
Algoritmos computacionales para órbitas de transferencia



Presentado por
Edward Andrés Gil González

LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA

Fundación Universitaria Los Libertadores

Facultad de Ingeniería y Ciencias Básicas

Programa de Ingeniería Aeronáutica

Bogotá D.C, Colombia

2019

Página dejada en blanco intencionalmente - No forma parte del cuerpo del trabajo.

Algoritmos computacionales para órbitas de transferencia

Presentado por

Edward Andrés Gil González

en cumplimiento parcial de los requerimientos para optar al título

FUNDACIÓN UNIVERSITARIA

de

Ingeniero Aeronáutico

Dirigida por:

Jorge Luis Nisperuza Toledo, Ph. D.

Codirector:

Vladimir Alfonso Ballesteros Ballesteros, M. Eng.

Fundación Universitaria Los Libertadores

Facultad de Ingeniería y Ciencias Básicas

Programa de Ingeniería Aeronáutica

Bogotá D.C, Colombia

2019

Página dejada en blanco intencionalmente - No forma parte del cuerpo del trabajo.

Notas de aceptación



LOS LIBERTADORES

FUNDACIÓN UNIVERSITARIA

Firma del jurado 1

Firma del jurado 2

Bogotá D.C, Diciembre de 2019.

Página dejada en blanco intencionalmente - No forma parte del cuerpo del trabajo.



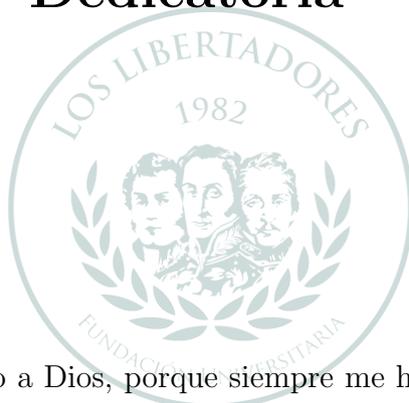
LOS LIBERTADORES

FUNDACIÓN UNIVERSITARIA

Las directivas de la Fundación Universitaria Los Libertadores, los jurados calificadores y el cuerpo docente no son responsables por los criterios e ideas expuestas en el presente documento. Estos corresponden únicamente a los autores y a los resultados de su trabajo.

Página dejada en blanco intencionalmente - No forma parte del cuerpo del trabajo.

Dedicatoria

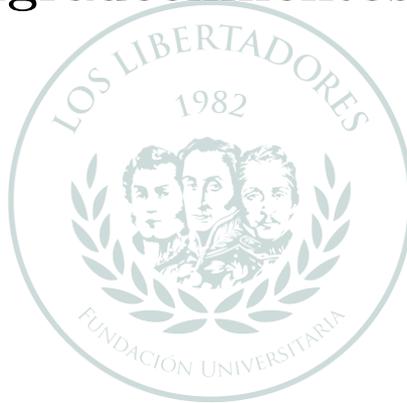


Este trabajo está dedicado a Dios, porque siempre me ha llevado de su mano, me ha otorgado discernimiento, comprensión y sabiduría en todo momento, también a todas las personas que hacen parte de mi vida, las cuales me han brindado su apoyo para culminar este programa académico, siendo uno de los primeros pasos para cumplir las metas que me he propuesto.

FUNDACIÓN UNIVERSITARIA

Página dejada en blanco intencionalmente - No forma parte del cuerpo del trabajo.

Agradecimientos



LOS LIBERTADORES FUNDACIÓN UNIVERSITARIA

En primer lugar, agradezco a Dios por la salud, sabiduría y discernimiento que me ha brindado durante mis estudios cursados, por el acompañamiento permanente para llevar a cabo todos los proyectos propuestos y por darme lo más preciado, mi familia.

A mis padres Cristina González Cubillos y Víctor Gil Hurtado por brindarme su compañía, amor, enseñanza, dedicación, consejos y apoyo, por ser la fuente de mi inspiración y por la confianza que depositaron en mi para formarme como un profesional.

A mis hermanos Diego y Deivy Gil por el apoyo incondicional y el ejemplo que me han brindado desde pequeño, así como los valores que me han inculcado para ser una persona de bien y trabajadora.

Expreso mis agradecimientos al PhD. Jorge Luis Nisperuza por las asesorías, consejos, tiempo, compromiso y acompañamiento en el desarrollo de este proyecto, por compartir sus conocimientos conmigo y el impulso que me otorgó para ver el estudio como un gran tesoro.

Agradezco a la Fundación Universitaria Los Libertadores, por la formación brindada en el desarrollo de mi carrera profesional, por permitirme conocer, compartir y estudiar con grandes compañeros y amigos. De igual manera, agradezco a los docentes que hicieron parte de mi enseñanza, el tiempo dedicado y los consejos que impartieron en mí.

Página dejada en blanco intencionalmente - No forma parte del cuerpo del trabajo.

Índice general

1	Introducción	3
2	Planteamiento del Problema	5
2.1	Objetivos	7
2.1.1	Objetivo General	7
2.1.2	Objetivos Específicos	7
2.2	Justificación	8
3	Marco Teórico	9
3.1	Elementos orbitales	9
3.2	Transferencia de Hohmann	11
3.3	Transferencia de Hohmann bi-elíptica	14
4	Marco Metodológico	19
4.1	Metodología	19
4.1.1	Diseño Metodológico	19
5	Diseño de algoritmo computacional	21
5.1	Procedimiento de programación	21
5.2	Validación de algoritmos creados	22
6	Resultados y Análisis	25
6.1	Cálculo de elementos orbitales para las trayectorias de transferencia.	26
6.1.1	Transferencia de Hofmann.	26
6.1.2	Transferencia de Hohmann Bi-elíptica.	27
6.1.3	Análisis de tiempos de vuelo e incrementos de energía.	28
6.2	Gráficas de las órbitas de transferencia.	29
7	Conclusiones y Recomendaciones	37
7.1	Recomendaciones	38
	Apéndice A Primer Apéndice	41

Apéndice B Segundo Apéndice **53**

Apéndice C Tercer Apéndice **59**

Índice de figuras

3.1	Elementos Orbitales.	9
3.2	Transferencia de Hohmann	11
3.3	Transferencia de Hohmann entre órbitas elípticas coaxiales.	13
3.4	Gráficas de contorno $\Delta v_{Total})3'$ y $\Delta v_{Total})3$ para diferentes tamaños relativos en la Figura 3.3.	15
3.5	Transferencia de Hohmann Bi-elíptica entre la órbita interna 1 y la órbita externa 2.	15
3.6	Órbitas para las cuales la transferencia bi-elíptica es menos eficiente o más eficiente que la transferencia de Hohmann.	17
5.1	Diagrama de procedimiento de programación	23
6.1	Transferencias de Hohmann y Hohmann bi-elíptica. En azul y rojo las órbitas de los planetas b y c , respectivamente.	30
6.2	Transferencias de Hohmann y Hohmann bi-elíptica. En azul y rojo las órbitas de los planetas b y d , respectivamente.	31
6.3	Transferencias de Hohmann y Hofmann bi-elíptica. En azul y rojo las órbitas de los planetas b y e respectivamente.	32
6.4	Transferencias de Hohmann y Hohmann bi-elíptica. En azul y rojo las órbitas de los planetas b y f , respectivamente.	33
6.5	Transferencias de Hohmann y Hohmann bi-elíptica. En azul y rojo las órbitas de los planetas b y g , respectivamente.	34
6.6	Transferencias de Hohmann y Hohmann bi-elíptica. En azul y rojo las órbitas de los planetas b y h , respectivamente.	35
A.1	Algoritmo para el cálculo de vectores de estado y elementos orbitales dada una fecha	41
A.2	Algoritmo para el cálculo de vectores de estado y elementos orbitales dada una fecha (continuación 1)	42
A.3	Algoritmo para el cálculo de vectores de estado y elementos orbitales dada una fecha (continuación 2)	43

A.4	Algoritmo para el cálculo de vectores de estado y elementos orbitales dada una fecha (continuación 3)	44
A.5	Algoritmo para el cálculo de vectores de estado y elementos orbitales dada una fecha (continuación 4)	45
A.6	Algoritmo para el cálculo de vectores de estado y elementos orbitales dada una fecha (continuación 5)	46
A.7	Algoritmo para el cálculo de vectores de estado y elementos orbitales dada una fecha (continuación 6)	47
A.8	Algoritmo para el cálculo de vectores de estado y elementos orbitales dada una fecha (continuación 7)	48
A.9	Algoritmo para el cálculo de vectores de estado y elementos orbitales dada una fecha (continuación 8)	49
A.10	Algoritmo para el cálculo de vectores de estado y elementos orbitales dada una fecha (continuación 9)	50
A.11	Algoritmo para el cálculo de vectores de estado y elementos orbitales dada una fecha (continuación 10)	51
A.12	Algoritmo para el cálculo de vectores de estado y elementos orbitales dada una fecha (continuación 11)	52
B.1	Algoritmo para el cálculo de la trayectoria desde un planeta 1 hasta a un planeta 2	53
B.2	Algoritmo para el cálculo de la trayectoria desde un planeta 1 hasta a un planeta 2 (continuación 1)	54
B.3	Algoritmo para el cálculo de la trayectoria desde un planeta 1 hasta a un planeta 2 (continuación 2)	55
B.4	Algoritmo para el cálculo de la trayectoria desde un planeta 1 hasta a un planeta 2 (continuación 3)	56
B.5	Algoritmo para el cálculo de la trayectoria desde un planeta 1 hasta a un planeta 2 (continuación 4)	57
B.6	Algoritmo para el cálculo de la trayectoria desde un planeta 1 hasta a un planeta 2 (continuación 5)	58
C.1	Algoritmo para el calculo de la trayectoria desde un planeta 1 hasta a un planeta 2 realizado por el Autor	59
C.2	Algoritmo para el calculo de la trayectoria desde un planeta 1 hasta a un planeta 2 estipulado en [8] (a)	60
C.3	Algoritmo para el calculo de la trayectoria desde un planeta 1 hasta a un planeta 2 estipulado en [8] (b)	61

Página dejada en blanco intencionalmente - No forma parte del cuerpo del trabajo.

Abreviaturas

θ	Anomalía verdadera
ω	Argumento del perihelio
Ω	Ascensión recta del nodo ascendente
G	Constante gravitacional
h	Momento angular
ε	Energía específica
e	Excentricidad
i	Inclinación respecto al plano de la eclíptica
$\hat{\omega}$	Longitud del perihelio
r_a	Radio de apoapsis
r_p	Radio de periapsis
a	Semieje mayor
t_H	Tiempo de vuelo de transferencia de Hohmann
t_B	Tiempo de vuelo de transferencia de Hohmann Bi-elíptica
ua	Unidades Astronómicas
N	Vector de la línea nodal
t	Variable angular

Página dejada en blanco intencionalmente - No forma parte del cuerpo del trabajo.

Algoritmos computacionales para órbitas de transferencia

Resumen

El cálculo de trayectorias interplanetarias es tema activo de investigación dado el constante envío de sondas hacia los diferentes planetas de nuestro Sistema Solar. En este sentido, este proyecto aborda el desarrollo e implementación de códigos computacionales en el cálculo de trayectorias de transferencia para el envío de sondas interplanetarias, haciendo uso de trayectorias del tipo Hohmann con asistencia gravitacional en pasos intermedios.

En este trabajo se determinan los elementos orbitales y las gráficas para las trayectorias de sondas enviadas desde la Tierra hacia algunos planetas del Sistema Solar, así como otros tipos de sistemas planetarios, corroborando los resultados con misiones ya efectuadas y obteniendo trayectorias de transferencia adecuadas para los planetas en consideración durante un periodo establecido.

Los resultados de esta investigación adquieren relevancia puesto que serán útiles para la comunidad científica interesada en futuras misiones interplanetarias, entre ellas las diferentes agencias espaciales internacionales.

Palabras claves: Asistencia gravitacional, elementos orbitales, transferencia de Hohmann.

Página dejada en blanco intencionalmente - No forma parte del cuerpo del trabajo.

Algoritmos computacionales para órbitas de transferencia

Abstract

The calculation of interplanetary trajectories is an active topic of research given the constant sending of probes to the different planets of our Solar System. In this sense, this project addresses the development and implementation of computational codes in the calculation of transfer paths for sending interplanetary probes, making use of Hohmann-type paths with gravitational assistance in intermediate steps.

In this work, the orbital elements and the graphs for the trajectories of probes sent from the Earth to some planets of the Solar System, as well as other types of planetary systems are determined, corroborating the results with missions already carried out and obtaining adequate transfer paths for the planets under consideration for a set period.

The results of this research acquire relevance since they will be useful for the scientific community interested in future interplanetary missions, including the different international space agencies.

Keywords: Gravitational assistance, orbital elements, Hohmann transfer.

Capítulo 1

Introducción

En el ámbito espacial es de gran relevancia reducir el gasto energético proporcionado a un vehículo, por ende, es necesario realizar un estudio del cálculo de los parámetros orbitales para el envío de sondas. En [1] se estudia una aproximación algebraica al problema de la transferencia orbital minimizando su costo. Los autores desarrollan una prueba teórica con el fin de optimizar la doblemente impulsada transferencia de Hohmann entre dos trayectorias circulares. En [2] los autores abordan el problema de maniobras orbitales para un satélite moviéndose en trayectorias Keplerianas, en presencia de un campo de fuerzas centrales, de esta manera determinan los resultados teóricos apropiados para simulaciones reales y análisis de misiones.

De igual manera, es importante relacionar el trabajo en [3] acerca del uso de la asistencia gravitacional en misiones planetarias. En este se estudia la factibilidad de efectuar un cambio neto en la velocidad y la dirección del movimiento de un vehículo espacial como consecuencia de su paso a través de la influencia ejercida por el campo gravitacional de un planeta o satélite planetario. También describe el encuentro en diciembre de 1973 entre la nave espacial Pioneer 10 con el planeta Júpiter como un ejemplo concreto de asistencia gravitacional. La referencia [4] contiene el estudio de la dinámica asociada al concepto de asistencia gravitacional, utilizando como ejemplo el envío de la sonda Cassini en la década de los 90.

La referencia [5] explora órbitas elípticas de transferencias Bi-impulsivas no-coplanares con consumo mínimo de energía. Análogamente, en el reciente trabajo de la referencia [6] se utilizan órbitas de transferencia del tipo Hohmann no-coplanares, por medio del uso de ángulos los cuales son precisados como parámetros para minimizar la energía. Los autores validan su método mediante simulación numérica, corroborando sus resultados con parámetros orbitales de los satélites Sputnik I and Vanguard I. Asimismo, en [7] se construye un nuevo tipo de transferencia orbital desde la Tierra hasta Marte que finaliza en captura balística. Como resultado se muestran ahorros substanciales en Δv , comparados con las transferencias clásicas de Hohmann bajo ciertas condiciones establecidas.

