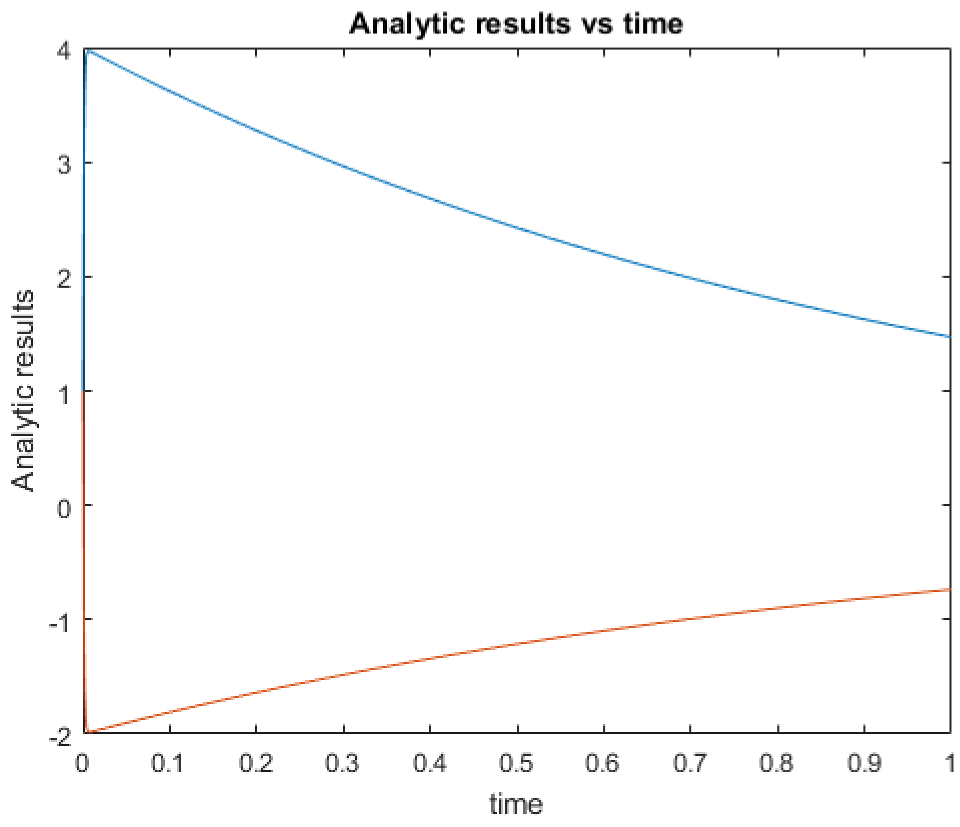
Table of Contents

Ordinary Differential Equations

ECE 523 hw10 Ordinary Differential Equations Homework Author:Qihao He Due date: 11/28/2016

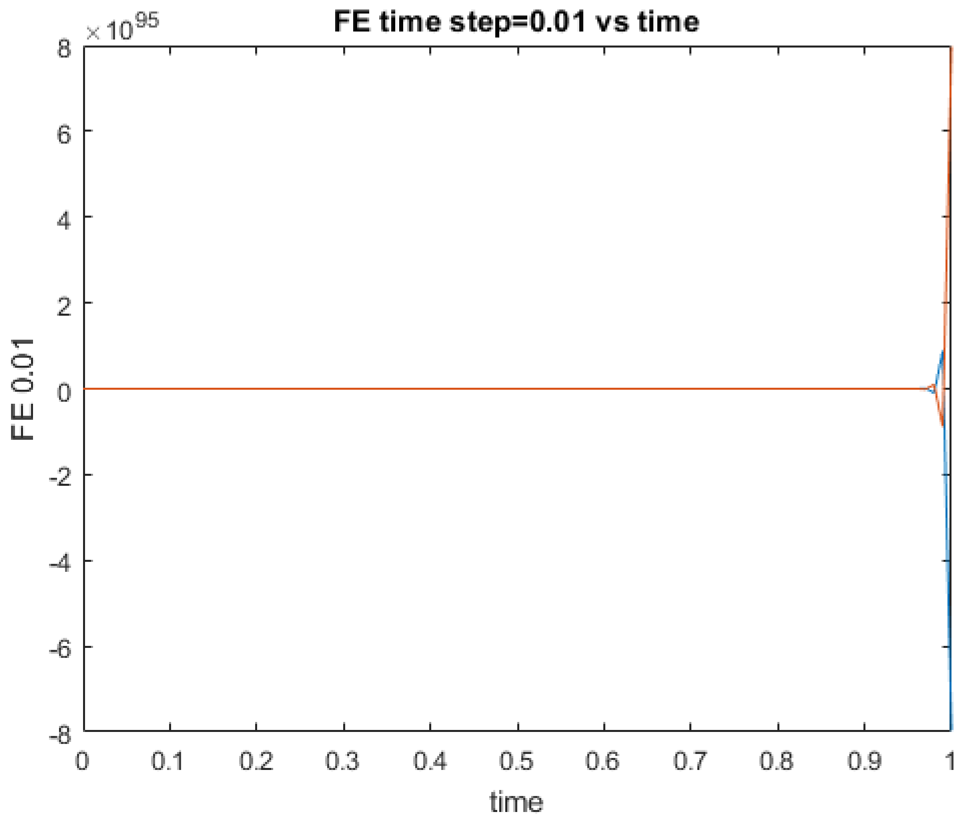
```
%-----~ Initialize Variables ~----
clear;
xo=[1;1];
M=[998 1998;-999 -1999];% lambda
%-----~ Analytic solution ~----
f1=@(t) 4*exp(-t)-3*exp(-1e3*t);
f2=@(t) -2*exp(-t)+3*exp(-1e3*t);
f=@(t) [f1(t);f2(t)];
t=linspace(0,1,1000);
AS=f(t);
AS=AS.';%transpose
figure(1)
plot(t,AS);
title('Analytic results vs time');
xlabel('time');
ylabel('Analytic results');
```

