

**Will Legg**

**04-11-2024**

**REFLECTION PROMPT:** This prompt is a bit different and part of a prompt from a previous week, as it is often constructive. We're including a weird, and hopefully fun spin on it at the end. Consider a member of your current group,

**1. who is someone that you think is contributing positively as a group member, and describe what it is they are doing.** Eleazer from our group is very strong with his calculation skills and also has many ideas about how we can solve any problem. He always begins thinking out his ideas and talking with us in the group to try to make his approach as direct and thorough as possible. Nicholas and Mike are very outgoing and positive during our solving process and they definitely encourage discussion within our group.

**2. What have you learned from them that you think you will apply the next time you are working in groups in another class.** Moving forward I think I will always try to encourage as much discussion as possible within a group and try my best to make sure that everyone is at the same level of understanding. I'll also try my best to balance between my brainstorming vs. actual generation of plan, and not try to move directly from brainstorming to solving. All of my groupmates have been great fun to work with and they're all great problem solvers. I think one of the reasons we've found some success as a group is because they're easy to talk to and are open to having a laugh throughout our time we've spent in the trenches of the boartiger physics realm.

**Finally, the weird bit, 3. what is their fictional origin story that has resulted in them becoming excellent at contributing to a group in this way (bitten by radioactive spider, struck by lightening...)?**

I have a feeling that my groupmates were probably exposed to a radioactive green ooze in the sewers of New York City and taken in as pupils of a very knowledgeable rat, maybe named "*Springer*" or something similar. Instead of learning the ways of combat, they learned how to become some of the best neo-physicians that I've met so far.

PASTE IMAGES OF YOUR TUESDAY WHITEBOARDS HERE:

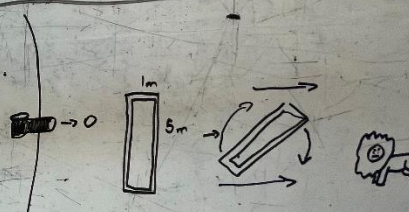
Notes  
 Block:  $5 \times 1 \times 1$  m, wall thickness  $10 \text{ cm}$   
 steel density:  $7850 \text{ kg/m}^3$   
 cannonballs:  $0.3 \text{ m}$  diameter,  $1600 \text{ m/s}$   
 $m = 200 \text{ kg}$

Plan  
 1. Find the volume of the prism  
 1.2 calculate its weight  
 2. Find moment of inertia of the prism  
 3. calculate linear velocity of the cube  
 4. Find angular speed  
 4.1 solve for  $\omega$

Equations  
 $m_1 v_1 + m_2 v_2 = (m_1 + m_2) v_f$   
 $\Delta L_{\text{tot}, A} = \tau_{\text{tot}, A} \Delta t$   
 $m \cdot v = p$   
 $\text{Density}_{\text{steel}} = \frac{m}{V_{\text{steel}}}$   
 $L_i = L_f$

Assumptions  
 • System: cannonball + cube  
 • no external force  
 • net torque  $= 0$   
 $I_{\text{cm}} = \frac{1}{12} (a^2 + b^2) m$   
 • instantaneous collision  
 • weight of cannonball after stick is negligible to Center of mass

Tasking  
 - Volume & mass of steel  
 - how fast linearly the rectangular prism will go  
 - the angular velocity of the block



1. Find Volume of steel prism  
 $V_{\text{big}} - V_{\text{small}} = V_{\text{real}}$   
 $V_{\text{big}} = L \cdot w \cdot h$   
 $V_{\text{small}} = (L - 20 \text{ cm})(w - 20 \text{ cm})(h - 20 \text{ cm})$   
 $V_{\text{big}} = 5 \cdot 1 \cdot 1 = 5 \text{ m}^3$   
 $V_{\text{small}} = 4.8 \cdot 0.8 \cdot 0.8 = 3.072 \text{ m}^3$   
 $V_{\text{real}} = 5 - 3.072 = 1.928 \text{ m}^3$   
 $V_{\text{real}} = 1.928 \text{ m}^3$   $m_{\text{steel}} = 15,134.8 \text{ kg}$   
 $I = 32742062$   $v_{\text{fr}} = \text{m/s}$

