EXPANSIÓN DE SEÑALES PERIÓDICAS EN SERIES DE FOURIER MEDIANTE PYTHON

[1] Juan Manuel Juárez Rodríguez, m21371210@apizaco.tecnm.mx
[2] Ana Patricia Jaimes Marcelo, m21371210@apizaco.tecnm.mx[3]
Raúl Cortés Maldonado, raul.cm@apizaco.tecnm.mx
[4] Haydee Patricia Martínez Hernández, hayde.mh@apizaco.tecnm.mx

RESUMEN

La expansión en series de Fourier es una herramienta matemática que permite resolver problemas complejos de la ingeniería, por ejemplo, para resolver problemas de vibraciones de cuerdas, transferencia de calor, procesamiento de señales, entre otros. El objetivo fundamental de las series de Fourier es descomponer señales periódicas, continuas o a trozos, en términos de una combinación lineal de señales sinusoidales y cosenoidales. En este trabajo se presenta un estudio descriptivo-cualitativo de alcance explorato- rio, que propone una herramienta didáctica computacional para la reconstrucción de señales periódicas utilizando las series trigonométricas de Fourier, aplicando el sistema de álgebra computacional de Python para obtener los coeficientes a_0, a_n y b_n,ra señales periódicas de hasta cuatro trozos definidas por el usuario. Asimismo, se presenta una propuesta metodológica para aplicarla en cursos de nivel superior para el estudio analítico y gráfico de la expansión en series de Fourier de señales periódicas definidas a trozos.

PALABRAS CLAVE

Coeficientes de Fourier
Python
Señal a trozos
Expansión

ABSTRACT

ABSTRACT: Fourier series expansion is a mathematical tool that allows solving complex engineering problems, for example, to solve problems of string vibrations, heat transfer, signal processing, among others. The fundamental objective of the Fourier series is to decompose periodic signals, continuous or piecewise, in terms of a linear combination of sinusoidal and cosine signals. In this work, a descrip- tive-qualitative study of exploratory scope is presented, which proposes a computational didactic tool for the reconstruction of periodic signals using Fourier trigonometric series, applying the Python computational algebra system to obtain the coefficients a_0, $a_n$andb_n, periodic signals of up to four user-defined chunks. Likewise, a methodological proposal is presented to apply it in higher level courses for the analytical and graphical study of the Fourier series expansion of piecewise defined periodic signals.

KEYWORDS

Coefficients of Fourier
Python
Chunky signal
Expansion

1,2 *TecNM-Instituto Tecnológico de Apizaco/Estudiantes*
3,4 *TecNM-Instituto Tecnológico de Apizaco/Docentes*

I. INTRODUCCIÓN:

La formación de los estudiantes de ingeniería se caracteriza por los fundamentos matemáticos y físicos que deben comprender y dominar desde los primeros ciclos del nivel superior. En particular, el tema de series trigonométricas de Fourier representa una gran dificultad de comprensión para los estudiantes, ya que involucra la resolución de integrales complejas, así como el gráfico de funciones generado por otras funciones (Gonzalez Avid y Lopez Gianmarco, 2021).

El uso de las tecnologías y herramientas didácticas para la solución de este tipo problemas permite que el estudiante adquiera una mayor comprensión del tema debido a las prácticas y escenarios con lo que se ve involucrado, además de lograr conseguir analizar un mayor número de ejemplos sin la necesidad de dedicarle un tiempo desmesurado a la solución de problemas que representen una mayor complejidad como es el caso de funciones de más de tres trozos (Giménez-Palomares et al., 2018), (Maggiolini et al., 2017).

Las funciones periódicas pueden representarse por medio de series de Fourier (Suslov, 2002), su aplicación es amplia en el ramo de la ingeniería, por ejemplo, (Marturet P. et al., 2015) modelaron potenciales energéticos en turbinas helicoidales, donde la posición angular y la oscilación del torque de la turbina son resueltas mediante una función de series infinitas sinusoidales partiendo de las series de Fourier, por otro lado (Cruz Sanabria et al., 2021) en su trabajo realizaron métodos de interpolación Series de Fourier y Whittaker para el monitoreo de cultivos mediante el procesamiento de imágenes y el uso de los dos métodos antes mencionados, obtienen series temporales de datos continuos a partir de datos discontinuos. Por su parte (G. de Carvalho et al., 2005) realizaron un modelo para predecir la productividad de cultivo de café donde utilizó un análisis armónico por series de Fourier y a partir de ello calcularon los coeficientes de Fourier par finalmente realizar la suma de siete sumas parciales, sin embargo, su modelo no resultó adecuado debido a presentó errores enla estimación.

