

Solucion de Reto – Equipo 4

Carlos Marvan A00838096

Luis Gerardo A00836928

David Mireles A00836010

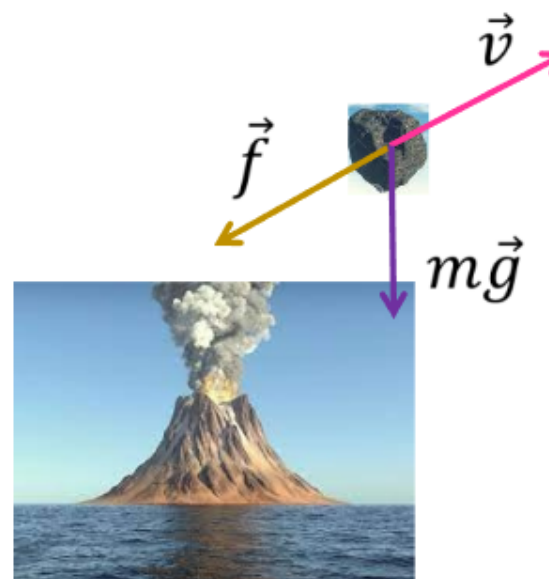
Gerardo Daniel A00836801



PLANTEAMIENTO ECUACION DIFERENCIAL



Ecuaciones de EULER



Valores iniciales

v_o θ x_o y_o

Condiciones iniciales:

Para x :

$$ma_x = -bv_x$$

$$\frac{dv_x}{dt} = -\frac{b}{m}v_x = a_x$$

$$v_{xn+1} = v_{xn} + dt \left(-\frac{b}{m}v_{xn} \right)$$

$$= v_{xn} + a_{xn}dt$$

$$h = dt$$

$$\frac{dx}{dt} = v_x$$

$$x_{n+1} = x_n + dtv_{xn}$$

$$= x_n + v_{xn}dt$$

Para y :

$$ma_y = -bv_y - mg$$

$$\frac{dv_y}{dt} = -\frac{b}{m}v_y - g = a_y$$

$$v_{yn+1} = v_{yn} + dt \left(-\frac{b}{m}v_{yn} - g \right)$$

$$= v_{yn} + a_{yn}dt$$

$$\frac{dy}{dt} = v_y$$

$$y_{n+1} = y_n + dtv_{yn}$$

$$= y_n + v_{yn}dt$$

$$y_o, v_{oy} = v_o \sin \theta$$

y_0
$y_1 = y_0 + h \cdot f(x_0, y_0)$
$y_2 = y_1 + h \cdot f(x_1, y_1)$
$y_3 = y_2 + h \cdot f(x_2, y_2)$
$y_4 = y_3 + h \cdot f(x_3, y_3)$

PLANTEAMIENTO COMPUTACIONAL

1. Investigar sobre el comportamiento de los volcanes y los datos más importantes sobre los proyectiles.
2. Aprender las características de una parabola.
3. Implementar nuestros conocimientos y aplicarlos en excel.
4. Aprendimos sobre como aplicar una parabola tomando en cuenta la friccion del aire utilizando el metodo de euler.
5. Finalmente copiamos lo anterior en un nuevo software conocido como MathLab en cual le agregamos una interfaz con la cual pudimos observar de una mejor manera los resultados obtenidos anteriormente.

PARÁMETRO	MÍNIMO	MÁXIMO
Velocidad inicial	85 m/s	140 m/s
Velocidad de impacto	83.333m/s	138.889m/s
Angulo	25,30,40,45	70
Diámetro	0.06m	3.5m
Masa	0.259kg	78.532kg
Densidad	2300 kg/m3	2800 kg/m3
Viscosidad del aire		17.2 x 10 ⁻⁶

Fuerza de viscosidad = $6 \cdot \pi \cdot \text{viscosidad del aire} \cdot \text{radio}$



