TAREA 1

Fecha de entrega: 27/09/2018 23:59 hrs

Problema 1 (40 %)

En clase vimos un método sencillo de para estimar la derivada de una función el cuál produce errores de orden $\mathcal{O}(h)$. La siguiente expresión para estimar la derivada de una función produce errores de orden $\mathcal{O}(h^4)$ (a cambio de un mayor número de evaluaciones de la función):

$$f'(x) = \frac{-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h)}{12h} + \mathcal{O}(h^4)$$

Compare el método más sencillo con el método propuesto de orden $\mathcal{O}(h^4)$ considerando la función $f(x) = -cos(x)$ para $x = 1.\text{XXX}$ radianes (donde XXX corresponde a los 3 últimos dígitos de su RUT, antes del dígito verificador). Para ello:

1. defina un rango apropiado de valores h a explorar y compare su estimación numérica de la derivada con el valor entregado por la función `math.sin(1.`XXX`)` de `python`.

2. Primero asegúrese de hacer todos sus cálculos utilizando números de tipo `float32`. Luego compare su resultado con el resultado que se obtendría utilizando números de tipo `float64` ó `float128` si su computador soporta el tipo de arquitectura correcto.

En su informe, explique el comportamietno observado al hacer estas comparaciones, en particular explique: ¿dónde esta la ganancia entre un método de $\mathcal{O}(h^4)$ vs. un método $\mathcal{O}(h)$?; ¿cuál es la ganancia al usar números de mayor precisión?; ¿por qué la exactitud no mejora monotónicamente con un menor h?

> **Ayuda.** Ud. debe decidir qué gráficos son los más interesantes para hacer la comparación que se le pide. En particular, en este caso tiene sentido utilizar escalas logarítmicas en ambos ejes. Utilice como guía el `jupyter notebook` que se utilizó en clases para demostrar un ejemplo similar.

Problema 2 (60 %)

Poco después del Big Bang, el Universo era denso y muy caliente, la radiación y el plasma formado por protones y electrones libres se mantenían en equilibrio térmico. Con la expansión del Universo tanto la radiación como el plasma se enfrían. Eventualmente la temperatura baja lo suficiente para que protones y electrones se combinen en átomos neutros. La radiación remanente, por su parte, no puede ser absorbida por estos átomos neutros y comienza a viajar libremente por el Universo. Dicha radiación, que continua perdiendo energía/disminuyendo su temperatura, fue detectada por primera vez por Arno Penzias y Robert Wilson en 1964 (Premio Nobel de Física 1978) y se le conoce como la radiación de fondo de microondas, es una de las evidencias más sólidas de que el Universo fue alguna vez mucho más denso y caliente que hoy.

La teoría predice que la radiación remanente del Big Bang debería tener el espectro (distribución de energía por unidad de frecuencia) de un cuerpo negro. La radiación de un cuerpo negro en unidades de [Energía / tiempo / Area / frecuencia / ángulo sólido] está dada por la función de Planck:

$$B_\nu(T) = \frac{2h\nu^3/c^2}{e^{h\nu/k_BT} - 1}$$

donde h es la constante de Planck, c es la velocidad de la luz en el vacío, k_B es la constante de Boltzmann, T es la temperatura del cuerpo negro y ν es la frecuencia de la radiación. En 1989 se puso en órbita el satélite COBE (Cosmic Background Explorer) con el objetivo de estudiar en detalle la radiación de fondo de microondas. Usaremos sus datos para intentar determinar la temperatura de la radiación de fondo.

1. El archivo `firas_monopole_spec_v1.txt` contiene el espectro del monopolo medido por el instrumento `FIRAS` del satélite `COBE` (más información en el siguiente link). Explore el archivo para encontrar las unidades. Lea el archivo y grafique el espectro de la radiación de fondo de microondas incluyendo la incertidumbre de cada punto. Recuerde anotar los ejes incluyendo las unidades.

