Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ingeniería Eléctrica
IEE2103 – Señales y Sistemas

Tarea 2

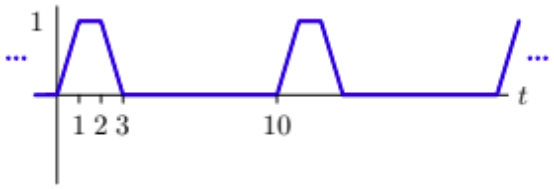
Entrega: 25 de septiembre a las 23:59 hrs.

Instrucciones generales

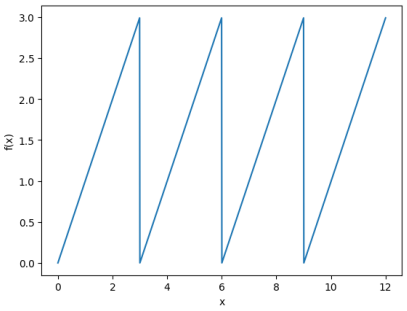
1. Para la parte de implementación, debe adjuntar un código desarrollado en Python (*.ipynb*).
2. Deben entregar un PDF con el desarrollo llamado *Apellido*_Tarea1.pdf (donde *Apellido* es su apellido).
3. La tarea es individual, pero pueden comentar las preguntas con sus compañeros.
4. Cada eje de los gráfico debe incluir un nombre descriptivo y su respectiva unidad.

Ejercicios Teóricos

1. **[2 puntos]** Determine la serie de Fourier de las siguientes señales periódicas:



(a) Primera Señal Periódica



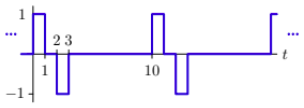
(b) Segunda Señal Periódica

Solución:

a. Lo primero que se debe saber es que si los coeficientes de la serie de fourier de $f(x)$ están dados por c_k, entonces los coeficientes de $\frac{df(x)}{dx}$ serán d_k tal que:

$$c_k = j\frac{2\pi k}{T}d_k$$

Entonces notamos que la función a) tiene como derivada la función:



Cuya serie está dada para $k \neq 0$ por:

$$c_k = \frac{j2}{\pi k} e^{-j3\pi k/10} \sin(\pi k/5) \sin(\pi k/10)$$

Por lo tanto:

$$c_k = j\frac{2\pi}{T} k d_k$$

Tal que para $k \neq 0$:

$$d_k = \frac{10}{\pi^2 k^2} e^{-j3\pi k/10} \sin(\pi k/5) \sin(\pi k/10)$$

Para $k = 0$ es cosa de calcular el valor medio de la función, donde se obtiene $k_0 = 1/5$.

b. Primero notamos que esta función tiene periodo $T = 3$ y amplitud $A = 3$.

Partiendo por a_0:

$$a_0 = \frac{1}{T}\int_0^T f(t)dt = \frac{1}{T}\int_0^T \frac{A}{T} \cdot t dt = \frac{A}{2}$$

Luego determinamos a_n, con $\omega = 2\pi/T$:

$$a_n = \frac{1}{T}\int_0^T f(t)dt = \frac{1}{T}\int_0^T t cos(n\omega t)dt$$

$$a_n = t \cdot \frac{sin(n\omega t)}{n\omega}\Big|_0^T + \frac{cos(n\omega t)}{(n\omega)^2}\Big|_0^T$$

Pero $\omega = \frac{2\pi}{T}$:

$$a_n = T \cdot \frac{sin(n\frac{2\pi}{T}T)}{n\frac{2\pi}{T}} - 0 + \frac{cos(n\frac{2\pi}{T}T) - 1}{(n \cdot \frac{2\pi}{T})^2}$$

$$= T \cdot \frac{sin(n \cdot 2\pi)}{n\frac{2\pi}{T}} + \frac{cos(n \cdot 2\pi) - 1}{(n \cdot \frac{2\pi}{T})^2}$$

Pero n es entero, por lo que el seno se anula y el coseno es igual a 1:

$$a_n = 0$$

A continuación calculamos b_n:

$$b_n = \frac{2}{T}\int_0^T \frac{A}{T} \cdot t \sin(n\omega t)dt = \frac{-A}{n\pi}$$

2. **[2 puntos]** Utilizando transformada de Fourier muestre si las siguientes expresiones son verdaderas o falsas:

I)

$$\int_{-\infty}^{\infty} \frac{\sin^2(x-a)}{(x-a)^2}dx = \pi \quad \forall a \in \mathbb{R}$$

