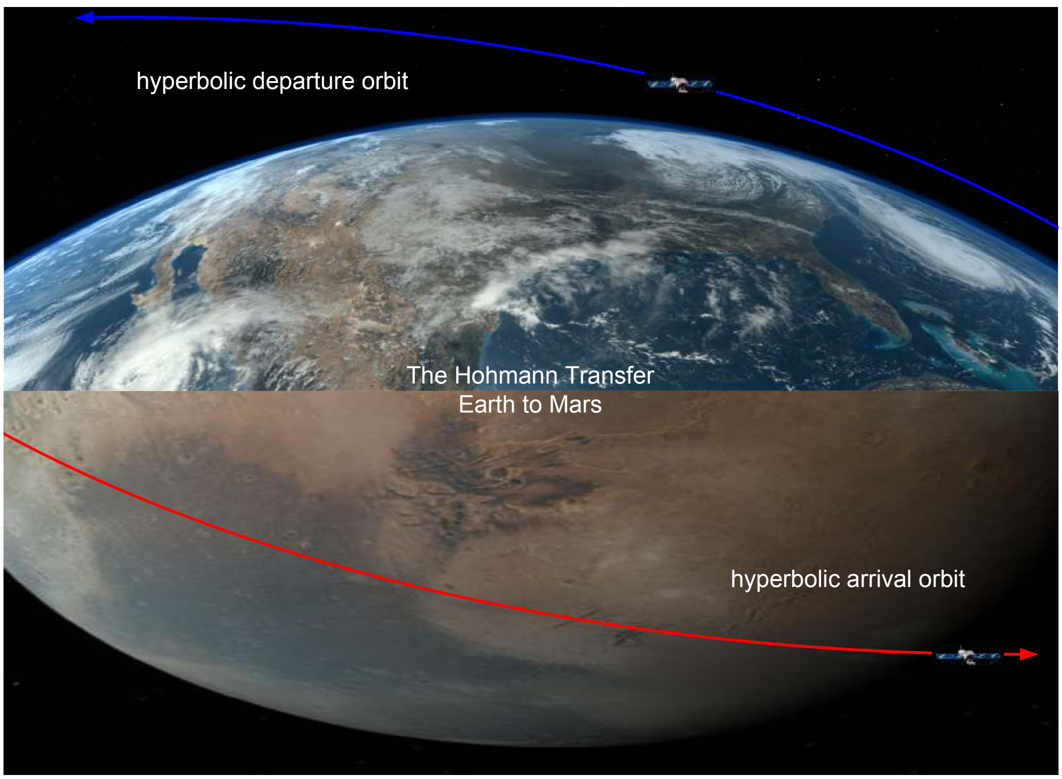
Extending the Patched-Conic Approximation to the Restricted Four-Body Problem

Thomas R. Reppert
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061

May 3, 2006



Abstract

This paper presents an investigation of how to refine patched-conic orbit approximations with a restricted four-body orbit setup. Patched-conic orbit approximations offer an efficient method for evaluating interplanetary orbits. By approximating the actual orbit as a series of two-body orbits, they offer a greatly simplified way of analyzing missions. However, this computational simplification comes at a cost. Patched-conic approximations are limited in their ability to fully represent a particular orbit. Therefore, we must use a numerical integration technique to more precisely describe interplanetary missions. By extending the patched-conic approximation to a restricted four-body problem, we achieve more a precise orbit transfer description. Taking into consideration the gravitational influences of the sun, Earth, and Mars at all times, we compute a spacecraft's transfer orbit from Earth to Mars. All orbits are assumed to be planar. Thus, the integrator provides a more precise estimate of the state of the vehicle upon its arrival at Mars. After the orbit calculation is complete, we perform a series of sanity checks in an attempt to verify the legitimacy of the integration. Of particular interest is how the departure hyperbolic periapses velocity may be modified in order to achieve a desired arrival position and velocity. Thus, the spacecraft's initial state is adjusted and the effects upon its arrival state are measured. Lastly, the predicted values of required departure burn are compared for the Hohmann solution, the patched-conic approximation, and the restricted four-body problem.

Nomenclature

t	Time
P	Transfer period
γ	Phase difference
Φ	Burn angle
σ	Heliocentric heading angle
ϑ	Planet-centric heading angle
a	Semi-major axis
e	Eccentricity
$\boldsymbol{h}$	Angular momentum
p	Semilatus rectum
μ	Gravitational coefficient
$\boldsymbol{r}$	Orbit position
n	Mean orbit rate
d	Distance
$\boldsymbol{v}$	Heliocentric velocity
$\boldsymbol{\nu}$	Planet-centric velocity
$\boldsymbol{y}$	State vector
$\boldsymbol{k}$	Slope estimate
g	Integration time step
$\boldsymbol{\phi}$	Increment function
G	Universal gravitation constant
m	Point mass

Subscript

c	Critical orbit
e	Elliptic orbit
h	Hyperbolic orbit
$\oplus$	Earth
♂	Mars
$\odot$	Sun
p	Periapses point
0	Parking orbit departure stage
1	Hyperbolic orbit departure stage
2	Hyperbolic orbit arrival stage
3	Parking orbit arrival stage

Constant	
Sun's gravitational coefficient $\mu_{\odot}$ (km^3/s^2)	1.326e+011
Earth's gravitational coefficient $\mu_{\oplus}$ (km^3/s^2)	3.985e+005
Mars' gravitational coefficient $\mu_{♂}$ (km^3/s^2)	4.282e+004
Earth's mean orbit radius $r_{\oplus}$ (km)	1.496e+008
Mars' mean orbit radius $r_{♂}$ (km)	2.2794e+008
Earth's mean orbit rate $n_{\oplus}$ (rad/s)	1.9901e-007
Mars' mean orbit rate $n_{♂}$ (rad/s)	1.0581e-007
Earth's mean orbit velocity $\upsilon_{\oplus}$ (km/s)	29.772
Mars' mean orbit velocity $\upsilon_{♂}$ (km/s)	24.119
Hohmann semi-major axis $a_{\oplus}$ (km)	1.8877e+008

Contents

List of Tables

List of Figures

Chapter 1

Introduction

A Hohmann transfer is an interplanetary mission that requires a change in true anomaly of 180 degrees. It is a particular type of minimum energy transfer orbit. In fact, the Hohmann transfer is an interplanetary mission that requires a minimum initial burn in order to reach the foreign planet.[1] The Hohmann is commonly used to transfer from one circular orbit to another. Thus, it is an attractive option for designing future missions from Earth to Mars.

Analytic solutions relating the planets' mean heliocentric orbit radii to the required departure burn have already been established.[2] These equations were developed chiefly through the application of conservation laws, including the conservations of both angular momentum and energy. But these solutions only provide a rough estimate of how to reach Mars' sphere of influence. We desire a higher fidelity method for estimating the required initial burn. In addition, we seek a method which allows us to alter the departure orbit geometry and to analyze the effects upon the arrival at Mars. The analytic Hohmann solution fully disregards the gravitational influences of both Earth and Mars. By failing to consider the spacecraft's departure and arrival orbits, this solution only provides a preliminary estimate of the required departure burn.

The patched-conic approximation has thus been developed as a more accurate solution to interplanetary transfer description. It involves partitioning the overall transfer into distinct conic solutions. For instance, as a spacecraft travels from Earth to Mars, its orbit is approximated as a hyperbolic departure, an elliptic transfer, and a hyperbolic arrival. The patched-conic approximation breaks the entire orbit down into several two-body problems. In other words, only one celestial body's influence is considered to be acting upon the spacecraft at all times. This approximation provides a much better understanding of the relation between the departure orbit and the overall transfer than the analytic Hohmann solution. However, the patched-conic approximation is still limited in that it only considers the gravity of one celestial body at a time. During a true Hohmann transfer from Earth to Mars, the sun will have some minute gravitational effect upon the spacecraft during both the departure and arrival orbits. A numerical integrator is necessary in order to take such smaller perturbations into account.

When looking to design a real-time interplanetary mission from Earth to Mars, we seek a higher fidelity orbit description than the patched-conic approximation. Thus, we introduce the restricted four-body problem, which offers a more precise representation of the transfer orbit. The restricted four-body problem, applied to a Hohmann transfer from Earth to

Mars, considers the gravitational influences of Earth, the sun, and Mars at all times. Thus, unlike the patched-conic approximation, this orbit integration scheme allows us view the direct effect of altered initial conditions upon the hyperbolic arrival orbit. Perhaps most importantly, the restricted four-body problem presents a method of analyzing a highly non-linear transfer orbit without breaking the actual orbit into separate parts.

Chapter 2

Patched-Conic Approximation

The patched-conic approximation offers an efficient method for developing interplanetary orbits. By partitioning the overall orbit into a series of two-body orbits, it greatly simplifies mission analysis. For instance, the initial part of an interplanetary voyage may be approximated as a hyperbola with the departure planet at the focus. Once the spacecraft leaves the departure planet's sphere of influence, the orbit may be approximated as an ellipse whose focus is centered at the sun. Once the vehicle enters the arrival planet's sphere of influence, the orbit may again be approximated as hyperbolic, with its focus now centered at the arrival planet. For each of the three portions of the orbit, one gravitational force is assumed to be acting upon the spacecraft at a time.[1]

To illustrate the efficiency of the patched-conic approximation, we partition the standard Hohmann transfer of a spacecraft traveling from Earth to Mars into three separate conic stages. During the initial portion of the voyage, we approximate the transfer as a hyperbolic departure orbit with its primary focus positioned at the center of the Earth. After escaping the Earth's sphere of influence, the spacecraft then enters its elliptic orbit about the sun. Following this second stage, the spacecraft enters Mars' sphere of influence. Once again, we approximate the motion as a hyperbolic orbit, this time with its focus located at the center of Mars. All portions of the transfer orbit are assumed to be planar. Because each portion of the voyage is considered a two-body problem, there is never more than one gravitational force acting upon the spacecraft at a given time.

2.1 Establishing the Initial Offset Angle

If the spacecraft is to intercept Mars after its transfer, there needs to exist some specific offset angle $\gamma(t_1)$ between Earth and Mars at the initial time t_1. Figure 2.1 illustrates the Hohmann transfer orbit from Earth to Mars. Note that time t_1 corresponds to the instant at which the spacecraft leaves Earth's sphere of influence, while time t_2 denotes the instant when the craft enters Mars' sphere of influence. If $n_\oplus$ denotes the mean Hohmann orbit rate, then the Hohmann transfer period is given by

$$P = \frac{1}{2}\frac{2\pi}{n_\oplus} = \pi\sqrt{\frac{a_\oplus^3}{\mu_\odot}} \tag{2.1}$$

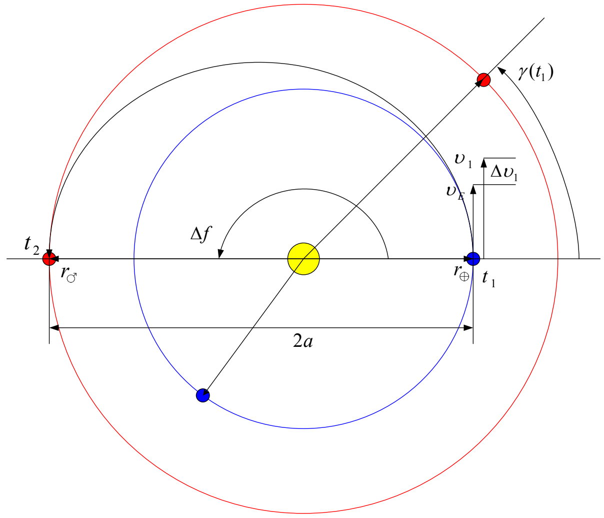


Figure 2.1: Illustration of the Hohmann transfer from Earth to Mars.

where $a_\oplus$ is the semi-major axis of the transfer orbit, as shown in Figure 2.1. Taking the value of $a_\oplus$ from the constants listing, the transfer period P is determined to be 258.979 days. Because Mars travels a distance $n_♂ P$ during the spacecraft's travel, the initial phase difference between Earth and Mars must be

$$\gamma(t_1) = \pi - n_♂ P \tag{2.2}$$

where $n_♂$ corresponds to Mars' mean orbit rate. Using the value of $n_♂$ given in the constants listing, the initial offset angle for the Hohmann transfer is found to be 44.343 deg.

2.2 Determining the Heliocentric Departure Velocity

The typical application of the patched-conic solution is to determine approximately what $\Delta\upsilon$ is needed to complete a certain transfer mission. This method is most accurate in establishing the magnitude of the $\Delta\upsilon$, as opposed to its direction or timing. We first seek the necessary heliocentric velocity υ_1 as the spacecraft leaves the Earth's sphere of influence. This particular velocity is illustrated in Figure 2.1. The υ_1 necessary to complete the Hohmann transfer may be computed as

$$\upsilon_1 = \sqrt{\frac{2\mu_\odot}{r_\oplus + r_♂}\left(\frac{r_♂}{r_\oplus}\right)} \tag{2.3}$$

where $\mu_\odot$ denotes the sun's gravitational coefficient, $r_\oplus$ denotes the Earth's mean orbit radius, and $r_♂$ denotes Mars' mean orbit radius. For a complete derivation of Equation 2.3, consult Schaub and Junkins.[2] Using the quantities for $\mu_\odot$, $r_\oplus$, and $r_♂$ given in the constants listing, the heliocentric velocity υ_1 is computed to be 32.715 km/s. Once the heliocentric departure velocity is calculated, Δv_1 may be computed as

$$\Delta v_1 = v_1 - v_\oplus = v_\oplus \left(\sqrt{\frac{2r_♂}{r_\oplus + r_♂}} - 1 \right) \tag{2.4}$$

where $v_\oplus$ is the Earth's mean heliocentric velocity, as shown in Figure 2.1. Because $r_♂ > r_\oplus$, the resulting Δv_1 will be positive. Using the value of $v_\oplus$ in the constants listing, the Δv_1 for the Hohmann transfer is calculated as 2.943 km/s. After performing the Δv_1, the spacecraft leaves Earth's sphere of influence and enters into its elliptic orbit about the sun. During this portion of the mission, the sun is the only celestial body considered to be influencing the motion of the spacecraft. At the end of the Hohmann transfer, the spacecraft enters Mars' sphere of influence and begins its hyperbolic arrival orbit. The departure and arrival orbits will now be considered.

