



UNIVERSIDAD
POLITÉCNICA
DE MADRID

AMPLIACIÓN DE MATEMÁTICAS II

Programación

M^a Blanca Boado Cuartero
David Fernández Pulido
Ignacio García Guerrero

7 de abril de 2019

MÁSTER UNIVERSITARIO EN SISTEMAS ESPACIALES

Índice

1. Introducción	1
2. Estabilidad de los puntos de Lagrange	2
3. Matlab	4
3.1. Obtención de los puntos de Lagrange	4
3.2. Análisis de estabilidad	6
3.3. Integración numérica	7
3.3.1. Dormann Prince (ODE45)	8
3.3.2. Adam-Basforth-Moulton (ODE113)	8
3.3.3. Análisis de resultados	9
4. GMAT	11
4.1. Resultados y comparativa	13
5. Conclusiones	17

1. Introducción

El principal objetivo de este semestre era calcular los puntos de Lagrange y estudiar su estabilidad utilizando diferentes integradores, mientras que el objetivo secundario de este semestre era comparar los diferentes integradores disponibles en Matlab en el cálculo de las órbitas alrededor de los puntos de Lagrange. Los puntos de Lagrange son las cinco posiciones en un sistema orbital cerca de dos cuerpos grandes donde un objeto más pequeño puede estar manteniendo su posición relativa a los dos cuerpos. En otros puntos el objeto acabaría orbitando alrededor de uno de los dos cuerpos, pero en los puntos de Lagrange las fuerzas gravitatorias de los dos cuerpos grandes y las fuerzas centrípetas del movimiento orbital se cancelarían, lo que causaría que el objeto se mantuviese estacionario en relación a los dos cuerpos.

Los puntos de Lagrange son exclusivos para cada combinación de dos cuerpos orbitales, es decir, existen cinco puntos para la combinación Tierra-Luna y otros cinco diferentes para Tierra-Sol. Teóricamente, los tres primeros puntos (de denominación L_1 , L_2 y L_3) están en línea recta, juntando los centros de los dos cuerpos grandes y son los puntos inestables. Los puntos L_4 y L_5 forman un triángulo equilátero con los centros de los cuerpos y son los puntos estables, que quiere decir que los objetos pueden orbitar alrededor de ellos en un sistema de coordenadas relacionado con los dos cuerpos. En la Figura 1 se puede ver una representación de los puntos de Lagrange de un sistema Planeta-Sol.

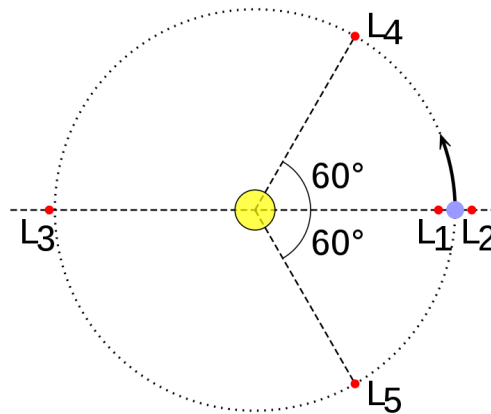


Figura 1 – Puntos de Lagrange

2. Estabilidad de los puntos de Lagrange

Para investigar la estabilidad de los puntos de Lagrange primero estableceremos las ecuaciones de movimiento del objeto pequeño.

$$\begin{aligned}\ddot{x} &= 2\dot{y} + 1 - \frac{(1-\mu)(x+\mu)}{d^3} - \frac{\mu(x-1+\mu)}{r^3} \\ \ddot{y} &= -2\dot{x} + y - \frac{(1-\mu)y}{d^3} - \frac{\mu y}{r^3} \\ \ddot{z} &= -\frac{(1-\mu)z}{d^3} - \frac{\mu z}{r^3}\end{aligned}\tag{1}$$

Donde el radio y la distancia vienen definidos por las siguientes ecuaciones.

$$d^2 = (x+\mu)^2 + y^2 + z^2\tag{2}$$

$$r^2 = (x-1+\mu)^2 + y^2 + z^2\tag{3}$$

Siendo el parámetro μ la relación entre las masas de los dos cuerpos grandes.

$$\mu = \frac{m_1}{m_1 + m_2}\tag{4}$$

donde m_1 (en nuestro caso, la Luna) es mucho menor que m_2 (la Tierra), por que μ sera menor que la unidad.

