Taller: Transformada de Fourier- parte 1

Juan José Segura Flórez
201510684

26 de septiembre de 2019

1. Problemas

- Elabore un programa para calcular la transformada de Fourier $g(\omega)$ de las siguientes funciones.

  (a) $f(t) = e^{-t^2}$

  (b) $f(t) = e^{-t^2} \cos t$

  En ambos casos, usted debe comparar el resultado numérico con el analítico.

- Calcule la transformada discreta de Fourier de los datos en el archivo adjunto al presente taller.
- Vuelva a realizar el punto anterior, haciendo uso de la transformada rápida de Fourier.

2. Solución punto 1

2.1. Inciso a

Para este ejercicio se obtó por hacer uso de un paradigma de programación distinto al que se ha venido usando (paradigma estructurado de programación); todo el punto 1 está hecho usando el paradigma de programación tipo POO (programación orientada a objetos).

En la carpeta llamada *punto_1* se encuentra un programa llamado *punto_1_a.py*. Dicho programa soluciona el problema 1 en el inciso a mediante una clase que se ha llamado *Fourier_Transform*. Dicha clase posee 4 atributos:

- function: dicho atributo hace referencia a la función que se usará para realizar la transformada de fourier inversa, es decir, $f(t)$.
- N: dicho atributo hace referencia al número de puntos usados para gráficar la función $f(t)$.
- Omega: dicho atributo hace referencia a la frecuencia angular.
- lim_min: dicho atributo hace referecia al límite inferior de integración en t.
- lim_max: dicho atributo hace referencia al límite superior de integración en t.

En la clase existen dos tipos de métodos:

- Métodos *set*: Estos métodos permiten seleccionar o elegir un valor dado para cada uno de los atributos.
- Métodos *get*: Estos método permiten obtener un valor que posee un objeto o que cálcula un objeto.

Tanto el programa *punto_1_a.py* como el programa *punto_1_b.py* hacen uso de la misma clase, sin embargo, el uso de la clase se diferencia en la introducción de funciones distintas como argumento para la creación de un objeto llamado *object_1*. El resultado final del programa está gráficado en la figura 1.
Como se puede ver en la figura 1, la parte imaginaria es un valor muy cercano a cero, mientras que la parte real de la transformada de Fourier inversa númerica se parece mucho a la transformada de Fourier inversa analítica (véase figura 2). La fórmula analítica que permite hallar la transformada de Fourier inversa es:

$$g(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-t^2} e^{i\omega t} dt \tag{1}$$

Si se realiza la integral en Mathematica:

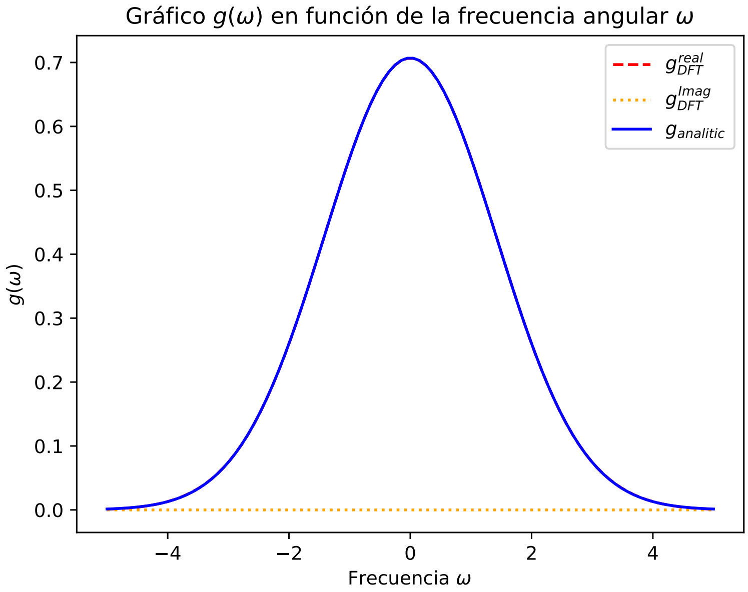

Figura 1: Resultado de la transformada de Fourier inversa: En línea azul se encuentra la transformada de Fourier analítica. En líneas interrumpidas roja se encuentra la transformada de Fourier inversa númerica parte real (no se alcaza a ver porque es parecida a la transformada de Fourier analítica). En línea punteada naranja se encuentra la transformada de Fourier inversa númerica parte imaginaria.

