Evaluacion 02

Santos Valenzuela

Diciembre 2019

1 Introduccion

En la segunda evaluacion requerimos realizar dos problemas.

2 Primer problema

Nos dan el siguiente programa y nos piden corregirlo para que funcione

```
PROGRAM Triangle
     IMPLICIT NONE
     REAL :: a, b, c, Area
     PRINT *, ’Welcome, please enter the&
              &lengths of the 3 sides.’
     READ *, a, b
     PRINT *, ’Triangle’’s area:  ’’, Area(a,b,c)
    END PROGRAM Triangle
    FUNCTION Area(x,y,z)
     IMPLICIT NONE
     REAL :: Area            ! function type
     REAL, INTENT( IN ) :: x, y, z
     REAL :: theta, height
     theta = ACOS(((x**2+y**2-z**2)/(2.0*x*y))
     height = x*SIN(theta); Area = 0.5*y*height
    END FUNCTION Area
```

Tras realizar las correcciones pertinentes, el programa quedo de la siguiente manera

```
PROGRAM Triangle
IMPLICIT NONE

 REAL :: a, b, c, Area

  PRINT *, ’Welcome, please enter the lengths of the 3 sides.’
  READ *, a, b, c
  PRINT *, 'Triangles area:', Area(a,b,c)

END PROGRAM Triangle

FUNCTION Area(x,y,z)
IMPLICIT NONE

 REAL :: Area   !Function type
 REAL, INTENT( IN ) :: x, y, z
 REAL :: theta, height

  theta = ACOS((x**2+y**2-z**2)/(2.0*x*y))
  height = x*SIN(theta); Area = 0.5*y*height

END FUNCTION Area
```

Ya teniendo el programa corregido se nos pide basarnos en el para realizar uno similar para calcular el volumen de un paralelepipedo. Realizandolo queda de esta forma

```
PROGRAM Paralelepipedo
IMPLICIT NONE

 REAL :: a, b, c, Volumen

  PRINT *, 'Favor de ingresar los tres lados distintos de su paralelepipedo par&
           &a realizar el calculo del volumen'
  READ *, a, b, c
  PRINT *, 'Volumen del paralelepipedo', Volumen(a,b,c)

END PROGRAM Paralelepipedo

FUNCTION Volumen(x,y,z)
IMPLICIT NONE

 REAL :: Volumen !Tipo Function
 REAL, INTENT( IN ) :: x, y, z

  Volumen= x*y*z

END FUNCTION Volumen
```