1.2 Find Mass of prism  
 $D = \frac{m}{V}$   
 $m = DV$   $m =$   
 $m = 7850 \cdot 1.928$   
 $m = 15134.8 \text{ kg}$

2 Find moment of Inertia of prism  
 $I = \frac{(a^2 + b^2)}{12} \cdot m_w - \left( \frac{(a - 0.2)^2 + (b - 0.2)^2}{12} \right) \cdot m_h$   
 $m_w = 5 \cdot 1 \cdot 1 \cdot 7850 = 39,250$   
 $m_h = 4.8 \cdot 0.8 \cdot 0.8 \cdot 7850 = 24,115.2$   
 $I = \frac{(5^2 + 1^2)}{12} \times 39,250 - \left( \frac{4.8^2 + 0.8^2}{12} \right) \cdot 24,115.2$   
 $I = 37,454.34$

Find Velocity of rectangular prism  
 $m_1 v_1 + m_2 v_2 = (m_1 + m_2) v_f$   
 $200 \cdot 1600 + 0 = (200 + 15,134.8) v_f$   
 $v_f = 20.87 \text{ m/s}$



4. Love everyone

5. Solve for angular speed

$$L_i = L_f \quad r = 2.5 \text{ m}$$

$$L_i = L_r + L_b \quad L_b = r \cdot m_b \cdot v_b$$

$$L_f = L_c$$

$$L_i = 0 + (r \cdot m_b \cdot v_b)$$

$$L_i = 2.5 \cdot 200 \cdot 1600$$

$$L_i = 800000 \text{ kg m}^2/\text{s}$$

$$800000 = L_f = I \cdot \omega$$

$$\omega = \frac{L_i}{I} = \frac{800,000}{37,454.34}$$

$$\omega = 21.359 \text{ rad/s}$$

6. Find total speed when hitting bullseye

$$v = \omega r$$

$$v = 21.359 \cdot 2.5$$

$$v = 53.398 \text{ m/s}$$

$$v_f = v_r + v_i$$


$$v_r = 53.398 \quad v_i = 20.89$$

$$v_f = 53.398 + 20.89$$

$$v_f = 74.29 \text{ m/s @ the fastest point}$$

Possible Improvements ☺

- take into account air resistance and friction on ice
- Assume cannons orbit accurate
- Don't assume maximum elastic collision
- Weight of cannonball's change the center of mass



PASTE IMAGES OF YOUR THURSDAY WHITEBOARDS HERE:

Facts:

- initial velocity ball: 1700 m/s
- $m_{\text{ball}} = 400 \text{ kg}$
- $m_{\text{cannon}} = 1700 \text{ kg}$
- $I_{\text{cannon}} = 33 \text{ kg m}^2$

Plan

- update glowscrip with given values.
- update the pos. of block and ball

Lacking

- paths that cannonball & block trace.
- $\Phi_b(\text{cannonball})$

Assumptions

- cannonball only strikes block 1
- no friction or air resistance
- ball strikes block perpendicular

