Problem Statement: A 72 g puck is at rest 2.90 meters up a ramp which is elevated 1.60 meters above a horizontal counter. The ramp forms a 38-degree angle above the horizontal. Given that fiction exists between the puck and ramp with a μ value of 0.17, find the distance the puck lands away from the platform. Then create a program/spreadsheet that maximizes the distance landed away from the base of the platform by changing the angle between the ramp and the horizontal.

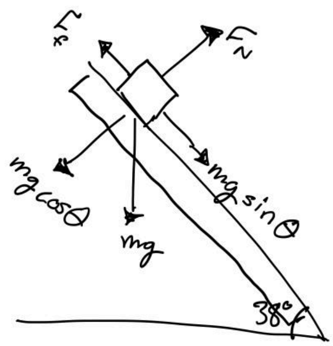
Process:

a.) To find the distance the puck lands away from the base of the platform, our initial approach was to break the problem down into two parts: the dynamics portion with the motion down the inclined ramp and the projectile portion. In the dynamics portion of the problem, we wanted to find the final velocity of the puck after it had completely slid down the ramp. To do this, we needed to identify the acceleration of the puck using the equation of Newton's 2nd Law represented as F = ma. We identified the forces parallel to motion as the component of gravity alongside motion and the force of friction. Since this force corresponds with the direction of acceleration, we can give this force a positive value. The force of friction, which will be expressed negatively in the F = ma equation, can be determined by multiplying the coefficient of μ and the normal force. Since the puck is on an incline, normal force will be a reflection of the vertical component of gravity and not gravity itself. Thus, we can calculate the normal force to be equal to mg*cos(θ). In our F = ma equation, we can substitute F for (mg*sin*θ-μ*mg*cos(θ)), since the net force parallel to motion would be the force of friction subtracted from the horizontal component of gravity. Once we know F, we can isolate a (a=F/m) since we know the mass of the puck. After finding acceleration, we can use the no-t kinematic equation ($V^2 = V_0^2 + 2a\Delta x$) to isolate V since we know the values for x (length of ramp), V_0 (0 because we start at rest), and acceleration. This will result in the final velocity of this phase. We can now move into the projectile motion portion of the problem to find the horizontal distance the puck lands from the base of the platform. To find the initial velocity of the drop of the puck, we can use the V we just found in the previous step and multiply it by sin θ. Now we can use the no v equation ($\Delta y = V_0 * t + ½ * t^2$) to find the time it took for the puck to reach zero. After this, we can now just plug the time and horizontal velocity (v * cos θ) values into Δx = vt to find the distance the puck landed from the base of the counter.

b.) To find the most optimal angle for distance in this problem, we elected to write a program in Python using the numpy library to handle the math. Our approach was to make a function that could convert from a given angle theta to the final distance, and then call that function repeatedly to find all of the values of angles and distance, and then solve for X_{BC}. We can then start by initializing the constants m, L, g, h, and mu (μ) and solving the problem normally as we would on paper. We had to do some reordering to make sure that we could assign the values regularly within the program. For example, to solve for the second t, we had to use the quadratic formula to directly put it in a form such that "t = …". When we get the final distance, we can make the function return that as a result. We can then create a for loop to go through every angle combination from 10 - 90 at a precision to the hundredths place, creating a values list and a values dictionary. The list contains the distance values while the dictionary contains the angle input and the distance. We then find the match for the distance and the input value and return both.