Solución:

Podemos expresar la integral como:

$$\int_{-\infty}^{\infty} sinc^2(\frac{x-a}{\pi})dx$$

Recordando el par de Fourier:

$$sinc^2(x) \rightarrow \wedge(u)$$

Podemos concluir que

$$sinc^2(\frac{x-a}{\pi}) \rightarrow \pi e^{-i2\pi ua} \wedge (\pi u)$$

Y sabiendo la siguiente propiedad:

$$F(0) = \int_{-\infty}^{\infty} f(x)dx$$

Basta con evaluar la transformada de Fourier de nuestra función en 0. Esto es:

$$\pi e^{-i2\pi 0a} \wedge (0) = \pi \cdot 1 \cdot 1 = \pi$$

Por lo tanto, esta expresión es verdadera.

II)

$$\int_0^{\infty} \frac{\cos(\omega t)}{1+\omega^2}d\omega = \frac{\pi}{2}e^{-|t|}$$

Solución:
Al ser la función par, podemos expresar la integral como:

$$\frac{1}{2}\int_{-\infty}^{\infty}\frac{\cos(\omega t)}{1+\omega^2}d\omega$$

Definimos $f(\omega)$ y $g(\omega)$ para mantener el orden

$$g(\omega)=\frac{1}{1+\omega^2}$$

$$f(\omega)=\frac{\cos(\omega t)}{1+\omega^2}=\frac{1}{1+\omega^2}\cdot\cos(\omega t)=g(\omega)\cdot\cos(\omega t)$$

Recordamos la propiedad de modulación:

$$f(t)\cos(\alpha t)\xrightarrow{\mathcal{F}}\frac{1}{2}[F(u-\frac{\alpha}{2\pi})+F(u+\frac{\alpha}{2\pi})]$$

Entonces, sabemos que:

$$F(u)=\frac{1}{2}[G(u-\frac{t}{2\pi})+G(u+\frac{t}{2\pi})]$$

Por lo que tenemos que encontrar $G(u)$. Recordamos el par:

$$e^{-|t|}\xrightarrow{\mathcal{F}}\frac{2}{1+(2\pi u)^2}$$

y, por dualidad:

$$\frac{2}{1+(2\pi t)^2}\xrightarrow{\mathcal{F}}e^{-|-u|}=e^{-|u|}$$

Buscamos que esto se parezca a $g(\omega)$:

$$g(\omega)=\frac{1}{1+\omega^2}$$

$$g(2\pi\omega)=\frac{1}{1+(2\pi\omega)^2}$$

$$2\cdot g(2\pi\omega)=\frac{2}{1+(2\pi\omega)^2}$$

Entonces:

$$2\cdot g(2\pi\omega)\xrightarrow{\mathcal{F}}e^{-|u|}$$

y aplicando las propiedades de escalamiento y linealidad:

$$g(\omega)=\frac{2}{2}\cdot g(\frac{2\pi}{2\pi}\omega)\xrightarrow{\mathcal{F}}G(u)=\frac{1}{2}\cdot 2\pi e^{-|2\pi u|}=\pi e^{-|2\pi u|}$$

Reemplazamos $G(u)$ en la expresión que encontramos para $F(u)$:

$$F(u) = \frac{1}{2}[\pi e^{-|2\pi(u-\frac{t}{2\pi})|} + \pi e^{-|2\pi(u+\frac{t}{2\pi})|}]$$

Sabemos además que $F(0)$ corresponde a la integral de $f(\omega)$ en todo el dominio, por lo que evaluamos en $u = 0$:

$$F(0) = \frac{1}{2}[\pi e^{-|-t|} + \pi e^{-|t|}] = \frac{\pi}{2}[e^{-|t|} + e^{-|t|}] = \frac{\pi}{2}[2e^{-|t|}] = \pi e^{-|t|}]$$

Como esto corresponde a la integral sobre todo el dominio, la expresión dada es la mitad de este valor:

$$\frac{1}{2}\int_{-\infty}^{\infty} f(\omega)d\omega = \frac{\pi}{2}e^{-|t|}$$

Por lo tanto, la expresión es verdadera.

3. **[2 puntos]** Determine la transformada de Fourier de las siguientes funciones:

   I)
   $$f(t) = t \cdot e^{i\omega t}$$

   II)
   $$f(t) = t \cdot e^{-\pi t^2}$$

Solución:
Para resolver este problema vamos a usar la siguiente propiedad: sea $f(t) \to F(u)$ un par de Fourier, entonces $-i2\pi t f(t) \to F'(u)$. Esta propiedad se puede demostrar integrando por partes al calcular la transformada inversa de $F'(u)$.

