Interplanetary Trajectory Analysis for 2020-2040 Mars Missions including Venus Flyby Opportunities

by

Takuto Ishimatsu

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
[June 2008]
May 2008

Author .....
Department of Aeronautics and Astronautics
May 23, 2008

Certified by .....
Jeffrey A. Hoffman
Professor of the Practice
Thesis Supervisor

Accepted by .....
Prof. David L. Darmofal
Associate Department Head
Chair, Committee on Graduate Students

Interplanetary Trajectory Analysis for 2020-2040 Mars Missions including Venus Flyby Opportunities

by

Takuto Ishimatsu

Submitted to the Department of Aeronautics and Astronautics
on May 23, 2008, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

This thesis develops a tool which is capable of calculating ballistic interplanetary trajectories with planetary flyby options based on the knowledge of astrodynamics and analyzes Mars trajectories in the time frame 2020 to 2040, including transfer trajectories with Venus flybys. Using the trajectory programs developed in this work, we investigate the relation between departure and arrival dates and energy required for the transfer trajectories. The contours of C3 or ΔV_{tot} for a range of departure dates and times of flight would be useful for the creation of a long-term Earth-Mars and Mars-Earth transportation schedule for mission planning purposes. For planetary flybys, we allow simple powered flybys with the velocity impulse at periapsis to expand the flyby mission windows. Having obtained the results for Earth-Mars and Mars-Earth trajectories by a full-factorial computation, we discuss the nature of the trajectories and the competitiveness of Earth-Venus-Mars flyby trajectory windows with Earth-Mars direct trajectory windows.

Thesis Supervisor: Jeffrey A. Hoffman
Title: Professor of the Practice

Acknowledgments

In the course of this thesis work, I was lucky to gain a large amount of energy and adjust my trajectory by making "multiple flybys" of many people around me.

First of all, I would like to thank my advisor Professor Jeffrey Hoffman. He guided me with his appropriate advice and generous heart even though I was late to start this thesis work. I could not have completed this thesis and overcome other challenging difficulties without his continuous support.

I would like to express my gratitude to Professor Richard Battin for his inspiring and time-honored class at MIT (Astrodynamics), which became a starting point of my research. He kindly helped me with my poor English when I just came to MIT. I am also appreciative of Professor Olivier de Weck for his interesting suggestion through his class at MIT (Multidisciplinary System Design Optimization), which shed light on another aspect and helped push forward this research. I would also like to thank Wilfried Hofstetter for his mentorship through this work. It was he who proposed this research topic. I truly enjoyed the opportunity to work on this topic and share thoughts with him.

On the more personal side, I could not have survived the challenging days without Phillip Cunio, who has constantly supported and encouraged me since I came to MIT, Shinya Umeno, who has had both technical and non-technical discussions with me. Wasabis and Japanese Association of MIT (JAM) are an irreplaceable part of my life at MIT. I thank all the members for sharing the best time with me.

And last but not least, I sincerely thank my family, Father, Mother, Yuto, and Yui for understanding my dream, encouraging me to study abroad, and loving me. My special thanks also go to my one and only, Yoshiko Aizono. I owe what I am today to her.

This thesis is dedicated to my grandfather, Fumio Ishimatsu, who was a railroad engineer and passed away last March, and my grandmother, Tomiko Fukuhara, who passed away this May.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The Vision for Space Exploration, which was announced at NASA Headquarters on January 14, 2004, calls for the sustained and affordable robotic and human program to explore the Moon, Mars, and beyond [1]. The major element to be developed as part of the Vision is the Crew Exploration Vehicle, or CEV, which later became known as the Orion spacecraft (Figure 1-1). The Orion's development is taking place in parallel with missions to complete the International Space Station (ISS) using the Space Shuttle. After the Space Shuttle will be retired along with the completion of ISS around the year 2010, the Orion program will provide a state-of-the-art human space flight system capable of safely transferring astronauts to and from ISS. According to the Vision, America will also send human explorers back to the Moon. Making its first flights early in the next decade, Orion will be playing an important role in Project Constellation to send a new generation of explorers to the Moon, and then onward to Mars and other destinations in the solar system.

1.1.1 Mars Exploration Studies

Thus the next several decades will see an increasing number of both unmanned and manned missions to Mars. A robotic exploration of Mars is for the purpose of prepar-

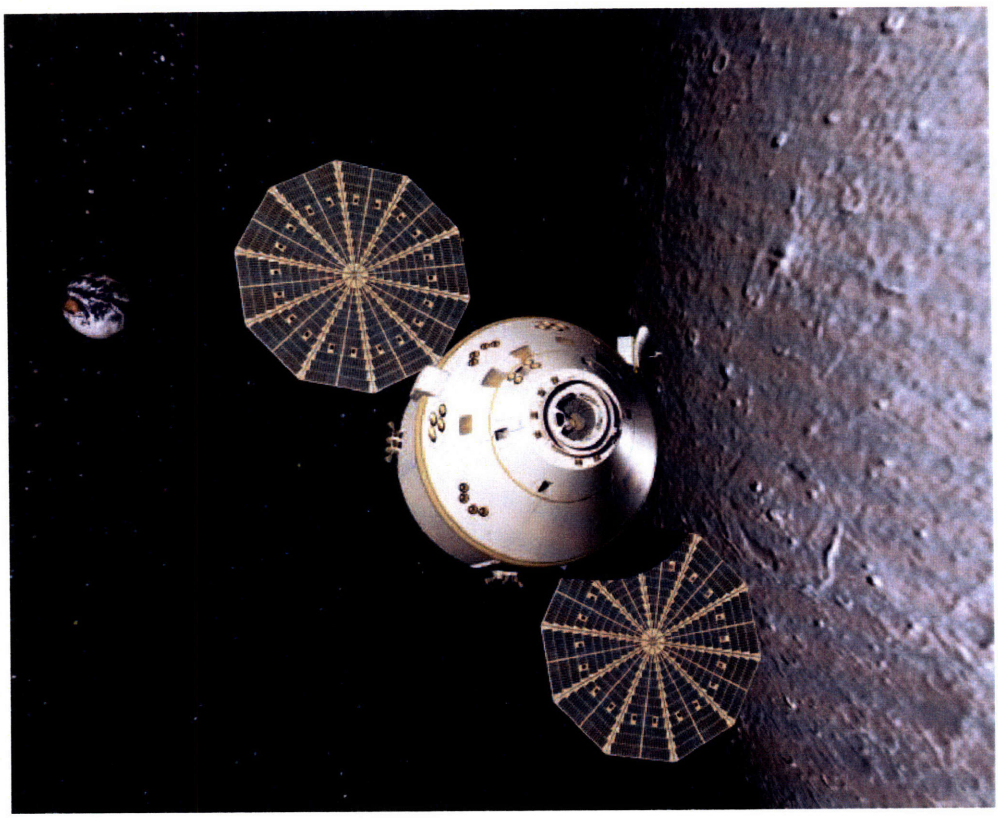


Figure 1-1: Orion crew exploration vehicle (CEV)

ing for future human exploration as well as for the purpose of searching for evidence of life and understanding the history of the solar system. Then we will embark on human expeditions to Mars after acquiring adequate knowledge about the planet using robotic missions and after successfully demonstrating sustained human exploration missions to the Moon.

A wide variety of Mars exploration studies have been conducted in the past. Many of them are related to the system architecture [2–9]. These studies focus on the Moon-Mars commonality, sustainability, and affordability, which are important in light of NASA's budget constraints. The reference mission architecture is presented by the NASA Mars Exploration Study Team [10, 11].

In addition to the system architecture, at the same time, we should also focus on the mission scenarios, especially the transfer trajectories. Several studies on Mars trajectories have been carried out [12–17]. Planning future missions requires trajec-

tory data years in advance. As described in more details later in Chapter 2, a transfer trajectory governed solely by the solar gravity can be uniquely determined by the positions of departure and arrival and the time between them. Since from astronomical data we know the heliocentric positions of the planets at a specific time in the future, we can calculate transfer trajectories in advance, which would be useful for future mission planning. The next section describes the type of Mars mission opportunities and trajectories.

1.1.2 Mars Mission Opportunities

As shown in Figure 1-2, there are two main types of interplanetary trajectories from Earth to Mars as a practical way: Earth-Mars direct trajectory and Earth-Venus-Mars flyby trajectory. These are ballistic trajectories in the Sun's gravitational field.

In general, Earth-Mars direct trajectory is practically achievable when Earth at departure and Mars at arrival are on the opposite side of the Sun, which means that the trajectory goes halfway around the Sun as in Figure 1-2. Such a flight opportunity recurs every 780 days since the same relative angular positions of Earth and Mars repeat after a synodic period of approximately 2.136 years (780 days).

On the other hand, an alternate trajectory is Earth-Venus-Mars flyby trajectory which goes first inward toward Venus and utilizes the gravitational field of Venus to accelerate the spacecraft outward toward Mars. The gravitational field of Venus is used to reduce the propulsion requirement. This trajectory is practically achievable when Earth at departure and Venus at encounter are on the opposite side of the Sun and at the same time Earth at departure and Mars at arrival are on the same side of the Sun. This type results in a relatively long trajectory which goes about 360° around the Sun. The relative configurations among Earth, Mars, and Venus repeat after a syzygistic period of approximately 6.4 years (2338 days), which contains almost exactly three Earth-Mars and four Earth-Venus synodic periods [18]. Therefore, we would expect that the periodicity of the Venus flyby missions is also governed by this 6.4-year syzygistic cycle, at least in a qualitative sense since the planets move on elliptic inclined orbits.

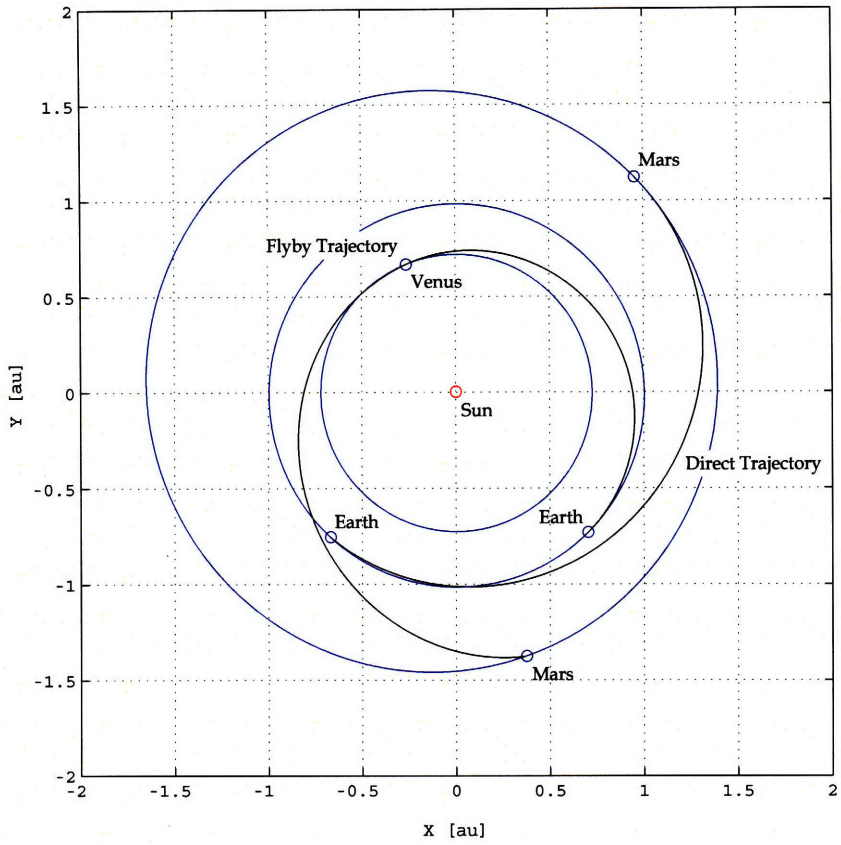


Figure 1-2: Direct and flyby trajectories to Mars

For the reasons described above, a mission that uses a direct trajectory for both legs of the trip is referred to as a "conjunction class" mission, and a mission that uses a flyby trajectory is referred to as an "opposition class" mission. The average stay on the surface for a conjunction class mission is 300 to 550 days while the stay is reduced to 30 to 80 days by flying past Venus in one of the legs of the mission. In this thesis, however, we would rather use the terms "direct" and "flyby" instead of "conjunction class" and "opposition class" since we look at the trajectories as a one-way, not as an entire mission.

1.2 Thesis Objectives and Outline

The objectives of this thesis are to develop a tool which is capable of calculating ballistic interplanetary trajectories including flyby opportunities as a general case, to

create the contours of the energy required for Mars trajectories, and to analyze them in terms of launch windows. As described later in Chapter 3, we use C3 to evaluate the energy required for the trajectories. The contours of C3 for a range of departure dates and arrival dates (or time of flight) would be useful for the creation of a long-term Earth-Mars and Mars-Earth transportation schedule for mission planning purposes. Therefore, the C3 contours produced in this thesis would serve as a calendar of Mars mission windows. This thesis focuses on the Mars missions in the time frame 2020 to 2040, during which such missions seem most relevant.

The theoretical basis of astrodynamics in preparation for the code implementation is described in Chapter 2. The implementation and validation of the MATLAB algorithms are presented in Chapter 3. The MATLAB scripts of each module and trajectory programs developed in this thesis work are listed in Appendix A. The simulation results are discussed in Chapter 4, followed by the conclusions and future work in Chapter 5. In Appendix B, the effect of inclination and eccentricity of the planetary orbits on the C3 contours is discussed using the hypothetical C3 contours with a coplanar, circular assumption.

Chapter 2

Astrodynamics

This chapter reviews the fundamental astrodynamics of classical two-body problems and the two key issues in this study, the Lambert's problems and planetary flybys, before we step into the implementation of MATLAB codes for interplanetary trajectory calculation. Using the vector-based approach to the classical problem of determining the motion of two bodies due solely to their own mutual gravitational attraction, several fundamental properties of the orbits are introduced.

2.1 Astrodynamics Overview

In this section, we develop the vector equation of two-body motion and derive the laws of conservation of angular momentum and energy and several orbit formulas. Here they are kept to the minimum necessary for the implementation of trajectory calculation codes. For more details on astrodynamics, refer to Battin [19], Curtis [20], and so on.

2.1.1 Equation of Relative Motion

The motion of two bodies governed solely by their mutual gravitational attraction is described by the following pair of nonlinear vector differential equations

$$m_1\ddot{\mathbf{r}}_1 = \frac{Gm_1m_2}{r_{12}^3}(\mathbf{r}_2 - \mathbf{r}_1) \tag{2.1a}$$

$$m_2\ddot{\mathbf{r}}_2 = \frac{Gm_2m_1}{r_{21}^3}(\mathbf{r}_1 - \mathbf{r}_2) \tag{2.1b}$$

together with a set of initial conditions such as the position vectors $\mathbf{r}_1(t_0)$, $\mathbf{r}_2(t_0)$ and the velocity vectors $\mathbf{v}_1(t_0)$, $\mathbf{v}_2(t_0)$ specified at some particular instant of time t_0. The equation describing the motion of m_2 relative to m_1 is readily obtained by differencing Eqs. (2.1) after first canceling the common mass factors. Thus, we have

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} \tag{2.2}$$

where

$$\mathbf{r} = \mathbf{r}_2 - \mathbf{r}_1 \tag{2.3}$$

is the position vector of m_2 relative to m_1 and

$$\mu = G(m_1 + m_2) \tag{2.4}$$

Eq. (2.2) is the fundamental differential equation of the two-body problem. Since $\dot{\mathbf{r}} = \mathbf{v}$ is the velocity of m_2 relative to m_1, we can write Eq. (2.2) as

$$\frac{d\mathbf{v}}{dt} = -\frac{\mu}{r^3}\mathbf{r} \tag{2.5}$$

2.1.2 Angular Momentum and Eccentricity

By taking the vector product of Eq. (2.5) with the position vector $\mathbf{r}$, we have

$$\mathbf{r} \times \frac{d\mathbf{v}}{dt} = \frac{d}{dt}(\mathbf{r} \times \mathbf{v}) - \mathbf{v} \times \mathbf{v} = \frac{d}{dt}(\mathbf{r} \times \mathbf{v}) \tag{2.6a}$$

$$\mathbf{r} \times \left(-\frac{\mu}{r^3}\mathbf{r}\right) = -\frac{\mu}{r^3}\mathbf{r} \times \mathbf{r} = \mathbf{0} \tag{2.6b}$$

so that

$$\frac{d}{dt}(\mathbf{r} \times \mathbf{v}) = \mathbf{0} \tag{2.7}$$

and, by integrating, obtain

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} \tag{2.8}$$

where $\mathbf{h}$ is the integration constant. The vector $\mathbf{h}$ is interpreted as a massless angular momentum. Therefore, the angular momentum is constant and the motion takes place in the plane $\mathbf{h} \cdot \mathbf{r} = 0$, which means that, at any given time, the position vector $\mathbf{r}$ and the velocity vector $\mathbf{v}$ lie in the same plane. Thus, the path of m_2 around m_1 lies in a single plane. Using the polar coordinate expression for $\mathbf{r}$ and $\mathbf{v}$, we have

$$\mathbf{r} = r\mathbf{i}_r \tag{2.9a}$$

$$\mathbf{v} = \frac{dr}{dt}\mathbf{i}_r + r\frac{d\theta}{dt}\mathbf{i}_\theta \tag{2.9b}$$

so that

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} = r^2\frac{d\theta}{dt}\mathbf{i}_z = h\mathbf{i}_z \tag{2.10}$$

During the differential time interval dt the position vector $\mathbf{r}$ sweeps out an area

$$dA = \frac{1}{2}rv_\theta dt = \frac{1}{2}r^2\frac{d\theta}{dt}dt \tag{2.11}$$

Therefore, using Eq. (2.10) we have

$$\frac{dA}{dt} = \frac{h}{2} \tag{2.12}$$

dA/dt is called the areal velocity, and according to Eq. (2.12) it is constant. This result is known as Kepler's second law of planetary motion: equal areas swept out in equal times.

The vector product of Eq. (2.5) with the angular momentum vector $\mathbf{h}$ yields

$$\frac{d\mathbf{v}}{dt} \times \mathbf{h} = \frac{d}{dt}(\mathbf{v} \times \mathbf{h}) - \mathbf{v} \times \frac{d\mathbf{h}}{dt} = \frac{d}{dt}(\mathbf{v} \times \mathbf{h}) \tag{2.13}$$

since $\mathbf{h}$ is a constant vector and also

$$-\frac{\mu}{r^3}\mathbf{r}\times\mathbf{h}=-\mu\frac{h}{r^2}\mathbf{i}_r\times\mathbf{i}_z=\mu\frac{h}{r^2}\mathbf{i}_\theta=\mu\frac{d\theta}{dt}\mathbf{i}_\theta=\mu\frac{d\mathbf{i}_r}{dt}=\mu\frac{d}{dt}\left(\frac{\mathbf{r}}{r}\right) \tag{2.14}$$

using Eq. (2.10). Therefore, we get

$$\frac{d}{dt}\left(\mathbf{v}\times\mathbf{h}\right)=\mu\frac{d}{dt}\left(\frac{\mathbf{r}}{r}\right) \tag{2.15}$$

or

$$\mu\frac{d}{dt}\left(\frac{\mathbf{v}\times\mathbf{h}}{\mu}-\frac{\mathbf{r}}{r}\right)=0 \tag{2.16}$$

which may be integrated to obtain

$$\mathbf{e}=\frac{\mathbf{v}\times\mathbf{h}}{\mu}-\frac{\mathbf{r}}{r} \tag{2.17}$$

The constant vector $\mathbf{e}$ is called the eccentricity vector since its magnitude e is the eccentricity of the orbit. Taking the dot product of both sides of Eq. (2.17) with the vector $\mathbf{h}$ yields

$$\mathbf{e}\cdot\mathbf{h}=\frac{(\mathbf{v}\times\mathbf{h})\cdot\mathbf{h}}{\mu}-\frac{\mathbf{r}\cdot\mathbf{h}}{r} \tag{2.18}$$

Since $\mathbf{v}\times\mathbf{h}$ is perpendicular to both $\mathbf{v}$ and $\mathbf{h}$, it follows that $(\mathbf{v}\times\mathbf{h})\cdot\mathbf{h}=0$. Likewise, since $\mathbf{h}=\mathbf{r}\times\mathbf{v}$ is perpendicular to both $\mathbf{r}$ and $\mathbf{v}$, it is true that $\mathbf{r}\cdot\mathbf{h}=0$. Therefore, we have $\mathbf{e}\cdot\mathbf{h}=0$, that is, $\mathbf{e}$ is perpendicular to $\mathbf{h}$, which means $\mathbf{e}$ must lie in the orbital plane.

2.1.3 Energy Integral and Orbit Equation

Calculating the magnitude of the eccentricity vector yields

$$e^2=\mathbf{e}\cdot\mathbf{e}=\frac{1}{\mu^2}\left(\mathbf{v}\times\mathbf{h}\right)\cdot\left(\mathbf{v}\times\mathbf{h}\right)-\frac{2}{\mu r}\mathbf{r}\cdot\left(\mathbf{v}\times\mathbf{h}\right)+1 \tag{2.19}$$

Since $\mathbf{h}$ and $\mathbf{v}$ are orthogonal, the first term can be calculated as

$$\frac{1}{\mu^2}\left(\mathbf{v}\times\mathbf{h}\right)\cdot\left(\mathbf{v}\times\mathbf{h}\right)=\frac{1}{\mu^2}\left|\mathbf{v}\times\mathbf{h}\right|^2=\frac{h^2v^2}{\mu^2} \tag{2.20}$$

For the second term of Eq. (2.19) we have, using the scalar triple product property $\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) = \mathbf{c} \cdot (\mathbf{a} \times \mathbf{b})$,

$$-\frac{2}{\mu r}\mathbf{r} \cdot (\mathbf{v} \times \mathbf{h}) = -\frac{2}{\mu r}\mathbf{h} \cdot (\mathbf{r} \times \mathbf{v}) = -\frac{2}{\mu r}\mathbf{h} \cdot \mathbf{h} = -\frac{2h^2}{\mu r} \tag{2.21}$$

Therefore, Eq. (2.19) can be expressed in the form

$$1 - e^2 = \frac{2h^2}{\mu r} - \frac{h^2 v^2}{\mu^2} = \frac{h^2}{\mu}\left(\frac{2}{r} - \frac{v^2}{\mu}\right) \tag{2.22}$$

The first factor

$$p = \frac{h^2}{\mu} \tag{2.23}$$

has the dimension of length and is known as the *parameter*. The second factor must be a constant of the motion. Thus, we define

$$a = \left(\frac{2}{r} - \frac{v^2}{\mu}\right)^{-1} \tag{2.24}$$

which has also the dimension of length and is called the semimajor axis as stated later. When expressed in the form

$$\frac{v^2}{2} - \frac{\mu}{r} = -\frac{\mu}{2a} \tag{2.25}$$

we can identify $v^2/2$ as the kinetic energy and $-\mu/r$ as the potential energy, considered to be zero at infinity. It follows that the quantity $-\mu/2a$ is the total energy which is a constant of the orbit. Eq. (2.25) is a statement of conservation of energy, namely, that the specific mechanical energy is the same at all points of the trajectory. When Eq. (2.25) is expressed in the equivalent form

$$v^2 = \mu\left(\frac{2}{r} - \frac{1}{a}\right) \tag{2.26}$$

the resulting relation is the energy integral, also known as the vis-viva integral.

Clearly from Eqs. (2.22)-(2.24), the quantities p, a, and e are related by

$$p = a\left(1 - e^2\right) \tag{2.27}$$

Since p is never negative, we can see that e must be less than one if a is positive and greater than one if a is negative.

Taking the dot product of both sides of Eq. (2.17) with $\mathbf{r}$ yields

$$\mathbf{r} \cdot \mathbf{e} = \frac{\mathbf{r} \cdot (\mathbf{v} \times \mathbf{h})}{\mu} - \frac{\mathbf{r} \cdot \mathbf{r}}{r} = \frac{h^2}{\mu} - \frac{r^2}{r} = p - r \tag{2.28}$$

From the definition of the dot product we have

$$\mathbf{r} \cdot \mathbf{e} = re\cos\theta \tag{2.29}$$

where θ is the angle called the true anomaly between the fixed vector $\mathbf{e}$ and the variable position vector $\mathbf{r}$. In terms of the eccentricity and the true anomaly, we may therefore write Eq. (2.28) as

$$re\cos\theta = p - r \tag{2.30}$$

or

$$r = \frac{p}{1 + e\cos\theta} \tag{2.31}$$

This is the orbit equation, and it defines the path of the body m_2 around m_1 with constants of p and e. The point corresponding to $\theta = 0$, at which r is a minimum, is called pericenter or periapsis.

To convert the orbit equation to rectangular Cartesian coordinates, let the xy plane be the plane of motion with the x axis directed along the eccentricity vector. Then if x, y are the coordinates of a point on the orbit, we have $x = r\cos\theta$ and $r = p - ex$ from Eq. (2.30). Therefore, for the case $e \neq 1$, we can use Eq. (2.27) to write

$$y^2 = r^2 - x^2 = (p - ex)^2 - x^2 = \left(1 - e^2\right)\left[a^2 - (x + ea)^2\right] \tag{2.32}$$

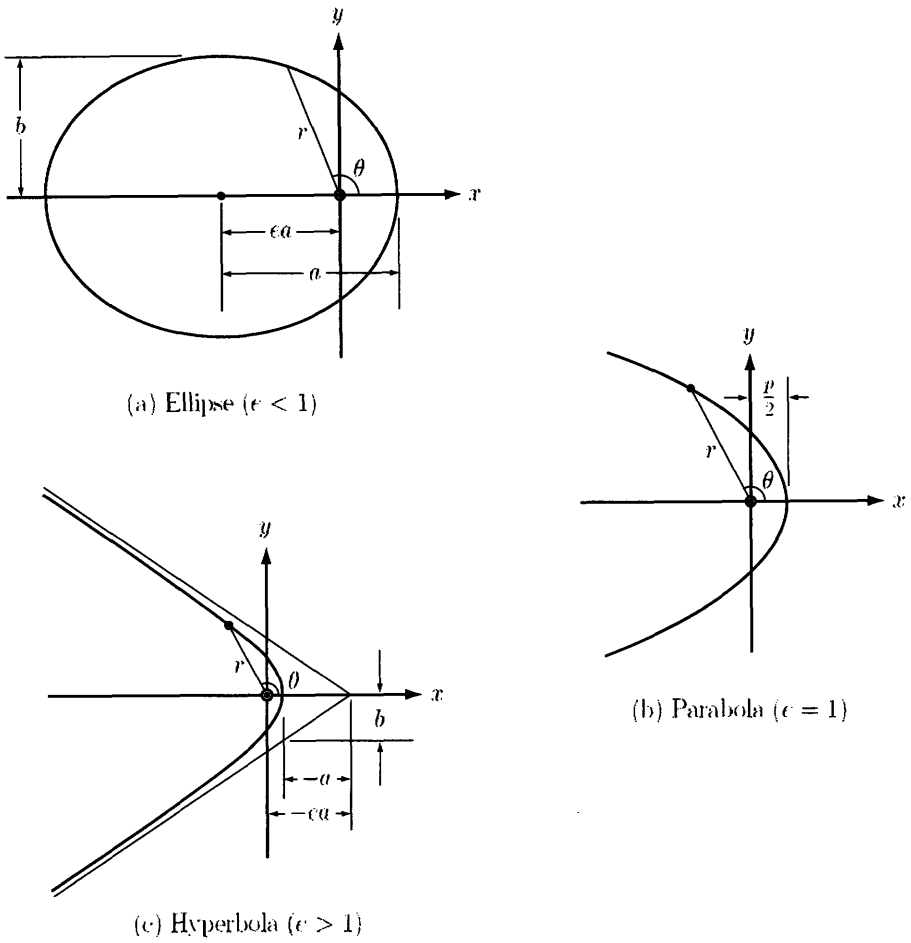


Figure 2-1: Conic sections

or

$$\frac{(x+ea)^2}{a^2} + \frac{y^2}{a^2(1-e^2)} = 1 \tag{2.33}$$

Eq. (2.33) represents a circle, ellipse, or hyperbola according as the eccentricity is zero, less than one, or greater than one. On the other hand, for the case $e = 1$, we have

$$y^2 = r^2 - x^2 = (p-x)^2 - x^2 \tag{2.34}$$

or

$$x = -\frac{1}{2p}y^2 + \frac{p}{2} \tag{2.35}$$

which represents a parabola. These cases are illustrated in Figures 2-1. The circle, ellipse, parabola, and hyperbola are often called conic sections because they can all

be obtained as sections cut from a right circular cone by a plane.

Because of its geometric significance, the energy constant a is termed the semimajor axis and is positive for ellipses, negative for hyperbolas, and infinite for parabolas. In astronomy, the semimajor axis of the Earth's orbit is frequently chosen as the unit of length called the astronomical unit.

2.1.4 Orbital Elements

To define an orbit in the plane requires two parameters: eccentricity and angular momentum. Other parameters, such as the semimajor axis and the specific energy are obtained from these two. To locate a point on the orbit requires a third parameter, the true anomaly, which leads us to the time since periapsis. Describing the orientation of an orbit in three dimensions requires three additional parameters, called the Euler angles, which are illustrated in Figure 2-2.

First, we locate the intersection of the orbital plane with the equatorial XY plane. That line is called the node line. The point on the node line where the orbit passes above the equatorial plane from below it is called the ascending node. The angle between the positive X axis and the node line is the first Euler angle Ω, the right ascension of the ascending node. The right ascension is a positive number lying between 0° and 360°.

The dihedral angle between the orbital plane and the equatorial plane is the inclination i, measured counterclockwise around the node line vector from the equator to the orbit. The inclination is also the angle between the positive Z axis and the normal to the plane of the orbit. Recall from Section 2.1.2 that the angular momentum vector $\mathbf{h}$ is normal to the plane of the orbit. Therefore, the inclination i is the angle between the positive Z axis and $\mathbf{h}$. The inclination is a positive number between 0° and 180°.

It remains to locate the periapsis of the orbit. Recall from Section 2.1.3 that periapsis lies at the intersection of the eccentricity vector $\mathbf{e}$ with the orbital path. The third Euler angle ω, the argument of periapsis, is the angle between the node line and the eccentricity vector $\mathbf{e}$, measured in the plane of the orbit. The argument

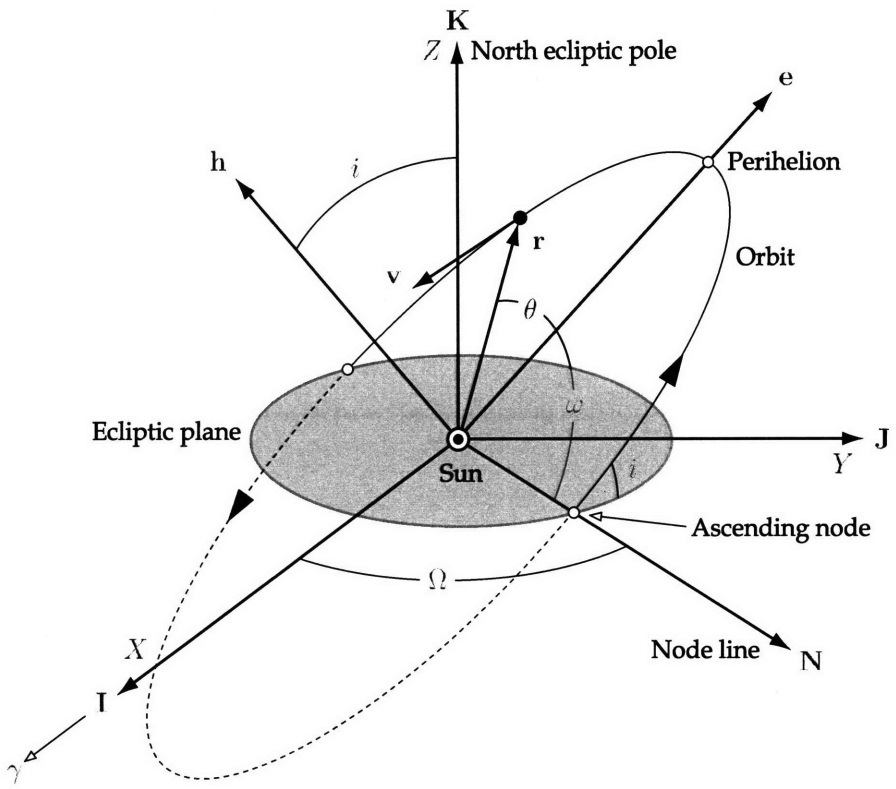


Figure 2-2: Orbit in the heliocentric ecliptic frame

of periapsis is a positive number between 0° and 360°.

In summary, the six orbital elements are angular momentum h, eccentricity e, right ascension (RA) of the ascending node Ω, inclination i, argument of periapsis ω, and true anomaly θ.

2.2 Lambert's Problem

When we think about interplanetary trajectory from planet to planet, we are given two position vectors and the time between them. Determination of an orbit, having a specified transfer time and connecting two position vectors, is called Lambert's problem. Solving Lambert's problem should define the orbital elements of the desired transfer orbit. In this section, we develop a solution to Lambert's problem to implement the trajectory calculation program.

2.2.1 Lambert's Theorem

One of the most remarkable theorems in astrodynamics is the one discovered by Lambert having to do with the time to traverse an elliptic arc. Lambert's theorem states that the orbital transfer time depends only upon the semimajor axis, the sum of the two radii, and the distance between the initial and final positions (chord length). If $t_2 - t_1$ is the time to describe the arc from the initial point P_1 to the final point P_2 and c is the length of the chord, then the theorem states that

$$\sqrt{\mu}\,(t_2 - t_1) = F\,(a, r_1 + r_2, c) \tag{2.36}$$

and the eccentricity is not involved. Since the two radii r_1, r_2 and the chord length c are already known from the problem definition, the semimajor axis a is the only unknown parameter. Thus, the problem is now to find the correct value of a that will give the desired transfer time.

The theorem is true not only for an ellipse but also for a general conic. The solution of the boundary-value problem like Eq. (2.36) will require an iterative process.

2.2.2 Lagrange Coefficients

In general, if the position and velocity of an orbiting body are known at a given instant, then the position and velocity at any later time can be expressed in terms of the initial values. Thus, the trajectory is determined once we find $\mathbf{v}_1$ because the position and velocity of any point on the path are determined by $\mathbf{r}_1$ and $\mathbf{v}_1$, that is,

$$\mathbf{r}_2 = f\mathbf{r}_1 + g\mathbf{v}_1 \tag{2.37a}$$

$$\mathbf{v}_2 = \dot{f}\mathbf{r}_1 + \dot{g}\mathbf{v}_1 \tag{2.37b}$$

The f and g functions are referred to as the Lagrange coefficients. Solving Eq. (2.37a) for $\mathbf{v}_1$ yields

$$\mathbf{v}_1 = \frac{1}{g}\,(\mathbf{r}_2 - f\mathbf{r}_1) \tag{2.38}$$

Substitute this result into Eq. (2.37b) to get

$$\mathbf{v}_2 = \dot{f}\mathbf{r}_1 + \frac{\dot{g}}{g}(\mathbf{r}_2 - f\mathbf{r}_1) = \frac{\dot{g}}{g}\mathbf{r}_2 - \frac{f\dot{g} - \dot{f}g}{g}\mathbf{r}_1 \tag{2.39}$$

Calculate $\mathbf{h}$ using Eqs. (2.37),

$$\mathbf{h}_2 = \mathbf{r}_2 \times \mathbf{v}_2 = (f\mathbf{r}_1 + g\mathbf{v}_1) \times (\dot{f}\mathbf{r}_1 + \dot{g}\mathbf{v}_1) \tag{2.40}$$

Expanding the right-hand side yields

$$\mathbf{h}_2 = (f\mathbf{r}_1 \times \dot{f}\mathbf{r}_1) + (f\mathbf{r}_1 \times \dot{g}\mathbf{v}_1) + (g\mathbf{v}_1 \times \dot{f}\mathbf{r}_1) + (g\mathbf{v}_1 \times \dot{g}\mathbf{v}_1) \tag{2.41}$$

Since $\mathbf{r}_1 \times \mathbf{r}_1 = \mathbf{v}_1 \times \mathbf{v}_1 = \mathbf{0}$, then we get, factoring out the scalars f, g, $\dot{f}$, and $\dot{g}$,

$$\mathbf{h}_2 = (f\dot{g} - \dot{f}g)(\mathbf{r}_1 \times \mathbf{v}_1) = (f\dot{g} - \dot{f}g)\mathbf{h}_1 \tag{2.42}$$

But the angular momentum is constant, which means $\mathbf{h}_2 = \mathbf{h}_1$, so that

$$f\dot{g} - \dot{f}g = 1 \tag{2.43}$$

Thus, substituting Eq. (2.43) into Eq. (2.39), we get

$$\mathbf{v}_2 = \frac{1}{g}(\dot{g}\mathbf{r}_2 - \mathbf{r}_1) \tag{2.44}$$

Clearly from Eqs. (2.38) and (2.44), Lambert's problem is solved once we determine the Lagrange coefficients f, g, and $\dot{g}$.

The Lagrange coefficients in terms of the change in true anomaly $\Delta\theta$ are

$$f = 1 - \frac{\mu r_2}{h^2}(1 - \cos\Delta\theta) \tag{2.45a}$$

$$g = \frac{r_1 r_2}{h}\sin\Delta\theta \tag{2.45b}$$

$$\dot{f} = \frac{\mu}{h}\frac{1-\cos\Delta\theta}{\sin\Delta\theta}\left[\frac{\mu}{h^2}(1-\cos\Delta\theta) - \frac{1}{r_1} - \frac{1}{r_2}\right] \tag{2.45c}$$

$$\dot{g} = 1 - \frac{\mu r_1}{h^2}(1 - \cos\Delta\theta) \tag{2.45d}$$

Solving Lambert's problem is equivalent to determining these Lagrange coefficients, which requires an iterative procedure.

2.2.3 Solution with Universal Variables

Over the years a variety of techniques for solving Lambert's problem has been developed, including Gauss' method, Battin-Vaughan algorithm [19], and so on. In order to solve Lambert's problem in more general way that is valid for all orbits, we introduce a universal variable, or universal anomaly χ. If we take advantage of the universal variable formulation, we can cover all of elliptical orbits, parabolas, and hyperbolas with the same set of Lagrange coefficients [21]. The Lagrange coefficients in terms of the universal anomaly χ are

$$f = 1 - \frac{\chi^2}{r_1}C(z) \tag{2.46a}$$

$$g = \Delta t - \frac{1}{\sqrt{\mu}}\chi^3 S(z) \tag{2.46b}$$

$$\dot{f} = \frac{\sqrt{\mu}}{r_1 r_2}\chi\left[zS(z) - 1\right] \tag{2.46c}$$

$$\dot{g} = 1 - \frac{\chi^2}{r_2}C(z) \tag{2.46d}$$

where $z = \alpha\chi^2$ and α is the reciprocal of the semimajor axis. Thus, $\alpha < 0$, $\alpha = 0$, and $\alpha > 0$ represent hyperbolas, parabolas, and ellipses, respectively. The functions

$C(z)$ and $S(z)$ belong to the class known as Stumpff functions, and they are related to the circular and hyperbolic trig functions as follows:

$$S(z) = \begin{cases} \dfrac{\sqrt{z} - \sin\sqrt{z}}{(\sqrt{z})^3} & (z > 0 : \text{ellipses}) \\ \dfrac{\sinh\sqrt{-z} - \sqrt{-z}}{(\sqrt{-z})^3} & (z < 0 : \text{hyperbolas}) \\ \dfrac{1}{6} & (z = 0 : \text{parabolas}) \end{cases} \tag{2.47}$$

$$C(z) = \begin{cases} \dfrac{1 - \cos\sqrt{z}}{z} & (z > 0 : \text{ellipses}) \\ \dfrac{\cosh\sqrt{-z} - 1}{-z} & (z < 0 : \text{hyperbolas}) \\ \dfrac{1}{2} & (z = 0 : \text{parabolas}) \end{cases} \tag{2.48}$$

The unknowns on the right-hand side of Eqs. (2.45) and (2.46) are h, χ, and z, whereas Δt, r_1, and r_2 are given.

By equating the two expressions for g in Eq. (2.45b) and Eq. (2.46b), we have

$$\frac{r_1 r_2}{h}\sin\Delta\theta = \Delta t - \frac{1}{\sqrt{\mu}}\chi^3 S(z) \tag{2.49}$$

To eliminate the unknown angular momentum h, equate the expressions for f in Eqs. (2.45a) and Eq. (2.46a),

$$1 - \frac{\mu r_2}{h^2}(1 - \cos\Delta\theta) = 1 - \frac{\chi^2}{r_1}C(z) \tag{2.50}$$

Upon solving this for h we obtain

$$h = \sqrt{\frac{\mu r_1 r_2 (1 - \cos\Delta\theta)}{\chi^2 C(z)}} \tag{2.51}$$

Substituting Eq. (2.51) into Eq. (2.49), simplifying and rearranging terms yields

$$\sqrt{\mu}\Delta t = \chi^3 S(z) + \chi\sqrt{C(z)}\left(\sin\Delta\theta\sqrt{\frac{r_1 r_2}{1 - \cos\Delta\theta}}\right) \tag{2.52}$$

The term in parentheses on the right is a constant comprised solely of the given parameters. If we assign it the symbol A,

$$A = \sin\Delta\theta\sqrt{\frac{r_1 r_2}{1-\cos\Delta\theta}} \tag{2.53}$$

then Eq. (2.52) is written in the simpler form

$$\sqrt{\mu}\Delta t = \chi^3 S(z) + A\chi\sqrt{C(z)} \tag{2.54}$$

The right side of this equation contains both of the unknown variables χ and z. We cannot use the fact that $z = \alpha\chi^2$ to reduce the unknowns to one since α is the reciprocal of the semimajor axis of the unknown orbit. In order to find a relationship between z and χ which does not involve orbital parameters, we equate the expressions for $\dot{f}$ in Eqs. (2.45c) and Eq. (2.46c) to obtain

$$\frac{\mu}{h}\frac{1-\cos\Delta\theta}{\sin\Delta\theta}\left[\frac{\mu}{h^2}(1-\cos\Delta\theta) - \frac{1}{r_1} - \frac{1}{r_2}\right] = \frac{\sqrt{\mu}}{r_1 r_2}\chi\left[zS(z) - 1\right] \tag{2.55}$$

Substituting for the angular momentum using Eq. (2.51) and dividing out common factors leads us to

$$\frac{\sqrt{1-\cos\Delta\theta}}{\sqrt{r_1 r_2}\sin\Delta\theta}\sqrt{C(z)}\left[\chi^2 C(z) - r_1 - r_2\right] = zS(z) - 1 \tag{2.56}$$

We recognize the reciprocal of A on the left, so we can rearrange this expression to read as follows,

$$\chi^2 C(z) = r_1 + r_2 + A\frac{zS(z) - 1}{\sqrt{C(z)}} \tag{2.57}$$

The right-hand side depends exclusively on z. Define the left-hand side as $y(z)$, so that

$$\chi = \sqrt{\frac{y(z)}{C(z)}} \tag{2.58}$$

where

$$y(z) = r_1 + r_2 + A\frac{zS(z) - 1}{\sqrt{C(z)}} \tag{2.59}$$

Eq. (2.58) is the relation between χ and z that we were seeking. Substituting it back into Eq. (2.54) yields

$$\sqrt{\mu}\Delta t = \left[\frac{y(z)}{C(z)}\right]^{\frac{3}{2}} S(z) + A\sqrt{y(z)} \tag{2.60}$$

We can use this equation to solve for z, given the time interval Δt. It must be done iteratively. Using Newton's method, we form the function

$$F(z) = \left[\frac{y(z)}{C(z)}\right]^{\frac{3}{2}} S(z) + A\sqrt{y(z)} - \sqrt{\mu}\Delta t \tag{2.61}$$

and its differential function

$$F'(z) = \frac{1}{2\sqrt{yC^5}}\left\{\left[2CS' - 3C'S\right]y^2 + \left[AC^{\frac{5}{2}} + 3CSy\right]y'\right\} \tag{2.62}$$

where C' and S' are the differential functions of the Stumpff functions, which are given by

$$S'(z) = \frac{1}{2z}\left[C(z) - 3S(z)\right] \tag{2.63}$$

$$C'(z) = \frac{1}{2z}\left[1 - zS(z) - 2C(z)\right] \tag{2.64}$$

y' is obtained by differentiating $y(z)$ in Eq. (2.59),

$$y' = \frac{A}{2C^{\frac{3}{2}}}\left\{\left[1 - zS\right]C' + 2\left[S + zS'\right]C\right\} \tag{2.65}$$

If we substitute Eqs. (2.63) and (2.64) into this expression, we get a much simpler form

$$y'(z) = \frac{A}{4}\sqrt{C(z)} \tag{2.66}$$

Substituting Eq. (2.66) along with Eqs. (2.63) and (2.64) into Eq. (2.62) yields

$$F' = \begin{cases} \left[\frac{y}{C}\right]^{\frac{3}{2}} \left\{\frac{1}{2z}\left[C - \frac{3}{2}\frac{S}{C}\right] + \frac{3}{4}\frac{S^2}{C}\right\} + \frac{A}{8}\left[3\frac{S}{C}\sqrt{y} + A\sqrt{\frac{C}{y}}\right] & (z \neq 0) \\ \frac{\sqrt{2}}{40}y(0)^{\frac{3}{2}} + \frac{A}{8}\left[\sqrt{y(0)} + A\sqrt{\frac{1}{2y(0)}}\right] & (z = 0) \end{cases} \tag{2.67}$$

$F(z)$ in Eq. (2.61) and $F'(z)$ in Eq. (2.67) are used in Newton's formula for the iterative procedure,

$$z_{i+1} = z_i - \frac{F(z_i)}{F'(z_i)} \tag{2.68}$$

Once we find the correct value of z, we get the Lagrange coefficients. Substituting Eqs. (2.58) and (2.60) into Eqs. (2.46) yields

$$f = 1 - \frac{y(z)}{r_1} \tag{2.69a}$$

$$g = A\sqrt{\frac{y(z)}{\mu}} \tag{2.69b}$$

$$\dot{f} = \frac{\sqrt{\mu}}{r_1 r_2}\sqrt{\frac{y(z)}{C(z)}}\left[zS(z) - 1\right] \tag{2.69c}$$

$$\dot{g} = 1 - \frac{y(z)}{r_2} \tag{2.69d}$$

In Chapter 3, we develop a MATLAB implementation of this method to solve Lambert's problem.