2.3 Leaving Earth's Sphere of Influence

The following discussion offers a closer examination of how the spacecraft escapes Earth's sphere of influence. Figure 2.2 offers an illustration of the departure. Note that time t_0 corre-

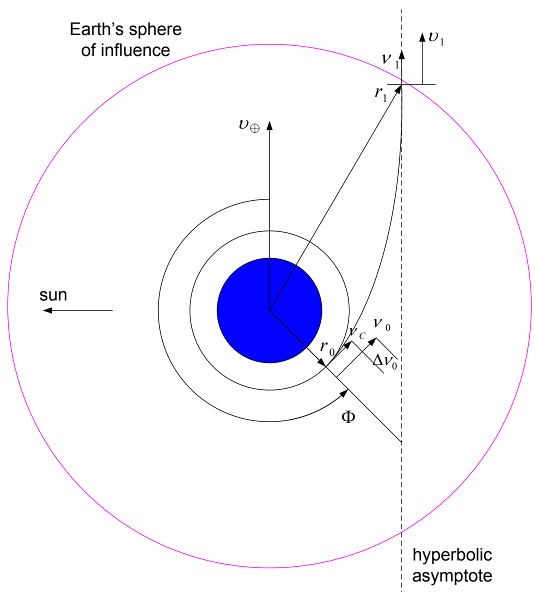


Figure 2.2: Illustration of the hyperbolic departure from Earth's sphere of influence.

sponds to the spacecraft's departure from the initial parking orbit, while time t_1 again corresponds to the instant at which the spacecraft escapes Earth's sphere of influence. Throughout the following analysis, heliocentric velocities are expressed as v_i, while planet-centric velocities are denoted as ν_i. In an attempt to leave Earth's sphere of influence, either a parabolic or hyperbolic orbit is necessary. But because the spacecraft is required to converge to some velocity v_1 as it leaves Earth's sphere of influence, the departure orbit must be hyperbolic.

The necessary Earth-relative velocity ν_1 at the limit of the sphere of influence is computed as

$$\nu_1 = \upsilon_1 - \upsilon_\oplus \tag{2.5}$$

Thus, the corresponding Earth-relative velocity ν_1 for the Hohmann transfer is 2.943 km/s. We can also use the vis-viva equation[2] to determine the Earth-relative velocity ν_1 as

$$\nu_1 = \sqrt{\frac{2\mu_\oplus}{r_1} - \frac{\mu_\oplus}{a_h}} \approx \sqrt{-\frac{\mu_\oplus}{a_h}} \tag{2.6}$$

where $\mu_\oplus$ denotes the Earth's gravitational coefficient and a_h corresponds to the semi-major axis of the departure hyperbola. We approximate $r_1 \approx \infty$ due to the assumption that the spacecraft trajectory asymptotically approaches its limiting value at time t_1. Therefore, we can relate the departure hyperbola's semi-major axis to either ν_1 or υ_1 via

$$a_h = \frac{-\mu_\oplus}{\nu_1^2} = -\frac{\mu_\oplus}{(\upsilon_1 - \upsilon_\oplus)^2} \tag{2.7}$$

Because ν_1 equals 2.943 km/s, we find the hyperbolic semi-major axis a_h to be -46010 km. Using the vis-viva equation once again, the Earth-relative speed ν_0 that the vehicle must have in order to initiate the hyperbolic transfer orbit at t_0 becomes

$$\nu_0 = \sqrt{\frac{2\mu_\oplus}{r_0} - \frac{\mu_\oplus}{a_h}} \tag{2.8}$$

where r_0 denotes the spacecraft's initial parking orbit radius about the Earth. After substituting the relation for a_h given in Equation 2.7, the speed ν_0 is expressed as

$$\nu_0^2 = \nu_1^2 + \frac{2\mu_\oplus}{r_0} \tag{2.9}$$

At this point, it is important to note that once ν_1 and r_0 are chosen for a particular mission, the corresponding patched-conic approximation for ν_0 is set. However, because ν_1 is determined via the semi-major axis of the elliptic transfer orbit, we truly set ν_0 with our choices of $a_\oplus$ and r_0. Throughout the remainder of Chapter 2, we consider the value of $a_\oplus$ given in the constants listing (1.8877e+008 km), as well as an initial parking orbit radius r_0 of 7500 km. Using these values, we find a corresponding ν_0 of 10.722 km/s.

In order to maintain its initial parking orbit about Earth, the spacecraft has a critical speed of

$$\nu_c = \sqrt{\frac{\mu_\oplus}{r_0}} \tag{2.10}$$

which is calculated to be 7.290 km/s given the previous values of $\mu_\oplus$ and r_0. Thus, in order to begin the hyperbolic transfer, the initial burn required is given as

$$\Delta\nu_0 = \nu_0 - \nu_c = \sqrt{2\nu_c^2 + \nu_1^2} - \nu_c \tag{2.11}$$

As shown in Figure 2.2, the point where the initial $\Delta\nu_0$ burn must be applied is defined via the angle Φ. For any transfer to an outer planet, the spacecraft's velocity should asymptotically align itself with the Earth's heliocentric velocity. Thus, the burn angle Φ may be determined from the geometry of the departure hyperbola as

$$\Phi = \cos^{-1}\left(\frac{1}{e_h}\right) + \pi \tag{2.12}$$

where e_h refers to the eccentricity of the hyperbolic departure orbit. For a complete derivation of Equation 2.12, refer to Bate, Mueller, and White.[3] In order to find the departure eccentricity, we analyze the orbit's angular momentum. Referring to the definition of angular momentum, as well as the orbit geometry, we find that

$$h_h^2 = \mu_\oplus p = \mu_\oplus a_h(1 - e_h^2) = \mu_\oplus r_p(1 + e_h) \tag{2.13}$$

where h_h denotes the angular momentum of the departure orbit, p refers to the departure semilatus rectum, and r_p represents the radius at periapses. But because the burn point at time t_0 is the periapses point of the departure hyperbola, we know that $r_0 = r_p$. Therefore, the angular momentum can also be expressed as

$$h_h^2 = r_0^2\nu_0^2 \tag{2.14}$$

Relating Equations 2.13 and 2.14, we we can now express the departure orbit eccentricity as

$$e_h = \frac{r_0\nu_0^2}{\mu_\oplus} - 1 = \frac{r_0\nu_1^2}{\mu_\oplus} + 1 \tag{2.15}$$

Using the previously stated values of r_0 and ν_0 (7500 km and 10.722 km/s, respectively), the departure eccentricity e_h is given as 1.163. It is important to note that all hyperbolic orbits must have an eccentricity greater than one. Finally, referring back to Equation 2.12 and using the calculated value of e_h, we find the initial burn angle Φ to be roughly 211 degrees.

The patched-conic solution analytically approximates the required velocity ν_1 at the Earth's sphere of influence necessary to begin the Hohmann transfer to Mars. More rigorous details, such as the effect of the distance between the spacecraft velocity direction and the Earth's heliocentric velocity direction on the orbit, require the use of numerical integration techniques. These techniques will be introduced in Chapter 3.

2.4 Entering Mars' Sphere of Influence

The following section presents the patched-conic approximation of how the spacecraft enters Mars' sphere of influence. Figure 2.3 offers an illustration of the arrival orbit. It is typical for any spacecraft travelling to an outer planet to enter that planet's sphere of influence ahead of the planet. The spacecraft reaches the outer planet at the apoapses of the transfer orbit. Therefore, the spacecraft's speed will be less than that of the planet, allowing the planet to overtake it. Once again, using the vis-viva equation,[2] we find the heliocentric arrival velocity υ_2 of the spacecraft to be

$$\upsilon_2 = \sqrt{2\mu_\odot\left(\frac{1}{r_♂} - \frac{1}{r_\oplus}\right) + \upsilon_1^2} \tag{2.16}$$

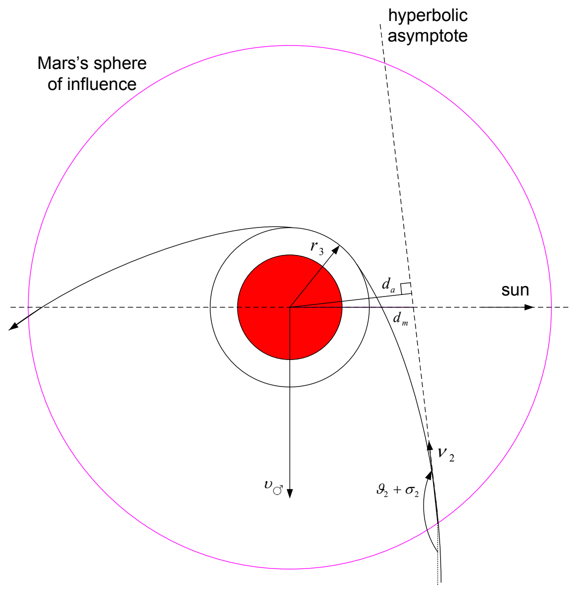


Figure 2.3: Illustration of the hyperbolic arrival at Mars' sphere of influence.

Given the previous calculation of v_1, we find v_2 to be 21.471 km/s for the transfer to Mars. In general, the spacecraft's heliocentric velocity will be tangent to that of the Earth when it begins the Hohmann. But when it arrives at Mars, it will most likely cross Mars' sphere of influence with some heading angle σ_2.[3]. In order to compute the σ_2 heading angle between the sun-normal direction and the craft's heliocentric velocity, we again recall the formal definition of angular momentum as

$$\boldsymbol{h} = \boldsymbol{r} \times \boldsymbol{v} \tag{2.17}$$

Assuming that the radius of Earth's sphere of influence is negligible compared with the major heliocentric orbit axis, we find $h_e = r_{\oplus} v_1$. As the spacecraft enters Mars' sphere of influence, its angular momentum can also be given as

$$h_e = |\boldsymbol{r}_{♂} \times \boldsymbol{v}_2| = r_{♂} v_2 \sin(\frac{\pi}{2} - \sigma_2) = r_{♂} v_2 \cos(\sigma_2) \tag{2.18}$$

Thus, we find the heading angle relative to the sun normal direction to be

$$\sigma_2 = \cos^{-1}\left(\frac{h_e}{r_{♂} v_2}\right) = \cos^{-1}\left(\frac{r_{\oplus} v_1}{r_{♂} v_2}\right) \tag{2.19}$$

The heading angle corresponding to the Hohmann transfer is determined to be roughly 0 degrees. For a perfect Hohmann transfer, the value of σ_2 would be exactly equal to 0 degrees. To compute the spacecraft's Mars-centric velocity vector $\boldsymbol{\nu}_2$, Mars' heliocentric velocity must be subracted from that of the spacecraft:

$$\boldsymbol{\nu}_2 = \boldsymbol{v}_2 - \boldsymbol{v}_{♂} \tag{2.20}$$

Via the law of cosines, the magnitude of $\boldsymbol{\nu}_2$ is calculated as

$$\nu_2 = \sqrt{v_2^2 + v_{♂}^2 - 2 v_2 v_{♂} \cos\sigma_2} \tag{2.21}$$

which yields a value of 2.648 km/s for the given Hohmann. The heading angle ϑ_2 between the $\boldsymbol{\upsilon}_2$ and $\boldsymbol{\nu}_2$ velocity vectors is roughly 180 degrees for the Hohmann transfer, where Figure 2.4 offers an illustration of the triangular geometry. For a perfect Hohmann transfer, the

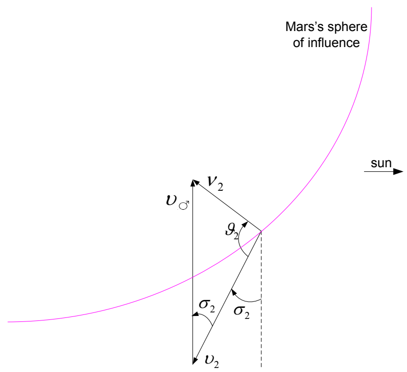


Figure 2.4: Illustration of the two heading angles upon entering Mars' sphere of influence.

value of ϑ_2 would be exactly 180 degrees.