3 Segundo Problema

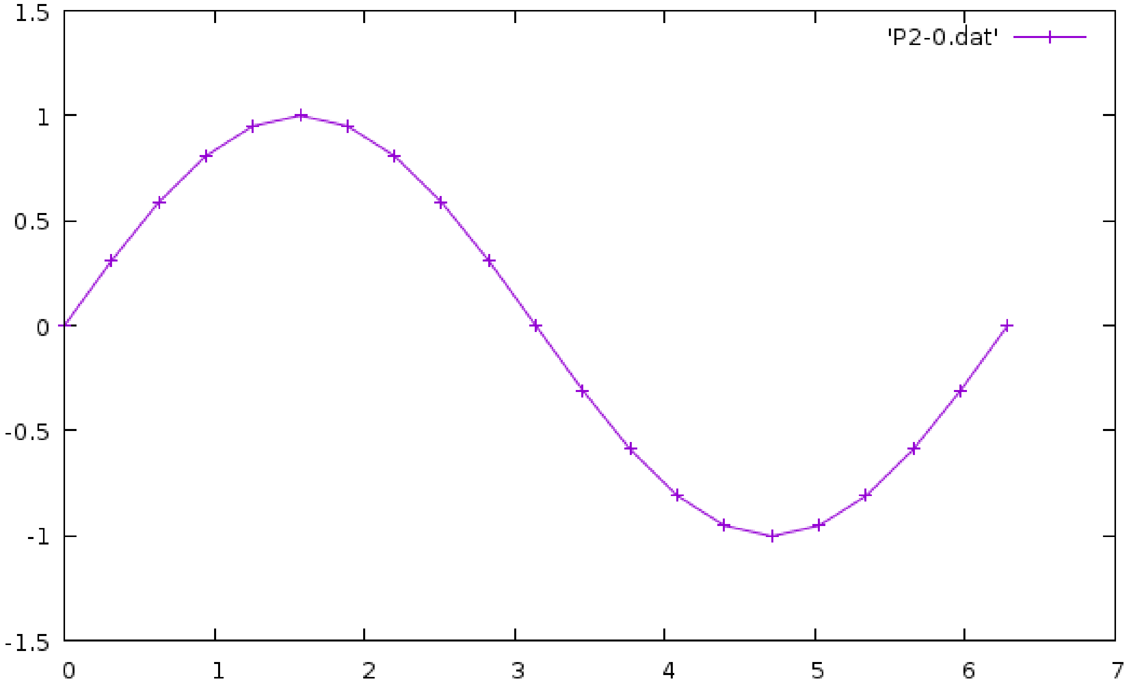
Para el segundo problema se nos facilita un codigo que resuelve el movimiento de un objeto sujeto a un resorte, obedeciendo la ley de Hooke. El codigo es el siguiente

```
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                                                                       !
! Please Note:                                                          !
!                                                                       !
! (1) This computer program is written by Tao Pang in conjunction with  !
!     his book, "An Introduction to Computational Physics," published   !
!     by Cambridge University Press in 1997.                            !
!                                                                       !
! (2) No warranties, express or implied, are made for this program.     !
!                                                                       !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
PROGRAM ONE_D_MOTION
!
! Program for the motion of a particle subject to an external
! force f(x) = -x.   We have divided the total time 2*pi into
! 10000 intervals with an equal time step.   The position and
! velocity of the particle are written out at every 500 steps.
! Copyright (c) Tao Pang 1997.
!
  IMPLICIT NONE
  INTEGER, PARAMETER :: N=10001,IN=500
  INTEGER :: I
  REAL :: PI,DT
  REAL, DIMENSION (N):: T,V,X
!
! Assign constants, initial position, and initial velocity
!
  PI   = 4.0*ATAN(1.0)
  DT   = 2.0*PI/FLOAT(N-1)
  X(1) = 0.0
  T(1) = 0.0
  V(1) = 1.0
!
! Recursion for position and velocity at later time
!
  DO I = 1, N-1
    T(I+1) = DT*I
    X(I+1) = X(I)+V(I)*DT
    V(I+1) = V(I)-X(I)*DT
  END DO
!
! Write the position and velocity every 500 steps
```

```
!
  WRITE (6,"(3F16.8)") (T(I),X(I),V(I),I=1,N,IN)
END PROGRAM ONE_D_MOTION
```

Para comenzar el trabajo con este codigo nos piden agregar modificaciones para escribir los datos en un documento .dat para posteriormente graficarlo, el codigo modificado y la grafica de los datos se muestran a continuacion

