Laboratorio 1

Representación en Frecuencia

Joaquín Firpo, Bruno Moreira

Resumen - En este laboratorio se analiza el comportamiento de la serie de Fourier mediante simulaciones computacionales, comparando sus resultados con la teoría. Se exploran los aspectos fundamentales de esta herramienta, sus ventajas y limitaciones, presentando los resultados de manera gráfica y analítica.

Sección 1

I. INTRODUCCIÓN

Se desarrolló un código en Python para hallar las sumatorias correspondientes a la de la serie de Fourier [1] para comprobar si es posible realizar una representación computacional de la serie de Fourier que se aproxime a la señal original.

II. DESARROLLO

Problema 1

Para calcular los coeficientes de la serie de Fourier, primero es necesario encontrar series de intervalo infinito. Para ello, se creó un código capaz de calcular la convergencia de una sumatoria determinada como:

$$c = \sum_{k=-N}^{N} b_k$$

Se utilizó una estructura 'for', la cual se encarga de sumar los términos de b_k en el intervalo simétrico seleccionado N. Para modificar el intervalo se utilizaron estructuras condicionales 'if'.

Para verificar su funcionamiento se utilizaron tres casos de prueba:

$$C_1 = \sum_{k=0}^{N} \frac{1}{2^k}$$

$$C_2 = \sum_{k=1}^{-N} \left(\frac{1}{5}\right)^{-k}$$

$$C_3 = \sum_{k=1}^{N} \frac{(-1)^k}{|k|}$$

Como nuestro intervalo N no puede ser infinito, ya que esto nos llevaría mucho tiempo, se optó por tomar valores muy grandes de N y compararlos con la convergencia real de la serie infinita. Para lograr la representación de la serie adecuada, se utilizó la librería mpmath, la cual nos permite trabajar con más cifras decimales.

Para determinar cuántas cifras, se contempló el caso que más rápido tiende su valor limite, en este caso C_1 y cuando N es mayor. Teniendo en cuenta la ecuación 1 se necesitarán aproximadamente 3010 cifras .

$$Nlog10(2) \approx 0.30103N = 3010 \text{ cifras}$$

Ecuación 1: Cifras necesitas para alcanzar cierta precisión decimal.

N	10	100	1000	10000	Valor límite
C_1	-9.77e-4	-7.89e-31	-9.33e-302	-5.01e-3011	2
C_2	1.73e-17	1.73e-17	1.73e-17	1.73e-17	0.25
C_3	1.291	1.376	1.385	1.386	Diverge

Tabla 1: Verificación de convergencia de la serie implementada en 'Problema 1'.

Para calcular el valor límite de la sumatoria se tuvieron en cuenta las reglas de convergencia de las series infinitas, y se evaluó su diferencia con respecto a el valor obtenido por la sumatoria discreta.

Observando la (**Tabla 1**) podemos deducir que, si se toma un valor de N lo suficientemente grande, el programa implementado se aproxima al valor de convergencia real de la serie, sin embargo, para el caso C_2 esto no se observa. Para comprobar si efectivamente converge a lo esperado, se calculan diferentes ΔN mediante otra estructura 'for' y una lista la cual almacena los distintos valores los cuales serán comparados.

	ΔN(10,100)	ΔN(100,1000)	ΔN(1000,10000)
C_2	2.56e-8	3.169e-71	1.386e-700

Tabla 2: Verificación de convergencia de la serie C_2.

Con esta reinterpretación de los valores obtenidos podemos concluir que C_2 también converge a lo esperado.

