

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Máster en Ingeniería Aeronáutica



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Trabajo de Fin de Máster

Obtención y análisis de una trayectoria
interplanetaria a Marte introduciendo
aceleraciones perturbadoras

Autor:

Eva Terradillos Estévez

Tutores:

Santiago Emmanuel Moll López

José Antonio Morano Fernández

Alicia Herrero Debón

Valencia, 10/09/2020

Resumen

La finalidad principal del presente trabajo es resolver la ecuación fundamental del movimiento relativo de dos cuerpos en el espacio, introduciendo un alto nivel de precisión, que se alcanza tomando en consideración las aceleraciones perturbadoras más relevantes al efectuar los cálculos. El proceso seguido ha sido integrar numéricamente la ecuación diferencial y la validación de los datos mediante casos reales. En concreto, se realiza un estudio de la trayectoria del satélite Terra durante dos meses y de la Mars2020 desde su lanzamiento hasta su encuentro con Marte. Las perturbaciones que van a ser implementadas comprenden las aceleraciones debidas al potencial gravitatorio terrestre, la atmósfera terrestre, a los armónicos zonales y la gravedad de algunos cuerpos del Sistema Solar que influyen a la hora de determinar la órbita. Para conseguir todo lo anterior, se ha desarrollado una aplicación que permite trabajar con dos sistemas de referencia distintos, geocéntrico y heliocéntrico, con las perturbaciones deseadas o, en su caso, el problema de dos cuerpos, y con maniobras impulsivas o no.

Abstract

The main purpose of this document is to solve the equation of the relative motion of two bodies in space, with a high level of accuracy, achieved by taking into consideration the most relevant perturbations in calculations. The process consists of developing the numerical integration of the differential equations and the validation of the data using real cases, like a study of the trajectory of the Terra satellite during two months and of Mars2020 from its launch to its encounter with Mars. The perturbations to be implemented include accelerations due to the Earth's gravitational potential, the Earth's atmosphere, the zonal harmonics, and the gravity of some Solar System bodies which influence the orbit determination. All of the above has been achieved by the development of an application that allows working with two different reference systems, geocentric and heliocentric, with perturbations or not, and with impulsive or non-impulsive maneuvers.

Resum

La finalitat principal d'aquest treball és resoldre l'equació fonamental de el moviment relatiu de dos cossos en l'espai, introduint un alt nivell de precisió, que s'aconsegueix considerant les acceleracions pertorbadores més rellevants a l'efectuar els càlculs. El procés seguit ha sigut integrar numèricament l'equació diferencial i la validació de les dades mitjançant casos reals. En concret, es realitza un estudi de la trajectòria de el satèl·lit Terra durant dos mesos i el de la Mars2020 des del seu llançament fins que es troba amb Mart. Les pertorbacions que van a ser implementades comprenen les acceleracions degudes a el potencial gravitatori terrestre, l'atmosfera terrestre, als harmònics zonals i la gravetat d'alguns cossos de el sistema solar que influeixen a l'hora de determinar l'òrbita. Per aconseguir tot l'anterior, s'ha desenvolupat una aplicació que permet treballar amb dos sistemes de referència diferents, geocèntric i heliocèntric, amb les pertorbacions desitjades o, si escau, el problema de dos cossos, i permet afegir maniobres impulsives.

Índice

Índice de figuras	7
Índice de cuadros	11
Nomenclatura	12
1 INTRODUCCIÓN	1
2 ESTADO DEL ARTE	3
2.1 Historia de las misiones espaciales	3
2.2 Diseño de misiones espaciales	7
2.3 Propagación de la trayectoria	10
3 FUNDAMENTOS TEÓRICOS	12
3.1 Sistemas de referencia	12
3.1.1 Sistemas Interplanetarios: Sistema Heliocéntrico	12
3.1.2 Sistema Inercial Geocéntrico	13
3.2 Vector de estado	13
3.2.1 Coordenadas cartesianas	14
3.2.2 Elementos orbitales	14
3.3 Ecuación del movimiento	18
3.3.1 Técnicas de propagación	18
3.4 Perturbaciones	20
3.4.1 Potencial gravitatorio terrestre	22
3.4.2 Resistencia atmosférica	25
3.4.3 Efectos gravitatorios de un tercer cuerpo	28
3.4.4 Presión de radiación solar	29
3.5 Maniobras interorbitales	31
3.5.1 Maniobras impulsivas	32
3.5.2 Transferencia de Hohmann	33

4	APLICACIÓN EN MATLAB	34
4.1	Estructura de la aplicación	34
5	PROGRAMA	43
5.1	Estructura	43
5.2	Funciones	44
5.2.1	Tratamiento de datos	44
5.2.2	Integración	46
5.2.3	Perturbaciones	47
6	VALIDACIÓN	48
6.1	Problema de dos cuerpos: función Integracion2body	48
6.2	Método de Cowell	51
6.2.1	Perturbación causada por los planetas, la Luna y el Sol: función IntegracionPlan	51
6.2.2	Perturbación causada por la presión de radiación solar: función IntegracionSRP	54
6.2.3	Perturbación causada por la no uniformidad del campo gravitatorio terrestre: función IntegracionHarm	56
6.2.4	Perturbación causada por la atmósfera terrestre: función IntegracionAtm	59
6.2.5	Todas las perturbaciones consideradas: función IntegracionPlanSRPHarmAtm	61
7	RESULTADOS	64
7.1	Trayectoria del satélite geocéntrico Terra	64
7.2	Cambio de altura del perigeo y del apogeo de una órbita geocéntrica	69
7.2.1	Cambio de altura del perigeo	69
7.2.2	Cambio de altura del apogeo	71
7.3	Trayectoria de la Mars2020	74
8	CONCLUSIONES Y FUTURAS LÍNEAS DE ESTUDIO	80
8.1	Conclusiones	80
8.2	Futuras líneas de estudio	82
9	REFERENCIAS	84

ANEXOS	86
A Código de la Aplicación	86
B Código de MATLAB	126
C Presupuesto	171
C.1 Recursos humanos	171
C.2 Software para el despliegue	171
C.3 Recursos para el desarrollo	171
C.4 Presupuesto total	172

Índice de figuras

1	Satélite Sputnik . Primer satélite de la historia [3].	3
2	Promera mujer en viajar al espacio, Valentina Tershkova en el satélite Vostok 6 [3].	4
3	Misión Apollo 11 . Primera vez que humanos llegaban a la superficie de la Luna [3].	5
4	Foto de estrellas lejanas obtenida por el Telescopio Espacial Hubble [3].	6
5	Estación Espacial Internacional [3].	6
6	Arquitectura de la misión formada por los distintos elementos [4].	8
7	Sistema Heliocéntrico [15]	12
8	Sistema Geocéntrico [16]	13
9	Representación de los elementos orbitales [17].	14
10	Elementos de una órbita elíptica	16
11	Representación de las distintas anomalías dentro de la órbita elíptica.	17
12	Tipos de perturbaciones según su efecto en los elementos orbitales [4].	20
13	Influencia de las distintas perturbaciones en un satélite que orbita alrededor de la Tierra [9].	21
14	Tipos de armónicos esféricos en función del valor de sus índices [12].	23
15	Bandas de los armónicos zonales de distintos ordenes [8].	23
16	Resumen de algunos coeficientes de los armónicos esféricos [12].	24
17	Evolución de los modelos atmosféricos [8].	26
18	Valores de la densidad de referencia (ρ_0), de la altura de referencia (h_0) y de la altitud de escala (H), en función de la altura del satélite (h_{ellp}) [8].	27
19	Relación velocidad de maniobra - consumo de combustible para distintos impulsos específicos [14].	32
20	Esquema de la transferencia de Hohmann elíptica [14].	33
21	Diagrama de flujo de la utilización de la aplicación diseñada.	34
22	Definición de las fechas inicial y final del trayecto y selección del sistema de coordenadas	35
23	Definición del vector de estado inicial.	36
24	Elección sobre maniobra con impulsos o no. Al ser sí, se necesitan el número de impulsos.	36
25	Definición de los datos de los distintos impulsos.	37
26	Opciones de integración para órbitas heliocéntricas.	37

27	Opciones de integración para órbitas geocéntricas.	39
28	Elección sobre maniobra con impulsos o no. Al ser <i>no</i> , se habilitan las opciones de integración.	39
29	Elección de la opción <i>Two Body</i>	40
30	Elección de la opción <i>Zonal Harmonics</i> y <i>Atmósfera</i>	41
31	Elección de la opción <i>Planetas</i> y <i>SRP</i>	42
32	Diagrama de flujo de las distintas funciones utilizadas para desarrollar el código del programa.	43
33	Error absoluto obtenido en el valor de la posición a lo largo de la trayectoria, en la comparación con los resultados del propagador <i>GMAT</i> , para una simulación sin perturbaciones.	49
34	Error relativo obtenido en el valor de la posición a lo largo de la trayectoria en la comparación con los resultados del propagador <i>GMAT</i> , para una simulación sin perturbaciones.	49
35	Error absoluto obtenido en el valor de la velocidad a lo largo de la trayectoria en la comparación con los resultados del propagador <i>GMAT</i> , para una simulación sin perturbaciones.	50
36	Error relativo obtenido en el valor de la velocidad a lo largo de la trayectoria en la comparación con los resultados del propagador <i>GMAT</i> , para una simulación sin perturbaciones.	50
37	Error absoluto obtenido en el valor de la posición a lo largo de la trayectoria, en la comparación con los resultados del propagador <i>GMAT</i> , para una simulación considerando la gravedad de la Luna, Marte y Júpiter.	52
38	Error relativo obtenido en el valor de la posición a lo largo de la trayectoria en la comparación con los resultados del propagador <i>GMAT</i> , para una simulación considerando la gravedad de la Luna, Marte y Júpiter.	52
39	Error absoluto obtenido en el valor de la velocidad a lo largo de la trayectoria en la comparación con los resultados del propagador <i>GMAT</i> , para una simulación considerando la gravedad de la Luna, Marte y Júpiter.	53
40	Error relativo obtenido en el valor de la velocidad a lo largo de la trayectoria en la comparación con los resultados del propagador <i>GMAT</i> , para una simulación considerando la gravedad de la Luna, Marte y Júpiter.	53
41	Error absoluto obtenido en el valor de la posición a lo largo de la trayectoria, en la comparación con los resultados del propagador <i>GMAT</i> , para una simulación considerando la presión de radiación solar.	54
42	Error relativo obtenido en el valor de la posición a lo largo de la trayectoria en la comparación con los resultados del propagador <i>GMAT</i> , para una simulación considerando la presión de radiación solar.	55

43	Error absoluto obtenido en el valor de la velocidad a lo largo de la trayectoria en la comparación con los resultados del propagador <i>GMAT</i> , para una simulación considerando la presión de radiación solar.	55
44	Error relativo obtenido en el valor de la velocidad a lo largo de la trayectoria en la comparación con los resultados del propagador <i>GMAT</i> , para una simulación considerando la presión de radiación solar.	56
45	Error absoluto obtenido en el valor de la posición a lo largo de la trayectoria, en la comparación con los resultados del propagador <i>GMAT</i> , para una simulación considerando los armónicos zonales.	57
46	Error relativo obtenido en el valor de la posición a lo largo de la trayectoria en la comparación con los resultados del propagador <i>GMAT</i> , para una simulación considerando los armónicos zonales.	57
47	Error absoluto obtenido en el valor de la velocidad a lo largo de la trayectoria en la comparación con los resultados del propagador <i>GMAT</i> , para una simulación considerando los armónicos zonales.	58
48	Error relativo obtenido en el valor de la velocidad a lo largo de la trayectoria en la comparación con los resultados del propagador <i>GMAT</i> , para una simulación considerando los armónicos zonales.	58
49	Error absoluto obtenido en el valor de la posición a lo largo de la trayectoria, en la comparación con los resultados del propagador <i>GMAT</i> , para una simulación considerando la resistencia atmosférica.	59
50	Error relativo obtenido en el valor de la posición a lo largo de la trayectoria en la comparación con los resultados del propagador <i>GMAT</i> , para una simulación considerando la resistencia atmosférica.	60
51	Error absoluto obtenido en el valor de la velocidad a lo largo de la trayectoria en la comparación con los resultados del propagador <i>GMAT</i> , para una simulación considerando la resistencia atmosférica.	60
52	Error relativo obtenido en el valor de la velocidad a lo largo de la trayectoria en la comparación con los resultados del propagador <i>GMAT</i> , para una simulación considerando la resistencia atmosférica.	61
53	Error absoluto obtenido en el valor de la posición a lo largo de la trayectoria, en la comparación con los resultados del propagador <i>GMAT</i> , para una simulación considerando todas las perturbaciones estudiadas.	62
54	Error relativo obtenido en el valor de la posición a lo largo de la trayectoria en la comparación con los resultados del propagador <i>GMAT</i> , para una simulación considerando todas las perturbaciones estudiadas.	62
55	Error absoluto obtenido en el valor de la velocidad a lo largo de la trayectoria en la comparación con los resultados del propagador <i>GMAT</i> , para una simulación considerando todas las perturbaciones estudiadas.	63

56	Error relativo obtenido en el valor de la velocidad a lo largo de la trayectoria en la comparación con los resultados del propagador <i>GMAT</i> , para una simulación considerando todas las perturbaciones estudiadas.	63
57	Satélite Terra lanzado en 1999 por la NASA [18].	64
58	Datos en la aplicación para realizar la propagación de la trayectoria del satélite Terra.	65
59	Trayectoria del satélite Terra desde el 1 de marzo al 1 de mayo de 2020.	66
60	Error absoluto obtenido en el valor de la posición a lo largo de la trayectoria propagada del satélite Terra.	67
61	Error relativo obtenido en el valor de la posición a lo largo de la trayectoria propagada del satélite Terra.	67
62	Error absoluto obtenido en el valor de la velocidad a lo largo de la trayectoria propagada del satélite Terra.	68
63	Error relativo obtenido en el valor de la velocidad a lo largo de la trayectoria propagada del satélite Terra.	68
64	Datos en la aplicación para realizar el cambio de altura del perigeo.	70
65	Cambio de altura en el perigeo.	70
66	Cambio de altura en el perigeo (planta).	71
67	Datos en la aplicación para realizar el cambio de altura del apogeo.	72
68	Cambio de altura en el apogeo.	73
69	Cambio de altura en el apogeo (planta).	73
70	Mars2020 en la prueba de preparación para el vacío térmico [20]	74
71	Trayectoria de la Mars2020 lanzada por la NASA [20].	75
72	Datos de la trayectoria de la Mars2020 proporcionados por Horizons. [7]	75
73	Datos de la trayectoria de la Mars2020 que se incluirán en la aplicación.[Horizons]	76
74	Datos en la aplicación para simular la órbita de la Mars2020.	76
75	Error absoluto obtenido en el valor de la posición a lo largo de la trayectoria propagada de la Mars2020.	77
76	Error relativo obtenido en el valor de la posición a lo largo de la trayectoria propagada de la Mars2020.	78
77	Error absoluto obtenido en el valor de la velocidad a lo largo de la trayectoria propagada de la Mars2020.	79
78	Error relativo obtenido en el valor de la velocidad a lo largo de la trayectoria propagada de la Mars2020.	79

Índice de cuadros

- 1 Presupuestos para los recursos humanos del proyecto. 171
- 2 Presupuesto para los recursos de software utilizados a lo largo del proyecto. . . . 171
- 3 Presupuesto para los recursos informáticos necesarios para el desarrollo del proyecto. 171
- 4 Presupuesto total para el desarrollo del proyecto. 172

Nomenclatura

Lista de Abreviaturas

AGIS	<i>Astrometric Global Iterative Solution</i>
ECI	<i>Earth-Centered Inertial</i> . Sistema Inercial Geocéntrico
GMAT	<i>General Mission Analysis Tool</i>
ICBM	Misiles Balísticos Intercontinentales
LEO	<i>Low Earth Orbit</i>
MSISE-90	<i>(Mass Spectrometer - Incoherent Scatter</i>
NASA	<i>National Aeronautics and Space Administration</i>
STK	<i>System Tool Kit</i>
VOP	Variación de Parámetros

Lista de Símbolos

$\bar{\Delta}$	Vector gradiente. Operador nabla
\bar{r}	Derivada del vector posición. Vector velocidad
\bar{a}_{atm}	Vector aceleración causado por la perturbación debida a la resistencia atmosférica
\bar{a}_{Harm}	Vector aceleración causado por la perturbación debida al potencial gravitatorio terrestre.
\bar{a}_{J_2}	Vector aceleración causado por el armónico zonal de orden 2 (J_2)
\bar{a}_{J_3}	Vector aceleración causado por el armónico zonal de orden 3 (J_3)
\bar{a}_{J_4}	Vector aceleración causado por el armónico zonal de orden 4 (J_4)
\bar{a}_{J_5}	Vector aceleración causado por el armónico zonal de orden 5 (J_5)
\bar{a}_{J_6}	Vector aceleración causado por el armónico zonal de orden 6 (J_6)
$\bar{a}_{perturbed}$	Aceleración causada por la perturbación
\bar{a}_{plan}	Vector aceleración causado por la perturbación debida a la gravedad de un tercer cuerpo

\bar{B}	Vector de Laplace
\bar{e}	Vector excentricidad
\bar{e}_v	Vector unitario que describe la dirección de la aceleración causada por la resistencia atmosférica
\bar{F}_g	Fuerza de atracción gravitatoria entre dos cuerpos
\bar{h}	Vector del momento angular específico
\bar{I}	Vector unitario en la dirección i
\bar{K}	Vector unitario en la dirección k
\bar{n}	Vector unitario en la dirección del nodo ascendente
\bar{r}	Vector posición
\bar{r}_{Sol}	Vector posición del Sol
$\bar{r}_{bodyShadow}$	Vector posición del cuerpo que puede generar la sombra
\bar{r}_{body}	Vector posición
\bar{s}	Vector distancia ente cuerpo estudiado y cuerpo que puede generar la sombra
\bar{v}	Vector velocidad
\bar{X}	Vector de estado
Δm	Consumo de propelente
Δm	Porción de masa de la atmósfera que choca contra una sección A del satélite en un intervalo de tiempo
Δp	Impulso ejercicio sobre el satélite por Δm
Δt	Intervalo de tiempo
Δv	Cambio en magnitud y dirección del vector velocidad
Δv_A	Cambio de velocidad en el punto A de la transferencia de Hohmann
Δv_B	Cambio de velocidad en el punto B de la transferencia de Hohmann
Δv_{total}	Cambo de velocidad total en la transferencia de Hohmann
$\dot{\bar{X}}$	Derivada del vector de estado
$\dot{\theta}$	Derivada de la anomalía verdadera

\dot{x}	Derivada del valor de la coordenada x del vector posición. Valor de la coordenada x del vector velocidad
\dot{y}	Derivada del valor de la coordenada y del vector posición. Valor de la coordenada y del vector velocidad
\dot{z}	Derivada del valor de la coordenada z del vector posición. Valor de la coordenada z del vector velocidad
\hat{u}_h	Vector unitario en la dirección del momento angular específico
\hat{u}_r	Vector unitario en la dirección radial
\hat{u}_T	Vector unitario en la dirección tangencial
\hat{u}_x	Vector unitario en la dirección de la coordenada x
\hat{u}_y	Vector unitario en la dirección de la coordenada y
\hat{u}_z	Vector unitario en la dirección de la coordenada z
$\Lambda_{n,m}$	Constante del modelo para obtener el potencial gravitatorio terrestre
Λ_{sat}	Longitud geocéntrica del satélite
μ	Parámetro gravitacional estándar
Ω	Ascensión Recta del Nodo Ascendente
ω	Argumento del perigeo
ω_{\oplus}	Velocidad angular con la que rota la Tierra
ϕ_{sat}	Latitud geocéntrica del satélite
ρ	Densidad del aire a una cierta altura
ρ_0	Densidad de referencia
ρ_Q	Distancia desde el objeto al punto estudiado
θ	Anomalía verdadera
ε	Energía mecánica específica
A	Expresión para el cálculo de la función S
A	Punto en el que se realiza el primer impulso de la transferencia de Hohmann
A	Sección del satélite contra la que choca una masa de aire de la atmósfera
A	Área que se encuentra expuesta a la radiación solar

a	Coeficiente para la obtención de la función S
a	Semieje mayor de la órbita
A/m	Relación área-masa
a_{p_x}	Coordenada x de la aceleración perturbadora
a_{p_y}	Coordenada y de la aceleración perturbadora
a_{p_z}	Coordenada z de la aceleración perturbadora
B	Punto en el que se realiza el segundo impulso de la transferencia de Hohmann
b	Coeficiente para la obtención de la función S
b	Semieje menor de la órbita
c	Coeficiente para la obtención de la función S
c	Distancia al foco
c	Velocidad de la luz
C_D	Coeficiente de resistencia aerodinámica
C_e	Velocidad de eyección característica o efectiva
C_R	Coeficiente de reflectividad
$C_{n,m}$	Constante del modelo para obtener el potencial gravitatorio terrestre
E	Anomalía excéntrica
E	Energía relativa
e	Excentricidad de la órbita
F_{SRP}	Fuerza generada por la radiación solar
G	Constante de gravitación universal
g_0	Aceleración de la gravedad a nivel del mar
H	Altitud de escala
h	Altura del satélite
h	Momento angular específico
h_0	Altura de referencia

h_{ellp}	Altura del satélite
i	Inclinación
I_{sp}	Impulso específico de los motores
J_2	<i>Zonal Harmonic</i> de orden 2
J_3	<i>Zonal Harmonic</i> de orden 3
J_4	<i>Zonal Harmonic</i> de orden 4
J_5	<i>Zonal Harmonic</i> de orden 5
J_6	<i>Zonal Harmonic</i> de orden 6
J_n	<i>Zonal Harmonic</i>
$J_{n,m}$	Constante del modelo para obtener el potencial gravitatorio terrestre
M	Anomalía media
M	Masa del tercer cuerpo
m	Masa del cuerpo atraído por un cuerpo central
m	Masa del satélite
m	Orden que define las funciones de Legendre en el modelo para obtener el potencial gravitatorio terrestre
m_f	Masa del cohete tras el apagado de los motores
m_i	Masa del cohete antes del encendido de los motores
m_Q	Masa del punto estudiado
M_{\oplus}	Masa del cuerpo central
mc	Momento. Presión solar
n	Orden que define los polinomios y funciones de Legendre en el modelo para obtener el potencial gravitatorio terrestre
P_n	Polinomio de Legendre de grado n
P_n	<i>Zonal Harmonic</i>
$P_{n,m}$	Función de Legendre de primer tipo, de orden n y m
p_{SRP}	Presión de radiación solar

Q	Puntos en los que se estudia el potencial gravitatorio terrestre
r	Posición
r_0	Vector posición inicial
r_a	Radio del apoapsis de la órbita
r_p	Radio del periapsis de la órbita
r_x	Coordenada x del vector posición
r_y	Coordenada y del vector posición
r_z	Coordenada z del vector posición
R_{\oplus}	Radio ecuatorial medio de la Tierra
r_{\oplus}	Posición del tercer cuerpo
$R_{bodyShadow}$	Radio del cuerpo que puede generar sombra
r_{sat}	Posición satélite
R_{Sol}	Radio del Sol
S	Función eclipse
$S_{n,m}$	Constante del modelo para obtener el potencial gravitatorio terrestre
SF	Constante de radiación solar. Flujo Solar
t	Instante de tiempo
U	Potencial gravitatorio por unidad de masa
u_0	Vector de estado inicial
v	Velocidad
v_0	Vector velocidad inicial
v_a	Velocidad en el apogeo
v_p	Velocidad en el perigeo
v_r	Velocidad radial
v_T	Velocidad tangencial
v_x	Coordenada x del vector velocidad

v_y	Coordenada y del vector velocidad
v_z	Coordenada z del vector velocidad
v_{rel}	Velocidad relativa
x	Coeficiente para el cálculo de la expresión A
x	Valor de la coordenada x del vector posición
y	Coeficiente para el cálculo de la expresión A
y	Valor de la coordenada y del vector posición
z	Valor de la coordenada z del vector posición

1 INTRODUCCIÓN

El número de misiones espaciales a aumentado a lo largo de los últimos años debido no solo a la mejora de las herramientas tecnológicas sino a un mayor interés por conocer los misterios del universo. Uno de los pasos claves en el diseño de estas misiones consiste en la creación de la trayectoria que tomará la nave o el satélite una vez situado en el espacio. Este diseño deberá de realizarse con la mayor exactitud posible, y teniendo en cuenta todos los posibles aspectos que afectarán a la trayectoria. Para ello se emplean métodos de propagación dentro de propagadores diseñados para obtener la trayectoria en función de los parámetros iniciales que se indiquen.

El objetivo de este trabajo consiste en la creación de una aplicación que se comporte como dichos propagadores, es decir, que a partir de ciertos datos iniciales permita obtener la trayectoria completa de una nave o satélite. Para ello se podrán emplear dos sistemas de referencia distintos, geocéntrico y heliocéntrico, y distintas técnicas de propagación, el problema de dos cuerpos, y el método de Cowell. Además con este último se podrán incluir distintas perturbaciones causadas por: la gravedad de los planetas, el Sol y la Luna, la presión de radiación solar, la no uniformidad del campo gravitatorio terrestre y la atmósfera terrestre.

Además, en esta aplicación se incluirán la opción de implementar impulsos en la trayectoria generada que permitirán corregir o cambiar de órbita para poder asegurar el éxito de la misión espacial diseñada.

Con todo esto se estudiarán casos reales de misiones y se observarán los errores obtenidos comparando los resultados con los valores reales del vector de estado, para poder ver las posibilidades de aplicación que tendrá el programa diseñado.

El trabajo se dividirá en distintos capítulos que tratarán distintos aspectos de todo lo estudiado en el desarrollo del mismo:

- Capítulo 2 - Estado del arte

Se comienza el trabajo a partir de un estudio previo de la historia sobre las misiones espaciales. Continuando con la explicación acerca del análisis y diseño de las mismas. Para terminar introduciendo los propagadores orbitales, su utilidad y presentando alguno de ellos.

- Capítulo 3 - Fundamentos teóricos

En este capítulo se explican las bases teóricas para el estudio de órbitas y trayectorias espaciales. Se comienza con una explicación de los distintos sistemas de referencia utilizados, para continuar con la explicación del vector de estado y sus distintas formas. A continuación se expone uno de los conceptos clave de la mecánica orbital, la ecuación del movimiento, y algunas técnicas de propagación a las que da lugar. Lo siguiente será la descripción de las distintas perturbaciones empleadas en la propagación. Para finalizar con la explicación de las maniobras orbitales.

- Capítulo 4 - Aplicación en MATLAB

En este apartado se explicarán los conceptos necesarios para entender lo que es una aplicación de MATLAB y cómo se crea. Además se explicarán algunas de las funciones que se crearon en dicha aplicación para el correcto funcionamiento de la misma.

- Capítulo 5 - Programa

Aquí se explicará la estructura del programa de cálculo creado, en el cual se emplean funciones de la aplicación y de MATLAB, para el tratamiento de datos y la integración.

- Capítulo 6 - Validación

En este capítulo se pretenden validar los distintos métodos de integración creados, por lo que se compararán los valores obtenidos con los valores calculados por un propagador (GMAT). Se comienza con la validación del problema de dos cuerpos, para pasar al método de Cowell con las distintas perturbaciones de manera individual y, finalmente, en conjunto.

- Capítulo 7 - Resultados

Tras la comprobación de los integradores, se muestran los resultados obtenidos de la propagación de casos reales y la generación de casos propios, en el caso del cambio de altura del apogeo y del perigeo. En los casos en los que se propagan misiones reales se realiza, además, una comparativa entre los vectores de estado obtenidos a lo largo de la trayectoria y se presentan los errores al compararlos con valores reales.

- Capítulo 8 - Conclusiones y futuras líneas de estudio

Finalmente se resumen los resultados obtenidos en los capítulos anteriores (resultados y validación), y se presentan posibles líneas de estudio futuras que puedan implementarse posteriormente a la aplicación desarrollada.

- Anexos

Aquí se presentan los códigos tanto de la aplicación como de MATLAB y finalmente, un presupuesto del desarrollo del proyecto.

2 ESTADO DEL ARTE

El interés del ser humano por el cielo y el universo se remonta hasta épocas antes de Cristo. Por lo que era de esperar que a medida que la humanidad mejorara sus tecnologías y materiales, estos se pudieran emplear para realizar misiones encargadas de salir fuera del planeta Tierra, es decir, misiones espaciales.

2.1 Historia de las misiones espaciales

Fue en la segunda mitad del siglo XIX cuando comenzó el desarrollo de cohetes que superaban la fuerza de la gravedad y alcanzaban velocidades orbitales. Aunque no fue hasta los años 30 y 40, cuando nace la idea de utilizar dichos cohetes como armas de larga distancia. Por ello, a finales de la Segunda Guerra Mundial, la Alemania Nazi atacó a Londres con misiles V2 de más de 300 km de alcance, y que formaba un arco de casi 100 km de altura a más de 5600 km por hora [1], [2].

Tras la segunda Guerra Mundial, en el periodo conocido como Guerra Fría, Estados Unidos y la Unión Soviética compitieron por el acceso a la tecnología del V2, creando sus propios programas de misiles balísticos intercontinentales (ICBM). Y aunque la mayoría de científicos encargados del diseño y creación del misil V2 desertaron a EEUU, fueron los soviéticos quienes llevaban ventaja en esta investigación gracias al apropiamiento de algunas piezas y diseños del V2 y al trabajo realizado por Tsiolkovsky con respecto a los cohetes.

Finalmente, fue el diseñador Sergei Korolev quien desarrolló el primer ICBM, llamado R7, y basado en el diseño del V2, que inició lo que se denominaría carrera espacial. Esta competición llegó a su punto más crítico el 4 de octubre de 1957 con el lanzamiento del satélite **Sputnik**, impulsado por el cohete R7, que emitía ondas sonoras desde un transmisor de radio (imagen 1).



Figura 1: Satélite **Sputnik**. Primer satélite de la historia [3].

Una vez en el espacio la órbita del Sputnik tenía un periodo de 96 minutos, que permitió que todo el mundo pudiera escuchar las ondas que emitía, demostrando así la mayor capacidad tecnológica de la URSS frente a las estadounidense. Pero dicha demostración de conocimientos espaciales no se quedó ahí. Un mes más tarde, en noviembre de 1957, se lanzó el **Sputnik II** que transportó al primer ser vivo al espacio, la perra Laika.

Por otra parte, antes del lanzamiento de **Sputnik**, EEUU realizó dos intentos fallidos de lanzamientos de satélites al espacio antes de conseguir lanzar el cohete que transportaba el satélite **Explorer** el 31 de enero de 1958. Esto fue, en gran parte, gracias a los conocimientos proporcionados por los ingenieros alemanes que habían convertido el cohete V2 en uno más potente, el Júpiter C o Juno. A diferencia del **Sputnik**, este satélite llevaba varios instrumentos con el fin de realizar distintos experimentos científicos, como un contador para detectar los rayos cósmicos, que fue utilizado para la investigación que demostró la existencia de los cinturones de radiación de Van Allen, tan importantes a la hora de diseñar misiones espaciales alrededor de la Tierra.

En cuanto a las misiones tripuladas, fue la Unión Soviética la primera en conseguir realizar un viaje espacial tripulado en 1961, mediante el satélite **Vostok 1**, convirtiendo al ruso Yuri Gagarin en el primer ser humano en orbitar la Tierra, y durando su vuelo 108 minutos y llegando a una altitud de 327 km. EEUU envió en el mismo año a Alan Shepard al espacio en un vuelo suborbital, y en 1962 John Glenn se convirtió en el primer estadounidense en orbitar la Tierra.

Además de intentar conseguir que el ser humano llegara al espacio, ambas potencias se centraron en otros objetivos. En 1959 la URSS ya había conseguido que llegara a la Luna el primer objeto fabricado por los humanos, el **Luna 2**. También consiguió que se produjera el primer paseo espacial y con la misión **Vostok 6**, Valentina Tereshkova se convirtió en la primera mujer en viajar al espacio (imagen 2).



Figura 2: Promera mujer en viajar al espacio, Valentina Tereshkova en el satélite **Vostok 6** [3].

A lo largo de la década de los 60, se realizaron varias misiones espaciales no tripuladas que fotografiaron y sondearon la Luna. Pero fue en 1961 cuando el presidente Kennedy marcó un nuevo objetivo "Aterrizar con un hombre en la Luna y devolverlo sano y salvo a la Tierra en el plazo

de una década". Por lo que la NASA creó el programa Proyecto Géminis, en el que se realizaban distintas pruebas a los astronautas para estudiar su capacidad para soportar un vuelo espacial de varios días. De este proyecto se pasó al más conocido Proyecto Apollo que con el **Apollo 11**, en 1969, consiguió que el ser humano pisara la Luna, siendo Neil Armstrong el primer astronauta en pisarla, dando *una gran salto para la humanidad* (imagen 3). Además de esta misión entre 1969 y 1972 se realizaron seis misiones dentro de este proyecto para explorar la Luna.

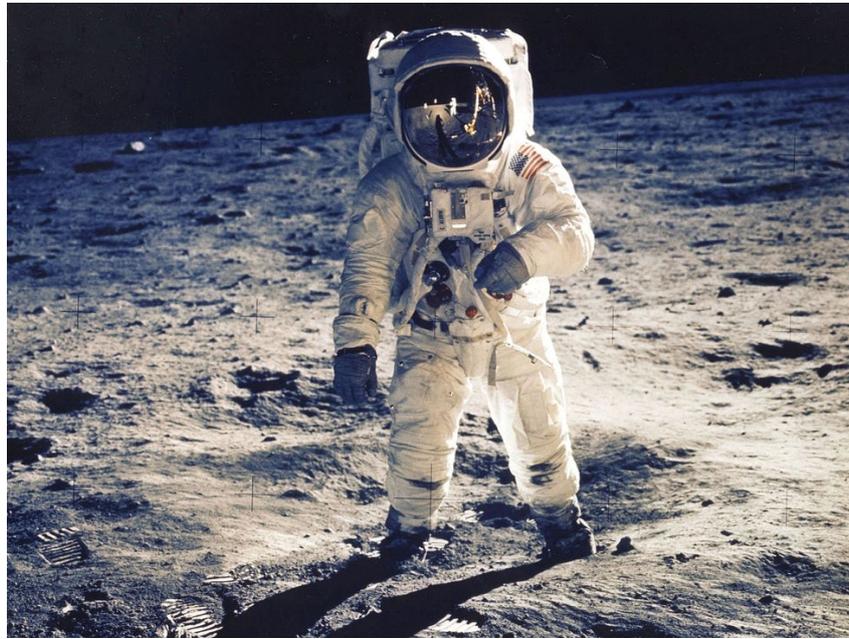


Figura 3: Misión **Apollo 11**. Primera vez que humanos llegaban a la superficie de la Luna [3].

A principios de la década de los 70 era ya normal la creación y utilización de satélites de comunicaciones y navegación. Además la serie de sondas espaciales **Mariner** se encargaban de cartografiar la superficie de Mercurio, Venus y Marte. Y a finales de esta década fue la nave **Voyager** la que se encargó de obtener imágenes detalladas de Júpiter, Saturno, sus anillos y sus lunas.

La siguiente fase en las exploración espacial comienza con la creación de estaciones espaciales. La primer estación espacial en órbita terrestre fue lanzada en 1971 y fue la estación soviética **Salyut 1**. La NASA también creó su estación espacial llamada **Skylab** donde se realizaban estudios de la Tierra y de los efectos de los vuelos espaciales en el cuerpo humano, por parte de astronautas y científicos.

Después de todas estas misiones realizadas de manera separada entre los dos países, se llevo a cabo la primera misión espacial con tripulación estadounidense y rusa, el proyecto de prueba **Apolo Soyuz**.

Por otra parte, en la década de los 80, en las órbitas terrestres comenzaron a aparecer más satélites de comunicación para la transmisión de programas de televisión y la captación de señales. Además también han ayudado a conocer más datos sobre el cambio climático, descubriendo un agujero de ozono sobre la Antártida. Permiten además, localizar incendios forestales y realizar fotografías para entender el alcance de catástrofes como el desastre de la central nuclear de Chernóbil.

El 24 de abril de 1990 se lanzó el observatorio orbital más avanzado jamás construido, **El Telescopio Espacial Hubble**. Este ha tomado imágenes de las galaxias más lejanas jamás vistas, de supernovas y de planetas y sistemas planetarios vecinos (imagen 4). Este tipo de misiones ayudan a comprender mejor el universo, y nuestro lugar en él.

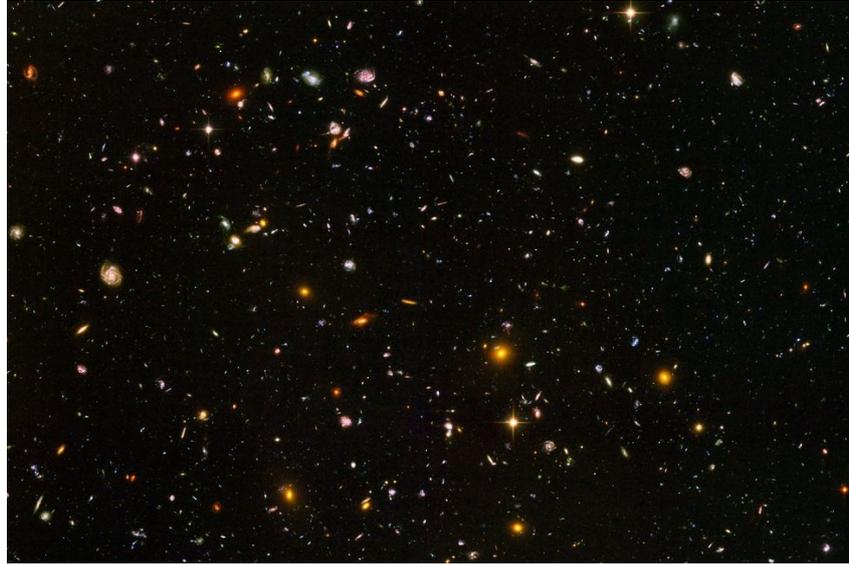


Figura 4: Foto de estrellas lejanas obtenida por el Telescopio Espacial **Hubble** [3].

Actualmente existen varios países que realizan distintas investigaciones en la **Estación Espacial Internacional**, proyecto formado por diferentes socios que contribuyen a su diseño y construcción, que simboliza la cooperación entre los distintos países, incluso aquellos que anteriormente eran competidores. Esta estación espacial comenzó a utilizarse en el año 2000, y supuso un cambio en la forma de entender las misiones espaciales (imagen 5).

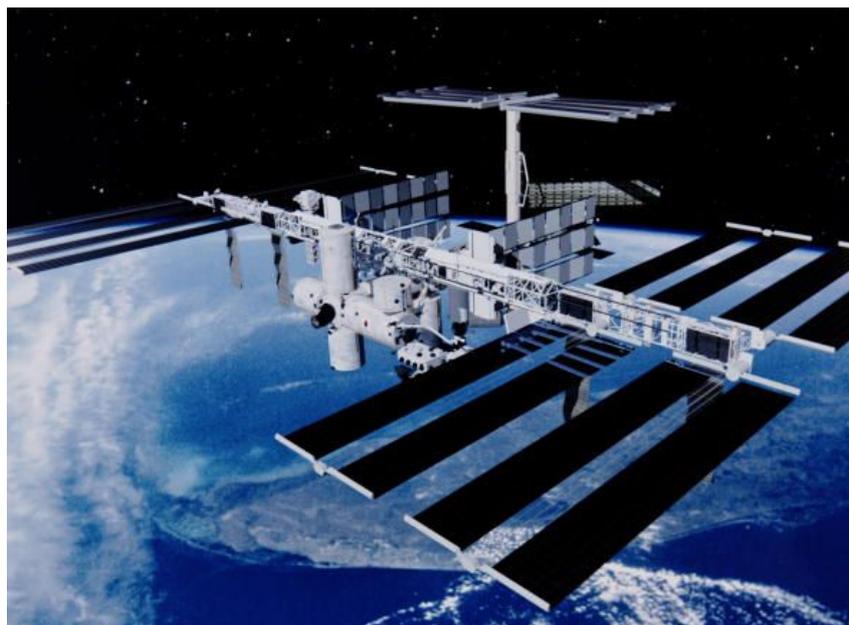


Figura 5: Estación Espacial Internacional [3].

Por otra parte, existe un gran número de misiones espaciales que se utilizan no para el estudio de la Tierra, sino para conocer más sobre el universo en el que nos encontramos. Todas estas misiones se realizan con sondas, que permiten descubrir nuevos datos como que las lunas Europa y Enceladus de Júpiter y Saturno, respectivamente, tienen océanos bajo su superficie de hielo, en la que es posible que exista vida.

En cuanto al estudio de planetas, actualmente el planeta al que se destinan muchas de las misiones espaciales es Marte. La relativa cercanía de Marte a la Tierra permite enviar sondas y naves para su estudio. Dos de las misiones más destacadas, debido a la cantidad de datos de la superficie del planeta que permiten recoger son la misión del **Rover Curiosity** y la de la **Mars2020** que permitió llevar el **Rover Perseverance** y el **Mars Helicopter** a la superficie del planeta rojo. Mientras que la primera se lanzó en 2012, la Mars2020 como su nombre indica se lanzó en el año 2020, llegando a la superficie de Marte en febrero de 2021 [20].

2.2 Diseño de misiones espaciales

Todas las misiones mencionadas anteriormente se han diseñado y analizado mediante un proceso común. Ya que, aunque cada una de ellas tenga unos requisitos propios, es necesario seguir un esquema de diseño común.

Cabe destacar que este proceso de diseño y análisis es un proceso iterativo en el cual se van redefiniendo tanto los requerimientos de la misión y los métodos para conseguir cumplirlos. Por lo que el proceso explicado a continuación se repetirá si en alguno de los puntos es necesario cambiar alguna de esas dos cosas [4].

1. Definición objetivos

(a) Definición de los objetivos generales y las limitaciones de la misión

En primer lugar será necesario definir de manera clara qué es lo que se quiere conseguir con la misión a definir y por qué es necesario. Esta información vendrá especificada en el documento conocido como *mission statement of*. A lo largo del diseño de la misión este punto será el que marcará si se está cumpliendo lo que se ha propuesto en un primer momento o no. Si no se cumpliera sería necesario rehacer el proceso de diseño, ya sea redefiniendo este punto y modificando los puntos posteriores a él, o simplemente corrigiendo las diferencias desde el primer punto en el que se produjeran.

(b) Estimación cuantitativa de los requisitos y necesidades de la misión

En segundo lugar, se deberá cuantificar hasta qué punto se quieren alcanzar los objetivos generales, dadas las necesidades de la misión, la tecnología aplicable, y, uno de los aspectos más importantes en las misiones espaciales, el coste económico de la misión. Esta primera estimación puede variar y negociarse a medida que el diseño de la misión avanza, ya que si se fijan demasiado pronto es posible que cualquier error en la estimación inicial implique un sobrecoste excesivo que impida el futuro desarrollo de la misión. Por lo que estos requisitos se apoyarán en los datos obtenidos a lo largo del diseño y análisis de la misión, para tomar un valor u otro.

2. Definición las características de la misión

A continuación, lo siguiente que se realizará será definir y caracterizar una misión espacial que cumpla con los objetivos predefinidos anteriormente.

(a) Definición alternativa de los conceptos de la misión

En este punto, el primer paso será desarrollar los conceptos alternativos de la misión. Un concepto de misión o de operaciones es una declaración completa de cómo funcionará la misión una vez se desarrolle. En él deberían incluirse aspectos como la forma en la que se obtendrán y enviarán los datos, la manera en la que se controlará la misión y el calendario que seguirá la misma. Aquí comienza a realizarse el estudio de la trayectoria que seguirá la misión, que se desarrollará con exactitud más adelante.

(b) Definición alternativa de la arquitectura de la misión

A continuación se definen las posibles arquitecturas de la misión, es decir, las combinaciones posibles de los elementos de la misión que puedan cumplir los requisitos de la misma. En la imagen 6 se observan los distintos elementos que pueden formar la misión, que se deberán de definir a la hora de crear la arquitectura de la misión. En este paso se deberán de buscar distintas alternativas de definir cada uno de los elementos mostrados en esta figura, que permitan cumplir los objetivos de la misión.

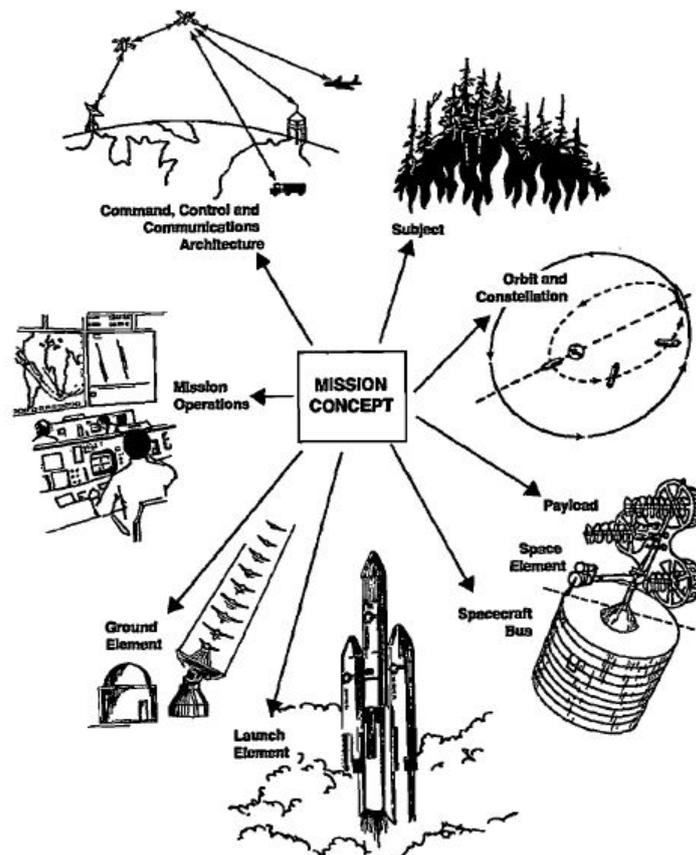


Figura 6: Arquitectura de la misión formada por los distintos elementos [4].

(c) Identificación de los principales factores del sistema

Este punto se basa en la búsqueda de los principales factores que pueden afectar significativamente al coste o al rendimiento de la misión. Entre estos factores se suelen incluir el número de satélites, la altitud de la órbita, la potencia necesaria, y el tamaño y peso de los distintos instrumentos que se llevarán en la misión.

Todo esto se suele realizar debido a que el hecho de identificar estos factores permitirá enfocar los esfuerzos en los parámetros más influyentes en el diseño y coste de la misión, permitiendo así poder cumplir los objetivos de la mejor forma posible sin excederse del presupuesto estipulado.

(d) Caracterización de los conceptos y arquitectura de la misión

Este paso es el más complicado dentro del diseño de la misión ya que en él se definen con detalle todo lo que hará la misión. Se determina el presupuesto de potencia, peso y procesamiento de datos. Además, es el paso más costoso debido a la gran cantidad de personas que son necesarias para la correcta caracterización de la misión. El hecho de caracterizar detalladamente todos los elementos de la misión permite gestionar correctamente el proceso futuro, aunque se debe tener claro que lo realmente importante a la hora de definir detalladamente la misión serán sus objetivos y factores anteriormente concretados.

Es en este punto en el que se desarrolla el completo estudio de la trayectoria de la misión, mediante la propagación de la misma. Esto es necesario para conocer la potencia y peso de la misión final, entre otras cosas. Cabe destacar que este desarrollo parte de los datos predefinidos en los puntos a) y b).

3. Evaluación de la misión

Una vez definidos y caracterizados los conceptos alternativos de la misión, se evalúan los sistemas definidos.

(a) Identificar los requerimientos críticos de la misión

Para realizar dicha evaluación, el primer paso consiste en identificar los requisitos críticos o clave, para determinar el coste y la complejidad de la misión. Tal y como se ha comentado anteriormente, los factores del sistema, previamente definidos, tienen gran influencia en el coste de la misión, pero estos no suelen convertirse en requerimientos del sistema. Por lo que se deberá de mantener una relación entre estos factores y los requerimientos de la misión, ya que serán importantes a lo largo del diseño y desarrollo de la misma. Cabe destacar que para los conceptos de misión alternativos pueden crearse distintos requisitos críticos.

(b) Evaluar la utilidad de la misión

A partir de todos los datos definidos anteriormente, se realizará el análisis de utilidad de la misión, en el cual se cuantificará el grado de cumplimiento de los distintos requisitos y objetivos de la misión en función del coste y de las opciones de diseño existentes. Es importante saber, que este análisis deberá de realizarse conjuntamente entre aquellas personas encargadas del proceso de diseño de la misión y aquellos quienes vayan a utilizar los datos obtenidos de la misión, y, por tanto, probablemente proporcionen el dinero necesario para llevarla a cabo.

(c) Definir el concepto de la misión (*baseline*)

A continuación se pasa a la definición de uno o más diseños de referencia de la misión. Este diseño es una definición única de la misión que cumple todos o la mayoría de requisitos y objetivos de la misión. Esta definición consiste en un conjunto de valores de todos los parámetros del sistema que en conjunto permiten el funcionamiento correcto de la misión.

Para realizar dicho diseño se emplea la *baseline*, ya que será necesario redefinir y modificar muchos parámetros simultáneamente. La *baseline* permite limitar el número de opciones a evaluar, ya que en lugar de examinar todas las

combinaciones posibles, se estudia el impacto de variar cada uno de los parámetros individualmente, dentro de los diseños de referencia creados. A medida que se van consiguiendo valores óptimos la *baseline* se asemeja al diseño final del sistema. Sin embargo, es necesario destacar que la *baseline* es solo el punto de partida del diseño necesaria para la obtener dichos valores, pero no es una definición final de los parámetros de la misión.

4. Definición de los requerimientos

(a) Definir los requerimientos del sistema

Para poder desarrollar y construir al completo la misión es necesario proporcionar a las personas encargadas de la construcción de cada sistema, los requisitos específicos de cada uno de ellos, obtenidos a partir de los objetivos y limitaciones de la misión.

(b) Asignar los distintos requerimientos a los elementos del sistema

Por último se desglosan los distintos requisitos específicos en los distintos componentes de la misión, y se asigna un presupuesto para cada componente en función del peso y la potencia. En la lista final de requisitos se podrá comprobar como de bien se ha realizado el trabajo de análisis, diseño y asignación.

2.3 Propagación de la trayectoria

Como se ha visto en el apartado anterior uno de los puntos claves a la hora de diseñar una misión espacial, consiste en el estudio de la trayectoria de la nave, ya que influirá en gran parte de los requerimientos de dicha misión. Pero el estudio de la trayectoria de cuerpos en el espacio no comenzó con las misiones espaciales, sino que siempre ha sido uno de los problemas más interesante en astrodinámica [11].

El inicio de la solución de la propagación de órbitas en el espacio, comenzó con la mecánica clásica, con la que Johann Kepler descubrió que un cuerpo central puede obligar a otro a orbital sobre él con una trayectoria circular o elíptica, cuya explicación física fue más tarde resuelta por Sir Isaac Newton al descubrir como la fuerza de gravedad del cuerpo central atraía al otro cuerpo. Posteriormente, durante varios siglos, varios científicos avanzaron en el desarrollo de distintos modelados de las fuerzas que afectan a los cuerpos en el espacio, permitiendo así conocer mejor los efectos en los cuerpos en el espacio y, por tanto en sus órbitas.

Actualmente, con todos estos modelos de fuerzas ya conocidos se han desarrollado programas que permiten conocer la trayectoria de los cuerpos en el espacio, partiendo de unos datos iniciales, ya sean fechas, vectores de estado, o ambos. Estos programas emplean distintos propagadores para obtener la órbita del cuerpo estudiado. teniendo en cuenta las perturbaciones deseadas por el usuario.

A continuación se presentan dos de los programas más empleados en la actualidad:

- *Systems Tool Kit (STK)*

Este programa es una plataforma de AGIS (*AAstrometric Global Iterative Solution*), que permite crear escenarios para analizar y visualizar distintos sistemas de sectores como el aeroespacial, el de defensa o el de telecomunicaciones. Simula la misión espacial definida y permite obtener informes con datos y gráficos para su posterior análisis [5].

- *General Mission Analysis Tool (GMAT)*

Este programa es el único software empresarial, de código abierto, que permite el diseño, optimización y navegación de misiones espaciales. Desarrollado por colaboradores de la NASA, la industria privada y el sector público, permite simular misiones que van desde orbitar la Tierra a salir del sistema solar y más allá [6].

Además de estos programas, cabe destacar la existencia de una herramienta, desarrollada por la NASA, que obtiene el vector de estado de un gran número de cuerpos en el espacio, *Horizons*. Partiendo de algunos valores reales conocidos propaga el vector de estado del cuerpo y permite posteriormente descargar las efemérides que se deseen.

3 FUNDAMENTOS TEÓRICOS

Para posicionar un objeto en el espacio, más concretamente, para conocer su posición y velocidad dentro de la órbita que describe alrededor de un cuerpo central, es necesario el conocimiento de los conceptos principales de la mecánica orbital.

3.1 Sistemas de referencia

A la hora de definir la órbita de un cuerpo determinado, en primer lugar, será necesario definir el sistema de referencia en el que se está trabajando. Los sistemas de referencia se definen por un origen, un plano de referencia y una dirección principal. Además es necesario especificar el sentido en el que se encuentra la dirección positiva del sistema. Para este trabajo se han empleado sistemas de referencia que siguen el triedro a derechas, por lo que el sentido positivo del movimiento en ellos será hacia la derecha de los ejes.

En cuanto al origen, se pueden encontrar tres tipos diferentes: el centro del sistema, el centro de masas del sistema o baricentro, o el baricentro en un sistema no inercial en rotación o sinódico.

Cabe destacar la diferencia entre sistemas inerciales y no inerciales. Los primeros cumplen el principio de inercia, es decir, en ellos se debe actuar sobre el cuerpo estudiado para que este tenga una cierta aceleración. Por otra parte, los segundos no permiten la aplicación de esta ley. Los sistemas estudiados en este trabajo son sistemas inerciales. [8].

3.1.1 Sistemas Interplanetarios: Sistema Heliocéntrico

Para el estudio de órbitas y maniobras interplanetarias se emplea el sistema heliocéntrico, cuyo origen, tal y como su nombre indica, es el centro del Sol. En cuanto a su plano fundamental, este es el plano de la eclíptica, que forma 23.45° con el ecuador terrestre.

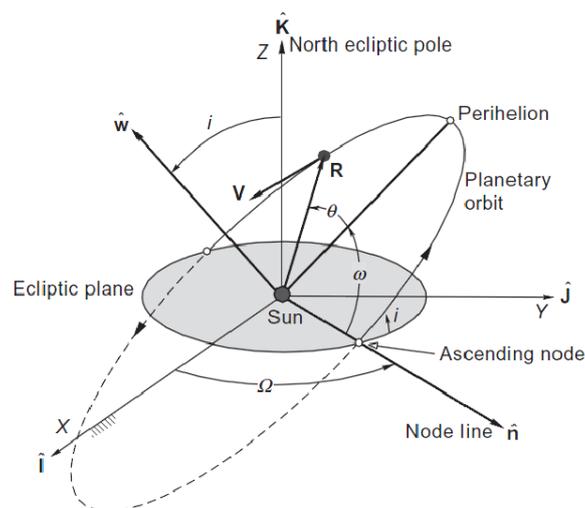


Figura 7: Sistema Heliocéntrico [15]

Por otra parte, su dirección principal es el eje \hat{X} que apunta, dentro del plano de la eclíptica, a la posición que tendría la Tierra en su órbita alrededor del Sol el primer día de primavera. El eje \hat{Y} se encuentra también en el plano de la eclíptica formando 90° con el eje \hat{X} . Y el eje \hat{Z} completa el triedro a derechas. (figura 7)

3.1.2 Sistema Inercial Geocéntrico

Este sistema es el principal sistema utilizado para el estudio de órbitas alrededor de la Tierra. Tiene su origen en el centro de la Tierra, su plano fundamental es el ecuador y su dirección principal viene dada por el primer punto de Aries, al cual apunta el eje \hat{X} . El eje \hat{Y} se encuentra también en el plano ecuatorial formando 90° con el eje \hat{X} . Y el eje \hat{Z} completa el triedro a derechas apuntando al norte. (figura 8)

Debido al movimiento del plano de referencia del sistema, este no puede considerarse completamente inercial y será imposible de fijar. Para poder utilizarlo se toma un sistema en un determinado instante de tiempo, y se especifica como transformar los vectores necesarios para situar el nuevo sistema. El sistema J2000 es una de las aproximaciones más exactas a un sistema inercial geocéntrico, este es también conocido como sistema ECI.

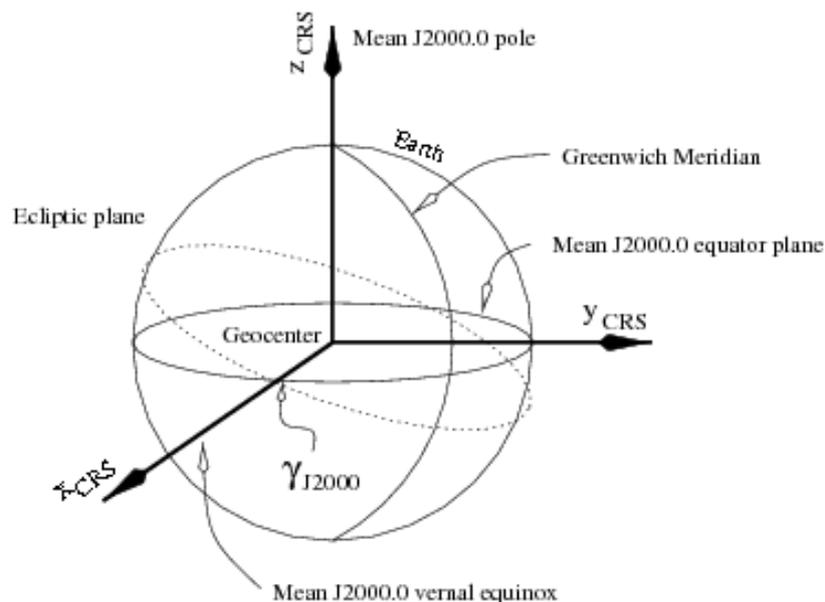


Figura 8: Sistema Geocéntrico [16]

3.2 Vector de estado

Uno de los conceptos clave necesario para describir la situación de un objeto en el espacio es el vector de estado. Este está compuesto por un conjunto de seis valores que permiten obtener la posición y velocidad del objeto, respecto a un sistema de referencia determinado.

Estos seis valores pueden ser coordenadas cartesianas que definen directamente la posición y velocidad del objeto, facilitando así la transformación entre distintos sistemas de referencia. También es posible formar el vector de estado con una serie de seis elementos (orbitales, equinocciales, etc).

3.2.1 Coordenadas cartesianas

En este caso, la posición y velocidad del cuerpo vendrán definidas por vectores (\vec{r} y \vec{v} respectivamente) de tres coordenadas:

$$\vec{r} = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} = x\vec{u}_x + y\vec{u}_y + z\vec{u}_z \quad \vec{v} = \dot{\vec{r}} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \dot{x}\vec{u}_x + \dot{y}\vec{u}_y + \dot{z}\vec{u}_z \quad (3.1)$$

3.2.2 Elementos orbitales

En la figura 9 se representan los elementos orbitales que permiten posicionar un cuerpo en el espacio, y conocer el tamaño y la orientación de la órbita que recorre.