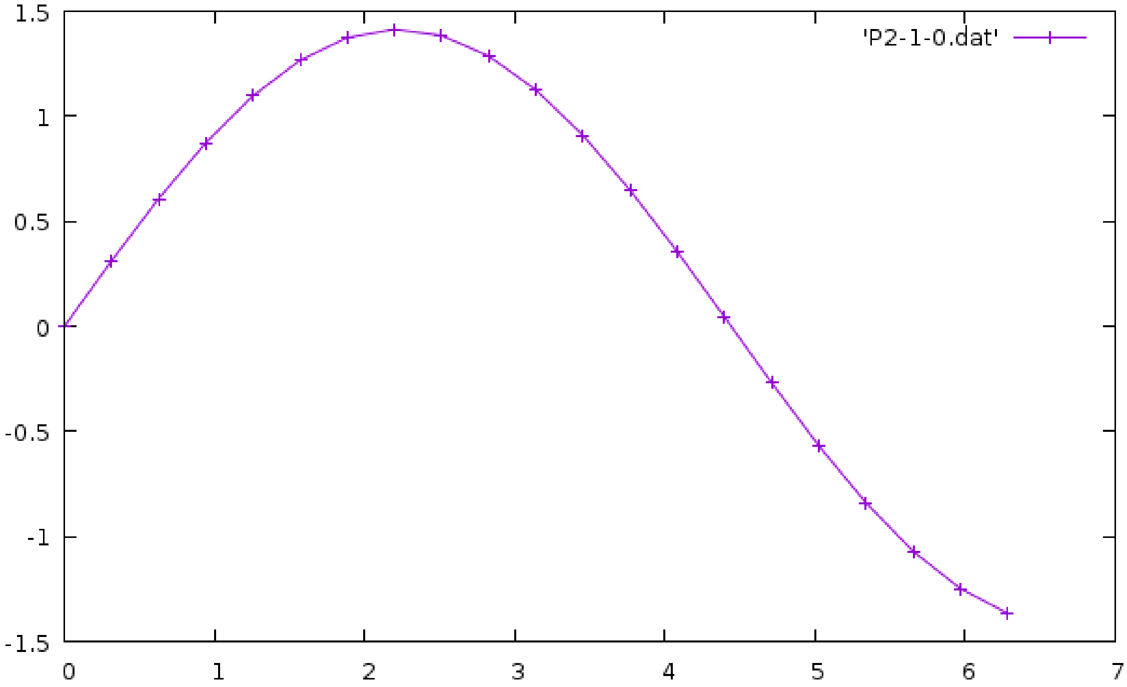
```
PROGRAM ONE_D_MOTION
IMPLICIT NONE

! Program for the motion of a particle subject to an external force f(x) = -x.
! We have divided the total time 2*pi into 10000 intervals with an equal time
! step. The position and velocity of the particle are written out at every 500
! steps.

 INTEGER, PARAMETER :: N=10001,IN=500
 INTEGER :: I
 REAL :: PI,DT
 REAL, DIMENSION (N):: T,V,X

! Assign constants, initial position, and initial velocity

  PI   = 4.0*ATAN(1.0)
  DT   = 2.0*PI/FLOAT(N-1)
  X(1) = 0.0
  T(1) = 0.0
  V(1) = 1.0

! Recursion for position and velocity at later time

   DO I = 1, N-1
     T(I+1) = DT*I
     X(I+1) = X(I)+V(I)*DT
     V(I+1) = V(I)-X(I)*DT
   END DO

! Write the position and velocity every 500 steps
    OPEN (6,FILE='P2-0.dat')
     WRITE (6,"(3F16.8)") (T(I),X(I),V(I),I=1,N,IN)
    CLOSE (6)
END PROGRAM ONE_D_MOTION
```



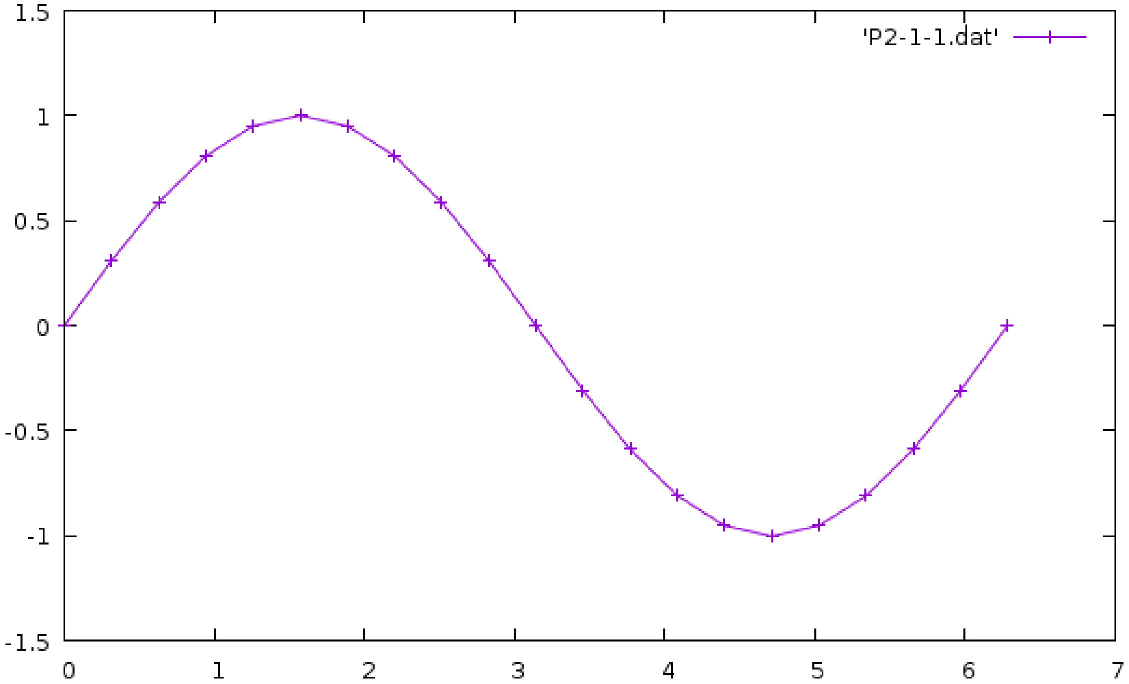
Siguiendo con el problema 2, ahora nos piden modificar el codigo para resortes con constante k y comparar distintas graficas para tres valores de k. A continuacion el codigo modificado y las tres graficas

```
PROGRAM ONE_D_MOTION
IMPLICIT NONE

! Program for the motion of a particle subject to an external force f(x) = -x.
! We have divided the total time 2*pi into 10000 intervals with an equal time
! step. The position and velocity of the particle are written out at every 500
! steps.

 INTEGER, PARAMETER :: N=10001,IN=500
 INTEGER :: I
 REAL :: PI,DT,k
 REAL, DIMENSION (N):: T,V,X

  WRITE(*,*) 'Ingresa el valor de k'
  READ*, k

! Assign constants, initial position, and initial velocity

  PI   = 4.0*ATAN(1.0)
  DT   = 2.0*PI/FLOAT(N-1)
  X(1) = 0.0
  T(1) = 0.0
  V(1) = 1.0

! Recursion for position and velocity at later time

   DO I = 1, N-1
     T(I+1) = DT*I
     X(I+1) = X(I)+V(I)*DT
     V(I+1) = V(I)-k*X(I)*DT
   END DO

! Write the position and velocity every 500 steps
    OPEN (6,FILE='P2-1.dat')
     WRITE (6,"(3F16.8)") (T(I),X(I),V(I),I=1,N,IN)
    CLOSE (6)
END PROGRAM ONE_D_MOTION
```

