

## A MATLAB Script for Preliminary Earth-to-Mars Mission Design

This document describes a MATLAB script named `e2m.m` that can be used to design ballistic interplanetary missions from Earth park orbit to B-plane encounter at Mars. The software assumes that interplanetary injection occurs *impulsively* from a circular Earth park orbit. The B-plane coordinates are expressed in a Mars-centered (areocentric) mean equator and equinox of date coordinate system. These B-plane targets are enforced via a user-defined periapsis radius and orbital inclination of the arrival hyperbola.

The first part of this MATLAB script solves for the minimum delta-v using a patched-conic, two-body Lambert solution for the transfer trajectory from Earth to Mars. The second part implements a simple *shooting* method that attempts to minimize an impulsive trajectory correction maneuver (TCM) located at the Earth's sphere-of-influence (SOI) while numerically integrating the spacecraft's heliocentric equations of motion and targeting to components of the B-plane relative to Mars.

The spacecraft motion within the Earth's SOI includes the Earth's  $J_2$  oblate gravity effect and the point-mass perturbation of the moon. The heliocentric equations of motion include the point-mass gravity of the sun and the first seven planets of the solar system.

The user can select one of the following delta-v optimization options for the two-body solution of the interplanetary transfer trajectory:

- minimize launch delta-v
- minimize arrival delta-v
- minimize total delta-v
- no optimization

The major computational steps implemented in this script are as follows:

- solve the two-body, patched-conic interplanetary Lambert problem for the energy  $C_3$ , declination (DLA) and asymptote (RLA) of the outgoing hyperbola
- compute the orbital elements of the launch hyperbola and the components of the interplanetary injection delta-v vector
- perform geocentric orbit propagation from perigee of the launch hyperbola to the Earth's sphere-of-influence (SOI)
- perform an n-body heliocentric orbit propagation from the Earth's SOI to the B-plane at Mars encounter
- target to the user-defined B-plane coordinates by minimizing the heliocentric delta-v vector or trajectory correction maneuver (TCM) at the Earth's sphere-of-influence

This MATLAB script uses the SNOPT nonlinear programming algorithm to solve both the patched-conic and numerically integrated trajectory optimization problems. With the appropriate coordinate

transformations, the software can be easily modified for any combination of departure and arrival planets within our solar system.

## Input data file

This section describes a typical input data file for the software. In the following discussion the actual input file contents are in *courier* font and all explanations are in *times italic* font.

Each data item within an input file is preceded by one or more lines of *annotation* text. Do not delete any of these annotation lines or increase or decrease the number of lines reserved for each comment. However, you may change them to reflect your own explanation. The annotation line also includes the correct units and when appropriate, the valid range of the input. ASCII text input is not case sensitive but must be spelled correctly.

*The first six lines of any input file are reserved for user comments. These lines are ignored by the software. However the input file must begin with six and only six initial text lines.*

```
*****
** interplanetary trajectory optimization
** patched-conic + n-body b-plane targeting
** Mars '03 mars03.in
** October 20, 2004
*****
```

*The first input is an integer that defines the type of patched-conic trajectory optimization.*

```
*****
* simulation type *
*****
1 = minimize launch delta-v
2 = minimize arrival delta-v
3 = minimize total delta-v
4 = no optimization
-----
1
```

*The next input defines an initial guess for the launch calendar date. Please be sure to include all digits of the calendar year.*

```
launch calendar date initial guess (month, day, year)
6,1,2003
```

*These two numbers define the lower and upper search interval for the launch calendar date.*

```
launch date search boundary (days)
30
```

*The next input defines an initial guess for the arrival calendar date.*

```
arrival calendar date initial guess (month, day, year)
12,1,2003
```

*These two numbers define the lower and upper search interval for the arrival calendar date.*

```
arrival date search boundary (days)
30
```

*The software allows the user to specify an initial guess for the launch and arrival calendar dates and a search interval. For any guess for launch time  $t_L$  and user-defined search interval  $\Delta t$ , the launch time  $t$  is constrained as follows:*

$$t_L - \Delta t \leq t \leq t_L + \Delta t$$

Likewise, for any guess for arrival time  $t_A$  and user-defined search interval, the arrival time  $t$  is constrained as follows:

$$t_A - \Delta t \leq t \leq t_A + \Delta t$$

For fixed launch and/or arrival times, the search interval should be set to 0.

The next set of inputs defines the characteristics of the launch hyperbola.

```
*****
* geocentric phase modeling
*****

perigee altitude of launch hyperbola (kilometers)
185.2

launch azimuth (degrees)
93.0

launch site latitude (degrees)
28.5
```

These next two inputs define the radius of closest approach and the orbital inclination of the encounter hyperbola at Mars.

```
*****
* encounter planet targeting
*****

radius of closest approach (kilometers)
5000.0

orbital inclination (degrees)
60.0
```

The final input is an integer that specifies the type of targeting at Mars encounter. For most problems, b-plane targeting is recommended.

```
type of targeting
(1 = b-plane, 2 = orbital elements)
1
```

## Program example

The following is the solution created with this MATLAB script for this example. The output is organized in the following major sections:

- First Pass
  1. two body Lambert solution
  2. departure hyperbola orbital elements and state vector
  3. time and conditions at Earth SOI
- Targeting Pass
  1. time and conditions at Mars closest approach

