PH2102 Bonus Homework

Lim Zhi Yuan, U1640242D, T4

April 23, 2018

Abstract

In this report, we apply methods of numerical analysis in order to determine the trajectory of a charged particle in a Penning Trap and plot the trajectory using MATLAB.

1 Numerical Analysis

1.1 Lorentz Force

In a Penning Trap, the B-field is given by, $\vec{\mathbf{B}} = B_0\hat{\mathbf{z}}$ and the electric potential is $V(\vec{\mathbf{r}}) = \frac{V_0}{2}\frac{z^2-s^2/2}{d^2}$ in cylindrical coordinates, where d is the *trap characteristic*. In order to compute the Lorentz Force on the charged particle, we require the electric field within the trap. Using $\vec{\mathbf{E}}(\vec{\mathbf{r}}) = -\nabla V$, we obtain,

$$\vec{\mathbf{E}}(\vec{\mathbf{r}}) = -\frac{V_0}{2d^2}\left(2z\,\hat{\mathbf{z}} - s\,\hat{\mathbf{s}}\right) \tag{1}$$

Converting to Cartesian coordinates,

$$\vec{\mathbf{E}}(\vec{\mathbf{r}}) = \frac{V_0}{2d^2}\left(x\,\hat{\mathbf{x}} + y\,\hat{\mathbf{y}} - 2z\,\hat{\mathbf{z}}\right) \tag{2}$$

Now, suppose $\vec{\mathbf{v}} = v_x\,\hat{\mathbf{x}} + v_y\,\hat{\mathbf{y}} + v_z\,\hat{\mathbf{z}}$. Then,

$$\vec{\mathbf{v}} \times \vec{\mathbf{B}} = B_0 v_y\,\hat{\mathbf{x}} - B_0 v_x\,\hat{\mathbf{y}} \tag{3}$$

The Lorentz Force is given by $\vec{\mathbf{F}} = q\left[\vec{\mathbf{E}} + \left(\vec{\mathbf{v}} \times \vec{\mathbf{B}}\right)\right]$. Thus, by Newton's Second Law,

$$\vec{\mathbf{a}} = \frac{q}{m}\left[\vec{\mathbf{E}} + \left(\vec{\mathbf{v}} \times \vec{\mathbf{B}}\right)\right] \tag{4}$$

Component-wise, this gives,

$$a_x = \frac{q}{m}\left(\frac{V_0 x}{2d^2} + B_0 v_y\right) \tag{5}$$

$$a_y = \frac{q}{m}\left(\frac{V_0 y}{2d^2} - B_0 v_x\right) \tag{6}$$

$$a_z = -\frac{q}{m}\left(\frac{V_0 z}{d^2}\right) \tag{7}$$

1.2 Divided/Finite Differences

Let $x[n]$, $v_x[n]$ and $a_x[n]$ denote the x component of the position, velocity and acceleration at the n-th time step. We will make use of the following finite difference approximations[1].

$$a_x[n] = \frac{x[n+1] - 2x[n] + x[n-1]}{h^2} \tag{8}$$

$$v_x[n] = \frac{x[n+1] - x[n-1]}{2h} \tag{9}$$

From (5) and (8), we have,

$$\frac{x[n+1] - 2x[n] + x[n-1]}{h^2} = \frac{q}{m}\left(\frac{V_0 x[n]}{2d^2} + B_0 v_y[n]\right) \tag{10}$$

Applying (9) and making $x[n+1]$ the subject of the formula,

$$\frac{x[n+1] - 2x[n] + x[n-1]}{h^2} = \frac{q}{m}\left[\frac{V_0 x[n]}{2d^2} + B_0\left(\frac{y[n+1] - y[n-1]}{2h}\right)\right] \tag{11}$$

$$\therefore x[n+1] = \left(\frac{qV_0h^2}{2md^2} + 2\right)x[n] - x[n-1] + \frac{qB_0h}{2m}\left(y[n+1] - y[n-1]\right) \tag{12}$$

Similarly, from (6) and (7), we obtain,

$$y[n+1] = \left(\frac{qV_0h^2}{2md^2} + 2\right)y[n] - y[n-1] - \frac{qB_0h}{2m}\left(x[n+1] - x[n-1]\right) \tag{13}$$

$$z[n+1] = \left(-\frac{qV_0h^2}{md^2} + 2\right)z[n] + z[n-1] \tag{14}$$

From here, we define the following constants,

$$k_1 = \frac{qV_0h^2}{2md^2} + 2 \tag{15}$$

$$k_2 = -\frac{qV_0h^2}{md^2} + 2 \tag{16}$$

$$k_3 = \frac{qB_0h}{2m} \tag{17}$$

Noting the cross-dependence between $x[n+1]$ and $y[n+1]$, this would cause problems when iterating through the time steps. To resolve this, we substitute (13) into (12) and obtain,

$$x[n+1] = \frac{1}{1+k_3^2}\left[k_1x[n] - (1-k_3^2)x[n-1] + k_1k_3y[n] - 2k_3y[n-1]\right] \tag{18}$$

