



## Reto. Órbitas especiales de asteroides

Jueves 15 de junio de 2023

---

### Introducción

El objetivo principal de esta situación problema ha sido desarrollar un sistema que nos permita simular la interacción entre tres objetos bajo la influencia de las fuerzas gravitacionales que actúan entre ellos. Sin embargo, este sistema se ve limitado por varios factores, como el campo gravitacional generado por cada objeto en función de su masa y las condiciones iniciales, como la velocidad y la posición relativa de cada objeto con respecto a los demás.

La importancia de contar con un sistema de este tipo radica en la posibilidad de realizar simulaciones precisas del movimiento de objetos en el espacio real. El programa desarrollado puede adaptarse para considerar cualquier condición y parámetro de un cuerpo en el espacio, lo que nos brinda un alto nivel de confianza en los resultados obtenidos. Esto se debe a que al diseñar el programa, se han tenido en cuenta conceptos fundamentales que describen el movimiento de los astros, como las leyes de Kepler y la constante gravitacional.

En esta entrega, hemos combinado los conceptos implementados en los documentos anteriores con los conocimientos adquiridos en clase para resolver el problema inicial planteado en la situación problemática del curso.

### Teoría empleada

La ley de la gravitación universal propuesta por Isaac Newton describe de manera formal la fuerza de atracción entre dos objetos debido a su masa. Según esta ley, dos cuerpos se atraen con una fuerza que es directamente proporcional al producto de sus masas e inversamente proporcional al cuadrado de la distancia que los separa. Esta relación se puede expresar mediante la siguiente ecuación:

$$\mathbf{F}_g = -G \frac{M \cdot m}{r^2} \hat{\mathbf{r}}$$

Donde  $\mathbf{F}_g$  representa la fuerza gravitatoria,  $G$  es la constante gravitacional que tiene un valor de aproximadamente  $6.67 \cdot 10^{-11} \frac{N \cdot m^2}{kg^2}$ ,  $M$  y  $m$  son las masas de los objetos en cuestión, y  $r$  es la distancia entre ellos.

Además de la ley de la gravitación universal, es importante mencionar el problema de los  $N$  cuerpos, que surge al intentar calcular las interacciones gravitatorias entre más de dos objetos. En este problema, cada objeto interactúa gravitatoriamente con todos los demás objetos presentes, lo que resulta en un sistema dinámico complejo. La solución analítica para este problema no es posible en la mayoría de los casos, por lo que se recurre a métodos numéricos y simulaciones por computadora para estudiar y predecir el comportamiento de sistemas con múltiples cuerpos interactuantes.

Un método numérico comúnmente utilizado para abordar el problema de los  $N$  cuerpos es el método de Leapfrog, también conocido como método de Verlet. Este algoritmo se utiliza para simular el movimiento de partículas en un sistema dinámico y emplea una aproximación de segundo orden para integrar las ecuaciones de movimiento. A diferencia del método de Euler, el método de Leapfrog realiza modificaciones en la aproximación de la posición. En el método de Euler, la posición se aproxima mediante la siguiente fórmula:

$$x_1 = x_0 + hv_0$$

En el caso del método de Leapfrog, se toma el valor intermedio del intervalo:

$$x_1 = x_0 + hv_{\frac{1}{2}}$$

De manera general, un paso para aproximar la posición y la velocidad en un tiempo  $t_n$  a partir de tiempos anteriores se puede escribir como:

$$v_{n+\frac{1}{2}} = v_n + \frac{1}{2}hF(x_n)$$

$$x_{n+1} = x_n + hv_{n+\frac{1}{2}}$$

$$v_{n+1} = v_{n+\frac{1}{2}} + \frac{1}{2}hF(x_{n+1})$$

Estos pasos permiten avanzar en el tiempo y obtener una aproximación del movimiento de las partículas en el sistema dinámico. El método de Leapfrog es ampliamente utilizado en la simulación de sistemas gravitatorios y ofrece una buena precisión y estabilidad en comparación con otros métodos numéricos.

### Planteamiento del problema

Para abordar el problema de los tres cuerpos, se procede a definir un sistema de coordenadas tridimensional que consta de tres masas puntuales ubicadas en posiciones arbitrarias:  $P_1(m_1)$ ,  $P_2(m_2)$  y  $P_3(m_3)$ . Los vectores de posición respectivos de estas masas se expresan de la siguiente manera:

$$\mathbf{r}_1 = x_1\hat{\mathbf{x}} + y_1\hat{\mathbf{y}} + z_1\hat{\mathbf{z}}$$

$$\mathbf{r}_2 = x_2\hat{\mathbf{x}} + y_2\hat{\mathbf{y}} + z_2\hat{\mathbf{z}}$$

$$\mathbf{r}_3 = x_3\hat{\mathbf{x}} + y_3\hat{\mathbf{y}} + z_3\hat{\mathbf{z}}$$

De manera similar, se definen los vectores relativos entre cada cuerpo de la siguiente manera:

$$\vec{\mathbf{z}}_{12} = (x_2 - x_1)\hat{\mathbf{x}} + (y_2 - y_1)\hat{\mathbf{y}} + (z_2 - z_1)\hat{\mathbf{z}}$$

$$\vec{\mathbf{z}}_{13} = (x_3 - x_1)\hat{\mathbf{x}} + (y_3 - y_1)\hat{\mathbf{y}} + (z_3 - z_1)\hat{\mathbf{z}}$$

$$\vec{\mathbf{z}}_{23} = (x_3 - x_2)\hat{\mathbf{x}} + (y_3 - y_2)\hat{\mathbf{y}} + (z_3 - z_2)\hat{\mathbf{z}}$$

Con el fin de simplificar, podemos establecer las siguientes definiciones para los vectores relativos:

$$\vec{\mathbf{z}}_{21} = -\vec{\mathbf{z}}_{12}$$

$$\vec{\mathbf{z}}_{31} = -\vec{\mathbf{z}}_{13}$$

$$\vec{\mathbf{z}}_{32} = -\vec{\mathbf{z}}_{23}$$

Finalmente, la representación espacial de nuestro sistema de tres cuerpos se muestra en la Figura 1, donde se observa la interacción entre los tres cuerpos en su configuración relativa.