Finalmente, uno de los referentes metodológicos para este proyecto se encuentra en la referencia [8] , los investigadores desarrollan algoritmos para el cálculo de parámetros orbitales para el envío de sondas interplanetarias en código MatLab. Sin embargo, no se establece consideraciones pertinentes a la asistencia gravitacional.

Capítulo 2

Planteamiento del Problema

El lanzamiento de una sonda de investigación planetaria a través del Sistema Solar implica una serie de análisis que incluyen telemetría, sistemas de posicionamiento, ventanas de lanzamiento adecuadas, entre otros [9],[10] y [11]. Una de las variables críticas y relevantes para lograr poner en órbita una sonda alrededor de un planeta es encontrar una correcta trayectoria de transferencia, para lo cual a menudo se hace uso de trayectorias de transferencias del tipo Hohmann y no- Hohmann. Particularmente, el uso de la transferencia Hohmann ha sido extensamente implementada en el envío de sondas interplanetarias [12]. Bajo ciertas consideraciones de lanzamiento y con el uso de asistencia gravitacional [13], es posible optimizar el camino recorrido por la sonda, permitiendo ahorros de tiempo y costo en la misión.

Este trabajo busca desarrollar e implementar códigos computacionales para calcular las trayectorias de lanzamiento para el envío de una sonda desde la tierra hacia diferentes planetas del Sistema Solar, haciendo uso de trayectorias del tipo Hohmann y No-Hohmann, y asistencia gravitacional en pasos intermedios. Lo anterior requiere encontrar los elementos orbitales y las gráficas y de las trayectorias para el envío de sondas desde la Tierra hacia algunos planetas del Sistema Solar. Para este fin se hace necesario realizar un estudio detallado de los vectores de estado de la posición y velocidad orbital de los planetas considerados con el fin de determinar las trayectorias de transferencia óptimas para el envío de la sonda.

Así, se pretende desarrollar e implementar códigos en el paquete computacional Matlab para obtener los elementos orbitales de las misiones Mariner 10 (Mercurio), Magallanes (Venus), Mars Global Surveyor (Marte) y Voyager 1 (Jupiter y Saturno), como ejercicio de corroboración de los algoritmos. Este ejercicio dará validez a nuestro código computacional permitiendo el uso del mismo para futuros trabajos, donde se requiera como procedimiento encontrar los elementos orbitales y las simulaciones de trayectorias de misiones interplanetarias futuras. Según lo estipulado anteriormente, se formula siguiente pregunta de investigación ¿Es posible desarrollar de manera factible un Algoritmo computacional

en Matlab que permita obtener los elementos orbitales para la trayectoria de una nave espacial en viaje interplanetario?

2.1 Objetivos

2.1.1 Objetivo General

Desarrollar un algoritmo computacional en Matlab que permita obtener los elementos orbitales para la trayectoria de una nave espacial en viaje interplanetario, haciendo uso de trayectorias del tipo Hohmann y asistencia gravitacional.

2.1.2 Objetivos Específicos

- Sintetizar las ecuaciones de movimiento involucradas en una órbita de transferencia de tipo Hohmann con asistencia gravitacional.
- Desarrollar funciones en Matlab para las ecuaciones de movimiento.
- Desarrollar scripts en Matlab que calculen los elementos orbitales.
- Corroborar los resultados obtenidos con dos paquetes computacionales utilizados por la comunidad científica

2.2 Justificación

En el marco del proyecto de investigación *Cálculo de elementos orbitales para el envío de sondas interplanetarias*, código interno ING-003-18, y dentro de las actividades del Semillero de Investigación en Mecánica Orbital del Programa de Ingeniería Aeronáutica, surge la oportunidad de contribuir al desarrollo del proyecto a través de la modalidad de Trabajo de Grado de los estudiantes. En este sentido, el desarrollo de este trabajo contribuye a los resultados de investigación del Proyecto en mención, al tiempo que fortalece las competencias de formación investigativa de los estudiantes del programa.

Desde lo disciplinar, el estudio de las órbitas de transferencia adquiere relevancia dado el constante envío de sondas para la exploración planetaria, entre ellas las desarrolladas por las agencias internacionales NASA (National Aeronautics and Space Administration), ESA (European Space Agency), State Space Corporation ROSCOSMOS, y JAXA (Japan Aerospace Exploration Agency). Los resultados de trabajos de estas características son de constante consulta por los investigadores en el campo de la astronáutica para el desarrollo de futuras misiones.

Por otra parte, la determinación de órbitas de transferencia para el envío de sondas interplanetarias requiere el uso de paquetes computacionales que funcionan a modo de “cajas negras”, donde el usuario frecuentemente no tiene la oportunidad de conocer la física involucrada en los resultados que arroja el programa. Estos softwares funcionan a modo de entrada-salida, donde el usuario no tiene acceso al *kernel* de las ecuaciones que dan origen a los resultados que arroja el paquete computacional. En este sentido, este trabajo de grado desarrolla rutinas computacionales desde la misma programación de las ecuaciones de movimiento. El resultado final son códigos computacionales que permiten obtener los elementos orbitales para órbitas de transferencia tipo Hohmann con asistencia gravitacional.

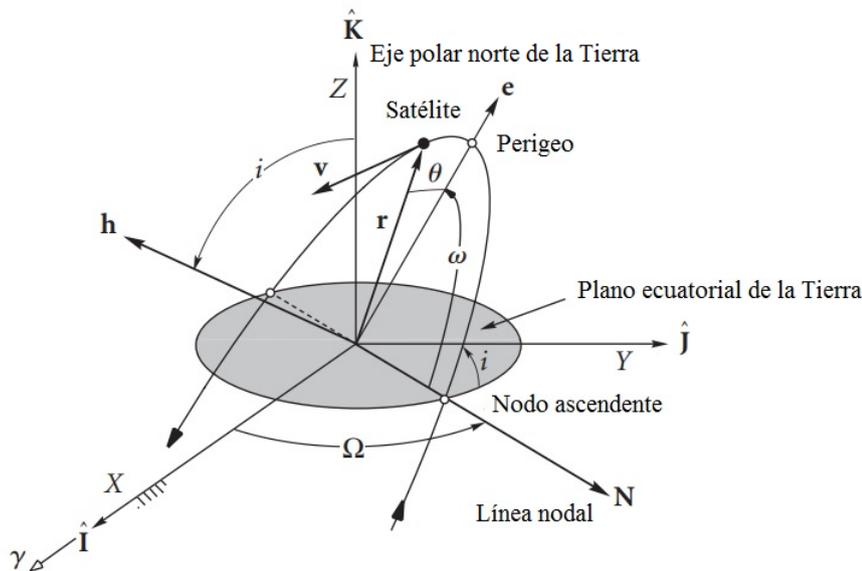
Capítulo 3

Marco Teórico

3.1 Elementos orbitales

Los elementos orbitales son los parámetros necesarios para precisar una órbita en un plano requerido. La excentricidad y el momento angular son usados para definir otros parámetros como lo son el semieje mayor, el periodo y la energía específica (en una elipse). La anomalía verdadera es un parámetro que permite ubicar un punto en la órbita al transcurrir un tiempo específico desde el perigeo.

Figura 3.1: Elementos Orbitales.



Fuente:[8]

Tres parámetros adicionales son requeridos para la orientación de una órbita en tres dimensiones, los cuales tienen por nombre ángulos de Euler. En primer lugar, se localiza

Cuadro 3.1: Elementos orbitales de la Figura 3.1

Elemento orbital	Descripción (unidad de medida)
h	Momento angular (km^2/s)
e	Excentricidad (adimensional)
Ω	Ascensión recta del nodo ascendente ($^\circ$)
i	Inclinación respecto al plano de la eclíptica ($^\circ$)
ω	Argumento del perihelio ($^\circ$)
θ	Anomalía verdadera ($^\circ$)
a	Semieje mayor (km)

la intersección entre el plano de la órbita y el plano Ecuatorial (XY), determinando de esta manera la línea nodal, el vector de la línea nodal \mathbf{N} tiene una prolongación hacia afuera desde el origen y hasta punto llamado nodo ascendente, lugar en donde la órbita pasa por encima del plano ecuatorial desde abajo. En el otro extremo de la línea nodal se encuentra el nodo descendente, punto en el cual la órbita pasa por debajo del plano ecuatorial. El primer ángulo de Euler, ascensión recta del nodo ascendente (Ω), se define como el ángulo entre el eje positivo X y la línea nodal, teniendo en cuenta que este criterio es un número positivo entre 0° y 360° .

La inclinación (i) es segundo ángulo de Euler, parámetro definido como el ángulo diedro entre el plano orbital y el plano ecuatorial, su medida es estipulada por la regla de la mano derecha, en sentido antihorario, alrededor del vector de la línea nodal desde el ecuador hasta la órbita, toma valores entre 0° y 180° . Una manera equivalente de medir el parámetro i es con el ángulo formado entre el eje Z y \mathbf{h} .

Finalmente, el argumento del perigeo (ω) es el tercer ángulo de Euler, es el ángulo entre el vector de la línea nodal \mathbf{N} y el vector excentricidad \mathbf{e} , medido en el plano de la órbita, este parámetro toma valores positivos entre 0° y 360° . Se debe considerar que el perigeo se encuentra en la intersección del vector excentricidad \mathbf{e} con la trayectoria orbital.

Igualmente, es necesario realizar un estudio concerniente a las trayectorias de transferencia elípticas entre órbitas circulares, debido a ello, se hace preciso escribir la ecuación de la elipse en coordenadas polares. Así, si r es el radio polar medido en unidades astronómicas (ua), y se ubica el origen de coordenadas en uno de los focos de la elipse, se tiene que:

$$r = \frac{a(1 - e^2)}{1 + e \cos t} \quad (3.1)$$

donde a es el semieje mayor (en ua); e es la excentricidad, cuyo valor está en el rango

$0 < e < 1$; y t es la variable angular medida en radianes, $0 \leq t < 2\pi$.

Por otra parte, el semieje mayor a de la órbita de transferencia entre dos órbitas con apoapsis (distancia del foco al punto más alejado de la elipse) r_1 y r_2 , se escribe:

$$a = \frac{1}{2}(r_1 + r_2) \quad (3.2)$$

En último lugar, la excentricidad e , en términos del semieje mayor a está dada por:

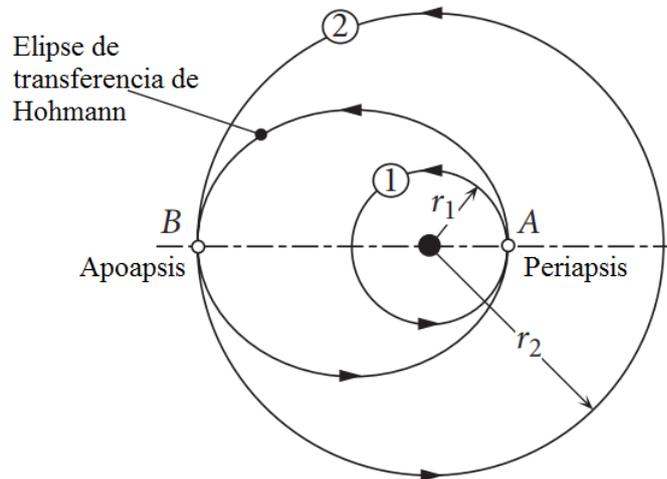
$$e = \frac{r_a}{a} - 1 \quad (3.3)$$

siendo r_a el apoapsis

3.2 Transferencia de Hohmann

La transferencia de Hohmann es la transferencia con mejor eficiencia energética para transferir un impulso entre dos orbitas circulares coplanares que comparten una perspectiva en común [8]. La transferencia de Hohmann se considera como una órbita elíptica, la cual es tangente a los círculos en su línea del ábside. La Figura 3.2, establece el periapsis y apoapsis de la órbita elíptica de transferencia, los cuales son los radios internos y externos de las órbitas circulares, respectivamente.

Figura 3.2: Transferencia de Hohmann



Fuente:[8]

La maniobra de vuelo se da solo en la mitad de la elipse, puede ocurrir en cualquier dirección, desde la órbita circular interna hasta la órbita circular externa, o de manera recíproca. Se debe considerar, que la energía de la órbita de transferencia depende solamente de su semieje mayor (a). La energía específica de una órbita elíptica es negativa,

se denota por la siguiente expresión:

$$\varepsilon = -\frac{\mu}{2a} \quad (3.4)$$

Para obtener un aumento de energía es necesario reducir su magnitud, para que ε sea menos negativa. Por consiguiente, entre mayor sea el semieje mayor, la energía que presenta la órbita tiene un aumento significativo. En la Figura 3.2 se denota el aumento de energía a medida que se da el movimiento desde el círculo interior hacia la exterior. En la órbita interior "A" se requiere un incremento en la velocidad Δv_A , el impulso del vehículo debe darse en la dirección del vuelo, con el fin de dirigirlo hacia la trayectoria elíptica de mayor energía. Posterior al desplazamiento realizado de "A" hasta "B", es indispensable generar otro incremento de velocidad de avance Δv_B , posicionando al vehículo en la órbita externa, la cual presenta una energía superior. Si no se genera el último delta-v, el vehículo retornaría hasta "A" por medio de la elipse de transferencia de Hohmann. El gasto total de energía se comprende con el requerimiento del delta-v total, el cual se denomina como $\Delta v_{Total} = \Delta v_A + \Delta v_B$.

La expresión de delta-v, es necesaria si la transferencia comienza en "B" en la órbita circular externa, ya que, es indispensable reducir la energía del vehículo espacial hacia la órbita circular interna. Lo que se pretende con ello, es realizar un empuje en el sentido opuesto a la dirección del vuelo, con el fin de frenar el movimiento. El Δv representa la misma cantidad de consumo de combustible sin importar la dirección de propulsión, debido a ello, es de mayor consideración las magnitudes al sumar los Δv_s .