Como sabemos que la velocidad $(\dot{x}, \dot{y}, \dot{z})$ y la aceleración $(\ddot{x}, \ddot{y}, \ddot{z})$ son cero, resolvemos el sistema de las ecuaciones del movimiento para encontrar los puntos de Lagrange. Una vez tenemos los puntos debemos estudiar su estabilidad para ver si son los correctos. Para ello calculamos la matriz jacobiana de las velocidades y aceleraciones. Nuestra matriz será:

$$A = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}\tag{5}$$

Para mejor visualización de la matriz jacobiana en el informe haremos el siguiente cambio:

$$\begin{aligned}\ddot{x} &= 2\dot{y} + \Omega_x \\ \ddot{y} &= -2\dot{x} + \Omega_y \\ \ddot{z} &= \Omega_z\end{aligned}\tag{6}$$

Y la matriz jacobiana con respecto a las posiciones x, y y z y las velocidades en esos puntos será:

$$J = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \Omega_{xx} & \Omega_{xy} & \Omega_{xz} & 0 & 2 & 0 \\ \Omega_{xy} & \Omega_{yy} & \Omega_{yz} & -2 & 0 & 0 \\ \Omega_{xz} & \Omega_{yz} & \Omega_{zz} & 0 & 0 & 0 \end{bmatrix}\tag{7}$$

Donde las Ω_{nn} representan las dobles derivadas de las ecuaciones del movimiento.

Sustituyendo los puntos dentro de la matriz jacobiana y calculando sus autovalores podremos ver la estabilidad de cada punto. El siguiente paso es pintar las órbitas del objeto pequeño en cada punto para visualizar su estabilidad. Para esto le añadiremos una perturbación a la velocidad en x .

3. Matlab

Se procede, a continuación, a la implementación de todo lo comentado de manera teórica en el apartado anterior, usando la herramienta software MATLAB para la obtención y representación de los resultados.

El código generado se compone de diferentes funciones, cada una de las cuales resuelve una sección del problema de los 3 cuerpos planteado. De esta manera resulta más sencillo de implementar y facilita la visualización.

3.1. Obtención de los puntos de Lagrange

En primer lugar para la resolución del sistema se emplean las funciones "LagrangePtsCalculator" y "LagrangePtsOrder". La primera de ellas recibe como input el producto de la constante de gravitación universal por la masa del cuerpo, así como las incógnitas x, y, z para la resolución del sistema. Dentro de la función se plantean las ecuaciones que gobiernan el movimiento de los tres cuerpos y se resuelve el sistema mediante la función solve de MATLAB. Esta función solve devuelve una matriz de puntos que son solución del sistema. Sin embargo, no necesitamos todos los puntos obtenidos. Es por ello que se realiza en su interior la llamada a la función "LagrangePtsOrder".

```
function [L]=LagrangePtsCalculator(mu,vars)
x=vars(1); y=vars(2); z=vars(3);

eqns=[x-(1-mu)*(x+mu)/((x+mu)^2+y^2+z^2)^(3/2)-mu*(x-1+mu)/((x-1+mu)^2+y^2+z^2)^(3/2) == 0
      y-(1-mu)*y/((x+mu)^2+y^2+z^2)^(3/2)-mu*y/((x-1+mu)^2+y^2+z^2)^(3/2) == 0
      -(1-mu)*z/((x+mu)^2+y^2+z^2)^(3/2)-mu*z/((x-1+mu)^2+y^2+z^2)^(3/2) == 0];

pts=solve(eqns,vars);

L=LagrangePtsOrder(pts);

end
```

Figura 2 – Función "LagrangePtsCalculator".

La función "LagrangePtsOrder" se encarga del filtrado y la ordenación de toda la matriz solución del sistema, así como de la conversión del tipo de datos a un formato tipo double, legible para el usuario. Esta función devuelve finalmente los 5 puntos de Lagrange que se necesitan.

El filtrado se realiza a partir del hecho de que los puntos de Lagrange se encuentran en el mismo plano $z=0$, por lo que se descartan aquellos puntos cuyas componentes real o imaginaria en la dirección z , sean distintas de cero.

Además se sabe que las coordenadas de los puntos deben situarse en el plano real, por lo que se filtran los resultados descartando aquellas soluciones con componente imaginaria.

```
function [L]=LagrangePtsOrder(pts)
    pts=double([vpa(pts.x),vpa(pts.y),vpa(pts.z)]);

    f=find( imag(pts(:,1))>-1e-10 & imag(pts(:,1))<1e-10 & imag(pts(:,2)) == 0 ...
            & real(pts(:,3)) == 0 & imag(pts(:,3)) == 0);

    pts=real(pts(f,:));

    L(1,:)=pts(1,:); L(2,:)=pts(5,:); L(3,:)=pts(7,:);
    L(4,:)=pts(4,:); L(5,:)=pts(3,:);
end
```

Figura 3 – Función "LagrangePtsOrder".

Como resultado final se obtiene una matriz con las coordenadas de los puntos de Lagrange ordenados tal y como se conocen en la realidad. Las coordenadas de los puntos de Lagrange obtenidos se presentan en la siguiente tabla:

Tabla 1 – Coordenadas puntos de Lagrange.

	X	Y
L1	0.8367	0
L2	1.1559	0
L3	-1.003	0
L4	0.4878	0.866
L5	0.4878	-0.866

3.2. Análisis de estabilidad

Una vez conocidos los 5 puntos de Lagrange se procede a la realización de un análisis de su estabilidad y su comparación con los datos reales conocidos. Para ello, tal y como se ha comentado en anteriores apartados, es necesario el cálculo de la matriz Jacobiana y la obtención de sus autovalores.