I) $f(t)$ se puede expresar de la siguiente forma:

$$f(t) = \frac{1}{-i2\pi} \cdot -i2\pi t \cdot e^{i\omega t}$$

Luego, la tranformada de Fourier de $f(t)$ sería:

$$F(u) = \frac{1}{-i2\pi}(\mathcal{F}\{e^{i\omega t}\})'$$

Hay varías formas de determinar la transformada de una exponencial compleja, pero la más simple es notando que es parte de la base de Fourier, por lo que:

$$\mathcal{F}\{e^{i\omega t}\} = \delta(t - \frac{\omega}{2\pi})$$

Finalmente:

$$F(u) = \frac{i}{2\pi}\delta'(t - \frac{\omega}{2\pi})$$

II) $f(t)$ se puede expresar de la siguiente forma:

$$f(t) = \frac{1}{-i2\pi} \cdot -i2\pi t \cdot Gauss(x)$$

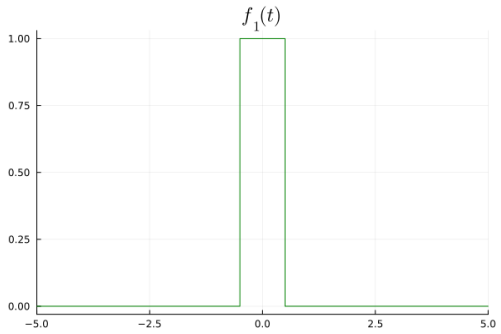
Más aún, sabemos que $Gauss(x) \rightarrow Gauss(u)$:

$$F(u) = \frac{i}{2\pi} Gauss'(u)$$

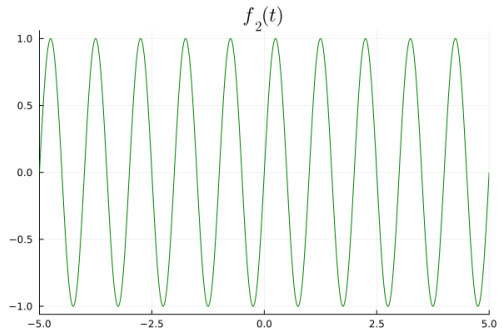
$$F(u) = \frac{i}{2\pi} \cdot -2\pi u \cdot Gauss(u)$$

$$F(u) = -iu \cdot Gauss(u)$$

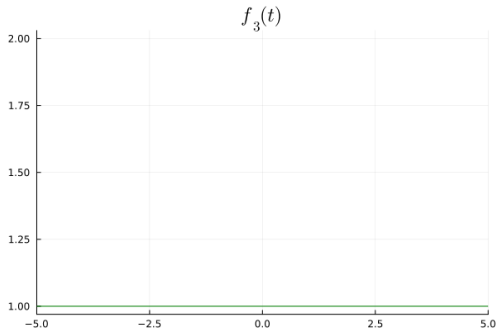
4. **[2 puntos]** A continuación se presentan 6 gráficos: tres son funciones en el tiempo y los otros tres son las transformadas de Fourier de estas funciones, calculadas numéricamente en código. Indique la correspondencia de cada función en el tiempo a cada función en frecuencia, teniendo en mente que una obtención numérica implica resultados no exactos. Justifique sus respuestas.



