



UNIVERSIDAD DE SONORA

DEPARTAMENTO DE FÍSICA

ACTIVIDAD 7

Alumno:

Luis Alfonso Torres Flores

Profesor

Carlos Lizárraga Celaya

18 de Abril de 2017

Índice

Índice	1
1. Breve resumen	2
2. Introducción	2
3. Procedimiento	2
3.1. Baltimore	3
3.1.1. Código de amplitud	3
3.1.2. Código de frecuencia	3
3.1.3. Código de fase	3
3.1.4. Función	4
3.1.5. Gráfica	4
3.2. Topolobampo	5
3.2.1. Código de amplitud	5
3.2.2. Código de frecuencia	5
3.2.3. Código de fase	5
3.2.4. Función	5
3.2.5. Gráfica	6
4. Error relativo	6
4.1. Código	6

1. Breve resumen

En la anterior actividad se utilizó la transformada discreta de Fourier, quitándole a las olas sus fases y obteniendo como resultado la gráfica respectiva a ese trabajo, ahora iremos de atrás para adelante y buscaremos obtener la gráfica inicial que se obtiene haciendo uso de los datos.

2. Introducción

Hasta el momento hemos trabajado con las mareas de Baltimore, Maryland y Topolobampo, Sinaloa. Usamos datos de los meses de enero, febrero y marzo del año 2016. Debido a la naturaleza del trabajo, este se realizó encima del código de la actividad 6, pues esta actividad es una regresión a la gráfica original. Nos encontramos con un problema aquí, si bien podemos utilizar una función donde simplemente podríamos ingresar una amplitud y el coseno del producto de nuestro periodo, nuestra frecuencia y un valor t , pero hay algo que se llegó a omitir, la fase. En la actividad 6, gracias al código empleado ahí, quitamos la fase, borramos ese factor que nos decía en que determinados tiempos volvería a aparecer dicha marea, por lo que el principal problema es encontrar esa fase. Se usaron datos de la actividad 6, como viene siendo la hora en la que ocurren los principales puntos de las mareas, esos picos representativos de nuestra grafica de la transformada de Fourier que nos ayudaran a realizar la regresión. Si nos damos cuenta, esta actividad puede cerrar un circulo con la actividad anterior, puesto que llegamos a un punto con la transformada de Fourier y ahora nos estamos regresando al inicio con la mayor precisión posible.

3. Procedimiento

Después de la transformada de Fourier nosotros ya habíamos pedido que se nos dieran las alturas superiores a un valor y con esto la hora o número de dato que le corresponde. Nosotros necesitamos las amplitudes representativas, sus frecuencias y su fase correspondiente. Ya tenemos los nodos identificados, solo nos quedaríamos con las horas en las que ocurren, con lo que usaríamos el código que se menciona en sus correspondientes apartados para cada uno de los lugares. Si bien notamos solo hacemos referencia a dichos datos de las horas, toando 12 datos en caso de Baltimore ignorando al cero y 4 en el caso de Topolobampo, puesto que su grafica con la transformada de Fourier se puede notar que no tiene muchas alturas representativas.

La frecuencia fue lo que se buscó de forma consecutiva, como bien notaremos abajo se sumó un valor 1092 a todas ellas, esto se debe a que solo estamos buscando datos del lado derecho de la gráfica, valores positivos del eje X , por lo que al sumarle dicho valor junto con el dato obtendríamos la frecuencia que observamos en la gráfica.

Por último, tenemos la fase, el mayor problema que se presentó en esta actividad. La fase se buscó en forma de un ángulo para su posterior coloca miento en la función empleada en este trabajo. Dicha función tiene un formato de

$(Amplitud) * \cos(Periodo * Frecuencia * fase)$, que como bien utilizamos distintas amplitudes representativas, tenemos una sumatoria de cosenos con sus respectivas amplitudes, frecuencias y fases. Las cuales se graficaron y podremos observar. Se colocó en color azul la gráfica que se obtiene directamente con los datos descargados de las paginas oficiales de sus correspondientes lugares y en rojo la gráfica obtenida con este procedimiento.

