

Higher Order Equations

- Must make into a system of first-order equations to use ODE solvers
- Nonlinear is OK!
- Pendulum example:

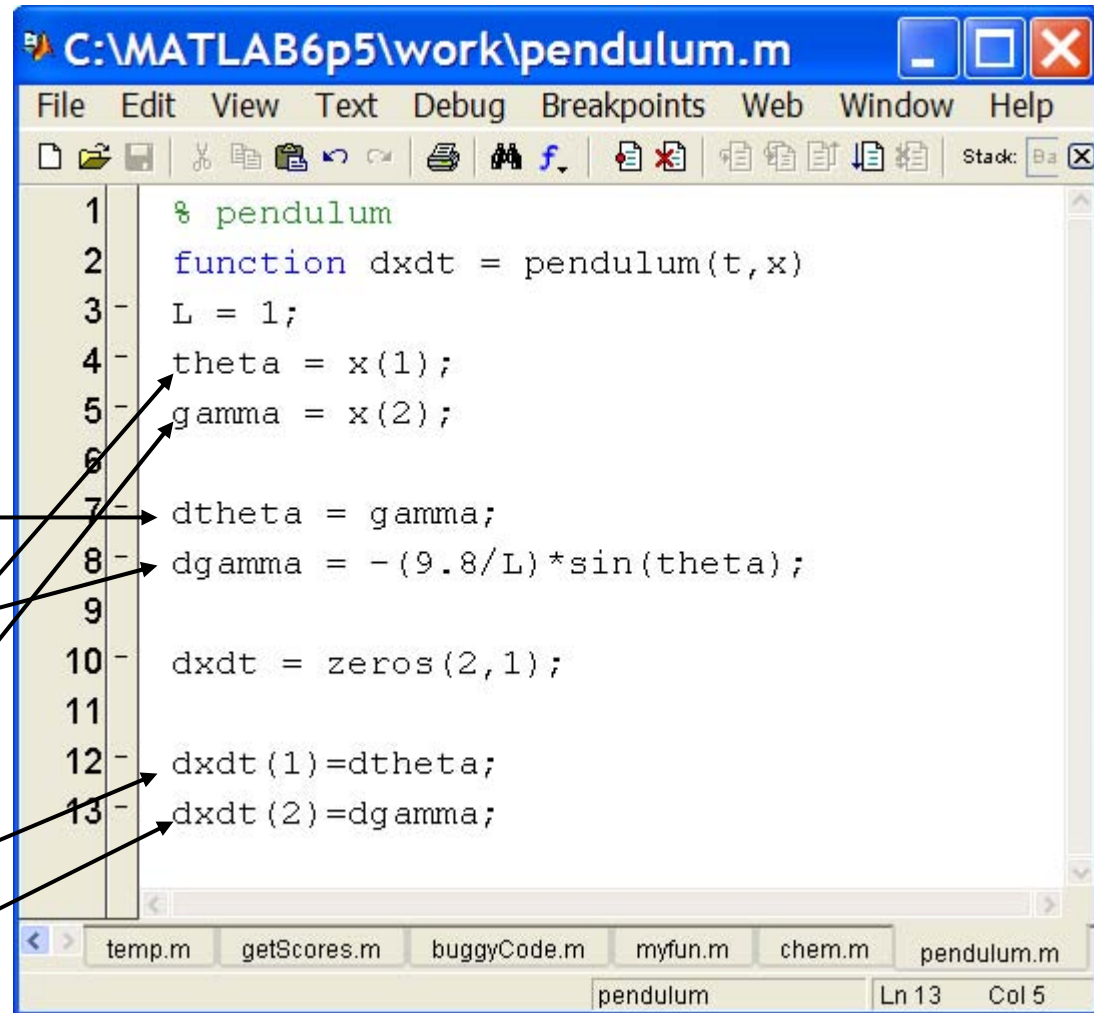
$$\ddot{\theta} + \frac{g}{L} \sin(\theta) = 0$$

$$\ddot{\theta} = -\frac{g}{L} \sin(\theta)$$

$$\text{let } \dot{\theta} = \gamma$$

$$\dot{\gamma} = -\frac{g}{L} \sin(\theta)$$

$$\begin{aligned} \bar{x} &= \begin{bmatrix} \theta \\ \gamma \end{bmatrix} \\ \frac{d\bar{x}}{dt} &= \begin{bmatrix} \dot{\theta} \\ \dot{\gamma} \end{bmatrix} \end{aligned}$$

