

```

PROGRAM Euler_Cromer
!*****
! Se resuelve el pendulo no-lineal, amortiguado y con forzamiento
!
!
!*****

REAL*8, DIMENSION(:), ALLOCATABLE :: theta,omega,t
REAL*8 :: length,dt
!

!print*,"numero de pasos"
!read*, n
n = 10000
ALLOCATE (theta(0:n),omega(0:n),t(0:n))
!
!
call inicializa(theta, omega, t, n, length, dt)
call calcula (theta, omega, t, n, length, dt)
call despliega (theta, omega, t, n, length, dt)
!
END PROGRAM Euler_Cromer
!
!
SUBROUTINE inicializa(theta, omega, t, n ,length, dt)
  INTEGER, INTENT (IN) :: n
  REAL*8, DIMENSION(0:n) :: theta,omega,t
  REAL*8 :: length,dt
  !print*, 'Angulo inicial del pendulo (en radianes)'
  !read*, theta(0)
  theta(0) = 0.2d0
  !print*, 'Velocidad angular inicial del pendulo (en radianes/s)'
  !read*, omega(0)
  omega(0) = 0.d0
  t(0)=0.d0
  !print*, 'Longitud del pendulo (in m)'
  !read*, length
  length = 9.80d0
  !print*, 'Tamaño de paso (en segundos)'
  !read*, dt
  dt=0.001
END SUBROUTINE inicializa
!
!
SUBROUTINE calcula(theta, omega, t, n, length, dt)
  INTEGER, INTENT (IN) :: n
  REAL*8, DIMENSION(0:n) :: theta,omega,t

```

```

REAL*8 :: length,dt,g,periodo,k1,k2,l1,l2
INTEGER :: i
PI= 4.*ATAN(1.)
i= 0
g= 9.80d0
q=1/2.0d0
!print*," Amplitud de la fuerza"
!read*, df
df = 0.d0
!df=0.0d0, 0.5d0, 1.20d0
dfr=2/3.d0
DO
t(i+1) = t(i) + dt
k1 = (-1)*(g/length) *sin(theta(i)) * dt + q * omega(i)*dt+df*sin(dfr*t(i))*dt
l1 = omega(i) * dt
k2 = (-1)*(g/length) *sin(theta(i)+l1) * dt + q * (omega(i)+k1)*dt+df*sin(dfr*t(i+1))
l2 = (omega(i)+k1) * dt
omega(i+1) = omega(i) + (0.5d0)*(k1+k2)
theta(i+1) = theta(i) + (0.5d0)*(l1+l2)
!omega(i+1) = omega(i) - (g/length) *sin(theta(i)) * dt + q * omega(i)*dt+df*sin(dfr
!theta(i+1) = theta(i) + omega(i+1) * dt ! Metodo de Cromer
if (theta(i+1) > PI ) theta(i+1)=theta(i+1)-2.*PI
if (theta(i+1) < -PI) theta(i+1)=theta(i+1)+2.*PI
IF (i >= n-1) EXIT
i=i+1
ENDDO
END SUBROUTINE calcula
SUBROUTINE despliega(theta, omega, t, n, length, dt)
INTEGER, INTENT (IN) :: n
REAL*8, DIMENSION(0:n) :: theta,omega,t
REAL*8 :: length,dt
INTEGER :: i
CHARACTER(LEN=10), PARAMETER :: f1 = '(3ES16.6)'
CHARACTER(10) :: archivo
!print*," archivo de datos"
!read*, archivo
archivo = "pen.dat"
OPEN (UNIT=1,FILE=archivo,STATUS='UNKNOWN')
!
WRITE(1,f1)(theta(i),omega(i),t(i), i=0,n)
!
CLOSE(1)
END SUBROUTINE despliega