Solution:

Dynamics portion:



Givens:

$m = 7.2g = 0.072\ kg$

$X = 2.9$ meters

1) solve for acceleration

$F = ma$

$F_a - F_f = ma$

$F_a = mg\cos\theta$

$F_f = \mu F_N \quad (F_N = mg\cos\theta)$

$F_f = \mu mg\cos\theta = (0.17)(0.072)(9.8)\cos(38)$

$F_f = 0.0945\ N$

$F_a = (0.072)(9.8)\sin(38)$

$F_a = 0.4344\ N$

$F_a - F_f = ma$

$0.4344 - 0.0945 = 0.072a$

$\boxed{a = 4.72\ m/s^2}$

2) Find final velocity

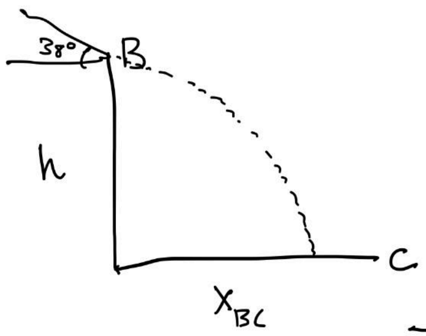
$V^2 = V_0^2 + 2a\Delta x$

$V^2 = (0)^2 + 2(4.72)(2.9)$

$V^2 = 27.376$

$\boxed{V = 5.23\ m/s}$

Projectile motion:



Givens:

$h = 1.6$ meters

$g = 9.8\ m/s^2$

$\theta = 38°$

$V_0 = 5.23\ m/s$

H	V
$\Delta x = V_{0x} t$	$\Delta y = V_{0y} t - \frac{1}{2} g t^2$
$\Delta x = V \cos\theta\, t$	$\Delta y = V \sin\theta\, t - \frac{1}{2} g t^2$
$\Delta x = 5.23 \cos(38)(0.33)$	$\Delta y = (5.23) \sin(38) t - 4.9 t^2$
$\Delta x = 1.36 m$	$-1.6 = 3.22t - 4.9t^2$
	$t = 0.33$

The puck lands 1.36 meters away from the base of the platform.

Part b code:

```python
import numpy as np

def thetaToDistance(theta):...

# create vals list and dictionary
vals = []
valsdict = {}

# test the function across angles 10.00 (angles below return nan) and 90.00
for i in range(1000, 9001):
    i = i / 100
    vals.append(thetaToDistance(i))
    valsdict[i] = thetaToDistance(i)

# prints max distance value in vals list and finds matching key in dictionary
print("The max distance is " + str(max(vals)))
print("The angle is: " + str(list(valsdict.keys())[list(valsdict.values()).index(max(vals))]))
```

```python
def thetaToDistance(theta):  2 usages
    # initialize constants
    m = 0.072
    L = 2.9
    h = 1.6
    g = 9.8
    mu = 0.17
    # configure theta so that it can be used with numpy package
    theta = np.radians(theta)
    # solve for acceleration
    mgCosTheta = m*g*np.cos(theta)
    mgSinTheta = m*g*np.sin(theta)
    Fn = mgCosTheta
    Ff = mu*Fn
    Fnet = mgSinTheta - Ff
    a = Fnet / m
    # get time to go down the ramp
    t_1 = np.sqrt(L/(a/2))
    # get final velocity off of the ramp
    v = t_1*a
    # solve for velocity components (y is negative because it is in the downwards direction)
    v_0x = v*np.cos(theta)
    v_0y = -(v*np.sin(theta))
    # solve for the time to hit the ground using the no v equation and quadratic formula
    t = (-v_0y - np.sqrt(v_0y ** 2 - (4 * (-4.9) * h))) / (2 * (-4.9))
    if t <= 0:
        # conditional logic to ensure that t is positive
        t = (-v_0y + np.sqrt(v_0y ** 2 - (4 * (-4.9) * h))) / (2 * (-4.9))
    # time * velocity = the distance traveled
    final = t * v_0x
    return final
```

Output:
The max distance is 1.5178430764124302
The angle is: 26.65

Extensions:

1. The puck is launched with an angle of 45 degrees from the horizontal from the counter (y=0) with a velocity of 7 m/s onto the ramp. Assuming the ramp is infinitely long, where does the puck land on the ramp? Also, solve the Xbc value when it lands on the counter again.
2. What happens if L is variable from a certain range?
3. How would the optimal angle change if μ was variable with L. $\mu(L) = 0.39L^2 + 0.079$