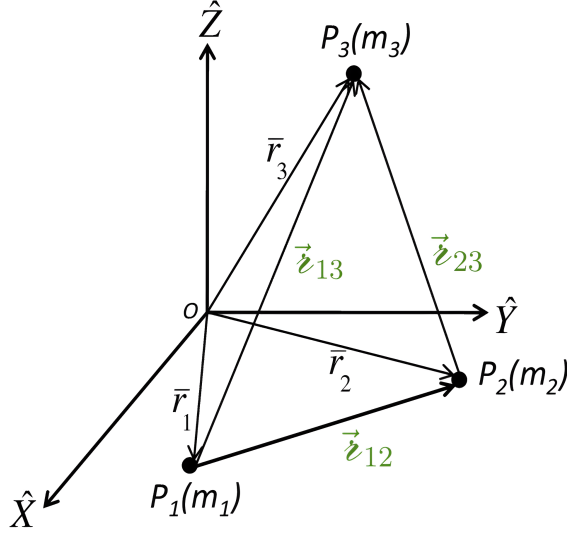


Figura 1: Diagrama tridimensional del sistema de tres cuerpos

Ahora, de acuerdo con la ley de gravitación universal, se puede expresar la fuerza de la siguiente manera:

$$\mathbf{F} = -G \frac{M \cdot m}{r^2} \hat{\mathbf{r}}$$

Donde, para nuestro sistema en particular, estaría definida por:

$$\mathbf{F}_T = \sum_{\substack{j=1 \\ j \neq i}}^N G \frac{m_j \cdot m_i}{r_{ij}^2} \hat{\mathbf{z}}_{ij}$$

En este caso, las fuerzas experimentadas por cada cuerpo se definen mediante:

$$\mathbf{F}_1 = Gm_1 \left[ \frac{m_2}{r_{12}^2} \hat{\mathbf{z}}_{12} + \frac{m_3}{r_{13}^2} \hat{\mathbf{z}}_{13} \right] \quad (1)$$

$$\mathbf{F}_1 = Gm_1 \left\{ \frac{m_2 [(x_2 - x_1) \hat{\mathbf{x}} + (y_2 - y_1) \hat{\mathbf{y}} + (z_2 - z_1) \hat{\mathbf{z}}]}{\left[ (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 \right]^{\frac{3}{2}}} + \frac{m_3 [(x_3 - x_1) \hat{\mathbf{x}} + (y_3 - y_1) \hat{\mathbf{y}} + (z_3 - z_1) \hat{\mathbf{z}}]}{\left[ (x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2 \right]^{\frac{3}{2}}} \right\}$$

$$\mathbf{F}_2 = Gm_2 \left[ -\frac{m_1}{r_{12}^2} \hat{\mathbf{z}}_{12} + \frac{m_3}{r_{23}^2} \hat{\mathbf{z}}_{23} \right] \quad (2)$$

$$\mathbf{F}_2 = Gm_2 \left\{ \frac{m_3 [(x_3 - x_2) \hat{\mathbf{x}} + (y_3 - y_2) \hat{\mathbf{y}} + (z_3 - z_2) \hat{\mathbf{z}}]}{\left[ (x_3 - x_2)^2 + (y_3 - y_2)^2 + (z_3 - z_2)^2 \right]^{\frac{3}{2}}} - \frac{m_1 [(x_2 - x_1) \hat{\mathbf{x}} + (y_2 - y_1) \hat{\mathbf{y}} + (z_2 - z_1) \hat{\mathbf{z}}]}{\left[ (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 \right]^{\frac{3}{2}}} \right\}$$

$$\mathbf{F}_3 = -Gm_3 \left[ \frac{m_2}{r_{23}^2} \hat{\mathbf{z}}_{23} + \frac{m_1}{r_{13}^2} \hat{\mathbf{z}}_{13} \right] \quad (3)$$

$$\mathbf{F}_3 = -Gm_3 \left\{ \frac{m_2 [(x_3 - x_2) \hat{\mathbf{x}} + (y_3 - y_2) \hat{\mathbf{y}} + (z_3 - z_2) \hat{\mathbf{z}}]}{\left[ (x_3 - x_2)^2 + (y_3 - y_2)^2 + (z_3 - z_2)^2 \right]^{\frac{3}{2}}} + \frac{m_1 [(x_3 - x_1) \hat{\mathbf{x}} + (y_3 - y_1) \hat{\mathbf{y}} + (z_3 - z_1) \hat{\mathbf{z}}]}{\left[ (x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2 \right]^{\frac{3}{2}}} \right\}$$

Para describir con precisión las trayectorias de cada cuerpo, es indispensable formular el sistema de ecuaciones diferenciales correspondiente, el cual se define a través de las ecuaciones (1), (2) y (3).

$$\ddot{\mathbf{r}}_1 = G \left[ \frac{m_2}{z_{12}^2} \hat{\mathbf{z}}_{12} + \frac{m_3}{z_{13}^2} \hat{\mathbf{z}}_{13} \right] \quad (4)$$

$$\ddot{\mathbf{r}}_2 = G \left[ -\frac{m_1}{z_{12}^2} \hat{\mathbf{z}}_{12} + \frac{m_3}{z_{23}^2} \hat{\mathbf{z}}_{23} \right] \quad (5)$$

$$\ddot{\mathbf{r}}_3 = -G \left[ \frac{m_2}{z_{23}^2} \hat{\mathbf{z}}_{23} + \frac{m_1}{z_{13}^2} \hat{\mathbf{z}}_{13} \right] \quad (6)$$

Este sistema será resuelto empleando métodos numéricos, específicamente a través del algoritmo Velocity Verlet, debido a la falta de una solución analítica disponible en la actualidad. Además, se propondrán diversas condiciones iniciales para establecer escenarios específicos.

### Experimento preliminar

Una vez que se programó el método numérico para resolver las ecuaciones diferenciales, se probó con tres objetos con masas similares adimensionales: 1, 1.5 y 2.0. Se eligieron las siguientes condiciones iniciales para probar el modelo y observar producir varias orbitaciones entre los cuerpos:

$$\begin{aligned} \mathbf{r}_1(0) &= [5.2, 0.0, 1.3] & \mathbf{v}_1(0) &= [0.1, 0.2, 0.0] \\ \mathbf{r}_2(0) &= [0.0, 0.0, 3.5] & \mathbf{v}_2(0) &= [0.4, 0.0, 0.2] \\ \mathbf{r}_3(0) &= [4.1, 0.0, 0.0] & \mathbf{v}_3(0) &= [0.0, 0.3, 0.4] \end{aligned}$$

Las trayectorias obtenidas se muestran en la figura 2. Se puede observar que se producen por lo menos 5 orbitaciones entre dos cuerpos de manera acotada, es decir sin que se salgan disparados.

