# EXAMPLE – desolve

```
[x, y] = dsolve('Dx+y=0,D2y-Dy=0')

[x, y] = dsolve('Dx+y=0,D2y-Dy=0','x(0)=2,y(0)=1,Dy(0)=1')

[x, y] = dsolve('Dx+y=0,D2y-Dy=0','x(0)=2,y(0)=1,Dy(0)=1,D2y(0)=4')
```

What will happen if input **contains third-order differential** equation?

```
[x, y] = dsolve('Dx+y=0,D3y-Dy=0','x(0)=2,y(0)=1,Dy(0)=1,D2y(0)=4')
```

If the input like this,

```
[x, y] = dsolve('Dx+y=0,D6y-D4y=0','x(0)=2,y(0)=1,Dy(0)=1,D2y(0)=4')
```

If you do not like ‘t’ as variant

```
[x, y] = dsolve('Dx+y=0,D6y-D4y=0','x(0)=2,y(0)=1,Dy(0)=1,D2y(0)=4','p')
```

# EXAMPLE – ode

```
tspan = [0 5];
y0 = 0;
[t,y] = ode45(@(t,y) 2*t^2 + t, tspan, y0);
plot(t,y);
```

What if a more complex equation?

$$y_1'' - \mu\left(1 - y_1^2\right) y_1' + y_1 = 0,$$

$$\begin{cases} dx(1) = x(2) \\ dx(2) = x(3) \\ \ldots\ldots \\ dx(n-1) = x(n) = f(t, x(n-1), \ldots x(2), x(1)) \end{cases}$$

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= \mu(1 - y_1^2)y_2 - y_1. \end{aligned}$$

# EXAMPLE – ode

```
function dx=myequ(t,x)
   dx = zeros(2,1);
   dx(1) = x(2);
   dx(2) = (1-x(1)^2)*x(2)-x(1);
end

function dx=myequ(t,y)
   dx = [x(2); (1 – x(1) ^ 2) * x(2) – x(1)]

Coefficient input:

function dx=myequ(t,x, alpha, beta)
   dx = zeros(2,1);
   dx(1) = x(2);
   dx(2) = (1-alpha*x(1)^2)*x(2)-beta*x(1);
End

function dx=myequ(t,y,alpha,beta)
   dx = [x(2); (1 – alpha*x(1) ^ 2) * x(2) – beta*x(1)]
```

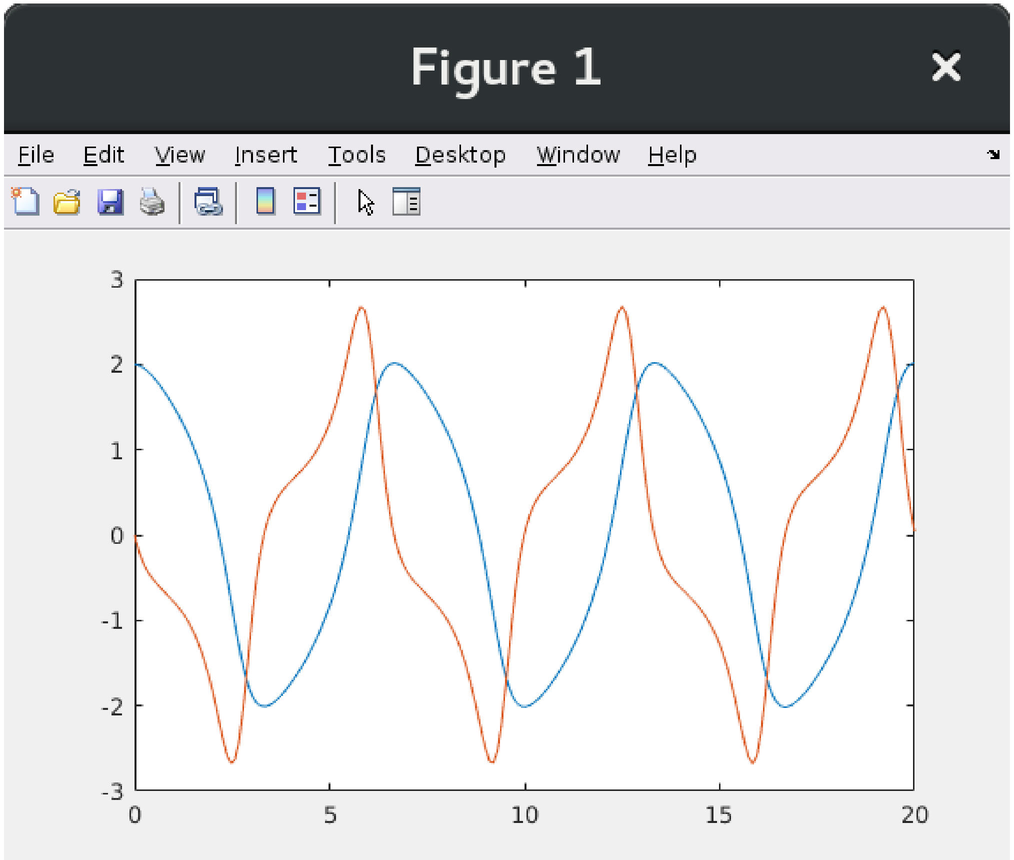
```
[t,y] = ode45(@(t,y) odefcn(t,y,A,B), tspan, y0)
```