*The first output section summarizes the two-body Lambert solution. The solution is provided in the heliocentric, earth mean equator and equinox of J2000 coordinate system.*

```

-----
two-body Lambert solution
-----

minimize departure delta-v

departure calendar date      05-Jun-2003
departure universal time     14:46:46.546
departure julian date        2452796.11581651
arrival calendar date        24-Dec-2003
arrival universal time       15:23:10.886
arrival julian date          2452998.14109821
transfer time                 202.025282 days

heliocentric orbital elements of the Earth at departure
(mean equator and equinox of J2000)
-----
      sma (km)      eccentricity      inclination (deg)      argper (deg)
1.4965147326e+008  1.6237346599e-002  2.3439054671e+001  1.0245240439e+002
      raan (deg)      true anomaly (deg)      arglat (deg)      period (days)
7.2430845695e-004  1.5204742997e+002  2.5449983436e+002  3.6545322928e+002

heliocentric orbital elements of the transfer orbit
(mean equator and equinox of J2000)
-----
      sma (km)      eccentricity      inclination (deg)      argper (deg)
1.8838714746e+008  1.9427720614e-001  2.3490037881e+001  2.5349091882e+002
      raan (deg)      true anomaly (deg)      arglat (deg)      period (days)
4.5596571320e-001  5.9131918849e-001  2.5408223801e+002  5.1616340902e+002

heliocentric orbital elements of Mars at arrival
(mean equator and equinox of J2000)
-----
      sma (km)      eccentricity      inclination (deg)      argper (deg)
2.2793930706e+008  9.3541889964e-002  2.4677224952e+001  3.3297923712e+002
      raan (deg)      true anomaly (deg)      arglat (deg)      period (days)
3.3716583265e+000  7.0759517454e+001  4.3738754577e+001  6.8697217107e+002

```

*The following output summarizes the characteristics of the departure hyperbola.*

```

-----
departure hyperbola characteristics
(mean equator and equinox of J2000)
-----

c3                          8.787141 km^2/sec^2
asymptote right ascension    349.621254 degrees
asymptote declination        -6.697391 degrees
perigee altitude             185.200000 kilometers
launch azimuth                93.000000 degrees

```

```

launch site latitude          28.500000  degrees

departure delta-v vector and magnitude

x-component of delta-v       1047.634749  meters/sec
y-component of delta-v      -3032.560850  meters/sec
z-component of delta-v      -1675.729077  meters/sec

delta-v magnitude           3619.672888  meters/sec

orbital elements and state vector of departure hyperbola

      sma (km)          eccentricity      inclination (deg)      argper (deg)
-4.53617906003996e+004  +1.14468873281079e+000  +2.86442848562298e+001  +1.95039551583339e+002

      raan (deg)        true anomaly (deg)      arglat (deg)          period (min)
+2.03563959979895e+000  +2.54444374517081e-014  +1.95039551583339e+002  +1.66666500000000e+003

      rx (km)          ry (km)          rz (km)          rmag (km)
-6.28143348793509e+003  -1.71886477045716e+003  -8.16412391582116e+002  +6.56334000000000e+003

      vx (kps)         vy (kps)         vz (kps)         vmag (kps)
+3.30315643182673e+000  -9.56156035304362e+000  -5.28351630841718e+000  +1.14127044726184e+001

```

*This section of the program output summarizes the flight conditions at the Earth's sphere-of-influence.*

```

geocentric orbital elements and state vector at Earth SOI
(mean equator and equinox of J2000)
-----
calendar date      08-Jun-2003

universal time     18:20:04.132

julian date       2452799.26393672

      sma (km)          eccentricity      inclination (deg)      argper (deg)
-4.56135016246948e+004  +1.14313384732558e+000  +2.84942802048343e+001  +1.94982474359514e+002

      raan (deg)        true anomaly (deg)      arglat (deg)          period (min)
+2.05472022566763e+000  +1.49491123158776e+002  +3.44473597518290e+002  +1.66666500000000e+003

      rx (km)          ry (km)          rz (km)          rmag (km)
+8.98130620908472e+005  -1.85447062652953e+005  -1.18080687221920e+005  +9.24647107795642e+005

      vx (kps)         vy (kps)         vz (kps)         vmag (kps)
+3.02779245685643e+000  -5.52102904784320e-001  -3.58431687973537e-001  +3.09851868708445e+000

```

```

heliocentric orbital elements and state vector at Earth SOI
(mean equator and equinox of J2000)
-----
calendar date      08-Jun-2003

universal time     18:20:04.132

      sma (km)          eccentricity      inclination (deg)      argper (deg)
+1.90725839773535e+008  +2.04057155463721e-001  +2.34926881132669e+001  +2.53488221745382e+002

      raan (deg)        true anomaly (deg)      arglat (deg)          period (min)
+4.63204559016168e-001  +3.94833188811951e+000  +2.57436553633502e+002  +7.57159027273036e+005

      rx (km)          ry (km)          rz (km)          rmag (km)
-3.19342997904423e+007  -1.36207432576492e+008  -5.90898594364571e+007  +1.51867954751507e+008

      vx (kps)         vy (kps)         vz (kps)         vmag (kps)
+3.16260178824110e+001  -6.55310205371182e+000  -2.95941474377289e+000  +3.24331048351928e+001

```

*This section of the program output summarizes the delta-v characteristics at exit from the Earth's sphere-of-influence.*

```

Earth SOI delta-v vector and magnitude
(mean equator and equinox of J2000)
-----
x-component of delta-v      2.859915  meters/second
y-component of delta-v      19.689313  meters/second
z-component of delta-v      -2.659073  meters/second

delta-v magnitude           20.072838  meters/second