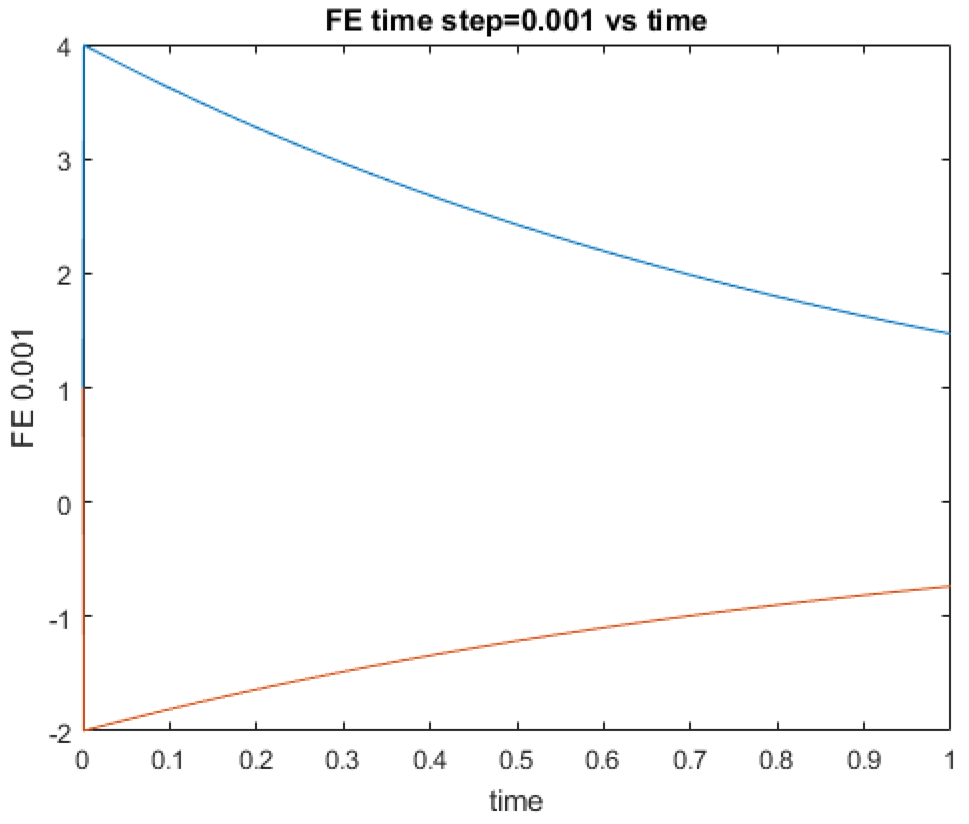


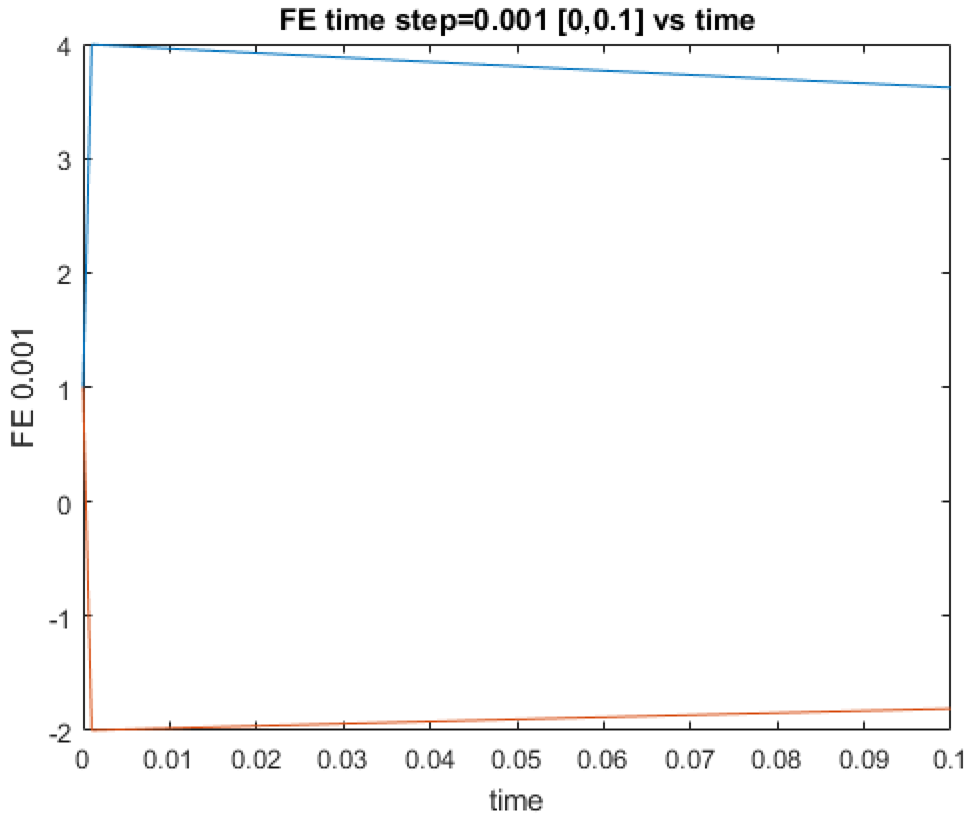
Question1 Forward Euler Algorithm

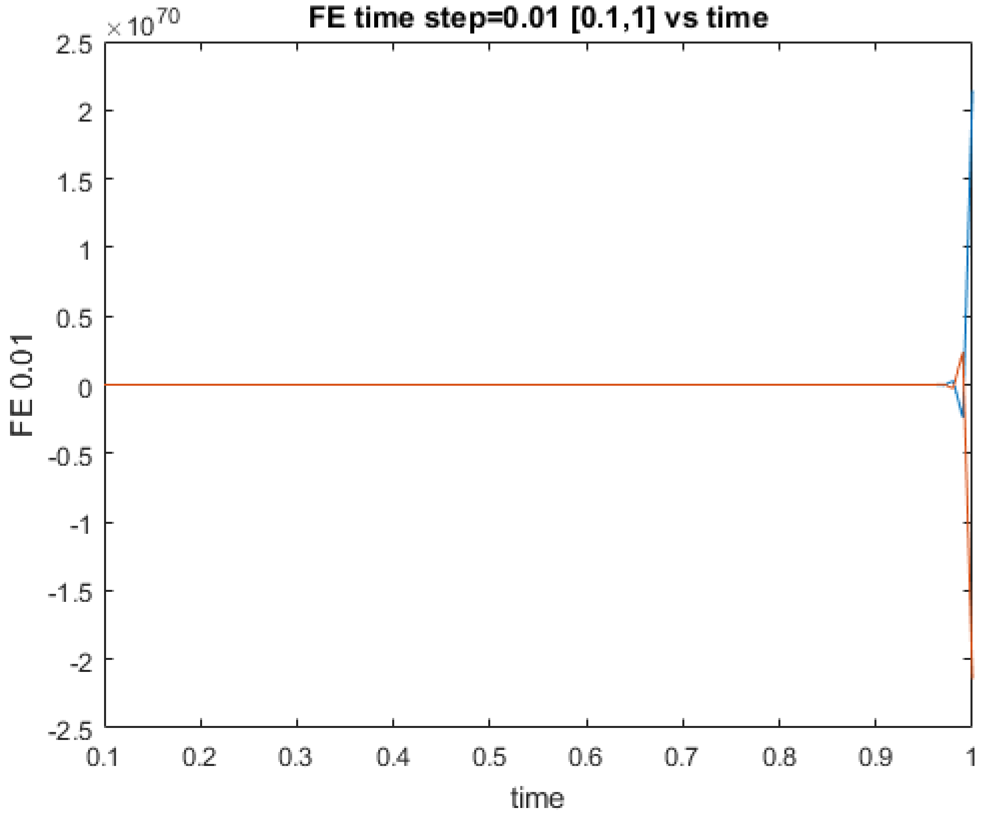
```
[FE_a_tspan,FE_a_xc,FE_a_exitflag]=FE(M,xo,0.01,0,1);
[FE_b_tspan,FE_b_xc,FE_b_exitflag]=FE(M,xo,0.001,0,1);
[FE_c1_tspan,FE_c1_xc,FE_c1_exitflag]=FE(M,xo,0.001,0,0.1);
[FE_c2_tspan,FE_c2_xc,FE_c2_exitflag]=FE(M,FE_c1_xc(:,size(FE_c1_tspan,2)),0.01,0.
figure(2)
plot(FE_a_tspan,FE_a_xc);
title('FE time step=0.01 vs time');
xlabel('time');
ylabel('FE 0.01')
figure(3)
plot(FE_b_tspan,FE_b_xc);
title('FE time step=0.001 vs time');
xlabel('time');
ylabel('FE 0.001')
figure(4)
plot(FE_c1_tspan,FE_c1_xc);
title('FE time step=0.001 [0,0.1] vs time');
xlabel('time');
ylabel('FE 0.001')
figure(5)
plot(FE_c2_tspan,FE_c2_xc);
title('FE time step=0.01 [0.1,1] vs time');
xlabel('time');
```

```
ylabel('FE 0.01')

% The FE results are stable when the time step is selected as 0.001.
% Because the time constant of the ODEs is 1s and 1ms,the FE algorithm
 is
% unstable for step sizes greater than 2 time constants.
% Making the time step back to 0.01 after using 0.001 as time step for
 a
% certain time does not guarantee the stability of the resutls.
```




FE time step=0.001 vs time
FE 0.001
time
4
3
2
1
0
-1
-2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