Problema 2

Utilizando una versión simplificada del código anterior, ya estamos en condiciones de calcular la suma parcial de una serie de Fourier.

```python
import cmath
import numpy as np
import matplotlib.pyplot as plt
from math import pi, sin

def suma_parcial_fourier(t, f0, N):
    suma_total = 2/3
    for k in range(-N, N + 1):
        if k % 3 == 0:
            a_k = 0
        elif k != 0:
            a_k =(2*(cmath.exp(-k*(1/3)*pi*1j))*(sin(k*(1/3)*pi)))/(k*pi)
            suma_total += a_k * cmath.exp(2*pi*k*f0*t*1j)
    return suma_total

def señal_original(t):
    t_mod = t % 3  # Tomar la parte fraccionaria para el período
    if 0 <= t_mod < 1:
        return 2
    else:
        return 0
```

Código 2:Función que calcula la serie de Fourier

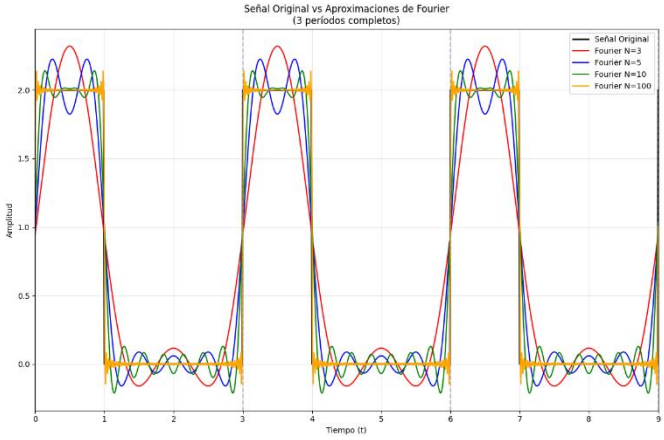
Se creo una función con la estructura anterior y con las variables de entrada, $f_0, N\ y\ t$.
Para calcular serie de Fourier se definieron los términos a_k como se muestra en la línea 12 del código 2, teniendo en cuanta (Ejercicio 6 del repartido 4). A su vez, se implementaron las librerías presentes en las líneas 1 a la 4 para facilitar la operación con números complejos y la implementación de graficas.
Para verificar que el programa funciona correctamente se prueban 5 valores de t diferentes variando el valor de N para aproximarnos al valor real para ello se utilizó otra estructura 'for' para los valores 10, 100, 1000 y 10000 en conjunto con la función presente en la línea 6.

t(s)	0.5	1	1.5	2	2.5
10	2.010	1.034	0.005	0.069	0.005
100	~2.000	1.004	5.455e-5	0.007	5.455e-5
1000	~2.000	~1.000	5.508e-7	0.001	5.508e-7
10000	~2.000	~1.000	5.513e-9	7.351e-5	5.513-9

Tabla 3: Valores de la serie de Fourier para N y t distintos.

Si se tiene en cuenta la señal cuadrada original x(t) del Ejercicio 6 se percibe que nuestra aproximación mediante la serie de Fourier concuerda con lo esperado.
Para observar esto de una forma más visual se optó por graficar las señales producidas por la suma parcial de Fourier para distintos valores de N, 3, 5, 10 y 100(grafica 1).

Para ello se definen algunos parámetros como el tiempo de inicio y final de la representación y los puntos a ser graficados, para definir t con un espaciado predeterminado. Es importante tener en cuenta que este procedimiento es necesario porque estamos intentando hacer una representación n de algo continuo con herramientas que funcionan de forma discreta, hay discretizar la variable del tiempo para ser capaces de representar los distintos valores.
Se implemento una función para graficar la señal original, la cual devuelve el valor constante 2 o 0 según t. Y se grafica usando la librería de mathplot.
Para calcular y ordenar las distintas representaciones de la serie de Fourier se utilizó una lista 'y_fourier[N]' y mediante un for y la función llamada 'suma_parcial_fourier' se crearon listas con los puntos a graficar.



Gráfica 1: Representación de serie de Fourier de x(t) para distintos valores de N **.**
Observando la gráfica 1 podemos concluir que la representación de la serie de Fourier realizada en el Problema 2 se aproxima a la función original y al aumentar N, esta aproximación parece tender a la función original como lo muestran los valores de la tabla 3.

