

## Partícula Cargada

**Planteamiento del problema:** Modele un anillo cargado uniformemente como un conjunto de cargas puntuales colocadas en un círculo. Coloque el anillo en el plano  $yz$  de manera que su eje coincida con el eje de las  $x$ . Asignele una carga total  $Q = 50 \times 10^{-9} \text{ C}$ .

1. Para la evolución utilice RK4 clásico.
2. Libera un electrón desde el reposo desde una posición inicial de  $\vec{r}=(0.15,0,0)\text{m}$  y modela su movimiento. Considere que la única interacción relevante es la fuerza eléctrica.
3. Utilice la segunda ley de Newton para plantear la ecuación diferencial que rige la dinámica del sistema.
4. Inicia con un  $\Delta t = 1 \times 10^{-17} \text{ s}$
5. Varíe  $\Delta t$  y el número de cargas puntuales que conforman el anillo para asegurar que tu cálculo es preciso.
6. Explora con diferentes posiciones iniciales que no se encuentren sobre el eje del anillo. Modifique también la velocidad inicial del movimiento. Describa las trayectorias obtenidas.
7. Calcule en sus simulaciones, la energía total del sistema como función del tiempo. Construye las gráficas de la energía cinética (K), potencial (U) y (K+U) vs tiempo, para esta órbita, ¿Puedes hacer una descripción del flujo de energía en el sistema utilizando estas gráficas?
8. Para cada conjunto de condiciones iniciales utilizadas, acompañe sus resultados con las gráficas donde se observen las trayectorias obtenidas.

## Análisis Preliminar

Primero iniciamos con la colocación de las partículas en el anillo, para ello tomamos como base la idea de las coordenadas polares, tomando en cuenta que las cargas están distribuidas uniformemente a lo largo del anillo, es decir, el ángulo que separa cada partícula en el anillo es constante y depende de la cantidad de carga que se desee utilizar. De esta forma la expresión base es la siguiente:

$$P_i = (0, r \cos \theta, r \sin \theta) \quad \text{Posición de la } i\text{-ésima partícula en el anillo}$$

Para implementar lo anterior, codificaré una subrutina en Fortran la cual tendrá el nombre de **PosaXYZ**.

## Codificación: PosaXYZ

### Especificación:

- *Entradas:* n0, r0, posAn0
  - # n0 : Número de cargas
  - # r0 : Radio del anillo
  - # posAn0 : Arreglo de posición de cargas
- *Salidas:* posAn0
  - # posAn0 : Arreglo de posición de cargas
- *Auxiliares:* La, h, x,y,z, i
  - # La: Longitud de arco (Separación entre cada carga)
  - # h: Ancho de paso
  - # x,y,z: Posición de las cargas en el espacio
  - # i: Contador

### Algoritmo:

1. INICIO
2. Asignar valores de entrada: La, x, y, z

3. Calcular el ancho del paso:

$$h = \frac{2\pi}{n0}$$

4. Iniciar el Loop: Mientras  $i \leq N$  calcular:

$$x = 0, \quad y = r \cdot \cos(LA), \quad z = r \cdot \sin(LA)$$

$$\text{posAn0} = [0, y, z]$$

$$LA = i \cdot h$$

5. Regresar el arreglo de posiciones: posAn0

6. FIN

Se menciona en el planteamiento del problema que solo se tome en cuenta la fuerza eléctrica para modelar el movimiento de la partícula. La esencia de esta sección es utilizar el campo eléctrico generado por el anillo en la  $i$ -ésima posición del electrón. Posteriormente utilizar el principio de superposición de fuerzas para conocer el campo eléctrico neto sobre la carga de prueba en un instante de tiempo determinado.

Para implementar lo anterior, codificaré una subrutina en Fortran la cual tendrá el nombre de **IntPart**.

### Codificación: IntPart

#### Especificación:

- *Entradas:* posAn, posPar, n, r, Q, FT

# n : Número de cargas

# r : Radio del anillo

# Q : Carga total del anillo

# FT : Fuerza total

# posAn : Arreglo de posición de cargas

# posPar : Vector de posición de la carga de prueba

- *Salidas:*E

# E : Campo eléctrico total

- *Auxiliares:*pPun,rParPun,dq,Ce,FC,magr i

# pPun: Vector posicion de la particula en el anillo

# rParPun:Vector distancia entre la carga de prueba y la  $i$ -ésima carga en el anillo

# dq: Carga de cada partícula en el anillo

# Ce: Carga del electrón

# FC: Fuerza de Coulomb

# i: Contador

#### Algoritmo:

1. INICIO

2. Asignar valores de entrada: dq,Ce

3. Iniciar el Loop: Mientras  $i \leq N$  calcular:

$$\text{pPun} = [\text{posAn}(i, 1), \text{posAn}(i, 2), \text{posAn}(i, 3)] \quad (\text{Vector en R3})$$

