Métodos Numéricos para la Ciencia e Ingeniería
Informe Tarea 1

Benjamín Guerrero
24 de Septiembre, 2015

1. Pregunta 1

1.1. Introducción

Se busca graficar los datos que se encuentran el archivo sun_AM0.dat, que se incluye en el repositorio. Al abrir este archivo en Microsoft Word (o un programa equivalente), se ve que este archivo contiene lo siguiente:

```
# Wavelength (nm) Power (W*m-2*nm-1)
# Source:
http://www.pveducation.org/pvcdrom/appendices/standard-solar-
spectra
119.5   6.190E-05
120.5   5.614E-04
121.5   4.901E-03
122.5   1.184E-03
123.5   4.770E-05
```

…

La primera oración dice que la primera columna de datos muestra la longitud de onda en nm (10^{-9}m) y la segunda columna muestra el espectro del sol, que se mide en $\frac{W}{m^2 * nm}$. La segunda oración muestra la página web de la cual se sacaron los datos.

Finalmente, vienen las columnas de datos. Se recuerda que `E-x` representa 10^{-x}. La cantidad de datos es de 1697 por columna.

El gráfico pedido debe estar en convención astronómica, es decir, el espectro tiene que estar en el sistema CGS (centímetro-gramo-segundo), y la longitud de onda debe estar en Angstroms (10^{-10}m) o en micrómetros. En este caso, se usarán Angstroms.

1.2. Procedimiento

Se crea en Python un archivo llamado lectura.py, que se va a usar para graficar los datos. Se importan las librerías *numpy* y *matplotlib.pyplot*, y luego se usa el siguiente comando: *x,y = numpy.loadtxt('sun_AM0.dat', unpack=True)*. Esto abre el archivo y define *x* e *y* como '*numpy.ndarray*', que contienen los datos de las columnas 1 y 2, respectivamente. Luego, se convierten a la convención astronómica.

Finalmente, usando funciones de *matplotlib.pyplot*, se construye el gráfico. Dada la magnitud de los datos, se usa escala logarítmica para ambos ejes.

1.3. Resultados

Una vez hecho esto, el programa retorna el siguiente gráfico:

