

# Tiro parabólico con fricción del aire

Rios Quijada Danira

10 de Abril de 2015

## 1. Introducción

En esta práctica, realizamos un programa en FORTRAN para calcular la posición de un objeto lanzado con tiro parabólico, en ciertos instantes de tiempo, tanto en condiciones ideales, como tomando en consideración la resistencia del aire y comparando ambos resultados.

## 2. Un poco de física

En nuestro estudio del movimiento de proyectiles, habitualmente despreciamos la resistencia del aire, sin embargo esta tiene un efecto apreciable en el movimiento de proyectiles.

No es difícil incluir la resistencia del aire en las ecuaciones del proyectil, pero resolverlas para la posición y la velocidad en función del tiempo puede resultar un poco complejo.

Cuando omitimos la resistencia del aire, la única fuerza actuando en el proyectil es el peso  $p = mg$  y los componentes de la aceleración son sencillos  $a_x = 0$  y  $a_y = -g$ .

Ahora incluyendo la resistencia del aire actuando sobre el proyectil, la  $f$  será más o menos proporcional al cuadrado de la velocidad del proyectil.  $f = Dv^2$ , donde  $v^2 = v_x^2 + v_y^2$ . La dirección de  $f$  es opuesta a la dirección de  $v$  por ende  $f = -Dvv$ . Entonces:  $f_x = -Dvv_x$  y  $f_y = -Dvv_y$ .

Por la segunda ley de Newton  $a_x = -(D/m)vv_x$  y  $a_y = -g - (D/m)vv_y$ .

La constante  $D$  depende de la densidad del aire, así como del área en contacto del proyectil y el coeficiente de fricción del tipo de proyectil.  $D = ([\rho]CA)/2$ .

Los componentes de la aceleración están en constante cambio al igual que los componentes de la velocidad, pero en periodos de tiempo pequeños  $[\Delta]t$ . Durante el intervalo de tiempo  $[\Delta]t$   $a_x = [\Delta]v_x/([\Delta]t)$  y la velocidad en  $x$  cambiará de

acuerdo a  $[\Delta]v_x = a_x[\Delta]t$ , lo mismo pasa con los componentes en y. Llegamos a  $v_x + [\Delta]v_x = v_x + a_x[\Delta]t$  y  $v_y + [\Delta]v_y = v_y + a_y[\Delta]t$ .

Mientras el proyectil se mueve sus coordenadas logicamente estan cambiando, la coordenada x cambia de la siguiente manera:  $[\Delta]x = (v_x + [\Delta]v_x/2)[\Delta]t = v_x[\Delta]t + 1/2a_x([\Delta]t)^2$  en el eje y es exactamente la misma expresion. Entonces las coordenadas del proyectil para el final del intervalo son:

$$x + [\Delta]x = x + v_x[\Delta]t + 1/2a_x([\Delta]t)^2 \text{ y } y + [\Delta]y = y + v_y[\Delta]t + 1/2a_y([\Delta]t)^2$$

### 3. Código

El código que utilizamos para realizar el programa fue el siguiente:

!Con este programa podremos calcular la posicion de un proyectil en tiro parabolico, conciderando la resistencia del aire y comparalo con una situacion ideal.

```
module 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 :: npts= 2000
!Densidad del aire al nivel del mar
REAL, PARAMETER :: densidad = 1.18
Real, parameter :: cfe = 0.47
end module constantes

program tiro_resistencia
use constantes
implicit none
real :: xi, yi, vi, theta
real :: xmaxsf, ymaxsf, timesf, xmaxcf, ymaxcf, timecf
real :: diferenciay, diferenciay
write (*,*), 'Deme las coordenadas iniciales de x & y en m'
read *, xi, yi
write (*,*), 'Deme la velocidad inicial del proyectil en m/s'
read *, vi
write (*,*), 'Deme el angulo inicial de tiro en grados'
read *, theta

call sin_friccion (xi,yi,vi,theta,xmaxsf,ymaxsf,timesf)
call con_friccion (xi,yi,vi,theta,xmaxcf,ymaxcf,timecf)
```

```

diferenciay = ((ymaxsf-ymaxcf)/ymaxcf) * 100.0
diferenciay = ((ymaxsf-ymaxcf)/ymaxcf) * 100.0

print *, "~~~~~"
print *, "||En condiciones ideales||"
print *, "Un tiro parabolico con:"
print *, "coordenadas iniciales x=",xi,"y=",yi
print *, "velocidad inicial", vi,"m/s"
print *, "y un angulo de", theta, "radianes"
print *, "durara en el aire un tiempo de:", timesf,"s"
print *, "alcanzara una h maxima de:", ymaxsf,"m"
print *, "tendra un alcance maximo en el eje x de:", xmaxsf,"m"
print *, "~~~~~"
print *, "||Con resistencia del aire||"
print *, "Un tiro parabolico con:"
print *, "coordenadas iniciales x=",xi,"y=",yi
print *, "velocidad inicial", vi,"m/s"
print *, "y un angulo de", theta,"radianes"
print *, "durara en el aire un tiempo de:", timecf,"s"
print *, "alcanzara una h maxima de:", ymaxcf,"m"
print *, "tendra un alcance maximo en el eje x de:", xmaxcf,"m"
print *, "~~~~~"
print *, "La diferencia porcentual en el eje x es:", diferenciay
print *, "La diferencia porcentual en el eje y es:", diferenciay
print *, "~~~~~"

end program tiro_resistencia

subroutine sin_friccion (xi,yi,vi,theta,xmaxsf,ymaxsf,timesf)
use constantes
implicit none

real, dimension (1:npts) :: x,y,t
real :: xi, yi, vi, theta !Entran
real :: xmaxsf, ymaxsf, timesf

integer :: i

theta=theta*rad !Convirtiendo grados a radianes

xmaxsf = xi+((vi*vi*sin(2*theta))/(9.8))
ymaxsf = yi+(((vi*vi)*(sin(theta)*sin(theta)))/(19.6))