# EXAMPLE – ode

SpanT = [0 20]
X0 = [2 0]

[t, x] = ode45(@myequ,SpanT,X0);

plot(t, x)



# EXAMPLE – ode

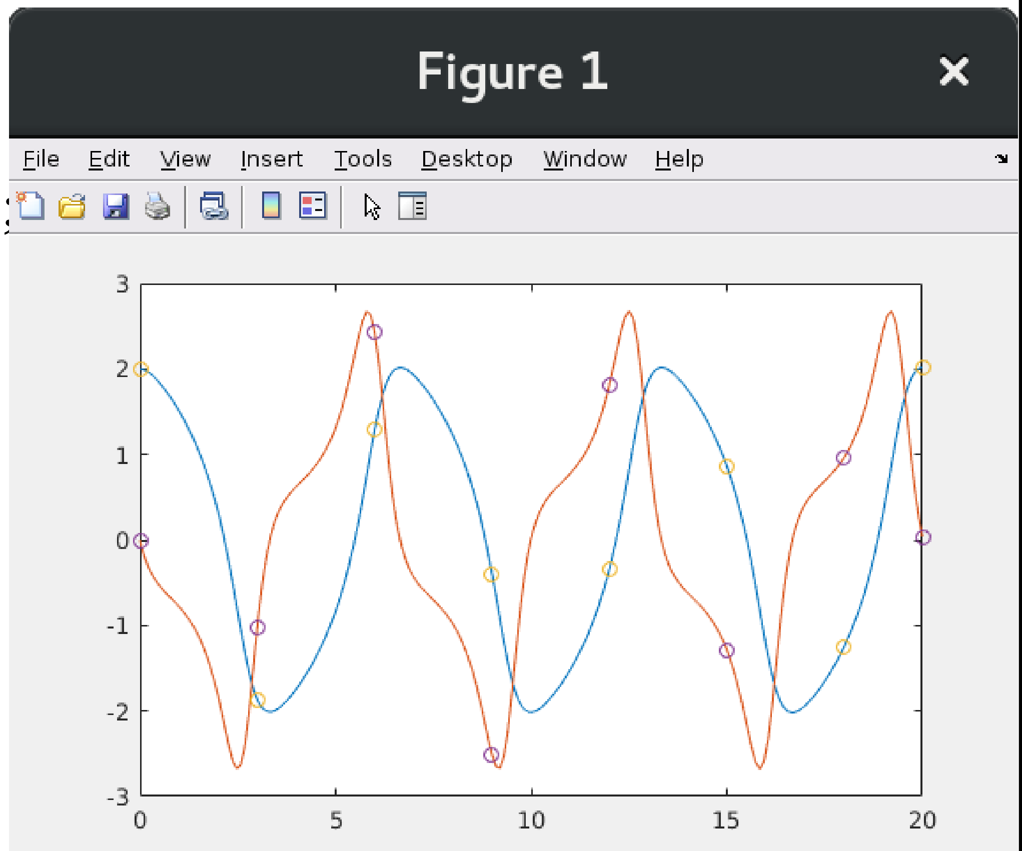
If we only what some solutions for certain t:

...

Hold on
SpanT = [0 3 6 9 12 15 18 20]
X0 = [2 0]

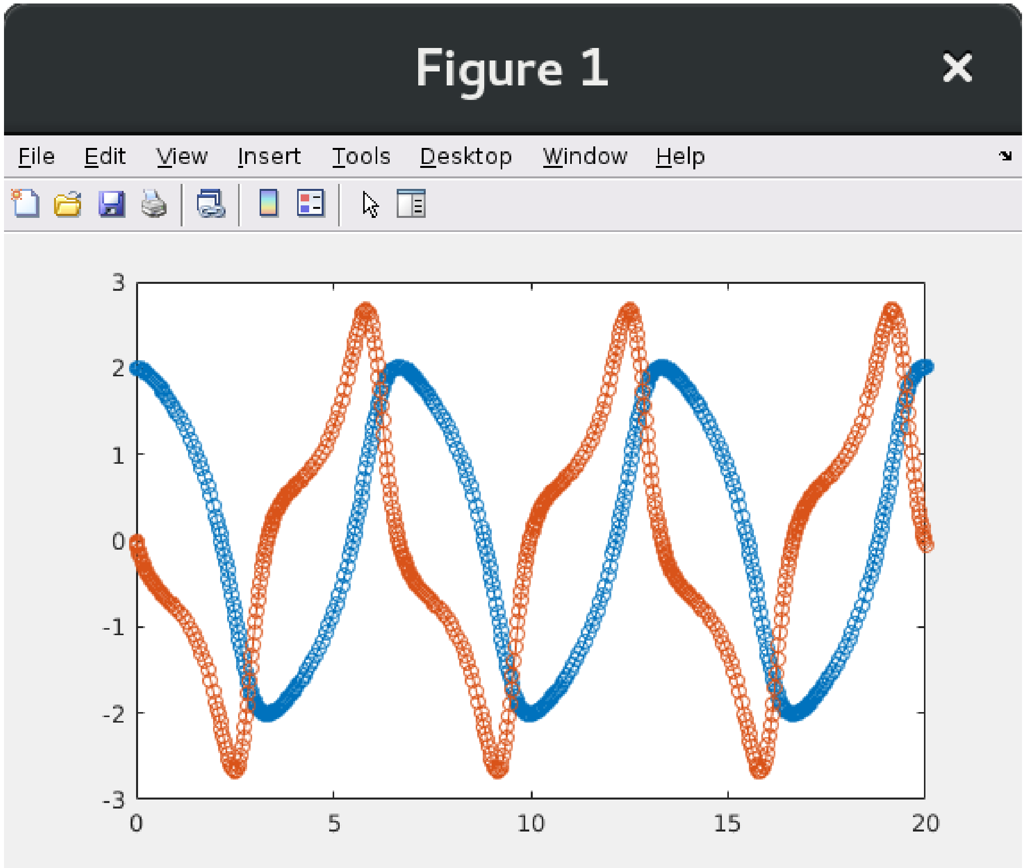
[t, x] = ode45(@myequ,SpanT,X0);