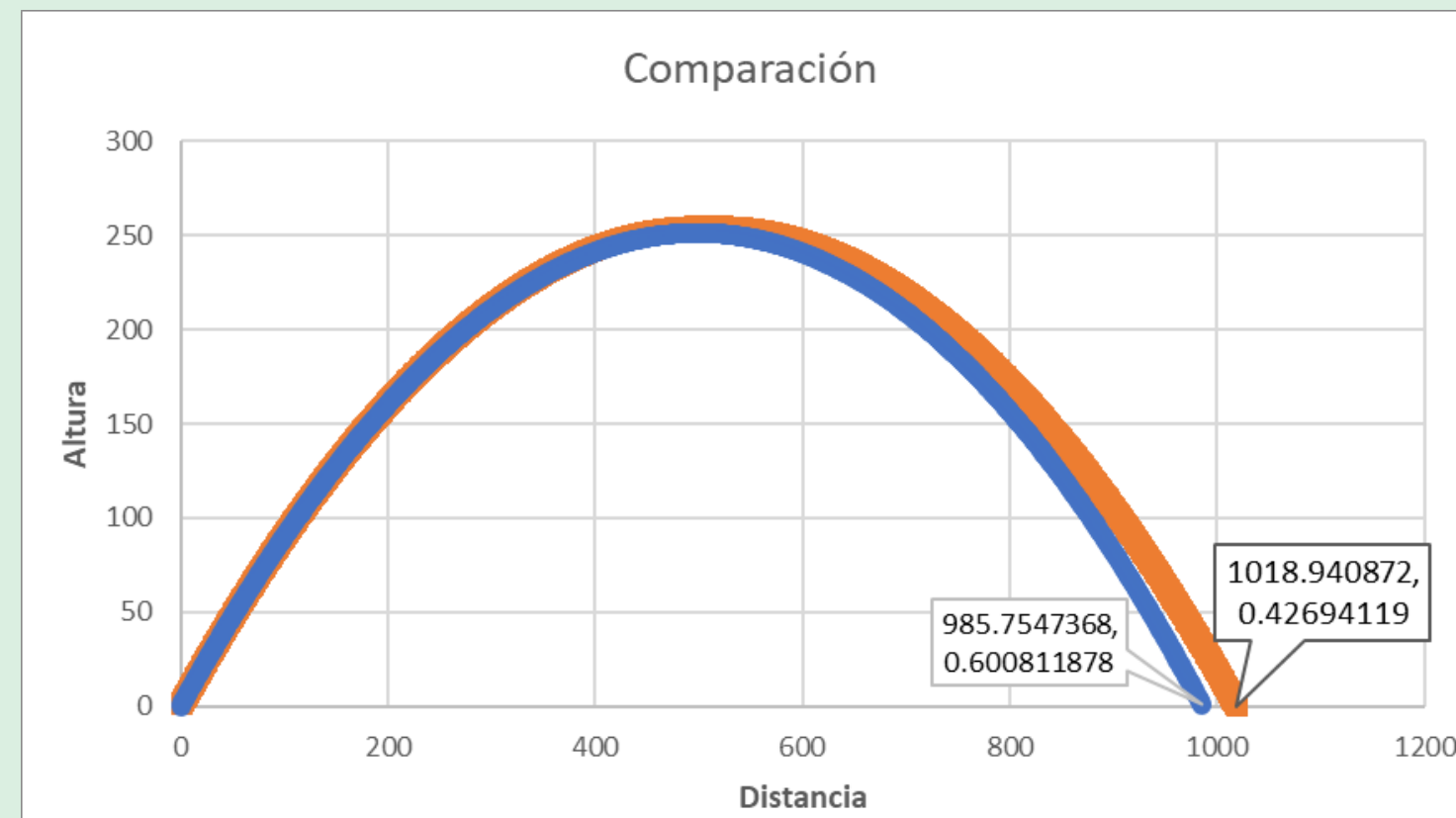
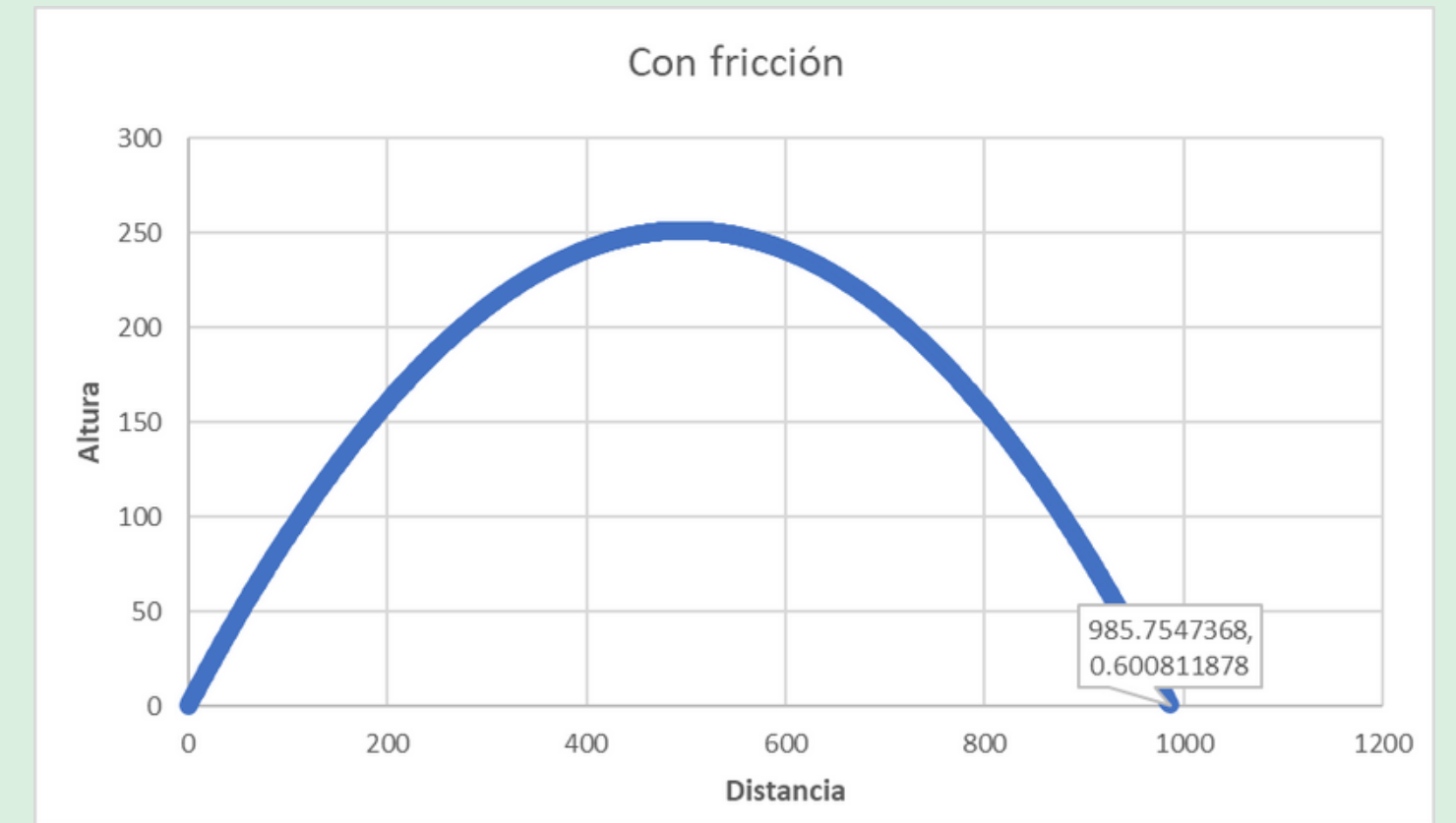
