

Mecánica Celeste

Práctico 4: Problema de n-cuerpos: aplicaciones del problema circular de 3 cuerpos restringido (PC3CR)

Curso 2024

14 de mayo – Entrega: 4 de junio

1. Un sistema binario esta compuesto por una estrella B de masa $0.08 M_{\odot}$ que orbita una estrella A de masa $1.5 M_{\odot}$ con un semieje mayor de 1000 UA . Detertmine si es posible la existencia de un exoplaneta en una órbita estable en alguno de los puntos Lagrangianos L_4 o L_5 . Justifique su respuesta.
2. Considere el PC3CR Sol-Júpiter con masas $1 M_{\odot}$ y $1/1000 M_{\odot}$. Separadas 5.2 UA . Encuentre la amplitud y el período de oscilación en el eje z (i.e., $z(t)$):
 - a) Una partícula sin masa en L_1 que es apartada infinitesimalmente del plano de movimiento.
 - b) Una partícula sin masa desplazada infinitesimalmente de L_5 : $\mathbf{r} = (2.607, -4.496, 0.007) \text{ UA}$, y $\mathbf{v} = (0.006533, 0.003772, 0.0) \text{ UA/día}$.

$$\begin{aligned}\vec{\mathbf{r}} &= (2.607, -4.496, 0.007) \text{ UA} \\ \vec{\mathbf{v}} &= (0.006533, 0.003772, 0.0) \text{ UA/día}\end{aligned}$$

- Luego, utilice el integrador IBS (ver instrucciones) y compare con el resultado teórico obtenido para la oscilación del objeto cercano.
 - El objeto debería moverse alrededor de L_5 siguiendo una trayectoria de libración muy particular. Grafique y describa esa trayectoria.
 - Grafique la evolución de la partícula en el plano a vs. σ ; con $\sigma = \lambda - \lambda_2$.
3. Utilizando el integrador IBS grafique la evolución alrededor de L_4 de un objeto que se encuentra en: $\vec{\mathbf{r}} = (2.60, 4.503, 0.00) \text{ UA}$, y $\vec{\mathbf{v}} = (-0.006533, 0.003772, 0.0) \text{ UA/día}$.
 4. Un cometa con elementos orbitales: $a = 3.662150 \text{ UA}$, $e = 0.826771$, $I = 21.92781^\circ$ sufre un encuentro con Júpiter. Realice un programa que calcule la constante de Tisserand para el cometa. Luego, con el mismo programa determine si alguno de estos cometas observados algunos años después puede ser el cometa inicial:

Cometa	a (UA)	e	I ($^\circ$)
a	3.256178	0.725466	17.33265
b	4.162150	0.926651	51.12684
c	4.578197	0.626811	31.82703
d	5.013227	0.879914	23.83391
e	4.987654	0.850920	29.09408

Instrucciones para utilizar el integrador numérico

EL INTEGRADOR DE BULIRSH-STOER:

Dado que el problema de n-cuerpos no tiene solución analítica para estudiar la dinámica de sistemas de tres o más cuerpos necesitamos utilizar métodos numéricos que nos permitan integrar las ecuaciones de movimiento. Si bien existen numerosos procedimientos para realizar esta tarea el método de Bulirsh -

Stoer es uno de los más efectivos para integrar funciones suaves que no tengan saltos o puntos singulares y es sencillo de programar.

