

# **Matlab ile Diferansiyel Denklem COZUMLERI**

- Bir ya da daha fazla fonksiyonun türevlerini içeren denklemlere diferansiyel denklem denir.
- Bir yada daha fazla bağımlı değişken ve bir bağımsız değişkenden oluşan diferansiyel denklemlere adi diferansiyel denklem denir.
- Bir veya daha fazla bağımlı ve birden fazla bağımsız değişkenin olduğu diferansiyel denklemlere de kısmi diferansiyel denklem denir.

**MATLAB ortamında diferansiyel denklemler hem sayısal hem de sembolik (analitik) çözülebilir**

# Sembolik Diferansiyel Denklem Terimleri

$y$	$y$
$\frac{dy}{dt}$	Dy
$\frac{d^2 y}{dt^2}$	D2y
$\frac{d^n y}{dt^n}$	Dny

Verilen bağımsiz değişken  $t$  ve bağımlı değişken  $y(t)$ ,

Burada  $A_0, A_1, \dots, A_n$ , sabitlerdir.

$$A_n \frac{d^n y}{dt^n} + \dots + A_1 \frac{dy}{dt} + A_0 y(t) = f(t)$$

Matlab'ta diferansiyel denklemlerin sembolik çözümleri "dsolve" komutu ile bulunur.

$$b_2 \frac{d^2 y}{dt^2} + b_1 \frac{dy}{dt} + b_0 y = A \sin at$$

$$y(0) = C_1 \quad \text{and} \quad y'(0) = C_2$$

```
>> y = dsolve('b2*D2y+b1*D1y+b0*y=A*sin(a*t)',  
              'y(0)=C1', 'Dy(0)=C2')
```

```
>> ezplot(y, [t1 t2])
```

$$x \frac{dy}{dx} + y = e^{2x} - \sin(x)$$

```
Command Window
>> f=dsolve('x*Dy+y=exp(2*x)-sin(x)')

f =

exp(2*x) - sin(x) - C5/exp(t/x)

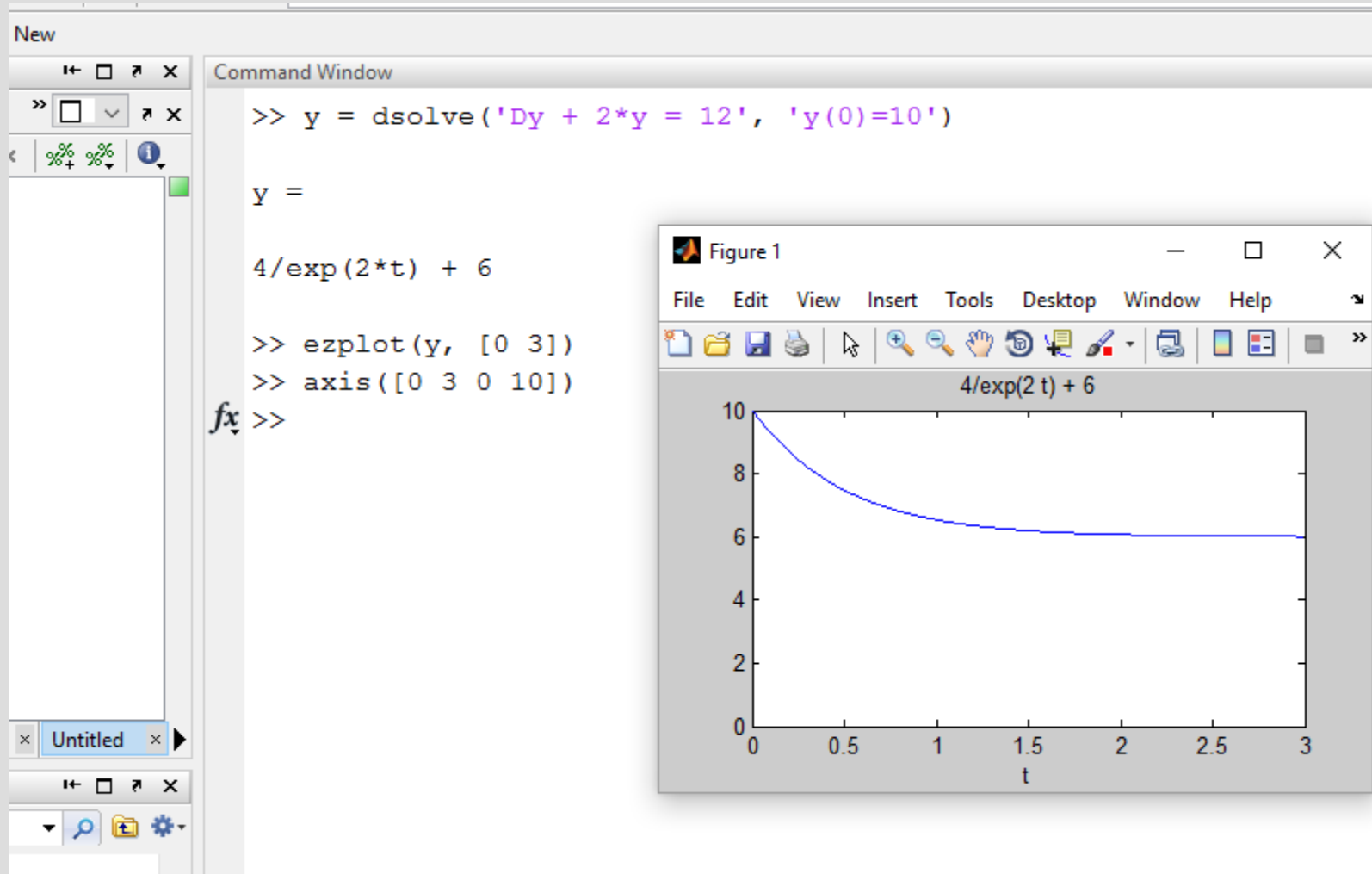
>> pretty(f)

          C5
exp(2 x) - sin(x) - ----
                    / t \
                   exp| - |
                    \ x /

fx >> |
```

$$\frac{dy}{dt} + 2y = 12$$

$$y(0) = 10$$



# Ornek

$$\frac{dy}{dt} + 2y = 12 \sin 4t \quad y(0) = 10$$

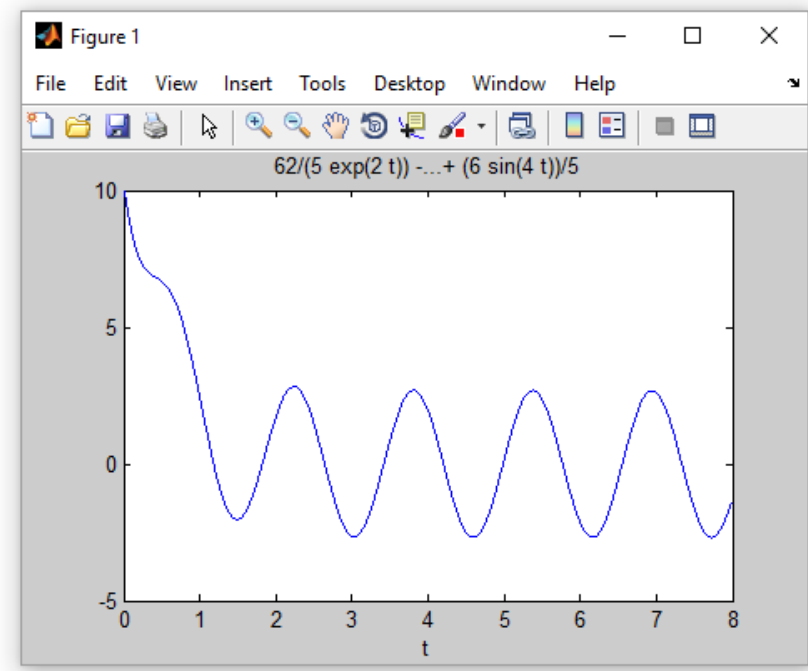


Editor - Untitled

```
1
```

Command Window

```
>> y=dsolve('Dy + 2*y = 12*sin(4*t)', 'y(0)=10')  
  
y =  
  
62/(5*exp(2*t)) - (12*cos(4*t))/5 + (6*sin(4*t))/5  
  
>> ezplot(y, [0 8])  
>> axis([0 8 -5 10])  
fx >>
```



Current Folder

<< MATLAB >> R2011a >> bin >

Name

- m3registry
- registry
- util
- win64

Details

$$\frac{d^2 y}{dt^2} + 3 \frac{dy}{dt} + 2y = 24$$

$$y(0) = 10$$

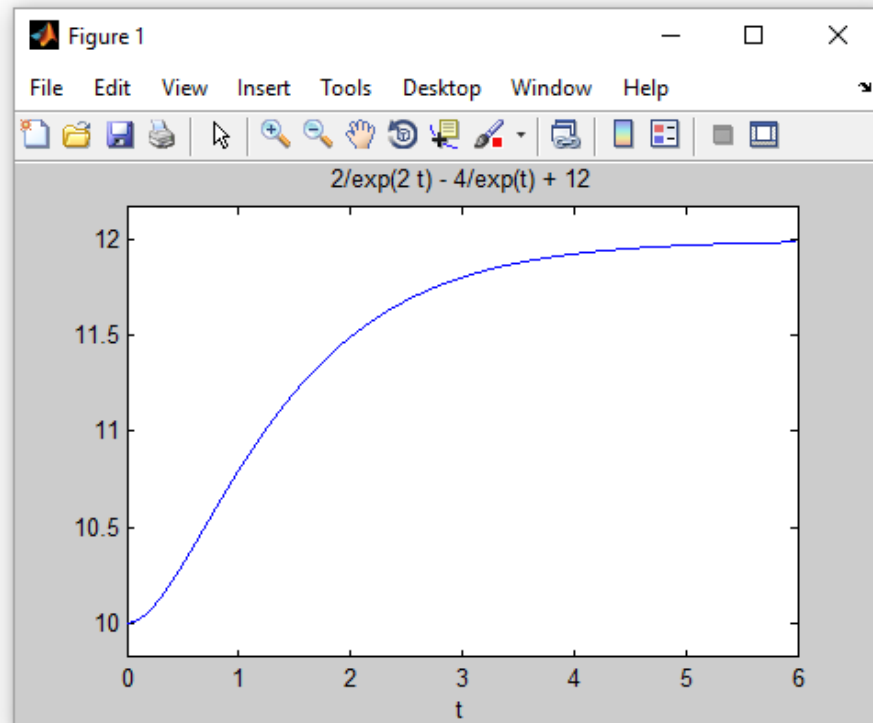
$$y'(0) = 0$$

```
Command Window
>> y = dsolve('D2y + 3*Dy + 2*y = 24', 'y(0)=10', 'Dy(0)=0')

y =

2/exp(2*t) - 4/exp(t) + 12

>> ezplot(y, [0 6])
fx >>
```