plot(t, x, ‘o’)



# EXAMPLE – ode

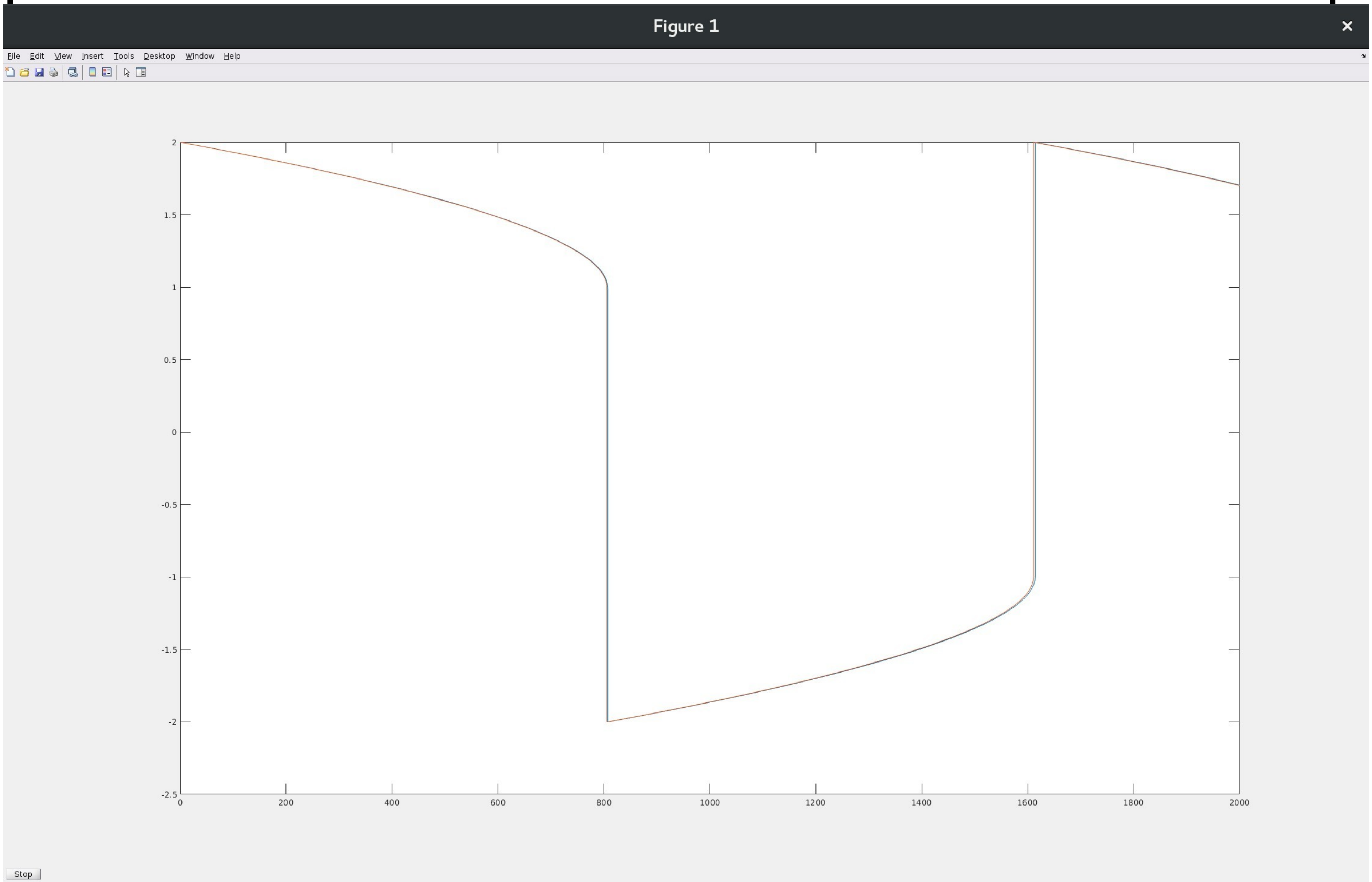
```
SpanT = [0 20]
X0 = [2 0]
options = odeset('RelTol',1e-5,'Stats','on','OutputFcn',@odeplot)

[t, x] = ode45(@myequ,SpanT,X0, options);
```