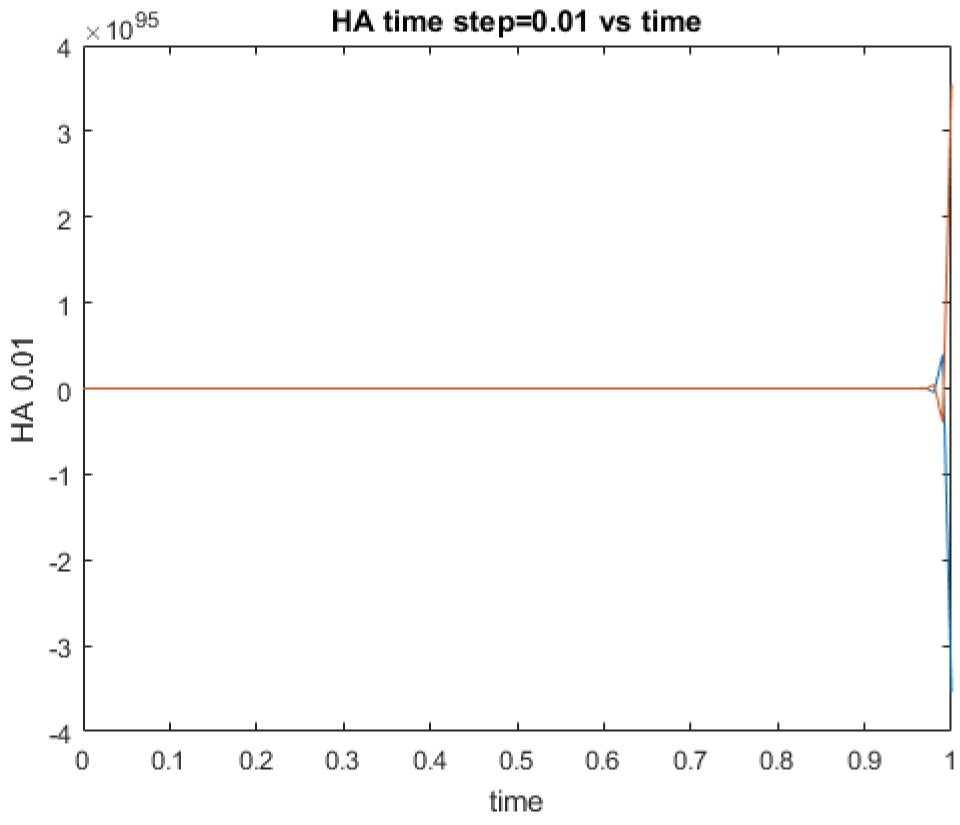

FE time step=0.001 [0,0.1] vs time
FE 0.001
time
4
3
2
1
0
-1
-2
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1