Realizar el análisis y cálculo de coeficientes de Fourier de forma escrita, así como las integraciones que involucran para obtener su correspondiente expansión den series de Fou- rier, resulta sumamente complicado para funciones definidas a trozos que consideran la presencia de funciones rampa o sinusoidales, sobre todo cuando se desea obtener una suma parcial mayor a tres. Por este motivo, surge la necesidad desde el punto de vista didáctico el ejecutar estas operaciones a través de un código de programación. El enfoque de este trabajo está dedicado al aspecto practico y didáctico, desarrollando así una herramienta computacional que resulte de utilidad para los estudiantes de licenciatura y posgrado, de tal manera que permita analizar problemas complejos de cursos introductorios de Análisis de señales sin necesidad de desarrollarlos de forma escrita a detalle durante las actividades de aula.

En este trabajo se presenta un estudio descriptivo-cualitativo de alcance exploratorio, que propone una herramienta didáctica computacional para la reconstrucción de señales periódicas utilizando las series trigonométricas de Fourier, aplicando el sistema de álgebra computacional de Python para obtener la forma analítica de los coeficientes a_0, a_n y b_n, para señales periódicas de hasta cuatro trozos definidas por el usuario. Asimismo, se presenta una propuesta metodológica para aplicarla en cursos de nivel superior para el estudio analítico y gráfico de la expansión en series de Fourier de señales periódicas definidas a trozos.

El contenido del documento se distribuye de la siguiente manera, la sección Metodología describe los fundamentos teóricos de las series de Fourier, así como el diagrama de flujo para el código desarrollado en Python, el cual toma como referencia aquel propuesto por (Luna, 2020). En la sección Resultados se presenta un ejemplo desarrollado siguiendo la metodología propuesta para la expansión en series de Fourier de la función $f(x)$ definida a cuatro trozos, así como la obtención de la suma parcial seis ($n = 6$). Así mismo, se presentan las gráficas de la reconstrucción obtenida para 3 funciones definidas con cuatro trozos distintos entre sí, describiendo el procedimiento para ingresar dichas funciones en la herramienta didáctica. Finalmente se presentan las conclusiones sobre el trabajo desarrollado.

II. METODOLOGÍA

Marco Teórico

Las series de Fourier constituyen una importante herramienta para la obtención de soluciones de ecuaciones diferenciales. Su teoría básica concierne a la descomposición de una función periódica ***f(t)*** en términos de una suma de funciones senoidales y cosenoidales de diferentes frecuencias, ésta última siendo un múltiplo de la frecuencia de la señal original (Cruz Sanabria et al., 2021); esto es, la función ***f(t)*** se puede descomponer en una serie armónica infinita.

La expansión en series de Fourier en su forma trigonométrica se obtiene mediante la expresión:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n cos(n\omega_0 t) + b_n sen\ (n\omega_0 t\ ) \quad (1)$$

donde f(t) es una señal periódica de periodo T expresada en el dominio del tiempo, $\omega_0 = 2\pi/T$ es la frecuencia fundamental, el entero n tiene unidades de Hz y, los términos a_0, a_n y b_n, son los coeficientes trigonométricos de Fourier, los cuales se calculan a partir de las siguientes expresiones el intervalo de un periodo T:

$$a_0 = \frac{2}{T} \int_T f(t)dt, \quad (2)$$

$$a_n = \frac{2}{T} \int_T f(t)\cos\left(n\omega_0 t\right)dt, \quad (3)$$

$$b_n = \frac{2}{T} \int_T f(t)\mathrm{sen}\left(n\omega_0 t\right)dt. \quad (4)$$

Si bien le Ec. (1) no es la más conveniente para el análisis de señales y otros problemas complejos, esta expresión es el primer acercamiento que tienen los estudiantes de cursos introductorios de Análisis de señales de las carreras de electrónica y Tecnologías de la Información y Comunicaciones, por mencionar algunas (TecNM/ITApizaco, 2022). En este sentido y considerando el punto de vista didáctico, resulta de gran relevancia aplicar herramientas computacionales que permitan a los estudiantes centrarse en el análisis y aplicaciones de las series de Fourier, en vez de dedicar tiempos considerables para el cálculo de integraciones cuyos procesos pueden resultar largos y complejos; en particular, cuando se tienen funciones definidas en más de dos trozos.

