Matlab small project

Cheng Shi

Table of content:

1. Four population growth models

1.1 Exponential growth

Eq: $\begin{cases} \dfrac{dx}{dt} = rx \\ x(0) = x_0 \end{cases}$

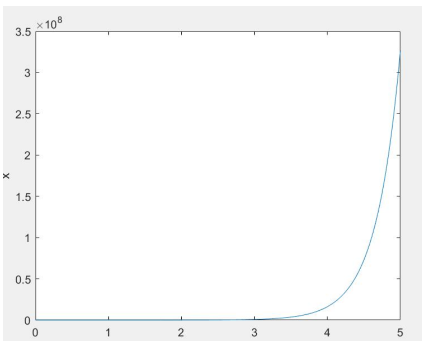
We can easily get $x(t) = x_0 e^{rt}$

Let r= 3 x_0 =100

Code:

```
function Exponential_growth

t=0:0.001:5;    % time scalex
initial_x=100;

[t,x]=ode45( @rhs, t, initial_x);

plot(t,x);
xlabel('t'); ylabel('x');

    function dxdt=rhs(t,x)
        dxdt = 3*x;
    end
end
```

Figure:



1.2 Logistic growth

Eq: $\begin{cases} \dfrac{dx}{dt} = r(1-\dfrac{x}{K})x \\ x(0) = x_0 \end{cases}$

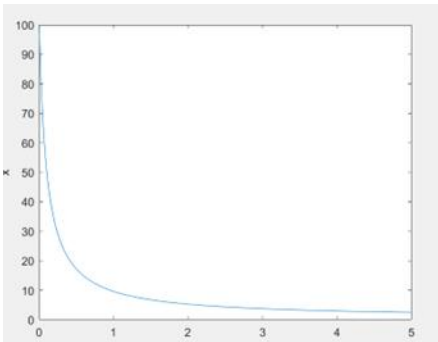
We can get $x(t) = \dfrac{K}{1+(\dfrac{K}{x_0}-1)\bullet e^{-rt}}$

Let r= 0.1 K= 1 x_0 =100

Code:

```
function Logistic_growth
t=0:0.001:5;    % time scalex
initial_x=100;

[t,x]=ode45( @rhs, t, initial_x);

plot(t,x);
xlabel('t'); ylabel('x');

    function dxdt=rhs(t,x)
        dxdt = 0.1*x*(1-x);
    end
end
```

Figure:



1.3 Logistic growth with constant harvesting

Eq: $\begin{cases} \dfrac{dx}{dt} = r(1-\dfrac{x}{K})x\text{-}H \\ x(0) = x_0 \end{cases}$

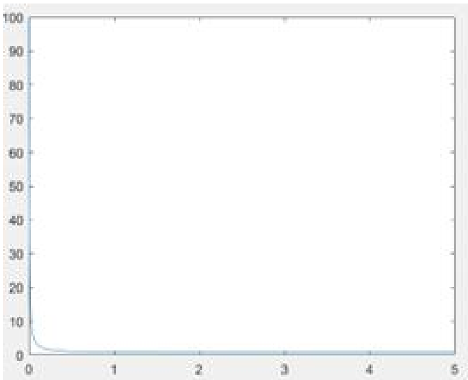
Let x_0 =100,We can get:

(1) r=4 K=1 H=0.5

Code:

```
function LGWCH1

t=0:0.001:5;   % time scalex
initial_x=100;

[t,x]=ode45( @rhs, t, initial_x);

plot(t,x);
xlabel('t'); ylabel('x');

    function dxdt=rhs(t,x)
        dxdt = 4*x*(1-x)-0.5;
    end
end
```

Figure:

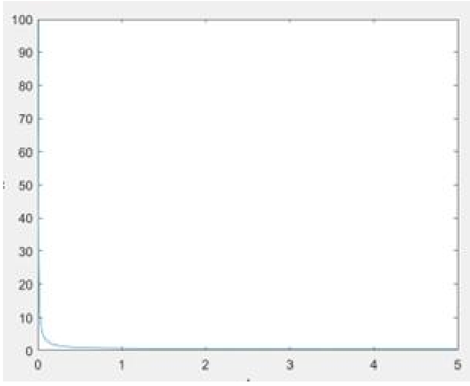


(2) r=4 K=1 H=1

Code:

```
function LGWCH2

t=0:0.001:5;   % time scalex
initial_x=100;

[t,x]=ode45( @rhs, t, initial_x);

plot(t,x);
xlabel('t'); ylabel('x');

    function dxdt=rhs(t,x)
        dxdt = 4*x*(1-x)-1;
    end
end
```

Figure:

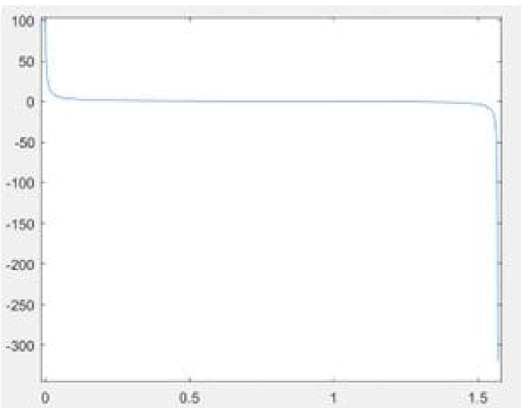


(3) r=4 K=1 H=2

Code:

```
function LGWCH3

t=0:0.001:5;    % time scalex
initial_x=100;

[t,x]=ode45( @rhs, t, initial_x);

plot(t,x);
xlabel('t'); ylabel('x');

    function dxdt=rhs(t,x)
        dxdt = 4*x*(1-x)-2;
    end
end
```

Figure:



1.4 Logistic growth with population-dependent harvesting

Eq: $$\begin{cases} \dfrac{dx}{dt} = r(1-\dfrac{x}{K})x\text{-}H\dfrac{x}{A+x} \\ x(0) = x_0 \end{cases}$$

Let h=H/rK , a=A/K

Then we can get:

Eq: $\begin{cases} \dfrac{dx}{dt} = (1\text{-}x)x\text{-h}\,\dfrac{x}{\text{a}+x} \\ x(0) = x_0 \end{cases}$

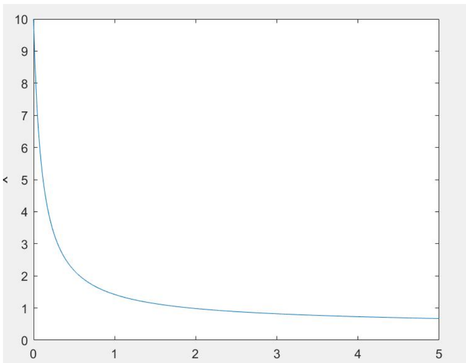
Let x_0 =10

(1) h=1 a=3

Code:

```
function LGWPDH1

t=0:0.001:5;   % time scalex
initial_x=10;

[t,x]=ode45( @rhs, t, initial_x);

plot(t,x);
xlabel('t'); ylabel('x');

    function dxdt=rhs(t,x)
        dxdt =x*(1-x)-1/(3+x);
    end
end
```

Figure:

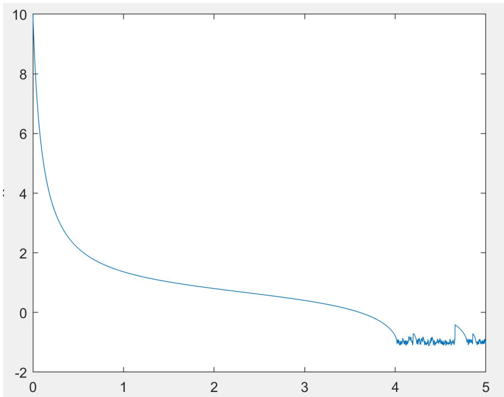


(2) h=1 a=1

Code:

```
function LGWPDH2

t=0:0.001:5;   % time scalex
initial_x=10;

[t,x]=ode45( @rhs, t, initial_x);

plot(t,x);
xlabel('t'); ylabel('x');

    function dxdt=rhs(t,x)
        dxdt =x*(1-x)-1/(1+x);
    end
end
```

