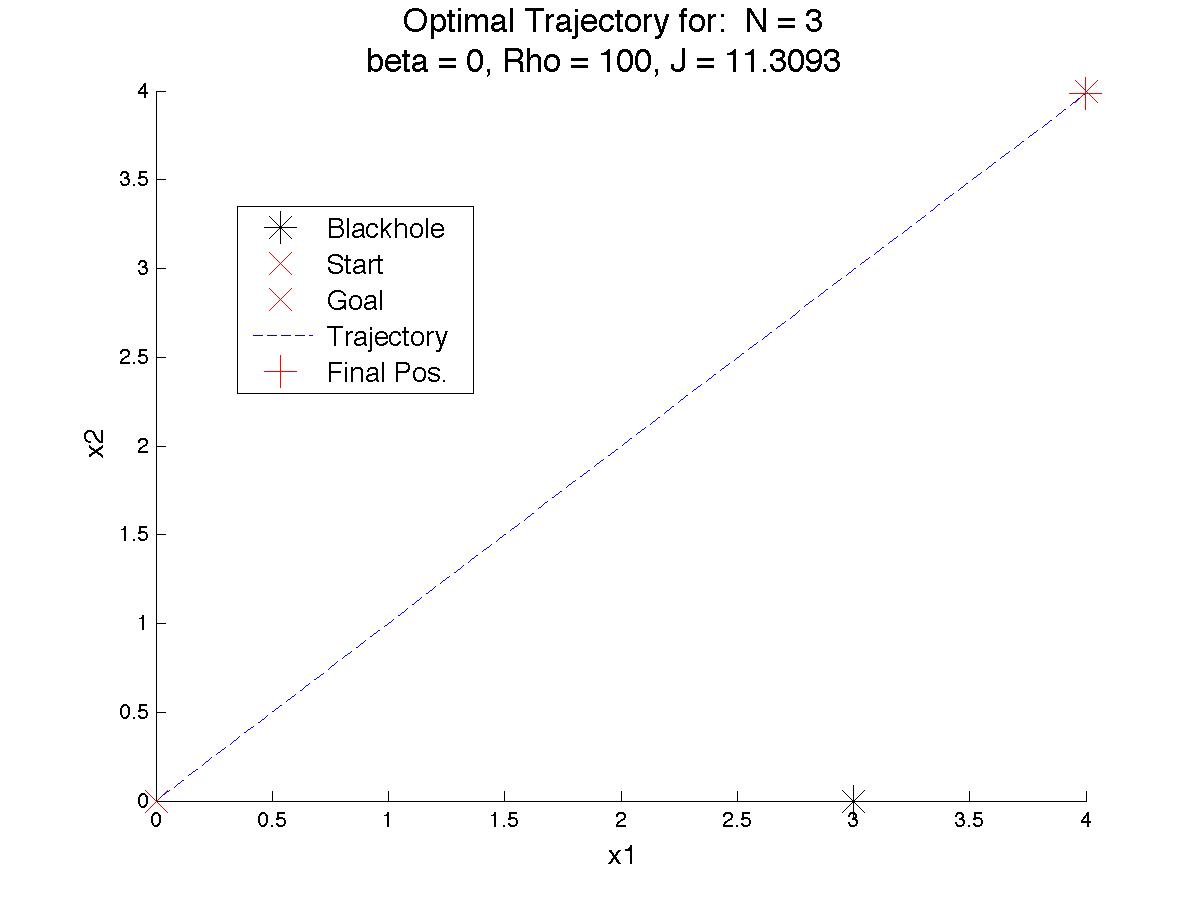
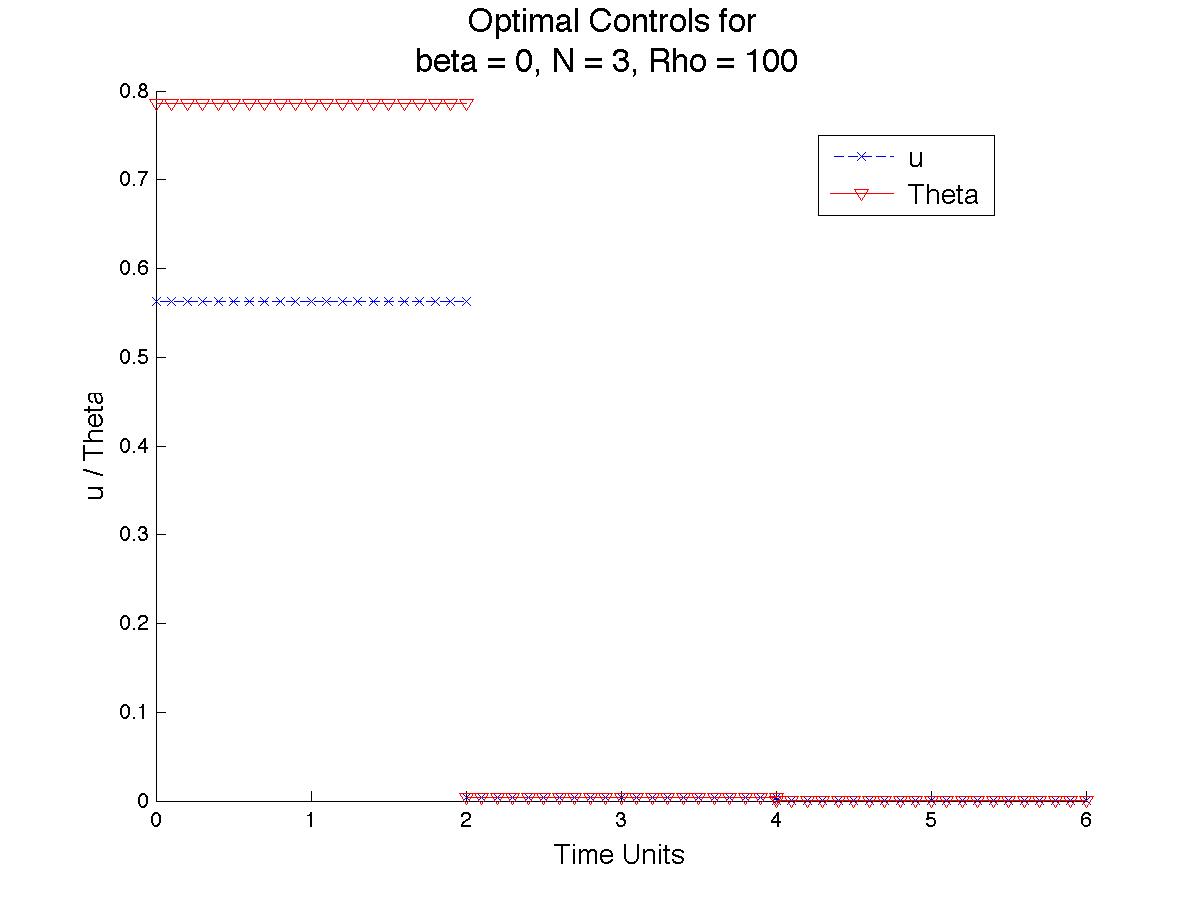
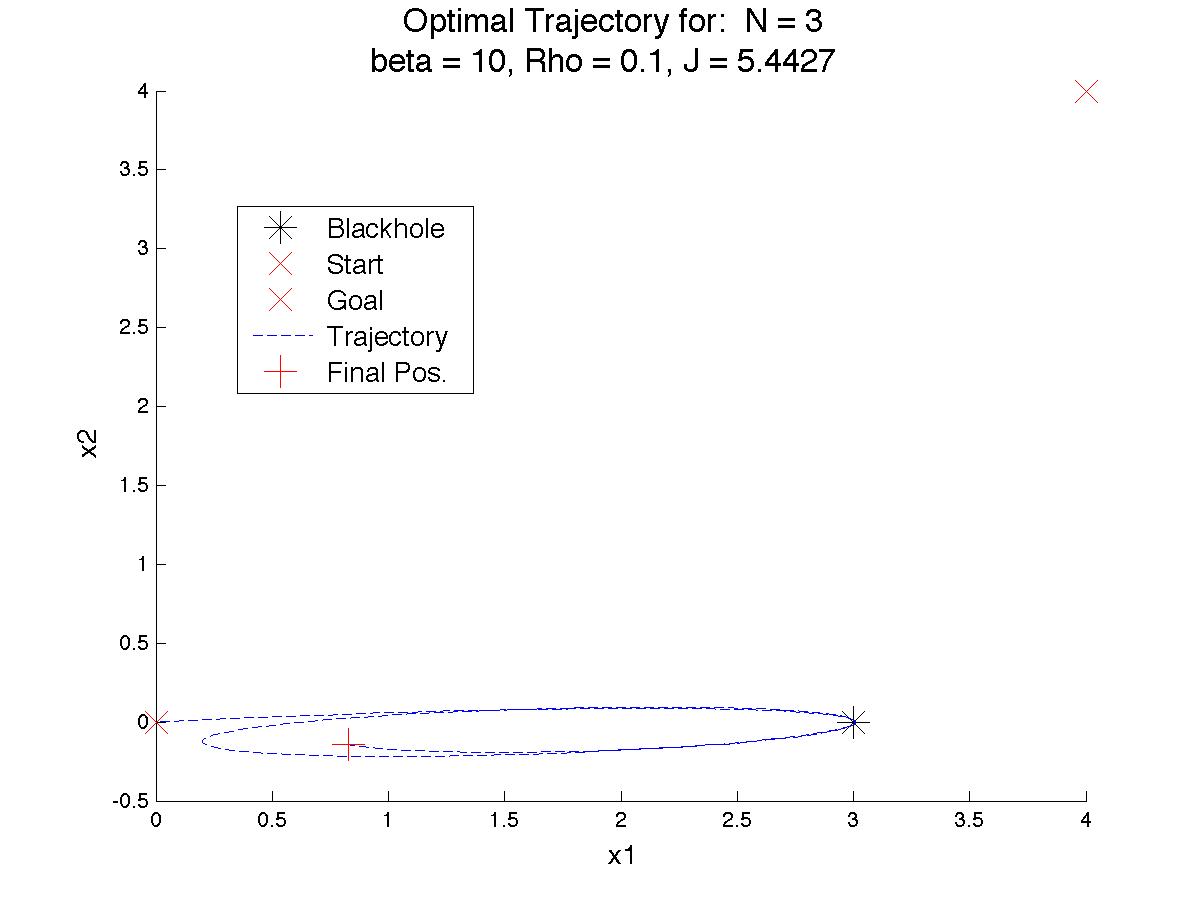
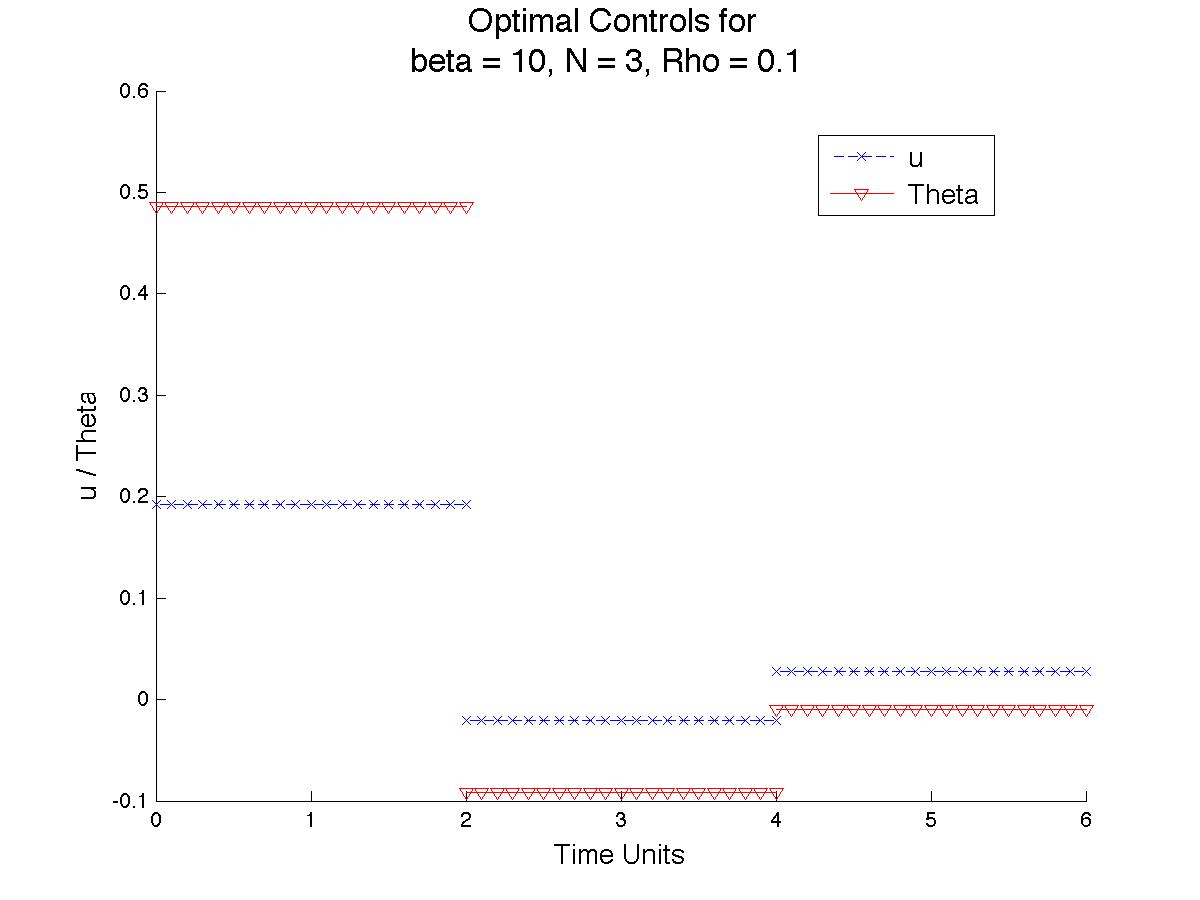
*Problem 1a:*

