Homework 1

Luke Verlangieri

October 1, 2025

1. Vector Algebra:

```
a.
#returns a float(scalar placeholder)
def dot_product(vector_a, vector_b) -> float:
    # Vectors must be same length
    if len(vector_a) != len(vector_b): raise IndexError
    product = 0 #init sum
    #add the product of all vars
    for var in range(len(vector_a)): product += vector_a[var] * vector_b[var]
    return product #return the value

#indexes
# - 0 : x
# - 1 : y
# - 2 : z
def cross_product(vector_a, vector_b) -> list: #returns a list vector
    #Vectors must be of R_3 and same length
    if len(vector_a) != len(vector_b) or len(vector_b) != 3 : raise IndexError
    return  [
        vector_a[1]*vector_b[2] - vector_a[2]*vector_b[1], #(a_y*b_z - a_z*b_y)
        vector_b[0]*vector_a[2] - vector_a[0]*vector_b[2], #(b_x*a_z - a_x*b_z)
        vector_a[0]*vector_b[1] - vector_b[0]*vector_a[1]  #(a_x*b_y - b_x*a_y)
    ]
b.
#returns a scalar
#performs a dot product operation of A on the resulting cross product of B X C
def scalar_triple_product(a,b,c) -> float:
    return dot_product(a, cross_product(b,c))

#returns a vector
#performs a cross product operation of A on the resulting cross product of B X C
def vector_triple_product(a,b,c) -> list:
    return cross_product(a, cross_product(b,c))
```

```
c & d & e.

A scalar triple product first creates a plane between the vectors in the
second half of the arguments, and then performs a dot product on that vector.
Because the cross product returns an orthogonal vector to the vectors in the
cross product, the closer the absolute value of scalar triple product of a,b,c
is to 0, the more allinged vector a is with the plane created by vector b & c

A = [6,18,5]
B = [-2,-5,-8]
C = [2,1,12]

print(f"A * (B X C) {abs(vector_functions.scalar_triple_product(A,B,C))}")
print(f"C * (A X B) {abs(vector_functions.scalar_triple_product(C,A,B))}")
print(f"B * (A X C) {abs(vector_functions.scalar_triple_product(B,A,C))}")

Output:
    A * (B X C) 128
    C * (A X B) 128
    B * (A X C) 128
```

When a vector is colinear with another vector the dot product will return a product of their magnitudes, $||A|| * ||BXC|| \approx 1044$, the scalar triple product returns 128 meaning that relative to a orthogonal vector to the B C plane created by the cross product, vector A is close to the plane created by vectors B and C. Additionally the scalar triple product is cyclic, meaning that the vectors can be in any orientation and the value will be the same. Qualitatively the vectors are always the same distance and angle apart and therefore it doesnt matter what vectors you choose to be the plane, the scalar triple product will always be the same.

Luke Verlorgior | AA310 HW

given $t = 22s$

$g = 9.81 \frac{m}{s^2}$

a) projectile motion — Ball Released From Rest 22 s of Free Fall

$Y_f = Y_i + V_{iy}\Delta t - \frac{1}{2}g\Delta t^2$

$Y_f = 0 \quad V_{iy} = 0$

$$Y_i = \frac{1}{2}(9.81 \frac{m}{s^2})(22s)^2 = 2274.02\ m$$

B) what speed would the vehicle be going after 22s?

Released from Freefall

$V_{iy} = 0$

$V_{fy} = \cancel{V_{iy}}^{0} - gt$

$$V_{fy} = -(9.81 \frac{m}{s^2})(22s) = -215.8 \frac{m}{s}$$

C) $M = 80{,}000\ kg \quad F_{Tpere} = 14{,}000\ lbf \quad 5\ engines$

Total thrust $= F_{Tpere} \cdot 5\ engines = 42\ kips \Rightarrow 186.9\ kN$

Weight force $= 80{,}000\ kg \cdot 9.81 \frac{m}{s^2} = 784.8\ kN$

$$\Sigma F_y = F_T - W = 186.9 - 784.8 = -597.1\ kN$$

Without lift the plane doesn't generate enough thrust to overcome the force of gravity that's why a plane doesn't take up and down but a rocket can

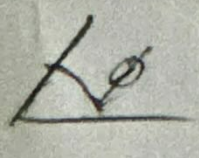
d) Assuming Lift = Thrust · 12 · $\cos\phi$

$\Sigma F_y = F_L - W$

$mg = 12F_T \cos\phi$

$\phi = \cos^{-1}\left(\frac{mg}{12F_T}\right)$

Max angle when forces are equal max lift when the plane ~~horizontal~~ parallel to the horizon earth



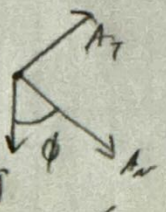
$$\phi = \cos^{-1}\left(\frac{80000 \cdot 9.81}{12 \cdot 186.9kN}\right) = 69.5^\circ$$

e) V_0 to achieve free fall for 22s

$\ddot{x} = 0$ $\quad \dot{x} = V_0 \cos\phi$ $\quad x = V_0 \cos\phi \cdot t + x_0$

$\ddot{y} = -g$ $\quad \dot{y} = V_0 \sin\phi - gt$ $\quad y = (V_0 \sin\phi)t + y_0 - \frac{1}{2}gt^2$