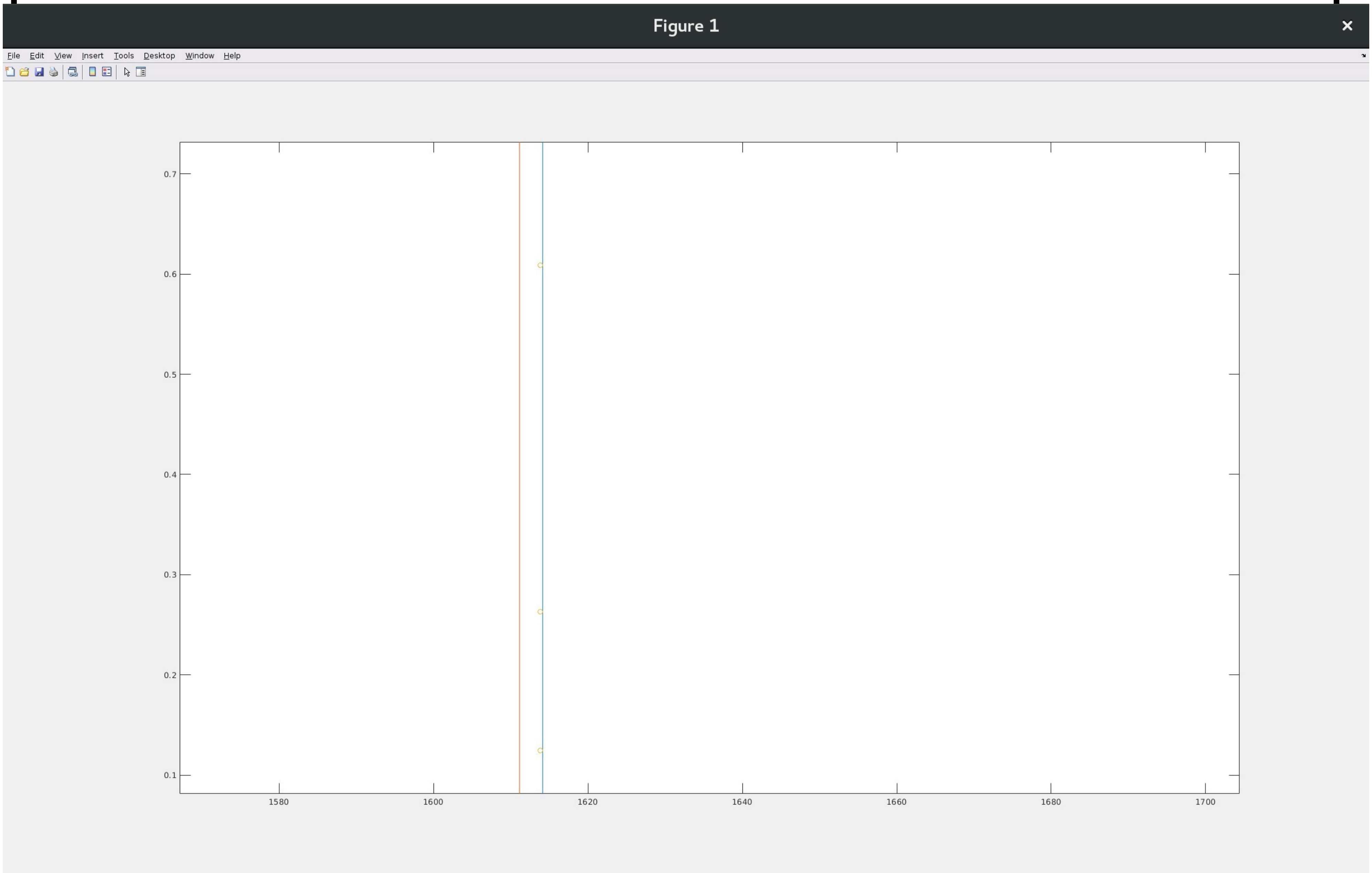
# EXAMPLE – Compare different ode methods

```
function dx=myequ3(t,x)
dx = [x(2); 1000*( 1 - x(1) ^ 2) * x(2) - x(1)];
End

[t, y] = ode45(@myequ3,[0 2000],[2 0]);
[t2, y2] = ode15s(@myequ3,[0 2000],[2 0]);
plot(t, y(:,1))
hold on
plot(t2, y2(:,1))

[t 3 , y 3 ] = ode113(@myequ3,[0 2000],[2 0]);
hold on
plot(t 3 , y 3 (:,1))
```

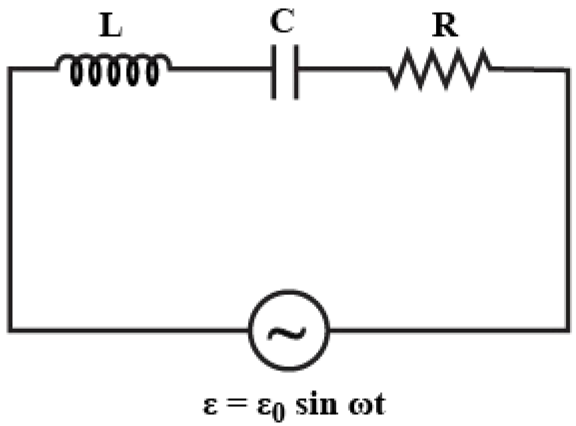
# EXAMPLE – Compare different ode methods



