**This report includes the following:**

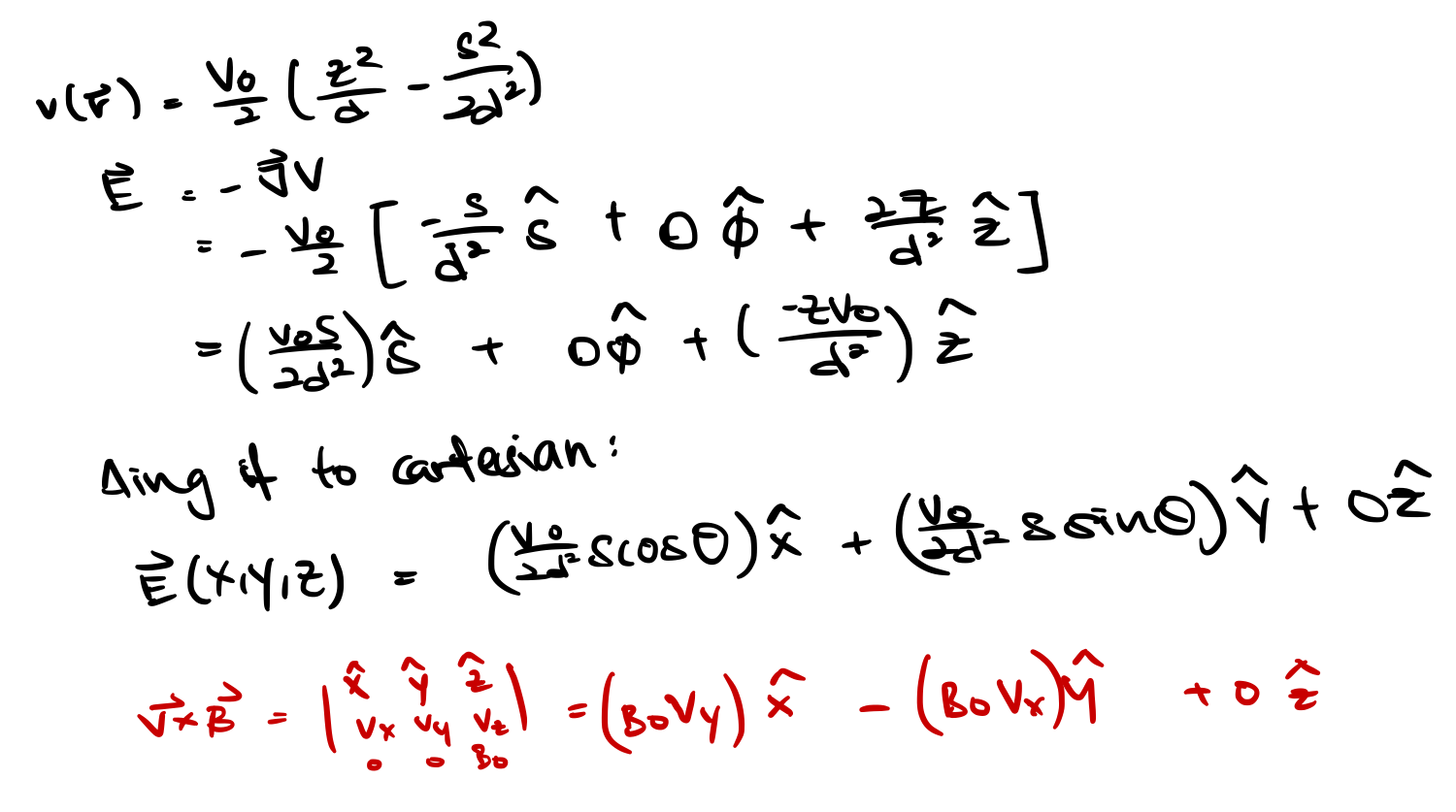
1. **First attempt in solving the question (Wrong method)**
   * 1. **Derivation of results**
     2. **Matlab plot**
     3. **Matlab codes**
     4. **Analysis**
2. **Second attempt in solving the question (Correct method)**
   * 1. **Derivation of results**
     2. **Matlab plots**
     3. **Matlab codes with instructions**
     4. **Boundary conditions of Penning trap with sited link**

**First approach (Wrong method)**

**Derivation of results**

In attempt to solve this question, I first calculate the Lorentz force experienced by the charge (for example, an electron).

