Métodos Numéricos para la ciencia y la ingenieria

Informe de Tarea 1

Autor: Leonardo Leiva
Profesor: Valentino González C.
Auxiliar: Felipe Pesce
24 de septiembre de 2015

1. Introducción

Durante este informe se mencionarán los pasos realizados para obtener una aproximación del Radio Solar por medio de calcular la luminosidad total del sol y la energía total por unidad de area emitida por un cuerpo negro con la temperatura efectiva del sol.
Para ello se usarán métodos de integración hechos manualmente y luego se compararán los resultados con los métodos que vienen incluidos en las librerías de Anaconda. Junto con esto, se comparará la velocidad que tarda en realizar los procesos.

2. Luminosidad total del Sol

Se tiene un archivo que contiene el espectro del sol, medido justo por afuera de nuestra atmósfera. Se extraen los datos y se ubican en dos listas para hacer más simple su manejo. Mediante la librería Astropy se le asignan las unidades de medida adecuadas: para la primera lista "x", que corresponde a longitud de onda $[m]$ e "y", que corresponde a unidades de Flujo $[W/(m^2nm)]$. Una vez hecho esto, se traspasó a unidades del sistema cgs debido a la convención astronímica.
Una vez estaban los datos ordenados con sus unidades apropiadas, se grafican los datos para observar el comportamiento de las dos variables. El gráfico corresponde a la figura 1. Este proceso se encuentra desarrollado en el archivo "Tarea01.py".

Luego, usando los datos separados en ambas listas, se realizó la aproximación numérica del área bajo la curva de la figura 1. Para esto se implementó el método del trapecio. Con este procedimiento se logró tener $K = 1366090{,}79 erg/(cm^2 * s)$, que es equivalente a $K = 1366{,}09 W/(m^2)$ y que coincide bastante bien con la Constante Solar. El método usado para calcular se encuentra en el archivo Ïntegral1.py".

3. Radio del Sol

Se realizará una aproximación de la energía total por unidad de área emitida por un sol, usando la temperatura efectiva del sol $T_{eff} = 5778K$. La ecuación permitiría hacerlo con cualquier cuerpo negro, pero si usamos la temperatura del sol, obtenemos el resultado particular que buscamos. La función que permite obtener este resultado es la función de Planck:

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

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 puede demostrar que la integral de la funcion anterior corresponde a:

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

De esta integral se puede obtener un resultado analítico que correponde a $\pi/15$, pero al intentar integrar numéricamente de forma manual es dificil plantear de manera directa el algoritmo adecuado por tener uno de los límites tendiendo a infinito. Pero para evitar este problema, podemos usar un tambio de variable $x = tan(y)$. Luego, el diferencial queda $dx = dy(x^2 + 1)$. Luego, remplazando en la fórmula anterior:

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

Esta integral tiene sus límites del dominio bien definidos, por lo que se puede integrar numéricamente. Antes de escribir el código hay que notar que en el dominio considerado se debe notar que la función se indetermina en $y = 0$, pero como la función a integrar toma valores muy pequeños para y cercanos a 0, entonces se evitó el problema integrando desde un número un poco mayor que 1 haciendo el arreglo de y desde una potencia de 10 suficientemente pequeña (ej $y = 0{,}0001$) o partiendo desde el segundo elemento del arreglo. De forma analítica podemos justificar este ajuste dado que el límite de la función cuando y tiende a 0 es 0 (no esdificil mostrarlo usando L'Hopital, por ejemplo). Si se grafica la función entre 0 y $\pi/2$ se puede ver también que cuando y tiende a $\pi/2$, la función se va a 0, por lo que no tendríamos otros inconvenientes con indeterminaciones (esto se explica porque $exptany)$ tiende a infinito mas rápido que $tan(y)^5$.
Aclarando estos puntos es sencillo hacer la integración por método del trapecio. el valor obtenido es $Integral = 6{,}4939$. Notar que este valor corresponde solamente a la integral (sin multiplicar por las constantes). Este valor coincide con el valor analítico mencionado más atrás dentro de las cifras significativas. Luego, la integral vale $P = 63200679{,}69 J/(m^2 s)$ (el método usado está en Ïntegral2.py").
Vale la pena mencionar que en el archivo mencionado se encuentra un cálculo de integral por método de Simpson que fue planteado según ecuaciones vistas en clases. Lo que llama la atención es que entrega exactamente los mismos valores que para la integración que el método del trapecio. Esto puede significar que para esta función en particular ambos métodos coinciden, o que no quedó bien implementado el método. La solución a esta pregunta quedará pendiente porque no se ha logrado encontrar el error por más que sea revisado y porque no he podido probar otra función en dichos métodos (para verificar, por ejemplo, que pudiera ser el caso de que calcen los resultados en la función analizada). Ahora buscaremos calcular el radio solar a través de la ecuación:

