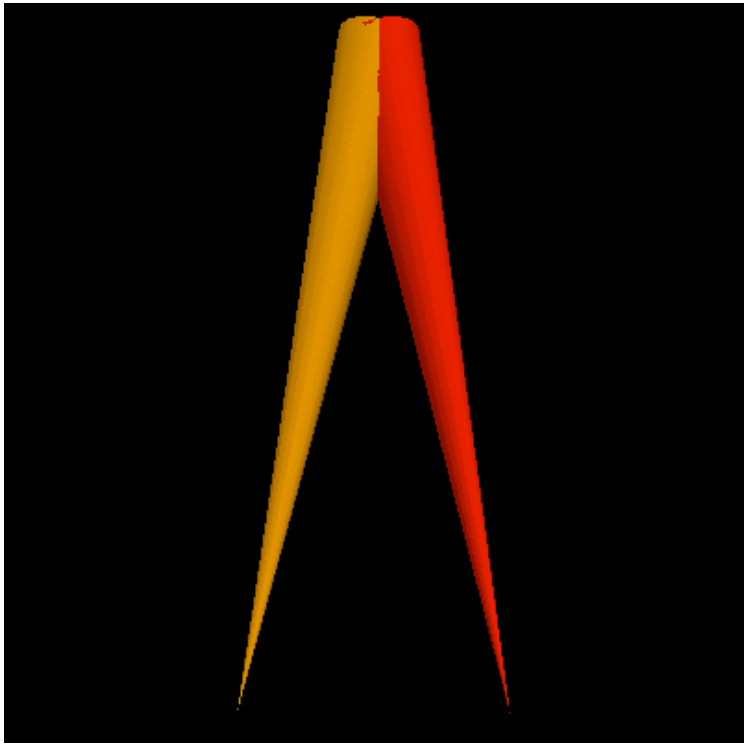
Pendulum and Walker

Physics 7682 / CIS 6229

Science Goals:

- Solving Differential Equations
  - Basic ideas of ODE solvers
  - Accuracy, Stability, Fidelity
- Nonlinear Dynamics Concepts
  - Phase Plane Portraits
  - Poincare Sections
  - Period Doubling
  - Chaos
- Models of locomotion



Graphics, functions as objects, manipulating data sets

Solving Differential Equations

- "Easy task"
  - Excellent off-the-shelf algorithms and functions

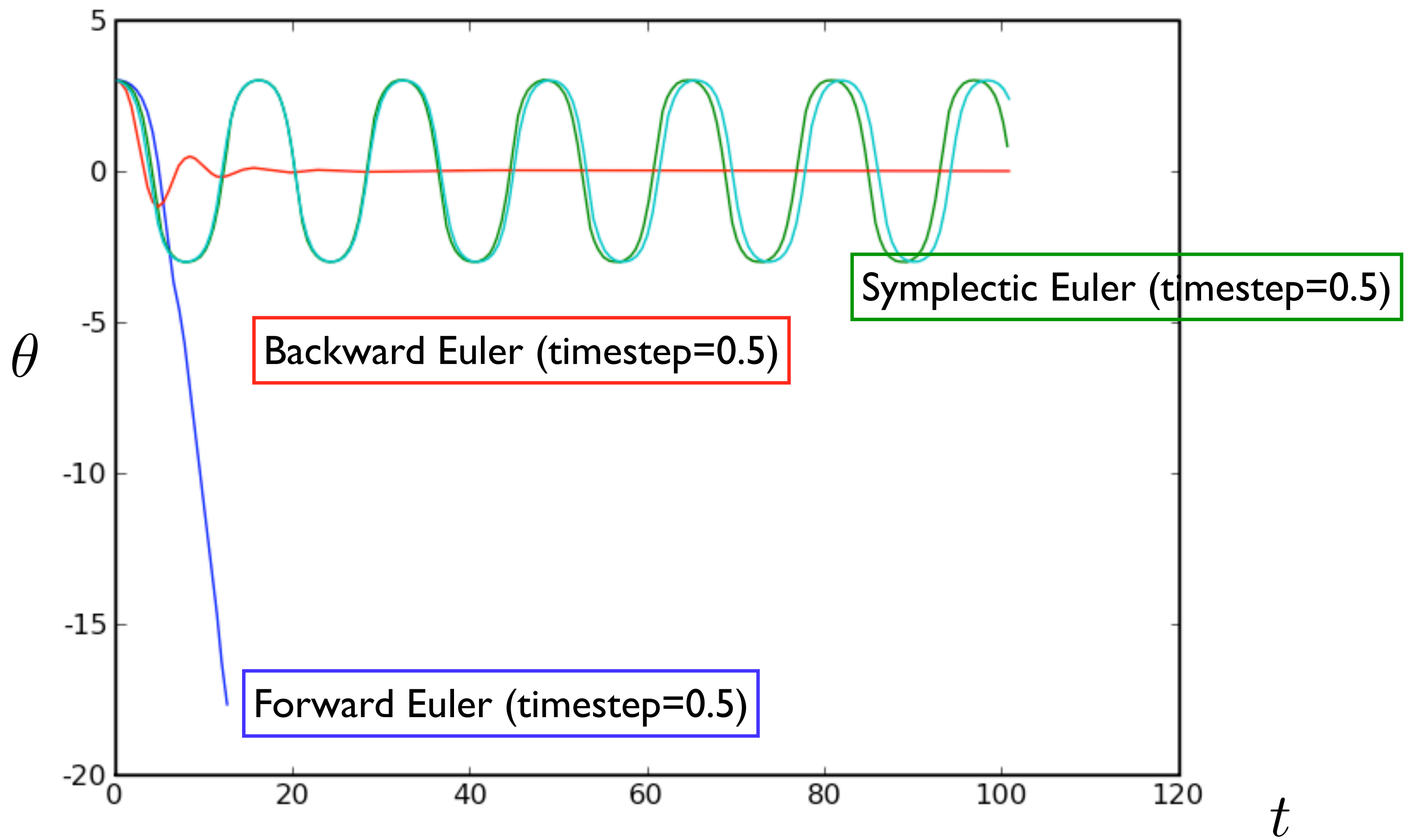
Basic Idea:
convert differential equation into difference equation