Herramienta didáctica desarrollada en Python

Python es un lenguaje de programación de amplio campo de aplicación y practica de trabajar, su sintaxis es considerada de las más completas y entendibles, ya que en ocasiones parece pseudocódigo. Por esta y otras razones más, su uso se ha incrementado notablemente en diferentes áreas de trabajo. Además, es importante mencionar que Python es un lenguaje de programación libre con acceso a gran cantidad de librerías que lo hacen un programa fiable de utilizar, se conoce como un desarrollador de alto nivel y multiplataforma, ya que puede ejecutarse en diversas distribuciones como son Linux, Windows, MacOS, entre otras (Ivet Challenger Perez, 2014). Fue creado por Guido Van Rossum a inicios de los 90´s, inicio con similitud al lenguaje de programación Perl, pero cuidando siempre una estructura y una sintaxis limpia.

El programa para calcular los coeficientes trigonométricos de Fourier y, de esta manera obtener la expansión en Se- ries de Fourier, Ecs. 1-4, se realiza mediante el lenguaje de código abierto de Python, el cual contiene una gran cantidad de librerías para el análisis matemático, tomando como referencia la diversidad de ejemplos existentes para el tema bajo consideración y adaptándolo a nuestras necesidades (Luna, 2020), (Arias Hernández et al., 2016).

En la Fig. 1 se muestra el diagrama de flujo para desarrollar el programa que calcula las expresiones analíticas de las Ecs. 2-4. En general, proponemos cinco bloques que definen a nuestro código fuente, cada bloque realiza una tarea específica que se describe a continuación.

Bloque I: Se importan las librerías a utilizar y las variables simbólicas (n y t), se comienza con import seguido del nombre de la librería.

Bloque II: se declara el intervalo de la función, el periodo T, la frecuencia angular ω y las funciones a trozos. Para definir la función $f(t)$ se utiliza la función picewise de Python de la librería **sympy**, si se desea ingresar una nueva función a trozos, los valores que deben ser modificados comienzan en A de la figura 2 ingresando el intervalo mínimo y máximo T1 y T2 respectivamente, después, en B se ingresan las funciones de cada trozo siendo F1, F2, F3 y F4 los trozos que define la función $f_1(t)$, finalmente, en C, se ingresan los intervalos para los cuales es válido cada trozo de la función. El procedimiento para el cálculo de la n suma parcial descrito en la Fig. 2 se repite para cualquier función a trozos que desee ingresar, modificando los valores únicamente en este bloque.