equations / pseudo code

$$I_{\text{cm}} = \frac{2}{5} m r^2 - \text{sphere}(I)$$

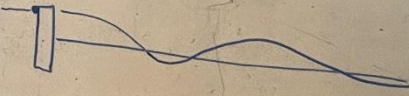
$$I_b = \frac{m^2 x^2}{12} + m$$

$$KE = \frac{1}{2} m v^2$$

$$KE_{\text{rot}} = \frac{1}{2} I \omega^2$$

$$\omega = \frac{v}{r}$$

representation



```

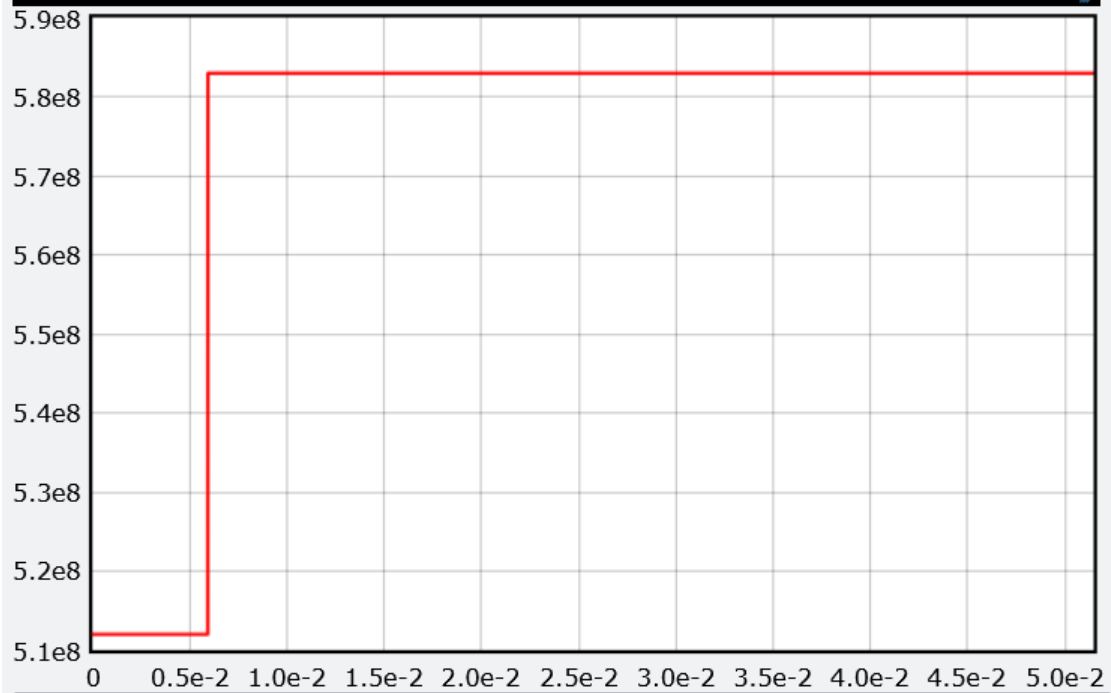
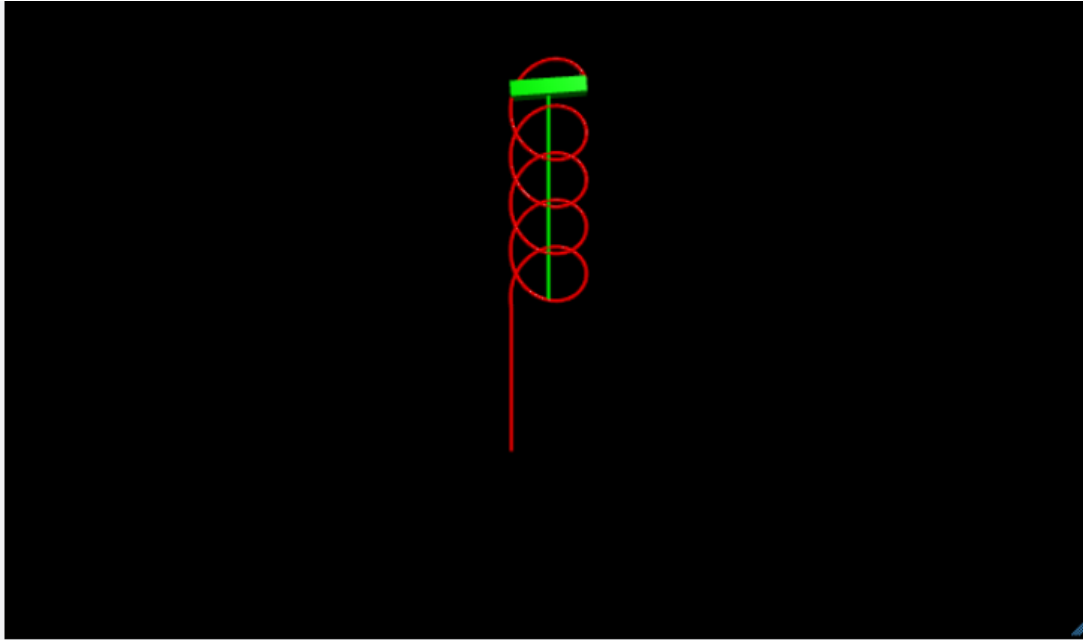
1 Web VPython 3.2
2 # GlowScript 2.9 VPython
3
4 get_library('https://cdn.rawgit.com/PERLMSU/physutil/master/js/physutil.js')
5
6
7 #Window setup
8 scene.center = vec(0,0,0)
9 scene.range = 5
10
11 #Objects
12 block = box(pos=vec(0,0,0), length=5, width=1, height=1, color=color.green,make_trail=True)
13 ball = sphere(pos=vector(-2.45,-10,0), radius=0.15, color=color.red,make_trail=True)
14
15 #Parameters and Initial Conditions
16 mball = 400
17 mblock = 1700
18 mtot = mball + mblock
19
20 vball = vector(0,1600,0)
21 # total translational velocity (block + ball) using conservation of momentum
22 vtrans = (mball*vball)/(mball + mblock)
23
24 pball = mball*vball
25 ptot = pball
26
27 lball = cross(ball.pos,pball)
28 ltot = lball
29
30 # moment of inertia for block
31 Iblock = 33
32
33 #Time and time step
34 t = 0
35 tf = 10
36 dt = 0.00001
37
38 #Graph
39
40 Egraph = PhysGraph(numPlots = 1)
41 E = 0
42
43 print("Position on x-axis where ball collides w/ block: " + ball.pos.x + " m" + "\n")
44