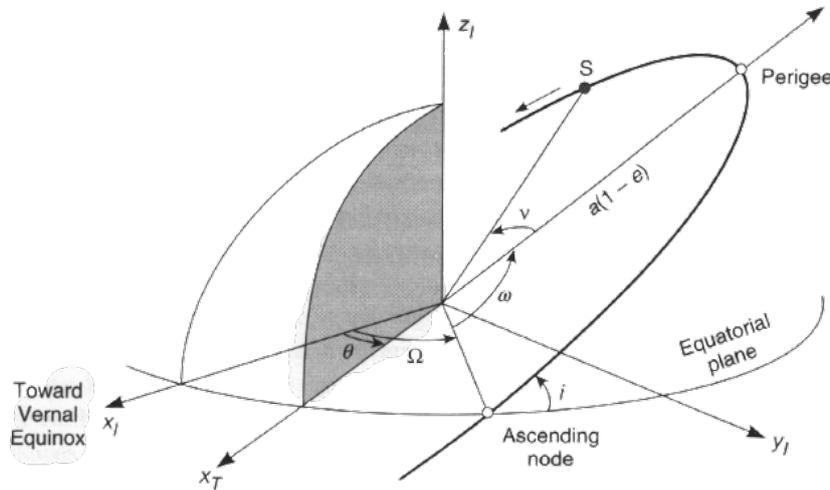


Figura 9: Representación de los elementos orbitales [17].

Antes de conocer dichos elementos, es interesante definir algunos parámetros de la órbita independientes de la masa del cuerpo que la recorre. Estos parámetros serán definidos a partir de las integrales de movimiento, obtenidas a partir de la ecuación de movimiento explicada posteriormente [8].

- Integral de la energía:

Esta integral asegura la conservación de la energía mecánica debido a que el campo gravitatorio es conservativo. A partir de ella se puede obtener una expresión sencilla que permite calcular la energía mecánica específica:

$$\varepsilon = \frac{v^2}{2} - \frac{\mu}{r} = cte \quad (3.2)$$

- Integral del área:

Esta integral define el momento angular específico como una variable constante dentro de la órbita, según la siguiente ecuación:

$$\vec{h} = \vec{r} \times \vec{v} = cte \quad (3.3)$$

Desarrollando la ecuación anterior se llega a una definición del momento angular a partir de la velocidad transversal del cuerpo:

$$\begin{aligned}\bar{h} &= \bar{r} \times \bar{v} = r\bar{u}_r \times (v_r\bar{u}_r + v_T\bar{u}_T) = rv_r(\bar{u}_r \times \bar{u}_r) + rv_T(\bar{u}_r \times \bar{u}_T) \\ \bar{h} &= rv_T(\bar{u}_r \times \bar{u}_T) = rv_T\bar{u}_h \\ |\bar{h}| &= h = rv_T\end{aligned}\quad (3.4)$$

En este desarrollo las variables u son vectores unitarios que indican la dirección de las variables a las que multiplican: posición y velocidad radial (\bar{u}_r), velocidad tangencial (\bar{u}_T) y momento angular específico (\bar{u}_h).

- Integral de Laplace

Esta integral define una constante \bar{B} , denominada vector de Laplace, que apunta al periapsis de la órbita y cuyo significado físico cobra sentido en la definición de la anomalía verdadera, ya que esta es el ángulo que forman el vector posición con el vector de Laplace.

La siguiente ecuación muestra la definición de la constante mencionada:

$$\bar{B} = \bar{v} \times \bar{h} - \mu \frac{\bar{r}}{r}\quad (3.5)$$

Por otra parte el módulo de dicho vector es:

$$|\bar{B}| = \sqrt{\mu + 2h^2\varepsilon}\quad (3.6)$$

En cuanto a los seis elementos orbitales que definen la órbita del cuerpo, a continuación se presentan cada uno de ellos:

- Semieje mayor (a)

Este parámetro define el tamaño de la órbita descrita por el cuerpo. En una órbita elíptica o circular tendrá un valor positivo, mientras que en una hiperbólica será negativo y en una parabólica infinito. Se estudiarán únicamente órbitas elípticas o, en su caso, circulares, por lo que el valor del semieje mayor siempre se corresponderá con la mitad de la distancia entre el apoapsis y el periapsis, tal y como se muestra a continuación:

$$a = \frac{r_a + r_p}{2}\quad (3.7)$$

donde r_a es el radio del apoapsis y r_p el radio del periapsis.

- Excentricidad (e)

Con este parámetro se define la forma de la órbita. En órbitas elípticas su valor se encuentra entre 0 y 1, siendo 0 en una órbita circular, 1 en una parabólica, y mayor que 1 en las hiperbólicas.

En la siguiente imagen se muestran como serían algunos de los parámetros de la órbita elíptica necesarios para obtener la excentricidad:

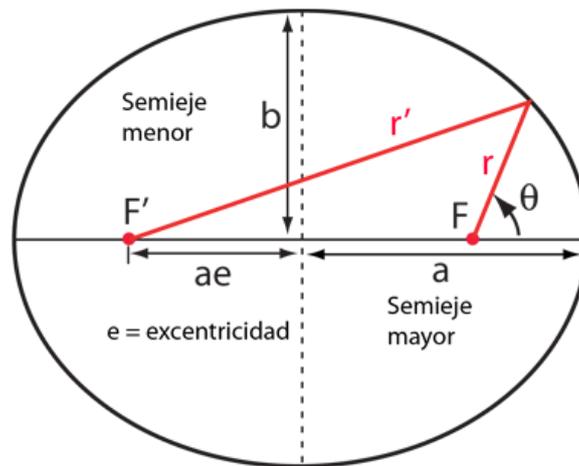


Figura 10: Elementos de una órbita elíptica

Normalmente, es posible calcular el valor de la excentricidad a partir de el semieje mayor (a) y la distancia al foco (c), obtenida a partir de el semieje mayor y el menor (b), o únicamente a partir de ambos semiejes:

$$b = \sqrt{a^2 - c^2} \quad e = \frac{c}{a} = \sqrt{1 - \left(\frac{b}{a}\right)^2} = \frac{\sqrt{a^2 - b^2}}{a} \quad (3.8)$$

Cabe destacar que, además, teniendo el valor del semieje mayor y de la excentricidad es posible calcular el momento angular específico sin necesidad de conocer los valores de posición y velocidad:

$$h = rv_T = r^2\dot{\theta} = \sqrt{\mu a (1 - e^2)} \quad (3.9)$$

siendo v_T la velocidad tangencial del cuerpo, y θ el valor de la anomalía verdadera, uno de los elementos orbitales que se explicará más adelante.

Por otra parte, también es interesante, y necesario, conocer el vector excentricidad, ya que permitirá definir el valor de algunos de los siguientes elementos orbitales.

$$\bar{e} = \frac{\bar{B}}{\mu} = \frac{\bar{v} \times \bar{h}}{\mu} - \frac{\bar{r}}{r} \quad (3.10)$$

- **Inclinación (i)**

Este parámetro indica el ángulo que forma el plano orbital con el plano de referencia. Este ángulo varía entre 0° y 180° , siendo mayor de 90° en una órbita retrógrada y menor si es posigrada. Si su valor es igual a 0° , la órbita pasará por el ecuador, mientras que si es 90° en el caso de la Tierra será una órbita polar.

Siendo el vector \bar{K} igual a $[0, 0, 1]$, la inclinación de la órbita se obtendrá a partir de él y del vector del momento angular, tal y como se muestra a continuación:

$$\cos i = \frac{\bar{h} \times \bar{K}}{|\bar{h}| |\bar{K}|} = \frac{\bar{h} \times \bar{K}}{h} \quad (3.11)$$

- Ascensión Recta del Nodo Ascendente (Ω)

Este parámetro es el ángulo medido sobre el ecuador, que forma el nodo ascendente, es decir, el punto del ecuador por el cual la órbita pasa del sur al norte con el punto Aries. En este caso, el valor del ángulo varía entre 0° y 360° , y se obtiene a partir de la siguiente ecuación:

$$\cos \Omega = \frac{\bar{I} \cdot \bar{n}}{|\bar{n}|} = \frac{\bar{I} \cdot (\bar{h} \times \bar{K})}{|\bar{h} \times \bar{K}|} \quad (3.12)$$

donde el vector \bar{I} es $[1, 0, 0]$, y \bar{n} es la dirección del nodo ascendente, y es igual a $\bar{h} \times \bar{K}$.

- Argumento del perigeo (ω)

Es el ángulo que forma el nodo ascendente con el perigeo, medido sobre el plano orbital. Al igual que la ascensión recta del nodo ascendente, el argumento del perigeo varía entre 0° y 360° .

$$\omega = \frac{\bar{n} \cdot \bar{e}}{|\bar{n}| |\bar{e}|} = \frac{(\bar{h} \times \bar{K}) \cdot \bar{e}}{|\bar{h} \times \bar{K}| |\bar{e}|} \quad (3.13)$$

- Anomalía

Este último elemento orbital permite posicionar el cuerpo dentro de la órbita en cada instante de tiempo. Por lo que dentro de una órbita concreta, sin tener en cuenta las perturbaciones que pueda haber, es el único elemento que depende del tiempo.

Dentro de este parámetro se pueden diferenciar tres tipos: anomalía media (M), anomalía excéntrica (E) y anomalía verdadera (θ). La imagen 11 representa las tres anomalías:

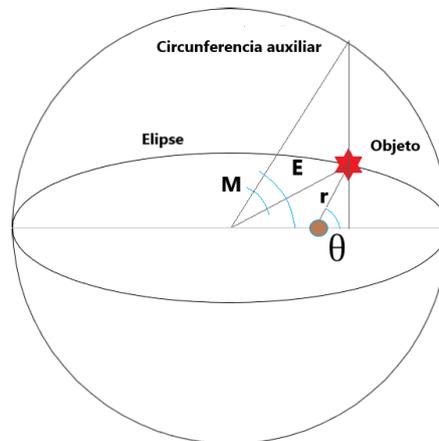


Figura 11: Representación de las distintas anomalías dentro de la órbita elíptica.

- Anomalía media (M)

Tal y como se observa en la figura anterior, la anomalía media es el ángulo que forma la línea que une el centro de la elipse con el objeto, con la línea que une el centro con la proyección del objeto en una circunferencia auxiliar que tiene de radio el semieje mayor de la elipse.

- Anomalía excéntrica (E)

Esta anomalía representa el ángulo que forma el semieje mayor de la elipse con la línea que une el centro de la elipse con la proyección del cuerpo en la circunferencia auxiliar.

- Anomalía verdadera (θ)

Esta anomalía es la más utilizada ya que permite calcular la posición y velocidad del objeto en coordenadas cartesianas. Representa el ángulo que forman el semieje mayor de la elipse con el vector posición del cuerpo. Este ángulo también varía entre 0° y 360° .

Aunque puede obtenerse a partir del valor de la anomalía excéntrica, es más sencilla la obtención de la misma a partir del vector posición y el vector excentricidad:

$$\cos \theta = \frac{\vec{e} \cdot \vec{r}}{|\vec{e}| |\vec{r}|} \quad (3.14)$$

3.3 Ecuación del movimiento

Otro de los principales conceptos dentro de la mecánica orbital es la ecuación del movimiento. La ley de gravitación universal (ecuación 3.15), que afirma que dos cuerpos se atraen con una fuerza \vec{F}_g proporcional al producto de sus masas (M_\oplus y m) e inversamente proporcional al cuadrado de la distancia que los separa (r), junto con la segunda ley de Newton (ecuación 3.15), según la cuál para un sistema de masa constante, la suma de las fuerzas ($\sum \vec{F}$) es igual a dicha masa (m) por su aceleración ($\ddot{\vec{r}}$), y la suposición de que la masa del cuerpo a estudiar es mucho menor que la del cuerpo central, permiten obtener la ecuación del movimiento, que demuestra que la aceleración de un cuerpo orbitando sobre otro, debida al efecto de la gravedad de este último, es la mostrada en la ecuación 3.16.

$$\sum \vec{F} = m\ddot{\vec{r}}, \quad \vec{F}_g = -\frac{GM_\oplus m}{r^2} \frac{\vec{r}}{|\vec{r}|}, \quad (3.15)$$

$$\ddot{\vec{r}} = -\frac{\mu}{r^2} \frac{\vec{r}}{|\vec{r}|} \quad (3.16)$$

Cabe destacar, que los resultados obtenidos al aplicar esta ecuación solo serán válidos, siempre que previamente se asuman las siguientes suposiciones [8]:

1. La masa del cuerpo a estudiar, comparada con la del cuerpo central, es lo suficientemente pequeña como para no tenerla en cuenta.
2. El sistema de coordenadas elegido para el problema debe de ser inercial. Esto se cumple tanto para el sistema heliocéntrico, utilizado para viajes interplanetarios, como para el sistema geocéntrico.
3. Ambos cuerpos, el central y el estudiado, se considerarán simétricamente esféricos y de densidad uniforme, para así poder tratarlos como masas puntuales.
4. Ninguna fuerza, excepto la gravitatoria entre ambos cuerpos, actuará en la propagación a realizar.

3.3.1 Técnicas de propagación

La ecuación del movimiento, anteriormente obtenida, define el problema de dos cuerpos, problema ideal de propagación, el cual no tienen en cuenta ninguna perturbación adicional, y

que constituye la base para el desarrollo de muchas de las técnicas de propagación utilizadas en la actualidad.

Tal y como se observa en las suposiciones realizadas anteriormente, el problema de dos cuerpos no permite tener en cuenta ninguna otra fuerza, excepto la de atracción entre los dos cuerpos involucrados. Esto no obtiene resultados realistas a lo largo del tiempo ya que, tal y como demuestran los datos obtenidos de las observaciones realizadas sobre planetas y cuerpos en el espacio, a lo largo de la órbitas descritas por los objetos, se producen variaciones irregulares no contempladas por el problema de dos cuerpos.

Por ello es necesario desarrollar otras técnicas para así poder tener en cuenta las distintas perturbaciones que se presentan en las órbitas descritas por distintos cuerpos en el espacio.

Existen varias formas para abordar el problema presentado por las perturbaciones. Estas se agrupan en tres tipos: analíticas (técnicas generales de perturbación), numéricas (técnicas especiales de perturbación) y semianalíticas (técnicas semianalíticas) [8].

- Técnicas especiales de perturbación

Estas técnicas integran numéricamente las ecuaciones de movimiento incluyendo las aceleraciones producidas por las perturbaciones que se quieren estudiar. Estas técnicas se utilizan principalmente por su rapidez computacional con respecto a las técnicas analíticas, ya que se puede añadir cualquier tipo de perturbación sin necesidad de modificar las ecuaciones iniciales.

Una de las técnicas especiales de perturbación más empleada es el **método de Cowell**, ésta parte de la base de que el problema de dos cuerpos no tiene en cuenta las perturbaciones y, por ello, añade una aceleración causada por dichas perturbaciones ($\bar{a}_{perturbed}$) a la formulación del problema de dos cuerpos (3.17).

$$\ddot{\vec{r}} = -\frac{\mu}{r^2} \frac{\vec{r}}{|\vec{r}|} + \bar{a}_{perturbed} \quad (3.17)$$

Esta aceleración ($\bar{a}_{perturbed}$) es la aceleración total causada por las fuerzas que actúan sobre el cuerpo, por lo que su forma dependerá del número y tipo de perturbaciones tenidas en cuenta. Además para las perturbaciones más significativas, y que se estudiarán en este trabajo, la formulación presentada permite añadir sus efectos linealmente, facilitando la propagación, y permitiendo calcular el vector de estado (\bar{X}) de la siguiente forma:

$$\bar{X} = \begin{bmatrix} r_x \\ r_y \\ r_z \\ v_x \\ v_y \\ v_z \end{bmatrix} \quad \dot{\bar{X}} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ -\frac{\mu r_x}{r^3} + a_{p_x} \\ -\frac{\mu r_y}{r^3} + a_{p_y} \\ -\frac{\mu r_z}{r^3} + a_{p_z} \end{bmatrix} \quad (3.18)$$

Como se ha comentado anteriormente, estas técnicas emplean métodos numéricos de integración para resolver las ecuaciones. Estos pueden ser de paso único o de paso variable, utilizando pasos distintos para mejorar la estimación.

En este trabajo se ha empleado un método de paso variable, el *ode45*, que viene implementado en MATLAB. Este se basa en los método Runge-Kutta, mejorado por

Dormand-Prince. Es un método de salida densa, es decir, capaz de generar polinomios, interpolando entre las soluciones que se han obtenido numéricamente. Esto permite obtener valores en todos los tiempos deseados. Este método aplica un método de orden 5 para resolver los sistemas de ecuaciones. Y al ser de paso variable permite mejorar la eficiencia de la integración.

- Técnicas generales de perturbación

Estas técnicas remplazan las ecuaciones de movimiento obtenidas anteriormente con aproximaciones analíticas que permiten una integración analítica tanto del movimiento ideal como del perturbado. Para integrar este último se realizan aproximaciones mediante series de expansión de las aceleraciones producidas por las perturbaciones. Por lo que las técnicas de perturbación generales normalmente son más difíciles de desarrollar que las especiales.

Por otra parte las técnicas generales de perturbación permiten obtener directamente los efectos que tienen las perturbaciones sobre cada uno de los elementos orbitales, en lugar de únicamente ver el efecto en las componentes de la posición y velocidad como en las técnicas especiales.

El método más común dentro de estas técnicas es el de **Variación de Parámetros** o (VOP). Este método fue desarrollado por Euler y mejorado por Lagrange y Gauss. Mientras que el método de Lagrange únicamente puede emplearse con aceleraciones conservativas, el de Gauss permite tener en cuenta además la no conservativas.

Este método parte del problema de dos cuerpos, en el cual los elementos orbitales dejan de ser constantes para pasar a definir una órbita perturbada en un instante de tiempo determinado. Cabe destacar que este método presenta algunos problemas para órbitas con excentricidades y/o inclinaciones pequeñas, ya que da lugar a singularidades.

3.4 Perturbaciones

Las perturbaciones no son más que desviaciones del movimiento normal e idealizado de los objetos.

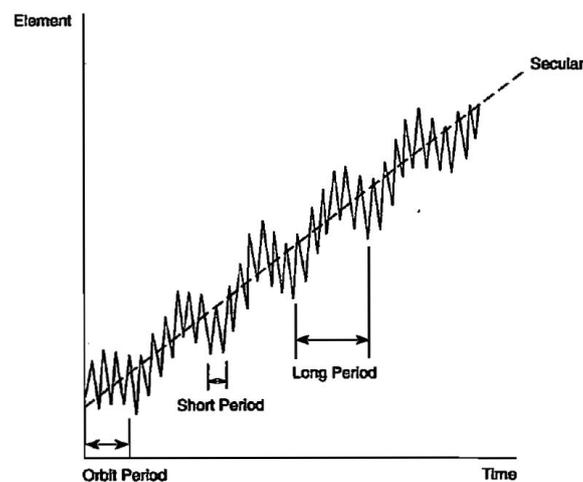


Figura 12: Tipos de perturbaciones según su efecto en los elementos orbitales [4].

A la hora de estudiar las distintas perturbaciones se pueden diferenciar tres tipos según varíen los elementos orbitales. En la figura 12 se representa la variación sufrida en uno de los elementos orbitales debida a distintos tipos de perturbaciones.

Mientras que las variaciones seculares representan una variación lineal en el tiempo del elemento orbital estudiado, las variaciones de periodo corto tienen un periodo de actuación menor que el periodo de la órbita generada por el cuerpo estudiado, y las variaciones de largo periodo tendrán un periodo mayor que dicha órbita [4].

Por otra parte, la influencia de las distintas perturbaciones dependerá de la distancia a la que se encuentre el cuerpo estudiado del cuerpo central del sistema y del resto de cuerpos, además de sus características, como el ratio área-masa. En la figura 13 se representa la influencia de la distancia al cuerpo central en las distintas perturbaciones posibles, sobre un satélite que orbita alrededor de la Tierra.

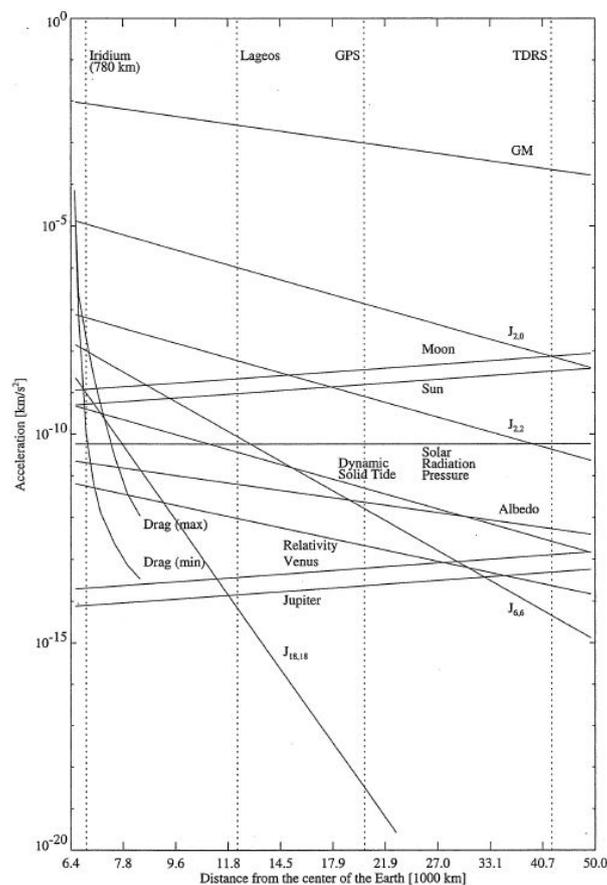


Figura 13: Influencia de las distintas perturbaciones en un satélite que orbita alrededor de la Tierra [9].

Tal y como se puede observar, a la hora de estudiar la órbita de un objeto alrededor de la Tierra, las perturbaciones más influyentes son las debidas a la no uniformidad del campo gravitatorio terrestre, la resistencia atmosférica, los efectos gravitatorios de un tercer cuerpo y la presión de radiación solar [9].

Para las órbitas interplanetarias la no uniformidad del campo gravitatorio terrestre y la resistencia atmosférica dejan de tener tanta importancia, por lo que solo se tendrán en cuenta

los efectos gravitatorios de un tercer cuerpo, teniendo mayor importancia que en órbitas geocéntricas los planetas, y la presión de radiación solar [8], [9], [4], [12].

3.4.1 Potencial gravitatorio terrestre

Anteriormente, para el estudio de las órbitas no perturbadas se ha supuesto que la masa total de la Tierra se encuentra concentrada en el centro del sistema de coordenadas. Una suposición más realista consistirá en utilizar el potencial gravitatorio por unidad de masa (U). Esto se debe a que la masa de la Tierra no es uniforme en todos los puntos, por lo que el vector de gravedad variará cambiando la aceleración perturbadora en cada uno de los puntos estudiados.

Este potencial gravitatorio por unidad de masa se obtiene del sumatorio del potencial de todos los puntos de la Tierra. En el origen únicamente se incluye la constante de gravitación universal (G), mientras que en el resto de puntos el potencial se obtiene de la división entre la masa en ese punto (m_Q) y la distancia desde el objeto a dicho punto (ρ_Q) [8]:

$$U = G \sum_{Q=1}^{\infty} \frac{m_Q}{\rho_Q} \quad (3.19)$$

Suponiendo que el cuerpo del cual se quiere obtener el potencial es un cuerpo con forma y distribución de densidad arbitrarias, como la Tierra, y que el potencial externo es estático, es decir, se desprecian los efectos de la tierra y las mareas, el potencial gravitatorio se puede expresar de la siguiente forma [8], [12]:

$$U = \frac{\mu}{r} \left[1 + \sum_{n=2}^{\infty} \sum_{m=0}^n \left(\frac{R_{\oplus}}{r} \right)^n P_{n,m}(\sin \phi_{sat}) \{C_{n,m} \cos(m\Lambda_{sat}) + S_{n,m} \sin(m\Lambda_{sat})\} \right] \quad (3.20)$$

donde r , ϕ_{sat} y Λ_{sat} son las coordenadas esféricas del punto en el que se encuentra el satélite: r es la distancia desde el centro de masas de la Tierra, ϕ_{sat} es la latitud geocéntrica y Λ_{sat} la longitud geocéntrica. Por otra parte los términos $P_{n,m} \sin \phi$ se asocian a las funciones de Legendre de primer tipo, de orden n y m , y los parámetros $C_{n,m}$ y $S_{n,m}$ son constantes del modelo. Por último, la variable μ es el parámetro gravitacional de la Tierra y R_{\oplus} es el radio ecuatorial medio de la Tierra.

Separando los términos de $m = 0$ y $m \neq 0$ de la ecuación 3.20, y redistribuyendo los términos se obtiene la siguiente expresión:

$$U = \frac{\mu}{r} \left[1 - \sum_{n=2}^{\infty} J_n \left(\frac{R_{\oplus}}{r} \right)^n P_n(\sin \phi_{sat}) + \sum_{n=2}^{\infty} \sum_{m=1}^n J_{n,m} \left(\frac{R_{\oplus}}{r} \right)^n P_{n,m}(\sin \phi_{sat}) \{ \cos m(\Lambda_{sat} - \Lambda_{sat,n,m}) \} \right] \quad (3.21)$$

en la cual los términos $P_n \sin \phi$ hacen referencia a los polinomios de Legendre de grado n y los términos $J_{n,m}$ y $\Lambda_{n,m}$ remplazan a los parámetros $C_{n,m}$ y $S_{n,m}$.

En la ecuación 3.21, el primer término representa el potencial de Newton, es decir, el potencial gravitacional de un cuerpo esférico con distribución de masa simétrica. El segundo refleja la influencia de las desviaciones en la forma y la distribución de la masa en dirección norte-sur, mientras que el tercero representa dichas desviaciones tanto en la dirección norte-sur como en la dirección este-oeste.

A continuación se presentan una serie de ecuaciones generales que permiten obtener los parámetros nombrados anteriormente para cada grado deseado:

$$\begin{aligned}
 x &= \sin \phi; \\
 P_n(x) &= \frac{1}{(-2)^n n!} \frac{d^n}{dx^n} (1 - x^2)^n \\
 P_{n,m}(x) &= (1 - x^2)^{m/2} \frac{d^m P_n(x)}{dx^m} \\
 J_{n,m} &= \sqrt{C_{n,m}^2 + S_{n,m}^2} \\
 \Lambda_{n,m} &= \frac{1}{m} \arctan \frac{S_{n,m}}{C_{n,m}}
 \end{aligned}
 \tag{3.22}$$

Para entender un poco mejor los coeficientes anteriores es necesario presentar los que son los armónicos esféricos. Estos son funciones armónicas definidas por los polinomios de Legendre y representadas en la superficie de una esfera unitaria. Los índices m y n de estas funciones determinan las líneas que aparecen en la esfera.

Dentro de estos armónicos esféricos cabe destacar la existencia de tres tipos distintos en función del valor de los índices. En la imagen 14 se presentan por este orden: *zonal harmonics* (armónicos zonales) cuyo índice n es distinto de 0 y su índice m es de orden cero (dentro de estos se puede destacar las variables P_n y J_n vistas anteriormente); *sectorial harmonics* cuyos índices n y m tienen el mismo valor distinto de 0; *tesseral harmonics* los cuales tienen índices distintos y no nulos.



Figura 14: Tipos de armónicos esféricos en función del valor de sus índices [12].

Por otra parte, en la figura 15 se representan las líneas de algunos de los armónicos zonales dependiendo del valor del índice n .

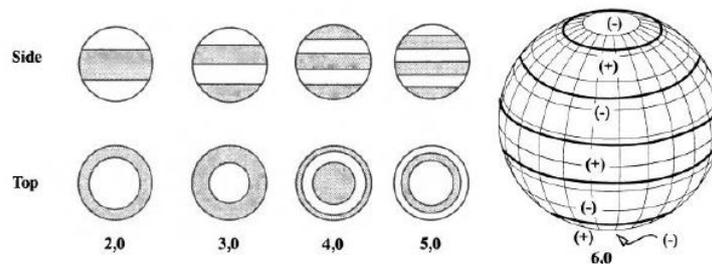


Figura 15: Bandas de los armónicos zonales de distintos ordenes [8].

A continuación se presenta un resumen de algunos de los valores de los armónicos esféricos anteriormente mencionados:

n	m	$J_{n,m} (10^{-6})$	$\Lambda_{n,m} (^\circ)$	n	m	$J_{n,m} (10^{-6})$	$\Lambda_{n,m} (^\circ)$
<i>Zonal harmonics</i>							
2	0	1082.6357	-----	6	0	0.5406168	-----
3	0	-2.5324737	-----	7	0	-0.3505229	-----
4	0	-1.6199743	-----	8	0	-0.2040168	-----
5	0	-0.2279051	-----	9	0	-0.1221502	-----
<i>Tesseral and sectorial harmonics</i>							
2	1	0.0018225	98.3325	5	2	0.1177945	-13.1874
2	2	1.8155628	-14.9287	5	3	0.0165311	-51.5184
3	1	2.2094849	6.9684	5	4	0.0023321	42.6063
3	2	0.3744510	-17.1887	5	5	0.0017034	-15.0730
3	3	0.2213884	20.9932	6	1	0.0632600	160.7431
4	1	0.6786576	-138.5480	6	2	0.0468934	-41.2906
4	2	0.1675835	31.0578	6	3	0.0012011	2.9578
4	3	0.0604207	-3.8219	6	4	0.0018140	-25.0853
4	4	0.0076442	30.3487	6	5	0.0004837	-23.2946
5	1	0.0971238	-123.6853	6	6	0.0000554	-14.6194

Figura 16: Resumen de algunos coeficientes de los armónicos esféricos [12].

El valor más alto se corresponde con el del J_2 , que se diferencia en varios órdenes de magnitud con el resto. Además, para igualdad de ordenes en el índice n , los armónicos zonales tienen mayor valor, y por lo tanto mayor influencia en la órbita. Por ello, para este trabajo se han estudiado únicamente estos armónicos esféricos.

Finalmente, una vez conocido el valor de los coeficientes a tener en cuenta se puede obtener la aceleración producida por la perturbación:

$$\bar{a}_{Harm} = -\bar{\nabla} \left(U + \frac{\mu}{r} \right) \quad (3.23)$$

Sustituyendo en la ecuación anterior el valor del potencia U por el segundo término de la ecuación 3.21:

$$\bar{a}_{Harm} = -\bar{\nabla} \left[U + \frac{\mu}{r} J_n \left(\frac{R_{\oplus}}{r} \right)^n P_n(\sin \phi_{sat}) \right] \quad (3.24)$$

y aplicando el valor de la variable P_n que corresponda, se obtiene una ecuación válida para todos los órdenes que se quieran estudiar.

En este trabajo se han estudiado la influencia de los armónicos zonales desde $n = 2$ hasta $n = 6$. Por lo que las ecuaciones necesarias para obtener las componentes de las aceleraciones producidas bajo la influencia de dichos armónicos, serán las siguientes [12]:

$$\bar{a}_{J_2} = \left\{ \begin{array}{l} \frac{3}{2} J_2 \frac{\mu}{r^2} \left(\frac{R_{\oplus}}{r} \right)^2 \frac{x}{r} \left(-1 + 5 \left(\frac{z}{r} \right)^2 \right) \\ \frac{3}{2} J_2 \frac{\mu}{r^2} \left(\frac{R_{\oplus}}{r} \right)^2 \frac{y}{r} \left(-1 + 5 \left(\frac{z}{r} \right)^2 \right) \\ \frac{3}{2} J_2 \frac{\mu}{r^2} \left(\frac{R_{\oplus}}{r} \right)^2 \frac{z}{r} \left(-3 + 5 \left(\frac{z}{r} \right)^2 \right) \end{array} \right\} \quad (3.25)$$

$$\begin{aligned}
\bar{a}_{J_3} &= \left\{ \begin{array}{l} \frac{5}{2} J_3 \frac{\mu}{r^2} \left(\frac{R_{\oplus}}{r} \right)^3 \frac{x}{r} \left(-3 \left(\frac{z}{r} \right) + 7 \left(\frac{z}{r} \right)^3 \right) \\ \frac{5}{2} J_3 \frac{\mu}{r^2} \left(\frac{R_{\oplus}}{r} \right)^3 \frac{y}{r} \left(-3 \left(\frac{z}{r} \right) + 7 \left(\frac{z}{r} \right)^3 \right) \\ \frac{3}{2} J_3 \frac{\mu}{r^2} \left(\frac{R_{\oplus}}{r} \right)^3 \left(1 - 10 \left(\frac{z}{r} \right)^2 + \frac{35}{3} \left(\frac{z}{r} \right)^4 \right) \end{array} \right\} \\
\bar{a}_{J_4} &= \left\{ \begin{array}{l} \frac{5}{8} J_4 \frac{\mu}{r^2} \left(\frac{R_{\oplus}}{r} \right)^4 \frac{x}{r} \left(3 - 42 \left(\frac{z}{r} \right)^2 + 63 \left(\frac{z}{r} \right)^4 \right) \\ \frac{5}{8} J_4 \frac{\mu}{r^2} \left(\frac{R_{\oplus}}{r} \right)^4 \frac{y}{r} \left(3 - 42 \left(\frac{z}{r} \right)^2 + 63 \left(\frac{z}{r} \right)^4 \right) \\ \frac{5}{8} J_4 \frac{\mu}{r^2} \left(\frac{R_{\oplus}}{r} \right)^4 \frac{z}{r} \left(15 - 70 \left(\frac{z}{r} \right)^2 + 63 \left(\frac{z}{r} \right)^4 \right) \end{array} \right\} \\
\bar{a}_{J_5} &= \left\{ \begin{array}{l} \frac{3}{8} J_5 \frac{\mu}{r^2} \left(\frac{R_{\oplus}}{r} \right)^5 \frac{x}{r} \left(35 \left(\frac{z}{r} \right) - 210 \left(\frac{z}{r} \right)^3 + 231 \left(\frac{z}{r} \right)^5 \right) \\ \frac{3}{8} J_5 \frac{\mu}{r^2} \left(\frac{R_{\oplus}}{r} \right)^5 \frac{y}{r} \left(35 \left(\frac{z}{r} \right) - 210 \left(\frac{z}{r} \right)^3 + 231 \left(\frac{z}{r} \right)^5 \right) \\ \frac{1}{8} J_5 \frac{\mu}{r^2} \left(\frac{R_{\oplus}}{r} \right)^5 \left(-15 + 315 \left(\frac{z}{r} \right)^2 - 945 \left(\frac{z}{r} \right)^4 + 693 \left(\frac{z}{r} \right)^6 \right) \end{array} \right\} \\
\bar{a}_{J_6} &= \left\{ \begin{array}{l} \frac{1}{16} J_6 \frac{\mu}{r^2} \left(\frac{R_{\oplus}}{r} \right)^6 \frac{x}{r} \left(-35 + 945 \left(\frac{z}{r} \right)^2 - 3465 \left(\frac{z}{r} \right)^4 + 3003 \left(\frac{z}{r} \right)^6 \right) \\ \frac{1}{16} J_6 \frac{\mu}{r^2} \left(\frac{R_{\oplus}}{r} \right)^6 \frac{y}{r} \left(-35 + 945 \left(\frac{z}{r} \right)^2 - 3465 \left(\frac{z}{r} \right)^4 + 3003 \left(\frac{z}{r} \right)^6 \right) \\ \frac{1}{16} J_6 \frac{\mu}{r^2} \left(\frac{R_{\oplus}}{r} \right)^6 \frac{z}{r} \left(-245 + 2205 \left(\frac{z}{r} \right)^2 - 4851 \left(\frac{z}{r} \right)^4 + 3003 \left(\frac{z}{r} \right)^6 \right) \end{array} \right\}
\end{aligned} \tag{3.26}$$

3.4.2 Resistencia atmosférica

Aunque el aire de la capa superior de la atmósfera tiene una densidad muy baja, la resistencia que produce es una de las perturbaciones más importantes en satélites de órbitas LEO (*Low Earth Orbit*). Además, llega a ser la mayor perturbación, por encima incluso de la no uniformidad del campo gravitatorio terrestre, para órbitas con alturas de hasta 300 km [8].

La resistencia atmosférica actúa en la dirección opuesta a la velocidad del satélite, provocando una reducción de la energía de su la orbita, que se traduce en la variación del tamaño de dicha órbita. Esto puede llegar a provocar que la altura de la órbita disminuya hasta un punto en el que se produzca la reentrada del satélite en la atmósfera terrestre [4].

Debido a la no conservación de la energía total del sistema, la resistencia atmosférica se define como una perturbación no conservativa, que provoca perturbaciones no conservativas en las componentes de la velocidad. Además, al variar el tamaño de la órbita se modifican tanto el semieje mayor de la misma como el valor de la excentricidad. En cambio el resto de elementos orbitales sufren variaciones periódicas bajo la influencia de esta perturbación [8].

El modelado de la resistencia atmosférica no es sencillo, debido a tres puntos [9]:

- Las propiedades físicas de la atmósfera, y en especial su densidad en la parte superior, no se conocen con exactitud.
- El modelado de la fuerza de resistencia necesitaría conocer con detalle la interacción que se produce entre el gas o las partículas y las distintas superficies del satélite estudiado.
- Se deben de tener en cuenta las diferencias en el comportamiento de los satélites no esféricos en el flujo con respecto a los esféricos.

Debido a estas tres razones, para obtener un modelo de la resistencia aerodinámica preciso, se deberían de conocer conceptos de energía molecular, termodinámica, aerodinámica o meteorología, además de la mecánica orbital. Además, se deberán conocer tanto las características de la troposfera como de la ionosfera.

A lo largo de la historia se han desarrollado muchos modelos distintos para conseguir obtener con exactitud el modelado de la atmósfera. Algunos de ellos se presentan en la imagen 17:

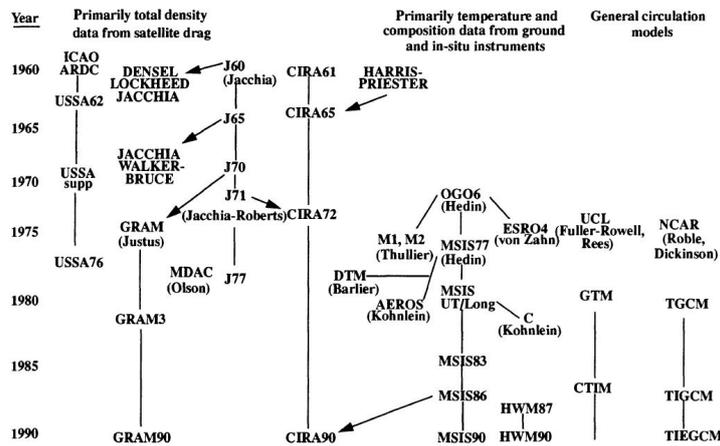


Figura 17: Evolución de los modelos atmosféricos [8].

En este trabajo se ha empleado el modelo exponencial, ya que la finalidad del mismo no es conseguir la mayor exactitud posible a costa de un gran coste computacional, sino desarrollar las herramientas que permitan obtener unos resultados correctos. Esto no quita que posteriormente sea posible aplicar otro modelo atmosférico más preciso.

El modelo exponencial se emplea para alturas orbitales entre 0 y 1000 km. Este asume que la densidad de la atmósfera decae exponencialmente desde la superficie, y que las partículas se distribuyen simétricamente. Por lo que la ecuación para aplicar dicho modelo será la siguiente:

$$\rho = \rho_0 e^{\left[\frac{h-h_0}{H}\right]} \tag{3.27}$$

donde ρ es la densidad a obtener, ρ_0 la densidad de referencia, h la altura del satélite, h_0 la altura de referencia y H la *altitud de escala*.

Tanto la densidad y la altura de referencia, como la *altitud de escala* vienen tabuladas tal y como se muestra en la figura 18

Altitude h_{ellp} (km)	Base Altitude h_o (km)	Nominal Density ρ_o (kg/m ³)	Scale Height H (km)	Altitude h_{ellp} (km)	Base Altitude h_o (km)	Nominal Density ρ_o (kg/m ³)	Scale Height H (km)
0–25	0	1.225	7.249	150–180	150	2.070×10^{-9}	22.523
25–30	25	3.899×10^{-2}	6.349	180–200	180	5.464×10^{-10}	29.740
30–40	30	1.774×10^{-2}	6.682	200–250	200	2.789×10^{-10}	37.105
40–50	40	3.972×10^{-3}	7.554	250–300	250	7.248×10^{-11}	45.546
50–60	50	1.057×10^{-3}	8.382	300–350	300	2.418×10^{-11}	53.628
60–70	60	3.206×10^{-4}	7.714	350–400	350	9.158×10^{-12}	53.298
70–80	70	8.770×10^{-5}	6.549	400–450	400	3.725×10^{-12}	58.515
80–90	80	1.905×10^{-5}	5.799	450–500	450	1.585×10^{-12}	60.828
90–100	90	3.396×10^{-6}	5.382	500–600	500	6.967×10^{-13}	63.822
100–110	100	5.297×10^{-7}	5.877	600–700	600	1.454×10^{-13}	71.835
110–120	110	9.661×10^{-8}	7.263	700–800	700	3.614×10^{-14}	88.667
120–130	120	2.438×10^{-8}	9.473	800–900	800	1.170×10^{-14}	124.64
130–140	130	8.484×10^{-9}	12.636	900–1000	900	5.245×10^{-15}	181.05
140–150	140	3.845×10^{-9}	16.149	1000–	1000	3.019×10^{-15}	268.00

Figura 18: Valores de la densidad de referencia (ρ_0), de la altura de referencia (h_0) y de la *altitud de escala* (H), en función de la altura del satélite (h_{ellp}) [8].

La dependencia de la resistencia aerodinámica con la velocidad relativa del objeto (v_{rel}) se puede obtener fácilmente. Partiendo de una pequeña porción de masa de la atmósfera (Δm) que choca contra una sección A del satélite en un intervalo de tiempo Δt :

$$\Delta m = \rho A v_{rel} t \quad (3.28)$$

donde la variable ρ hace referencia a la densidad de la atmósfera en ese punto, el impulso ejercido sobre el satélite se puede definir de la siguiente forma:

$$\Delta p = \Delta m v_{rel} = \rho A v_{rel}^2 \Delta t \quad (3.29)$$

Por lo que, teniendo en cuenta que la fuerza resultante es $F = \Delta p / \Delta t$, la aceleración inducida al satélite por parte de la resistencia atmosférica será la siguiente:

$$\bar{a}_{atm} = -\frac{1}{2} C_D \frac{A}{m} \rho v_{rel}^2 \bar{e}_v \quad (3.30)$$

siendo C_D el coeficiente de resistencia aerodinámica, que describe la interacción de la atmósfera sobre la superficie del satélite, cuyo valor varía normalmente entre 1.5 y 3; A/m la relación área-masa del satélite; y \bar{e}_v el vector unitario que describe la dirección de la aceleración ($\bar{e}_v = \bar{v}_{rel} / v_{rel}$). El factor 1/2 se ha introducido para asegurar la consistencia de la notación, siguiendo la notación para fuerzas aerodinámicas ya conocidas.

Finalmente, para obtener la aceleración expresada en sus tres componentes (x, y, z), será necesario obtener la velocidad relativa:

$$\bar{v}_{rel} = \frac{d\bar{r}}{dt} - \bar{\omega}_{\oplus} \times \bar{r} = \left\{ \begin{array}{l} \frac{dx}{dt} - \bar{\omega}_{\oplus} y \\ \frac{dy}{dt} - \bar{\omega}_{\oplus} x \\ \frac{dz}{dt} \end{array} \right\} \quad (3.31)$$

donde $\bar{\omega}_{\oplus}$ es la velocidad angular con la que rota la Tierra, que tiene un valor de $7.292 \cdot 10^{-4}$ rad/s [9].

3.4.3 Efectos gravitatorios de un tercer cuerpo

Otra de las perturbaciones a tener en cuenta a la hora es estudiar una órbita, es el efecto gravitatorio causado por un tercer cuerpo. Mientras que para órbitas geoestacionarias esta perturbación no es tan significativa como las otras dos anteriores (únicamente pueden ser destacables los efectos de la Luna y el Sol, en órbitas más altas que las LEO), para misiones interplanetarias es necesario tener en cuenta todos los posibles planetas y cuerpos de gran tamaño que puedan afectar a la trayectoria.

Esta perturbación causa variaciones periódicas en los elementos orbitales, pero únicamente la ascensión recta del nodo ascendente, el argumento del perigeo y la anomalía media experimentan variaciones seculares. La variación en la anomalía media es menor que el movimiento medio, por lo que apenas afecta a la órbita. Pero las variaciones producidas en la ascensión recta del nodo ascendente y en el argumento del perigeo, sí que son importantes, sobretodo en las órbitas de gran altitud.

Partiendo de la ley de gravitación universal de Newton se obtiene que la aceleración de un cuerpo en un punto de masa (M) es:

$$\bar{a} = GM \frac{r_{\oplus} - r_{sat}}{|r_{\oplus} - r_{sat}|^3} \quad (3.32)$$

donde r_{sat} y r_{\oplus} son las coordenadas del satélite y del punto de masa M respectivamente.

Esta ecuación está incompleta, ya que es necesario tener en cuenta el movimiento del centro de la Tierra, para poder emplear un sistema de coordenadas inercial. El centro de la Tierra se ve afectado por una aceleración que puede expresarse de la siguiente forma:

$$\bar{a} = GM \frac{r_{sat}}{|r|^3} \quad (3.33)$$

Juntando ambas expresiones se obtiene la ecuación capaz de describir la aceleración producida por la perturbación de un tercer cuerpo:

$$\bar{a}_{plan} = GM \left(\frac{r_{\oplus} - r_{sat}}{|r_{\oplus} - r_{sat}|^3} - \frac{r_{sat}}{|r|^3} \right) \quad (3.34)$$

3.4.4 Presión de radiación solar

Por último, la presión de radiación solar es una perturbación que afecta a todos los cuerpos que se encuentran orbitando dentro del sistema solar. Esta es una fuerza, relativamente pequeña, producida por la absorción o reflejo de fotones [9].

Un fotón lleva un momento igual a su energía dividida por la velocidad de la luz. Este momento es intercambiado en la interacción del fotón con una superficie sólida. Esto produce una presión de radiación igual al vector diferencia entre el momento incidente del flujo y el reflejado. La radiación solar se comporta como un flujo constante de partículas emitido por el Sol. Esta es la fuente principal de presión de radiación a la que puede verse sometido un cuerpo orbitando dentro del sistema solar [13].

Al igual que ocurre con la resistencia atmosférica esta perturbación no es conservativa, varía la energía del cuerpo sobre el que actual al variar su momento [8]. Además, causa variaciones periódicas en todos los elementos orbitales [4].

En cuanto a la importancia de esta perturbación, en órbitas geocéntricas la presión de radiación solar aumenta su importancia con la altura, siendo prácticamente nula en órbitas LEO. Por otra parte, en órbitas heliocéntricas se deberá tener en cuenta en todo momento, ya que puede afectar en gran medida a la trayectoria del cuerpo estudiado. Asimismo, el efecto de esta perturbación será mayor a menor coeficiente balístico del cuerpo, es decir, a más esbelto sea y mayor área frontal tenga [4].

A la hora de analizar esta perturbación, el mayor problema se encuentra a la hora de predecir los ciclos solares y sus variaciones, ya que la intensidad de la perturbación dependerá de la actividad solar, siendo más intensa en los periodos de tormentas solares. Por otra parte, también es necesario conocer el área sobre el que actúa dicha perturbación, ya que solo afecta al área que se encuentra expuesta a la radiación solar. Por último, a lo largo de la trayectoria del cuerpo es posible que existan momentos de penumbra o eclipse por la sombra de un cuerpo entre él y el Sol. Esto se deberá estudiar, ya que en dichos momentos no existirá el efecto de la radiación solar sobre el cuerpo estudiado [8].

Para obtener la aceleración causada por la radiación solar será necesario obtener la expresión de la fuerza causada por la misma. Esta se obtendrá a partir del valor de la presión de radiación solar, definida como cantidad de momento [8]. Cabe destacar que para el desarrollo realizado, se ha supuesto un modelo esférico del cuerpo estudiado, en el cual se asume que el cuerpo es una esfera invariante con propiedades térmicas constantes [13].

Partiendo de la ecuación de la relatividad de Einstein ($E = mc^2$), y sabiendo que E es la energía relativa, c la velocidad de la luz y mc el momento:

$$mc = \frac{E}{c} \quad (3.35)$$

se puede obtener dicho momento, o presión solar, una vez conocida la energía. Esta energía se ha tomado igual a la constante de radiación solar, o flujo solar: $SF : 1353 \text{ W/m}^2$. Por lo que la presión de radiación solar (p_{SR}) será:

$$p_{SR} = \frac{SF}{c} = \frac{1352 \text{ W/m}^2}{3 \cdot 10^8 \text{ m/s}} = 4.51 \cdot 10^{-6} \frac{\text{N}}{\text{m}^2} \quad (3.36)$$

Una vez calculada la presión, se obtiene la expresión de la fuerza generada por la radiación solar. El modelo esférico tomado permite suponer que existe un área del satélite constante expuesta a la radiación solar. Por lo que únicamente es necesario definir dicha área y el coeficiente de reflectividad c_R [13]. Teniendo en cuenta todo lo anterior, se obtiene la siguiente expresión:

$$F_{SR} = p_{SR} c_R A \quad (3.37)$$

en la cual, el valor de p_{SR} corresponde con el valor de la presión de radiación solar obtenido anteriormente; c_R es el coeficiente de reflectividad que varía entre 0 y 2, siendo 0 cuando no existe radiación incidente sobre el cuerpo estudiado, 1 cuando se absorbe toda la radiación recibida, y 2 cuando toda la radiación se refleja; y A es el área que se encuentra expuesta a la radiación Solar.

Finalmente, es posible obtener una primera expresión de la aceleración producida por la radiación solar:

$$a_{SRP} = \frac{F}{m} = \frac{p_{SR} c_R A}{m} \quad (3.38)$$

De manera vectorial, se sabe que el vector aceleración se aleja del Sol, por lo que:

$$\vec{a}_{SRP} = -\frac{p_{SR} c_R A}{m} \frac{\vec{r}_{body}}{|\vec{r}_{body}|} \quad (3.39)$$

donde \vec{r}_{body} es el vector posición del cuerpo estudiado. [8]

Esta expresión puede verse mejorada al tener en cuenta los efectos de penumbra o eclipse mencionados anteriormente, mediante una función S :

$$\vec{a}_{SRP} = -S \frac{p_{SR} c_R A}{m} \frac{\vec{r}_{body}}{|\vec{r}_{body}|} \quad (3.40)$$

en la que:

- $S = 0$: Si el cuerpo está en umbra. No le afecta la radiación del Sol.
- $S = 1$: Si el cuerpo está iluminado. Le afecta la radiación del Sol en todo el área indicada.
- $0 < S < 1$: Si el cuerpo se encuentra en penumbra. Solo le afecta la radiación en una parte del área estudiada.

Para el cálculo de dicha función, será necesario realizar los siguientes pasos:

1. Calcular el vector distancia entre el cuerpo estudiado (*body*) y el cuerpo que puede generar la sombra (*bodyShadow*):

$$\vec{s} = \vec{r}_{body} - \vec{r}_{bodyShadow} \quad (3.41)$$

2. Obtener los coeficientes a , b y c :

$$a = \arcsin\left(\frac{R_{Sol}}{|\vec{r}_{Sol} - \vec{r}_{body}|}\right) \quad (3.42)$$

siendo R_{Sol} el radio del Sol, y r_{Sol} las coordenadas del Sol en el sistema de referencia empleado.

$$b = \arcsin \left(\frac{R_{bodyShadow}}{|\bar{s}|} \right) \quad (3.43)$$

siendo $R_{bodyShadow}$ el radio del cuerpo que puede generar la sombra.

$$c = \arccos \left(\frac{-\bar{s}^T \cdot (\bar{r}_{Sol} - \bar{r}_{body})}{|\bar{s}| |\bar{r}_{Sol} - \bar{r}_{body}|} \right) \quad (3.44)$$

3. Dependiendo de los valores de estos coeficientes S tendrá un valor u otro:

- Si $|a - b| < c < a + b$

$$S = 1 - \frac{A}{\pi a^2} \quad (3.45)$$

donde A se obtiene de la expresión:

$$A = a^2 \arccos \left(\frac{x}{a} \right) + b^2 \arccos \left(\frac{c - x}{b} \right) - c y \quad (3.46)$$

siendo:

$$x = \frac{c^2 + a^2 - b^2}{2c} \quad y = \sqrt{a^2 - x^2} \quad (3.47)$$

- Si $a + b \leq c$

$$S = 1 \quad (3.48)$$

- Si $b - a > c$ y $a < b$

$$S = 0 \quad (3.49)$$

- Si $a - b > c$ y $a < b$

$$S = 0 \quad (3.50)$$

Cabe destacar que en este trabajo se han tenido en cuenta las sombras producidas por la Tierra, la Luna y Venus, como base de partida, teniendo en cuenta que la mayoría de órbitas actuales, tanto geocéntricas como heliocéntricas, pueden verse afectadas por alguno de los tres cuerpos. Por lo que la función S final será el resultado de la multiplicación de las funciones obtenidas para cada uno de los tres cuerpos [9].

3.5 Maniobras interorbitales

Las trayectorias definidas en las misiones espaciales suelen incluir cambios de órbitas o correcciones en las órbitas definidas. Los cambios de órbitas son típicos de misiones interplanetarias, y suelen necesitar grandes impulsos para producirse. Por otra parte, las correcciones dentro de la misma órbita son necesarias debido a la influencia de las perturbaciones, ya que pueden llegar a modificar la órbita poniendo en riesgo la misión inicial definida. Estas correcciones son de menor intensidad que los cambios de órbita. Además, también pueden emplearse impulsos para maniobras como la reentrada o el *rendezvous* entre dos naves espaciales [14].

3.5.1 Maniobras impulsivas

Para el estudio de todo lo anterior, en este trabajo se han utilizado maniobras impulsivas. Estas maniobras son un tipo idealizado de maniobras, mediante las cuales se supone un impulso que modifica el vector velocidad de manera instantánea mientras mantiene el vector posición que tenía la nave previamente. Este tipo de maniobras se ajustan a la realidad en los casos en los que el vector posición se modifica únicamente cuando los motores del cohete se encuentran encendidos, es decir, en aquellos casos en los que los tiempos de encendido de los motores son cortos debido a que dan un gran impulso. Sabiendo que Δv representa el cambio en el vector velocidad, en magnitud y dirección, se puede obtener la relación que tiene dicho cambio de velocidad con el consumo de propelente Δm . Partiendo de la definición de impulso específico del combustible: periodo, en segundos, durante el cual 1 kg de combustible junto con oxidante producirá 1 kg de fuerza:

$$I_{sp} = \frac{\text{Empuje}}{\text{Consumo de combustible a nivel del mar}} \quad (3.51)$$

se puede obtener la ecuación de Tsiolkovski en el espacio libre y despreciando las fuerzas de resistencia atmosférica y gravedad (ecuación 3.52). Esta ecuación permite obtener el cambio de velocidad a partir del impulso específico de los motores (I_{sp}), la aceleración de la gravedad a nivel del mar g_0 y las masas antes (m_i) y después (m_f) del encendido de los motores, o de dichas masas y la velocidad de eyección característica o efectiva C_e .

$$\Delta v = I_{sp} g_0 \ln \left(\frac{m_i}{m_f} \right) = C_e \ln \left(\frac{m_i}{m_f} \right) \quad (3.52)$$

Despejando la proporción de masa de la ecuación, se obtiene otra relación entre el consumo de combustible y la velocidad de la maniobra:

$$\frac{\Delta m}{m_i} = 1 - e^{-\frac{\Delta v}{I_{sp} g_0}} \quad (3.53)$$

En la siguiente gráfica se muestran algunos combustibles con distintos impulsos específicos, y la relación que mantiene cada uno de ellos entre la velocidad de maniobra y el consumo de combustible.

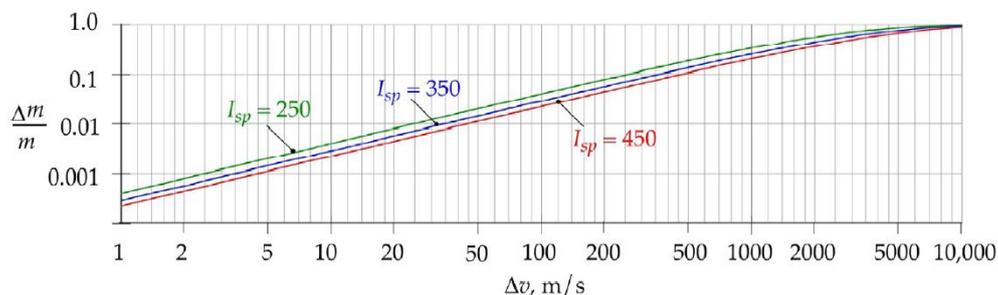


Figura 19: Relación velocidad de maniobra - consumo de combustible para distintos impulsos específicos [14].

Se observa como, a mayor impulso específico el consumo de combustible es menor. Además, para incrementos de velocidad cercanos a 1 km/s este consumo es un 25 % mayor que

al inicio de la maniobra. Por lo que a la hora de diseñar una maniobra orbital será necesario tener en cuenta, no solo la velocidad necesaria, sino que también es importante estudiar el consumo de combustible a lo largo de la misma. Así, la órbita óptima normalmente será aquella en la que la relación consumo-velocidad sea más baja.

3.5.2 Transferencia de Hohmann

Dentro de las maniobras impulsivas, una de las más destacadas es la transferencia de Hohmann. Esta es la transferencia bi-impulsiva más eficiente desde el punto de vista energético, y permite realizar un cambio de órbita entre órbitas circulares coplanarias, es decir, cuyo foco principal es el mismo. La transferencia consiste en la utilización de una órbita elíptica de transferencia entre las órbitas de origen y destino. En la imagen 20 se representa gráficamente la transferencia de Hohmann.

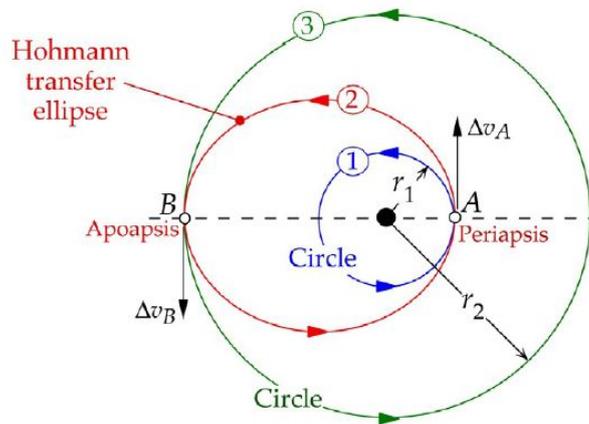


Figura 20: Esquema de la transferencia de Hohmann elíptica [14].

