

Programación en Lenguaje Java

Práctica 3.2. Movimiento de una partícula cerca de una placa cargada



Michael González Harbour
Mario Aldea Rivas

Departamento de Matemáticas,
Estadística y Computación

Este tema se publica bajo Licencia:

[Creative Commons BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

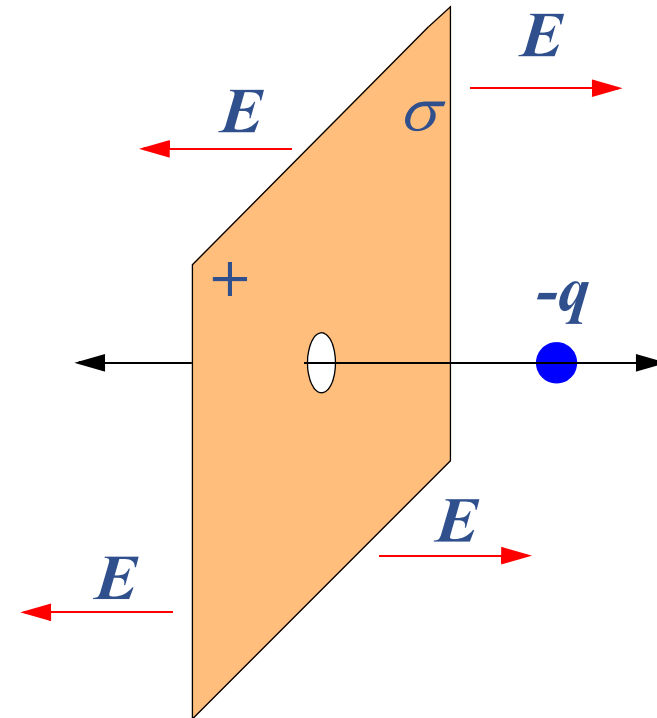
Práctica 3-2: Movimiento de una partícula cerca de una placa cargada

Objetivos: Practicar con la instrucción condicional y con los Strings

Simularemos el movimiento de oscilación no lineal de una partícula de masa m cargada con una carga negativa $-q$, en las proximidades de una placa indefinida cargada con una densidad de carga positiva σ

En la placa se ha hecho un pequeño agujero para que pueda pasar la partícula cargada

El movimiento se produce en la dirección horizontal



Ecuaciones del movimiento

La fuerza del campo eléctrico E de la placa sobre la partícula es constante en módulo

$$F = \frac{\sigma}{2\varepsilon_0}q$$

siendo $\varepsilon_0 = 8.85 \cdot 10^{-12} \text{ F/m}$ la permitividad del vacío (faradios por metro)

La fuerza es de sentido contrario al campo eléctrico:

- cuando la partícula está a la derecha $x > 0$ de la placa, la fuerza es negativa $F < 0$
- cuando la partícula está a la izquierda $x < 0$ de la placa, la fuerza es positiva $F > 0$

Ecuaciones del movimiento

Partiendo desde una posición inicial $x=A$, y velocidad inicial nula, el movimiento sigue tres etapas:

- *Primera etapa*: desde $x=A$, moviéndose hacia la izquierda hasta la posición $x=0$

$$v = -\frac{F}{m}t \quad x = A - \frac{1}{2}\frac{F}{m}t^2$$

- esta primera etapa acaba en $t=t_0$ y con $v=-v_0$

$$v_0 = \sqrt{\frac{2FA}{m}} \quad t_0 = \sqrt{\frac{2mA}{F}}$$

Ecuaciones del movimiento

- **Segunda etapa:** desde $x=0$, siguiendo hacia la izquierda al valor de $x=-A$, siguiendo luego hacia la derecha hasta que x vuelve a ser cero

$$v = -v_0 + \frac{F}{m}(t - t_0) \quad x = -v_0(t - t_0) + \frac{1}{2}\frac{F}{m}(t - t_0)^2$$

- al final de esta etapa se regresa al origen en el instante $t=3\cdot t_0$, con velocidad v_0
- **Tercera etapa:** desde que sale del origen $x=0$ hacia la derecha, hasta que regresa a la posición inicial $x=A$

$$v = v_0 - \frac{F}{m}(t - 3t_0) \quad x = v_0(t - 3t_0) - \frac{1}{2}\frac{F}{m}(t - 3t_0)^2$$

- al final, llega a la posición inicial $x=A$ en el instante $4\cdot t_0$, con velocidad nula $v=0$

Clase ParticulaCargada

Queremos crear una clase que permita resolver estas ecuaciones, de acuerdo al diagrama de clases

