

Movimiento de proyectiles incluyendo fricción del aire

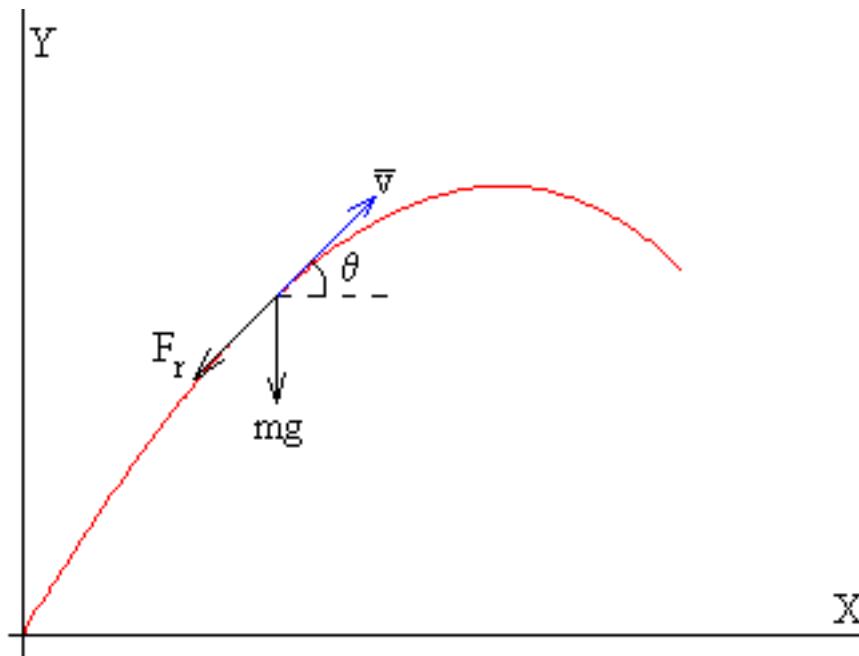
Hugo de Jesús Valenzuela Chaparro

8 de mayo de 2015

1. Tiro parabólico con fricción del aire

El estudio de la trayectoria de un proyectil es un problema que ha sido de interés por mucho tiempo. Sea con aplicaciones militares o en los deportes. Para estudiarse se separa en dos tipos de movimientos, un movimiento rectilíneo uniforme con velocidad constante en el eje x y un movimiento rectilíneo uniformemente acelerado en el eje y (con la aceleración de la gravedad). Las ecuaciones de movimiento del proyectil sin considerar la fricción están dadas por las ecuaciones: $x = v t \cos \theta$ y $y = v t \sin \theta - \frac{1}{2} g t^2$ donde x y y son las variables de posición del proyectil, v es la rapidez inicial con la que se lanzó, g la aceleración debida a la gravedad y θ el ángulo de lanzamiento inicial. Para determinar de forma unívoca la trayectoria de un proyectil, solo es necesario conocer 2 cantidades: la rapidez inicial v y el ángulo θ con el que se lanzó.

Todo objeto de masa m que se mueve a muy alta velocidad en un fluido de densidad ρ , experimenta una fuerza de arrastre F_r contraria a la dirección de su movimiento y es dada por la ecuación $F_r = \frac{1}{2} \rho v^2 C_D A$. Donde v 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. Por ejemplo, para una esfera el área transversal es $A = \pi r^2$, y el coeficiente de arrastre es $C_D = 0,47$.