****

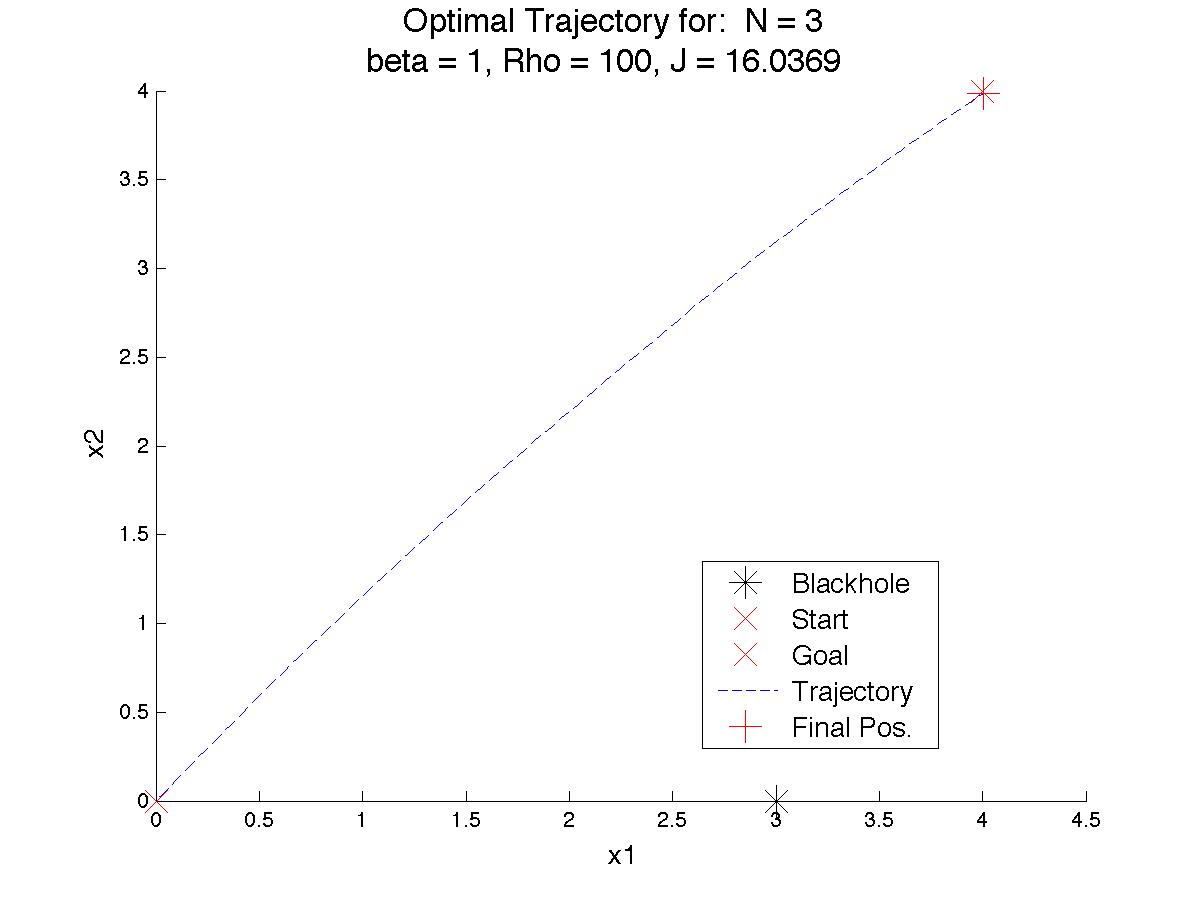
****

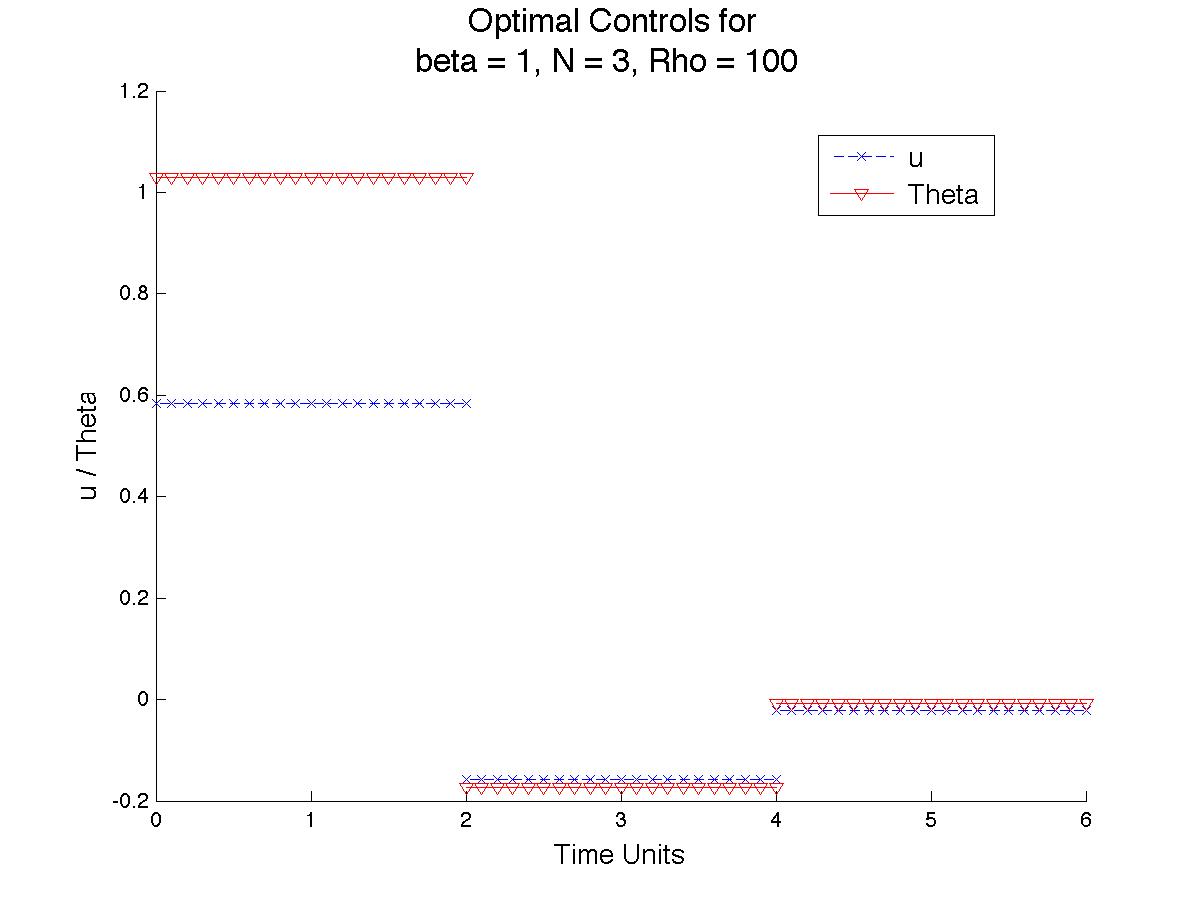
**b:**

****

****

**c:**

****

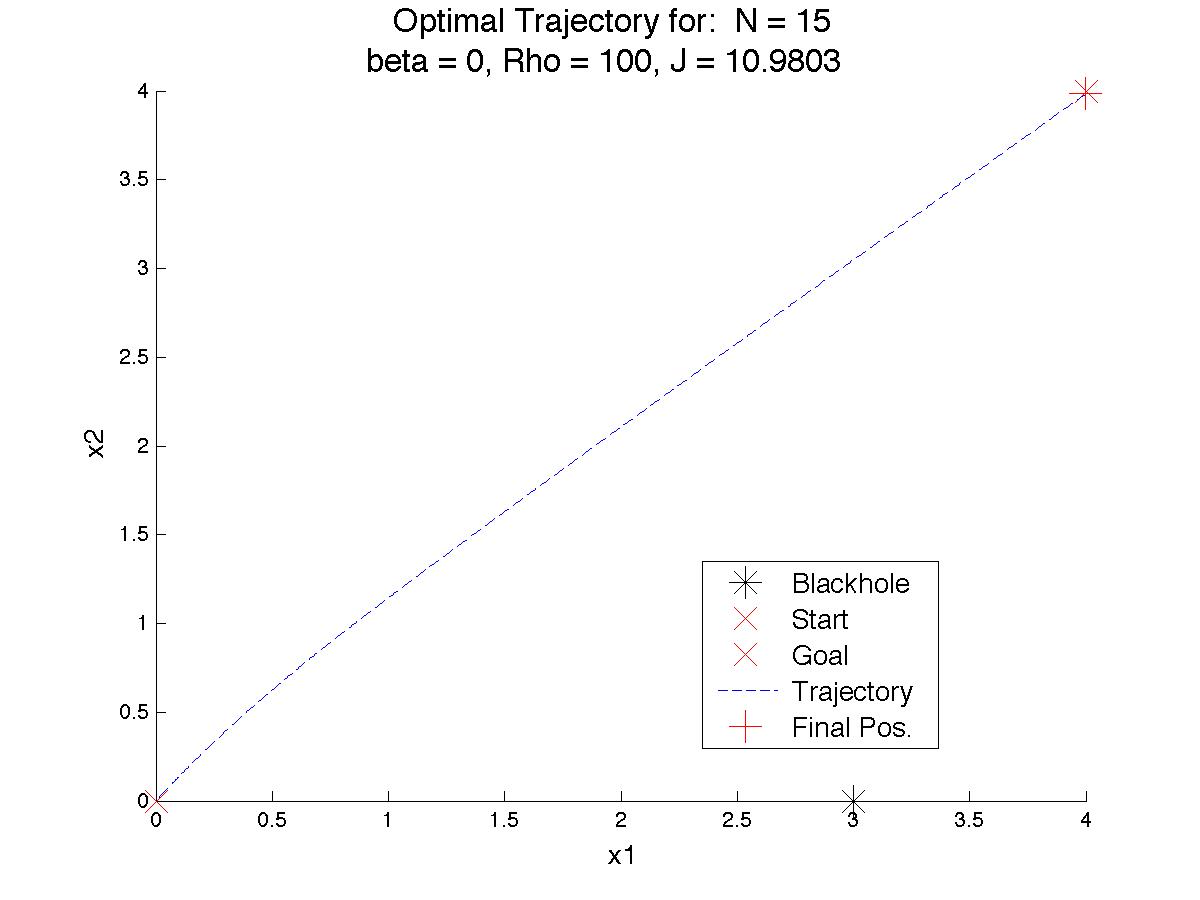
****

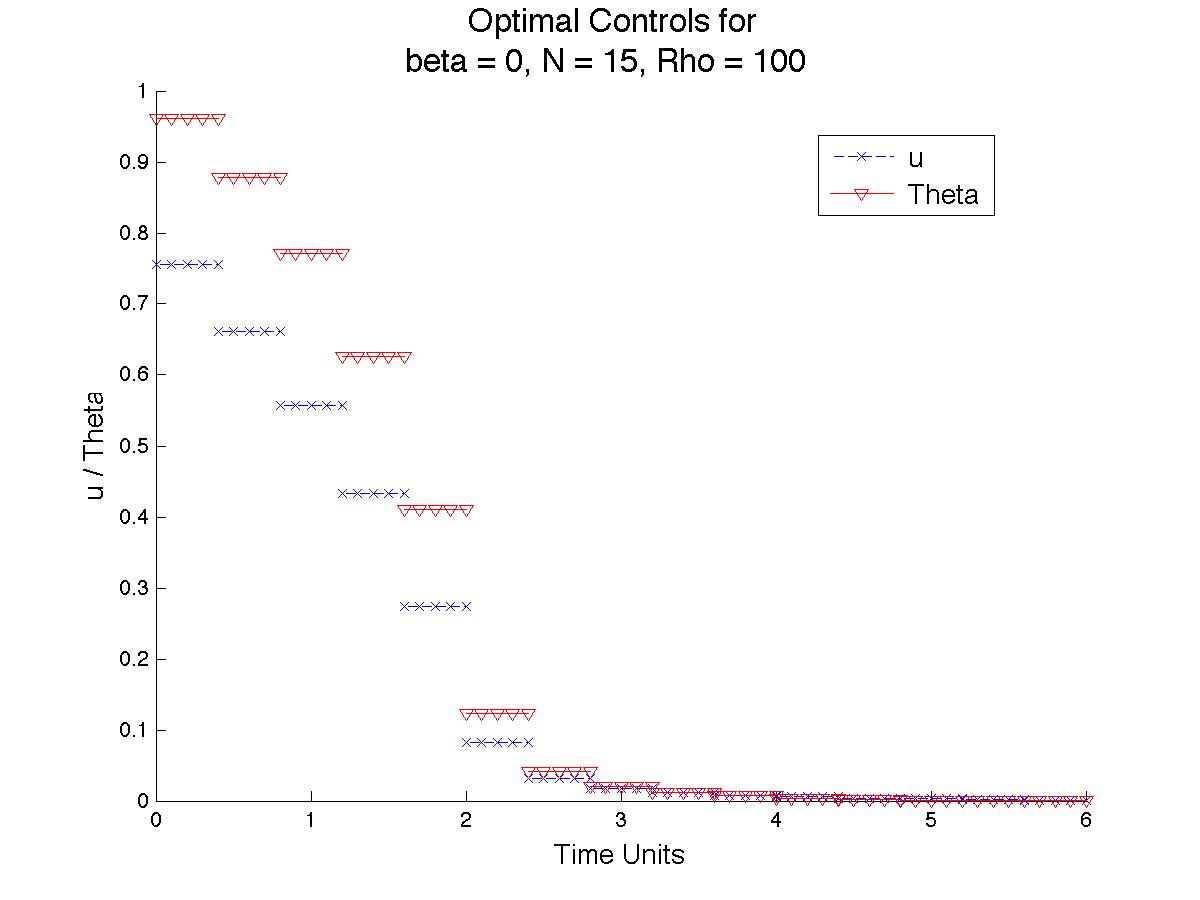
d:

For these three cases and all that follow, fmincon was provided with bounds for the controls (*u* ∈[-1,1] and θ ∈[-π, π]). In the case of (a), the effects of the black hole were not present, and so the optimal solution is essentially an initial impulse towards the objective, with subsequent phases simply corrections for not aiming directly at the objective due to the higher cost of using larger angles. The high cost of missing the target provides the incentive to actually do something with the 6 time units. In part (c), the black hole has a minor but noticeable affect, and so the trajectory takes this into consideration by providing the rocket with a larger initial angle and thrust, and more prominent guidance in the second stage. Naturally, the cost for this case is larger.