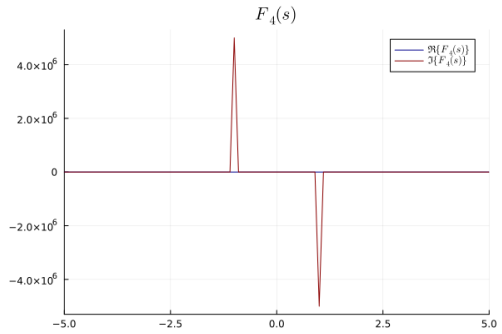
f_1



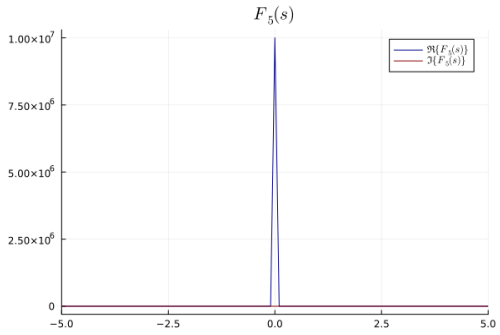
f_2



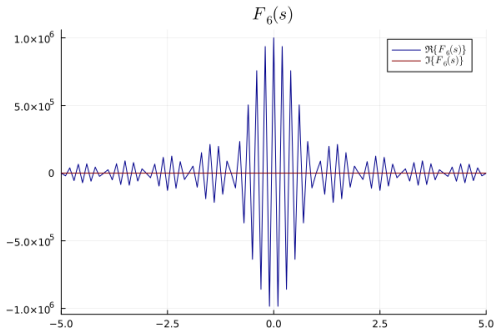
f_3



F_4



F_5



F_6

Las funciones graficadas corresponden a:

- $f_1(t) = \sqcap(t)$
- $f_2(t) = \sin(\pi t)$
- $f_3(t) = 1$

Cuyas transformadas teóricas son:

- $F_1(s) = sinc(s)$
- $F_2(s) = -i \uparrow\downarrow (s)$
- $F_3(s) = \delta(s)$

Con esto se pueden identificar las transformadas correspondientes en el gráfico, por lo que la respuesta sería:

- $f_1 \to F_6$
- $f_2 \to F_4$
- $f_3 \to F_5$

Ejercicios de Implementación

1. *a)* **[1 punto]** Considere la función $f(x)$ de la tarea anterior:

$$f(x) = \left(3 \cdot Gauss\left(\frac{x}{3}\right) - \sqcap\left(x - \frac{3}{2}\right) - \sqcap\left(x + \frac{3}{2}\right)\right) \cdot ramp\,(x + 3)$$

Para el mismo intervalo de x ($x \in [-6, 6]$). Investigue sobre la función convolve de numpy para calcular:

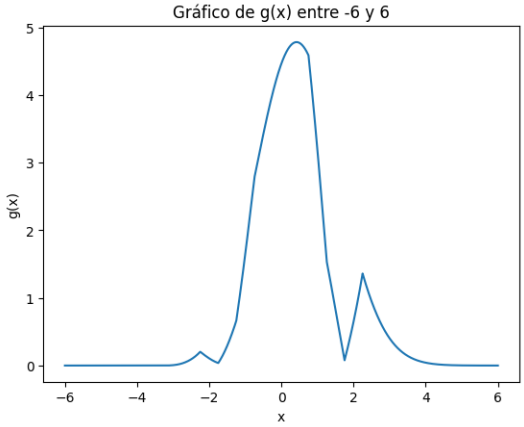
$$g(x) = f(x) * \sqcap(2 \cdot x)$$

Luego grafique $g(x)$ para $x \in [-6, 6]$.

Consideración adiconales:

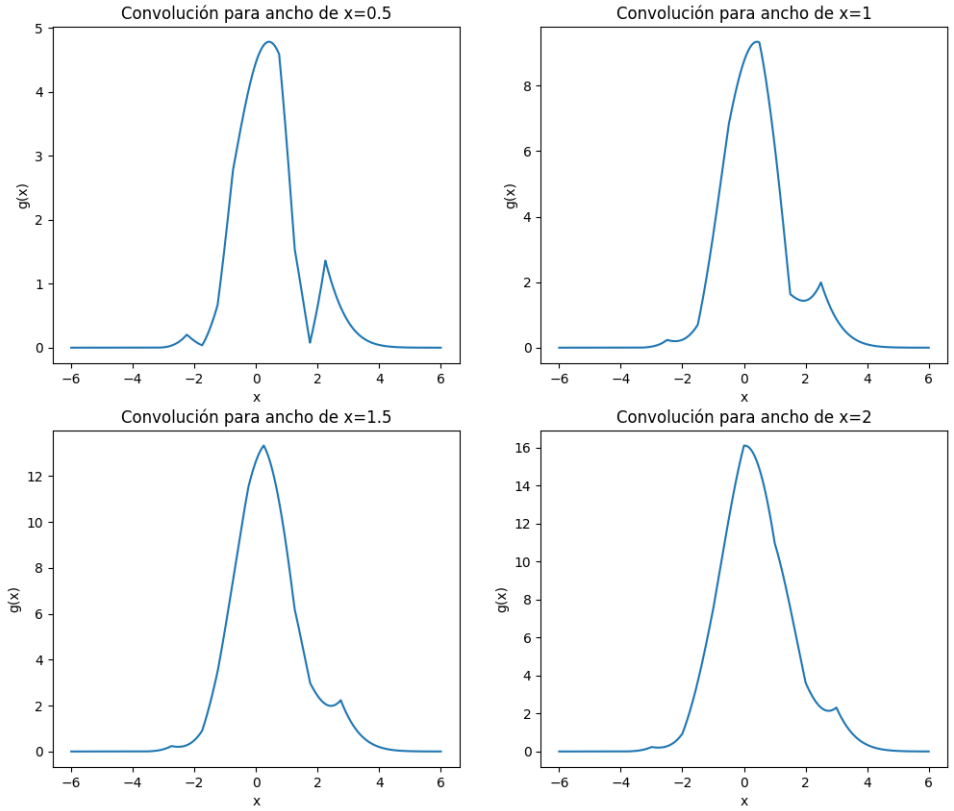
- Para el vector de x utilice un total de 12000 muestras.
- Para tener un resultado fidedigno con lo que se vería en el mundo continuo, pondere el resultado de la convolución por 0.001.
- Al usar la función convolve, utilice el modo *'same'* en la última entrada.