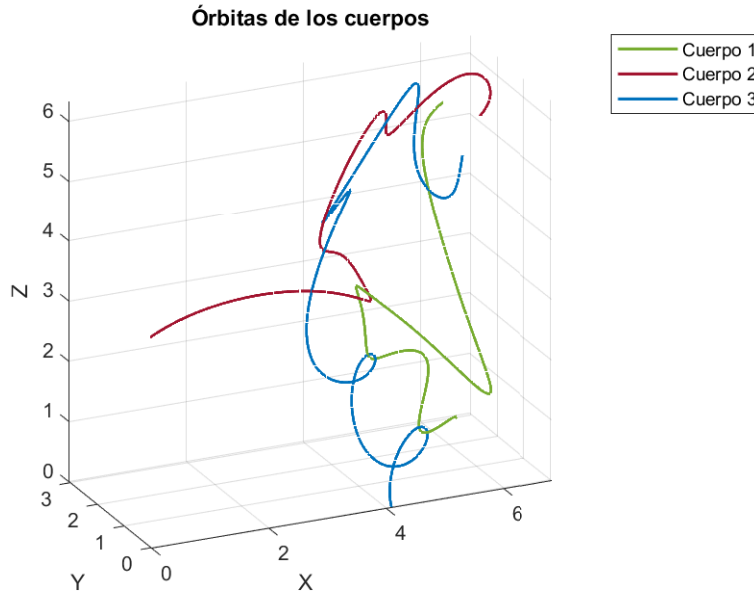


Figura 2: Trayectoria de los cuerpos para el experimento preliminar

### Comportamiento del centro de masas

A continuación se calculó el vector centro de masas que se expresa por:

$$\mathbf{r}_{cm} = \frac{m_1 * \mathbf{r}_1 + m_2 * \mathbf{r}_2 + m_3 * \mathbf{r}_3}{m_1 + m_2 + m_3}$$

Se gráfico también junto con los otros tres cuerpos con el fin de confirmar que tuviera dirección constante. Esto se puede observar en la figura 3.

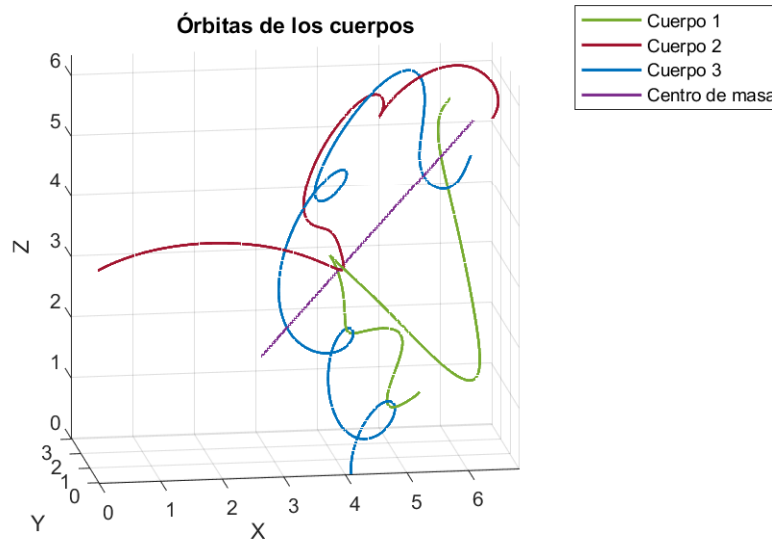


Figura 3: Comportamiento del centro de masa

Posteriormente se calculó la velocidad del centro de masas utilizando diferencias finitas y se gráfico su magnitud. Como se puede observar en la figura 4, esta es constante a lo largo de la simulación.

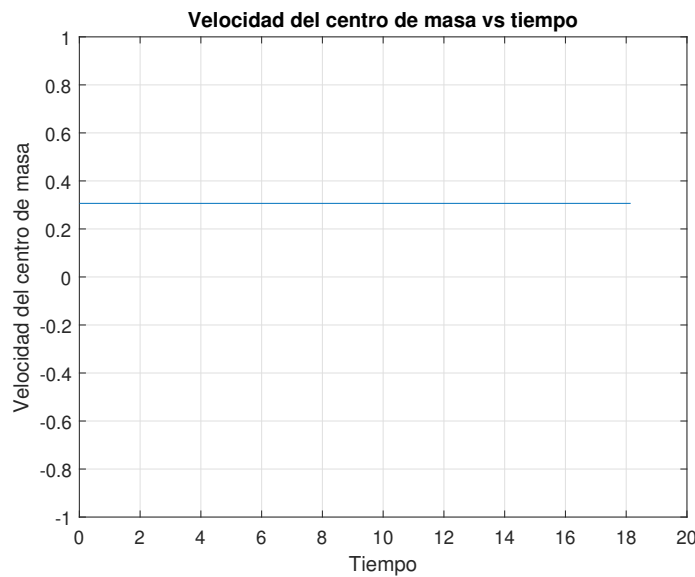


Figura 4: Conservación de la velocidad del centro de masas

## Conservación de la energía mecánica

La energía mecánica se define como la suma de la energía potencial y la cinética de la siguiente forma:

$$E = K + U$$

Para esto se calculó la energía cinética como:

$$K = \frac{1}{2}m|\mathbf{v}|^2$$

Y se realiza la suma para cada cuerpo con sus masas y la magnitud de sus velocidades respectivas. Por otro lado, para la energía potencial:

$$U = -\frac{GMm}{z}$$

Para el cuál se realiza para la interacción entre cada par de cuerpos, donde  $M$  es la masa del primer cuerpo,  $m$  la masa del segundo y  $z$  es la magnitud del vector de separación entre los dos cuerpos.

