A MATLAB Script for Designing Interplanetary Launch Trajectories

This document describes a MATLAB script named `launch1.m` that can be used for the preliminary design and analysis of launch trajectories for interplanetary spacecraft. Many of the algorithms implemented in this script are based on the classic technical reports, "Design of Lunar and Interplanetary Ascent Trajectories", by Victor C. Clarke, Jr., JPL Technical Report No. 32-30, March 15, 1962, and "Interplanetary Mission Design Handbook, Volume 1, Part 2", by A. Sergeyevsky, G. Snyder, and R. Cunniff, JPL Publication 82-43, September 15, 1983.

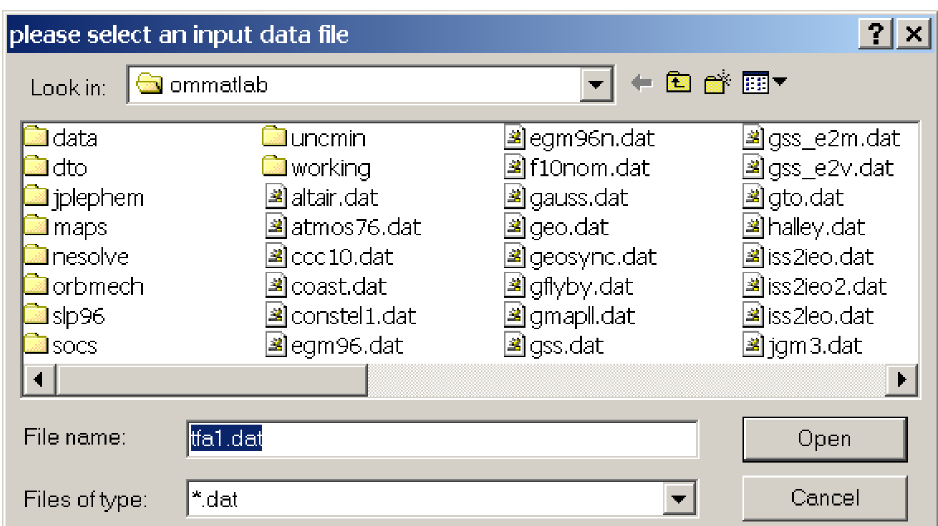
The Earth departure trajectory for interplanetary missions is usually defined by a "targeting specification" which consists of twice the specific (per unit mass) orbital energy C_3, and the right ascension α_∞ (RLA) and declination δ_∞ (DLA) of the outgoing asymptote. The launch azimuth, park orbit altitude, and launch site coordinates are also specified. This MATLAB script assumes that interplanetary injection originates from a circular Earth park orbit.

In order to estimate the park orbit coast duration, the user must also provide estimates of typical Earth central angles for launch-to-park-orbit-inject, and one or more propulsive and intermediate coasting maneuvers. The current version of this MATLAB script provides at most two injection propulsive maneuvers separated by a short coast. However, the software can be easily extended to include additional events.

This MATLAB script determines the launch universal time, orbital elements of the departure hyperbola, injection delta-v, and estimates park orbit coast times for both short and long coast injection opportunities. This information can be used to create initial guesses for other trajectory simulations.

Running the script

The `launch1` script is "data-driven" by a single ASCII input file created by the user. The software will prompt you for the name of this file with a screen display similar to the following:



The default filename extension is `.dat`. However, the script will read a compatible data file with any filename extension the user prefers.

Input data files

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. The following is an input data file for the Mars Exploration Rover mission. The information for this file was extracted from "Mission Design Overview for the Mars Exploration Rover Mission", by Ralph B. Roncoli and Jan M. Ludwinski, AIAA 2002-4823.

Each data item within a `launch1` 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 four lines of any input file are reserved for user comments. These lines are ignored by the software. However the input file must begin with four and only four initial text lines.

```
**********************************
interplanetary injection data file
MER-A mission - mer-a.dat
**********************************
```

The first input item is the name of the simulation. This helps with traceability and will be displayed with the screen display of the solution.

```
mission name
------------
MER-A Mission
```

The next input is the launch calendar date of the simulation. Please include all 4 digits of the calendar year.

```
launch calendar date (month, day, year)
5,30,2003
```

The next data item is the launch azimuth in degrees.

```
launch azimuth (degrees)
93
```

The next data item is the geodetic latitude of the launch site in degrees.

```
launch site geodetic latitude (degrees)
28.285579
```

The next data item is the east longitude of the launch site in degrees.

```
launch site east longitude (degrees)
279.434701
```

The next data item is the altitude of the circular park orbit.

```
circular park orbit altitude (kilometers)
185.2
```