Ninguna órbita existente es perfectamente circular, por lo tanto, se deben considerar las orbitas anteriormente mencionadas como elípticas, de este modo se expresa que las órbitas tienen en común la línea del ápside, es decir son coaxiales. En la Figura 3.3, se demuestra que la elipse de transferencia debe ser tangente al punto inicial y el objetivo 1 y 2. De igual manera, existen dos órbitas de transferencia denominadas como 3 y 3', para determinar cuál de estas requiere menor gasto energético es necesario obtener el delta-v total. Para ello, el cálculo de las velocidades en los puntos A, A', B y B' es imprescindible en cada órbita con estos puntos en común. En primer lugar, se determina la excentricidad en una órbita elíptica, por medio del radio de periapsis r_p y el apoapsis r_a mediante la siguiente ecuación.

$$e = \frac{r_a - r_p}{r_a + r_p} \quad (3.5)$$

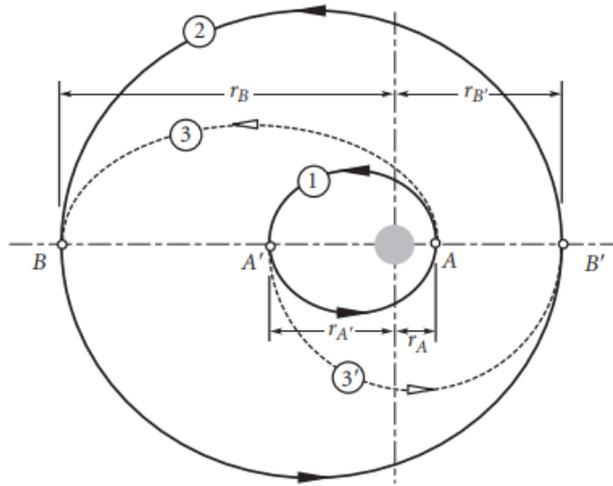
r_a y r_p son los radios del apoapsis y el periapsis, respectivamente. El radio del periapsis es evaluado en la ecuación de la órbita.

$$r_p = \frac{h^2}{\mu} \frac{1}{1+e} = \frac{h^2}{\mu} \frac{1}{1+2r_a} \quad (3.6)$$

El momento angular es obtenido resolviendo la anterior expresión, el cual se encuentra determinado por los radios del apoapsis y periapsis.

$$h = \sqrt{2\mu} \sqrt{\frac{r_a r_p}{r_a + r_p}} \quad (3.7)$$

Figura 3.3: Transferencia de Hohmann entre órbitas elípticas coaxiales.



Fuente:[8]

En esta figura, $r_{A'}/r_0 = 3$, $r_B/r_0 = 8$ y $r_{B'}/r_0 = 4$

La ecuación 3.7 es utilizada para evaluar el momento angular en cada uno de las cuatro órbitas demostradas en la Figura 3.3.

$$h_1 = \sqrt{2\mu} \sqrt{\frac{r_A r_{A'}}{r_A + r_{A'}}} \quad (3.8)$$

$$h_2 = \sqrt{2\mu} \sqrt{\frac{r_B r_{B'}}{r_B + r_{B'}}} \quad (3.9)$$

$$h_3 = \sqrt{2\mu} \sqrt{\frac{r_{A'} r_B}{r_{A'} + r_B}} \quad (3.10)$$

$$h_4 = \sqrt{2\mu} \sqrt{\frac{r_{A'} r_{B'}}{r_{A'} + r_{B'}}} \quad (3.11)$$

Por último, el requerimiento de delta-v total para las dos posibles trayectorias de transferencia.

$$\Delta v_{Total)3} = \Delta v_A + \Delta v_B \quad (3.12)$$

$$\Delta v_{Total)3'} = \Delta v_{A'} + \Delta v_{B'} \quad (3.13)$$

Una vez determinadas los requerimientos para cada una de las trayectorias de transferencia, se establece que si $\Delta v_{Total)3'}$ y $\Delta v_{Total)3}$ 1, la órbita 3 es la más eficiente. Sin embargo, si $\Delta v_{Total)3'}$ y $\Delta v_{Total)3}$ 1, la órbita 3' tiene mayor eficiencia que la órbita 3.

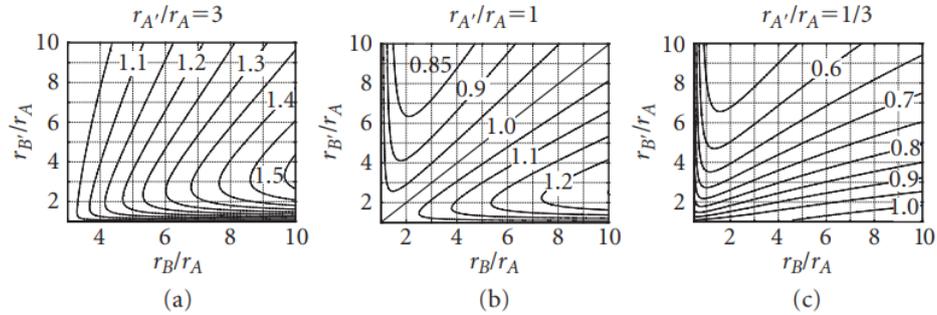
Existen tres gráficos de contorno de $\Delta v_{Total)3'}/\Delta v_{Total)3}$, expresados para tres representaciones diferentes de la órbita interna 1 de la Figura 3.3. La Figura 3.4 (a) es presentada para $r_{A'}/r_A = 3$, situación en la que el punto "A" es el periapsis de la elipse inicial, caso demostrado en la Figura 3.3. En la Figura 3.4 (b) para $r_{A'}/r_A = 1$, expresando de esta manera que la elipse inicial es un círculo. Finalmente, en la Figura 3.4 (c) para $r_{A'}/r_A = 1/3$, representando de esta manera a una órbita similar a la Figura 3.3, pero su punto "A" es considerado como el apoapsis en cambio de periapsis.

La Figura 3.4 (a), para la cual $r_{A'} > r_A$, implica que si el punto "A" es el periapsis de la órbita 1, entonces la transferencia de la órbita 3 es la más eficiente. La Figura 3.4 (c), para la cual $r_{A'} < r_A$, muestra que si el punto "A" es el periapsis de la órbita 1, entonces la transferencia de la órbita 3 es la más eficiente. Según los resultados obtenidos, se determina que la órbita de transferencia con inicio en el periapsis en la órbita interna 1 tiene mayor eficiencia, independientemente de la forma de la órbita de destino externa, puesto que su energía cinética es superior. Si la órbita inicial es un círculo como lo demuestra la Figura 3.4 (b), la transferencia de la órbita 3 es más eficiente, si se cumple $r_{B'} > r_B$. Es decir, desde una órbita circular interna, la elipse de transferencia debe terminar en la apoapsis de la elipse del objetivo externo, donde la velocidad es menor.

3.3 Transferencia de Hohmann bi-elíptica

A diferencia de la transferencia de Hohmann la transferencia de Hohmann bi-elíptica utiliza dos órbitas de transferencia elípticas para llegar a la órbita final y, por tanto, se necesitan tres impulsos en vez de dos.[14]

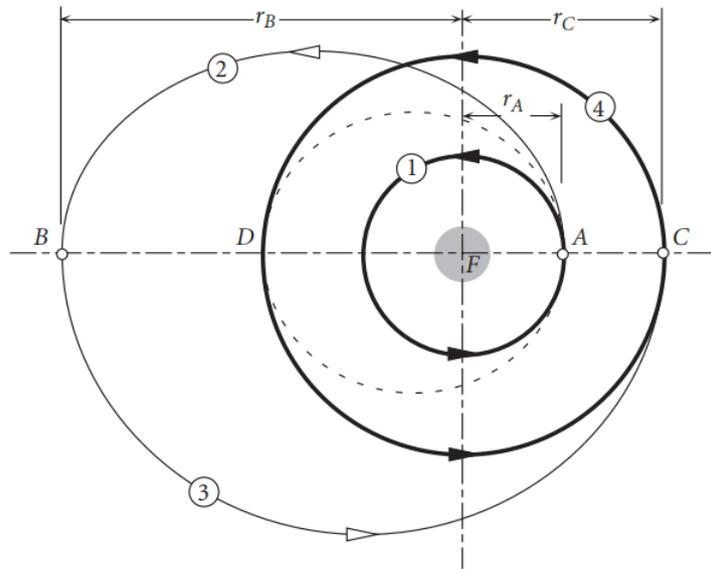
Figura 3.4: Gráficas de contorno Δv_{Total} para diferentes tamaños relativos en la Figura 3.3.



Fuente:[8]

Teniendo en cuenta que $r_B > r_{A'}$ y $r_{B'} > r_A$

Figura 3.5: Transferencia de Hohmann Bi-elíptica entre la órbita interna 1 y la órbita externa 2.



Fuente:[8]

En la Figura 3.5 se observa la transferencia de Hohmann, correspondiente a la elipse formada por segmentos, es tangente a la circunferencia 1 y la circunferencia 4 e inicia desde la órbita 1 (interna) hasta la órbita 4 (externa). La transferencia bi-elíptica de Hohmann utiliza dos semi-elipses coaxiales, 2 y 3, que se extienden más allá de la órbita del objetivo externo y son tangentes a cada una de las órbitas circulares. De igual manera, son contiguas entre sí en B, punto del apoapsis de ambas semi-elipses. El punto B debe estar ubicado lo suficientemente lejos del foco, con el fin de que Δv_B sea muy pequeña. Cuando r_B se acerca al infinito, v_B se acerca a cero. Para que el esquema bi-elíptico sea más eficiente energéticamente que la transferencia de Hohmann, debe ser cierto que.

$$\Delta v_{total, Bi-elíptica} < \Delta v_{total, Hohmann} \quad (3.14)$$

Los análisis de Delta-v de las transferencias de Hohmann y bi-elípticas conllevan a los siguientes resultados.

$$\Delta v)_{Hohmann} = \left[\frac{1}{\sqrt{\alpha}} - \frac{\sqrt{2}(1-\alpha)}{\sqrt{\alpha}(1+\alpha)} - 1 \right] \sqrt{\frac{\mu}{r_A}} \quad (3.15)$$

donde $\alpha = r_C/r_A$, siendo G la constante gravitacional y m_p la masa del planeta asociado a la trayectoria de salida.

Por otra parte, el tiempo de vuelo t_H desde el punto A hasta el punto B se calcula mediante;

$$t_H = \frac{1}{2} \left(\frac{2\pi}{\sqrt{\mu}} a^{3/2} \right) \quad (3.16)$$

siendo a el semieje mayor de la órbita de transferencia de Hohmann.

El incremento energético para una transferencia bi-elíptica se escribe,

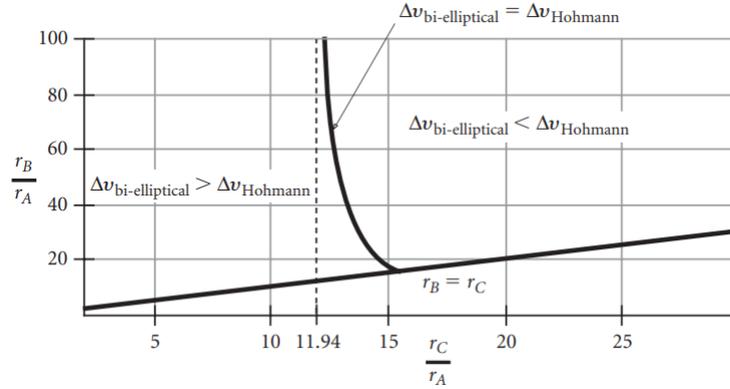
$$\Delta v_{Bi-elíptica} = \left[\sqrt{\frac{2(\alpha+\beta)}{\alpha\beta}} - \frac{1+\sqrt{\alpha}}{\sqrt{\alpha}} - \sqrt{\frac{2}{\beta(1+\beta)}}(1-\beta) \right] \sqrt{\frac{\mu}{r_A}} \quad (3.17)$$

Los parámetros α , β y μ , se definen de la siguiente manera.

$$\alpha = \frac{r_C}{r_A}; \beta = \frac{r_B}{r_A}; \mu = G \cdot m_p. \quad (3.18)$$

La gráfica de comparación entre Δv_{total} Hohmann y bi-elíptico en función de α y β expuesta en la Figura 3.6, muestra las regiones en las que la diferencia es positiva, negativa y cero.

Figura 3.6: Órbitas para las cuales la transferencia bi-elíptica es menos eficiente o más eficiente que la transferencia de Hohmann.



Fuente:[8]

La órbita de transferencia de Hohmann es de mayor eficiencia, si el radio r_C del objetivo exterior sea menor 11.9 veces que el radio del objetivo interior r_A , si dicha proporción supera los 15, la órbita de transferencia con mayor eficiencia es la bi-elíptica. Igualmente, se evidencia que los valores más grandes del radio de apoapsis r_B son de mayor beneficio para la transferencia bi-elíptica para las dos relaciones (α y β), en contraposición los valores menores favorecen a la transferencia de Hohmann.

Las pequeñas ganancias en eficiencia energética pueden percibirse más que compensadas por los tiempos de vuelo mucho más largos alrededor de las trayectorias bi-elípticas en comparación con el tiempo de vuelo en la semi-elipse única de la transferencia de Hohmann. El tiempo de vuelo t_B desde el punto A hasta el punto C se calcula mediante:

$$t_B = \frac{1}{2} \left(\frac{2\pi}{\sqrt{\mu}} a_2^{3/2} + \frac{2\pi}{\sqrt{\mu}} a_3^{3/2} \right) \quad (3.19)$$

siendo $a_{2,3}$ los semiejes mayores de las órbitas de transferencia bi-elíptica.

Página dejada en blanco intencionalmente - No forma parte del cuerpo del trabajo.

Capítulo 4

Marco Metodológico

4.1 Metodología

Este trabajo es realizado bajo un estudio exploratorio, según Sampieri [15] este tipo de estudio pretende examinar un problema o tema que presenta pocos estudios, es novedoso o se desea indagar los temas desde áreas de nuevas perspectivas, este tipo de estudio sirve para llevar a cabo una investigación de manera completa y así sugerir postulaciones para futuras investigaciones. De igual manera, el trabajo es planteado bajo una metodología de investigación aplicada [16] , la cual tiende a solucionar los problemas o desarrollar ideas, con el objetivo de conseguir innovación y mejoras de procesos en un plazo corto o medio según sea determinado. Este tipo de metodología fue adoptada en este trabajo; ya que, tiene como característica para su aplicabilidad la utilización de conocimiento en bibliografía ya existente, por medio de las fuentes, analizar, sistematizar y emplearlo en el trabajo para el desarrollo del mismo y en cumplimiento a los objetivos planteados.

4.1.1 Diseño Metodológico

A continuación, es presentado el diseño metodológico, el cual ha sido realizado con base a los parámetros establecidos mediante el estudio y metodología anteriormente estipulados, para de esta manera desarrollar los objetivos planteados.

Etapa 1: Fundamentación teórica por medio de bases de datos especializadas, junto con el estudio de la física y la matemática involucrada en las trayectorias de Hohmann con asistencia gravitacional.

Etapa 2: Estudio y escritura de generación de funciones en Matlab, implementando las ecuaciones de movimiento adaptándolas a este lenguaje de programación.

Etapa 3: Estudio de generación e integración de los Scripts y Funciones en Matlab para la implementación de los mismos en el trabajo.

Etapa 4: Comparación de resultados con paquetes existentes, para una posterior validación del proyecto.

Capítulo 5

Diseño de algoritmo computacional

Este capítulo tiene como finalidad explicar el método usado para la realización de los algoritmos computacionales para órbitas de transferencias, mediante la sintetización y posterior programación de las ecuaciones estudiadas e indicadas en el Marco Teórico, lo anterior usando el lenguaje de programación estipulado por el software Matlab y haciendo uso de la interfaz AppDesigner [17].

5.1 Procedimiento de programación

Teniendo en cuenta el procedimiento de fundamentación y creación de códigos en Matlab expuesto en [18], se realiza la generación de código para la creación de los algoritmos computacionales para órbitas de transferencia. En la Figura 5.1, se presenta el proceso que se debe llevar a cabo en dicha programación.