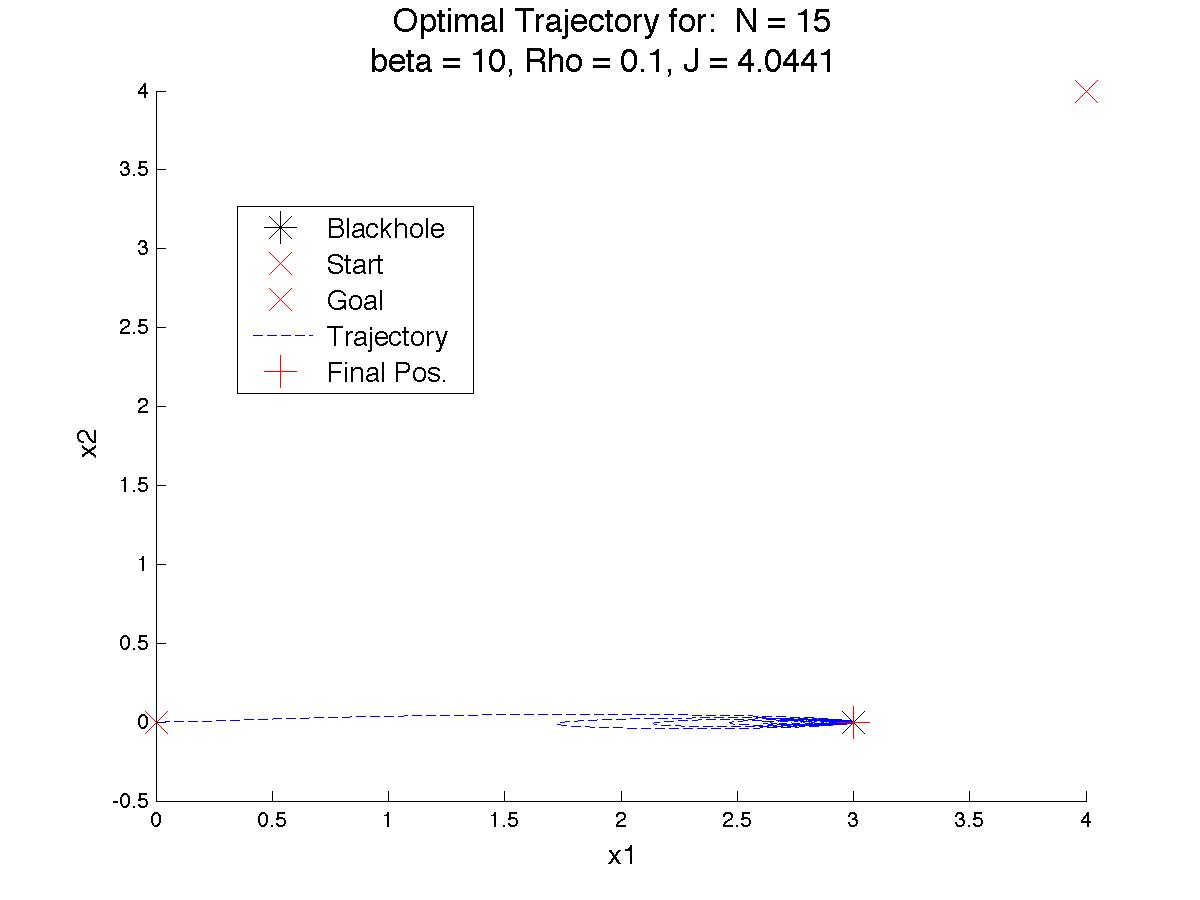
Case (b) presents a similar case for all control methods carried out in this HW. The optimal guidance found inevitably leaves the rocket shooting towards the black hole from the get go, and spiraling around it a few times, remaining far out of reach of the final goal. If we perform a simple calculation, we can see that the acceleration from the black hole initially in the x1 component is larger than our thrust can provide, implying that our rocket has no choice but to move towards the black hole with regards to the x1 component. From this and subsequent simulations for this case in the various portions of the problem, it seems that it is the case that we are inside the black hole’s “event horizon” with respect to our thrusters’ abilities; that is, our only choice in these cases is to move closer to, and eventually fall into, the black hole. In some instances of attempting to obtain an optimal solution, this form of demise manifests itself, always as a huge number of orbits around the black hole coupled with numerous error messages from the integrator. On the rare occasion that the ship does not enter the black hole, it always ends up in some position like the one shown above. Though the cost is less here than in the other two cases, it is not really representative of the “goal” of getting to some destination, and is deflated due to the lack of thrust needed to move anywhere.

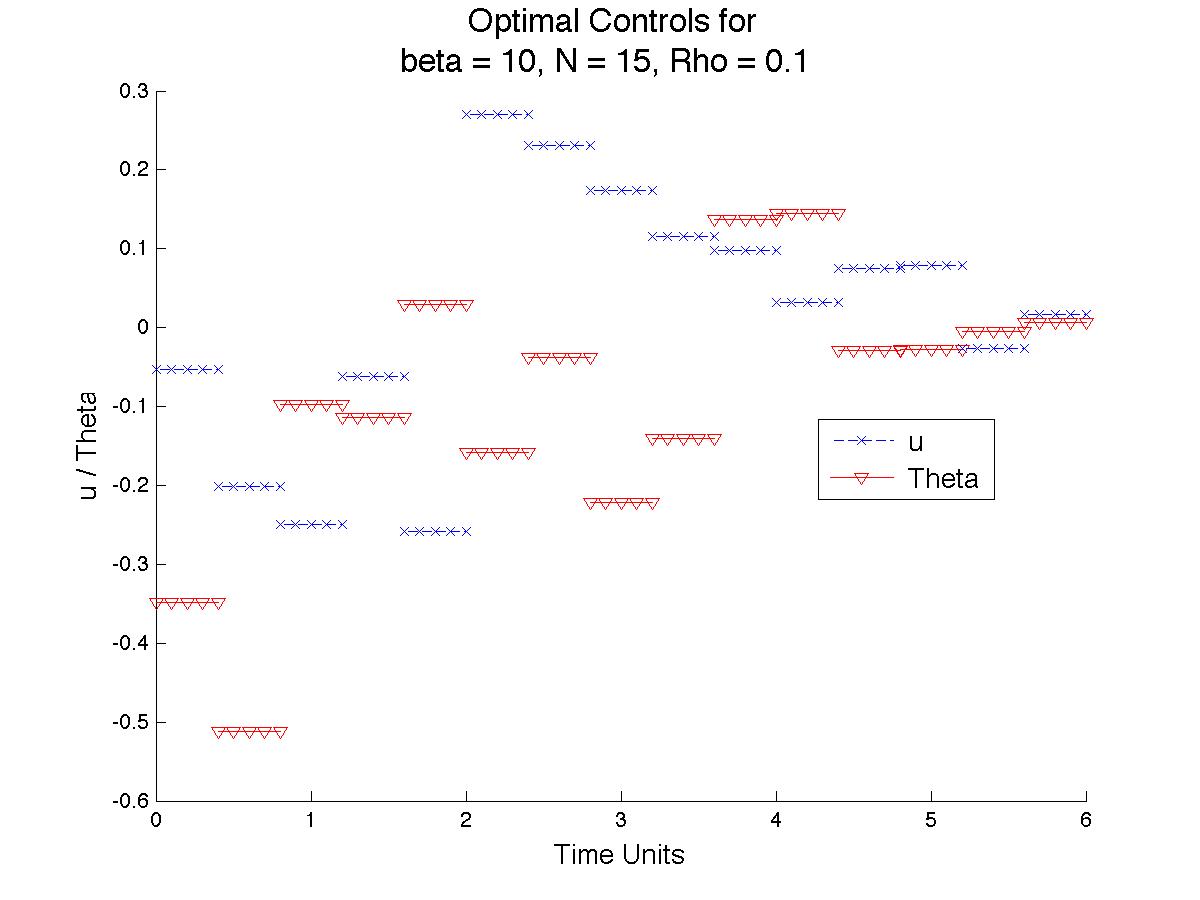
Part e:



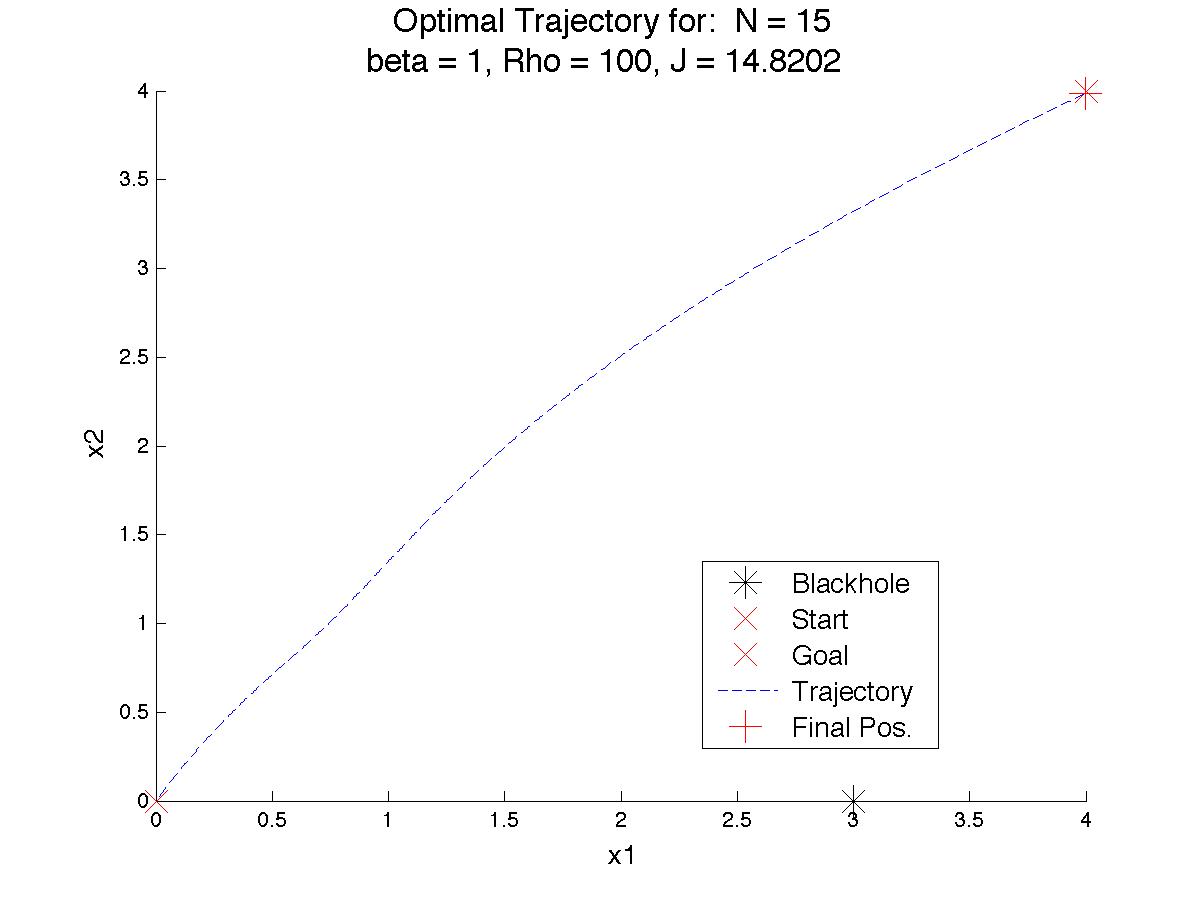
****

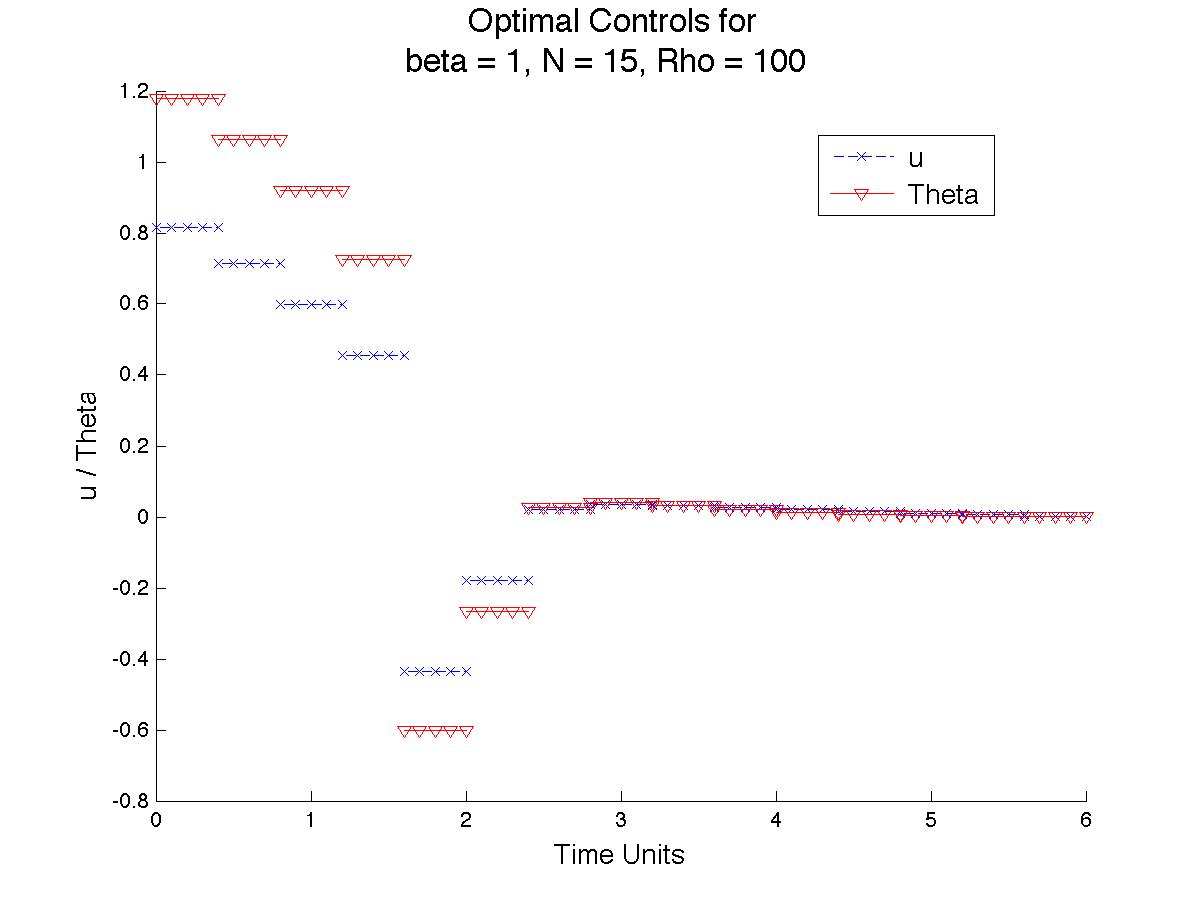
**b:**

****

****

**c:**

****

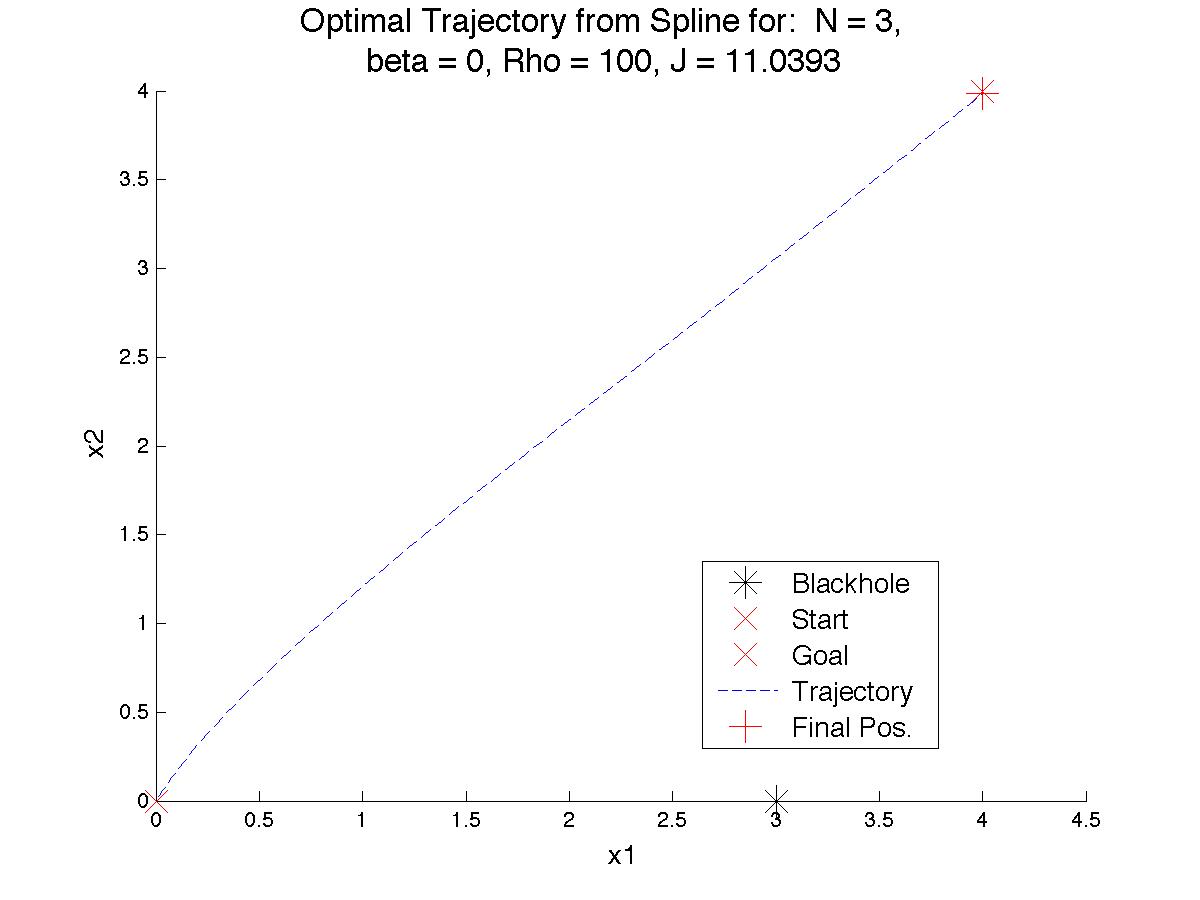
****

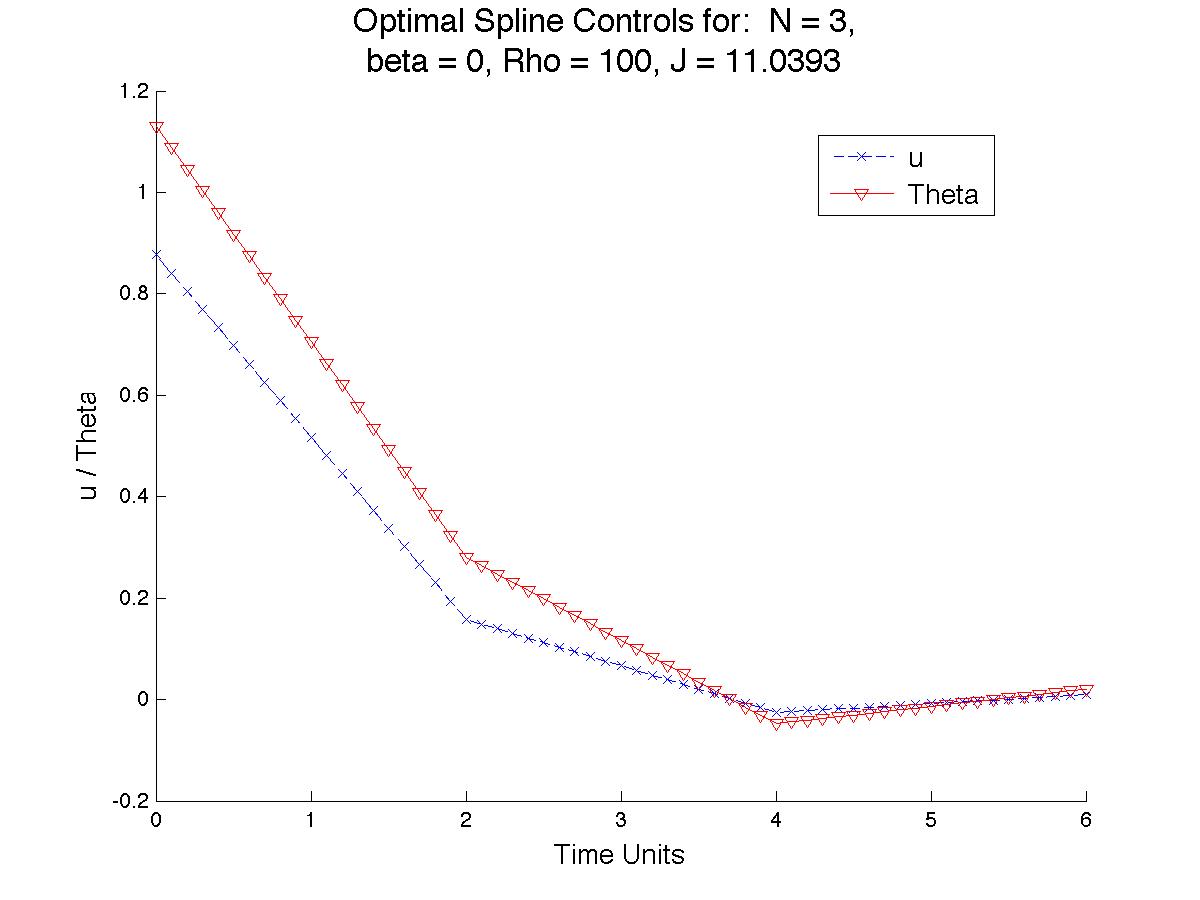
d)

The general observation from cases (a) and (c) is that by allowing more freedom in the control, we can reduce the cost given the same cost function and objective. We see that in both cases, the optimal controls are initially larger than in the three-piecewise case and decline in magnitude over time as smaller and smaller adjustments are made to reach the final destination. The shorter time spans for which the controls are active allow for this trade-off that results in a cost decline. In case (a) we also note a “notch” of sorts in the trajectory, utilizing this idea to shoot