

I want to congratulate my two friends who contributed the most to these projects: Paulo and Palome

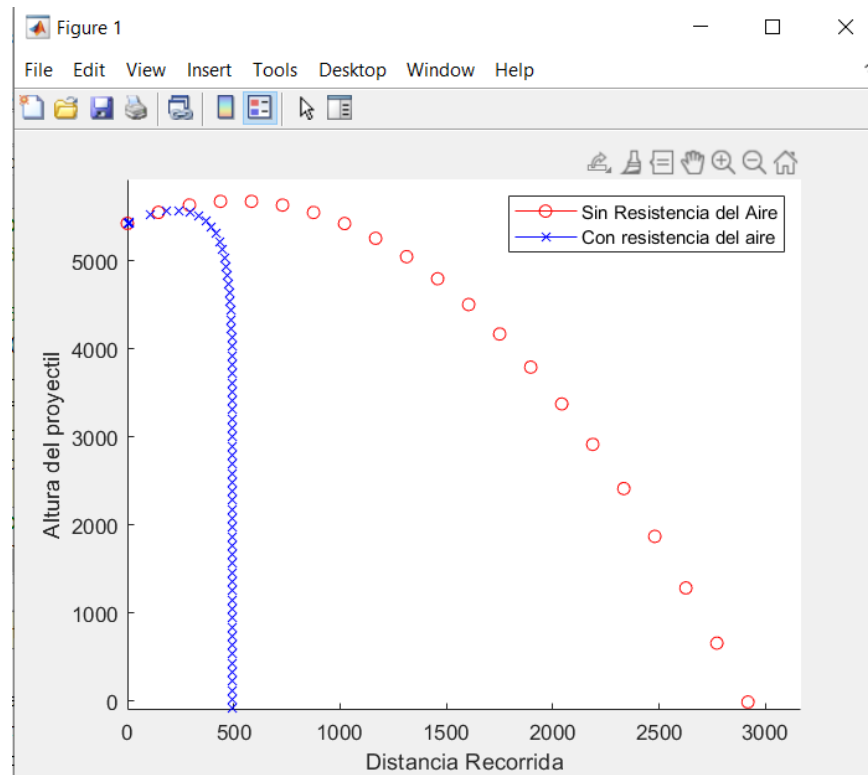
Volcano projectiles simulation file

```
Volcan.m  x  +
%% INGRESO DE DATOS INICIALES
% Mayor parte del credito a Pol y Palomitas
% Paulo Ogando, Cesar Emiliano Palome, Jose Luis Madrigal, Jorge Isidro Blanco, Fernando Emilio
% 45, 100, 5426, 5, 1.5, 250
- angulo = input("Dame el angulo de salida del proyectil en grados: ");
- velocidad_inicial = input("Dame la Velocidad Inicial del Proyectil en m/s: ");
- altura_volcan = input("¿Cuánto mide el Volcan?(m): ");
- resistencia_aire_b = input("Medida de la resistencia del aire: ");
- exponente_velocidad_n = input("Exponente de la Velocidad entre 1.1 y 1.9: ");
- masa_proyectil = input("¿Cuánto pesa el proyectil?(kg): ");
%% CALCULO DE OTROS DATOS RELEVANTES PARA EL CALCULO DE LA TRAYECTORIA
- Voy = velocidad_inicial * sind(angulo);
- Vox = velocidad_inicial * cosd(angulo);
- tiempo_recorrido = ((-(-Voy)+sqrt(Voy^2-(4*(4.9)*-altura_volcan)))/(2*4.9);
- Axj = (-resistencia_aire_b*(Vox^2+Voy^2)^((exponente_velocidad_n-1)/2)*Vox)/masa_proyectil;
- Ayj = ((-resistencia_aire_b*(Vox^2+Voy^2)^((exponente_velocidad_n-1)/2)*Voy)/masa_proyectil)-9.81;
- dt = tiempo_recorrido/5000;
- c = 1;
- t = 0;
- x = 0;
- Verlet = [c t x altura_volcan Vox Voy Axj Ayj];
%% METODO DE VERLET
- for i = 1:20000;
-     Verlet(c+1,:) = [c+1
```

Inputs

```
>> Volcan
Dame el angulo de salida del proyectil en grados: 45
Dame la Velocidad Inicial del Proyectil en m/s: 100
¿Cuánto mide el Volcan?(m): 5426
Medida de la resistencia del aire: 5
Exponente de la Velocidad entre 1.1 y 1.9: 1.5
¿Cuánto pesa el proyectil?(kg): 250
fx |
```

Graph with air friction and without it (remember this is shown as an animation to see trajectory)



