Actividad 6

Paula Moreno García

Noviembre 2018

1 Introducción

En el próximo documento se presentará el método de Runge-Kutta de orden 4 o también llamado RK4. Se utilizará un código con fortran del mismo método para resolver la ecuación del péndulo. De igual fomra, se presentarán gráficas de cada uno de los ángulos.

1.1 Método

El método Rk4 es uno de los métodos más comúnmente utilizados de la forma Runge-Kutta. La solución que ofrece este método es una tabla con la función que genera solución, con valores "y" correspondientes a valores específicos de "x". Se necesita lo siguiente:
1.- Especificar el intervalo de x.
2.- Una ecuación diferencial de primer orden. $y' = f(x, y)$
3.- Conocemos un punto inicial de y que sería x_0. El método RK4 consiste en determinar constantes apropiadas de modo que una fórmula como: $y_i + 1 = y_1 + ak_1 + bk_2 + ck_3 + dk_4$
coincida con un desarrollo de Taylor hasta el término de h4. Se conocen a las ecuaciones del método:

$$k_1 = hf(x_i, y_i)$$

$$k_2 = hf(x_i + \frac{1}{2}\ h, y_i + \frac{1}{2}k_1)$$

$$k_3 = hf(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_2)$$

$$k_4 = hf(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_3)$$

Ahora resolveremos la ecuación del péndulo con 15, 30, 45, 60, y 75 grados como ángulos. Yo utilicé una longitud de cuerda de 10 cm, elegí un tiempo de oscilación de 30 segundos y el ancho de paso fue de .1.

```
Program Pendulo
IMPLICIT NONE
```

```
Real :: l, a, h, m, Ang_0, grados
!Variables para rk4
Real :: k1, k2, k3, k4, l1, l2, l3, l4
Real :: aux, aux1, aux2, aux3, aux4
!Vectores sin dimension
Real, allocatable :: t(:), W(:), Teta(:), Ang(:)
Integer :: i, n
real, external :: func
character:: output2*12

Print*, "longitud de la cuerda"
Read*, l

Print*, "Angulo inicial del pendulo"
read*,grados
Ang_0= (3.1416*grados)/180

Print*, "Tiempo de oscilacion"
Read*, a

Print*, "Ancho de paso"
Read*, h

print*,"nombre archivo de salida t vs grados"
read*,output2

!SE calcula el numero de particiones
m=a/h
!se toma un numero entero de particiones
n=NINT(m)

!Se le da dimension a los vectores

!Valores del angulo(radianes)
Allocate(Teta(n))
!Valores del angulo(grados)
Allocate(Ang(n))
!Valores del tiempo
Allocate(t(n))
!Valores de la velocidad angular
Allocate(W(n))


Print*, "Gracias!"

!Ponemos valores iniciales en los arreglos
```

```
Teta(1)=Ang_0
Ang(1)=grados

t(1)=0
W(1)=0


!!!!
Do i=2,n

!Primer pendiente
k1= h*W(i-1)
l1= h*func(Teta(i-1),l)

!Segunda pendiente
aux2= Teta(i-1)+(k1/2)
k2= h*(W(i-1)+(l1/2))
l2= h*func(aux2,l)

!tercer pendiente
aux3= Teta(i-1)+(k2/2)
k3= h*(W(i-1)+(l2/2))
l3= h*func(aux3,l)

!cuarta pendiente
aux4= Teta(i-1)+k3
k4= h*(W(i-1)+l3)
l4= h*func(aux4,l)

!hacemos una suma para teta
Aux= k1+(2*k2)+(2*k3)+k4
!hacemos una suma para la rapidez angular
Aux1= l1+(2*l2)+(2*l3)+l4

!Calculamos las nuevas rapidez y angulo
W(i)= W(i-1) + (aux1/6)
Teta(i)= Teta(i-1) + (aux/6)
Ang(i) = Teta(i)*(180/3.1416)
!aux/6 es el promedio de las pendientes

!Calcula paso del tiempo
t(i)=h*(i-1)
```

```
End do

Open(3,file=output2)
Do i=1,n
Write(3,*)t(i), Ang(i)

End do
Close (3)

End program Pendulo

Function func(Teta,l)
implicit none
Real :: Teta, func, l

!Esta funcion se usa con angulos grandes
func=(-9.81/l)*(Sin(Teta))
```