La interfaz de usuario genera una facilidad en la selección de los planetas que se desean estudiar mediante el despliegue de un menú de opciones (como ejemplo son expuestos los planetas del sistema solar). Una vez determinado el planeta de salida para la misión de estudio, se introduce la fecha deseada del envío del vehículo espacial, esta variable es compuesta por *Año*, *Mes*, *Día*, *Hora*, *Minuto* y *Segundo*, análogamente se realiza el procedimiento mencionado para el planeta de llegada del vehículo. Una vez terminado el proceso de selección y escritura de los datos por parte del usuario, se efectúan procedimientos de cálculo de parámetros de gran

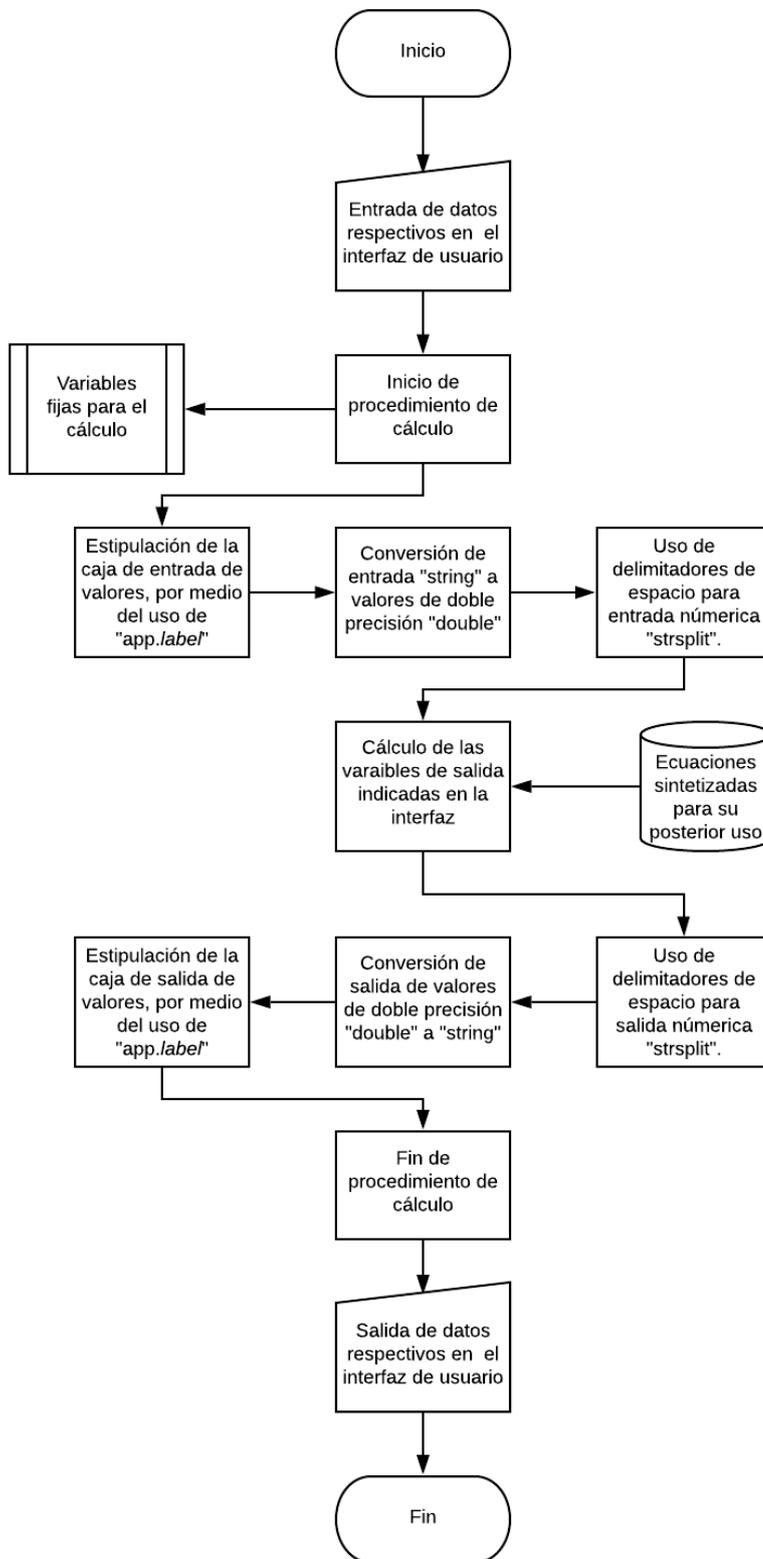
relevancia, tales como el vector posición, vector velocidad y sus respectivas magnitudes para los planetas de salida y llegada de la misión. Asimismo se exponen el vector de velocidad, la magnitud de dicho vector y la velocidad necesaria por parte del vehículo espacial para salir de la esfera de influencia gravitacional que ejercen los planetas sobre este a la salida y llegada de los mismos. Finalmente, son presentados los elementos orbitales de la trayectoria de vuelo desde el planeta de salida hasta el planeta de llegada, también es calculado el tipo de vuelo en días para las fechas estipuladas previamente.

Para visualizar de mejor manera este procedimiento evidenciado en la Figura 5.1, es necesario dirigirse a los apéndices A y B, en los cuales se muestran las ventanas de interfaz de usuario y posteriormente el proceso de programación de cada una de las mismas.

5.2 Validación de algoritmos creados

Una vez desarrollados los algoritmos computacionales para orbitas de transferencia, se debe proceder a una verificación del funcionamiento eficaz de los códigos realizados por medio de otros paquetes computacionales propios de Mecánica Orbital. Para ello, fue tomado como referencia el paquete computacional estipulado en [8], donde por medio de la utilización de uno de sus códigos se contrastó con una misión real llevada a cabo por la Nasa denominada Mars Global Surveyor [19], esta misión tuvo como Fecha de lanzamiento el 7 de noviembre de 1996 y como Fecha de arribo el 12 de septiembre de 1997. Como se evidencia en las Figuras pertenecientes al Apéndice C, no se presentaron errores en el procedimiento de cálculo en ninguna de las variables presentadas para la misión estipulada, validando de esta manera la efectividad de los diversos códigos generados para los algoritmos computacionales para órbitas de transferencia.

Figura 5.1: Diagrama de procedimiento de programación



Fuente: [Autor]

Página dejada en blanco intencionalmente - No forma parte del cuerpo del trabajo.

Capítulo 6

Resultados y Análisis

TRAPPIST-1 es una estrella enana roja del tipo espectral $M8 V$, localizada en la constelación de Acuario, a una distancia de 39.13 años luz. Como sistema planetario fue descubierta en 2015 a través del telescopio TRAPPIST (acrónimo de *Transiting Planets and Small Planets with a Small Telescope*), de ahí el nombre de la estrella [20]. En 2017 se confirmó que este sistema está compuesto de siete planetas terrestres, denotados como *textitb, c, d, e, f, g* y *h*, según la referencia [21],[22].

Cuadro 6.1: Elementos orbitales del Sistema TRAPPIST-I

Planeta	Periodo (d)	Semieje mayor (ua)	Excentricidad	Inclinación ($^{\circ}$)
b	1.510848 ± 0.000019	0.01111	< 0.081	89.65 ± 0.25
c	2.421848 ± 0.000028	0.01522	< 0.083	89.67 ± 0.17
d	4.049610 ± 0.000063	0.021 ± 0.06	< 0.070	89.75 ± 0.16
e	6.099570 ± 0.000091	0.028	< 0.085	89.86 ± 0.11
f	9.20648 ± 0.00053	0.037	< 0.063	89.680 ± 0.034
g	12.35281 ± 0.00044	0.045	< 0.061	89.710 ± 0.025
h	18.76626 ± 0.00068	0.063 ± 0.027	0.086 ± 0.032	89.80 ± 0.10

El Cuadro 6.1 muestra los elementos orbitales de interés del sistema planetario. El marco de referencia utilizado es heliocéntrico, esto es, con la estrella TRAPPIST-1 en uno de los focos. La inclinación de los planetas es medida respecto al plano de la eclíptica terrestre. A continuación, algunos aspectos importantes a resaltar a partir del Cuadro 6.1 y de la referencia [21].

1. Nótese que la excentricidad de las órbitas es relativamente pequeña, siendo el planeta más exterior h el de mayor excentricidad. Esto permitirá en primera aproximación, considerar las órbitas planetarias como si fuesen circulares.
2. De la última columna de la tabla se observa que la inclinación de cada planeta respecto a los demás difiere en menos de 0.3° . Esto significa que con muy buena aproximación las órbitas de los planetas son coplanares, requisito necesario para el estudio de transferencias del tipo Hohmann y Hohmann bi-elípticas que nos conciernen.
3. Adicionalmente, no se tienen datos de la posición del perihelio de los planetas puesto que las observaciones aún no permiten establecer este parámetro orbital. Si la línea de los ápsides es diferente para las órbitas de salida y llegada de la sonda, el estudio de la transferencia de Hohmann y Hohmann bi-elíptico debe ser reformulado. No obstante, para nuestro caso, el hecho de considerar órbitas circulares nos evita este inconveniente.

6.1 Cálculo de elementos orbitales para las trayectorias de transferencia.

Para el cálculo de los elementos orbitales de las trayectorias de transferencia se consideraron lanzamientos desde el planeta más interior b , hacia cada uno de los planetas exteriores. Se hizo uso los datos orbitales del sistema planetario listados en el Cuadro 6.1, así como de la herramienta computacional Mathematica [23] para los cálculos numéricos.

Para nuestro sistema planetario el valor numérico de μ es:

$$\mu = 1,06435 \times 10^{10} \text{ km}^3/\text{s}^2 \quad (6.0)$$

6.1.1 Transferencia de Hofmann.

Los parámetros orbitales a y e , mostrados en las ecuaciones 3.2 y 3.3, respectivamente, son calculados, el incremento $\Delta v_{T-HOHHMANN}$ dado por

3.15, y el tiempo de vuelo t_H son obtenidos a partir de 3.3, para la órbita de transferencia de Hohmann. El Cuadro 6.2 muestra los resultados.

Cuadro 6.2: Elementos orbitales para transferencia de Hohmann

Planeta salida	Planeta llegada	$\Delta v_{T,HOHMANN}$ (km/s)	a (ua)	e	$t_H(d)$
b	c	11,5819	0.013165027	0.56096	0.97412
b	d	21.2846	0.016054974	0.308004	1,31189
b	e	28.146	0.019555024	0.431859	1,76347
b	f	33.2732	0.024055021	0.538142	2,40597
b	g	36.0659	0.028055012	0.603992	3,03038
b	h	39.5245	0.037055006	0.700175	4,59993

6.1.2 Transferencia de Hohmann Bi-elíptica.

De manera análoga al caso anterior, haciendo uso de las expresiones para $a_{2,3}$, y $e_{2,3}$ en 3.3 y 3.12, $\Delta v_{T-BI-ELÍPTICO}$ y t_B son calculadas de 3.17 y 3.19. Aquí $a_{2,3}$, y $e_{2,3}$, denotan los semiejes mayores y las excentricidades, respectivamente, de las porciones de órbita de transferencia bi-elíptica. El Cuadro 6.3 muestra los resultados. Es importante anotar que la columna 3, expresada como el Punto B, indica que se ha tomado como apoapsis (punto B en la Figura 3.5) para la trayectoria bi-elíptica, el radio de órbita del planeta que se indica. Así, por ejemplo, en la segunda fila de la tabla, el punto apoapsis seleccionado corresponde al radio orbital del planeta e . En este trabajo se ha tomado como punto apoapsis el radio orbital del planeta inmediatamente más externo al planeta de llegada de la sonda, como se aprecia en el Cuadro 6.3. Esto se hizo con el fin de dejar abierta la posibilidad de alcanzar el planeta inmediatamente exterior para algún tipo de maniobra de impulso gravitacional, o con fines de exploración planetaria. Para el planeta más externo h , se tomó como apoapsis dos veces su radio de orbital, como se aprecia en la tabla.

Cuadro 6.3: Elementos orbitales para transferencia de Hohmann Bi-elíptica

Planeta salida	Planeta llegada	Punto B	$\Delta v_{T,BI.ELÍPTICA}$ (km/s)	a_2 (ua)	a_3 (ua)	e_2	e_3	$t_B(d)$
b	c	d	21.692	0.016054974	0.018110017	0.308004	0.15958	2,38355
b	d	e	28.4256	0.019555024	0.024500014	0.431859	0.142857	4,23651
b	e	f	33.501	0.024055021	0.032499995	0.538142	0.138462	6,18435
b	f	g	36.1658	0.028055012	0.040999982	0.603992	0.097561	8,38412
b	g	h	39.7824	0.037055006	0.054000033	0.700175	0.166667	12,6922
b	h	2h	43.5001	0.06855512	0.094500008	0.83794	0.333333	30,3094

6.1.3 Análisis de tiempos de vuelo e incrementos de energía.

El Cuadro 6.4 es construido a partir de los Cuadros 6.2 y 6.3, que da cuenta de la relación entre los incrementos de velocidad, así como la relación entre los tiempos de vuelo asociados a cada maniobra de transferencia. Específicamente, en la tercera columna se indica el cociente entre los incrementos de velocidad de la órbita bi-elíptica y la órbita de Hohmann. Obsérvese que en todos los casos este cociente es mayor que 1, tendiendo a ser cada vez más pequeño conforme aumenta la separación orbital entre los planetas involucrados. Esto significa que, en todos los casos de estudio, la órbita de transferencia de Hohmann resulta ser la de mayor eficiencia energética. Como se observa también de el Cuadro 6.4, la menor diferencia entre los dos tipos de transferencia corresponde al lanzamiento de una sonda entre los planetas b y f , con una diferencia de aproximadamente 9 %.