La idea central del método es avanzar el sistema de partículas para el cual conocemos posición y velocidad en un instante t_0 a otro instante $t_1 = t_0 + H$, donde H es un intervalo de tiempo no infinitesimal, utilizando una simple extrapolación lineal:

$$\vec{r}(t)_1 \simeq \vec{r}(t_0) + H \times \vec{v}(t_0),$$

$$\vec{v}(t)_1 \simeq \vec{v}(t_0) + H \times \vec{a}(t_0),$$

para cada partícula del sistema. Es claro que las posiciones y velocidades conseguidas de este modo son aproximadas y no de muy buena calidad, así que para mejorarlas se realiza una segunda aproximación cubriendo el intervalo de tiempo H en dos pasos $H/2$:

$$\vec{r}(t_0 + \frac{H}{2}) \simeq \vec{r}(t_0) + \frac{H}{2} \times \vec{v}(t_0),$$

$$\vec{v}(t_0 + \frac{H}{2}) \simeq \vec{v}(t_0) + \frac{H}{2} \times \vec{a}(t_0),$$

$$\vec{r}(t_0 + H)_2 \simeq \vec{r}(t_0 + \frac{H}{2}) + \frac{H}{2} \times \vec{v}(t_0 + \frac{H}{2}),$$

$$\vec{v}(t_0 + H)_2 \simeq \vec{v}(t_0 + \frac{H}{2}) + \frac{H}{2} \times \vec{a}(t_0 + \frac{H}{2}).$$

La nueva estimación será algo mejor que la primera, pero posiblemente aún no sea suficientemente buena así que posiblemente sea necesario obtener aproximaciones sucesivas cubriendo el intervalo de tiempo H con intervalos que correspondan a la mitad de los utilizados en el paso anterior ($H/4$, $H/8$, $H/16$, etc.). Este proceso de aproximación al valor correcto se denomina MÉTODO DE EXTRAPOLACIÓN DE RICHARDSON y el método de Bulirsh-Stoer hace uso intensivo de él. Para decidir finalmente si la aproximación es lo suficientemente buena se calcula la diferencia entre dos aproximaciones sucesivas y si este valor es menor que una cierta tolerancia fijada de antemano el proceso termina y se pasa a estimar (\vec{r}, \vec{v}) para el instante $(t_0 + 2H)$ tomando ahora como instante inicial el $(t_0 + H)$.

EL PROGRAMA IBS:

El programa **ibs** realiza la integración de un sistema de hasta 50 partículas con o sin masa que se mueven en el pozo de potencial de un objeto de masa M_\odot utilizando el método explicado más arriba. Si bien el programa está muy comentado para que quede claro que es lo que hace vamos a destacar algunos puntos importantes para facilitar su uso.

Para comenzar, en la línea 10 del programa se indica cuál debe ser el contenido del archivo de entrada:

```
! Los datos de entrada se leen del archivo IBS.IN en el
! siguiente orden:
!
! cg y vtol      cte. Gauss y error del integrador
! T0 y Tf       instante inicial y final en DJ
! step y isal   paso de integracion y nro. de pasos para impresion
! ncuer        nro. de cuerpos en total
! nombre y rm   nombre (a12) y masa (frac. masa central)
! x,y,z        coord. eclipticas en UA
! vx,vy,vz     vel. eclipticas en UA/dia
! ...
!
```

Entonces, si queremos integrar un sistema de cuatro cuerpos durante un lapso de 240 años, con un paso de 1 día, donde el primero tiene una masa M_\odot y se encuentra en el centro de coordenadas; el segundo tienen una masa $M_\odot/1000$ y se mueve en una órbita circular de semieje mayor $5.2UA$; el tercero es una partícula de masa infinitesimal en el punto L_4 y la cuarta una partícula de masa infinitesimal se encuentra cercana al punto L_5 , el archivo de entrada será:

```

0.01720209895 1.d-12 ! ctd. Gauss y tolerancia
0. 87660.           ! tiempo inicial y final DJ
1. 10             ! paso temporal y Nro. de pasos para sacar info
3 ! nro. de objetos (ademas de la masa central)
conmasa           0.10000000000000E-02
  0.52000000000000E+01  0.00000000000000E+00  0.00000000000000E+00
  0.00000000000000E+00  0.75473902837323E-02  0.00000000000000E+00
sinmasa1          0.00000000000000E+00
  0.26060000000000E+01  0.45083320996791E+01  0.00600000000000E+00
 -0.65329660511751E-02  0.37718097082526E-02  0.00000000000000E+00
sinmasa2          0.00000000000000E+00
  0.26000000000000E+01 -0.45033320996790E+01  0.00000000000000E+00
  0.65329660511751E-02  0.37718097082527E-02  0.00000000000000E+00

