

# Simulación numérica de un Ciclotrón

Nicolas Maldonado Baracaldo

201423809

Facultad de Ciencias, Universidad de Los Andes

Departamento de Física

Aceleradores de Partículas y Sus Aplicaciones

Profesor Bernardo Gómez Moreno, Dr. rer. nat.

29 de Mayo, 2019

## 1. Introducción

El ciclotrón es un tipo de acelerador de partículas que acelera partículas cargadas mediante un campo eléctrico de radiofrecuencia (RF) mientras estas realizan una trayectoria en espiral, desde el centro hacia afuera, bajo la influencia de un campo magnético constante.

El desarrollo del ciclotrón inició en 1929 cuando Ernest O. Lawrence (1901–1958) quiso aplicar los hallazgos de Rolf Widerøe (1902–1996), la aceleración de partículas usando campos RF, de manera sucesiva para alcanzar energías tan altas como las alcanzadas en el momento por otras máquinas de muy alto voltaje. Para evitar tener que usar una larga cadena de electrodos, Lawrence propuso usar un solo electrodo, curvando el haz con un campo magnético para lograr trayectorias circulares de manera que el haz pasara por el electrodo en cada media vuelta.

Entre 1930 y 1932, Lawrence y su equipo en la Universidad de California, Berkeley construyeron los primeros ciclotrones, de 4" y luego 11" de diámetro, y lograron aceleraciones de más de 1 MeV, algo nunca antes visto. En 1932, ya trabajando en la construcción de una máquina de 27", Lawrence patentó su invento, por el cual recibió el premio Nobel en física en 1939.

La era del ciclotrón finalmente llegó a su fin en 1945, cuando se descubrió que estas máquinas estaban limitadas por los efectos relativis-

tas que surgían al alcanzar una cierta energía. En este momento se estaba construyendo un ciclotrón de 184", el cual se modificó de manera que se actualizara el campo RF a medida que las partículas ganaban energía, manteniendo así la máquina sincronizada. Éste sería el primer sincrociclotrón, máquinas que eventualmente también fueron remplazadas por los sincrotrones, los aceleradores más modernos actualmente.

El principio de funcionamiento del ciclotrón se desprende del movimiento circular de una partícula cargada bajo la influencia de un campo magnético y el hecho de que el periodo de este movimiento es independiente de la velocidad de la partícula (derivación 1). Así, a medida que se aceleran las partículas dentro de la máquina su radio aumenta de manera que el periodo se conserve, dando lugar a un movimiento en espiral.

$$\begin{aligned} F_B &= q\mathbf{v} \times \mathbf{B} \\ F_c &= \frac{mv^2}{r} \\ qvB &= \frac{mv^2}{r} \\ r &= \frac{mv}{qB} \\ T &= \frac{2\pi}{\omega} = \frac{2\pi r}{v} = \frac{2\pi mv}{qBv} \\ T &= \frac{2\pi m}{qB} \end{aligned} \tag{1}$$

Como se cuenta con un periodo constante, es fácil fijar una frecuencia para el campo RF de

manera que en cada paso de las partículas por el electrodo éstas sean aceleradas y no frenadas, la llamada frecuencia de ciclotrón (ecuación 2).

$$f_C = \frac{qB}{2\pi m} \quad (2)$$

Habiendo fijado esta frecuencia para el campo RF, se asegura que al pasar las partículas por el electrodo, cada medio periodo, éstas estén sometidas a una fuerza de Lorentz por la acción de ambos campos, eléctrico y magnético (ecuación 3), la cual continúa curvando la trayectoria (componente magnética) y acelera las partículas (componente eléctrica). Sin embargo, a pesar de que la energía que ganan las partículas viene del campo eléctrico, se encuentra que la energía de salida del haz está únicamente determinada por la magnitud del campo magnético y por el radio máximo,  $R$ , del recorrido (derivación 4).

$$F_{Lorentz} = q\mathbf{E} + q\mathbf{v} \times \mathbf{B} \quad (3)$$

$$\begin{aligned} F_B &= q\mathbf{v} \times \mathbf{B} \\ F_c &= \frac{mv^2}{r} \\ qvB &= \frac{mv^2}{R} \\ v &= \frac{qBR}{m} \\ K &= \frac{1}{2}mv^2 = \frac{q^2B^2R^2}{2m} \end{aligned} \quad (4)$$

El diseño básico de un ciclotrón (figura 1<sup>1</sup>) consiste en un electroimán con sus polos situados uno arriba y uno abajo de un disco, generando un campo magnético uniforme al interior de éste. El disco en sí está dividido en dos mitades semi-circulares llamadas Ds (o frecuentemente "dees" en inglés) separadas por una pequeña brecha entre sus segmentos lineares, espacio en el cual se encuentra un campo eléctrico de radiofrecuencia y espacialmente uniforme creado al conectar cada D a una terminal de un oscilador con la frecuencia adecuada (ecuación 2).