3.1. Baltimore

3.1.1. Código de amplitud

```
A0m = np.absolute(yf[0]/N)
O1m = np.absolute(yf[4]/N)
S1m = np.absolute(yf[6]/N)
N2m = np.absolute(yf[7]/N)
M2m = np.absolute(yf[11]/N)
Q2m = np.absolute(yf[12]/N)
F2m = np.absolute(yf[14]/N)
S2m = np.absolute(yf[18]/N)
D2m = np.absolute(yf[24]/N)
G2m = np.absolute(yf[29]/N)
H2m = np.absolute(yf[30]/N)
J2m = np.absolute(yf[44]/N)
K2m = np.absolute(yf[176]/N)
```

3.1.2. Código de frecuencia

```
TS1 = xf[int(1092 + 4),]
TN2 = xf[int(1092 + 6),]
TM2 = xf[int(1092 + 7),]
TS2 = xf[int(1092 + 11),]
TQ2 = xf[int(1092 + 12)]
TR2 = xf[int(1092 + 14)]
TT2 = xf[int(1092 + 18)]
TY2 = xf[int(1092 + 24)]
TU2 = xf[int(1092 + 29)]
TI2 = xf[int(1092 + 30)]
TO2 = xf[int(1092 + 44)]
TP2 = xf[int(1092 + 176)]
```

3.1.3. Código de fase

```
TS1s = np.angle(yf[int(4),])
TN2s = np.angle(yf[int(6),])
```

```

TM2s = np.angle(yf[int(7),])
TS2s = np.angle(yf[int(11),])
TQ2s = np.angle(yf[int(12)])
TR2s = np.angle(yf[int(14)])
TT2s = np.angle(yf[int(18)])
TY2s = np.angle(yf[int(24)])
TU2s = np.angle(yf[int(29)])
TI2s = np.angle(yf[int(30)])
TO2s = np.angle(yf[int(44)])
TP2s = np.angle(yf[int(176)])

```

3.1.4. Función

```

y = df['Altura']
w = 2,0 * np.pi

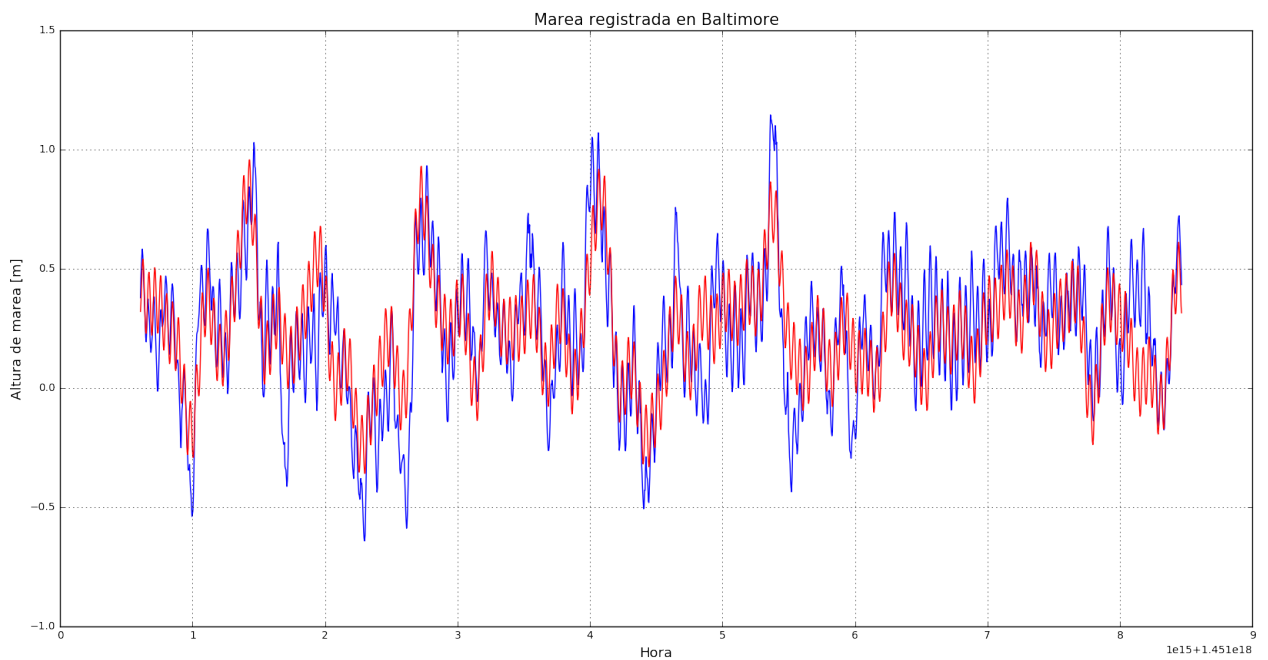
```

```

def f(t) :
return A0m + 2*(O1m*np.cos(w*TS1*t+TS1s) + S1m*np.cos(w*TN2*t+TN2s) +
N2m*np.cos(w*TM2*t+TM2s) + M2m*np.cos(w*TS2*t+TS2s) + Q2m*np.cos(w*
TQ2*t+TQ2s) + F2m*np.cos(w*TR2*t+TR2s) + S2m*np.cos(w*TT2*t+TT2s) +
D2m*np.cos(w*TY2*t+TY2s) + G2m*np.cos(w*TU2*t+TU2s) + H2m*np.cos(w*
TI2*t+TI2s) + J2m*np.cos(w*TO2*t+TO2s) + K2m*np.cos(w*TP2*t+TP2s))