A continuación se presenta el gráfico de la convolución de $f(x)$ con un rect de ancho $x = 0.5$:



b) **[1 punto]** Repita el procedimiento para un rect de ancho $x = 1$, $x = 1.5$ y 2. Grafique las cuatro convoluciones realizadas en esta pregunta en una misma figura, y comente sobre el efecto de convolucionar una función con un rect. ¿Qué está pasando a medida que aumentamos el largo del rect?

A continuación se presenta la convolución de $f(x)$ con rects de distinto ancho:



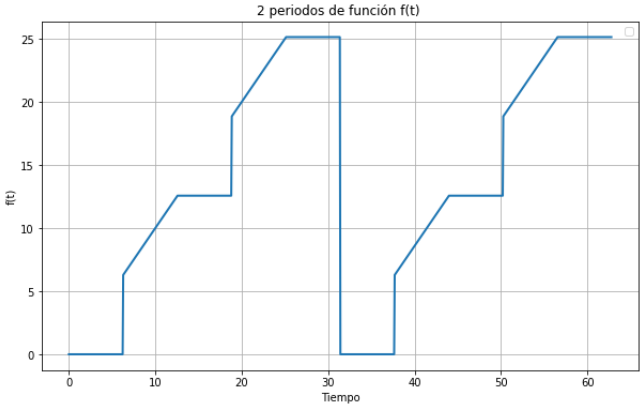
De la figura anterior notamos que al convolucionar una función con un rect esta se suaviza, filtrando componentes de alta frecuencia. Por ejemplo, notamos como el *peak* de la función original disminuye en tamaño. También notamos como a medida que aumentamos el largo del rect, la función se suaviza aún más.

2. Cree un código en Python en el que vaya siguiendo los pasos a continuación para poder obtener de manera numérica la serie de Fourier de una señal.

   a) **[1 punto]** Cree usando la libreria de Python numpy la siguiente función a tramos. Considere que f(t) es una señal periódica de periodo 10π.

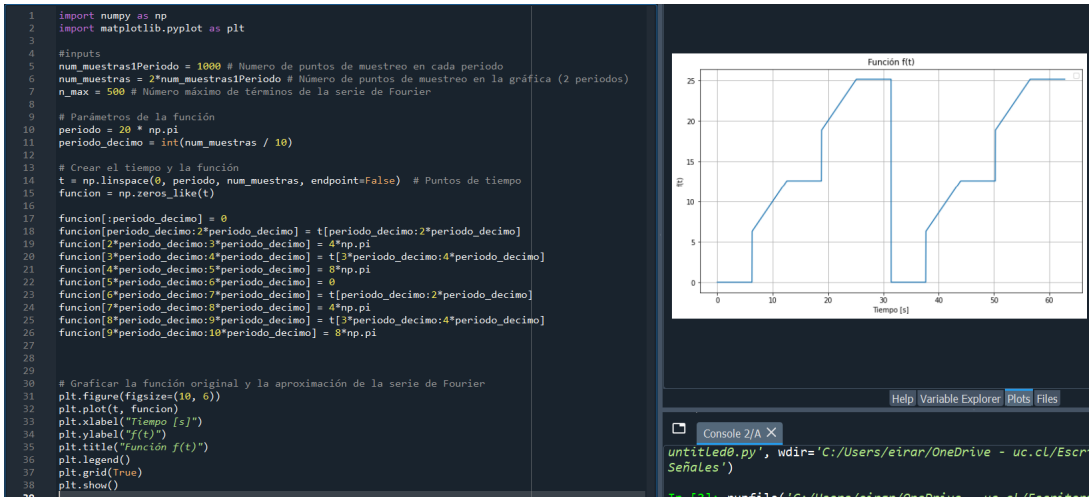
$$f(t) = \begin{cases} 0, & \text{si t} \in [0\pi : 2\pi] \\ t, & \text{si t} \in [2\pi : 4\pi] \\ 4\pi, & \text{si t} \in [4\pi : 6\pi] \\ t, & \text{si t} \in [6\pi : 8\pi] \\ 8\pi, & \text{si t} \in [8\pi : 10\pi] \end{cases}$$

