

Herramientas Computacionales - Tarea 6

SEMANA 8 - POO EN PYTHON
2017-II

Los dos archivos (código fuente de python e imagenl) deben subirse a SicuaPlus en un único archivo .zip con el nombre del estudiante en el formato `NombreApellido.zip` antes que termine la clase.

Se busca sistematizar el proceso de predicción de trayectorias en un laboratorio que analiza imágenes provenientes de experimentos con cámaras de burbujas para caracterizar partículas elementales. En este experimento se pone un líquido transparente que se encuentra justo debajo del punto de ebullición dentro de un solenoide que genera un campo magnético uniforme y se toman imágenes de los caminos de burbujas que las partículas cargadas que dejan al pasar. Esto resulta en imágenes como la siguiente: .



Figura 1: imagen tomada de http://cerncourier.com/cws/article/cern/28742/1/cernparticles1_11-02

Para lograr predecir las trayectorias eficientemente se quiere crear una clase en Python llamada `Particula`. Esta, eventualmente será utilizada para resolver las ecuaciones de movimiento de la partícula bajo cuestión, las cuales están dadas por la segunda ley de Newton teniendo en cuenta la fuerza de Lorentz:

$$m\mathbf{a} = q\mathbf{v} \times \mathbf{B} + q\mathbf{E}, \quad (1)$$

donde el campo eléctrico \mathbf{E} viene de la diferencia de potencial que acelera a las partículas incidentes en primer lugar.

Para este ejercicio se tomará en cuenta solo el movimiento en el plano xy de la partícula dado un campo magnético uniforme $\mathbf{B} = B\hat{z}$ y un campo eléctrico uniforme $\mathbf{E} = E_x\hat{x} + E_y\hat{y}$.

1. (0.5 puntos) Por favor comenten todo su código. Específicamente, comentar el proposito de cada variable importante, clase, función y ejecución cuyo significado no sea evidente. A cada función deben comentarle el significado de sus parámetros de entrada y sus retornos. A cada iteración deben comentarle su objetivo. Se acepta declarar variables con nombre evidente como `number` en lugar de comentar su significado.
2. (2.0 puntos) Dentro del archivo `particula.py` crear una clase cuyas instancias serán objetos de tipo `Particula` los cuales tendrán atributos
 - Posición en x : x
 - Velocidad en x : V_x
 - Posición en y : y
 - Velocidad en y : V_y
 - Carga q
 - Masa m

Adicionalmente, la clase `Particula` debe tener los métodos:

- `__init__`: Carga las condiciones iniciales del objeto.
 - `CalculaFuerzaX`: Calcula la fuerza de Lorentz que depende de la velocidad en y , V_y a cada instante $F = qV_yB + qE_x$. Se pasan como parámetros B y E_x .
 - `CalculaFuerzaY`: Calcula la fuerza de Lorentz que depende de la velocidad en x , V_x a cada instante $F = -qV_xB + qE_y$. Se pasan como parámetros B y E_y .
 - `Muevete`: Utilizando el método de Euler, actualiza la posición y la velocidad en x y y en cada Δt . Se pasan como parámetro Δt .
 - `Imprime`: Imprime la posición x y y de la partícula.
3. (2.5 puntos) El script debe iniciarse con una masa de 1 en el punto (2,0) con velocidad (0,5) y carga -1. Los campos aplicados van a ser $E_x = 5$, $E_y = 5$ y $B = 10$. Se simula la trayectoria por 10.0s de la misma manera que en el ejemplo que se dio en el video de la semana. Los datos pueden redirigirse hacia un archivo `trayectoria.dat` y estos deben graficarse en un archivo `trayectoria.png`.