$$\frac{df}{dt} = -f \quad \longrightarrow \quad \frac{f(t_{i+1}) - f(t_i)}{\delta t} \approx \frac{df}{dt}(t_i) = -f(t_i)$$

Different algorithms use different approximations:
Pendulum exercise explores choices

"Order" of algorithm: how accuracy scales with timestep

Not all first order algorithms are created equal



(Understand by looking at phase space plots)

General Procedure

1. Convert to set of first order equations

$$\frac{d^2\theta}{dt^2} = -\frac{g}{L}\sin(\theta)$$

$$\frac{d\theta}{dt} = \omega$$
$$\frac{d\omega}{dt} = -\frac{g}{L}\sin(\theta)$$

2. Choose units

$$\frac{d\theta}{dt} = \omega$$
$$\frac{d\omega}{dt} = -\sin(\theta)$$

Using Scipy's integrator

```
def PendulumDerivArray(vars,t):
    theta,omega=vars  # unpack vars
    return numpy.array([omega,-sin(theta)])

times=arange(0,100,0.1)

InitialConditions=[3,0]

trajectory=odeint(PendulumDerivArray,InitialConditions,times)
```

Returns an array of the form [[theta0,omega0],...

```
array([[3,0],[2.9929382,-0.0143531],...])
```

plot theta, and omega vs time with

```
plot(times,trajectory)
```

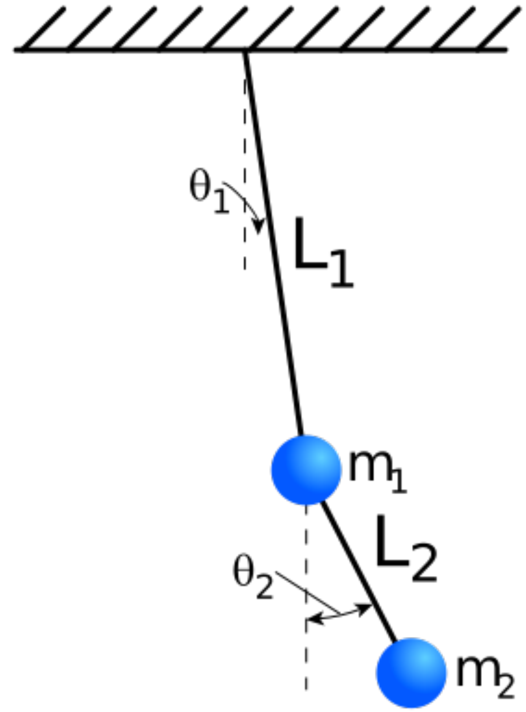
just theta:

```
plot(times,trajectory[:,0])
```

Interesting construction: pass a function to another function. Functions are objects. Functions can even return functions:

smartodeint is defined in pendulum.py is an integrator which returns functions.