Tal y como se ha mencionado anteriormente, la transferencia de Hohmann es una transferencia bi-impulsiva, por lo que será necesario realizar dos cambios de velocidad. Estos cambios de velocidad se producen en dos puntos clave del órbita, el periapsis (punto A) y el apoapsis (punto B). En el primero de ellos se produce un incremento en la velocidad igual a Δv_A . Este incremento será el necesario para que el satélite o nave espacial consiga alcanzar una órbita elíptica de mayor energía. Una vez llegado al punto B de la imagen, se realiza el segundo impulso, que permite pasar de la órbita de transferencia a la órbita final deseada, también de mayor energía. Este generará un incremento de velocidad igual a Δv_B . Por lo que finalmente, el cambio de velocidad final total producido durante la transferencia será:

$$\Delta v_{total} = \Delta v_A + \Delta v_B \quad (3.54)$$

Cabe destacar que este incremento de velocidad total será el mismo si se quiere pasar de la órbita más pequeña a la más grande, es decir, de A a B, que de la más grande a la más pequeña, de B a A. Ya que este incremento se mide únicamente en magnitud y, aunque en el caso en el que se pretende pasar de B a A se reduce la energía de la órbita, la magnitud del cambio de velocidad total será la misma.

4 APLICACIÓN EN MATLAB

Como se ha mencionado anteriormente, el objetivo de este trabajo radica en el diseño de una aplicación que permite diseñar y propagar las órbitas y trayectorias de misiones espaciales. Para ello se ha empleado la herramienta *app Designer* de MATLAB. Esta permite la creación de una interfaz gráfica en la que se pueden emplear distintas funciones previamente desarrolladas en MATLAB. Por lo que será sencillo implementar las distintas funciones de lectura de datos y propagación de la órbita que se desarrollen.

Para la aplicación se partirá de unos datos iniciales que el usuario deberá de conocer: fechas de salida y llegada, vector de estado inicial, y condiciones de la propagación (perturbaciones e impulsos). Finalmente por pantalla saldrá el valor del vector de estado final, pero además se obtendrán dos archivos de texto con los valores de la posición y velocidad a lo largo de toda la trayectoria.

4.1 Estructura de la aplicación

Para realizar lo anteriormente descrito, la aplicación seguirá el diagrama de flujo presentado a continuación:

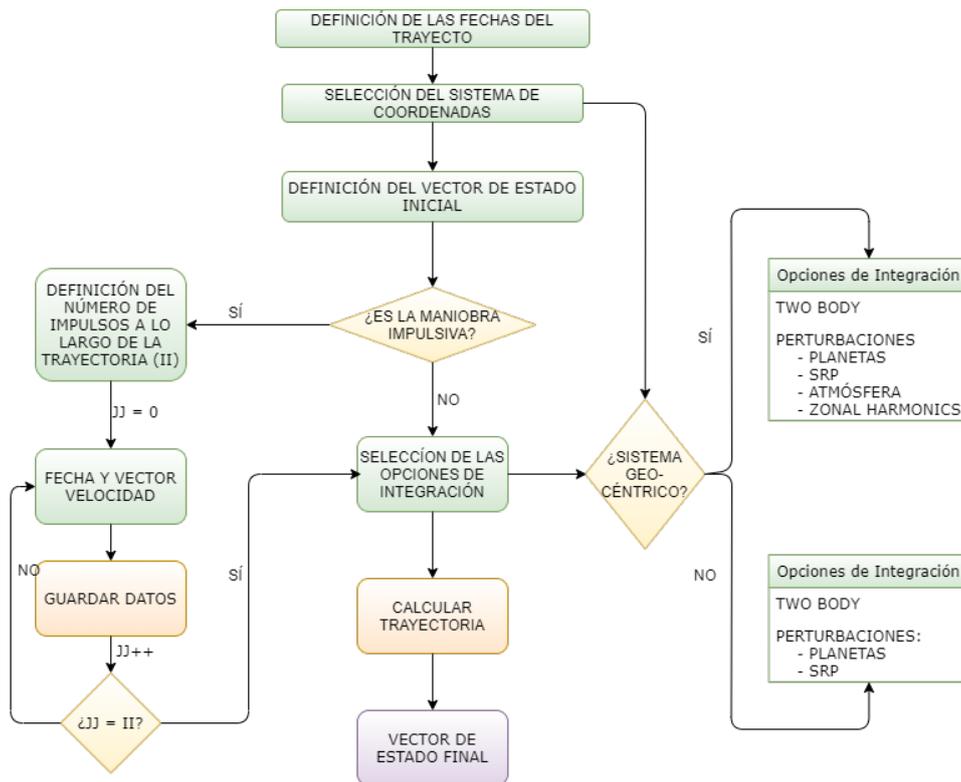


Figura 21: Diagrama de flujo de la utilización de la aplicación diseñada.

En primer lugar se piden las fechas inicial y final del trayecto y el sistema de coordenadas en el que se va a trabajar.(22).

The screenshot shows a MATLAB application window with the following elements:

- Fechas Trayecto:**
 - Fecha Inicial:** Año: 2020, Mes: 12, Día: 23; Hora: 0, Minuto: 0, Segundo: 0.
 - Fecha Final:** Año: 2021, Mes: 1, Día: 12; Hora: 0, Minuto: 0, Segundo: 0.
- Foco de la órbita:** A dropdown menu with 'Sol' and 'Tierra' options.
- ¿Maniobra Impulsiva?:** Two buttons labeled 'Sí' and 'No'.
- Opciones de Integración:** Checkboxes for 'Two Body', 'Zonal Harmonics', 'Atmósfera', 'SRP', and 'Planetas'.
- Vector de estado final:** A section with two columns: 'Posición' (x, y, z) and 'Velocidad' (Vx, Vy, Vz), each with input fields set to 0.
- Botón de Cálculo:** A central button labeled 'Cálculo'.

Figura 22: Definición de las fechas inicial y final del trayecto y selección del sistema de coordenadas

Aunque el usuario comience definiendo los datos de las fechas iniciales, hasta que no se pulsa el botón de *Cálculo* no se hace uso de estos datos. Por lo que el primer elemento en realizar alguna función es la lista *Drop Down* mediante la cual se selecciona el sistema de coordenadas, **Foco de la órbita**.

Tal y como se muestra en la figura 23, una vez seleccionado un sistema de coordenadas válido (geocéntrico o heliocéntrico), mediante la selección del cuerpo central (Tierra o Sol), la función relacionada con esta lista activará los *Edit Fields* necesarios para definir el vector de estado inicial del cuerpo que se va a propagar.

Además también se activará la pregunta **Maniobra impulsiva**, que pretende preguntar acerca de la utilización de impulsos o no a lo largo de la trayectoria del cuerpo.

Para poder realizar todo lo anterior, dentro de la aplicación se ha creado el código 1, presentado en el anexo A.

A continuación, al igual que pasa con los datos de las fechas, el vector de estado inicial no se empleará hasta pulsar el botón de *Cálculo*. Por lo que, tras conocer el vector de estado inicial, el siguiente elemento que realizará alguna función en la aplicación serán los botones utilizados para elegir si a lo largo de la trayectoria se producirá algún impulso o no.

Esta selección se realizará mediante un *State Button*, que permitirán generar un código sencillo para que, si se elige una de las dos opciones, no se pueda seleccionar la otra, y si se vuelve a pulsar sobre la elegida, se podrá volver a realizar la selección. Estos botones son los **botones Sí y No**, y además de asegurar lo anterior, cada uno de ellos activará distintos componentes.

Sol ▼

Fechas Trayecto

Fecha Inicial			Fecha Final			Vector de estado inicial			
Año	Mes	Día	Año	Mes	Día	Posición		Velocidad	
2020	12	23	2021	1	12	x	5e+08	Vx	23
						y	-1.2e+07	Vy	12
Hora	Minuto	Segundo	Hora	Minuto	Segundo	z	1e+05	Vz	1
0	0	0	0	0	0				

¿Maniobra Impulsiva?

Opciones de Integración

Two Body

Zonal Harmonics

Atmósfera

SRP Planetas

Vector de estado final

Posición		Velocidad	
x	0	Vx	0
y	0	Vy	0
z	0	Vz	0

Figura 23: Definición del vector de estado inicial.

Si en la trayectoria se produce algún impulso, se pulsará el botón **Sí**. Este activará el **contador de impulsos totales**, en el cual se deberá de insertar el número de impulsos que se desean realizar a lo largo de la trayectoria. El código mediante el cual se controla el botón **Sí** se presenta en el anexo A, y es el código 2. Además este código se asegura de ocultar todos los componentes que estén relacionados con el botón **Sí**, cuando este no esté seleccionado.

Sol ▼

Fechas Trayecto

Fecha Inicial			Fecha Final			Vector de estado inicial			
Año	Mes	Día	Año	Mes	Día	Posición		Velocidad	
2020	12	23	2021	1	12	x	5e+08	Vx	23
						y	-1.2e+07	Vy	12
Hora	Minuto	Segundo	Hora	Minuto	Segundo	z	1e+05	Vz	1
0	0	0	0	0	0				

¿Maniobra Impulsiva?

Número de Impusos Totales

Opciones de Integración

Two Body

Zonal Harmonics

Atmósfera

SRP Planetas

Vector de estado final

Posición		Velocidad	
x	0	Vx	0
y	0	Vy	0
z	0	Vz	0

Figura 24: Elección sobre maniobra con impulsos o no. Al ser sí, se necesitan el número de impulsos.

Fechas Trayecto Sol ▼

Fecha Inicial **Fecha Final** **Vector de estado inicial**

Año	Mes	Día	Año	Mes	Día	Posición		Velocidad	
2020	12	23	2021	1	12	x	5e+08	Vx	23
Hora	Minuto	Segundo	Hora	Minuto	Segundo	y	-1.2e+07	Vy	12
0	0	0	0	0	0	z	1e+05	Vz	1

¿Maniobra Impulsiva?

Opciones de Integración
 Two Body
 Zonal Harmonics
 Atmósfera
 SRP Planetas

Vector de estado final

Posición		Velocidad	
x	0	Vx	0
y	0	Vy	0
z	0	Vz	0

Número de Impulsos Totales:

Fecha del Impulso número 1:

Año: Mes: Día:

Hora: Minuto: Segundo:

Valor del Impulso número 1:

Vx: Vy: Vz:

Figura 25: Definición de los datos de los distintos impulsos.

En la figura 25 se muestra como al insertar un número de impulsos mayor que 0 se pedirá la información de cada uno de ellos, más concretamente, la fecha en la que se produce, y las tres componentes de la velocidad que tendrá el cuerpo después del impulso. La activación de estos componentes se realizará mediante el código relacionado con el **contador de impulsos totales**, presentado en el anexo A (código 3). Tras introducir cada uno de los datos, se deberá de pulsar el botón *Guardar*, activado también debido a que el número de impulsos es mayor que cero.

Fechas Trayecto Sol ▼

Fecha Inicial **Fecha Final** **Vector de estado inicial**

Año	Mes	Día	Año	Mes	Día	Posición		Velocidad	
2020	12	23	2021	1	12	x	5e+08	Vx	23
Hora	Minuto	Segundo	Hora	Minuto	Segundo	y	-1.2e+07	Vy	12
0	0	0	0	0	0	z	1e+05	Vz	1

¿Maniobra Impulsiva?

Opciones de Integración
 Two Body
 Zonal Harmonics
 Atmósfera
 SRP Planetas

Vector de estado final

Posición		Velocidad	
x	0	Vx	0
y	0	Vy	0
z	0	Vz	0

Número de Impulsos Totales:

Fecha del Impulso número 2:

Año: Mes: Día:

Hora: Minuto: Segundo:

Valor del Impulso número 2:

Vx: Vy: Vz:

Figura 26: Opciones de integración para órbitas heliocéntricas.

Al pulsar el botón **Guardar** se generará un archivo con los datos del impulso que se utilizarán en el cálculo de la trayectoria. La primera línea del archivo se corresponde con los datos de la fecha en la que se produce el impulso mientras que en la segunda se guardan los datos del vector velocidad. Se generarán tantos archivos de texto como impulsos haya a lo largo de la trayectoria.

Además de esto, se borrarán de la pantalla los datos introducidos anteriormente para cada una de las variables del impulso, para así poder introducir los datos del siguiente. Y se aumentará el número que indica el impulso en el que se está trabajando, tanto al introducir la fecha como al introducir la velocidad. Esto asegura que el usuario sepa cuáles son los datos que debe de introducir en la aplicación, sin dar lugar a equivocaciones por no saber cuantas veces se ha pulsado el botón **Guardar**.

Finalmente, si no quedasen más impulsos por guardar, tras pulsar le botón por última vez, se desactivaría el componente y se activarían las opciones de integración posibles, dependiendo del sistema de referencia seleccionado, tal y como se muestra en la imagen 26.

Todas las funciones relacionadas con el botón **Guardar** se han programado mediante el código 4 del anexo A.

Por otra parte, como se puede observar en la figura 26, a la hora de seleccionar las opciones que se tienen para propagar la órbita se han empleado *Check Boxes*. Así, únicamente se seleccionarán aquellas opciones que se deseen y el resto no se tendrá en cuenta.

Como es de esperar, si se pretende realizar una propagación sin perturbaciones, es decir, seleccionar la opción de *Two Body*, el resto de opciones dejarán de estar disponibles. Por el contrario, si se selecciona alguna de las perturbaciones, se deshabilitará la opción de *Two Body*, y se mantendrán únicamente las perturbaciones para seleccionar aquellas que le interesen al usuario. Más adelante se verá como se realiza la selección de las opciones de integración y los datos que son necesarios para cada una de las perturbaciones.

Además, se puede apreciar como para las órbitas heliocéntricas no se tienen en cuenta todas las perturbaciones programadas. Esto es debido a que, tal y como se vio en apartados anteriores, para las orbitas heliocéntricas las perturbaciones causadas por el potencial gravitatorio terrestre y la atmósfera de la tierra, no tienen gran influencia, debido a las grandes distancias que separarán al cuerpo de la Tierra. Cabe destacar que esto es cierto, siempre y cuando se utilice el sistema geocéntrico para cuerpos que no orbiten en órbitas bajas de la Tierra. Es decir, se ha supuesto que si el cuerpo orbita alrededor de la Tierra, el usuario utilizará el sistema geocéntrico para la propagación.

Por otra parte, si se empleara el sistema geocéntrico se podrían tener en cuenta todas las perturbaciones posibles, tal y como muestra la figura 27.

Esta distinción entre heliocéntricas y geocéntricas se incluirá en cada una de las funciones que activen las **Opciones de Integración**, por lo que se podrá ver en los códigos presentados en el anexo A.

The screenshot shows a MATLAB application window with the following sections:

- Fechas Trayecto:** A dropdown menu set to 'Tierra'.
- Fecha Inicial:** Fields for Año (2020), Mes (12), Día (23), Hora (0), Minuto (0), Segundo (0).
- Fecha Final:** Fields for Año (2021), Mes (1), Día (12), Hora (0), Minuto (0), Segundo (0).
- Vector de estado inicial:**
 - Posición:** x (5e+04), y (-1200), z (1e+05).
 - Velocidad:** Vx (23), Vy (12), Vz (1).
- ¿Maniobra Impulsiva?:** 'Sí' button is active, 'No' button is disabled.
- Opciones de Integración:**
 - Two Body
 - Zonal Harmonics
 - Atmósfera
 - SRP
 - Planetas
- Other controls:** 'Número de Impulsos Totales' (2), 'Fecha del Impulso número' (2), 'Valor del Impulso número' (2), and 'Guardar Datos' button.
- Vector de estado final:** All position (x, y, z) and velocity (Vx, Vy, Vz) fields are set to 0.

Figura 27: Opciones de integración para órbitas geocéntricas.

En cuanto a la opción de no producirse ningún impulso a lo largo de la órbita representada en la figura 28 (pulsar el botón **No**), esta habilita directamente las opciones de integración, sin necesidad de indicar nada más que los datos iniciales ya introducidos.

También, deshabilita el botón **Sí** para que no sea posible pulsar sobre él, y que si se vuelve a pulsar sobre él mismo, se desactiven las opciones de integración y todas las componentes relacionadas con él. El código con sus distintas funciones será el número 5 del anexo A.

The screenshot shows the same MATLAB application window as Figure 27, but with the following changes:

- ¿Maniobra Impulsiva?:** 'No' button is active, 'Sí' button is disabled.
- Opciones de Integración:** All checkboxes are now visible and active:
 - Two Body
 - Zonal Harmonics
 - Atmósfera
 - SRP
 - Planetas
- Other controls:** 'Número de Impulsos Totales' (2), 'Fecha del Impulso número' (2), 'Valor del Impulso número' (2), and 'Cálculo' button.
- Vector de estado final:** All position (x, y, z) and velocity (Vx, Vy, Vz) fields are set to 0.

Figura 28: Elección sobre maniobra con impulsos o no. Al ser *no*, se habilitan las opciones de integración.

En cuanto a las **Opciones de Integración**, si se pretende propagar sin tener en cuenta ninguna perturbación, siguiendo el problema de dos cuerpos, se debe seleccionar la opción *Two Body*, que como se mencionó anteriormente deshabilita las perturbaciones. Además, esta opción habilita el botón de cálculo que permite iniciar la propagación al pulsarlo. El código que realiza dichas funciones es el código 6 del anexo A.

The screenshot shows a MATLAB application window titled 'Fechas Trayecto'. At the top right, there is a dropdown menu set to 'Tierra'. The interface is divided into several sections:

- Fecha Inicial:** Fields for Año (2020), Mes (12), and Día (23). Below these are fields for Hora (0), Minuto (0), and Segundo (0).
- Fecha Final:** Fields for Año (2021), Mes (1), and Día (12). Below these are fields for Hora (0), Minuto (0), and Segundo (0).
- Vector de estado inicial:** A table with two columns: 'Posición' and 'Velocidad'.

	Posición	Velocidad
x	5e+04	Vx: 23
y	-1200	Vy: 12
z	1e+05	Vz: 1
- ¿Maniobra Impulsiva?:** Two buttons, 'Si' and 'No', with 'No' selected.
- Opciones de Integración:** A list of checkboxes:
 - Two Body
 - Zonal Harmonics
 - Atmósfera
 - SRP
 - Planetas
- Cálculo:** A button to start the calculation.
- Vector de estado final:** A table with two columns: 'Posición' and 'Velocidad'.

	Posición	Velocidad
x	0	Vx: 0
y	0	Vy: 0
z	0	Vz: 0

Figura 29: Elección de la opción *Two Body*.

Por otra parte, si se desea realizar la propagación teniendo en cuenta una o varias perturbaciones, bastará con seleccionar a una de ellas para desactivar la opción de *Two Body*, y esta no volverá a estar disponible hasta que no estén todas las opciones, referentes a las perturbaciones, desactivadas. Asimismo, al activar alguna de las opciones de perturbación se harán visibles las opciones o datos que deberá definir el usuario.

En primer lugar, al seleccionar la perturbación causada por la no uniformidad del potencial gravitatorio terrestre, se activan una serie de *State Buttons* que permiten seleccionar cuales de los armónicos zonales se quieren tener en cuenta, desde el J_2 al J_6 . Además, habrá otro botón denominado *Todos* que si se elige, seleccionará todos los armónicos zonales, permitiendo al usuario no tener que ir seleccionando uno a uno.

En cuanto a la perturbación atmosférica, como se vio en apartados anteriores, es necesario que se conozca la relación área/masa, siendo este área lo que se encuentra perpendicular a la atmósfera terrestre, y el coeficiente de resistencia, por lo que el usuario deberá introducir ambos datos, en los *Text Fileds* que se activarán al seleccionar dicha perturbación. Estos datos no se guardarán en la función referente a esta *Check Box*, ya que se emplearán en los cálculos de la propagación.

En la figura 30 se observa como quedarían los datos a especificar por parte del usuario. Se puede ver como, al no estar ni seleccionado ninguno de los armónicos zonales, ni definidas las características necesarias para la perturbación atmosférica, el botón de *Cálculo* no está activado. Y este no se activará hasta que ambas cosas estén especificadas por parte del usuario, para

asegurar que no exista ningún error o problema al realizar la propagación.

El código que define las funciones de la *Check Box* de la perturbación causada por el potencial gravitatorio es el código 7, mientras que el de la perturbación atmosférica es del 8. Ambos se encuentran en el anexo A.

Además, también se han programado códigos para los *State Buttons* y los *Text Fields* de ambas perturbaciones, para asegurar que el botón de *Cálculo* se active cuando los datos de todas las perturbaciones estén bien definidos. Todos ellos se presentan en el anexo A.

Figura 30: Elección de la opción *Zonal Harmonics* y *Atmósfera*

Al seleccionar la opción de *SRP* se activarán dos *Text Fileds* en los que se pide al usuario introducir el coeficiente de reflectividad del cuerpo a estudiar y la relación área/masa, donde el área es la perpendicular a la radiación solar. Al igual que para la resistencia atmosférica, hasta que los dos valores sean distintos de cero, no se activará el botón de *Cálculo*.

Por último, para la perturbación causada por los planetas, se han tenido en cuenta todos los planetas del sistema solar, la Luna, y el Sol si el sistema en el que se trabaja es geocéntrico. En este sistema no se tendrá en cuenta el efecto de la Tierra, ya que se tiene en cuenta al propagar directamente la órbita. Al igual que el efecto del Sol en el sistema heliocéntrico.

La selección de los planetas a tener en cuenta en la perturbación se realiza mediante *State Buttons* que permiten la selección de cada uno de los cuerpos por separado. Además, al igual que para los armónicos zonales, existe un botón, *Todos*, que permite la selección de todos los cuerpos a la vez sin necesidad de ir seleccionando uno a uno.

Además, tal y como se realiza con los armónicos zonales, cuando al menos uno de los cuerpos (o de los armónicos zonales es su caso) esté seleccionado, se activará el botón de *Cálculo*. En la imagen 31 se observa como al tener valores válidos, distintos de cero, para la perturbación de *SRP* y seleccionado algún cuerpo en la opción de los planetas, el botón de *Cálculo* está activado.

Las funciones que definen el comportamiento de las *Check Boxes* explicadas anteriormente se presentan en los código 9 y 10, respectivamente. Además también se encuentran los códigos de cada uno de los elementos activados por dichas *Check Boxes*, necesarios para garantizar la correcta activación del botón *Cálculo*, y el óptimo funcionamiento del programa. Todo ello se encuentra recogido en el anexo A.

The screenshot shows a MATLAB application window with the following sections:

- Fechas Trayecto:** A dropdown menu set to 'Tierra'. Below it are two date selection groups: 'Fecha Inicial' (Year: 2020, Mes: 12, Día: 23, Hora: 0, Minuto: 0, Segundo: 0) and 'Fecha Final' (Year: 2021, Mes: 1, Día: 12, Hora: 0, Minuto: 0, Segundo: 0).
- Vector de estado inicial:** A table with two columns: 'Posición' and 'Velocidad'.

	Posición	Velocidad
x	5e+04	Vx: 23
y	-1200	Vy: 12
z	1e+05	Vz: 1
- ¿Maniobra Impulsiva?:** Two buttons: 'Si' (disabled) and 'No' (active).
- Opciones de Integración:** Three checkboxes: 'Two Body' (disabled), 'Zonal Harmonics' (disabled), and 'Atmósfera' (disabled).
- Datos Satélite:** Two checkboxes: 'SRP' (checked) and 'Planetas' (checked). Two input fields: 'A/m (SRP)' with value 1.2 and 'CR' with value 0.2.
- Cálculo:** A central button.
- Vector de estado final:** A table with two columns: 'Posición' and 'Velocidad'.

	Posición	Velocidad
x	0	Vx: 0
y	0	Vy: 0
z	0	Vz: 0
- Planetas:** A grid of buttons: 'Todos', 'Sol', 'Luna', 'Mercurio', 'Venus', 'Marte', 'Júpiter', 'Saturno', 'Urano', and 'Neptuno'. 'Marte' is currently selected.

Figura 31: Elección de la opción *Planetas* y *SRP*

Tras pulsar el botón de cálculo, se activarán los *Text Fields* del vector de estado final, comenzará la propagación y cuando se llegue al último instante, se mostrarán los resultados de la posición y velocidad finales por pantalla. Además se obtendrán dos archivos de texto con la posición y la velocidad a lo largo de la trayectoria. La función de este botón se presentará más adelante, junto con las funciones que desarrollan el código de la propagación.

5 PROGRAMA

Dentro del programa desarrollado existen dos tipos de funciones. En primer lugar, están aquellas que aseguran el correcto funcionamiento de la aplicación. Estas funciones se encuentran dentro de la aplicación, han sido explicadas en el apartado anterior y se pueden encontrar en el anexo A. Estas son necesarias debido a que no es posible garantizar el correcto uso de los distintos componentes por parte del usuario, por lo que se deberán evitar todos los posibles errores que se puedan producir.

Por otro lado, existen las funciones que realizan las operaciones y cálculos necesarios para obtener las trayectorias deseadas por el usuario. El código formado por estas funciones parte del diseño de la aplicación, es decir, se estructurará en base a cómo está desarrollada la aplicación.

Se comienza utilizando los datos proporcionados por el usuario para generar los datos iniciales de los que partirá la integración. En la integración, dependiendo de la elección realizada por el usuario será necesario emplear unas u otras funciones para tener en cuenta las distintas perturbaciones o no. Y tras ello se realiza el tratamiento de datos, para mostrar los resultados obtenidos.

5.1 Estructura

Partiendo del hecho de que la aplicación funciona correctamente, debido a las funciones comentadas en el apartado anterior, el código empleado para realizar las propagaciones de las trayectorias deseadas se estructura en distintas funciones. Estas funciones se aplican siguiendo el diagrama de flujo presentado en la figura 32.

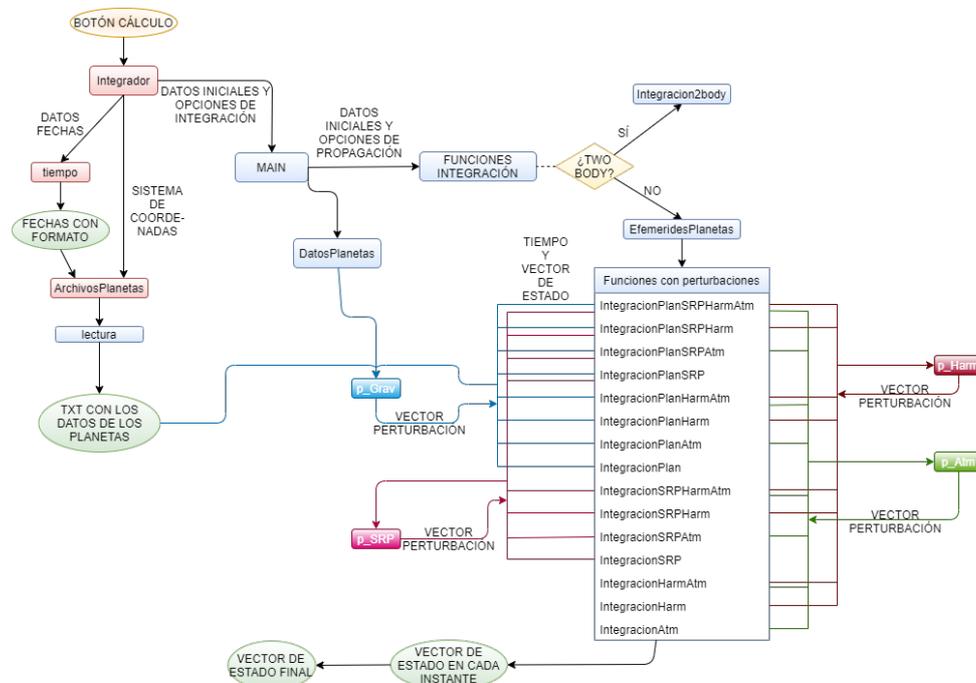


Figura 32: Diagrama de flujo de las distintas funciones utilizadas para desarrollar el código del programa.

Como se observa en la imagen anterior, el proceso de cálculo del código comienza al pulsar el botón de *Cálculo* de la aplicación. La función que controla dicho botón llama a la función *Integrador*, encargada de todo el proceso de cálculo de la trayectoria. En esta función se utilizan los datos utilizados por el usuario para emplear el método de integración adecuado.

En la figura 32 las funciones implementadas directamente en la aplicación se representan en color rojo, mientras que las representadas en azul son funciones creadas en MATLAB que han sido llamadas dentro de la aplicación. Aunque las funciones referentes a cada una de las perturbaciones estén todas ellas programadas en funciones de MATLAB, se han representado con distintos colores para diferenciar cuáles son las llamadas por cada una de las funciones de integración con perturbaciones. Por otra parte, los círculos verdes representan los resultados o datos obtenidos durante la ejecución del programa. El rombo amarillo es una cuestión, que permite dividir las opciones de integración. Y finalmente, el círculo naranja inicial representa el botón de *Cálculo*.

5.2 Funciones

Aunque el esquema seguido por el código ha sido representado anteriormente, para poder aclarar el funcionamiento del programa creado, es necesario definir lo que hace cada una de las funciones programadas. A continuación se presentan las distintas funciones dependiendo de sus labores, y siguiendo la estructura vista anteriormente.

5.2.1 Tratamiento de datos

Los datos introducidos por el usuario serán leídos por funciones de la aplicación, para obtener las variables necesarias para la correcta propagación.

Como se ha mencionado anteriormente, todo el proceso comienza al pulsar el botón de *Cálculo*. Por lo que la función asociada a dicho botón será el centro de todo el código. En ella se activan los *Text Fields* del vector de estado final, y se rellenan con los datos obtenidos de la función *Integrador*, la cual también es llamada por la función asociada al botón *Cálculo*. El código de dicha función es el código 32 que se encuentra en el anexo B.

Si la función asociada al botón de *Cálculo* es el centro del código, la función *Integrador* es la función principal, ya que en ella se obtienen los distintos datos necesarios para realizar la propagación, a partir de los datos introducidos por el usuario. El código de esta función se encuentra recogido en el anexo B, con el nombre de código 33.

En primer lugar, en esta función se utilizan los datos introducidos por el usuario para crear la variable vector de estado inicial (u_0), y las fechas en el formato adecuado, mediante la función *tiempo* (código 34 del anexo B). Además, se extrae el sistema de referencia de la selección realizada por el usuario.

A continuación, se divide la función en dos, dependiendo de si el usuario ha pulsado el botón **No** o el botón **Sí**. Para la opción **Sí**, se realizará un bucle de 1 a tantos impulsos como defina el usuario. En este bucle se realizaría lo mismo que si se pulsa el botón **No**, aunque partiendo de que los valores del vector de estado inicial tras cada impulso están formados por la posición final del último impulso y la velocidad definida por el usuario. Por lo que la preparación necesaria para la integración será exactamente la misma para ambos casos.

La función **Integrador** comienza llamando a **ArchivosPlanetas**, función utilizada para obtener los archivos de texto que contengan las coordenadas de los distintos cuerpos necesarios para la propagación, dentro del intervalo de tiempo estudiado, y en el sistema de referencia seleccionado. Para la obtención de los distintos archivos, se deben borrar todos los archivos anteriormente creados para posteriormente, crear los archivos de cada uno de los cuerpos mediante la función **lectura** creada en MATLAB (código 36 del anexo B).

El número de archivos creado dependerán de las perturbaciones seleccionadas. Si se selecciona la perturbación causada por un tercer cuerpo, se deberán crear únicamente los archivos de los cuerpos seleccionados. Por otra parte, si se selecciona la perturbación causada por la presión de radiación solar se deberán de generar los archivos con las coordenadas de la Luna, Venus, y la Tierra, si se trabaja en el sistema heliocéntrico o el Sol, si se trabaja en el sistema geocéntrico. Todo esto se ha programado según el código 35 presentado en el anexo B. Además, cabe destacar que esta función devuelve un vector *value* que contiene el nombre de los cuerpos que el usuario a querido tener en cuenta.

En la función **lectura** se realiza una lectura de los archivos del cuerpo estudiado, en las fechas y sistema de referencia seleccionado, para obtener sus efemérides. Junto con esta aplicación se adjuntan los archivos de texto que contienen dichas efemérides para ambos sistemas de referencia, desde 2020 hasta 2030. Son archivos que contienen las efemérides del cuerpo durante un mes en intervalos de un minuto. Si se pretende realizar trayectorias fuera de dicho intervalo se deberán descargar las efemérides de los planetas de la web de *HORIZONS* con el mismo formato que los anteriores, y nombrarlos según el esquema seguido: "*año + '-' + mes + Cuerpo + LetraDelSistemaDeCoordenadas + '.txt'*".

La lectura de los archivos, por lo tanto, se realiza directamente con los datos proporcionados por el usuario. Pero posteriormente se deberán de seleccionar los datos que interesen, ya que al tener una duración de un mes, es posible que no se necesiten todos los datos del archivo. Todo dependerá de las fechas indicadas por el usuario. Para ello, a partir de las fechas indicadas, se ha obtenido la fecha juliana, ya que en los archivos de texto descargados de *HORIZONS* también se proporciona dicho dato. Así se copiarán los datos que se encuentren entre estas dos fechas en el archivo generado del planeta seleccionado.

Tras obtener los distintos archivos de texto necesarios, en la función *Integrador* se obtiene el tiempo de integración a partir de la fecha inicial y la fecha final indicadas.

Posteriormente, se pasa a la definición de las opciones de integración. Estas no solo hacen referencia a los errores tomados, sino que también se obtendrán las opciones de integración descritas por el usuario, es decir, si la propagación se realizará mediante la resolución del problema de dos cuerpos o si, por el contrario, se tendrán en cuenta las distintas perturbaciones. Si se tienen en cuenta, entonces también se obtendrán los datos de cada una de ellas, necesarios para obtener las aceleraciones que producen. Con todo esto finalmente, se habrá definido un vector *integracion* en el cual se apuntarán las perturbaciones que el usuario desea tener en cuenta, un vector con los datos del satélite (coeficiente de resistencia, el coeficiente de reflectividad, y relación área/masa), y un vector en el que se guarden los valores de los armónicos zonales que se quieran incluir en la integración.

Una vez conocidos todos los datos necesarios, esta función llama a la función **Main** de MATLAB, la cual se encargará de la integración.

5.2.2 Integración

El proceso de integración comienza con la llamada realizada por la función **Integrador** a la función **Main** programada en MATLAB. Esta función parte de todos los datos obtenidos anteriormente, que son necesarios para la propagación de la trayectoria.

En primer lugar, tal y como se puede observar en el código 37 presentado en el anexo B, se definen numerosas variables globales, para que el resto de funciones tengan libre acceso a ellas. A estas variables se les asigna el valor directamente, debido a que han sido obtenidas previamente, o mediante otras funciones, como es el caso de las variables *mu*, *R* y *m*. Estas tres variables representan estructuras en las que se almacenan los parámetros gravitacionales, los radios medios, y las masas de los distintos planetas, respectivamente. Al ser datos preestablecidos, mediante la función **DatosPlanetas** se almacenan en las estructuras (código 38 del anexo B). En el caso de la estructura de los parámetros gravitacionales, se añade una variable más, *mu.Foco* cuyo valor variará en función de sistema de referencia elegido, igual a la de la Tierra si es geocéntrico y a la del Sol si es heliocéntrico.

Finalmente la función **Main** realiza la integración mediante el método **ode45** implementado en MATLAB. Este método necesita de una función en la que se planteó la integración, más concretamente, la técnica de propagación a utilizar. El hecho de tener varias opciones de integración, problema de dos cuerpos o método de Cowell con distintas perturbaciones y sus combinaciones, da lugar a plantear varias funciones a integrar, para no aumentar excesivamente el tiempo de integración.

Por lo que, se ha planteado un bucle en el cual se estudia el vector *integracion* creado en la función **Integrador**. Si el primer valor de este vector es igual a "*TwoBody*" se empleará la función en la que se resuelve el problema de dos cuerpos. Si no es así, se emplearán funciones que resuelven el método de Cowell, y se irá recorriendo dicho vector para ver cuáles son las perturbaciones a tener en cuenta, para saber qué función se deberá de emplear.

Además, en el caso de que el primer elemento del vector *integracion* no fuera igual a "*TwoBody*", y se hubiera seleccionado al menos una de las opciones siguientes: "*Planetas*" o "*SRP*", se llamará a la función **EfemeridesPlanetas**, cuyo código se presenta en el anexo B (código 40). Esta función lee los archivos de texto que contienen las efemérides de los distintos cuerpos, creados anteriormente, y guarda sus datos en la estructura global "*Efemerides*", utilizada por distintas funciones posteriormente.

En el anexo B se presentan los códigos de todas las funciones que plantean las distintas técnicas de propagación. Como simplemente se diferencian en el número de perturbaciones a tener en cuenta, únicamente se explicarán los códigos 39 y 41, que se corresponden con las funciones que plantean el problema de dos cuerpos y el método de Cowell con todas las perturbaciones, respectivamente.

La función **Integracion2body** partirá de la formulación del problema de dos cuerpos. En ella se introduce el vector de estado y el instante de tiempo en el que se está integrando. Además, se llama a la variable global *mu* para utilizar la variable *mu.Foco* anteriormente descrita y necesaria para la formulación de dicho problema. Igualando las variables del vector *du* a las derivadas de las seis componentes del vector se obtiene dicho vector, que será el que se integre mediante el método **ode45**. Cabe destacar que las derivadas de las componentes de la velocidad se igualan a la expresión 3.16, que describe el problema de dos cuerpos.

En cuanto a la función **IntegracionPlanSRPHarmAtm**, realiza exactamente lo mismo que la

función anteriormente descrita, añadiendo todas las perturbaciones que han sido estudiadas, mediante el método de Cowell. Por lo que la expresión de las derivadas de la velocidad será la mostrada en la ecuación 3.17, que describe el método de Cowell.

Las aceleraciones causadas por las perturbación se obtendrán mediante funciones que se explicarán detalladamente más adelante. Pero es importante destacar que para la perturbación causada por un tercer cuerpo, se ha empleado un bucle para ir sumando las perturbaciones causadas por cada uno de los cuerpos. Para ello se hace uso de la variable *values*, igual al vector *value* obtenido de la función **ArchivosPlanetas**, que contiene el nombre de los distintos cuerpos que el usuario quiere tener en cuenta. Y la perturbación total producida por las distintas perturbaciones será la suma de todas sus aceleraciones en ese instante de tiempo.

Finalmente, si se están empleando el sistema geocéntrico, tras obtener el vector de estado a lo largo de la órbita, se utilizará la función **plotter**, la cual permite crear una gráfica con la imagen de la Tierra, en la cual se represente la trayectoria seguida por la nave a lo largo de la propagación. Esto permite visualizar de manera más clara las órbitas generadas por la nave. Este código utiliza la función creada por Clay M. Thompson y Will Campbell **earth_sphere** para generar la imagen de la Tierra. Ambas funciones se presenta en el anexo B (códigos 56 y 57, respectivamente).

5.2.3 Perturbaciones

Existe cuatro funciones principales dedicadas a la obtención de cada una de las perturbaciones, que hacen uso de otras para conseguir datos que necesitan. Todas ellas parten del vector de estado y del instante de tiempo en el que se está integrando.

La primera es la función **p_Grav**, que obtiene la perturbación causada por un tercer cuerpo. Hace uso del nombre de este tercer cuerpo para obtener los datos necesarios para calcular la aceleración perturbadora. Por otra parte, la función **p_SRP** obtiene la aceleración causada por la presión de radiación solar, a partir de los datos del coeficiente de reflectividad y la relación área/masa definidos por el usuario. Además hace uso de la función **shadow** para obtener la función *S* que indica la existencia o no de un eclipse en ese instante de tiempo. Por otra parte, la función **p_Harm** obtiene la aceleración causada por el potencial gravitatorio terrestre a partir de los datos de la Tierra necesarios y los armónicos zonales seleccionados. Estos armónicos tendrán de valor inicial 0, y si el usuario decide emplearlos tomarán su valor correspondiente para ser utilizados en la obtención de la aceleración perturbadora. Finalmente, la función **p_Atm** emplea el radio de la Tierra, el coeficiente de resistencia atmosférica y la relación área/masa, para el cálculo de la aceleración. Cabe destacar que también hace uso de la función **atmosfera** para calcular la densidad mediante interpolación exponencial. Los códigos de estas funciones son, respectivamente, el 58, el 59, el 60 y el 61 del anexo B.

Por último, es necesario mencionar que en algunas funciones se hace uso de las efemérides de los planetas para el cálculo de las aceleraciones. Para obtenerlas se emplea la función **read** que permite obtener las efemérides de los cuerpos en el instante más cercano al instante en el que se encuentra la integración.

6 VALIDACIÓN

Una vez desarrollada la aplicación se pasa a la validación de la misma, es decir, a la comprobación de su correcto funcionamiento. Esta validación se ha realizado tras comprobar que la aplicación es consistente y que no se producen errores en la introducción de datos. Por lo que, en este apartado, únicamente se ha estudiado el correcto funcionamiento de las funciones de integración y de la utilización de maniobras impulsivas a lo largo de la trayectoria realizada por el cuerpo estudiado. Esta validación se ha realizado para cada una de las propagaciones posibles: problema de dos cuerpos y método de Cowell con cada una de las cuatro perturbaciones consideradas.

Todas estas comprobaciones se han realizado mediante la comparación de los datos obtenidos con la aplicación con los datos obtenidos del propagador *GMAT*. Este propagador permite la simulación de cualquier tipo de órbitas, tanto geocéntricas como heliocéntricas, y permite tener en cuenta las cuatro perturbaciones consideradas en este trabajo, combinándolas como prefiera el usuario.

Además, todas ellas parten de la misma posición inicial del satélite simulado, mostrado en la ecuación 6.1. Y simulan el mismo periodo de tiempo, 10 días, desde el 1 de agosto al 11 de agosto del 2020, comenzando y terminando a las doce de la noche de ambos días.

$$\bar{r} = (8000, 0, 6000) \text{ km} \qquad \bar{v} = (0, 7, 0) \text{ km/s} \qquad (6.1)$$

El código encargado de realizar todas las comprobaciones se presenta en el anexo B, y es el número 65.

6.1 Problema de dos cuerpos: función *Integracion2body*

En primer lugar, se ha simulado una órbita ideal sin ningún tipo de perturbación, para poder así comprobar que el planteamiento del problema de dos cuerpos, del que parte el método de Cowell, y el desarrollo de la aplicación estaban correctamente programados. Tras realizar la comprobación, se han obtenido las gráficas 33 y 34.

En ellas se puede ver como el error máximo producido va aumentando a lo largo del tiempo. El valor del error oscila rápidamente, cosa que se debe a la cercanía del satélite a la Tierra, es decir, el error varía debido a que es mucho menor cuando está cerca del perigeo y al ser una órbita de poca altitud el satélite pasa por dicho punto con mayor frecuencia.

Por otra parte, también se observa como los valores máximos de los errores obtenidos no son altos. No llegando a los 3 km en el error absoluto, y estando cercano al 0.025 % para el error relativo.

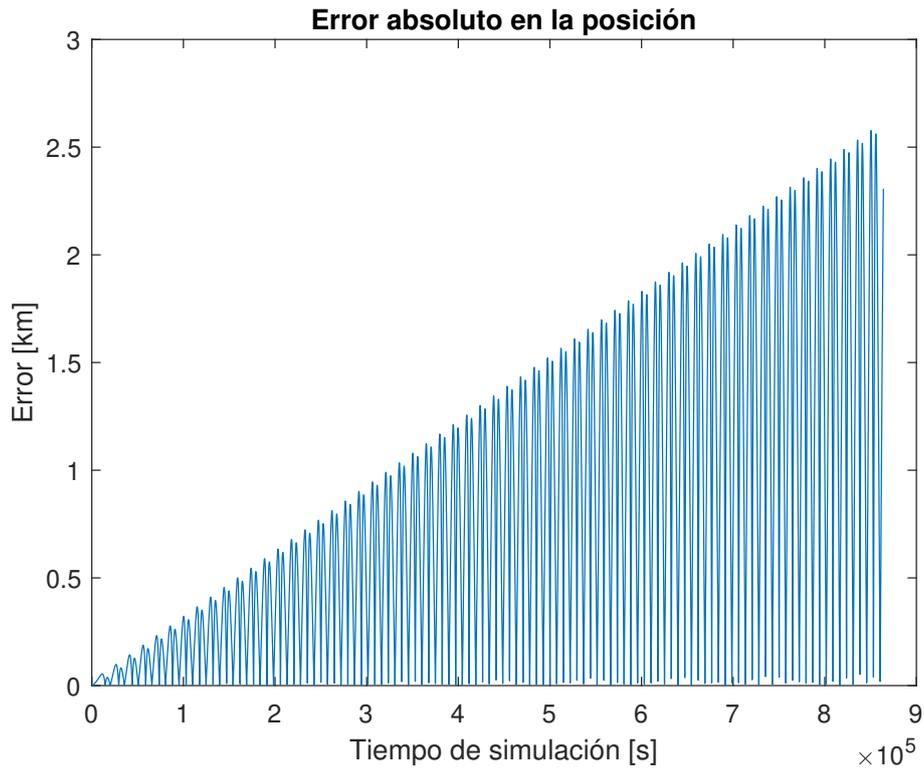


Figura 33: Error absoluto obtenido en el valor de la posición a lo largo de la trayectoria, en la comparación con los resultados del propagador *GMAT*, para una simulación sin perturbaciones.

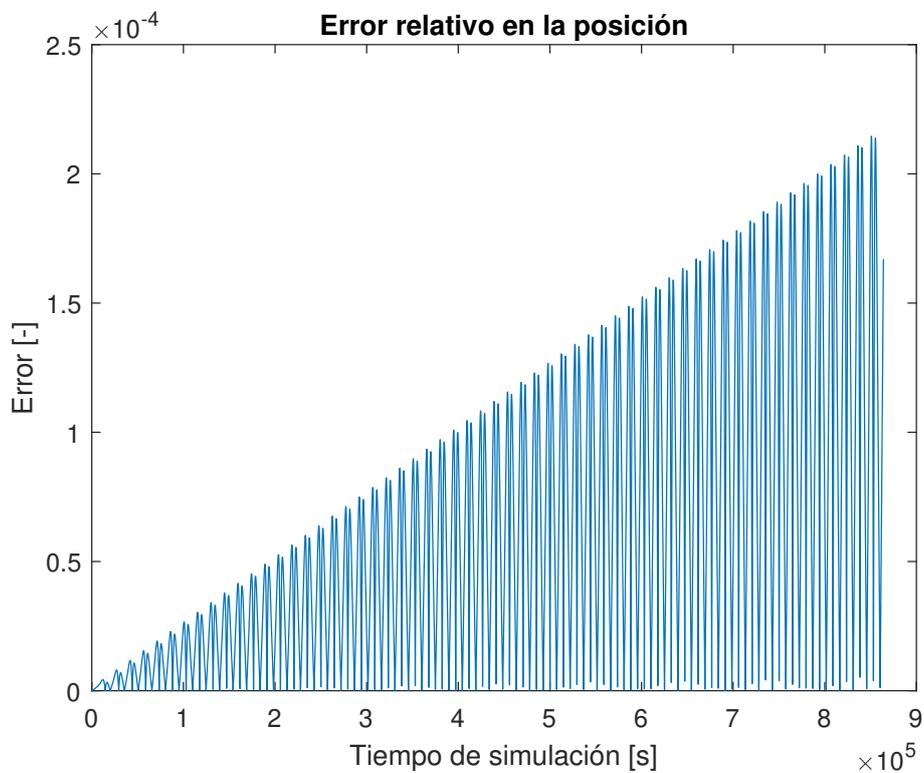


Figura 34: Error relativo obtenido en el valor de la posición a lo largo de la trayectoria en la comparación con los resultados del propagador *GMAT*, para una simulación sin perturbaciones.

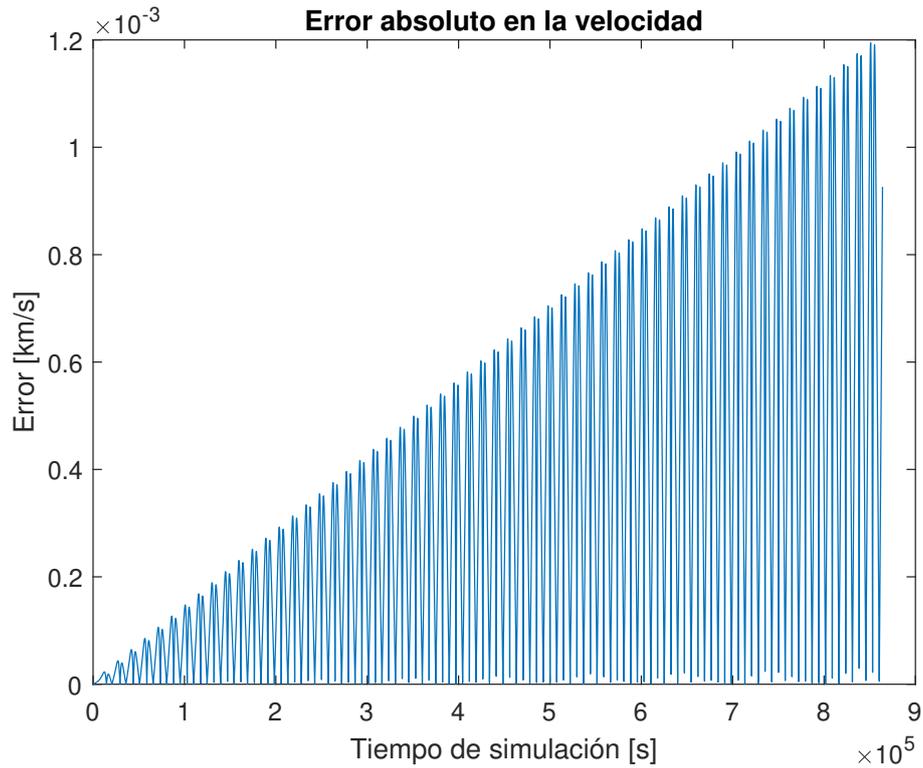


Figura 35: Error absoluto obtenido en el valor de la velocidad a lo largo de la trayectoria en la comparación con los resultados del propagador *GMAT*, para una simulación sin perturbaciones.

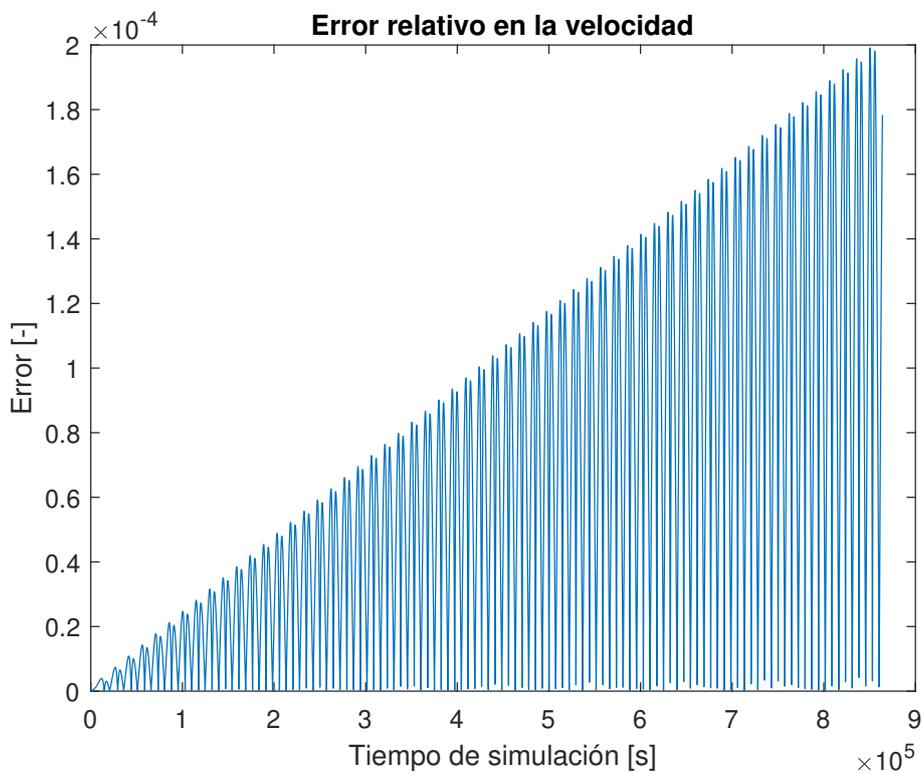


Figura 36: Error relativo obtenido en el valor de la velocidad a lo largo de la trayectoria en la comparación con los resultados del propagador *GMAT*, para una simulación sin perturbaciones.

En cuanto a los errores en la velocidad del satélite, tal y como se puede ver en las figuras 35 y 36 siguen el mismo patrón que las gráficas de la posición, produciéndose también variaciones rápidas en el error obtenido, pero manteniendo un aumento casi lineal del error máximo.

En lo referente a los valores de error obtenidos, tampoco son altos en este caso ya que rondan los $1.2 \cdot 10^{-3}$ km/s para el error absoluto y un 0.02 % en el error relativo.

Todos estos resultados aseguran el buen comportamiento del propagador programado en la aplicación. Por lo que se continuará probando el resto de opciones.

6.2 Método de Cowell

Tras comprobar que la base del método de Cowell, el problema de dos cuerpos, funciona, se pasa a realizar la validación de las distintas perturbaciones. Como se ha comentado esto se ha realizado comparando los valores obtenidos con los calculados por *GMAT*. La introducción de estas perturbaciones en *GMAT* se explicará más detalladamente en cada apartado.

6.2.1 Perturbación causada por los planetas, la Luna y el Sol: función IntegracionPlan

La validación del método de Cowell se ha iniciado considerando la perturbación causada por la gravedad de los planetas, la Luna y el Sol. La implementación de esta perturbación en la aplicación permite seleccionar los cuerpos que se quieren tener en cuenta, al igual que pasa en *GMAT*. En este caso al propagarse una órbita geocéntrica se ha decidido tener en cuenta la gravedad de la Luna, Marte y Júpiter.

Tras obtener los resultados con *GMAT* y con la aplicación programada, se comparan para ver las diferencias entre ellos. En las gráficas 37 y 38 se presentan los errores obtenidos en la obtención de la posición.

En ambas gráficas se observa como el valor del error máximo aumenta de manera lineal, aunque los errores obtenidos oscilan rápidamente entre valores máximos y mínimos (muy cercanos a cero), debido a lo explicado anteriormente. Por otra parte, los valores obtenidos son prácticamente iguales a los obtenidos para el problema de dos cuerpos, incluso algo menores.

Todo lo anterior también ocurre en los errores obtenidos en la estimación de la velocidad presentados en las figuras 39 y 40. Por lo que se puede afirmar que la implementación de la perturbación causada por la gravedad de los distintos cuerpos mencionados es correcta.

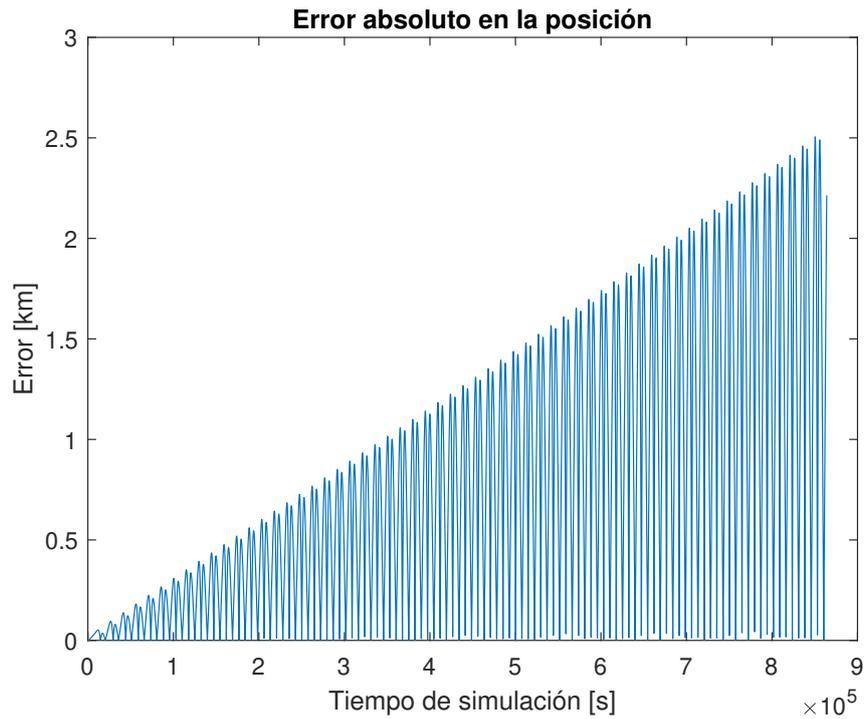


Figura 37: Error absoluto obtenido en el valor de la posición a lo largo de la trayectoria, en la comparación con los resultados del propagador *GMAT*, para una simulación considerando la gravedad de la Luna, Marte y Júpiter.

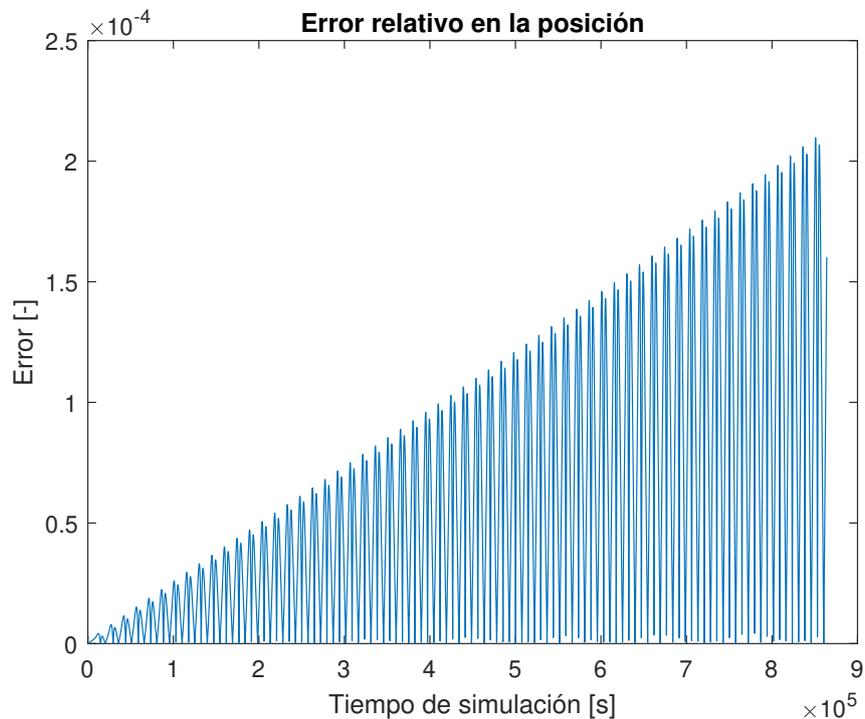


Figura 38: Error relativo obtenido en el valor de la posición a lo largo de la trayectoria en la comparación con los resultados del propagador *GMAT*, para una simulación considerando la gravedad de la Luna, Marte y Júpiter.

