<u>Pauta Auxiliar 1</u>

P1

Para resolver este problema, en primer lugar estableceremos las ecuaciones que rigen el movimiento en este caso:

$$\vec{F} = m\ddot{\vec{y}}$$

Segunda ley de Newton

$$\vec{F} = -mg - k\vec{y}$$

Fuerzas externas sobre la caja (gravedad y resorte)

Para resolver este problema de forma numérica, se usará el método de Verlet para la segunda derivada:

$$\ddot{x}(t_i) = \frac{x_{i+1} - 2x_i + x_{i-1}}{\Delta t^2}$$

Segunda derivada discreta

Para aplicar esta fórmula y encontrar la solución numérica de la posición con respecto al tiempo, se reemplazará la aceleración:

$$\ddot{y}_i = -g - \frac{k}{m}y_i$$

Luego de aplicar esta equivalencia se reordenará la ecuación de la segunda derivada discreta para dejar como incógnito el término x_{i-1}

$$y_{i+1} = \left(-g - \frac{k}{m}y_i\right)\Delta t^2 + 2y_i - y_{i-1}$$

Reescribimos la masa con el dato dado en el enunciado:

$$y_{i+1} = (-g - \frac{k}{1+t}y_i)\Delta t^2 + 2y_i - y_{i-1}$$

Para resolver esta ecuación usaremos una serie de arreglos en MATLAB:

Para el cálculo numérico se le deben dar valores a las constantes, estos se pueden cambiar para analizar el impacto que tienen en el movimiento de la caja.

```
g=9.81        % Notar que si definimos el origen donde el resorte tiene largo natural, L no es necesario.
k=2;
Vo=0;         % la caja parte del reposo en este caso en particular

dt=0.0001;
t=0:dt:10;

y=zeros(length(t),1);   % se arma el arreglo de posición del mismo largo que el de tiempo

y(1)=3;
y(2)=(y(1) + Vo.*dt);    % si la caja tuviese velocidad inicial, este termino la consideraria

for i=2:length(y)-1

        y(i+1)=(-g-(k/(1+t(i))).*y(i)).*dt.*dt + 2.*y(i) – y(i-1);
                                              % iteración de la ecuacion escrita mas arriba

        end

plot(t,y)       % grafico de la posición con respecto al tiempo
```

P2

Para este problema, calcularemos las velocidades como la derivada discreta de la posición usando la derivada hacia atrás:

$$\dot{x}(t_i) \approx \frac{x_i - x_{i-1}}{\Delta t}$$

Para la aceleración se usará la derivada centrada de segundo orden:

$$\ddot{x}(t_i) \approx \frac{x_{i+1} - 2x_i + x_{i-1}}{\Delta t^2}$$

Por lo visto en Introducción a la Física, sabemos que podemos separar el desplazamiento y las fuerzas en verticales y horizontales, por lo que:

$$F_x = m\ddot{x} = -\gamma\dot{x}$$
$$F_y = m\ddot{y} = -\gamma\dot{y} - mg$$

Cambiando ahora los términos de aceleración y velocidad por los de arriba y reordenando los términos:

$$x_{i+1} = 2x_i - x_{i-1} - \frac{\gamma}{m}(x_i - x_{i-1})\Delta t$$

$$y_{i+1} = 2y_i - y_{i-1} - \frac{\gamma}{m}(y_i - y_{i-1})\Delta t - g\Delta t^2$$

Ahora para calcular con MATLAB:

```
m=3;
g=9.81;
r=2;        % coeficiente de roce

dt=0.001;
t=1:dt:10;

% condiciones iniciales
x(1)=0;
y(1)=0;

% notar que ahora solo definimos el primer valor de cada arreglo.

A=pi/4; % angulo inicial

Vo=10;  % velocidad inicial

x(2)=x(1) + Vo*cos(A)*dt;
y(2)=y(1) + Vo*sin(A)*dt;


% algoritmo de Verlet

for i=2:(length(t)-1)
        x(i+1)=2.*x(i)-x(i-1)- (r/m).*(x(i)-x(i-1)).*dt;
        y(i+1)=2.*y(i)-y(i-1)- (r/m).*(y(i)-y(i-1)).*dt – g.*dt.*dt;

% podemos ver que los terminos se van agregando a los arreglos a medida que estos se calculan.

        if y(i)<= 0 break

% este if determina cuando se llega al piso

                end     % este end es para el if

        end             % este end es para el for

alcance = x(length(x))
% como los terminos fueron creados a medida que se calculaban, este termino corresponde al ultimo
% valor de x antes de impactar el suelo, es decir: el alcance.

hold on
plot(x,y)
grid on
xlabel('alcance [m]')
ylabel('altura [m]')
title('Proyectil con Roce')
```

P3

En este ejercicio también trabajaremos separadamente las coordenadas:

Simplificaremos el problema suponiendo que el resorte 1 sólo efectúa una fuerza en y mientras que el resorte 2 sólo en x.

$$F_x = m\ddot{x} = -k_2 x$$
$$F_y = m\ddot{y} = -k_1 y - mg$$

Con esto el problema queda similar a los ejercicios anteriores, se usará nuevamente la derivada centrada de segundo orden:

$$x_{i+1} = \frac{(-k_2 x_i)}{m}\Delta t^2 + 2x_i - x_{i-1}$$

$$y_{i+1} = \frac{(-k_2 y_i)}{m}\Delta t^2 - g\Delta t^2 + 2y_i - y_{i-1}$$

Luego con una iteración de MATLAB se puede llegar al resultado.