Double Pendulum

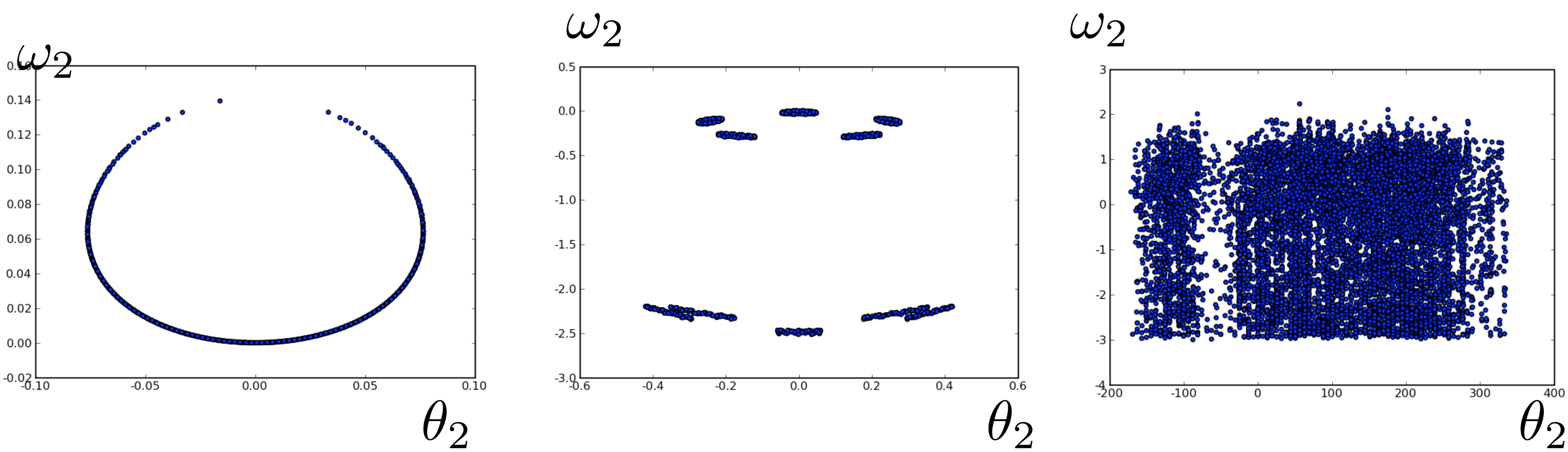


- Classic example of chaotic system
  - sensitive dependance on initial conditions
- Trajectories are in 5 dimensional space
  - how to deal with all that data!

Python demo + Real Demo

Poincare Sections

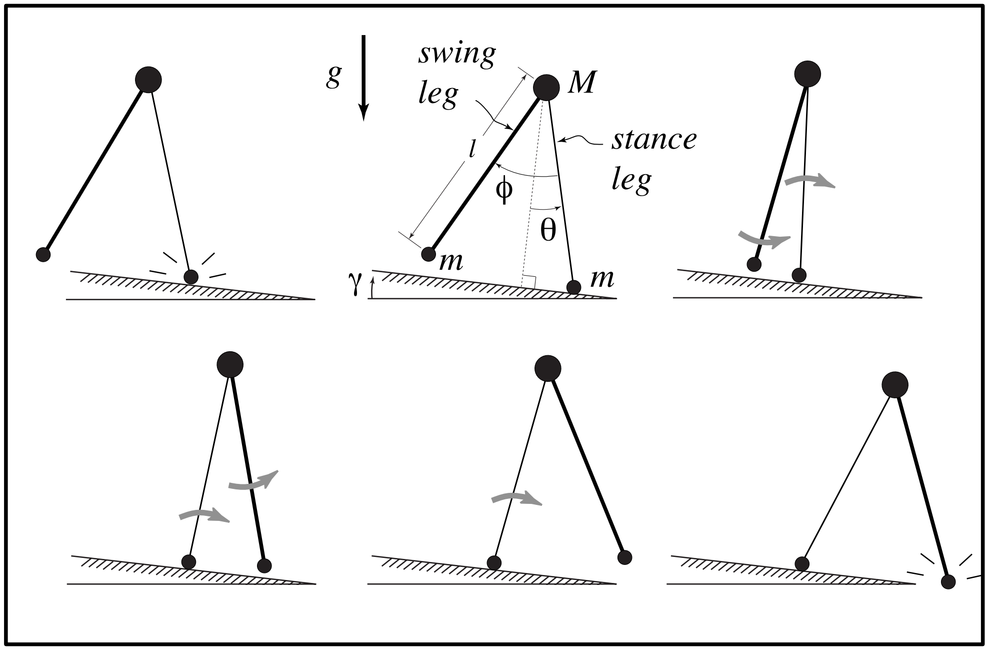
- Trajectories in 5D
  - Translational invariance in time: 4D phase space suffices
  - Conservation of Energy: trajectories live on 3D manifold
  - Poincare section: look at θ_2 and ω_2 on this manifold when $\theta_1 = 0$