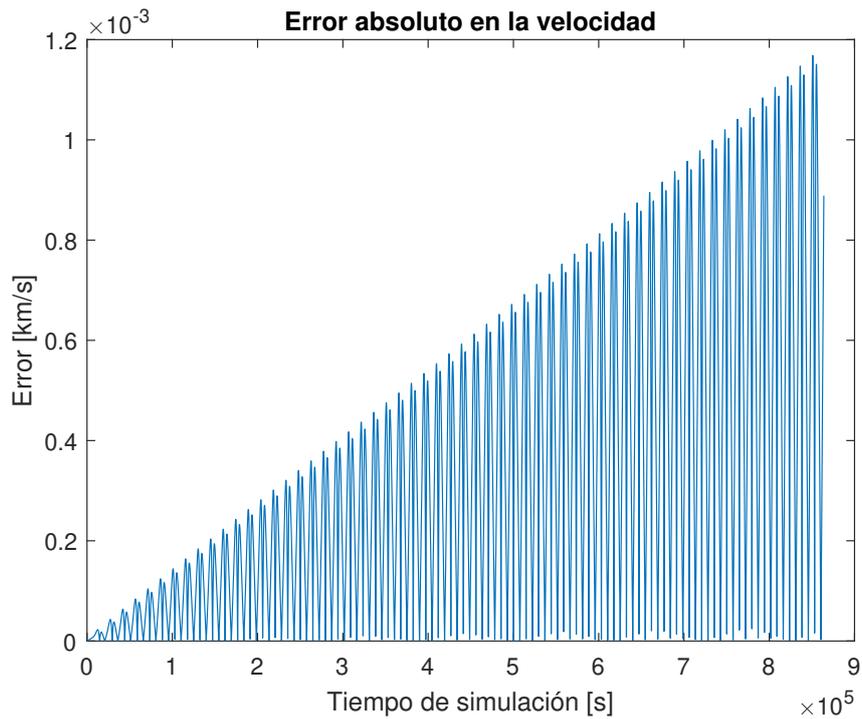


Figura 39: Error absoluto obtenido en el valor de la velocidad a lo largo de la trayectoria en la comparación con los resultados del propagador *GMAT*, para una simulación considerando la gravedad de la Luna, Marte y Júpiter.

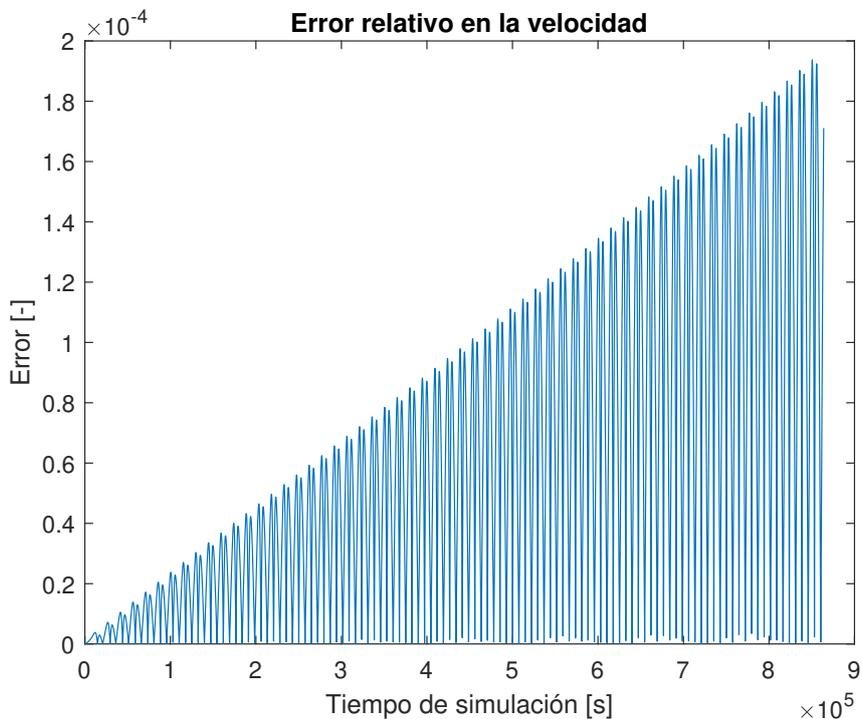


Figura 40: Error relativo obtenido en el valor de la velocidad a lo largo de la trayectoria en la comparación con los resultados del propagador *GMAT*, para una simulación considerando la gravedad de la Luna, Marte y Júpiter.

6.2.2 Perturbación causada por la presión de radiación solar: función IntegracionSRP

A continuación, se ha pasado a validar la implementación de la perturbación causada por la presión de radiación solar. Para ello se han tomado como datos del satélite una relación área (perpendicular a la radiación solar)/masa de 1 m^2 , y un coeficiente de reflectividad de 1.8. Estos datos han sido introducidos tanto en la aplicación como en el propagador de *GMAT*, en el cual se pueden definir las distintas características del satélite. Además se ha supuesto un modelo esférico en ambos propagadores, para que exista la mayor semejanza posible entre ambos.

Con las dos propagaciones realizadas, se estudian los errores entre ambas. En la figura 41 y 42 se presentan los errores cometidos en la obtención de la posición. Se observa como el error máximo aumenta con el tiempo, aunque en este caso no de manera lineal. Aún así, los valores de los errores máximos obtenidos no son altos, menos de 5 km en el error absoluto que se traduce en menos de un 0.045 % de error relativo máximo.

En cuanto a la velocidad (figuras 43 y 44), como es de esperar, sigue la misma tendencia que la posición. Obteniéndose un error absoluto máximo cercano a 0.0025 km/s, y un error relativo de 0.04 %.

Todos estos resultados permiten confirmar la validez del propagador programado en la aplicación.

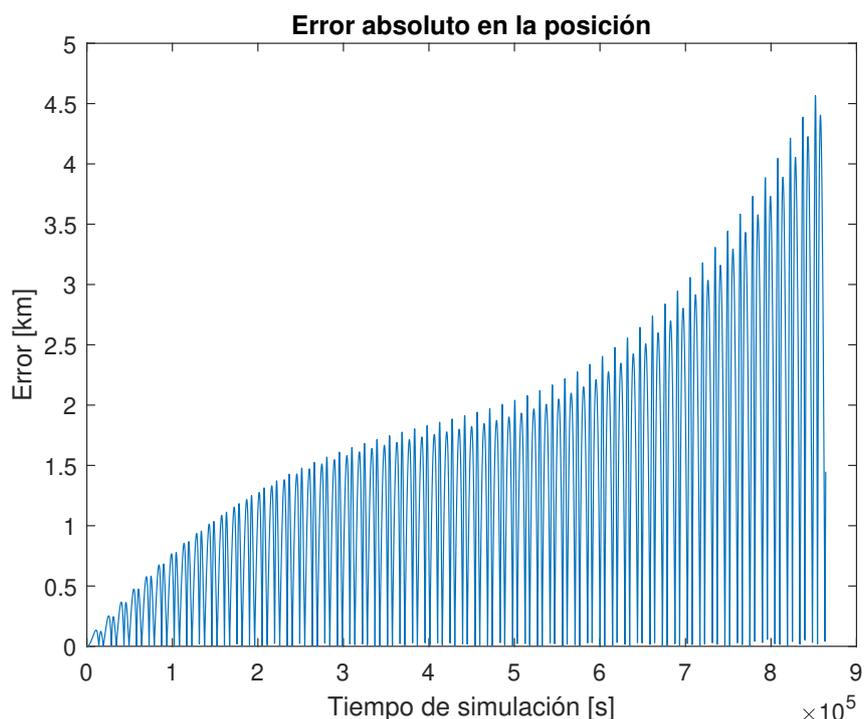


Figura 41: Error absoluto obtenido en el valor de la posición a lo largo de la trayectoria, en la comparación con los resultados del propagador *GMAT*, para una simulación considerando la presión de radiación solar.

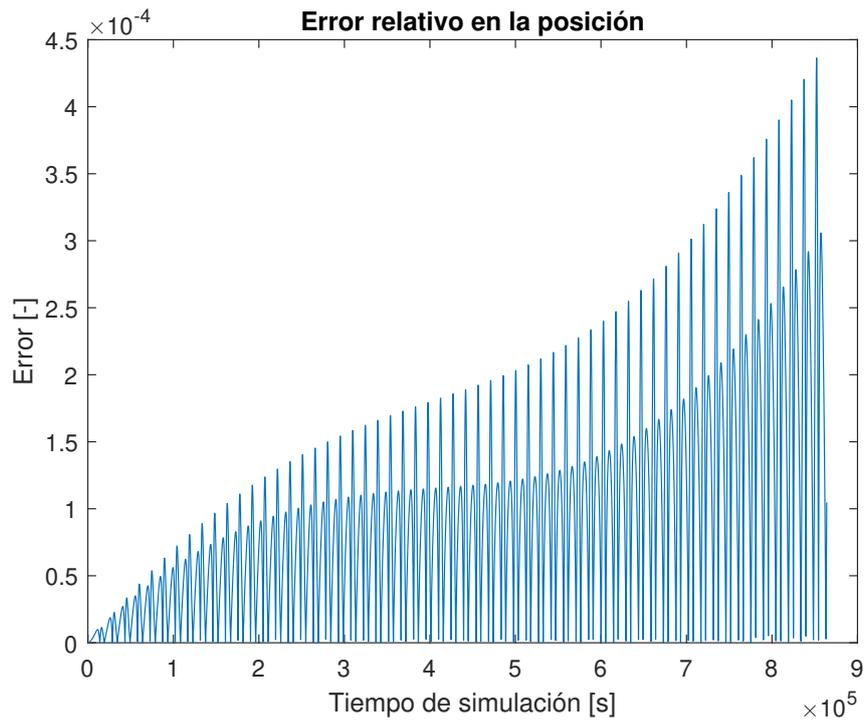


Figura 42: Error relativo obtenido en el valor de la posición a lo largo de la trayectoria en la comparación con los resultados del propagador *GMAT*, para una simulación considerando la presión de radiación solar.

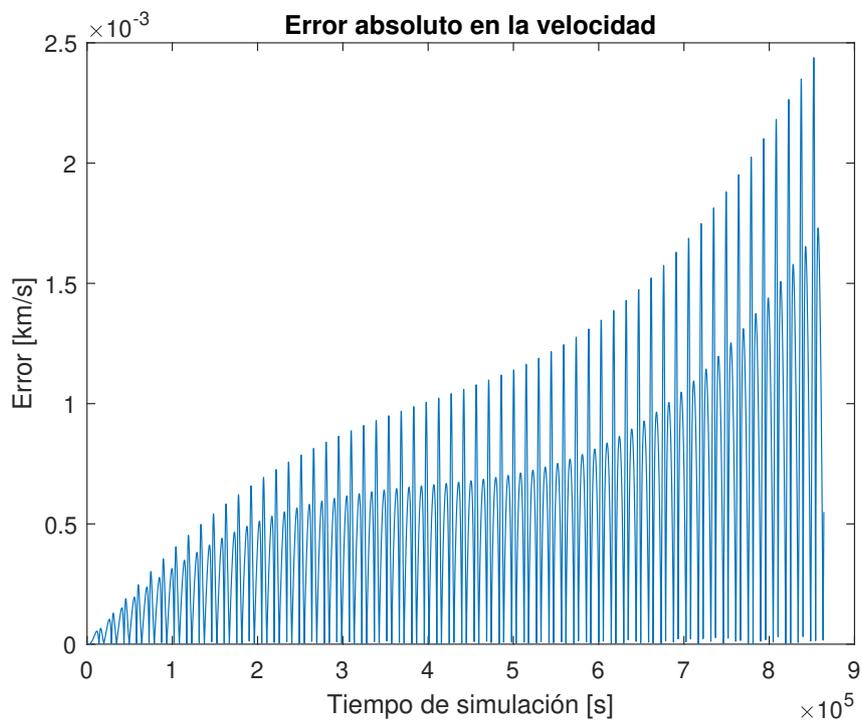


Figura 43: Error absoluto obtenido en el valor de la velocidad a lo largo de la trayectoria en la comparación con los resultados del propagador *GMAT*, para una simulación considerando la presión de radiación solar.

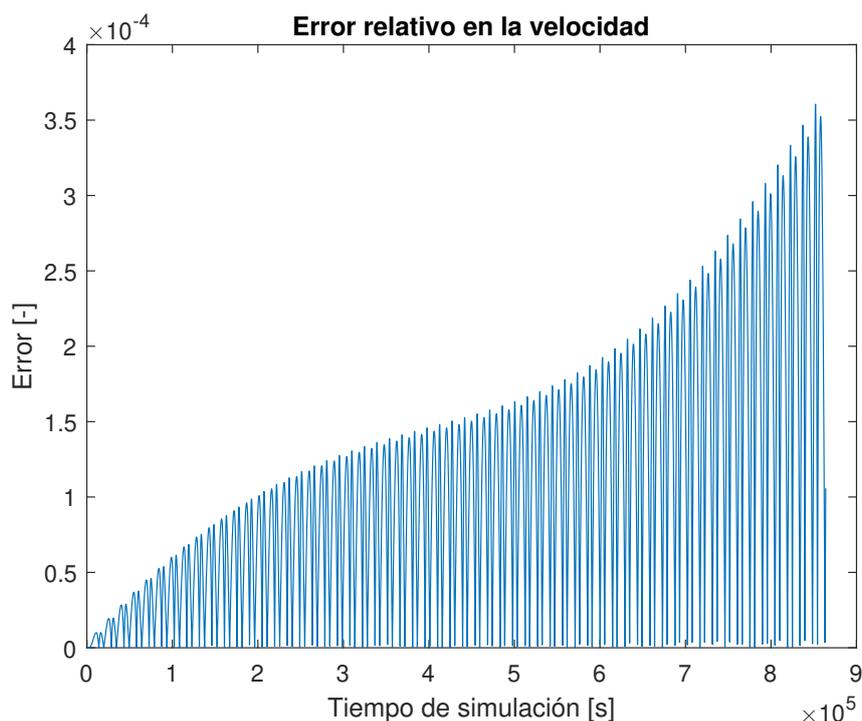


Figura 44: Error relativo obtenido en el valor de la velocidad a lo largo de la trayectoria en la comparación con los resultados del propagador *GMAT*, para una simulación considerando la presión de radiación solar.

6.2.3 Perturbación causada por la no uniformidad del campo gravitatorio terrestre: función IntegracionHarm

En cuanto a la perturbación causada por el campo gravitatorio terrestre, definida por los armónicos zonales, está se ha implementado en ambos propagadores únicamente teniendo en cuenta el J_2 . Cabe destacar que en *GMAT* la implementación se ha realizado seleccionando el modelo JMG2 y, dentro de este utilizando las variables de orden 0 y grado 2.

Los errores obtenidos en el valor de la posición a lo largo de la propagación se muestran en las gráficas 45 y 46. La evolución de los valores máximos de los errores en este caso tampoco es lineal, aunque se asemeja más a una aproximación lineal que en el caso de la perturbación causada por los planetas. Por otra parte, se observa que los errores obtenidos son mayores que en el resto de casos, esto es debido a que el modelo utilizado por *GMAT* para el cálculo de las aceleraciones producidas por esta perturbación es distinto al empleado por la aplicación, aunque para ambos casos se haya seleccionado el J_2 . Por todo ello se observan errores absolutos máximos cercanos a los 60 km y errores relativos máximos que rondan el 0.5%.

En las velocidades (figuras 47 y 48) se observa también un aumento de los errores máximos, siendo el absoluto menor de 0.03 km/s y el relativo algo mayor al 0.45%

Aunque se observan errores mayores que en otros casos, estos no superan el 1.5% en ningún caso, por lo que se ha decidido tomar esta implementación como buena teniendo en cuenta que los modelos tomados por ambos propagadores no son exactamente los mismos.

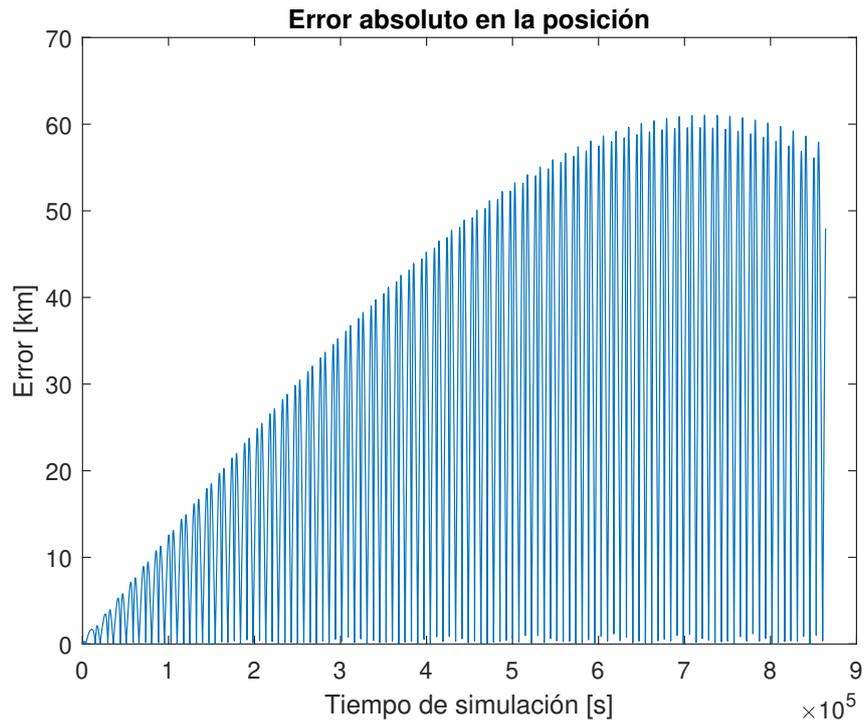


Figura 45: Error absoluto obtenido en el valor de la posición a lo largo de la trayectoria, en la comparación con los resultados del propagador *GMAT*, para una simulación considerando los armónicos zonales.

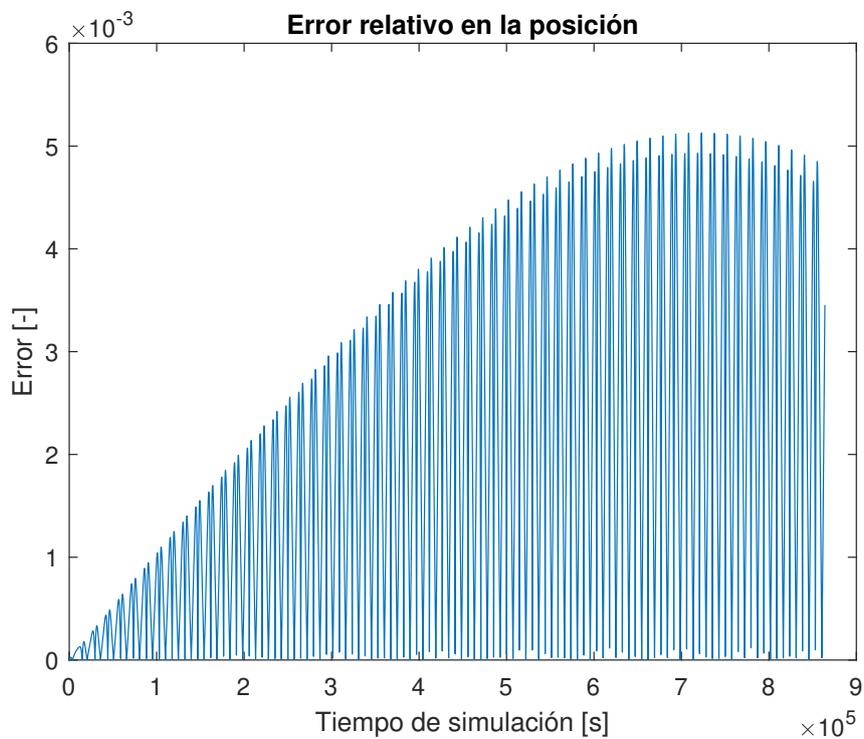


Figura 46: Error relativo obtenido en el valor de la posición a lo largo de la trayectoria en la comparación con los resultados del propagador *GMAT*, para una simulación considerando los armónicos zonales.

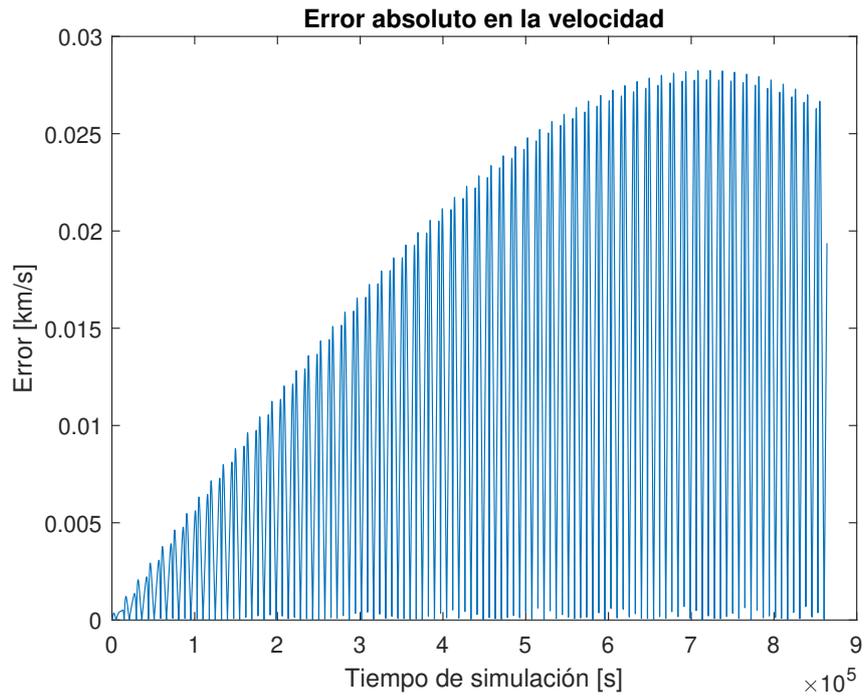


Figura 47: Error absoluto obtenido en el valor de la velocidad a lo largo de la trayectoria en la comparación con los resultados del propagador *GMAT*, para una simulación considerando los armónicos zonales.

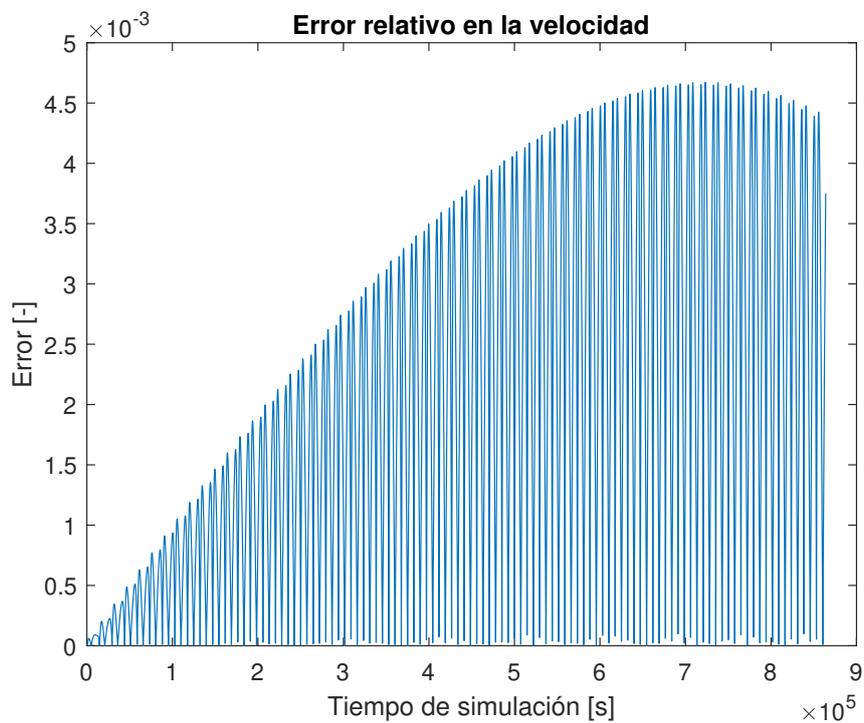


Figura 48: Error relativo obtenido en el valor de la velocidad a lo largo de la trayectoria en la comparación con los resultados del propagador *GMAT*, para una simulación considerando los armónicos zonales.

6.2.4 Perturbación causada por la atmósfera terrestre: función IntegratioAtm

Por último, se han estudiado los errores obtenidos con la perturbación causada por la atmósfera terrestre. Para ello, se han tomado como datos del satélite una relación área (perpendicular a la atmósfera)/masa de 15 m^2 y un coeficiente de resistencia de 2.2. Además, cabe destacar que para propagar teniendo en cuenta esta perturbación en *GMAT* es necesario definir un campo gravitatorio terrestre y, con él, los armónicos zonales a tener en cuenta. Por lo que se ha seleccionado nuevamente el J_2 .

En cuanto al modelo atmosférico propiamente dicho, *GMAT* no permite emplear el modelo exponencial, únicamente tiene implementados el modelo **MSISE-90** que permite estudiar la atmósfera entre 0 y 700 km de altitud y el modelo **Jacchia-Roberts** que estudia la atmósfera entre 90 y 2500 km de altitud. Debido a la baja altura de la órbita y a la utilización del modelo exponencial por parte de la aplicación, se ha decidido emplear el modelo **MSISE-90**.

Por todo esto, tal y como se observa en las figuras de los errores (figuras 49 y 50 para la posición, y 51 y 52 para la velocidad), se han obtenido errores absolutos máximos de 30 km para la posición y 0.014 km/s para la velocidad, que se traducen en unos errores relativos de 0.25 % para la posición y de algo más que 0.22 % para la velocidad.

Salvando las diferencias entre los métodos empleados, los errores obtenidos no son excesivamente elevados y se deben claramente a la diferencia entre los modelos de perturbación empleados. Por lo que se ha tomado como correcta la implementación de la perturbación atmosférica.

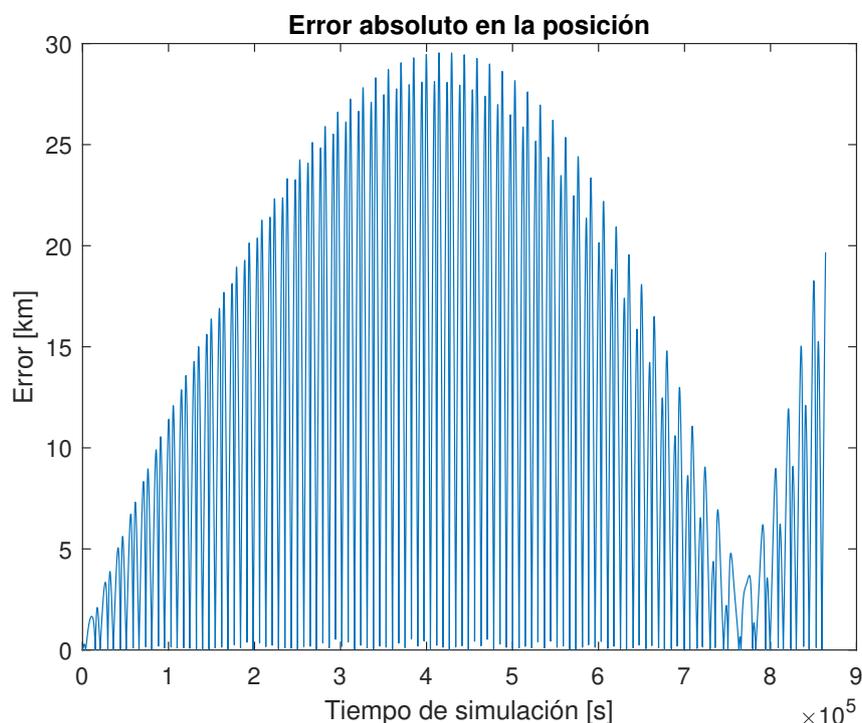


Figura 49: Error absoluto obtenido en el valor de la posición a lo largo de la trayectoria, en la comparación con los resultados del propagador *GMAT*, para una simulación considerando la resistencia atmosférica.

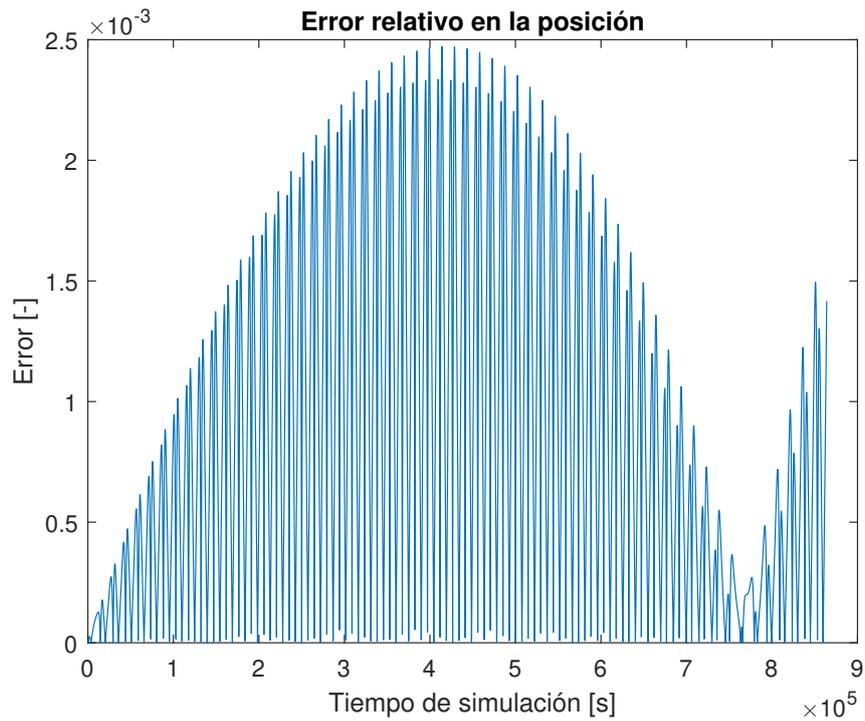


Figura 50: Error relativo obtenido en el valor de la posición a lo largo de la trayectoria en la comparación con los resultados del propagador *GMAT*, para una simulación considerando la resistencia atmosférica.

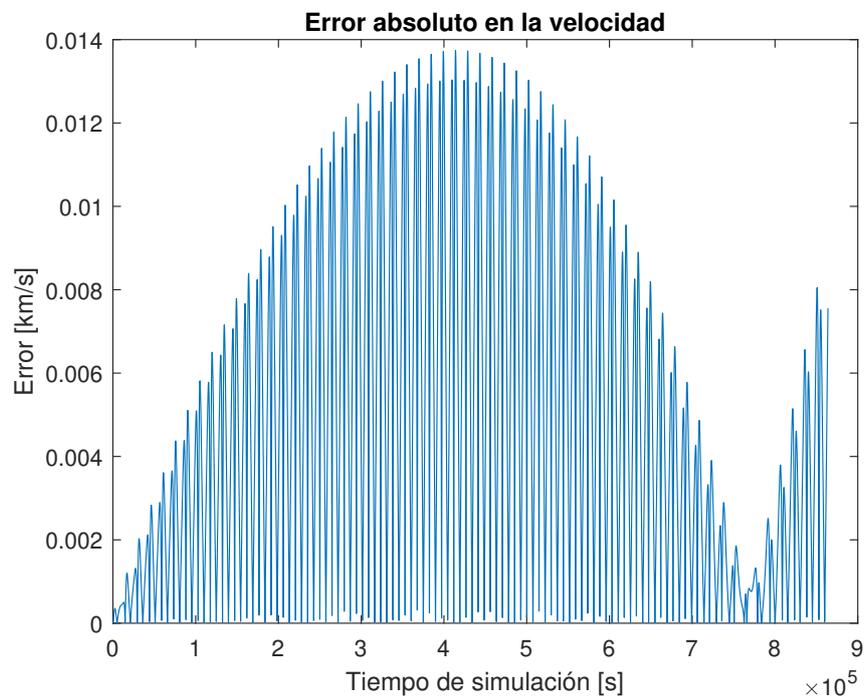


Figura 51: Error absoluto obtenido en el valor de la velocidad a lo largo de la trayectoria en la comparación con los resultados del propagador *GMAT*, para una simulación considerando la resistencia atmosférica.

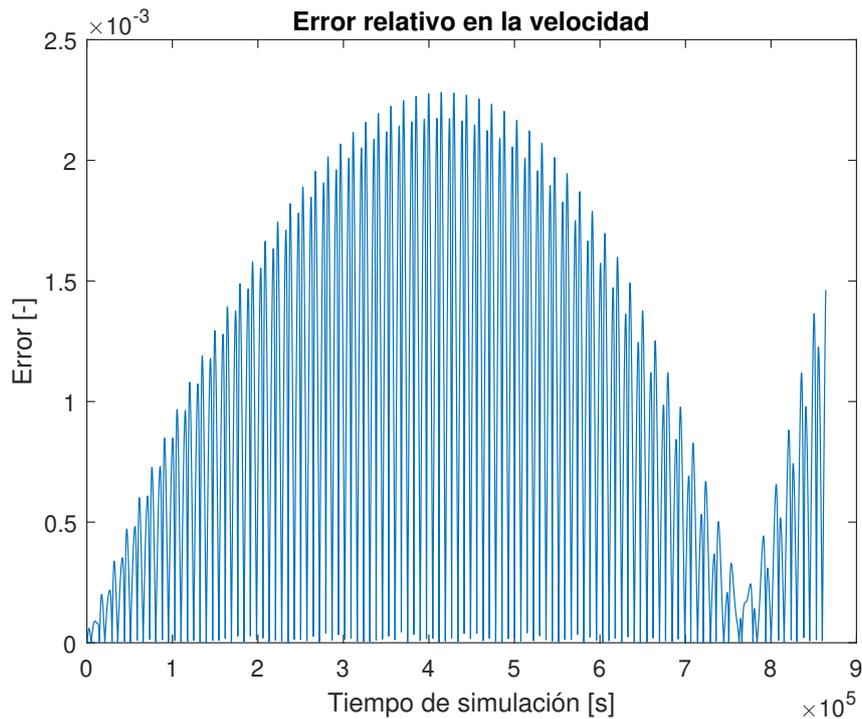


Figura 52: Error relativo obtenido en el valor de la velocidad a lo largo de la trayectoria en la comparación con los resultados del propagador *GMAT*, para una simulación considerando la resistencia atmosférica.

6.2.5 Todas las perturbaciones consideradas: función `IntegracionPlanSRPHarmAtm`

Finalmente, se han implementado las cuatro perturbaciones estudiadas de manera conjunta y comparado los resultados obtenidos con la aplicación, con los calculados por *GMAT*. Los datos empleados son exactamente los mismo que los utilizados para las validaciones anteriores, por lo que las características del satélite serán las mismas y se estudiarán las perturbaciones causadas por los planetas y armónicos zonales seleccionados anteriormente.

Con todo esto, se han obtenido los errores representados en las gráficas 53 y 54 para la posición, y 55 y 56 para la velocidad. En todas ellas se observa una evolución del error variable a lo largo del tiempo, llegando a su pico al final de la propagación. Además, tal y como era previsible, se obtienen errores mayores que en el resto de validaciones, debido a la suma de los errores individuales de cada una de las perturbaciones estudiadas. En la posición estos errores llegan a ser de unos 150 km, es decir, cerca de un 1.4% de error relativo. Mientras que para la velocidad son de unos 0.07 km/s, algo menos 1.2%.

Aunque los errores sean mayores, no superan el 1.5% de error relativo, por lo que se ha decidido tomar como buena la implementación de todas las perturbaciones.

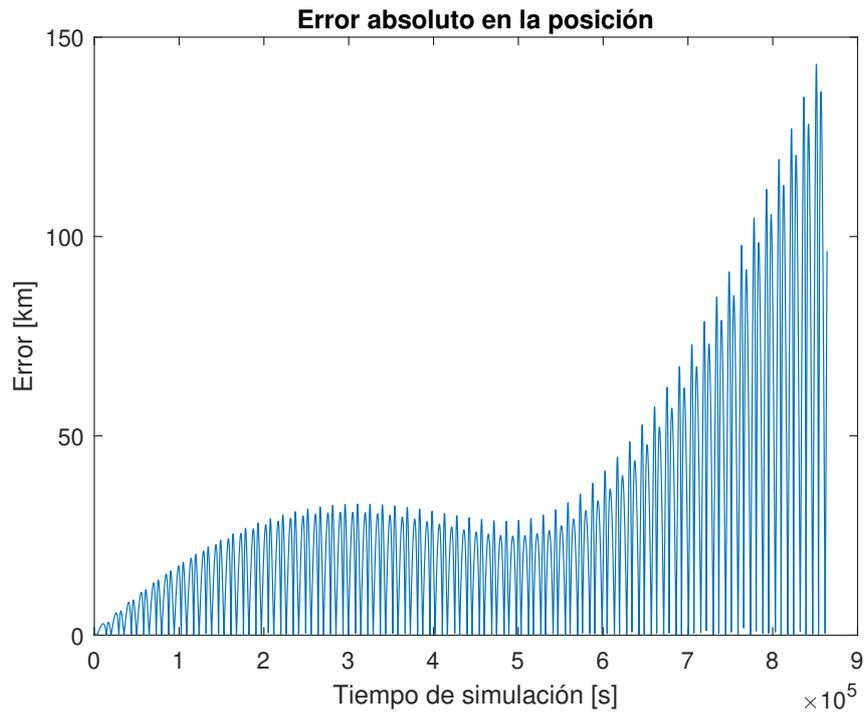


Figura 53: Error absoluto obtenido en el valor de la posición a lo largo de la trayectoria, en la comparación con los resultados del propagador *GMAT*, para una simulación considerando todas las perturbaciones estudiadas.

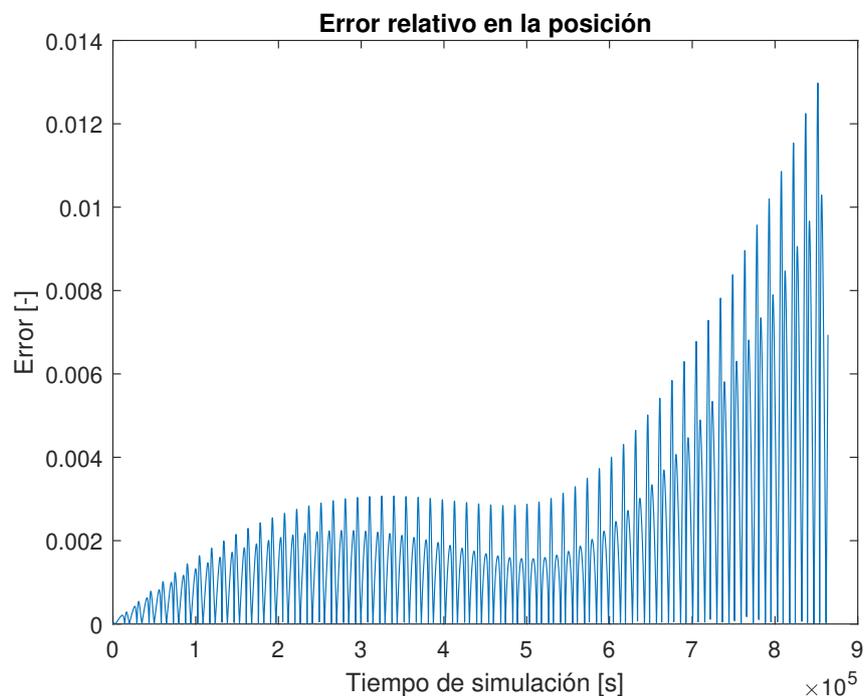


Figura 54: Error relativo obtenido en el valor de la posición a lo largo de la trayectoria en la comparación con los resultados del propagador *GMAT*, para una simulación considerando todas las perturbaciones estudiadas.

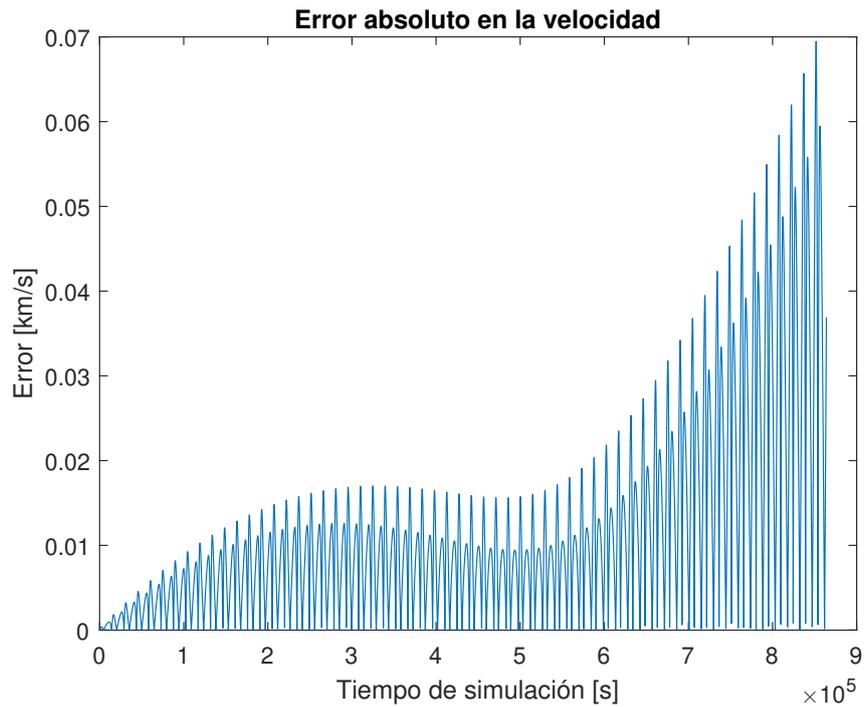


Figura 55: Error absoluto obtenido en el valor de la velocidad a lo largo de la trayectoria en la comparación con los resultados del propagador *GMAT*, para una simulación considerando todas las perturbaciones estudiadas.

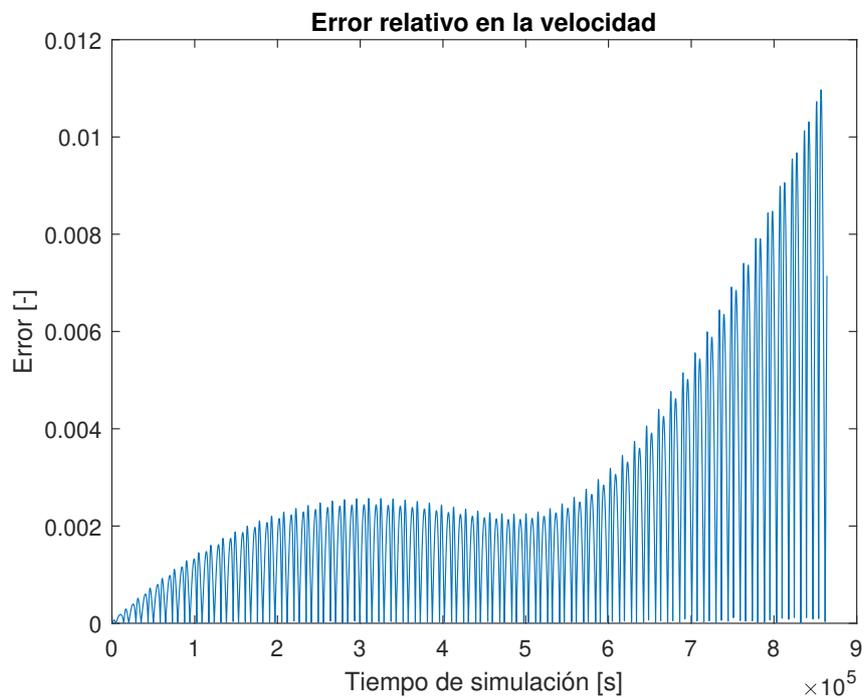


Figura 56: Error relativo obtenido en el valor de la velocidad a lo largo de la trayectoria en la comparación con los resultados del propagador *GMAT*, para una simulación considerando todas las perturbaciones estudiadas.

7 RESULTADOS

Tras la validación de los distintos integradores, se pasa a la obtención de distintas trayectorias mediante la aplicación programada. En primer lugar se presentará la propagación de la trayectoria de un satélite real que se encuentra en una órbita geocéntrica, y su comparativa con los valores reales de su trayectoria. A continuación, se han realizado dos maniobras impulsivas en coordenadas geocéntricas, más concretamente, un aumento de la altura del apogeo de la órbita, y un aumento del perigeo. Finalmente, se ha estudiado una trayectoria impulsiva en coordenadas heliocéntricas. Esta trayectoria será la generada por la *Mars2020* en su viaje desde la Tierra hasta Marte. Y se compararán los resultados obtenidos con los reales.

7.1 Trayectoria del satélite geocéntrico Terra

En este apartado se pretende propagar la trayectoria del satélite Terra, un satélite que orbita en órbita circular, polar, sincronizada con el Sol, y de baja altura alrededor de la Tierra (órbita LEO). Este satélite es un satélite diseñado por la NASA, que tiene como finalidad ayudar a comprender el clima y el cambio climático, mediante el estudio de las relaciones entre la atmósfera terrestre, la Tierra, el hielo, el océano y el equilibrio energético. Fue lanzado en el año 1999 y se prevé que siga en funcionamiento hasta más allá de 2023 [18]. A continuación se muestra una imagen de dicho satélite (figura 57):



Figura 57: Satélite Terra lanzado en 1999 por la NASA [18].

En cuanto a la propagación, para este caso se ha partido de la posición y velocidad de dicho satélite el 1 de marzo de 2020 a las doce de la noche (ecuación 7.1), y se ha propagado su órbita durante dos meses.

$$\begin{aligned} r_0 &= [-721.64924357, -3456.2101807, -6147.4108502] \text{ km} \\ v_0 &= [-5.4288517119, 4.741000445968, -2.02825050700] \text{ km/s} \end{aligned} \quad (7.1)$$

Para este caso, la aplicación se ha configurado tal y como se muestra en la figura 58. Se han tenido en cuenta las perturbaciones causadas por la no uniformidad del campo gravitatorio terrestre, la atmósfera terrestre, con los datos proporcionados por la NASA, y la gravedad de todos los planetas, el Sol y la Luna. No se ha empleado la perturbación causada por la presión de radiación solar, debido a falta de datos.

Además, en la figura mostrada a continuación también se muestra el vector de estado del satélite en el instante final de la propagación.

Fechas Trayecto Tierra ▼

Fecha Inicial **Fecha Final** **Vector de estado inicial**

Fecha Inicial			Fecha Final			Posición		Velocidad	
Año	Mes	Día	Año	Mes	Día	x	y	Vx	Vy
2020	3	1	2020	5	1	-721.6	-3456	-5.429	4.741
Hora	Minuto	Segundo	Hora	Minuto	Segundo	z		Vz	
0	0	0	0	0	0	-6147		-2.028	

¿Maniobra Impulsiva?
 Si No

Opciones de Integración
 Two Body
 Zonal Harmonics
 Atmósfera
 SRP Planetas

Datos Satélite
 Zonal Harmonics:

 A/m (Atm): CD:

Vector de estado final

Posición		Velocidad	
x	y	Vx	Vy
652.6	3490	5.478	-4.689
z		Vz	
6086		2.18	

Planetas

Figura 58: Datos en la aplicación para realizar la propagación de la trayectoria del satélite Terra.

Tras esta propagación se obtiene la imagen 59, en la que se muestra la trayectoria realizada por el satélite al rededor de la Tierra a lo largo de la propagación. Como se puede observar la órbita no se representa con una línea fina, cosa que ocurre cuando es siempre la misma, sino que parece ser más gruesa. Esto significa que la órbita ha sufrido variaciones a lo largo de la trayectoria propagada, y se ha movido formando distintas órbitas más o menos cercanas a la inicial.

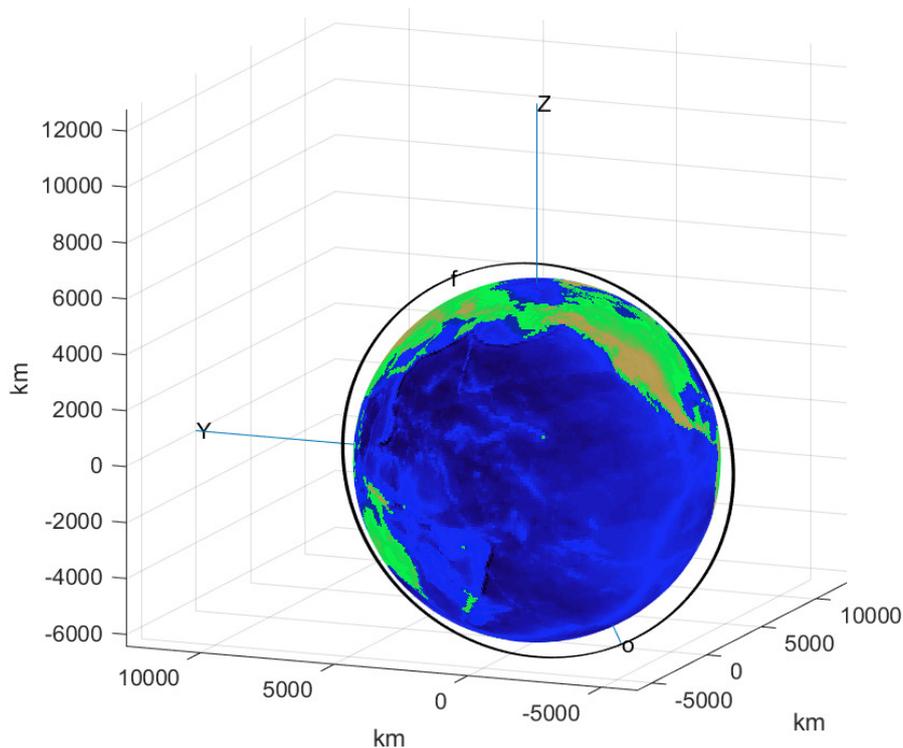


Figura 59: Trayectoria del satélite Terra desde el 1 de marzo al 1 de mayo de 2020.

Con los datos obtenidos de la posición y velocidad del satélite, se utiliza el código 66, mostrado en el anexo B, para calcular los errores con respecto a los datos obtenidos del *Horizons*. A continuación se muestran las gráficas que muestran los errores de posición (60 y 61) y de velocidad (62 y 63).

Cabe destacar que para la representación de dichas gráficas de manera clara, se ha decidido calcular el error entre valores obtenidos cada 3 horas, ya que así se mostrará mejor la evolución del error a lo largo del tiempo, sin tener demasiados puntos en la gráfica.

En todas las gráficas se observa como el error oscila a lo largo de la trayectoria. Se sabe que el satélite tiene un periodo de 99 min, lo que significa que realiza unas 14 órbitas por día. Esto da lugar a las oscilaciones de mayor frecuencia, las líneas verticales que se observan en las gráficas. Por otra parte, las tres ondas mostradas en las figuras, se deben a la órbita del satélite. Esta es una órbita circular, y al no existir divergencia entre los valores obtenidos, se producen oscilaciones que forman esas ondas mostradas [19].

En cuanto a las dos primeras, las referentes a la posición, se puede ver como los errores absolutos no superan los 40 km, mientras que los errores relativos máximos son menores al 0.6%. Por otra parte en las gráficas de la velocidad, se ha obtenido un error absoluto máximo menos a los 0.045 km/s y un error relativo máximo similar al de la posición, 0.6%.

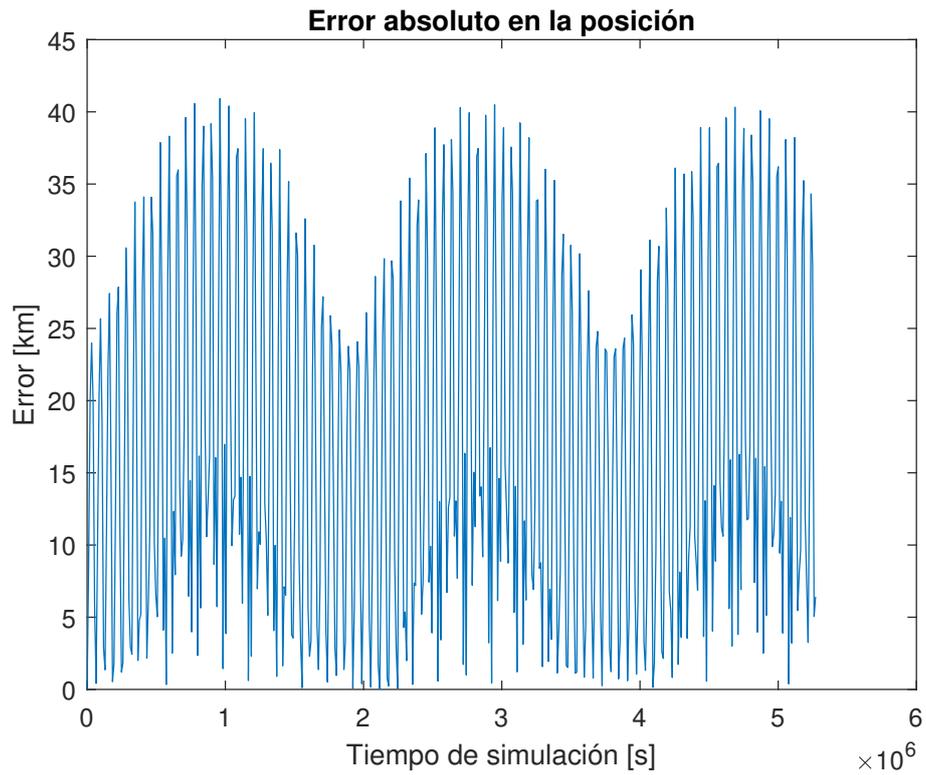


Figura 60: Error absoluto obtenido en el valor de la posición a lo largo de la trayectoria propagada del satélite Terra.

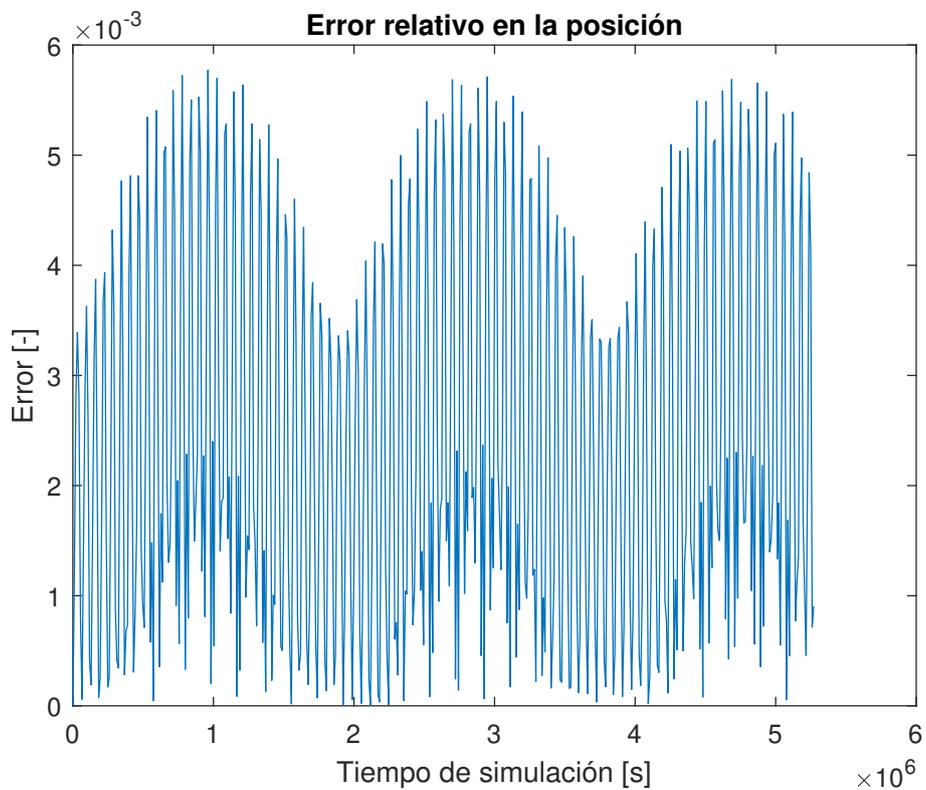


Figura 61: Error relativo obtenido en el valor de la posición a lo largo de la trayectoria propagada del satélite Terra.

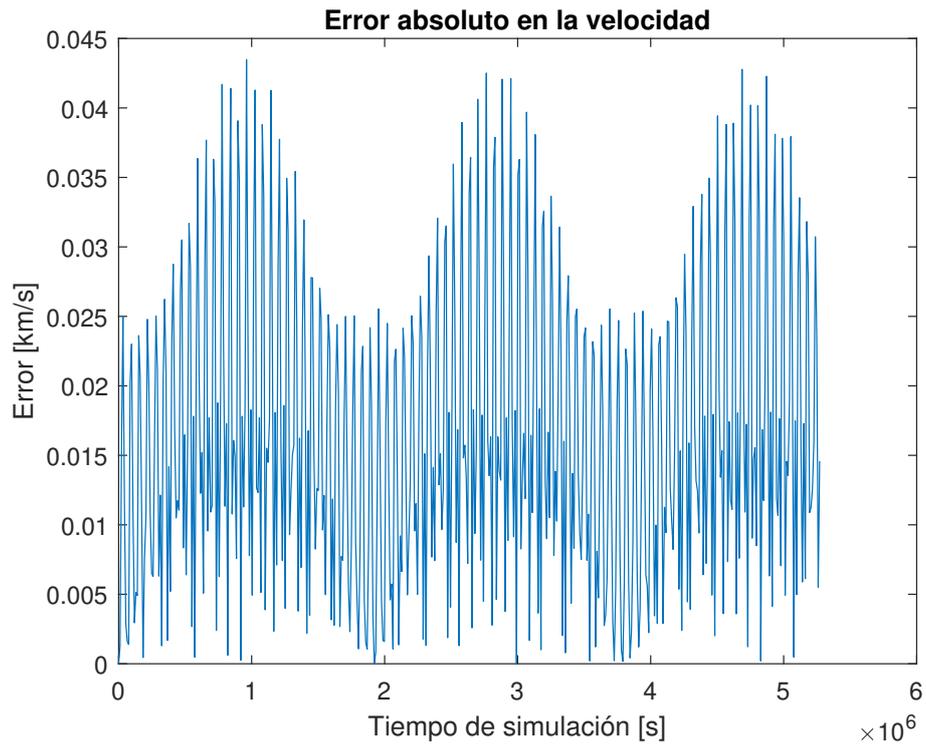


Figura 62: Error absoluto obtenido en el valor de la velocidad a lo largo de la trayectoria propagada del satélite Terra.

