FUNDACIÓN UNIVERSITARIA KONRAD LORENZ
CENTROS DE INVESTIGACIONES

A continuación, encontrarán los criterios para la presentación de Trabajos de Grado o Trabajos Práctica Investigativa (TPI). El estilo de presentación debe cumplir con los lineamientos de las normas más recientes de IEEE para Ingenierías o AMS para Matemáticas.

1. IDENTIFICACIÓN GENERAL DEL TRABAJO DE GRADO ASOCIADO A LA PRÁCTICA / PRACTICA INVESTIGATIVA	
TITULO DEL TRABAJO	Desarrollo de integrales definidas mediante el uso de la identidad de Parseval
DIRECTOR TRABAJO DE GRADO	Melo Jiménez Rafael
AUTOR	Cuesta Velasquez Luz Helena
PALABRAS CLAVE	Identidad de Parseval Series de Taylor Series de Fourier Números de Bernoulli
AÑO / PERIODO	**2020-I**
MODALIDAD	Trabajo de Grado (Pregrado)

1. DESCRIPCIÓN DEL ESTUDIO (RESUMEN O ABSTRACT)

Resumen

Este documento se basa en el desarrollo de integrales definidas mediante el uso de la identidad de Parseval, debido a la complejidad de las funciones a integrar es necesario encontrar el desarrollo en series de Fourier de las funciones a integrar, para esto encontraremos el desarrollo en series de Taylor de las funciones cotangente y tangente hiperbólica, los resultados obtenidos nos llevan a cuestionar como se comporta el resultado de la integral a medida que variamos la constante r, es por esto que utilizamos el método de integración MonteCarlo en el software Python para encontrar el valor de la integral para algunos valores de r, en donde está definido, obteniendo comportamientos similares a funciones conocidas como cuadráticas o racionales.

2. INTRODUCCIÓN

(JUSTIFICACIÓN Y ENMARCAMIENTO CONCEPTUAL Y TEÓRICO DEL PROBLEMA SU EXTENSIÓN DEBE ESTAR ENTRE 1 Y 2 PAGINAS)

Es importante notar cómo podemos abordar un mismo problema de diferentes formar, y el por qué se utiliza un método y no otro. En el cálculo integral conocemos métodos de integración como sustitución, por partes, fracciones parciales, sustitución trigonométrica, entre otros. Aún así, hay situaciones en la cuales no nos es posible desarrollar ciertas integrales de forma analítica, es por esto que recurrimos a los métodos numéricos, como; Simpson, Trapecio, entre otros.

El documento que se va a desarrollar se base el artículo Solving some definite integrals using parseval’s theorem, para esto debemos abordar lo siguiente.

Las Series trigonométricas de Fourier, o simplemente series de Fourier fueron desarrolladas por el matemático francés Jean-Baptiste Joseph Fourier (21 de marzo de 1768 en Auxerre - 16 de mayo de 1830 en París).

La idea que subyace en las series de Fourier es la descomposición de una señal periódica en términos de señales periódicas básicas (senos y cosenos) cuyas frecuencias son múltiplos de la señal original, surgen en la tarea práctica de representar funciones periódicas generales. Como aplicación constituyen una herramienta muy importante en la solución de problemas en los que intervienen ecuaciones diferenciales ordinarias y parciales.

La identidad de Parseval permite expresar la integral del cuadrado de una función por los coeficientes de la serie de Fourier, es decir, .Sea f : R → R una función continua y periódica de periodo 2π, si la serie de Fourier de f converge uniformemente a f en [0, 2π] [5]. Entonces:

$$\frac{1}{\pi}\int_0^{2\pi}(f(x))^2\,dx = \frac{a_0^2}{2}+\sum_{k=1}^{\infty}(a_k^2+b_k^2).$$