2.3 Planetary Flyby

A planetary flyby or gravity assist is the use of the relative movement and gravity of a planet to alter the path and speed of a spacecraft, typically in order to save fuel. Planetary flyby can be used to decelerate a spacecraft when traveling to an inner planet or accelerate a spacecraft when traveling to an outer planet. When we design

an interplanetary trajectory from planet to planet using as little fuel as possible, we might be able to take advantage of this flyby option. In this section, we formulate the flyby technique to develop a MATLAB program.

2.3.1 Sphere of Influence

First of all, we need to consider a planet's gravitational sphere of influence. The Sun, of course, is the dominant celestial body in the solar system. It is over 1,000 times more massive than the largest planet, Jupiter, and has a mass of over 300,000 Earths. However, near a planet, the influence of the planet's own gravity exceeds that of the Sun. According to the inverse-square nature of the law of gravity, the force of gravity drops off rapidly with distance from the center of attraction. Eventually, the force of the Sun's gravitational field overwhelms that of the planet. At some point, the perturbing effect of the Sun on the vehicle's orbit around the planet overtakes the perturbing effect of the planet on the vehicle's orbit around the Sun. The surface boundary over which these two effects are equal is almost spherical. This boundary surface is the concept of the sphere of influence, which originated with Laplace.

Considering the three-body system comprising a planet of mass m_P, the Sun of mass $m_\odot$, and a vehicle allows us to estimate the radius of a planet's gravitational sphere of influence as follows.

$$r_{SOI} = a_P \left(\frac{m_P}{m_\odot} \right)^{\frac{2}{5}} \tag{2.70}$$

where r_{SOI} is the radius of the sphere of influence and a_P is the semimajor axis of the planet. Within the planet's sphere of influence defined by r_{SOI}, the motion of the spacecraft is determined by its equations of motion relative to the planet. Outside of the sphere of influence, the path of the spacecraft is computed relative to the Sun.

The spheres of influence of the planets of the solar system are listed in Table 2.1. For each of the three planets, Earth, Venus, and Mars, which we focus on in this study, we can see from the table that the radius of sphere of influence is less than one percent of the semimajor axis, which implies that these three spheres of influence

Table 2.1: Sphere of influence (SOI) radius of the planets

Planet	a_P [au]	$m_P/m_\odot$	r_{SOI} [km]	r_{SOI}/a_P [%]
Mercury	0.38709893	0.00000017	112,000	0.1960
Venus	0.72333199	0.00000245	616,000	0.5697
Earth	1.00000011	0.000002999	925,000	0.6177
Mars	1.52366231	0.00000032	577,000	0.2524
Jupiter	5.20336301	0.000954786	48,200,000	6.1939
Saturn	9.53707032	0.000285584	54,800,000	3.8220
Uranus	19.19126393	0.000043727	51,800,000	1.8043
Neptune	30.06896348	0.000051776	86,600,000	1.9304

are very small relative to the interplanetary space in which a spacecraft will traverse. Therefore, during most of the time of flight, a spacecraft is exposed to the Sun's gravitational field only. It follows that basically, we only have to consider the Sun's gravitational field when we design an interplanetary trajectory.

2.3.2 Planetary Flyby Orbits

A spacecraft which enters a planet's sphere of influence and does not impact the planet or go into orbit around it will continue in its hyperbolic trajectory through periapsis and exit the sphere of influence. At the inbound crossing point, the hyperbolic excess velocity $\mathbf{V}_{\infty i}$ of the spacecraft (relative to the planet) equals the heliocentric velocity $\mathbf{v}_i$ of the spacecraft minus the planet's heliocentric velocity $\mathbf{v}_P$,

$$\mathbf{V}_{\infty i} = \mathbf{v}_i - \mathbf{v}_P \tag{2.71}$$

Similarly, at the outbound crossing we have

$$\mathbf{V}_{\infty o} = \mathbf{v}_o - \mathbf{v}_P \tag{2.72}$$

Let 2ν and r_m be the required turn angle and the minimum passing distance, respectively. The turn angle 2ν is readily computed from the inbound and outbound

relative velocity vectors. Thus

$$\sin 2\nu = \frac{|\mathbf{V}_{\infty o} \times \mathbf{V}_{\infty i}|}{V_\infty^2} \tag{2.73}$$

where

$$V_\infty = V_{\infty i} = V_{\infty o} = \sqrt{\frac{\mu_P}{-a}} \tag{2.74}$$

From the fact that the true anomaly at infinity is $\nu + 90°$, we have

$$\cos(\nu + 90°) = -\frac{1}{e} \tag{2.75}$$

so that

$$\csc \nu = e \tag{2.76}$$

The minimum passing distance can then be determined from Eq. (2.31),

$$r_m = \frac{a(1-e^2)}{1+e} = a(1-e) = \frac{\mu_P}{V_\infty^2}(e-1) = \frac{\mu_P}{V_\infty^2}(\csc \nu - 1) \tag{2.77}$$

If r_m is of reasonable magnitude, the solution is complete and a satisfactory flyby path has been found.

2.3.3 Powered Flyby Maneuver

In Section 2.3.2, we assumed that $\mathbf{V}_{\infty i}$ and $\mathbf{V}_{\infty o}$ are of equal magnitude so that the spacecraft does not need to fire its rocket engine. But an opportunity of such a free flyby is rare. Instead, if we make some small burn at flyby, we might be able to expand the flyby mission window, which would be useful for future mission planning purposes. Figure 2-3 shows a hyperbolic flyby trajectory with a velocity impulse at periapsis. In this study, for simplicity we consider the velocity impulse at periapsis only. In general, if we want to impart the maximum amount of kinetic energy to a spacecraft whose velocity varies with time, we should do it when it's moving the fastest. During a planetary flyby, this happens at periapsis, the closest approach to the planet. In this section, we derive some relations between r_m, ΔV_{PFM}, and

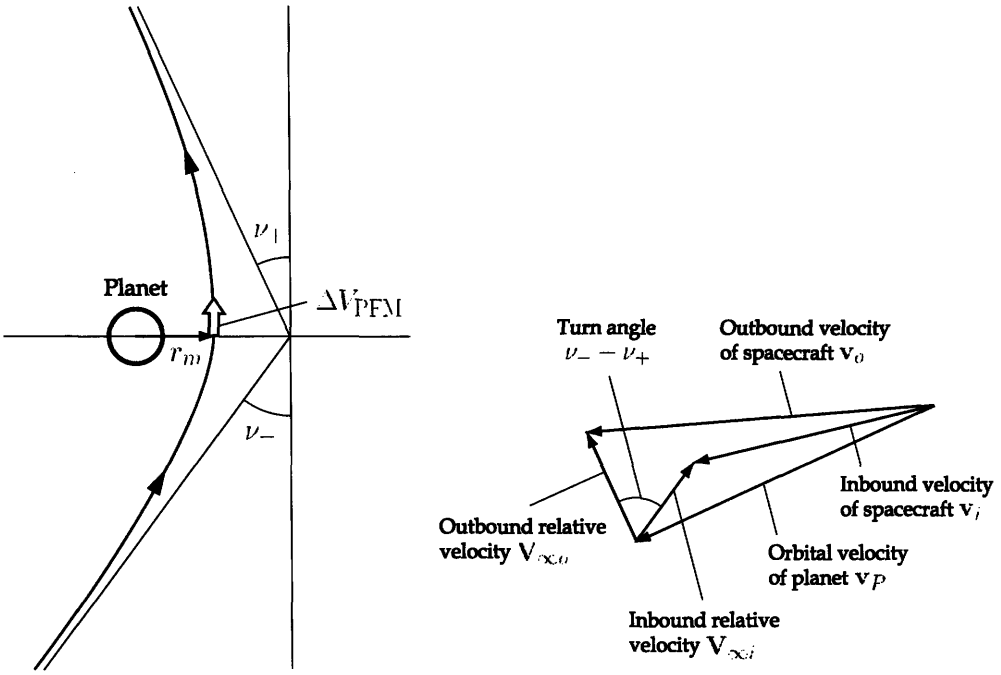


Figure 2-3: Planetary flyby

the velocity vectors $\mathbf{V}_{\infty i}$ and $\mathbf{V}_{\infty o}$ as a more general case of flybys with a powered maneuver at periapsis.

As illustrated in Figure 2-3, let the '$-$' and '$+$' subscripts represent the pre-periapsis and post-periapsis trajectories, respectively. Thus, the turn angle is $\nu_- + \nu_+$ instead of the sum of two identical angles 2ν. According to the inbound and outbound velocity vectors $\mathbf{V}_{\infty i}$, $\mathbf{V}_{\infty o}$, the total turn angle $\nu_- + \nu_+$ must satisfy

$$\sin(\nu_- + \nu_+) = \frac{|\mathbf{V}_{\infty o} \times \mathbf{V}_{\infty i}|}{V_{\infty o} V_{\infty i}} \tag{2.78}$$

Since the pre-periapsis and post-periapsis trajectories intersect at periapsis r_m, we get, using Eq. (2.77) for each trajectory,

$$r_m = \frac{\mu_P}{V_{\infty i}^2}(\csc \nu_- - 1) \tag{2.79a}$$

$$r_m = \frac{\mu_P}{V_{\infty o}^2}(\csc \nu_+ - 1) \tag{2.79b}$$

so that

$$\sin \nu_- = \frac{1}{1 + \frac{V_{\infty i}^2}{V_{om}^2}} \tag{2.80a}$$

$$\sin \nu_+ = \frac{1}{1 + \frac{V_{\infty o}^2}{V_{om}^2}} \tag{2.80b}$$

where V_{om} is a circular speed at radius r_m so that

$$V_{om}^2 = \frac{\mu_P}{r_m} \tag{2.81}$$

From Eqs. (2.78) and (2.80), we get an implicit equation for r_m

$$\sin^{-1}\left(\frac{|\mathbf{V}_{\infty o} \times \mathbf{V}_{\infty i}|}{V_{\infty o} V_{\infty i}}\right) = \sin^{-1}\left(\frac{1}{1 + \frac{V_{\infty i}^2}{V_{om}^2}}\right) + \sin^{-1}\left(\frac{1}{1 + \frac{V_{\infty o}^2}{V_{om}^2}}\right) \tag{2.82}$$

In Eq. (2.82), the only unknown parameter is r_m through V_{om}. Therefore, we can solve Eq. (2.82) to determine r_m by an iterative function such as the MATLAB `"fzero"` function.

Once we determine r_m, we can compute the velocity of spacecraft at periapsis on the pre- and post-periapsis trajectories, V_{m-} and V_{m+}. Thus, using Eq. (2.26), we have

$$V_{m-} = \sqrt{\mu_P\left(\frac{2}{r_m} - \frac{1}{a_-}\right)} = \sqrt{V_{\infty i}^2 + \frac{2\mu_P}{r_m}} \tag{2.83a}$$

$$V_{m+} = \sqrt{\mu_P\left(\frac{2}{r_m} - \frac{1}{a_+}\right)} = \sqrt{V_{\infty o}^2 + \frac{2\mu_P}{r_m}} \tag{2.83b}$$

By differentiating Eqs. (2.83), we obtain the change in velocity required at periapsis

$$\Delta V_{\text{PFM}} = V_{m+} - V_{m-} = \sqrt{V_{\infty o}^2 + \frac{2\mu_P}{r_m}} - \sqrt{V_{\infty i}^2 + \frac{2\mu_P}{r_m}} \tag{2.84}$$

which is positive when accelerating and negative when decelerating. A free flyby in Section 2.3.2 corresponds to $\Delta V_{\text{PFM}} = 0$ since $V_{\infty i} = V_{\infty o}$. Ideally, ΔV_{PFM} should be

zero so that we would not need to consume any fuel. But it would be worthwhile to look into the possibility of powered flyby, since allowing a small amount of ΔV_{PFM} might get us a much broader mission window.

Chapter 3

Interplanetary Trajectory Calculation Program

In Chapter 2, we reviewed some astrodynamics key issues that are needed for an interplanetary trajectory design including a planetary flyby. Based on the knowledge, this chapter presents the implementation of MATLAB algorithm on a module by module basis, the overall flowchart of those modules for a trajectory calculation, and the code validation by comparison with the data from Jet Propulsion Laboratory.

3.1 Module Implementation

An interplanetary trajectory calculation program developed in this study is a modularly-configured program. This section explains the implementation of MATLAB code for each module based on the knowledge from Chapter 2. Each MATLAB algorithm is presented in Appendix A.

3.1.1 Time of Flight

To design an interplanetary trajectory, we need to begin with the selection of the date of event, such as departure, encounter (if applicable), and arrival. For the computation of the time of flight and the generation of the planetary ephemeris, it

is convenient to introduce the Julian day number, which is the number of days since noon UT on 1 January 4713 BC. Since the Julian day count is uniform and continuous and does not involve leap years or different numbers of days in different months, we can specify the date by one-to-one correspondence and find the number of days between two events by simply subtracting the Julian day of one from that of the other.

J_0 is the symbol for the Julian day number at 0 hr UT. At any other UT, the Julian day is given by

$$JD = J_0 + \frac{UT}{24} \tag{3.1}$$

One of the simplest formulas for obtaining J_0 from the ordinary year (y), month (m), and day (d) is given by

$$J_0 = 367y - \text{INT}\left[\frac{7}{4}\left(y + \text{INT}\left[\frac{m+9}{12}\right]\right)\right] + \text{INT}\left[\frac{275m}{9}\right] + d + 1721013.5 \tag{3.2}$$

where y, m, and d are integers lying in the following ranges

$$\begin{aligned} 1901 \leq y \leq 2099 \\ 1 \leq m \leq 12 \\ 1 \leq d \leq 31 \end{aligned} \tag{3.3}$$

INT $[x]$ means to retain only the integer portion of x without rounding. Appendices A.1 and A.2 list MATLAB scripts for the Julian day number and the time of flight, respectively.

3.1.2 Planetary Ephemeris

The state vector $\mathbf{r}$, $\mathbf{v}$ of a planet is defined relative to the heliocentric ecliptic frame of reference as illustrated in Figure 2-2. The J2000 vernal equinox defines the inertial X axis. In order to design an interplanetary trajectory, we must be able to determine the state vector of a planet at any given time.

In preparation for the planetary ephemeris generation, we need two algorithms:

Newton's method to solve Kepler's equation and the computation of the state vector $\mathbf{r}$ and $\mathbf{v}$ from the orbital elements h, e, Ω, i, ω, and θ.

The relationship between mean anomaly M_e and eccentric anomaly E, known as Kepler's equation, is written as

$$M_e = E - e \sin E \tag{3.4}$$

If we are given e and M_e, then substituting them into Kepler's equation we get the following expression for E,

$$E - e \sin E = M_e \tag{3.5}$$

We cannot solve this transcendental equation directly for E. An accurate solution requires an iterative procedure such as Newton's method represented by Eq. (2.68). To apply Newton's method to the solution of Kepler's equation, we form the function

$$f(E) = E - e \sin E - M_e \tag{3.6}$$

and seek the value of eccentric anomaly that makes $f(E) = 0$. Since

$$f'(E) = 1 - e \cos E \tag{3.7}$$

for this problem, Eq. (2.68) becomes

$$E_{i+1} = E_i - \frac{E_i - e \sin E_i - M_e}{1 - e \cos E_i} \tag{3.8}$$

Algorithm 3.1

Given the eccentricity e and the mean anomaly M_e, solve Kepler's equation for the eccentric anomaly E. Appendix A.3 lists a MATLAB implementation of this algorithm.

1. Choose an initial estimate of the root E as follows. If $M_e < \pi$, then $E = M_e + e/2$. If $M_e > \pi$, then $E = M_e - e/2$.

2. At any given step, having obtained E_i from the previous step, calculate $f(E_i) = E_i - e\sin E_i - M_e$ and $f'(E_i) = 1 - e\cos E_i$.
3. Calculate $\text{ratio}_i = f(E_i)/f'(E_i)$.
4. If $|\text{ratio}_i|$ exceeds the chosen tolerance (10^{-8}), then calculate an updated value of E

$$E_{i+1} = E_i - \text{ratio}_i \tag{3.9}$$

Return to step 2.

5. If $|\text{ratio}_i|$ is less than the tolerance, then accept E_i as the solution to within the chosen accuracy.

We still need another algorithm to compute the state vector $\mathbf{r}$ and $\mathbf{v}$ from the orbital elements h, e, Ω, i, ω, and θ. Since the orbit lies in the $\bar{x}\bar{y}$ plane of the perifocal frame, the components of the state vector of a body relative to its perifocal reference can be written

$$\{\mathbf{r}\}_{\bar{x}} = \frac{h^2}{\mu_\odot}\frac{1}{1+e\cos\theta}\begin{bmatrix}\cos\theta\\ \sin\theta\\ 0\end{bmatrix} \tag{3.10}$$

$$\{\mathbf{v}\}_{\bar{x}} = \frac{\mu_\odot}{h}\begin{bmatrix}-\sin\theta\\ e+\cos\theta\\ 0\end{bmatrix} \tag{3.11}$$

where the subscript $\bar{x}$ represents the perifocal $\bar{x}\bar{y}\bar{z}$ coordinate system.

The transformation from the perifocal frame into the heliocentric ecliptic frame can be accomplished by the sequence of three rotations through the Euler angles Ω, i, and ω. The transformation matrix $\mathbb{Q}_{\bar{x}X}$ from the perifocal $\bar{x}\bar{y}\bar{z}$ into the heliocentric ecliptic XYZ is the product of the three orthogonal rotation matrices,

$$\mathbb{Q}_{\bar{x}X} \equiv \begin{bmatrix}\cos\Omega & -\sin\Omega & 0\\ \sin\Omega & \cos\Omega & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & \cos i & -\sin i\\ 0 & \sin i & \cos i\end{bmatrix}\begin{bmatrix}\cos\omega & -\sin\omega & 0\\ \sin\omega & \cos\omega & 0\\ 0 & 0 & 1\end{bmatrix} \tag{3.12}$$

If the components of the state vector are given in the heliocentric ecliptic frame

$$\begin{aligned} \mathbf{r} &= \{\mathbf{r}\}_X = \begin{bmatrix} X & Y & Z \end{bmatrix}^T \\ \mathbf{v} &= \{\mathbf{v}\}_X = \begin{bmatrix} v_X & v_Y & v_Z \end{bmatrix}^T \end{aligned} \tag{3.13}$$

the transformation from perifocal to heliocentric ecliptic components is

$$\begin{aligned} \{\mathbf{r}\}_X &= \mathbb{Q}_{\bar{x}X} \{\mathbf{r}\}_{\bar{x}} \\ \{\mathbf{v}\}_X &= \mathbb{Q}_{\bar{x}X} \{\mathbf{v}\}_{\bar{x}} \end{aligned} \tag{3.14}$$

Algorithm 3.2

Given the orbital elements h, e, Ω, i, ω, and θ, compute the state vector $\mathbf{r}$ and $\mathbf{v}$ in the heliocentric ecliptic frame of reference. A MATLAB implementation of this procedure is listed in Appendix A.4.

1. Calculate position vector $\{\mathbf{r}\}_{\bar{x}}$ in perifocal coordinates using Eq. (3.10).
2. Calculate velocity vector $\{\mathbf{v}\}_{\bar{x}}$ in perifocal coordinates using Eq. (3.11).
3. Calculate the matrix $\mathbb{Q}_{\bar{x}X}$ of the transformation from perifocal to heliocentric ecliptic coordinates using Eq. (3.12).
4. Transform $\{\mathbf{r}\}_{\bar{x}}$ and $\{\mathbf{v}\}_{\bar{x}}$ into the heliocentric frame by means of Eq. (3.14).

After having obtained Algorithms 3.1 and 3.2, we step into the planetary ephemeris generation. Table 3.1 provides the orbital elements of the planets and their rates of change per century with respect to the J2000 epoch (1 January 2000, 12 hr UT). The table, covering the years 1800 to 2050, is sufficiently accurate for our needs.

In order to interpret Table 3.1, note the following: 1 astronomical unit (1 au) is 149,597,870.691 km, the average distance between the Earth and the Sun. 1 arcsecond (1") is 1/3600 of a degree. Ω is the right ascension of the ascending node relative to the J2000 vernal equinox. $\tilde{\omega}$, the longitude of perihelion, is defined as $\tilde{\omega} = \omega + \Omega$, where ω is the argument of perihelion. L, the mean longitude, is defined as $L = \tilde{\omega} + M$, where M is the mean anomaly. $\dot{a}$, $\dot{e}$, $\dot{\Omega}$, etc., are the rates of change of the orbital elements per Julian century. 1 century (Cy) equals 36,525 days.

Table 3.1: Planetary orbital elements and their centennial rates

Planet	a [au] $\dot{a}$ [au/Cy]	e $\dot{e}$ [1/Cy]	i [deg] $\dot{i}$ ["/Cy]	Ω [deg] $\dot{\Omega}$ ["/Cy]	$\tilde{\omega}$ [deg] $\dot{\tilde{\omega}}$ ["/Cy]	L [deg] $\dot{L}$ ["/Cy]
Mercury	0.38709893	0.20563069	7.00487	48.33167	77.45645	252.25084
	0.00000066	0.00002527	-23.51	-446.30	573.57	538,101,628.29
Venus	0.72333199	0.00677323	3.39471	76.68069	131.53298	181.97973
	0.00000092	-0.00004938	-2.86	-996.89	-108.80	210,664,136.06
Earth	1.00000011	0.01671022	0.00005	-11.26064	102.94719	100.46435
	-0.00000005	-0.00003804	-46.94	-18228.25	1198.28	129597740.63
Mars	1.52366231	0.09341233	1.85061	49.57854	336.04084	355.45332
	-0.00007221	0.00011902	-25.47	-1020.19	1560.78	68905103.78
Jupiter	5.20336301	0.04839266	1.30530	100.55615	14.75385	34.40438
	0.00060737	-0.00012880	-4.15	1217.17	839.93	10925078.35
Saturn	9.53707032	0.05415060	2.48446	113.71504	92.43194	49.94432
	-0.00301530	-0.00036762	6.11	-1591.05	-1948.89	4401052.95
Uranus	19.19126393	0.04716771	0.76986	74.22988	170.96424	313.23218
	0.00152025	-0.00019150	-2.09	-1681.4	1312.56	1542547.79
Neptune	30.06896348	0.00858587	1.76917	131.72169	44.97135	304.88003
	-0.00125196	0.00002514	-3.64	-151.25	-844.43	786449.21
Pluto	39.48168677	0.24880766	17.14175	110.30347	224.06676	238.92881
	-0.00076912	0.00006465	11.07	-37.33	-132.25	522747.90

Algorithm 3.3

Determine the state vector of a planet at a given date and time. All angular calculations must be adjusted so that they lie in the range 0° to 360°. The gravitational parameter of the Sun is $\mu_\odot = 4\pi^2$ (au^3/year^2). This procedure is implemented in MATLAB in Appendix A.5.

1. Use Eqs. (3.1) and (3.2) to calculate the Julian day number JD.

2. Calculate T_0, the number of Julian centuries between J2000 and the date in question

$$T_0 = \frac{JD - 2451545}{36525} \tag{3.15}$$

3. If Q is any one of the six planetary orbital elements listed in Table 3.1, then

calculate its value at JD by means of the formula

$$Q = Q_0 + \dot{Q}T_0 \tag{3.16}$$

where Q_0 is the value listed for J2000 and $\dot{Q}$ is the tabulated rate. All angular quantities must be adjusted to lie in the range 0° to 360°.

4. Use the semimajor axis a and the eccentricity e to calculate the angular momentum h at JD from Eqs. (2.23) and (2.27)

$$h = \sqrt{\mu_\odot a\left(1 - e^2\right)} \tag{3.17}$$

5. Obtain the argument of perihelion ω and mean anomaly M at JD from the results of step 3 by means of the definitions

$$\omega = \tilde{\omega} - \Omega \tag{3.18}$$

$$M = L - \tilde{\omega} \tag{3.19}$$

6. Substitute the eccentricity e and the mean anomaly M at JD into Kepler's equation and calculate the eccentric anomaly E.
7. Calculate the true anomaly θ using the relationship between the true anomaly θ and the eccentric anomaly E

$$\theta = 2\tan^{-1}\left(\sqrt{\frac{1+e}{1-e}}\tan\frac{E}{2}\right) \tag{3.20}$$

8. Use h, e, Ω, i, ω, and θ to obtain the heliocentric position vector $\mathbf{r}$ and velocity vector $\mathbf{v}$ by means of Algorithm 3.2.

3.1.3 Lambert's Problem

As stated in Section 2.2, given the time of flight and the position vectors of the departure and arrival points, we must solve Lambert's problem to determine an orbit. To solve Lambert's problem with universal variables, we need to implement the Stumpff

functions represented by Eqs. (2.47) and (2.48). Appendix A.6 lists MATLAB implementations of these two functions. According to Section 2.2.3, the procedure of Lambert's problem solution with universal variables is described as follows.

Algorithm 3.4

Solve Lambert's problem. A MATLAB implementation appears in Appendix A.7. Given $\mathbf{r}_1$, $\mathbf{r}_2$ and Δt, the steps are:

1. Calculate r_1 and r_2 by taking the absolute values of $\mathbf{r}_1$ and $\mathbf{r}_2$, respectively.
2. Calculate $\Delta\theta$ by

$$\Delta\theta = \begin{cases} \cos^{-1}\left(\dfrac{\mathbf{r}_1 \cdot \mathbf{r}_2}{r_1 r_2}\right) & \text{if } (\mathbf{r}_1 \times \mathbf{r}_2)_Z \geq 0 \\ 360^\circ - \cos^{-1}\left(\dfrac{\mathbf{r}_1 \cdot \mathbf{r}_2}{r_1 r_2}\right) & \text{if } (\mathbf{r}_1 \times \mathbf{r}_2)_Z < 0 \end{cases} \tag{3.21}$$

3. Calculate A in Eq. (2.53).
4. By iteration, using Eqs. (2.61), (2.67), and (2.68), solve Eq. (2.60) for z.
5. Calculate y using Eq. (2.59).
6. Calculate the Lagrange f, g, $\dot{f}$, and $\dot{g}$ functions using Eq. (2.69).
7. Calculate $\mathbf{v}_1$ and $\mathbf{v}_2$ from Eqs. (2.38) and (2.44).

3.1.4 Orbital Elements

As opposed to Algorithm 3.2, we also need an algorithm that can compute the orbital elements h, e, Ω, i, ω, and θ for a given state vector $\mathbf{r}$ and $\mathbf{v}$. The step-by-step procedure to determine the six orbital elements is outlined in Algorithm 3.5. Note that each notation appears in Figure 2-2.

Algorithm 3.5

Obtain orbital elements from the state vector. A MATLAB implementation of this procedure appears in Appendix A.8.

1. Calculate r by taking the absolute value of $\mathbf{r}$.
2. Calculate v by taking the absolute value of $\mathbf{v}$.

3. Calculate the radial velocity,

$$v_r = \frac{\mathbf{r} \cdot \mathbf{v}}{r} \tag{3.22}$$

Note that if $v_r \geq 0$, the spacecraft is flying away from perihelion. If $v_r < 0$, it is flying towards perihelion.

4. Calculate the angular momentum vector $\mathbf{h}$ from Eq. (2.8).
5. Calculate the first orbital element h by taking the absolute value of $\mathbf{h}$.
6. Calculate the inclination,

$$i = \cos^{-1}\left(\frac{h_Z}{h}\right) \tag{3.23}$$

This is the second orbital element. Note that i must lie between 0° and 180°.

7. Calculate

$$\mathbf{N} = \mathbf{K} \times \mathbf{h} \tag{3.24}$$

This vector defines the node line.

8. Calculate N by taking the absolute value of $\mathbf{N}$.
9. Calculate the RA of the ascending node,

$$\Omega = \begin{cases} \cos^{-1}\left(\dfrac{N_X}{N}\right) & \text{if } N_Y \geq 0 \\ 360° - \cos^{-1}\left(\dfrac{N_X}{N}\right) & \text{if } N_Y < 0 \end{cases} \tag{3.25}$$

This is the third orbital element. Note that the range of the $\cos^{-1}$ function is usually defined between 0° and 180°. The ascending node lies on the positive side of the vertical XZ plane ($0° \leq \Omega \leq 180°$) if $N_Y \geq 0$, whereas the ascending node lies on the negative side of the XZ plane ($180° < \Omega < 360°$) if $N_Y < 0$.

10. Calculate the eccentricity vector $\mathbf{e}$ from Eq. (2.17).
11. Calculate the fourth orbital element e by taking the absolute value of $\mathbf{e}$.
12. Calculate the argument of perihelion,

$$\omega = \begin{cases} \cos^{-1}\left(\dfrac{\mathbf{N} \cdot \mathbf{e}}{Ne}\right) & \text{if } e_Z \geq 0 \\ 360° - \cos^{-1}\left(\dfrac{\mathbf{N} \cdot \mathbf{e}}{Ne}\right) & \text{if } e_Z < 0 \end{cases} \tag{3.26}$$

This is the fifth orbital element. Perihelion lies above the ecliptic plane ($0° \leq \omega \leq 180°$) if $\mathbf{e}$ points up (in the positive Z direction), and perihelion lies below the plane ($180° < \Omega < 360°$) if $\mathbf{e}$ points down.

13. Calculate the true anomaly,

$$\theta = \begin{cases} \cos^{-1}\left(\dfrac{\mathbf{e}\cdot\mathbf{r}}{er}\right) & \text{if } v_r \geq 0 \\ 360° - \cos^{-1}\left(\dfrac{\mathbf{e}\cdot\mathbf{r}}{er}\right) & \text{if } v_r < 0 \end{cases} \tag{3.27}$$

This is the sixth orbital element. If the spacecraft is flying away from perihelion ($v_r \geq 0$), then $0° \leq \theta \leq 180°$, whereas if the spacecraft is flying towards perihelion ($v_r < 0$), then $180° < \theta < 360°$.

14. Calculate the semimajor axis and the period,

$$a = \frac{h^2}{\mu_\odot}\frac{1}{1-e^2} \tag{3.28}$$

$$T = \begin{cases} 2\pi\sqrt{\dfrac{a^3}{\mu_\odot}} & \text{if } e < 1 \\ \infty & \text{if } e \geq 1 \end{cases} \tag{3.29}$$

3.1.5 Planetary Flyby

Given the encounter date for planetary flyby as well as the departure and arrival dates, we can determine the inbound and outbound velocity vectors at a flyby planet in the heliocentric perspective ($\mathbf{v}_i$ and $\mathbf{v}_o$ in Figure 2-3) by solving two Lambert's problems separately. Once we find $\mathbf{v}_i$ and $\mathbf{v}_o$, we can get the inbound and outbound relative velocities $\mathbf{V}_{\infty i}$ and $\mathbf{V}_{\infty o}$ from Eqs. (2.71) and (2.72), since we know the orbital velocity of a flyby planet $\mathbf{v}_P$ on the encounter date.

We need an algorithm that is capable of handling not only a free flyby but also a powered flyby. In other words, given the inbound and outbound relative velocities $\mathbf{V}_{\infty i}$ and $\mathbf{V}_{\infty o}$, the algorithm can determine a velocity impulse required at periapsis ΔV_{PFM} as well as the minimum passing distance r_m. As stated in Section 2.3.3,

the MATLAB "fzero" function is applied to Eq. (2.82) to solve for r_m. For the "fzero" computation to quickly converge, we give the lower and upper bounds for r_m, which can be obtained analytically.

Let $V_{\infty\min}$ and $V_{\infty\max}$ represent the smaller one and the larger one of the two velocities $V_{\infty i}$ and $V_{\infty o}$, respectively

$$\begin{aligned} V_{\infty\min} &= \min\left(V_{\infty i}, V_{\infty o}\right) \\ V_{\infty\max} &= \max\left(V_{\infty i}, V_{\infty o}\right) \end{aligned} \tag{3.30}$$

If we form the function of r

$$f(r) = \sin^{-1}\left(\frac{1}{1+\frac{V_{\infty\min}^2}{\mu_P}r}\right) + \sin^{-1}\left(\frac{1}{1+\frac{V_{\infty\max}^2}{\mu_P}r}\right) \tag{3.31}$$

then, from Eqs. (2.78)-(2.82) we have

$$f(r_m) = \nu_- + \nu_+ \tag{3.32}$$

Since the $\sin^{-1}$ term in Eq. (3.31) is a monotonically decreasing function of V_∞^2, then we have

$$f_{\min}(r) \leq f(r) \leq f_{\max}(r) \tag{3.33}$$

where

$$\begin{aligned} f_{\min}(r) &= 2\sin^{-1}\left(\frac{1}{1+\frac{V_{\infty\max}^2}{\mu_P}r}\right) \\ f_{\max}(r) &= 2\sin^{-1}\left(\frac{1}{1+\frac{V_{\infty\min}^2}{\mu_P}r}\right) \end{aligned} \tag{3.34}$$

We know from Eqs. (3.31) and (3.34) that f, $f_{\min}$, and $f_{\max}$ are monotonically decreasing functions of r and the inequality (3.33) is true for any positive real r.

Therefore, if we define $r_{\min}$ and $r_{\max}$ such that

$$f_{\min}(r_{\min}) = f(r_m) = f_{\max}(r_{\max}) = \nu_- + \nu_+ \tag{3.35}$$

then we have

$$r_{\min} \leq r_m \leq r_{\max} \tag{3.36}$$

We can use this inequality as the bounds for r_m in the `"fzero"` computation. Substituting Eq. (3.35) into Eqs. (3.34) yields

$$\begin{aligned} r_{\min} &= \frac{\mu_P}{V_{\infty\max}^2}\left[\csc\left(\frac{\nu_- + \nu_+}{2}\right) - 1\right] \\ r_{\max} &= \frac{\mu_P}{V_{\infty\min}^2}\left[\csc\left(\frac{\nu_- + \nu_+}{2}\right) - 1\right] \end{aligned} \tag{3.37}$$

where

$$\nu_- + \nu_+ = \sin^{-1}\left(\frac{|\mathbf{V}_{\infty o} \times \mathbf{V}_{\infty i}|}{V_{\infty o} V_{\infty i}}\right) \tag{3.38}$$

Algorithm 3.6

Given inbound and outbound velocity vectors relative to the flyby planet $\mathbf{V}_{\infty i}$ and $\mathbf{V}_{\infty o}$, compute a minimum passing altitude h_m and a velocity impulse required for powered flyby maneuver ΔV_{PFM}. Since r_m can be solved only implicitly via Eq. (2.82), this algorithm uses the MATLAB `"fzero"` function with the bounds given by Eqs. (3.37). A MATLAB implementation appears in Appendix A.9.

1. Calculate $V_{\infty i}$ and $V_{\infty o}$ by taking the absolute values of $\mathbf{V}_{\infty i}$ and $\mathbf{V}_{\infty o}$, respectively.
2. Use Eq. (3.38) to calculate the total turn angle $\nu_- + \nu_+$.
3. Calculate the bounds $r_{\min}$ and $r_{\max}$ by Eqs. (3.37).
4. Use the `"fzero"` function to calculate r_m.
5. Calculate h_m by subtracting the flyby planet's radius r_P from r_m.
6. Calculate ΔV_{PFM} from Eq. (2.84).

3.2 Trajectory Program Implementation

In this section, we integrate the modules described in Section 3.1 into two complete trajectory programs: one is a direct trajectory from planet 1 to planet 2, and the other is a flyby trajectory from planet 1 to planet 3 via planet 2.

3.2.1 Hyperbolic Excess Velocity and C3

Before stepping into the trajectory program implementation, we introduce the hyperbolic excess velocity and C3 to quantitatively evaluate the calculated trajectory.

The vector difference between the velocity required on the transfer trajectory and the velocity of a departure planet with respect to the Sun is called the hyperbolic excess velocity. If $\mathbf{v}_{SC}$ is the spacecraft's departure velocity vector with respect to the Sun and $\mathbf{v}_P$ is the departure planet's velocity vector with respect to the Sun at the time, then the hyperbolic excess velocity is

$$\mathbf{V}_\infty = \mathbf{v}_{SC} - \mathbf{v}_P \tag{3.39}$$

The hyperbolic excess velocity is important because it is a measure of the energy required from the launch vehicle system. It is traditional to use C3, which is V_∞^2, as a measure of the minimum energy requirement needed to accomplish the mission. Launch will be feasible if the launch vehicle C3 capability is above the C3 required for the mission. Therefore, C3 as the property of the transfer trajectory is one of the most important indicators of the mission feasibility.

Although C3 is typically the term for departure, we define C3 not only for departure but also for arrival in this study since the arrival C3 is related to the aerocapture feasibility at an arrival planet's atmosphere. If we define $C3_d$ as the departure C3 and $C3_a$ as the arrival C3,

$$C3_d = V_{\infty d}^2 \tag{3.40a}$$

$$C3_a = V_{\infty a}^2 \tag{3.40b}$$

where

$$V_{\infty d} = |\mathbf{V}_{\infty d}| = |\mathbf{v}_d - \mathbf{v}_1| \tag{3.41a}$$

$$V_{\infty a} = |\mathbf{V}_{\infty a}| = |\mathbf{v}_a - \mathbf{v}_2| \tag{3.41b}$$

and $\mathbf{v}_1$ is the departure planet's velocity vector at departure, $\mathbf{v}_2$ is the arrival planet's velocity vector at arrival, $\mathbf{v}_d$ and $\mathbf{v}_a$ are the spacecraft's velocity vectors at departure and arrival, respectively.

Both $C3_d$ and $C3_a$ are the key drivers for mission design. Since the C3 contours for a range of departure dates and arrival dates as shown in Chapter 4 can visualize the mission windows directly, they would be useful for mission planning purposes.

3.2.2 Direct Trajectory

The mission of a direct trajectory is to send a spacecraft directly from planet 1 to planet 2 in a specified time between the departure date and the arrival date. The flow chart in Figure 3-1 shows the overall structure of this procedure.

Algorithm 3.7

Given planet 1, planet 2, and the departure and arrival dates, determine the direct trajectory from planet 1 to planet 2 and $C3_d$ and $C3_a$. This procedure is implemented in MATLAB in Appendix A.10.

1. Calculate the time of flight as described in Section 3.1.1.
2. Use Algorithm 3.3 to determine the state vector $\mathbf{r}_1$ and $\mathbf{v}_1$ of planet 1 at departure and the state vector $\mathbf{r}_2$ and $\mathbf{v}_2$ of planet 2 at arrival.
3. Determine the position vectors of spacecraft at departure and arrival $\mathbf{r}_d$ and $\mathbf{r}_a$ by copying $\mathbf{r}_1$ and $\mathbf{r}_2$, respectively.
4. Use $\mathbf{r}_d$, $\mathbf{r}_a$, and the time of flight in Algorithm 3.4 to find the spacecraft velocity $\mathbf{v}_d$ at departure from planet 1's sphere of influence and its velocity $\mathbf{v}_a$ upon arrival at planet 2's sphere of influence.
5. Use Algorithm 3.5 to calculate the orbital elements of transfer trajectory.
6. Calculate $\mathbf{V}_{\infty d}$, $\mathbf{V}_{\infty a}$, $C3_d$, and $C3_a$ from Eqs. (3.40) and (3.41).

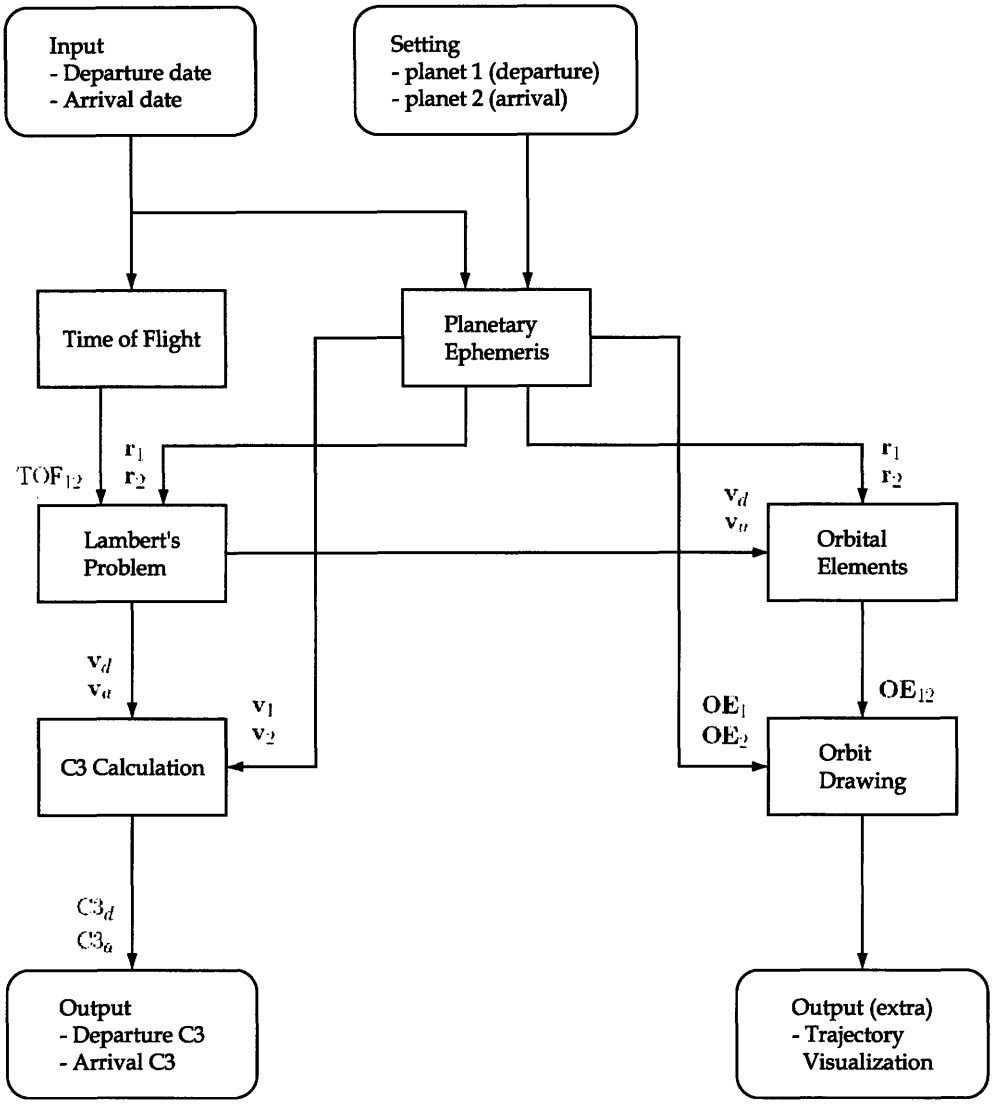


Figure 3-1: Flow chart of direct trajectory program

A mission example of a direct trajectory from Earth to Mars is shown below. Figure 3-2 shows the trajectory of this example, which is viewed from the ecliptic north pole. The motions of the planets and spacecraft proceed counterclockwise.

Departure date:	t_1 = May 9, 2003
Arrival date:	t_2 = December 29, 2003
Time of flight:	TOF = 234 [days]
Departure C3:	$C3_d = 12.6509$ [km^2/s^2]
Arrival C3:	$C3_a = 8.2671$ [km^2/s^2]

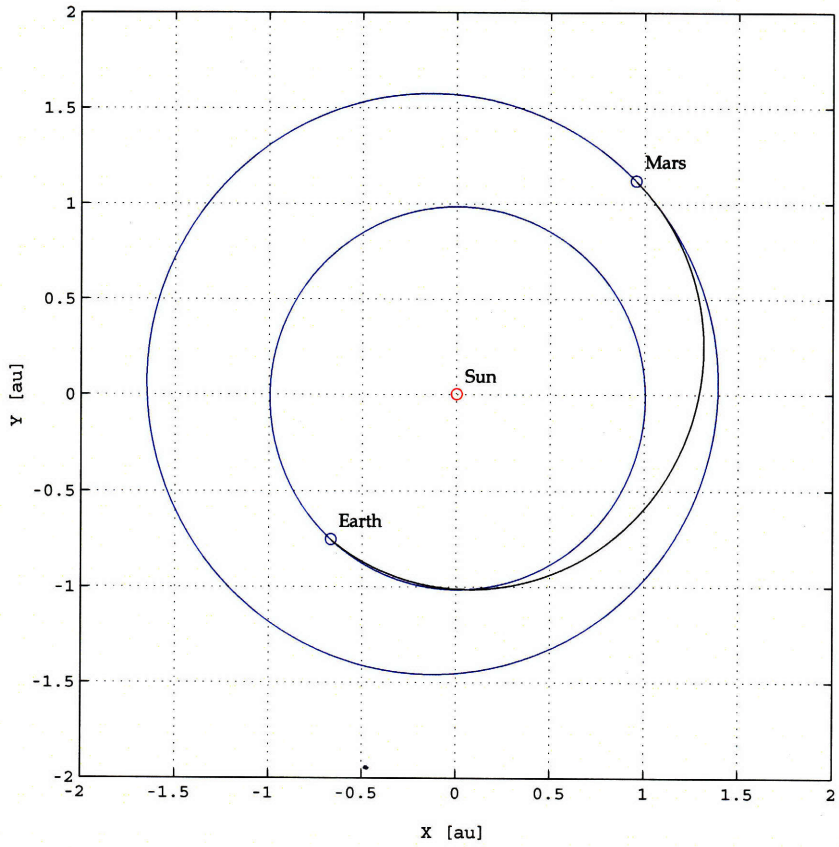


Figure 3-2: Direct trajectory example viewed from ecliptic north pole

3.2.3 Flyby Trajectory

The mission of a flyby trajectory is to send a spacecraft from planet 1 to planet 3 via planet 2 in each specified time between the departure date and the encounter date and between the encounter date and the arrival date. The flyby mission is broken down into three parts: the two separate ballistic trajectory phases and the flyby phase. Therefore, we solve two Lambert's problems separately to obtain $C3_d$ and $C3_a$, and then use Algorithm 3.6 to compute h_m and ΔV_{PFM}. The flow chart in Figure 3-3 shows the overall structure of this procedure.

Algorithm 3.8

Given planet 1, planet 2, planet 3, and the departure, encounter, and arrival dates, determine the flyby trajectory from planet 1 to planet 3 via planet 2 and $C3_d$, $C3_a$, h_m, and ΔV_{PFM}. This procedure is implemented in MATLAB in Appendix A.11.

1. Calculate the time of flight for each of the two Lambert's problems TOF_{12} and TOF_{23} by means described in Section 3.1.1.

2. Use Algorithm 3.3 to determine the state vector $\mathbf{r}_1$ and $\mathbf{v}_1$ of planet 1 at departure, the state vector $\mathbf{r}_2$ and $\mathbf{v}_2$ of planet 2 at encounter, and the state vector $\mathbf{r}_3$ and $\mathbf{v}_3$ of planet 3 at arrival.

3. Determine the position vectors of spacecraft at departure, encounter, and arrival $\mathbf{r}_d$, $\mathbf{r}_e$, and $\mathbf{r}_a$ by copying $\mathbf{r}_1$, $\mathbf{r}_2$, and $\mathbf{r}_3$, respectively.

