

# Reporte de práctica 6

Alejandro Guirado García

13 de abril de 2015

## 1. Introducción

En esta actividad se realizó un programa ejecutable para observar el movimiento de un proyectil con la fuerza de fricción producida por el aire. Se nos pide además, tomar ciertas evidencias y comparar los resultados del movimiento de un objeto cuando hay y cuando no hay fricción de arrastre. Además hay que considerar las diferentes formas que puede tener un objeto, ya que la fricción con la superficie o la fuerza de arrastre también varía.

Todo objeto de masa  $m$  que se mueve a muy alta velocidad en un fluido de densidad  $\rho$ , experimenta una fuerza de arrastre  $F_D$  contraria a la dirección de su movimiento y es dada por la ecuación:

$$F_D = \frac{1}{2} \rho u^2 C_D A$$

donde  $u$  es la magnitud del vector velocidad del objeto,  $C_D$  es el coeficiente de arrastre (adimensional),  $A$  es el área transversal presentada por el objeto (sección eficaz). Por ejemplo, para una esfera el área transversal es  $A = \pi r^2$ , y el coeficiente de arrastre es  $C_D = 0.47$ .

## 2. Tiro parabolico con fuerza de arrastre

La actividad pide calcular el error al no considerar el arrastre del aire, al lanzar una bola esférica de masa  $m=0.25\text{kg}$ , y radio  $R = 0.05\text{m}$ . Haz un cálculo con y sin arrastre y compara para ángulos complementarios por arriba y debajo de  $45^\circ$ .

## 3. Código en Fortran

Sección de declaración de variables Sección de declaración de constantes Subrutina para cálculo de trayectoria con entradas:  $dt$ ,  $x_0$ ,  $y_0$ ,  $v_{0x}$ ,  $v_{0y}$ ,  $a_x$ ,  $a_y$ ; salida:  $t$ ,  $x$ ,  $y$  Subrutina para cálculo de fuerza de arrastre con entrada:  $m$ , densidad,  $A$ ,  $C_D$ ,  $u_{0x}$ ,  $v_{0y}$ ; salida:  $a_x$ ,  $a_y$ .

```
!Programa para calcular la posicion de un proyectil en movimiento parabolico, consinderando la friccion
module Constantes !Declaracion de constantes
implicit NONE
real, parameter :: rad=(4.0*atan(1.0))/180 ! Conversion de angulos
real, parameter :: pi=4.0*atan(1.0)
integer, parameter :: pts= 7000 !Numero de puntos a calcular
real, parameter :: rf = 1.29
real, parameter :: cfe = 0.47
end module Constantes

program proyectilf
use Constantes
implicit none
```

```

real :: x0,y0,v0,gr0
real :: xmax,ymax,tiempo,xmaxf,ymaxf,tiempof
real :: error
write (*,*), 'Insterte las coordenadas iniciales de x & y en metros'
read *, x0, y0
write (*,*), 'Inserte la velocidad inicial del proyectil en m/s'
read *, v0
write (*,*), 'Insterte el angulo inicial de tiro en grados'
read *, gr0

call sinfriccion (x0,y0,v0,gr0,xmax,ymax,tiempo)
call friccion (x0,y0,v0,gr0,xmaxf,ymaxf,tiempof)

error = ((xmax-xmaxf)/xmaxf * 100)

print *, "Coordenadas en", x0,y0
print *, "Velocidad inicial de", v0,"m/s"
print *, "Angulo de", gr0
print *, "----"
print *, "Sin friccion"
print *, "Tiempo en el aire=", tiempo,"s"
print *, "Altura maxima=", ymax, "m"
print *, "Alcance horizontal=", xmax,"m"
print *, "----"
print *, "Con friccion"
print *, "Tiempo en el aire=", tiempof,"s"
print *, "Altura maxima=", ymaxf,"m"
print *, "Alcance horizontal=", xmaxf,"m"
print *, "Y el error al no considerar ninguna el arrastre del aire es=", error

end program proyectilf

subroutine sinfriccion (x0,y0,v0,gr0,xmax,ymax,tiempo)
use Constantes
implicit none
integer :: I
real, dimension (1:pts) :: x, y, t
real :: x0, y0, v0, gr0
real :: xmax, ymax, tiempo

gr0=gr0*rad
xmax = x0+((v0*v0*sin(2*gr0))/(9.8))
ymax = y0+((v0*v0*sin(gr0)*sin(gr0))/(19.6))
tiempo= (2*v0*sin(gr0))/(9.8)

open (1, file= "sinfriccion.dat")
do I=1, pts, 1
t(I)= float(I)*0.01
x(I)= x0+(v0*cos(gr0)*t(I))
y(I)= y0+(v0*sin(gr0)*t(I))-(4.9*t(I)*(t(I)))

write (1,1001) x(I), y(I)
1001 format (f11.5,f11.5)

if (y(I)<0) exit