# EXAMPLE – Compare different ode methods



# EXAMPLE – LCR circuit




**A series LCR circuit consists of an inductor L a capacitor C and a resistor R connected across a source of emf ε=ε0 sin ωt.**

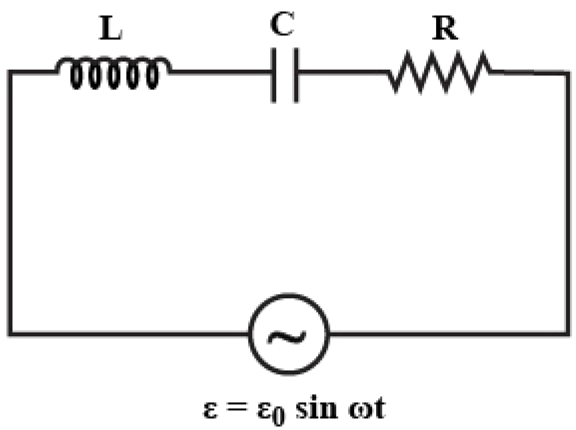
$$\frac{d^2I}{dt^2}+\frac{R}{L}\frac{dI}{dt}+\frac{I}{LC}=\frac{1}{L}\frac{de(t)}{dt}$$

We set L = 10, R = 200 , C = 5, w = 100

The equation can be changed as

D^2I/Dt^2 = 10 * cos(100t) – 20 * DI/Dt - I  / 50

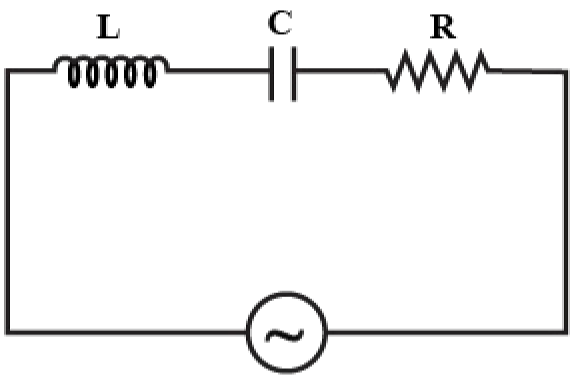
