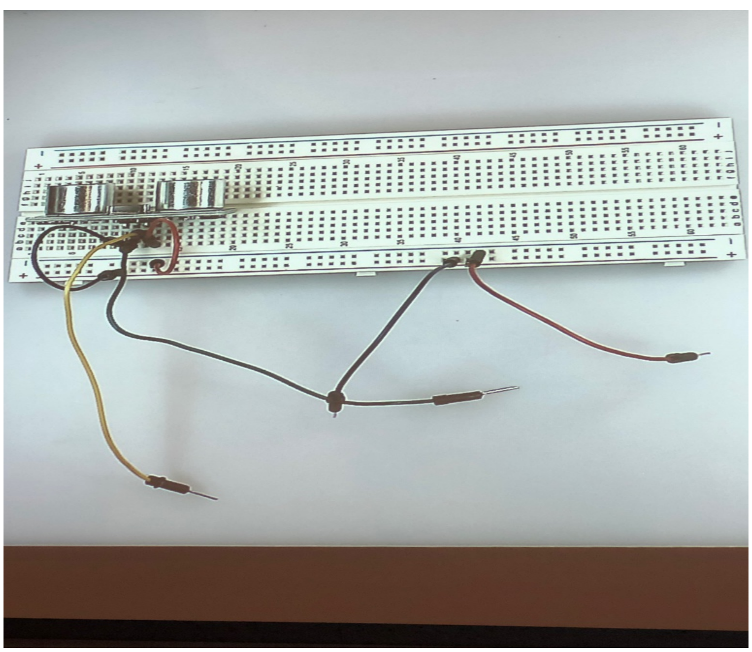
Announcements:

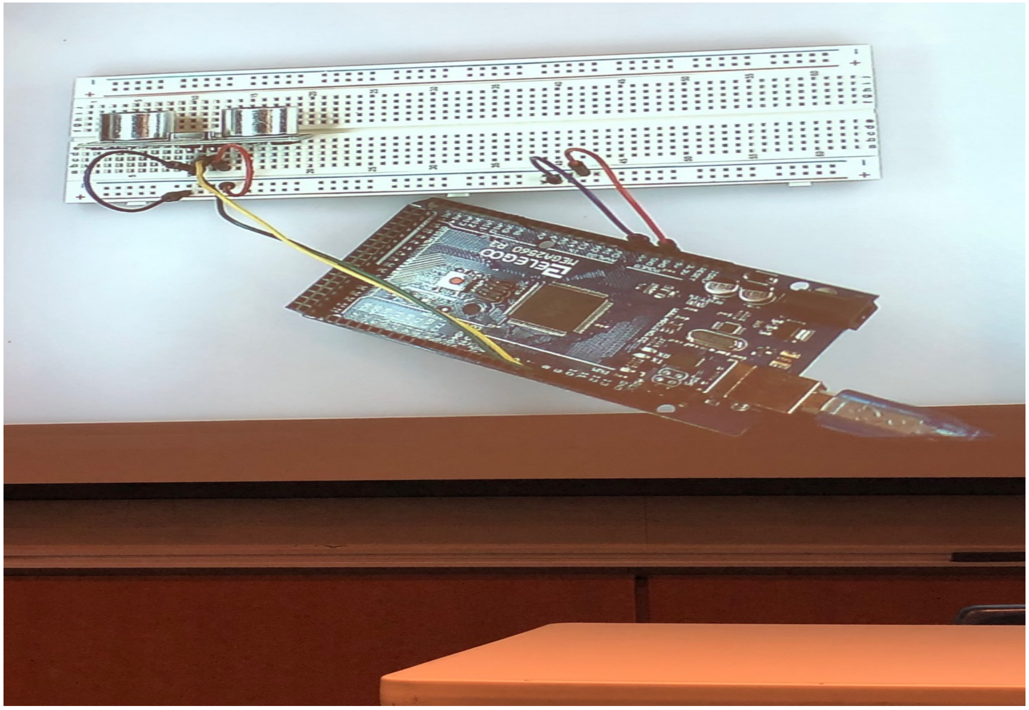
- Homework 2 due 3/31
    - 2 problems
    - You should include video description and code
    - You zip and then submit on sakai
    - 3 weeks to get the assignment done, no exceptions, if you have questions as them early.
- Problem 1
    - Load a 3d model and display it on the screen
        - Humanoid model
            - You can import a human model or robot model
    - Write controls to rotate the arms
    - Upper arms, lower arms (4 b/c of 2 arms)
    - Legs
        - 8 controls total
    - You can find a 3d model or make one it is up to you
- Problem 2
    - Intruder Alarm
        - Measure distance with an ultrasonic sensor. If the distance falls below a threshold you should start to beep and if the intruder gets out of that safety radius the beeping should stop.
        - If you mount the sensor on the servo motor to scan 180 degrees, this will be considered to be extra credit. More details can be found on the class website
- Expect to have feedback on program proposal and homework 1 after break.