4. Use $\mathbf{r}_d$, $\mathbf{r}_e$, and the time of flight TOF_{12} in Algorithm 3.4 to find the spacecraft velocity $\mathbf{v}_d$ at departure from planet 1's sphere of influence and its velocity $\mathbf{v}_{e12}$ upon arrival at planet 2's sphere of influence.

5. Repeat step 4 for the other Lambert's problem.

6. Use Algorithm 3.5 to calculate the orbital elements OE_{12} and OE_{23}.

7. Calculate $\mathbf{V}_{\infty d}$, $\mathbf{V}_{\infty a}$, C3_d, and C3_a from Eqs. (3.40) and (3.41).

8. Obtain the physical data of planet 2 r_P and μ_P.

9. Calculate $\mathbf{V}_{\infty i}$ and $\mathbf{V}_{\infty o}$ by

$$\mathbf{V}_{\infty i} = \mathbf{v}_{e12} - \mathbf{v}_2 \tag{3.42}$$

$$\mathbf{V}_{\infty o} = \mathbf{v}_{e23} - \mathbf{v}_2 \tag{3.43}$$

10. Use Algorithm 3.6 to calculate h_m and ΔV_{PFM}.

A mission example of a flyby trajectory from Earth to Mars via Venus is shown below. Figure 3-4 shows the trajectory of this example, in which the motions of the planets and spacecraft proceed counterclockwise.

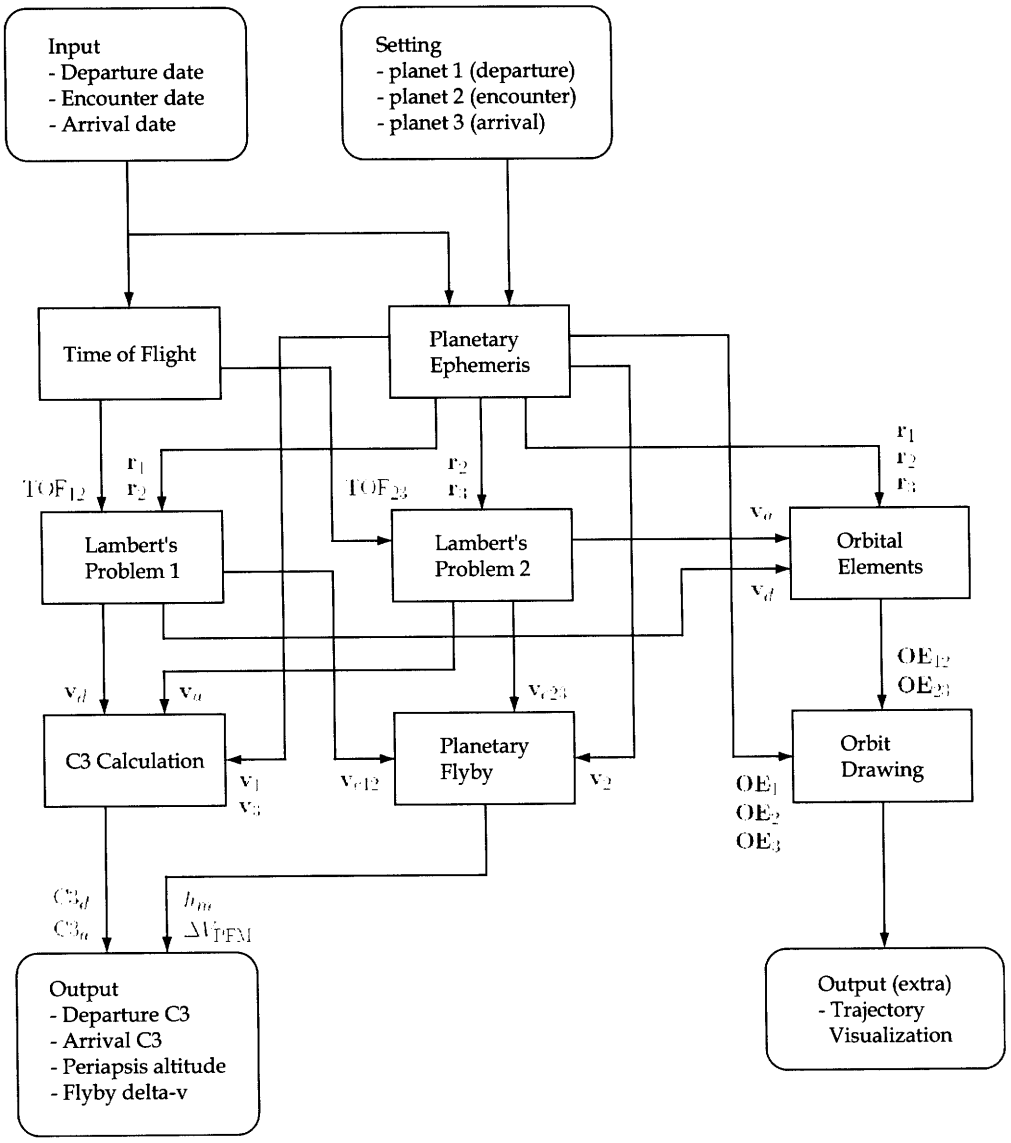


Figure 3-3: Flow chart of flyby trajectory program

Departure date:	$t_1 =$ August 6, 2002
Encounter date:	$t_2 =$ December 16, 2002
Arrival date:	$t_3 =$ June 9, 2003
Time of flight (Earth-Venus):	$\mathrm{TOF}_{12} = 132$ [days]
Time of flight (Venus-Mars):	$\mathrm{TOF}_{23} = 175$ [days]
Departure C3:	$\mathrm{C3}_d = 12.3245$ [km^2/s^2]
Arrival C3:	$\mathrm{C3}_a = 51.9276$ [km^2/s^2]
Minimum passing altitude:	$h_m = 2061.6$ [km]
Powered flyby ΔV:	$\Delta V_{\mathrm{PFM}} = -3.8879 \times 10^{-4}$ [km/s]

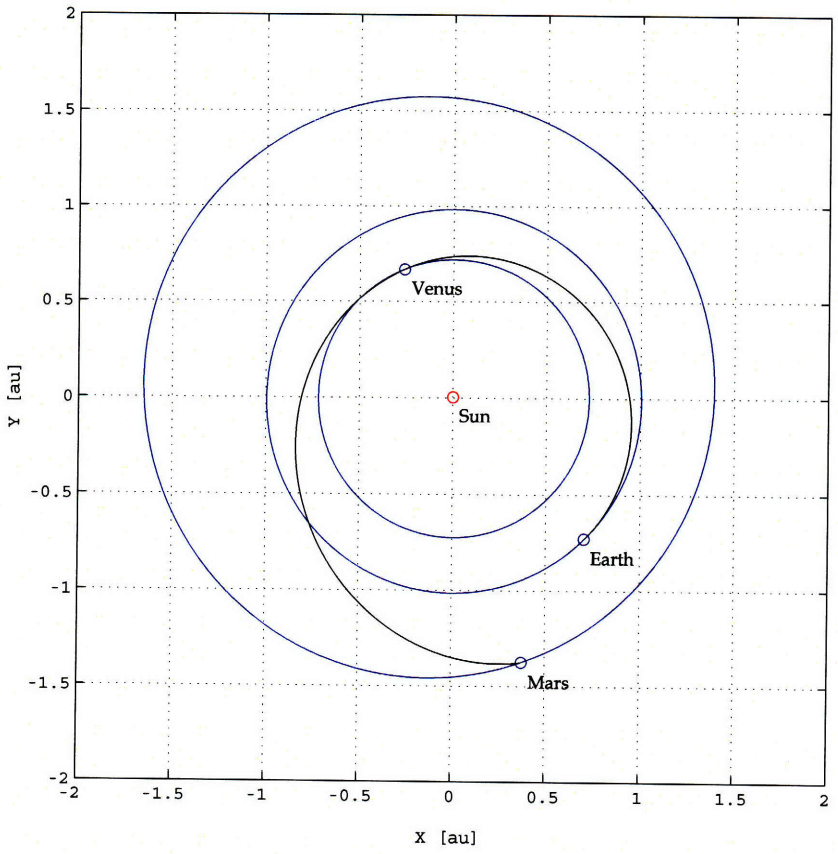


Figure 3-4: Flyby trajectory example viewed from ecliptic north pole

3.3 Code Validation

Having implemented the two complete trajectory programs, we need to verify that these programs would produce reliable results. Therefore, we check the validity of the code with the existing data. For code validation, this section compares the trajectory data produced by our programs with those given by Jet Propulsion Laboratory (JPL).

3.3.1 2002-2020 Mars Trajectory

The trajectory data for ballistic Earth to Mars and Mars to Earth trajectories for the years 2002 through 2020 are presented by JPL [22]. In the JPL's study, $C3_d$ and $V_{\infty a}$ for 2002-2020 Mars trajectories are given in the form of table along with the departure and arrival dates. Therefore, we performed the trajectory calculations for each set of departure and arrival dates. The comparison is listed in Table 3.2.

For simplicity, the departure and arrival hours were set at 12 hr UT for all of our calculations. As for each flyby trajectory (No. 1, 6, 24, 27, 28, 33, and 38 in Table 3.2), we picked an optimal encounter date such that ΔV_{PFM} would be nearly zero for a given set of departure and arrival dates. The calculated data were rounded to one decimal place to be aligned with those of JPL.

Table 3.2 shows that the excellent agreement was obtained; the errors are at most within a few percent. Some part of the errors are due to the displacement of the departure and arrival hours since they were all set at 12 hr UT in this study. Thus, the trajectory programs of the present study are expected to give reliable results also for 2020-2040 Mars trajectories.

3.3.2 C3 Contours

To visualize the mission windows on the calendar, it would be useful to create the C3 contours for a range of departure and arrival dates. JPL presented such plots for the Venus missions of the years 1975-1990 [23]. Figure 3-5 shows the JPL's C3 contour plot for the Venus 1988 opportunity and Figure 3-6 is the equivalent produced by our program. Since x axis represents the departure date and y axis represents the arrival date, a specific time of flight is represented by the 45° line. We can see that these two figures are in excellent agreement in position of the contour lines and minimum C3 points.

As we can see from Figures 3-5 and 3-6, there are two regions of minimum C3. To discuss why the minimum C3 regions are split into the two "craters," we fixed the departure date at March 28, 1988 and scanned the arrival dates along the vertical line in Figure 3-6. Figure 3-7 shows the two trajectories for the arrival on July 13 and September 17. These two represent the minimum C3 trajectories for the departure on March 28, which drop on the floor of each crater in Figure 3-6.

Planetary trajectories are classified based on the length of the transfer ellipse. If the spacecraft travels less than a 180° true anomaly, the trajectory is called type I. If the spacecraft travels more than 180° and less than 360°, the trajectory is called type II. Type III, IV, and beyond exist but are seldom used. The two trajectories shown

in Figure 3-7 are classified into type I and type II, respectively.

Figure 3-8 shows the three-dimensional view of the trajectories for the arrival on August 15, 18, and 22 as well as July 13 and September 17. As we can see from this figure, the orbital planes of August trajectories are highly inclined against the ecliptic plane. As the Venus changes its position from July through September, the transfer orbit, whose plane must include the three points (Earth, Venus, and Sun), would gradually tilt down to vertical, flip over at a certain time in August, and then again gradually tilt down to horizontal. This is due to the slightly different inclinations between the orbits of the Earth and Venus; if the orbits of the planets were coplanar, the plane of the transfer orbit would fit in the planets' orbital planes. For more details about the effect of inclination, see Appendix B.

Highly inclined transfer orbit requires high energy for plane change at departure. This causes the "mountain wall" between the two craters in Figure 3-6. As described in Chapter 4, this is also the case for Mars trajectories.

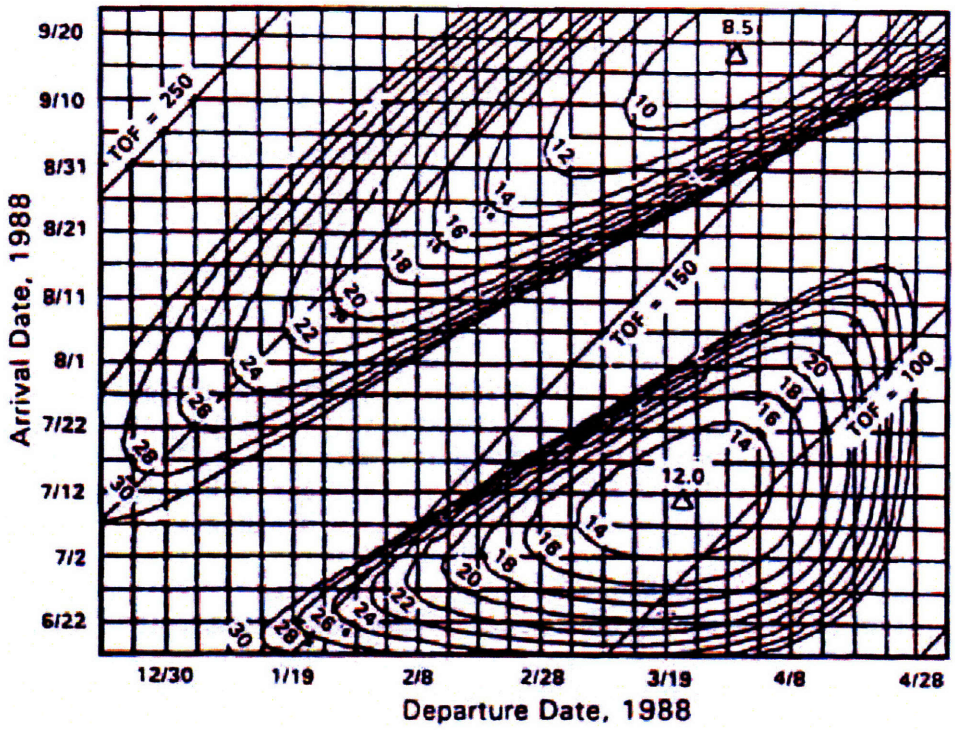


Figure 3-5: C3 contours: 1988 Earth-Venus (JPL)

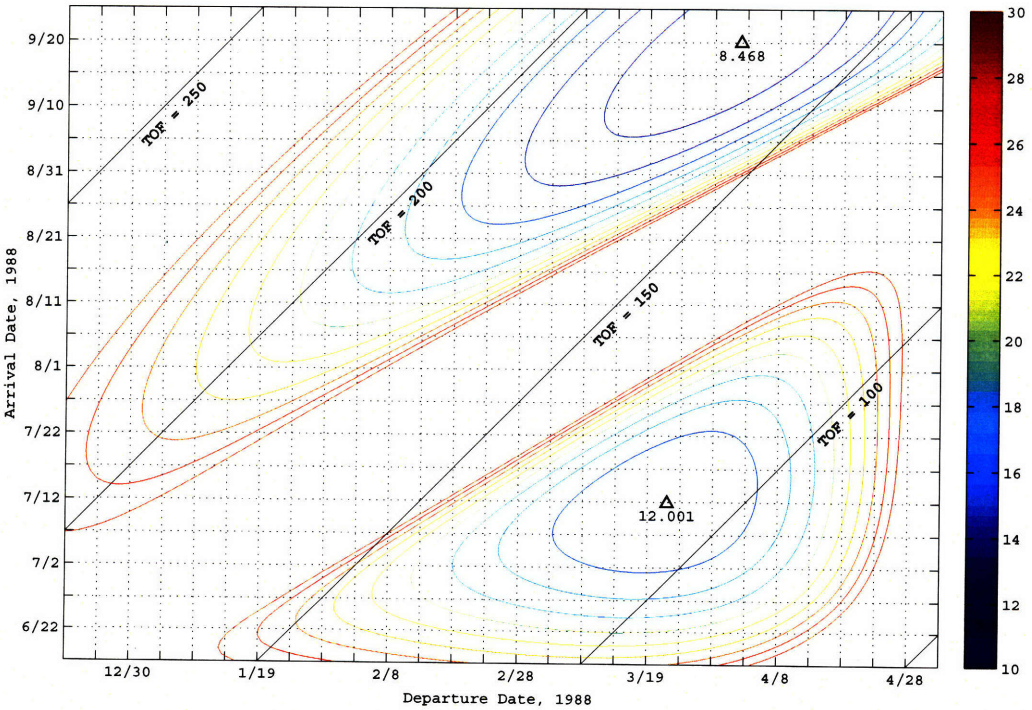


Figure 3-6: C3 contours: 1988 Earth-Venus (present)

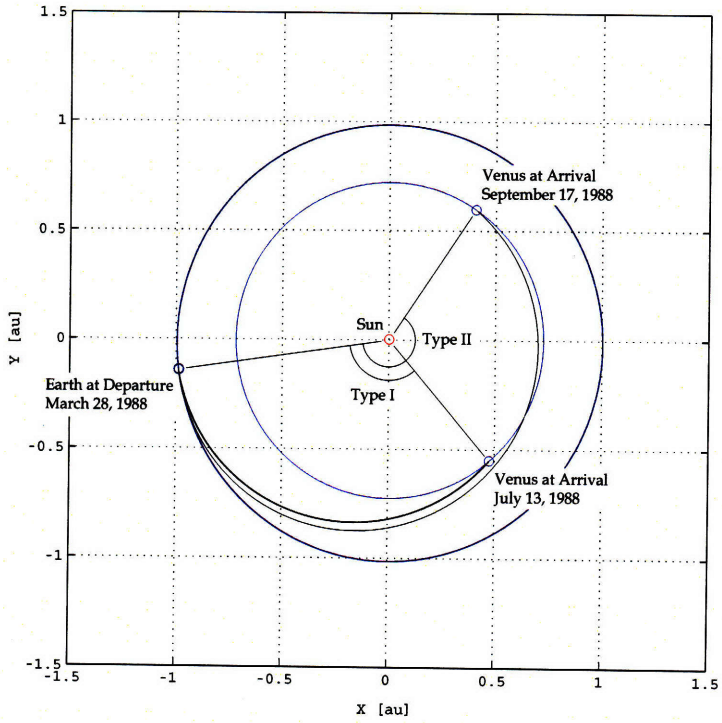


Figure 3-7: Type I and II trajectories: 1988 Earth-Venus

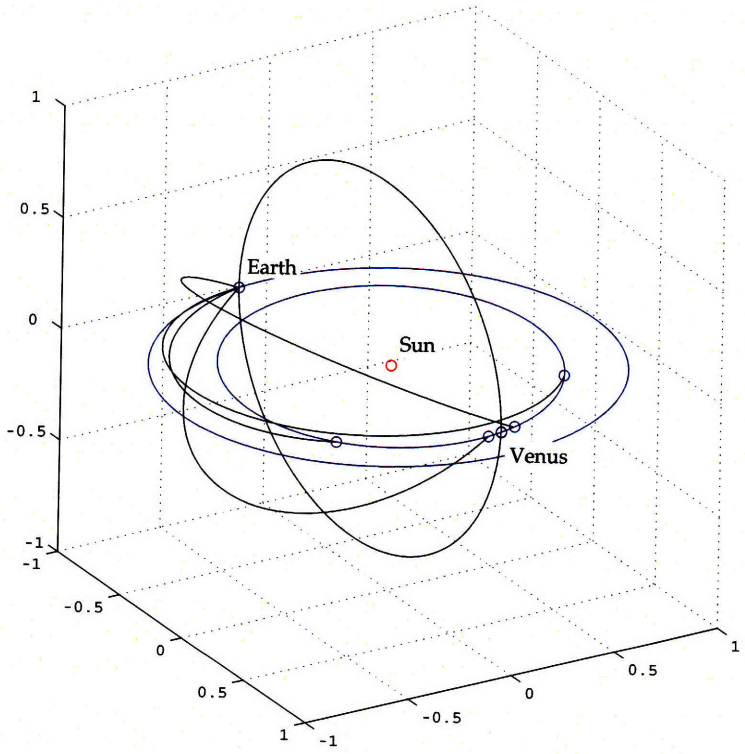


Figure 3-8: 3D view of trajectories: 1988 Earth-Venus

Table 3.2: Mars trajectory data 2002-2020

				JPL		Present	
#	Trajectory	Departure	Arrival	$C3_d$	$V_{\infty a}$	$C3_d$	$V_{\infty a}$
1	2002 Earth-Venus-Mars	8/06/02	6/09/03	12.3	7.2	12.3	7.2
2	2003 Mars-Earth	2/26/03	11/12/03	9.6	3.3	9.6	3.2
3	2003 Mars-Earth	4/18/03	11/10/03	7.4	3.0	7.4	3.0
4	2003 Earth-Mars	5/09/03	12/29/03	12.7	2.8	12.7	2.9
5	2003 Earth-Mars	6/07/03	12/26/03	8.8	2.7	8.8	2.7
6	2004 Earth-Venus-Mars	5/31/04	4/03/05	20.1	8.4	20.1	8.3
7	2005 Mars-Earth	6/28/05	1/06/06	13.6	3.7	13.6	3.7
8	2005 Mars-Earth	7/08/05	3/31/06	13.2	3.8	13.2	3.8
9	2005 Earth-Mars	8/10/05	2/22/06	15.9	3.2	15.9	3.2
10	2005 Earth-Mars	9/02/05	10/11/06	15.4	3.5	15.4	3.5
11	2007 Mars-Earth	7/21/07	4/29/08	10.2	2.9	10.2	2.9
12	2007 Mars-Earth	7/31/07	2/29/08	14.2	4.4	14.2	4.4
13	2007 Earth-Mars	9/22/07	9/26/08	12.7	2.8	12.7	2.8
14	2007 Earth-Mars	9/23/07	4/19/08	18.8	3.9	18.8	3.9
15	2009 Mars-Earth	7/28/09	5/16/10	7.8	2.9	7.8	2.9
16	2009 Mars-Earth	8/22/09	5/10/10	9.4	3.2	9.4	3.2
17	2009 Earth-Mars	10/14/09	9/07/10	10.3	2.5	10.3	2.5
18	2009 Earth-Mars	10/25/09	6/09/10	16.1	4.1	16.1	4.1
19	2011 Mars-Earth	8/12/11	7/10/12	6.8	3.5	6.8	3.5
20	2011 Earth-Mars	11/08/11	8/31/12	8.9	2.8	8.9	2.8
21	2011 Earth-Mars	11/15/11	7/25/12	9.0	3.7	9.0	3.7
22	2013 Mars-Earth	9/28/13	6/15/14	6.2	5.7	6.2	5.7
23	2013 Mars-Earth	9/28/13	8/29/14	5.8	5.0	5.8	5.0
24	2013 Earth-Venus-Mars	10/24/13	2/17/15	14.6	12.0	14.6	11.7
25	2013 Earth-Mars	12/27/13	7/23/14	9.0	5.3	9.0	5.3
26	2013 Earth-Mars	1/01/14	11/25/14	8.8	4.4	8.8	4.4
27	2015 Earth-Venus-Mars	5/25/15	4/17/16	15.8	7.3	15.6	7.3
28	2015 Earth-Venus-Mars	5/29/15	12/25/16	18.9	12.0	18.9	11.8
29	2015 Mars-Earth	11/30/15	7/24/16	5.6	5.2	5.6	5.2
30	2015 Mars-Earth	12/13/15	10/08/16	5.4	5.5	5.4	5.5
31	2016 Earth-Mars	2/20/16	8/19/16	8.9	5.3	8.9	5.3
32	2016 Earth-Mars	3/21/16	1/20/17	8.0	5.4	8.0	5.4
33	2017 Earth-Venus-Mars	3/21/17	1/17/18	18.8	11.0	18.8	10.9
34	2018 Mars-Earth	3/14/18	10/12/18	6.1	3.2	6.1	3.2
35	2018 Mars-Earth	3/14/18	11/12/18	6.3	3.9	6.4	3.8
36	2018 Earth-Mars	5/07/18	1/14/19	8.4	3.5	8.4	3.5
37	2018 Earth-Mars	5/17/18	1/07/19	7.7	3.3	7.7	3.3
38	2020 Earth-Venus-Mars	3/11/20	6/21/21	13.2	7.6	13.1	7.6
39	2020 Mars-Earth	6/06/20	12/14/20	11.4	3.3	11.4	3.3
40	2020 Mars-Earth	6/26/20	3/11/21	14.1	4.2	14.1	4.2
41	2020 Earth-Mars	7/18/20	1/27/21	13.2	2.9	13.2	2.9
42	2020 Earth-Mars	8/24/20	10/09/21	16.5	3.8	16.5	3.8

Chapter 4

Simulation Results and Discussions for Mars Missions

In Chapter 3, we developed trajectory programs that can calculate direct and flyby trajectories. By a full-factorial investigation using these programs, we create the contours of $C3_d$, $C3_a$, and as described later, ΔV_{tot} for Mars missions in the time frame 2020 to 2040, during which such missions seem most relevant. By visualizing the regions of some reasonable level of ΔV_{tot} on the contours, they will serve as a "launch window calendar." This chapter presents how to create the contours, followed by the discussions of the results.

4.1 Criteria for Mission Feasibility

As the criteria for mission feasibility, we define four constraints as follows:

$C3_d$: $C3_d$ determines the launch feasibility. In the JPL study [22], a feasible launch assumed that $C3_d$ is less than 25 km^2/s^2. Considering envisioned advances in technology, we used a $C3_d$ of 30 km^2/s^2 as the criteria for launch feasibility.

$C3_a$: For orbit insertion at arrival, propulsive capture generally requires a minimum arrival velocity, while aerocapture tolerates higher arrival velocities. In the JPL's study [22], the authors stated that a tolerable V_∞ of the future thermal

protection systems (TPS) would be up to 8 km/s. From this value of the tolerance, we determined that 60 km^2/s^2 is an upper bound for $C3_a$ ($= V_\infty^2$).

h_m: For flyby missions, considering the Venusian atmosphere, the flyby trajectory must pass well above the surface. We assumed that a minimum passing altitude h_m of a feasible flyby must be 100 km above the surface.

ΔV_{PFM}: Allowing a large amount of $|\Delta V_{\mathrm{PFM}}|$ would not make sense because we originally wanted to save fuel by taking advantage of the Venus gravitational field. In this study, we selected 0.3 km/s as a reasonable upper bound for $|\Delta V_{\mathrm{PFM}}|$.

These four constraints are expressed in the form of inequality:

$$C3_d \leq 30 \ [km^2/s^2] \tag{4.1a}$$

$$C3_a \leq 60 \ [km^2/s^2] \tag{4.1b}$$

$$h_m \geq 100 \ [km] \tag{4.1c}$$

$$|\Delta V_{\mathrm{PFM}}| \leq 0.3 \ [km/s] \tag{4.1d}$$

As described later, these constraints are used for data filtering in the contours and data skipping in the computation.

4.2 Full-Factorial Computation

Since we now have MATLAB programs that return the trajectory data for a given combination of dates of events, by wrapping them in a repetition statement (`"for"` statement), we obtain a sequence of trajectory data for a specific range of departure and arrival dates. Thus we can draw the contours of C3 on the t_1-t_3 plane, where for commonality between direct and flyby cases, we define the Earth departure date as t_1, Venus encounter date as t_2 if the mission uses flyby, and Mars arrival date as t_3. If we want to create the contours for a much longer time frame, however, we should take as y-axis the time of flight instead of the arrival date for easy handling of the chart; if we took the arrival date as y-axis, the contours would be rising diagonally

from bottom left to top right. Thus, as shown later, we draw the contours on t_1-TOF plane, where TOF $= t_3 - t_1$.

To avoid an enormous computation cost at a time and keep the trajectory data in multiple compact files, we divided the 7500-day period of 2020-2040 into six 1250-day periods. Therefore, each period corresponds to one of the following periods:

$$\begin{array}{ll} \text{Period 1:} & \text{Jan 1, 2020 – Jun 4, 2023} \\ \text{Period 2:} & \text{Jun 4, 2023 – Nov 5, 2026} \\ \text{Period 3:} & \text{Nov 5, 2026 – Apr 8, 2030} \\ \text{Period 4:} & \text{Apr 8, 2030 – Sep 9, 2033} \\ \text{Period 5:} & \text{Sep 9, 2033 – Feb 10, 2037} \\ \text{Period 6:} & \text{Feb 10, 2037 – Jul 14, 2040} \end{array} \tag{4.2}$$

As shown later, each contour plot is produced according to this division of period.

For each period, we first conducted full-factorial computations and created contour plots for direct and flyby trajectories separately. Then we put them together into a single chart to obtain an integrated contour plot. The following two sections go into more details for direct and flyby cases, respectively.

4.2.1 $C3_d$ Contours for Direct Trajectory

For direct trajectories, the procedure is straightforward: it requires just doubly wrapping Algorithm 3.7 in a couple of `"for"` statements for departure and arrival dates. Taking as an example Earth-Mars direct trajectories in Period 1, Figure 4-1 shows the contours of $C3_d$ with departure date as x-axis and TOF as y-axis. The range is 1250 days in the x direction and 700 days in the y direction with a step size of 2 days in both x and y directions. Therefore, the grid consists of 626 nodes in the x direction and 351 nodes in the y direction. Since y-axis is TOF instead of arrival date, each 45° diagonal line represents a specific arrival date. Therefore, when we look at some point in the figure, we can find the arrival date by looking down along the arrival-date line as well as the departure date from right below.

In Figure 4-1, we can observe two launch windows in late 2020 and late 2022, each

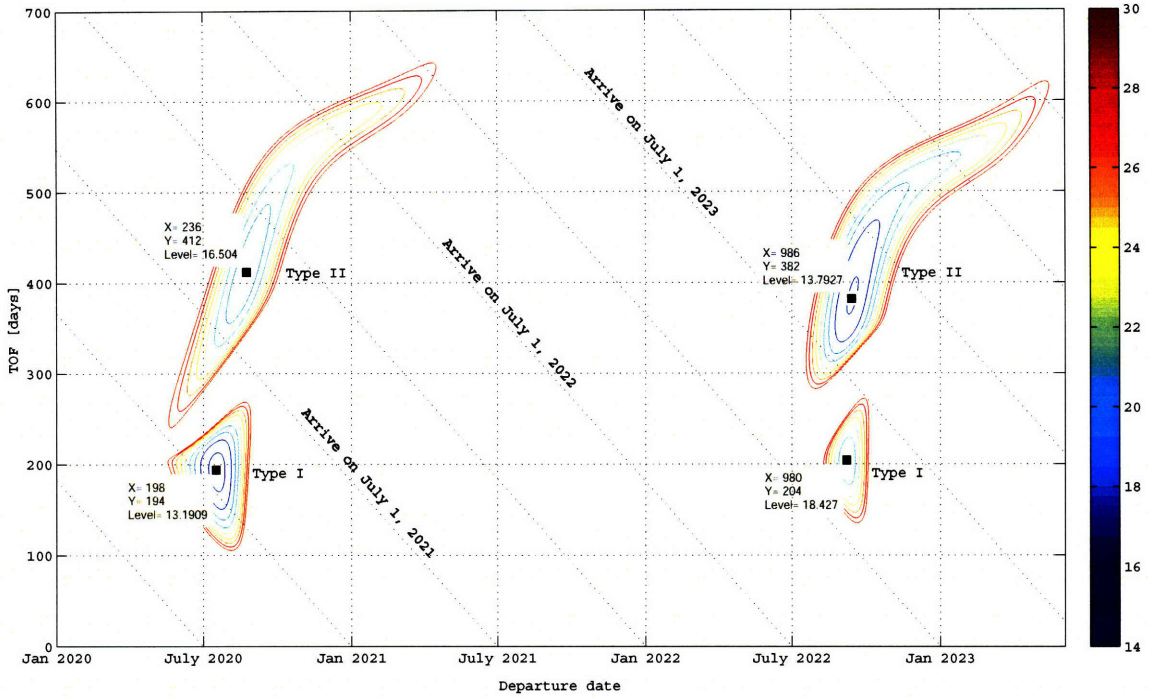


Figure 4-1: $C3_d$ contours: Earth-Mars direct Period 1

of which consists of two "craters" of local minimum $C3_d$. According to the constraint for $C3_d$ in Eq. (4.1a), the figure displays $C3_d$ up to 30, which is represented by the outmost edges of the craters. As stated earlier in Section 3.3.2, the lower and upper craters in each window correspond to type I and type II trajectories, respectively.

The required computation time to produce this contour plot was about 12 hours (Intel®Core™2 Duo processor at 2.40 GHz).

4.2.2 $C3_d$ Contours for Flyby Trajectory

For flyby trajectories, the procedure is a bit tricky. In Section 3.2.3, we developed Algorithm 3.8 for flyby trajectories. If we simply wrap this algorithm in triple "`for`" statements for departure, encounter, and arrival dates, however, it will take an immeasurable time, which is estimated to be more than 6000 hours, compared to 12 hours for direct trajectories. But, indeed, this procedure repeats the same calculations many times; after we solve a flyby trajectory for specific t_1, t_2, and t_3, for example, if we later solve for t_1, t_2, and $t_3 + 1$, then we solve the same Lambert's problem for

the combination of t_1 and t_2. This duplicated computation occurs hundreds of times, which wastes a lot of time.

Instead of such a straightforward approach, we should note the fact that the problem can be decomposed into two direct-trajectory problems, that is, Earth-Venus and Venus-Mars trajectories. These two trajectories are independently determined. Therefore, we can in advance conduct full-factorial computations for Earth-Venus and Venus-Mars direct-trajectory problems separately, and store the data in memory cache. Since for a flyby calculation, we need $\mathbf{v}_{e12}$, which is a spacecraft's velocity vector at Venus arrival from an Earth-Venus trajectory, and $\mathbf{v}_{e23}$, which is a spacecraft's velocity vector at Venus departure on a Venus-Mars trajectory, then we can use the information of these velocity vectors from the memory cache obtained from the two decomposed problems that have been computed in advance.

By scanning t_2 for given t_1 and t_3, we picked an optimal t_2 that would minimize $C3_d$ for the combination of t_1 and t_3 while satisfying all the constraints in Eqs. (4.1). To save more time in this process, since we do not want to calculate infeasible points with large $C3_d$ and $C3_a$, we screened them out by the constraints in Eqs. (4.1a) and (4.1b), and skipped a flyby calculation for such infeasible points.

This sequence of process greatly contributed to reducing the computation time; the total computation time was about 24 hours, which is just twice as much as the direct case despite the fact that the problem was added another dimension by t_2.

Taking as an example Earth-Venus-Mars flyby trajectories in Period 1, Figure 4-2 shows the contours of $C3_d$ with departure date as x-axis and TOF as y-axis. As explained above, each point in the figure is internally optimized with respect to t_2. The figure displays only the feasible regions that satisfy Eqs. (4.1).

In Figure 4-2, we can observe two launch windows in early 2020 and late 2021. Unlike the direct trajectory, these launch windows are smaller and more complex in shape. In addition, both edges and surfaces of the contours appear to be nonsmooth. This is due to the constraints for h_m and ΔV_{PFM} in Eqs. (4.1c) and (4.1d).

The mission windows are also restricted in terms of t_2. To examine the structure of t_2 space, we fixed t_1 and t_3 at certain dates and scanned t_2. Taking as an example

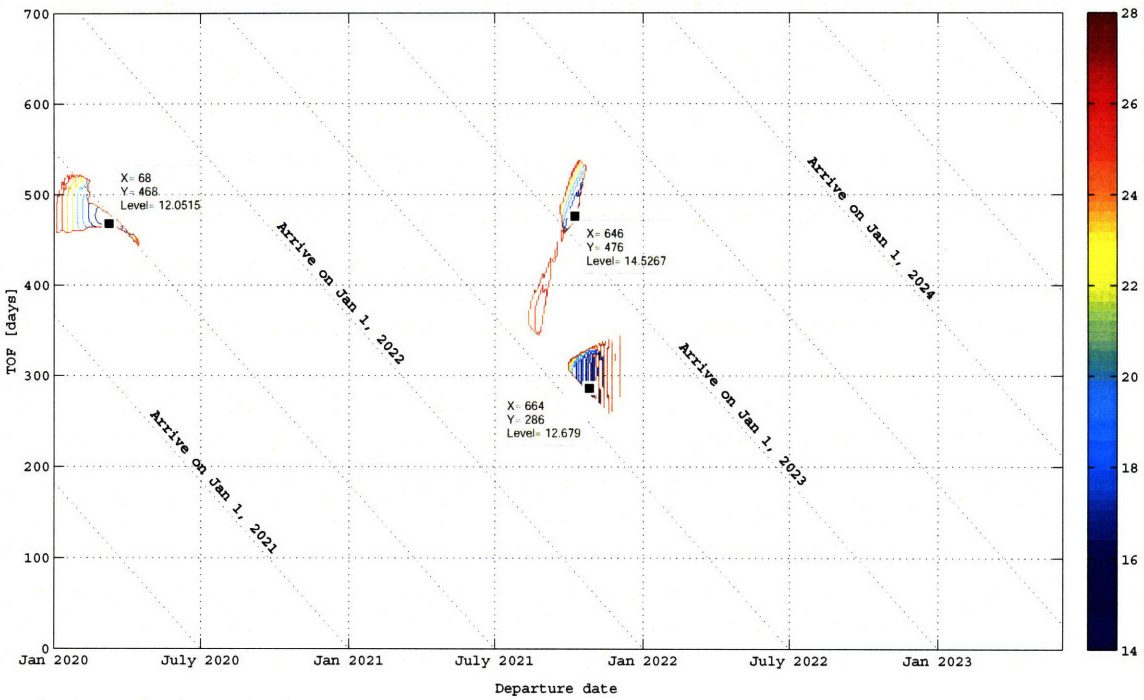


Figure 4-2: $C3_d$ contours: Earth-Venus-Mars flyby Period 1

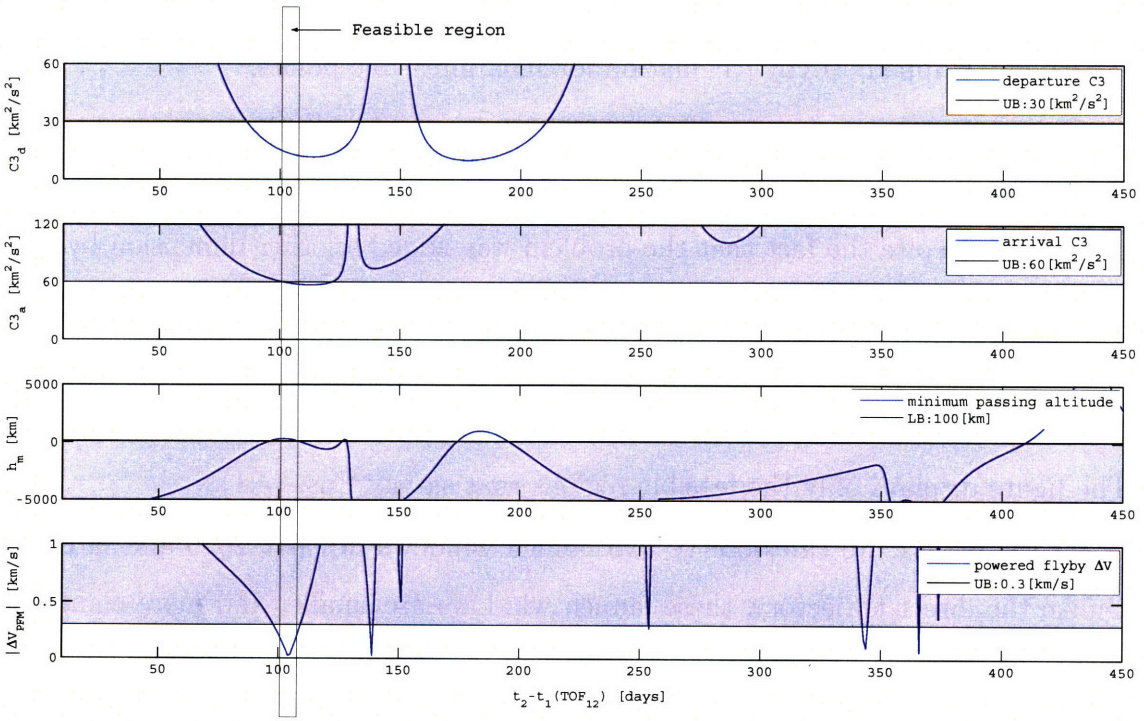


Figure 4-3: $C3_d$, $C3_a$, h_m, and $|\Delta V_{\text{PFM}}|$ with respect to $t_2 - t_1$

the upper left window in Figure 4-2, which corresponds to No. 38 in Table 3.2, the Earth departure date t_1 was fixed at March 11, 2020, and the Mars arrival date t_3 was fixed at June 21, 2021. Figure 4-3 shows $C3_d$, $C3_a$, h_m, and $|\Delta V_{\mathrm{PFM}}|$ with respect to $t_2 - t_1$, which represents the number of days from Earth departure to Venus encounter. Infeasible regions by the constraints are grayed out. As we can see from this figure, these quantities are highly non-convex in the t_2 direction. Since the constraints filter these highly non-convex quantities, it makes sense that the contours in Figure 4-2 have a complicated structure. We should also note that feasible windows are considerably small. Therefore, the grid with a step size of 2 days might be somewhat coarse.

4.2.3 Integrated ΔV_{tot} Contours

Now that we have obtained the trajectory data for both direct and flyby cases by full-factorial computations, we will put them together into a single chart to create a complete "launch window calendar." Since we have two contour plots for direct and flyby on the same range in the t_1-t_3 plane, selecting a superior one of the two trajectories at each point on the grid gives a direct/flyby integrated contour plot.

If we use $C3_d$ for integration, however, it might be unfair since a flyby trajectory also requires ΔV_{PFM} for powered flyby maneuver. Instead, by converting $C3_d$ into ΔV_d, which is the ΔV required for departure, and we can uniformly treat the ΔV for departure and powered flyby. Thus we define

$$\Delta V_{\mathrm{tot}} = |\Delta V_d| + |\Delta V_{\mathrm{PFM}}| \tag{4.3}$$

and use ΔV_{tot} instead of $C3_d$ for integration. In this modification, by establishing an upper bound for ΔV_{tot}, we do not have to consider the constraint for ΔV_{PFM} in Eq. (4.1d). As discussed in the previous section, the ΔV_{PFM} constraint is one of the major factors that caused the highly non-convex nature of the problem. Therefore, this modification might be able to remove the nonsmoothness to some extent.

In conversion from $C3_d$ to ΔV_d, we assumed a departure hyperbola starting from a circular parking orbit with an altitude of 300 km. The departure hyperbola has a

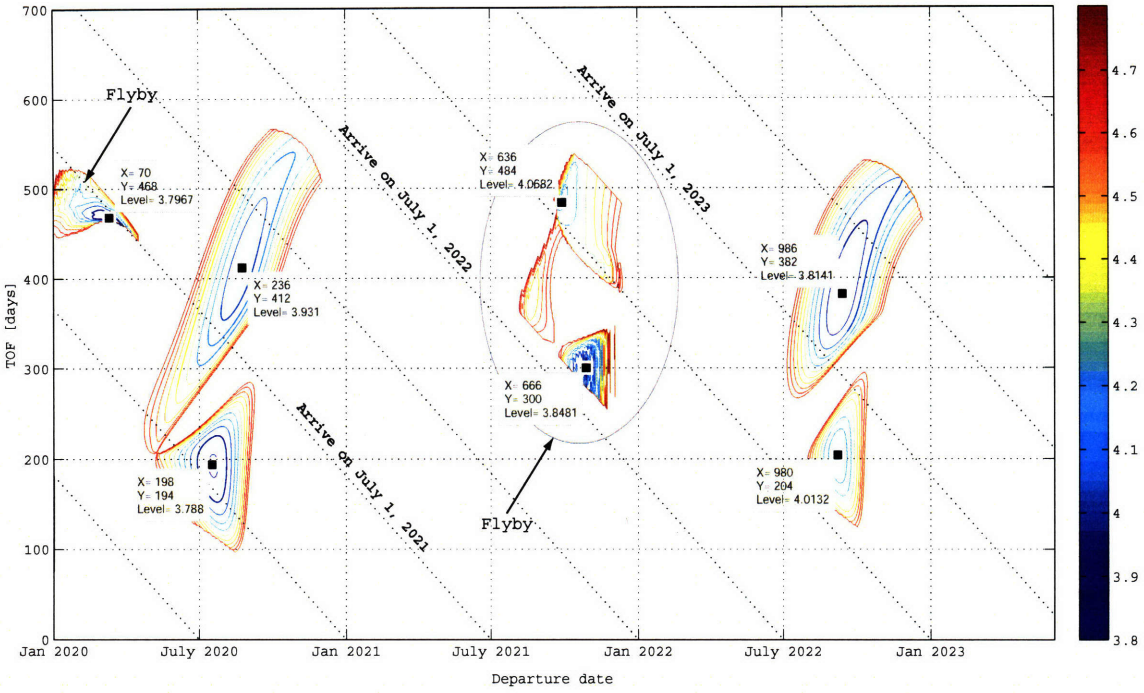


Figure 4-4: ΔV_{tot} contours: Earth-Mars direct/Earth-Venus-Mars flyby Period 1

periapsis radius equal to the radius of the parking orbit. If r_{po} is the radius of the parking orbit, the velocity required at the injection point is

$$V_d = \sqrt{\text{C3}_d + \frac{2\mu_P}{r_{\text{po}}}} \tag{4.4}$$

Therefore, the ΔV required for departure is

$$\Delta V_d = V_d - V_{\text{po}} = \sqrt{\text{C3}_d + \frac{2\mu_P}{r_{\text{po}}}} - \sqrt{\frac{\mu_P}{r_{\text{po}}}} \tag{4.5}$$

where V_{po} is a circular speed at radius r_{po}.

Figure 4-4 shows the direct/flyby integrated ΔV_{tot} contours. Assuming a 300-km parking orbit around the Earth, $\text{C3}_d = 30$ [km^2/s^2] approximately corresponds to $\Delta V_d = 4.5$ [km/s]. Therefore, the figure displays ΔV_{tot} up to 4.8 km/s. Note that the craters for direct trajectory are filtered by the C3_a constraint in Eq. (4.1b) while those in Figure 4-1 are not.

4.3 Earth-Mars Trajectory

This section presents the results for 2020-2040 Earth-Mars trajectories, which include Earth-Mars direct trajectories and Earth-Venus-Mars flyby trajectories. First, we analyze the Earth-Mars direct trajectories using $C3_d$ and $C3_a$ contours to gain a general understanding of the nature of the trajectories. Next, from the perspective of launch window, we use integrated ΔV_{tot} contours to discuss how competitive the flyby trajectories are with the direct.

4.3.1 Earth-Mars Direct Trajectory

Figures 4-5 through 4-16 show the $C3_d$ and $C3_a$ contours for Period 1-6 in (4.2), respectively. These contours display $C3_d$ up to 30 km^2/s^2 and $C3_a$ up to 60 km^2/s^2. Note that the contours are not filtered by each other's constraints. Therefore, the craters in the figures do not mean exactly feasible regions. The data tips display the values of each local minimum.