Se ha implementado una función llamada "JacobianMatrix", que se encarga de definir la matriz Jacobiana mediante la definición de las derivadas de las ecuaciones que rigen el comportamiento del sistema. La función tiene como output la matriz 7.

```
function [A]=JacobianMatrix(mu,x,y,z)
d=sqrt((x+mu)^2+y^2+z^2);
r=sqrt((x-1+mu)^2+y^2+z^2);

omega_xx = 1-(1-mu)/d^3-mu/r^3+3*(1-mu)*(x+mu)^2/d^5+3*mu*(x-1+mu)^2/r^5;
omega_yy = 1-(1-mu)/d^3-mu/r^3+3*(1-mu)*y^2/d^5+3*mu*y^2/r^5;
omega_zz = -(1-mu)/d^3-mu/r^3+3*(1-mu)*z^2/d^5+3*mu*(x-1+mu)^2/r^5;
omega_xy = 3*(1-mu)*(x+mu)*y/d^5+3*mu*(x-1+mu)*y/r^5;
omega_yx = omega_xy;
omega_xz = 3*(1-mu)*(x+mu)*z/d^5+3*mu*(x-1+mu)*z/r^5;
omega_zx = omega_xz;
omega_yz = 3*(1-mu)*z*y/d^5+3*mu*z*y/r^5;
omega_zy = omega_yz;

A=vpa([0          0          0          1          0          0;
       0          0          0          0          1          0;
       0          0          0          0          0          1;
       omega_xx   omega_xy   omega_xz   0          2          0;
       omega_xy   omega_yy   omega_yz  -2          0          0;
       omega_xz   omega_yz   omega_zz   0          0          0]);

end
```

Figura 4 – Función "JacobianMatrix".

Una vez obtenida la matriz Jacobiana, se substituye en ella las coordenadas del punto de Lagrange introducido por el usuario y se calculan los autovalores de la matriz en dicho punto. De esta manera es posible conocer la estabilidad de cada punto de Lagrange. Los resultados se muestran en la siguiente tabla:

Tabla 2 – Autovalores de cada matriz para cada punto de Lagrange.

L1	L2	L3	L4	L5
$0 + 1.01i$	$-2.15 + 0i$	$-2.93 + 0i$	$0 + 0.95i$	$0 + 0.95i$
$0 - 1.01i$	$2.15 + 0i$	$2.93 + 0i$	$0 - 0.95i$	$0 - 0.95i$
$-0.22 + 0i$	$0 + 1.86i$	$0 + 2.33i$	$0 + 0.29i$	$0 + 0.29i$
$0.22 + 0i$	$0 - 1.86i$	$0 - 2.33i$	$0 - 0.29i$	$0 - 0.29i$
$0 + 1i$	$2.12 + 0i$	$2.33 + 0i$	$0 + 0.99i$	$0 + 0.99i$
$0 - 1i$	$-2.12 + 0i$	$-2.33 - 0i$	$0 - 0.99i$	$0 - 0.99i$

Observando los resultados se aprecia de manera inmediata que los puntos L1, L2 y L3 son inestables por tener alguno de sus autovalores parte real positiva, mientras que los puntos L4 y L5, serán marginalmente estables puesto que sus autovalores se encuentran sobre el eje imaginario.

3.3. Integración numérica

Conocidos los puntos de Lagrange y su estabilidad se procede a la integración numérica del movimiento de un cuerpo situado en cada uno de esos puntos ante una perturbación dada.

Para ello es necesario la creación de una función que devuelva la matriz del sistema y tenga como input el vector de estado, "systemFunction.^{en} nuestro caso.

```
function [F]=systemFunction(~,U,mu)

% U(1)=x U(2)=y U(3)=z U(4)=x' U(5)=y' U(6)=z'
d=sqrt((U(1)+mu)^2+U(2)^2+U(3)^2);
r=sqrt((U(1)-1+mu)^2+U(2)^2+U(3)^2);

F=[ U(4);
    U(5);
    U(6);
    U(1)+2*U(5)-(1-mu)*(U(1)+mu)/d^3-mu*(U(1)-1+mu)/r^3;
    U(2)-2*U(4)-(1-mu)*U(2)/d^3-mu*U(2)/r^3;
    -(1-mu)*U(3)/d^3-mu*U(3)/r^3];

end
```

Figura 5 – "systemFunction": Código

A partir de esta función del sistema se procede a la integración mediante diferentes esquemas numéricos como son Dormann-Prince 853, y el Adam-Bashford.

3.3.1. Dormann Prince (ODE45)

Este esquema numérico es de tipo Runge-Kutta encajado, consistente en la implementación de múltiples etapas intermedias entre cada paso de integración. A partir del paso temporal en el instante anterior se proporciona una estimación del error en los resultados y se modifica el salto temporal. De esta manera se consigue aumentar la precisión de los resultados.