El problema está en encontrar el desarrollo en series de Fourier de la función, ¿qué tan pertinente es encontrarlo? En las integrales que se van a desarrollar en este documento, es fundamental el uso de esta identidad, pues gracias a ella podemos establecer de forma analítica el resultado de dicha integral, esto debido a la complejidad de las funciones con el uso de los métodos tradicionales, para esto expresamos las funciones tangente hiperbólica y cotangente hiperbólica en series de Taylor, recordemos que el desarrollo en series de Taylor de una función f real o compleja f(x) infinitamente diferenciable en el entorno de un número real o complejo a es la siguiente serie de potencias:

$$f(a)+\frac{f'(a)}{1!}(x-a)+\frac{f''(a)}{2!}(x-a)^2+\frac{f^{(3)}(a)}{3!}(x-a)^3+\cdots+\frac{f^{(N)}(a)}{N!}(x-a)^N+\cdots,$$

lo cual se puede escribir de manera más compacta mediante, [2]:

$$\sum_{k=0}^{\infty} f^k(a)\frac{(x-a)^k}{k!}.$$

Dicha expresión se conoce como la serie de Taylor para la función f centrada en a [10].

Debido al resultado obtenido en cada integral, surge la pregunta ¿qué ocurre cuándo variamos el valor de r? Para esto utilizamos el método de MonteCarlo en el software Python, para encontrar el valor de la integral para algunos valores de r en el intervalo definido. Esto con el fin de encontrar una relación con la función a integrar y polinomios de interpolación simplificando los cálculos analíticos.

3. RESULTADOS Y DISCUSIÓN

SE DEBERÁ MOSTRAR, EN FORMA ORGANIZADA Y PRECISA LOS RESULTADOS DEL TRABAJO DE INVESTIGACIÓN, Y PRESENTAR LAS CONCLUSIONES SOBRE LOS MISMOS. SU EXTENSIÓN DEBE ESTAR ENTRE 2 Y 4 PÁGINAS.

En el desarrollo de las integrales definidas, analíticamente encontraos su resultado por medio de sumas infinitas, sin embargo, gracias al software Python, nos fue posible observar lo que ocurría con el valor de la integral a medida que variábamos el parámetro r.
Encontrando que el comportamiento de la solución se puede ver como una curva en forma de catenaria, o con comportamiento similar a una función racional.

Gráficamente podemos observar que de ser posible encontrar el polinomio de interpolación, podríamos aproximarnos al resultado de la integral si conocemos el valor de r, además dicho resultado se puede ver como un polinomio que solo depende de r, esto no solo sería novedoso, sino que además permitiría minimizar y simplificar los cálculos.

Finalmente, es interesante ver que el tipo de solución de algunas de las funciones estudiadas tiene el comportamiento que se presenta en el estudio de la economía conductual cuando se analiza el precio y la demanda, o la inversión en la bolsa de valores, es decir, se podría estudiar en esta relación y ver cómo esas funciones se relacionan con situaciones de la vida cotidiana, de lo cual surge la siguiente pregunta, ¿existe dicha relación? De ser así, ¿ayudaría a mejorar el modelamiento de dichas situaciones?

4. REFERENTES TÉORICOS Y EMPÍRICOS CONSULTADOS.

TODAS REFERENCIAS CONSULTADAS EN LA REVISIÓN SISTEMÁTICA
(AUNQUE NO APAREZCAN EN EL ARTÍCULO)