Identical to the process used for the departure orbit, we use the energy (vis-viva) equation to determine the semi-major axis of the arrival orbit through

$$\frac{1}{a_h} = \frac{2}{r_2} - \frac{\nu_2^2}{\mu_♂} \tag{2.22}$$

Making the patched-conic assumption that the spacecraft's approach orbit is hyperbolic, we approximate a_h as

$$a_h = -\frac{\mu_♂}{\nu_2^2} \tag{2.23}$$

where $r_2 \approx \infty$. If the Hohmann orbit were perfect, the spacecraft would directly hit the Martian surface. To avoid this occurence, the hyperbolic arrival trajectory is aimed such that it will miss Mars by some miss distance d_m, as shown in Figure 2.3. However, from the spacecraft's perspective, it is easiest to estimate the shortest distance d_a between the approach asymptote and Mars, given by

$$d_a = d_m \sin(\vartheta_2 + \sigma_2) \tag{2.24}$$

Similar to the departure orbit, we examine the spacecraft's constant angular momentum in order to determine the arrival eccentricity. Figure 2.3 illustrates how the angular momentum simplifies to

$$h_h = |\boldsymbol{r}_2 \times \boldsymbol{\nu}_2| = d_a \nu_2 \tag{2.25}$$

Substituting Equation 2.25 into Equation 2.13, we find the hyperbolic eccentricity e_h as

$$e_h = \sqrt{1 + \frac{d_a^2 \nu_2^4}{\mu_♂}} \tag{2.26}$$

Finally, the periapses radius r_p of the arrival orbit can be calculated by substituting Equation 2.23 into the angular momentum expression of Equation 2.13 as

$$r_p = \frac{\mu_{\male}}{\nu_2^2}(e_h - 1) \tag{2.27}$$

The transfer mission is usually designed in such a way that the periapses radius is equivalent to the final parking orbit radius. Thus, the final orbit radius about Mars is uniquely determined once both the eccentricity e_h and arrival speed ν_2 are given. Because e_h depends upon the miss distance, the arrival is actually set with prescribed values of d_m and ν_2.

Chapter 3

Restricted Four-Body Problem

In this section, we extend the patched-conic approximation to a restricted four-body problem. Taking into consideration the gravitational influences of the sun, Earth, and Mars at all times, we determine the spacecraft's transfer orbit from Earth to Mars. The motion is not analyzed with respect to the three separate spheres of influence. Instead, this method incorporates the gravitational effects of each celestial body even when the spacecraft is beyond the body's sphere of influence. The analysis incorporates these comparatively minute effects in order to better estimate the exact state of the vehicle upon arrival at Mars. All orbital motion during the transfer is assumed to be planar. Thus, the effects of Earth's and Mars' orbit inclinations are not considered when integrating the spacecraft's trajectory.

Because the restricted four-body problem is considerably more complex than the two-body problem, numerical integration is used to develope the spacecraft's orbit. Integrating numerically allows for the incorporation of the sun's, Earth's, and Mars' gravitational influences at all times. After completing the orbit integration, we perform a series of sanity checks on the results in an attempt to verify their legitimacy. Of particular interest is how the initial conditions of the four-body setup may be modified in order to achieve the desired arrival position and velocity. Thus, adjustments in the initial position and velocity data are made and the effects upon the spacecraft's arrival at Mars are measured.

3.1 Derivation of the Equations of Motion

3.1.1 Application of the n-Body Problem

Before beginning the numerical integration process, we must first derive the equations of motion that we wish to integrate. Figure 3.1 offers an illustration of the coordinate frames we use to designate the state of the spacecraft for all time t. The $\mathcal{S}$: $\{\hat{\boldsymbol{s}}_1, \hat{\boldsymbol{s}}_2, \hat{\boldsymbol{s}}_3\}$ frame is an inertial frame centered at the sun. Thus, we make the assumption that the sun is stationary during the spacecraft's transfer orbit. The $\mathcal{E}$: $\{\hat{\boldsymbol{e}}_1, \hat{\boldsymbol{e}}_2, \hat{\boldsymbol{e}}_3\}$ frame is a non-rotating frame centered at Earth. We use this frame to describe the state of the spacecraft with respect to Earth. In addition, the $\mathcal{M}$: $\{\hat{\boldsymbol{m}}_1, \hat{\boldsymbol{m}}_2, \hat{\boldsymbol{m}}_3\}$ frame is a non-rotating frame centered at Mars. In a similar manner, we use the M frame to track the state of the spacecraft relative to Mars. Now that the coordinate frames have been established, we can express the position

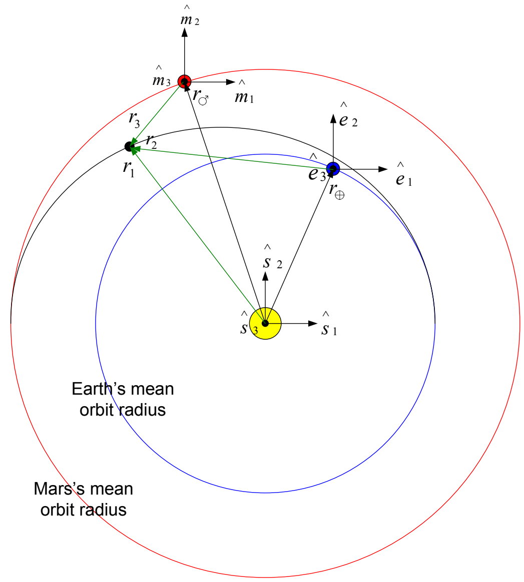


Figure 3.1: Definition of the coordinate frames and position vectors used during the derivation of the spacecraft's equations of motion.

of the spacecraft with respect to the three origins. As shown in Figure 3.1, the spacecraft's position with respect to the sun, Earth, and Mars are labelled $\boldsymbol{r_1}$, $\boldsymbol{r_2}$, and $\boldsymbol{r_3}$, respectively.

For a general n-body problem, the total force $\boldsymbol{f}_i$ acting upon mass m_i, due to the other $n-1$ masses, is

$$\boldsymbol{f}_i = G \sum_{j=1}^{n} \frac{m_i m_j}{r_{ij}^3} (\boldsymbol{r}_j - \boldsymbol{r}_i) \tag{3.1}$$

where G is the universal gravitation constant. Note that the term for which $i = j$ is to be omitted. Newton's Second Law of Motion states

$$\boldsymbol{f}_i = m_i \frac{d^2 \boldsymbol{r}_i}{dt^2} \tag{3.2}$$

Therefore, the n vector differential equations

$$\frac{d^2 \boldsymbol{r}_i}{dt^2} = G \sum_{j=1}^{n} \frac{m_j}{r_{ij}^3} (\boldsymbol{r}_j - \boldsymbol{r}_i) \tag{3.3}$$

along with appropriate initial conditions completely describe the motion of the system of n particles. Consult Battin for the complete derivation of Equation 3.3.[1] With our restricted four-body assumption, we neglect the gravitational effects of the spacecraft upon the three celestial bodies. We also treat the two planetary orbits as perfect circles, neglecting any relatively small deviations from these idealized orbits. Thus, for our specific case, we can apply Equation 3.3 as

$$\ddot{\boldsymbol{r}}_1 + \frac{\mu_\odot}{r_1^3} \boldsymbol{r}_1 + \frac{\mu_\oplus}{r_2^3} \boldsymbol{r}_2 + \frac{\mu_♂}{r_3^3} \boldsymbol{r}_3 = 0 \tag{3.4}$$

where $\ddot{\boldsymbol{r}}_1$ represents the second inertial derivative of $\boldsymbol{r}_1$ with respect to time. Also, we have used the relation

$$\mu = G(m_1 + m_2) \tag{3.5}$$

in order to express our equations of motion of the spacecraft in terms of the three gravitational coefficients μ_i of the celestial bodies. By analysis of Equation 3.4, we see that as the spacecraft's trajectory is outside the spheres of influence of Earth and Mars, the second term will govern. Thus, because the sun's gravitational coefficient is so much greater than that of Earth and Mars, the sun will provide the primary gravitational force during most of the transfer. But when the spacecraft is sufficiently close to either planet (i.e. within their spheres of influence), the third and fourth terms will govern. The patched-conic approximation involved a simplified version of Equation 3.4 by assuming the smaller terms for each portion of the transfer to be zero. But our four-body integrator will take these small perturbations into consideration in order to achieve a more accurate representation of the spacecraft's state development.

3.1.2 Motion of Earth and Mars

As stated earlier, we assume the motion of both Earth and Mars to be circular. Figure 3.2 offers an illustration of Earth's circular orbit about the sun. We define the angle θ from

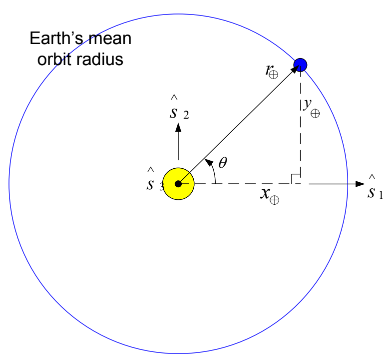


Figure 3.2: Illustration of Earth's motion with respect to the sun during the spacecraft's Hohmann transfer.

the positive $\hat{\boldsymbol{s}}_1$ direction to the Earth's position vector $\boldsymbol{r}_\oplus$. Thus, we can express the $\hat{\boldsymbol{s}}_1$ component of the Earth's position as

$$x_\oplus = r_\oplus \cos\theta \tag{3.6}$$

Similarly, we find that the $\hat{\boldsymbol{s}}_2$ component of $\boldsymbol{r}_\oplus$ is

$$y_\oplus = r_\oplus \sin\theta \tag{3.7}$$

In order to express the position of Earth in terms of time t, we can use Earth's mean orbit rate as $\theta = n_\oplus t$. Finally, we write Earth's position vector in the S frame as

$$\boldsymbol{r}_\oplus(t) = {}^{\mathcal{S}}\begin{pmatrix} r_\oplus \cos n_\oplus t \\ r_\oplus \sin n_\oplus t \\ 0 \end{pmatrix} \tag{3.8}$$

where, due to our planar orbit assumption, the third component is always equal to zero. Using the same method, we find Mars' position vector expressed in $\mathcal{S}$-frame components as

$$\boldsymbol{r}_♂(t) = {}^{\mathcal{S}}\begin{pmatrix} r_♂ \cos n_♂ t \\ r_♂ \sin n_♂ t \\ 0 \end{pmatrix} \tag{3.9}$$

Therefore, we can write the three spacecraft position vectors $\boldsymbol{r}_1$, $\boldsymbol{r}_2$, and $\boldsymbol{r}_3$ as

$$\boldsymbol{r}_1(t) = {}^{\mathcal{S}}\begin{pmatrix} x_1(t) \\ y_1(t) \\ 0 \end{pmatrix}, \quad \boldsymbol{r}_2(t) = {}^{\mathcal{S}}\begin{pmatrix} x_1(t) - r_\oplus \cos n_\oplus t \\ y_1(t) - r_\oplus \sin n_\oplus t \\ 0 \end{pmatrix}, \quad \boldsymbol{r}_3(t) = {}^{\mathcal{S}}\begin{pmatrix} x_1(t) - r_♂ \cos n_♂ t \\ y_1(t) - r_♂ \sin n_♂ t \\ 0 \end{pmatrix} \tag{3.10}$$

With all position vectors expressed in the inertial S frame as functions of time, we now set up the numerical algorithm used to determine the spacecraft's state vector during the Hohmann transfer.

3.2 Numerical Integrator

We do not have an analytical solution to the restricted four-body problem of a spacecraft's Hohmann transfer from Earth to Mars. Therefore, we require a numerical integration technique in order to estimate the spacecraft's state vector over time. The integration technique chosen to perform this task is the Classical Fourth-Order Runge-Kutta Method.[4] We choose this integrator because Runge-Kutta methods reach the accuracy of a Taylor series expansion without the necessity of computing the higher derivative terms. The generalized form of the method is

$$\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \boldsymbol{\phi} g \tag{3.11}$$

where $\boldsymbol{y_i}$ and $\boldsymbol{y_{i+1}}$ denote the state vector at times t_i and t_{i+1}, respectively, and $\boldsymbol{\phi}$ is the representative slope over the current time step g. The increment function $\boldsymbol{\phi}$ is expressed as

$$\boldsymbol{\phi} = a_1\boldsymbol{k}_1 + a_2\boldsymbol{k}_2 + \cdots + a_n\boldsymbol{k}_n \tag{3.12}$$

where the a's are constants and the $\boldsymbol{k}$'s are individual slope estimates. Note that vector notation is used for the state variables and slope estimates. We use this notation because the Runge-Kutta Method can be used to simultaneously integrate a system of ordinary differential equations. The spacecraft's state vector that we integrate for the Hohmann transfer from Earth to Mars contains six elements, and therefore six simultaneous differential equations are solved.

