

Jeanette Arjona Hernández/ICT - A01236226

Sebastián Reséndiz García/INA - A01236336

José Alfonso López Blanco/IIT- A01236245

Arif Morán Velázquez/IFI- A01234442

Alan Martínez Varela/ICT - A01236407

Modelación Computacional de sistemas Electromagnéticos

Simulación de la caída de un dipolo magnético en un tubo conductor

Prof. Alejandro García Ruíz

Torreón, Coahuila México

Martes 14 de junio del 2022

**Introducción**

En este documento, se los obtiene la modelación cinemática de un dipolo magnético en un tubo conductor a través del planteamiento de una ecuación diferencial. Dicha ecuación se resolverá a través de un código en MATLAB por el método numérico de Runge Kutta de cuarto orden.

**Fuerza magnética e inducción magnética**

La fuerza magnética es la fuerza que experimenta una carga o más apegado a la realidad, una corriente debido a un campo magnético.

La inducción magnética es un proceso mediante el cual la variación en flujo magnético por unidad de tiempo genera una fuerza electromotriz en un segundo circuito. Este fenómeno es descrito matemáticamente por la ley de Lenz y de Faraday. Así mismo conocemos, similar a la ley de gauss, que el flujo magnético en un área gaussiana cerrada, es igual a la cantidad del flujo del campo que cruza perpendicularmente la superficie.

Ley de Lenz

**Fuerza magnética en un dipolo.**

La fuerza magnética se define como

La energía potencial que experimenta un imán es proporcional a su momento magnético y el campo magnético de un segundo dipolo.

**Aplicación de la segunda ley de newton**

La segunda ley de Newton al movimiento de un dipolo magnético.

* Los fundamentos de los fenómenos del magnetismo (Leyes de Faraday).

**Tracker**

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

**Gráfico, Gráfico de líneas

Descripción generada automáticamente**

* Como crear una solución de software para representar el movimiento de un dipolo magnético.

Cálculos para la obtención de la ecuación diferencial

Aplicamos la Ley de Ohms

Campo magnético de la Espira

Sustituimos la corriente inducida obtenida en el paso anterior

Cálculos para la fuerza sobre el imán

Aplicamos la Ley de Ohms

Aplicamos le dé Lorentz a una espira

Obtendremos la fuerza que ejerce cada espira

Pasaremos el Campo magnético a coordenadas cilíndricas

Aplicamos segunda ley de newton

anchom=app.AnchodelImanxEditField.Value;

sizem=app.LargodelImanEditField.Value;

d=app.diametrodeltuboEditField.Value; %Diamtetro del tubo

h =app.alturadeltuboEditField.Value; %altura del tubo

h2=app.alturasobreeltuboEditField.Value;

m=app.masaimanEditField.Value;

B=app.CampomagneticodelimanEditField.Value;

k=1e-07; %Miu0/4pi

miu0=pi\*4e-07;% Constante de permitibidad magnetica en el vacio

n=35; %numero de atomos en el polo

miu1=(2\*pi\*B\*0.001^2)/miu0; %Momento magnetico del iman

ps=(miu1\*sizem); %Fuerza del polo

lambda=(ps)/(n^2); %Fuerza de polo de cada atomo en la parte Norte

lambda1=-(ps)/(n^2);%%Fuerza de polo de cada atomo en la parte sur

% Creacion de atomos en cada polo

xx=linspace(-anchom/2,anchom/2,n);

yy=linspace(0,sizem/2,n);

[xqn1,yqn1]=meshgrid(xx,yy);

qn1=ones(size(xqn1)).\*lambda; %Asigancion de fuerza de polo a cada atomo

yyy=linspace(-sizem/2,0,n);

[xqn2,yqn2]=meshgrid(xx,yyy);

qn2=ones(size(xqn2)).\*lambda1; %Asigancion de fuerza de polo a cada atomo

%Meshgrid, asignación puntos en el espacio

n2=sizem\*5;

bx=linspace(-anchom/2-n2,anchom/2+n2,n);