Sección 2

Problema 1

Para evaluar si la señal obtenida mediante la serie de Fourier converge efectivamente a la señal original, se analiza la energía promedio del error en un período (ecuación 2). Si dicha energía tiende a cero, de acuerdo con la condición de convergencia de la Serie de Fourier (SDF), se puede concluir que $x(t)$ converge a su desarrollo en serie de Fourier [1].

$$e_N(t) = x(t) - \sum_{k=-N}^{N} a_k e^{Jktw}$$

Ecuación 2: Ecuacion de error para serie de Fourier.

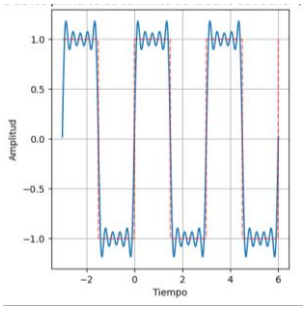
```python
#Calcular la convergencia de la serie en bace a la potencia promedio del error
def funcion_error(t_error, funcion_cuadrada, fourier_sum_error_real, T=3):
    # Recalcular la señal cuadrada y la suma de Fourier con el t refinado
    diferencia = funcion_cuadrada(t_error, periodo=T) - fourier_sum_error_real

    # Integral según Oppenheim (Sección 3.4)
    error = (1.0 / T) * np.trapezoid(diferencia**2, t_error)

    return error
```

Código 3: Función que calcula la serie de Fourier

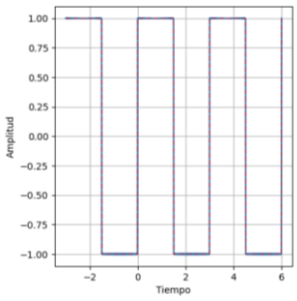
Se crea una función que se muestra en la línea 39 la cual mediante la función de trapecio y la definición de una función cuadrada es capaz de calcular la energía promedio de $e_N(t)$.
Es importante aclarar que se definió un intervalo de tiempo 't_error', el cual aumenta la densidad de puntos tomados para realizar la integral, ya que, si no, al aumentar los valores de k e integrar por un intervalo mayor se pierde información. Se decidió tomar un factor de '10k' para asegurarnos que esto no suceda.
A continuación, se presenta una tabla que muestran como varia el error al cambiar el valor de 'k'.

k	$E_{e_N(t)}$.
100	3.917-3
1000	3.918e-4
5000	7.837e-05

Tabla 4: Valores de la serie de $E_{e_N(t)}$ para distintos valores de 'k'.



k=10

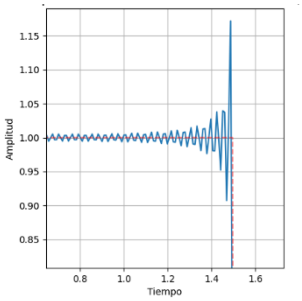


k=500

Gráfica 2: Suma de los primeros 10 y 500 términos de la serie de Fourier (Parte Real)

Al observar la tanto la tabla 4 como la gráfica 2, se percibe como al aumentar los valores de k, la señal efectivamente converge a la serie de Fourier calculada.

Problema 2:



Gráfica 3: Suma de los primeros 100 términos de la serie de Fourier (Parte Real), con aumento en la discontinuidad.

Al observar con más detenimiento que sucede con la gráfica 3 en los puntos de salto de la señal se concluye que presentan el fenómeno de Gibbs. Este ocurre cuando se intenta aproximar mediante el uso de series de Fourier los puntos de discontinuidad de las funciones [2].Este fenómeno no desaparece incluso al aumentar el número de términos de la serie; sin embargo, su amplitud de oscilación se estabiliza, permitiendo calcular que valor real tiene la representación de Fourier en ese punto, realizando el promedio entre los lados de la discontinuidad.

t(s)	Diferencia
1.47	9.2692e-2
1.48	2.2646e-2
1.49	1.71617e-1
1.50	1.575436

Tabla 5: Análisis de discontinuidades en la serie de Fourier.

