



FACULTADE DE MATEMÁTICAS

Traballo Fin de Grao

Aproximación Trigonométrica y Transformada rápida de Fourier. Algoritmo de Cooley-Tukey

Francisco Alborés Pérez

Miércoles 26 de Enero de 2022

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

GRAO DE MATEMÁTICAS

Traballo Fin de Grao

Aproximación Trigonométrica y
Transformada rápida de Fourier.
Algoritmo de Cooley-Tukey

Francisco Alborés Pérez

Miércoles 26 de Enero de 2022

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Trabajo propuesto

Área de Coñecemento: Matemática Aplicada
Título: Aproximación Trigonométrica y Transformada rápida de Fourier. Algoritmo de Cooley-Tukey
Breve descripción do contido
<p>La primera parte del trabajo está dedicada a la introducción de la Transformada de Fourier Discreta desde cuatro perspectivas distintas: aproximación de la Transformada de Fourier Continua, aproximación de los coeficientes de la serie de Fourier, cálculo de los polinomio trigonométricos de interpolación y aproximación óptima en el sentido de mínimos cuadrados discretos.</p> <p>La segunda parte del trabajo esta dedicada al cálculo efectivo de la Transformada de Fourier Discreta mediante la Transformada rápida de Fourier y el algoritmo de Cooley-Tukey. Se describe, implementa y se hacen estudios comparativos de dicho algoritmo.</p>
Recomendacións
Outras observacións

Índice general

Resumen	VIII
Introducción	XI
1. Transformada de Fourier Continua(FT) y Discreta(DFT).	1
1.1. Transformada de Fourier Continua(FT).	1
1.2. Transformada de Fourier Discreta(DFT)	2
2. Serie de Fourier y DFT	7
2.1. Aproximación óptima en el sentido de mínimos cuadrados	7
2.2. Aproximación óptima por polinomio trigonométricos	9
2.3. Cálculo de los coeficientes de Fourier y DFT	11
3. Interpolación trigonométrica y DFT	15
3.1. Interpolación trigonométrica	15
3.2. Interpolación por polinomios de fase	16
3.3. Coeficientes de polinomio trigonométrico de interpolación y DFT	20
4. Ajuste trigonométrico y DFT	25
4.1. Problema de ajuste trigonométrico	25
4.2. Ajuste trigonométrico y DFT	29
5. Transformada rápida de Fourier(FFT). Algoritmo de Cooley-Tukey	31
5.1. Transformada rápida de Fourier(FFT)	31
5.2. Algoritmo de Cooley-Tukey	35
6. Resultados	41
Anexo: Códigos en Matlab	45

Resumen

El objetivo principal de este trabajo es presentar, analizar, implementar y comparar el algoritmo de Cooley-Tukey para el cálculo de la Transformada Rápida de Fourier conocida por sus siglas en inglés: FFT (Fast Fourier Transform). Esto se hace en los Capítulos 5 y 6, mientras que los otros cuatro capítulos están dedicados a motivar y justificar la necesidad de la Transformada Rápida de Fourier como algoritmo del cálculo de la Transformada de Fourier Discreta (en inglés, DFT, Discret Fourier Transform) que es introducida desde cuatro puntos de vista distintos.

En el primer capítulo se introduce la DFT como una aproximación discreta de la Transformada de Fourier Continua mediante fórmulas de cuadratura para aproximar la integral que la define.

En el Capítulo 2 planteamos la aproximación de una función en el sentido de mínimos cuadrados por un polinomio trigonométrico de Fourier de grado m . Cuando m tiende a ∞ tendremos la conocida serie de Fourier de la función. La serie de Fourier es una potente herramienta de aproximación de funciones. Mostraremos su relación con la Transformada Discreta de Fourier a través del cálculo efectivo de los coeficientes de dicha serie utilizando fórmulas de cuadratura para el cálculo aproximado de las integrales que las definen.

La interpolación trigonométrica será el objeto de estudio del Capítulo 3. En él plantearemos la interpolación con polinomios trigonométricos y polinomios de fase. En ambas situaciones se expondrá la estrecha relación del cálculo de los coeficientes del polinomio con la Transformada de Fourier Discreta.

En el Capítulo 4 introducimos la DFT a través del cálculo efectivo de los coeficientes del polinomio trigonométrico de ajuste en el sentido de mínimos cuadrados discretos.

El Capítulo 5 llegamos a la parte central del trabajo, donde se expone un método efectivo para el cálculo de la Transformada Discreta: la Transformada Rápida de Fourier y, en particular, el algoritmo de Cooley-Tukey.

Por último, ocupamos el último capítulo con una recopilación de varios ejemplos de las situaciones expuestas en el trabajo poniendo en práctica el algoritmo desarrollado y mostrando sus beneficios frente a un cálculo directo de la Transformada de Fourier Discreta.

El trabajo termina con un anexo donde se recogen los códigos del algoritmo programado en Matlab.

Abstract

The main objective of this project is to present, analyze, implement and compare CooleyTukey's algorithm in order to calculate the Fast Fourier Transform (FFT), known in Spanish as Transformada Rápida de Fourier. This is calculated in chapters 5 and 6, while the other four chapters are dedicated to promote and justify the need of Fast Fourier Transform as the algorithm to calculate the Discrete Fourier Transform (DFT) (in Spanish: Transformada de Fourier Discreta) that is presented in four different points of view.

In the first chapter DFT is introduced as a discrete approximation of Continuous Fourier Transform with the use of quadrature formulas to approximate the integral that defines it.

In the Chapter 2 we propose the approximation of a function in the way of least-squares with grade m Fourier's trigonometric polynomial. When m tends to ∞ we obtain the known Fourier's series of the function. Fourier's series is a potent tool to approximate functions. We will show how it relates to the Discrete Fourier Transform with the effective calculation of the coefficients of the referred series by using quadrature formulas that define the approximated calculation of the integrals.

Trigonometric interpolation is the main object of study in Chapter 3, in which we will present the interpolation of trigonometric polynomial and phase polynomial. For both cases we will prove how the coefficients' calculation of the polynomial is strongly related to the Discrete Fourier Transformation.

In the Chapter 4 we introduce the DFT by the effective calculation of the coefficients of the trigonometrical polynomial fit in the way of discrete least-squares.

In the Chapter 5 we reach the main part of the project, where the effective method for the calculation of the Discrete Transform is exposed: the Fast Fourier Transform and, in particular, the Cooley-Tukey's algorithm.

In the end, we use the last chapter as a recompilation of different examples of the situations exposed in the project using the developed algorithm and showing its benefits in opposition to the direct calculation of the Discrete Fourier Transform.

The project concludes with an annex where the programmed Matlab codes of the algorithm are collected.

Introducción

Jean-Baptiste Joseph Fourier (1768-1830) fue un matemático y físico francés conocido por sus trabajos sobre la descomposición de funciones periódicas en series trigonométricas convergentes llamadas Series de Fourier, método con el cual consiguió resolver la ecuación del calor.

La transformada de Fourier o FT (por sus siglas del inglés Fourier transform) es una transformación matemática empleada para transformar señales entre el dominio del tiempo (o espacial) y el dominio de la frecuencia, que tiene muchas aplicaciones en la física y la ingeniería. En el caso de una función periódica en el tiempo (por ejemplo, un sonido musical continuo pero no necesariamente sinusoidal), la transformada de Fourier se puede simplificar para el cálculo de un conjunto discreto de amplitudes complejas, llamado coeficientes de las series de Fourier, que representan el espectro de frecuencia de la señal del dominio-tiempo original. Desde el punto de vista matemático es una aplicación del espacio $L^1(\mathbb{R})$ en si mismo.

La transformada discreta de Fourier o DFT (del inglés, discrete Fourier transform) es un tipo de transformada discreta utilizada en el análisis de Fourier. Transforma una función matemática en otra, obteniendo una representación en el dominio de la frecuencia, siendo la función original una función en el dominio del tiempo. La DFT requiere que la función de entrada sea una secuencia discreta y de duración finita. Dichas secuencias se suelen generar a partir del muestreo de una función continua, como puede ser la voz humana. Por estas razones, se dice que la DFT es una transformada de Fourier para análisis de señales de tiempo discreto y dominio finito. Las funciones sinusoidales base que surgen de la descomposición tienen las mismas propiedades. La entrada de la DFT es una secuencia finita de números reales o complejos, de modo que es ideal para procesar información almacenada en soportes digitales. En particular, la DFT se utiliza comúnmente en procesamiento digital de señales y otros campos relacionados dedicados a analizar las frecuencias que contiene una señal muestreada, también para resolver ecuaciones diferenciales parciales, y para llevar a cabo operaciones como convoluciones o multiplicaciones de grandes números enteros, cálculo de coeficientes en la serie de Fourier, aproximación en el sentido de mínimos cuadrados

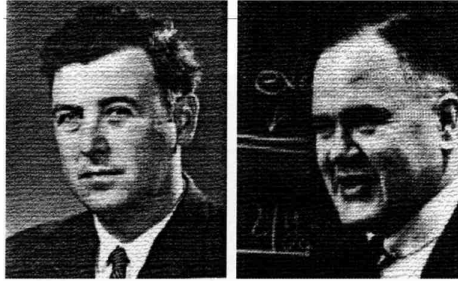


Figura 1: James William Cooley (Centro de Investigación Watson de IBM) y John Wilder Tukey (Universidad de Princeton), reconocidos como los autores del algoritmo de la Transformada Rápida de Fourier.

o ajuste trigonométrico. Desde el punto de vista matemático la DFT es una aplicación del espacio vectorial \mathbb{C}^N en si mismo que involucra del orden de $2N^2$ operaciones elementales.

Un factor muy importante para este tipo de aplicaciones es que la DFT puede ser calculada de forma eficiente en la práctica utilizando el algoritmo de la transformada rápida de Fourier o FFT (Fast Fourier Transform), rebajando el número de operaciones elementales a un orden de $N \log_2 N$.

La transformada rápida de Fourier es de importancia fundamental en el análisis matemático y ha sido objeto de numerosos estudios. La aparición de un algoritmo eficaz para esta operación fue un hito en la historia de la informática. El algoritmo de la transformada de Fourier Rápida (FFT), fue popularizado en 1965 por los matemáticos estadounidenses James William Cooley y John Wilder Tukey.

Fué en abril de 1965 cuando la revista *Mathematics Computation* publicaba un artículo de cinco páginas, más bien modesto y sin grandes pretensiones, bajo el título “An algorithm for machine calculation of complex Fourier series”. Inmediatamente, la comunidad científica relacionada con el procesamiento digital de señales estaba volcada sobre el algoritmo descrito en el artículo, que abría las puertas a un terreno lleno de posibilidades hasta ese momento prohibido debido a la imposibilidad de realizar directamente los cálculos que necesitaba la transformada de Fourier discreta por su elevado coste operacional con tiempos de cálculo inoperativos en la práctica.

En este trabajo de fin de grado rendimos un pequeño homenaje a estos dos investigadores describiendo el algoritmo, implementándolo en lenguaje Matlab y mostrando su comportamiento frente al cálculo directo en tres aplicaciones prácticas de índole matemática: el cálculo de los primeros coeficientes de Fourier de una función, el cálculo del polinomio trigonométrico de interpolación y el cálculo del polinomio de ajuste trigonométrico

Capítulo 1

Transformada de Fourier Continua(FT) y Discreta(DFT).

El objetivo de este capítulo es presentar una aproximación discreta de la Transformada de Fourier Continua, llamada así en honor al matemático Joseph Fourier (1768-1830), una importante herramienta en el área de las matemáticas aplicadas y la física.

Este capítulo se ha desarrollado siguiendo [1] (pág. 2-28)

1.1. Transformada de Fourier Continua(FT).

Para una función $f : x \in \mathbb{R} \rightarrow f(x) \in \mathbb{R}$ tal que $f \in L^1(\mathbb{R})$, definimos su Transformada de Fourier como la función $F : s \in \mathbb{R} \rightarrow F(s) \in \mathbb{R}$ dada por

$$F(s) = \int_{-\infty}^{\infty} f(x)e^{-i2\pi sx} dx, \quad (1.1)$$

donde i es la unidad imaginaria, $i = \sqrt{-1}$, y utilizamos la notación habitual

$$e^{ix} = \cos x + i \sin x, x \in \mathbb{R}.$$

Esta transformada admite una inversa denominada Transformada inversa de Fourier que está dada por

$$f(x) = \int_{-\infty}^{\infty} F(s)e^{i2\pi xs} ds. \quad (1.2)$$

La Transformada de Fourier permite examinar una función desde otro punto de vista al descomponerla en sus frecuencias constitutivas y almacenar la amplitud de cada componente en el dominio de la frecuencia representando así dicha función mediante su espectro de frecuencias. Tratamos de ilustrarlo con un ejemplo.

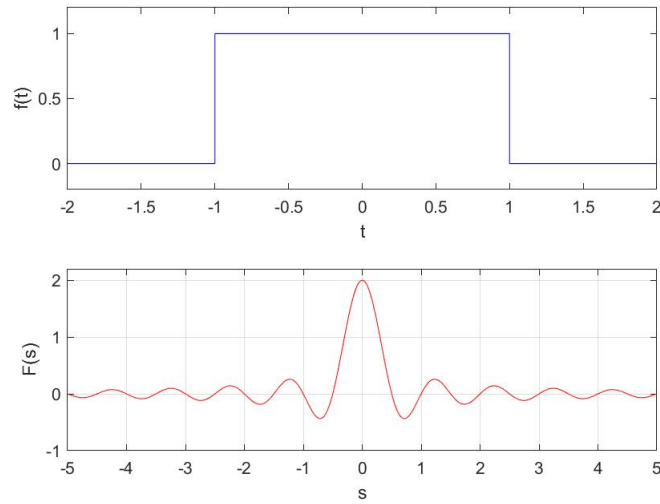


Figura 1.1: Representación de la función $f(t)$ y su Transformada de Fourier, $F(s)$.

