PARTICLE DYNAMICS

John R. Williams and *Abel Sanchez*

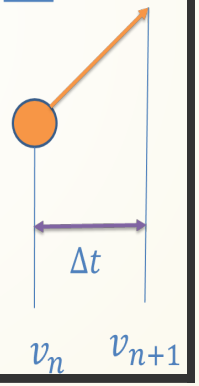
Loops

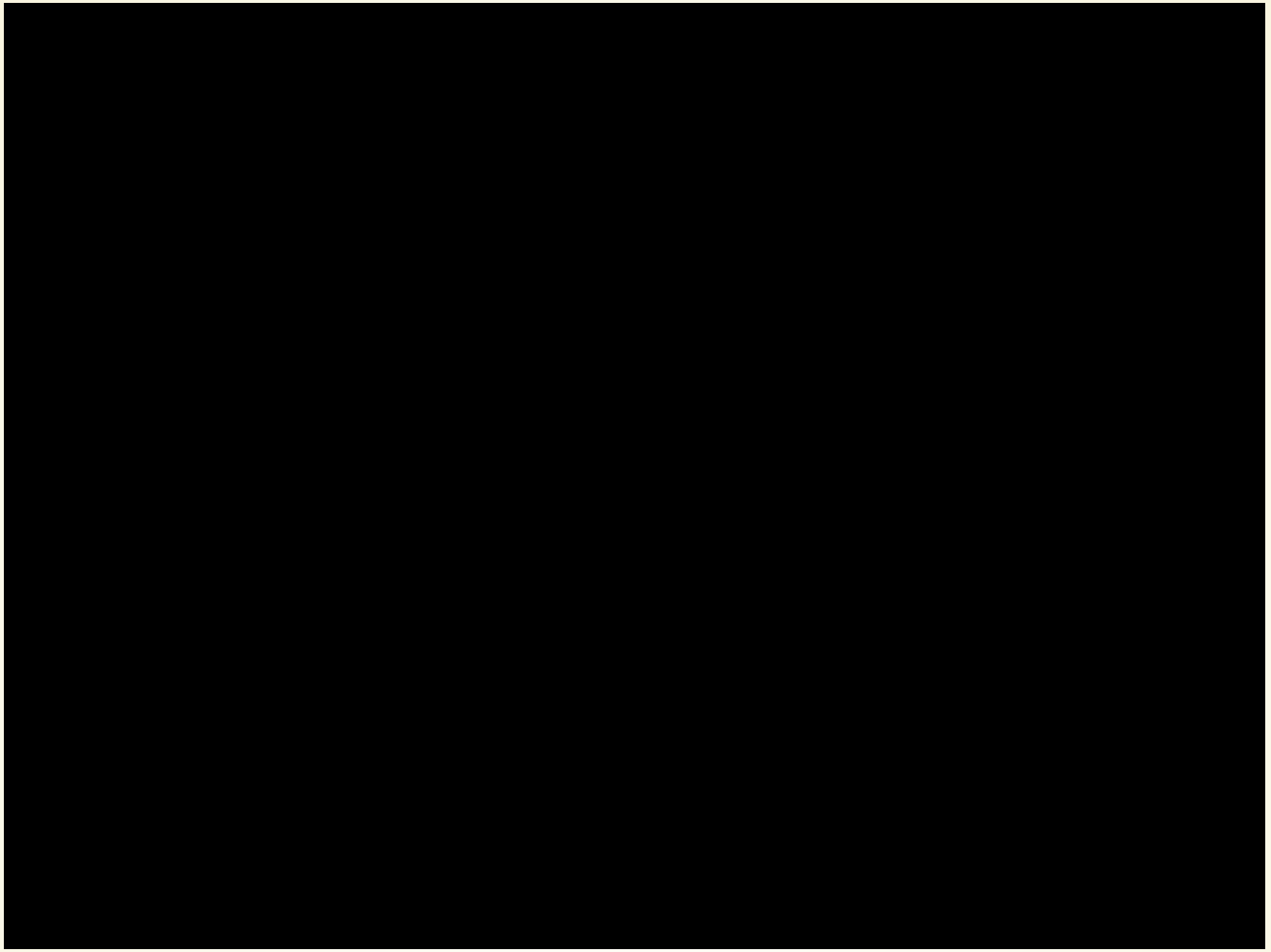
1) Calculate the total force on each particle
2) Update Velocity
3) Update Position