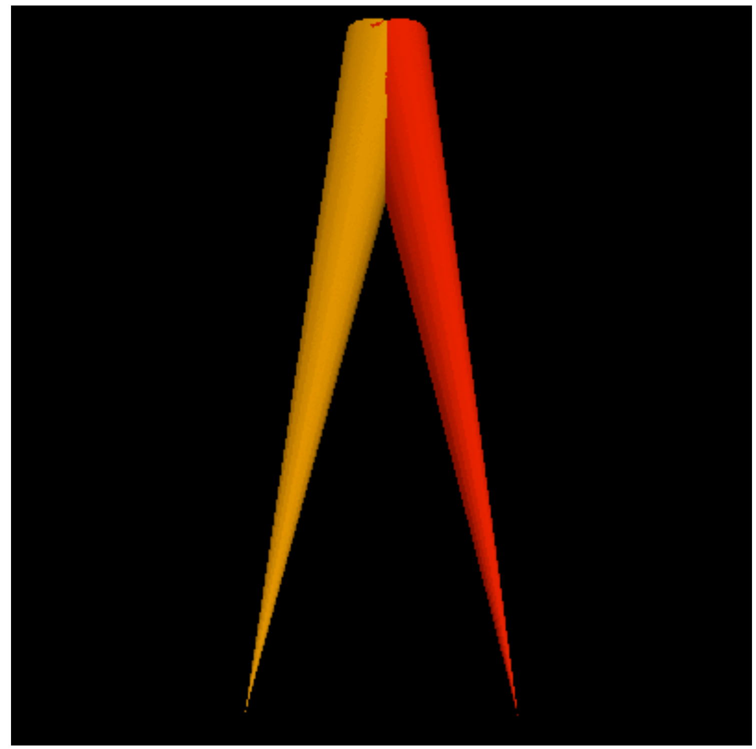


$\omega_1(t=0) = 0.1$ $\omega_1(t=0) = 1.3$ $\omega_1(t=0) = 1.5$

Walker

- Simplified model of bipedal motion:

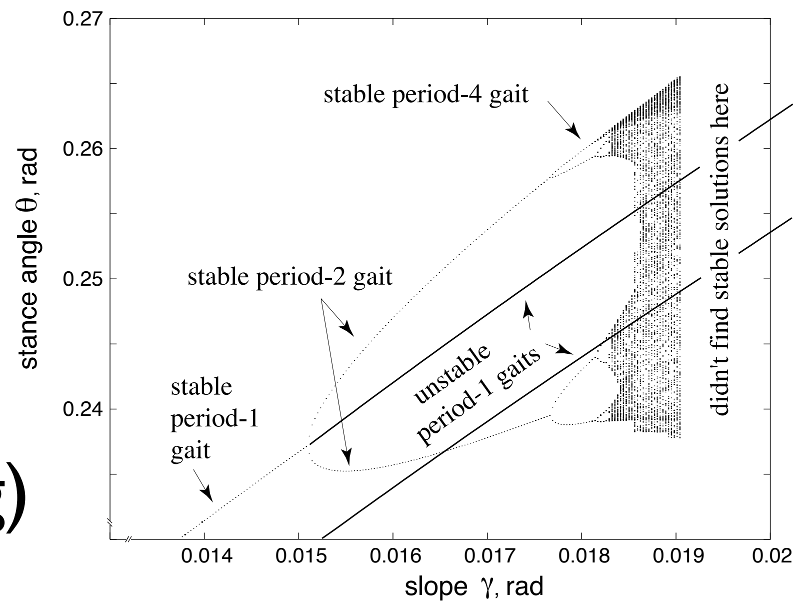




Double pendulum: but pivot point switches from end of one “leg” to other whenever there is a “heelstrike”

Walker

- Remarkable: passive model -- can walk
- Nontrivial "phase diagram"



- Chaos, stable/unstable limit cycles, period doubling (limping)
- Biological significance: Evolution of geometry of legs?