I. **Ultrasonic Sensor**

Here is the starting setup:



Final setup:



a. Basic code for Arduino:
   i. Globals
      1. Const int trigPin = 9;
         Const int echoPin =10;
         Long duration;
         Long distance;
   ii. Void setup
         pinMode(trigPin, OUTPUT) // pin 9 measures the distance
         pinMode(echoPin, INPUT) // pin 10 sends and responds to the signal
         Serial.begin(9600) // configures the serial monitor for the code
   iii. Void loop()
         digitalWrite(trigPin, LOW)
         delayMicroseconds(2) // these delays help to limit the noise and properly capture the communication
         digitalWrite(trigPin, HIGH)
         delayMicroseconds(10)
         digitalWrite(trigPin, LOW)
         duration = pulseIn(echoPin, HIGH)
         distance = duration*.034/2;
         Serial.print("Distance: ")
         Serial.println(distance)

1. The sensor has two lobes one sends and the next receives the time that it takes to get from one lobe to the next. This is then converted to the distance. Distance of going and distance of coming back is accounted for by dividing by 2.

II. Ordinary Differential Equations

a. All of graphics is to highlight something about reality. Ultimately you will be integrating some equations at some point in time. Most of graphics have the equations that we discuss in this class ingrained in it.

$$\frac{du}{dt} = \dot{u} = u(v - 2)$$

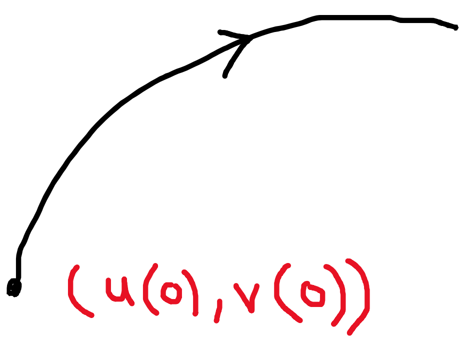
$$\frac{du}{dt} = \dot{v} = v(1 - u)$$

$$u(t), v(t)\} \; Function\ of\ Time$$

****Note: The value of the derivative of u depends on v and vice versa. The input is going to be the value of u(0), v(0) the output is the u(t), v(t) for any t. u and v are simply variables, they can be replaced by other variables.

b. If you think of *u* as *x* and *v* as *y,* they help to tell the trajectory of a point moving forward when we consider them on a plane (u(0), v(0)).

III. The **Lotka-Volterra Model** was used as an explanation of the predator-prey problem:



i. This problem is essentially that in nature if there is a predator species and a prey species if there is too much prey and less predators, the predators will eat and flourish until they prey decrease. Eventually there will not be enough food for all of the predators and then they will die off giving the prey the ability to populate and their species increases. As one group increases the next decreases until there is a balance.

ii. This represents the Lotka Volterra model. On planes these are shown as circles because they represent a cycle

IV. Discretize Time

a. We are going to take steps of fixed delta t

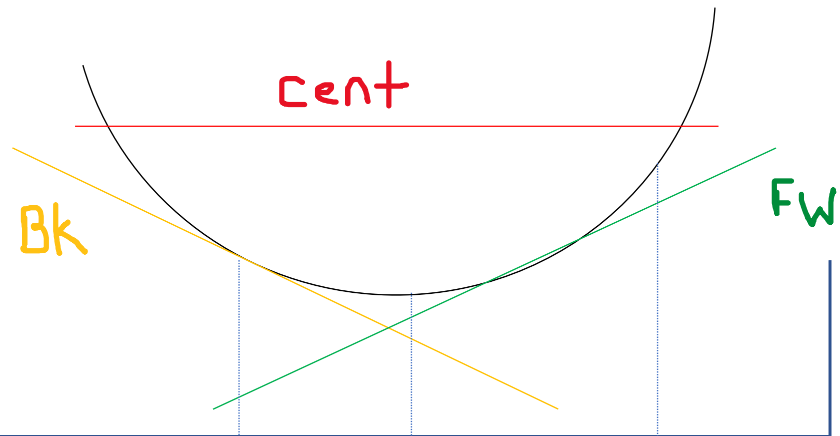
b. We are going to adopt the Convention:

$$u^0 = u(o) \qquad v^0 = v(0)$$

$$u^1 = u\ (\Delta\ t) \qquad v^1 = v\ (\Delta\ t)$$

$$u^2 = u\ (2\ \Delta\ t) \qquad v^2 = v\ (2\ \Delta\ t)$$

$$u^n = u\,(n \vartriangle t) \qquad v^n = v\,(n \vartriangle t)$$



$$\frac{du}{dt} = \frac{\delta n\ (change\ of\ n)}{\delta t\ (time\ in\ which\ the\ time\ was\ measured)}$$

<u>Our options</u>

1. $\frac{u^n - u^{n-1}}{\Delta t}$ (Backward differencing)
2. $\frac{u^{n+1} - u^n}{\Delta t}$ (Forward differencing)
   a. Both 1 and 2 are 1st order accurate, meaning, if you make the denominator smaller it becomes more accurate.
3. $\frac{u^{n+1} - u^{n-1}}{2\Delta t}$ (Central differencing)
   a. 2nd order accurate

V. Forward differencing (FW)

a. 1st order

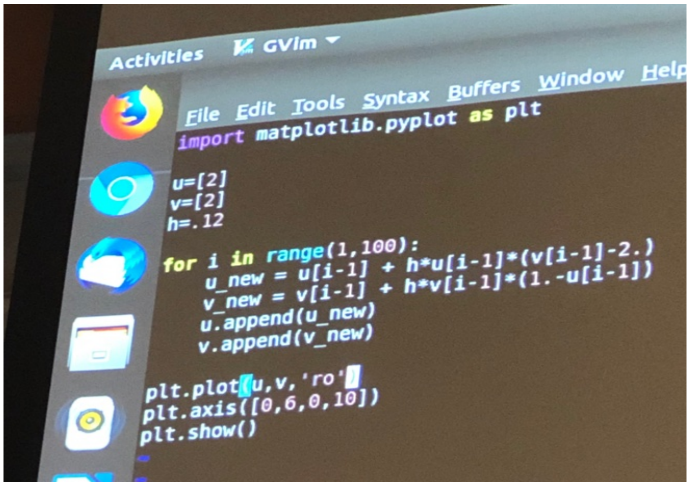
$$\dot{u} \equiv u\,(v-2)$$
$$\dot{v} \equiv v(1-u)$$
$$\frac{u^{n+1} - u^n}{\Delta t} = u^n(v^n - 2)$$
$$\frac{v^{n+1} - v^n}{\Delta t} = v^n(1 - u^n)$$

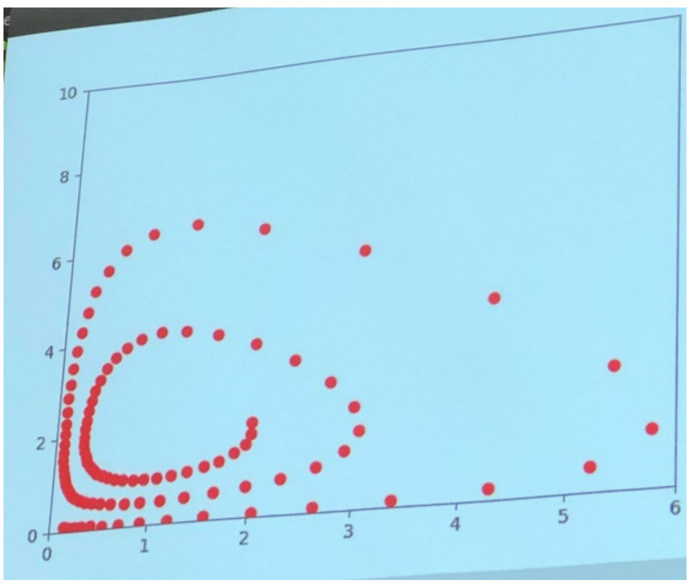
Where n is steps forward

$$u^{n+1} = u^n + \Delta t\, u^n(v^n - 2)$$
$$v^{n+1} = v^n + \Delta t\, v^n(1 - u^n)$$
$$(u^0, v^0) \rightarrow (u^1, v^1) \rightarrow (u^2, v^2) \ldots (u^n, v^n)$$