Roller coaster car file

```

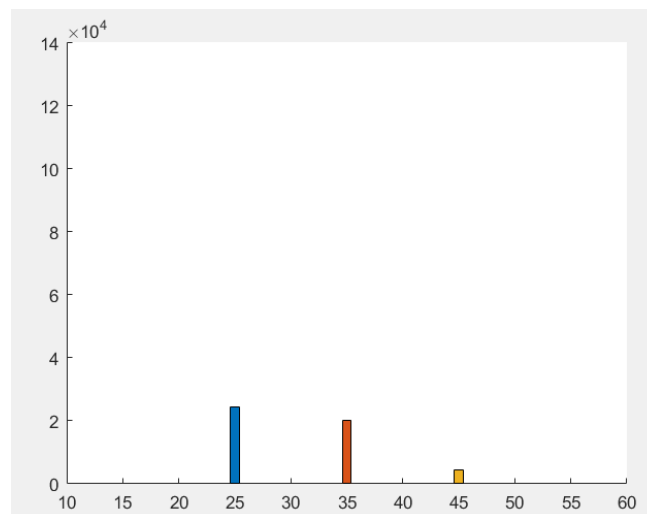
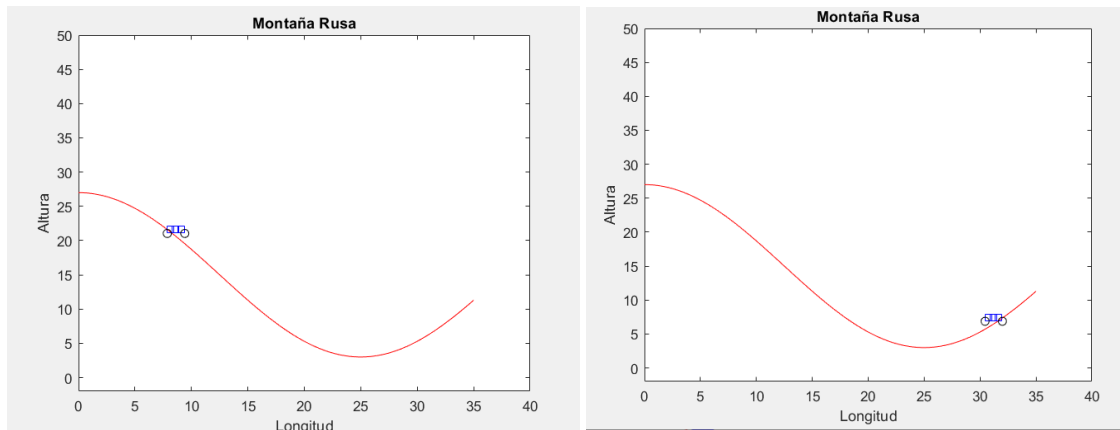
Montana_rusa.m  x  +
1  %% INGRESO DE DATOS INICIALES
2  % Mayor parte del credito a Pol y Palomitas
3  % Paulo Ogando, Cesar Emiliano Palome, Jose Luis Madrigal, Jorge Isidro Blanco, Fernando Emilio
4  % -50, 3, 100, 0.1, 35, 30, 0.03
5  v = input("Dame la Velocidad Inicial del carrito en m/s: ");
6  k = input("Medida de la resistencia del aire: ");
7  m= input("¿Cuánto pesa el carrito?(kg): ");
8  mk = input("Dame el coeficiente de friccion de la pista(menor a .5): ");
9  largo = input("Dame la longitud de la montaña rusa: ");
10 posinicial = input("Dame la posicion inicial del carrito: ");
11 dt = input("Dame delta t: ");
12 %% Grafica Montaña Rusa
13 X = 0:.2:largo;
14 Y = [];
15 for x = 0:.2:largo;
16     paren = (pi*x)/25;
17     y = 12*cos(paren) + 15;
18     Y = [Y y];
19 end
20 hold on
21 plot(X,Y,'r');
22 title("Montaña Rusa");
23 ylabel("Altura");
24 xlabel("Longitud");
25 axis([0 largo+5 -2 50])
26 %% Datos Iniciales para calcular posicion
27 Fun = @(x) (12*cos((pi*x)/25) + 15);
28 FunD = @(x) ((-12*pi*sin((pi*x)/25))/25);
29 Fun2D = @(x) ((-12*pi^2*cos((pi*x)/25))/625);

```

Inputs

```
>> Montana_rusa
Dame la Velocidad Inicial del carrito en m/s: -50
Medida de la resistencia del aire: 3
¿Cuánto pesa el carrito?(kg): 100
Dame el coeficiente de fricción de la pista(menor a .5): 0.1
Dame la longitud de la montaña rusa: 35
Dame la posicion inicial del carrito: 30
Dame delta t: 0.03
```

Simulation (this is an animation as well)



Implementation in App designer

```

% Callbacks that handle component events
methods (Access = private)

% Button pushed function: InicioButton
function InicioButtonPushed(app, event)
    v= app.VelocidadinicialmsEditField_2.Value;
    m= app.PesodelCarritokgEditField.Value;
    k= app.resistenciadelaireEditField.Value;
    mk = app.coeficientedefriccionEditField.Value;
    largo = app.longituddelamontaaEditField.Value;
    posinicial=app.posicioninicialEditField.Value;
    dt = app.DtEditField.Value;
    %% Grafica Montaña Rusa
    X = 0:.2:largo;
    Y = [];
    for x = 0:.2:largo
        paren = (pi*x)/25;
        y = 12*cos(paren) + 15;
        Y = [Y y];
    end
    plot(app.UIAxes, X,Y,'r');
    title(app.UIAxes,"Montaña Rusa");
    ylabel(app.UIAxes,"Altura");
    xlabel(app.UIAxes,"Longitud");
    axis(app.UIAxes,[-10 largo+25 -2 50])
    %% Datos Iniciales para calcular posicion
    Fun = @(x)(12*cos((pi*x)/25) + 15);
    FunD = @(x)((-12*pi*sin((pi*x)/25))/25);
    Fun2D = @(x)((-12*pi^2*cos((pi*x)/25))/625);
    g = 9.81;
    xprima = 0;
    xpos = posinicial;

```