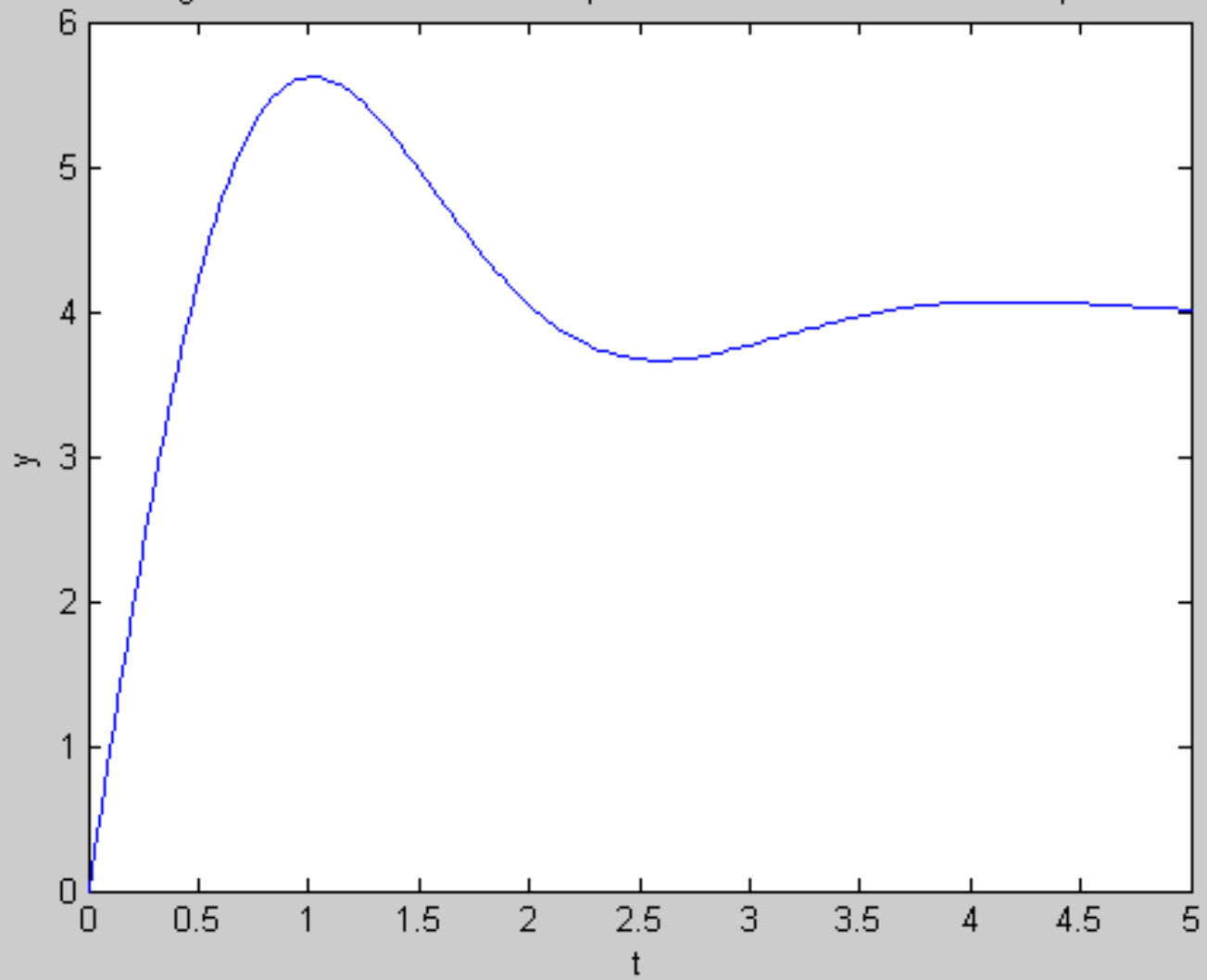
$$\frac{d^2 y}{dt^2} + 2 \frac{dy}{dt} + 5y = 20 \quad y(0) = 0 \quad y'(0) = 10$$

The image shows a MATLAB 7.12.0 (R2011a) interface. The Command Window contains the following code and output:

```
>> y = dsolve('D2y + 2*Dy + 5*y = 20','y(0) = 0','Dy(0) = 10')
y =
(3*sin(2*t))/exp(t) - (4*cos(2*t))/exp(t) + 4
>> ezplot(y, [0 5])
fx >>
```

The Figure 1 window displays a plot of the solution  $y(t)$  for  $t$  in the range  $[0, 5]$ . The plot shows a blue curve that starts at  $(0, 0)$ , rises to a peak of approximately 5.5 at  $t \approx 1$ , then decays and oscillates, crossing the  $t$ -axis at  $t \approx 2.5$ , and eventually levels off around  $y = 4$  for  $t > 4$ . The title of the plot is  $(3 \sin(2 t))/\exp(t) - (4 \cos(2 t))/\exp(t) + 4$ .

Figure 11-4. Solution of Example 11-4 based on dsolve and ezplot.



Örnek: 
$$\frac{d^2x}{dt^2} + 0.2 \frac{dx}{dt} + x = 10e^{-t}$$

$$x(0)=0 \ ; \ x'(0)=5$$

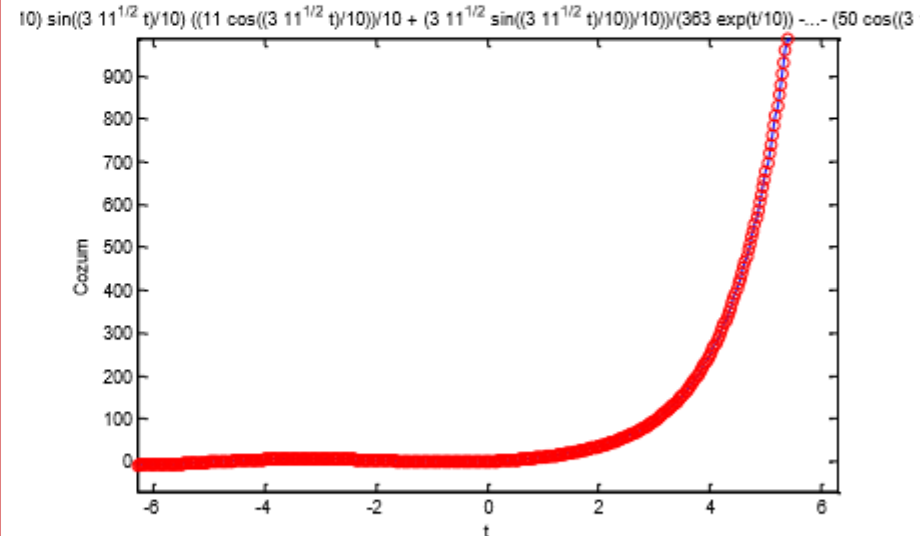
```
>>Cozum=dsolve('D2x+0.2*Dx+x=10*exp(t)', 'x(0)=0',  
Dx(0)=5')
```

Cozum =

$$\begin{aligned} & (500 \cdot 11^{1/2} \cdot \exp((11 \cdot t)/10) \cdot \sin((3 \cdot 11^{1/2} \cdot t)/10) \\ & ) \cdot ((11 \cdot \cos((3 \cdot 11^{1/2} \cdot t)/10))/10 + \\ & (3 \cdot 11^{1/2} \cdot \sin((3 \cdot 11^{1/2} \cdot t)/10))/10) / (363 \cdot \exp(t/10)) - \\ & (500 \cdot 11^{1/2} \cdot \exp((11 \cdot t)/10) \cdot \cos((3 \cdot 11^{1/2} \cdot t)/10) / 10) \\ & \cdot ((11 \cdot \sin((3 \cdot 11^{1/2} \cdot t)/10))/10 - \\ & (3 \cdot 11^{1/2} \cdot \cos((3 \cdot 11^{1/2} \cdot t)/10))/10) / (363 \cdot \exp(t/10)) - \\ & (50 \cdot \cos((3 \cdot 11^{1/2} \cdot t)/10)) / (11 \cdot \exp(t/10)) \end{aligned}$$

```
>> ezplot(Cozum)
```

diferansiyel denklemini başlangıç koşullarına göre çözdürün ve grafiğini çizin.



$x''' + 2x' = 2t$  Şeklindeki diferansiyel denklemini aşağıdaki sınır şartları ile dsolve komutu ile çözdürüp grafiğini çizelim;

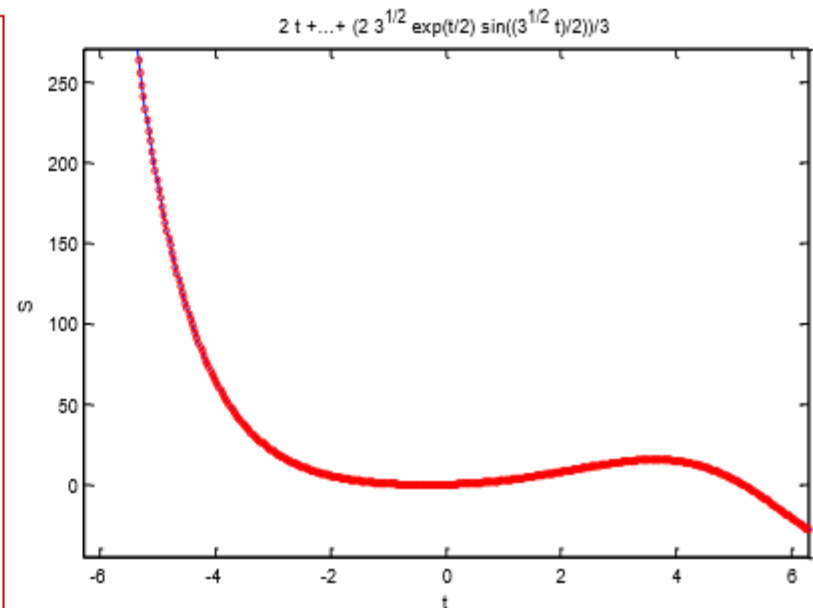
$$x(0)=0 ; x'(0)=1 ; x''(0)=3$$

```
>>  
S=dsolve('D3x+x=2*t','x(0)=0','Dx(0)=1','D2x(0)  
=3')
```

S =

$$2*t + 4/(3*\exp(t)) - (4*\exp(t/2)*\cos((3^{1/2}*t)/2))/3 + (2*3^{1/2}*\exp(t/2)*\sin((3^{1/2}*t)/2))/3$$

```
>> ezplot(S)
```



Cozmeyi deneyiniz

$$2 \frac{dy}{dt} + y = 8$$

$$\frac{d^2 y}{dt^2} + 5 \frac{dy}{dt} + 6y = 3t$$

$$3 \frac{d^4 y}{dt^4} - \frac{d^3 y}{dt^3} + 8 \frac{dy}{dt} = \sin t$$

$$y'' - 3y' + 4y = 0$$

$$y^{(4)} + 8y^{(3)} + 24y'' + 32y' + 16y = 0$$

$$100 \frac{d^2 N}{dt^2} - 20 \frac{dN}{dt} + N = 0$$

$$y'' - y' - 2y = 4x^2$$

$$y'' - y' - 2y = e^{3x}$$

$$y'' - y' - 2y = \sin 2x$$

$$\frac{dI}{dt} + 20I = 6 \sin(2t)$$

$$xy' + y = xy^3$$

$$y' = \frac{y+x}{x}$$

$$\frac{dT}{dx} + kT = 100k$$

Adi diferansiyel denklemlerini çözünüz.