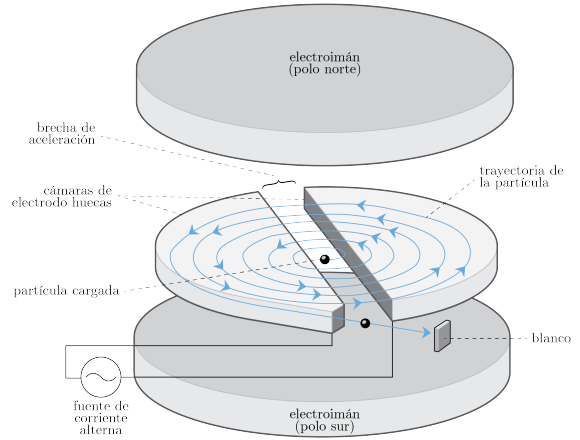


Figura 1: Diseño básico de un ciclotrón

## 2. Método

Se escribió un código que permitiera simular numéricamente el funcionamiento de un ciclotrón, tomando como valores de entrada los parametros de la máquina misma, los parametros de la partícula (protón), y algunos parametros para la simulación —la posición y velocidad iniciales de la partícula, el tiempo de operación a simular, y el número total de puntos a tomar.

Se definen dos funciones para usar durante la ejecución de la simulación. La primera,  $F_{Lorentz}()$ , calcula la fuerza de Lorentz correspondiente a la posición, la velocidad, y el tiempo ingresados como parametros, si bien la velocidad es un parametro necesario en el cálculo de esta fuerza (ecuación 3), y el tiempo es el parametro que determina la amplitud del campo eléctrico según la frecuencia de ciclotrón (ecuación 2), la posición se ingresa como el parametro en dos funciones de Heaviside que sirven para incluir o excluir, según sea el caso, la componente eléctrica de la fuerza, puesto que ésta solo la experimentarán las partículas cuando pasan entre las Ds.

La segunda función,  $RK4()$ , utiliza el más utilizado método iterativo de Runge-Kutta, el de cuarto orden, para aproximar las soluciones

<sup>1</sup>Adaptada de El-Saftaway, *Regulating the performance parameters of accelerated particles* por María Alejandra Sánchez.

de una ecuación diferencial ordinaria de segundo orden con discretización temporal. Puesto que el método puede usarse para aproximar soluciones a sistemas de la forma (5), se reconoce que se puede aplicar a la simulación numérica de un ciclotrón, teniendo en cuenta que  $\mathbf{r}(t)$ , las posiciones, pueden hallarse a partir de  $\mathbf{v}(t, \mathbf{r}) = \dot{\mathbf{r}}(t, \mathbf{r})$ , las velocidades, y estas a su vez pueden hallarse a partir de  $\mathbf{a}(t, \mathbf{r}, \mathbf{v}) = \dot{\mathbf{v}}(t, \mathbf{r}, \mathbf{v}) = \mathbf{F}_{\text{Lorentz}}(t, \mathbf{r}, \mathbf{v})/m$ , las aceleraciones calculadas directamente de las fuerzas de Lorentz que apliquen en cada punto de la trayectoria, todo con pasos de tamaño  $dt$ .

$$\begin{aligned} \dot{y}_1 &= y_2 \\ \dot{y}_2 &= f(t, y_1, y_2) \end{aligned} \quad (5)$$

Así pues, el código toma los parámetros de

entrada para inicializar la posición, la velocidad, y la aceleración, crea un arreglo para la discretización temporal con  $N$  elementos temporales en el rango  $[0, t_{\text{max}}]$  con separación  $dt$ , y procede a iterar sobre ellos. En cada iteración se actualizan  $r$ ,  $v$ , y  $a$ , a partir de los cuales se van guardando en sus respectivos arreglos las posiciones en  $x$  y en  $y$ , los momentos en  $x$  y en  $y$ , y los radios de la trayectoria calculados a partir de la posición,  $r_{xy}$  (método geométrico) y del momento,  $r_p$  (método válido para ciclotrones según ecuación 6).

$$r(t) = \frac{\|\mathbf{p}(t)\|}{qB} \quad (6)$$

Finalmente se usan los arreglos creados para generar una gráfica de  $y$  vs.  $x$ , mostrando la trayectoria de las partículas, así como gráficas de  $x$  vs.  $p_x$ ,  $y$  vs.  $p_y$ , y  $r_{xy}$  vs.  $r_p$ .