Note that F, E, V and B are all directional. In another word, these variables must be evaluated in their components X, Y and Z:



The first attempt to write the derived method in Matlab was to evaluate the force at small intervals of time dt. Using the force, the acceleration can then be measured using the Newton’s second law. After which, kinematics equations were used to determine velocity and then the displacement for each time interval dt. Plots for displacement in X, Y and Z were then plotted as shown:

**Matlab plot**

AngJiaJunRay\_EM\_Bonus\_HW\_First\_attempt\_WRONG.m

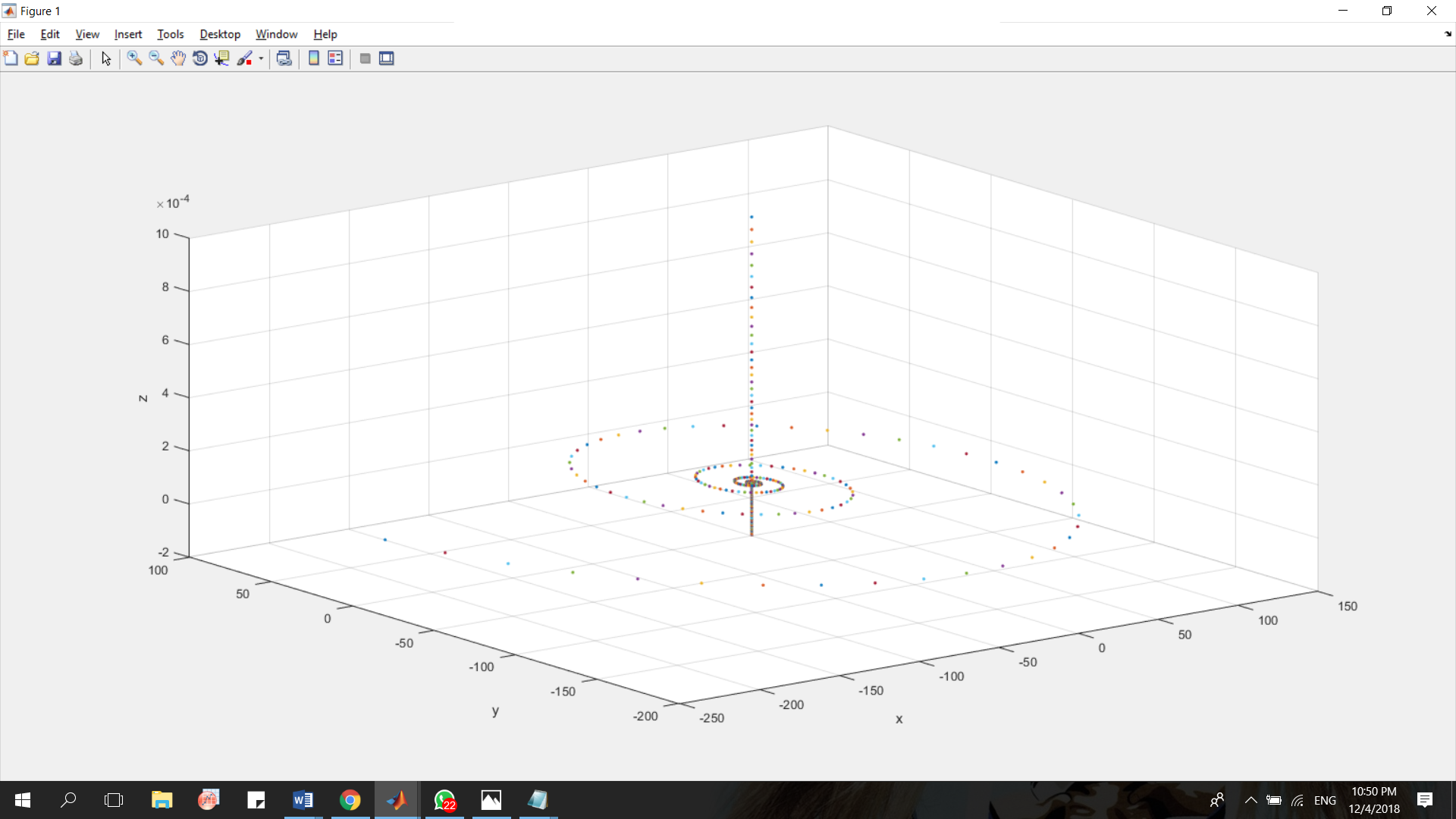


Figure 1: Plot of incorrect coding method

**Matlab codes**

% Motion of a charged partilce in uniform cross B and E fields

% S.I. units used unless otherwise stated

clear all;

close all;

clc;

% =========================================================================

% INPUT PARAMETERS

% =========================================================================

% Particle e = 1.602e-19 proton m = 1.67e-27

q = 1.602e-19;

m = 1.67e-27;

% Initial velocites and displacements

ux = 0; uy = 400; uz = 0;

u = [ ux uy uz ]; % Initial velocity

u\_norm = norm(u);

x0 = 0.001; y0 = 0.001; z0 = 0.001;

s = [ x0 y0 z0 ]; % Initial Displacement

% Fields

% Constants

% Time interval t

v0 = 35.75; % Initial potential

d = sqrt (0.5 \* (.01215 \* .01215 + 0.5 \* 0.015 \* 0.015));

