

# Tiro Parabólico con Resistencia del Aire

César Andrés Pérez Robinson

October 27, 2017

## 1 Descripción de la Actividad

Para esta actividad se genera una gráfica con gnuplot sobre el tiro parabólico de un objeto con resistencia del aire. Guiándonos de ecuaciones de movimiento, se resuelven utilizando el método de integración de Euler. Se calcula la velocidad terminal, por lo que es necesario conocer la geometría del objeto, puesto que de ella depende el coeficiente de arrastre.

## 2 Geometría del objeto

Se escogió un objeto similar a una pelota de béisbol cuyas dimensiones son:

Radio: 0.05 m

Masa: 0.250 kg

El coeficiente de arrastre de la esfera: 0.47

## 3 Gráfica y Código Utilizado

La grafica de Figura 1 fue generada a través de gnuplot y muestra el movimiento del proyectil con velocidades iniciales 10, 20, 30, 40, 50, 60, 70, 80, 90 y 100.

El código utilizado es el siguiente:

```
program vector
```

```
    implicit none
```

```
    ! definimos constantes
```

```
    real, parameter :: g = 9.81
```

```
    real, parameter :: pi = 3.1415927
```

```
    real, parameter :: cd = 0.47
```

```
    real, parameter :: de = 1.22
```

```
    real, parameter :: r = 0.05
```

```
    real, parameter :: m = 0.250
```

```
    real, parameter :: dt = 0.01
```

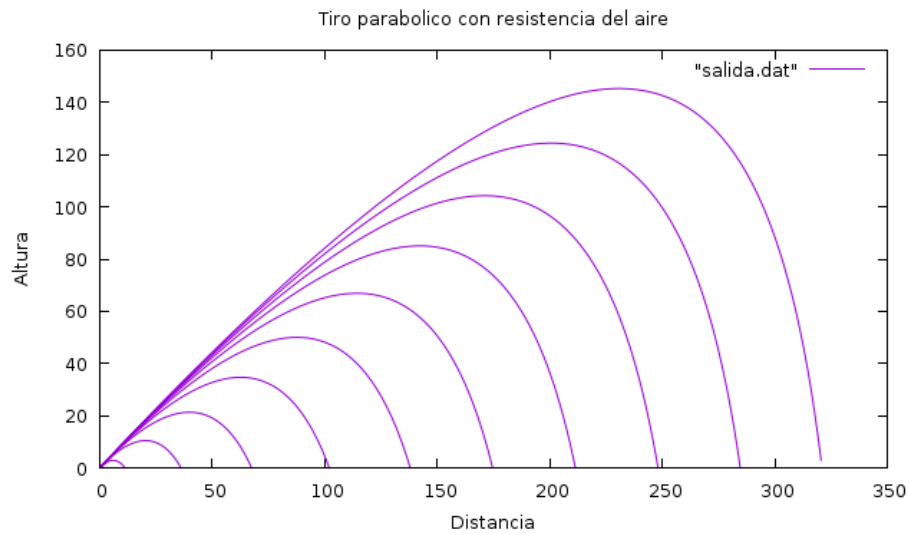


Figure 1: Resistencia del Aire

```
! definimos las variables
real :: u, t, a
integer :: i, j
integer, parameter :: ntimes = 1000
real, dimension (1 : ntimes) :: x, y, vx, vy
real :: fi, fj
real, parameter :: Area = pi * r * r
real, parameter :: Vet = SQRT(2 * m * g / (de * Area * cd))
real, parameter :: c = m * g / Vet

open(unit=15, file="salida.dat", status="unknown")

! Leer los valores para el ángulo a y la velocidad inicial u desde
! la terminal
write (*,*) 'Dame el ángulo'
read (*,*) a

! convirtiendo ángulo a radianes
a = a * pi / 180.0

! las ecuaciones de la posición en x y y
! x = u * cos(a) * i
```

```

!y = u * sin(a) * i - 0.5 * g * i * i

do j = 10, 100, 10
  fj = float(j)
  u = fj

  write(15,*) " "

  do i = 1, 2
    fi = float(i)
    t = fi * dt

    x(i) = u * cos(a) * t
    y(i) = u * sin(a) * t - 0.5 * g * t * t
    vx(i) = u
    vy(i) = u * sin(a) - g * t

    write(15,*) x(i), y(i)

  end do

  do i = 3, ntimes
    fi = float(i)
    t = fi * dt

    vx(i) = vx(i - 1) * (1 - dt * c / m)

    x(i) = x(i - 1) + dt * vx(i - 2) * (1 - dt * c / m)

    vy(i) = vy(i - 1) * (1 - dt * c / m) - dt * g

    y(i) = y(i - 1) + dt * vy(i - 2) - dt * dt * g + dt * c / m * &
      & vy(i - 2)

    if (y(i) < 0) exit

    write(15,*) x(i), y(i)

  end do
end do

close(unit=15)

```

```
!write(*,*) 'Modulus squared = ',x  
end program vector
```