The next data item is the launch energy C_3 in the unit of $km^2/\sec^2$.

```
C3 (twice specific orbital energy; (km/sec)^2)
9.28
```

The next data item is the declination of the launch hyperbola, DLA, in degrees.

```
DLA (declination of launch hyperbola; degrees)
2.27
```

The next data item is the right ascension of the launch hyperbola, RLA, in degrees.

```
RLA (right ascension of launch hyperbola; degrees)
352.59
```

The next data item is an estimate of the Earth central angle from liftoff to park orbit injection, in degrees.

```
central angle from launch to park orbit inject (degrees)
24.0
```

The next data item is an estimate of the Earth central angle of the first and perhaps only interplanetary injection propulsive maneuver, in degrees.

```
central angle for first injection maneuver (degrees)
9.0
```

The next data item is an estimate of the Earth central angle of a short coast between a first and second propulsive maneuver. If there is no second maneuver, set this value to 0.

```
central angle between first and second injection maneuvers (degrees)
7.0
```

The next data item is an estimate of the Earth central angle for a second propulsive maneuver, in degrees. If there is no second maneuver, set this value to 0

```
central angle for second injection maneuver (degrees)
8.0
```

The final data item is an estimate of the injection true anomaly on the departure hyperbola, in degrees. For an impulsive maneuver, this value is usually 0 which corresponds to perigee of the departure hyperbola. For a finite-burn maneuver, this value is non-zero and usually positive.

```
injection true anomaly (degrees)
8.0
```

The injection true anomaly could also include an additional central angle increment that represents a short coast to a target interface point (TIP) which is often part of a target specification. Some spacecraft missions often require that C_3, DLA and RLA be enforced at a TIP and not necessarily at burnout of the upper stage.

Typical script simulation example

This section is a typical output provided by the `launch1` MATLAB script. It illustrates the characteristics for both a short and long park orbit coast mission.

```
-------------------------------------------------
launch1.m - interplanetary launch characteristics
-------------------------------------------------

MER-A Mission

descending injection
====================

launch calendar date                      30-May-2003

launch universal time                     18:29:39.192


launch azimuth                           93.000000  degrees

launch site geodetic latitude            28.446462  degrees

launch site geocentric declination       28.285572  degrees

launch site east longitude              279.434701  degrees


launch hyperbola characteristics
--------------------------------

c3                                        9.280000  (km/sec)^2

asymptote declination                     2.270000  degrees

asymptote right ascension               352.590000  degrees

semimajor axis                       -42952.633782  kilometers

eccentricity                            1.15280406

inclination                              28.431148  degrees

argument of perigee                     214.608488  degrees

raan                                    348.391220  degrees

orbital period                           88.195573  minutes

first park orbit coast time              19.367145  minutes

first park orbit coast angle             79.053538  degrees

range angle                             269.217201  degrees

rasc of launch site at 0 hr             166.529456  degrees

rasc of launch site at launch            84.702262  degrees
```

arglat of launch site at launch	95.554950	degrees
greenwich sidereal time at 0 hr	247.094755	degrees
asymptote true anomaly	150.163663	degrees
asymptote argument of latitude	4.772151	degrees
ascending node to launch site	96.311041	degrees
ascending node to asymptote	4.198780	degrees
local circular velocity	7793.033366	meters/second
injection velocity	11434.279080	meters/second
injection delta-v	3641.245714	meters/second

ascending injection

launch calendar date	30-May-2003	
launch universal time	07:05:07.054	
launch azimuth	93.000000	degrees
launch site geodetic latitude	28.446462	degrees
launch site geocentric declination	28.285572	degrees
launch site east longitude	279.434701	degrees

launch hyperbola characteristics

c3	9.280000	(km/sec)^2
asymptote declination	2.270000	degrees
asymptote right ascension	352.590000	degrees
semimajor axis	-42952.633782	kilometers
eccentricity	1.15280406	
inclination	28.431148	degrees
argument of perigee	25.064186	degrees
raan	176.788780	degrees
orbital period	88.195573	minutes
first park orbit coast time	61.126695	minutes

```
first park orbit coast angle          249.509236  degrees

range angle                            79.672900  degrees

rasc of launch site at 0 hr           166.529456  degrees

rasc of launch site at launch         273.099821  degrees

arglat of launch site at launch        95.554950  degrees

greenwich sidereal time at 0 hr       247.094755  degrees

asymptote true anomaly                150.163663  degrees

asymptote argument of latitude        175.227849  degrees

ascending node to launch site          96.311041  degrees

ascending node to asymptote           175.801220  degrees

local circular velocity              7793.033366  meters/second

injection velocity                  11434.279080  meters/second

injection delta-v                    3641.245714  meters/second
```