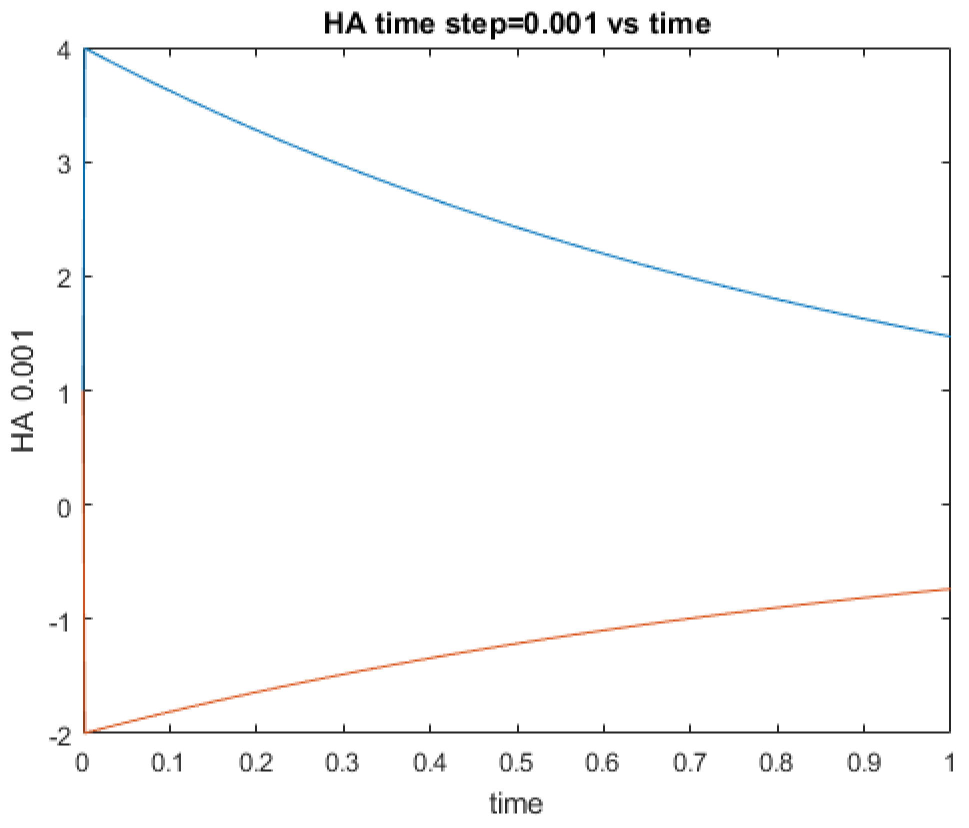


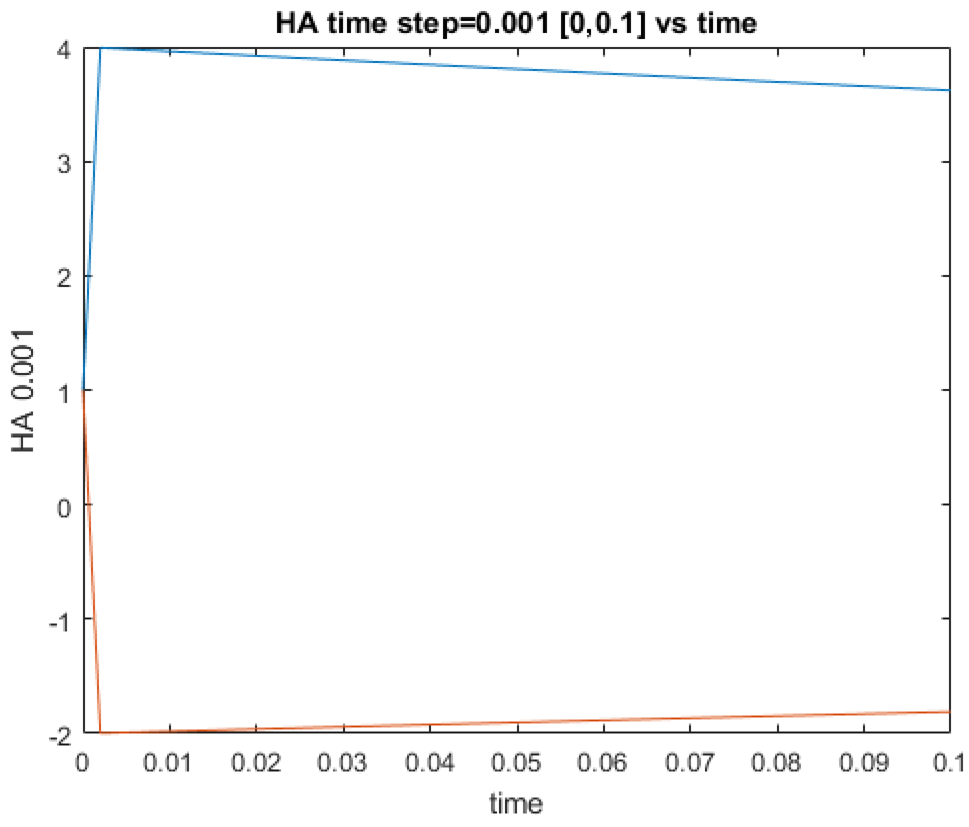
Question2 Heun Algorithm

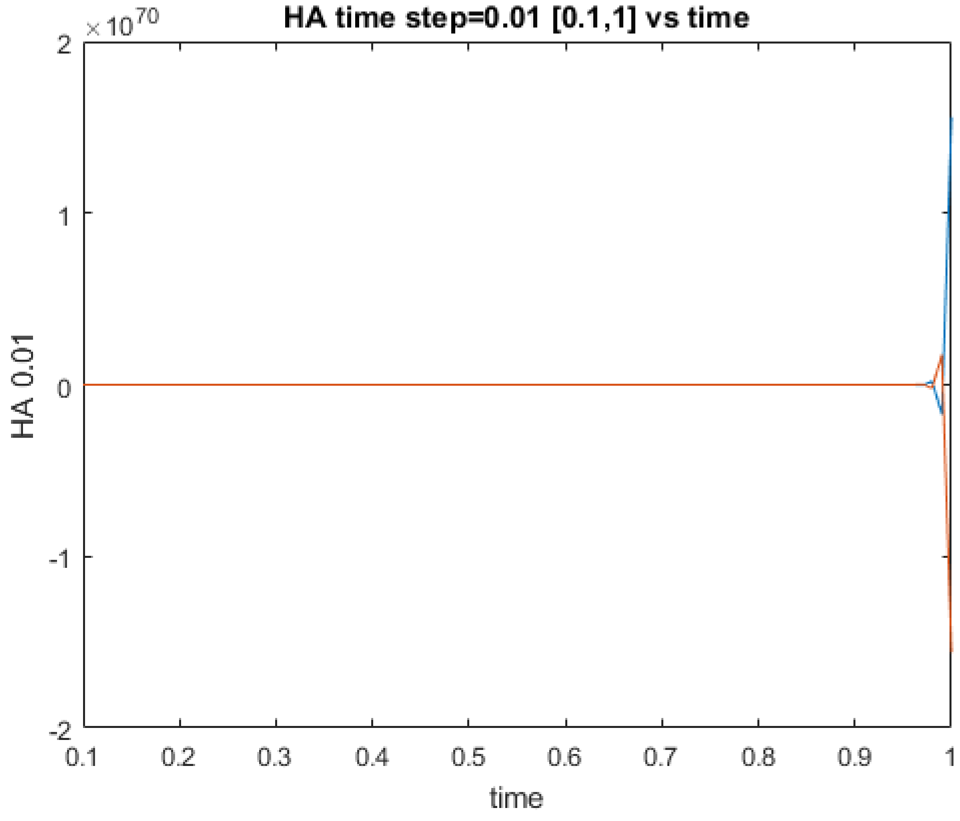
```
[HA_a_tspan,HA_a_xc,HA_a_exitflag]=HA(M,xo,0.01,0,1);
[HA_b_tspan,HA_b_xc,HA_b_exitflag]=HA(M,xo,0.001,0,1);
[HA_c1_tspan,HA_c1_xc,HA_c1_exitflag]=HA(M,xo,0.001,0,0.1);
[HA_c2_tspan,HA_c2_xc,HA_c2_exitflag]=HA(M,HA_c1_xc(:,size(HA_c1_tspan,2)),0.01,0.
figure(6)
plot(HA_a_tspan,HA_a_xc);
title('HA time step=0.01 vs time');
xlabel('time');
ylabel('HA 0.01')
figure(7)
plot(HA_b_tspan,HA_b_xc);
title('HA time step=0.001 vs time');
xlabel('time');
ylabel('HA 0.001')
figure(8)
plot(HA_c1_tspan,HA_c1_xc);
title('HA time step=0.001 [0,0.1] vs time');
xlabel('time');
ylabel('HA 0.001')
figure(9)
plot(HA_c2_tspan,HA_c2_xc);
title('HA time step=0.01 [0.1,1] vs time');
xlabel('time');
```

```
ylabel('HA 0.01')
```




HA time step=0.001 vs time
HA 0.001
time
4
3
2
1
0
-1
-2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

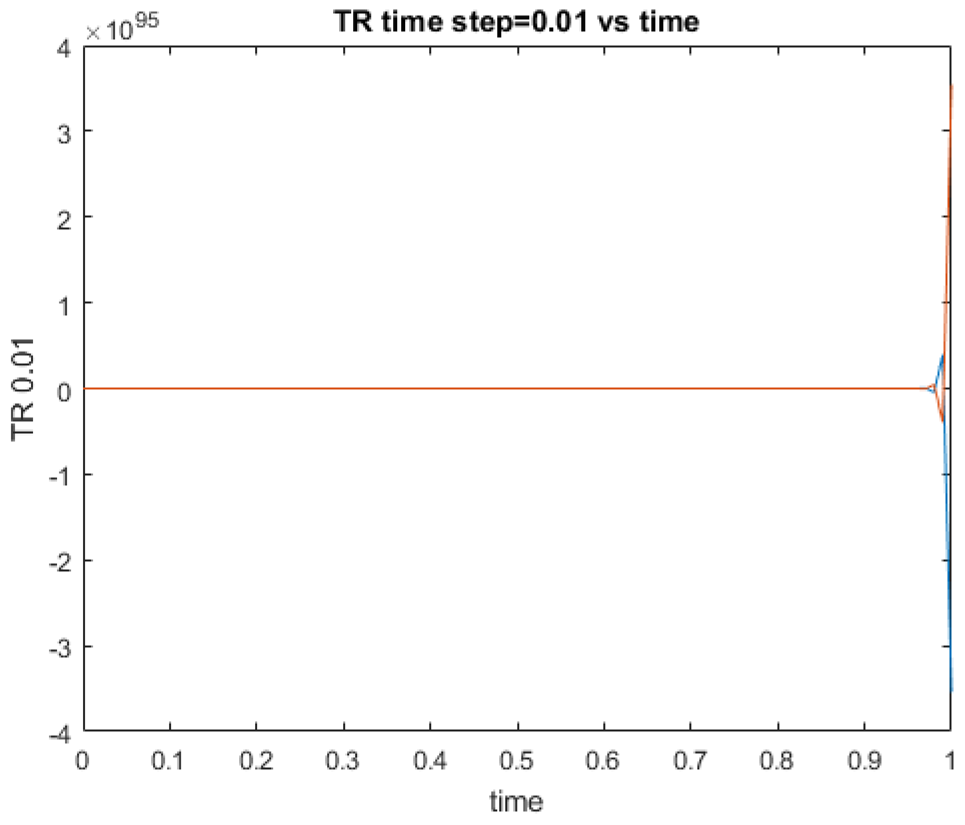

HA time step=0.001 [0,0.1] vs time
HA 0.001
time
4
3
2
1
0
-1
-2
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1

