



---

## **PROGRAMACIÓN BÁSICA PROYECTO 1.-** **CALCULAR LAS ORBITAS DE LOS PLANETAS DEL SISTEMAS** **SOLAR.**

---

ESTUDIANTE DE LIC. FÍSICA LUIS ALBERTO PÉREZ MARTÍNEZ  
01 DE OCTUBRE DE 2018.

PROGRAMA PARA CALCULAR LA ORBITA DE UN PLANETA EN TORNO A UNA ESTRELLA.

Se creo un archivo de texto llamado "parametros.txt" en el cual se definió en la primer línea las variables "M" correspondiente a la masa de la estrella en unidades de masa solar y la "G" correspondiente al valor de la gravedad en unidades ... en la segunda línea se definió el valor de "a" tiempo de evolución en unidades de años y "h" tiempo de incremento de tiempo. En la tercer línea se definió las coordenadas "x0", "y0", "z0" y las velocidades "Vx0", "Vy0", "Vz0", en unidades astronómicas en días. Se repitieron las últimas dos filas para los demás planetas, con los valores correspondientes a cada uno.

En el programa se incluyeron las siguientes librerías: "#include<stdio.h>" y "#include<math.h>", la librería math.h se utilizo para las raíces cuadradas y los exponentes en las ecuaciones que se utilizaron.

```
1. #include<stdio.h>
2. #include<math.h>
3.
4. int main( ){
5. }
```

Ya dentro del `int main( )` se declararon las variables enteras: "M" ( masa de la estrella), "i" (variable de control) , "j" (variable de control), " o [ 9 ]" (tiempo de repeticiones) ; también se declararon las siguientes variables flotantes: "h [ 9 ]", "G" y las variables dobles: "x [ 9 ]", "y [ 9 ]", "z [ 9 ]", "vx [ 9 ]", "vy [ 9 ]", "vz [ 9 ]", "x0 [ 9 ]", "y0 [ 9 ]", "z0 [ 9 ]", "vx0 [ 9 ]", "vy0 [ 9 ]", "vz0 [ 9 ]" (posiciones, velocidades iniciales y finales).

```
1. #include<stdio.h>
2. #include<math.h>
3.
4. int main( ){
5.
6.     int M, i, j, o[9];
7.     float h[9], G;
8.     double x[9], y[9], z [9], vx[9], vy[9], vz[9], vx0[9], vy0[9], vz0[9],
9.           x0[9], y0[9], z0[9], vx0[9], vy0[9], vz0[9];
10. }
```

A continuación se abrió el archivo de "parametros.txt" creado previamente, y se leyeron las variables de la M , G y los vectores de los tiempos de evolución, incremento, las posiciones y velocidades; para los vectores se utilizó un ciclo de 9 repeticiones para poder leer los datos correspondientes a los 9 planetas.

```
10. FILE *data;
11. data=fopen("parametros.txt", "r");
12. fscanf(data, "%d %f",&M,&G);
13. for(i=0;i<9;i++){
14.
15.     fscanf(data,"%lf %lf",&a[i],&h[i];
```



```
16.          fscanf(data,"%lf          %lf          %lf          %lf          %lf
%lf",&x0[i],&y0[i],&z0[i],&vx0[i] ,&vy0[i],&vz0[i];
17.
18.      }
19. }
```

Para poder guardar el valor de repetición de cada planeta guardamos el valor de la división de el tiempo de evolución entre el incremento y convertimos las velocidades iniciales de años a días multiplicándolas por 365 y se cierra el documentos paramentos.txt al final del ciclo.

```
...
17.
18.          o[i]=a[i]/h[i];
19.
20.          vx0[i]= vx0[i]*365;
21.          vy0[i]= vy0[i]*365;
22.          vz0[i]= vz0[i]*365;
23.
24.      }
25. fclose(data);
26. }
```

Para escribir los datos en cada de las posiciones encada segundo correspondientes a cada planeta primero se realizo solo para uno, Mercurio y tomando como base el código se copio par los 8 planetas faltan tente.

