UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN

CÁLCULO CIENTÍFICO (6109)
SEMESTRE I-2025
02 DE JULIO DE 2025

Proyecto: Transformada de Fourier

Fecha de Entrega: Miércoles 02-07-2025

Introducción

De la teoría de interpolación se sabe que **si** $x_0, x_1, \ldots, x_n$ son números reales distintos, **entonces** para valores arbitrarios $y_0, y_1, \ldots, y_n$ existe un único polinomio, denotado por p_n tal que

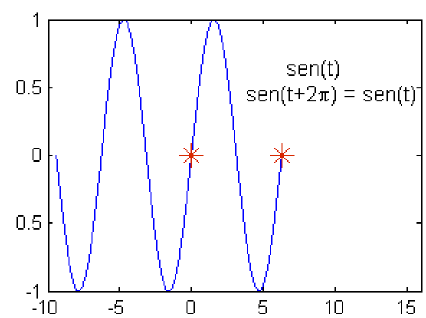
$$p_n(x_i) = f(x_i) = y_i. \tag{1}$$

Note que $p_n \in \Pi_n$ donde Π_n representa el espacio vectorial de todos los polinomios de grado menor o igual a n. Si $\mathcal{B}_n$ es una base de Π_n, denotada por $\mathcal{B}_n = \{b_0, b_1, \ldots, b_n\}$, entonces *el polinomio interpolador* p_n posee la siguiente forma

$$f(x) \approx p_n(x) = \sum_{j=0}^{n} c_j b_j(x),$$

donde los coeficientes c_j de la combinación lineal que definen a p_n se obtienen imponiendo las llamadas n *condiciones de interpolación* descritas en (1).

Ahora bien, por un momento imagine que la función f posee cierta periodicidad en su comportamiento, es decir, f está asociada a algún fenómeno de naturaleza cíclica (sonidos musicales, trayectorias de cuerpos celestes, etc). En estos casos es claro que aproximar a f mediante un elemento de Π_n (polinomio) no resulta la mejor elección. De hecho es mucho más conveniente aproximar f como una combinación lineal de funciones senoidales, es decir, utilizar interpolación trigonométrica en vez de interpolación polinomial. En este nuevo contexto, decimos que la f es una función periódica, si existe una constante $p > 0$, que denominaremos *período*, tal que $f(t) = f(t + p)$ para todo t. Por ejemplo, en figura se observa la función seno que es periódica con $p = 2\pi$.



Interpolación trigonométrica

Sabemos que **es necesario tener una base del espacio** Π_n **para hallar el polinomio interpolante** en los **nodos** (x_i, y_i). Por tanto, **para emular** lo aprendido debemos, en primer lugar, cambiar el espacio vectorial; hallar una base del mismo e identificar los nodos de interpolación. En ese sentido,

1. Sea $\mathcal{V}$ el espacio de las funciones periódicas con $p = 2\pi$, definidas en el campo complejo[1].

2. En este espacio, el conjunto $\mathcal{B} \subset \mathcal{V}$ definido como $\mathcal{B} = \{\phi_0, \phi_1, \ldots, \phi_{n-1}\}$ forma una base ortonormal donde

$$\phi_j := \phi_j(x) = e^{ijx}, \text{ para } j = 0, 1, \ldots, n. \tag{2}$$

[1] las funciones de $\mathcal{V}$ están definidas como $f : \mathbb{C} \to \mathbb{C}$

siendo i la unidad compleja, es decir, $i = \sqrt{-1}$; mientras que el producto interno usado se define como

$$\langle f, g \rangle = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t)\overline{g(t)}dt, \tag{3}$$

donde $\overline{g(t)}$ denota al transpuesto conjugado de $g(t)$.

Pregunta: Demuestre que $\mathcal{B} = \{\phi_0, \phi_1, \ldots, \phi_{n-1}\}$ es un conjunto ortonormal con el producto interno descrito en (3).