In the `launch1` solution display screens,

`rasc` = right ascension

`arglat` = argument of latitude = sum of argument of perigee and true anomaly

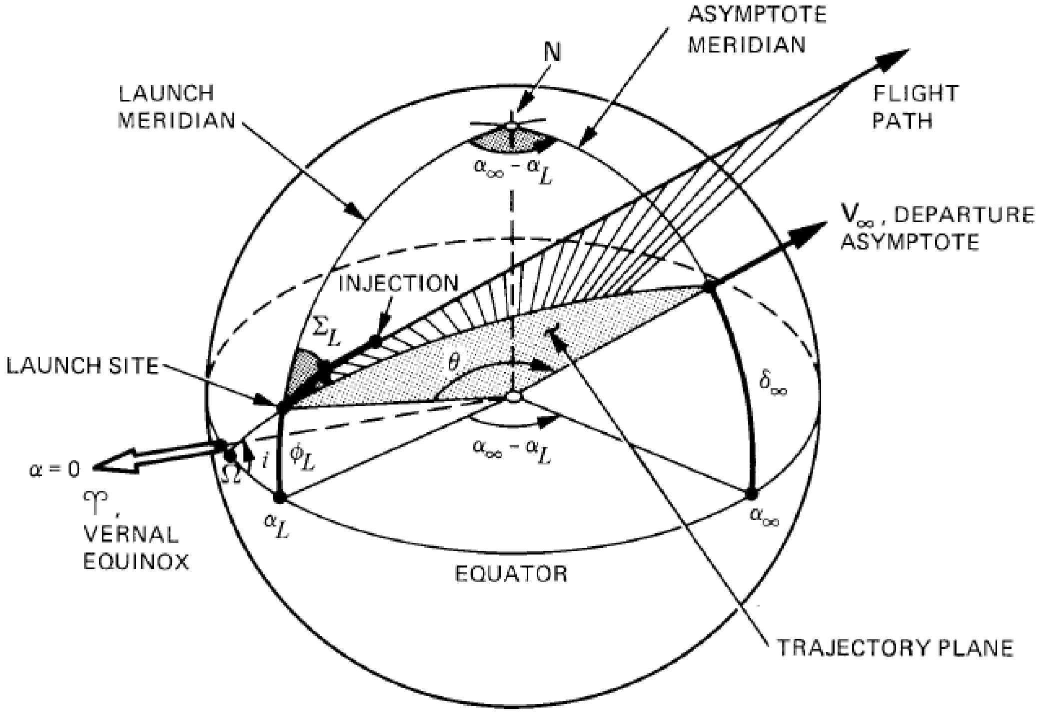
`raan` = right ascension of the ascending node

The output shown for `ascending node to launch site` and `ascending node to asymptote` are the angles measured from the ascending node *along the equator* to the launch site at the launch time and launch hyperbola asymptote, respectively. Also, since the calculations assume an initial circular park orbit, the true anomaly on the hyperbola at injection is identically zero (injection occurs at perigee) and the argument of latitude of the maneuver is equal to the argument of perigee.

Technical discussion

This section is a brief summary of the geometry and orbital mechanics involved in the design and analysis of interplanetary injection trajectories.

The following diagram, extracted from "JPL Publication 82-43, Interplanetary Mission Design Handbook", illustrates the basic geometry of the interplanetary launch problem. In this figure, α represents inertial right ascension measured relative to the Vernal Equinox, δ represents geocentric declination measured relative to the Earth's equator, Σ_L is the launch azimuth measured positive clockwise from true north, Ω is the right ascension of the ascending node (RAAN), and ϕ_L is the geocentric latitude of the launch site. Items with an "infinity" subscript refer to the hyperbolic asymptote of the outgoing or departure trajectory.



Departure trajectory orientation

The transfer trajectory must contain the launch site, the outgoing asymptote of the departure hyperbola and the center of the Earth at the moment of launch. The constraining equation for this orientation is

$$\hat{\mathbf{h}} \bullet \hat{\mathbf{s}} = \hat{h}_x \hat{s}_x + \hat{h}_y \hat{s}_y + \hat{h}_z \hat{s}_z = 0$$

where $\hat{\mathbf{h}}$ is the unit angular momentum of the launch hyperbola and $\hat{\mathbf{s}}$ is a unit vector in the direction of the outgoing asymptote. This constraint forces the angular momentum vector of the departure trajectory to be perpendicular to the outgoing asymptote.

