

Fecha de Entrega: **Abril 7 de 2015 antes de las 21:59 COT**

## Instrucciones de Entrega

Todo el código fuente y los datos se debe encontrar en un repositorio público en github con un commit final hecho antes de la fecha de entrega. El nombre del repositorio debe ser **CM20151\_HW5\_Apellido1Apellido2**. El link al repositorio lo deben enviar a través de **sicuplus** antes de la fecha/hora límite.

En cada parte del ejercicio se entrega 1/3 de los puntos si el código propuesto es razonable, 1/3 si se puede ejecutar y 1/3 si entrega resultados correctos.

### 1. 70 pt Solucion de la ecuación de Poisson en el espacio de Fourier

La ecuación de Poisson (Ec.1) es una de las más comunes en física, debido a que permite relacionar una densidad con un potencial:

$$\nabla^2 \phi = \rho \quad (1)$$

Dónde  $\phi$  es el potencial y  $\rho$  la densidad. Haciendo la transformada rápida de Fourier de la Ec.1 el potencial gravitacional se puede expresar como:

$$\hat{\phi} = -\hat{\rho} \quad (2)$$

De esta manera, al hallar la densidad en el espacio de fourier, encontramos el potencial. Para encontrar la solución a la ecuación de Poisson, obtenemos la transformada inversa de este potencial  $\hat{\phi}$ .

- (a) 25 pt En el archivo **Serena-Venus.txt** se encuentra información acerca de un sistema de partículas. Cada observación representa una partícula de masa  $M = 1$ , donde las columnas 2, 3, 4 corresponden a la posición en X, Y, Z de dicha partícula respectivamente. De esta manera, es posible construir la densidad en el espacio de fourier  $\hat{\rho}$ . Para esto, es posible construir una matriz de densidades de  $1000 \times 1000 \times 1000$  en la cual se encuentre la densidad en cada punto, y por lo tanto se pueda hallar el potencial gravitacional en cada punto del espacio. El método para encontrar esta densidad es el siguiente:

$$\rho_{i,j,k} = \frac{m}{\Delta x \Delta y \Delta z} \sum_{p=1}^{N_p} W(x_i - x_p) W(y_i - y_p) W(z_i - z_p) \quad (3)$$

Donde  $i, j, k = x, y, z$ ,  $m$  es la masa de las partículas en la celda,  $N_p$  es el número total de partículas y  $W(x)$  está definido por:

$$W(x) = \begin{cases} 1 + x/\Delta x & -\Delta x < x \\ 1 - x/\Delta x & x < \Delta x \\ 0 & \text{dlc} \end{cases}$$

- (b) 15 pt Hacer la transformada inversa de Fourier para el potencial gravitacional  $\hat{\phi}$  para encontrar  $\phi$ .
- (c) 15 pt Derivar el potencial gravitacional y encontrar la fuerza gravitacional en todos los puntos del espacio.
- (d) 10 pt Escribir un código en python que encuentre los mínimos y máximos de la fuerza gravitacional. El código debe realizar una gráfica en donde se representen los puntos donde están las partículas y en colores la fuerza gravitacional. Adicionalmente, en dicha gráfica debe haber contornos en donde se observe claramente las regiones donde la fuerza es máxima y mínima. A manera de guía, se presenta una ilustración de dicha figura sin contornos:

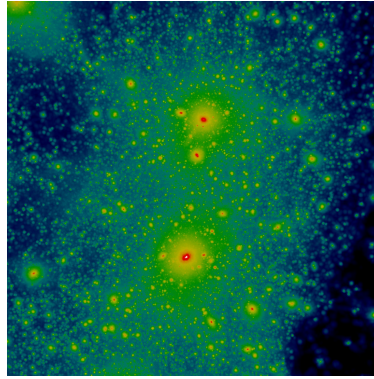


Figura 1: Campo de densidad de Serena-Venus

- (e) 5 pt Haga un Makefile que genere todos los outputs descritos anteriormente en el orden adecuado.

2. 30 pt Integración con el Método del Rechazo

En un documento `.Rmd`,

- (a) 12 pt Implemente una función que calcule una integral utilizando el método del rechazo<sup>1</sup>. La función debe generar una representación visual del cálculo de la integral (ver Fig.2). Para qué tipo de integrales funciona su solución?
- (b) 8 pt Evalúe la integral de la función  $h(x) = [\cos(50x) + \sin(20x)]^2$  sobre  $[0, 1]$  utilizando el método creado anteriormente. Compare su desempeño con algún método de la función `integrate`.
- (c) 10 pt Evalúe la integral de la función  $p(y) = \int_2^\infty \frac{1}{\pi(1+y^2)}$  utilizando el método creado anteriormente. Compare su desempeño con algún método de la función `integrate`.

A manera de ayuda, la siguiente figura es una representación gráfica del método del rechazo:

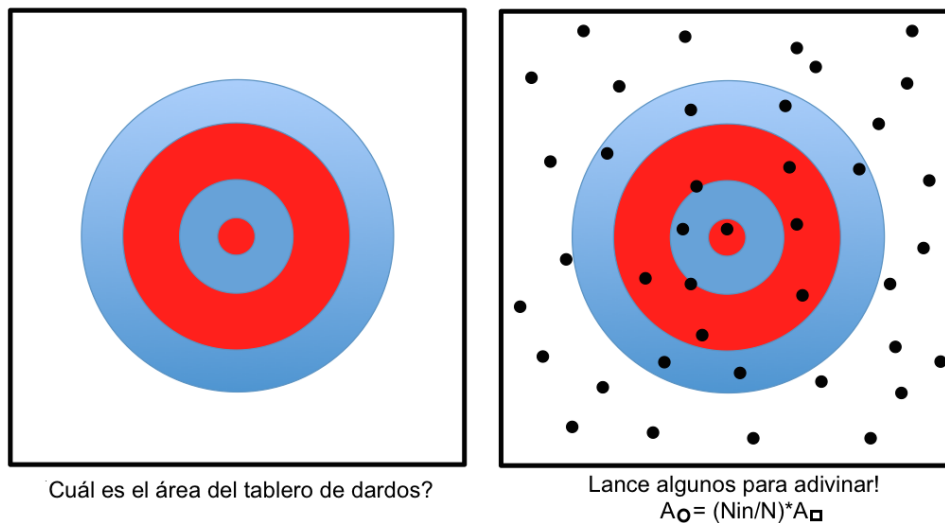


Figura 2: Método de rechazo

---

<sup>1</sup> Descrito acá: [http://en.wikipedia.org/wiki/Rejection\\_sampling](http://en.wikipedia.org/wiki/Rejection_sampling)