2. Código en Fortran

A continuación se presenta el código de un programa en Fortran que calcula la trayectoria de un proyectil considerando la fricción del aire y también no considerándola.

```
!constantes
MODULE constantes
  IMPLICIT NONE
  real, parameter :: pi= 4.0*atan(1.0), g=9.81, den=1.19
  INTEGER, PARAMETER :: npts=5000
  !Coeficiente de arrastre de la esfera
  REAL,PARAMETER :: dc=0.47
END MODULE constantes
!Subrutina para trayectoria sin friccion
SUBROUTINE SinFriccion(x0,y0,u,a_grados,mx,my,ft)
```

```
  USE constantes
```

```

implicit none
!Defining constants:

real :: u, a, t, a_grados, my, mx, ft, Vx, Vy, FA,x0,y0

real:: x(150),y(150)
integer :: i

!where g is gravity, pi is "pi"
!u is object's initial velocity
!a is object's initial angle (grades)
!t is time during the simulation
!x and y are arrays with 150 rows
!Seek user input
!write(*,*) 'Ingresa el angulo de lanzamiento (Real)'
!read *, a_grados
!write(*,*) 'Ingresa la velocidad inicial del proyectil en m/s (Real)'
!read *, u
!Convertir angulo a radianes
a = (a_grados*pi)/180.0
!Calcular componentes de la velocidad en x (Vx) e y (Vy)
IF (a_grados == 90.0) THEN
Vx=0.0
Vy=u
ELSE IF (a_grados == 0.0) THEN
Vx=0.0
Vy=0.0
ELSE
Vx=u*cos(a)
Vy=u*sin(a)
END IF

!Calcular el tiempo que el objeto esta en el aire, siendo el mismo
!en las componentes x e y
ft=(2.0*Vy)/g

!calcular altura maxima

```

```

my=(Vy**2)/19.62

!Print results
! print*, "Velocidad inicial:", u,"m/s"
!print*, "Angulo de lanzamiento (grados):", a_grados
! print*, "Tiempo total de vuelo:", ft, "s"
! print*, "La altura maxima es:", my,"m"

!open .dat file and start writing on it using the algorithm
open(1, file='sinfriccion.dat')

do i=1,npts

    !displacement of object in x and y direction
    t = (float(i)*0.1)
    x(i) = x0+ Vx*t
    y(i) = y0+ (Vy*t)-(4.905*t*t)
    !write output in file "proj.dat" for plotting
    write(1,1001) x(i), y(i)
    1001 format (f11.5, f11.5)
    !kill the loop when the object hits the ground
    IF (y(i)<0) exit
end do
close(1)
!close file

!Desplazamiento en la direccion x hasta que el objeto toca el suelo
mx=x(i)
! print*, "El desplazamiento total en la direccion x es:", mx,"m"

END SUBROUTINE SinFriccion
!Subrutina para trayectoria con friccion del aire

```

```

SUBROUTINE ConFriccion (x0,y0,v0,a_grados,mxf,myf,ftwf)
    USE constantes
    IMPLICIT NONE
    REAL,DIMENSION (0:npts) :: xf, yf,vxf, vyf, ayf, axf, tf
    REAL :: x0,y0,v0,a_grados,mxf,myf,ftwf
    REAL :: masa,radio,a,Dom,mx,area
    INTEGER :: i
    a = (a_grados*pi)/180.0
    PRINT*, "Escribe la masa de la esfera (kg)"
    READ*, masa
    PRINT*, "Escribe el radio de la esfera (m)"
    READ*, radio
    area=pi*radio*radio
    !Condiciones iniciales
    xf(0)=x0
    yf(0)=y0
    vxf(0)=v0*cos(a)
    vyf(0)=v0*sin(a)
    Dom=((area*den*dc)/2.0)/masa
    axf(0)=-Dom*vxf(0)*vyf(0)
    ayf(0)=-g-(Dom*vyf(0)*vyf(0))
    tf(0)=0

    !open .dat file and start writing on it using the algorithm

    open(2, file='confriccion.dat')
        write (2,1001) xf(0), yf(0)
        1001 format (f11.5, f11.5)

        do i=0,npts

            tf(i+1)= tf(i)+0.01
            vxf(i+1)=vxf(i)+axf(i)*tf(i+1)
            vyf(i+1)=vyf(i)+ayf(i)*tf(i+1)
            axf(i+1)=(-Dom)*vxf(i)*vxf(i)
            ayf(i+1)=-g-(Dom*vyf(i)*vyf(i))
            xf(i+1)=xf(i)+vxf(i)*tf(i+1)+((axf(i)*tf(i+1)*tf(i+1))/2.0)