Ejemplo 1.1. Consideramos la función:

$$f(t) = \begin{cases} A & , \quad |t| < T \\ \frac{A}{2} & , \quad t = \pm T \\ 0 & , \quad |t| > T \end{cases} \quad (1.3)$$

Cuya Transformada de Fourier viene dada por la función siguiente (véase [2], pág. 15-16):

$$F(s) = 2AT \frac{\sin(2\pi Ts)}{2\pi Ts}. \quad (1.4)$$

En la Figura 1.1 tenemos una representación de la función $f(t)$ y su transformada para el caso $A = 1$ y $T = 1$.

1.2. Transformada de Fourier Discreta (DFT)

Pretendemos introducir una aproximación discreta de la Transformada de Fourier continua que surgirá como una aproximación natural de dicha transformación continua.

La transformada de Fourier dada por la expresión (1.1) está definida en toda la recta por lo que, para hacer una aproximación discreta nos restringiremos a un intervalo finito real arbitrario

$$f : x \in (0, a) \rightarrow f(x) \in \mathbb{R},$$

y supondremos que fuera del intervalo $(0, a)$, $f(x) = 0$. De este modo, la Transformada de Fourier continua viene dada por la expresión

$$F(s) = \int_0^a f(x) e^{-i2\pi sx} dx, \quad (1.5)$$

la cual pretendemos aproximar numéricamente. Para ello dividimos el intervalo de integración en N subintervalos de longitud $\Delta x = \frac{a}{N}$ lo que nos proporciona el conjunto de nodos $x_k = k\Delta x, k = 0, \dots, N$.

Aplicando la regla del trapecio compuesta a la integral (1.5) obtenemos que

$$\int_0^a g(x) dx \approx \frac{a}{2N} (g(0) + 2 \sum_{k=1}^{N-1} g(x_k) + g(a)) \quad (1.6)$$

donde $g(x) = f(x)e^{-i2\pi sx}$. Añadiendo la condición $g(0) = g(a)$

$$F(s) = \int_0^a g(x) dx \approx \frac{a}{N} \sum_{k=0}^{N-1} g(x_k) = \frac{a}{N} \sum_{k=0}^{N-1} f(x_k) e^{-i2\pi sx_k} \quad (1.7)$$

La expresión anterior puede ser evaluada en cualquier valor de s por lo que, para aproximar F , tenemos que determinar cuántos y qué valores de s usar.

A través de la transformada, la función $f(x)$ está asociada al dominio de frecuencia donde se define $F(s)$, que denotaremos por $[0, \Omega]$, donde los nodos $x_k, k = 0, \dots, N-1$, se relacionan con N nodos de este dominio dados por $s_j = j\Delta s, j = 0, \dots, N-1$.

Analizando el integrando (1.5), la expresión

$$e^{-i2\pi sx} = \cos(2\pi sx) - i \sin(2\pi sx) \quad (1.8)$$

para un valor fijado de s , consiste en ondas (senos y cosenos) de período $1/s$ denominadas modos. Suponiendo que estos modos tengan un número entero de períodos en $[0, a]$ bien ajustado en el intervalo, consideremos el modo con mayor período posible. Dicho modo tiene, por tanto, una frecuencia de $1/a$ la cual es la más baja asociada al intervalo $[0, a]$.

Denotemos esta unidad de frecuencia por $\Delta s = 1/a$ que será la distancia entre los nodos del dominio de frecuencia $[0, \Omega]$ los cuales vienen dados por $s_j = j\Delta s = j/a, j = 0, \dots, N-1$. Obteniendo la relación $\Omega = N\Delta s = N/a$, es decir, $a\Omega = N$. Obsérvese que la última igualdad afirma que el dominio de la variable x y el dominio de frecuencia varían de forma inversa.

Por otro lado, dado que $\Delta x = a/N$ y $\Delta s = 1/a$, se tiene: $\Delta x \Delta s = 1/N$

4CAPÍTULO 1. TRANSFORMADA DE FOURIER CONTINUA (FT) Y DISCRETA (DFT).

Es decir, hemos obtenido las relaciones siguientes

$$a\Omega = N, \quad (1.9)$$

$$\Delta x \Delta s = \frac{1}{N}, \quad (1.10)$$

con la cuales podemos volver sobre la expresión (1.7) y aproximar numéricamente $F(s)$ en los nodos $s_j, j = 0, \dots, N - 1$:

$$F(s_j) \approx \frac{a}{N} \sum_{k=0}^{N-1} f(x_k) e^{-i2\pi s_j x_k}, \quad j = 0, \dots, N - 1. \quad (1.11)$$

Ahora bien, teniendo en cuenta las relaciones (1.9) y (1.10), tenemos que

$$s_j x_k = (j\Delta s)(k\Delta x) = \frac{j}{a} \frac{ka}{N} = \frac{jk}{N},$$

obteniendo la aproximación siguiente

$$F(s_j) \approx \frac{a}{N} \sum_{k=0}^{N-1} f(x_k) e^{-\frac{i2\pi kj}{N}} = a \underbrace{\frac{1}{N} \sum_{k=0}^{N-1} f(x_k) e^{-\frac{i2\pi kj}{N}}}_{F_j}, \quad j = 0, \dots, N - 1. \quad (1.12)$$

Por tanto, la aproximación de la Transformada Continua de Fourier está determinada por los valores $F_j, j = 0, \dots, N - 1$. Dicha serie de números complejos es lo que se conoce como **Transformada de Fourier Discreta** (en inglés, Discret Fourier Transform ó DFT) de los números $f(x_k), k = 0, \dots, N - 1$.

Si adoptamos la siguiente notación:

$$w_k = e^{\frac{i2\pi k}{N}}, \quad k = 0, \dots, N - 1,$$

la Transformada de Fourier Discreta queda expresada como

$$F_j = \frac{1}{N} \sum_{k=0}^{N-1} f_k w_k^{-j} = \frac{1}{N} \sum_{k=0}^{N-1} f_k e^{-2\pi i k j / N}, \quad j = 0, \dots, N - 1, \quad (1.13)$$

donde $f_k = f(x_k)$. Esto nos lleva a la siguiente definición formal.

Definición 1.2. Sea N un número entero, $N \geq 1$, y $f_k, k = 0, \dots, N - 1$, un conjunto de números complejos. Definimos la Transformada de Fourier Discreta del conjunto $f_k, k = 0, \dots, N - 1$ como el conjunto de números $F_j, j = 0, \dots, N - 1$, dados por:

$$F_j = \frac{1}{N} \sum_{k=0}^{N-1} f_k w_k^{-j} = \frac{1}{N} \sum_{k=0}^{N-1} f_k e^{-\frac{i2\pi kj}{N}}, \quad j = 0, \dots, N - 1. \quad (1.14)$$

Se denotará $F_j = D\{f_k\}_j, j = 0, \dots, N - 1$.

Como veremos en los siguientes capítulos, el cálculo directo de los N números complejos F_j cuando N es un número de puntos considerable, supone un coste operacional enorme. Afortunadamente existen estrategias de cálculo que permiten calcular dichas cantidades en tiempos de cálculo aceptables. Son las estrategias de la Transformada Rápida de Fourier (en inglés, FFT) y un ejemplo es el algoritmo de Cooley-Tukey.

6 *CAPÍTULO 1. TRANSFORMADA DE FOURIER CONTINUA (FT) Y DISCRETA (DFT).*

Capítulo 2

Serie de Fourier y DFT

En este capítulo motivamos la Transformada de Fourier Discreta a través de la aproximación de los coeficientes de la Serie de Fourier de una función.

Comenzaremos introduciendo el concepto de polinomio de Fourier como la solución del problema de aproximación trigonométrica por mínimos cuadrados. Cuando el grado del polinomio tiende a $+\infty$ se obtiene la serie de Fourier, herramienta matemática de enorme importancia en la aproximación de funciones.

El contenido de este capítulo está inspirado en [7] (págs. 165-168 y 189-199).

2.1. Aproximación óptima en el sentido de mínimos cuadrados

Sea $(a, b) \subset \mathbb{R}$ y $L^2(a, b)$ el espacio de Hilbert

$$L^2(a, b) = \{f : (a, b) \rightarrow \mathbb{R} : f \text{ medible, } \int_a^b |f(x)|^2 dx < +\infty\}.$$

dotado del producto escalar

$$(f, g) = \int_a^b f(x)g(x)dx,$$

y la norma

$$\|f\| = \left(\int_a^b f(x)^2 dx \right)^{\frac{1}{2}}.$$

Consideremos un subespacio de dimensión finita $S_n \subset L^2(a, b)$ con base $\{\varphi_0, \dots, \varphi_n\}$. Definimos el problema de mínimos cuadrado de la siguiente forma:

Dada una función $f \in L^2(a, b)$ encontrar $p \in S_n$ de forma que:

$$\|f - p\| = \min_{q \in S_n} \|f - q\|.$$

A la función $p \in S_n$ la denominamos aproximación óptima de f en S_n en el sentido de mínimos cuadrados.

Veamos ahora que el anterior problema es equivalente a la minimización de un funcional cuadrático en \mathbb{R}^{n+1} . En efecto, basta escribir $p, q \in S_n$ de la forma:

$$p = a_0\varphi_0 + \dots + a_n\varphi_n, \quad q = \alpha_0\varphi_0 + \dots + \alpha_n\varphi_n.$$

Entonces:

$$\begin{aligned} \frac{1}{2}\|f - q\|^2 &= \frac{1}{2} \left(f - \sum_{i=0}^n \alpha_i \varphi_i, f - \sum_{i=0}^n \alpha_i \varphi_i \right) \\ &= \frac{1}{2}\|f\|^2 - \sum_{i=0}^n \alpha_i (f, \varphi_i) + \frac{1}{2} \sum_{i,j=0}^n \alpha_i \alpha_j (\varphi_i, \varphi_j) \\ &= \frac{1}{2}\|f\|^2 - v^T b + \frac{1}{2} v^T A v := J(v). \end{aligned}$$

donde

$$v = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix}, \quad b = \begin{pmatrix} (f, \varphi_0) \\ (f, \varphi_1) \\ \vdots \\ (f, \varphi_n) \end{pmatrix}, \quad A = \begin{pmatrix} (\varphi_0, \varphi_0) & (\varphi_0, \varphi_1) & \cdots & (\varphi_0, \varphi_n) \\ (\varphi_1, \varphi_0) & (\varphi_1, \varphi_1) & \cdots & (\varphi_1, \varphi_n) \\ \vdots & \vdots & \ddots & \vdots \\ (\varphi_n, \varphi_0) & (\varphi_n, \varphi_1) & \cdots & (\varphi_n, \varphi_n) \end{pmatrix}.$$

Así pues, el elemento $p \in S_n$ es solución del problema de mínimos cuadrados si y solo si $u = (a_0, \dots, a_n)^T$ es solución del siguiente problema de minimización de un funcional cuadrático:

$$u \in \mathbb{R}^{n+1}, \quad J(u) = \min_{v \in \mathbb{R}^{n+1}} J(v).$$

Además, la matriz A es simétrica y definida positiva: dado $v = (\alpha_0, \dots, \alpha_n) \neq 0$, se tiene que

$$v^T A v = \sum_{i,j=0}^n \alpha_i (\varphi_i, \varphi_j) \alpha_j = \left(\sum_{i=0}^n \alpha_i \varphi_i, \sum_{j=0}^n \alpha_j \varphi_j \right) = \left\| \sum_{i=0}^n \alpha_i \varphi_i \right\|^2 > 0.$$

Por tanto, el problema de minimización del funcional cuadrático tiene una única solución u que coincide con la única solución del sistema lineal:

$$A u = b,$$

el cual define las denominadas ecuaciones normales.

Con todo esto, queda justificado el siguiente teorema:

Teorema 2.1. *El problema de aproximación óptima en el sentido de mínimos cuadrados tiene una única solución $p \in S_n$.*

2.2. Aproximación óptima por polinomio trigonométricos

A nosotros nos interesa de manera especial la aproximación óptima en el sentido de mínimos cuadrados mediante polinomios trigonométricos. Esta aproximación está específicamente indicada para fenómenos o datos periódicos. Para ello será conveniente utilizar combinaciones lineales finitas de las funciones trigonométricas básicas:

$$\left\{ \frac{1}{2}, \cos(x), \sin(x), \cos(2x), \sin(2x), \dots, \cos(mx), \sin(mx), \dots \right\}.$$

Tales funciones son 2π -periódicas con las que aproximaremos funciones en $L^2(a, a + 2\pi)$. Sin pérdida de generalidad, tomaremos como intervalo de referencia el intervalo $(0, 2\pi)$, aunque puede ser extrapolado a cualquier intervalo arbitrario (véase [7] pág.197-199).

