Gravity-Assist Trajectory Optimization – SNOPT

This document describes a MATLAB script named `flyby_snopt` that can be used to optimize interplanetary, patched-conic trajectories that include a zero sphere of influence (ZSOI) single gravity assist maneuver. The user specifies the departure, flyby and arrival planets along with initial guesses for the departure, flyby and arrival calendar dates, and lower and upper bounds for the flyby altitude. This script searches for a patched-conic, gravity-assist trajectory that satisfies the flyby mission constraints (V_∞ matching and bounded flyby altitude) and minimizes the departure, arrival or total *impulsive* heliocentric delta-v for the mission. The type of trajectory optimization is specified by the user.

The software provides a simple two-dimensional graphic display of the heliocentric planet orbits and the interplanetary transfer trajectory. It also provides a three-dimensional graphic display of the flyby trajectory relative to the gravity-assist planet along with the corresponding classical orbital elements.

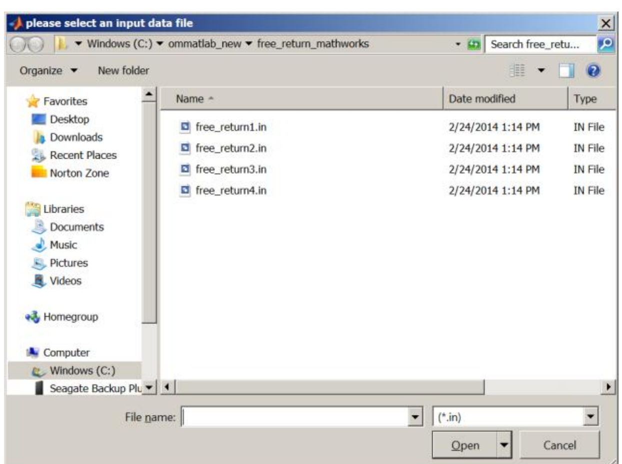
This MATLAB script reads JPL lunar and solar ephemerides in a machine-independent binary format (kernels) which are available from the SPICE web site and by anonymous ftp from ftp://ssd.jpl.nasa.gov/pub/eph/planets/bsp. These `*.bsp` ephemeris files are IEEE-Little Endian style of binary kernel. This is the binary form native to PC/Linux, PC/Windows and MAC/Intel machines. Additional information about JPL ephemerides can be found at http://naif.jpl.nasa.gov/naif/.

The `flyby_snopt` script uses routines from the MICE software suite to read and evaluate the JPL ephemeris file. *Platform-specific* MICE mex files, support functions and binary ephemeris files are available at naif.jpl.nasa.gov/naif/toolkit_MATLAB.html. MICE is a MATLAB implementation of the SPICE library created by JPL.

In this MATLAB script the optimization is performed using the SNOPT nonlinear programming (NLP) algorithm. Information about MATLAB versions of SNOPT for several computer platforms can be found at Professor Philip Gill's web site which is located at https://ccom.ucsd.edu/~optimizers.

User interaction with the script

This MATLAB script is data-driven by a text file created by the user. After typing `flyby_snopt` in the MATLAB command window, the software will ask you for the name of this simulation definition input data file with a screen display similar to



The file type defaults to names with a `*.in` filename extension. However, you can select any `flyby_snopt` compatible ASCII data file by selecting the Files of type: field or by typing the name of the file directly in the File name: field.

Input data file format and contents

The input for the `flyby_snopt` script is a simple text file created by the user. This section describes a typical simulation definition input data file for the software.

Each data item within an input file is preceded by one or more lines of *annotation* text. Unless you edit the source code, 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.

The software allows the user to specify an initial guess for the departure, flyby and arrival calendar dates, and lower and upper bounds on the actual departure, flyby and arrival calendar dates found during the optimization process. For any guess for departure time t_L and user-defined departure time lower and upper bounds Δt_l and Δt_u, the departure time t is constrained as follows:

$$t_L - \Delta t_l \le t \le t_L + \Delta t_u$$

Likewise, for any guess for arrival time t_A and user-defined arrival time bounds Δt_l and Δt_u, the arrival time t is constrained as follows:

$$t_A - \Delta t_l \le t \le t_A + \Delta t_u$$

For fixed departure, flyby or arrival times, the lower and upper bounds should be set to `0.001`.

The following is a typical simulation definition file named `flyby_Earth_Venus_Mars_2023.in`. It models an Earth to Mars mission in 2023 with a Venus gravity-assist. Required user inputs are shown in bold font

```
*****************************************
** gravity-assist trajectory optimization
** MATLAB script ==> flyby_snopt.m
** Earth-Venus-Mars trajectory
** flyby_Earth_Venus_Mars_2023.in
*****************************************

      optimization menu
      =================

 <1> minimize departure delta-v

 <2> minimize arrival delta-v

 <3> minimize total delta-v

selection (1, 2 or 3)
1

departure calendar date guess
-----------------------------

(1 <= month <= 12, 1 <= day <= 31, year = all digits!)
9, 14, 2023

departure date search boundary in days
30
```

```
departure planet

  <1> Mercury
  <2> Venus
  <3> Earth
  <4> Mars
  <5> Jupiter
  <6> Saturn
  <7> Uranus
  <8> Neptune
  <9> Pluto

selection (1-9)
3

flyby calendar date guess
-------------------------

(1 <= month <= 12, 1 <= day <= 31, year = all digits!)
2, 10, 2024

flyby date search boundary in days
30

 flyby planet

  <1> Mercury
  <2> Venus
  <3> Earth
  <4> Mars
  <5> Jupiter
  <6> Saturn
  <7> Uranus
  <8> Neptune
  <9> Pluto

selection (1-9)
2

lower bound for flyby altitude (kilometers)
500

upper bound for flyby altitude (kilometers)
10000

arrival calendar date guess
---------------------------

(1 <= month <= 12, 1 <= day <= 31, year = all digits!)
7, 16, 2024

arrival date search boundary in days
30

 arrival celestial body

  <1> Mercury
  <2> Venus
  <3> Earth
  <4> Mars
  <5> Jupiter
  <6> Saturn
```

```
  <7> Uranus
  <8> Neptune
  <9> Pluto

selection (1-9)
4
```

Numerical solution and trajectory graphics

This section summarizes the program output for this example. The software provides the heliocentric orbital elements and state vectors of each leg of the transfer trajectory in the Earth mean ecliptic and equinox of J2000 coordinate system described in Appendix E. These results also include the characteristics of the hyperbolic flyby trajectory with respect to the flyby planet as well as the trajectory characteristics at the time of entrance to and exit from the sphere-of-influence (SOI) relative to the flyby planet. The time scale is Barycentric Dynamical Time (TDB). See Appendix A for a description of this numerical information.

```
flyby_snopt - patched-conic gravity assist - SNOPT version
==========================================================

input data file ==> flyby_Earth_Venus_Mars_2023.in

minimize departure delta-v

departure planet   Earth
flyby planet       Venus
arrival planet     Mars

DEPARTURE CONDITIONS
====================

TDB calendar date     06-Sep-2023
TDB time              10:31:20.965
TDB Julian day        2460193.9384371

heliocentric departure delta-v vector and magnitude
(mean ecliptic and equinox of J2000)
------------------------------------

departure delta-vx         -1607.032972 meters/second
departure delta-vy         -4668.000844 meters/second
departure delta-vz            47.344719 meters/second

departure delta-v           4937.107288 meters/second

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

asymptote right ascension    249.514983 degrees

asymptote declination        -21.549021 degrees

departure energy              24.375028 kilometer^2/second^2

spacecraft post-impulse heliocentric orbital elements and state vector
(mean ecliptic and equinox of J2000)
------------------------------------

        sma (km)              eccentricity        inclination (deg)         argper (deg)
 +1.14986567608479e+08  +3.12443617249927e-01  +1.04516142084132e-01  +1.77071440116458e+02

       raan (deg)         true anomaly (deg)        arglat (deg)           period (days)
 +3.43032945683014e+02  +1.83160089137690e+02  +2.31529254148253e-01  +2.46139307453156e+02

        rx (km)                 ry (km)                 rz (km)               rmag (km)
 +1.44421525017069e+08  -4.34261586559188e+07  +1.11165577450208e+03  +1.50809177920645e+08
```

```
       vx (kps)                vy (kps)               vz (kps)              vmag (kps)
 +6.49558656266758e+00  +2.37414012645204e+01  +4.48806463308101e-02  +2.46139958777215e+01

departure planet heliocentric orbital elements and state vector
(mean ecliptic and equinox of J2000)
------------------------------------

       sma (km)              eccentricity       inclination (deg)         argper (deg)
 +1.49575888593219e+08  +1.64691435392948e-02  +4.79205681076953e-03  +2.95800134587807e+02

      raan (deg)          true anomaly (deg)        arglat (deg)          period (days)
 +1.68320738148593e+02  +2.39143601797964e+02  +1.74943736385771e+02  +3.65176394327263e+02

       rx (km)                 ry (km)                rz (km)               rmag (km)
 +1.44421525017069e+08  -4.34261586559188e+07  +1.11165577450208e+03  +1.50809177920645e+08

       vx (kps)                vy (kps)               vz (kps)              vmag (kps)
 +8.10261953488215e+00  +2.84094021087236e+01  -2.46407265184345e-03  +2.95422845693060e+01
```

