

La solución a este taller debe subirse por SICUA antes de las 10:00PM del viernes 24 de Febrero del 2017. Los archivos código fuente deben subirse en un único archivo `.zip` con el nombre `NombreApellido_hw1.zip`, por ejemplo yo debería subir el zip `VeronicaArias_hw1.zip` (10 puntos). Recuerden que es un trabajo individual.

1. (35 points) **Curvas de rotación de Galaxias**

Una de las primeras evidencias indirectas de la existencia de materia oscura fue el resultado de las observaciones hechas por Vera Rubin de las curvas de rotación de las galaxias. Estas observaciones consisten en medir las velocidades de las estrellas en función de la distancia de éstas al centro de la galaxia. A partir de estas velocidades uno puede inferir la cantidad de masa que debe tener la galaxia. Al compararla con la masa inferida a partir de la luminosidad de la galaxia (masa de la materia luminosa), Vera Rubin encontró que la masa inferida a partir de las velocidades es mayor que la masa "luminosa" y que por lo tanto debe haber materia oscura.

En <http://iopscience.iop.org/1538-3881/122/5/2396/fulltext/datafile3.txt> están datos de velocidad observada (columna 7) en función del radio en kpc (columna 2) para una serie de galaxias. Además hay datos de velocidades esperadas si la masa de la galaxia fuera solo la del gas, la del disco de la galaxia o la del bulbo de la galaxia (columnas 4, 5 y 6 respectivamente).

El primer ejercicio consiste en desarrollar un script (llamado `CurvasDeRotacion.sh.`) que baje y cree un archivo de datos y una rutina en python (llamado `LOTS_RotationCurves.py.`) con varias funciones que les permitan leer, organizar y graficar los datos.

El script `CurvasDeRotacion.sh.` debe:

- Crear un directorio llamado `RotationCurves` y entrar a dicho directorio
- Bajar el archivo `datafile3.txt`
- A partir de `datafile3.txt` generar un archivo llamado `RotationCurve_F571-8.txt` que contenga los datos de la galaxia F571-8
- Correr la rutina `LOTS_RotationCurves.py.` en python
- Borrar el archivo `datafile3.txt`

La rutina de Python `LOTS_RotationCurves.py.` debe:

- Leer el archivo llamado `RotationCurve_F571-8.txt` y guardar las variables relevantes en arrays
- Graficar en una misma gráfica la velocidad medida en función del radio (en color negro) y la suma de las velocidades esperadas (gas+disco+bulbo) (en color verde). Esta gráfica debe ser clara, con *labels* para las distintas curvas y ejes debidamente rotulados.
- Guardar la gráfica anterior (sin mostrarla) en `RotationCurvePlot.pdf`

2. (55 points) **Convección en la atmósfera**

La convección es uno de los mecanismos de transferencia de calor en la atmósfera. Para que haya convección se necesita que una burbuja de aire que se desplaza una distancia Δz hacia arriba, después de expandirse para quedar a la misma presión que sus alrededores, sea menos densa que estos, y por lo tanto siga subiendo. A partir de datos de temperatura T en función de la altura z se puede calcular el gradiente vertical de temperatura $\frac{dT}{dz}$. Asumiendo una atmósfera seca, se puede determinar si a distintas alturas la atmósfera es estable o inestable. Para esto se compara el gradiente de temperatura observado $\frac{dT}{dz}$ con el gradiente de temperatura adiabático seco $(\frac{dT}{dz})_{ad} = -9.8^\circ/km$.

Para este ejercicio deben utilizar datos de temperatura medidos por una radiosonda en Bogotá (archivo `DatosRadioSonda.dat`, tomado de <http://weather.uwyo.edu/upperair/sounding.html>). Deben escribir un script (llamado `Convection.sh`) que genere un archivo de datos y una rutina en python (llamado `PLOTS_Convection.py`) con varias funciones que les permitan leer, organizar, interpolar, derivar y graficar los datos.

El script `Convection.sh` debe:

- Crear un directorio llamado `Convection` y entrar a dicho directorio
- Mover el archivo `DatosRadioSonda.dat` al directorio `Convection`
- A partir de `DatosRadioSonda.dat` generar un archivo llamado `TempHeight.txt` que contenga sólo los datos de temperatura y altura.
- Correr la rutina `PLOTS_Convection.py` en python

La rutina de Python `PLOTS_Convection.py` debe:

- Leer el archivo llamado `TempHeight.txt` y guardar las variables relevantes en arrays
- Graficar Temperatura medida en función de la altura (en color rojo). Esta gráfica debe ser clara, con *labels* para la curva y ejes debidamente rotulados.
- Guardar la gráfica anterior (sin mostrarla) en `TemperaturePlot.pdf`
- Interpoliar los datos medidos entre $z = 2500$ y $z = 25000m$ (que tienen intervalos irregulares) con el método de splines para intervalos de 150m. Hay un ejemplo en: <https://github.com/ComputoCienciasUniandes/MetodosComputacionales/tree/master/notes/07.Interpolation>.
Generar a partir de la interpolación, una serie de datos de Temperatura versus altura para intervalos regulares de altura
- Calcular el gradiente de temperatura usando derivación numérica.
- Comparar el gradiente vertical de temperatura con el gradiente adiabático a lo largo de z . Graficar los puntos para los cuales el **valor absoluto** del gradiente de temperatura es mayor que el del adiabático (criterio para que haya convección) y guardar esa gráfica (sin mostrarla) en `ConvectionPLOT.pdf`
- Graficar el gradiente vertical de temperatura en función de z y el gradiente adiabático en función de la altura. Esta gráfica debe ser clara, con *labels* para la curva y ejes debidamente rotulados.
- Guardar la gráfica anterior (sin mostrarla) en `GradientsPlot.pdf`