Por tanto, consideramos el subespacio vectorial finito de $L^2(0, 2\pi)$ de polinomios trigonométricos de grado menor o igual que m , T_m :

$$T_m = \left\langle \frac{1}{2}, \cos(x), \sin(x), \cos(2x), \sin(2x), \dots, \cos(mx), \sin(mx) \right\rangle,$$

del Teorema 2.1, se tiene el siguiente resultado:

Teorema 2.2. *Para toda función $f \in L^2(0, 2\pi)$, existe un único polinomio trigonométrico de grado $\leq m$, $s_m \in T_m$, aproximación óptima de f en el sentido de mínimos cuadrados:*

$$s_m \in T_m, \|f - s_m\| = \min_{\tau \in T_m} \|f - \tau\|.$$

El polinomio $s_m \in T_m$ se denomina *polinomio de Fourier* de f de grado m y viene dado por la siguiente expresión:

$$s_m = \frac{a_0}{2} + \sum_{j=1}^m (a_j \cos(jx) + b_j \sin(jx)), \quad (2.1)$$

donde los coeficientes del polinomio, denominados coeficientes de Fourier de f , son de la forma:

$$a_0 = \frac{1}{\pi} \int_0^{2\pi} f(x) dx \quad (2.2)$$

$$a_j = \frac{1}{\pi} \int_0^{2\pi} f(x) \cos(jx) dx, j = 1, 2, \dots, m. \quad (2.3)$$

$$b_j = \frac{1}{\pi} \int_0^{2\pi} f(x) \sin(jx) dx, j = 1, 2, \dots, m. \quad (2.4)$$

Definición 2.3. La serie funcional cuya suma parcial m -enésima es s_m se denomina serie de Fourier de f . Por tanto, de (2.1) se deduce que la serie de Fourier se escribe formalmente como:

$$\lim_{m \rightarrow \infty} s_m(x) = \frac{a_0}{2} + \sum_{j=1}^{\infty} a_j \cos(jx) + b_j \sin(jx). \quad (2.5)$$

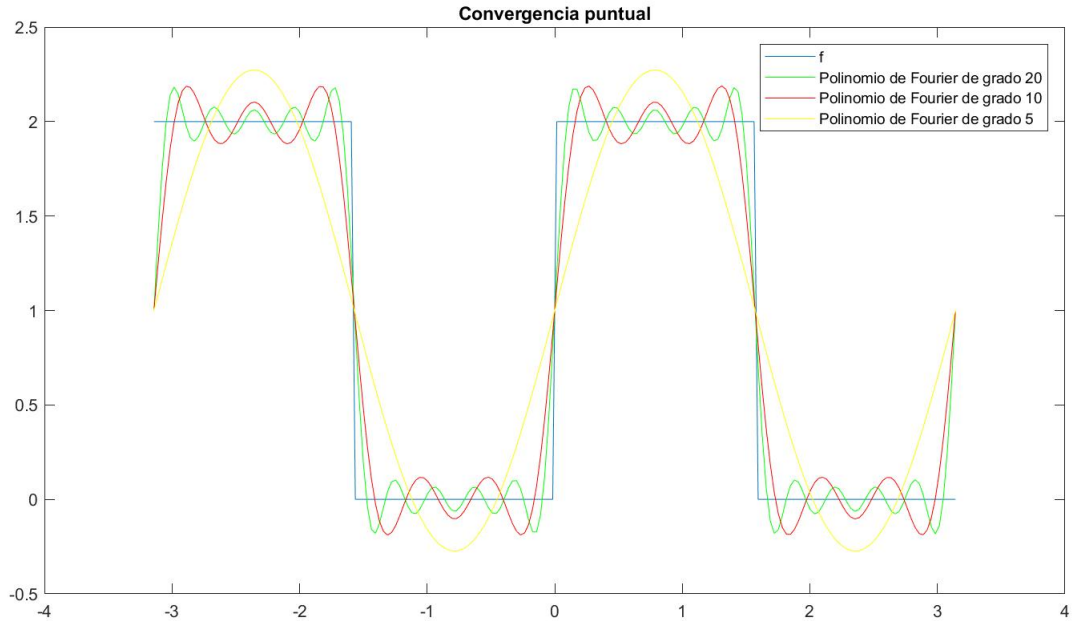


Figura 2.1: Convergencia puntual de la Serie de Fourier para la función $f(x) = 1 + \text{sgn}(\sin(2x))$.

Utilizando el hecho de que es sistema

$$\left\{ \frac{1}{2}, \cos(x), \sin(x), \cos(2x), \sin(2x), \dots, \cos(mx), \sin(mx), \dots \right\},$$

es una base ortogonal numerable en $L^2(0, 2\pi)$ se deduce el siguiente resultado fundamental del Análisis de Fourier (uno de los más importantes del Análisis Matemático):

Teorema 2.4. Para toda función $f \in L^2(0, 2\pi)$ la serie de Fourier asociada a f

$$\frac{a_0}{2} + \sum_{j=1}^{\infty} a_j \cos(jx) + b_j \sin(jx),$$

converge a la función f en $L^2(0, 2\pi)$: $f = \lim_{m \rightarrow +\infty} s_m(x)$ en $L^2(0, 2\pi)$.

Además:

$$\|f\|^2 = \int_0^{2\pi} f^2(x) dx = \lim_{m \rightarrow +\infty} \|s_m\|^2 = \pi \left[\frac{a_0^2}{2} + \sum_{j=1}^{\infty} (a_j^2 + b_j^2) \right].$$

La convergencia anterior es en $L^2(0, 2\pi)$. Existen resultados de convergencia puntual y uniforme como el que recordamos a continuación.

Teorema 2.5. Sea $f \in L^2(0, 2\pi)$ una función continua a trozos. Entonces, su serie de Fourier converge puntualmente en todos los puntos de $[0, 2\pi]$ de la siguiente forma:

- Si f es continua en $x \in (0, 2\pi)$:

$$\lim_{m \rightarrow \infty} s_m(x) = \frac{a_0}{2} + \sum_{j=1}^{\infty} a_j \cos(jx) + b_j \sin(jx) = f(x).$$

- Si f tiene una discontinuidad en $x \in (0, 2\pi)$:

$$\lim_{m \rightarrow \infty} s_m(x) = \frac{a_0}{2} + \sum_{j=1}^{\infty} a_j \cos(jx) + b_j \sin(jx) = \frac{1}{2}[f(x^+) + f(x^-)].$$

- En $x = 0$ y $x = 2\pi$ tenemos que:

$$\lim_{m \rightarrow \infty} s_m(0) = \lim_{m \rightarrow \infty} s_m(2\pi) = \frac{1}{2}[f(0^+) + f(2\pi^-)].$$

Además, si f es continua en $[0, 2\pi]$ y periódica, entonces la sucesión de polinomios trigonométricos $(\sigma_n)_{n \geq 0}$, con $\sigma_n = \frac{1}{n}[s_0 + s_1 + \dots + s_{n-1}]$, $n \geq 1$, converge uniformemente a f en $[0, 2\pi]$.

Demostración. Véase [3] (págs., 303-305). □

La Figura 2.1 ilustra la convergencia puntual de la serie de Fourier de una función escalonada, donde vemos gráficamente la aplicación del Teorema 2.5. Vemos a continuación cómo calculamos los coeficientes del polinomio de Fourier s_m .

2.3. Cálculo de los coeficientes de Fourier y DFT

Veremos en esta sección cómo aparece la Transformada de Fourier Discreta cuando intentamos el cálculo efectivo de los coeficientes de Fourier. Recordemos que dichos coeficientes están dados por las siguientes expresiones:

$$a_j = \frac{1}{\pi} \int_0^{2\pi} f(x) \cos(jx) dx, j = 1, 2, \dots, m. \quad (2.6)$$

$$b_j = \frac{1}{\pi} \int_0^{2\pi} f(x) \sin(jx) dx, j = 1, 2, \dots, m. \quad (2.7)$$

El cálculo efectivo de los coeficientes se realiza mediante la fórmula de cuadratura de rectángulos compuesta y utilizando N puntos igualmente espaciados:

$$x_k = \frac{2k\pi}{N}, k = 0, \dots, N - 1. \quad (2.8)$$

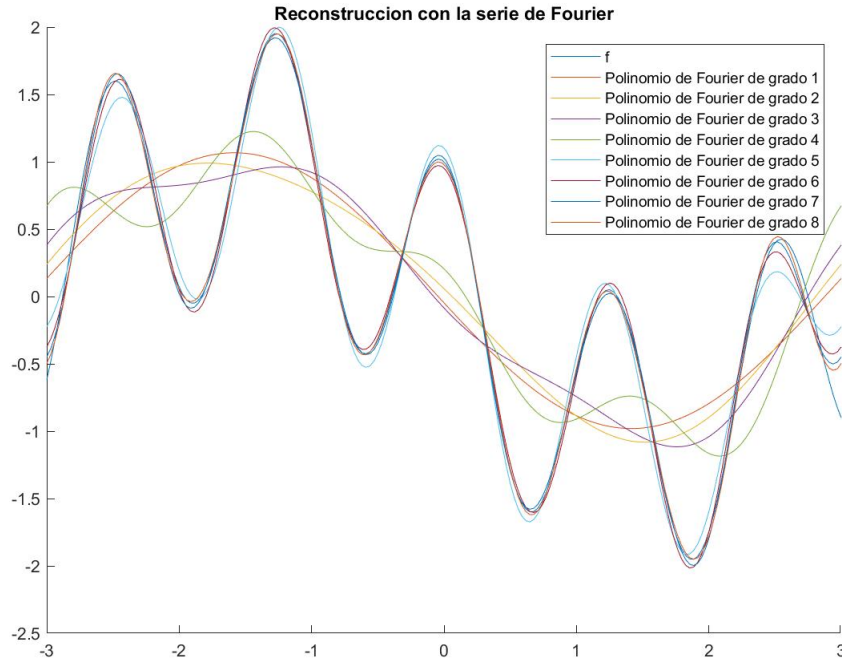


Figura 2.2: Reconstrucción a partir de la Serie de Fourier de la función $f(x) = \cos(5x) - \text{sen}(x)$.

para aproximar las integrales (2.6) y (2.7).

De esta forma aproximamos los coeficientes a_j y b_j por $A_j^{(N)}$ y $B_j^{(N)}$, respectivamente, de la siguiente forma:

$$a_j = \frac{1}{\pi} \int_0^{2\pi} f(x) \cos(jx) \simeq A_j^{(N)} = \frac{2}{N} \sum_{k=0}^{N-1} f_k \cos(jx_k) = \frac{2}{N} \sum_{k=0}^{N-1} f_k \cos\left(\frac{2jk\pi}{N}\right), \quad (2.9)$$

$$b_j = \frac{1}{\pi} \int_0^{2\pi} f(x) \sin(jx) \simeq B_j^{(N)} = \frac{2}{N} \sum_{k=0}^{N-1} f_k \sin(jx_k) = \frac{2}{N} \sum_{k=0}^{N-1} f_k \sin\left(\frac{2jk\pi}{N}\right), \quad (2.10)$$

donde $f_k = f(x_k)$, $j = 0, \dots, N-1$. Por tanto, definiendo los siguientes números complejos:

$$\beta_j = \frac{1}{N} \sum_{k=0}^{N-1} f_k e^{-ijx_k} = \frac{1}{N} \sum_{k=0}^{N-1} f_k e^{-\frac{2\pi i k j}{N}}, \quad j = 0, \dots, N-1, \quad (2.11)$$

se cumplen las siguientes relaciones:

$$A_0^{(N)} = 2\beta_0,$$

$$A_j^{(N)} = \beta_j + \beta_{N-j}, j = 1, \dots, m, m \leq \left\lfloor \frac{N}{2} \right\rfloor,$$
$$B_j^{(N)} = i(\beta_j - \beta_{N-j}), j = 1, \dots, m, m \leq \left\lfloor \frac{N}{2} \right\rfloor.$$

Concluimos así que para el cálculo de los coeficientes de Fourier necesitamos realizar los cálculos (2.11) que no es más que la Transformada de Fourier Discreta del conjunto $f_k, k = 0, \dots, N - 1$. La figura 2.2 muestra que para una buena aproximación es necesario calcular muchos coeficientes y para cada uno el número de puntos N de la fórmula de cuadratura debe ser muy elevado. Por tanto, eficiente y rápido de los β_j es necesario.

Capítulo 3

Interpolación trigonométrica y DFT

En este capítulo introducimos la interpolación trigonométrica y nos centraremos en poner de manifiesto la relación entre el cálculo del polinomio trigonométrico de interpolación y la Transformada de Fourier Discreta.

Este capítulo está basado en [6] (págs. 159-170) y en [4] (págs. 401-414).

3.1. Interpolación trigonométrica

Para la interpolación de fenómenos periódicos es conveniente el uso de polinomios trigonométricos, esto es, la combinación lineal de las funciones trigonométricas básicas:

$$\cos(kx) \text{ y } \sin(kx), k = 1, 2, 3, \dots, n.$$

Supondremos que nuestra función objeto de interpolación tiene período $T = 2\pi$ y la restringimos al intervalo $[0, 2\pi]$ pues, en caso de que dicha función este definida en un intervalo arbitrario $[a, b]$, podemos efectuar un cambio de variable de las funciones trigonométricas básicas y trabajar en el intervalo $[0, 2\pi]$.

Definición 3.1. i) Para $n = 2m$, $m \in \mathbb{N}$, $m \geq 0$, denotamos por $T_n = T_{2m}$ el espacio de polinomios trigonométricos de la forma:

$$\tau(x) = \frac{a_0}{2} + \sum_{k=1}^m (a_k \cos(kx) + b_k \sin(kx)), \quad a_k, b_k \in \mathbb{R}.$$

ii) Para $n = 2m - 1$, $m \in \mathbb{N}$, $m \geq 1$, denotamos por $T_n = T_{2m-1}$ el espacio de polinomios trigonométricos de la forma:

$$\frac{a_0}{2} + \sum_{k=1}^{m-1} (a_k \cos(kx) + b_k \sin(kx)) + \frac{a_m}{2} \cos(mx), \quad a_k, b_k \in \mathbb{R}.$$

Observación 3.2. Se notará que T_{2m} , es el espacio de polinomios trigonométricos introducido en el capítulo anterior. Es muy fácil probar que:

1. El conjunto $\{\frac{1}{2}, \cos(x), \sin(x), \cos(2x), \sin(2x), \dots, \cos(mx), \sin(mx)\}$ es una base del espacio $T_n = T_{2m}$, cuya dimensión es por tanto $n + 1 = 2m + 1$.
2. El conjunto $\{\frac{1}{2}, \cos(x), \sin(x), \cos(2x), \sin(2x), \dots, \cos((m-1)x), \sin((m-1)x), \frac{1}{2} \cos(mx)\}$ es una base del espacio $T_n = T_{2m-1}$, cuya dimensión es por tanto $n + 1 = 2m$.