### 3. Código

A continuación se muestra en su totalidad el código empleado, escrito en Python 3.6:

```

1 ##### Numerical Simulation of a Cyclotron #####
2 ### Nicolas Maldonado Baracaldo
3 ### 201423809
4 #####
5
6 import numpy as np
7 import matplotlib.pyplot as plt
8
9 ### Cyclotron parameters: ###
10 E = np.array([6000000,0,0]) #Electric field
11 B = np.array([0,0,-1.64]) #Magnetic field
12 R = 0.923 #Radius
13 d = 0.120 #Distance between dees
14
15 ### Particle parameters: ###
16 q = 1.6021766*10**-19 #Particle charge
17 m = 1.6726219*10**-27 #Particle mass
18
19 ### Simulation parameters: ###
20 r_0 = np.array([0,0.05,0]) #Initial position
21 v_0 = np.array([100,0,0]) #Initial velocity
22 t_max = 1.3*10**-6 #Simulation running time
23 N = 10000 #Number of samples for simulation
24
25 ### Function to calculate Lorentz force: ###
26 def F_Lorentz(r, v, t):
27     return q*E*np.cos((q*B[2]/m)*t)*np.heaviside(r[0]+d/2, 1)*np.heaviside(-r[0]+d/2, 1)\
28         + q*np.cross(v, B)
29 #end of F_Lorentz() function
30
31 ### Function to update r, v, a via 4th order Runge-Kutta: ###
32 def RK4(t):
33     global r
34     global v

```

```

35     global a
36     global dt
37
38     #First step:
39     k1 = v
40     l1 = a
41
42     #Second step:
43     k2 = v + dt/2*l1
44     l2 = F.Lorentz(r+dt/2*k1, k2, t+dt/2)/m
45
46     #Third step:
47     k3 = v + dt/2*l2
48     l3 = F.Lorentz(r+dt/2*k2, k3, t+dt/2)/m
49
50     #Fourth step:
51     k4 = v + dt*l3
52     l4 = F.Lorentz(r+dt*k3, k4, t+dt)/m
53
54     #Averages:
55     k = (k1 + 2*k2 + 2*k3 + k4)/6
56     l = (l1 + 2*l2 + 2*l3 + l4)/6
57
58     #Update acceleration:
59     a = F.Lorentz(r, v, t+dt)/m
60     #Update velocity
61     v = v + dt*l
62     #Update position
63     r = r + dt*k
64 #end of RK4() function
65
66 t = np.linspace(0,t_max,N) #Times array
67 dt = t[1] - t[0] #Time step
68 r = r_0 #Set initial position
69 v = v_0 #Set initial velocity
70 a = F.Lorentz(r, v, 0)/m #Set initial acceleration
71
72 x = np.array([r[0]]) #X positions array
73 y = np.array([r[1]]) #Y positions array
74
75 p_x = np.array([m*v[0]/(5.36*10**-22)]) #X momenta array
76 p_y = np.array([m*v[1]/(5.36*10**-22)]) #Y momenta array
77
78 r_xy = np.array([np.sqrt(r[0]**2 + r[1]**2)]) #Position-based radii array
79 r_p = np.array([np.sqrt((m*v[0])**2 + (m*v[1])**2)/(-q*B[2])]) #Momentum-based radii array
80
81 ### Loop over times ###
82 for i in range(1,t.size):
83     #Update variables:
84     RK4(t[i-1])
85
86     #Store positions:
87     x = np.append(x, [r[0]])
88     y = np.append(y, [r[1]])
89
90     #Store momenta:
91     p_x = np.append(p_x, [m*v[0]/(5.36*10**-22)])
92     p_y = np.append(p_y, [m*v[1]/(5.36*10**-22)])
93
94     #Store radii:
95     r_xy = np.append(r_xy, [np.sqrt(r[0]**2 + r[1]**2)])
96     r_p = np.append(r_p, [np.sqrt((m*v[0])**2 + (m*v[1])**2)/(-q*B[2])])
97
98     #Stop loop when approaching radius:
99     if r[0] > 0.935*R:
100         break
101     #end if statement
102 #end of for loop

