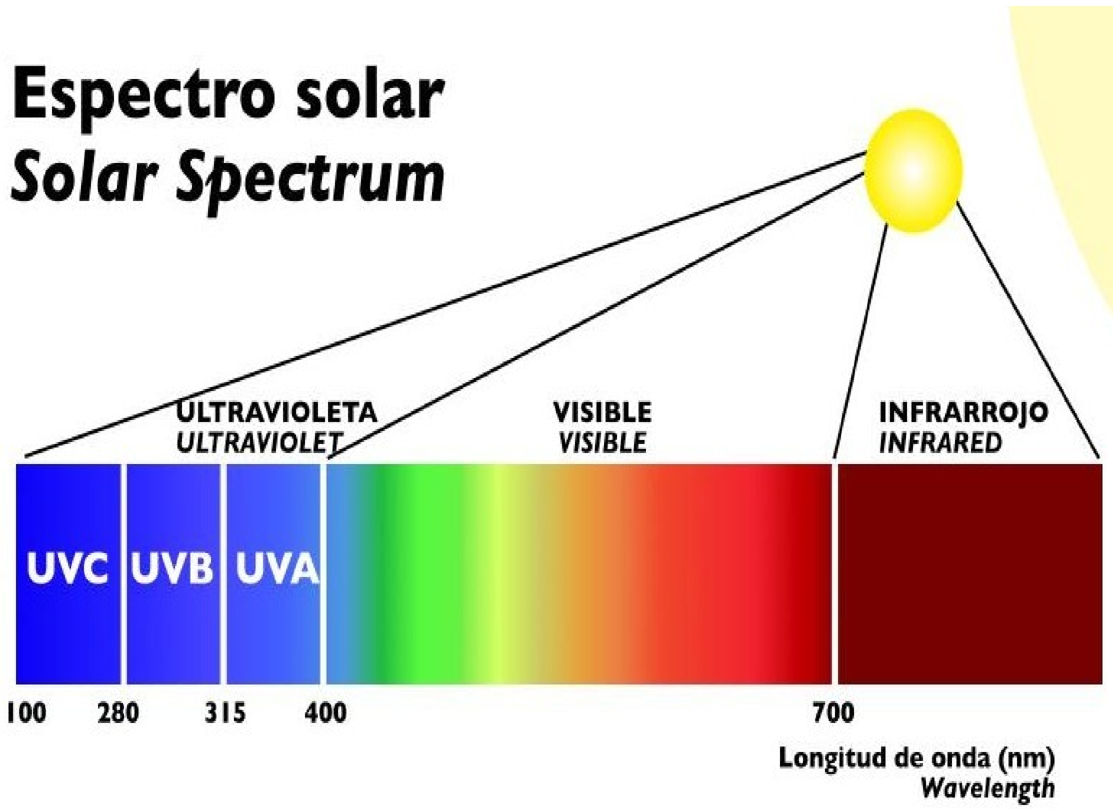
INFORME
TAREA #1
MÉTODOS NUMÉRICOS PARA LA CIENCIA E INGENIERÍA



Curso: Métodos numéricos para la Ciencia e Ingeniería

Nombre: Benjamín Matías Venegas Ríos

Profesor: Valentino González

Auxiliar: Felipe Pesce

Fecha: 24/09/2015

PREGUNTA 1

1.1. INTRODUCCIÓN

Se buscaba graficar el flujo solar vs la longitud de onda dado el archivo sun_AM0.dat. Era importante en esta parte pasar el flujo a unidades CGS, mientras que la longitud de onda a Angstrom o a micrón (elección de cada alumno). Además, que al momento de plotear los datos, se rotularan ambos ejes y pusiera un título apropiado.

