Métodos Numéricos para la Ciencia e Ingeniería
FI3104-1

Tarea 1

Maximiliano Dirk Vega Aguilera

1 Introducción

La tarea 1 consistió en calcular de forma numérica la luminosidad del Sol, a partir del espectro solar medido en desde la Tierra, y calcular el radio solar a partir de la luminosidad obtenida y la ecuación de cuerpo negro de Plank conociendo la temperatura efectiva del Sol.

El espectro solar nos indica la potencia medida en cada longitud de onda. Los datos utilizados entregan longitud de ondas (λ) en [nm] y la potencia por unidad de área por unidad de longitud de onda [W m^{-2} nm^{-1}], esto es equivalente al flujo recibido por unidad de longitud de onda.

Al integrar el flujo por longitud de onda en λ se obtiene el flujo total recibido (F_r). A partir de este flujo se puede calcular la luminosidad solar a partir de la distancia Tierra-Sol mediante la ecuación:

$$L = 4\pi d^2 F_r \tag{1}$$

Conociendo la temperatura efectiva del Sol, y suponiendo que se comporta como cuerpo negro, se puede calcular la radiación que debe emitir para estar a esa temperatura mediante la ecuación de cuerpo negro de Planck:

$$B_\lambda(T) = \frac{2\pi hc^2/\lambda^5}{e^{hc/\lambda k_B T} - 1} \tag{2}$$

Esta ecuación nos entrega el flujo por longitud de onda emitido por el Sol, al integrarlo en todo λ se obtiene el flujo total emitido (F_e). Utilizando este flujo y conociendo la luminosidad se puede calcular el radio del Sol mediante:

$$L = 4\pi R^2 F_e \tag{3}$$

Notar que las ecuaciones (1) y (2) son la misma ecuación. La luminosidad depende del flujo medido y de la distancia a la que se mide.

2 Desarrollo

Para el desarrollo de esta tarea se utilizó el lenguaje python y sus librerías numpy, matplotlib, astropy y scipy. Además utilizó el repositorio de git para mantener un registro del avance.

Lo primero que se hizo fue leer el archivo entregado por el profesor usando la rutina loadtxt de numpy y crear variables para la longitud de onda y la potencia. Una vez obtenido los datos del archivo se realizó un cambio de unidades a cgs con λ en [Å]. Luego se graficó Flujo por longitud de onda vs λ en escala log-log usando el paquete pyplot de matplotlib.

Lo segundo que se hizo fue calcula de forma numérica el flujo total a partir de los datos. Para ello se integró usando el método del trapecio, este método consiste en tomar dos puntos consecutivos, calculando el área formada por el trapecio de altura Δx y bases $f(x)$ de cada punto, y sumando todos los trapecios formados, como en la siguiente ecuación para el termino i y el siguiente:

$$I = \Sigma \frac{f(x_i) + f(x_{i+1})}{2}(x_{i+1} - x_i) \tag{4}$$

La integral se programó con un for, utilizando λ como variable x y la potencia como $f(x)$. Luego de calcularla se le agregaron las unidades en cgs usando el paquete units de astropy. Una vez obtenido el flujo, se calculó la luminosidad del Sol usando la ecuación (1) tomado como distancia Tierra-Sol la unidad astronómica $1.5\text{x}10^{13}$ [cm].

En tercer lugar se calculó el flujo total a partir de la radiación de cuerpo de Plank para el Sol, sabiendo que su Temperatura efectiva (T) es 5778 [K]. Para ello se integró la ecuación (2) en todo λ (de 0 a ∞). Lo primero que se notó fue que mediante el apropiado cambio de variable, la integral de la ecuación (2) es equivalente a:

$$P = \frac{2\pi h}{c^2}\left(\frac{k_B T}{h}\right)^4 \int_0^\infty \frac{x^3}{e^x - 1} dx \tag{5}$$

