

# Math 211

Lecture #14  
MATLAB's ODE Solvers

February 16, 2001

## Solvers

MATLAB has several solvers.

- ode45
  - ◊ This is the first choice.
- ode23
- Stiff solvers
  - ◊ ode15s

Return

## ode45

- Uses variable step size.
  - ◊ Specify error tolerance instead of step size.
  - ◊ MATLAB chooses the step size at each step to achieve the limit on the error.
  - ◊ The default tolerance is good enough for this course.
- Syntax  
`[t,y] = ode45(derfile,[t0,tf],y0);`

Solvers

Return

## Solving Systems

- Example:

$$x' = v$$

$$v' = -9.8 - 0.04v|v|$$

- Change to vector notation. (Use MATLAB vector notation)

$$\diamond u(1) = x$$

$$\diamond u(2) = v$$

Return

## Derivative m-file ball.m

```
function upr = ball(t,u)

x = u(1);
v = u(2);
xpr = v;
vpr = -9.8 - 0.04*v*abs(v);
upr = [xpr; vpr];
```

System

## Derivative m-file ballshort.m

```
function upr = ballshort(t,u)

upr = zeros(2,1);
upr(1) = u(2);
upr(2) = -9.8 - 0.04*u(2)*abs(u(2));
```

System

## Solving Higher Order Equations

- Reduce to a first order system and solve the system.
- Example: The motion of a pendulum is modeled by

$$\theta'' = -\frac{g}{L} \sin \theta - D\theta'.$$

- Introduce  $\omega = \theta'$ . Notice

$$\omega' = -\frac{g}{L} \sin \theta - D\omega.$$

Return

## Equivalent First Order System

$$\theta' = \omega$$

$$\omega' = -\frac{g}{L} \sin \theta - D\omega$$

- Change to vector notation. (Use MATLAB vector notation)
  - ◇  $u(1) = \theta$
  - ◇  $u(2) = \omega$

Higher order

Return

## Derivative m-file pend.m

```
function upr = pend(t,u)

L= 1;
global D
th = u(1);
om = u(2);
thpr = om;
ompr = -(9.8/L)*sin(th) - D*om;
upr = [thpr; ompr];
```

Pendulum