Atributos:

- **q**: valor absoluto de carga de la partícula en culombios
- **sigma**: densidad de carga de la placa en culombios por metro cuadrado
- **m**: masa de la partícula en Kg
- **posIni**: posición inicial, A , en metros
- **fuerza**: módulo de la fuerza del campo eléctrico, en Newtons

ParticulaCargada
-double q -double sigma -double m -double posIni -double fuerza
+ParticulaCargada (double q, double sigma, String particula) +double periodo() +double velocidadOrigen() +double posicion(double t) +double velocidad(double t)

Clase ParticulaCargada

Métodos:

- *constructor*: pone los valores iniciales de los atributos
 - *q* y *sigma* toman sus valores de los parámetros del mismo nombre
 - *m* y *posIni* se sacan del parámetro *particula* que es un String que contiene dos números separados por uno o varios espacios en blanco; el primero es la masa de la partícula en Kg y el segundo la posición inicial, en metros; si no hay espacios en blanco ambos atributos se ponen a valor `Double.NaN` para indicar el error
 - *fuerza* se calcula con la fórmula de la página 2
- *periodo*: retorna el periodo del movimiento oscilatorio, que es $4t_0$
- *velocidadOrigen*: retorna v_0
- *posicion*: retorna la posición x , dado el tiempo t
- *velocidad*: retorna la velocidad v , dado el tiempo t

Clase ParticulaCargada

Nota: en los métodos `posicion` y `velocidad`, como el movimiento es periódico, si $t > periodo$, quitarle un número entero de periodos para dejarlo en el intervalo $[0, periodo)$

Usaremos unidades del SI: culombios para la carga, newtons para la fuerza, metros, segundos, ...

Notas sobre los strings

Nota: Disponemos de los siguientes métodos de la clase `String`, teniendo en cuenta que los caracteres se numeran empezando por cero

Operación	Descripción
<code>String substring (int ini, int fin)</code>	Retorna el trozo del string que está comprendido entre los caracteres ini (incluido) a fin (excluido)
<code>int length()</code>	Retorna el número de caracteres
<code>int indexOf(char c)</code>	Buscar el carácter c en el String; si lo encuentra retorna la posición en que está; si no, retorna -1
<code>boolean equals(String s)</code>	Compara el string con s; retorna true si sus contenidos son iguales y false si no
<code>String trim()</code>	Retorna una copia del string original omitiendo los espacios en blanco iniciales y finales

Notas sobre los strings

Nota: Obsérvese que para comparar dos strings no debe usarse el operador de igualdad:

```
if (s1==s2) ... // mal (aunque compila)
if (s1.equals(s2)) ... // bien
```

Nota: Para convertir un texto a número real usar el siguiente método de la clase `Double`

Operación	Descripción
<code>static double parseDouble(String s)</code>	Convierte el texto <code>s</code> que se le pasa como parámetro a número real y lo retorna

Por ejemplo, si `s1` es un `String`:

```
double p=Double.parseDouble(s1);
```

Parte obligatoria

Realización:

- Escribir la clase `ParticulaCargada` y probar sus métodos
- Escribir en otra clase un programa principal que haga:
 - Crea un objeto de la clase `ParticulaCargada` con una densidad de carga $\sigma=150 \mu\text{C}/\text{m}^2$, una partícula de carga y masa igual a la del electrón, y posición inicial $A=0.001 \text{ m}$
 - Muestra en pantalla la siguiente información:
 - periodo ($4t_0$) y velocidad (v_0) en el origen
 - posición para $t=0.5t_0, t=t_0, t=1.5t_0, t=2t_0, t=2.5t_0, t=3t_0, t=3.5t_0, t=4t_0, t=4.5t_0, t=5.5t_0$
 - velocidad para esos mismos instantes

Entregar:

- El código java de las dos clases
- Los resultados de la ejecución del programa

Parte avanzada: gráfica

Escribir otro programa principal que muestre una gráfica de los resultados de la posición en función del tiempo (expresado como número de periodos), usando un objeto de la clase `Grafica`

- Hacerlo para 1.5 periodos, usando para ese intervalo 30 puntos

Intentar usar los métodos de la clase `Grafica` para poner símbolos en cada punto, elegir un color para la gráfica, y poner leyenda a la gráfica con `ponTitulo`.

Si se desea, superponer en la misma gráfica una segunda gráfica para las velocidades

- Normalizar los valores de la velocidad dividiéndolos todos por v_0 y por 1000, para que las gráficas de la posición y velocidad sean comparables