

UNIVERSIDAD DE SONORA



"El saber de mis hijos
hará mi grandeza"

FÍSICA COMPUTACIONAL I

Actividad 4

Predicción de la marea con la transformada rápida de Fourier

Gómez García, Manuel Ignacio 10/Diciembre/2018

1. Introducción

Algo que se busca hoy en día es generar una manera de predecir las mareas presentes en una zona con una gran precisión, sin embargo, esto resulta bastante complicado debido a la cantidad de factores que influyen en ello, pero existe otra manera de obtener una aproximación decente de ello, lo cual se basa en el análisis armónico de las ondas que mueven la marea.

Como sabemos, la onda que es descrita por el mar no tiene una ecuación sencilla, sino que más bien esta compuesta por diversas ondas armónicas que son las constituyentes de la onda final (la marea), pero ¿cómo es que podemos conocer dichos constituyentes?, ¿qué genera estos movimientos armónicos? pues si bien los factores a analizar serán las posiciones de dos astros bien conocidos, el Sol y la Luna, ya que son factores muy importantes en las mareas presentes sobre la Tierra.

Según la alineación de la Tierra con el Sol y la Luna se crean diversas movimientos armónicos que pueden ser diurnos (aproximadamente cada 24 horas), semidiurnos (aproximadamente cada 12 horas) u de otro tipo. De hecho, existen ciertos nombres para los armónicos, de modo que sea más fácil identificar cuáles influyen en cierta zona (figura 1).

Símbolo	Constituyente	Período, T (hora)	Frecuencia, ω (rad/hora)
M2	Semidiurno lunar	12.42	0.5059
S2	Semidiurno solar	12	0.5236
K1	Diurno luna-sol	23.93	0.2626
O1	Diurno lunar declinación	25.82	0.2433
N2	Semidiurno lunar elíptico largo	12.66	0.4963
M4	Primer sobremarea de M2	6.21	1.0118
S4	Primer sobremarea de S2	6	1.0472
MS4	Compuesto luna-sol	6.1	1.03
M6	Segunda sobremarea de M2	4.14	1.5177

Figura 1: Algunos de los constituyentes existentes [3].

Ahora sólo resta encontrar los constituyentes en nuestra zona de interés en base a los niveles registrados de la marea y para ello haremos uso de la Transformada Rápida de Fourier (FFT, por sus siglas en inglés). Esta función tiene como único requisito que la serie de datos introducidos contenga una cantidad de 2^n puntos y las ventajas que brinda son bastantes puesto que permite calcular la Transformada de Fourier Discreta (DFT, por sus siglas en inglés)

y su inversa, reduce el tiempo de cálculo de n^2 pasos a $n \cdot \log_2(n)$, es capaz de resolver ecuaciones de derivadas parciales y algoritmos de multiplicación rápida de grandes enteros, aunque una limitación de esta transformada es que la resolución espectral $\Delta\omega$ es inversamente proporcional al tiempo total Δt de recogida de datos en la serie temporal.

Haciendo uso de la FFT obtendremos los constituyentes de nuestra zona de estudios, ya que contamos con nuestra serie temporal conteniendo información del nivel del mar a lo largo de cierto tiempo determinado.

2. Desarrollo

Para comenzar, es necesario obtener una serie de datos de alguna estación a orillas del mar. En esta ocasión se utilizará una serie temporal proporcionada por el Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE) [7], con la cual estaremos trabajando (figura 2).

The image shows a web application titled "PREDICCIÓN DE MAREAS" with the subtitle "Series Temporales". At the top, there is a navigation bar with four tabs: "Casos", "Meteorológicas", "Datos", and "Predicciones". The "Predicciones" tab is currently selected and highlighted in dark blue. Below the navigation bar, the form contains several input fields and dropdown menus for configuring the data series:

- Localidad:** A dropdown menu showing "Guaymas, Sonora".
- Fecha Inicial:** A date input field showing "01/01/2016".
- Fecha Final:** A date input field showing "06/30/2016".
- Zona de Tiempo:** A dropdown menu showing "(GMT) Greenwich Mean Time: Dublin, Edinburgh, Lisbon, London".
- Opción:** A dropdown menu showing "Serie Temporal".
- DT:** A dropdown menu showing "60 minutos".
- Nivel de Referencia:** A dropdown menu showing "NMM: Nivel Medio del Mar".