Luego, graficamos la energía cinética, la potencial y la mecánica con respecto al tiempo, y podemos observar como la energía mecánica no cambia, pues el crecimiento en la potencial se compensa con la cinética y viceversa.

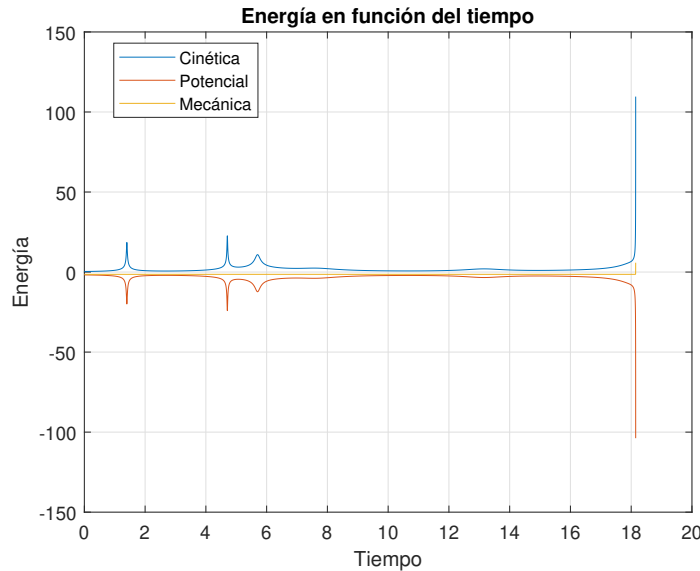


Figura 5: Comprobación de la conservación de la energía mecánica total

## Colisión entre dos cuerpos

Las condiciones iniciales que se usaron para el experimento preliminar, desde el inicio fueron seleccionadas para producir un choque (además de las orbitaciones). El procedimiento que se siguió fue posicionar los cuerpos de manera que no estuvieran en el mismo plano, y a partir de prueba y error se fueron modificando las velocidades iniciales hasta alcanzar un criterio de choque. Este criterio se definió como un acercamiento menor a 0.01 entre cada cuerpo.

En el código se incluyó también un paso para eliminar los datos de posición después de la colisión, ya que por error numérico los cuerpos salen disparados.

En la figura 6 se muestran la posición de la colisión, que de acuerdo con el resultado de la simulación este fue en (6.6024, 2.5447, 4.9644).

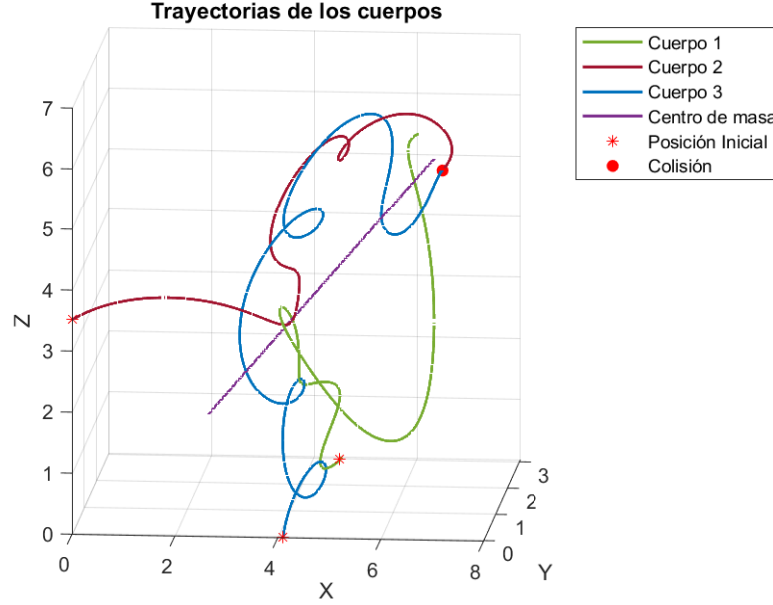


Figura 6: Colisión entre dos cuerpos en una trayectoria curvada

### Interacción del Sol, Júpiter y un asteroide

Para este caso se buscó utilizar datos realistas para las masas, posiciones y velocidades iniciales. Las unidades que se utilizaron son: Masas Solares, Unidades Astronómicas y años.

Las masas utilizadas en la simulación fueron:

$$m_S = 1M_{\odot}$$

$$m_J = 0.0009543M_{\odot}$$

$$m_a = 1e - 12M_{\odot}$$

Las condiciones iniciales que producen una órbita retrógrada de un Asteroide alrededor de Júpiter, que a su vez orbita el Sol son las siguientes:

$$\begin{aligned} \mathbf{r}_S(0) &= [0.0, 0.0, 0.0] & \mathbf{v}_S(0) &= [0.0, 0.0, 0.0] \\ \mathbf{r}_J(0) &= [0.0, 5.2, 0.0] & \mathbf{v}_J(0) &= [2\pi * 5.2/11.86, 0.0, 0.0] \\ \mathbf{r}_a(0) &= [0.0, 5.3, 0.0] & \mathbf{v}_a(0) &= [(2\pi * 5.2/11.86) * 0.8125, 0.0, 0.0] \end{aligned}$$

Las velocidades utilizadas en las condiciones iniciales para Júpiter se derivan de su periodo orbital alrededor del Sol. La expresión utilizada,  $2 * \pi * 5.2/11.86$ , es una aproximación de la velocidad angular de Júpiter en su órbita alrededor del Sol.

- 5.2 es la distancia promedio entre Júpiter y el Sol en unidades astronómicas (UA).
- 11.86 es el período orbital de Júpiter alrededor del Sol en años.
- $2 * \pi$  es utilizado para convertir el período orbital en una medida angular, ya que un círculo completo corresponde a  $2 * \pi$  radianes.

Es importante notar que Júpiter y el Asteroide se encuentran alineados en el eje x y en el eje y tienen una separación de 0.1 UA. Por otro lado, la velocidad inicial del asteroide es aproximadamente el 80 % de la de Júpiter y en la misma dirección para la órbita sea retrógrada.

En la figura 7 se muestran las trayectorias resultantes, vistas desde el plano  $xy$ .