the rocket, initially, on a trajectory that leads above the target. We note that the control for (c) follows the same pattern as (a), except that fmincon has settled on negative values for the angle and thrust values, which result in the same control as choosing the positive values for both elements.

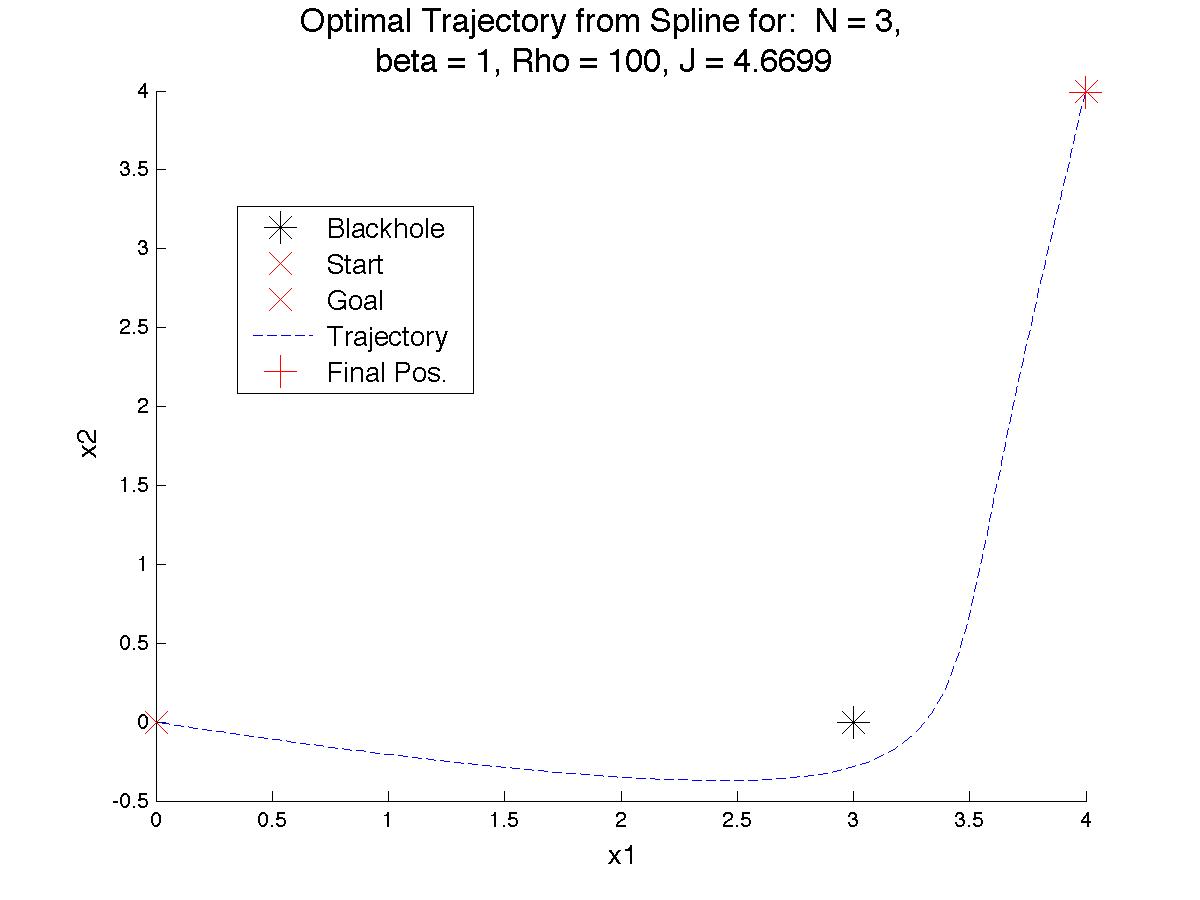
Case (b) again shows the inevitable demise of the rocket, given the conditions. In this case, we show an instance of spiraling. The erratic control seems to indicate some attempt by the integrator / optimizer combination to evade a singularity obtained by the intersection of the rocket-black hole pair. In this instance, MATLAB acknowledges some gap between the locations of the two objects, though for practical purposes, the ship is in the black hole.