   A continuación, grafique f(t) por dos periodos usando matplotlib, es decir, de 0 a 20π. Su gráfica debería quedar como la siguiente:



Incluya fotos de su gráfica resultante así como del código usado.

Solución:
La gráfica ya está hecha en el enunciado. Un posible código que la implemente sería el siguiente:



b) [**2 puntos**] La siguiente parte tiene como finalidad que usted aprenda a integrar de forma numérica. A continuación se muestra un código de ejemplo que integra la siguiente función:

$$IntegralDeseada = \int_0^2 10t\,dt$$

```python
import numpy as np

num_muestras = 1000
periodo = 2
t = np.linspace(0, periodo, num_muestras, endpoint=False)
funcion = np.zeros_like(t)
funcion = 10*t

suma_acumulada = 0
for i in range (num_muestras):
    suma_acumulada += funcion[i]*periodo/num_muestras
print(suma_acumulada)

```

El anterior código realiza una aproximación numérica a la integral deseada. El resultado que se imprimirá con este código será de 19.8, pero sabemos que el valor correcto es de 20. Vemos que en la línea 3 se define que la aproximación se realizará tomando 1000 muestras. Si este número se aumenta, el error cometido entre el valor obtenido y el valor real disminuirá.

Modifique el código anterior para que calcule la siguiente integral:

$$\int_0^{5\pi} t^2 \cdot \cos(t)\, dt$$

Muestre una foto del código escrito, y especifique el valor obtenido cuando el número de muestras es de 1000, 10000 y de 100000. ¿Qué puede decir al respecto?

Solución:

Una posible implementación de código sería la siguiente:

```python
import numpy as np

periodo = 5*np.pi
num_muestras_lista = [1000,10000,100000]
valor_real = -10*np.pi #Obtenido con Wolfram u otro programa

for num_muestras in num_muestras_lista:
    t = np.linspace(0, periodo, num_muestras, endpoint=False)
    funcion = np.zeros_like(t)
    funcion = t**2*np.cos(t)
    suma_acumulada = 0
    for i in range (num_muestras):
        suma_acumulada += funcion[i]/num_muestras*periodo
    print("Para " + str(num_muestras) + " muestras La aproximación da " + str(float(f"{suma_acumulada:.{3}f}")))
    error = abs((suma_acumulada-valor_real)/valor_real * 100)
    print("Para " + str(num_muestras) + " muestras el error con respecto al valor real es de "+str(float(f"{error:.{3}f}"))+" %")

```

Con el siguinete output, el cual claramente muestra que más meustras agrega precisión al resultado obtenido.

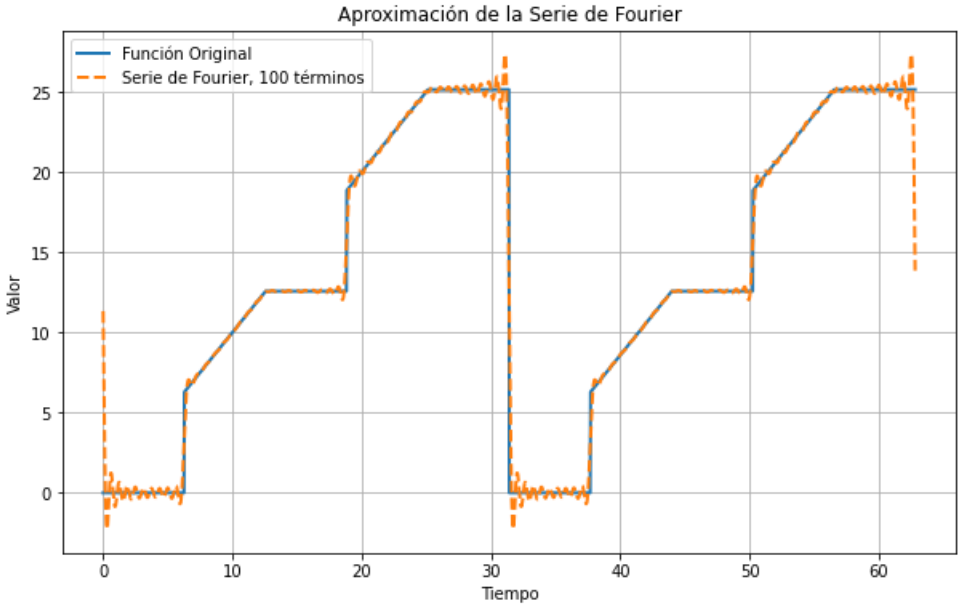
```
Para 1000 muestras la aproximación da -29.479
Para 1000 muestras el error con respecto al valor real es de 6.166 %
Para 10000 muestras la aproximación da -31.222
Para 10000 muestras el error con respecto al valor real es de 0.617 %
Para 100000 muestras la aproximación da -31.397
Para 100000 muestras el error con respecto al valor real es de 0.062 %
```