```

donde el número de pasos para sacar información es el intervalo en el cual el integrador guarda la información en un archivo de salida (en este caso 10 pasos, o sea cada 10 días). La primer partícula sin masa esta separada de L_4 por un vector con componentes (0.006,0.006,0.006).

Siguiendo con el programa, en la línea 22 se indica cómo se guardan las componentes de posición y velocidad para cada cuerpo en la matriz x:

```

! El program almacena las componentes de posicion y velocidad
! en la matriz X. Si las componentes de posicion son (x,y,z)
! y las de velocidad (vx,vy,vz), el orden de almacenamiento es:
!
! X(1) x_1
! X(2) vx_1
! X(3) y_1
! X(4) vy_1
! X(5) z_1
! X(6) vz_1
! X(7) x_2
! ...

```

que es una información importante cuando se desea extraer la posición y velocidad de uno cualquiera de los cuerpos.

El programa comienza efectivamente en la línea 41, en la línea 64 lee el archivo de entrada y en la 79 decide si integrará hacia adelante o hacia atrás en el tiempo. En la línea 113 se abre un archivo por cuerpo para guardar los resultados de la integración. Como tenemos 3 cuerpos (uno con masa y dos sin masa) debemos abrir tres archivos, entonces modificamos el programa con:

```

...
! aqui se va a guardar la informacion para cada objeto
! en un archivo diferente
!*****
!
  open(7,file='ibs-1.out')
  open(8,file='ibs-2.out')
  open(9,file='ibs-3.out')

!*****
! hasta aqui. Recuerde cerrar los archivos antes de
! salir!
!*****
...

```

En la línea 123 hay un if que decide si el programa hizo los pasos necesarios para guardar información en los archivos de salida. Esta es la parte a modificar según la información que uno quiere sacar. Por ejemplo, si cada vez que se escribe en los archivos se quiere mostrar en pantalla el tiempo y guardar para cada objeto el tiempo, nombre, posición y velocidad, el programa debe ser modificado de la siguiente manera:

```

...
! si corresponde, guarda informacion en el
! archivo de salida
!
  if((jit == isal).or.(jit == -1))then

!*****
! aqui es donde se obtiene la informacion de salida
! asi que es la parte a modificar si es necesario
!*****
!

! saco tiempo a pantalla
!
  write(*,*)t0

! guardo (x,y,z) y (vx,vy,vz) para los tres
! primeros objetos
!
  write(7,*)t0,nom(1),x(1),x(3),x(5),x(2),x(4),x(6)
  write(8,*)t0,nom(2),x(7),x(9),x(11),x(8),x(10),x(12)
  write(9,*)t0,nom(3),x(13),x(15),x(17),x(14),x(16),x(18)

!*****
! de aqui en adelante no modificar
!*****
...

```

Finalmente, en la línea 179 hay que cerrar todos los archivos de salida:

```

...
!*****
! se cierran los archivos de salida
!*****
!
  close(7)
  close(8)
  close(9)
...

```

Si hacemos correr el integrador obtendremos como salida tres archivos con el día, nombre, componentes del vector posición y componentes del vector velocidad para cada objeto. Por ejemplo, en el caso del cuerpo con masa (archivo `ibs-1.out`):

```

0. conmasa      5.2 0. 0. ...
10. conmasa     5.199452287504945 0.07547125327427517 0. ...
20. conmasa     5.197809265400158 0.15092660787607512 0. ...
30. conmasa     5.195071279802477 0.22635016848211817 0. ...

```

```

40. conmasa      5.191238907492297 0.3017260464668039 0. ...
50. conmasa      5.186312955792051 0.3770383632492906 0. ...
...