Cuadro 6.4: Relaciones para los incrementos de velocidad y para los tiempos de vuelo

Planeta salida	Punto llegada	$\Delta v_{T,BI.ELÍPTICA}/\Delta v_{T,HOHMANN}$ (km/s)	t_B/t_H
b	c	1.87293	2.96016
b	d	1.3355	3.22932
b	e	1.19026	3.50692
b	f	1.08694	3.48472
b	g	1.10305	4.18833
b	h	1.10059	6.5891

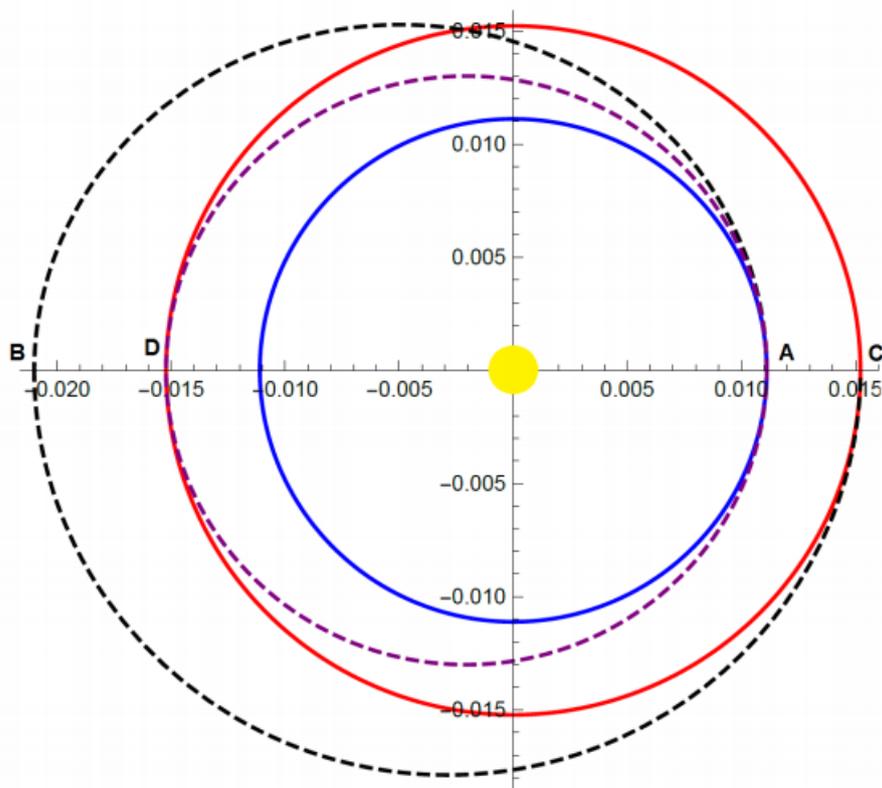
En contraste, el lanzamiento cuya diferencia energética resulta mayor para los dos tipos de transferencia es el asociado a los planetas más interiores b y c , cuya diferencia en los incrementos de velocidad es de alrededor del 87 %. Este análisis es consistente con la teoría desarrollada en el estudio

de estos tipos de transferencia [21], donde se muestra que sólo cuando el cociente entre los radios externos e interno es mayor a 11.9, la transferencia de Hohmann será el mecanismo más eficiente en términos energéticos para efectuar la transferencia. Por otra parte, la cuarta columna de el Cuadro 6.4 muestra los cocientes entre el tiempo de vuelo bi-elíptico y el de Hohmann. Claramente, la trayectoria bi-elíptica emplea mayores tiempos de vuelo, incrementándose con la separación entre las órbitas planetarias, llegando incluso a ser del orden de 6.6 veces mayor para el caso de una sonda enviada desde el planeta b hasta el planeta más exterior h .

6.2 Gráficas de las órbitas de transferencia.

Con los datos reportados en el Cuadro 6.1, y a partir de los elementos orbitales hallados en este trabajo y descritos en los Cuadros 6.2 , 6.3 y 6.4, se procede a generar gráficas de las órbitas de transferencia observadas en las Figuras 6.1 a la 6.6, donde se utilizó un marco de referencia heliocéntrico con respecto a la estrella TRAPPIST-1 y se toma una escala 1:1 donde la unidad de medida son las unidades astronómicas (UA). Cabe resaltar, que a medida que el apoapsis B entre el planeta de salida y el planeta de llegada aumenta la excentricidad e_2 . De igual manera, los semiejes mayor de cada órbita influyen en la forma de las órbitas de transferencia bi-elípticas para los casos dados. La elaboración de estas gráficas fue con el uso del software Mathematica. [23]

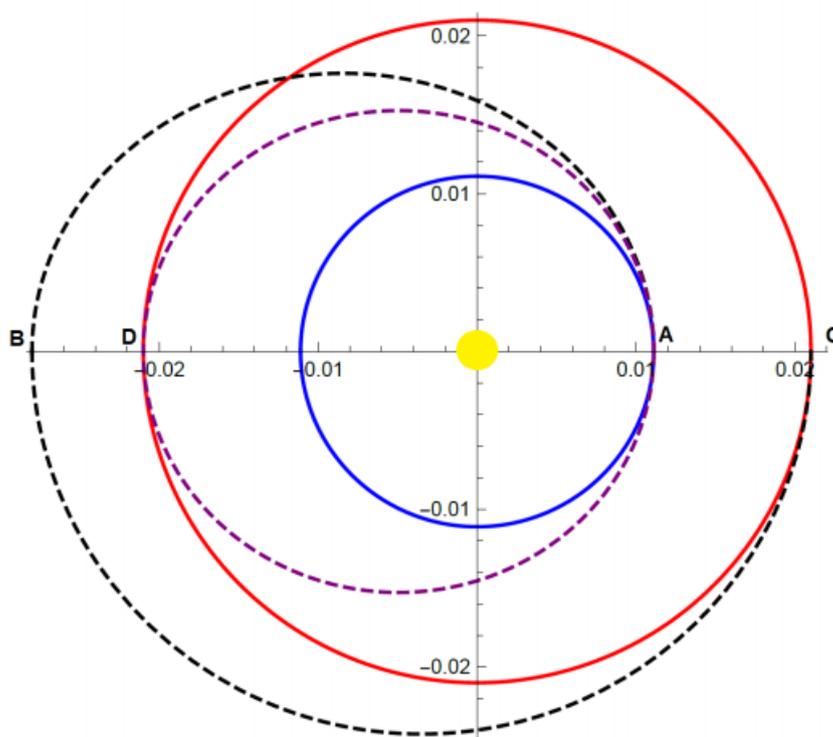
Figura 6.1: Transferencias de Hohmann y Hohmann bi-elíptica. En azul y rojo las órbitas de los planetas b y c , respectivamente.



Fuente: [Autor]

Las curvas punteadas púrpura y negra corresponden a la transferencia de Hohmann y Hohmann bi-elíptica, respectivamente. El apoapsis B de la órbita bi-elíptica corresponde al radio orbital del planeta d . [Escala en (ua)]

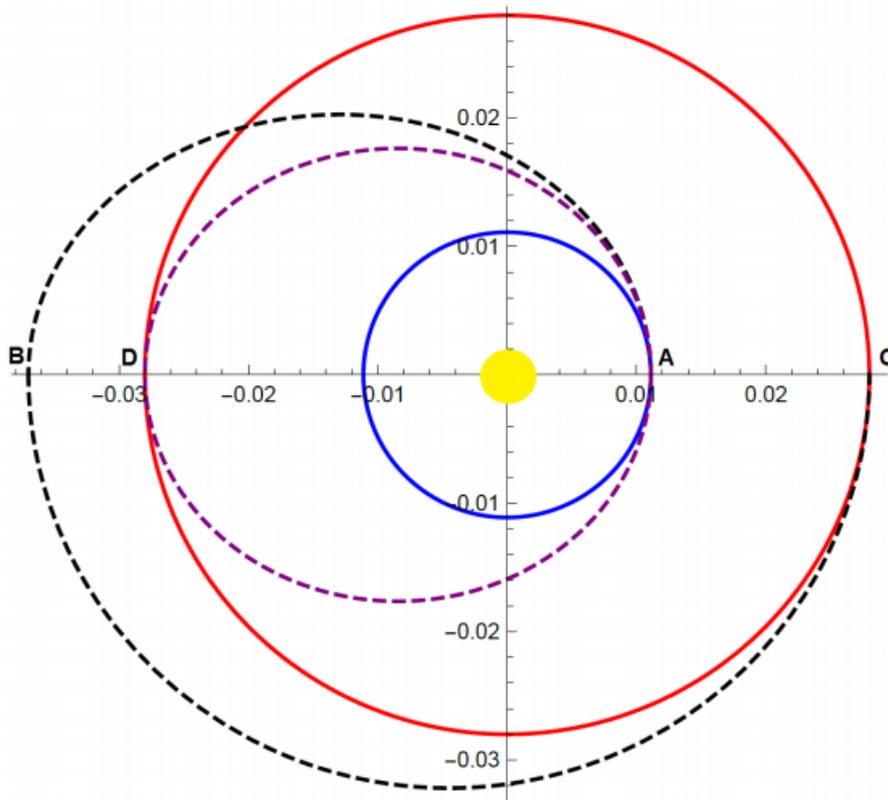
Figura 6.2: Transferencias de Hohmann y Hohmann bi-elíptica. En azul y rojo las órbitas de los planetas b y d , respectivamente.



Fuente: [Autor]

Las curvas punteadas púrpura y negra corresponden a la transferencia de Hohmann y Hohmann bi-elíptica, respectivamente. El apoapsis B de la órbita bi-elíptica corresponde al radio orbital del planeta e . [Escala en (ua)]

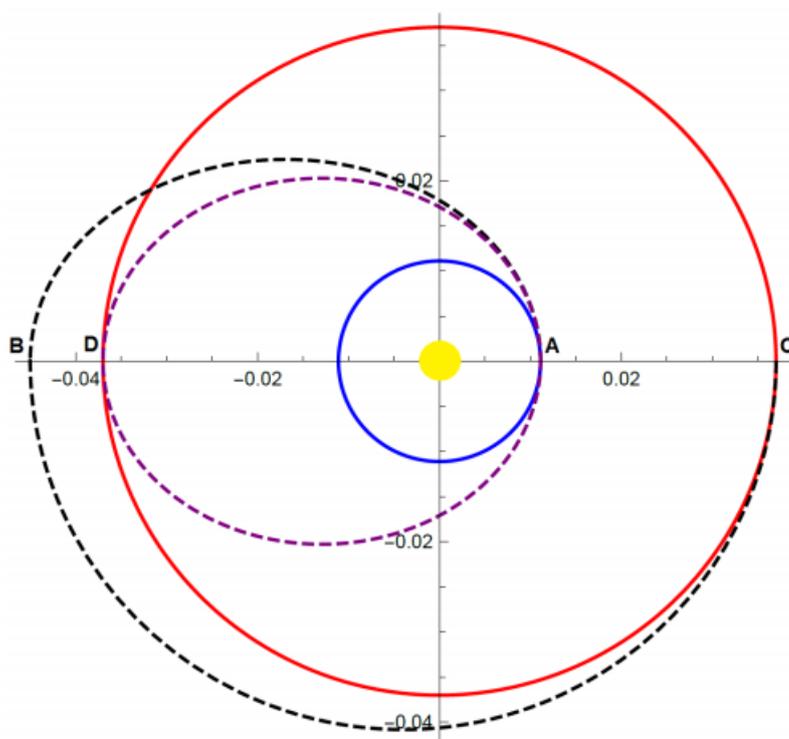
Figura 6.3: Transferencias de Hohmann y Hofmann bi-elíptica. En azul y rojo las órbitas de los planetas b y e respectivamente.



Fuente: [Autor]

Las curvas punteadas púrpura y negra corresponden a la transferencia de Hohmann y Hohmann bi-elíptica, respectivamente. El apoapsis B de la órbita bi-elíptica corresponde al radio orbital del planeta f . [Escala en (ua)]

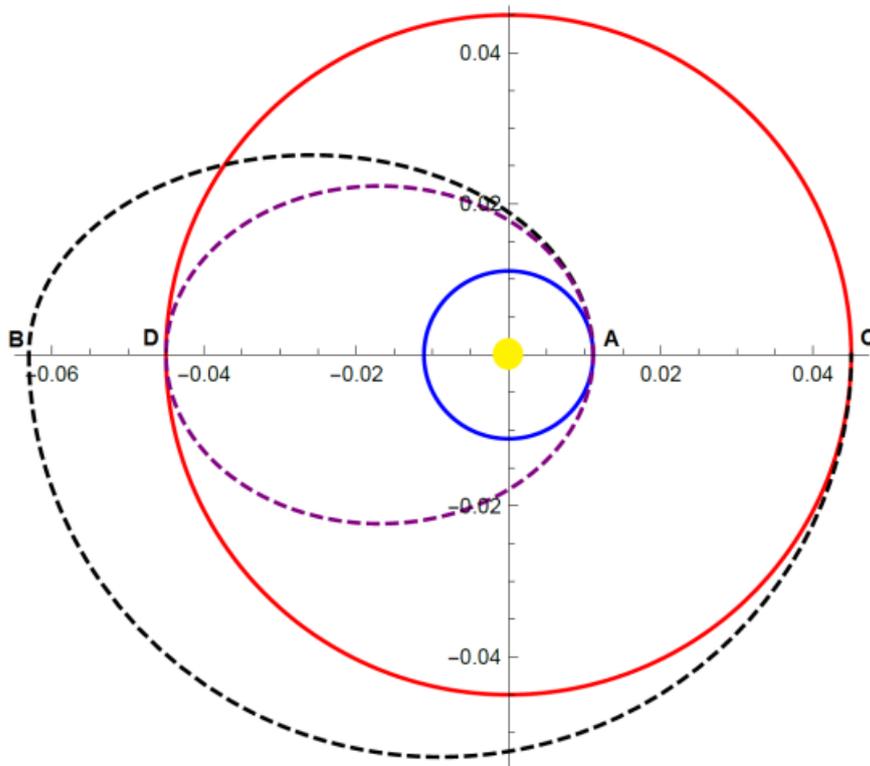
Figura 6.4: Transferencias de Hohmann y Hohmann bi-elíptica. En azul y rojo las órbitas de los planetas b y f , respectivamente.



Fuente: [Autor]

Las curvas punteadas púrpura y negra corresponden a la transferencia de Hohmann y Hohmann bi-elíptica, respectivamente. El apoapsis B de la órbita bi-elíptica corresponde al radio orbital del planeta g . [Escala en (ua)]

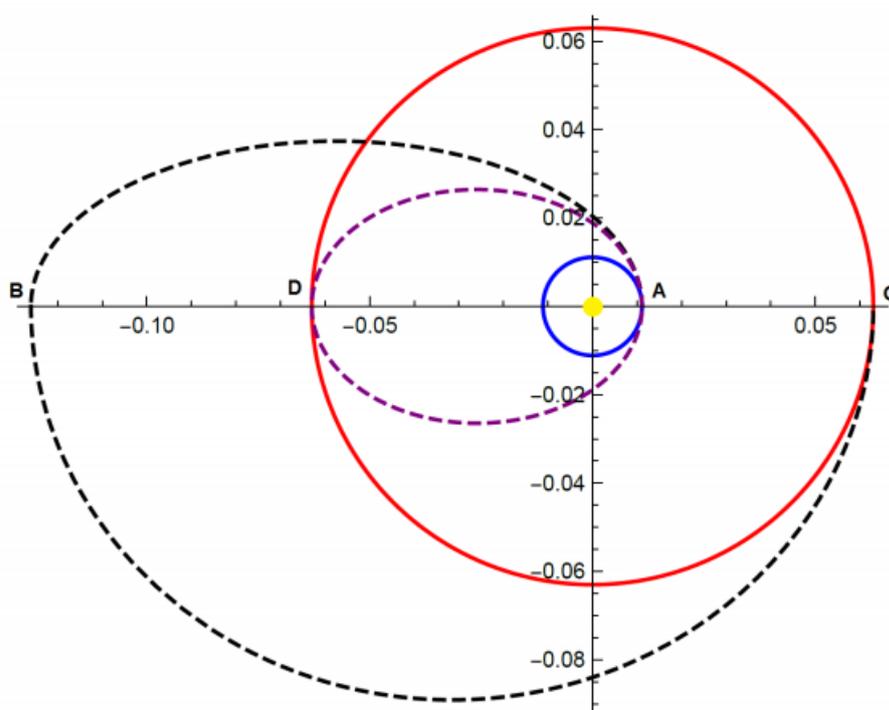
Figura 6.5: Transferencias de Hohmann y Hohmann bi-elíptica. En azul y rojo las órbitas de los planetas b y g , respectivamente.



