Universidad de Sonora

División de Ciencias Exactas y Naturales

Física Computacional I

#2 Programas

Integrante:
Alejandro Guirado García

2 de febrero de 2016

1. Programas

En esta actividad modificaremos algunos códigos para obtener otras incognitas. Se mostrarán los códigos inciales y los modificados.

1.1. Código # 1

En el código inicial se puede obtener la altura de un objeto que está cayendo de una torre de altura "h"(sin tomar en cuenta fuerzas de fricción), en un determinado tiempo.

```
h = float(input("Proporciona la altura de la torre: "))
t = float(input("Ingresa el tiempo: "))
s = 0.5*9.81*t**2
print("La altura de la pelota es", h-s, "metros")
```

En el cógido modificado, se puede obtener el tiempo en que tardaría en llegar al suelo, soltado desde una altura "h",sin tomar en cuenta fuerzas de fricción.

```
h = float(input("Proporciona la altura de la torre: "))
t=sqrt(2*h/9.81)
print("El tiempo que tarda la pelota en llegar al suelo es", t ,"segundos")
```

1.2. Código # 2

Un satélite orbita la Tierra a una altura h, con un periodo T en segundos.

Demuestre que la altitud h del satélite sobre la superficie de la Tierra esta dado por la expresión

$$(R+h)^3 = (GMT^2)/(4pi^2) \tag{1}$$

donde G = 6.67 x 10-11 m^3 kg^{-1} s^{-2} es la constante de Gravitación Universal de Newton, M = 5.97 $\times 10^{24}$ kg es la masa de la Tierra y R=6371 km es su radio. ESe pide un programa que pida al usuario ingresar el valor deseado de T y regrese la altura h correspondiente en metros.

Demostración Si suponemos que los satélites recorren orbitas circulares.

$$\overrightarrow{F} = m\dot{a} = m\frac{\overrightarrow{v^2}}{\overrightarrow{r}} \tag{2}$$

fuerza responsable del movimiento orbital sería la fuerza gravitatoria por parte de la tierra , por lo tanto tomando en cuenta solamente los módulos, nos queda:

$$\frac{GmM}{r^2} = m\frac{v^2}{r} \tag{3}$$

Donde m es la masa de la tierra y M la masa del satélite.Se sabe que r.^{es} la distancia entre entre el centro de la tierra y el satélite.

Considerando el periodo del satélite:

$$v = \frac{2\pi r}{T} \tag{4}$$

Sustituyendo en y simplificando nos queda la siguiente expresión:

$$\frac{GM}{r^2} = \frac{4r\pi^2}{T} \tag{5}$$

Por último, despejando a r, como r es la distancia del centro de la tierra al satélite se puede expresar r como (r'+h) donde r' sería el radio de la tierra y h la distancia de la superficie al satélite.

$$(R+h)^3 = (GMT^2)/(4pi^2) \tag{6}$$

Lo cual es lo que se quería demostrar.

Código

```
from math import pi

T = float(input("Proporciona el período de órbita en segundos: "))
k = (6.67e-11*5.97e24)/(4*pi*pi)
h = (k*T*T)**(1./3.)-6371000
print("La altura del objeto con respecto a la superficie terrestre es",
h, "metros")
```

Después se nos pide, verificar la distancia del satélite a la tierra en base a su periodo, se mostrará en las siguientes 3 imagenes:

```
In [6]: from math import pi

        T = float(input("Proporciona el período de órbita en segundos: "))
        k = (6.67e-11*5.97e24)/(4*pi*pi)
        h = (k*T*T)**(1./3.)-6371000
        print("La altura del objeto con respecto a la superficie terrestre es",
        h, "metros")

        Proporciona el período de órbita en segundos: 86400
        ('La altura del objeto con respecto a la superficie terrestre es', 358559
        10.176174976, 'metros')
```