At the bottom of the form, there are two buttons: "Predecir" (highlighted in dark blue) and "Limpiar".

Figura 2: Ajustes utilizados para la serie de datos.

Haciendo uso de Jupyter Notebook, crearemos un cuadernillo de trabajo Python 3 e importamos las bibliotecas necesarias para el código que estamos por elaborar (figura 3).

```
In [1]: 1 import pandas as pd
        2 import numpy as np
        3
        4 import matplotlib.pyplot as plt
        5 %matplotlib inline
        6
        7 import plotly.offline as ploff
        8 import plotly.graph_objs as go
        9 ploff.init_notebook_mode(connected=True)
```

Figura 3: Importación de bibliotecas necesarias.

Ahora procesaremos los datos para tener un mejor manejo de ellos, para esto será necesario agregar, renombrar, ordenar, eliminar y modificar algunas de las columnas con las que ya contamos (figura 4).

Una vez procesados los datos, haremos una gráfica en Plotly para visualizar los valores registrados del nivel del mar, de esta manera podremos darnos una idea del movimiento (figura 5).

Tras correr el código mostrado anteriormente, una gráfica será generada justo debajo del bloque de código (figura 6).

Crearemos 3 marcos de datos distintos para analizar en distintas magnitudes la marea y buscar constituyentes con frecuencias menores o mayores (figura 7).

Una vez generados los marcos de datos para cada lapso por analizar, procedemos con el código y la gráfica de cada uno de ellos.

2.1. Marea de Guaymas durante 2 días

- Código (figura 8)
- Gráfica (figura 9)

```

In [2]: 1 #Lectura de los datos.
2 Guaymas_6M = pd.read_table('./PredMar-Guaymas-2018-1.txt', sep='\s+', skiprows=7,
3                               header=None, dtype=object)
4
5 #Agregar/modificar algunas columnas.
6 Guaymas_6M['Fecha'] = Guaymas_6M[0] + '-' + Guaymas_6M[1] + '-' + Guaymas_6M[2] + ' ' +
7
8 #Guaymas_6M.columns = ['a', 'b', 'c', 'd', 'e', 'f', 'Pred(m)', 'Fecha']
9 Guaymas_6M.rename(columns={6:'PredMar(m)'}, inplace=True)
10
11 #Eliminar columnas que no usaremos más.
12 Guaymas_6M.drop([0, 1, 2, 3, 4, 5], axis=1, inplace=True)
13
14 #Reordenar columnas.
15 Guaymas_6M = Guaymas_6M[['Fecha', 'PredMar(m)']]
16
17 #Cambiamos el tipo de la columna PredMar(m).
18 Guaymas_6M['PredMar(m)'] = Guaymas_6M['PredMar(m)'].astype(float)
19
20 Guaymas_6M

```

```

+ Guaymas_6M[3] + ':' + Guaymas_6M[4] + ':' + Guaymas_6M[5]

```

Figura 4: Procesamiento completo aplicado a los datos empleados.

2.2. Marea de Guaymas durante 1 semana

- Código (figura 10)
- Gráfica (figura 11)

```
In [3]: 1 Datos = go.Scatter(name='Predicción (m)',
2           x=Guaymas_6M['Fecha'],
3           y=Guaymas_6M['PredMar(m)'])
4
5 Apariencia = go.Layout(xaxis=dict(title='Fecha'),
6           yaxis=dict(title='Nivel de la marea (m)'),
7           title='Predicción de la marea (Guaymas 2018-1)')
8
9 Datos = [Datos]
10
11 graf = go.Figure(data=Datos, layout=Apariencia)
12
13 ploff.iplot(graf, 'Prediccion de la Marea')
```

Figura 5: Configuración empleada para generar el gráfico en Plotly.