$$a = \frac{1}{m}\sum F$$

$$v_{n+1} = v_n + a\,\Delta t$$

$$x_{n+1} = x_n + v\,\Delta t$$



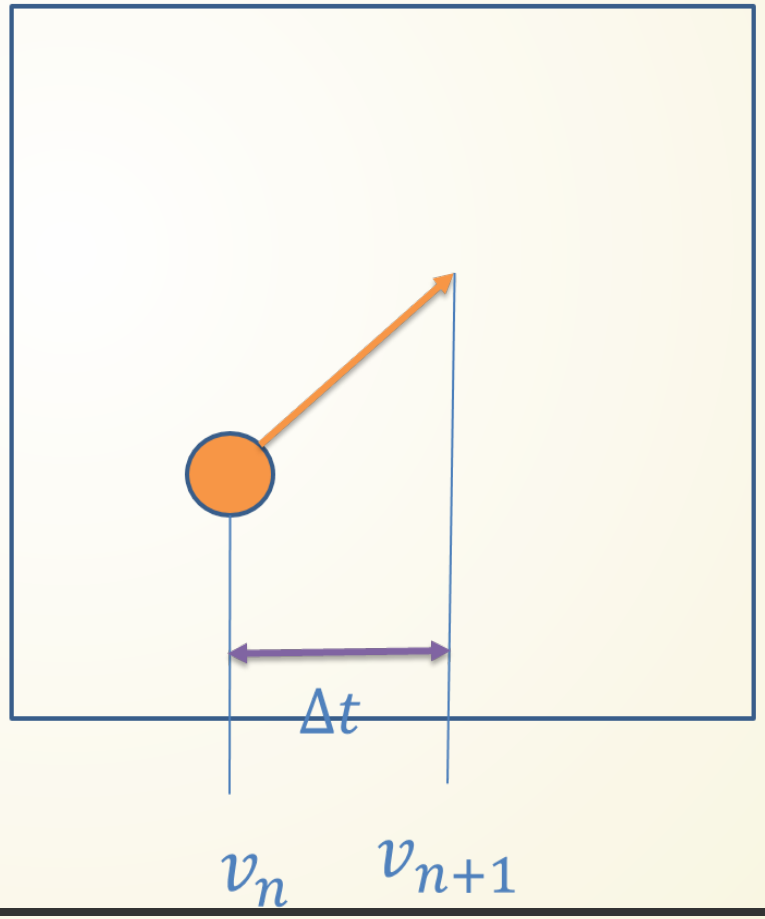


NEWTON'S EQUATIONS

Using Newton's Equation F = ma we get a Difference Equation for the Velocity Update

$$a = F/m$$

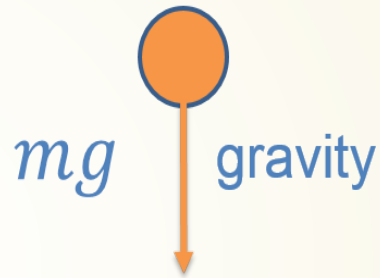
$$v_{n+1} = v_n + \frac{F}{m}\Delta t$$



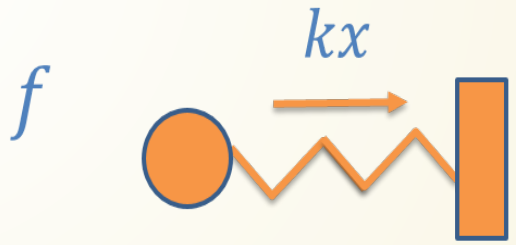
NEWTON'S EQUATIONS

Newton's Equation

$$m\ddot{x} = f$$



$$m\ddot{x} = f - kx$$

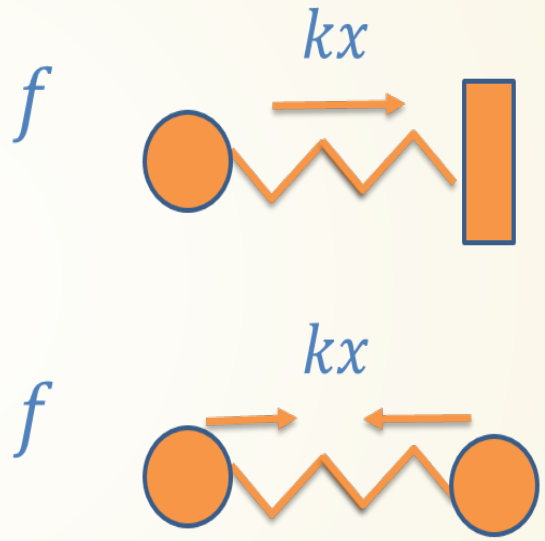


We can deal with multiple forces on a body by summing them together. Here we add a spring force