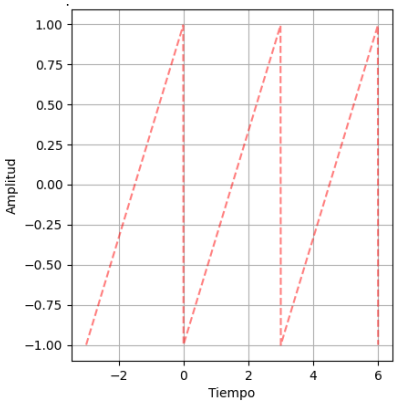
Para comprobar numéricamente que este fenómeno realmente ocurre en las discontinuidades. Se compara los valores de la señal y de la serie de Fourier previamente calculada en las cercanías del punto de interés en este caso se eligió la discontinuidad presente en t = 1.5.Si se observa la tabla 5, se presenta de forma numérica lo observado en el gráfico 3.

Problema 3:

```
8 def funcion_sierra(t, periodo=3.0):
9     # Función sierra: crece linealmente de -1 a 1
  en cada período
10     return 2 * ((t % periodo) / periodo) - 1
11     #Cada vez que t supera periodo, empieza de
  nuevo desde 0
```

Código 4: Función señal de sierra

Se definió una nueva señal, mostrada en la línea 8 del código 4, la cual, mediante una función que retorna el valor de la señal '2t' según en que punto múltiplo del periodo fundamental se encuentre, construye una señal de sierra como se muestra en la Gráfica 4.

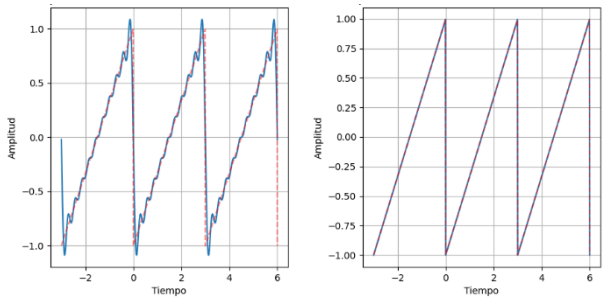


Gráfica 4: Señal de sierra con periodo 3.

Se repite el mismo procedimiento que el Problema 1 para la señal de sierra.

k	$E_{e_N(t)}$
100	1.948e-3
1000	1.958 e-4
5000	3.918 e-5

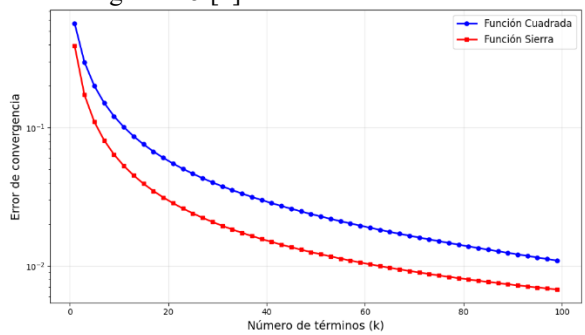
Tabla 6: Valores de la serie de $E_{e_N(t)}$ para distintos valores de 'k'.



k=10 k=500

Gráfica 5: Suma de los primeros 10 y 500 términos de la serie de Fourier para la señal de sierra.

Observando nuevamente la tabla 6 y la gráfica 5, se verifica como la serie de Fourier para una señal de sierra también converge a la señal original. Sin embargo, si se compara con el caso de la señal cuadrada parece que se necesitan menos valores de k para obtener una aproximación igual de precisa. Esto se deber a que la señal de sierra presenta más 'suavidad' esto se muestra en la gráfica 5 [2].