```

Si tratamos de graficar la posición de cualquiera de estos objetos encontramos que describen siempre un círculo coincidente en el plano X-Y (graficar la columna 3 contra la 4) lo que es razonable porque están en la misma órbita pero con anomalías medias diferentes y porque lo que tenemos es un *problema restringido de tres cuerpos*. El único caso destacado es que el objeto denominado "sinmasa2" se encuentra más alejado de L_5 por lo cual debería describir un cierto movimiento alrededor de ese punto lagrangiano (**ver ejercicio 1 de la parte de aplicación numérica**) tal como vimos en la teoría.

Para visualizar mejor el movimiento de los objetos sin masa debemos dejar fijo en el espacio al cuerpo con masa. Para ello podemos trabajar con los datos de los archivos de salida o calcularlo directamente durante la integración. En este último caso, definimos primero un par de vectores auxiliares y un par de variables auxiliares al principio del programa:

```

...
real(d) :: tau,dis
real(d),dimension(3)::vec2,vec3
...

```

y modificamos el programa a partir de la línea 139 asumiendo que el cuerpo con masa está en la posición (5.2,0.0,0.0):

```

...
! guardo (x,y,z) para el objeto con masa
!
      write(7,*)t0,nom(1),x(1),x(3),x(5)

! usando la posición del cuerpo con masa calculo la posición
! relativa de la partícula 2 para un sistema de coordenadas en
! rotación
!
      dis=sqrt(x(7)**2+x(9)**2)
      tau=atan2(x(9),x(7))-atan2(x(3),x(1))
      if(tau < 0._d)tau=tau+2._d*pi
      vec2(1)=dis*cos(tau)
      vec2(2)=dis*sin(tau)
      vec2(3)=x(11)
      write(8,*)t0,nom(2),vec2

! usando la posición del cuerpo con masa calculo la posición
! relativa de la partícula 3 para un sistema de coordenadas en
! rotación
!
      dis=sqrt(x(13)**2+x(15)**2)
      tau=atan2(x(15),x(13))-atan2(x(3),x(1))
      if(tau < 0._d)tau=tau+2._d*pi
      vec3(1)=dis*cos(tau)
      vec3(2)=dis*sin(tau)
      vec3(3)=x(17)
      write(9,*)t0,nom(3),vec3
...

```

Si hacemos correr el integrador obtendremos como salida un archivo con el día, nombre, y los componentes del vector posición para el objeto con masa, y dos archivos con el día, nombre, y los componentes

del vector posición relativos al objeto con masa para los restantes. Por ejemplo, en el caso del cuerpo con masa (archivo `ibs-1.out`):

```
0. conmasa      5.2 0. 0.
10. conmasa     5.199452287504945 0.07547125327427517 0.
20. conmasa     5.197809265400158 0.15092660787607512 0.
30. conmasa     5.195071279802477 0.22635016848211817 0.
40. conmasa     5.191238907492297 0.3017260464668039 0.
50. conmasa     5.186312955792051 0.3770383632492906 0.
60. conmasa     5.180294462396142 0.4522712536384562 0.
70. conmasa     5.173184695152346 0.5274088691750382 0.
...
```

Si ahora graficamos la posición de cualquiera de los objetos sin masa encontramos que describen un movimiento *respecto del objeto con masa* que sigue una trayectoria alrededor de los puntos lagrangianos triangulares, incluso el objeto que se ubicaba *exactamente* en L_4 (**ver ejercicio 3 más abajo**).

Si en lugar de coordenadas rectangulares se desea obtener elementos orbitales antes de guardar los resultados en los archivos de salida se deberán obtener los elementos osculadores a partir de la posición y la velocidad del objeto. Por ejemplo, se puede utilizar la subrutina `VECTOE` que está disponible en la página del curso en el archivo `CURSO-MEC-3.F90`.