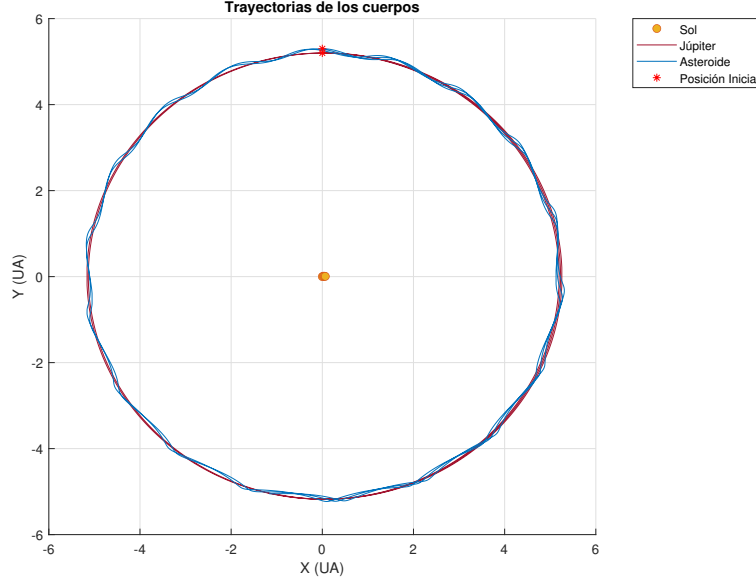


Figura 7: Asteroide en una órbita retrógrada alrededor de Júpiter

### Captura de un asteroide extrasolar por la gravedad de Júpiter

Para este último caso se buscaba simular la captura de un asteroide extra-solar por el campo gravitacional de Júpiter. Las condiciones iniciales que producen este resultado (en las mismas unidades que en el caso anterior) son las siguientes:

$$\begin{aligned} \mathbf{r}_S(0) &= [0.0, 0.0, 0.0] & \mathbf{v}_S(0) &= [0.0, 0.0, 0.0] \\ \mathbf{r}_J(0) &= [5.2, 0.0, 0.0] & \mathbf{v}_J(0) &= [0.0, 2\pi * 5.2/11.86, 0.0] \\ \mathbf{r}_a(0) &= [5.2, 0.0, 0.5] & \mathbf{v}_a(0) &= [0.0, (2\pi * 5.2/11.86) * 0.99, 0.1] \end{aligned}$$

En esta simulación, el asteroide se encuentra inicialmente a una distancia de 5.2239 UA del Sol, con velocidades iniciales en las direcciones  $y$  y  $z$ . Durante las pruebas iniciales, se intentó colocar al asteroide en el mismo plano orbital que Júpiter, requiriendo una velocidad lo suficientemente grande para acercarse al planeta, pero no tan grande como para que fuera capturado por el Sol y desviado. Después de varios experimentos, se logró obtener el resultado deseado. Como se puede observar en la figura 8, el asteroide queda en una órbita alrededor de Júpiter en el eje  $z$ .



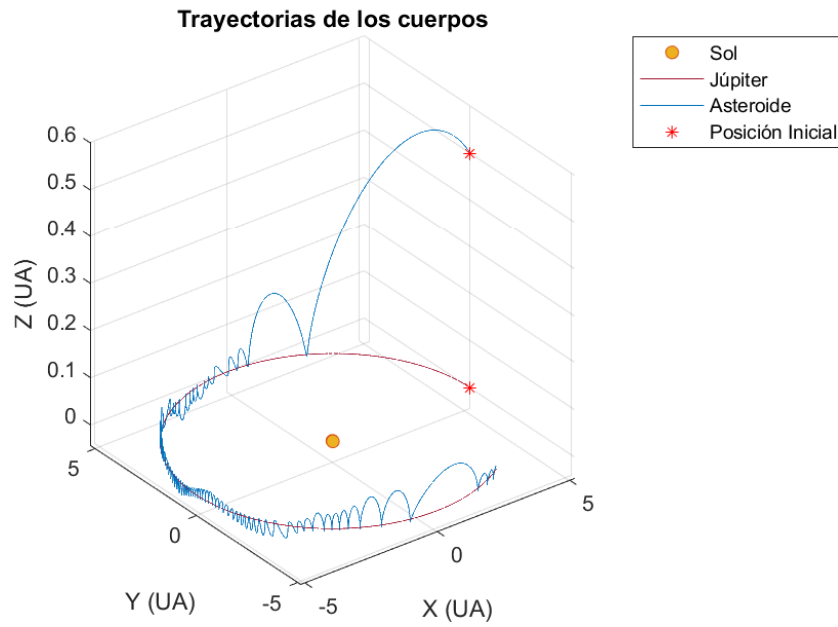


Figura 8: Captura de un asteroide extr asolar por Júpiter

Primero se ven oscilaciones inestables, pero para un período de tiempo prolongado, se observa que la órbita se estabiliza. En la figura 9, se puede apreciar este comportamiento.

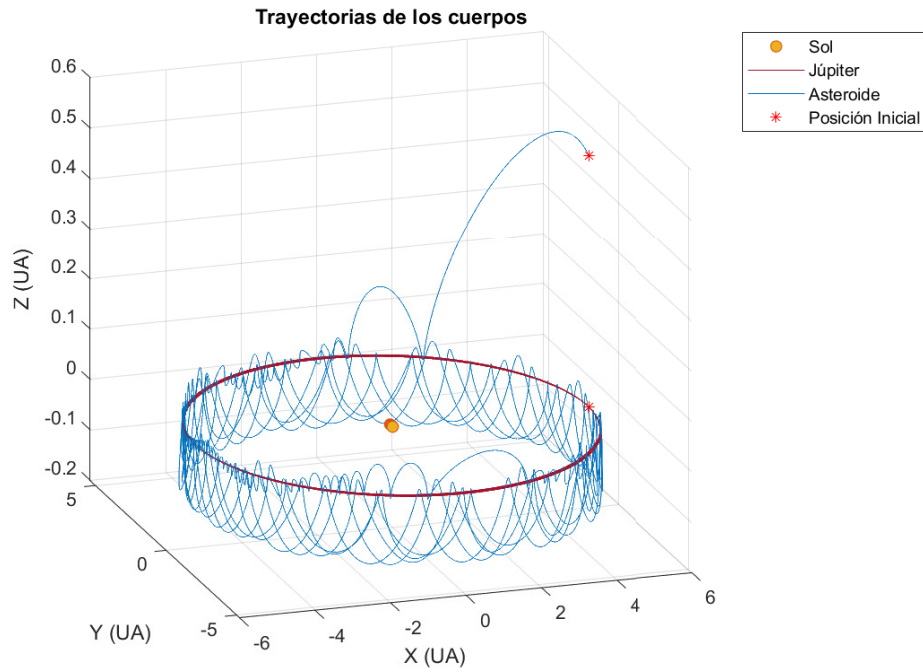


Figura 9: Órbita de un asteroide extra solar capturado por Júpiter para un periodo de tiempo prolongado,

