# MA1506

# Lab Assignment

**declaration**: I certify that the work submitted here represents solely my own efforts. I am aware of the University's regulations about, and penalties for, plagiarism.

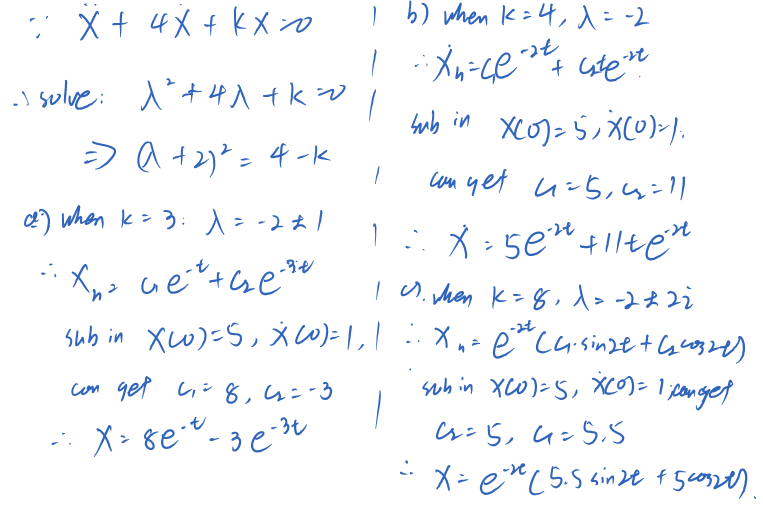
Name: **Song Yangyu**

Matric No.: **A0077863N**

Lecture Group: **C**

Signature:

Question 1)



1. SciLab Commends:

function res = f1(t)

res = 8\*exp(-t) - 3\*exp(-3\*t)

endfunction

function res = f2(t)

res = 5\*exp(-2\*t) + 11\*t.\*exp(-2\*t)

endfunction

function res = f3(t)

res = exp(-2\*t).\*(5.5\*sin(2\*t)+5\*cos(2\*t))

endfunction

t = 0:0.08:8;

x1 = f1(t);

x2 = f2(t);

x3 = f3(t);

plot(t,x1,'r-');

plot(t,x2,'g.');

plot(t,x3,'b-.');

xlabel('t');

ylabel('x');

title('Song Yangyu on Q1');

// another method

clear;

clc;

function res = f1(t,x)

res = [x(2);-4\*x(2)-3\*x(1)]

endfunction;

function res = f2(t,x)

res = [x(2);-4\*x(2)-4\*x(1)]

endfunction;

function res = f3(t,x)

res = [x(2);-4\*x(2)-8\*x(1)]

endfunction;

t = 0:0.08:8;

x1 = ode([5;1],0,t,f1);

x2 = ode([5;1],0,t,f2);

x3 = ode([5;1],0,t,f3);

plot(t,x1(1,:),'r-');

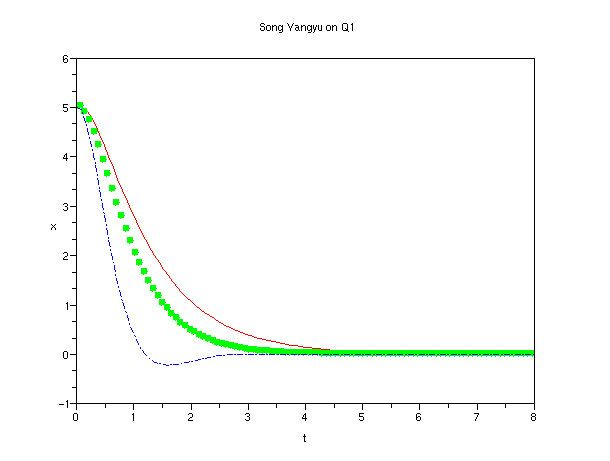
plot(t,x2(1,:),'g.');

plot(t,x3(1,:),'b-.');

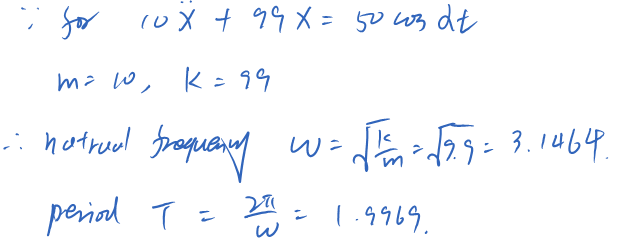
red line (over damping):