# EXAMPLE – LCR circuit



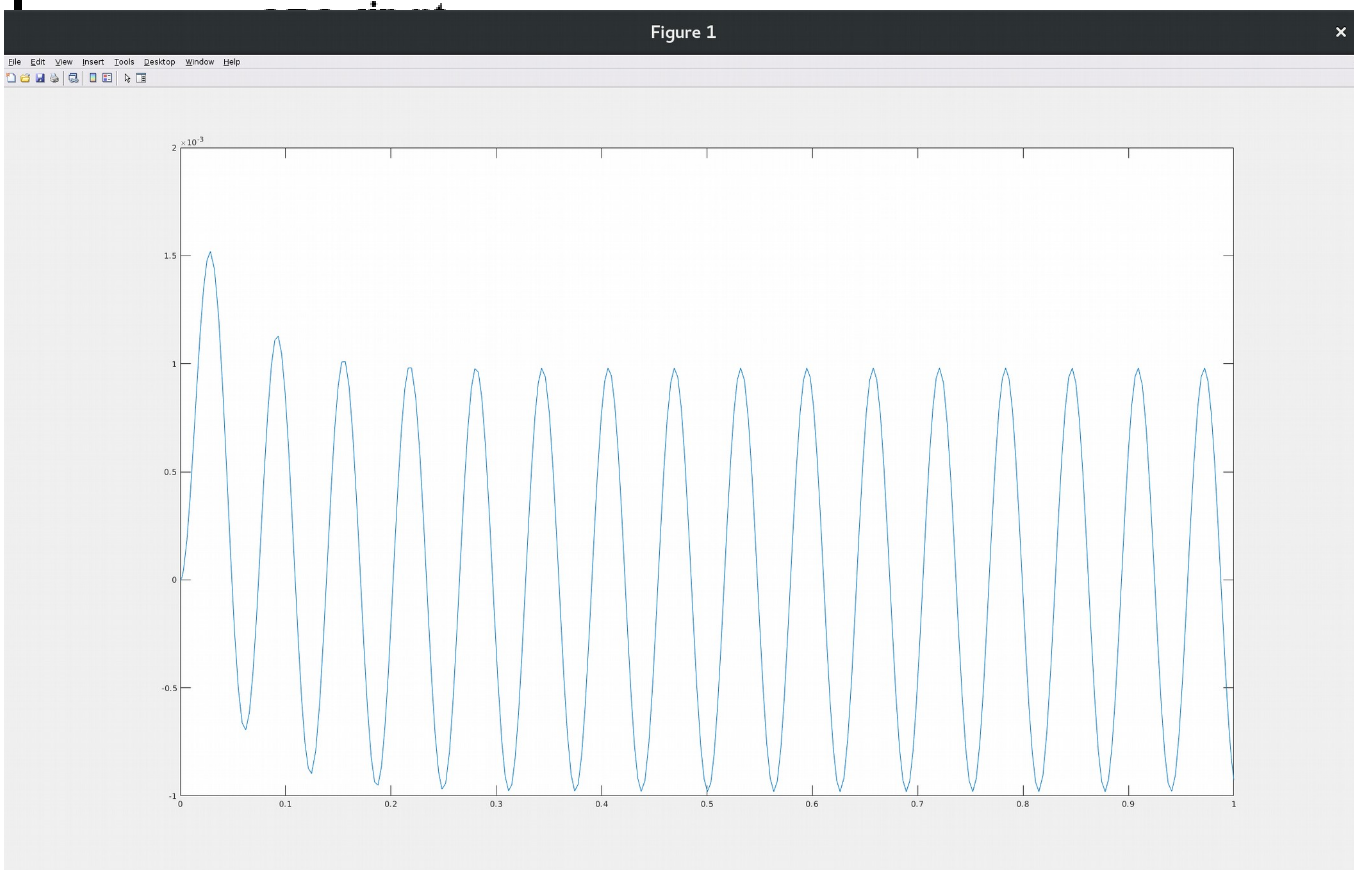

**A series LCR circuit consists of an inductor L a capacitor C and a resistor R connected across a source of emf ε=ε0 sin ωt.**

```
function dx=LCRequ(t,x)
dx = [x(2); 10*cos(100*t)-20*x(2)-x(1)/50];
End

SpanT = [0 1]
X0 = [0 0]
[t x] = ode45(‘LCRequ’,SpanT,X0)
```

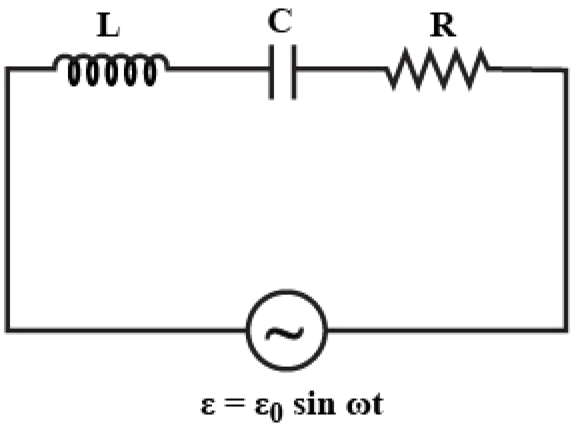
# EXAMPLE – LCR circuit



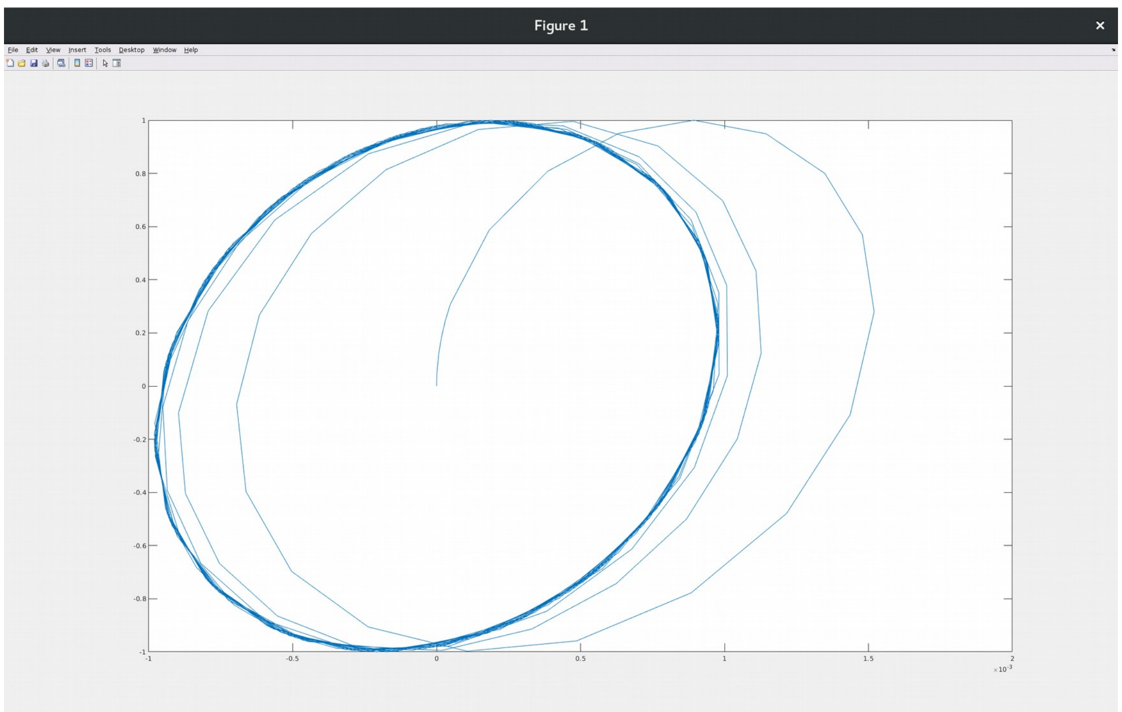

plot(t, x(:,1))