```

```

misael@misael-VirtualBox:~/Descargas$ gfortran penduloRKII.f90
misael@misael-VirtualBox:~/Descargas$ ./a.out
misael@misael-VirtualBox:~/Descargas$ gnuplot

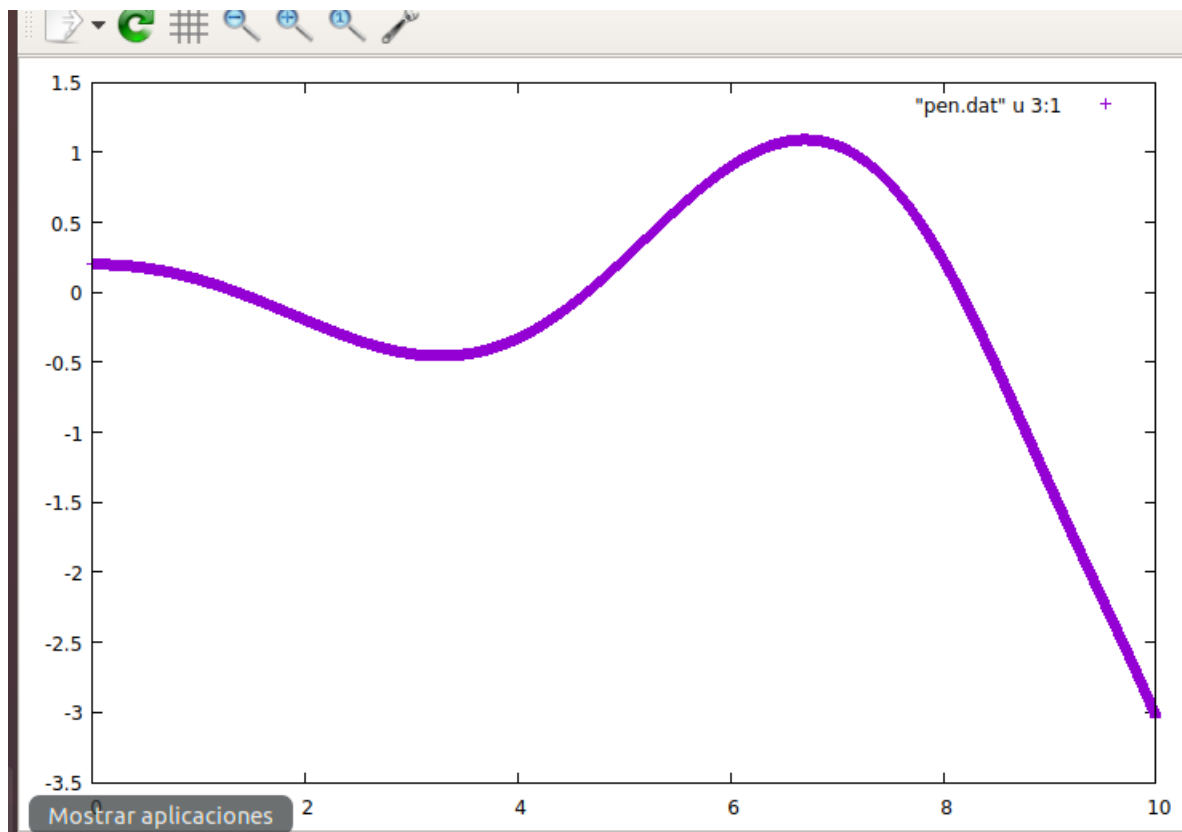
G N U P L O T
Version 5.2 patchlevel 2    last modified 2017-11-01

Copyright (C) 1986-1993, 1998, 2004, 2007-2017
Thomas Williams, Colin Kelley and many others

gnuplot home:      http://www.gnuplot.info
faq, bugs, etc:    type "help FAQ"
immediate help:    type "help" (plot window: hit 'h')

Terminal type is now 'qt'
gnuplot> p "pen.dat" u 3:1
gnuplot>

```



```

PROGRAM Euler_Cromer

```

```

!*****
! Se resuelve el pendulo no-lineal, amortiguado y con forzamiento
!
!
!*****

```

```

REAL*8, DIMENSION(:), ALLOCATABLE :: theta,omega,t
REAL*8 :: length,dt
!

