

Proyectiles

José Ramón Pérez Navarro

06 Septiembre 2019

Introducción

En esta actividad realizada durante la presente semana, aplicamos el concepto "do-loops" para estructurar procesos repetitivos, así como también, se leyeron entradas y salidas de datos de un archivo de texto a otro utilizando la instrucción "open" para tratar de replicar el movimiento durante el lanzamiento de un proyectil, el ángulo, la posición en (x, y) y el cambio de las mismas en determinado instante de tiempo.

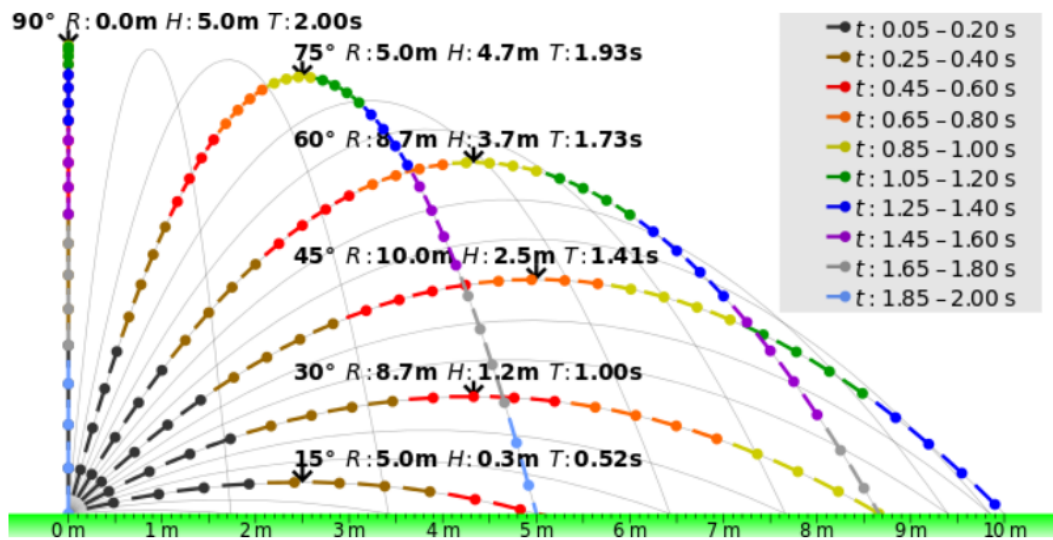


Figura 1:

Procedimiento

1. Se utilizo el programa "proyectil.f90" que leía la velocidad inicial u , el ángulo α y el tiempo total de vuelo t .
2. Se elaboro una estructura do-loop, para que el programa escribiese en la pantalla la posición (x,y) del proyectil cada 0.1 segundos para un ángulo específico.
3. Se definió un archivo de nombre "*salida.dat*" con la instrucción "*open*", donde se escribieron las posiciones (x,y) , para cada instante de tiempo.
4. Mediante Gnuplot, se gráfico el archivo "*salida.dat*". Desde una terminal se escribió "*gnuplotz*" dentro de la interfaz de Gnuplot, se ingresaron los siguientes comandos: *plot "salida.dat" with lines*, de esta manera se consiguió el gráfico
5. Posteriormente se incluyo otra estructura "loop" de manera que repitiese los cálculos anteriores, pero para 6 ángulos distintos: 15, 30, 45, 60, 75 y 90 grados. Los datos de las posiciones (x,y) , se escribieron también al archivo "*salida.dat*"
6. Se gráfico lo obtenido en el paso anterior