% Trap constant sqrt(.5\*(z\_0\*\*2 + .5 \* rho\_0\*\*2))

t = 5e-8; % Time interval between each plot

% Magnetic field

bx = 0; by = 0; bz = 0.030;

B = [ bx by bz ];

for time = 0 : t : 1e-3

% Pull out x, y and z from displacement (which changes every loop)

x1 = s(1,1);

y1 = s(1,2);

z1 = s(1,3);

% Since the additional quadraple electric potential superimposed is in

% cylindrical coordinates, we have to convert it the above into such

% coordinates, calculate the potential, and then convert it back to

% cartesian for plotting

% Exploting cart2pol function to obtain cylindrical coordinates

[ angle radius height ] = cart2pol(x1, y1, z1);

% Es = ( ( v0 .\* radius ) ./ ( 2 \* d .\* d ) );

Ex = ( ( v0 .\* radius ) ./ ( 2 \* d .\* d ) ) \* cos( angle );

Ey = ( ( v0 .\* radius ) ./ ( 2 \* d .\* d ) ) \* sin( angle );

Ez = - ( ( v0 .\* height ) ./ ( d .\* d ) );

% Ez remains the sameEz = - ( ( v0 .\* height ) ./ ( d .\* d ) );

% Generating the total E field

E = [ Ex Ey Ez ];

% To calculate the Lorentz Force experienced by the charge

% f = q \* ( E + V x B )

% C = cross(A,B) returns the cross product of the vectors

% A and B. That is, C = A x B. A and B must be 3 element vectors.

f = q .\* ( E + cross(u,B) );

% For that specific force f, acceleration a = f / m

a = f ./ m;

% To calculate new velocity u + a \* t

u\_add = a.\* t;

u = u + u\_add;

u\_norm\_1 = norm (u);

uu = u / u\_norm\_1 \* u\_norm;

% To calculate new displacement where s = u \* t + 0.5 \* a \* t \* t

s\_add = ( uu .\* t + 0.5 \* a .\* t .\* t );

s = s + s\_add;

% Plotting function using drawnow

plot3(x1, y1, z1, '.', 'LineWidth', 1.2);

xlabel('x'); ylabel('y'); zlabel('z');

drawnow;

grid on;

hold on;

view(3);