```

3.1.5. Gráfica



3.2. Topolobampo

3.2.1. Código de amplitud

```
A0m = np.absolute(yf[0]/N)
O1m = np.absolute(yf[85]/N)
S1m = np.absolute(yf[91]/N)
N2m = np.absolute(yf[176]/N)
M2m = np.absolute(yf[182]/N)
```

3.2.2. Código de frecuencia

```
fO1 = yf[int(1092)]
TS1 = xf[int(1092 + 85),]
TN2 = xf[int(1092 + 91),]
TM2 = xf[int(1092 + 176),]
TS2 = xf[int(1092 + 182),]
```

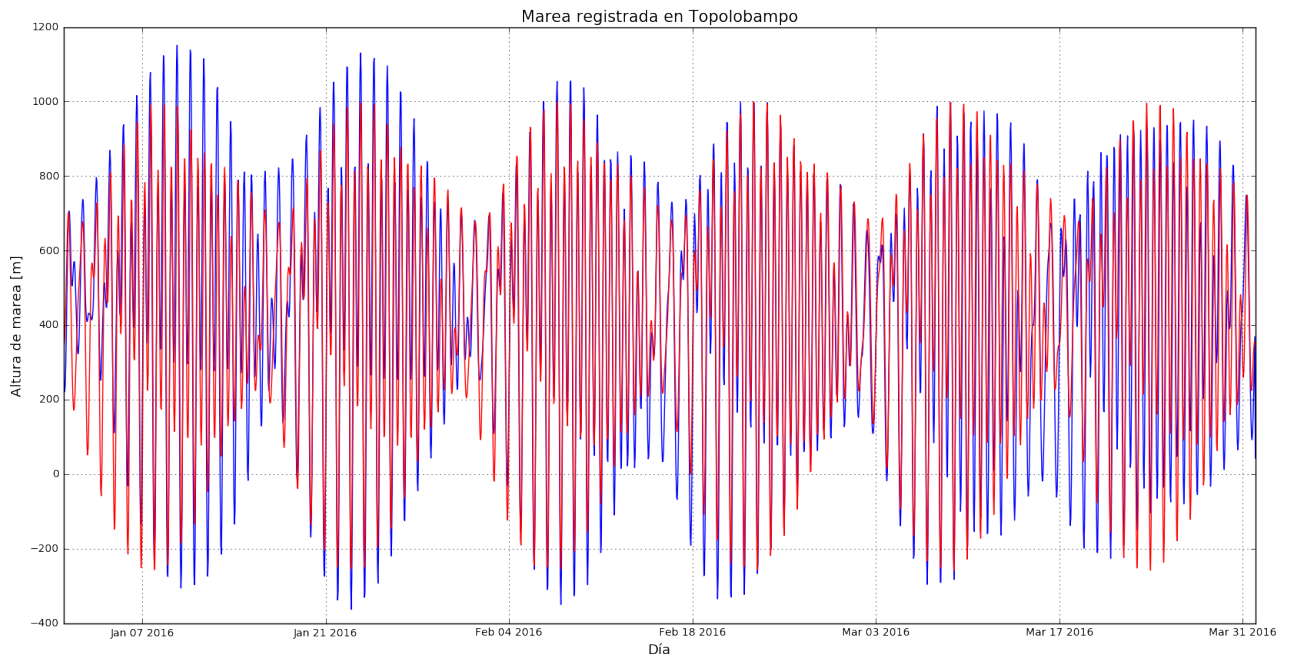
3.2.3. Código de fase

```
fO1s = np.angle(yf[int(1092)])
TS1s = np.angle(yf[int(85),])
TN2s = np.angle(yf[int(91),])
TM2s = np.angle(yf[int(176),])
TS2s = np.angle(yf[int(182),])
```

3.2.4. Función

```
y = df['Altura'] w = 2,0 * np.pi
def f(t) :
return A0m + 2 * (O1m * np.cos(w * TS1 * t + TS1s) + S1m * np.cos(w * TN2 * t +
TN2s) + N2m * np.cos(w * TM2 * t + TM2s) + M2m * np.cos(w * TS2 * t + TS2s))
```

3.2.5. Gráfica



4. Error relativo

Se calculó el error relativo respecto a la gráfica original y la gráfica obtenida mediante el anterior procedimiento. Se mostrará el código y una tabla con sus errores de las mareas de los lugares estudiados.

4.1. Código

$$y_0 = df['Altura']$$

$$y_1 = f(df['T'])$$

$$E = \text{sum}(\text{abs}(y_0 - y_1) ** 2) / \text{sum}(\text{abs}(y_0) ** 2)$$

E

Lugar	Error relativo
Baltimore	23.92 %
Topolobampo	6.24 %