$$\text{rParPun} = \text{posPar} - \text{pPun} \quad (\text{Vector en R3})$$

$$dE = E(\text{dq}, \text{magr}) \cdot \text{uParPun} \quad (\text{Vector en R3})$$

$$E = E + dE \quad (\text{Vector en R3})$$

4. Regresar el campo eléctrico: E

5. FIN

La siguiente sección del código tendrá la función de calcular, actualizar la posición de la partícula y con ello calcular la energía cinética, potencial y total del sistema. Para deducir las ecuaciones de movimiento utilizamos la segunda ley de newton:

$$\sum \vec{F} = m \vec{a} \implies \vec{F}_E = q \vec{E} = m \vec{a}$$

Teniendo en cuenta que el elemento en movimiento es el electrón con masa  $m_e$  y carga  $q_e$ :

$$m_e \vec{a}_e = q_e \vec{E} \implies \vec{a}_e(t) = \frac{d\vec{v}}{dt} = \frac{q_e \vec{E}}{m_e}$$

Así, para calcular la posición del electrón al transcurrir el tiempo tenemos:

$$\vec{v}(t + \Delta t) = \frac{d\vec{x}}{dt} = \vec{v}(t) + \vec{a}_e(t) \cdot \Delta t$$

$$\vec{x}(t + \Delta t) = \vec{x}(t) + \vec{v}(t) \cdot \Delta t$$

Para implementar lo anterior, codificaré una subrutina en Fortran la cual tendrá el nombre de **NewC**.

## Resultados

Para este ejercicio los resultados obtenidos fueron los siguientes:  
Elementos de simulación 1:

1. Posicion inicial del electrón:  $x = 0.15$  m,  $y = 0.0$  m,  $z = 0.0$  m
2. Se libera del reposo
3. Ancho de paso:  $\Delta t = 1 \times 10^{-17}$  s
4. Tiempo total de simulación:  $t = 1 \times 10^{-5}$  s

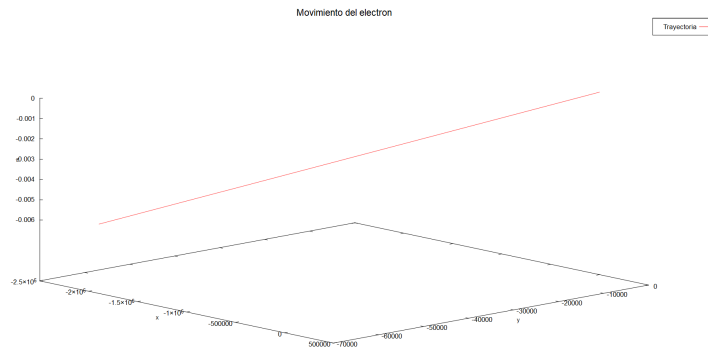


Figura 1: Gráfica de trayectoria del electrón en el tiempo.

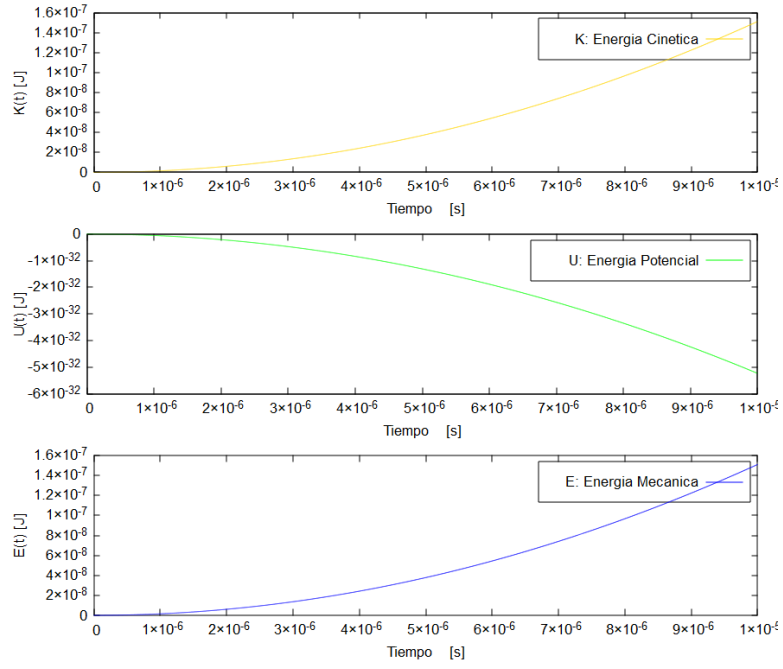


Figura 2: Gráficas de U, K y E en el tiempo.