1. Howard Anton and Albert Herr, Calculus with analytic geometry, Wiley New York, 1988.

2. Tom M Apóstol, Análisis matemático, Reverte, 1996.
3. Jorge Usbaldo Fierros Bobadilla and Hermenegildo Rivera Martínez, Análisis de la convergencia de las series de Fourier, 1996.
4. James Ward Brown, Ruel Vance Churchill, et al., Complex variables and applications, Boston: McGraw-Hill Higher Education,, 2009.
5. Ramón Bruzual and Marisela Domínguez, Series de Fourier, Universidad Central de Venezuela-Año (2003).
6. Ruel V Churchill, James Ward Brown, and Lorenzo Abellanas Rapun, Variable compleja y aplicaciones, McGraw-Hill, 1986.
7. Javier Duoandikoetxea, Lecciones sobre las series y transformadas de Fourier, Recuperado de http://www. ugr. es/acanada/docencia/matematicas/analisisdefourier/Duoandikoetxeafourier. pdf (2003).
8. David José Fernández Bretón, Números de Bernoulli: Un estudio sobre s importancia, consecuencia y algunas aplicaciones en la teoría de números, PhD. thesis, 2012.
9. Sanz Gil Javier Gómez Pérez Javier Lastra Sedano Alberto Jiménez Garrido Jesús Javier Galindo Soto Félix, Tristán Vega Luis Alberto, Guía teórico-práctica de variable compleja para estudios de grado, 2015.
10. Angelo Genocchi, Calcolo differenziale e principii di calcolo integrale, vol. 1, Bocca, 1884.
11. Joan Girbau, Trigonometria esferica i hiperbolica, Materials matemátics (2006), 0001–14.
12. Genaro González, Series de Fourier, transformadas de Fourier y aplicaciones, Divulgaciones matemáticas 5 (1997), no. 1/2, 43–60
13. Erwin Kreyszig and José Hernán Pérez Castellanos, Matemáticas avanzadas para ingeniería, no. 517.38 K74 2000., Limusa, 1996.
14. A Parseval, Integration generale et complete de deux equations importantes dans la mecanique des fluides, Mem. presentes Institut Sci., I (1805), 524–545.
15. Marc-Antoine Parseval, Mémoire sur les séries et sur l'intégration compléte d'une équation aux différences partielles linéaires du second ordre, á coefficients constants, Mém. prés. par divers savants, Acad. des Sciences, Paris, (1) 1 (1806), 638–648.
16. M´ethode g´en´erale pour sommer, par le moyen des intégrales définies, la suite donnée par le théoréme de m. lagrange, au moyen de laquelle il trouve une valeur qui satisfait á une équation algébrique ou transcendante. [communicated in 1804], M´emoires présentés par divers savants 1 (1806), 567–586.
17. Manuel López Pellicer, Presentación de la obra de h. Poincaré (1854- 1912), Arbor 178 (2004), no. 704, 621–624.
18. Chii-Huei Yu, Solving some definite integrals using parseval's theorem, American Journal of Numerical Analysis 2 (2014), no. 2, 60–64.
19. Antonio Cipriano Santiago Zaragoza and María José Santiago Puertas, El análisis complejo y su historia, Editorial Liber Factory, 2014.

5. APENDICES

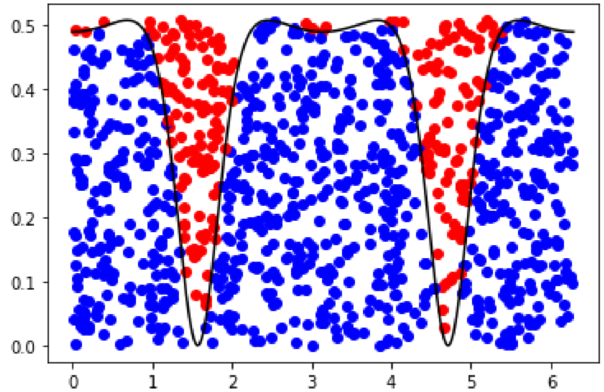
SE DEBE ANEXAR EL ARTÍCULO Y LOS DEMÁS ANEXOS QUE SE CONSIDEREN PERTINENTES

```
import math as m
```

```python
import numpy as np
import sympy as sym
import matplotlib.pyplot as plt

def maxfunction(f,a,b):
    t=np.linspace(a,b,10000)
    y=f(t)
    return max(y)
def minfunction(f,a,b):
    t=np.linspace(a,b,10000)
    y=f(t)
    return min(y)
def monteCarlo2(a,b,n,f):
    maxf=maxfunction(f,a,b)
    minf=minfunction(f,a,b)
    minf=0
    xrand=np.random.rand(n)*(b-a)+a
    yrand=np.random.rand(n)*(maxf-minf)+minf
    cont=0
    x=np.linspace(a,b,1000)
    plt.plot(x,f(x),color="black")
    for i in range(len(xrand)):
        if(f(xrand[i])>=yrand[i]):
            cont=cont+1
            plt.scatter(xrand[i],yrand[i],color="blue")
        else:
            plt.scatter(xrand[i],yrand[i],color="red")
    return (b-a)*(maxf-minf)*cont/n
```