1.2. PROCEDIMIENTO

Para resolver este problema, se procedió a importar desde el editor de código el módulo numpy (import numpy as np), librería que contiene una rutina llamada numpy.loadtxt, con el fin de poder leer el archivo y sacar la información de cada columna (datos de longitud de onda y flujo). Por lo tanto, se creó una variable denominada data,(leyendo el archivo antes mencionado con loadtxt) la cual guardaría la información de todo el texto. Luego,se crearon 2 variables (londa y flujo), las cuales conservarían la primera y segunda columna ,evitando el espacio vacío que había entre estos datos. Para eso,al arreglo data que se había creado, “lo partimos quitándole los espacios vacíos” con el típico comando de python “data[:,0] y data[:,1]” ,donde cada uno leerá la primera y la segunda columna, respectivamente. Cabe destacar que, como la pregunta especificaba que el flujo estuviese en unidades CGS (el flujo venía en unidades internacionales) y la longitud de onda en una unidad a elección(se eligió micron), se tuvo que multiplicar cada arreglo por un factor de conversión. En el caso del arreglo londa, el factor fue de 0.001, mientras que para el arreglo del flujo fue de 1000.

Con los arreglos listos y en las unidades pedidas, siguió la parte gráfica, para lo cual antes de partir, debimos nuevamente importar una herramienta, en este caso, matplotlib (import matplotlib.pyplot as plt), que nos permitiría gráficar los datos obtenidos del archivo sun_AM0.dat. Entonces, luego de haber hecho esto, simplemente, se plotearon los arreglos y se rotularon los ejes respectivos, además de haber añadido un título y una grilla, que ordenaría un poco más el gráfico. La figura se guardó como png (“savefig”) y también como muestra directa(“show”). No está de mas decir que para estudiar el gráfico y el máximo de flujo dado una longitud de onda particular, hay que hacerle zoom.