```

```

timesf = (2*vi*sin(theta))/(9.8)

!Registramos los datos calculados
open (1, file="sin_friccion.dat")

!Calculando la posicion para cada t(i)
do i=1, npts, 1
t(i)=float(i)*0.01
x(i) = xi + (vi*cos(theta)*t(i))
y(i) = yi + (vi*sin(theta)*t(i)) - (4.9*t(i)*t(i))

!Registrando resultados
write (1,1001) x(i), y(i)
1001 format (f11.5,f11.5)
if (y(i)<0) exit
end do

CLOSE (1)
end subroutine sin_friccion

subroutine con_friccion (xi,yi,vi,theta,xmaxcf,ymaxcf,timecf)
use constantes

implicit none
integer :: i

real, DIMENSION (0:npts) :: p,w,u,velp,velw,ap,aw
real :: xi, yi, vi, theta !Entrada
REAL :: xmaxcf, ymaxcf, timecf !Salida
REAL :: D, area, radio,masa !D representa una constate que depende del area,
!el coeficiente y la densidad

print *, "Deme la masa de la esfera en kg"
read *, masa

print *, "Deme el radio de la esfera en m"
read *, radio

area = pi*radio*radio

!Definiendo condiciones iniciales
p(0) = xi
w(0) = yi

```

```

velp(0) = vi*cos(theta)
velw(0) = vi*sin(theta)

D = (0.5*densidad*area*cfe)

ap(0) = -(D/masa)*velp(0)*velp(0)
aw(0) = 9.8-(D/masa)*velw(0)*velw(0)

u(0) = 0
!Abriendo archivo.dat
open (2, file="con_friccion.dat")

!Registrando valores iniciales
write(2,1001) p(0),w(0)
1001 format (f11.5,f11.5)

!Calculando posicion para cada t(i)
do i=0, npts, 1

u(i+1) = u(i) + 0.01

velp(i+1) = velp(i)+ap(i)*u(i+1)
velw(i+1) = velw(i)+aw(i)*u(i+1)

ap(i+1) = -(D/masa)*velp(i)*velp(i)
aw(i+1) = -9.8-(D/masa)*velw(i)*velw(i)

p(i+1) = p(i)+velp(i)*u(i+1)+(0.5*ap(i)*u(i+1)*u(i+1))
w(i+1) = w(i)+velw(i)*u(i+1)+(0.5*aw(i)*u(i+1)*u(i+1))

!Resgistrando valores
write (2,*) p(i+1), w(i+1)
if (w(i)<0) exit
end do

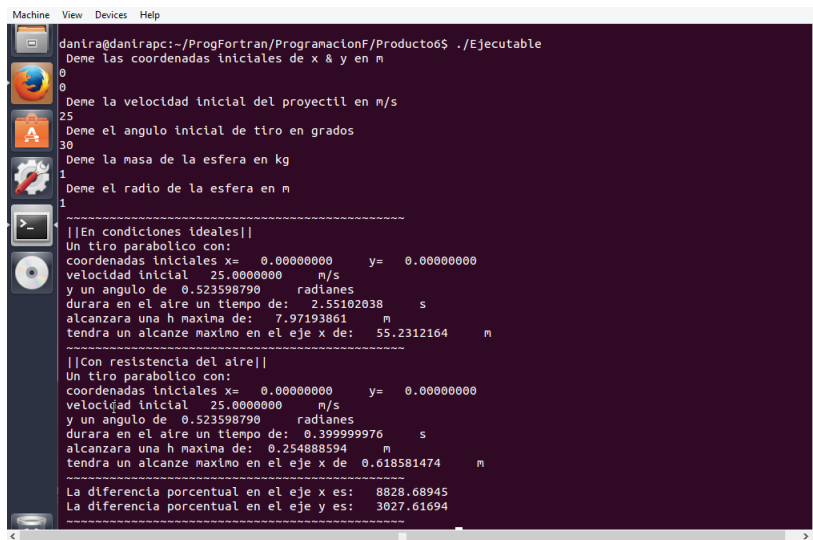
close (2)
xmaxcf = p(i)
ymaxcf = MAXVAL(w)
timecf = u(i)*10.0
end subroutine con_friccion