ATENCIÓN: Seguramente se está preguntando **por qué $\mathcal{B}$ es base de un espacio de funciones periódicas, si las funciones que la conforman no parecen ser periódicas sino exponenciales**, ver (2). Sin embargo, cada ϕ_k es de hecho una función periódica escrita de forma implícita. Para ver la conexión, basta recordar la famosa **fórmula de Euler**:

$$e^{i\theta} = \cos\theta + i\sin\theta,$$

y recuerde que ϕ_k se define como $\phi_k = e^{ikx}$.

Como dato curioso, observe que al sustituir θ por π en la fórmula de Euler se tiene que $\cos\pi = -1$ y $\sin\pi = 0$ y por tanto

$$\boxed{e^{i\pi} + 1 = 0}$$

que es una de las identidades más hermosas de las Matemáticas. Su belleza radica en su simplicidad y en que relaciona 5 números emblemáticos de las Matemáticas:

- e, π: números irracionales,
- i: unidad compleja,
- 1, 0: elementos neutros de la multiplicación y la suma, que son las operaciones esenciales para definir espacios vectoriales.

3. Los nodos x_k (distintos entre si) serán $x_k = 2\pi k/n$ con $k = 0, 1, \ldots, n-1$. Esta escogencia se hace para simplificar cálculos posteriores; y además toma en cuenta la periodicidad de la función.

4. Finalmente, tenemos que $y_k = f(x_k)$. Sin embargo, más adelante volveremos sobre estos valores y_k.

Ahora traiga a la mente un resultado básico de álgebra lineal estudiado en clases: Si $\mathcal{V}$ es un espacio con producto interno $\langle .\,, . \rangle$ y si $\mathcal{B} = \{\varphi_0, \varphi_1, \ldots, \varphi_{n-1}\}$ es una base ortonormal de $\mathcal{V}$; entonces cada $f \in \mathcal{V}$ se puede escribir como

$$f = \langle \varphi_0, f \rangle \varphi_0 + \langle \varphi_1, f \rangle \varphi_1 + \cdots + \langle \varphi_{n-1}, f \rangle \varphi_{n-1}.$$

Pregunta: Observe que esta última expresión define *al interpolador ideal de f* pero no tiene ni un gramo de sentido práctico. **¿Por qué?**

Para tener un mecanismo útil desde el punto de vista práctico para aproximar f, usaremos un *"casi-producto interno discreto"* en vez del producto interno continuo definido en (3). Este *"casi-producto interno"* se define como:

$$\langle f, g \rangle_n = \frac{1}{n} \sum_{k=0}^{n-1} f(x_k)\overline{g(x_k)}. \tag{4}$$

Preguntas: Sabemos que la función $\langle\ ,\ \rangle : \mathcal{V}\times\mathcal{V} \to \mathbb{R}$ es un **producto interno** si cumple lo siguiente:

1. $\langle x, x\rangle \geq 0$ y $\langle x, x\rangle = 0 \Leftrightarrow x = 0$
2. $\langle x, y\rangle = \overline{\langle y, x\rangle}$[a]
3. $\langle x, \alpha y + \beta z\rangle = \alpha\langle x, y\rangle + \beta\langle x, z\rangle$

(1) La función definida en (4) cumplen todas las propiedades menos una. Pruebe las que se satisfacen y de un contraejemplo para la que no.

(2) Demuestre que las funciones ϕ_j cumplen que

$$\langle \phi_j, \phi_k\rangle_n = \begin{cases} n & \text{si } (j-k) \text{ es divisible por } n \\ 0 & \text{en otros casos} \end{cases}$$

para $0 \leq k, j \leq n$.

En este contexto decimos que $\mathcal{B} = \{\phi_0, \phi_1, \ldots, \phi_{n-1}\}$ es un conjunto ortonormal (con el *casi-producto interno* descrito en (4)).

[a] aparece el transpuesto conjugado por el uso de la aritmética compleja.