PATCHED-CONIC FLYBY CONDITIONS

```
==============================

TDB calendar date     15-Feb-2024
TDB time              02:56:03.364
TDB Julian day        2460355.6222612

departure-to-flyby time      161.683824 days

v-infinity in             11083.236329 meters/second
v-infinity out            11083.236334 meters/second

flyby altitude             4729.749013 kilometers

maximum turn angle           35.408043 degrees
actual turn angle            22.719984 degrees

heliocentric delta-v       4366.192082 meters/second

max heliocentric delta-v   7326.580266 meters/second

spacecraft planet-centered orbital elements and state vector at periapsis
(mean ecliptic and equinox of J2000)
------------------------------------

       sma (km)              eccentricity       inclination (deg)         argper (deg)
 -2.64460722051835e+03  +5.07684321861030e+00  +1.54105271352839e+01  +5.72396443565961e+01

      raan (deg)          true anomaly (deg)        arglat (deg)
 +1.17952843816249e+02  +0.00000000000000e+00  +5.72396443565961e+01

       rx (km)                 ry (km)                rz (km)               rmag (km)
 -1.04557688397713e+04  +1.05639507242989e+03  +2.40933245042609e+03  +1.07816490128581e+04

       vx (kps)                vy (kps)               vz (kps)              vmag (kps)
 -9.01503368816242e-01  -1.33604209644633e+01  +1.94575557183949e+00  +1.35314271759117e+01

spacecraft b-plane coordinates at periapsis
(mean ecliptic and equinox of J2000)
------------------------------------

b-magnitude                  13163.221836  kilometers
b dot r                      -2555.177141  kilometers
b dot t                      12912.841626  kilometers
b-plane angle                  348.806978  degrees
v-infinity                   11083.236329  meters/second
r-periapsis                  10781.649013  kilometers
decl-asymptote                  10.660837  degrees
rasc-asymptote                 254.880140  degrees

flight path angle                0.000000  degrees

spacecraft heliocentric orbital elements and state vector
```

```
(mean ecliptic and equinox of J2000)
------------------------------------

        sma (km)              eccentricity       inclination (deg)         argper (deg)
 +1.14986567608479e+08  +3.12443617249928e-01  +1.04516142084132e-01  +1.77071440116458e+02

       raan (deg)          true anomaly (deg)       arglat (deg)          period (days)
 +3.43032945683014e+02  +9.82624021479394e+01  +2.75333842264398e+02  +2.46139307453156e+02

        rx (km)                 ry (km)                 rz (km)               rmag (km)
 -2.19066166308027e+07  -1.06407616952762e+08  -1.97316440089486e+05  +1.08639402297473e+08

        vx (kps)                vy (kps)                vz (kps)              vmag (kps)
 +3.12260847459193e+01  -1.77185957677201e+01  -1.42920586878088e-02  +3.59028858004777e+01

flyby planet heliocentric orbital elements and state vector
(mean ecliptic and equinox of J2000)
------------------------------------

        sma (km)              eccentricity       inclination (deg)         argper (deg)
 +1.08208794041158e+08  +6.74588362162296e-03  +3.39439112574848e+00  +5.52962100119693e+01

       raan (deg)          true anomaly (deg)       arglat (deg)          period (days)
 +7.66120300493675e+01  +1.26461638698471e+02  +1.81757848710440e+02  +2.24700554576704e+02

        rx (km)                 ry (km)                 rz (km)               rmag (km)
 -2.19066166308025e+07  -1.06407616952762e+08  -1.97316440089487e+05  +1.08639402297473e+08

        vx (kps)                vy (kps)                vz (kps)              vmag (kps)
 +3.40671274605866e+01  -7.20371585990588e+00  -2.06463455198509e+00  +3.48815912974074e+01
```

ARRIVAL CONDITIONS

```
=================

TDB calendar date     16-Jun-2024
TDB time              00:00:00.000
TDB Julian day        2460477.5000000

flyby-to-arrival time        121.877739 days

heliocentric arrival delta-v vector and magnitude
(mean ecliptic and equinox of J2000)
------------------------------------

arrival delta-vx            -3323.760760 meters/second
arrival delta-vy             6225.317568 meters/second
arrival delta-vz              494.077761 meters/second

arrival delta-v              7074.325215 meters/second

arrival hyperbola characteristics
(Earth mean equator and equinox of J2000)
-----------------------------------------

asymptote right ascension    249.514983 degrees

asymptote declination        -21.549021 degrees

arrival energy                50.046077 kilometer^2/second^2

spacecraft pre-impulse heliocentric orbital elements and state vector
(mean ecliptic and equinox of J2000)
------------------------------------

        sma (km)              eccentricity       inclination (deg)         argper (deg)
 +1.55134310204787e+08  +3.93116940063928e-01  +1.45210940616138e+00  +1.25933805748784e+02

       raan (deg)          true anomaly (deg)       arglat (deg)          period (days)
 +7.42581214494069e+01  +1.60330990699542e+02  +2.86264796448326e+02  +3.85719946000429e+02

        rx (km)                 ry (km)                 rz (km)               rmag (km)
 +2.08178930929628e+08  +1.91800881289026e+06  -5.06611502429432e+06  +2.08249397507528e+08
```

```
        vx (kps)                vy (kps)                vz (kps)                vmag (kps)
 +4.02641386001630e+00  +2.00716547914920e+01  +3.98018538466537e-02  +2.04715636634949e+01

spacecraft post-impulse heliocentric orbital elements and state vector
(mean ecliptic and equinox of J2000)
------------------------------------

        sma (km)              eccentricity          inclination (deg)         argper (deg)
 +2.27932912278456e+08  +9.32833678797215e-02  +1.84785283228378e+00  +2.86688048489641e+02

       raan (deg)          true anomaly (deg)         arglat (deg)           period (days)
 +4.94891347477531e+01  +2.43359246912706e+01  +3.11023973180911e+02  +6.86943262036181e+02

        rx (km)                 ry (km)                 rz (km)                 rmag (km)
 +2.08178930929628e+08  +1.91800881289026e+06  -5.06611502429432e+06  +2.08249397507528e+08

        vx (kps)                vy (kps)                vz (kps)                vmag (kps)
 +7.02653100452780e-01  +2.62969723595310e+01  +5.33879614701090e-01  +2.63117750085490e+01

arrival planet heliocentric orbital elements and state vector
(mean ecliptic and equinox of J2000)
------------------------------------

        sma (km)              eccentricity          inclination (deg)         argper (deg)
 +2.27932912278455e+08  +9.32833678797216e-02  +1.84785283228378e+00  +2.86688048489641e+02

       raan (deg)          true anomaly (deg)         arglat (deg)           period (days)
 +4.94891347477531e+01  +2.43359246912705e+01  +3.11023973180911e+02  +6.86943262036180e+02

        rx (km)                 ry (km)                 rz (km)                 rmag (km)
 +2.08178930929628e+08  +1.91800881289035e+06  -5.06611502429432e+06  +2.08249397507528e+08

        vx (kps)                vy (kps)                vz (kps)                vmag (kps)
 +7.02653100452765e-01  +2.62969723595310e+01  +5.33879614701091e-01  +2.63117750085490e+01
```