$$K = P * (R_s/a_0)^2 \tag{4}$$

donde P corresponde a la integral mencionada en el párrafo anterior, K es el valor de la primera integral calculada, R_s es el Radio Solar y a_0 es una unidad astronómica. Con esta ecuación, despejando el radio se puede obtener. El valor conseguido es $R_s = 695511508{,}148$ m, el cual coincide bastante bien en las cifras significativas con el radio solar real. Este cálculo está desarrollado en la archivo Çomparacion.py".

4. Comparaciones de Resultados y Tiempos de procesamiento

Luego de tener los valores bastante cercanos a los reales, podemos ser aun más rigurosos, buscando las diferencias entre métodos numéricos de integración más precisos contenidos en las librerias de anaconda. En particular, se usará el método del trapecio y "quad"de scipy. El método que se mencionará en esta sección está en el archivo ÇomparacionReal.py".
En esta parte se realizaron la mayoría de los cálculos de las partes anteriores. Se agregaron los calculos de K y de la integral de Planck mediante trapecios de scipy junto con el de Planck mediante la función "quad".
La diferencia entre los valores de K del método de trapecios creado y el que está implementado en scipy es $\Delta I_1 = 3{,}492 * 10^(-9) erg/(cm^2 s)$ que es 15 ordenes de magnitud menor al valor de la integral. Eso significa que ambos métodos se parecen mucho.
Para el caso de la integral de Planck, tenemos que la diferencia entre métodos de trapecio (creado y el que se encuentra en la libreria) es 0 de acuerdo a las cifras que usa el programa. Esto quiere decir que son idénticos, o muy parecidos. De cualquiera manera, para este caso, la diferencia es despreciable. En cuanto a la comparación entre el método creado y la función "quad" $\Delta I_3 = 4{,}261 * 10^(-6)\ J/(m^2 s)$ que es 13 órdenes de magnitud menor. No es tan despreciable la diferencia como en los casos anteriores, pero aun así no pasa a ser significativa. Era previsible que la diferencia entre metodos de trapecio fuera pequeña porque están basadas en las mismas

ideas. Que la diferencia fuera apreciable (a muy bajos ordenes de magnitud) para la primera integral puede deberse a que era una función discreta (dados los datos que fueron entregados, como se puede ver en la figura 1), mientras que la función de Planck era continua y sin demasiadas irregularidades.
Para el caso de diferentes métodos de integración (trapecios vs quad) el método creado es bastante sencillo y no ofrece la posibilidad de mejorar demasiado la toma de datos dado que para obtener una pequeña mejora cuando se disminuye el paso que realiza la integración (se necesita una función analítica para poder hacer esto o una gran cantidad de datos), pero la precisión con respecto al tiempo que tardaría en funcionar no lo compensaría.
Por otro lado, la función predeterminada "quad"tiene métodos de optimización que le permitan mejorar la toma de datos donde hay variaciones imporntantes y tomar pocos datos donde la función es suave. Esto, además, explicaría la diferencia de tiempos que será mencionada a continuación.
Mediante el uso de la función time.time() se contó el tiempo que tarda el programa en realizar las acciones que se requieren. En este caso, se contabilizó el tiempo que tardaba cada programa hecho a mano y los de las librerías. Como este conteo es solo para una muestra (no repite el proceso y toma promedio estadístico), mencionar el tiempo exacto que marca es poco representativo porque depende de la situación actual del computador. De cualquier manera, se dará una noción cualitativa y una aproximación general de acuerdo a las veces que se hizo correr el programa.
La diferencia de tiempos para la primera integral es bastante grande. Es del orden de los segundos y propone que el método incluido en las librerías es mejor. Esto es consistente con lo esperado, ya que probablemente busque optimizar el tiempo, recorriendo mas rápido aquellos pasos que son mas suaves.
Para los otros dos casos, las diferencias son en milésimas de segundos. Puede deberse a que la optimización para buscar un mejor valor (mencionada en la parte anterior) haya perjudicado el tiempo de demora. De cualquier manera, la diferenciasigue indicando que mejores los procesos de las librerías.