We can observe periodic patterns, which are obviously produced by the planets' revolution. As stated earlier in Chapter 1, the same relative angular positions of Earth and Mars repeat after a synodic period of approximately 780 days. The differences between x-coordinates of the data tips vary between 750 to 810, and the average is about 780. As explained in Appendix B in more detail, the contour patterns vary due to the effect of inclination and eccentricity. These two factors make the relative configuration different even when the relative angular positions are the same. In other words, the relative configuration almost recurs when multiples of the synodic period are close to an integer number of years. Seven and eight cycles of the synodic period are nearly 15 and 17 years, respectively. Thus, similar opportunities should repeat every 7 to 8 launch windows. In the sequence of contours, ten launch windows in total appear during 2020 through 2040. It is true that the second window in 2022 appears to be similar to the ninth in 2037 and the tenth in 2039 in terms of shape and value of local minimum.

As mentioned in Section 4.2.1, each window consists of two craters; the lower

and upper correspond to type I and type II trajectories, respectively. As we can see from the $C3_d$ contours, the time of flight for type I trajectory is around 200 days, and the time of flight for type II ranges from 200 days to 500 days and beyond. We might say that type I trajectory is superior since a short flight time would be better. However, we should also take into account the time of stay on Mars and return flight; we have to consider the duration of the entire mission. Therefore, we must select a good combination of Earth-Mars opportunity and Mars-Earth opportunity.

As for the values of local minimum $C3_d$, the fourth to seventh launch windows (2026-2033) have relatively small $C3_d$ less than 10. As for the values of local minimum $C3_a$, on the other hand, the launch windows with small $C3_d$ tend to have relatively large $C3_a$, and vice versa. As shown in Figure 3-2, the Mars orbit is one-sided due to its eccentricity. Therefore, it sometimes requires a larger trans-Mars orbit, which is one of the reasons for the variation of local minimum $C3_d$. We should also note that the type II trajectory has a smaller value of local minimum $C3_d$ except for the first and eighth windows. There would be a trade-off between $C3_d$ and TOF. Therefore, we should select the type of trajectory on a mission-by-mission basis.

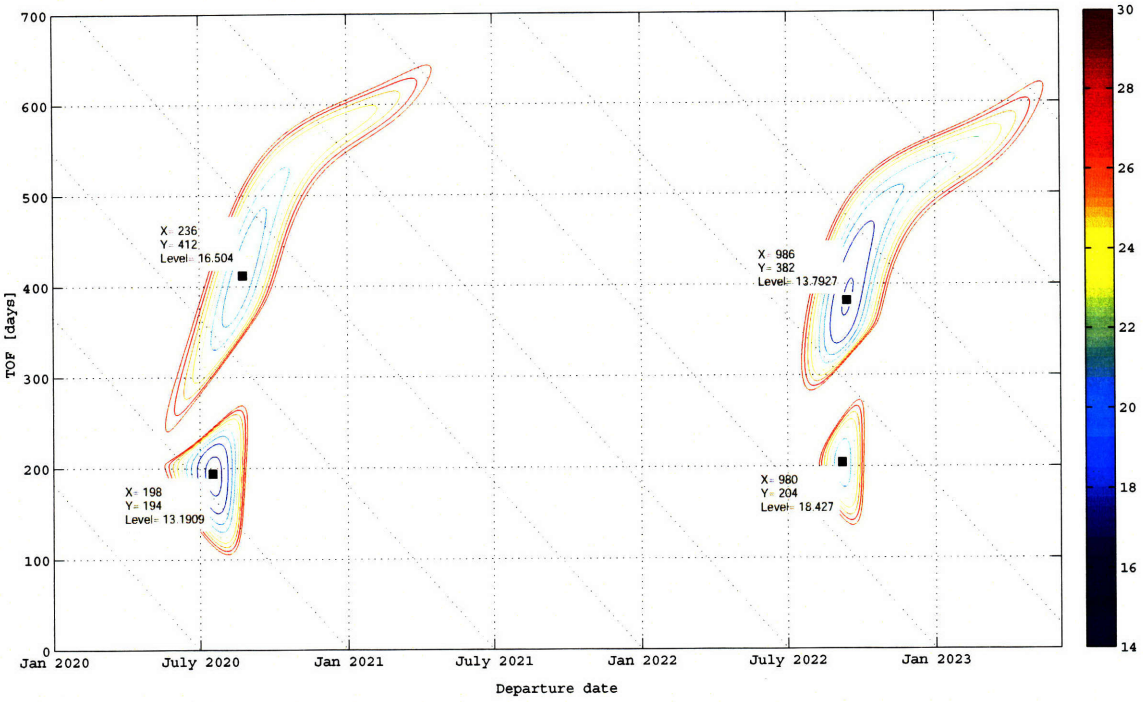


Figure 4-5: $C3_d$ contours: Earth-Mars direct Period 1

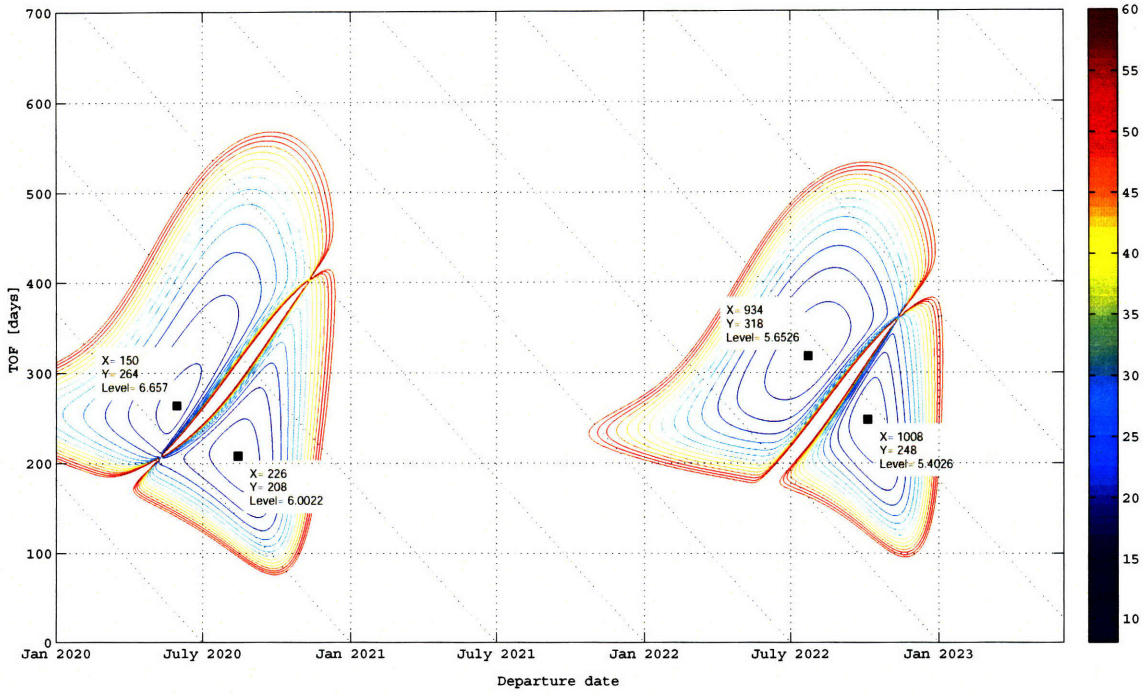


Figure 4-6: $C3_a$ contours: Earth-Mars direct Period 1

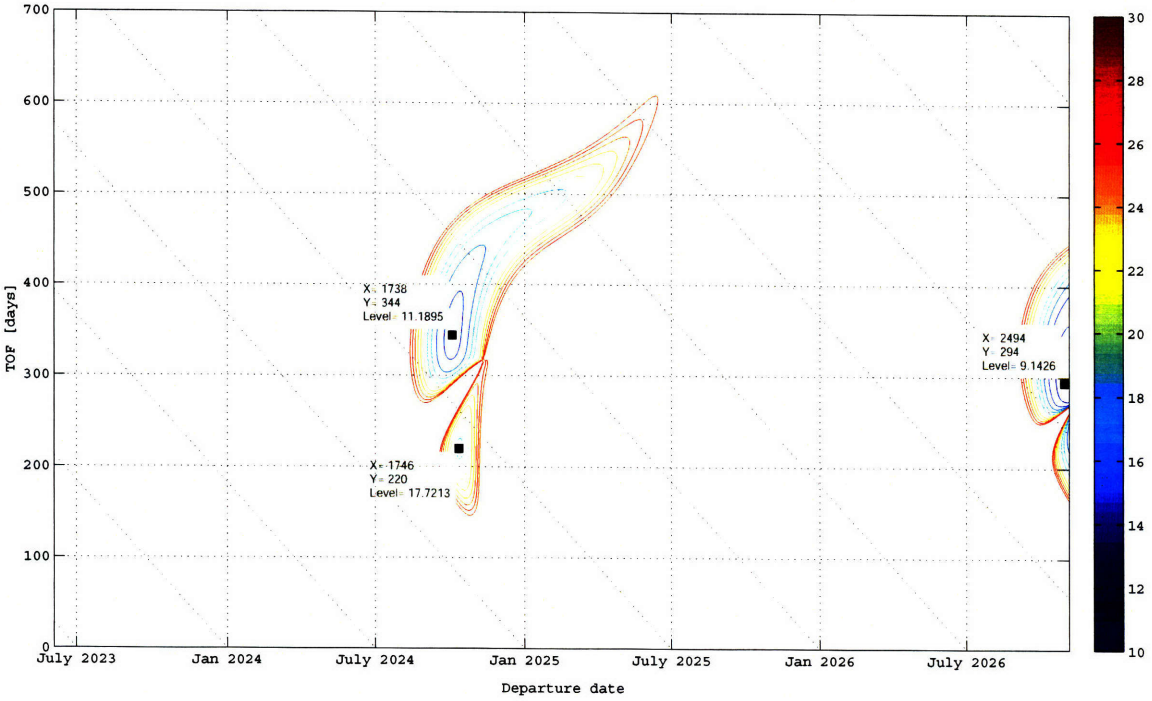


Figure 4-7: $C3_d$ contours: Earth-Mars direct Period 2

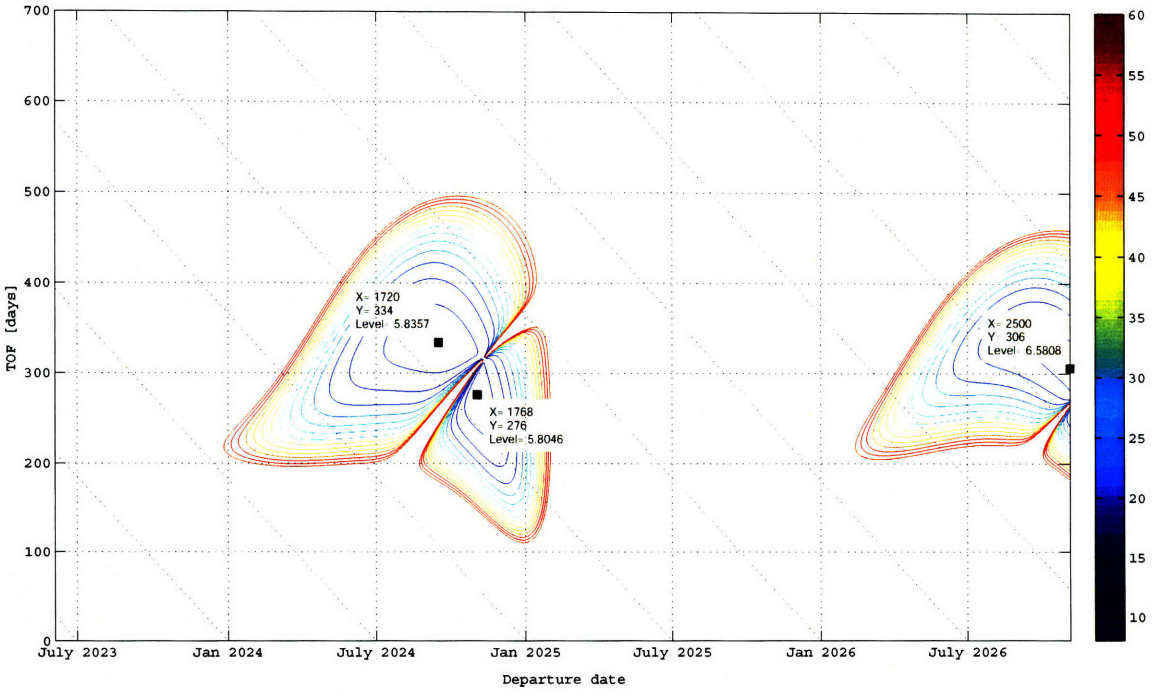


Figure 4-8: $C3_a$ contours: Earth-Mars direct Period 2

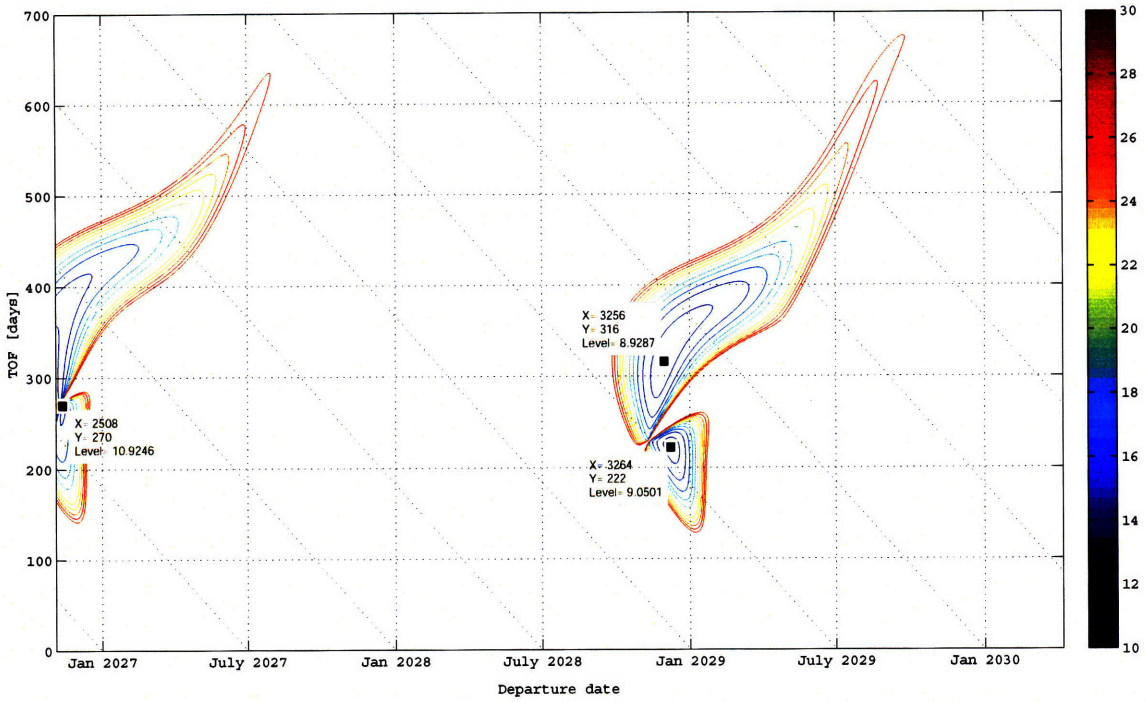


Figure 4-9: $C3_d$ contours: Earth-Mars direct Period 3

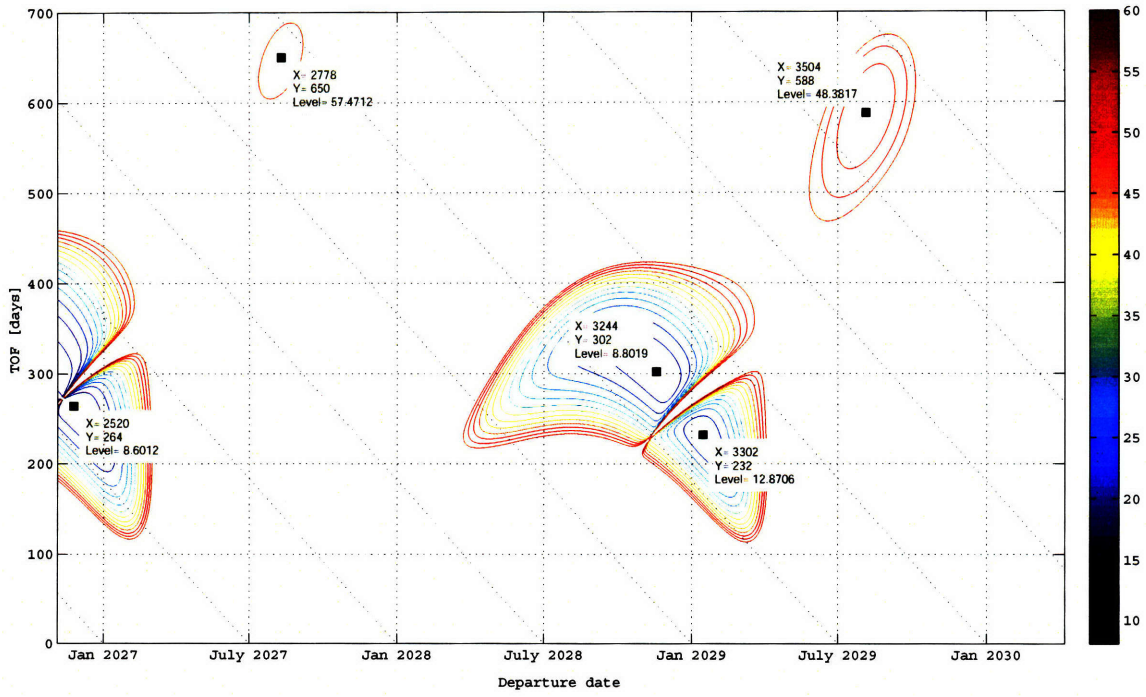


Figure 4-10: $C3_a$ contours: Earth-Mars direct Period 3

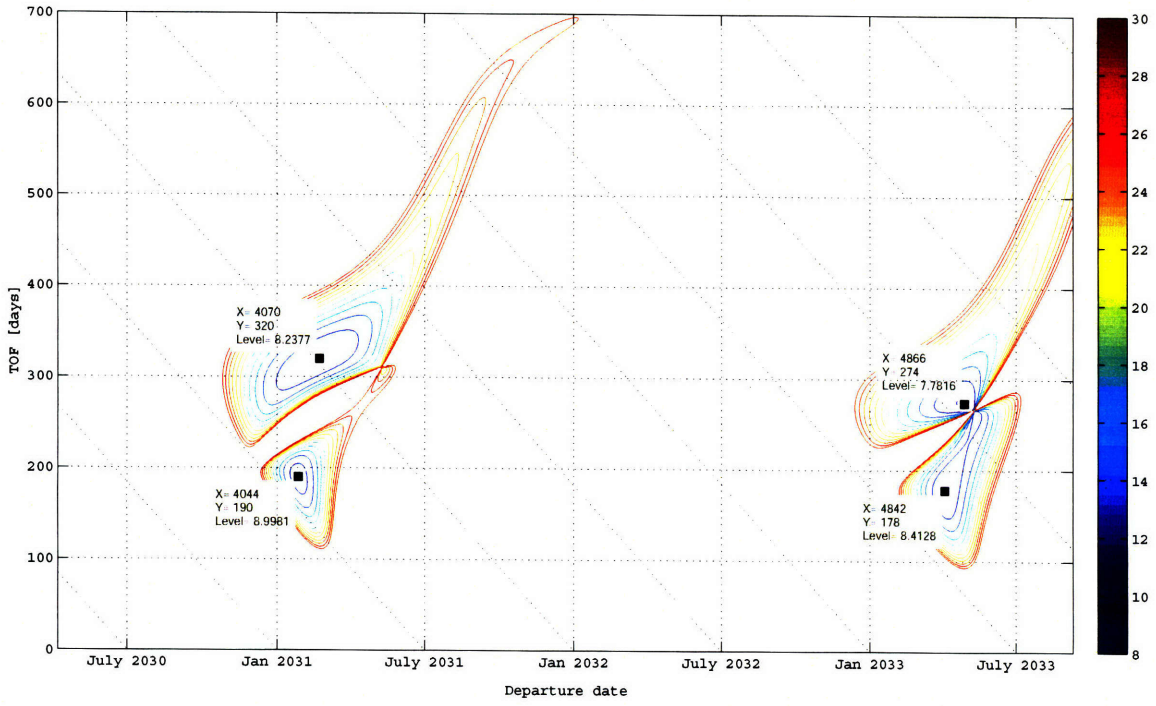


Figure 4-11: $C3_d$ contours: Earth-Mars direct Period 4

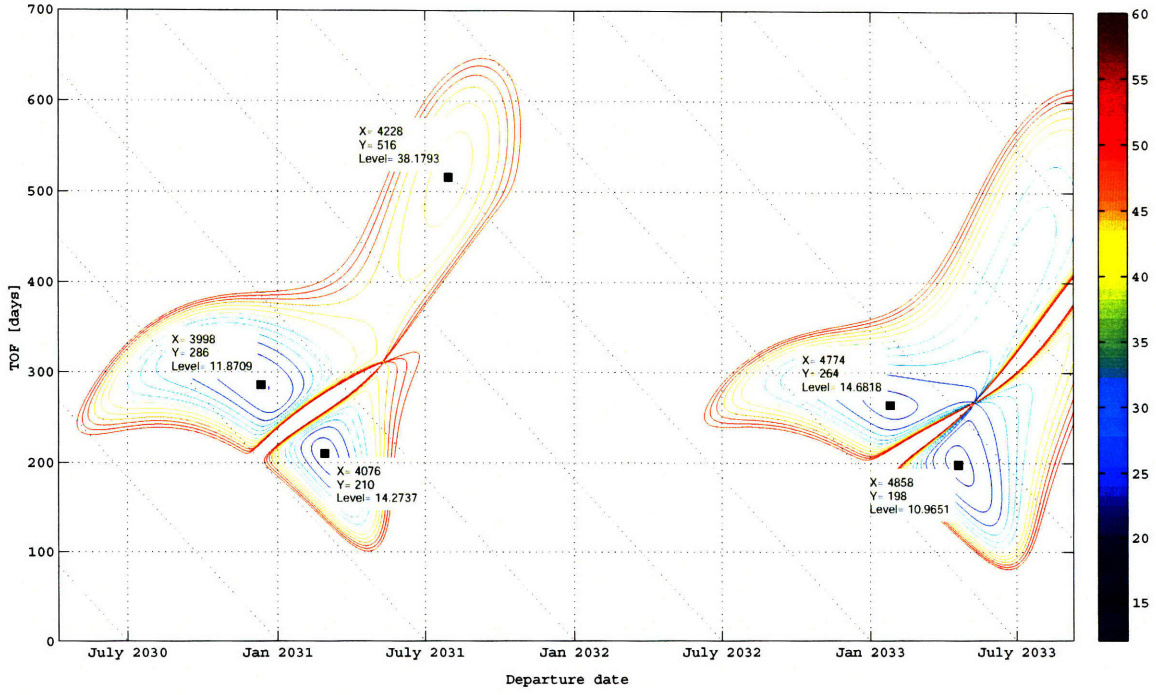


Figure 4-12: $C3_a$ contours: Earth-Mars direct Period 4

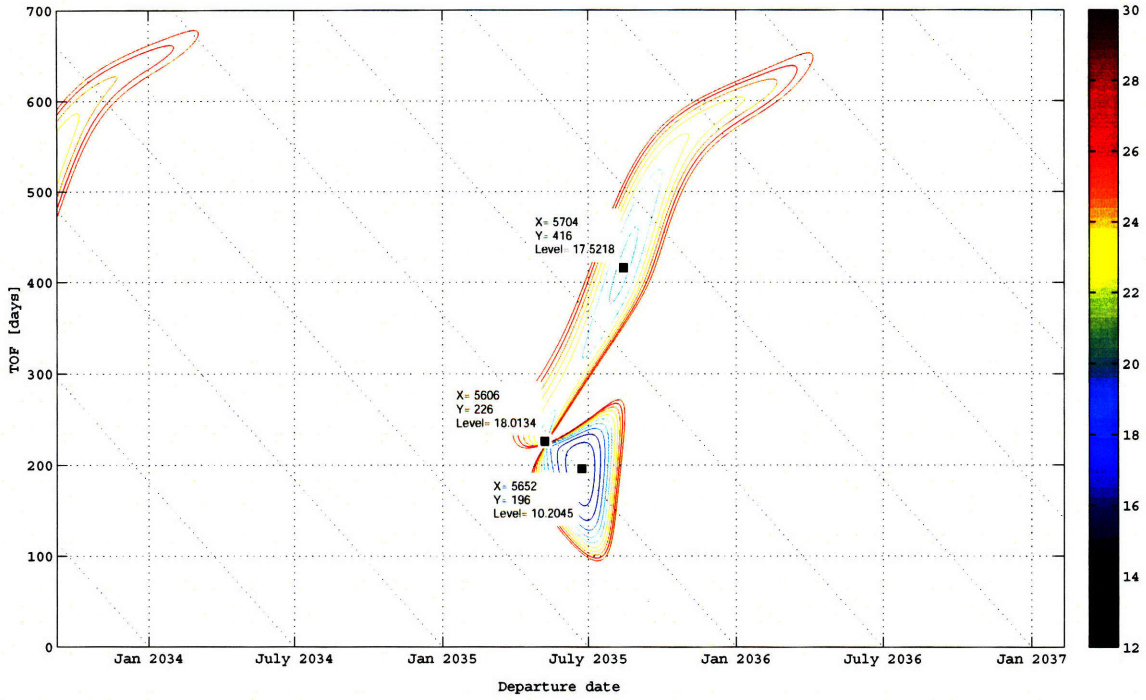


Figure 4-13: $C3_d$ contours: Earth-Mars direct Period 5

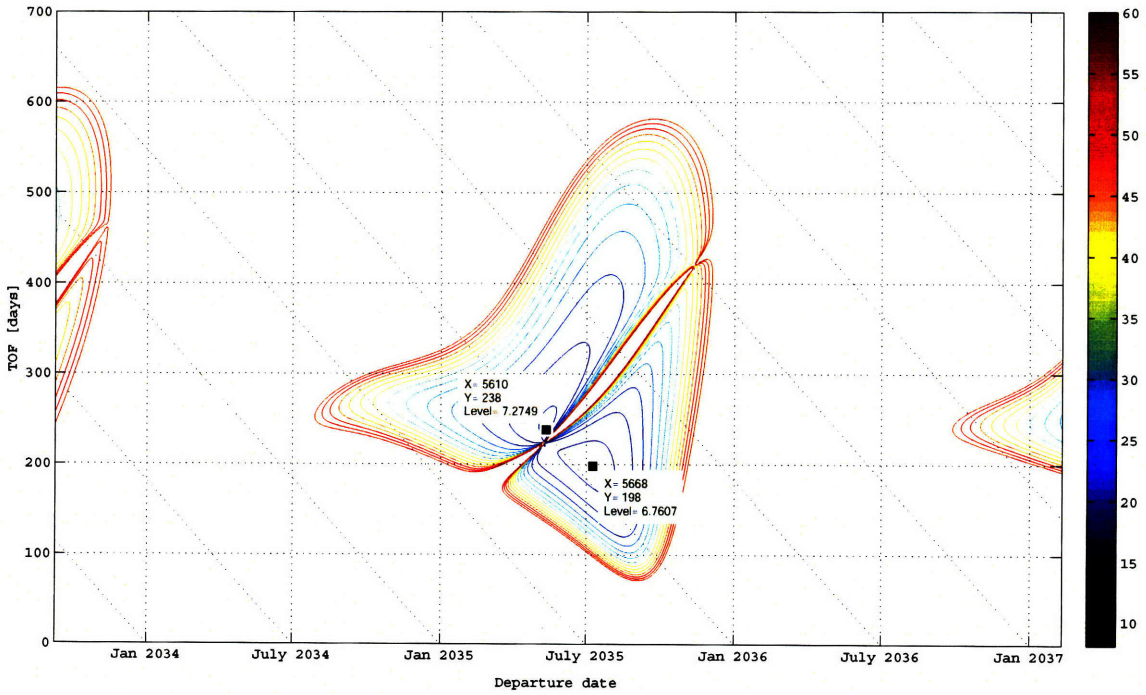


Figure 4-14: $C3_a$ contours: Earth-Mars direct Period 5

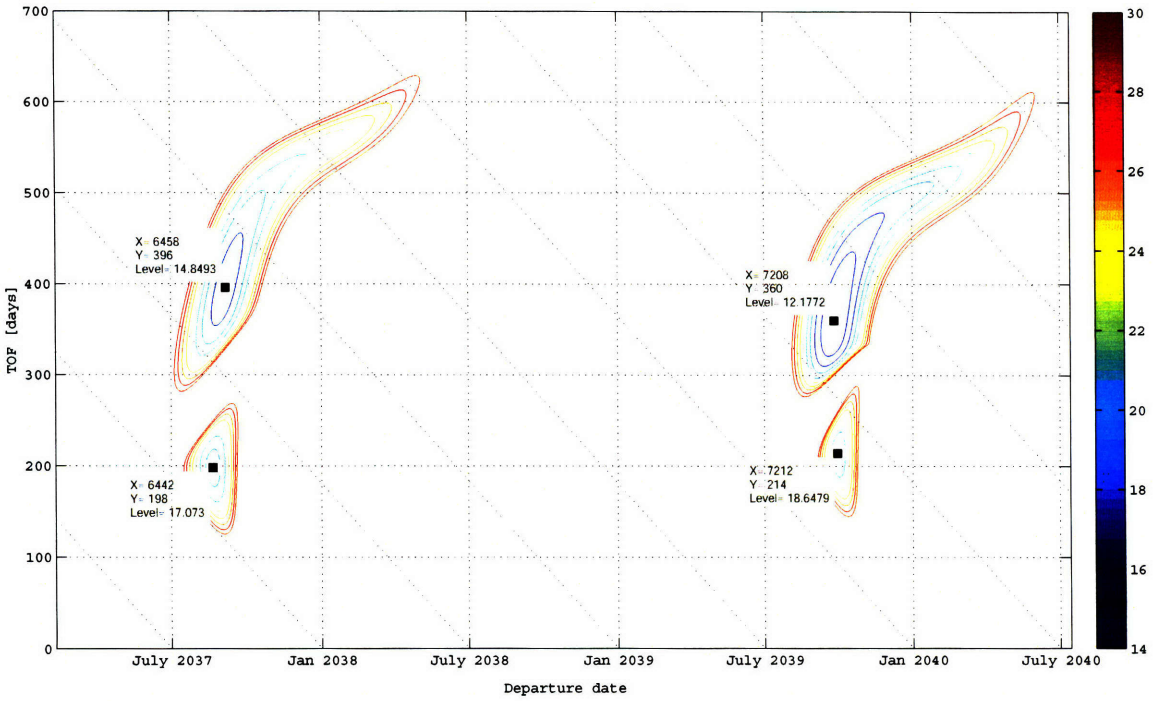


Figure 4-15: $C3_d$ contours: Earth-Mars direct Period 6

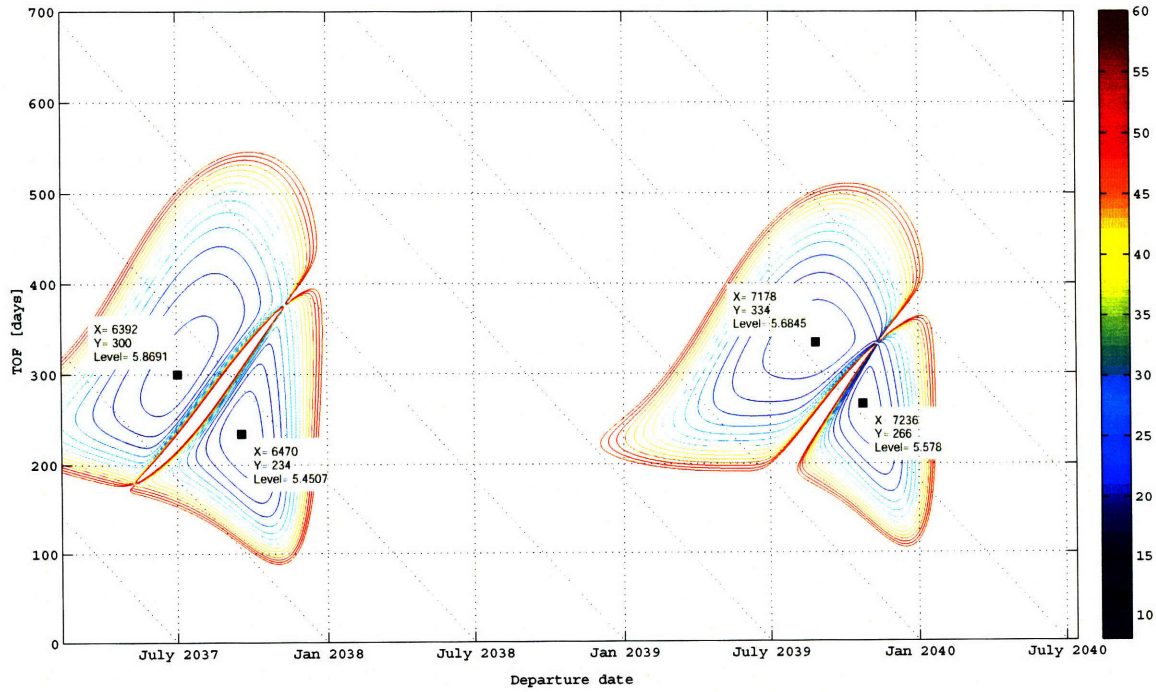


Figure 4-16: $C3_a$ contours: Earth-Mars direct Period 6

4.3.2 Earth-Mars Launch Window

Figures 4-17 through 4-22 show the integrated ΔV_{tot} contours for Period 1-6 in (4.2), respectively. These contours display ΔV_{tot} up to 4.8 km/s. Note that the contours are filtered by the constraints for $C3_d$, $C3_a$, and h_m. Therefore, the craters in the figures mean exactly feasible regions. The data tips display the values of each local minimum.

As stated in Chapter 1, the relative configuration among Earth, Mars, and Venus is known to repeat after a syzygistic period of approximately 6.4 years. In the figures, flyby windows can be divided into three groups of similar patterns: the first one is early 2020, early 2026, and early 2039, the second one is late 2021, early 2028, mid-2034, and the last one is late 2023, early 2030, and mid-2036. All the three groups have a period of about 6.4 years, which agree with Earth-Venus-Mars syzygistic period. Thus we have 3 flyby opportunities in 6.4 years, whether they are competitive or not. In total, we have 10 direct windows and 9 flyby windows during 2020 through 2040.

The local minimum ΔV_{tot} with the data tip is listed in Table 4.1. The average of local minimum ΔV_{tot} for direct trajectory is about 3.76 while the average for flyby trajectory is around 4.07. But some flyby opportunities have competitive ΔV_{tot} with the neighbor direct opportunities and some opportunities are dominated by the neighbors. As shown in Table 4.1, we determined the "competitiveness" of each opportunity by the following criteria.

- If two neighbor opportunities have almost the same departure date, and one has both earlier arrival date and lower ΔV_{tot} than the other, the other is regarded as "dominated" since a short mission duration is always better.

- If two neighbor opportunities have almost the same arrival date, and one has both later departure date and lower ΔV_{tot} than the other, the other is regarded as "dominated" since a short time of flight would be better.

- If an opportunity does not have neighbors, the opportunity is non-dominated and thus regarded as "competitive" since it adds a new launch window even if it has a relatively high ΔV_{tot}.

In Figure 4-22, for example, a flyby window in mid-2039 is dominated by a direct (type II) window in late 2039 since the direct window has a lower ΔV_{tot} and a later departure date while arriving on almost the same date.

As a result, seven out of eleven flyby opportunities are competitive. Therefore, in terms of launch windows, seven out of nine flyby windows are competitive. Earth-Venus-Mars flyby trajectories tend to have a relatively high ΔV_{tot} but give new opportunities. Having more launch windows available gives us a flexibility of mission planning.

Just for reference, Earth-Venus $C3_d$ contours and Venus-Mars $C3_a$ contours are presented in Appendix C.

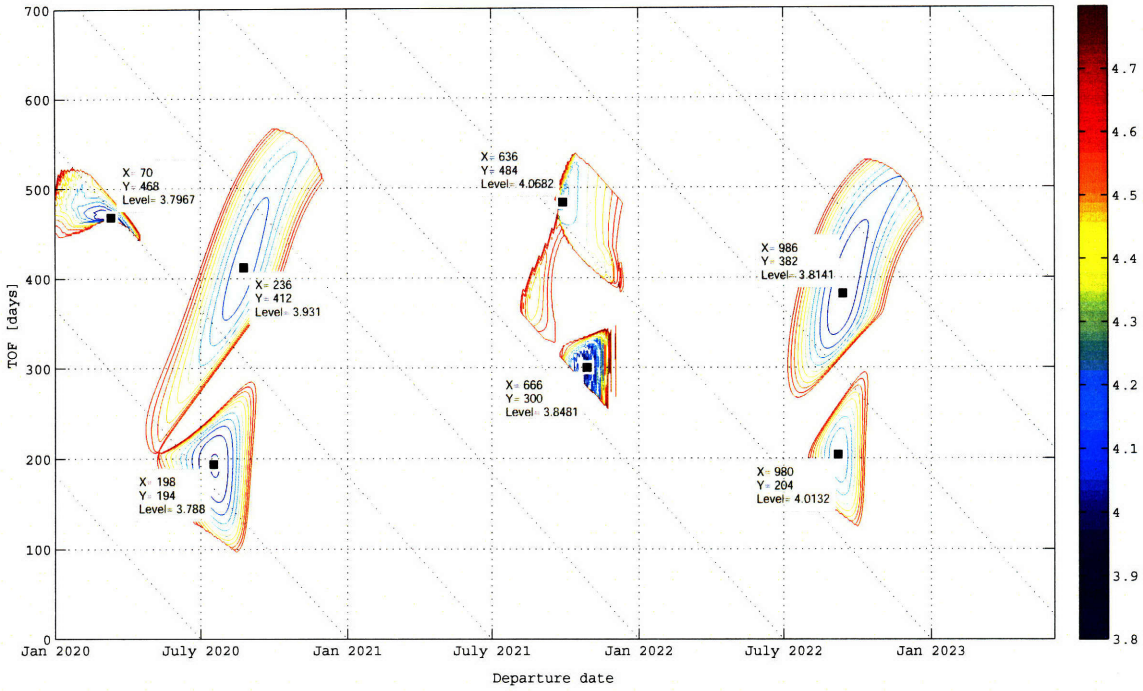


Figure 4-17: ΔV_{tot} contours: Earth-Mars direct/Earth-Venus-Mars flyby Period 1

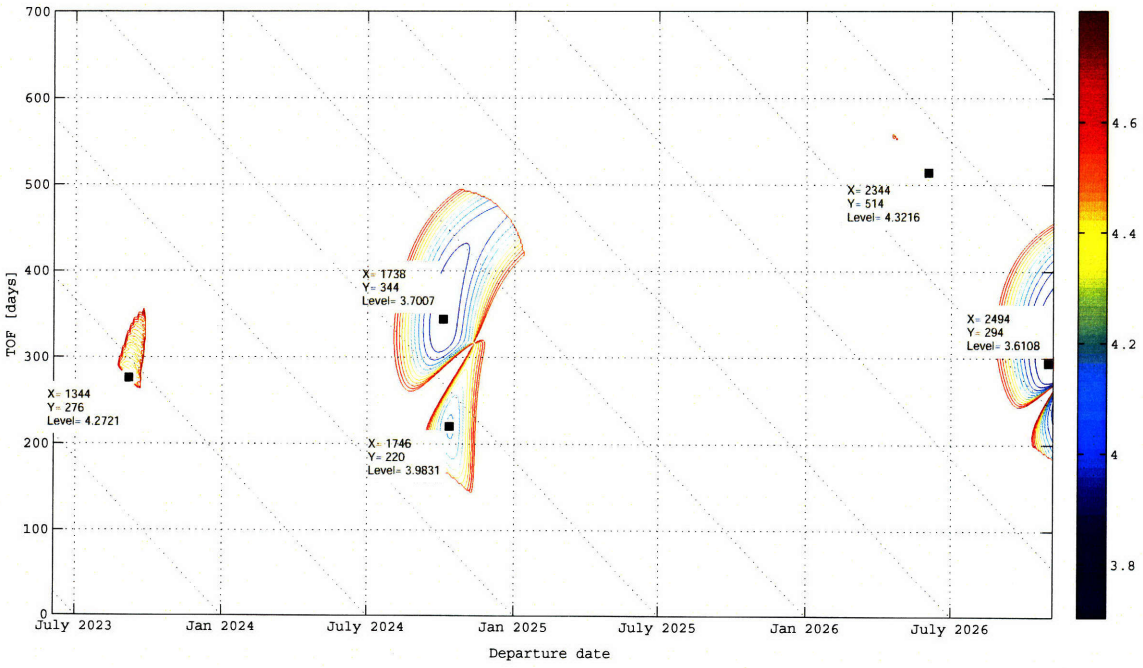


Figure 4-18: ΔV_{tot} contours: Earth-Mars direct/Earth-Venus-Mars flyby Period 2

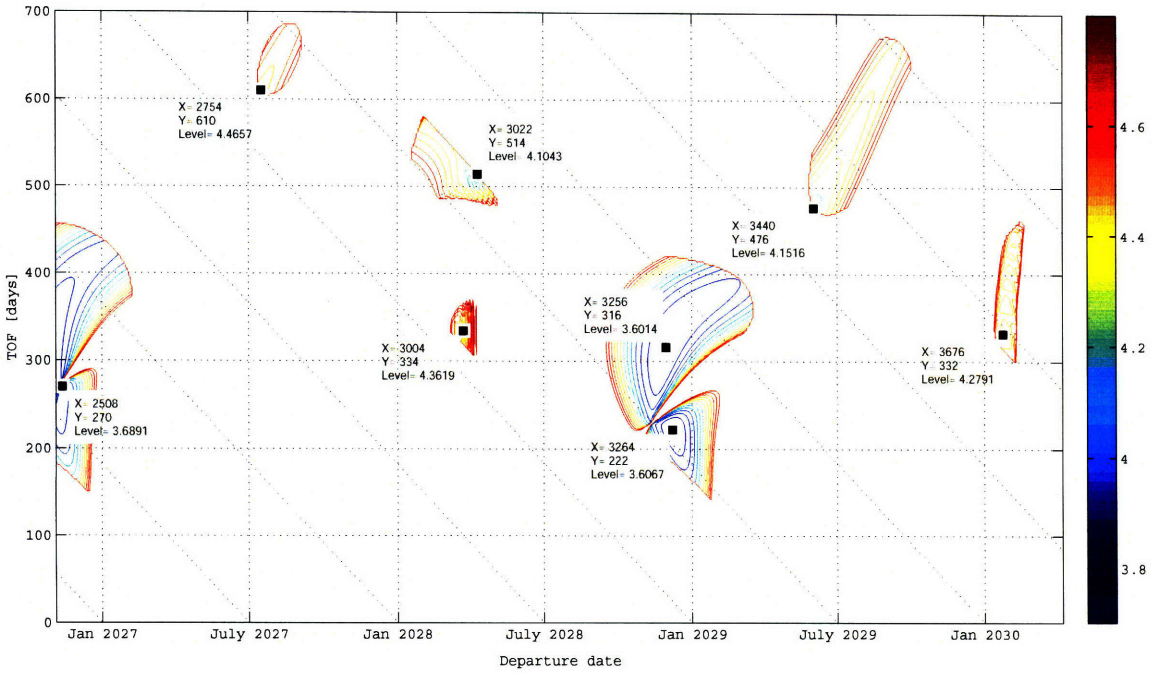


Figure 4-19: ΔV_{tot} contours: Earth-Mars direct/Earth-Venus-Mars flyby Period 3

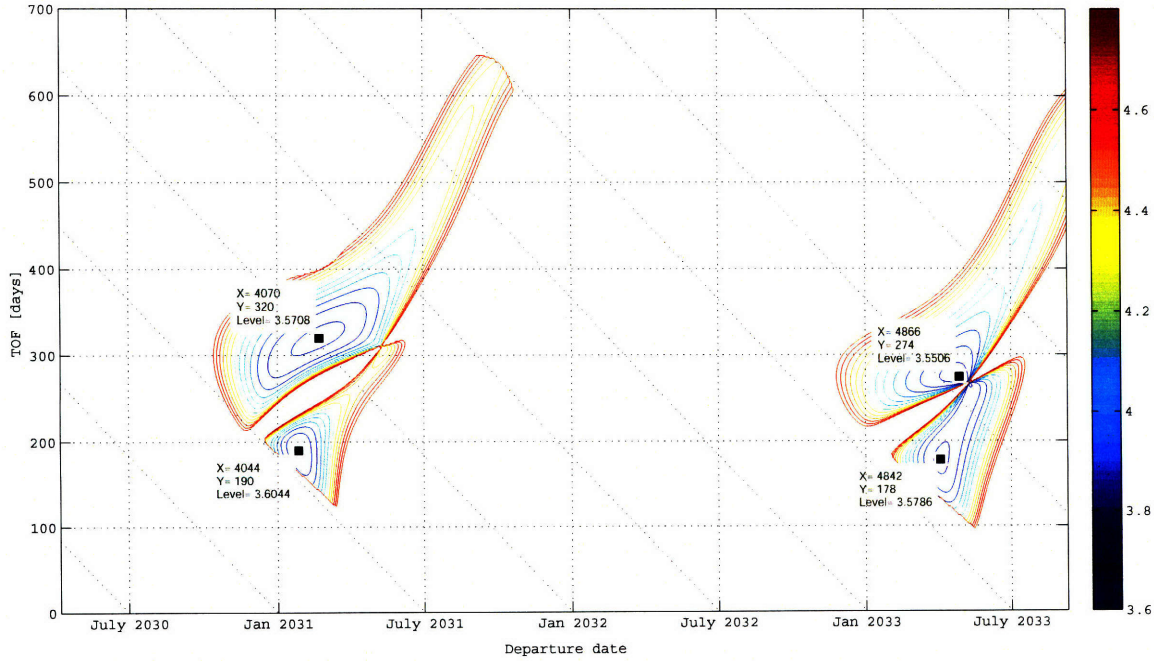


Figure 4-20: ΔV_{tot} contours: Earth-Mars direct/Earth-Venus-Mars flyby Period 4