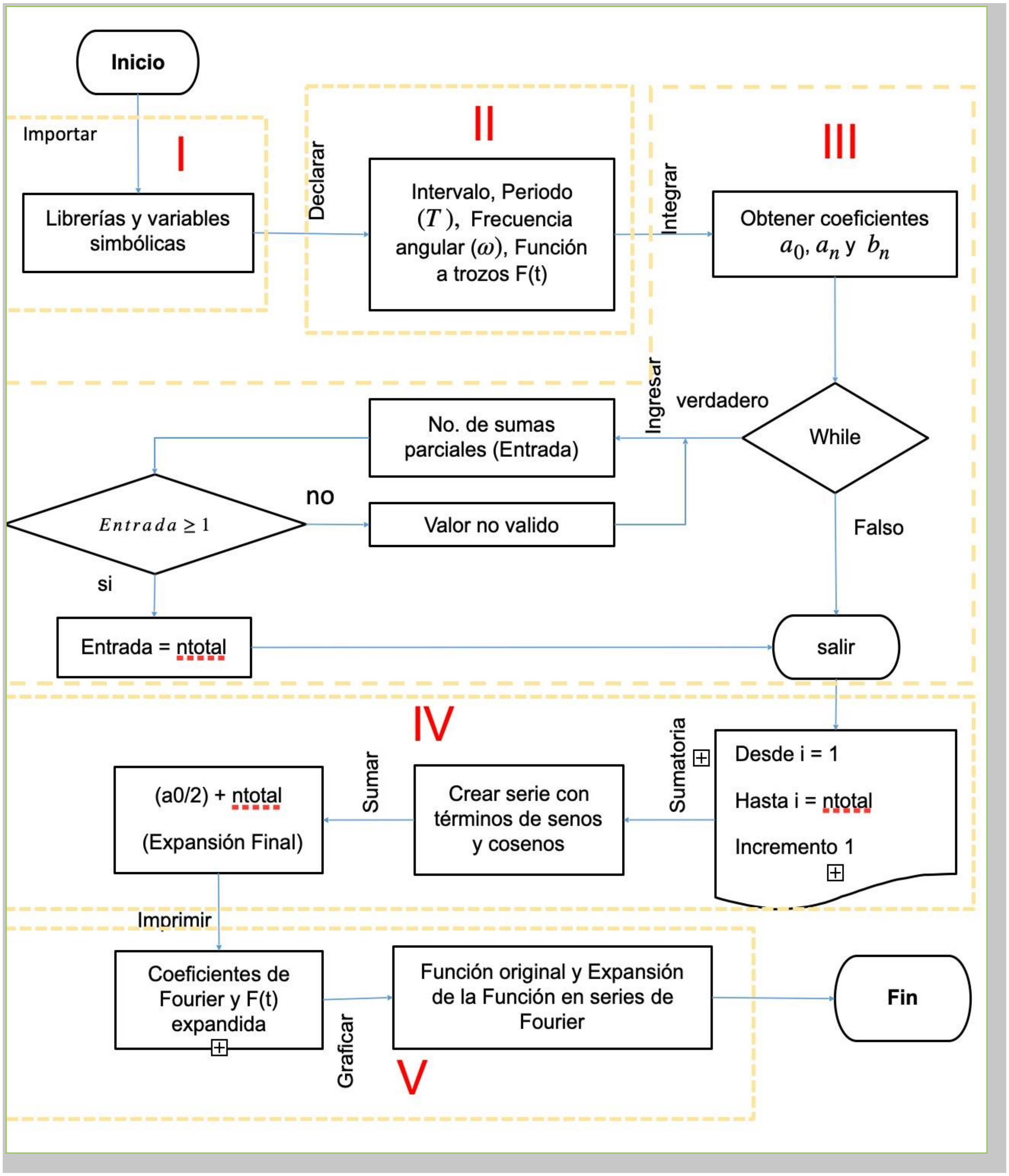


Figura 1.
Diagrama de flujo para calcular los coeficientes trigonométricos de Fourier para una señal periódica f(t).

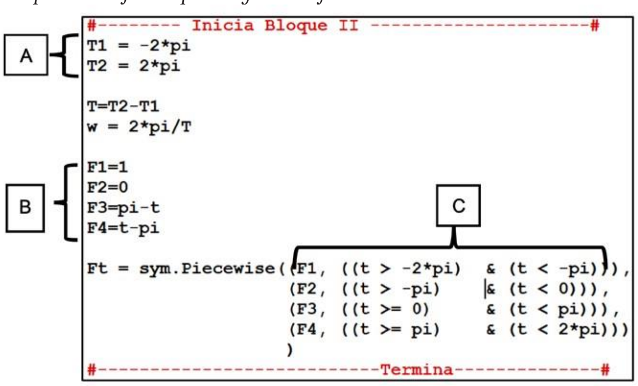
Bloque III: la función $f(t)$ se renombra como “f_integral” y se calcula el valor de los coeficientes mediante el proceso de integración, posteriormente se crea una variable llamada “Entrada” iniciándola con un valor igual a cero. Posteriormente, se crea una condicional dentro de un ciclo **while**, que pide que sea teclado un numero entero positivo de sumas parciales a realizar guardando el valor en la variable “Entrada”, si la condición se cumple sale del bucle y devuelve el valor numérico y lo guarda en una nueva variable llamada “ntotal”, la cual índica el número de sumas parciales a realizar en la serie de Fourier.

Boque IV: se crea una variable llamada “expansion” con un valor inicial igual a cero, enseguida se crea un ciclo **for** que realiza iteraciones desde 1 hasta el valor ingresado, posteriormente se evalúa para cada valor, y finalmente se realiza la sumatoria de los a_n y b_n, más $\frac{a_0}{2}$ para crear la expansión en series de Fourier.

Bloque V: se imprimen los valores de a_0, a_n y b_n, y $f(t)$ expandida en series de Fourier; también se grafica la función original y la función expandida.

Figura 2.

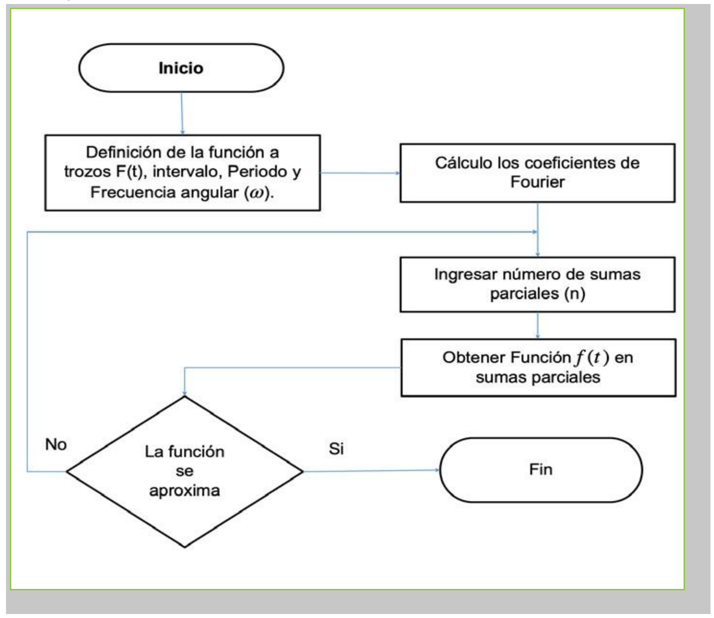
Bloque II Modificable para ingresar la función a trozos deseada.