By=linspace(-sizem/2-n2,sizem/2+n2,n);

[X,Y]=meshgrid(bx,By);

x3=X;

y3=Y;

distx=[]; %Distancias en x

disty=[]; %Distancias en y

num1=[]; %Parte de arriba de la ecuacion del campo magnetico de componentes en x del polo norte

num2=[];%Parte de arriba de la ecuacion del campo magnetico de componentes en y del polo norte

p=1;

xqn1=xqn1';

xqn2=xqn2';

yqn1=yqn1';

yqn2=yqn2';

for i=1:length(xqn1)^2 %Distancias X Y de cada atomo en el polo norte

distx=[distx;x3-xqn1(i)];

disty=[disty;y3-yqn1(i)];

num1=[num1;k.\*qn1(i).\*distx(p:i\*length(bx),1:length(bx))];

num2=[num2;k.\*qn1(i).\*disty(p:i\*length(bx),1:length(bx))];

p=(i\*length(bx))+1 ;

end

denom=(distx.^2+disty.^2).^1.5; % Parte de abajo de al ecuacion del campo magnetico

Bxx=num1./denom; % Matriz con todos los componentes x de cada atomo polo norte

Bx=zeros(length(bx),length(bx)); %Matriz donde se sumaran todos los coponentes x de cada atomo polo norte

m22=1;

for q=1:length(bx)

Bx=Bx+Bxx(m22:q\*length(bx),1:length(bx));%suma

m22=q\*length(bx)+1;

end

Byy=num2./denom; % Matriz con todos los componentes y de cada atomo polo norte

By=zeros(length(bx),length(bx));%Matriz donde se sumaran todos los coponentes x de cada atomo polo norte

a=1;

for r=1:length(bx)

By=By+Byy(a:r\*length(bx),1:length(bx));% Suma

a=r\*length(bx)+1;

end

%Parte negativa

% Se efectuan exactamente los mismos caluclos pero con los atomos situados

% en el polo sur del iman

distx2=[];

disty2=[];

num3=[];

num4=[];

h3=1;

for i2=1:length(xqn2).^2

distx2=[distx2;x3-xqn2(i2)];

disty2=[disty2;y3-yqn2(i2)];

num3=[num3;k.\*qn2(i2).\*distx2(h3:i2\*length(bx),1:length(bx))];

num4=[num4;k.\*qn2(i2).\*disty2(h3:i2\*length(bx),1:length(bx))];

h3=(i2\*length(bx))+1 ;

end

denom2=(distx2.^2+disty2.^2).^1.5;

Bxx2=num3./denom2;

Bx2=zeros(length(bx),length(bx));

u=1;

for i3=1:length(bx)

Bx2=Bx2+Bxx2(u:i3\*length(bx),1:length(bx));

u=i3\*length(bx)+1;

end

Byy2=num4./denom2;

By2=zeros(length(bx),length(bx));

v=1;

for i4=1:length(bx)

By2=By2+Byy2(v:i4\*length(bx),1:length(bx));

v=i4\*length(bx)+1;

end

%%%%%%% Superposicion de la matriz con coponentes x y de los polos norte y sur%

bx=Bx+Bx2;

by=By+By2;

mb=sqrt(bx.^2+by.^2); %magnitud de los vectores

bx1=bx./mb; % Vectores unitarios

by1=by./mb;

%Graficacion

quiver(app.UIAxes3,X,Y,bx1,by1,'k')

axis(app.UIAxes3,'equal')

hold(app.UIAxes3,'on')

plot(app.UIAxes3,xqn1,yqn1,'r\*') %Graficacion de los atomos en el polo norte

streamslice(app.UIAxes3,X,Y,bx1,by1) %Lineas de campo magnetico

plot(app.UIAxes3,xqn2,yqn2,'b\*')%Graficacion de los atomos en el polo sur

xlim(app.UIAxes3,[-anchom/2-n2,anchom/2+n2])

ylim(app.UIAxes3,[-sizem/2-n2,sizem/2+n2])

hold(app.UIAxes3,'off')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Procedimiento de caida%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

g=0.0005; %Grosor 2 mm

pitch = 0.02;

syms th

r = d/2;

x = r\*sin(th);

y = r\*cos(th);

z1 = th/(2\*pi)\*pitch;

tmax = 2\*pi\*h/pitch;

miu0=4\*pi\*(10^-7);

miu1=(2\*pi\*B\*0.001^2)/miu0;

ro=1.72e-08;

om=59.6\*10^6;

