
AMPLIACIÓN DE MATEMÁTICAS: ENTORNOS DE PROGRAMACIÓN

ENTREGA II: ESQUEMAS NUMÉRICOS DE ALTO ORDEN APLICADOS AL PROBLEMA DE CAUCHY

ALEJANDO FERNÁNDEZ HERRERO
RAÚL MORENO GINÉS
ÁNGEL LUIS PORRAS HERMOSO

*MÁSTER UNIVERSITARIO EN SISTEMAS ESPACIALES
UPM*



ENTREGA II
18 DE MARZO DE 2019

ÍNDICE

1	INTRODUCCIÓN Y OBJETIVOS	3
2	PUNTOS DE LAGRANGE	3
2.1	Modelización matemática	3
2.2	Obtención de puntos críticos y análisis de estabilidad	5
3	INTEGRACIÓN NUMÉRICA DEL PROBLEMA DE CAUCHY	5
4	COMPARACIÓN DE LOS RESULTADOS CON GMAT	13
5	SOFTWARE EMPLEADO	15
6	CONCLUSIONES	16

Lista de tablas

1	Puntos críticos del sistema y la parte real mayor del autovalor asociado. Se puede observar que existen tres puntos inestables (L1, L2 y L3) y dos puntos marginalmente estables (L4 y L5)	5
2	Tiempo de CPU	10

1 INTRODUCCIÓN Y OBJETIVOS

En trabajos anteriores se ha desarrollado Software numérico en *Fortran 90* para resolver el problema de Cauchy mediante integradores numéricos diferenciales de bajo orden. En concreto, se ha integrado el problema de los N-cuerpos con esquemas numéricos de tipo Euler (explícito e implícito) y Runge-Kutta (de orden 2 y 4). Sin embargo, problemas de mecánica orbital más complejos necesitan de integradores temporales más sofisticados para poder modelizar el comportamiento del sistema.

En este trabajo se estudiará el comportamiento de un cuerpo alrededor de los puntos de Lagrange entre la Tierra y la Luna. Se comenzará planteando las ecuaciones que rigen el movimiento del cuerpo en los puntos de Lagrange. Posteriormente, se obtendrán los *ceros* del sistema con el objetivo de determinar si el sistema es estable o inestable, alrededor de los puntos de Lagrange, mediante el cálculo de los autovalores asociados a cada punto de Lagrange. Una vez obtenidos y caracterizados todos los puntos de Lagrange se estudiará el comportamiento de un cuerpo sometido a la acción de la Tierra y de la Luna en el entorno de cada uno de estos puntos de Lagrange. En definitiva, se calculará el problema de los 3-cuerpos imponiendo como condiciones iniciales del cuerpo que modeliza al satélite la de los puntos de Lagrange introduciendo una pequeña perturbación.

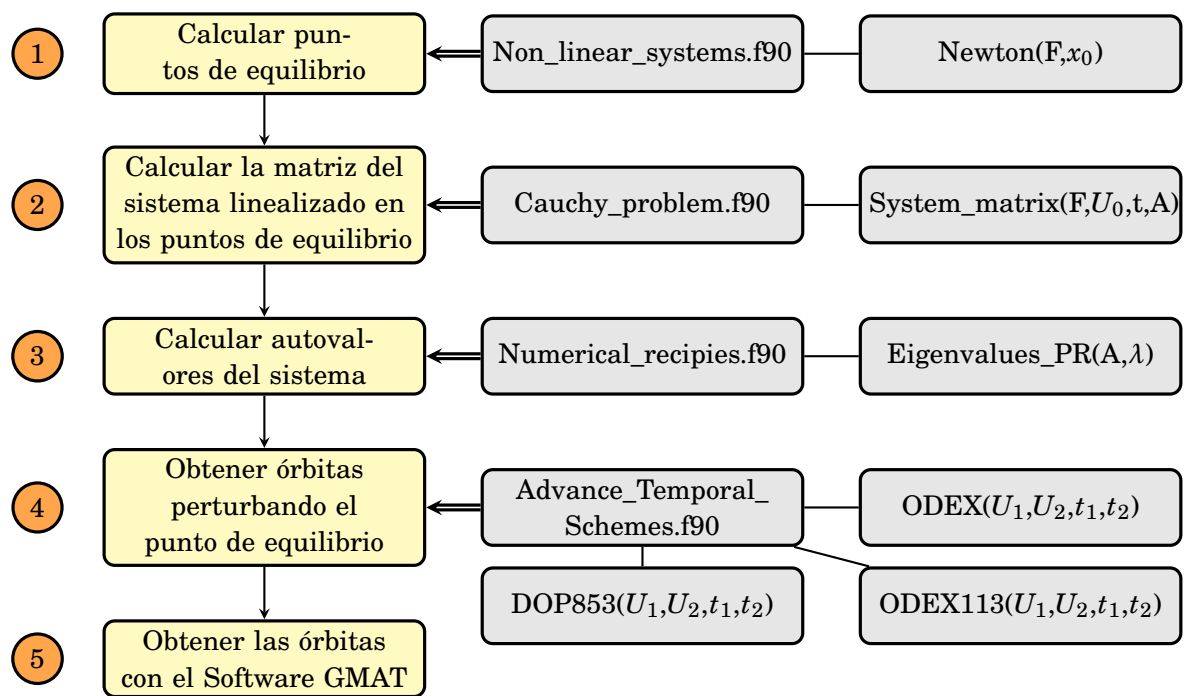


Figura 1: Fases del trabajo

2 PUNTOS DE LAGRANGE

2.1 Modelización matemática

Los puntos de Lagrange son las cinco posiciones en un sistema orbital donde un satélite (o un objeto de masa despreciable frente al resto de cuerpos) sometido a la fuerza gravitatoria de otros dos cuerpos masivos, puede permanecer estacionario, teóricamente, con respecto al cuerpo de menor masa.

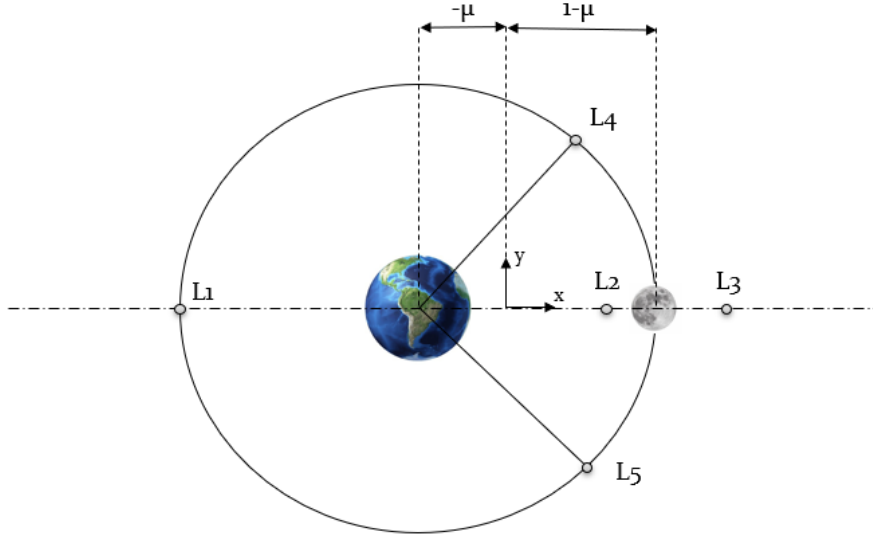


Figura 2: Puntos de Lagrange entre la Tierra y la Luna.