1.3. RESULTADOS

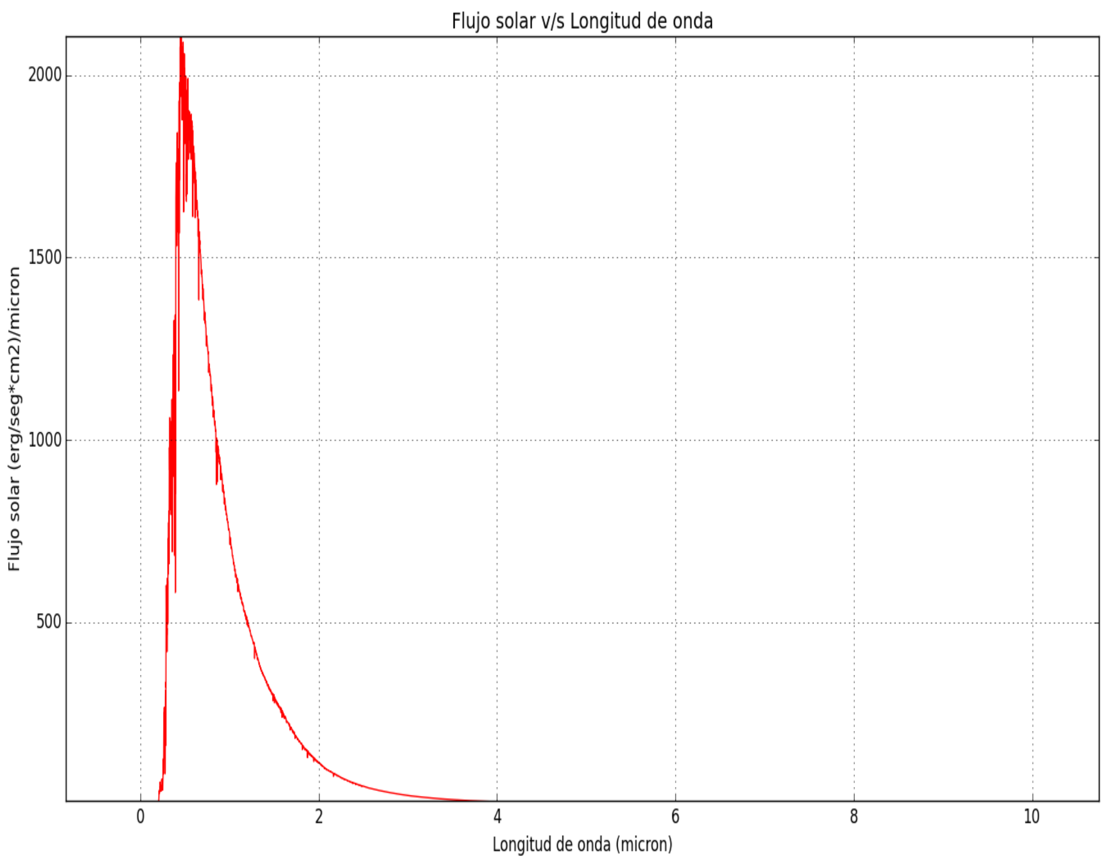


figura1.1 Flujo solar vs Longitud de onda para la pregunta 1.

PREGUNTA 2

2.1. INTRODUCCIÓN

Para la pregunta 2 se requería elegir un método apropiado para integrar el espectro en longitud de onda(pregunta 1), y posteriormente, obtener la luminosidad solar(energía por unidad de tiempo) en base al resultado obtenido por la integral numérica. En este caso, se decidió calcular la integral numérica por medio del método del trapecio(la versión más simple, independiente de las divisiones que se hagan del intervalo) y ocupar geometría para calcular la luminosidad solar.

2.2. PROCEDIMIENTO

Para resolver esta parte mediante el Método del Trapecio se ocupó un while que recorriera cada uno de los datos de los arreglos (flujo y londa) ,partiendo con la integral numérica en valor nulo afuera del ciclo , y así cada vez que entrara a este, se fueran sumando las áreas hasta llegar al último dato (1696). Cabe destacar,que al igual que la integral definida como nula afuera del ciclo, se definió un índice i, que también partiría de cero y recorrería todos los datos hasta llegar al último del archivo sun_AM0.dat.

La integral numérica definida por el Método del Trapecio era:

integral2+=(deltalonda/2)*(fa+fb)   # Sumando areas

donde   deltalonda=(b-a); fa= flujo i ; fb= flujo i+1

Para calcular la luminosidad solar, simplemente se requirió ocupar que es proporcional al área de una esfera con radio 1 u.a (distancia promedio sol-tierra,transformada en metros) . Finalmente, se calculó con la siguiente fórmula:

lumsolar= 4*(np.pi)*(au)**2 *(integral2)

donde au= 1 unidad astronómica [metros]

2.3. RESULTADOS

Para la integral numérica se obtuvo:

integral2=1366.09079684 [Watts/m2]

Para la luminosidad solar se obtuvo:

lumsolar=3.84184866671*(10 **26) [Watts]

PREGUNTA 3

3.1. INTRODUCCIÓN

Para esta pregunta se buscaba integrar numéricamente la función de planck (ecuación de más abajo) eligiendo un método apropiado, para estimar la energía total por unidad de área por unidad de tiempo emitida por un cuerpo negro con la temperatura efectiva del sol (Teff=5778 [°K]) y comparar con el resultado analítico. Además ,se pedía comparar la integral obtenido con la energía total por unidad de tiempo calculada en la pregunta 2 ,para así estimar el radio efectivo del sol.

La función de planck a integrar numéricamente está dada por:

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

Donde h = constante de planck
c = velocidad de la luz en el vacío
T = temperatura del cuerpo negro (temperatura efectiva del Sol en este caso)
λ = longitud de onda
k= constante de boltzmann

Y la integral de esta ecuación está dada por:

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

3.2. PROCEDIMIENTO

Para realizar la integración numérica de la función de planck, en primer lugar, había que calcular la integral impropia que va de cero a infinito para luego, simplemente, multiplicar por un factor determinado sólo por constantes. No obstante, dado que numéricamente no se puede trabajar con infinito, se realizó un cambio de variable y=arctan(x) para así convertir la integral impropia en una integral definida con límites 0 y π/2. La integral entre cero e infinito se transformó en:

$$\int_0^{\frac{\pi}{2}} \frac{\sec^2(y)\tan^3(y)}{-1+e^{\tan(y)}}\, dy = 6.49394$$

Donde, el valor que aparece, es lo esperado analíticamente.

