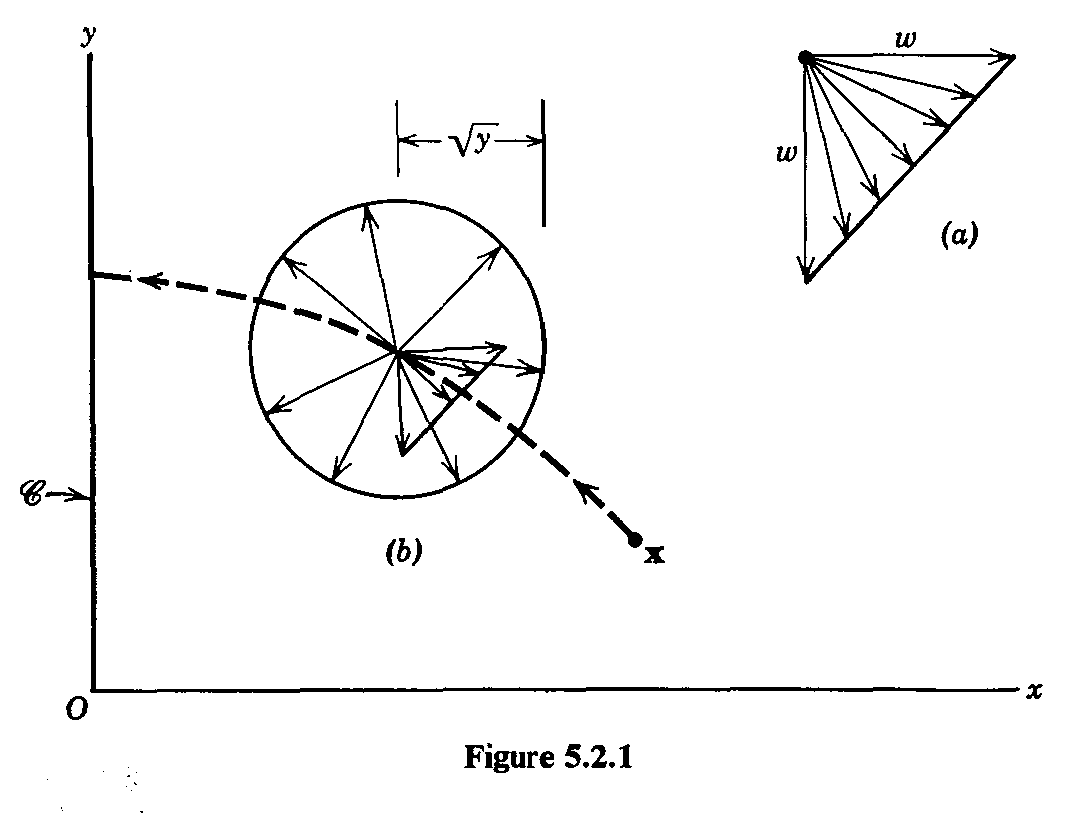
Dolichobrachistochrone



























 with 



syms lambda\_x1 lambda\_x2 alpha

eqn = -lambda\_x1\*sin(alpha)+lambda\_x2\*cos(alpha) == 0;

s = solve(eqn,alpha);

two solution, Which is correct ???

val =

-log((lambda\_x1^2 + lambda\_x2^2)^(1/2)/(lambda\_x1 - lambda\_x2\*1i))\*1i

val =

-log(-(lambda\_x1^2 + lambda\_x2^2)^(1/2)/(lambda\_x1 - lambda\_x2\*1i))\*1i

换一个方式

Fsolve

x\_1 = x(:,1);x\_2 = x(:,2);lambda\_x1 = x(:,3);lambda\_x2 = x(:,4);

beta = u(:,1);

result = zeros(size(lambda\_x1));

for i = 1:length(lambda\_x1)

options = optimoptions('fsolve','Display','none');

F = @(alpha) -lambda\_x1(i)\*sin(alpha)+lambda\_x2(i)\*cos(alpha);

x0 = -2;

[alpha] = fsolve(F,x0,options);

result(i) = alpha;

end

% disp(result)

cosalpha = cos(result);

sinalpha = sin(result);

dx(:,1) = sqrt(x\_2) .\* cosalpha + (beta + 1)./2;

dx(:,2) = sqrt(x\_2) .\* sinalpha + (beta - 1)./2;

dx(:,3) = 0;

dx(:,4) = -(lambda\_x1.\*cosalpha+lambda\_x2.\*sinalpha)./(2.\*sqrt(x\_2));