Las ecuaciones del movimiento del problema de los tres cuerpos restringido se pueden expresar de la siguiente forma:

$$\ddot{x} = 2\dot{y} + x - \frac{(1-\mu)(x+\mu)}{d^3} - \frac{\mu(x-1+\mu)}{r^3} \quad (1)$$

$$\ddot{y} = -2\dot{x} + y - \frac{(1-\mu)y}{d^3} - \frac{\mu y}{r^3} \quad (2)$$

$$\ddot{z} = -\frac{(1-\mu)z}{d^3} - \frac{\mu z}{r^3} \quad (3)$$

donde han sido adimensionalizadas con el parámetro μ , que se expresa como $\mu = \frac{m_{Luna}}{m_{Tierra} + m_{Luna}}$. El parámetro d se expresa como $d = (x + \mu)^2 + y^2 + z^2$, y el parámetro r es $r^2 = (x - 1 + \mu)^2 + y^2 + z^2$.

El sistema 1- 3, de segundo orden, se puede reducir a un sistema de primer orden, con el objetivo de resolverlo numéricamente, de la forma:

$$\frac{d\mathbf{U}}{dt} = \mathbf{F}(\mathbf{U}, t), \quad (4)$$

donde \mathbf{U} y \mathbf{F} se pueden expresar de la siguiente forma:

$$\mathbf{U} = \begin{Bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{Bmatrix}; \quad \mathbf{F} = \begin{Bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{Bmatrix}. \quad (5)$$

Las componentes del vector $\mathbf{F}(\mathbf{U}, t)$ se pueden expresar como:

$$F(1) = U(4) \quad (6)$$

$$F(2) = U(5) \quad (7)$$

$$F(3) = U(6) \quad (8)$$

$$F(4) = 2 \cdot U(5) + U(1) - \frac{(1-\mu)(U(1)+\mu)}{d^3} - \frac{\mu(U(1)-1+\mu)}{r^3} \quad (9)$$

$$F(5) = -2U(4) + U(2) - \frac{(1-\mu)U(2)}{d^3} - \frac{\mu U(2)}{r^3} \quad (10)$$

$$F(6) = -\frac{(1-\mu)U(3)}{d^3} - \frac{\mu U(3)}{r^3} \quad (11)$$

2.2 Obtención de puntos críticos y análisis de estabilidad

El primer paso será calcular los puntos críticos del sistema 1- 3. Los *ceros* del sistema se calculará usando las librerías que se incluyen en el siguiente enlace <https://github.com/jahrWork/NumericalHUB>. En este enlace se encuentran una serie de proyectos .f90, entre los cuales se puede encontrar Software relativo a resolver sistemas de ecuaciones mediante el método de Newton, obtención de la matriz del sistema, problemas de autovalores y autovectores, resolvedores del problema de Cauchy, etc.

Los puntos críticos del sistema 1- 3, obtenidos mediante el método de Newton se muestran a continuación:

Tabla 1: Puntos críticos del sistema y la parte real mayor del autovalor asociado. Se puede observar que existen tres puntos inestables (L1, L2 y L3) y dos puntos marginalmente estables (L4 y L5)

	x	y	z	Re(λ)
L1	-1,005	0,000	0,000	0,178
L2	0,837	0,000	0,000	2,932
L3	1,156	0,000	0,000	2,159
L4	0,488	0.866	0,000	2,05E-6
L5	0,488	-0,866	0,000	2,84E-6

3 INTEGRACIÓN NUMÉRICA DEL PROBLEMA DE CAUCHY

Como se ha comentado anteriormente, para resolver problemas complejos de dinámica orbital (problema de Cauchy) es necesario la integración de las ecuaciones de movimiento mediante esquemas numéricos sofisticados. En este caso se van a utilizar tres esquemas temporales distintos: Dormand-Prince (DOPRI853), Adams-Bashford-Moulton (ODE113) y Gregg-Bulisch-Steer (ODEX). A continuación se hará un breve resumen de las características más representativas de cada uno de ellos:

- **DOPRI853:** este esquema numérico pertenece a la familia de Runge-kutta empotrados, y utiliza un procedimiento multietapa, en el que, en cada paso de tiempo se evalúa la función varias veces (en concreto 6 veces por cada paso), con el objetivo de obtener la solución de cuarto y quinto orden del sistema. La diferencia entre ambas soluciones se toma como el error de la solución de cuarto orden. El objetivo del integrador es adaptar el paso temporal en cada iteración que este error no supere un límite concreto.

El código software que se ha utilizado relativo a este integrador fue desarrollado por E. Hairer y G. Wanner del departamento de matemáticas de la universidad de Génova.

- **ODE113**: es un esquema multipaso que halla el error de la solución numérica como la diferencia entre lo que se denomina predictor y corrector con el objetivo de validar el paso temporal.
- **ODEX**: es un método de integración numérica simpléctico (conserva la energía mecánica del problema), mediante el cual se va estimando el error mediante la creación de varias etapas (tantas como sean necesarias según el problema estudiado). Para calcular la etapa posterior se utiliza el método denominado Leap Frog.

El código software que se ha utilizado relativo a este integrador fue desarrollado por E. Hairer y G. Wanner del departamento de matemáticas de la universidad de Génova.

Para adaptar el software relativo a los esquemas numéricos a las necesidades del problema de la integración de las ecuaciones del movimiento de un cuerpo alrededor de los puntos de Lagrange, ha sido necesario la creación de los denominados *Wrappers*. El objetivo de estos *Wrappers* es que los distintos integradores sean capaces de operar con una serie de argumentos que se definen en nuestro problema, independientemente de los argumentos originales que presenta cada integrador. Específicamente, se quiere que los integradores trabajen con los argumentos U_1 (vector de estado en el instante actual), U_2 (vector de estado en el instante posterior), t_1 y t_2 (tiempo actual y tiempo en el instante posterior respectivamente). En definitiva, el objetivo de este *envoltorio* o *Wrapper* es que, independientemente de los argumentos de las subrutinas de los esquemas numéricos (que vienen dados por el proveedor del Software específico que se necesite), se introduzca como argumento el vector de estado en un instante t_1 , y el propio instante de tiempo t_1 , y se obtenga como *output* el vector de estado en el instante t_2 , sin tener en cuenta explícitamente los argumentos internos que intervienen en el esquema numérico. En definitiva, un *Wrapper* no es más que una entidad que trata de encapsular una parte del código con el objetivo de ocultar su complejidad, mediante el uso de interfaces.

