

Métodos numéricos para la ciencia e ingeniería:

Tarea 1*

*(no logré realizar el punto 3)

Borja Mir

24 de Septiembre del 2015

1. Introducción

Se busca estudiar el espectro del Sol justo afuera de nuestra atmosfera, para esto se analiza el archivo `sun_AM0.dat`, el cual contiene la energía por unidad de tiempo por unidad de área para distintas longitudes de onda de este.

Primero se creó un gráfico de Flujo vs Longitud de Onda a partir de los datos del archivo para crear un algoritmo que integre el espectro en longitud de onda y así poder calcular la luminosidad total del Sol (energía por unidad de tiempo total).

Y finalmente comparar los resultados del algoritmo escrito por uno con los obtenidos por el módulo *scipy* tanto en datos como en velocidad de ejecución.

2. Procedimiento

Para crear el gráfico se procedió a importar los datos a python utilizando la rutina *numpy.genfromtxt* del módulo *numpy* la cual genera arreglos, en este caso del tipo *float*, a partir de un documento de texto. Los datos fueron transformados a la convención astronómica, es decir a CGS en el caso del flujo y a ångström para las longitudes de onda. Luego fue plotado a través del módulo *matplotlib* (figura 1) y se le aplicó un \log_{10} a ambos ejes para una mejor apreciación.

Antes de escribir el algoritmo para calcular la luminosidad se tuvo que elegir el método numérico de integración que más se adecuara a la función, luego de un análisis de los datos el cual demostró un Δx muy poco variable se decidió proseguir utilizando el método trapezoidal, el cual da el siguiente resultado.

$$\int_{x_0}^{x_0+\Delta x} f(x) = \frac{\Delta x}{2} [f(x_0) + f(x_0 + \Delta x)] + \mathcal{O}(\Delta x^3)$$

Para replicar este método se definió la siguiente función en python:

```
26 def integraltrap(x,y):
27     """
28     Recibe preimagen e imagen de funcion y
29     encuentra integral utilizando el metodo del trapezio
30     """
31     contador=0
32     integral=0
33     trapecio=0
34     while contador<len(x)-1:
35         trapecio=(y[contador]+y[contador+1])*(x[contador+1]-x[contador])/2
36         integral=integral+trapecio
37         contador=contador+1
38     return integral
```

Y por el otro lado se realizó la misma integral utilizando la rutina *scipy.integrate.trapz()* del modulo *scipy*. Los resultados se pueden observar en la Tabla 1.

3. Resultados

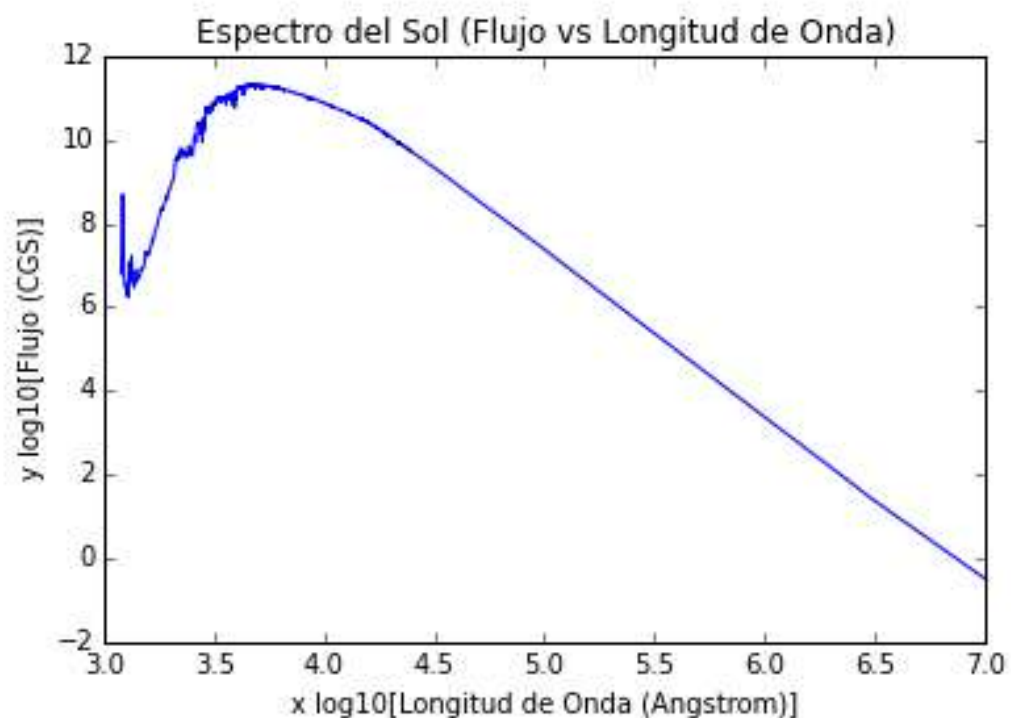


Figura 1

	Algoritmo	Scipy
Espectro integrado en longitud de onda	1.366 e15	1.366 e15
Constante solar	1366.1	1366.1
Luminosidad total	3.838 e26	3.838 e26
Tiempo	No se logró correr %timeit	No se logró correr %timeit

Tabla 1

4. Conclusiones

Analizando los resultados podemos notar que el algoritmo creado fue muy preciso a la hora de realizar la integración si es que lo comparamos con los obtenidos a través del módulo *scipy*, lástima que no se haya logrado estimar la velocidad de ejecución para ambos casos ya que a partir de ello se pudo haber obtenido conclusiones bastante interesantes.