Entonces, en el código se implementó un arreglo que iría desde un valor muy cercano a 0 y a π/2. Esto se hizo de esta manera, debido a que la función presentaba singularidades en esos puntos. Sin embargo, el área bajo la función que se "restó" es despreciable, como se comprobará más adelante al ver los resultados y comparar con lo analítico. También, el arreglo se creó con un cierto tamaño que pudiese ser modificado y así ver la tolerancia de la integral. Luego de ser escrito el arreglo con los límites especificados, se añadió la función a integrar (arriba).

Después, siguió la parte de integrar la función ocupando en este caso el Método del Trapecio pero en una versión más trabajada que en la pregunta 2. El resto fue análogo a lo hecho en la pregunta 2, definiendo una integral e índices nulos fuera del ciclo con tal de ir sumando cada vez hasta llegar al final del arreglo mencionado anteriormente. En esta parte, la integral que iba sumando áreas fue:

```
integral3+=(deltay/2)*( fnormalizada[1]+ fnormalizada[n-2]+2*fnormalizada[i+1])
```

Donde fnormalizada = la función a integrar (figura arriba)
deltay = (b-a)/n
b = límite superior del arreglo (cercano a π/2)
a = límite inferior del arreglo (cercano a 0)
n = tamaño del arreglo
i = índice que recorre el arreglo de la función

NOTA: Se omite el primer y último término de la función a integrar debido a la misma razón por la que el arreglo tomó valores cercanos a 0 y π/2.

Y además, dado que se obtuvo un valor para la integral, se procedió a calcular la integral de la función de planck, simplemente, multiplicando el factor de constantes por la integral calculada. En esta parte se importó astropy con el submódulo astropy.constants con el fin de tener todas las constantes involucradas en el SI. De esta forma, la integral de la función de planck se calculó como:

Ifplanck= ((2*np.pi*h)/c**2)*((k*T/h)**4)*integral3

Finalmente, ya habiendo calculado la integral de la función de planck, se compararon las "energías" calculadas de 2 y 3 , para así concluir que el radio efectivo solar está vinculado directamente a la razón de éstas "energías". La fórmula para el radio efectivo se determinó como:

radioeff=sqrt((lumsolar/fplanck)/(4*np.pi))

3.3. RESULTADOS

El valor numérico para la integral3 calculada fue de:

6.4940818453

El valor numérico para la radiación de un cuerpo negro con la temperatura efectiva solar (integral de la función de planck) fue de:

63202066.0045 [Watts/m2]

PREGUNTA 4

4.1. INTRODUCCIÓN

En esta pregunta se pedía importar en primer lugar la herramienta matemática scipy en python ,que incluye los métodos scipy.integrate.quad y scipy.integrate.trapz para re-calcular las integrales calculadas en 2 y 3 .Luego,comparar los valores obtenidos y la velocidad de ejecución del algoritmo escrito por el código implementado versus scipy.

4.2. PROCEDIMIENTO

En primer lugar, se re-calculó la integral3, mediante el método de integrate.quad. Para eso se creó una función x3, ocupando lambda, lo que permite definir una función que toma los valores que se le entregen. La función que se introdujo fue la función sin normalizar, es decir:

$$\frac{x^3}{e^x - 1}$$

Luego, solamente se definieron los límites, en este caso entre 0 e infinito, y de esta forma se obtuvo el valor teórico para la integral3 calculada en la pregunta 3 con cierto error asociado.

Además del método de integrate.quad se utilizó el método de integrate.trapz para calcular la misma integral3. Lo único que había que hacer era crear un arreglo partiendo desde valores cercanos a 0 (a) y cercanos a $\pi/2$ (b) para evitar divergencia. La función que se ejecutó en esta parte fue fnormalizada.

Por otro lado, para la integral2 el método usado fue integrate.trapz (sirve para arreglos), en donde tan solo se necesitaban ingresar los arreglos de flujo y londa usados en la pregunta 1 para calcularla.

Ahora, para los tiempos (velocidad de ejecución del programa) se requirió de %timeit escrito en python para que entregara las velocidades correspondientes de cada método numérico o teórico. En este caso se le aplicó %timeit a los 2 métodos para determinar la integral3 además del método numérico, y por otra parte, en el caso de la integral2 se le aplicó %timeit al método numérico y al teórico.

