Calculadora Cosmológica: Características del Software

Guadalupe Cañas Herrera

Resumen—Descripción acerca del programa "CosmoCalculator": cómo instalar y ejecutar el programa, así como el contenido del mismo. Se describen los algoritmos utilizados en cada rutina que compone el programa en términos de las ecuaciones de Friedmann.

1. Instalación y ejecución

El código del programa "CosmoCalculator" está escrito en lenguaje C. Para poder ejecutar el programa es necesario tener instalado en el ordenador un entorno de desarollo con el que se permite ejecutar código escrito en C. A continuación se plantean dos posibilidades para instalar y ejecutar el programa en diferentes sistemas operativos:

1.1. En Mac OS X

Para ejecutar el programa en OS X, se recomienda tener instalado el entorno de desarrollo integrado Xcode de Apple Inc. Este entorno es suministrado gratuitamente junto con Mac OS X desde la versión 10.3, y actualmente desde la versión 10.9 es descargable desde la App Store sin necesidad de recurrir a Apple Developer. Xcode incluye la colección de compiladores del proyecto GNU (GCC), y puede compilar código C. El GNU Compiler Collection (colección de compiladores GNU) es un conjunto de compiladores creados por el proyecto GNU y son considerados estándar para los sistemas operativos derivados de UNIX, de código abierto y también de propietarios, como Mac OS X. GCC es software libre y lo distribuye la Free Software Foundation (FSF) bajo la licencia general pública GPL.

Con la colección de compiladores GNU ya instalado en nuestro ordenador de la mano de Xcode, es posible ejecutar el programa "CosmoCalculator.c" desde la terminal. Para ello, guardamos el archivo "CosmoCalculator.c" en la localización desde la cual nos gustaría ejecutar el programa. El autor recomienda, si se trabaja con OS X, guardar el archivo en el directorio del Usuario Apple tal y como muestra la figura 2. La elección de esta localización es debido al hecho de que para compilar el programa se va a utilizar la terminal, y la terminal siempre escoje por defecto como lugar de partida el directorio del Usuario Apple. Así, si se guarda el archivo en este lugar desde el primer momento, la persona que desee ejecutar el programa no tiene que preocuparse por moverse a otro directorio desde la terminal.

Una vez guardado el programa, se procede a abrir una línea de comandos con la terminal como muestra la figura 1 para compilar nuestro archivo "CosmoCalculator.c". Desde la terminal, se llamará al compilador GNU C Compiller incluido en el GNU Compiler Collection. Para ello, se escribirá el comando "gcc". El comando "gcc" tiene una serie de funciones (visibles desde "gcc –help"), entre las que se incluyen la posibilidad de renombrar al archivo de salida que nos devolverá el copilador tras compilar el programa. Esta función se logra con "gcc -o NombreArchivoSalida NombrePrograma.c". En el ejemplo presentado en la Figura 1, se observa como al archivo de salida obtenido tras compilar "CosmoCalculator.c" se le ha llamado "Ejecutable", ya que será un archivo .out (formato de archivo usado en versiones de sistemas operativos tipo Unix para ejecutables). En caso de que el programa ".c" tenga algún error o alguna advertencia detectada por el compilador, aparecerá en la pantalla de la terminal.

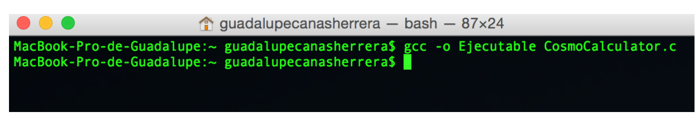

Figura 1. Ejemplo que muestra como se utiliza la función "gcc -o NombreArchivoSalida NombrePrograma.c" desde la terminal.

En la Figura 2 es posible observar que tras la instrucción introducida en la línea de comandos, se ha creado el archivo "Ejecutable.out" en el mismo directorio en el cual se guardó y desde el cual se compiló "CosmoCalculator.c".

