N-Body Simulation

CSE1010 Project 6, Fall 2012

Date: 10/22/2012

Name: William Dickson

Section: 009L

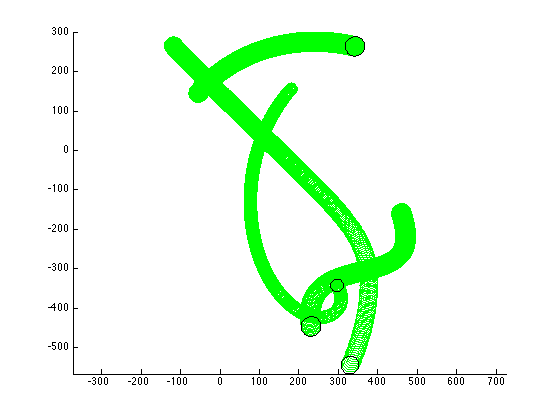
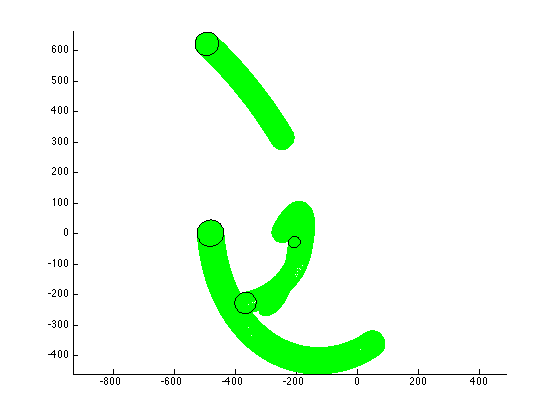
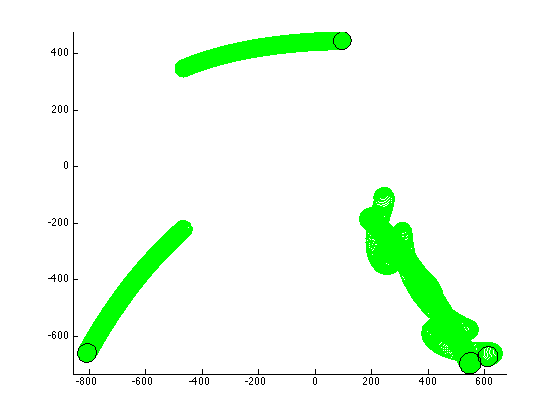
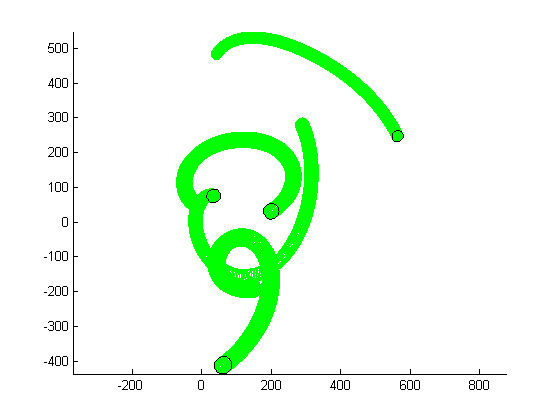
TA: Levon Nazaryan

Instructor: Jeffrey A. Meunier

1. **Introduction**

This project is a discrete simulation solution of an N-body gravitational force problem. Given some number of bodies in empty space, the bodies are allowed to move freely and interact with each other under gravitational force, although in this simulation you will not simulate collisions between the bodies. The result yields an interesting animation.