```

*This section of the program output summarizes the flight conditions at closest approach to Mars. It includes the B-plane coordinates.*

```

time and conditions at closest approach
(areocentric mean equator and IAU node of epoch)
-----
calendar date      24-Dec-2003

universal time     02:07:45.142

julian date        2452997.58871693

      sma (km)          eccentricity          inclination (deg)          argper (deg)
-5.84690415255194e+003  +1.85515337609291e+000  +5.99999941846519e+001  +1.13981853744237e+002

      raan (deg)         true anomaly (deg)          arglat (deg)             period (min)
+1.05661355079019e+002  +2.33569249142951e-006  +1.13981856079930e+002  +1.66666500000000e+003

      rx (km)           ry (km)           rz (km)           rmag (km)
-1.65077883650482e+003  -2.57340625531487e+003  +3.95632502905394e+003  +4.9999982574646e+003

      vx (kps)          vy (kps)          vz (kps)          vmag (kps)
+2.18745004000982e+000  -4.07936921300577e+000  -1.74072587894118e+000  +4.94533289467694e+000

flight path angle      +1.5176304898e-006  degrees

b-plane coordinates of incoming hyperbola
(areocentric mean equator and IAU node of epoch)
-----
b-magnitude           9136.139105  kilometers
b dot r                -7889.410497
b dot t                4607.194347
b-plane angle          300.283729  degrees
v-infinity             2706.467505  meters/second
r-periapsis            4999.999826  kilometers
decl-asymptote         7.472205  degrees
rasc-asymptote         281.318492  degrees

```

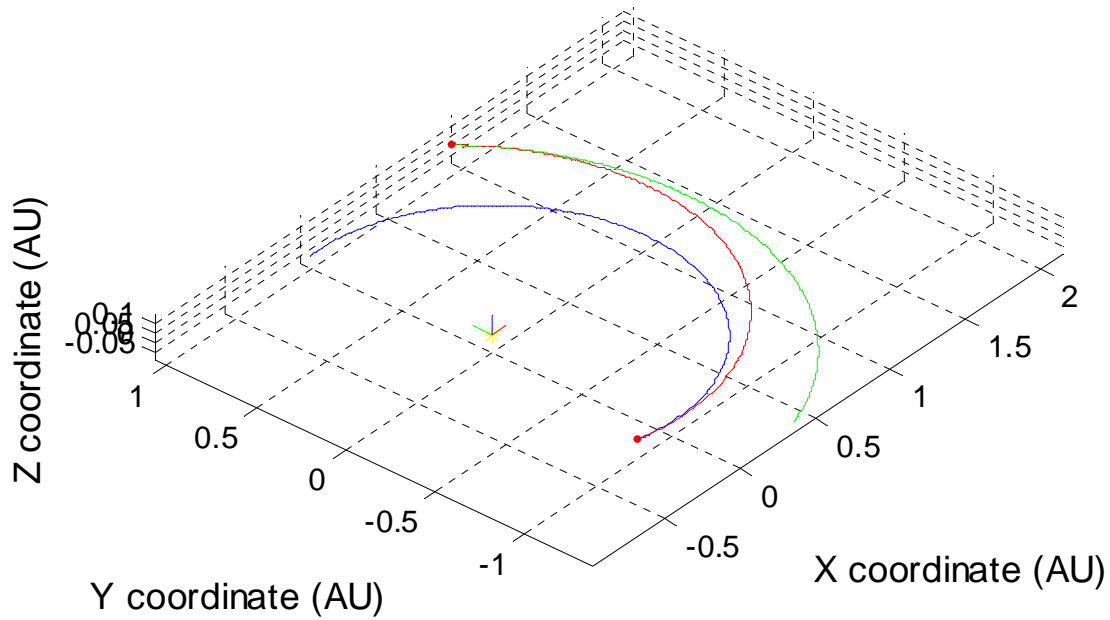
*From the results of this simulation, we can see that the n-body effects and B-plane targeting require a 20 meters/second trajectory correction maneuver at the Earth's sphere-of-influence.*

*This solution can be used as initial conditions for a trajectory simulation that attempts to minimize the total mission delta-v. The control variables for this simulation might include the orbital elements of the initial park orbit, the time of the trajectory correction maneuver (TCM) and the components of the injection and TCM maneuvers themselves.*

This script will also create a graphics display of the interplanetary and encounter trajectories. The interactive graphic features of MATLAB will allow the user to rotate and “zoom” the displays in and out. These capabilities allow the user to interactively find the “best” viewpoint as well as verify basic orbital geometry of the heliocentric and areocentric trajectories.

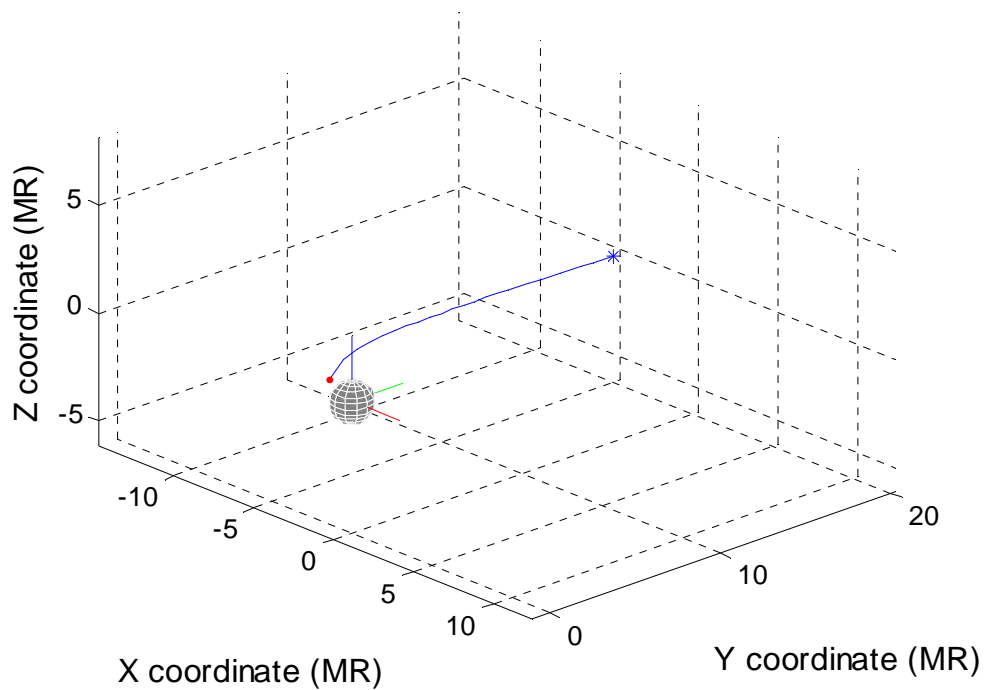
The following is a heliocentric, ecliptic view of the transfer and planetary orbits. The x-axis of this system is red, the y-axis green and the z-axis is blue.