Figura 1: Con un periodo de 84600 segundos

```
In [6]: from math import pi

        T = float(input("Proporciona el período de órbita en segundos: "))
        k = (6.67e-11*5.97e24)/(4*pi*pi)
        h = (k*T*T)**(1./3.)-6371000
        print("La altura del objeto con respecto a la superficie terrestre es",
        h, "metros")

        Proporciona el período de órbita en segundos: 86400
        ('La altura del objeto con respecto a la superficie terrestre es', 358559
        10.176174976, 'metros')
```

Figura 2: Con un periodo de 5400 segundos

```
In [13]: from math import pi

         T = float(input("Proporciona el período de órbita en segundos: "))
         k = (6.67e-11*5.97e24)/(4*pi*pi)
         h = (k*T*T)**(1./3.)-6371000
         print("La altura del objeto con respecto a la superficie terrestre es",
         h, "metros")

         Proporciona el período de órbita en segundos: 3600
         ('La altura del objeto con respecto a la superficie terrestre es', -12958
         54.7452501757, 'metros')
```

Figura 3: Con un periodo de 3600 segundos

1.3. Código # 3

Coordenadas polares. Un punto en el espacio en el sistema de coordenadas polares se describe por las cantidades (r, θ).

La relación entre coordenadas polares y el sistema de coordenadas cartesianos, esta dada por las ecuaciones: x = r $\cos\theta$, y = $r \sin\theta$.

El siguiente programa para calcular las coordenadas cartesianas a partir de las coordenadas polares:

```
from math import sin,cos,pi
r = float(input("Introduce r: "))
d = float(input("Ingresa theta en grados: "))
theta = d*pi/180
x = r*cos(theta)
y = r*sin(theta)
print("x =",x," y =",y)
```

Se nos pide modificar el código para obtener coordenas esféricas a partír de rectangulares.

```
from math import pi,atan,acos
x = float(input("Introduce x: "))
y = float(input("Introduce y: "))
z = float(input("Introduce z: "))
r = sqrt(x*x+y*y+z*z)
theta = atan(r/z)
phi = acos(z/(sqrt((x*x)+(y*y)+(z*z))))
print("r =",r," theta =",theta," phi =",phi)
```

Al dar los siguientes valores Introduce: x=10, y=15, z=20 r= 26.92582403567252, theta= 0.9319311825594854, phi = 0.7335813236400831.

1.4. Código # 4

El siguiente código nos permite ingresar un número, el cual si es par nos arrojará .even" si es impar nos arrojará .odd".

```
n = int(input("Enter an integer: "))
if n%2==0:
     print("even")
else:
    print("odd")
```

El siguiente código nos permite ingresar dos números y al final nos dice que dos números pusimos.

```
print("Enter two integers, one even, one odd.")
m = int(input("Enter the first integer: "))
n = int(input("Enter the second integer: "))
while (m+n)%2==0:
    print("One must be even and the other odd.")
    m = int(input("Enter the first integer: "))
    n = int(input("Enter the second integer: "))
print("The numbers you chose are",m,"and",n)
```

1.5. Código # 5

Los Números de Fibonacci es una sucesión de números enteros aparecen en toda la naturaleza.

El siguiente programa calcula la secuencia de Fibonacci, introduce la condición de control while

```
f1,f2 = 1,1
while f2<1000:
     print(f2)
      f1,f2 = f2,f1+f2
      -----------------------------------
```

```
1
2
3
5
8
13
21
34
55
89
144
233
377
610
987
```

Se busca que basados en está ídea se escriba un programa para los números del catalán, que son dados por la fórmula de recurrencia:

$$C_0 = 1, C_{(n+1)} = \frac{2(2n+1)}{(n+2)} C_n \tag{7}$$

Imprimiendo cualquier numero menor o igual a 1000000.

```
n,c = 2,6

while c<100000:

       print(c)

             n,c = n+1,(((4*n)+2)/(n+2))*c
   -------------------------------
6
12
24
72
216
648
1944
5832
17496
52488
```