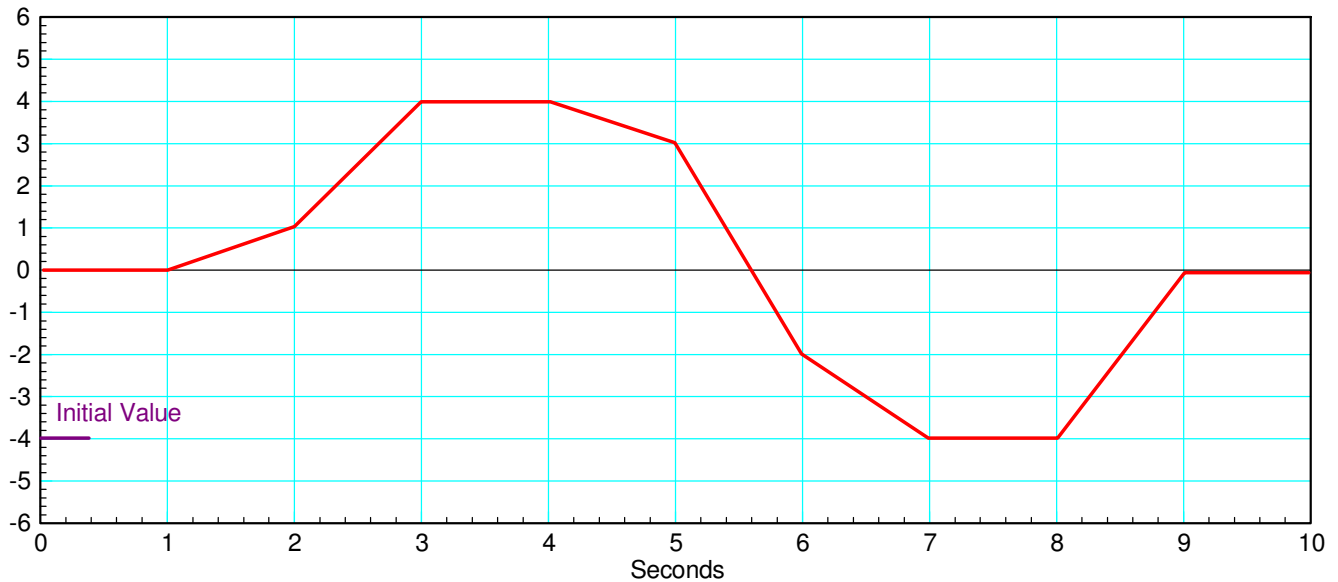


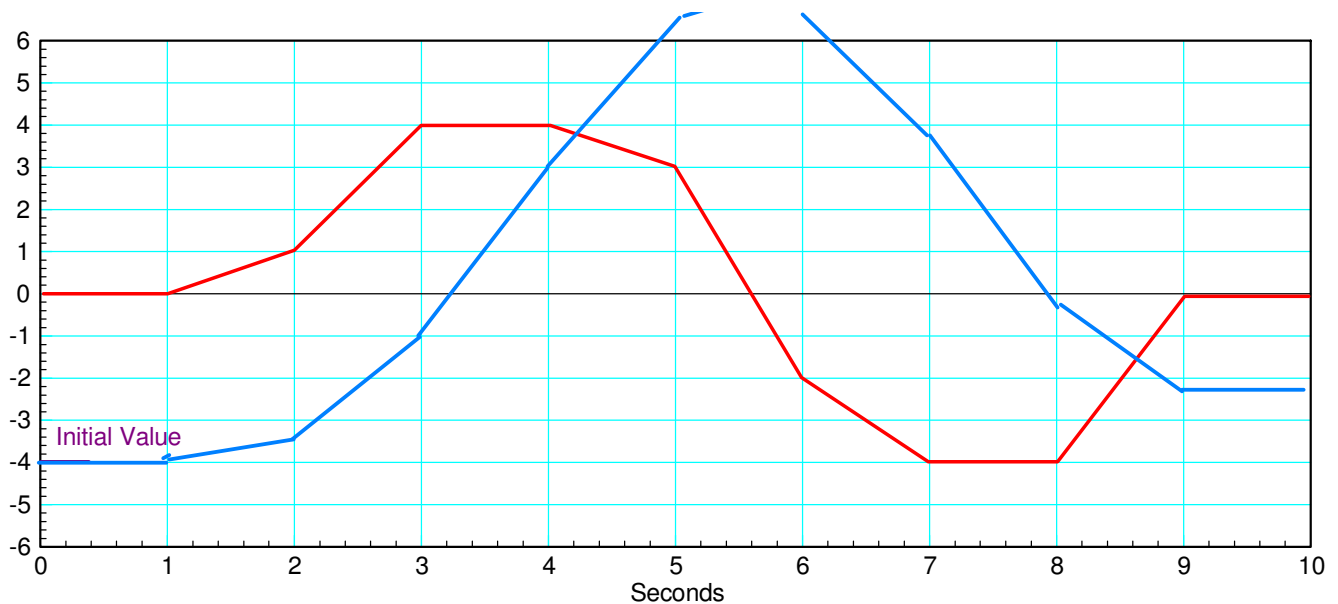
ECE 111 - Homework #7:

Math 166: Integration

1) Sketch the integral of the following function. Assume its initial value is -4.



If this is how much money you are depositing (positive) or withdrawing (negative) from your checking account, what is the balance at each instance?



Numerical Integration

2) Use numerical methods to determine the integral of y

$$y = \left(\frac{\cos(3x)}{1+x^2} \right)$$

$$z = \int y \cdot dx$$

for $-10 < x < 10$. (a plot is sufficient). Assume $z(-10) = 0$.

Create a Matlab function to integrate

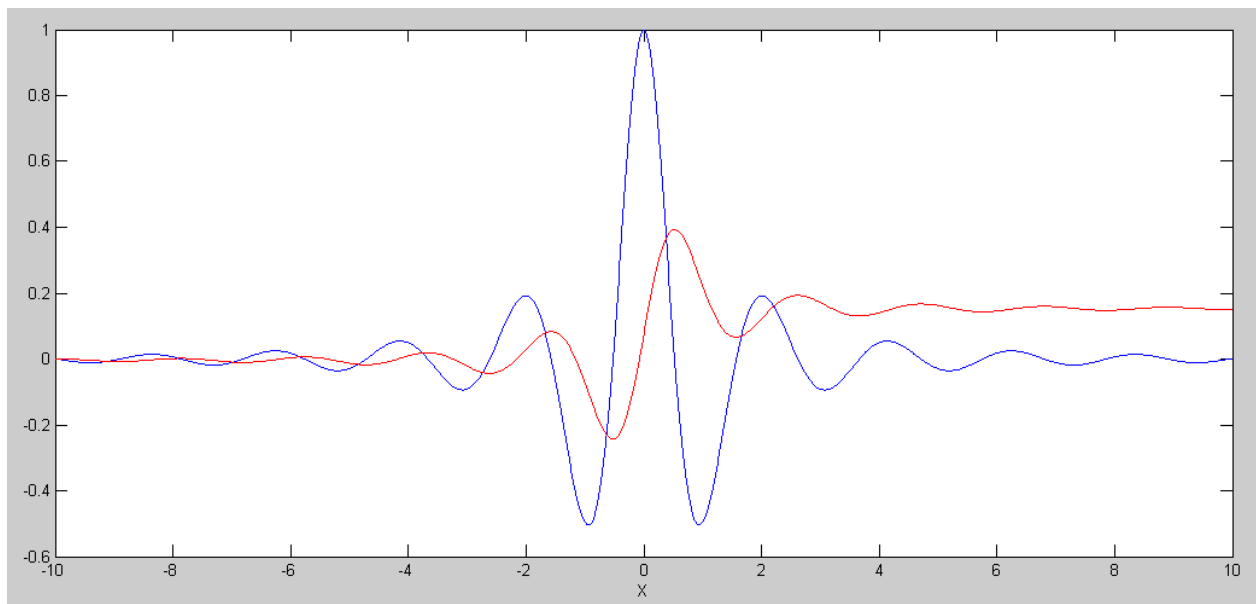
```
function [y ] = Integrate( x, dy )
npt = length(x);
y = 0*dy;
for i=2:npt
    y(i) = y(i-1) + 0.5*(dy(i) + dy(i-1)) * (x(i) - x(i-1));
end
end
```

Use this to integrate y(x)

```
>> x = [-10:0.01:10]';
>> y = cos(3*x) ./ (1 + x.^2);
>> plot(x,y,'b',x,Integrate(x,y),'r')
>>
```

Note:

- If you can get the function into Matlab, you can find the integral using numerical methods
- You'll cover how to find the equation for z(x) when you take Calculus II
- Whatever this equation is, it looks like the red line below.



y(x) (blue) and its integral (red)