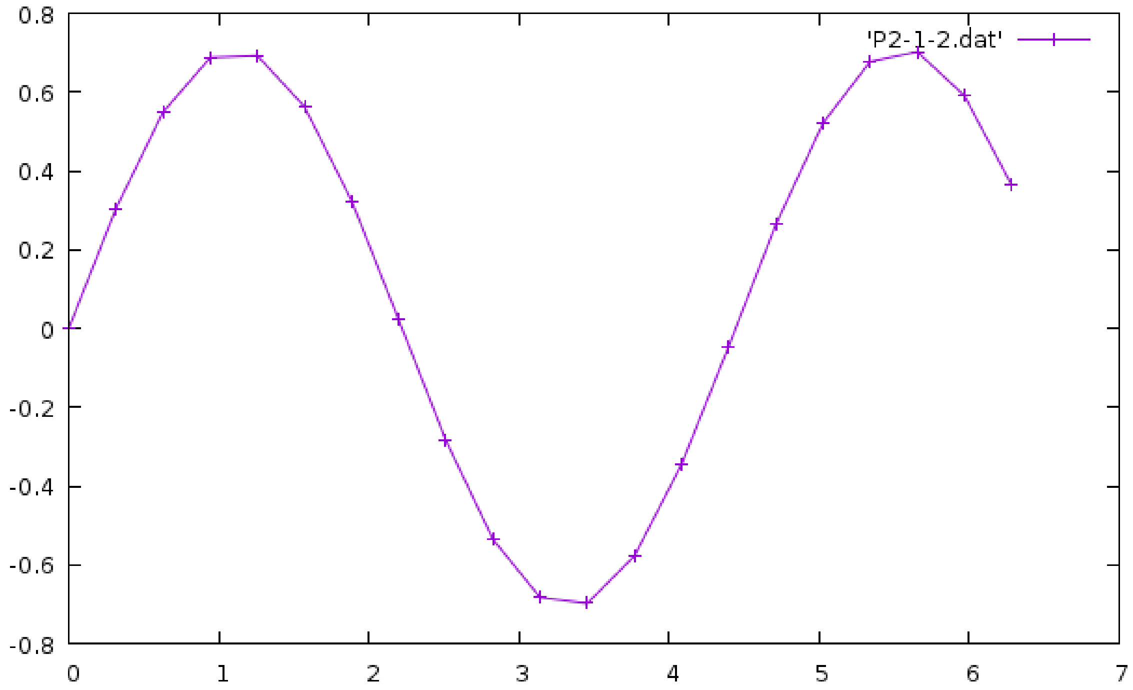
Para k = 0.5



Para k = 1.0



Para k = 2.0


'P2-1-2.dat'
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
1
2
3
4
5
6
7