Diferansiyel denklemlerin sayısal çözümleri için bu denklemleri ikiye ayırmak mümkün.

- Doğrusal (Linear) Diferansiyel Denklemler
- Doğrusal olmayan (Nonlinear) Diferansiyel Denklemler

Doğrusal diferansiyel denklemlerin çözümü analitik olarak kolay yapılabilirken, doğrusal olmayanları yapmak zordur.

Bunun için Euler ve Runge-Kutta yaklaşımları gibi çeşitli sayısal yöntemler geliştirilmiştir

Matlab'ta diferansiyel denklemlerin çözümünde kullanılan “ode23” ve “ode45” adında iki adet Runge-Kutta fonksiyonu mevcuttur.

Bunlardan “ode23” fonksiyonu ikinci ve üçüncü dereceden Runge-Kutta integrasyon denklemlerini,

ode45” fonksiyonu ise dördüncü ve beşinci dereceden Runge-Kutta integrasyon denklemlerini kullanır.

Bu denklemlerin çözümü için de m-file dosyasının oluşturulması gerekir.

NOT :Diferansiyel denklemlerin MATLAB’da sayısal çözümlerinde ode23, ode45, ode113, ode15s, ode23s, ode23t, ode23tb, ode15i olarak adlandırılan komutlar kullanılır. Bu komutlarda kullanılan çözüm yöntemleri farklıdır

## .m dosyası hazırlayarak Diferansiyel Denklem Çözümü

❖ ode23 veya ode45 fonksiyonu aşağıdaki gibi kullanılır:  
`[t,y]=ode23('fonksiyon_adi',[t0 tson],y0,secenekler,p1,p2);`

❖ fonksiyon\_adi: diferansiyel denklemin tanımlı olduğu harici fonksiyon dosyasıdır. Adı tırnak içerisinde yazılmalıdır. bu dosya fonksiyon tanımlama kurallarına uygun olarak hazırlanmış olmalıdır. Tırnak içerisine fonksiyonun ve dolayısıyla dosyanın adı yazılır. (.m) dosya uzantısının yazılmaması gerektiği unutulmamalıdır.

❖ Burada t değişkenine t0 dan başlayarak tson'a kadar iterasyonda kullanılan t değerleri atanır.

❖ y değişkeninde ise birinci sütuna t'ye (bağımsız değişkene) karşı y değerleri (bağımlı değişken değerleri) sırasıyla diğer sütunlara ise 1.,2., ... n. dereceden türevlerinin aldığı değerler atanır.

## Steps for solving a single first-order ODE:

For the remainder of this section the independent variable is taken as  $t$  (time). This is done because in many applications time is the independent variable, and also to be consistent with the information in the **Help** menu of MATLAB.

### Step 1: Write the problem in a standard form.

Write the equation in the form:

$$\frac{dy}{dt} = f(t, y) \quad \text{for } t_0 \leq t \leq t_f, \quad \text{with } y = y_0 \text{ at } t = t_0.$$

As shown above, three pieces of information are needed for solving a first order **ODE**: An equation that gives an expression for the derivative of  $y$  with respect to  $t$ , the interval of the independent variable, and the initial value of  $y$ . The solution is the value of  $y$  as a function of  $t$  between  $t_0$  and  $t_f$ .

An example of a problem to solve is:

$$\frac{dy}{dt} = \frac{t^3 - 2y}{t} \quad \text{for } 1 \leq t \leq 3 \quad \text{with } y = 4.2 \quad \text{at } t = 1.$$

**Step 2: Create a user-defined function (in a function file) or an anonymous function.**

The ODE to be solved has to be written as a user-defined function (in a function file) or as an anonymous function. Both calculate  $\frac{dy}{dt}$  for given values of  $t$  and  $y$ . For the example problem above, the user-defined function (which is saved as a separate file) is:

```
function dydt=ODEexpl(t,y)
dydt=(t^3-2*y)/t;
```

**Step 3: Select a method of solution.**

Select the numerical method that you would like MATLAB to use in the solution. Many numerical methods have been developed to solve first-order ODEs, and several of the methods are available as built-in functions in MATLAB. In a typical numerical method, the time interval is divided into small time steps. The solution starts at the known point  $y_0$ , and then by using one of the integration methods the value of  $y$  is calculated at each time step. Table 9-1 lists seven ODE solver commands, which are MATLAB built-in functions that can be used for solving a first-order ODE. A short description of each solver is included in the table.

**Table 9-1: MATLAB ODE Solvers**

<b>ODE Solver Name</b>	<b>Description</b>
ode45	For nonstiff problems, one-step solver, best to apply as a first try for most problems. Based on explicit Runge-Kutta method.
ode23	For nonstiff problems, one-step solver. Based on explicit Runge-Kutta method. Often quicker but less accurate than ode45.
ode113	For nonstiff problems, multistep solver.

**Table 9-1: MATLAB ODE Solvers (Continued)**

<b>ODE Solver Name</b>	<b>Description</b>
<code>ode15s</code>	For stiff problems, multistep solver. Use if <code>ode45</code> failed. Uses a variable order method.
<code>ode23s</code>	For stiff problems, one-step solver. Can solve some problems that <code>ode15s</code> cannot.
<code>ode23t</code>	For moderately stiff problems.
<code>ode23tb</code>	For stiff problems. Often more efficient than <code>ode15s</code> .

In general, the solvers can be divided into two groups according to their ability to solve stiff problems and according to whether they use on-step or multi-step methods. Stiff problems are ones that include fast and slowly changing components and require small time steps in their solution. One-step solvers use information from one point to obtain a solution at the next point. Multistep solvers use information from several previous points to find the solution at the next point. The details of the different methods are beyond the scope of this book.

It is impossible to know ahead of time which solver is the most appropriate for a specific problem. A suggestion is to first try `ode45`, which gives good results for many problems. If a solution is not obtained because the problem is stiff, trying the solver `ode15s` is suggested.

#### **Step 4: Solve the ODE.**

The form of the command that is used to solve an initial value ODE problem is the same for all the solvers and for all the equations that are solved. The form is:

```
[t, y] = solver_name (ODEfun, tspan, y0)
```

#### **Additional information:**

<code>solver_name</code>	Is the name of the solver (numerical method) that is used (e.g. <code>ode45</code> or <code>ode23s</code> )
<code>ODEfun</code>	The function from Step 2 that calculates $\frac{dy}{dt}$ for given values of $t$ and $y$ . If it was written as a user-defined function, the function handle is entered. If it was written as an anonymous function, the name of the anonymous function is entered. (See the example that follows.)
<code>tspan</code>	A vector that specifies the interval of the solution. The vector must have at least two elements but can have more. If the vector has only two elements, the elements must be <code>[t0 tf]</code> , which are the initial and final points of the solution interval. The



tspan

Çözüm aralığını belirten bir vektördür. Vektör en az iki elemana sahip olmalıdır, ancak daha fazlasına da sahip olabilir. Vektörün sadece iki elemanı varsa, elemanlar çözüm aralığının ilk ve son noktaları olan  $[t_0 \ t_f]$  olmalıdır. tspan, ancak, ilk ve son noktalar arasında ek noktalara sahip olabilir.

y0

y'nin başlangıç değeridir (aralığın ilk noktasında y değeri).

[t,y]

ODE'nin çözümü olan çıktıdır. t ve y sütun vektörleridir. İlk ve son noktalar, aralığın başlangıç ve bitiş noktalarıdır. Aradaki boşluk ve nokta sayısı, tspan giriş vektörüne bağlıdır. Eğer tspan'ın iki unsuru varsa (başlangıç ve bitiş noktaları), t ve y vektörleri çözücü tarafından hesaplanan her bir entegrasyon adımında çözümü içerir. Eğer tspan ikiden fazla noktaya sahipse (ilk ve sonuncu arasında ek noktalar), t ve y vektörleri sadece bu noktalarda çözüm içerir. Tspan'daki nokta sayısı, program tarafından çözüm için kullanılan zaman adımlarını etkilemez

For example, consider the solution to the problem stated in Step 1:

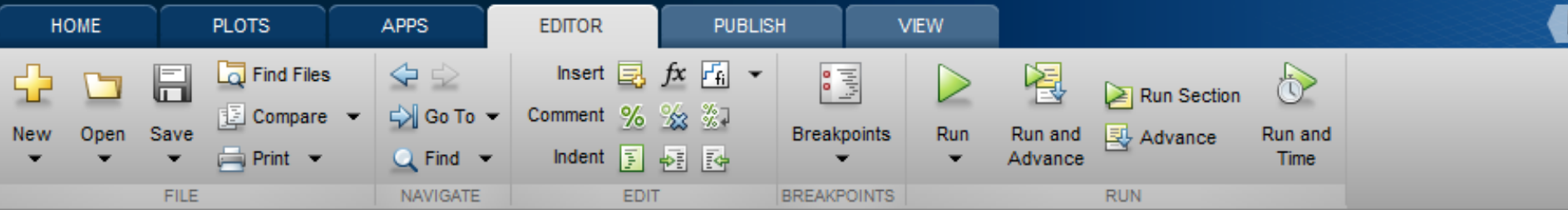
$$\frac{dy}{dt} = \frac{t^3 - 2y}{t} \quad \text{for } 1 \leq t \leq 3 \quad \text{with } y = 4.2 \text{ at } t = 1,$$

**Step 2: Create a user-defined function (in a function file) or an anonymous function.**

The ODE to be solved has to be written as a user-defined function (in a function file) or as an anonymous function. Both calculate  $\frac{dy}{dt}$  for given values of  $t$  and  $y$ .

For the example problem above, the user-defined function (which is saved as a separate file) is:

```
function dydt=ODEexp1(t,y)
dydt=(t^3-2*y)/t;
```



C:\Users\ilknur\Desktop\Egitim\DERSLER\MATLAB\Matlab2018

```
Editor - C:\Users\ilknur\Desktop\Egitim\DERSLER\MATLAB\Matlab2018\ODEEx...
ODEexp1.m
1 function dydt=ODEexp1(t,y)
2   dydt=(t^3-2*y)/t;
3
```

```
Command Window
>> [t y]=ode45(@ODEexp1,[1:0.5:3],4.2)

t =

    1.0000
    1.5000
    2.0000
    2.5000
    3.0000

y =

    4.2000
    2.4528
    2.6000
    3.7650
    5.8444

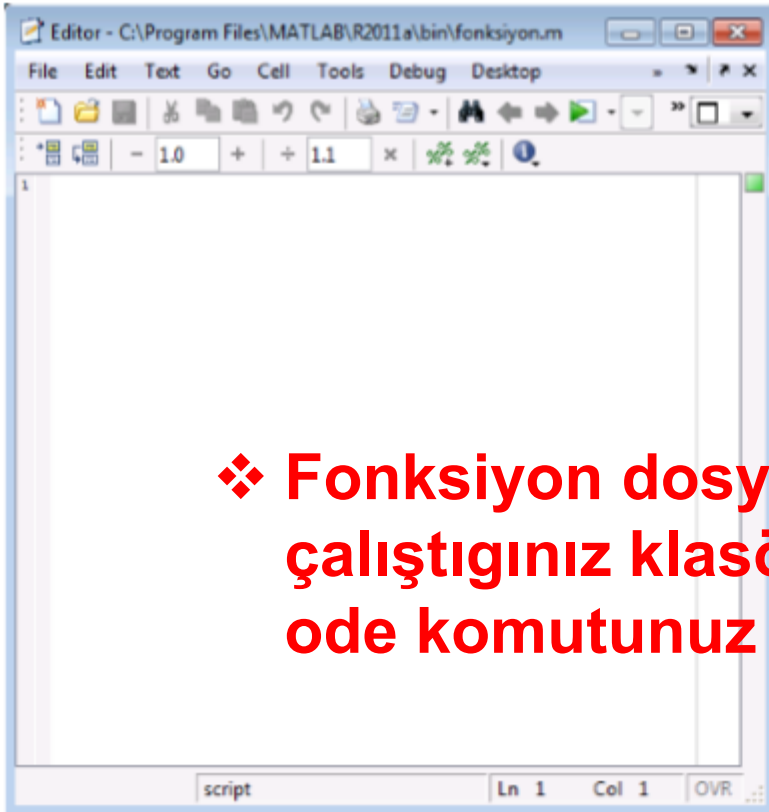
fx >>
```

❖ Fonksiyon dosyanızın bulunduğu klasör ve çalıştığınız klasör aynı olmalıdır aksi takdirde ode komutunuz çalışmayacaktır

Name	Value
t	[1;1.5000;2;2.5000;3]
y	[4.2000;2.4528;2.6000;...