Primero se abrió el archivo en el que se van a escribir las posiciones y vellosidades:

```
...
26.
27.      FILE *mer;
28.      mer=fopen("Mercurio.txt", "w");
29.      fprintf(mer, "%lf\t %lf\t %lf\t %lf\t %lf\t %lf\n", x0[0],
y0[0], z0[0], vx0[0], vy0[0], vz0[0] );
31.      fclose(mer);
32. }
```

Note se que se tomaron posiciones y vellosidades correspondientes a la posición 0 de el vector, esto va a variar dependiendo de el planeta que se trate.

Para poder escribir los datos correspondientes se realizaron las operaciones para los datos finales y se igualaron los datos finales a los iniciales dentro de un ciclo para poder repetir el proceso sin necesidad de escribirlo varias veces, para ello se utilizó la variable de control "i" y se puso la condición de que fuera menor a el tiempo de evolución entre el incremento para que se calculara la orbita de un año aproximadamente, y que tuviera un paso de uno en uno.

```
...
30.      fprintf(mer, "%lf\t %lf\t %lf\t %lf\t %lf\t %lf\n", x0[0],
y0[0], z0[0], vx0[0], vy0[0], vz0[0] );
31.      for(i =0; i<o[0]; i++){
28.
30.          r= sqrt(pow(x0[0],2) + pow(y0[0],2) + pow(z0[0],2));
31.
32.          x = x0 + vx0*h;
33.          y = y0 + vy0*h;
34.          z = z0 + vz0*h;
35.          vx = vx0 - (h[0]*G*M*x0[0]/(pow(r, 3)));
36.          vy = vy0 - (h[0]*G*M*y0[0]/(pow(r, 3)));
37.          vz = vz0 - (h[0]*G*M*z0[0]/(pow(r, 3)));
38.
39.
40.          x0[0] = x[0];
41.          y0[0] = y[0];
42.          z0[0] = z[0];
43.          vx0[0] = vx[0];
```



```

44.         vy0[0] = vy[0];
45.         vz0[0] = vz[0];
46.
47.     }
48.     fclose(mer);
49. }

```

Para imprimir las posiciones finales se utilizó una condición, para que no se imprimieran todos los puntos, sino cada 500. Ya que si se registrar todos los puntos, el archivo de resultados iba a superar el peso para poderlo subir a la plataforma de "github". Para ello se utilizó la variable de control "c":

```

...
31.         c=0;
31.         for(i =0; i<o[0]; i++){
28.
30.             r= sqrt(pow(x0[0],2) + pow(y0[0],2) + pow(z0[0],2));
...
45.         vz0[0] = vz[0];
46.
47.         if (c%500 == 0){
48.             fprintf(mer, "%f\t %f\t %f\t %f\t %f\t %f\n", x[0], y[0],
z[0], vx[0], vy[0], vz[0] );
49.         }
50.
51.         c++;
52.     }
53.     fclose(mer);
54. }

```

Terminado el código de un planeta lo único que se hizo fue cambiar las variables de los vectores a su posición correspondientes, por ejemplo en Mercurio se utilizó la posición "0" para Venus se van a utilizar la posición "1", y así sucesivamente hasta llegar a la posición 8 que corresponde a Plutón.

