

ENTREGA 1: ANÁLISIS NUMÉRICO DE UN CAMBIO DE ÓRBITA DE UN SATÉLITE

Alejandro Fernández Herrero Ángel Luis Porras Hermoso
Raúl Moreno Ginés

January 30, 2019

Abstract

En este informe se recogen aquellos trabajos e hitos que se han planteado durante el primer semestre de 1º de MUSE de 2018. Se plantean los problemas y ejercicios propuestos así como la metodología seguida para el diseño de software de los programas implicados. Adicionalmente se plantea estudiar el impacto que sufre en la órbita de diseño un satélite al que le falle el control de apagado del motor cohete eléctrico, para analizar el riesgo e impacto que este tipo de fallo tendría en una hipotética misión.

Contents

1	Introducción	1
2	Modelos numéricos	2
3	Metodologías y herramientas de programación	3
4	Ecuaciones de Bessel	5
5	Propagador Armónico	7
5.1	Resultado de la Simulación	7
5.2	Conclusiones obtenidas de la simulación	8
6	N cuerpos	9
6.1	Sistema Tierra-Luna-Satélite	9
6.2	Sistema Solar	20
7	Cambio de órbita de un satélite	26
7.1	Simulación	26
7.2	Conclusión	29
8	Conclusiones	30

List of Figures

1	Esquema de los pasos a seguir en el diseño software	3
2	órbita de una ecuación de Bessel	5
3	Función de Bessel en función del tiempo	6
4	Órbita del oscilador armónico. Resultado de la simulación usando un propagador tipo Euler explícito	7
5	Órbita del oscilador armónico. Resultado de la Simulación utilizando un esquema numérico tipo Rungue-Kutta 4	8
6	Condiciones iniciales bajo las que se ha simulado el sistema	10
7	Resultado de la Simulación utilizando un esquema numérico tipo Rungue-Kutta 4 para 28 días de simulación. Los valores se han dividido por 10^6	11
8	Resultado de la Simulación utilizando un esquema numérico tipo Rungue-Kutta 2 para 28 días de simulación. Los valores se han dividido por 10^6	12
9	Resultado de la Simulación utilizando un esquema numérico tipo Euler para 28 días de simulación. Los valores se han dividido por 10^6	13
10	Resultado de la Simulación utilizando un esquema numérico tipo Rungue-Kutta 4 para un año de simulación. Los valores se han dividido por 10^6	14
11	Resultado de la Simulación utilizando un esquema numérico tipo Rungue-Kutta 2 para un año de simulación. Los valores se han dividido por 10^6	15
12	Resultado de la Simulación utilizando un esquema numérico tipo Euler para un año de simulación. Los valores se han dividido por 10^6	16
13	Intersección de la órbita en el plano de Poincaré variando las condiciones iniciales:1	17
14	Intersección de la órbita en el plano de Poincaré variando las condiciones iniciales:2	17
15	Intersección de la órbita en el plano de Poincaré variando las condiciones iniciales:3	18
16	Intersección de la órbita en el plano de Poincaré aumentado la cantidad de periodos simulados	18
17	Intersección de la órbita en el plano de Poincaré aumentado la cantidad de periodos simulados	19
18	Condiciones iniciales bajo las que se ha simulado el sistema	20
19	Resultado de la Simulación utilizando un esquema numérico tipo Runge-Kutta 4 para 10 años de simulación. Los valores se han dividido por 10^6	20

20	Resultado de la Simulación utilizando un esquema numérico tipo Rungue-Kutta 2 para 10 años de simulación. Los valores se han dividido por 10^6	21
21	Resultado de la Simulación utilizando un esquema numérico tipo Euler para 10 años de simulación. Los valores se han dividido por 10^6	22
22	Resultado de la Simulación utilizando un esquema numérico tipo Rungue-Kutta 4 para 50 años de simulación. Los valores se han dividido por 10^6	23
23	Resultado de la Simulación utilizando un esquema numérico tipo Rungue-Kutta 2 para 50 años de simulación. Los valores se han dividido por 10^6	24
24	Resultado de la Simulación utilizando un esquema numérico tipo Euler para 50 años de simulación. Los valores se han dividido por 10^6	24
25	Resultado de la Simulación con motor parado (condiciones iniciales)	26
26	Resultado de la Simulación con un empuje máximo de 0.001 N	27
27	Resultado de la Simulación con un empuje máximo de 0.01 N	27
28	Resultado de la Simulación con un empuje máximo de 0.1 N	28
29	Resultado de la Simulación con un empuje máximo de 0.5 N	28
30	Resultado de la Simulación con un empuje máximo de 1 N .	29
31	Llamada al resolutor del problema de Cauchy	31

1 Introducción

El cambio de órbita de un satélite es una maniobra necesaria en la puesta en órbita de un satélite. Este cambio de órbita permite al sistema cambiar desde una órbita de aparcamiento alcanzable por el lanzador a la órbita final pretendida en el proyecto.

Para realizar correctamente esta maniobra es necesario ejecutar previamente un análisis mediante cálculo numérico. Para este análisis comúnmente se usan distintos lenguajes de programación para reducir el tiempo de cálculo.

A lo largo de este informe se presentarán metodologías de programación y métodos de estimación distintos para optimizar el cálculo numérico. Además, se presentarán una serie de modelos de distintos problemas similares al estudiado y los códigos utilizados para dichos modelos.

2 Modelos numéricos

Los modelos simulados a lo largo del informe se implementarán usando diferentes esquemas numéricos o propagadores. El principal esquema numérico utilizado es el Runge-Kutta. Este propagador tiene diferentes niveles, que alcanzan precisiones cada vez mayores. Para los análisis preliminares, sin demasiado detalle, se ha empleado el Runge-Kutta de nivel 2, mientras que para análisis más precisos se ha utilizado el Runge-Kutta de nivel 4.

A parte del Runge-Kutta se ha implantado también un esquema numérico de tipo Euler explícito. Este modelo numérico, como se irá viendo a lo largo del informe, presenta problemas de convergencia para determinadas funciones, lo que hará que para obtener resultados lo más precisos posible se recurra a los esquemas del tipo Runge-Kutta.

3 Metodologías y herramientas de programación

Todo proyecto software que se quiere llevar a acabo debería seguir los pasos que se propone la Figura 1:

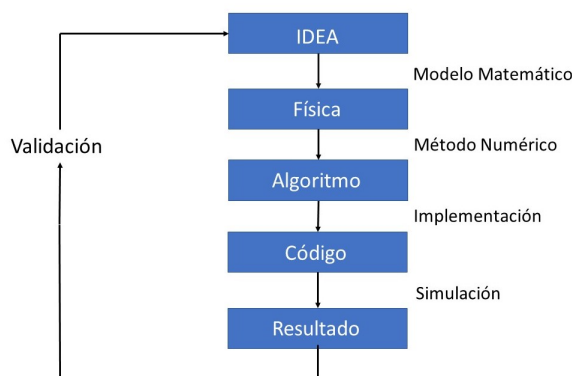


Figure 1: Esquema de los pasos a seguir en el diseño software

Este esquema nos explica que partiendo de una idea, lo primero que se debe hacer es plantear la física de nuestro problema a través de un modelo matemático. Una vez se ha definido el modelo matemático a usar se pasa al siguiente paso que es decidir el esquema numérico a emplear, como ya se ha comentado antes es esquema numérico será elegido en función del problema que se quiere resolver y la precisión que se quiere obtener. Este paso nos definirá el algoritmo de resolución de nuestro problema. Con los dos elementos anteriormente dichos se debe implementar el problema en un código. Para este trabajo se ha elegido FORTRAN, Además se ha para mayor claridad del software se ha dividido los archivos del programa; por ejemplo se han separado en archivos distintos módulos , unos módulos contienen las subrutinas encargadas de los esquemas numéricos, mientras que otros módulos contienen las funciones de la física de cada problema, finalmente existen otros archivos que relacionan estos dos módulos para resolver el programa. De esta manera se consigue separar la física del método numérico facilitando la comprensión del código. Una vez obtenido el código se llevan acabo distintas simulaciones, obteniendo así resultados que deberán ser validados y comparados con la idea original. Este proceso es iterativo y se espera que se repita este proceso hasta obtener unos resultados satisfactorios.

Además de seguir el esquema anteriormente descrito; a lo largo del curso en el que ha tenido cabida este proyecto se han llevado a cabo distintas metodologías de programación que facilitan la labor del equipo de ingenieros a lo largo de la simulación numérica de los distintos modelos.