We define $k_4 = \frac{1}{1+k_3^2}$. Thus,

$$x[n+1] = k_4\left[k_1x[n] - (1-k_3^2)x[n-1] + k_1k_3y[n] - 2k_3y[n-1]\right] \tag{19}$$

We iteratively compute the particle's position in the Penning Trap using (19), (13) and (14). However, these equations are only valid when $n > 2$. Thus, we need to impose some initial conditions for the system.

[1]Adapted from [1]

1.3 Initial Conditions

Suppose the particle has initial position, $\vec{\mathbf{x}}$, and initial velocity, $\vec{\mathbf{v_0}}$, where

$$\vec{\mathbf{x}} = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix}, \quad \vec{\mathbf{v_0}} = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} \tag{20}$$

Thus, the initial conditions are defined as in Table 1[2], where t is the size of each time-step.

Table 1: Initial conditions for finite difference method

$x[1] = x_0,$	$y[1] = y_0,$	$z[1] = z_0,$
$v_x[1] = u_x,$	$v_y[1] = u_y,$	$v_z[1] = u_z,$
$x[2] = x[1] + v_x[1] \cdot t,$	$y[2] = y[1] + v_y[1] \cdot t,$	$z[2] = z[1] + v_z[1] \cdot t$

1.4 Summary

From the analysis in Sections 1.1 to 1.3, we have the following equations,

$$x[n+1] = k_4 \left[k_1 x[n] - (1 - k_3^2) x[n-1] + k_1 k_3 y[n] - 2k_3 y[n-1]\right] \tag{21}$$

$$y[n+1] = k_1 y[n] - y[n-1] - k_3 \left(x[n+1] - x[n-1]\right) \tag{22}$$

$$z[n+1] = k_2 z[n] + z[n-1] \tag{23}$$

and the initial conditions as shown in Table 1 (in Section 1.3).

2 Simulation with MATLAB software

Listing 1: Setup of parameters

```
%%% Section 1: Setting of parameters %%%

q = 1.6e-19; % Charge
m = 1.67e-27 * 85; % Mass of charged particle
B_0 = 5; % B-field strength
rho_0 = sqrt(2) * 1.12e-3;
z_0 = 1.12e-3;
V_0 = 2; % E-field strength
x0 = 1; % Initial position (x-component)
```

[2]Adapted from [1]

```
y0 = 1; % Initial velocity (y–component)
z0 = 1; % Initial velocity (z–component)
ux = 0; % Initial velocity (x–component)
uy = 0; % Initial velocity (y–component)
uz = 0; % Initial velocity (z–component)
t_max = 1e–4; % Length of time to simulate
dt = 1e–7; % Length of finite time interval

d = sqrt(0.5 * (z_0^2 + (rho_0^2)/2)); % Trap characteristic
```

In the first section of the MATLAB code, we set up the parameters for the Penning Trap. These parameters represent quantities which are as described by their respective comments (except for ρ_0 and z_0) and can be freely changed by the user.

ρ_0 and z_0 describe the shape of the hyperbole that provides the dipole potential for the trap and they are defined by the dimensions of the trap as shown in the image below. The corresponding *trap characteristic*, d is then computed.

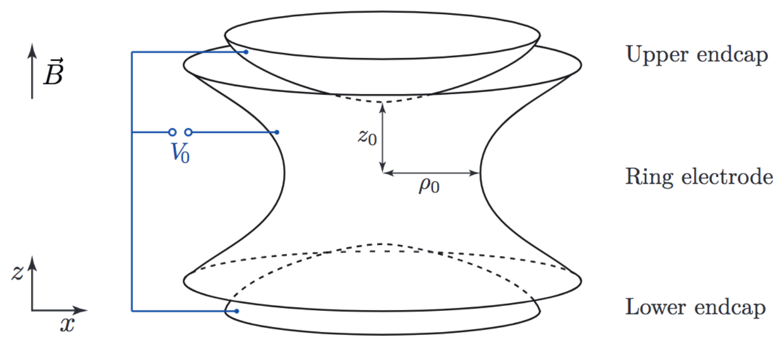

Figure 1: Dimensions of the Penning Trap[3]

Listing 2: Checking if the particle will be trapped

```

%%% Section 2: Checking for Convergence %%%

check = sqrt(2 * (m / q) * (V_0 / z_0^2));
if B_0 < check
    disp('WARNING: Particle will not be trapped')
else
    disp('Parameters are OK')
```

[3]Taken from [2]

We then check if the particle gets trapped based on the condition derived in [2] (specifically, Eq (2.14)). If the particle cannot be trapped, we simply display a warning to the user.

