Métodos numéricos para la ciencia y la ingeniería
Tarea 1

Jean Paul Martel

24 de septiembre 2015

1 Pregunta 1

1.1 Introducción

Se desea graficar el espectro del Sol (flujo vs. longitud de onda) a partir de los datos obtenidos del archivo *sun_AMO.dat* .Este flujo corresponde al medido justo afuera de nuestra atmósfera en unidades de energía por unidad de tiempo por unidad de área por unidad de longitud de onda. Se usará sistema CGS para las unidades de flujo y μm para la longitud de onda.

1.2 Procedimiento

A través de la rutina *numpy.loadtxt*() se lee el archivo y los datos quedan en un arreglo bidimensional con longitud de onda en la primera columna y flujo en la segunda.

Después se obtienen arreglos independientes para cada variable *londa* y $flujo$ y las unidades son convertidas a los sistemas requeridos (longitud de onda en μm y flujo en $ergs\ s^{-1}\ \mathrm{cm}^2 \mu m^{-1}$). Finalmente se procede a graficar usando la rutina $plt.plot(londa, flujo)$. Este gráfico es presentado en la siguiente sección. El código para esta pregunta se puede encontrar en $pregunta1.py$.

1.3 Resultados

El gráfico obtenido para el espectro del sol puede verse a continuación:

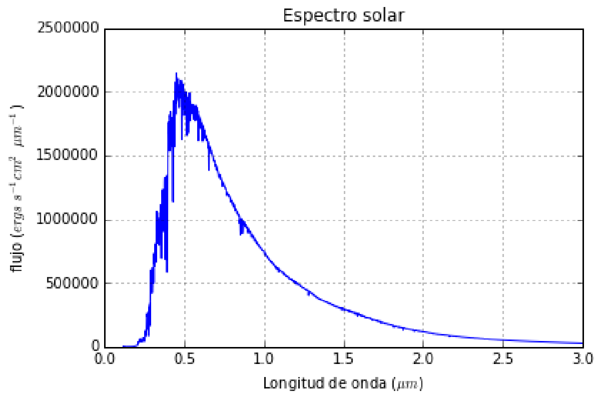

Figure 1: Espectro del Sol. Flujo vs. Lonigtud de onda.

1.4 Conclusiones

Como es de esperar el flujo, debido a que la radiación del Sol se asemeja mucho a la de un cuerpo negro, se tiene un peak cercano a los $0.5\mu m$ y luego decrece rápidamente cuando la longitud de onda tiende a infinito.

2 Pregunta 2

2.1 Introducción

Se busca integrar numéricamente el espectro solar con un método adecuado. Es decir integrar el flujo respecto a la longitud de onda. A partir de esto, calcular la luminosidad del sol.

2.2 Procedimiento

Se utiliza la regla del trapecio compuesta como método de integración, el cual consiste en la siguiente fórmula:

$$\int_a^b f(x)dx = h\left[\frac{f(a)+f(b)}{2} + \sum_{k=1}^{n-1} f(x_k)\right]$$

Con $h = \frac{a+b}{n}$ y n el números de divisiones en que se divide el intervalo de integración. En nuestro caso cada x_k corresponde a un valor para la longitud de onda y cada $f(x_k)$ es el valor del flujo para esa longitud de onda (todos los datos escritos en las unidades originales del archivo). Los datos se obtienen de la misma manera que la pregunta 1 desde el archivo *sun_AMO.dat*, obteniendo un arreglo para cada parámetro.

Se definió la función *trapecio*() que va adquiriendo cada dato y realiza el calculo anterior.

El código de este algoritmo se puede encontrar en *pregunta*2.*py*.

El valor de esta integral corresponde a un "flujo" recibido en unidades de Luminosidad por unidad de área, y convertido a sistema CGS. Si se llama F^* a este valor. Se relaciona entonces con la luminosidad solar de esta forma:

$$L_{sol} = 4\pi r^2 F^*$$

Donde r es la distancia entre la tierra al sol ($1UA$). $4\pi r^2$ corresponde al área de una esfera de radio r.

2.3 Resultados

El valor obtenido con la integración fue (en unidades de $ergs\ s^{-1}$ cm^2):

$$\int_a^b f(\lambda)d\lambda = F^* = 1366090.79684$$

En este caso $f(\lambda)$ es el flujo (energía por unidad de tiempo de área y longitud de onda). El parámetro λ representa la longitud de onda. Entonces, de este valor F^* calculamos la luminosidad del Sol. Conociendo que la unidad astronómica (r) vale 1.49598 X 10^{13} cm

$$L_{sol} = 4\pi r^2 F^* = 3.842\ X\ 10^{33} ergs\ s^{-1}$$

2.4 Conclusiones

El valor aceptado para la Luminosidad solar es 3.846 X 10^{13} $ergs\ s^{-1}$ Con el valor obtenido por integración numérica realizada con el método del trapezoide, y con los datos de *sun_AMO.dat* se estimó una luminosidad que sólo tiene un error porcentual de 0.104

Por tanto se puede concluir que este método funciona como una buena aproximación para este caso.

3 Pregunta 3

3.1 Introducción

La radiacion de un cuerpo negro en unidades de energia por unidad de tiempo por unidad de área por unidad de longitud de onda esta dada por la funcion de Planck:

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

donde h es la constante de Planck, c es la velocidad de la luz en el vacio, k_B es la constante de Bolzmann, T es la temperatura del cuerpo negro y λ es la longitud de onda. Se pretende integrar numericamente la funcion de Planck para estimar la energia total por unidad de area emitida por un cuerpo negro con la temperatura efectiva del sol. Además estimar el radio efectivo del sol.

