

Universidad ORT Uruguay

Facultad de Ingeniería

Proyecto Numérico 1

Fundamentos de Sistemas Ciberfísicos

18 de Junio del 2020

Matias Hernández – 169236

Gianfranco Drago – 198490

Ejercicio 1)

Código:

```
clear
// Def: Aceleración gravitatoria, Unidad: m/s^2
g=9.8
// Def: Paso de integración Delta t para el método de Euler, Unidad: s
dt=0.001
// Def: Posición y inicial, a nivel del mar, Unidad:m
y0=0
// Def: Posición x inicial, a nivel del mar, Unidad:m
x0=0
// Def: Rapidez inicial, Unidad: m/s
v0=30
// Def: Ángulo inicial, Unidad: grados
Theta0=0
// Def: Tiempo inicial, Unidad: s
t0=0
```

```
function [t, x, y]=ObtenerTrayectoriaPorEuler(Theta);
```

```
// Condiciones iniciales
```

```
x(1)=x0
```

```
y(1)=y0
```

```
t(1)=t0
```

```
vx=v0*cosd(Theta)
```

```
vy(1)=v0*sind(Theta)
```

```
i=1
```

```
while y(i)>=0
```

```
    t(i+1)=t(i)+dt
```

```
    vy(i+1)=vy(i)-g*dt
```

```
    y(i+1)=y(i)+vy(i)*dt
```

```
    x(i+1)=x(i)+vx*dt
```

```
    i=i+1
```

```
end
```

```
endfunction
```

```
function [ThetaXMax]=ObtenerThetaXMaxEnsayoError();
```

```
// Condiciones iniciales
```

```
Theta = Theta0
```

```
ThetaXMax = Theta0
```

```
XMax = x0
```

```
while Theta<=90
```

```

[t,x,y] = ObtenerTrayectoriaPorEuler(Theta)
if x($)>XMax then
    ThetaXMax = Theta
    XMax = x($)
end
Theta=Theta+5
end
endfunction

function [ThetaMax]=ObtenerThetaXMaxAnalitico();
    ThetaMax=atand(v0/sqrt(v0^2+2*g*y0))
endfunction

function [Ne, xA, yA]=ObtenerTrayectoriaAnalitica(Theta);
    xA=v0*cosd(Theta)*t
    yA=-g*t.^2/2+v0*sind(Theta)*t
    Ne=length(t);
endfunction

// Parte 1)a)

[ThetaXMaxAnalitico] = ObtenerThetaXMaxAnalitico()
disp("Theta para que x sea xMax (Analitico):")
disp(ThetaXMaxAnalitico)

[ThetaXMaxEnsayoError] = ObtenerThetaXMaxEnsayoError()
disp("Theta para que x sea xMax (Ensayo y Error):")
disp(ThetaXMaxEnsayoError)

// Parte 1)b)

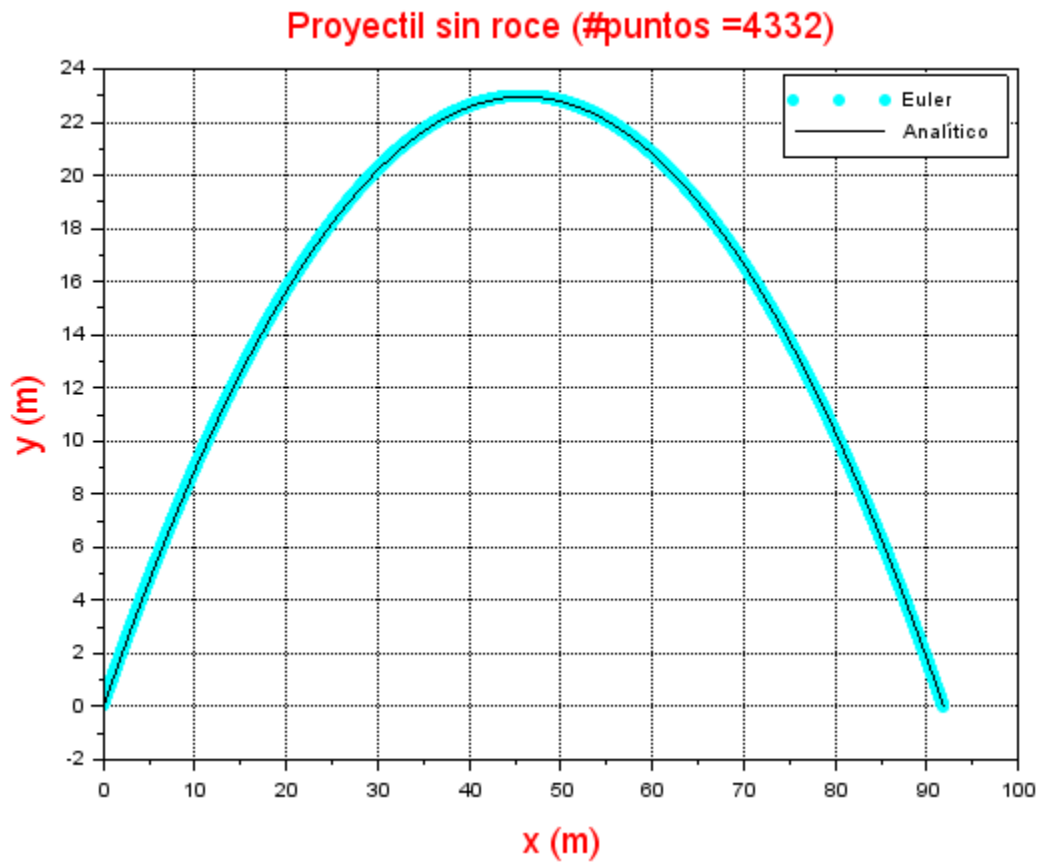
// Obtenemos los puntos de la trayectoria por Metodo de Euler
[t,xE,yE] = ObtenerTrayectoriaPorEuler(ThetaXMaxEnsayoError)
// Obtenemos los puntos de la trayectoria Analiticamente
[Ne,xA,yA] = ObtenerTrayectoriaAnalitica(ThetaXMaxAnalitico)

// Configuraciones de la Grafica
xgrid;
xlabel("x (m)","fontsize",4,"color","red")
ylabel("y (m)","fontsize",4,"color","red");
title("Proyectil sin roce (#puntos =" + string(Ne) + ")", "color","red","fontsize",4);
// Grafica de la trayectoria con Metodo de Euler
plot(xE,yE,"c.")
// Grafica de la trayectoria Analiticamente
plot(xA,yA,"k");
legend(["Euler";"Analítico"]);