```

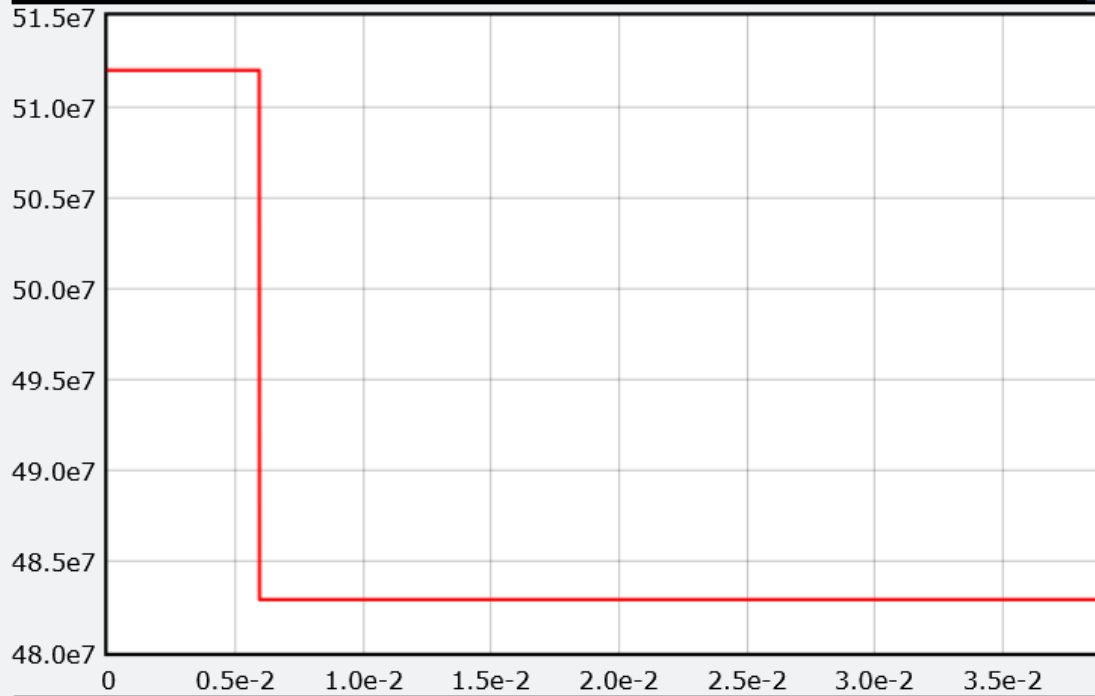
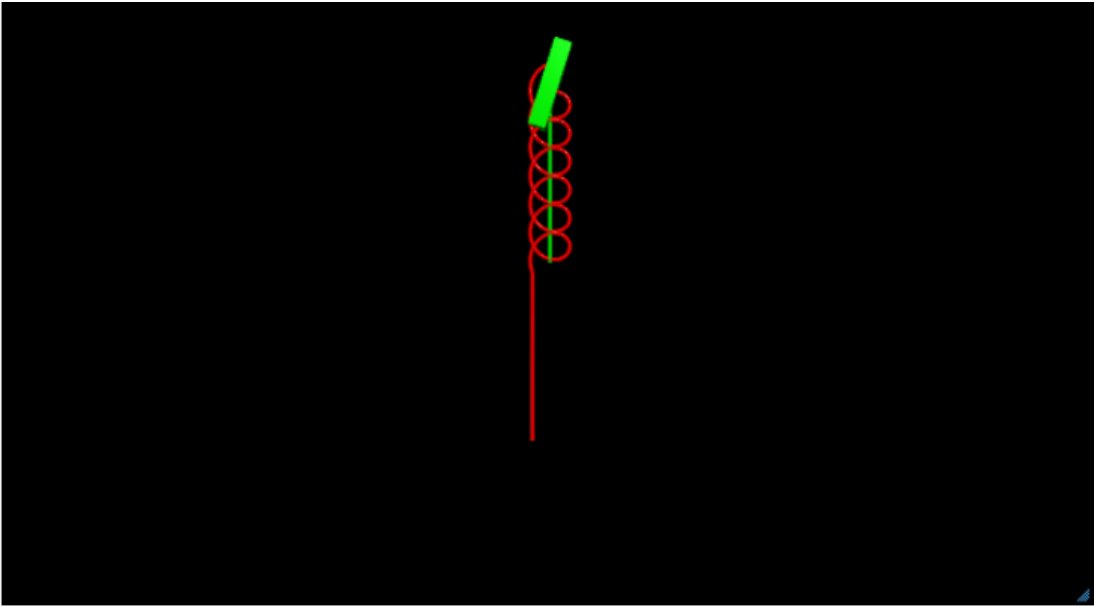
```

45 #Calculation Loop
46 while t<tf:
47     rate(500)
48
49     if ball.pos.y<-0.5 or abs(ball.pos.x)>2.5 or abs(ball.pos.z)>0.5:
50
51         ball.pos = ball.pos + (pball/mball)*dt
52
53
54         # calculate Energy of system prior to collison
55         KEb = (1/2)*(mball)*(mag(vball)**2)
56
57         Egraph.plot(t,KEb)
58
59
60     else:
61         # update position of block/ball based on translational velocity
62         block.pos = block.pos + (vtrans*dt)
63         ball.pos = ball.pos + (vtrans)*dt
64
65         # calculate moment of inertia for ball/block
66         Iball = mball * (mag(block.pos - ball.pos)**2)
67         # Total moment of inertia
68         Itot = Iball + Iblock
69
70         # update position of ball block based on rotational velocity
71         block.rotate(angle=(mag(Ltot)/(Itot))*dt,axis=Ltot/mag(Ltot),origin=block.pos)
72         ball.rotate(angle=(mag(Ltot)/(Itot))*dt,axis=Ltot/mag(Ltot),origin=block.pos)
73
74         # calculate energy of system after collision
75         KEtrans = (1/2)*(mtot)*(mag(vtrans)**2)
76         KErot = (1/2)*((mag(Ltot)**2)/(Itot))
77         KE = KEtrans + KErot
78         Egraph.plot(t, KE)
79         t = t + dt
80
81
82 print("KE of system before collision: " + KEb + " J" + "\n")
83 print("KE of system after collision: " + KE + " J" + "\n")
84

```



Position on x-axis where ball collides w/ block: -2.45 m



Position on x-axis where ball collides w/ block: -1 m