//AUTHOR: DANIEL CRISP

//DATE: FRIDAY, NOVEMBER 11th, 2016

//DESC: ANIMATION

#QUESTION 2: Decaying Circle

MATLAB CODE:

plotdecay.m

function plotdecay( r0, tau, T, omega, x, y, r, t)

%plotdecay plots decaying circular motion.

clc;

for it = 1:length(t)

plot(x(1:it),y(1:it),'b',x(it),y(it),'ro',[0,x(it)],[0,y(it)],'g','MarkerSize',8,'MarkerFaceColor','r')

axis([-r0 r0 -r0 r0]), grid on

xlabel('x')

ylabel('y')

title('Decaying Circular Motion')

drawnow

end

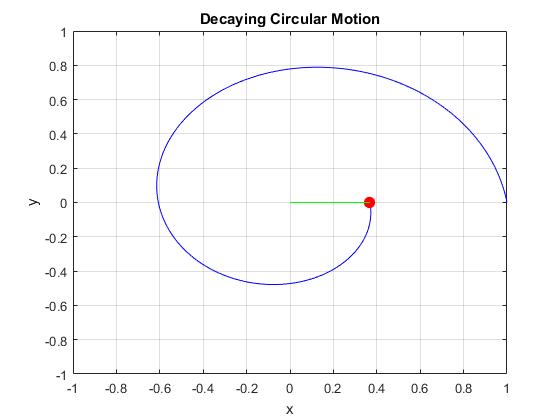
end

MATLAB OUTPUT:

>>r0 = 1; tau = 1; T = 1; omega = 2\*pi/T; t = linspace(0,T,1000);

>>r = r0\*exp(-t/tau); x = r.\*cos(omega\*t); y = r.\*sin(omega\*t);

>>plotdecay(r0,tau,T,omega,x,y,r,t)



#QUESTION 4: Lissajous

MATLAB CODE:

plotLJ.m

function plotLJ( r0, T, phi, a, b, t)

%plotdecay plots decaying circular motion.

clc;

R = r0;

omega = 2\*pi/T;

x = R\*cos(a\*omega\*t);

y = R\*sin(b\*omega\*t + phi);

plot(x,y); axis([-r0 r0 -r0 r0]); grid on

animateON = input('Do you want to see this animated? Enter 0 or 1: ');

if animateON==1

for it = 1:length(t)

plot(x(1:it),y(1:it),'b',x(it),y(it),'ro',[0,x(it)],[0,y(it)],'g','MarkerSize',8,'MarkerFaceColor','r')

axis([-r0 r0 -r0 r0]), grid on

xlabel('x')

ylabel('y')

title('Lissajous')

drawnow

end

end

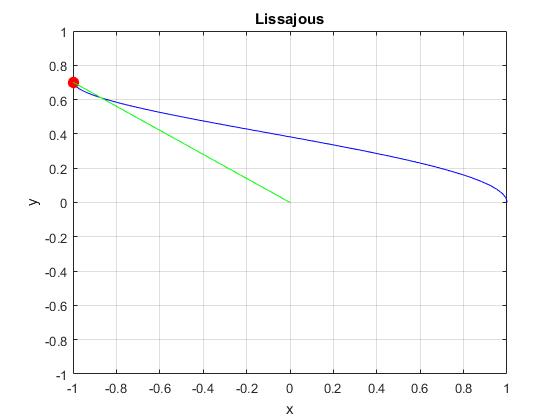
end

MATLAB OUTPUT:

>>T = 1; a = 1; b = .25; phi = pi; R = 1; omega = 2\*pi/T;

>>t = linspace(0,T,1000);

>>plotLJ(R, T, omega, 2\*pi, 1, .25, t);



#QUESTION 7: Epicycles

MATLAB CODE:

function epicycles( T1, Tratio, R, r )

%plotdecay plots decaying circular motion.

clc;

%Assemble time array 't'

Nc = 10; %number of cycles

Ntpc = 200; %number of points per cycle

tot\_time = Nc\*T1;

Nt = Nc\*Ntpc; %length of array time (number of times')

t = linspace(0,tot\_time,Nt);

%Assemble equations. R for centered circle's radius, r for circulating

%circle's radius.

Rmax = R + r;

T2 = T1/Tratio;

omega1 = 2\*pi/T1;

omega2 = 2\*pi/T2;

x = R\*cos(omega1\*t) + r\*cos(omega2\*t);

y = R\*sin(omega1\*t) + r\*sin(omega2\*t);

plot(x,y); axis([-Rmax Rmax -Rmax Rmax]); grid on

animateON = input('Do you want to see this plot animated? Enter 0 or 1: ');

if animateON==1

for it = 1:length(t)

plot(x(1:it),y(1:it),'b',x(it),y(it),'ro',[0,x(it)],[0,y(it)],'g','MarkerSize',8,'MarkerFaceColor','r')

axis([-Rmax Rmax -Rmax Rmax]), grid on

xlabel('x')

ylabel('y')

title('Epicycle Animation')

drawnow

end

end

end

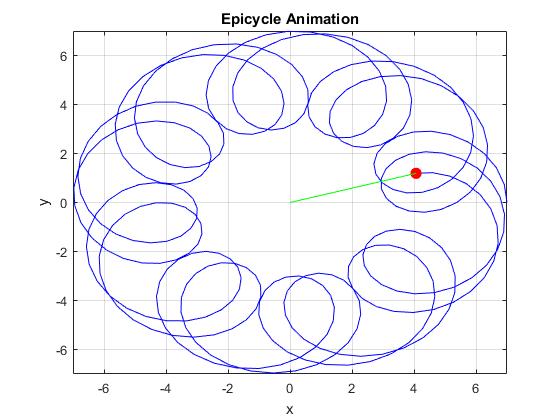
MATLAB OUTPUT:

>>epicycles(1,9.25,5,2)

As the tratio changes, successive cycles can end up can be shifted or made to overlay completely. using a tratio of 9 makes it look like there's only one pass. where the simple rational number of 9.25 had at least 1 in 10 passes re-drawing the first one, a non-simple number like 9.277 doesn't appear to return to re-draw it's initial pass at all.

