Métodos numéricospara la ciencia e Ingeniería: Informe Tarea 1

José Guillermo Araya

September 20, 2015

1 Pregunta 1

1.1 Introducción

Se busca graficar el espectro del Sol, esto es, flujo vs longitud de onda, en unidades cgs a partir de los datos importados desde el archivo sun_AMO.dat, que corresponden al espectro medido justo afuera de la atmósfera

1.2 Procedimiento

En primer lugar se importan los datos desde el archivo mencionado con el comando "numpy.loadtxt()", lo que crea un arreglo bidimensional con los valores a trabajar, donde la primera columna corresponde a la longitud de onda en nm, y la segunda a la energía por unidad de tiempo por unidad de área por unidad de longitud de onda correspondiente, a cada coolumna mencionada se copia en un arreglo unidimensional, w_length para la longitud de onda, y flux para el flujo. El arreglo de longitudes de onda es multiplicado por 10, para pasar desde nm a Angstrom, y el de flujo es multiplicado de la misma manera por 10 para obtener unidades cgs. Finalmente se grafican ambos arreglos usando escala logarítmica para su mejor visualización.

1.3 Resultados

El gráfico final del espectro solar se puede observar en la figura 1

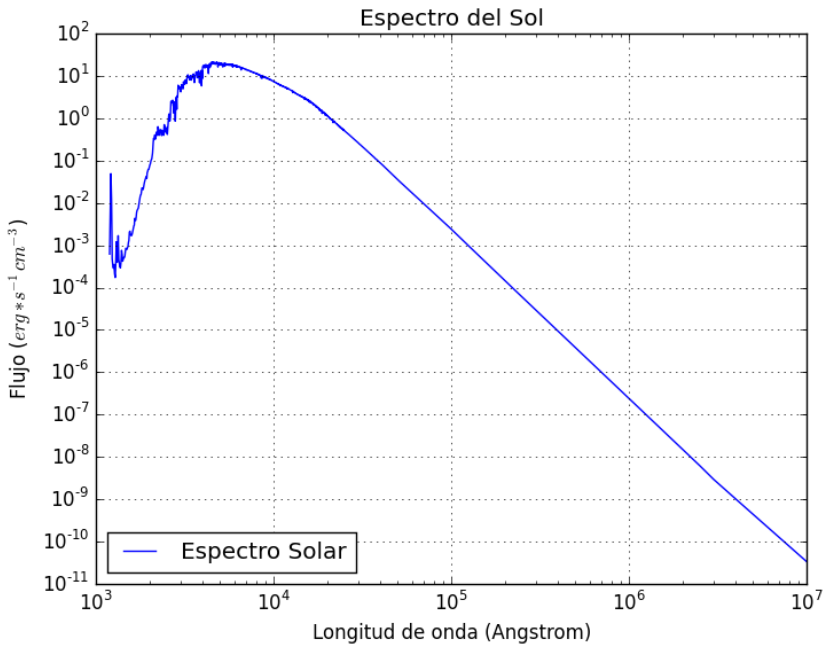

Figure 1: Espectro solar:Flujo ($erg*s^{-1}cm^{-3}$) vs Longitud de onda (Angstrom)

1.4 Conclusiones

Podemos ver que el gráfico arroja la forma esperada del espectro solar, es decir, decrece hacia el infinito luego de alcanzazar un peak cercano a los $5 * 10^3$ Angstrom (color verde)

2 Pregunta 2

2.1 Introducción

Se requiere integrar el espectro del Sol utilizando un método de integración numérica y a partir de este resultado encontrar la luminosidad total del SOl

2.2 Procedimiento

Para realizar la integración numérica se programará el método de los trapecios, donde la fórmula viene dada por:

$$\int_a^b f(x)dx \approx \frac{(b-a)}{2}(f(a)+f(b))$$

De modo que se calcula la integral entre cada par de datos consecutivos y luego se suman todos los valores para obtener el resultado final. Este proceso se realiza con las unidades originales del archivo, de modo que integramos el flujo por unidad de área por unidad de longitud de onda en longitud de onda, luego este valor se multiplica por 10^3 para obtener unidades cgs. El valor de la integral corresponde a la luminosidad solar por unidad de área, de esta manera basta multiplicar por el área de una esfera de radio 1UA para obtener la luminosidad total, esto es porque los datos fueron tomados a una distancia de aproximadamente 1UA del sol.