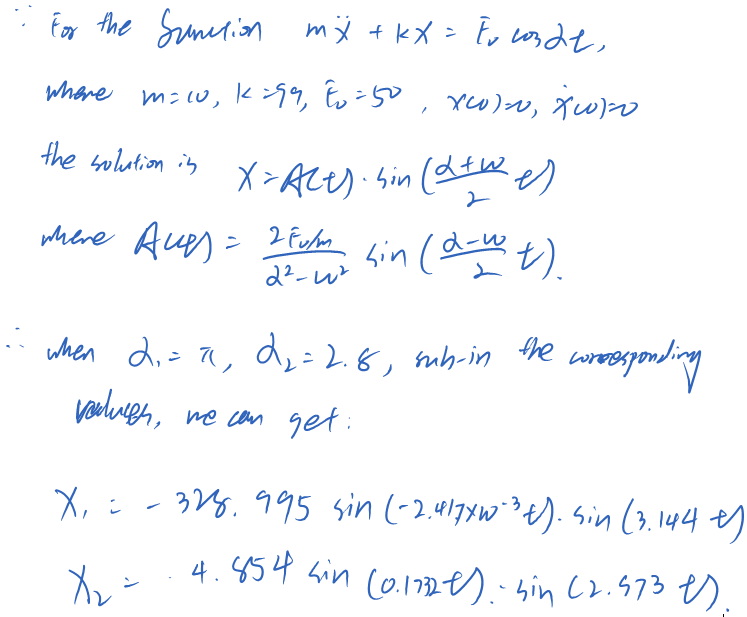
green dotted line (critical damping):

blue dashed line (under damping):



Question 2)





SciLab Script:

function res = f(a,b,c,t)

res = a .\* sin(b\*t) .\* sin(c\*t)

endfunction

T = 1.9969

t = linspace(0,10\*T,1000);

x1 = f(-328.995,-2.417e-3,3.144,t);

x2 = f(4.854,0.1732,2.973,t);

plot(t,x1,'r--');

plot(t,x2,'g-.');

xlabel('t');

ylabel('x');

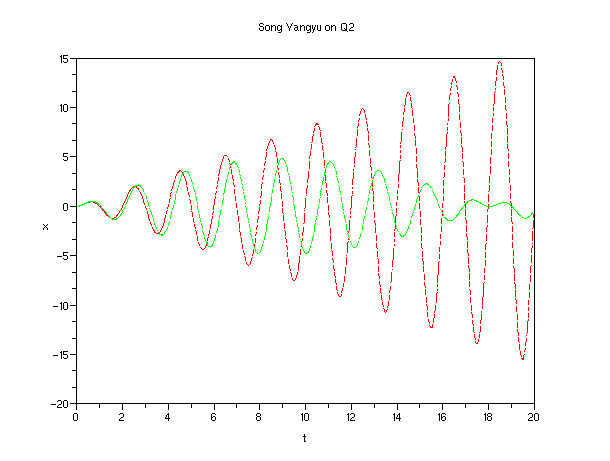
title('Song Yangyu on Q2');

Info on the Graph:

x1: red dashed line;

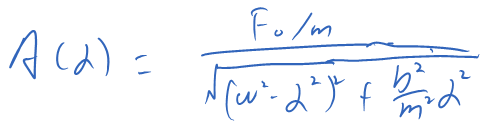
x2: green line

1. x1 (α1 = π) corresponds to resonance like behavior.



Question 3:

1. For the Forced Oscillator with damping, the amplitude response is:



where in this problem, F0 = 50, m = 10, b = 5, k = 120, ω = sqrt(b/k). Sub in these values, and this is the Sci Script:

F0 = 50;

m = 10;

b = 5;

k = 120;

omega = sqrt(k/m);

alpha = linspace(0,30,1000);

function res = f(F0,m,omega,b,alpha)

res = (F0/m)./sqrt((omega^2.-alpha^2)^2 + b^2/m^2.\*alpha^2)

endfunction;

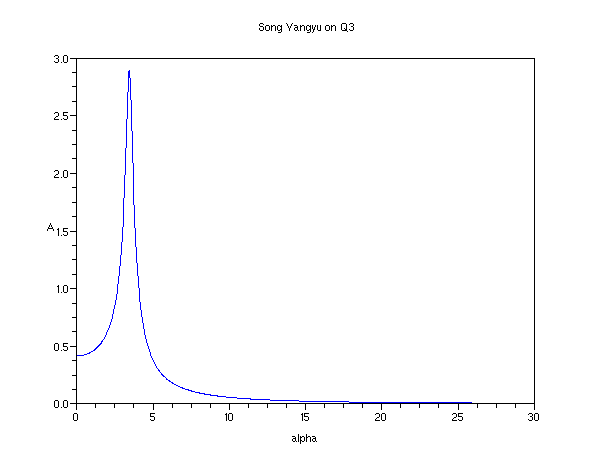
A = f(F0,m,omega,b,alpha);

plot(alpha,A);