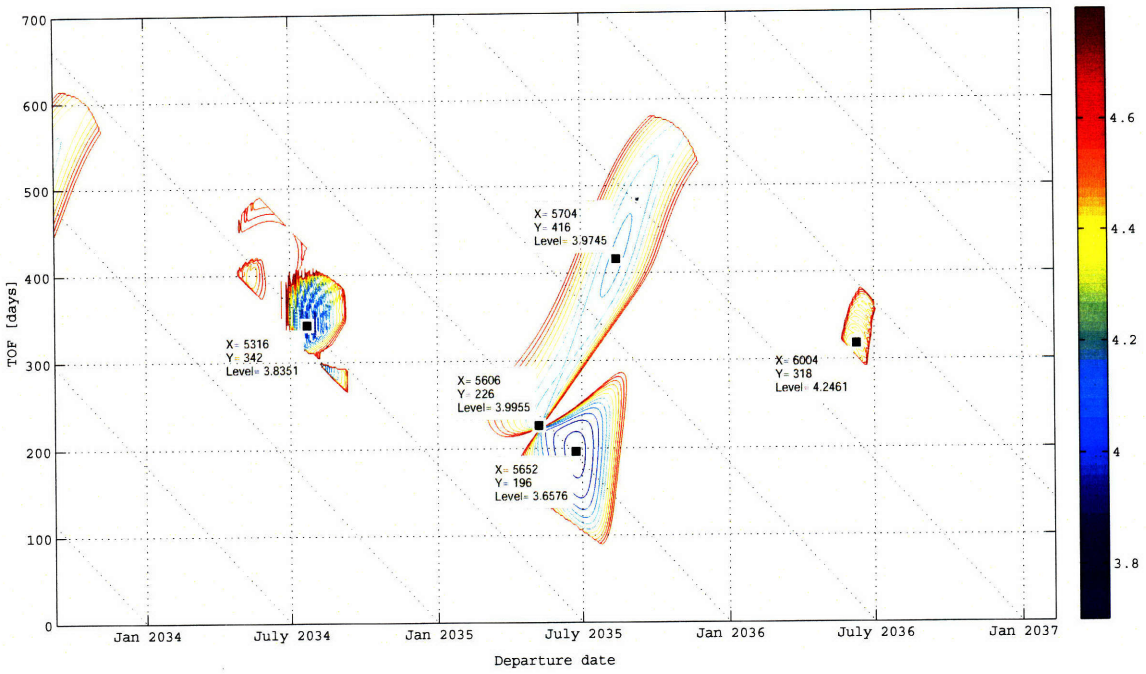


Figure 4-21: ΔV_{tot} contours: Earth-Mars direct/Earth-Venus-Mars flyby Period 5

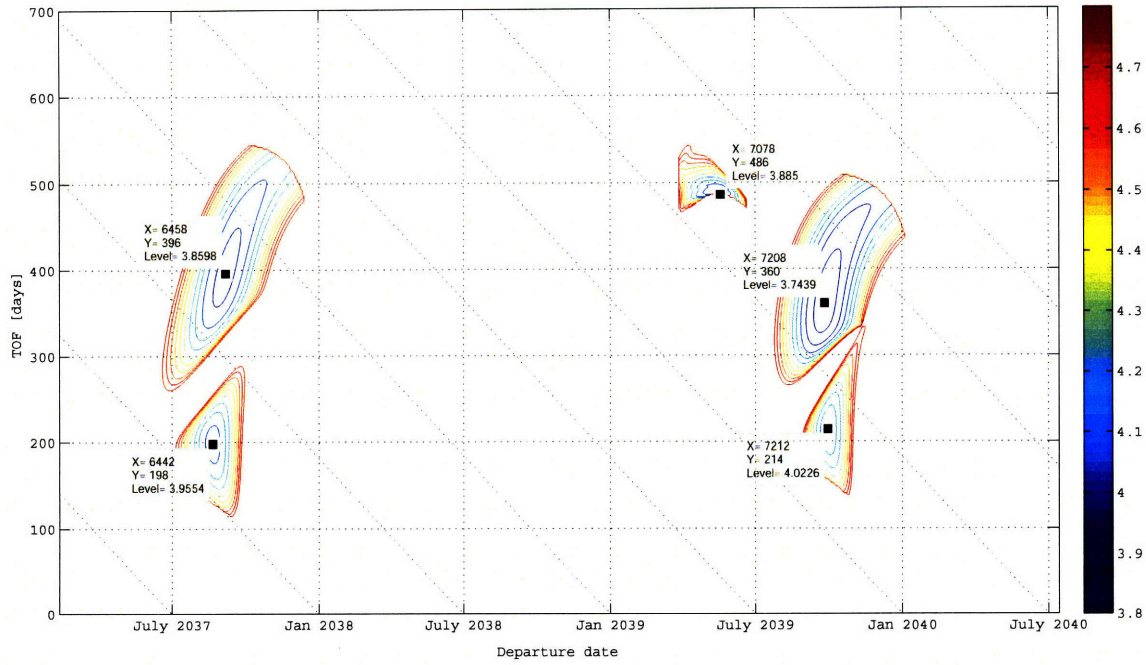


Figure 4-22: ΔV_{tot} contours: Earth-Mars direct/Earth-Venus-Mars flyby Period 6

Table 4.1: Earth-Mars trajectory data 2020-2040

#	Trajectory	Departure	Arrival	ΔV_{tot}	Competitiveness
1	2020 Earth-Venus-Mars	3/11/20	6/22/21	3.797	competitive
2	2020 Earth-Mars	7/17/20	1/27/21	3.788	competitive
3	2020 Earth-Mars	8/24/20	10/10/21	3.931	competitive
4	2021 Earth-Venus-Mars	9/28/21	1/25/23	4.068	dominated
5	2021 Earth-Venus-Mars	10/28/21	8/24/22	3.848	competitive
6	2022 Earth-Mars	9/07/22	3/30/23	4.013	competitive
7	2022 Earth-Mars	9/13/22	9/30/23	3.814	competitive
8	2023 Earth-Venus-Mars	9/06/23	6/08/24	4.272	competitive
9	2024 Earth-Mars	10/04/24	9/13/25	3.701	competitive
10	2024 Earth-Mars	10/12/24	5/20/25	3.983	competitive
11	2026 Earth-Venus-Mars	6/02/26	10/29/27	4.322	dominated
12	2026 Earth-Mars	10/30/26	8/20/27	3.611	competitive
13	2026 Earth-Mars	11/13/26	8/10/27	3.689	competitive
14	2027 Earth-Mars	7/17/27	3/18/29	4.466	dominated
15	2028 Earth-Venus-Mars	3/23/28	2/20/29	4.362	competitive
16	2028 Earth-Venus-Mars	4/10/28	9/06/29	4.104	dominated
17	2028 Earth-Mars	11/30/28	10/12/29	3.601	competitive
18	2028 Earth-Mars	12/08/28	7/18/29	3.607	competitive
19	2029 Earth-Mars	6/02/29	9/21/30	4.152	competitive
20	2030 Earth-Venus-Mars	1/24/30	12/22/30	4.279	competitive
21	2031 Earth-Mars	1/27/31	8/05/31	3.604	competitive
22	2031 Earth-Mars	2/22/31	1/08/32	3.571	competitive
23	2033 Earth-Mars	4/04/33	9/29/33	3.579	competitive
24	2033 Earth-Mars	4/28/33	1/27/34	3.551	competitive
25	2034 Earth-Venus-Mars	7/22/34	6/29/35	3.835	competitive
26	2035 Earth-Mars	5/08/35	12/20/35	3.996	dominated
27	2035 Earth-Mars	6/23/35	1/05/36	3.658	competitive
28	2035 Earth-Mars	8/14/35	10/03/36	3.975	competitive
29	2036 Earth-Venus-Mars	6/09/36	4/23/37	4.246	competitive
30	2037 Earth-Mars	8/21/37	3/07/38	3.955	competitive
31	2037 Earth-Mars	9/06/37	10/07/38	3.860	competitive
32	2039 Earth-Venus-Mars	5/19/39	9/16/40	3.885	dominated
33	2039 Earth-Mars	9/26/39	9/20/40	3.744	competitive
34	2039 Earth-Mars	9/30/39	5/01/40	4.023	competitive

4.4 Mars-Earth Trajectory

We should also investigate the return trajectory. This section presents the results for 2020-2040 Mars-Earth trajectories, which includes Mars-Earth direct trajectories and Mars-Venus-Earth flyby trajectories. As in the previous section, we first analyze the Mars-Earth direct trajectories using $C3_d$ and $C3_a$ contours, and then discuss the launch window using integrated ΔV_{tot} contours.

4.4.1 Mars-Earth Direct Trajectory

Figures 4-23 through 4-34 show the $C3_d$ and $C3_a$ contours for Period 1-6 in (4.2), respectively. These contours display $C3_d$ up to 30 km^2/s^2 and $C3_a$ up to 60 km^2/s^2. Note that the contours are not filtered by each other's constraints. Therefore, the craters in the figures do not mean exactly feasible regions. The data tips display the values of each local minimum.

Again, we can observe periodic patterns. The differences between x-coordinates of the data tips vary between 730 to 830, and the average is about 780, which is the Earth-Mars synodic period. The second window in 2022 is similar to the ninth in 2039 in terms of shape and value of local minimum, which implies that Mars-Earth direct opportunities are also governed by the synodic cycle.

The characteristics stated in Section 4.3.1 hold true with the return trajectory in a qualitative sense. Type I and II trajectories appear in separate craters in each window. The fourth to seventh launch windows (2026-2033) have relatively low $C3_d$ while relatively high $C3_a$.

One of the major differences from the Earth-Mars direct case is that compared to the Earth-Mars case, $C3_d$ is lower and $C3_a$ is higher for the Mars-Earth case. This is simply because Mars has a higher potential energy with respect to the Sun and thus it allows a spacecraft at Mars to have a lower kinetic energy than at Earth while the Earth-Mars and Mars-Earth transfer trajectories are basically identical. In addition, since Mars is smaller than Earth, the Martian gravitational field is weaker. Therefore, the ΔV_d required for trans-Earth injection can be much smaller to achieve the same

level of $C3_d$, which makes a return trip easier. This would allow us to have a higher upper bound for $C3_d$. However, taking into account other inconveniences of an alien environment, we stick to the same constraints for $C3_d$ in Eq. (4.1a).

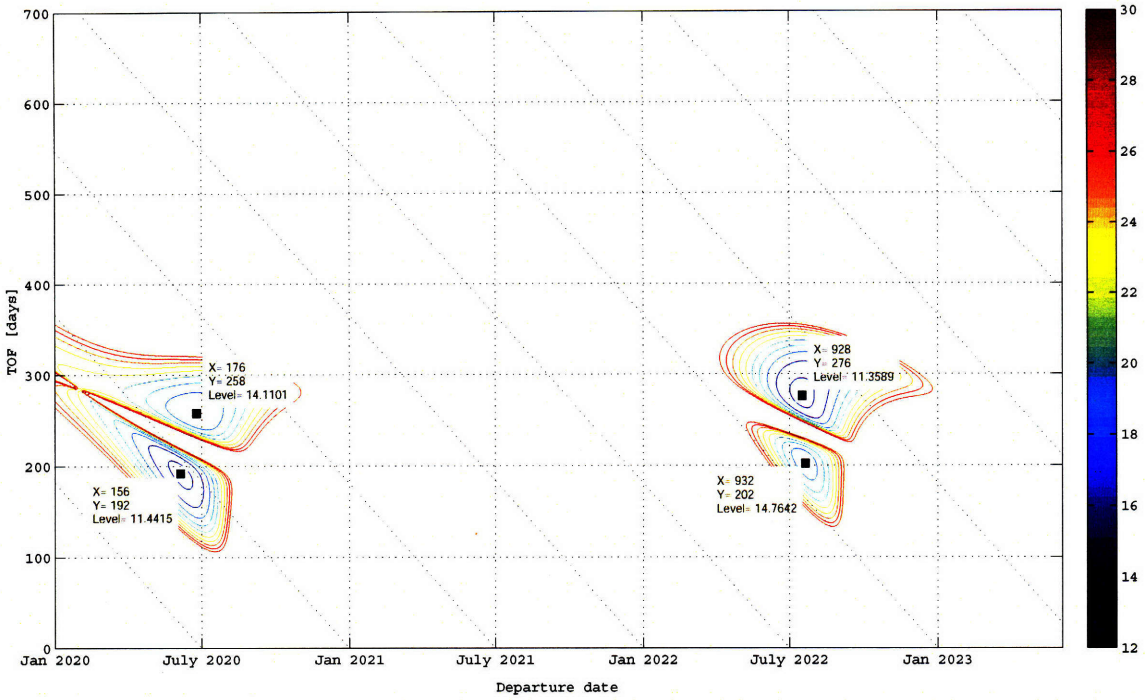


Figure 4-23: $C3_d$ contours: Mars-Earth direct Period 1

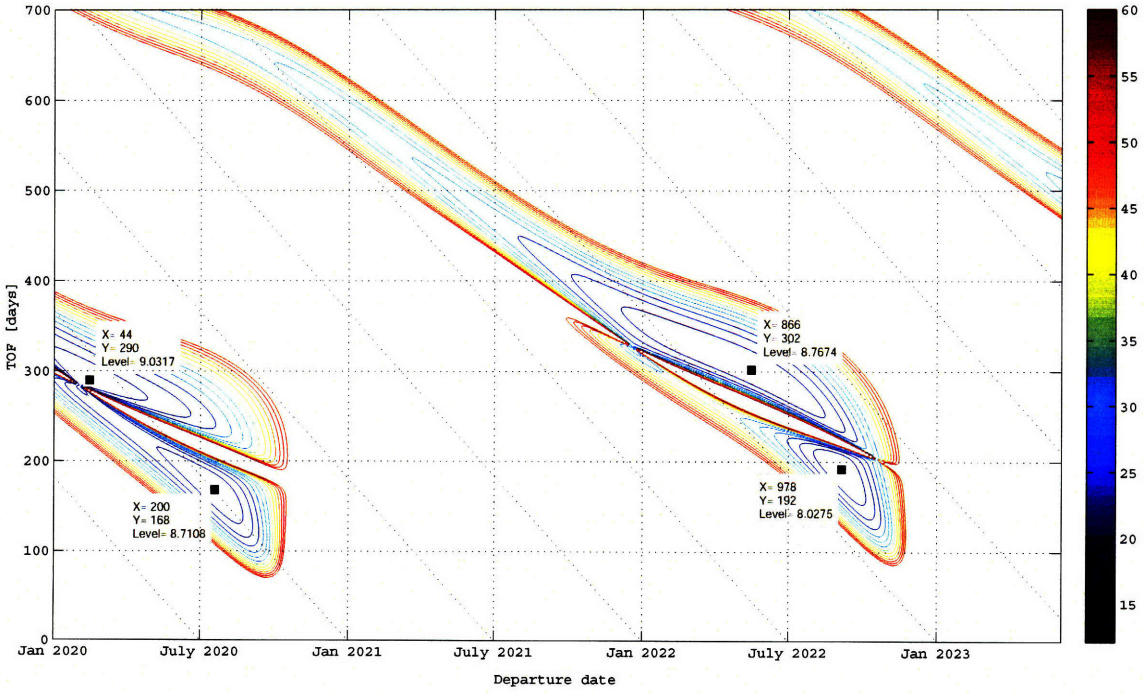


Figure 4-24: $C3_a$ contours: Mars-Earth direct Period 1

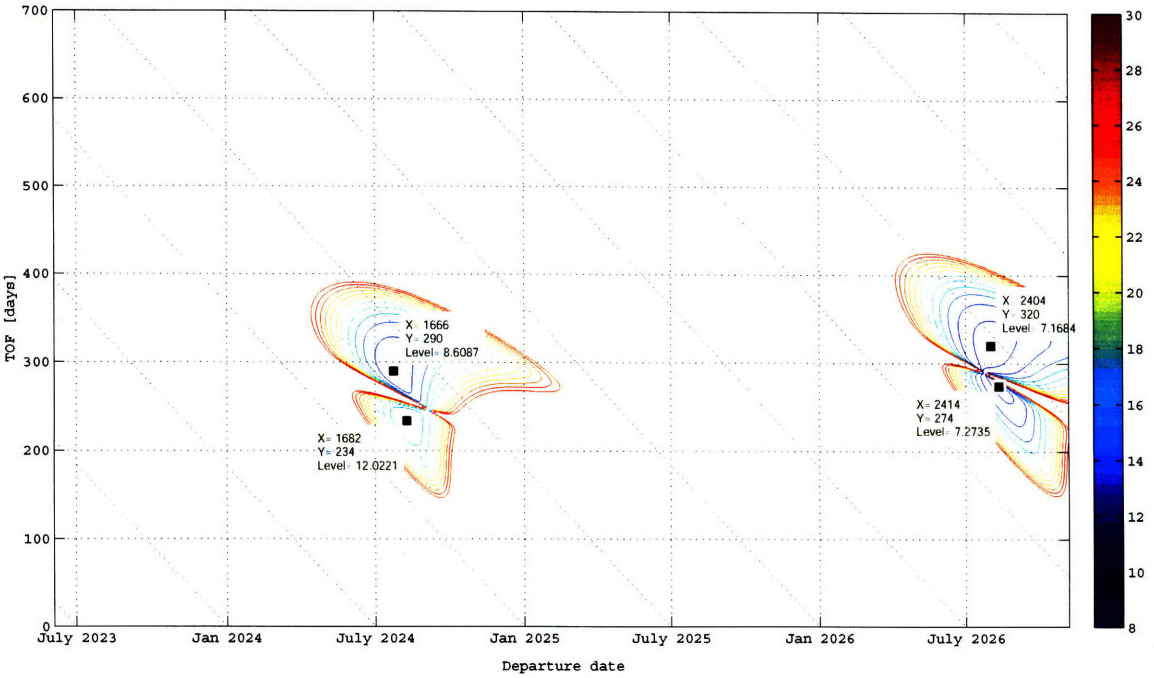


Figure 4-25: $C3_d$ contours: Mars-Earth direct Period 2

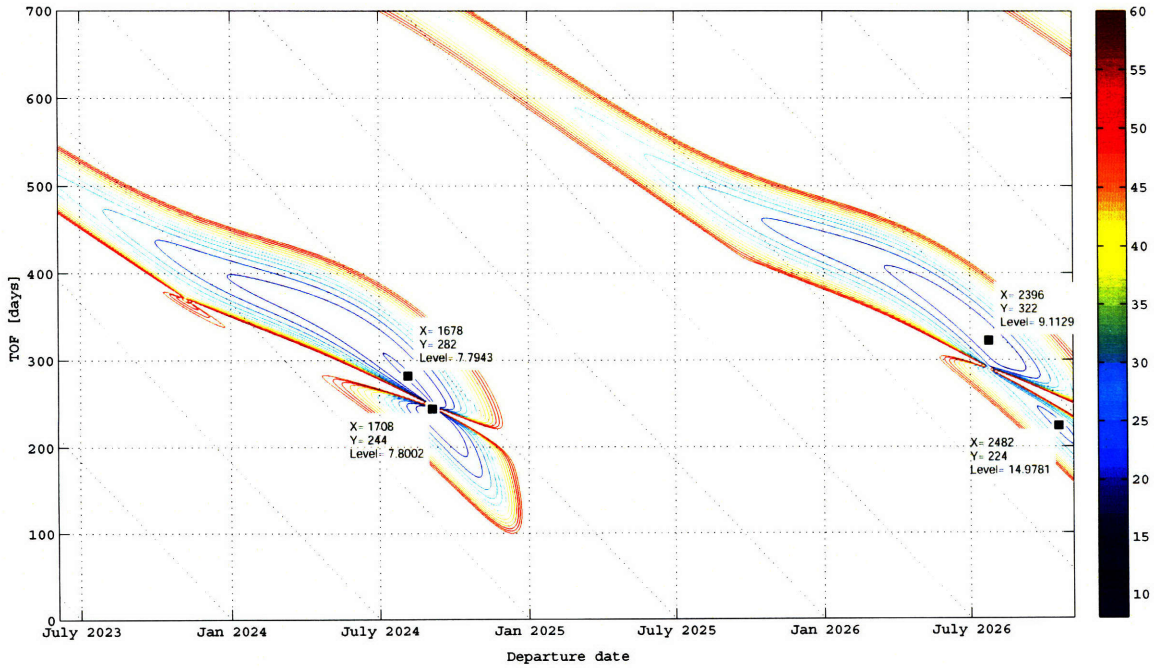


Figure 4-26: $C3_a$ contours: Mars-Earth direct Period 2

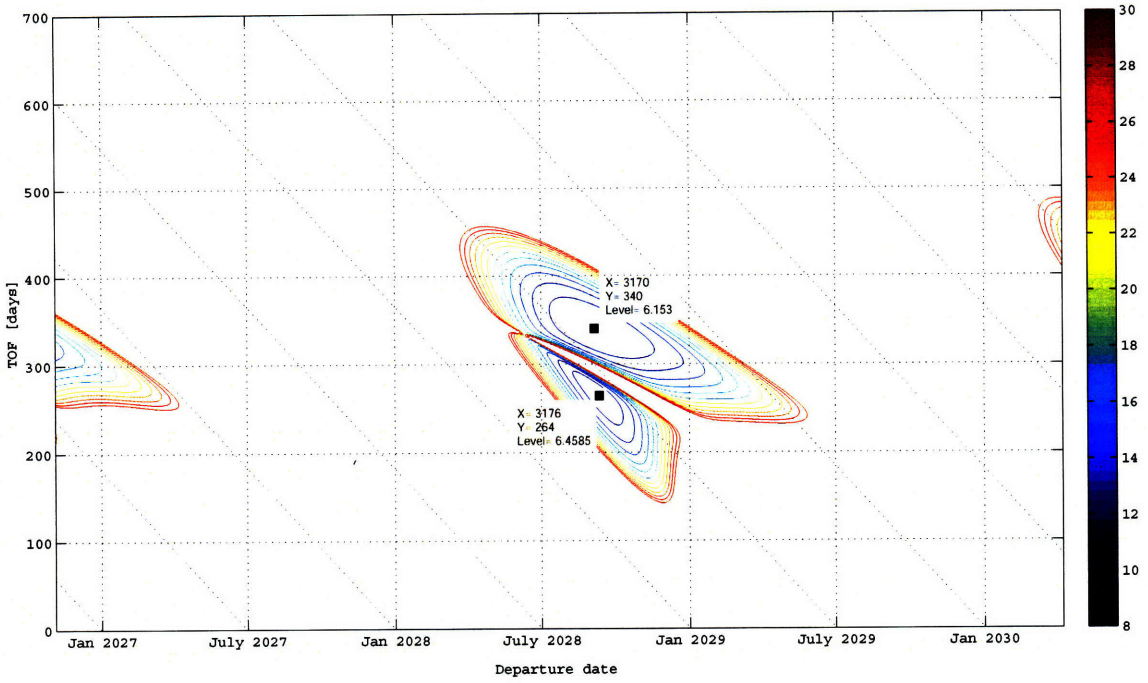


Figure 4-27: $C3_d$ contours: Mars-Earth direct Period 3

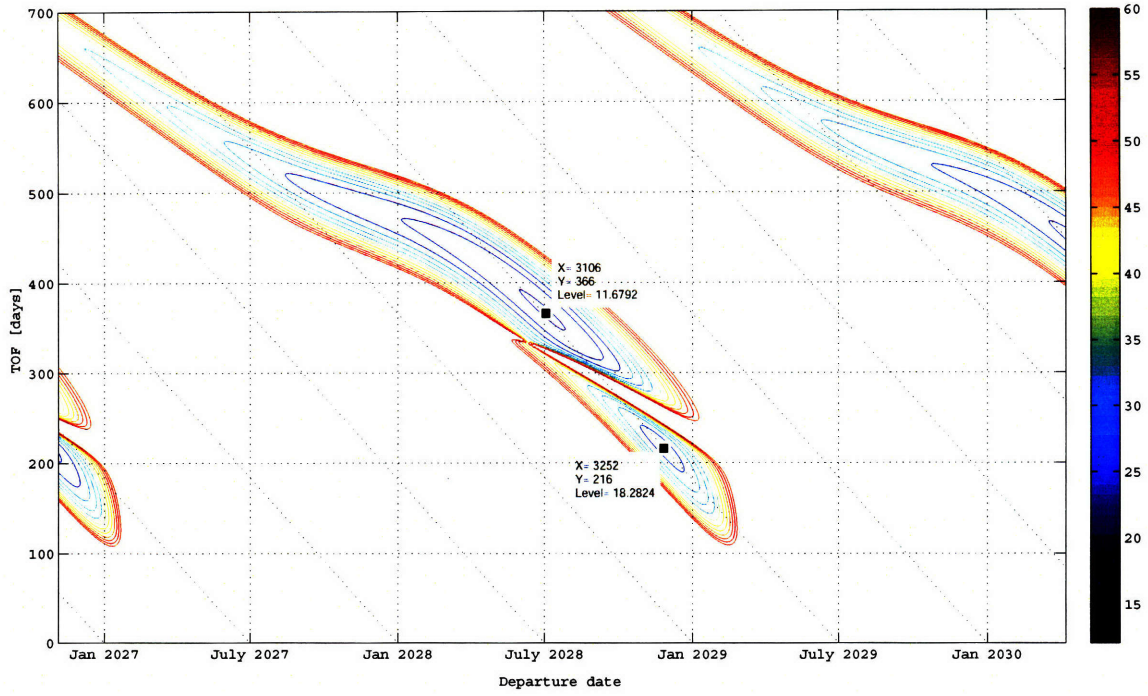


Figure 4-28: $C3_a$ contours: Mars-Earth direct Period 3

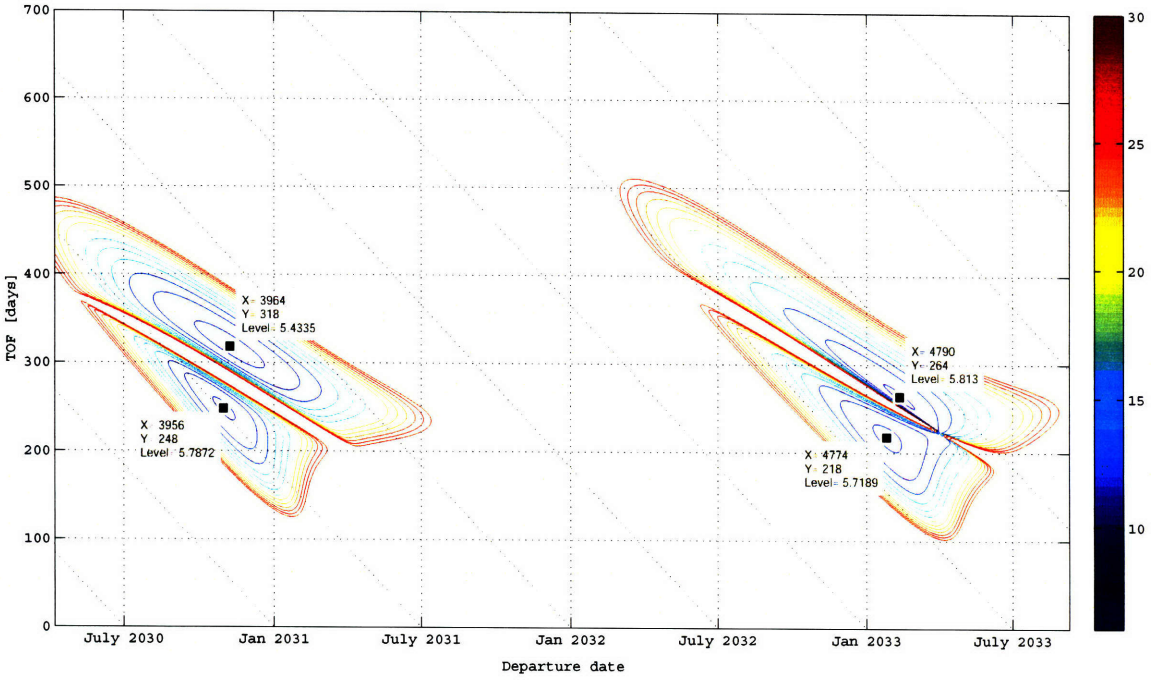


Figure 4-29: $C3_d$ contours: Mars-Earth direct Period 4

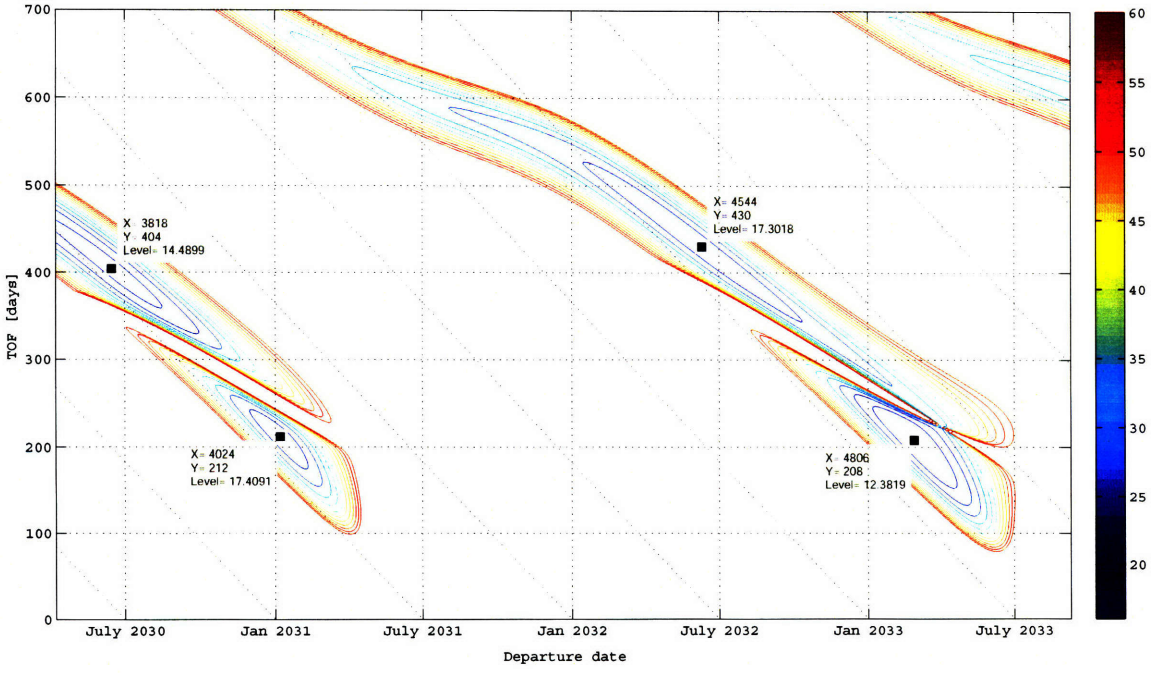


Figure 4-30: $C3_a$ contours: Mars-Earth direct Period 4

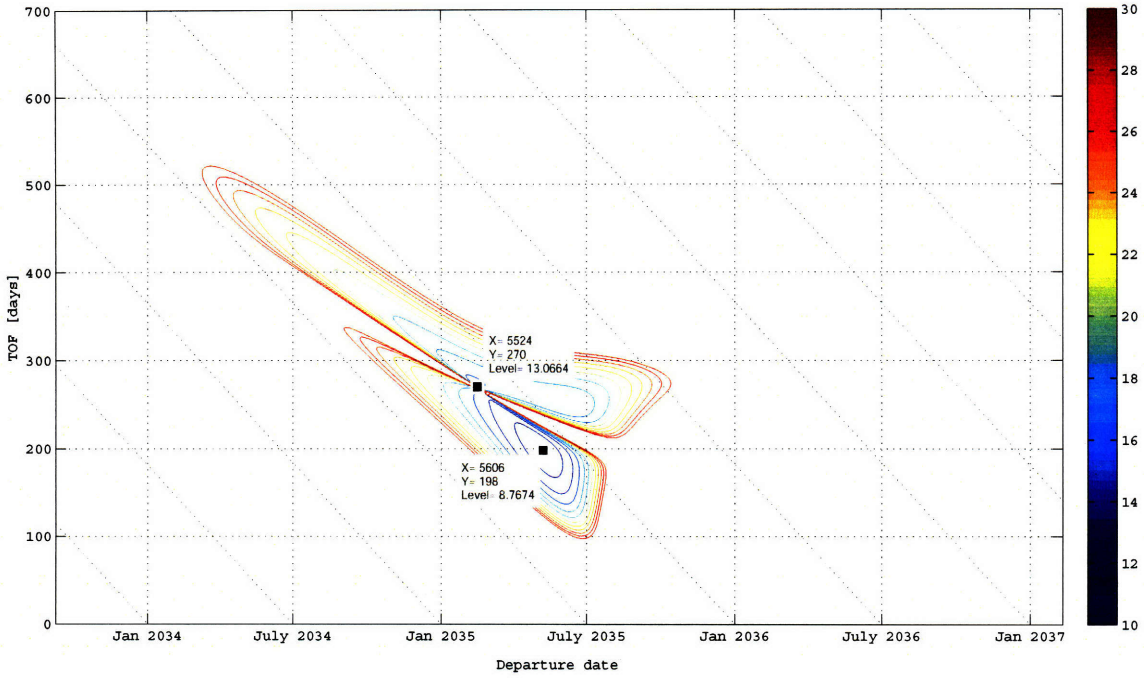


Figure 4-31: $C3_d$ contours: Mars-Earth direct Period 5

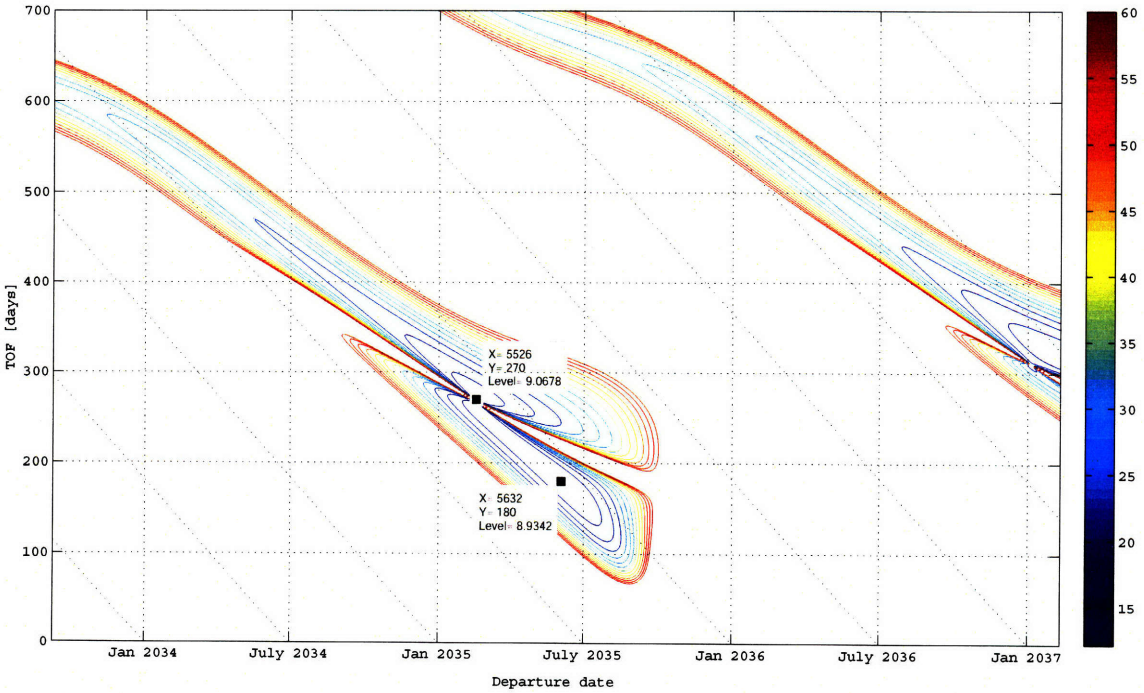


Figure 4-32: $C3_a$ contours: Mars-Earth direct Period 5

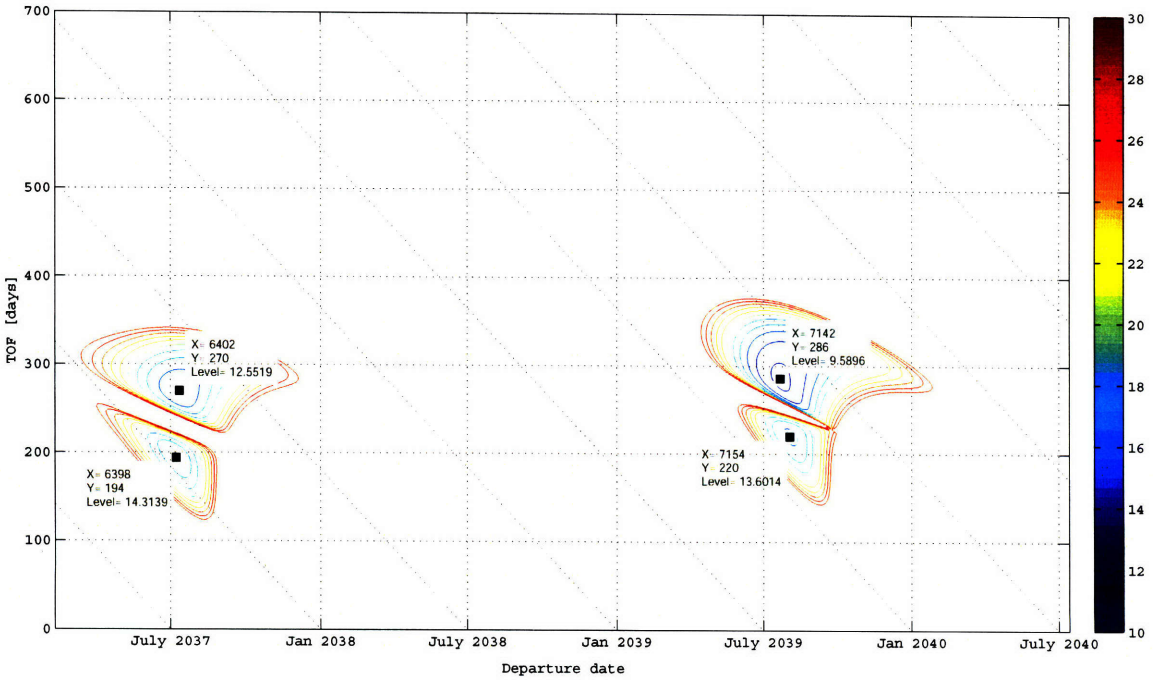


Figure 4-33: $C3_d$ contours: Mars-Earth direct Period 6

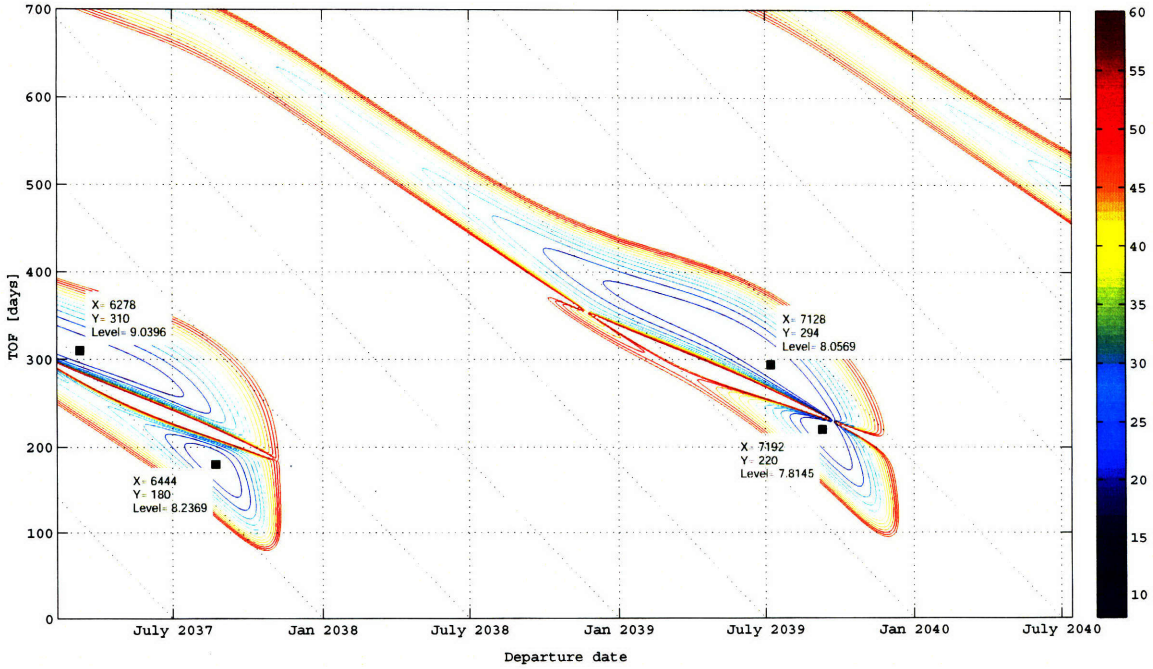


Figure 4-34: $C3_a$ contours: Mars-Earth direct Period 6

4.4.2 Mars-Earth Launch Window

In general, Mars-Venus-Earth flyby trajectories are said to be inefficient since we first have to head for Venus, an inferior planet, which requires higher energy. However, for the sake of completeness, we also investigated Mars-Venus-Earth flyby trajectories.

Figures 4-35 through 4-40 show the integrated ΔV_{tot} contours for Period 1-6 in (4.2), respectively. Since a $C3_d$ of 30 km^2/s^2 corresponds to a ΔV_d of about 3.9 km/s for Mars departure, these contours display ΔV_{tot} up to 4.2 km/s. Note that the contours are filtered by the constraints for $C3_d$, $C3_a$, and h_m. Therefore, the craters in the figures mean exactly feasible regions. The data tips display the values of each local minimum.

During 2020 through 2040, we have in total 10 direct windows and 6 flyby windows, which is less than 9 windows for the Earth-Venus-Mars case. But we can again observe three groups of similar patterns: the first one is early 2021, mid-2027, and early 2040, the second one is late 2022 and early 2029, and the third one is late 2030 through early 2031. We would say these three groups are also governed by a syzygistic cycle of 6.4 years.

The local minimum ΔV_{tot} with the data tip is listed in Table 4.2. The average of local minimum ΔV_{tot} for direct trajectories is about 2.32 while the average for flyby trajectories is around 3.45. Unlike the Earth-Mars case, flyby opportunities have much higher ΔV_{tot} compared to direct flights. Using the same criteria explained in Section 4.3.2, we determined the "competitiveness" of each opportunity.

As a result, we found that eight out of ten flyby opportunities are competitive. Therefore, in terms of launch windows, all the six flyby windows are competitive. Mars-Venus-Earth flyby trajectories have much higher ΔV_{tot} but make more launch windows available. Although in general, Mars-Venus-Earth flyby trajectories are said to be not preferable in terms of launch energy required, it is interesting that all the flyby windows are found to be "competitive" since they give new opportunities that cannot be replaced by the direct opportunities. Therefore, we need to perform a trade-off to see if we should take a flyby opportunity at the cost of ΔV_{tot}.

Just for reference, Mars-Venus $C3_d$ contours and Venus-Earth $C3_a$ contours are presented in Appendix C.