Listing 3: Calculation of constants

```
%%% Section 3: Calculation of constants %%%

k1 = ((q * V_0 * dt^2) / (2 * m * d^2)) + 2;
k2 = -((q * V_0 * dt^2) / (m * d^2)) + 2;
k3 = (q * B_0 * dt) / (2 * m);
```

In this section, we calculate the constants as we defined before in Section 1.2.

Listing 4: Setting up `MATLAB` plotter

```
%%% Section 4: Initialising MATLAB plot %%%

clf;
hold on;
grid on;
```

Here, we simply set up the `MATLAB` plotter, using the `view(3)` command to force a 3D view.

Listing 5: Setup of initial conditions

```
%%% Section 5: Initial conditions %%%

Nintervals = round(t_max/dt);
x = zeros(Nintervals,1);
y = zeros(Nintervals,1);
z = zeros(Nintervals,1);
vx = zeros(2,1);
vy = zeros(2,1);
vz = zeros(2,1);

% Initialising for n = 1

x(1) = x0;
y(1) = y0;
z(1) = z0;
```

```
vx(1) = ux;
vy(1) = uy;
vz(1) = uz;

plot3(x(1), y(1), z(1), 'ro') % Plot the initial position of particle
drawnow;

% n = 2;

x(2) = x(1) + vx(1) * dt;
y(2) = y(1) + vy(1) * dt;
z(2) = z(1) + vz(1) * dt;

plot3(x(2), y(2), z(2), 'ro') % Plot the position of particle at n = 2
drawnow;
```

In this section, we set up the initial conditions of the particle as described in Section 1.3. The vectors named `x`, `y` and `z` store the x-, y- and z-coordinates of the particle at each timestep respectively. At the same time, we also plot the positions of the particle at the 1st and 2nd timestep.

Listing 6: Setup of initial conditions

```

%%% Section 6: Computation and plotting of trajectory %%%

% n > 2;
for t = 3:Nintervals
    x(t) = k4 * ((k1 * x(t-1)) - ((1 - k3^2) * x(t-2)) + (k1 * k3 * y(t-1)) - (2 * k3 *
        y(t-2)));
    y(t) = k1 * y(t-1) - y(t-2) - k3 * x(t) + k3 * x(t-2);
    z(t) = k2 * z(t-1) - z(t-2);
    plot3(x(t), y(t), z(t), 'ro')
    drawnow;
```

Lastly, we calculate the subsequent positions (in each timestep) of the particle using (19), (13) and (14) and plot it. A sample plot of the trajectory of a trapped particle is shown below

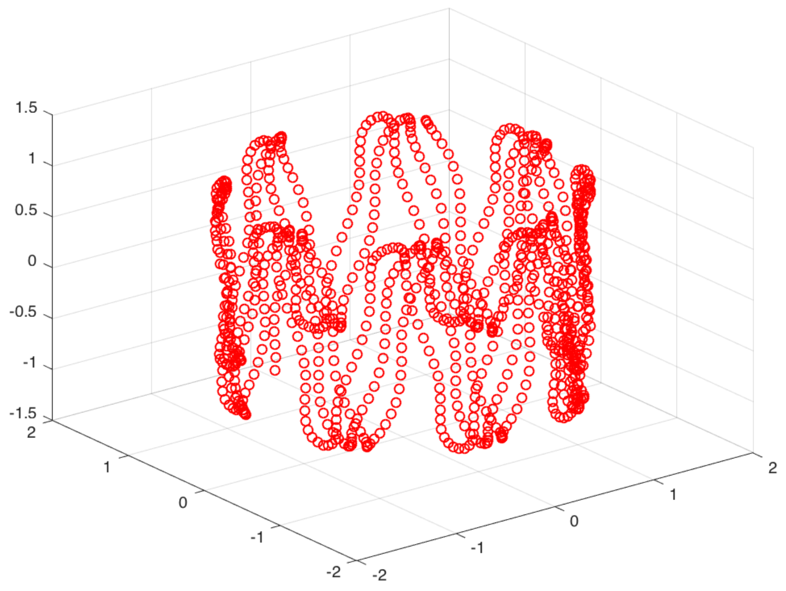

Figure 2: Sample plot of trajectory

References

[1] Cooper, I. *Doing Physics with MATLAB*. School of Physics, University of Sydney.

[2] group, B. Cyclotron frequency in a penning trap. Available at https://www.physi.uni-heidelberg.de/Einrichtungen/FP/anleitungen/F47.pdf, sep 2015.