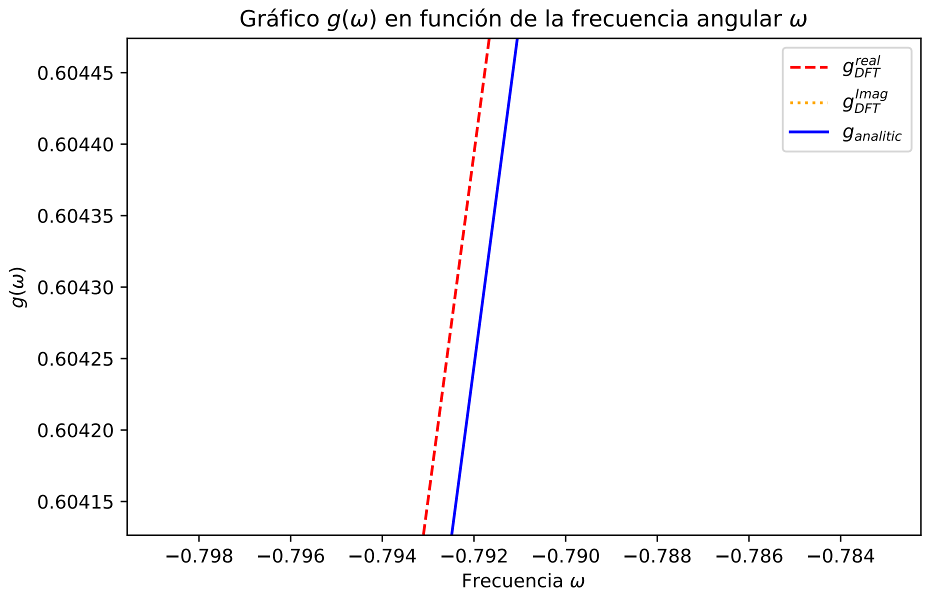

Figura 2: Resultado de la transformada de Fourier inversa: Zoom para comparar la transformada de Fourier inversa númerica parte real con la transformada de Fourier inversa analítica

```
Integrate[
 1/Sqrt[2*Pi]*Exp[-t^2]*Exp[I*omega*t], {t, -Infinity, Infinity}]
```

el resultado será:

$$g(\omega) = \frac{e^{-\frac{\omega^2}{4}}}{\sqrt{2}} \tag{2}$$

La cual es la función dibujada en línea azul en la figura 2.

Finalmente, se ha calculado el error relativo promedio de cada punto obtenido en la transformada de Fourier inversa númerica con respecto a los puntos obtenidos en la transformada de Fourier inversa analítica mediante la siguiente fórmula:

$$\text{Error} = \frac{1}{N}\sum_i \left|1 - \frac{g_i^{númerical}}{g_i^{analitic}}\right| \tag{3}$$

Y para este caso el error relativo es:

$$\text{Error} = 0{,}012964024687989063 \tag{4}$$

2.2. Inciso b

El procedimiento es el mismo que el anterior inciso salvo que se está usando otra función, analíticamente (y usando Mathematica) se tendrá que la transformada de Fourier inversa es:

$$g(\omega) = \frac{1 + e^{\omega}}{2\sqrt{2}} e^{-\frac{1}{4}(1+\omega)^2} \tag{5}$$

Y al gráficar se obtiene la figura 3, que tiene una fórmula similar a la transformada de Fourier inversa del anterior inciso.