Para un futuro proyecto, se podría intentar capturar un asteroide desde una distancia mayor, mediante la incorporación de un evento adicional que afecte la velocidad del asteroide. Esta técnica se conoce comúnmente como transferencia de Hohmann, donde se logra una transferencia de órbita mediante un impulso adicional. En este caso, se propuso posicionar el asteroide de manera que, al iniciar su órbita alrededor del Sol, tenga tiempo y oportunidad de acoplarse a la órbita de Júpiter desde un plano diferente, gracias a una velocidad adicional en la dirección  $z$ . Tanto la posición como este parámetro se variaron hasta lograr el atrapamiento deseado, el cual se pudo justificar durante un período prolongado.

## Conclusión

El problema de los tres cuerpos ha sido uno de los dilemas más grandes para los físicos de la edad moderna. Aunque no se ha encontrado al día de hoy una solución analítica, fue posible modelar varios casos mediante métodos numéricos en este reto.

Se puede decir que se cumplieron con los objetivos planteados. Se modeló un sistema en el espacio tridimensional, que pudiera admitir tres masas, cada una con sus respectivas velocidades y posiciones iniciales arbitrarias. Se aplicó la teoría vista en clase para calcular la fuerza gravitacional que cada cuerpo ejerce fuerza sobre los otros dos, y solucionar las trayectorias. El código implementado permite la variación de parámetros como la masa y las condiciones iniciales, para así poder modelar diversos sistemas. En este caso, se modeló uno con masas similares, y dos casos para el sistema del Sol, Júpiter y un asteroide. También se pudo comprobar la conservación de la energía mecánica total y el centro de masa.

Este tipo de soluciones es muy útil ya que nos sirve para modelar y entender el comportamiento del universo en su organización con respecto a las masas y el uso de la mecánica clásica. Además, se pueden hacer predicciones de sucesos futuros, como la posibilidad que un asteroide entrara a nuestro sistema, o que dos cuerpos choquen y las consecuencias que estos tendrían.

Pudimos notar que el sistema efectivamente es caótico, pues con una pequeña variación a las condiciones iniciales del sistema, este se comporta de manera muy diferente. Un ejemplo fue el atrapamiento del asteroide, pues para que este quedara retenido en la órbita de Júpiter, tuvimos que ajustar las condiciones varias veces, modificando solamente unos pequeños dígitos.

## Código implementado

A continuación se muestra el código en MATLAB dividido por secciones. Las condiciones iniciales y las gráficas de las trayectorias cambian dependiendo del caso. Si no se especifica en el título de la sección, aplica para todos los casos.

```
1 %% Inicialización (Choque)
2 % Parámetros del sistema
3 G = 1;    % Constante de gravitación universal
4 dt = 0.001;    % Paso de tiempo
5 TN = 19;    % Tiempo final de simulación
6
7 % Masas
8 M1 = 1.5;
9 M2 = 2.0;
10 M3 = 1.0;
11
12 % Posiciones y velocidades iniciales
13 r1_0 = [5.2, 0.0, 1.3];
14 v1_0 = [0.1, 0.2, 0.0];
15
16 r2_0 = [0.0, 0.0, 3.5];
17 v2_0 = [0.4, 0.0, 0.2];
18
19 r3_0 = [4.1, 0.0, 0.0];
20 v3_0 = [0.0, 0.3, 0.4];
21
22 %% Inicialización (órbita retrógrada)
23 % Parámetros del sistema (Masas solares, UA y años)
24 G = 39.478;    % Constante de gravitación universal
25 dt = 0.001;    % Paso de tiempo
26 TN = 30;    % Tiempo final de simulación
27
28 % Masas
29 M1 = 1; % Masa del Sol
30 M2 = 0.0009543; % Masa de Júpiter
31 M3 = 1e-12; % Masa del Asteroide
32
33 % Posiciones y velocidades iniciales
34 r1_0 = [0, 0, 0]; v1_0 = [0, 0, 0.0];
35
36 r2_0 = [0.0, 5.2, 0.0]; v2_0 = [2*pi*5.2/11.86, 0.0, 0.0];
37
38 r3_0 = [0.0, 5.3, 0]; v3_0 = [(2*pi*5.2/11.86)*0.8125, 0, 0];
39
40 %% Inicialización (atrapamiento)
41 % Parámetros del sistema
42 G = 39.478;    % Constante de gravitación universal
43 dt = 0.001;    % Paso de tiempo
44 TN = 10;    % Tiempo final de simulación
45
```

```

46 % Masas
47 M1 = 1;
48 M2 = 0.0009543;
49 M3 = 1e-12;
50
51 % Posiciones y velocidades iniciales
52 r1_0 = [0, 0, 0]; v1_0 = [0, 0,0.0];
53
54 r2_0 = [5.2, 0.0, 0.0]; v2_0 = [0.0, 2*pi*5.2/11.86, 0.0];
55
56 r3_0 = [5.2, 0.0, 0.5]; v3_0 = [0.0, (2*pi*5.2/11.86)*0.99,0.1];
57
58 %% Inicialización de variables
59 N = ceil(TN/dt); % Número de pasos de tiempo
60 t=linspace(0,TN,N);
61
62 %Posiciones
63 r1 = zeros(N, 3);
64 r2 = zeros(N, 3);
65 r3 = zeros(N, 3);
66
67 % Velocidades
68 v1 = zeros(N, 3);
69 v2 = zeros(N, 3);
70 v3 = zeros(N, 3);
71
72 % Velocidades para medio paso
73 v1_half = zeros(N, 3); v2_half = zeros(N,3); v3_half = zeros(N, 3);
74
75 % Asignar condiciones iniciales
76 r1(1, :) = r1_0;
77 r2(1, :) = r2_0;
78 r3(1, :) = r3_0;
79
80 v1(1, :) = v1_0;
81 v2(1, :) = v2_0;
82 v3(1, :) = v3_0;
83
84 %% Bucle principal de simulación
85 for i = 1:N-1
86     % Cálculo de las fuerzas
87     F1 = calculateForce(r1(i, :), r2(i, :), M1, M2, G) + calculateForce(
r1(i, :), r3(i, :), M1, M3, G);
88     F2 = calculateForce(r2(i, :), r1(i, :), M2, M1, G) + calculateForce(
r2(i, :), r3(i, :), M2, M3, G);
89     F3 = calculateForce(r3(i, :), r1(i, :), M3, M1, G) + calculateForce(
r3(i, :), r2(i, :), M3, M2, G);
90
91     % Velocidad (half step)
92     v1_half(i,:) = v1(i,:) + 0.5 * F1/M1 * dt;