MISSION SUMMARY

```
===============

total delta-v             12011.432503 meters/second

total energy                 74.421106 kilometer^2/second^2

total mission duration       283.561563 days

PATCHED-CONIC SOI ENTRANCE CONDITIONS
=====================================

TDB calendar date     14-Feb-2024
TDB time              11:29:24.548
TDB Julian day        2460354.9787563

spacecraft planet-centered orbital elements and state vector
(mean ecliptic and equinox of J2000)
------------------------------------

        sma (km)              eccentricity          inclination (deg)         argper (deg)
 -2.66586631936811e+03  +1.00009927772681e+00  +1.04035555483480e+02  +1.90180387531175e+02

       raan (deg)          true anomaly (deg)         arglat (deg)
 +2.57577435984032e+02  +1.80810804732837e+02  +1.09911922640122e+01

        rx (km)                 ry (km)                 rz (km)                 rmag (km)
 -1.57970611810468e+05  -5.84678529942140e+05  +1.13990361539001e+05  +6.16277129297257e+05

        vx (kps)                vy (kps)                vz (kps)                vmag (kps)
 +2.84170590593612e+00  +1.05183171955843e+01  -2.05000497354402e+00  +1.10866049570783e+01

spacecraft b-plane coordinates at sphere-of-influence
(mean ecliptic and equinox of J2000)
------------------------------------

b-magnitude                      37.565576  kilometers
b dot r                         -36.403735  kilometers
```

```
b dot t                          -9.270414  kilometers
b-plane angle                   255.712980  degrees
v-infinity                    11038.955932  meters/second
r-periapsis                       0.264661  kilometers
decl-asymptote                  -10.655777  degrees
rasc-asymptote                   74.881490  degrees

flight path angle               -89.996523  degrees

PATCHED-CONIC SOI EXIT CONDITIONS
=================================

TDB calendar date     15-Feb-2024
TDB time              18:22:42.853
TDB Julian day        2460356.2657738

spacecraft planet-centered orbital elements and state vector
(mean ecliptic and equinox of J2000)
------------------------------------

        sma (km)              eccentricity       inclination (deg)         argper (deg)
 -2.66606206666511e+03  +1.00025625470173e+00  +1.62184343393553e+02  +1.62828970403693e+02

       raan (deg)          true anomaly (deg)       arglat (deg)
 +7.96090127060637e+01  +1.78697438568347e+02  +3.41526408972041e+02

        rx (km)                 ry (km)                 rz (km)               rmag (km)
 -7.74381261481494e+04  +6.08466268747807e+05  -5.97464351536169e+04  +6.16277129300431e+05

        vx (kps)                vy (kps)                vz (kps)              vmag (kps)
 -1.39201141931487e+00  +1.09458527970707e+01  -1.07446411927525e+00  +1.10862014409921e+01
```

After the solution is displayed, the software will ask the user if he or she would like to create a graphics display of the heliocentric planet orbits and flyby trajectory with the following prompt:

```
would you like to display trajectory graphics for this mission (y = yes, n = no)
?
```

If the user’s response is y for yes, the script will request a plot step size for the interplanetary trajectories with

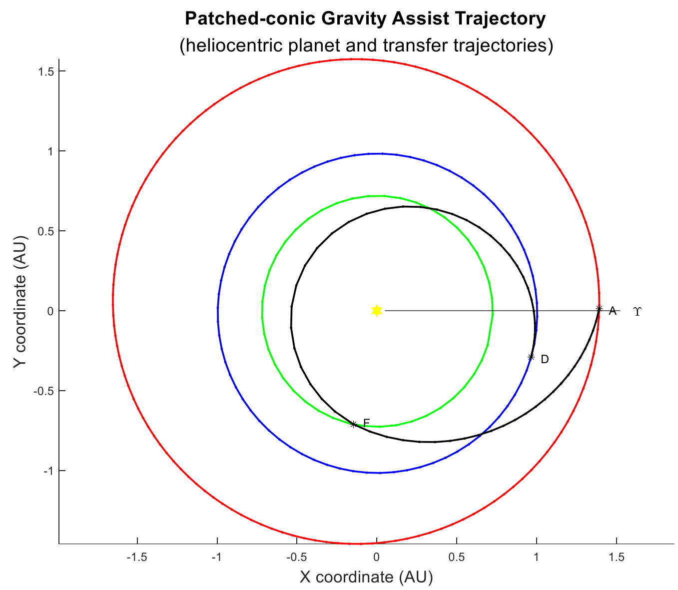
```
please input the plot step size (days)
?
```

The software will also ask for a plot span in days for the flyby graphics.

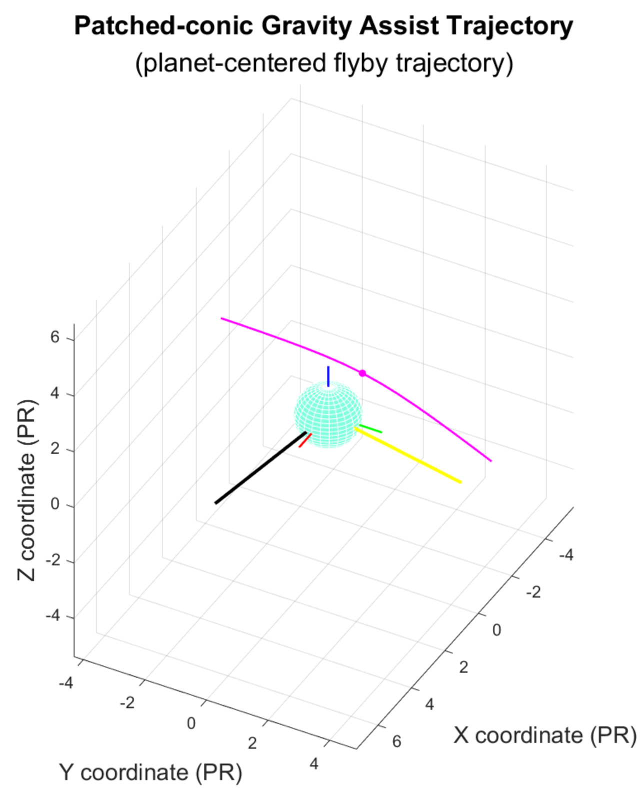
```
please input the flyby plot span (days)
?
```

The following is the heliocentric graphics display for this example. This plot is a *north ecliptic* view where we are looking down on the ecliptic plane from the north celestial pole. The vernal equinox direction is the labeled line pointing to the right, the departure planet is labeled with an L, the flyby planet is labeled with an F and the arrival planet is labeled with an A. The locations of the departure, flyby and arrival trajectory events are marked with an asterisk. Please note the x and y axes are labeled in Astronomical Units (AU).

The departure orbit is the blue trace, the flyby orbit the green trace and the arrival planet orbit is the red trace. The gravity-assist transfer trajectory is the black trace. The orbit traces are labeled with small dots at the step size provided by the user.

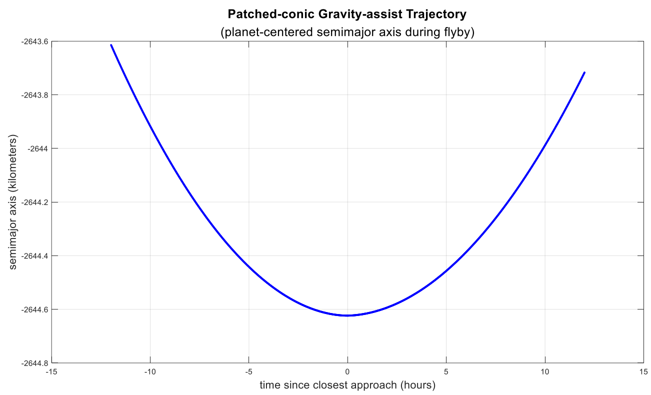


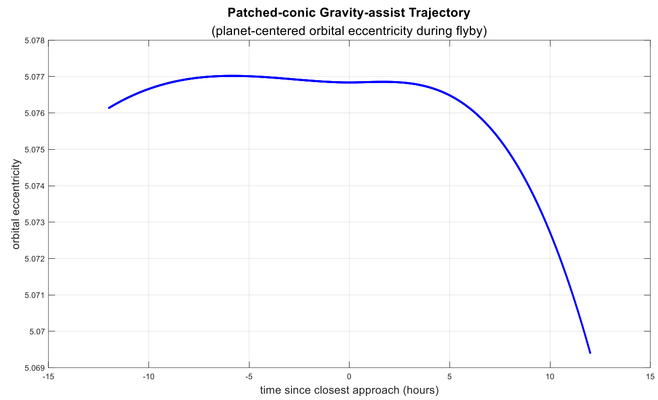
The following is the flyby graphics display for this example. The direction of the sun is the yellow line and the black line points in the direction of the heliocentric velocity vector of the flyby planet. This display is labeled with a planet-centered, inertial mean ecliptic and equinox of J2000 coordinate system. The x-axis of this system is red, the y-axis is green and the z-axis is blue. The flyby hyperbola is colored magenta with the incoming leg marked with an asterisk and periapsis labeled with a small circle. Please note that the axes are labeled in radii of the flyby planet (PR).