El término constate se calculó a parte de la integral usando el paquete constants de astropy. La integral se calculó mediante el método del trapecio, pero dado que no se puede sumar hasta infinito primero se realizó el cambio de variable $x = tan(x) \rightarrow dx = sec^2(x)dx$, con lo que la integral quedó de la forma:

$$I = \int_0^{\pi/2} \frac{tan(x)^3}{e^{tan(x)} - 1} \frac{1}{cos(x)^2} dx \tag{6}$$

Esta integral también se programó con un for en un intervalo $[\epsilon, \frac{\pi}{2} - \epsilon]$, dado que en 0 y pi/2 se indetermina, con puntos equiespaciados. Tanto ϵ como el paso fueron de 0.001. Una vez obtenido el flujo y utilizando la luminosidad obtenida anteriormente se calculó el radio solar usando la ecuación (3).

Finalmente, las integrales antes calculadas se recalcularon usando el paquete integrate de scipy. Para la primera integral se usó la rutina trapz usando las mismas variables. Para la segunda integral se usó la rutina quad, y debido a

que el programa tiraba un error se integro la función integrada en la ecuación 5. Luego se compararon los resultados entregados por mis rutinas y las rutinas scipy.

3 Resultados

Al graficar el espectro solar se obtuvo el gráfico de la figura1

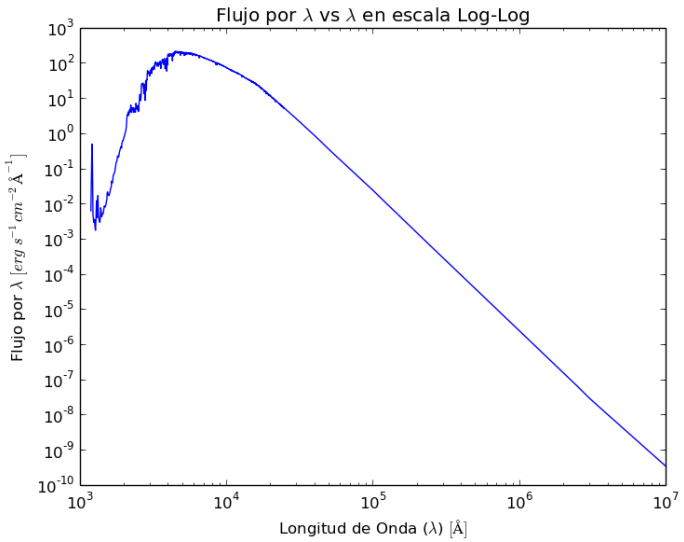

Figure 1: Espectro solar obtenido a partir de los datos del archivo 'sun_AM0.dat'.

Los valores numéricos obtenidos se resumen en la tabla 1.

Parámetro	Resultado	Unidades
Flujo Recibido	1366090.79	erg / (cm2 s)
Luminosidad	3.86×10^{33}	erg / s
2^a Integral vs $\frac{\pi^4}{15}$	4.99×10^{-10}	-
Flujo Emitido	63200679707.6	erg / (cm2 s)
Radio del Sol	69738109053.6	cm
Flujo Recibido sci	1366090.79	erg / (cm2 s)
Flujo Emitido sci	63200679712.4	erg / (cm2 s)
Δ Flujo Recibido	6.98×10^{-10}	erg / (cm2 s)
Δ Flujo Emitido	4.86	erg / (cm2 s)

Table 1: Valores obtenidos numéricamente.

Debido a que no se supo implementar %timeit para mis rutinas no se in-

cluyen las velocidades de ejecución ni las comparaciones con las del paquete scipy.integrate.

4 Conclusión

Los resultados obtenidos son coherentes con los valores reales de los parámetros, lo que nos indica que calcular de forma numérica es una forma de obtener resultados correctos, aun cuando se hacen ciertas aproximaciones en los límites complicados.

Por otro lado las diferencias entre los valores obtenidos con mis rutinas y los obtenidos con las rutinas de scipy pequeñas, siendo el obtenido por la rutina trapz prácticamente igual al obtenido por mi rutina, pero notandose una pequeña diferencia en la rutina quad vs la mía.