```

```

        yf(i+1)=yf(i)+vyf(i)*tf(i+1)+((ayf(i)*tf(i+1)*tf(i+1))/2.0)
        !write output in file "proj.dat" for plotting
        write(2,1001) xf(i+1), yf(i+1)
        !write (2,1001) x(i), y(i)
        !1001 format (f11.5, f11.5)
        !kill the loop when the object hits the ground
        IF (yf(i)<0) exit
    end do
    close(2)
    !close file

!Resultados
ftwf=tf(i)*10.0
myf=maxval(yf)
mxf=xf(i)

END SUBROUTINE ConFriccion

!Programa maestro
PROGRAM ProjectilesFriccion
    USE constantes
    IMPLICIT NONE
    REAL :: v0,a_grados,mx,my,ft,x0,y0,mxf,myf,ftwf,Et,Ex,Ey
    REAL,DIMENSION (0:npts) :: x, y,vx, vy, ay, ax, t
    PRINT*, "Este programa calcula el alcance en x y y en un tiro parabolico, &
    & para cuando hay friccion y cuando no la hay"
    PRINT*, "Ingresa la posicion inicial x0 y y0 (m), luego el angulo inicial (grados)
    & la velocidad inicial (m/s), respectivamente"
    READ*,x0,y0,a_grados,v0

    !Llamar subrutinas
    CALL SinFriccion(x0,y0,v0,a_grados,mx,my,ft)
    CALL ConFriccion(x0,y0,v0,a_grados,mxf,myf,ftwf)

    Et=((ABS(ftwf-ft))/ftwf)*100.0
    Ex=((ABS(mxf-mx))/mxf)*100.0
    Ey=((ABS(myf-my))/myf)*100.0

```

```

PRINT*, "Sin friccion, los resultados son:"
PRINT*, "El desplazamiento total en x es",mx, "m"
PRINT*, "La altura maxima es", my, "m"
PRINT*, "El tiempo total de vuelo es" ,ft, "s"
PRINT*, "-----"
PRINT*, "Con friccion, los resultados son:"
PRINT*, "El desplazamiento total en x es", mxf, "m"
PRINT*, "La altura maxima es", myf, "m"
PRINT*, "el tiempo total de vuelo es", ftwf, "s"
PRINT*, "-----"
PRINT*, "Error relativo poncertual al no considerar la friccion del aire:"
PRINT*, "En el desplazamiento x es del",Ex,"%"
PRINT*, "En el desplazamiento y es del",Ey,"%"
PRINT*, "En el tiempo total de vuelo es del",Et,"%"
END PROGRAM ProjectilesFriccion

