Métodos Numéricos para la Ciencia e Ingeniería
Informe Tarea 10

Benjamín Guerrero
18.862.370-0
1 de Diciembre, 2015

1. Pregunta 1

1.1. Introducción

En este problema pide modelar el contínuo (recta) y la línea de absorción del siguiente gráfico:

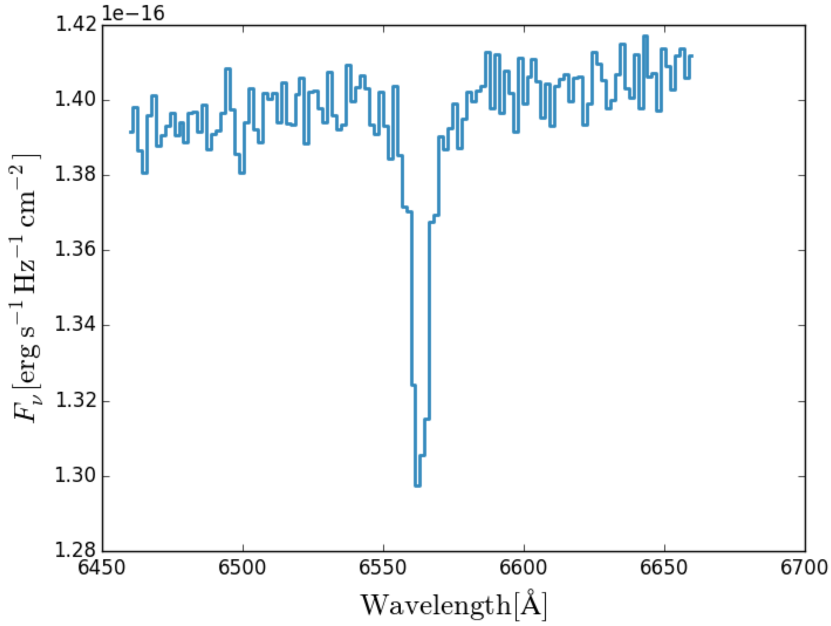

Fig 1: Gráfico frecuencia v/s longitud de onda.

El gráfico de la figura 1 se construyó a partir de los datos presentes en el archivo *espectro.dat*, que tiene 2 columnas de datos. La primera contiene la longitud de onda en Angstroms, y la segunda contiene el flujo por unidad de frecuencia (en [erg s^{-1} Hz^{-1} cm^{-2}]).

Se pide modelar la linea como si tuviese una forma gaussiana, y después como si tuviera un perfil de Lorentz. Luego, se pide comparar los modelos con los datos reales mediante un gráfico.

Finalmente, se pide proveer una tabla con los mejores parámetros de cada modelo, y la función χ^2 de cada modelo.

1.2. Procedimiento

Para resolver este problema se crea un archivo llamado *P1.py*, que retornará lo pedido. Primero, se importa *scipy.stats*, una clase que contiene los módulos *norm* (que retorna una formula de Gauss) y *lorentz* (que retorna un perfil de Lorentz). También se importan *numpy*, *matplotlib.pyplot*, y las funciones para hacer *curvefit*.

Ahora, se definen las funciones *linea*, *gauss*, y *lorentz*, que retornan las funciones que describe cada una (linea es y=x*m+n)

Luego, se definen las funciones modelo para gauss y lorentz, que retornan (línea – gauss/lorentz), básicamente.

Luego, se define la función *chi_2*, que recibe datos, la función modelo y los parámetros, y retorna la función χ^2 de ese modelo.

Con esto, se puede empezar a escribir el main. Primero, se definen los datos del archivo *espectro.dat* usando *np.loadtxt*.

Ahora, se quiere usar *curvefit* para extraer los parámetros óptimos. Para eso, se debe hacer una aproximación inicial de los parámetros. Esto se hace "al ojo", es decir, se deduce la aproximación inicial observando la figura 1. Así, se definen:

$$A\ (amplitud\ de\ la\ linea\ de\ absorción) =\ 0{,}1 * 10^{-16}$$

$$\mu(posición\ de\ la\ línea\ de\ absorción) = 6560$$

$$\sigma(ancho\ de\ la\ curva) = 5$$

$$m(pendiente) = \frac{(0{,}04 * 10^{-16})}{200}$$

$$n(coef\ de\ posición) = 1$$

(También es posible usar *polyfit* para sacar *m* y *n* antes de usar *curvefit*.)