La Fig. 3 muestra el diagrama de flujo para obtener la ***n*** suma parcial para la aproximación de la señal $f(t)$, si la expansión no se aproxima gráficamente a la señal original, se repite la operación hasta encontrar el parámetro ***n*** que nos garantice la aproximación de la señal.

Figura 3.

Diagrama de flujo para calcular la suma parcial n para la aproximación de la señal f(t).



Propuesta de secuencia para la enseñanza de las Series Trigonométricas de Fourier

La presente propuesta de secuencia para abordar el tema de series trigonométricas de Fourier, se fundamenta en el contenido del curso de Análisis de Señales y Sistemas de Comunicación de la carrera de Ingeniería en Tecnologías de la Información y Comunicaciones, así como en el curso de Matemáticas de la Maestría en Ingeniería Mecatrónica que se ofertan en el TecNM/Instituto Tecnológico de Apizaco (TecNM/ITApizaco, 2022).

Objetivo de la secuencia de aplicación: Calcular de forma analítica a través de un procedimiento manual y computacional los coeficientes de las series trigonométricas de Fourier, asimismo, obtener las gráficas para n sumas parciales.

Competencia específica: Analiza el comportamiento de la serie de Fourier como serie representativa de una función periódica.

Actividades de aprendizaje: Llevar a cabo ejercicios que incluyan series trigonométricas

Actividades:

Actividad 1. Grafique la señal utilizando alguna herramienta computacional.

Actividad 2. Obtener a_0, realice detalladamente la integración de forma manual y compruebe el resultado utilizando herramientas de sistemas de álgebra computacional.

Actividad 3. Obtener a_n, realice detalladamente la integración de forma manual y compruebe el resultado utilizando herramientas de sistemas de álgebra computacional.

Actividad 4. Obtener b_n, realice detalladamente la integración de forma manual y compruebe el resultado utilizando herramientas de sistemas de álgebra computacional.

Actividad 5. Obtenga la suma parcial S_6, muestre el procedimiento detalladamente de forma manual y utilizando herramientas de sistemas de álgebra computacional.

Actividad 6. Grafique la suma parcial S_6 y la señal original.

III. RESULTADOS

A continuación se muestran los resultados obtenidos por un estudiante a utilizar la herramienta computacional desarrollada en Python, así como el correspondiente cálculo manual realizado por el mismo estudiante de la carrera de Ingeniería en Tecnologías de la Información y Comunicaciones del TecNM/ITApizaco, en la asignatura de Análisis de Fourier y Sistemas de Comunicación, para la función $f(t)$ definida a cuatro trozos (Eq. 5), siguiendo la propuesta de secuencia de enseñanza de las series trigonométricas propuesta en la Sección II, donde se obtienen sumas parciales para n= 6 de la serie de Fourier.

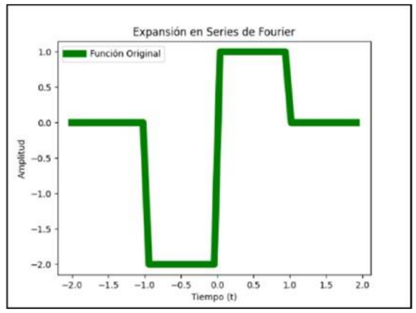
$$f(x) = \begin{cases} 0 \quad para \quad -2 \le x < -1 \\ -2 \;\; para -1 \le x < \;0 \\ 1 \;\; para \qquad 0 \le x < 1 \\ 0 \;\; para \qquad 1 \le x < 2 \end{cases}$$

(5)

En la figura 4 se muestra el gráfico de la función utilizando la herramienta computacional desarrollada Python obtenido para la Actividad 1.

Figura 4.

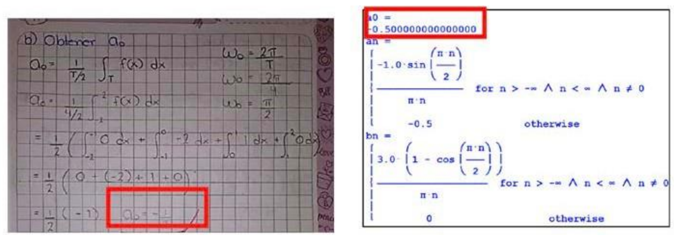
Grafica de $f(t)$



En la figura 5 se muestra el resultado de la actividad 2 para el cálculo detallado del coeficiente a_0, obtenido mediante integración directa por un estudiante (ver panel izquierdo), su comprobación al utilizar la herramienta didáctica desarrollada en Python se muestra en el panel derecho de la misma imagen.