En MATLAB este tipo de esquema numérico corresponde al integrador ODE45, por lo que mediante Wrappers, es posible el uso del código ya existente en las librerías de MATLAB para la integración numérica.

A esta función es necesario introducirle como input la matriz del sistema, el intervalo de integración y las condiciones iniciales del problema. Además es posible controlar parámetros como la tolerancia para la mejora en los resultados.

```
tspan=[0,100];
InitCond=[L(in,1),L(in,2),L(in,3), 10^-2,0 ,0]';
opt=odeset('RelTol',1e-13,'AbsTol',1e-20);

[~,U_sol]=ode45(@(t,U) systemFunction(t,U,mu),tspan, InitCond,opt);
```

Figura 6 – Llamada a la función ODE45.

Como se puede observar, se introduce en las condiciones iniciales las coordenadas del punto de Lagrange seleccionado por el usuario, así como una pequeña perturbación, en este caso en la dirección x.

3.3.2. Adam-Basforth-Moulton (ODE113)

En este caso el Adam-Bashforth es un método lineal multipaso compuesto por un predictor (Adam-Basforth) y un corrector (Adam-Moulton). Utiliza la información de los N pasos anteriores para la obtención del resultado en N+1.

En MATLAB, este tipo de integrador numérico se corresponde con la función ODE113, que utiliza esquemas de hasta orden 13. Del mismo modo que para el ODE45, esta función requiere de la matriz del sistema, el intervalo de integración y las condiciones iniciales del problema.

```
tspan=[0,100];
InitCond=[L(in,1),L(in,2),L(in,3), 10^-2,0 ,0]';
opt=odeset('RelTol',1e-13,'AbsTol',1e-20);

[t,U_sol]=ode113(@ (t,U) systemFunction(t,U,mu),tspan, InitCond,opt);
```

Figura 7 – Llamada a la función ODE113.

3.3.3. Análisis de resultados

Se procede a la realización del análisis de la trayectoria seguida por un cuerpo situado en cada punto de Lagrange. Cabe destacar que, frente a una tolerancia relativa muy pequeña, ambos esquemas numéricos convergen entorno a la misma solución, por lo que los resultados presentados se han obtenido mediante la integración con ODE45 (Dormann-Prince) para 100 pasos temporales.

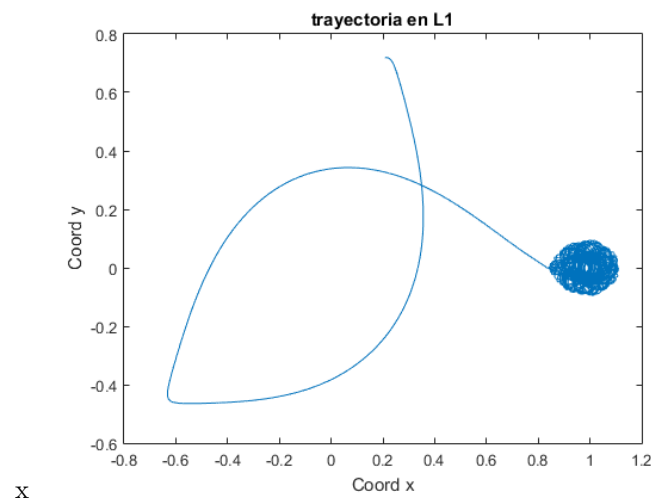


Figura 8 – Trayectoria seguida en el punto L1.

En este caso el cuerpo se desplaza de manera caótica entorno al otro cuerpo para, al final de la simulación salir despedido de dicha trayectoria. Esto se debe a que al no encontrar una solución satisfactoria, la integración se detiene dando lugar a esta trayectoria de salida.

Se verifica la inestabilidad del punto frente a una perturbación del orden de 10^{-2} m/s.

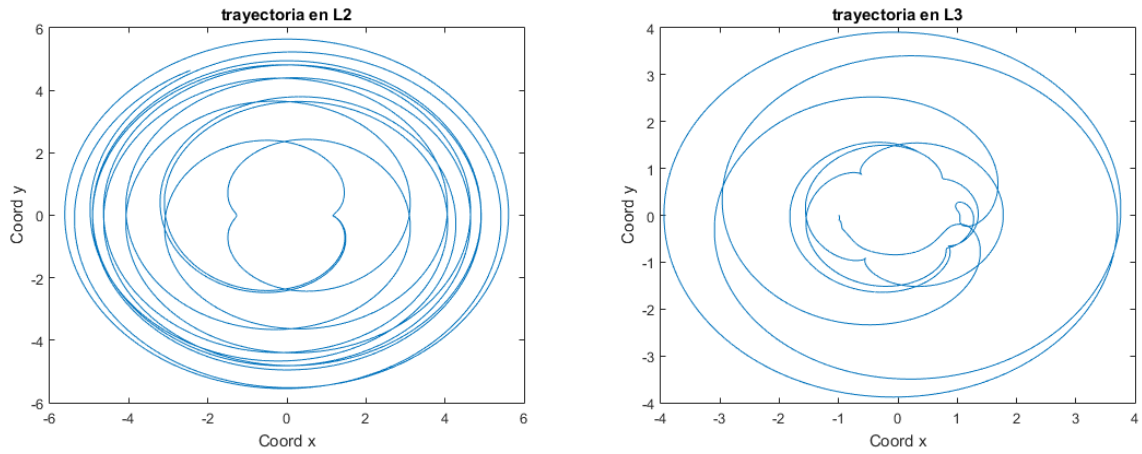


Figura 9 – Trayectoria seguida en los puntos L2 (izq.) y L3 (der.).