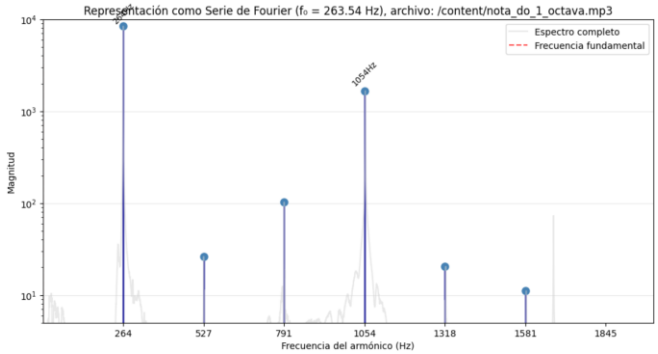
Grafica 5: $E_{e_{N(t)}}$ en función de k, para la señal de sierra y cuadrada.

Sin embargo, aunque en este caso parezca cumplirse, no es trivial definir la velocidad de convergencia de una serie de Fourier. No obstante, se observa que la derivada de una señal $x(t)$ siempre requiere más términos k para aproximarse con la misma exactitud que la señal original. Esto se puede demostrar directamente a partir de la definición de diferenciación de la serie de Fourier: $\frac{dx(t)}{dt} = jka_k$.

Sección 3

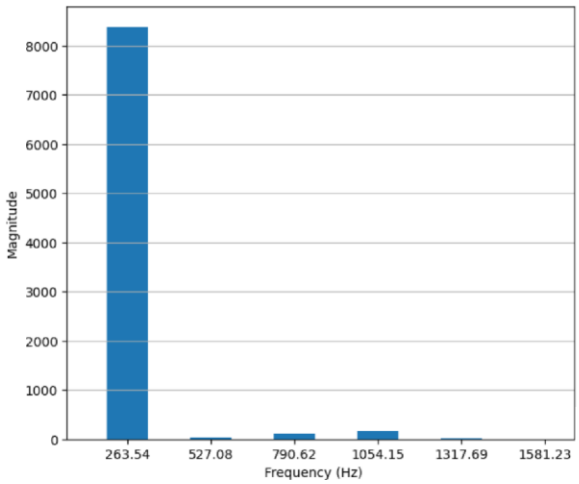
Problema 1

Con el programa provisto se analizaron dos notas, Do (1 octava) y La (2 octava), para el primer caso, el programa devuelve diversas gráficas. La siguiente grafica muestra la nota descompuesta según sus frecuencias.



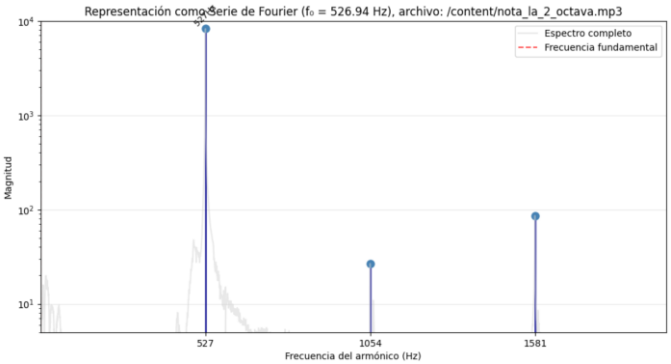
Gráfica 6: Descomposición en frecuencias de la nota Do

Como se puede apreciar en la gráfica 6 la nota Do está compuesta por una frecuencia fundamental de 264Hz , también puede observarse los demás armónicos de la nota con menores magnitudes.
En cuanto a sus armónicos, se observa que tiene una predominancia entorno a los 264Hz, el programa también ofrece un gráfico (7) con mayor precisión sobre esto.



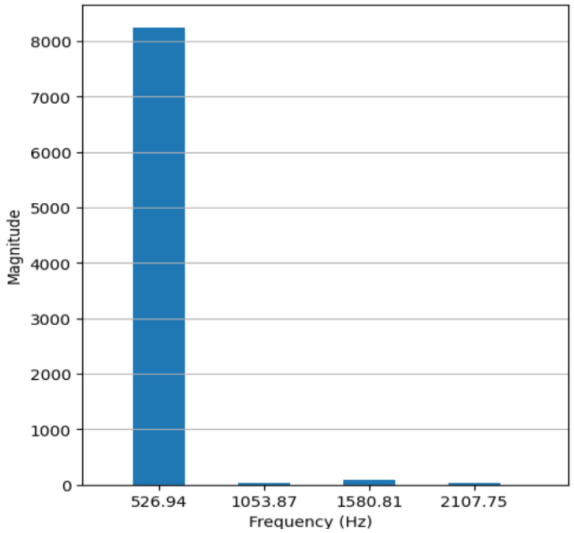
Gráfica 7: Contenido Espectral en Frecuencia Fundamental y Armónicos para la nota Do.

