uniformemente a F, tenemos que $S_N F$ converge uniformemente a f en $[a,b]$. Solo nos faltará ver que $S_N G$ converge a 0 uniformemente en $[a+\delta, b-\delta]$ para tener el resultado. Fijemos entonces un $\delta > 0$ y tomemos $x \in [a+\delta, b-\delta]$. Reescribimos la suma parcial como:

$$S_N G(x) = \frac{1}{2\pi}\int_{-\pi}^{\pi} G(x-t)\frac{\sin(N+\frac{1}{2})t}{\sin\frac{t}{2}}dt$$

Si $-\delta \geq t \geq \delta \implies a \geq x-\delta \geq x-t \geq x+\delta \geq b$, entonces $G(x-t) = 0$. Luego,

$$S_N G(x) = \frac{1}{2\pi}\int_{-\pi}^{-\delta} G(x-t)\frac{\sin(N+\frac{1}{2})t}{\sin\frac{t}{2}}dt + \frac{1}{2\pi}\int_{\delta}^{\pi} G(x-t)\frac{\sin(N+\frac{1}{2})t}{\sin\frac{t}{2}}dt$$

Basta aplicar el lema anterior a las funciones $G \in L^1([-\pi,\pi])$ y $\phi(t) = \frac{1}{\sin t/2}$, que es de clase $\mathscr{C}^1$ en los intervalos $[-\pi,-\delta]$ y $[\delta,\pi]$.

□

Teorema A.5. *Sea $f \in L^1([-L,L])$ y $t_0 \in \mathbb{R}$ un punto tal que f es derivable tanto a izquierda como a derecha de t_0. Entonces,*

$$\lim_{N\to\infty} S_N f(t_0) = \frac{f(t_0^+) - f(t_0^-)}{2}$$

Donde $f(t_0^+)$ y $f(t_0^-)$ denotan los respectivos límites laterales de f en t_0.

Lema A.6. *Sea $\alpha_N = \frac{L}{2(N+1)}, N \in \mathbb{N}$. Entonces,*

$$\lim_{n\to\infty} S_N(\psi_{t_0})(\alpha_N) = \frac{2}{\pi}\int_0^{\pi}\frac{\sin x}{x}dx \equiv \gamma$$

Donde ψ_{t_0} es la función $2L$-periódica generada por

$$\psi_{t_0}(x) := \begin{cases} -1 & si\ x \in [t_0-L, t_0) \\ 1 & si\ x \in [t_0, t_0+L) \end{cases}$$

Teorema A.7. *(Fenómeno de Gibbs). Sea $f \in L^1([-L,L])$ y $t_0 \in \mathbb{R}$ un punto tal que f es derivable tanto a izquierda como a derecha de t_0. Además, supongamos que la diferencia entre los límites laterales es positiva, $f(t_0^+) - f(t_0^-) := 2A > 0$. Entonces,*

$$\lim_{N\to\infty} S_N f(t_0 + \frac{L}{2(N+1)}) = f(t_0^+) + A(\gamma - 1)$$
$$\lim_{N\to\infty} S_N f(t_0 - \frac{L}{2(N+1)}) = f(t_0^-) - A(\gamma - 1)$$

De forma más general, si $\{t_N\}_{N=0}^{\infty}$ es una sucesión que converge a t_0 cuando $N \to \infty$, entonces:

$$\limsup_{N\to\infty} S_N f(t_N) \leq f(t_0^+) + A(\gamma - 1)$$
$$\liminf_{N\to\infty} S_N f(t_N) \geq f(t_0^-) - A(\gamma - 1)$$

Demostración. Consideremos la función $h = f - g = f - A\psi_{t_0}$. El punto t_0 es un punto de continuidad para h, ya que

$$h(t_0^+) - h(t_0^-) = f(t_0^+) - f(t_0^-) - (g(t_0^+) - g(t_0^-)) = 2A - 2A = 0$$

Además, h tiene derivadas laterales en t_0 y

$$h(t_0) = \frac{h(t_0^+) + h(t_0^-)}{2} = \frac{f(t_0^+) + f(t_0^-)}{2}$$

Entonces, si vemos (con $\alpha_N = \frac{L}{2(N+1)}$) que

$$\lim_{N\to\infty} S_N(h)(t_0 + \alpha_N) = h(t_0) \tag{A.1}$$

habremos demostrado el resultado, pues

$$\lim_{N\to\infty} S_N(f)(t_0 + \alpha_N) = \lim_{N\to\infty} S_N(h)(t_0 + \alpha_N) + \lim_{N\to\infty} S_N(g)(t_0 + \alpha_N) = h(t_0) + A\gamma$$