En el caso de este proyecto, se ha utilizado principalmente la metodología

de *Extreme Programming*, que permite agilizar la labor de programación. Esta metodología consiste en dividir los equipos de programación en *Drivers*, los encargados de redactar el código, y *Reviewers*, encargados de contrastar lo que escriben los *Drivers* y detectar posibles errores en el código. Este sistema permite detectar con mayor facilidad erratas a lo largo de una secuencia de programación y realizar simultáneamente las labores de escritura y revisión. Además, al estar los *Reviewers* constantemente revisando el código, estos ingenieros, o programadores, van asimilando el código.

Dentro de las herramientas de programación destacan la técnica *top-down* y la técnica de continuación.

La técnica de continuación permite llegar a la solución de un problema complejo partiendo de un problema más sencillo a través de herramientas matemáticas. Esta herramienta permite simular, en primeras aproximaciones, problemas complejos con físicas difíciles de implementar descomponiéndolos en problemas más sencillos de implementar.

La técnica *top-down* permite la resolución de un problema compuesto de distintos componentes mediante la descomposición y análisis de esos subsistemas. Esta herramienta es útil para poder reutilizar partes del código para distintos problemas, ya que al programar cada parte del sistema en una subrutina diferente, permite llamar a dichas partes del código desde distintos códigos principales para la simulación de diferentes modelos.

Como ya se ha comentado anteriormente, se ha decidido usar el lenguaje de programación FORTRAN para este análisis. La utilización de este lenguaje implica el uso de las librerías *dislin* para la representación de gráficas. Las librerías utilizadas no son las propias de *dislin*, sino que se ha utilizado una modificación realizada por nuestro compañero Imanol Sardón Delgado, que nos las ha cedido para su uso. Esta librería permite incluir fácilmente distintos colores en las gráficas, así como títulos de gráfica y de eje y leyendas.

4 Ecuaciones de Bessel

Como primer paso en este curso se realizó el análisis de las funciones de Bessel, una de las funciones más estudiadas en la parte analítica de la asignatura dentro de la que esta englobada este estudio.

Este problema, aunque sencillo, permite familiarizarse con el entorno de programación utilizado, así como el uso de la herramienta *dislin* para la representación de gráficas. Además, se utilizará este ejemplo para validar los esquemas numéricos. La función de Bessel es el resultado de la siguiente ecuación diferencial ordinaria de orden dos:

$$z^2 \frac{d^2 w}{dz^2}(z) + z \frac{dw}{dz}(z) + z^2 w(z) = 0 \quad (1)$$

Los resultados obtenidos de este sencillo análisis se presentan en las gráficas a continuación:

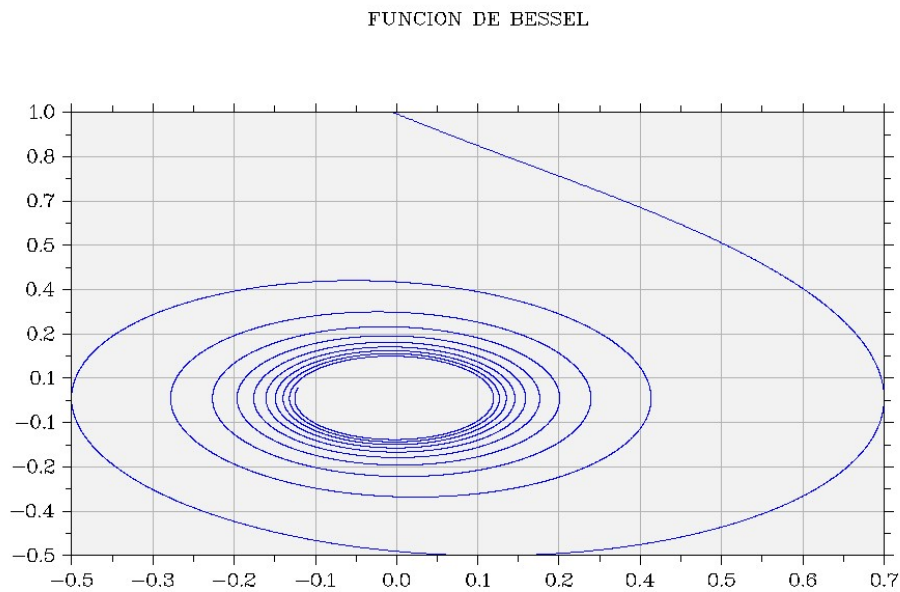


Figure 2: órbita de una ecuación de Bessel

FUNCION DE BESSEL

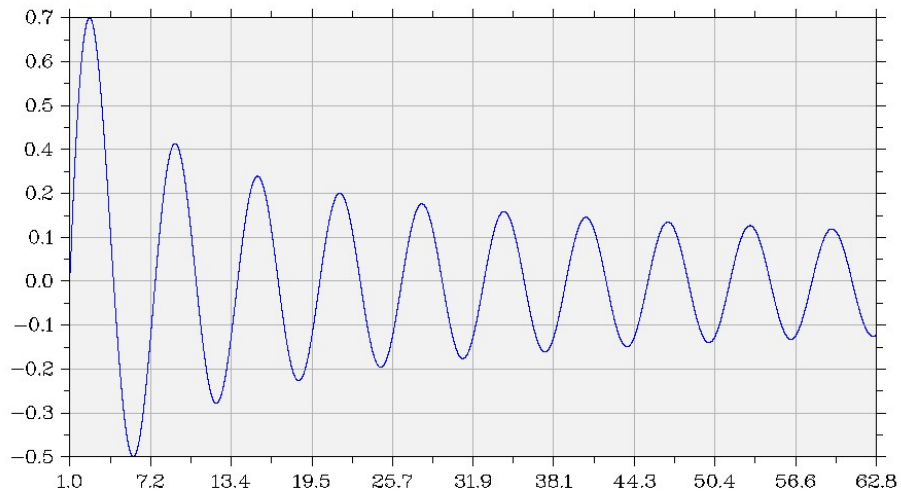


Figure 3: Función de Bessel en función del tiempo

5 Propagador Armónico

Otro de los modelos estudiados como paso previo a la resolución del problema planteado es el de un propagador armónico.

Este es un modelo sencillo, con una física más fácil de programar que la del proyecto propuesto, que servirá como prueba para las distintas metodologías y las herramientas presentadas en el apartado anterior.

5.1 Resultado de la Simulación

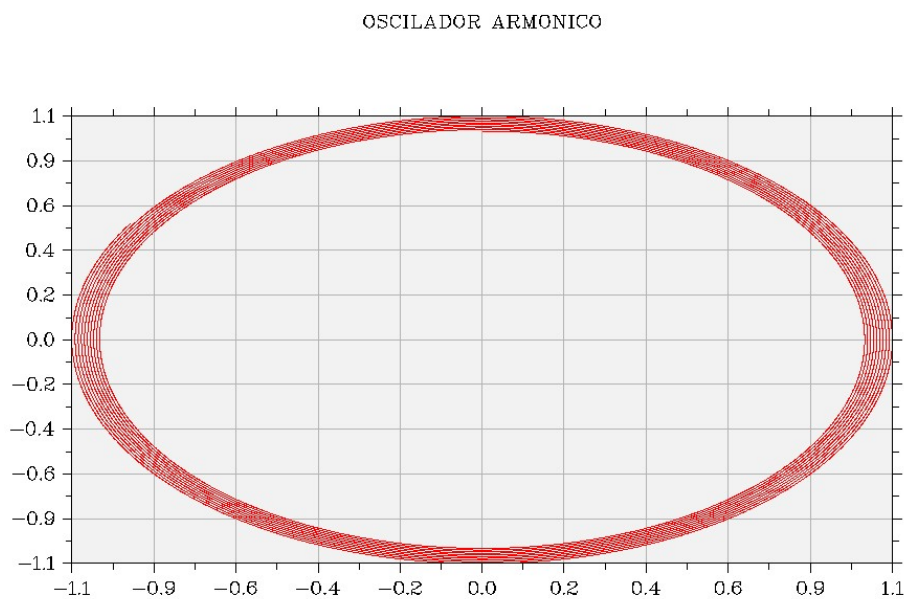


Figure 4: Órbita del oscilador armónico. Resultado de la simulación usando un propagador tipo Euler explícito

