

La solución a este taller debe subirse por SICUA antes de las 8:00AM del jueves 1 de diciembre del 2016.

Los códigos deben encontrarse en un único repositorio de `github` con el nombre `NombreApellido_hw5`. Por ejemplo yo debería subir crear un repositorio con el nombre `JaimeForero_hw5`. El último commit de ese repositorio debe ser anterior a las 8:00AM del jueves 1 de diciembre del 2016. Solamente se calificarán las tareas que tengan la dirección del repositorio en `sicuplus`.

Adentro de ese repositorio deben existir tres carpetas con los nombres `punto_1`, `punto_2` y `punto_3` por cada uno de los puntos de esta tarea.

Dentro de cada carpeta deben estar los siguientes elementos.

- Un código fuente en C que calcula las cadenas de Markov.
- Un código en Python que lee los datos producidos por el código en C y produce visualizaciones de las densidades de probabilidad y los valores más probables de los parámetros. Para esto deben utilizar la librería `corner` (<http://corner.readthedocs.io/en/latest/pages/quickstart.html>).
- Un `makefile` que sique la estructura lógica del punto para compilar y ejecutar el código en C, ejecutar el código en Python y borrar todos los archivos auxiliares.

1. **Lugar de un sismo** Una fuente sísmica se activo al tiempo $t = 0$ en un lugar desconocido de la superficie de la Tierra. Las ondas sísmicas producida por la explosión se grabaron por una red de seis estaciones sísmicas ubicadas en las siguientes coordenadas (en km): $(x_1, y_1) = (3, 15)$, $(x_2, y_2) = (4, 15)$, $(x_3, y_3) = (5, 15)$, $(x_4, y_4) = (3, 16)$, $(x_5, y_5) = (4, 16)$, $(x_6, y_6) = (5, 16)$. Los tiempos de llegada (en segundos) de las ondas sísmicas fueron $t_{obs,1} = 3,12 \pm \sigma_t$, $t_{obs,2} = 2,98 \pm \sigma_t$, $t_{obs,3} = 2,84 \pm \sigma_t$, $t_{obs,4} = 3,26 \pm \sigma_t$, $t_{obs,5} = 3,12 \pm \sigma_t$, $t_{obs,6} = 2,98 \pm \sigma_t$, donde $\sigma_t = 0,10$.

- (10 puntos) Escriba un programa en C que implemente Metropolis-Hastings para encontrar la distribución de probabilidad de la posición del epicentro dados los parámetros observacionales. Asuma que la Tierra es bidimensional y que la velocidad de propagación de las ondas es constante e igual a 5 km/s.
- (10 puntos) Escriba un programa en Python que grafique la densidad de probabilidad en el plano x, y y muestre los valores más probables junto a sus incertidumbres.
- (5 puntos) Escriba un `makefile` que enlace correctamente todos los pasos anteriores.

2. **Ley de gravitación** Los siguientes datos (tomados de este artículo <https://arxiv.org/abs/0903.5308>) muestran la velocidad y posición instantánea de los planetas en el sistema solar para el primero de abril del 2009.

Table 1. Planet Ephemerides for 2009-Apr-01 00:00:00.0000 (CT^a)

Planet	x (AU)	y (AU)	z (AU)	v_x (AU yr ⁻¹)	v_y (AU yr ⁻¹)	v_z (AU yr ⁻¹)
Mercury	0.324190175	0.090955208	-0.022920510	-4.627851589	10.390063716	1.273504997
Venus	-0.701534590	-0.168809218	0.037947785	1.725066954	-7.205747212	-0.198268558
Earth	-0.982564148	-0.191145980	-0.000014724	1.126784520	-6.187988860	0.000330572
Mars	1.104185888	-0.826097003	-0.044595990	3.260215854	4.524583075	0.014760239
Jupiter	3.266443877	-3.888055863	-0.057015321	2.076140727	1.904040630	-0.054374153
Saturn	-9.218802228	1.788299816	0.335737817	-0.496457364	-2.005021061	0.054667082
Uranus	19.930781147	-2.555241579	-0.267710968	0.172224285	1.357933443	0.002836325
Neptune	24.323085642	-17.606227355	-0.197974999	0.664855006	0.935497207	-0.034716967

^a CT is a coordinate time used in connection with ephemerides. It differs from UTC by about 66 seconds (see http://ssd.jpl.nasa.gov/?horizons_doc#timesys).

Note. — The xyz -coordinate system is defined as follows: the xy -plane is given by the plane of the Earth's orbit at J2000.0, the x -axis is out along the ascending node of the instantaneous plane of the Earth's orbit and the Earth's mean equator at J2000.0, and the z -axis is perpendicular to the xy -plane in the directional (+ or -) sense of Earth's north pole at J2000.0. The origin of the coordinate system is given by the barycenter of the Solar System. One year is defined as 365.25 days.

Figura 1: Posiciones y velocidades instantáneas de planetas con respecto al Sol .

- (10 puntos) Escriba un programa en C que implemente Metropolis-Hastings para encontrar la distribución de probabilidad del logaritmo de la masa del Sol y del exponente que describe como escala la aceleración radial en función de la distancia al Sol. Asuma que las órbitas de los planetas son perfectamente circulares.
 - (10 puntos) Escriba un programa en Python que grafique la densidad de probabilidad en el plano $\log_{10} M_{sol}, \alpha$ y muestre los valores más probables junto a sus incertidumbres.
 - (5 puntos) Escriba un `makefile` que enlace correctamente todos los pasos anteriores.
3. **Poblaciones** Un grupo de biólogos toma datos por casi una década de una población de presas y predadores. Los biólogos intuyen que el número de presas x y el numero de predadores y se describe por un modelo del tipo Lotka-Volterra con las siguientes ecuaciones:

$$\frac{dx}{dt} = x(\alpha - \beta y), \quad (1)$$

$$\frac{dy}{dt} = -y(\gamma - \delta x). \quad (2)$$

donde α, β, γ y δ son parámetros libres que se quieren buscar a partir de los datos experimentales.

- (30 puntos) Escriba un programa en C que implemente Metropolis-Hastings para encontrar la distribución de probabilidad de los parámetros dados los datos observacionales.

`lotka_volterra_obs.dat`¹.

- (15 puntos) Escriba un programa en Python que grafique la densidad de probabilidad de los parámetros y muestre los valores más probables de junto a sus incertidumbres.
- (5 puntos) Escriba un `makefile` que enlace correctamente todos los pasos anteriores.

¹El archivo se encuentra en https://github.com/ComputoCienciasUniandes/MetodosComputacionalesDatos/blob/master/homework/2016-20/hw5/lotka_volterra_obs.dat