2.3 Resultados

Del algoritmo mencionado se obtienen los siguientes valores:

$$I = \int_a^b F(x)dx \approx 1366090.79$$

en unidades de $erg * s^{-1} * cm{-}2$
Donde $F(x)$ es el flujo por unidad de área por unidad de longitud de onda, I es el flujo por unidad de área
Aplicando la corrección por el área de la esfera:

$$L = 4\pi(1,496 * 10^{13})^2 I = 3.84195 * 10^{33} erg * s^{-1}$$

Donde L es la luminosidad total del Sol.

2.4 Conclusiones

Dado un error de $\epsilon = 1.86264514923 * 10^{-9} erg * s^{-1} * cm^{-2}$ para la luminosidad por unidad de área, se concluye que la curva es lo suficientemente suave, la cantidad y distribución de datos favorable como para que la integral quede correctamente aproximada por el método de los trapecios programado. Realizando la aproximación de la esfera de radio 1UA se obtiene un valor de Luminosidad solar cercano a el valor esperado $L_{esperado} = 3.846 * 10^{33} erg * s^{-1}$

3 Pregunta 3

3.1 Introducción

Se debe integrar numéricamente.mediante un algoritmo que puede refinar el resultado de acuerdo a la tolerancia querida, la función de Planck para estimar la energía total por unidad de área emitida por un cuerpo negro con la temperatura efectiva del sol, para luego compararla con la energía total calculada en la pregunta 2 para estimar el radio efectivo del sol.

3.2 Procedimiento

Primero se probará que:

$$\int_0^\infty B_\lambda(T)d\lambda = \int_0^\infty \frac{2\pi hc^2/\lambda^5}{e^{hc/\lambda k_B T} - 1} d\lambda = \frac{2\pi h}{c^2}\left(\frac{k_B T}{h}\right)^4 \int_0^\infty \frac{x^3}{e^x - 1} dx$$

En efecto:
usando el cambio $x = \frac{hc}{\lambda k_B T}$

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

como el dominio es infinito requiere de normalización mediante el cambio $y = \arctan(x)$, entonces:

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

Usando que $1 + \tan(y)^2 = \frac{1}{\cos(y)^2}$, se obtiene finalmente:

$$\frac{2\pi h}{c^2}\left(\frac{k_B T}{h}\right)^4 \int_0^{\frac{\pi}{2}} \frac{(\tan(y))^3}{(e^{\tan y} - 1)}(1+\tan(y)^2)dy = \frac{2\pi h}{c^2}\left(\frac{k_B T}{h}\right)^4 \int_0^{\frac{\pi}{2}} \frac{(\tan(y))^3}{\cos(y)^2(e^{\tan y} - 1)}dy$$

que es integrable mediante el método de los trapecios. La fórmula compuesta para el método de los trapecios es:

$$\int_a^b f(x)dx \approx \frac{(b-a)}{2n}(f(a) + f(b) + \sum_{i=1}^{n-1} 2f(a + i\frac{b-a}{n})$$

