

# Métodos Computacionales

## Tarea 2 — 2019–10

Parte I: para que vayan avanzando en la tarea...

Los archivos:

`hw2.mk`,

`Fourier.py`

`Plots_hw2.py`

`Resultados_hw2.tex`. y Los que falta por anunciar...,

deben estar comprimidos en `ApellidoNombre_hw2.zip` y deben descomprimirse en un directorio `ApellidoNombre_hw2`. Este directorio comprimido debe contener únicamente los archivos mencionados anteriormente (más los archivos de la parte 2) y debe subirlo a SICUA antes de las 8:00 pm del lunes 6 de mayo de 2019. Adicionalmente debe haber trabajado su tarea en un repositorio de GitHub cuyo enlace debe también subir a sicua. Su repositorio debe tener varios commits hechos durante las 4 semanas que tiene para trabajar en la tarea. Recuerde que es un trabajo completamente individual (no puede usar códigos que usted no haya escrito en su totalidad ni puede recibir ayuda de nadie cuando esté escribiendo su código). Recuerde también que los códigos deben correr sin botar errores en los computadores de compufísica (Y110B). Si su código no corre la nota de ese ejercicio será de cero. Recuerde que es buena práctica comentar el código y elegir buenos nombres para sus variables. Eso será tenido en cuenta durante el proceso de corrección de esta tarea. No se aceptarán trabajos entregados tarde ni por otro medio distinto a SICUA. El comando `gmake -f hw4.mk` debe ser suficiente para que todos los códigos de la tarea corran, se generen las gráficas y se produzca el archivo `Resultados_hw2.pdf`.

### 1. ( points) **Github**

En esta tarea se evaluará el uso de Github de la siguiente manera:

- Los archivos de solución deben, además de subirse a SICUA, estar en un repositorio cuyo enlace debe ser enviado por SICUA cuando se envíe la solución de la tarea.
- Debe haber al menos 15 commits de la tarea en dónde se vea claramente la **evolución** del trabajo (no es válido hacer varios "commits" justo antes de la hora de entrega de la tarea).
- Los commits se deben hacer para los archivos individuales de la tarea a medida que su trabajo en cada archivo vaya avanzando. Se espera que haya commits durante al menos 3 de las 4 semanas que tienen para trabajar en la tarea.

### 2. ( points) **Transformada de Fourier: implementación propia, paquetes de scipy y espectrogramas.**

La idea de este ejercicio es implementar un código que les permita entender y hacer espectrogramas de señales (ver por ejemplo: <https://pnsn.org/spectrograms/what-is-a-spectrogram>). En este ejercicio deben primero usar su implementación propia de la transformada de fourier para analizar dos señales isimples y luego usar paquetes de scipy y de matplotlib para hacer espectrogramas de una señal real de un sismo. No olviden poner labels y leyendas en las gráficas. **NOTA importante:** TODAS las graficas mencionadas a continuacion deben guardarse **sin ser mostradas** y ser incluidas en el pdf de resultados. Para esto debe escribir un script de python llamado `Fourier.py` que:

- Almacene los datos de `signalSuma.dat` y de `signal.dat`. Estas dos señales están conformadas por dos ondas sinusoidales. En el primer caso la señal es la suma de las dos ondas, en el segundo la señal está conformada por, primero una de las ondas y luego la siguiente.
- Haga una gráfica con dos subplots uno con los datos de `signal.dat` y otro con los de `signalSuma.dat`.
- Haga la transformada de Fourier de ambas señales usando su **implementación propia** de la transformada discreta de Fourier.
- Haga una gráfica de las transformadas de Fourier de ambas señales. Esta gráfica debe ser en función de las frecuencias (bueno si no usa el paquete `fttfreq`. Indique esto con un mensaje en la terminal.)
- Usando el paquete `matplotlib.pyplot.specgram` (ver: [https://matplotlib.org/api/\\_as\\_gen/matplotlib.pyplot.specgram.html](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.specgram.html)) haga un espectrograma de las dos señales.
- Almacene los datos de `temblor.txt`. Estos datos son datos reales de una señal sísmica
- Haga una gráfica de la señal en función del tiempo.
- Haga la transformada de Fourier de la señal usando paquetes de `scipy` y gráfíquela.
- haga un espectrograma de la señal.

**NOTA importante:** En el pdf de resultados debe incluir todas las gráficas mencionadas anteriormente y debe incluir además un análisis de sus resultados. Debe en particular: Para las señales `signal.dat` y `signalSuma.dat` describir las diferencias en las señales, las frecuencias de los picos principales de las transformadas de Fourier y las diferencias entre los espectrogramas. Para la señal `temblor.txt` debe describir la señal original, la transformada de Fourier y explicar explícitamente qué puede ver en el espectrograma. Estas explicaciones deben ser cortas y **precisas**.

### 3. ( points) **Gráficas, makefiles y pdf de resultados.**

El código `Plots_hw2.py` debe:

- Leer y guardar los datos generados por los códigos en C++ de los ejercicios de ecuaciones diferenciales que serán anunciados después de semana santa.
- Graficar. Guarde dichas gráficas sin mostrarlas (para poder luego incluirlas en el pdf de resultados).

Modifique el archivo `Resultados_hw2.tex` para poder:

- Organizar las gráficas obtenidas Haga una sección por cada ejercicio de la tarea y describa brevemente sus resultados. En el ejercicio de Fourier, debe incluir todas las gráficas y debe explicar lo que observa en las gráficas. Debe en particular describir lo que observa en las gráficas de espectrogramas (también llamados time frequency plots). Este archivo debe estar incluido dentro de las dependencias del makefile y debe permitir generar un archivo `Resultados_hw2.pdf`

El archivo `hw2.mk` debe:

- Incluir todas las dependencias y reglas necesarias para generar y actualizar el archivo: `Resultados_hw2.pdf`. Los archivos que deben subir a Sicua (comprimidos en `ApellidoNombre_hw2.zip`) son:

hw2.mk,  
Fourier.py  
Plots\_hw2.py  
Resultados\_hw2.tex. y Los que falta por anunciar...,