end

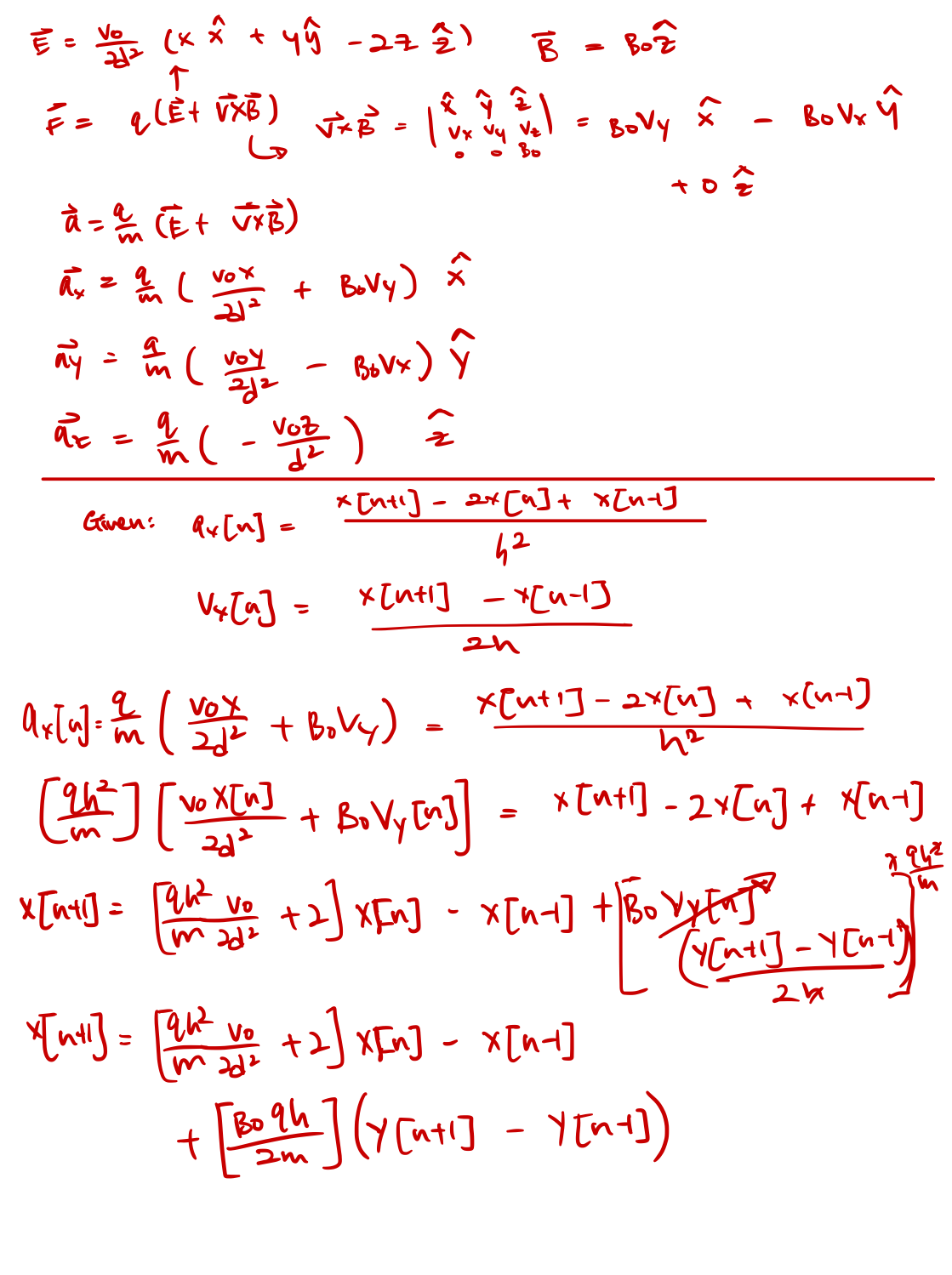
**Analysis**

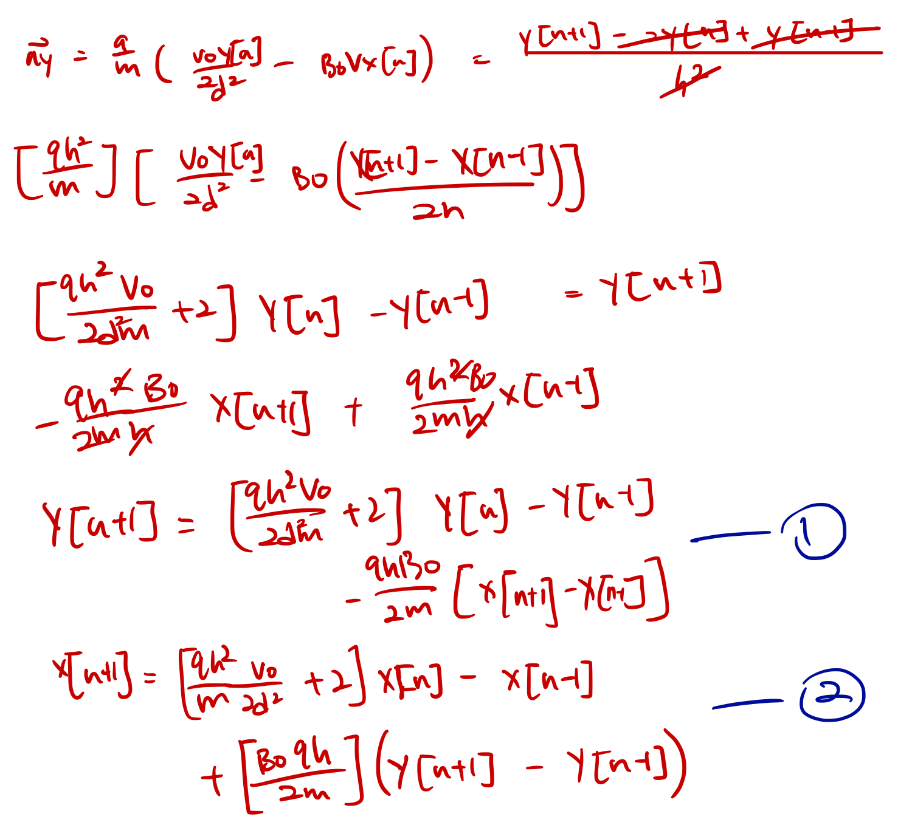
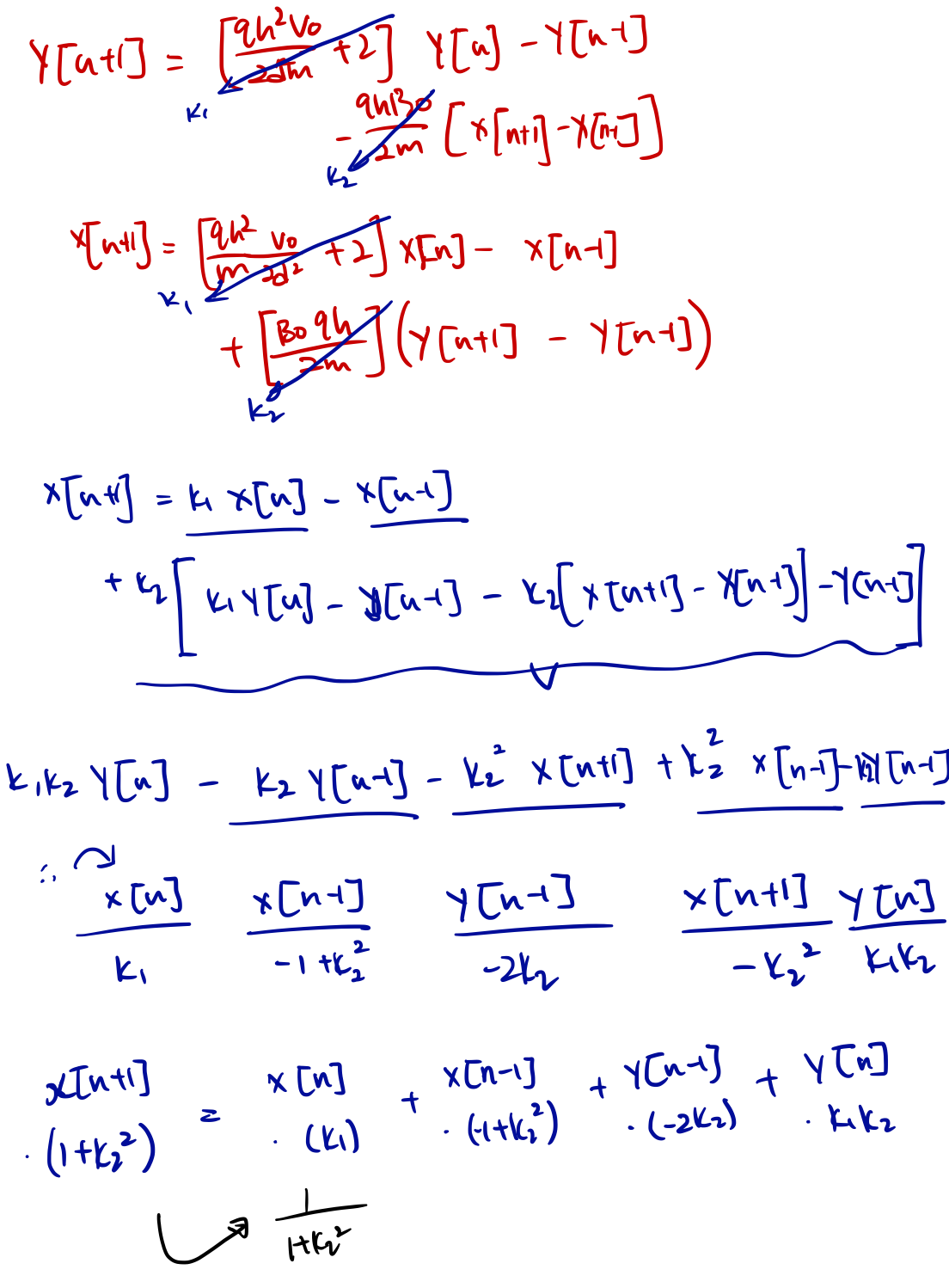
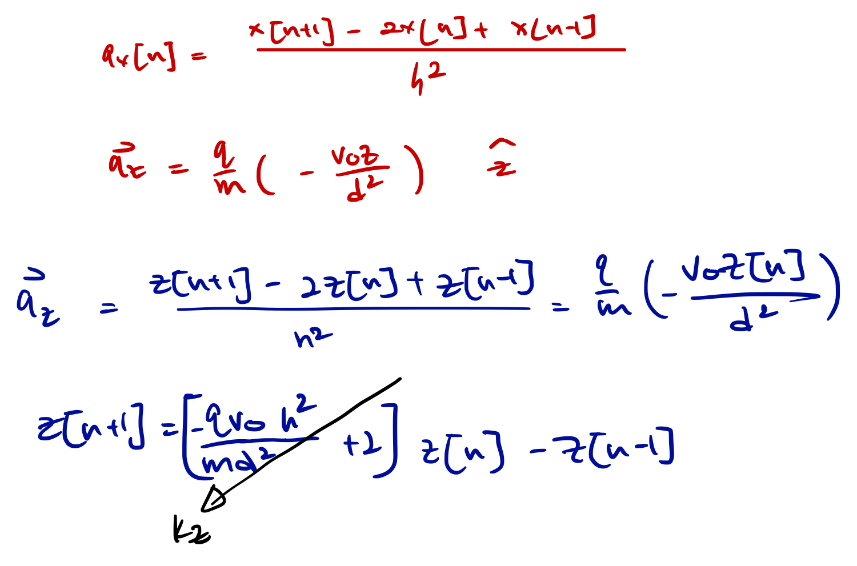
Attention should be paid to the fact that the graph is diverging, implying that the charge was not trapped as it should be. The Physics behind this method is not wrong, it is the arithmetic errors that occur because the time should be continuous. The displacement of the charge obtained by diving the entire duration into intervals only gives an approximation. Even when we increase the number of intervals, it does not give the exact and absolute displacement. As such, the errors from the approximation surfaces when the duration of time is relatively long. I attempted to normalize the magnitude of displacement. However, this will only work for the case of , since theoretically, with the direction of a constant force perpendicular to the direction of motion towards the centre of circle will only result in a change of direction and not the magnitude. However, this is not the case for . Therefore, this method of plotting using Matlab is wrong, despite the correct way of thought.