OSCILADOR ARMONICO

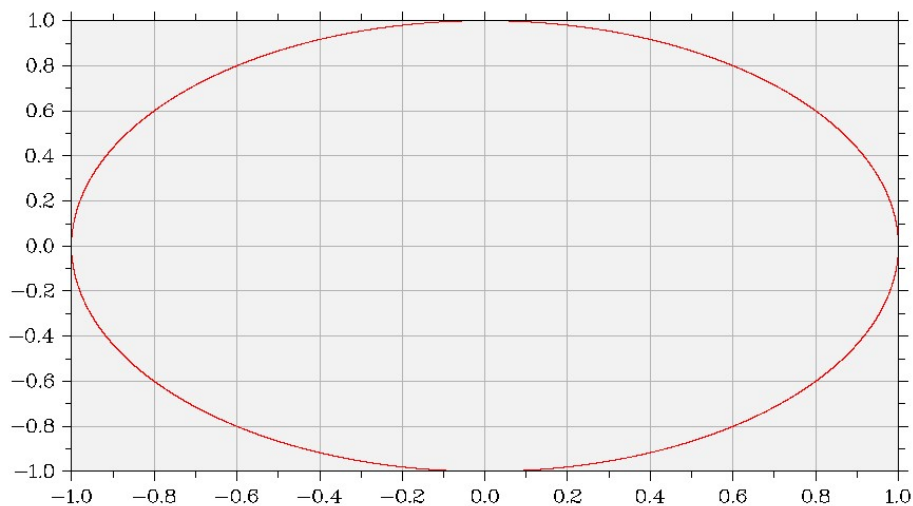


Figure 5: Órbita del oscilador armónico. Resultado de la Simulación utilizando un esquema numérico tipo Runge-Kutta 4

5.2 Conclusiones obtenidas de la simulación

Tratándose de un sistema muy sencillo donde la solución es un coseno. Se puede observar que después de 10 periodos el esquema numérico de tipo Euler, provoca una divergencia de las órbitas creando una forma espiral, por otra parte el esquema numérico tipo Runge-Kutta ha conseguido mantener el resultado esperado. Todo esto es debido a que el esquema numérico Euler no es convergente ni estable dentro del dominio del problema. De este primer y sencillo problema se determina que la elección del esquema numérico es determinante a la hora de poder simular satisfactoriamente un problema, pues nuestros resultados dependen del mismo, por ejemplo el estudio de la estabilidad de la solución de un problema de este carácter no podría ser llevado a cabo con un esquema numérico Euler, ya que de por sí el esquema sería inestable, incapacitando el estudio de estabilidad.

6 N cuerpos

Una vez realizado un análisis de un problema sencillo, como es el del propagador armónico, se pretende expandir todas las ideas implantadas en un modelo físico más complejo, como es el modelo de los N cuerpos.

El objetivo de realizar el estudio de este problema es comprobar que toda la metodología implementado en el propagador es aplicable a cualquier modelo físico, independientemente de la complejidad del mismo. Una vez se haya conseguido adaptar todo lo aprendido con el modelo sencillo al modelo de los N cuerpos la extrapolación de las herramientas a cualquier modelo físico, como el del cambio de órbita del satélite, será mucho más sencilla.

Para este problema se han realizado dos simulaciones distintas. En la primera se estudia el sistema Tierra-Luna-Satélite, que se asemeja al problema de los tres cuerpos donde la masa de un cuerpo es despreciable frente al resto. Dada la relevancia histórica de este problema se ha decidido estudiar el plano de Poincaré de este modelo y la estabilidad de Richardson. Por otra parte también se ha hecho una pequeña simulación de algunos de los planetas del sistema solar, para comprobar el funcionamiento del software para un mayor número de elementos.

6.1 Sistema Tierra-Luna-Satélite

Primero se empieza exponiendo los resultados obtenidos de distintas simulaciones donde se ha variado el tipo de esquema numérico y el tiempo total de simulación. Las condiciones iniciales del problema se recogen en la Figura 6

```

!*****
!      Variables
!*****
real :: t0 = 1, tf = 28*24*3600
integer :: i, condition
integer :: nbodys = 3 !CAMBIAR nbodys aquí y en el Funcion_F
integer, parameter :: n = 25000 !n=pasos de tiempo
real, allocatable :: U(:,,:), Up(:) !cambiar dimension a vector para poicare
real :: t (0:n) !, U(0:n,2)

allocate (U(0:n,4*nbodys),Up(0:6*nbodys))
!*****

!*****
! CONDICIONES INICIALES PARA EL SISTEMA TIERRA-LUNA-SATÉLITE
!*****

      U(0,:)= [ 0. ,0. , &      !POSICIÓN X-Y DE LA TIERRA
                3.84e8 , 0. , &  !POSICIÓN X-Y DE LA LUNA
                0. , - 5e7 , &   !POSICIÓN X-Y DEL SATÉLITE
                0. , -10. , &    !VELOCIDAD EN X-Y DE LA TIERRA
                0. , 1e3 , &     !VELOCIDAD EN X-Y DE LA LUNA
                3.e3, 0. ]       !VELOCIDAD EN X-Y DEL SATÉLITE
!*****

m(0)=10e24 !TIERRA
m(1)=10e22 !LUNA
m(2)=100   !SATÉLITE

```

Figure 6: Condiciones iniciales bajo las que se ha simulado el sistema

POSICION DE LA TIERRA-LUNA-SATELITE

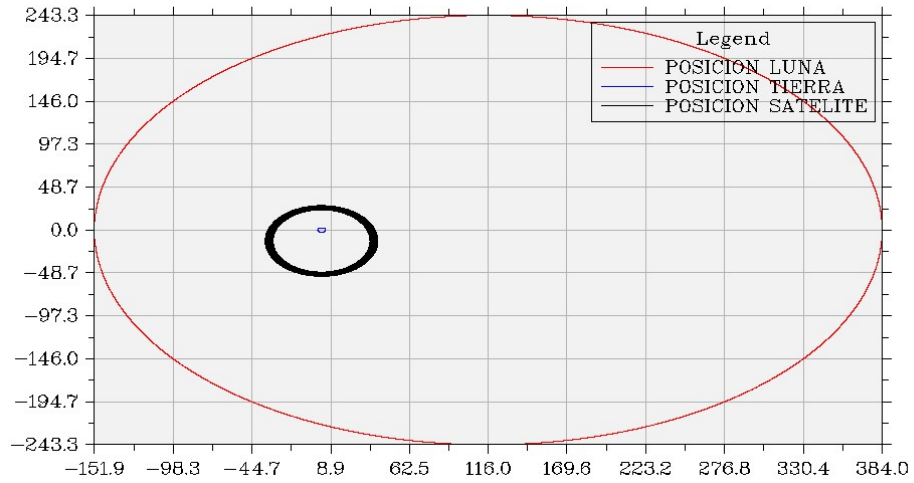


Figure 7: Resultado de la Simulación utilizando un esquema numérico tipo Rungue-Kutta 4 para 28 días de simulación. Los valores se han dividido por 10^6

POSICION DE LA TIERRA-LUNA-SATELITE

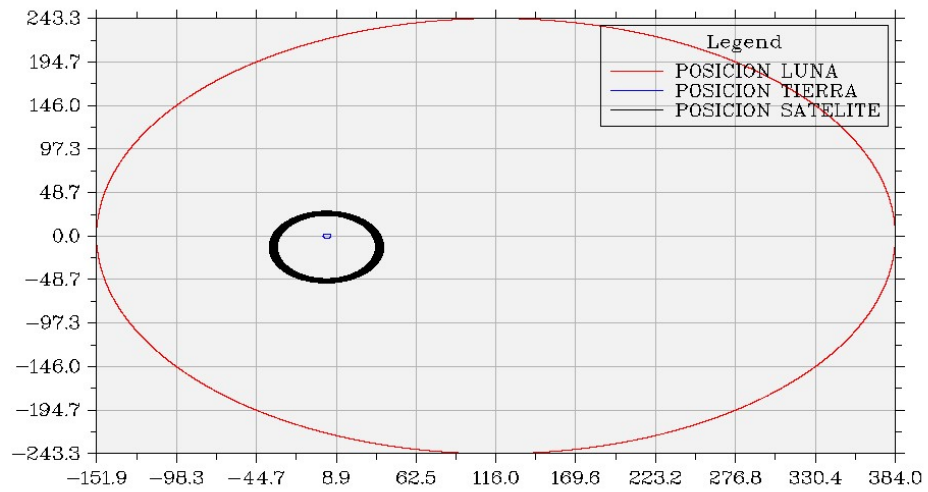


Figure 8: Resultado de la Simulación utilizando un esquema numérico tipo Rungue-Kutta 2 para 28 días de simulación. Los valores se han dividido por 10^6