The departure asymptote unit vector 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

$\alpha_\infty =$ right ascension of departure asymptote

$\delta_\infty =$ declination of departure asymptote

Furthermore, the angular momentum unit vector must also satisfy the following dot product constraint

$$\hat{\mathbf{h}} \bullet \hat{\mathbf{h}} = \hat{h}_x^2 + \hat{h}_y^2 + \hat{h}_z^2 = 1$$

These two equations can be solved simultaneously to yield the following two equations for the x and y components of the unit angular momentum vector.

$$\hat{h}_x = -\frac{\left(\hat{h}_y \hat{s}_y + \hat{h}_z \hat{s}_z\right)}{\hat{s}_x}$$

$$\hat{h}_y = -\frac{\hat{h}_z \hat{s}_y \hat{s}_z \pm \hat{s}_x \sqrt{1 - \hat{s}_z^2 - \hat{h}_z^2}}{\hat{s}_x^2 + \hat{s}_y^2}$$

The positive and negative sign in front of the square root in the second equation indicates that there are two possible launch time solutions. One solution corresponds to a short transfer and the other represents a long transfer mission.

The z-component of the unit angular momentum vector can be determined from the orbital inclination *i* <u>or</u> the flight azimuth Σ_L and launch site geocentric declination δ_L according to

$$\hat{h}_z = \cos i = \cos \delta_L \sin \Sigma_L$$

Notice that it is physically impossible to have an orbit in which

$$h_z^2 > 1 - s_z^2$$

If the declination of the launch hyperbola is greater than the launch site latitude, there is a range of azimuths (symmetric about east) at which launching is not possible.

Therefore, launch azimuth is restricted such that the following constraint must be satisfied

$$\sin^2 \Sigma_L \le \frac{\cos^2 \delta_\infty}{\cos^2 \delta_L}$$

Furthermore, the launch azimuth *sector* is symmetric about east with the following lower and upper limits

$$\sin^2 \Sigma_{L_{LIMIT}} = \pm \frac{\cos^2 \delta_\infty}{\cos^2 \delta_L}$$

Launch time

When the launch site passes through the departure trajectory plane, the following two geometric conditions must be satisfied

$$\hat{\mathbf{r}}_L \bullet \hat{\mathbf{h}} = \hat{r}_{L_x} \hat{h}_x + \hat{r}_{L_y} \hat{h}_y + \hat{r}_{L_z} \hat{h}_z = 0$$

$$\left(\hat{\mathbf{h}} \times \hat{\mathbf{r}}_L\right) \bullet \hat{\mathbf{i}}_z = \cos \sigma = \sqrt{\cos^2 \delta_L - \hat{h}_z^2}$$

where $\hat{\mathbf{r}}_L$ is the inertial unit position vector of the launch site and $\hat{\mathbf{i}}_z$ is a unit vector aligned with the Earth's spin axis and is given by $\hat{\mathbf{i}}_z = [0,0,1]^T$.

The inertial unit position vector of the launch site can be determined from

$$\hat{\mathbf{r}}_L = \begin{Bmatrix} \cos\delta_L \cos\theta_L \\ \cos\delta_L \sin\theta_L \\ \sin\delta_L \end{Bmatrix}$$

where θ_L is the *local sidereal time* of the launch site at the time of launch t_L. The local sidereal time at the launch time can be determined from the following expression

$$\theta_L = \theta_{g_0} + \lambda_L^E + \omega_e t_L$$

where

$\theta_{g_0} =$ Greenwich sidereal time at 0 hours UT

$\lambda_L^E =$ east longitude of launch site

$\omega_e =$ inertial rotation rate of the Earth

The local sidereal time is identical to the right ascension of a ground site.

The previous two constraint equations can be solved simultaneously to yield the following equations for the right ascension of the launch site at the launch time

$$\sin\alpha_L = \frac{\hat{h}_y \hat{h}_z \sin\delta_L - \hat{h}_x \sqrt{\cos^2\delta_L - \hat{h}_z^2}}{\left(\hat{h}_z^2 - 1\right)\cos\delta_L}$$

$$\cos\alpha_L = \frac{\hat{h}_z \hat{h}_x \sin\delta_L + \hat{h}_y \sqrt{\cos^2\delta_L - \hat{h}_z^2}}{\left(\hat{h}_z^2 - 1\right)\cos\delta_L}$$

These two equations should be evaluated with a four quadrant inverse tangent function to determine the right ascension of the launch site. The right ascension of the launch site as a function of launch azimuth is given by these next two equations

$$\sin\alpha_L = \frac{\hat{h}_y \sin\delta_L \sin\Sigma_L - \hat{h}_x \cos\Sigma_L}{\hat{h}_z^2 - 1}$$

$$\cos\alpha_L = \frac{\hat{h}_x \sin\delta_L \sin\Sigma_L + \hat{h}_y \cos\Sigma_L}{\hat{h}_z^2 - 1}$$

The three components of the unit angular momentum vector are functions of the launch azimuth and the components of the outgoing asymptote unit vector. Therefore, for a given geographic launch site, the launch time is determined by the launch azimuth and the declination and right ascension of the outgoing asymptote.