```

```

93     v2_half(i,:) = v2(i,:) + 0.5 * F2/M2 * dt;
94     v3_half(i,:) = v3(i,:) + 0.5 * F3/M3 * dt;
95
96     % Actualización de las posiciones
97     r1(i+1, :) = r1(i, :) + v1_half(i, :) * dt;
98     r2(i+1, :) = r2(i, :) + v2_half(i, :) * dt;
99     r3(i+1, :) = r3(i, :) + v3_half(i, :) * dt;
100
101     % Cálculo de las nuevas fuerzas
102     F1_new = calculateForce(r1(i+1, :), r2(i+1, :), M1, M2, G) +
calculateForce(r1(i+1, :), r3(i+1, :), M1, M3, G);
103     F2_new = calculateForce(r2(i+1, :), r1(i+1, :), M2, M1, G) +
calculateForce(r2(i+1, :), r3(i+1, :), M2, M3, G);
104     F3_new = calculateForce(r3(i+1, :), r1(i+1, :), M3, M1, G) +
calculateForce(r3(i+1, :), r2(i+1, :), M3, M2, G);
105
106     % Cálculo de las velocidades
107     v1(i+1, :) = v1_half(i, :) + 0.5 * F1_new/M1 * dt;
108     v2(i+1, :) = v2_half(i, :) + 0.5 * F2_new/M2 * dt;
109     v3(i+1, :) = v3_half(i, :) + 0.5 * F3_new/M3 * dt;
110 end
111
112 %% Criterio de choque
113 threshold = 0.01; % Valor máximo de acercamiento para definir un choque
114 collision = false;
115 collision_position = zeros(1, 3); % Almacena la posición de la colisión
116
117 for i = 1:N
118     if norm(r1(i, :) - r2(i, :)) <= threshold
119         collision = true;
120         collision_position = (r1(i, :) + r2(i, :))/2; % Calcula el
promedio de las posiciones de los cuerpos en el momento de la colisió
n
121         collision_index=i;
122         break;
123     end
124     if norm(r1(i, :) - r3(i, :)) <= threshold
125         collision = true;
126         collision_position = (r1(i, :) + r3(i, :))/2; % Calcula el
promedio de las posiciones de los cuerpos en el momento de la colisió
n
127         collision_index=i;
128         break;
129     end
130     if norm(r2(i, :) - r3(i, :)) <= threshold
131         collision = true;
132         collision_position = (r2(i, :) + r3(i, :))/2; % Calcula el
promedio de las posiciones de los cuerpos en el momento de la colisió
n
133         collision_index=i;

```

```

134         break;
135     end
136 end
137
138 if collision
139     disp('Se ha producido un choque entre dos cuerpos. ');
140     disp(['Posición de la colisión: (', num2str(collision_position(1)),
141         ', ', num2str(collision_position(2)), ', ', num2str(
142         collision_position(3)), ')']);
143     % Elimina los datos después de la colisión
144     r1(collision_index:end,:)=[];
145     r2(collision_index:end,:)=[];
146     r3(collision_index:end,:)=[];
147
148     v1(collision_index:end,:)=[];
149     v2(collision_index:end,:)=[];
150     v3(collision_index:end,:)=[];
151
152     t(collision_index:end)=[];
153     N=collision_index;
154 else
155     disp('No se ha producido ningún choque. ');
156 end
157
158 %% Cálculo del centro de masas
159 r_cm = (M1*r1 + M2*r2 + M3*r3)/(M1 + M2 + M3);
160 v_cm = zeros(N-1,3);
161 for i=2:N-2
162     v_cm(i,1)=(r_cm(i+1,1)-r_cm(i-1,1))/(2*dt);
163     v_cm(i,2)=(r_cm(i+1,2)-r_cm(i-1,2))/(2*dt);
164     v_cm(i,3)=(r_cm(i+1,3)-r_cm(i-1,3))/(2*dt);
165 end
166 v_cm_magnitude = sqrt(sum(v_cm.^2,2));
167 figure(3)
168 plot(t(2:end-1), v_cm_magnitude(2:end-1))
169 title('Velocidad del centro de masa vs tiempo')
170 ylim ([v_cm(1)-1 v_cm(1)+1])
171 xlabel('Tiempo')
172 ylabel('Velocidad del centro de masa')
173 grid on
174
175 %% Cálculo de la energía mecánica
176 K1 = 0.5*M1*sum(v1.^2, 2);
177 K2 = 0.5*M2*sum(v2.^2, 2);
178 K3 = 0.5*M3*sum(v3.^2, 2);
179 U = -G*M1*M2./sqrt(sum((r1 - r2).^2, 2)) - G*M1*M3./sqrt(sum((r1 - r3)
180     .^2, 2)) - G*M2*M3./sqrt(sum((r2 - r3).^2, 2));
181 K = K1 + K2 + K3;
182 E = K + U;
183 figure(4);