Bajo este escenario, la siguiente función p se define como el **interpolador trigonométrico** de la función f en los nodos $x_k = 2\pi k/n$:

$$f(x) \approx p(x) = \sum_{j=0}^{n-1} c_j\phi_j = \sum_{j=0}^{n-1} c_j e^{ijx}, \tag{5}$$

donde

$$c_j = \langle f, \phi_j\rangle_n = \frac{1}{n}\sum_{k=0}^{n-1} f(x_k)\overline{\phi_j(x_k)} = \frac{1}{n}\sum_{k=0}^{n-1} f(x_k)e^{-ijx_k}, \quad j = 0, 1, \ldots, n-1 \tag{6}$$

pues si $\phi_j = e^{ijx}$ entonces $\overline{\phi_j} = e^{-ijx}$.

Preguntas:

(a) ¿El uso del *casi-producto interno* definido por (4) elimina las limitaciones observadas al usar el producto interno continuo definido en (3)?

(b) Desde el punto de vista práctico, ¿las expresiones (5) y (6) son útiles para construir el interpolador de f?

Transformada Discreta de Fourier

Antes de continuar haremos algunas consideraciones para simplificar la notación:

- Recuerde que $y_k = p(x_k) = f(x_k)$.
- Recordando que $x_k = 2\pi k/n$, analicemos el valor de $\phi_j(x_k)$:

$$\phi_j(x_k) = e^{ijx_k} = e^{\frac{ij2\pi k}{n}} = e^{\frac{2\pi i}{n}jk} = \left(e^{\frac{2\pi i}{n}}\right)^{jk},$$

y de manera análoga

$$\overline{\phi_j(x_k)} = e^{-ijx_k} = \left(e^{\frac{-2\pi i}{n}}\right)^{jk}.$$

- Usando el cambio $\omega = e^{\frac{-2\pi i}{n}}$ se tiene que (6) se puede reescribir como:

$$c_j = \frac{1}{n}\sum_{k=0}^{n-1} y_k \omega^{jk}, \quad j = 0, 1, \dots, n-1. \tag{7}$$

- Mientras que al evaluar (5) en $x = x_k$

$$y_k = f(x_k) = \sum_{j=0}^{n-1} c_j \overline{\omega}^{jk}, \quad k = 0, 1, \dots, n-1 \tag{8}$$

  pues recuerde que $\overline{\omega} = e^{\frac{2\pi i}{n}}$.

ATENCIÓN: Las transformaciones lineales definidas por (7) y (8) se denominan la **transformada discreta de Fourier**; e **inversa de la transformada discreta de Fourier**, respectivamente.

Preguntas: **Como bien deben recordar**, dado que (7) y (8) son **transformaciones lineales** deben poder escribirse mediante **productos matriz-vector**.

(a) Si $y = (y_0, \dots, y_j, \dots, y_{n-1})^T$, determine la matriz F asociada a (7) tal que $c = Fy$ con $c = (c_0, \dots, c_j, \dots, c_{n-1})^T$

(b) Demuestre que $F^H F = \frac{1}{n} I$, donde I representa la identidad de orden n y F^H es la transpuesta conjugada de F, es decir, si $F = [f_{kj}]$ entonces $F^H = [\overline{f}_{jk}]$. Observe que de aquí se desprende que $F^{-1} = nF^H$.

(c) Demuestre que en efecto (8) se puede escribir matricialmente como $y = F^{-1}c$.

Nota: F se conoce como *matriz de Fourier*.

Algoritmo de la Transformada Rápida de Fourier

No pierda de vista que el objetivo es hallar los n coeficientes c_j que definen al interpolante trigonométrico descrito en (5). De la pregunta anterior queda claro que estos c_j *no son más que el resultado de hacer un producto matriz-vector* y por tanto, el costo computacional es de aproximadamente n^2 operaciones en aritmética compleja.