The right ascension of the launch site at the launch time is also given by

$$\alpha_L = \theta_{g_0} + \lambda_L^E + \omega_e t_L$$

Finally, this equation can be solved for the launch time to yield

$$t_L = \frac{\alpha_L - \theta_{g_0} - \lambda_L^E}{\omega_e}$$

Launch azimuth

For a given launch site, launch time and outgoing asymptote, the corresponding flight azimuth can be computed from the following two equations

$$\sin \Sigma_L = \frac{\cos i}{\cos \delta_L}$$

$$\cos \Sigma_L = \sin i \cos\left(\alpha_{\mho} - \alpha_L\right)$$

and the four quadrant inverse tangent

$$\Sigma_L = \tan^{-1}\left(\sin \Sigma_L, \cos \Sigma_L\right)$$

where $\alpha_{\mho}$ is the right ascension of the *descending* node which can be determined from the following expression

$$\alpha_{\mho} = \tan^{-1}\left(\hat{n}_y, \hat{n}_x\right)$$

In this equation, $\hat{\mathbf{n}}$ is a unit vector pointing in the direction of the descending node. This vector can be computed from the following cross product

$$\hat{\mathbf{n}} = \hat{\mathbf{h}} \times \hat{\mathbf{i}}_z$$

where $\hat{\mathbf{i}}_z$ is a unit vector aligned with the Earth's spin axis and is given by $\hat{\mathbf{i}}_z = \left[0, 0, 1\right]^T$.

Range angle and park orbit coast time estimate

The range angle is measured in the orbit plane from the launch site to the outgoing asymptote and can be determined from

$$\sin\eta = \frac{\sin(\alpha_\infty - \alpha_L)\cos\delta_\infty}{\sin\Sigma_L}$$

$$\cos\eta = \sin\delta_\infty \sin\phi_L + \cos\delta_\infty \cos\phi_L \cos(\alpha_\infty - \alpha_L)$$

and the four quadrant inverse tangent

$$\eta = \tan^{-1}(\sin\eta, \cos\eta)$$

The range angle is also equal to the following combination of Earth central angles and true anomalies

$$\eta = \theta_1 + \sum_{i=2}^{N} \theta_i + \theta_c + \theta_\infty - \theta_I$$

where

$\theta_1 =$ launch-to-park orbit inject Earth central angle

$\theta_i =$ Earth central angle for i^{th} propulsive or coast maneuver

$\theta_c =$ park orbit coast Earth central angle

$\theta_\infty =$ true anomaly of outgoing asymptote

$\theta_I =$ true anomaly at interplanetary injection

and N is the total number of additional propulsive or coasting maneuvers. The Earth central angle is the geocentric angle measured along the orbit plane from the beginning to the end of an orbital event such as a coast or maneuver.

The cosine of the range angle is also equal to the dot product of the unit inertial position vector of the launch site and the unit outgoing asymptote vector which is defined in the next section. Please see Appendix B for additional information and a MATLAB function that computes the range angle using this approach.

The park orbit coast Earth central angle is computed from

$$\theta_c = \eta - \theta_1 - \sum_{i=2}^{N} \theta_i - \theta_\infty + \theta_I$$

The true anomaly of the outgoing asymptote is given by $\theta_\infty = \cos^{-1}(-1/e)$ and determines where perigee of the departure trajectory is located relative to the outgoing asymptote.

Finally, the coast time can be estimated from $t_c = \theta_c / n$, where n is the two-body or Keplerian (unperturbed) mean motion of the park orbit and is equal to $\sqrt{\mu / r_{po}^3}$ and r_{po} is the geocentric radius of the initial circular Earth park orbit.

Departure hyperbola orbital characteristics

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 geocentric 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 α_∞ and declination δ_∞ 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 Σ_L and launch site geocentric latitude ϕ_L from the equation

$$i = \cos^{-1}\left(\cos\phi_L \sin\Sigma_L\right)$$

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

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

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

$\alpha_\infty =$ right ascension of departure asymptote

$\delta_\infty =$ declination of departure asymptote

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 $\hat{\mathbf{u}}_z = \left[0\;0\;1\right]^T$ 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

$$\cos\theta_\infty = -\frac{\mu}{r_p V_\infty^2 + \mu}$$

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

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

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

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.