El problema general de interpolación trigonométrica se plantea de la forma siguiente:

Dados N puntos distintos x_0, x_1, \dots, x_{N-1} en el intervalo $[0, 2\pi)$ y N valores reales $f(x_0), f(x_1), \dots, f(x_{N-1})$, encontrar un polinomio trigonométrico tal que

$$\tau(x_k) = f(x_k) \quad , k = 0, \dots, N - 1.$$

Si $N - 1 = 2M$, $M \geq 0$, es relativamente fácil establecer que $\tau \in T_{N-1}$ existe y es único. Para el caso N impar el problema es más delicado. Afortunadamente para el caso más importante en la práctica el problema se resuelve cualquiera que sea la paridad de N . Este caso corresponde a tener los nodos igualmente espaciados:

$$x_k = \frac{2k\pi}{N} \quad , k = 0, \dots, N - 1. \quad (3.1)$$

Con este tipo de particiones del intervalo vamos a transformar el problema de interpolación trigonométrica en un problema de interpolación compleja utilizando polinomios de fase y demostrar la unicidad de la solución cualquiera que sea $N \geq 1$.

3.2. Interpolación por polinomios de fase

Definición 3.3. Un **polinomio de fase** o **polinomio exponencial** de grado menor o igual que $N - 1$ con $N \geq 1$, es una función $p : [0, 2\pi] \mapsto \mathbb{C}$ tal que

$$p(x) = \beta_0 + \beta_1 e^{ix} + \dots + \beta_{N-1} e^{i(N-1)x} = \sum_{k=0}^{N-1} \beta_k e^{ikx},$$

donde $\beta_i \in \mathbb{C}$, $i = 0, \dots, N - 1$,

Denotamos por $\mathcal{F}_{N-1}(\mathbb{C})$ el espacio de polinomios exponenciales de grado $\leq N - 1$. El problema de interpolación compleja queda planteado de la siguiente manera:

Dados n valores $f_0, f_1, \dots, f_{N-1} \in \mathbb{R}$, encontrar un polinomio de fase, p , tal que:

$$p \in \mathcal{F}_{N-1}(\mathbb{C}) \quad , p(x_k) = f_k \quad , k = 0, 1, \dots, N - 1.$$

Naturalmente la siguiente cuestión que debemos plantearnos concierne a la existencia y unicidad de la solución de este problema. El siguiente teorema apunta en esta dirección:

Teorema 3.4. *El problema de la interpolación por polinomios de fase tiene una y solo una solución*

$$p \in \mathcal{F}_{N-1}(\mathbb{C}), p(x) = \sum_{k=0}^{N-1} \beta_k e^{ikx}, \quad (3.2)$$

donde los coeficientes $\beta_k \in \mathbb{C}, k = 0, \dots, N-1$, son la única solución del sistema lineal complejo siguiente:

$$\begin{pmatrix} 1 & w_0 & w_0^2 & \cdots & w_0^{N-1} \\ 1 & w_1 & w_1^2 & \cdots & w_1^{N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w_{N-1} & w_{N-1}^2 & \cdots & w_{N-1}^{N-1} \end{pmatrix} \cdot \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{N-1} \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{N-1} \end{pmatrix} \quad (3.3)$$

siendo $w_k = e^{ix_k} = e^{i2\pi k/N}, k = 0, \dots, N-1$.

Demostración. Nuestro objetivo es encontrar un polinomio de fase de la forma (3.2) tal que:

$$p(x_j) = f_j, j = 0, \dots, N-1.$$

Por lo tanto,

$$p(x_j) = \sum_{k=0}^{N-1} \beta_k e^{ikx_j} = f_j,$$

expresión que se corresponde con el sistema (3.3). Puesto que la matriz de coeficientes es de tipo Vandermonde con determinante $\prod_{0 \leq j < i \leq N-1} (w_i - w_j) \neq 0$, pues $w_i \neq w_j$, el sistema lineal tiene una única solución. \square

Una vez establecida la interpolación mediante polinomios de fase, se relacionará con la interpolación trigonométrica mediante el siguiente teorema:

Teorema 3.5. *Sea $(x_k, f_k), k = 0, \dots, N-1$, un conjunto de puntos donde $x_k = \frac{2k\pi}{N}$.*

Sea p el único polinomio de fase $p \in \mathcal{F}_{N-1}(\mathbb{C})$ de la forma $p(x) = \sum_{k=0}^{N-1} \beta_k e^{ikx}$ tal que $p(x_k) = f_k, k = 0, \dots, N-1$. Entonces, un polinomio trigonométrico $\tau \in T_{N-1}$ verifica que $\tau(x_k) = f_k, k = 0, \dots, N-1$, si y solo si

- $N = 2m + 1$

$$\tau(x) = \frac{a_0}{2} + \sum_{k=1}^m (a_k \cos(kx) + b_k \sin(kx)),$$

$$a_0 = 2\beta_0, a_j = \beta_j + \beta_{N-j}, b_j = i(\beta_j - \beta_{N-j}), j = 1, \dots, m. \quad (3.4)$$

■ $N = 2m$

$$\tau(x) = \frac{a_0}{2} + \sum_{k=1}^m (a_k \cos(kx) + b_k \sin(kx)) + \frac{a_m}{2} \cos(mx),$$

$$a_0 = 2\beta_0, a_j = \beta_j + \beta_{N-j}, b_j = i(\beta_j - \beta_{N-j}), j = 1, \dots, m-1, a_m = 2\beta_m. \quad (3.5)$$

Demostración. Probaremos que si $\tau(x_k) = f_k, k = 0, \dots, N-1$, entonces los coeficientes vienen dados por (3.4).

Observemos en primer lugar que

$$e^{-hix_k} = e^{\frac{2\pi i(N-h)k}{N}} = e^{(N-h)ix_k}$$

por tanto,

$$\cos(hx_k) = \frac{e^{hix_k} + e^{(N-h)ix_k}}{2}, \quad \sin(hx_k) = \frac{e^{hix_k} - e^{(N-h)ix_k}}{2i}. \quad (3.6)$$

Ahora bien, para el caso $N = 2m+1$, la expresión del polinomio trigonométrico para los valores $(x_k, f_k), k = 0, \dots, N-1$ viene dada por:

$$\tau(x_k) = \frac{a_0}{2} + \sum_{j=1}^m (a_j \cos(jx_k) + b_j \sin(jx_k)), k = 0, \dots, N-1.$$

Como $\tau(x_k) = f_k = p(x_k)$, tenemos que

$$\frac{a_0}{2} + \sum_{j=1}^m (a_j \cos(jx_k) + b_j \sin(jx_k)) = \beta_0 + \sum_{j=1}^m \beta_j e^{ijx_k} + \sum_{j=m+1}^{2m} \beta_j e^{ijx_k}.$$

Sustituyendo por las expresiones (3.6):

$$\frac{a_0}{2} + \sum_{j=1}^m \left(a_j \left(\frac{e^{hix_k} + e^{(N-h)ix_k}}{2} \right) + b_j \left(\frac{e^{hix_k} - e^{(N-h)ix_k}}{2i} \right) \right) = \beta_0 + \sum_{j=1}^m \beta_j e^{ijx_k} + \sum_{j=1}^m \beta_j e^{i(N-j)x_k}.$$

De donde se deducen las expresiones (3.4).

El caso $N = 2m$ es análogo.

Supongamos ahora el polinomio con coeficientes dados por (3.4).

Considerando el caso $N = 2m + 1$ y las igualdades (3.4), tenemos que:

$$\begin{aligned}
\tau(x_k) &= \frac{a_0}{2} + \sum_{j=1}^m (a_j \cos(jx_k) + b_j \sin(jx_k)) \\
&= \beta_0 + \sum_{j=1}^m [(\beta_j + \beta_{N-j} \cos(jx_k) + i(\beta_j + \beta_{N-j}) \sin(jx_k))] \\
&= \beta_0 + \sum_{j=1}^m \beta_j (\cos(jx_k) + i \sin(jx_k)) + \sum_{j=1}^m \beta_{N-j} (\cos(jx_k) - i \sin(jx_k)) \\
&= \beta_0 + \sum_{j=1}^m \beta_j e^{ijx_k} + \sum_{j=1}^m \beta_{N-j} e^{-ijx_k} \\
&= \beta_0 + \sum_{j=1}^m \beta_j e^{ijx_k} + \sum_{j=1}^m \beta_{N-j} e^{-i(N-j)x_k} \\
&= \beta_0 + \sum_{j=1}^m \beta_j e^{ijx_k} + \sum_{j=m+1}^{N-1} \beta_j e^{-ijx_k} \\
&= \sum_{j=0}^{N-1} \beta_j e^{ijx_k} = p(x_k) = f_k.
\end{aligned}$$

Para el caso $N = 2m$ se hace de forma análoga. □

Mediante este teorema quedan establecidas las relaciones entre los coeficientes del polinomio trigonométrico y del polinomio de fase:

- $N = 2m + 1$

$$\beta_0 = \frac{a_0}{2}, \beta_j = \frac{1}{2}(a_j - ib_j), \beta_{N-j} = \frac{1}{2}(a_j + ib_j) \quad j = 1, \dots, m,$$

$$a_0 = 2\beta_0, a_j = \beta_j + \beta_{N-j}, b_j = i(\beta_j - \beta_{N-j}), \quad j = 1, \dots, m.$$

- $N = 2m$

$$\beta_0 = \frac{a_0}{2}, \beta_j = \frac{1}{2}(a_j - ib_j), \beta_{N-j} = \frac{1}{2}(a_j + ib_j) \quad j = 1, \dots, m-1, \beta_m = \frac{a_m}{2},$$

$$a_0 = 2\beta_0, a_j = \beta_j + \beta_{N-j}, b_j = i(\beta_j - \beta_{N-j}), \quad j = 1, \dots, m, a_m = 2\beta_m.$$

Corolario 3.6. *El problema de interpolación trigonométrica para el conjunto de valores (x_k, f_k) donde $x_k = \frac{2k\pi}{N}$, $k = 0, \dots, N-1$ tiene una y solo una solución dada por, en el caso $N = 2m + 1$, la expresión (3.4) y, en el caso $N = 2m$, por la expresión (3.5).*

3.3. Coeficientes de polinomio trigonométrico de interpolación y DFT

Antes de relacionar los coeficientes del polinomio de fase interpolador con la Transformada de Fourier Discreta introduzcamos la siguiente herramienta, el producto escalar euclidiano en \mathbb{C}^N :

Definición 3.7. Definimos el producto escalar euclidiano en \mathbb{C}^N como

$$(u, v) = \sum_{i=1}^N u_i \bar{v}_i, \quad u = \{u_i\}_{i=1}^N, v = \{v_i\}_{i=1}^N \in \mathbb{C}^N,$$

donde \bar{v}_i es el complejo conjugado de v_i .

El siguiente teorema nos será de utilidad:

Teorema 3.8 (Ortogonalidad de la exponencial compleja). *Sean j y k números enteros y N un número entero positivo. Entonces*

$$\sum_{n=0}^{N-1} e^{\frac{i2\pi nj}{N}} e^{-\frac{i2\pi nk}{N}} = N\delta_N(j-k), \quad (3.7)$$

donde $\delta_N(m) = \begin{cases} 1 & \text{si } m = 0 \text{ ó } m = N \\ 0 & \text{en otro caso.} \end{cases}$.

Demostración. Observemos que

$$\sum_{n=0}^{N-1} e^{\frac{i2\pi nj}{N}} e^{-\frac{i2\pi nk}{N}} = \sum_{n=0}^{N-1} e^{\frac{i2\pi n(j-k)}{N}}.$$

- Para el caso $j = k$, es trivial.
- Probemos el caso $j \neq k$.

Sea $z = e^{\frac{i2\pi(j-k)}{N}}$. Se verifica que $z^N = 1$ pues:

$$z^N = e^{i2\pi(j-k)} = \cos(2\pi(j-k)) + i \sin(2\pi(j-k)) = 1.$$

Puesto que $z \neq 0$ ($j \neq k$) y

$$z^N - 1 = (z - 1) \sum_{n=0}^{N-1} z^n,$$

se tiene que $\sum_{n=0}^{N-1} z^n = 0$.

Tenemos entonces los dos casos siguientes:

3.3. COEFICIENTES DE POLINOMIO TRIGONOMÉTRICO DE INTERPOLACIÓN Y DFT21

- Si $j - k$ no es un múltiplo de N . Entonces $z \neq 1$ y

$$\sum_{n=0}^{N-1} z^n = \sum_{n=0}^{N-1} e^{\frac{i2\pi n(j-k)}{N}} = 0.$$

- Si $j - k$ es un múltiplo de N . Entonces $e^{\frac{i2\pi n(j-k)}{N}} = 1$ y

$$\sum_{n=0}^{N-1} e^{\frac{i2\pi n(j-k)}{N}} = \sum_{n=0}^{N-1} 1 = N.$$

□

Si denotamos por $w_k = e^{ix_k} = e^{\frac{i2\pi k}{N}}$, $k = 0, \dots, N - 1$, podemos reformular el Teorema 3.8 de la siguiente forma:

Teorema 3.9. *Se verifica que*

$$(w^m, w^p) = N\delta_N(m - p), \quad 0 \leq m, p \leq N - 1.$$

Estamos en condiciones de establecer la relación entre los coeficientes del polinomios de fase y la Transformada de Fourier Discreta. En efecto:

Teorema 3.10. *Si el polinomio de fase $p(x) = \sum_{j=0}^{N-1} \beta_j e^{ijx}$ es el único que satisface*

$$p(x_k) = f_k, \quad k = 0, \dots, N - 1,$$

donde f_k son valores complejos y $x_k = \frac{2k\pi}{N}$, entonces los coeficientes del polinomio vienen dados por

$$\beta_j = \frac{1}{N} \sum_{k=0}^{N-1} f_k w_k^{-j} = \frac{1}{N} \sum_{k=0}^{N-1} f_k e^{-i2\pi kj/N}, \quad j = 0, \dots, N - 1. \quad (3.8)$$

Demostración. Por el Teorema 3.4 sabemos que los coeficientes β_j , $j = 0, \dots, N - 1$ son la única solución del sistema lineal (3.3) escrito en la forma compacta siguiente:

$$\sum_{k=0}^{N-1} \beta_k e^{ikx} = f,$$

donde $f = (f_0, \dots, f_{N-1})^T \in \mathbb{C}^N$. Por tanto,

$$\frac{1}{N} \sum_{k=0}^{N-1} f_k w_k^{-j} = \frac{1}{N} \sum_{k=0}^{N-1} f_k \bar{w}_k^j = \frac{1}{N} (f, w^j) = \frac{1}{N} \sum_{k=0}^{N-1} \beta_k (w^k, w^j) = \frac{1}{N} N \beta_j = \beta_j$$

con $j = 0, \dots, N - 1$

□

De nuevo constatamos que los coeficientes $\beta_k, k = 0, \dots, N-1$, se obtienen mediante la Transformada de Fourier Discreta de los números $f_k, k = 0, \dots, N-1$. Además tenemos el siguiente corolario.