```

### 3.1. Resultados

Cabe denotar que en el código se cambia la salida estándar de los datos de posición a 2 archivos, llamados “sinfriccion.dat” y “confriccion.dat”, los cuales utilizamos posteriormente para graficar las coordenadas con gnuplot. Una vez concluido el programa, lo compilamos, he aquí los resultados al ejecutarlo con los ángulos de 30, 60 y 45 grados.

#### 3.1.1. 30 grados



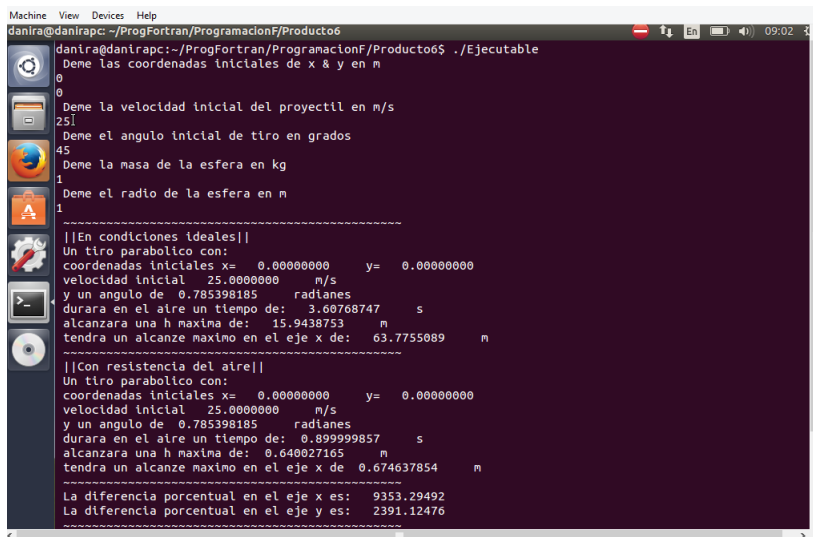
```
Machine View Devices Help
danira@danirapc:~/ProgFortran/ProgramacionF/Producto6$ ./Ejecutable
Deme las coordenadas iniciales de x & y en m
0
0
Deme la velocidad inicial del proyectil en m/s
25
Deme el angulo inicial de tiro en grados
30
Deme la masa de la esfera en kg
1
Deme el radio de la esfera en m
1

||En condiciones ideales||
Un tiro parabolico con:
coordenadas iniciales x= 0.00000000 y= 0.00000000
velocidad inicial 25.0000000 m/s
y un angulo de 0.523598790 radianes
durara en el aire un tiempo de: 2.55102038 s
alcanzara una h maxima de: 7.97193861 m
tendra un alcance maximo en el eje x de: 55.2312164 m

||Con resistencia del aire||
Un tiro parabolico con:
coordenadas iniciales x= 0.00000000 y= 0.00000000
velocidad inicial 25.0000000 m/s
y un angulo de 0.523598790 radianes
durara en el aire un tiempo de: 0.399999976 s
alcanzara una h maxima de: 0.254888594 m
tendra un alcance maximo en el eje x de: 0.618581474 m

La diferencia porcentual en el eje x es: 8828.68945
La diferencia porcentual en el eje y es: 3027.61694
```