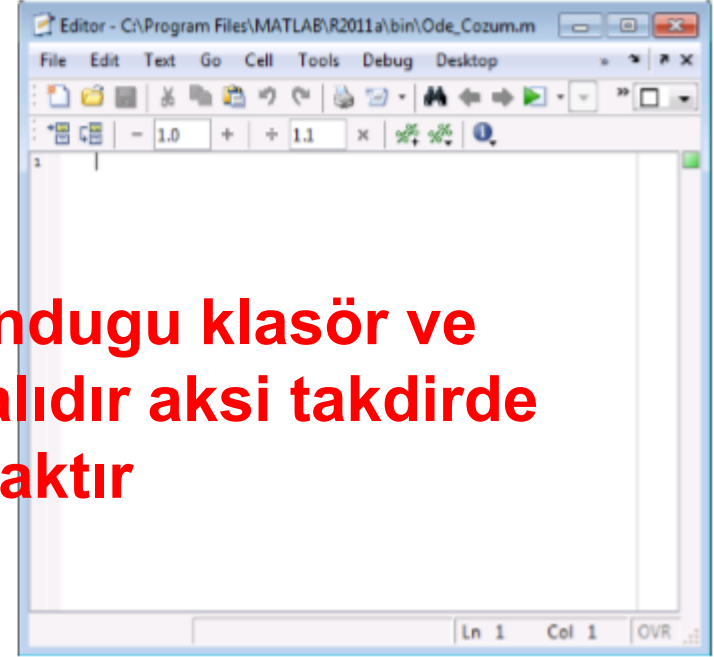
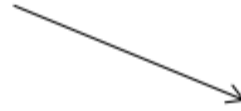
>> edit fonksiyon

Command window'a editör penceresinde girilecek fonksiyonun adı yazılmalıdır ve kaydedilirken .m dosyası olarak kaydedilir.



Mesela;

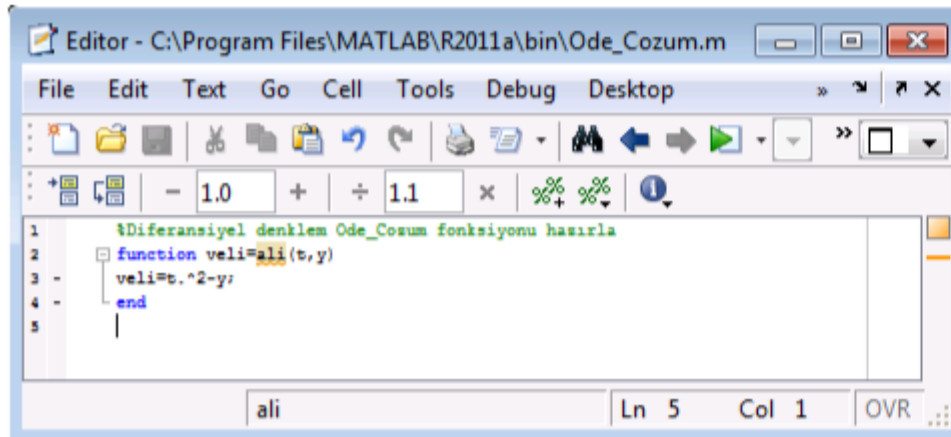
>> edit Ode\_Cozum  
olsun...



❖ **Fonksiyon dosyanızın bulunduğu klasör ve çalıştığınız klasör aynı olmalıdır aksi takdirde ode komutunuz çalışmayacaktır**

❖ Fonksiyon tanımlaması:  $y' = \frac{dy}{dt} = t^2 - y$  şeklindeki adi diferansiyel denklemi editör penceresinde tanımlayalım;

```
%Diferansiyel denklem Ode_Cozum fonksiyonu hazırla  
function veli=ali(t,y)  
veli=t.^2-y;  
end
```



The screenshot shows the MATLAB Editor window for the file 'Ode\_Cozum.m'. The code is as follows:

```
1 %Diferansiyel denklem Ode_Cozum fonksiyonu hazırla  
2 function veli=ali(t,y)  
3 -   veli=t.^2-y;  
4 -   end  
5 |
```

The status bar at the bottom indicates the cursor is at line 5, column 1.

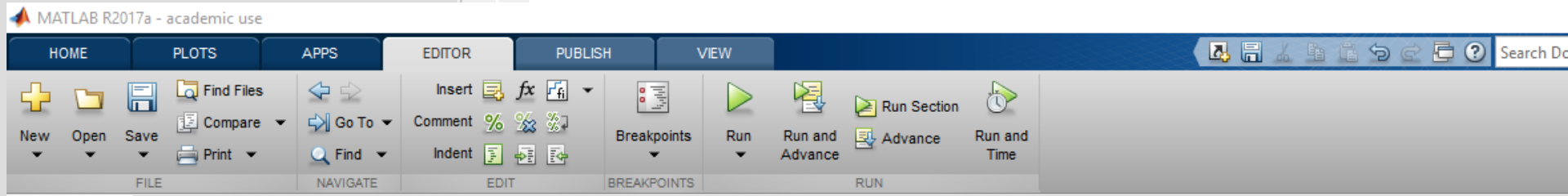
Ode\_Cozum isimli fonksiyon, otomatik olarak kullanılmak üzere, save sekmesine tıklandığında kaydedilmiş olur.

$$y' = \frac{dy}{dt} = t^2 - y$$

şeklindeki adi diferansiyel denklemin çözümünü,  $y(0)=1$  başlangıç koşuluyla,  $t(0)=0$ 'dan  $t_{son}=3$ 'e kadarki değerleri için yapalım;

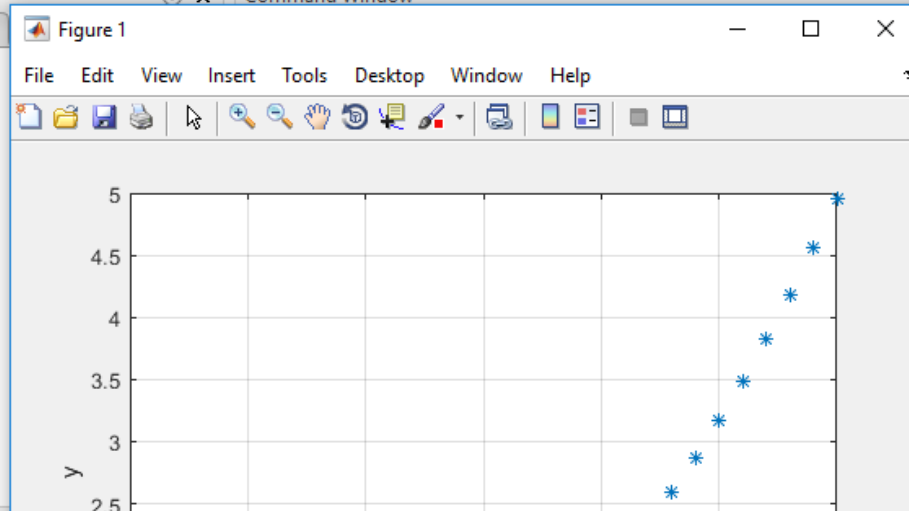
Önce fonksiyon dosyası oluşturulur

```
Editor - C:\Users\ilknur\Desktop\Egitim\DERSLER\MATLAB\Matlab2018\ornekc...  
ODEexp1.m x ornekcozum.m x Untitled2* x +  
1 function A=ornekcozum(t,y)  
2 - A=t^2-y;
```



C:\Users\ilknur\Desktop\Egitim\DERSLER\MATLAB\Matlab2018

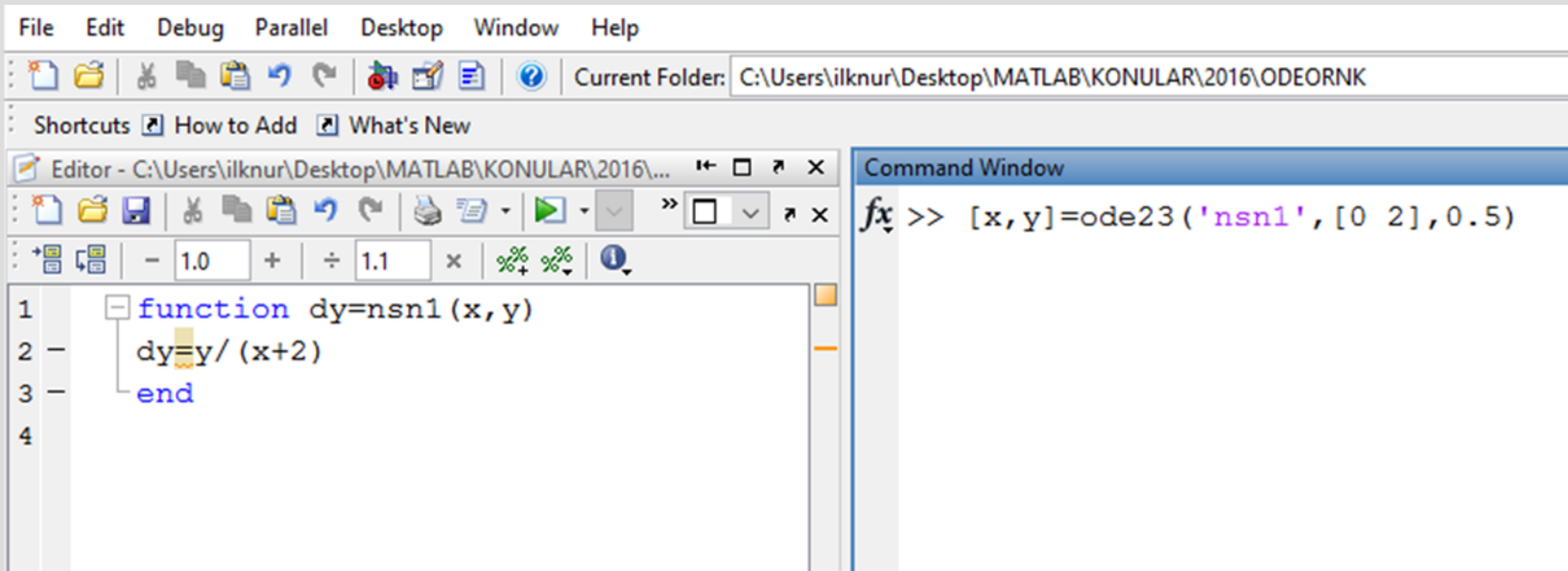
```
Editor - Untitled2*  
ODEexp1.m x ornekcozum.m x Untitled2* x +  
1 tspan=[0:0.1:3];  
2 y0=1;  
3 [t y]=ode45(@ornekcozum,tspan,y0)  
4 plot(t,y,'*')  
5 xlabel('t')  
6 ylabel('y')  
7 grid on
```





# ODE23 ve ODE45 ILE ORNEK COZUMLER

$$y' = \frac{y}{x+2} \quad y(0)=0.5$$
$$y[0 \ 2]=?$$



The image shows a MATLAB interface with two windows. The top window is the Editor, and the bottom window is the Command Window.