An alternative algorithm for computing launch time

This section describes a slightly different algorithm for computing launch time. It is based on informal notes provided by Johnny Kwok at JPL.

The equations for computing the classical orbital elements of the departure hyperbola are as follows:

semimajor axis

$$a = -\mu/C_3$$

orbital eccentricity

$$e = 1 - \frac{r_{po}}{a}$$

orbital inclination

$$i = \cos^{-1}\left(\cos\phi_L \sin\Sigma_L\right)$$

argument of perigee

$$\omega = c - \theta_\infty$$

for injection during the *decending* part of the park orbit

$$c = \sin^{-1}\left(\frac{\sin\delta_L}{\sin i}\right)$$

for injection during the *ascending* part of the park orbit

$$c = 180^\circ - \sin^{-1}\left(\frac{\sin\delta_L}{\sin i}\right)$$

right ascension of the ascending node

$$\Omega = \alpha_\infty - h$$

where h is the angle from the ascending node to the launch asymptote projection on the equator, measured along the equator and determined from

$$h = \tan^{-1}\left(\sin c \sin i, \cos c\right) = \alpha_\infty - \Omega$$

The launch time can be determined from this next set of calculations:

The launch site sidereal time at 0 hours UTC (Universal Coordinated Time) is computed using

$$\alpha_0 = \theta_{g0} + \lambda_L^E \, .$$

The launch site local sidereal time at launch is given by $\alpha_L = \alpha_\infty - h + g$ where g is the angle from the ascending node to the launch site, measured along the equator and determined from

$$g = \tan^{-1}\left(\sin d \cos i, \cos d\right) = \alpha_L - \Omega$$

For a launch azimuth $\Sigma_L < 90^\circ$, the argument of latitude of the launch site d is given by

$$d = \sin^{-1}\left(\frac{\sin\phi_0}{\sin i}\right)$$

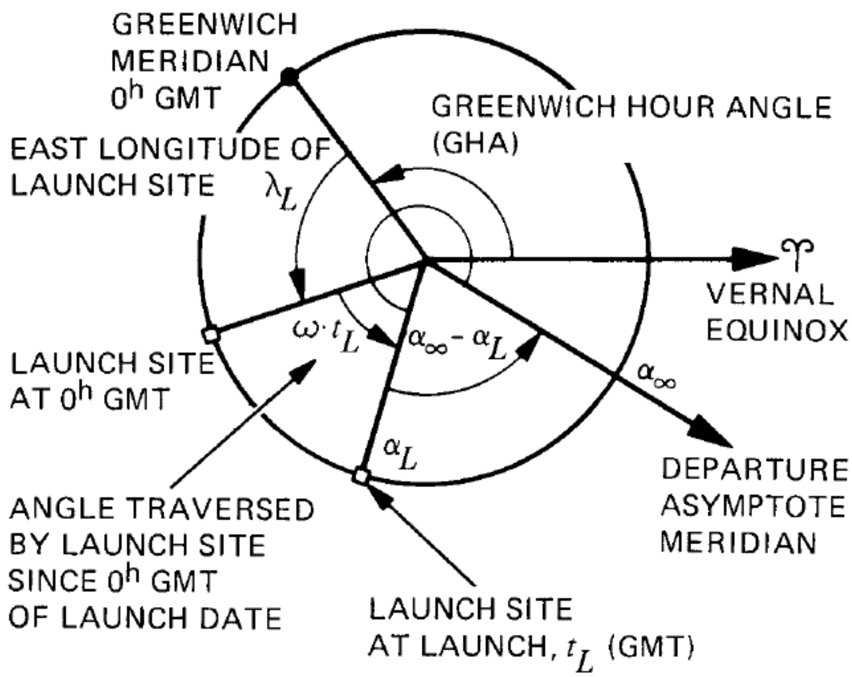
Otherwise

$$d = 180^\circ - \sin^{-1}\left(\frac{\sin\phi_0}{\sin i}\right).$$

Finally, the launch time can be determined from

$$t_L = \frac{\alpha_L - \alpha_0}{\omega_e}$$

The following diagram, also extracted from "JPL Publication 82-43, Interplanetary Mission Design Handbook", illustrates the important angular quantities and relative geometry involved in the interplanetary launch problem. Please note that all angles are measured in the Earth's equatorial plane.



In this diagram, GMT should be replaced with the modern day definition, UTC.