## Heliocentric Transfer Trajectory



This next plot is a view of the encounter trajectory in the Mars-centered mean equator and equinox of date coordinate system. The small red dot is the periapsis of the encounter hyperbola.

## Mars-centered Trajectory



**Technical discussion**

### *Solving the two body Lambert problem*

Lambert's problem is concerned with the determination of an orbit that passes between two positions within a specified time-of-flight. This classic astrodynamics problem is also known as the orbital two-point boundary value problem (TPBVP).

The time to traverse a trajectory depends only upon the length of the semimajor axis  $a$  of the transfer trajectory, the sum  $r_i + r_f$  of the distances of the initial and final positions relative to a central body, and the length  $c$  of the chord joining these two positions. This relationship can be stated as follows:

$$tof = tof(r_i + r_f, c, a)$$

From the following form of Kepler's equation

$$t - t_0 = \sqrt{\frac{a^3}{\mu}} (E - e \sin E)$$

we can write

$$t = \sqrt{\frac{a^3}{\mu}} [E - E_0 - e(\sin E - \sin E_0)]$$

where  $E$  is the eccentric anomaly associated with radius  $r$ ,  $E_0$  is the eccentric anomaly at  $r_0$ , and  $t = 0$  when  $r = r_0$ .

At this point we need to introduce the following trigonometric sum and difference identities:

$$\sin \alpha - \sin \beta = 2 \sin \frac{\alpha - \beta}{2} \cos \frac{\alpha + \beta}{2}$$

$$\cos \alpha - \cos \beta = -2 \sin \frac{\alpha - \beta}{2} \sin \frac{\alpha + \beta}{2}$$

$$\cos \alpha + \cos \beta = 2 \cos \frac{\alpha - \beta}{2} \cos \frac{\alpha + \beta}{2}$$

If we let  $E = \alpha$  and  $E_0 = \beta$  and substitute the first trig identity into the second equation above, we have the following equation:

$$t = \sqrt{\frac{a^3}{\mu}} \left\{ E - E_0 - 2 \sin \frac{E - E_0}{2} \left( e \cos \frac{E + E_0}{2} \right) \right\}$$

With the two substitutions given by

$$e \cos \frac{E + E_0}{2} = \cos \frac{\alpha + \beta}{2}$$

$$\sin \frac{E - E_0}{2} = \sin \frac{\alpha - \beta}{2}$$

the time equation becomes

$$t = \sqrt{\frac{a^3}{\mu}} \left\{ (\alpha - \beta) - 2 \sin \frac{\alpha - \beta}{2} \cos \frac{\alpha + \beta}{2} \right\}$$

From the elliptic relationships given by

$$r = a(1 - e \cos E)$$

$$x = a(\cos E - e)$$

$$y = a \sin E \sqrt{1 - e^2}$$

and some more manipulation, we have the following equations:

$$\cos \alpha = \left( 1 - \frac{r + r_0}{2a} \right) - \frac{c}{2a} = 1 - \frac{r + r_0 + c}{2a} = 1 - \frac{s}{a}$$

$$\sin \beta = \left( 1 - \frac{r + r_0}{2a} \right) + \frac{c}{2a} = 1 - \frac{r + r_0 - c}{2a} = 1 - \frac{s - c}{a}$$

This part of the derivation makes use of the following three relationships:

$$\cos \frac{\alpha - \beta}{2} \cos \frac{\alpha + \beta}{2} = 1 - \frac{r + r_0}{2a}$$

$$\sin \frac{\alpha - \beta}{2} \sin \frac{\alpha + \beta}{2} = \sin \frac{E - E_0}{2} \sqrt{1 - \left( e \cos \frac{E + E_0}{2} \right)^2}$$

$$\left( \sin \frac{\alpha - \beta}{2} \sin \frac{\alpha + \beta}{2} \right)^2 = \left( \frac{x - x_0}{2a} \right)^2 + \left( \frac{y - y_0}{2a} \right)^2 = \left( \frac{c}{2a} \right)^2$$

With the use of the half angle formulas given by

$$\sin \frac{\alpha}{2} = \sqrt{\frac{s}{2a}} \quad \sin \frac{\beta}{2} = \sqrt{\frac{s - c}{2a}}$$

and several additional substitutions, we have the time-of-flight form of Lambert's theorem

$$t = \sqrt{\frac{a^3}{\mu}} [(\alpha - \beta) - (\sin \alpha - \sin \beta)]$$

A discussion about the angles  $\alpha$  and  $\beta$  can be found in “Geometrical Interpretation of the Angles  $\alpha$  and  $\beta$  in Lambert’s Problem” by J. E. Prussing, *AIAA Journal of Guidance and Control*, Volume 2, Number 5, Sept.-Oct. 1979, pages 442-443.