Fuente: [Autor]

Las curvas punteadas púrpura y negra corresponden a la transferencia de Hohmann y Hohmann bi-elíptica, respectivamente. El apoapsis B de la órbita bi-elíptica corresponde al radio orbital del planeta h . [Escala en (ua)]

Figura 6.6: Transferencias de Hohmann y Hohmann bi-elíptica. En azul y rojo las órbitas de los planetas b y h , respectivamente.



Fuente: [Autor]

Las curvas punteadas púrpura y negra corresponden a la transferencia de Hohmann y Hohmann bi-elíptica, respectivamente. El apoapsis de la órbita bi-elíptica corresponde a dos veces el radio orbital del planeta h . [Escala en (ua)]

Página dejada en blanco intencionalmente - No forma parte del cuerpo del trabajo.

Capítulo 7

Conclusiones y Recomendaciones

Por medio de la realización de este proyecto de grado, se desarrolló una serie de algoritmos computacionales para órbitas de transferencia, a través de su ejecución se pueden obtener parámetros tales como elementos orbitales para la trayectoria de una nave espacial en viaje interplanetario, los vectores de estado de los planetas de salida y llegada y el tiempo de vuelo que va a tomar realizar esta maniobra espacial. Según lo presentado en el objetivo general, encaminado a desarrollar un algoritmo computacional en Matlab que permita obtener los elementos orbitales para la trayectoria de una nave espacial en viaje interplanetario, haciendo uso de trayectorias del tipo Hohmann y asistencia gravitacional, se consumó gracias a lo presentado en el Capítulo 5 y tal como se muestra en los Apéndices A y B.

Las ecuaciones presentadas en el Marco teórico, fueron sintetizadas a través del procedimiento evidenciado en la sección 5.1 perteneciente al Capítulo 5, para así desarrollar los scripts en Matlab de los Algoritmos computacionales pertinentes a trayectorias de Tipo Hohmann y asistencia gravitacional, como es expuesto en los Apéndices A y B.

Mediante la sección 5.2, en la cual se realiza una validación de los algoritmos mediante un proceso comparativo para una misión común denotados en el Apéndice C, se establece que los Algoritmos realizados son eficaces para el calculo de los parámetros requeridos para el calculo de trayectorias de tipo Hohmann y asistencia gravitacional.

Finalmente, estos Algoritmos computacionales fueron utilizados para

dos trabajos expuestos mediante la modalidad de ponencia, el primero de ellos se realizó en la ciudad de París, Francia en la *20 International Conference on Aerospace and Aviation Engineering*, la segunda ponencia se realizó en el Centro de Estudios Aeronáuticos en el *II Encuentro de Investigación, Desarrollo e Innovación en el Sector Aeronáutico*. Adicionalmente, el desarrollo de este proyecto de grado adquiere un valor agregado debido a la creación de Algoritmos computacionales propios, de esta manera se pretende no depender de uso de softwares especializados de alto costo adquisitivo y generando una herramienta para el uso por parte de la institución Universitaria Los Libertadores.

7.1 Recomendaciones

El desarrollo de Algoritmos computacionales en Matlab haciendo uso de trayectorias del tipo Hohmann y asistencia gravitacional, fue realizado con éxito, mostrando procedimientos, datos y cálculos de manera efectiva. Sin embargo, es necesario destacar las siguientes recomendaciones:

- Para el uso de este tipo de Algoritmos computacionales en sistemas como lo es TRAPPIST-I, es relevante ejercer y aprovechar estudios más profundos acerca de la exploración espacial por parte de la comunidad científica hacia este tipo de exoplanetas, como lo es el reciente envío de la sonda Tess por parte de la NASA, ya que en este sistema solo se han podido obtener datos por parte de la sonda Kepler enviada por la misma compañía.
- Establecer parámetros de error de escritura en los códigos, así como un estudio profundo de programación en este tipo de software empleado, ayudaría a crear rutinas computacionales eficientes para posteriormente utilizarlos en trabajos de cálculo de trayectorias de tipo Hohmann y asistencia gravitacional, evitando demoras en la solución de objetivos planteados.

Bibliografía

- [1] A. SHIRAZI. *Spacecraft trajectory optimization: A review of models, objectives, approaches and solutions*. 2018.
- [2] R. ALAMANAC. *El problema de dos cuerpos y las leyes de Kepler*. Universidad de la Rioja, 2016.
- [3] G DI MADURO. *Continuous maneuvers for spacecraft formation flying reconfiguration using relative orbit elements*. Acta Astronáutica, 2018.
- [4] XIN MA. “An overview of the autonomous navigation for a gravity-assist interplanetary spacecraft”. En: *NASA Space Vehicle Desing Criteria (Chemical Propulsion)* (2013).
- [5] W. SCHULZ. *Transferencias Bi-Impulsivas entre Orbitas Elipticas ñao Coplanares com Consumo Minimo de Combustivel*. Space Mechanics y Control Division, National Institute of Space Research (INPE), Brazil, 1997.
- [6] P. ROJO A. RINCON. *On non-coplanar Hohmann transfer using angles as parameters*. Vol. 359. Astrophysics y Space Science, 2015, pág. 1.
- [7] A. POSNER. *The Hohmann–Parker effect measured by the Mars Science Laboratory on the transfer from Earth to Mars: Consequences and opportunities*. Planetary y Space Science, 2013.
- [8] H. CURTIS. “Orbital mechanics for engineering students”. En: (2015).
- [9] J. MASON J. TRASK y L. COVERSTONE. “Optimal Interplanetary Trajectories Using Constant Radial Thrust and Gravitational Assists”. En: *Journal of Guidance, Control, and Dynamics* (2004), pág. 3.
- [10] J.ANDERSON. “Hohmann transfer orbit”. En: *Part of the Series Encyclopedia of Earth Science*. Springer (2015), págs. 309-310.
- [11] W. SCHULZ. “Optimal space maneuvers in three dimensions”. En: *Journal of the brazilian society of mechanical sciences and engineering* (2016), págs. 375-377.
- [12] D. VALLADO. *Fundamentals of Astrodynamics and Applications*. Springe, 2015, págs. 311-312.

- [13] XIN MA. *An overview of the autonomous navigation for a gravity-assist interplanetary spacecraft*. 2013.
- [14] A. RODRIGUEZ. *Diseño de Transferencia Óptimas con Cambio de Plano*. 2014.
- [15] R. SAMPIERI. *Metodología de la Investigación*. 2010.
- [16] J. CEGARRA. *Metodología de la investigación científica y tecnológica*. 2004.
- [17] MATHWORKS. *Tutoriales de MATLAB y Simulink*. Citado el 11 de enero de 2019. URL: <https://es.mathworks.com/support/learn-with-matlab-tutorials.html>.
- [18] R. MARTINEZ. *Programación en Matlab*. 2015.
- [19] NASA. *Mars Global Surveyor*. Citado el 17 de agosto de 2019. URL: <https://mars.nasa.gov/mars-exploration/missions/mars-global-surveyor/>.
- [20] XU. JUNYAN XIE. LI Z. YIQUN. *Hohmann Transfer via Constrained Optimization*. 2017.
- [21] M. GUILLON. *Temperature earth-sized planets transiting a nearby ultracool dwarf star*. 2016.
- [22] M. GUILLON. *Seven temperate terrestrial planets around the nearby ultracool dwarf star*. 2017.
- [23] Inc Wolfram Research. *Mathematica, Version 11.3*. Champaign, IL, 2019.

Apéndice A

Primer Apéndice

Figura A.1: Algoritmo para el cálculo de vectores de estado y elementos orbitales dada una fecha

The image shows a software interface with two main sections: 'PANEL DE ENTRADAS' and 'Elementos Orbitales'. The 'PANEL DE ENTRADAS' section includes a dropdown menu for 'Escoger Planeta' (set to 'Mercurio'), and input fields for 'Año' (1901-2099), 'Mes' (1-12), 'Día' (1-31), 'Hora' (0-23), 'Minuto' (0-60), and 'Segundo' (0-60). A 'CALCULAR' button is located below these fields. Below the input fields is a 'Día Juliano' input field. The 'Elementos Orbitales' section contains input fields for: 'Momentum angular [km²/s]', 'Excentricidad', 'RA del nodo ascendente [°]', 'Inclinación [°]', 'Argumento del perigeo [°]', 'Anomalía verdadera [°]', 'Semieje mayor [km]', 'Longitud del perihelio [°]', 'Longitud media [°]', 'Anomalía media [°]', and 'Anomalía excentrica [km]'. The 'Vector de Estado' section includes input fields for: 'Vector posición [km]', 'Magnitud posición', 'Vector velocidad [km/s]', and 'Magnitud velocidad'.

Fuente: [Autor]

Figura A.2: Algoritmo para el cálculo de vectores de estado y elementos orbitales dada una fecha (continuación 1)

```

68 function CALCULARButtonPushed(app, event)
69 - global mu
70 mu = 132712000000;
71 - global au
72 au = 149597871;
73 %deg = 180/pi;
74 deg = pi/180;
75
76 result=0;
77
78 %...Equation 5.48:
79 % j0 = J0(year, month, day);
80
81 %year1 = app.r1kmEditField.Value;
82 %l2 = app.r2kmEditField.Value;
83
84 %R1 = str2double(strsplit(l1, ','));
85 %R2 = str2double(strsplit(l2, ','));
86
87 - year = str2double(strsplit(app.AoEditField_2.Value));
88 month = str2double(strsplit(app.MesEditField_2.Value));
89 - day = str2double(strsplit(app.DaEditField_2.Value));
90 hour = str2double(strsplit(app.HoraEditField_2.Value));
91 - minute = str2double(strsplit(app.MinutoEditField_2.Value));
92 second = str2double(strsplit(app.SegundoEditField_2.Value));
93
94 %Día Juliano
95
96 j0 = 367*year - fix(7*(year + fix((month + 9)/12))/4)...
97 + fix(275*month/9) + day + 1721013.5;
98 ut = (hour + minute/60 + second/3600)/24;
99 %...Equation 5.47
100 jd = j0 + ut;
101 %Fin día Juliano

```

Fuente: [Autor]

Figura A.3: Algoritmo para el cálculo de vectores de estado y elementos orbitales dada una fecha (continuación 2)

```

102
103 %...Obtain the data for the selected planet from Table 8.1:
104
105 %J200 TABLES
106 J2000_elements = ...
107 [ 0.38709893 0.20563069 7.00487 48.33167 77.45645 252.25084
108 0.72333199 0.00677323 3.39471 76.68069 131.53298 181.97973
109 1.00000011 0.01671022 0.00005 -11.26064 102.94719 100.46435
110 1.52366231 0.09341233 1.85061 49.57854 336.04084 355.45332
111 5.20336301 0.04839266 1.30530 100.55615 14.75385 34.40438
112 9.53707032 0.05415060 2.48446 113.71504 92.43194 49.94432
113 19.19126393 0.04716771 0.76986 74.22988 170.96424 313.23218
114 30.06896348 0.00858587 1.76917 131.72169 44.97135 304.88003
115 39.48168677 0.24880766 17.14175 110.30347 224.06676 238.92881];
116
117 - cent_rates = ...
118 [ 0.00000066 0.00002527 -23.51 -446.30 573.57 538101628.29
119 0.00000092 -0.00004938 -2.86 -996.89 -108.80 210664136.06
120 -0.00000005 -0.00003804 -46.94 -18228.25 1198.28 129597740.63
121 -0.00007221 0.00011902 -25.47 -1020.19 1560.78 68905103.78
122 0.00060737 -0.00012880 -4.15 1217.17 839.93 10925078.35
123 -0.00301530 -0.00036762 6.11 -1591.05 -1948.89 4401052.95
124 0.00152025 -0.00019150 -2.09 -1681.4 1312.56 1542547.79
125 -0.00125196 0.00002514 -3.64 -151.25 -844.43 786449.21
126 -0.00076912 0.00006465 11.07 -37.33 -132.25 522747.90];
127 %END J200 TABLES
128
129 %Ecuacion 8.104a
130 t0 = (jd - 2451545)/36525;
131 %Fin 8.104a
132

```

Fuente: [Autor]

Figura A.4: Algoritmo para el cálculo de vectores de estado y elementos orbitales dada una fecha (continuación 3)

```

133     %...Convert from AU to km:
134
135 -   if(strcmp(app.EscogerPlanetaDropDown_2.Value, 'Mercurio'))
136       a = J2000_elements(1,1)* au+ cent_rates(1,1)*au*t0;
137 -   app.SemiejemayorkmTextArea.Value = num2str(a, '\n %g');
138
139 -   e = J2000_elements(1,2)+ cent_rates(1,2)*t0;
140   app.ExcentricidadTextArea.Value = num2str(e, '\n %g');
141
142   h = sqrt(mu*a*(1 - e^2));
143 -   app.Momentumangularkm2sTextArea.Value = num2str(h);
144
145 -   incl = J2000_elements(1,3)+ (cent_rates(1,3)/3600)*t0;
146   app.InclinacinTextArea.Value = num2str(incl, '\n %g');
147
148   RA = zero_to_360(J2000_elements(1,4)+ (cent_rates(1,4)/3600)*t0);
149 -   app.RAdeInodoascendenteTextArea.Value = num2str(RA, '\n %g');
150
151 -   w_hat = zero_to_360(J2000_elements(1,5)+ (cent_rates(1,5)/3600)*t0);
152   app.LongituddePerihelioTextArea.Value = num2str(w_hat, '\n %g');
153
154   L = zero_to_360(J2000_elements(1,6)+ (cent_rates(1,6)/3600)*t0);
155 -   app.LongitudmediaTextArea.Value = num2str(L, '\n %g');
156
157 -   w = zero_to_360(w_hat - RA);app.ArgumentodePerigeoTextArea.Value = num2str(w);
158   app.ArgumentodePerigeoTextArea.Value = num2str(w, '\n %g');
159
160   M = zero_to_360((L - w_hat));
161 -   app.AnomaliaMediaTextArea.Value = num2str(M, '\n %g');
162
163   %...Algorithm 3.1 (for which M must be in radians)
164   %E = kepler_E(e, M*deg);
165 -   error = 1e-8;
166   %...Select a starting value for E:
167   ... ..