#### 3.1.2. 45 grados



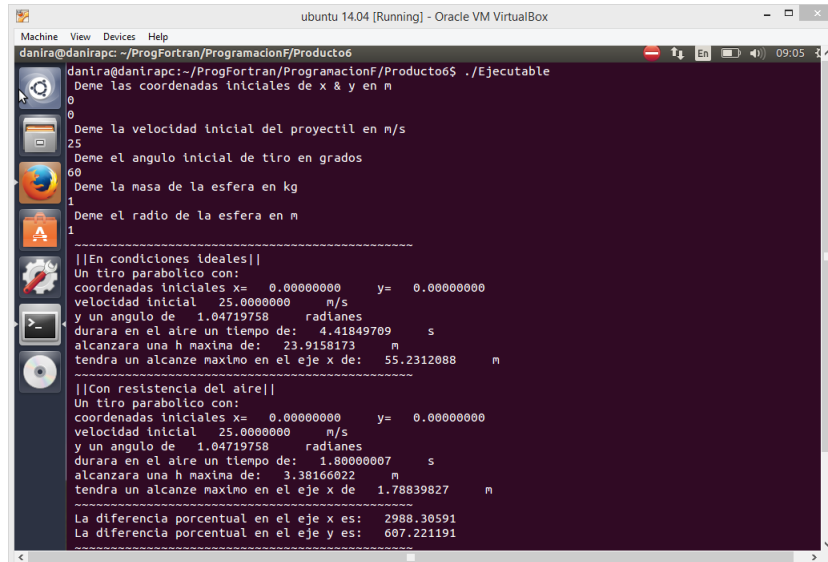
```
Machine View Devices Help
danira@danirapc:~/ProgFortran/ProgramacionF/Producto6$ ./Ejecutable
Deme las coordenadas iniciales de x & y en m
0
0
Deme la velocidad inicial del proyectil en m/s
25
Deme el angulo inicial de tiro en grados
45
Deme la masa de la esfera en kg
1
Deme el radio de la esfera en m
1

||En condiciones ideales||
Un tiro parabolico con:
coordenadas iniciales x= 0.00000000 y= 0.00000000
velocidad inicial 25.0000000 m/s
y un angulo de 0.785398185 radianes
durara en el aire un tiempo de: 3.60768747 s
alcanzara una h maxima de: 15.9438753 m
tendra un alcance maximo en el eje x de: 63.7755089 m

||Con resistencia del aire||
Un tiro parabolico con:
coordenadas iniciales x= 0.00000000 y= 0.00000000
velocidad inicial 25.0000000 m/s
y un angulo de 0.785398185 radianes
durara en el aire un tiempo de: 0.8999999857 s
alcanzara una h maxima de: 0.640027165 m
tendra un alcance maximo en el eje x de: 0.674637854 m

La diferencia porcentual en el eje x es: 9353.29492
La diferencia porcentual en el eje y es: 2391.12476
```

### 3.1.3. 60 grados

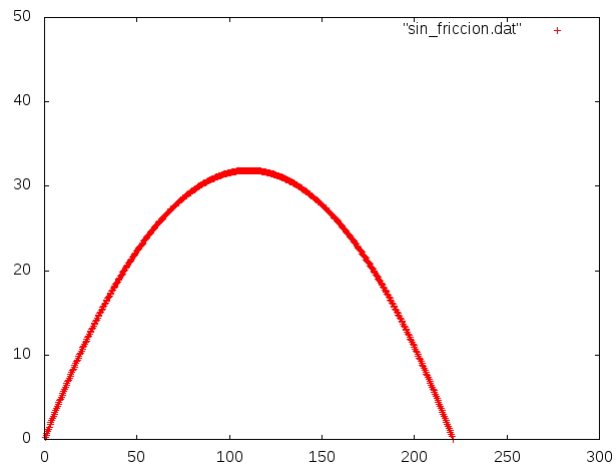


```
Machine View Devices Help
ubuntu 14.04 [Running] - Oracle VM VirtualBox
danira@danirapc: ~/ProgFortran/ProgramacionF/Producto6
danira@danirapc:~/ProgFortran/ProgramacionF/Producto6$ ./Ejecutable
Deme las coordenadas iniciales de x & y en m
0
Deme la velocidad inicial del proyectil en m/s
25
Deme el angulo inicial de tiro en grados
60
Deme la masa de la esfera en kg
1
Deme el radio de la esfera en m
1
||En condiciones Ideales||
Un tiro parabolico con:
coordenadas iniciales x= 0.00000000 y= 0.00000000
velocidad inicial 25.0000000 m/s
y un angulo de 1.04719758 radianes
durara en el aire un tiempo de: 4.41849709 s
alcanzara una h maxima de: 23.9158173 m
tendra un alcance maximo en el eje x de: 55.2312088 m
||Con resistencia del aire||
Un tiro parabolico con:
coordenadas iniciales x= 0.00000000 y= 0.00000000
velocidad inicial 25.0000000 m/s
y un angulo de 1.04719758 radianes
durara en el aire un tiempo de: 1.80000007 s
alcanzara una h maxima de: 3.38166022 m
tendra un alcance maximo en el eje x de 1.78839827 m
-----
La diferencia porcentual en el eje x es: 2988.30591
La diferencia porcentual en el eje y es: 607.221191
```