```

Gráfica:

Ángulo inicial para que el alcance horizontal sea máximo, por ensayo y error, con precisión de 0,5 grado: 45 grados.



Ejercicio 2)

Código:

```
clear
// Def: Aceleración gravitatoria, Unidad: m/s^2
g=9.8
// Def: Posición y inicial, a nivel del mar, Unidad:m
y0=0
// Def: Posición x inicial, a nivel del mar, Unidad:m
x0=0
// Def: Tiempo inicial, Unidad: s
t0=0
// Def: Ángulo inicial, Unidad: grados
Theta0=0
// Def: Rapidez inicial, Unidad: m/s
v0=30
// Def: Masa pelota, Unidad: kg
m=0.45
// Def: Radio pelota, Unidad: m
r=0.11
// Def: Paso de integración Delta t para el método de Euler, Unidad: s
dt=0.001
// Def: Densidad del aire al nivel del mar, Unidad: kg/m^3
rho=1.12
// Def: Área efectiva pelota, m^2
A=%pi*r^2

function [t, x, y]=ObtenerTrayectoriaPorEulerConAire(Theta);
// Condiciones iniciales
x(1)=x0
y(1)=y0
t(1)=t0
vx(1)=v0*cosd(Theta)
vy(1)=v0*sind(Theta)

i=1
while y(i)>=0
    v=sqrt(vx(i)^2+vy(i)^2)
    // Sum Fx = m*ax
    // m*ax = -(1/2)*rho*A*v*v*vx
    // ax = -(1/2)*(1/m)*rho*A*v*v*vx
    ax=-(1/2)*(1/m)*rho*A*v*v*vx(i)
    // Sum Fy = m*ay
    // m*ay = -m*g-(1/2)*rho*A*v*v*vy
```

```

// ay = -g-(1/2)*(1/m)*rho*A*v*vy
ay=-g-(1/2)*(1/m)*rho*A*v*vy(i)
t(i+1)=t(i)+dt
vy(i+1)=vy(i)+ay*dt
vx(i+1)=vx(i)+ax*dt
y(i+1)=y(i)+vy(i)*dt
x(i+1)=x(i)+vx(i)*dt
i=i+1
end
endfunction

function [t, x, y]=ObtenerTrayectoriaPorEulerSinAire(Theta);
// Condiciones iniciales
x(1)=x0
y(1)=y0
t(1)=t0
vx=v0*cosd(Theta)
vy(1)=v0*sind(Theta)

i=1
while y(i)>=0
    t(i+1)=t(i)+dt
    vy(i+1)=vy(i)-g*dt
    y(i+1)=y(i)+vy(i)*dt
    x(i+1)=x(i)+vx*dt
    i=i+1
end
endfunction

function [ThetaXMax]=ObtenerThetaXMaxConAire();
// Condiciones iniciales
Theta = Theta0
ThetaXMax = Theta0
XMax = x0

while Theta<=90
    [t,x,y] = ObtenerTrayectoriaPorEulerConAire(Theta)
    if x($)>XMax then
        ThetaXMax = Theta
        XMax = x($)
    end
    Theta=Theta+1
end
endfunction

[ThetaXMaxConAire] = ObtenerThetaXMaxConAire()

```

```

disp("Theta para que x sea xMax (Con Aire):")
disp(ThetaXMaxConAire)

// Obtenemos los puntos de la trayectoria por Metodo de Euler
[tS,xS,yS] = ObtenerTrayectoriaPorEulerSinAire(ThetaXMaxConAire)
[tA,xA,yA] = ObtenerTrayectoriaPorEulerConAire(ThetaXMaxConAire)
// Configuraciones de la Grafica
xgrid;
xlabel("x (m)","fontsize",4,"color","red")
ylabel("y (m)","fontsize",4,"color","red");
title("Proyectil sin aire vs con aire","color","red","fontsize",4);
// Grafica de la trayectoria con Metodo de Euler
plot(xS,yS,"r")
plot(xA,yA,"k")

legend(["Euler Sin Aire";"Euler Con Aire"]);

```

Gráfica:

Ángulo inicial para que el alcance horizontal sea máximo, por ensayo y error, con precisión de 1 grado: 36 grados.