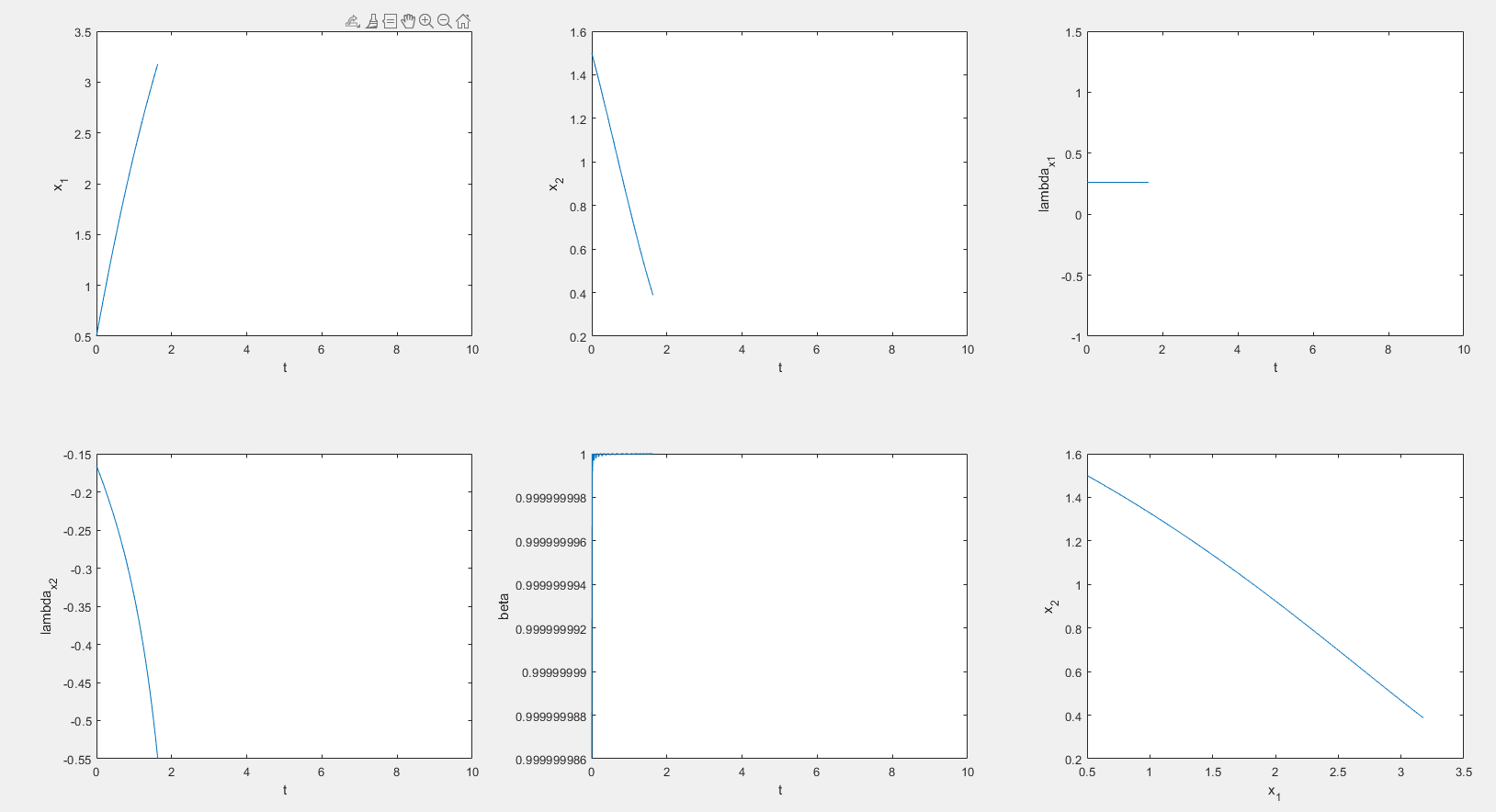
% Initial conditions for system

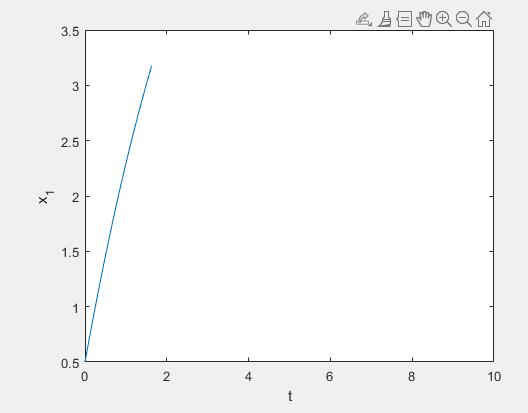
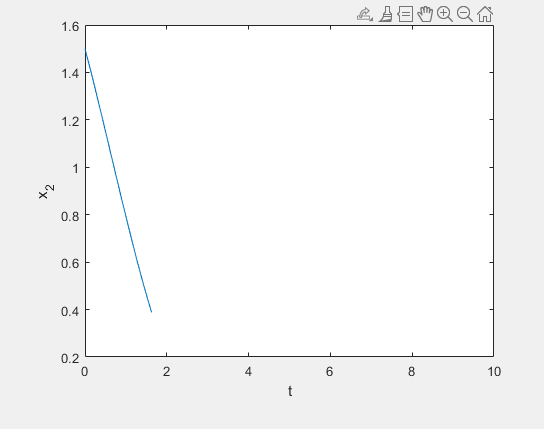
problem.states.x0=[0.5 1.5 1 -1];

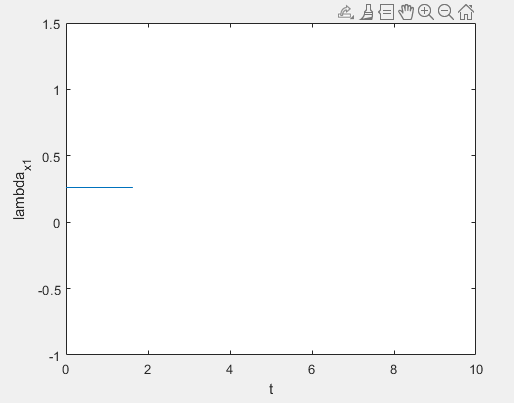
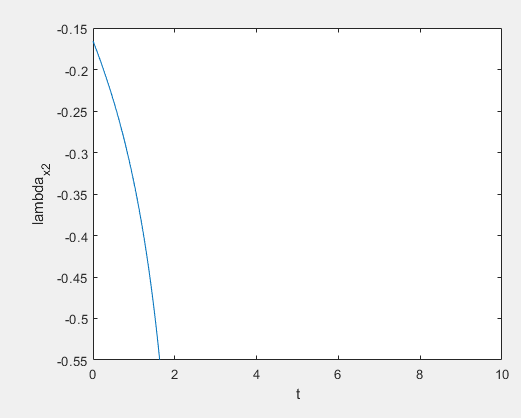
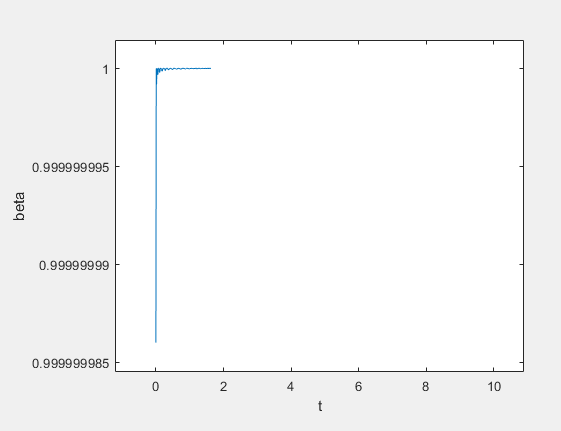
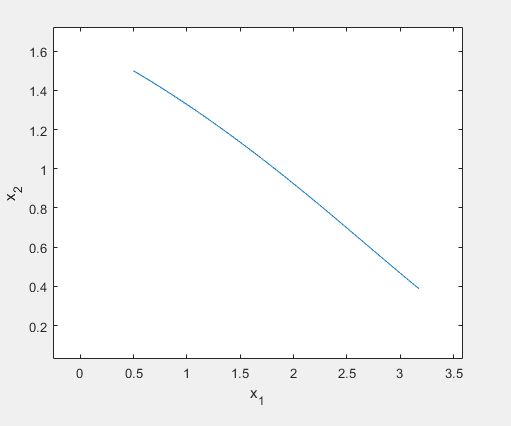
% Initial conditions for system. Bounds if x0 is free s.t. x0l=< x0 <=x0u

problem.states.x0l=[0.5 1.5 -10 -10];

problem.states.x0u=[0.5 1.5 10 10];



EXIT: Optimal Solution Found.

computation time:

442.3191

结果完全不正确

计算时间过长。

result = zeros(size(lambda\_x1));

for i = 1:length(lambda\_x1)

options = optimoptions('fsolve','Display','none');