Luego, se hace *curvefit* para cada modelo y se sacan los parámetros óptimos.

Con estos, se puede sacar el χ^2 para cada modelo.

Ahora, queremos construir los gráficos para la comparación. Para esto primero hay que definir nuevos datos a partir de los parámetros óptimos. Con este fin, se define la función *def_datos*, que recibe un modelo y los parámetros, y retorna los datos que siguen el modelo.

Ahora, hay que graficar esos datos. Por lo tanto, se crea la función *graficar_datos*, que grafica los datos originales (es decir, más o menos lo que está en la fig 1), junto con los datos fiteados con los modelos.

Finalmente, el programa retorna el gráfico llamado *Grafico_P1.jpg*, los valores de los parámetros y los valores de χ^2 para los dos modelos.

1.3. Resultados

Para el modelo de perfil de Lorentz, los parámetros óptimos fueron, aprox. los siguientes:

$$A = 1{,}114 * 10^{-16}$$

$$\mu = 6563{,}2$$

$$\sigma = 3{,}219$$

$$m = 7{,}923 * 10^{-21}$$

$$n = 8{,}811 * 10^{-17}$$

Para el modelo de Gauss, fueron aproximadamente los siguientes:

$$A = 8{,}223 * 10^{-17}$$

$$\mu = 6563{,}223$$

$$\sigma = 3{,}258$$

$$m = 7{,}803 * 10^{-21}$$

$$n = 8{,}877 * 10^{-17}$$

También se sacó el valor de χ^2 para cada modelo. Aproximadamente, estos son:

$$\chi^2(Lorentz) = 5{,}006 * 10^{-35}$$

$$\chi^2(Gauss) = 5{,}204 * 10^{-35}$$

Finalmente, el gráfico pedido es el siguiente:

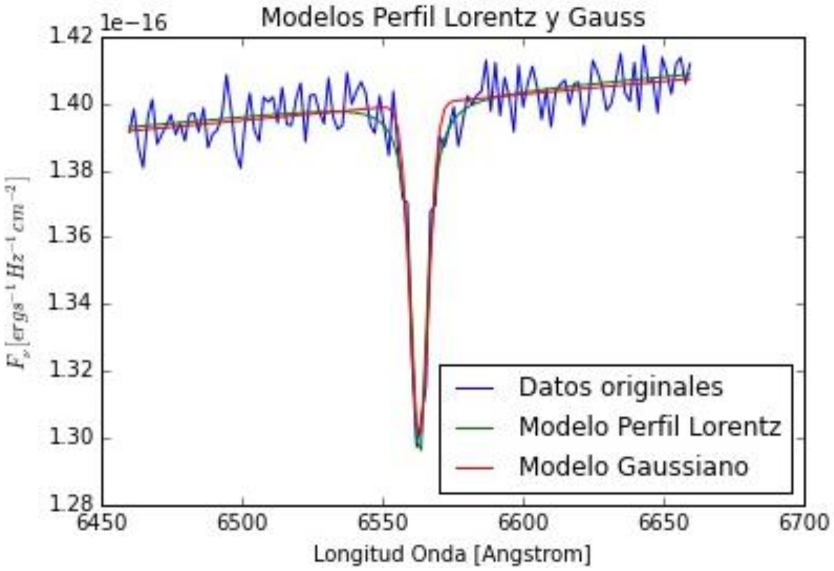

Figura 2: Comparación entre datos originales y modelos.

2. Pregunta 2

2.1. Introducción

Ahora, se pide determinar cuál de los dos métodos es el que representa mejor a los datos. Como los errores no son normales, la prueba de χ^2 no sirve. Así, se pide usar Kolmogorov-Smirnov (K-S) para determinar la hipótesis nula.

2.2. Procedimiento

Como se pide usar los modelos usados anteriormente, esta parte de la tarea también será realizada por *P1.py*. (Sí, el nombre del archivo se vuelve inapropiado.)

Primero, al código ya escrito se le añaden las funciones *cdf* (que es la función de distribución acumulativa), y *generar_dist_acumulada*, que genera la distribución de probabilidad para cada modelo, y que se necesitan para usar K-S.

Luego, en el main , luego de llamar a la función *generar_dist_acumulada*, se usa la función *kstest* de *scipy.stats* para realizar el test de K-S.

Además, se usa una función llamada *graficar_prob* para graficar las probabilidades acumuladas para cada modelo.

