

Do-Loops y escribiendo en archivos en Fortran

Carlos David Ureña Pérez

23 de Octubre del 2019

1 Introducción

La actividad de la semana en palabras del profesor: "En esta actividad aplicaremos el concepto de do-loops, estructurar procesos repetitivos, así como también aprenderemos a leer entradas y escribir la salida de los datos, de un archivo de texto a otro, utilizando la instrucción de open de Fortran. En forma paralela, también aprenderemos a utilizar el gratificador Gnuplot, para intentar reproducir la gráfica que aparece al inicio de la actividad y tomada del artículo de la Wikipedia sobre el movimiento de proyectiles". Por lo que se creó un programa el cual calculará la posición x y y en diferentes instantes de tiempo.

2 Actividad

1. Elabora un programa que lea la velocidad inicial u y el tiempo total de vuelo t .
2. Estructura tu primer do-loop, para que el programa escriba en la pantalla la posición (x,y) del proyectil cada 0.1 segundos para un ángulo específico.
3. Define un archivo "salida.dat" con la instrucción open, donde se escribirán las posiciones (x,y) , para cada instante de tiempo. Verifica posteriormente que el archivo contiene resultados coherentes.
4. Utiliza Gnuplot, para graficar el archivo "salida.dat". Desde una terminal ejecuta el programa gnuplot. Dentro de la interfaz de Gnuplot, escribe: `plot "salida.dat" with lines`
5. Logrado lo anterior, ahora incluye otra estructura loop que repita los cálculos anteriores, pero para 6 ángulos distintos: 15, 30, 45, 60, 75 y 90 grados. Los datos de las posiciones (x,y) , se escribirán también al archivo "salida.dat"
6. Utiliza Gnuplot para graficar de nuevo el archivo "salida.dat". Busca en Internet cómo graficar varias curvas con Gnuplot y agregar etiquetas, para tratar de reproducir la figura de la Wikipedia.

3 Elaboración del Programa parte 1: Do loop

Para cumplir con el segundo punto de la actividad se elaboró el programa utilizando el siguiente código:

```
program projectile
implicit none
!definimos constantes
real, parameter :: g=9.80665
real, parameter :: pi=3.1415927
real, parameter :: dt= 0.1
!definimos variables
real :: angulo, t, v0, x, y
integer :: n, m
!Leer valores para el tiempo total y la rapidez inicial desde la terminal
write(*,*) "Dame el tiempo total y la rapidez inicial"
read(*,*) t, v0
angulo=45
!Convirtiendo el angulo a radianes
angulo= angulo * pi / 180.0
DO n=0, 100
t=float(n)*dt
! Las ecuaciones de la posición de la posición en x y y
x = v0 * cos(angulo) * t
y = v0 * sin(angulo) * t - 0.5 * g * t * t
!escribiendo el resultado en la pantalla
print*, x, y
END DO
end program projectile
```

4 Elaboración del Programa parte 2: Captura de datos utilizando un OPEN

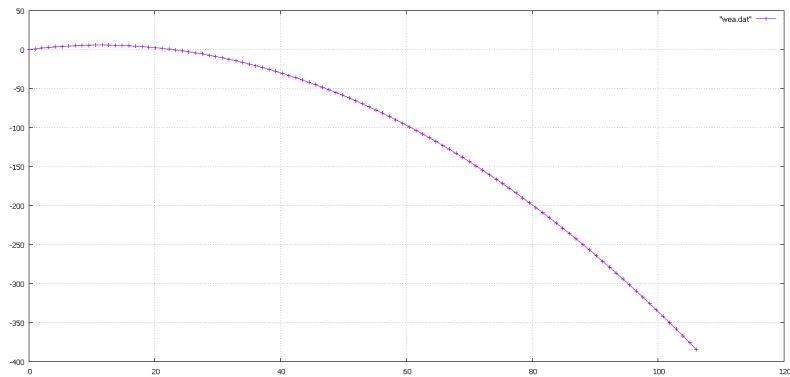
```
program projectile
implicit none
!definimos constantes
real, parameter :: g=9.80665
real, parameter :: pi=3.1415927
real, parameter :: dt= 0.1
!definimos variables
real :: angulo, t, v0, x, y
integer :: n, m
!Leer valores para el tiempo total y la rapidez inicial desde la terminal
write(*,*) "Dame el tiempo total y la rapidez inicial"
read(*,*) t, v0
angulo=45
```

```

OPEN(unit=5, file="wea.dat", access="Append")
!Convirtiendo el angulo a radianes
angulo= angulo * pi / 180.0
DO n=0, 100
  t=float(n)*dt
  ! Las ecuaciones de la posición de la posición en x y y
  x = v0 * cos(angulo) * t
  y = v0 * sin(angulo) * t - 0.5 * g * t * t
  !escribiendo el resultado en la pantalla
  write(5,*) x, y
  print*, x, y
END DO
close (5)
end program projectile

```

5 Grafica del programa utilizando gnuplot



6 Do loop para ángulos de 15, 30, 45, 60, 75 y 90 grados

```

program projectile
  implicit none
  ! definimos constantes
  real, parameter :: g = 9.80665
  real, parameter :: pi = 3.1415927
  real, parameter :: dt= 0.1
  ! definimos las variables
  real :: angulo, t, v0, x, y
  INTEGER :: n, m
  ! Leer valores para el ángulo a, el tiempo t, y la velocidad inicial u desde la
terminal
  write(*,*) 'Dame el tiempo total y la rapidez inicial'

```

```

read(*,*) t, v0
angulo=45
OPEN(unit=5,file="salida.dat",access="Append")
DO m=15, 90, 15
  angulo=float(m)
  ! convirtiendo ángulo a radianes
  angulo = angulo * pi / 180.0
  DO n=0, 100
    t=float(n)*dt
    ! las ecuaciones de la posición en x y y
    x = v0 * cos(angulo) * t
    y = v0 * sin(angulo) * t - 0.5 * g * t * t
    ! escribiendo el resultado en la pantalla
    write(5,*) x, y
    PRINT*, x, y
    IF(y<0 .AND. t/=0) EXIT
  END DO
END DO
close (5)
end program projectile

```

7 Utilizar Gnuplot para graficar los 6 ángulos