Figure:

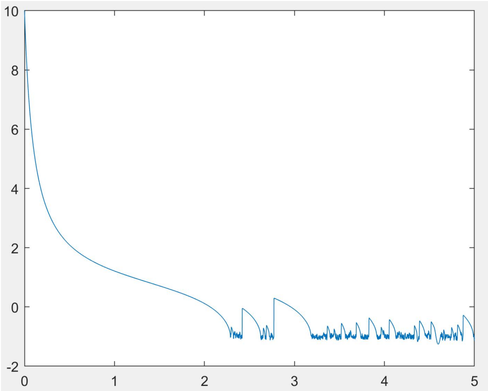


(3) h=2 a=1

Code:

```
function LGWPDH3

t=0:0.001:5;   % time scalex
initial_x=10;

[t,x]=ode45( @rhs, t, initial_x);

plot(t,x);
xlabel('t'); ylabel('x');

    function dxdt=rhs(t,x)
        dxdt =x*(1-x)-2/(1+x);
    end
end
```

Figure:

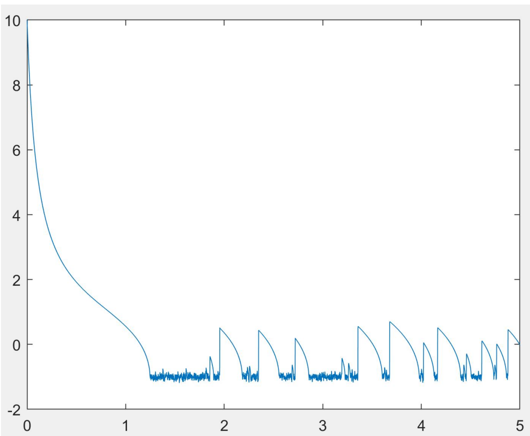


(4) h=5 a=1

Code:

```
function LGWPDH4

t=0:0.001:5;    % time scalex
initial_x=10;

[t,x]=ode45( @rhs, t, initial_x);

plot(t,x);
xlabel('t'); ylabel('x');

    function dxdt=rhs(t,x)
        dxdt =x*(1-x)-5/(1+x);
    end
end
```

Figure:

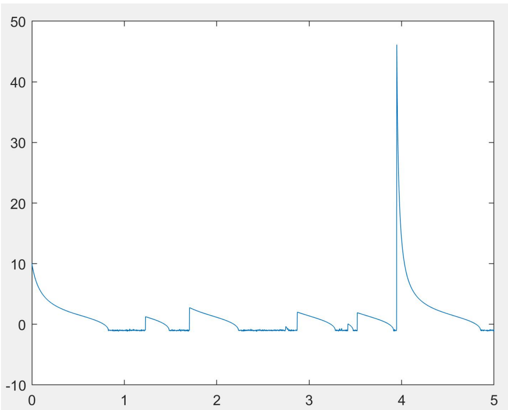


(5)h=10 a=1

Code:

```
function LGWPDH5

t=0:0.001:5;    % time scalex
initial_x=10;

[t,x]=ode45( @rhs, t, initial_x);

plot(t,x);
xlabel('t'); ylabel('x');

    function dxdt=rhs(t,x)
        dxdt =x*(1-x)-10/(1+x);
    end
end
```

Figure:



2. Fourier Series

Eq: f(t）= $\begin{cases} 0 & -\dfrac{T}{2} \le t < 0 \\ 1 & 0 \le t \le \dfrac{T}{2} \end{cases}$