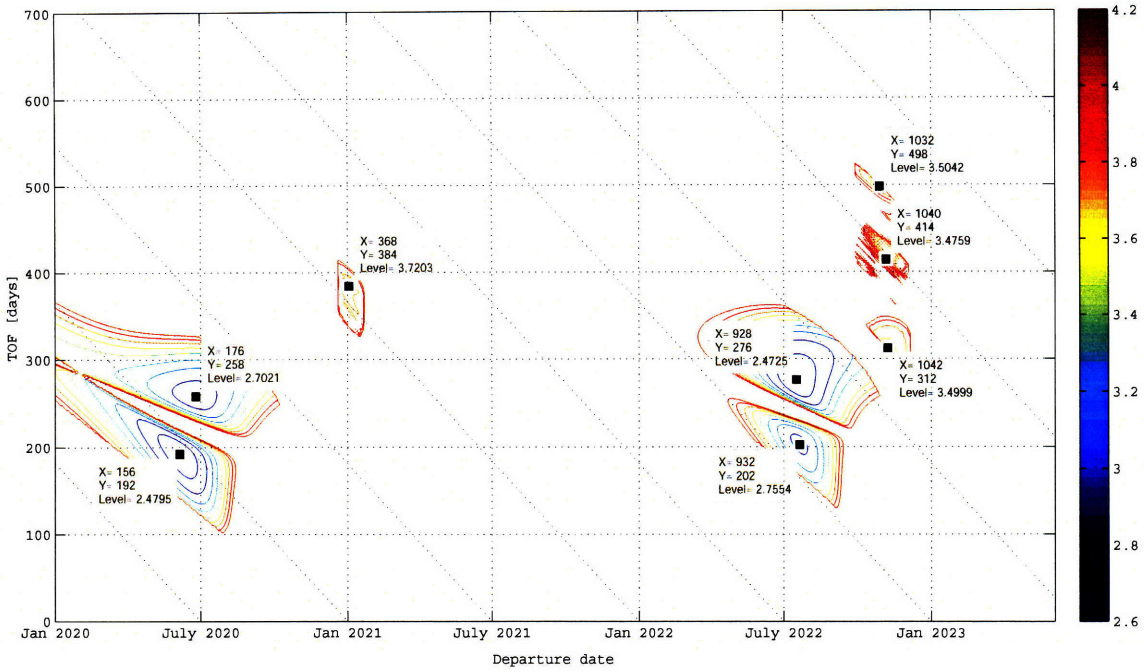


Figure 4-35: ΔV_{tot} contours: Mars-Earth direct/Mars-Venus-Earth flyby Period 1

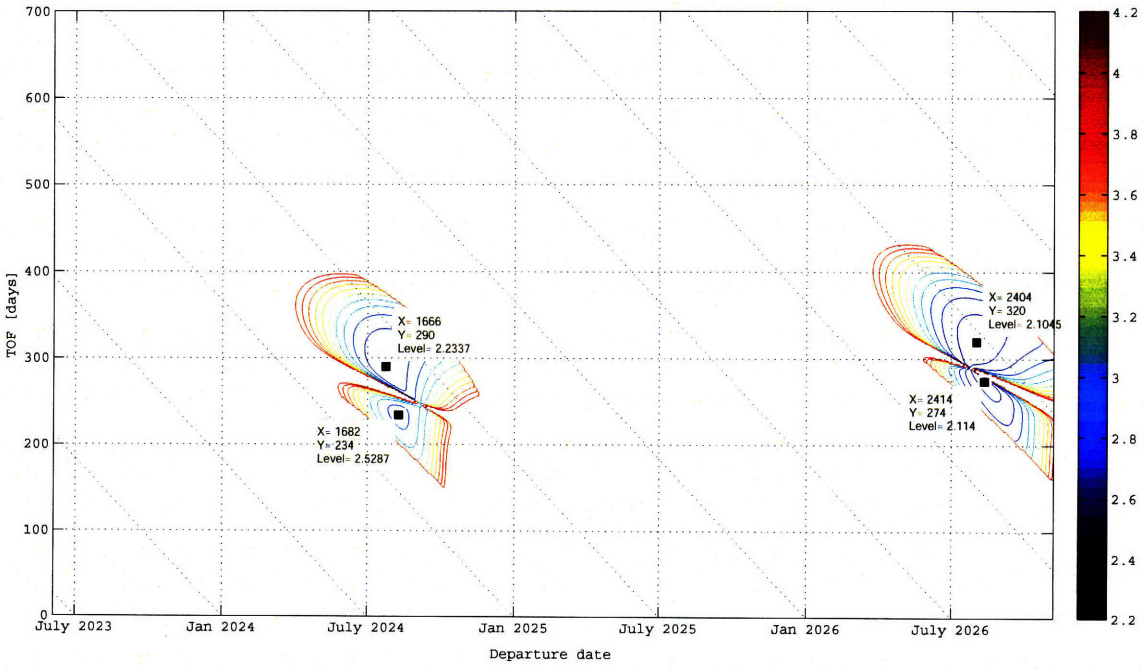


Figure 4-36: ΔV_{tot} contours: Mars-Earth direct/Mars-Venus-Earth flyby Period 2

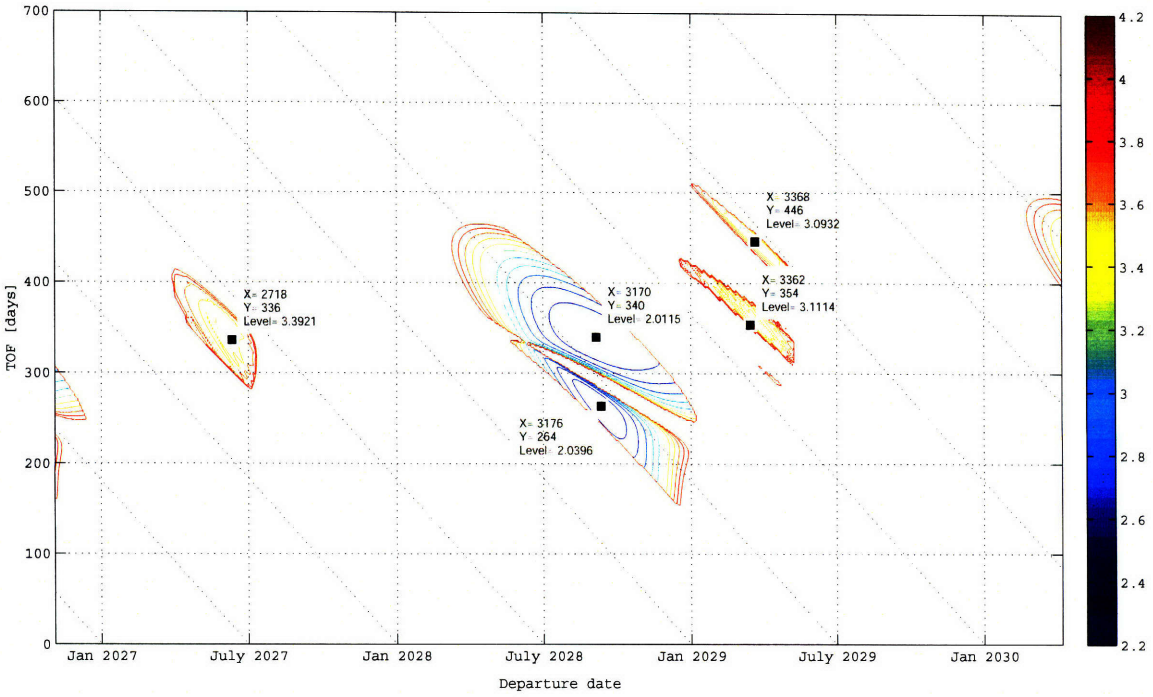


Figure 4-37: ΔV_{tot} contours: Mars-Earth direct/Mars-Venus-Earth flyby Period 3

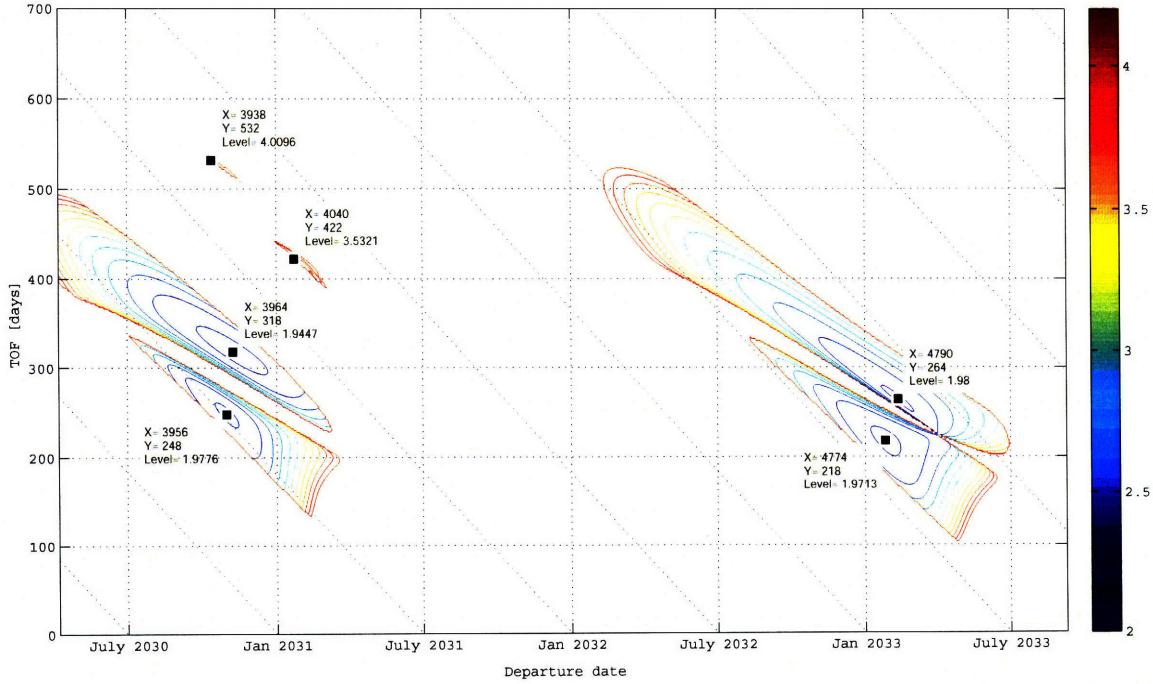


Figure 4-38: ΔV_{tot} contours: Mars-Earth direct/Mars-Venus-Earth flyby Period 4

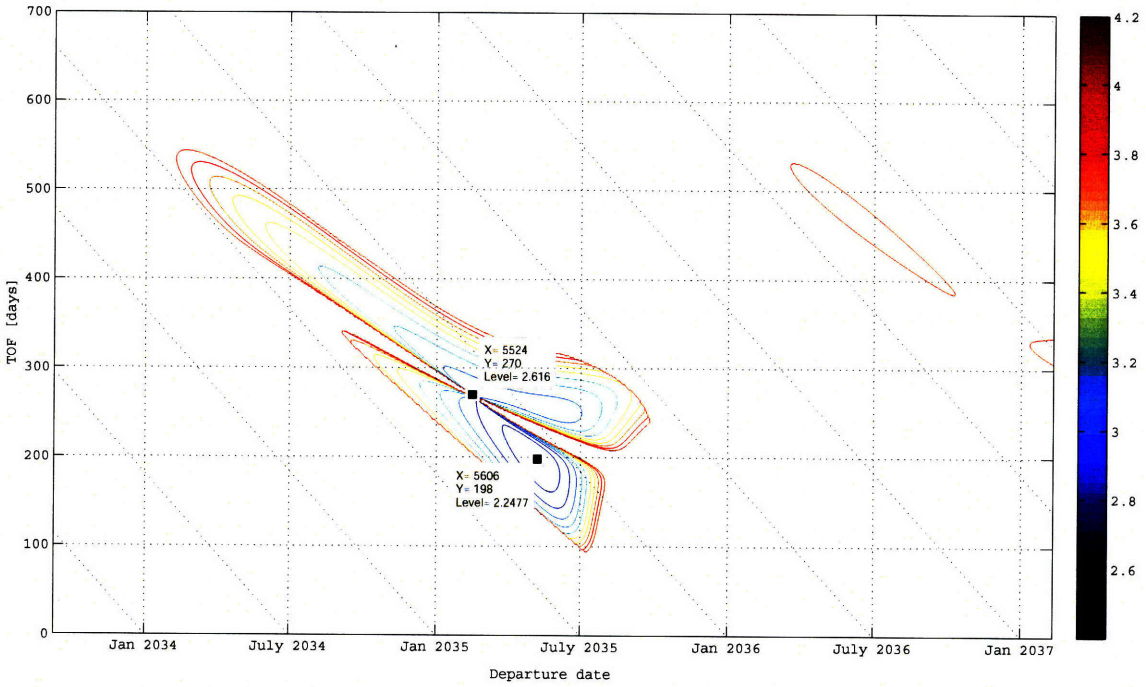


Figure 4-39: ΔV_{tot} contours: Mars-Earth direct/Mars-Venus-Earth flyby Period 5

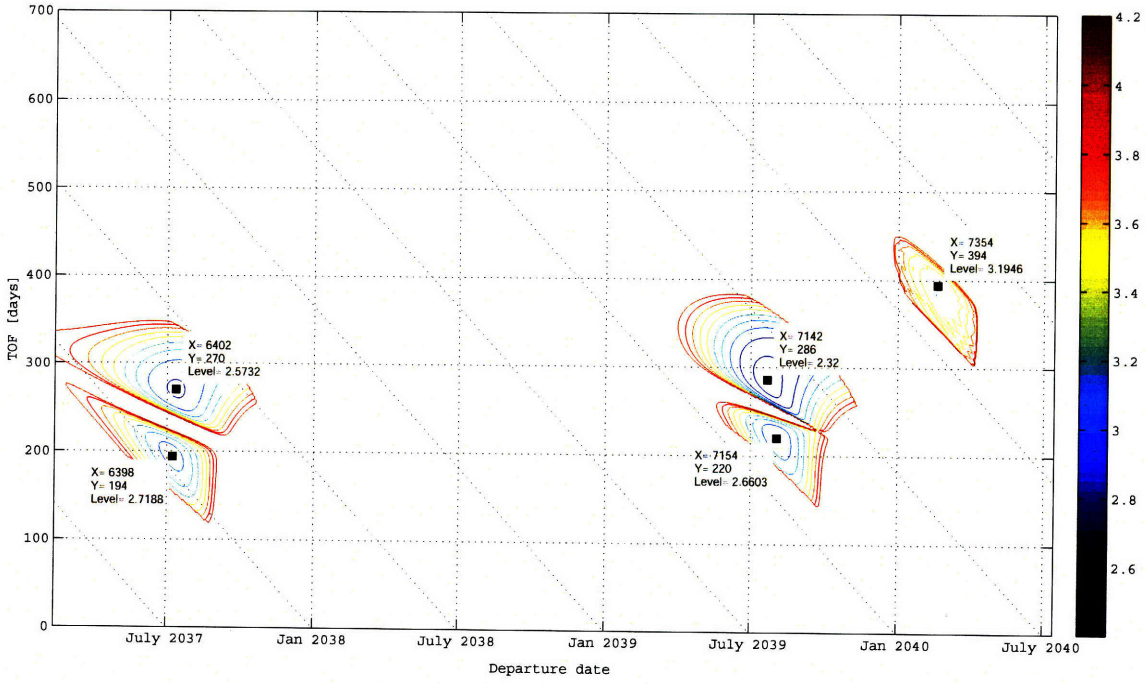


Figure 4-40: ΔV_{tot} contours: Mars-Earth direct/Mars-Venus-Earth flyby Period 6

Table 4.2: Mars-Earth trajectory data 2020-2040

#	Trajectory	Departure	Arrival	ΔV_{tot}	Competitiveness
1	2020 Mars-Earth	6/05/20	12/14/20	2.480	competitive
2	2020 Mars-Earth	6/25/20	3/10/21	2.702	dominated
3	2021 Mars-Venus-Mars	1/03/21	1/22/22	3.720	competitive
4	2022 Mars-Earth	7/17/22	4/19/23	2.473	competitive
5	2022 Mars-Earth	7/21/22	2/08/23	2.755	competitive
6	2022 Mars-Venus-Earth	10/29/22	3/10/24	3.504	competitive
7	2022 Mars-Venus-Earth	11/06/22	12/25/23	3.476	competitive
8	2022 Mars-Venus-Earth	11/08/22	9/16/23	3.500	dominated
9	2024 Mars-Earth	7/24/24	5/10/25	2.234	competitive
10	2024 Mars-Earth	8/09/24	3/31/25	2.529	competitive
11	2026 Mars-Earth	8/01/26	6/17/27	2.105	competitive
12	2026 Mars-Earth	8/11/26	5/12/27	2.114	competitive
13	2027 Mars-Venus-Earth	6/11/27	5/12/28	3.392	competitive
14	2028 Mars-Earth	9/05/28	8/11/29	2.012	competitive
15	2028 Mars-Earth	9/11/28	6/02/29	2.040	competitive
16	2029 Mars-Venus-Earth	3/16/29	3/05/30	3.111	competitive
17	2029 Mars-Venus-Earth	3/22/29	6/11/30	3.093	competitive
18	2030 Mars-Venus-Earth	10/13/30	3/28/32	4.010	dominated
19	2030 Mars-Earth	10/31/30	7/06/31	1.978	competitive
20	2030 Mars-Earth	11/08/30	9/22/31	1.945	competitive
21	2031 Mars-Venus-Earth	1/23/31	3/20/32	3.532	competitive
22	2033 Mars-Earth	1/26/33	9/01/33	1.971	competitive
23	2033 Mars-Earth	2/11/33	11/02/33	1.980	dominated
24	2035 Mars-Earth	2/15/35	11/12/35	2.616	dominated
25	2035 Mars-Earth	5/08/35	11/22/35	2.248	competitive
26	2037 Mars-Earth	7/08/37	1/18/38	2.719	competitive
27	2037 Mars-Earth	7/12/37	4/08/38	2.573	competitive
28	2039 Mars-Earth	7/22/39	5/03/40	2.320	competitive
29	2039 Mars-Earth	8/03/39	3/10/40	2.660	competitive
30	2040 Mars-Venus-Earth	2/19/40	3/19/41	3.195	competitive

Chapter 5

Conclusions and Future Work

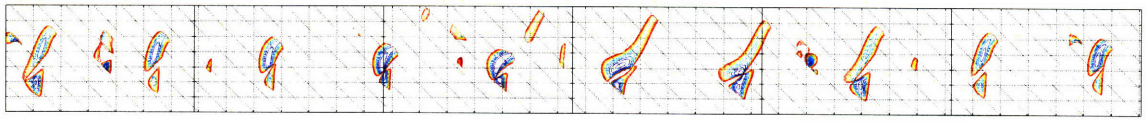
This chapter presents a summary of results and conclusions of this work, followed by suggestions for future work to deepen understanding of the results and apply this knowledge to future mission planning.

5.1 Summary and Conclusions

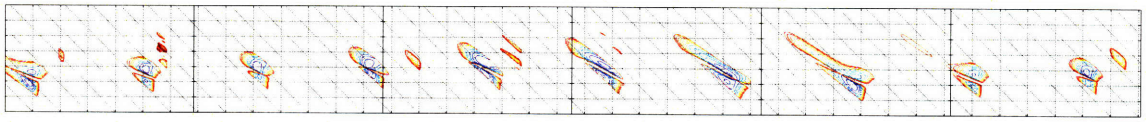
In Chapter 1, the background and motivation for this work were presented. The next several decades will see an increasing number of both unmanned and manned missions to Mars. Planning future missions requires trajectory data years in advance. Thus we decided to focus on the transfer trajectories to Mars and create a "launch window calendar", which would be useful for future mission planning.

Chapter 2 reviewed the fundamental astrodynamics and the two key issues in this study, the Lambert's problems and planetary flybys, and introduced important properties of the trajectories, which were needed for the implementation of MATLAB codes for interplanetary trajectory calculation.

Chapter 3 presented the MATLAB implementation of interplanetary trajectory programs and their validation by comparison with the data from JPL [22, 23]. The trajectory program developed in this work is a modularly-configured program that can calculate ballistic interplanetary trajectories between the planets of the solar system, including planetary flybys. The code validation was conducted using the JPL data in



(a) Earth-Mars direct/Earth-Venus-Mars flyby



(b) Mars-Earth direct/Mars-Venus-Earth flyby

Figure 5-1: ΔV_{tot} contours: Earth-Mars/Mars-Earth 2020-2040

the form of table (Table 3.2) and C3 contours (Figures 3-5 and 3-6). Both the results showed an excellent agreement. Thus, the trajectory programs of the present study are expected to give reliable results also for 2020-2040 Mars trajectories discussed in Chapter 4.

Having developed trajectory programs that can calculate both direct and flyby trajectories, Chapter 4 conducted a full-factorial investigation to create $C3_d$, $C3_a$, and ΔV_{tot} contours for Mars missions in the time frame 2020 to 2040, during which such missions seem most relevant. By visualizing the regions of some reasonable level of ΔV_{tot} on the contours, they will serve as a "launch window calendar," which is shown in Figures 5-1. Since the contours in these figures are filtered by the constraints for $C3_d$, $C3_a$, and h_m in Eqs. (4.1), the craters in the figures represent exactly the regions of feasible missions in terms of the constraints we assumed.

For a full-factorial computation of flyby trajectories, instead of wrapping Algorithm 3.8 in triple `"for"` statements for departure, encounter, and arrival dates, we took advantage of the completely decomposable nature of the problem, which greatly contributed to reducing the computation time. The total computation time to produce Figures 5-1 was about 500 hours (Intel®Core™2 Duo processor at 2.40 GHz) since they required the data of Earth-Mars, Earth-Venus, Venus-Mars, Mars-Earth, Mars-Venus, and Venus-Earth direct trajectories for the 7500-day period.

From C3 and ΔV_{tot} contours (Figures 4-5 through 4-40 in Chapter 4 and also Figures C-1 through C-24 in Appendix C), we found that in principle, launch windows

obey the synodic and/or syzygistic cycles of the planets.

For Earth-Mars trajectories during 2020 through 2040, it was found that we have in total 10 direct windows and 9 flyby windows. Table 4.1 lists the local minimum ΔV_{tot} of each opportunity and its "competitiveness" determined by the criteria in Section 4.3.2. As a result, we found that seven out of nine flyby windows are competitive. Earth-Venus-Mars flyby trajectories tend to have a relatively high ΔV_{tot} but give new opportunities. Having more launch windows available gives us flexibility in mission planning.

For Mars-Earth trajectories during 2020 through 2040, it was found that we have in total 10 direct windows and 6 flyby windows. Table 4.2 lists the local minimum ΔV_{tot} of each opportunity and its "competitiveness" determined by the criteria in Section 4.3.2. As a result, we found that all the six flyby windows are competitive. Mars-Venus-Earth flyby trajectories have much higher ΔV_{tot} but make more launch windows available. Although in general, Mars-Venus-Earth flyby trajectories are said to be not preferable in terms of launch energy required, it is interesting that all the flyby windows are found to be "competitive" since they give new opportunities that cannot be replaced by the direct opportunities. Therefore, we need to perform a trade-off to see if we should take a flyby opportunity at the cost of ΔV_{tot}.

5.2 Future Work

This section presents several possible future research directions.

The C3 and/or ΔV_{tot} data produced by a full-factorial computation will serve as a database for future mission planning. Therefore, the next step we should take is to apply these results to an actual mission design. For both unmanned and manned missions, we can use the data to perform a trade-off between C3 (or ΔV_{tot}), time of flight, departure date, and arrival date, on a mission-by-mission basis. Depending on a time window which a mission being planned falls on, we can weigh the advantages and disadvantages of direct and flyby trajectories.

For human Mars missions, we should also consider the total duration of the mis-

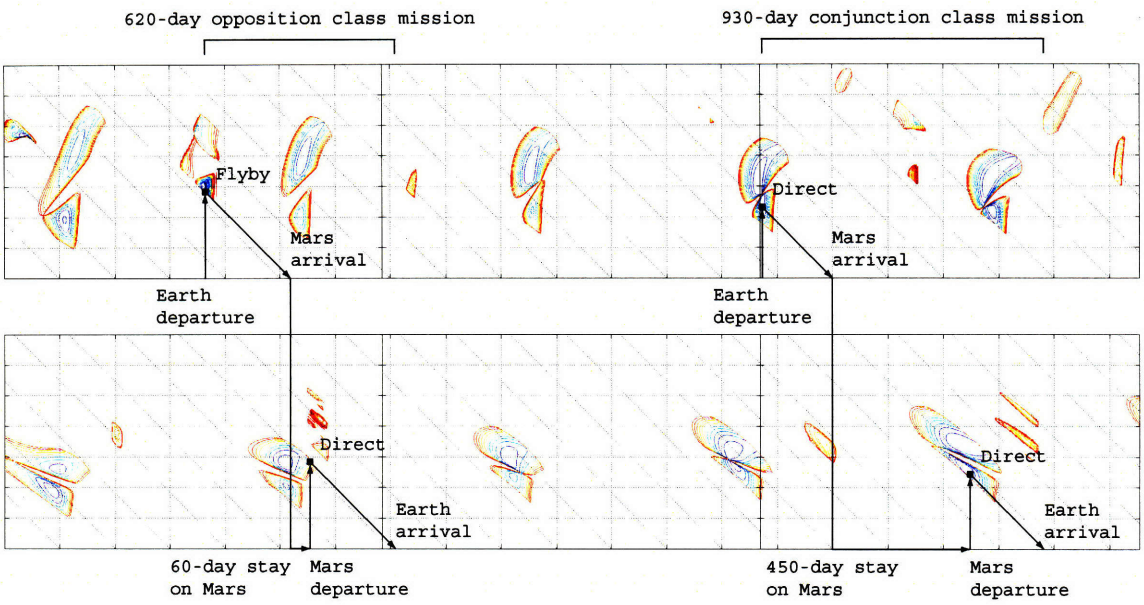


Figure 5-2: Examples of conjunction and opposition class Mars missions

sion. As stated in Chapter 1, Mars missions are divided into two types: conjunction class missions and opposition class missions. A conjunction class mission uses a direct trajectory for both legs of the trip, while an opposition class mission uses a flyby trajectory. Figure 5-2 shows the examples of conjunction and opposition class Mars missions. The example of a conjunction class in the figure is a 930-day mission with a 450-day stay on Mars, while the example of an opposition class is a 620-day mission with a 60-day stay on Mars. In this manner, we can use the ΔV_{tot} contours created in this thesis work as a tool to perform a possible trade-off between ΔV_{tot}, time of flight, time of stay on Mars, total duration of the mission, and so on when we consider the mission scenarios of Earth departure, Mars arrival, Mars stay, Mars departure, and Earth arrival.

Therefore, we should sort out all possible mission scenarios, or develop a tool that can find out from the database an optimal combination of dates of events for given requirements and constraints such as a time window, time of stay on Mars, total duration of the mission, upper bounds for $C3_d$ and $C3_a$, and so on.

If the future advances in technology such as propulsion systems and thermal protection systems can push the envelope of acceptable $C3_d$, $C3_a$ and ΔV_{PFM} to higher

level, we will be able to expand the launch windows. In addition, we assumed in this study that a powered flyby maneuver is the velocity impulse at periapsis only. Instead, if we can optimize the maneuver on the flyby trajectory near a planet to achieve the desired turn angle and velocity, the flyby launch windows might be improved to some extent. For direct trajectories, considering type III trajectories and beyond (more than one trip around the Sun) might also extend the launch windows even though they have inherently long flight times.

In parallel with this thesis work, we conducted a study on optimization of Earth-Mars interplanetary trajectory selection [24]. This study developed a tool that detects the optimal combination of Earth departure, Venus encounter, and Mars arrival dates for a given time window, using sequential quadratic programming (SQP), which is one of the gradient methods, and genetic algorithm (GA), which is one of the heuristic approaches. We found that SQP did not work well because of the highly non-convex nature of the problem especially with respect to t_2, as explained in Section 4.2.2. We also found that GA had difficulty in finding a feasible solution due to considerably restricted feasible regions in the t_1-t_2-t_3 space. Therefore, one suggestion for the future work on optimization tools is GA-SQP hybrid approach. We might be able to solve this optimization problem by first finding a feasible point by GA and then approaching an optimal solution by SQP. We should also attempt other methods such as simulated annealing (SA), which might be more suitable for this problem.

On a final note, the results presented in Chapter 4 are only one application of the trajectory programs developed in this thesis for 2020-2040 Mars missions. Since the programs are applicable to any other planet in the solar system and any other time frame, they can calculate the trajectories, produce the database, and create a "launch window calendar" on an as-needed basis.

Appendix A

MATLAB Algorithms

This appendix lists MATLAB scripts which implement all of the algorithms presented in Chapter 3. The programs use only the most basic features of MATLAB and are commented so as to make reading the code as easy as possible.

A.1 Julian Day Number

```
% ------------------------------------------------------------------
% Julian day number
  function JD=JulianDay(year,month,day,hour)
% ----------------------------------------
% Input: year(1901-2099),month(1-12),day(1-31),hour(0-24)
% Output: Julian day number (JD[day])
% ------------------------------------------------------------------

J0=367*year-fix(7*(year+fix((month+9)/12))/4)...
   +fix(275*month/9)+day+1721013.5;
JD=J0+hour/24; % Julian day number [day]
% ------------------------------------------------------------------
```

A.2 Time of Flight

```
% ------------------------------------------------------------------
```

```
% Time of flight
  function TOF=FlightTime(year1,month1,day1,hour1,...
                          year2,month2,day2,hour2)
% ------------------------------------------------
% Input:
% Departure date (year1,month1,day1,hour1)
% Arrival date (year2,month2,day2,hour2)
% Output:
% Time of flight (TOF[day])
% ------------------------------------------------------------------

JD1=JulianDay(year1,month1,day1,hour1); % departure date
JD2=JulianDay(year2,month2,day2,hour2); % arrival date
TOF=JD2-JD1; % time of flight [day]
% ------------------------------------------------------------------
```

A.3 Kepler's Equation

```
% ------------------------------------------------------------------
% Eccentric anomaly
  function E=KeplerE(e,M)
% ------------------------
% Input: Eccentricity (e[-]), Mean anomaly (M[rad])
% Output: Eccentric anomaly (E[rad])
%
% Newton's Method
% E-e*sin(E)=M
% ------------------------------------------------------------------

tol=1e-8; % error tolerance

if M<pi
    E=M+e/2;
else
    E=M-e/2;
end
```

```
ratio=1;
while abs(ratio)>tol
    ratio=(E-e*sin(E)-M)/(1-e*cos(E));
    E=E-ratio;
end
% ------------------------------------------------------------------
```

A.4 State Vector

```
% ------------------------------------------------------------------
% State Vector
  function [R,V]=StateVector(h,e,RA,incl,w,TA)
% ------------------------------------------
% Input: Orbital elements (OE)
% Output: State vector (R,V)
%
% Angular momentum (h[au^2/year])
% Eccentricity (e[-])
% Right ascension of the ascending node (RA[rad])
% Inclination (incl[rad])
% Argument of perihelion (w[rad])
% True anomaly (TA[rad])
% ------------------------------------------------------------------

% Declare global variable
global mu

% Constant
mu=4*pi^2; % gravitational parameter [au^3/year^2]

% 1. Calculate position vector Rp in perifocal coordinates
Rp=(h^2/mu)*(1/(1+e*cos(TA)))*[cos(TA);sin(TA);0]; %[au]

% 2. Calculate velocity vector Vp in perifocal coordinates
Vp=(mu/h)*[-sin(TA);e+cos(TA);0]; %[au/year]
```

```
% 3. Calculate transformation matrix QpX
R3_RA=[ cos(RA)   sin(RA)   0;
       -sin(RA)   cos(RA)   0;
           0         0      1];
R1_i=[1        0           0;
      0    cos(incl)   sin(incl);
      0   -sin(incl)   cos(incl)];
R3_w=[ cos(w)   sin(w)   0;
      -sin(w)   cos(w)   0;
          0        0     1];
QXp=R3_w*R1_i*R3_RA;
QpX=QXp';

% 4. Calculate R,V
R=QpX*Rp; %[au]
V=QpX*Vp; %[au/year]
% ------------------------------------------------------------------
```

A.5 Planetary Ephemeris

```
% ------------------------------------------------------------------
% Planetary Ephemeris
  function [R,V,OE,OEe]=PlanetaryEphemeris(planet_id,...
                                           year,month,day,hour)
% ------------------------------------------------------------
% Input:
% Planet (planet_id)
% Date (year,month,day,hour)
% Output:
% Orbital elements (OE)
% State vector (R,V)
%
% planet_id
%   1: Mercury
%   2: Venus
```

```
%   3: Earth
%   4: Mars
%   5: Jupiter
%   6: Saturn
%   7: Uranus
%   8: Neptune
%   9: Pluto
%
% Heliocentric orbital elements (OE)
% [h;e;RA;incl;w;TA]
% [a;w_tilde;L;M;E]
%         h: angular momentum [au^2/year]
%         e: eccentricity [-]
%        RA: right ascension [rad]
%      incl: inclination [rad]
%         w: argument of perihelion [rad]
%        TA: true anomaly [rad]
%         a: semimajor axis [au]
%   w_tilde: longitude of perihelion (=RA+w) [rad]
%         L: mean longitude (=w_tilde+M) [rad]
%         M: mean anomaly [rad]
%         E: eccentric anomaly [rad]
%
% State vector R,V
%   R: heliocentric position vector [au]
%   V: heliocentric velocity vector [au/year]
% ------------------------------------------------------------------

% Declare global variable
global mu

% Constant
mu=4*pi^2; % gravitational parameter [au^3/year^2]

% Obtain data for selected planet
[J2000OE,J2000OErates]=PlanetaryElements(planet_id);
```

```
% 1. Calculate Julian day number
JD=JulianDay(year,month,day,hour); %[day]

% 2. Calculate T0
T0=(JD-JulianDay(2000,1,1,12))/36525;

% 3. Calculate Q
Q=J2000OE+J2000OErates*T0;
a=Q(1); %[au]
e=Q(2); %[-]

% 4. Calculate h
h=sqrt(mu*a*(1-e^2)); %[au^2/year]

% 5. Calculate angular elements
incl=Q(3); %[deg]
RA=zero360(Q(4)); %[deg]
w_tilde=zero360(Q(5)); %[deg]
L=zero360(Q(6)); %[deg]
w=zero360(w_tilde-RA); %[deg]
M=zero360(L-w_tilde); %[deg]
incl=incl*pi/180; %[rad]
RA=RA*pi/180; %[rad]
w_tilde=w_tilde*pi/180; %[rad]
L=L*pi/180; %[rad]
w=w*pi/180; %[rad]
M=M*pi/180; %[rad]

% 6. Calculate E
E=KeplerE(e,M); %[rad]

% 7. Calculate TA
TA=2*atan(sqrt((1+e)/(1-e))*tan(E/2)); %[rad]
TA=zero360(TA*180/pi)*pi/180; %[rad]
OE=[h;e;RA;incl;w;TA];
```

```
OEe=[a;w_tilde;L;M;E];

% 8. Calculate state vector R,V
[R,V]=StateVector(h,e,RA,incl,w,TA);
% ------------------------------------------------------------------

% Subfunctions used in the main body:

% ------------------------------------------------------------------
% Planetary elements
  function [J2000OE,J2000OErates]=PlanetaryElements(planet_id)
% ------------------------------------------------------------

% Planetary orbital elements J2000
% a[au],e[-],i[deg],RA[deg],w_tilde[deg],L[deg]
J2000elements=...
[ 0.38709893 0.20563069  7.00487  48.33167  77.45645 252.25084;
  0.72333199 0.00677323  3.39471  76.68069 131.53298 181.97973;
  1.00000011 0.01671022  0.00005 -11.26064 102.94719 100.46435;
  1.52366231 0.09341233  1.85061  49.57854 336.04084 355.45332;
  5.20336301 0.04839266  1.30530 100.55615  14.75385  34.40438;
  9.53707032 0.05415060  2.48446 113.71504  92.43194  49.94432;
 19.19126393 0.04716771  0.76986  74.22988 170.96424 313.23218;
 30.06896348 0.00858587  1.76917 131.72169  44.97135 304.88003;
 39.48168677 0.24880766 17.14175 110.30347 224.06676 238.92881];

% Centennial rates J2000
% a_dot[au/Cy],e_dot[1/Cy],i_dot["/Cy],
% RA_dot["/Cy],w_tilde_dot["/Cy],L_dot["/Cy]
J2000rates=...
[ 0.00000066  0.00002527 -23.51   -446.30   573.57 538101628.29;
  0.00000092 -0.00004938  -2.86   -996.89  -108.80 210664136.06;
 -0.00000005 -0.00003804 -46.94 -18228.25  1198.28 129597740.63;
 -0.00007221  0.00011902 -25.47  -1020.19  1560.78  68905103.78;
  0.00060737 -0.00012880  -4.15   1217.17   839.93  10925078.35;
 -0.00301530 -0.00036762   6.11  -1591.05 -1948.89   4401052.95;
```

```
  0.00152025 -0.00019150  -2.09  -1681.4   1312.56   1542547.79;
 -0.00125196  0.00002514  -3.64   -151.25  -844.43    786449.21;
 -0.00076912  0.00006465  11.07    -37.33  -132.25    522747.90];

J2000OE=J2000elements(planet_id,:);
J2000OErates=J2000rates(planet_id,:);

% Convert from ["] to [deg]
J2000OErates(3:6)=J2000OErates(3:6)/3600;

return
% ------------------------------------------------------------------

% ------------------------------------------------------------------
% Angular elements in range 0-360 degrees
  function y=zero360(x)
% ---------------------

if x>=360
    x=x-fix(x/360)*360;
elseif x<0
    x=x-(fix(x/360)-1)*360;
end

y=x;

return
% ------------------------------------------------------------------
```

A.6 Stumpff Functions

```
% ------------------------------------------------------------------
% Stumpff function S(z)
  function S=StumpffS(z)
% ---------------------
% Input: z
```

```
% Output: S(z)
% --------------------------------------------------------------------

if z>0
    S=(sqrt(z)-sin(sqrt(z)))/(sqrt(z))^3; % ellipse
elseif z<0
    S=(sinh(sqrt(-z))-sqrt(-z))/(sqrt(-z))^3; % hyperbola
else
    S=1/6; % parabola
end
% --------------------------------------------------------------------

% --------------------------------------------------------------------
% Stumpff function C(z)
  function C=StumpffC(z)
% ----------------------
% Input: z
% Output: C(z)
% --------------------------------------------------------------------

if z>0
    C=(1-cos(sqrt(z)))/z; % ellipse
elseif z<0
    C=(cosh(sqrt(-z))-1)/(-z); % hypebola
else
    C=1/2; % parabola
end
% --------------------------------------------------------------------
```

A.7 Lambert's Problem

```
% --------------------------------------------------------------------
% Lambert Problem
  function [V1,V2]=Lambert(R1,R2,t)
% -------------------------------
% Input:
```

```
% Time of flight (t[year])
% Position Vectors (R1,R2[au])
% Output:
% Velocity vectors (V1,V2[au/year])
%
% Universal anomaly algorithm
% ------------------------------------------------------------------

% Declare global variables
global mu
global r1 r2 A

% Constant
mu=4*pi^2; % gravitational parameter [au^3/year^2]

% 1. Calculate r1,r2
r1=norm(R1); % initial position radius [au]
r2=norm(R2); % final position radius [au]

% 2. Calculate theta
c12=cross(R1,R2); % cross product of R1 into R2 [au]
theta=acos(dot(R1,R2)/(r1*r2)); % transfer angle [rad]
if c12(3)<0
    theta=2*pi-theta; %[rad]
end

% 3. Calculate A
A=sin(theta)*sqrt(r1*r2/(1-cos(theta)));

% 4. Iterate z,F(z),F'
% 4a. Determine approximate starting value for z
z=-100;
while F(z,t)<0
    z=z+0.1;
end
```

```
% 4b. Set an error tolerance and a limit on the number of iterations
tol=100*eps;
nmax=100;
% 4c. Iterate z
ratio=1;
n=0;
while (abs(ratio)>tol)&(n<=nmax)
    n=n+1;
    ratio=F(z,t)/dFdz(z);
    z=z-ratio;
end

% 5. Calculate y
y=y(z);

% 6. Calculate the Lagrange coefficients f,g,fdot,gdot
f=1-y/r1;
g=A*sqrt(y/mu);
fdot=sqrt(mu)/(r1*r2)*sqrt(y/C(z))*(z*S(z)-1);
gdot=1-y/r2;

% 7. Calculate V1,V2
V1=1/g*(R2-f*R1);
V2=1/g*(gdot*R2-R1);
return
% ------------------------------------------------------------------

% Subfunctions used in the main body:

% ------------------------------------------------------------------
% y(z)
  function y=y(z)
% ---------------

global r1 r2 A
y=r1+r2+A*(z*S(z)-1)/sqrt(C(z));
```

```
return
% ------------------------------------------------------------------


% ------------------------------------------------------------------
% F(z,t)
  function F=F(z,t)
% -----------------

global mu A
F=(y(z)/C(z))^1.5*S(z)+A*sqrt(y(z))-sqrt(mu)*t;
return
% ------------------------------------------------------------------


% ------------------------------------------------------------------
% dFdz(z)
  function dFdz=dFdz(z)
% ---------------------

global A
if z==0
    dFdz=sqrt(2)/40*y(0)^1.5+A/8*(sqrt(y(0))+A*sqrt(1/(2*y(0))));
else
    dFdz=(y(z)/C(z))^1.5*(1/(2*z)*(C(z)-(3/2)*S(z)/C(z))...
         +(3/4)*S(z)^2/C(z))...
         +A/8*(3*S(z)/C(z)*sqrt(y(z))+A*sqrt(C(z)/y(z)));
end
return
% ------------------------------------------------------------------


% ------------------------------------------------------------------
% C(z)
  function C=C(z)
% -------------

C=StumpffC(z);
return
```

```
% ------------------------------------------------------------------

% ------------------------------------------------------------------
% S(z)
function S=S(z)
% -------------

S=StumpffS(z);
return
% ------------------------------------------------------------------
```

A.8 Orbital Elements

```
% ------------------------------------------------------------------
% Orbital Elements
  function [a,T,OE]=OrbitalElements(R,V)
% -------------------------------------
% Input:
% State vector (R,V)
% Output:
% Semimajor axis (a[au])
% Period (T[year])
% Orbital elements (OE)
%   Angular momentum (h[au^2/year])
%   Eccentricity (e[-])
%   Right ascension of the ascending node (RA[rad])
%   Inclination (incl[rad])
%   Argument of perihelion (w[rad])
%   True anomaly (TA[rad])
% ------------------------------------------------------------------

% Declare global variable
global mu

% Constant
mu=4*pi^2; % gravitational parameter [au^3/year^2]
```

```
% 1. Calculate r
r=norm(R); % radius [au]

% 2. Calculate v
v=norm(V); % velocity [au/year]

% 3. Calculate vr
vr=dot(R,V)/r; % radial velocity [au/year]

% 4. Calculate H
H=cross(R,V); % angular momentum vector [au^2/year]

% 5. Calculate h
h=norm(H); % angular momentum [au^2/year]

% 6. Calculate incl
incl=acos(H(3)/h); % inclination [rad]

% 7. Calculate N
N=cross([0 0 1],H); % node line vector [au^2/year]

% 8. Calculate n
n=norm(N); % node line [au^2/year]

% 9. Calculate RA
if n>eps*10^6
    RA=acos(N(1)/n); % ascending node [rad]
    if N(2)<0
        RA=2*pi-RA;
    end
else
    RA=0;
end

% 10. Calculate E
```

```
E=cross(V,H)/mu-R/r; % eccentricity vector

% 11. Calculate e
e=norm(E); % eccentricity

% 12. Calculate w
if n>eps*10^6
    if e>eps*10^6
        w=acos(dot(N,E)/(n*e)); % argument of perihelion [rad]
        if E(3)<0
            w=2*pi-w;
        end
    else
        w=0;
    end
else
    if e>eps*10^6
        w=acos(dot([1 0 0],E)/(1*e)); % argument of perihelion [rad]
        if E(3)<0
            w=2*pi-w;
        end
        if E(3)==0
            if E(2)<0
                w=2*pi-w;
            end
        end
    else
        w=0;
    end
end

% 13. Calculate TA
if e>eps*10^6
    TA=acos(dot(E,R)/(e*r)); % true anomaly [rad]
    if vr<0
        TA=2*pi-TA;
```

```
    end
elseif n>eps*10^6
    TA=acos(dot(N,R)/(n*r));
    NR=cross(N,R);
    if NR(3)<0
        TA=2*pi-TA;
    end
else
    TA=acos(dot([1 0 0],R)/r);
    if R(2)<0
        TA=2*pi-TA;
    end
end

% 14. Calculate a,T
a=h^2/mu/(1-e^2); % semimajor axis [au]
if e<1
    T=2*pi*sqrt(a^3/mu); % period [year]
else
    T=Inf;
end

% Orbital elements OE
OE=[h;e;RA;incl;w;TA]; % orbital element vector
% ------------------------------------------------------------------
```

A.9 Planetary Flyby

```
% ------------------------------------------------------------------
% Planetary Flyby
  function [rm,hm,nud,dV]=Flyby(Vi,Vo,rp,mup)
% ----------------------------------------
% Input:
% Inbound velocity vector (Vi[km/s])
% Outbound velocity vector (Vi[km/s])
% Gravitational parameter of flyby planet (mup[km^3/s^2])
```

```
% Radius of flyby planet (rp[km])
% Output:
% Altitude at periapsis (hp[km])
% Turn angle (nud[deg])
% Delta-v at periapsis (dVPFM[km/s])
%
% Powered flyby maneuver if needed
% ------------------------------------------------------------------

% 1. Calculate inbound and outbound velocities vi,vo
vi=norm(Vi); %[km/s]
vo=norm(Vo); %[km/s]

% 2. Calculate turn angle nu
nu=asin(norm(cross(Vi,Vo))/(vi*vo)); %[rad]
nud=rad2deg(nu); % [rad]->[deg]

% Radius at periapsis
if nud<=1.0
    rm=Inf;
else
    % 3. Calculate lower and upper bounds rmin,rmax
    rmi=mup/vi^2*(csc(nu/2)-1);
    rmo=mup/vo^2*(csc(nu/2)-1);
    rmin=min(rmi,rmo);
    rmax=max(rmi,rmo);

    % 4. Calculate rm using fzero function
    f=@(x)asin(1/(1+vi^2/mup*x))+asin(1/(1+vo^2/mup*x))-nu;
    rm=fzero(f,[rmin,rmax]); %[km]
end

% 5. Calculate altitude at periapsis hm
hm=rm-rp; %[km]

% 6. Calculate delta-v required for powered flyby maneuver dV
```

```
dV=sqrt(vo^2+2*mup/rm)-sqrt(vi^2+2*mup/rm); %[km/s]
% ------------------------------------------------------------------
```

A.10 Direct Trajectory

```
% ------------------------------------------------------------------
% Direct Trajectory
  function [TOF,OE,Vinfd,Vinfa,C3d,C3a]=DirectTrajectory...
                                          (departure,arrival)
% ------------------------------------------------------------
% Input:
% Departure planet and date
% (departure=[planet1,year1,month1,day1,hour1])
% Arrival planet and date
% (arrival=[planet2,year2,month2,day2,hour2])
% Output:
% Time of flight (TOF[day])
% Orbital elements OE
%   Angular momentum (h[au^2/year])
%   Eccentricity (e[-])
%   Right ascension of the ascending node (RA[rad])
%   Inclination (incl[rad])
%   Argument of perigee (w[rad])
%   True anomaly (TA[rad])
%   Semimajor axis (a[au])
%   Period (T[year])
% Departure velocity vector (V1inf[km/s])
% Arrival velocity vector (V2inf[km/s])
% Departure C3 (C3d[km^2/s^2])
% Arrival C3 (C3a[km^2/s^2])
%
% Ballistic interplanetary trajectory from planet 1 to planet 2
% ------------------------------------------------------------------

% Declare global variable
global mu
```

```
% Constant
mu=4*pi^2; % gravitational parameter [au^3/year^2]
conversion=149597870.691/(365.25*24*60*60); % [au/year]->[km/s]
% 1au=149,597,870.691km
% 1year=365.25*24*60*60s

% Departure planet and date
planet1=departure(1);
year1=departure(2);
month1=departure(3);
day1=departure(4);
hour1=departure(5);

% Arrival planet and date
planet2=arrival(1);
year2=arrival(2);
month2=arrival(3);
day2=arrival(4);
hour2=arrival(5);

% 1. Calculate time of flight
TOF=FlightTime(year1,month1,day1,hour1,year2,month2,day2,hour2);
t=TOF/365.25; %[year]

% 2. Determine planets' state vector R1,V1,R2,V2
[R1,V1,OE1]=PlanetEphemeris(planet1,year1,month1,day1,hour1);
[R2,V2,OE2]=PlanetEphemeris(planet2,year2,month2,day2,hour2);

% 3. Determine spacecraft's position vector Rd,Ra
Rd=R1;
Ra=R2;

% 4. Solve Lambert's problem for Vd,Va
[Vd,Va]=Lambert(Rd,Ra,t);
```

```
% 5. Calculate orbital elements of transfer trajectory
[a,T,OEd]=OrbitalElements(Rd,Vd);
[a,T,OEa]=OrbitalElements(Ra,Va);
OE=[OEd;OEa(6);a;T];

% 6. Calculate Vinfd,Vinfa,C3d,C3a
Vinfd=(Vd-V1)*conversion; % outbound velocity vector at departure
Vinfa=(Va-V2)*conversion; % inbound velocity vector at arrival
vinfd=norm(Vinfd); % outbound velocity at departure [km/s]
vinfa=norm(Vinfa); % inbound velocity at arrival [km/s]
C3d=vinfd^2; % departure C3 [km^2/s^2]
C3a=vinfa^2; % arrival C3 [km^2/s^2]
% ------------------------------------------------------------------
```

A.11 Flyby Trajectory

```
% ------------------------------------------------------------------
% Interplanetary Flyby Trajectory
  function [TOF,OE12,OE23,Vinfd,Vinfa,C3d,C3a,hm,dVPFM]=...
           FlybyTrajectory(departure,encounter,arrival)
% ------------------------------------------------------------
% Input:
% Departure planet and date
% (departure=[planet1,year1,month1,day1,hour1])
% Encounter planet and date
% encounter=[planet2,year2,month2,day2,hour2]
% Arrival planet and date
% (arrival=[planet3,year3,month3,day3,hour3])
% Output:
% Time of flight (TOF[day])
% Orbital elements OE12,OE23
%   Angular momentum (h[au^2/year])
%   Eccentricity (e[-])
%   Right ascension of the ascending node (RA[rad])
%   Inclination (incl[rad])
%   Argument of perigee (w[rad])
```

```
%   True anomaly (TA[rad])
%   Semimajor axis (a[au])
%   Period (T[year])
% Departure velocity vector (V1inf[km/s])
% Arrival velocity vector (V2inf[km/s])
% Departure C3 (C3d[km^2/s^2])
% Arrival C3 (C3a[km^2/s^2])
% Minimum passing altitude (hm[km])
% Flyby delta-v (dVPFM[km/s])
%
% Interplanetary trajectory from planet 1 to planet 3 via planet 2
% ------------------------------------------------------------------

% Declare global variable
global mu

% Constant
mu=4*pi^2; % gravitational parameter [au^3/year^2]
conversion=149597870.691/(365.25*24*60*60); % [au/year]->[km/s]
% 1au=149,597,870.691km
% 1year=365.25*24*60*60s

% Departure planet and date
planet1=departure(1);
year1=departure(2);
month1=departure(3);
day1=departure(4);
hour1=departure(5);

% Encounter planet and date
planet2=encounter(1);
year2=encounter(2);
month2=encounter(3);
day2=encounter(4);
hour2=encounter(5);
```

```
% Arrival planet and date
planet3=arrival(1);
year3=arrival(2);
month3=arrival(3);
day3=arrival(4);
hour3=arrival(5);

% 1. Calculate time of flight
TOF12=FlightTime(year1,month1,day1,hour1,year2,month2,day2,hour2);
TOF23=FlightTime(year2,month2,day2,hour2,year3,month3,day3,hour3);
TOF=FlightTime(year1,month1,day1,hour1,year3,month3,day3,hour3);
t12=TOF12/365.25; %[year]
t23=TOF23/365.25; %[year]

% 2. Determine planets' state vector R1,V1,R2,V2,R3,V3
[R1,V1,OE1]=PlanetEphemeris(planet1,year1,month1,day1,hour1);
[R2,V2,OE2]=PlanetEphemeris(planet2,year2,month2,day2,hour2);
[R3,V3,OE3]=PlanetEphemeris(planet3,year3,month3,day3,hour3);

% 3. Determine spacecraft's position vector Rd,Ra
Rd=R1;
Re=R2;
Ra=R3;

% 4. Solve Lambert's problem for Vd,Ve12
if t12>=0
    [Vd,Ve12]=Lambert(Rd,Re,t12);
else
    Vd=[Inf;Inf;Inf];
    Ve12=[Inf;Inf;Inf];
end

% 5. Solve Lambert's problem for Ve23,Va
if t23>=0
    [Ve23,Va]=Lambert(Re,Ra,t23);
else
```

```
    Ve23=[Inf;Inf;Inf];
    Va=[Inf;Inf;Inf];
end

% 6. Calculate orbital elements of transfer trajectory
% Orbital elements of transfer trajectory 12
[a12,T12,OE12d]=OrbitalElements(Rd,Vd);
[a12,T12,OE12e]=OrbitalElements(Re,Ve12);
OE12=[OE12d;OE12e(6);a12;T12];
% Orbital elements of transfer trajectory 23
[a23,T23,OE23e]=OrbitalElements(Re,Ve23);
[a23,T23,OE23a]=OrbitalElements(Ra,Va);
OE23=[OE23e;OE23a(6);a23;T23];

% 7. Calculate Vinfd,Vinfa,C3d,C3a
Vinfd=(Vd-V1)*conversion; % outbound velocity vector at departure
Vinfa=(Va-V3)*conversion; % inbound velocity vector at arrival
vinfd=norm(Vinfd); % outbound velocity at departure [km/s]
vinfa=norm(Vinfa); % inbound velocity at arrival [km/s]
C3d=vinfd^2; % departure C3 [km^2/s^2]
C3a=vinfa^2; % arrival C3 [km^2/s^2]

% 8. Obtain physical data of planet 2
[rp,mup]=PlanetaryPhysicalData(planet2);

% 9. Calculate Vi,Vo
Vi=(Ve12-V2)*conversion;
Vo=(Ve23-V2)*conversion;

% 10. Calculate hm,dVPFM
[rm,hm,nud,dVPFM]=Flyby(Vi,Vo,rp,mup);
% ------------------------------------------------------------------

% Subfunctions used in the main body:

% ------------------------------------------------------------------
```

```
% Planetary Physical Data
  function [rp,mup]=PlanetaryPhysicalData(planet_id)
% ------------------------------------------------

% rp[km],mup[km^3/s^2]
PPD=[ 2440      22032.1; % Mercury
      6052     324858.8; % Venus
      6378     398600.4; % Earth
      3396      42828.3; % Mars
     71490  126711995.4; % Jupiter
     60270   37939519.7; % Saturn
     25560    5780158.5; % Uranus
     24760    6871307.8; % Neptune
      1195       1020.9]; % Pluto
rp=PPD(planet_id,1); %[km]
mup=PPD(planet_id,2); %[km^3/s^2]
return
% ------------------------------------------------------------------
```

Appendix B

Effect of Inclination and Eccentricity

This appendix discusses the effect of inclination and eccentricity of the planetary orbits on the C3 contours. To visualize the effect, we produced the hypothetical Earth-Mars C3 contours with the assumption of coplanar, circular orbits and compared them with those for the real case. In this assumption, the planetary orbits are perfectly coplanar in the ecliptic plane and perfectly circular with radii of the planets' semimajor axes. This can be achieved by just fixing $i = 0$ and $e = 0$ in Algorithm 3.3 (Planetary Ephemeris).