La función h es continua y de clase $\mathscr{C}^{(1}$ a trozos en $(t_0 - \delta, t_0 + \delta)$. Por tanto, se da convergencia uniforme de la serie de Fourier en cualquier subintervalo cerrado. En particular, dado $\varepsilon > 0$ podemos encontrar un $n_1 \in \mathbb{N}$ tal que

$$|S_N(h)(y) - h(y)| < \frac{\varepsilon}{2}, \quad \forall y \in \left[t_0 - \frac{\delta}{2}, t_0 + \frac{\delta}{2}\right], \quad \forall n \geq n_1$$

Por otro lado, por la continuidad de h, podemos elegir $n_2 \in \mathbb{N}$ tal que $|\alpha_N| < \delta/2$ y que

$$|h(t_0 + \alpha_N) - h(t_0)| < \frac{\varepsilon}{2} \quad \forall n \geq n_2$$

Entonces, si $n \geq n_0 = \max\{n_1, n_2\}$, se tiene que

$$|S_N(h)(t_0 + \alpha_N) - h(t_0)| \leq |S_N(h)(t_0 + \alpha_N) - h(t_0) + \alpha_N| + |h(t_0 + \alpha_N) - h(t_0)| < \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon$$

Demostrando así (A.1), y con ello el teorema.

□

Corolario A.8. *Sea $f \in L^1([-L,L])$ tal que f posee una discontinuidad de salto en $t_0 \in \mathbb{R}$. Además, supongamos que f es de clase $\mathscr{C}^{(1}$ en un entorno $(t_0 - \varepsilon, t_0 + \varepsilon) \setminus \{t_0\}$. Entonces $S_N f$ no converge uniformemente a f en $(t_0 - \delta, t_0 + \delta) \setminus \{t_0\}$ $\forall \delta$ tal que $0 < \delta < \varepsilon$.*

Demostración. Supongamos que $S_N f$ converge uniformemente $(t_0-\delta,t_0+\delta)\setminus\{t_0\}\ \forall\delta$ tal que $0<\delta<\varepsilon$. En particular, $\forall\varepsilon_1>0$ existe un $N_1\in\mathbb{N}$ tal que para todo $N>N_1$ se tiene que

$$|S_N f(t)-f(t)|<\varepsilon_1,\quad \forall t\in(t_0-\delta,t_0+\delta)\setminus\{t_0\} \tag{A.2}$$

Por otro lado, por el teorema anterior se tiene que:

$$\lim_{N\to\infty}|S_N f(t_0\pm\frac{L}{2(N+1)})-f(t_0\pm\frac{L}{2(N+1)})|=A(\gamma-1)$$

Con ello, basta tomar un $\varepsilon_1<A(\gamma-1)$ para obtener la contradicción. Por un lado, existe un $N_2\in\mathbb{N}$ tal que $|\alpha_N|<\delta$, $\forall N\geq N_2$. Y por otro lado, existe un $N_3\in\mathbb{N}$ y $\varepsilon_2>0$ tal que

$$A(\gamma-1)-\varepsilon_2<|S_N f(t_0\pm\frac{L}{2(N+1)})-f(t_0\pm\frac{L}{2(N+1)})|<A(\gamma-1)+\varepsilon_2,\quad \forall N\geq N_3$$

con ε_2 cumpliendo que $\varepsilon_1<A(\gamma-1)-\varepsilon_2$. Así, tomando un $N\geq N_0$, donde $N_0=\text{máx}\{N_1,N_2,N_3\}$, tendríamos que $t_0\pm\frac{L}{2(N+1)}\in(t_0-\delta,t_0+\delta)\setminus\{t_0\}$ y que

$$\varepsilon_1<A(\gamma-1)-\varepsilon_2<|S_N f(t_0\pm\frac{L}{2(N+1)})-f(t_0\pm\frac{L}{2(N+1)})|$$

Que se contradice con (A.2), probando el resultado.

□

Corolario A.9. *Sea f un función $\mathscr{C}^1$ a trozos salvo en $t_0\in\mathbb{R}$. Entonces, para todo $\delta>0$ existe un $N\in\mathbb{N}$ suficientemente grande tal que:*

$$\sup\{|f(t)-S_N f(t)|:t\in(-\delta,\delta)\}\ \leq\ c(A)$$

Donde $c(A)$ es una constante dependiente del salto $A:=\frac{f(t_0^+)-f(t_0^-)}{2}$.