



# UNIVERSIDAD DE SONORA

División de Ciencias Exactas y Naturales  
Departamento de Física

---

## Reconstruyendo la señal

---

Autor: Natalia Hinostroza Moya

Profesor: Carlos Lizárraga Celaya

18 de abril de 2017

# Resumen

Previamente, en la actividad 6, descompusimos en intervalos de tiempo las mareas de los sitios escogidos por cada uno de los alumnos del curso. Con apoyo de la Transformada de Fourier Discreta (DFT), extraimos los modos naturales de la marea.

Obtenido lo anterior, en esta actividad se pide reconstruir la señal de los valores obtenidos por la DFT, así como encontrar el error relativo de aproximar la señal de la marea.

## 1. Introducción

Con los datos de la actividad pasada utilizaremos la serie de Fourier la cual se utiliza para el estudio de movimientos armónicos. En esta actividad compararemos la serie de Fourier que se obtiene directamente y la que nosotros crearemos con los datos recopilados y estudiados anteriormente.

## 2. Serie de Fourier

La serie de Fourier es una serie infinita que converge puntualmente a una función periódica y continua por partes. Esta serie forma parte de una de las herramientas matemáticas básicas de análisis de Fourier que se emplea para el análisis de funciones periódicas a través de la descomposición de la función en una suma infinita de funciones sinusoidales simples.

Su nombre es en honor al matemático francés Jean-Baptiste Joseph Fourier, que fue quien desarrolló la teoría mientras estudiaba la ecuación de calor. Fue el primero en estudiar las series armónicas.

La serie de Fourier, además de ser una herramienta sumamente útil en la teoría de la matemática abstracta, es muy utilizada en muchas ramas de la ingeniería. Algunas de sus áreas de aplicación incluyen análisis vibratorios, acústica, óptica, procesamiento de imágenes y señales, y comprensión de datos.

## 3. Mazatlán

```
def f(t):  
    return -900 + (103)*np.sin(np.pi*t/12.9) + (177.3)*np.sin(np.pi*t/12.1) + (79.8)*np.sin(np.pi*t/12.1)  
    + (72.5)*np.sin(np.pi*t/6.3) + (64.6)*np.sin(np.pi*t/6.2) + (342.6)*np.sin(np.pi*t/6.2) + (102.5)  
    + (0.15)*np.sin(np.pi*t/13.16) + (0.42)*np.sin(np.pi*t/12.2) + (0.08)*np.sin(np.pi*t/6.3)
```

```

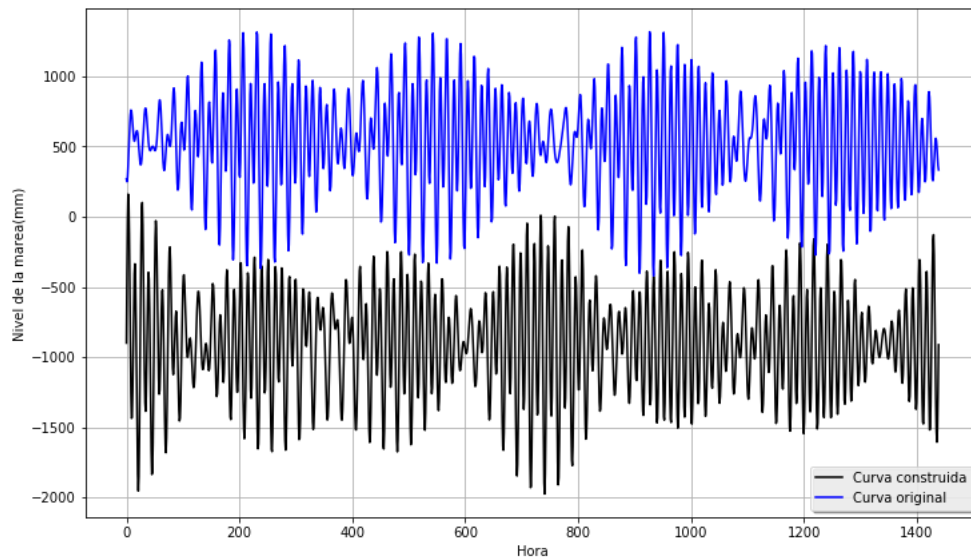
#interpolat la funcion
df[u'WaterLevel'] = df[u'WaterLevel'].astype(float).interpolate(method='spline', order=2)

fig = plt.gcf()
fig.set_size_inches(12, 7)

plt.plot(df[u'Hora'], f(df[u'Hora']), c='black', label='Curva construida')
plt.plot(df[u'Hora'], df[u'WaterLevel'], c='blue', label='Curva original')

plt.ylabel('Nivel de la marea(mm)')
plt.xlabel('Hora')
plt.legend(fancybox=True, shadow=True)
plt.grid(True)

```



## 4. San Francisco

```

def f(t):
    return 3.5 + (0.43)*np.sin(np.pi*t/12.9) + (1)*np.sin(np.pi*t/12) + (0.32)*np.sin(np.pi*t/6.3)

#interpolat la funcion
df[u'WaterLevel'] = df[u'WaterLevel'].astype(float).interpolate(method='spline', order=2)

```

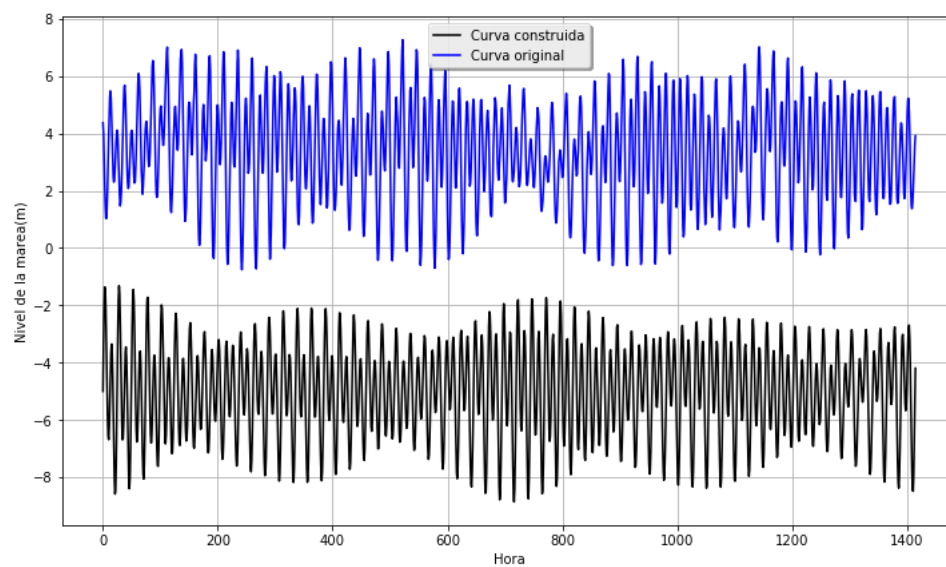
```

fig = plt.gcf()
fig.set_size_inches(12, 7)

plt.plot(df[u'Hora'], f(df[u'Hora']), c='black', label='Curva construida')
plt.plot(df[u'Hora']+175, df[u'WaterLevel'], c='blue', label='Curva original')

plt.ylabel('Nivel de la marea(m)')
plt.xlabel('Hora')
plt.legend(fancybox=True, shadow=True)
plt.grid(True)

```



## 5. Conclusión

La curva original y la curva construida con los datos que había acomodado no son iguales, pero se puede notar la similitud entre ellas. Otra cosa que puede notarse a simple vista es que están desfasadas, no logré acomodarlas para que quedarán en fase.