Aunque se ha comentado el esquema numérico ODE113, este esquema no se verá reflejado a lo largo de los resultados obtenidos, aunque se haya desarrollado un *Wrapper* acorde a las necesidades del esquema numérico, esto es debido a que existe un error de compilación del propio archivo que contiene el esquema ODE113. Debido a que este trabajo está relacionado con la creación de *Wrapper*, no de solucionar errores de programas ajenos, y a la falta de conocimiento y medios para subsanar el problema se ha dejado comentado la subrutina que contiene el *Wrapper* del ODE113.

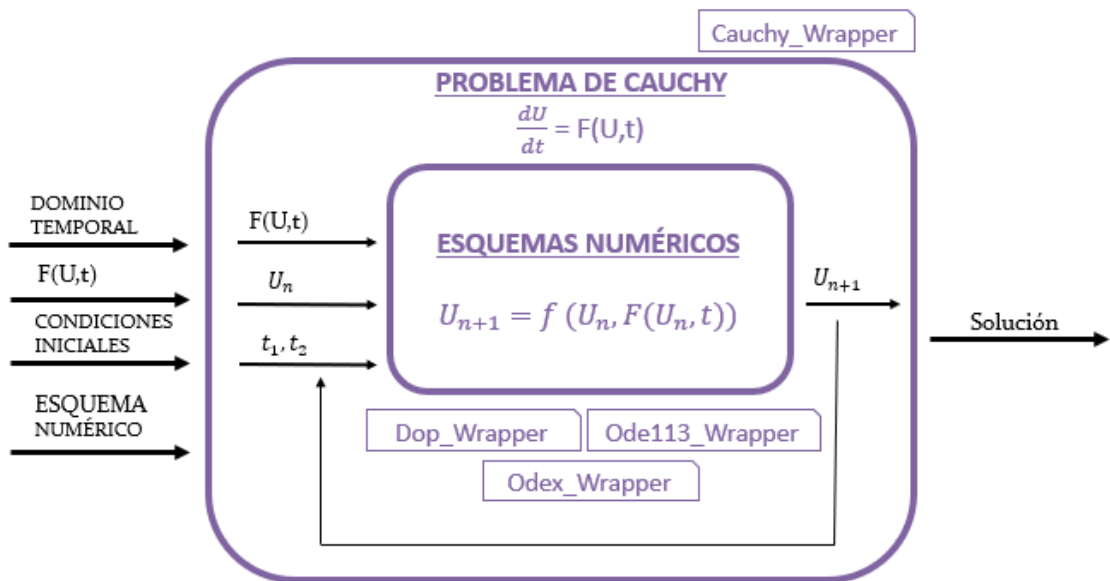


Figura 3: Esquema del envoltorio del problema de Cauchy.

A continuación se representan las órbitas entorno a los puntos de Lagrange calculados anteriormente, perturbando estos puntos un épsilon $\epsilon = 1 \cdot 10^{-3}$. Se resolverá el sistema de ecuaciones diferenciales 1- 3, imponiendo como condición inicial $U_0 + \epsilon$. La tolerancia que se ha utilizado para validar el paso temporal es del 10^{-6} .

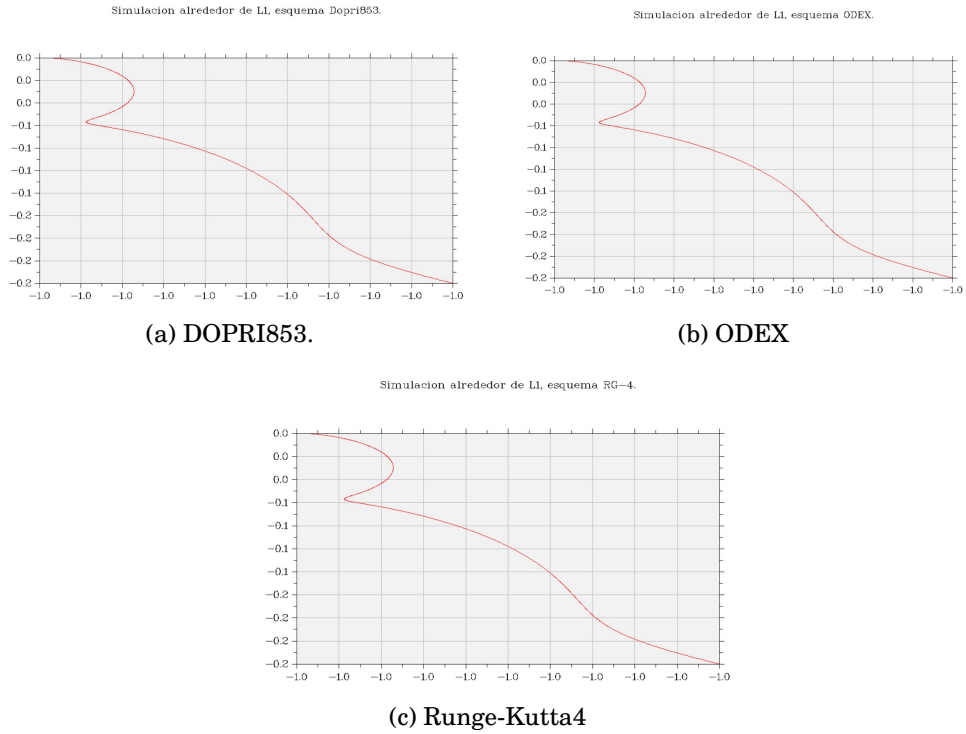


Figura 4: órbita alrededor del punto L1 con el esquema temporal DORPI853, ODEX y Runge-Kutta 4 para un tiempo final de integración de 4π .

