

# Proyecto 1

Universidad de Guanajuato, División de Ciencias e Ingenierías  
Programación Básica

Profesora: Dra. Alma Xochitl González Morales

Alumna: Guadalupe Sinaí Florián Landa

Septiembre 29 del 2018

## 1 Descripción del Programa

En este programa se calcularon las órbitas de 8 planetas y un satélite usando las ecuaciones de fuerza aplicado al movimiento de los planetas al rededor de su estrella, que en este caso es el sol.

Se utilizó el método de Euler, que consistió en establecer un conjunto de posiciones y velocidades a un tiempo inicial e ir actualizando el tiempo posterior para obtener las posiciones y velocidades nuevas en el tiempo  $i$ -ésimo.

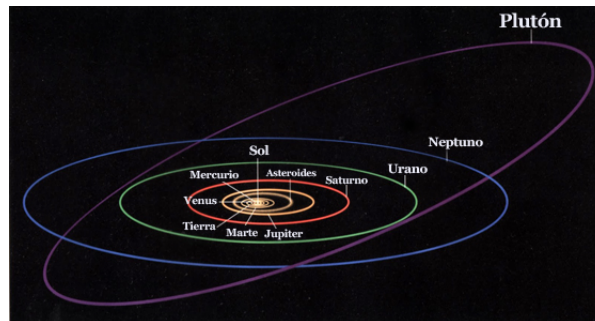


Figure 1: Órbitas de los 8 planetas del sistema solar y el satélite Plutón.

## 2 Comandos usados en el Programa

Primero se indicaron las librerías que se usarán en el programa, las cuales incluyen definiciones comunmente usadas en algoritmos. En este programa usamos la librería estandar:

*include < stdio.h >* (1)

y la librería que incluye varias funciones matemáticas. Una característica es que pueden ser definidas como `double` y regresar el resultado como `double`, que nos será de utilidad para los resultados de las coordenadas de los planetas que buscamos:

$$\textit{include} < \textit{math.h} > \quad (2)$$

finalmente se incluyen las librerías de `stdlib` y `string` para el uso de la función `snprintf` que explicaremos más adelante:

$$\textit{include} < \textit{stdlib.h} > \quad (3)$$

$$\textit{include} < \textit{string.h} > \quad (4)$$

Usamos la función maestra del programa, que sirve como punto de partida para la ejecución del programa. Un programa deja de ejecutarse normalmente al final del `main`, por lo tanto el corchete `}` se colocará hasta el final de nuestro programa:

$$\textit{intmain}()\{\} \quad (5)$$

Después declaramos las variables para indicar los archivos con los cuáles estaremos trabajando en el programa. Uno lo usaremos para leer las posiciones y velocidades iniciales del programa, y el otro lo usaremos para imprimir las coordenadas en un lapso de tiempo `T` dividido en fragmentos de valor `h`:

$$\textit{FILE} * \textit{archivo}; \quad (6)$$

$$\textit{FILE} * \textit{resultados}; \quad (7)$$

Después declaro mis variables de tipo `entras`, que serían la masa de la estrella `M` y el tiempo total en el cuál se calcularán las posiciones de los planetas:

$$\textit{intM}, T, a, k; \quad (8)$$

También declaramos las variables de punto flotante, el cuál ocupa hasta 8 bytes de información y tomará en cuenta hasta 10 números después del punto decimal. Esto nos será de mucha ayuda con los cálculos que realizaremos ya que la precisión será de gran importancia para que el resultados nos de la elipse buscada:

$$\textit{double} x0, y0, z0, vx0, vy0, vz0, X, Y, Z, VX, VY, VZ, r, G, m, h, i; \quad (9)$$

Finalmente declaro mi variable de tipo `caracter` que usaré en mi función `snprintf` para determinar el número máximo de caracteres que se usarán en el comando:

*char cadena* [32]; (10)

Abro el archivo dónde se encuentran los valores de las posiciones y velocidades iniciales del primer planeta para su lectura:

*archivo = fopen("coordenadas\_archivo.txt", "r");* (11)

Se inicia un ciclo for para ejecutar la instrucción `snprintf` un número de 9 veces:

*for (a = 0; a <= 8; a++) {}* (12)

Usamos el comando `snprintf` para formatear y guardar un número máximo de caracteres al buffer, en este caso serán 32 caracteres usando `sizeof(char)*32`. Se indica que el documento se llamará `Planeta%i.txt`, el `%i` pondrá un número `k` que aumentará de tamaño con la función `k++` (que colocaremos más adelante); la cuál incrementará una unidad por cada archivo de texto que se escriba por ciclo `a`:

*snprintf (cadena, sizeof (char) \* 32, "Planeta %i.txt" ,k);* (13)

Usamos `resultados=` para indicar que se escribirá el archivo `FILE *resultados` usando el nombre de la cadena de caracteres llamada `-cadena-` creada por `snprintf`:

*resultados = fopen (cadena,"wb");* (14)

El programa procede a leer el archivo de donde se leerán las posiciones y velocidades iniciales de nuestro planeta, así como su masa `m`, el tiempo total del ciclo alrededor de la estrella `T`, la masa de la estrella `M`, y la fracción del tiempo en la cuál se adquirirán las coordenadas `h`, todo en unidades astronómicas y años. Esta lectura de datos se repetirá en cada ciclo de tal forma que lea 3 renglones de información por planeta, hasta completar los 8 planetas y el satélite Plutón:

*fscanf (archivo," %lf %i %i %lf", &m, &M, &T, &h);*

*fscanf (archivo," %lf %lf %lf", &x0, &y0, &z0);*

*fscanf (archivo," %lf %lf %lf", &vx0, &vy0, &vz0);*

En seguida le indicamos al programa escribir el documento de texto donde se guardarán las coordenadas de cada planeta:

$$resultados = fopen("resultados_orbita.txt", "w"); \quad (15)$$

Procedemos a indicar el valor de la constante de gravitación universal calculada en unidades astronómicas. Indicamos que dicha constante es igual a 4 multiplicado \* por pi al cuadrado pow(pi,2):

$$G = 4.0 * pow(3.141592653589, 2); \quad (16)$$

Así mismo, convertimos las velocidades iniciales, dadas en unidades astronómicas y días, por \*365 para convertirlas en años:

$$vx0*=365.242;$$

$$vy0*=365.242;$$

$$vz0*=365.242;$$

Antes de iniciar el ciclo le indicamos al programa imprimir los caracteres de día, posición y velocidades para clasificar las coordenadas dadas en los resultados:

$$fprintf(resultados, "Día X Y Z VX VY VZn"); \quad (17)$$

En el siguiente paso, iniciamos un ciclo for desde el día cero, que tendrá las coordenadas de las posiciones y velocidades iniciales, completará un tiempo T en años, y tendrá un espaciado de h, es decir, el tiempo T se dividirá en fracciones de h para calcular las coordenadas por día.

$$for(i = 0.0; i <= T; i += h) \{ \} \quad (18)$$

Dentro del ciclo for incluiremos las siguientes operaciones que se realizarán por ciclo por planeta:

Para calcular las coordenadas de la posición siguiente (X,Y,Z) en la fracción de tiempo h, se sumará la posición inicial (x0,y0,z0) a la velocidad inicial (vx0,vy0,vz0) multiplicada por la fracción de tiempo h:

$$X=x0+vx0*h;$$

$$Y=y0+vy0*h;$$

$$Z=z0+vz0*h;$$

Para calcular el radio de la posición actual, que es la distancia entre la estrella (el sol) y el planeta, se sacará la raíz cuadrada de la suma de las posiciones iniciales  $x_0, y_0, z_0$  cada una elevadas al cuadrado:

$$r = \text{sqrt}(\text{pow}(x_0, 2.0) + \text{pow}(y_0, 2.0) + \text{pow}(z_0, 2.0)); \quad (19)$$

Para calcular las velocidades de la posición siguiente (VX,VY,VZ) se multiplicará la constante de gravitación G, la masa del sol M y la posición inicial suando \*, y luego se dividirá usando / entre el radio al cubo usando la potencia  $\text{pow}(r,3)$ ; todo se multiplicará por la fracción de tiempo h y se restará de la velocidad inicial ( $v_{x0}, v_{y0}, v_{z0}$ )

$$VX = v_{x0} - h * ((G * M * x_0) / \text{pow}(r, 3.0));$$

$$VY = v_{y0} - h * ((G * M * y_0) / \text{pow}(r, 3.0));$$

$$VZ = v_{z0} - h * ((G * M * z_0) / \text{pow}(r, 3.0));$$

Ya dados los resultados de las siguientes posiciones y velocidades, éstas tomarán el valor de posición y velocidad iniciales la siguiente vez que se repita el ciclo, para esto igualaremos los valores de  $(x_0, y_0, z_0)$  y  $(v_{x0}, v_{y0}, v_{z0})$  con  $(X, Y, Z)$  y  $(VX, VY, VZ)$ :

$$x_0 = X;$$

$$y_0 = Y;$$

$$z_0 = Z;$$

$$v_{x0} = VX;$$

$$v_{y0} = VY;$$

$$v_{z0} = VZ;$$

Ya hechos los calculos imprimimos los resultados:

Indicamos las coordenadas por día para ubicarnos en el lapso de tiempo, tomando el valor del salto  $i=h$  multiplicado por 365 usando \*:

```
fprintf (resultados, " %lf ", i*365);

fprintf (resultados, " %lf %lf %lf %lf %lf %lf ", X, Y, Z, VX, VY, VZ);
```

Usamos fprintf para imprimir todo lo escrito en el documento, luego usamos ( %lf,) para indicar la impresión del resultado del cálculo y (,X) para indicar la variable a la que pertenece dicho resultado. Y así sucesivamente para cada uno.

Procedemos a cerrar el archivo resultados dónde se imprime el primer set de coordenadas y velocidades:

```
fclose (resultados);
```

(20)

Cerramos el ciclo for i responsable de los cálculos de las coordenadas:

```
}
```

(21)

Se usa k++ para incrementar en una unidad el número a la derecha del nombre del documento \*resultados, cada vez que uno de ellos se escribe y finaliza:

```
k ++;
```

(22)

Cerramos el ciclo for a responsable del ciclo de la escritura de los archivos que contienen resultados

```
}
```

(23)

Cerramos el archivo donde se encontraba la información de los 8 planeta y el satélite Plutón:

```
fclose (archivo);
```

(24)

Usamos return 0 para ver si la secuencia de instrucciones se sucedió correctamente, de lo contrario enviará un signo de error para poder corregir algún comando o secuencia de instrucciones, y finalmente cerramos la función maestra del programa para finalizarlo:

```
return 0;

}
```

