Tarea programada 1

Esta tarea puede realizarse en parejas o individual. Por favor asegúrese de enviarle al asistente los dos nombres y números de carné.

Para esta tarea, puede escoger entre los siguientes lenguajes de programación o paquetes:
C/C++, Fortran, Python, Mathematica, Maxima y gnuplot.
Puede usar una combinación de lenguajes o paquetes, y puede usar paquetes adicionales de cálculo científico para C, Fortran of Python (p.ej., GLS, Sympy, Numpy o Matplotlib). El uso de inteligencia artificial está prohibida porque el objetivo de la tarea es ejercitar sus habilidades de solución de problemas. En un documento PDF aparte, usted debe colocar los resultados y respuestas a los enunciados de abajo, y añadir un pequeño párrafo que describa cómo lo hizo, un solo párrafo por problema. Por ejemplo, si usted define una función

```
def puntos_menos_cero(n,L):
    puntos = np.linspace(-L,L,n)
    puntos[np.isclose(puntos,0)] = np.nan
    return puntos
```

su descripción debe contener una frase que diga algo así como "hicimos una función que crea una lista de puntos y se asegura que ninguno de ellos sea cero". Atención: ese párrafo no tiene puntos, pero si no aparece no se calificarán los demás resultados del problema. Ud. debe enviar el código/códigos que utilizó para generar los resultados como archivos adjuntos a su solución. El contenido del párrafo de descripción debe ajustarse al código que usted envió. Se harán controles aleatorios de correspondencia entre lo que dice el párrafo y el código que usted envió. El objetivo de estos controles es disuadir la simple copia de resultados e incentivar que el estudiante (-ae) comprenda cómo se resuelve el problema de forma programada.

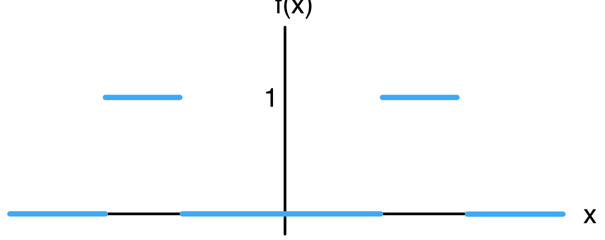
Cada problema completo vale 1/5 del total de puntos, y las subdivisiones del problema dividen equitativamente su puntaje.

Lista de tareas

Sugerencia para empezar (no calificado):

Pruebe hacer la transformada de la función con un solo escalón $f(x) = \mathrm{rect}(x)$ (cuya transformada sabemos que es $F(k) = \mathrm{sinc}(k/2)$) con fast Fourier transform (FFT). Grafique la parte real e imaginaria de la transformada. ¿Hacia adónde en esos arrays mapean las "frecuencias" (k) negativas?

1. Considere una función cuya gráfica sea



(usted define el ancho de la parte que no es cero)

1.1 Calcule la transformada de Fourier numérica de $f(x)$. ¡Tenga cuidado con el intervalo de definición!

1.2 Grafique la función $A(k) = F(k)\bar{F}(k)$, donde $F(k) = \mathscr{F}\{f(x)\}$.

1.3 Grafique la función $A(k)$ cuando se cambia el ancho de ambos escalones

1.4. Considere una onda luminosa plana que pasa por dos rendijas. ¿Qué relación tiene la función $A(k)$ que usted calculó arriba con el patrón que la luz hace en una pared muy lejana? Revise su conocimiento de óptica ondulatoria y escriba la ecuación que usted está resolviendo con este ejercicio.

2. Calcule la transformada de Fourier numérica bidimensional de una función definida por

$$f(r) = \begin{cases} 1, & r \leq a \\ 0, & r > a \end{cases}$$

Es decir, un círculo. (Sugerencia: convierta a coordenadas Cartesianas y construya un array bidimensional, use en numpy la función ff2). Usted elige a. Grafique la función $F(\mathbf{k})\bar{F}(\mathbf{k})$, como en el ejercicio anterior. En Python, use imshow o pcolormesh.

3. Acompañando este enunciado, usted encontrará tres archivos de texto con una columna de floats. Estos archivos contienen ondas sonoras que representan:

"signal1.txt" : la nota "la" de un instrumento musical

"vocalE.txt": una vocal humana "e"

"vocalU.txt": una vocal humana "u"

Las ondas sonoras son muestras que duran 1 segundo a una tasa de 44100 bit/s (es decir, el tiempo que corresponde a esas ondas sonoras es un array que va de 0 a 1 segundo y que tiene 44100 entradas).

3.1 Grafique cada onda sonora en función del tiempo

3.2 Calcule la transformada numérica de Fourier de cada señal.

3.3 ¿Cuáles frecuencias hay en cada señal?

4. Considere las funciones, definidas en $0 \leq x \leq 1000$.

$$f_1(x) = \begin{cases} 1, & |x| \leq 2 \\ 0, & |x| > 2 \end{cases}$$

$$f_2(x) = x$$

$$f_3(x) = x + 500$$

$$f_4(x) = \exp(-x^2)$$

4.1 Grafique todas las funciones anteriores

4.2 Calcule la transformada de Fourier de cada una

4.3 Calcule la transformada discreta seno de Fourier (sugerencia: scipy.fftpack)

4.4 Calcule la transformada discreta coseno de Fourier

4.3 En una misma gráfica pero una para cada función, grafique la parte real, la parte imaginaria y el valor absoluto ($\sqrt{F\bar{F}}$) de la transformada de Fourier

4.4 En una misma gráfica pero una para cada función, grafique la transformada seno y la transformada coseno.

4.5 Comente: ¿Cómo se compara cada transformada? ¿Qué partes de la función tiene cada transformada y qué partes faltan? ¿Para qué funciones son iguales las transformadas de Fourier y las de seno y coseno?

5. Transformada de Fourier de puntos en 2D:

5.1 Construya tres arrays bidimensionales con 30 entradas en cada dirección, y llénelos de ceros.

5.2 Cambie las siguientes entradas a 1:

array1[5,5], array1[5,10]
array2[5,5], array2[25,5]
array3[10,15], array3[25,20]

5.3 Calcule la transformada bidimensional de Fourier para cada array

5.4 Grafique las partes reales e imaginarias de cada array transformado (imshow o pcolormesh en Python)

5.5 Basado en los resultados, conteste: ¿Qué pasa con la transformada cuando las entradas que son 1 ("puntos") se acercan o alejan? ¿Qué pasa cuando esos puntos se rotan 90º? ¿Qué pasa si esos puntos se mueven en un ángulo distinto?

La cristalografía de rayos X se basa en pares de transformada de Fourier. Los rayos X provocan un patrón de difracción en un detector lejano, y se obtiene la transformada de Fourier del patrón del cristal. Con la transformada inversa puede obtenerse la estructura del cristal.