Figures B-1 and B-2 show the hypothetical Earth-Mars C3 contours for departure and arrival, respectively. For comparison, the actual C3 contours are shown in Figures B-3 and B-4. These figures show the period 2020-2023, which would be enough for the discussion here. The datatips are displayed at each local minimum C3 (floor of each crater) in these figures. We can see that there are two major differences between the real case and the hypothetical case: one is that for each mission window, the two craters are combined into one big crater in the hypothetical case, and the other is that the two contour patterns in 2020 and 2022 are identical in the hypothetical case while they are slightly different in the real case.

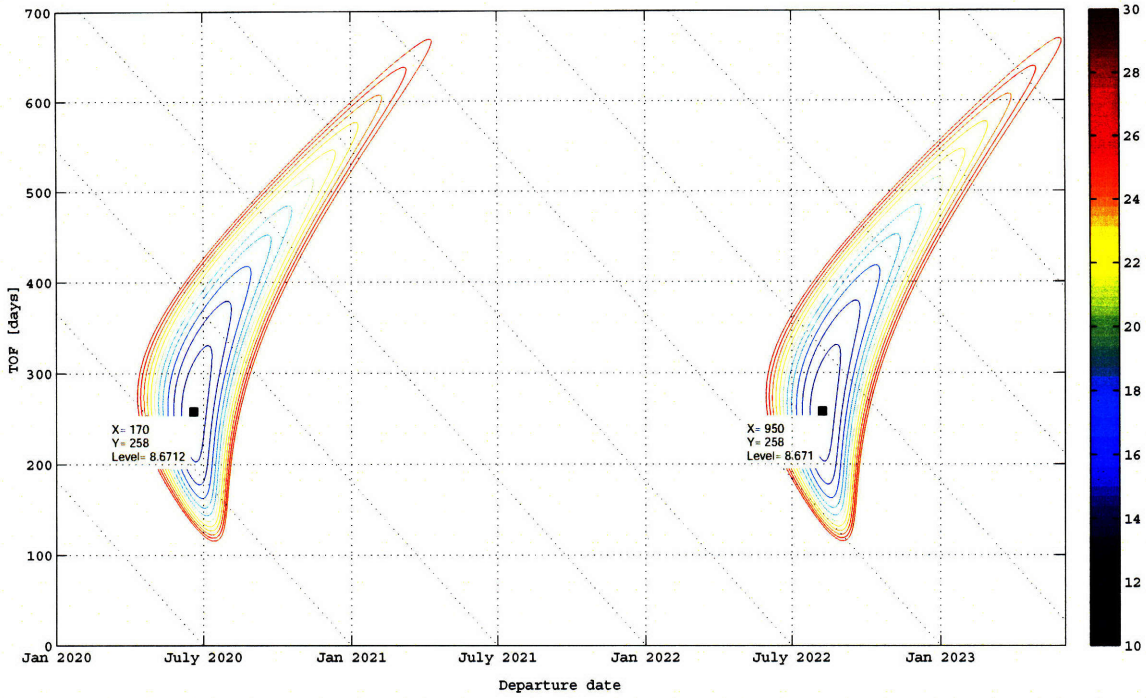


Figure B-1: $C3_d$ contours: 2020-2023 Earth-Mars (coplanar circular)

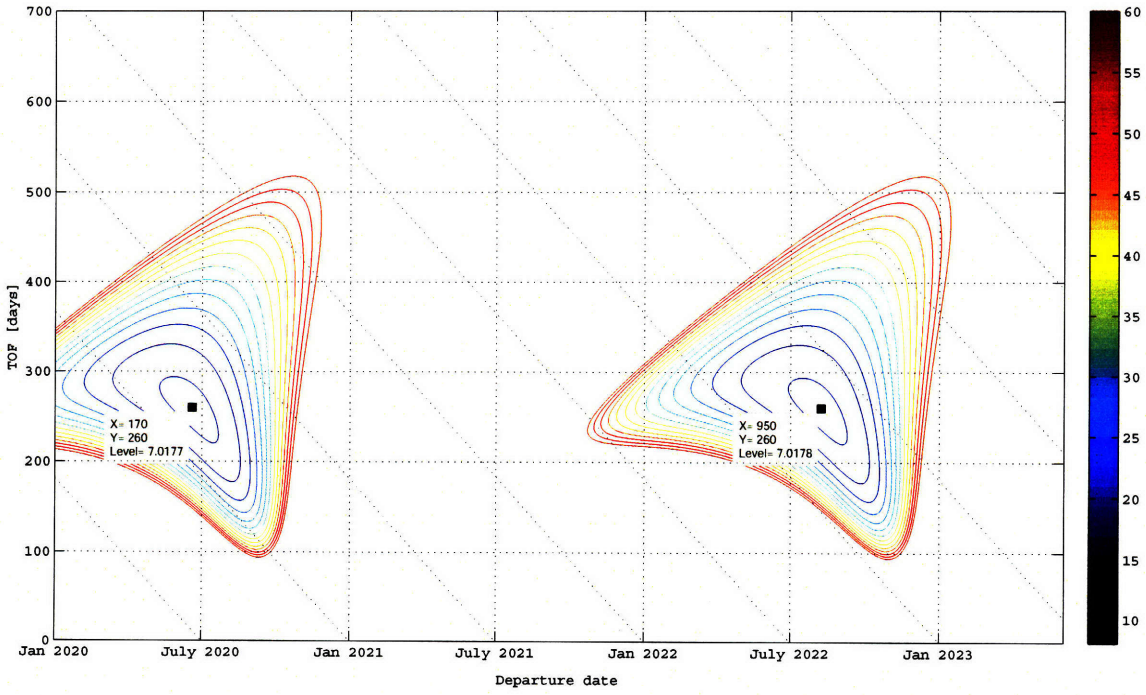


Figure B-2: $C3_a$ contours: 2020-2023 Earth-Mars (coplanar circular)

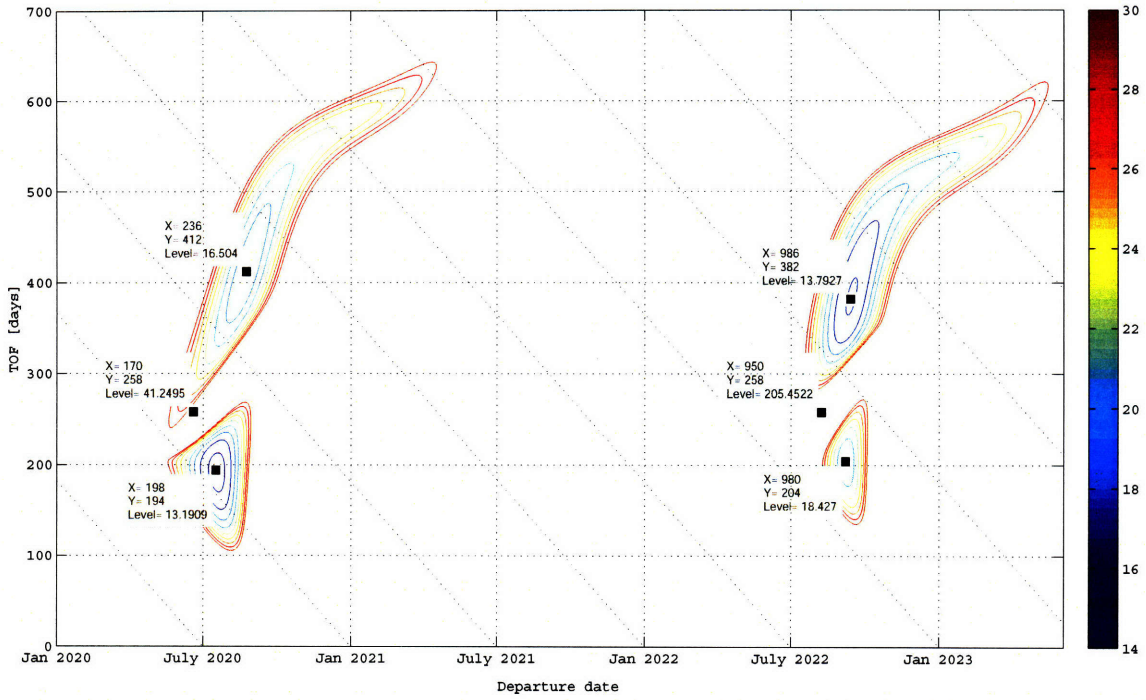


Figure B-3: $C3_d$ contours: 2020-2023 Earth-Mars

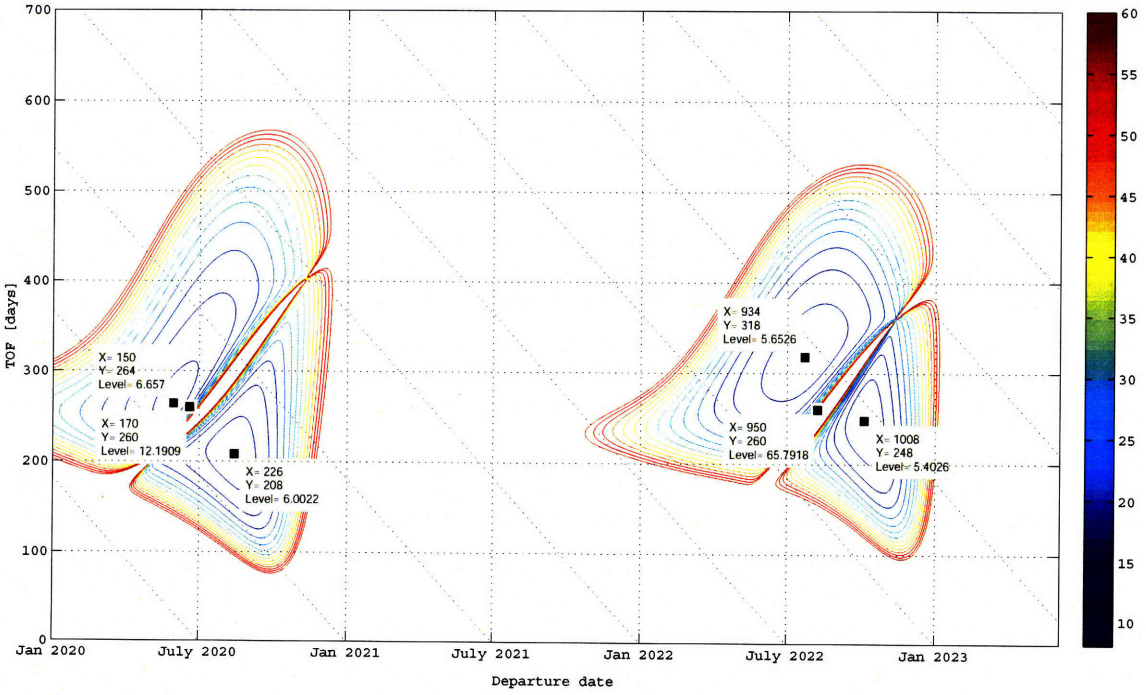


Figure B-4: $C3_a$ contours: 2020-2023 Earth-Mars

B.1 Two Craters and Mountain Wall

For the real case, as shown in Figures B-3 and B-4, there are two regions of local minimum C3 for each mission window. For a spacecraft moving solely under the influence of solar gravity, the trajectory plane must include the positions of the Earth at departure and Mars at arrival, and the Sun as the center of attraction. If the departure and arrival positions are nearly 180° apart, as measured with the Sun at the vertex, then the trajectory plane can and generally will be inclined at a large angle to the ecliptic plane. Thus, as stated in Section 3.3.2, we would say that the three-dimensional effect of inclination causes the "mountain wall" between the two craters since highly inclined transfer orbit causes high relative velocities at both departure and arrival. For the coplanar case, however, since the Earth and Mars always lie in the ecliptic plane, the orbit of the spacecraft, which must include the Earth at departure, Mars at arrival, and Sun, also lies in the ecliptic plane. Therefore, the C3 contours have only one minimum C3 in each mission window.

The transfer orbit for this minimum C3 is shown in Figure B-5. This orbit looks like a Hohmann transfer orbit, which is an ellipse, with the Sun at one focus, whose perihelion and aphelion are the points of tangency with the Earth and Mars orbits, respectively. In general, the Hohmann transfer orbit is known to require the least expenditure of fuel for the transfer if the planetary orbits were coplanar circles. The ideal minimum C3 can be theoretically calculated as

$$C3_d = \Delta v_d^2 = \left[\sqrt{\frac{\mu_\odot}{r_\oplus}}\left(\sqrt{\frac{2r_{\sigma'}}{r_\oplus + r_{\sigma'}}} - 1\right)\right]^2 = 8.6707 \text{ [km}^2/\text{s}^2\text{]} \tag{B.1a}$$

$$C3_a = \Delta v_a^2 = \left[\sqrt{\frac{\mu_\odot}{r_{\sigma'}}}\left(1 - \sqrt{\frac{2r_\oplus}{r_\oplus + r_{\sigma'}}}\right)\right]^2 = 7.0163 \text{ [km}^2/\text{s}^2\text{]} \tag{B.1b}$$

where

$$\begin{aligned} \mu_\odot &= 4\pi^2 \text{ [au}^3/\text{year}^2\text{]} \\ r_\oplus &= 1.0000 \text{ [au]} \\ r_{\sigma'} &= 1.5236 \text{ [au]} \end{aligned} \tag{B.2}$$

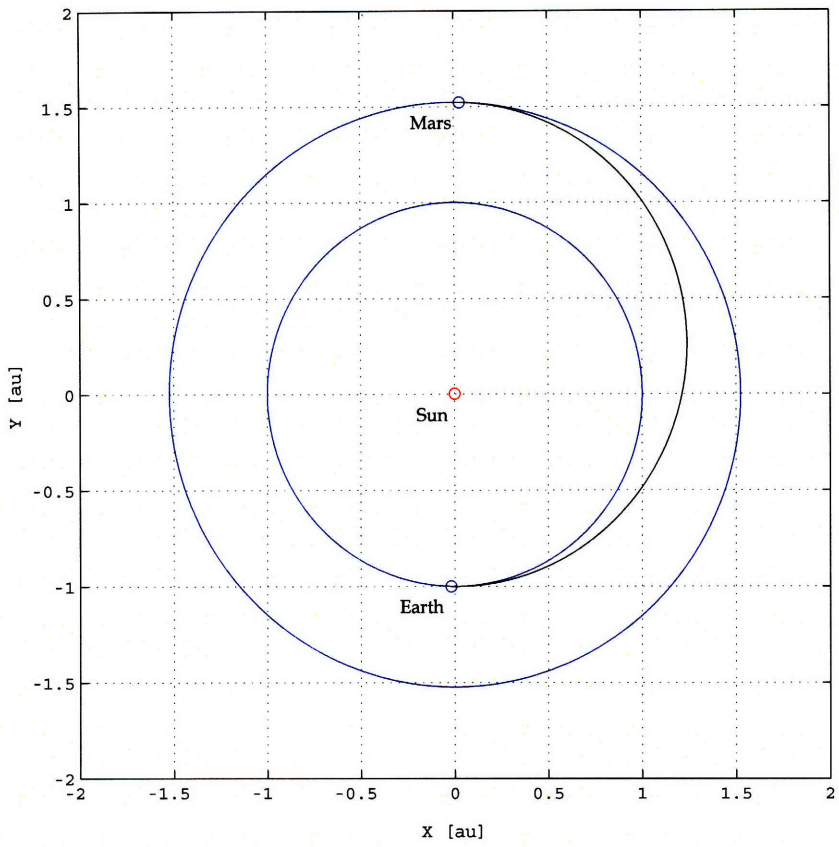
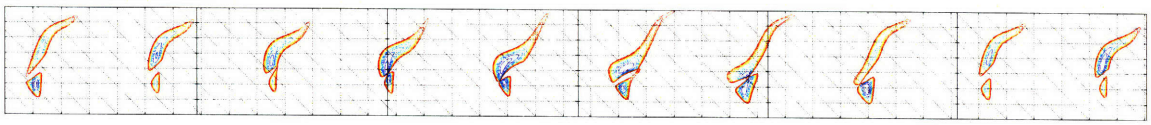


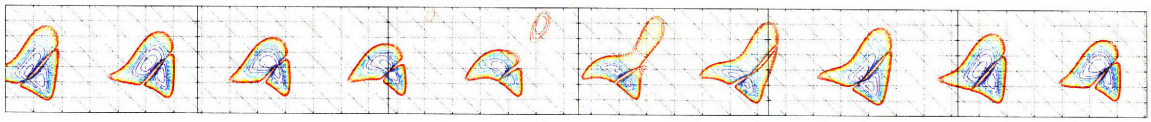
Figure B-5: Minimum C3 trajectory: 2020 Earth-Mars (coplanar circular)

The C3 values in Figures B-1 and B-2 almost agree with those in Eqs. (B.1), which implies that the transfer orbit in Figure B-5 is virtually a Hohmann transfer orbit.

However, the orbits of the planets are not coplanar, and although the angle between the orbital planes of the Earth and Mars is only 1.85°, the effect on C3 is not a minor one; a big crater for each mission window is split into two craters of local minimum C3 by the "mountain wall". Therefore, the coplanar assumption is too simplistic to predict exactly when the actual timing of preferable departure and arrival will be.



(a) Departure C3 at Earth



(b) Arrival C3 at Mars

Figure B-6: C3 contours: Earth-Mars direct 2020-2040

B.2 Periodic Variation in Contour Pattern

The two consecutive mission windows in Figures B-1 and B-2 are identical in shape of the contour lines and depth of minimum C3 points. Obviously we can infer that the contour pattern would continue to duplicate itself every launch window since the Earth and Mars orbits are perfectly coplanar and circular. In Figure B-6, on the other hand, the effect of eccentricity can be observed in the sequential contour pattern slightly changing per launch window cycle. Although the same Earth-Mars phase angle recurs every synodic period of approximately 2.136 years (780 days), the mutual planetary configurations are slightly different due to the eccentricity. Almost the same Sun-Earth-Mars configuration in absolute space is achieved when multiples of the synodic period are close to an integer number of years. As shown in Figure B-6, the gradual change of the sequential contour pattern appears to be repeated almost every seven to eight mission windows (approximately 15-17 years). Thus, we would say that the eccentricity of the planets primarily induces the periodic variation in the sequential contour pattern.

Appendix C

Direct Trajectories between Earth, Venus, and Mars

For reference, this appendix presents $C3_d$ contours for Earth-Venus and Mars-Venus direct trajectories and $C3_a$ contours for Venus-Mars and Venus-Earth direct trajectories, which are not covered in Chapter 4.

C.1 Earth-Venus $C3_d$ Contours and Venus-Mars $C3_a$ Contours

Figures C-1 through C-12 show the Earth-Venus $C3_d$ contours and Venus-Mars $C3_a$ contours for Period 1-6 in (4.2), respectively. These contours display $C3_d$ up to 30 km^2/s^2 and $C3_a$ up to 60 km^2/s^2. The data tips display the values of each local minimum. We used these data to filter Earth-Venus-Mars flyby opportunities for the creation of integrated ΔV_{tot} contours in Section 4.3.2.

In the Earth-Venus $C3_d$ contours, the intervals between x-coordinates are about 583 days on average, which agrees with the Earth-Venus synodic period, 583.92 days.

In the Venus-Mars $C3_a$ contours, the intervals between x-coordinates are about 333 days on average, which agrees with the Venus-Mars synodic period, 333.92 days.