F = @(alpha) -lambda\_x1(i)\*sin(alpha)+lambda\_x2(i)\*cos(alpha);

x0 = -2;

[alpha] = fsolve(F,x0,options);

result(i) = alpha;

end

% disp(result)

cosalpha = cos(result);

sinalpha = sin(result);

子程序

Dolichobrachistochrone\_Dynamics\_Sim

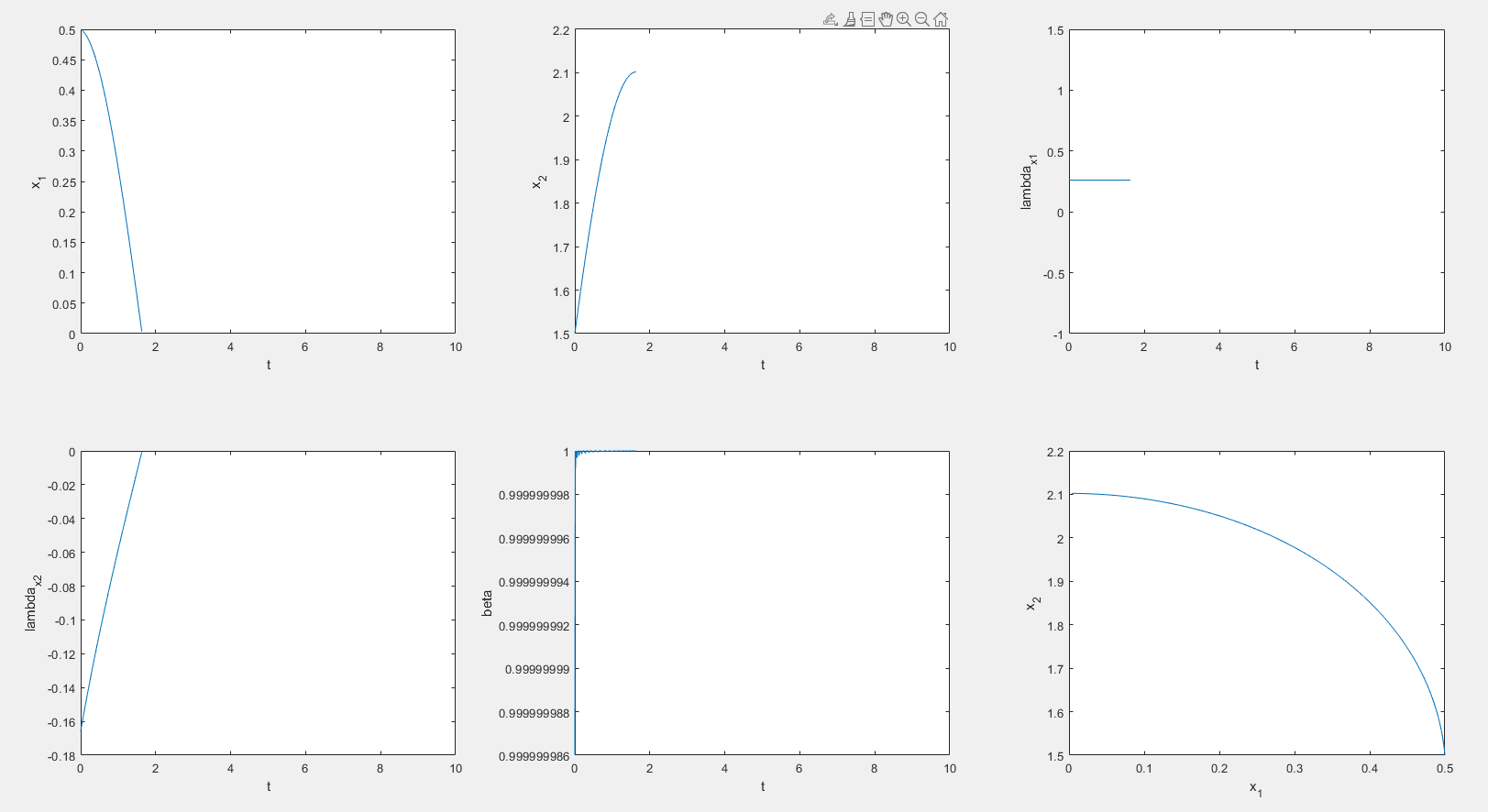
中的代码写错了

x0 = -2;

现在更正为

x0 = 2;

simulate的结果正确了



% Initial conditions for system

problem.states.x0=[1.5 1.5 1 -2];