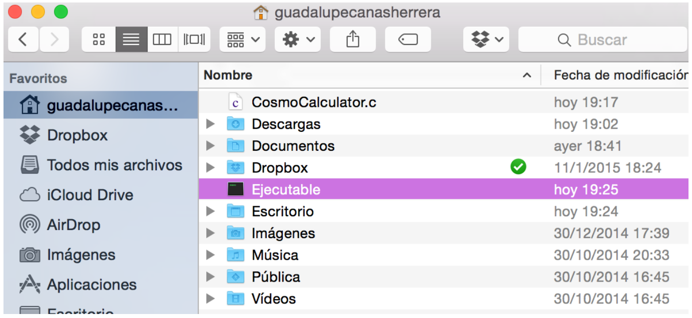

Figura 2. Ejemplo que muestra como se crea el archivo ejecutable en el directorio desde el cual se compiló el programa .

Finalmente, ya sólo nos queda ejecutar el archivo “Ejecutable.out” desde la línea de comandos escribiendo “./Ejecutable”. Automáticamente se ejecutará el programa “Cosmological Calculator”, que te pedirá por pantalla los datos iniciales (en la Figura 3 se ha ejecutado el archivo “Ejecutable.out”).

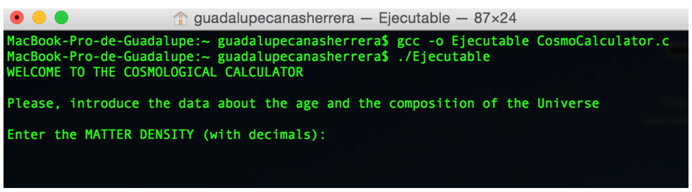

Figura 3. Ejemplo que muestra como se ejecuta el archivo “.out” desde la terminal y qué aspecto tiene la calculadora cosmológica en la terminal.

Otra manera de ejecutar el programa más rápidamente es llamar al compilador “gcc” sin niguna función, escribiendo “gcc CosmoCalculator.c”. El compilador creará por defecto un archivo ejecutable llamado “a.out”. Nuevamente, ejecutamos dicho archivo desde la línea de comandos. Observese como el resultado presente en la figura 4 es el mismo que en la figura 3.

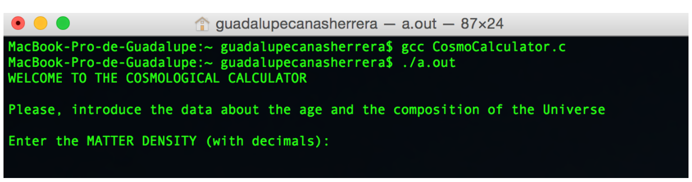

Figura 4. Ejemplo que muestra una manera alternativa de compilar y ejecutar el archivo “CosmoCalculator.c” de una manera más rapida.

1.2. En Windows

Para instalar y compilar el programa en Windows, el lector puede dirigirse al informe “Características del Software”de Palmerina González Izquierdo.

2. Rutinas del programa y características del código

El programa está compuesto por una rutina principal (main) y tres funciones que se requieren en dicha rutina. Se ha necesitado incluir dos cabeceras para el correcto funcionamiento del código.

1. **stdio.h**: es un archivo de cabecera de la biblioteca estándar del lenguaje de programación C que contiene las definiciones de macros, las constantes, las declaraciones de funciones y la definición de tipos usados por varias operaciones estándar de entrada y salida.
2. **math.h**: es un archivo de cabecera de la biblioteca estándar del lenguaje de programación C diseñado para operaciones matemáticas básicas.