Con este gráfico se facilita la identificación del armónico dominante, pudiéndose observar que se halla en los 263,54Hz.
Ahora para la siguiente nota, tratándose de La (2 octava) con el programa, se calcula su serie de Fourier entregando la siguiente gráfica 8.



Gráfica 8: Descomposición en frecuencias de la nota La.

Se observa que, a diferencia de la nota estudiada anteriormente, esta se compone de menos armónicos, aunque alcanza una magnitud del mismo orden que se veía anteriormente.
Su frecuencia fundamental es de 527Hz
De igual forma que antes, para mayor precisión del armónico dominante se utiliza otro gráfico (9) provisto a continuación por el programa de análisis.

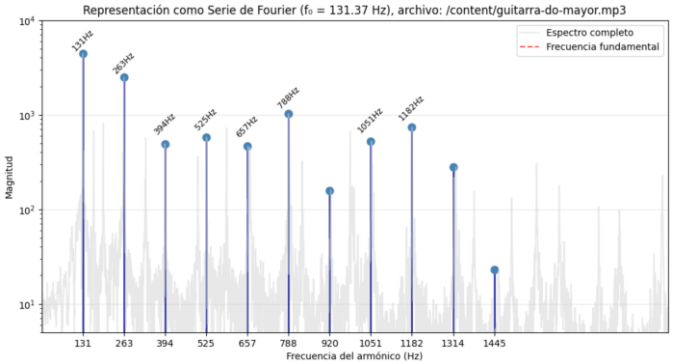


Gráfica 9: Descomposición en armónicos nota La

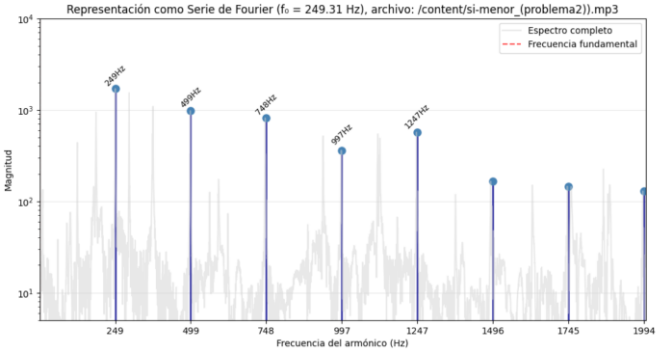
Esta grafica permite observar con una mayor claridad que frecuencia fundamental tiene la nota La y además permite determinar su armónico dominante, para frecuencia de 526Hz.

Problema 2

Mediante el mismo programa se analizaron dos acordes de guitarra, para posteriormente identificar que notas los componen y como se relacionan en frecuencia, los acordes estudiados son Do mayor y Si menor.
Para el acorde Do el programa calcula la siguiente descomposición.