**Editor Window:** The file path is `C:\Users\ilknur\Desktop\MATLAB\KONULAR\2016\...`. The code in the editor is:

```
1 function dy=nsn1(x,y)
2     dy=y/(x+2)
3 end
4
```

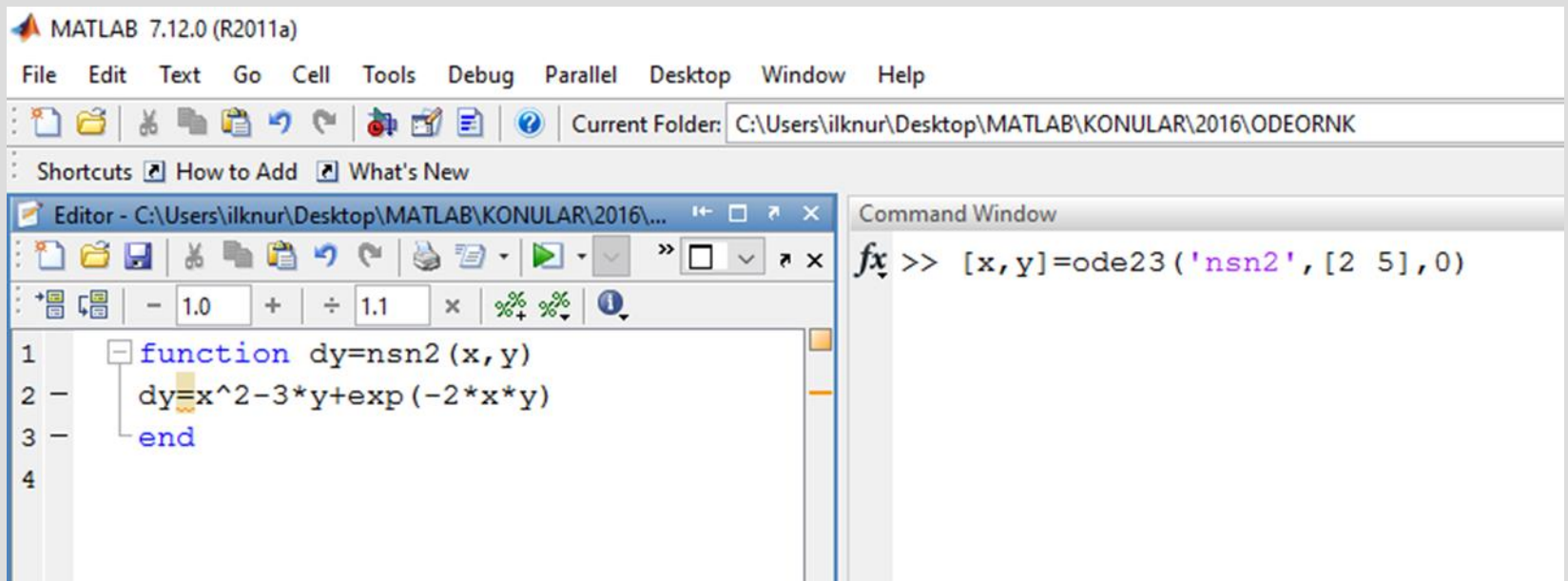
**Command Window:** The command entered is:

```
>> [x,y]=ode23('nsn1',[0 2],0.5)
```



$$y' = x^2 - 3y + e^{-2xy}$$

$$y(0)=1$$
$$y[2,5]=?$$



$$y' = \frac{-ty}{\sqrt{2-y^2}}$$

$$y(0)=1$$
$$t_0=0 \quad t_s=5$$

ode23 ve ode45 cozumunu  
karsilastiriniz (grafik ile)

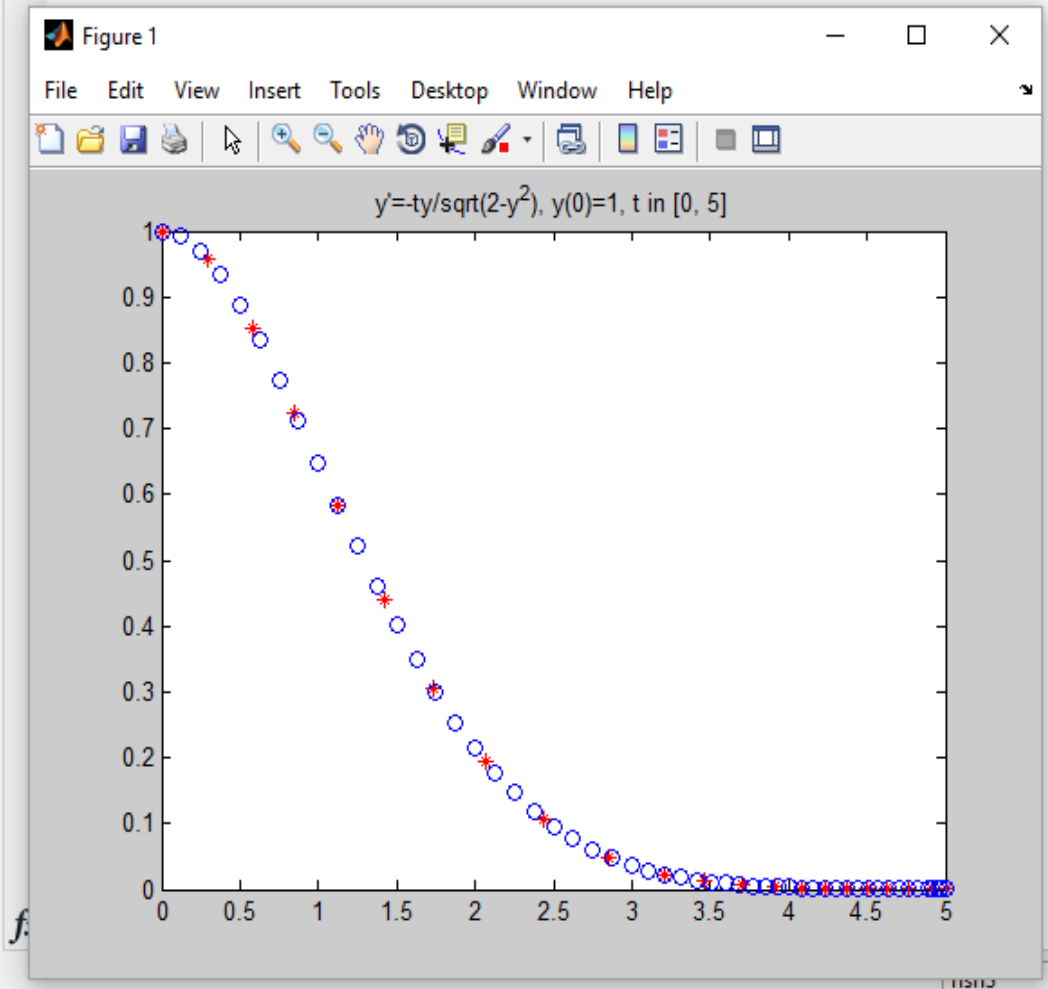
Shortcuts How to Add What's New

```
Editor - C:\Users\ilknur\Desktop\MATLAB\KONULAR\2016\ODEOR...  
1 function dy=nsn3(t,y);  
2     dy=(-t*y)/sqrt(2-y^2);  
3     end  
4
```

Current Folder: << MATLAB >> KONULAR >> 2016 >> ODEORNK

- Name ▲
- akim.m
- fun1.m
- nsn1.asv
- nsn1.m\*

```
Command Window  
>> [t,y]=ode23('nsn3',[0 5],1);  
[a,b]=ode45('nsn3',[0 5],1);  
plot(t,y,'r*',a,b,'bo')  
title('y'=-ty/sqrt(2-y^2), y(0)=1, t in [0, 5])
```