Corolario 3.11. *Los coeficientes del único polinomio trigonométrico de grado $\leq N$ dado por*

$$\begin{aligned} \blacksquare \tau(x) &= \frac{a_0}{2} + \sum_{k=1}^m (a_k \cos(kx) + b_k \sin(kx)) , \text{ si } N = 2m + 1, \\ \blacksquare \tau(x) &= \frac{a_0}{2} + \sum_{k=1}^{m-1} (a_k \cos(kx) + b_k \sin(kx)) + \frac{a_m}{2} \cos(mx) , \text{ si } N = 2m, \end{aligned}$$

que verifica las condiciones del problema de interpolación:

$$\tau(x_k) = f_k , x_k = \frac{2\pi k}{N}, k = 0, \dots, N-1,$$

están dados por las siguientes expresiones:

$$\begin{aligned} a_0 &= \frac{2}{N} \sum_{k=0}^{N-1} f_k, \\ a_r &= \frac{2}{N} \sum_{k=0}^{N-1} f_k \cos(rx_k) = \frac{2}{N} \sum_{k=0}^{N-1} f_k \cos\left(\frac{2rk\pi}{N}\right), r = 1, \dots, m, \\ b_r &= \frac{2}{N} \sum_{k=0}^{N-1} f_k \sin(rx_k) = \frac{2}{N} \sum_{k=0}^{N-1} f_k \sin\left(\frac{2rk\pi}{N}\right), r = 1, \dots, m. \end{aligned}$$

Demostración. Se tiene que:

$$\begin{aligned} a_0 &= 2\beta_0 = \frac{2}{N} \sum_{k=0}^{N-1} f_k, \\ a_r &= \beta_r + \beta_{N-r} = \frac{1}{N} \sum_{k=0}^{N-1} f_k [e^{-i2kr\pi/N} + e^{-i2k(N-r)\pi/N}] \\ &= \frac{1}{N} \sum_{k=0}^{N-1} f_k [e^{-i2kr\pi/N} + e^{i2kr\pi/N}] = \frac{2}{N} \sum_{k=0}^{N-1} f_k \cos\left(\frac{2rk\pi}{N}\right), r = 1, \dots, m, \end{aligned}$$

$$\begin{aligned} b_r &= i(\beta_r - \beta_{N-r}) = \frac{i}{N} \sum_{k=0}^{N-1} f_k [e^{-i2kr\pi/N} - e^{-i2k(N-r)\pi/N}] \\ &= \frac{i}{N} \sum_{k=0}^{N-1} f_k [e^{-i2kr\pi/N} - e^{i2kr\pi/N}] = \frac{2}{N} \sum_{k=0}^{N-1} f_k \sin\left(\frac{2rk\pi}{N}\right), r = 1, \dots, m. \end{aligned}$$

□

3.3. COEFICIENTES DE POLINOMIO TRIGONOMÉTRICO DE INTERPOLACIÓN Y DFT23

Hemos constatado por tanto que en el cálculo del polinomio trigonométrico de interpolación de los puntos (x_k, f_k) con $x_k = 2\pi k/N, k = 0, \dots, N - 1$, es fundamental el cálculo de los coeficientes del polinomio de fase $\beta_k, k = 0, \dots, N - 1$, dados por la Transformada de Fourier Discreta:

$$\beta_j = \frac{1}{N} \sum_{k=0}^{N-1} f_k e^{-i2\pi kj/N}, j = 0, \dots, N - 1, \quad (3.9)$$

cuyo cálculo efectivo será objeto el algoritmo de Cooley-Tukey.

Capítulo 4

Ajuste trigonométrico y DFT

En el Capítulo 2 hemos estudiado el problema de aproximación óptima en el sentido de mínimos cuadrados utilizando polinomios trigonométricos. En este capítulo nos plantearemos un problema semejante pero en sentido discreto.

Los contenidos de este capítulo están basados en [7] (pág. 199-207 y 210-215).

4.1. Problema de ajuste trigonométrico

Sea $N \geq 1$ y sean $x_k = \frac{2k\pi}{N}$, $k = 0, \dots, N-1$, N puntos igualmente espaciados en $[0, 2\pi]$. Considerando N valores f_0, \dots, f_{N-1} y un m tal que $N-1 \geq 2m$, planteamos el problema de aproximación trigonométrica por mínimos cuadrados discretos o ajuste trigonométrico del modo siguiente:

Encontrar $\phi \in T_m = \langle \frac{1}{2}, \cos(x), \sin(x), \cos(2x), \sin(2x), \dots, \cos(mx), \sin(mx) \rangle$ tal que :

$$\sum_{k=0}^{N-1} |f_k - \phi(x_k)|^2 = \min_{\tau \in T_m} \sum_{k=0}^{N-1} |f_k - \tau(x_k)|^2.$$

El polinomio trigonométrico ϕ es de la forma:

$$\phi(x) = \frac{a_0}{2} \sum_{j=1}^m (a_j \cos(jx) + b_j \sin(jx)),$$

Este problema de minimización es equivalente a la minimización de un funcional cuadrático en \mathbb{R}^{2m+1} . En efecto, escribiendo

$$\tau(x) = \frac{a_0}{2} \sum_{j=1}^m (\alpha_j \cos(jx) + \beta_j \sin(jx)),$$

$$\phi(x) = \frac{a_0}{2} \sum_{j=1}^m (a_j \cos(jx) + b_j \sin(jx)),$$

tenemos que el problema de minimización anterior es equivalente a:

Encontrar $(a_0, a_1, b_1, \dots, a_m, b_m)^T \in \mathbb{R}^{2m+1}$ tal que

$$\begin{aligned} J(a_0, a_1, b_1, \dots, a_m, b_m) &:= \frac{1}{2} \sum_{k=0}^{N-1} \left| f_k - \frac{a_0}{2} \sum_{j=1}^m (a_j \cos(jx_k) + b_j \sin(jx_k)) \right|^2 \\ &= \min_{(\alpha_0, \alpha_1, \beta_1, \dots, \alpha_m, \beta_m) \in \mathbb{R}^{2m+1}} J(\alpha_0, \alpha_1, \beta_1, \dots, \alpha_m, \beta_m) \\ &= \min_{(\alpha_0, \alpha_1, \beta_1, \dots, \alpha_m, \beta_m) \in \mathbb{R}^{2m+1}} \frac{1}{2} \sum_{k=0}^{N-1} \left| f_k - \frac{a_0}{2} \sum_{j=1}^m (\alpha_j \cos(jx_k) + \beta_j \sin(jx_k)) \right|^2. \end{aligned}$$

Si denotamos por

$$v = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \beta_1 \\ \vdots \\ \alpha_m \\ \beta_m \end{pmatrix} \in \mathbb{R}^{2m+1}, u = \begin{pmatrix} a_0 \\ a_1 \\ b_1 \\ \vdots \\ a_m \\ b_m \end{pmatrix} \in \mathbb{R}^{2m+1},$$

se tiene

$$\frac{1}{2} \sum_{k=0}^{N-1} \left| f_k - \frac{a_0}{2} \sum_{j=1}^m (\alpha_j \cos(jx_k) + \beta_j \sin(jx_k)) \right|^2 = \frac{1}{2} \sum_{k=0}^{N-1} (Av - b)_k^2,$$

donde

$$b = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{N-1} \end{pmatrix}, A = \begin{pmatrix} \frac{1}{2} & \cos x_0 & \sin x_0 & \cdots & \cos mx_0 & \sin mx_0 \\ \frac{1}{2} & \cos x_1 & \sin x_1 & \cdots & \cos mx_1 & \sin mx_1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{1}{2} & \cos x_{N-1} & \sin x_{N-1} & \cdots & \cos mx_{N-1} & \sin mx_{N-1} \end{pmatrix}.$$

Por tanto,

$$\frac{1}{2} \sum_{k=0}^{N-1} (Av - b)_k^2 = \frac{1}{2} \|Av - b\|^2 \quad (4.1)$$

$$= \frac{1}{2} (Av - b)^T (Av - b) \quad (4.2)$$

$$= \frac{1}{2} v^T A^T Av - v^T A^T b + \frac{1}{2} b^T b := J(v). \quad (4.3)$$

y $u = (a_0, a_1, b_1, \dots, a_m, b_m)^T \in \mathbb{R}^{2m+1}$ es solución del sistema lineal

$$A^t A u = A^t b.$$

Sea

$$\tilde{\phi}_0 = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \vdots \\ \frac{1}{2} \end{pmatrix}, \tilde{\phi}_k = \begin{pmatrix} \cos kx_0 \\ \cos kx_1 \\ \vdots \\ \cos kx_{N-1} \end{pmatrix}, \tilde{\tau}_k = \begin{pmatrix} \sin kx_0 \\ \sin kx_1 \\ \vdots \\ \sin kx_{N-1} \end{pmatrix}, k = 1, 2, \dots, m.$$

Entonces el sistema de ecuaciones normales

$$(A^t A)u = A^t b,$$

se escribe de la forma siguiente:

$$\begin{pmatrix} (\tilde{\phi}_0, \tilde{\phi}_0) & (\tilde{\phi}_0, \tilde{\phi}_1) & (\tilde{\phi}_0, \tilde{\tau}_1) & \cdots & (\tilde{\phi}_0, \tilde{\phi}_m) & (\tilde{\phi}_0, \tilde{\tau}_m) \\ (\tilde{\phi}_1, \tilde{\phi}_0) & (\tilde{\phi}_1, \tilde{\phi}_1) & (\tilde{\phi}_1, \tilde{\tau}_1) & \cdots & (\tilde{\phi}_1, \tilde{\phi}_m) & (\tilde{\phi}_1, \tilde{\tau}_m) \\ (\tilde{\tau}_1, \tilde{\phi}_0) & (\tilde{\tau}_1, \tilde{\phi}_1) & (\tilde{\tau}_1, \tilde{\tau}_1) & \cdots & (\tilde{\tau}_1, \tilde{\phi}_m) & (\tilde{\tau}_1, \tilde{\tau}_m) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ (\tilde{\phi}_m, \tilde{\phi}_0) & (\tilde{\phi}_m, \tilde{\phi}_1) & (\tilde{\phi}_m, \tilde{\tau}_1) & \cdots & (\tilde{\phi}_m, \tilde{\phi}_m) & (\tilde{\phi}_m, \tilde{\tau}_m) \\ (\tilde{\tau}_m, \tilde{\phi}_0) & (\tilde{\tau}_m, \tilde{\phi}_1) & (\tilde{\tau}_m, \tilde{\tau}_1) & \cdots & (\tilde{\tau}_m, \tilde{\phi}_m) & (\tilde{\tau}_m, \tilde{\tau}_m) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ b_1 \\ \vdots \\ a_m \\ b_m \end{pmatrix} = \begin{pmatrix} (\tilde{\phi}_0, b) \\ (\tilde{\phi}_1, b) \\ (\tilde{\tau}_1, b) \\ \vdots \\ (\tilde{\phi}_m, b) \\ (\tilde{\tau}_m, b) \end{pmatrix}, \quad (4.4)$$

donde (\cdot, \cdot) designa el producto escalar euclídeo en \mathbb{R}^N definido por:

$$(\tilde{u}, \tilde{v}) = \sum_{j=0}^{N-1} u_j v_j, \tilde{u}, \tilde{v} \in \mathbb{R}^N.$$

A continuación probaremos que la matriz $A^t A$ es una matriz diagonal inversible y por lo tanto el polinomio ϕ existe y es único. Nuestro objetivo es el cálculo explícito de sus coeficientes. Nos resulta de gran utilidad el siguiente lema.

Lema 4.1. *Para todo $r \in \mathbb{Z}$ se tiene que:*

$$\sum_{j=0}^{N-1} \cos rx_j = \sum_{j=0}^{N-1} \cos \frac{2r\pi j}{N} = \begin{cases} 0 & \text{si } r \neq 0. \\ N & \text{si } r = 0. \end{cases}$$

$$\sum_{j=0}^{N-1} \sin rx_j = \sum_{j=0}^{N-1} \sin \frac{2r\pi j}{N} = 0.$$

Demostración. Véase [7] (pág. 212). □

Con esto, estamos ya en condiciones de probar el siguiente teorema:

Teorema 4.2. *La matriz $A^T A$ del sistema (4.4) es diagonal y se tiene que:*

$$\begin{aligned}(\tilde{\phi}_0, \tilde{\phi}_0) &= \frac{N}{4} \\(\tilde{\phi}_k, \tilde{\phi}_k) &= \frac{N}{2}, k = 1, \dots, m. \\(\tilde{\tau}_k, \tilde{\tau}_k) &= \frac{N}{2}, k = 1, \dots, m.\end{aligned}$$

Demostración.

$$\begin{aligned}(\tilde{\phi}_0, \tilde{\phi}_0) &= \sum_{j=0}^{N-1} \frac{1}{2} \frac{1}{2} = \frac{N}{4}, \\(\tilde{\phi}_0, \tilde{\phi}_k) &= \frac{1}{2} \sum_{j=0}^{N-1} \cos kx_j = 0, k = 1, \dots, m. \\(\tilde{\phi}_0, \tilde{\tau}_k) &= \frac{1}{2} \sum_{j=0}^{N-1} \sin kx_j = 0, k = 1, \dots, m. \\(\tilde{\phi}_k, \tilde{\phi}_l) &= \sum_{j=0}^{N-1} \cos kx_j \cos lx_j = \frac{1}{2} \sum_{j=0}^{N-1} [\cos(k-l)x_j + \cos(k+l)x_j] = \begin{cases} 0 & \text{si } k \neq l \\ \frac{N}{2} & \text{si } k = l \end{cases}, k, l = 1, \dots, m. \\(\tilde{\phi}_k, \tilde{\tau}_l) &= \sum_{j=0}^{N-1} \cos kx_j \sin lx_j = \frac{1}{2} \sum_{j=0}^{N-1} [\sin(k-l)x_j + \sin(k+l)x_j] = 0, k, l = 1, \dots, m. \\(\tilde{\tau}_k, \tilde{\tau}_l) &= \sum_{j=0}^{N-1} \sin kx_j \sin lx_j = \frac{1}{2} \sum_{j=0}^{N-1} [\cos(k-l)x_j - \cos(k+l)x_j] = \begin{cases} 0 & \text{si } k \neq l \\ \frac{N}{2} & \text{si } k = l \end{cases}, k, l = 1, \dots, m.\end{aligned}$$

□

Como consecuencia del resultado anterior tenemos el siguiente corolario.