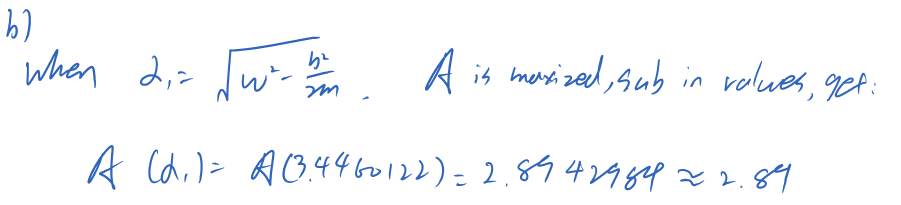
xlabel('alpha');

ylabel('A');

title('Song Yangyu on Q3');



3 b)



// additional for calculating the max of A

alpha\_1 = sqrt(omega^2 -b^2/(2\*m));

A\_max = f(F0,m,omega,b,alpha\_1);

// use previous result to check:

[max\_value index] = max(A);

// output:

// index =

// 116.

// max\_value =

// 2.8930211

// this can get exactly the same value (precise to 2 dicimal place)

Question 4:

Sci Lab Code:

// a)

function res = f(t,x)

res = [1;(x(2)-1)^2 - x(2)]

endfunction

fchamp(f,0,0:0.2:4,0:0.2:4);

// b)

function res = g(t,x)

res = (x-1)^2 - x

endfunction

t = 0:0.2:4;

x1 = ode(1,0,t,g);

x2 = ode(2,0,t,g);

x3 = ode((3+sqrt(5))/2,0,t,g);

plot(t,x1,'r-');

plot(t,x2,'g.');

plot(t,x3,'k--');

x2($) // return 0.3827296, i.e., roughly (3-sqrt(5))/2

// c)

x4 = ode(2.4,0,t,g);

plot(t,x4,'b-.');

xlabel('t');

ylabel('x');

title('Song Yangyu on Q4');

// another way

function res = f3(t)

res = 1.5 -1.118\*tanh(1.118.\*t-1.113)

endfunction;

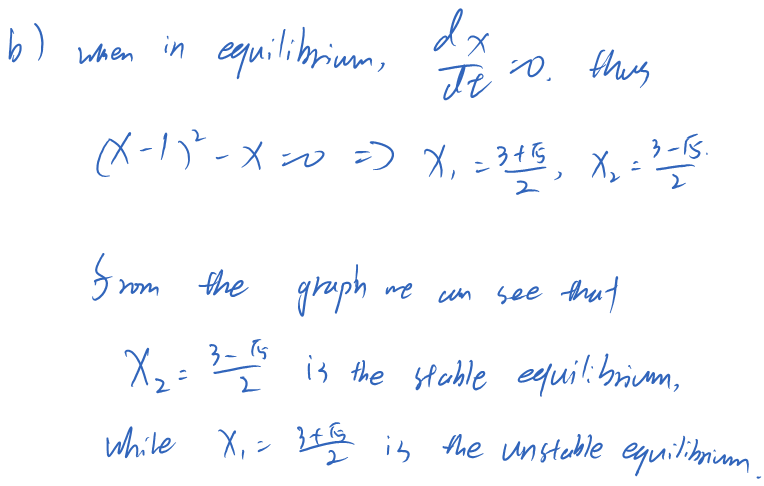
t = 0:0.2:4;

x5 = f3(t);

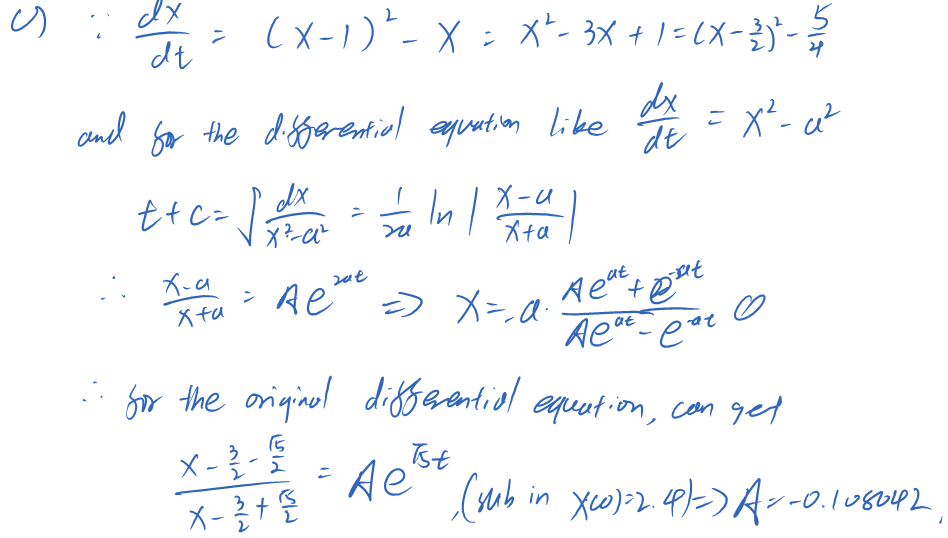
plot(t,x5);