As r=R the hole inside the plot diminishes, but then as r>R it returns and gets larger.

see code for animateOn.



#QUESTION 8: A random walk in two dimensions

MATLAB CODE:

function rw2d(x0, y0, N)

%Animation of the random walking of a 'person' making 'N' steps of

%length 1 in only one of the four cardinal directions...

%North, South, East or West. x0 and y0 are starting positions.

%% Set Walk Parameters

[x0,y0,N] = request(x0,y0,N);

%% Initialize Arrays

X(1:N) = 0;

Y(1:N) = 0;

%% Calculate Walk

[X,Y] = calculate\_walk(x0,y0,N);

%% Plot Walk

plot\_walk(X,Y);

end

function [x0,y0,N] = request(x0,y0,N)

%Clear console

clc;

%Display initially sent values

fprintf('\nThe 2d walk will start at x = %i, and y = %i.',x0,y0);

fprintf('\nAnd there will be %i steps.\n\n',N);

%Request

prompt = 'Do you want to input different values? Y/N...\n';

response = input(prompt,'s');

%If yes, request new

if response =='Y'||response=='y'

clc;

x0 = input('\nPlease enter a new x0 to start at: ');

y0 = input('\nPlease enter a new y0 to start at: ');

N = input('\nPlease enter the number of steps "N" you want taken: ');

clc;

fprintf('\nStarting at ( %i , %i ), and taking %i steps...\n\n',x0,y0,N);

end

end

function [X,Y] = calculate\_walk(x0,y0,N)

X(1) = x0; Y(1) = y0;

for i=2:N

%Each random step is of length 1, and in one of the four cardinal

%directions

idirection = randi(4);

switch idirection

case 1

% One step North

X(i) = X(i-1);

Y(i) = Y(i-1)+1;

case 2

% One step South

X(i) = X(i-1);

Y(i) = Y(i-1)-1;

case 3

% One step East

X(i) = X(i-1)+1;

Y(i) = Y(i-1);

case 4

% One step West

X(i) = X(i-1)-1;

Y(i) = Y(i-1);

end

end

end

function plot\_walk(X,Y)

% Draw the person's walk path

Xlen = length(X);

for i = 2:Xlen

plot(X(1), Y(1), 'r\*', ...

X(1:i), Y(1:i), ...

X(i), Y(i), 'ro');

axis equal

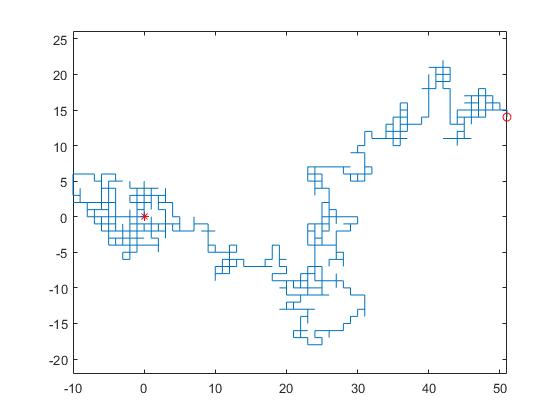
drawnow

end

end

MATLAB OUTPUT:

>>rw2d(0,0,1000)



#QUESTION 9: Vertical motion and energy

I could not seem to get my code to work in time.. graphs were showing straight lines, when they should have been.

verticalMotion(0,.8,0,0,17.8,0,.058)

function verticalMotion(x0,y0,z0,vx0,vy0,vz0,m)

%Clear console

clc;

%Initial Variables

g = 9.81; %Acceleration of gravity (9.81 m/s^2)

X0 = [x0;y0;z0]; %Initial Position

V0 = [vx0;vy0;vz0]; %Initial Velocity

a = [0;-g;0]; %Acceleration (only considering gravity)

%Time

tmax = 4; %seconds

Nt = 500; %time steps

t = linspace(0,tmax,Nt);

syms time; %for easier integration

%Force (F=ma)

F = m\*a; %Only considering gravity so far...

%Velocity

V = V0 + int(F,time); %Velocity Vector

%V.y = V(2);

%Velocity = V.y(time);

S = sqrt(sum(V.^2)); %Speed

%Position

X = X0 + int(V,time);

%Energy

E(1) = sqrt(sum((F.\*X).^2)); %Potential Energy

E(2) = 1/(2\*m\*S^2); %Kinetic Energy

TE = sum(E); %Total Energy

% Display plots of ball height, velocity, and energy

figure

subplot(2,2,1); plot(t,subs(X(2),t));

title('Height of Ball vs. Time'); xlabel('time (s)'); ylabel('Height (m)');

grid on

subplot(2,2,2); plot(t,subs(S,t));

title('Velocity vs. Time'); xlabel('time (s)'); ylabel('Velocity (m/s)');

grid on

subplot(2,2,3); plot(t,subs(E(1),t),'b',t,subs(E(2),t),'r');

title('KE'); xlabel('time (s)'); ylabel('Kinetic Energy');

grid on

subplot(2,2,4); plot(t,subs(TE,t));

title('PE'); xlabel('time (s)'); ylabel('Potential Energy');

grid on

end

#QUESTION 10: Two dimensional billiards