Geocentric Declination of the Launch Site

This MATLAB script uses the following equation to calculate the geocentric declination of the launch site given the geodetic latitude ϕ_L and geodetic altitude *h*.

$$\delta_L = \phi_L + \left(\frac{-\sin 2\phi_L}{\hat{h}+1}\right) f + \left\{\frac{-\sin 2\phi_L}{2\left(\hat{h}+1\right)^2} + \left[\frac{1}{4\left(\hat{h}+1\right)^2} + \frac{1}{4\left(\hat{h}+1\right)}\right]\sin 4\phi_L\right\} f^2$$

In this equation, the geocentric distance *r* and geodetic altitude *h* have been normalized by $\hat{\rho} = r / r_{eq}$ and $\hat{h} = h / r_{eq}$, respectively. r_{eq} is the equatorial radius of the Earth and *f* is the flattening factor.

Spherical Equations

This section provides additional equations that may be useful for interplanetary mission design and analysis. For example, they can be used to determine the characteristics of a launch hyperbola from the classical orbital elements at the target interface point (TIP).

The asymptote unit vector in terms of the classical orbital elements of a hyperbolic orbit is given by

$$\hat{\mathbf{s}} = \begin{Bmatrix} \cos\Omega\cos(\omega+\theta) - \sin\Omega\sin(\omega+\theta)\cos i \\ \sin\Omega\cos(\omega+\theta) + \cos\Omega\sin(\omega+\theta)\cos i \\ \sin(\omega+\theta)\sin i \end{Bmatrix}$$

In this expression, Ω is the right ascension of the ascending node (RAAN), ω is the argument of periapsis and θ is the true anomaly.

The declination of the asymptote (DLA) is given by

$$\delta_\infty = \sin^{-1}\left[\sin(\omega+\theta_\infty)\sin i\right] = \sin^{-1}\left[\sin(u_\infty)\sin i\right]$$

where $u_\infty = \omega + \theta_\infty$ is the argument of latitude of the launch asymptote. In this expression θ_∞ is the true anomaly of the launch hyperbola "at infinity" and is a function of the orbital eccentricity e of the hyperbola according to $\theta_\infty = \cos^{-1}(-1/e)$.

Finally, from the following two expressions

$$\sin(\alpha_\infty - \Omega) = \frac{\tan\delta_\infty}{\tan i}$$

$$\cos(\alpha_\infty - \Omega) = \frac{\cos u_\infty}{\cos\delta_\infty}$$

the right ascension of the asymptote (RLA) can be determined from

$$\alpha_\infty = \Omega + \tan^{-1}\left(\frac{\tan\delta_\infty}{\tan i}, \frac{\cos u_\infty}{\cos\delta_\infty}\right)$$

where the inverse tangent in this equation is a four quadrant operation.

Algorithm resources

(1) “Design of Lunar and Interplanetary Ascent Trajectories”, Victor C. Clarke, Jr., JPL Technical Report No. 32-30, March 15, 1962.

(2) *An Introduction to the Mathematics and Methods of Astrodynamics*, Richard H. Battin, AIAA Education Series, 1987.

(3) *Analytical Mechanics of Space Systems*, Hanspeter Schaub and John L. Junkins, AIAA Education Series, 2003.

(4) *Spacecraft Mission Design*, Charles D. Brown, AIAA Education Series, 1992.

(5) *Orbital Mechanics*, Vladimir A. Chobotov, AIAA Education Series, 2002.

(6) “A Computer Simulation of the Orbital Launch Window Problem”, Archie C. Young and Pat R. Odom, AIAA 67-615, 1967.

(7) “Launch Parameters for Interplanetary Flights”, W. C. Riddell, *American Rocket Society Journal*, December 1960.

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

(9) “Launch-on-Time Analysis for Space Missions”, JPL Technical Report No. 32-29, August 15, 1960.

Appendix A

Greenwich Apparent Sidereal Time

This appendix describes the algorithm used in this MATLAB script to calculate Greenwich apparent sidereal time. This numerical method calculates the apparent Greenwich sidereal time using the first few terms of the IAU 1980 nutation algorithm.

The Greenwich apparent sidereal time is given by the expression

$$\theta = \theta_m + \Delta\psi \cos\left(\varepsilon_m + \Delta\varepsilon\right)$$

where θ_m is the Greenwich mean sidereal time, $\Delta\psi$ is the nutation in longitude, ε_m is the mean obliquity of the ecliptic and $\Delta\varepsilon$ is the nutation in obliquity.