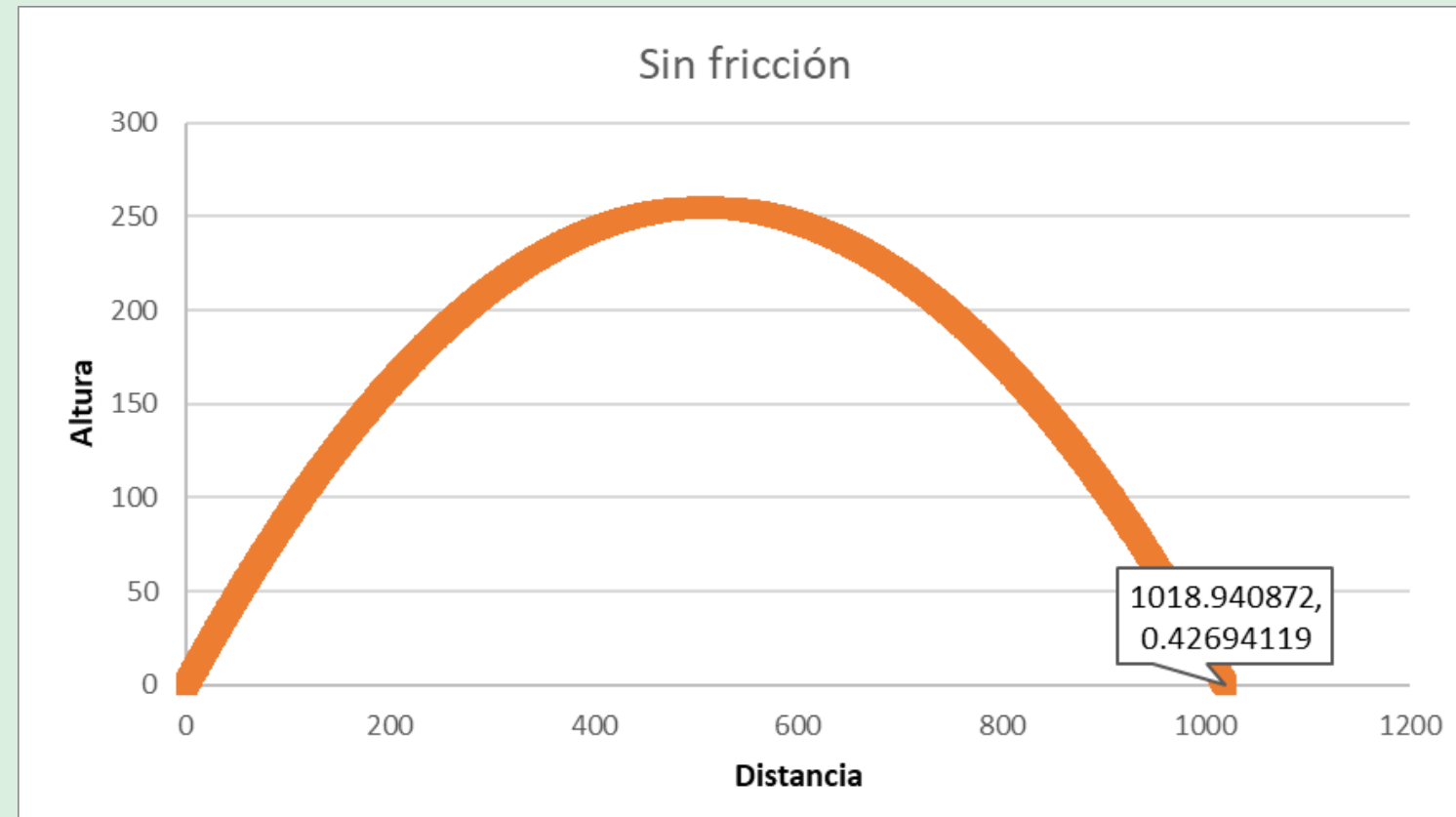
SIN FRICCIÓN