   > **Ayuda.**
   >
   > - El módulo `numpy` contiene la rutina `numpy.loadtxt` que le puede ser útil para leer el archivo.
   > - Para plotear se recomienda usar el módulo `matplotlib`. Hay muchos ejemplos, con código incluido en el siguiente link, en particular, este ejemplo sencillo le puede ser útil.
   > - La medición de FIRAS es impresionante por su precisión. Las barras de error son muy pequeñas por lo que es difícil mostrarlas en un gráfico. Multiplíquelas por un factor grande (por ejemplo, 400) para que se puedan ver (no olvide indicar lo que hizo en el informe, *caption* de la figura, etc.)
   > - $1\mathrm{MJy} = 10^{-20}\mathrm{Wm^{-2}Hz^{-1}}$

2. Una forma (no la mejor) de medir la temperatura del cuerpo negro que da origen a esta radiación es utilizar algo parecido a la ley de Stefan-Boltzman. Lo haremos integrando la función de Planck en frecuencia, de lo cual resulta:

   $$P = \frac{2h}{c^2}\left(\frac{k_B T}{h}\right)^4 \int_0^\infty \frac{x^3}{e^x - 1} dx$$

   La integral se puede calcular analíticamente con resultado $\pi^4/15$ pero para efectos de esta tarea, elija un método apropiado y **calcule la integral numéricamente** para luego comparar con el resultado analítico. Implemente un algoritmo que permita ir refinando el valor de la integral con una tolerancia elegida por Ud.

   > **Ayuda.**
   >
   > - El módulo `astropy` contiene el submódulo `astropy.constants` que incluye todas las constantes necesarias además de rutinas para cambiar unidades. Le podría ser útil pero no es necesario que lo use.
   > - La integral que es necesario calcular es entre 0 e ∞ así que requiere ser normalizada. Puede intentar el cambio de variable $y = arctan(x)$ u otro que le parezca conveniente.

3. Ahora elija un método apropiado para integrar en frecuencia el **espectro observado** (el del archivo `firas_monopole_spec_v1.txt`). Se pide que escriba su propio algoritmo para llevar a cabo la integración, más adelante usaremos librerías de libre disposición. Iguale su resultado con la integral de la función de Planck calculada en el punto anterior (teniendo cuidado con

las unidades), la única variable libre debería ser la temperatura que Ud. puede ahora calcular resolviendo la ecuación que acaba de escribir. Su determinación debería ser levemente distinta a la temperatura reportada por COBE de 2.725 K. ¿Qué puede explicar la diferencia?

4. Repita su gráfico de la parte 1. pero esta vez agregue una función de Planck con una temperatura de 2.725 K y otra con la temperatura que Ud. calculó en la parte 3.

5. El módulo `scipy` incluye las funciones `scipy.integrate.trapz` y `scipy.integrate.quad`. Utilícelos para re-calcular las integrales calculadas en 2. y 3. (según corresponda, revise la ayuda para averiguar cuál función aplica en cada caso). Compare los valores obtenidos y la velocidad de ejecución del algoritmo escrito por Ud. vs. el de `scipy`. ¿A qué se debe la diferencia?

   > **Ayuda.** En la consola `ipython` existe la `ipython magic` `%timeit` que permite estimar velocidades de funciones.

Otras instrucciones importantes.

- Lea siempre estas instrucciones, **no son las mismas en todas las tareas** y las diferencias suelen ser importantes.

- Utilice `git` durante el desarrollo de la tarea para mantener un historial de los cambios realizados. La siguiente cheat sheet le puede ser útil. Esto no será evaluado esta vez pero evaluaremos el uso efectivo de git en el futuro, así que empiece a usarlo.

- La tarea se entrega como un `push` simple a su repositorio privado. El `push` debe incluir todos los códigos usados además de su informe.

- El informe debe ser entregado en formato `pdf`, este debe ser claro sin información ni de más ni de menos. Esto es importante, no escriba de más, esto no mejorara su nota sino que al contrario. 5 páginas es un largo razonable para la presente tarea. Asegúrese de utilizar figuras efectivas y/o tablas para resumir sus resultados. Revise su ortografía.

- No olvide indicar su RUT en el informe.