We get $a_0 = \frac{1}{2}$ $a_{k=0}$ $b_k = \frac{1}{k\pi}(1 - \cos k\pi)$

the Fourier Series of the function $f(t) = \frac{1}{2} + \sum_{k=1}^{\infty} \frac{1}{k\pi}(1 - \cos k\pi) \sin k\frac{2\pi}{T} t$

let T=1 w0=2pi

Code:

```
% Fourier series of periodic square wave signals
t = -1:0.001:1;
w0 = 2 * pi;
y =max(square(pi*2*t,50),0);
%Periodic square wave signal
plot(t,y), grid on;
axis([-1 1 -1.5 1.5]);
n_max = [1 3 5 7 31];
N = length(n_max);
for k = 1:N
    n = 1:2:n_max(k);
    b = (1-cos(n*pi))./(pi*n);
    x =0.5+b *sin(w0*n'*t);
    figure;
    plot(t,y);
    hold on;
    plot(t,x);
    hold off;
    axis([-1 1 -1.5 1.5]),grid on;
    title(['Maximum harmonic number=',num2str(n_max(k))]);
end
```

Original figure:

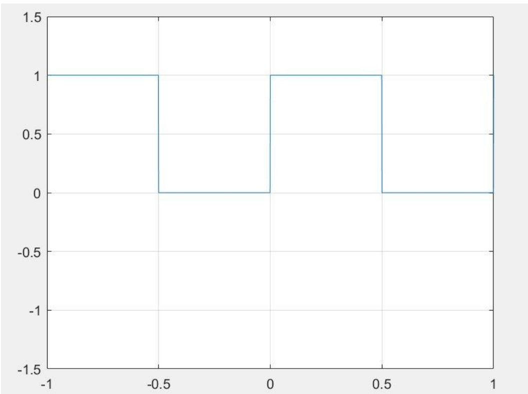
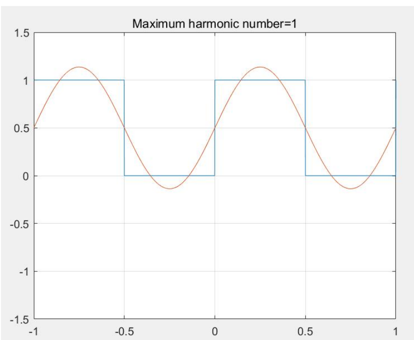
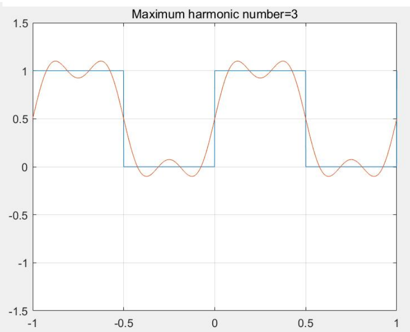
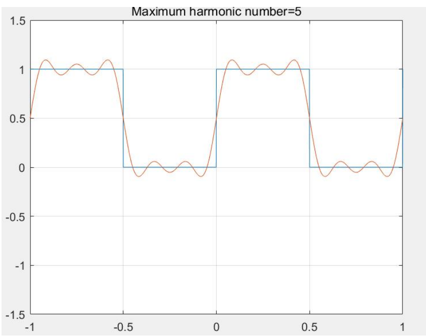
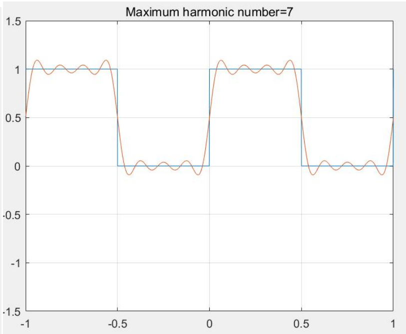