1							
2		Velocidad	100 m/s				
3		Angulo	45 0.785398163				
4		Vxo	70.71068				
5		Vyo	70.71068		dt	0.01	
6		xmax	1019.368				
7		hmax	254.842				
8		tvuelo	14.41604 tmedio_vuelo	7.208020196			
9		g	-9.81				
10							
11			Tiempo	Velocidadx	Velocidady	Distancia	Altura
12			0.01	70.71067812	70.61257812	0.707107	0.706616
13			0.02	70.71067812	70.51447812	1.414214	1.412252
14			0.03	70.71067812	70.41637812	2.12132	2.116906
15			0.04	70.71067812	70.31827812	2.828427	2.820579
16			0.05	70.71067812	70.22017812	3.535534	3.523271
17			0.06	70.71067812	70.12207812	4.242641	4.224983
18			0.07	70.71067812	70.02397812	4.949747	4.925713
19			0.08	70.71067812	69.92587812	5.656854	5.625462
20			0.09	70.71067812	69.82777812	6.363961	6.324231
21			0.1	70.71067812	69.72967812	7.071068	7.022018
22			0.11	70.71067812	69.63157812	7.778175	7.718824
23			0.12	70.71067812	69.53347812	8.485281	8.414649
24			0.13	70.71067812	69.43537812	9.192388	9.109494
25			0.14	70.71067812	69.33727812	9.899495	9.803357