El algoritmo de la *Transformada Rápida de Fourier*, FFT por sus siglas en inglés[2], es tan famoso porque es capaz de calcular dichos coeficientes en un tiempo aproximado de $n \log_2 n$ cuando n es potencia de 2. Observe la Tabla para comprender la magnitud de la reducción.

n	$n \log_2 n$	n^2
$2^7 = 128$	896	16384
$2^8 = 256$	2048	65536
$2^9 = 512$	4608	262144
$2^{10} = 1024$	10240	1048576

No describiremos en detalle el algoritmo de la transformada rápida, que en cierto sentido requiere el manejo de estructuras de árbol [3], pero podemos ilustrar con un ejemplo simple donde radica la rapidez de dicho algoritmo. Considere el polinomio $p(z) = z^n - 1$ y los valores complejos $\{\omega^0, \omega^1, \omega^2, \dots, \omega^{n-1}\}$ que son raíces de p.

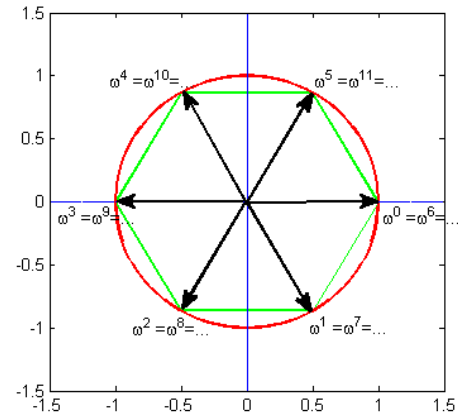
[2] Fast Fourier Transform

[3] Por algo surge el $\log_2$.

donde ω, tal y como lo hemos definido anteriormente, es igual a $e^{\frac{-2\pi i}{n}}$. Estas raíces se *repiten* en el siguiente sentido: Para $k \geq n$ se tiene que $\omega^k = \omega^{k \bmod n}$. Por ejemplo, para $n = 6$ se tiene que

$\omega^0 = \omega^6 = \dots$	$\omega^1 = \omega^7 = \dots$	$\omega^2 = \omega^8 = \dots$
$\omega^3 = \omega^9 = \dots$	$\omega^4 = \omega^{10} = \dots$	$\omega^5 = \omega^{11} = \dots$

y en la Figura se puede observar donde se ubica cada ω^k. Ahora bien, recuerde que para construir los c_j se requieren muchas potencias de ω pero, como acabamos de ver, la repetición de estas potencias nos permite calcularlas de forma inteligente evitando cálculos innecesarios. Además observe que $\omega^0 = 1$ y $\omega^3 = -1$.



Análisis espectral

Ya descrito muy brevemente el contexto de la interpolación trigonométrica y su relación con la transformada de Fourier, ahora presentamos un uso muy simple de la misma: el análisis espectral de marcado de número telefónicos[4]: Cuando pulsa un dígito en el teclado del teléfono (dato discreto) se produce un sonido (dato continuo). **El objetivo (muy simple) de este proyecto es que entiendan y programen el proceso de codificación y decodificación asociado a esta simple actividad, usando la transformada discreta de Fourier**.

Para lo anterior conviene discutir un poco sobre el sistema de marcación por tonos o DTMF por sus siglas en ingles (*Dual-Tone Multi-Frequency*)[5]. Este sistema de codificación es muy simple y a groso modo consiste en lo siguiente:

A cada fila y columna del teclado de un teléfono se le asigna una frecuencia básica. Así, el vector denotado por $f_r \in \mathbb{R}^4$ contiene las frecuencias para las 4 filas, mientras que $f_c \in \mathbb{R}^3$ contiene las frecuencias para las 3 columnas. La señal de sonido y asociada a la tecla ubicada en la posición (k, j) tendrá una duración de T segundos y se genera mediante la superposición de dos tonos fundamentales con frecuencias $f_r(k)$ y $f_c(j)$ respectivamente, es decir, $y = (y_1 + y_2)/2$, donde

$$y_1 = \sin(2\pi f_r(k)t) \quad \text{y} \quad y_2 = \sin(2\pi f_c(j)t), \quad \text{con } t \in [0, T].$$