Figura 5.

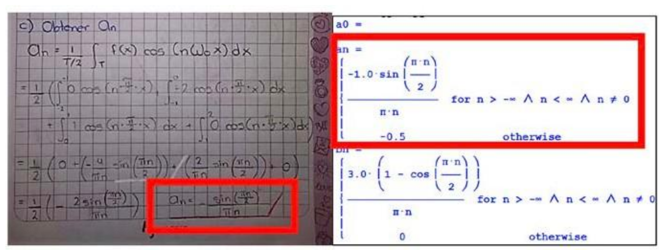
Calculo de coeficiente a_0 *y comprobación con la herramienta didáctica propuesta, el recuadro en rojo resalta el resultado final.*



En la figura 6 se muestra el resultado de la actividad 3 para la obtención detallada de a_n obtenido mediante integración directa por un estudiante (ver panel izquierdo), su comprobación al utilizar la herramienta didáctica desarrollada en Python se muestra en el panel derecho de la misma imagen.

Figura 6.

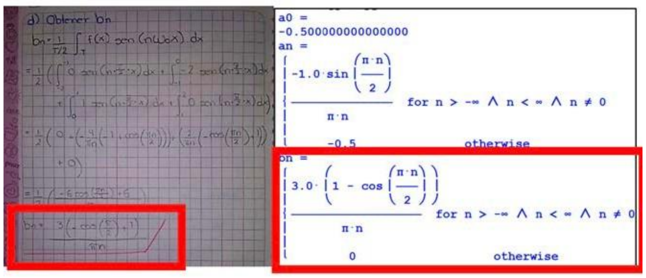
Calculo de coeficiente a_n *y comprobación con la herramienta didáctica propuesta, el recuadro en rojo resalta el resultado final.*



En la figura 7 se muestra el resultado de la actividad 4 para la obtención detallada de b_n, obtenido mediante integración directa por un estudiante (ver panel izquierdo), su comprobación al utilizar la herramienta didáctica desarrollada en Python se muestra en el panel derecho de la misma imagen.

Figura 7.

Calculo de coeficiente y comprobación con la herramienta didáctica propuesta, el recuadro en rojo resalta el resultado final.



A continuación, se muestra el resultado detallado para la actividad 6, donde se obtiene la suma parcial S_6, desarrollado de forma escrita en un procesador de textos por el mismo estudiante, sin embargo, no se llega a un resultado correcto debido a que al estudiante no le fue posible realizar la reducción de términos trigonométricos. En la figura 8 se muestra la suma parcial utilizando herramienta de sistema de álgebra computacional Python.

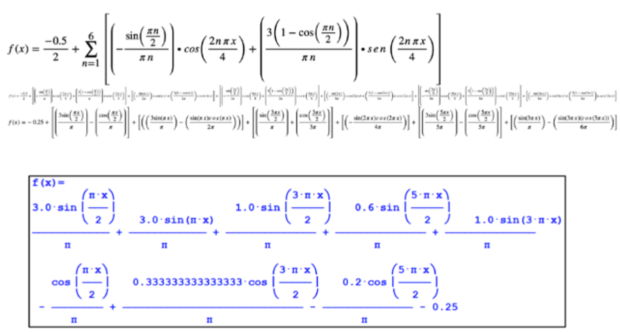


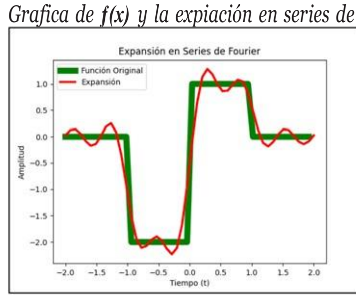
Figura 8.

*Suma para **n=6** parcial obtenida mediante la herramienta didáctica de Python.*

En figura 9 se muestra el resultado de la actividad número 7, el cual muestra la gráfica de la función original ***f(x)*** comparada con la reconstruida a partir de ***n= 6***.

Figura 9.

*Grafica de **f(x)** y la expiación en series de Fourier para **n=6**.*

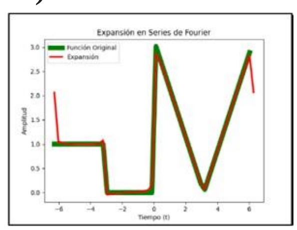


A continuación, en la Fig. 10 se muestran las gráficas de los resultados obtenidos con n=100 de tres funciones diferentes, definidas a cuatro trozos. Es preciso mencionar que, para validar resultados, la forma analítica de los coeficientes trigonométricos de Fourier fue calculado por integración directa por los autores, se observa que estos resultados y los obtenidos con el programa desarrollado son iguales.