POSICION DE LA TIERRA-LUNA-SATELITE

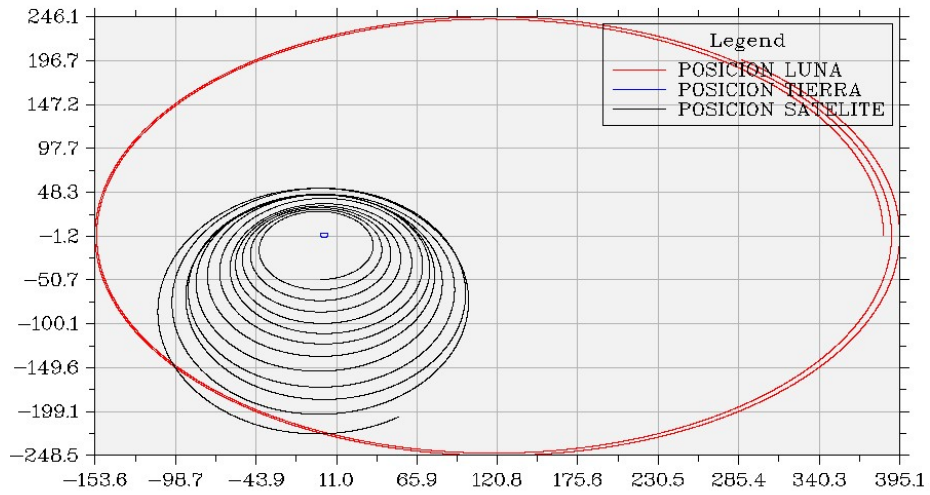


Figure 9: Resultado de la Simulación utilizando un esquema numérico tipo Euler para 28 días de simulación. Los valores se han dividido por 10^6

POSICION DE LA TIERRA-LUNA-SATELITE

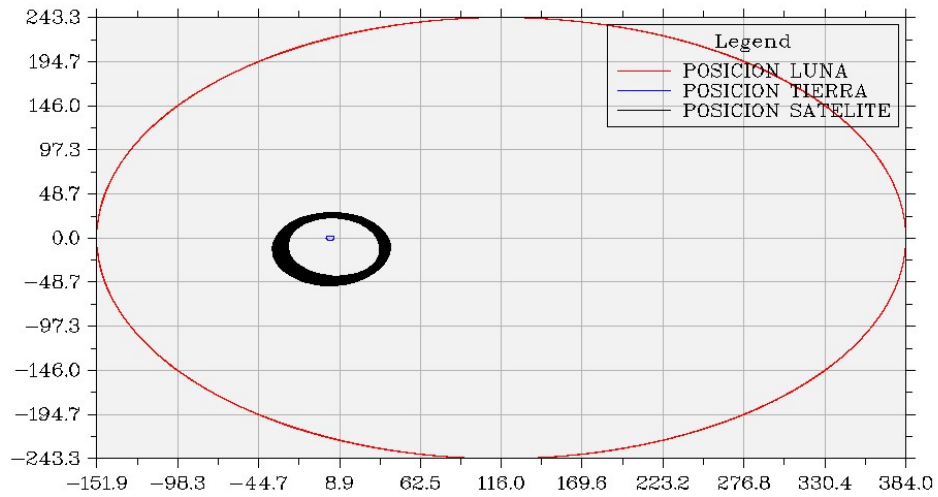


Figure 10: Resultado de la Simulación utilizando un esquema numérico tipo Rungue-Kutta 4 para un año de simulación. Los valores se han dividido por 10^6

POSICION DE LA TIERRA-LUNA-SATELITE

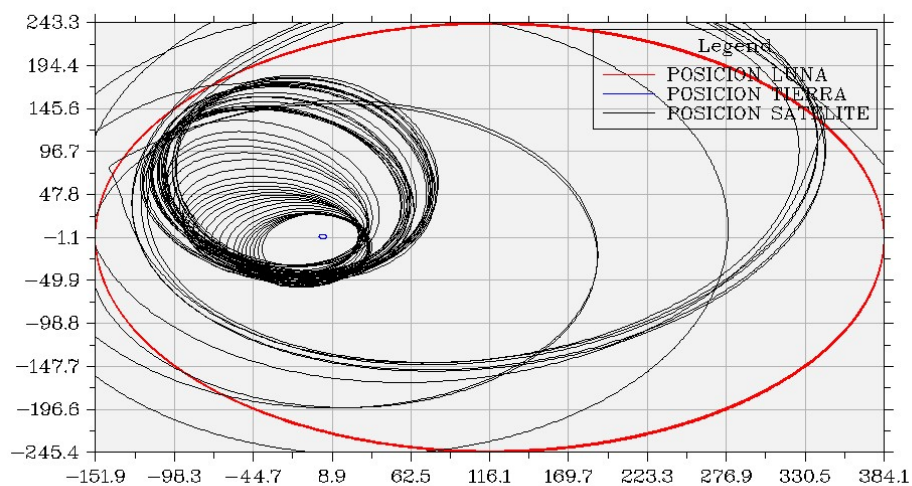


Figure 11: Resultado de la Simulación utilizando un esquema numérico tipo Rungue-Kutta 2 para un año de simulación. Los valores se han dividido por 10^6

POSICION DE LA TIERRA-LUNA-SATELITE

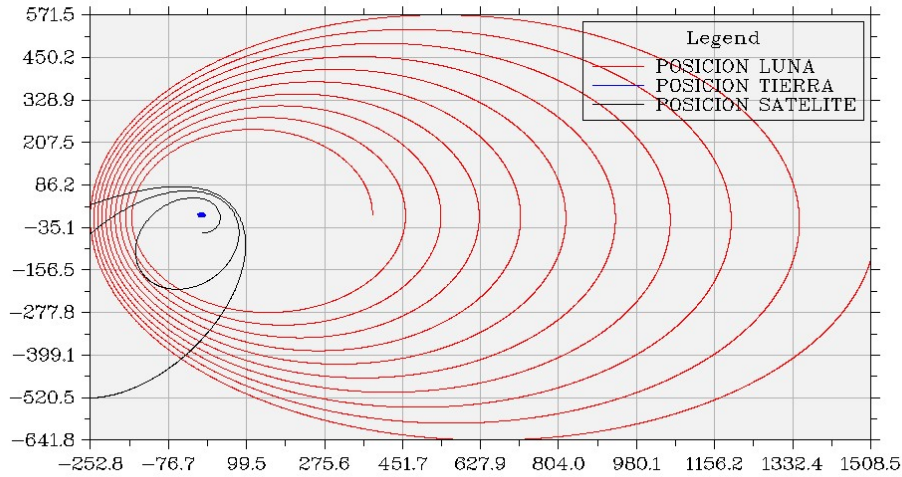


Figure 12: Resultado de la Simulación utilizando un esquema numérico tipo Euler para un año de simulación. Los valores se han dividido por 10^6

Como se puede observar, y se ha comentado anteriormente, la elección del tipo de esquema numérico es fundamental. Observando las figuras se aprecia que el mejor esquema numérico para simular estos sistemas es el Runge-Kutta de orden 4, Por lo tanto en los siguientes puntos se usará el esquema numérico Runge-Kutta 4. Un punto a destacar es que el satélite no describe una órbita perfectamente circular como podría esperarse *a priori* por lo tanto este objeto va ser el próximo elemento de estudio. Para analizar más en detalle este aspecto se ha realizado el plano de Poincaré. En este punto se han simulado varias órbitas para distintas condiciones iniciales obteniéndose un mapa de puntos distinto para cada condición inicial. También se ha representado el plano de Poincaré variando la cantidad de tiempo simulado.