4.3. RESULTADOS

El valor teórico para la integral3 con el método de quad es de :

6.49393940227
con un error asociado de 2.62847002892*(10**-9)

El valor teórico para la integral3 con el método de trapz es de:

6.49389262989

El valor numérico para la integral3 con el método del trapecio versión 2 es de:

6.4940818453

El valor teórico para la integral2 con el método de trapz es de:

1366.09079684 [Watts/m2]

El valor numérico para la integral2 con el método del trapecio versión 1 es de:

1366.09079684 [Watts/m2]

Tiempo para pregunta 3

Tiempo quad3:

37.2 nspor loop

Tiempo trapz3:

37.6 ns por loop

Tiempo numérico3:

38.9 ns por loop

Tiempo para pregunta 2

Tiempo trapz 2:

37.3 ns por loop

Tiempo numérico 2 :

37.9 ns por loop

CONCLUSIÓN

En vista de los resultados obtenidos, se puede deducir que el flujo solar alcanza un máximo a cierta longitud de onda (considerando el espectro justo fuera de la atmósfera terrestre). Si bien no todas las longitudes de onda alcanzan a penetrar a la atmósfera debido a que algunas tienen menor energía que otras, al haber un flujo solar máximo, hace que esas longitudes de onda logren penetrar, por tener mayor energía, y justamente es lo que el ser humano logra ver desde la superficie terrestre. De hecho, vemos el sol de un color anaranjado-amarillo porque son estas ondas las que alcanzan una intensidad Máxima.(el color verde también llega a filtrarse con una gran intensidad)

Por otro lado, en la parte 2 se pudo obtener un valor aproximado de la potencia sobre área usando el método del trapecio versión 1 , la cual es idéntica a la obtenida mediante scipy. Eso habla de que el método numérico logró ajustarse y aproximarse de buena forma a la función en particular. No obstante, si comparamos la velocidad de ejecución de estos métodos podemos ver que el tiempo numérico de lectura es 0.6 décimas de ns mayor al tiempo teórico usando scipy. Esta diferencia se debe a que scipy se ahorra búsqueda cuando ya tiene una "visión " más clara del valor de la integral, mientras que, el método de trapecio numérico recorre todo el intervalo de integración sin saltarse ni uno, por más que la integral ya esté prácticamente definida. En todo caso la diferencia es casi despreciable. El método numérico es bien preciso y rápido.

En el caso de la parte 3, al calcular la integral de la función de planck,dejando de lado el factor de constantes, había un desafío numérico nuevamente, y el método del trapecio versión 2 arrojó un valor que si vemos en la parte 4, es casi igual a lo que calculó scipy ,mediante los métodos teóricos integrate.quad e integrate.trapz. De hecho, la diferencia entre lo numérico y lo teórico fueron de milésimas. Esto nos dice que, sin lugar a dudas, el método numérico del trapecio versión 2 es una buena forma de obtener la integral en cuestión. Una cosa importante es que se puede alterar o refinar el valor de la integral3 con una tolerancia elegida arbitrariamente. En el caso de este trabajo, se decidió tomar 100 valores para los arreglos , y tomar valores de a y b cercanos a los pedidos, ya que la función tenía singularidades en estos puntos. A pesar de esto, como se pudo comprobar a través de los resultados, las aproximaciones estuvieron ajustadas y la integral3 se acercó mucho a lo esperado. Por otra parte, en cuanto a la velocidad de ejecución, nuevamente, el método numérico tardó casi 1 ns más que los métodos teóricos. Esto si bien, habla de que el método numérico es más lento, también hay que considerar lo que se mencionó antes. Scipy trata de calcular lo más rápido posible los valores que se le entregan. Así que ésta diferencia la podemos menospreciar.

Para concluir, los métodos numéricos usados mostraron efectivamente que son una buena aproximación cuando se necesita calcular algún parámetro físico. Es por eso que varias veces son utilizados para calcular lo que a priori se ve complicado.