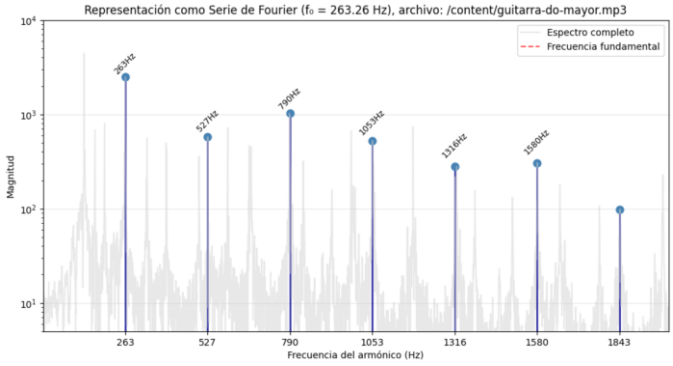


Gráfica 10: armónicos del acorde Do mayor con guitarra



Gráfica 11: armónicos del acorde Si menor con guitarra

Para determinar que notas compones dichos acordes, se debe identificar en el gráfico, que frecuencias tiene y descartar cuáles son sus armónicos.
En el grafico 10 se observa que están resaltados varios puntos, todos significan la misma nota ya que todos son múltiplos, se trata de la misma nota en varios armónicos, para este caso al comparar la frecuencia de 131Hz con la tabla de frecuencias musicales por octavas disponible en [3], se determina que el acorde se compone principalmente de la nota Do tercera octava (aunque el valor está apenas por encima).
El problema es que al investigar sobre la composición del acorde Do mayor se encuentra que consta de más notas que no fueron detectadas.
Para averiguar un poco más sobre la composición del acorde se repitió el proceso, pero modificando el rango de frecuencia del programa, de esta forma el programa podrá visualizar frecuencias desde 200Hz hasta los 2000Hz, obteniendo la siguiente grafica.



Gráfica 12: armónicos del acorde Do menor con acercamiento en el grafico

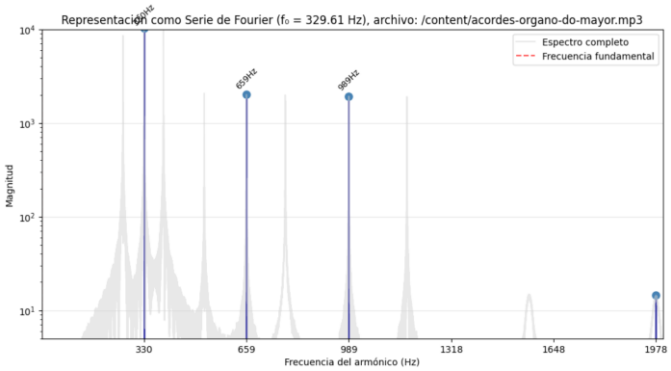
En el nuevo grafico se puede observar un poco más los armónicos de otras frecuencias fundamentales, lo que son las notas faltantes, a estos puntos no se le pueden determinar un valor exacto porque no están marcados, pero sabiendo que las notas faltantes son Mi y Sol, se puede afirmar que deben ser armónicos de las frecuencias 20,60Hz (Mi octava 0) o 24,49Hz (Sol octava 0)
Para el acorde de Si menor ocurre algo muy similar, se tiene varios múltiplos del armónico, tomando el fundamental se obtiene una frecuencia de 249Hz, al compararla con la misma tabla [3] utilizada anteriormente se concluye que es la nota Si tercera octava, nuevamente surge el problema de que al

investigar el acorde se encuentra que debería de tener más notas, por lo que bajo el mismo razonamiento los armónicos que no quedan marcados en el grafico 11 pertenecen a alguna de las notas faltantes Re(18,35Hz) y Fa(20,60Hz).

Observaciones
Para los acordes estudiados solo se pudo determinar con certeza una nota, sabiendo que el acorde Do mayor está conformado por las notas Do, Mi, Sol y que el acorde Si menor se conforma por Si, Re y Fa.
La poca visibilidad de las demás notas puede atribuirse a que al momento del programa calcular la transformada rápida de Fourier solo tome en cuenta los armónicos de mayor magnitud, teniendo solo una nota con claridad.