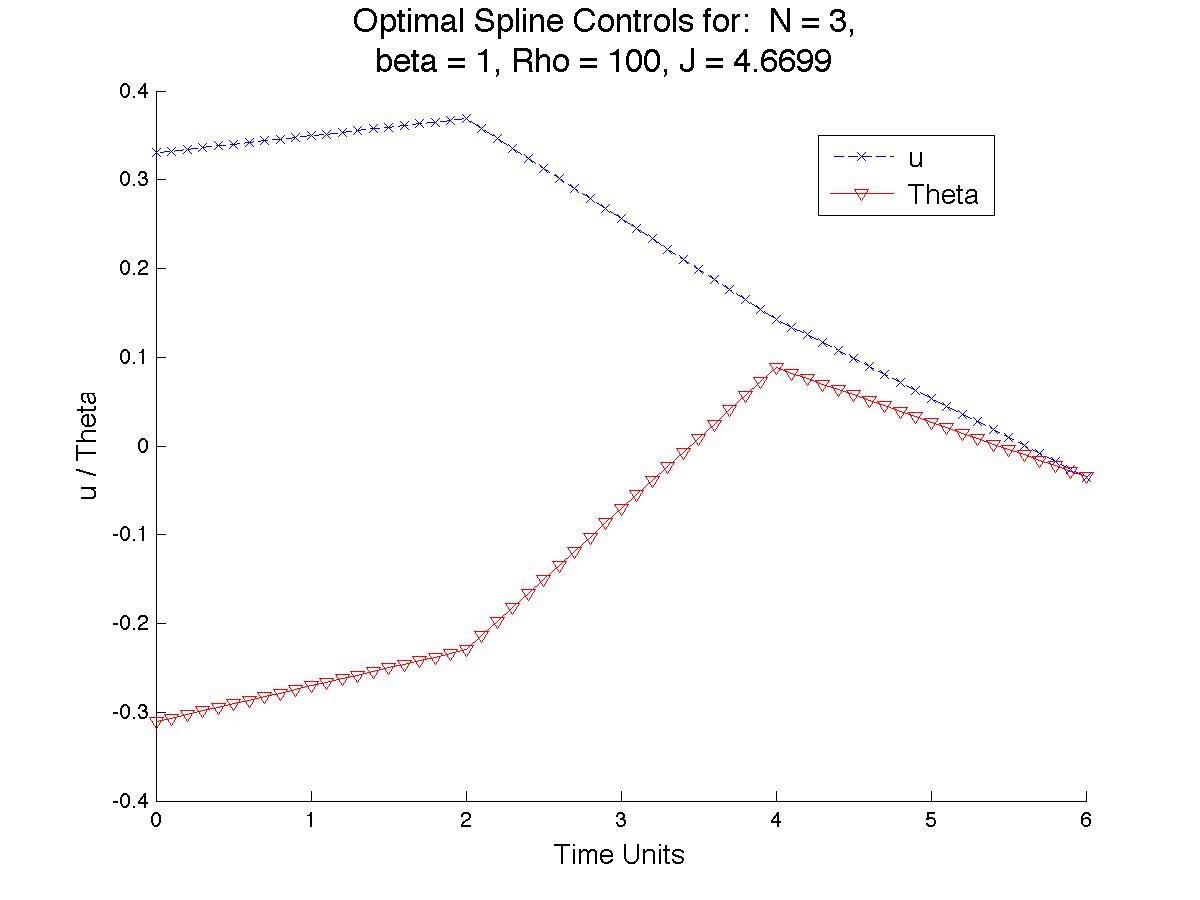
*Problem 2a:*

****

****

**c:**

****

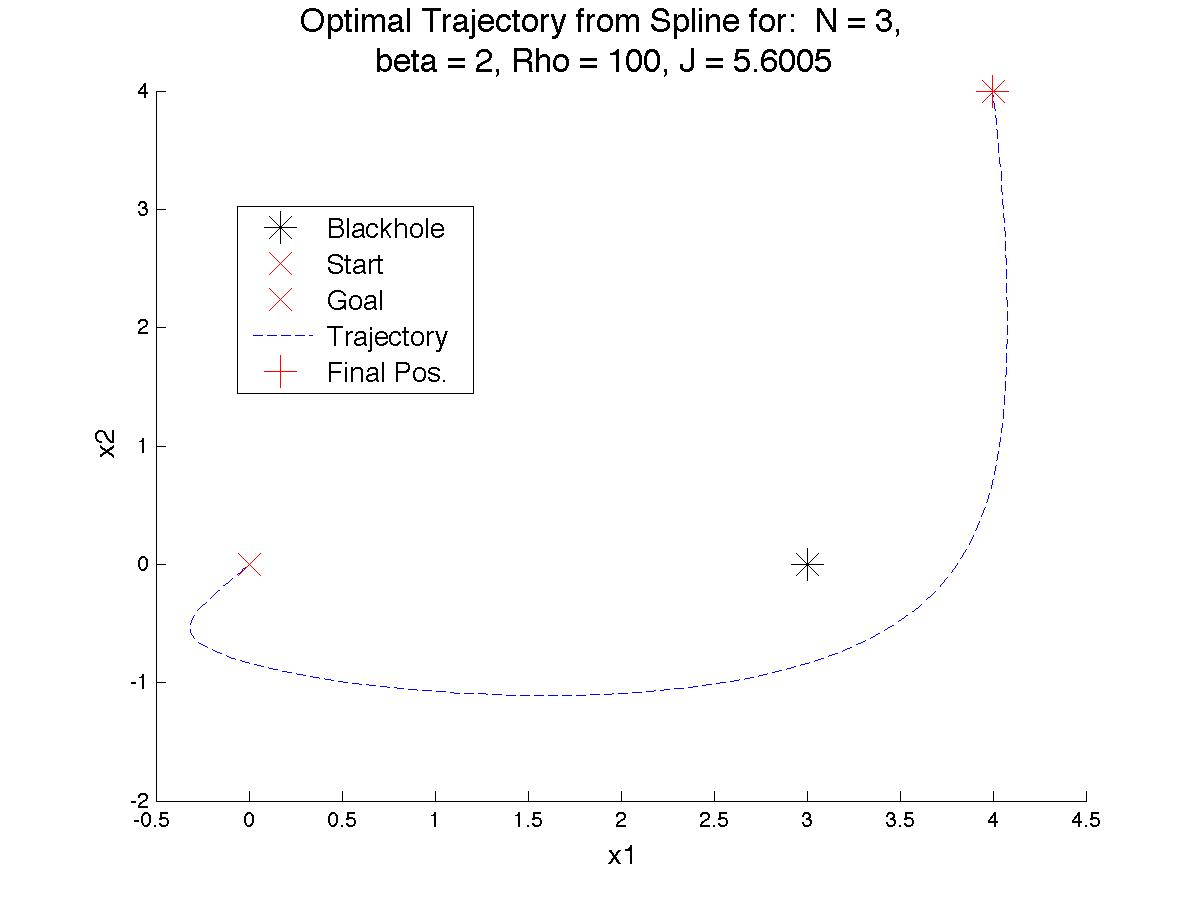
****

d:

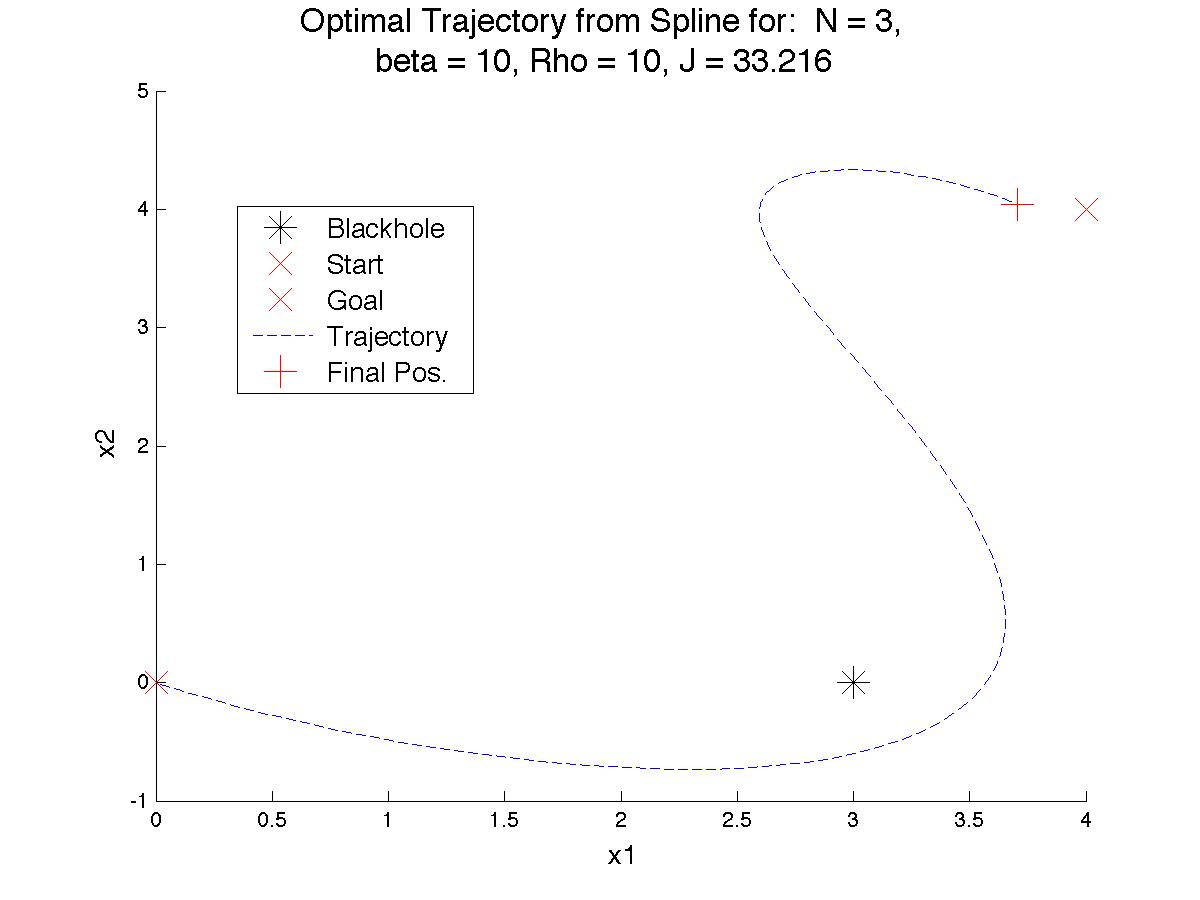
The diagrams for part (b) are omitted for brevity; the rocket’s trajectory mimics the other two cases. The results for part (a) again show a semi-direct path to the objective location, again with a large initial push pointed above the final position, followed by the tapering off of controls to reduce the overall cost from the two previous cases. The reduced cost is a result of the control’s ability to change continuously over time, as opposed to being restricted to discrete instances.