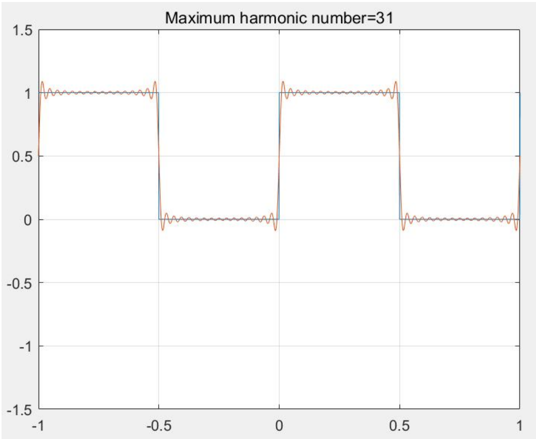
Figure with Fourier number n=1,3,5,7,31 :





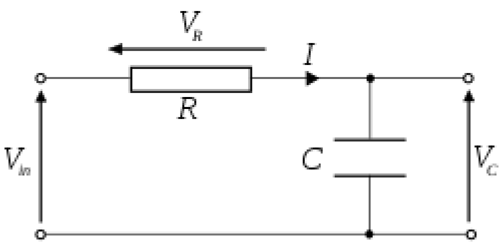






3. First order systems

RC circuits:



Eq:

$$\frac{dv_c}{dt} + \frac{1}{RC}v_c = \frac{1}{RC}v(t)$$

Solution:

$$v_c(t) = v_c(0)e^{-\frac{t}{RC}} + \frac{1}{RC}\int_0^t v(\tau)e^{-\frac{1}{RC}(t-\tau)}d\tau$$

Let R=1 C=1 Vi=1

Code:

```
%% impulse response, step response, ramp response, and frequency response:
% Impulse V(t) = dirac(t)*Vi
% Step:  V(t) = 1*Vi
% Ramp: V(t) = t*Vi
% Frequency: V(t) = exp(jwt)*Vi
%V(t)/(R*C)-Vc/(R*C)=dVc/dt
t = -1:0.001:1;
Vi = 1;
w  = 1;
R  = 1;
C  = 1;
% w  = logspace(1,4,20);
figure();
```

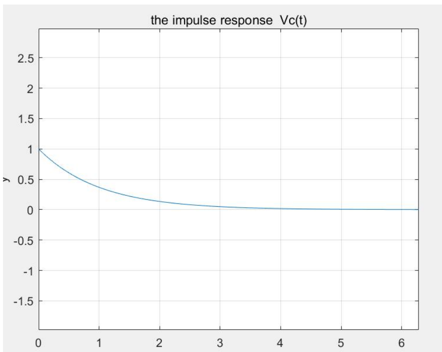
(1) impulse response

$$V(t)=\begin{cases} 0 & t<0 \\ \delta(t) & 0\le t \end{cases}$$

Code:

```
%% impulse response:
syms  t s
V =dirac(t)*Vi;
Vir = Vi;
Vcir = Vir/(s+1);
V1 = ilaplace(Vcir);
ezplot(t,V1);
title('the impulse response  Vc(t)')
grid on;

figure();
```

Figure:



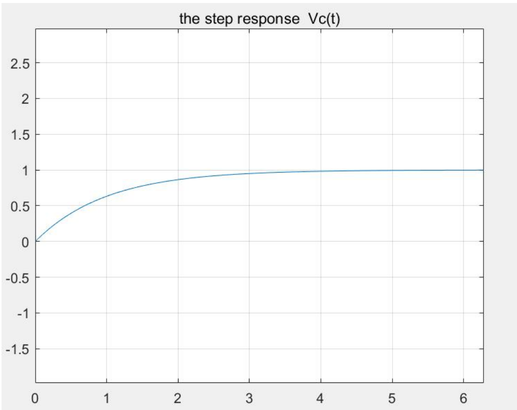
(2) step response

$$V(t)=\begin{cases} 0 & t<0 \\ 1 & 0\le t \end{cases}$$

Code:

```
%% step response:
syms  t s
V     = 1*Vi;
Vsr  = Vi/s;
Vcsr = Vsr/(s+1);
V2 = ilaplace(Vcsr);
ezplot(t,V2);
title('the step response  Vc(t)')
grid on;

figure();
```