MonteCarlo

```
a = 0
b = 2*np.pi
n = 1000
def g1(x,r=np.e/np.pi):
    return (np.sinh(r*np.cos(x))*np.cosh(r*np.cos(x))/(np.sinh(r*np.cos(x))**2+np.cos(r*np.sin(x))**2))*
*2 #int 1

monteCarlo2(a,b,n,g1)
2.512256072859972
```



Método del trapecio

```
def f(x,r=np.e/np.pi):
    return (np.sinh(r*np.cos(x))*np.cosh(r*np.cos(x))/(np.sinh(r*np.cos(x))**2+np.cos(r*np.sin(x))**2))*
*2 #int 1

a = 0
b = 2*np.pi
```

```
n = 1000
h = (b-a)/n
x = a

suma = f(x)

for i in range(n):
    x = x+h
    suma = suma + 2*f(x)

suma = suma + f(b)
area = h*(suma/2)


print('n : ', n )
print('Integral: ', area)
```

```
n :  1000
Integral:  2.4160229560282197
```

Regla de Simpson

```
def g(x,r=np.e/np.pi):
    return (np.sinh(r*np.cos(x))*np.cosh(r*np.cos(x))/(np.sinh(r*np.cos(x))**2+np.cos(r*np.sin(x))**2))*
*2 #int 1

a = 0
b = 2*np.pi
n = 1000
h = (b-a)/n
suma = 0
```

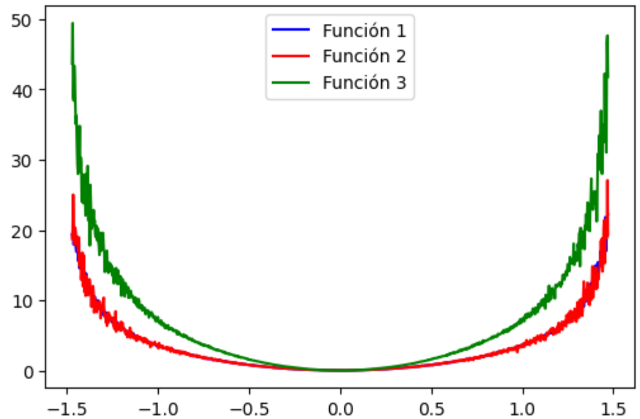
```python
for i in range(n):
    b = a + h
    m = (a+b)/2
    area = ((b-a)/6)*(g(a) + 4*g(m) + g(b))
    suma = suma + area
    a = b
print('Integral: ', suma)
```

```
Integral:  2.481295337375054
```

Código 2

```python
import numpy as np
import matplotlib.pyplot as plt



def maxfunction(f,a,b,r=np.e/np.pi):
    t=np.linspace(a,b,10000)
    y=f(t,r)
    return max(y)
def monteCarlo(a,b,n,f,r=np.e/np.pi):
    maxf=maxfunction(f,a,b,r)
    minf=0
    xrand=np.random.rand(n)*(b-a)+a
    yrand=np.random.rand(n)*(maxf-minf)+minf
    cont=0
    for i in range(len(xrand)):
        if(f(xrand[i],r)>=yrand[i]):
            cont=cont+1
    return (b-a)*(maxf-minf)*cont/n
```