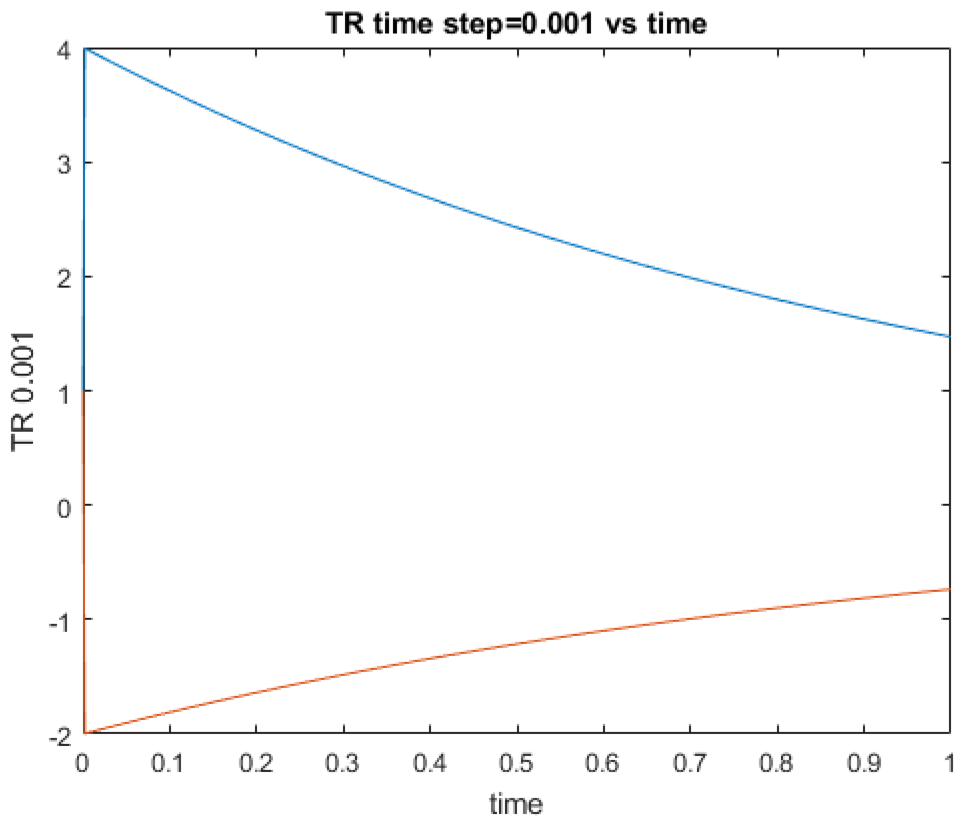


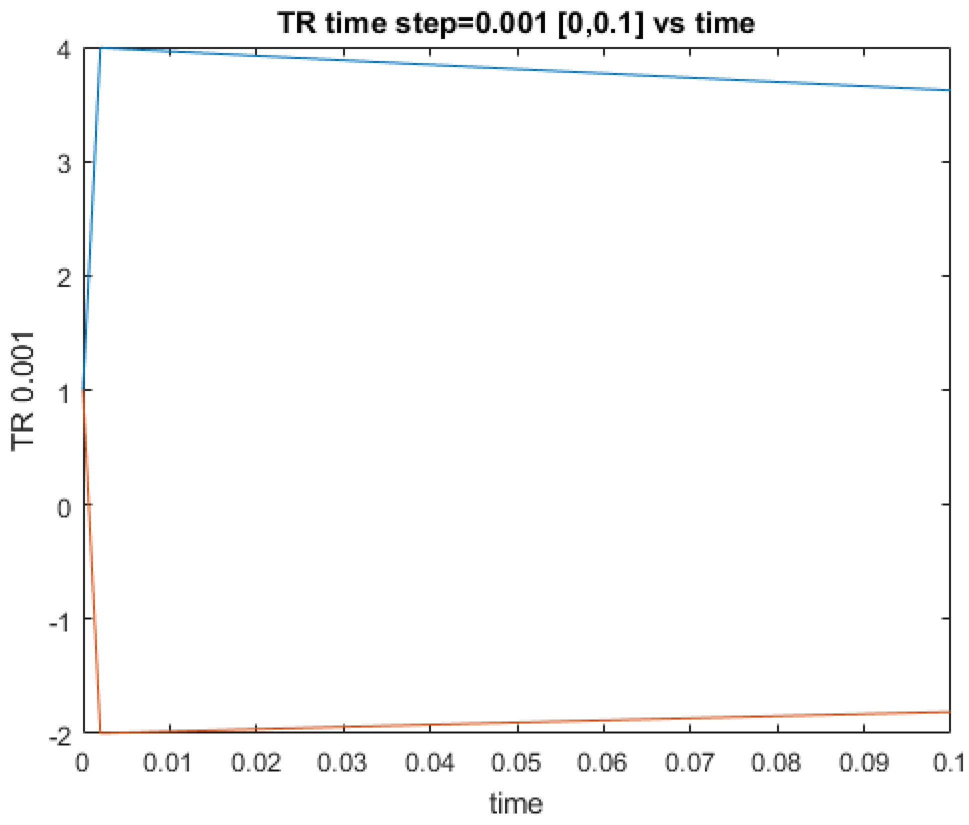
Question3 Trapezoidal Rule

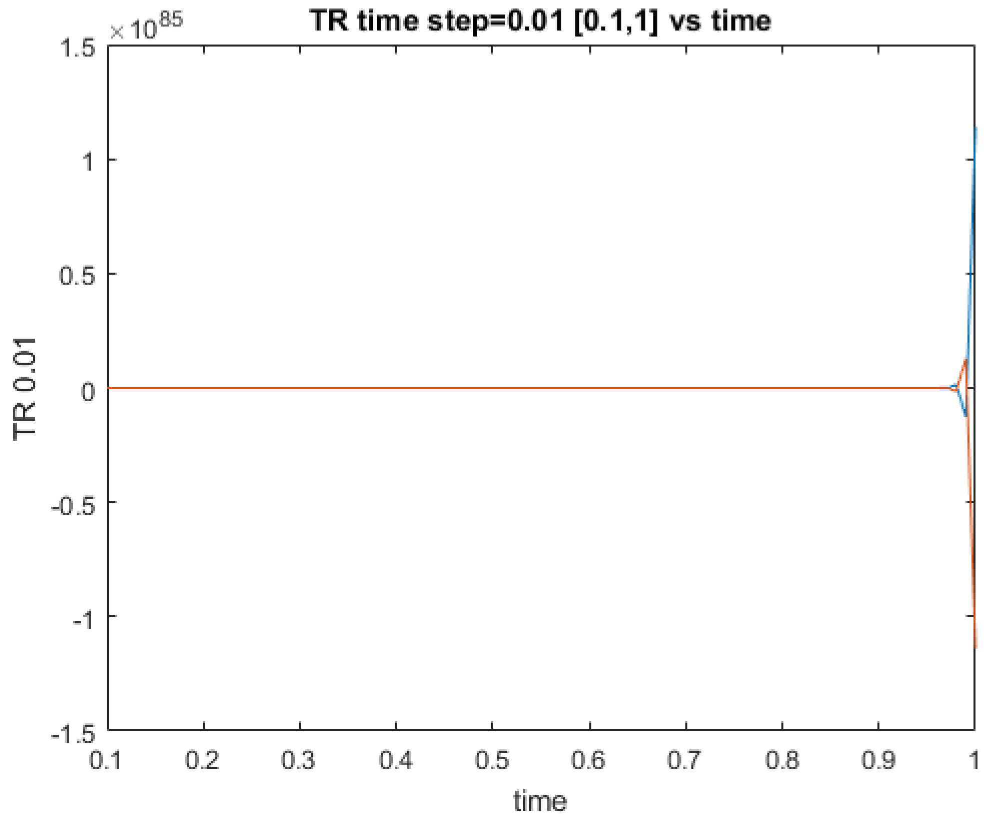
```
[TR_a_tspan,TR_a_xc,TR_a_exitflag]=TR(M,xo,0.01,0,1);
[TR_b_tspan,TR_b_xc,TR_b_exitflag]=TR(M,xo,0.001,0,1);
[TR_c1_tspan,TR_c1_xc,TR_c1_exitflag]=TR(M,xo,0.001,0,0.1);
[TR_c2_tspan,TR_c2_xc,TR_c2_exitflag]=TR(M,TR_c1_xc(:,size(TR_c1_tspan,2)),0.01,0.
figure(10)
plot(TR_a_tspan,TR_a_xc);
title('TR time step=0.01 vs time');
xlabel('time');
ylabel('TR 0.01')
figure(11)
plot(TR_b_tspan,TR_b_xc);
title('TR time step=0.001 vs time');
xlabel('time');
ylabel('TR 0.001')
figure(12)
plot(TR_c1_tspan,TR_c1_xc);
title('TR time step=0.001 [0,0.1] vs time');
xlabel('time');
ylabel('TR 0.001')
figure(13)
plot(TR_c2_tspan,TR_c2_xc);
title('TR time step=0.01 [0.1,1] vs time');
xlabel('time');
```

```
ylabel('TR 0.01')
```




TR time step=0.001 vs time
TR 0.001
time
4
3
2
1
0
-1
-2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


TR time step=0.001 [0,0.1] vs time
TR 0.001
time
4
3
2
1
0
-1
-2
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1



Question4 What did you learn from the above?

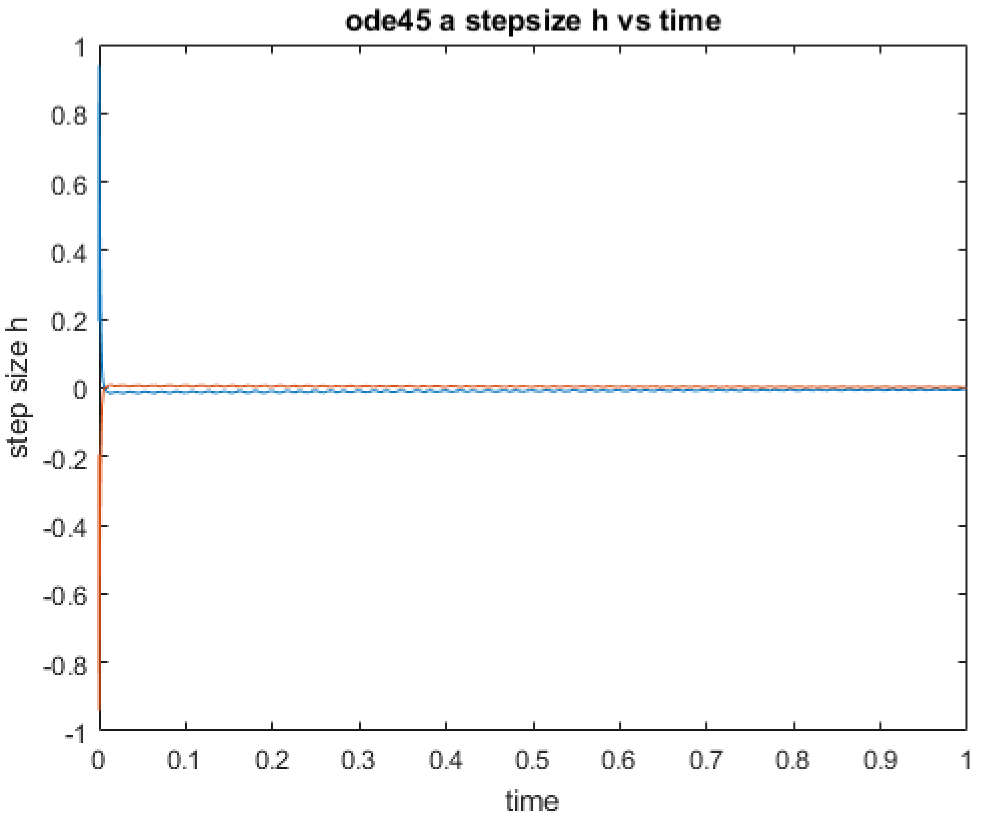
I learned that for stiff equations like these that making the time step small enough to not bigger than the 2 time constants is essential to make the results stable. Even after using the small enough time step to run for certain amount of time does not guarrantee using a bigger time step would still make the results stable. Compareing the results of the FE and Heun and TR, I find that with the same time step 0.001, TR's result are the closest to analytic results, Heun second, FE worst.