c) [**3 puntos**] Ahora se le pide implementar un código que calcule la Serie de Fourier de la función f(t) del ítem a).
Para calcular los coeficientes tendrá que integrar numéricamente. Hágalo parecido a lo que aprendió a hacer en item b). Dentro de la integral de los coeficientes, se encuentra la función ponderada por una exponencial compleja. Para la función considere usar el código ya creado en item a), y para la exponencial compleja, la librería numpy puede serle útil para tratar con números complejos.
Se recomienda fuertemente usar la formulación compleja de la Serie de Fourier. De esta manera, usted sólo deberá calcular una constante (c_k) y no dos (a_k y b_k). Para efectos de la correción, se exigirá que cada vez que se pida truncar la serie al "k-ésimo"término, se consideren todos los coeficientes que hayan desde c_{-k} hasta c_k. Por ejemplo, si se pide encontrar la Serie de Fourier considerando los primeros 100 términos, el código debe calcular el valor de los 201 pesos que hay entre c_{-100} y c_{100}. Todos los demás pesos serán cero para llevar a cabo la aproximación.

Hint: Investigue cómo puede serle útil la información de que la función a aproximar sea puramente real. Además cuenta con la función np.real() que permite extraer la parte real de un número complejo. Si usted desarrolla correctamente la matemática se dará cuenta de que la combinación de ambas cosas le permitirá olvidarse de los c_k con k negativos, y así simplificar la serie.

1) En primer lugar, investigue qué es el fenómeno de Gibbs y descríbalo brevemente.

2) Se muestra como ejemplo a continuación una aproximación a la función f(t) considerando 1000 muestras por periodo, considerando los primeros 100 términos de la Serie de Fourier.



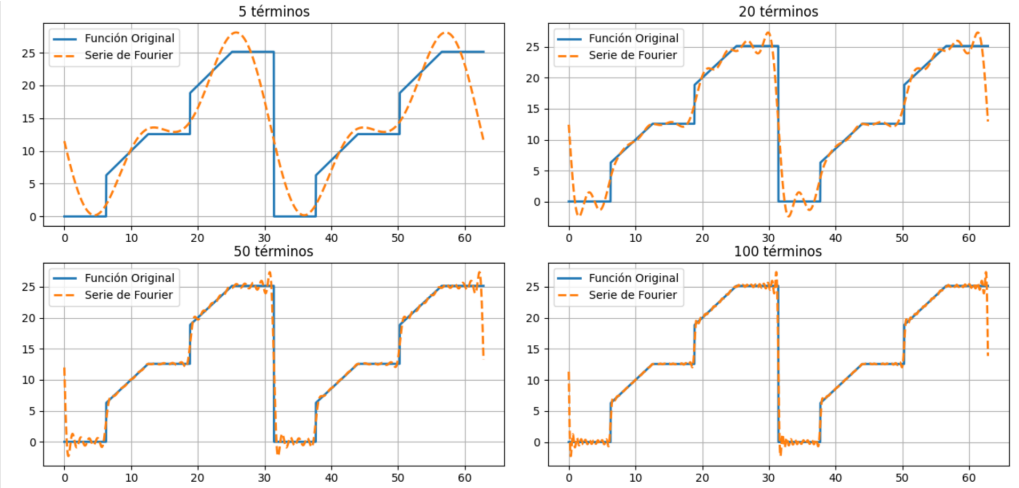
Cree su propio código y calcule con él la misma Serie de Fourier, pero para los primeros 5, 20 y 50 términos. Debe dejar fija la cantidad de muestras por periodo en 1000. Explique qué está pasando con el error entre la función aproximante y la función real. ¿Cómo se manifiesta el fenómeno de Gibbs en cada caso?, ¿En qué puntos del dominio se hace más notable?
Adjunte fotos de su código así como los gráficos que considere interesantes para su explicación.