De nuevo, el resultado son trayectorias alrededor de un punto central, pero en ningún caso periódicas, tendiendo en ambos casos a irse alejando progresivamente de dicho punto. Por tanto, concluimos que para una perturbación de 10^{-2} m/s, ambos puntos pueden considerarse inestables, tal y como se dedujo en anteriores apartados.

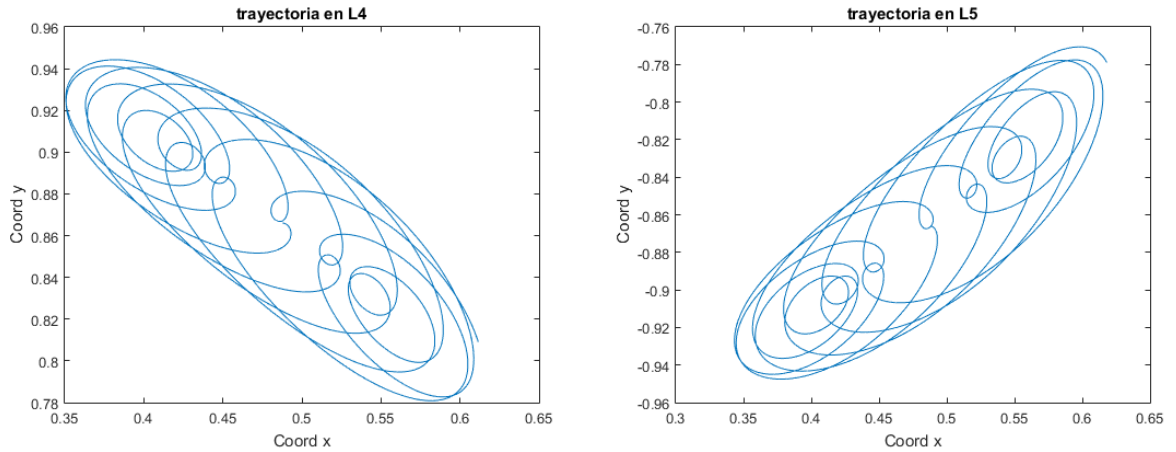


Figura 10 – Trayectoria seguida en los puntos L4 (izq.) y L5 (der.).

En este caso para ambos puntos, los resultados son muy similares debido a su posición simétrica respecto al eje x global del modelo. En ambos casos se aprecia la repetición periódica de la trayectoria. Se puede concluir que ambas soluciones son estables frente a perturbaciones.

4. GMAT

A través de la herramienta GMAT empleando sus integradores se procede de forma análoga a determinar las órbitas de los puntos de Lagrange a partir de una perturbación inicial.

GMAT posee una librería de *Special Points*, donde se encuentran los puntos de Lagrange para distintos cuerpos. Por tanto, se definen estos 5 puntos especiales, seleccionando la Tierra como cuerpo principal y la Luna de secundario. De esta forma ya estarán ubicados en las coordenadas correspondientes.

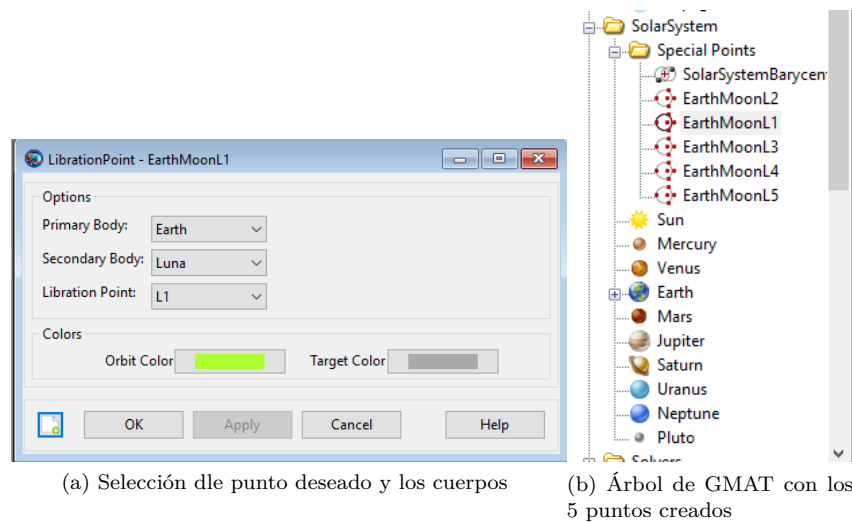


Figura 11 – Librería de puntos especiales de GMAT

Una vez seleccionados se define el sistema de referencia. Se orientan los ejes del mismo modo que el en modelo de MATLAB, con el eje x situado en la recta que une la Tierra y la Luna. El origen del sistema de referencia se ubica en las coordenadas del punto de Lagrange correspondiente.