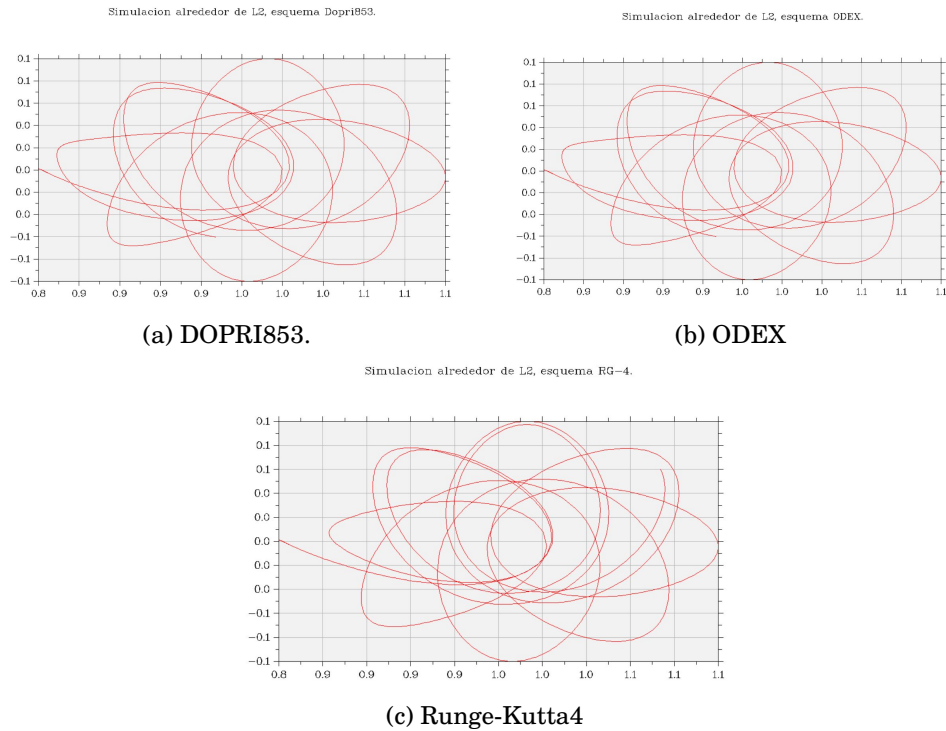


Figura 5: órbita alrededor del punto L2 con el esquema temporal DORPI853, ODEX y Runge-Kutta 4 para un tiempo final de integración de 4π .

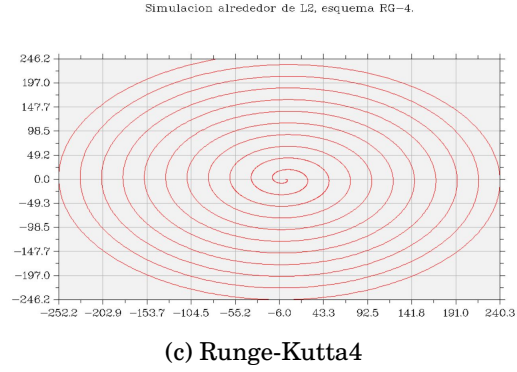
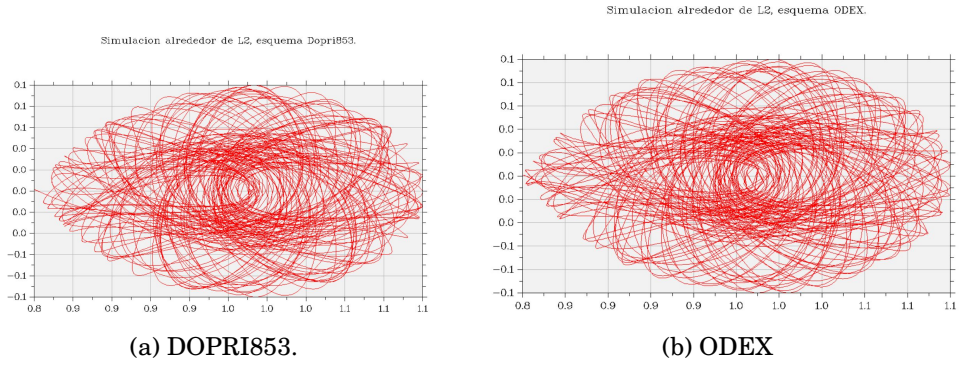


Figura 6: órbita alrededor del punto L2 con el esquema temporal DORPI853, ODEX y Runge-Kutta 4 para un tiempo final de integración de 50π .

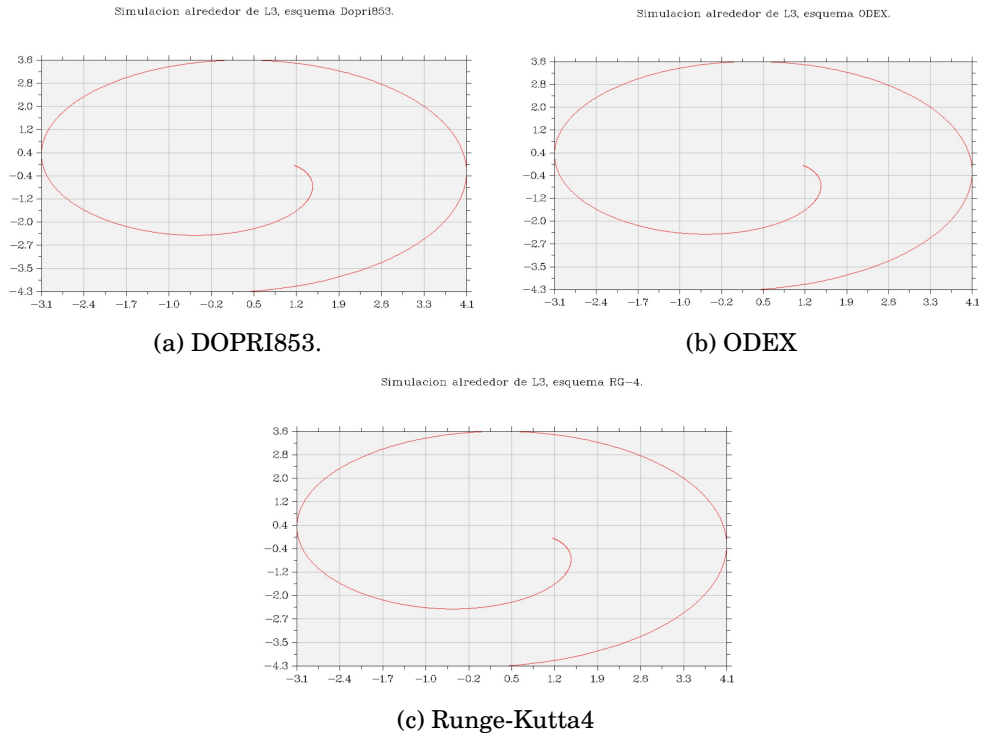


Figura 7: órbita alrededor del punto L3 con el esquema temporal DORPI853, ODEX y Runge-Kutta 4 para un tiempo final de integración de 4π .