Para la simulación 2 tomaremos los siguientes elementos:

1. Posición inicial del electrón:  $x = 10.0$  m,  $y = -5.0$  m,  $z = 10.0$  m
2. Velocidad inicial:  $x = 300.0$  m/s,  $y = 250.0$  m/s,  $z = -50.0$  m/s
3. Ancho de paso:  $\Delta t = 1 \times 10^{-17}$  s
4. Tiempo total de simulación:  $t = 1 \times 10^{-5}$  s

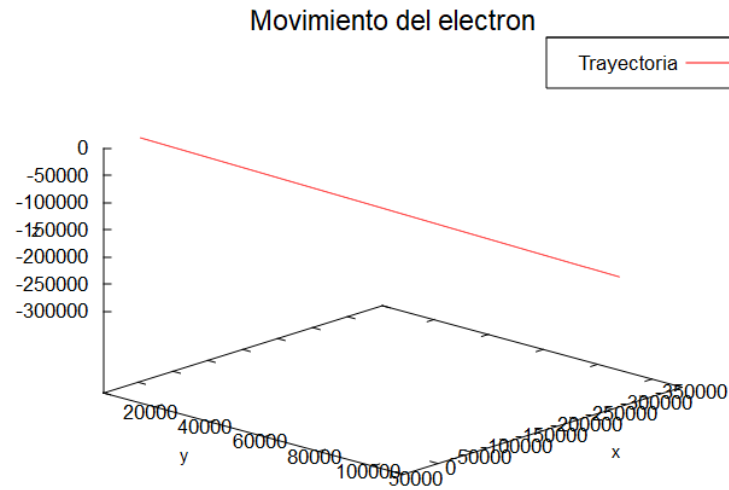


Figura 3: Gráfica de trayectoria del electrón en el tiempo.

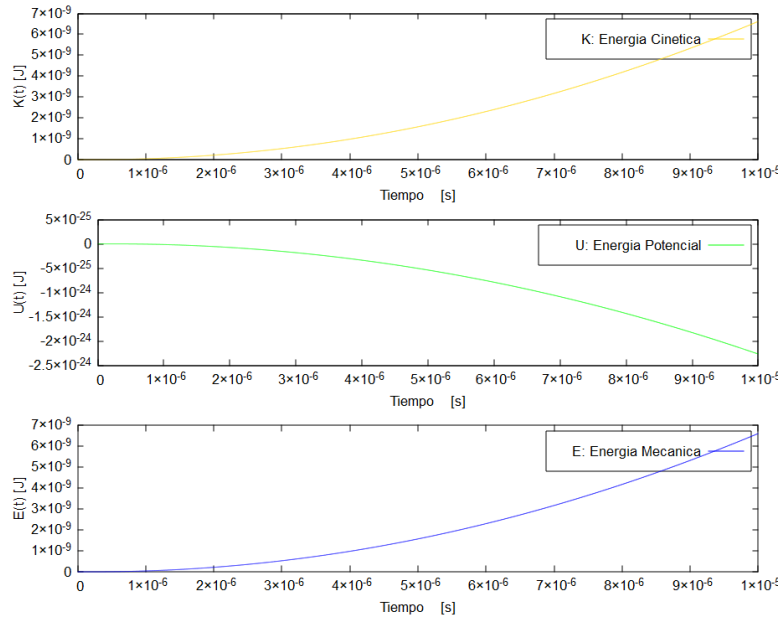


Figura 4: Gráficas de  $U$ ,  $K$  y  $E$  en el tiempo.

