Calculadora cosmólogica: Características del software.

Javier Gutiérrez Solórzano

1 de febrero de 2014

Resumen

A continuación se realizará una breve descripción de cómo se puede ejecutar el código en los entornos de desarrollo eclipse y Bluej. También se describirán los algoritmos utilizados en cada una de las clases de las que consta la calculadora en relación con las ecuaciones de Friedmann.

1. Instalación y ejecución del código.

Para ejecutar el programa se debe tener instalado un entorno de desarrollo con el que poder ejecutar código java como eclipse o Bluej. Además se necesitará instalar el paquete *fundamentos*. El paquete *fundamentos* es un paquete de software libre, desarrollado por Michael González y Mariano Benito Hoz que nos proporciona un conjunto de clases para entrada/salida gráfica. Se utilizará para introducir los datos en el programa, para obtener los valores numéricos y los dibujos de las gráficas. Toda la información necesaria relativa al paquete (incluyendo descarga e instalación) se encuentra en la página web [1].

(No queda muy claro cómo instalarlo en eclipse. La forma de proceder es la siguiente: descargamos el paquete *fundamentos.jar* desde [1]; dentro del entorno de eclipse pinchamos sobre el paquete calculadora, botón derecho *Build Path* $\longrightarrow$ *Configure Build Path...*; pestaña *libraries* $\longrightarrow$ *Add External JARs...*, buscamos el paquete lo selecionamos y después *Ok*)

2. Características generales del código.

El software consta de dos programas independientes escritos en Java llamados *calculadora* y *calculadoraGrafica*. Ambos constan de cuatro clases, tres de ellas son comunes y otra que contiene el programa principal que es propia de cada proyecto. El código se ha distibuido de esa forma porque para el que lo ha escrito, resulta mas fácil de entender y de explicar.

El primer programa se utiliza para calcular los valores numéricos de distancia propia, distancia de diametro angular y de luminosidad en función del corrimiento al rojo y la edad actual del universo. El segundo nos proporciona tres ventanas en las que representan gráficamente la distancia propia, de diámetro angular y de luminosidad en función de z, la edad del universo en función del corrimiento al rojo y el factor de escala en función de la edad del universo. Las ventanas aparecerán en el orden en el que se ha indicado siendo una ventana sustituida por otra a medida que pulsamos el recuadro aceptar en la ventana de la gráfica.

2.1. Descripción detallada de cada clase.

clase *Trapecio.*

Permite obtener el valor de la integral definida de una función genérica f por el método de los trapecios. Para ello debemos indicarle que función integrar, los límites de integración y el número de intervalos que queremos tomar para realizar la integral. El código se encuentra en cualquier libro de cálculo numérico o en la página web [2].

clase *FuncionA* y clase *FuncionC*

En éstas clases se definen las funciones sobre las que se realiza la integral. Constan de un contructor y de un método que nos devuelve la función. En la clase *FuncionA* se define la función:

$$\frac{1}{x^2\sqrt{\Omega_\Lambda + \Omega_k x^{-2} + \Omega_M x^{-3}}} \tag{1}$$

y en *FuncionC* la función

$$\frac{1}{x\sqrt{\Omega_\Lambda + \Omega_k x^{-2} + \Omega_M x^{-3}}} \tag{2}$$

clase *distancia.*

Admite como parámetros de entrada z, Ω_M, Ω_Λ y H_0 en (km/s)/Mpc. Contiene el método principal del progama que nos proporciona los valores numéricos.

En la parte superior de la clase se declaran las variables y se prepara la pantalla en la que se introducen los datos. Tras éstos aparecen los primeros cálculos, el primero de ellos calcula Ω_k que nos determina qué curvatura tiene el espacio con los parámetros de entrada que hemos introducido. Un segundo cálculo obtiene el valor del límite inferior de la integral definida de la función 1 que se realiza a continuación creando el objeto $S1$ de la clase *FuncionA*. La integral tiene límite superior 1, inferior $\frac{1}{1+z}$ y se realiza dividiendo el intervalo de integración en 1000 partes. La distinción entre las distintas curvaturas es necesaria porque los tres tipos distintos de curvatura, nula, positiva o negativa, necesitan tres algoritmos distintos para calcular el valor de la distancia propia dp.