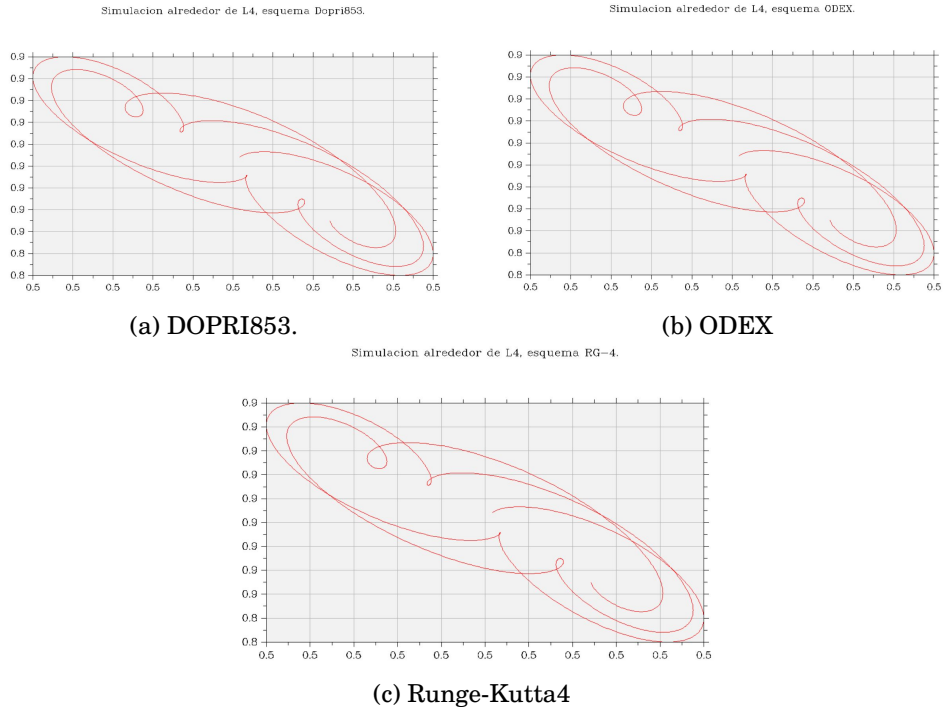


Figura 8: órbita alrededor del punto L4 con el esquema temporal DOPRI853, ODEX y Runge-Kutta 4 para un tiempo final de integración de 15π .

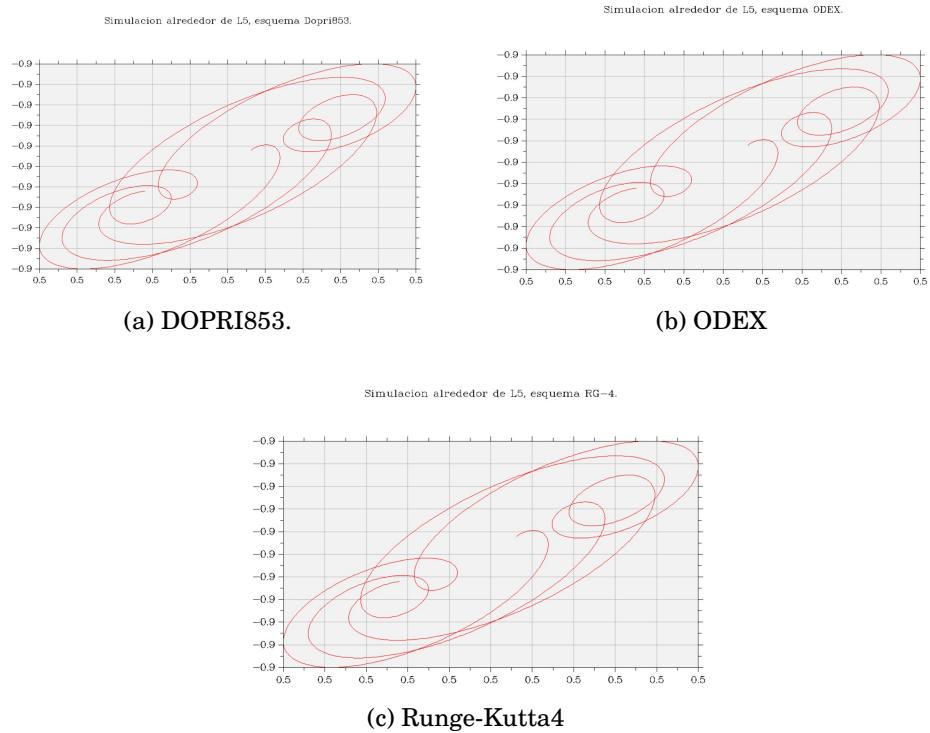


Figura 9: órbita alrededor del punto L5 con el esquema temporal DOPRI853, ODEX y Runge-Kutta 4 para un tiempo final de integración de 15π .

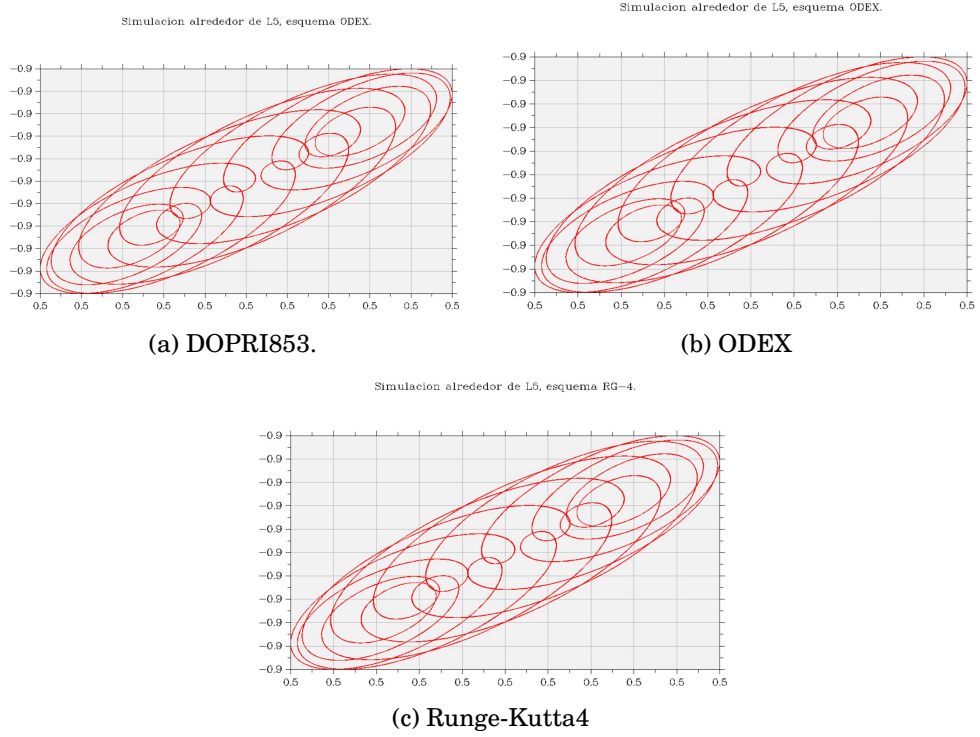


Figura 10: órbita alrededor del punto L5 con el esquema temporal DOPRI853, ODEX y Runge-Kutta 4 para un tiempo final de integración de 300π .

A continuación se muestran los tiempos de CPU que se tarda en obtener el último paso:

Tabla 2: Tiempo de CPU

	$t_{CPU}(s)$
DOPRI estables ($t_f = 4\pi$)	0,016
DOPRI estables ($t_f = 300\pi$)	0,094
DOPRI inestables	0,109
ODEX estables ($t_f = 4\pi$)	0,016
ODEX estables ($t_f = 300\pi$)	0,094
ODEX inestables	0,125
RK4 estables ($t_f = 300\pi$)	0,016

Por último se incluirá una representación del error entre la solución (posición) obtenida mediante el esquema *Odex* y el *DOPRI853*.

error entre esquema Odex y Dopri853, para L1.