También se cambió el nombre de el archivo de escritura para cada planeta "nombre de el planeta correspondiente.txt", así como la variable para abrirlo que corresponde a las primeras tres letras del planeta: como se puede ver en la siguiente imagen:

```

32.         for(j=0; j<o[1]; j++){
33.
34.             r=sqrt(pow(x0[0],2)+pow(y0[0],2)+pow(z0[0],2));
35.
36.             x[0]=x0[0]+(vx0[0]*h[0]); //coordenada de x en un lapso de tiempo despues de h segundos
37.             y[0]=y0[0]+(vy0[0]*h[0]); //coordenada de y en un lapso de tiempo despues de h segundos
38.             z[0]=z0[0]+(vz0[0]*h[0]); //coordenada de z en un lapso de tiempo despues de h segundos
39.             vx[0]=vx0[0]-((h[0]*G*x0[0])/pow(r,3));
40.             vy[0]=vy0[0]-((h[0]*G*y0[0])/pow(r,3));
41.             vz[0]=vz0[0]-((h[0]*G*z0[0])/pow(r,3));
42.
43.             y0[0]=y[0];
44.             vy0[0]=vy[0];
45.             z0[0]=z[0];
46.             vz0[0]=vz[0];
47.             x0[0]=x[0];
48.             vx0[0]=vx[0];
49.
50.             if (c%500==0)
51.             {
52.                 fprintf(mer, "%f\t %f\t %f\t %f\t %f\t %f\n", x[0], y[0], z[0], vx[0], vy[0], vz[0]);
53.             }
54.             c++;
55.         }
56.         fclose(mer);
57.
58.         FILE *ven;
59.         ven=fopen("Venus.txt", "w");
60.         c=0;
61.         fprintf(ven, "%f\t %f\t %f\t %f\t %f\t %f\n", x[1], y[1], z[1], vx[1], vy[1], vz[1]);
62.         for(j=0; j<o[1]; j++){
63.
64.             r=sqrt(pow(x0[1],2)+pow(y0[1],2)+pow(z0[1],2));
65.
66.             x[1]=x0[1]+(vx0[1]*h[1]); //coordenada de x en un lapso de tiempo despues de h segundos
67.

```

Figura 1.1 Proyecto1.c



Como se puede observar en las partes subrayadas con rojo se cambió el código de cada planeta de la forma anteriormente mencionada.

#### GRÁFICAS DE LA ROBITAS DE LOS PLANETAS.

Para gráficas los planetas se utilizo el comando “gnuplot”, y se grafico en 3 dimensiones utilizando las columnas de 1:2:3 que correspondían a las posiciones de la orbita a cada paso de tiempo.

Se realizo el paso anterior para cada archivo creado y se guardaron las graficas en formato “.png” con su nombre correspondiente.

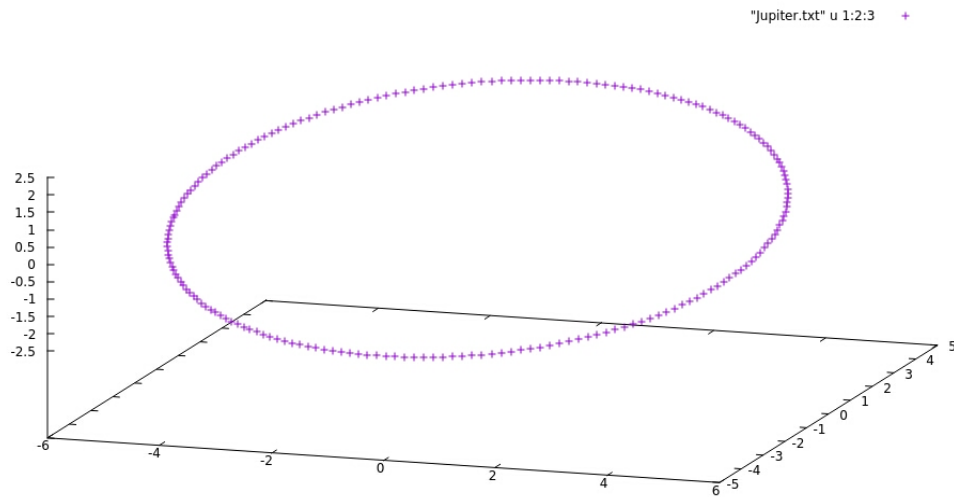


Imagen 1. orbita de Júpiter.