```

```

103
104 #Fix null acceleration:
105 a = np.array([0,0,0])
106
107 ### Loop over remaining times ###
108 for i in range(x.size,t.size):
109     #Update position:
110     r = r + dt*v
111
112     #Store positions:
113     x = np.append(x, [r[0]])
114     y = np.append(y, [r[1]])
115
116     #Store momenta:
117     p_x = np.append(p_x, [m*v[0]/(5.36*10**-22)])
118     p_y = np.append(p_y, [m*v[1]/(5.36*10**-22)])
119
120     #Store radii:
121     r_xy = np.append(r_xy, [np.sqrt(r[0]**2 + r[1]**2)])
122     r_p = np.append(r_p, [np.sqrt((m*v[0])**2 + (m*v[1])**2)/(-q*B[2])])
123
124     #Stop loop once radius is reached:
125     if r[1] > R:
126         break
127     #end if statement
128 #end of for loop
129
130 p_l = np.linspace(0.5211189*np.pi,1.4788811*np.pi,100) #Angles for left dee
131 x_l = np.append([-d/2,-d/2],0.906996*np.cos(p_l)) #X coordinates of left dee
132 y_l = np.append([-0.86,0.95],0.906996*np.sin(p_l)+0.045) #Y coordinates of left dee
133
134 p_r = np.linspace(-0.4788811*np.pi,0.4788811*np.pi,100) #Angles for right dee
135 x_r = np.append([d/2,d/2],0.906996*np.cos(p_r)) #X coordinates of right dee
136 y_r = np.append([0.95,-0.86],0.906996*np.sin(p_r)+0.045) #Y coordinates of right dee
137
138 #Plot y positions vs. x positions:
139 plt.figure(figsize=[12,12])
140 plt.rc('xtick',labelsize=12)
141 plt.rc('ytick',labelsize=12)
142 plt.xlabel(r'$x_{(m)}$',fontsize=14)
143 plt.ylabel(r'$y_{(m)}$',fontsize=14)
144 plt.plot(x,y,color='#74AAE1',linewidth=2)
145 plt.plot(x_l,y_l,color='grey')
146 plt.plot(x_r,y_r,color='grey')
147 plt.savefig('path.png')
148 plt.close()
149
150 #Plot x positions vs. x momenta:
151 plt.figure(figsize=[12,12])
152 plt.rc('xtick',labelsize=12)
153 plt.rc('ytick',labelsize=12)
154 plt.xlabel(r'$p_x$(MeV/c)',fontsize=14)
155 plt.ylabel(r'$x_{(m)}$',fontsize=14)
156 plt.plot(p_x,x,color='#74AAE1',linewidth=2)
157 plt.savefig('momentum_x.png')
158 plt.close()
159
160 #Plot y positions vs. y momenta:
161 plt.figure(figsize=[12,12])
162 plt.rc('xtick',labelsize=12)
163 plt.rc('ytick',labelsize=12)
164 plt.xlabel(r'$p_y$(MeV/c)',fontsize=14)
165 plt.ylabel(r'$y_{(m)}$',fontsize=14)
166 plt.plot(p_y,y,color='#74AAE1',linewidth=2)
167 plt.savefig('momentum_y.png')
168 plt.close()
169
170 #Plot position-based radii vs. momentum-based radii:

```

```

171 plt.figure(figsize=[12,12])
172 plt.rc('xtick',labelsize=12)
173 plt.rc('ytick',labelsize=12)
174 plt.xlabel(r'$r_p$(m)',fontsize=14)
175 plt.ylabel(r'$r_{xy}$(m)',fontsize=14)
176 plt.plot(r_p,r_xy,color='#74AAE1',linewidth=2)
177 plt.savefig('radii.png')
178 plt.close()

```

## 4. Parámetros

Se ingresaron en el código todos los parámetros necesarios para la simulación numérica, como se explica en la sección 2, separados en tres partes: parámetros del ciclotrón (7), parámetros de la partícula (8), y parámetros iniciales para la simulación (9).