Para la simulación 3 tomaremos los siguientes elementos:

1. Posicion inicial del electrón:  $x = -21.0$  m,  $y = 1.0$  m,  $z = -10.0$  m
2. Velocidad inicial:  $x = 3000.0$  m/s,  $y = 2500.0$  m/s,  $z = -500.0$  m/s
3. Ancho de paso:  $\Delta t = 1 \times 10^{-17}$  s
4. Tiempo total de simulación:  $t = 1 \times 10^{-5}$  s

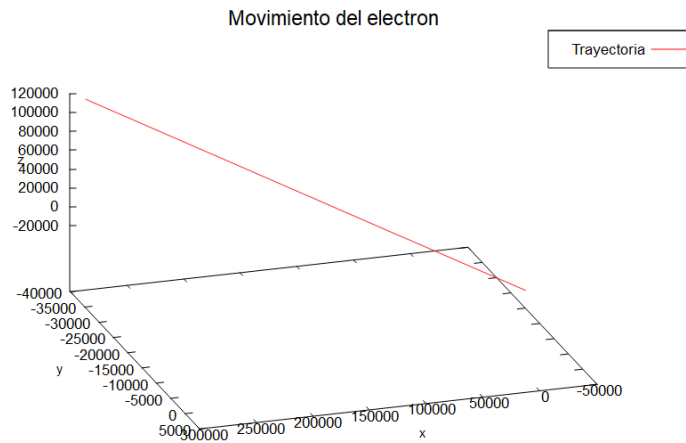


Figura 5: Gráfica de trayectoria del electrón en el tiempo.

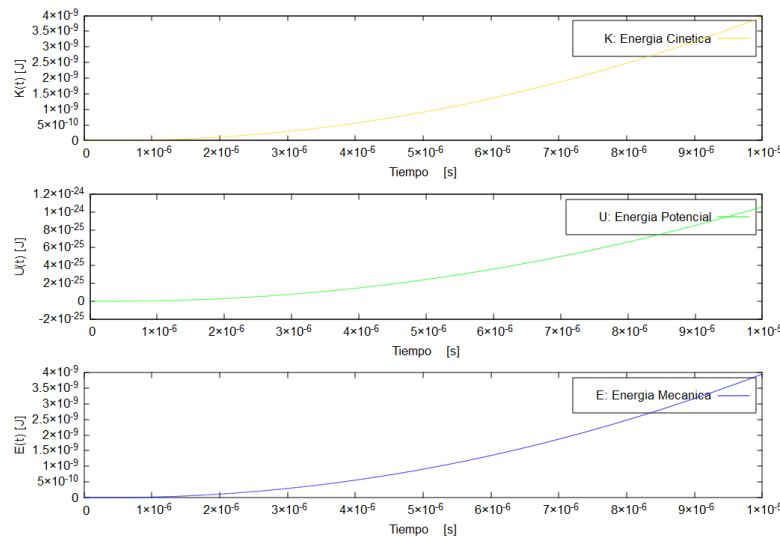


Figura 6: Gráficas de  $U$ ,  $K$  y  $E$  en el tiempo.

En todas las gráficas el comportamiento del electrón es similar, en cuanto a que la trayectoria descrita es una línea recta en el espacio. Lo cual claramente no es señal de una buena modelación del fenómeno.

De igual forma con el comportamiento de las gráficas de energía contra tiempo, el comportamiento es muy similar, solo que a es modificado por el movimiento en el eje  $z$  de cada simulación, de forma que si el electrón se mueve en positivo en este eje la energía potencial es creciente y si el electrón se mueve en negativo en este eje la energía potencial es decreciente. Algo que me deja desconcertado, pues no es lo que yo esperaba.

## Conclusión

En esta ocasión, debo de decir que la práctica fue completamente fallida, pues además de que perdí mucho tiempo en la realización de la misma, los resultados no fueron los esperados.

El ejercicio como tal no me parece que sea difícil, creo que mi planteamiento es correcto, pero al momento de implementar el método de RK4 encontré mi mayor obstáculo pues al implementarlo errores de Overflow se presentaban, algo que me frustra bastante, pues no pude dar con la sección del código que arrojaba este error. Este error no solo lo obtuve programando el lenguaje Fortran, también lo obtuve usando lenguaje Python. La única forma que pude encontrar para resolver de esta situación fue usar el método de Euler-Cromer, pero los resultados no fueron los esperados. Considero que mis resultados son erróneos ya que creo yo que lo que pasaría en el caso de la primer simulación es que el electrón oscila sobre el eje  $x$  o orbitaría de alguna forma al anillo, pero mi programa no arroja resultados de este tipo.

No toda la experiencia de realizar esta práctica es mala, pues creo que aprendí una gran lección, el tiempo que debo emplear al resolver algún problema debe ser medido y debo de tomarlo en cuenta siempre.