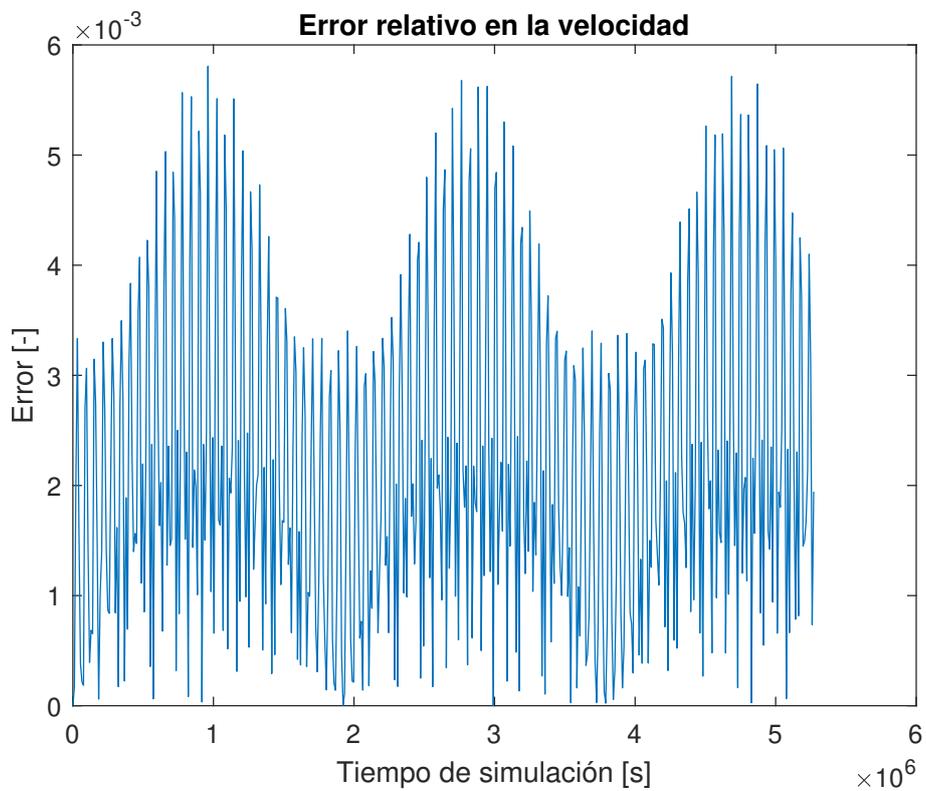


Figura 63: Error relativo obtenido en el valor de la velocidad a lo largo de la trayectoria propagada del satélite Terra.

7.2 Cambio de altura del perigeo y del apogeo de una órbita geocéntrica

A la hora de realizar maniobras impulsivas, dos de las más conocidas son el cambio de altura del perigeo y el cambio de altura del apogeo de la órbita. En este trabajo se han simulado ambas mediante la aplicación desarrollada.

Para ello se ha simulado la órbita geocéntrica formada por un satélite cuyo vector de estado inicial se muestra a continuación:

$$r_0 = [7000, 1000, 6000] \text{ km} \qquad v_0 = [2, 7, 0] \text{ km/s} \qquad (7.2)$$

Además, cabe destacar que, para ambos casos se parte del 5 de diciembre de 2020 a las doce de la noche, y se simula la trayectoria hasta que el satélite haya completado su nueva órbita tras el impulso dado. En cuanto a este impulso, también para ambos casos, consistirá en un aumento de 1 km/s en la coordenada x de la velocidad.

7.2.1 Cambio de altura del perigeo

En primer lugar, se presentará el cambio de altura en el perigeo en la órbita geocéntrica generada por el satélite. Para implementar esta maniobra en la aplicación, ha sido necesario un estudio previo, ya que se debe conocer el momento en el que se tiene que realizar el impulso, es decir, el momento en el que el satélite se encuentra en el apogeo de la órbita, ya que será el único punto común de las dos órbitas, la de antes del impulso y la de después.

Para ello se ha seguido el código 67 mostrado en el anexo B. En él se muestra como, en primer lugar, se realiza una propagación de la órbita durante cinco días, que permitirá generar varias órbitas. Así, posteriormente, se podrá obtener la anomalía verdadera a lo largo de la trayectoria a partir de los vectores de estado calculados. Con ellos, se buscan los instantes en los que el satélite se encuentra en el apogeo, los instantes en los que la anomalía verdadera es igual a 180° . El segundo de estos instantes será en el que se produzca el impulso para aumentar la altura del perigeo. De esta forma se podrá mostrar completamente la órbita generada por el satélite previa la impulso. Posteriormente, se tomarán los valores de la velocidad en ese punto, y se modificará la coordenada x de la misma, añadiéndole 1 km/s.

Con estos valores obtenidos, se pasa a la aplicación, en la cual se introducen todos los datos y se crea un impulso, cuyos valores se guardan en un archivo de texto generado por la misma aplicación. En la figura 64 se muestran algunos de los datos introducidos, y el vector de estado final, obtenido tras la propagación.

Tierra ▼

Fechas Trayecto

Fecha Inicial			Fecha Final			Vector de estado inicial			
Año	Mes	Día	Año	Mes	Día	Posición		Velocidad	
2020	12	5	2020	12	5	x	7000	Vx	2
						y	1000	Vy	7
Hora	Minuto	Segundo	Hora	Minuto	Segundo	z	6000	Vz	0
0	0	0	18	0	0				

¿Maniobra Impulsiva?

Si No

Número de Impulsos Totales:

Fecha del Impulso número

Año	Mes	Día
0	0	0
Hora	Minuto	Segundo
0	0	0

Valor del Impulso número

Vx	Vy	Vz
0	0	0

¿Maniobra Impulsiva?

Two Body
 Zonal Harmonics
 Atmósfera
 SRP Planetas

Vector de estado final

Posición		Velocidad	
x	-7567	Vx	-3.323
y	1.314e+04	Vy	-2.54
z	-8354	Vz	-1.581

Figura 64: Datos en la aplicación para realizar el cambio de altura del perigeo.

Como se observa en la figura anterior la propagación se ha realizado empleando el problema de dos cuerpos, para así asegurar que se puede apreciar correctamente la maniobra impulsiva realizada, en las imágenes obtenidas (figura 65 y 66).

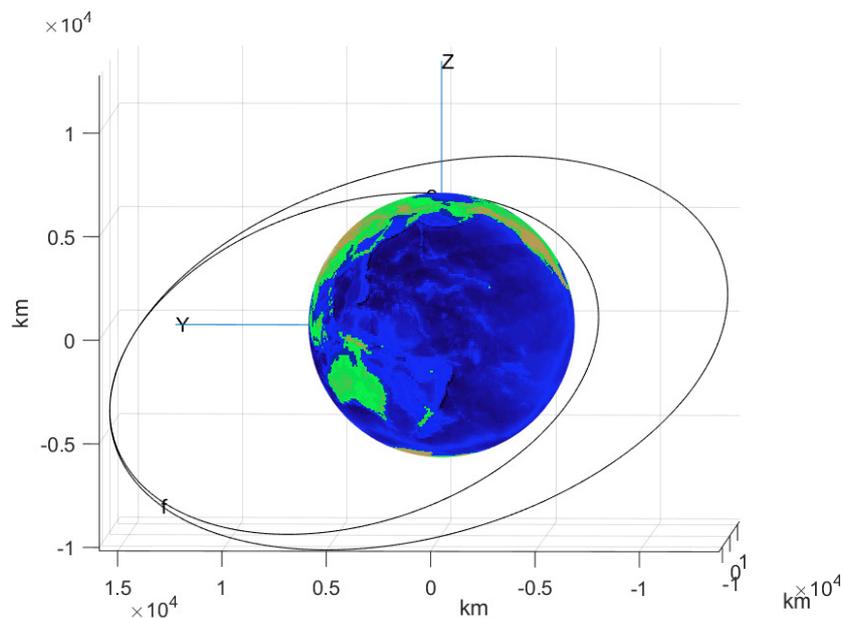


Figura 65: Cambio de altura en el perigeo.

Aunque en la primera figura ya se muestra cómo se ha aumentado la altura del perigeo correctamente, manteniendo el apogeo en el mismo punto, es la figura 65 la que permite ver claramente cómo ha sido el cambio de órbita, mostrando toda la trayectoria del satélite. En ella se observa como el satélite parte de la órbita más pequeña (punto o), y tras llegar al apogeo por segunda vez, se produce el impulso que permite que se genere la órbita de mayor tamaño. Tras recorrerla completamente, la propagación se para en el punto f.

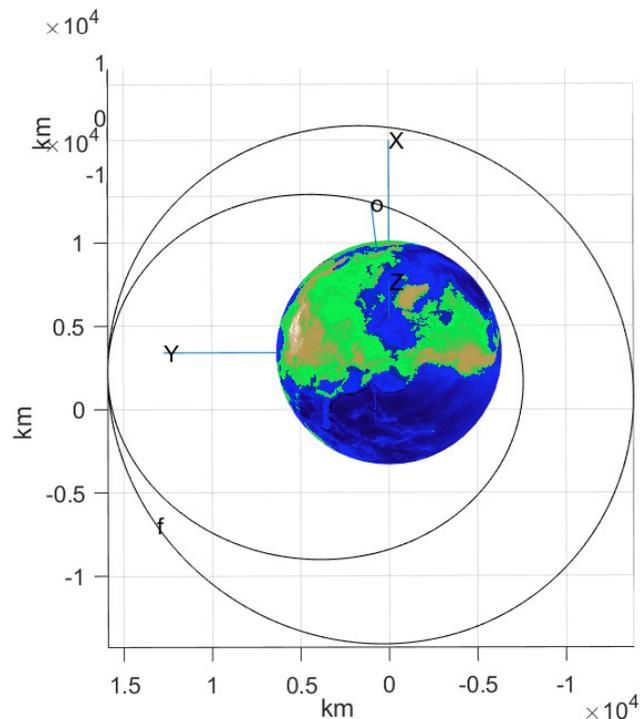


Figura 66: Cambio de altura en el perigeo (planta).

7.2.2 Cambio de altura del apogeo

En cuanto al cambio de altura del apogeo de la órbita, el proceso de cálculo es prácticamente el mismo que el anterior. En primer lugar se vuelve a propagar la órbita durante cinco días, obteniéndose los vectores de estado del satélite a lo largo de la misma. Con estos vectores, se obtendrá la anomalía verdadera en cada uno de los instantes estudiados. Y posteriormente, se buscarán los instantes en los que el satélite se encuentra en el perigeo, ya que, como se ha comentado anteriormente, el punto donde se produce el impulso será el único punto común entre las dos órbitas y, para aumentar únicamente la altura del apogeo y asegurar que la del perigeo se mantenga, se deberá de producir el impulso en el perigeo de la órbita.

Este punto se caracteriza porque en él la anomalía verdadera es nula, por lo que se deberán de buscar los valores de la anomalía iguales a 0 o a 360, o cercanos a ellos. Para ello, se han buscado sus los valores máximos y mínimos, y posteriormente los cercanos. Y se ha obtenido el primer instante de tiempo en el que la anomalía verdadera se acerca a ser nula o 360. Con ello, se ha obtenido la fecha en la que se producirá el impacto, que se empleará en la aplicación.

En la figura 67 se presentan algunos de los datos introducidos en la aplicación, y el valor del vector de estado final en la propagación de la órbita.

Tierra ▼

Fechas Trayecto

Fecha Inicial			Fecha Final			Vector de estado inicial		
Año	Mes	Día	Año	Mes	Día	Posición		Velocidad
<input type="text" value="2020"/>	<input type="text" value="12"/>	<input type="text" value="5"/>	<input type="text" value="2020"/>	<input type="text" value="12"/>	<input type="text" value="5"/>	x	<input type="text" value="7000"/>	Vx <input type="text" value="2"/>
<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="15"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	y	<input type="text" value="1000"/>	Vy <input type="text" value="7"/>
						z	<input type="text" value="6000"/>	Vz <input type="text" value="0"/>

¿Maniobra Impulsiva?

Número de Impulsos Totales ▼

Fecha del Impulso número

Año	Mes	Día	Hora	Minuto	Segundo
<input type="text" value="0"/>					

Valor del Impulso número

Vx	Vy	Vz
<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>

Vector de estado final

Posición		Velocidad	
x	<input type="text" value="-3159"/>	Vx	<input type="text" value="-1.696"/>
y	<input type="text" value="3.083e+04"/>	Vy	<input type="text" value="-0.3569"/>
z	<input type="text" value="-1.136e+04"/>	Vz	<input type="text" value="-1.231"/>

Figura 67: Datos en la aplicación para realizar el cambio de altura del apogeo.

Tras la propagación se obtiene la nueva órbita con la altura del apogeo aumentada (figuras 68 y 69). Al igual que en el anterior caso, se muestran dos figuras con distintas perspectivas. Una más global, con perspectiva, y otra en planta, en la que se pretende mostrar claramente la trayectoria generada por el satélite.

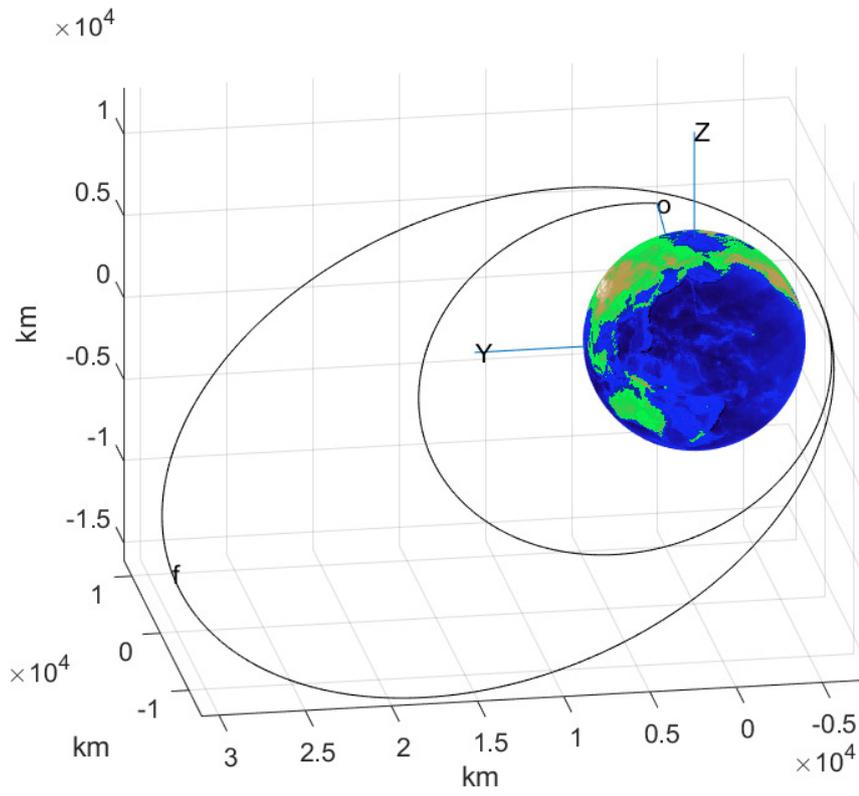


Figura 68: Cambio de altura en el apogeo.

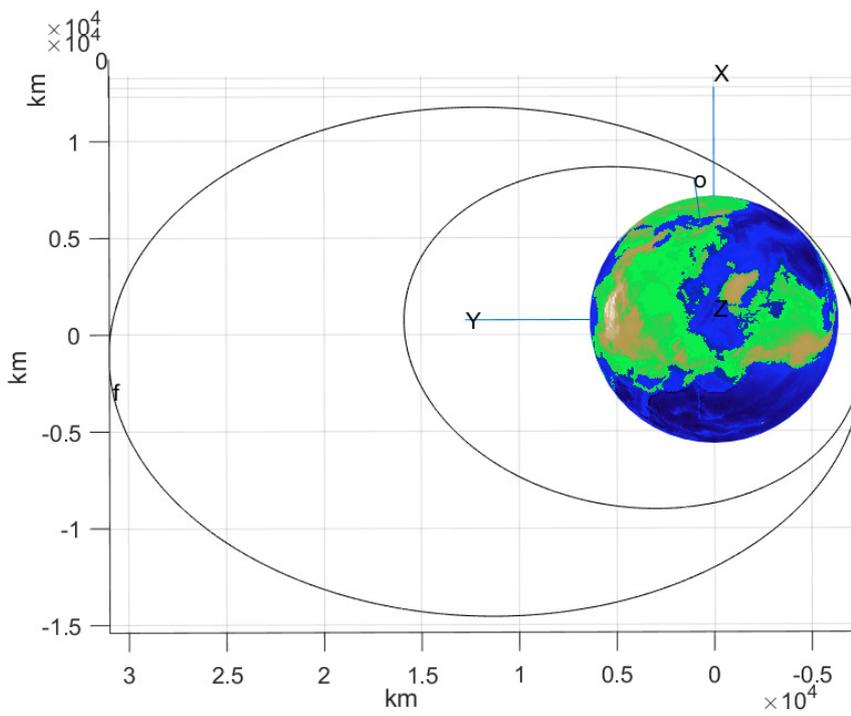


Figura 69: Cambio de altura en el apogeo (planta).

7.3 Trayectoria de la Mars2020

Por último se ha estudiado la trayectoria de la Mars 2020. Esta orbitó en una órbita heliocéntrica, con el fin de realizar un viaje de la Tierra a Marte, para llevar a dicho planeta el rover *Perseverance* y el *Mars Helicopter* utilizados para el estudio del planeta rojo. En la figura 70 se muestra una imagen de la nave estudiada.



Figura 70: Mars2020 en la prueba de preparación para el vacío térmico [20]

Esta nave despegó el 30 de julio de 2020 y llegó a Marte el 18 de febrero de 2021, realizando varias maniobras impulsivas a lo largo de su trayectoria, tal y como se muestra en la imagen 71.

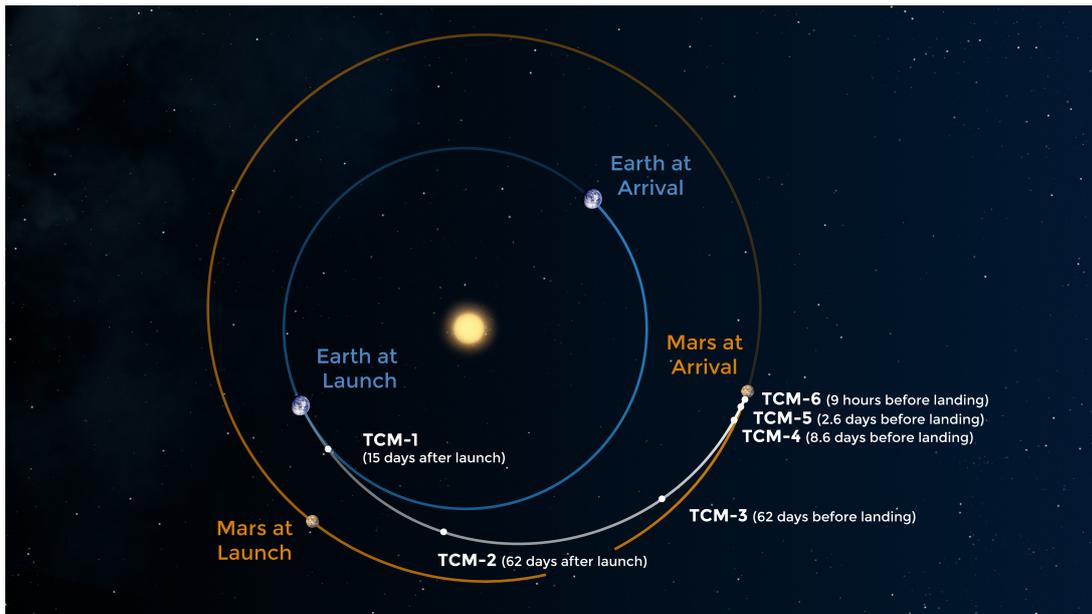


Figura 71: Trayectoria de la Mars2020 lanzada por la NASA [20].

Para la simulación de dicho viaje se ha partido del momento en el que la nave deja la Tierra por completo, y se ha terminado justo antes de comenzar la entrada a Marte. Aunque en la figura 71 se representa claramente en qué momentos se producen los impulsos, es necesario conocer las fechas exactas de los mismos para introducir correctamente los datos en la aplicación. Para ello se han obtenido dichas fechas de los datos proporcionados por el *Horizons*. En esta herramienta se pueden pedir los datos de la nave que quieras propagar, y para la Mars2020 proporciona datos acerca de su trayectoria, mostrados en la figura 72.

```
SPACECRAFT TRAJECTORY:
Post-launch fit to tracking data. Trajectory starts about an hour after
launch, after probe separation.

Trajectory files (from JPL Navigation)          Start          End
-----
m2020_traj_ops_od002v1.V0.1                    2020-Jul-30 12:53 2020-Jul-31 19:01
m2020_traj_ops_od018v1.V0.1                    2020-Jul-31 19:01 2020-Aug-15 17:00
m2020_traj_ops_od038v1.V0.1                    2020-Aug-15 17:00 2020-Sep-30 22:01
m2020_traj_ops_od071v1.V0.1                    2020-Sep-30 22:01 2020-Dec-18 21:01
m2020_traj_ops_od089v1.V0.1                    2020-Dec-18 21:01 2021-Feb-17 20:37
ed1-landed.v0.1                                2021-Feb-17 20:37 2026-Feb-19 00:00
*****
```

Figura 72: Datos de la trayectoria de la Mars2020 proporcionados por Horizons. [7]

Como se ha comentado, solo se estudiará dicha trayectoria hasta el momento en el que comienza la entrada en Marte, por lo que el último intervalo no se tendrá en cuenta. Por otra parte, con las fechas de los impulsos ya conocidas, se han descargado todos los datos de la trayectoria de la página de *Horizons*, para así conocer los vectores velocidad tras el impulso.

A continuación se pasa a preparar los datos para introducirlos en la aplicación, tras estudiar los archivos de texto descargados del *Horizons*, se toman los siguientes datos para definir la trayectoria en la aplicación:

FECHAS GLOBALES

Start time : A.D. 2020-Jul-31 12:53:00.0000 TDB
 Stop time : A.D. 2021-Feb-17 20:37:00.0000 TDB

VECTOR DE ESTADO INICIAL

Vector posición: 9.486989630665962E+07, -1.187498492820257E+08, 1.299270440212935E+05
 Vector velocidad: 2.649107572954160E+01, 1.961844272653306E+01, 1.256065185884915E+00

IMPULSO 1

Start time : A.D. 2020-Jul-31 19:01:00.0000 TDB
 Vector velocidad: 2.636913856915408E+01, 1.970550160965373E+01, 1.241246683048804E+00

IMPULSO 2

Start time : A.D. 2020-Aug-15 17:00:00.0000 TDB
 Vector velocidad: 2.089859189116402E+01, 2.463665854434477E+01, 1.136800367490464E+00

IMPULSO 3

Start time : A.D. 2020-Sep-30 22:01:00.0000 TDB
 Vector velocidad: 1.841015606057742E+00, 2.932534087885502E+01, 6.878421034553988E-01

IMPULSO 4

Start time : A.D. 2020-Dec-18 21:01:00.0000 TDB
 Vector velocidad: -1.773339156245813E+01, 1.607767864552212E+01, -1.611724390112643E-01

Figura 73: Datos de la trayectoria de la Mars2020 que se incluirán en la aplicación.[Horizons]

Con estos datos definidos el menú de la aplicación quedará tal y como se muestra en la figura 74, con 4 archivos con los datos de los impulsos creados.

Fechas Trayecto

Sol

Fecha Inicial

Año	Mes	Día
2020	7	31
Hora	Minuto	Segundo
12	53	0

Fecha Final

Año	Mes	Día
2021	2	17
Hora	Minuto	Segundo
20	37	0

Vector de estado inicial

Posición		Velocidad	
x	9.487e+07	Vx	26.49
y	-1.187e+08	Vy	19.62
z	1.299e+05	Vz	1.256

¿Maniobra Impulsiva?

Número de Impulsos Totales

Fecha del Impulso número

Año	Mes	Día
0	0	0
Hora	Minuto	Segundo
0	0	0

Valor del Impulso número

Vx	Vy	Vz
0	0	0

Opciones de Integración

Two Body
 Zonal Harmonics
 Atmósfera
 SRP Planetas

Vector de estado final

Posición		Velocidad	
x	-1.041e+06	Vx	-21.18
y	2.345e+08	Vy	2.809
z	5.039e+06	Vz	-0.5585

Planetas

<input type="button" value="Todos"/>
<input checked="" type="button" value="Luna"/>
<input type="button" value="Venus"/>
<input checked="" type="button" value="Tierra"/>
<input checked="" type="button" value="Marte"/>
<input checked="" type="button" value="Júpiter"/>
<input type="button" value="Saturno"/>
<input type="button" value="Urano"/>
<input type="button" value="Neptuno"/>

Figura 74: Datos en la aplicación para simular la órbita de la Mars2020.

En cuanto a las perturbaciones tomadas, por falta de datos no se ha incluido la perturbación debida a la presión de radiación solar, y en cuanto a la perturbación causada por la gravedad de otros cuerpo, únicamente se han tenido en cuenta aquellos que más influencia he tenido en la trayectoria generada por la nave (Tierra, Luna, Marte y Júpiter).

Tras introducir los datos en la aplicación, se pasa al estudio de errores mediante el código 69 presentado en el anexo B. Con él se han obtenido las gráficas de errores mostradas a continuación (figuras 75 y 76 para la posición, y 77 y 78 para la velocidad).

Comenzando por las dos primera gráficas obtenidas, referentes a la posición, en ellas se observa como se forman picos en los instante en los que se producen los impulsos, que son mayores a medida que se acerca el final de la propagación. Esto indica que el hecho de tener en cuenta los impulsos a lo largo de la maniobra reduce notablemente los errores en la propagación. También se puede ver como el error aumenta a medida que se va realizando la propagación, lo que es normal, debido a que no se han tenido en cuenta todas las perturbaciones posibles. Y para un viaje interplanetario la perturbación causada por la presión de radiación solar, puede tener grandes efectos.

Por otra parte, se observa como aunque los errores absolutos parecen elevados, estando los máximos cerca de $2 \cdot 10^5$ km, en el error relativo se demuestra como no lo son tanto, ya que este error no lleva al 0.09 %.

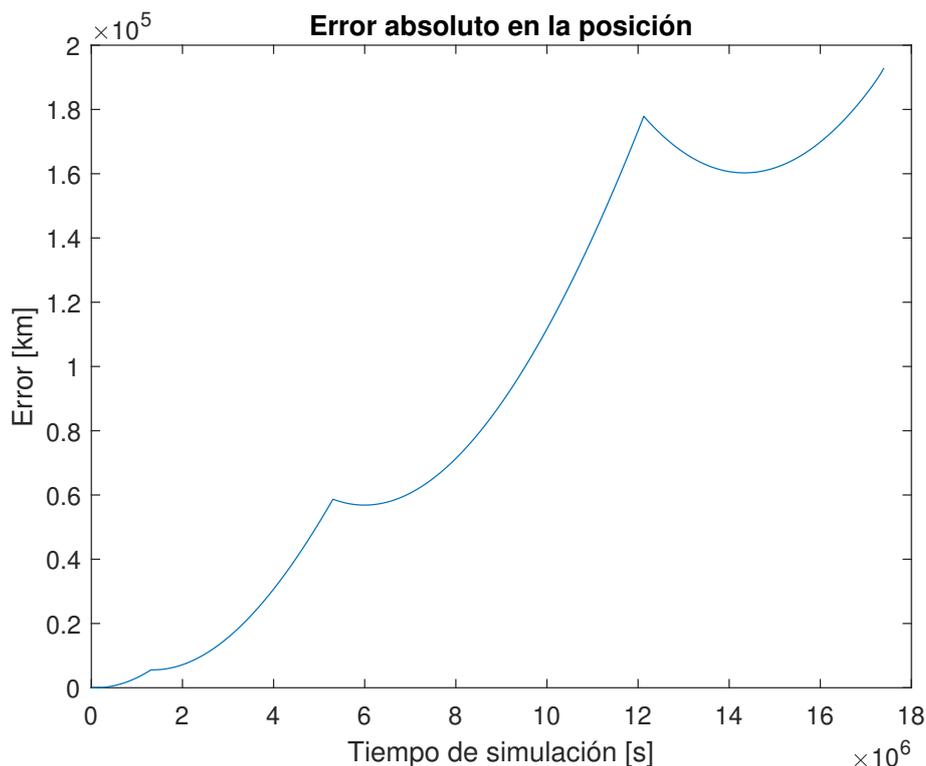


Figura 75: Error absoluto obtenido en el valor de la posición a lo largo de la trayectoria propagada de la Mars2020.

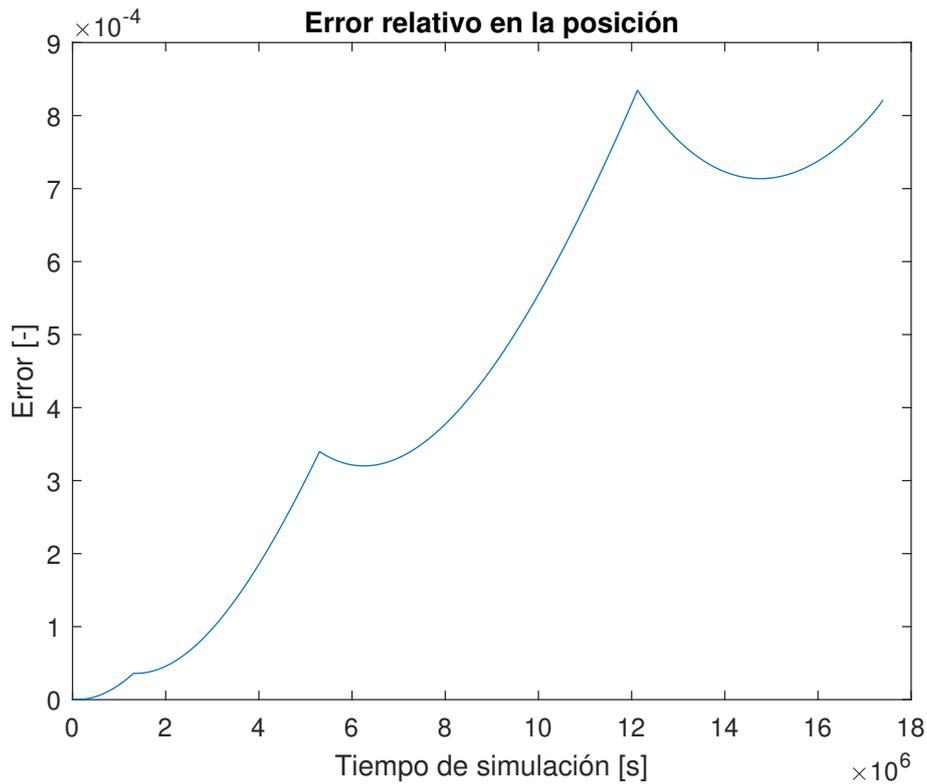


Figura 76: Error relativo obtenido en el valor de la posición a lo largo de la trayectoria propagada de la Mars2020.

En cuanto a los errores producidos en la propagación de la velocidad. En las siguientes figuras se muestra la evolución del error. En ellas se puede observar como se producen varios picos debido a los impulsos realizados a lo largo de la trayectoria. Se sabe que se producen en esos instantes debido a que coinciden con los picos obtenidos en los errores de la posición y además, el error disminuye a cero, debido a que el vector velocidad en ese instante es el mismo que proporciona *Horizons*.

En cuanto a los valores de los errores obtenidos, se ve como, al igual que con la posición, el error aumenta a medida que la nave se acerca a Marte. Pero, aun así, el error absoluto máximo en la velocidad es cercano a $7 \cdot 10^{-3}$ km/s, y el error relativo máximo a 0.03 %.

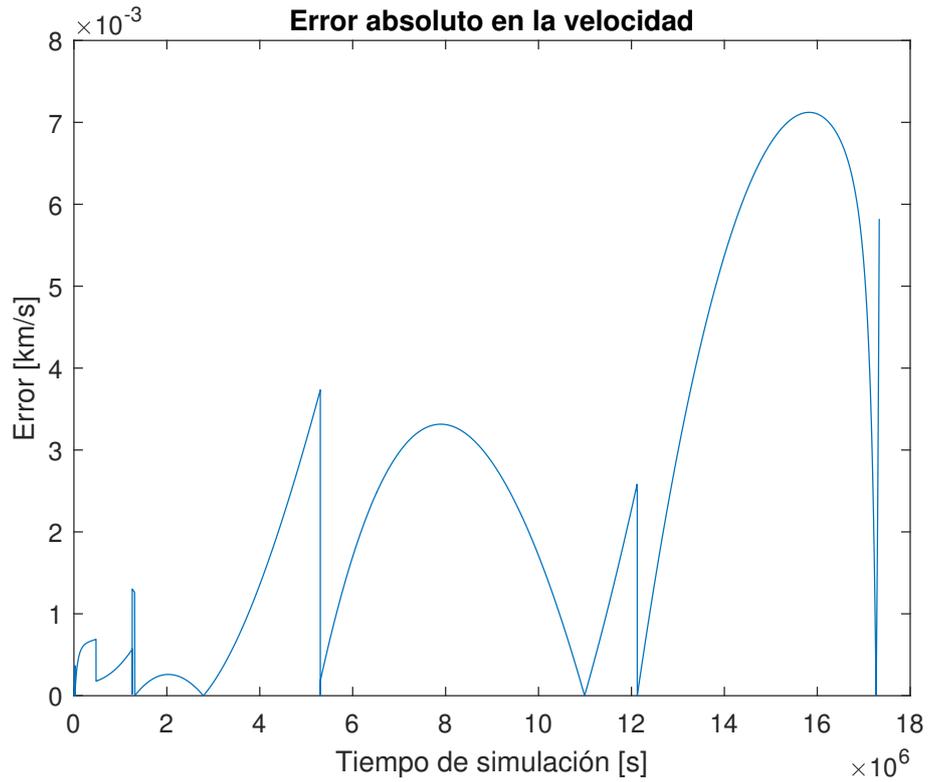


Figura 77: Error absoluto obtenido en el valor de la velocidad a lo largo de la trayectoria propagada de la Mars2020.

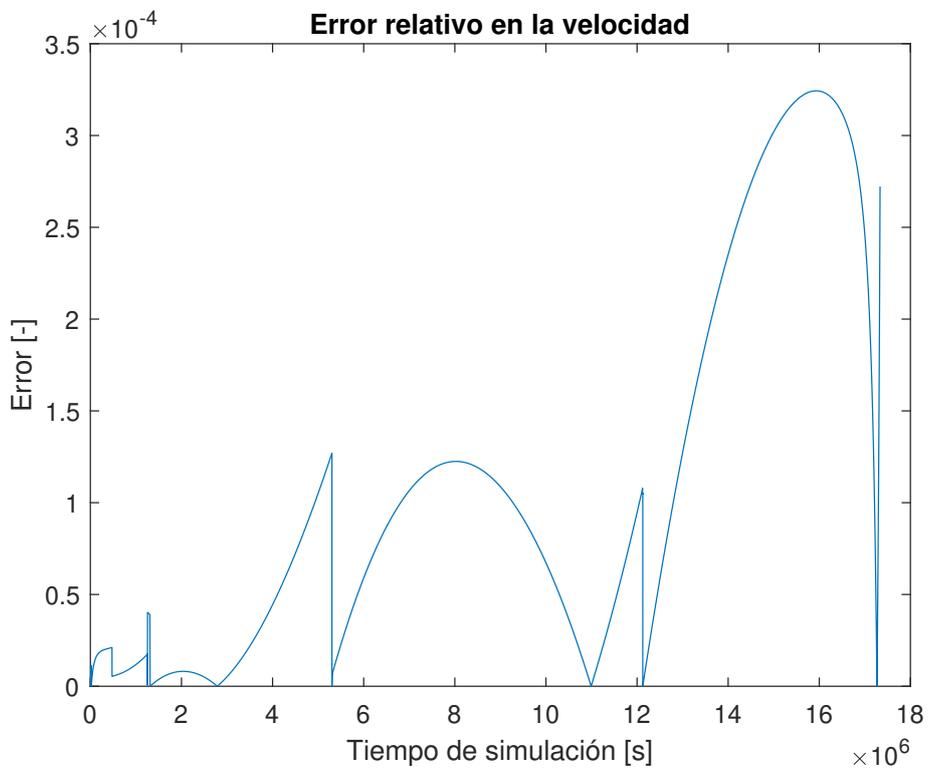


Figura 78: Error relativo obtenido en el valor de la velocidad a lo largo de la trayectoria propagada de la Mars2020.

8 CONCLUSIONES Y FUTURAS LÍNEAS DE ESTUDIO

8.1 Conclusiones

En los últimos años ha aumentado el interés por las misiones espaciales necesarias para el desarrollo tecnológico en la Tierra, el estudio de nuestro planeta, o el estudio de otros planetas y otros cuerpos en el espacio. Dentro del diseño de estas misiones uno de los puntos más importantes es el estudio de la trayectoria a recorrer por la nave o satélite. Este estudio previo se realiza mediante propagadores.

En este trabajo se ha generado una aplicación capaz de realizar la propagación de cualquier trayectoria en dos sistemas de referencia distintos, geocéntrico y heliocéntrico. Además, permite realizar lo anterior mediante dos métodos de propagación diferentes, el problema de dos cuerpos y el método de Cowell. Y dentro de este último, permite emplear cualquiera de las perturbaciones más influyentes en la trayectoria del cuerpo a propagar.

Dentro de estas perturbaciones, en el caso de una órbita heliocéntrica únicamente se permite tener en cuenta dos de ellas, la perturbación causada por la presión de radiación solar y la perturbación causada por la gravedad de otros cuerpos, más concretamente los planetas del sistema solar, el Sol y la Luna. Esto se debe a la poca influencia, prácticamente nula, que tendrán las otras dos perturbaciones, en órbitas no cercanas a la Tierra. Estas perturbaciones, la causada por la no uniformidad del campo gravitatorio terrestre y la perturbación atmosférica, por el contrario, sí que se podrán tener en cuenta al propagar en el sistema geocéntrico. Por lo que en este sistema se podrán tener en cuenta cualquiera de las cuatro perturbaciones implementadas.

Además, la aplicación programada permite introducir impulsos en la trayectoria propagada. Esto permitirá simular trayectorias más realistas, ya que es una práctica común en las misiones espaciales, que permite cambios o ajustes de órbita para distintos fines. Al introducir el impulso solo se deberá conocer la fecha en la que se produce y el vector velocidad tras él.

Tras completarse la aplicación, previo a la creación y comparación de trayectorias, se validan los integradores mediante la comparación de los resultados obtenidos con los resultados proporcionados con los generados por el propagador GMAT. Este propagador permite la simulación de órbitas teniendo en cuenta las perturbaciones que se han introducido en la aplicación, además de poder emplear cualquiera de los dos sistemas de referencia simulados. Aunque para realizar estas validaciones se ha empleado únicamente el sistema geocéntrico, ya que dentro de los integradores a validar no existen diferencias entre ambos sistemas de referencia.

Para estas comparaciones, se simula la órbita de un satélite con unas coordenadas iniciales aleatorias, durante una duración de 10 días, y se estudian los distintos métodos de propagación y perturbaciones por separado. Se comenzará con el problema de dos cuerpos, para continuar con el de Cowell. Y dentro de este último cada perturbación se estudiará tanto individualmente como en conjunto con el resto. Por último, la validez de los resultados dependerá de los errores obtenidos en el cálculo de la posición y la velocidad a lo largo de la órbita.

Comenzando con el problema de dos cuerpos, se observa como el error máximo aumenta de manera lineal con el tiempo de propagación tanto para la posición como para la velocidad. Además, en cuanto a los valores, el máximo del error absoluto en la posición es menor de 3 km, el del error relativo en la posición es menor al 0.025 %. Y para la velocidad el máximo error

absoluto ronda los $1.2 \cdot 10^{-3}$ km/s, y el relativo el 0.02 %.

Pasando al método de Cowell, se comienza por estudiar el integrador que propaga teniendo en cuenta la perturbación causada por los planetas, la Luna y el Sol. Para este ejemplo únicamente se han tenido en cuenta las perturbaciones causadas por la Luna, Marte y Júpiter, ya que serán los que más afecten a la órbita propagada. Los valores de los errores para este caso, son prácticamente los mismos que los obtenidos con el problema de dos cuerpos. Esto se debe a que el vector estado a lo largo de la órbita es prácticamente el mismo que en el caso anterior, debido a la gran distancia que existe entre el satélite (en órbita LEO) y los cuerpos que perturban su trayectoria.

A continuación, se valida el integrador que contiene la perturbación causada por la presión de radiación solar. Para tener en cuenta esta perturbación ha sido necesario introducir algunas características del satélite: el coeficiente de reflectividad, que tendrá un valor de 1.8, y la relación área/masa de valor 1 m^2 . Con estos datos, al comparar la posición se obtienen un error absoluto máximo que ronda los 4.5 km, y un error relativo máximo menor del 0.045 %. En la velocidad los errores máximos son de 0.0025 km/s y del 0.04 %, respectivamente.

En cuanto al integrador que incluye la perturbación causada por la no uniformidad del campo gravitatorio terrestre, para este caso, solamente se ha tenido en cuenta el armónico zonal J_2 , esto se debe a que así será más fácil que ambos propagadores se parezcan, ya que en GMAT se utiliza otro método para obtener dicha perturbación (JMG2). En la comparativa de resultados se obtiene un error absoluto máximo en la posición de 60 km, y un error relativo máximo que ronda el 0.5 %. En cuanto a la velocidad el error absoluto máximo es menor de 0.03 km/s y el relativo algo mayor del 0.45 %. Las diferencias más grandes se deben a lo anteriormente comentado, los métodos empleados no son exactamente los mismos.

Por otra parte, en la comparativa teniendo en cuenta la perturbación atmosférica se ha tenido que incluir también la perturbación anterior debido a las opciones del propagador de GMAT. Además, se han especificado más características del satélite: coeficiente de resistencia de 2.2, y la relación área/masa de 15 m^2 . Por otra parte, cabe destacar que para esta propagación GMAT emplea otro modelo atmosférico, el modelo JAcchia-Roberts. En cuanto a los valores de los errores máximos obtenidos, en la posición el error absoluto es cercano a 30 km mientras que el relativo ronda el 0.25 %, y en la velocidad son cercanos a 0.014 km/s y 0.22 %, respectivamente.

Finalmente, se comparan los integradores teniendo en cuenta todas las perturbaciones a la vez, obteniéndose un error absoluto máximo para la posición de 150 km, y para la velocidad de 0.07 km/s. En cuanto al error relativo máximo es menor a 1.5 % para la posición y a 1.2 %.

Tras realizar todas las comparativas, se observa como en ningún caso se obtiene errores excesivamente grandes, y que las diferencias obtenidas se basan sobretodo en los distintos métodos empleados en la propagación. Por lo que quedarán validados los integradores programados.

Pasando entonces a los resultados, en esta parte se han estudiado órbitas y trayectorias reales, la del satélite geocéntrico Terra y la de la Mars2020. Además se han generado dos de las maniobras impulsivas más empleadas en misiones espaciales.

Comenzando con la del satélite, se ha propagado su órbita durante dos meses teniendo en cuenta tres perturbaciones distintas, la generada por la uniformidad del campo terrestre, la producida por la atmósfera terrestre, y la producida por la gravedad de los planetas, el Sol y la

Luna. Para la primera de ellas se han tenido en cuenta todos los armónicos zonales incluidos en la aplicación, al igual que para la perturbación causada por la gravedad de otros cuerpos, en la cual se han incluido todos los posibles. En cuanto a la perturbación atmosférica, basando en datos proporcionados por la NASA, se ha tomado una relación área/masa de 0.0046 m^2 y un coeficiente de resistencia atmosférica de 2.2.

Con todo lo anterior, se ha propagado la órbita obteniendo los siguiente valores en los errores máximos. El error absoluto de la posición es menor de 40 km y el error relativo menor de 0.6%, mientras que el error absoluto de la velocidad es algo menor de 0.045 km/s, y el relativo algo menor de 0.6%.

A continuación, se han generado las dos maniobras mencionadas anteriormente, que serán un cambio de altura en el perigeo y un cambio de altura en el apogeo de al órbita. Ambas se han realizado partiendo del mismo vector de estado, en la misma fecha.

Previo a la realización de ambas maniobras, se tuvo que realizar un estudio para ver en que instante se debía de producir el impulso. Conocidos los puntos en los que se debían de producir, apogeo y perigeo respectivamente, se obtuvieron las fechas en las que se producirían. Y con ellas el vector velocidad después del impulso ya que se aumentaría el 1 km/s la coordenada x de la velocidad en ese instante de tiempo para así conseguir el aumento de altura en el punto deseado. Con estos datos se obtuvieron las maniobras perfectamente realizadas tal y como se muestra en las figuras 66 y 69.

Por último, se simula la trayectoria realizada por la Mars2020 desde que deja la Tierra hasta que comienza su llegada a Marte. Durante esta trayectoria se producen varios impulsos por lo que además de emplear el sistema geocéntrico se utilizará la opción de maniobras impulsivas de la aplicación. Con los datos de los impulsos obtenidos de la pagina de *Horizons*, se propaga la trayectoria deseada.

Tras ello, se obtiene los errores a lo largo de la misma, al comparar con los valores del *Horizons*. En cuanto a la posición se ha obtenido un error absoluto máximo cercano a $2 \cdot 10^5$ km, y un error relativo máximo cercano a 0.09%. Para la velocidad el error absoluto máximo se acerca a $7 \cdot 10^3$ km/s y el relativo a 0.03%.

8.2 Futuras líneas de estudio

Aunque la aplicación desarrollada permite simular las trayectorias más comunes empleadas en las misiones espaciales, existen algunas mejoras que se podrían implementar.

En primer lugar, la aplicación utiliza vectores de estado en coordenadas cartesianas tanto para introducir el vector inicial como a la hora de devolver dicho vector. A la hora de estudiar la órbita en la que se encuentra el cuerpo es más sencillo trabajar con elementos orbitales, o incluso equinocciales.

Además, dentro de esto se puede destacar que para la introducción de los datos referentes a la velocidad tras el impulso, también podría emplearse otro tipo de datos, como por ejemplo el incremento en cada una de las componentes, o la velocidad en coordenadas perifocales, ya sea con incremento o directamente, ya que también se emplea para la definición de maniobras orbitales.

Por otra parte, también pueden realizarse mejoras en la parte de cálculo. Como se ha visto en la validación los propagadores más desarrollados emplean para algunas de las perturbaciones métodos distintos a los desarrollados en este trabajo. Por lo que sería de interés mejorar los métodos empleados, para así reducir aún más los errores cometidos y poder propagar con mayor exactitud la órbita obtenida.

Otro aspecto a tener en cuenta, es el método de propagación a emplear. En este trabajo se han empleado únicamente el problema de dos cuerpos y el método de Cowell, pero hoy en día también se emplea el método de variación de parámetros, que permite integrar las ecuaciones directamente a partir de los elementos orbitales o equinociales, y tener en cuenta todas las perturbaciones estudiadas en este trabajo. Por lo que el hecho de añadir esta opción en la integración sería una mejora interesante en el desarrollo de la aplicación.

Además, a la hora de realizar la integración, existen métodos de mayor orden que el ode45, que pueden proporcionar menores errores en la integración realizada, y que pueden programarse en MATLAB.

Por último, aunque tenga menor importancia, a la hora de conocer las coordenadas de los planetas, el Sol y la Luna, necesarias para tener en cuenta algunas de las perturbaciones estudiadas, en este trabajo se han empleado las coordenadas descargadas de la página de *Horizons*, pero podrían obtenerse de manera automática, descargándolas automáticamente la aplicación, o de manera más sencilla, calculándolas mediante una función. Esta última posiblemente aumentaría el tiempo de propagación, pero siempre que no sea demasiado importante el tiempo y que el método esté bien implementado, puede ser una solución efectiva.

Con todo lo anterior se puede asegurar una mejora en la aplicación que permita obtener trayectorias a partir de datos distintos y con mayor exactitud.

9 REFERENCIAS

- [1] F. WILKINSON, *The History of Space Exploration*, NATIONAL GEOGRAPHIC, 2020. [Online]. Available: <https://www.nationalgeographic.org/article/history-space-exploration/12th-grade/> [Accessed: 17-Aug-2021].
- [2] A BRIEF HISTORY OF SPACE EXPLORATION, 2018. [Online]. Available: <https://aerospace.org/article/brief-history-space-exploration> [Accessed: 17-Aug-2021].
- [3] A. WATT, *The 15 Most Important Space Missions in History*, 2021. [Online]. Available: <https://www.popularmechanics.com/space/g25941053/famous-space-missions-through-history/?slide=13> [Accessed: 17-Aug-2021].
- [4] W. J. LARSON AND J. R. WERTZ, *Space Mission Analysis and Design*, 3rd ed. 2005
- [5] *Systems Tool Kit (STK)*, 2021. [Online]. Available: <https://www.agi.com/products/stk> [Accessed: 17-Aug-2021].
- [6] *Design And Integration Tools General Mission Analysis Tool (GMAT)*, 2021. [Online]. Available: <https://software.nasa.gov/software/GSC-17177-1> [Accessed: 17-Aug-2021].
- [7] R. S. PARK, A. B. CHAMBERLIN, "HORIZONS Web-Interface", Jet Propulsion Laboratory, California Institute of Technology, [Online]. Available: <https://ssd.jpl.nasa.gov/horizons.cgi> [Accessed: 22-Aug-2021].
- [8] D. VALLADO AND W. MCCLAIN. *Fundamentals of Astrodynamics and Applications*. Hawthorne, Calif.: Microcosm Press, 2007
- [9] O. MONTENBRUCK AND E. GILL, *Satellite Orbits*. Berlin: Springer Berlin, 2001
- [10] K. SERGIE A., *Basic Celestial MEchanics*. 2011
- [11] M.FARAHMAND, *ORBITAL PROPAGATORS FOR HORIZON SIMULATION FRAMEWORK*, Faculty of California Polytechnic State University, San Luis Obispo. 2009
- [12] K. F. WAKEER, *Fundamentals of Astrodynamics*. 2015
- [13] R. H. LYON, *Geosynchronous Orbit Determination Using Space Surveillance Network Observations and Improved Radiative Force Modeling*. MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 2004
- [14] H. D. CURTIS, *Orbital MEchanics for Engineering Students*. Daytona Beach, Florida: Elsevier, Butterworth-Heinemann, 4th ed. 2020
- [15] D. CONTE, D. SPENCER *Targeting the Martian Moons via Direct Insertion into Mars' Orbit*, AIAA/AAS ASTRODYNAMIC SPECIALIS, 2015. [Online]. Available: https://www.researchgate.net/publication/281378287_Targeting_the_Martian_Moons_via_Direct_Insertion_into_Mars%27Orbitpf3. [Accessed: 20-Aug-2021].
- [16] *Conventional Celestial Reference System*, European Space Agency, 2012. [Online]. Available: https://gssc.esa.int/navipedia/index.php/Conventional_Celestial_Reference_System. [Accessed: 20-Aug-2021]

-
- [17] M. H. PAJARES *Introduction to GNSS Data Processing. 3.3 Orbital Elements*, UPC-IonSAT res. group, 1998. [Online]. Available: [http : //chapman.upc.es/lectures/legad/node14.html](http://chapman.upc.es/lectures/legad/node14.html). [Accessed: 20-Aug-2021].
- [18] K. THOME, T. OWEN *Terra The EOS Flagship*, National Aeronautics and Space Administration (NASA), [Online]. Available: [https : //terra.nasa.gov/](https://terra.nasa.gov/). [Accessed: 22-Aug-2021].
- [19] L. CHEN, X. BAI, Y. LIANG, K. LI *Orbital Prediction Error Propagation of Space Objects*, National University of Defense Technology, 2017. [Online]. Available: [https : //www.researchgate.net/publication/311394876_Orbital_Prediction_Error_Propagation_of_Space_Objects](https://www.researchgate.net/publication/311394876_Orbital_Prediction_Error_Propagation_of_Space_Objects). [Accessed: 22-Aug-2021].
- [20] "Mars 2020 Mission Perseverance Rover", [Online]. Available: [https : //mars.nasa.gov/mars2020/](https://mars.nasa.gov/mars2020/) [Accessed: 22-Aug-2021].