```

Fuente: [Autor]

Figura A.5: Algoritmo para el cálculo de vectores de estado y elementos orbitales dada una fecha (continuación 4)

```

166     %...Select a starting value for E:
167 -   M1=M*deg;
168
169 -   if (M1)<pi
170     E=M1+ e/2;
171 -   else
172     E=M1- e/2;
173 -   end
174     %...Iterate on Equation 3.14 until E is determined to within
175     %...the error tolerance:
176     ratio = 1;
177 -   while abs(ratio) > error
178     ratio = (E - e*sin(E) - M1)/(1 - e*cos(E));
179 -   E=E- ratio;
180     end
181 -   app.AnomalaexcentricakmTextArea.Value = num2str(E/deg, '\n %g');
182
183     %...Equation 3.10 (converting the result to degrees):
184     TA = zero_to_360...
185     (2*atan(sqrt((1 + e)/(1 - e))*tan(E/2))/deg);
186     app.AnomalaverdaderaTextArea.Value = num2str(TA, '\n %g');
187
188     %...Algorithm 4.2 (for which all angles must be in radians):
189
190     RA1 = RA*deg;
191 -   incl1 = incl*deg;
192     w1 = w*deg;
193 -   TA1 = TA*deg;
194
195     %...Equations 4.37 and 4.38 (rp and vp are column vectors):
196     rp = (h^2/mu) * (1/(1 + e*cos(TA1))) * (cos(TA1)*[1;0;0] + sin(TA1)*[0;1;0]);
197 -   vp = (mu/h) * (-sin(TA1)*[1;0;0] + (e + cos(TA1))*[0;1;0]);
198     %...Equation 4.39:
199 -   R3_W = [ cos(RA1) sin(RA1) 0

```

Fuente: [Autor]

Figura A.6: Algoritmo para el cálculo de vectores de estado y elementos orbitales dada una fecha (continuación 5)

```

195     %...Equations 4.37 and 4.38 (rp and vp are column vectors):
196     rp = (h^2/mu) * (1/(1 + e*cos(TA1))) * (cos(TA1)*[1;0;0] + sin(TA1)*[0;1;0]);
197 -   vp = (mu/h) * (-sin(TA1)*[1;0;0] + (e + cos(TA1))*[0;1;0]);
198     %...Equation 4.39:
199 -   R3_W = [ cos(RA1)  sin(RA1)  0
200            -sin(RA1)  cos(RA1)  0
201            0          0        1];
202     %...Equation 4.40:
203 -   R1_i = [1      0      0
204            0  cos(incl1) sin(incl1)
205            0 -sin(incl1) cos(incl1)];
206     %...Equation 4.41:
207 -   R3_w = [ cos(w1)  sin(w1)  0
208            -sin(w1)  cos(w1)  0
209            0          0        1];
210     %...Equation 4.44:
211 -   Q_pX = R3_W'*R1_i'*R3_w';
212     %...Equations 4.46 (r and v are column vectors):
213 -   r = Q_pX*rp;
214     v = Q_pX*vp;
215     %...Convert r and v into row vectors:
216     r = r';
217 -   v = v';
218
219 -   app.VectorposicinkmTextArea.Value = num2str(real(r), '\n [%g %g %g]');
220
221 -   app.MagnitudposicinTextArea.Value = num2str(norm(real(r)), '\n %g\n');
222
223 -   app.VectorvelocidadkmsTextArea.Value = num2str(real(v), '\n [%g %g %g]');
224
225 -   app.MagnitudvelocidadTextArea.Value = num2str(norm(real(v)), '\n %g\n');
226
227
228     elseif(strcmp(app.EscogerPlanetaDropDown_2.Value, 'Venus'))

```

Fuente: [Autor]

Figura A.7: Algoritmo para el cálculo de vectores de estado y elementos orbitales dada una fecha (continuación 6)

```

320 elseif(strcmp(app.EscogerPlanetaDropDown_2.Value, 'Tierra'))
321
322     a = J2000_elements(3,1)* au+ cent_rates(3,1)*au*t0;
323 -   app.SemiejemayorkmTextArea.Value = num2str(a,'\n %g');
324
325 -   e = J2000_elements(3,2)+ cent_rates(3,2)*t0;
326     app.ExcentricidadTextArea.Value = num2str(e,'\n %g');
327
328     h = sqrt(mu*a*(1 - e^2));
329 -   app.Momentumangularkm2sTextArea.Value = num2str(h,'\n %g');
330
331 -   incl = J2000_elements(3,3)+(cent_rates(3,3)/3600)*t0;
332     app.InclinaciónTextArea.Value = num2str(incl,'\n %g');
333
334     RA = zero_to_360(J2000_elements(3,4)+ (cent_rates(3,4)/3600)*t0);
335 -   app.RAdeInodoascendenteTextArea.Value = num2str(RA,'\n %g');
336
337 -   w_hat = zero_to_360(J2000_elements(3,5)+ (cent_rates(3,5)/3600)*t0);
338     app.LongituddeperihelioTextArea.Value = num2str(w_hat,'\n %g');
339
340     L = zero_to_360(J2000_elements(3,6)+ (cent_rates(3,6)/3600)*t0);
341 -   app.LongitudmediaTextArea.Value = num2str(L,'\n %g');
342
343 -   w = zero_to_360(w_hat - RA);
344     app.ArgumentodeperigeoTextArea.Value = num2str(w,'\n %g');
345
346     M = zero_to_360((L - w_hat));
347 -   app.AnomalamediaTextArea.Value = num2str(M,'\n %g');
348
349     %...Algorithm 3.1 (for which M must be in radians)
350     %E = kepler_E(e, M*deg);
351 -   error = 1e-8;
352     %...Select a starting value for E:
353 -   M1=M*deg;

```

Fuente: [Autor]

Figura A.8: Algoritmo para el cálculo de vectores de estado y elementos orbitales dada una fecha (continuación 7)

```

352     %...Select a starting value for E:
353 -     M1=M*deg;
354
355 -     if (M1)<pi
356 -     E=M1+ e/2;
357 -     else
358 -     E=M1- e/2;
359 -     end
360     %...Iterate on Equation 3.14 until E is determined to within
361     %...the error tolerance:
362     ratio = 1;
363 -     while abs(ratio) > error
364 -     ratio = (E - e*sin(E) - M1)/(1 - e*cos(E));
365 -     E=E- ratio;
366 -     end
367 -     app.AnomalaexcentricakmTextArea.Value = num2str(E/deg, '\n %g');
368
369     %...Equation 3.10 (converting the result to degrees):
370     TA = zero_to_360...
371     (2*atan(sqrt((1 + e)/(1 - e))*tan(E/2))/deg);
372     app.AnomalaverdaderaTextArea.Value = num2str(TA, '\n %g');
373
374     %...Algorithm 4.2 (for which all angles must be in radians):
375
376     RA1 = RA*deg;
377 -     incl1 = incl*deg;
378     w1 = w*deg;
379 -     TA1 = TA*deg;
380
381     %...Equations 4.37 and 4.38 (rp and vp are column vectors):
382     rp = (h^2/mu) * (1/(1 + e*cos(TA1))) * (cos(TA1)*[1;0;0] + sin(TA1)*[0;1;0]);
383 -     vp = (mu/h) * (-sin(TA1)*[1;0;0] + (e + cos(TA1))*[0;1;0]);

```

Fuente: [Autor]

Figura A.9: Algoritmo para el cálculo de vectores de estado y elementos orbitales dada una fecha (continuación 8)

```

384         %...Equation 4.39:
385 -       R3_W = [ cos(RA1) sin(RA1) 0
386               -sin(RA1) cos(RA1) 0
387               0 0 1];
388         %...Equation 4.40:
389 -       R1_i = [1 0 0
390               0 cos(incl1) sin(incl1)
391               0 -sin(incl1) cos(incl1)];
392         %...Equation 4.41:
393 -       R3_w = [ cos(w1) sin(w1) 0
394               -sin(w1) cos(w1) 0
395               0 0 1];
396         %...Equation 4.44:
397 -       Q_pX = R3_W'*R1_i'*R3_W';
398         %...Equations 4.46 (r and v are column vectors):
399 -       r = Q_pX*rp;
400         v = Q_pX*vp;
401         %...Convert r and v into row vectors:
402         r = r';
403         v = v';
404
405 -       app.VectorposicinkmTextArea.Value = num2str(real(r), '\n [%g %g %g]');
406
407 -       app.MagnitudposicinTextArea.Value = num2str(norm(real(r)), '\n %g\n');
408
409 -       app.VectorvelocidadkmsTextArea.Value = num2str(real(v), '\n [%g %g %g]');
410
411 -       app.MagnitudvelocidadTextArea.Value = num2str(norm(real(v)), '\n %g\n');
412
413
414
415 -     elseif(strcmp(app.EscogerPlanetaDropDown_2.Value, 'Marte'))

```

Fuente: [Autor]

Figura A.10: Algoritmo para el cálculo de vectores de estado y elementos orbitales dada una fecha (continuación 9)

```

878     else
879 -         a = J2000_elements(9,1)* au+ cent_rates(9,1)*au*t0;
880             app.SemiejemayorkmTextArea.Value = num2str(a,'\n %g');
881
882             e = J2000_elements(9,2)+ cent_rates(9,2)*t0;
883 -         app.ExcentricidadTextArea.Value = num2str(e,'\n %g');
884
885 -         h = sqrt(mu*a*(1 - e^2));
886             app.Momentumangularkm2sTextArea.Value = num2str(h,'\n %g');
887
888             incl = J2000_elements(9,3)+ (cent_rates(9,3)/3600)*t0;
889 -         app.InclinacinTextArea.Value = num2str(incl,'\n %g');
890
891 -         RA = zero_to_360(J2000_elements(9,4)+ (cent_rates(9,4)/3600)*t0);
892             app.RAdeInodoascendenteTextArea.Value = num2str(RA,'\n %g');
893
894 -         w_hat = zero_to_360(J2000_elements(9,5)+ (cent_rates(9,5)/3600)*t0);
895             app.LongituddelperihelioTextArea.Value = num2str(w_hat,'\n %g');
896
897 -         L = zero_to_360(J2000_elements(9,6)+ (cent_rates(9,6)/3600)*t0);
898             app.LongitudmediaTextArea.Value = num2str(L,'\n %g');
899
900 -         w = zero_to_360(w_hat - RA);
901             app.ArgumentodelperigeoTextArea.Value = num2str(w,'\n %g');
902
903 -         M = zero_to_360((L - w_hat));
904             app.AnomalamediaTextArea.Value = num2str(M,'\n %g');
905
906             %..Algorithm 3.1 (for which M must be in radians)
907             %E = kepler_E(e, M*deg);
908             error = 1e-8;
909             %..Select a starting value for E:
910             M1=M*deg;

```

Fuente: [Autor]

Figura A.11: Algoritmo para el cálculo de vectores de estado y elementos orbitales dada una fecha (continuación 10)

```

912     if (M1)<pi
913 -     E=M1+ e/2;
914     else
915 -     E=M1- e/2;
916     end
917     %...Iterate on Equation 3.14 until E is determined to within
918     %...the error tolerance:
919 -     ratio = 1;
920     while abs(ratio) > error
921 -     ratio = (E - e*sin(E) - M1)/(1 - e*cos(E));
922     E=E- ratio;
923 -     end
924     app.AnomalaexcentricakmTextArea.Value = num2str(E/deg, '\n %g');
925
926     %...Equation 3.10 (converting the result to degrees):
927 -     TA = zero_to_360...
928     (2*atan(sqrt((1 + e)/(1 - e))*tan(E/2))/deg);
929 -     app.AnomalaverdaderaTextArea.Value = num2str(TA, '\n %g');
930
931     %...Algorithm 4.2 (for which all angles must be in radians):
932
933 -     RA1 = RA*deg;
934     incl1 = incl*deg;
935 -     w1 = w*deg;
936     TA1 = TA*deg;
937
938     %...Equations 4.37 and 4.38 (rp and vp are column vectors):
939 -     rp = (h^2/mu) * (1/(1 + e*cos(TA1))) * (cos(TA1)*[1;0;0] + sin(TA1)*[0;1;0]);
940     vp = (mu/h) * (-sin(TA1)*[1;0;0] + (e + cos(TA1))*[0;1;0]);
941     %...Equation 4.39:
942     R3_W = [ cos(RA1) sin(RA1) 0
943             -sin(RA1) cos(RA1) 0
944             0 0 1];
945     % Equation 4.40:

```

Fuente: [Autor]

Figura A.12: Algoritmo para el cálculo de vectores de estado y elementos orbitales dada una fecha (continuación 11)

```

944         0     0     1];
945     %...Equation 4.40:
946     R1_i = [1     0     0
947             0   cos(incl1) sin(incl1)
948             0  -sin(incl1) cos(incl1)];
949     %...Equation 4.41:
950     R3_w = [ cos(w1) sin(w1) 0
951             -sin(w1) cos(w1) 0
952             0     0     1];
953     %...Equation 4.44:
954     Q_pX = R3_w'*R1_i'*R3_w';
955     %...Equations 4.46 (r and v are column vectors):
956     r = Q_pX*rp;
957 -     v = Q_pX*vp;
958     %...Convert r and v into row vectors:
959 -     r = r';
960     v = v';
961
962     app.VectorposicinkmTextArea.Value = num2str(real(r), '\n [%g %g %g]');
963
964     app.MagnitudposicinTextArea.Value = num2str(norm(real(r)), '\n %g\n');
965
966     app.VectorvelocidadkmsTextArea.Value = num2str(real(v), '\n [%g %g %g]');
967
968     app.MagnitudvelocidadTextArea.Value = num2str(norm(real(v)), '\n %g\n');
969
970 end
971
972 app.DaJulianoTextArea.Value = num2str(jd, '\n\n %11.3f');
973
974
975 -     end
976     end
977

```

Fuente: [Autor]

Apéndice B

Segundo Apéndice

Figura B.1: Algoritmo para el cálculo de la trayectoria desde un planeta 1 hasta a un planeta 2

SALIDA/DEPARTURE		LLEGADA/ARRIVAL			
Panel de entrada Planeta de salida		Panel de entrada Planeta de llegada			
Escoger Planeta de salida: Mercurio		Escoger Planeta de llegada: Mercurio			
Año	(1901 - 2099)	Año	(1901 - 2099)		
Mes	(1 - 12)	Mes	(1 - 12)		
Día	(1 - 31)	Día	(1 - 31)		
Hora	(0 - 23)	Hora	(0 - 23)		
Minuto	(0 - 60)	Minuto	(0 - 60)		
Segundo	(0 - 60)	Segundo	(0 - 60)		
<input type="button" value="Calcular"/>					
Resultado Planeta de Salida		Resultado Planeta de Llegada		Elementos orbitales de la trayectoria de vuelo	
Día Juliano		Día Juliano		Momentum angular [km ² /s]	
Vector posición del planeta [km]		Vector posición del planeta [km]		Excentricidad	
Magnitud posición		Magnitud posición		RA del nodo ascendente [°]	
Vector velocidad del planeta [km/s]		Vector velocidad del planeta [km/s]		Inclinación de la eclíptica [°]	
Magnitud velocidad		Magnitud velocidad		Argumento del perihelio [°]	
Vector velocidad de la nave espacial a la salida [km/s]		Vector velocidad de la nave espacial a la llegada [km/s]		Anomalía verdadera salida [°]	
Magnitud velocidad nave espacial v-infinita a la salida [km/s]		Magnitud velocidad nave espacial v-infinita a la llegada [km/s]		Anomalía verdadera llegada [°]	
Magnitud velocidad infinita salida		Magnitud velocidad infinita llegada		Semieje mayor [km]	
Tiempo de vuelo				Periodo [días]	