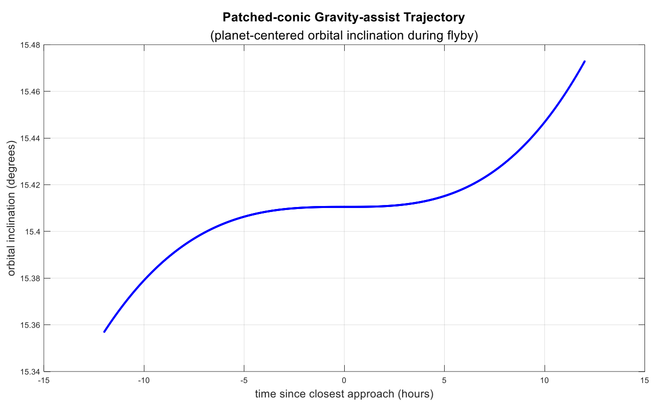


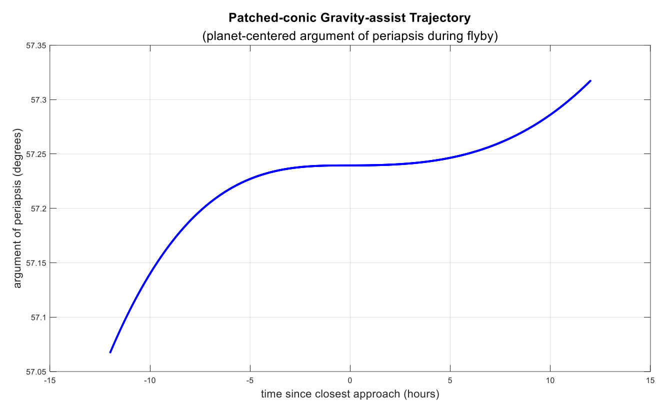
This script will also create `tif` disk files of these two graphic displays.

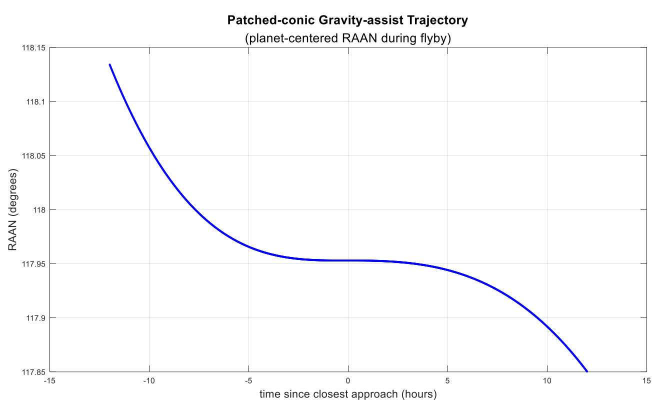
The script will also ask the user if he or she would like to create graphic displays of the flyby planet-centered classical orbital elements and speed during the flyby. Here are the graphics for this example.

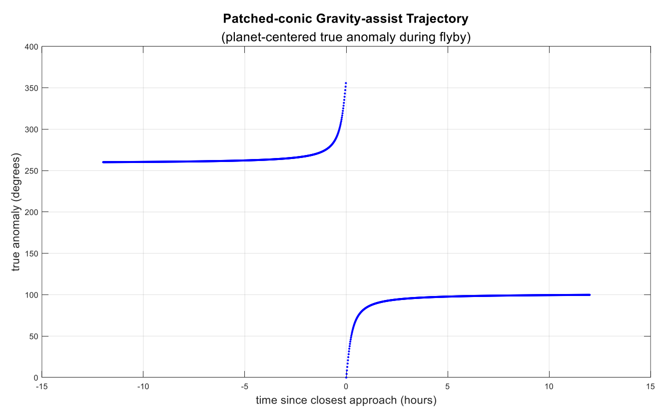

Patched-conic Gravity-assist Trajectory
(planet-centered semimajor axis during flyby)
semimajor axis (kilometers)
time since closest approach (hours)

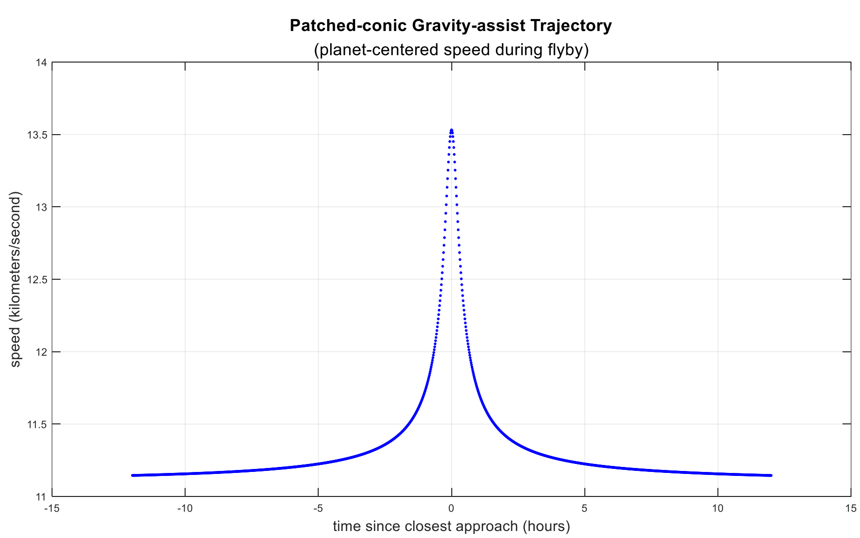

Patched-conic Gravity-assist Trajectory
(planet-centered orbital eccentricity during flyby)
orbital eccentricity
time since closest approach (hours)


Patched-conic Gravity-assist Trajectory
(planet-centered orbital inclination during flyby)
orbital inclination (degrees)
time since closest approach (hours)


Patched-conic Gravity-assist Trajectory
(planet-centered argument of periapsis during flyby)
argument of periapsis (degrees)
time since closest approach (hours)


Patched-conic Gravity-assist Trajectory
(planet-centered RAAN during flyby)
RAAN (degrees)
time since closest approach (hours)


Patched-conic Gravity-assist Trajectory
(planet-centered true anomaly during flyby)
true anomaly (degrees)
time since closest approach (hours)


Patched-conic Gravity-assist Trajectory
(planet-centered speed during flyby)
speed (kilometers/second)
time since closest approach (hours)

Technical discussion

The computational steps used to create an initial guess for the spacecraft state, and the departure and arrival delta-v characteristics are as follows:

(1) compute the state vector of the departure planet at the departure date initial guess

(2) compute the state vector of the flyby planet at the flyby date initial guess

(3) compute the state vector of the arrival planet at the arrival date initial guess

(4) solve Lambert's problem for the departure-to-flyby leg and determine the initial velocity vector of this leg

(5) compute the departure delta-v vector from the departure planet's velocity vector and the initial velocity vector of the first leg of the transfer trajectory

(6) solve Lambert's problem for the second heliocentric leg and determine the initial and final velocity vectors of the second leg

(7) compute the arrival delta-v vector from the arrival planet's velocity vector and the final velocity vector of the second leg of the transfer trajectory

(8) compute the flyby incoming v-infinity vector from the flyby planet's velocity vector and the final velocity vector of the first leg of the transfer trajectory

The Lambert 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. Gooding's 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. Appendix B provides information about Lambert's problem.