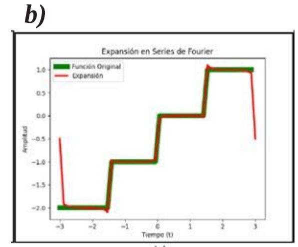
Figura 10.

Funciones de cuatro trozos obtenidos por medio de la herramienta didáctica propuesta.

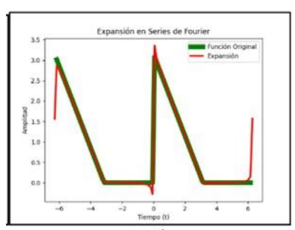
a)



b)



c)



En contraste con otras investigaciones, por ejemplo (Martínez, 2020) presenta resultados de una aplicación móvil para resolver y graficar series de Fourier siendo una manera más práctica al momento de ingresar las funcionesa trozos debido a la interfaz desarrollada de la aplicación, los resultados mostrados en esta investigación están orientados a la enseñanza y en demostrar el proceso de solución del algoritmo, no obstante, existe una limitante de solución para funciones compuestas estrictamente de uno, dos y tres trozos, por ende no es posible reconstruir señales definidas en más tres trozos. Por otro lado, los resultados de este trabajo se limitan a un número de funciones a trozos en particular, pero si a que los estudiantes tengan nociones con el manejo del software y declaración de variables, además en este caso no se ofrece los pasos mostrados para el algoritmo de solución, pero si al igual (Martínez, 2020) se logra obtener una representación simbólica para la comprobación de los coeficientes calculados. En la siguiente etapa de esta investigación, los autores están desarrollando la interfaz gráfica que permita facilitar el proceso de captura de señales, independientemente del número de trozos en que sean definidas.

IV.CONCLUSIONES

En este trabajo se ha presentado una herramienta didácti- ca desarrollada en Python que permite calcular la expresión analítica para los coeficientes de la serie trigonométrica de Fourier de señales periódicas definidas a trozos. En particular, se ha aplicado el cálculo simbólico para obtener las expansiones en series de Fourier para señales definidas a cuatro trozos. El programa desarrollado se propone como una herramienta didáctica para comprobar los resultados que se obtienen de forma escrita mediante integración directa, esto resulta de gran relevancia ya que permite centrarse en el análisis de los resultados, así como abordar ejemplos de mayor complejidad que difícilmente se resuelven de forma escrita en cursos de Análisis de señales y sistemas de comunicación.

Es importante mencionar que, para aprovechar al máximo este recurso didáctico, es necesario que los estudiantes posean nociones básicas de programación en Python, así como conocimientos sólidos de algebra elemental y cálculo diferencial, para poder ingresar en el programa propuesto las funciones a expandir. Finalmente, el programa desarrollado puede adecuarse para utilizar la forma compleja de las series de Fourier, así como para realizar el análisis espectral de señales periódicas a trozos.

REFERENCIAS BIBLIOGRÁFICAS

Arias Hernández, J. D., Jímenez López, A. F., & Porras Castro, H. (2016). Desarrollo de aplicaciones en python para el aprendizaje de física computacional. Revista Ingeniería, Investigación y Desarrollo, 16(1), 77-82.

Cruz Sanabria, H., Sánchez, M. G., Belda, S., Rivera Caicedo, J. P., & Fajardo Delgado, D. (2021). Interpolation of discontinuous optical data usingFourier series and Whittaker method: a toolboxwritten in python. Revista de Difusión Científica, Ingeniería y Tecnologías, 15(3), 140-146.

G. de Carvalho, L., C. Sediyama, G., R. Cecon, P., & R. Alves, H. M. (2005). APLICAÇÃO DA ANÁLISE HARMÔNICA POR SÉRIES DE FOURIER PARA A PREVISÃO DE PRODUTIVIDADE DA CULTURA DO CAFÉ NO ESTADO DE MINAS GERAIS. Engenharia agricola, 25(3), 732-741.

Giménez-Palomares, F., Lapuebla-Ferri, A., & Monsoriu-Serra, J. A. (2018). Trabajando interactivamente con series de Fourier y trigonométricas. Congreso Universitario de Innovación Educativa, 439-449.

Gonzalez Avid, R., & Lopez Gianmarco, V. (2021). Desarrollo de una interfaz gráfica para el cálculo de series de Fourier en Matlab. BIOTECH & ENGINEERING Untels, 1(2), 33-41. https://doi.org/DOI: https://doi.org/10.52248/eb.Vol1Iss02.15

Ivet Challenger Perez, Y. D. (2014). El lenguaje de programación Python. Ciencias Holguín, Vol. XX, num. 2, 1-13.