Donde n es el número de subintervalos en que se divide $[a, b]$

Para resolver numéricamente la integral se creó el método "refinar_integral", que recibe una función, el punto de inicio, el punto final y la tolerancia pedida. Éste calcula el valor de la integral, luego divide cada subintervalo por la mitad y vuelve a calcular la integral, para luego compararla con la anterior, de modo de que si la diferencia es menor que la tolerancia se devuelve el valor, en el caso contrario se sigue en el ciclo.
La tolerancia se define para la integral $\frac{c^2}{2\pi h}\left(\frac{h}{k_B T}\right)^4 \int_0^\infty B_\lambda(T)d\lambda$, a la que luego se le agregan las constantes respectivas
Para optimizar este algoritmo para cada nueva iteración se calcula solamente aquellos valores de $f(x)$ que previamente no se tenían, en este caso todos los $f(a+i\frac{b-a}{n})$ donde i es impar. Un modo fácil de lograr esto es guardar el valor de la integral dividido por $\frac{(b-a)}{2n}$, así los nuevos valores de f se suman directamente.
Usando el mismo razonamiento de la pregunta anterior, multiplicamos el valor obtenido de energía por unidad de tiempo por unidad de área por el área de una esfera de radio R para obtener la luminosidad que debe ser igual a la calculada en la pregunta 2, finalmente de esto podemos despejar el radio efectivo del Sol, R.

3.3 resultados

Se obtiene el valor de la integral (con tolerancia 0.01):

$$P = \int_0^\infty B_\lambda(T)d\lambda = 6.32 * 10^{10}$$

Comparando con el valor teórico de la integral $\int_0^\infty \frac{x^3}{e^x-1} = \frac{\pi^4}{15}$ obtenemos un error de:

$$\epsilon = 35923.3303528 erg * s^{-1} * cm^{-2}$$

Obtenemos el radio efectivo del sol con la relación:

$$4\pi(R)^2 P = L = 3.84195 * 10^{33}$$

entonces:

$$R = \frac{L}{4\pi P} = 69552120993.4cm$$

3.4 Conclusiones

Dado el error obtenido se concluye que la tolerancia escogida aproxima correctamente el valor teórico de la integral. Además se observa que con el valor aproximado de la integral se llega a un valor cercano al radio conocido del sol $R = 6,96 * 10^{10}cm$, lo que justifica la tolerancia escogida.

4 Pregunta 4

4.1 Introducción

Se debe recalcular las integrales obtenidas en la pregunta 2 y 3 usando el módulo scipy de python para comparar tanto los valores de estas como los tiempos de ejecución para cada método de integración

4.2 Procedimiento

Para calcular la integral del espectro solar a partir de los datos cargados del archivo "sun_AMO.dat", se utiliza el método trapz(), al que se le entregan los arreglos respectivos de longitud de onda y flujo, luego el resultado se transforma a unidades cgs. Para recalcular la integral de la pregunta 2 se utiliza el método integrate.quad(), al que se le entrega la función de planck sin las constantes, luego este resultado se multiplica por las constantes respectivas. Para determinar el error de cada cálculo se resta el valor obtenido por integración numérica programada con el valor entregado por el método de scipy. Finalmente para estimar el tiempo de ejecución de cada método se usa la instrución "timeit" de python.

4.3 resultados

El error obtenido para la integración de la Luminosidad solar por unidad de área es de:

$$\epsilon = 1.86264514923 * 10^{-9} erg * s^{-1} * cm^{-2}$$

El error en la integración de la función de planck resulta:

$$\epsilon_2 = 35923.33 erg * s^{-1} * cm^{-2}$$

Las estimaciones de tiempo de ejecución para cada método se pueden ver en la siguiente tabla:

Función a integrar	Tiempo Método Scipy (ms)	Tiempo Método Trapecios(ms)
Espectro Solar	$47.7 * 10^{-3}$	2.06
Función de Planck	1.77	$526 * 10^{-3}$

Table 1: Tiempos estimados de ejecución.

4.4 Conclusiones

Para el cálculo de la integral del espectro solar, se ve de la tabla 1 que el método de scipy demora dos ordenes de magnitud menos que el método de los trapecios programado, esto se puede atribuir a que el método programado calcula la integral para cada par consecutivo de valores, es decir, se utiliza el mismo salto en x para toda la función cuando se podría adaptar con un salto más corto para pendientes pronunciadas y más largo para pendientes cercanas a 0.

Para integrar la función de planck, se observa que, esta vez, el método programado demora un orden de magnitud menos que el de scipy, lo que está directamente relacionado con la tolerancia escogida, obtenemos un algoritmo más rápido a costa de hacerlo menos preciso.