Gravity-assist flight mechanics

The vector relationship between the incoming v-infinity vector $\mathbf{v}_\infty^-$, the outgoing v-infinity vector $\mathbf{v}_\infty^+$ and the two legs of the heliocentric transfer orbit are as follows:

$$\mathbf{v}_\infty^- = \mathbf{v}_{fb} - \mathbf{v}_{to_1}$$

$$\mathbf{v}_\infty^+ = \mathbf{v}_{to_2} - \mathbf{v}_{fb}$$

where

$\mathbf{v}_{fb} =$ heliocentric velocity vector of the flyby planet at the flyby date

$\mathbf{v}_{to_1} =$ heliocentric velocity vector of the first transfer orbit at the flyby date

$\mathbf{v}_{to_2} =$ heliocentric velocity vector of the second transfer orbit at the flyby date

The turn angle of the planet-centered trajectory during the flyby is determined from

$$\delta = \frac{\pi}{2} - \frac{1}{2}\sin^{-1}\left(\frac{\left|\mathbf{v}_\infty^- \times \mathbf{v}_\infty^+\right|}{\left|\mathbf{v}_\infty^-\right|\left|\mathbf{v}_\infty^+\right|}\right) = 2\sin^{-1}\left(\frac{1}{1 + r_p v_\infty^2 / \mu}\right)$$

where r_p is the periapsis radius of the flyby hyperbola, v_∞ is the magnitude of the incoming (or outgoing) v-infinity vector and μ is the gravitational constant of the flyby planet.

The maximum turn angle possible during a gravity assist flyby occurs when the spacecraft just grazes the planet's surface. It is given by

$$\delta_{\max} = 2\sin^{-1}\left(\frac{1}{1 + r_p v_\infty^2 / \mu}\right)$$

The semimajor axis and orbital eccentricity of the flyby hyperbola are given by

$$a = -\mu / \left|\mathbf{v}_\infty^-\right|^2 = -\mu / \left|\mathbf{v}_\infty^+\right|^2$$

$$e = -1/\cos\theta_\infty = 1 - \frac{r_p}{a} = 1 + \frac{r_p v_\infty^2}{\mu}$$

where θ_∞ is the true anomaly at infinity which is determined from the following expression:

$$\theta_\infty = \frac{\pi}{2} - \frac{1}{2}\sin^{-1}\left(\frac{\left|\mathbf{v}_\infty^- \times \mathbf{v}_\infty^+\right|}{\left|\mathbf{v}_\infty^-\right|\left|\mathbf{v}_\infty^+\right|}\right)$$

The periapsis radius of the flyby hyperbola is determined from the expression $r = a(1-e)$ and the flyby altitude is $h = r - r_s$ where r_s is the radius of the flyby planet.

The heliocentric speed gained during the flyby and the heliocentric delta-v vector caused by the close encounter can be determined from the following two equations:

$$\Delta v_{fb} = 2v_\infty / e$$

$$\Delta \mathbf{v}_{fb} = \mathbf{v}_h^- - \mathbf{v}_h^+$$

where *e* is the orbital eccentricity of the hyperbolic flyby trajectory.

In the second equation $\mathbf{v}_h^-$ is the heliocentric velocity vector of the spacecraft prior to the flyby and $\mathbf{v}_h^+$ is the heliocentric velocity vector after the flyby. For any planet it can be shown that the *maximum* heliocentric delta-v possible from a close flyby is given by the expression

$$\Delta v_{\max} = \sqrt{\frac{\mu}{r_s}}$$

This corresponds to a "grazing" flyby at the planet's surface and is equal to the "local circular velocity" for the flyby planet at the surface.

During the optimization analysis, the software enforces the following nonlinear constraints

$$\left|\mathbf{v}_\infty^-\right| - \left|\mathbf{v}_\infty^+\right| = 0$$

$$h_l \le h_{fb} \le h_u$$

The first equation is the v-infinity matching equality constraint and the second equation defines the flyby altitude lower and upper bounds. In the second expression h_{fb} is the actual flyby altitude computed by the software, h_l is the user-defined lower bound, and h_u is the user-defined upper bound for the flyby altitude. To enforce a "fixed" flyby altitude, set $h_l = h_u$.

Departure trajectory

The orientation of the departure hyperbola is specified in terms of the right ascension and declination of the asymptote of the departure hyperbola. These coordinates can be calculated using the components of the planet-centered departure unit v-infinity vector.

The inertial right ascension of the asymptote is determined from $\alpha = \tan^{-1}\left(\hat{v}_{\infty_y}, \hat{v}_{\infty_x}\right)$ and the geocentric declination of the asymptote is given by $\delta = 90^0 - \cos^{-1}\left(\hat{v}_{\infty_z}\right)$ where $\hat{v}_{\infty_x}, \hat{v}_{\infty_y}, \hat{v}_{\infty_z}$ are the Cartesian components of the v-infinity unit vector.

In a typical "targeting spec" for an interplanetary mission, the right ascension is called RLA and the declination is called DLA. The corresponding departure energy is called C3 which is equal to twice the specific (per unit mass) orbital energy of the trajectory. The actual *geocentric* hyperbolic injection delta-v depends on the orbital characteristics of the Earth park orbit.

In this MATLAB script the heliocentric planetary coordinates and therefore the $\Delta\mathbf{v}$ vectors are computed in the J2000 mean ecliptic and equinox coordinate system. In order to determine the orientation of the geocentric departure hyperbola, the heliocentric delta-v vector must be transformed to the Earth mean equator and equinox of J2000 (EME2000) frame.

The required transformation is given by

$$\Delta\mathbf{v}_{eq} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varepsilon & -\sin\varepsilon \\ 0 & \sin\varepsilon & \cos\varepsilon \end{bmatrix} \Delta\mathbf{v}_{ec} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.917482062069182 & \text{-}0.397777155931914 \\ 0 & 0.397777155931914 & 0.917482062069182 \end{bmatrix} \Delta\mathbf{v}_{ec}$$

where $\Delta\mathbf{v}_{ec}$ is the delta-velocity vector in the ecliptic frame, $\Delta\mathbf{v}_{eq}$ is the delta-velocity vector in the Earth equatorial frame and ε is the mean obliquity of the ecliptic at J2000.

The J2000 value of the mean obliquity of the ecliptic is equal to $\varepsilon = 23^\circ 26' 21.'' 448$. The conversion of coordinates from the equatorial system to the ecliptic system involves the transpose of this fundamental matrix. Appendix E provides information about the coordinate and time systems used in this script.

Calculating sphere-of-influence entrance and exit

This section describes the numerical methods used to predict the time at which the spacecraft enters or exits the sphere-of-influence (SOI) of the flyby planet.

The sphere-of-influence *objective function* is given by $\Delta = \left|\mathbf{r}_{p-sc}\right| - r_{SOI}$ where $\mathbf{r}_{p-sc}$ is the radius vector of the spacecraft relative to the flyby planet and r_{SOI} is the SOI radius of the flyby planet. The SOI radius is a function of the gravity of the flyby planet.

During the search for SOI conditions, the heliocentric equations of motion of the spacecraft subject to the point-mass gravity of the sun are defined by

$$\mathbf{a} = -\mu_s \frac{\mathbf{r}_{s-sc}}{\left|\mathbf{r}_{s-sc}\right|^3}$$

where μ_s is the gravitational constant of the sun and $\mathbf{r}_{s-sc}$ is the heliocentric position vector from the sun to the spacecraft.

The following is the MATLAB main script source code that predicts SOI entrance and exit. It uses the root-finder or *events* option of the built-in `ode45` function to evaluate the two-body heliocentric equations of motion defined in a MATLAB function named `twobody_eqm` while searching for the time at which the flyby planet-to-spacecraft distance equals the SOI radius of the flyby planet.

```
% --------------------------------------------
% two-body integration of the trajectory from
% first impulse to SOI entrance at flyby planet
% --------------------------------------------

% set up for ode45

options = odeset('RelTol', 1.0e-11, 'AbsTol', 1.0e-12, 'Events', @soi_entrance);

% define maximum search time (seconds)

tof = 86400.0 * (jdtdb2 - jdtdb1);

% -------------------------------------------------
% solve for flyby planet SOI entrance conditions
% two-body forward integration of the trajectory
% from first impulse to SOI entrance at flyby planet
% -------------------------------------------------

[~, ~, tevent, ~, ~] = ode45(@two_body_heqm, [0 tof], [rito1 vito1], options);
```

The following is the MATLAB source code that computes the scalar value of the difference between the current planet-centered radius and the value of the planet's SOI radius.

```
function [value, isterminal, direction] = soi_event(t, y)

% entrance to SOI of flyby planet event function

% input

%  t = time since departure (seconds)
%  y = spacecraft heliocentric state vector at time t
%      (kilometers and kilometers/second)

% output

%  value = delta-SOI (kilometers)

% Orbital Mechanics with MATLAB

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
global ip2 rsoi jdtdb1 rp2sc vp2sc

% current TDB julian date

jdtdb = jdtdb1 + t / 86400.0;

% compute position and velocity of the flyby planet
% (kilometers and kilometers/second)

[rfbp, vfbp] = p2000_ecl(ip2, jdtdb);

rsc = y(1:3);

vsc = y(4:6);

% state vector from flyby planet to spacecraft
% (kilometers and kilometers/second)

rp2sc = rfbp - rsc;

vp2sc = vfbp - vsc;

% objective function (delta-SOI distance; kilometers)

value = norm(rp2sc) - rsoi(ip2);

isterminal = 1;

direction = [];
```

In this code, `rp2sc` and `vp2sc` represent the flyby planet-centered position and velocity vectors at entrance to the SOI.