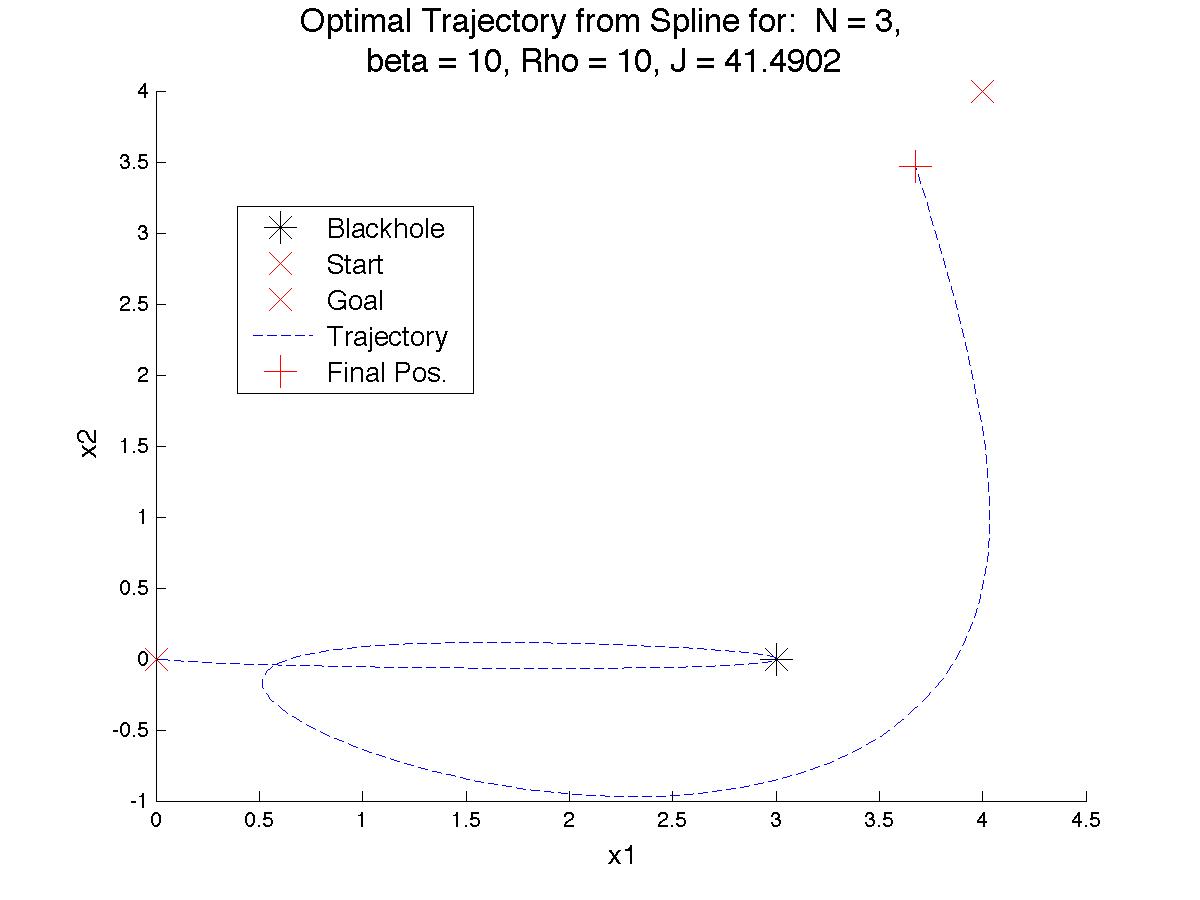
Part (c) displays more interesting behavior than the other two part (c) cases. Due to the ability to continuously modify the controls, albeit in a linear fashion, the rocket is able to use the pull of the black hole to help propel itself to the final destination by slinging around it, resulting in a greatly reduced cost in comparison to the other two part (c) cases, and in comparison to the optimal solution for part (a). The ability to use this path in this case and not the previous (N = 15) case is likely due to the inability of the N = 15 case to modify the control of the ship fast enough to avoid collision with the black hole once the ship is in close enough proximity to sling around the black hole. As noted in class, the system dynamics do not allow for momentum transfer, and so the use of the black hole as a sling of sorts in this system has the drawback that any velocity gained by the rocket while approaching the black hole is lost as the rocket moves away from the black hole. As a side note, the solver initially settled on a path of the more traditional form observed in the previous instances of case (c) with a cost of similar value. Small perturbations from the control obtained in that case resulted in the presented trajectory and control.

*Problem 3:* For this problem, the spline problem is observed with different parameter values than in the above problems. For the first case, the time is extended to ten units; beta is set at 2 and rho at 100. The resulting trajectory is similar to that from 2c, as is the cost. We note, though, that the initial movement of the rocket is almost directly away from the objective, likely as a means to adjust flight time so that it approximately reaches the destination at T = 10. The trajectory plot is shown below:

****

Next, we revisit the beta = 10 case to see if we can determine the parameters that allow a successful transit from the origin to near the objective, instead of simply spiraling into the black hole. From the dynamics we see that allowing for a maximum *u* value of 30/27, or about 1.1, will definitely allow for escape, we start with that bound in the first figure, and see that we seem to be able to escape from the black hole and obtain a final position closer to the desired location than in previous cases. From the optimal control obtained from this case, we observed a maximum control value of around 1.063 being used. The second plot is of the trajectory obtained using 1.062 as the upper *u* bound (1.06 was too small). We see that though there is no contact between the black hole and the ship, the trajectory of the ship takes it within 0.01 spatial units of the black hole. Observing the trajectory and the previous case, it could be possible that calculation errors allow the trajectory presented to occur. Changing rho to 100 in both cases results in similar trajectories with endpoints closer to the desired location, but with a larger overall cost.





function x = runme2(N,Start,Val0,Const)

Tfin = 6;

options=optimset('Algorithm','interior-point');

lb = [-1;-pi;-1;-pi;-1;-pi;-1;-pi];

ub = -lb;

x = fmincon(@(Values)myfunSpline1(Values,Start,Tfin,N,Const),Val0,[],[],[],[],lb,ub, options);

[TotT,TotxSol] = myfunSpline(x,Start,Tfin,N,Const);

function [TotT,TotxSol] = myfunSpline(Values,x0,Tfin,N,Const)

beta = Const(1);

rho = Const(2);

res = Const(3);

TotxSol = [];

TotT = 0;

xStart = [x0,Values(1),Values(2)];

J = 0;

for Section = 1:N

U = [Values((2\*Section-1):(2\*Section))'; Values((2\*Section+1):(2\*Section+2))'];

[T,xSol] = ode45(@(t1,x1)DiffEqIn(t1,x1,U,beta,Tfin/N),[0,Tfin/N],xStart);

[~,JTmp] = ode45(@(t,j)DiffEqJ(t,j,U,Tfin/N),[0,Tfin/N],[0,U(1,:)]);

J = J + JTmp(end,1);

xStart = xSol(end,:);

TotT = [TotT; T+TotT(end)];

TotxSol = [TotxSol;xSol];

end

TotT = TotT(2:end);

J = J + rho\*((xSol(end,1)-4)^2 + (xSol(end,2)-4)^2);

%Make plot for Controls

figure;

title({['\fontsize{16} Optimal Spline Controls for: N = ' num2str(N) ', '];...

['beta = ' num2str(beta) ', Rho = ' num2str(rho) ', J = ' num2str(J)]});

xlabel('\fontsize{13} Time Units');

ylabel('\fontsize{13} u / Theta');

hold on

for S = 1:N

U = [Values((2\*S-1):(2\*S))'; Values((2\*S+1):(2\*S+2))'];

tint = ((S-1)\*(Tfin/N)):res:(S\*(Tfin/N));

u1 = U(1,1) + (U(2,1)-U(1,1))\*(0:res:(Tfin/N))/(Tfin/N);

u2 = U(1,2) + (U(2,2)-U(1,2))\*(0:res:(Tfin/N))/(Tfin/N);

plot(tint,u1,'--bx',...

tint,u2,'-rv');

end

legend('\fontsize{13} u',...

'\fontsize{13} Theta',...

'Location','Best');

hold off

%Make plot for Trajectory

figure;

title({['\fontsize{16} Optimal Trajectory from Spline for: N = ' num2str(N) ', '];...

['beta = ' num2str(beta) ', Rho = ' num2str(rho) ', J = ' num2str(J)]});

xlabel('\fontsize{13} x1');

ylabel('\fontsize{13} x2');

hold on

plot(3,0,'\*k','MarkerSize',16);

plot(0,0,'xr','MarkerSize',16);

plot(4,4,'xr','MarkerSize',16);

plot(TotxSol(:,1),TotxSol(:,2),'--b');

plot(TotxSol(end,1),TotxSol(end,2), '+r','MarkerSize',16);

legend('\fontsize{13} Blackhole',...

'\fontsize{13} Start',...

'\fontsize{13} Goal',...

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

'\fontsize{13} Final Pos.',...

'Location','Best');

function dxdy = DiffEqIn(~,x,U,beta,span)

%[T,Y] = solver(odefun,tspan,y0,options)

%U = [u1, theta1; u2, theta2]

u = diff(U)/span;

q = (beta / ((x(2)^2 + (x(1)-3)^2)^(3/2))).\*[3-x(1);-x(2)];

dxdy = [x(3);...

x(4);...

x(5)\*cos(x(6)) + q(1);...

x(5)\*sin(x(6)) + q(2);...

u(1);...

u(2)];

function dj = DiffEqJ(~,j,U,span)

%U = [u1, theta1; u2, theta2]

u = diff(U)/span;

dj = [(10\*j(2)^2 + 4\*j(3)^2);...

u(1);...

u(2)];

“myfunSpline1” is a bare-bones version of “myfunSpline” made simply for feeding into fmincon. The non-spline versions involve the same basic structure, but with the obvious simplifications.