Ahora bien, es necesario una discretización de la variable tiempo para así generar valores discretos (muestras) de y. En este contexto, se emplea una constante llamada *sampling rate*, que determina la cantidad de *muestras por segundo* que se tomarán, y que denotaremos por F_s. Específicamente, el intervalo $[0, T]$ se divide desde $t_0 = 0$ hasta $t_n = T$ con $\frac{1}{F_s}$ de separación entre cada medición.

Requerimientos

Tenga en cuenta que este es un mini(mini)-proyecto de investigación; y por tanto deben ser curiosos, plantearse preguntas, buscar información que ayude a responder esas preguntas y lo más importante: tener creatividad para resolver los requerimientos (y preguntas teóricas) indicadas en el enunciado.

[4]Upps! buuueno cuando usábamos el teléfono para marcar un número telefónico

[5]Dios santo! Acabo de descubrir que hay una canción que se llama DTMF por las siglas Debí Tirar Más Fotos! Creo que oír eso me produjo un *derrumbe* cerebral!

Específicamente se requiere que elabore una interfaz simple que:

1. Permita pulsar los dígitos en un teclado telefónico estándar (como el mostrado previamente). Cada pulso debe ir acompañado de: **(a)** el sonido asociado al dígito marcado; **(b)** la gráfica de la señal senoidal asociada (gráfica continua) y **(c)** la gráfica de las frecuencias que generan la gráfica continua (gráfica discreta).

2. Permita cargar una señal almacenada en un archivo externo asociada a un número telefónico de 11 dígitos; que usted deberá decodificar usando la transformada de Fourier para identificar cuáles fueron los dígitos marcados. En este caso, debe **(a)** reproducir la señal cargada; **(b)** decodificar la señal y mostrar los dígitos marcados; **(c)** graficar la señal cargada.

Consideraciones finales

Para efectos de este proyecto:

1. La duración de cada pulso será de 0.25 segundos (aprox), es decir, $T = 0.25$.

2. El *sampling rate* será de 32768, es decir, $F_s = 32768$.

3. Los valores de f_r y f_c son estándar y se encuentran fácilmente en la Web. Específicamente:

$$f_r = [697 \quad 770 \quad 852 \quad 941] \quad \text{y} \quad f_c = [1209 \quad 1336 \quad 1477]$$

Adicionalmente tenga en cuenta que:

- Para la interfaz simple se recomienda usar Tkinter[6]
- El modulo `fft` provee todo lo necesario para el manejo asociado al análisis de Fourier.
- El módulo `playsound` de Python permite el manejo de sonidos simples(como los de este proyecto).
- El proyecto debe ser realizado por dos personas máximo.
- En la fecha prevista deben entregar: (1) un **Informe del proyecto** (archivo `.pdf`) escrito en LaTeX[7] con las respuestas a las preguntas planteadas en el enunciado (respuestas razonadas). No se limiten a responder dichas preguntas de manera puntual: denles un contexto que haga coherente la lectura del informe. Adicionalmente, el informe debe contener la explicación del proceso de decodificación. (2) un **archivo fuente en Python** que contenga todos los códigos asociados a la solución de los requerimientos.
- El informe no debe exceder las 10 páginas y deben ser rigurosos con las citas a la bibliografía consultada (libros, sitios webs, etc). Adicionalmente, se tomarán en cuenta aspectos concernientes a la programación: eficiencia, facilidad de uso, etc.
- Fecha de entrega: **Miércoles 30/06/2025**

Grupo Docente de Cálculo Científico I / Realizado por M. Monsalve

[6] Es muy simple de usar

[7] Lo siento, la vida ya es bastante gris como para tener que leer textos que involucran expresiones matemáticas en Word. Les recomiendo *Overleaf* (editor en línea de LaTeX)