Luna, D. R. (29 de 10 de 2020). https://github.com/DavidReveloLuna/ProcesamientoDatos

Maggiolini, L., Klimovsky, E., & Quaglia , J. (2017). Enseñando la matemática desde la electrónica. Experiencia didáctica en la cátedra Análisis de señales y sistemas. Asociación Colombina de Facultades de Ingeniería, 12(24), 38-42.

Martínez, J. F. (2020). APLICACIÓN MÓVIL PARA DESARROLLAR Y GRAFICAR SERIES DE FOURIER.

Marturet P., G. J., Gutiérrez, E., & Carabollo, S. (2015). SERIES DE FOURIER PARA LA MODELACIÓN. Universidad, Ciencia y Tecnlogia, 19(76), 118-127.

Suslov, S. K. (2002). Some Expansions in Basic Fourier Series and Related Topics. Journal of Approximation Theory, 115(2), 289-353. https://doi.org/https://doi.org/10.1006/jath.2001.3659

TecNM/ITApizaco. (11 de 11 de 2022). Tecnológico Nacional de México Campus Apizaco. Tecnológico Nacional de México Campus Apizaco: https://www.apizaco.tecnm.mx/ing-tics/

ANEXOS

A continuación se presenta el código fuente en Python para obtener los coeficientes trigonométricos de las series de Fourier, así como la suma parcial para aproximar una señal ***f(x).***

```python
#-------- Inicia Bloque I ---------------------#

import numpy as np
import sympy as sym
from sympy import *
import scipy as sp

import matplotlib.pyplot as plt
from scipy import integrate
from sympy.abc import t, n

#--------------------------Termina-------------#
```

```python
#-------- Inicia Bloque II --------------------#

T1 = -2*pi    #\
              # > Intervalo
T2 = 2*pi     #/

T=T2-T1       #-> Periodo
w = 2*pi/T    #-> Frecuencia angular

F1=1         #\
F2=0         # \
F3=pi-t      #  > Función a Trozos
F4=t-pi      # /
             #/


Ft = sym.Piecewise((F1, ((t >= -2*pi) & (t < -pi))),
                   (F2, ((t >= -pi) & (t < 0))),
                   (F3, ((t >= 0) & (t < pi))),
                   (F4, ((t >= pi) & (t < 2*pi))))

#--------------------------Termina--------------#
```

```
#-------- Inicia Bloque III --------------------#
f_integral = Ft
a0 = (2/T)*sym.integrate(f_integral,(t,T1,T2))

f_integral = Ft*sym.cos(n*w*t)
an = (2/T)*sym.integrate(f_integral,(t,T1,T2))
an = sym.simplify(an)

f_integral = Ft*sym.sin(n*w*t)
bn = (2/T)*sym.integrate(f_integral,(t,T1,T2))
bn = sym.simplify(bn)

Entrada=0
while(1):
    print("¿Cuantas sumas parciales deseas Graficar?")
    Entrada=int(input())
    if Entrada>=1:
        print("un momento...")
        ntotal = Entrada
        break
    else:
        print("valor no valido")
#---------------------------Termina---------------#
#-------- Inicia Bloque IV -----------------------#
expansion = 0
for i in range(1,ntotal+1):

    an1 = an.subs(n,i)
    bn1 = bn.subs(n,i)

    if abs(an1) < 0.0001: an1 = 0
    if abs(bn1) < 0.0001: bn1 = 0

    expansion=expansion + an1*sym.cos(i*w*t)
    expansion = expansion + bn1*sym.sin(i*w*t)

expansion = a0/2+expansion
#---------------------------Termina---------------#
```

```
#-------- Inicia Bloque V -----------------------#
print("a0 = ")
sym.pprint(a0)
print("an = ")
sym.pprint(an)
print("bn = ")
sym.pprint(bn)

print('F(t)= ')
sym.pprint(expansion)

fexpandida = sym.lambdify(t,expansion)
f = sym.lambdify(t,Ft)

T1=int(T1)
T2=int(T2)
v_t = np.linspace(T1,T2)

f1 = fexpandida(v_t)
f2 = f(v_t)

plt.plot(v_t,f2,c='g',label = 'Función Original',linewidth=8)
plt.plot(v_t,f1,c='r', label = 'Expansión',linewidth=3)

plt.xlabel('Tiempo (t)')
plt.ylabel('Amplitud')
plt.title('Expansión en Series de Fourier')
plt.legend()
plt.show()
#---------------------------Termina---------------#
```