NEWTON'S EQUATIONS

Newton's Equation

$$m\ddot{x} = f - kx$$

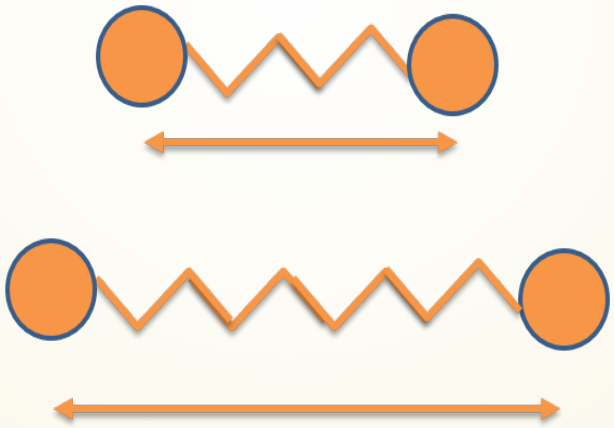


In the case of two particles connected by a spring the force is equal but in opposite directions on the two bodies.

NEWTON'S EQUATIONS

Vector Geometry

Force in spring will be proportional to change in length. ie final - original

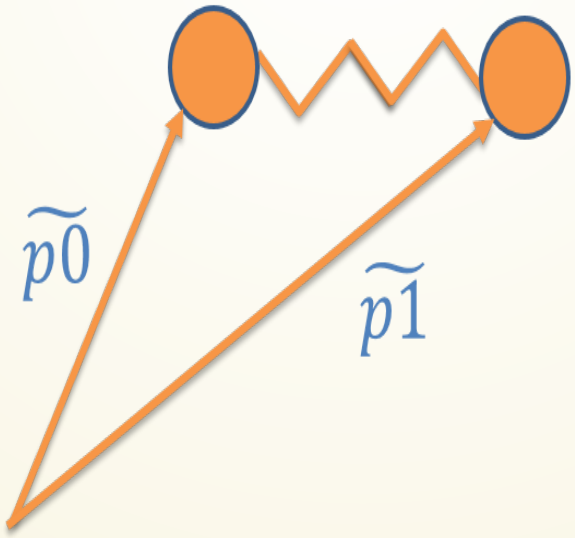


$$F = k(L_f - L_0)$$

NEWTON'S EQUATIONS

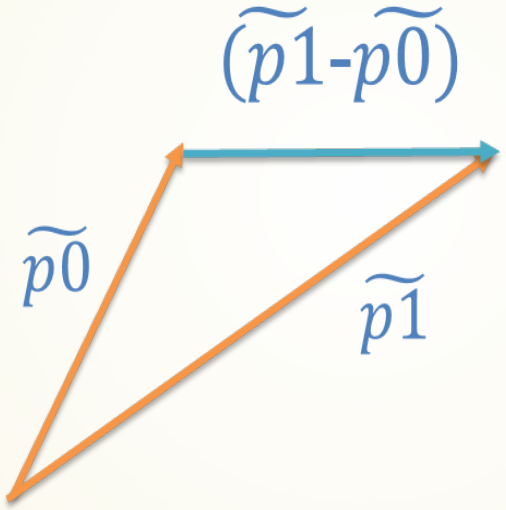
Vector Geometry

What is distance between points p0 and p1



NEWTON'S EQUATIONS

Magnitude of a Vector

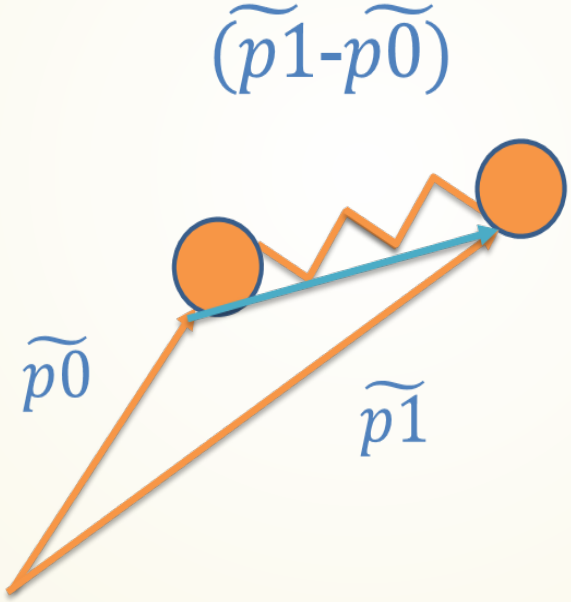


We need the magnitude of the vector

$$\widetilde{|(p1\text{-}p0)|}$$

NEWTON'S EQUATIONS

What is direction of spring force



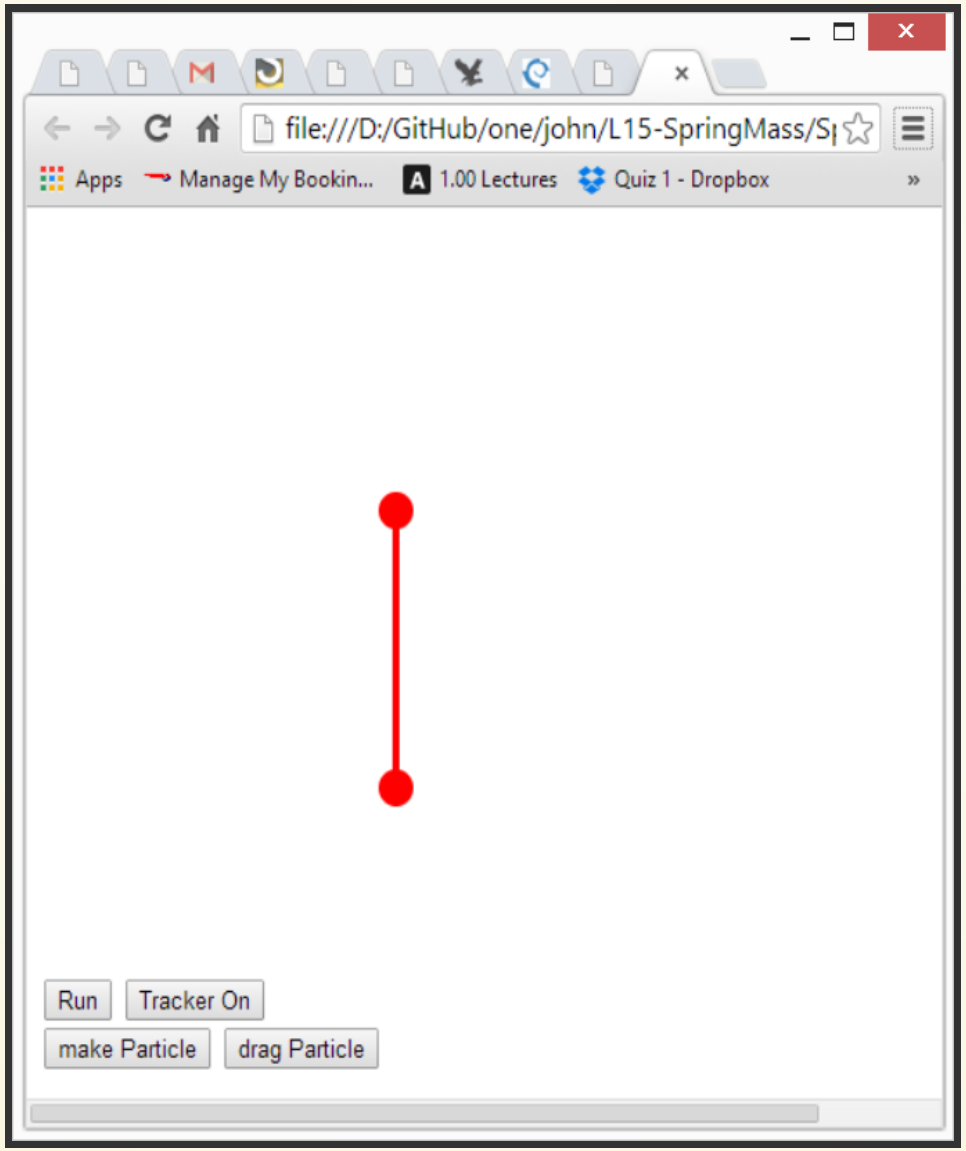
We need a unit vector along the spring

```
function Spring(p0,p1,stiffness,velocity){
        var masses = [p0,p1];
        var originalLength = p1.center.minus(p0.center).abs();

        var updateSpringForces = function updateSpringForces(){
                var c0 = masses[0].center;
                var c1 = masses[1].center;
            var len1 = c1.minus(c0).abs();
            var forcemag = (len1 - originalLength)*stiffness;
            var unitVec = c1.minus(c0).unit();

            var f = unitVec.scale(forcemag);
            masses[0].force = masses[0].force.plus(f);
            masses[1].force = masses[1].force.minus(f);
          };

  return {masses:masses, originalLength:originalLength,
    updateSpringForces:updateSpringForces};
}
```