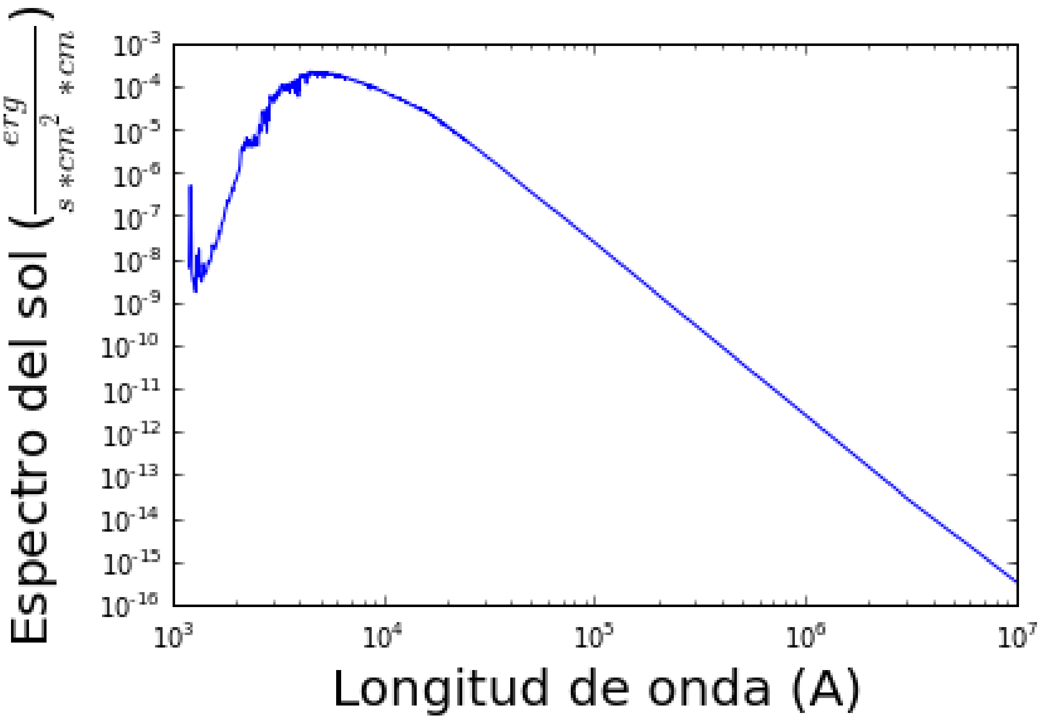

Figura 1: Gráfico de espectro v/s longitud de onda en convención astronómica.

En la figura 1, el espectro se mide en $\frac{erg}{s*cm^2*cm}$. Un $\frac{erg}{s*cm^2*cm}$ equivale a $10^4 \frac{W}{m^2*nm}$.

2. Pregunta 2

2.1. Introducción

Ahora, se busca obtener la luminosidad del Sol integrando el espectro por la longitud de onda (ambos dados por el archivo sun_AM0.dat), usando un algoritmo personal.

2.2. Procedimiento

Aquí, se va a usar el método trapezoidal en cada intervalo para aproximar la integral. Este método se demostró en cátedra y es el siguiente:

$$\int_a^b f(x)dx \approx (b-a)\frac{f(a)+f(b)}{2} \quad (1)$$

Se crea un archivo de nombre integralespectro.py, que debería retornar la integral pedida. Se extraen los datos del archivo usando *numpy.loadtxt*, tal como en la parte 1. (*x*=longitudes de onda, *y*=espectro)

Ahora, en (1), hacemos cambio de variables: x=λ, f(x)=espectro, a=valor de λ, b=siguiente valor de λ.

Primero, se usa *x.size* para saber cuántos datos hay en el array *x* (1697). Dada la simetría, *y* tiene la misma cantidad de datos. Luego, $\forall i \in (0, x.size - 1)$ se define *a=x.item(i),b=x.item(i+1), f(a)=y.item(i),* y *f(b)=y.item(i+1).*

Finalmente, insertamos esto en (1), y la suma de los resultados será la aproximación de la integral.

La sumatoria es igual a usar el método compuesto, también visto en clases:

$$\int_a^b f(x)dx \sim \frac{\Delta x}{2}\left[f(a) + 2 * \sum_{i=1}^{n-1} f(x_i) + f(b)\right],$$

$$x_i = a + i * \Delta x$$

2.3. Resultados

La luminosidad resultante es de 1366.09079684 W.

3. Pregunta 3

3.1. Introducción

Ahora, se debe obtener la energía total por unidad de área integrando la función de Planck, que se define como:

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

En donde h es la constante de Planck, c es la velocidad de la luz en el vacío, k_B es la constante de Bolzmann, T es la temperatura del cuerpo negro y λ es la longitud de onda. En este caso, T = 5778 K. Aquí, $\lambda \in [0, \infty]$

Luego, con los resultados de las integrales, se debe calcular el radio del Sol.

3.2. Procedimiento

Se usará el método del punto medio, que también se demostró en cátedra. Para esto, se crea un archivo de nombre integralplanck.py.

Primero, se importa el submódulo *astropy.constants*, que incluye las constantes que la función usa.

Haciendo el cambio de variable $x = \frac{hc}{\lambda k_B T}$, $dx = -\frac{hc}{\lambda^2 k_B T} d\lambda$ se llega a la siguiente integral:

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

Ahora, se hace el cambio de variable $y = \arctan(x)$, $dy = \frac{1}{1+x^2} dx$. Así, la integral (sin las constantes) queda:

$$\int_0^{\frac{\pi}{2}} \frac{\tan(y)^3 * (1 + \tan(y)^2)}{e^{\tan(y)} - 1} dy \; (2)$$

Ahora, podemos usar el método de los puntos medios con esta integral (ya que la función, por complicada que parezca, es continua por álgebra y composición de funciones continuas). Este método es el siguiente:

$$\int_a^b f(x)dx \sim (b-a)\, f\left(\frac{a+b}{2}\right)$$

Para esto, se divide la recta (0, π/2) en 8 segmentos de medida π/16, y se realiza el método para cada segmento. Luego, se suman los resultados, y el resultado es la integral buscada.