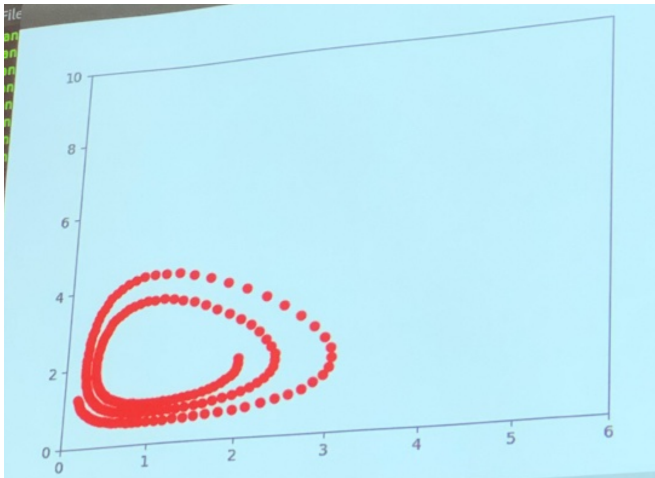
VI. **Forward Differencing code**

a. The code itself says that you are starting at this point and then spiraling out.





b. If you change the time steps the spiral gets smaller. The smaller the step the more accurate you become



c. How far away you move in time is limited.

VII. Backward differencing (BK)

d. 1st order
e. Your derivative changes how your tangent will look
f. Your current value will depend on the values of the next time step

$$\frac{u^{n+1}-u^n}{\Delta t} = u^{n+1}(v^{n+1}-2) \ldots (1)$$

$$\frac{v^{n+1}-v^n}{\Delta t} = v^{n+1}(1-u^{n+1}) \ldots (2)$$

Equation

(1) $u^{n+1} = u^n + \Delta t\, u^{n+1}(v^{n+1}-2)$

$$u^{n+1}\{\,1 - \Delta t\,(v^{n+1}-2)\} = u^n$$

$$u^{n+1} = \frac{u^n}{1-\Delta t(v^{n+1}-2)} \ldots (3)$$

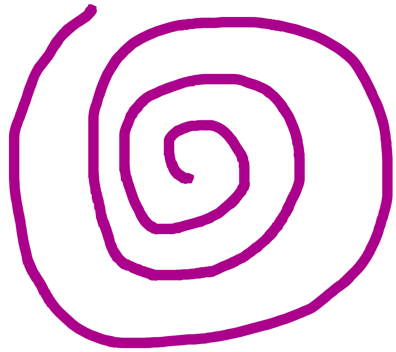
Substitute 3 in 2, $\frac{v^{n+1}-v^n}{\Delta t} = v^{n+1}\{\frac{1-u^n}{1-\Delta t(v^{n+1}-2)}\}$

$$= v^{n+1}\{\frac{1-\Delta t\,(v^{n+1}-2)-u^n}{1-\Delta t(v^{n+1}-2)}\}$$

⇨ $(v^{n+1} - v^n)\{1-\Delta t\,(v^{n+1}-2)\} = \Delta t\, v^{n+1}\{1-\Delta t\,(v^{n+1}-2) - u^n\}$

We know this quantity ahead of time
Quadric equation is necessary to solve in v^{n+1}

You will see if you try the code for this you will see that a spiral goes inward.



VIII. Central differencing (Cent)

g. Looking at the value in front of us
h. 2nd order accurate

IX. Symplectic Differencing

i. The proper approach would be to use Symplectic differencing

$$\dot{u} = u(v-2)$$
$$\dot{v} = v(1-u)$$

$$\frac{u^{n+1}-u^{n}}{\Delta t} = u^{n}(v^{n+1}-2)$$

$$\frac{v^{n+1}-v^{n}}{\Delta t} = v^{n+1}(1-u^{n})$$

$$u^{n+1}(v^{n}-2)$$

$$v^{n}(1-u^{n+1})$$

j. If you solve either method, you will get closer to the right answer. These are energy conserving
k. Doing forward differencing for one variable and backward differencing for the next.

*****************You should try and code both backward & symplectic differencing