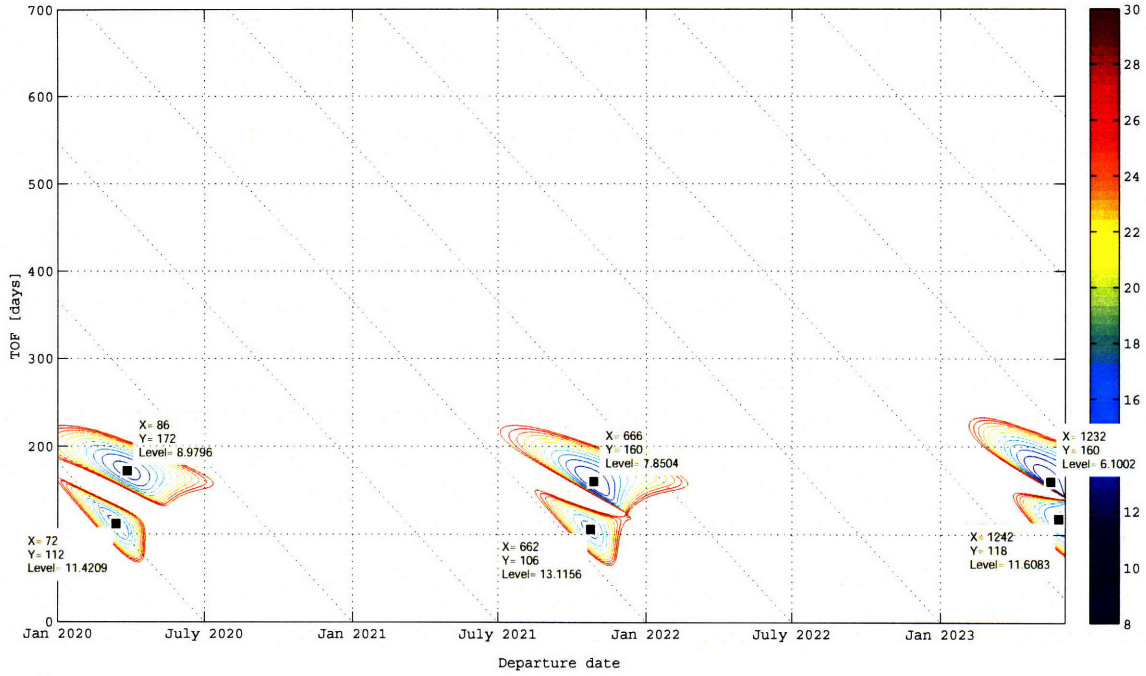


Figure C-1: $C3_d$ contours: Earth-Venus direct Period 1

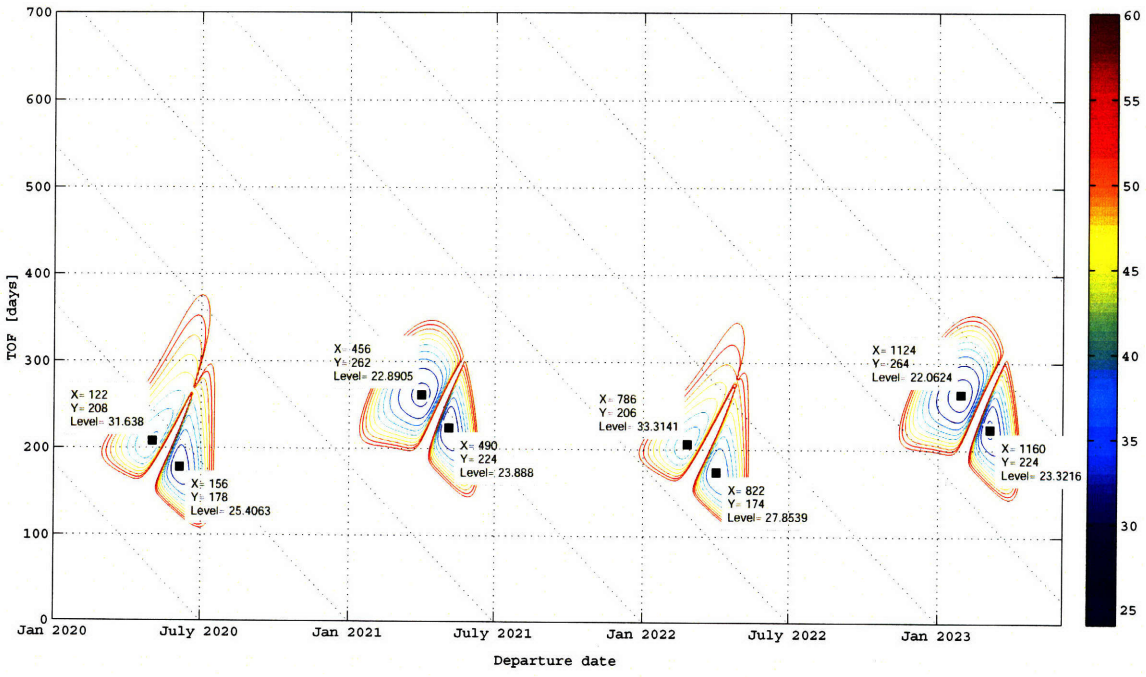


Figure C-2: $C3_a$ contours: Venus-Mars direct Period 1

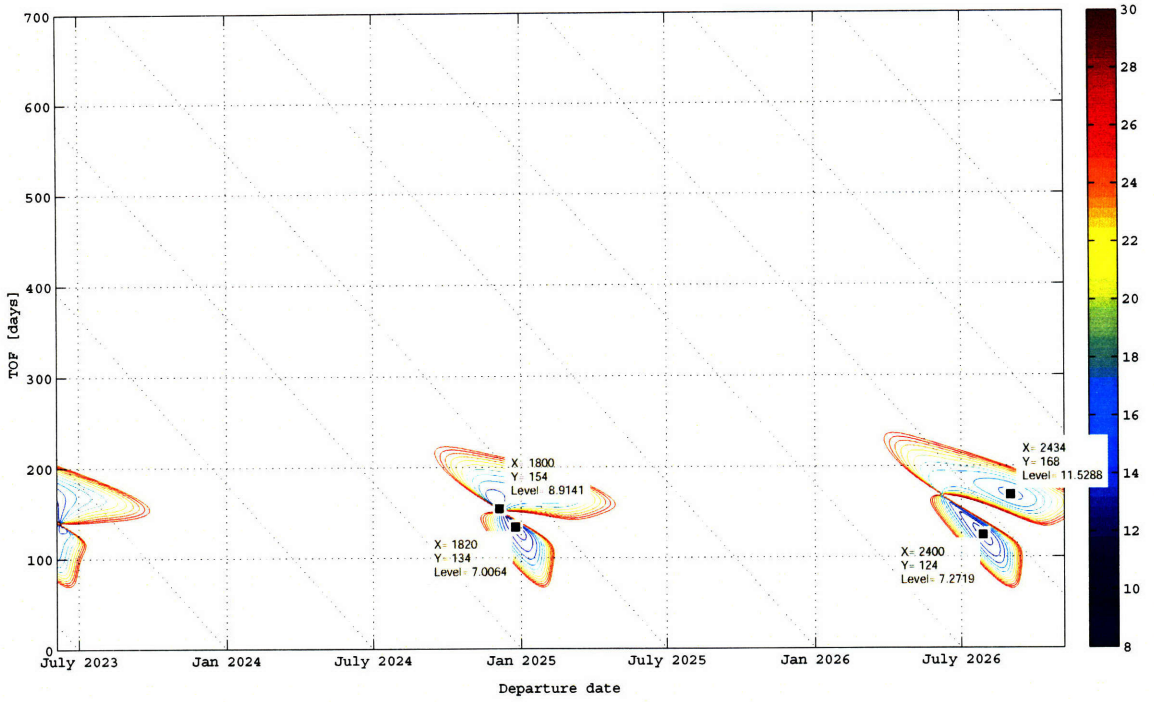


Figure C-3: $C3_d$ contours: Earth-Venus direct Period 2

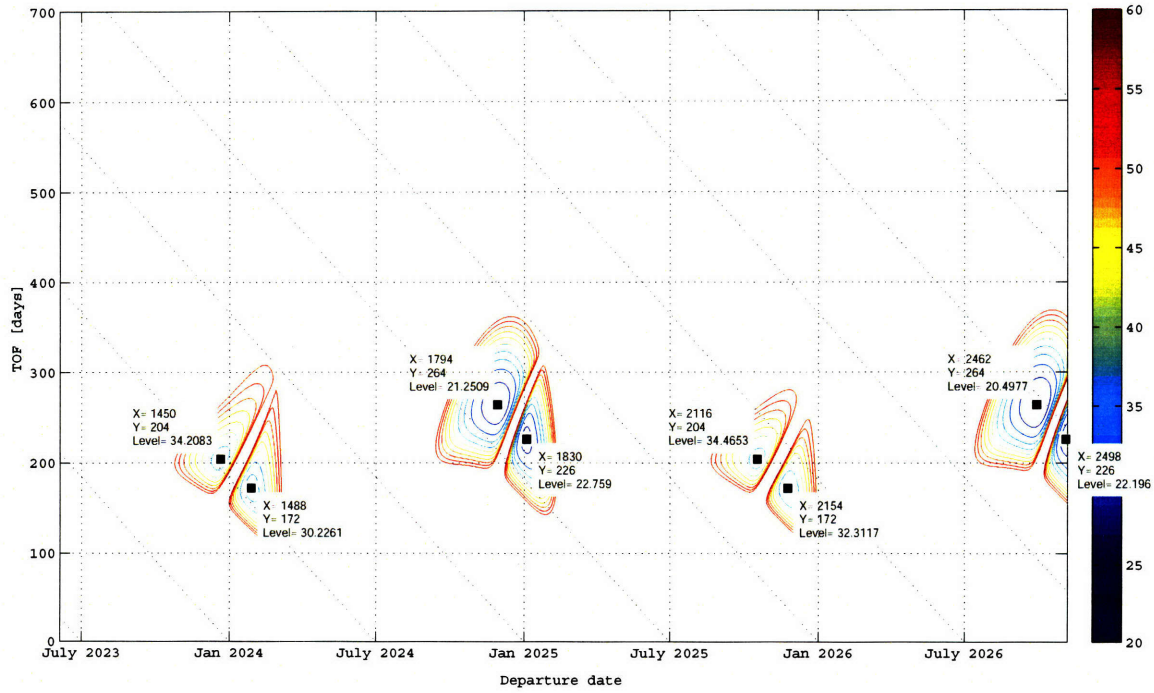


Figure C-4: $C3_a$ contours: Venus-Mars direct Period 2

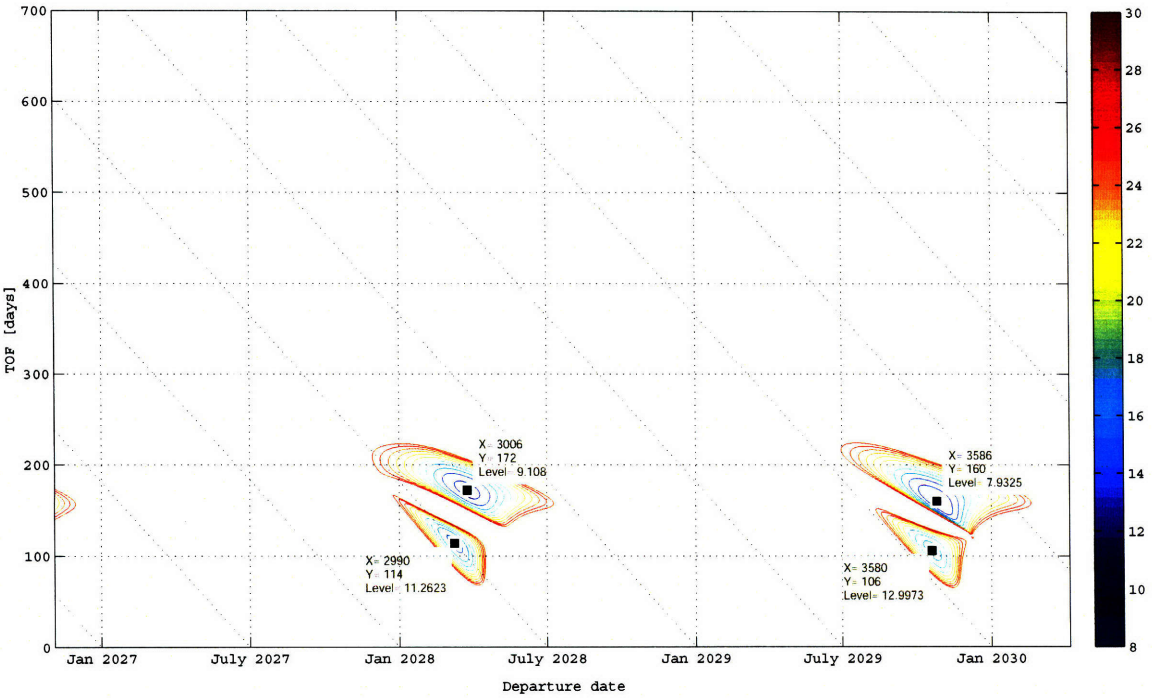


Figure C-5: $C3_d$ contours: Earth-Venus direct Period 3

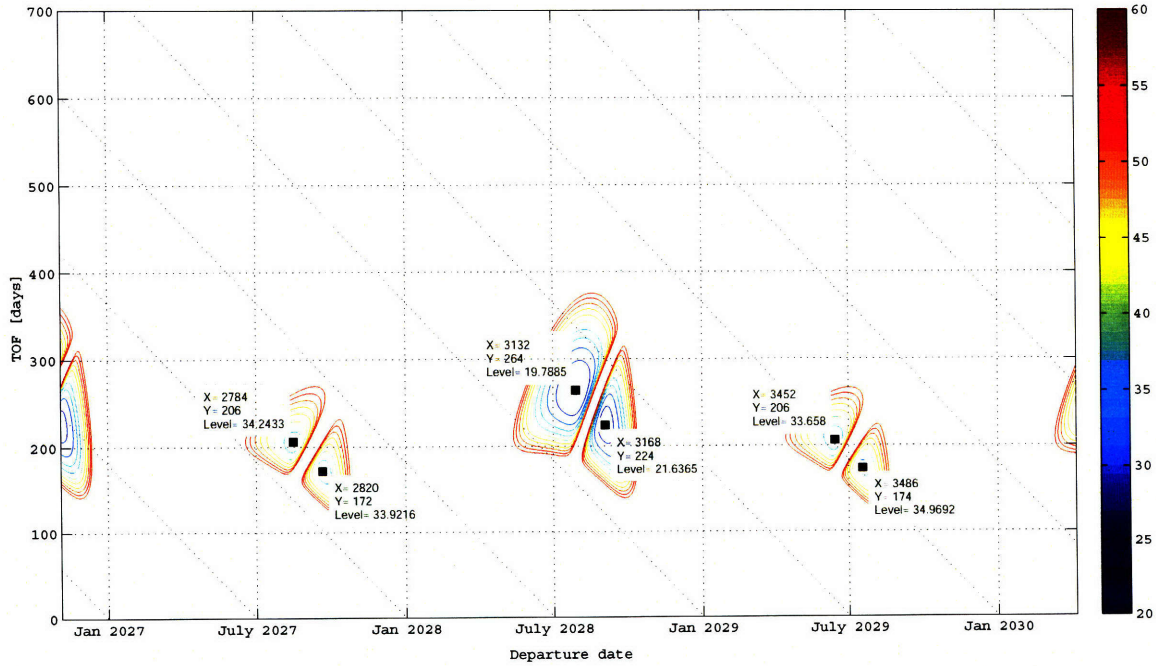


Figure C-6: $C3_a$ contours: Venus-Mars direct Period 3

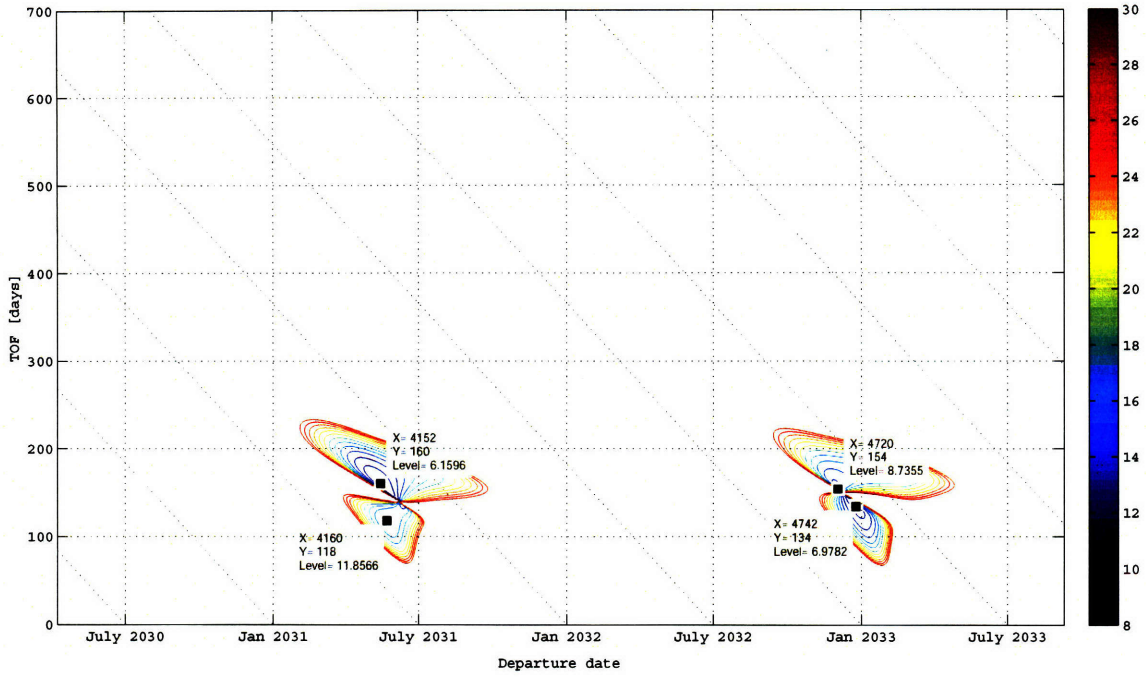


Figure C-7: $C3_d$ contours: Earth-Venus direct Period 4

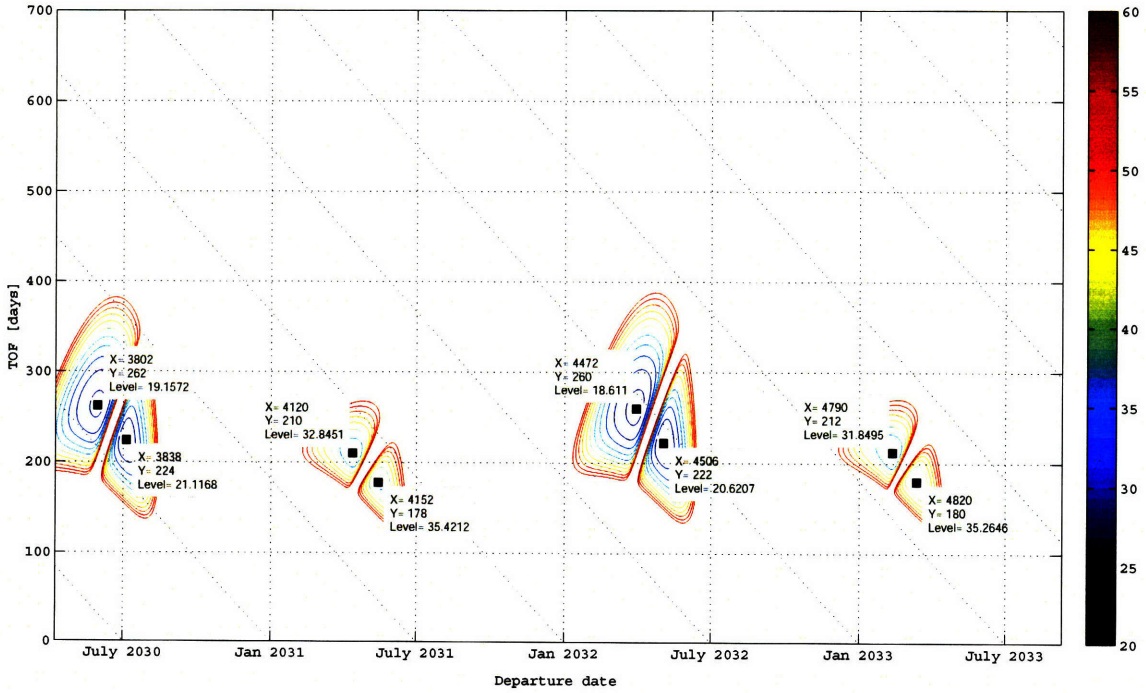


Figure C-8: $C3_a$ contours: Venus-Mars direct Period 4

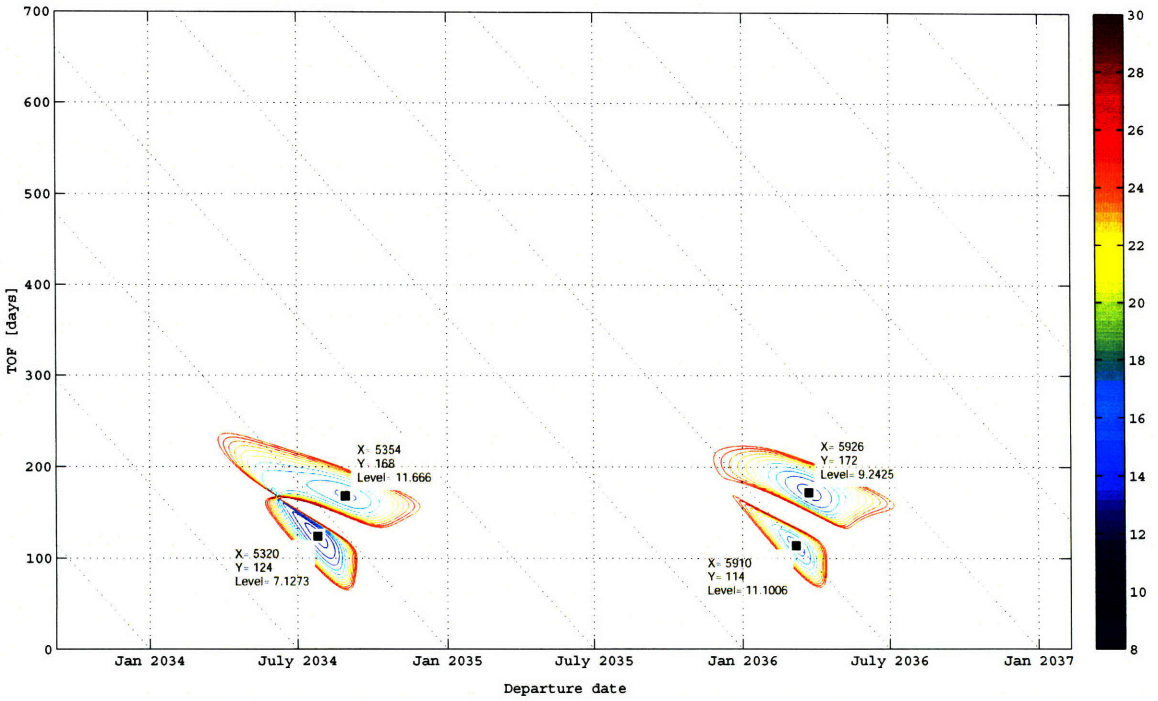


Figure C-9: $C3_d$ contours: Earth-Venus direct Period 5

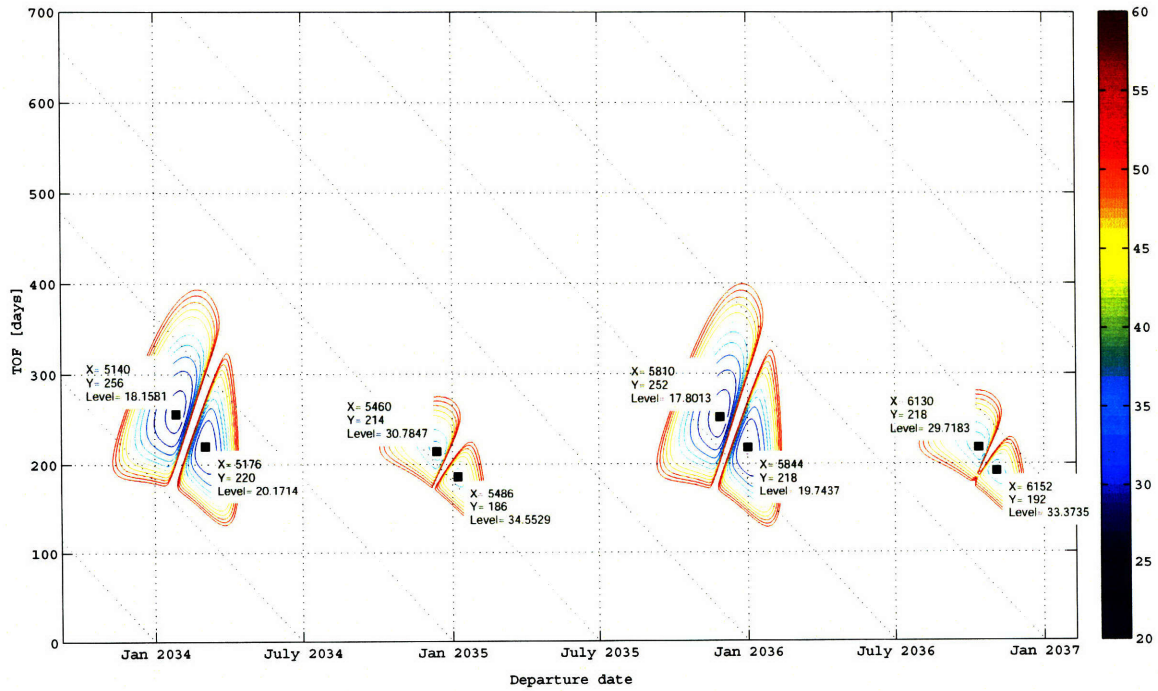


Figure C-10: $C3_a$ contours: Venus-Mars direct Period 5

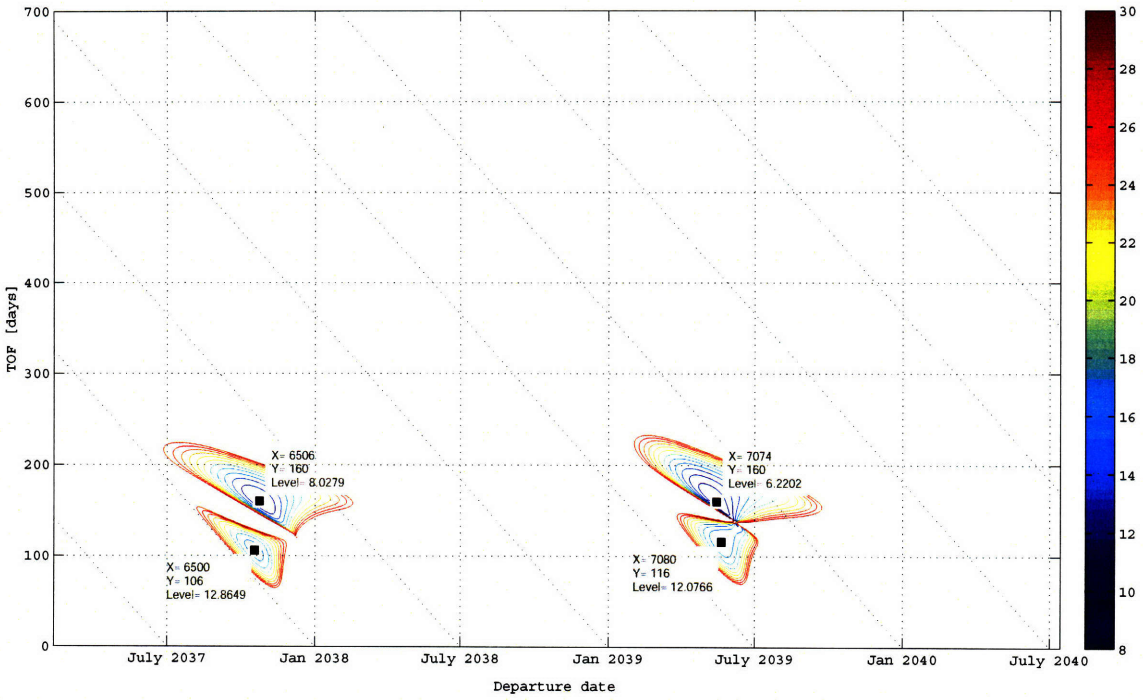


Figure C-11: $C3_d$ contours: Earth-Venus direct Period 6

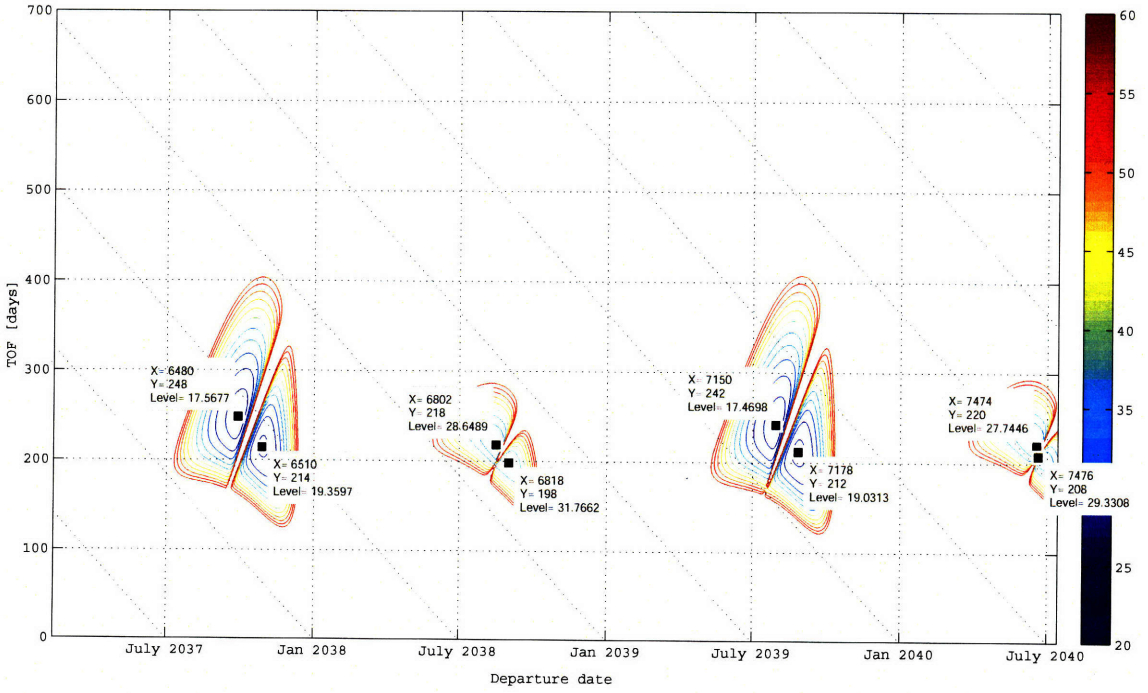


Figure C-12: $C3_a$ contours: Venus-Mars direct Period 6

C.2 Mars-Venus $C3_d$ Contours and Venus-Earth $C3_a$ Contours

Figures C-13 through C-24 show the Mars-Venus $C3_d$ contours and Venus-Earth $C3_a$ contours for Period 1-6 in (4.2), respectively. These contours display $C3_d$ up to 30 km^2/s^2 and $C3_a$ up to 60 km^2/s^2. The data tips display the values of each local minimum. We used these data to filter Mars-Venus-Earth flyby opportunities for the creation of integrated ΔV_{tot} contours in Section 4.4.2.

In the Mars-Venus $C3_d$ contours, the intervals between x-coordinates are about 331 days on average, which agrees with the Earth-Venus synodic period, 333.92 days.

In the Venus-Earth $C3_a$ contours, the intervals between x-coordinates are about 583 days on average, which agrees with the Venus-Mars synodic period, 583.92 days.

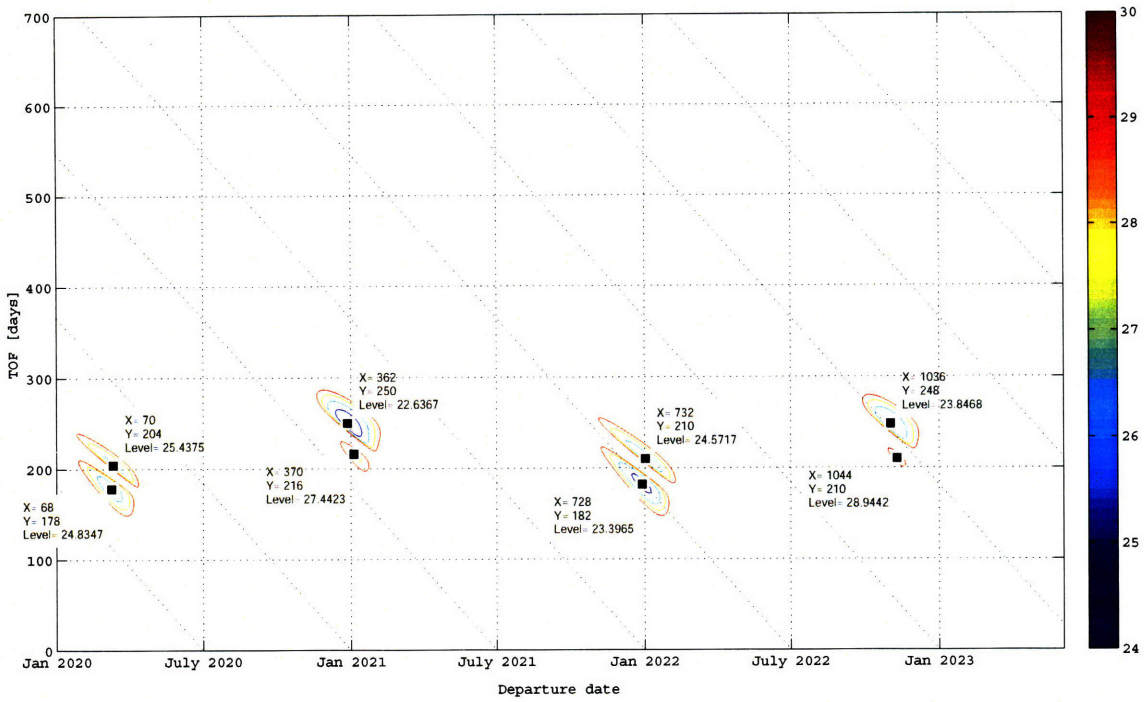


Figure C-13: $C3_d$ contours: Mars-Venus direct Period 1

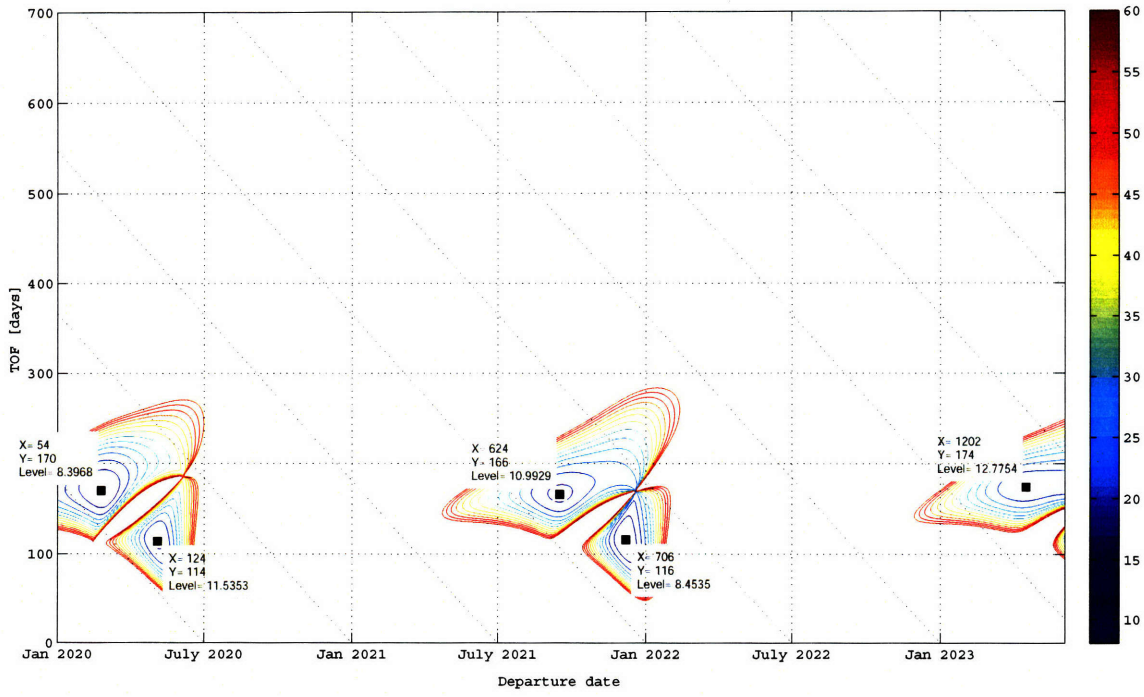


Figure C-14: $C3_a$ contours: Venus-Earth direct Period 1

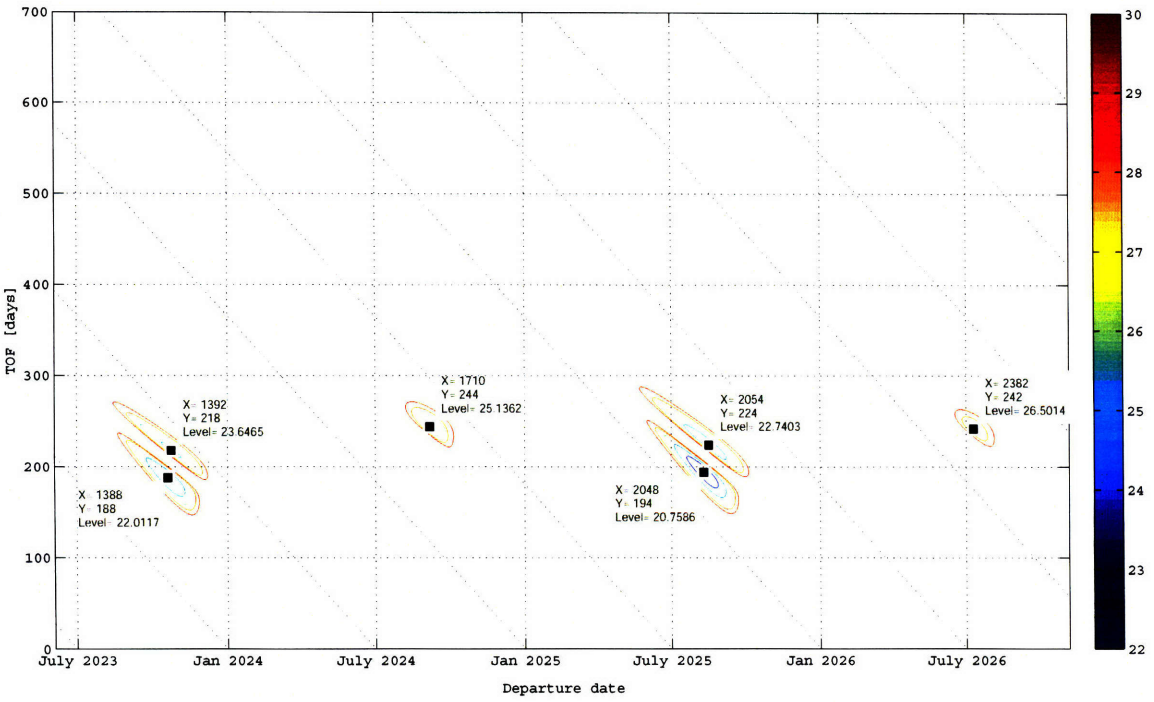


Figure C-15: $C3_d$ contours: Mars-Venus direct Period 2

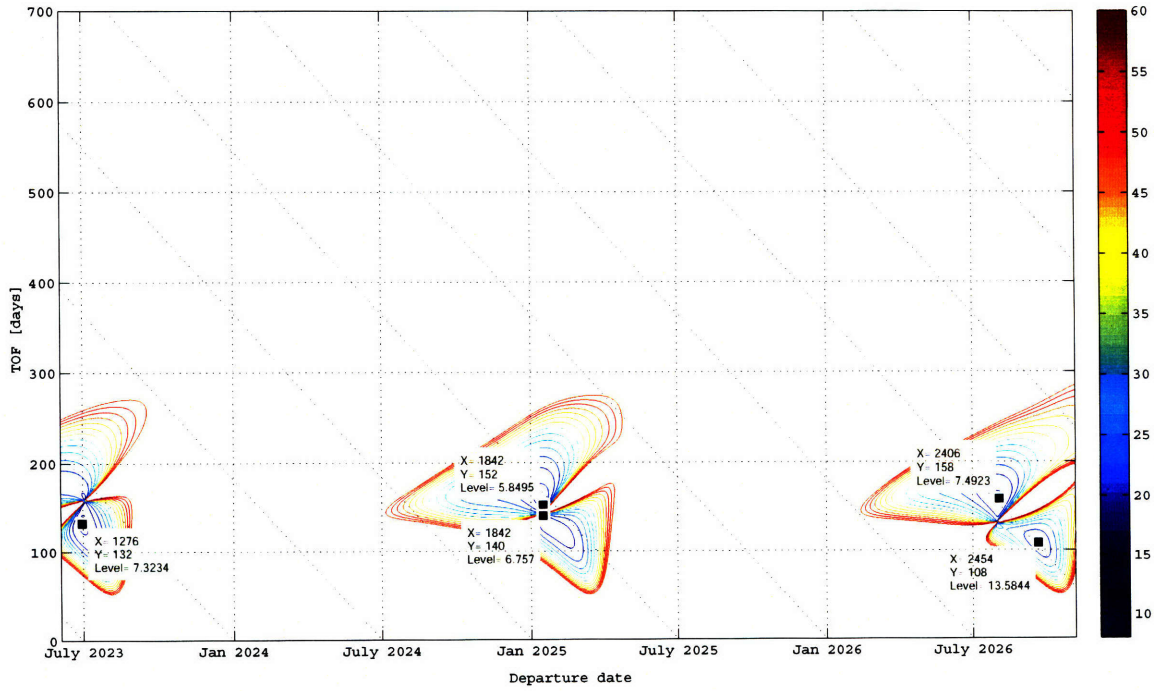


Figure C-16: $C3_a$ contours: Venus-Earth direct Period 2

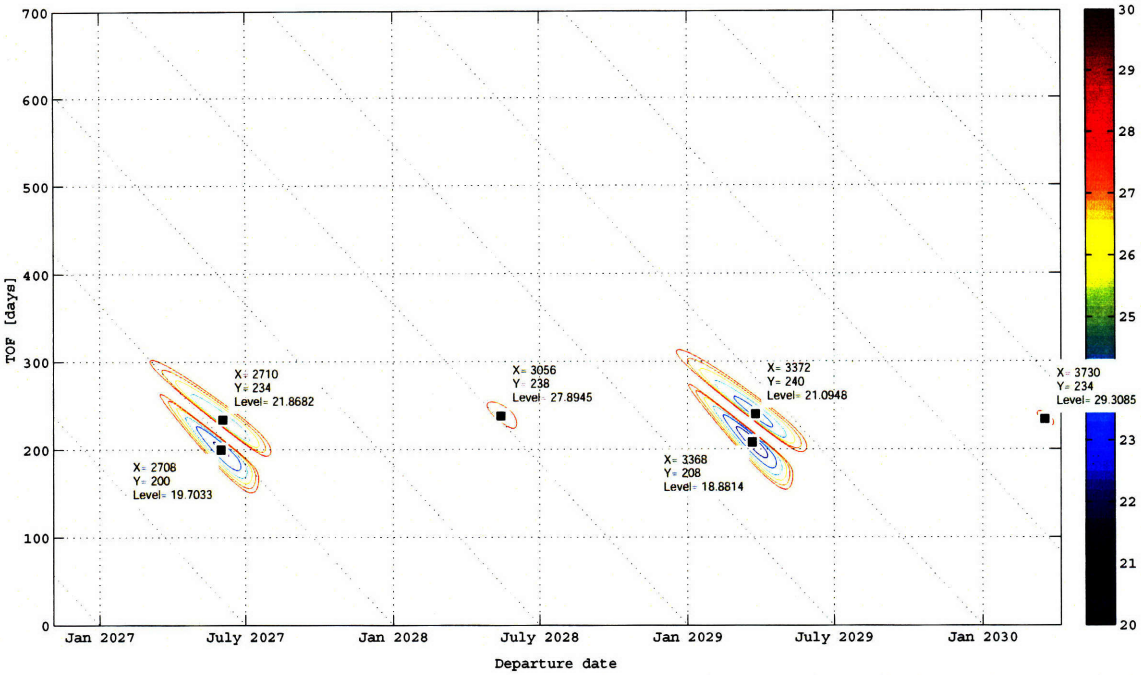


Figure C-17: $C3_d$ contours: Mars-Venus direct Period 3

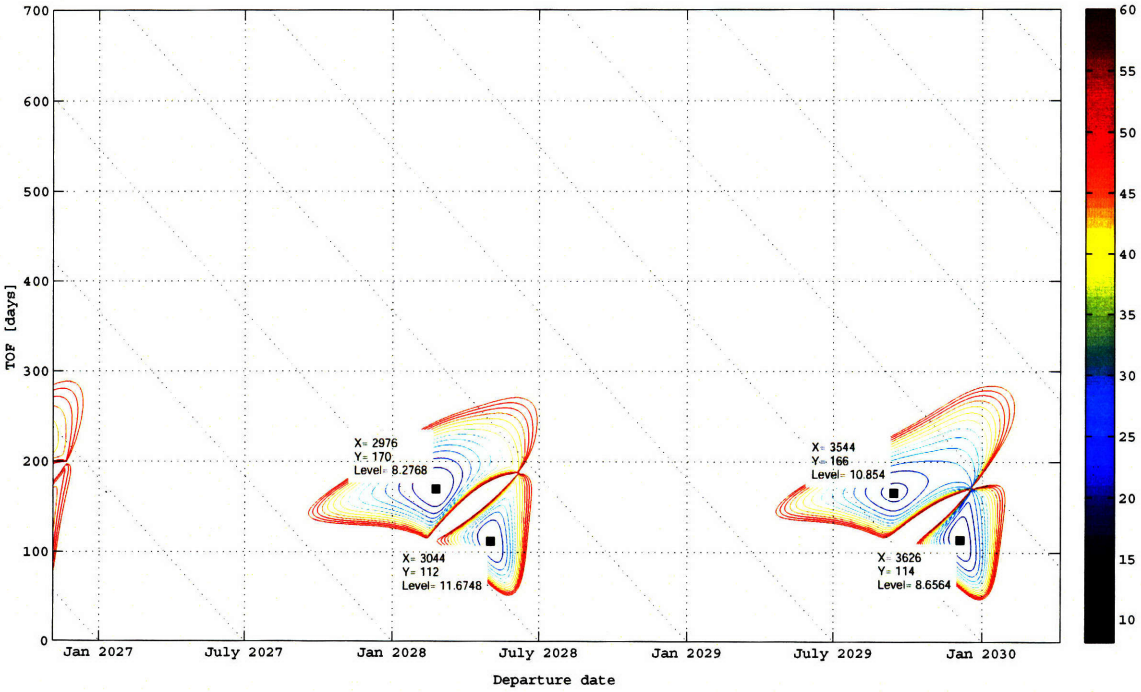


Figure C-18: $C3_a$ contours: Venus-Earth direct Period 3

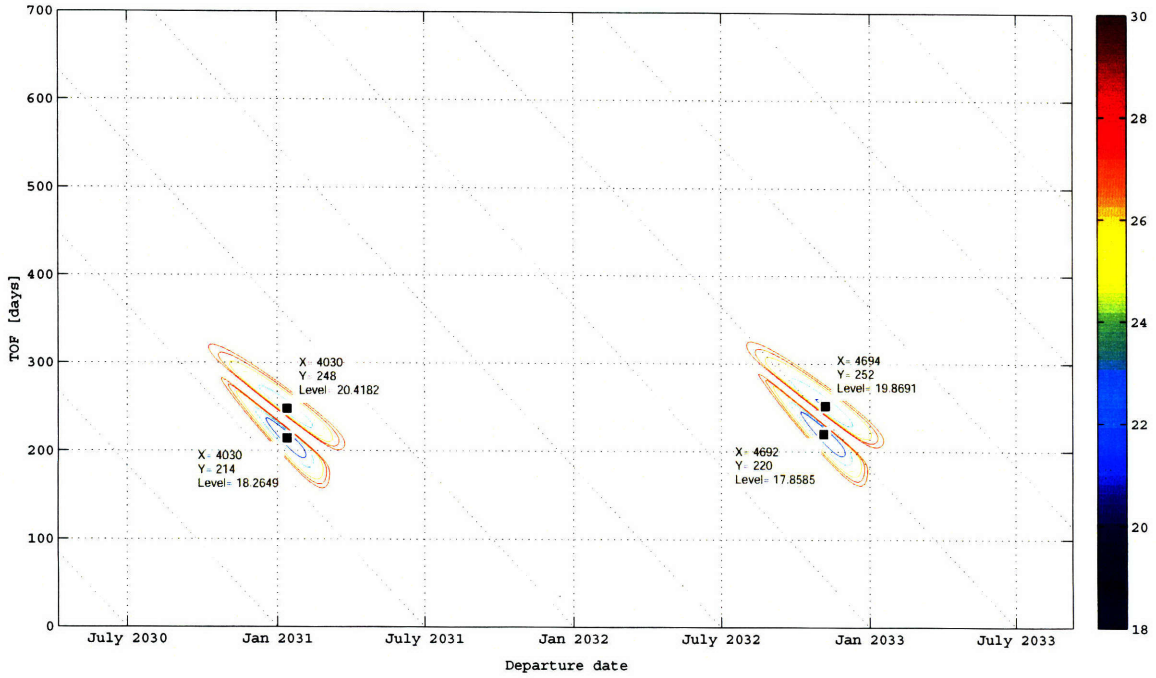


Figure C-19: $C3_d$ contours: Mars-Venus direct Period 4

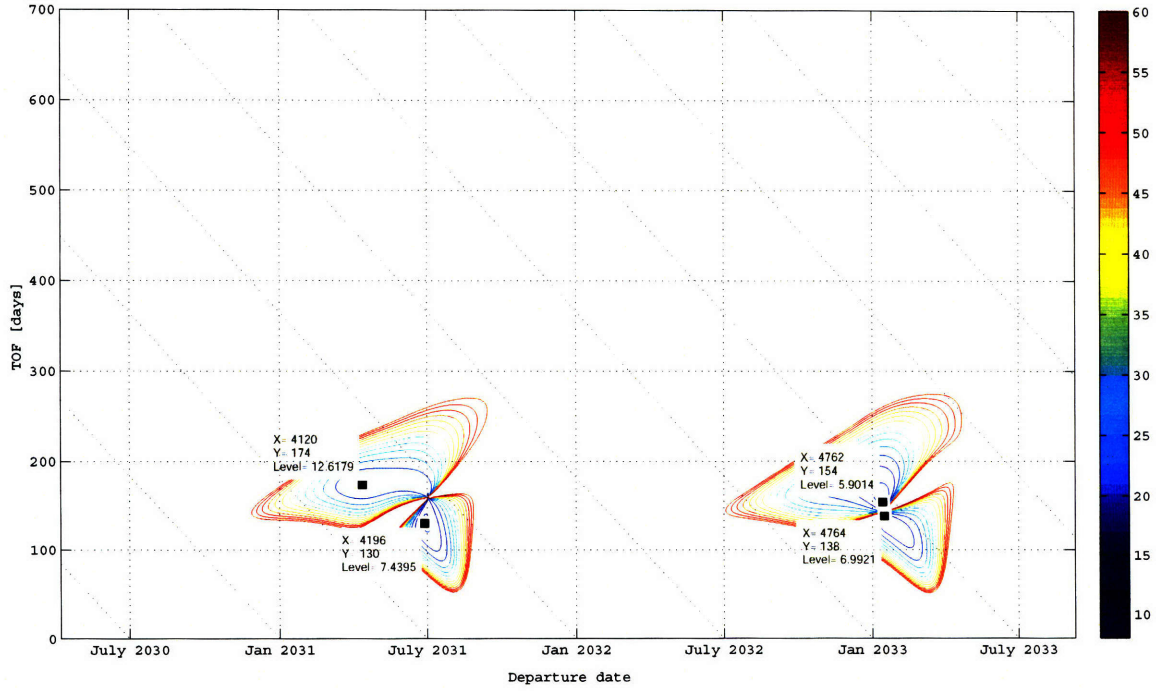


Figure C-20: $C3_a$ contours: Venus-Earth direct Period 4

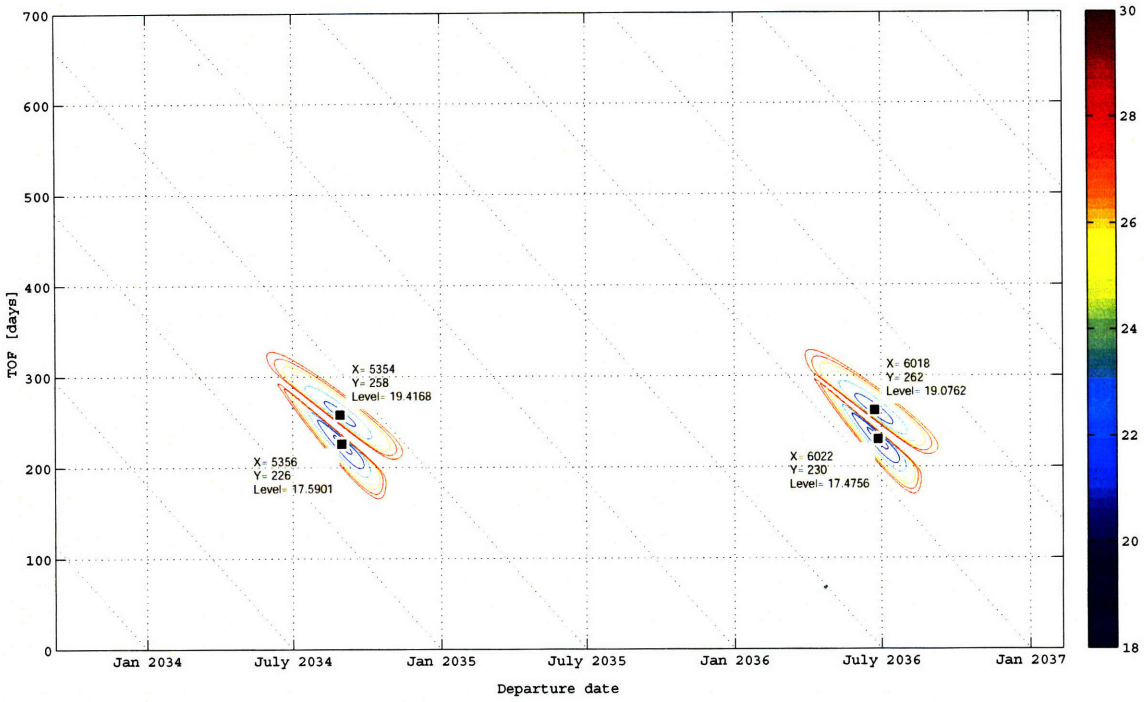


Figure C-21: $C3_d$ contours: Mars-Venus direct Period 5

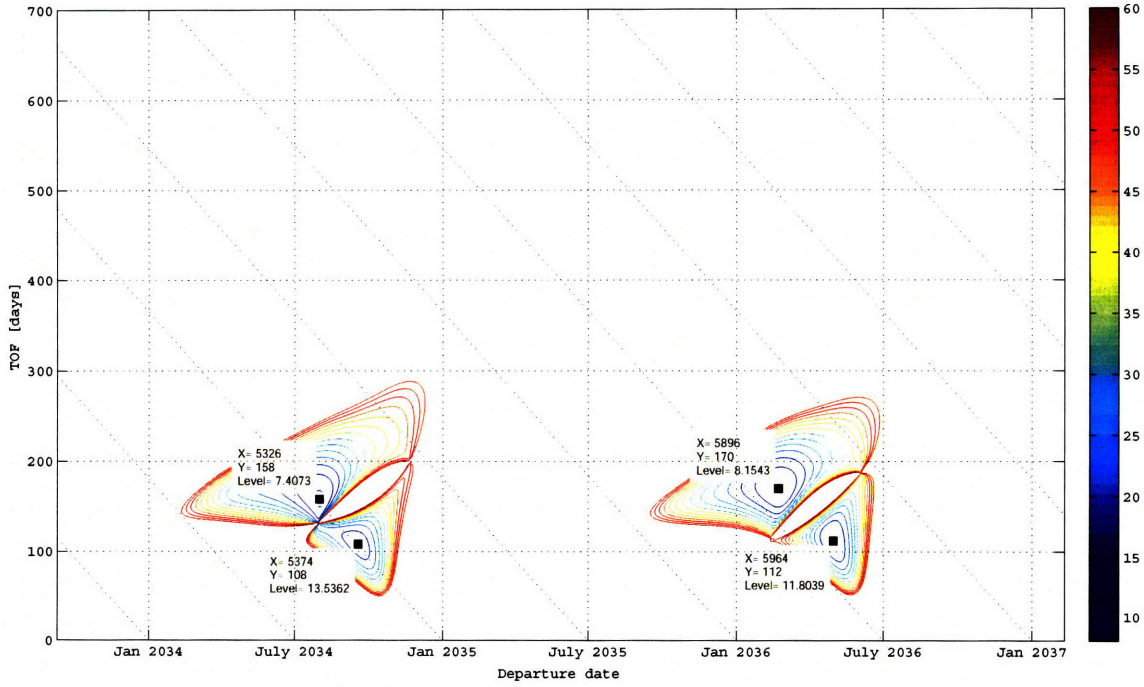


Figure C-22: $C3_a$ contours: Venus-Earth direct Period 5

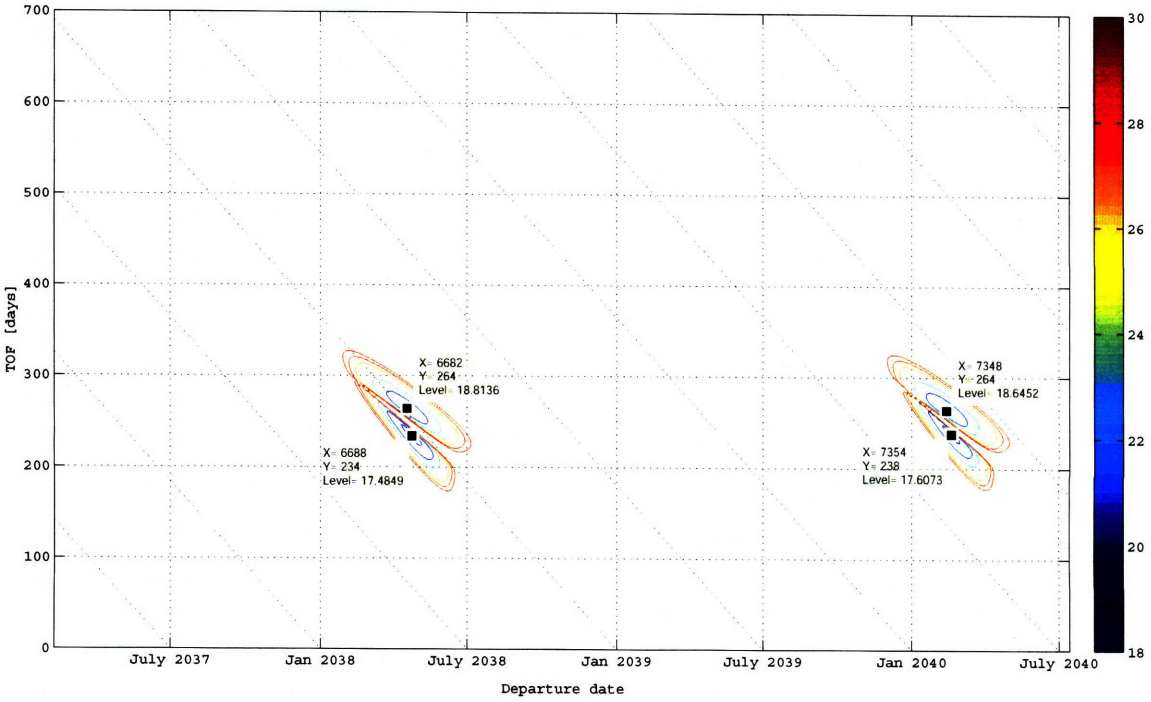


Figure C-23: $C3_d$ contours: Mars-Venus direct Period 6

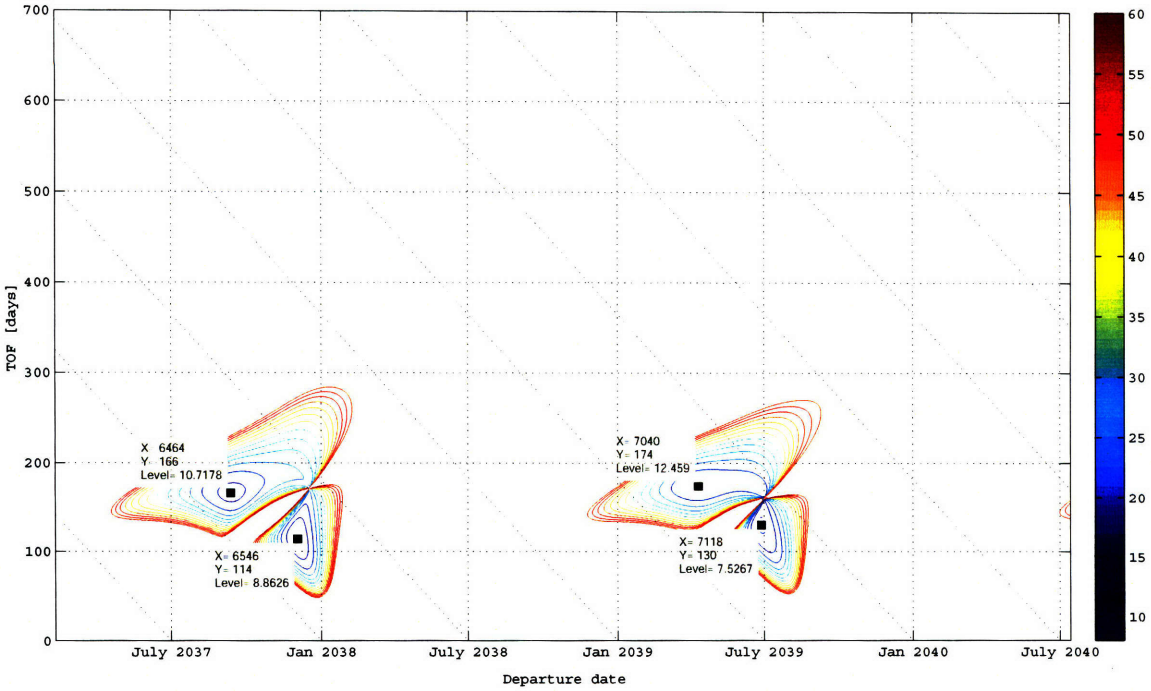


Figure C-24: $C3_a$ contours: Venus-Earth direct Period 6

Bibliography

[1] President George W. Bush. A Renewed Spirit of Discovery – The President's Vision for U.S. Space Exploration. Technical report, The White House, Washington, D.C., January 2004.

[2] Wilfried K. Hofstetter. Extensible Modular Landing Systems for Human Moon and Mars Exploration. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, December 2004.

[3] Wilfried K. Hofstetter, Olivier L. de Weck, and Edward F. Crawley. Modular Building Blocks for Manned Spacecraft: A Case Study for Moon and Mars Landing Systems. Proceedings of the 15th Annual International Symposium, Rochester, NY, 2005. INCOSE.

[4] Paul D. Wooster, Wilfried K. Hofstetter, and Edward F. Crawley. Crew Exploration Vehicle Destination for Human Lunar Exploration: The Lunar Surface. Number AIAA 2005-6626 in Space 2005, Long Beach, CA, August-September 2005. AIAA.

[5] Wilfried K. Hofstetter, Paul D. Wooster, William D. Nadir, and Edward F. Crawley. Affordable Human Moon and Mars Exploration through Hardware Commonality. Number AIAA 2005-6757 in Space 2005, Long Beach, CA, August-September 2005. AIAA.

[6] Gergana A. Bounova, Jaemyung Ahn, Wilfried K. Hofstetter, Paul D. Wooster, Rania Hassan, and Olivier L. de Weck. Selection and Technology Evaluation of Moon/Mars Transportation Architectures. Number AIAA 2005-6790 in Space 2005, Long Beach, CA, August-September 2005. AIAA.

[7] Paul D. Wooster, Wilfried K. Hofstetter, William D. Nadir, and Edward F. Crawley. The Mars-Back Approach: Affordable and Sustainable Exploration of the Moon, Mars, and Beyond using Common Systems. Number IAC-05-D3.1.06 in 56th International Astronautical Congress 2005, Fukuoka, Japan, October 2005. IAF.

[8] Wilfried K. Hofstetter, Paul D. Wooster, and Edward F. Crawley. Extending NASA's Exploration Systems Architecture towards Long-term Crewed Moon and Mars Operations. Number AIAA 2006-5746 in SpaceOps 2006 Conference, Rome, Italy, June 2006. AIAA.

[9] Paul D. Wooster. Strategies for Affordable Human Moon and Mars Exploration. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, February 2007.

[10] Human Exploration of Mars: The Reference Mission of the NASA Mars Exploration Study Team. Technical Report NASA SP-6107, Johnson Space Center, Houston, TX, July 1997.

[11] Reference Mission Version 3.0 – Addendum to the Human Exploration of Mars: The Reference Mission of the NASA Mars Exploration Study Team. Technical Report NASA SP-6107-ADD, Johnson Space Center, Houston, TX, June 1998.

[12] Rollion W. Gillespie and Stanley Ross. Venus-Swingby Mission Mode and Its Role in the Manned Exploration of Mars. *Journal of Spacecraft and Rockets*, 4(2):170–175, 1967.

[13] Gerald Walberg. How Shall We Go to Mars? A Review of Mission Scenarios. *Journal of Spacecraft and Rockets*, 30(2):129–139, March-April 1993.

[14] Robert D. Braun, Richard W. Powell, and Lin C. Hartung. Effect of Interplanetary Trajectory Options on a Manned Mars Aerobrake Configuration. Technical Report NASA TP-3019, NASA Langley Research Center, Hampton, VA, August 1990.

[15] Lorenzo Casalino, Guido Colasurdo, and Dario Pastrone. Optimization Procedure for Preliminary Design of Opposition-Class Mars Missions. *Journal of Guidance, Control, and Dynamics*, 21(1):134–140, January-February 1998.

[16] Damon F. Landau and James M. Longuski. A Reassessment of Trajectory Options for Human Missions to Mars. Number AIAA 2004-5095 in AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Providence, RI, August 2004. AIAA.

[17] Paul D. Wooster, Robert D. Braun, Jaemyung Ahn, and Zachary R. Putnam. Trajectory Options for Human Mars Missions. Number AIAA 2006-6308 in AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Keystone, Colorado, August 2006. AIAA.

[18] Anon. *Space Flight Handbooks Volume III: Planetary Flight Handbook. NASA SP-35.* NASA, 1968.

[19] Richard H. Battin. *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition.* AIAA Education Series, 1999.

[20] Howard D. Curtis. *Orbital Mechanics for Engineering Students.* Elsevier Aerospace Engineering Series, first edition, 2005.

[21] Roger R. Bate, Donald D. Mueller, and Jerry E. White. *Fundamentals of Astrodynamics.* Dover Publications, 1971.

[22] Steve Matousek and Andrey B. Sergeyevsky. To Mars and Back: 2002-2020 Ballistic Trajectory Data for the Mission Architect. Number AIAA 98-4396 in AIAA/AAS Astrodynamics Specialist Conference, Boston, MA, August 1998. AIAA.

[23] Andrey B. Sergeyevsky. Mission Design Data for Venus, Mars, and Jupiter through 1990. Number JPL-TM-33-736-VOL-1; NASA-CR-143505, Pasadena, CA, September 1975. Jet Propulsion Laboratory.

[24] Takuto Ishimatsu and Shinya Umeno. Optimization of Earth-Mars Interplanetary Trajectory Selection. May 2008.