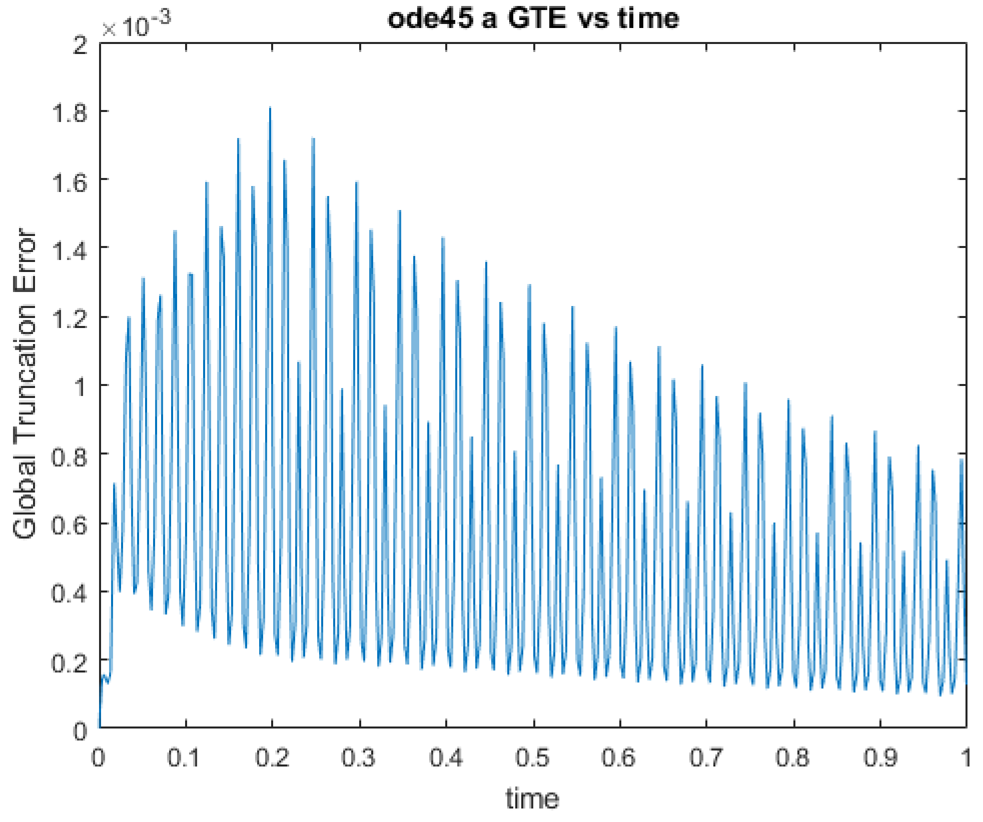
Question5 Question6 ode45 ode15s

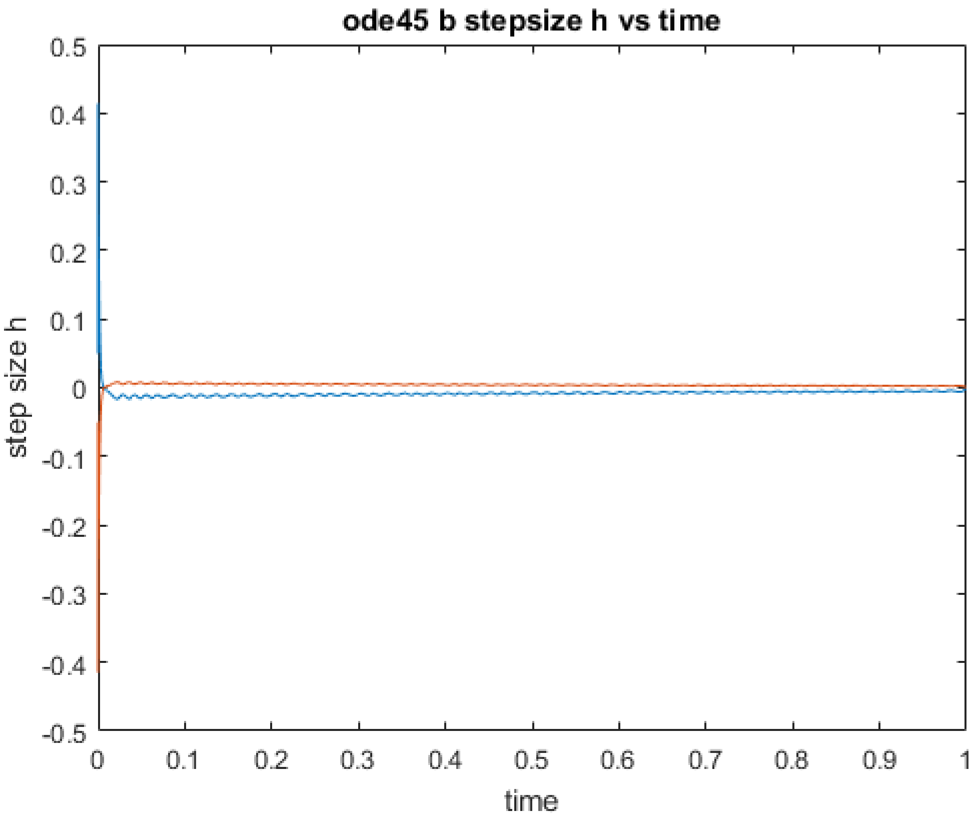
```
%-----~ ode45 a plot step size h vs time ~------
myode=@(t,x) M*x;% my ode function
tspan = [0 1];%initial condition
opts_a = odeset('Refine',1,'RelTol',1e-3,'AbsTol',1e-6);
[ode45_a_t,ode45_a_x] = ode45(myode, tspan, xo, opts_a);
ode45_a_stepsize=diff(ode45_a_x);
ode45_a_t(size(ode45_a_t,1))=[];%truncate the last bit
figure(14)
plot(ode45_a_t,ode45_a_stepsize);
title('ode45 a stepsize h vs time');
xlabel('time');
ylabel('step size h')

%-----~ ode45 a plot global truncation error vs time ~------
% ode45_a_x
```

