

Movimiento de proyectil con resistencia del aire

José Ramón Pérez Navarro

September 2019

1 Introducción

En esta ocasión analizamos el lanzamiento de un proyectil tomando en cuenta la resistencia del aire.

Código

```
program pelotabeisbol

implicit none

! definimos constantes
real, parameter :: g = -9.81
real, parameter :: pi = 3.1415927

! definimos las variables

real :: sx, sy, k, tymax, vymax, txmax
real :: vxmax, xmax, vx, vy, dt, t, voy, vox, v, x, y
real :: a, m, vo, vt
! definimos contadores
integer :: n, npasos

! output data into a file
open(11, file = 'salida.dat', status = 'unknown')

! Leer valores para el ángulo a, el tiempo t, la masa, la velocidad inicial
! vo y la velocidad terminal vt
write(*,*) 'Dame el ángulo, el numero de pasos, la masa, la velocidad inicial y vt'
read(*,*) a, npasos, m, vo, vt

! convirtiendo ángulo a radianes
a = a * pi / 180.0
dt= 0.1
```

```

!calulando constante k
  k = (m * g) / vt

! La velocidad iniciales al tiempo t
  vox = vo * cos(a)
  voy = vo * sin(a)

! Loop

  do n = 0, npasos
    t= float(n)* dt

    ! Movimiento en eje x

    vx = vox * exp ((-k*t) / m )

    ! Condicion de velocidad horizontal por si supera la velocidad terminal

    if (vt + vx > 0 ) then

      vx = vt* (-1)

    end if

    ! Movimiento en eje y

    vy = (voy - ((m * g) / k)) *exp ((- k * t) / m) + (m * g) / k

    ! Condicion velocidad vertical por si supera la velocidad terminal

    if (vt + vy > 0 .and. vy > 0) then

      vy = vt * (-1)

    else if (vy < vt .and. vy < 0) then

      vy = vt

    end if

    if (vy==0) then

      tymax = t
      vymax = vy

```

```

end if

! calculando las posiciones en x, y

sx = (m / k) * vx * (1 -exp ((-k * t) / m)) * 3

sy = ((m/k) * (vy - ((m * g ) / k)) * ( 1 - (exp ((- k / m) * t ))) + (( m * g) / k) *

! salir cuando la altura sea menor que cero

if (sy < 0.0 ) then

    txmax = t

exit

end if
print *, sx, sy
write (1,*) sx, sy

end do

write(1,*) ' '

! trayectoria sin resistencia al aire

do n = 0, npasos
t= float(n)* dt

    ! Calculando las posiciones (x, y) respecto al tiempo.

    x=vox * t
    y=voy * t + 0.5 * g * t * t

    ! Salir cuando la altura sea menor que cero
    if (y<0) exit

    !Posiciones
    print *, x, y
    write(1,*) x, y
end do
close (1)

!Calculando la posición en x cuando llega al suelo

xmax=(m/k)*vx*(1-exp((-k*txmax)/m))*3

```

```

!Calculando las velocidades en (x,y) al momento del alcance maximo

    vxmax=v0x*exp((-k*txmax)/m)

! Cuando la velocidad horizontal supera la velocidad terminal

if (vt+vx>0) then

    vx=vt*(-1) !La velocidad es igual a la velocidad terminal

end if

vy=(v0y-((m*g)/k))*exp((-k*txmax)/m)+(m*g)/k

if (vy<vt .and. vy<0) then
    vy=vt
end if

v=sqrt((vx*vx)+(vy*vy))

print *, txmax, xmax, v
write (*,*) txmax, xmax, v

end program pelotabeisbol

```

Gráfica de los datos

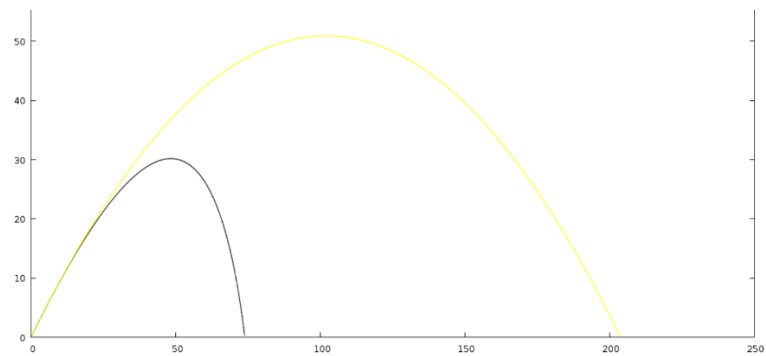


Figure 1:

Conclusión

Los datos obtenidos con el código concuerdan con los obtenidos en el programa de lanzamientos de proyectiles del sitio phet.