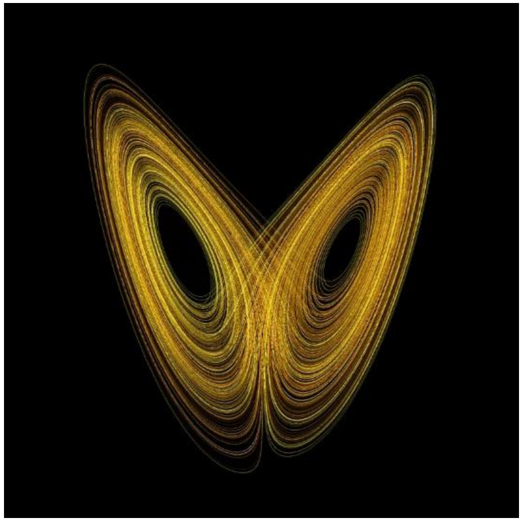
Using MATLAB for Dynamic Simulations

MIP track, week 1

Dynamical Systems as ODE's



http://math.case.edu/files/2014/01/image3031.jpg

$$F\left(x, y, y', \cdots y^{(n-1)}\right) = y^{(n)}$$

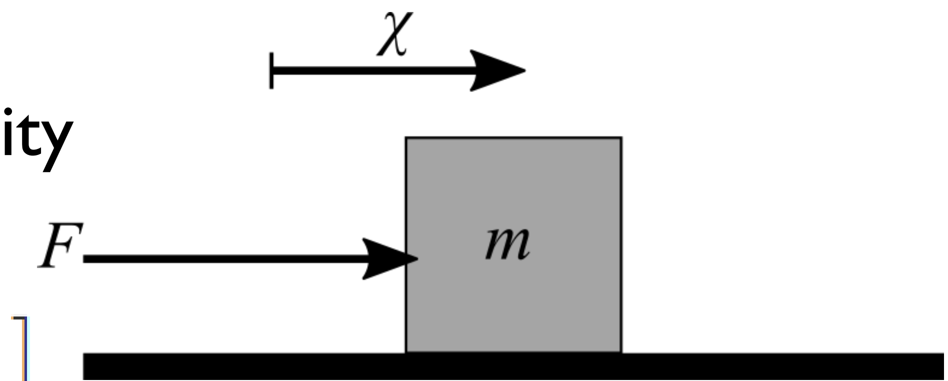
- Order of an ODE is the highest order derivative that appears
- Mechanical systems are usually second order
- Recall Newton's second law $m\ddot{\chi} = F$

State-space for Second Order Systems

- Inertia means need velocity
- Define vector equation

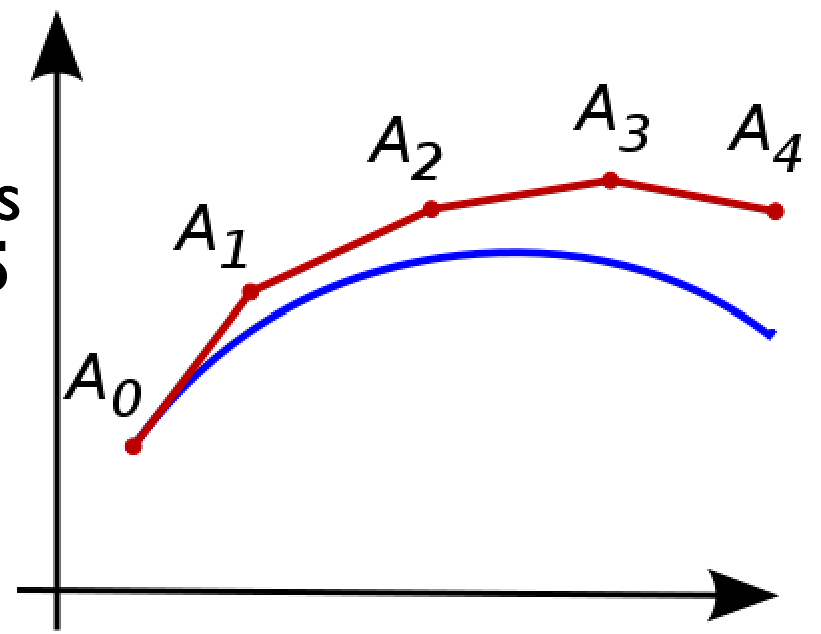
$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} := \begin{bmatrix} \chi \\ \dot{\chi} \end{bmatrix} \implies \dot{x} = \begin{bmatrix} x_2 \\ F/m \end{bmatrix}$$

- x is called the state
- First order
- MATLAB can integrate



Numerical ODE Integration

- Consider $\dot{x} = \alpha$
- Could use fixed timestep, *dt*
- Set $x(t_{k+1}) = x(t_k) + \alpha \cdot dt$
- If $\alpha(t)$ is not fixed, the results will be inaccurate
- MATLAB estimates how the right side is changing and picks the best timestep *dt* in **ode45**
- Only works when right hand side is *smooth*



Example: Harmonic Oscillator

- Consider $\ddot{x} = -x$
- Initial condition $x(0) = 1, \dot{x}(0) = 0$

```
function ode_example()
 X0 = [1,0];
 tspan=[0,10];
 [t, X] = ode45(@shosc, tspan, X0);
 clf
 hold all
 plot(t, X(:,1))
 plot(t, X(:,2))
 hold off
end

function Xd = shosc(t, X)
 x = X(1);
 xd = X(2);
 Xd = [xd; -x];
end
```