$$y_1' = 2y_1 + y_2 + 5y_3 + e^{-2t}$$

$$y_2' = -3y_1 - 2y_2 - 8y_3 + 2e^{-2t} - \cos(3t)$$

$$y_3' = 3y_1 + 3y_2 + 2y_3 + \cos(3t)$$

$$y_1(0) = 1, y_2(0) = -1, y_3(0) = 0 \quad t \in [0, \pi/2]$$

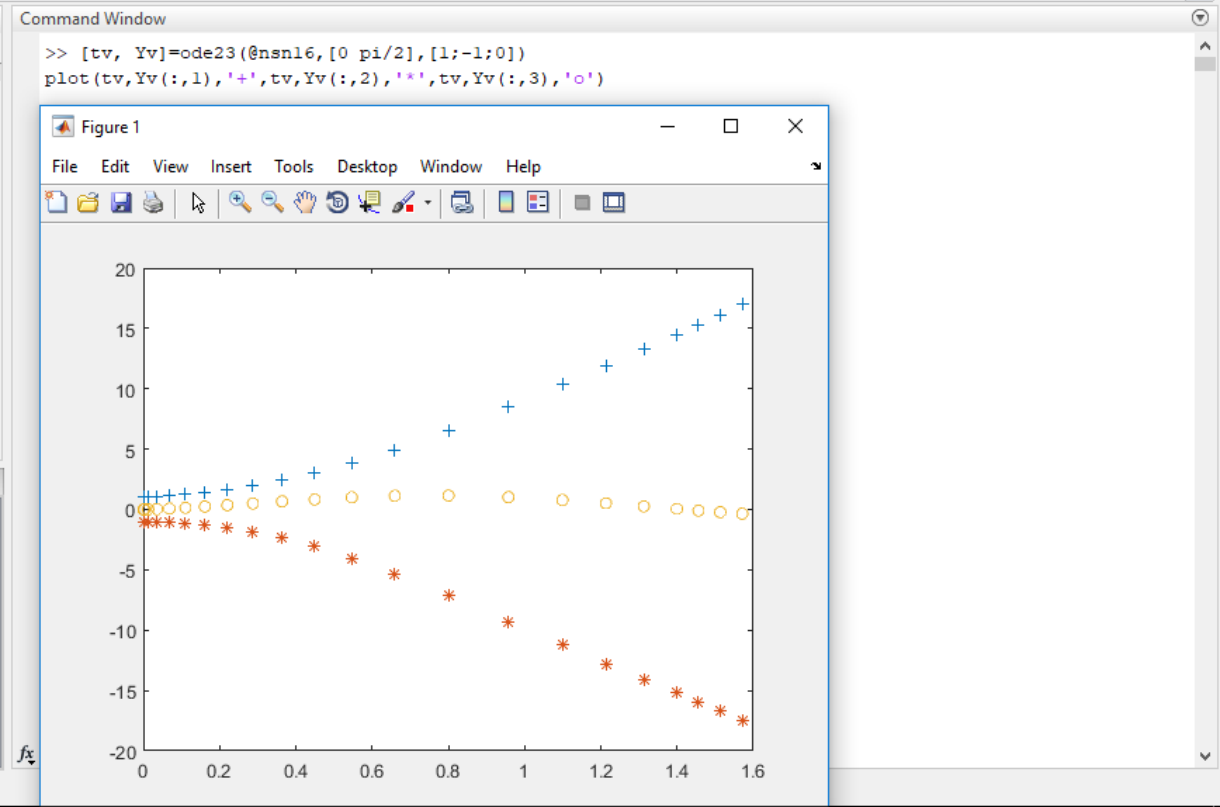
ode23 veya ode45 kullanarak cozunuz ve grafik olarak gosteriniz

```

Editor - C:\Users\ilknur\Desktop\Egitim\DERSLER\MATLAB\Matlab2018\
ODEExp1.m x nsn16.m x
1 function Fv=nsn16(t,Y)
2     Fv(1,1)=2*Y(1)+Y(2)+5*Y(3)+exp(-2*t)
3     Fv(2,1)=-3*Y(1)-2*Y(2)-8*Y(3)+2*exp(-2*t)-cos(3*t)
4     Fv(3,1)=3*Y(1)+3*Y(2)+2*Y(3)+cos(3*t)
5 end
6
7
    
```

Workspace

Name	Value
t	[1:1.5000;2:2.5000;3]
tv	24x1 double
y	[4.2000;2.4528;2.6000;...]
Yv	24x3 double



Van der Pol (  $\ddot{x} + (x^2 - 1)\dot{x} + x = 0$  ) Asagidaki sekilde yazilabilir. Bu denklemin cozumunu matlab ile yapiniz

$$\begin{aligned}\dot{x}_1 &= x_1(1 - x_2^2) - x_2 & \mathbf{x} & [0 \quad 0.25] \\ \dot{x}_2 &= x_1 & 0 < t < 20\end{aligned}$$

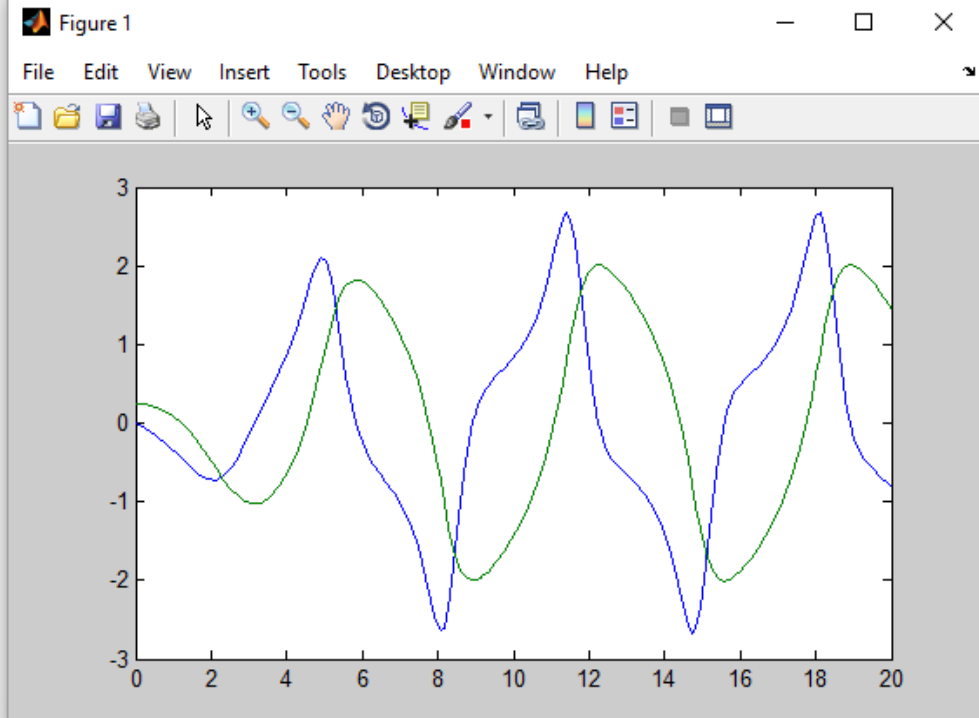


```
1 function noktac=nsn5(t,x)
2     noktac=[x(1)*(1-x(2)^2)-x(2);x(1)]
3 end
4
5
6
```

Name

- akim.m
- fun1.m
- nsn1.asv
- nsn1.m\*

```
fx >> t0 = 0; tf = 20;
      x0 = [0 0.25]; % Initial conditions
      [t,x] = ode23('nsn5',t0,tf,x0);
      plot(t,x)
```



File menu: New, Open, Save, Find Files, Compare, Print, Go To, Find, Insert, Comment, Indent

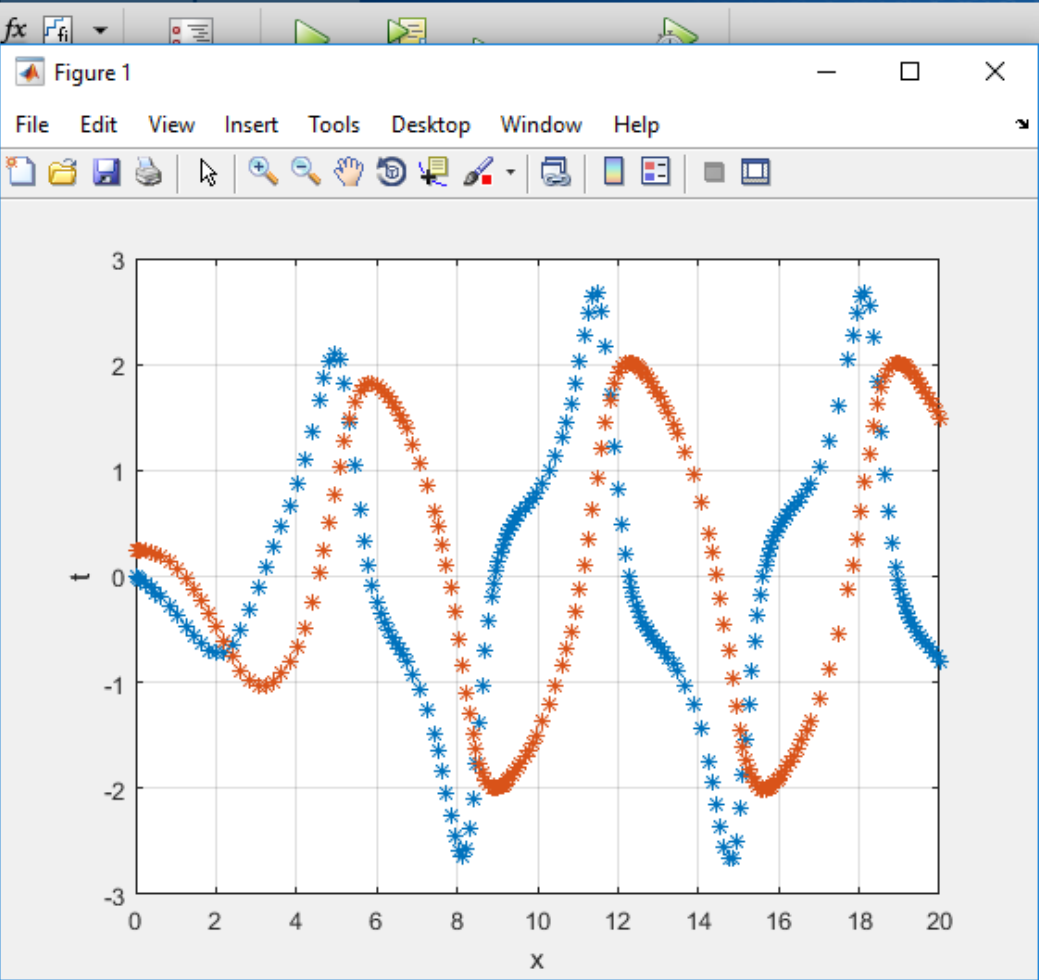
C:\Users\ilknur\Desktop\Egitim

Editor - Untitled2\*  
ODEexp1.m ornekcozum.m Untitled2\* nsn5.n

```
1 tspan=[0 20]
2 x0=[0 0.25]
3 [t x]=ode45(@nsn5,tspan,x0)
4 plot(t,x,'*')
5 xlabel('x')
6 ylabel('t')
7 grid on
```

Workspace

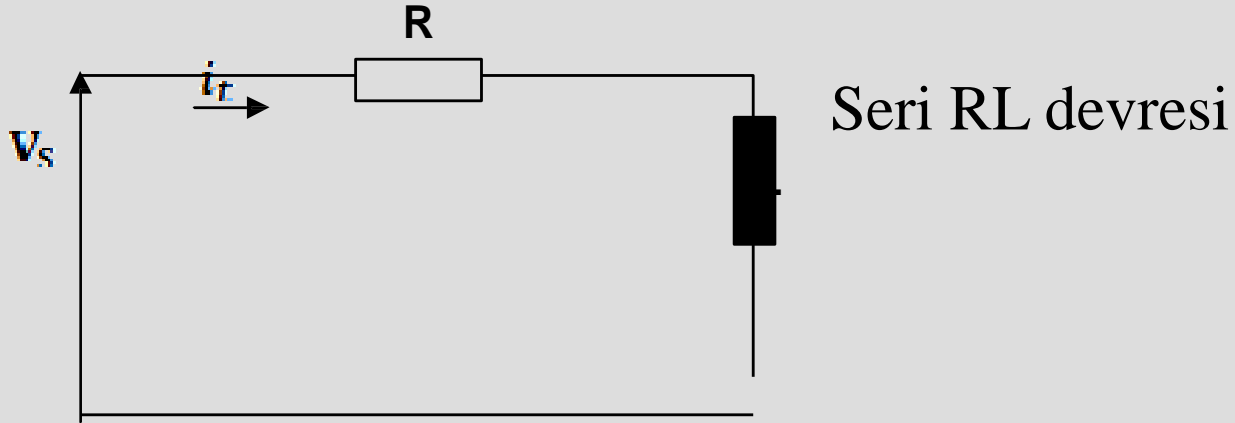
Name	Value
cozum	1x1 struct
t	201x1 double
tspan	[0,20]
x	201x2 double
x0	[0,0.2500]
y	31x1 double
y0	1



0.7854	2.7854
-0.6301	1.7403
-0.6703	1.6831
-0.7105	1.6224
-0.7520	1.5581
-0.7961	1.4900