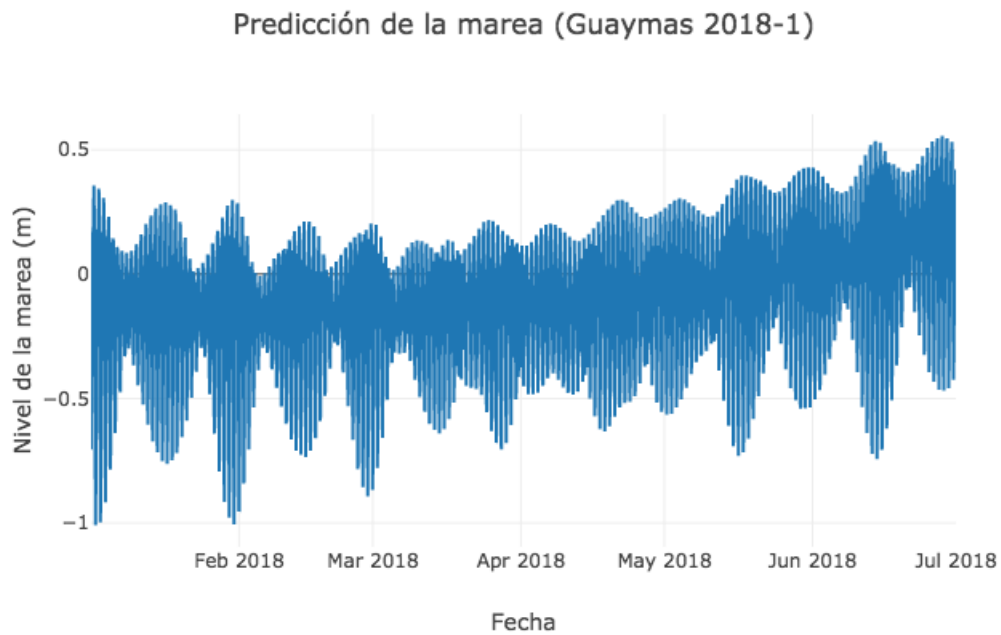


Figura 6: Nivel de la marea para Guaymas.

```
In [4]: 1 #Crearemos diversos DF para distintos lapsos de tiempo.
2 Guaymas_2Dias = Guaymas_6M[Guaymas_6M.Fecha < '2018-01-03 00:00:00']
3
4 Guaymas_1Semana = Guaymas_6M[Guaymas_6M.Fecha < '2018-01-08 00:00:00']
5
6 Guaymas_3Meses = Guaymas_6M[Guaymas_6M.Fecha < '2018-04-01 00:00:00']
```

Figura 7: Analizaremos 3 lapsos de tiempo distintos para obtener una vista más amplia del fenómeno.

```
In [5]: 1 #FFT para Guaymas_2Dias
2
3 Fs = 150.0; # sampling rate
4 Ts = 1.0/Fs; # sampling interval
5 t = Guaymas_2Dias['Fecha'] # time vector
6
7 ff = 5; # frequency of the signal
8 y = Guaymas_2Dias['PredMar(m)']
9
10 n = len(y) # length of the signal
11 k = np.arange(n)
12 T = n/Fs
13 frq = k/T # two sides frequency range
14 frq = frq[range(n//2)] # one side frequency range
15
16 Y = np.fft.fft(y)/n # fft computing and normalization
17 Y = Y[range(n//2)]
18
19 fig, ax = plt.subplots(2, 1)
20 ax[0].plot(t,y)
21 ax[0].set_xlabel('Time')
22 ax[0].set_ylabel('Amplitude')
23 ax[1].plot(frq,abs(Y),'r') # plotting the spectrum
24 ax[1].set_xlabel('Freq (Hz)')
25 ax[1].set_ylabel('|Y(freq)|')
26
27 Plot = ploff.plot_mpl(fig, filename='PredGuaymas2D.html') #Abre en navegador.
28 Plot = ploff.iplot_mpl(fig) #Abre en jupyter notebook.
```

Figura 8: Código para el análisis durante 2 días.