We can use different types of Runge-Kutta methods by varying the number of terms in the increment function $\boldsymbol{\phi}$. The Fourth-Order Runge-Kutta Method ($n = 4$) has a global truncation error on the order of g^4.[4] Figure 3.3 offers an illustration of one iteration of the Fourth-Order Runge-Kutta Method. Using this method, we integrate the state variable as

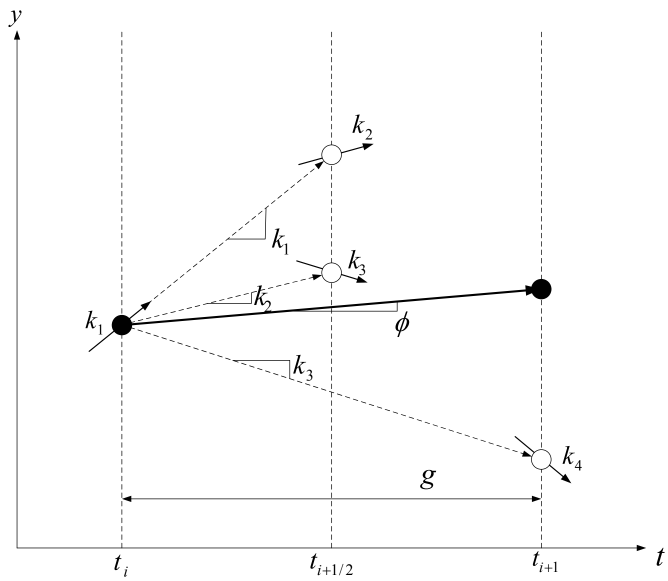


Figure 3.3: Illustration of the calculation of slope estimates during one iteration of the Fourth-Order Runge-Kutta Method.

$$\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \frac{1}{6}(\boldsymbol{k}_1 + 2\boldsymbol{k}_2 + 2\boldsymbol{k}_3 + \boldsymbol{k}_4)g \tag{3.13}$$

where

$$\boldsymbol{k}_1 = \boldsymbol{f}(t_i, \boldsymbol{y}_i) \tag{3.14a}$$

$$\boldsymbol{k}_2 = \boldsymbol{f}(t_i + \frac{1}{2}g, \boldsymbol{y}_i + \frac{1}{2}\boldsymbol{k}_1 g) \tag{3.14b}$$

$$\boldsymbol{k}_3 = \boldsymbol{f}(t_i + \frac{1}{2}g, \boldsymbol{y}_i + \frac{1}{2}\boldsymbol{k}_2 g) \tag{3.14c}$$

$$\boldsymbol{k}_4 = \boldsymbol{f}(t_i + g, \boldsymbol{y}_i + \boldsymbol{k}_3 g) \tag{3.14d}$$

and the weighting coefficients of Equation 3.12 have been given the values

$$a_1 = \frac{1}{6}, \quad a_2 = \frac{2}{6}, \quad a_3 = \frac{2}{6}, \quad a_4 = \frac{1}{6} \tag{3.15}$$

Because each of the $\boldsymbol{k}$'s represents a slope estimate, Equation 3.13 uses a weighted slope average to more efficiently determine the state vector at the future time t_{i+1}.

At this point, it must be noted that we can use a variable time step in order to improve the efficiency of the Runge-Kutta integrator. For instance, if the time step is increased for some portion of the integration, then the $\boldsymbol{k}_i$ slope estimates are averaged over larger changes in time $t_{i+1} - t_i$ during that portion. This variable time step is very convenient when applied to the restricted four-body problem. As the spacecraft travels through either Earth's or Mars' sphere of influence, it accelerates at a much greater rate than during the heliocentric portion of the mission. Therefore, it is very computationally efficient to increase the integration time step g during the heliocentric portion of the transfer orbit. The next section discusses how to implement the variable time step within the four-body algorithm.

3.3 Four-Body Algorithm

This section provides an explanation of the algorithm we use to integrate the spacecraft's state over the course of the transfer orbit. The Appendix offers a listing of the MatLab code used to execute the algorithm. As shown in Equation 3.4, the equations of motion of the spacecraft are nonlinear. We can express the initial value problem of any nonlinear system as

$$\dot{\boldsymbol{y}}(t) = \boldsymbol{f}(t, \boldsymbol{y}(t)), \quad \boldsymbol{y}(t_0) = \boldsymbol{y}_0, \quad a < t < b \tag{3.16}$$

where $\boldsymbol{y}_0$ represents the vector of initial conditions and a and b denote the time limits of integration.[5] In the case of the Hohmann transfer, the state vector $\boldsymbol{y}(t)$ and its inertial derivative $\dot{\boldsymbol{y}}(t)$ are given by

$$\boldsymbol{y}(t) = \begin{pmatrix} x_1(t) \\ y_1(t) \\ 0 \\ \dot{x}_1(t) \\ \dot{y}_1(t) \\ 0 \end{pmatrix}, \quad \dot{\boldsymbol{y}}(t) = \begin{pmatrix} \dot{x}_1(t) \\ \dot{y}_1(t) \\ 0 \\ \ddot{x}_1(t) \\ \ddot{y}_1(t) \\ 0 \end{pmatrix} \tag{3.17}$$

After writing Equation 3.4 as a system of scalar equations and substituting into Equation 3.17, we find that

$$\dot{\boldsymbol{y}}(t) = \begin{pmatrix} \dot{x}_1(t) \\ \dot{y}_1(t) \\ 0 \\ -\frac{\mu_\odot}{r_1^3}x_1 - \frac{\mu_\oplus}{r_2^3}x_2 - \frac{\mu_♂}{r_3^3}x_3 \\ -\frac{\mu_\odot}{r_1^3}y_1 - \frac{\mu_\oplus}{r_2^3}y_2 - \frac{\mu_♂}{r_3^3}y_3 \\ 0 \end{pmatrix} \tag{3.18}$$

Thus, we have derived the rate of change of the spacecraft's state vector as a function of both its current state $\boldsymbol{y}(t)$ and the current time t. We can now use the Fourth-Order Runge-Kutta Method to calculate the spacecraft's state vector over the course of the entire Hohmann transfer from Earth to Mars.

Once we establish the desired initial conditions and the appropriate time interval for the integration, we perform an iterative loop. For each time step of this loop, the following calculations are made:

1. Use the current state vector and time to determine the $\boldsymbol{k}_1$ slope estimate via Equation 3.18.

2. Use $\boldsymbol{y}(t)$ and the computed value of $\boldsymbol{k}_1$ to calculate the state at the midpoint of the interval g as

$$\boldsymbol{y}(t+\frac{g}{2}) = \boldsymbol{y}(t) + \boldsymbol{k}_1\frac{g}{2} \tag{3.19}$$

3. Use the current time t and the computed state vector at $t+\frac{g}{2}$ to calculate the new slope estimate $\boldsymbol{k}_2$.
4. Update the state vector at the midpoint of the interval by substituting $\boldsymbol{k}_2$ into Equation 3.19.
5. Repeat this process until all slope estimates $\boldsymbol{k}_1$, $\boldsymbol{k}_2$, $\boldsymbol{k}_3$, and $\boldsymbol{k}_4$ have been determined.
6. Calculate the increment function $\boldsymbol{\phi}$ as a weighted average of the four slope estimates, via Equation 3.12.
7. Use the current state vector $\boldsymbol{y}(t)$ and the weighted slope estimate $\boldsymbol{\phi}$ to calculate the future state $\boldsymbol{y}(t+h)$ as in Equation 3.13.

We use a variable time step g to increase the speed of integration. The spacecraft acceleration is greatest as it travels through the spheres of influence of Earth and Mars. Therefore, we set the planet-centric time step g_h to a lower value as the spacecraft travels through each sphere of influence. In fact, to be conservative, we implement the smaller time step g_h whenever the craft is within 1.5 times the sphere of influence of either Earth or Mars. Using a smaller time step for both the hyperbolic departure and arrival orbits allows us to accurately integrate the two portions of the transfer that are most sensitive to integration error. During the heliocentric portion of the trip, we implement a larger time step g_e in order to improve computational efficiency. The integration process does not require as small a time step during the elliptic orbit because the craft does not accelerate to the extent that it does during the hyperbolic orbits. For the specific case of the planar Hohmann transfer from Earth to Mars, the time step g_e can be at least 1000 times greater than g_h. Even with such a large difference between the two time step values, the transfers achieved are roughly identical to those that result from using the smaller step g_h throughout the entire orbit. Thus, using a variable time step allows us to greatly improve computational efficiency without sacrificing integration accuracy.

As previously stated, the Appendix contains the MatLab code used to run the iterative loop. It also contains the functions used to integrate the circular motion of both Earth and Mars about the sun. Now that we have established how the four-body integrator works, we check the validity of its output.

3.4 Validity Check

Before using the Runge-Kutta integrator to examine the restricted four-body problem, we first perform a checks upon the integrator results. It must be noted at this point that, unless stated otherwise, the initial offset angle between Earth and Mars used during integration is that solved for in section 2.1 (44.343 degrees). In order to perform the first check, we set the initial state of the spacecraft such that it should maintain a relatively circular orbit about Earth during the entire period of integration. We assign the critical velocity $\boldsymbol{\nu}_c$ of 7.290 km/s at the initial parking orbit radius of 7500 km. Figure 3.4 displays the motion of the spacecraft relative to Earth during the orbit. As expected, the spacecraft does not

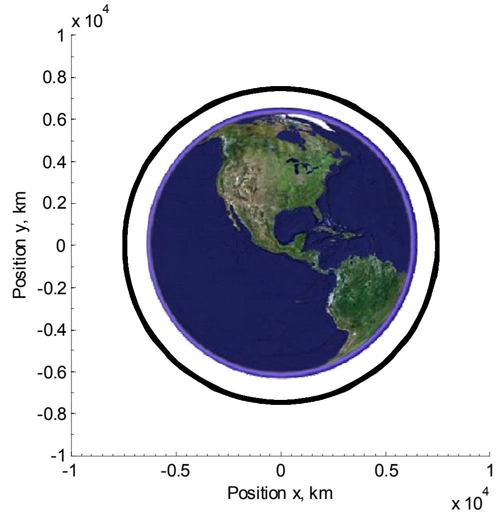
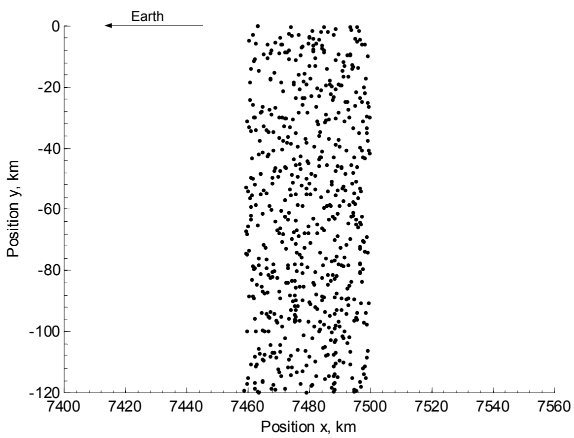


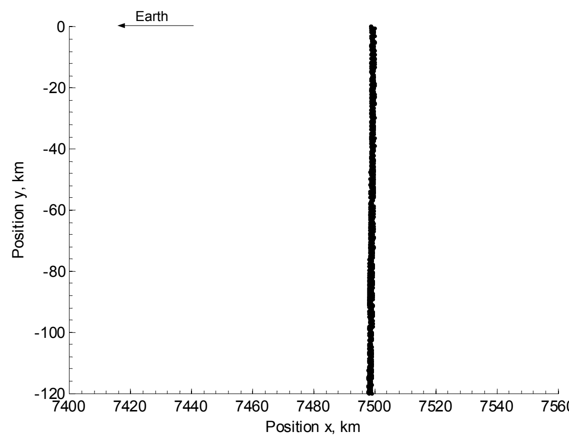
Figure 3.4: Check on a critical orbit about Earth with an initial altitude of 7500 km. The x and y positions are taken relative to the center of the non-rotating Earth frame $\mathcal{E}$.

appear to deviate substantially from an altitude of 7500 km. But we must still focus on a smaller portion of the orbit in order to determine how much the spacecraft deviates from an altitude of 7500 km. Figure 3.5 displays a small portion of the nearly-circular orbit for two different planet-centric step sizes g_h. For a step size of 100 seconds, the spacecraft stays within roughly 40 km of its initial 7500 km orbit radius. But when the step size is halved to 50 seconds, the motion of the spacecraft deviates only 3 km from the 7500 km radius. Thus, the error accrued in Figure 3.4 is most likely round-off error due to the size and precision limits of the integrator itself. Once the step size is set to 50 seconds, the spacecraft stays within 0.04 percent of its initial orbit radius. These results seem reasonable, as we expect the orbit radius to deviate slightly from its initial value due to the small gravitational influences of the sun and Mars.

After performing a check on the validity of the Runge-Kutta integrator, we now use it to examine the application of the four-body problem to a Hohmann transfer from Earth to Mars. We are interested in how the analytical solutions of the patched-conic approximation compare with the actual output of the integrator. In addition, we determine the effect of varying the initial conditions upon the development of the transfer. Thus, we can comment on the advantages and disadvantages of various orbit setups.



(a) $g_h = 100$ sec.



(b) $g_h = 50$ sec.

Figure 3.5: Illustration of the variation in altitude during a critical orbit about Earth. The x and y positions are taken relative to the center of the non-rotating Earth frame $\mathcal{E}$.

Chapter 4

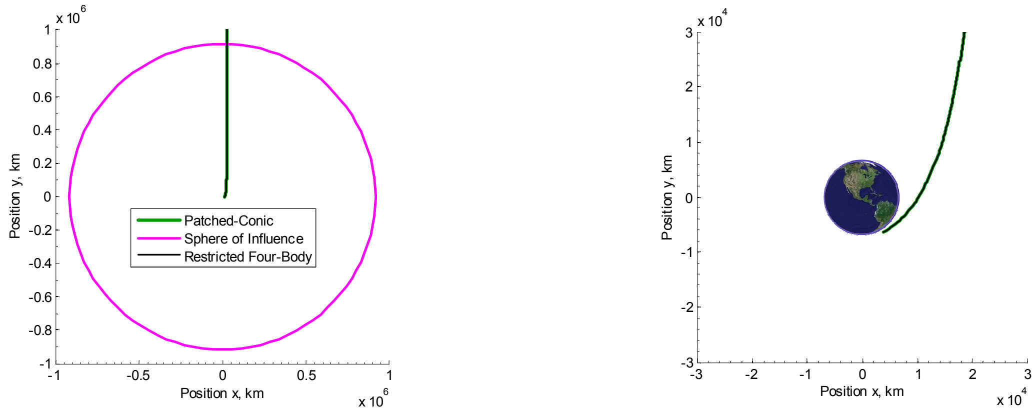
Integration Results

4.1 Comparison of the Patched-Conic and Restricted Four-Body Solutions

As noted in Chapter 2, the patched-conic approximation essentially partitions the overall transfer orbit into three separate two-body problems. During the first portion of the mission, we assume that the Earth provides the only gravitational influence upon the spacecraft. After the craft leaves Earth's sphere of influence, we then assume the sun to provide the sole gravitational influence. Once the spacecraft has entered Mars' sphere of influence, we treat Mars as the only source of gravitational force.