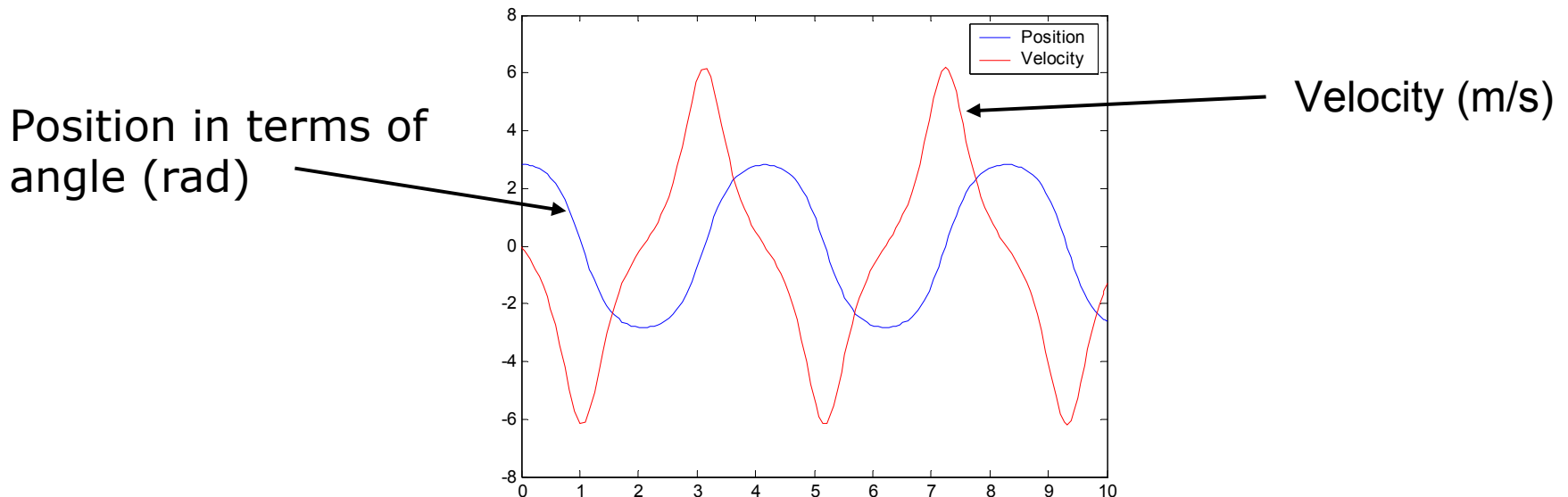


A screenshot of a MATLAB script editor window titled 'C:\MATLAB6p5\work\pendulum.m'. The script defines a function 'pendulum' that takes time 't' and state 'x' as inputs and returns the derivative 'dxdt'. The script sets 'L = 1', extracts 'theta = x(1)' and 'gamma = x(2)', and then calculates 'dtheta = gamma' and 'dgamma = -(9.8/L)*sin(theta)'. Finally, it sets 'dxdt(1) = dtheta' and 'dxdt(2) = dgamma'. The script is shown with line numbers 1 through 13. Arrows from the mathematical equations on the left point to the corresponding lines in the script: from the definition of gamma to line 7, from the equation for dgamma to line 8, and from the state vector definition to lines 4 and 5.

```
1 % pendulum
2 function dxdt = pendulum(t,x)
3 L = 1;
4 theta = x(1);
5 gamma = x(2);
6
7 dtheta = gamma;
8 dgamma = -(9.8/L)*sin(theta);
9
10 dxdt = zeros(2,1);
11
12 dxdt(1)=dtheta;
13 dxdt(2)=dgamma;
```

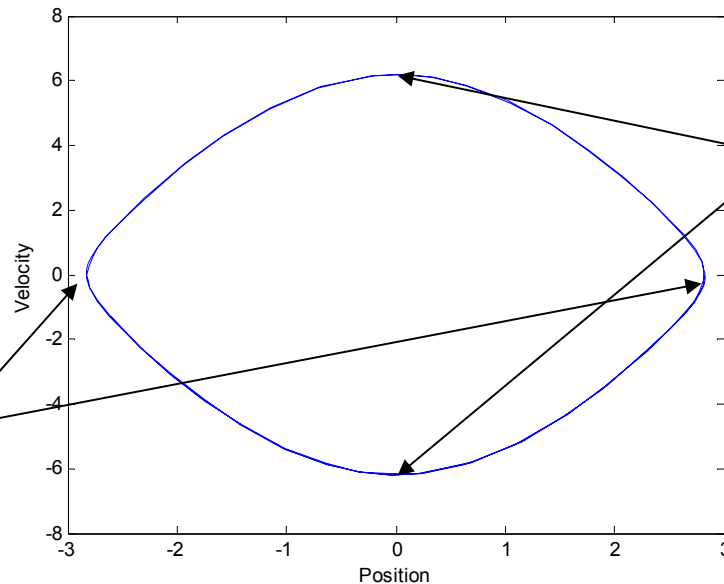
Plotting the Output

- We can solve for the position and velocity of the pendulum:
 - » `[t,x]=ode45('pendulum',[0 10],[0.9*pi 0]);`
 - assume pendulum is almost horizontal
 - » `plot(t,x(:,1));`
 - » `hold on;`
 - » `plot(t,x(:,2),'r');`
 - » `legend('Position','Velocity');`



Plotting the Output

- Or we can plot in the phase plane:
 - » `plot(x(:,1),x(:,2));`
 - » `xlabel('Position');`
 - » `yLabel('Velocity');`
- The phase plane is just a plot of one variable versus the other:



Velocity=0 when
theta is the greatest

Velocity is greatest
when theta=0