Corolario 4.3. *Considerando el problema de aproximación trigonométrica por mínimos cuadrados discretos o ajuste trigonométrico tiene solución única y el polinomio trigonométrico solución de dicho problema viene dado por:*

$$\phi(x) = \frac{a_0}{2} \sum_{j=1}^m (a_j \cos(jx) + b_j \sin(jx)),$$

donde

$$\begin{aligned}a_0 &= \frac{(\tilde{\phi}_0, b)}{(\tilde{\phi}_0, \tilde{\phi}_0)} = \frac{2}{N} \sum_{j=0}^{N-1} f_j, \\a_k &= \frac{(\tilde{\phi}_k, b)}{(\tilde{\phi}_k, \tilde{\phi}_k)} = \frac{2}{N} \sum_{j=0}^{N-1} f_j \cos kx_j, k = 1, \dots, m, \\b_k &= \frac{(\tilde{\tau}_k, b)}{(\tilde{\tau}_k, \tilde{\tau}_k)} = \frac{2}{N} \sum_{j=0}^{N-1} f_j \sin kx_j, k = 1, \dots, m.\end{aligned}$$

4.2. Ajuste trigonométrico y DFT

Estas expresiones de los coeficientes del polinomio trigonométrico solución son las mismas que las vistas en la aproximación de los coeficientes de Fourier (véase (2.9) y (2.10)). Por tanto, podemos relacionarlas con la Transformada de Fourier Discreta:

$$\beta_j = \frac{1}{N} \sum_{k=0}^{N-1} f_k e^{-ijx_k} = \frac{1}{N} \sum_{k=0}^{N-1} f_k e^{\frac{-2\pi i k j}{N}}, j = 0, \dots, N-1,$$

mediante las mismas relaciones:

$$a_0 = 2\beta_0, a_j = \beta_j + \beta_{N-j}, b_j = i(\beta_j - \beta_{N-j}), j = 1, \dots, m. \quad (4.5)$$

Vemos, por tanto, que la Transformada de Fourier Discreta es necesaria para el cálculo del polinomio trigonométrico de ajuste en el sentido de mínimos cuadrados discretos. Dejando claro, una vez más, la necesidad de implementar un cálculo eficiente de dicha transformada que nos permita realizar ajustes trigonométricos con un número elevado de puntos.

Capítulo 5

Transformada rápida de Fourier(FFT). Algoritmo de Cooley-Tukey

5.1. Transformada rápida de Fourier(FFT)

En los capítulos anteriores, hemos considerado varias situaciones donde la Transformada de Fourier Discreta juega un papel fundamental pero debemos centrarnos ahora en la cuestión más importante desde el punto de vista práctico: el cálculo eficiente de la misma, es decir, dados los valores $f_k, k = 0, \dots, N - 1$, calcular los N valores

$$\beta_j = \frac{1}{N} \sum_{k=0}^{N-1} f_k e^{\frac{-i2\pi kj}{N}}, j = 0, \dots, N - 1. \quad (5.1)$$

Ahora bien, suponiendo calculada la cantidad $\lambda = e^{\frac{-i2\pi}{N}}$, $\beta_j, j = 0, \dots, N - 1$, se escribe como:

$$\beta_j = \frac{1}{N} \sum_{k=0}^{N-1} f_k (\lambda^j)^k,$$

y por tanto se obtiene como la evaluación en λ^j del polinomio de grado $\leq N - 1$, $q(z) = \sum_{k=0}^{N-1} f_k z^k$.

Observemos que, utilizando la regla de Horner, el coste operacional de este cálculo sería de N multiplicaciones y N sumas. Por tanto el coste operacional aproximado de la Transformada de Fourier Discreta (cálculo de los N coeficientes) es del orden de $2N^2$ operaciones (N^2 multiplicaciones y N^2 sumas).

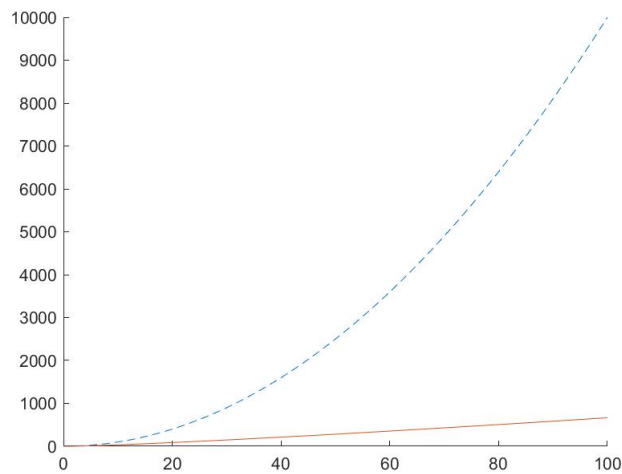


Figura 5.1: Comparación gráfica del orden del coste operacional del cálculo directo (línea discontinua) y del cálculo mediante el algoritmo de Cooley-Tukey (línea continua).

Esta cantidad de cálculos es inabordable en los ordenadores usuales para grandes valores de N . Por ejemplo, en los problemas prácticos de procesamiento de imágenes es usual encontrar $N = 10^6$ lo que supondría $2 \cdot 10^{12}$ operaciones elementales. En un ordenador con 2 Mgflops ($2 \cdot 10^6$ operaciones elementales por segundo) necesitaríamos 10^6 segundos de CPU (esto es, más de 11 días) para calcular la Transformada de Fourier Discreta. Durante mucho tiempo se creyó que este obstáculo era insalvable hasta que en 1965 J.W.Cooley y J.Tukey presentaron en su trabajo bajo el título “An Algorithm for the Machine Calculation of complex Fourier Series” un método revolucionario que permite calcular la Transformada de Fourier Discreta con un número de operaciones del orden de $N \log_2 N$.

Esto permitió reducir drásticamente el tiempo de cálculo y abrió nuevas expectativas en muchas áreas de aplicación. La gráfica 5.1 nos ayuda a hacernos una idea del cambio que significó.

El algoritmo de Cooley-Tukey no es el único existente para la evaluación de la DFT aunque es uno de los principales junto a la versión desarrollada por Gentleman y Sande en 1966 denominado método de Sande-Tukey. El conjunto de variantes de estos métodos se conocen como Transformada Rápida de Fourier (en inglés, Fast Fourier Transform ó FFT). Estos algoritmos se basan en la idea de “divide y vencerá” utilizando una factorización de N y descomponiendo el problema original en subproblemas de orden menor.

Dicho proceso se comporta muy bien en el caso de que N sea una potencia de dos, caso en el que nos centraremos en el presente trabajo: $N = 2^m$, $m > 0$, entero.

Observación 5.1. Existen técnicas para el caso de tener una descomposición de N de la forma $N = r_1, r_2 \dots r_m$, con r_k entero, tanto para el algoritmo de Cooley-Tukey como para el de Sande-Tukey (véase [2], pág, 184).

Antes de introducir el algoritmo de Cooley-Tukey enunciaremos un resultado en términos de los polinomios exponenciales de interpolación que será utilizado repetidamente en el desarrollo del algoritmo y que permite una mejor comprensión del mismo. Seguimos aquí las referencias [6] (pág. 170-178) y [4] (pág. 414-420).

Supongamos $N = 2M$ y consideremos x_0, \dots, x_{N-1} como puntos de interpolación tales que $x_k = \frac{2k\pi}{N} = \frac{k\pi}{N}$ con $k = 0, \dots, N - 1$. El teorema que anunciamos a continuación establece una importante relación entre los polinomios exponenciales de grado $\leq M - 1$ que interpolan en los puntos $x_{2j}, j = 0, \dots, M - 1$ los valores pares f_{2j} y los valores impares $f_{2j+1}, j = 0, \dots, M - 1$ con el polinomio de fase de grado $\leq N - 1$ que interpola la tabla $(x_k, f_k), k = 0, \dots, N - 1$.

Teorema 5.2. Sean $q(x) = \sum_{j=0}^{M-1} \alpha_j e^{ijx}$ y $r(x) = \sum_{j=0}^{M-1} \gamma_j e^{ijx}$ los únicos polinomios exponenciales de grado $\leq M - 1$ tales que:

$$q(x_{2j}) = f_{2j}, r(x_{2j}) = f_{2j+1}, j = 0, \dots, M - 1.$$

Entonces, el único polinomio exponencial $p(x) = \sum_{j=0}^{N-1} \beta_j e^{ijx}$ tal que:

$$p(x_k) = f_k, k = 0, \dots, N - 1$$

viene dado por la expresión:

$$p(x) = \frac{1}{2}q(x)(1 + e^{iMx}) + \frac{1}{2}r(x - \frac{\pi}{M})(1 - e^{iMx}). \quad (5.2)$$

Demostración. Teniendo en cuenta que los polinomios q y r son polinomios de fase de grado $\leq M - 1$ se tiene que el polinomio de fase dado por la expresión (5.2) es de grado $\leq 2M - 1 = N - 1$. Veamos entonces la condición de interpolación, esto es, veamos que:

$$p(x_k) = f_k, k = 0, \dots, N - 1.$$

En efecto, se tiene que :

$$\begin{aligned} p(x_k) &= \frac{1}{2}q(x_k)(1 + e^{iMx_k}) + \frac{1}{2}r(x_k - \frac{\pi}{M})(1 - e^{iMx_k}) \\ &= \frac{1}{2}q(x_k)(1 + e^{ik\pi}) + \frac{1}{2}r(x_{k-1})(1 - e^{ik\pi}). \end{aligned}$$

Ahora bien, observemos que:

$$e^{ik\pi} = \cos(k\pi) + i \sin(k\pi) = \begin{cases} 1 & \text{si } k \text{ par} \\ -1 & \text{si } k \text{ impar} \end{cases}$$

por lo que

$$p(x_k) = \begin{cases} q(x_k) & \text{si } k \text{ par} \\ r(x_{k-1}) & \text{si } k \text{ impar.} \end{cases}$$

Por tanto, en el caso de que $k = 2j$,

$$p(x_k) = q(x_k) = q(x_{2j}) = f_{2j} = f_k,$$

o, si $k = 2j + 1$,

$$p(x_k) = r(x_{k-1}) = r(x_{2j}) = f_{2j+1} = f_k.$$

□

Corolario 5.3. *En las condiciones del teorema anterior, los coeficientes del polinomio de fase $p(x) = \sum_{j=0}^{N-1} \beta_j e^{ijx}$ están dados por:*

$$\beta_j = \frac{1}{2}\alpha_j + \frac{1}{2}e^{-ij\pi/M}\gamma_j, \quad \beta_{M+j} = \frac{1}{2}\alpha_j - \frac{1}{2}\gamma_j e^{-ij\pi/M}, \quad j = 0, \dots, M-1. \quad (5.3)$$

Demostración. Del teorema anterior obtenemos que:

$$r\left(x - \frac{\pi}{M}\right) = \sum_{j=0}^{M-1} \gamma_j e^{ij(x-\pi/M)} = \sum_{j=0}^{M-1} \gamma_j e^{ijx} e^{-ij\pi/M} = \sum_{j=0}^{M-1} e^{-ij\pi/M} \gamma_j e^{ijx}.$$

Por tanto,

$$\begin{aligned} p(x) &= \frac{1}{2}q(x)(1 + e^{iMx}) + \frac{1}{2}r\left(x - \frac{\pi}{M}\right)(1 - e^{iMx}) \\ &= \frac{1}{2} \sum_{j=0}^{M-1} [(1 + e^{iMx})\alpha_j e^{ijx} + e^{-ij\pi/M} \gamma_j e^{ijx} (1 - e^{iMx})] \\ &= \frac{1}{2} \sum_{j=0}^{M-1} e^{ijx} (\alpha_j + \gamma_j e^{-ij\pi/M}) + (\alpha_j - \gamma_j e^{-ij\pi/M}) e^{iMx}. \end{aligned} \quad (5.4)$$

□

Una vez presentado este teorema y denotando por $R(N)$ el número de multiplicaciones necesarias para el cálculo de los coeficientes del polinomio de fase de grado $\leq N-1$, el cual interpola un conjunto de puntos igualmente espaciados dados por $x_k = \frac{2k\pi}{N}$, $k = 0, \dots, N-1$, vamos a establecer una cota para dicho número dada por el siguiente teorema:

Teorema 5.4. $R(2^m) \leq m2^m$

Demostración. En primer lugar se prueba que:

$$R(2M) \leq 2R(M) + 2M. \quad (5.5)$$

En virtud del corolario anterior, tenemos que los coeficientes β_j con $j = 0, \dots, N - 1 = 2M - 1$, del polinomio exponencial vienen dados por los coeficientes $\alpha_0, \dots, \alpha_{M-1}$ y $\gamma_0, \dots, \gamma_{M-1}$ a través de la expresión (5.3) realizándose $2M$ multiplicaciones: en efecto, suponiendo calculada la expresión $\frac{1}{2}e^{-ij\pi/M}$ para $j = 0, \dots, M - 1$, se realizan M multiplicaciones para la expresión $\frac{1}{2}e^{-ij\pi/M}\gamma_j$ con $j = 0, \dots, M - 1$ y M divisiones para $\frac{1}{2}\alpha_j, j = 0, \dots, M - 1$. Como los coeficientes α_j y γ_j con $j = 0, \dots, M - 1$ se calculan ambos con $R(M)$ multiplicaciones, obtenemos la desigualdad (5.5). Ahora, procedemos por inducción para demostrar la desigualdad del teorema.