We first analyze the spacecraft's motion through Earth's sphere of influence for both the patched-conic approximation and the restriced four-body solution. We have already seen the development of a Runge-Kutta integrator for the restricted four-body problem. Using the same process, we develop a Runge-Kutta integrator to integrate the spacecraft's departure orbit as a simpler two-body problem. The Appendix offers the MatLab code used to perform the two-body integration. Having both the four-body and two-body integrators, we can now compare the spacecraft's motion for the two cases. We set the initial parking orbit radius and the elliptic semi-major axis to be 7500 km and 1.8877e+008 km, respectively. These initial conditions match those used in Chapter 2 to apply the patched-conic approximation to the Hohmann transfer from Earth to Mars. In addition, we set the time step g_h to 50 seconds for the transfer orbit. Figure 4.1 illustrates the integrated solutions for both the two-body and four-body problems. As shown by Figure 4.1, the patched-conic approximation basically matches the spacecraft motion under the restricted four-body problem. In both cases, the spacecraft leaves Earth's sphere of influence with a heliocentric velocity in the same direction as that of the Earth. Thus, the results support the patched-conic prediction of a departure velocity $\boldsymbol{v_1}$ parallel to the Earth's heliocentric velocity $\boldsymbol{v}_{\oplus}$.

Using the same initial conditions ($r_0 = 1500$ km, $a_e = 1.8877\text{e}{+}008$ km), we now examine the spacecraft's motion during the entire Hohmann transfer. With a step size g_h of 50 seconds, we use the Runge-Kutta integrator to calculate the spacecraft's state over the transfer orbit. Figure 4.2 displays a plot of the integrated orbit from Earth to Mars. The patched-conic approximation predicts that the spacecraft's trajectory will be an perfect ellipse with a semi-major axis a_e equal to 1.8877e+008 km. As shown by Figure 4.2, this



(a) Motion through Earth's entire sphere of influence.

(b) Close-up of the first stages of the departure.

Figure 4.1: Comparison of the patched-conic and restricted four-body predictions of the spacecraft's motion through Earth's sphere of influence. Positions x and y are relative to the center of the Earth frame $\mathcal{E}$.

is a good prediction when we analyze the motion on heliocentric orders of magnitude. The spacecraft's transfer orbit does indeed appear to be almost a perfect ellipse. At this point, it must be noted that the actual semi-major axis of the transfer will be slightly different from that used to graph Figure 4.2. The difference comes from the fact that the spacecraft performs the departure burn at some offset distance from the center of the Earth. The patched-conic approximation ignores this minute detail, but we will study its effects upon the arrival orbit in a later section.

We are also interested in the spacecraft's motion during its arrival orbit through Mars' sphere of influence. Thus, using the same initial conditions and the same integrator, we plot the spacecraft's motion relative to Mars. Figure 4.3 displays the planet-centric trajectory of the spacecraft. As Figure 4.3 shows, the Hohmann transfer enters Mars' sphere of influence at a Mars-centric heading angle ϑ_2 that is very close to 180 degrees. If the spacecraft were on a perfect Hohmann transfer, the ϑ_2 heading angle would be exactly equal to 180 degrees, given its definition in Figure 2.4. It is reassuring to know that the restricted four-body problem yields a transfer orbit that does enter the Martian sphere of influence. Further, the heading angle upon entry is very similar to that of a perfect Hohmann transfer. As predicted by the patched-conic approximation, the spacecraft enters the sphere of influence from the front door. In other words, Mars intercepts the spacecraft at the end of the transfer.

Perhaps the most important feature of Figure 4.3 is the similarity between the two graphs. Halving the planet-centric time step g_h from 50 to 25 seconds has roughly no effect on the spacecraft's arrival orbit. Therefore, we can set the planet-centric time step to be at least 50 seconds without sacrificing significant accuracy. For the orbits shown in Figure 4.3, the heliocentric time step g_e is maintained at 50000 seconds.

While the spacecraft does stay within Mars' sphere of influence, it also overshoots the planet by roughly 4e+005 kilometers. At first, such a result would seem counter-intuitive

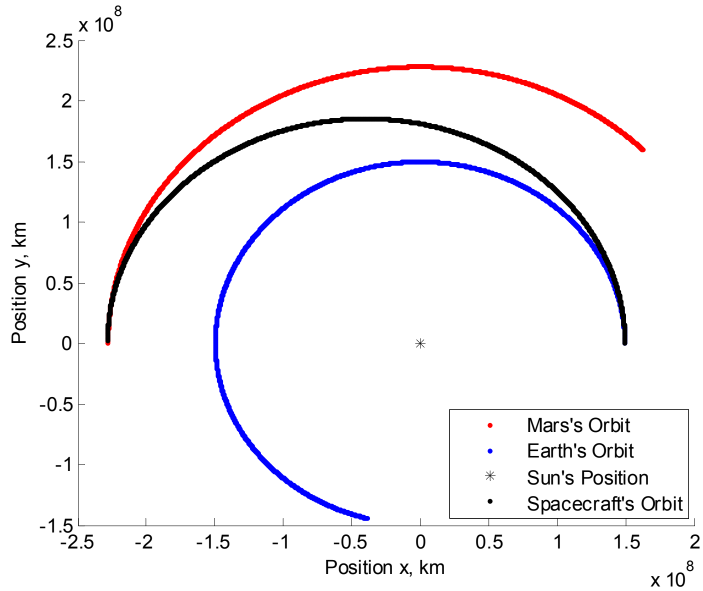
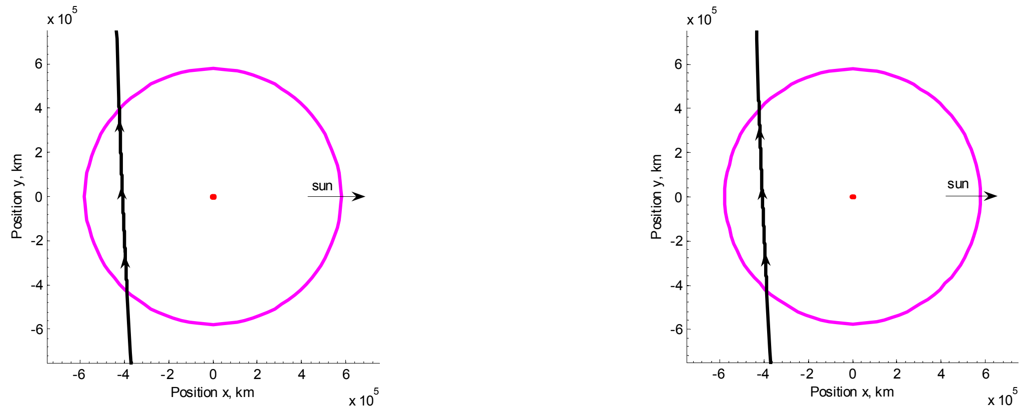


Figure 4.2: Illustration of the spacecraft's heliocentric motion during the entire Hohmann transfer. Step size used with the four-body integrator is $g_h = 50$ seconds.

The spacecraft actually begins its orbit slightly farther away from the sun than if it were to begin a perfect Hohmann transfer directly from Earth's surface. However, as the spacecraft's initial position shifts slightly farther away from the sun, we do not alter the calculation of the initial escape velocity $\boldsymbol{\nu}_0$. When we view the problem from the heliocentric point of view, we recognize that the spacecraft does not require as much initial speed when beginning the orbit slightly farther away from the sun. Such a statement can be proved by applying the conservation of angular momentum to the elliptic heliocentric orbit. Therefore, because we keep the same $\boldsymbol{\nu}_0$ at a larger distance from the sun, the spacecraft is expected to overshoot Mars by some miss distance d_m.

In many respects, the four-body integration supports the patched-conic approximations given in Chapter 2. The hyperbolic departure orbit is identical for the two- and four-body scenarios. The integrated Hohmann transfer to Mars' sphere of influence closely matches a perfect ellipse. Perhaps most importantly, the spacecraft penetrates Mars' sphere of influence in same fashion predicted by the patched-conic approximation. The results support the validity of using the patched-conic approximation as a rough estimate of the $\Delta \boldsymbol{v}_1$ needed to perform the Hohmann transfer. Having compared the two- and four-body problems as well as their results, we now analyze the effect of changing certain initial conditions of the transfer orbit. Of particular interest is how the arrival orbit is altered due to the changes in initial conditions.



(a) $g_h = 50$ sec. (b) $g_h = 25$ sec.

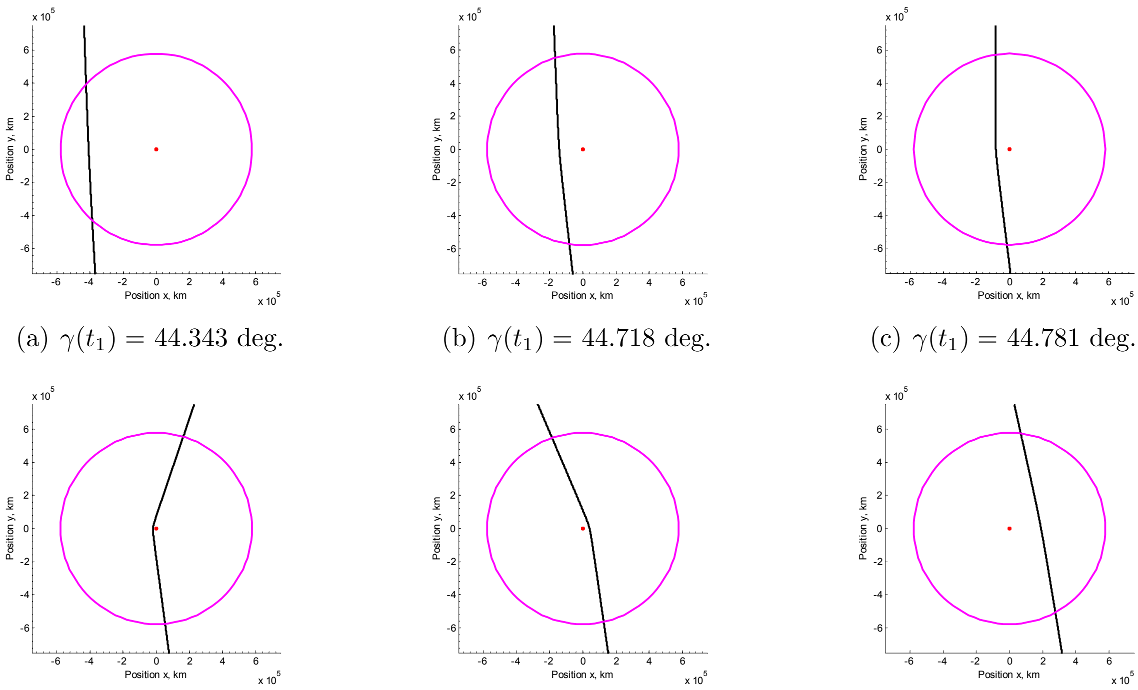
Figure 4.3: Illustration of the spacecraft's arrival orbit through Mars' sphere of influence.

4.2 Altering the Initial Conditions

4.2.1 Changing the Mars Offset Angle

As discussed in Chapter 2, there must exist some initial offset angle $\gamma(t_1)$ between Earth and Mars. If there were no initial offset angle, the spacecraft would perform the Hohmann transfer without ever entering Mars' sphere of influence. Up to this point, we have performed all numerical integrations with the initial offset angle computed in Chapter 2 (44.343 degrees). By changing this initial offset angle, we can examine the effect that it has upon the hyperbolic arrival orbit. Thus, we perform a series of restricted four-body integrations, varying this offset angle $\gamma(t_1)$. Figure 4.4 displays a group of arrival orbits for six different Mars offset angles. The planet-centric step size used to integrate the six different cases is 50 seconds. At this point we are most concerned with the relative geometries of the orbits depending upon the initial Mars offset angle. As shown by Figure 4.3, a step size g_h of 50 seconds is small enough to accurately give us the relative geometries. Figure 4.4 illustrates that increasing the initial offset angle $\gamma(t_1)$ affects the hyperbolic arrival orbit in two ways. It noticeably varies the miss distance d_m between the spacecraft's projected trajectory and the sun direction. As $\gamma(t_1)$ is increased from 44.343 to 44.843 degrees, the miss distance decreases and the eccentricity of the hyperbolic arrival increases. But for all of these orbit geometries, the spacecraft orbits Mars in a clockwise fasion. Once $\gamma(t_1)$ surpasses 44.843 degrees, the spacecraft begins performing counter-clockwise orbits about Mars. This effect is particularly important if we want to ultimately have a geostationary orbit about Mars. In such a case, we would need to be orbiting Mars in the same direction as the planetary rotation.