SNOPT algorithm implementation

This section provides details about the part of the `flyby_snopt` MATLAB script that solves this nonlinear programming (NLP) problem using SNOPT. In this classic trajectory optimization problem, the departure, flyby and arrival calendar dates are the bounded *control variables* and the user-specified scalar ΔV is the *objective function* or *performance index*. The numerical solution must also satisfy the flyby altitude inequality mission constraint and the v-infinity matching equality constraint.

The algorithm requires an initial guess for the control variables. For this problem they are given by the following MATLAB source code

```
% number of control variables

ncv = 3;

xg = zeros(ncv, 1);

% number of mission constraints

nmc = 2;

% control variables initial guesses
% (departure, flyby, and arrival tdb julian dates)
```

```
xg(1) = jdtdbi1 - jdtdb0;

xg(2) = jdtdbi2 - jdtdb0;

xg(3) = jdtdbi3 - jdtdb0;
```

where `jdtdbi1`, `jdtdbi2` and `jdtdbi3` are the initial user-provided departure, flyby and arrival calendar date guesses, and `jdtdb0` is a reference Julian date equal to `2451544.5` (January 1, 2000 TDB). This offset value is used to *scale* the control variables.

The algorithm also requires lower and upper bounds for the control variables, objective function and mission constraints. These are determined from the user-defined initial guesses and search boundaries.

```
% bounds on control variables

xlwr = zeros(ncv, 1);
xupr = zeros(ncv, 1);

xlwr(1) = xg(1) - ddays1;
xupr(1) = xg(1) + ddays1;

xlwr(2) = xg(2) - ddays2;
xupr(2) = xg(2) + ddays2;

xlwr(3) = xg(3) - ddays3;
xupr(3) = xg(3) + ddays3;

% bounds on objective function

flow = zeros(nmc + 1, 1);
fupp = zeros(nmc + 1, 1);

flow(1) = 0.0;
fupp(1) = +inf;

% "normalized" bounds on flyby altitude inequality constraint

flow(2) = fbalt_lwr / req(ip2);
fupp(2) = fbalt_upr / req(ip2);

% bounds on v-infinity matching (equality) constraint

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

where `ddays1`, `ddays2` and `ddays3` are the user-defined departure, flyby and arrival search boundaries, respectively.

The call to the SNOPT interface function is as follows

```
[x, ~, ~, ~, ~] = snopt(xg, xlwr, xupr, xmul, xstate, ...
    flow, fupp, fmul, fstate, 'flyby_func');
```

where `flyby_func` is the name of the MATLAB function that solves Lambert's problem for each leg of the interplanetary transfer trajectory, computes the current value of the objective function and evaluates the current v-infinity match and flyby altitude bounds for the gravity-assist portion of the problem.

Algorithm resources

"Gravity-Assist Trajectory Design with MATLAB", C. David Eagle, AAS 99-138, AAS/AIAA Space Flight Mechanics Meeting, Breckenridge, CO, February 7-10, 1999.

"Optimal Interplanetary Orbit Transfers by Direct Transcription", John T. Betts, *The Journal of the Astronautical Sciences*, Vol. 42, No. 3, July-September 1994, pp. 247-268.

"Multiple Gravity Assist Interplanetary Trajectories", A. V. Labunsky, O. V. Papkov, and K. G. Sukhanov, Gordon and Breach Science Publishers, 1998.

"Modern Astrodynamics", Victor R. Bond and Mark C. Allman, Princeton University Press, 1996.

"Automated Design of Gravity-Assist Trajectories to Mars and the Outer Planets", J.M. Longuski and S.N. Williams, *Celestial Mechanics and Dynamical Astronomy*, **52**: 207-220, 1991.

"Gravitational Assist in Celestial Mechanics – A Tutorial", J. A. Van Allen, American Journal of Physics, **71** (5), May 2003.

"Notes for the Gravitational Assisted Trajectories Lectures", E. Barrabes, G. Gomez, and J. Rodriquez-Canabal, Advanced Topics in Astrodynamics Summer Course, Barcelona, July 2004.

"Error Analysis of Multiple Planet Trajectories", F. M. Sturms Jr., JPL Space Programs Summary, No. 37-27, Vol. IV.

"Trajectory Analysis of a 1970 Mission to Mercury via a Close Encounter with Venus", E. Cutting, F. M. Sturms Jr., *AIAA Journal of Spacecraft and Rockets*, Vol.3, No. 5, pp. 624-631, 1966.

"A Graphical Method for Gravity-assist Trajectory Design", Nathan Strange, James Longuski, Astrodynamics Specialist Conference, 2000.

"Mars Free Returns via Gravity Assist from Venus", Masataka Okutsu, James Longuski, Astrodynamics Specialist Conference, 2000.

"Application of Tisserand's Criterion to the Design of Gravity Assist Trajectories", James Miller, Connie Weeks, AIAA/AAS Astrodynamics Specialist Conference and Exhibit, 2002.

"Fast Mars Free-Returns via Venus Gravity Assist", Kyle M. Hughes, Peter J. Edelman, James Longuski, Mike Loucks, John P. Carrico, Dennis A. Tito, AIAA/AAS Astrodynamics Specialist Conference, 2014.

"Design of High-accuracy Multiple Flyby Trajectories Using Constrained Optimization", Dennis V. Byrnes and Larry E. Bright, AAS 95-307, 1995

"An Essay on the Application and Principle of Gravity-assist Trajectories for Space Flight", Victor C. Clarke, Jr., Jet Propulsion Laboratory, April 10, 1970.

"Optimization of Interplanetary Trajectories with Unpowered Planetary Swingbys", Carl G. Sauer, Jr., AAS 87-424, 1987.

APPENDIX A

Contents of the Simulation Summary

This appendix is a brief summary of the information contained in the simulation summary displays produced by the `flyby_snopt` software.

The simulation summary screen display contains the following information:

calendar date = calendar date of trajectory event

TDB time = TDB time of trajectory event

TDB julian date = julian date of trajectory event on TDB time scale

sma (au) = semimajor axis in astronomical unit

eccentricity = orbital eccentricity (non-dimensional)

inclination (deg) = orbital inclination in degrees

argper (deg) = argument of periapsis in degrees

raan (deg) = right ascension of the ascending node in degrees

true anomaly (deg) = true anomaly in degrees

arglat (deg) = argument of latitude in degrees. The argument of latitude is the sum of true anomaly and argument of perigee.

period (days) = orbital period in days

rx (km) = x-component of the spacecraft's position vector in kilometers

ry (km) = y-component of the spacecraft's position vector in kilometers

rz (km) = z-component of the spacecraft's position vector in kilometers

rmag (km) = scalar magnitude of the spacecraft's position vector in kilometers

vx (kps) = x-component of the spacecraft's velocity vector in kilometers per second

vy (kps) = y-component of the spacecraft's velocity vector in kilometers per second

vz (ksp) = z-component of the spacecraft's velocity vector in kilometers per second

vmag (kps) = scalar magnitude of the spacecraft's velocity vector in kilometers per second

deltav-x = x-component of the impulsive TCM velocity vector in meters/second

deltav-y = y-component of the impulsive TCM velocity vector in meters/second

