



function sol = NumSolveCode(Start,Stop,Guess,Init,Final)

%Creates initial guess for solution to bvp

%solinit = bvpinit(xDomain, yguess, params)

solinit = bvpinit(linspace(Start,Stop,Stop-Start+1),Guess);

%Part (i)

sol = bvp4c(@OdeFunc, @(ya,yb) BCFunc(ya,yb,Init,Final), solinit);

% %Part (ii)

% sol = bvp4c(@OdeFunc, @BCFunc, solinit);

color = {'-b+','-ro','--b\*','--rx',':bd',':rs','-.bv','-.r^'};

%plot part (i)

x = linspace(Start,Stop);

%Evaluates solution to problem @ points from x

y = deval(sol,x);

figure;

title('\fontsize{16} Numerically Calculated Trajectory, Part (i)');

xlabel('\fontsize{13} Timestep');

ylabel('\fontsize{13} Position');

hold on

plot(x,y(1,:),color{4});

hold off

% %plot part (ii)

% x = linspace(Start,Stop);

% %Evaluates solution to problem @ points from x

% y = deval(sol,x);

% figure;

% title('\fontsize{16} Numerically Calculated Trajectories, Part (ii)');

% xlabel('\fontsize{13} Timestep');

% ylabel('\fontsize{13} Position');

% hold on

% plot(x,y(1,:),color{4},x,y(2,:),color{5});

% legend('\fontsize{13} x1 Trajectory',...

% '\fontsize{13} x2 Trajectory',...

% 'Location','Best');

% hold off

%Functions for 1st part of HW02

function dxdy = OdeFunc(~,y)

dxdy = [y(2);...

y(2) - sin(y(1))];

function res = BCFunc(ya,yb,Init,Final)

res = [ya(1) - Init;...

yb(1) - Final];

% % Functions for 2nd Part of HW02

% function dxdy = OdeFunc(~,y)

%

% dxdy = [y(1) + 2\*y(2);...

% 3\*y(1) - 4\*y(2)];

%

% function res = BCFunc(ya, yb)

%

% res = [ya(1) - 1;...

% yb(2) - 2];