UNIVERSITY OF SHEFFIELD

Foundation of Control Systems

Matlab Programming

AWABULLAH SYED

12/4/2020

Name: Awabullah M. Syed

Student Number: 2002 000 33

Module Code: ACS 6101

Module Assessor: Dr Viktor Fedun

Awab Syed

Question 1

```
%Author: Awabullah M Syed
%Student ID: 2002 000 33
%Date Completed: 10 November 2020
```

Question 1 - Matrix Selection

Code

```
%***************************Question 1*****************************************************%
%Code Description / Purpose:
    %This code allows the user to select either matrix M1 or
    %M2 and then perform relevant operation. Relevant operation includes
    %finding eigenvalues (D) and eigenvectors(V &W) of the selected matrix
    %if the determinant is nonzero, otherwise displays that the determinant
    %is 0. The code also guards against poor user input including the use
    %of invalied notation.
clear
clc
M1 = [-2 -4.1 2 3; -2 1.1 2 3.3;4.1 2 5 8.1; 1 3 2.2 1.5]; %Adds matrix M1
M2 = [1 3 9 2; 5 8 4 3; 10 16 8 6; 2 3 1 8]; %Adds matrix M2
y = [-1 2 -2 4]'; %Adds vector y
disp('Matrix M1 is'); disp(M1) %User will be displayed matrix M1
disp('Matrix M2 is'); disp(M2) %User will be displayed matrix M2
%-------------------Ask user to select M1 or M2--------------------------------------------%
A = input('Please select either matrix "M1" or "M2"');
if A == M1 | A == M2 %guards against any poor input integer i.e. 3 or i
    if det(A) ~= 0 %Determinant of A is nonzero (not zero)
%Eigenvector
        [V,D,W] = eig(A); %V = right eigenvectors & W is left eigenvectors
        disp('The right eigenvector is'); disp(V) %A*V = V*D
        disp('The left eigenvector is'); disp(W)  %W*A = D*W
%Eigenvalues
        a = diag(D); %Taking the diagonal of D - Eigenvalues
        disp('Eigenvalues of the selected matrix are');disp(a) %Eigenvalues
%Solution x to the equation Ax = y
        x = A\y;
        disp('The solution x to the equation Ax = y is'); disp(x) %x matrix
    else %if the determinant of the selected matrix is 0
        disp('The Determinant of the selected matrix is zero')%Informs user
    end
else  %Guards against poor input
    disp('Please select a valid matrices or use correct notation "M1" or "M2"')
end
```

Output

User Input = M1 (Matrix)

```
Matrix M1 is
   -2.0000   -4.1000    2.0000    3.0000
   -2.0000    1.1000    2.0000    3.3000
    4.1000    2.0000    5.0000    8.1000
    1.0000    3.0000    2.2000    1.5000
Matrix M2 is
     1     3     9     2
     5     8     4     3
    10    16     8     6
     2     3     1     8
The right eigenvector is
    0.1327   -0.5711   -0.8371    0.1333
    0.3097    0.6342   -0.4381    0.0020
    0.8691   -0.4541    0.2289   -0.8104
    0.3621    0.2558    0.2343    0.5705
The left eigenvector is
    0.2105   -0.5149   -0.6547    0.3066
    0.2900    0.8320   -0.6288   -0.2855
    0.5608   -0.1676    0.1630   -0.3019
    0.7464   -0.1206    0.3867    0.8564
Eigenvalues of the selected matrix are
    9.7134
    2.7999
   -5.5320
   -1.3812
The solution x to the equation Ax = y is
   -0.7381
    0.6578
    2.2121
   -1.4012
```

User Input = M2 (Matrix)

```
Matrix M1 is
   -2.0000   -4.1000    2.0000    3.0000
   -2.0000    1.1000    2.0000    3.3000
    4.1000    2.0000    5.0000    8.1000
    1.0000    3.0000    2.2000    1.5000
Matrix M2 is
     1     3     9     2
     5     8     4     3
    10    16     8     6
     2     3     1     8
The Determinant of the selected matrix is zero
```

Poor User Input = ‘6’

Selecting user input of 6 as an example

```
Please select a valid matrix or use correct notation "M1" or "M2"
```