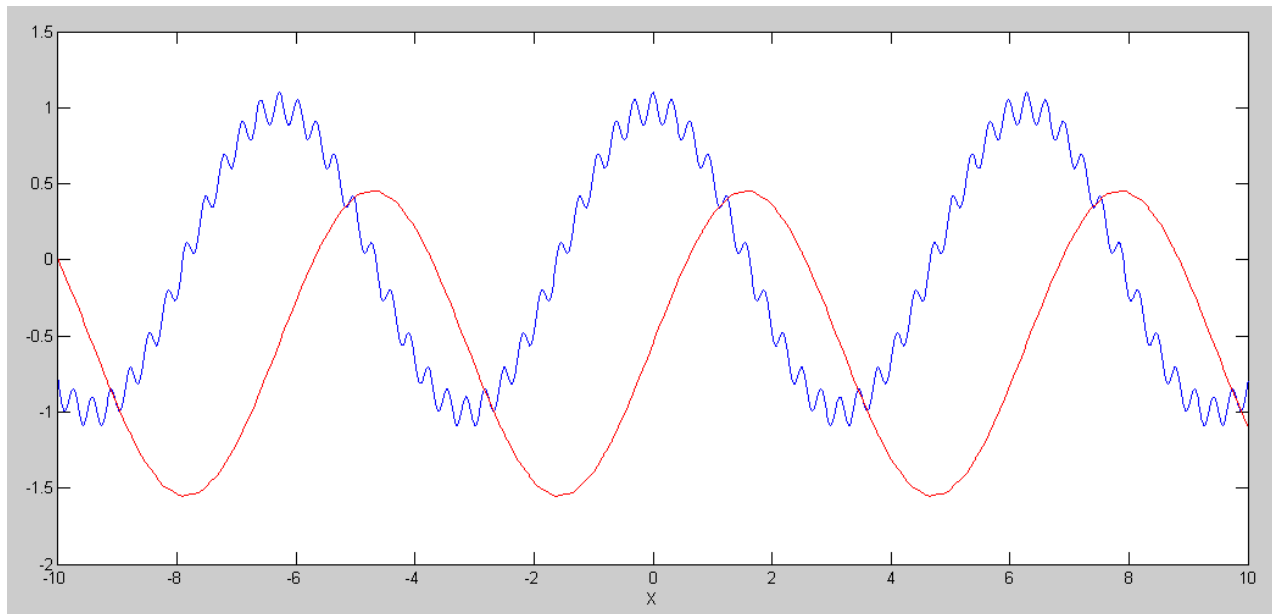
3) Use numerical methods to determine the integral of y

$$y = \cos(x) + 0.1 \sin(20x)$$

$$z = \int y \cdot dx$$

for $-10 < x < 10$. (a plot is sufficient). Assume $z(-10) = 0$.

```
>> x = [-10:0.01:10]';  
>> y = cos(x) + 0.1*cos(20*x);  
>> plot(x,y,'b',x,Integrate(x,y),'r')  
>> xlabel('X')  
>>
```



y(x) (blue) and it's intgegral (red)

Note:

- If you can get the funciton into Matlab, you can find the integral using numerical methods
- Integration attenuated high-frequency terms (good)
- You'll cover how to find the equation for z(x) when you take Calculus II
- Whatever this equation is, it looks like the red line below.

Animation in Matlab with Numerical Integration

4) Calculate the position of a bouncing ball in freefall:

- The acceleration is $y'' = -1.625 \text{ m/s}^2$ (gravity on the Moon)
- If the ball hits the ground ($y < 0$) the velocity becomes positive: $y' = |y'|$
- The initial position is ($x = 0, y = 3$)
- The initial velocity is ($x' = +1, y' = +2$)