Para $m = 0$, se interpola en $2^0 = 1$ puntos, por lo que el polinomio viene dado por $p(x) = f_0$ y no se realiza ninguna multiplicación: $R(2^0) = 0$. Suponiendo cierto el teorema para el caso $m \geq 1$. En virtud de la desigualdad (5.5) se tiene que:

$$R(2^{m+1}) = R(2 \cdot 2^m) \leq 2 \cdot R(2^m) + 2 \cdot 2^m \leq 2 \cdot (m2^m) + 2 \cdot 2^m = (m+1)2^{m+1}.$$

□

Observación 5.5. Utilizando el teorema anterior llegamos a la conclusión de que, si $N = 2^m$, tenemos que $R(N) \leq N \log_2 N$.

5.2. Algoritmo de Cooley-Tukey

A través de la utilización repetida del Teorema 5.2 deduciremos el algoritmo de Cooley-Tukey para el cálculo de la Transformada discreta de Fourier cuando $N = 2^m, m \geq 1$.

Los nodos de interpolación vienen dados por:

$$x_j = \frac{2j\pi}{N} = \frac{2j\pi}{2^m} = \frac{j\pi}{2^{m-1}}, j = 0, \dots, 2^m - 1.$$

Para $0 \leq n \leq m$, vamos a considerar los nodos auxiliares

$$x_j^{(n)} = \frac{2j\pi}{2^n} = \frac{2j\pi}{2^m} 2^{m-n} = x_{j \cdot 2^{m-n}}, j = 0, \dots, 2^n - 1.$$

Estos nodos cumplen las siguientes relaciones:

$$x_{2j}^{(n+1)} = \frac{2(2j)\pi}{2^{n+1}} = \frac{2j\pi}{2^n} = x_j^{(n)}, j = 0, \dots, 2^n - 1. \quad (5.6)$$

$$x_k + x_j^{(n)} = x_k + x_{j \cdot 2^{m-n}} = \frac{2k\pi}{N} + \frac{2(j \cdot 2^{m-n})\pi}{N} = \frac{2(k + j \cdot 2^{m-n})\pi}{N} = x_{k+j \cdot 2^{m-n}}. \quad (5.7)$$

Para $0 \leq k \leq 2^{m-n} - 1$, definimos la siguiente notación:

$$f_{k,j}^{(n)} := f(x_{k+j \cdot 2^{m-n}}) = f_{x_{k+j \cdot 2^{m-n}}}, j = 0, \dots, 2^n - 1.$$

Observemos que los índices $x_{k+j \cdot 2^{m-n}}$ para $0 \leq k \leq 2^{m-n} - 1$ y $0 \leq j \leq 2^n - 1$ son coherentes pues:

$$k + j \cdot 2^{m-n} \leq (2^{m-n} - 1) + (2^n - 1)2^{m-n} = 2^m - 1 = N - 1.$$

Para $n = m$ se tiene únicamente $k = 0$ y

$$f_{0,j}^{(m)} = f_j, j = 0, \dots, 2^m - 1.$$

Además, para $0 \leq n < m$ y $j = 0, \dots, 2^n - 1$ tenemos las siguientes relaciones:

$$f_{k,2j}^{(n+1)} = f_{x_{k+2j \cdot 2^{m-n-1}}} = f_{x_{k+j \cdot 2^{m-n}}} = f_{k,j}^{(n)}, \quad 0 \leq 2j \leq 2^{n+1} - 2, \quad (5.8)$$

$$f_{k,2j+1}^{(n+1)} = f_{x_{k+(2j+1) \cdot 2^{m-n-1}}} = f_{x_{k+2^{m-n-1}+j \cdot 2^{m-n}}} = f_{k+2^{m-n-1},j}^{(n)}, \quad 0 \leq 2j+1 \leq 2^{n+1} - 1. \quad (5.9)$$

Ahora, para $0 \leq n \leq m$ y $0 \leq k \leq 2^{m-n} - 1$ denotamos por $p_k^{(n)}$ el único polinomio de fase de grado $\leq 2^n - 1$ tal que

$$p_k^{(n)}(x_j^{(n)}) = f_{k,j}^{(n)}, j = 0, \dots, 2^n - 1. \quad (5.10)$$

En particular para $n = m$ se tiene que $k = 0$ y el polinomio de fase anterior es un polinomio de grado $\leq 2^m - 1 = N - 1$ tal que

$$p_0^{(m)}(x_j^{(m)}) = f_{0,j}^{(m)}, j = 0, \dots, 2^m - 1,$$

esto es,

$$p_0^{(m)}(x_j) = f_j, j = 0, \dots, N - 1,$$

es decir, $p_0^{(m)}$ es el polinomio de fase que buscamos para la interpolación.

Ahora bien, para $n < m$, por definición, para $0 \leq k \leq 2^{m-n-1} - 1$ se tiene que:

$$p_k^{(n+1)}(x_j^{(n+1)}) = f_{k,j}^{(n+1)}, j = 0, \dots, 2^{n+1} - 1.$$

Teniendo en cuenta que $2^{n+1} = 2 \cdot 2^n$, podemos aplicar el Teorema 5.2 para el caso $M = 2^n$, por lo que:

$$p_k^{(n+1)}(x) = \frac{1}{2} q_k^{(n+1)}(x) (1 + e^{i2^n x}) + \frac{1}{2} r_k^{(n+1)}(x - \frac{\pi}{2^n}) (1 - e^{i2^n x}) \quad (5.11)$$

donde $q_k^{(n+1)}$ y $r_k^{(n+1)}$ son los únicos polinomios de fase de grado $\leq 2^n - 1$ tales que

$$q_k^{(n+1)}(x_{2j}^{(n+1)}) = f_{k,2j}^{(n+1)}, j = 0, \dots, 2^n - 1, \quad (5.12)$$

$$r_k^{(n+1)}(x_{2j}^{(n+1)}) = f_{k,2j+1}^{(n+1)}, j = 0, \dots, 2^n - 1. \quad (5.13)$$

Ahora bien, si tenemos en cuenta las relaciones (5.6), (5.8) y (5.9), podemos reescribir las anteriores condiciones de la siguiente manera:

$$q_k^{(n+1)}(x_j^{(n)}) = f_{k,j}^{(n)}, j = 0, \dots, 2^n - 1,$$

$$r_k^{(n+1)}(x_j^{(n)}) = f_{k+2^{m-n-1},j}^{(n)}, j = 0, \dots, 2^n - 1.$$

De esta forma, y debido a la unicidad de los polinomios de interpolación, deducimos que:

$$q_k^{(n+1)} = p_k^{(n)}, r_k^{(n+1)} = p_{k+2^{m-n-1}}^{(n)},$$

y, volviendo sobre la expresión (5.11), podemos reescribirla de la siguiente forma, que es una relación recurrente para $k = 0, \dots, 2^{m-n-1} - 1$:

$$p_k^{(n+1)}(x) = \frac{1}{2}p_k^{(n)}(x)(1 + e^{i2^n x}) + \frac{1}{2}p_{k+2^{m-n-1}}^{(n)}(x - \frac{\pi}{2^n})(1 - e^{i2^n x}). \quad (5.14)$$

Comenzando en $n = 0$ se tiene que $j = 0$, obteniendo por tanto los polinomios de grado cero siguientes:

$$p_k^{(0)}(x_0^{(0)}) = f_{k,0}^{(0)} = f(x_k), k = 0, \dots, 2^m - 1.$$

A medida que avanzamos en el algoritmo se calculan polinomios de fase de grado superior en una unidad al anterior a través de los polinomios calculados anteriormente. Donde, finalmente, se llega a que $n = m$ y $k = 0$ y se obtiene el polinomio

$$p_0^{(m)}(x_j) = f_j, j = 0, \dots, 2^m - 1,$$

esto es, el polinomio de fase buscado. La Figura 5.2 muestra el fundamento del algoritmo de Cooley-Tukey para el caso $m = 3$ y $N = 8$.

Mediante la relación de recurrencia obtenida para los polinomios de fase y teniendo en cuenta el Corolario 5.3, podemos obtener una nueva relación de recurrencia para los coeficientes de los polinomios de fase. Denotando

$$p_k^{(n)}(x) = \sum_{j=0}^{2^n-1} \beta_{k,j}^{(n)} e^{ijx}, 0 \leq n \leq m, 0 \leq k \leq 2^{m-n} - 1,$$

obtenemos que para $0 \leq n \leq m - 1$ y $0 \leq k \leq 2^{m-n-1} - 1$ se tiene:

$$\beta_{k,j}^{(n+1)} = \frac{1}{2}[\beta_{k,j}^{(n)} + e^{-ij\pi/2^n} \beta_{k+2^{m-n-1},j}^{(n)}], j = 0, \dots, 2^n - 1, \quad (5.15)$$

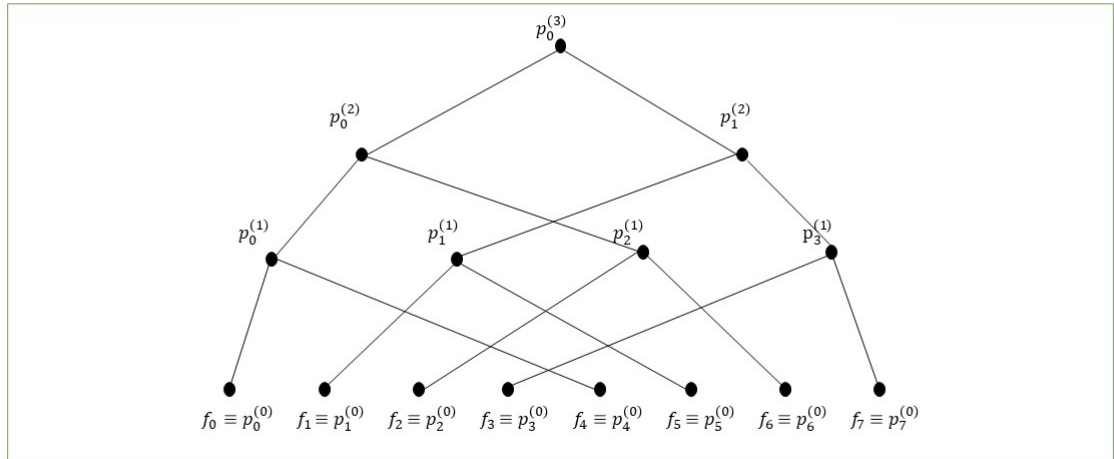


Figura 5.2: Ilustración del algoritmo de Cooley-Tukey para $m = 3$ y $N = 8$.

$$\beta_{k,j+2^n}^{(n+1)} = \frac{1}{2} [\beta_{k,j}^{(n)} - e^{-ij\pi/2^n} \beta_{k+2^{m-n-1},j}^{(n)}], \quad j = 0, \dots, 2^n - 1. \quad (5.16)$$

Observemos que en la primera iteración de la relación de recurrencia para los polinomios de fase, es decir, para el caso $n = 0$, los coeficientes de estos polinomios de grado cero vienen dados por:

$$\beta_{k,0}^{(0)} = f_k, \quad k = 0, \dots, 2^m - 1 = N - 1,$$

y que, en la última interacción, $n = m$, se tiene que:

$$\beta_{0,j}^{(m)}, \quad j = 0, \dots, 2^m - 1 = N - 1,$$

son los coeficientes del polinomio de fase buscado.