Secondly, the changes in initial offset angle $\gamma(t_1)$ have a slight effect upon the arrival heading angle $\sigma_2 + \vartheta_2$. Note that, as the offset angle is increased from 44.343 to 45.031 degrees, the heading angle decreases from its initial value of roughly 180 degrees. The reason for this slight decrease in heading angle is that the spacecraft is now penetrating



(a) $\gamma(t_1) = 44.343$ deg. (b) $\gamma(t_1) = 44.718$ deg. (c) $\gamma(t_1) = 44.781$ deg.

(d) $\gamma(t_1) = 44.843$ deg. (e) $\gamma(t_1) = 44.906$ deg. (f) $\gamma(t_1) = 45.031$ deg.

Figure 4.4: Series of hyperbolic arrival orbits corresponding to six different initial offset angles between Earth and Mars. The x and y positions are taken relative to the Mars-centered frame $\mathcal{M}$. The step size g_h used is 50 seconds.

Mars' sphere of influence at an earlier time on its Hohmann transfer. Thus, the heading angle begins to regress from the ideal value of 180 degrees for a perfect Hohmann transfer.

4.2.2 Changing Mars' Heliocentric Orbit Radius

As noted in Section 4.1, when we use the patched-conic approximation to estimate the necessary initial conditions for the Hohmann transfer, the arrival orbit overshoots Mars by roughly 4e+005 kilometers. Therefore, if we want to achieve a certain hyperbolic periapses radius r_3 about Mars, we must alter at least one initial condition. Referring to the patched-conic arrival orbit solutions presented in Section 2.4, we find that the r_3 parking radius depends upon the miss distance d_m and the velocity $\boldsymbol{\nu}_2$. Equation 2.24 gives the relation between the actual miss distance d_m and the perpendicular miss distance d_a. The planar Hohmann transfer from Earth to Mars will always yield a planet-centric velocity ν_2 roughly equal to 2.648 km/s, as calculated in Section 2.4. Thus, to achieve a specific parking orbit radius about Mars, we can alter the miss distance d_a until the necessary arrival geometry is obtained. One way to alter the miss distance d_a of the arrival hyperbola is to make small changes in Mars' heliocentric orbit radius. For instance, if the spacecraft overshoots Mars by too great a distance, we subtract the extra miss distance from Mars' orbit radius and iterate the same Hohmann transfer. Using such a method, we can determine what Martian heliocentric orbit radius will yield the miss distance d_a corresponding to our desired parking radius r_3. The Appendix offers a listing of the Matlab code used to perform such an iteration.

In addition, Figure 4.5 offers a flow chart illustrating the $r_{♂}$ correction process.

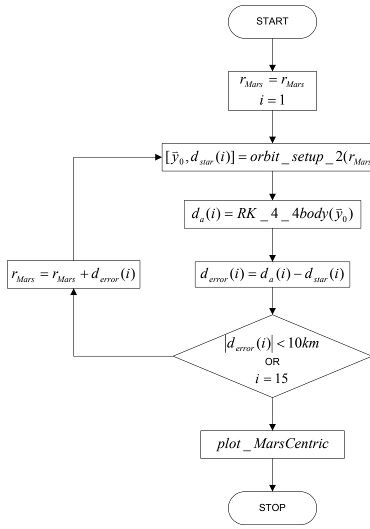
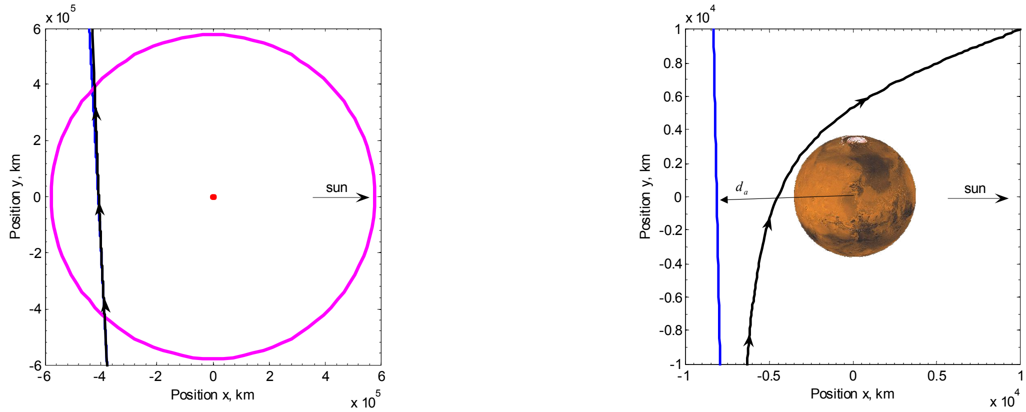


Figure 4.5: Flow chart depicting the loop used to iteratively correct Mars' orbit radius $r_{♂}$ in order to achieve the desired arrival parking radius r_3 about Mars.

The advantage of using a variable time step is accentuated when we perform the given iteration to determine a unique arrival orbit geometry. In performing the iteration, we are integrating the Hohmann transfer a number of times in order to analyze changes in the arrival hyperbola. Thus, being able to quickly integrate the heliocentric portion of each Hohmann transfer is a valuable asset. Figure 4.6 offers an illustration of both the corrected and uncorrected arrival orbit geometries. For the iterations performed, we set the desired Mars parking radius r_3 to 4000 km. The initial iteration yields a miss distance of roughly 4e+005 kilometers. But after seven iterations are performed, the miss distance is almost exactly equal to the necessary value d_{star} of 8142 kilometers. Both graphs of Figure 4.6 show the projection of the $\boldsymbol{\nu}_2$ velocity upon entry into Mars' sphere of influence as a blue line. We use this projection to then calculate the perpendicular distance to the center of Mars, corresponding to the actual miss distance d_a. Table 4.1 offers a listing of the actual distance d_a, necessary distance d_{star}, and distance error d_{error} for each iteration.

Note that by the seventh iteration, the magnitude of d_{error} has dropped below 1 kilometer. Figure 4.6 illustrates how the seventh iteration yields an arrival orbit with a periapses radius r_3 of roughly 4000 kilometers. Thus, we have taken the restricted four-body problem and found one set of initial conditions that result in a desired final parking orbit radius about Mars. Table 4.1 also offers a listing of the necessary miss distance d_{star} values for each iteration. We iterate the necessary miss distance value because the arrival speed ν_2 is altered slightly for each new value of the Mars orbit radius. Because the changes in ν_2 for each iteration are so small, the magnitude of d_{star} changes only slightly. By the fifth iteration, the value of d_{star} has already reached its approximate final value of -8.1419e+003 km. Figure 4.7 offers a plot of the miss distance error magnitude for the Mars orbit radius correction process. After three iterations, most of the correction to Mars' orbit radius $r_{♂}$ has already



(a) $r_{\male}$ uncorrected

(b) $r_{\male}$ corrected

Figure 4.6: Illustration of both the uncorrected and corrected arrival orbit geometries for the Mars orbit radius iteration. Values x and y are defined relative to the non-rotating Mars frame $\mathcal{M}$. Seven iterations were performed before achieving the final corrected arrival.

Iteration	d_a, km	d_{star}, km	d_{error}, km
1	-4.09519e+005	-8.05389e+003	-4.01465e+005
2	-9.34220e+004	-8.12439e+003	-8.52976e+004
3	-1.95337e+004	-8.13963e+003	-1.13941e+004
4	-9.49786e+003	-8.14167e+003	-1.35619e+003
5	-8.30000e+003	-8.14192e+003	-1.58085e+002
6	-8.16003e+003	-8.14194e+003	-1.80811e+001
7	-8.14283e+003	-8.14195e+003	-8.83155e-001

Table 4.1: Table of miss distance values calculated during each iteration of the Mars orbit radius correction scheme.

been made. Between iterations three and seven, much smaller corrections are made to Mars' orbit radius, and the corrections in the error d_{error} are therefore also much less.

Our iteration yields an orbit with a periapses radius about Mars roughly equal to the desired value of 4000 km. But the spacecraft rotates about Mars in a retrograde fashion. If we desire the same periapses radius, but corresponding to a posigrade rotation about Mars, we would need to change the necessary miss distance d_{star} to a positive value. However, in doing so, we would need to alter the first correction of Mars' orbit radius to be sure that the second iteration does not yield an orbit that strikes Mars' surface. This singularity would significantly change the results of the iterative process. One suggestion for iterating to yield a posigrade orbit is to over-correct the first Mars orbit radius. Once the $r_{\male}$ orbit radius has been over-corrected, we can then iterate using the method explained in this section to yield the desired periapses radius r_3.

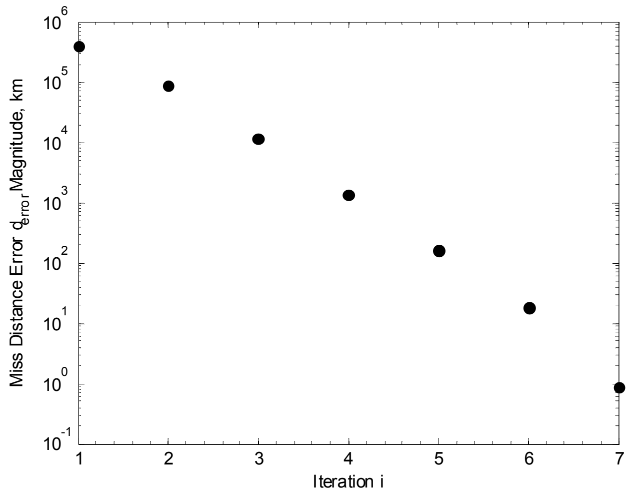


Figure 4.7: Graph of the miss distance error magnitude d_{error} for each step of the Mars orbit radius correction scheme.

4.3 Comparison of Predicted Δv Values

So far, we have analyzed three different ways to estimate the necessary Δv value to travel from Earth to Mars on a Hohmann transfer. We first view the transfer orbit as a single elliptic orbit with a change in true anomaly of 180 degrees. Such an approximation treats the sun as the only gravitational influence upon the spacecraft during the transfer. The gravitational effects of both Earth and Mars are ignored entirely.

Our second representation of the Hohmann transfer is as a series of two-body orbits about Earth, the sun, and Mars, respectively. Because we represent each portion of the orbit as a conic solution, we term this solution the patched-conic approximation. The patched-conic approximation allows us to take into account the gravity of Earth and Mars as the spacecraft travels through the planets' spheres of influence. However, this approximation ignores the gravitational effects of the planets when the spacecraft is traveling outside of their spheres of influence. The patched-conic approximation yields a better estimate of the Δv required to reach Mars than the simple Hohmann solution. This better estimate results from taking into account the gravitational influence of Earth as the spacecraft performs its hyperbolic departure orbit.

The final representation of the transfer orbit is as a restricted four-body orbit. Thus, we take the gravity of Earth, the sun, and Mars into consideration for the duration of the entire transfer orbit. We also examine the effects of altering certain departure orbit conditions upon the nature of the arrival orbit. More specifically, we determine the required Δv to achieve a particular periapses radius r_3 about Mars. Such a calculation cannot be made when examining the orbit using either the Hohmann approximation or the patched-conic approximation. Table 4.2 provides a listing of the Δv estimates corresponding to each of the three Hohmann transfer representations.

Orbit Approximation	Δv (km/s)
Hohmann Transfer	2.943
Patched-Conic	3.432
Restricted Four-Body	3.428

Table 4.2: Table showing the differences in required Δv estimates for the Hohmann transfer, patched-conic approximation, and restricted four-body problem. The value of Δv for the four-body approximation corresponds to a desired Mars parking orbit radius r_3 of 4000 km.

Note that the difference in required Δv values lies mostly in going from the general Hohmann approximation to the patched-conic approximation. There exists only a 0.004 km/s difference in the Δv values for the patched-conic approximation and the restricted four-body problem. However, integrating the restricted four-body problem allows us to confirm that this seemingly minute difference allows us to achieve a desired Mars periapses radius of 4000 km. Such minute details become extremely important when attempting to establish a true interplanetary mission plan.

Chapter 5

Conclusion

The original analytic solution to the Hohmann transfer from Earth to Mars offers a crude estimate of the Δv required to perform the transfer. Because it neglects the gravitational effects of both Earth and Mars, this orbit solution cannot achieve the same accuracy as the patched-conic approximation. However, this simple orbit representation does provide a suitable rough estimate of the initial burn required to reach Mars' sphere of influence.

The patched-conic approximation provides a much better estimate of the Δv required to reach Mars on a Hohmann transfer. Its consideration of the planets' gravitational influences as the spacecraft travels through their spheres of influence makes this solution much more credible than the simple Hohmann solution. By breaking the entire orbit into three separate conic solutions, we can begin to see the effects of the departure orbit geometry on both the elliptic transfer and hyperbolic arrival. However, the patched-conic approximation does not allow us to alter certain departure orbit conditions and see the direct effect upon the arrival. Instead, we much solve each of the three conic solutions as separate orbits.

The restricted four-body integration scheme allows us to view the Hohmann transfer from Earth to Mars as one entire orbit. Thus, while taking into consideration the gravity of Earth, the sun, and Mars for all time, we can analyze the effects of altering certain initial conditions upon the arrival orbit. In addition, we can determine the necessary departure burn to achieve a desired parking orbit radius r_3 about Mars. The patched-conic approximation does not allow for such precise orbit modeling. Treating the Hohmann transfer as a restricted four-body problem yields an even higher fidelity representation of the transfer orbit.