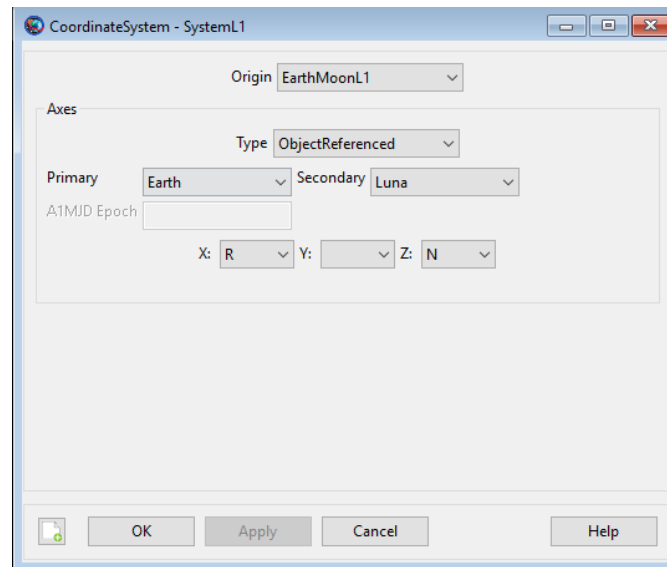


Figura 12 – Sistema de referencia para L1 en GMAT

Finalmente, se crean los satélites correspondientes en cada punto de Lagrange empleado los sistemas de referencia creados anteriormente, induciéndoles una perturbación inicial.

Se tomará la misma perturbación que el modelo de MATLAB, 0.01 m/s en dirección x y se propagará durante 200 días.

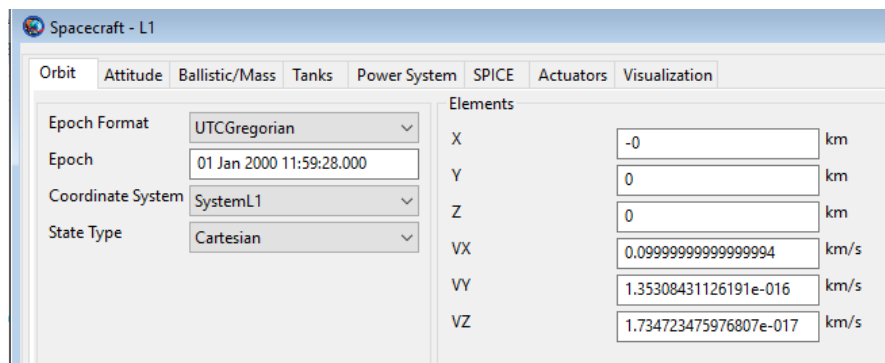


Figura 13 – Parámetros iniciales del satélite en GMAT

Para la comparación con los resultados obtenidos por MATLAB se emplea como propagador *PrinceDormand853* que se corresponde con *ode45* de MATLAB. Se indican como masas pun-

tuales la Tierra y la Luna ya que serán los astros que mayor influencia ejercerán, sin definir la Tierra en profundidad debido a la gran distancia a la misma.