Con el siguiente código:

```
1 program projectile
2   implicit none
3   ! definimos constantes
4   real, parameter :: g = 9.8
5   real, parameter :: pi = 3.1415927
6   integer :: n, npasos, m
7   ! definimos las variables
8   real :: a, t, u, x, y, dt, angulo, dangulo
9   ! output data into a file
10  open(11, file = 'salida.dat', status = 'unknown')
11  ! Leer valores la velocidad inicial u, y el tiempo de vuelo y el angulo a desde la terminal
12  write(*,*) 'Dame la velocidad inicial, el angulo y el numero de pasos'
13  read(*,*) u, a, npasos
14  ! convirtiendo ángulo a radianes
15  a = a * pi / 180.0
16  dt = 0.1
17  ! Loop
18  dangulo = 15.0
19  dangulo = dangulo * pi / 180.0
20  do m = 1, 6
21    angulo = float(m) * dangulo
22
23    do n = 0, npasos
24      t = float(n) * dt
25      x = u * cos(angulo) * t
26      y = u * sin(angulo) * t - 0.5 * g * t * t
27
28      print *, n, x, y
29      write(11,*) x, y
30
31      if (y < 0.0) exit
32    end do
33    write(11,*) " "
34  end do
35 end program projectile
```

Figura 2:

Se obtuvo el siguiente gráfico:

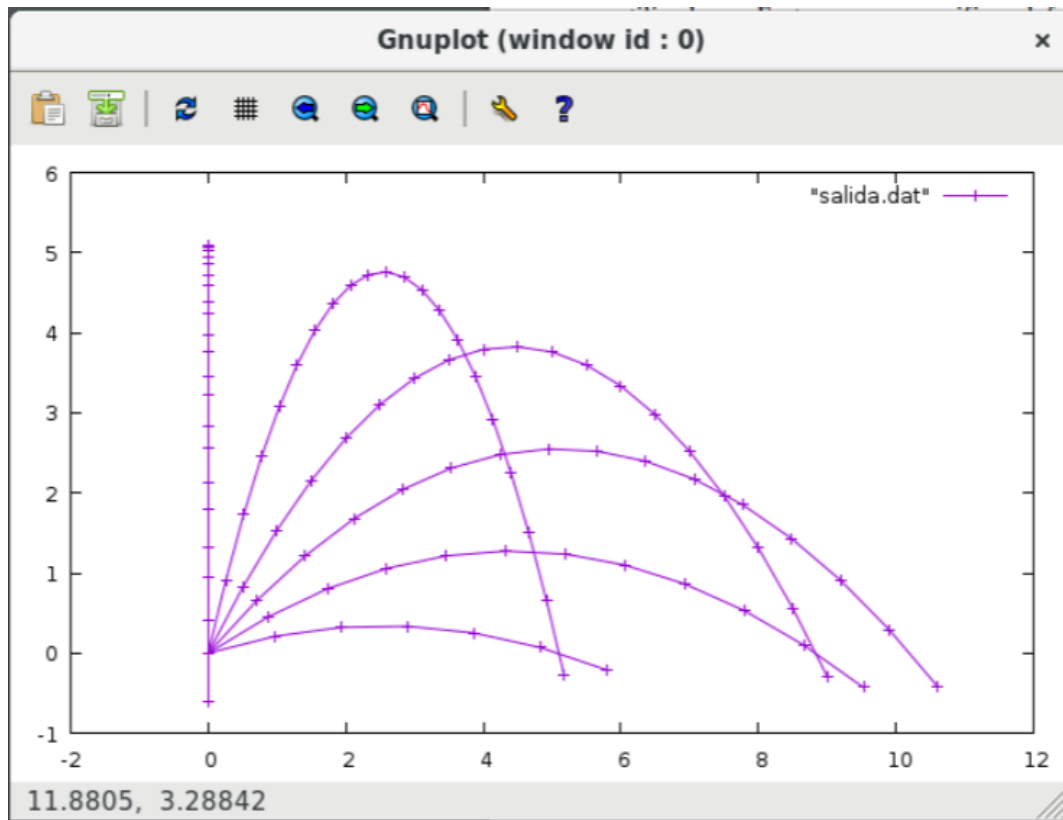


Figura 3:

Conclusión

En esta actividad se aprendió lo básico de hacer "do-loops" para hacer conteos, repeticiones, bucles, etc. También se aprendió a leer las entradas y salidas de datos con el comando `.openz` como obtener las gráficas de estos datos mediante la ejecución del programa de "gnuplot" desde la terminal.