One idea for future work is to examine the applicability of the established four-body integrator to other interplanetary missions. Such missions need not necessarily be Hohmann transfers. They could also lead to arrivals at a different planet from Mars. The sensitivity of the four-body integrator to perturbations of these different orbits could then be analyzed. Still other future work could focus on increasing the accuracy of the presented four-body orbit modeling scheme. For instance, atmospheric drag is a disturbance that must be considered for both the departure and arrival orbits. In addition, the planar orbit assumption could be dropped by taking into account the slight orbit inclination difference between Earth and Mars. The spacecraft would then have a full three-dimensional state vector to be integrated over the course of the transfer. In short, much work remains in yielding an orbit modeling scheme that presents what would actually occur in a real-time transfer from Earth to Mars.

Appendix A

MatLab Two-Body Code

A.1 two_body.m

```
% Integrate the equations of motion of a satellite orbiting about one
% celestial body, ignoring the gravitational effects of all other celestial
% bodies

% Thomas Reppert 04/30/06

ti =             0; % initial time, s
tf = 2.2376*10^(5); %   final time, s
h  =            50; %    time step, s

% compute the initial satellite state
% note: in orbit_setup.m, must taylor the calculations of the initial state
% vector components to the desired two-body problem
y0 = orbit_setup;

% use the Fourth-Order Runge-Kutta integrator to compute the satellite's
% state from the initial time ti to the final time tf
[t,state] = RK_4_2body(ti, tf, h, y0);

% plot the satellite's motion relative to the celestial body
plot_2body(state)
```

A.2 RK_4_2body.m

```
function [t,y] = RK_4_2body(ti, tf, h, y0)
% RK_4_2body:
% uses the Fourth-Order Runge-Kutta technique to integrate the equations of
% motion of a satellite in a planar orbit about one celestial body,
% ignoring the gravitational effects of all other celestial bodies
```

```
% input:
% initial time ti
% final time tf
% time step h
% initial state y0

% output:
% integrated satellite state vector y

% Thomas Reppert 04/30/06

t = ti:h:tf; % initialize the time vector, s
n = length(t); % compute the length of the time vector
m = length(y0); % compute the length of the initial state vector

y = zeros(m,n); % preallocate the state vector y
y(:,1) = y0'; % initialize the first column of y

for i = 1:n-1
    % estimate slope at t and assign to k1
    k1 = dydt_2body(t(i), y(:,i));

    % assign (y + k1*h/2) value to temporary state y_temp
    y_temp = y(:,i) + 0.5*k1*h;

    % use the new y_temp to estimate slope at (t + h/2) as k2
    k2 = dydt_2body(t(i)+0.5*h, y_temp);

    % assign (y + k2*h/2) value to temporary state y_temp
    y_temp = y(:,i) + 0.5*k2*h;

    % use the new y_temp to estimate slope at (t + h/2) as k3
    k3 = dydt_2body(t(i)+0.5*h, y_temp);

    % assign (y + k3*h) value to temporary state y_temp
    y_temp = y(:,i) + k3*h;

    % use the new y_temp to estimate slope at (t + h) as k4
    k4 = dydt_2body(t(i)+h, y_temp);

    % add current state y(i) to slope_avg*h for the new state y(i+1)
    y(:,i+1) = y(:,i) + (k1+2*k2+2*k3+k4)/6*h;
end
```

A.3 dydt_2body.m

```
function slope_est = dydt_2body(t,y)
% dydt_2body:
% calculates the value of each slope estimate k for the Fourth-Order
% Runge-Kutta integration

% Thomas Reppert 04/30/06

mu_earth = 3.986*10^(5); % Earth's gravitational coefficient, km^3/s^2

% compute the current distance between the satellite and the celestial
% body, km
r_sat = sqrt(y(1)^2 + y(2)^2 + y(3)^2);

% compute the coefficient matrix used to calculate the dydt slope estimate
A = [
 0.0,                 0.0,  0.0, 1.0, 0.0, 0.0;
 0.0,                 0.0,  0.0, 0.0, 1.0, 0.0;
 0.0,                 0.0,  0.0, 0.0, 0.0, 1.0;
-mu_earth/r_sat^(3),  0.0,  0.0, 0.0, 0.0, 0.0;
 0.0, -mu_earth/r_sat^(3),  0.0, 0.0, 0.0, 0.0;
 0.0,  0.0, -mu_earth/r_sat^(3), 0.0, 0.0, 0.0];

% calculate the dydt slope estimate (with 6 components)
slope_est = A*y;
```

Appendix B

MatLab Four-Body Code

B.1 four_body.m

```
% Integrate the equations of motion of a satellite on a Hohmann transfer
% from Earth to Mars, taking into consideration the gravity of Earth, Mars,
% and the sun for all time t

% Thomas Reppert 04/23/06

ti =            0; % initial time, s
tf = 2.2376*10^(7); %   final time, s

% compute the initial satellite state
y0 = orbit_setup;

% use the Fourth-Order Runge-Kutta integrator to compute the satellite's
% state from the initial time ti to the final time tf
[t, state, SvsE, SvsM] = RK_4_4body(ti, tf, y0);

% plot the satellite's motion relative to the sun
plot_HelioCentric(state)

% plot the satellite's motion relative to Earth
plot_EarthCentric(SvsE)

% plot the satellite's motion relative to Mars
plot_MarsCentric(SvsM)
```

B.2 orbit_setup.m

```
% Compute the initial heliocentric position and velocity of a satellite on
% a Hohmann transfer from Earth to Mars.
```

```
% Thomas Reppert 04/23/06

function y0 = orbit_setup

mu_sun = 1.326*10^(11); % sun's gravitational coefficient, km^3/s^2
mu_earth = 3.986*10^(5); % Earth's gravitational coefficient, km^3/s^2
r_earth = 1.496*10^(8); % Earth's mean orbit radius, km
r_mars = 2.2794*10^(8); % Mars' mean orbit radius, km

r_0 = 7500; % initial parking orbit radius, km

c = r_earth + r_mars; % Hohmann transfer chord length, km

v_1 = sqrt(2*mu_sun/c*(r_mars/r_earth)); % spacecraft's velocity when ...

n_earth = sqrt(mu_sun/r_earth^(3)); % Earth's mean orbit rate, rad/s

v_earth = r_earth*n_earth; % Earth's heliocentric velocity, km/s

nu_1 = v_1 - v_earth; % spacecraft's Earth-centric departure velocity ...

a = -mu_earth/nu_1^2; % hyperbolic semi-major axis, km

nu_0 = sqrt(2*mu_earth/r_0 - mu_earth/a); % spacecraft's Earth-centric ...

nu_c = sqrt(mu_earth/r_0); % initial Earth-centric critical velocity, km/s

e = r_0*nu_0^2/mu_earth - 1; % departure orbit eccentricity

Phi = acos(1/e) + pi; % initial burn angle Phi, rad

% initial satellite state vector
% -----------------------------

% heliocentric position, km
x_0 = r_earth + r_0*cos(3*pi/2-Phi);
y_0 =          -r_0*sin(3*pi/2-Phi);
z_0 =                             0;

% heliocentric velocity, km/s
xdot_0 =           nu_0*cos(Phi-pi);
ydot_0 = v_earth + nu_0*sin(Phi-pi);
zdot_0 =                          0;
```

```
y0 = [x_0, y_0, z_0, xdot_0, ydot_0, zdot_0];
```

B.3 RK_4_4body.m

```
function [t, y, SvsE, SvsM, d_a] = RK_4_4body(ti, tf, y0)
% RK_4_4body:
% uses the Fourth-Order Runge-Kutta technique to integrate the equations of
% motion of a satellite in a planar Hohmann transfer from Earth to Mars
% (type dydt = f(t,y)), taking into consideration the gravity of the sun,
% Earth, and Mars for all time t

% input:
% initial time ti
% final time tf
% initial state y0

% output:
% integrated heliocentric state vector y
% integrated Earth-centric position SvsE
% integrated Mars-centric position SvsM
% arrival orbit miss distance d_a, km

% Thomas Reppert 04/23/06

t = zeros(1,20000); % preallocate the time vector t
y = zeros(6,20000); % preallocate the state vector y

t(1) = ti;    % set the initial time
y(:,1) = y0'; % set the initial state

% calculate the offset angle between Earth and Mars
mu_sun = 1.326*10^(11); % sun's gravitational coefficient, km^3/s^2
r_earth = 1.496*10^(8); % Earth's mean orbit radius, km
r_mars = 2.2794*10^(8); % Mars' mean orbit radius, km
n_mars = sqrt(mu_sun/r_mars^(3)); % Mars' mean orbit rate, rad/s
a = (r_earth + r_mars)/2; % semi-major axis of the Hohmann transfer, km
P = 2*pi*sqrt(a^(3)/mu_sun); % period of the Hohmann transfer, s
offset = pi - n_mars*(P/2); % offset angle between Earth and Mars, rad

SvsE = zeros(4,20000); % preallocate the satellite's position vector ...
SvsM = zeros(4,20000); % preallocate the satellite's position vector ...
SvsE(:,1) = SatvsEarth(t(1), y(:,1));        % compute the initial ...
SvsM(:,1) = SatvsMars(t(1), y(:,1), offset); % compute the initial ...
```

```
% perform the Runge-Kutta integration
% -----------------------------------

rE_SOI = 916600; % Earth's SOI radius, km
rM_SOI = 577400; % Mars' SOI radius, km

i = 1; % initialize the integration counter i

while(1)

    % loop while the current time t(i) is still less than the final time tf
    if t(i) >= tf
        y = y(:,1:i);
        SvsE = SvsE(:,1:i);
        SvsM = SvsM(:,1:i);
        break
    end

    % use a variable time step to speed up the integration: increase the
    % time step when the satellite's position wrt both Earth and Mars is
    % greater than 1.5*r_SOI of both Earth and Mars, respectively
    if (norm(SvsE(:,i)) > 1.5*rE_SOI) & (norm(SvsM(:,i)) > 1.5*rM_SOI)
        h = 50000;
        t(i+1) = t(i) + 50000;
    else
        h = 50;
        t(i+1) = t(i) + 50;
    end

    % estimate slope at t and assign to k1
    k1 = dydt_4body(t(i), y(:,i), offset);

    % assign (y + k1*h/2) value to temporary state y_temp
    y_temp = y(:,i) + 0.5*k1*h;

    % use the new y_temp to estimate slope at (t + h/2) as k2
    k2 = dydt_4body(t(i)+0.5*h, y_temp, offset);

    % assign (y + k2*h/2) value to temporary state y_temp
    y_temp = y(:,i) + 0.5*k2*h;

    % use the new y_temp to estimate slope at (t + h/2) as k3
    k3 = dydt_4body(t(i)+0.5*h, y_temp, offset);

    % assign (y + k3*h) value to temporary state y_temp
```

```
    y_temp = y(:,i) + k3*h;

    % use the new y_temp to estimate slope at (t + h) as k4
    k4 = dydt_4body(t(i)+h, y_temp, offset);

    % add current state y(i) to slope_avg*h for the new state y(i+1)
    y(:,i+1) = y(:,i) + (k1+2*k2+2*k3+k4)/6*h;

    % compute the current satellite position wrt both Earth and Mars
    SvsE(:,i+1) = SatvsEarth(t(i+1), y(:,i+1));
    SvsM(:,i+1) = SatvsMars(t(i+1), y(:,i+1), offset);

    % upon entry into Mars' sphere of influence, plot and calculate the
    % miss distance d_a, km
    if (SvsM(4,i) >= rM_SOI) & (SvsM(4,i+1) < rM_SOI)
        d_a = MissDistance(i, SvsM);
    end

    % increment the integration counter i
    i = i + 1;
end
```

B.4 SatvsEarth.m

```
function R_2 = SatvsEarth(t, y)
% SatvsEarth:
% computes the satellite's position with respect to Earth at the specified
% time t during the Hohmann transfer

% input:
% current time t
% current satellite heliocentric state y

% output:
% satellite's position with respect to Earth R_2

% Thomas Reppert 04/23/06

E = earth_pos(t); % compute Earth's heliocentric position vector at the ...

% satellite's position wrt Earth, km
% ----------------------------------

x_2 = y(1) - E(1);
```

```
y_2 = y(2) - E(2);
z_2 = y(3) - E(3);
r_2 = sqrt(x_2^(2) + y_2^(2) + z_2^(2));
R_2 = [x_2, y_2, z_2, r_2]';
```

B.5 SatvsMars.m

```
function R_3 = SatvsMars(t, y, offset)
% SatvsMars:
% computes the satellite's position with respect to Mars at the specified
% time t during the Hohmann transfer

% input:
% current time t
% current satellite heliocentric state y

% output:
% satellite's position with respect to Mars R_3

% Thomas Reppert 04/23/06

M = mars_pos(t, offset); % compute Mars' heliocentric position vector ...

% satellite's position wrt Mars, km
% ---------------------------------

x_3 = y(1) - M(1);
y_3 = y(2) - M(2);
z_3 = y(3) - M(3);
r_3 = sqrt(x_3^(2) + y_3^(2) + z_3^(2));
R_3 = [x_3, y_3, z_3, r_3]';
```