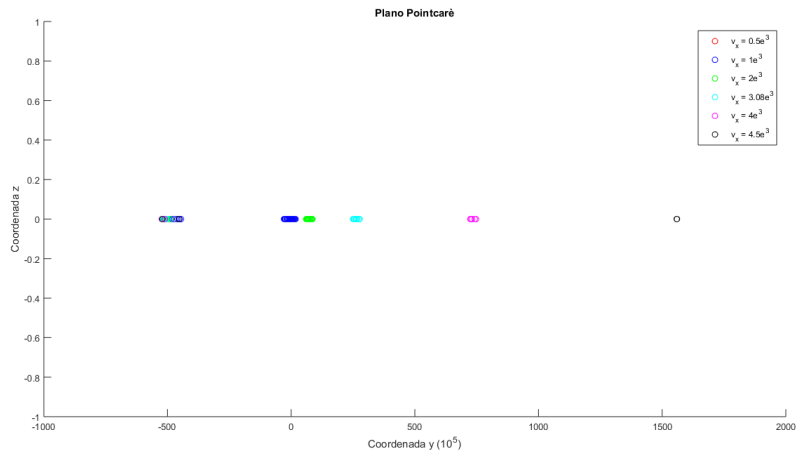


Figure 13: Intersección de la órbita en el plano de Poincaré variando las condiciones iniciales:1

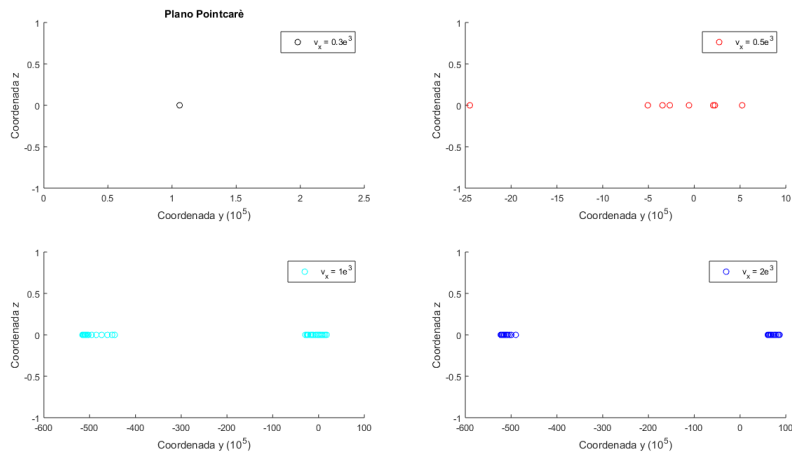


Figure 14: Intersección de la órbita en el plano de Poincaré variando las condiciones iniciales:2

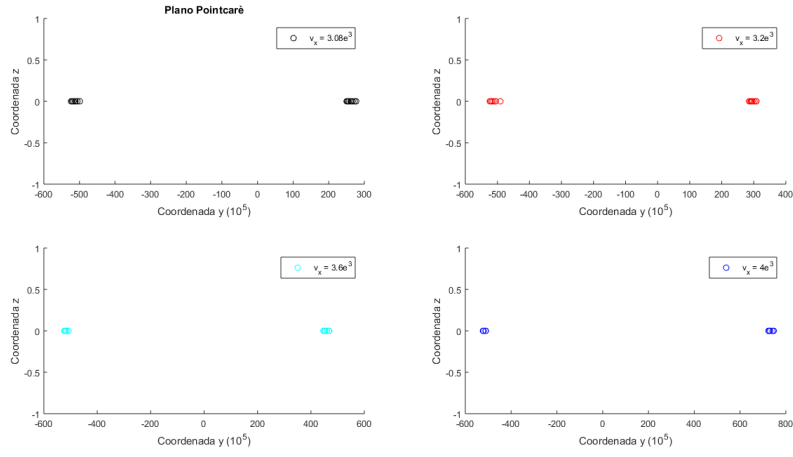


Figure 15: Intersección de la órbita en el plano de Poincaré variando las condiciones iniciales:3

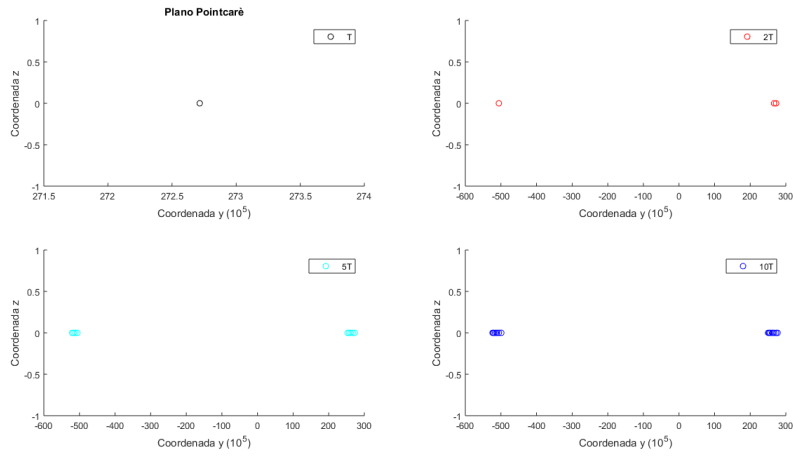


Figure 16: Intersección de la órbita en el plano de Poincaré aumentado la cantidad de periodos simulados

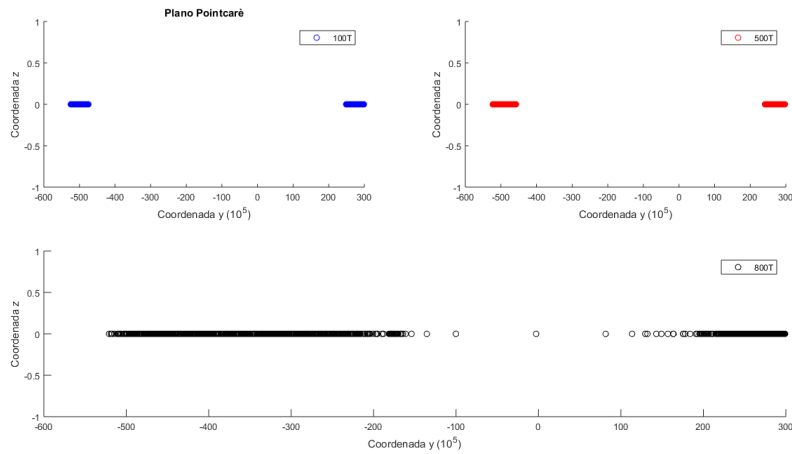


Figure 17: Intersección de la órbita en el plano de Poincaré aumentado la cantidad de periodos simulados

A la vista de las gráficas se aprecia que el plano de Poincaré varía tanto con las condiciones iniciales como con los periodos simulados. A su vez, se aprecia que en un mismo plano hay dos regiones de puntos, una en la zona negativa y otra en la positiva. Esto se debe a la forma elíptica de las soluciones, lo que provoca que haya dos puntos de intersección con el plano con cada vuelta, un punto en la zona del semieje negativo y el otro en la zona del semieje positivo.

6.2 Sistema Solar

Esta simulación se ha llevado acabo simulando ciertos elementos dentro del sistema solar entre ellos se encuentra: el Sol, la Tierra, la Luna y Marte. Las condiciones iniciales de dicha simulación se recogen en la Figura 18

```

*****
! CONDICIONES INICIALES PARA SOL-TIERRA-MARTE-LUNA
*****
U(0,:)= [ 1.5e11 , 0. , &      !POSICIÓN X-Y DE LA TIERRA
          1.5e11 , -3.84e8 , &   !POSICIÓN X-Y DE LA LUNA
          2.3e11 , 0. , &       !POSICIÓN X-Y DE MARTE
          0. , 0. , &          !POSICIÓN X-Y DEL SOL
          0. , 30.e3 , &       !VELOCIDAD EN X-Y DE LA TIERRA
          1.e3 , 30.e3 , &     !VELOCIDAD EN X-Y DE LA LUNA
          0. , 24.e3 , &      !VELOCIDAD EN X-Y DE MARTE
          0. , 0. ]           !VELOCIDAD EN X-Y DEL SOL
*****

m(0)=10e24 !TIERRA
m(1)=10e22 !LUNA
!m(2)=100  !SATÉLITE

m(2)=10e23 !MARTE
m(3)=2e30  !SOL

```

Figure 18: Condiciones iniciales bajo las que se ha simulado el sistema

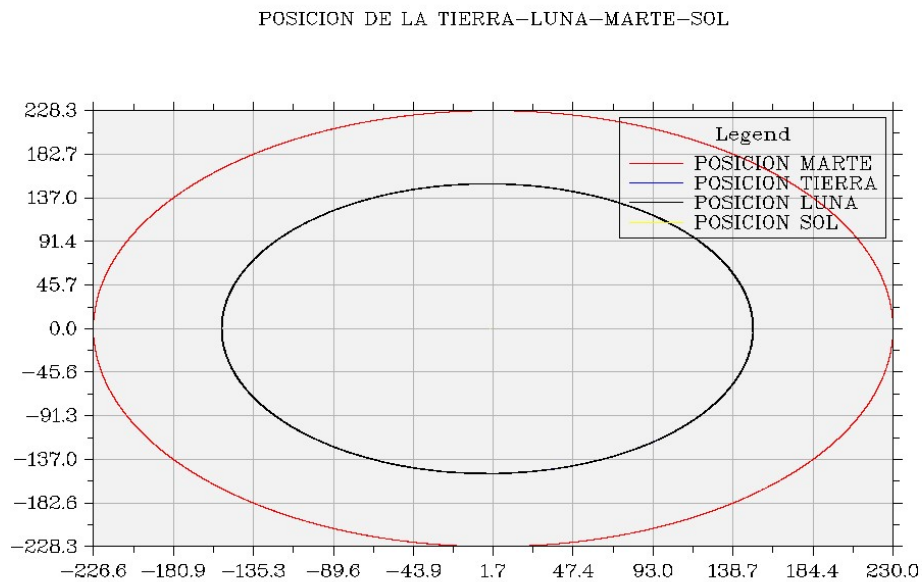


Figure 19: Resultado de la Simulación utilizando un esquema numérico tipo Runge-Kutta 4 para 10 años de simulación. Los valores se han dividido por 10^6