**a-Alternatif Akımda:**



$$V_s = 100 \sin \omega t$$

$$R = 0.4 \Omega$$

$$L = 10 \text{ mH}$$

$$f = 50 \text{ Hz}$$

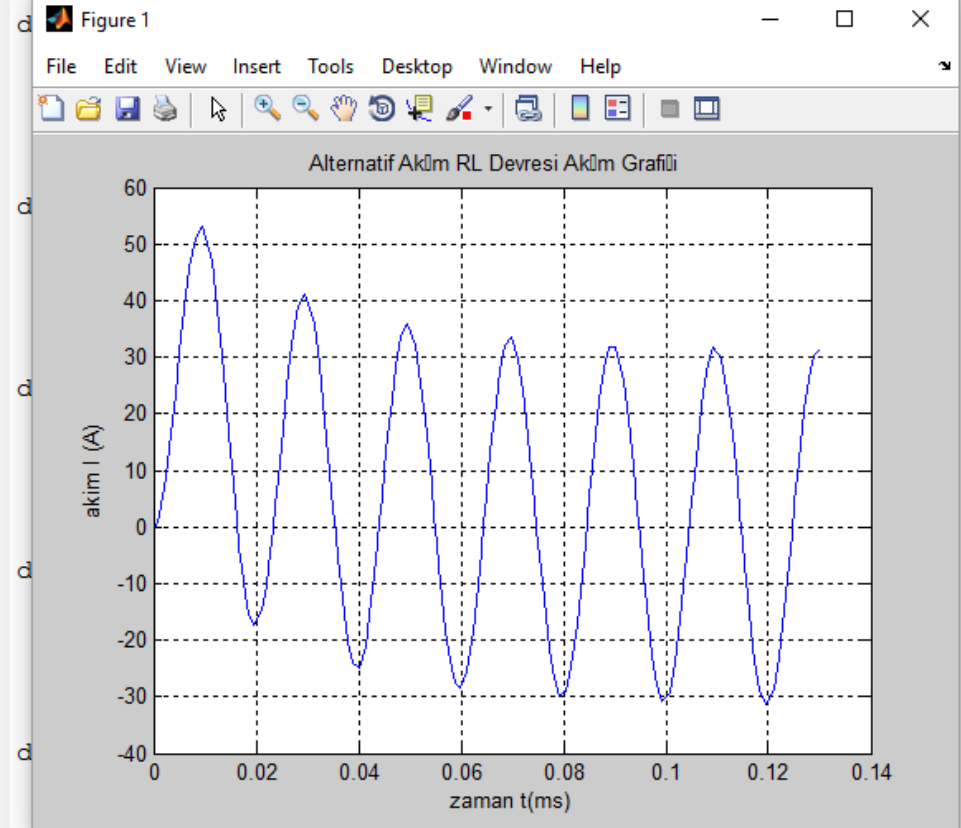
$$i_t = ?$$

$$V_s = R * i_t + L \frac{di}{dt}$$

$$\omega = 2\pi f$$

```
1 function dy_dt=akim(t,i)
2     dy_dt=((100*sin(314*t)-i.*0.4)/0.01)
3 end
4
```

```
>> [a b]=ode45('akim',[0,0.13],0)
```



Dogru akimda ise

$$V_s = 100$$

$$R = 0.4 \Omega$$

$$L = 10 \text{ mH}$$

$$f = 50 \text{ Hz}$$

$$i_t = ?$$

$$V_s = R * i_t + L \frac{di}{dt}$$

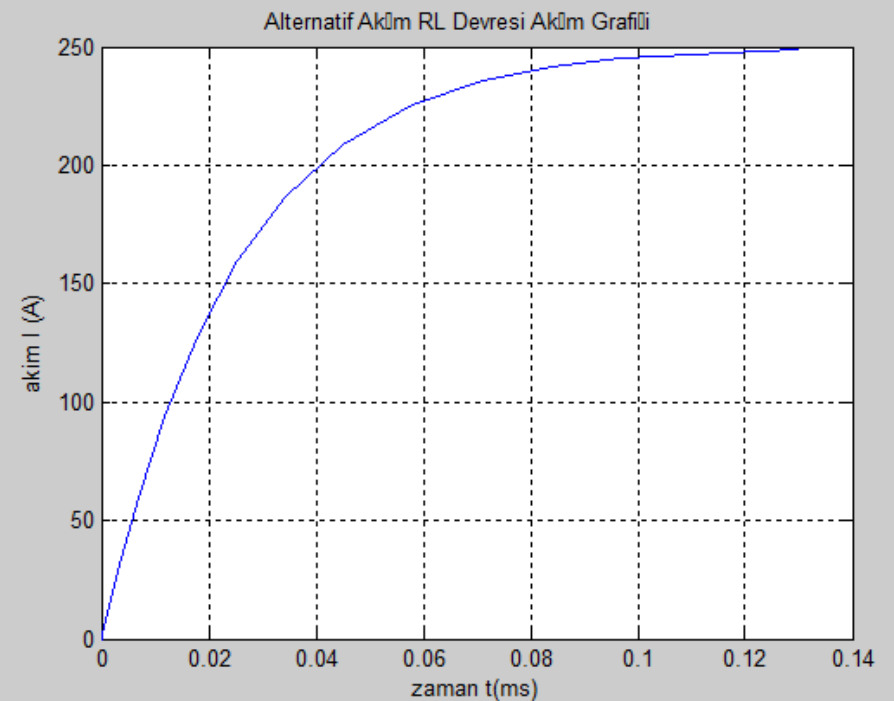
$$\omega = 2\pi f$$

```
1 function dy=dogrua(t,i)
2     dy=((100-i.*0.4)/0.01)
3 end
4
```

Name

- akim.asv
- akim.m\*
- dogrua.m
- dogruakim.m

```
>> [a b]=ode23('dogrua',0,0.130,0)
plot(a,b)
grid
xlabel('zaman t(ms)')
```



Örnek:  $x' = \frac{dx(t)}{dt} = -15x(t) + 2 \cos(2t) + t$

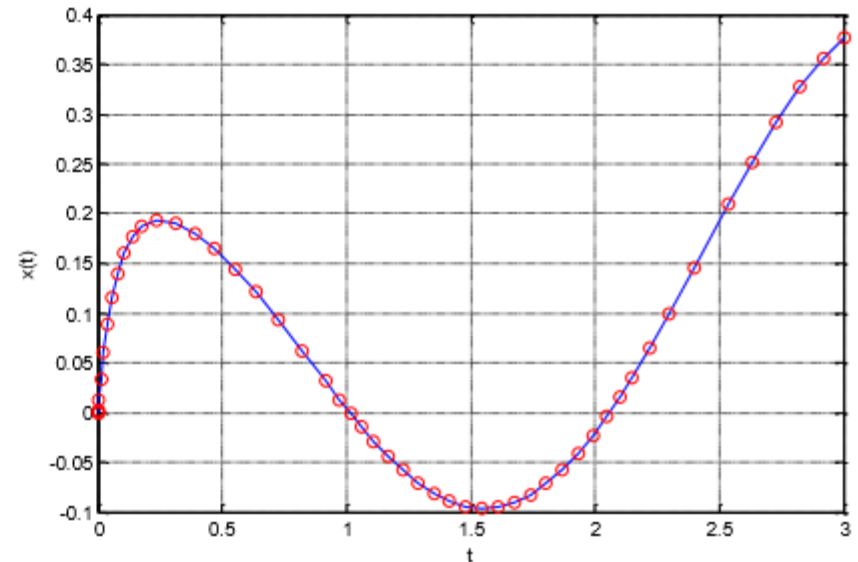
```
% Diferansiyel denklem (Ornek.m)  
function dx_dt=difdenk(t,x)  
dx_dt=-15*x+3*cos(2*t)+t;  
end
```

Fonksiyonun tanımlanması

şeklindeki adi diferansiyel denklemin  
çözümünü,  $x(0)=0$  başlangıç  
koşuluyla,  $t(0)=0$ 'dan  $t_{son}=3$ 'e  
kadarki değerleri için yapalım;

Çözüm:

```
>> %Program dosyası (Ornek.m)  
>> t0=0;  
>> tson=3;  
>> x0=0;  
>> [t,x]=ode23('Ornek',[t0 tson],x0);  
>> plot(t,x); xlabel('t'),ylabel('x(t)');grid on;
```



## ode23 veya ode45 komutlarıyla Diferansiyel Denklem Çözümü

Örnek:  $\frac{dx}{dt} = 2 \cdot x - x^2 - x \cdot y$

$$\frac{dy}{dt} = x \cdot y - y$$

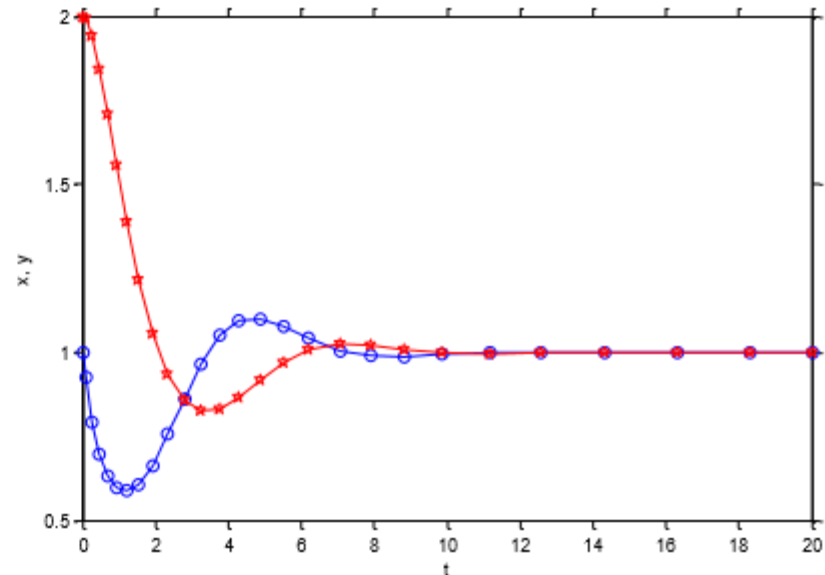
diferansiyel denklemini  $x(0)=1$  ve  $y(0)=2$  başlangıç koşullarına göre çözelim ve grafiğini çizelim.