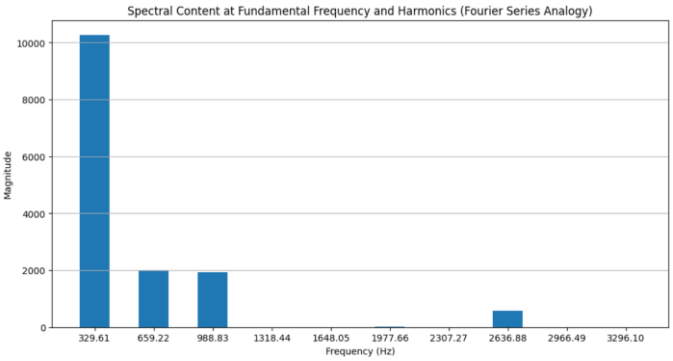
Problema 3

Para esta parte se seleccionó un acorde utilizado anteriormente, pero esta vez producido partir del instrumento órgano, con el acorde Do mayor en órgano se obtuvo la siguiente grafica donde se muestran los armónicos que tiene este acorde.



Gráfica 13: armónicos del acorde Do mayor en órgano

Como primera observación la gráfica 12 tiene un aspecto distinto de la gráfica 10, siendo que en ambas se trata del mismo acorde, la principal diferencia que el acorde del órgano no presenta tantos armónicos como con la guitara, pero en cambio tiene una frecuencia fundamental de 330Hz superando a la guitarra con 131Hz.
En la siguiente grafica se puede observar con una mayor precisión la frecuencia fundamental.



Gráfica 14: frecuencia y armónicos del acorde Do mayor

Observaciones
Para el mismo acorde se pudo observar que al tocarlo con una guitarra tiene una menor duración en el tiempo, es decir la magnitud de sus armónicos disminuye rápidamente atenuándose, algo propio de la vibración de las cuerdas de un instrumento, en cambio el órgano, un instrumento de viento si bien tiene una menor frecuencia no se presenta esta atenuación tan rápidamente como en las cuerdas.

Conclusión:

En este laboratorio se logró profundizar en el concepto de representaciones en frecuencia de señales periódicas a través de las series de Fourier, utilizando herramientas computacionales para su análisis y validación.

En primer lugar, se implementó un método en Python que permitió comprobar la convergencia de series infinitas, mostrando cómo al incrementar el número de términos N, los resultados obtenidos se aproximan al valor teórico esperado. Además, se pudo representar la serie de Fourier de diferentes señales, como la señal cuadrada y la de sierra, verificando sus diferencias en términos de suavidad y velocidad de convergencia.

Un aspecto relevante fue la identificación de las limitaciones de la serie de Fourier, especialmente en los puntos de discontinuidad donde aparece el fenómeno de Gibbs. Este efecto, aunque no desaparece, al aumentar el número de armónicos, permitió entender cómo se estabiliza la amplitud de las oscilaciones y cómo puede obtenerse un valor representativo mediante el promedio de los límites laterales.

Desde el punto de vista computacional, el uso de estructuras iterativas permitió visualizar la gran cantidad de cálculos necesarios para aproximar las series, destacando la importancia de métodos más eficientes como la Transformada Rápida de Fourier (FFT) para el análisis de señales más complejas.
Finalmente, la aplicación práctica del análisis espectral en el estudio de notas y acordes musicales facilitó la comprensión de estos conceptos teóricos, reforzando la idea de que la serie de Fourier no es solo una herramienta matemática, sino un modelo que describe fenómenos reales en ámbitos tan diversos como la ingeniería y la música.

Referencias

[1] A. V. Oppenheim and A. S. Willsky, *Signals and Systems*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 1997, ch. 3.
[2] K. Raeen, *A Study of The Gibbs Phenomenon in Fourier Series and Wavelets*, M.S. thesis, Dept. Mathematics, Univ. New Mexico, Albuquerque, NM, USA, 2008.
[3] S. Revuelta de la Peña, “Frecuencia de las Notas Musicales: Guía y Tablas (Hz),” Para Bajo Eléctrico. [En línea]. Disponible en: https://parabajoelectrico.com/frecuencia-notas-musicales/ [Accedido: 21-Sep-2025].

Apéndice

Todo el código producido se encuentra en el siguiente repositorio de GitHub:
https://github.com/BruniliomUY/Laboratorio-1---Representaci-n-en-Frecuencia-Se-ales-y-Sistemas-2025