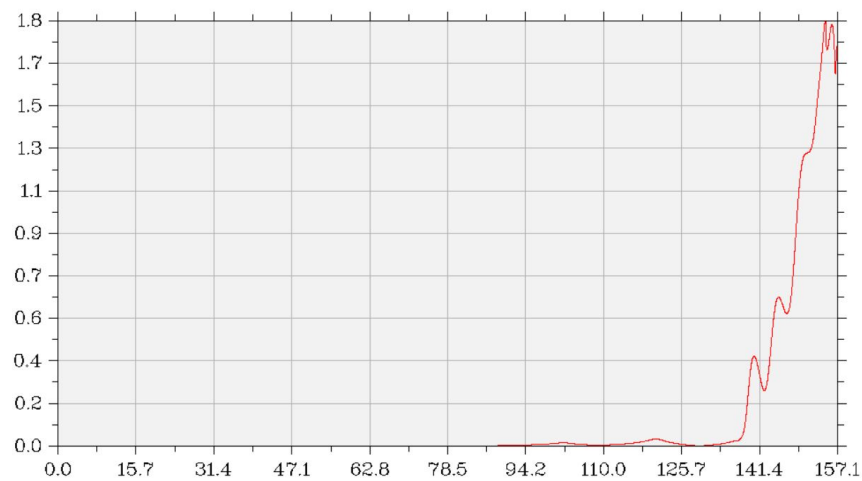


Figura 11: Error entre ambos esquemas numéricos. Se ha representado en el eje vertical la distancia entre la posición obtenida con el *Odex* y la obtenida con el *DOPRI853* en función del tiempo, para el punto L1.

error entre esquema Odex y Dopri853, para L2.

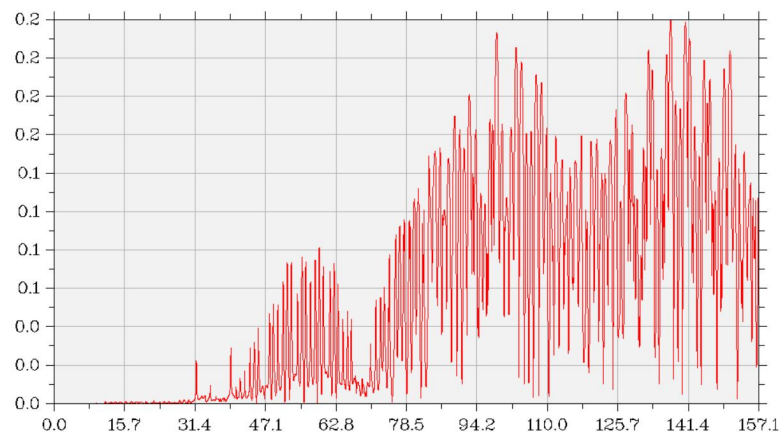


Figura 12: Error entre ambos esquemas numéricos. Se ha representado en el eje vertical la distancia entre la posición obtenida con el *Odex* y la obtenida con el *DOPRI853* en función del tiempo, para el punto L2.

error entre esquema Odex y Dopri853, para L3.

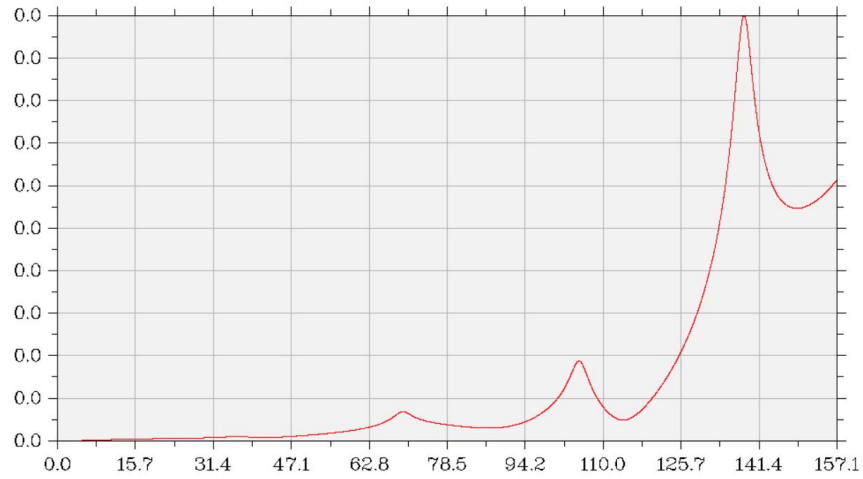


Figura 13: Error entre ambos esquemas numéricos. Se ha representado el el eje vertical la distancia entre la posición obtenida con el *Odex* y la obtenida con el *DOPRI853* en función del tiempo para el punto L3.

error entre esquema Odex y Dopri853, para L4.

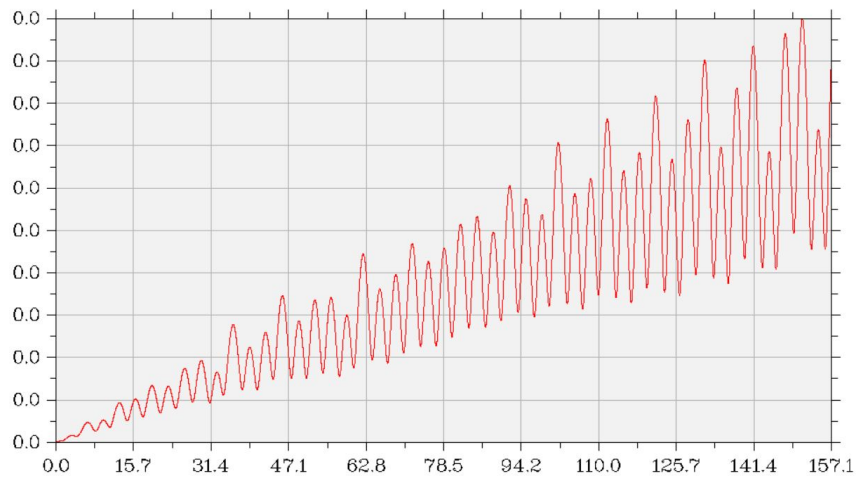


Figura 14: Error entre ambos esquemas numéricos. Se ha representado el el eje vertical la distancia entre la posición obtenida con el *Odex* y la obtenida con el *DOPRI853* en función del tiempo para el punto L4.

error entre esquema Odex y Dopri853, para L5.