Una vez hecho esto, se divide el resultado de la parte anterior por el resultado recién calculado. Esto dará el área del Sol (asumiendo tiempo constante). Luego, se asume que el Sol es una esfera, y se usa la ecuación $A = 4\pi r^2$ para calcular el radio.

3.3. Resultados

El resultado de la integral (2) es de 5.68126706749. Al multiplicarlo por las constantes, se obtiene 55291544.6312 J/m^2. El radio del Sol se estima en 178.293093038 metros(????)

4. Pregunta 4

4.1. Introducción

Se pide calcular las integrales anteriores usando el módulo *scipy.integrate.trapz* y *scipy.integrate.quad*. Luego, se pide comparar los resultados de esto con los resultados de los algoritmos anteriores, y la velocidad de ejecución.

Usando el comando help() revela que *scipy.integrate.trapz* usa el método trapezoidal para integrar, y *scipy.integrate.quad* usa un método de la librería de Fortran QUADPACK.

4.2. Procedimiento

Se crea un archivo llamado integralscipy.py, y se importan los módulos necesarios (numpy, astropy.constant,etc).

Se usa *scipy.integrate.trapz* para recalcular la primera integral (P2), y *scipy.integrate.quad* para calcular la segunda (P3), debido a que se tiene los resultados de la función en P2, y se tiene la función en P3 (esto es basado en las descripciones de los comandos)

Después, se usa *ipython magic %timeit* en la consola de ipython para calcular el tiempo de ejecución de los programas.

4.3. Resultados

Al ejecutar la primera integral con *scipy.integrate.trapz*, da 1366.09079684 W. Al ejecutar la segunda integral con *scipy.integrate.quad*, esta da 6.49393940226683.

Usando *%timeit*, se nota que los algoritmos creados para P2 y P3 demoran 28.7 y 28 ns ($10^{-9}s$) por loop en ejecutarse, respectivamente. Mientras tanto, *scipy.integrate.trapz* demora 97.1 ns por loop, y *scipy.integrate.quad* demora 98.6 ns por loop.

5. Conclusiones

El gráfico muestra que el espectro alcanza su mayor valor a un poco menos de 10^4Angstrom. A partir de allí, decrece de forma exponencial mientras aumenta la longitud de onda.

Ahora, el resultado de la primera integral es 1366 (tanto con el algoritmo como con *scipy.integrate.trapz*), que es el valor de la constante solar (K). (Sin embargo, la cte. solar se mide en W/m^2 , no en W).

En la segunda integral, se muestra que, si bien no es exacto (ya que la función tan(π/2) tiende a ∞ y existe un error asociado), el método del punto medio realiza una aproximación más o menos buena (diferencia menor a 1). Seguramente puede ser mejor si se aumenta el número de segmentos en los que se divide la recta [0,π/2]. Sin embargo, *scipy.integrate.quad* da una aproximación casi exacta del resultado analítico $\frac{\pi^4}{15}$.

Ahora, al multiplicar este resultado con las constantes, se llega a 55291544.6312 $\frac{J}{m^2}$, lo que es mucho menor que la luminosidad del Sol real. En efecto, esto da un radio solar de 178 m, cuando debiera ser 696.300.000 m. Esto puede deberse a errores en el enunciado, o a errores en las unidades de medida aplicadas, o a un error en la forma de calcular el radio del Sol.

También se aprecia que los algoritmos personales demoran menos en ejecutarse que los algoritmos de librería. Esto se explica porque los algoritmos creados son mucho más simples que los importados.