2.3. Marea de Guaymas durante 3 meses

- Código (figura 12)
- Gráfica (figura 13)

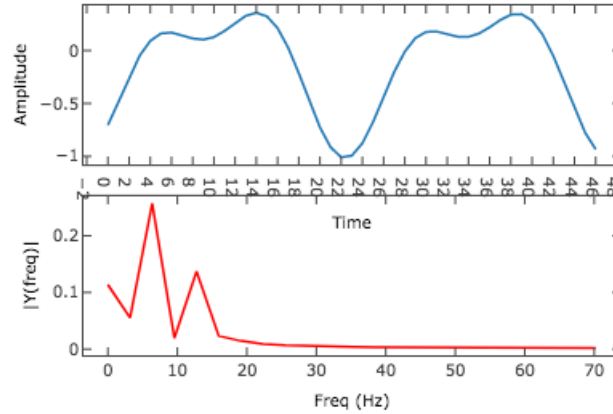


Figura 9: Gráfica para el análisis durante 2 días.

3. Conclusión

Tras ver las gráficas generadas podemos notar como es que se mantiene constante la aparición de ciertos picos que se elevan de manera drástica, los cuales tienen en común la frecuencia en cada uno de los casos que visualizamos, sin embargo, parece ser que la amplitud mostrada en los gráficos disminuye conforme aumentamos el rango de tiempo que abarca cada muestra.

Los constituyentes más notorios en la zona podríamos decir que son un semidiurno y una primer sobremarea, de las cuales la sobremarea presenta una mayor amplitud.

Referencias

- [1] (2018, Agosto 8). HARMONIC ANALYSIS. 10 de diciembre del 2018, de National Oceanic and Atmospheric Administration. Recuperado de <https://tidesandcurrents.noaa.gov/harmonic.html>
- [2] (2018, Agosto 13). HARMONIC ANALYSIS AND PREDICTION OF TIDES. 12 de diciembre del 2018, de Stony Brook Mathematics Department and Institute for Mathematical Sciences. Recuperado de


```

In [6]: 1 #FFT para Guaymas_1Semana
        2
        3 Fs = 150.0; # sampling rate
        4 Ts = 1.0/Fs; # sampling interval
        5 t = Guaymas_1Semana['Fecha'] # time vector
        6
        7 ff = 5; # frequency of the signal
        8 y = Guaymas_1Semana['PredMar(m)']
        9
        10 n = len(y) # length of the signal
        11 k = np.arange(n)
        12 T = n/Fs
        13 frq = k/T # two sides frequency range
        14 frq = frq[range(n//2)] # one side frequency range
        15
        16 Y = np.fft.fft(y)/n # fft computing and normalization
        17 Y = Y[range(n//2)]
        18
        19 fig, ax = plt.subplots(2, 1)
        20 ax[0].plot(t,y)
        21 ax[0].set_xlabel('Time')
        22 ax[0].set_ylabel('Amplitud')
        23 ax[1].plot(frq,abs(Y),'r') # plotting the spectrum
        24 ax[1].set_xlabel('Freq (Hz)')
        25 ax[1].set_ylabel('|Y(freq)|')
        26
        27 Plot = ploff.plot_mpl(fig, filename='PredGuaymas1S.html')
        28 Plot = ploff.iplot_mpl(fig)

```

Figura 10: Código para el análisis durante 1 semana.

http://www.math.stonybrook.edu/~tony/tides/harmonic.html?fbclid=IwAR3yu7gX_cD8MQ52NVjTt-WdWSiHnQbla390IeFimeRvxufgonarBFdcCQ4