Plot the path of the ball for two bounces

- i.e. find the x position of the ball at its 2nd bounce

Matlab Code

```
x = 0;
y = 3;

dx = 1;
dy = 2;

t = 0;
dt = 0.05;
g = -1.625;

Bounce = 0;

while (Bounce < 2)
    ddx = 0;
    ddy = g;

    dx = dx + ddx * dt;
    dy = dy + ddy * dt;

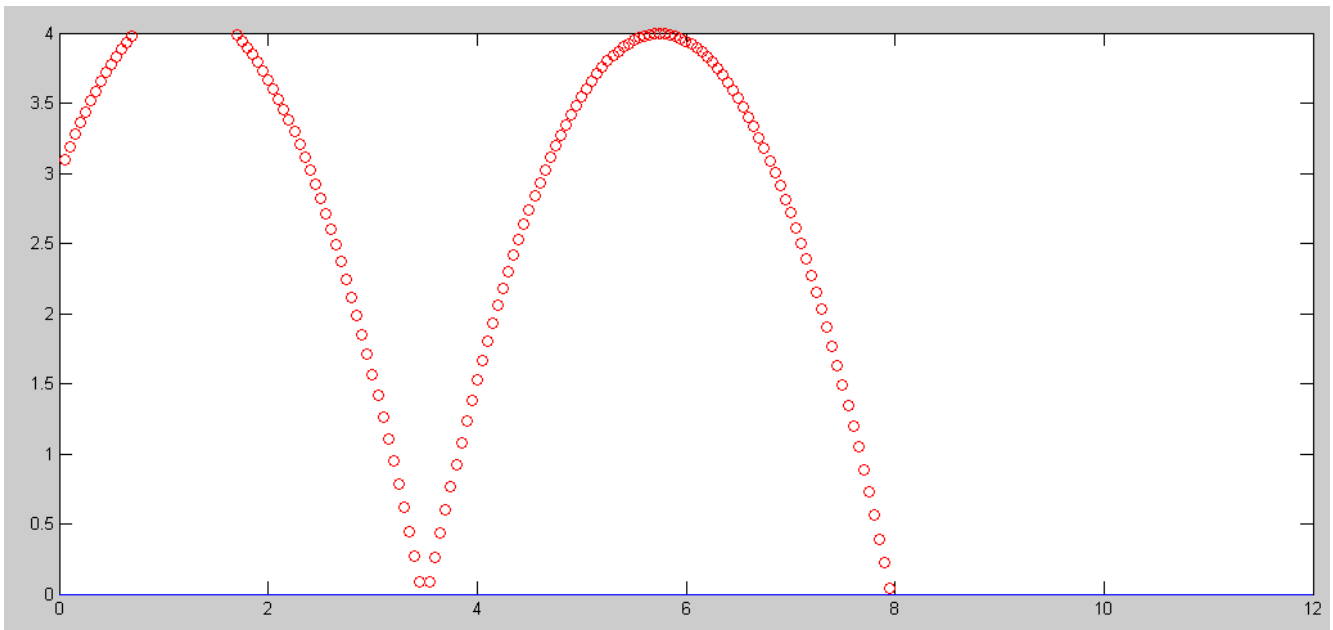
    x = x + dx * dt;
    y = y + dy * dt;

    t = t + dt;

    if (y < 0)
        dy = abs(dy);
        Bounce = Bounce + 1;
    end

    plot(x,y, 'ro', [0,12], [0,0], 'b', 0,10, 'b+');
    xlim([0,12]);
    ylim([0,4]);
    hold on

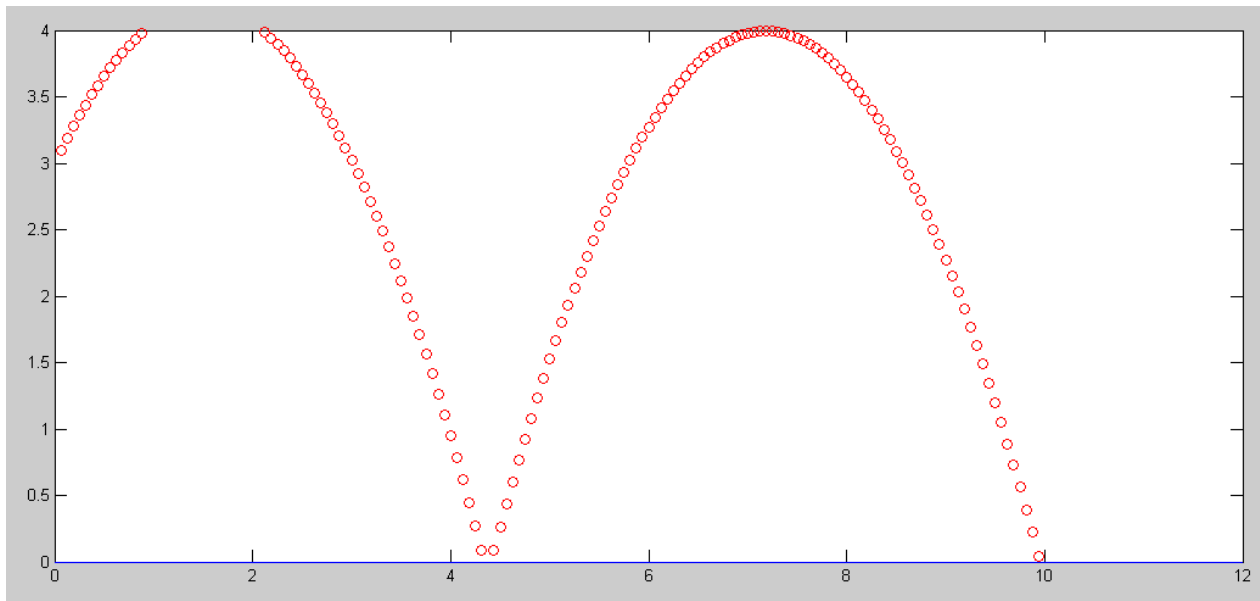
    pause(0.01);
end
```