Antes de entrar en la función main, se declaran tres constantes (el valor de la velocidad de la luz C, y dos cambios de unidades: a Giga-años y a Mega-parsecs) y 7 variables globales que son utilizadas tanto en el main como en las funciones:

1. Densidad de Materia Ω_M
2. Densidad de Energía Oscura o de vacío Ω_Λ
3. Densidad de Radiación Ω_R
4. Densidad de Curvatura Ω_k
5. Constante de Hubble H_0
6. Corrimiento al rojo z
7. Factor de escala $R(t)$ (se fija como $R(t_0) = 1$)
8. Parámetro de deceleración q_0

2.1. Función ecuacion1

En esta función se define la siguiente ecuación,

$$\frac{1}{x^2\sqrt{\Omega_\Lambda + \Omega_k x^{-2} + \Omega_M x^{-3} + \Omega_R x^{-4}}} \tag{1}$$

a la cual se le pasa como parámetro la variable x.

2.2. Función ecuacion2

En esta función se define la siguiente ecuación,

$$\frac{1}{x\sqrt{\Omega_\Lambda + \Omega_k x^{-2} + \Omega_M x^{-3} + \Omega_R x^{-4}}} \tag{2}$$

a la cual se le pasa como parámetro la variable doble x.

2.3. Función Integral

Esta función calcula el valor de la integral de una función cualquiera definida entre dos límite mediante el método conocido como “de los trapecios”. Es necesario pasarle a esta función varios parámetros,

1. límite de integración superior
2. límite de integración inferior
3. número de intervalos en el cual queremos dividir el area bajo nuestra función arbitraria
4. puntero que señala qué ecuación queremos integrar (en nuestro caso, la 1 o la 2)

El algorítmo para integrar cualquier función mediante el método de trapecios es básico en el cálculo numérico. En este caso, el método se ha estudiado mediante [1]. Se ha comprobado que a partir de 1000 intervalos (con un método de integración de trapecios al cual se le pasa un error de 0.01), la precisión del resultado no varía si se añaden más números de intervalos.[1]

2.4. Función principal: main

2.4.1. Primera parte: cálculo valores numéricos

En primer lugar, se piden por pantalla, se leen y se guardan los valores de Densidad de Materia Ω_M, Densidad de Energía Oscura o de vacío Ω_Λ, Densidad de Radiación Ω_R, Constante de Hubble H_0 y Corrimiento al rojo z.

A continuación se calcula la densidad de curvatura según,

$$\Omega_k = 1 - (\Omega_M + \Omega_\Lambda + \Omega_R) \tag{3}$$

El factor de escala como,

$$R(z) = \frac{1}{1+z} \tag{4}$$

donde se ha fijado $R(t_0) = 1$. Y el parámetro de deceleración como [2],

$$q_0 = 0{,}5\Omega_M + \Omega_R - \Omega_\Lambda \tag{5}$$

Posteriormente se calcula la integral de la ecuación 1 introduciendo como límite inferior el factor de escala $R(t)$, límite superior 1, y se hacen 1000 intervalos de integración. A este resultado se le llama "Integral1". Con esta integral es posible calcular la distancia comóvil radial como,

$$dc = \frac{c}{H_0} Integral1 \tag{6}$$

Con este resultado se puede calcular la distancia propia según la densidad de curvatura obtenida mediante la ecuación (3) (en el programa se distingue entre los tres posibles casos con una estructura "if... else if").

$$dp = \frac{c}{H_0\sqrt{-\Omega_k}} \sin \sqrt{\Omega_k} Integral1 \text{ si } \Omega_k < 0 \tag{7}$$

$$dp = \frac{c}{H_0\sqrt{-\Omega_k}} \sinh \sqrt{\Omega_k} Integral1 \text{ si } \Omega_k > 0 \tag{8}$$

$$dp = \frac{c}{H_0} Integral1 \text{ si } \Omega_k = 0 \tag{9}$$

1. Se han incluido otros dos métodos de integración numérica (Simpson y Cuadratura de Gauss-Legendre) con el fin de estudiar si el método explicado con anterioridad (integración mediante trapecios) presentaba un error sistemático. Para más información véase el Documento de Resultados y Análisis. El método de integración de Gauss-Legendre se ha obtenido de [4]. Estos metodos se encuentran comentados en el programa, pero el lector puede ejecutarlos simplemente descomentándolos y llamándolos en la función main mediante la creación de los resultados de la integral al igual que se hace con la función Integral.

donde c es la velocidad de la luz en el vacío en km/s.