Question 2

Question 2 - HIRES Reaction Problem

Code

```
%********************************Question 2************************************************%
%Code Description / Purpose:
    %The following code models the HIRES reaction problem using ODE45
    %and ODE15s. The code outputs two Figures; one & two. Figure 1 compares
    %the result of ODE45 with ODE15s while Figure 2 plots first hundred
    %points of ODE45 and ODE15
%------------------------------HIRES Function----------------------------------------------%
function xdot = HIRES(t,x)
xdot=zeros(8,1);
xdot = [-1.71*x(1) + 0.43*x(2) + 8.32*x(3) + 0.007;
    1.71*x(1) - 8.75*x(2);
    -10.03*x(3) + 0.43*x(4) + 0.035*x(5);
    8.32*x(2) + 1.71*x(3) - 1.12*x(4);
    -1.745*x(5) + 0.43*x(6) + 0.43*x(7);
    -280*x(6)*x(8) + 0.69*x(4) + 1.71*x(5) - 0.43*x(6) + 0.69*x(7);
    280*x(6)*x(8) - 1.81*x(7);
    -280*x(6)*x(8) + 1.81*x(7);];
end
clear
clc
%-----------------------------Initial Conditions-------------------------------------------%
x(1)=1; x(2)=0; x(3)=0; x(4)=0; x(5)=0; x(6)=0; x(7)=0; x(8)=0.0057;
ic = [x(1) x(2) x(3) x(4) x(5) x(6) x(7) x(8)];
t0 = 0; %Initial Time
tf = 300; %Final Time
opts = odeset('RelTol', 1e-3,'AbsTol',1e-6); %Options argument
%-----------------------------ODE45-------------------------------------------------------%
% Solution using ODE45 Function with time interval: 0 - 300 seconds
tic %Tic toc starting point for ode45
%x_45 & t_45 is used to differentiate between ode45 & ode15
[t_45,x_45] = ode45(@HIRES,[t0 tf],ic,opts);
execution_time45 = toc; %Execution time of  ODE45
disp(['The execution time for ODE45 is ',num2str(execution_time45)])
% ------------------------ODE15s-----------------------------------------------------------%
tic %Tic toc starting point for ode15s
[t_15,x_15] = ode15s(@HIRES,[t0 tf],ic,opts);
execution_time15 = toc; %Execution time of ODE15s
disp(['The execution time for ODE15s is ', num2str(execution_time15)])
```

```
%--------------------Plotting ODE45 & ODE15s (Figure 1)---------------------------------%
figure(1);
set(gcf,'Units','Normalized','OuterPosition',[0 0 0.5 1]); %Figure positioning
set(gcf,'Toolbar','none','Menu','none'); %Making Figure 1 fullscreen
% Plotting ODE45 in first half of Figure 1
subplot(2,1,1);
semilogx(t_45,x_45) %Logarithmic x-axis of ODE45
title({('ODE_4_5')...
    ['\fontsize{9} {\color{blue}Execution Speed: '...
    num2str(execution_time45),', RelTol: 1e-3,'...
    ',AbsTol: 1e-6}']
    })
xlabel('time $(seconds)$','interpreter','latex','FontSize',10);
ylabel('Concentration $(mol/dm^3)$','interpreter','latex','FontSize',10)
legend('x1','x2','x3','x4','x5','x6','x7','x8','Location','northeast')
% Plotting ODE15s in second half of Figure 1
subplot(2,1,2);
semilogx(t_15,x_15) %Logarithmic x-axis of ODE15s
title({('ODE_1_5')...
    ['\fontsize{9} {\color{blue}Execution Speed:'...
    num2str(execution_time15),', RelTol: 1e-3'...
    ', AbsTol: 1e-6}']
    })
xlabel('time $(seconds)$','interpreter','latex','FontSize',10);
ylabel('Concentration $(mol/dm^3)$','interpreter','latex','FontSize',10)
legend('x1','x2','x3','x4','x5','x6','x7','x8','Location','northeast')

%---------------------------------Second Figure---------------------------------------- %
figure(2);
k=1;
while k<9
    subplot(8,1,k);
    plot(t_45(1:100,:),x_45(1:100,k),'k.'); hold on
    plot(t_15(1:100,:),x_15(1:100,k),'g');
    xlabel('time $(seconds)$','interpreter','latex','FontSize',5);
    ylabel('Conc $(mol/dm^3)$','interpreter','latex','FontSize',5)
    legend('ODE_4_5','ODE_1_5_s','Location','northeast')
    title ({['First 100 Points of x' num2str(k)]})
    k = k + 1;
end
set(gcf,'Units','Normalized','OuterPosition',[0.5 0 0.5 1]);
set(gcf,'Toolbar','none','Menu','none');
```