Loops=sizem/g;%h/g;

%Tiempo

dt=0.01; %segundos

tl=3.2;

t=0:dt:tl;

%F=i\*(E+cross(v))

a=-9.81;

Pm=zeros(1,length(t));

Pm(1)=h+h2;

a1=zeros(1,length(t));

a1(1)=a;

v=zeros(1,length(t));

v(1)=0;

%Caida libre

for i=1:(length(t)-1)

Pm(i+1)=Pm(i)+v(i)\*dt+(a1(i)\*dt^2)/2;

v(i+1)=v(i)+a1(1)\*dt;

a1(i+1)=a;

if Pm(i+1)<=h

break

end

end

p=i;

R=log((r+g)/(r))/(2\*pi\*h\*om);%Ohms

c=1/R;

% km=(9\*(miu0^2)\*(miu1^2)\*(r^4)\*(Loops^2))/(4\*R\*m);

% f=@(z,v)-9.81-(km\*(z.^2)/((z.^2+r^2).^5)).\*v;

f=@(z,v)-9.81-((15/1024)\*(miu0^2)\*(miu1^2)\*c\*(1/(r^3)-1/(r+g)^3))\*v/m;

dydt=@(y)[y(2); f(y(1),y(2))];

for i5=p:length(t)-1

z0=[Pm(i5);v(i5)];

k1=dydt(z0);

k2=dydt(z0+(dt\*k1)/2);

k3=dydt(z0+(dt\*k2)/2);

k4=dydt(z0+dt\*k3);

Pm(i5+1)=Pm(i5)+(dt/6)\*(k1(1)+2\*k2(1)+2\*k3(1)+k4(1));

v(i5+1)=v(i5)+(dt/6)\*(k1(2)+2\*k2(2)+2\*k3(2)+k4(2));

a1(i5+1)=f(Pm(i5),v(i5));

if Pm(i5+1)<=0

break

end

end

%%%%%%%%%Graficacion%%%%%

fplot3(app.UIAxes,x, y, z1, [0 tmax],'color',[0.9100 0.4100 0.1700],'LineWidth', 1) %Tubo

axis(app.UIAxes,'equal')

grid(app.UIAxes,'on')

hold(app.UIAxes,'on')

%

zero=zeros(length(Pm));

for i4=1:i5

plot3(app.UIAxes,zero(i4),zero(i4),Pm(i4),'k.','MarkerSize',10)

pause(0.1)

plot(app.UIAxes2,t(i4),Pm(i4),'k\*','MarkerSize',5)

plot(app.UIAxes2,t(1:i4),Pm(1:i4),'k','Linewidth',1)

axis(app.UIAxes2,[0 t(i5) -0.1 max(Pm)])

grid(app.UIAxes2,'on')

plot(app.UIAxes4,t(i4),v(i4),'r\*','Markersize',5)

plot(app.UIAxes4,t(1:i4),v(1:i4),'r','Linewidth',1)

axis(app.UIAxes4,[0 t(i5) min(v)-0.1 0.001])

grid(app.UIAxes4,'on')

plot(app.UIAxes5,t(i4),a1(i4),'b\*','Markersize',5)

plot(app.UIAxes5,t(1:i4),a1(1:i4),'b','Linewidth',1)

axis(app.UIAxes5,[0 t(i5) min(a1)-0.1 max(a1)])

grid(app.UIAxes5,'on')

pause(0.1)

end