

Taller #2 de Métodos Computacionales

FISI 2028, Semestre 2016 - 1

Profesor: Germán Chaparro

16 de Febrero, 2016

Importante

- Los cinco archivos con el código en Python que soluciona esta tarea deben subirse a través de si-cuaplus antes de las 8:30PM del domingo 21 de Febrero como un único archivo zip con el nombre `NombreApellidos_hw3.zip`, por ejemplo yo debería subir un archivo llamado `GermanChaparro_hw2.zip`.
- La nota máxima de este taller es de 100 puntos. Los puntos indicados en cada literal solamente se otorgan si el programa compila y da los resultados esperados según la descripción de cada punto.

1. Conjunto de mandelbrot

(20 puntos) Escriba un programa que prepare en una grid de resolución de 1024×1024 píxeles las primeras n iteraciones del Conjunto de Mandelbrot. Ver: http://en.wikipedia.org/wiki/Mandelbrot_set.

El programa debe poder ejecutarse como

```
python plot_mandelbrot.py n
```

Donde n es el número de iteraciones. El programa debe producir un archivo en pdf llamado `mandelbrot_n.pdf` para guardar la gráfica.

2. Distancias sobre la esfera

(20 puntos) Escriba un programa que calcula la distancia en kilómetros entre dos pares de ciudades dada su longitud y latitud.

El programa debe poder ejecutarse como

```
python ditancia_esfera.py -144.0 +10.0 +20.0 -5.0
```

Donde los dos primeros valores corresponden a la latitud y longitud de la primera ciudad y los dos siguientes a la latitud y longitud de la segunda ciudad. Estas coordenadas deben estar en grados. La distancia debe imprimirse en pantalla.

3. Movimiento Browniano

Vamos a estudiar un caso de movimiento browniano simplificado donde una partícula se mueve en línea recta por una distancia de 1 antes de cambiar aleatoriamente su dirección. Este caso simplificado puede ayudarnos a describir diferentes problemas físicos desde la difusión de fotones en el interior del Sol o la propagación de fotones resonantes en una galaxias.

- a) (15 puntos) Escriba un programa en Python que describa el movimiento browniano de una partícula que comienza en el origen del sistema de coordenadas. El programa debe preparar una gráfica de la distancia hasta el origen como función del número de pasos. El programa debe poder ejecutarse como

```
python browniano_2D.py n
```

Donde n es el número máximo de iteraciones. La gráfica final debe llamarse `browniano_2D_n.pdf`.

- b) (20 puntos) Ahora consideremos un caso simplificado de la difusión de fotones en el interior solar. Los fotones producidos en el centro de la distribución de gas hacen un movimiento browniano de paso 1, para una esfera de tamaño 10^5 . Considere ahora que tiene n fotones. Calcule un histograma con los número de pasos que le toma a los fotones escapar de la distribución de gas.

El programa debe poder ejecutarse como

```
python difusion_solar_central.py n
```

Donde n es el número de fotones a propagar.

El histograma final debe llamarse `histo_difusion_solar_central_n.pdf`.

- c) (25 puntos) Considere ahora el mismo caso anterior pero esta vez los n fotones están distribuidos inicialmente de manera homogénea en la esfera.

El programa debe poder ejecutarse como

```
python difusion_solar_homogenea.py n
```

Donde n es el número de fotones a propagar.

El histograma final debe llamarse `histo_difusion_solar_homogenea_n.pdf`.