your turn now

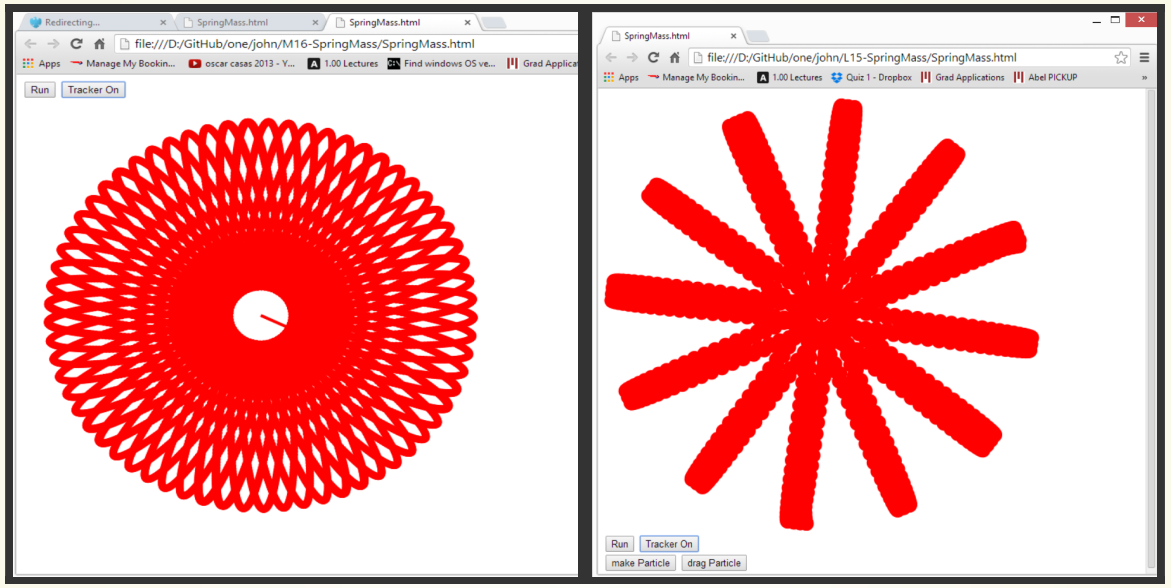
In class Assignment 1.

Here is code starter code to create masses and springs. We'll start with just two masses and one spring to test the code.



```
<!DOCTYPE html>
<html>
<head>
<script src = 'Vector2D.js'></script>
<script>
var canvas;
var context;
var stiffness = 1;
var mass    = 1;
var deltaT  = 0.1;
var gravity = -1;
var velx, vely;
var trackerFlag =0;
var particles=[];
var springs = [];

window.onload=getCanvas;
function getCanvas(){
  canvas = document.getElementById("myCanvas");
  context =  canvas.getContext ('2d');
}

function Run(){
  setInterval(UpdateAll, 10);
}
function UpdateAll(){
  // zero all the forces on the masses
  for(var i=0;i<particles.length;i++){
    particles[i].force = Vector(0,0);
  }
  for(i=0;i<springs.length;i++){  // update forces on masses
    springs[i].updateSpringForces();
  }
```

Create two particles. Drag the fixed one so that you get the other to oscillate and create at least two different patters for the orbits. Hand in your screen shots.



Assignment 2

Change the code so that 3 particles are tied together by 2 springs. Hand in a screen shot of the two particles oscillating about the fixed one.

Assignment 3 - Fracture of Materials

If the force in any of the springs exceeds 20 then remove that spring. ie the particles break apart. Make at least 3 groups of 3 particles per group and hand in a screen shot after two of the groups break apart. Hint: Find the code where the spring force is generated. Flag the spring if its greater than 20 (use setDeleteSpring = 'this' to grab a reference to the spring object.) Now after the loop over all springs use var k = springs.indexOf(setDeleteSpring) and then delete the spring from 'springs' by using 'splice'

THE END