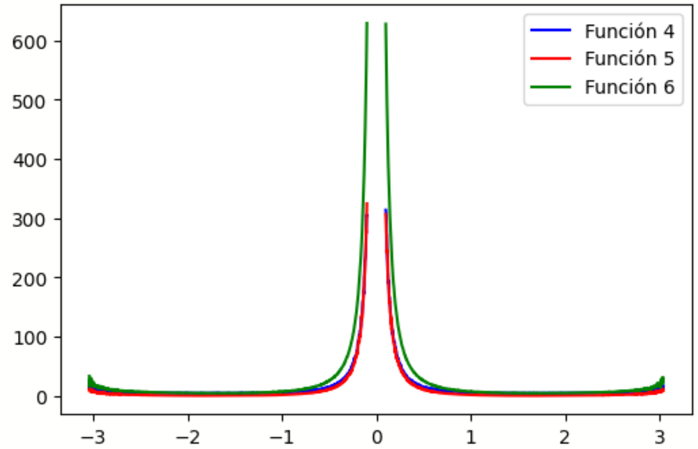
```
def f1(x,r=np.e/np.pi):
    return (np.sinh(r*np.cos(x))*np.cosh(r*np.cos(x))/(np.sinh(r*np.cos(x))**2+np.cos(r*np.sin(x))**2))*
*2 #INT 1

arreglo1=np.array([])
arreglo2=np.array([])
arreglo3=np.array([])
r1=np.linspace(-np.pi/2 + 0.1,np.pi/2 -0.1 ,1000)
a = 0
b = 2*np.pi
n = 1000
for i in r1:
    arreglo1=np.append(arreglo1,monteCarlo(a,b,n,f1,i))
    arreglo2=np.append(arreglo2,monteCarlo(a,b,n,f2,i))
    arreglo3=np.append(arreglo3,monteCarlo(a,b,n,f3,i))

plt.figure(dpi=100)
plt.plot(r1,arreglo1,color="blue",label="Función 1")
plt.plot(r1,arreglo2,color="red",label="Función 2")
plt.plot(r1,arreglo3,color="green",label="Función 3")
plt.legend()


plt.show()
```



Código 3

```
import numpy as np
import matplotlib.pyplot as plt



def maxfunction(f,a,b,r=np.e/np.pi):
    t=np.linspace(a,b,10000)
    y=f(t,r)
    return max(y)
def monteCarlo(a,b,n,f,r=np.e/np.pi):
    maxf=maxfunction(f,a,b,r)
    minf=0
    xrand=np.random.rand(n)*(b-a)+a
    yrand=np.random.rand(n)*(maxf-minf)+minf
    cont=0
    for i in range(len(xrand)):
        if(f(xrand[i],r)>=yrand[i]):
```

```
            cont=cont+1
    return (b-a)*(maxf-minf)*cont/n
arreglo4=np.array([])
arreglo5=np.array([])
arreglo6=np.array([])
arreglo7=np.array([])
arreglo8=np.array([])
arreglo9=np.array([])
r1=np.linspace(-np.pi+0.1,0-0.1,1000)
r2=np.linspace(0+0.1,np.pi-0.1,1000)
a = 0
b = 2*np.pi
n = 1000
for i in r1:
    arreglo4=np.append(arreglo4,monteCarlo(a,b,n,f4,i))
    arreglo5=np.append(arreglo5,monteCarlo(a,b,n,f5,i))
    arreglo6=np.append(arreglo6,monteCarlo(a,b,n,f6,i))

for i in r2:
    arreglo7=np.append(arreglo7,monteCarlo(a,b,n,f4,i))
    arreglo8=np.append(arreglo8,monteCarlo(a,b,n,f5,i))
    arreglo9=np.append(arreglo9,monteCarlo(a,b,n,f6,i))
plt.figure(dpi=100)
plt.plot(r1,arreglo4,color="blue",label="Función 4")
plt.plot(r1,arreglo5,color="red",label="Función 5")
plt.plot(r1,arreglo6,color="green",label="Función 6")
plt.plot(r2,arreglo7,color="blue")
plt.plot(r2,arreglo8,color="red")
plt.plot(r2,arreglo9,color="green")
plt.legend()

plt.show()
```


Función 4
Función 5
Función 6
600
500
400
300
200
100
0
−3
−2
−1
0
1
2
3