The Greenwich <u>mean</u> sidereal time is calculated using the expression

$$\theta_m = 280.46061837 + 360.98564736629\left(JD - 2451545.0\right) + 0.000387933T^2 - T^3 / 38710000$$

where $T = (JD - 2451545)/36525$ and JD is the Julian date. The mean obliquity of the ecliptic is determined from

$$\varepsilon_m = 23^0 26' 21.''448 - 46.''8150T - 0.''00059T^2 + 0.''001813T^3$$

The nutation in obliquity and longitude involves the following three trigonometric arguments (in degrees)

$$L = 280.4665 + 36000.7698T$$

$$L' = 218.3165 + 481267.8813T$$

$$\Omega = 125.04452 - 1934.136261T$$

The calculation of nutation uses the following two equations

$$\Delta\psi = -17.20\sin\Omega - 1.32\sin 2L - 0.23\sin 2L' + 0.21\sin 2\Omega$$

$$\Delta\varepsilon = 9.20\cos\Omega + 0.57\cos 2L + 0.10\cos 2L' - 0.09\cos 2\Omega$$

These corrections are in units of arc seconds.

The following is the MATLAB source code for this function.

```
function gst = gast1 (jdate)

% Greenwich apparent sidereal time

% major nutation terms only

% input
```

```
%  jdate = Julian date

% output

%  gst = Greenwich apparent sidereal time (radians)
%        (0 <= gst <= 2 pi)

% Orbital Mechanics with Matlab

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

pi2 = 2.0 * pi;

% conversion factors

dtr = pi/180;

atr = dtr/3600;

% time arguments

t = (jdate - 2451545) / 36525;

t2 = t * t;

t3 = t * t2;

% fundamental trig arguments

l = mod(dtr * (280.4665 + 36000.7698 * t), pi2);

lp = mod(dtr * (218.3165 + 481267.8813 * t), pi2);

lraan = mod(dtr * (125.04452 - 1934.136261 * t), pi2);

% nutations in longitude and obliquity

dpsi = atr * (-17.2 * sin(lraan) - 1.32 * sin(2 * l) ...
       - 0.23 * sin(2 * lp) + 0.21 * sin(2 * lraan));

deps = atr * (9.2 * cos(lraan) + 0.57 * cos(2 * l) ...
       + 0.1 * cos(2 * lp) - 0.09 * cos(2 * lraan));

% mean and apparent obliquity of the ecliptic

eps0 = mod(dtr * (23 + 26 / 60 + 21.448 / 3600) ...
       + atr * (-46.815 * t - 0.00059 * t2 + 0.001813 * t3), pi2);

obliq = eps0 + deps;

% greenwich mean and apparent sidereal time

gstm = mod(dtr * (280.46061837 + 360.98564736629 * (jdate - 2451545) ...
       + 0.000387933 * t2 - t3 / 38710000), pi2);

gst = mod(gstm + dpsi * cos(obliq), pi2);
```

Appendix B

Range Angle Calculation

This appendix describes a technique for computing the range angle using unit vectors. The two unit vectors are assumed to be coplanar.

The cosine of the transfer or "included" angle between two coplanar unit vectors measured relative to the same celestial body is given by

$$\cos\psi = \hat{\mathbf{u}}_1 \bullet \hat{\mathbf{u}}_2$$

The sine of the transfer angle is given by

$$\sin\psi = sign\left|(\hat{\mathbf{u}}_1 \times \hat{\mathbf{u}}_2)\bullet\hat{\mathbf{k}}\right|\sqrt{1-\cos^2\psi}$$

where $\hat{\mathbf{k}} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$.

Finally, the transfer angle is calculated using the following expression

$$\psi = \tan^{-1}(\sin\psi, \cos\psi)$$

where the inverse tangent is a four quadrant calculation.

The following is the MATLAB source code for this function.

```
function rangle = range_angle(uhat1, uhat2)

% range angle between two coplanar unit vectors

% input

%  uhat1 = first unit vector
%  uhat2 = second unit vector

% output

%  rangle = included angle between first and second unit vectors (radians)

% NOTE: direction is from uhat1 to uhat2

% Orbital Mechanics with Matlab

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% unit "normal" vector

xkhat(1) = 0.0;
xkhat(2) = 0.0;
xkhat(3) = 1.0;
```

```
% cosine of range angle

ctangle = dot(uhat1, uhat2);

r1xr2 = cross(uhat1, uhat2);

vdotwrk = dot(r1xr2, xkhat);

% sine of range angle

stangle = sign(vdotwrk) * sqrt(1.0 - ctangle^2);

% range angle (radians)

rangle = atan3(stangle, ctangle);
```