Figure:



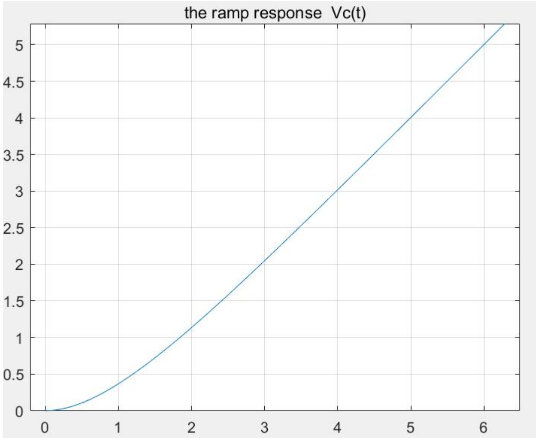
(3) ramp response,

$$V(t)=\begin{cases} 0 & t<0 \\ t & 0\le t \end{cases}$$

Code:

```
%% ramp response:
syms  t s
V     = Vi*t;
Vrr  = Vi/s^2;
Vcrr = Vrr/(s+1);
V3 = ilaplace(Vcrr);
ezplot(t,V3);
title('the ramp response  Vc(t)')
grid on;

figure();
```

Figure:



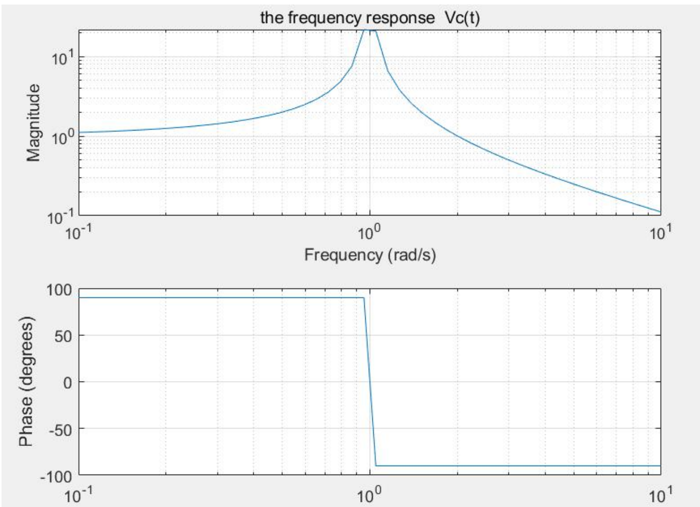
(4) frequency response

$$V(t)=\begin{cases} 0 & t<0 \\ e^{jw} & 0 \le t \end{cases}$$

Code:

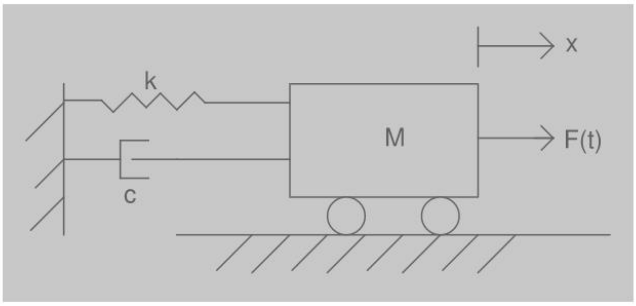
```
%% frequency response:
syms t s
V      = Vi.*exp(1i*w*t);
Vfr   = Vi*(s+1i)/(s^2+1);
Vcfr = Vfr/(s+1);
a = [1 0 1];
b = [0 1 1i];
w = logspace(-1,1);
freqs(b,a,w)
title('the frequency response  Vc(t)')
grid on;
```

Let w=1, then figure:



4. Second order systems

translational spring-mass-damper system:



We can get

$$m\ddot{x} + c\dot{x} + kx = F(t)$$

$$\ddot{x}+2\varepsilon\omega_n\dot{x}+\omega_n^2 x=f(t)$$

where $2\varepsilon\omega_n=\frac{c}{m}$, $\omega_n^2 = \frac{k}{m}$,$f(t)=\frac{F(t)}{m}$

(1) $f(t)=\begin{cases} 0 & t < 0 \\ F_0 & t \geq 0 \end{cases}$

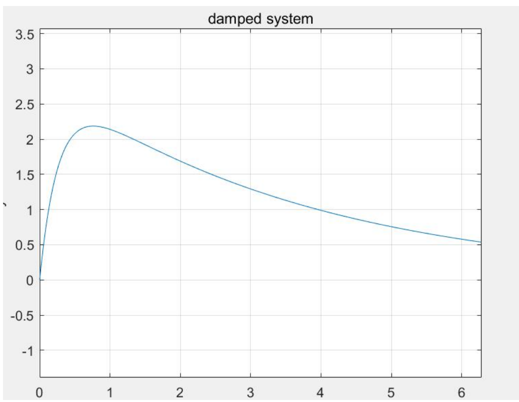
We choose F_0=10 $\omega_n = 1$

ε>1 damped system ε=2

Code:

```
%F=10 W=1f(t)=1*F a= ε
%% damped system:
syms  t s
a1=2;
X=10/(s^2+2*a1*s+1);
x = ilaplace(X);
ezplot(t,x);
title('damped system')
grid on;
figure();
```

Figure:

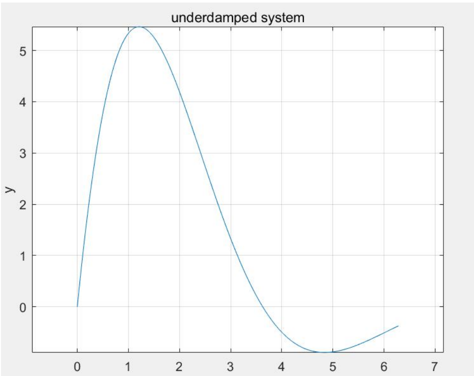


ε<1   underdamped   system   ε=0.5

Code:

```
%% underdamped system:
syms  t s
a2=0.5;
X=10/(s^2+2*a2*s+1);
x = ilaplace(X);
ezplot(t,x);
title('underdamped system')
grid on;
figure();
```

Figure:

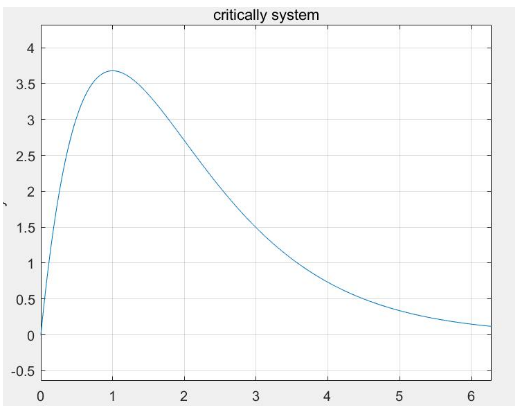


ε=1   critically damped   system

Code:

```
%% critically system:
syms  t s
a3=1;
X=10/(s^2+2*a3*s+1);
x = ilaplace(X);
ezplot(t,x);
title('critically system')
grid on;
```

Figure:



(2) f(t)= $\begin{cases} 0 & t < 0 \\ F_0 e^{jw} & t \geq 0 \end{cases}$

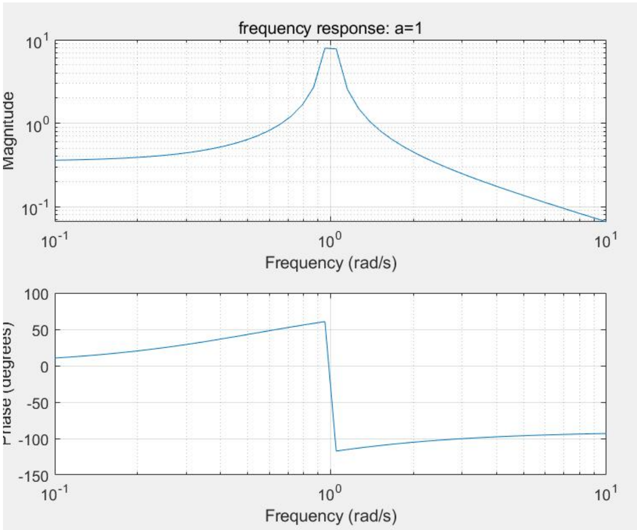
We choose F_0=1 $\omega_n = 1$ w=1

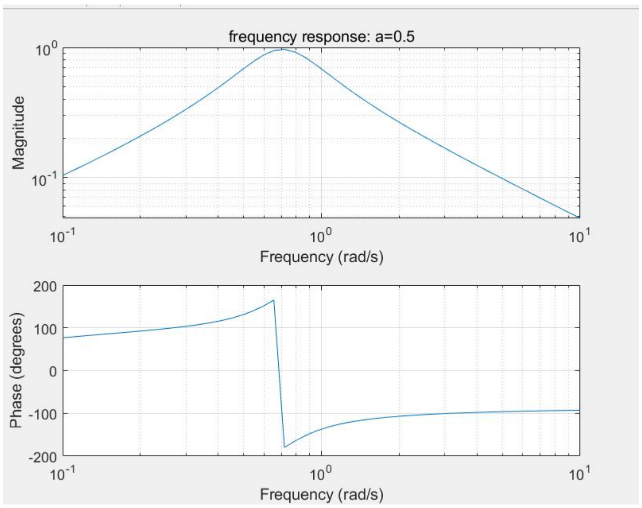
ε=0.5,1,2

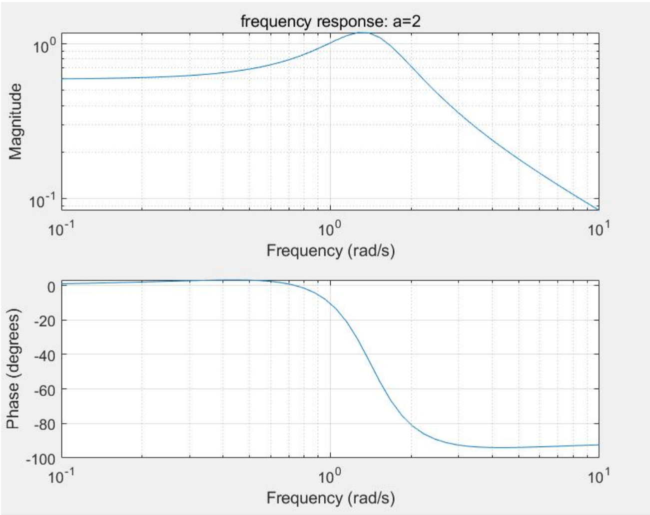
Code:

```
count  = 0;
StrName = {'a=2';'a=1';'a=0.5'};
for i = 1:3
    Value = [0.5,1,2];
    M = Value(i);
    count = count + 1
    figure()
    a0 = [1 (1/M-a) a*1/M];
    TMP1 = C+1*1/(a*M+1);
    TMP2 = 1*1/(a*M+1)*1/M-a*C;
    b = [0 TMP1 TMP2];
    w = logspace(-1,1);
    freqs(b,a0,w)
    title(['frequency response: ',StrName{count}])
    grid on;

end
```

Figure:






frequency response: a=2
Magnitude
10^0
10^{-1}
10^{-1}
10^0
10^1
Frequency (rad/s)
Phase (degrees)
0
-20
-40
-60
-80
-100
10^{-1}
10^0
10^1
Frequency (rad/s)