

Métodos Computacionales

Tarea 2 — 2018-10

La solución a este taller debe subirse por SICUA antes de las 5:00PM del lunes 5 de marzo del 2018. Si se entrega la tarea antes del lunes 26 de febrero del 2018 a las 11:59PM los ejercicios se van a calificar con el bono indicado.

(10 puntos) Los archivos del código deben subirse en un único archivo `.zip` con el nombre `NombreApellido_taller2.zip`, por ejemplo si su nombre es Maria Cano debería subir el zip `MariaCano_taller2.zip` al descomprimir el zip debe crearse la carpeta `MariaCano_taller2` y adentro deben estar los archivos solicitados.

1. Concentración de CO₂ en la atmósfera.

El objetivo de este ejercicio es explorar los datos de concentraciones de CO₂ en la atmósfera, mirar cuáles han sido las tendencias desde 1960 y estudiar la tasa a la cuál se está incrementando dicha concentración.

Para esto debe escribir un script `analiza_CO2.sh` para descargar los datos de `ftp://aftp.cmdl.noaa.gov/products/trends/co2/co2_annmean_mlo.txt` y correr un script de python `analiza.py` que analice los datos.

El script de python debe:

- (5 puntos) Graficar en el archivo `C02.png` la concentración de CO₂ atmosférico en función del tiempo.
- (5 puntos) Graficar en el archivo `derivada1_CO2.png` la tasa de cambio de dicha concentración en función del tiempo.
- (10 puntos) Graficar en el archivo `derivada2_CO2.png` la segunda derivada de la concentración de CO₂ en función del tiempo. Esta gráfica también debe incluir una línea horizontal que marca la pendiente media de la tasa de cambio (la gráfica anterior) calculada entre 1960 y 2016.

2. Análisis de datos de temperatura

En este ejercicio debe hacer un análisis de datos de las temperaturas promedio mensuales en Nottingham durante 20 años.

Para esto debe escribir un script `minombre_miapellido_temp.sh` para descargar los datos de `https://raw.githubusercontent.com/vincentarelbundock/Rdatasets/master/csv/datasets/nottem.csv` y correr un script de python `temperaturas.py` que analice los datos.

El script de python debe hacer una gráfica `temp_analisis.png` que

- (5 puntos) Muestre la evolución temporal de la temperatura.
- (5 puntos) Marque los máximos de la temperatura.
- (10 puntos) Marque los intervalos donde la temperatura crece.

3. Distribución Maxwelliana

Para un gas ideal la probabilidad de que la rapidez de una molécula sea v está determinada por una densidad de probabilidad Maxwelliana

$$\rho = C \frac{v^2}{\sigma^3} \exp\left(-\frac{1}{2} \frac{v^2}{\sigma^2}\right), \quad (1)$$

donde $\sigma = \sqrt{kT/M}$, con k la constante de Boltzmann, T la temperatura absoluta, M la masa de la molécula y C es una constante adimensional tal que $\int_0^\infty \rho(v)dv = 1$.

Escriba en el archivo `maxwell.py` las funciones necesarias para calcular numéricamente como función de la temperatura en el rango $100 < T/K < 1000$ para gases de Helio, Neon y Argon:

- (15 (20) puntos) La posición del pico de esta densidad de probabilidad, es decir, la velocidad v_{\max} para la cual $\frac{d\rho}{dv}|_{v_{\max}} = 0$.
- (15 (20) puntos) La fracción del número total de átomos que tiene una rapidez mayor o igual a 300 m/s, es decir, el cociente $f_{300} = \int_{300}^\infty \rho(v)dv / \int_0^\infty \rho(v)dv$.

Los resultados deben guardarse como dos gráficas: `pico.png` y `fraccion.png`.

4. Función Gamma

(20 (30) puntos) Escriba en el archivo `gamma.py` una función en Python (`def gamma(z)`) que devuelve en una variable `float` el valor de la función gamma para cualquier número real positivo mayor que uno.

La función gamma está definida por la integral

$$\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx. \quad (2)$$

NB: Para calcular integrales indefinidas con los métodos vistos en clase lo más fácil es reescribirla como la suma de dos integrales $\int_0^\infty = \int_0^1 + \int_1^\infty$ de tal manera que la primera integral definida es posible calcularla con la regla de Simpson. La segunda integral indefinida se puede convertir a una integral definida haciendo un cambio de variable para resolverla también con la regla de Simpson.