3) Disminuya la cantidad de muestras por período a 50. Vuelva a graficar para 5, 20 y 50 términos de la serie. ¿Qué diferecias nota?
4) ¿Cuál cree usted que son las ventajas y desventajas de trabajar con 1000 muestras por periodo v/s trabajar con 50 muestras?

Solución:

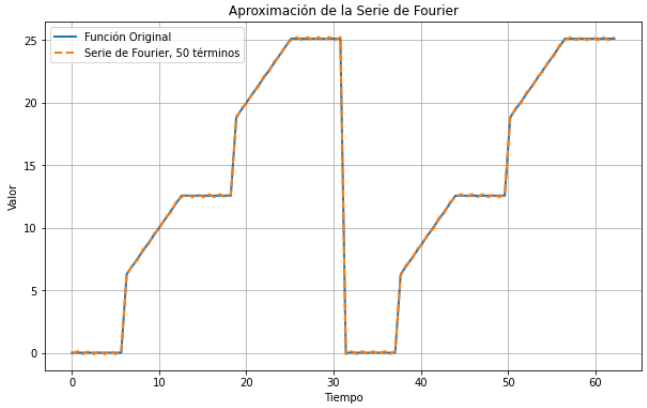
1) El fenómeno de Gibbs es un efecto que se observa cuando se intenta aproximar una señal discontinua o brusca mediante una serie de Fourier finita (se trunca a parti de cierta frecuencia muy alta). Este efecto se manifiesta como oscilaciones o picos alrededor de los puntos donde ocurre la discontinuidad de la señal original.

2) El código se adjunta por separado.



A medida que aumentamos la cantidad de coeficientes que consideramos, el error disminuye, llegando a irse prácticamente a cero en aquellas zonas continuas, mientras que en el vecindario de las discontinuidades el fenómeno de Gibbs hace que todavía quede bastante error pese a tener muchos términos la aproximación por Series de Fourier.

3) Para los casos de 5 y 20 términos casi no hay diferencia. Sin embargo, para el caso de 50 términos, la serie ahora parece aproximar de forma perfecta a la función sin importar los términos de frecuencia superior que fueron truncados.



El fenómeno de Gibbs no se observa porque estamos en presencia de un submuestreo, fenómeno que se verá con mayor detención más adelante en el curso cuando se vean las señales discretas. Esto quiere decir que se tienen tan pocas muestras para graficar, que no se logra dar suficiente detalle a la función en estas condiciones. De esta manera, las frecuencias más altas (las que se observarán en las discontinuidades y que generán el fenómento de Gibbs) no

tienen suficiente resolución en el tiempo como para lograrse ver en el plot, pues sólo se cuenta con 50 muestras para todo el intervalo.

4) Dentro de las ventajas de usar muchas muestras por sobre pocas están una mayor precisión, y por ende un menor error al discretizar la señal continua. Además se logra una mejor calidad visual. Sin embargo, las desventajas de usar muchas muestras es una mayor carga computacional y un consumo de memoria que podría resultar innecesario dependiendo del contexto. Sin embargo, el mayor riesgo de usar pocas muestras es el risgo de 'perder información de la señal original' al momento de muestrearla.

d) [**2 puntos**] [**Bonus**] Cree usted su propia función a tramos (como la desarrollada durante todo este ejercicio, pero una original) y calcule usando el mismo método su Serie de Fourier para 1000 muestras por periodo para los primeros 100 términos. Considere que el puntaje bonus que se le asignará dependerá de la creatividad que tenga a la hora de elegir la función que aproximará.

Solución:
Aquí la respuesta es bastante libre y lo que se premiará será usar correctamente el código ya creado para calcular la serie de Fourier de una señal creativa creada por usted. Sin embargo, a modo de ejemplo de lo que pudieron haber implementeado, se ajunta un código que lo que hace es generar de manera aleatoria una señal a partir de la suma de 10 sinusoides de amplitud, frecuencia y posición aleatoria. Posteriormente, se procede a calcular 4 aproximaciones distintas con una cantidad variable de términos en la respectiva Serie de Fourier. De esta forma se puede ver visualemente cuántos términos hacen falta para aproximar correctamente la señal original.