POSICION DE LA TIERRA-LUNA-MARTE-SOL

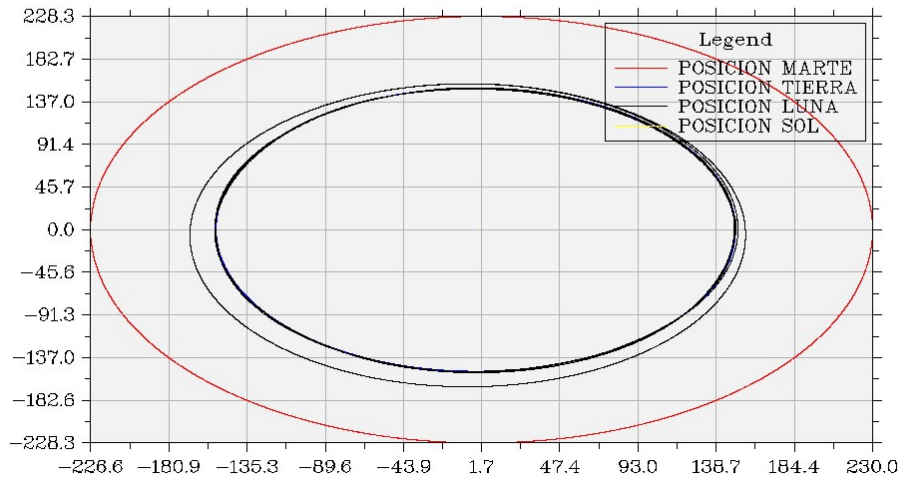


Figure 20: Resultado de la Simulación utilizando un esquema numérico tipo Rungue-Kutta 2 para 10 años de simulación. Los valores se han dividido por 10^6

POSICION DE LA TIERRA-LUNA-MARTE-SOL

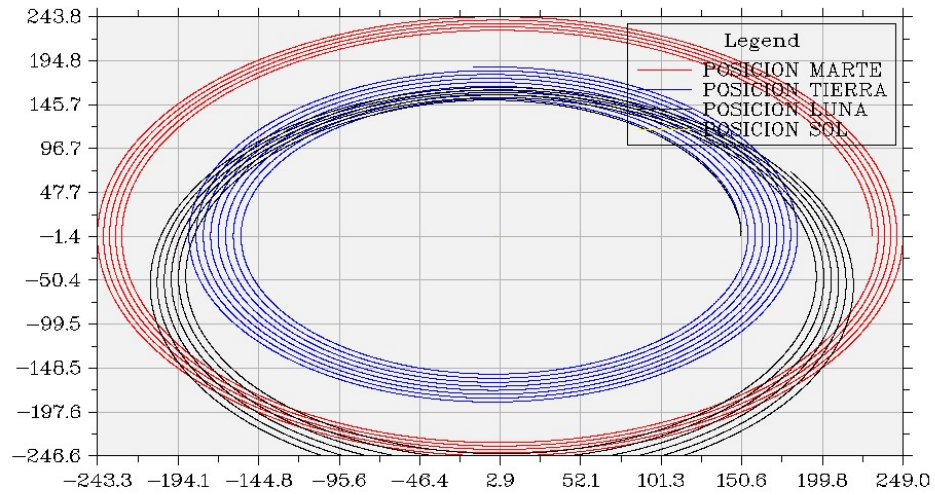


Figure 21: Resultado de la Simulación utilizando un esquema numérico tipo Euler para 10 años de simulación. Los valores se han dividido por 10^6

POSICION DE LA TIERRA-LUNA-MARTE-SOL

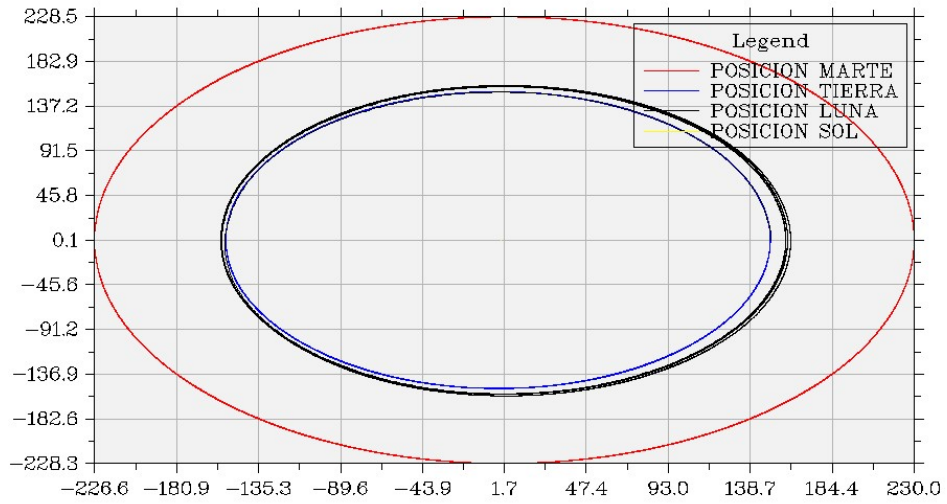


Figure 22: Resultado de la Simulación utilizando un esquema numérico tipo Rungue-Kutta 4 para 50 años de simulación. Los valores se han dividido por 10^6

POSICION DE LA TIERRA-LUNA-MARTE-SOL

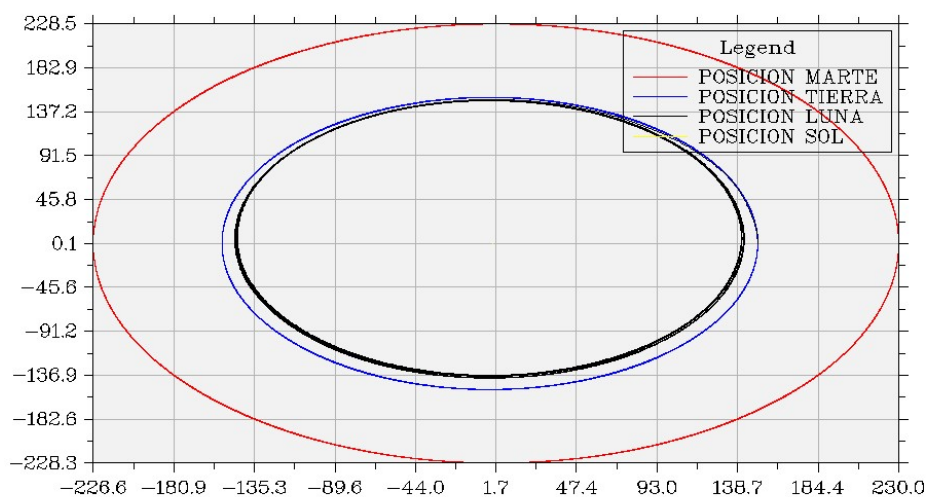


Figure 23: Resultado de la Simulación utilizando un esquema numérico tipo Rungue-Kutta 2 para 50 años de simulación. Los valores se han dividido por 10^6

POSICION DE LA TIERRA-LUNA-MARTE-SOL

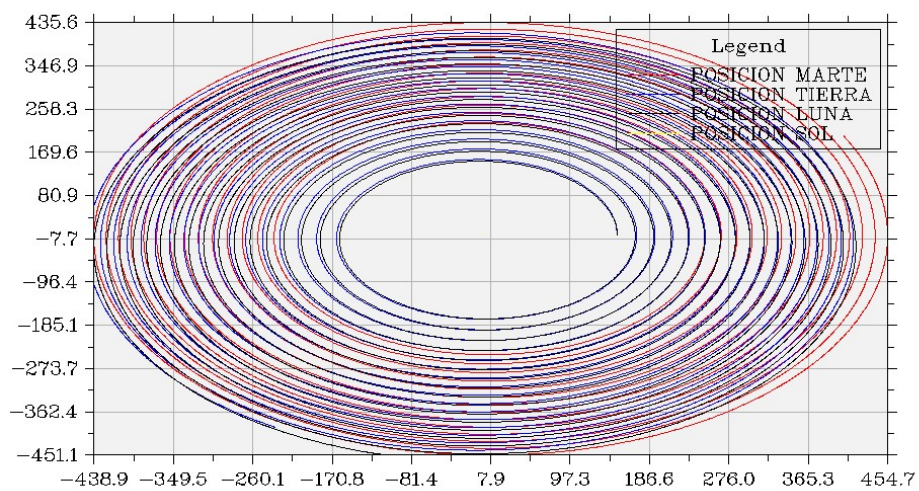


Figure 24: Resultado de la Simulación utilizando un esquema numérico tipo Euler para 50 años de simulación. Los valores se han dividido por 10^6

De nuevo, la solución obtenida con el esquema de tipo Euler diverge con mayor facilidad que la de Runge-Kutta. Además, en este caso, para una simulación de 50 años se empieza a apreciar una variación significativa en el esquema más preciso, el Runge-Kutta 4. Esto nos hace ver que para simular correctamente este sistema sería necesario implementar un esquema numérico de mayor orden, que permita corregir dicha desviación para tiempos grandes.