```

```

end do

close (1)
end subroutine sinfriccion

subroutine friccion (x0,y0,v0,gr0,xmaxf,ymaxf,tiempof)
use Constantes
implicit none
integer :: I
real, dimension (0:pts) :: p,q,c,velp,velq,lp,lq
real :: x0,y0,v0,gr0 !Entrada
real :: xmaxf,ymaxf,tiempof !Salida
real :: d, area, radio, masa
print *, "Ingrese la masa en kg"
read *, masa
print *, "Ingrese el radio de la esfera en metros"
read *, radio

area=pi*radio*radio

p(0) = x0
q(0) = y0
velp(0) = v0*cos(gr0)
velq(0) = v0*sin(gr0)
d = (0.5*r*f*area*cfe)/masa
lp(0) = -d*velp(0)*velp(0)
lq(0) = 9.8-(d*velq(0)*velq(0))
c(0) = 0

!Registrando valores
open (2, file="friccion.dat")
write (2,1001) p(0), q(0)
1001 format (f11.5,f11.5)

!Calculando posicion para t(I)
do I=0, pts, 1
c(I+1)= c(I)+ 0.01
velp(I+1) = velp(I)+lp(I)*c(I+1)
velq(I+1) = velq(I)+lq(I)*c(I+1)

lp(I+1) = -d*velp(I)*velp(I)
lq(I+1) = -9.8-(d*velp(I)*velp(I))

p(I+1) = p(I)+velp(I)*c(I+1)+(0.5*lp(I)*c(I+1)*c(I+1))
q(I+1) = q(I)+velq(I)*c(I+1)+(0.5*lq(I)*c(I+1)*c(I+1))
write (2,*) p(I+1), q(I+1)
if (q(I)<0) exit
end do
close (2)

xmaxf = p(I)
ymaxf = MAXVAL(q)

```

```

tiempof = c(I)*10
end subroutine friccion

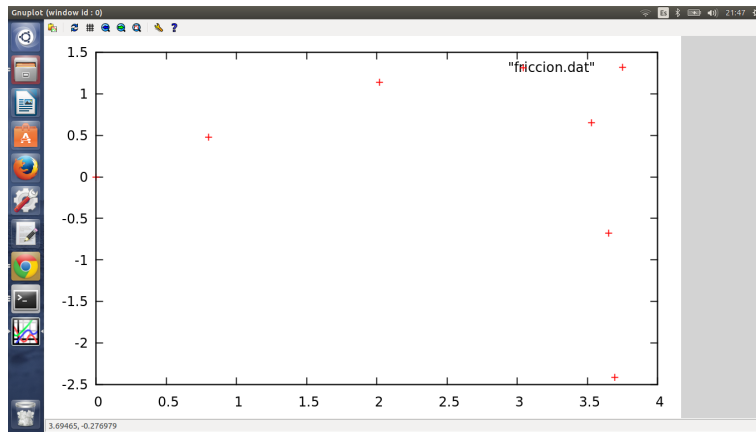
```

## 4. Ejemplos y gráficas

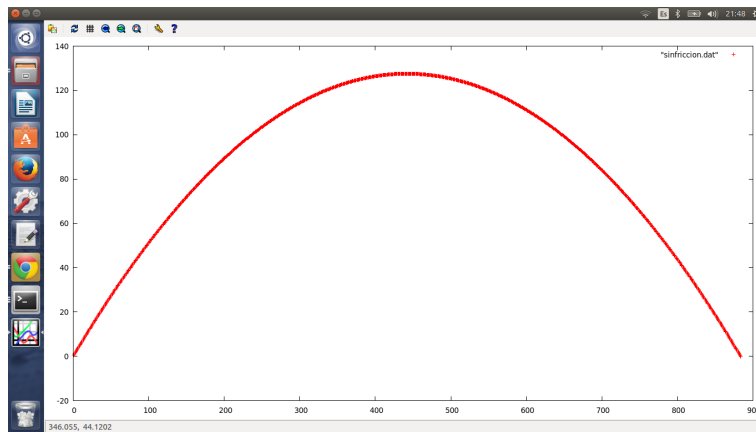
Tiros parabolicos empezando desde  $v_0=0$  y ambos ejes en 0. Variando ángulos para comparar los resultados.