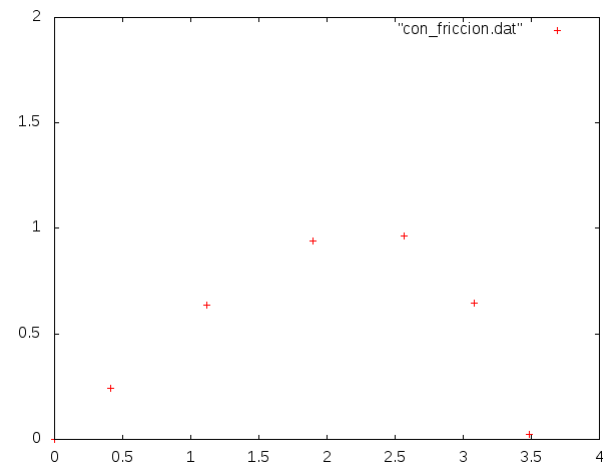
## 4. Graficando con Gnuplot

En esta parte de la práctica nos enfocamos en graficar la información que se capturo en los archivos, para eso utilizamos gnuplot. Obteniendo el siguiente resultado para cada ángulo de prueba;

### 4.1. 30 grados sin fricción

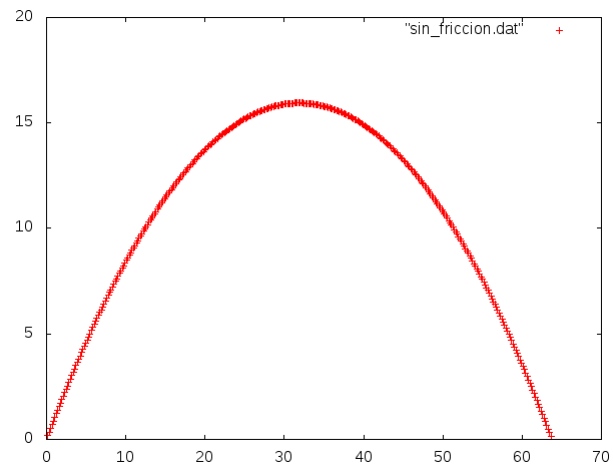


### 4.2. 30 grados con fricción

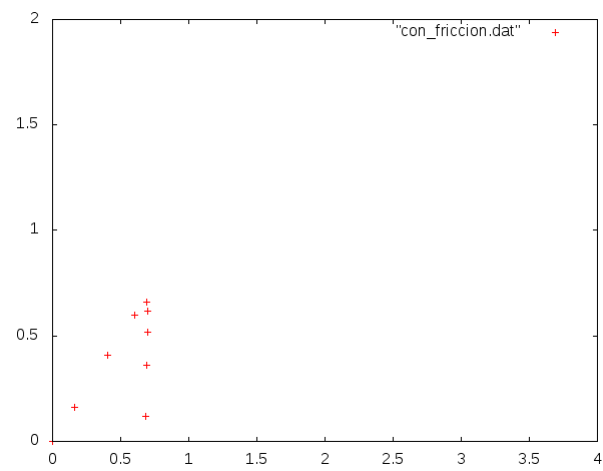




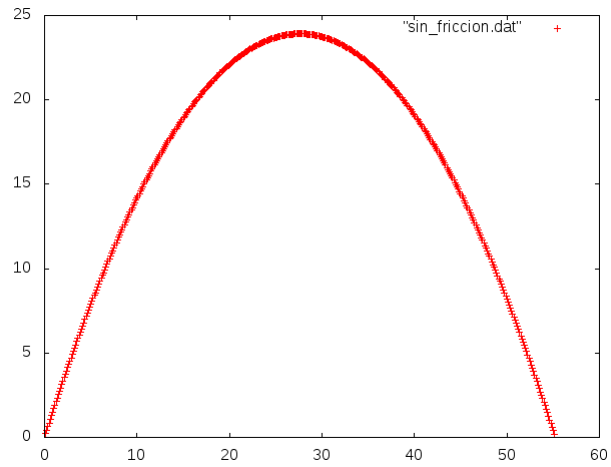
#### 4.3. 45 grados sin fricción



#### 4.4. 45 grados con fricción



#### 4.5. 60 grados sin fricción



#### 4.6. 60 grados con fricción