Las relaciones de recurrencia (5.15) y (5.16) para el cálculo de los coeficientes de los polinomios de interpolación dados en la relación (5.14) conforman la base del algoritmo de Cooley-Tukey que podemos resumir de la siguiente forma:

- Bucle en $k = 0, \dots, 2^m - 1$

$$\beta_{k,0}^{(0)} = f_k$$

- Fin bucle en k

- Bucle en $n = 0, \dots, m - 1$
 - Bucle en $k = 0, \dots, 2^{m-n-1} - 1$
 - Bucle en $j = 0, \dots, 2^n - 1$

$$\beta_{k,j}^{(n+1)} = \frac{1}{2}[\beta_{k,j}^{(n)} + e^{-ij\pi/2^n} \beta_{k+2^{m-n-1},j}^{(n)}]$$

$$\beta_{k,j+2^n}^{(n+1)} = \frac{1}{2}[\beta_{k,j}^{(n)} - e^{-ij\pi/2^n} \beta_{k+2^{m-n-1},j}^{(n)}]$$
 - Fin bucle en j
 - Fin bucle en k
- Fin bucle en n

Ahora bien, para la realización de estos cálculos podemos organizarlos de forma eficiente en base a los siguientes hechos: como para cada n se tiene que $0 \leq k \leq 2^{m-n} - 1$ y $0 \leq j \leq 2^n - 1$, los coeficientes $\beta_{k,j}^{(n)}$ ocupan $2^{m-n} \cdot 2^n = 2^m = N$ posiciones de memoria. Por lo tanto, utilizamos dos variables de memoria de longitud N , a y b , de forma que, en cada etapa del cálculo en la variable a y b se almacenarán los valores siguientes:

$$a \leftarrow \beta_{k,j}^{(n)}, \quad 0 \leq k \leq 2^{m-n} - 1 \text{ y } 0 \leq j \leq 2^n - 1,$$

$$b \leftarrow \beta_{k,j}^{(n+1)}, \quad 0 \leq k \leq 2^{m-n-1} - 1 \text{ y } 0 \leq j \leq 2^{n+1} - 1.$$

Deste modo, en la siguiente etapa:

$$a \leftarrow \beta_{k,j}^{(n+1)}, \quad 0 \leq k \leq 2^{m-n-1} - 1 \text{ y } 0 \leq j \leq 2^{n+1} - 1$$

$$b \leftarrow \beta_{k,j}^{(n+2)}, \quad 0 \leq k \leq 2^{m-n-2} - 1 \text{ y } 0 \leq j \leq 2^{n+2} - 1$$

El almacenamiento puede ajustarse, por ejemplo, a la regla siguiente:

$$a(2^n k + j) = \beta_{k,j}^{(n)}, \quad 0 \leq k \leq 2^{m-n} - 1, \quad 0 \leq j \leq 2^n - 1,$$

$$b(2^{n+1} k + j) = \beta_{k,j}^{(n+1)}, \quad 0 \leq k \leq 2^{m-n-1} - 1, \quad 0 \leq j \leq 2^{n+1} - 1.$$

Se introducen ahora los factores:

$$z_j := \frac{1}{2} e^{\frac{-i2j\pi}{N}} = \frac{1}{2} e^{\frac{-i2j\pi}{2^m}}, \quad j = 0, \dots, 2^m - 1 = N - 1,$$

de modo que para calcular el factor $e^{-ij\pi/2^n}$ de (5.15) y (5.16), basta tener en cuenta que :

$$\frac{1}{2} e^{-ij\pi/2^n} = \frac{1}{2} e^{-ij\pi 2^{m-n}/2^m} = z_j 2^{m-n-1}, \quad j = 0, \dots, 2^m - 1, \quad 0 \leq n \leq m.$$

Con esta organización de los cálculos podemos completar el pseudocódigo anterior de la siguiente forma:

- $w = e^{-i\pi/N}$

- Bucle en $j = 0, \dots, N - 1$
 - $z(j) = w^j/2$
 - $a(j) = f(2j\pi/N)$
- Fin bucle en j
- Bucle en $n = 0, \dots, m - 1$
 - Bucle en $k = 0, \dots, 2^{m-n-1} - 1$
 - Bucle en $j = 0, \dots, 2^n - 1$
 - $u = a(2^n k + j)/2$
 - $v = z(2^{m-n} j) \cdot a(2^n k + 2^{m-1} + j)$
 - $b(2^{n+1} k + j) = u + v$
 - $b(2^{n+1} k + j + 2^n) = u - v$
 - Fin bucle en j
 - Fin bucle en k
 - Bucle en $j = 0, \dots, N - 1$
 - $a(j) = b(j)$
 - Fin bucle en j
- Fin bucle en n .

Capítulo 6

Resultados

Una vez presentado el algoritmo de Cooley-Tukey vamos a comprobar en este capítulo de forma práctica que cumplimos nuestros objetivos, esto es, la posibilidad de calcular la Transformada Discreta de Fourier con un tiempo de cálculo muy inferior al dado por el cálculo directo de dicha transformada, obteniendo tiempos de cálculo admisibles.

Debido a que el tiempo de cálculo depende del ordenador donde se realicen, debemos especificar sus características. Los cálculos que mostraremos a continuación se han realizado en un ordenador con procesador 2,7 GHz Intel Core i5 de 7th Gen con memoria 8 GB 2133 MHz y con el sistema operativo Windows 10 Home x64.

Vemos a continuación algunos ejemplos de los problemas vistos en los capítulos 1 al 4 y la aplicación del algoritmo de Cooley-Tukey comparado con el cálculo directo de la Transformada de Fourier Discreta.

Ejemplo 6.1. Consideramos el cálculo aproximado de los coeficientes del polinomio de Fourier de grado m de una función $f \in L^2(0, 2\pi)$,

$$\frac{a_0}{N} + \sum_{j=1}^m a_j \cos(jx) + b_j \sin(jx), x \in [0, 2\pi],$$

cuyos coeficientes calculamos utilizando la transformada de Fourier discreta, siguiendo el Capítulo 2.

Vamos a considerar como ejemplo la siguiente función constante a trozos en $[0, 2\pi]$:

$$f(x) = \begin{cases} -\pi/4 & \text{si } x \in (0, \pi/2] \\ \pi/4 & \text{si } x \in (\pi/2, 3\pi/2) \\ -\pi/4 & \text{si } x \in [3\pi/2, 2\pi). \end{cases}$$

Calcularemos los coeficientes del polinomio de Fourier de grado $m = 10$ siguiendo las fórmulas del Capítulo 2, utilizando N nodos de cuadratura (ver (2.9) y (2.10)) y estudiamos

Puntos	Cooley-Tukey	Directo
$N = 2^8$	0,105242 s	0,802271 s
$N = 2^{10}$	0,185592 s	9,902223 s
$N = 2^{12}$	0,475230 s	147,213065 s
$N = 2^{14}$	1,757267 s	+10 min
$N = 2^{16}$	5,988409 s	+1 h

Cuadro 6.1: Comparativa del tiempo de cálculo

Puntos	Cooley-Tukey	Directo
$N = 2^8$	0,071886 s	0,088680 s
$N = 2^{10}$	0,114333 s	0,750772 s
$N = 2^{12}$	0,184071 s	10,21712 s
$N = 2^{14}$	0,483673 s	164,1685 s
$N = 2^{16}$	1,666707 s	2490,764269 s

Cuadro 6.2: Comparativa del tiempo de cálculo

el tiempo de cálculo comparando Cooley-Tukey y el cálculo directo de la transformada. Como podemos apreciar en la cuadro 6.1, la aproximación de los coeficientes del polinomio de Fourier es inadmisibles por el cálculo directo por lo que recurrimos al cálculo mediante el algoritmo de Cooley-Tukey, que nos proporciona unos tiempos razonables.

Ejemplo 6.2. Nuestro objetivo en este ejemplo es aproximar mediante interpolación trigonométrica la función $f(x) = e^{-x^2} \cos(5x)$ restringida al intervalo $[0, 2\pi]$, tomando un conjunto de nodos igualmente espaciados: $x_j = \frac{2\pi j}{N}, j = 0, \dots, N - 1$, tal como vimos en el Capítulo 3 .

En la tabla 6.2 vemos una comparación del tiempo empleado para distinto número de puntos utilizando el algoritmo de Cooley-Tukey y el cálculo directo de la Transformada Discreta de Fourier para la obtención de los coeficientes (ver sección 3.3).

Observamos que para un número de puntos $\leq 2^{10}$ aunque la diferencia de tiempos es ya elevada, los cálculos mediante el cálculo directo son asumibles. Para un número de puntos mayores que 2^{10} , el tiempo de cálculo empleado es demasiado alto, poniendo de manifiesto la utilidad del algoritmo de Cooley-Tukey al permitirnos unos tiempos de cálculo razonables.

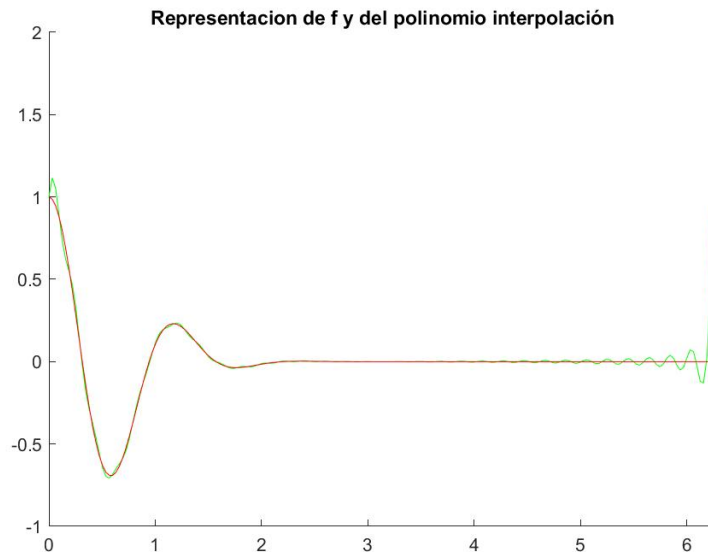


Figura 6.1: Aproximación de la función f mediante un polinomio trigonométrico de interpolación en N nodos con $N = 2^6$ (verde) y $N = 2^{14}$ (rojo).

En la Figura 6.1 se muestra una representación de la aproximación de la función f y del número de nodos $N = 2^6$ y $N = 2^{14}$. La reconstrucción es más precisa a mayor número de datos y, para $N = 2^{16}$, el algoritmo presenta una gran ventaja en el coste operacional y, por tanto, en el tiempo necesario.

Ejemplo 6.3. Nos centraremos ahora en un ejemplo de aproximación trigonométrica en el sentido de mínimos cuadrados discretos o ajuste discreto de una función f tal como hemos visto en el Capítulo 4 (ver relaciones (4.5)).

A modo de ejemplo consideremos la función:

$$f(x) = \begin{cases} \cos(x^2) & \text{si } x \in (0, \pi] \\ \pi - x & \text{si } x \in (\pi, 2\pi). \end{cases}$$

Comparamos, de nuevo, el algoritmo de Cooley-Tukey frente al cálculo directo para la obtención de los coeficientes del polinomio de ajuste.

Los tiempos de cálculo necesarios para la DFT por ambos métodos están reflejados en el cuadro 6.3 donde se manifiesta, una vez más, la imperiosa necesidad del uso del algoritmo expuesto en este trabajo cuando el número de datos que manejamos es considerable. La reconstrucción para $N = 2^8$ y $N = 2^{14}$ datos están representadas en la figura 6.2.

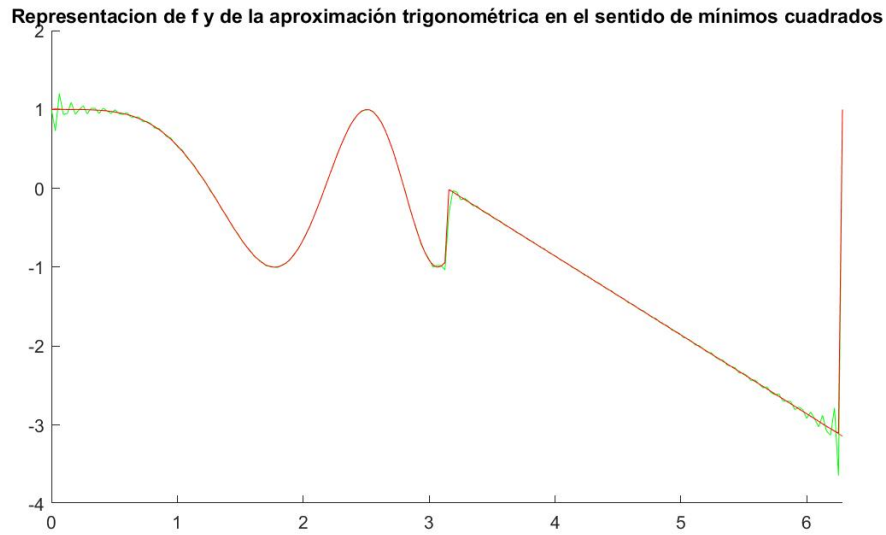


Figura 6.2: Aproximación de la función f mediante un polinomio trigonométrico en el sentido de mínimos cuadrados discretos para $N = 2^8$ (verde) y $N = 2^{14}$ (rojo).

Puntos	Cooley-Tukey	Directo
$N = 2^8$	0,092109 s	0,766054 s
$N = 2^{10}$	0,135679 s	10,093025 s
$N = 2^{12}$	0,251216 s	147,302132 s
$N = 2^{14}$	0,690212 s	+10 min
$N = 2^{16}$	2,404724 s	+1 h

Cuadro 6.3: Comparativa del tiempo de cálculo

Anexo: Códigos en Matlab

En este anexo incluimos las dos funciones de Matlab utilizadas para el cálculo directo de la Transformada discreta de Fourier y el cálculo de dicha transformada mediante el algoritmo de Cooley-Tukey.

```
function [b] = directo(N)
tic
b=zeros(N,1);
for j=0:N-1
    for k=0:N-1
        d(k+1)=f((2*pi*k)/N)*exp((-i*j*k*pi)/N);
        b(j+1)=b(j+1)+d(k+1);
    end
    b(j+1)=b(j+1)/N;
end
toc
end
```

```
function [a]= cooleytukey1(N,m)
tic
w=exp(-i*pi/N);
for j=0:N-1
    a(j+1)=f(2*j*pi/N);
    z(j+1)=w^j/2;
end
for n=0:m-1
    for k=0:2^(m-n)-1
        for j=0:2^n-1
            u=a(2^n*k+j+1)/2;
            v=z(2^(m-n)*j+1)*a(2^n*k+2^(m-1)+j+1);
            b(2^(n+1)*k+j+1)=u+v;
            b(2^(n+1)*k+j+2^n+1)=u-v;
        end
    end
end
for j=0:N-1
```

```
                a(j+1)=b(j+1);  
            end  
end  
toc  
end
```

Bibliografía

- [1] Birggs, W.L. and Henson, V.E., *The DFT: an owner's manual for the discret Fourier Transform*, Siam,(1995).
- [2] Brigham, E.O., *The Fast Fourier Trasnform*, Prentice Hall (1974).
- [3] Davis, P.J., *Interpolation and Approximation*, Dover (1975).
- [4] Kincaid, D. and Cheney, W.,*Numerical analisis*, Brooks/Cole Pub. (1991)
- [5] Stoer, J.-Bulirsch, R., *Introduction to Numercal Analisis*, Springer-Verlag, 2nd ed(1993)
- [6] Viaño, J.M. y Burguera, M., *Lecciones de Métodos Numéricos 3 Interpolación*, Tórculo Edicións (2000).
- [7] Viaño, J.M. y Burguera, M., *Lecciones de Métodos Numéricos 4 Optimización*, Andavira Editora (2013).