El primer algoritmo, para el caso de curvatura positiva, realiza la operación:

$$dp = \frac{c}{H_0\sqrt{-\Omega_k}} \sin\left(\sqrt{\Omega_k} S1\right) \tag{3}$$

siendo c la velocidad de la luz en el vacío en km/s. El signo $-$ delante de Ω_k es necesario para que al argumento de la raíz sea real. El segundo, cuando la curvatura es negativa, hace la operación:

$$dp = \frac{c}{H_0\sqrt{\Omega_k}} \sinh\left(\sqrt{\Omega_k} S1\right) \tag{4}$$

y el tercero si la curvatura es nula:

$$dp = \frac{c \cdot S1}{H_0} \tag{5}$$

La distinción se hace con un bule if a partir del valor de Ω_k.

El cálculo de la distancia angular da y de la distancia luminosidad dl se realiza mediante las expresiones:

$$da = \frac{dp}{1+z} \tag{6}$$

y

$$da = dp \cdot (1+z) \tag{7}$$

a partir del cálculo de la distancia propia calculado anteriormente.

El cálculo de la edad del universo se realiza multiplicando la inversa de la constante de hubble, por el valor de la integral sobre la función 2 entre los límites de integración 0 y 1 con 1000 intervalos de integración. En la misma multiplicación aparecen otras constantes con el fin de obtener el resultado en Gyr. El valor de la integral se obtiene creando el objeto SC de la clase FuncionC. La razón por la que aparece 0,0001 como límite de integración y no 0 es porque de no hacerse así el resultado de SC es 0, lo cuál es incorrecto.

Por último aparece código para mostrar en pantalla los resultados.

clase *graficas*.

Admite como parámetros de entrada Ω_M, Ω_Λ y H_0 en (km/s)/Mpc, que son los mismos que los de la clase anterior, solo que en éste caso es inecesario que le pasemos el valor de z. Contiene el método principal del progama que nos proporciona las representaciones gráficas.

La estructura es muy similar a la clase *distancia*. Lo primero que nos encontramos es el código donde se declaran las variables, se prepara la pantalla para introducir los datos y se calcula Ω_k. Después nos encontramos con secciones independientes de código encabezadas por un comentario, que tienen una estructura muy parecida. Las tres se inician creando un objeto de la clase *Grafica* del paquete *fundamentos*. Cada uno de los tres objetos creará una ventana con una gráfica a la que refiere el encabezamiento.

/**

* Dibuja la grafica de distancia propia, de diametro angular y de luminosidad en funcion de z en una misma grafica

*/

Se compone tres bucles for responsables de cada uno de los tres trazos correspondientes a dp, da y dl que aparecen en la primera ventana. En cada bucle for hay anidado un bucle if. Éstos son esencialmente iguales al que aparece en la clase *distancia*, pero se han añadido algunas modificaciones. La principal de ellas es que el cálculo de las distancias dp, da y dl se realizan dentro de bucles if independientes. Es decir el cálculo de dp no se aprovecha para calcular el valor de da y dl como ocurría anteriormente. Esto quiere decir que el código es bastente ineficiente, pues se estan realizando tres veces algunas operaciones. Como ventaja es un código muy fácil de escribir y de entender. El número de iteraciones del bucle for determina el número de puntos de la gráfica y después éstos se unen para formar un trazo contínuo.

/**
* Edad del universo en funcion del corrimiento al rojo
*/

Está compuesto de un único bule for que determina el número de puntos de la gráfica que se calculan. En primer lugar se declara el parámetro b que representa el valor del límite superior de integración que cambia en cada iteración. Después se pasa el crea un objeto de la clase *FuncionC* y se calcula el valor de la edad del universo de la misma forma que se hacía en la clase *distancia* y se representa cada punto.

/**
* Factor de escala en funcion de la edad del universo
*/

Éste algoritmo realiza exactamente los mismos cálculos que el código anterior; solo hay una diferencia y es la representación de éstos. En lugar de representar la edad como función del corrimiento al rojo, ahora representamos el parámetro de integración en función de la edad. Esto es así ya que la relación entre el parámetro de escala y se relaciona con el corrimiento al rojo con la expresión:

$$R = \frac{R_0}{1+z} \tag{8}$$

Si tomamos $R_0 = 1$ entonces $R = b$. De nuevo se podrían utilizar los cálculos anteriores para realizar la representación, pero en éste caso es bastante más probable que no queramos hacerlo de ese modo. La razón es que puede resultar útil representar el factor de escala mas allá del valor $R_0 = 1$, de hecho el algoritmo no toma $R = \frac{1}{1+z}$ sino $R = \frac{2}{1+z}$ con objeto de realizar una representación mas amplia, hasta $R(t) = 2$.

Referencias

[1] Dirección electrónica: http://www.ctr.unican.es/Fundamentos/

[2] Dirección electrónica: http://www.sc.ehu.es/sbweb/fisica/cursoJava/numerico/integracion/trapecio/trapecio.htm