CON FRICCIÓN

1												
2		Velocidad	100 m/s									
3		Angulo	45 0.785398163									
4		Vxo	70.71068					masa	5	dt	0.01	
5		Vyo	70.71068					r	0.75			
6		xmax						Y	0.0000172			
7		hmax						p	0.00024316			
8		tvuelo		tmedio_vuelo								
9		g	9.81									
10												
11			Tiempo	Velocidadx	Velocidady	Distancia	Altura	Ffx	Ffy	Afx	Afy	
12			0	70.71067812	70.71067812	0	0	-0.01719	-0.017194	-0.003438799	-9.813438799	
13			0.01	70.71064373	70.61254373	0.707106	0.706125	-0.01719	-0.0171701	-0.003438798	-9.813434027	
14			0.02	70.71060934	70.51440939	1.414213	1.41127	-0.01719	-0.0171463	-0.003438796	-9.813429255	
15			0.03	70.71057495	70.4162751	2.121318	2.115432	-0.01719	-0.0171224	-0.003438794	-9.813424482	
16			0.04	70.71054057	70.31814085	2.828424	2.818614	-0.01719	-0.0170985	-0.003438793	-9.81341971	
17			0.05	70.71050618	70.22000666	3.535529	3.520814	-0.01719	-0.0170747	-0.003438791	-9.813414937	
18			0.06	70.71047179	70.12187251	4.242633	4.222032	-0.01719	-0.0170508	-0.003438789	-9.813410165	
19			0.07	70.7104374	70.0237384	4.949738	4.92227	-0.01719	-0.017027	-0.003438788	-9.813405392	
20			0.08	70.71040302	69.92560435	5.656842	5.621526	-0.01719	-0.0170031	-0.003438786	-9.81340062	
21			0.09	70.71036863	69.82747034	6.363946	6.319801	-0.01719	-0.0169792	-0.003438784	-9.813395847	
22			0.1	70.71033424	69.72933639	7.071049	7.017094	-0.01719	-0.0169554	-0.003438783	-9.813391075	
23			0.11	70.71029985	69.63120248	7.778152	7.713406	-0.01719	-0.0169315	-0.003438781	-9.813386302	
24			0.12	70.71026546	69.53306861	8.485255	8.408737	-0.01719	-0.0169076	-0.003438779	-9.81338153	
25			0.13	70.71023108	69.4349348	9.192357	9.103086	-0.01719	-0.0168838	-0.003438778	-9.813376758	
26			0.14	70.71019669	69.33680103	9.899459	9.796454	-0.01719	-0.0168599	-0.003438776	-9.813371985	
27			0.15	70.7101623	69.23866731	10.60656	10.48884	-0.01719	-0.0168361	-0.003438774	-9.813367213	
28			0.16	70.71012791	69.14053364	11.31366	11.18025	-0.01719	-0.0168122	-0.003438773	-9.81336244	

GRÁFICAS



CODIGO COMPLETO EN MATLAB

1. Creamos una Interfaz con el comando 'Guide'

```
1
2 function varargout = Reto2(varargin)
3 gui_Singleton = 1;
4 gui_State = struct('gui_Name',       mfilename, ...
5                   'gui_Singleton',   gui_Singleton, ...
6                   'gui_OpeningFcn', @Reto2_OpeningFcn, ...
7                   'gui_OutputFcn',  @Reto2_OutputFcn, ...
8                   'gui_LayoutFcn',   [] , ...
9                   'gui_Callback',    []);
10 if nargin && ischar(varargin{1})
11     gui_State.gui_Callback = str2func(varargin{1});
12 end
13 if nargout
14     [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
15 else
16     gui_mainfcn(gui_State, varargin{:});
17 end
```

Definimos lo que hace cada una de las funciones en la interfaz

```
19 function Reto2_OpeningFcn(hObject, eventdata, handles, varargin)
20 axes(handles.axes1);
21 background = imread("Imagen Matlab3.jpg");
22 axis off;
23 imshow(background);
24 handles.output = hObject;
25 guidata(hObject, handles);
26
27
28 function varargout = Reto2_OutputFcn(hObject, eventdata, handles)
29 varargout{1} = handles.output;
30
31 function para_angulo_Callback(hObject, eventdata, handles)
32 angulo = str2double(get(hObject, 'String'))
33 function para_angulo_CreateFcn(hObject, eventdata, handles)
34 if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUicontrolBackgroundColor'))
35     set(hObject, 'BackgroundColor', 'white');
36 end
37 function para_veloci_Callback(hObject, eventdata, handles)
38
39 vi = str2double(get(hObject, 'String'))
40 function para_veloci_CreateFcn(hObject, eventdata, handles)
41 if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUicontrolBackgroundColor'))
42     set(hObject, 'BackgroundColor', 'white');
43 end
```