7 Cambio de órbita de un satélite

Para finalizar el análisis realizado a lo largo de este informe se retoma el problema principal del cambio de órbita de un satélite. En concreto, el objetivo principal es estimar el impacto del fallo del sistema de apagado del motor electrostático sobre la órbita final obtenida.

En el caso de que el sistema de apagado del motor no funcione correctamente y el motor no se detenga cuando corresponde, la órbita obtenida por el satélite será diferente a la que se quería obtener en un inicio. En esta parte del informe se estudiará la variación de la órbita final en el caso de que el motor no se apague.

7.1 Simulación

Se ha decidido utilizar un caso muy sencillo donde se observa la evolución de la órbita a lo largo del tiempo, variando entre las distintas simulaciones el valor de empuje máximo que proporciona el sistema de propulsión. Para estas simulaciones se considera que el satélite se encuentra en una órbita inicial que es circular, alrededor de la tierra, con una masa seca de 1000 kg y 200 kg de combustible. El impulso específico es de 2000 s. El esquema numérico usado para la simulación es el Runge-Kutta 4 y se simula para un tiempo igual a 100 periodos de la órbita nominal.

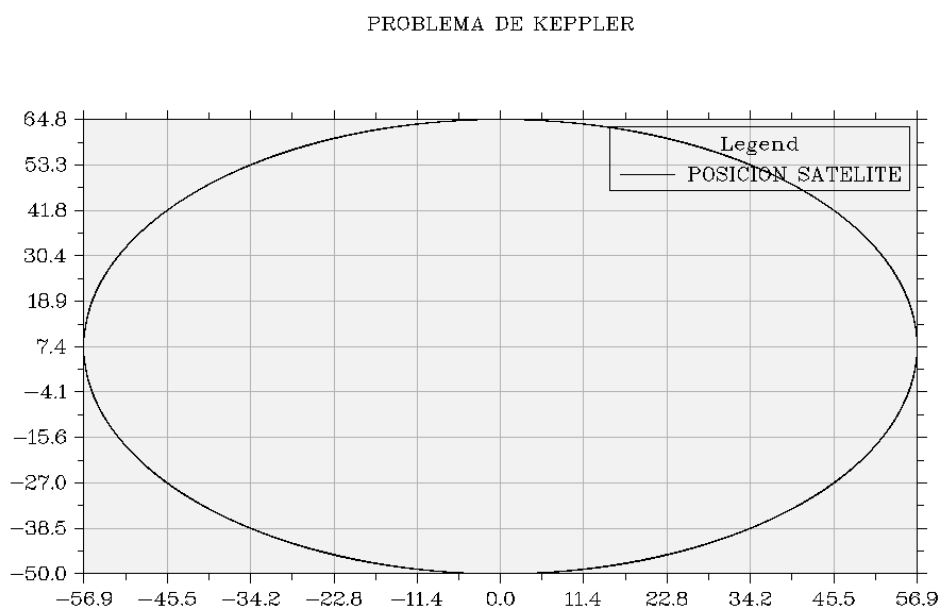


Figure 25: Resultado de la Simulación con motor parado (condiciones iniciales)