3.2 Procedimiento

Se quiere calcular

$$\int_0^\infty B_\lambda(T) d\lambda = \int_0^\infty \frac{2\pi hc^2/\lambda^5}{e^{hc/\lambda k_B T} - 1}) d\lambda$$

Pero haciendo el cambio de variable $x = \frac{hc}{\lambda K_B T}$, resulta

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

Ya que la integral presenta un límite superior ∞ se requiere normalizar la función. Utilizando el cambio de variable y=arctan(x) se obtiene:

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

Al igual que antes, se utiliza la regla del trapecio compuesta. A recordar:

$$\int_a^b f(x) dx = h \left[\frac{f(a) + f(b)}{2} + \sum_{k=1}^{n-1} f(x_k)\right]$$

Notar que el límite de la función que se integra cuando y tiende a 0 es 6. Este valor puede encontrarse, por ejemplo, con l'hopital. Se quiere calcular la integral con una tolerancia elegida (0.001), es decir, controlar cuanto distará el resultado numérico del analítico. Para esto se comienza con integrar con un método de trapezoide simple (1 intervalo); si este resultado no cumple con la tolerancia deseada se aumenta al doble la cantidad de intervalos. Se itera de esta manera hasta lograr la tolerancia deseada. Los detalles del código se encuentran en *pregunta3.py*

Después de la integración se calcula P utilizando las constantes en sistema CGS, y considerando la temperatura efectiva del sol como:

$$T_{eff} = 5776K$$

Luego de obtener P al integrar la función de Plank, se tiene que este cumple la siguiente relación:

$$L_{sol} = 4\pi R_{eff}^2 P$$

con R el radio efectivo del Sol, que es entonces;

$$R = \sqrt{\left(\frac{L_{sol}}{4\pi P}\right)}$$

3.3 Resultados

El valor analítico para

$$\int_0^\infty \frac{x^3}{e^x - 1} = \frac{\pi^4}{15} = 6.493939$$

Mientras que el obtenido es 6.493975 El valor calculado para P es:

$$P = \frac{2\pi h}{c^2} \left(\frac{k_B T}{h}\right)^4 \int_0^\infty \frac{x^3}{e^x - 1} = 63113569289.5$$

en unidades de ergs s^{-1} cm^2 Y se estima el radio efectivo del sol (en cm):

$$R = \sqrt{\left(\frac{L_{sol}}{4\pi P}\right)} = 69599192250.9$$

3.4 Conclusiones

En cuanto al valor obtenido para la integral P debe tenerse un error, el cual estaba ya determinado por la tolerancia escogida para este cálculo. Este resultado aproximado llevó a un valor para el radio efectivo (si el Sol radiara como cuerpo negro) que presenta un error porcentual de 0.05% respecto al encontrado el encontrado en bibliografía(69634200000 cm). Los cálculos son correctos, pues lograron una buena aproximación.

Se aprecia que el método del trapezoide sigue funcionando bien para una integral de límite infinito, si esta se normaliza adecuadamente.

4 Pregunta 4

4.1 Introducción

Ahora se busca re-calcular las integrales de las preguntas 1 y 2 utilizando los métodos predefinidos de python: scipy.integrate.trapz y scipy.integrate.quad. Finalmente comparar los valores obtenidos y la velocidad de ejecución de los algoritmos.

4.2 Procedimiento

Para re-calcular la integral del flujo en el espectro solar en la pregunta 2, se cargan nuevamente los datos de *sun_AMO.dat*. Luego a partir del módulo *scipy* se ejecuta la función *scipy.integrate.trapz* sobre estos datos. Para el caso de la función de Plank de la pregunta 3, se usa *scipy.integrate.quad* en cuyo argumento va la función en la integral ya normlizada P. Se utiliza el comando *timeit* de la función *magic* sobre cada método de integración y se obtienen así los tiempos de ejecución de cada uno. Se encuentran los detalles del algoritmo en *pregunta4.py*.

4.3 Resultados

Los valores calculados y sus respectivos tiempos para la integral de la pregunta 2 (espectro del Sol) se pueden apreciar en la siguiente tabla:

Parámetro calculado	Método Scipy	Algoritmo propio (trapezios)
Valor integral [$ergs\ s^{-1}$]	1366090.79684	1366090.79684
Tiempo [s]	0.000058.3	0.00356

Tabla 1: Comparación valores y tiempo de ejecución para integral del espectro del Sol.

Para la integral de la pregunta 3 (función de Plank) los valores calculados y tiempos de ejecución se encuentran en la siguiente tabla:

Parámetro calculado	Método Scipy	Algoritmo propio (trapezoides)
Valor integral [$ergs\ s^{-1}$]	63113219867.8	63113569289.5
Tiempo [s]	0.00265	1.58

Tabla 1: Comparación valores y tiempo de ejecución para integral de la función de Plank.

4.4 Conclusiones

Comparando los resultados para la integral de la pregunta 2 (espectro solar) el valor obtenido es el mismo, pero en cambio, el algoritmo creado aquí demora más de 10 veces que la función predefinida de scimpy. Una razón de la mala eficiencia podría ser que en cada iteración se calcula un valor que ya fue calculado en la iteración anterior,mejoraría creando un método que evite cálculos innecesarios. Para la integral de la función de Plank en la pregunta 3, con la función del módulo scipy se obtiene un valor más exacto que con el algoritmo creado, lo que es esperable pues se definió para este último una tolerancia arbitraria. La función de scipy es también mucho más eficiente demorando en ejecutar del orden de milisegundos, mientras que el calculado con la regla de trapecio tarda más de 1 segundo. Las diferencias en este cálculo deben ser al menos por: 1−los cálculos innecesarios que se hacen en el algoritmo propio. 2−el método de cuadratura que utiliza scipy es más preciso que el de trapezoides compuesto.