CODIGO COMPLETO EN MATLAB

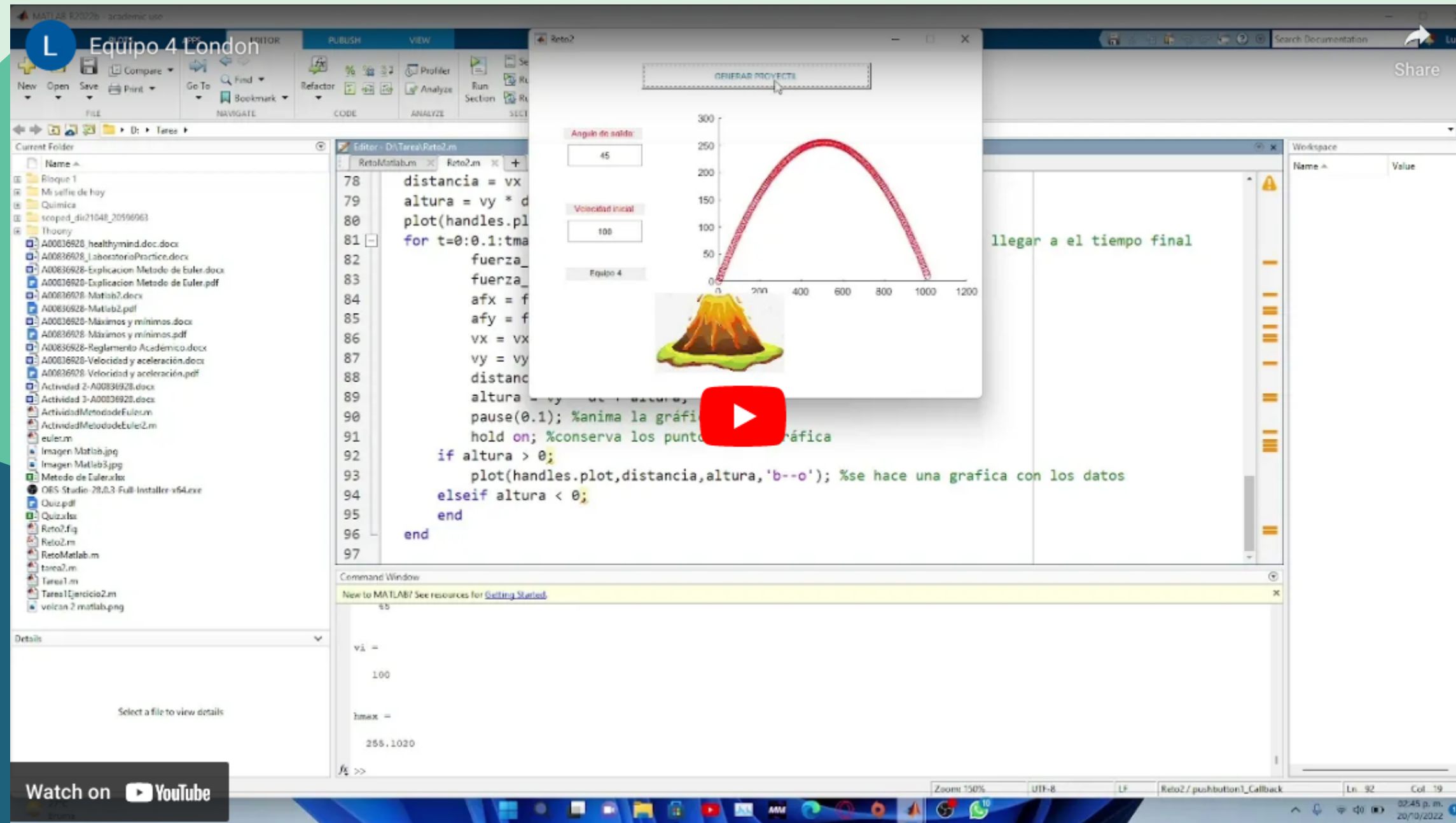
Iniciamos a programar y graficar la trayectoria del proyectil sin resistencia del aire

```
45 %inicio de funciones
46 function pushbutton1_Callback(hObject, eventdata, handles)
47 cla(handles.plot, 'reset');
48 vi= str2double(get(handles.para_veloci, 'String'));
49 angulo= str2double(get(handles.para_angulo, 'String'));
50 radianes = deg2rad(angulo);
51 g = -9.8;
52 vx0 = vi*cos(radianes);
53 vyo = vi*sin(radianes);
54 tmedio = (-vyo / g);
55 tmax = tmedio * 2;
56 xmax = tmax * vx0;
57 hmax = vyo * tmedio+(1/2)*(g*tmedio^2)
58 for t=0:0.1:tmax %se abre ciclo for con aumentos de .1 segundos hasta llegar a el tiempo final
59     distancia = vx0 * t;
60     altura = vyo*t+(1/2)*(g*t^2);
61     pause(0.1); %anima la gráfica
62     hold on; %conserva los puntos en la gráfica
63     plot(handles.plot,distancia,altura, 'r--o',tmax,hmax); %se hace una grafica con los datos
64 end
```