Çözüm için ilk önce  $x$ 'i  $y(1)$ ,  $y$ 'yi  $y(2)$  olarak tanımlayıp, iki denklemin eş zamanlı çözümü için **inline** komutu ile tek bir fonksiyon altında toplanması gerekir;

```
>> f=inline('[2*y(1)-y(1)^2-y(1)*y(2);y(1)*y(2)-y(2)]','t','y');  
>> [t,y]=ode23(f,[0,20],[1;2])  
>> plot(t,y)
```

veya

```
>> f=inline('[2*y(1)-y(1)^2-y(1)*y(2);y(1)*y(2)-y(2)]','t','y');  
>> [t,y]=ode45(f,[0,20],[1;2])  
>> plot(t,y)
```



HOME PLOTS APPS EDITOR PUBLISH VIEW

File: New, Open, Save, Compare, Print  
 Navigate: Go To, Find  
 Editor: Insert, Comment, Indent  
 Breakpoints: Breakpoints  
 Run: Run, Run and Advance, Run Section, Advance, Run and Time

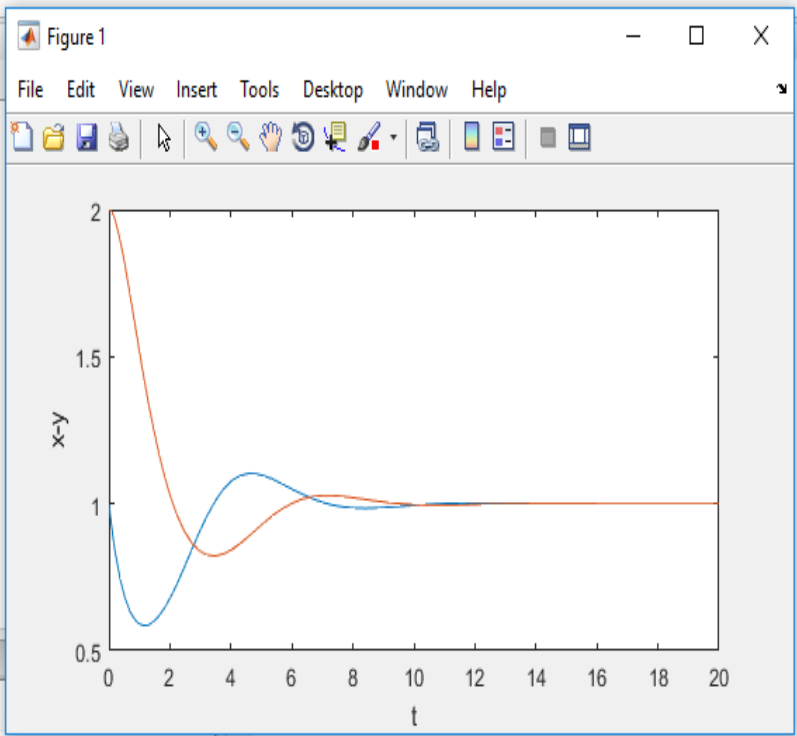
C:\Users\ilknur\Desktop\Egitim\DERSLER\MATLAB\Matlab2018

```

Editor - Untitled4*
ODEExp1.m x ornekcozum.m x Untitled2* x nsn5.m x Untitled4* x +
1 bagla=inline('2*y(1)-y(1)^2-y(1)*y(2);y(1)*y(2)-y(2)'),'t','y')
2 [t y]=ode45(bagla,[0 20],[1;2])
3 plot(t,y)
4 xlabel('t')
5 ylabel('x-y')
    
```

Workspace

Name	Value
bagla	1x1 inline
t	65x1 double
y	65x2 double



1.0000	0.9999
1.0000	0.9999
1.0000	0.9999
1.0000	1.0000
1.0000	1.0000
1.0000	1.0000

fx >>

Örnek:

$$-r_A = -\frac{dC_A(t)}{dt} = kC_A^2$$

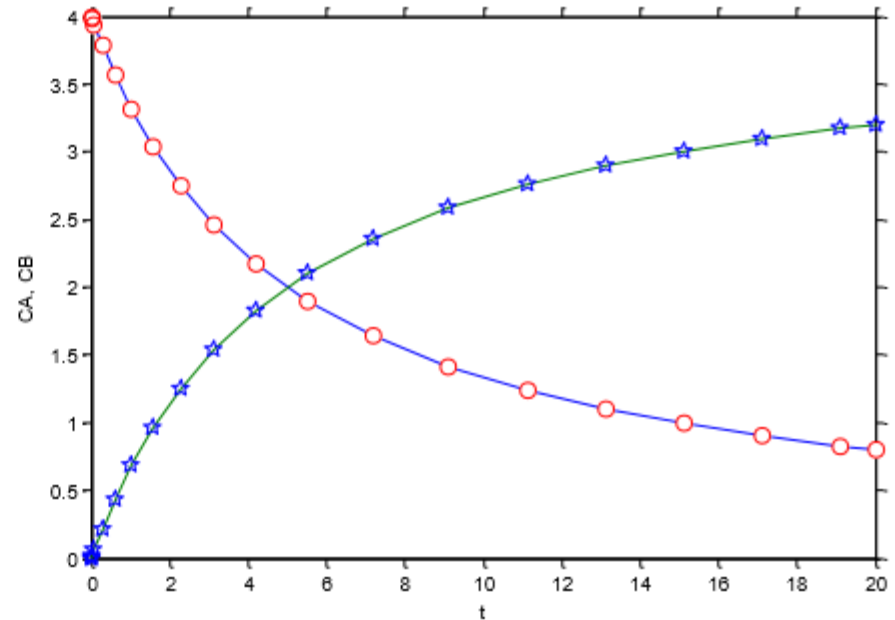
$$r_B = \frac{dC_B(t)}{dt} = k \cdot C_A^2$$

şeklindeki 2. dereceden bir reaksiyon hız ifadesini,  $C_A(0)=4$  mol/L ve  $C_B=0$  mol/L başlangıç koşuluyla,  $t(0)=0$ 'dan  $t_{son}=20$  dak'ya kadarki değişimi için çözelim;  $k=0.05$

Çözüm için ilk önce  $C_A$ 'yı  $y(1)$  ve  $C_B$ 'yi  $y(2)$  olarak tanımlayıp, bir fonksiyonda tanımlayalım;

$$\frac{dy(1)}{dt} = -0.05 \cdot y(1)^2 \quad \frac{dy(2)}{dt} = 0.05 \cdot y(1)^2$$

```
>> f=inline('[-0.05*y(1)^2;0.05*y(1)^2'],'t','y');  
>> [t,y]=ode23(f,[0,20],[4;0]);  
>> plot(t,y)
```







**Örnek:**  $A \xrightarrow{k_1} R \xrightarrow{k_2} S$  şeklindeki seri reaksiyonda,  $C_A(0)=4$  mol/L ve  $C_R=C_S=0$  mol/L başlangıç koşuluyla, her bir komponent için  $t(0)=0$ 'dan  $t_{son}=200$  dak'ya kadarki değişimi için çözelim;  $k_1=0.05$ ,  $k_2=0.012$

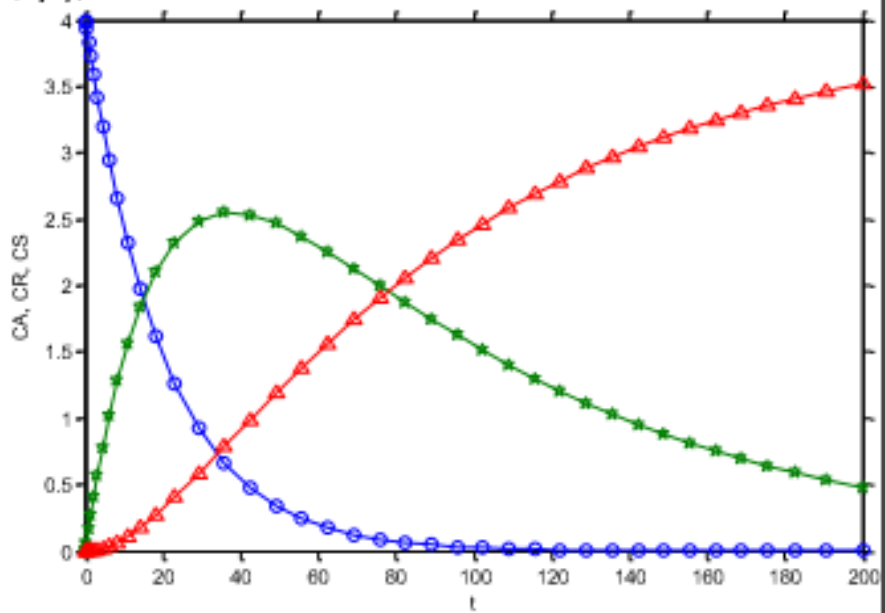
$$-\frac{dC_A}{dt} = k_1 C_A$$

$$\frac{dC_R}{dt} = k_1 C_A - k_2 C_R$$

$$\frac{dC_S}{dt} = k_2 C_R$$

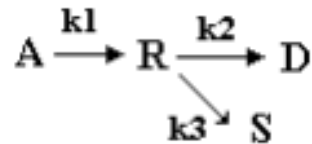
Çözüm için ilk önce  $C_A$ 'yi  $y(1)$ ,  $C_R$ 'yi  $y(2)$  ve  $C_S$ 'yi  $y(3)$  olarak tanımlayıp, bir fonksiyonda tanımlayalım;

```
>> f=inline('[-0.05*y(1);0.05*y(1)-0.012*y(2);0.012*y(2)]','t','y');
>> [t,y]=ode23(f,[0,200],[4;0;0]);
>> plot(t,y)
```



Yukarıdaki diferansiyel denklemleri başlangıç koşullarına göre 15'nci dakikaya kadar 30 adımda çözdürün ve grafiğini çizin.

6.



Seklindeki reaksiyonda her bir komponente ait hız ifadeleri aşağıda verilmiştir. 0'dan 80'nci dakikaya kadar, her komponentin konsantrasyon değişimini bulunuz ve grafik olarak gösteriniz. Başlangıçta CA konsantrasyonu 2 mol/l'tir, CR ve CS yoktur. Sırasıyla  $k_1=0.1$  ;  $k_2=0.05$  ve  $k_3=0.03$  1/dk.'dir.

$$\frac{dCA}{dt} = -k_1 \cdot CA$$

$$\frac{dCD}{dt} = k_2 \cdot CR$$

$$\frac{dCR}{dt} = k_1 \cdot CA - (k_2 + k_3) \cdot CR$$

$$\frac{dCS}{dt} = k_3 \cdot CR$$

```

1 function Kz=mys1(t,Y)
2     Kz(1,1)=-0.1*Y(1)
3     Kz(2,1)=0.1*Y(1)-0.09*Y(2)
4     Kz(3,1)=0.05*Y(2)
5     Kz(4,1)=0.04*Y(2)
6 end

```

7  
8

```

1 function Kz=mys1(t,Y)
2     Kz(1,1)=-0.1*Y(1)
3     Kz(2,1)=0.1*Y(1)-0.09*Y(2)
4     Kz(3,1)=0.05*Y(2)
5     Kz(4,1)=0.04*Y(2)
6 end

```

7  
8

```
>> [tz Yz]=ode23('mys1',[0 80],[2;0;0;0])
```

Kz =

```
-0.2000
```

Kz =

```
-0.2000
```

```
0.2000
```

Yz =

```
2.0000         0         0         0
```

```
1.9999    0.0001    0.0000    0.0000
```

```
1.9995    0.0005    0.0000    0.0000
```

```
1.9975    0.0025    0.0000    0.0000
```

```
1.9876    0.0124    0.0000    0.0000
```

```
1.9413    0.0579    0.0004    0.0003
```

```
1.8956    0.1019    0.0014    0.0011
```

```
1.8454    0.1491    0.0031    0.0025
```

```
1.7810    0.2077    0.0062    0.0050
```

```
1.7014    0.2773    0.0118    0.0094
```

```
1.6058    0.3564    0.0210    0.0168
```

```
1.4941    0.4421    0.0354    0.0283
```

fx

```
1.3668    0.5303    0.0571    0.0457
```