Fuente: [Autor]

Figura B.2: Algoritmo para el cálculo de la trayectoria desde un planeta 1 hasta a un planeta 2 (continuación 1)

```

117 - global mu r_1 r_2 tof_1 A_1
118     mu = 132712000000;
119 - global au
120     au = 149597871;
121     %deg = 180/pi;
122     deg = pi/180;
123
124
125     %...Obtain the data for the selected planet from Table 8.1:
126
127     %J200 TABLAS
128     J2000_elements = ...
129     [ 0.38709893 0.20563069 7.00487 48.33167 77.45645 252.25084
130       0.72333199 0.00677323 3.39471 76.68069 131.53298 181.97973
131       1.00000011 0.01671022 0.00005 -11.26064 102.94719 100.46435
132       1.52366231 0.09341233 1.85061 49.57854 336.04084 355.45332
133       5.20336301 0.04839266 1.30530 100.55615 14.75385 34.40438
134       9.53707032 0.05415060 2.48446 113.71504 92.43194 49.94432
135       19.19126393 0.04716771 0.76986 74.22988 170.96424 313.23218
136       30.06896348 0.00858587 1.76917 131.72169 44.97135 304.88003
137       39.48168677 0.24880766 17.14175 110.30347 224.06676 238.92881];
138
139 - cent_rates = ...
140     [ 0.00000066 0.00002527 -23.51 -446.30 573.57 538101628.29
141       0.00000092 -0.00004938 -2.86 -996.89 -108.80 210664136.06
142       -0.00000005 -0.00003804 -46.94 -18228.25 1198.28 129597740.63
143       -0.00007221 0.00011902 -25.47 -1020.19 1560.78 68905103.78
144       0.00060737 -0.00012880 -4.15 1217.17 839.93 10925078.35
145       -0.00301530 -0.00036762 6.11 -1591.05 -1948.89 4401052.95
146       0.00152025 -0.00019150 -2.09 -1681.4 1312.56 1542547.79
147       -0.00125196 0.00002514 -3.64 -151.25 -844.43 786449.21
148       -0.00076912 0.00006465 11.07 -37.33 -132.25 522747.90];
149     %FIN J200 TABLAS

```

Fuente: [Autor]

Figura B.3: Algoritmo para el cálculo de la trayectoria desde un planeta 1 hasta a un planeta 2 (continuación 2)

```

151
152 %%%%          DEPARTURE
153
154 %...Equation 5.48:
155 % j0 = J0(year, month, day);
156
157 - year1 = str2double(strsplit(app.AoEditField_2.Value));
158 - month1 = str2double(strsplit(app.MesEditField_2.Value));
159 - day1 = str2double(strsplit(app.DaEditField_2.Value));
160 - hour1 = str2double(strsplit(app.HoraEditField_2.Value));
161 - minute1 = str2double(strsplit(app.MinutoEditField_2.Value));
162 - second1 = str2double(strsplit(app.SegundoEditField_2.Value));
163
164 %Día Juliano Salida
165 - j01 = 367*year1 - fix(7*(year1 + fix((month1 + 9)/12))/4)...
166 -       + fix(275*month1/9) + day1 + 1721013.5;
167 - ut1 = (hour1 + minute1/60 + second1/3600)/24;
168 %...Equation 5.47
169 - jd1 = j01 + ut1;
170
171
172 app.DaJulianoTextArea.Value = num2str(jd1, '\n\n %11.3f');
173
174 %Fin día Juliano
175
176
177 %Ecuacion 8.104a
178 t01 = (jd1 - 2451545)/36525;
179 %Fin 8.104a
180

```

Fuente: [Autor]

Figura B.4: Algoritmo para el cálculo de la trayectoria desde un planeta 1 hasta a un planeta 2 (continuación 3)

```

181 %...Convert from AU to km:
182
183 - if(strcmp(app.EscogerPlanetadesalidaDropDown.Value, 'Mercurio'))
184     a1 = J2000_elements(1,1)* au+ cent_rates(1,1)*au*t01;
185     %app.SemiejemayorkmTextArea.Value = num2str(a,'\n %g');
186
187 -     e1 = J2000_elements(1,2)+ cent_rates(1,2)*t01;
188     %app.ExcentricidadTextArea.Value = num2str(e,'\n %g');
189
190     h1 = sqrt(mu*a1*(1 - e1^2));
191     %app.Momentumangularkm2sTextArea.Value = num2str(h);
192
193 -     incl1 = J2000_elements(1,3)+ (cent_rates(1,3)/3600)*t01;
194     %app.InclinacinTextArea.Value = num2str(incl,'\n %g');
195
196     RA1 = zero_to_360(J2000_elements(1,4)+ (cent_rates(1,4)/3600)*t01);
197     %app.RAdeInodoascendenteTextArea.Value = num2str(RA,'\n %g');
198
199 -     w_hat1 = zero_to_360(J2000_elements(1,5)+ (cent_rates(1,5)/3600)*t01);
200     %app.LongituddeperihelioTextArea.Value = num2str(w_hat,'\n %g');
201
202     L1 = zero_to_360(J2000_elements(1,6)+ (cent_rates(1,6)/3600)*t01);
203     %app.LongitudmediaTextArea.Value = num2str(L,'\n %g');
204
205 -     w1 = zero_to_360(w_hat1 - RA1);
206     %app.ArgumentodeperigeoTextArea.Value = num2str(w,'\n %g');
207
208     M1 = zero_to_360((L1 - w_hat1));
209     %app.AnomalamediaTextArea.Value = num2str(M,'\n %g');
210
211 %...Algorithm 3.1 (for which M must be in radians)
212 %E = kepler_E(e, M*deg);
213 - error = 1e-8;

```

Fuente: [Autor]

Figura B.5: Algoritmo para el cálculo de la trayectoria desde un planeta 1 hasta a un planeta 2 (continuación 4)

```

214     %...Select a starting value for E:
215     M11=M1*deg;
216
217     if (M11)<pi
218     E1=M11+ e1/2;
219     else
220     E1=M11- e1/2;
221     end
222     %...Iterate on Equation 3.14 until E is determined to within
223     %...the error tolerance:
224     ratio = 1;
225     while abs(ratio) > error
226     ratio = (E1 - e1*sin(E1) - M11)/(1 - e1*cos(E1));
227     E1=E1- ratio;
228     end
229
230     %app.AnomalaexcentricakmTextArea.Value = num2str(E1/deg,'\n %g');
231
232     %...Equation 3.10 (converting the result to degrees):
233     TA1 = zero_to_360...
234     (2*atan(sqrt((1 + e1)/(1 - e1))*tan(E1/2))/deg);
235
236     %app.AnomalaverdaderaTextArea.Value = num2str(TA,'\n %g');
237
238     %...Algorithm 4.2 (for which all angles must be in radians):
239
240     RA11 = RA1*deg;
241     incl11 = incl1*deg;
242     w11 = w1*deg;
243     TA11 = TA1*deg;
244
245     %...Equations 4.37 and 4.38 (rp and vp are column vectors):
246     rp1 = (h1^2/mu) * (1/(1 + e1*cos(TA11))) * (cos(TA11)*[1;0;0] + sin(TA11)*[0;1;0]);

```

Fuente: [Autor]

Figura B.6: Algoritmo para el cálculo de la trayectoria desde un planeta 1 hasta a un planeta 2 (continuación 5)

```

245     %...Equations 4.37 and 4.38 (rp and vp are column vectors):
246     rp1 = (h1^2/mu) * (1/(1 + e1*cos(TA11))) * (cos(TA11)*[1;0;0] + sin(TA11)*[0;1;0]);
247 -   vp1 = (mu/h1) * (-sin(TA11)*[1;0;0] + (e1 + cos(TA11))*[0;1;0]);
248     %...Equation 4.39:
249 -   R3_w1 = [ cos(RA11)  sin(RA11)  0
250             -sin(RA11)  cos(RA11)  0
251             0           0         1];
252     %...Equation 4.40:
253 -   R1_i1 = [1           0
254             0   cos(incl11)  sin(incl11)
255             0  -sin(incl11)  cos(incl11)];
256     %...Equation 4.41:
257 -   R3_w1 = [ cos(w11)  sin(w11)  0
258             -sin(w11)  cos(w11)  0
259             0           0         1];
260     %...Equation 4.44:
261 -   Q_pX1 = R3_w1'*R1_i1'*R3_w1';
262     %...Equations 4.46 (r and v are column vectors):
263 -   r1 = Q_pX1*rp1;
264     v1 = Q_pX1*vp1;
265     %...Convert r and v into row vectors:
266     Rp1 = r1';
267 -   Vp1 = v1';
268
269 -   app.VectorposicindelplanetakmTextArea.Value = num2str(real(Rp1), '\n [%g   %g   %g]');
270
271 -   app.MagnitudposicinTextArea.Value = num2str(norm(real(Rp1)), '\n %g\n');
272
273 -   app.VectorvelocidaddelplanetakmsTextArea.Value = num2str(real(Vp1), '\n [%g   %g   %g]');
274
275 -   app.MagnitudvelocidadTextArea.Value = num2str(norm(real(Vp1)), '\n %g\n');
276
277 -   elseif(strcmp(app.EscogerPlanetadesalidaDropDown.Value, 'Venus'))
278     a1 = J2000_elements(2,1)* au+ cent_rates(2,1)*au*t01;

```

Fuente: [Autor]

Apéndice C

Tercer Apéndice

Figura C.1: Algoritmo para el calculo de la trayectoria desde un planeta 1 hasta a un planeta 2 realizado por el Autor

The screenshot shows a software interface titled "UI Figure" with the following sections:

- SALIDA/DEPARTURE**: Panel de entrada Planeta de salida. Escoger Planeta de salida: Tierra. Año: 1996 (1901 - 2099). Mes: 11 (1 - 12). Día: 7 (1 - 31). Hora: 0 (0 - 23). Minuto: 0 (0 - 60). Segundo: 0 (0 - 60).
- LLEGADA/ARRIVAL**: Panel de entrada Planeta de llegada. Escoger Planeta de llegada: Marte. Año: 1997 (1901 - 2099). Mes: 9 (1 - 12). Día: 12 (1 - 31). Hora: 0 (0 - 23). Minuto: 0 (0 - 60). Segundo: 0 (0 - 60).
- Resultado Planeta de Salida**: Día Juliano: 2450394.500. Vector posición del planeta [km]: [1.04994e+08 1.04655e+08 988.331]. Magnitud posición: 1.48244e+08. Vector velocidad del planeta [km/s]: [-21.5149 20.9865 0.000132283]. Magnitud velocidad: 30.0554. Vector velocidad de la nave espacial a la salida [km/s]: [-24.4282 21.7819 0.948047]. Magnitud velocidad nave espacial: 32.7427. v-infinita a la salida [km/s]: [-2.91322 0.795397 0.947915]. Magnitud velocidad infinita salida: 3.16513. Tiempo de vuelo: 309 días.
- Resultado Planeta de Llegada**: Día Juliano: 2450703.500. Vector posición del planeta [km]: [-2.08329e+07 -2.18404e+08 -4.06287e+06]. Magnitud posición: 2.19433e+08. Vector velocidad del planeta [km/s]: [25.0385 -0.220288 -0.620622]. Magnitud velocidad: 25.0472. Vector velocidad de la nave espacial a la llegada [km/s]: [22.158 -0.196685 -0.457847]. Magnitud velocidad nave espacial: 22.1636. v-infinita a la llegada [km/s]: [-2.88049 0.0236029 0.162775]. Magnitud velocidad infinita llegada: 2.88518.
- Elementos orbitales de la trayectoria de vuelo**: Momentum angular [km²/s]: 4.84553e+09. Excentricidad: 0.205785. RA del nodo ascendente [°]: 44.8942. Inclínación de la eclíptica [°]: 1.6621. Argumento del perihelio [°]: 19.9741. Anomalía verdadera salida [°]: 340.039. Anomalía verdadera llegada [°]: 199.695. Semieje mayor [km]: 1.84742e+08. Periodo [días]: 501.254.

Fuente: [Autor]

Figura C.2: Algoritmo para el calculo de la trayectoria desde un planeta 1 hasta a un planeta 2 estipulado en [8] (a)

```
Example Mars Global Surveyor
-----
Departure:

Planet: Earth
Year : 1996
Month : November
Day : 7
Hour : 0
Minute: 0
Second: 0

Julian day: 2450394.500

Planet position vector (km) = [1.04994e+08 1.04655e+08 988.331]
Magnitude                   = 1.48244e+08

Planet velocity (km/s)     = [-21.515 20.9865 0.000132284]
Magnitude                   = 30.0554

Spacecraft velocity (km/s) = [-24.4282 21.7819 0.948049]
Magnitude                   = 32.7427

v-infinity at departure (km/s) = [-2.91321 0.79542 0.947917]
Magnitude                   = 3.16513

Time of flight              = 309 days

Arrival:

Planet: Mars
Year : 1997
Month : September
Day : 12
Hour : 0
Minute: 0
Second: 0

Julian day: 2450703.500
```

Fuente: [8]

Figura C.3: Algoritmo para el calculo de la trayectoria desde un planeta 1 hasta a un planeta 2 estipulado en [8] (b)

```

Planet position vector (km)      = [-2.08329e+07 -2.18404e+08 -4.06287e+06]
Magnitude                        = 2.19433+08

Planet velocity (km/s)          = [25.0386 -0.220288 -0.620623]
Magnitude                        = 25.0472

Spacecraft Velocity (km/s)      = [22.1581 -0.19666 -0.457847]
Magnitude                        = 22.1637

v-infinity at arrival (km/s)    = [-2.88049 0.023628 0.162776]
Magnitude                        = 2.88518

Orbital elements of flight trajectory:

Angular momentum (km^2/s)       = 4.84554e+09
Eccentricity                     = 0.205785
Right ascension of the ascending node (deg) = 44.8942
Inclination to the ecliptic (deg) = 1.6621
Argument of perihelion (deg)    = 19.9738
True anomaly at departure (deg) = 340.039
True anomaly at arrival (deg)   = 199.695

Semimajor axis (km)             = 1.84742e+08
Period (days)                  = 501.254
-----

```

Fuente: [8]