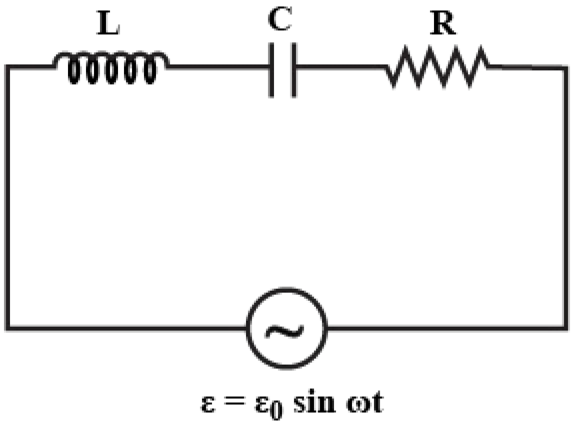
# EXAMPLE – LCR circuit




plot(x(:,1), sin(100t))

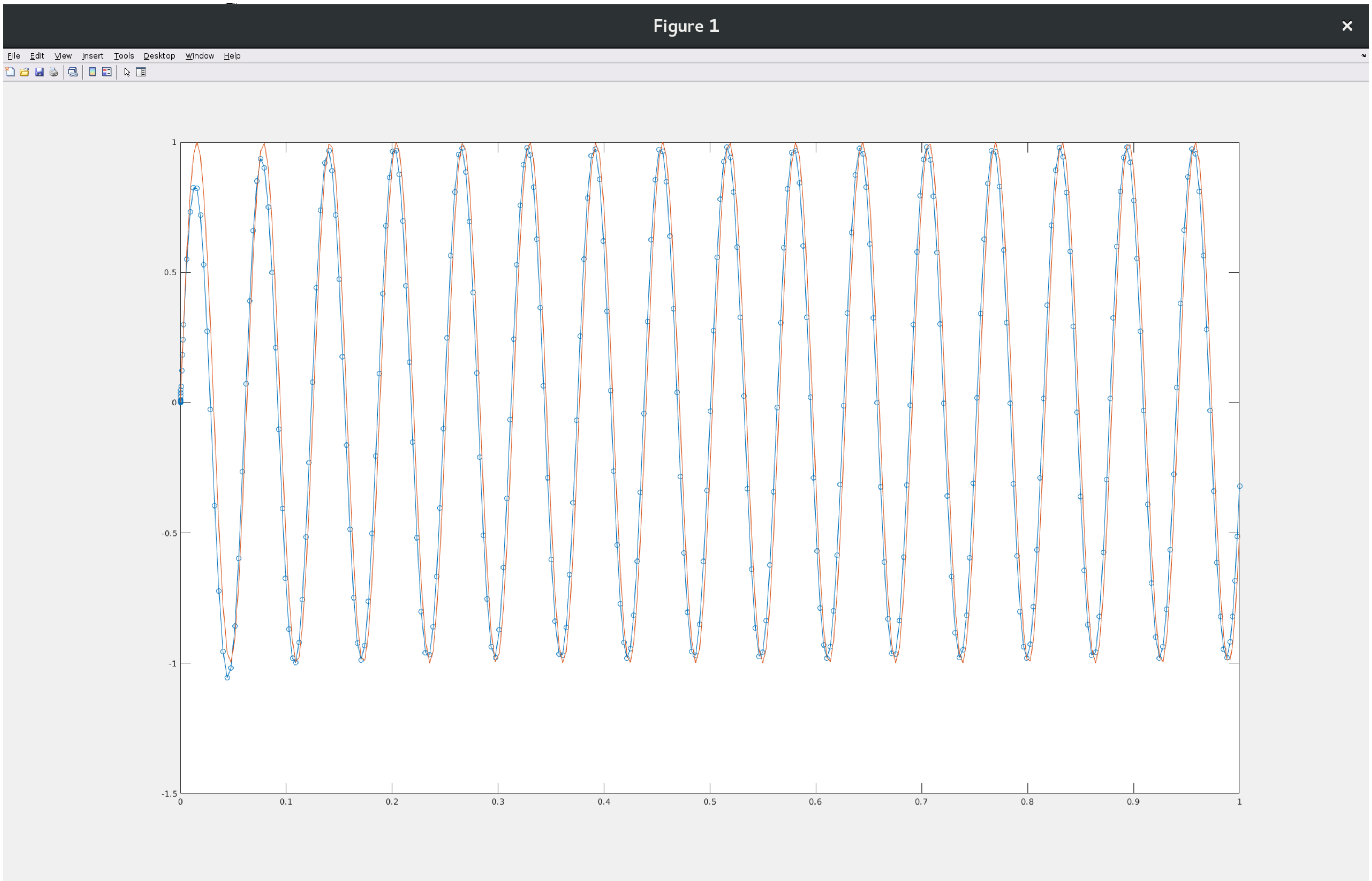


# EXAMPLE – LCR circuit

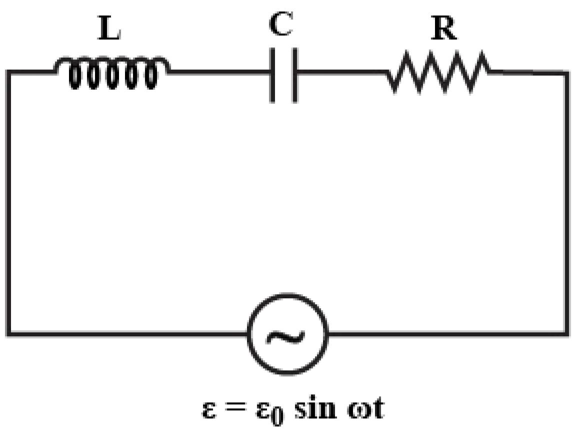



plot(t,10*x(:,2), '-o',t, sin(100*t), '-')