% Initial conditions for system. Bounds if x0 is free s.t. x0l=< x0 <=x0u

problem.states.x0l=[1.5 1.5 -10 -10];

problem.states.x0u=[1.5 1.5 10 10];

%% Solve with mesh refinement

clear all;close all;format compact;

[problem,guess]=Dolichobrachistochrone; % Fetch the problem definition

options= problem.settings(5); % Get options and solver settings

[solution,MRHistory]=solveMyProblem( problem,guess,options);

[ tv, xv, uv ] = simulateSolution( problem, solution, 'ode113', 0.01 );

8257r-3.4811812e+00 1.66e-01 7.99e-05 -9.3 7.08e-12 - 1.00e+00 3.81e-06h 19

8258r-3.4811812e+00 1.66e-01 7.99e-05 -9.3 7.08e-12 - 1.00e+00 3.81e-06h 19

8259r-3.4811812e+00 1.66e-01 7.99e-05 -9.3 7.08e-12 - 1.00e+00 3.81e-06h 19

iter objective inf\_pr inf\_du lg(mu) ||d|| lg(rg) alpha\_du alpha\_pr ls

8260r-3.4811812e+00 1.66e-01 7.99e-05 -9.3 7.08e-12 - 1.00e+00 3.81e-06h 19

8261r-3.4811812e+00 1.66e-01 7.99e-05 -9.3 7.08e-12 - 1.00e+00 3.81e-06h 19

8262r-3.4811812e+00 1.66e-01 7.99e-05 -9.3 7.08e-12 - 1.00e+00 3.81e-06h 19

8263r-3.4811812e+00 1.66e-01 7.99e-05 -9.3 7.08e-12 - 1.00e+00 3.81e-06h 19

8264r-3.4811812e+00 1.66e-01 6.99e-05 -9.3 7.08e-12 - 1.00e+00 1.00e+00w 1

改变初值，从17点计算到20点，一直计算都没有结果。

这还仅仅是只有5个网格的情况，增加网格数量，问题计算一次会更慢。

result = zeros(size(lambda\_x1));

for i = 1:length(lambda\_x1)

options = optimoptions('fsolve','Display','none');

F = @(alpha) -lambda\_x1(i)\*sin(alpha)+lambda\_x2(i)\*cos(alpha);

alpha\_0 = 2;

[alpha] = fsolve(F,alpha\_0,options);

result(i) = alpha;

end

disp('here')

cosalpha = cos(result);

sinalpha = sin(result);

>> main\_Dolichobrachistochrone

Determine sparsity structure

Formatting matrices for the hermite-simpson approximation

Generate finite-difference pertubation vectors

here

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

This program contains Ipopt, a library for large-scale nonlinear optimization.

Ipopt is released as open source code under the Eclipse Public License (EPL).

For more information visit http://projects.coin-or.org/Ipopt

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

This is Ipopt version 3.12.9, running with linear solver ma57.

Number of nonzeros in equality constraint Jacobian...: 16

Number of nonzeros in inequality constraint Jacobian.: 426

Number of nonzeros in Lagrangian Hessian.............: 446

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

Total number of variables............................: 43

variables with only lower bounds: 0

variables with lower and upper bounds: 43

variables with only upper bounds: 0

Total number of equality constraints.................: 2

Total number of inequality constraints...............: 36

inequality constraints with only lower bounds: 0

inequality constraints with lower and upper bounds: 36

inequality constraints with only upper bounds: 0

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

iter objective inf\_pr inf\_du lg(mu) ||d|| lg(rg) alpha\_du alpha\_pr ls

0 -5.0000000e+00 6.08e-01 0.00e+00 0.0 0.00e+00 - 0.00e+00 0.00e+00 0

WARNING: Problem in step computation; switching to emergency mode.

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

1r-5.0000000e+00 6.08e-01 9.98e+02 0.0 0.00e+00 20.0 0.00e+00 0.00e+00R 1

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

2r-4.9999998e+00 6.08e-01 9.83e+02 1.3 5.26e+02 - 4.67e-02 4.70e-10f 2

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

here

584 OptionsStruct.ProblemdefOptions = obj.ProblemdefOptions;

Matlab exception:

Unknown exception

\*\*\* Error using Ipopt Matlab interface: \*\*\*

There was an error when executing the Hessian callback function.

Exception of type: IpoptException in file "Unknown File" at line -1:

Exception message: Unknown Exception caught in Ipopt

here

303 end

优化一次需要对微分方程调用很多次（大约是105次），而微分方程中有非线性方程的优化迭代求解。慢上加慢。