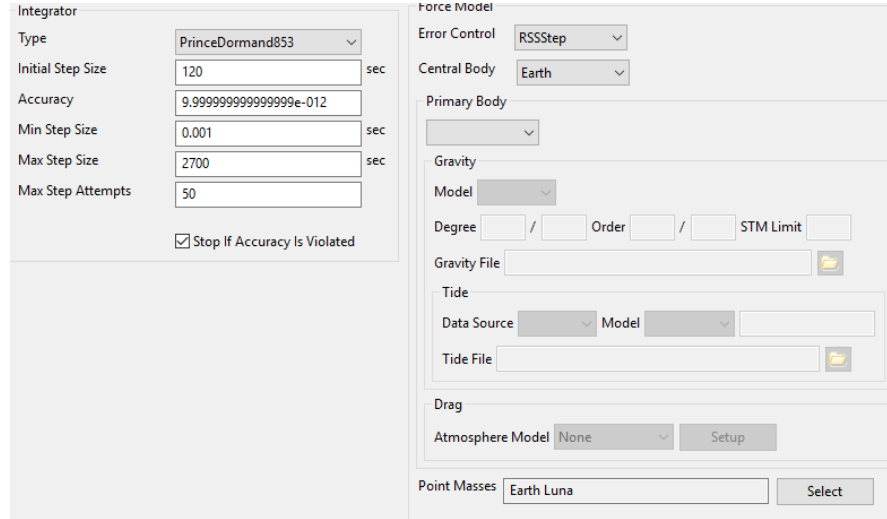


Figura 14 – Propagador *PrinceDormand853* y modelo empleado

4.1. Resultados y comparativa

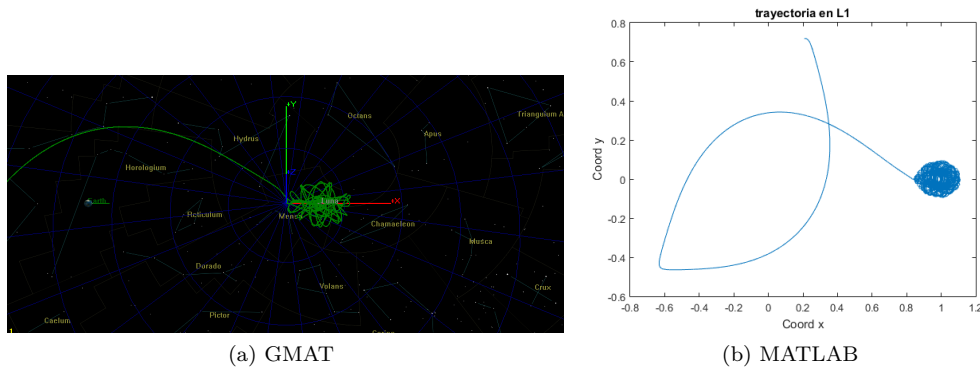


Figura 15 – Punto Lagrange L1

Se puede apreciar la inestabilidad del punto, dado que al pasar cierto tiempo se aleja del punto en cuestión, tras realizar unas órbitas caóticas alrededor de la Luna. Los resultados obtenidos por ambos métodos se puede apreciar que son similares tanto en comportamiento como forma de la órbita.

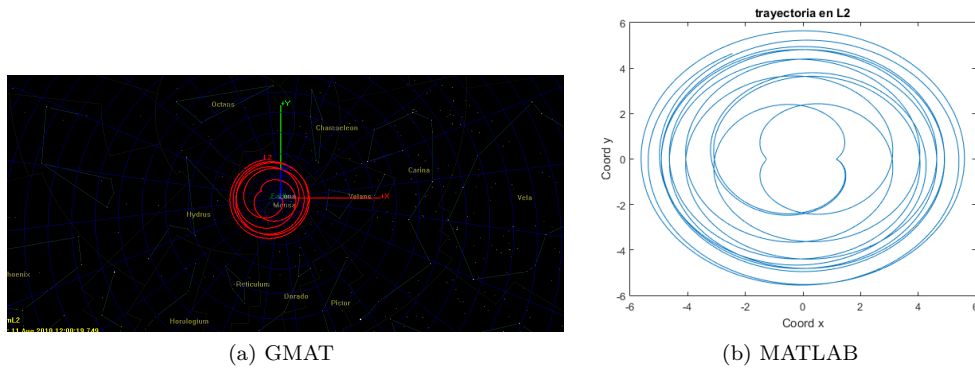


Figura 16 – Punto Lagrange L2

Se observa la inestabilidad de la órbita, dado que esta se aleja cada vez más del punto. Se comprueba que los son similares tanto en comportamiento como forma de la órbita.

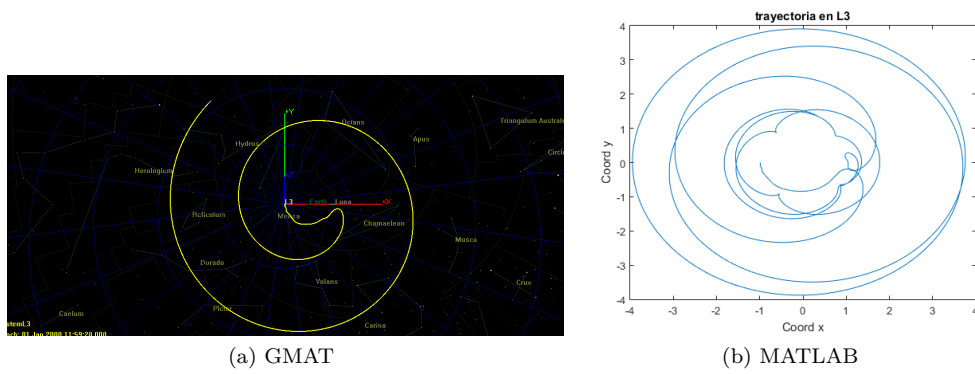


Figura 17 – Punto Lagrange L3

De nuevo la inestabilidad se aprecia en la órbita alejándose indefinidamente del punto. En cuanto los puntos estables:

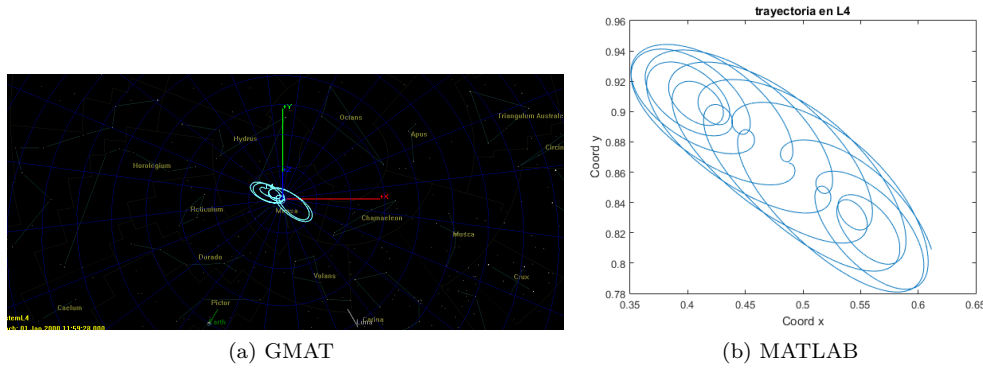


Figura 18 – Punto Lagrange L4

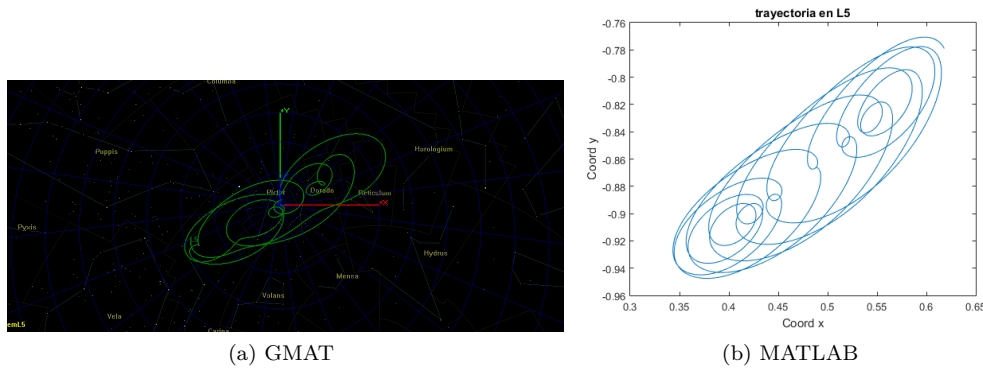


Figura 19 – Punto Lagrange L5

Se aprecia la estabilidad ya que la órbita se repite cíclicamente y no se aleja del punto de origen, por lo que se ve su estado de marginalmente estable.

Cabe destacar que con MATLAB es mas viable simular una mayor cantidad de tiempo, ya que GMAT al aumentar el número de días por encima de 250, la herramienta visual no es capaz de dibujar toda la órbita.

Además, los tiempos de simulación de MATLAB obtenidos para simular una mayor cantidad de tiempo real son mucho menores que en GMAT:

Tabla 3 – Tiempo real de integración en función del método empleado

Punto Lagrange	Tiempo [s]		
	ode45	ode113	GMAT
L1	13.16	3.33	12.17
L2	1.87	0.29	5.21
L3	1.89	0.38	4.78
L4	1.77	0.35	4.11
L5	1.78	0.33	3.81

5. Conclusiones

De todos los análisis anteriormente realizados se desprenden una serie de conclusiones que se presentan a continuación para proporcionar una idea más clara, alguna de las cuales ya ha sido comentada con anterioridad.

- El código utilizado ha permitido obtener los puntos de Lagrange tal y como aparecen en la realidad entre la tierra, el Sol y La Luna.
- Los dos métodos de integración numérica utilizados poseen características diferentes y por tanto para la obtención de resultados validados con ambos métodos ha sido necesario fijar los niveles de tolerancia de la solución.
- En este caso, los valores del tiempo de computación requerido por cada integrador han mostrado que el Dormand-Prince resulta notablemente más lento que el Adam-Basforth.
- Como se puede apreciar en la tabla 3, en general el tiempo de integración que requiere GMAT es muy superior al requerido entorno a unas 4 veces superior al de MATLAB a excepción del punto L1 donde para el caso de *ode45* mejora levemente los tiempos.
- Gracias al uso de dos herramientas software diferentes ha sido posible la verificación de los resultados, así como la validación de la herramienta GMAT para la resolución de cualquier tipo de problema numérico.
- Gracias a esta práctica hemos podido demostrar mediante un caso práctico la veracidad de los teoremas de estabilidad que se han estudiado en la parte teórica de esta asignatura.
- Gracias al aprendizaje en esta asignatura del uso de Wrappers, ha resultado más sencillo la resolución de un problema tan complejo como el problema de los N-Cuerpos.
- En cuanto al uso de herramientas software, esta práctica, así como el desarrollo global de la asignatura, nos ha permitido una mejor organización de los códigos utilizados mediante subrutinas o funciones, así como una visión global de los diferentes lenguajes y entornos de programación.