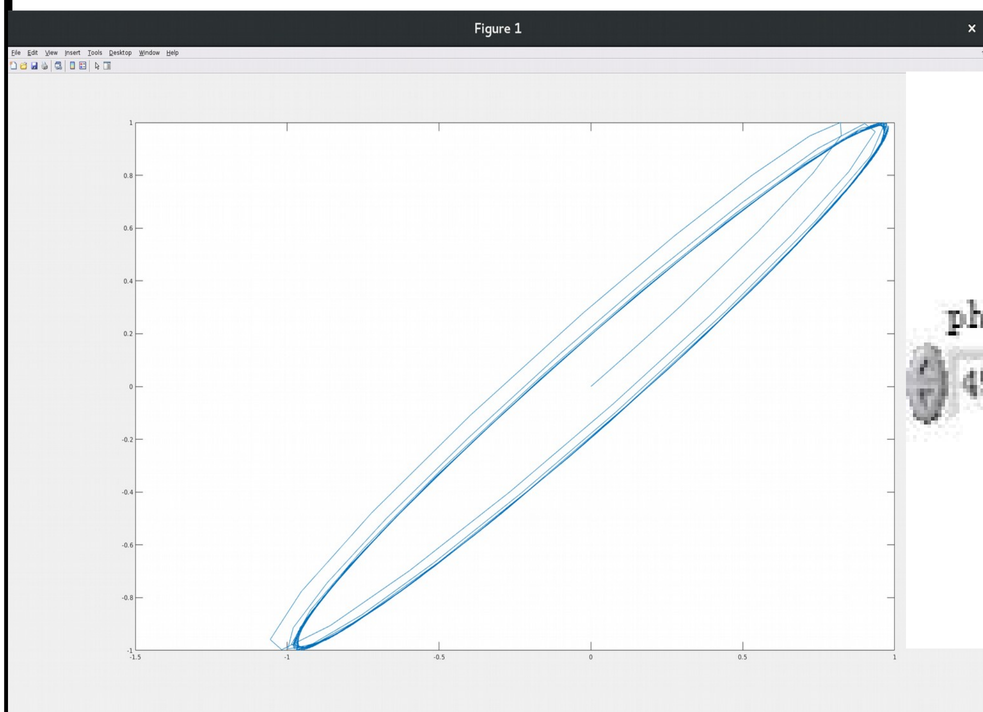
# EXAMPLE – LCR circuit

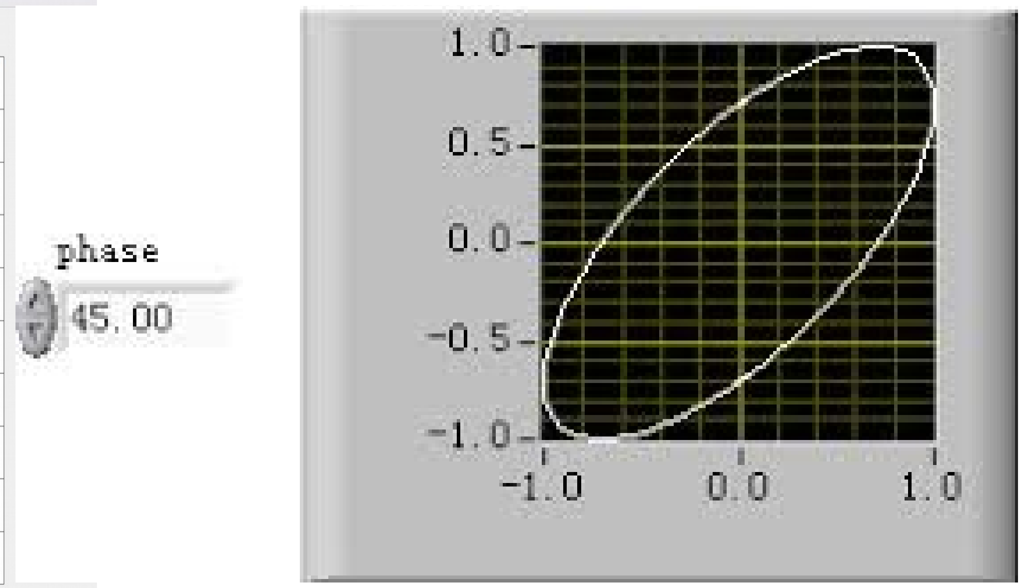


# EXAMPLE – LCR circuit




Plot(10 * x(:,2), sin(100t))



# EXAMPLE – LCR circuit