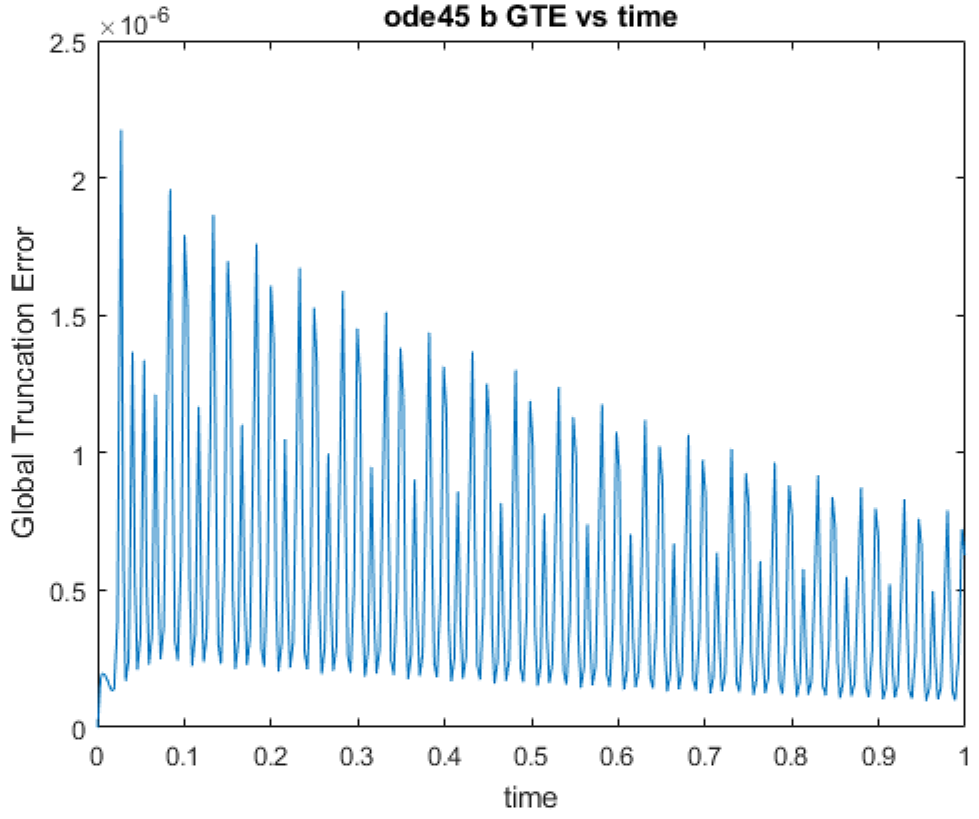
```
ode45_a_GTE=zeros(2,size(ode45_a_t,1));
for k=1:size(ode45_a_t,1)
    ode45_a_GTE(:,k)=f(ode45_a_t(k))-ode45_a_x(k);
end
ode45_a_GTE=ode45_a_GTE.';
k=1:size(ode45_a_t,1);
figure(15)
plot(ode45_a_t(k),ode45_a_GTE(k));
title('ode45 a GTE vs time');
xlabel('time');
ylabel('Global Truncation Error')

%-----~ ode45 b plot step size h vs time ~------
opts_b = odeset('Refine',1,'RelTol',1e-6,'AbsTol',1e-9);
[ode45_b_t,ode45_b_x] = ode45(myode, tspan, xo, opts_b);
ode45_b_stepsize=diff(ode45_b_x);
ode45_b_t(size(ode45_b_t,1))=[];%truncate the last bit
figure(16)
plot(ode45_b_t,ode45_b_stepsize);
title('ode45 b stepsize h vs time');
xlabel('time');
ylabel('step size h')

%-----~ ode45 b plot global truncation error vs time ~------
ode45_b_GTE=zeros(2,size(ode45_b_t,1));
for k=1:size(ode45_b_t,1)
    ode45_b_GTE(:,k)=f(ode45_b_t(k))-ode45_b_x(k);
end
ode45_b_GTE=ode45_b_GTE.';
k=1:size(ode45_b_t,1);
figure(17)
plot(ode45_b_t(k),ode45_b_GTE(k));
title('ode45 b GTE vs time');
xlabel('time');
ylabel('Global Truncation Error')

%-----~ ode15s c plot step size h vs time ~------
[ode15s_c_t,ode15s_c_x] = ode15s(myode, tspan, xo, opts_a);
ode15s_c_stepsize=diff(ode15s_c_x);
ode15s_c_t(size(ode15s_c_t,1))=[];%truncate the last bit
figure(18)
plot(ode15s_c_t,ode15s_c_stepsize);
title('ode15s c stepsize h vs time');
xlabel('time');
ylabel('step size h')

%-----~ ode15s c plot global truncation error vs time ~------
ode15s_c_GTE=zeros(2,size(ode15s_c_t,1));
for k=1:size(ode15s_c_t,1)
    ode15s_c_GTE(:,k)=f(ode15s_c_t(k))-ode15s_c_x(k);
end
ode15s_c_GTE=ode15s_c_GTE.';
k=1:size(ode15s_c_t,1);
figure(19)
```

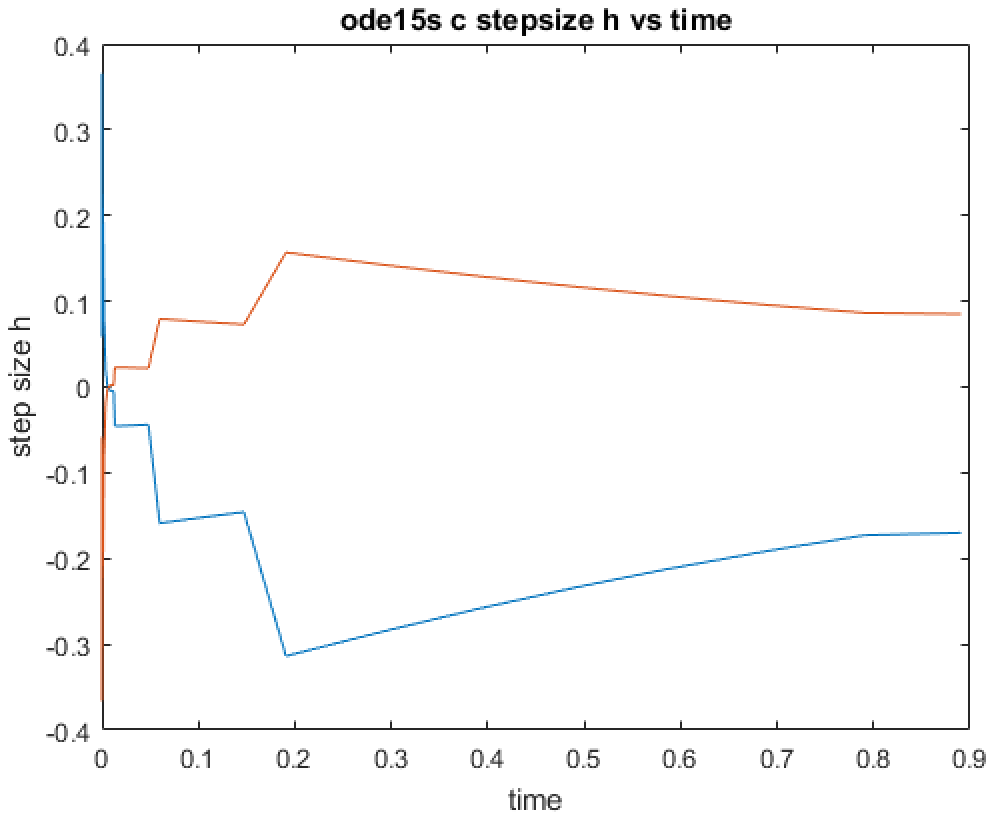
```
plot(ode15s_c_t(k),ode15s_c_GTE(k));
title('ode15s c GTE vs time');
xlabel('time');
ylabel('Global Truncation Error')

%-----~ ode15s d plot step size h vs time ~------
[ode15s_d_t,ode15s_d_x] = ode15s(myode, tspan, xo, opts_b);
ode15s_d_stepsize=diff(ode15s_d_x);
ode15s_d_t(size(ode15s_d_t,1))=[];%truncate the last bit
figure(20)
plot(ode15s_d_t,ode15s_d_stepsize);
title('ode15s d stepsize h vs time');
xlabel('time');
ylabel('step size h')

%-----~ ode15s d plot global truncation error vs time ~------
ode15s_d_GTE=zeros(2,size(ode15s_d_t,1));
for k=1:size(ode15s_d_t,1)
    ode15s_d_GTE(:,k)=f(ode15s_d_t(k))-ode15s_d_x(k);
end
ode15s_d_GTE=ode15s_d_GTE.';
k=1:size(ode15s_d_t,1);
figure(21)
plot(ode15s_d_t(k),ode15s_d_GTE(k));
title('ode15s d GTE vs time');
xlabel('time');
ylabel('Global Truncation Error')
```