Ahora, además de los datos pedidos para la parte anterior, el programa además retorna las estadísticas de K-S (Dn_scipy) y los niveles de confianza para cada modelo. También retorna 2 gráficos de probabilidad (*P_Lorentz.jpg* y *P_Gauss.py*).

2.3. Resultados

Para el perfil de Lorentz, el valor de Dn_scipy es de 0,164, y el nivel de confianza es de 0,0024.

Para el modelo de Gauss, el valor de Dn_scipy es de 0,172, y el nivel de confianza es de 0,0012.

Los gráficos de probabilidades acumuladas son los siguientes:

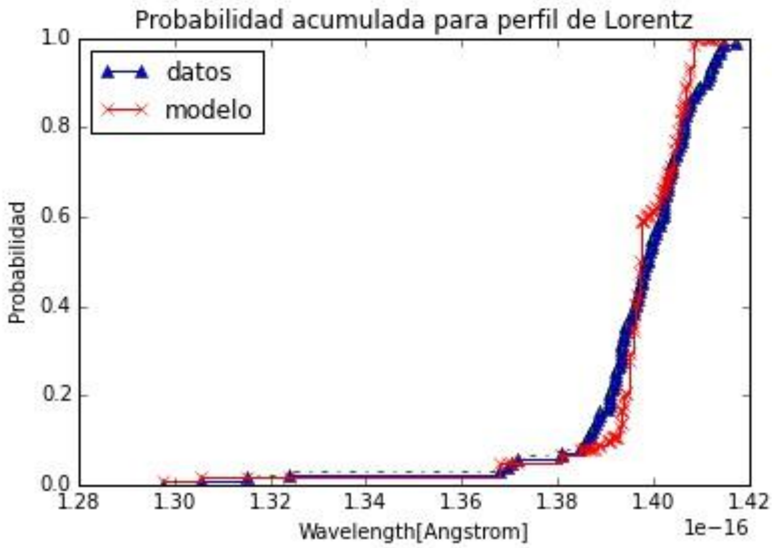

Fig 3: Probabilidad acumulada de Lorentz

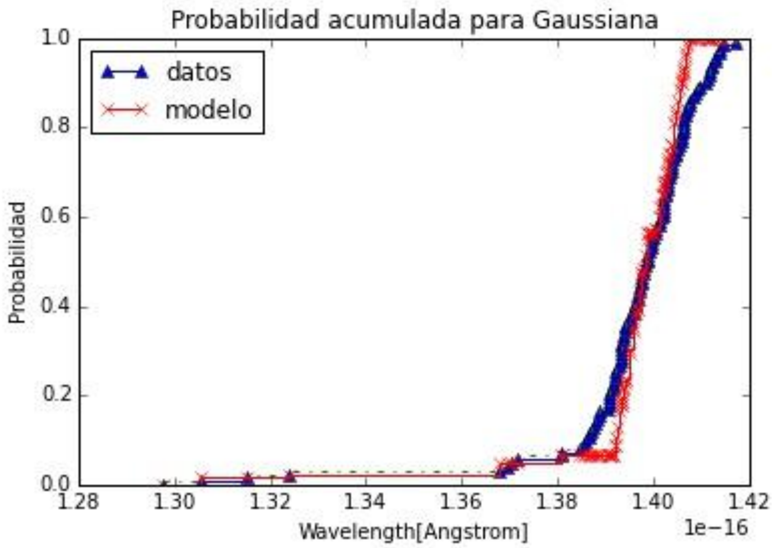

Fig 4: Probabilidad acumulada de Gauss

3. Conclusiones

A partir de los resultados, se ve en la figura 2 que los modelos son fiteos aceptables, si se les compara con los datos originales. Se nota que, por la mayor parte, los parámetros óptimos y los valores de χ^2 son muy parecidos entre los dos modelos, lo que era esperado puesto que los dos modelos son similares.

Lo más notable fue la diferencia entre el coeficiente de posición "al ojo", y el coeficiente óptimo. Sin embargo, al intentar cambiar el valor inicial, se produce un peor fiteo de los modelos. Es posible que esto se deba a que el valor inicial se hace "al ojo".

Ahora, viendo los resultados de la parte 2, se ve que ninguno de los modelos son "buenos", ya que tienen un muy bajo nivel de confianza (menor al 0,25%), al hacer el test de K-S. Eso sí, se ve que el nivel de confianza del perfil de Lorentz 2 veces mayor que el nivel de confianza del modelo gaussiano.

Así, de acuerdo a este test, el perfil de Lorentz es el mejor modelo de los dos.