- [3] (2018). THE USE OF THE HARMONIC ANALYSIS METHOD FOR ANALYSING TIDAL LEVELS. 12 de diciembre del 2018, de Ivory Research. Recuperado de <https://www.ivoryresearch.com/writers/james-barton-ivory-research-writer/>
- [4] (2018) FAST FOURIER TRANSFORM IN MATPLOTLIB. 10 de diciembre del 2018, de Plotly. Recuperado de <https://plot.ly/matplotlib/fft/#basic-fft-plot-with-matplotlib>
- [5] (2018, Octubre 21). TRANSFORMADA RÁPIDA DE FOURIER. 12 de diciembre del 2018, de Wikipedia. Recuperado de

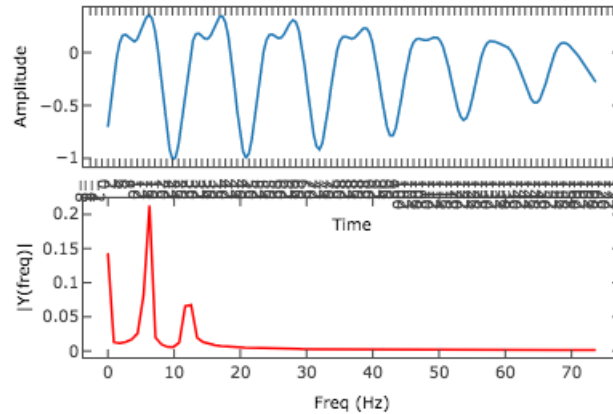


Figura 11: Gráfica para el análisis durante 1 semana.

https://es.wikipedia.org/wiki/Transformada_r%C3%A1pida_de_Fourier

- [6] (2016). TRANSFORMADA RÁPIDA DE FOURIER (I). 12 de diciembre del 2018, de Franco García, Ángel. Recuperado de http://www.sc.ehu.es/sbweb/fisica3/datos/fourier/fourier_1.html
- [7] (2018). PREDICCIÓN DE MAREAS - SERIES TEMPORALES. 10 de diciembre del 2018, de Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE). Recuperado de <http://redmar.cicese.mx/meteoro/graph/prediccion.php>

```

In [7]: 1 #FFT para Guaymas_3Meses
2
3 Fs = 150.0; # sampling rate
4 Ts = 1.0/Fs; # sampling interval
5 t = Guaymas_3Meses['Fecha'] # time vector
6
7 ff = 5; # frequency of the signal
8 y = Guaymas_3Meses['PredMar(m)']
9
10 n = len(y) # length of the signal
11 k = np.arange(n)
12 T = n/Fs
13 frq = k/T # two sides frequency range
14 frq = frq[range(n//2)] # one side frequency range
15
16 Y = np.fft.fft(y)/n # fft computing and normalization
17 Y = Y[range(n//2)]
18
19 fig, ax = plt.subplots(2, 1)
20 ax[0].plot(t,y)
21 ax[0].set_xlabel('Time')
22 ax[0].set_ylabel('Amplitude')
23 ax[1].plot(frq,abs(Y),'r') # plotting the spectrum
24 ax[1].set_xlabel('Freq (Hz)')
25 ax[1].set_ylabel('|Y(freq)|')
26
27 Plot = ploff.plot_mpl(fig, filename='PredGuaymas3M.html')
28 Plot = ploff.iplot_mpl(fig)

```

Figura 12: Código para el análisis durante 3 meses.

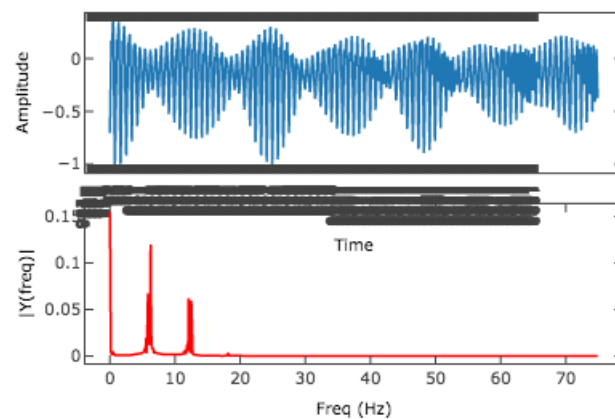


Figura 13: Gráfica para el análisis durante 3 meses.