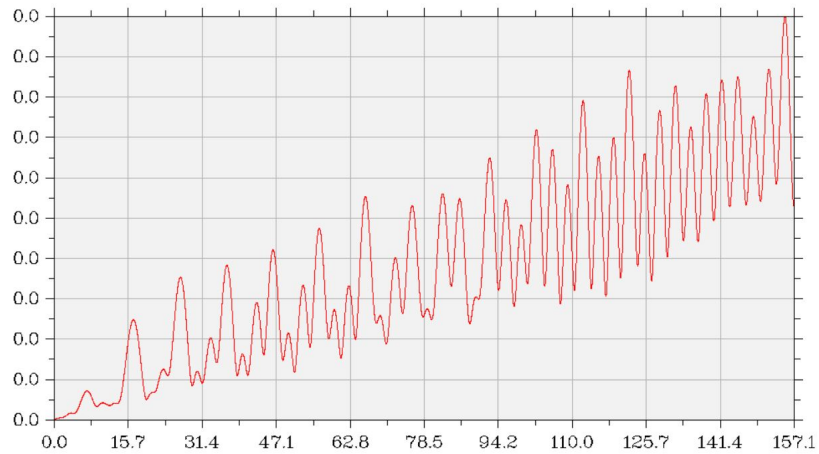


Figura 15: Error entre ambos esquemas numéricos. Se ha representado en el eje vertical la distancia entre la posición obtenida con el *Odex* y la obtenida con el *DOPRI853* en función del tiempo para el punto L5.

4 COMPARACIÓN DE LOS RESULTADOS CON GMAT

GMAT es un software especializado en el cálculo de órbitas mediante la integración numérica del problema de los N-cuerpos con esquemas de alto orden. En concreto, se pueden utilizar esquemas temporales del tipo *PrinceDormad78*, *Adams Basgforth Moulton*, *Runge-Kuttas* de alto orden, etc.

Se quieren obtener órbitas de satélites alrededor de los puntos de Lagrange cuando se perturban los puntos críticos del sistema 1- 3. A continuación se adjuntan las órbitas obtenidas propagando durante 100 días:

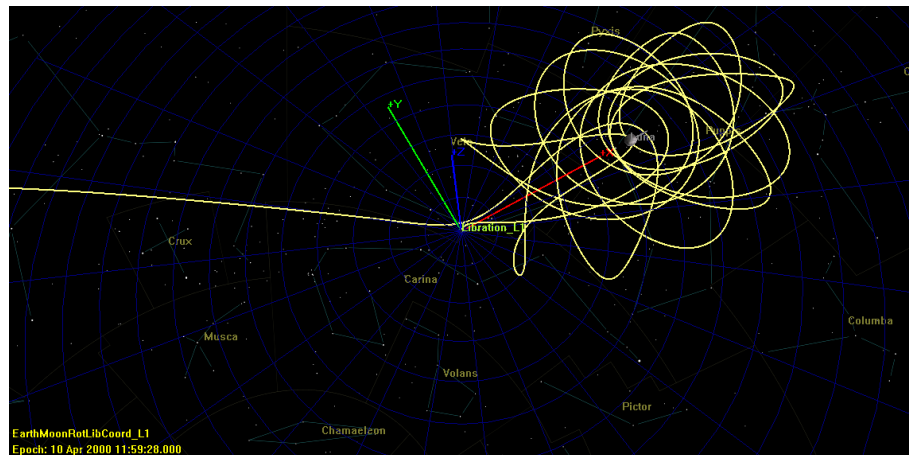


Figura 16: Punto de Lagrange L1.

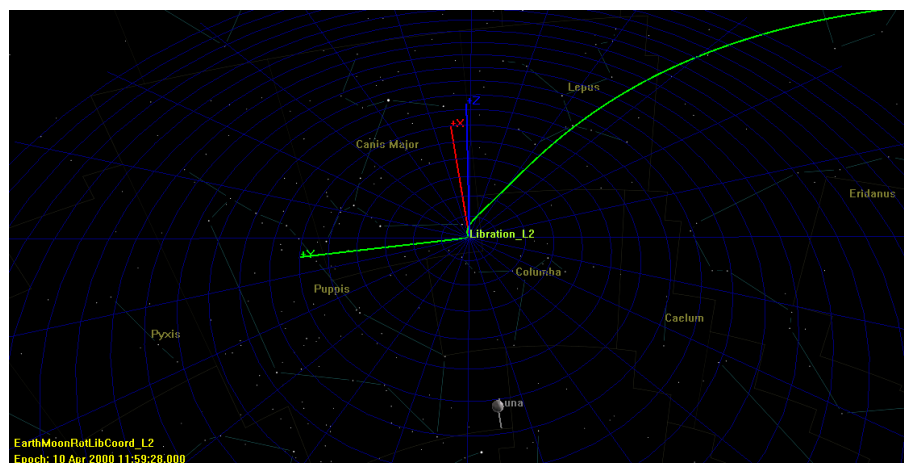


Figura 17: Punto de Lagrange L2.

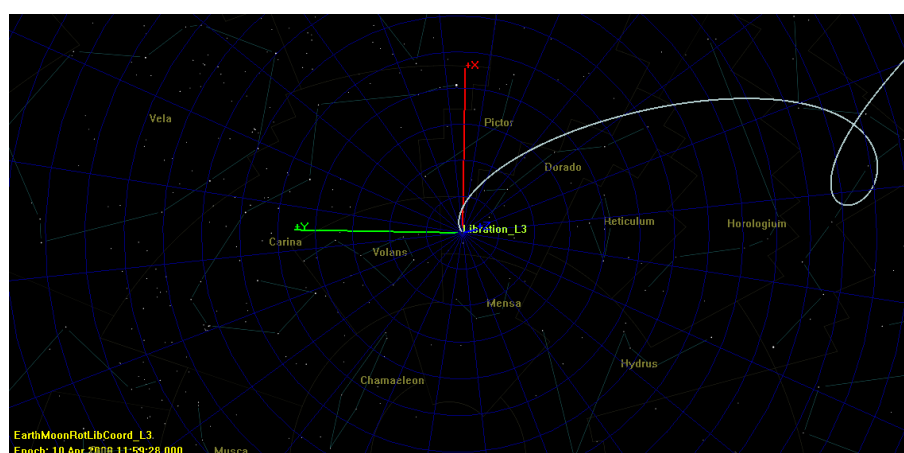


Figura 18: Punto de Lagrange L3.