A partir de la distancia propia es posible calcular la distancia de diámetro angular como,

$$da = dpR(z) \tag{10}$$

Y la distancia de luminosidad,

$$dl = \frac{dp}{R(z)} \tag{11}$$

La edad del universo se calcula mediante la ecuación,

$$t = \frac{1}{H_0} Integral2 \tag{12}$$

donde integral2 es la integral de la ecuación 2 desde 0 hasta 1 haciendo 1000 intervalos de integración (obsérvese que en el programa no se ha fijado 0 como el límite inferior de integración sino 0,0001 ya que sino el lagoritmo de la integración mediante trapecios no funciona). Además, es posible calcular el tiempo que tarda la luz en viajar desde un determinado corrimiento al rojo z hasta el presente calculando,

$$t_{light} = \frac{1}{H_0} Integral2\_z \tag{13}$$

donde integral2_z es la integral de la ecuación 2 desde un R(t) determinado hasta 1 haciendo 1000 intervalos de integración. Finalmente, se calcula el tiempo que tarda un fotón en viajar desde el big bang hasta un determinado corrimiento al rojo z como,

$$t_z = t - t_{light} \tag{14}$$

Todos los tiempos se expresan en Giga-años realizando los correspondientes cambios de unidades. La razón de expresar el tiempo en Giga-años viene determinado por el hecho de que la edad del universo es aproximadamente 13,75 Gyr, con lo cual se observa que los Gyr son las mejores unidades para expresar el tiempo de manera sencilla. Además, durante el análisis de los resultados, se ha comparado resultados con otras calculadoras cosmológicas que expresan el tiempo en Gyr, lo que permite comparar resultados de manera más rápida y eficiente.

2.4.2. Segunda parte: creación del fichero .dat

El programa también crea un fichero .dat llamado "datos cosmo k.dat" (aquí se sustituye la k por el valor correspondiente de manera que no se sobreescriban los archivos una vez creados y se puedan importar a otros programas sin problemas).

Se vuelven a calcular la distancia propia, de diámetro angular la distancia de luminosidad, el factor de escala y la edad del universo en función de z. Para ello se utiliza un bucle for desde z=0 hasta z=100 en intevalos de 0.01, calculando nuevamente el resultado integral1 e integral2 y utilizando un nuevo factor de escala que nos permite calcular la edad del universo en un futuro. Este nuevo factor de escala se define como,

$$R(z) = \frac{2{,}5}{1+z} \tag{15}$$

Para obtener la edad del universo en el futuro, se deberían de usar corrimientos al azul ($z < 0$) pero como los métodos ecuacion1 y ecuacion2 están escritos en función de x (que en realidad es un cambio de variable relacionado con el favor de escala, véase el documento 2) es necesario utilizar el resultado presente en la ecuación (15) que redefine este factor de escala hasta el futuro con $R(t_{futuro} = 2,5)$. Estos resultados se escriben en columna debidamente separadas y con una cabecera que indica qué magnitud se está calculando.

Finalmente, el archivo se guarda en el directorio desde el cual se ejecuta el programa. Este archivo se utilizará para representar datos gráficamente. En este caso, el programa utilizado para realizar gráficas es Mathetica. Véase [3] para obtener información acerca de cómo importar archivos .dat a Mathematica.

REFERENCIAS

[1] http://www.sc.ehu.es/sbweb/fisica/cursoJava/numerico/integracion/trapecio/trapecio.htm (VISITADA 10/12/2014)
[2] B. W. Carroll/ D. A. Ostlie "An Introduction to Modern Astrophysics" Second Edition. Pearson New International Edition.
[3] http://reference.wolfram.com/language/tutorial/ImportingAndExportingData.html (VISITADA 10/12/2014)
[4] http://rosettacode.org/wiki/Numerical_integration/Gauss-Legendre_Quadrature#C (VISITADA 01/02/2015)