**Second approach (Correct method)**

**Derivation of results**

As introduced in the link provided in the question paper, the theory seeks to evaluate the displacement of the electron using its previous displacement and velocity, when it experiences Lorentz force. By equating the acceleration due to Lorentz force to the mathematical equations as explained, the displacements at each time interval dt is accurately calculated. Below is the rough work of determining the displacement:



To sum the above mathematical steps, these are my results:

|  |  |
| --- | --- |
|  | Magnitude of charge |
|  | Duration of each time interval |
|  | Magnitude of Voltage |
|  | Mass of charge |
|  | Magnitude of Penning trap constant |

With , and being constants as followed:

Displacement in each direction is given as such:

(1)

(2)

(3)

**Matlab plots**

With these results, I work to plot the displacement of the charge at each time interval dt using Matlab: (evidently, the charge is **TRAPPED**)

AngJiaJunRay\_EM\_Bonus\_HW.m

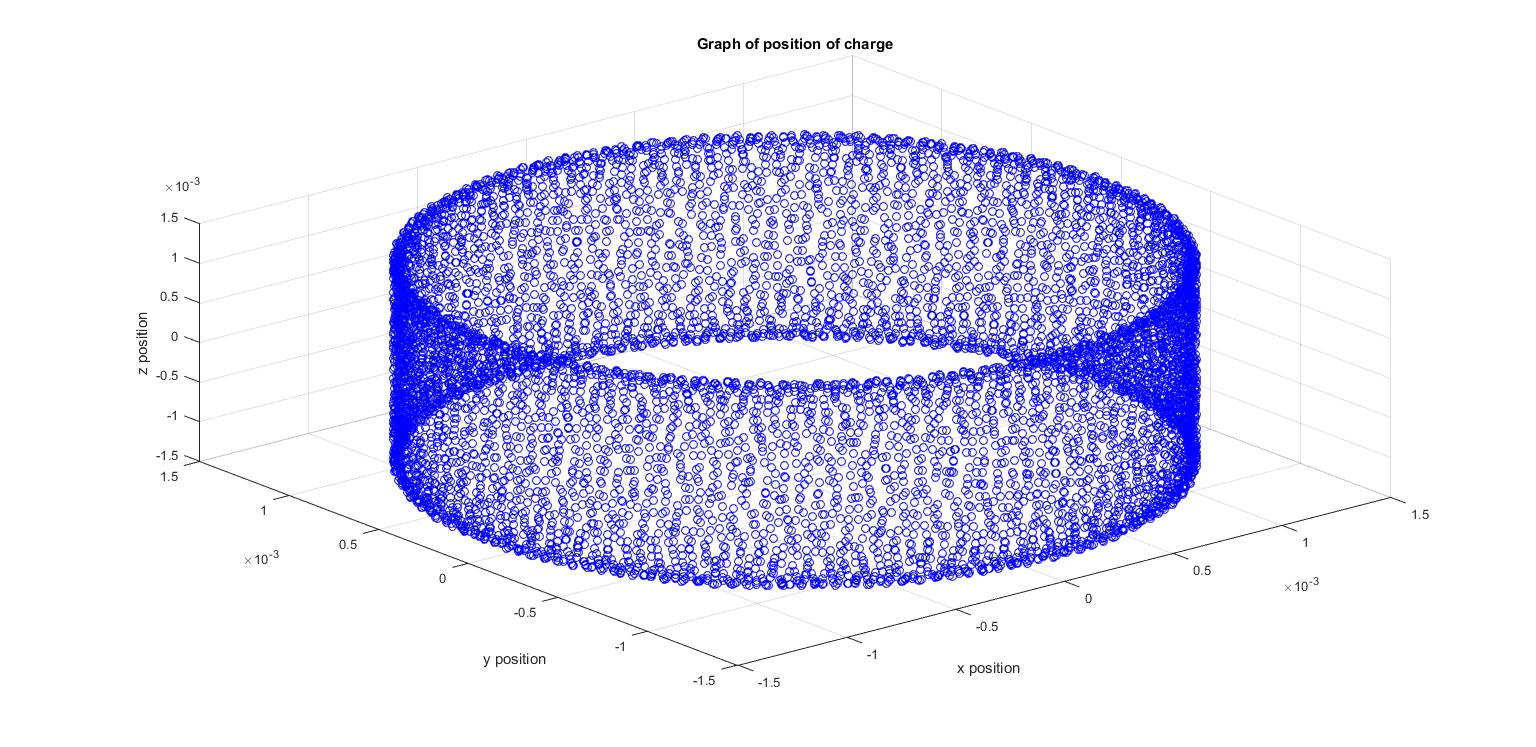


Figure 2: Plot of correct coding method (Side view)