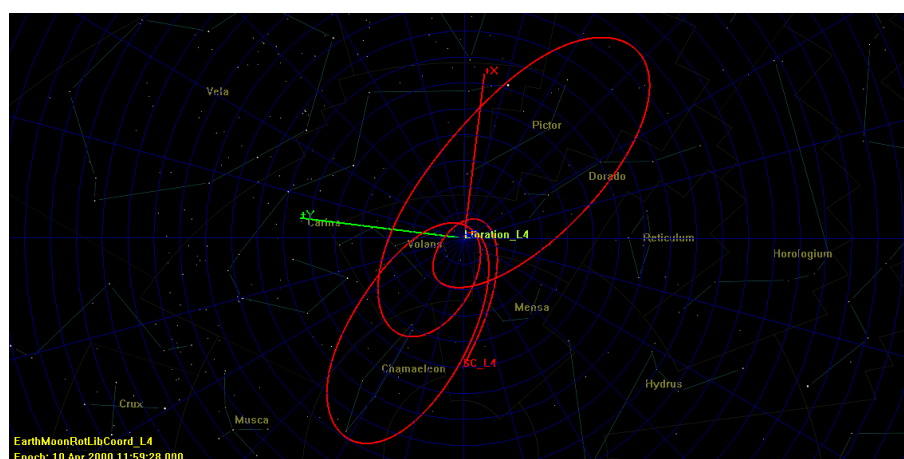


Figura 19: Punto de Lagrange L4.

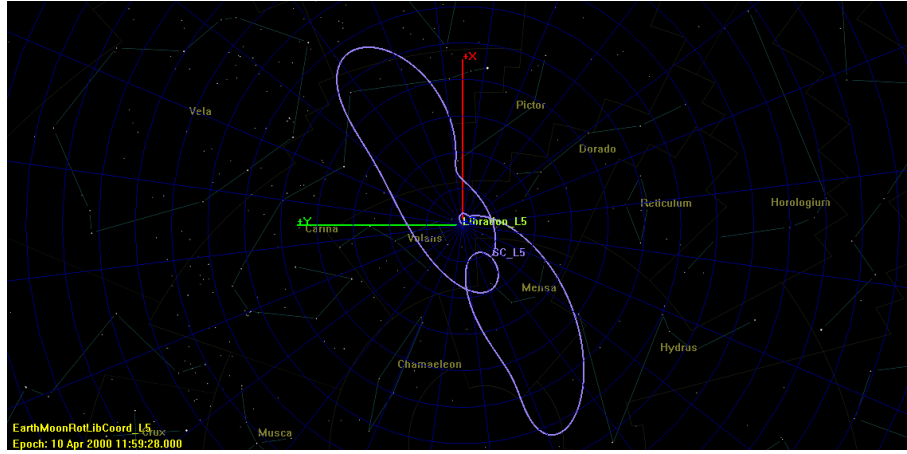


Figura 20: Punto de Lagrange L5.

5 SOFTWARE EMPLEADO

En esta sección se hará un resumen del Software empleado.

- **Lagrange_Points:** Este archivo se usa como programa principal y en él se llama a las siguientes subrutinas:
 - **Subrutina *Newton*** que resuelve un sistema de ecuaciones no lineales. Como argumentos se debe meter la función F del sistema a resolver (en este caso las ecuaciones del movimiento del problema de los tres cuerpos restringido a órbitas circulares), y una semilla inicial.

```
call Newton(F=Lagrange_mod, x0=U_mod(i,:))
```

Figura 21: Llamada a la subrutina *Newton*.

- **Subrutina *System_matrix*** para obtener la matriz linealizada de un sistema. Se obtendrá la matriz del sistema del problema de los tres cuerpos restringido linealizada respecto a los puntos críticos del sistema. Como argumentos se ha de introducir la función F del sistema, el punto donde se quiere linealizar y el instante de tiempo donde se quiere linealizar. Como salida se devuelve la matriz A linealizada del sistema.

```
call System_matrix( F = Lagrange, U0 = U_matrix(i,:),  
                  t = t(0), A = Liniar_Matrix)
```

Figura 22: Llamada a la subrutina *System_Matrix*.

- **Subrutina *Eigenvalues_QR*** que obtiene los autovalores de la matriz del sistema. Como argumento se introduce la matriz, y como resultado se devuelven los autovalores de la matriz.

```
call Eigenvalues_QR( A = Liniar_Matrix, lambda = lambda)
```

Figura 23: Llamada a la subrutina *Eigenvalues_QR*.

- **Subrutina *Cauchy_Wrapper*** que resuelve el problema de Cauchy. Como argumentos se introducen el dominio temporal, el sistema de ecuaciones que se quiera integrar (F) y

el esquema numérico que se quiera emplear. Como resultado se devuelve el vector de estado en el tiempo final.

```
call Cauchy_wrapper(Time_Domain=t, Differential_operator=Lagrange,
                    Scheme= Dop_Wrap, Solution=U)
```

Figura 24: Llamada a la subrutina *problemaCauchy*.

- **Advanced_Cauchy_problem:** En este módulo se hacen todos los wrappers de los integradores y las llamadas a éstos. Además, se añade la subrutina F_Call que llama a la función F del sistema.
- **Advance_Schemes:** contiene los integradores diferenciales utilizados (DOPRI853, ODEX Y ODE).
- **Function_Lagrange_mod:** Se define el sistema de ecuaciones diferenciales asociado a problema de los tres cuerpos restringido (la función $F(U)$).

6 CONCLUSIONES

- A excepción del punto L1, los esquemas *Odex* y *Dopri* obtienen **soluciones parecidas** para los casos estudiados, por lo que ambos integradores resultan **válidos** para el estudio de perturbaciones alrededor de los puntos de Lagrange.
- Para **tiempos cortos**, los tres esquemas estudiados (*DOPRI*, *ODEX* y *Runge-Kutta4*) obtienen soluciones parecidas. Es para **tiempos largos** cuando el esquema Runge-Kutta4 empieza a diverger con respecto a los otros dos, fenómeno que se muestra más acusado para los puntos inestables. Por tanto, para integrar problemas de mecánica orbital durante tiempos largos es conveniente usar esquemas numéricos de orden mayor.
- Se ha observado que el esquema *DOPRI*, cuando integra largos intervalos de tiempo es **más rápido computacionalmente** que el *ODEX*. Para tiempos cortos no hay prácticamente diferencia.
- Se ha observado la utilidad de hacer envolturas de código (**Wrappers**), de forma que para resolver cualquier problema de Cauchy se introduzcan los **mismos argumentos** de entrada y se obtengan los mismos argumentos de salida. De esta forma, para cada problema sólo es necesario especificar el dominio temporal donde se quiera integrar, la función $F(U)$ relativa al problema y el esquema que se quiera utilizar (*ODEX*, *DOPRI* o *Runge-Kutta4* en este caso), obteniéndose el vector de estado en el tiempo final.

Referencias

- [1] Juan Antonio Hernández Ramos, Belén Moreno Santamaría, Imanol Sardón Delgado y Miguel ángel Rapado Tamarit. *Programming with Visual Studio: Fortran & Python*. Department of Applied Mathematics School of Aeronautical and Space Engineering Technical University of Madrid (UPM), 2018.
- [2] Juan A. Hernández Ramos y Javier Escoto López. *How to learn Applied Mathematics through modern FORTRAN*. Department of Applied Mathematics School of Aeronautical and Space Engineering Technical University of Madrid (UPM), 2018.