deltav-z = z-component of the impulsive TCM velocity vector in meters/second

delta-v = scalar magnitude of the impulsive TCM delta-v in meters/seconds

b-magnitude = magnitude of the b-plane vector

b dot r = dot product of the b-vector and r-vector

b dot t = dot product of the b-vector and t-vector

theta = orientation of the b-plane vector in degrees

v-infinity = magnitude of incoming v-infinity vector in kilometers/second

r-periapsis = periapsis radius of incoming hyperbola in kilometers

decl-asy = declination of incoming v-infinity vector in degrees

rasc-asy = right ascension of incoming v-infinity vector in degrees

fpa = flight path angle in degrees

Appendix B

Numerical Solutions of Lambert's Problem

Lambert's problem is concerned with the determination of a two-body 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 functionally as $tof = tof\left(r_i + r_f, c, a\right)$.

From the following form of Kepler's equation

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

we can write

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

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 sun and difference identities:

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

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

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

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

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

the time equation becomes

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

From the elliptic relationships given by

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

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

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

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

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

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

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

With the use of the half angle formulas given by

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

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

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

A discussion about the angles α and β can be found in "Geometrical Interpretation of the Angles α and β in Lambert's Problem" by J. E. Prussing, AIAA *Journal of Guidance and Control*, Volume 2, Number 5, Sept.-Oct. 1979, pages 442-443.

Gooding's solution of Lambert's problem

The algorithm used in `flyby_snopt` is based on the numerical 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.

Gedeon's solution of Lambert's problem

Another practical numerical method for solving Lambert's problem is described in "A Practical Note on the Use of Lambert's Equation" by Geza Gedeon, *AIAA Journal*, Volume 3, Number 1, 1965, pages 149-150. 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. Additional information can also be found in G. S. Gedeon, "Lambertian Mechanics", Proceedings of the 12th International Astronautical Congress, Vol. I, 172-190.

The *elliptic* form of the general Lambert Theorem is

$$t = \sqrt{\frac{a^3}{\mu}}\left[(1-k)m\pi + k(\alpha - \sin\alpha) \mp (\beta - \sin\beta)\right]$$

where k may be either +1 (posigrade) or –1 (retrograde), and m is the number of revolutions about the central body.

The Gedeon algorithm introduces the following parameter

$$z = \frac{s}{2a}$$

and solves the problem with a Newton-Raphson procedure. In this equation, a is the semimajor axis of the transfer orbit and

$$s = \frac{r_1 + r_2 + c}{2}$$

This algorithm also makes use of the following constant

$$w = \pm\sqrt{1 - \frac{c}{s}}$$

The function to be solved iteratively is given by:

$$N(z) = \frac{1}{z|z|^{1/2}\,2^{1/2}}\left\{\frac{1-k}{2}m\pi + k\left[|z|^{1/2} - |z|^{1/2}(1-z)^{1/2}\right] - \left[w|z|^{1/2} - w|z|^{1/2} - w|z|^{1/2}\left(1-w^2 z\right)^{1/2}\right]\right\}$$

The Newton-Raphson algorithm also requires the derivative of this equation given by

$$N'(z) = \frac{dN}{dz} = \frac{1}{|z|2^{1/2}}\left\{\frac{k}{(1-z)^{1/2}} - \frac{w^3}{\left(1-w^2 z\right)^{1/2}} - \frac{3N(z)}{2^{1/2}}\right\}$$

The iteration for z is as follows

$$z_{n+1} = z_n - \frac{N(z_n)}{N'(z_n)}$$

Shooting method with state transition matrix updates

An initial guess for this algorithm is created by first solving the two-body form of Lambert's problem. At each *shooting* iteration, the initial delta-velocity vector is updated according to

$$\Delta\mathbf{V} = \left[\Phi_{12}\right]^{-1} \Delta\mathbf{r}$$

where the error in the final position vector $\Delta\mathbf{r}$ is determined from the difference between the two-body final position vector $\mathbf{r}_{tb}$ and the final position vector predicted by numerical integration $\mathbf{r}_{\mathrm{int}}$ of the orbital equations of motion as follows

$$\Delta\mathbf{r} = \mathbf{r}_{tb} - \mathbf{r}_{\mathrm{int}}$$

The new initial velocity vector can now be calculated from

$$\mathbf{V}_{n+1} = \mathbf{V}_n + \Delta\mathbf{V}$$

The sub-matrix Φ_{12} of the full state transition matrix is as follows:

$$\Phi_{12} = \left[\frac{\partial\mathbf{r}}{\partial\mathbf{V}_0}\right] = \begin{bmatrix} \partial x / \partial\dot{x}_0 & \partial x / \partial\dot{y}_0 & \partial x / \partial\dot{z}_0 \\ \partial y / \partial\dot{x}_0 & \partial y / \partial\dot{y}_0 & \partial y / \partial\dot{z}_0 \\ \partial z / \partial\dot{x}_0 & \partial z / \partial\dot{y}_0 & \partial z / \partial\dot{z}_0 \end{bmatrix}$$

This sub-matrix consists of the partial derivatives of the rectangular cartesian components of the final position vector with respect to the initial velocity vector.

Nonlinear programming solution of Lambert's problem

In this classic trajectory optimization problem, the components of the initial and final delta-v vectors are the *control variables* and the scalar magnitude of the flyby or rendezvous ΔV is the *objective function* or *performance index*. The NLP implementation uses the two-body solution for Lambert's problem as its initial guess.

For the flyby problem, this method attempts to match all three components of the position vector. For the rendezvous problem, the NLP attempts to match all three components of both the target position and velocity vectors. These mission requirements are formulated as equality constraints.

APPENDIX C

Classical Orbital Elements of a Hyperbolic Flyby Orbit

This appendix describes the MATLAB function and mathematical equations used to determine the classical orbital elements of a hyperbolic flyby orbit at periapsis of the trajectory. The inputs to this MATLAB function are the incoming and outgoing v-infinity velocity vectors, the periapsis radius of the flyby hyperbola, and the gravitational constant of the flyby planet.

The syntax of this MATLAB function is