1. **Test runs**

****

1. **Source Code**

% N-Body Simulation

% CSE1010 Project 6, Fall 2012

% William Dickson

% 10/22/2012

% TA: Levon Nazaryan

% Section: 009L

% Instructor: Jeffrey A. Meunier

clc % clear command window

clear % clear all variables

clf % clear plot area

axis equal

hold on

numBodies = 4;

global G

G = 50; % not -9.8 in this project!

massFactor = 40;

minMass = 5;

distanceFactor = 1000;

velocityFactor = 3;

color = 'green';

% Calculations

for n = 1:numBodies

Masses(n) = rand \* massFactor + minMass;

Xs(n) = rand \* distanceFactor - distanceFactor/2;

Ys(n) = rand \* distanceFactor - distanceFactor/2;

Dxs(n) = rand \* velocityFactor - velocityFactor/2;

Dys(n) = rand \* velocityFactor - velocityFactor/2;

end

% Draw circle

plotBodies(Xs,Ys,Masses,color)

% Change Xs and Ys

[Xs Ys] = moveBodies(Xs, Ys, Dxs, Dys);

% Plot all changes

while true

plotBodies(Xs,Ys,Masses,color)

[Xs,Ys] = moveBodies(Xs,Ys,Dxs,Dys);

[Dxs,Dys] = accelerateBodies(Xs,Ys,Dxs,Dys,Masses);

plotBodies(Xs,Ys,Masses,'black')

pause(eps)

end

function [Dxs,Dys] = accelerateBodies(Xs,Ys,Dxs,Dys,Masses)

% Given the x and y positions of the bodies as well as the current

% velocities (Dxs, Dys) and their masses (radii) this program calculates

% all the accelerations of the bodies by using the accelerateBody on each

% body

% Use: accelerateBodies(Xs,Ys,Dxs,Dys,Masses)

for numBodies = 1:length(Masses)

[Dxs(numBodies),Dys(numBodies)] = accelerateBody(numBodies,Xs,Ys,Dxs,Dys,Masses);

end

end

function [dx,dy] = accelerateBody(b,Xs,Ys,Dxs,Dys,Masses)

% Given a body (number), the x and y coordinates, the Masses(Radii), and the change

% in x and y due to velocity this caclulates the acceleration of the body

% due to the other bodies and adds it to the velocity.

% Use: accelerateBody(b,Xs,Ys,Dxs,Dys,Masses)

for bodyNum = 1:length(Masses)

if bodyNum ~= b

[fx,fy] = calculateForce(b,bodyNum,Xs,Ys,Masses);

ax = fx/Masses(b);

ay = fy/Masses(b);

dx = Dxs(b) + ax;

dy = Dys(b) + ay;

end

end

end

function [fx fy] = calculateForce(body1,body2,Xs,Ys,Masses)

% Given two bodies, the x and y coordinates and the Masses(Radii) this will

% calculate the x and y components of the force between the bodies

% Use: moveBodies(Xs,Ys,Dxs,Dys)

global G

dx = Xs(body2)-Xs(body1);

dy = Ys(body2)-Ys(body1);

r = sqrt(dx^2 + dy^2);

if r < Masses(body1) + Masses(body2)

r = Masses(body1) + Masses(body2);

end

f = G\*Masses(body1)\*Masses(body2)/r^2;

angle = atan2(dy, dx);

fx = f \* cos(angle);

fy = f \* sin(angle);

function circle(x,y,radius,color)

% Draws a circle at x,y having given radius and color.

% Use: circle(x,y,radius,'color')

theta = 0:.01:2\*pi;

X = cos(theta)\*radius + x;

Y = sin(theta)\*radius + y;

plot(X,Y,color)

end

function [Xs,Ys] = moveBodies(Xs,Ys,Dxs,Dys)

% Changes the Xs and Ys values by Dxs and Dys respectively

% Use: moveBodies(Xs,Ys,Dxs,Dys)

Xs = Xs + Dxs;

Ys = Ys + Dys;

end

function plotBodies(Xs,Ys,Masses,color)

% Draws a number of circles equal to length of vector Masses given Xs, Ys and color.

% Use: plotBodies(Xs,Ys,Masses,color)

numBodies = length(Masses);

for b = 1:numBodies

circle(Xs(b),Ys(b),Masses(b),color);

end

end