```

3. Resultados del programa corriendo

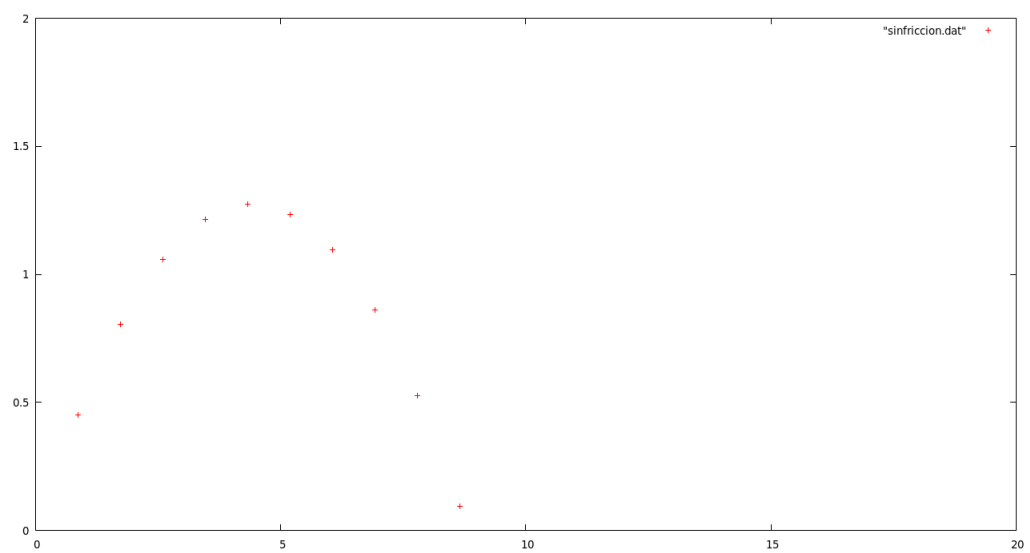
3.1. $\theta = 30$

```

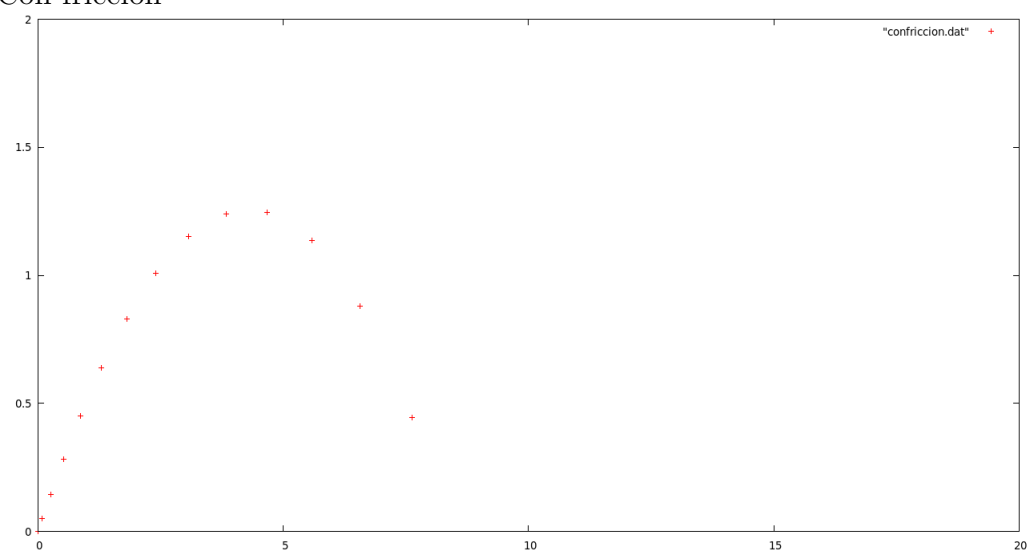
grunge@grunge-Inspiron-N5010:~/ProgFortran/ProgramacionF/Producto6$ ./ProjectilesFriccion
Este programa calcula el alcance en x y y en un tiro parabolico, para cuando hay friccion y cuando no la hay
Ingresa la posicion inicial x0 y y0 (m), luego el angulo inicial (grados) y la velocidad inicial (m/s), respectivamente
0
0
30
10
Escribe la masa de la esfera (kg)
0.25
Escribe el radio de la esfera (m)
0.05
Sin friccion, los resultados son:
El desplazamiento total en x es  9.52627945      m
La altura maxima es  1.27420998      m
El tiempo total de vuelo es  1.01936793      s
-----
Con friccion, los resultados son:
El desplazamiento total en x es  8.74584675      m
La altura maxima es  1.24483430      m
el tiempo total de vuelo es  1.39999986      s
-----
Error relativo poncertual al no considerar la friccion del aire:
En el desplazamiento x es del  8.92346668      %
En el desplazamiento y es del  2.35980582      %
En el tiempo total de vuelo es del  27.1879978      %
grunge@grunge-Inspiron-N5010:~/ProgFortran/ProgramacionF/Producto6$ █