Output

Execution Time of ODE45 and ODE15s (Tic Toc)

```
The execution time for ODE45 is 1.032
The execution time for ODE15s is 0.02049
```

Figure 1 – Using HIRES function to plot ODE45 and ODE15s over a time interval of 0 to 300s

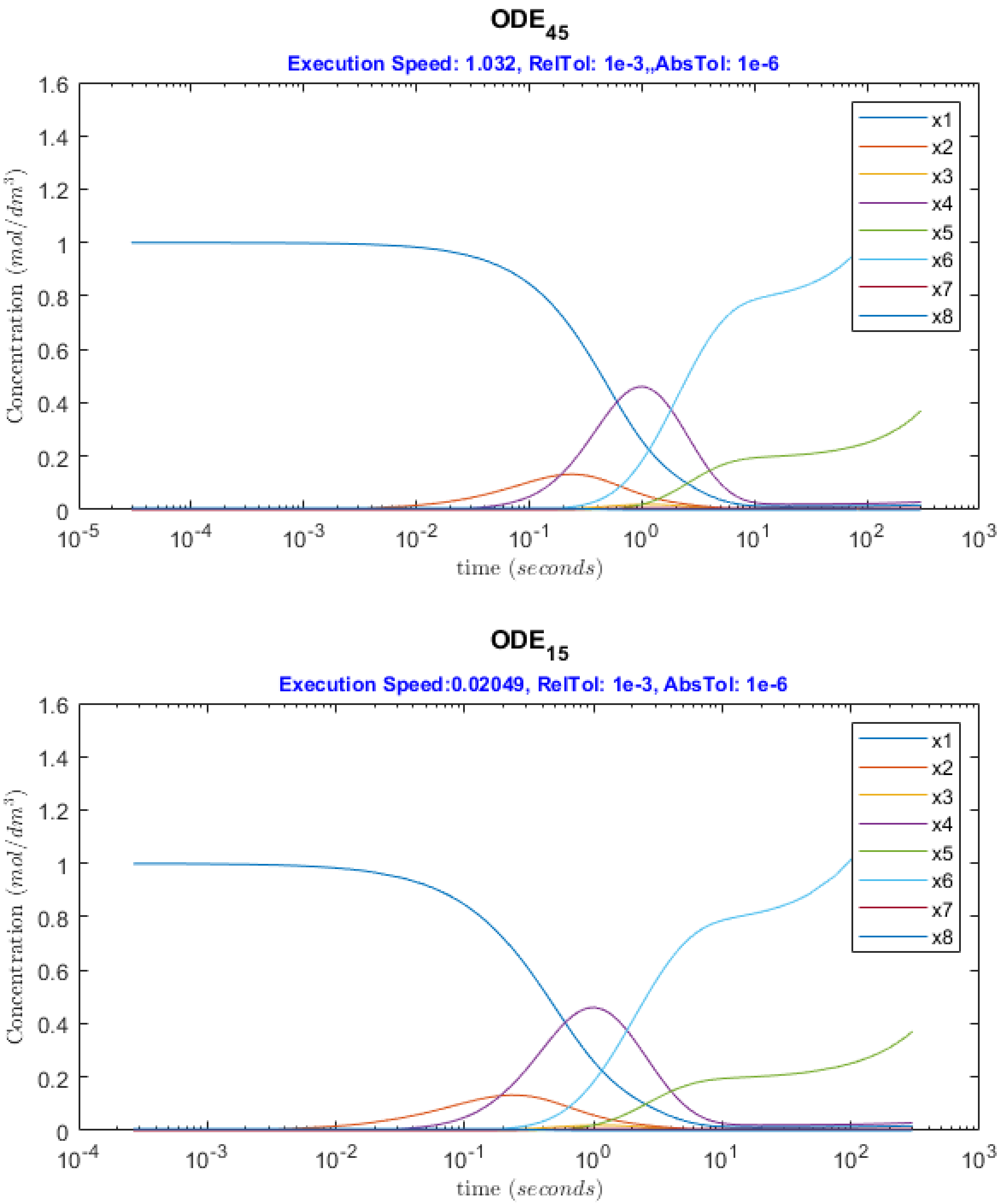
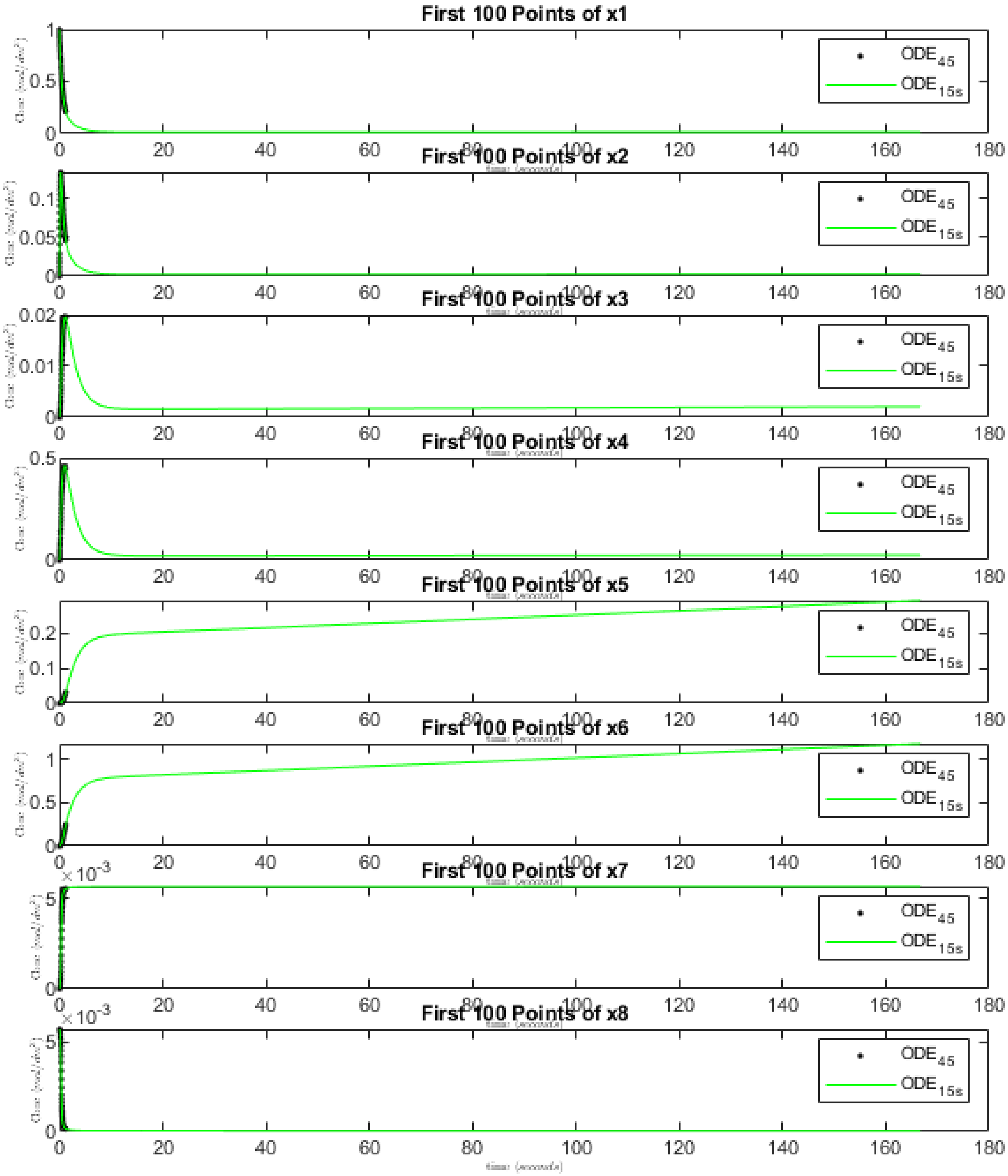


Figure 2 - Eight subplot of first 100 points of data from ODE45 and ODE15s


First 100 Points of x1
First 100 Points of x2
First 100 Points of x3
First 100 Points of x4
First 100 Points of x5
First 100 Points of x6
First 100 Points of x7
First 100 Points of x8
ODE$_{45}$
ODE$_{15s}$
Conc (mol/m^3)
time (seconds)
$\times 10^{-3}$