B.6 dydt_4body.m

```
function slope_est = dydt_4body(t, y, offset)
% dydt_4body:
% calculates the value of each slope estimate k for the Fourth-Order
% Runge-Kutta integration

% Thomas Reppert 04/23/06

mu_sun  = 1.326*10^(11); % sun's gravitational coefficient, km^3/s^2
mu_earth = 3.986*10^(5); % Earth's gravitational coefficient, km^3/s^2
```

```
mu_mars  = 4.282*10^(4); % Mars' gravitational coefficient, km^3/s^2

E =        earth_pos(t); % compute Earth's heliocentric position vector ...
M = mars_pos(t, offset); % compute Mars' heliocentric position vector ...

% satellite's position wrt the sun r_1, km
% ---------------------------------------

x_1 = y(1);
y_1 = y(2);
z_1 = y(3);
r_1 = sqrt(x_1^(2)+ y_1^(2) + z_1^(2));

% satellite's position wrt Earth r_2, km
% --------------------------------------

x_2 = y(1) - E(1);
y_2 = y(2) - E(2);
z_2 = y(3) - E(3);
r_2 = sqrt(x_2^(2) + y_2^(2) + z_2^(2));

% satellite's position wrt Mars r_3, km
% -------------------------------------

x_3 = y(1) - M(1);
y_3 = y(2) - M(2);
z_3 = y(3) - M(3);
r_3 = sqrt(x_3^(2) + y_3^(2) + z_3^(2));

% calculate the dydt slope estimate components 1:6
% ------------------------------------------------

SE(1) = y(4);
SE(2) = y(5);
SE(3) = y(6);

SE(4) = -mu_sun/r_1^(3)*x_1 - mu_earth/r_2^(3)*x_2 - mu_mars/r_3^(3)*x_3;
SE(5) = -mu_sun/r_1^(3)*y_1 - mu_earth/r_2^(3)*y_2 - mu_mars/r_3^(3)*y_3;
SE(6) = -mu_sun/r_1^(3)*z_1 - mu_earth/r_2^(3)*z_2 - mu_mars/r_3^(3)*z_3;

slope_est = [SE(1), SE(2), SE(3), SE(4), SE(5), SE(6)]';
```

B.7 earth_pos.m

```
function pos_vec = earth_pos(t)
% earth_pos:
% calculates Earth's position vector at the specified time t for a
% circular orbit about the sun

% input:
% current time t

% output:
% Earth's heliocentric position vector pos_vec

% Thomas Reppert 04/23/06

mu_sun  = 1.326*10^(11); % sun's gravitational coefficient, km^3/s^2
r_earth = 1.496*10^(8); % Earth's mean orbit radius, km
n_earth = sqrt(mu_sun/r_earth^(3)); % Earth's mean orbit rate, rad/s

% Earth's current heliocentric position vector, km
% ------------------------------------------------

pos_vec(1) = r_earth*cos(n_earth*t);
pos_vec(2) = r_earth*sin(n_earth*t);
pos_vec(3) = 0;
```

B.8 mars_pos.m

```
function pos_vec = mars_pos(t, offset)
% mars_pos:
% calculates Mars' position vector at the specified time t for a
% circular orbit about the sun

% input:
% current time t

% output:
% Mars' heliocentric position vector pos_vec

% Thomas Reppert 04/23/06

mu_sun  = 1.326*10^(11); % sun's gravitational coefficient, km^3/s^2
r_mars = 2.2794*10^(8); % Mars' mean orbit radius, km
n_mars = sqrt(mu_sun/r_mars^(3)); % Mars' mean orbit rate, rad/s
```

```
% Mars' current heliocentric position vector, km
% ----------------------------------------------

pos_vec(1) = r_mars*cos(n_mars*t + offset);
pos_vec(2) = r_mars*sin(n_mars*t + offset);
pos_vec(3) = 0;
```

B.9 plot_HelioCentric.m

```
function plot_HelioCentric(helio)
% plot_HelioCentric:
% plots the satellite's motion relative to the sun

% input: satellite's integrated heliocentric state vector helio

% Thomas Reppert 04/23/06

% plot the sun's position
plot3(0,0,0,'k*')
view(0,90)
hold on
axis equal

% plot the satellite's integrated heliocentric motion
x = helio(1,:);
y = helio(2,:);
z = helio(3,:);
plot3(x,y,z,'k.')
xlabel('Position x, km')
ylabel('Position y, km')
```

B.10 plot_EarthCentric.m

```
function plot_EarthCentric(SvsE)
% plot_EarthCentric:
% plots the satellite's motion relative to Earth

% input: satellite's integrated Earth-centric state vector SvsE

% Thomas Reppert 04/23/06

rE = 6378.14; % Earth's mean radius, km
```

```
rE_SOI = 916600; % Earth's mean SOI radius, km
theta = 0:pi/50:2*pi; % angular plotting parameter for rE, rE_SOI

% plot the surface of Earth
xE = rE.*cos(theta);
yE = rE.*sin(theta);
zE = zeros(1,101);
plot3(xE, yE, zE,'b-')
view(0,90)
hold on
axis equal

% plot Earth's sphere of influence
xE_SOI = rE_SOI.*cos(theta);
yE_SOI = rE_SOI.*sin(theta);
zE_SOI = zeros(1,101);
plot3(xE_SOI, yE_SOI, zE_SOI,'m-')

% plot the satellite's integrated Earth-centric motion
plot3(SvsE(1,:),SvsE(2,:),SvsE(3,:),'k-')
xlabel('Position x, km')
ylabel('Position y, km')
```

B.11 plot_MarsCentric.m

```
function plot_MarsCentric(SvsM)
% plot_MarsCentric:
% plots the satellite's motion relative to Mars

% input: satellite's integrated Mars-centric state vector SvsM

% Thomas Reppert 04/23/06

rM = 3400; % Mars' mean radius, km
rM_SOI = 577400; % Mars' mean SOI radius, km
theta = 0:pi/50:2*pi; % angular plotting parameter for rM, rM_SOI

% plot the surface of Mars
xM = rM.*cos(theta);
yM = rM.*sin(theta);
zM = zeros(1,101);
plot3(xM, yM, zM,'r-')
view(0,90)
hold on
```

```
axis equal

% plot Mars' sphere of influence
xM_SOI = rM_SOI.*cos(theta);
yM_SOI = rM_SOI.*sin(theta);
zM_SOI = zeros(1,101);
plot3(xM_SOI, yM_SOI, zM_SOI,'m-')

% plot the satellite's integrated Mars-centric motion
plot3(SvsM(1,:),SvsM(2,:),SvsM(3,:),'k-')
xlabel('Position x, km')
ylabel('Position y, km')
```

Appendix C

MatLab Miss Distance Code

C.1 set_r_3.m

```
% Set a desired r_3 parking orbit radius about Mars.  Then iterate the
% Hohmann transfer from Earth to Mars, computing for each iteration the
% necessary change in Mars' orbit radius r_2 to achieve the desired r_3.

% Thomas Reppert 04/24/06

ti =            0; % initial Hohmann transfer time, s
tf = 2.2376*10^(7); %   final Hohmann transfer time, s

r_mars = 2.2794*10^(8); % Mars' mean orbit radius, km

r_2 = r_mars; % set the first estimate of the necessary r_2 equal to ...

fprintf('Iteration     Miss Distance d_a, km     Desired d_star, km ...
fprintf('---------------------------------------------------------- ...

i = 1; % initialize the loop counter i

while(1)
    % compute the initial satellite state and the necessary miss distance
    % d_a_star in order to achieve the desired r_3
    [y0, d_star(i)] = orbit_setup_2(r_2);

    % use the Fourth-Order Runge-Kutta integrator to compute the satellite's
    % state from the initial time ti to the final time tf
    [t, state, SvsE, SvsM, d_a(i)] = RK_4_4body(ti, tf, y0);

    % compute the error in the miss distance as the difference between the
    % actual distance and the necessary distance
```

```
    d_error(i) = d_a(i) - d_star(i);

    fprintf('%2d              %+8.5e              %+8.5e           %+8.5e ...

    % if the error in miss distance drops below 1000 km, or 15 iterations
    % have been performed, plot the final orbit and exit the loop
    if abs(d_error(i)) < 1e+001 | i == 15
        plot_MarsCentric(SvsM)
        break
    else
        r_2 = r_2 + d_error(i);
    end

    % increment the loop counter i
    i = i + 1;
end

% plot the error function for the iterative process
plot(1:i,abs(d_error),'.k')
xlabel('Iteration i')
ylabel('Miss Distance Error d_{error} Magnitude, km')
```

C.2 orbit_setup_2.m

```
function [y0, d_a_star] = orbit_setup_2(r_2)
% orbit_setup_2:
% compute the initial heliocentric position and velocity of a satellite on
% a Hohmann transfer from Earth to Mars, in addition to the necessary miss
% distance for a desired final parking orbit radius r_3 about Mars

% input:
% Mars' orbit radius r_2

% output:
% initial satellite state y0
% necessary miss distance d_a_star, given a desired Mars parking orbit
% radius r_3

% Thomas Reppert 04/24/06

mu_sun = 1.326*10^(11); % sun's gravitational coefficient, km^3/s^2
mu_earth = 3.986*10^(5); % Earth's gravitational coefficient, km^3/s^2
mu_mars  = 4.282*10^(4); % Mars' gravitational coefficient, km^3/s^2
r_earth = 1.496*10^(8); % Earth's mean orbit radius, km
```

```
r_mars = 2.2794*10^(8); % Mars' mean orbit radius, km

% departure orbit parameters
% --------------------------

r_0 = 7500; % initial parking orbit radius, km

c = r_earth + r_2; % Hohmann transfer chord length, km

v_1 = sqrt(2*mu_sun/c*(r_2/r_earth)); % spacecraft's velocity when ...

n_earth = sqrt(mu_sun/r_earth^(3)); % Earth's mean orbit rate, rad/s

v_earth = r_earth*n_earth; % Earth's heliocentric velocity, km/s

nu_1 = v_1 - v_earth; % spacecraft's Earth-centric departure velocity at ...

a = -mu_earth/nu_1^2; % hyperbolic semi-major axis, km

nu_0 = sqrt(2*mu_earth/r_0 - mu_earth/a); % spacecraft's Earth-centric ...

nu_c = sqrt(mu_earth/r_0); % initial Earth-centric critical velocity, km/s

e = r_0*nu_0^2/mu_earth - 1; % departure orbit eccentricity

Phi = acos(1/e) + pi; % initial burn angle Phi, rad

% initial satellite state vector
% ------------------------------

% heliocentric position, km
x_0 = r_earth + r_0*cos(3*pi/2-Phi);
y_0 =          -r_0*sin(3*pi/2-Phi);
z_0 =                             0;

% heliocentric velocity, km/s
xdot_0 =           nu_0*cos(Phi-pi);
ydot_0 = v_earth + nu_0*sin(Phi-pi);
zdot_0 =                          0;

y0 = [x_0, y_0, z_0, xdot_0, ydot_0, zdot_0];

% arrival orbit parameters
% ------------------------
```

```
r_3 = 4000; % desired Mars parking orbit radius, km

v_2 = sqrt(2*mu_sun*(1/r_2 - 1/r_earth) + v_1^2); % spacecraft's ...

sigma_2 = acos(r_earth*v_1/(r_2*v_2)); % heading angle between the ...

n_mars = sqrt(mu_sun/r_mars^(3)); % Mars' mean orbit rate, rad/s

v_mars = r_mars*n_mars; % Mars' heliocentric velocity, km/s

nu_2 = sqrt(v_2^2 + v_mars^2 - 2*v_2*v_mars*cos(sigma_2)); % spacecraft ...

e = r_3*nu_2^2/mu_mars + 1; % arrival orbit eccentricity

d_a_star = -sqrt((e^2 - 1)*(mu_mars/nu_2^2)^2); % necessary miss ...
```

C.3 MissDistance.m

```
function d_a = MissDistance(i, SvsM)
% MissDistance:
% plots a line tangent to the satellite's Mars-centric velocity upon entry
% into Mars' sphere of influence and calculates the miss distance d_a, km

% input:
% integration counter i
% satellite's integrated Mars-centric state vector SvsM

% output:
% miss distance d_a, km

% Thomas Reppert 04/23/06

% create a linear fit for the satellite's motion upon entry into Mars'
% sphere of influence
x_poly = [SvsM(1,i-8:i+1)];
y_poly = [SvsM(2,i-8:i+1)];
poly = polyfit(x_poly,y_poly,1);

% plot a tangent to the Mars-centric velocity upon entry
x_tan = [-5e+005:10:0];
y_tan = poly(1).*x_tan + poly(2);
plot(x_tan,y_tan,'b--')
hold on
```

```
% calculate the miss distance d_a, km, with the appropriate sign relative
% to the Mars frame M
r_tan = sqrt(x_tan.^2 + y_tan.^2);
[d_a, index] = min(r_tan);
if sign(x_tan(index)) == -1
    d_a = -d_a;
end
```

Bibliography

[1] Richard H. Battin. *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition.* American Institute of Aeronautics and Astronautics, Inc. Reston, VA., 1999.

[2] Hanspeter Schaub and John L. Junkins. *Analytical Mechanics of Space Systems.* American Institute of Aeronautics and Astronautics, Inc. Reston, VA., 2003.

[3] Roger R. Bate, Donald D. Mueller, and Jerry E. White. *Fundamentals of Astrodynamics.* Dover Publications, Inc. New York, NY., 1971.

[4] Steven C. Chapra. *Applied Numerical Methods with MatLab.* McGraw-Hill Companies, Inc. New York, NY., 2005.

[5] Werner Kohler and Lee Johnson. *Elementary Differential Equations with Boundary Value Problems.* Pearson Education, Inc. New York, NY., 2004.