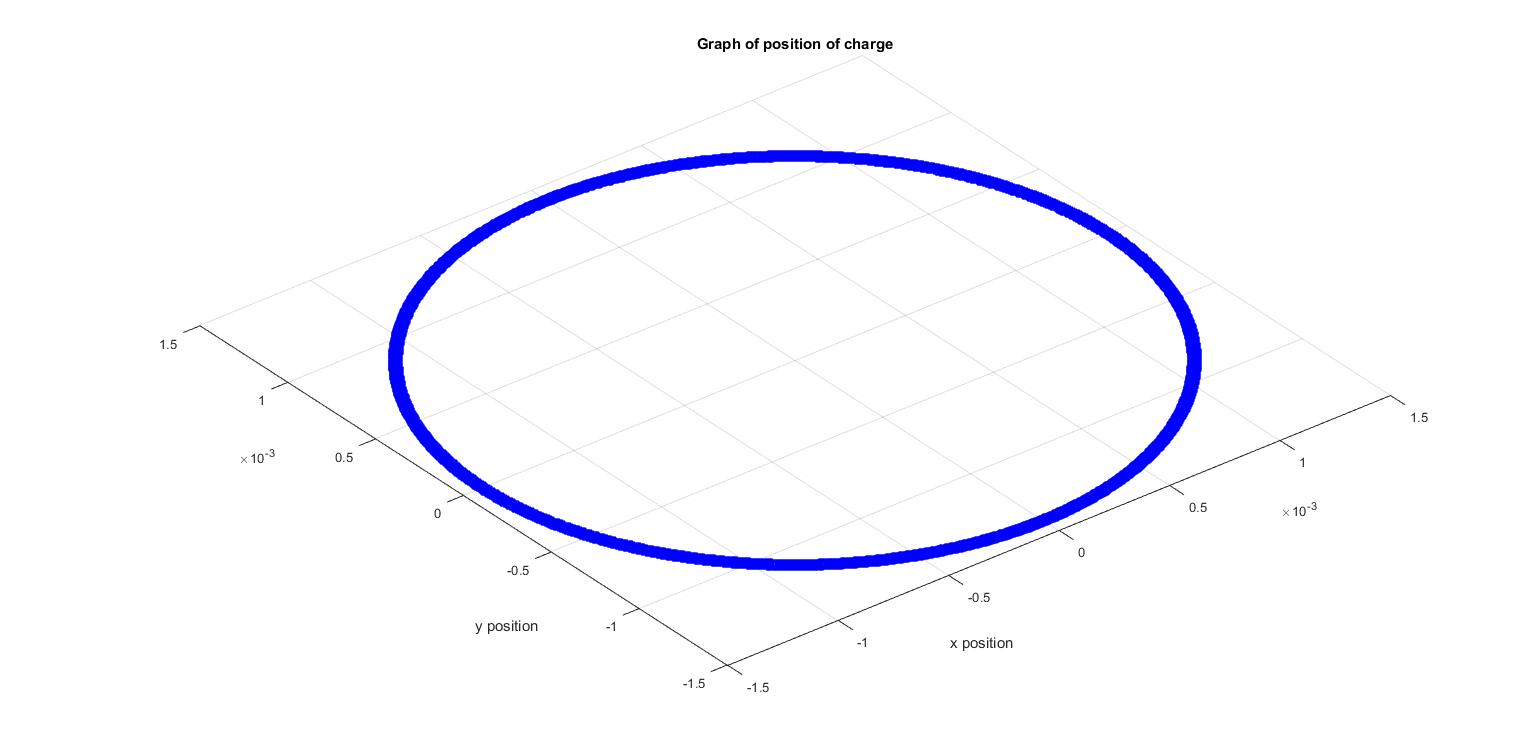


Figure 3: Plot of correct coding method (Top view)

**Matlab codes with instructions**

The Matlab codes are rather self-explanatory. Each step of my work is accompanied with explanation or information of the code. Note that the first part of my code, bounded by “INPUT PARAMETERS” and “END OF PARAMETERS INPUT”, contains information regarding magnitude of particle charge, mass, initial displacements and velocities in x, y and z directions, magnetic field strength, initial voltage, length of each time interval, duration of plot and penning trap constant. These are variables that can be altered according to the user. However, not all initial conditions allow the charge to be trapped. The conditions that are required can be simplified into a single equation and is included at the end of this report.