5. Conclusión

De acuerdo a lo planteado, podemos observar que la aproximación numérica para ambas integrales es bastante buena. Aun así, es importante saber que existe un error asociado, dada la discretización de la función, vale decir, no se podrá tener el valor exacto para todas las funciones ni para todas las escalas.
Otro aspecto a considerar es que existen métodos que van a permitir reducir mucho mas el error que otros a costo de un tiempo mayor de trabajo. Por esto hay que conocer los métodos para saber como aplicarlos y cuando es necesario para mejorar los datos así como saber cuando basta con usar un método más simple, pero que permita realizar el cálculo mas rápido.

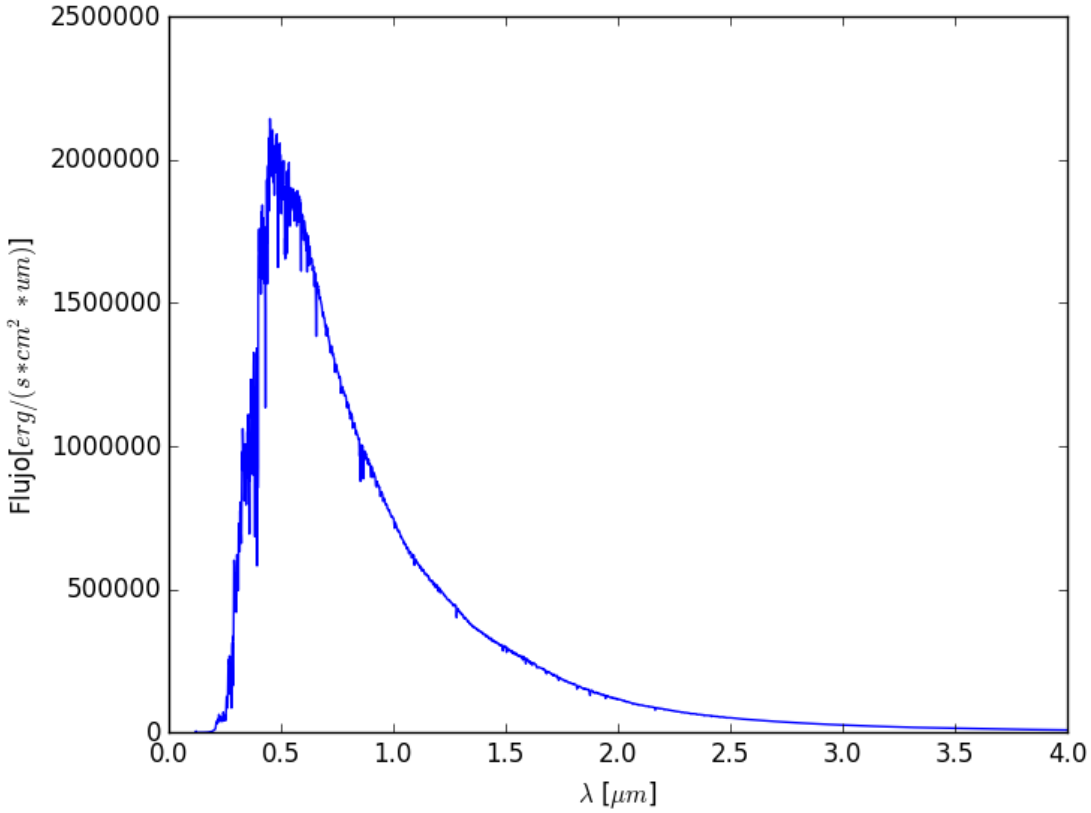

Figura 1: Flujo vs Longitud de Onda