The algorithm used in this MATLAB script is based on the method described in “A Procedure for the Solution of Lambert’s Orbital Boundary-Value Problem” by R. H. Gooding, *Celestial Mechanics and Dynamical Astronomy* **48**: 145-165, 1990. This iterative solution is valid for elliptic, parabolic and hyperbolic transfer orbits which may be either posigrade or retrograde, and involve one or more revolutions about the central body.

### *Designing the launch hyperbola*

This section describes the algorithm used to determine the Earth-centered-inertial (ECI) state vector of a departure hyperbola for interplanetary missions. In the discussion that follows, interplanetary injection is assumed to occur *impulsively* at perigee of the departure hyperbola. The method described here is based on the fundamental characteristics of the B-plane coordinate system.

The departure trajectory for interplanetary missions is specified by the orbital energy  $C_3$ , and the right ascension  $\alpha_\infty$  and declination  $\delta_\infty$  of the outgoing asymptote. The perigee radius of the departure hyperbola is specified and the orbital inclination is computed from the user-defined launch azimuth  $\Sigma_L$  and launch site geocentric latitude  $\phi_L$  from the equation  $i = \cos^{-1}(\cos \phi_L \sin \Sigma_L)$ .

The algorithm used to design the departure hyperbola only works for geocentric orbit inclinations that satisfy the following constraint

$$|i| > |\delta_\infty|$$

If this inequality is not satisfied, the software will print the following error message

```
park orbit error!!
|inclination| must be > |asymptote declination|
```

The code will also print the inclination of the park orbit, the declination of the launch hyperbola and stop. The user can then change either the azimuth or launch site latitude to satisfy this constraint and restart the program.

A unit vector in the direction of the departure asymptote is given by

$$\hat{\mathbf{S}} = \begin{Bmatrix} \cos \delta_\infty \cos \alpha_\infty \\ \cos \delta_\infty \sin \alpha_\infty \\ \sin \delta_\infty \end{Bmatrix}$$

where

$$\begin{aligned}\alpha_\infty &= \text{right ascension of departure asymptote} \\ \delta_\infty &= \text{declination of departure asymptote}\end{aligned}$$

The T-axis direction of the B-plane coordinate system is determined from the following vector cross product:

$$\hat{\mathbf{T}} = \hat{\mathbf{S}} \times \hat{\mathbf{u}}_z$$

where  $\hat{\mathbf{u}}_z = [0 \ 0 \ 1]^T$  is a unit vector perpendicular to the Earth's equator.

The following cross product operation completes the B-plane coordinate system.

$$\hat{\mathbf{R}} = \hat{\mathbf{S}} \times \hat{\mathbf{T}}$$

The B-plane angle is determined from the orbital inclination of the departure hyperbola  $i$  and the declination of the outgoing asymptote according to

$$\cos \theta = \frac{\cos i}{\cos \delta_\infty}$$

The unit angular momentum vector of the departure hyperbola is given by

$$\hat{\mathbf{h}} = \hat{\mathbf{T}} \sin \theta - \hat{\mathbf{R}} \cos \theta$$

The sine and cosine of the true anomaly at infinity are given by the next two equations

$$\begin{aligned}\cos \theta_\infty &= -\frac{\mu}{r_p V_\infty^2 + \mu} \\ \sin \theta_\infty &= \sqrt{1 - \cos^2 \theta_\infty}\end{aligned}$$

where  $V_\infty = \sqrt{C_3} = V_L - V_p$  is the spacecraft's velocity at infinity,  $V_L$  is the heliocentric departure velocity determined from the Lambert solution,  $V_p$  is the heliocentric velocity of the departure planet, and  $r_p$  is the user-specified perigee radius of the departure hyperbola.

A unit vector in the direction of perigee of the departure hyperbola is determined from

$$\hat{\mathbf{r}}_p = \hat{\mathbf{S}} \cos \theta_\infty - (\hat{\mathbf{h}} \times \hat{\mathbf{S}}) \sin \theta_\infty$$

The ECI position vector at perigee is

$$\mathbf{r}_p = r_p \hat{\mathbf{r}}_p$$

The scalar magnitude of the perigee velocity can be determined from

$$V_p = \sqrt{\frac{2\mu}{r_p} + V_\infty^2}$$

A unit vector aligned with the velocity vector at perigee is

$$\hat{\mathbf{v}}_p = \hat{\mathbf{h}} \times \hat{\mathbf{r}}_p$$

The ECI velocity vector at perigee of the departure hyperbola is given by

$$\mathbf{v}_p = V_p \hat{\mathbf{v}}_p$$

Finally, the classical orbital elements of the departure hyperbola can be determined from the position and velocity vectors at perigee. The injection delta-v vector and magnitude can be determined from the velocity difference between the park orbit and departure hyperbola at the orbital location of the impulsive maneuver.

### *Propagating the spacecraft's trajectory*

This section describes the algorithms used to propagate the spacecraft's trajectory during both the geocentric and heliocentric phases of the mission.