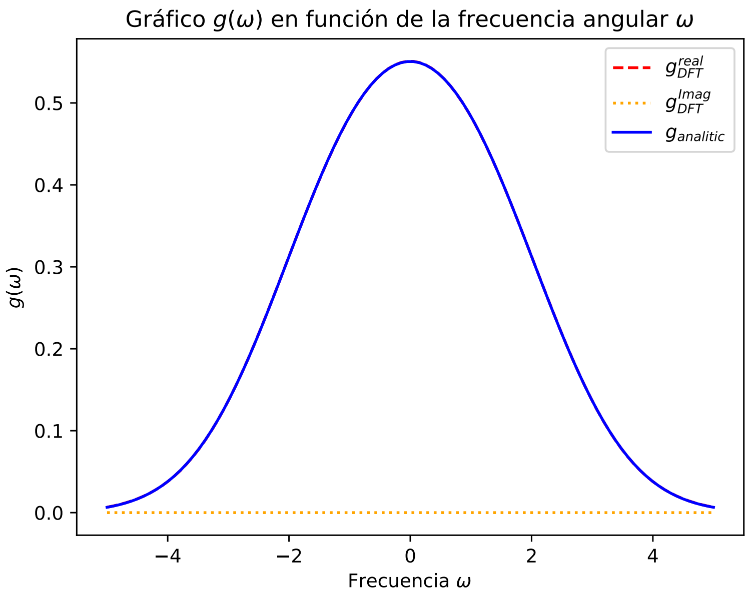

Figura 3: Resultado de la transformada de Fourier inversa: En línea azul se encuentra la transformada de Fourier analítica. En líneas interrumpidas roja se encuentra la transformada de Fourier inversa númerica parte real (no se alcaza a ver porque es parecida a la transformada de Fourier analítica). En línea punteada naranja se encuentra la transformada de Fourier inversa númerica parte imaginaria.

Como se puede ver en la figura 3, la parte imaginaria de la transformada de Fourier inversa númerica es prácticamente 0 al igual que en el anterior problema. Si se hace un zoom en la gráfica 3 obtendremos la gráfica 4. Finalmente, se calcula el error relativo promedio de cada punto obtenido mediante la transformada de Fourier inversa númerica con respecto a los puntos obtenidos de la transformada de Fourier inversa analítica obteniendo:

$$\text{Error} = 0{,}003020151969642654 \tag{6}$$

Lo cual dice que el error relativo es menor para este inciso que para el inciso anterior.

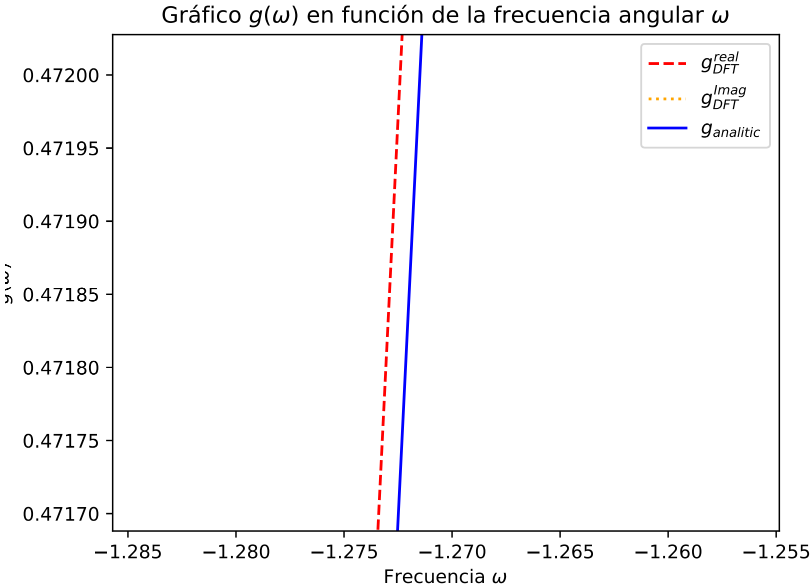

Figura 4: Resultado de la transformada de Fourier inversa: Zoom para comparar la transformada de Fourier inversa númerica parte real con la transformada de Fourier inversa analítica

3. Solución punto 2

La solución a este problema está en el programa *punto_2.py*. El tiempo de cálculo de esta transformada es de 8,930188417434692 segundos para el computador con el cual trabajo. Y las gráficas que se obtienen son mostradas en la figura 4.

