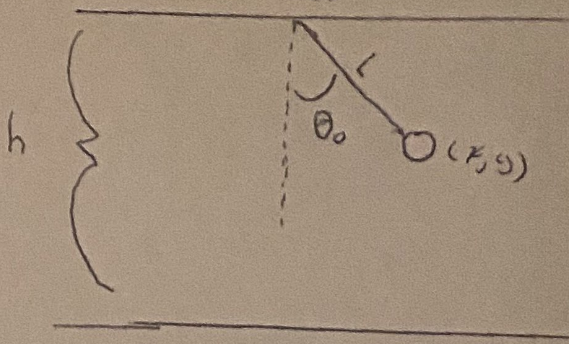
Simple Pendulum Problem

What is the height of the bob after 19.5 seconds?

(not to scale)



First find equation of motion using the Lagrangian:

$h = 5m$, $L = 1m$, $\theta_0 = 40°$

$$\mathcal{L} = \underset{K.E.}{T} - \underset{P.E.}{V} \quad ; \quad T = \frac{1}{2} m(\dot{x}^2 + \dot{y}^2), \quad V = mgy$$

we can simplify the ~~Ag~~ Lagrangian by using θ:

$$x = L\sin\theta, \quad y = -L\cos\theta \quad \rightarrow \quad \dot{x} = L(\cos\theta)\dot{\theta}, \quad \dot{y} = L(\sin\theta)\dot{\theta}$$

$$T = \frac{1}{2} m(L^2 \sin^2\theta\, \dot{\theta}^2 + L^2\cos^2\theta\, \dot{\theta}^2) = \frac{1}{2} m L^2 \dot{\theta}^2, \quad V = -mgL\cos\theta$$

so $$\mathcal{L}(\theta, \dot{\theta}) = \frac{1}{2} m L^2 \dot{\theta}^2 + mgL\cos\theta$$

using Euler-Lagrange $$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}}\right) - \frac{\partial \mathcal{L}}{\partial \theta} = 0$$

$$\frac{d}{dt}(mL^2\dot{\theta}) + mgL\sin\theta = 0 \quad \rightarrow \quad mL^2\ddot{\theta} + mgL\sin\theta = 0$$

$$L\ddot{\theta} + g\sin\theta = 0$$

$$\boxed{\ddot{\theta} = -\frac{g}{L}\sin\theta}$$

$\theta_0 = 40°$ is too large for the small angle approximation

But, we can accurately solve the coupled ODE

System using a ~~Rung~~ Runge-Kutte 4th ordor scheme

Using derived equation of motion I will set up the RK4 scheme:

- Equation of motion:

$$\ddot{\theta} = -\frac{g}{L}\sin\theta$$

- Define state vectors from coupled ODE

$$q = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}, \quad q' = f(q) = \begin{bmatrix} \dot{\theta} \\ -\frac{g}{L}\sin\theta \end{bmatrix}$$

- RK4 step (from $t_n \rightarrow t_{n+1} = t_n + h$)
  - $k_1 = f(q_n)$
  - $k_2 = f(q_n + \frac{h}{2}k_1)$
  - $k_3 = f(q_n + \frac{h}{2}k_2)$
  - $k_4 = f(q_n + h k_3)$

- Written out with $q = [\theta, \dot{\theta}]^T$
  - $k_1^\theta = \dot{\theta}_n$;
  - $k_2^\theta = \dot{\theta}_n + \frac{h}{2}k_1^{\dot{\theta}}$;
  - $k_3^\theta = \dot{\theta}_n + \frac{h}{2}k_2^{\dot{\theta}}$;
  - $k_3^\theta = \dot{\theta}_n + h k_3^{\dot{\theta}}$;
  - $k_1^{\dot{\theta}} = -\frac{g}{L}\sin\theta_2$
  - $k_2^{\dot{\theta}} = -\frac{g}{L}\sin(\theta_n + \frac{h}{2}k_1^\theta)$
  - $k_3^{\dot{\theta}} = -\frac{g}{L}\sin(\theta_n + \frac{h}{2}k_2^\theta)$
  - $k_4^{\dot{\theta}} = -\frac{g}{L}\sin(\theta_n + h k_3^\theta)$