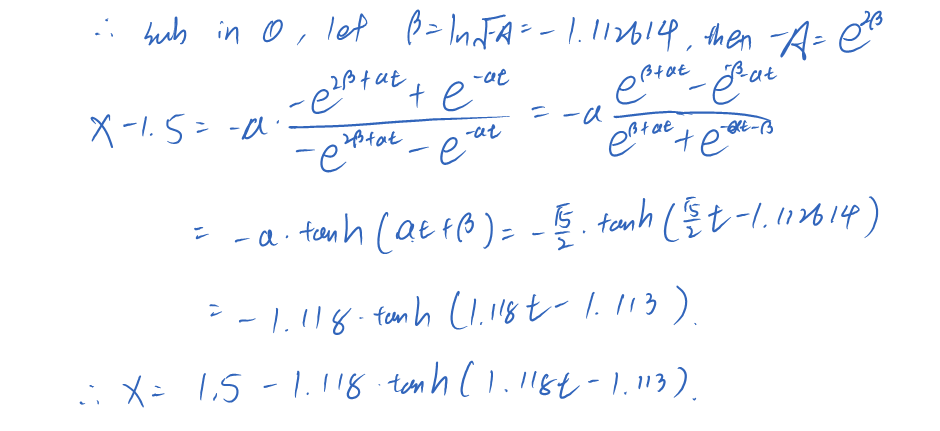
// this would coincide with the ode graph got eailier.

4. b)

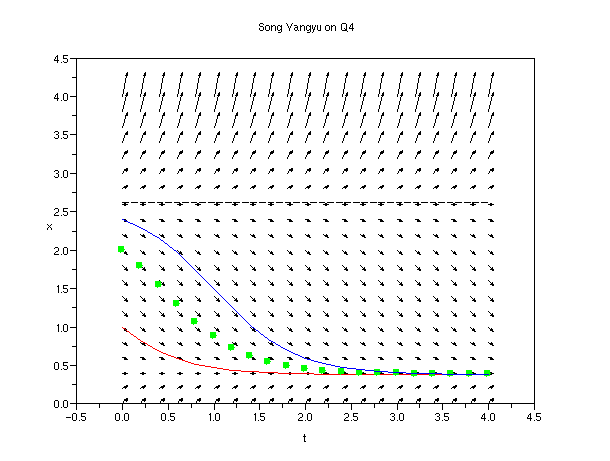


4. c)





The Graph:



Question 5)

b) Output: 0.2370092

c) code:

function res = myfunc(t,x)

res = [x(2);-x(2)-sqrt(x(1))./t]

endfunction;

t = 0.01:0.01:15;

t0 = 0.01;

x0 = [4;10];

x = ode(x0,t0,t,myfunc);

plot(t,x(1,:));

xlabel('t');

ylabel('x');

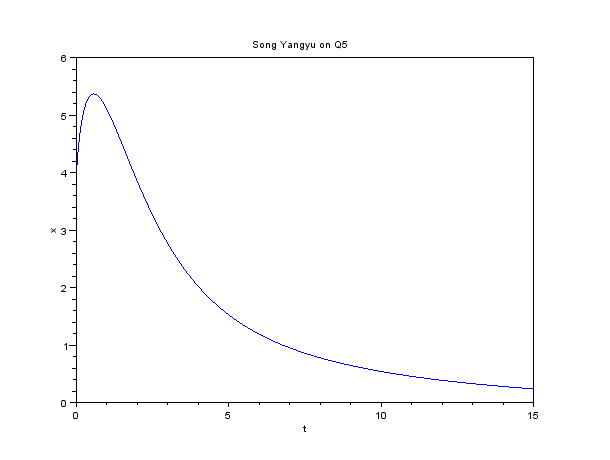
title('Song Yangyu on Q5');

// for part b

x(1,$);

// output: 0.2370092

Graph:



Question 6:

a)

M =

b) Eigen values:

1, 0.4, 0.3, 0.4

c) distribution of the population after 5 years:

A: 38.3%, B: 28.5%, C:16.6%, D: 16.6%

Sci Lab Code Used:

// the probability matrix

M = [0.6,0.3,0.1,0.3;0.2,0.5,0.3,0.1;0.1,0.1,0.5,0.1;0.1,0.1,0.1,0.5]

// the original population

S = [0.7;0.1;0.1;0.1];

// finding the eigenvalues

mtlb\_eig(M)

// distribution of the population

// after 5 years

M^5 \* S