### Geocentric trajectory propagation

This part of the trajectory analysis implements a *special perturbation* technique which numerically integrates the vector system of second-order, nonlinear differential equations of motion of a spacecraft given by

$$\vec{a}(\vec{r}, \vec{v}, t) = \vec{r}''(\vec{r}, \vec{v}, t) = \vec{a}_g(\vec{r}) + \vec{a}_m(\vec{r}, t)$$

where

$t$  = time

$\vec{r}$  = inertial position vector of the satellite

$\vec{v}$  = inertial velocity vector of the satellite

$\vec{a}_g$  = acceleration due to Earth gravity

$\vec{a}_m$  = acceleration due to the Moon

The system of six first-order differential equations subject to Earth gravity is defined by

$$\dot{y}_1 = v_x = y_4$$

$$\dot{y}_2 = v_y = y_5$$

$$\dot{y}_3 = v_z = y_6$$

$$\dot{y}_4 = -\mu \frac{r_x}{r^3} \left\{ 1 + \frac{3 J_2 r_{eq}^2}{2 r^2} \left( 1 - \frac{5 r_z^2}{r^2} \right) \right\}$$

$$\dot{y}_5 = -\mu \frac{r_y}{r^3} \left\{ 1 + \frac{3 J_2 r_{eq}^2}{2 r^2} \left( 1 - \frac{5 r_z^2}{r^2} \right) \right\}$$

$$\dot{y}_6 = -\mu \frac{r_z}{r^3} \left\{ 1 + \frac{3 J_2 r_{eq}^2}{2 r^2} \left( 3 - \frac{5 r_z^2}{r^2} \right) \right\}$$

where  $r = \sqrt{r_x^2 + r_y^2 + r_z^2} = \sqrt{y_1^2 + y_2^2 + y_3^2}$ . In these equations  $\mu$  and  $r_{eq}$  are the gravitational constant and equatorial radius of the Earth, respectively and  $J_2$  is the oblateness gravity coefficient.

The acceleration contribution of the Moon represented by a *point mass* is given by

$$\bar{a}_m(\vec{r}, t) = -\mu_m \left( \frac{\vec{r}_{m-b}}{|\vec{r}_{m-b}|^3} + \frac{\vec{r}_{e-m}}{|\vec{r}_{e-m}|^3} \right)$$

where

$\mu_m$  = gravitational constant of the Moon

$\vec{r}_{m-b}$  = position vector from the Moon to the satellite

$\vec{r}_{e-m}$  = position vector from the Earth to the Moon

### Heliocentric trajectory propagation

The general vector equation for *point-mass* perturbations such as the Moon or planets is given by

$$\ddot{\mathbf{r}} = -\sum_{j=1}^n \mu_j \left[ \frac{\mathbf{d}_j}{d_j^3} + \frac{\mathbf{s}_j}{s_j^3} \right]$$

In this equation,  $\mathbf{s}_j$  is the vector from the primary body to the secondary body  $j$ ,  $\mu_j$  is the gravitational constant of the secondary body and  $\mathbf{d}_j = \mathbf{r} - \mathbf{s}_j$ , where  $\mathbf{r}$  is the position vector of the spacecraft relative to the primary body.

To avoid numerical problems, use is made of Battin's  $F(q)$  function given by

$$F(q_k) = q_k \left[ \frac{3 + 3q_k + q_k^2}{1 + (\sqrt{1 + q_k})^3} \right]$$

where

$$q_k = \frac{\mathbf{r}^T (\mathbf{r} - 2\mathbf{s}_k)}{\mathbf{s}_k^T \mathbf{s}_k}$$

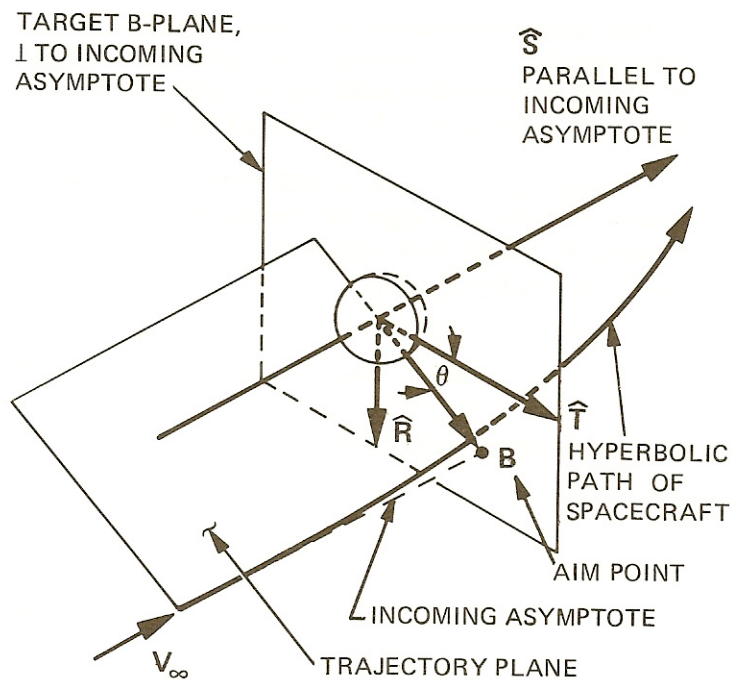
The third-body acceleration can now be expressed as

$$\ddot{\mathbf{r}} = -\sum_{k=1}^n \frac{\mu_k}{d_k^3} [\mathbf{r} + F(q_k)\mathbf{s}_k]$$

In this MATLAB script the heliocentric coordinates of the planets are based on the JPL Development Ephemeris DE410. These coordinates are provided in the Earth mean equator and equinox of J2000 coordinate system (EME2000).

### *B-plane targeting*

The derivation of B-plane coordinates is described in the classic JPL reports, “A Method of Describing Miss Distances for Lunar and Interplanetary Trajectories” and “Some Orbital Elements Useful in Space Trajectory Calculations”, both by William Kizner. The following diagram illustrates the fundamental geometry of the B-plane coordinate system.



The software solves the B-plane targeting problem by minimizing the delta-v vector at the SOI while satisfying two nonlinear *equality constraint* equations. These constraint equations are the differences between components of the *required* B-plane and the B-plane components *predicted* by the software.