```

```

!print*,"numero de pasos"
!read*, n
n = 10000
ALLOCATE (theta(0:n),omega(0:n),t(0:n))
!
!
call inicializa(theta, omega, t, n, length, dt)
call calcula (theta, omega, t, n, length, dt)
call despliega (theta, omega, t, n, length, dt)
!
END PROGRAM Euler_Cromer
!
!
SUBROUTINE inicializa(theta, omega, t, n ,length, dt)
  INTEGER, INTENT (IN) :: n
  REAL*8, DIMENSION(0:n) :: theta,omega,t
  REAL*8 :: length,dt
  !print*, 'Angulo inicial del pendulo (en radianes)'
  !read*, theta(0)
  theta(0) = 0.2d0
  !print*, 'Velocidad angular inicial del pendulo (en radianes/s)'
  !read*, omega(0)
  omega(0) = 0.d0
  t(0)=0.d0
  !print*, 'Longitud del pendulo (in m)'
  !read*, length
  length = 9.80d0
  !print*, 'Tamaño de paso (en segundos)'
  !read*, dt
  dt=0.001
END SUBROUTINE inicializa
!
!
SUBROUTINE calcula(theta, omega, t, n, length, dt)
  INTEGER, INTENT (IN) :: n
  REAL*8, DIMENSION(0:n) :: theta,omega,t
  REAL*8 :: length,dt,g,periodo,k1,k2,k3,k4,l1,l2,l3,l4
  INTEGER :: i
  PI= 4.*ATAN(1.)
  i= 0
  g= 9.80d0
  q=1/2.0d0
  !print*, " Amplitud de la fuerza"
  !read*, df
  df = 0.d0
  !df=0.0d0, 0.5d0, 1.20d0

```

```

dfr=2/3.d0
D0
t(i+1) = t(i) + dt
k1 = (-1)*(g/length) *sin(theta(i)) * dt + q * omega(i)*dt+df*sin(dfr*t(i))*dt
l1 = omega(i) * dt
k2 = (-1)*(g/length) *sin(theta(i)+(l1*(0.5d0))) * dt + q * (omega(i)+(k1*(0.5d0)))*
l2 = (omega(i)+(k1*(0.5d0))) * dt
k3 = (-1)*(g/length) *sin(theta(i)+(l2*(0.5d0))) * dt + q * (omega(i)+(k2*(0.5d0)))*
l3 = (omega(i)+(k2*(0.5d0))) * dt
k4 = (-1)*(g/length) *sin(theta(i)+(l3*(0.5d0))) * dt + q * (omega(i)+(k3*(0.5d0)))*
l4 = (omega(i)+(k3*(0.5d0))) * dt
omega(i+1) = omega(i) + (1/6d0)*(k1+2*k2+2*k3+k4)
theta(i+1) = theta(i) + (1/6d0)*(l1+2*l2+2*l3+l4)
!omega(i+1) = omega(i) - (g/length) *sin(theta(i)) * dt + q * omega(i)*dt+df*sin(dfr
!theta(i+1) = theta(i) + omega(i+1) * dt ! Metodo de Cromer
if (theta(i+1) > PI ) theta(i+1)=theta(i+1)-2.*PI
if (theta(i+1) < -PI) theta(i+1)=theta(i+1)+2.*PI
IF (i >= n-1) EXIT
i=i+1
ENDDO
END SUBROUTINE calcula
SUBROUTINE despliega(theta, omega, t, n, length, dt)
  INTEGER, INTENT (IN) :: n
  REAL*8, DIMENSION(0:n) :: theta,omega,t
  REAL*8 :: length,dt
  INTEGER :: i
  CHARACTER(LEN=10), PARAMETER :: f1 = '(3ES16.6)'
  CHARACTER(10) :: archivo
  !print*," archivo de datos"
  !read*, archivo
  archivo = "pen.dat"
  OPEN (UNIT=1,FILE=archivo,STATUS='UNKNOWN')
  !
  WRITE(1,f1)(theta(i),omega(i),t(i), i=0,n)
  !
  CLOSE(1)
END SUBROUTINE despliega

```

```
misael@misael-VirtualBox:~/Descargas$ gfortran penduloRKIV.f90
misael@misael-VirtualBox:~/Descargas$ ./a.out
misael@misael-VirtualBox:~/Descargas$ gnuplot
```

```
GNUPLOT
Version 5.2 patchlevel 2    last modified 2017-11-01

Copyright (C) 1986-1993, 1998, 2004, 2007-2017
Thomas Williams, Colin Kelley and many others

gnuplot home:      http://www.gnuplot.info
faq, bugs, etc:    type "help FAQ"
immediate help:    type "help" (plot window: hit 'h')
```

```
Terminal type is now 'qt'
gnuplot> p "pen.dat" u 3:1
gnuplot> exit
```