```

```

181 plot(t,K,t,U,t,E);
182 title('Energía en función del tiempo');
183 xlabel('Tiempo');
184 ylabel('Energía');
185 legend('Cinética','Potencial','Mecánica','Location','best')
186 grid on;
187
188 %% Gráfico de las trayectorias y animación (Tres cuerpos cualquiera)
189 figure(2);
190 h(1)=plot3(r1(:, 1), r1(:, 2), r1(:, 3), 'Color', "#77AC30",LineWidth
    =1.5);
191 hold on;
192 h(2)=plot3(r2(:, 1), r2(:, 2), r2(:, 3), 'Color', "#A2142F",LineWidth
    =1.5);
193 h(3)=plot3(r3(:, 1), r3(:, 2), r3(:, 3), 'Color', "#0072BD",LineWidth
    =1.5);
194 h(4)=plot3(r_cm(:,1),r_cm(:,2),r_cm(:,3),LineWidth=1.5);
195 h(5)=plot3(r1(1,1),r1(1,2),r1(1,3),'r*');
196 h(6)=plot3(r2(1,1),r2(1,2),r2(1,3),'r*');
197 h(7)=plot3(r3(1,1),r3(1,2),r3(1,3),'r*');
198 title('Trayectorias de los cuerpos');
199 xlabel('X');
200 ylabel('Y');
201 zlabel('Z');
202 if norm(collision_position)>0
203     h(8)=scatter3(collision_position(1),collision_position(2),
        collision_position(3),'filled','r');
204     legend(h([1,2,3,4,5,8]),{'Cuerpo 1', 'Cuerpo 2', 'Cuerpo 3','Centro
        de masa','Posición Inicial','Colisión'});
205 else
206     legend(h([1,2,3,4,5]),{'Cuerpo 1', 'Cuerpo 2', 'Cuerpo 3','Centro de
        masa','Posición Inicial'})
207 end
208 grid on;
209
210 %Animación de las órbitas
211 figure(1);
212 h=plot3(r1(:, 1), r1(:, 2), r1(:, 3),'Color', "#77AC30",LineWidth=1.5);
213 hold on;
214 h2=plot3(r2(:, 1), r2(:, 2), r2(:, 3), 'Color', "#A2142F",LineWidth=1.5)
    ;
215 h3=plot3(r3(:, 1), r3(:, 2), r3(:, 3), 'Color', "#0072BD",LineWidth=1.5)
    ;
216 h4=plot3(r_cm(:,1),r_cm(:,2),r_cm(:,3),LineWidth=1.5);
217 title('Órbitas de los cuerpos');
218 xlabel('X');
219 ylabel('Y');
220 zlabel('Z');
221 legend('Cuerpo 1', 'Cuerpo 2', 'Cuerpo 3','Centro de masa');
222 grid on;

```

```

223 axis equal;
224
225 skip = 150; % Saltar 'skip' pasos de tiempo en cada iteración de la
      animación
226
227 for i = 1:skip:N-1
228     set(h, 'XData', r1(1:i, 1), 'YData', r1(1:i, 2), 'ZData', r1(1:i, 3)
      );
229     set(h2, 'XData', r2(1:i, 1), 'YData', r2(1:i, 2), 'ZData', r2(1:i,
      3));
230     set(h3, 'XData', r3(1:i, 1), 'YData', r3(1:i, 2), 'ZData', r3(1:i,
      3));
231     set(h4, 'XData', r_cm(1:i, 1), 'YData', r_cm(1:i, 2), 'ZData', r_cm
      (1:i, 3));
232     drawnow;
233 end
234 %% Gráfico de las trayectorias y animación (Sol, Jupiter y Asteroide)
235 figure(2);
236 h(1)=scatter3(r1(:, 1), r1(:, 2), r1(:, 3), 'MarkerFaceColor', "#EDB120
      ", 'MarkerEdgeColor', "#D95319");
237 hold on;
238 h(2)=plot3(r2(:, 1), r2(:, 2), r2(:, 3), 'Color', "#A2142F");
239 h(3)=plot3(r3(:, 1), r3(:, 2), r3(:, 3), 'Color', "#0072BD");
240 % h(4)=plot3(r_cm(:,1),r_cm(:,2),r_cm(:,3));
241 % h(5)=plot3(r1(1,1),r1(1,2),r1(1,3),'r*');
242 h(4)=plot3(r2(1,1),r2(1,2),r2(1,3),'r*');
243 h(5)=plot3(r3(1,1),r3(1,2),r3(1,3),'r*');
244 title('Trayectorias de los cuerpos');
245 xlabel('X (UA)');
246 ylabel('Y (UA)');
247 zlabel('Z (UA)');
248 if norm(collision_position)>0
249     h(6)=scatter3(collision_position(1),collision_position(2),
      collision_position(3),'filled','r');
250     legend(h([1,2,3,4,6]),{'Sol', 'Júpiter', 'Asteroide','Posición
      Inicial','Colisión'});
251 else
252     legend(h([1,2,3,4]),{'Sol', 'Júpiter', 'Asteroide','Posición Inicial
      '});
253 end
254 grid on;
255
256 % Animación de las órbitas
257 figure(1);
258 h=scatter3(r1(:, 1), r1(:, 2), r1(:, 3), 'MarkerFaceColor', "#EDB120", '
      MarkerEdgeColor', "#D95319");
259 hold on;
260 h2=plot3(r2(:, 1), r2(:, 2), r2(:, 3), 'Color', "#A2142F");
261 h3=plot3(r3(:, 1), r3(:, 2), r3(:, 3), 'Color', "#0072BD");
262 title('Órbitas de los cuerpos');

```



```

263 xlabel('X');
264 ylabel('Y');
265 zlabel('Z');
266 legend('Sol', 'Júpiter', 'Asteroide');
267 grid on;
268 axis equal;
269
270 skip = 150; % Saltar 'skip' pasos de tiempo en cada iteración de la
    animación
271
272 for i = 1:skip:N-1
273     set(h, 'XData', r1(1:i, 1), 'YData', r1(1:i, 2), 'ZData', r1(1:i, 3)
    );
274     set(h2, 'XData', r2(1:i, 1), 'YData', r2(1:i, 2), 'ZData', r2(1:i,
    3));
275     set(h3, 'XData', r3(1:i, 1), 'YData', r3(1:i, 2), 'ZData', r3(1:i,
    3));
276     drawnow;
277 end
278
279 % Función para calcular la fuerza gravitacional entre dos cuerpos
280 function F = calculateForce(r1, r2, M1, M2, G)
281     r = r2 - r1;
282     dist = norm(r);
283     F = G*M1*M2/(dist^3) * r;
284 end

```