Definimos lo que hace cada una de las funciones en la interfaz

```
65 dt = 0.1
66 masa = 5;
67 radio = .75;
68 vis_aire = 17.2*(10^-6)
69 fuerza_vis = 6*pi*vis_aire*radio;
70 distancia = 0;
71 altura = 0;
72 fuerza_fricx = -fuerza_vis*vx0;
73 fuerza_fricy = -fuerza_vis*vyo;
74 afx = fuerza_fricx/masa;
75 afy = fuerza_fricy/masa;
76 vx = vx0 + afx * dt;
77 vy = vyo + g * dt + afy * dt;
78 distancia = vx * dt;
79 altura = vy * dt;
80 plot(handles.plot,distancia,altura, 'b--o');
81 for t=0:0.1:tmax %se abre ciclo for con aumentos de .1 segundos hasta llegar a el tiempo final
82     fuerza_fricx = -fuerza_vis*vx;
83     fuerza_fricy = -fuerza_vis*vy;
84     afx = fuerza_fricx/masa;
85     afy = fuerza_fricy/masa;
86     vx = vx + afx * dt;
87     vy = vy + g * dt + afy * dt;
88     distancia = vx * dt + distancia;
89     altura = vy * dt + altura;
90     pause(0.1); %anima la gráfica
91     hold on; %conserva los puntos en la gráfica
92     if altura > 0;
93         plot(handles.plot,distancia,altura, 'b--o'); %se hace una grafica con los datos
94     elseif altura < 0;
```

PROGRAMA CORRIENDO



CONCLUSIONES

David Mireles Gutierrez

Durante esta clase he fortalecido todos mis conocimientos con los que antes contaba ya que, siempre me había confundido con la aceleración centrípeta el uso de la palabra tangencial, tensiones etc...De igual manera aprendí a usar softwares como lo es excel y matlab. Gracias a esto me siento más con más seguridad al hacer actividades en excel porque antes me confundía mucho en como funcionaba. De igual manera este bloque me ayudó a mejorar mi habilidad de analizar y responder los problemas de física.

Carlos Ignacio Marvan Ramirez

Con el fin de este proyecto los aprendizajes que nos llevamos fueron de gran importancia ya que todo lo que hicimos nos ayudo a aplicar los aprendizajes que obtuvimos a lo largo de el periodo ayudandonos a desarrollar mejores habilidades y otras nuevas que pudimos ir implementando conforme avanzavamos en los temas, como el poder usar matlab, desarrollar mas cosas en excel utilizar formulas y mas programas.

Gerardo Daniel García de la Garza

Me llevo bastante aprendizaje, a pesar de ya contar con algo de conocimiento sobre el tiro parabólico, el tener que utilizar todas esas fórmulas para plasmarlas en herramientas como excel o matlab fue algo más laborioso pero más práctico y ahí es donde está la clave, el hacer las cosas más prácticas y óptimas para poder solucionar problemas más grandes, el tener que desglosar la información y datos hace que utilicen más la lógica y esto sirve para la vida profesional, resolver problemas con resultados óptimos capaces de ser reutilizados.

Luis Gerardo Juarez Garcia

En este reto investigamos y aplicamos nuestros conocimientos vistos en física, matemáticas y computación para programar una interfaz que nos indique y prediga la aproximación de un proyectil que saldría directamente de un volcán en erupción.

Pasamos por diferentes etapas como la de investigación, donde pudimos obtener las formulas necesarias para el desarrollo posterior del reto, luego aplicamos nuestros conocimientos en Excel y lo juntamos con nuestro conocimiento de matemáticas para poder juntarlo todo en una tabla con método de Euler. Finalmente traducimos todo lo anterior y lo optimizamos en Matlab para obtener nuestro preciado código.

CARLOS MARVAN A00838096



Bye bye

DAVID MIRELES A00836010

GERARDO DANIEL A00836801

LUIS GERARDO A00836928