```

Sin fricción



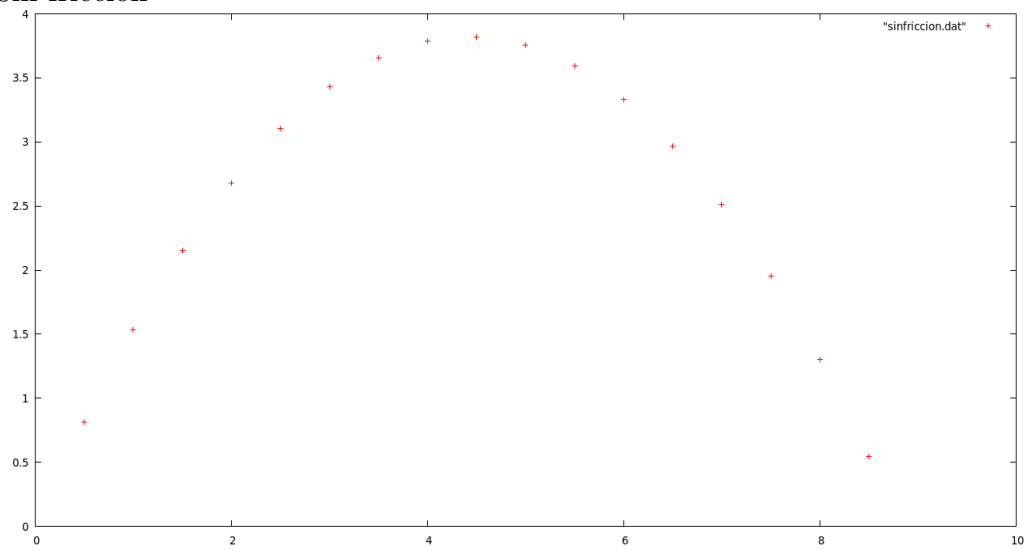
Con fricción



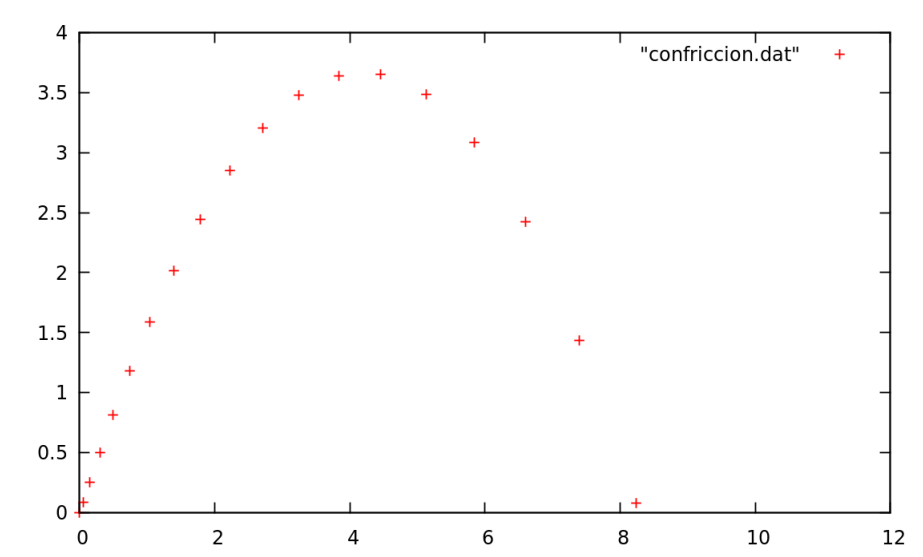
3.2. $\theta = 60$

```
grunge@grunge-Inspiron-N5010:~/ProgFortran/ProgramacionF/Producto6$ ./ProyectilesFriccion
Este programa calcula el alcance en x y y en un tiro parabolico, para cuando hay friccion y cuando no la hay
Ingresa la posicion inicial x0 y y0 (m), luego el angulo inicial (grados) y la velocidad inicial (m/s), respectivamente
0
0
60
10
Escribe la masa de la esfera (kg)
0.25
Escribe el radio de la esfera (m)
0.05
Sin friccion, los resultados son:
El desplazamiento total en x es 8.99999905 m
La altura maxima es 3.82263017 m
El tiempo total de vuelo es 1.76559722 s
-----
Con friccion, los resultados son:
El desplazamiento total en x es 9.11707115 m
La altura maxima es 3.65275407 m
el tiempo total de vuelo es 1.90000010 s
-----
Error relativo poncertual al no considerar la friccion del aire:
En el desplazamiento x es del 1.28409779 %
En el desplazamiento y es del 4.65063047 %
En el tiempo total de vuelo es del 7.07383537 %
grunge@grunge-Inspiron-N5010:~/ProgFortran/ProgramacionF/Producto6$
```