Given the user-defined closest approach radius  $r_{ca}$  and orbital inclination  $i$ , and the incoming v-infinity magnitude  $v_\infty$  and the right ascension  $\alpha_\infty$  and declination  $\delta_\infty$  of the incoming asymptote vector at moment of closest approach, the following series of equations can be used to determine the required B-plane target vector:

$$\mathbf{B} \cdot \mathbf{T} = b_t \cos \theta$$

$$\mathbf{B} \cdot \mathbf{R} = b_t \sin \theta$$

where

$$b_t = \sqrt{\frac{2\mu r_{ca}}{v_\infty^2} + r_{ca}^2} = r_{ca} \sqrt{1 + \frac{2\mu}{r_{ca} v_\infty^2}}$$

and

$$\cos \theta = \frac{\cos i}{\cos \delta_\infty}$$

$$\sin \theta = -\sqrt{1 - \cos^2 \theta}$$

$$\sin \delta_\infty = |\hat{\mathbf{s}} \times \hat{\mathbf{z}}| = \sqrt{s_x^2 + s_y^2}$$

$$\hat{\mathbf{z}} = [0 \quad 0 \quad 1]^T$$

The arrival asymptote unit vector  $\hat{\mathbf{S}}$  is given by

$$\hat{\mathbf{S}} = \begin{Bmatrix} \cos \delta_\infty \cos \alpha_\infty \\ \cos \delta_\infty \sin \alpha_\infty \\ \sin \delta_\infty \end{Bmatrix}$$

where  $\delta_\infty$  and  $\alpha_\infty$  are the declination and right ascension of the asymptote of the incoming hyperbola.

*Important note!!*

This technique only works for aerocentric orbit inclinations that satisfy

$$|i| > |\delta_\infty|$$

If this inequality is not satisfied, the software will print the following error message

```
b-plane targeting error!!
```

```
|inclination| must be > |asymptote declination|
```

It will also display the actual declination of the asymptote and stop. The user should then edit the input file, include a valid orbital inclination and restart the simulation.

The following computational steps summarize the calculation of the *predicted* B-plane vector from a planet-centered position vector  $\mathbf{r}$  and velocity vector  $\mathbf{v}$  at closest approach.

angular momentum vector

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

$$\hat{\mathbf{h}} = \frac{\mathbf{h}}{|\mathbf{h}|}$$

radius rate

$$\dot{r} = \frac{\mathbf{r} \cdot \mathbf{v}}{|\mathbf{r}|}$$

semiparameter

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

semimajor axis

$$a = \frac{r}{\left(2 - \frac{rv^2}{\mu}\right)}$$

orbital eccentricity

$$e = \sqrt{1 - p/a}$$

true anomaly

$$\cos \theta = \frac{p - r}{er}$$

$$\sin \theta = \frac{\dot{r}h}{e\mu}$$

B-plane magnitude

$$B = \sqrt{p|a|}$$

fundamental vectors

$$\hat{\mathbf{z}} = \frac{r\mathbf{v} - \dot{r}\mathbf{r}}{h}$$

$$\hat{\mathbf{p}} = \cos \theta \hat{\mathbf{r}} - \sin \theta \hat{\mathbf{z}}$$

$$\hat{\mathbf{q}} = \sin \theta \hat{\mathbf{r}} + \cos \theta \hat{\mathbf{z}}$$

S vector

$$\mathbf{S} = -\frac{a}{\sqrt{a^2 + b^2}} \hat{\mathbf{p}} + \frac{b}{\sqrt{a^2 + b^2}} \hat{\mathbf{q}}$$

B vector

$$\mathbf{B} = \frac{b^2}{\sqrt{a^2 + b^2}} \hat{\mathbf{p}} + \frac{ab}{\sqrt{a^2 + b^2}} \hat{\mathbf{q}}$$

T vector

$$\mathbf{T} = \frac{(S_y^2, -S_x^2, 0)^T}{\sqrt{S_x^2 + S_y^2}}$$

R vector

$$\mathbf{R} = \mathbf{S} \times \mathbf{T} = (-S_z T_y, S_z T_x, S_x T_y - S_y T_x)^T$$

### *Targeting to the Mars-centered Periapsis Radius and Orbital Inclination*

For this targeting option, the equality constraints enforced by the SNOPT nonlinear programming algorithm are

$$r_p - r_{ca} = 0$$

$$\cos i - \hat{\mathbf{h}}_z = 0$$

where  $r_p$  and  $i$  are the user-defined periapsis radius and orbital inclination, respectively, and  $\hat{\mathbf{h}}_z$  is the z-component of the unit angular momentum vector at closest approach to Mars.

The mission elapsed time at which the spacecraft reaches closest approach to Mars is predicted using the event prediction capability of the MATLAB `ode45` algorithm. During the numerical integration of the spacecraft's geocentric equations of motion, the `ode45` numerical method searches for the time at which the flight path angle *with respect to Mars* is nearly zero within a small tolerance. This constraint corresponds to closest approach to Mars. The *predicted* B-plane coordinates are based on the Mars-centered flight conditions at closest approach.

Close approach is predicted with the following *mission constraint*

$$\gamma = \sin^{-1} \left( \frac{\mathbf{r} \cdot \mathbf{v}}{|\mathbf{r} \cdot \mathbf{v}|} \right)$$

where  $\mathbf{r}$  and  $\mathbf{v}$  are the Mars-centered position and velocity vectors, respectively.

### *Geocentric-to-areocentric coordinate transformation*

This section describes the transformation of coordinates between the Earth mean equator and equinox of J2000 and areocentric mean equator and IAU node of epoch coordinate systems. This transformation is used to compute the B-plane coordinates at encounter.