1.2 Gráficas

A continuación. se presentan las gráficas con los diferentes ángulos en orden de menor a mayor iniciando con 15 grados:

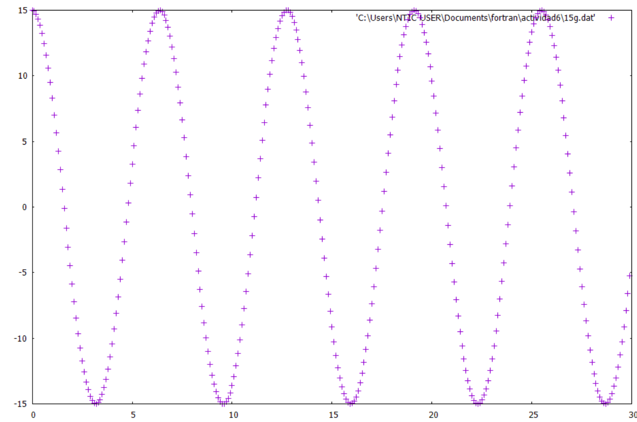

Figure 1: Gráfica con 15 grados

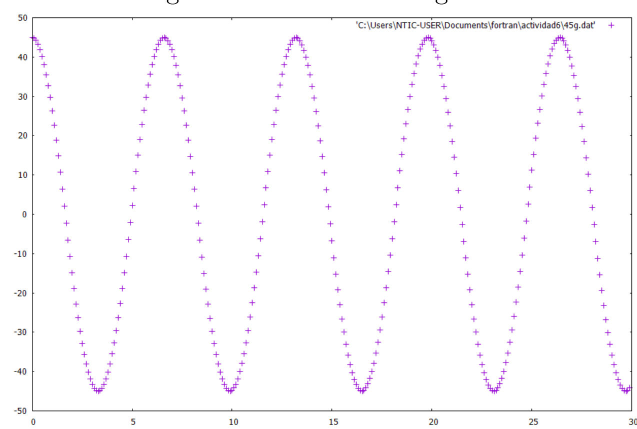

Figure 2: Gráfica con 45 grados

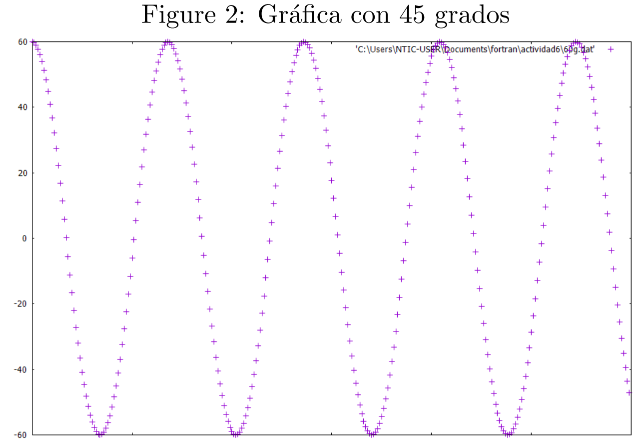

Figure 3: Gráfica con 60 grados

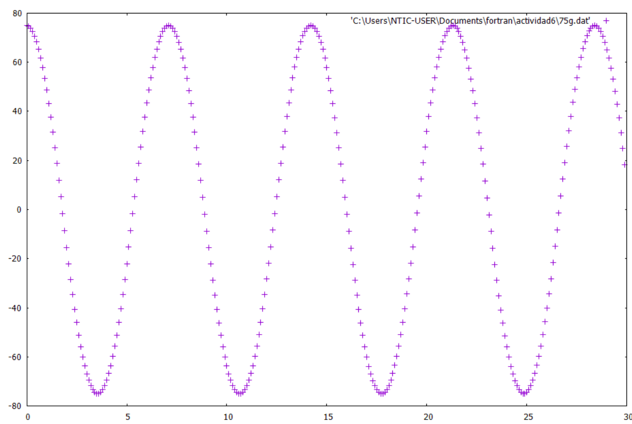

Figure 4: Gráfica con 75 grados