A Código de la Aplicación

Listing 1: Función en la aplicación por el *Drop Down* Foco de la órbita.

```

1
2 function FocodelarbitaDropDownValueChanged(app, event)
3     value = app.FocodelarbitaDropDown.Value;
4
5     app.SButton.Value= 0;
6     app.NoButton.Value = 0;
7     app.SButtonValueChanged;
8     app.NoButtonValueChanged;
9
10    app.OpcionesdeIntegracinLabel.Enable = 'Off';
11    app.TwoBodyCheckBox.Enable           = 'Off';
12    app.ZonalHarmonicsCheckBox.Enable    = 'Off';
13    app.AtmosferaCheckBox.Enable         = 'Off';
14    app.PlanetasCheckBox.Enable         = 'Off';
15    app.SRPCCheckBox.Enable              = 'Off';
16
17    if isequal(value, 'Foco de la orbita')
18
19        app.VectordeestadoinicialLabel.Enable = 'Off';
20        app.PosicinInicialLabel.Enable       = 'Off';
21        app.VelocidadInicialLabel.Enable     = 'Off';
22
23        app.x0EditField.Enable = 'Off';
24        app.y0EditField.Enable = 'Off';
25        app.z0EditField.Enable = 'Off';
26
27        app.xEditFieldLabel.Enable = 'Off';
28        app.yEditFieldLabel.Enable = 'Off';
29        app.zEditFieldLabel.Enable = 'Off';
30
31        app.Vx0EditField.Enable = 'Off';
32        app.Vy0EditField.Enable = 'Off';
33        app.Vz0EditField.Enable = 'Off';
34
35        app.VxEditFieldLabel.Enable = 'Off';
36        app.VyEditFieldLabel.Enable = 'Off';
37        app.VzEditFieldLabel.Enable = 'Off';
38
39        app.ManiobraImpulsivaLabel.Enable = 'Off';
40
41        app.SButton.Enable = 'Off';
42        app.NoButton.Enable = 'Off';
43
44    elseif isequal(value, 'Sol')
45
46        app.VectordeestadoinicialLabel.Enable = 'On';
47        app.PosicinInicialLabel.Enable       = 'On';
48        app.VelocidadInicialLabel.Enable     = 'On';
49
50        app.x0EditField.Enable = 'On';
51        app.y0EditField.Enable = 'On';
52        app.z0EditField.Enable = 'On';
53
54        app.xEditFieldLabel.Enable = 'On';
55        app.yEditFieldLabel.Enable = 'On';

```

```

56     app.zEditFieldLabel.Enable = 'On';
57
58     app.Vx0EditField.Enable = 'On';
59     app.Vy0EditField.Enable = 'On';
60     app.Vz0EditField.Enable = 'On';
61
62     app.VxEditFieldLabel.Enable = 'On';
63     app.VyEditFieldLabel.Enable = 'On';
64     app.VzEditFieldLabel.Enable = 'On';
65
66     app.ManiobraImpulsivaLabel.Enable = 'On';
67     app.SButton.Enable = 'On';
68     app.NoButton.Enable = 'On';
69     app.SButtonValueChanged;
70     app.NoButtonValueChanged;
71
72     elseif isequal(value, 'Tierra')
73
74         app.VectordeestadoinicialLabel.Enable = 'On';
75         app.PosicinInicialLabel.Enable = 'On';
76         app.VelocidadInicialLabel.Enable = 'On';
77
78         app.x0EditField.Enable = 'On';
79         app.y0EditField.Enable = 'On';
80         app.z0EditField.Enable = 'On';
81
82         app.xEditFieldLabel.Enable = 'On';
83         app.yEditFieldLabel.Enable = 'On';
84         app.zEditFieldLabel.Enable = 'On';
85
86         app.Vx0EditField.Enable = 'On';
87         app.Vy0EditField.Enable = 'On';
88         app.Vz0EditField.Enable = 'On';
89
90         app.VxEditFieldLabel.Enable = 'On';
91         app.VyEditFieldLabel.Enable = 'On';
92         app.VzEditFieldLabel.Enable = 'On';
93
94         app.ManiobraImpulsivaLabel.Enable = 'On';
95         app.SButton.Enable = 'On';
96         app.NoButton.Enable = 'On';
97     end
98 end

```

Listing 2: Función en la aplicación por el *State Button* Sí.

```

1
2 function SButtonValueChanged(app, event)
3
4     value = app.SButton.Value;
5     app.NoButton.Enable = 'Off';
6
7     app.NmerodeImpusosTotalesSpinner.Visible = 'On';
8     app.NmerodeImpusosTotalesSpinner.Enable = 'On';
9     app.NmerodeImpusosTotalesSpinnerLabel.Visible = 'On';
10    app.NmerodeImpusosTotalesSpinnerLabel.Enable = 'On';
11
12    if value == 0
13        app.NoButton.Enable = 'On';

```

```
14
15     % Numero de Impulsos
16     app.NmerodeImpusosTotalesSpinner.Value = 0;
17     app.NmerodeImpusosTotalesSpinner.Visible = 'Off';
18     app.NmerodeImpusosTotalesSpinner.Enable = 'Off';
19     app.NmerodeImpusosTotalesSpinnerLabel.Visible = 'Off';
20     app.NmerodeImpusosTotalesSpinnerLabel.Enable = 'Off';
21
22     % Titulo de Fecha
23     app.FechadelImpulsonmeroEditField.Visible = 'Off';
24     app.FechadelImpulsonmeroEditField.Enable = 'Off';
25     app.FechadelImpulsonmeroEditFieldLabel.Visible = 'Off';
26     app.FechadelImpulsonmeroEditFieldLabel.Enable = 'Off';
27
28     % Valores de Fecha
29     app.AoEditField.Visible = 'Off';
30     app.AoEditField.Enable = 'Off';
31     app.AoEditFieldLabel_2.Visible = 'Off';
32     app.AoEditFieldLabel_2.Enable = 'Off';
33
34     app.MesEditField.Visible = 'Off';
35     app.MesEditField.Enable = 'Off';
36     app.MesEditFieldLabel_2.Visible = 'Off';
37     app.MesEditFieldLabel_2.Enable = 'Off';
38
39     app.DaEditField.Visible = 'Off';
40     app.DaEditField.Enable = 'Off';
41     app.DaEditFieldLabel_2.Visible = 'Off';
42     app.DaEditFieldLabel_2.Enable = 'Off';
43
44     app.HoraEditField.Visible = 'Off';
45     app.HoraEditField.Enable = 'Off';
46     app.HoraEditFieldLabel_2.Visible = 'Off';
47     app.HoraEditFieldLabel_2.Enable = 'Off';
48
49     app.MinutoEditField.Visible = 'Off';
50     app.MinutoEditField.Enable = 'Off';
51     app.MinutoEditFieldLabel_2.Visible = 'Off';
52     app.MinutoEditFieldLabel_2.Enable = 'Off';
53
54     app.SegundoEditField.Visible = 'Off';
55     app.SegundoEditField.Enable = 'Off';
56     app.SegundoEditFieldLabel_2.Visible = 'Off';
57     app.SegundoEditFieldLabel_2.Enable = 'Off';
58
59     % Titulo del Impulso
60     app.ValordelImpulsonmeroEditField.Visible = 'Off';
61     app.ValordelImpulsonmeroEditField.Enable = 'Off';
62     app.ValordelImpulsonmeroEditFieldLabel.Visible = 'Off';
63     app.ValordelImpulsonmeroEditFieldLabel.Enable = 'Off';
64
65     % Valores del Impulso
66     app.VxEditField.Visible = 'Off';
67     app.VxEditField.Enable = 'Off';
68     app.VxLabel.Visible = 'Off';
69     app.VxLabel.Enable = 'Off';
70
71     app.VyEditField.Visible = 'Off';
72     app.VyEditField.Enable = 'Off';
73     app.VyLabel.Visible = 'Off';
74     app.VyLabel.Enable = 'Off';
```

```

75
76     app.VzEditField.Visible = 'Off';
77     app.VzEditField.Enable  = 'Off';
78     app.VzLabel.Visible    = 'Off';
79     app.VzLabel.Enable     = 'Off';
80
81     % Guardar
82     app.GuardarDatosButton.Visible = 'Off';
83     app.GuardarDatosButton.Enable  = 'Off';
84 end
85
86     value_N = app.NoButton.Value;
87
88     if value == 0 && value_N == 0
89
90         app.TwoBodyCheckBox.Value      = 0;
91         app.PlanetasCheckBox.Value     = 0;
92         app.SRPCheckBox.Value          = 0;
93         app.AtmsferaCheckBox.Value     = 0;
94         app.ZonalHarmonicsCheckBox.Value = 0;
95     end
96
97 end

```

Listing 3: Función en la aplicación por el contador de impulsos totales.

```

1
2 function NmerodeImpusosTotalesSpinnerValueChanged(app, event)
3
4     value = app.NmerodeImpusosTotalesSpinner.Value;
5
6     if value > 0
7         % Titulo de Fecha
8         app.FechadelImpulsonmeroEditField.Visible = 'On';
9         app.FechadelImpulsonmeroEditField.Enable  = 'On';
10        app.FechadelImpulsonmeroEditFieldLabel.Visible = 'On';
11        app.FechadelImpulsonmeroEditFieldLabel.Enable  = 'On';
12
13        % Valores de Fecha
14        app.AoEditField.Visible = 'On';
15        app.AoEditField.Enable  = 'On';
16        app.AoEditFieldLabel_2.Visible = 'On';
17        app.AoEditFieldLabel_2.Enable  = 'On';
18
19        app.MesEditField.Visible = 'On';
20        app.MesEditField.Enable  = 'On';
21        app.MesEditFieldLabel_2.Visible = 'On';
22        app.MesEditFieldLabel_2.Enable  = 'On';
23
24        app.DaEditField.Visible = 'On';
25        app.DaEditField.Enable  = 'On';
26        app.DaEditFieldLabel_2.Visible = 'On';
27        app.DaEditFieldLabel_2.Enable  = 'On';
28
29        app.HoraEditField.Visible = 'On';
30        app.HoraEditField.Enable  = 'On';
31        app.HoraEditFieldLabel_2.Visible = 'On';
32        app.HoraEditFieldLabel_2.Enable  = 'On';
33

```

```
34     app.MinutoEditField.Visible = 'On';
35     app.MinutoEditField.Enable = 'On';
36     app.MinutoEditFieldLabel_2.Visible = 'On';
37     app.MinutoEditFieldLabel_2.Enable = 'On';
38
39     app.SegundoEditField.Visible = 'On';
40     app.SegundoEditField.Enable = 'On';
41     app.SegundoEditFieldLabel_2.Visible = 'On';
42     app.SegundoEditFieldLabel_2.Enable = 'On';
43
44     % Titulo del Impulso
45     app.ValordelImpulsonmeroEditField.Visible = 'On';
46     app.ValordelImpulsonmeroEditField.Enable = 'On';
47     app.ValordelImpulsonmeroEditFieldLabel.Visible = 'On';
48     app.ValordelImpulsonmeroEditFieldLabel.Enable = 'On';
49
50     % Valores del Impulso
51     app.VxEditField.Visible = 'On';
52     app.VxEditField.Enable = 'On';
53     app.VxLabel.Visible = 'On';
54     app.VxLabel.Enable = 'On';
55
56     app.VyEditField.Visible = 'On';
57     app.VyEditField.Enable = 'On';
58     app.VyLabel.Visible = 'On';
59     app.VyLabel.Enable = 'On';
60
61     app.VzEditField.Visible = 'On';
62     app.VzEditField.Enable = 'On';
63     app.VzLabel.Visible = 'On';
64     app.VzLabel.Enable = 'On';
65
66     % Guardar
67     app.GuardarDatosButton.Visible = 'On';
68     app.GuardarDatosButton.Enable = 'On';
69
70
71     app.FechadelImpulsonmeroEditField.Value = 1;
72     app.ValordelImpulsonmeroEditField.Value = 1;
73     else
74
75     % Titulo de Fecha
76     app.FechadelImpulsonmeroEditField.Visible = 'Off';
77     app.FechadelImpulsonmeroEditField.Enable = 'Off';
78     app.FechadelImpulsonmeroEditFieldLabel.Visible = 'Off';
79     app.FechadelImpulsonmeroEditFieldLabel.Enable = 'Off';
80
81     % Valores de Fecha
82     app.AoEditField.Visible = 'Off';
83     app.AoEditField.Enable = 'Off';
84     app.AoEditFieldLabel_2.Visible = 'Off';
85     app.AoEditFieldLabel_2.Enable = 'Off';
86
87     app.MesEditField.Visible = 'Off';
88     app.MesEditField.Enable = 'Off';
89     app.MesEditFieldLabel_2.Visible = 'Off';
90     app.MesEditFieldLabel_2.Enable = 'Off';
91
92     app.DaEditField.Visible = 'Off';
93     app.DaEditField.Enable = 'Off';
94     app.DaEditFieldLabel_2.Visible = 'Off';
```

```

95     app.DaEditFieldLabel_2.Enable = 'Off';
96
97     app.HoraEditField.Visible = 'Off';
98     app.HoraEditField.Enable = 'Off';
99     app.HoraEditFieldLabel_2.Visible = 'Off';
100    app.HoraEditFieldLabel_2.Enable = 'Off';
101
102    app.MinutoEditField.Visible = 'Off';
103    app.MinutoEditField.Enable = 'Off';
104    app.MinutoEditFieldLabel_2.Visible = 'Off';
105    app.MinutoEditFieldLabel_2.Enable = 'Off';
106
107    app.SegundoEditField.Visible = 'Off';
108    app.SegundoEditField.Enable = 'Off';
109    app.SegundoEditFieldLabel_2.Visible = 'Off';
110    app.SegundoEditFieldLabel_2.Enable = 'Off';
111
112    % Titulo del Impulso
113    app.ValordelImpulsonmeroEditField.Visible = 'Off';
114    app.ValordelImpulsonmeroEditField.Enable = 'Off';
115    app.ValordelImpulsonmeroEditFieldLabel.Visible = 'Off';
116    app.ValordelImpulsonmeroEditFieldLabel.Enable = 'Off';
117
118    % Valores del Impulso
119    app.VxEditField.Visible = 'Off';
120    app.VxEditField.Enable = 'Off';
121    app.VxLabel.Visible = 'Off';
122    app.VxLabel.Enable = 'Off';
123
124    app.VyEditField.Visible = 'Off';
125    app.VyEditField.Enable = 'Off';
126    app.VyLabel.Visible = 'Off';
127    app.VyLabel.Enable = 'Off';
128
129    app.VzEditField.Visible = 'Off';
130    app.VzEditField.Enable = 'Off';
131    app.VzLabel.Visible = 'Off';
132    app.VzLabel.Enable = 'Off';
133
134    % Guardar
135    app.GuardarDatosButton.Visible = 'Off';
136    app.GuardarDatosButton.Enable = 'Off';
137
138    app.FechadelImpulsonmeroEditField.Value = 0;
139    app.ValordelImpulsonmeroEditField.Value = 0;
140    end
141 end

```

Listing 4: Función en la aplicación por el botón Guardar.

```

1
2 function GuardarDatosButtonPushed(app, event)
3
4     indice = app.FechadelImpulsonmeroEditField.Value;
5
6     if indice > 0
7         % Valores Fecha
8         year = app.AoEditField.Value;
9         month = app.MesEditField.Value;

```

```
10     day = app.DaEditField.Value;
11     hour = app.HoraEditField.Value;
12     min = app.MinutoEditField.Value;
13     sec = app.SegundoEditField.Value;
14
15     date = [year, month, day, hour, min, sec];
16
17     % Valores Impulso
18     Vx = app.VxEditField.Value;
19     Vy = app.VyEditField.Value;
20     Vz = app.VzEditField.Value;
21
22     vel= [Vx, Vy, Vz];
23
24     save(['Impulso_' num2str(indice) '.txt'], ...
25         'date', 'vel', '-ascii', '-double');
26
27     if indice < app.NmerodeImpusosTotalesSpinner.Value
28         app.FechadelImpulsonmeroEditField.Value = indice + 1;
29         app.ValordelImpulsonmeroEditField.Value = indice + 1;
30
31         % Reinicio de los valores
32         app.AoEditField.Value = 0;
33         app.MesEditField.Value = 0;
34         app.DaEditField.Value = 0;
35         app.HoraEditField.Value = 0;
36         app.MinutoEditField.Value = 0;
37         app.SegundoEditField.Value = 0;
38
39         app.VxEditField.Value = 0;
40         app.VyEditField.Value = 0;
41         app.VzEditField.Value = 0;
42
43     elseif indice==app.NmerodeImpusosTotalesSpinner.Value
44
45         % Reinicio de los valores
46         app.AoEditField.Value = 0;
47         app.MesEditField.Value = 0;
48         app.DaEditField.Value = 0;
49         app.HoraEditField.Value = 0;
50         app.MinutoEditField.Value = 0;
51         app.SegundoEditField.Value = 0;
52
53         app.VxEditField.Value = 0;
54         app.VyEditField.Value = 0;
55         app.VzEditField.Value = 0;
56
57         app.OpcionesdeIntegracinLabel.Enable = 'On';
58         app.TwoBodyCheckBox.Enable = 'On';
59         app.PlanetasCheckBox.Enable = 'On';
60         app.SRPCheckBox.Enable = 'On';
61
62         if isequal(app.FocodelarbitaDropDown.Value, 'Sol')
63             app.ZonalHarmonicsCheckBox.Enable = 'Off';
64             app.AtmosferaCheckBox.Enable = 'Off';
65         else
66             app.ZonalHarmonicsCheckBox.Enable = 'On';
67             app.AtmosferaCheckBox.Enable = 'On';
68         end
69
70     app.GuardarDatosButton.Enable = 'Off';
```

```

71
72         end
73     end
74 end

```

Listing 5: Función en la aplicación por el *State Button* No.

```

1
2 function NoButtonValueChanged(app, event)
3     value = app.NoButton.Value;
4
5     if value == 1
6         app.OpcionesdeIntegracinLabel.Enable = 'On';
7         app.TwoBodyCheckBox.Enable           = 'On';
8         app.PlanetasCheckBox.Enable         = 'On';
9         app.SRPCheckBox.Enable               = 'On';
10
11         if isequal(app.FocodelarbitaDropDown.Value, 'Sol')
12             app.ZonalHarmonicsCheckBox.Enable = 'Off';
13             app.AtmosferaCheckBox.Enable     = 'Off';
14         else
15             app.ZonalHarmonicsCheckBox.Enable = 'On';
16             app.AtmosferaCheckBox.Enable     = 'On';
17         end
18
19         app.SButton.Enable = 'Off';
20
21     elseif value == 0
22         app.SButton.Enable = 'On';
23
24         % Opciones de integracion
25         app.OpcionesdeIntegracinLabel.Enable = 'Off';
26         app.TwoBodyCheckBox.Enable           = 'Off';
27         app.PlanetasCheckBox.Enable         = 'Off';
28         app.SRPCheckBox.Enable               = 'Off';
29         app.AtmosferaCheckBox.Enable         = 'Off';
30         app.ZonalHarmonicsCheckBox.Enable    = 'Off';
31
32         app.ClculoButton.Enable = 'Off';
33
34         % Planetas
35
36         app.PlanetasLabel.Visible = 'Off';
37         app.PlanetasLabel.Enable  = 'Off';
38
39         app.TodosButton.Visible = 'Off';
40         app.TodosButton.Enable  = 'Off';
41         app.LunaButton.Visible  = 'Off';
42         app.LunaButton.Enable   = 'Off';
43         app.SolButton.Visible   = 'Off';
44         app.SolButton.Enable    = 'Off';
45         app.MercurioButton.Visible = 'Off';
46         app.MercurioButton.Enable = 'Off';
47         app.VenusButton.Visible  = 'Off';
48         app.VenusButton.Enable   = 'Off';
49         app.TierraButton.Visible = 'Off';
50         app.TierraButton.Enable  = 'Off';
51         app.MarteButton.Visible  = 'Off';
52         app.MarteButton.Enable   = 'Off';

```

```
53     app.JpiterButton.Visible = 'Off';
54     app.JpiterButton.Enable = 'Off';
55     app.SaturnoButton.Visible = 'Off';
56     app.SaturnoButton.Enable = 'Off';
57     app.UranoButton.Visible = 'Off';
58     app.UranoButton.Enable = 'Off';
59     app.NeptunoButton.Visible = 'Off';
60     app.NeptunoButton.Enable = 'Off';
61
62     app.TodosButton.Value = 0;
63     app.LunaButton.Value = 0;
64     app.SolButton.Value = 0;
65     app.MercurioButton.Value = 0;
66     app.VenusButton.Value = 0;
67     app.TierraButton.Value = 0;
68     app.MarteButton.Value = 0;
69     app.JpiterButton.Value = 0;
70     app.SaturnoButton.Value = 0;
71     app.UranoButton.Value = 0;
72     app.NeptunoButton.Value = 0;
73
74     % SRP y Atm
75
76     app.DatosSatlliteLabel.Visible = 'Off';
77     app.DatosSatlliteLabel.Enable = 'Off';
78
79     app.CREditFieldLabel.Visible = 'Off';
80     app.CREditFieldLabel.Enable = 'Off';
81     app.CREditField.Visible = 'Off';
82     app.CREditField.Enable = 'Off';
83
84     app.AmEditFieldLabel.Visible = 'Off';
85     app.AmEditFieldLabel.Enable = 'Off';
86     app.AmEditField.Visible = 'Off';
87     app.AmEditField.Enable = 'Off';
88
89     app.CDEditFieldLabel.Visible = 'Off';
90     app.CDEditFieldLabel.Enable = 'Off';
91     app.CDEditField.Visible = 'Off';
92     app.CDEditField.Enable = 'Off';
93
94     app.CREditField.Value = 0;
95     app.CDEditField.Value = 0;
96     app.AmEditField.Value = 0;
97
98     % Zona Harmonics
99
100     app.ZonalHarmonicsLabel.Visible = 'Off';
101     app.ZonalHarmonicsLabel.Enable = 'Off';
102
103     app.TodosButton_2.Visible = 'Off';
104     app.TodosButton_2.Enable = 'Off';
105
106     app.J2Button.Visible = 'Off';
107     app.J2Button.Enable = 'Off';
108     app.J3Button.Visible = 'Off';
109     app.J3Button.Enable = 'Off';
110     app.J4Button.Visible = 'Off';
111     app.J4Button.Enable = 'Off';
112     app.J5Button.Visible = 'Off';
113     app.J5Button.Enable = 'Off';
```

```

114         app.J6Button.Visible = 'Off';
115         app.J6Button.Enable = 'Off';
116
117         app.TodosButton_2.Value = 0;
118         app.J2Button.Value = 0;
119         app.J3Button.Value = 0;
120         app.J4Button.Value = 0;
121         app.J5Button.Value = 0;
122         app.J6Button.Value = 0;
123
124     end
125
126     value_B = app.SButton.Value;
127
128     if value == 0 && value_B == 0
129
130         app.TwoBodyCheckBox.Value = 0;
131         app.PlanetasCheckBox.Value = 0;
132         app.SRPCheckBox.Value = 0;
133         app.AtmsferaCheckBox.Value = 0;
134         app.ZonalHarmonicsCheckBox.Value = 0;
135     end
136
137 end

```

Listing 6: Función en la aplicación por la *Check Box* Two Body.

```

1
2     function TwoBodyCheckBoxValueChanged(app, event)
3         value = app.TwoBodyCheckBox.Value;
4
5         if value == 1
6             app.ClculoButton.Enable = 'On';
7
8             app.ZonalHarmonicsCheckBox.Enable = 'Off';
9             app.AtmsferaCheckBox.Enable = 'Off';
10            app.PlanetasCheckBox.Enable = 'Off';
11            app.SRPCheckBox.Enable = 'Off';
12
13        else
14
15            app.ClculoButton.Enable = 'Off';
16
17            app.PlanetasCheckBox.Enable = 'On';
18            app.SRPCheckBox.Enable = 'On';
19
20            if isequal(app.FocodelarbitaDropDown.Value, 'Tierra')
21
22                app.ZonalHarmonicsCheckBox.Enable = 'On';
23                app.AtmsferaCheckBox.Enable = 'On';
24            else
25
26            end
27        end
28
29    end
30

```

Listing 7: Función en la aplicación por la *Check Box* Zonal Harmonics.

```

1
2 function ZonalHarmonicsCheckBoxValueChanged(app, event)
3     value = app.ZonalHarmonicsCheckBox.Value;
4     if value == 1
5
6         app.ClculoButton.Enable = 'Off';
7
8         app.TwoBodyCheckBox.Enable = 'Off';
9
10        app.ZonalHarmonicsLabel.Visible = 'On';
11        app.ZonalHarmonicsLabel.Enable = 'On';
12
13        app.TodosButton_2.Visible = 'On';
14        app.TodosButton_2.Enable = 'On';
15
16        app.J2Button.Visible = 'On';
17        app.J2Button.Enable = 'On';
18        app.J3Button.Visible = 'On';
19        app.J3Button.Enable = 'On';
20        app.J4Button.Visible = 'On';
21        app.J4Button.Enable = 'On';
22        app.J5Button.Visible = 'On';
23        app.J5Button.Enable = 'On';
24        app.J6Button.Visible = 'On';
25        app.J6Button.Enable = 'On';
26
27
28    else
29        app.ZonalHarmonicsLabel.Visible = 'Off';
30        app.ZonalHarmonicsLabel.Enable = 'Off';
31
32        app.TodosButton_2.Visible = 'Off';
33        app.TodosButton_2.Enable = 'Off';
34
35        app.J2Button.Visible = 'Off';
36        app.J2Button.Enable = 'Off';
37        app.J3Button.Visible = 'Off';
38        app.J3Button.Enable = 'Off';
39        app.J4Button.Visible = 'Off';
40        app.J4Button.Enable = 'Off';
41        app.J5Button.Visible = 'Off';
42        app.J5Button.Enable = 'Off';
43        app.J6Button.Visible = 'Off';
44        app.J6Button.Enable = 'Off';
45
46        app.TodosButton_2.Value = 0;
47        app.J2Button.Value = 0;
48        app.J3Button.Value = 0;
49        app.J4Button.Value = 0;
50        app.J5Button.Value = 0;
51        app.J6Button.Value = 0;
52
53        if app.AtmosferaCheckBox.Value == 0 ...
54            && app.SRPCheckBox.Value == 0 ...
55            && app.PlanetasCheckBox.Value == 0
56
57            app.TwoBodyCheckBox.Enable = 'On';
58            app.ClculoButton.Enable = 'Off';
59        end

```

```

60         if app.AtmosferaCheckBox.Value == 1
61             if app.AmAtmEditField.Value ≠ 0 && app.CDEditField.Value ≠ 0
62                 app.ClculoButton.Enable = 'On';
63             else
64                 app.ClculoButton.Enable = 'Off';
65             end
66         end
67         if app.SRPCheckBox.Value == 1
68             if app.CREditField.Value ≠ 0 && app.AmsRPEditField.Value ≠ 0
69                 app.ClculoButton.Enable = 'On';
70             else
71                 app.ClculoButton.Enable = 'Off';
72             end
73         end
74         if app.PlanetasCheckBox.Value == 1
75             if app.TodosButton.Value == 0 && app.SolButton.Value == 0 ...
76                 && app.LunaButton.Value == 0 ...
77                 && app.MercurioButton.Value == 0 ...
78                 && app.VenusButton.Value == 0 ...
79                 && app.MarteButton.Value == 0 ...
80                 && app.JpiterButton.Value == 0 ...
81                 && app.SaturnoButton.Value == 0 ...
82                 && app.UranoButton.Value == 0 ...
83                 && app.NeptunoButton.Value == 0
84
85                 app.ClculoButton.Enable = 'Off';
86             else
87                 app.ClculoButton.Enable = 'On';
88
89             end
90
91         end
92
93     end
94 end

```

Listing 8: Función en la aplicación por la *Check Box* Atmósfera.

```

1
2 function AtmosferaCheckBoxValueChanged(app, event)
3     value = app.AtmosferaCheckBox.Value;
4
5     if value == 1
6
7         app.ClculoButton.Enable = 'Off';
8
9         app.TwoBodyCheckBox.Enable = 'Off';
10
11        app.DatosSatliteLabel.Visible = 'On';
12        app.DatosSatliteLabel.Enable = 'On';
13
14        app.CDEditFieldLabel.Visible = 'On';
15        app.CDEditFieldLabel.Enable = 'On';
16        app.CDEditField.Visible = 'On';
17        app.CDEditField.Enable = 'On';
18
19        app.AmAtmEditFieldLabel.Visible = 'On';
20        app.AmAtmEditFieldLabel.Enable = 'On';
21        app.AmAtmEditField.Visible = 'On';

```

```
22         app.AmAtmEditField.Enable           = 'On';
23
24     else
25
26         app.CEditFieldLabel.Visible = 'Off';
27         app.CEditFieldLabel.Enable  = 'Off';
28         app.CEditField.Visible      = 'Off';
29         app.CEditField.Enable       = 'Off';
30
31         app.CEditField.Value = 0;
32
33         if app.SRPCheckBox.Value == 0
34             app.DatosSatliteLabel.Visible = 'Off';
35             app.DatosSatliteLabel.Enable  = 'Off';
36         end
37
38         app.AmAtmEditFieldLabel.Visible = 'Off';
39         app.AmAtmEditFieldLabel.Enable  = 'Off';
40         app.AmAtmEditField.Visible     = 'Off';
41         app.AmAtmEditField.Enable       = 'Off';
42
43         app.AmAtmEditField.Value = 0;
44
45
46         if app.ZonalHarmonicsCheckBox.Value == 0 ...
47             && app.SRPCheckBox.Value == 0 ...
48             && app.PlanetasCheckBox.Value == 0
49
50             app.TwoBodyCheckBox.Enable = 'On';
51             app.ClculoButton.Enable = 'Off';
52         end
53         if app.ZonalHarmonicsCheckBox.Value == 1
54             if app.TodosButton_2.Value == 0 ...
55                 && app.J2Button.Value == 0 ...
56                 && app.J3Button.Value == 0 ...
57                 && app.J4Button.Value == 0 ...
58                 && app.J5Button.Value == 0 ...
59                 && app.J6Button.Value == 0
60
61                 app.ClculoButton.Enable = 'Off';
62             else
63                 app.ClculoButton.Enable = 'On';
64             end
65         end
66         if app.SRPCheckBox.Value == 1
67             if app.CREditField.Value ≠ 0 && app.AmSRPEditField.Value ≠ 0
68                 app.ClculoButton.Enable = 'On';
69             else
70                 app.ClculoButton.Enable = 'Off';
71             end
72         end
73         if app.PlanetasCheckBox.Value == 1
74             if app.TodosButton.Value == 0 && app.SolButton.Value == 0 ...
75                 && app.LunaButton.Value == 0 ...
76                 && app.MercurioButton.Value == 0 ...
77                 && app.VenusButton.Value == 0 ...
78                 && app.MarteButton.Value == 0 ...
79                 && app.JpiterButton.Value == 0 ...
80                 && app.SaturnoButton.Value == 0 ...
81                 && app.UranoButton.Value == 0 ...
82                 && app.NeptunoButton.Value == 0
```

```

83
84         app.ClculoButton.Enable = 'Off';
85     else
86         app.ClculoButton.Enable = 'On';
87
88     end
89
90     end
91
92     end
93     end

```

Listing 9: Función en la aplicación por la *Check Box* SRP (*Solar Radiation Presure*).

```

1
2 function SRPCheckBoxValueChanged(app, event)
3     value = app.SRPCheckBox.Value;
4     if value == 1
5
6         app.ClculoButton.Enable = 'Off';
7
8         app.TwoBodyCheckBox.Enable = 'Off';
9
10        app.DatosSatlliteLabel.Visible      = 'On';
11        app.DatosSatlliteLabel.Enable      = 'On';
12
13        app.CREditFieldLabel.Visible = 'On';
14        app.CREditFieldLabel.Enable = 'On';
15        app.CREditField.Visible = 'On';
16        app.CREditField.Enable = 'On';
17
18        app.AmSRPEditFieldLabel.Visible = 'On';
19        app.AmSRPEditFieldLabel.Enable = 'On';
20        app.AmSRPEditField.Visible = 'On';
21        app.AmSRPEditField.Enable = 'On';
22
23    else
24
25        app.CREditFieldLabel.Visible = 'Off';
26        app.CREditFieldLabel.Enable = 'Off';
27        app.CREditField.Visible = 'Off';
28        app.CREditField.Enable = 'Off';
29
30        app.CREditField.Value = 0;
31
32
33        if app.AtmsferaCheckBox.Value == 0
34            app.DatosSatlliteLabel.Visible      = 'Off';
35            app.DatosSatlliteLabel.Enable      = 'Off';
36        end
37
38        app.AmSRPEditFieldLabel.Visible = 'Off';
39        app.AmSRPEditFieldLabel.Enable = 'Off';
40        app.AmSRPEditField.Visible = 'Off';
41        app.AmSRPEditField.Enable = 'Off';
42
43        app.AmSRPEditField.Value = 0;
44
45

```

```
46     if app.ZonalHarmonicsCheckBox.Value == 0 ...
47         && app.AtmsferaCheckBox.Value == 0 ...
48         && app.PlanetasCheckBox.Value == 0
49
50     app.TwoBodyCheckBox.Enable = 'On';
51     app.ClculoButton.Enable = 'Off';
52     end
53     if app.AtmsferaCheckBox.Value == 1
54         if app.AmAtmEditField.Value ≠ 0 && app.CDEditField.Value ≠ 0
55             app.ClculoButton.Enable = 'On';
56         else
57             app.ClculoButton.Enable = 'Off';
58         end
59     end
60     if app.ZonalHarmonicsCheckBox.Value == 1
61         if app.TodosButton_2.Value == 0 ...
62             && app.J2Button.Value == 0 ...
63             && app.J3Button.Value == 0 ...
64             && app.J4Button.Value == 0 ...
65             && app.J5Button.Value == 0 ...
66             && app.J6Button.Value == 0
67
68             app.ClculoButton.Enable = 'Off';
69         else
70             app.ClculoButton.Enable = 'On';
71         end
72     end
73     if app.PlanetasCheckBox.Value == 1
74         if isequal(app.FocodelarbitaDropDown.Value, 'Sol')
75             if app.TodosButton.Value == 0 ...
76                 && app.LunaButton.Value == 0 ...
77                 && app.MercurioButton.Value == 0 ...
78                 && app.VenusButton.Value == 0 ...
79                 && app.TierraButton.Value == 0 ...
80                 && app.MarteButton.Value == 0 ...
81                 && app.JpiterButton.Value == 0 ...
82                 && app.SaturnoButton.Value == 0 ...
83                 && app.UranoButton.Value == 0 ...
84                 && app.NeptunoButton.Value == 0
85
86                 app.ClculoButton.Enable = 'Off';
87             else
88                 app.ClculoButton.Enable = 'On';
89             end
90         elseif isequal(app.FocodelarbitaDropDown.Value, 'Tierra')
91             if app.TodosButton.Value == 0 ...
92                 && app.SolButton.Value == 0 ...
93                 && app.LunaButton.Value == 0 ...
94                 && app.MercurioButton.Value == 0 ...
95                 && app.VenusButton.Value == 0 ...
96                 && app.MarteButton.Value == 0 ...
97                 && app.JpiterButton.Value == 0 ...
98                 && app.SaturnoButton.Value == 0 ...
99                 && app.UranoButton.Value == 0 ...
100                 && app.NeptunoButton.Value == 0
101
102                 app.ClculoButton.Enable = 'Off';
103             else
104                 app.ClculoButton.Enable = 'On';
105             end
106         end
107     end
end
```

```

107
108         end
109
110     end
111 end

```

Listing 10: Función en la aplicación por la *Check Box* Planetas.

```

1
2     function PlanetasCheckBoxValueChanged(app, event)
3         value = app.PlanetasCheckBox.Value;
4
5         if value == 1
6             app.ClculoButton.Enable = 'Off';
7
8             app.TwoBodyCheckBox.Enable = 'Off';
9
10            % Planetas
11
12            app.PlanetasLabel.Visible = 'On';
13            app.PlanetasLabel.Enable = 'On';
14
15            app.TodosButton.Visible = 'On';
16            app.TodosButton.Enable = 'On';
17            app.LunaButton.Visible = 'On';
18            app.LunaButton.Enable = 'On';
19            app.MercurioButton.Visible = 'On';
20            app.MercurioButton.Enable = 'On';
21            app.VenusButton.Visible = 'On';
22            app.VenusButton.Enable = 'On';
23            if isequal(app.FocodelarbitaDropDown.Value, 'Sol')
24                app.TierraButton.Visible = 'On';
25                app.TierraButton.Enable = 'On';
26                app.SolButton.Enable = 'Off';
27                app.SolButton.Visible = 'Off';
28            elseif isequal(app.FocodelarbitaDropDown.Value, 'Tierra')
29                app.TierraButton.Visible = 'Off';
30                app.TierraButton.Enable = 'Off';
31                app.SolButton.Enable = 'On';
32                app.SolButton.Visible = 'On';
33            end
34            app.MarteButton.Visible = 'On';
35            app.MarteButton.Enable = 'On';
36            app.JpiterButton.Visible = 'On';
37            app.JpiterButton.Enable = 'On';
38            app.SaturnoButton.Visible = 'On';
39            app.SaturnoButton.Enable = 'On';
40            app.UranoButton.Visible = 'On';
41            app.UranoButton.Enable = 'On';
42            app.NeptunoButton.Visible = 'On';
43            app.NeptunoButton.Enable = 'On';
44
45
46        else
47
48            % Planetas
49
50            app.PlanetasLabel.Visible = 'Off';
51            app.PlanetasLabel.Enable = 'Off';

```

```
52
53     app.TodosButton.Visible = 'Off';
54     app.TodosButton.Enable = 'Off';
55     app.LunaButton.Visible = 'Off';
56     app.LunaButton.Enable = 'Off';
57     app.SolButton.Enable = 'Off';
58     app.SolButton.Visible = 'Off';
59     app.MercurioButton.Visible = 'Off';
60     app.MercurioButton.Enable = 'Off';
61     app.VenusButton.Visible = 'Off';
62     app.VenusButton.Enable = 'Off';
63     app.TierraButton.Visible = 'Off';
64     app.TierraButton.Enable = 'Off';
65     app.MarteButton.Visible = 'Off';
66     app.MarteButton.Enable = 'Off';
67     app.JpiterButton.Visible = 'Off';
68     app.JpiterButton.Enable = 'Off';
69     app.SaturnoButton.Visible = 'Off';
70     app.SaturnoButton.Enable = 'Off';
71     app.UranoButton.Visible = 'Off';
72     app.UranoButton.Enable = 'Off';
73     app.NeptunoButton.Visible = 'Off';
74     app.NeptunoButton.Enable = 'Off';
75
76
77     app.TodosButton.Value = 0;
78     app.LunaButton.Value = 0;
79     app.SolButton.Value = 0;
80     app.MercurioButton.Value = 0;
81     app.VenusButton.Value = 0;
82     app.TierraButton.Value = 0;
83     app.MarteButton.Value = 0;
84     app.JpiterButton.Value = 0;
85     app.SaturnoButton.Value = 0;
86     app.UranoButton.Value = 0;
87     app.NeptunoButton.Value = 0;
88
89     if app.ZonalHarmonicsCheckBox.Value == 0 ...
90         && app.AtmsferaCheckBox.Value == 0 ...
91         && app.SRPCheckBox.Value == 0
92
93     app.TwoBodyCheckBox.Enable = 'On';
94     app.ClculoButton.Enable = 'Off';
95     end
96     if app.ZonalHarmonicsCheckBox.Value == 1
97         if app.TodosButton2.Value == 0 ...
98             && app.J2Button.Value == 0 ...
99             && app.J3Button.Value == 0 ...
100             && app.J4Button.Value == 0 ...
101             && app.J5Button.Value == 0 ...
102             && app.J6Button.Value == 0
103
104             app.ClculoButton.Enable = 'Off';
105         else
106             app.ClculoButton.Enable = 'On';
107         end
108     end
109     if app.AtmsferaCheckBox.Value == 1
110         if app.AmAtmEditField.Value ≠ 0 && app.CDEditField.Value ≠ 0
111             app.ClculoButton.Enable = 'On';
112         else
```

```

113         app.ClculoButton.Enable = 'Off';
114     end
115 end
116 if app.SRPCheckBox.Value == 1
117     if app.CREditField.Value ≠ 0 && app.AmsRPeditField.Value ≠ 0
118         app.ClculoButton.Enable = 'On';
119     else
120         app.ClculoButton.Enable = 'Off';
121     end
122 end
123 end
124 end

```

Listing 11: Función en la aplicación por el *Button* Todos (*Zonal Harmonics*).

```

1
2 function TodosButton_2ValueChanged(app, event)
3     value = app.TodosButton_2.Value;
4     if value == 0
5         app.ClculoButton.Enable = 'Off';
6
7         app.J2Button.Value = 0;
8         app.J3Button.Value = 0;
9         app.J4Button.Value = 0;
10        app.J5Button.Value = 0;
11        app.J6Button.Value = 0;
12
13    else
14
15        app.J2Button.Value = 1;
16        app.J3Button.Value = 1;
17        app.J4Button.Value = 1;
18        app.J5Button.Value = 1;
19        app.J6Button.Value = 1;
20
21        if app.AtmsferaCheckBox.Value == 0 ...
22            && app.SRPCheckBox.Value == 0 ...
23            && app.PlanetasCheckBox.Value == 0
24            app.ClculoButton.Enable = 'On';
25        end
26        if app.AtmsferaCheckBox.Value == 1
27            if app.AmAtmEditField.Value ≠ 0 && app.CDEditField.Value ≠ 0
28                app.ClculoButton.Enable = 'On';
29            else
30                app.ClculoButton.Enable = 'Off';
31            end
32        end
33        if app.SRPCheckBox.Value == 1
34            if app.CREditField.Value ≠ 0 && app.AmsRPeditField.Value ≠ 0
35                app.ClculoButton.Enable = 'On';
36            else
37                app.ClculoButton.Enable = 'Off';
38            end
39        end
40        if app.PlanetasCheckBox.Value == 1
41            if app.TodosButton.Value == 0 && app.SolButton.Value == 0 ...
42                && app.LunaButton.Value == 0 ...
43                && app.MercurioButton.Value == 0 ...
44                && app.VenusButton.Value == 0 ...

```

```

45         && app.MarteButton.Value == 0 ...
46         && app.JpiterButton.Value == 0 ...
47         && app.SaturnoButton.Value == 0 ...
48         && app.UranoButton.Value == 0 ...
49         && app.NeptunoButton.Value == 0
50
51         app.ClculoButton.Enable = 'Off';
52
53         else
54             app.ClculoButton.Enable = 'On';
55
56         end
57     end
58
59 end
60 end

```

Listing 12: Función en la aplicación por el *Button J2 (Zonal Harmonics)*.

```

1
2 function J2ButtonValueChanged(app, event)
3     value = app.J2Button.Value;
4     if value == 1
5         if app.AtmsferaCheckBox.Value == 0 ...
6             && app.SRPCheckBox.Value == 0 ...
7             && app.PlanetasCheckBox.Value == 0
8             app.ClculoButton.Enable = 'On';
9         end
10        if app.AtmsferaCheckBox.Value == 1
11            if app.AmAtmEditField.Value ≠ 0 && app.CEditField.Value ≠ 0
12                app.ClculoButton.Enable = 'On';
13            else
14                app.ClculoButton.Enable = 'Off';
15            end
16        end
17        if app.SRPCheckBox.Value == 1
18            if app.CREditField.Value ≠ 0 ...
19                && app.AmsRPEditField.Value ≠ 0
20                app.ClculoButton.Enable = 'On';
21            else
22                app.ClculoButton.Enable = 'Off';
23            end
24        end
25        if app.PlanetasCheckBox.Value == 1
26            if app.TodosButton.Value == 0 && app.SolButton.Value == 0 ...
27                && app.LunaButton.Value == 0 ...
28                && app.MercurioButton.Value == 0 ...
29                && app.VenusButton.Value == 0 ...
30                && app.MarteButton.Value == 0 ...
31                && app.JpiterButton.Value == 0 ...
32                && app.SaturnoButton.Value == 0 ...
33                && app.UranoButton.Value == 0 ...
34                && app.NeptunoButton.Value == 0
35
36            app.ClculoButton.Enable = 'Off';
37            else
38                app.ClculoButton.Enable = 'On';
39            end
40        end

```

```

41         else
42             if app.J3Button.Value == 0 && app.J4Button.Value == 0 ...
43                 && app.J5Button.Value == 0 && app.J6Button.Value == 0 ...
44                 && app.TodosButton_2.Value == 0
45                 app.ClculoButton.Enable = 'Off';
46             end
47         end
48     end

```

Listing 13: Función en la aplicación por el *Button J3 (Zonal Harmonics)*.

```

1
2 function J3ButtonValueChanged(app, event)
3     value = app.J3Button.Value;
4     if value == 1
5         if app.AtmosferaCheckBox.Value == 0 ...
6             && app.SRPCheckBox.Value == 0 ...
7             && app.PlanetasCheckBox.Value == 0
8         app.ClculoButton.Enable = 'On';
9         end
10        if app.AtmosferaCheckBox.Value == 1
11            if app.AmAtmEditField.Value ≠ 0 && app.CEditField.Value ≠ 0
12                app.ClculoButton.Enable = 'On';
13            else
14                app.ClculoButton.Enable = 'Off';
15            end
16        end
17        if app.SRPCheckBox.Value == 1
18            if app.CEditField.Value ≠ 0 && app.AmSRPEditField.Value ≠ 0
19                app.ClculoButton.Enable = 'On';
20            else
21                app.ClculoButton.Enable = 'Off';
22            end
23        end
24        if app.PlanetasCheckBox.Value == 1
25            if app.TodosButton.Value == 0 && app.SolButton.Value == 0 ...
26                && app.LunaButton.Value == 0 ...
27                && app.MercurioButton.Value == 0 ...
28                && app.VenusButton.Value == 0 ...
29                && app.MarteButton.Value == 0 ...
30                && app.JpiterButton.Value == 0 ...
31                && app.SaturnoButton.Value == 0 ...
32                && app.UranoButton.Value == 0 ...
33                && app.NeptunoButton.Value == 0
34            app.ClculoButton.Enable = 'Off';
35            else
36                app.ClculoButton.Enable = 'On';
37            end
38        end
39    end
40    else
41        if app.J2Button.Value == 0 && app.J4Button.Value == 0 ...
42            && app.J5Button.Value == 0 && app.J6Button.Value == 0 ...
43            && app.TodosButton_2.Value == 0
44            app.ClculoButton.Enable = 'Off';
45        end
46    end
47 end
48

```

Listing 14: Función en la aplicación por el *Button J4 (Zonal Harmonics)*.

```

1
2 function J4ButtonValueChanged(app, event)
3     value = app.J4Button.Value;
4     if value == 1
5         if app.AtmosferaCheckBox.Value == 0 ...
6             && app.SRPCheckBox.Value == 0 ...
7             && app.PlanetasCheckBox.Value == 0
8             app.ClculoButton.Enable = 'On';
9         end
10        if app.AtmosferaCheckBox.Value == 1
11            if app.AmAtmEditField.Value ≠ 0 && app.CEditField.Value ≠ 0
12                app.ClculoButton.Enable = 'On';
13            else
14                app.ClculoButton.Enable = 'Off';
15            end
16        end
17        if app.SRPCheckBox.Value == 1
18            if app.CEditField.Value ≠ 0 && app.AmSRPEditField.Value ≠ 0
19                app.ClculoButton.Enable = 'On';
20            else
21                app.ClculoButton.Enable = 'Off';
22            end
23        end
24        if app.PlanetasCheckBox.Value == 1
25            if app.TodosButton.Value == 0 && app.SolButton.Value == 0 ...
26                && app.LunaButton.Value == 0 ...
27                && app.MercurioButton.Value == 0 ...
28                && app.VenusButton.Value == 0 ...
29                && app.MarteButton.Value == 0 ...
30                && app.JpiterButton.Value == 0 ...
31                && app.SaturnoButton.Value == 0 ...
32                && app.UranoButton.Value == 0 ...
33                && app.NeptunoButton.Value == 0
34
35            app.ClculoButton.Enable = 'Off';
36        else
37            app.ClculoButton.Enable = 'On';
38
39        end
40
41    end
42    else
43        if app.J2Button.Value == 0 && app.J3Button.Value == 0 ...
44            && app.J5Button.Value == 0 && app.J6Button.Value == 0 ...
45            && app.TodosButton2.Value == 0
46            app.ClculoButton.Enable = 'Off';
47        end
48    end
49 end

```

Listing 15: Función en la aplicación por el *Button J5 (Zonal Harmonics)*.

```

1
2 function J5ButtonValueChanged(app, event)
3     value = app.J5Button.Value;
4     if value == 1
5         if app.AtmosferaCheckBox.Value == 0 ...
6             && app.SRPCheckBox.Value == 0 ...

```

```

7         && app.PlanetasCheckBox.Value == 0
8     app.ClculoButton.Enable = 'On';
9     end
10    if app.AtmosferaCheckBox.Value == 1
11        if app.AmAtmEditField.Value ≠ 0 && app.CEditField.Value ≠ 0
12            app.ClculoButton.Enable = 'On';
13        else
14            app.ClculoButton.Enable = 'Off';
15        end
16    end
17    if app.SRPCheckBox.Value == 1
18        if app.CREditField.Value ≠ 0 && app.AmsRPEditField.Value ≠ 0
19            app.ClculoButton.Enable = 'On';
20        else
21            app.ClculoButton.Enable = 'Off';
22        end
23    end
24    if app.PlanetasCheckBox.Value == 1
25        if app.TodosButton.Value == 0 && app.SolButton.Value == 0 ...
26            && app.LunaButton.Value == 0 ...
27            && app.MercurioButton.Value == 0 ...
28            && app.VenusButton.Value == 0 ...
29            && app.MarteButton.Value == 0 ...
30            && app.JpiterButton.Value == 0 ...
31            && app.SaturnoButton.Value == 0 ...
32            && app.UranoButton.Value == 0 ...
33            && app.NeptunoButton.Value == 0
34
35        app.ClculoButton.Enable = 'Off';
36    else
37        app.ClculoButton.Enable = 'On';
38
39    end
40
41    end
42    else
43        if app.J2Button.Value == 0 && app.J3Button.Value == 0 ...
44            && app.J4Button.Value == 0 && app.J6Button.Value == 0 ...
45            && app.TodosButton.2.Value == 0
46        app.ClculoButton.Enable = 'Off';
47    end
48    end
49    end

```

Listing 16: Función en la aplicación por el *Button J6 (Zonal Harmonics)*.

```

1
2 function J6ButtonValueChanged(app, event)
3     value = app.J6Button.Value;
4     if value == 1
5         if app.AtmosferaCheckBox.Value == 0 ...
6             && app.SRPCheckBox.Value == 0 ...
7             && app.PlanetasCheckBox.Value == 0
8         app.ClculoButton.Enable = 'On';
9     end
10    if app.AtmosferaCheckBox.Value == 1
11        if app.AmAtmEditField.Value ≠ 0 && app.CEditField.Value ≠ 0
12            app.ClculoButton.Enable = 'On';
13        else

```

```

14         app.ClculoButton.Enable = 'Off';
15     end
16 end
17 if app.SRPCheckBox.Value == 1
18     if app.CREditField.Value ≠ 0 && app.AmSRPEditField.Value ≠ 0
19         app.ClculoButton.Enable = 'On';
20     else
21         app.ClculoButton.Enable = 'Off';
22     end
23 end
24 if app.PlanetasCheckBox.Value == 1
25     if app.TodosButton.Value == 0 && app.SolButton.Value == 0 ...
26         && app.LunaButton.Value == 0 ...
27         && app.MercurioButton.Value == 0 ...
28         && app.VenusButton.Value == 0 ...
29         && app.MarteButton.Value == 0 ...
30         && app.JpiterButton.Value == 0 ...
31         && app.SaturnoButton.Value == 0 ...
32         && app.UranoButton.Value == 0 ...
33         && app.NeptunoButton.Value == 0
34
35         app.ClculoButton.Enable = 'Off';
36     else
37         app.ClculoButton.Enable = 'On';
38
39     end
40
41     end
42 else
43     if app.J2Button.Value == 0 && app.J3Button.Value == 0 ...
44         && app.J4Button.Value == 0 && app.J5Button.Value == 0 ...
45         && app.TodosButton_2.Value == 0
46     app.ClculoButton.Enable = 'Off';
47     end
48 end
49 end

```

Listing 17: Función en la aplicación por el *Edit Fiel Am* (Atm) (Relación área/masa).

```

1
2 function AmAtmEditFieldValueChanged(app, event)
3     value = app.AmAtmEditField.Value;
4
5     if value ≠ 0 && app.CDEditField.Value ≠ 0
6         if app.ZonalHarmonicsCheckBox.Value == 0 ...
7             && app.SRPCheckBox.Value == 0 ...
8             && app.PlanetasCheckBox.Value == 0
9             app.ClculoButton.Enable = 'On';
10        end
11        if app.ZonalHarmonicsCheckBox.Value == 1
12            if app.TodosButton_2.Value == 0 ...
13                && app.J2Button.Value == 0 ...
14                && app.J3Button.Value == 0 ...
15                && app.J4Button.Value == 0 ...
16                && app.J5Button.Value == 0 ...
17                && app.J6Button.Value == 0
18
19            app.ClculoButton.Enable = 'Off';
20        else

```

```

21         app.ClculoButton.Enable = 'On';
22     end
23 end
24 if app.SRPCheckBox.Value == 1
25     if app.CREditField.Value ≠ 0 && app.AmSRPEditField.Value ≠ 0
26         app.ClculoButton.Enable = 'On';
27     else
28         app.ClculoButton.Enable = 'Off';
29     end
30 end
31 if app.PlanetasCheckBox.Value == 1
32     if app.TodosButton.Value == 0 && app.SolButton.Value == 0 ...
33         && app.LunaButton.Value == 0 ...
34         && app.MercurioButton.Value == 0 ...
35         && app.VenusButton.Value == 0 ...
36         && app.MarteButton.Value == 0 ...
37         && app.JpiterButton.Value == 0 ...
38         && app.SaturnoButton.Value == 0 ...
39         && app.UranoButton.Value == 0 ...
40         && app.NeptunoButton.Value == 0
41
42         app.ClculoButton.Enable = 'Off';
43
44     else
45         app.ClculoButton.Enable = 'On';
46
47     end
48 end
49 else
50     app.ClculoButton.Enable = 'Off';
51 end
52 end

```

Listing 18: Función en la aplicación por el *Edit Field* CD (Coeficiente de fricción).

```

1
2 function CDEditFieldValueChanged(app, event)
3     value = app.CDEditField.Value;
4
5     if value ≠ 0 && app.AmAtmEditField.Value ≠ 0
6         if app.ZonalHarmonicsCheckBox.Value == 0 ...
7             && app.SRPCheckBox.Value == 0 ...
8             && app.PlanetasCheckBox.Value == 0
9             app.ClculoButton.Enable = 'On';
10        end
11        if app.ZonalHarmonicsCheckBox.Value == 1
12            if app.TodosButton_2.Value == 0 ...
13                && app.J2Button.Value == 0 ...
14                && app.J3Button.Value == 0 ...
15                && app.J4Button.Value == 0 ...
16                && app.J5Button.Value == 0 ...
17                && app.J6Button.Value == 0
18
19                app.ClculoButton.Enable = 'Off';
20            else
21                app.ClculoButton.Enable = 'On';
22            end
23        end
24        if app.SRPCheckBox.Value == 1

```

```

25         if app.CREditField.Value ≠ 0 && app.AmSRPEditField.Value ≠ 0
26             app.ClculoButton.Enable = 'On';
27         else
28             app.ClculoButton.Enable = 'Off';
29         end
30     end
31     if app.PlanetasCheckBox.Value == 1
32         if app.TodosButton.Value == 0 && app.SolButton.Value == 0 ...
33             && app.LunaButton.Value == 0 ...
34             && app.MercurioButton.Value == 0 ...
35             && app.VenusButton.Value == 0 ...
36             && app.MarteButton.Value == 0 ...
37             && app.JpiterButton.Value == 0 ...
38             && app.SaturnoButton.Value == 0 ...
39             && app.UranoButton.Value == 0 ...
40             && app.NeptunoButton.Value == 0
41
42             app.ClculoButton.Enable = 'Off';
43         else
44             app.ClculoButton.Enable = 'On';
45
46         end
47     end
48 else
49     app.ClculoButton.Enable = 'Off';
50 end
51 end

```

Listing 19: Función en la aplicación por el *Edit Fiel Am* (SRP) (Relación área/masa).

```

1
2 function AmSRPEditFieldValueChanged(app, event)
3     value = app.AmSRPEditField.Value;
4
5     if value ≠ 0 && app.CREditField.Value ≠ 0
6         if app.ZonalHarmonicsCheckBox.Value == 0 ...
7             && app.SRPCheckBox.Value == 0 ...
8             && app.PlanetasCheckBox.Value == 0
9             app.ClculoButton.Enable = 'On';
10        end
11        if app.ZonalHarmonicsCheckBox.Value == 1
12            if app.TodosButton_2.Value == 0 ...
13                && app.J2Button.Value == 0 ...
14                && app.J3Button.Value == 0 ...
15                && app.J4Button.Value == 0 ...
16                && app.J5Button.Value == 0 ...
17                && app.J6Button.Value == 0
18
19                app.ClculoButton.Enable = 'Off';
20            else
21
22                app.ClculoButton.Enable = 'On';
23            end
24        end
25    if app.AtmCheckBox.Value == 1
26        if app.CDEditField.Value ≠ 0 && app.AmAtmEditField.Value ≠ 0
27            app.ClculoButton.Enable = 'On';
28        else
29            app.ClculoButton.Enable = 'Off';

```

```

30     end
31   end
32   if app.PlanetasCheckBox.Value == 1
33     if isequal(app.FocodelarbitaDropDown.Value, 'Sol')
34       if app.TodosButton.Value == 0 ...
35         && app.LunaButton.Value == 0 ...
36         && app.MercurioButton.Value == 0 ...
37         && app.VenusButton.Value == 0 ...
38         && app.TierraButton.Value == 0 ...
39         && app.MarteButton.Value == 0 ...
40         && app.JpiterButton.Value == 0 ...
41         && app.SaturnoButton.Value == 0 ...
42         && app.UranoButton.Value == 0 ...
43         && app.NeptunoButton.Value == 0
44
45         app.ClculoButton.Enable = 'Off';
46       else
47         app.ClculoButton.Enable = 'On';
48       end
49     elseif isequal(app.FocodelarbitaDropDown.Value, 'Tierra')
50       if app.TodosButton.Value == 0 ...
51         && app.SolButton.Value == 0 ...
52         && app.LunaButton.Value == 0 ...
53         && app.MercurioButton.Value == 0 ...
54         && app.VenusButton.Value == 0 ...
55         && app.MarteButton.Value == 0 ...
56         && app.JpiterButton.Value == 0 ...
57         && app.SaturnoButton.Value == 0 ...
58         && app.UranoButton.Value == 0 ...
59         && app.NeptunoButton.Value == 0
60
61         app.ClculoButton.Enable = 'Off';
62       else
63         app.ClculoButton.Enable = 'On';
64       end
65     end
66   end
67 else
68   app.ClculoButton.Enable = 'Off';
69 end
70 end
71 end

```

Listing 20: Función en la aplicación por el *Edit Field* CR (Coeficiente de reflectividad).

```

1
2 function CREditFieldValueChanged(app, event)
3     value = app.CREditField.Value;
4
5     if value ≠ 0 && app.AmSRPEditField.Value ≠ 0
6         if app.ZonalHarmonicsCheckBox.Value == 0 ...
7             && app.SRPCheckBox.Value == 0 ...
8             && app.PlanetasCheckBox.Value == 0
9             app.ClculoButton.Enable = 'On';
10        end
11        if app.ZonalHarmonicsCheckBox.Value == 1
12            if app.TodosButton-2.Value == 0 ...
13                && app.J2Button.Value == 0 ...
14                && app.J3Button.Value == 0 ...

```

```
15         && app.J4Button.Value == 0 ...
16         && app.J5Button.Value == 0 ...
17         && app.J6Button.Value == 0
18
19     app.ClculoButton.Enable = 'Off';
20     else
21         app.ClculoButton.Enable = 'On';
22     end
23 end
24 if app.AtmosferaCheckBox.Value == 1
25     if app.CDEditField.Value ≠ 0 ...
26         && app.AmAtmEditField.Value ≠ 0
27         app.ClculoButton.Enable = 'On';
28     else
29         app.ClculoButton.Enable = 'Off';
30     end
31 end
32 if app.PlanetasCheckBox.Value == 1
33     if isequal(app.FocodelarbitaDropDown.Value, 'Sol')
34         if app.TodosButton.Value == 0 ...
35             && app.LunaButton.Value == 0 ...
36             && app.MercurioButton.Value == 0 ...
37             && app.VenusButton.Value == 0 ...
38             && app.TierraButton.Value == 0 ...
39             && app.MarteButton.Value == 0 ...
40             && app.JpiterButton.Value == 0 ...
41             && app.SaturnoButton.Value == 0 ...
42             && app.UranoButton.Value == 0 ...
43             && app.NeptunoButton.Value == 0
44
45             app.ClculoButton.Enable = 'Off';
46         else
47             app.ClculoButton.Enable = 'On';
48         end
49     elseif isequal(app.FocodelarbitaDropDown.Value, 'Tierra')
50         if app.TodosButton.Value == 0 ...
51             && app.SolButton.Value == 0 ...
52             && app.LunaButton.Value == 0 ...
53             && app.MercurioButton.Value == 0 ...
54             && app.VenusButton.Value == 0 ...
55             && app.MarteButton.Value == 0 ...
56             && app.JpiterButton.Value == 0 ...
57             && app.SaturnoButton.Value == 0 ...
58             && app.UranoButton.Value == 0 ...
59             && app.NeptunoButton.Value == 0
60
61             app.ClculoButton.Enable = 'Off';
62         else
63             app.ClculoButton.Enable = 'On';
64         end
65     end
66 end
67 else
68     app.ClculoButton.Enable = 'Off';
69 end
70 end
```

Listing 21: Función en la aplicación por el *Button* Todos (Planetas).

```

1
2 function TodosButtonValueChanged(app, event)
3     value = app.TodosButton.Value;
4
5     if value == 0
6         app.ClculoButton.Enable = 'Off';
7         app.LunaButton.Value     = 0;
8         app.MercurioButton.Value = 0;
9         app.VenusButton.Value    = 0;
10        if isequal(app.FocodelarbitaDropDown.Value, 'Sol')
11            app.TierraButton.Value = 0;
12        elseif isequal(app.FocodelarbitaDropDown.Value, 'Tierra')
13            app.TierraButton.Enable = 'Off';
14            app.SolButton.Value    = 0;
15        end
16        app.MarteButton.Value     = 0;
17        app.JpiterButton.Value    = 0;
18        app.SaturnoButton.Value   = 0;
19        app.UranoButton.Value     = 0;
20        app.NeptunoButton.Value   = 0;
21    else
22        app.LunaButton.Value       = 1;
23        app.MercurioButton.Value   = 1;
24        app.VenusButton.Value      = 1;
25        if isequal(app.FocodelarbitaDropDown.Value, 'Sol')
26            app.TierraButton.Value = 1;
27        elseif isequal(app.FocodelarbitaDropDown.Value, 'Tierra')
28            app.TierraButton.Enable = 'Off';
29            app.SolButton.Value    = 1;
30        end
31        app.MarteButton.Value     = 1;
32        app.JpiterButton.Value    = 1;
33        app.SaturnoButton.Value   = 1;
34        app.UranoButton.Value     = 1;
35        app.NeptunoButton.Value   = 1;
36
37        if app.ZonalHarmonicsCheckBox.Value == 0 ...
38            && app.AtmsferaCheckBox.Value == 0 ...
39            && app.SRPCheckBox.Value == 0
40
41            app.ClculoButton.Enable = 'On';
42        end
43        if app.ZonalHarmonicsCheckBox.Value == 1
44            if app.TodosButton_2.Value == 0 ...
45                && app.J2Button.Value == 0 ...
46                && app.J3Button.Value == 0 ...
47                && app.J4Button.Value == 0 ...
48                && app.J5Button.Value == 0 ...
49                && app.J6Button.Value == 0
50
51                app.ClculoButton.Enable = 'Off';
52            else
53                app.ClculoButton.Enable = 'On';
54            end
55        end
56        if app.AtmsferaCheckBox.Value == 1
57            if app.AmAtmEditField.Value ≠ 0 && app.CDEditField.Value ≠ 0
58                app.ClculoButton.Enable = 'On';
59            else
60                app.ClculoButton.Enable = 'Off';
61            end

```

```

62         end
63         if app.SRPCheckBox.Value == 1
64             if app.CREditField.Value ≠ 0 && app.AmsRPEditField.Value ≠ 0
65                 app.ClculoButton.Enable = 'On';
66             else
67                 app.ClculoButton.Enable = 'Off';
68             end
69         end
70     end
71 end

```

Listing 22: Función en la aplicación por el *Button Sol*.

```

1
2 function SolButtonValueChanged(app, event)
3     value = app.SolButton.Value;
4     if value == 1
5         if app.ZonalHarmonicsCheckBox.Value == 0 ...
6             && app.AtmosferaCheckBox.Value == 0 ...
7             && app.SRPCheckBox.Value == 0
8
9             app.ClculoButton.Enable = 'On';
10            end
11            if app.ZonalHarmonicsCheckBox.Value == 1
12                if app.TodosButton_2.Value == 0 ...
13                    && app.J2Button.Value == 0 ...
14                    && app.J3Button.Value == 0 ...
15                    && app.J4Button.Value == 0 ...
16                    && app.J5Button.Value == 0 ...
17                    && app.J6Button.Value == 0
18
19                    app.ClculoButton.Enable = 'Off';
20                else
21                    app.ClculoButton.Enable = 'On';
22                end
23            end
24            if app.AtmosferaCheckBox.Value == 1
25                if app.AmAtmEditField.Value ≠ 0 && app.CDEditField.Value ≠ 0
26                    app.ClculoButton.Enable = 'On';
27                else
28                    app.ClculoButton.Enable = 'Off';
29                end
30            end
31            if app.SRPCheckBox.Value == 1
32                if app.CREditField.Value ≠ 0 && app.AmsRPEditField.Value ≠ 0
33                    app.ClculoButton.Enable = 'On';
34                else
35                    app.ClculoButton.Enable = 'Off';
36                end
37            end
38        end
39    else
40        if app.LunaButton.Value == 0 ...
41            && app.MercurioButton.Value == 0 ...
42            && app.VenusButton.Value == 0 ...
43            && app.TierraButton.Value == 0 ...
44            && app.MarteButton.Value == 0 ...
45            && app.JpiterButton.Value == 0 ...
46            && app.SaturnoButton.Value == 0 ...