A unit vector in the direction of the pole of Mars can be determined from

$$\hat{\mathbf{p}}_{Mars} = \begin{bmatrix} \cos \alpha_p \cos \delta_p \\ \sin \alpha_p \cos \delta_p \\ \sin \delta_p \end{bmatrix}$$

The IAU 2000 right ascension and declination of the pole of Mars in the EME2000 coordinate system are given by the following expressions

$$\alpha_p = 317.68143 - 0.1061T$$

$$\delta_p = 52.88650 - 0.0609T$$

where  $T$  is the time in Julian centuries given by  $T = (JD - 2451545.0)/36525$  and  $JD$  is the TDB Julian Date.

The unit vector in the direction of the *IAU-defined* x-axis is computed from

$$\hat{\mathbf{x}} = \hat{\mathbf{p}}_{J2000} \times \hat{\mathbf{p}}_{Mars}$$

where  $\hat{\mathbf{p}}_{J2000} = [0 \ 0 \ 1]^T$  is unit vector in the direction of the pole of the J2000 coordinate system.

The unit vector in the y-axis direction of this coordinate system is

$$\hat{\mathbf{y}} = \hat{\mathbf{p}}_{Mars} \times \hat{\mathbf{x}}$$

Finally, the components of the matrix that transforms coordinates from the EME2000 system to the Mars-centered mean equator and IAU node of epoch system are as follows:

$$\mathbf{M} = \begin{bmatrix} \hat{\mathbf{x}} \\ \hat{\mathbf{y}} \\ \hat{\mathbf{p}}_{Mars} \end{bmatrix}$$

## SNOPT algorithm implementation

This section provides details about the parts of the MATLAB script that solve these nonlinear programming (NLP) problems using the SNOPT 6.0 algorithm. In this classic patched-conic trajectory optimization problem, the launch and arrival calendar dates are the *control variables* and the user-specified  $\Delta V$  is the *objective function* or *performance index*.

MATLAB versions of SNOPT 6.0 for several computer platforms can be found at Professor Philip Gill's web site which is located at <http://scicomp.ucsd.edu/~peg/>.

The SNOPT algorithm requires an initial guess for the control variables. For this problem they are given by

```
xg(1) = jdate1 - jdate0;  
xg(2) = jdate2 - jdate0;  
xg = xg' ;
```

where `jdate1` and `jdate2` are the initial user-provided launch and arrival date guesses, and `jdate0` is a reference Julian date equal to 2451544.5 (January 1, 2000). This offset value is used to *scale* the control variables.

The algorithm also requires lower and upper bounds for the control variables. These are determined from the initial guesses and user-defined search boundaries as follows:

```
% bounds on control variables  
  
xlwr(1) = xg(1) - ddays1;  
xupr(1) = xg(1) + ddays1;  
  
xlwr(2) = xg(2) - ddays2;  
xupr(2) = xg(2) + ddays2;  
  
xlwr = xlwr' ;  
xupr = xupr' ;  
  
xlwr = xlwr' ;  
xupr = xupr' ;
```

where `ddays1` and `ddays2` are the user-defined launch and arrival search boundaries, respectively.

The algorithm also requires lower and upper bounds on the objective function. For this problem these bounds are given by

```
% bounds on objective function  
  
flow(1) = 0.0d0;  
fupp(1) = +Inf;
```

The actual call to the SNOPT MATLAB interface function is as follows

```
[x, f, inform, xmul, fmul] = snopt(xg, xlwr, xupr, flow, fupp, 'e2m_deltav');
```

where `e2m_deltav` is the name of the MATLAB function that solves Lambert's problem and computes the current value of the objective function.

The following is the MATLAB source code for the TCM optimization algorithm.

```
% initial guess for soi delta-v vector
```

```

xg(1) = 0.0;

xg(2) = 0.0;

xg(3) = 0.0;

% lower and upper bounds for components
% of soi delta-v vector (meters/second)

for i = 1:1:3
    xlwr(i) = -50.0;

    xupr(i) = +50.0;
end

% bounds on objective function

flow(1) = 0.0d0;

fupp(1) = +Inf;

% bounds on final b-plane/orbital element
% equality constraints

flow(2) = 0.0d0;
fupp(2) = 0.0d0;

flow(3) = 0.0d0;
fupp(3) = 0.0d0;

flow = flow';

fupp = fupp';

```

The actual call to the SNOPT MATLAB interface function for this part of the script is as follows

```
[x, f, inform, xmul, fmul] = snopt(xg, xlwr, xupr, flow, fupp, 'e2m_shoot');
```

where `e2m_shoot` is the name of the MATLAB function that implements the simple shooting method.

## References and Bibliography

- “Update to Mars Coordinate Frame Definitions”, R. A. Mase, JPL IOM 312.B/015-99, 15 July 1999.
- “JPL Planetary Ephemeris DE410”, E. M. Standish, JPL IOM 312.N-03-009, 24 April 2003.
- “Report of the IAU/IAG Working Group on Cartographic Coordinates and Rotational Elements of the Planets and Satellites: 2000”, *Celestial Mechanics and Dynamical Astronomy*, **82**: 83-110, 2002.
- “IERS Conventions (2003)”, IERS Technical Note 32, November 2003.
- “Planetary Constants and Models”, R. Vaughan, JPL D-12947, December 1995.
- “Preliminary Mars Planetary Constants and Models for Mars Sample Return”, D. Lyons, JPL IOM 312/99.DTL-1, 20 January 1999.

“Interplanetary Mission Design Handbook, Volume 1, Part 2”, JPL Publication 82-43, September 15, 1983.

“User’s Guide for SNOPT Version 6, A Fortran Package for Large-Scale Nonlinear Programming”, Philip E. Gill, Walter Murray and Michael A. Saunders, December 2002.

## Computational Chart