And update $\theta, \dot{\theta}$ with:

$$\theta_{n+1} = \theta_n + \frac{h}{6}(k_1^\theta + k_2^\theta + k_3^\theta + k_4^\theta)$$

$$\dot{\theta}_{n+1} = \dot{\theta}_n + \frac{h}{6}(k_1^{\dot{\theta}} + k_2^{\dot{\theta}} + k_3^{\dot{\theta}} + k_4^{\dot{\theta}})$$

This has been implemented in the attached python code with

~~h = 0.0005~~ $h = 0.005$ seconds, $t_{final} = 18.5$ seconds, $nt = t_{final}/h$,

$\theta_0 = 40°$, $g = 9.81 m/s^2$

yielding a final $y_{18.5} = -0.7944$. Adding 5m to account for the cieling height gives final answer $\boxed{4.21\,m}$

In [1]:
```
import numpy as np

g=9.81 # m/s^2
pendulum_length = 1 # meters
theta_0 = 40 # degrees
t_final = 18.5 # seconds
dt=0.005 # seconds
n_steps=int(t_final/dt)

def pendulum_rk4(dt, n_steps, initial_theta_degrees, l):
    ''' RK4 integrator to solve the Simple Pendulum ODE system:
        d^2 theta / d theta ^2 = -(g/l)sin(theta)
        returns the x,y position, time, theta and d theta/ dt arrays with n_
    '''
    theta_old = initial_theta_degrees*(np.pi/180)

    theta_dot_old = 0

    pos_x = np.zeros(n_steps+1)
    pos_y = np.zeros(n_steps+1)
    times = np.zeros(n_steps+1)
    thetas = np.zeros(n_steps+1)
    theta_dots = np.zeros(n_steps+1)

    ## initial conditions
    pos_x[0] = np.sin(theta_old)*l
    pos_y[0] = -np.cos(theta_old)*l
    times[0] = 0
    thetas[0] = theta_old


    t_new = 0
    for i in range(n_steps):
        t_new += dt


        k1_theta = theta_dot_old
        k1_theta_dot = -(g/l)*np.sin(theta_old) ## use ODE equation for k_th

        k2_theta = theta_dot_old + k1_theta_dot*dt/2
        k2_theta_dot = -(g/l)*np.sin(theta_old + k1_theta*dt/2)

        k3_theta = theta_dot_old + k2_theta_dot*dt/2
        k3_theta_dot = -(g/l)*np.sin(theta_old + k2_theta*dt/2)


        k4_theta = theta_dot_old + k3_theta_dot*dt
        k4_theta_dot =  -(g/l)*np.sin(theta_old + k3_theta*dt)

        ## use k1,k2,k3,k4 to update theta and dtheta/ dt
        theta_new = theta_old + dt * (k1_theta + 2*k2_theta + 2*k3_theta + k
        theta_dot_new = theta_dot_old + dt * (k1_theta_dot + 2*k2_theta_dot
```

```
        ### update arrays
        pos_x[i+1] = np.sin(theta_new)*l
        pos_y[i+1] = -np.cos(theta_new)*l
        times[i+1] = times[i]+dt
        thetas[i+1] = theta_new
        theta_dots[i+1] = theta_dot_new
        ###

        theta_old = theta_new
        theta_dot_old = theta_dot_new

    return pos_x, pos_y, times, thetas, theta_dots



pos_x, pos_y, times, thetas, theta_dots = pendulum_rk4(dt, n_steps, theta_0,

print(f"After {round(times[-1], 2)} seconds, the pedulum bob has height {rou
```

```
After 18.5 seconds, the pedulum bob has height 4.21 meters
```