```
function oev = fbhyper(mu, vinfi, vinfo, rp)

% classical orbital elements of a planet-centered
% flyby hyperbola at periapsis passage

% input

%  mu    = flyby planet gravitational constant (kilometers^3/seconds^2)
%  vinfi = incoming v-infinity vector (kilometers/second)
%  vinfo = outgoing v-infinity vector (kilometers/second)
%  rp    = planet-centered periapsis distance (kilometers)

% output

%  oev(1) = semimajor axis (kilometers)
%  oev(2) = orbital eccentricity (non-dimensional)
%  oev(3) = orbital inclination (radians)
%  oev(4) = argument of perigee (radians)
%  oev(5) = right ascension of ascending node (radians)
%  oev(6) = true anomaly (radians)
```

Algorithm equations

The semimajor axis can be determined from the gravitational constant of the flyby planet and the scalar magnitude of the flyby v-infinity vector according to the expression

$$a = -\frac{\mu}{v_\infty^2}$$

The orbital eccentricity of the flyby hyperbola is given by

$$e = 1 - \frac{r_p}{a} = 1 + \frac{r_p v_\infty^2}{\mu}$$

A unit vector parallel to the incoming asymptote is given by this next equation

$$\hat{\mathbf{s}} = \frac{\mathbf{v}_\infty^-}{\left|\mathbf{v}_\infty^-\right|}$$

A unit vector perpendicular to the orbit plane of the flyby hyperbola is calculated from

$$\hat{\mathbf{w}} = \frac{\mathbf{v}_\infty^- \times \mathbf{v}_\infty^+}{\left|\mathbf{v}_\infty^- \times \mathbf{v}_\infty^+\right|}$$

The orbital inclination relative to the ecliptic plane is determined from the z-component of this unit vector as follows

$$i = \cos^{-1}\left(w_z\right)$$

A unit vector along the z-axis of the ecliptic coordinate system is given by

$$\hat{\mathbf{u}}_z = \begin{pmatrix}0 & 1 & 1\end{pmatrix}^T$$

From a unit vector in the direction of the ascending node given by

$$\hat{\mathbf{n}} = \hat{\mathbf{u}}_z \times \hat{\mathbf{w}}$$

the right ascension of the ascending node can be computed using a four-quadrant inverse tangent function as follows

$$\Omega = \tan^{-1}\left(n_y, n_x\right)$$

A unit vector in the direction of the periapsis of the flyby hyperbola is given by

$$\hat{\mathbf{p}} = -\cos\theta_\infty \hat{\mathbf{s}} + \sin\theta_\infty \hat{\mathbf{b}}$$

where

$$\cos\theta_\infty = -1/e$$

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

and $\hat{\mathbf{b}} = \hat{\mathbf{s}} \times \hat{\mathbf{w}}$. In these expressions θ_∞ is the orbital true anomaly at infinity.

The sine and cosine of the argument of periapsis are given by

$$\cos\omega = \hat{\mathbf{n}} \cdot \hat{\mathbf{p}}$$

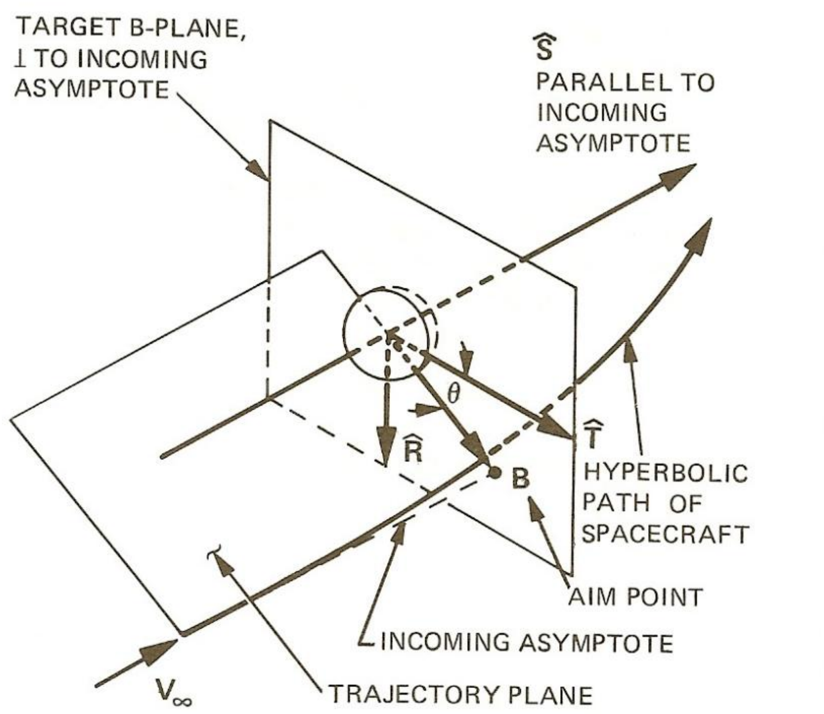
$$\sin\omega = \sqrt{1 - \cos^2\omega}$$

If $p_z < 0$ then $\sin\omega = -\sin\omega$. Finally, the argument of periapsis is determined from a four-quadrant inverse tangent function as $\omega = \tan^{-1}\left(\sin\omega, \cos\omega\right)$.

Appendix D

B-Plane Geometry and Coordinates

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 geometry of the B-plane coordinate system.



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 δ_∞ and α_∞ are the declination and right ascension of the asymptote of the incoming hyperbola at the arrival planet.

B-plane calculations

This section describes the conversion of inertial coordinates of an arrival or departure hyperbola to fundamental B-plane coordinates and vectors.

angular momentum vector

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

radius rate

$$\dot{r} = \mathbf{r} \bullet \mathbf{v} / |\mathbf{r}|$$

semi-parameter

$$p = h^2 / \mu$$

semimajor axis

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

orbital eccentricity

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

true anomaly

$$\cos v = \frac{p - r}{e r} \qquad \sin v = \frac{\dot{r} h}{e \mu}$$

B-plane magnitude

$$B = r_p \sqrt{1 + \frac{2\mu}{r_p V_\infty^2}} = \frac{\mu}{V_\infty^2} \sqrt{\left(1 + V_\infty^2 \frac{r_p}{\mu}\right)^2 - 1}$$

fundamental vectors

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

$$\hat{\mathbf{p}} = \cos\theta\,\hat{\mathbf{r}} - \sin\theta\,\hat{\mathbf{z}} \qquad \hat{\mathbf{q}} = \sin\theta\,\hat{\mathbf{r}} + \cos\theta\,\hat{\mathbf{z}}$$

S vector

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

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

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

R vector

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

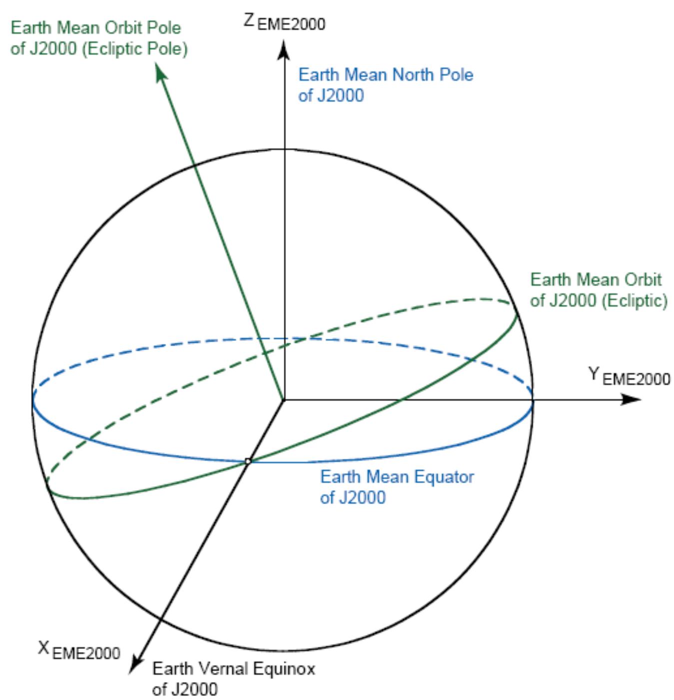
where **r** and **v** are the spacecraft inertial position and velocity vectors and μ is the planet's gravitational constant.

Appendix E

Coordinate and Time Systems

The following figure illustrates the geometry of the Earth mean equator and equinox (EME2000 or J2000) coordinate system. The origin of this Earth-centered-inertial (ECI) coordinate system is the geocenter and the fundamental plane is the Earth's mean equator.

The z-axis of this system is normal to the Earth's mean equator at epoch J2000, the x-axis is parallel to the vernal equinox of the Earth's mean orbit at epoch J2000, and the y-axis completes the right-handed coordinate system. The epoch J2000 is the Julian day 2451545.0 which corresponds to January 1, 2000, 12 hours Terrestrial Time (TT).



Transformation between the J2000 ecliptic and equatorial coordinate systems.

The required matrix-vector transformation is given by

$$\mathbf{V}_{eq} = \begin{bmatrix} 1 & -0.000000479966 & 0 \\ 0.000000440360 & 0.917482137087 & 0.397776982902 \\ -0.000000190919 & -0.397776982902 & 0.917482137087 \end{bmatrix} \mathbf{V}_{ec}$$

where $\mathbf{V}_{ec}$ is a vector in the ecliptic frame, and $\mathbf{V}_{eq}$ is a vector in the equatorial frame. The conversion of equatorial to ecliptic coordinates involves the transpose of this matrix.

Terrestrial Time, TT

Terrestrial Time is the time scale that would be kept by an ideal clock on the geoid - approximately, sea level on the surface of the Earth. Since its unit of time is the SI (atomic) second, TT is independent of the variable rotation of the Earth. TT is meant to be a smooth and continuous "coordinate" time scale independent of Earth rotation. In practice TT is derived from International Atomic Time (TAI), a time scale kept by real clocks on the Earth's surface, by the relation TT = TAI + $32^s.184$. It is the time scale now used for the precise calculation of future astronomical events observable from Earth.

TT = TAI + 32.184 seconds

TT = UTC + (number of leap seconds) + 32.184 seconds

In the second equation, UTC is Universal Coordinated Time.

Barycentric Dynamical Time, TDB

Barycentric Dynamical Time is the time scale that would be kept by an ideal clock, free of gravitational fields, co-moving with the solar system barycenter. It is always within 2 milliseconds of TT, the difference caused by relativistic effects. TDB is the time scale now used for investigations of the dynamics of solar system bodies.

TDB = TT + periodic corrections

where typical periodic corrections (*USNO Circular 179*) are

$$
\begin{aligned}
TDB = TT &+ 0.001657\sin\left(628.3076T + 6.2401\right) \\
&+ 0.000022\sin\left(575.3385T + 4.2970\right) \\
&+ 0.000014\sin\left(1256.6152T + 6.1969\right) \\
&+ 0.000005\sin\left(606.9777T + 4.0212\right) \\
&+ 0.000005\sin\left(52.9691T + 0.4444\right) \\
&+ 0.000002\sin\left(21.3299T + 5.5431\right) \\
&+ 0.000010T\sin\left(628.3076T + 4.2490\right) + \cdots
\end{aligned}
$$

In this equation, the coefficients are in seconds, the angular arguments are in radians, and T is the number of Julian centuries of TT from J2000; T = (Julian day(TT) – 2451545.0) / 36525.