### 4.1. Ángulo de 30 grados y $v_0=50\text{m/s}$

- Con fricción

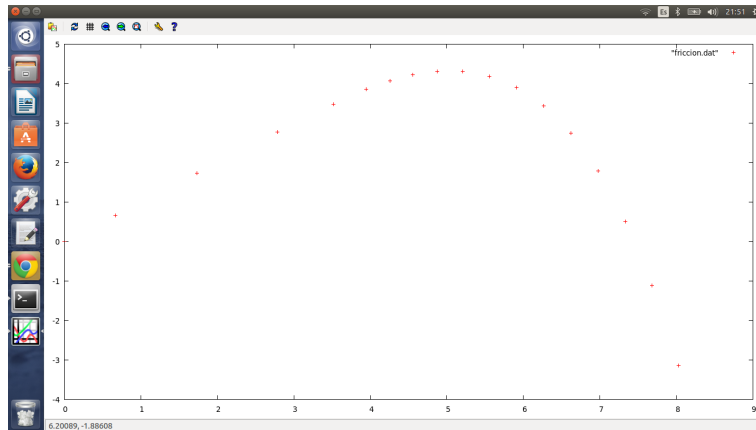


- Sin fricción

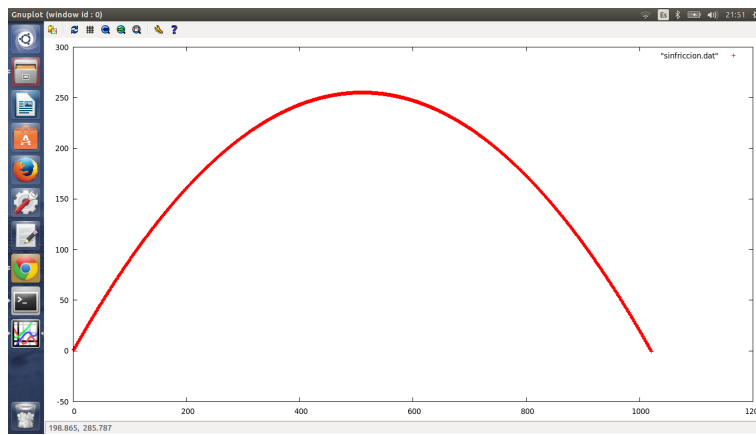


### 4.2. Angulo de 45 grados y $v_0=50\text{m/s}$

- Con fricción

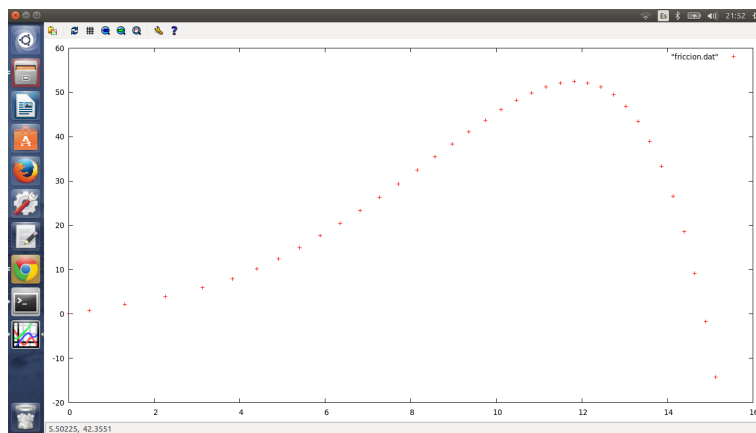


- Sin fricción

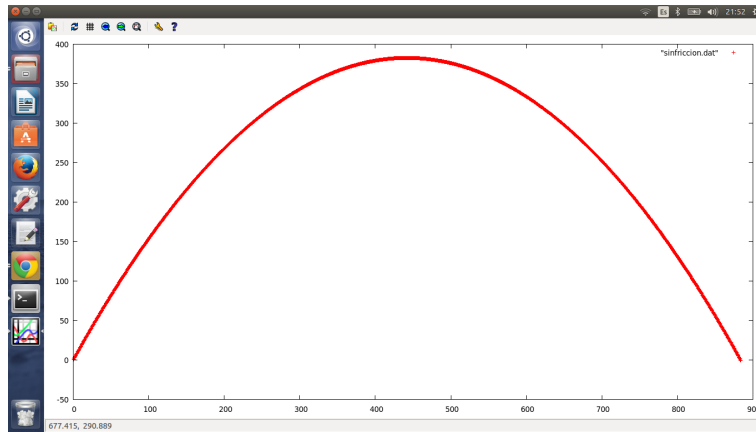


#### 4.3. Angulo de 60 grados y $v_0=50\text{m/s}$

- Con fricción



- Sin fricción



## 5. Conclusión

Existe una notoria diferencia al tomar en cuenta la resistencia que ofrece el aire, ante movimientos en su flujo. Además su oposición es directamente proporcional a la velocidad del objeto en movimiento, es decir, a mayor velocidad mayor será la fuerza de empuje del aire.