```

```

47         && app.UranoButton.Value == 0 ...
48         && app.NeptunoButton.Value == 0 ...
49         && app.TodosButton.Value == 0
50
51         app.ClculoButton.Enable = 'Off';
52     end
53 end
54
55 end

```

Listing 23: Función en la aplicación por el *Button* Luna.

```

1
2 function LunaButtonValueChanged(app, event)
3     value = app.LunaButton.Value;
4     if value == 1
5         if app.ZonalHarmonicsCheckBox.Value == 0 ...
6             && app.AtmosferaCheckBox.Value == 0 ...
7             && app.SRPCheckBox.Value == 0
8
9             app.ClculoButton.Enable = 'On';
10        end
11        if app.ZonalHarmonicsCheckBox.Value == 1
12            if app.TodosButton_2.Value == 0 ...
13                && app.J2Button.Value == 0 ...
14                && app.J3Button.Value == 0 ...
15                && app.J4Button.Value == 0 ...
16                && app.J5Button.Value == 0 ...
17                && app.J6Button.Value == 0
18
19            app.ClculoButton.Enable = 'Off';
20        else
21            app.ClculoButton.Enable = 'On';
22        end
23    end
24    if app.AtmosferaCheckBox.Value == 1
25        if app.AmAtmEditField.Value ≠ 0 && app.CDEditField.Value ≠ 0
26            app.ClculoButton.Enable = 'On';
27        else
28            app.ClculoButton.Enable = 'Off';
29        end
30    end
31
32    if app.SRPCheckBox.Value == 1
33        if app.CREditField.Value ≠ 0 && app.AmSRPEditField.Value ≠ 0
34            app.ClculoButton.Enable = 'On';
35        else
36            app.ClculoButton.Enable = 'Off';
37        end
38    end
39    end
40    else
41        if isequal(app.FocodelarbitaDropDown.Value, 'Sol')
42            if app.TodosButton.Value == 0 ...
43                && app.MercurioButton.Value == 0 ...
44                && app.VenusButton.Value == 0 ...
45                && app.TierraButton.Value == 0 ...
46                && app.MarteButton.Value == 0 ...
47                && app.JpiterButton.Value == 0 ...

```

```

48         && app.SaturnoButton.Value == 0 ...
49         && app.UranoButton.Value == 0 ...
50         && app.NeptunoButton.Value == 0
51
52         app.ClculoButton.Enable = 'Off';
53     end
54     elseif isequal(app.FocodelarbitaDropDown.Value, 'Tierra')
55         if app.TodosButton.Value == 0 ...
56             && app.SolButton.Value == 0 ...
57             && app.MercurioButton.Value == 0 ...
58             && app.VenusButton.Value == 0 ...
59             && app.MarteButton.Value == 0 ...
60             && app.JpiterButton.Value == 0 ...
61             && app.SaturnoButton.Value == 0 ...
62             && app.UranoButton.Value == 0 ...
63             && app.NeptunoButton.Value == 0
64
65             app.ClculoButton.Enable = 'Off';
66         end
67     end
68 end
69 end

```

Listing 24: Función en la aplicación por el *Button* Mercurio.

```

1
2 function MercurioButtonValueChanged(app, event)
3     value = app.MercurioButton.Value;
4     if value == 1
5         if app.ZonalHarmonicsCheckBox.Value == 0 ...
6             && app.AtmsferaCheckBox.Value == 0 ...
7             && app.SRPCheckBox.Value == 0
8
9             app.ClculoButton.Enable = 'On';
10        end
11        if app.ZonalHarmonicsCheckBox.Value == 1
12            if app.TodosButton_2.Value == 0 ...
13                && app.J2Button.Value == 0 ...
14                && app.J3Button.Value == 0 ...
15                && app.J4Button.Value == 0 ...
16                && app.J5Button.Value == 0 ...
17                && app.J6Button.Value == 0
18
19            app.ClculoButton.Enable = 'Off';
20        else
21            app.ClculoButton.Enable = 'On';
22        end
23    end
24    if app.AtmsferaCheckBox.Value == 1
25        if app.AmAtmEditField.Value ≠ 0 && app.CDEditField.Value ≠ 0
26            app.ClculoButton.Enable = 'On';
27        else
28            app.ClculoButton.Enable = 'Off';
29        end
30    end
31    if app.SRPCheckBox.Value == 1
32        if app.CREditField.Value ≠ 0 && app.AmsRPeditField.Value ≠ 0
33            app.ClculoButton.Enable = 'On';
34        else

```

```

35         app.ClculoButton.Enable = 'Off';
36     end
37
38     end
39     else
40         if isequal(app.FocodelarbitaDropDown.Value, 'Sol')
41             if app.TodosButton.Value == 0 ...
42                 && app.LunaButton.Value == 0 ...
43                 && app.VenusButton.Value == 0 ...
44                 && app.TierraButton.Value == 0 ...
45                 && app.MarteButton.Value == 0 ...
46                 && app.JpiterButton.Value == 0 ...
47                 && app.SaturnoButton.Value == 0 ...
48                 && app.UranoButton.Value == 0 ...
49                 && app.NeptunoButton.Value == 0
50
51                 app.ClculoButton.Enable = 'Off';
52             end
53             elseif isequal(app.FocodelarbitaDropDown.Value, 'Tierra')
54                 if app.TodosButton.Value == 0 ...
55                     && app.SolButton.Value == 0 ...
56                     && app.LunaButton.Value == 0 ...
57                     && app.VenusButton.Value == 0 ...
58                     && app.MarteButton.Value == 0 ...
59                     && app.JpiterButton.Value == 0 ...
60                     && app.SaturnoButton.Value == 0 ...
61                     && app.UranoButton.Value == 0 ...
62                     && app.NeptunoButton.Value == 0
63
64                     app.ClculoButton.Enable = 'Off';
65                 end
66             end
67         end
68     end

```

Listing 25: Función en la aplicación por el *Button* Venus.

```

1
2 function VenusButtonValueChanged(app, event)
3     value = app.VenusButton.Value;
4     if value == 1
5         if app.ZonalHarmonicsCheckBox.Value == 0 ...
6             && app.AtmsferaCheckBox.Value == 0 ...
7             && app.SRPCheckBox.Value == 0
8
9             app.ClculoButton.Enable = 'On';
10            end
11            if app.ZonalHarmonicsCheckBox.Value == 1
12                if app.TodosButton_2.Value == 0 ...
13                    && app.J2Button.Value == 0 ...
14                    && app.J3Button.Value == 0 ...
15                    && app.J4Button.Value == 0 ...
16                    && app.J5Button.Value == 0 ...
17                    && app.J6Button.Value == 0
18
19                    app.ClculoButton.Enable = 'Off';
20                else
21                    app.ClculoButton.Enable = 'On';
22            end

```

```

23         end
24     if app.AtmosferaCheckBox.Value == 1
25         if app.AmAtmEditField.Value ≠ 0 && app.CDEditField.Value ≠ 0
26             app.ClculoButton.Enable = 'On';
27         else
28             app.ClculoButton.Enable = 'Off';
29         end
30     end
31     if app.SRPCheckBox.Value == 1
32         if app.CREditField.Value ≠ 0 && app.AmSRPEditField.Value ≠ 0
33             app.ClculoButton.Enable = 'On';
34         else
35             app.ClculoButton.Enable = 'Off';
36         end
37     end
38     end
39     else
40         if isequal(app.FocodelarbitaDropDown.Value, 'Sol')
41             if app.TodosButton.Value == 0 ...
42                 && app.LunaButton.Value == 0 ...
43                 && app.MercurioButton.Value == 0 ...
44                 && app.TierraButton.Value == 0 ...
45                 && app.MarteButton.Value == 0 ...
46                 && app.JpiterButton.Value == 0 ...
47                 && app.SaturnoButton.Value == 0 ...
48                 && app.UranoButton.Value == 0 ...
49                 && app.NeptunoButton.Value == 0
50
51             app.ClculoButton.Enable = 'Off';
52             end
53         elseif isequal(app.FocodelarbitaDropDown.Value, 'Tierra')
54             if app.TodosButton.Value == 0 ...
55                 && app.SolButton.Value == 0 ...
56                 && app.LunaButton.Value == 0 ...
57                 && app.MercurioButton.Value == 0 ...
58                 && app.MarteButton.Value == 0 ...
59                 && app.JpiterButton.Value == 0 ...
60                 && app.SaturnoButton.Value == 0 ...
61                 && app.UranoButton.Value == 0 ...
62                 && app.NeptunoButton.Value == 0
63
64             app.ClculoButton.Enable = 'Off';
65             end
66         end
67     end
68 end

```

Listing 26: Función en la aplicación por el *Button* Tierra.

```

1
2 function TierraButtonValueChanged(app, event)
3     value = app.TierraButton.Value;
4     if value == 1
5         if app.ZonalHarmonicsCheckBox.Value == 0 ...
6             && app.AtmosferaCheckBox.Value == 0 ...
7             && app.SRPCheckBox.Value == 0
8
9             app.ClculoButton.Enable = 'On';
10            end

```

```

11         if app.ZonalHarmonicsCheckBox.Value == 1
12             if app.TodosButton_2.Value == 0 ...
13                 && app.J2Button.Value == 0 ...
14                 && app.J3Button.Value == 0 ...
15                 && app.J4Button.Value == 0 ...
16                 && app.J5Button.Value == 0 ...
17                 && app.J6Button.Value == 0
18
19             app.ClculoButton.Enable = 'Off';
20         else
21             app.ClculoButton.Enable = 'On';
22         end
23     end
24     if app.AtmosferaCheckBox.Value == 1
25         if app.AmAtmEditField.Value ≠ 0 && app.CDEditField.Value ≠ 0
26             app.ClculoButton.Enable = 'On';
27         else
28             app.ClculoButton.Enable = 'Off';
29         end
30     end
31     if app.SRPCheckBox.Value == 1
32         if app.CREditField.Value ≠ 0 && app.AmSRPEditField.Value ≠ 0
33             app.ClculoButton.Enable = 'On';
34         else
35             app.ClculoButton.Enable = 'Off';
36         end
37     end
38     end
39     else
40         if app.LunaButton.Value == 0 ...
41             && app.MercurioButton.Value == 0 ...
42             && app.VenusButton.Value == 0 ...
43             && app.MarteButton.Value == 0 ...
44             && app.JpiterButton.Value == 0 ...
45             && app.SaturnoButton.Value == 0 ...
46             && app.UranoButton.Value == 0 ...
47             && app.NeptunoButton.Value == 0 ...
48             && app.TodosButton.Value == 0
49         app.ClculoButton.Enable = 'Off';
50     end
51     end
52     end

```

Listing 27: Función en la aplicación por el *Button* Marte.

```

1
2 function MarteButtonValueChanged(app, event)
3     value = app.MarteButton.Value;
4     if value == 1
5         if app.ZonalHarmonicsCheckBox.Value == 0 ...
6             && app.AtmosferaCheckBox.Value == 0 ...
7             && app.SRPCheckBox.Value == 0
8
9         app.ClculoButton.Enable = 'On';
10    end
11    if app.ZonalHarmonicsCheckBox.Value == 1
12        if app.TodosButton_2.Value == 0 ...
13            && app.J2Button.Value == 0 ...
14            && app.J3Button.Value == 0 ...

```

```

15         && app.J4Button.Value == 0 ...
16         && app.J5Button.Value == 0 ...
17         && app.J6Button.Value == 0
18
19         app.ClculoButton.Enable = 'Off';
20     else
21         app.ClculoButton.Enable = 'On';
22     end
23 end
24 if app.AtmosferaCheckBox.Value == 1
25     if app.AmAtmEditField.Value ≠ 0 && app.CEditField.Value ≠ 0
26         app.ClculoButton.Enable = 'On';
27     else
28         app.ClculoButton.Enable = 'Off';
29     end
30 end
31 if app.SRPCheckBox.Value == 1
32     if app.CREditField.Value ≠ 0 && app.AmSRPEditField.Value ≠ 0
33         app.ClculoButton.Enable = 'On';
34     else
35         app.ClculoButton.Enable = 'Off';
36     end
37 end
38 end
39 else
40     if isequal(app.FocodelarbitaDropDown.Value, 'Sol')
41         if app.TodosButton.Value == 0 ...
42             && app.LunaButton.Value == 0 ...
43             && app.MercurioButton.Value == 0 ...
44             && app.VenusButton.Value == 0 ...
45             && app.TierraButton.Value == 0 ...
46             && app.JpiterButton.Value == 0 ...
47             && app.SaturnoButton.Value == 0 ...
48             && app.UranoButton.Value == 0 ...
49             && app.NeptunoButton.Value == 0
50
51             app.ClculoButton.Enable = 'Off';
52         end
53     elseif isequal(app.FocodelarbitaDropDown.Value, 'Tierra')
54         if app.TodosButton.Value == 0 ...
55             && app.SolButton.Value == 0 ...
56             && app.LunaButton.Value == 0 ...
57             && app.MercurioButton.Value == 0 ...
58             && app.VenusButton.Value == 0 ...
59             && app.JpiterButton.Value == 0 ...
60             && app.SaturnoButton.Value == 0 ...
61             && app.UranoButton.Value == 0 ...
62             && app.NeptunoButton.Value == 0
63
64             app.ClculoButton.Enable = 'Off';
65         end
66     end
67 end
68 end

```

Listing 28: Función en la aplicación por el *Button* Júpiter.

```

1
2 function JpiterButtonValueChanged(app, event)

```

```
3     value = app.JpiterButton.Value;
4     if value == 1
5         if app.ZonalHarmonicsCheckBox.Value == 0 ...
6             && app.AtmsferaCheckBox.Value == 0 ...
7             && app.SRPCheckBox.Value == 0
8
9             app.ClculoButton.Enable = 'On';
10            end
11            if app.ZonalHarmonicsCheckBox.Value == 1
12                if app.TodosButton_2.Value == 0 ...
13                    && app.J2Button.Value == 0 ...
14                    && app.J3Button.Value == 0 ...
15                    && app.J4Button.Value == 0 ...
16                    && app.J5Button.Value == 0 ...
17                    && app.J6Button.Value == 0
18
19                    app.ClculoButton.Enable = 'Off';
20                else
21                    app.ClculoButton.Enable = 'On';
22                end
23            end
24            if app.AtmsferaCheckBox.Value == 1
25                if app.AmAtmEditField.Value ≠ 0 && app.CDEditField.Value ≠ 0
26                    app.ClculoButton.Enable = 'On';
27                else
28                    app.ClculoButton.Enable = 'Off';
29                end
30            end
31            if app.SRPCheckBox.Value == 1
32                if app.CREditField.Value ≠ 0 && app.AmSRPEditField.Value ≠ 0
33                    app.ClculoButton.Enable = 'On';
34                else
35                    app.ClculoButton.Enable = 'Off';
36                end
37            end
38        end
39    else
40        if isequal(app.FocodelarbitaDropDown.Value, 'Sol')
41            if app.TodosButton.Value == 0 ...
42                && app.LunaButton.Value == 0 ...
43                && app.MercurioButton.Value == 0 ...
44                && app.VenusButton.Value == 0 ...
45                && app.TierraButton.Value == 0 ...
46                && app.MarteButton.Value == 0 ...
47                && app.SaturnoButton.Value == 0 ...
48                && app.UranoButton.Value == 0 ...
49                && app.NeptunoButton.Value == 0
50
51                app.ClculoButton.Enable = 'Off';
52            end
53            elseif isequal(app.FocodelarbitaDropDown.Value, 'Tierra')
54                if app.TodosButton.Value == 0 ...
55                    && app.SolButton.Value == 0 ...
56                    && app.LunaButton.Value == 0 ...
57                    && app.MercurioButton.Value == 0 ...
58                    && app.VenusButton.Value == 0 ...
59                    && app.MarteButton.Value == 0 ...
60                    && app.SaturnoButton.Value == 0 ...
61                    && app.UranoButton.Value == 0 ...
62                    && app.NeptunoButton.Value == 0
63
```

```

64         app.ClculoButton.Enable = 'Off';
65         end
66     end
67 end
68 end

```

Listing 29: Función en la aplicación por el *Button* Saturno.

```

1
2 function SaturnoButtonValueChanged(app, event)
3     value = app.SaturnoButton.Value;
4     if value == 1
5         if app.ZonalHarmonicsCheckBox.Value == 0 ...
6             && app.AtmsferaCheckBox.Value == 0 ...
7             && app.SRPCheckBox.Value == 0
8
9             app.ClculoButton.Enable = 'On';
10            end
11            if app.ZonalHarmonicsCheckBox.Value == 1
12                if app.TodosButton_2.Value == 0 ...
13                    && app.J2Button.Value == 0 ...
14                    && app.J3Button.Value == 0 ...
15                    && app.J4Button.Value == 0 ...
16                    && app.J5Button.Value == 0 ...
17                    && app.J6Button.Value == 0
18
19                    app.ClculoButton.Enable = 'Off';
20                else
21                    app.ClculoButton.Enable = 'On';
22                end
23            end
24            if app.AtmsferaCheckBox.Value == 1
25                if app.AmAtmEditField.Value ≠ 0 && app.CDEditField.Value ≠ 0
26                    app.ClculoButton.Enable = 'On';
27                else
28                    app.ClculoButton.Enable = 'Off';
29                end
30            end
31            if app.SRPCheckBox.Value == 1
32                if app.CREditField.Value ≠ 0 && app.AmSRPEditField.Value ≠ 0
33                    app.ClculoButton.Enable = 'On';
34                else
35                    app.ClculoButton.Enable = 'Off';
36                end
37            end
38        end
39    else
40        if isequal(app.FocodelarbitaDropDown.Value, 'Sol')
41            if app.TodosButton.Value == 0 ...
42                && app.LunaButton.Value == 0 ...
43                && app.MercurioButton.Value == 0 ...
44                && app.VenusButton.Value == 0 ...
45                && app.TierraButton.Value == 0 ...
46                && app.MarteButton.Value == 0 ...
47                && app.JpiterButton.Value == 0 ...
48                && app.UranoButton.Value == 0 ...
49                && app.NeptunoButton.Value == 0
50
51            app.ClculoButton.Enable = 'Off';

```

```

52         end
53     elseif isequal(app.FocodelarbitaDropDown.Value, 'Tierra')
54         if app.TodosButton.Value == 0 ...
55             && app.SolButton.Value == 0 ...
56             && app.LunaButton.Value == 0 ...
57             && app.MercurioButton.Value == 0 ...
58             && app.VenusButton.Value == 0 ...
59             && app.MarteButton.Value == 0 ...
60             && app.JpiterButton.Value == 0 ...
61             && app.UranoButton.Value == 0 ...
62             && app.NeptunoButton.Value == 0
63
64             app.ClculoButton.Enable = 'Off';
65         end
66     end
67 end
68 end

```

Listing 30: Función en la aplicación por el *Button* Urano.

```

1
2 function UranoButtonValueChanged(app, event)
3     value = app.UranoButton.Value;
4     if value == 1
5         if app.ZonalHarmonicsCheckBox.Value == 0 ...
6             && app.AtmsferaCheckBox.Value == 0 ...
7             && app.SRPCheckBox.Value == 0
8
9             app.ClculoButton.Enable = 'On';
10        end
11        if app.ZonalHarmonicsCheckBox.Value == 1
12            if app.TodosButton_2.Value == 0 ...
13                && app.J2Button.Value == 0 ...
14                && app.J3Button.Value == 0 ...
15                && app.J4Button.Value == 0 ...
16                && app.J5Button.Value == 0 ...
17                && app.J6Button.Value == 0
18
19                app.ClculoButton.Enable = 'Off';
20            else
21                app.ClculoButton.Enable = 'On';
22            end
23        end
24        if app.AtmsferaCheckBox.Value == 1
25            if app.AmAtmEditField.Value ≠ 0 && app.CDEditField.Value ≠ 0
26                app.ClculoButton.Enable = 'On';
27            else
28                app.ClculoButton.Enable = 'Off';
29            end
30        end
31        if app.SRPCheckBox.Value == 1
32            if app.CREditField.Value ≠ 0 && app.AmSRPEditField.Value ≠ 0
33                app.ClculoButton.Enable = 'On';
34            else
35                app.ClculoButton.Enable = 'Off';
36            end
37        end
38    end
39 else

```

```

40         if isequal(app.FocodelarbitaDropDown.Value, 'Sol')
41             if app.TodosButton.Value == 0 ...
42                 && app.LunaButton.Value == 0 ...
43                 && app.MercurioButton.Value == 0 ...
44                 && app.VenusButton.Value == 0 ...
45                 && app.TierraButton.Value == 0 ...
46                 && app.MarteButton.Value == 0 ...
47                 && app.JpiterButton.Value == 0 ...
48                 && app.SaturnoButton.Value == 0 ...
49                 && app.NeptunoButton.Value == 0
50
51             app.ClculoButton.Enable = 'Off';
52         end
53     elseif isequal(app.FocodelarbitaDropDown.Value, 'Tierra')
54         if app.TodosButton.Value == 0 ...
55             && app.SolButton.Value == 0 ...
56             && app.LunaButton.Value == 0 ...
57             && app.MercurioButton.Value == 0 ...
58             && app.VenusButton.Value == 0 ...
59             && app.MarteButton.Value == 0 ...
60             && app.JpiterButton.Value == 0 ...
61             && app.SaturnoButton.Value == 0 ...
62             && app.NeptunoButton.Value == 0
63
64         app.ClculoButton.Enable = 'Off';
65     end
66 end
67 end
68 end

```

Listing 31: Función en la aplicación por el *Button* Neptuno.

```

1
2 function NeptunoButtonValueChanged(app, event)
3     value = app.NeptunoButton.Value;
4     if value == 1
5         if app.ZonalHarmonicsCheckBox.Value == 0 ...
6             && app.AtmsferaCheckBox.Value == 0 ...
7             && app.SRPCheckBox.Value == 0
8
9         app.ClculoButton.Enable = 'On';
10    end
11    if app.ZonalHarmonicsCheckBox.Value == 1
12        if app.TodosButton_2.Value == 0 ...
13            && app.J2Button.Value == 0 ...
14            && app.J3Button.Value == 0 ...
15            && app.J4Button.Value == 0 ...
16            && app.J5Button.Value == 0 ...
17            && app.J6Button.Value == 0
18
19        app.ClculoButton.Enable = 'Off';
20    else
21        app.ClculoButton.Enable = 'On';
22    end
23 end
24 if app.AtmsferaCheckBox.Value == 1
25     if app.AmAtmEditField.Value ≠ 0 && app.CDEditField.Value ≠ 0
26         app.ClculoButton.Enable = 'On';
27     else

```

```
28         app.ClculoButton.Enable = 'Off';
29     end
30 end
31 if app.SRPCheckBox.Value == 1
32     if app.CREditField.Value ≠ 0 && app.AmSRPEditField.Value ≠ 0
33         app.ClculoButton.Enable = 'On';
34     else
35         app.ClculoButton.Enable = 'Off';
36     end
37
38     end
39 else
40     if isequal(app.FocodelarbitaDropDown.Value, 'Sol')
41         if app.TodosButton.Value == 0 ...
42             && app.LunaButton.Value == 0 ...
43             && app.MercurioButton.Value == 0 ...
44             && app.VenusButton.Value == 0 ...
45             && app.TierraButton.Value == 0 ...
46             && app.MarteButton.Value == 0 ...
47             && app.JpiterButton.Value == 0 ...
48             && app.SaturnoButton.Value == 0 ...
49             && app.UranoButton.Value == 0
50
51         app.ClculoButton.Enable = 'Off';
52     end
53     elseif isequal(app.FocodelarbitaDropDown.Value, 'Tierra')
54         if app.TodosButton.Value == 0 ...
55             && app.SolButton.Value == 0 ...
56             && app.LunaButton.Value == 0 ...
57             && app.MercurioButton.Value == 0 ...
58             && app.VenusButton.Value == 0 ...
59             && app.MarteButton.Value == 0 ...
60             && app.JpiterButton.Value == 0 ...
61             && app.SaturnoButton.Value == 0 ...
62             && app.UranoButton.Value == 0
63
64         app.ClculoButton.Enable = 'Off';
65     end
66 end
67 end
68 end
```

B Código de MATLAB

Listing 32: Función en la aplicación por el *Button* Cálculo.

```

1
2 function ClculoButtonPushed(app, event)
3
4     app.VectordeestadofinalLabel.Enable = 'On';
5
6     app.PosicinFinalLabel.Enable = 'On';
7     app.VelocidadFinalLabel.Enable = 'On';
8
9     app.xEditField_2Label.Enable = 'On';
10    app.yEditField_2Label.Enable = 'On';
11    app.zEditField_2Label.Enable = 'On';
12
13    app.xfEditField.Enable= 'On';
14    app.yfEditField.Enable = 'On';
15    app.zfEditField.Enable = 'On';
16
17    app.VxEditField_2Label.Enable = 'On';
18    app.VyEditField_2Label.Enable = 'On';
19    app.VzEditField_2Label.Enable = 'On';
20
21    app.VxfEditField.Enable = 'On';
22    app.VyfEditField.Enable = 'On';
23    app.VzfEditField.Enable = 'On';
24
25    [PosFinal, VelFinal] = Integrador(app);
26
27    app.xfEditField.Value = PosFinal(1);
28    app.yfEditField.Value = PosFinal(2);
29    app.zfEditField.Value = PosFinal(3);
30
31    app.VxfEditField.Value = VelFinal(1);
32    app.VyfEditField.Value = VelFinal(2);
33    app.VzfEditField.Value = VelFinal(3);
34
35    end

```

Listing 33: Función *Integrador* de la aplicación.

```

1
2 function [PosFinal, VelFinal] = Integrador(app)
3
4     % Vector inicial
5     Pos_X0 = app.x0EditField.Value;
6     Pos_Y0 = app.y0EditField.Value;
7     Pos_Z0 = app.z0EditField.Value;
8
9     Vel_X0 = app.Vx0EditField.Value;
10    Vel_Y0 = app.Vy0EditField.Value;
11    Vel_Z0 = app.Vz0EditField.Value;
12
13    u0 = [Pos_X0;Pos_Y0;Pos_Z0;Vel_X0;Vel_Y0;Vel_Z0];
14
15    [date_0, date_f] = tiempo(app);

```

```

16
17     if isequal(app.FocodelarbitaDropDown.Value, 'Sol')
18         letra = 'H';
19     elseif isequal(app.FocodelarbitaDropDown.Value, 'Tierra')
20         letra = 'G';
21     end
22
23     if app.NoButton.Value == 1
24
25         [value] = ArchivosPlanetas(app, date_0, date_f, letra)
26
27         % Tiempo de integraci n
28         dt = etime(date_f, date_0);
29
30         t0 = 0; % Instante inicial
31         tf = t0+dt; % Instante final
32         diff_t = 60; % Segundos entre dos iteraciones
33         int = (tf/diff_t)+1; % N mero de intervalos
34         tspan = linspace(t0,tf,int);
35
36         % Opciones de integraci n
37         if isequal(app.FocodelarbitaDropDown.Value, 'Sol')
38             options=odeset('AbsTol',1e-14,'RelTol',1e-14);
39         elseif isequal(app.FocodelarbitaDropDown.Value, 'Tierra')
40             options=odeset('AbsTol',1e-8,'RelTol',1e-8);
41         end
42
43
44         integracion = ["", "", "", ""];
45         DatosSat = [0,0,0];
46         Js = [0,0,0,0,0];
47
48         if app.TwoBodyCheckBox.Value == 1
49             integracion(1) = "Two Body";
50         else
51             if app.PlanetasCheckBox.Value == 1
52                 integracion(1) = "Planetas";
53             end
54             if app.SRPCheckBox.Value == 1
55                 DatosSat(1) = app.CREditField.Value;
56                 DatosSat(2) = app.AmAtmEditField.Value;
57                 integracion(2) = "SRP";
58             end
59             if app.ZonalHarmonicsCheckBox.Value == 1
60                 integracion(3) = "Harmonics";
61
62                 if app.J2Button.Value == 1
63                     Js(1) = 1082.63e-6;
64                 end
65                 if app.J3Button.Value == 1
66                     Js(2) = -2.52e-6;
67                 end
68                 if app.J4Button.Value == 1
69                     Js(3) = -1.61e-6;
70                 end
71                 if app.J5Button.Value == 1
72                     Js(4) = -0.15e-6;
73                 end
74                 if app.J6Button.Value == 1
75                     Js(5) = 0.57e-6;
76                 end
77             end

```

```

77
78         end
79         if app.AtmosferaCheckBox.Value == 1
80             integracion(4) = "Atmosfera";
81             DatosSat(2) = app.AmAtmEditField.Value;
82             DatosSat(3) = app.CDEditField.Value;
83         end
84
85     end
86
87     [u] = Main(letra,value,integracion,DatosSat,Js,tspan,u0,options);
88
89     Pos = u(:,1:3);
90     Vel = u(:,4:6);
91
92     else
93
94         % Se realizan los distintos intervalos
95
96         for ii = 1:app.NmerodeImpusosTotalesSpinner.Value
97
98             file = ['Impulso-', num2str(ii), '.txt'];
99             data = importdata(file);
100
101             date(ii,:) = data(1,:);
102             velo(ii,:) = data(2,:);
103
104             if ii == 1
105                 dt = etime(date(ii,:),date_0);
106
107                 [value] = ArchivosPlanetas(app,date_0, date(ii,:),letra)
108
109             else
110                 dt = etime(date(ii,:),date(ii-1,:))
111
112                 [value] = ArchivosPlanetas(app,date(ii-1,:),date(ii,:),letra)
113
114             end
115
116             t0 = 0; % Instante inicial
117             tf = t0+dt; % Instante final
118             diff_t = 60; % Segundos entre dos iteraciones
119             int = (tf/diff_t)+1; % N mero de intervalos
120
121             tspan = linspace(t0,tf,int);
122
123             % Opciones de integraci n
124             if isequal(app.FocodelarbitaDropDown.Value,'Sol')
125                 options=odeset('AbsTol',1e-14,'RelTol',1e-14);
126             elseif isequal(app.FocodelarbitaDropDown.Value,'Tierra')
127                 options=odeset('AbsTol',1e-8,'RelTol',1e-8);
128             end
129
130             integracion = ["", "", "", ""];
131             DatosSat = [0,0,0];
132             Js = [0,0,0,0,0];
133
134             if app.TwoBodyCheckBox.Value == 1
135                 integracion(1) = "Two Body";
136             else
137                 if app.PlanetasCheckBox.Value == 1

```

```

138         integracion(1) = "Planetas";
139     end
140     if app.SRPCheckBox.Value == 1
141         DatosSat(1) = app.CREditField.Value;
142         DatosSat(2) = app.AmAtmEditField.Value;
143         integracion(2) = "SRP";
144     end
145     if app.ZonalHarmonicsCheckBox.Value == 1
146         integracion(3) = "Harmonics";
147
148         if app.J2Button.Value == 1
149             Js(1) = 1082.63e-6;
150         end
151         if app.J3Button.Value == 1
152             Js(2) = -2.52e-6;
153         end
154         if app.J4Button.Value == 1
155             Js(3) = -1.61e-6;
156         end
157         if app.J5Button.Value == 1
158             Js(4) = -0.15e-6;
159         end
160         if app.J6Button.Value == 1
161             Js(5) = 0.57e-6;
162         end
163     end
164     end
165     if app.AtmosferaCheckBox.Value == 1
166         integracion(4) = "Atmosfera";
167         DatosSat(2) = app.AmAtmEditField.Value;
168         DatosSat(3) = app.CDEditField.Value;
169     end
170
171 end
172
173 [u] = Main(letra,value,integracion,DatosSat,Js,tspan,u0,options);
174
175 [u0] = [u(end,1);u(end,2);u(end,3);...
176         velo(ii,1);velo(ii,2);velo(ii,3)];
177
178 if ii == 1
179     Pos = u(1:end-1,1:3);
180     Vel = u(1:end-1,4:6);
181
182 else
183     Pos = [Pos;u(1:end-1,1:3)];
184     Vel = [Vel;u(1:end-1,4:6)];
185
186 end
187
188
189 end
190
191 % Para el ltimo tramo:
192
193 dt = etime(date_f,date(ii,:))
194
195 [value] = ArchivosPlanetas(app,date(ii,:),date_f,letra)
196
197 t0 = 0; % Instante inicial
198 tf = t0+dt; % Instante final

```

```

199     diff_t = 60; % Segundos entre dos iteraciones
200     int = (tf/diff_t)+1; % N mero de intervalos
201
202     tspan = linspace(t0,tf,int);
203
204     % Opciones de integraci n
205     if isequal(app.FocodelarbitaDropDown.Value,'Sol')
206         options=odeset('AbsTol',1e-14,'RelTol',1e-14);
207     elseif isequal(app.FocodelarbitaDropDown.Value,'Tierra')
208         options=odeset('AbsTol',1e-8,'RelTol',1e-8);
209     end
210
211     integracion = ["", "", "", ""];
212     DatosSat = [0,0,0];
213     Js = [0,0,0,0,0];
214
215     if app.TwoBodyCheckBox.Value == 1
216         integracion(1) = "Two Body";
217     else
218         if app.PlanetasCheckBox.Value == 1
219             integracion(1) = "Planetas";
220         end
221         if app.SRPCheckBox.Value == 1
222             DatosSat(1) = app.CREditField.Value;
223             DatosSat(2) = app.AmAtmEditField.Value;
224             integracion(2) = "SRP";
225         end
226         if app.ZonalHarmonicsCheckBox.Value == 1
227             integracion(3) = "Harmonics";
228
229             if app.J2Button.Value == 1
230                 Js(1) = 1082.63e-6;
231             end
232             if app.J3Button.Value == 1
233                 Js(2) = -2.52e-6;
234             end
235             if app.J4Button.Value == 1
236                 Js(3) = -1.61e-6;
237             end
238             if app.J5Button.Value == 1
239                 Js(4) = -0.15e-6;
240             end
241             if app.J6Button.Value == 1
242                 Js(5) = 0.57e-6;
243             end
244
245         end
246         if app.AtmsferaCheckBox.Value == 1
247             integracion(4) = "Atmosfera";
248             DatosSat(2) = app.AmAtmEditField.Value;
249             DatosSat(3) = app.CDEditField.Value;
250         end
251     end
252
253     [u] = Main(letra,value,integracion,DatosSat,Js,tspan,u0,options);
254
255     Pos = [Pos;u(:,1:3)];
256     Vel = [Vel;u(:,4:6)];
257
258
259

```

```

260     end
261
262     save('Posición.txt', 'Pos', '-ascii', '-double');
263     save('Velocidad.txt', 'Vel', '-ascii', '-double');
264
265     PosFinal = u(end,1:3);
266     VelFinal = u(end,4:6);
267
268     if isequal(app.FocodelarbitaDropDown.Value, 'Tierra')
269
270         u = [Pos(:,1), Pos(:,2), Pos(:,3), Vel(:,1), Vel(:,2), Vel(:,3)];
271
272         dt = etime(date_f, date_0)
273
274         t0 = 0; % Instante inicial
275         tf = t0+dt; % Instante final
276         diff_t = 60; % Segundos entre dos iteraciones
277         int = (tf/diff_t)+1; % Número de intervalos
278         t = linspace(t0,tf,int);
279
280         plotter(t,u)
281     end
282
283 end
284
285 end

```

Listing 34: Función tiempo de la aplicación.

```

1
2 function [date_0, date_f] = tiempo(app)
3
4     % Fecha inicial
5     year_0 = app.Ao0EditField.Value;
6     month_0 = app.Mes0EditField.Value;
7     day_0 = app.Da0EditField.Value;
8     hour_0 = app.Hora0EditField.Value;
9     min_0 = app.Minuto0EditField.Value;
10    sec_0 = app.Segundo0EditField.Value;
11
12    % Fecha final
13    year_f = app.AofEditField.Value;
14    month_f = app.MesfEditField.Value;
15    day_f = app.DafEditField.Value;
16    hour_f = app.HorafEditField.Value;
17    min_f = app.MinutofEditField.Value;
18    sec_f = app.SegundofEditField.Value;
19
20    % Creación de las fechas
21    date_0 = [year_0, month_0, day_0, hour_0, min_0, sec_0]; % Fecha inicial
22    date_f = [year_f, month_f, day_f, hour_f, min_f, sec_f]; % Fecha final
23
24
25    end

```

Listing 35: Función ArchivosPlanetas de la aplicación.

1

```

2 function [value] = ArchivosPlanetas(app,date_0,date_f,letra)
3
4     % Leer todos los archivos de los planetas
5
6     if exist('Mercurio_Ephemerides.txt') == 2
7         delete('Mercurio_Ephemerides.txt');
8     end
9     if exist('Venus_Ephemerides.txt') == 2
10        delete('Venus_Ephemerides.txt');
11    end
12    if exist('Tierra_Ephemerides.txt') == 2
13        delete('Tierra_Ephemerides.txt');
14    end
15    if exist('Marte_Ephemerides.txt') == 2
16        delete('Marte_Ephemerides.txt');
17    end
18    if exist('Jupiter_Ephemerides.txt') == 2
19        delete('Jupiter_Ephemerides.txt');
20    end
21    if exist('Saturno_Ephemerides.txt') == 2
22        delete('Saturno_Ephemerides.txt');
23    end
24    if exist('Urano_Ephemerides.txt') == 2
25        delete('Urano_Ephemerides.txt');
26    end
27    if exist('Neptuno_Ephemerides.txt') == 2
28        delete('Neptuno_Ephemerides.txt');
29    end
30    if exist('Sol_Ephemerides.txt') == 2
31        delete('Sol_Ephemerides.txt');
32    end
33    if exist('Luna_Ephemerides.txt') == 2
34        delete('Luna_Ephemerides.txt');
35    end
36
37    value = '0';
38
39    if (app.SRPCheckBox.Value == 1) || (app.LunaButton.Value == 1)
40        planeta = 'Luna';
41        [Luna] = lectura(date_0,date_f,planeta,letra);
42        planeta = {planeta};
43        if isequal(value,'0')
44            value = planeta;
45        else
46            value = [value,planeta];
47        end
48    end
49
50    if (app.MercurioButton.Value == 1)
51        planeta = 'Mercurio';
52        [Mercurio] = lectura(date_0,date_f,planeta,letra);
53        planeta = {planeta};
54        if isequal(value,'0')
55            value = planeta;
56        else
57            value = [value,planeta];
58        end
59    end
60    if (app.SRPCheckBox.Value == 1) || (app.VenusButton.Value == 1)
61        planeta = 'Venus';
62        [Venus] = lectura(date_0,date_f,planeta,letra);

```

```
63     planeta = {planeta};
64     if isequal(value,'0')
65         value = planeta;
66     else
67         value = [value,planeta];
68     end
69 end
70
71 if isequal(app.FocodelarbitaDropDown.Value,'Sol')
72     if (app.SRPCheckBox.Value == 1) || (app.TierraButton.Value == 1)
73         planeta = 'Tierra';
74         [Tierra] = lectura(date_0,date_f,planeta,letra);
75         planeta = {planeta};
76         if isequal(value,'0')
77             value = planeta;
78         else
79             value = [value,planeta];
80         end
81     end
82 elseif isequal(app.FocodelarbitaDropDown.Value,'Tierra')
83     if app.SolButton.Value == 1 || app.SRPCheckBox.Value == 1
84         planeta = 'Sol';
85         [Sol] = lectura(date_0,date_f,planeta,letra);
86         planeta = {planeta};
87         if(app.SolButton.Value == 1)
88             if isequal(value,'0')
89                 value = planeta;
90             else
91                 value = [value,planeta];
92             end
93         end
94     end
95 end
96 if (app.MarteButton.Value == 1)
97     planeta = 'Marte';
98     [Marte] = lectura(date_0,date_f,planeta,letra);
99     planeta = {planeta};
100    if isequal(value,'0')
101        value = planeta;
102    else
103        value = [value,planeta];
104    end
105 end
106 if (app.JpiterButton.Value == 1)
107     planeta = 'J piter';
108     [Jupiter] = lectura(date_0,date_f,planeta,letra);
109     planeta = {planeta};
110     if isequal(value,'0')
111         value = planeta;
112     else
113         value = [value,planeta];
114     end
115 end
116 if (app.SaturnoButton.Value == 1)
117     planeta = 'Saturno';
118     [Saturno] = lectura(date_0,date_f,planeta,letra);
119     planeta = {planeta};
120     if isequal(value,'0')
121         value = planeta;
122     else
123         value = [value,planeta];
```

```

124         end
125     end
126     if (app.UranoButton.Value == 1)
127         planeta = 'Urano';
128         [Urano] = lectura(date_0,date_f,planeta,letra);
129         planeta = {planeta};
130         if isequal(value,'0')
131             value = planeta;
132         else
133             value = [value,planeta];
134         end
135     end
136     if (app.NeptunoButton.Value == 1)
137         planeta = 'Neptuno';
138         [Neptuno] = lectura(date_0,date_f,planeta,letra);
139         planeta = {planeta};
140         if isequal(value,'0')
141             value = planeta;
142         else
143             value = [value,planeta];
144         end
145     end
146 end
147

```

Listing 36: Función lectura de MATLAB.

```

1
2 function Efemerides = lectura(date_0,date_f,planeta,letra)
3
4 % Eva Terradillos
5
6 % Esta funci n obtiene las efemerides del planeta deseado, leyendo los
7 % archivos txt de cada uno y seleccionando los valores que corresponden
8 % al intervalo de tiempo estudiado.
9
10 % Julian date inicial
11
12 JD0_J2000_0 = 367*date_0(1)-fix((7*(date_0(1)+fix((date_0(2)+9)/12)))/4)+...
13     fix(275*date_0(2)/9)+date_0(3)+1721013.5;
14
15 JD_J2000_0 = JD0_J2000_0+(((date_0(6)/60)+date_0(5))/60)+date_0(4)/24;
16
17 % Julian date final
18
19 JD0_J2000_f = 367*date_f(1)-fix((7*(date_f(1)+fix((date_f(2)+9)/12)))/4)+...
20     fix(275*date_f(2)/9)+date_f(3)+1721013.5;
21
22 JD_J2000_f = JD0_J2000_f+(((date_f(6)/60)+date_f(5))/60)+date_f(4)/24;
23
24 %% Lectura TXT con fechas
25
26 % Archivos en los que buscar
27
28 % Valor inicial
29
30 file_0 = [num2str(date_0(1)), '-', num2str(date_0(2)), ' ', planeta, ...
31     letra, '.txt'];
32

```

```

33 delimiterIn = ',';
34 p_0 = importdata(file_0,delimiterIn);
35 JD_0_ = p_0.textdata(:,1);
36
37 numericIds = cellfun(@str2num, JD_0_, 'un', 0);
38 emptyIdx = cellfun(@isempty, numericIds);
39 JD_0_(emptyIdx) = [];
40
41 for ii = 1:length(JD_0_)
42     JD_0(ii) = str2num(convertCharsToStrings(cell2mat(JD_0_(ii))));
43 end
44
45 [minimo,pos_0] = min(abs(JD_J2000_0-JD_0));
46
47 % Valor final
48
49 file_f = [num2str(date_f(1)), '-', num2str(date_f(2)), ' ', planeta, ...
50     letra, '.txt'];
51
52
53 p_f = importdata(file_f,delimiterIn);
54 JD_f_ = p_f.textdata(:,1);
55
56 numericIds = cellfun(@str2num, JD_f_, 'un', 0);
57 emptyIdx = cellfun(@isempty, numericIds);
58 JD_f_(emptyIdx) = [];
59
60 for ii = 1:length(JD_f_)
61     JD_f(ii) = str2num(convertCharsToStrings(cell2mat(JD_f_(ii))));
62 end
63
64 [minimo,pos_f] = min(abs(JD_J2000_f-JD_f));
65
66
67 if isequal(file_0,file_f)
68     Efemerides = p_0.data(pos_0:pos_f,:);
69 else
70     Efemerides = p_0.data(pos_0:end,:);
71     dif_year = date_f(1) - date_0(1);
72
73     if dif_year == 0 % Si diferencia de a o es = 0
74
75         for ii = (date_0(2) + 1): (date_f(2) - 1)
76
77             file_ii = [num2str(date_0(1)), '-', num2str(ii), ' ', planeta, ...
78                 letra, '.txt'];
79             p_ii = importdata(file_ii,delimiterIn);
80             Efemerides = [Efemerides;p_ii.data(1:end,:)];
81
82         end
83
84     elseif dif_year == 1
85
86         % A o inicial
87         for ii = (date_0(2) + 1):12
88
89             file_ii = [num2str(date_0(1)), '-', num2str(ii), ' ', planeta, ...
90                 letra, '.txt'];
91             p_ii = importdata(file_ii,delimiterIn);
92             Efemerides = [Efemerides;p_ii.data(2:end,:)];
93

```

```

94
95     end
96
97     % A o final
98
99     for ii = 1:(date_f(2) - 1)
100
101         file_ii = [num2str(date_f(1)), '-', num2str(ii), ' ', planeta, ...
102                 letra, '.txt'];
103         p_ii = importdata(file_ii,delimiterIn);
104         Efemerides = [Efemerides;p_ii.data(2:end,:)];
105
106     end
107 else
108     % A o inicial
109     for ii = (date_0(2) + 1):12
110
111         file_ii = [num2str(date_0(1)), '-', num2str(ii), ' ', planeta, ...
112                 letra, '.txt'];
113         p_ii = importdata(file_ii,delimiterIn);
114         Efemerides = [Efemerides;p_ii.data(2:end,:)];
115
116     end
117
118     % A os en medio
119     for jj = 1:(dif_year - 1)
120         for ii = 1:12
121
122             file_ii = [num2str(date_0(1) + jj), '-', num2str(ii), ' ', ...
123                     planeta, letra, '.txt'];
124             p_ii = importdata(file_ii,delimiterIn);
125             Efemerides= [Efemerides;p_ii.data(2:end,:)];
126
127         end
128     end
129
130     % A o final
131     for ii = 1:(date_f(2) - 1)
132
133         file_ii = [num2str(date_f(1)), '-', num2str(ii), ' ', planeta, ...
134                 letra, '.txt'];
135         p_ii = importdata(file_ii,delimiterIn);
136         Efemerides = [Efemerides;p_ii.data(2:end,:)];
137
138     end
139
140 end
141 Efemerides = [Efemerides;p_f.data(2:pos_f,:)];
142 end
143
144 if isequal(planeta,'Mercurio')
145     save('Mercurio_Ephemerides.txt','Efemerides','-ascii','-double');
146 elseif isequal(planeta,'Venus')
147     save('Venus_Ephemerides.txt','Efemerides','-ascii','-double');
148 elseif isequal(planeta,'Tierra')
149     save('Tierra_Ephemerides.txt','Efemerides','-ascii','-double');
150 elseif isequal(planeta,'Marte')
151     save('Marte_Ephemerides.txt','Efemerides','-ascii','-double');
152 elseif isequal(planeta,'J piter')
153     save('Jupiter_Ephemerides.txt','Efemerides','-ascii','-double');
154 elseif isequal(planeta,'Saturno')

```

```

155     save('Saturno_Ephemerides.txt','Efemerides','-ascii','-double');
156 elseif isequal(planeta,'Urano')
157     save('Urano_Ephemerides.txt','Efemerides','-ascii','-double');
158 elseif isequal(planeta,'Neptuno')
159     save('Neptuno_Ephemerides.txt','Efemerides','-ascii','-double');
160 elseif isequal(planeta,'Sol')
161     save('Sol_Ephemerides.txt','Efemerides','-ascii','-double');
162 elseif isequal(planeta,'Luna')
163     save('Luna_Ephemerides.txt','Efemerides','-ascii','-double');
164 end
165
166 end

```

Listing 37: Función Main de MATLAB.

```

1
2 function [u] = Main(letra,value,integracion,DatosSat,Js,tspan,u0,options)
3
4 global mu R m
5 global file p Efemerides
6 global diff_t
7 global values
8 global centro
9 global cR Am cD
10 global J2 J3 J4 J5 J6
11
12 values = value;
13 centro = letra;
14 cR      = DatosSat(1);
15 Am      = DatosSat(2);
16 cD      = DatosSat(3);
17 J2      = Js(1);
18 J3      = Js(2);
19 J4      = Js(3);
20 J5      = Js(4);
21 J6      = Js(5);
22
23 [mu,R,m] = DatosPlanetas();
24
25 diff_t = 60; %Segundos entre intervalos
26
27 if isequal(centro,'H')
28     mu.Foco = mu.Sol;
29 elseif isequal(centro,'G')
30     mu.Foco = mu.T;
31 end
32
33 % -----
34 % Opciones
35 % -----
36 % 1. Two Body
37 % -----
38 % 1. Planetas
39 % 2. SRP
40 % 3. Harmonics
41 % 4. Atmosfera
42 % -----
43
44 if isequal(integracion(1),"Two Body")

```

```

45     [t,u]=ode45(@Integracion2body,tspan,u0,options);
46 elseif isequal(integracion(1),"Planetas")
47     [Efemerides] = EfemeridesPlanetas();
48     if isequal(integracion(2),"SRP")
49         if isequal(integracion(3),"Harmonics")
50             if isequal(integracion(4),"Atmosfera")
51                 [t,u]=ode45(@IntegracionPlanSRPHarmAtm,tspan,u0,options);
52             else
53                 [t,u]=ode45(@IntegracionPlanSRPHarm,tspan,u0,options);
54             end
55         else
56             if isequal(integracion(4),"Atmosfera")
57                 [t,u]=ode45(@IntegracionPlanSRPATm,tspan,u0,options);
58             else
59                 [t,u]=ode45(@IntegracionPlanSRP,tspan,u0,options);
60             end
61         end
62     else
63         if isequal(integracion(3),"Harmonics")
64             if isequal(integracion(4),"Atmosfera")
65                 [t,u]=ode45(@IntegracionPlanHarmAtm,tspan,u0,options);
66             else
67                 [t,u]=ode45(@IntegracionPlanHarm,tspan,u0,options);
68             end
69         else
70             if isequal(integracion(4),"Atmosfera")
71                 [t,u]=ode45(@IntegracionPlanAtm,tspan,u0,options);
72             else
73                 [t,u]=ode45(@IntegracionPlan,tspan,u0,options);
74             end
75         end
76     end
77 end
78
79 elseif isequal(integracion(1), "")
80     if isequal(integracion(2),"SRP")
81         [Efemerides] = EfemeridesPlanetas();
82         if isequal(integracion(3),"Harmonics")
83             if isequal(integracion(4),"Atmosfera")
84                 [t,u]=ode45(@IntegracionSRPHarmAtm,tspan,u0,options);
85             else
86                 [t,u]=ode45(@IntegracionSRPHarm,tspan,u0,options);
87             end
88         else
89             if isequal(integracion(4),"Atmosfera")
90                 [t,u]=ode45(@IntegracionSRPATm,tspan,u0,options);
91             else
92                 [t,u]=ode45(@IntegracionSRP,tspan,u0,options);
93             end
94         end
95     else
96         if isequal(integracion(3),"Harmonics")
97             if isequal(integracion(4),"Atmosfera")
98                 [t,u]=ode45(@IntegracionHarmAtm,tspan,u0,options);
99             else
100                [t,u]=ode45(@IntegracionHarm,tspan,u0,options);
101            end
102        else
103            if isequal(integracion(4),"Atmosfera")
104                [t,u]=ode45(@IntegracionAtm,tspan,u0,options);
105            end

```

```

106         end
107
108     end
109
110 end
111
112 end

```

Listing 38: Función DatosPlanetas de MATLAB.

```

1
2 function [mu,R,m] = DatosPlanetas()
3
4 % Eva Terradillos Est vez
5
6 % Esta funci n guarda disitntas caractersticas de los planetas, el Sol y
7 % la Luna, para que puedan ser utilizadas m s adelante. Se guardan como
8 % estructuras dependiendo del tipo de dato que sea.
9
10 global mu R m
11
12 % Parmetros gravitacionales (mu)
13
14 mu.Sol = 1.32712440018e+11; % [km^3/s^2] Sol heliocntrica
15 %mu.Sol = 1.328910432305248e+11; % [km^3/s^2] Sol Baricentro
16 mu.Luna = 4.902800066e+3; % [km^3/s^2] Luna
17 mu.Me = 2.203195793207798e+04; % [km^3/s^2] Mercurio
18 mu.V = 3.248599099127831e+05; % [km^3/s^2] Venus
19 mu.T = 3.986020525817198e+05; % [km^3/s^2] Tierra
20 mu.Ma = 4.282854737033189e+04; % [km^3/s^2] Marte
21 mu.J = 1.266870458526067e+08; % [km^3/s^2] J piter
22 mu.S = 3.793136004141563e+07; % [km^3/s^2] Saturno
23 mu.U = 5.793974827652425e+06; % [km^3/s^2] Urano
24 mu.N = 6.835127697636819e+06; % [km^3/s^2] Neptuno
25
26 % Radios (R)
27
28 R.Sol = 1392000; % [km] Sol
29 R.T = 6378; % [km] Tierra
30 R.L = 1737.1; % [km] Luna
31 R.V = 6051.8; % [km] Venus
32
33 % Masas (m)
34
35 m.Sol = 1.989e30; % Masa del Sol [kg]
36 m.T = 5.974e24; % Masa de la Tierra [kg]
37 m.M = 641.9e21; % Masa de Marte [kg]
38
39 end

```

Listing 39: Función Integracion2body de MATLAB.

```

1
2 function du=Integracion2body(t,u)
3
4
5 global mu
6

```

```

7 x=u(1);
8 y=u(2);
9 z=u(3);
10 vx=u(4);
11 vy=u(5);
12 vz=u(6);
13
14 du1=vx;
15 du2=vy;
16 du3=vz;
17 du4=-mu.Foco*(x/(x^2+y^2+z^2)^(3/2));
18 du5=-mu.Foco*(y/(x^2+y^2+z^2)^(3/2));
19 du6=-mu.Foco*(z/(x^2+y^2+z^2)^(3/2));
20
21 du=[du1;du2;du3;du4;du5;du6];
22
23 t
24 end

```

Listing 40: Función EfemeridesPlanetas de MATLAB.

```

1
2 function [Efemerides] = EfemeridesPlanetas()
3
4 % Eva Terradillos Est vez
5
6 % Esta funci n se encarga de leer todos los archivos txt que contienen las
7 % efemrides de los distintos planetas, el Sol y la Luna. Todas las
8 % variables son guardadas en estructuras globales distintas para poder
9 % utilizarse en cualquier momento por el programa principal u otras
10 % funciones.
11
12 global file Efemerides
13
14 clear Efemerides;
15
16 % Efemrides del Sol
17 if exist('Sol.Ephemerides.txt') == 2
18     file.Sun = 'Sol.Ephemerides.txt';
19     Efemerides.Sol = importdata(file.Sun);
20 end
21
22 % Efemerides de la Luna
23 if exist('Luna.Ephemerides.txt') == 2
24     file.Lu = 'Luna.Ephemerides.txt';
25     Efemerides.Luna = importdata(file.Lu);
26 end
27
28 % Efemrides de Mercurio
29 if exist('Mercurio.Ephemerides.txt') == 2
30     file.Me = 'Mercurio.Ephemerides.txt';
31     Efemerides.Mercurio = importdata(file.Me);
32 end
33
34 % Efemrides de Venus
35 if exist('Venus.Ephemerides.txt') == 2
36     file.V = 'Venus.Ephemerides.txt';
37     Efemerides.Venus = importdata(file.V);
38 end

```

```

39
40 % Efemerides de la Tierra
41 if exist('Tierra_Ephemerides.txt') == 2
42     file.T = 'Tierra_Ephemerides.txt'; % Efem rides de la Tierra
43     Efemerides.Tierra = importdata(file.T);
44 end
45
46 % Efem rides de Marte
47 if exist('Marte_Ephemerides.txt') == 2
48     file.Ma = 'Marte_Ephemerides.txt'; % Efem rides de Marte
49     Efemerides.Marte = importdata(file.Ma);
50 end
51
52 % Efem rides de J piter
53 if exist('Jupiter_Ephemerides.txt') == 2
54     file.J = 'Jupiter_Ephemerides.txt'; % Efem rides de Jupiter
55     Efemerides.Jupiter = importdata(file.J);
56 end
57
58 % Efem rides de Saturno
59 if exist('Saturno_Ephemerides.txt') == 2
60     file.S = 'Saturno_Ephemerides.txt'; % Efem rides de Saturno
61     Efemerides.Saturno = importdata(file.S);
62 end
63
64 % Efem rides de Urano
65 if exist('Urano_Ephemerides.txt') == 2
66     file.U = 'Urano_Ephemerides.txt'; % Efem rides de Urano
67     Efemerides.Urano = importdata(file.U);
68 end
69
70 % Efemerides de Neptuno
71 if exist('Neptuno_Ephemerides.txt') == 2
72     file.N = 'Neptuno_Ephemerides.txt'; % Efem rides de Neptuno
73     Efemerides.Neptuno = importdata(file.N);
74 end
75
76
77 end

```

Listing 41: Función IntegracionPlanSRPHarmAtm de MATLAB.

```

1
2 function du=IntegracionPlanSRPHarmAtm(t,u)
3
4 global mu values
5
6 x=u(1);
7 y=u(2);
8 z=u(3);
9 vx=u(4);
10 vy=u(5);
11 vz=u(6);
12
13 for ii = 1:length(values)
14     p_grav(ii,:) = p_Grav(t,u,values(ii));
15 end
16
17

```

```

18 if ii > 1
19     p = sum(p_grav) + p_SRP(t,u) + p_Harm(t,u) + p_Atm(t,u);
20 else
21     p = p_grav + p_SRP(t,u) + p_Harm(t,u) + p_Atm(t,u);
22 end
23
24
25 du1=vx;
26 du2=vy;
27 du3=vz;
28 du4=-mu.Foco*(x/(x^2+y^2+z^2)^(3/2))+p(1);
29 du5=-mu.Foco*(y/(x^2+y^2+z^2)^(3/2))+p(2);
30 du6=-mu.Foco*(z/(x^2+y^2+z^2)^(3/2))+p(3);
31
32 du=[du1;du2;du3;du4;du5;du6];
33
34 t
35 end

```

Listing 42: Función IntegracionPlanSRPHarm de MATLAB.

```

1
2 function du=IntegracionPlanSRPHarm(t,u)
3
4 global mu Efemerides values
5
6 x=u(1);
7 y=u(2);
8 z=u(3);
9 vx=u(4);
10 vy=u(5);
11 vz=u(6);
12
13
14 for ii = 1:length(values)
15     p_grav(ii,:) = p_Grav(t,u,values(ii));
16 end
17
18 if ii > 1
19     p = sum(p_grav)+p_SRP(t,u)+p_Harm(t,u);
20 else
21     p = p_grav+p_SRP(t,u)+p_Harm(t,u);
22 end
23
24 du1=vx;
25 du2=vy;
26 du3=vz;
27 du4=-mu.Foco*(x/(x^2+y^2+z^2)^(3/2))+p(1);
28 du5=-mu.Foco*(y/(x^2+y^2+z^2)^(3/2))+p(2);
29 du6=-mu.Foco*(z/(x^2+y^2+z^2)^(3/2))+p(3);
30
31 du=[du1;du2;du3;du4;du5;du6];
32
33 t
34 end