Sin fricción



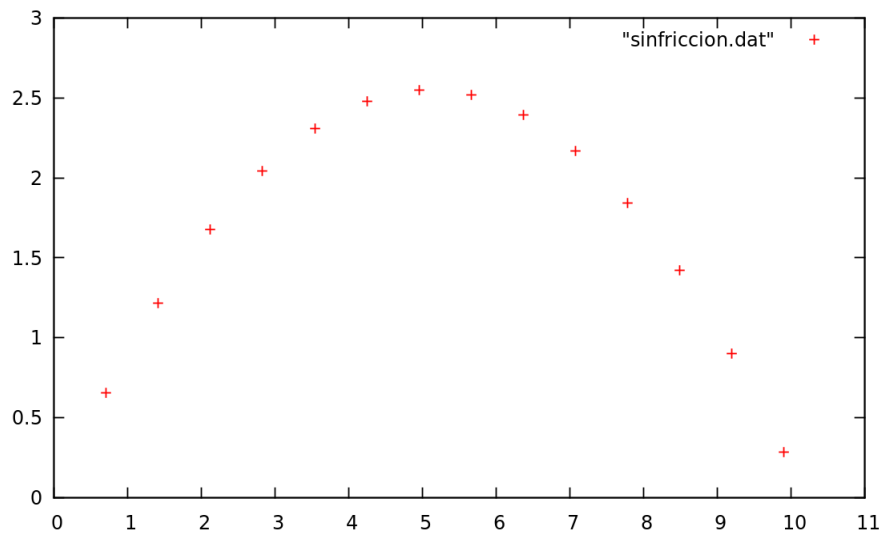
Con fricción



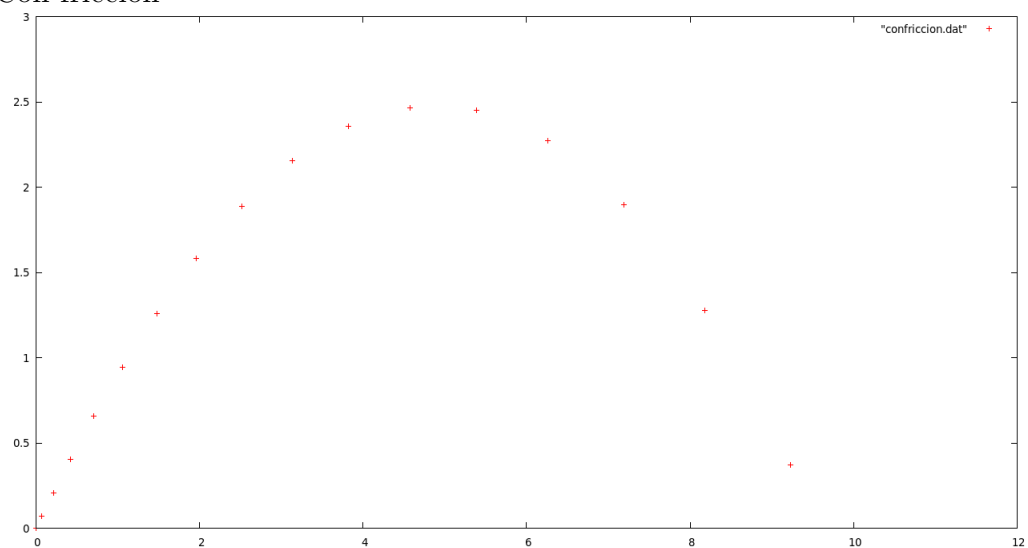
3.3. $\theta = 45$

```
grunge@grunge-Inspiron-N5010:~/ProgFortran/ProgramacionF/Producto6$ ./ProyectilesFriccion
Este programa calcula el alcance en x y en un tiro parabolico, para cuando hay friccion y cuando no la hay
Ingresa la posicion inicial x0 y y0 (m), luego el angulo inicial (grados) y la velocidad inicial (m/s), respectivamente
0
0
45
10
Escribe la masa de la esfera (kg)
0.25
Escribe el radio de la esfera (m)
0.05
Sin friccion, los resultados son:
El desplazamiento total en x es 10.6066017 m
La altura maxima es 2.54841995 m
El tiempo total de vuelo es 1.44160402 s
-----
Con friccion, los resultados son:
El desplazamiento total en x es 10.3278751 m
La altura maxima es 2.46631026 m
el tiempo total de vuelo es 1.70000005 s
-----
Error relativo poncertual al no considerar la friccion del aire:
En el desplazamiento x es del 2.69877958 %
En el desplazamiento y es del 3.32925200 %
En el tiempo total de vuelo es del 15.1997652 %
grunge@grunge-Inspiron-N5010:~/ProgFortran/ProgramacionF/Producto6$
```

Sin fricción



Con fricción



Los resultados tiene sentido, pues cuando hay fricción se puede apreciar que las distancias máximas en x e y son menores que cuando no la hay. Además el tiempo total de vuelo es mayor cuando hay fricción, y en el programa corriendo puede verse, esto se debe a que cuando el objeto va cayendo el aire actúa como “paracaídas” debido a la fricción generada.