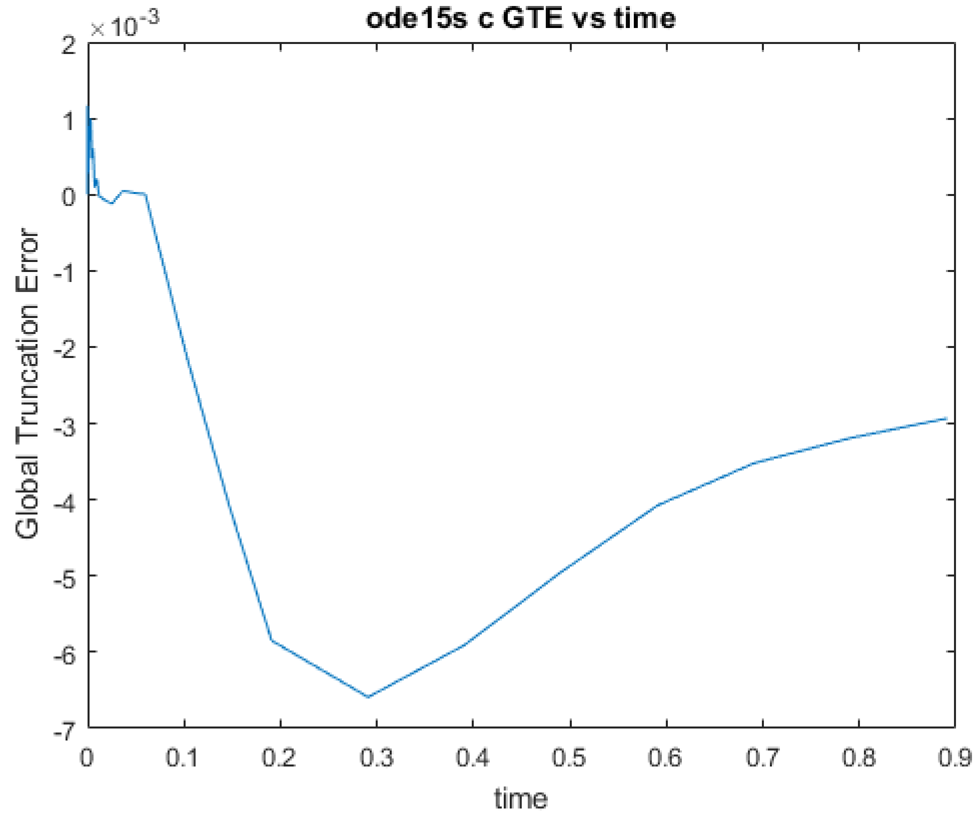

ode45 a stepsize h vs time
step size h
time

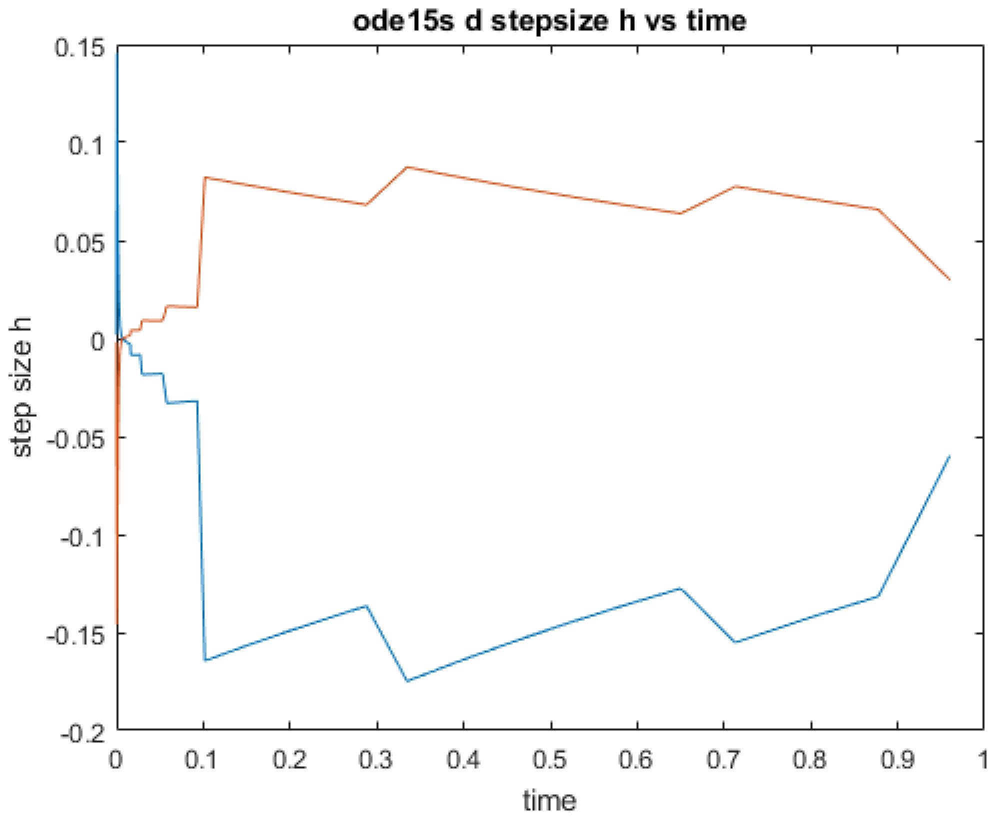

ode45 a GTE vs time
×10^-3
Global Truncation Error
time

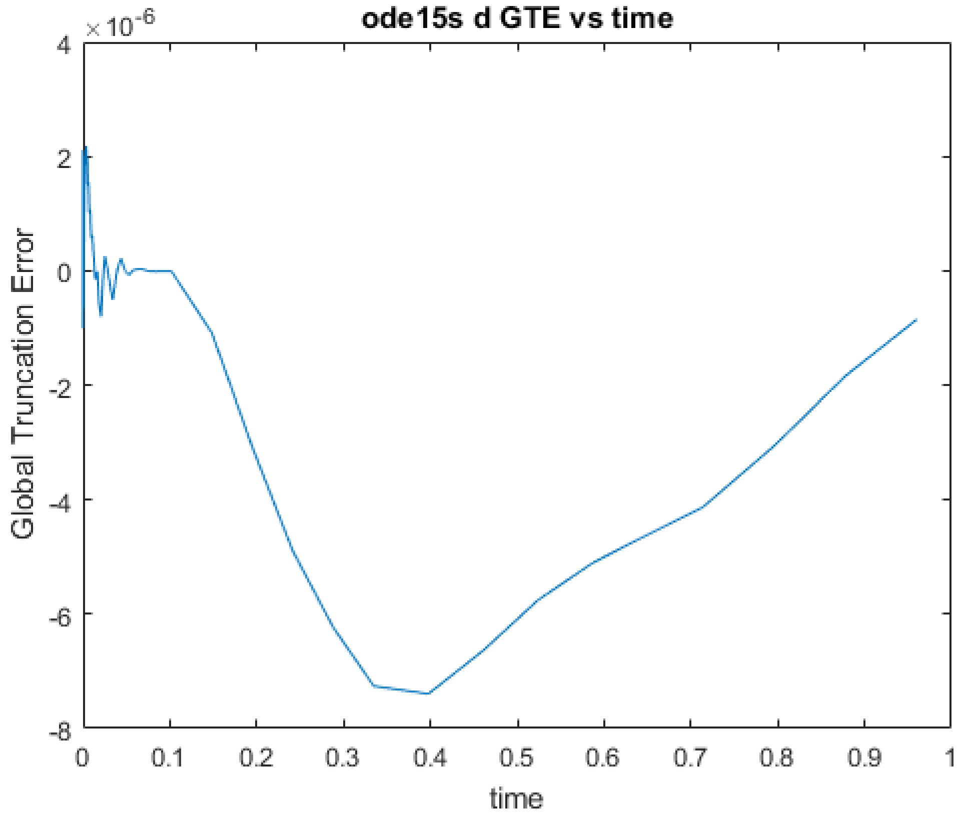

ode45 b stepsize h vs time
step size h
time
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


ode45 b GTE vs time
$\times 10^{-6}$
Global Truncation Error
time
2.5
2
1.5
1
0.5
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


ode15s c stepsize h vs time
step size h
time
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9


$\times 10^{-3}$
ode15s c GTE vs time
Global Truncation Error
time
2
1
0
-1
-2
-3
-4
-5
-6
-7
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9


ode15s d stepsize h vs time
step size h
time
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


ode15s d GTE vs time
$\times 10^{-6}$
Global Truncation Error
time
4
2
0
-2
-4
-6
-8
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Question7

Compare the results of using the non-stiff method ode45 to the stiff method ode15s. The step size of the non-stiff method is at a very small scale for almost the whole period. The step size of the stiff method increases greatly as the function gets more flat. The global truncation error is bouncing when choosing the non-stiff method, and it is more smooth when using the stiff method.

Compare the loose tolerance results to the tight tolerance results. When choosing the loose tolerance, the scale of the GTE is 1e-3, and the tight tolerance GTE is 1e-6.

Question8

What did you learn from the above? Using a stiff method for a stiff equation is more suitable because it would save a lot of time step and making the GTE smoother. Choosing a tighter tolerance would be resulting closer results.

Published with MATLAB® R2016a