% Motion of a charged partilce in uniform cross B and E fields

% S.I. units used unless otherwise stated

clear all;

close all;

clc;

clf;

view(3);

hold on;

grid on;

% =========================================================================

% INPUT PARAMETERS

% =========================================================================

% Particle e = 1.602e-19 proton m = 1.67e-27

q = 1.602e-19; m = 1.67e-27;

% Initial displacements

x0 = 0.001; y0 = 0.001; z0 = 0.001;

% Initial velocites

ux = 1e-3; uy = 1e-3; uz = 0;

% B-field strength (Only appled to Z direction)

B\_0 = 5.05;

% Initial voltage strength

V\_0 = 2;

% Time parameters

dt = 1e-7; % Time interval t (which is given as h in the formulas)

duration = 1e-3; % Total duration

index = round(duration/dt); % Number of intervals / plots

% Constants for penning trap

d = 1.12e-3;

% =========================================================================

% END OF PARAMETERS INPUT

% =========================================================================

The code as explained in my equations require the first two terms of the displacement and the first term of the velocity to be computed beforehand, explaining the initialization and the manual plotting of the first 2 terms. Subsequent plotting can be generated using equations 1, 2 and 3.

% Initializing displacement for plots

x = zeros(index,1);

y = zeros(index,1);

z = zeros(index,1);

% Initializing velocity for use

% Only the first 2 cases have to be initialized

vx = zeros(2,1);

vy = zeros(2,1);

vz = zeros(2,1);

% Initializing acceleration for use

% Only the first 2 cases have to be initialized

ax = zeros(2,1);

ay = zeros(2,1);

az = zeros(2,1);

%%%%%%%%%%%% Evaluating the parameters %%%%%%%%%%%%

%%% Derivations can be found in the rough notes %%%

k1 = ((q\*dt\*dt\*V\_0)/(2\*d\*d\*m))+2;

k2 = (B\_0\*q\*dt)/(2\*m);

% Constant for z component

% Independent of x and y

kz = -(q\*V\_0\*dt\*dt)/(m\*d\*d)+2;

% Initialising for n = 1

x(1) = x0;

y(1) = y0;

z(1) = z0;

plot3(x(1),y(1),z(1),'bo')

drawnow;

% Initialising for n = 2 by first calculating the first two terms

% of the velocity in each direction

vx(1) = ux;

vy(1) = uy;

vz(1) = uz;

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),'bo')

drawnow;

% Generating plots for n = 3 onwards

% using the initialized first and second terms

for i = 3:index

x(i) = ((1+k2^2))^(-1)\*((k1\*x(i-1))+((-1+k2^2)\*x(i-2))+((-2)\*k2\*y(i-2))+(k1\*k2\*y(i-1)));

y(i) = k1\*y(i-1)+(-y(i-2))+(-k2\*x(i))+k2\*x(i-2);

z(i) = kz\*z(i-1)-z(i-2);

plot3(x(i),y(i),z(i),'bo')

drawnow;

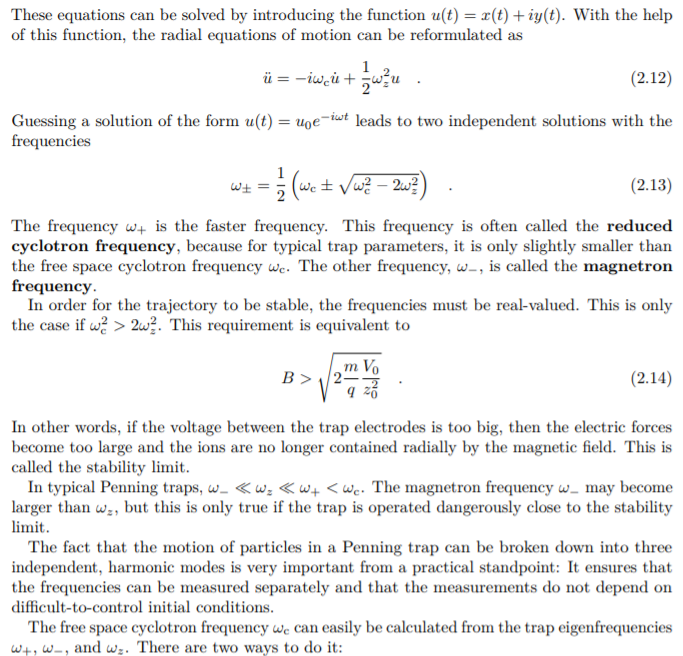
end

**Boundary conditions of Penning trap with sited link**

However, not all initial conditions allow the charge to be trapped. A very well written article on cyclotron motion can be found in the link below:

<https://www.physi.uni-heidelberg.de/Einrichtungen/FP/anleitungen/F47.pdf>

It explains thoroughly and derived the conditions that have to be met for a charge to be trapped in the cyclotron motion, and simplified into equation (2.14) in the article.



Conclusion

With this, I conclude the assignment.