- 5) Determine the initial velocity on x' so that the ball hits a target at $(x=10, y=0)$ on the second bounce
- note: this is a $f(x) = 0$ problem

This is a linear function and scales

- The 2nd bounce hits at $x = 8.0000$
- Increase the velocity by $10/8$ to get it to land at $x = 10.000$



f(x) = 0: Shoot Game

- Pick a random number from 50 to 100 for your target.
- Pick a random number from 30 to 70 for your firing angle

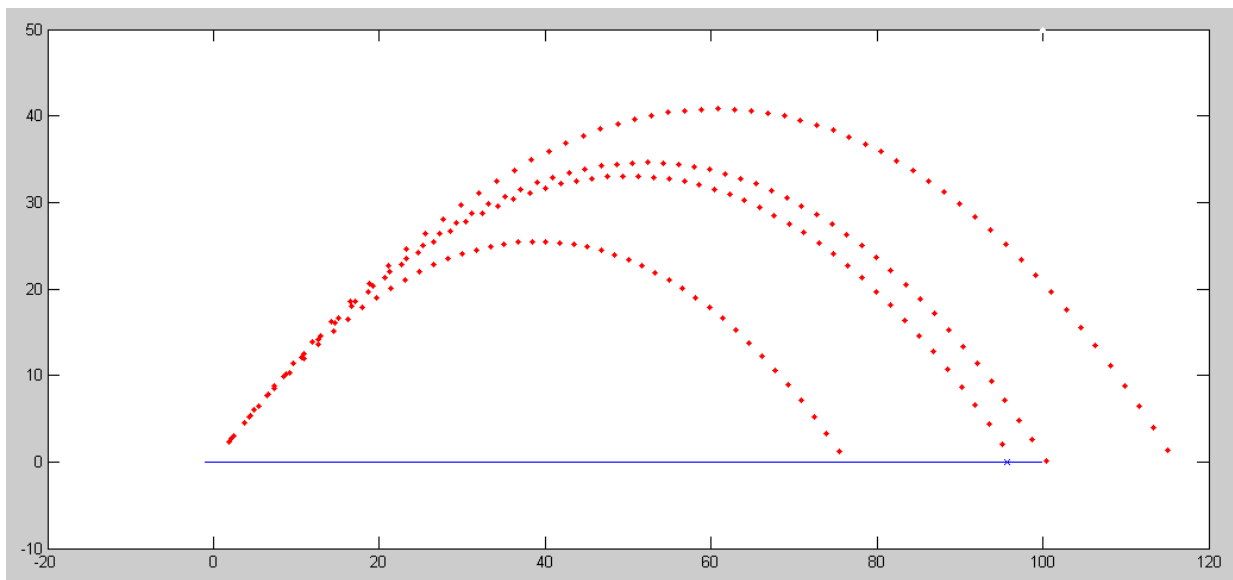
6) Use trial and error to find the initial velocity (X) to fire a tennis ball to hit the target (result is zero)

```
>> Shoot(30, Angle, Target)
ans = -19.3565
```

```
>> Shoot(40, Angle, Target)
ans = 20.2234
```

```
>> Shoot(36, Angle, Target)
ans = 4.8192
```

```
>> Shoot(35, Angle, Target)
ans = 0.8531
```



7) Repeat using California (or Newton's) method to find the initial velocity (X) to fire the tennis ball to hit the target

```
>> X1 = 30;
>> Y1 = Shoot(X1, Angle, Target)

Y1 = -19.3565

>> Y1 = Shoot(X1, Angle, Target)

Y1 = -19.3565

>> X2 = 40;
>> Y2 = Shoot(X2, Angle, Target)

Y2 = 20.2234

>> X3 = X2 - (X2-X1) / (Y2-Y1) * Y2

X3 = 34.8905

>> Y3 = Shoot(X3, Angle, Target)

Y3 = 0.4165

>> X4 = X3 - (X3-X2) / (Y3-Y2) * Y3

X4 = 34.7830

>> Y4 = Shoot(X4, Angle, Target)

Y4 = -0.0122

>>
```

Comment

- Guessing over and over works
- You can find the solution a lot faster if you're willing to do some calculations