$$\left\{ \begin{array}{ll} E = 6 \text{ MV} & \text{Magnitud campo eléctrico} \\ B = 1,64 \text{ T} & \text{Magnitud campo magnético} \\ R = 0,923 \text{ m} & \text{Radio máximo} \\ d = 120 \text{ mm} & \text{Separación entre las Ds} \end{array} \right. \quad (7)$$

$$\left\{ \begin{array}{ll} q = 1.602 \times 10^{-19} \text{ C} & \text{Carga del protón} \\ m = 1.673 \times 10^{-27} \text{ kg} & \text{Masa del protón} \end{array} \right. \quad (8)$$

$$\left\{ \begin{array}{ll} r_0 = 0,05 \text{ m} & \text{Posición en } y \\ v_0 = 100 \text{ m/s} & \text{Velocidad en } x \\ t_{max} = 1.3 \times 10^{-6} \text{ s} & \text{Tiempo máximo} \\ N = 10000 & \text{Número de puntos} \end{array} \right. \quad (9)$$

## 5. Resultados

Como se detalla en la sección 2, la última parte del código genera cuatro gráficas a partir de los valores que arroja la aproximación numérica realizada. La primera de ellas, la de la trayectoria de las partículas (figura 2), sirve principalmente para confirmar la precisión de la simulación pues la trayectoria de las partículas en un ciclotrón es bien conocida. En efecto se logra la trayectoria en espiral con radio creciente que se espera y puede además comprobarse que los detalles de esta trayectoria varían al variar los parámetros de entrada de acuerdo al funcionamiento elucidado en la sección 1.

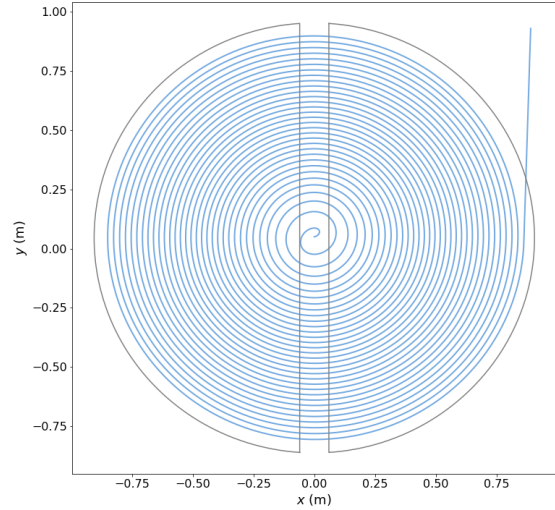


Figura 2: Trayectoria de las partículas.

Habiendo comprobado ya la validez del modelo pueden ahora analizarse las demás gráficas con el fin de entender más sobre este acelerador de partículas. Las gráficas de posición vs. momento (figuras 3 y 4) permiten ver precisamente cómo van ganando energía las partículas a lo largo de su recorrido y cómo se relaciona esto con la trayectoria que trazan. A medida que aumenta la energía de las partículas, es decir, que ganan momento lineal, visto en este caso en su componente en  $x$  y en su componente en  $y$ , y estando sometidas las partículas al mismo campo magnético, las distancias que logran alcanzar sobre cada eje aumentan igualmente. De igual manera, dada la trayectoria misma que siguen las partículas, éstas oscilan sobre cada eje entre los valores positivo y negativo de esta distancia máxima, y de igual manera el vector momento rota, haciendo oscilar cada uno de sus componentes entre los valores positivo y negativo del

momento neto. Este comportamiento resulta, finalmente, en que el par momento-posición, sobre un mismo eje dado, evolucione en el tiempo trazando una espiral como la de la trayectoria de las partículas.

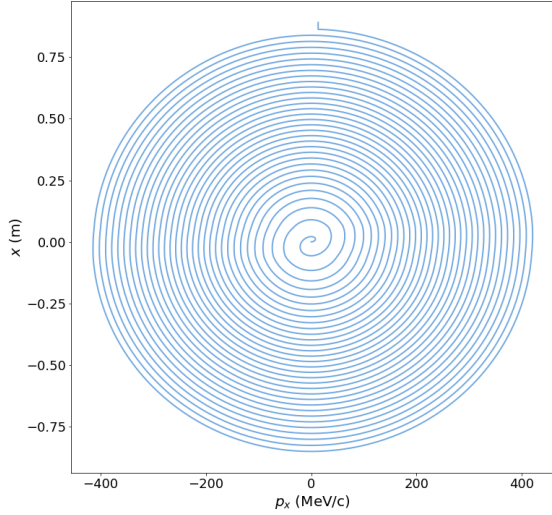


Figura 3: Posición en  $x$  vs. momento en  $x$ .