$a_n = \frac{V^2}{\rho}$



Assuming $\Delta y = 0$
→ Starts and stops at same height

$V_0 \sin\phi \cdot t = \frac{1}{2}gt^2 \rightarrow V_0 = \frac{gt}{2\sin\phi} = \frac{9.81 \frac{m}{s^2} \cdot 22s}{2 \cdot \sin(64.5°)}$

$$\rightarrow V_0 = 115.19 \frac{m}{s}$$

plane peaks when $\dot{y} = 0$ $\quad \phi = 64.5°$

$$gt_p = V_0 \sin\phi \rightarrow t_p = \frac{V_0 \sin\phi}{g} = \frac{(115.19)(\sin(64.5°))}{9.81 \frac{m}{s^2}}$$

$t_p = 11s$

$$\Delta h = t\left((V_0 \sin\phi) + -\frac{1}{2}gt\right) \rightarrow (11s)\left(115.19 \frac{m}{s} \sin(64.5) - \frac{1}{2}(9.81)(11)\right)$$

$$\Delta h = 593.5 \text{ m}$$

F) Yes, everything in the aircraft is moving at the same speed as the aircraft. Therefore relative to the plane all objects have a velocity of 0. This is what gives the "no gravity" effect. This is a valid frame for measurements because all objects exist in the same frame of reference undergoing the same external behavior.

Curtis 1.8 determines the required change in flight path angle of the plane to match the free fall motion of the people/items in the plane.

Orbital Speed:

a.

Newton's law of graviation and second law of motion, as well as normal Curtis eqn 1.25

$$F = ma, \quad F_g = \frac{Gm_1m_2}{r^2}, \quad a_n = \frac{v^2}{\rho}$$

All forces acting on the satellite are in the normal direction (pointing toward the center of the earth).

$$\Sigma F_n = F_g = ma_n$$

after doing some algebra with the functions above, we result in

$$v = \sqrt{\frac{GM}{r}}$$

Where G is the graviational constant, M is the mass of the earth, and r is the orbital raduis of the object

b.

```
G = 6.6743 * math.pow(10,-11) # m3/(kg * s^2)
MASS_EARTH = 5.972 * math.pow(10,24) # kg

#returns the velocity in m/s for a object in circular orbit around the earth
def circular_orbit_velocity(orbital_r) -> float:
return math.sqrt((G * MASS_EARTH / orbital_r))
```

c.
Using the code mentioned in part b,

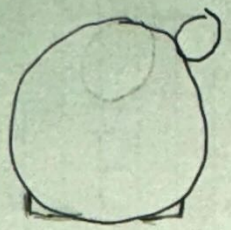
Satellite	Altitude	Orbital Speed
LEO	460 km	7.639 km/s
MEO	20,200 km	3.873 km/s
GEO	35,780 km	3.075 km/s

Table 1: Satellite speeds in orbit

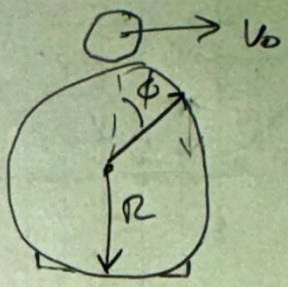
d. Astronaughts wear the mach 25 patch due to the fact that the ISS flies about mach 25 around the earth, its now a staple and memoir of astronaughts.

a)

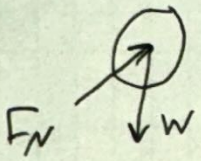
i)



About 60°



ii)



iii) To remain on the cylinder, the normal force must be nonzero

b) $PE_0(\phi = 0) = mgh = 2mgr$

c) $KE_0(\phi = 0) = \frac{1}{2}mv_0^2$

d) $\Delta KE = -\Delta PE \rightarrow -\Delta PE = PE_0 - PE(\phi)$

$-\Delta PE = 2mgr - mgr(1+\cos\phi) = mgr(1-\cos\phi)$

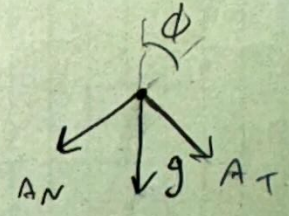
$KE_f = KE_0 + mgr(1-\cos\phi) = \frac{1}{2}mv_0^2 + mgr(1-\cos\phi)$

e) Corb's 625 $\rightarrow a_n = \frac{v_f^2}{\rho}$ $\quad \rho = r$

$\frac{1}{2}mv_f^2 = \frac{1}{2}mv_0^2 + mgr(1-\cos\phi)$

$v_f^2 = v_0^2 + 2gr(1-\cos\phi)$

$a_n = \frac{1}{r}\left(v_0^2 + 2gr(1-\cos\phi)\right)$



$a_n = g\cos\phi \rightarrow gr\cos\phi = v_0^2 + 2gr(1-\cos\phi)$

f.

Resolving the equation on the previous page for angle, we find that

$$\phi = \arccos(\frac{v_0^2 + 2gr}{3gr})$$

When $v_0 = 0$, we find that

$$\phi = \arccos(\frac{2}{3}) = 48.19\,\text{deg}$$

In python:

```
# (v_0^2 + 2gr)/3gr = cos(phi)
v_0 = 0
lhs = 2/3
phi = math.acos(lhs) * 180/math.pi #convert to degrees
print(f'Departure Angle : {round(phi,3)}')

Output:
    Departure Angle : 48.19
```

g.

This time, instead resolving the equation for v_0 we result in

$$v_0 = \sqrt{gr(3\cos(\phi) - 2)}$$

Plugging this into python

```
#v_0 = sqrt(gr(3cos(phi) - 2))

g = 9.81 # m/s
phi = 45 * math.pi/180 #convert to radians
const = math.sqrt(g * (3*math.cos(phi) - 2)) #compute the solved version
print(f'V_0 : {round(const,3)} * r^(1/2)')

Output:
    V_0 : 1.091 * r^(1/2)
```