FI3104B - Metodos Numericos Para la Ciencia y la Ingenieria
Tarea 1

Camila Sandivari Gana
Profesor: Valentino Gonzalez
Auxiliar: felipe Pesce

September 24, 2015

1 Introduction

El objetivo de este reporte es mostrar como usando métodos númericos de integración en el programa python se resuelven cálculos en torno al estudio de ciertas cantidades de interés de un cuerpo negro, en este caso el sol.
Se pueden identificar tres etapas del problema, primero se analizan los datos de espectro del Sol, medido justo afuera de nuestra atmosfera, en unidades de energia por unidad de tiempo por unidad de area por unidad de longitud de onda; segundo se integra el espectro para calcular la luminosidad o constante solar; tercero se integra la ecuación 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, kB es la constante de Bolzmann, T es la temperatura del cuerpo negro y Λ es la longitud de onda, para obtener la temperatura efectiva y deducir el radio del sol.

2 Procedimiento

2.1 Parte 1

Se importa la librería pylab, se cargan los datos un archivo entregado que contiene el espectro del sol para nombrarlos como $x : longitudesdeonda$ e $y : flujo$ y graficar y según x.

2.2 Parte 2

Se integran los datos cargados y graficados en la parte 1 utilizando el método numérico de trapecios

$$\int_{x_o}^{x_o+\Delta x} f(x)dx = (\Delta x)/2 * [(f(x_0) + f(x_0 + \Delta x) + \sigma(\Delta x^3)] \qquad (1)$$

con $y = f(x), x = x$ según los datos obtenidos en 1 usando el código adjunto y sumando dentro de un ciclo $While$. Se comparan los resultados con la funcion predefinida en la librería *scipy* con la funcion *scipy.integrate.trapz*
El resultado corresponde a la constante solar K y la luminosidad solar corresponde a $L_s = K * S$ con $S = 2,81 * 10^2 3m^2$ la superficie esférica.

2.3 Parte 3

Se resuelve la función de planck en su versión integral

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

; con un cambio de variable en la integral $x = \tan(u)$ se integra la siguiente expresión

$$\int_0^{\pi/2} \frac{\tan(u)^3 + \tan(u)^5}{\exp(\tan(u) - 1)} \tag{2}$$

con el método de simpson

$$\int_{x_o}^{x_o+\Delta x} f(x)dx = (\Delta x)/3*[(f(x_0)+4*f(x_0+\Delta x)+f(x_0+2*\Delta x)+\sigma(\Delta x^3)] \tag{3}$$

con la función (2) usando el código adjunto y sumando dentro de un ciclo $While$.Se comparan los resultados con la funcion predefinida en la librería *scipy* con la funcion *scipy.integrate.quad*

3 Resultados

3.1 Parte 1

Se obtiene el siguinte gráfico para el espectro del sol

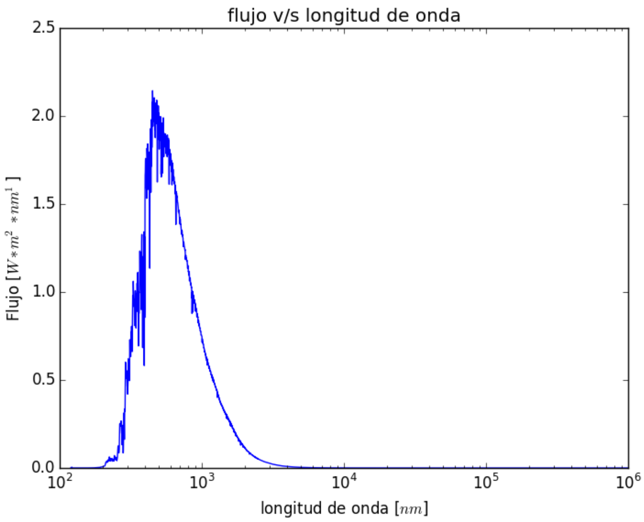

Figure 1: espectro del sol

3.2 Parte 2

Se obtiene el valor de la constante solar y a partir de eso la luminosidad

- integracion por trapecio = $1366.09079684W/m^2$
- integracion con funcion de $scipy.integrate$ = $1366.09079684W/m^2$
- luminosidad = $3.83871513912e + 26kW$

3.3 Parte 3

Se obtiene el valor de la integral que corresponde al flujo energético y el radio solar

- solución planck con método simpson = $63200679.7061W$
- solución planck con función de $scipy.integrate$ = $6.32006797e + 07W$
- solucion analitica planck = $63200679.7124W$
- radio solar = $695511508.079K^2m^2s(1/2)/J(1/2)$

3.4 Parte 4

Tiempos obtenidos con "timeit"

- Tiempo parte 2 algoritmo propio= $47.2nsporloop$
- Tiempo parte 2 scipy.integrate.trapz= $39.3nsporloop$
- Tiempo parte 3 algoritmo propio= $47.3nsporloop$
- Tiempo parte 3 scipy.integrate.quad = $54.4nsporloop$

4 Conclusiones

Los métodos numéricos de integración implementados resuelven la integral tan precisamente como las funciones existentes en python y en cuanto al tiempo que se demoran en ejecutar las integrales es mayor en el caso de la parte 2 y menor en la parte 3, esta diferencia puede deberse a la eficiencia del código en cuanto a los métodos usados para los ciclos.