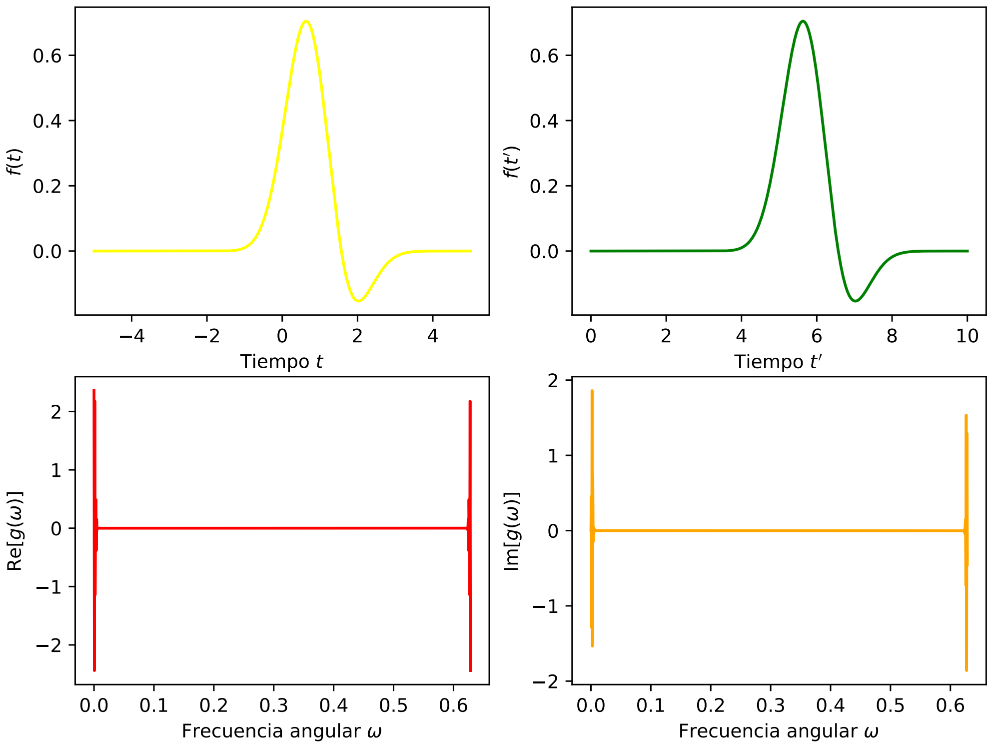

Figura 5: Graficas de $f(t)$ vs t, $f(t)$ vs t', Re$[g(\omega)]$ vs ω y Im$[g(\omega)]$ vs ω.

4. Solución punto 3

La solución a este problema está en el programa *punto_3.py*. El tiempo de cálculo de esta transformada es de 2,9752535820007324 segundos para el computador con el cual trabajo. Y las gráficas que se obtienen son mostradas en la figura 5. En conclusión, los gráficos de este punto y el anterior son los mismos, solo que el

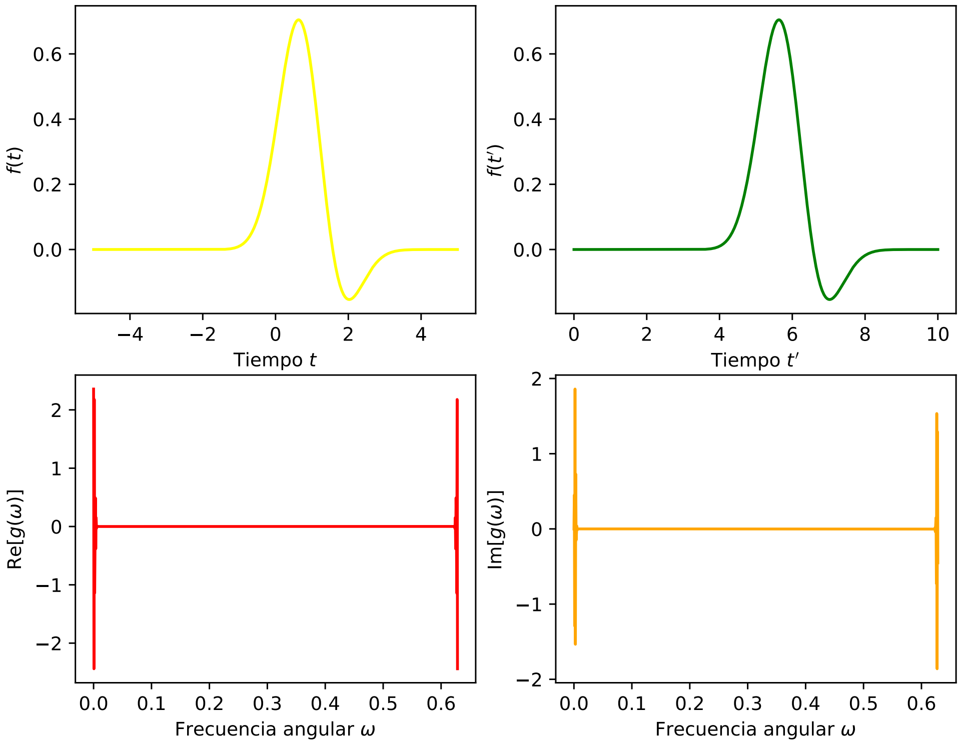

Figura 6: Graficas de $f(t)$ vs t, $f(t)$ vs t', Re$[g(\omega)]$ vs ω y Im$[g(\omega)]$ vs ω.

algoritmo hace más rápidamente la transformada de fourier inversa.