```

Listing 43: Función IntegracionPlanSRPAtm de MATLAB.

```

1
2 function du=IntegracionPlanSRPATm(t,u)
3
4 global mu values
5
6 x=u(1);
7 y=u(2);
8 z=u(3);
9 vx=u(4);
10 vy=u(5);
11 vz=u(6);
12
13 for ii = 1:length(values)
14     p_grav(ii,:) = p_Grav(t,u,values(ii));
15 end
16
17
18 if ii > 1
19     p = sum(p_grav) + p_SRP(t,u) + p_Atm(t,u);
20 else
21     p = p_grav + p_SRP(t,u) + p_Atm(t,u);
22 end
23
24
25 du1=vx;
26 du2=vy;
27 du3=vz;
28 du4=-mu.Foco*(x/(x^2+y^2+z^2)^(3/2))+p(1);
29 du5=-mu.Foco*(y/(x^2+y^2+z^2)^(3/2))+p(2);
30 du6=-mu.Foco*(z/(x^2+y^2+z^2)^(3/2))+p(3);
31
32 du=[du1;du2;du3;du4;du5;du6];
33
34 t
35 end

```

Listing 44: Función **IntegracionPlanSRP** de MATLAB.

```

1
2 function du=IntegracionPlanSRP(t,u)
3
4 global mu Efemerides values
5
6 x=u(1);
7 y=u(2);
8 z=u(3);
9 vx=u(4);
10 vy=u(5);
11 vz=u(6);
12
13
14 for ii = 1:length(values)
15     p_grav(ii,:) = p_Grav(t,u,values(ii));
16 end
17
18 if ii > 1
19     p = sum(p_grav)+p_SRP(t,u);
20 else
21     p = p_grav+p_SRP(t,u);

```

```

22 end
23
24 du1=vx;
25 du2=vy;
26 du3=vz;
27 du4=-mu.Foco*(x/(x^2+y^2+z^2)^(3/2))+p(1);
28 du5=-mu.Foco*(y/(x^2+y^2+z^2)^(3/2))+p(2);
29 du6=-mu.Foco*(z/(x^2+y^2+z^2)^(3/2))+p(3);
30
31 du=[du1;du2;du3;du4;du5;du6];
32
33 t
34 end

```

Listing 45: Función IntegracionPlanHarmAtm de MATLAB.

```

1
2 function du=IntegracionPlanHarmAtm(t,u)
3
4 global mu values
5
6 x=u(1);
7 y=u(2);
8 z=u(3);
9 vx=u(4);
10 vy=u(5);
11 vz=u(6);
12
13 for ii = 1:length(values)
14     p-grav(ii,:) = p-Grav(t,u,values(ii));
15 end
16
17
18 if ii > 1
19     p = sum(p-grav) + p-Harm(t,u) + p-Atm(t,u);
20 else
21     p = p-grav + p-Harm(t,u) + p-Atm(t,u);
22 end
23
24
25 du1=vx;
26 du2=vy;
27 du3=vz;
28 du4=-mu.Foco*(x/(x^2+y^2+z^2)^(3/2))+p(1);
29 du5=-mu.Foco*(y/(x^2+y^2+z^2)^(3/2))+p(2);
30 du6=-mu.Foco*(z/(x^2+y^2+z^2)^(3/2))+p(3);
31
32 du=[du1;du2;du3;du4;du5;du6];
33
34 t
35 end

```

Listing 46: Función IntegracionPlanHarm de MATLAB.

```

1
2 function du=IntegracionPlanHarm(t,u)
3
4 global mu Efemerides values

```

```

5
6 x=u(1);
7 y=u(2);
8 z=u(3);
9 vx=u(4);
10 vy=u(5);
11 vz=u(6);
12
13
14 for ii = 1:length(values)
15     p_grav(ii,:) = p_Grav(t,u,values(ii));
16 end
17
18 if ii > 1
19     p = sum(p_grav)+p_Harm(t,u);
20 else
21     p = p_grav+p_Harm(t,u);
22 end
23
24 du1=vx;
25 du2=vy;
26 du3=vz;
27 du4=-mu.Foco*(x/(x^2+y^2+z^2)^(3/2))+p(1);
28 du5=-mu.Foco*(y/(x^2+y^2+z^2)^(3/2))+p(2);
29 du6=-mu.Foco*(z/(x^2+y^2+z^2)^(3/2))+p(3);
30
31 du=[du1;du2;du3;du4;du5;du6];
32
33 t
34 end

```

Listing 47: Función IntegracionPlanAtm de MATLAB.

```

1
2 function du=IntegracionPlanAtm(t,u)
3
4 global mu values
5
6 x=u(1);
7 y=u(2);
8 z=u(3);
9 vx=u(4);
10 vy=u(5);
11 vz=u(6);
12
13 for ii = 1:length(values)
14     p_grav(ii,:) = p_Grav(t,u,values(ii));
15 end
16
17
18 if ii > 1
19     p = sum(p_grav) + p_Atm(t,u);
20 else
21     p = p_grav + p_Atm(t,u);
22 end
23
24
25 du1=vx;
26 du2=vy;

```

```

27 du3=vz;
28 du4=-mu.Foco*(x/(x^2+y^2+z^2)^(3/2))+p(1);
29 du5=-mu.Foco*(y/(x^2+y^2+z^2)^(3/2))+p(2);
30 du6=-mu.Foco*(z/(x^2+y^2+z^2)^(3/2))+p(3);
31
32 du=[du1;du2;du3;du4;du5;du6];
33
34 t
35 end

```

Listing 48: Función IntegracionPlan de MATLAB.

```

1
2 function du=IntegracionPlan(t,u)
3
4 global mu values
5
6 x=u(1);
7 y=u(2);
8 z=u(3);
9 vx=u(4);
10 vy=u(5);
11 vz=u(6);
12
13
14 for ii = 1:length(values)
15     p_grav(ii,:) = p_Grav(t,u,values(ii));
16 end
17
18 if ii > 1
19     p = sum(p_grav);
20 else
21     p = p_grav;
22 end
23
24
25 du1=vx;
26 du2=vy;
27 du3=vz;
28 du4=-mu.Foco*(x/(x^2+y^2+z^2)^(3/2))+p(1);
29 du5=-mu.Foco*(y/(x^2+y^2+z^2)^(3/2))+p(2);
30 du6=-mu.Foco*(z/(x^2+y^2+z^2)^(3/2))+p(3);
31
32 du=[du1;du2;du3;du4;du5;du6];
33
34 t
35 end

```

Listing 49: Función IntegracionSRPHarmAtm de MATLAB.

```

1
2 function du=IntegracionSRPHarmAtm(t,u)
3
4 global mu
5
6 x=u(1);
7 y=u(2);
8 z=u(3);

```

```

 9 vx=u(4);
10 vy=u(5);
11 vz=u(6);
12
13
14 p = p_SRP(t,u) + p_Harm(t,u) + p_Atm(t,u);
15
16
17 du1=vx;
18 du2=vy;
19 du3=vz;
20 du4=-mu.Foco*(x/(x^2+y^2+z^2)^(3/2))+p(1);
21 du5=-mu.Foco*(y/(x^2+y^2+z^2)^(3/2))+p(2);
22 du6=-mu.Foco*(z/(x^2+y^2+z^2)^(3/2))+p(3);
23
24 du=[du1;du2;du3;du4;du5;du6];
25
26 t
27 end

```

Listing 50: Función IntegracionSRPHarm de MATLAB.

```

1
2 function du=IntegracionSRPHarm(t,u)
3
4 global mu Efemerides values
5
6 x=u(1);
7 y=u(2);
8 z=u(3);
9 vx=u(4);
10 vy=u(5);
11 vz=u(6);
12
13 p = p_SRP(t,u)+p_Harm(t,u);
14
15
16 du1=vx;
17 du2=vy;
18 du3=vz;
19 du4=-mu.Foco*(x/(x^2+y^2+z^2)^(3/2))+p(1);
20 du5=-mu.Foco*(y/(x^2+y^2+z^2)^(3/2))+p(2);
21 du6=-mu.Foco*(z/(x^2+y^2+z^2)^(3/2))+p(3);
22
23 du=[du1;du2;du3;du4;du5;du6];
24
25 t
26 end

```

Listing 51: Función IntegracionSRPAtm de MATLAB.

```

1
2 function du=IntegracionSRPAtm(t,u)
3
4 global mu
5
6 x=u(1);
7 y=u(2);

```

```

8 z=u(3);
9 vx=u(4);
10 vy=u(5);
11 vz=u(6);
12
13
14 p = p_SRP(t,u) + p_Atm(t,u);
15
16
17 du1=vx;
18 du2=vy;
19 du3=vz;
20 du4=-mu.Foco*(x/(x^2+y^2+z^2)^(3/2))+p(1);
21 du5=-mu.Foco*(y/(x^2+y^2+z^2)^(3/2))+p(2);
22 du6=-mu.Foco*(z/(x^2+y^2+z^2)^(3/2))+p(3);
23
24 du=[du1;du2;du3;du4;du5;du6];
25
26 t
27 end

```

Listing 52: Función IntegracionSRP de MATLAB.

```

1
2 function du=IntegracionSRP(t,u)
3
4 global mu
5
6 x=u(1);
7 y=u(2);
8 z=u(3);
9 vx=u(4);
10 vy=u(5);
11 vz=u(6);
12
13
14 p = p_SRP(t,u);
15
16
17 du1=vx;
18 du2=vy;
19 du3=vz;
20 du4=-mu.Foco*(x/(x^2+y^2+z^2)^(3/2))+p(1);
21 du5=-mu.Foco*(y/(x^2+y^2+z^2)^(3/2))+p(2);
22 du6=-mu.Foco*(z/(x^2+y^2+z^2)^(3/2))+p(3);
23
24 du=[du1;du2;du3;du4;du5;du6];
25
26 t
27 end

```

Listing 53: Función IntegracionHarmAtm de MATLAB.

```

1
2 function du=IntegracionHarmAtm(t,u)
3
4 global mu values
5

```

```

6 x=u(1);
7 y=u(2);
8 z=u(3);
9 vx=u(4);
10 vy=u(5);
11 vz=u(6);
12
13 p = p-Harm(t,u) + p-Atm(t,u);
14
15
16 du1=vx;
17 du2=vy;
18 du3=vz;
19 du4=-mu.Foco*(x/(x^2+y^2+z^2)^(3/2))+p(1);
20 du5=-mu.Foco*(y/(x^2+y^2+z^2)^(3/2))+p(2);
21 du6=-mu.Foco*(z/(x^2+y^2+z^2)^(3/2))+p(3);
22
23 du=[du1;du2;du3;du4;du5;du6];
24
25 t
26 end

```

Listing 54: Función **IntegracionHarm** de MATLAB.

```

1
2 function du = IntegracionHarm(t,u)
3
4 global mu
5
6 x=u(1);
7 y=u(2);
8 z=u(3);
9 vx=u(4);
10 vy=u(5);
11 vz=u(6);
12
13 p = p-Harm(t,u);
14
15
16 du1=vx;
17 du2=vy;
18 du3=vz;
19 du4=-mu.Foco*(x/(x^2+y^2+z^2)^(3/2))+p(1);
20 du5=-mu.Foco*(y/(x^2+y^2+z^2)^(3/2))+p(2);
21 du6=-mu.Foco*(z/(x^2+y^2+z^2)^(3/2))+p(3);
22
23 du=[du1;du2;du3;du4;du5;du6];
24
25 t
26 end

```

Listing 55: Función **IntegracionAtm** de MATLAB.

```

1
2 function du=IntegracionAtm(t,u)
3
4 global mu
5

```

```

6 x=u(1);
7 y=u(2);
8 z=u(3);
9 vx=u(4);
10 vy=u(5);
11 vz=u(6);
12
13
14 p = p-Atm(t,u);
15
16
17 du1=vx;
18 du2=vy;
19 du3=vz;
20 du4=-mu.Foco*(x/(x^2+y^2+z^2)^(3/2))+p(1);
21 du5=-mu.Foco*(y/(x^2+y^2+z^2)^(3/2))+p(2);
22 du6=-mu.Foco*(z/(x^2+y^2+z^2)^(3/2))+p(3);
23
24 du=[du1;du2;du3;du4;du5;du6];
25
26 t
27 end

```

Listing 56: Función **plotter** de MATLAB

```

1
2 function plotter(t,u)
3
4 global R
5
6 % Dibujar el planeta
7 figure(1)
8 earth_sphere('km')
9
10
11 % Dibujar ejes X, Y, Z
12 line([0 2*R.T], [0 0], [0 0]); text(2*R.T, 0, 0, 'X')
13 line([0 0], [0 2*R.T], [0 0]); text(0, 2*R.T, 0, 'Y')
14 line([0 0], [0 0], [0 2*R.T]); text(0, 0, 2*R.T, 'Z')
15
16 % Dibujo de la rbita con los datos obtenidos del integrador.
17 % El punto inicial se marca con una "o" y el final con una "f"
18 hold on
19 plot3(u(:,1), u(:,2), u(:,3), 'k')
20 line([0 u(1,1)], [0 u(1,2)], [0 u(1,3)])
21 text(u(1,1), u(1,2), u(1,3), 'o')
22 text(u(end,1), u(end,2), u(end,3), 'f')
23
24 % Dise ar la direcci n de la vista
25 view([1,1,.4])
26
27 % Dujar el movimiento de la rbita
28 % comet3(y(:,1), y(:,2), y(:,3));
29
30 % Propiedades del gr fico
31 grid on
32 axis equal
33 xlabel('km')
34 ylabel('km')

```

```

35 xlabel('km')
36 hold off

```

Listing 57: Función `earth_sphere` de MATLAB

```

1
2 function [xx,yy,zz] = earth_sphere(varargin)
3 %EARTH_SPHERE Generate an earth-sized sphere.
4 % [X,Y,Z] = EARTH_SPHERE(N) generates three (N+1)-by-(N+1)
5 % matrices so that SURFACE(X,Y,Z) produces a sphere equal to
6 % the radius of the earth in kilometers. The continents will be
7 % displayed.
8 %
9 % [X,Y,Z] = EARTH_SPHERE uses N = 50.
10 %
11 % EARTH_SPHERE(N) and just EARTH_SPHERE graph the earth as a
12 % SURFACE and do not return anything.
13 %
14 % EARTH_SPHERE(N,'mile') graphs the earth with miles as the unit rather
15 % than kilometers. Other valid inputs are 'ft' 'm' 'nm' 'miles' and 'AU'
16 % for feet, meters, nautical miles, miles, and astronomical units
17 % respectively.
18 %
19 % EARTH_SPHERE(AX,...) plots into AX instead of GCA.
20 %
21 % Examples:
22 % earth_sphere('nm') produces an earth-sized sphere in nautical miles
23 %
24 % earth_sphere(10,'AU') produces 10 point mesh of the Earth in
25 % astronomical units
26 %
27 % h1 = gca;
28 % earth_sphere(h1,'mile')
29 % hold on
30 % plot3(x,y,z)
31 % produces the Earth in miles on axis h1 and plots a trajectory from
32 % variables x, y, and z
33 % Clay M. Thompson 4-24-1991, CBM 8-21-92.
34 % Will Campbell, 3-30-2010
35 % Copyright 1984-2010 The MathWorks, Inc.
36 %% Input Handling
37 [cax,args,nargs] = axescheck(varargin{:}); % Parse possible Axes input
38 error(nargchk(0,2,nargs)); % Ensure there are a valid number of inputs
39 % Handle remaining inputs.
40 % Should have 0 or 1 string input, 0 or 1 numeric input
41 j = 0;
42 k = 0;
43 n = 50; % default value
44 units = 'km'; % default value
45 for i = 1:nargs
46     if ischar(args{i})
47         units = args{i};
48         j = j+1;
49     elseif isnumeric(args{i})
50         n = args{i};
51         k = k+1;
52     end
53 end
54 if j > 1 || k > 1

```

```

55     error('Invalid input types')
56 end
57 %% Calculations
58 % Scale factors
59 Scale = {'km' 'm' 'mile'           'miles'           'nm' ...
60         'au'           'ft'};
61         1     1000 0.621371192237334 0.621371192237334 0.539956803455724 ...
62         6.6845871226706e-009 3280.839895};
63 % Identify which scale to use
64 try
65     myscale = 6378.1363*Scale{2,strcmpi(Scale(1,:),units)};
66 catch %#ok<CTCH>
67     error('Invalid units requested. Please use m, km, ft, mile, miles, nm, ...
68         or AU')
69 end
70
71 % -pi ≤ theta ≤ pi is a row vector.
72 % -pi/2 ≤ phi ≤ pi/2 is a column vector.
73 theta = (-n:2:n)/n*pi;
74 phi = (-n:2:n)'/n*pi/2;
75 cosphi = cos(phi); cosphi(1) = 0; cosphi(n+1) = 0;
76 sintheta = sin(theta); sintheta(1) = 0; sintheta(n+1) = 0;
77 x = myscale*cosphi*cos(theta);
78 y = myscale*cosphi*sintheta;
79 z = myscale*sin(phi)*ones(1,n+1);
80 %% Plotting
81 if nargout == 0
82     cax = newplot(cax);
83     % Load and define topographic data
84     load('topo.mat','topo','topomap1');
85     % Rotate data to be consistent with the Earth-Centered-Earth-Fixed
86     % coordinate conventions. X axis goes through the prime meridian.
87     % http://en.wikipedia.org/wiki/Geodetic\_system#Earth\_Centred\_Earth\_Fixed
88     % .28ECEF\_or\_ECF.29.coordinates
89     %
90     % Note that if you plot orbit trajectories in the Earth-Centered-
91     % Inertial, the orientation of the continents will be misleading.
92     topo2 = [topo(:,181:360) topo(:,1:180)]; %#ok<NODEF>
93
94     % Define surface settings
95     props.FaceColor = 'texture';
96     props.EdgeColor = 'none';
97     props.FaceLighting = 'phong';
98     props.Cdata = topo2;
99     % Create the sphere with Earth topography and adjust colormap
100    surface(x,y,z,props,'parent',cax)
101    colormap(topomap1)
102 % Replace the calls to surface and colormap with these lines if you do
103 % not want the Earth's topography displayed.
104 %     surf(x,y,z,'parent',cax)
105 %     shading flat
106 %     colormap gray
107
108     % Refine figure
109     axis equal
110     xlabel(['X [' units ']]')
111     ylabel(['Y [' units ']]')
112     zlabel(['Z [' units ']]')
113     view(127.5,30)
114 else
115     xx = x; yy = y; zz = z;

```

116 end

Listing 58: Función p_Grav de MATLAB.

```

1
2 function p_Grav = p_Grav(t,u,planeta)
3
4 % Calcula el vector perturbación causado por la fuerza gravitatoria del
5 % planeta seleccionado a partir de las efemerides del mismo.
6
7 global mu Efemerides
8 planeta = char(planeta);
9
10 switch planeta
11     case 'Sol'
12         mu_ = mu.Sol;
13         B_data = Efemerides.Sol;
14     case 'Luna'
15         mu_ = mu.Luna;
16         B_data = Efemerides.Luna;
17     case 'Mercurio'
18         mu_ = mu.Me;
19         B_data = Efemerides.Mercurio;
20     case 'Venus'
21         mu_ = mu.V;
22         B_data = Efemerides.Venus;
23     case 'Tierra'
24         mu_ = mu.T;
25         B_data = Efemerides.Tierra;
26     case 'Marte'
27         mu_ = mu.Ma;
28         B_data = Efemerides.Marte;
29     case 'Jupiter'
30         mu_ = mu.J;
31         B_data = Efemerides.Jupiter;
32     case 'Saturno'
33         mu_ = mu.S;
34         B_data = Efemerides.Saturno;
35     case 'Urano'
36         mu_ = mu.U;
37         B_data = Efemerides.Urano;
38     case 'Neptuno'
39         mu_ = mu.N;
40         B_data = Efemerides.Neptuno;
41 end
42
43 Coord = read(t,B_data);
44
45 % Vector posición del cuerpo relativa al Sol
46 r_B = [Coord(1),Coord(2),Coord(3)];
47 % Posición del cuerpo relativa al Sol
48 r_Bnorm = norm(r_B);
49 % Vector posición del cuerpo relativa a la nave
50 r_BSat = [r_B(1)-u(1),r_B(2)-u(2),r_B(3)-u(3)];
51 % Posición del cuerpo relativa a la nave
52 r_BSat_ = norm(r_BSat);
53
54 p_Grav = mu_*((r_BSat/(r_BSat_^3))-(r_B/(r_Bnorm^3)));
55

```

```
56 end
```

Listing 59: Función p_SRP de MATLAB.

```

1
2 function p_SRP = p_SRP(t,u)
3
4 global Efemerides centro cR Am
5
6 % Calcula la perturbación causada por la presión de radiación solar en cada
7 % instante de tiempo, a partir de la posición con respecto al Sol.
8
9 % Solar Radiation Pressure
10 c = 2.998e8; %Speed of light (m/s)
11 S = 1367; %Solar constant (W/m^2)
12 P_c = S/c; %Solar pressure (Pa);
13
14 % Vector posición y velocidad
15 r_sat = [u(1),u(2),u(3)];
16 %v_sat = [u(4),u(5),u(6)];
17
18 % Distancia Sol - satelite
19 if isequal(centro,'G')
20     Sol = read(t,Efemerides.Sol);
21     r_sol = [Sol(1),Sol(2),Sol(3)];
22 elseif isequal(centro,'H')
23     r_sol = [0,0,0];
24 end
25 r = r_sol - r_sat;
26 r_norm = norm(r);
27
28 % Shadow function
29
30 S = shadow(r_sat, r_sol, t);
31
32 % Aceleración causada por la perturbación
33
34 p_SRP = - S * (P_c * cR * Am * r/r_norm)*1e-3;
35
36 end

```

Listing 60: Función p_Harm de MATLAB.

```

1
2 function p_Harm = p_Harm(t,u)
3
4 global mu R J2 J3 J4 J5 J6
5
6 x = u(1);
7 y = u(2);
8 z = u(3);
9
10 r = norm([x y z]);
11
12 p_Harm = [0,0,0];
13
14 p_Harm(1) = (((3/2)*(J2*mu.T*R.T^2)/(r^5))*(x*((5*z^2/(r^2))-1))) + ...
15     (1/2)*J3*(mu.T/r^2)*(R.T/r)^3*(5*x/r)*(7*(z/r)^3-3*(z/r)) + ...

```

```

16     (5/8)*J4*(mu.T/r^2)*(R.T/r)^4*(x/r)*(3-42*(z/r)^2+63*(z/r)^4) + ...
17     (J5/8)*(mu.T/r^2)*(R.T/r)^5*(3*x/r)*(35*z/r-210*(z/r)^3+231*(z/r)^5) - ...
18     J6/16*(mu.T/r^2)*(R.T/r)^6*(x/r)*(35-945*(z/r)^2+3465*(z/r)^4-3003*(z/r)^6);
19 p_Harm(2) = (((3/2)*(J2*mu.T*R.T^2)/(r^5))*(y*((5*z^2/(r^2))-1))) + ...
20     (1/2)*J3*(mu.T/r^2)*(R.T/r)^3*(5*y/r)*(7*(z/r)^3-3*(z/r)) + ...
21     (5/8)*J4*(mu.T/r^2)*(R.T/r)^4*(y/r)*(3-42*(z/r)^2+63*(z/r)^4) + ...
22     (J5/8)*(mu.T/r^2)*(R.T/r)^5*(3*y/r)*(35*z/r-210*(z/r)^3+231*(z/r)^5) - ...
23     J6/16*(mu.T/r^2)*(R.T/r)^6*(y/r)*(35-945*(z/r)^2+3465*(z/r)^4-3003*(z/r)^6);
24 p_Harm(3) = (((3/2)*(J2*mu.T*R.T^2)/(r^5))*(z*((5*z^2/(r^2))-1))) + ...
25     *J3*(mu.T/r^2)*(R.T/r)^3*(5*z/r)*(7*(z/r)^3-3*(z/r)) + ...
26     (5/8)*J4*(mu.T/r^2)*(R.T/r)^4*(z/r)*(3-42*(z/r)^2+63*(z/r)^4) + ...
27     (J5/8)*(mu.T/r^2)*(R.T/r)^5*(3*z/r)*(35*z/r-210*(z/r)^3+231*(z/r)^5) - ...
28     J6/16*(mu.T/r^2)*(R.T/r)^6*(z/r)*(35-945*(z/r)^2+3465*(z/r)^4-3003*(z/r)^6);
29
30 end

```

Listing 61: Función p_Atm de MATLAB.

```

1
2 function p_Atm = p_Atm(t,u)
3
4 global R Am cD
5
6 wE = [0 0 7.2921159e-5]';
7
8 x = u(1);
9 y = u(2);
10 z = u(3);
11 vx = u(4);
12 vy = u(5);
13 vz = u(6);
14
15 r = norm([x y z]);
16 vrel = [vx vy vz]-cross(wE,[x y z]);
17 vrelnorm = norm(vrel);
18
19
20 p_Atm(1) = (1/2)*atmosfera(r-R.T)*10^3*vrelnorm*(cD*Am)*vrel(1);
21 p_Atm(2) = (1/2)*atmosfera(r-R.T)*10^3*vrelnorm*(cD*Am)*vrel(2);
22 p_Atm(3) = (1/2)*atmosfera(r-R.T)*10^3*vrelnorm*(cD*Am)*vrel(3);
23
24 end

```

Listing 62: Función read de MATLAB.

```

1
2 function Coord = read(t,B_data)
3
4 % Esta funci n lee el archivo de Horizons, donde se proporcionan las
5 % efemrides de Venus para cada minuto. Debido a que la funci n ode45
6 % utiliza tiempos que pueden no coincidir con los que se presentan en el
7 % Horizon, se realizar una interpolaci n si esto ocurre.
8
9 global diff_t
10
11 % Este bucle genera un vector con las efemrides del cuerpo a partir de
12 % los datos del Horizons. Adem s, permite saber si para el tiempo estudiado
13 % Horizons proporciona los valores de las efemrides. Si no es as , se

```

```

14 % realiza una interpolación.
15
16 if t == 0
17     posicion = 1;
18     Coord = B.data(posicion,:);
19 else
20     posicion = floor((t/diff_t));
21     if posicion == 0
22         posicion = posicion + 1;
23     end
24     if mod(posicion,1) == 0
25         Coord = B.data(posicion,:);
26     else
27         posicion_1 = (floor(t/diff_t));
28         posicion_2 = (ceil(t/diff_t));
29
30         B_1 = B.data(posicion_1,:);
31         B_2 = B.data(posicion_2,:);
32
33         Coord = ((B_2-B_1)/(posicion_2-posicion_1))*(posicion-posicion_1)+B_1;
34
35     end
36 end
37
38 end

```

Listing 63: Función shadow de MATLAB.

```

1
2 function S = shadow(r_sat, r_sol, t)
3
4 % Esta función permite analizar la sombra creada por los distintos planetas
5 % y satélites a lo largo de la trayectoria de la Mars2020, según su
6 % posición (Montembruck p g 80).
7
8 global R Efemerides centro
9
10 % Tierra
11
12 if isequal(centro, 'H')
13     T = read(t, Efemerides.Tierra);
14     r_T = [T(1), T(2), T(3)];
15 elseif isequal(centro, 'G')
16     r_T = [0, 0, 0];
17 end
18
19 s_T = r_sat - r_T;
20
21 a_T = asind(R.Sol/norm(r_sol - r_sat));
22 b_T = asind(R.T/norm(s_T));
23 c_T = acosd(dot(-s_T', (r_sol - r_sat))/(norm(s_T)*norm(r_sol - r_sat)));
24
25 if abs(a_T - b_T) < c_T && c_T < a_T + b_T
26
27     x_T = (c_T^2 + a_T^2 - b_T^2)/(2*c_T);
28     y_T = sqrt(a_T^2 - x_T^2);
29
30     A_T = a_T^2 * acos(x_T/a_T) + b_T^2 * acos((c_T - x_T)/b_T) - c_T * y_T;
31

```

```

32     S_T = 1 - (A_T/(pi*a_T^2));
33
34     elseif a_T + b_T ≤ c_T
35         S_T = 1;
36     elseif b_T - a_T > c_T && a_T < b_T
37         S_T = 0;
38     elseif a_T - b_T > c_T && a_T < b_T
39         S_T = 0;
40     end
41
42
43     % Luna
44
45     L = read(t,Efemerides.Luna);
46     r_L = [L(1),L(2),L(3)];
47
48     s_L = r_sat - r_L;
49
50     a_L = asin(R.Sol/norm(r_sol - r_sat));
51     b_L = asin(R.L/norm(s_L));
52     c_L = acos(dot(-s_L',(r_sol - r_sat))/(norm(s_L)*norm(r_sol -r_sat)));
53
54     if abs(a_L - b_L) < c_L && c_L < a_L + b_L
55
56         x_L = (c_L^2 + a_L^2 - b_L^2)/(2*c_L);
57         y_L = sqrt(a_L^2-x_L^2);
58
59         A_L = a_L^2 * acos(x_L/a_L) + b_L^2 * acos((c_L - x_L)/b_L) - c_L * y_L;
60
61         S_L = 1 - (A_L/(pi*a_L^2));
62
63     elseif a_L + b_L ≤ c_L
64
65         S_L = 1;
66
67     elseif b_L - a_L > c_L && a_L < b_L
68
69         S_L = 0;
70
71     elseif a_L - b_L > c_L && a_L < b_L
72
73         S_L = 0;
74
75     end
76
77
78     % Venus
79
80     V = read(t,Efemerides.Venus);
81     r_V = [V(1),V(2),V(3)];
82
83     s_V = r_sat - r_V;
84
85     a_V = asin(R.Sol/norm(r_sol - r_sat));
86     b_V = asin(R.V/norm(s_V));
87     c_V = acos(dot(-s_V',(r_sol - r_sat))/(norm(s_V)*norm(r_sol -r_sat)));
88
89
90     if abs(a_V - b_V) < c_V && c_V < a_V + b_V
91
92         x_V = (c_V^2 + a_V^2 - b_V^2)/(2*c_V);

```

```

93     y_V = sqrt(a_V^2-x_V^2);
94
95     A_V = a_V^2 * acos(x_V/a_V) + b_V^2 * acos((c_V - x_V)/b_V) - c_V * y_V;
96
97     S_V = 1 - (A_V/(pi*a_V^2));
98
99     elseif a_V + b_V ≤ c_V
100
101         S_V = 1;
102
103     elseif b_V - a_V > c_V && a_V < b_V
104
105         S_V = 0;
106
107     elseif a_V - b_V > c_V && a_V < b_V
108
109         S_V = 0;
110
111     end
112
113
114     % Total
115
116     S = S_T * S_L * S_V;
117
118     end

```

Listing 64: Función **atmosfera** de MATLAB.

```

1
2  function densidad = atmosfera(altura)
3
4  % Esta funci n calcula la densidad de la atm sfera desde altitudes a nivel
5  % del mar hasta 1000 km utilizando la interpolaci n exponencial.
6
7  % Altitudes geom tricas (km):
8  hache = [0 25 30 40 50 60 70 80 90 100 110 120 130 140 150 180 200 250 300 ...
9          350 400 450 500 600 700 800 900 1000];
10
11 % Densidades de USSA76 (kg/m^3):
12 erre = [1.225 4.008e-2 1.841e-2 3.996e-3 1.027e-3 3.097e-4 8.283e-5 1.846e-5 ...
13         3.416e-6 5.606e-7 9.708e-8 2.222e-8 8.152e-9 3.831e-9 2.076e-9 5.194e-10 ...
14         2.541e-10 6.073e-11 1.916e-11 7.014e-12 2.803e-12 1.184e-12 5.215e-13 ...
15         1.137e-13 3.070e-14 1.136e-14 5.759e-15 3.561e-15];
16
17 % Alturas de escala (km):
18 Hache = [7.310 6.427 6.546 7.360 8.342 7.583 6.661 5.927 5.533 5.703 6.782 ...
19         9.973 13.243 16.322 21.652 27.974 34.934 43.342 49.755 54.513 58.019 ...
20         60.980 65.654 76.377 100.587 147.203 208.020];
21
22 % Para altitudes fuera del rango:
23 if altura > 1000
24     altura = 1000;
25 elseif altura < 0
26     altura = 0;
27 end
28
29 % Intervalo de interpolaci n:
30 for j = 1:27

```

```

31     if altura ≥ hache(j) && altura < hache(j+1)
32         i = j;
33     end
34 end
35
36 if altura == 1000
37     i = 27;
38 end
39
40 % Interpolación exponencial:
41 densidad = erre(i)*exp(-(altura - hache(i))/Hache(i));
42
43 end

```

Listing 65: Código para la validación de resultados a partir de la lectura de archivos de texto.

```

1
2 clc;clear all;close all;
3
4 % Eva Terradillos
5
6 % Este archivo se encarga de leer los txt obtenidos de las simulaciones con
7 % GMAT y con la aplicacion y compararlos.
8
9 % Tiempo de propagacion
10
11 t0 = 0; % Instante inicial [s]
12 tf = 10*24*3600; % Periodo a propagar [s]
13 diff_t = 60; % Segundos entre dos iteraciones
14 int = (tf/diff_t)+1; % Numero de intervalos
15 tspan = linspace(t0,tf,int);
16
17
18 %Lectura del archivos txt
19 % 1. Generado por STK
20
21 filename = 'Total.txt';
22 delimiterIn = ' ';
23 GMAT = importdata(filename,delimiterIn);
24
25 % Creación de los vectores de resultados
26
27 r_GMAT = [GMAT.data(:,1),GMAT.data(:,2),GMAT.data(:,3)];
28 v_GMAT = [GMAT.data(:,4),GMAT.data(:,5),GMAT.data(:,6)];
29
30 % 2. Generados por la aplicacion
31 filename = 'Posición.txt';
32 delimiterIn = ' ';
33 r_APP = importdata(filename,delimiterIn);
34
35 filename = 'Velocidad.txt';
36 delimiterIn = ' ';
37 v_APP = importdata(filename,delimiterIn);
38
39
40 % Calculo del error (se ha utilizado un paso de 60 segundos en ambos para
41 % realizar la comparativa en cada instante de tiempo simulado)
42
43

```

```

44 for ii = 1:length(tspan(1,:))
45
46     error_r_J(ii,:) = (r_APP(ii,:)-r_GMAT(ii,:));
47     error_v_J(ii,:) = (v_APP(ii,:)-v_GMAT(ii,:));
48
49     error_r_abs(ii) = abs(norm(r_APP(ii,:))-norm(r_GMAT(ii,:)));
50     error_v_abs(ii) = abs(norm(v_APP(ii,:))-norm(v_GMAT(ii,:)));
51
52     error_r_rel(ii) = error_r_abs(ii)/norm(r_GMAT(ii,:));
53     error_v_rel(ii) = error_v_abs(ii)/norm(v_GMAT(ii,:));
54
55
56 end
57
58
59 % Graficas de la evolucion del error a lo largo de la propagacion
60
61 pos_x = figure();
62 plot(tspan',error_r_J(:,1));
63 xlabel('Tiempo de simulacion [s]')
64 ylabel('Error [km]')
65 title('Error en la coordenada X de la posicion')
66
67 pos_y = figure();
68 plot(tspan',error_r_J(:,2));
69 xlabel('Tiempo de simulacion [s]')
70 ylabel('Error [km]')
71 title('Error en la coordenada Y de la posicion')
72
73 pos_z = figure();
74 plot(tspan',error_r_J(:,3));
75 xlabel('Tiempo de simulacion [s]')
76 ylabel('Error [km]')
77 title('Error en la coordenada Z de la posicion')
78
79 pos_abs = figure()
80 plot(tspan',error_r_abs);
81 xlabel('Tiempo de simulacion [s]')
82 ylabel('Error [km]')
83 title('Error absoluto en la posicion')
84
85 pos_rel = figure();
86 plot(tspan',error_r_rel);
87 xlabel('Tiempo de simulacion [s]')
88 ylabel('Error [-]')
89 title('Error relativo en la posicion')
90
91 vel_x = figure();
92 plot(tspan',error_v_J(:,1));
93 xlabel('Tiempo de simulacion [s]')
94 ylabel('Error [km/s]')
95 title('Error en la coordenada X de la velocidad')
96
97 vel_y = figure();
98 plot(tspan',error_v_J(:,2));
99 xlabel('Tiempo de simulacion [s]')
100 ylabel('Error [km/s]')
101 title('Error en la coordenada Y de la velocidad')
102
103 vel_z = figure()
104 plot(tspan',error_v_J(:,3));

```

```

105 xlabel('Tiempo de simulaci n [s]')
106 ylabel('Error [km/s]')
107 title('Error en la coordenada Z de la velocidad')
108
109 vel_abs = figure();
110 plot(tspan',error_v_abs);
111 xlabel('Tiempo de simulaci n [s]')
112 ylabel('Error [km/s]')
113 title('Error absoluto en la velocidad')
114
115 vel_rel = figure();
116 plot(tspan',error_v_rel);
117 xlabel('Tiempo de simulaci n [s]')
118 ylabel('Error [-]')
119 title('Error relativo en la velocidad')
120
121 % Guardar graficas
122 saveas(pos_x, 'error_pos_x.eps','epsc');
123 saveas(pos_y, 'error_pos_y.eps','epsc');
124 saveas(pos_z, 'error_pos_z.eps','epsc');
125 saveas(pos_abs, 'error_pos_abs.eps','epsc');
126 saveas(pos_rel, 'error_pos_rel.eps','epsc');
127
128 saveas(vel_x, 'error_vel_x.eps','epsc');
129 saveas(vel_y, 'error_vel_y.eps','epsc');
130 saveas(vel_z, 'error_vel_z.eps','epsc');
131 saveas(vel_abs, 'error_vel_abs.eps','epsc');
132 saveas(vel_rel, 'error_vel_rel.eps','epsc');

```

Listing 66: Comparativa entre la posición y la velocidad del satélite Terra calculadas con la aplicación, y obtenidas con los datos del *Horizons*.

```

1
2 clc; clear all; close all;
3
4 % Eva Terradillos
5
6 date_0 = [2020,03,01,00,00,00];
7 date_f = [2020,05,01,00,00,00];
8 dt = etime(date_f,date_0);
9
10 t0 = 0; % Instante inicial [s]
11 tf = t0+dt; % Periodo a propagar [s]
12 diff_t = 10800; % Segundos entre dos iteraciones
13 int = (tf/diff_t)+1; % N mero de intervalos
14 tspan = linspace(t0,tf,int);
15
16
17 %Lectura del archivos txt
18 % 1. Valores de Horizons para la Mars 2020
19
20 filename = 'Terra.txt';
21 Terra_ = importdata(filename);
22 TerraSat = Terra_.data;
23 r_Terra = [TerraSat(1:180:end,1),TerraSat(1:180:end,2),TerraSat(1:180:end,3)];
24 v_Terra = [TerraSat(1:180:end,4),TerraSat(1:180:end,5),TerraSat(1:180:end,6)];
25
26 % 2. Valores obtenidos de la aplicaci n
27

```

```

28 filename = 'Posici n.txt';
29 delimiterIn = ' ';
30 r_APP = importdata(filename,delimiterIn);
31 r_APP = [r_APP(1:180:end,1),r_APP(1:180:end,2),r_APP(1:180:end,3)];
32
33 filename = 'Velocidad.txt';
34 delimiterIn = ' ';
35 v_APP = importdata(filename,delimiterIn);
36 v_APP = [v_APP(1:180:end,1),v_APP(1:180:end,2),v_APP(1:180:end,3)];
37
38 % Errores
39
40 for ii = 1:(length(tspan(1,:)))
41
42     error_r_J(ii,:) = (r_APP(ii,:)-r_Terra(ii,:));
43     error_v_J(ii,:) = (v_APP(ii,:)-v_Terra(ii,:));
44
45     error_r_abs(ii) = abs(norm(r_APP(ii,:))-norm(r_Terra(ii,:)));
46     error_v_abs(ii) = abs(norm(v_APP(ii,:))-norm(v_Terra(ii,:)));
47
48     error_r_rel(ii) = error_r_abs(ii)/norm(r_Terra(ii,:));
49     error_v_rel(ii) = error_v_abs(ii)/norm(v_Terra(ii,:));
50
51
52 end
53
54
55 %% Gr ficas de la evoluci n del error a lo largo de la propagaci n
56
57 pos_x = figure();
58 plot(tspan',error_r_J(:,1));
59 xlabel('Tiempo de simulaci n [s]')
60 ylabel('Error [km]')
61 title('Error en la coordenada X de la posici n')
62
63 pos_y = figure();
64 plot(tspan',error_r_J(:,2));
65 xlabel('Tiempo de simulaci n [s]')
66 ylabel('Error [km]')
67 title('Error en la coordenada Y de la posici n')
68
69 pos_z = figure();
70 plot(tspan',error_r_J(:,3));
71 xlabel('Tiempo de simulaci n [s]')
72 ylabel('Error [km]')
73 title('Error en la coordenada Z de la posici n')
74
75 pos_abs = figure();
76 plot(tspan',error_r_abs);
77 xlabel('Tiempo de simulaci n [s]')
78 ylabel('Error [km]')
79 title('Error absoluto en la posici n')
80
81 pos_rel = figure();
82 plot(tspan',error_r_rel);
83 xlabel('Tiempo de simulaci n [s]')
84 ylabel('Error [-]')
85 title('Error relativo en la posici n')
86
87 vel_x = figure();
88 plot(tspan',error_v_J(:,1));

```

```

89 xlabel('Tiempo de simulaci n [s]')
90 ylabel('Error [km/s]')
91 title('Error en la coordenada X de la velocidad')
92
93 vel_y = figure();
94 plot(tspan',error_v-J(:,2));
95 xlabel('Tiempo de simulaci n [s]')
96 ylabel('Error [km/s]')
97 title('Error en la coordenada Y de la velocidad')
98
99 vel_z = figure();
100 plot(tspan',error_v-J(:,3));
101 xlabel('Tiempo de simulaci n [s]')
102 ylabel('Error [km/s]')
103 title('Error en la coordenada Z de la velocidad')
104
105 vel_abs = figure();
106 plot(tspan',error_v_abs);
107 xlabel('Tiempo de simulaci n [s]')
108 ylabel('Error [km/s]')
109 title('Error absoluto en la velocidad')
110
111 vel_rel = figure();
112 plot(tspan',error_v_rel);
113 xlabel('Tiempo de simulaci n [s]')
114 ylabel('Error [-]')
115 title('Error relativo en la velocidad')
116
117 % Guardar gr ficas
118 saveas(pos_x, 'error_pos.x.eps','epsc');
119 saveas(pos_y, 'error_pos.y.eps','epsc');
120 saveas(pos_z, 'error_pos.z.eps','epsc');
121 saveas(pos_abs, 'error_pos.abs.eps','epsc');
122 saveas(pos_rel, 'error_pos.rel.eps','epsc');
123
124 saveas(vel_x, 'error_vel.x.eps','epsc');
125 saveas(vel_y, 'error_vel.y.eps','epsc');
126 saveas(vel_z, 'error_vel.z.eps','epsc');
127 saveas(vel_abs, 'error_vel.abs.eps','epsc');
128 saveas(vel_rel, 'error_vel.rel.eps','epsc');

```

Listing 67: Estudio previo de la órbita para el cambio de altura en el perigeo.

```

1
2 clc;clear all; close all;
3
4 global mu
5
6 mu = 3.986020525817198e+05; % [km^3/s^2] Tierra
7
8 r_0 = [7000, 1000, 6000];
9 v_0 = [2, 7, 0];
10
11 u0 = [r_0(1),r_0(2),r_0(3),v_0(1),v_0(2),v_0(3)];
12
13 date_0 = [2020, 12, 5, 00, 00, 00];
14 date_f = [2020, 12, 10, 00, 00, 00];
15
16 dt = etime(date_f,date_0);

```

```

17
18 t0 = 0; % Instante inicial
19 tf = t0+dt; % Instante final
20 diff_t = 1; % Segundos entre dos iteraciones
21 int = (tf/diff_t)+1; % N mero de intervalos
22 tspan = linspace(t0,tf,int);
23
24 % Opciones de integraci n
25
26 options=odeset('AbsTol',1e-8,'RelTol',1e-8);
27
28 [t,u] = ode45(@Integracion,tspan,u0,options);
29
30 for ii = 1:length(u(:,1))
31
32     r_ = [u(ii,1),u(ii,2),u(ii,3)];
33     v_ = [u(ii,4),u(ii,5),u(ii,5)];
34
35     r = norm(r_);
36     v = norm(v_);
37
38     % Momento angular espec fico
39     h_vector = cross(r_,v_);
40
41     h = norm(h_vector);
42
43     % Excentricidad
44     B_vector = cross(v_,h_vector)-mu*(r_/r);
45
46     e_vector = B_vector/mu;
47     e = norm(e_vector);
48
49     % Anomala verdadera
50
51     rv_vector = v_.*r_;
52
53     K_vector = [0,0,1];
54     n_vector = cross(K_vector,h_vector);
55
56     if rv_vector > 0
57         theta(ii) = rad2deg(acos(dot(e_vector,r_)/(e*r)));
58     else
59         theta(ii) = 360-rad2deg(acos(dot(e_vector,r_)/(e*r)));
60     end
61
62 end
63
64 %% Aumento de la altura del perigeo (apogeo->theta=180)
65
66 pos = find((abs(180-theta))< 10e-5);
67
68 %% Obtenci n de la fecha en la que theta es m s cercano a 0/360
69
70 % Segundo
71 paso_1 = floor(pos/60); %[min]
72 pos_1 = paso_1*60; %[s]
73 seg = pos - pos_1; %[s]
74
75 % Minuto
76 paso_2 = floor(paso_1/60); %[h]
77 pos_2 = paso_2*60; %[min]

```

```

78 min    = paso_1 - pos_2;    %[min]
79
80 % Hora
81 paso_3 = floor(paso_2/24); %[d as]
82 pos_3  = paso_3 * 24;     %[h]
83 hora   = paso_2 - pos_3;  %[h]
84
85 % Al ser el paso_3 = 0, no es necesario obtener m s datos del momento en el
86 % que se quiere dar el impulso.
87
88 date_imp = date_0 + [0,0,0, hora(2), min(2), seg(2)];
89
90 % Para comprobar obtener los segundos entre fecha inicial y obtenida y
91 % deben de ser igual a la variable pos
92
93 dt_imp = etime(date_imp,date_0);

```

Listing 68: Estudio previo de la órbita para el cambio de altura en el apogeo.

```

1
2 clc;clear all; close all;
3
4 global mu
5
6 mu = 3.986020525817198e+05; % [km^3/s^2] Tierra
7
8 r_0 = [7000, 1000, 6000];
9 v_0 = [2, 7, 0];
10
11 u0 = [r_0(1),r_0(2),r_0(3),v_0(1),v_0(2),v_0(3)];
12
13 date_0 = [2020, 12, 5, 00, 00, 00];
14 date_f = [2020, 12, 10, 00, 00, 00];
15
16 dt = etime(date_f,date_0);
17
18 t0 = 0; % Instante inicial
19 tf = t0+dt; % Instante final
20 diff_t = 1; % Segundos entre dos iteraciones
21 int = (tf/diff_t)+1; % N mero de intervalos
22 tspan = linspace(t0,tf,int);
23
24 % Opciones de integraci n
25
26 options=odeset('AbsTol',1e-8,'RelTol',1e-8);
27
28 [t,u] = ode45(@Integracion,tspan,u0,options);
29
30 for ii = 1:length(u(:,1))
31
32     r_ = [u(ii,1),u(ii,2),u(ii,3)];
33     v_ = [u(ii,4),u(ii,5),u(ii,5)];
34
35     r = norm(r_);
36     v = norm(v_);
37
38     % Momento angular espec fico
39     h_vector = cross(r_,v_);
40

```

```

41     h = norm(h_vector);
42
43     % Excentricidad
44     B_vector = cross(v_-,h_vector)-mu*(r_-/r);
45
46     e_vector = B_vector/mu;
47     e = norm(e_vector);
48
49     % Anomalia verdadera
50
51     rv_vector = v_-.*r_-;
52
53     K_vector = [0,0,1];
54     n_vector = cross(K_vector,h_vector);
55
56     if rv_vector > 0
57         theta(ii) = rad2deg(acos(dot(e_vector,r_-)/(e*r)));
58     else
59         theta(ii) = 360-rad2deg(acos(dot(e_vector,r_-)/(e*r)));
60     end
61
62 end
63
64 %% Aumento de la altura del apogeo (perigeo->theta=0)
65
66 theta_max = max(theta);
67 theta_min = min(theta);
68
69 dif_max = 360 - theta_max;
70 dif_min = theta_min;
71
72 if dif_max < dif_min
73     pos = find(theta > floor(theta_max*10)/10);
74 else
75     pos = find(theta < ceil(theta_min*10)/10);
76 end
77
78 %% Obtención de la fecha en la que theta es más cercano a 0/360
79
80 % Segundo
81 paso_1 = floor(pos(1)/60); %[min]
82 pos_1 = paso_1*60;      %[s]
83 seg = pos(1:4) - pos_1;  %[s]
84
85 % Minuto
86 paso_2 = floor(paso_1/60); %[h]
87 pos_2 = paso_2*60;      %[min]
88 min = paso_1 - pos_2;   %[min]
89
90 % Hora
91 paso_3 = floor(paso_2/24); %[d as]
92 pos_3 = paso_3 * 24;    %[h]
93 hora = paso_2 - pos_3;  %[h]
94
95 % Al ser el paso_3 = 0, no es necesario obtener más datos del momento en el
96 % que se quiere dar el impulso.
97
98 date_imp_1 = date_0 + [0,0,0, hora(1), min(1), seg(1)];
99
100 % Para comprobar obtener los segundos entre fecha inicial y obtenida y
101 % deben de ser igual a la variable pos

```

```

102
103 dt_imp(1) = etime(date_imp_1,date_0);

```

Listing 69: Comparativa entre la posición y la velocidad de la Mars2020 calculadas con la aplicación y obtenidas con los datos del *Horizons*.

```

1
2 clc; clear all; close all;
3
4 % Eva Terradillos
5
6 date_0 = [2020,7,31,12,53,00];
7 date_f = [2021,2,17,20,37,00];
8 dt = etime(date_f,date_0);
9
10 t0 = 0; % Instante inicial [s]
11 tf = t0+dt; % Periodo a propagar [s]
12 diff_t = 60; % Segundos entre dos iteraciones
13 int = (tf/diff_t)+1; % N mero de intervalos
14 tspan = linspace(t0,tf,int);
15
16
17 %Lectura del archivos txt
18 % 1. Valores de Horizons para la Mars 2020
19
20 filename = 'Tramo1.txt';
21 delimiterIn = ' ';
22 Mars_1 = importdata(filename);
23
24 filename = 'Tramo2.txt';
25 delimiterIn = ' ';
26 Mars_2 = importdata(filename);
27
28 filename = 'Tramo3.txt';
29 delimiterIn = ' ';
30 Mars_3 = importdata(filename);
31
32 filename = 'Tramo4.1.txt';
33 delimiterIn = ' ';
34 Mars_41 = importdata(filename);
35
36 filename = 'Tramo4.2.txt';
37 delimiterIn = ' ';
38 Mars_42 = importdata(filename);
39
40 filename = 'Tramo4.3.txt';
41 delimiterIn = ' ';
42 Mars_43 = importdata(filename);
43
44 filename = 'Tramo5.1.txt';
45 delimiterIn = ' ';
46 Mars_51 = importdata(filename);
47
48 filename = 'Tramo5.2.txt';
49 delimiterIn = ' ';
50 Mars_52 = importdata(filename);
51
52
53 % Creaci n de los vectores de resultados

```

```

54
55 % r_Mars = [Mars_1.data(:,1),Mars_1.data(:,2),Mars_1.data(:,3)];
56 % v_Mars = [Mars_1.data(:,4),Mars_1.data(:,5),Mars_1.data(:,6)];
57
58 r_Mars(:,1) = [Mars_1.data(1:end-1,1);Mars_2.data(1:end-1,1);...
59     Mars_3.data(1:end-1,1);Mars_41.data(1:end-1,1);...
60     Mars_42.data(1:end-1,1);Mars_43.data(1:end-1,1);...
61     Mars_51.data(1:end-1,1);Mars_52.data(:,1)];
62
63 r_Mars(:,2) = [Mars_1.data(1:end-1,2);Mars_2.data(1:end-1,2);...
64     Mars_3.data(1:end-1,2);Mars_41.data(1:end-1,2);...
65     Mars_42.data(1:end-1,2);Mars_43.data(1:end-1,2);...
66     Mars_51.data(1:end-1,2);Mars_52.data(:,2)];
67
68 r_Mars(:,3) = [Mars_1.data(1:end-1,3);Mars_2.data(1:end-1,3);...
69     Mars_3.data(1:end-1,3);Mars_41.data(1:end-1,3);...
70     Mars_42.data(1:end-1,3);Mars_43.data(1:end-1,3);...
71     Mars_51.data(1:end-1,3);Mars_52.data(:,3)];
72
73
74 v_Mars(:,1) = [Mars_1.data(1:end-1,4);Mars_2.data(1:end-1,4);...
75     Mars_3.data(1:end-1,4);Mars_41.data(1:end-1,4);...
76     Mars_42.data(1:end-1,4);Mars_43.data(1:end-1,4);...
77     Mars_51.data(1:end-1,4);Mars_52.data(:,4)];
78
79 v_Mars(:,2) = [Mars_1.data(1:end-1,5);Mars_2.data(1:end-1,5);...
80     Mars_3.data(1:end-1,5);Mars_41.data(1:end-1,5);...
81     Mars_42.data(1:end-1,5);Mars_43.data(1:end-1,5);...
82     Mars_51.data(1:end-1,5);Mars_52.data(:,5)];
83
84 v_Mars(:,3) = [Mars_1.data(1:end-1,6);Mars_2.data(1:end-1,6);...
85     Mars_3.data(1:end-1,6);Mars_41.data(1:end-1,6);...
86     Mars_42.data(1:end-1,6);Mars_43.data(1:end-1,6);...
87     Mars_51.data(1:end-1,6);Mars_52.data(:,6)];
88
89 % 2. Valores obtenidos de la aplicaci n
90
91 filename = 'Posici n.txt';
92 delimiterIn = ' ';
93 r_APP = importdata(filename,delimiterIn);
94
95 filename = 'Velocidad.txt';
96 delimiterIn = ' ';
97 v_APP = importdata(filename,delimiterIn);
98
99
100 % Errores
101
102 for ii = 1:length(tspan(1,:))
103
104     error_r_J(ii,:) = (r_APP(ii,:)-r_Mars(ii,:));
105     error_v_J(ii,:) = (v_APP(ii,:)-v_Mars(ii,:));
106
107     error_r_abs(ii) = abs(norm(r_APP(ii,:))-norm(r_Mars(ii,:)));
108     error_v_abs(ii) = abs(norm(v_APP(ii,:))-norm(v_Mars(ii,:)));
109
110     error_r_rel(ii) = error_r_abs(ii)/norm(r_Mars(ii,:));
111     error_v_rel(ii) = error_v_abs(ii)/norm(v_Mars(ii,:));
112
113
114 end

```

```
115
116
117 %% Gráficas de la evolución del error a lo largo de la propagación
118
119 pos_x = figure();
120 plot(tspan', error_r_J(:,1));
121 xlabel('Tiempo de simulación [s]')
122 ylabel('Error [km]')
123 title('Error en la coordenada X de la posición')
124
125 pos_y = figure();
126 plot(tspan', error_r_J(:,2));
127 xlabel('Tiempo de simulación [s]')
128 ylabel('Error [km]')
129 title('Error en la coordenada Y de la posición')
130
131 pos_z = figure();
132 plot(tspan', error_r_J(:,3));
133 xlabel('Tiempo de simulación [s]')
134 ylabel('Error [km]')
135 title('Error en la coordenada Z de la posición')
136
137 pos_abs = figure();
138 plot(tspan', error_r_abs);
139 xlabel('Tiempo de simulación [s]')
140 ylabel('Error [km]')
141 title('Error absoluto en la posición')
142
143 pos_rel = figure();
144 plot(tspan', error_r_rel);
145 xlabel('Tiempo de simulación [s]')
146 ylabel('Error [-]')
147 title('Error relativo en la posición')
148
149 vel_x = figure();
150 plot(tspan', error_v_J(:,1));
151 xlabel('Tiempo de simulación [s]')
152 ylabel('Error [km/s]')
153 title('Error en la coordenada X de la velocidad')
154
155 vel_y = figure();
156 plot(tspan', error_v_J(:,2));
157 xlabel('Tiempo de simulación [s]')
158 ylabel('Error [km/s]')
159 title('Error en la coordenada Y de la velocidad')
160
161 vel_z = figure();
162 plot(tspan', error_v_J(:,3));
163 xlabel('Tiempo de simulación [s]')
164 ylabel('Error [km/s]')
165 title('Error en la coordenada Z de la velocidad')
166
167 vel_abs = figure();
168 plot(tspan', error_v_abs);
169 xlabel('Tiempo de simulación [s]')
170 ylabel('Error [km/s]')
171 title('Error absoluto en la velocidad')
172
173 vel_rel = figure();
174 plot(tspan', error_v_rel);
175 xlabel('Tiempo de simulación [s]')
```

```
176 ylabel('Error [-]')
177 title('Error relativo en la velocidad')
178
179 % Guardar gráficas
180 saveas(pos_x, 'error_pos_x.eps', 'epsc');
181 saveas(pos_y, 'error_pos_y.eps', 'epsc');
182 saveas(pos_z, 'error_pos_z.eps', 'epsc');
183 saveas(pos_abs, 'error_pos_abs.eps', 'epsc');
184 saveas(pos_rel, 'error_pos_rel.eps', 'epsc');
185
186 saveas(vel_x, 'error_vel_x.eps', 'epsc');
187 saveas(vel_y, 'error_vel_y.eps', 'epsc');
188 saveas(vel_z, 'error_vel_z.eps', 'epsc');
189 saveas(vel_abs, 'error_vel_abs.eps', 'epsc');
190 saveas(vel_rel, 'error_vel_rel.eps', 'epsc');
```

C Presupuesto

En este anexo se presenta el presupuesto que ha sido necesario para la realización de este trabajo.

C.1 Recursos humanos

En primer lugar se recogen el presupuesto del proyecto para los recursos humanos empleados.

Cuadro 1: Presupuestos para los recursos humanos del proyecto.

Categoría	Nº de personas	Nº de horas	Precio unitario (€)	Importe total (€)
Alumno	1	500	12	6000
Total (€)				6000

C.2 Software para el despliegue

A continuación en la tabla mostrada a continuación se recoge el presupuesto necesario para los recursos de software empleados a lo largo del desarrollo de este trabajo.

Cuadro 2: Presupuesto para los recursos de software utilizados a lo largo del proyecto.

Descripción	Unidades	Precio unitario (€)	Importe total (€)
Lincencia MATLAB	1	800/año	800
Total (€)			800

C.3 Recursos para el desarrollo

Por último se presenta el presupuesto necesario para los recursos informáticos empleados a lo largo del proyecto.

Cuadro 3: Presupuesto para los recursos informáticos necesarios para el desarrollo del proyecto.

Descripción	Unidades	Precio unitario (€)	Amortización	Importe amortizado	Importe total (€)
Ordenador personal	1	1200	25 %	300	300
Windows 10	1	200	25 %	50	50
Microsoft office 365	1	70/año	100 %	70	70
Total (€)				420	

C.4 Presupuesto total

Por lo que el presupuesto total necesario para el desarrollo del proyecto es el siguiente:

Cuadro 4: Presupuesto total para el desarrollo del proyecto.

Descripción	Importe total (€)
Recursos Humanos	6000
Software para el despliegue	800
Recursos para el desarrollo	420
	7220