PROBLEMA DE KEPLER

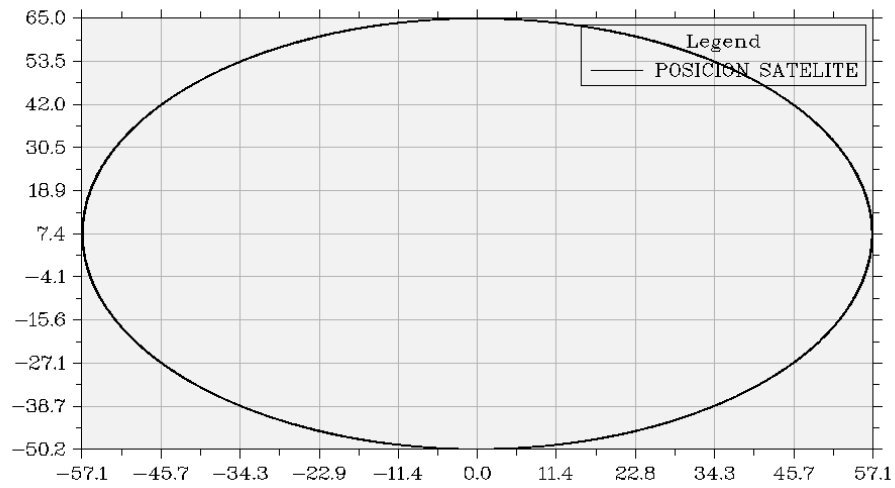


Figure 26: Resultado de la Simulación con un empuje máximo de 0.001 N

PROBLEMA DE KEPLER

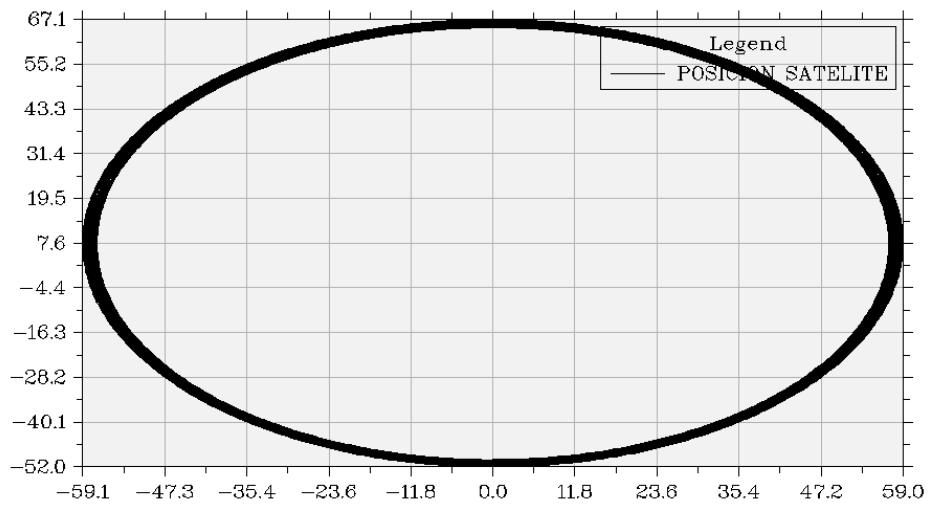


Figure 27: Resultado de la Simulación con un empuje máximo de 0.01 N

PROBLEMA DE KEPLER

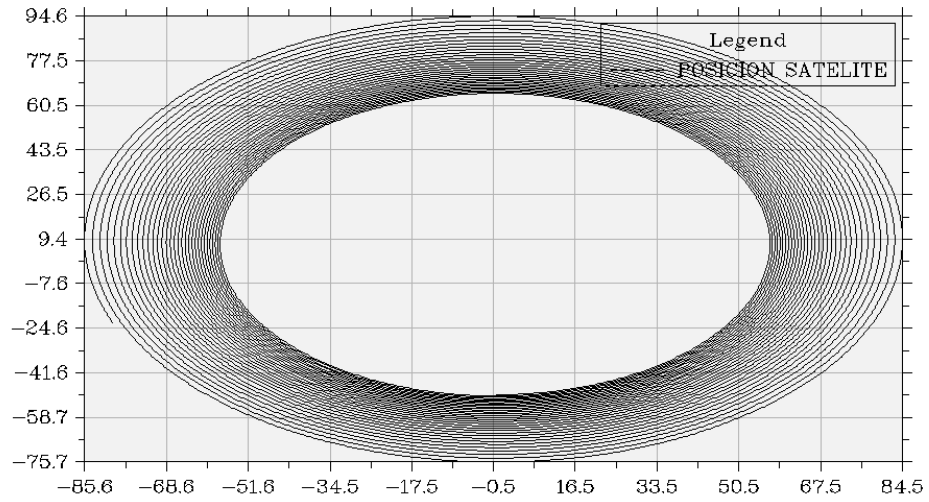


Figure 28: Resultado de la Simulación con un empuje máximo de 0.1 N

PROBLEMA DE KEPLER

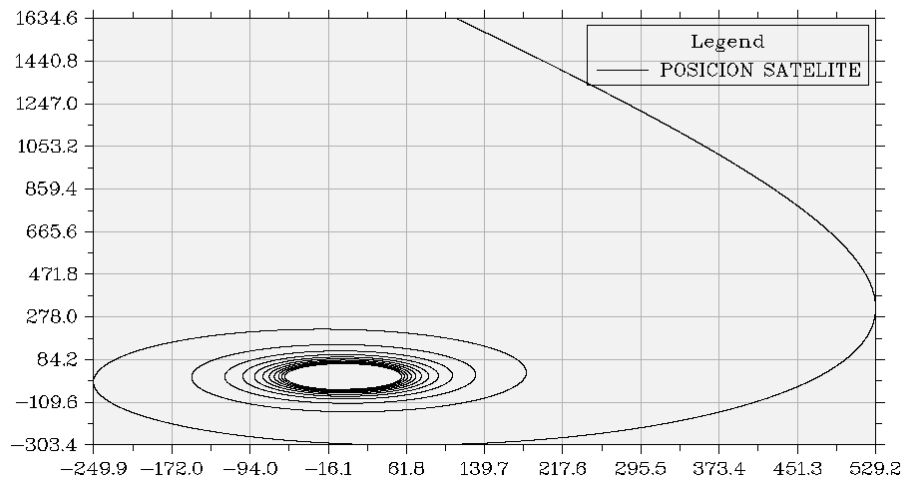


Figure 29: Resultado de la Simulación con un empuje máximo de 0.5 N

PROBLEMA DE KEPLER

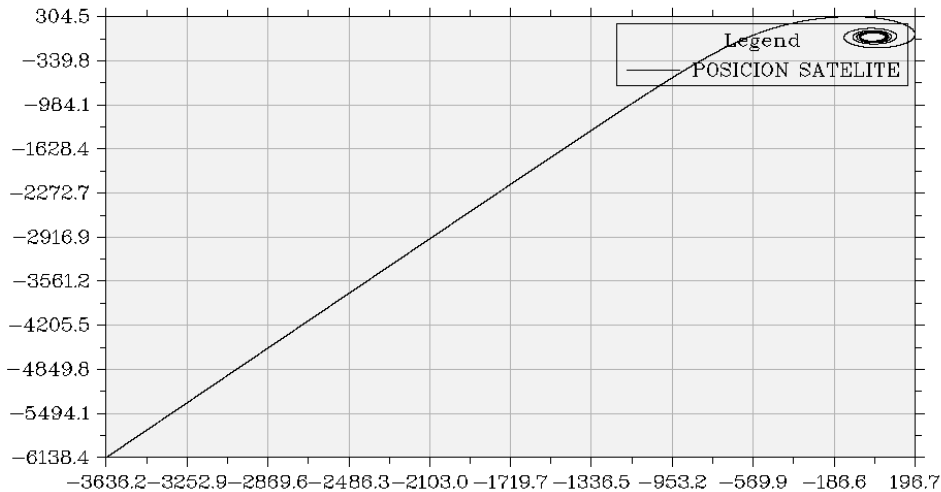


Figure 30: Resultado de la Simulación con un empuje máximo de 1 N

7.2 Conclusión

Como se puede observar la órbita del vehículo va creciendo debido al empuje que proporciona el motor, en los primeros casos dado que el empuje del motor es muy pequeño el satélite se mantiene en órbita alrededor de la tierra pero según se aumenta el empuje máximo esta órbita cada vez llega más lejos de la tierra hasta que adquiere la velocidad de escape y se aleja hacia el infinito a través de una hipérbola. En el análisis del problema estudiado, el del fallo en el sistema de apagado del motor, esto implica que si el motor tiene un empuje bajo, como es el caso de un motor electrostático, el no apagado del motor provocará la variación, significativa, de la órbita, pero manteniéndose dentro de la zona de influencia de la Tierra y permitiendo nuevas maniobras de corrección para alcanzar la órbita deseada. En el caso de disponer de un motor con un empuje nominal mayor, a partir de 0.5N, el no apagado del motor podría llegar a provocar la salida de la zona de influencia y la pérdida efectiva, sin posibilidad de reinserción, del satélite.

8 Conclusiones

- Se ha conseguido desarrollar de manera satisfactoria en FORTRAN un programa que es capaz de **resolver cualquier problema de EDOS de condiciones iniciales**, siempre y cuando se introduzca la función problema dentro del archivo pertinente. También se tienen que tener en cuenta las limitaciones existentes debido a los pocos esquemas numéricos que hay implementados.
- La selección del tipo de esquema numérico es importante, ya que una mala elección del tipo de esquema numérico puede desembocar en una solución errónea. Antes de empezar a simular cualquier problema conviene estudiar el tipo de sistema para seleccionar un esquema numérico acorde a lo que se quiere realizar. En el caso de los problemas propuestos a lo largo de este informe los esquemas numéricos del tipo Runge-Kutta han demostrado generar soluciones más precisas que los de tipo Euler. Por tanto, para problemas de órbitas es necesario utilizar **esquemas numéricos de alto orden** para conseguir **convergencia numérica** para tiempos de integración largos. El mayor orden que se ha utilizado en el trabajo es un Runge-Kutta de orden 4, que en ocasiones ha resultado insuficiente, sobretodo cuando se propagan muchos periodos.
- La distribución en funciones de los distintos elementos utilizados en los programas permite variar las funciones analizadas, los integradores y la evolución temporal de manera sencilla.

ANEXO I: SOFTWARE

En este apartado se realizará una breve explicación de los distintos módulos que se utilizan en el programa.

- **MAIN EDOS:** Corresponde al programa principal, el cual, utilizará el resto de módulos. En este programa se definirá la partición temporal, se impondrán las condiciones iniciales del problema de EDOS, y se hará la llamada a las subrutinas que resuelven el problema de ecuaciones diferenciales ordinarias. Como se puede observar en la siguiente figura, la llamada al resolutor del problema de Cauchy implica 4 argumentos: la partición temporal, el operador diferencial correspondiente al problema que se quiere integrar (NBodies, OsciladorArmónico, FunciónBessel, etc...), el esquema numérico que se quiere emplear y la matriz que guarda las soluciones del problema para cada paso de tiempo (U).

```
call Cauchy_Problem ( Time_Domain = t, Differential_Operator = N_bodys2D, Scheme = Runge_Kutta4, &
Solution=U) !POR DEFECTO ESTÁ EL ESQUEMA Runge_Kutta 2. Si se quiere utilizar otro se debe incluir en los argumentos.
```

Figure 31: Llamada al resolutor del problema de Cauchy

- **FuncionF:** este módulo guarda en subrutinas todos los problemas que se desee integrar. En concreto se han programado 5 problemas: función de Bessel, Oscilador Armónico, problema de Keppler, problema de los N cuerpos en 2 dimensiones y el problema de los N cuerpos en 3 dimensiones.
- **StateVectorEvolution:** Contiene el módulo *TemporalSchemes*. Se han programado los esquemas numéricos que se han empleado: Euler explícito, Runge-Kutta2 y Runge-Kutta4. Cabe destacar que los bucles que recorren todos los pasos temporales se han introducido en cada esquema numérico y no en el programa principal. Se ha programado un *abstract interface* para poder utilizar en todos los esquemas la función F relativa a cada problema.
- **ODEInterface:** En este módulo se ha definido un *abstract interface* para poder caracterizar las funciones F que definen cada problema. Esto se utiliza para definir completamente la forma que tienen las funciones F que se van a integrar y facilita la implementación y entendimiento del código. En el módulo StateVectorEvolution, se implementará un *procedure* (una única vez) para que todos los esquemas numéricos utilicen la función F del sistema de forma que se encuentre totalmente caracterizada matemáticamente. De esta forma se asegurará que la función del sistema sea del mismo orden que el vector de estado asociado a cada problema.
- **dislinmod:** Este módulo es un *wrapper* realizado por el compañero Imanol Sardón Delgado (quien nos ha permitido usar su código y nos ha ayudado a implementarlo), que se ha utilizado para facilitar el uso de la herramienta gráfica *dislin*.