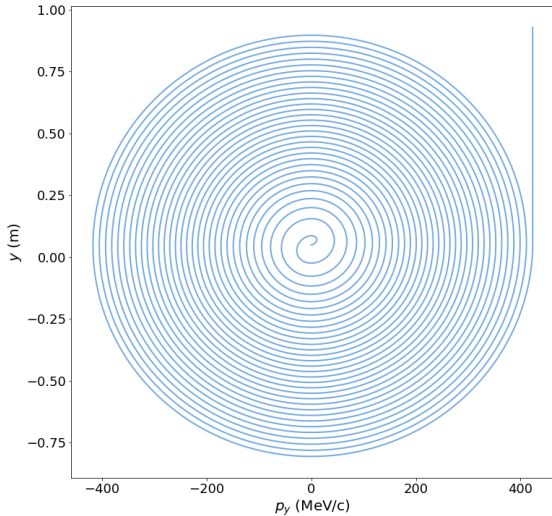


Figura 4: Posición en  $y$  vs. momento en  $y$ .

Finalmente, la última gráfica (figura 5) permite comparar el radio real del recorrido de las

partículas en cada instante de tiempo, es decir, el calculado a partir de su posición, con el radio esperado según la energía alcanzada por la partícula hasta dicho instante. El patrón que se observa responde simplemente a la manera pulsada en que el ciclotrón acelera las partículas, pues el campo eléctrico les transfiere suficiente energía para completar un recorrido de mayor radio antes de que efectivamente lo hagan, pero esta transferencia de energía solo se da en unos momentos del recorrido, permitiendo que el radio real aumente durante el resto del mismo hasta llegar a coincidir ambos. Este comportamiento se repite periódicamente, en cada medio círculo que completan las partículas.

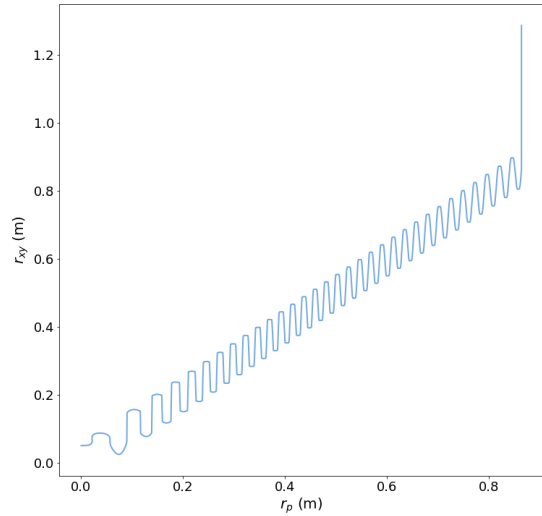


Figura 5: Comparación de radios.

## 6. Conclusiones

El ciclotrón fue un acelerador de partículas de gran importancia histórica, permitiendo alcanzar energías nunca antes vistas y llevando eventualmente al desarrollo del sincrotrón, el que hoy en día es el más moderno acelerador, y aún hoy en día se usan ampliamente los ciclotrones. Es por esto que una parte esencial de aprender sobre aceleradores de partículas es aprender sobre el ciclotrón y entender a fondo el principio

detrás de su funcionamiento. Si bien este funcionamiento puede ser explicado con relativa facilidad, y puede ésto apoyarse por todas las ecuaciones pertinentes, probablemente no hay mejor forma de entender verdaderamente un ciclotrón que simularlo por medio de métodos numéricos, pudiendo así ver paso a paso cómo opera la máquina y cómo pueden manipularse los diversos parámetros de la misma para lograr los resultados que se deseen.

## Referencias

- [1] "A Brief Historical Review of the Invention of the Cyclotron." Koeth Cyclotron. Marzo 10, 2016. Consultado Mayo 28, 2019. [http://koethcyclotron.org/?page\\_id=54](http://koethcyclotron.org/?page_id=54).
- [2] El-Shaftaway, Ashraf. "Regulating the Performance Parameters of Accelerated Particles." Ph.D. thesis, 2013.
- [3] "Runge-Kutta for a System of Differential Equations." Runge-Kutta for a System. Consultado Mayo 28, 2019. <https://www.nsc.liu.se/boein/f77to90/rk.html>.
- [4] "Theory of Operation." Koeth Cyclotron. Marzo 10, 2016. Consultado Mayo 28, 2019. [http://koethcyclotron.org/?page\\_id=64](http://koethcyclotron.org/?page_id=64).